



Homework 7

PEPE (Project Elaborating Program with Extension)
Proposal

Software Engineering Principle

Software Engineering Program,

Department of Computer Engineering,

School of Engineering, KMITL

67011235 Paphavee Yanmook

67011287 Ramida Laphasphokin

67011352 Theepakorn Phayonrat

PEPE (Project Elaborating Program with Extension)

Requirements

User Requirements

Functional

- The software shall allow users to create, view, edit and delete desired projects.
- The software shall allow users to create tasks, assign them to team members and set deadlines.
- The software shall allow users to update their task status such as in progress, completed, etc.
- The software shall allow users to view project timeline in a Gantt Chart and draw diagrams in canvas.
- The software shall allow users to collaborate and receive notifications for task update.

Non-Functional

- The application shall save the diagrams into *SVG* or *PNG* format and paste into any kind of document file.
- The system shall be accessible through modern web browsers without requiring additional plugins.
- The user interface shall be intuitive and easy to navigate for both managers and team members for collaborative actions, including modifications of the charts and diagrams.
- The system shall ensure data consistency and prevent unauthorized access to project data.
- The application shall maintain availability during concurrent usage by multiple users.

System Requirements

Functional

- The system shall support storage and retrieval of project metadata and design artifacts (e.g., diagrams, timelines) using JSON fields in PostgreSQL.
- The system shall log application errors, security events, and system activity using Django's logging framework for monitoring and debugging.
- The system shall provide a frontend implemented using React with TypeScript, enabling dynamic rendering of project boards, tasks, and project views without full page reloads.
- The system shall expose RESTful APIs implemented using Django and Django REST Framework (DRF) to handle user authentication, project management, task assignment, and metadata retrieval.
- The system shall support real-time communication using Django Channels and WebSockets, allowing instant synchronization of task updates, assignments, mentions, and notifications across connected clients.

Non-Functional

- The system shall maintain low-latency real-time updates, with WebSocket message propagation handled via Django Channels backed by Redis as a message broker.
- The system shall secure all client-server communication using HTTPS and secure WebSocket protocols, with reverse proxy support provided by Nginx.
- The system shall be scalable to support multiple concurrent users by utilizing ASGI servers (e.g., Uvicorn/Daphne) and stateless backend services.
- The system shall ensure data integrity and transactional consistency through PostgreSQL ACID-compliant transactions.
- The Database shall be normalized to the fifth normal form (5NF) to prevent the INSERT and DELETE problem in the relational database.