Requirement Document: Office Management System SFS

1. Introduction

This requirement document outlines the functional and non-functional requirements for the development of an Office Management System. The system aims to streamline various office operations and improve efficiency in managing attendance, salary, leave applications, meetings, project management, daily reporting, CVs, business documents, project estimation, and chat messaging using Rocket Chat. This document is intended for the project stakeholders, including developers, managers, and administrators.

2. Project Overview

The Office Management System is being developed to automate and digitize various administrative tasks within the office environment. It will provide a centralized platform for managing employee-related processes and facilitating communication. The key features of the system include attendance management, salary management, online leave application form, meeting notification, project management system, daily reports, CV manager, business document repository, project estimation form, and chat messaging using Rocket Chat.

3. Functional Requirements

3.1 Attendance System

- The system should support an attendance management feature.
- Employees should be able to clock in and clock out using a static IP for authentication.
- The system should record attendance data, including date, time, and employee identification.
- Managers should have access to attendance reports and statistics.

3.2 Salary Management

- The system should provide salary management functionality.
- It should calculate and track employee salaries based on predefined rules and parameters, such as working hours, overtime, and deductions.
- Salary data should be securely stored and accessible only to authorized personnel.
- Employees should be able to view their salary statements and payment history.

3.3 Online Leave Application Form

- Employees should be able to submit leave requests through an online form.
- The form should capture necessary details like leave type, dates, and reason.
- Managers should receive notifications for leave requests and be able to approve or reject them.
- Approved leave should be reflected in the employee's attendance records.

3.4 Meeting Notification

- The system should send notifications to employees for upcoming meetings.
- Notifications should include meeting details, time, location, and agenda.
- Employees should have the option to accept, decline, or request rescheduling for the meeting.
- Managers should be able to track meeting attendance and collect feedback.

3.5 Project Management System

- The system should include a project management module.
- It should allow tracking project progress, tasks, deadlines, and milestones.
- Project managers should be able to assign tasks to team members and monitor their status.
- The system should generate project reports, including Gantt charts and resource allocation.

3.6 Daily Reports

- Employees should be able to submit daily reports on their tasks and activities.
- The system should provide a template or form for entering the report data.
- Managers should be able to review and provide feedback on the reports.
- The system should maintain a record of submitted reports for future reference.

3.7 CV Manager

- The system should include a CV management feature.
- HR personnel should be able to store, search, and manage employee CVs.
- The system should allow uploading CV documents in various formats, such as PDF or Word.
- HR personnel should be able to update CVs as needed and retrieve them when required.

3.8 Business Document of Project

- The system should provide a centralized repository for project-related documents.
- Users should be able to upload, store, and retrieve documents.
- Document categories and tags should be used for efficient organization and search ability.
- Access to documents should be controlled based on user roles and permissions.

3.9 Project Estimation Form

- The system should include a project estimation form.
- Users should be able to input project requirements, such as scope, features, and timeline.
- The system should calculate cost and time estimations based on predefined algorithms.
- Users should be able to generate project estimation reports for client communication.

3.10 Chat Messaging using Rocket Chat

- The system should integrate with Rocket Chat for chat messaging functionality.
- Employees should be able to communicate with each other using chat channels.
- The system should support private messaging, group chats, and file sharing.
- Chat history should be logged and accessible for reference.

4. Non-Functional Requirements

- Performance: The system should respond promptly and handle multiple concurrent users efficiently.
- Security: The system should have robust security measures to protect employee data and sensitive information.
- Usability: The user interface should be intuitive and user-friendly, requiring minimal training.
- Compatibility: The system should be compatible with modern web browsers and mobile devices.
- Scalability: The system should be designed to handle future growth and increased usage.

5. System Architecture

The Office Management System will follow a multi-tier architecture, utilizing ReactJS for the front-end, Node.js for the back-end, and MongoDB for the database. The system will be built as a web application, leveraging the capabilities of these technologies to provide a robust and scalable solution. Integration with Rocket Chat will be accomplished using Rocket Chat's APIs.

6. Data Requirements

The system will utilize MongoDB as the database to store and manage the following data entities:

- Employee information (e.g., name, contact details, job title)
- Attendance records (date, time, status)
- Salary details (basic salary, overtime, deductions)
- Leave application details (leave type, dates, reason)
- Meeting details (time, location, agenda)
- Project data (tasks, deadlines, milestones)
- Daily reports (employee, date, tasks, achievements)
- CV documents (employee, file)
- Business documents (project-related files)
- Project estimation form data (requirements, estimation results)
- Chat messages (text, sender, receiver, timestamp)

7. User Interface

The user interface will be developed using ReactJS, ensuring a responsive and intuitive design. The front-end will incorporate modern UI/UX principles to provide an engaging user experience. The application will be designed to be compatible with different web browsers and responsive across various devices.

8. Assumptions and Constraints

- The development team will have expertise in ReactJS, Node.js, and MongoDB to efficiently build and maintain the system.
- Adequate server infrastructure will be provided to ensure smooth operations of the Node.js back-end and MongoDB database.
- The system will be developed following best practices and coding standards for ReactJS and Node.js development.
- The necessary dependencies and libraries required for ReactJS, Node.js, and MongoDB will be identified and included in the development environment.

9. Dependencies

- Integration with Rocket Chat for chat messaging functionality will require API documentation and access credentials.
- The system may depend on external libraries or frameworks for certain features, such as date/time calculations or chart generation.