



**National University**  
of computer and emerging sciences

**Foundation of Advancement  
Of Science and Technology**



## **SOFTWARE REQUIREMENT & DESIGN SPECIFICATIONS**

# **EMPLOYEE ATTENDANCE MANAGEMENT SYSTEM**

**VERSION 1.0**

**INSTRUCTOR MS. NIDA MUNAWAR (BCS-5E)**

**SOFTWARE DESIGN & ANALYSIS (CS-3004)**

- **MUHAMMAD HAMZA (K21-4579)**
- **MUHAMMAD SALAR (K21-4619)**

Foundation of Advancement of Science and Technology  
National University of Computer and Emerging Sciences  
Department of Computer Science  
Karachi, Pakistan  
Thursday, December 7, 2023

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Purpose Of Document . . . . .	3
1.2	Intended Audience . . . . .	3
1.3	Definition of Terms, Acronyms and Abbreviations . . . . .	3
1.4	Document Convention . . . . .	3
<b>2</b>	<b>Overall System Description</b>	<b>4</b>
2.1	Project Background . . . . .	4
2.2	Project Scope . . . . .	4
2.3	Not in Scope . . . . .	4
2.4	Project Objectives . . . . .	4
2.5	Stakeholders . . . . .	4
2.6	Operating Environment . . . . .	5
2.7	System Constraints . . . . .	5
2.8	Assumptions & Dependencies . . . . .	5
<b>3</b>	<b>External Interface Requirements</b>	<b>6</b>
3.1	Hardware Interfaces . . . . .	6
3.2	Software Interfaces . . . . .	6
3.3	Communications Interfaces . . . . .	6
<b>4</b>	<b>Functional Requirements</b>	<b>7</b>
4.1	Functional Hierarchy . . . . .	7
4.2	Use Cases . . . . .	8
4.2.1	Attendance Management System . . . . .	8
4.2.2	Record Employee Attendance . . . . .	9
4.2.3	View Attendance History . . . . .	10
4.2.4	Manage Employee Leaves . . . . .	11
<b>5</b>	<b>Non-functional Requirements</b>	<b>13</b>
5.1	Performance Requirements . . . . .	13
5.2	Safety Requirements . . . . .	13
5.3	Security Requirements . . . . .	13
5.4	User Documentation . . . . .	13
<b>6</b>	<b>System Architecture</b>	<b>14</b>
6.1	System Level Architecture . . . . .	14
6.2	Software Architecture . . . . .	15
<b>7</b>	<b>Design Strategy</b>	<b>17</b>
<b>8</b>	<b>Detailed System Design</b>	<b>18</b>
8.1	Database Design . . . . .	18
8.1.1	Class Diagram . . . . .	18
8.1.2	ER Diagram . . . . .	20
8.1.3	Data Dictionary . . . . .	21

<b>9</b>	<b>Application Design</b>	<b>22</b>
9.1	Sequence Diagram . . . . .	22
9.1.1	Attendance Management Sequence . . . . .	23
9.1.2	Login Sequence . . . . .	24
9.1.3	Manager Sequence . . . . .	25
9.1.4	Employee Sequence . . . . .	26
9.2	State Chart Diagram . . . . .	26
9.3	Activity Diagram . . . . .	27
9.3.1	Attendance Management System Activity Diagram . . . . .	28
9.3.2	Employee Attendance Activity Diagram . . . . .	29
9.3.3	Employee Registration Activity Diagram . . . . .	30
9.3.4	Set Permissions Activity Diagram . . . . .	31
9.4	Component Diagram . . . . .	32
9.5	Deployment Diagram . . . . .	33
<b>10</b>	<b>References &amp; Citations</b>	<b>34</b>
<b>11</b>	<b>Appendices</b>	<b>35</b>
11.1	Project Timeline . . . . .	35
11.1.1	Week 1 . . . . .	35
11.1.2	Week 2 . . . . .	35
11.1.3	Week 3 . . . . .	35
11.1.4	Week 4 . . . . .	36
11.1.5	Week 5 . . . . .	36

# SOFTWARE REQUIREMENT SPECIFICATIONS

## 1 Introduction

### 1.1 Purpose Of Document

This document is a Software Requirements Specification (SRS) that provides a detailed overview of the Employee Attendance Management System project. It describes the problem that needs to be solved, the proposed solution, the scope of the project, the features and functions of the system, the data model, the tools and techniques used, the project timeline, and the conclusion. This document is intended to help the development team, the stakeholders, and any other parties involved in the project to have a clear understanding of the requirements and goals.

### 1.2 Intended Audience

The intended audience for this document includes:

- **Development Team:** This group comprises the developers, coders, and software architects tasked with the creation, execution, and validation of the Employee Attendance Management System.
- **Project Managers:** These are the individuals charged with the administration of the project, the distribution of resources, and the assurance of its congruence with the company's objectives and deadlines.
- **Stakeholders:** These are the people or entities within the company who have a significant interest in the fruitful development and implementation of the Employee Attendance Management System. This category may encompass senior leadership, supervisors, and staff members who will utilize the system.
- **End Users:** These are the staff members and supervisors in the company who will engage with and derive advantages from the Employee Attendance Management System upon its rollout.

### 1.3 Definition of Terms, Acronyms and Abbreviations

Term	Description
ASP	Active Server Pages, a server-side script engine for dynamically generated web pages
SSL	Secure Socket Layer, a protocol for establishing encrypted links between networked computers
SQL	Structured Query Language, a standardized language for managing relational databases
JSON	JavaScript Object Notation, a data-interchange format that is easier for humans
SRS	Software Requirements Specification, a document for the software requirements for a system
SDS	Software Design Specification, a document for the design details of a system
HTML	Hyper-Text Markup Language, the standard markup language for web pages
CSS	Cascading Style Sheets, a language used for the presentation of a document in HTML or XML
TLS	Transport Layer Security, a protocol to provide communications security over a network
APIs	Application Programming Interface, a set of routines, and protocols for software applications

### 1.4 Document Convention

This document uses font size 16 for main headings, 14 for sub headings and 12 for body, also the overall font style is Times New Roman. Line spacing is single (1.0)

## 2 Overall System Description

### 2.1 Project Background

In the ever-evolving corporate landscape, adept management of tasks is essential for a company's prosperity. The inception of this project is rooted in the difficulties that numerous organizations encounter in effectively coordinating and monitoring employee responsibilities, which often results in unmet deadlines and diminished clarity. The demand for an integrated and intuitive system has given rise to the creation of the Employee Attendance Management System. This platform endeavours to simplify the allocation, supervision, and fulfilment of tasks, thereby cultivating an orderly and efficient workplace.

### 2.2 Project Scope

The Employee Attendance Management System project is centred on constructing an online platform that streamlines efficient attendance tracking within the company. The principal features include attendance logging, absence monitoring, deleting erroneous entries, user verification, and database management. The system aims to offer a complete solution that empowers authorized individuals to oversee attendance reliably, promoting accountability and punctual adherence to work schedules. The system will deliberately exclude any features not directly related to attendance management.

### 2.3 Not in Scope

The subsequent features are categorically excluded from the purview of this project:

- Complex project management capabilities that surpass basic attendance tracking.
- Synchronization with third-party attendance or HR management tools.
- Financial operations such as payroll processing or generating invoices.

### 2.4 Project Objectives

**Project Objectives** The project is committed to accomplishing the following goals:

- Simplify the process of recording and monitoring attendance.
- Improve visibility in attendance tracking within the company.
- Offer an intuitive Employee Attendance Management System for authorized users.
- Showcase effective database management for handling attendance data.

### 2.5 Stakeholders

**Stakeholders** The system's stakeholders encompass:

- **Managers:** Accountable for scheduling employee attendance and supervising adherence to work hours.
- **Employees:** Individuals who will engage with the system to check in and verify their attendance records.
- **Database Administrators:** Entrusted with the upkeep and administration of the core database system.

## 2.6 Operating Environment

The system is designed to function within a web-based framework. The hardware requirements are inclusive of standard computing devices that have access to the internet. It will be compatible with prevalent operating systems such as Windows or Linux. Additionally, the system is built to operate seamlessly alongside web browsers like Chromium-based browsers or Firefox.

## 2.7 System Constraints

System Constraints The system is subject to the following limitations:

- **Software Constraints:** The system must be compatible with designated web browsers.
- **Hardware Constraints:** The system requires standard computing devices that are connected to the internet.
- **Legal Constraints:** The system must comply with applicable data protection and privacy regulations.
- **User Constraints:** Users should have a basic understanding of web interfaces and operations.

## 2.8 Assumptions & Dependencies

The system's operation is predicated on the following:

- **Assumptions:** It is presumed that users possess an elementary proficiency with web interfaces and that the organization adheres to pertinent data protection legislation.
- **Dependencies:** The system's functionality is reliant on a consistent internet connection to facilitate instantaneous updates. Moreover, the progression of the project is contingent upon the accessibility of the requisite development assets.

## 3 External Interface Requirements

### 3.1 Hardware Interfaces

The Employee Attendance Management System is an online application that is not bound by specific hardware requirements. It is engineered to function on universally available computing devices such as desktops, laptops, and palmtops. The system's hardware interfaces are universally applicable and encompass any device equipped with web browsing functionality and internet access.

### 3.2 Software Interfaces

The system engages with the subsequent software elements:

- **MySQL Database (Version 8.2.0):** The Employee Attendance Management System utilizes MySQL for its database needs, encompassing data storage, access, and administration. The MySQL database is integrated via the XAMPP server.
- **Express.js (Node.js Framework):** The system's backend infrastructure is constructed with Express, a Node.js framework, which manages API calls, user authentication, and database exchanges.
- **React.js (Version 18):** The system's frontend is crafted using React, ensuring a lively and adaptable user interface for the Employee Attendance Management System.
- **Node.js (Version 18.16.1):** Node.js operates as the server-side JavaScript environment, executing server tasks and enabling interaction between the frontend and the database.

Data Items or Messages Exchanged:

- **Between Frontend and Backend:** The system utilizes JSON data formats to convey attendance records, verify user credentials, and deliver system feedback.
- **Between Backend and Database:** The communication involves SQL commands and their corresponding outcomes to execute database tasks.

### 3.3 Communications Interfaces

The system adheres to established web communication standards, which include:

- **HTTP/HTTPS Protocols:** Utilized for data exchange between the client-side (web browser) and the server-side (Node.js/Express backend), with HTTPS ensuring encrypted and secure data transfer.
- **RESTful API:** The backend offers RESTful API routes to streamline interactions between the frontend and backend, covering functionalities such as attendance management, user verification, and more.
- **Communication Security:** The implementation of SSL or TLS protocols will be in place to safeguard data integrity during transmission.
- **Data Transfer Rates:** While dependent on the user's internet speed, the system will be optimized for swift data transfer to enhance performance.
- **Synchronization Mechanisms:** Asynchronous communication will be employed to enable non-disruptive interactions, ensuring a smooth user experience.

## 4 Functional Requirements

### 4.1 Functional Hierarchy

For the Employee Attendance Management System project, the restructured functional hierarchy is:

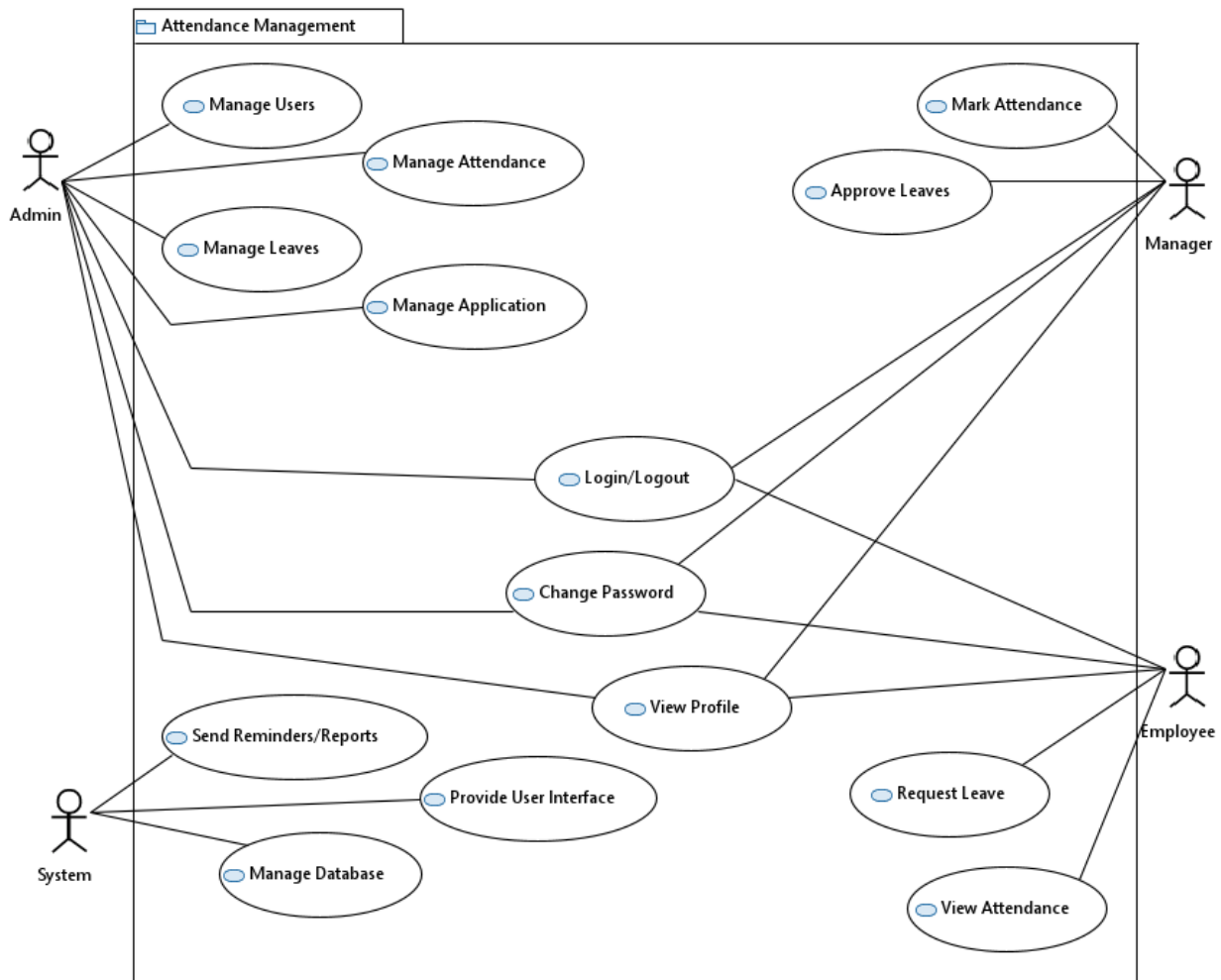
- **User Authentication**
  - Sign In
  - Sign Out
- **Attendance Tracking**
  - Record Attendance
  - Review Attendance
  - Approve Absence
  - Modify Attendance Entry
  - Remove Attendance Entry
- **Employee Directory (for managers)**
  - Register Employee
  - Browse Employee Records
  - Update Employee Information
  - Terminate Employee Record
- **Attendance Reporting (for managers)**
  - Generate Attendance Report
  - Access Attendance History
  - Edit Attendance Report
  - Discard Attendance Report
- **Employee Interface**
  - Access Personal Profile
  - Check Attendance Status
  - Confirm Attendance

This hierarchy outlines the primary operations of the Employee Attendance Management System and its organizational structure.



## 4.2 Use Cases

### 4.2.1 Attendance Management System



- **Actors:**

- Manager
- Employee
- Administrator
- System

- **Manager:**

- Log Attendance: The manager can record a new attendance entry for employees.
- Monitor Attendance: The manager can oversee the attendance records of specific employees.
- Review Attendance Status: The manager can check the status of each employee's attendance (present, absent, late).
- Edit Attendance: The manager can modify the details of the attendance record.
- Remove Attendance Entry: The manager can delete an attendance entry if it is incorrect or no longer needed.

- **Employee:**

- Check Attendance: The employee can review their own attendance records.
- Request Leave: The employee can request leave.

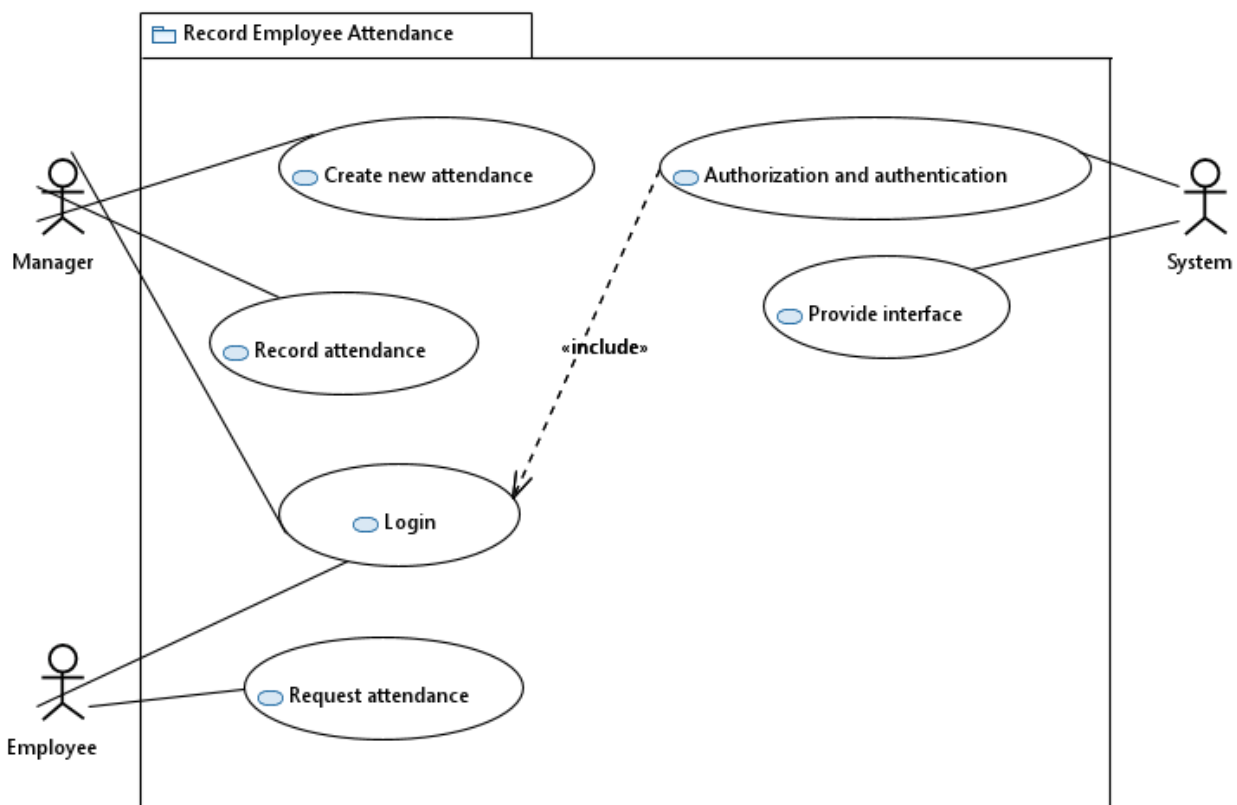
- **Administrator:**

- Manage Users (Add/Remove): The administrator can add or remove users from the system.
- Assign User Roles: The administrator can designate roles for the users (manager, employee).
- Adjust System Settings: The administrator can modify system settings like security protocols, data backups, etc.

- **System:**

- Dispatch Notifications: The system sends alerts regarding attendance logs, updates, etc.
- Compile Reports: The system can produce summaries about attendance records, employee punctuality, etc.

#### 4.2.2 Record Employee Attendance



Primary Actor: Employee, Manager

Supporting Actor: System

Preconditions: The employee is scheduled to work.

Postconditions: The employee's attendance record is updated in the system.

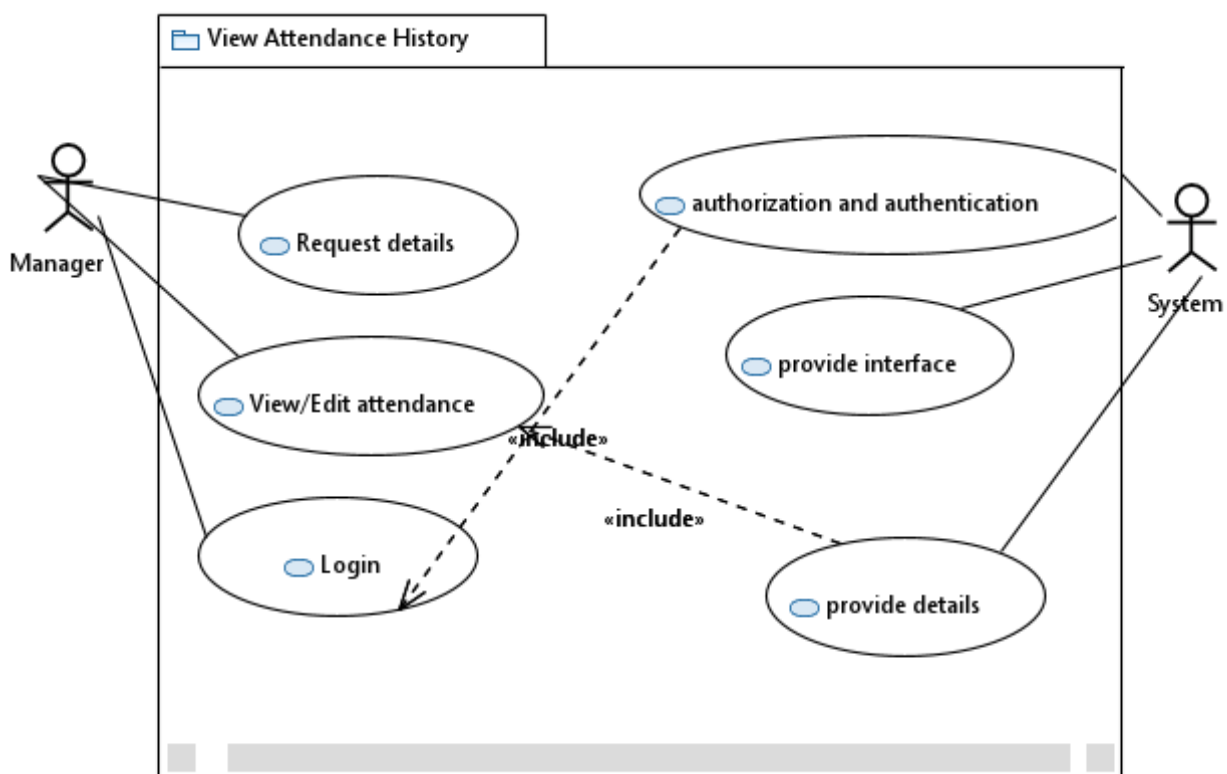
Main Success Scenario:

1. The employee arrives at the workplace.
2. The employee logs into the attendance recording system.
3. The manager marks the employee's attendance.
4. The employee begins work.

Extensions:

- If the employee encounters technical issues while logging in:
  1. The employee notifies the manager.
  2. The manager manually records the employee attendance.
- If the employee forgets to log in:
  1. The manager is alerted to the missing entry.
  2. The manager manually records the employee's entry time.

#### 4.2.3 View Attendance History



Primary Actor: Manager

Supporting Actor: System

Preconditions: The manager has valid credentials, and the attendance records are available in the system.

Postconditions: Manager views the selected employee's attendance history.

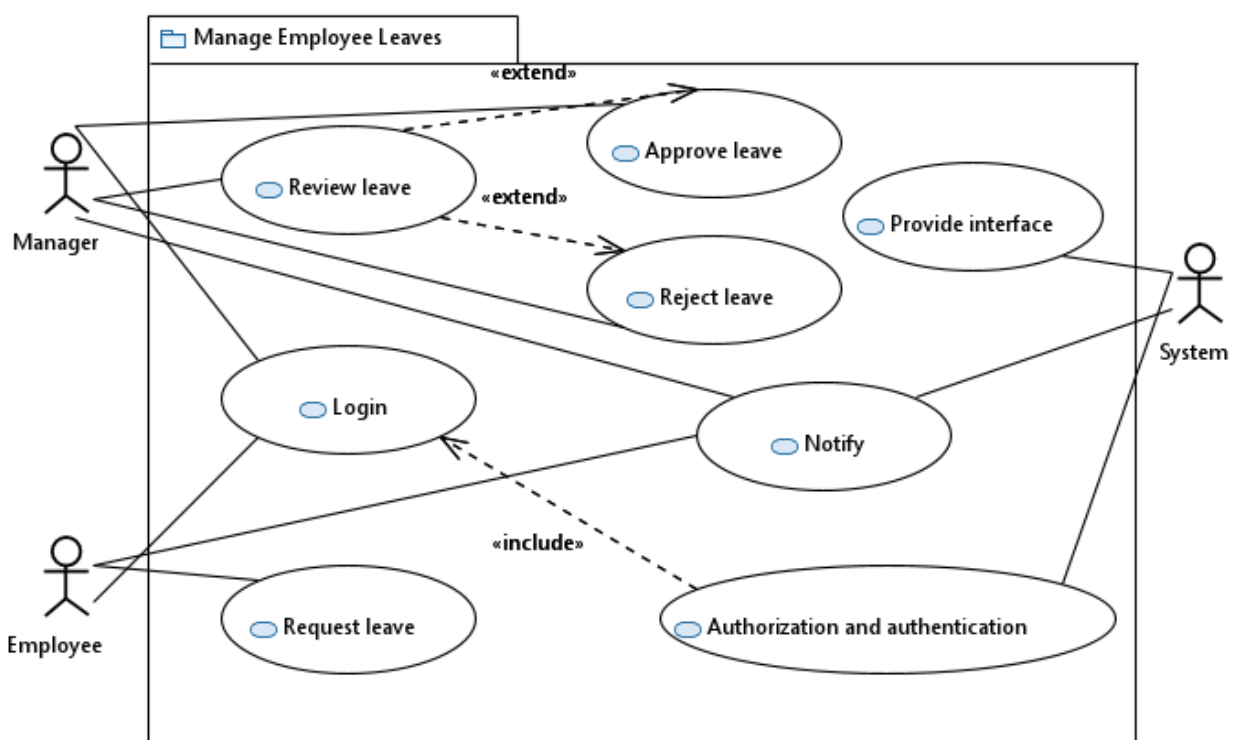
Main Success Scenario:

1. Manager logs in to the attendance management system.
2. The system validates the credentials and grants access.
3. Manager selects the "View Attendance History" option.
4. The system presents a list of employees.
5. Manager selects a specific employee.
6. The system retrieves and displays the attendance history of the selected employee.

Extensions:

- Invalid Credentials:
  1. If the entered credentials are invalid, the system notifies the Manager and prompts them to enter valid credentials.
- No Employees Listed:
  1. If there are no employees in the list, the system notifies the Manager that there are no records available.

#### 4.2.4 Manage Employee Leaves



Primary Actor: Manager, Employee

Supporting Actor: System

Preconditions: An employee has requested time off.

Postconditions: The employee's absence/leave is accurately reflected in the attendance system.

Main Success Scenario:

1. The employee submits a leave request to the manager.
2. The manager reviews and approves the request.
3. The attendance database is updated to reflect the approved absence.

Extensions:

- If the leave request is denied:
  1. The HR manager communicates the decision to the employee.
  2. The employee's attendance record remains unchanged.
- If the employee fails to submit a leave request:
  1. The manager follows up with the employee to address the absence.
  2. The absence is recorded as unauthorized in the attendance system.

## 5 Non-functional Requirements

### 5.1 Performance Requirements

- **Response Time:** The system is expected to deliver immediate reactions to user inputs to ensure a smooth and efficient user experience.
- **Concurrent Users:** The system is designed to accommodate multiple users simultaneously without noticeable performance issues.
- **Database Query Performance:** Operations involving the retrieval and modification of attendance data should be completed within seconds to promote effective attendance management.

### 5.2 Safety Requirements

- **Data Integrity:** Measures must be established to preserve the accuracy and consistency of attendance records, averting any potential data loss or distortion.
- **User Authentication:** A robust authentication system is required to restrict access to authorized users only, safeguarding confidential data.
- **Backup and Recovery:** Consistent backups of the attendance database should be conducted to reduce the risk of data loss, accompanied by a well-defined data recovery strategy in the event of system malfunctions.

### 5.3 Security Requirements

- **Data Encryption:** All data exchanges between the user interface and the server are to be encrypted via HTTPS to guarantee data confidentiality and security.
- **User Authorization:** The implementation of role-based access control is essential to limit access to critical features, ensuring that only authorized staff can execute operations such as logging attendance and editing records.
- **Password Security:** Passwords for user accounts must be securely encrypted with hashing algorithms to prevent unauthorized breaches.
- **Audit Trail:** The system is required to keep a comprehensive audit log that records significant actions and transactions for enhanced security and traceability.

### 5.4 User Documentation

- **User Guides:** Detailed user guides must be made available to instruct both supervisors and staff on the optimal use of the Employee Attendance Management System.
- **In-App Support:** The application should incorporate a built-in help feature, providing immediate, relevant support and instructional content for users.
- **Educational Resources:** Supplementary educational resources, such as instructional videos or walkthroughs, should be offered to aid in user familiarization and ensure a seamless adaptation to the system.
- **FAQ Section:** An FAQ segment should be present in the documentation to resolve routine inquiries and issues encountered by users.

# SOFTWARE DESIGN SPECIFICATIONS

## 6 System Architecture

### 6.1 System Level Architecture

The Employee Attendance Management System is architected as a web-based platform, adhering to the client-server model. This system is structured into several fundamental components:

- Frontend (Client-Side):
  - Developed with React, the frontend delivers a dynamic and interactive UI for the Employee Attendance Dashboard.
  - It encompasses modules for managing attendance, user login, and comprehensive dashboard administration.
- Backend (Server-Side):
  - Utilizing Express, a Node.js web application framework, the backend manages API interactions, user verification, and database communication.
  - It is tasked with processing user inquiries, validating credentials, database interfacing, and propagating information to the frontend.
- Database (DBMS):
  - MySQL serves as the database system, organizing and preserving data related to employees and their attendance.
  - The database maintains tables for employee records, attendance logs, and attendance tracking, ensuring smooth data access and management.
- Communication Standards:
  - The platform employs HTTP/HTTPS protocols for secure data exchange between the client (web browser) and the server (Express backend).
  - RESTful APIs are established to streamline the interaction between the frontend and backend, overseeing attendance management, user verification, and other essential operations.

#### System Decomposition:

- Client Layer (Frontend):
  - Functionalities: Attendance Logging, Hours Monitoring, Access Control, Interface Design.
  - Roles: Visualize the interface, interact with users, initiate server communication and present data.
- Server Layer (Backend):
  - Elements: API Services, Authentication Procedures, Database Connectivity.
  - Roles: Respond to client submissions, authenticate identities, manage database operations, and communicate with the client.

- Data Storage Layer (Database):
  - Components: Employee Directory, Attendance Records, Attendance Assignments.
  - Roles: Maintain and organize data on personnel, attendance specifics, and assignment of attendance responsibilities.

**Interaction Protocols:**

- Client-Server Dynamics:
  - The frontend interfaces with the backend via specified API services, exchanging data in JSON format.
- Server-Database Dynamics:
  - The backend engages with the MySQL database through SQL operations for data insertion, extraction, modification, and deletion.

**Execution Framework:**

- The React framework on the client side is operational within the user's web browser.
- The Express framework on the server side functions on a web server.
- The MySQL database system is stationed on a database server, interfaced through XAMPP.

**Global Design Strategies:**

- Error Handling:
  - A centralized system for error management will be deployed to provide uniform and elucidative error communications to the client when discrepancies occur.
- Security Framework:
  - Security protocols, including data safeguarding, identity verification, and rights management, will be implemented on both the client and server layers.

## 6.2 Software Architecture

The Employee Attendance Management System is structured using a three-tier architecture, which includes the User Interface (UI) Layer, Middle Tier, and Data Access Layer. Below is a detailed description of each layer and the interactions among them:

**Three-Tier Architecture:**

- User Interface Layer (Frontend):
  - The UI Layer is the visual front that users interact with. It is crafted using React to offer a dynamic and user-friendly Attendance Management Dashboard.
  - Key functionalities include attendance tracking, leave management, employee authentication, and dashboard administration.



- Middle Tier (Backend):
  - Serving as the bridge between the UI and the database, the Middle Tier is developed with Express, a Node.js framework.
  - It manages API requests, executes business logic, and interfaces with the Data Access Layer.
  - This tier is equipped with API routing, authentication services, and database communication logic.
- Data Access Layer (Database Layer):
  - Utilizing MySQL as the database system, the Data Access Layer securely stores employee attendance records, personal details, and leave data.
  - It performs SQL operations for data manipulation, including insertion, retrieval, modification, and deletion.

**Key Interactions:**

- From UI Layer to Middle Tier:
  - The UI Layer dispatches API calls to the Middle Tier, transmitting data pertinent to attendance management and employee verification.
- From Middle Tier to Data Access Layer:
  - The Middle Tier executes business rules, interacts with the Data Access Layer, and runs SQL commands on the MySQL database.
- From Data Access Layer to Middle Tier:
  - The Data Access Layer sends back query outcomes to the Middle Tier, which then processes the information and relays the necessary responses back to the UI Layer.

## 7 Design Strategy

The design strategy for the Employee Attendance Management System is anchored in principles that ensure flexibility, scalability, and maintainability. The following key considerations have shaped the system's architecture and high-level organization:

### 1. Future System Extension or Enhancement:

- **Strategy:** The system's modular design facilitates straightforward extensions or enhancements.
- **Reasoning:** This modularity allows for seamless integration of new features or updates, enabling the system to evolve with the organization's needs.

### 2. System Reuse:

- **Strategy:** A three-tier architecture enhances system reuse by delineating the UI, business logic, and data layers.
- **Reasoning:** Independent layer reuse streamlines updates and maintenance, fostering efficient code management.

### 3. User Interface Paradigms:

- **Strategy:** The system leverages React for a responsive and dynamic UI, in line with contemporary UI standards.
- **Reasoning:** An intuitive interface promotes user engagement and adoption, with React enabling a fluid single-page application experience.

### 4. Data Management (Storage, Distribution, Persistence):

- **Strategy:** MySQL is selected for its comprehensive data management capabilities.
- **Reasoning:** MySQL ensures dependable data storage and management, supporting intricate queries and data relationships.

### 5. Concurrency and Synchronization:

- **Strategy:** Asynchronous communication and RESTful APIs are utilized for effective concurrency management.
- **Reasoning:** This approach guarantees a responsive UI and streamlined synchronization, accommodating multiple users simultaneously.

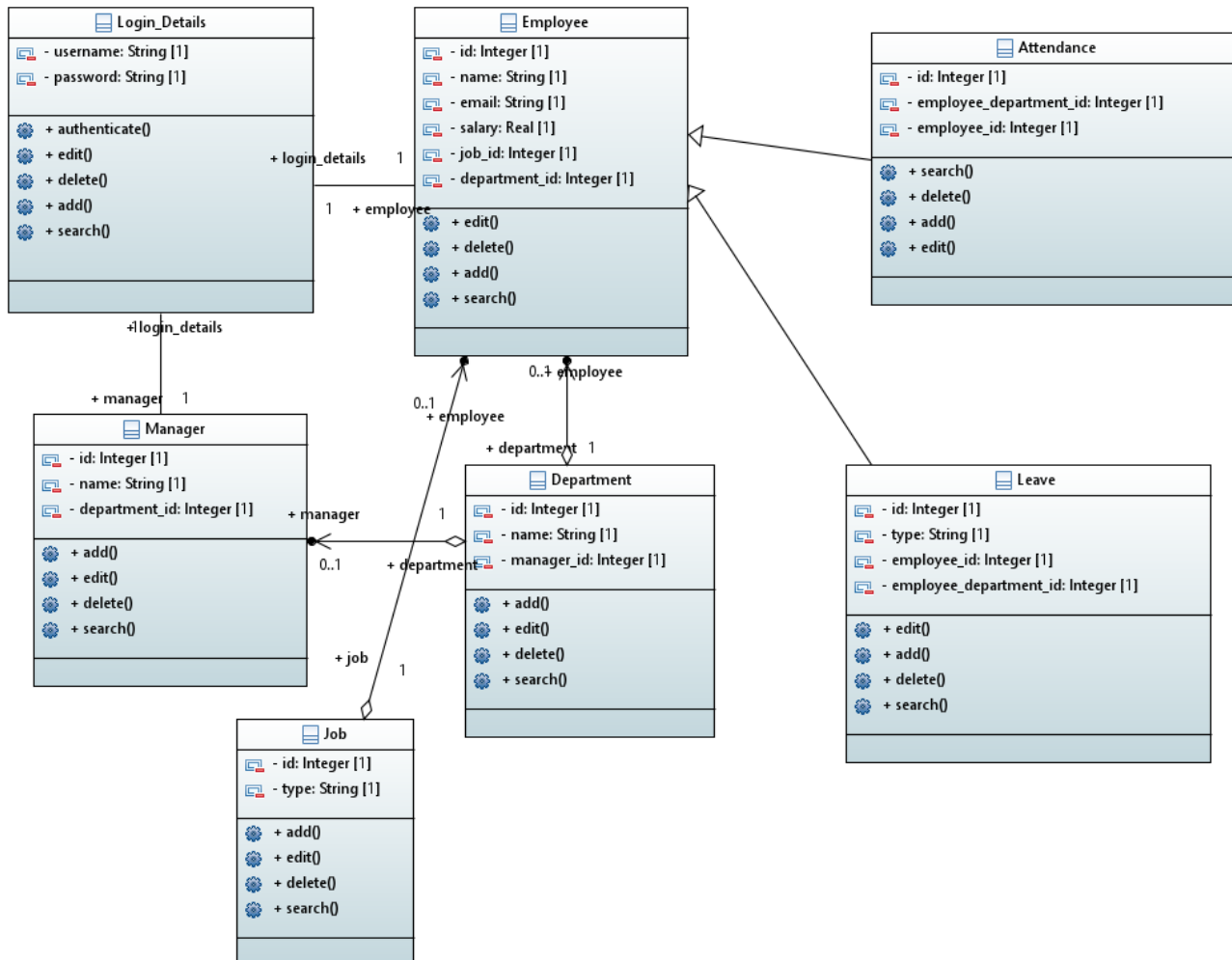
### Trade-offs:

- The three-tier architecture's modularity comes with increased deployment and maintenance complexity.
- React's requirement for a modern browser may restrict compatibility for users with outdated browsers.

## 8 Detailed System Design

### 8.1 Database Design

#### 8.1.1 Class Diagram



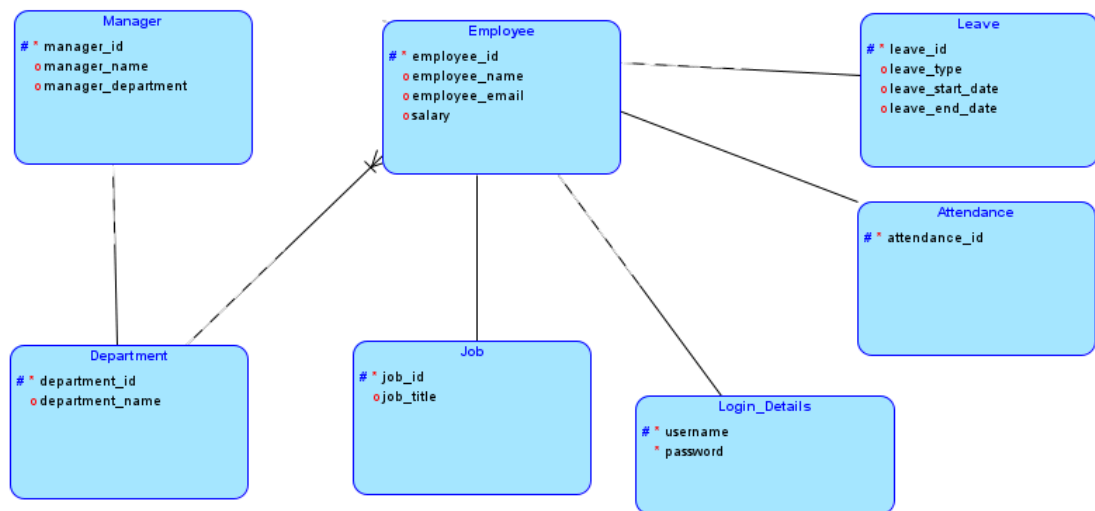
Here's a detailed description of the classes and their attributes, methods, and interactions:

1. **Employee**: This class represents an employee in the company. It has attributes such as `employeeID`, `name`, `Salary`, `departmentID`, `jobID`, and methods like `Markarrendance()`, `updateEmployee()`, `ApplyLeave()`. These methods are used to manage the employee data.
2. **Manager**: This class could represent a manager in the company. It is an instance of the **Employee** class. It has additional methods like `manageattendance()`, `manageleave()`, `manageDepartment()`, `addEmployee()`.
3. **Department**: This class represents the various departments within the company. It has attributes like `departmentID`, `departmentName`, and methods like `addDepartment()`, `deleteDepartment()`.
4. **Attendance**: This class represents the attendance of employees. It has attributes like `id`, `intime`, `date`, `status` and methods like `addAttendance()`, `deleteAttendance()`.
5. **Attendance**: This class represents the leaves of employees. It has attributes like `leaveid`, `Startdate`, `Enddate`, `reason` and methods like `addLeave()`, `deleteLeave()`.

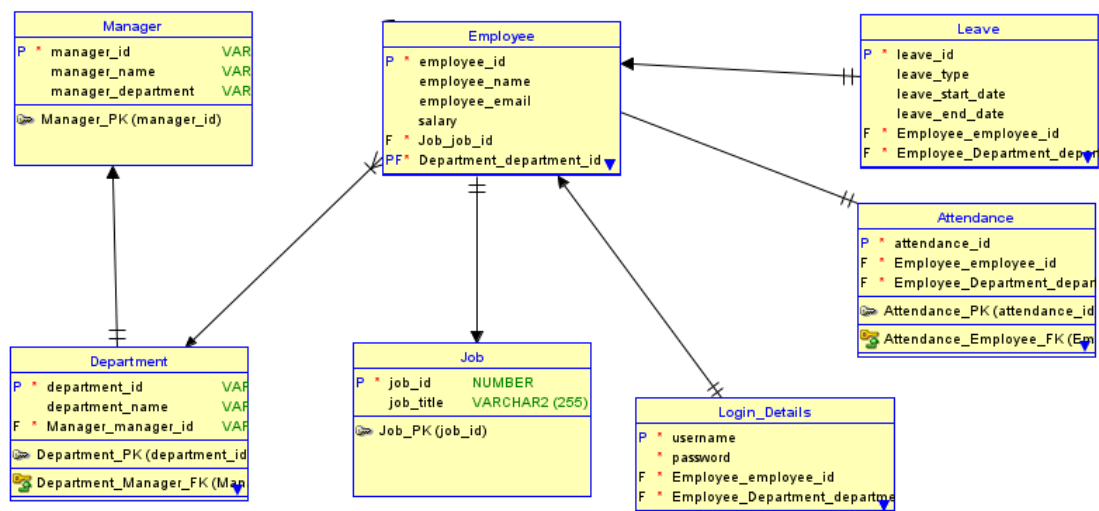
In the system's class diagram, the lines symbolize the associations between different classes. These connections illustrate how entities within the system interact with one another. For instance:

- An Employee is linked to a Department, indicating that each employee belongs to a specific department. This relationship is depicted by a line connecting the Employee and Department classes.
- A Manager has the authority to edit the attendance of an Employee, demonstrating the managerial role in attendance management. This is represented by a line that connects the Manager and Attendance classes to the Employee class.

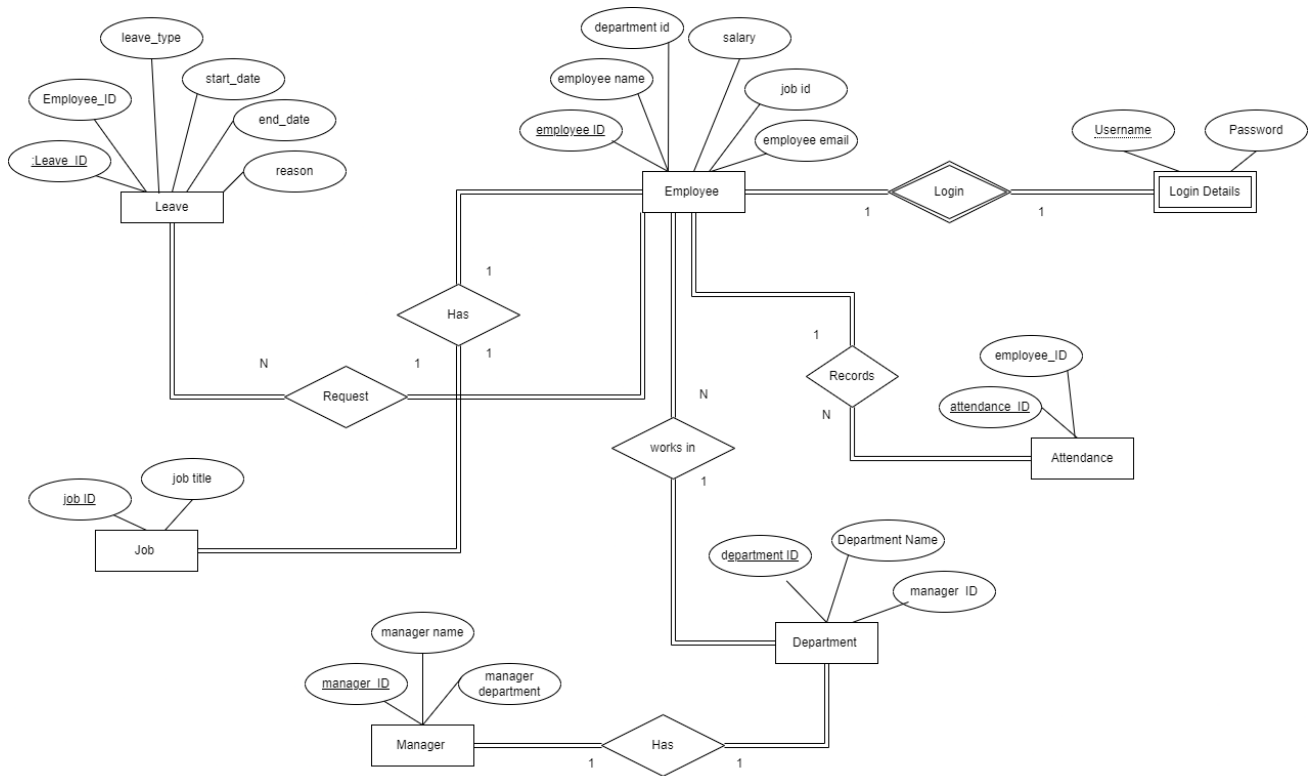
These relationships are crucial for understanding the system's structure and the flow of information between its components. They ensure that the system accurately reflects the organizational hierarchy and the distribution of responsibilities. **Logical Model:**



## Relational Model:



### 8.1.2 ER Diagram



#### Employee Attendance Management System Entities and Relationships

##### 1. Employee Entity:

- Represents an individual employee within the system.
- Attributes include:
  - Name: Full name of the employee.
  - Job: Designation or role within the company.
  - Department: Specific department the employee is part of.
  - Manager: Direct supervisor, potentially linked to another Employee entity.

##### 2. Attendance Entity:

- Denotes the Attendances of all the employees.
- Attributes encompass:
  - id: Refer to employee id from employees.
  - intime: Time of attendance.
  - Date: Date of a particular attendance.
  - Status: refers to the status of employee present absent or leave (e.g. 'A', 'P', 'L').

##### 3. Relationships and Multiplicity:

- The connections between entities are depicted through lines in the class diagram.
- Multiplicity, or the cardinality of a relationship, is denoted by numerical values or symbols adjacent to the entities. For instance:
  - A “1” near the Manager entity and an “N” near the Employee entity signifies that a single manager may be linked to multiple employees.
  - A “1” near both entities indicates one-to-one relationships.

### 8.1.3 Data Dictionary

**User Information:**

Column	Description	Type	Length	Nullable	Default	Key
id	Unique identifier for each user	Integer		No	Auto	PK
name	Username chosen by the user	Text	50	No		
email	Email address of the user	Text	100	No		
password	Hashed password authentication	Text		No		

**Attendance Information:**

Column	Description	Type	Length	Nullable	Default	Key
employee_id	Foreign key referencing Employees	Integer		No		FK
In_time	Time of marking the attendance	Time	100	No	Current	
Date	Date of attendance	Date		Yes	Current	
status	Status of presence	Char		No	-	

**Manager Information:**

Column	Description	Type	Length	Nullable	Default	Key
manager_id	Unique identifier for each manager	Integer		No	Auto	PK
name	Name of the manager	Text		No		

## 9 Application Design

### 9.1 Sequence Diagram

The sequence diagram illustrates the dynamic interactions among various components of the system, specifically focusing on the user, web page, database, employee dashboard, and manager dashboard. It delineates the distinct capabilities allocated to managers and employees. Detailed explanation:

1. User Interaction with Web Page:

- Initially, the user accesses the web page and inputs their credentials, typically through a login procedure involving a username and password.

2. Redirection to Dashboards:

- Post-authentication, the system identifies the user's role. Managers are directed to the manager dashboard, while other employees see the employee dashboard, indicating tailored interfaces for different user types.

3. Employee Dashboard Functions:

- Within the employee dashboard, users can:
  - View Attendance: Check their attendance records and history.
  - Record Attendance: Mark their arrival and departure times, which are then logged in the database.
  - Request Leave: Submit leave applications for approval, which are also tracked in the database.

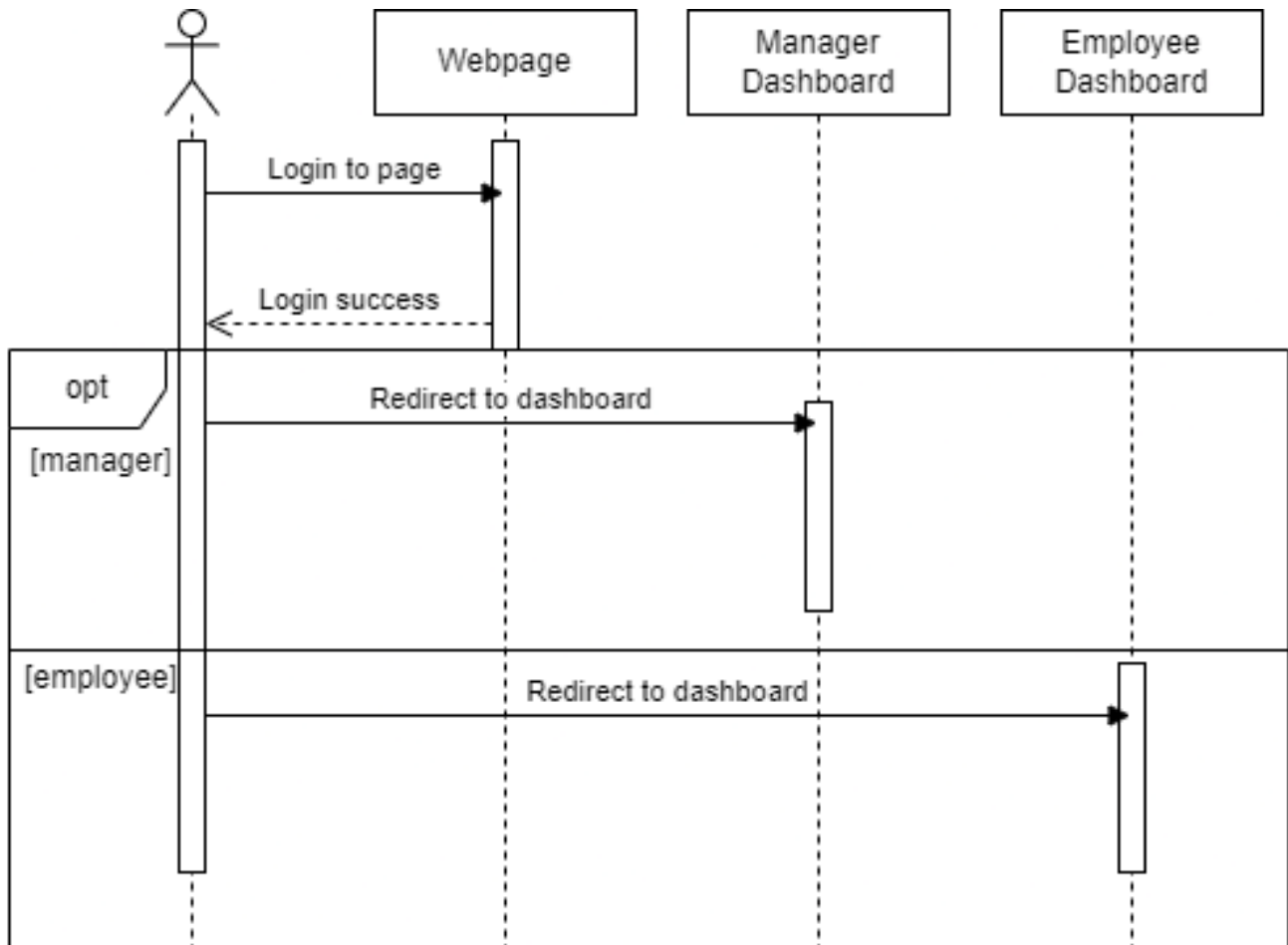
4. Manager Dashboard Functions:

- The manager dashboard allows managers to:
  - View Attendance Records: Oversee the attendance details of all employees.
  - Approve Leave: Review and approve leave requests from employees, with updates reflected in the database.
  - Manage Schedules: Adjust employee schedules and shifts, ensuring accurate attendance tracking.

5. Database Interactions:

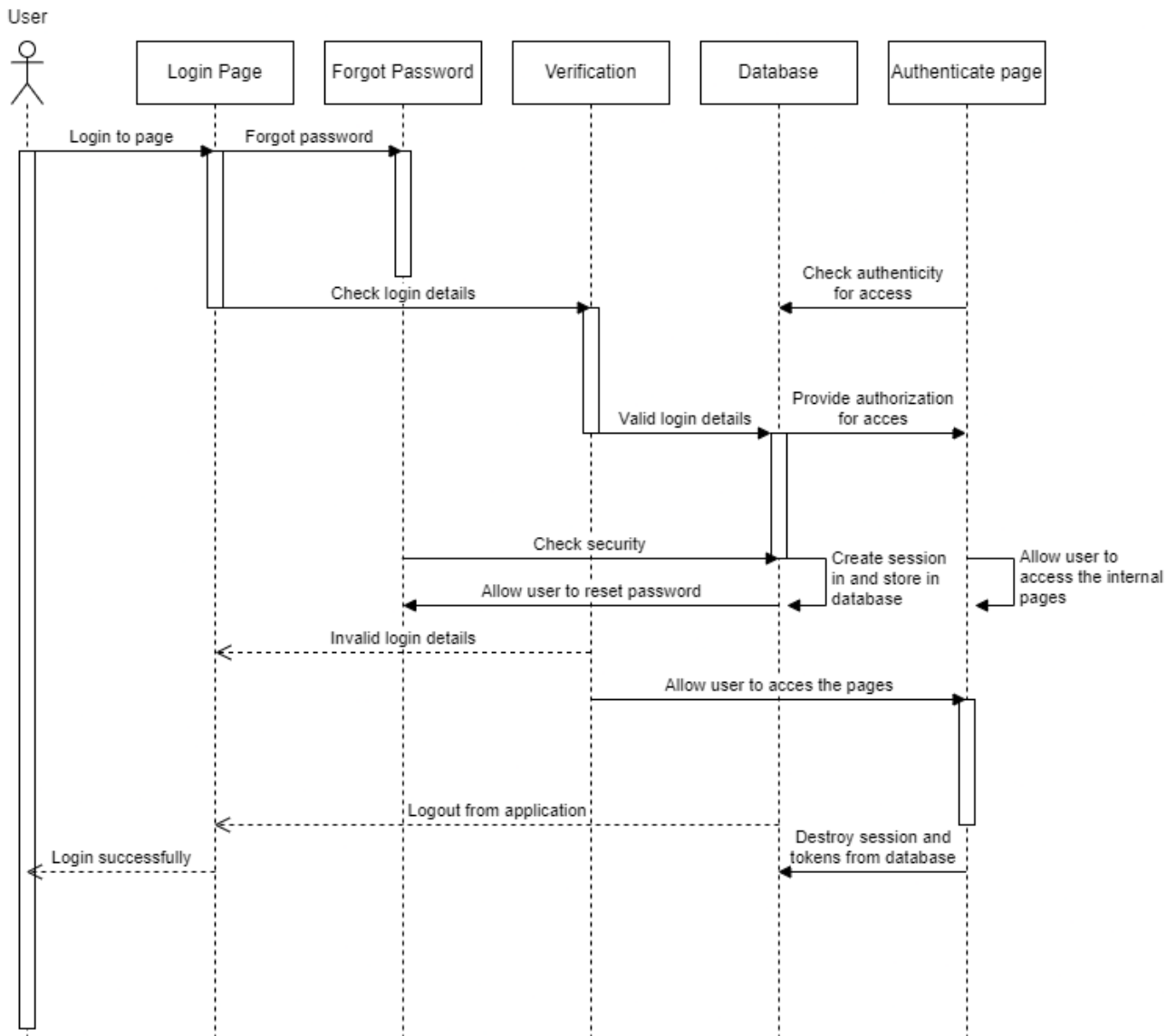
- The database plays a pivotal role in retrieving employee details and logging new attendance. It is updated in real-time as records are added or removed.

### 9.1.1 Attendance Management Sequence



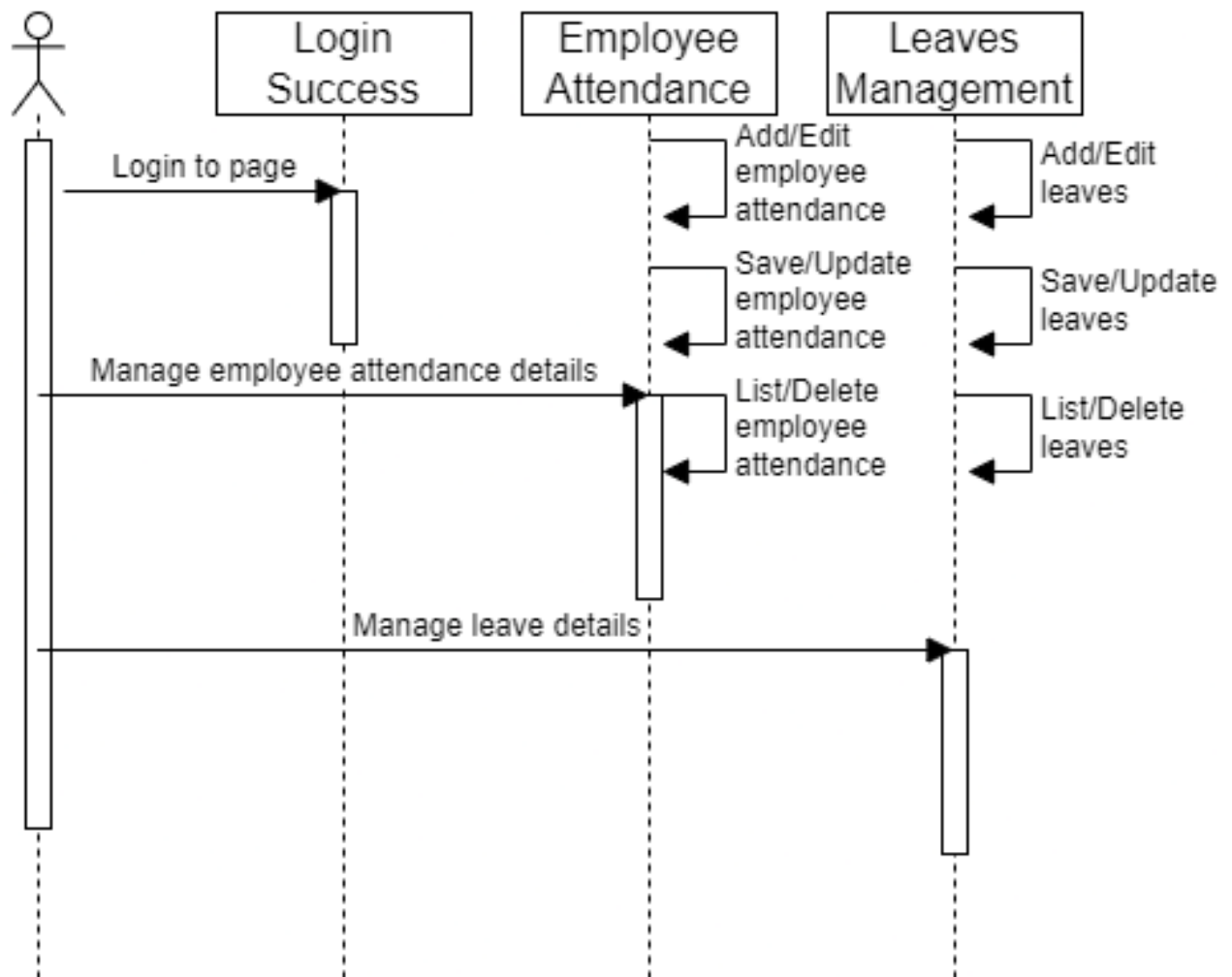


### 9.1.2 Login Sequence

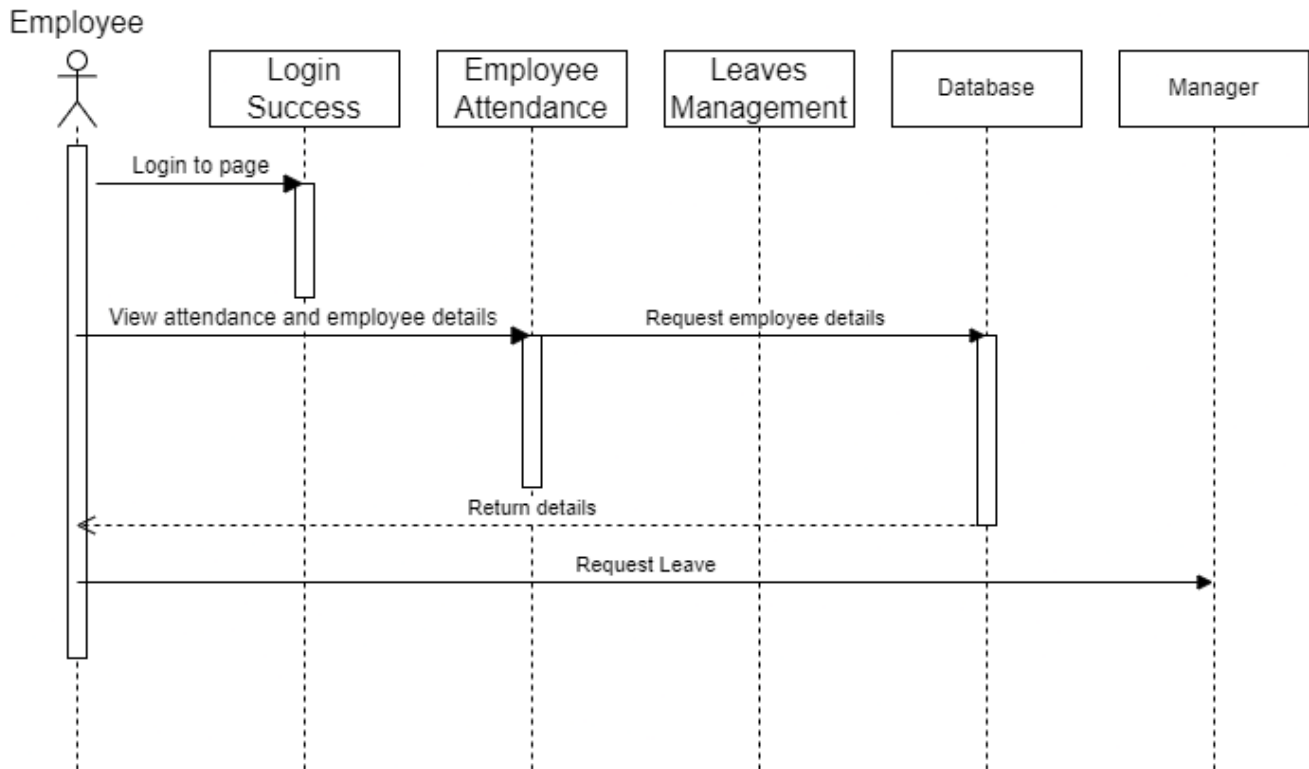


### 9.1.3 Manager Sequence

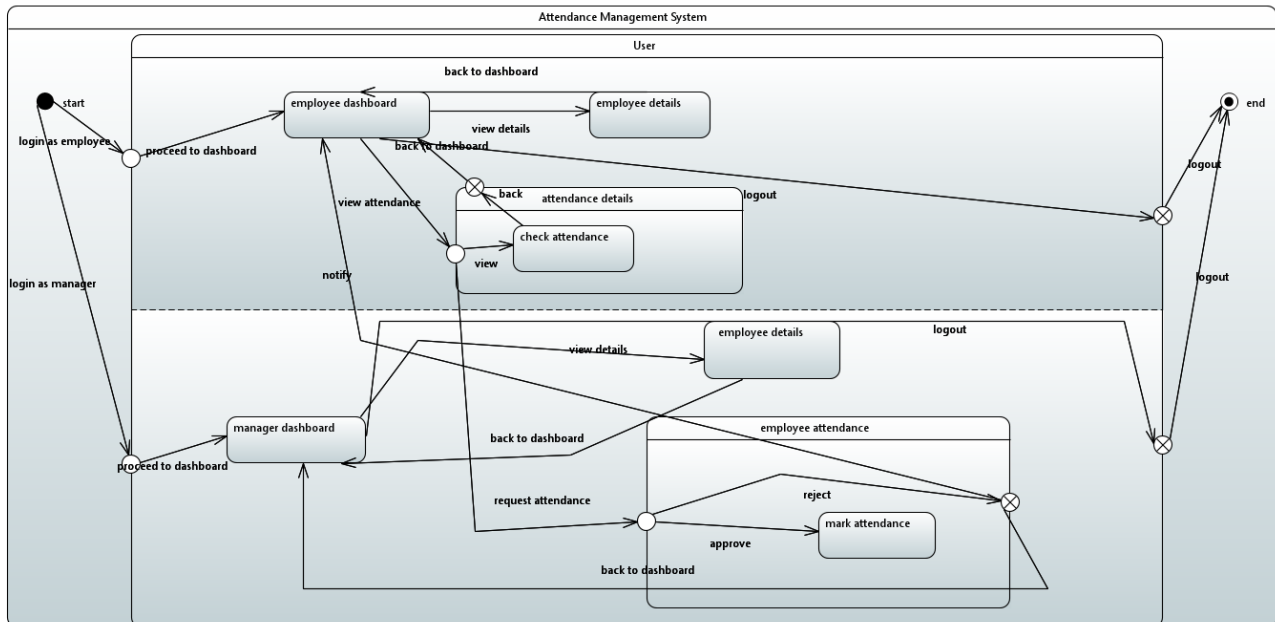
Manager



### 9.1.4 Employee Sequence



## 9.2 State Chart Diagram



1. Employee States: The diagram's upper section outlines the potential states for an employee.

- EmployeeDashboard: The default state where employees can view their attendance records.
- RecordAttendance: In this state, employees log their attendance.
- OnLeave: This state indicates that the employee is on approved leave.
- Present: This state signifies that the employee is marked present for the day.
- Absent: This state indicates an unrecorded or missed attendance.

- LoggedOut: This state occurs when the employee logs out of the system.
2. Manager States: The lower section of the diagram depicts the states a manager can experience.
    - ManagerDashboard: The initial state where managers can oversee attendance and leave requests.
    - ApproveLeave: In this state, managers approve or reject leave applications.
    - ReviewAttendance: Managers enter this state to review and manage attendance discrepancies.
    - CompletedReview: This state signifies the completion of attendance review.
    - TerminatedReview: This state occurs if a review process is canceled or terminated.
    - LoggedOut: This state represents the manager logging out of the system.
  3. Transitions: Arrows illustrate the state transitions triggered by specific actions. For instance, an employee moves from the “EmployeeDashboard” to “RecordAttendance” when they log their arrival. Similarly, a manager transitions from “ManagerDashboard” to “ApproveLeave” upon initiating the leave approval process.

### 9.3 Activity Diagram

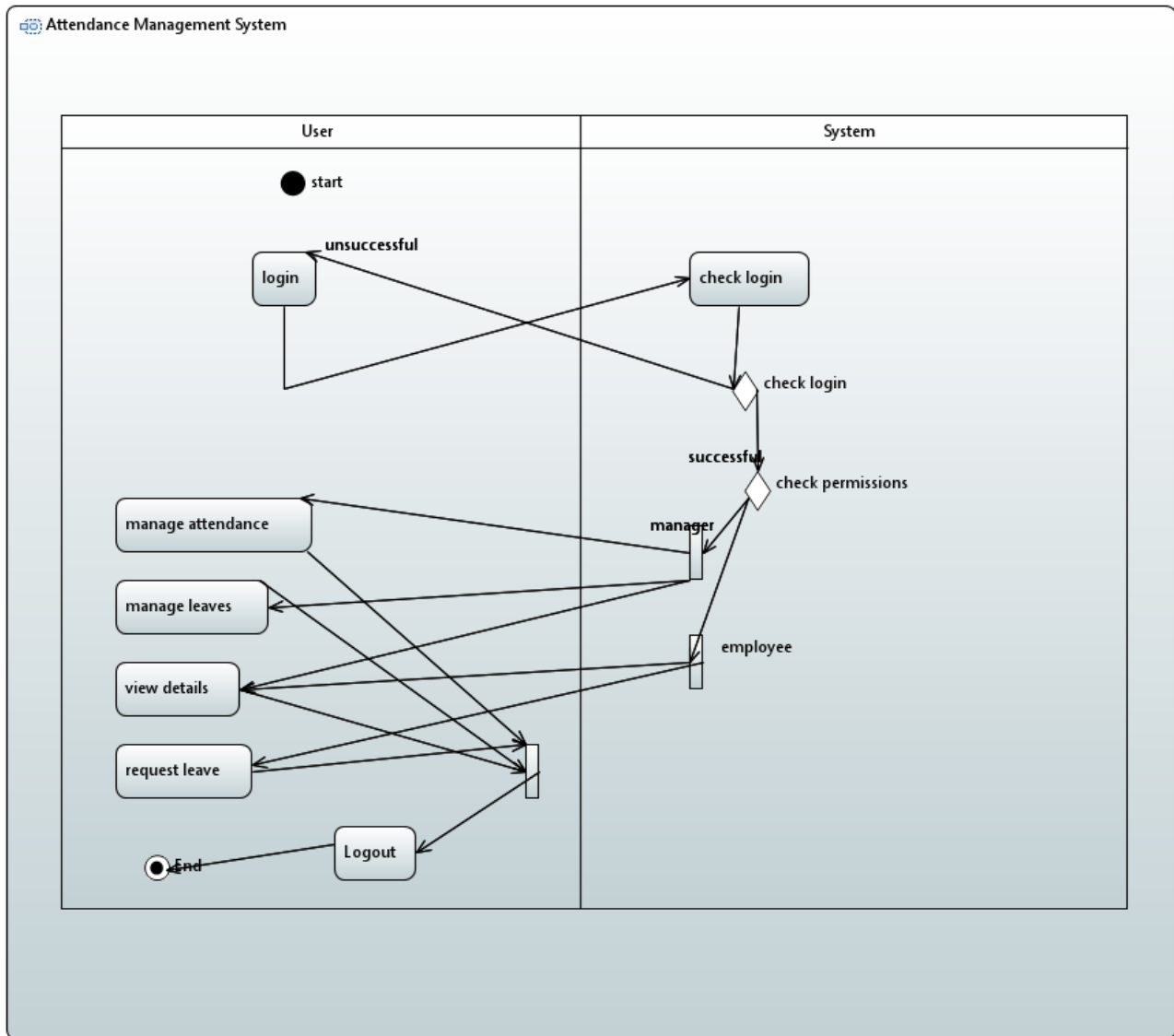
#### Employee Activity:

1. Employee Activity: The process begins with the employee logging into the system.
2. Web Page: Upon successful login, the employee is navigated to the dashboard where they can record attendance, view attendance history, and view personal details.
3. Database: This stage involves the verification of the employee’s credentials during login. If the credentials are validated, access to the dashboard is granted. Otherwise, the employee is prompted to retry the login.
4. Flow of Activities: The arrows in the diagram indicate the sequence of actions. For instance, post-login, the employee may opt to mark attendance, review their attendance records, or request leave.
5. Decision Points: Diamond shapes in the diagram depict decision-making junctures. For example, following a login attempt, the system assesses the authenticity of the credentials. If they are accurate, the employee proceeds to the dashboard; if not, a login retry is required.

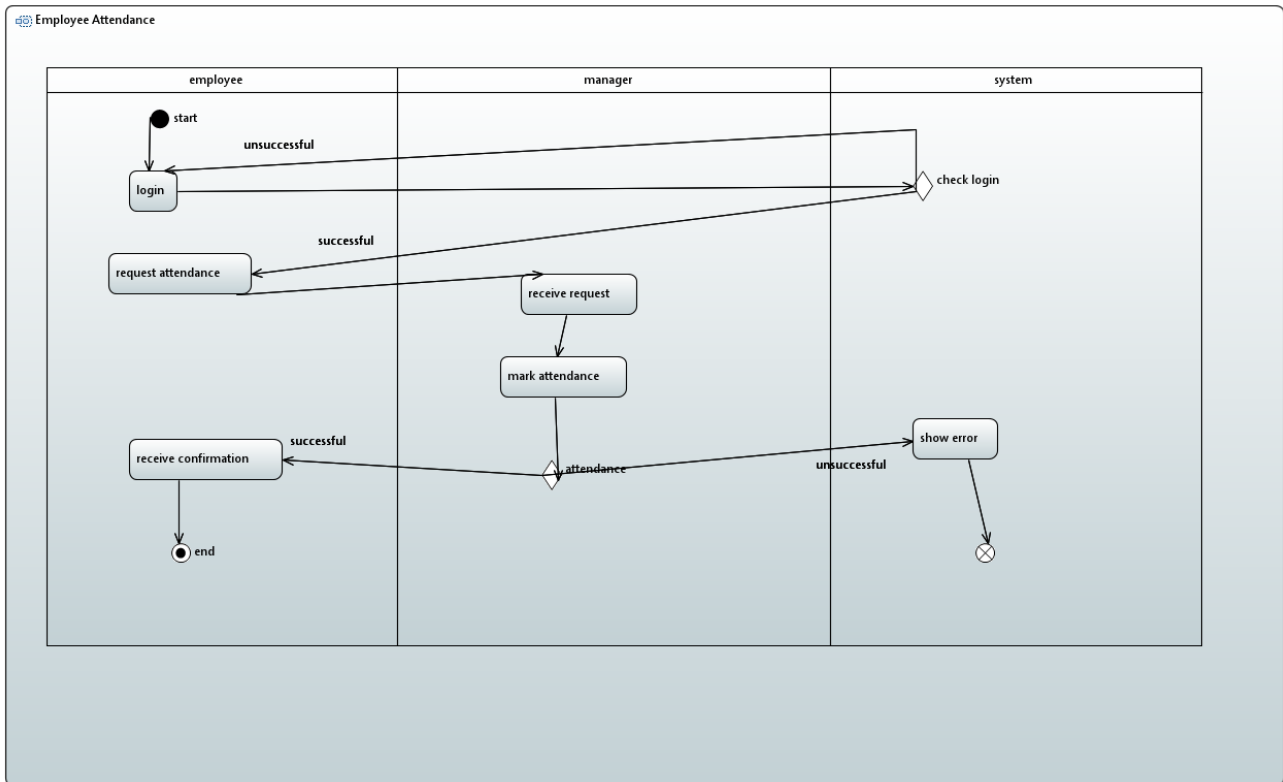
#### Manager Activity:

1. Manager Dashboard: The process initiates with the manager logging into the system.
2. Web Page: Following a successful login, the manager is taken to the dashboard where they can add, view, and remove employee records.
3. Database: This stage pertains to the authentication of the manager’s credentials during login. Correct credentials lead to dashboard access, while incorrect ones result in redirection to the login page.
4. Flow of Activities: The arrows signify the sequence of actions. For instance, after logging in, the manager may decide to register new employees, inspect attendance records, or manage employee details.
5. Decision Points: Diamond shapes indicate critical decision points. For example, post-login attempt, the system evaluates the credentials’ validity. If they are verified, the manager accesses the dashboard; if not, they are prompted to log in again.

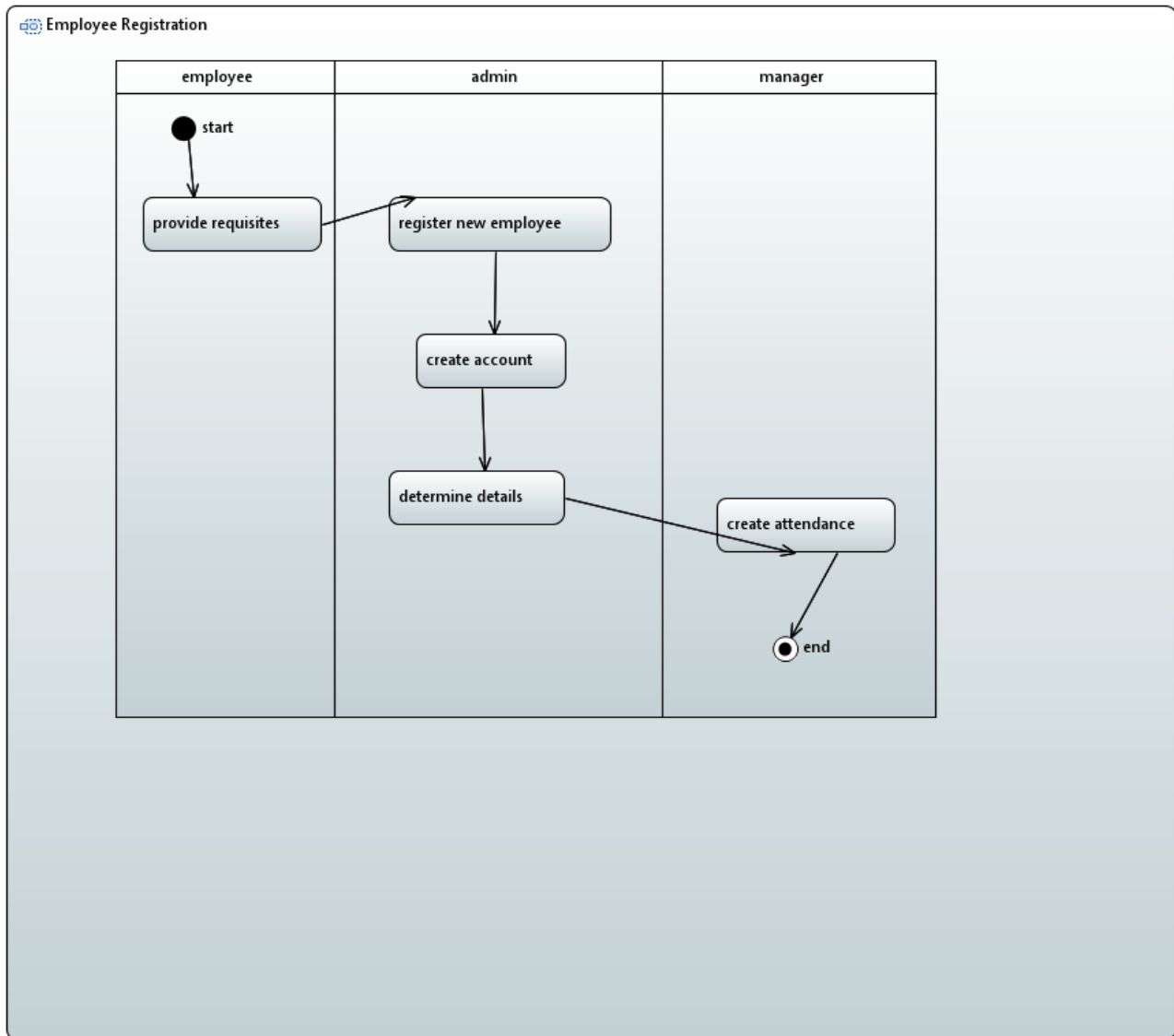
### 9.3.1 Attendance Management System Activity Diagram



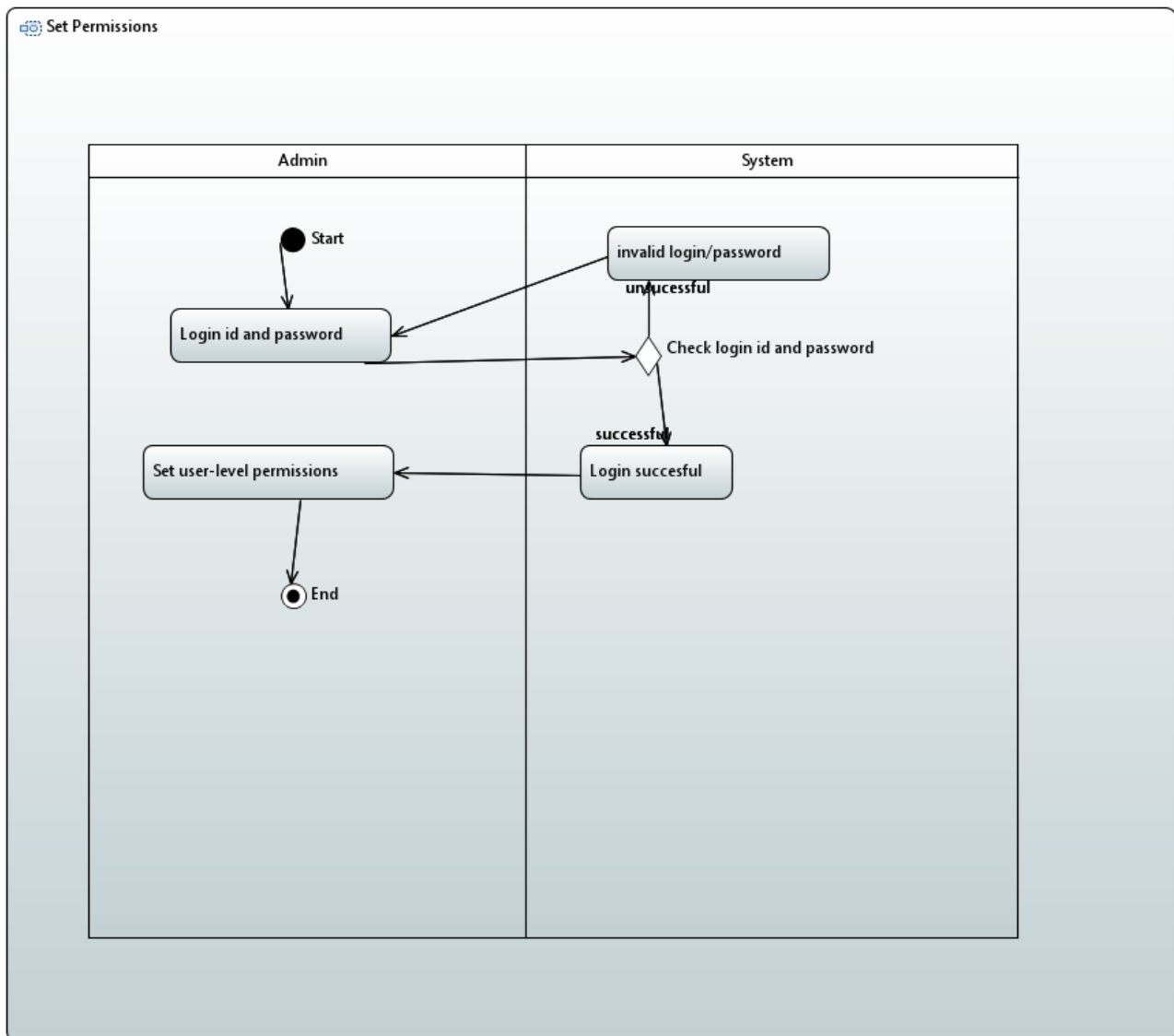
### 9.3.2 Employee Attendance Activity Diagram



### 9.3.3 Employee Registration Activity Diagram

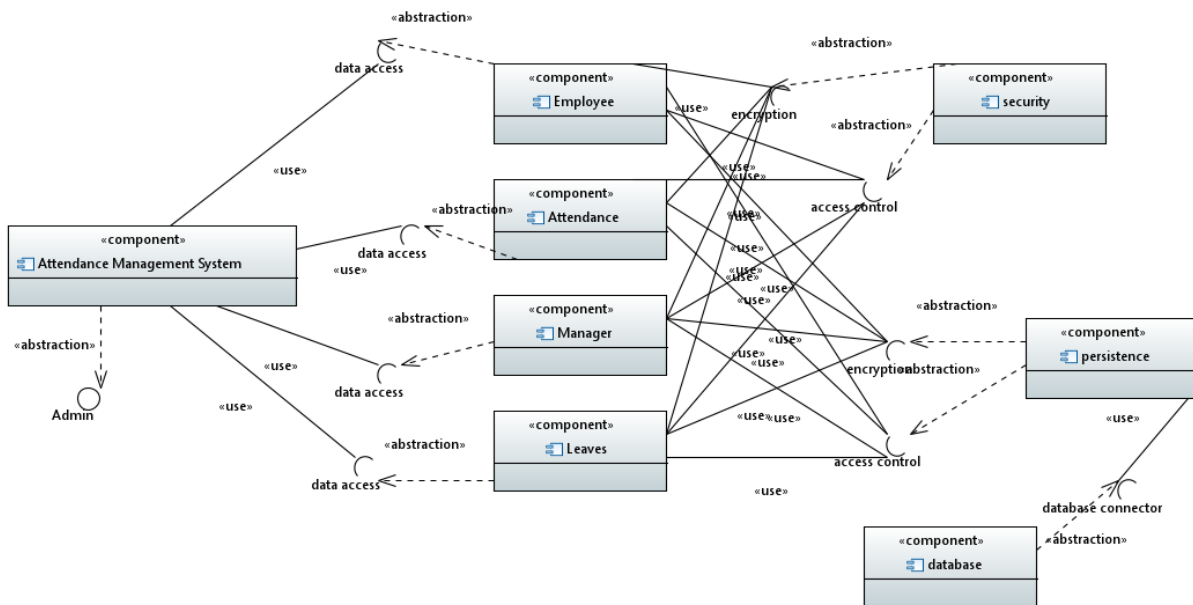


### 9.3.4 Set Permissions Activity Diagram





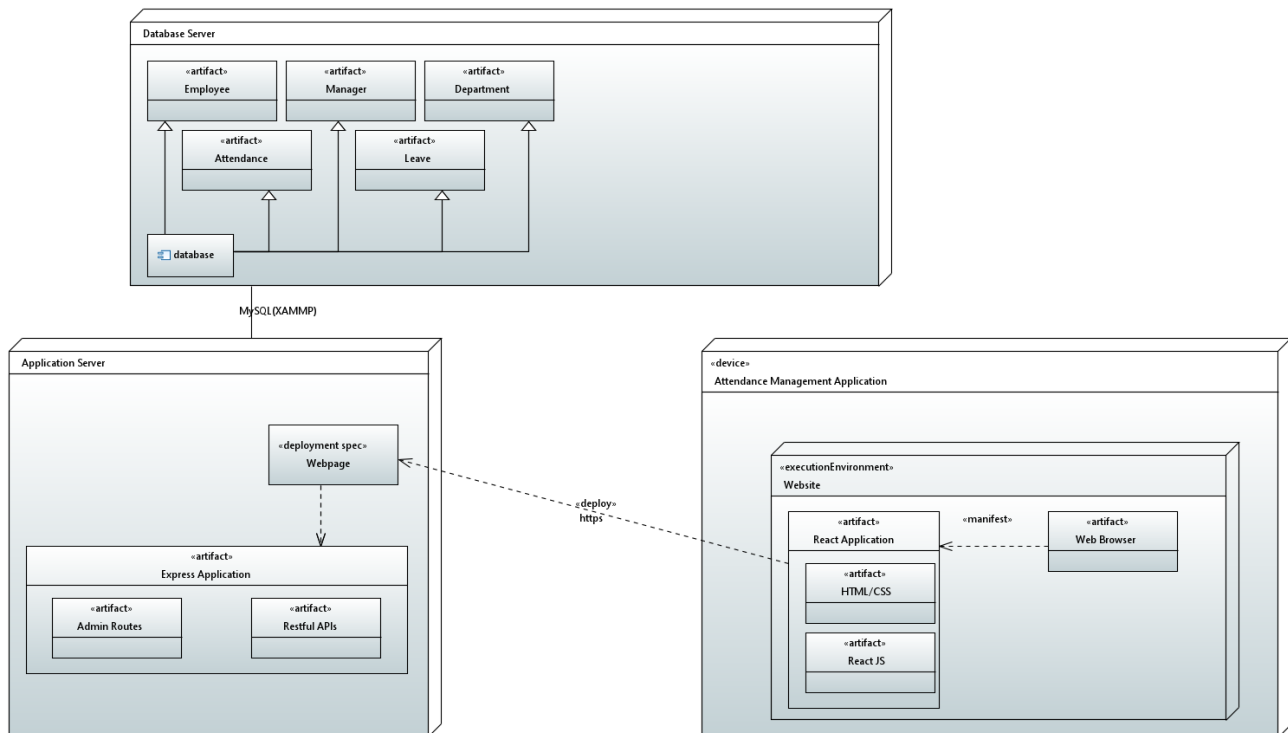
## 9.4 Component Diagram



### Breakdown of Component Diagram:

1. **System Components:** The diagram outlines the core system components.
  - **User Interface:** The primary interface for user interaction with the attendance management application.
  - **View Controller:** This component manages the display content on the user interface.
  - **Model:** Represents the attendance data and the business logic for data access and updates.
  - **Database:** The storage system for all attendance-related data.
2. **User Interface Components:** The diagram details the user interface components.
  - **Login:** Manages the authentication process for users.
  - **Attendance Dashboard:** The central hub for users to manage attendance records.
3. **Relationships:** The connecting lines in the diagram depict the inter-component relationships. The arrow direction shows dependency, such as the User Interface relying on the View Controller for display logic.

## 9.5 Deployment Diagram



Breakdown of the Deployment Diagram:

1. Client Device: Represents the user's device, such as a web browser or mobile app, where the application's user interface is accessed.
2. Web Server: Hosts the Express and Node.js application.
  - Attendance Module: Manages the recording, viewing, and updating of attendance records.
  - Leave Management Module: Handles leave requests and approvals.
3. Database Server: The MySQL database server.
  - Employee Table: Stores data related to employee profiles and attendance.
  - Leave Table: Maintains records of leave applications and statuses.

## 10 References & Citations

- MySQL Documentation:  
*Source : MySQL Official Documentation*
- Express Documentation:  
Source: Express Official Documentation
- React Documentation:  
Source: React Official Documentation:
- Node.js Documentation:  
Source: Node.js Official Documentation
- HTML/CSS Documentation:  
Source: MDN Web Docs
- JavaScript Documentation:  
Source: MDN Web Docs JavaScript

## 11 Appendices

### 11.1 Project Timeline

#### 11.1.1 Week 1

Tasks:

- Research and Planning:
  - Define detailed project requirements for attendance tracking.
  - Finalize the user stories and features for attendance management.
- Environment Setup:
  - Install and configure development tools (Node.js, MySQL).
  - Set up the project structure for the attendance system.

Updates:

- Research completed, and project requirements for attendance management documented.
- Development environment for attendance system configured successfully.

#### 11.1.2 Week 2

Tasks:

- Database Design:
  - Create the Entity-Relationship Diagram (ERD) for attendance data.
  - Set up the MySQL database with tables for employee attendance records.
- Backend Development:
  - Develop the Express.js backend server for attendance operations.
  - Implement database operations for attendance CRUD functionality.

Updates:

- ERD for attendance management completed and database structure defined.
- Basic backend functionality for attendance operations implemented.

#### 11.1.3 Week 3

Tasks:

- Frontend Development:
  - Set up the React.js frontend application for attendance interface.
  - Create components for recording, viewing, updating, and deleting attendance records.
- Integration:
  - Connect the frontend and backend to ensure seamless data flow for attendance management.

Updates:

- Basic frontend structure for attendance management in place.
- Initial integration between frontend and backend for attendance system achieved.

**11.1.4 Week 4**

## Tasks:

- User Interface Refinement:
  - Improve the UI for a better user experience in attendance tracking.
  - Implement features for attendance reporting, leave requests, and employee scheduling.
- Testing:
  - Begin testing phases (unit testing and integration testing) for attendance system.

## Updates:

- UI enhancements for attendance management completed.
- Initial testing phase for attendance system started.

**11.1.5 Week 5**

## Tasks:

- Testing and Debugging:
  - Conduct thorough testing of the entire attendance application.
  - Address and resolve any bugs or issues identified in the attendance system.
- Documentation:
  - Create user manuals and documentation for the attendance management system.

## Updates:

- Testing phase for attendance system ongoing, with identified bugs being addressed.
- Initial draft of user manuals for attendance system completed.