



Bilkent University - 2024/2025 Spring
CS 319 - Object-Oriented Programming
TA Management System Project
Section 1
Team 9
Deliverable 3

Yüksel Barkın Baydar - 22201939

Deniz Öztürk - 22102126

Mehmet Anıl Yeşil - 22102614

Cankutay Dünder - 22103284

Emre Uçar - 22203675

CONTENTS

1. Design Goals:..... 2

 1.1. Reliability..... 2

 1.2. Usability..... 2

2. Design Trade-Offs:..... 4

 2.1 Reliability vs. Flexibility..... 4

 2.2 Usability vs. Security..... 4

 2.3 Usability vs. Administrative Control..... 4

3. Subsystem Decomposition..... 5

 3.1 Diagram..... 5

 3.2 Subsystem Details..... 5

 3.2.1 Authentication Subsystem:..... 5

 3.2.2 Profile Management Subsystem:..... 6

 3.2.3 Notification Subsystem:..... 6

 3.2.4 Management Subsystem:..... 6

 3.2.5 Dean Tools Subsystem:..... 6

 3.2.6 Report Management Subsystem:..... 6

 3.2.7 System Management Subsystem:..... 6

 3.2.8 Application Façade Subsystem..... 6

1. Design Goals:

1.1. Reliability

Reliability is the top priority for our TA Management System. Previously, TA tasks such as grading, lab work, recitations, office hours, and exam proctoring were managed manually. These manual processes led to inaccuracies, unauthorized modifications, and scheduling conflicts. Such inconsistencies affected TAs' workloads, creating potential unfairness in task distribution and risking operational disruptions.

To effectively address these issues, our system implements structured, automated procedures. TA duties must be logged before set deadlines, after which faculty members explicitly approve or reject each submission. This structured review process significantly reduces human error and enhances the accuracy of recorded workload information, ensuring fair and transparent workload management.

Additionally, our automated proctoring-assignment feature further enhances reliability by systematically assigning proctoring duties based on objective criteria such as current workload, course-specific constraints, and TA availability. It prioritizes TAs with the least accumulated work, ensuring equitable distribution and minimizing the risk of unfair assignments or double bookings. Moreover, clear restrictions ensure that TAs are not assigned to exams conflicting with their own coursework or approved leave.

However, emphasizing reliability inherently introduces trade-offs. To ensure strict data integrity and minimize errors, the system limits flexibility after task approvals and proctor assignments. Once finalized, adjustments to these records require formal administrative procedures. Although this rigorously ensures accuracy and reduces unauthorized data modifications, it increases administrative overhead, potentially making quick adjustments hard for users.

1.2. Usability

Usability is identified as our second critical design goal, given the diverse technical backgrounds of users, including Teaching Assistants, Faculty Members, and Departmental Staff. To accommodate varying technical skills, the TA Management System provides intuitive interfaces specifically tailored to each user role. For example, TAs benefit from simplified forms to log their tasks, manage leave requests, and initiate swaps, ensuring minimal confusion or frustration. To ground this goal in measurable terms, we require that any routine TA action—such as logging a duty, requesting a swap, or submitting a leave request—be completable in

no more than *seven clicks* and *under 30 seconds* on a standard broadband connection.

Our system further enhances usability through proactive communication mechanisms, such as automated notifications and reminders. These automated messages alert users of important deadlines, upcoming responsibilities, or pending approvals. Specifically, TAs receive email notifications when approaching deadlines for duty logging, thus significantly reducing forgotten or delayed submissions. We will monitor notification-driven task completion, targeting a *90 % on-time submission rate* within the first semester of deployment to quantitatively verify the efficacy of these reminders. Faculty and administrative staff similarly benefit from alerts related to pending approvals and assignments, simplifying their oversight duties.

Additionally, usability is improved by providing clear and role-specific dashboards. Faculty members have direct and straightforward access to approve logged tasks and manage proctor assignments. Administrators receive an organized overview of workload distributions and administrative tasks, streamlining their management processes. Such clear, user-friendly presentations of system functions significantly reduce training requirements and enhance user adoption.

Despite these benefits, prioritizing usability creates specific trade-offs regarding security. While advanced security measures, such as two-factor authentication (2FA), would enhance the protection of sensitive data, they typically introduce friction in user interactions. To maintain ease of use, the system relies instead on institutional single sign-on (SSO) credentials with robust password requirements. This balanced approach ensures acceptable levels of security while preserving intuitive and rapid access for all users. Periodic usability audits will ensure that added security or feature updates never raise the click count for any core workflow above the seven-click threshold or extend task completion times beyond 30 seconds, preserving our quantitative usability targets.

2. Design Trade-Offs:

2.1 Reliability vs. Flexibility

Ensuring reliability in the TA Management System involves strictly controlling data modifications after initial approvals. Once TA duties, proctor assignments, or leave requests have been reviewed and approved by faculty or administrators, subsequent changes require formal administrative intervention. While this rigorous approach guarantees data integrity, minimizes errors, and ensures transparency, it introduces rigidity into the system. Users accustomed to quickly correcting minor errors or directly adjusting schedules may find the required administrative oversight burdensome. Thus, while reliability and accuracy significantly improve, system flexibility and ease of minor adjustments are inevitably constrained.

2.2 Usability vs. Security

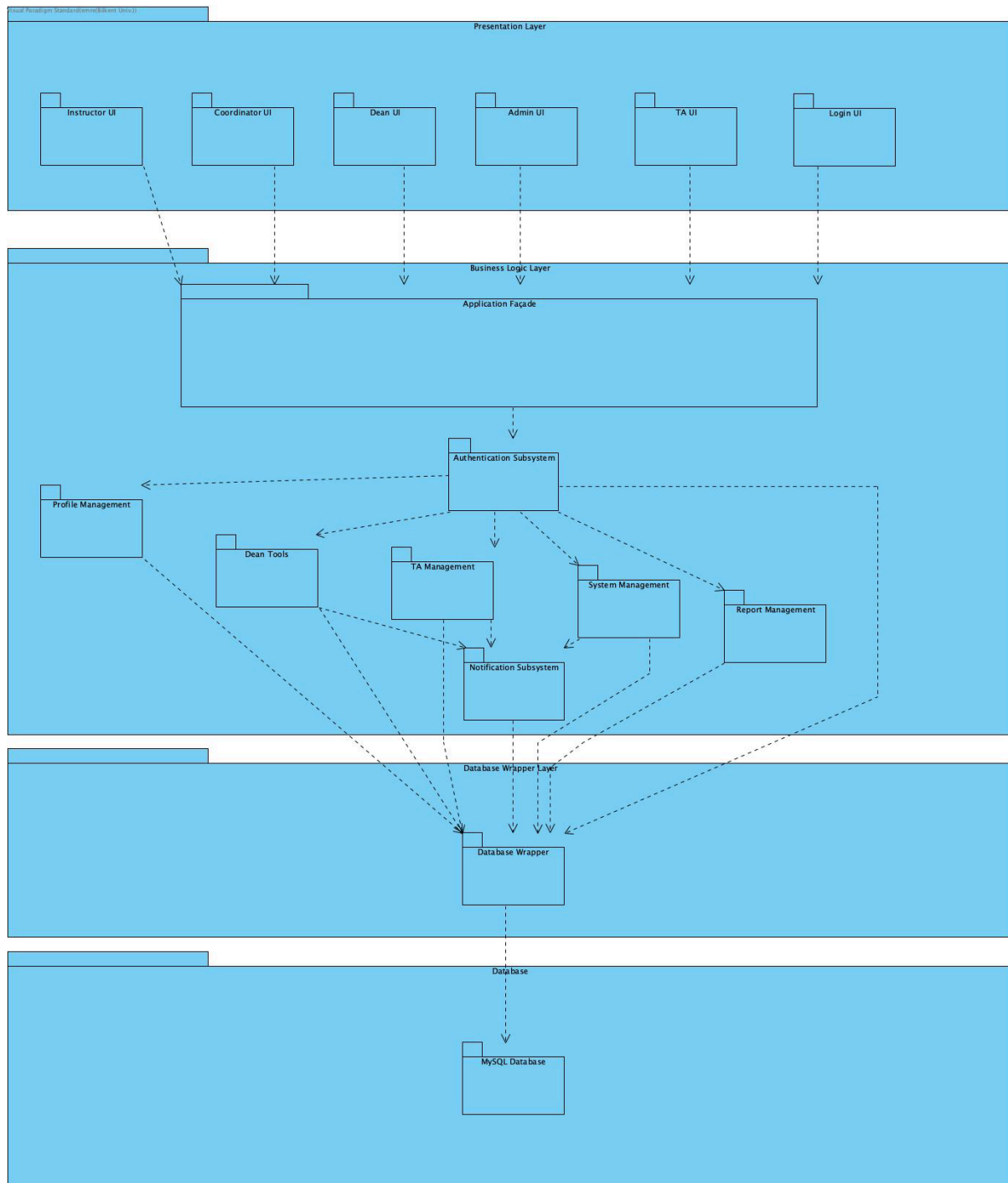
To maintain high usability, especially for less technically proficient users such as certain faculty members or newer TAs, our system avoids implementing excessively strict authentication methods such as two-factor authentication (2FA). While 2FA would undoubtedly enhance security by protecting sensitive user and workload data from unauthorized access, it introduces complexity and potential frustration for users who require frequent system access, especially under time constraints during peak academic periods. Instead, our system relies on institutional single sign-on (SSO) credentials, combined with robust password requirements and vigilant session management. This balanced approach ensures that user experience remains streamlined and approachable, despite sacrificing some potential security improvements.

2.3 Usability vs. Administrative Control

Emphasizing usability necessitates empowering users with intuitive, simplified interfaces for routine tasks such as duty logging, proctor swapping, and leave requests. This empowerment enhances user engagement, encourages timely compliance, and significantly reduces administrative support requests. However, simplifying user interactions and reducing administrative involvement in routine operations may limit the oversight capabilities of department staff or administrators. Consequently, the system must carefully balance simplified user workflows with sufficient administrative monitoring and control mechanisms, accepting a degree of reduced oversight to achieve high overall user satisfaction and engagement.

3.Subsystem Decomposition

3.1 Diagram



3.2 Subsystem Details

3.2.1 Authentication Subsystem:

Manages all login/logout and password-reset flows, securely validates credentials against stored hashes, issues JWT/session tokens, and enforces role-based access control for every API call.

3.2.2 Profile Management Subsystem:

Lets any authenticated user view and edit their own profile—updating contact details (with Bilkent-email and phone-format validation) and changing passwords (with complexity and “no reuse” checks)—while recording audit trails.

3.2.3 Notification Subsystem:

Centralizes event-driven messaging by generating in-app alerts and email notifications (via the Bilkent SMTP server) for actions such as leave-request outcomes, proctor-swap invitations, assignment updates, report completions, and system announcements.

3.2.4 Management Subsystem:

Provides TAs and department staff with tools to log or mark as complete all assigned duties (labs, grading, office hours, exam proctoring), submit and track leave requests, swap proctor assignments, review historical workloads, and generate detailed personal workload reports.

3.2.5 Dean Tools Subsystem:

Empowers deans to create, edit, or remove course records, schedule exams (with conflict detection on date, time, and room), assign or reassign proctoring duties across department pools; and initiate filtered or aggregated workload/assignment reports.

3.2.6 Report Management Subsystem:

Allows admins (and deans on demand) to submit, review, accept or reject report-generation requests—and then compile and deliver log-based or category-specific reports (login, swap, duty, proctor, etc.) in downloadable form.

3.2.7 System Management Subsystem:

Handles application-wide administration—deploying updates with minimal downtime, managing user role assignments and permissions, configuring global settings (cookie consent, GDPR options), and orchestrating automated backups and recovery procedures. It also publishes system-level notifications (maintenance windows, deployment updates, backup successes/failures, configuration changes) via the Notification Subsystem to keep administrators and end users informed.

3.2.8 Application Façade Subsystem

Serves as the unified API gateway that accepts all Presentation Layer requests and delegates them to the appropriate business subsystems. It invokes the Authentication Subsystem for login flows and forwards JWT tokens for validation on subsequent calls.