Department of Computer Engineering, Bilkent University
CS319 Object Oriented Software Engineering
Group 11
Design Goals & Subsystem Decomposition Diagram - D3
Section 1 - Eray Tüzün
22/04/2025

Furkan Mert Aksakal 22003191

Furkan Komaç 22102165

Alper Biçer 22003097

Erkan Can Arslan 22103948

Deniz Yazıcı 21902557

Contents

Top 2 Design Goals

The design goals stem from the analysis of the **non‑functional requirements** for the TA Management System. While building the platform, the two goals below will have highest priority—even when trade‑offs arise, as explained at the end of each section.

User‑friendliness / Usability

The system must be effortless for every stakeholder—TAs, instructors, and department staff—to learn and operate.

- **Intuitive dashboards**: Each role lands on a tailored home‑screen that shows only the actions relevant to it (e.g., "Log Duty," "Approve Tasks," "Assign Proctors"). Common actions are reachable in **≤ 2 clicks**.
- **Guided forms & smart defaults:** When a TA records a duty, the form pre‑selects the current course/section and auto‑computes effort hours from a drop‑down template, reducing typing and errors.
- **Real‑time feedback:** Toast notifications confirm a submission or flag validation errors **< 1 s** after the user clicks "Save," so the user never wonders whether an action succeeded.
- **Accessible visual cues:** Colour‑blind‑safe status badges (green = approved, yellow = pending, red = rejected) make state obvious at a glance.
- **Search & filter everywhere:** All tables (TA list, proctor roster, duty log) support type‑ahead search and multi‑facet filtering, enabling staff to locate information in < **5s**.

Reliability

The platform must consistently perform critical operations—duty logging, proctor assignment, leave blocking—without error or data loss.

- **ACID transactions:** Every create/update/delete on duties, proctor slots, or leave requests runs inside a single database transaction; if any sub‑step fails, nothing commits.
- **Exactly‑once scheduling:** A message‑queue feeds the proctor‑assignment engine. Consumer groups guarantee that a TA is never double‑assigned for the same exam.
- **Health‑checked micro‑services:** Containers restart automatically on failure; rolling updates ensure ≥ **99.9 % monthly uptime**—critical during finals week when load peaks.

- **Versioned audit log:** Immutable, append-only records (timestamp, actor, before/after JSON diff) allow rapid forensic analysis and restore trust when disputes arise.
- **Nightly point-in-time backups & PITR:** Automatic snapshots plus a 7-day WAL archive let admins recover to any moment within the past week in case of corruption or operator error.

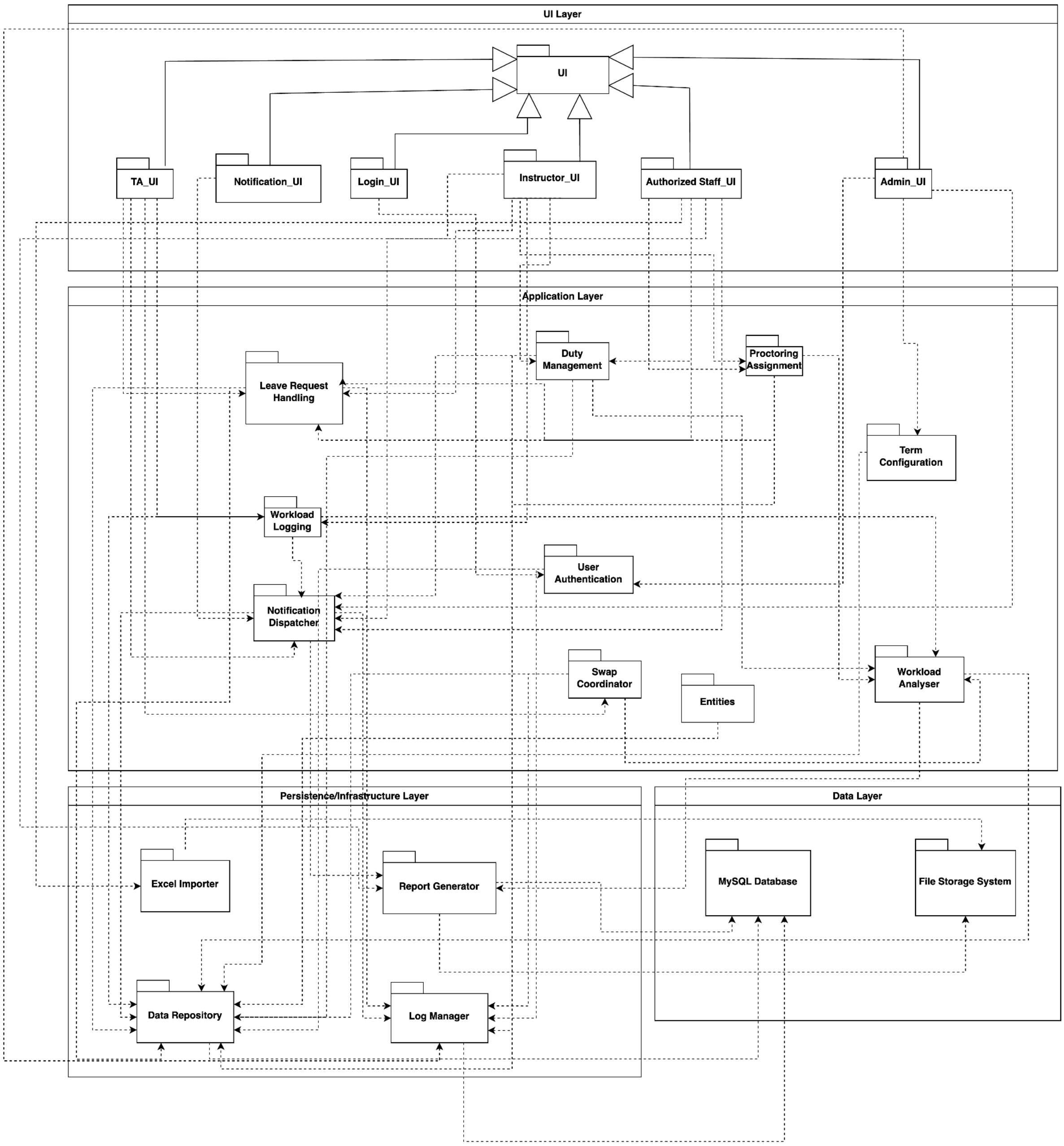
Possible Trade-offs

1. ***Reliability vs Modifiability*** – Defensive checks, contracts, and a > 85 % test-coverage gate slow down feature releases. Feature flags and blue/green deployments will let the team experiment safely.
2. ***Reliability vs Cost-effectiveness*** – Running redundant database replicas and a message bus increases infrastructure spend. We limit replicas to exam periods and scale down during off-peak months.
3. ***User-friendliness vs Efficiency*** – Rich UI components and live search add client-side weight, which may slightly increase initial page-load time. Lazy-loading and code-splitting will mitigate the impact.
4. ***User-friendliness vs Security*** – Keeping workflows "password-free" for instructors (SSO token only) improves UX but heightens risk if tokens leak. We counterbalance with 15-minute idle timeouts and device-bound refresh tokens.

By prioritising **User-friendliness** and **Reliability**, the TA Management System delivers a **smooth daily workflow** for its users while **protecting exam integrity** and historical workload records, in line with established system-design best practices.

Subsystem Decomposition Diagram

link: https://drive.google.com/file/d/1qpHeOokRLjLRHgvIW3bL7bTl1J7oLOyb/view?usp=sharing


UI Layer
UI
TA_UI
Notification_UI
Login_UI
Instructor_UI
Authorized Staff_UI
Admin_UI
Application Layer
Duty Management
Proctoring Assignment
Leave Request Handling
Term Configuration
Workload Logging
User Authentication
Notification Dispatcher
Swap Coordinator
Entities
Workload Analyser
Persistence/Infrastructure Layer
Data Layer
Excel Importer
Report Generator
MySQL Database
File Storage System
Data Repository
Log Manager