CS319 Object Oriented Software Engineering

TX-T11 Software Quartet | D4

Domain, Activity, Sequence Diagrams and Mockup UI

Section 1 - Anıl Koyuncu

26/04/2024

Kağan Rehber - 22101777

Furkan Komaç - 22102165

Seçkin Alp Kargı - 22001942

Erkan Can Arslan - 22103948

Design Goals

1) **Performance**

Goal: Achieve quick load times and efficient interaction handling to support academic activities without delays, supporting up to 2000 concurrent users.

Tradeoff: Investing in performance optimization may require more sophisticated hardware or cloud services, potentially increasing operational costs.

Reason for Selection: For an educational environment with a high number of users, ensuring responsiveness is crucial to maintaining a productive learning experience.

2) **Security and Privacy**

Goal: Use advanced encryption for data at rest and in transit, with strict access controls and role-based account management.

Tradeoff: High levels of security measures can impact system performance and increase complexity in system maintenance.

Reason for Selection: Protecting student and faculty information is paramount, especially to comply with data protection regulations and to maintain trust in the system.

3) **Scalability**

Goal: Design system for easy horizontal scaling to accommodate growth in user numbers and data volume.

Tradeoff: Scalability often involves more complex architecture and possibly greater upfront investment in scalable technologies.

Reason for Selection: Ensuring the system can handle growth without performance degradation is essential for the anticipated increase in users and data over time.

4) **Reliability**

Goal: Aim for high availability, minimal service disruption, with strategies synchronization and nightly backups.

Tradeoff: Increased reliability can lead to higher costs due to redundant systems and backup mechanisms.

Reason for Selection: Continuous access to educational materials and system functionalities is critical to avoid interrupting the educational process.

5) **User Interface and Human Factors**

Goal: Create an intuitive, accessible interface conforming to WCAG 2.0 guidelines, with responsive design across devices.

Tradeoff: Designing for broad accessibility may require additional resources and extended testing phases to ensure compatibility across all user types and devices.

Reason for Selection: A user-friendly interface ensures that all students, including those with disabilities, can navigate and use the system effectively, which is vital for educational equity.

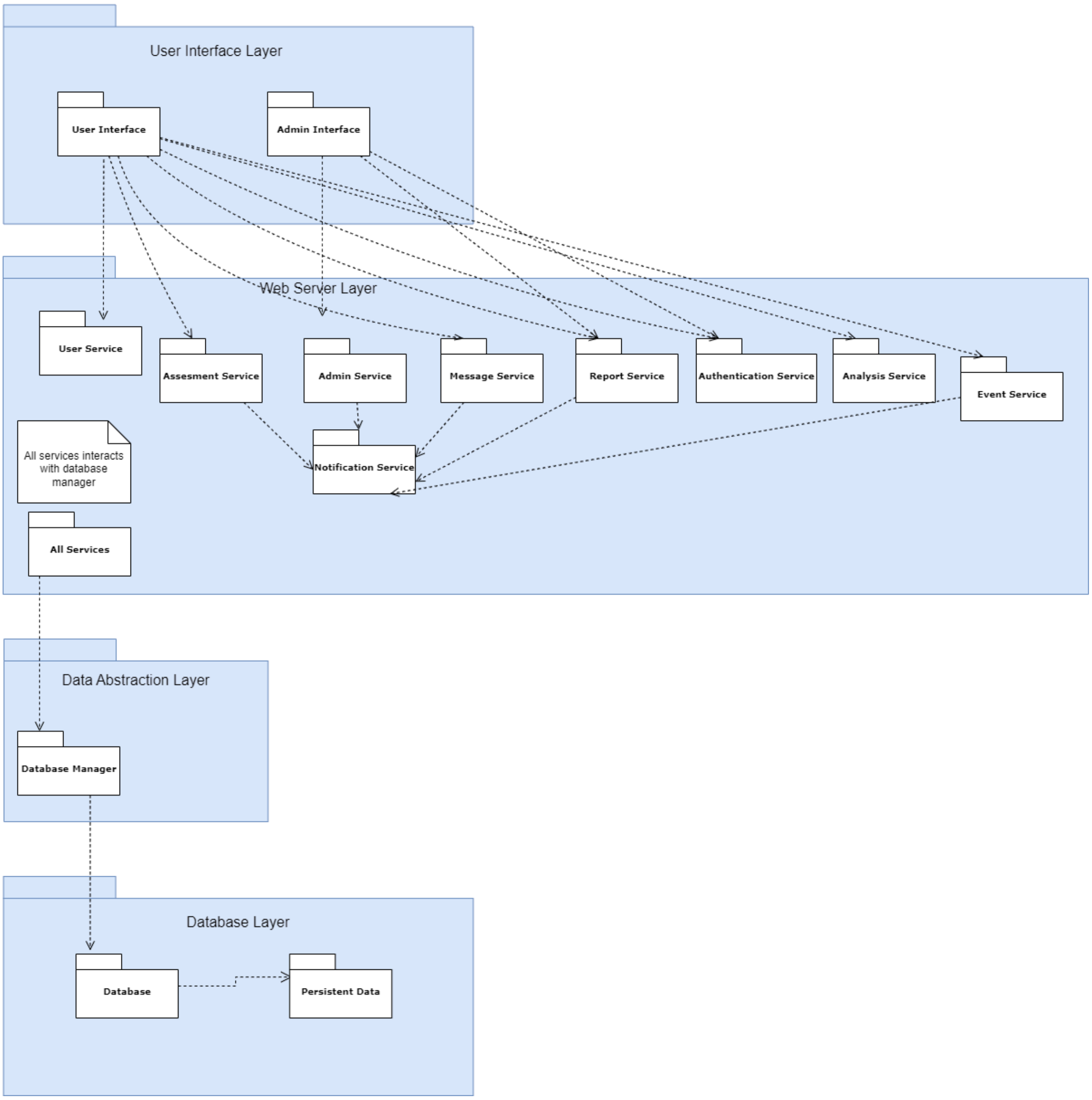
Architectural Style

When we choose a layered architecture for our software, it comes with several advantages, especially in terms of simplicity, organizational clarity, and ease of development. This architecture divides our system into distinct layers such as presentation, business, persistence, and database. Each layer’s defined roles enhance maintainability and allow for independent scalability and testing, improving the system’s robustness. Additionally, it supports high reusability, as components in one layer can be used by other layers without modification. However, the trade-offs include potential performance latency, as requests pass through multiple layers, and the risk of rigidity in cross-layer changes. Overall, the choice of a layered architecture balances development efficiency with the need for stability and modular separation, fitting well with the complex functionalities and scalability required by our educational software environment.

Connectors

For our system, a suitable type of connector could be a Middleware-enabled Connector. This type of connector would be ideal for managing interactions between various software components that handle student data, assessments, grading, and results. The Middleware-enabled Connector would provide essential services such as transaction management, security, and database connectivity, ensuring that the assessment system operates efficiently and securely. It would also support the integration of different software modules or services, such as user interfaces, reporting tools, and the database system, thereby facilitating a robust and scalable architecture for the university assessment system.

Subsystem Decomposition Diagram



Detailed View: https://drive.google.com/file/d/1imv9tfXxMBqVVVtwRhg-rP8Dt8xlHlg2/view?usp=sharing