**CS673 Software Engineering** 

**Team 2 - DAMPP**

**Software Design Document**

| Team Member | Role(s) | Signature | Date |
| --- | --- | --- | --- |
| Dipayan Mazumder | Security Lead | *DM* | 05-10-2024 |
| Pranjal Ekhande | Requirement Lead | *PE* | 05-10-2024 |
| Aman Jain | Configuration Lead | *AJ* | 05-10-2024 |
| Mukul Jangid | Design and Implementation lead | *MJ* | 05-10-2024 |
| Praveen Singh | Team Lead | *PS* | 05-10-2024 |

**Revision history**

| **Version** | **Author** | **Date** | **Change** |
| --- | --- | --- | --- |
| 1.0 | Dipayan Mazumder, Pranjal Ekhande | 05-10-2024 | First Draft |
|  |  |  |  |

1.0 [Introduction](#_heading=h.gjdgxs)

2.0 [Software Architecture](#_heading=h.30j0zll)

3.0 [Class Diagram](#_heading=h.1fob9te)

4.0 [UI Design (if applicable)](#_heading=h.3znysh7)

5.0 [Database Design (if applicable)](#_heading=h.2et92p0)

6.0 [Security Design](#_heading=h.tyjcwt)

7.0 [Business Logic and/or Key Algorithms](#_heading=h.3dy6vkm)

8.0 [Design Patterns](#_heading=h.1t3h5sf)

9.0 [Any Additional Topics you would like to include.](#_heading=h.4d34og8)

10.0 [References](#_heading=h.2s8eyo1)

11.0 [Glossary](#_heading=h.17dp8vu)

**Introduction**

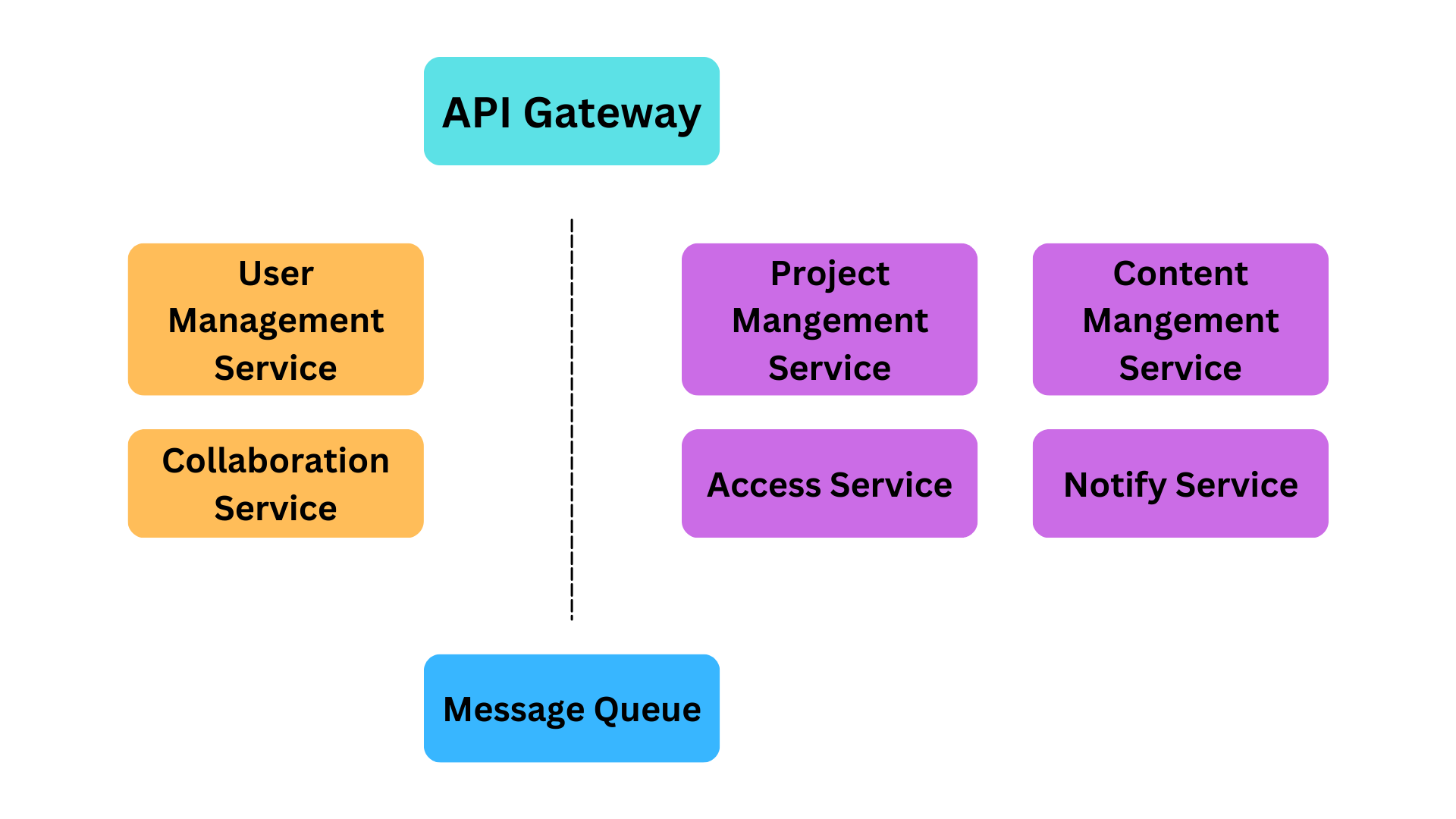
The Project Management Service is a crucial component of the project-based learning platform, designed to facilitate the creation, management, and tracking of projects for users. This service will provide a comprehensive set of features to support the entire project lifecycle, from project initiation to completion.

The key goals of the Project Management Service are:

1. Enable users to create and manage projects, including defining project details, setting milestones, and assigning tasks to team members.
2. Provide a centralized platform for tracking project progress, monitoring task completion, and generating reports and analytics.
3. Integrate seamlessly with other services in the platform, such as User Management, Collaboration, and Notification services, to provide a cohesive and collaborative learning experience.
4. Implement robust security measures, including authentication, authorization, and data protection, to ensure the confidentiality and integrity of project data.

**Software Architecture**

The Project Management Service will be implemented as a microservice within the project-based learning platform's microservice architecture. It will communicate with other services using RESTful APIs and follow the principles of loose coupling and high cohesion. The service will be designed with a layered architecture, separating concerns and responsibilities into distinct layers, such as the Presentation Layer (API controllers), Business Logic Layer (service classes), and Data Access Layer (repositories). An API Gateway will act as a single entry point for client requests, routing them to the appropriate services and handling cross-cutting concerns like authentication, rate limiting, and caching. The following diagram illustrates the high-level software architecture of the project-based learning platform, highlighting the Project Management Service and its interactions with other services:

​​  


# Class Diagram

The Project Management Service will consist of the following main classes and their relationships:

**Project**: Represents a project and its associated data, such as name, description, start and end dates, and status.

**Task**: Represents a task within a project, including details like description, due date, assignee, and status.

**User**: Represents a user of the platform, including their roles and permissions.

**ProjectMemberRole**: Defines the roles and permissions for project members (e.g., owner, contributor, viewer).

**ProjectTemplate**: Represents a predefined template for creating projects with a specific structure and set of tasks.

**ProjectReport**: Handles generating various reports and analytics for projects, such as progress reports, task completion reports, and team performance reports.

# UI Design (Pending)

In this section, you can describe your UI design. You can include both your initial design before the implementation and the screenshots of your UI after the implementation.

# Database Design

The Project Management Service will utilize a relational database to store project-related data. The following database schema illustrates the main tables and their relationships:

**Projects**: Stores project details like name, description, start and end dates, and status.

**Tasks**: Stores task details like description, due date, status, and assignee.

**Users**: Stores user information like name, email, and password (hashed).

**ProjectMembers**: Stores project member assignments and their roles.

**ProjectTemplates**: Stores predefined project templates and their associated tasks. The `ProjectMembers` table acts as a join table, connecting projects and users while also storing the assigned role for each member.

# Security Design

The Project Management Service will implement the following security measures:

**Authentication and Authorization**: Integration with the User Management Service for user authentication and role-based access control (RBAC). JSON Web Tokens (JWT) will be used for stateless authentication and authorization.

**Data Encryption**: Sensitive data, such as user passwords and project data, will be encrypted using industry-standard encryption algorithms like AES-256.

**Input Validation**: All user input will be validated and sanitized to prevent injection attacks like SQL injection and Cross-Site Scripting (XSS).

**Secure Communication**: Communication between the Project Management Service and other services will be secured using HTTPS and encrypted using TLS/SSL.

**Audit Logging**: Audit logs will be maintained to track user actions, including project creation, updates, and deletions, for auditing and forensic purposes.

**Secure Coding Practices**: The service will be developed following secure coding best practices, such as the OWASP Top 10 security guidelines, to mitigate common web application vulnerabilities.

# Business Logic and/or Key Algorithms

The Project Management Service will implement the following key algorithms and workflows:

**Project Creation and Initialization**: This workflow will handle the creation of new projects, including validating user input, initializing project details, and setting up project members and their roles.

**Task Scheduling and Dependency Management**: An algorithm will be implemented to handle task scheduling, taking into account task dependencies, project milestones, and resource availability.

**Project Progress Tracking**: A progress tracking algorithm will calculate the overall project progress based on completed tasks, task dependencies, and project milestones. **Project Reporting and Analytics**: Algorithms will be implemented to generate various reports and analytics, such as project progress reports, task completion reports, and team performance reports.

# Design Patterns

In this section, you shall describe any design patterns used in your software system.

The Project Management Service will leverage the following design patterns: 1. **Repository Pattern**: Used to abstract the data access layer and provide a centralized way to interact with the database.

**Observer Pattern**: Implemented to notify interested parties (e.g., project members, Notification Service) when project or task updates occur.

**Strategy Pattern**: Used to implement different task scheduling and reporting strategies, allowing for easy extensibility and customization.

**Decorator Pattern**: Employed to extend the functionality of project or task objects by adding additional behaviors or responsibilities.

**Facade Pattern**: Used to provide a simplified interface for complex subsystems, such as project reporting or task scheduling.

# Integration Considerations

The Project Management Service will integrate with the following services in the project-based learning platform:

**User Management Service**: For user authentication, authorization, and retrieval of user information.

**Collaboration Service**: To enable real-time communication, document sharing, and discussions within the project teams.

**Notification Service**: To send notifications to the project

# Any Additional Topics you would like to include.

N/A

# References

Pending

# Glossary

Pending