EMPLOYEE MANAGEMENT SYSTEM
FOR MINISTRY OF INDUSTRY

ACKNOWLEDGMENT

We would like to express our sincere gratitude to the Ministry of Industry for their support and collaboration throughout the development of the Employee Management System. Their insights, guidance, and feedback were invaluable in shaping the system to meet the organization's needs and objectives.

We also extend our appreciation to all stakeholders involved in this project, including executives, managers, and HR professionals, for their active participation and contribution. Their input and expertise were instrumental in defining requirements, validating functionalities, and ensuring the system's success.

Furthermore, we acknowledge the dedication and hard work of the project team members, including developers, UI/UX designers, QA engineers, system administrators, and support staff. Their tireless efforts, technical skills, and commitment to excellence were crucial in delivering a robust and comprehensive Employee Management System.

Last but not least, we thank all employees and end-users who provided valuable feedback and tested the system, contributing to its usability and effectiveness. Their engagement and feedback were essential in refining the system and ensuring its alignment with user needs.

This project would not have been possible without the collective effort, collaboration, and support of all stakeholders and team members involved. We look forward to continued success and innovation in workforce management with the Employee Management System.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVATIONS

EMS - Employee Management System

HR - Human Resource

ABSTRACT

The Employee Management System is a comprehensive web-based application designed for the Ministry of Industry to streamline workforce management processes. It focuses on enhancing operational efficiency, reducing administrative tasks, and promoting a productive work environment. Key features include employee information storage, attendance tracking, remote leave management, performance evaluations, and payroll processing. The system is accessible and user-friendly, ensuring consistency across devices and enabling data-driven decision-making. Security and integrity are paramount, with robust measures to protect sensitive information. The system's feasibility, including technical, operational, and economic aspects, is analyzed, showcasing its cost-effectiveness and achievable timeline. The scope encompasses scalable architecture, data security, integration capabilities, and user training, aiming to provide a reliable and efficient solution for HR processes.

CHAPTER 1

Introduction

1.1.Introduction

The Employee Management System is designed to simplify workforce management for the Ministry of Industry. It is a web-based application that enhances the storage of employee information, attendance tracking, and remote leave management to ensure smooth information flow. It also facilitates performance evaluations at the departmental level and payroll processing, making it accessible and usable for each employee as needed.

Ultimately, this system enables better resource management and effective time-tracking mechanisms, saving both time and money for the organization. The system simplifies usability by maintaining consistency and availability across devices. By integrating these functionalities into a single platform, our Employee Management System enhances operational efficiency, reduces administrative tasks, and promotes a productive work environment.

Integrity is a cornerstone of our system, ensuring that all data entered and processed remains accurate and reliable, which is vital for making informed decisions and maintaining trust within the organization. Privacy is meticulously maintained, with robust security measures in place to protect sensitive employee information from unauthorized access and potential breaches.

In summary, the proposed Employee Management System is a robust and comprehensive solution designed to meet the evolving needs of the Ministry of Industry. It not only enhances operational efficiency but also supports a productive and secure work environment, aligning with the modern expectations of technological systems.

1.2. Background

In today's fast-paced and technologically advanced world, systems are expected to meet the diverse needs of all users effectively. Modern systems must excel in availability, consistency, integrity, privacy, accessibility, and functionality to truly support the work of employees and enhance their productivity.

The proposed Employee Management System is designed with these critical aspects in mind, ensuring it meets the high standards required for contemporary workplace environments. By focusing on availability, the system guarantees that employees can access necessary information and tools at any time, from any location, thus fostering a flexible and dynamic work culture. Consistency in the system ensures that data and processes remain uniform across various departments and user levels, reducing errors and improving overall efficiency.

Accessibility is enhanced by designing a user-friendly interface that caters to employees at all levels, allowing them to navigate and utilize the system effortlessly. The functionality is comprehensive, covering essential aspects such as employee information management, attendance tracking, performance evaluations, payroll processing, and scheduling. Additionally, the system simplifies leave management, ensuring accurate tracking and compliance with leave policies.

The integration of these features into a single platform provides valuable opportunities for cost reduction, time savings, and improved communication transparency. By minimizing administrative tasks and streamlining processes, the system allows employees to focus more on their core responsibilities, ultimately leading to increased productivity and job satisfaction.

1.3. Statement of the Problem

The current system used by the company has several limitations:

- Attendance Tracking: The system employs a fingerprint-based attendance tracking system.
- Leave Management: The leave management system is not accessible to all employees and requires physical presence to request leave.
- Performance Evaluation:

  Personal performance evaluations are conducted in person.

  Departmental evaluations are carried out via a Telegram bot.

These constraints highlight the need for a more integrated and accessible system to streamline and enhance these processes.

1.4. Objectives of the Project

1.4.1 General Objective

The general objective of the Employee Management System project is to develop a comprehensive and integrated platform that simplifies and optimizes workforce management. This system aims to enhance operational efficiency, reduce administrative tasks, and promote a productive work environment by providing efficient and essential tools for managing employee information, attendance, performance, payroll and leave management.

1.4.2 Specific Objectives

1. **Enhance Performance Evaluations:** To provide tools for conducting thorough and unbiased performance reviews, helping in employee development and productivity.
2. **Simplify Leave Management:** To ensure accurate tracking of employee leave requests and balances, maintaining compliance with organizational leave policies.
3. **Enable Remote Access:** To provide remote access capabilities, allowing executives and managers to oversee workforce management from anywhere at any time.
4. **Reduce Administrative Burden:** To decrease the time and effort required for administrative tasks, allowing HR and management to focus on strategic initiatives.
5. **Promote Data-Driven Decision Making:** To integrate data analytics tools that support informed decision-making processes regarding workforce management.

1.5 Feasibility Analysis

1.5.1 Technical Feasibility

The Project Team Members have acquired the Programming language that is required for the successful completion of the Projects

The Programming language we used are CSS, JavaScript, HTML, , Python, Django

We also learned the spiral methodology for preparing a documentation and for organizing our team and for better effort.

1.5.2 Operational Feasibility

The system brings better achievement for the operations performed by American College of Technology by providing efficient registration, employee Information, attendance tracking, performance evaluation, payroll processing and scheduling. A system to manage employee records including leave management.

1.5.3 Economical feasibility

Since the proposed system focuses on enhancing the existing system, the financial outlay is relatively lower. The main costs involved include human resources, software tools, and materials. Here is a detailed estimate of these costs in Ethiopian Birr (ETB):

I. Human Resource Costs

Project Manager: ETB 30,000/month for 3 months

Total: ETB 90,000

Developers: 3 developers at ETB 25,000/month each for 3 months

Total: ETB 225,000

UI/UX Designer: ETB 20,000/month for 2 months

Total: ETB 40,000

QA Engineer: ETB 15,000/month for 2 months

Total: ETB 30,000

System Administrator: ETB 20,000/month for 2 months

Total: ETB 40,000

Total Human Resource Costs: ETB 425,000

II. Software Tools Costs

Development Tools and IDEs: ETB 30,000 (one-time)

Licenses for Software: ETB 50,000 (one-time)

Testing Tools: ETB 20,000 (one-time)

Total Software Tools Costs: ETB 100,000

III. Material Costs

Server Hardware Upgrades: ETB 50,000 (one-time)

Network Infrastructure: ETB 30,000 (one-time)

Miscellaneous Materials (stationery, etc.): ETB 10,000 (one-time)

Total Material Costs: ETB 90,000

Total Cost Estimate
Human Resource Costs: ETB 425,000
Software Tools Costs: ETB 100,000
Material Costs: ETB 90,000
Total Estimated Cost: ETB 615,000

This estimate provides a comprehensive overview of the costs involved in upgrading the Employee Management System, ensuring it aligns with the financial feasibility while enhancing the existing infrastructure.

1.5.4 Schedule Feasibility

The schedule feasibility for upgrading the Employee Management System is well-structured and achievable within a 3-month timeframe. The project involves key phases such as planning, development, testing, and deployment. During the first month, the project manager will coordinate with developers, the UI/UX designer, and the system administrator to finalize the project plan and initiate the development process. Over the subsequent two months, developers will work on integrating and enhancing the existing system functionalities, while the UI/UX designer ensures a user-friendly interface. Simultaneously, the QA engineer will conduct rigorous testing to identify and resolve any issues. The system administrator will oversee the hardware and network upgrades. This systematic approach ensures that each phase is completed efficiently, allowing for a smooth transition to the enhanced system within the proposed timeline. With dedicated resources and a focused strategy, the project is poised to meet its objectives on schedule, providing an upgraded, robust, and comprehensive Employee Management System.

1.6 Scope and limitations

1.6.1 Scope

The proposed employee management system is aimed to be developed to provide a comprehensive solution for managing the workforce effectively and efficiently. It will serve as a centralized platform for HR departments to streamline employee information management, attendance tracking, performance evaluation, payroll

processing, and leave management. The system will enhance productivity, optimize resource allocation, and improve decision-making processes related to workforce management. It will offer a user-friendly interface for HR administrators and employees to access and update relevant information, ensuring data accuracy and accessibility.The scope of the project includes designing a scalable architecture, ensuring data security, and compliance with regulations, providing integration capabilities with other systems, and offering training and support for users to maximize system utilization and benefits. Collaboration with HR professionals, software developers, and end-users will be essential to gather requirements, validate functionalities, and ensure successful implementation and adoption of the employee management system.

Data Security and Access Control: The system will ensure data security by securely storing employee data and restricting access to authorized personnel only. This will help in maintaining the confidentiality and integrity of sensitive information.

Scalability: The system will be designed to handle an increasing number of employees as the organization grows. Performance degradation will be minimal, ensuring that the system can accommodate the company's growth without compromising efficiency. By focusing on data security and scalability, the employee management system will provide a reliable and efficient solution for managing HR processes.

1.6.2 Limitations

1. User Interface Complexity: The system will not have a complex user interface. The focus will be on keeping the interface simple and easy to use for the users.
2. Third-Party Integration: This refers to the system's limitation in integrating with all third-party tools. Only essential services like time-tracking devices and payroll systems will be integrated with the system.
3. Customization Constraints: This indicates that there will be limited options for customization in the system. Users will not be able to make extensive modifications to predefined workflows or rules.
4. Reporting Complexity: The system will not support complex custom reports. While standard payroll reports will be generated, highly specialized reports may require additional development.

5. Cost: The system will have a set budget for development and maintenance. Additional features beyond the initial scope may incur extra costs and require budget approval.

1.7 Significance of the Project

The upgrade of the Employee Management System holds significant value for the Ministry of Industry by addressing the current system's limitations and enhancing its overall functionality.

This upgrade will ensure that all employees, regardless of their level, can access essential features such as leave requests and personal performance evaluations remotely, thereby eliminating the need for physical presence and in-person evaluations.

The improved system will enable better resource management, ensuring that employee information is consistently available and accurate, which is crucial for informed decision-making. Enhanced time-tracking mechanisms will lead to more efficient use of time and resources, ultimately saving money for the organization. Additionally, the focus on integrity and privacy with robust security measures will protect sensitive employee data, maintaining trust and compliance with data protection regulations.

By reducing administrative tasks and improving operational efficiency, the upgraded system will promote a more productive work environment. The comprehensive approach to usability and accessibility ensures that employees can interact with the system across various devices seamlessly, further enhancing their work experience. Overall, this project will not only modernize the existing infrastructure but also support the Ministry's strategic goals by providing a reliable, efficient, and secure employee management solution.

1.8. Beneficiaries of the system

The upgraded Employee Management System will benefit various stakeholders within the Ministry of Industry, making it a valuable asset for the organization. Employees at all levels will experience improved efficiency and convenience in managing their attendance, leave requests, and performance evaluations through the system's user-friendly interface and remote accessibility. HR personnel will benefit from streamlined processes for profile management, payroll

processing, and data analysis, enabling them to make informed decisions and allocate resources effectively. Managers and department heads will have access to real-time performance data and analytics, facilitating better workforce planning and decision-making. Additionally, the organization as a whole will benefit from enhanced resource management, reduced administrative tasks, and improved compliance with data security and privacy regulations, leading to a more productive and efficient work environment.

1.9 Methodology

1. Data Collection Method

Effective data collection is crucial for ensuring the system meets user needs and requirements. Methods include:

Interviews: Conducting one-on-one or group interviews with stakeholders to gather detailed requirements and understand workflows and functionalities.

Observation: Observing current processes to identify inefficiencies and areas for improvement.

2. System Analysis and Design Methodology

The methodology chosen for system analysis and design ensures a structured approach to development. A suitable methodology for this project is Iterative model due to its iterative nature and flexibility.

Requirements Gathering: Collecting and prioritizing requirements through user stories and backlog grooming.

System Design: Using UML diagrams (use case, class, sequence diagrams) to design system architecture.

Implementation: Developing the system in iterative sprints, with regular feedback and updates from stakeholders.

Testing: Continuous testing (unit, integration, and user acceptance testing) throughout the development cycle.

Deployment: Incremental deployment to ensure smooth transition and minimize disruption.

Maintenance: Ongoing support and updates based on user feedback and system performance.

3. CASE Tools

Computer-Aided Software Engineering (CASE) tools facilitate the design and development process. Recommended CASE tools include:

- UML Tools (Draw.io, LucidChart): Used for creating and managing UML diagrams to visualize and design software architecture.
- Project Management Tools (Jira, Trello): Used for Agile project management, task tracking, and collaboration among team members.
- Version Control (GitHub): While GitHub is also a version control system, it's included here as a case tool for its project management features such as issue tracking and project boards.
- Integrated Development Environment (IDE): Not specifically mentioned in the case tools list, as IDEs like Visual Studio and IntelliJ IDEA are more focused on coding and debugging rather than project management or documentation.
- Database Management (Microsoft SQL): While a database management system is listed, it's typically more associated with development than with Case tools, as it's used for designing and managing databases rather than for software documentation or management.

4. Requirement Specification Testing

Testing ensures that the system meets both functional and non-functional requirements. The testing phases include:

Unit Testing: Verifying individual components and functions (e.g., authentication, profile management).

Integration Testing: Ensuring that different modules (e.g., attendance, payroll) work together seamlessly.

System Testing: Testing the entire system for compliance with the specified requirements.

User Acceptance Testing (UAT): Validating the system with actual users to ensure it meets their needs and expectations.

1.10. Development tools and Technologies

The upgrade of the Employee Management System will leverage modern development tools and technologies to ensure a robust and efficient solution. Key tools and technologies include:

- Programming Languages: The system will be developed using languages such as Python, Django, HTML, CSS, and JavaScript for backend and frontend development.
- Database Management: PostgreSQL or MySQL will be used for database management to store employee information, attendance records, leave requests, and performance evaluations securely.
- Version Control: Git and GitHub will be utilized for version control, enabling collaborative development, code management, and tracking of changes.
- Integrated Development Environment (IDE): Development will be carried out using popular IDEs such as PyCharm or Visual Studio Code, providing developers with a productive coding environment and debugging tools.
- Testing Frameworks: Unit testing will be implemented using tools like pytest for back-end testing and Jest or Selenium for front-end testing, ensuring code quality and functionality.

1.11. Assumptions and Constraints

- Integration: Constraints may arise during the integration of new functionalities with existing systems or third-party services, requiring thorough testing and debugging.
- Security: Assumes that robust security measures, such as encryption, access controls, and regular security audits, will be implemented to protect sensitive employee data and ensure compliance with data protection regulations.
- Budget and Timeline: Assumes that the project budget and timeline are sufficient to cover development, testing, deployment, and training phases without major deviations or delays.
- Data Migration: Assumed that existing data from the current Employee Management System can be migrated seamlessly to the upgraded system without significant data loss or inconsistencies.

- Scalability and Load Constraints: The architecture should support horizontal scaling to handle increased load. Measure the number of transactions (e.g., profile updates, attendance records) the system can handle per unit of time and ensure it meets performance requirements.

CHAPTER 2
Description of the Existing System

2.1. Introduction

An Employee Management System (EMS) is a software solution designed to manage and streamline all aspects of employee information and administrative tasks within an organization. This system acts as a central repository for maintaining employee records, facilitating efficient HR operations, and ensuring seamless communication and collaboration between different departments.

1. **Centralized Data Management**: EMS centralizes all employee-related information, including personal details, job roles, performance records, and payroll data. This ensures that accurate and up-to-date information is readily accessible to authorized personnel.
2. **Enhanced Efficiency**: By automating routine HR tasks such as attendance tracking, leave management, and payroll processing, EMS significantly reduces manual effort and the likelihood of errors. This allows HR professionals to focus on more strategic initiatives.
3. **Improved Communication**: EMS fosters better communication within the organization by providing tools for internal messaging, announcements, and feedback collection. It enables employees and management to stay informed and connected.
4. **Compliance and Reporting**: EMS helps organizations comply with labor laws and regulations by maintaining accurate records and generating necessary reports. This includes compliance with tax laws, labor regulations, and industry-specific standards.
5. **Performance Management**: EMS facilitates the performance review process by providing tools for setting goals, tracking progress, and conducting

evaluations. This helps in identifying high performers and areas needing improvement.

Components of an Employee Management System

1. **Employee Database**: Stores comprehensive employee information, including personal details, contact information, job titles, and employment history.
2. **Attendance and Leave Management**: Tracks employee attendance, manages leave requests, and ensures compliance with company policies.
3. **Payroll Management**: Automates salary calculations, tax deductions, and benefits administration, ensuring timely and accurate payroll processing.
4. **Performance Appraisal**: Manages performance reviews, sets employee goals, and tracks performance metrics to support career development and promotions.
5. **Employee Self-Service**: Provides a portal for employees to access their information, submit leave requests, view payslips, and update personal details.

Benefits of Implementing an Employee Management System

- **Increased Productivity**: Automation of administrative tasks frees up HR staff to focus on strategic activities, enhancing overall productivity.
- **Cost Savings**: Reduces the need for paper-based processes and minimizes errors, leading to cost savings in the long run.
- **Improved Accuracy**: Centralized data and automated processes ensure higher accuracy in employee records, payroll, and compliance reporting.
- **Enhanced Employee Experience**: Self-service portals and efficient HR processes contribute to a positive employee experience, leading to higher satisfaction and retention rates.
- **Data-Driven Decisions**: Access to real-time data and analytics enables informed decision-making and strategic planning.

2.2 Players in existing system

In an Employee Management System (EMS), the term "players" typically refers to the key stakeholders or entities involved in the system. These players are the individuals

or groups that interact with or are affected by the EMS. Here are the primary players in an existing Employee Management System:

1. Employees

- **General Staff**: Regular employees who use the system to view and update their personal information, check their attendance, apply for leaves, and access their payroll information.
- **Managers**: Employees who have additional responsibilities, such as approving leave requests, conducting performance evaluations, and managing team schedules.
- **Executives**: Senior-level employees who might use the system for high-level reports and analytics related to overall company performance, employee satisfaction, and departmental efficiency.

2. Human Resources (HR) Department

- **HR Managers**: Oversee the recruitment, onboarding, and offboarding processes. They manage employee records, handle grievances, and ensure compliance with labor laws.
- **Payroll Administrators**: Responsible for managing employee compensation, benefits, tax deductions, and issuing paychecks.

3. IT Department

- **System Administrators**: Maintain the EMS, ensuring it is secure, up-to-date, and running smoothly. They handle user access control and data backups.
- **Support Staff**: Provide technical support to users, resolve system issues, and ensure integration with other systems and software used by the company.

4. Finance Department

- **Accountants**: Use the EMS to track payroll expenses, manage budgets related to employee salaries and benefits, and generate financial reports.

5. Legal and Compliance Officers

- Ensure that the EMS complies with all relevant labor laws, data protection regulations, and industry standards. They might use the system to generate reports for audits and compliance checks.

6. External Entities

- **Vendors**: External service providers who might interact with the EMS for various purposes, such as background checks, payroll processing, and benefits administration.
- **Regulatory Bodies**: Government agencies that may require periodic reports or audits to ensure the company is complying with labor laws and regulations.

7. Customers and Clients

- Although not direct users of the EMS, the performance and satisfaction of employees (as managed through the EMS) can significantly impact customer service and client relationships.

8. Senior Management and Executives

- **CEO and Senior Executives**: Use the EMS for strategic decision-making, reviewing organizational performance, and planning for future workforce needs.
- **Department Heads**: Monitor and manage the performance and productivity of their departments, leveraging data from the EMS.

Typical Features and Interactions in EMS

- **Employee Self-Service Portal**: Employees access personal data, leave balances, and payroll information.
- **Manager Dashboard**: Managers approve leave requests, monitor team performance, and manage schedules.
- **HR Dashboard**: HR professionals manage employee records, recruitment processes, and compliance.
- **Reporting and Analytics**: Executives and managers generate reports to gain insights into workforce trends and performance.

- **Integration with Other Systems**: The EMS often integrates with other software systems, such as accounting software, time-tracking systems, and benefits administration platforms.

2.3 Business rules in existing system

In an employee management system, business rules define how decisions are made regarding various aspects of employee interactions. Here are some common business rules:

1) Leave Approval: If an employee requests leave, the system checks eligibility (e.g., accrued leave balance, company policy) and approves or denies accordingly.
2) Salary Calculation: Based on employee data (e.g., role, experience), the system calculates salaries, bonuses, and deductions.
3) Performance Reviews: The system schedules and conducts performance reviews, considering factors like goals, feedback, and ratings.

2.4. Report and Other documents

Below is a comprehensive list of such documents, categorized by their purpose within the software development life cycle(SDLC).

From planning phase

- Requirements Specification (SRS)

  Description: Detailed description of the software's functional and non-functional requirements.

  Components:

  - Functional Requirements
  - Non-Functional Requirements
  - User Stories

From testing phase

- Test Plan

  Description: Document outlining the strategy, resources, and schedule for testing activities.

  Components:

  - Test Objectives
  - Test Scope

- Test Environment
- Test Schedule
- Test Deliverables

2.5. Practices to be preserved from the existing system

Preserving effective practices from existing systems in software engineering is crucial for maintaining the quality, efficiency, and reliability of software development processes. Here are several best practices that should be preserved:

1. **Code Reviews**
   - **Description**: Regularly review code through peer reviews or automated tools.
   - **Benefits**:
     - Improve code quality
     - Identify bugs early
     - Share knowledge among team members
2. **Risk Management**
   - **Description**: Identify, assess, and mitigate risks throughout the project lifecycle.
   - **Benefits**:
     - Proactively address potential issues
     - Improve project stability
     - Ensure project success
3. **User-Centered Design**
   - **Description**: Focus on user needs and usability throughout the design and development process.
   - **Benefits**:
     - Enhance user satisfaction
     - Improve product usability
     - Increase adoption rates
4. **Regular Retrospectives**
   - **Description**: Conduct regular retrospectives to reflect on what's working and what's not.
   - **Benefits**:
     - Foster continuous improvement
     - Address team issues

- Adapt processes to better suit the team

5. **Security Practices**
   - **Description**: Integrate security best practices into the development lifecycle.
   - **Benefits**:
     - Protect sensitive data
     - Ensure compliance with regulations
     - Reduce vulnerabilities

Preserving these best practices ensures that software engineering processes remain robust, efficient, and capable of delivering high-quality software. By continuously adhering to these practices, teams can maintain consistency, improve productivity, and achieve better project outcomes.

2.6. Proposed System

Here are the proposed features and practices for an improved system:

Centralized Employee Database: Maintains comprehensive employee records including Employee ID, Name, Phone Number, Email Address, Position, and Department.

Automated Leave Management: Employees can request leave through the system, which is then reviewed and approved or rejected by the department head.

Employee Self-Service Portal: Allows employees to update personal information, view payroll details, and track attendance records.

Automated Payroll Calculations: Calculates base salary and deductions automatically.

Integration with Attendance Data: Fetches attendance details to accurately calculate overtime and deductions.

Real-Time Attendance Recording: Employees can mark attendance using biometric devices or web-based check-in/out.

Comprehensive Attendance Reports: Provides detailed reports on employee attendance, including late arrivals and early departures.

Performance Review Management: Departments can submit performance reports and receive feedback from HR.

User Authentication: Ensures secure access to the system through username and password authentication.

Role-Based Access Control: Restricts access to sensitive information based on user roles.

Data Encryption: Protects sensitive data through encryption mechanisms.

Benefits of the Proposed System

Improved Efficiency

Automates repetitive tasks, allowing HR staff to focus on strategic activities.

Enhanced Data Accuracy

Ensures up-to-date and accurate information through real-time updates.

Allows employees to access their information and track requests.

Improves communication between employees and HR.

Streamlines HR operations, leading to long-term cost savings.

2.7. Requirements of the new system

To ensure the system meets both functional and non-functional requirements, here is a detailed breakdown:

2.7.1. Functional Requirements

1. The system must authenticate users (e.g., employees, managers) and grant appropriate access levels based on roles (read-only, edit, admin).
2. The system must allow HR to create, update, and delete employee profiles.
3. It must store essential employee details such as name, contact information, job title, department, and hire date.
4. The system must record employee attendance, including clock-in and clock-out times
5. The system should allow employees to request leaves (sick leave, vacation, etc.).
6. It must handle exceptions (e.g., sick leave, vacation) and calculate total work hours.
7. Notifications must be sent to supervisors or managers for any unexcused absences.
8. The system must facilitate performance reviews and evaluations.
9. It must allow managers to set evaluation criteria, schedule reviews, and provide feedback.
10. Performance metrics as a department(e.g., goals achieved, time management) must be tracked.

11. The system must calculate employee salaries based on predefined rules (hourly, monthly, etc.).
12. It must consider factors like overtime, bonuses, and deductions.
13. Payroll reports must be generated for accounting purposes.

2.7.2. Non Functional Requirements

1) The system should ensure data security and access control to prevent unauthorized modifications
2) Employee data should be stored securely, and access should be restricted to authorized personnel only.
3) Maintain logs of user actions (e.g., who accessed which records, modifications made).
4) The system should respond quickly to user requests (e.g., loading employee profiles, generating reports).
5) As the organization grows, the system should handle an increasing number of employees without performance degradation.
6) Regular backups should be taken, and data recovery procedures should be in place.
7) Provide meaningful error messages to guide users when they encounter issues.
8) Validate input data (e.g., employee IDs, phone numbers) to prevent incorrect or incomplete information.
9) Ensure consistent data across different modules (e.g., employee profiles, attendance records).
10) The system should adhere to labor laws regarding employee data privacy, leave entitlements, and working hours.

CHAPTER 3

System Analysis

3.1. Introduction

This phase is crucial in defining the necessary requirements and features in order to implement user's need effectively. Key activities during this phase include the creation of UML (Unified Modeling Language) diagrams, which serve as essential tools for visualizing the system's structure and behavior. By drawing use case diagrams, class diagrams, sequence diagrams, and activity diagrams, we can clearly delineate system components, their interactions, and workflows. This structured approach not only facilitates better communication among stakeholders but also lays a solid foundation for subsequent design and implementation phases, ensuring the system's robustness, scalability, and maintainability.

3.2. Scenario of use case

Use case: Employee Make a leave request

Scenario 1: Successfully gets the leave requested

The employee submits the leave request and gets approved by the department head.

Scenario 2 : Leave request rejected

The employees leave request gets declined by the department head

Use case: HR evaluates department

Scenario 1:Successful evaluation

The HR team collects relevant data and performance metrics for the department. Based on the evaluation, the HR department provides feedback to the department head, highlighting strengths and areas for improvement

Scenario 2: Unsatisfactory evaluation

THE HR team identifies significant performance gaps or policy violations within the department.The HR department monitors the implementation of the improvement plan and re-evaluates the department's performance.

3.3. Use case Diagram

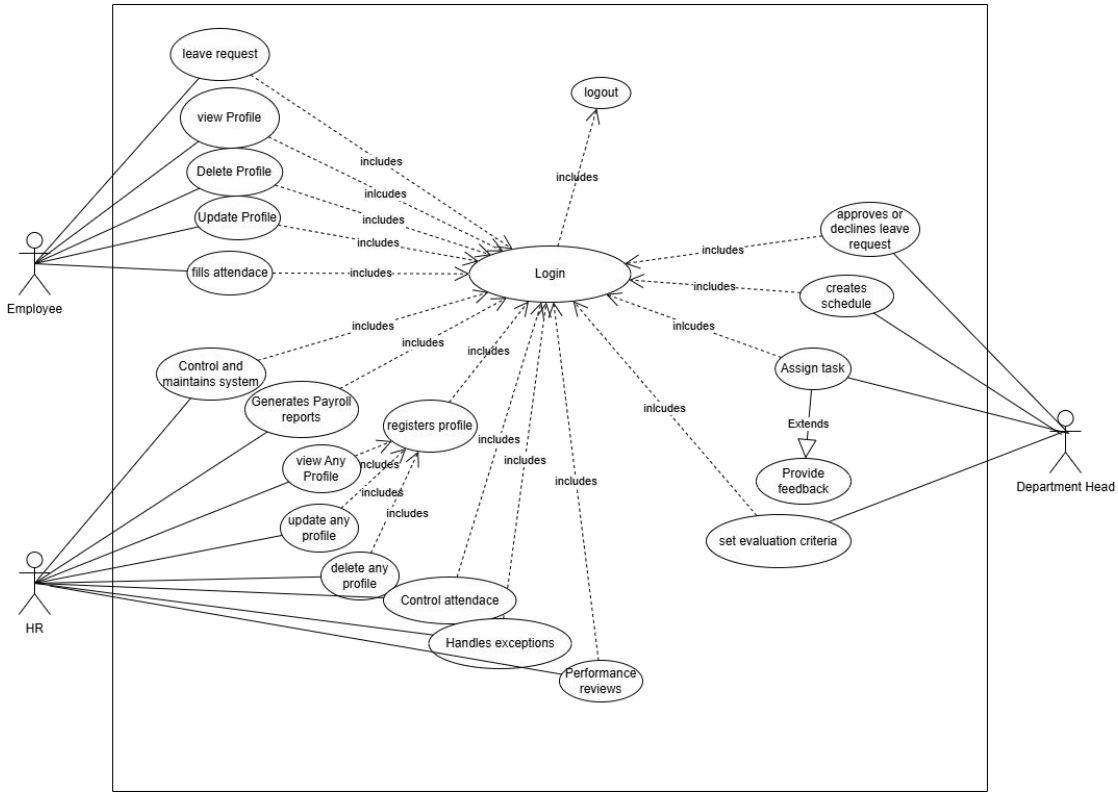


Fig 3.1. Use case Diagram of EMS

3.4 Use Case description

Usecase Name	Registration
Participating Actors	HR
Flow of Events	● The HR open the system ● The sign up menu displayed on the system ● The HR clocks the menu and the registration form is displayed ● Fill the complete information about employee's detail ● Click register button
Entry Condition	The employee must have complete information about their details
Exceptional Flow	If the HR enters an invalid information the system will notify it to enter a valid information or complete the whole information

Table 3.1. Use case description of Registration

Use case Name	Login
Participating Actors	Employee, HR, Department head
Flow of events	1. The actors opens the system 2. The login form is displayed on the index page 3. Employee and manager fill the correct username and password 4. Click login button 5. Logged successfully message is displayed 6. Redirected to their authorized page
Entry Condition	The HR or employee or Department head must be registered to the data base
Exceptional Flow	● If the employee or HR or department head does not fill the correct credentials. The system notifies the user to correct the information
Exit condition	● The registered user must be recorded in the database and returns a successful message

Table 3.2. Use case scenario of Login

Usecase name	Control Attendance
Participating Actors	HR
Flow of events	1. The HR opens the system and logs in using the correct username and password 2. The attendance menu is displayed on the HR's page 3. The HR clicks the menu and checks if every employee has attended 4. If any employee is absent without cause notifies the department head 5. Click done
Entry condition	The HR must be registered to the database and is given the HR role
Exceptional Flow	

Table 3.3. Use case description of Control Attendance

3.4. Analysis level class diagram (Conceptual Modeling)

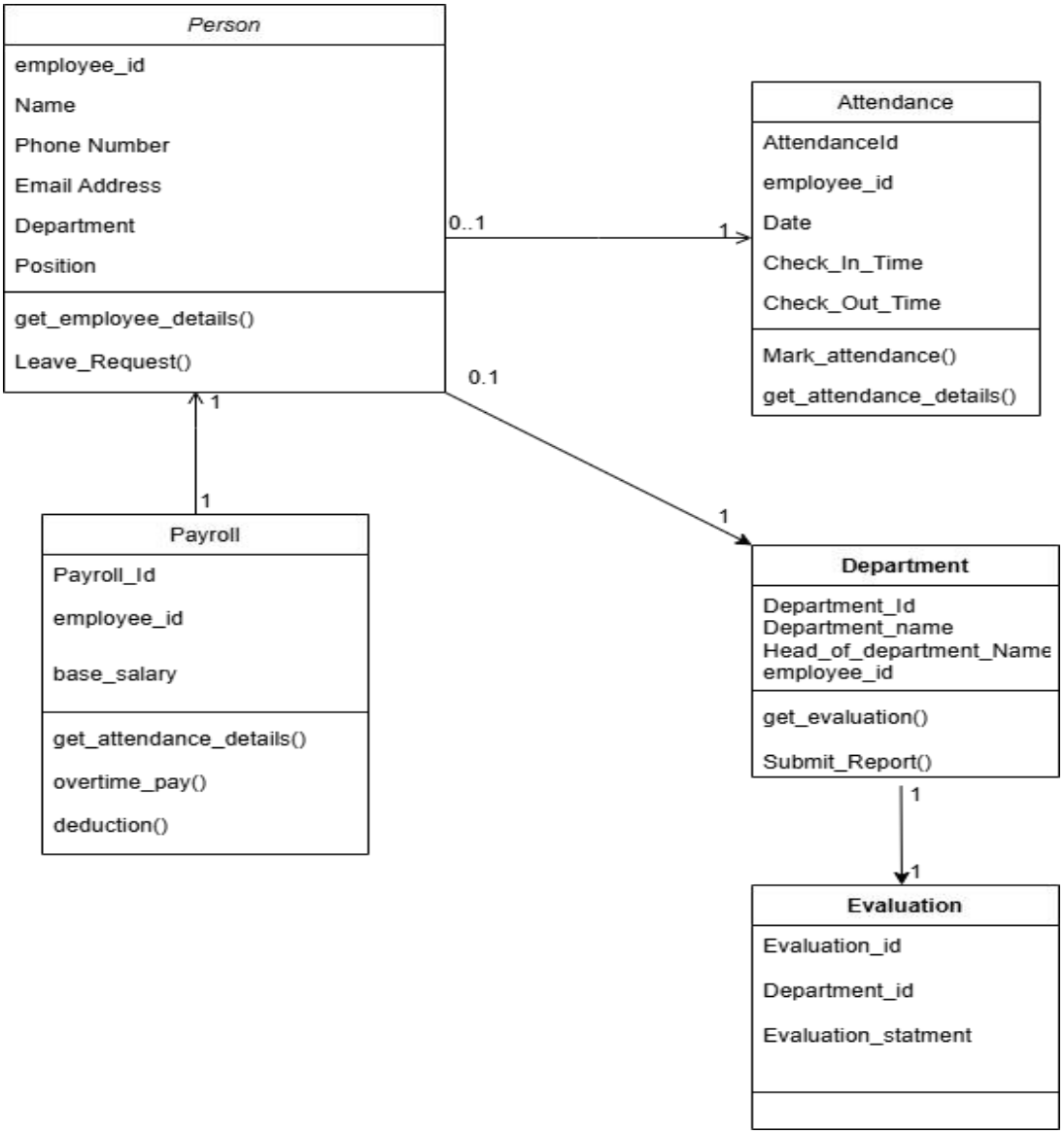


Fig 3.2. Conceptual Modeling

3.5. Sequence diagram

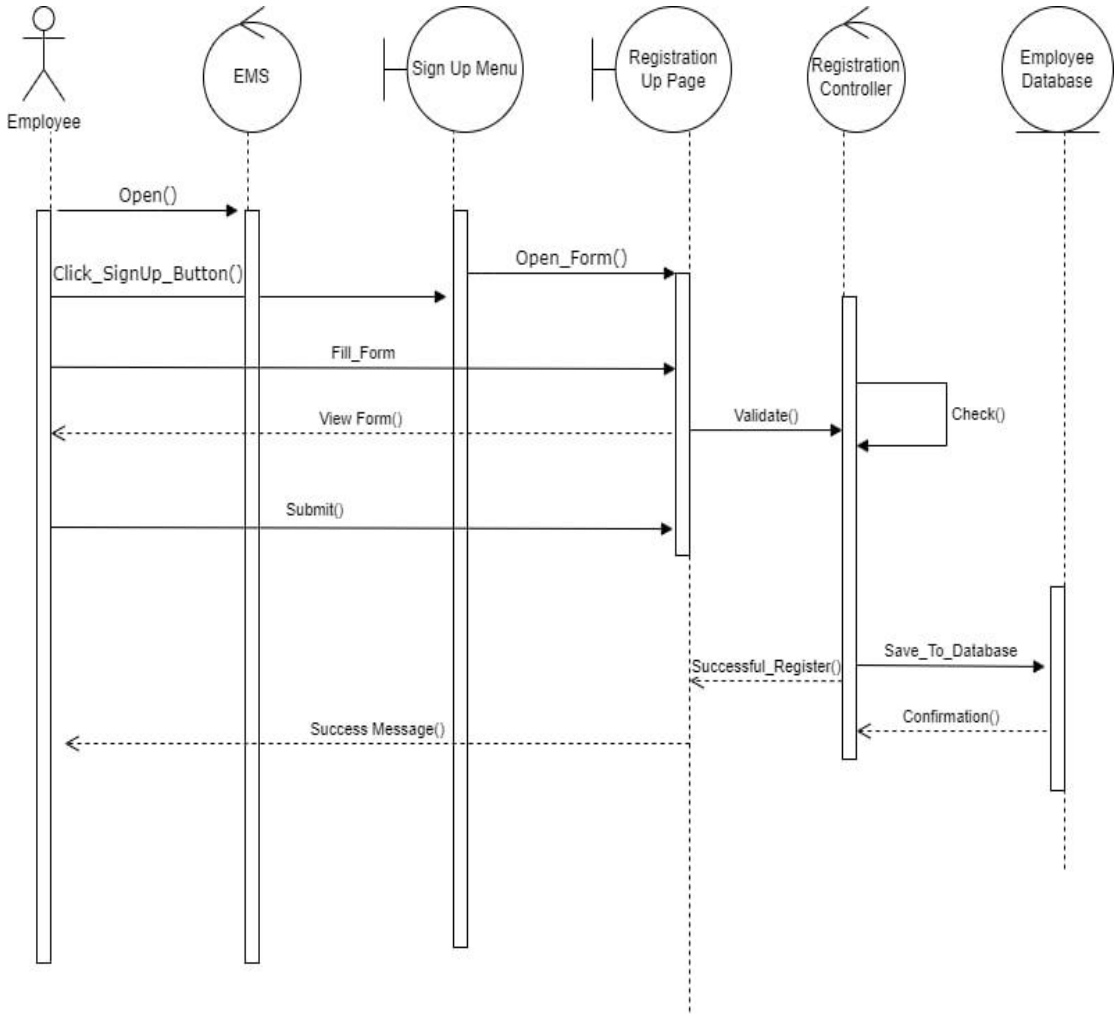


Fig 3.3 Sequence Diagram For Registration

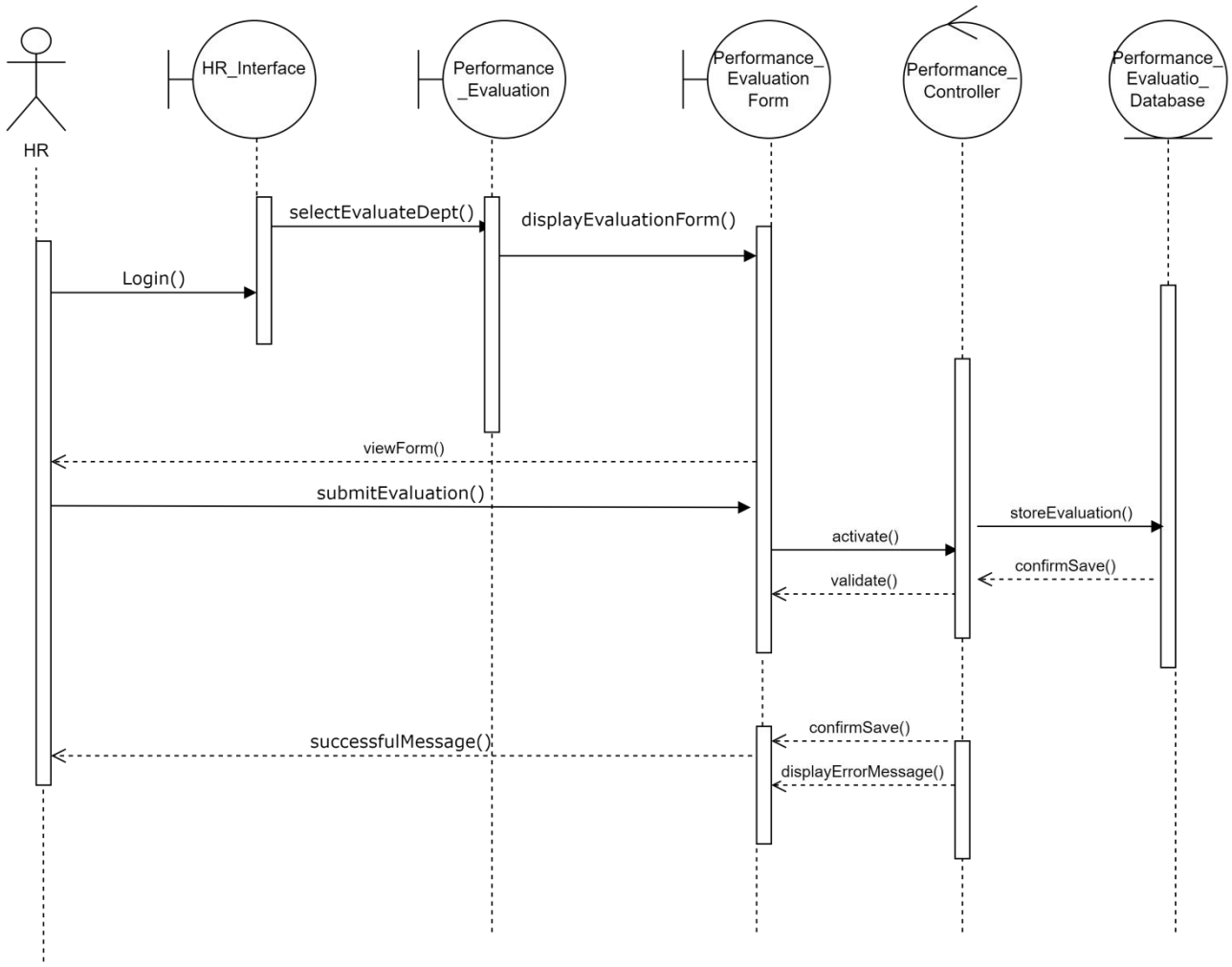


Fig 3.4. Sequence Diagram for Performance Evaluation of Department

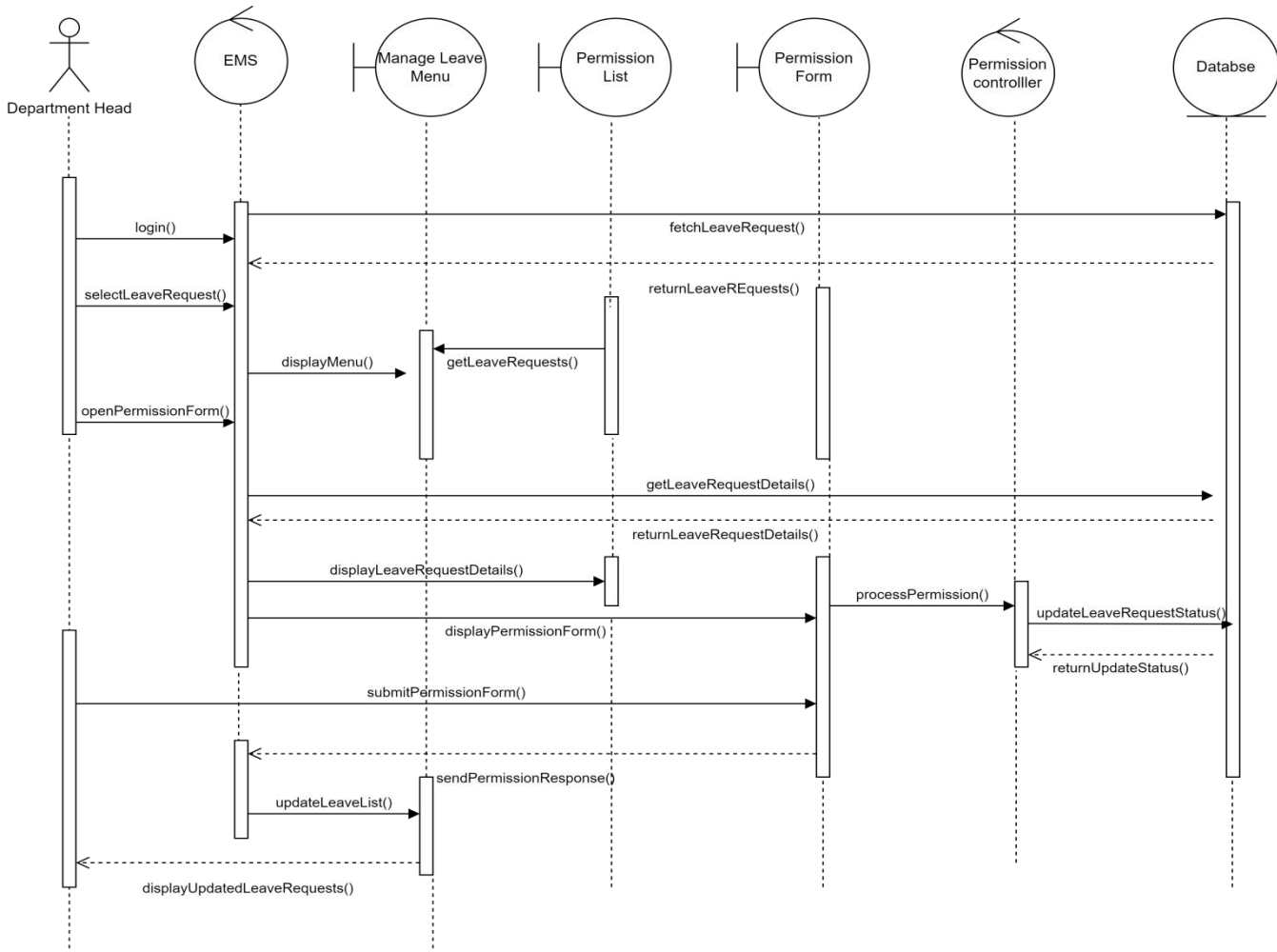


Fig 3.5. Sequence Diagram for Manage Leave Requests

3.6. State Chart Modeling

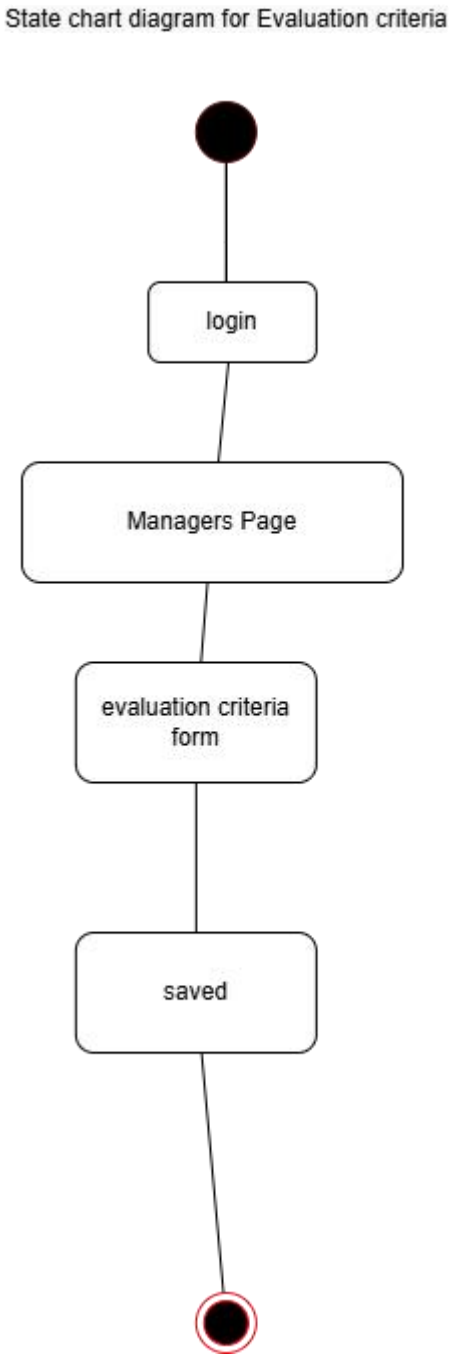


Fig 3.6. State chart Diagram for Evaluation Criteria

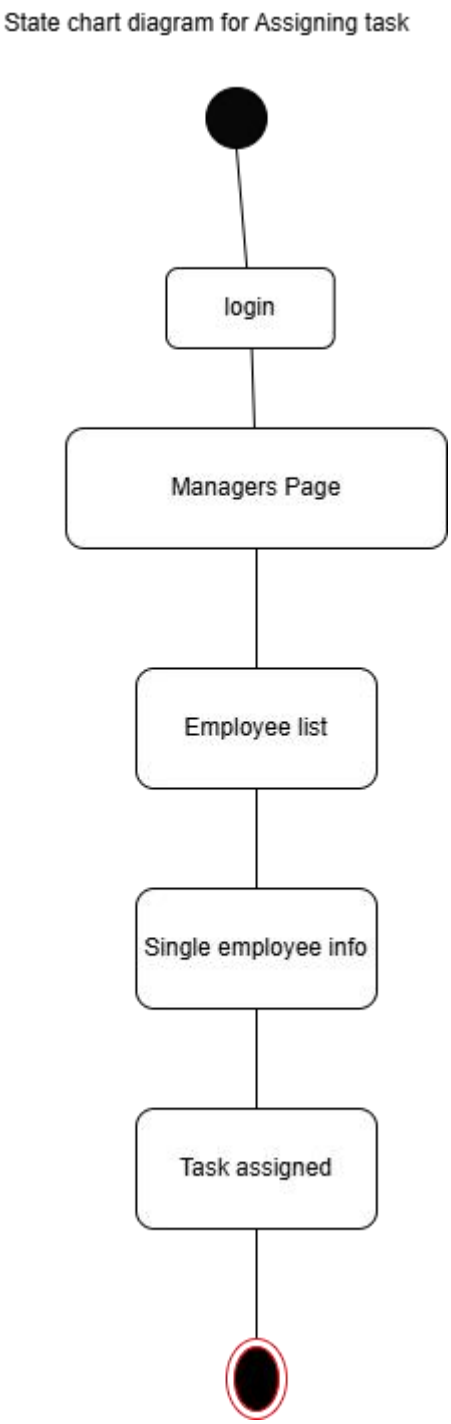


Fig 3.7. State Chart Diagram for Assigning Task

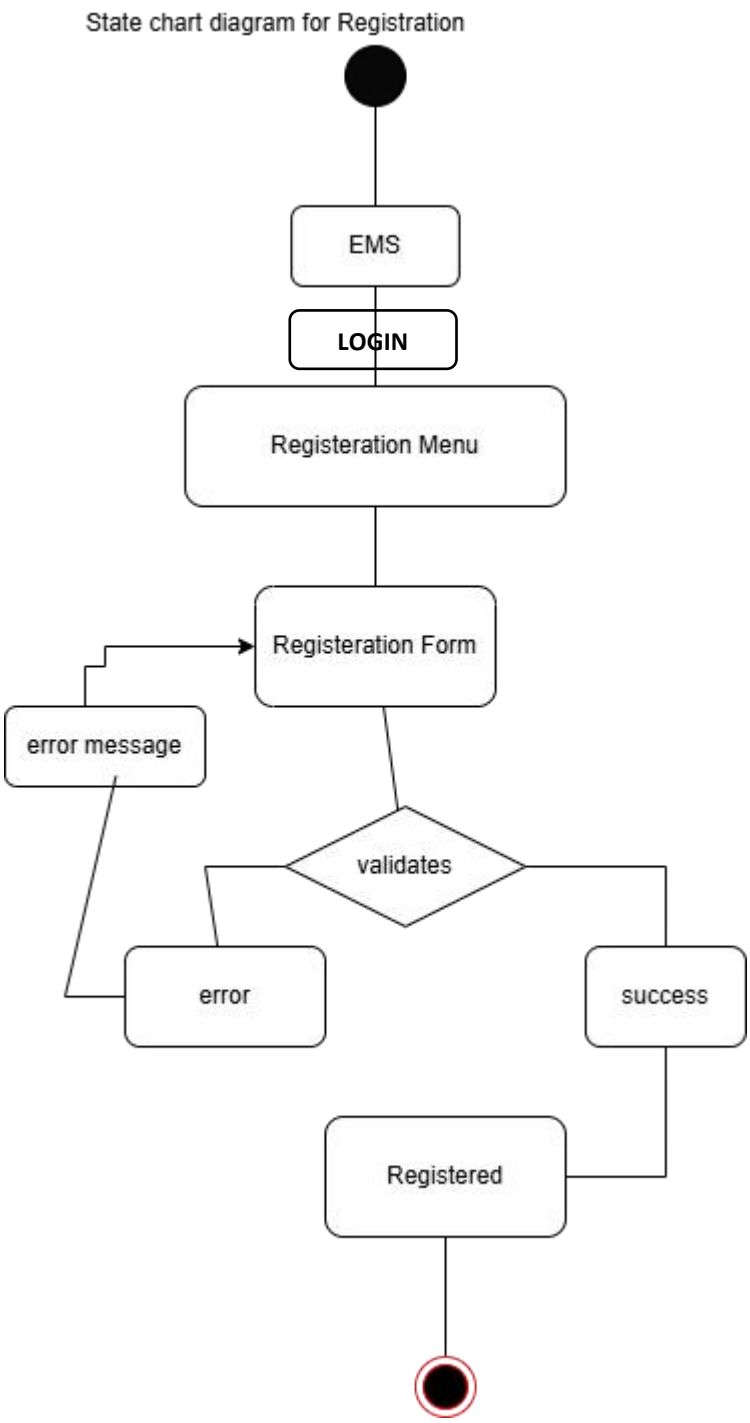


Fig 3.8. State Chart Diagram for Registration

3.7. Activity Diagrams

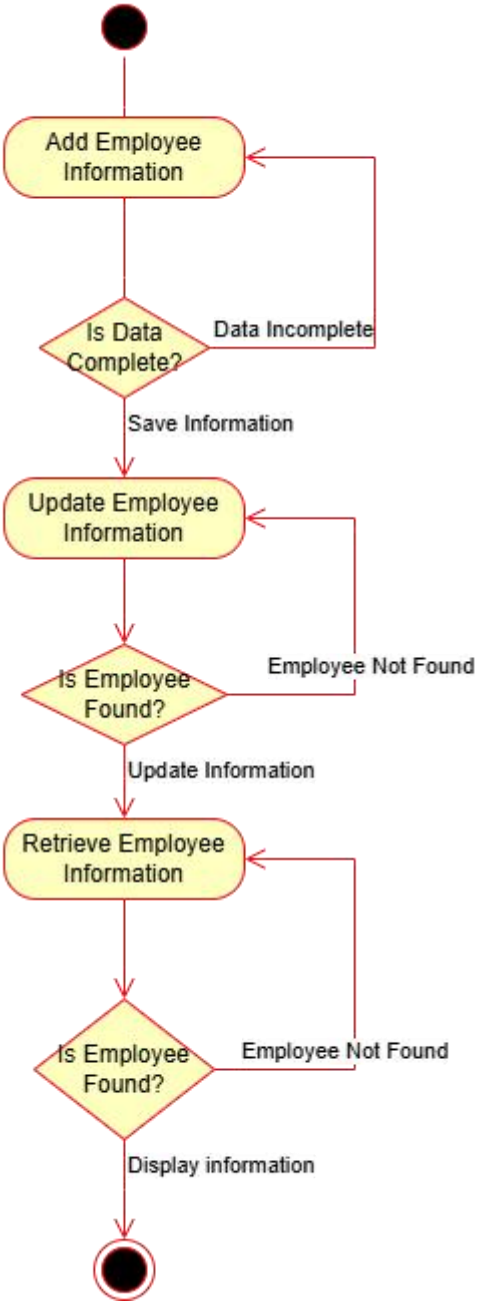


Fig 3.9 Activity Diagram for Employee Information

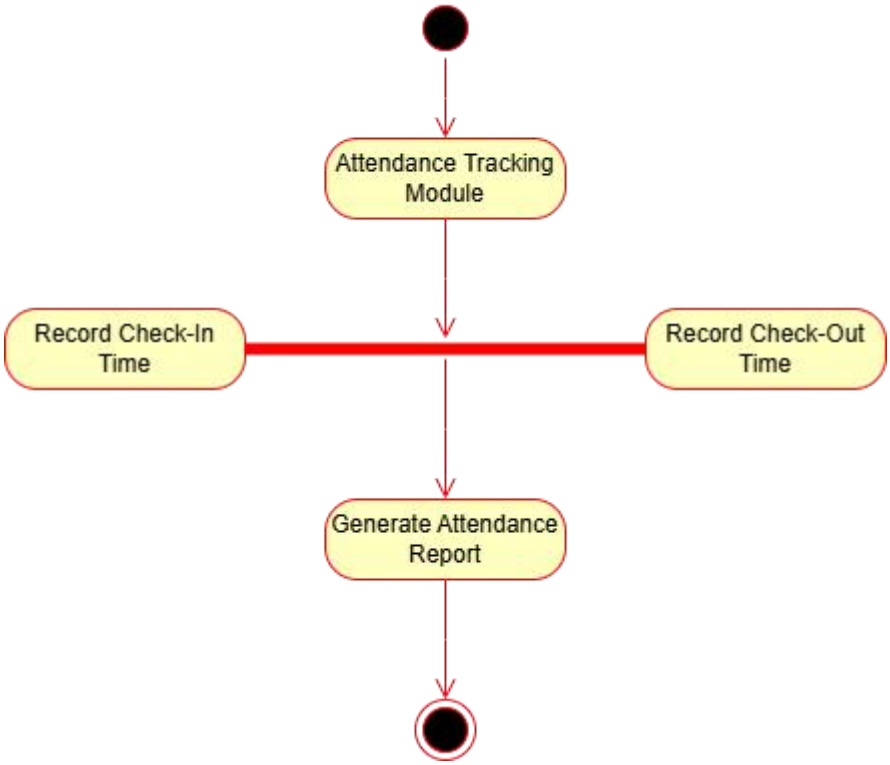


Fig 3.10 Activity Diagram for Attendance Tracking

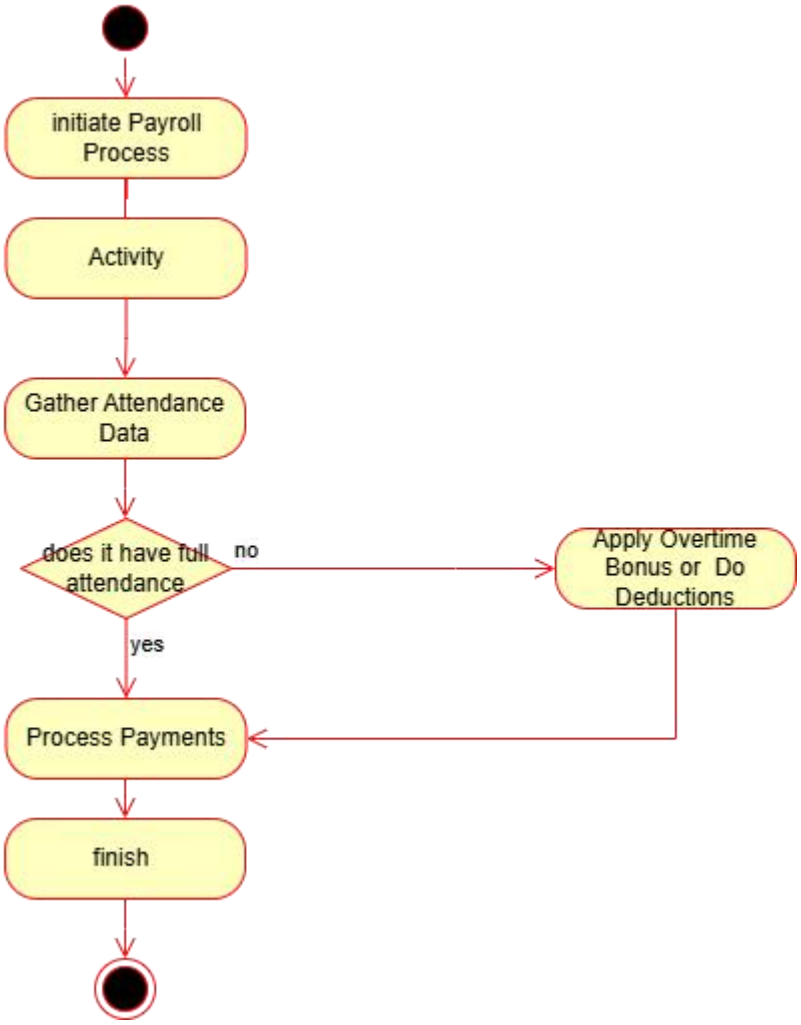


Fig 3.11 Activity Diagram for Payroll Processing

3.8. User Interface Prototyping

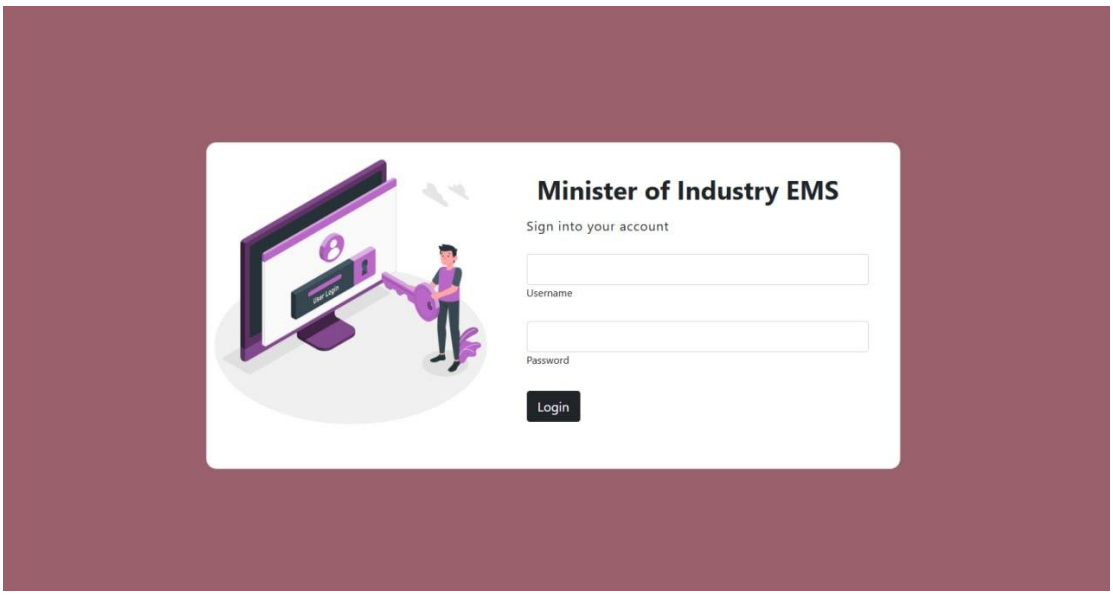


Image 3.0

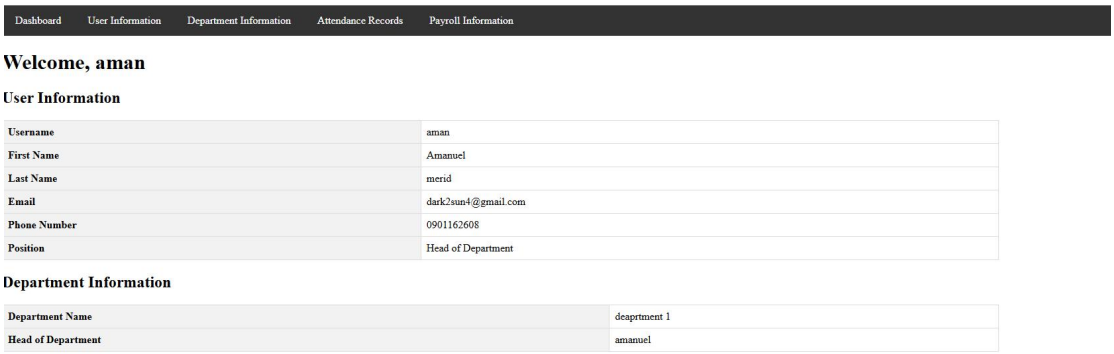


Image 3.1

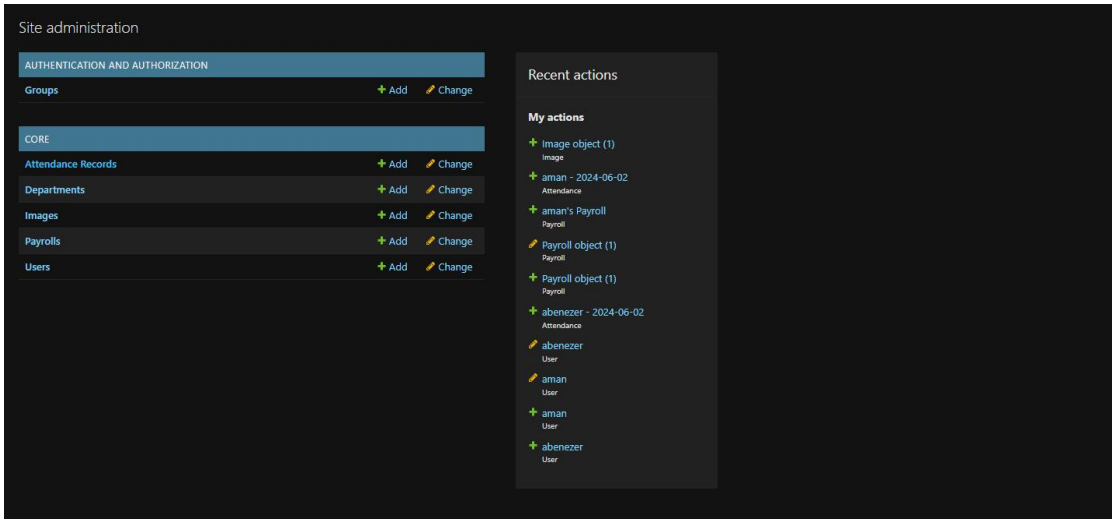


Image 3.2

CHAPTER 4

System Design

4.1. Introduction

The design of the Employee Management System is aimed at creating an efficient, user-friendly system to manage employee information, attendance, performance evaluations, payroll, scheduling, and leave management. The system is designed to ensure accurate tracking and compliance with leave policies, thereby simplifying leave management.

4.2. System Architecture

The architecture of the Employee Management System is designed to be robust, scalable, and secure. It consists of several components, each serving a specific purpose:

1. Employee Information Module

The Employee Information Module is designed to optimize employee information management. It creates a centralized database for securely storing and easily accessing detailed employee information. This includes personal details, job roles, department details, and more. The module ensures data integrity, security, and privacy while providing an efficient retrieval system.

2. Attendance Tracking Module

The Attendance Tracking Module aims to improve attendance tracking. It implements a real-time attendance tracking system that minimizes absenteeism and ensures compliance with company policies. It includes features like check-in and check-out times, breaks, and overtime tracking. The module provides real-time reports that can be used for payroll and performance evaluations.

3. Performance Evaluation Module

The Performance Evaluation Module is designed to enhance performance evaluations. It provides tools for conducting thorough and unbiased performance reviews, thereby aiding in employee development and productivity. The module supports periodic reviews and provides a platform for feedback and improvement plans.

4. Payroll Processing Module

The Payroll Processing Module is developed to automate payroll processing. It ensures accurate and timely payroll calculations, thereby reducing errors and

administrative workload. The module takes into account factors like attendance, overtime, bonuses, deductions, and other factors to calculate salaries.

5. Scheduling Module

The Scheduling Module is designed to optimize scheduling. It offers a flexible scheduling feature that allows managers to create, manage, and adjust work schedules easily. The module takes into account factors like employee availability, department requirements, and workload distribution.

6. Leave Management Module

The Leave Management Module aims to simplify leave management. It ensures accurate tracking of employee leave requests and balances, thereby maintaining compliance with organizational leave policies. The module manages leave requests and approvals, providing visibility into leave balances and history

4.3. Class Modeling

The class diagram for the employee management system has the core components of the system, each represented by a class.

The **Employee** class serves as the central entity with attributes such as **Employee_id, name, Phone_Number, Email_Address, Position, and Department**. It includes methods like **Get_employee_details** to retrieve an employee's information and **Leave_Request** to handle leave applications.

The **Payroll** class is linked to the Employee through the **Employee_id** and manages financial records with attributes like **Payroll_id, base_Salary** and methods such as **get_attendance_details()** to fetch attendance data, **overtime_pay()** to calculate **extra working hours, and deduction()** to apply any salary deductions.

Attendance tracks employee presence with fields like **AttendanceId, Date, Check_In_Time, and Check_Out_Time.** It features methods like **Mark_attendance()** to record attendance and **get_attendance_details()** to retrieve it.

The **Department** class outlines the organizational structure with attributes including **Department_Id, Department_name, and Head_of_department_Name**. It is associated with employees through the **employee_id** and includes a method

Submit_Report() to submit their work for the evaluation and **get_evaluation()** to assess departmental performance.

Evaluation class, connected to the Department via **Department_id,** holds performance reviews with an attribute **Evaluation_statement** that captures qualitative assessments of departmental achievements.

4.4. Component Diagram

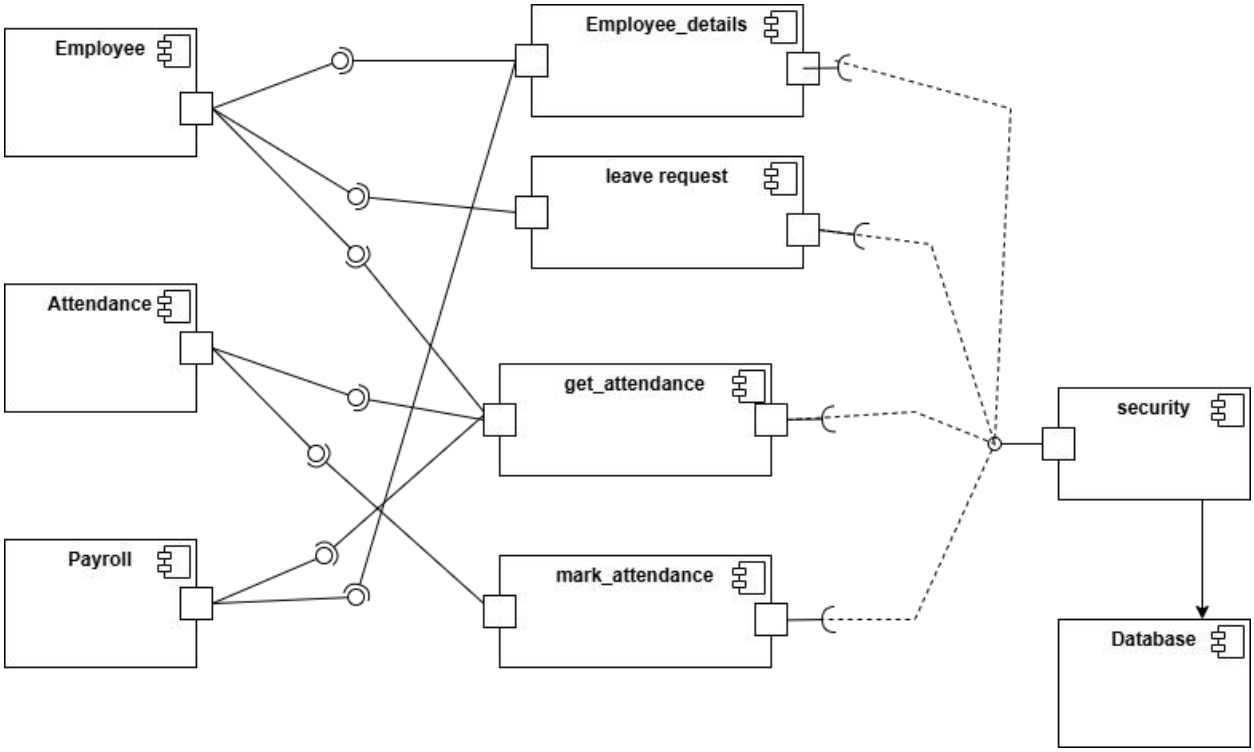


Fig 3.12. Component Diagram

4.5 Subsystem Decomposition

- **Employee Management Subsystem**

  Responsibility: Manage employee information and profiles.

  Modules:

  - Employee Registration
  - Update Details
  - Leave Request

- **Payroll Management Subsystem**

Responsibility: Manage financial records related to employee salaries.

Modules:

- Base Salary Calculation
- Overtime Pay Calculation
- Deduction Application

- **Attendance Management Subsystem**

Responsibility: Track employee attendance.

Modules:

- Mark Attendance
- Retrieve Attendance Details

- **Department Management Subsystem**

Responsibility: Manage departmental information and structure.

Modules:

- Manage Department Details
- Submit Reports
- Assess Performance

- **Evaluation Management Subsystem**

Responsibility: Handle performance evaluations.

Modules:

- Capture Evaluation Statements
- Retrieve Evaluation Statements

- **Authentication and Authorization Subsystem**

Responsibility: Manage user authentication and access control.

Modules:

- User Login
- Role-based Access Control

- Session Management

Explanation of Subsystems
Employee Management Subsystem:
Handles employee records, including adding new employees, updating information, and storing contact details.

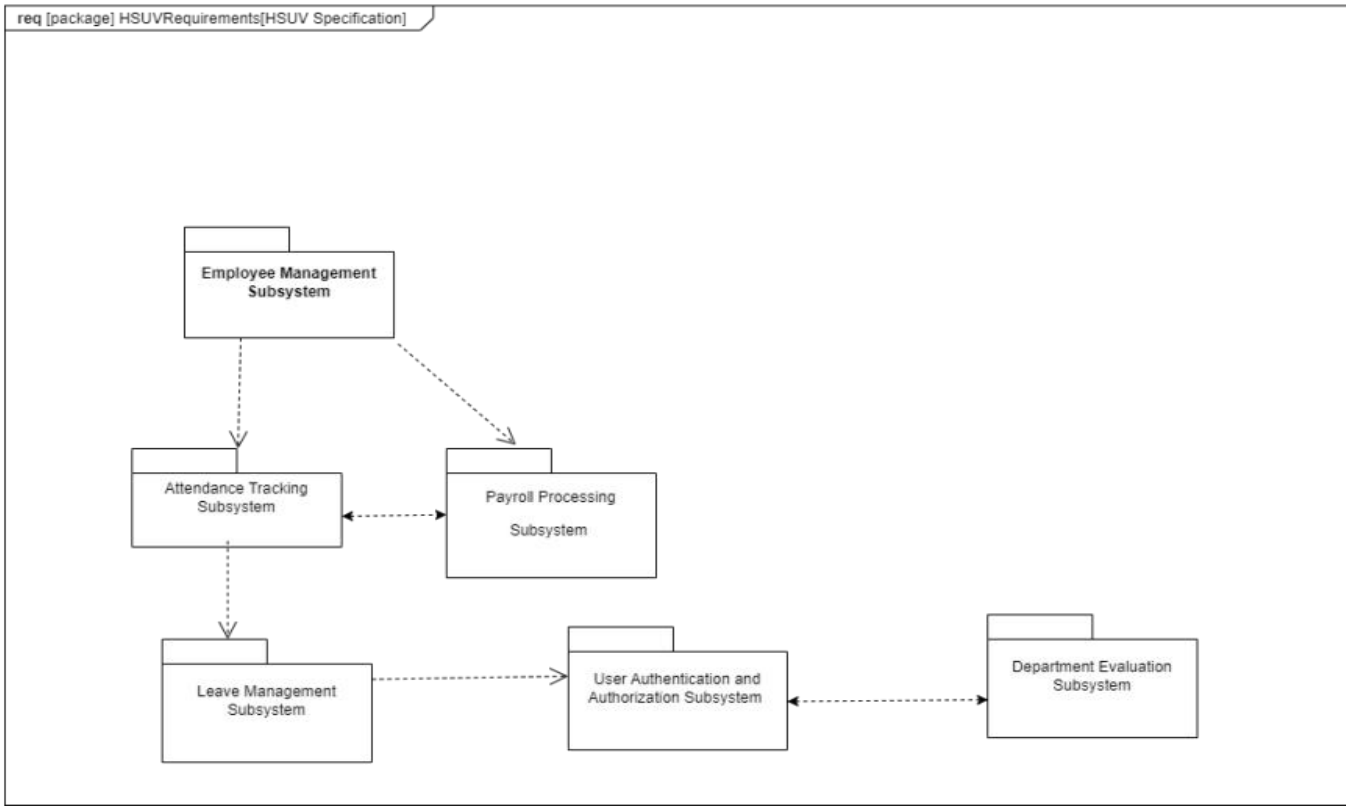
Attendance Tracking Subsystem:
Records daily attendance, check-in and check-out times, and generates attendance reports.

Leave Management Subsystem:
Manages leave requests, approvals, and tracking leave balances.

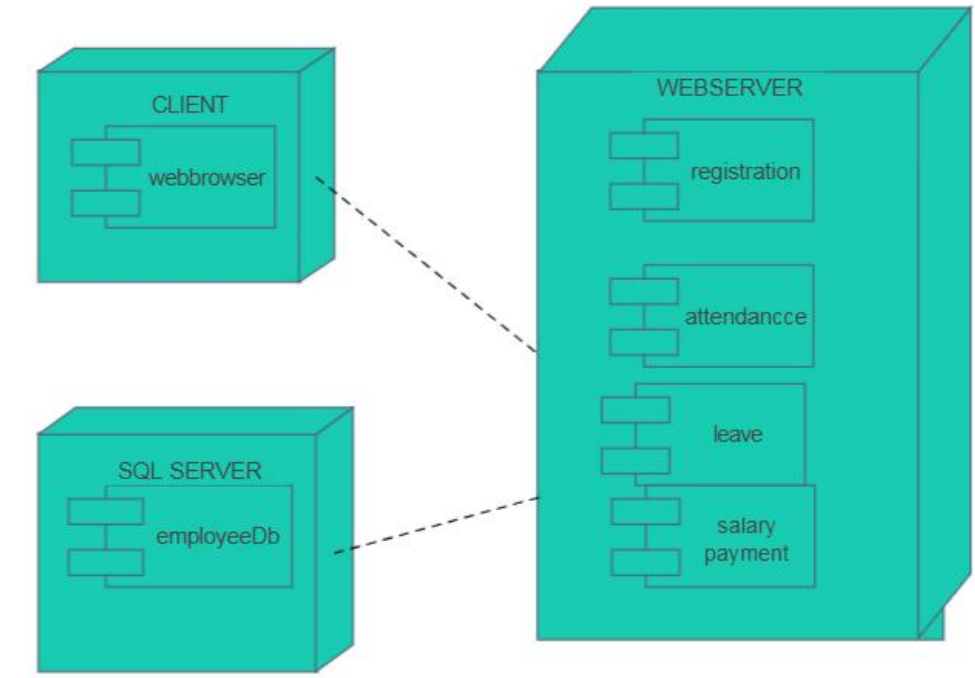
Payroll Processing Subsystem:
Processes payroll, calculates salaries based on attendance and leave data, and handles deductions and overtime payments.

User Authentication and Authorization Subsystem:
Manages user logins, roles, and permissions to ensure secure access to the system.

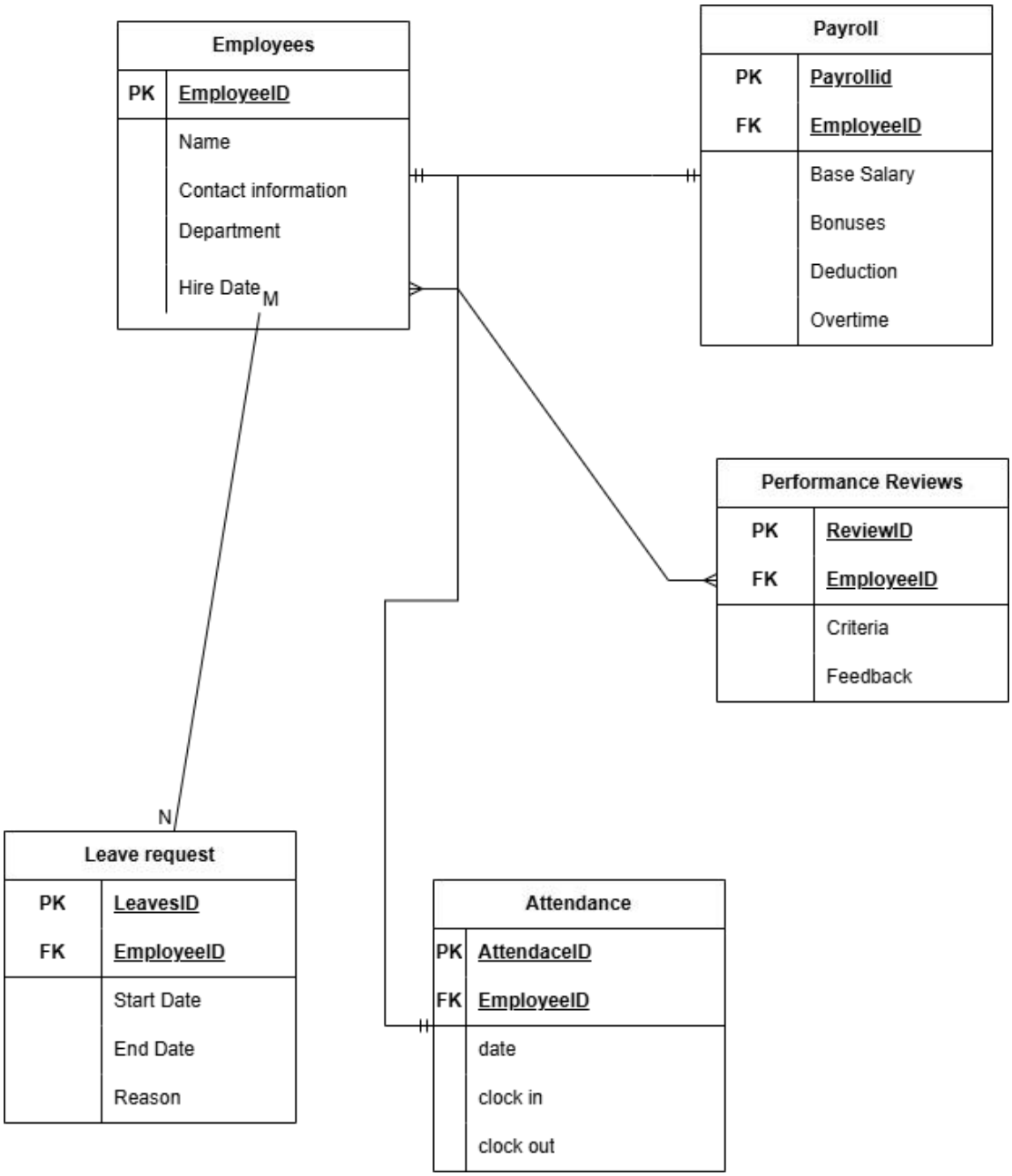
Department Evaluation Subsystem:
Evaluates departmental performance based on predefined metrics and generates evaluation reports.



4.6 Deployment Diagram



4.7 Database Design



CHAPTER 5

Implementation

5.1. Introduction

The implementation chapter outlines the practical steps taken to bring the Ministry of Industry Management System from concept to reality. This phase involves translating the system requirements and design specifications into a functional application

through coding, configuration, and integration of various components. The chapter begins by detailing the development environment and tools utilized, followed by a discussion of the system architecture and the key modules implemented.

By the end of this chapter, readers will gain a comprehensive understanding of the technical implementation of the Ministry of Industry Management System, including the challenges encountered, solutions devised, and the overall workflow that led to a functional and efficient system tailored to the needs of the Ministry's employees and managers.

5.2. Software and hardware acquisitions

Software Acquisitions

Operating System:

1. **Windows Server 2019/2022** or **Linux (Ubuntu Server 20.04 LTS or later)**: To host the application server, database server, and other necessary services.
2. **Windows 10/11** or **Ubuntu Desktop 20.04 LTS or later**: For client machines used by HR staff and department heads.

Database Management System (DBMS):

1. **PostgreSQL**: A robust, open-source relational database management system that can handle the system's data requirements.
2. **MySQL**: As an alternative relational database management system.

Web Server:

1. **Apache HTTP Server** or **Nginx**: For serving the web application.

Programming Languages and Frameworks:

1. **Python**: For server-side development.
2. **Django**: A high-level Python web framework for rapid development and clean, pragmatic design.

3. **HTML5**, **CSS3**: For client-side development.
4. **Bootstrap**: For responsive design and styling.

Version Control System:

1. **Git**: For source code management and version control.
2. **GitHub** or **GitLab**: For hosting repositories and collaboration.

Integrated Development Environment (IDE):

1. **Visual Studio Code**: For development and debugging.

Project Management Tools:

1. **Jira**: For task tracking and project management.

Testing Tools:

1. **Selenium**: For automated testing of the web application.
2. **PyTest**: For unit testing in Python.

Hardware Acquisitions

Servers:

Application Server, Database Server and Backup Server:

Network Hardware:

1. **Gigabit Ethernet Switches**: To ensure fast and reliable network connectivity.
2. **Firewalls**: For network security.

Backup and Storage Solutions:

1. **NAS (Network Attached Storage)**: For regular backups and data redundancy.
2. **External Hard Drives**: For additional backup options.

By acquiring the necessary software and hardware, the Ministry of Industry can ensure the successful implementation and operation of the management system, providing a robust and efficient solution for managing employee information, attendance, leave requests, performance evaluations, and payroll processing.

5.3 . Sample Code

```
def login_view(request):
    if request.method == 'POST':
        username = request.POST['username']
        password = request.POST['password']
        user = authenticate(request, username=username,
password=password)
        if user is not None:
            login(request, user)
            if user.role == 'admin':
                return redirect('admin_dashboard')
            elif user.role == 'editor':
                return redirect('editor_dashboard')
            elif user.role == 'viewer':
                return redirect('viewer_dashboard')
    return render(request, 'core/login.html',)

<head>
    {% load static %}
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.m
in.css" rel="stylesheet" integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC"
crossorigin="anonymous">
  </head>
  <section class="vh-100" style="background-color: #9A616D;">
    <div class="container py-5 h-100">
      <div class="row d-flex justify-content-center align-items-center
h-100">
        <div class="col col-xl-10">
          <div class="card" style="border-radius: 1rem;">
            <div class="row g-0">
              <div class="col-md-6 col-lg-5 d-none d-md-block">
                <img src="../media/Login-amico_1.png"
                  alt="login form" class="img-fluid" style="border-
radius: 1rem 0 0 1rem;" />
              </div>
              <div class="col-md-6 col-lg-7 d-flex align-items-center">
                <div class="card-body p-4 p-lg-5 text-black">
```

```
                <form method ="post" action="{% url 'login' %}">
                  {% csrf_token %}
                  <div class="d-flex align-items-center mb-3 pb-1">
                    <i class="fas fa-cubes fa-2x me-3" style="color:
#ff6219;"></i>
                    <span class="h1 fw-bold mb-0">Minister of Industry
EMS</span>
                  </div>

                  <h5 class="fw-normal mb-3 pb-3" style="letter-
spacing: 1px;">Sign into your account</h5>

                  <div data-mdb-input-init class="form-outline mb-4">
                    <input type="text" id="username" name="username"
class="form-control form-control-lg" required/>
                    <label class="form-label"
for="form2Example17">Username</label>
                  </div>

                  <div data-mdb-input-init class="form-outline mb-4">
                    <input type="password" id="password"
name="password" class="form-control form-control-lg" />
                    <label class="form-label"
for="form2Example27">Password</label>
                  </div>

                  <div class="pt-1 mb-4">
                    <button data-mdb-button-init data-mdb-ripple-init
class="btn btn-dark btn-lg btn-block" type="submit">Login</button>
                  </div>


                </form>

              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </section>
```

Login sample code

```
def leave_request(request):
    if request.method == 'POST':
        reason = request.POST.get('reason')
        requester = request.user
        leave_request = LeaveRequest.objects.create(
            requester=requester,
            reason=reason
        )
        # Redirect to leave request page
        return redirect('leave_request')

    return render(request, 'core/leave_request.html')
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Leave Request</title>
</head>
<body>
    <nav>
        <ul>
            <li><a href="{% url 'viewer_dashboard' %}">Viewer
Dashboard</a></li>
            <li><a href="{% url 'leave_request' %}">Leave
Request</a></li>
        </ul>
    </nav>

    <h1>Leave Request Form</h1>
    <form method="post">
        {% csrf_token %}
        <label for="reason">Reason for Leave:</label><br>
        <textarea id="reason" name="reason" rows="4"
cols="50"></textarea><br>
        <input type="submit" value="Submit">
    </form>
</body>
</html>
```

Leave request sample code

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Editor Dashboard</title>
    <style>

        table {
            width: 80%;
            margin-bottom: 20px;
            border-collapse: collapse;
        }
        th, td {
            border: 1px solid #ddd;
            padding: 8px;
            text-align: left;
        }
        th {
            background-color: #f2f2f2;
        }
        .navbar {
            overflow: hidden;
            background-color: #333;
        }
        .navbar a {
            float: left;
            display: block;
            color: white;
            text-align: center;
            padding: 14px 20px;
            text-decoration: none;
        }
        .navbar a:hover {
            background-color: #ddd;
            color: black;
        }


    </style>
</head>
<body>
```

```
<!-- Navbar -->
<div class="navbar">
    <a href="{% url 'editor_dashboard' %}">Dashboard</a>
    <a href="#user-info">User Information</a>
    <a href="#department-info">Department Information</a>
    <a href="#attendance">Attendance Records</a>
    <a href="#payroll">Payroll Information</a>
```

```
</div>
```

```
<h1>Welcome, {{ editor.username }}</h1>
```

```
<!-- User Information Table -->
<h2 id="user-info">User Information</h2>
<div class="pus">
<table>
    <tr>
        <th>Username</th>
        <td>{{ editor.username }}</td>
    </tr>
    <tr>
        <th>First Name</th>
        <td>{{ editor.first_name }}</td>
    </tr>
    <tr>
        <th>Last Name</th>
        <td>{{ editor.last_name }}</td>
    </tr>
    <tr>
        <th>Email</th>
        <td>{{ editor.email }}</td>
    </tr>
    <tr>
        <th>Phone Number</th>
        <td>{{ editor.phone_number }}</td>
    </tr>
    <tr>
        <th>Position</th>
        <td>{{ editor.position }}</td>
    </tr>
    <!-- Add other fields as needed -->
</table>
</div>
<!-- Department Information Table -->
<h2 id="department-info">Department Information</h2>
<table>
    <tr>
        <th>Department Name</th>
        <td>{{ department.department_name }}</td>
    </tr>
    <tr>
        <th>Head of Department</th>
        <td>{{ department.head_of_department }}</td>
    </tr>
    <!-- Add other department-related fields as needed -->
</table>
```

```
<!-- Attendance Records Table -->
<h2 id="attendance">Attendance Records for
{{ department.department_name }}</h2>
<table>
    <thead>
        <tr>
            <th>Employee Name</th>
            <th>Check-in Time</th>
            <th>Check-out Time</th>
            <th>Days Attended</th>
        </tr>
    </thead>
    <tbody>
        {% for record in attendance_records %}
        <tr>
            <td>{{ record.employee.username }}</td>
            <td>{{ record.checkin_time }}</td>
            <td>{{ record.checkout_time }}</td>
            <td>{{ record.days_attended }}</td>
        </tr>
        {% empty %}
        <tr>
            <td colspan="4">No attendance records found.</td>
        </tr>
        {% endfor %}
    </tbody>
</table>
```

```
<!-- Payroll Information Table -->
<h2 id="payroll">Payroll Information</h2>
<table>
    <thead>
        <tr>
            <th>Employee</th>
            <th>Basic Salary</th>
            <th>Days Missed</th>
            <th>Deducted Salary</th>
        </tr>
    </thead>
    <tbody>
        {% for payroll in payrolls %}
        <tr>
            <td>{{ payroll.employee.username }}</td>
            <td>{{ payroll.basic_salary }}</td>
            <td>{{ payroll.days_missed }}</td>
            <td>{{ payroll.deducted_salary }}</td>
        </tr>
        {% empty %}
        <tr>
```

```
            <td colspan="4">No payrolls found.</td>
        </tr>
        {% endfor %}
    </tbody>
</table>
```

```
<!-- Pending Leave Requests Table -->
<h2>Pending Leave Requests</h2>
<table>
    <thead>
        <tr>
            <th>Requester</th>
            <th>Reason</th>
            <th>Actions</th>
        </tr>
    </thead>
    <tbody>
        {% for request in pending_requests %}
        <tr>
            <td>{{ request.requester.username }}</td>
            <td>{{ request.reason }}</td>
            <td>
                <a href="{% url 'accept_leave_request'
request.id %}">Accept</a> /
                <a href="{% url 'reject_leave_request'
request.id %}">Reject</a>
            </td>
        </tr>
        {% empty %}
        <tr>
            <td colspan="3">No pending leave requests.</td>
        </tr>
        {% endfor %}
    </tbody>
</table>
<script>
    $(document).ready(function() {
    // Example: AJAX for accepting leave request
    $('.accept-button').click(function(event) {
        event.preventDefault();
        var url = $(this).attr('href');

        // AJAX request to accept leave request
        $.ajax({
            type: 'GET', // Assuming GET method for simplicity
            url: url,
            success: function(response) {
                alert(response.message); // Example: Show notification
message
```

```
                // Optionally update UI here
            },
            error: function(xhr, status, error) {
                console.error(xhr.responseText);
            }
        });
    });
```

```
    // Example: AJAX for rejecting leave request
    $('.reject-button').click(function(event) {
        event.preventDefault();
        var url = $(this).attr('href');

        // AJAX request to reject leave request
        $.ajax({
            type: 'GET', // Assuming GET method for simplicity
            url: url,
            success: function(response) {
                alert(response.message); // Example: Show notification
message
                // Optionally update UI here
            },
            error: function(xhr, status, error) {
                console.error(xhr.responseText);
            }
        });
    });
});
```

Dashboard sample code

5.4. Test Procedures

Employee Profile Management

Objectives

- Verify the ability to create, update, delete, and view employee profiles.

Unit Testing

Test Case: Create Employee Profile

- **Objective:** Ensure a new employee profile can be created.

- **Steps:**

    1. Log in as HR user.
    2. Navigate to "Employee Management".
    3. Click "Create Employee".
    4. Fill in the required details.
    5. Click "Save".

- **Expected Result:** Employee profile is created and saved in the database.

Test Case: Update Employee Profile

- **Objective:** Ensure an employee profile can be updated.
- **Steps:**

    1. Select an existing employee profile.
    2. Click "Edit".
    3. Modify the necessary fields.
    4. Click "Save".

- **Expected Result:** Employee profile is updated in the database.

Test Case: Delete Employee Profile

- **Objective:** Ensure an employee profile can be deleted.
- **Steps:**

    1. Select an existing employee profile.
    2. Click "Delete".
    3. Confirm deletion.

- **Expected Result:** Employee profile is removed from the database.

Test Case: View Employee Details

- **Objective:** Ensure employee details can be viewed.

- **Steps:**

    1. Select an existing employee profile.

- **Expected Result:** Employee details are displayed.

Integration Testing

- **Test Case: End-to-End Employee Management**

    - **Objective:** Ensure all CRUD operations for employee profiles work seamlessly.
    - **Steps:**

        1. Create an employee profile.
        2. Update the profile.
        3. View the profile.
        4. Delete the profile.

    - **Expected Result:** All operations complete successfully without errors.

User Acceptance Testing (UAT)

- **Test Case: HR User Creates and Manages Employee Profiles**

    - **Objective:** Ensure HR users can manage employee profiles as per requirements.
    - **Steps:**

        1. Log in as an HR user.
        2. Perform create, update, delete, and view operations on employee profiles.

    - **Expected Result:** HR user can manage employee profiles efficiently.

2. Leave Management

Objectives

- Verify the ability to submit, approve/reject, and view leave requests.

Unit Testing

Test Case: Submit Leave Request

- **Objective:** Ensure leave requests can be submitted.
- **Steps:**

  1. Log in as an employee.
  2. Navigate to "Leave Request".
  3. Select leave type, enter dates and reason.
  4. Click "Submit".

- **Expected Result:** Leave request is submitted and saved in the database.

Test Case: Approve/Reject Leave Request

- **Objective:** Ensure leave requests can be approved or rejected.
- **Steps:**

  1. Log in as a manager.
  2. Navigate to "Leave Requests".
  3. Select a pending request.
  4. Click "Approve" or "Reject".

- **Expected Result:** Leave request status is updated.

Test Case: View Leave History

- **Objective:** Ensure leave history can be viewed.
- **Steps:**

  1. Log in as an employee or manager.
  2. Navigate to "Leave History".

- **Expected Result:** Leave history is displayed.

Integration Testing

- **Test Case: End-to-End Leave Management**
    - **Objective:** Ensure submitting, approving/rejecting, and viewing leave requests work seamlessly.
    - **Steps:**
        1. Submit a leave request.
        2. Approve or reject the request.
        3. View the leave history.
    - **Expected Result:** All operations complete successfully without errors.

User Acceptance Testing (UAT)

- **Test Case: Employee Submits and Manager Approves Leave Requests**
    - **Objective:** Ensure employees and managers can manage leave requests as per requirements.
    - **Steps:**
        1. Log in as an employee.
        2. Submit a leave request.
        3. Log in as a manager.
        4. Approve or reject the request.
        5. Log in as the employee to view the leave history.
    - **Expected Result:** Employees and managers can manage leave requests efficiently.

3. Performance Evaluation

Objectives

- Verify the ability to schedule, submit, and view performance reviews.

Unit Testing

Test Case: Schedule Performance Review

- **Objective:** Ensure performance reviews can be scheduled.
- **Steps:**

    1. Log in as a manager.
    2. Navigate to "Performance Reviews".
    3. Select an employee and review date.
    4. Set evaluation criteria.
    5. Click "Schedule".

- **Expected Result:** Performance review is scheduled and saved in the database.

Test Case: Submit Performance Review

- **Objective:** Ensure performance reviews can be submitted.
- **Steps:**

    1. Log in as a manager.
    2. Navigate to "Performance Reviews".
    3. Select a scheduled review.
    4. Enter feedback and scores.
    5. Click "Submit".

- **Expected Result:** Performance review is submitted and saved in the database.

Test Case: View Performance Reviews

- **Objective:** Ensure performance reviews can be viewed.
- **Steps:**

    1. Log in as an employee or manager.
    2. Navigate to "Performance Reviews".

- **Expected Result:** Performance reviews are displayed.

Integration Testing

- **Test Case: End-to-End Performance Evaluation**
    - **Objective:** Ensure scheduling, submitting, and viewing performance reviews work seamlessly.
    - **Steps:**
        1. Schedule a performance review.
        2. Submit the performance review.
        3. View the performance review.
    - **Expected Result:** All operations complete successfully without errors.

User Acceptance Testing (UAT)

- **Test Case: Manager Schedules and Submits Performance Reviews**
    - **Objective:** Ensure managers can schedule and submit performance reviews as per requirements.
    - **Steps:**
        1. Log in as a manager.
        2. Schedule a performance review.
        3. Submit the performance review.
        4. Log in as the employee to view the review.
    - **Expected Result:** Managers can schedule and submit performance reviews efficiently.

4. Payroll Management

Objectives

- Verify the ability to calculate salary, generate payroll reports, and view payroll history.

Unit Testing

Test Case: Calculate Salary

- **Objective:** Ensure salary can be calculated.
- **Steps:**

    1. Log in as a payroll administrator.
    2. Navigate to "Payroll".
    3. Select an employee and pay period.
    4. Click "Calculate".

- **Expected Result:** Salary is calculated and displayed.

Test Case: Generate Payroll Reports

- **Objective:** Ensure payroll reports can be generated.
- **Steps:**

    1. Log in as a payroll administrator.
    2. Navigate to "Payroll Reports".
    3. Select a date range.
    4. Click "Generate Report".

- **Expected Result:** Payroll report is generated and displayed.

Test Case: View Payroll History

- **Objective:** Ensure payroll history can be viewed.
- **Steps:**

    1. Log in as an employee or payroll administrator.
    2. Navigate to "Payroll History".

- **Expected Result:** Payroll history is displayed.

Integration Testing

- **Test Case: End-to-End Payroll Management**

- **Objective:** Ensure calculating salary, generating reports, and viewing history work seamlessly.
- **Steps:**

    1. Calculate salary for an employee.
    2. Generate a payroll report.
    3. View payroll history.

- **Expected Result:** All operations complete successfully without errors.

User Acceptance Testing (UAT)

- **Test Case: Payroll Administrator Manages Payroll**

    - **Objective:** Ensure payroll administrators can manage payroll as per requirements.
    - **Steps:**

        1. Log in as a payroll administrator.
        2. Calculate salary for employees.
        3. Generate payroll reports.
        4. View payroll history.

    - **Expected Result:** Payroll administrators can manage payroll efficiently.

Test Case: User Login

Test Case ID: TC01
Testing Class: Black Box and White Box Test
Testing Name: Unit and Integration Test
Unit to Test: User Login
Assumptions: User successfully logs in

Test Data:

- **User Id:** (valid user_id, invalid user_id, empty user_id)

- **Password:** (valid password, invalid password, empty password)

Executions/Descriptions	Data	Expected Results	Actual Results
Enter valid user_id and valid password.	user_id = 01 password = 12345	Successful login	Successful Login
Enter invalid user_id or invalid password.	user_id = 00 password = 123456	Login failed, error message	Login Failed
Enter empty user_id and valid password.	user_id = -- password = 3456	Login failed, error message	Login Failed
Enter valid user_id and empty password.	user_id = 077 password = --	Login failed, error message	Login failed required message
Enter all fields empty	user_id = -- password = --	Login failed, error message	Required message,login failed

Test Case: Leave Request

Test Case ID: TC03
Testing Class: Black Box and White Box Test
Testing Name: Unit and Integration Test
Unit to Test: Leave Request
Assumptions: Employee submits leave request

Test Data:

- **Leave Type:** (sick leave, vacation, etc.)
- **Leave Dates:** (valid date range, invalid date range, empty)

Executions/Descriptions	Data	Expected Results	Actual Results
Submit leave request with valid data	Leave Type = sick leave Leave Dates = valid date range	Leave request submitted	Leave request submitted
Submit leave request with invalid date range	Leave Type = vacation Leave Dates = invalid date range	Error message displayed	Error message displayed
Submit leave request with empty leave type	Leave Type = --	Error message displayed	Requried message displayed
	Leave Dates = valid date range		
Submit leave request with empty leave dates	Leave Type = sick leave	Error message displayed	Error message displayed
	Leave Dates = --		

These test procedures cover the core functionalities of the Employee Management System, ensuring that each feature is thoroughly tested at different levels to ensure a robust and reliable application.

5.5. User Manual Preparation

1. Introduction

- **Purpose**: This manual provides step-by-step instructions for using the new employee management system implemented for the Ministry of Industry.

- **Audience**: Employees, HR personnel, and managers.

2. System Requirements

- **Hardware**: Standard PC or laptop with internet access.
- **Software**: Web browser (Chrome, Firefox, Safari), PDF reader.

3. Accessing the System

- **URL**: Provide the URL for the system login page.
- **Login**: Enter your user ID and password to access the system.

4. Employee Profile Management

- **View Profile**: Navigate to 'Profile' to view your details.
- **Edit Profile**: Click 'Edit' to update your information.
- **Save Changes**: After editing, click 'Save' to update your profile.

5. Attendance Management

- **Mark Attendance**: Go to 'Attendance' and click 'Check-In' and 'Check-Out' for recording your work hours.
- **View Attendance**: Navigate to 'Attendance History' to view your recorded attendance.

6. Leave Management

- **Request Leave**: Go to 'Leave' and fill out the leave request form.
- **Track Leave Requests**: View the status of your leave requests under 'Leave History'.

7. Payroll Information

- **View Payroll**: Access 'Payroll' to view your salary details and pay slips.
- **Download Pay Slips**: Click on the respective month to download your pay slip.

8. Performance Evaluation

- **View Evaluation**: Navigate to 'Evaluations' to view your performance reviews.
- **Submit Feedback**: Provide feedback on your evaluation through the feedback form.

9. Support

- **Contact Support**: For any issues or queries, contact the support team via the 'Help' section or email support@ems.com.

10. Logout

- **Secure Logout**: Click 'Logout' to securely exit the system.

This concise user manual ensures that users can quickly understand and navigate through the system's functionalists effectively.

5.6. Training, Installation, and Start-up Strategy

5.6.1. Training Strategy

Training Sessions:

1. **Initial Training**: Conduct an initial training session for all employees to familiarize them with the new system.
2. **Follow-up Sessions**: Schedule follow-up sessions to address any queries and provide additional guidance.

Training Materials:

1. **User Manual**: Distribute a concise user manual detailing system features and usage instructions.
2. **Video Tutorials**: Provide video tutorials covering key functionalities and common tasks.

Support:

1. **Help Desk**: Establish a help desk for users to contact for immediate assistance.

2. **FAQs**: Create a FAQ section on the system's homepage to address common questions.

5.6.2. Installation Strategy

System Deployment:

1. **Server Setup**: Install and configure the system on a secure server.
2. **Database Configuration**: Set up the database and ensure all tables and relationships are correctly implemented.

Software Installation:

1. **Web Application**: Deploy the web application on the organization's intranet or a dedicated web server.
2. **Dependencies**: Install necessary software dependencies and libraries.

Testing:

1. **Pte-Deployment Testing**: Conduct thorough testing in a staging environment to identify and fix any issues.
2. **Post-Deployment Testing**: Perform testing after deployment to ensure the system functions as expected in the live environment.

5.6.3. Start-up Strategy

Go-Live Preparation:

1. **Data Migration**: Transfer existing data to the new system.
2. **User Accounts**: Create user accounts and distribute login credentials.

Launch Plan:

1. **Soft Launch**: Begin with a soft launch to a small group of users to identify any immediate issues.
2. **Full Launch**: Gradually roll out the system to all users after resolving any issues identified during the soft launch.

Monitoring and Support:

1. **Monitoring**: Continuously monitor system performance and user feedback.
2. **Support Team**: Ensure the support team is available to address any issues or questions during the initial launch period.

This strategy ensures a smooth transition to the new system, providing comprehensive training, careful installation, and a structured start-up process.

CHAPTER 6
Conclusion And Recommendation

6.1. Conclusion

The development and implementation of the Employee Management System (EMS) project have been a significant step towards modernizing the certain processes within the organization. This system effectively addresses key functionalities such as employee management, payroll processing, attendance tracking, departmental evaluations, and performance reviews. Through the integration of these features, the EMS streamlines operations, reduces manual workloads, and enhances data accuracy, ultimately leading to improved overall efficiency.

The system's design, which includes both conceptual and detailed class modeling, ensures that the underlying architecture is robust and scalable, capable of accommodating future growth and additional features. Comprehensive testing at various stages—unit, integration, and user acceptance has ensured that the system is reliable and meets the intended requirements.

Furthermore, the training, installation, and start-up strategies were carefully planned and executed, ensuring a smooth transition for all users to the new system. The provision of user manuals, training sessions, and ongoing support has facilitated user adoption and minimized disruptions during the rollout phase.

6.2. Recommendation

Continuous Improvement:

1. Regularly update the EMS to incorporate user feedback and evolving organizational needs.
2. Schedule periodic reviews to identify and implement necessary enhancements or optimizations.

Scalability and Integration:

1. Plan for future scalability to accommodate an increasing number of users and additional modules.
2. Explore opportunities for integrating the EMS with other enterprise systems (e.g., ERP, CRM) for seamless data exchange and broader operational efficiency.

Security Enhancements:

1. Continuously monitor and upgrade the security protocols to protect sensitive HR data against potential threats.
2. Implement multi-factor authentication and regular security audits.

User Training and Support:

1. Offer ongoing training programs to ensure users stay updated with system functionalities and best practices.
2. Maintain a robust support system to address any user issues promptly and effectively.

Performance Monitoring:

1. Establish performance metrics and regular monitoring to ensure the system operates efficiently.
2. Use analytical tools to generate insights from employee data, aiding in strategic decision-making.

Feedback Mechanism:

1. Implement a structured feedback mechanism allowing users to report issues and suggest improvements.
2. Actively involve key stakeholders in discussions about future system enhancements.

By following these recommendations, the organization can ensure that the EMS remains a valuable tool that continues to meet the evolving needs of the HR department and the broader organization. The system's successful implementation marks a significant milestone in the organization's digital transformation journey, paving the way for further technological advancements and operational excellence.

References

Employee management system for ASTU | Yesewuyew Lij - Academia.edu

Use Case Diagrams | Unified Modeling Language (UML) - GeeksforGeeks

Appendix