CS 319 Object-Oriented Software Engineering Project Deliverable 4 (D4) - S2T7 - agora

Team Members:

- Bertan Uran - 22102541
- Egehan Yıldız - 22203014
- Ekin Köylü - 22103867
- Emre Yazıcıoğlu - 22201668
- İlke Latifoğlu - 22203818
- Merve Güleç - 22103231

CONTENTS

Design Goals

Security

Essential for systems handling sensitive data to prevent breaches and maintain user trust. However, it adds latency and development costs. It is prioritized to avoid reputational damage, legal issues, and user trust loss.

Extensibility

Enables adaptation to future requirements and scalable growth. However, it Increases initial complexity and risks overengineering. It is prioritized to reduce long-term costs and to support evolving needs.

Robustness

Ensures systems handle failures and edge cases without disruption. However, it adds performance overhead and development effort. It is prioritized to minimize downtime, maintaining user satisfaction and continuity.

Traceability of Requirements

Tracks requirements from inception to testing, ensuring alignment and scope management. However, it requires documentation and tooling. It is prioritized to reduce scope creep and to ensure business-technical alignment.

Availability

Critical for minimal downtime in user-facing and mission-critical systems.However, it increases costs and system complexity. It is prioritized to minimize downtime that directly impacts revenue, user experience, and trust.

Connectors

RESTful APIs

It provides structured endpoints for data exchange between the frontend and backend. It is ideal for efficient web-based systems.

Database ORM

It translates code into database queries and simplifies database operations. It ensures consistency and speeds up development.

WebSocket

It provides real-time communication between the client and server and enables updates for live features.

OAuth 2.0 or JWT for Authentication

It ensures secure login and role-based access. It integrates third-party services, while JWT supports scalable, stateless authentication.

Architectural Style

- **Layered Architecture:** It organizes the system into distinct layers which have specific responsibilities:
    - The presentation layer for user interaction.
    - The application layer for business logic.
    - The data layer for data storage and retrieval.

This architecture separates the system into frontend, backend and database with distinct responsibilities. In this way, maintenance, testing, and scaling are simplified.

Subsystem Decomposition Diagram