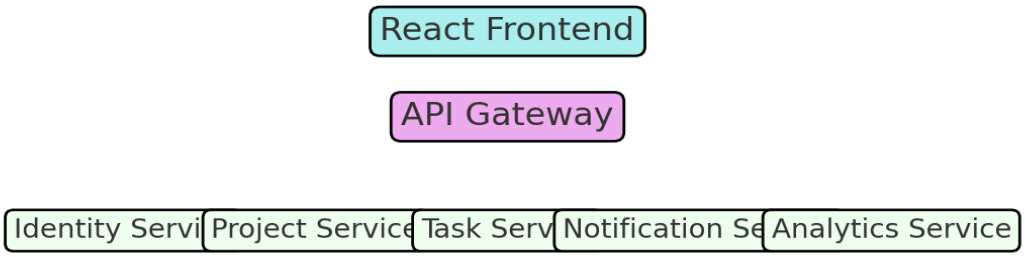
Task Management System (TMS)

Software Design Specification (SDS)

Project Overview: Task Management System (TMS) is a microservices-based web application for managing projects, tasks, notifications, and analytics. The system includes REST and gRPC APIs, a ReactJS frontend, and cloud-based CI/CD.

System Architecture

The system follows a modular microservice architecture. Each service owns its own database and communicates using REST or gRPC.



Microservices Overview

Service	Purpose	Communication
Identity Service	User registration, authentication, JWT handling	REST
Project Service	Manage projects and participants	REST + gRPC
Task Service	Manage tasks, status updates, assignments	REST + gRPC
Notification Service	Email and system notifications	gRPC
Analytics Service	Aggregated reports and statistics	gRPC
API Gateway	Routing, aggregation, and authentication	REST

Database Model (ER Overview)

User

Field	Type	Description
Id	GUID	Unique identifier
UserName	string(50)	User name
Email	string(100)	Email address

PasswordHash	string	Hashed password
Role	enum(User, Manager, Admin)	User role
CreatedAt	datetime	Registration date

Project

Field	Type	Description
Id	GUID	Project ID
Name	string(100)	Project name
Description	string	Project description
OwnerId	GUID	Project creator
CreatedAt	datetime	Creation date
UpdatedAt	datetime	Last update

Task

Field	Type	Description
Id	GUID	Task ID
Title	string(150)	Task title
Description	string	Task description
ProjectId	GUID	Project reference
AssignedUserId	GUID?	Assigned user
Status	enum(Open, InProgress, Done, Archived)	Task status
Priority	enum(Low, Medium, High, Critical)	Task priority
CreatedAt	datetime	Creation date
UpdatedAt	datetime	Last update

Technical Stack

Backend: ASP.NET Core 8, EF Core, Dapper, PostgreSQL, Serilog, FluentValidation **Frontend:** React 18, TypeScript, MUI, React Query, Vite **Communication:** REST + gRPC **CI/CD:** GitHub Actions / Azure DevOps / AWS CodePipeline **Documentation:** Swagger, gRPC-doc, README.md, Postman collection

Development Plan

1. Initialize repository and base architecture
2. Implement Identity and authentication
3. Build Project and Task services with database
4. Add gRPC communication and Notification Service
5. Develop React frontend and integrate API
6. Set up logging, monitoring, and CI/CD pipelines
7. Final documentation and deployment