**Final Project Report:**

**“Employee Management System”**

**Created by:**

Abdul Rehman Nazeer

22K-4078

**1. Introduction**

**1.1 Purpose**

This final project report describes the development and implementation of the Employee Management System for a software house. The system is designed to automate HR processes, including managing employee records, payroll, and performance reviews. This report discusses the project's objectives, development process, outcomes, testing, and recommendations for future improvements.

**1.2 Document Conventions**

This report follows standard conventions for project documentation, including consistent headings, font sizes, and terminology. Specific conventions are:

* Font: Times New Roman
* Headings: 12pt, bold
* Subheadings: 12pt, bold
* Body text: 12pt, regular

**1.3 Intended Audience and Reading Suggestions**

This report is intended for project stakeholders, HR professionals, software developers, project managers, and quality assurance personnel. It provides an overview of the Employee Management System's development, features, and outcomes. Readers should be familiar with basic HR processes and software development principles.

**1.4 Product Scope**

The Employee Management System is designed to manage HR tasks, such as maintaining employee records, payroll processing, attendance tracking, and performance management. It is intended to be scalable, secure, and user-friendly, providing a comprehensive solution for HR management in a software house environment.

**1.5 References**

* ChatGPT was used for jQuery and JavaScript references.

**2. Overall Description**

**2.1 Product Perspective**

The Employee Management System is a standalone web application developed using Django. It allows HR personnel to manage employee information, payroll, attendance, and performance. The system integrates with existing HR processes and is intended to improve efficiency and reduce manual tasks.

**2.2 Product Functions**

The system provides the following core functions:

* **User Management**: Manage different user roles (e.g., HR admin, employee).
* **Employee Management**: Add, update, and remove employee records.
* **Payroll Management**: Process payroll and generate reports.
* **Performance Management**: Monitor employee performance and conduct reviews.

**2.3 User Classes and Characteristics**

The Employee Management System has three main user classes:

* **HR Admin**: Responsible for managing employee records and overseeing payroll.
* **Employee**: Can access their personal information, request leave, and view payroll.
* **Manager**: Can manage teams, view employee information, and approve leave requests.

**2.4 Operating Environment**

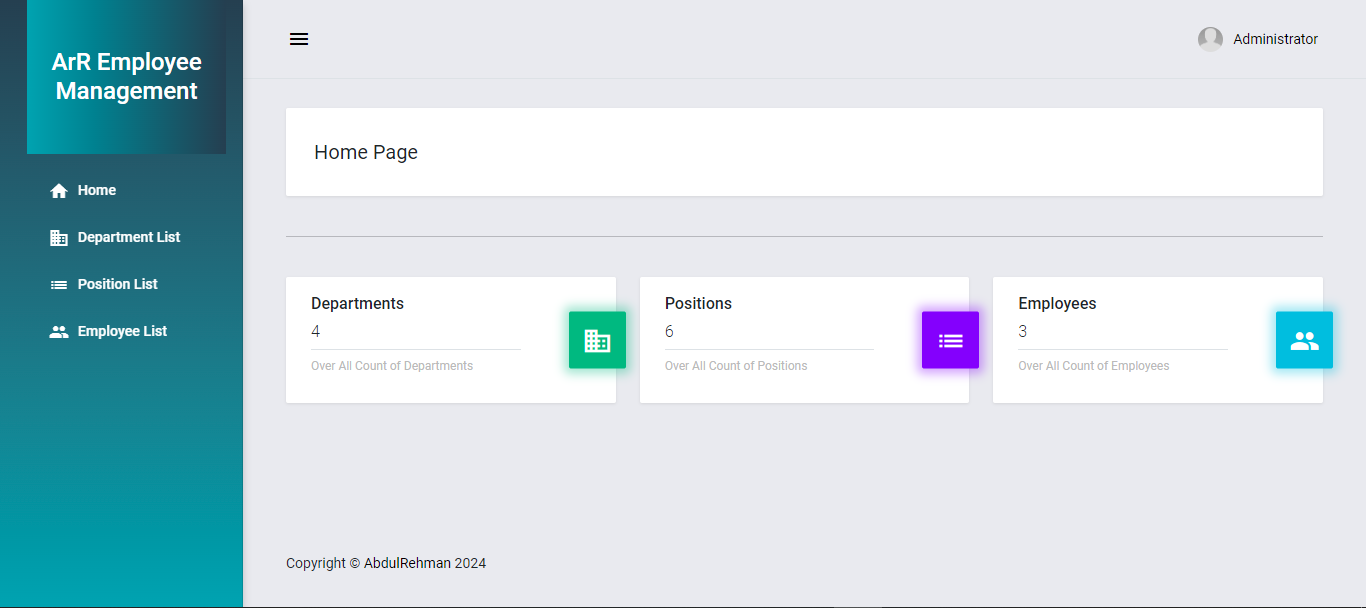
The Employee Management System is a web-based application accessible through standard web browsers. It is compatible with various operating systems, including Windows, macOS, and Linux. It requires a reliable internet connection for optimal performance.

**2.5 Design and Implementation Constraints**

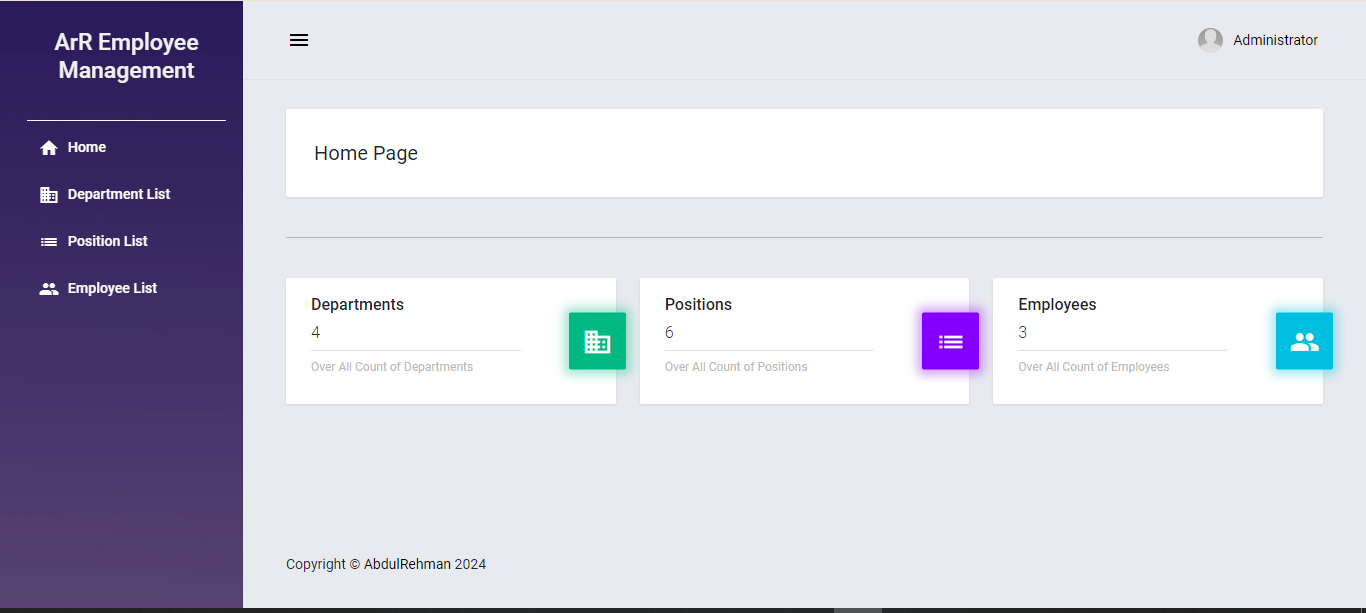
The system must be scalable, secure, and compatible with various web browsers. It should support modern encryption standards and integrate with third-party services. The following constraints are noted:

* **Security**: Ensure data protection and secure communication.
* **Scalability**: Handle an increasing number of users and data without performance issues.
* **Compatibility**: Work with multiple operating systems and browsers.

**Home Page: (before)**

****

**Home Page: (after improving sidebar)**

****

**2.6 User Documentation**

Comprehensive user documentation will be provided to guide users through the system's features. This includes step-by-step instructions, troubleshooting tips, and FAQs. Documentation should be accessible and user-friendly.

**2.7 Assumptions and Dependencies**

The following assumptions and dependencies are noted:

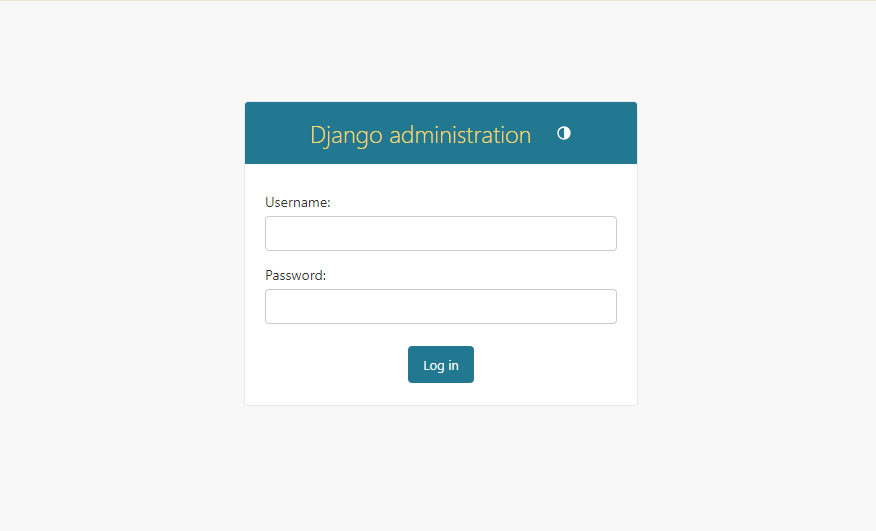
* Users have access to a stable internet connection.
* The system depends on reliable third-party services for some functionalities.
* Users have basic computer literacy skills.
* Development relies on modern programming languages and frameworks.

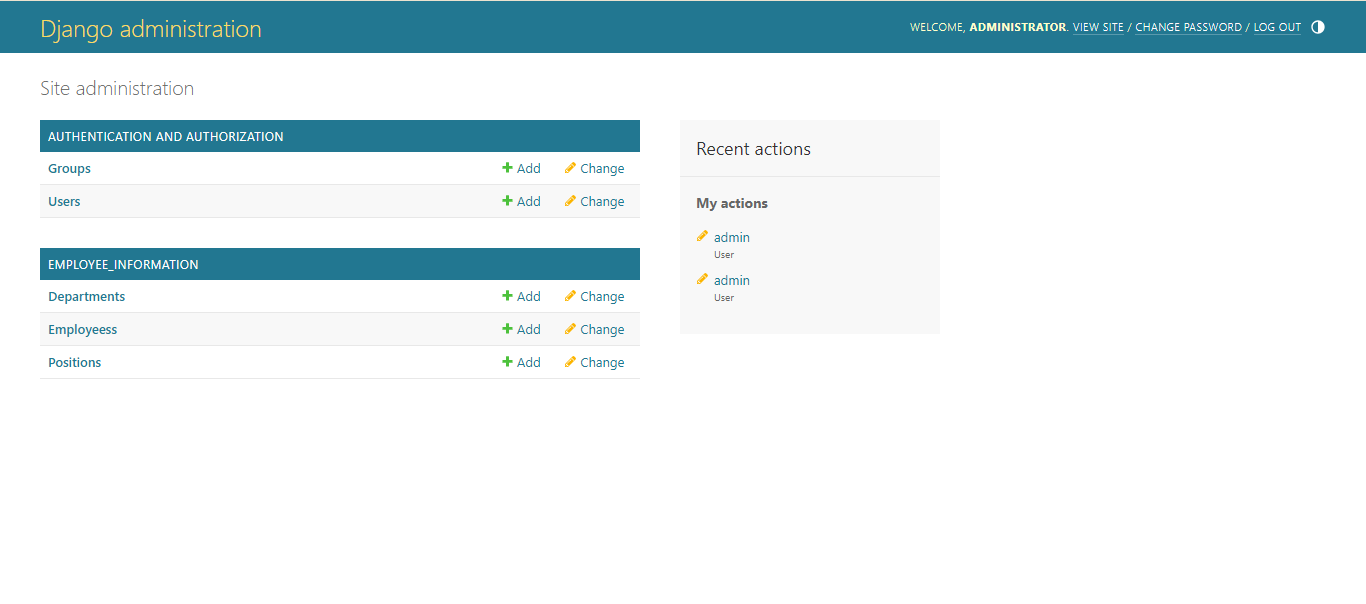
**3. External Interface Requirements**

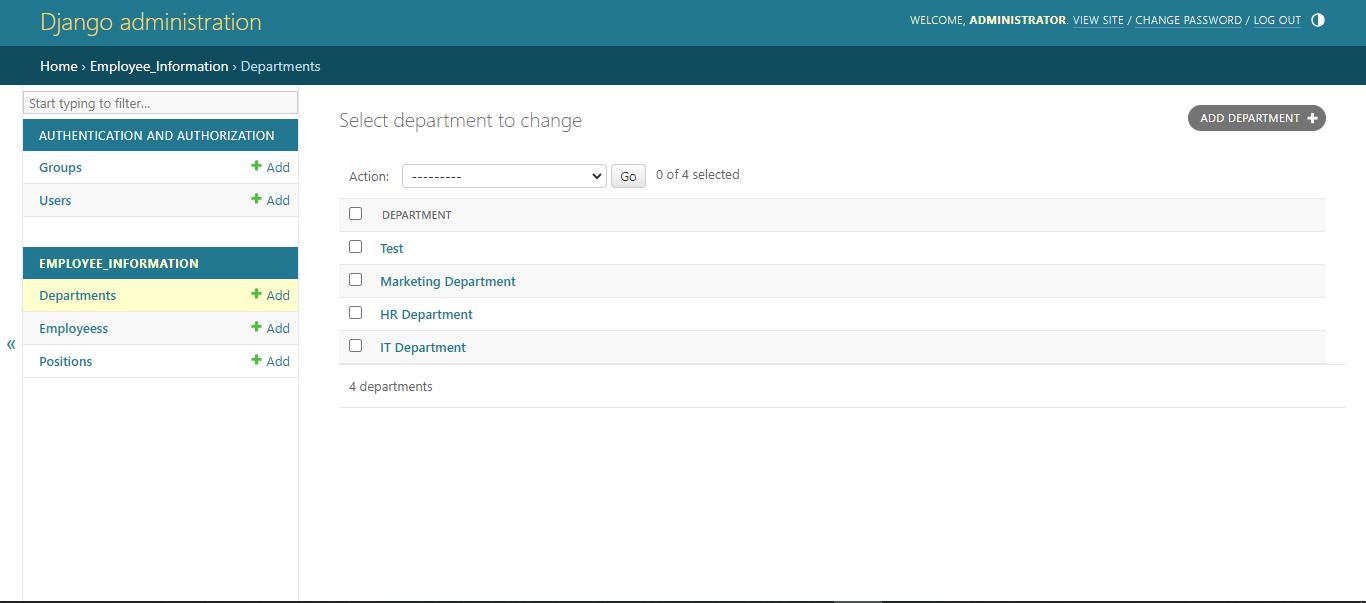
**3.1 User Interfaces**

The system provides user-friendly interfaces for HR admins, managers, and employees. The interfaces should be intuitive and allow users to navigate the system with ease. Key interfaces include:

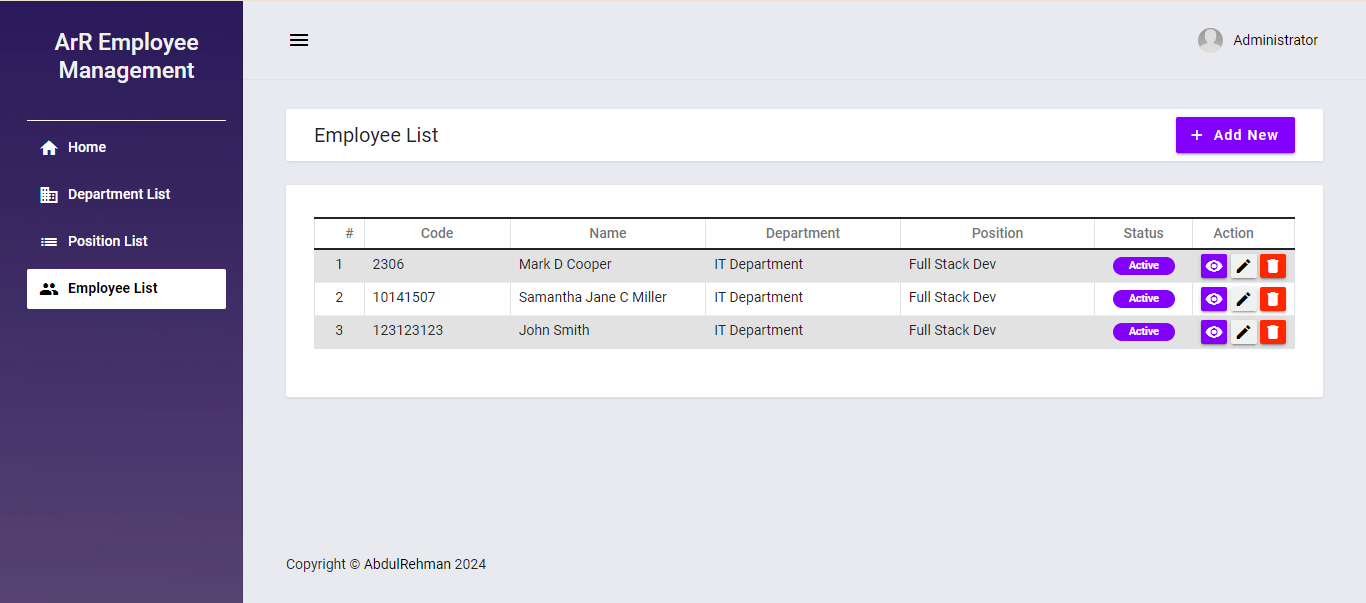
* **HR Admin Interface**: Manage employee records, payroll, and system settings.

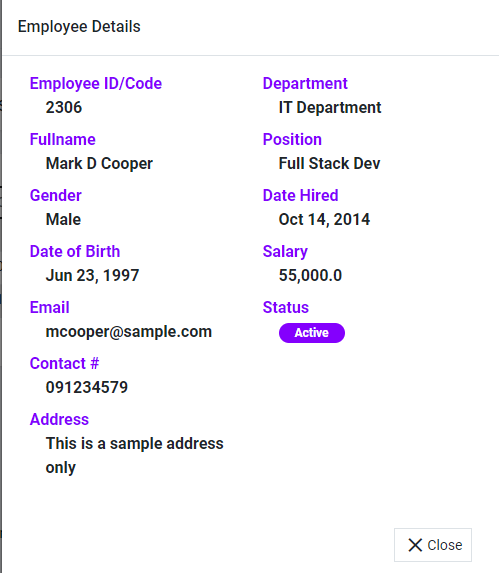




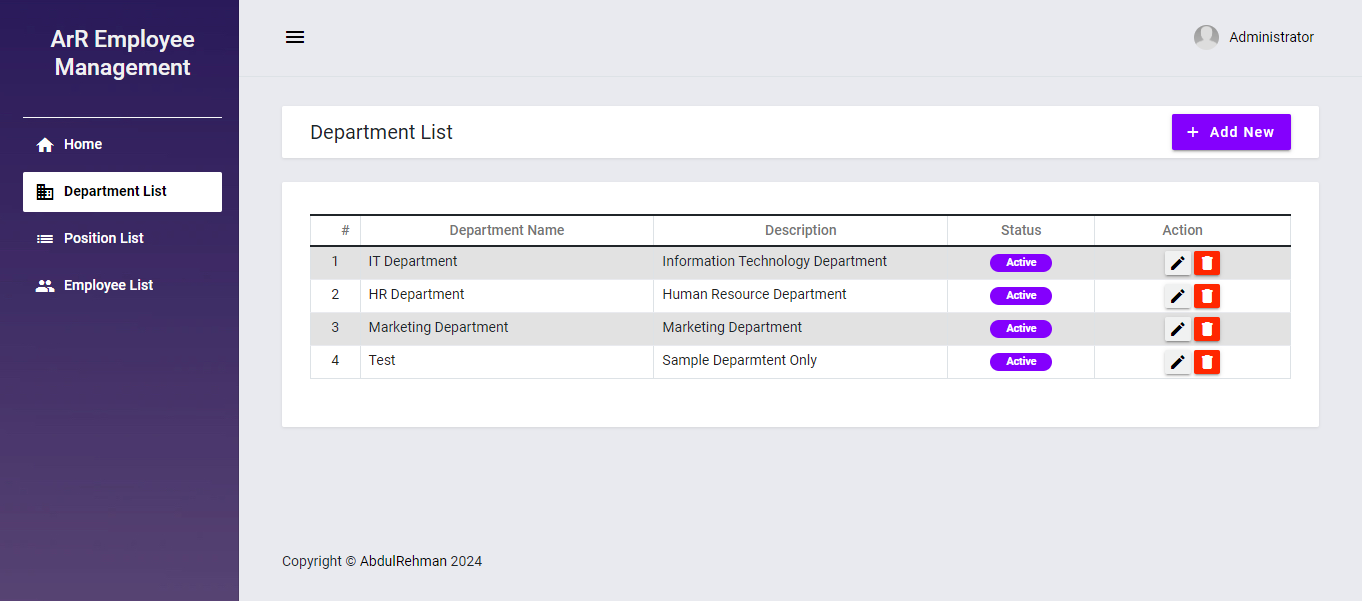


* **Employee Interface**: Access personal information, request leave, and view payroll.





* **Manager Interface**: Oversee team performance and manage employee information.



**3.2 Hardware Interfaces**

The system does not have specific hardware requirements. It is compatible with standard desktop and laptop computers, as well as tablets and smartphones, with at least two cores, 1.6 GHz clock frequency, and 2GB of RAM.

**3.3 Software Interfaces**

The system interfaces with various software components, including:

* **Django**: The backend framework used for development.
* **HTML/CSS/JavaScript**: Frontend technologies for user interface design.
* **SQL Databases**: For storing and retrieving data.

**3.4 Communications Interfaces**

The system uses various communication interfaces to interact with users and third-party services:

* **HTTP/HTTPS**: For secure communication between the client and server.
* **Email**: To send notifications and alerts to users.
* **SMS**: For critical notifications, if needed.

**4. System Features**

**4.1 User Management**

The system provides user management functionality, allowing HR admins to create, update, and delete user accounts. It supports role-based access control (RBAC) to ensure proper authorization.

**4.2 Employee Management**

The system provides functionalities for managing employees, including:

* **Add Employee**: Add new employees by providing personal information and job details.
* **Update Employee Information**: Update existing employee details.
* **Remove Employee**: Remove employees from the system.

**4.3 Payroll Management**

The system supports payroll management, allowing HR admins to calculate payroll and manage deductions. It can generate payroll reports and integrate with accounting systems if required.

**4.4 Report Generation**

The system provides report generation features, allowing HR admins to generate various reports for analytics and compliance. These reports can include payroll reports, employee information, and attendance records.

**5. Other Nonfunctional Requirements**

**5.1 Performance Requirements**

The system must meet certain performance requirements to ensure a seamless user experience:

* The system should respond to user interactions within 2-3 seconds.
* The system should support up to 500 concurrent users without significant performance degradation.
* The system should maintain an uptime of at least 99.5%.

**5.2 Safety Requirements**

Safety requirements focus on data protection and backup:

* **Data Backup**: The backup is created manually in advance in case of system failure.
* **Data Recovery**: In case of system failure, data recovery should be possible within 24 hours.

**5.3 Security Requirements**

Security is a critical aspect of the Employee Management System. The following security measures should be implemented:

* **Authentication**: User authentication should be required for system access.
* **Access Control**: Role-based access control (RBAC) ensures authorized access to sensitive information.
* **Data Encryption**: Sensitive data should be encrypted during transmission and at rest.
* **Prevent Attacks**: The system should prevent common attacks such as SQL injection and cross-site scripting.

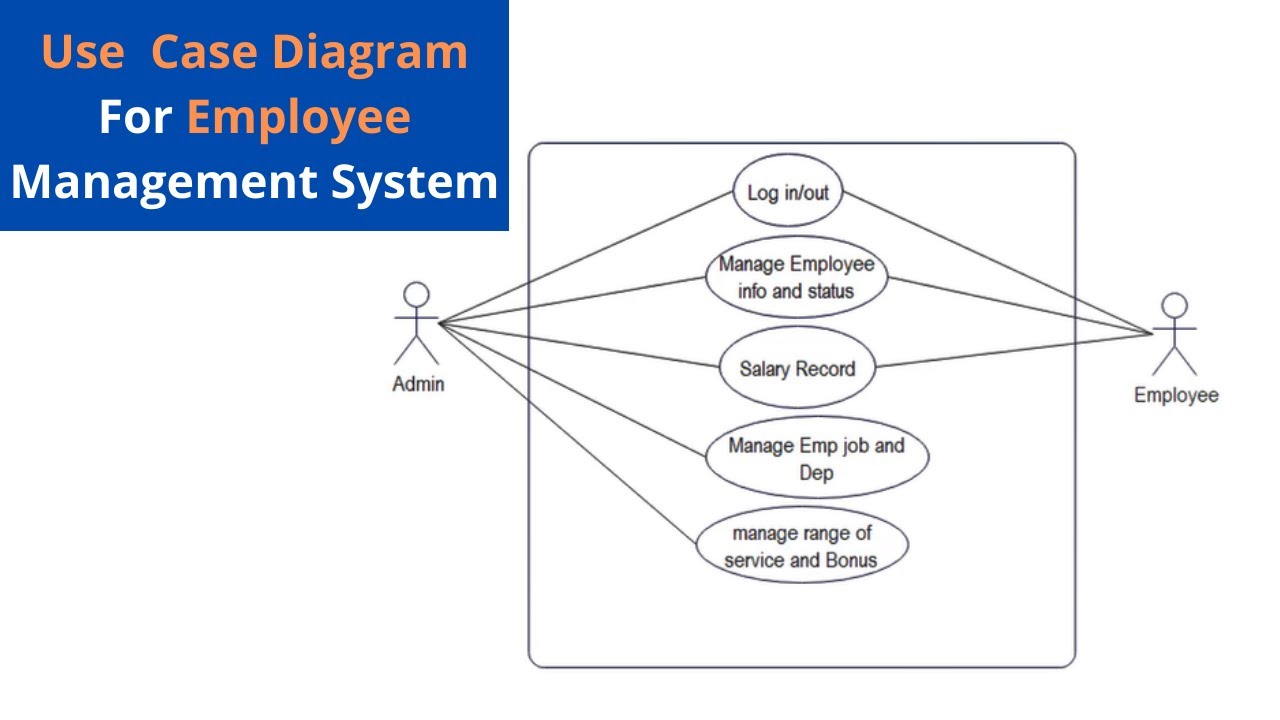
**5.4 Software Quality Attributes**

The system should meet specific software quality attributes to ensure usability and maintainability:

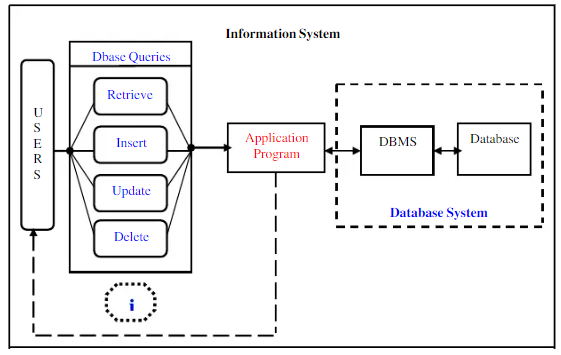
* **Reliability**: The system should have minimal downtime.
* **Maintainability**: The software should be designed with modularity to facilitate maintenance and updates.
* **Scalability**: The system should scale with an increasing number of users.
* **Usability**: The user interface should be intuitive and user-friendly.

**6. Use Case and Activity Diagram**

**6.1 Use Case Diagram**

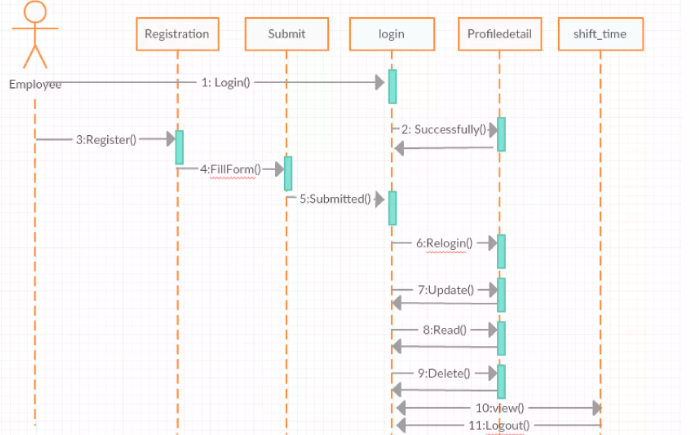


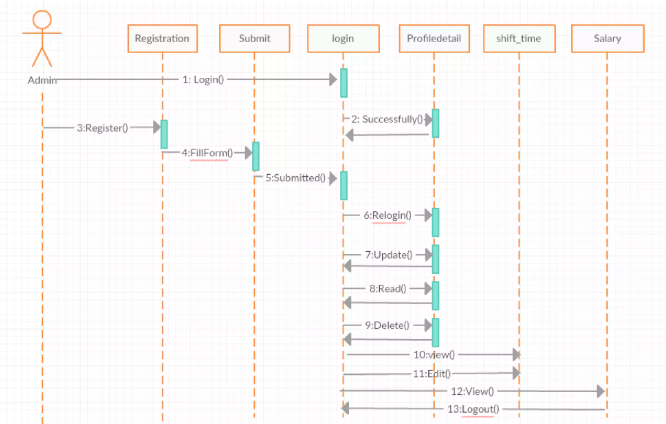
**6.2 Activity Diagram**

****

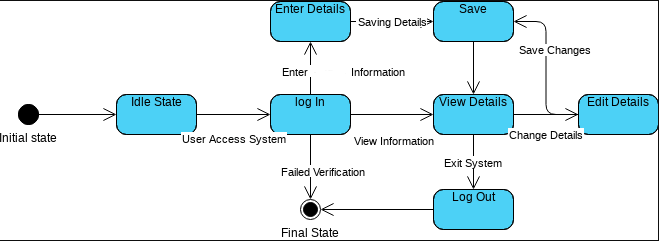
**6.3 Sequence Diagram**

**For User:**

****

**For Admin:  
**

**6.4 State Diagram**

****

**7. Testing**

**7.1 Introduction**

The testing section outlines the approach and results of testing the Employee Management System.

**7.2 Objectives**

The objective of this section is to ensure that the system's key features function as intended and meet the requirements outlined in this report.

**7.3 Testing Methods**

Various testing methods were used to ensure the functionality and reliability of the system:

* **Unit Testing**: Testing individual components to ensure correctness.
* **Integration Testing**: Testing interactions between different components.
* **System Testing**: Testing the complete system to ensure it meets requirements.
* **User Acceptance Testing (UAT)**: Testing the system with end-users to ensure it meets their needs and expectations.

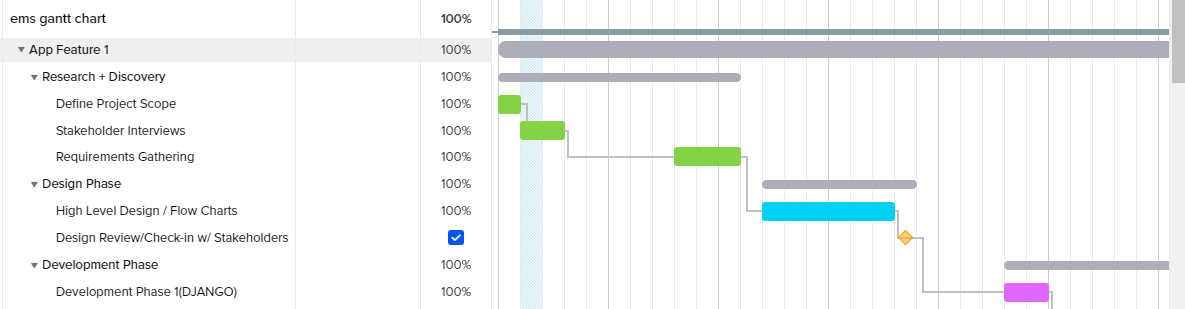
**7.4 Test Cases**

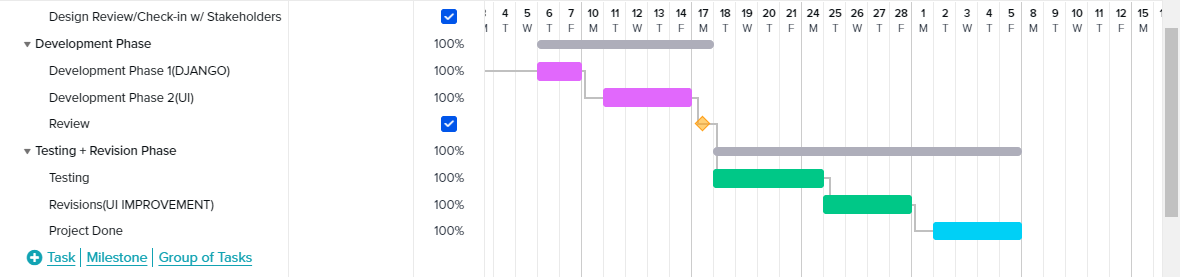
This section outlines test cases used to validate the system's functionality:

* **User Login**: Verify that users can log in with valid credentials and are denied access with invalid credentials.
* **Employee Management**: Test cases for adding, updating, and removing employee records.
* **Payroll Processing**: Test payroll calculations.

**8. Gantt Chart**

Used TeamGantt.com to make this





**9. Glossary**

Key terms used throughout this report:

* **Employee Management System**: A software system designed to automate and manage HR-related tasks.
* **HR**: Human Resources, responsible for employee management.
* **Payroll**: The process of calculating and distributing employee wages.
* **Attendance Tracking**: Monitoring and recording employee attendance.
* **Unit Testing**: Testing individual software components.
* **Integration Testing**: Testing interactions between software components.
* **User Acceptance Testing (UAT)**: Testing with end-users to validate the system.
* **Agile**: A flexible project management approach with iterative development and regular feedback loops.
* **Uptime**: The percentage of time a system is operational and available.