SYSTEM DESIGN DOCUMENT
Leave Management System

Prepared by: Akshit Arora
Technology Stack: Angular | ASP.NET Core | SQL Server

1. Introduction

The Leave Management System (LMS) is a web-based enterprise application designed to automate the process of leave application, approval, tracking, and management within an organization. The system replaces traditional manual workflows with a secure, scalable, and centralized digital solution.

2. Purpose of the Document

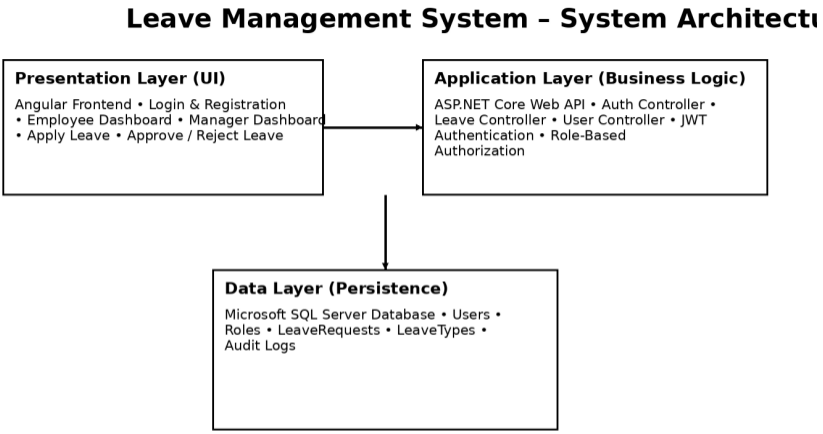
This System Design Document provides a detailed technical description of the architecture, components, data models, interfaces, and workflows of the Leave Management System. It is intended for technical reviewers, architects, and stakeholders in an enterprise environment.

3. System Overview

The Leave Management System follows a three-tier architecture consisting of Presentation, Application, and Data layers. The frontend is developed using Angular, the backend uses ASP.NET Core Web API, and data persistence is handled by Microsoft SQL Server.

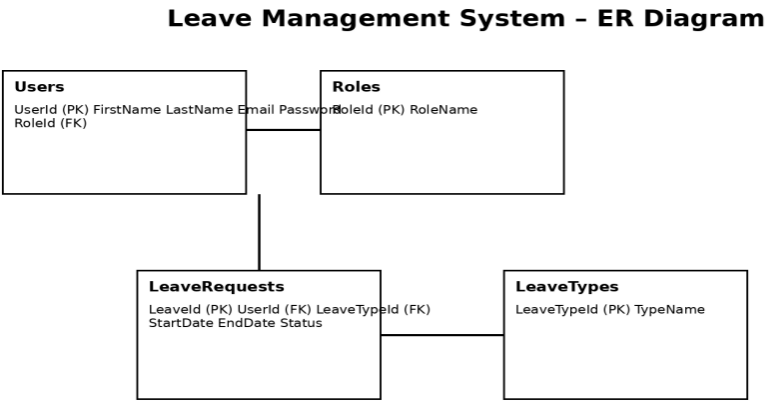
4. System Architecture

Figure below illustrates the high-level system architecture of the Leave Management System.



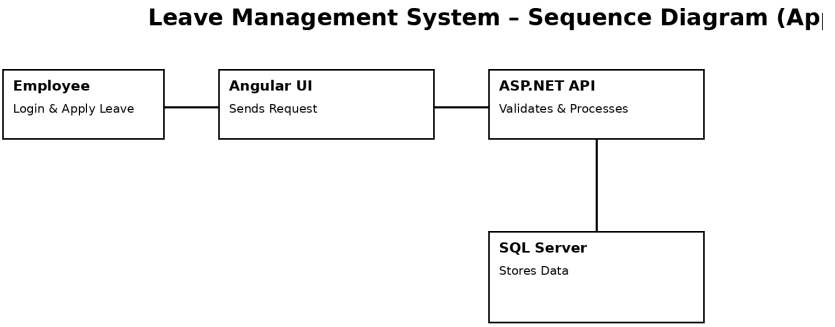
5. Database Design – ER Diagram

The ER diagram represents the relational database structure and entity relationships.



6. Sequence Diagram – Apply Leave

This sequence diagram demonstrates the interaction flow when an employee applies for leave, covering UI, backend processing, and database persistence.



7. Functional Modules

Frontend Modules:

- Authentication Module (Login & Registration)
- Employee Module (Apply Leave, View Status)
- Manager Module (Approve/Reject Leave)
- Dashboard Module (Role-based views)

Backend Modules:

- Authentication & Authorization
- Leave Management
- User & Role Management

8. API Design

POST /api/auth/register – Register user
POST /api/auth/login – Authenticate user
POST /api/leave/apply – Apply leave
GET /api/leave/user – Fetch employee leaves

GET /api/leave/pending – Fetch pending approvals
PUT /api/leave/approve – Approve/Reject leave

9. Security Architecture

• JWT-based authentication
• Role-based authorization
• HTTPS communication
• Server-side validation
• Secure password hashing

10. Non-Functional Requirements

Performance: Supports concurrent users with low latency
Scalability: Modular architecture allows horizontal scaling
Reliability: Ensures data consistency and fault tolerance
Security: Protects against unauthorized access

11. Deployment Architecture

The system can be deployed on cloud or on-premise infrastructure. The frontend is hosted on a web server, backend APIs on an application server, and SQL Server runs on a secure database server.

12. Conclusion

The Leave Management System is designed following enterprise software engineering principles. Its modular architecture, secure design, and scalable structure make it suitable for real-world enterprise usage.