CS 319
Object-Oriented Software Engineering

ProctorHub
Deliverable 3 - 1st Iteration

Section-1
24.04.2025

Emine İrem Esendemir - 22202103
Halil Arda Özongun - 22202709
Yunus Emre Erkan - 22203670
Elif Lara Oğuzhan - 22203138
Sude Ergün - 22203822

Table Of Content

Design Goals **3**
1. Reliability 3
2. Usability 4

Design Trade-Offs **5**
1. Reliability vs Flexibility 5
2. Usability vs Functionality 6

System Decomposition Diagram **7**
Subsytems **8**

Design Goals

1. Reliability

We prioritized reliability as our top design goal because the old version of the proctoring assignment was completely manual, therefore prone to errors. The errors can occur in processes requiring short time, such as assigning teaching assistants (TA) to proctorings or tracking their leave of absence request. Before this application Excel files were managed by the department secretary. This might lead to the potential for data loss, inconsistent updates, and accidental scheduling conflicts due to human error.

One example was the handling of leave of absence requests. TAs would share their request for leave via email, and the department secretary would manually note it in a spreadsheet. This process had no built-in conflict checking, often resulting in overlapping assignments or missed updates when plans changed at the last minute.

Another example was the workload tracking. The old system only counted how many exams each TA had proctored, without considering total working hours. This approach ignores fairness and gives the possibility of imbalanced workload distribution.

We aim to eliminate this possibilities by automating the process and providing a role-based system to provide consistency and accurate handling of proctorings, leave of absences and workloads data. Leave of absences' conflicts are checked by the system and necessary actions are taken, and only approved by authorized staff. Proctor assignments, whether automatic or manual, are based on structured data like availability, leave status, fair workload distribution, and departmental alignment, ensuring that no assignment is missed or miscommunicated. This ensures smooth operations and reliability across all processes.

2. Usability

We chose usability one of our top design goals since our system will be used by many different types of users. Each of these users has different needs, so we designed separate and simple interfaces for each role.

TAs can see the courses they are taking and the exams where they are assigned as proctors in their clear schedule table. They can log their work hours, send leave of absence requests, and manage proctor swaps. For swapping, TA can either send a personal request to another specific TA, or post the request in a TA forum. This gives TAs more control while keeping everything organized in one place.

Course instructors can use the system to enter exam details, assign TAs manually to proctor exams, choose TAs for their courses, and check TA workloads. Their interface is made to make these tasks simple and fast, without having to deal with complex menus or systems.

Only admins can download reports about the overall workload and system activity. Department Chair’s can also approve or reject leave requests. Dean’s Office can manage the multidepartment exam proctorings.

Department secretaries can track and control all of the workload, leave of absence, proctoring, and swap functionalities in their interface easily, instead of working with multiple spreadsheets and manually checking for conflicts.

By giving each user group only the tools they need, and keeping the interface simple and focused, our system is easy to learn and use, avoids confusion, saves time, and makes daily tasks much easier for everyone involved. This makes it more likely that everyone will actually use the system and stop relying on Excel or manual communication.

Design Trade-Offs

1. Reliability vs Flexibility

For enhancing reliability, we implemented several system controls and constraints that intentionally limit user flexibility in certain scenarios. These constraints help ensure the system behaves consistently and avoids scheduling conflicts, incorrect assignments, or data inconsistencies.

For example, TAs are not allowed to freely modify or delete their approved work hour logs or accepted proctoring assignments. This prevents accidental loss.

Similarly, leave of absence requests can only be submitted before the 48 hours of the upcoming assignments and must be approved by authorized staff before taking effect. This limits last-minute changes, but ensures that proctor assignments remain valid and predictable.

In automatic assignments, several constraints are applied to prevent scheduling conflicts and ensure fairness, such as TAs on leave or enrolled in the course as students cannot be assigned or only PhD students may be assigned to MS/PhD-level exams. This reduces the systems flexibility, however, keeps it reliable.

2. Usability vs Functionality

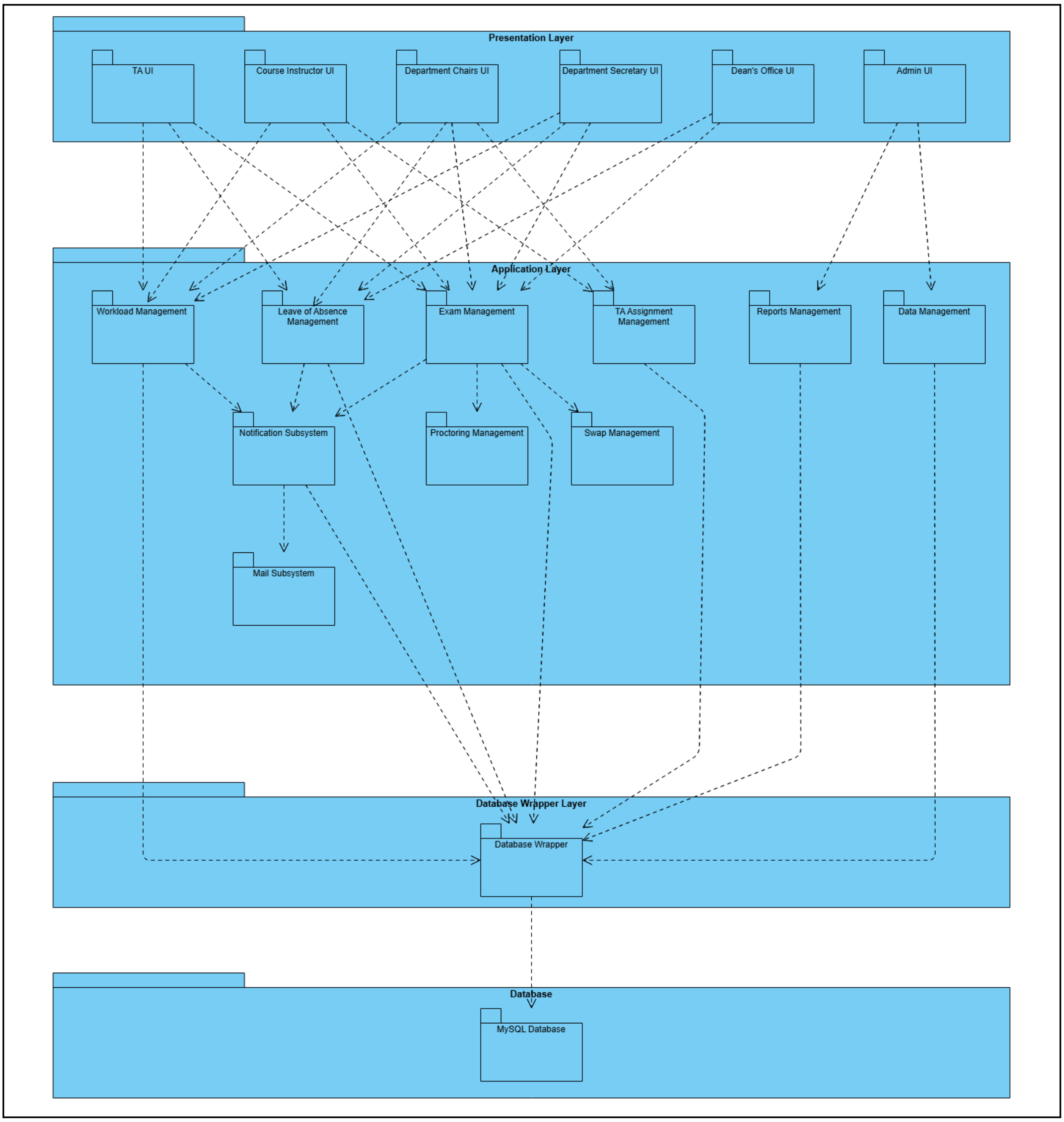
To prioritize usability, we designed system with a simple and intuitive interface for different users, aiming to ease of use. However, this emphasis on clarity and ease of use comes at the cost of reducing certain advanced functionalities.

For example, instead of offering complex filtering or customization options for reports and assignments, the system provides predefined views and templates to maintain consistency. Similarly, when submitting work hours, TAs select the task type from a predefined list (e.g., Lab, Recitation, Grading) rather than creating custom entries. This helps reduce user error and ensures standardized data for reporting, but it limits flexibility for edge cases or non-standard activities.

Similarly, when requesting a proctor swap, TAs can only select existing assignments from a filtered dropdown, rather than manually entering course or time details. This improves ease of use and prevents mistakes, but restricts more complex swap scenarios that might occur in special circumstances.

System Decomposition Diagram

Here is its drive link:
https://drive.google.com/file/d/18iU1f6xoUbyO5K9Z7gHBp35w9A7aTrnv/view



Subsytems

Workload Management: Tracks and logs the work hours submitted by TAs, categorized by task types. Submissions are reviewed and approved by course instructors.

Leave of Absence Management: Handles leave of absence requests submitted by TAs. It checks for conflicts with scheduled exams or assignments and routes the request to authorized personnel for approval or rejection.

Exam Management: Allows course instructors to create, update, and manage exam entries, including exam date, time, course, classes, and required number of proctors.

Proctoring Management: Controls the scheduling and coordination of proctoring duties. It ensures that TAs are properly assigned to exams based on availability, and system restrictions.

Swap Management: Enables TAs to request swaps for their proctoring duties. Requests are processed only if another eligible TA accepts the swap.

TA Assignment Management: Allows course instructors to specifically select TAs for their courses.

Reports Management: Generates detailed reports on TA workloads, proctoring histories, leave data, and overall system activity.

Data Management: Manages core system data such as user information, course information. It supports admin-level operations needed to maintain the integrity of backend data.

Notification Subsystem: Sends alerts and updates to users about key actions, including proctor assignments, leave approvals, swap status, and upcoming exams.

Mail Subsystem: Responsible for delivering email notifications triggered by the Notification Subsystem. Ensures external communication reaches users reliably.