TA Management System

Deliverable-1 Final Iteration

08.04.2025

Course: CS 319

Section: 01

Group No.: 7

Instructor: Eray Tüzün

Teaching Assistant: Yahya Elnouby

Group Members: Bilge İdil Öziş [22102365]

Burak Kağan Gür [22203365]

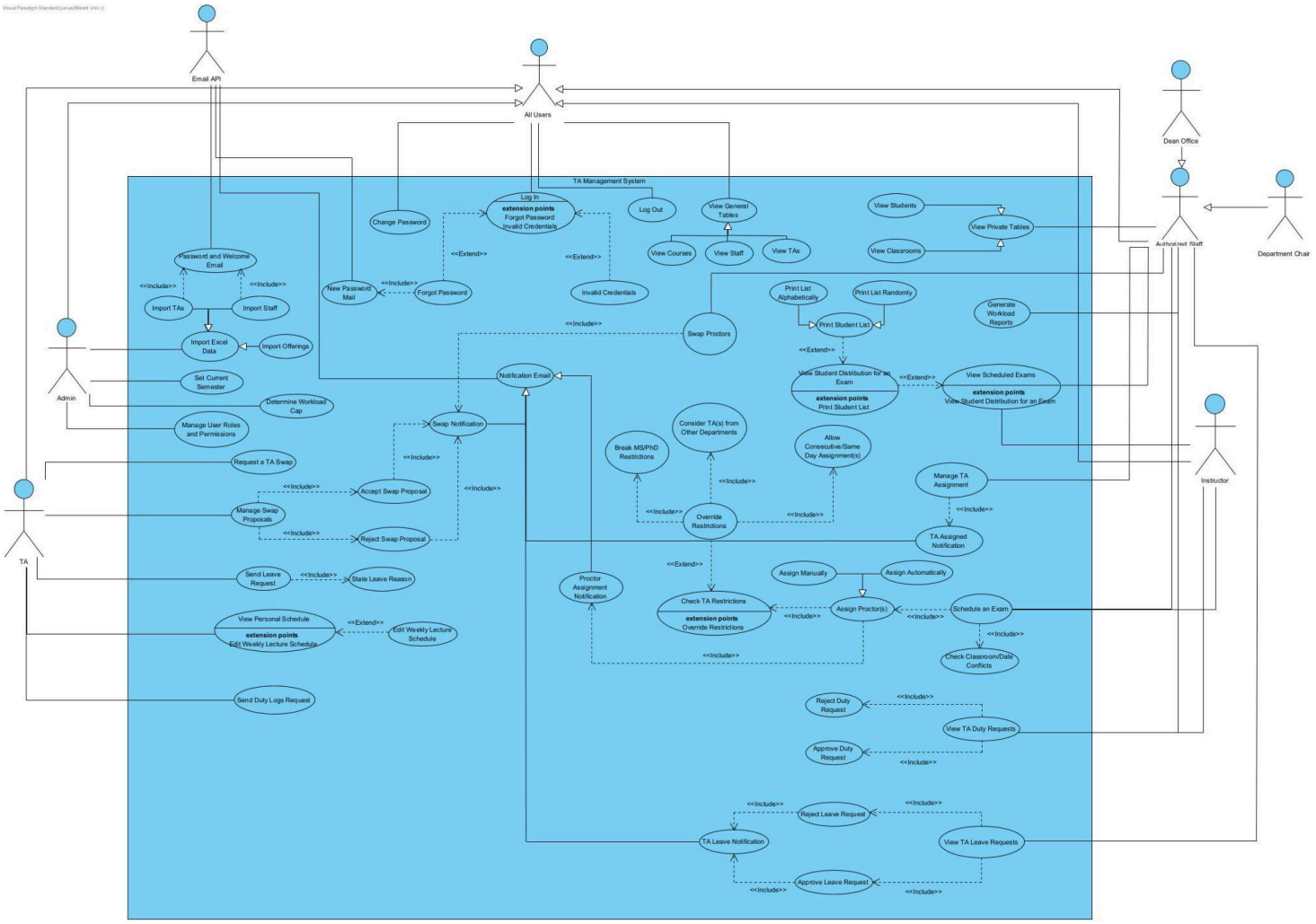
Daib Malik [22201013]

Yunus Günay [22203758]

Yusufbek Karamatov [22301586]

Contents

1. Use-Case Diagram



2. Use-Case Textual Description

2.1 Authentication & Home Page Package

Use Case Name: Login

1. Participating Actor: TA Management System User
2. Flow of Events:

   2.1 The actor opens the website.

   2.2 The actor enters their email and password.

   2.3 The actor presses the login button on the main page.

3. Entry Condition: The actor should be authorized to log in.
4. Exit Condition: Successful Login.
5. Quality Requirements: -

Use Case Name: Invalid Credentials

1. Participating Actor: TA Management System User
2. Flow of Events:

   2.1 The actor opens the website.

   2.2 The actor enters their login credentials and presses the login button on the main page.

   2.3 (Extending the Login Case) If the credentials do not match, the system displays an error message.

3. Entry Condition: The actor should enter their information incorrectly.
4. Exit Condition: An error is shown telling the user about the incorrect login attempt.
5. Quality Requirements: The system should detect when an email does not exist or password does not match with the email, and provide a clear, user-friendly error message within 4 seconds.

Use Case Name: Forget Password

1. Participating Actor: TA Management System User
2. Flow of Events:

   2.1 The actor clicks on the "Forgot Password?" link on the login page.

   2.2 The system prompts the actor to enter their registered email address.

   2.3 The actor enters their email and submits a request.

   2.4 The system checks if email exists. If the email is valid, the system generates a password reset link and sends an email (extends New Password Mail). If the email is not found, the system displays an error message.

   2.5 The actor is informed that an email has been sent (if valid).

3. Entry Condition: The actor clicks the "Forgot Password?" button on the login page.
4. Exit Condition: The system sends a password reset email or informs the user of an issue.
5. Quality Requirements: The password reset link should be secure and time-limited, i.e., should expire after 5 minutes and should be sent within 8 seconds. Moreover, the system should ensure that no security-sensitive information is exposed in error messages.

Use Case Name: Change Password

Participating Actor: All Users

Flow of Events:

1. The user logs in and navigates to the "Change Password" page.

2. They enter their current password and the new password (with confirmation).

3. The system verifies the current password.

4. If valid, the new password is securely hashed and stored.

5. A success message is displayed, and the session may optionally be refreshed.

Entry Condition: The user is authenticated and initiates a password change request.
Exit Condition: Password is updated in the database; user receives confirmation.

Quality Requirements:

- Passwords must be at least 8 characters and include at least one number and one symbol.
- The system must hash passwords using bcrypt or equivalent.
- Change must occur within 3 seconds and invalidate old sessions if enabled.

Use Case Name: New Password Mail

1. Participating Actor: Email API
2. Flow of Events:

   2.1 The system sends an email to the actor containing a message explaining the reset process and a secure link to reset the password.

   2.2 The actor receives the email and clicks the reset link.

   2.3 The password of the user is changed with the new one.

3. Entry Condition: The actor whose email is registered in the system requests a password reset.
4. Exit Condition: The password of the actor is changed.
5. Quality Requirements: The email should be sent securely and the system should limit the number of failed attempts to 5 to prevent brute-force attacks. The email should include a one-time verification token to prevent phishing attempts.

Use Case Name: Password and Welcome Email

Participating Actor: System (automatic trigger), Admin
Flow of Events:

1. Admin imports new users.

2. For each new user, the system generates a welcome email.

3. Email includes login link and temporary password.

Entry Condition: New user record is created.
Exit Condition: Welcome email is sent.
Quality Requirements:

- Email must be delivered within 5 seconds.

2.2 TA Duties Package

Use Case Name: Assign/Manage Duties

1. Participating Actor: Authorized Staff, Instructor (for specific courses)
2. Flow of Events:

   2..1 The actor logs into the system.

   2.2 The actor views existing duty assignments for TAs or creates a new one.

   2.3 The actor enters details (course, task type, date/time, expected duration).

   2.4 The system checks for conflicts (schedule, TA availability, workload caps).

   2.5 The actor confirms the assignment, and the system notifies the assigned TA and updates records.

3. Entry Condition: The actor has the necessary role/permissions and TAs exist in the system with the assigned courses.
4. Exit Condition: A new duty is assigned or an existing assignment is updated.
5. Quality Requirements: The system should prevent double-booking, enforce workload limits, and send confirmation emails to assigned TAs.

Use Case Name: Submit Duty Logs Request

1. Participating Actor: TA
2. Flow of Events:

   2.1 TA logs into the system and selects the course and the type of the duty (lab, grading, office hours, etc.)

   2.2 TA enters date/time and duration spent on the duty.

   2.3 TA submits the duty log request, and the system notifies the corresponding Instructor or Staff for approval.

3. Entry Condition: TA is assigned to at least one course.
4. Exit Condition: The duty log is recorded and pending approval.
5. Quality Requirements: The system should validate the data (non-negative duration, correct date format, etc.) and should ensure a response (approval/rejection) within 24 hours.

Use Case Name: View TA Duty Requests

1. Participating Actor: Instructor or Authorized Staff
2. Flow of Events:

   2.1 The actor logs into the system and goes to the requests page.

   2.2 The system displays a list of duty logs submitted by TAs.

3. Entry Condition: TAs have submitted duty logs.
4. Exit Condition: The actor can view and proceed to approve or reject requests.

5. Quality Requirements: The system must show real-time status (approved, rejected, pending) within 3 seconds of occurrence of any of those actions and provide relevant details.

Use Case Name: Define an Exam

1. Participating Actor: Instructor, Authorized Staff
2. Flow of Events:

   2.1 The actor navigates to the exam management section.

   2.2 The actor enters exam details (course, date, time, duration, location).

   2.3 The actor specifies the number of proctors needed.

   2.4 The system saves the exam information.

3. Entry Condition: The actor has appropriate permissions.
4. Exit Condition: New exam is created in the system.
5. Quality Requirements: The system should be able to save exam information in 4 seconds at most, and display relevant messages (saved exam, or rejected due to scheduling conflicts, or other issues). The system should ensure that exams are scheduled at least 7 days in advance.

Use Case Name: View Scheduled Exams/Proctoring Assignments

1. Participating Actor: Instructor, Authorized Staff, or Department Chair
2. Flow of Events:

   2.1 The actor navigates to the exams/proctoring assignments page.

   2.2 The system displays upcoming exams, assigned TAs, and any open proctoring slots.

3. Entry Condition: Exams and proctoring data must exist in the system.
4. Exit Condition: The actor reviews exam schedules and proctor assignments.
5. Quality Requirements: The interface should clearly distinguish between filled and unfilled proctor positions.

2.3 Information Management Package

Use Case Name: View Students

1. Participating Actor: Instructor, Authorized Staff
2. Flow of Events:

   2.1 The actor navigates to the student management section.

   2.2 The system displays a list of students with relevant information.

   2.3 The actor can filter or search for specific students.

3. Entry Condition: The actor has appropriate permissions.

4. Exit Condition: Student information is displayed.
5. Quality Requirements: The system should provide efficient filtering and sorting options and should display sorted results immediately, i.e., in 1 second.

Use Case Name: View Courses

1. Participating Actor: TA Management System User
2. Flow of Events:

   2.1 The actor navigates to the courses section.

   2.2 The system displays a list of courses based on the user's role.

   2.3 The actor can select a course to view more details.

3. Entry Condition: The actor has permissions to view course information.
4. Exit Condition: Course information is displayed according to user role.
5. Quality Requirements: Courses should be searchable by instructor, semester, or department, and searches should return results within 1 second.

Use Case Name: View Classrooms

1. Participating Actor: Instructor, Authorized Staff
2. Flow of Events:

   2.1 The actor navigates to the classroom management section.

   2.2 The system displays a list of available classrooms with capacity and equipment information.

   2.3 The actor can filter or search for specific rooms.

3. Entry Condition: The actor has appropriate permissions.
4. Exit Condition: Classroom information is displayed.
5. Quality Requirements: The system should indicate current and upcoming room availability.

Use Case Name: View Staff

1. Participating Actor: Authorized Staff, Department Chair
2. Flow of Events:

   2.1 The actor navigates to the staff management section.

   2.2 The system displays a list of staff members with relevant information.

   2.3 The actor can filter or search for specific staff members.

3. Entry Condition: The actor has appropriate permissions.
4. Exit Condition: Staff information is displayed.
5. Quality Requirements: Information should be organized by department and role.

Use Case Name: View Offerings

1. Participating Actor: TA Management System User
2. Flow of Events:

   2.1 The actor navigates to the course offerings section.

   2.2 The system displays current and upcoming course offerings.

   2.3 The actor can view details including instructors and assigned TAs.

3. Entry Condition: The actor has permissions to view offerings.
4. Exit Condition: Course offering information is displayed.
5. Quality Requirements: The interface should clearly show the status of TA assignments for each offering.

Use Case Name: Manage Information

1. Participating Actor: Authorized Staff, Department Chair
2. Flow of Events:

   2.1 The actor navigates to the information management section.

   2.2 The actor selects the type of information to manage (courses, classrooms, students, staff).

   2.3 The actor can view, add, edit, or delete information based on their permissions.

3. Entry Condition: The actor has appropriate permissions.
4. Exit Condition: Information is updated in the system.
5. Quality Requirements: Changes should be logged for audit purposes.

Use Case Name: Add

1. Participating Actor: Authorized Staff, Department Chair
2. Flow of Events:

   2.1 The actor selects the appropriate section (courses, students, staff, etc.).

   2.2 The actor clicks the "Add" button.

   2.3 The actor fills out the required information in the form.

   2.4 The system validates the information.

   2.5 The actor confirms, and new information is added to the system.

3. Entry Condition: The actor has permission to add information.
4. Exit Condition: New information is successfully added.
5. Quality Requirements: The system should add information within 4 seconds and check for duplicates and data integrity.

Use Case Name: Edit

1. Participating Actor: Authorized Staff, Department Chair
2. Flow of Events:

   2.1 The actor selects an existing record.

   2.2 The actor clicks the "Edit" button.

   2.3 The actor modifies information in the pre-filled form.

   2.4 The system validates the changes.

   2.5 The actor confirms, and the information is updated.

3. Entry Condition: The record exists and the actor has permission to edit.
4. Exit Condition: Information is updated in the system.
5. Quality Requirements: The system should save edited information within 3 seconds and maintain data history and audit trails.

Use Case Name: Delete

1. Participating Actor: Authorized Staff, Department Chair
2. Flow of Events:

   2.1 The actor selects an existing record.

   2.2 The actor clicks the "Delete" button.

   2.3 The system confirms the deletion request.

   2.4 The actor confirms, and the record is removed or marked as inactive.

3. Entry Condition: The record exists and the actor has permission to delete.
4. Exit Condition: The record is removed from active use.
5. Quality Requirements: The system should check for dependencies before deletion, and then delete within 4 seconds.

2.4 Request Management Package

Use Case Name: Request a TA Swap

1. Participating Actor: TA
2. Flow of Events:

   2.1 The actor logs into the system and goes to the duties page.

   2.2 The actor selects an assigned duty or proctoring slot they want to swap.

   2.3 The actor initiates a swap request, choosing another TA as the proposed swap partner.

   2.4 The system sends a swap proposal to the selected TA.

   2.5 If the second TA accepts, the system reassigns duties and updates workloads.

   2.6 Both TAs and any relevant instructor/staff are notified.

3. Entry Condition: TA has at least one active assignment.

4. Exit Condition: Duty/proctoring assignment is swapped or the request is declined.
5. Quality Requirements: The system must prevent scheduling conflicts or exceeding workload caps, and maintain a log of swaps to avoid repeated or cyclical swapping. A swap request should be processed within 3 seconds, and an email notification should be sent instantly.

Use Case Name: Send Leave Request

Participating Actor: TA

Flow of Events:

1. TA fills out leave form with type, dates, and reason.
2. System checks for conflicts (e.g., overlapping duties).
3. Request is submitted for review.

Entry Condition: TA is logged in and authenticated.

Exit Condition: Leave request is saved with status = “Pending”.

Quality Requirements:

- Reason must be at least 10 characters.
- Conflict checker must run within 2 seconds.

- Must support Medical, Conference, Vacation, and Other types.

Use Case Name: State Leave Reason

Participating Actor: TA

Flow of Events:

1. As part of the leave request, the TA provides a written reason.
2. The reason is stored and shown to reviewers.

Entry Condition: TA is editing or creating leave request.

Exit Condition: Reason field is saved or validated.

Quality Requirements:

- Input should allow multi-line text.
- Character limit: 500.
- Must be stored securely and included in notifications.

Use Case Name: Approve Request

1. Participating Actor: Instructor, Authorized Staff, or Department Chair
2. Flow of Events:

2.1 The actor logs into the system and goes to the requests page.

2.2 The actor reviews details of a request (duty log, leave, swap, assignment).

2.3 The actor clicks the approve button.

2.4 The system updates the request status to "Approved."

2.5 TAs are notified.

3. Entry Condition: A request is awaiting approval.
4. Exit Condition: Request status is updated to "Approved."
5. Quality Requirements: The system should log approval actions with timestamp and approver and notify TAs via both email and system notification within 3 seconds.

Use Case Name: Reject Request

1. Participating Actor: Instructor, Authorized Staff, or Department Chair
2. Flow of Events:

2.1 The actor logs into the system and goes to the requests page.

2.2 The actor reviews details of a request (duty log, leave, swap, assignment).

2.3 The actor clicks the reject button and optionally provides a reason.

2.4 The system updates the request status to "Rejected."

2.5 TAs are notified.

3. Entry Condition: A request is awaiting approval.
4. Exit Condition: Request status is updated to "Rejected."
5. Quality Requirements: The system should log rejection action and notify TAs via both email and system notification within 3 seconds. The system should require a reason for rejection and send a rejection message to the other TA as part of notification.

Use Case Name: Manage Swap Proposals

1. Participating Actor: TA (receiving or initiating a swap)
2. Flow of Events:

   2.1 TA receives a notification of a swap proposal.

   2.2 TA reviews the proposed swap details (duty type, date/time, etc.).

   2.3 TA accepts or rejects the proposal.

   2.4 If accepted, the system updates both TAs' workloads.

   2.5 Notifications are sent to both TAs and relevant staff.

3. Entry Condition: A swap request must have been initiated.
4. Exit Condition: The swap is completed or declined.
5. Quality Requirements: The system ensures no scheduling or workload conflicts.

Use Case Name: View Personal Schedule

1. Participating Actor: TA, Instructor
2. Flow of Events:

   2.1 The actor logs into the system and navigates to the personal schedule section.

   2.2 The system displays the actor's assigned duties, classes, and other commitments.

   2.3 The actor can filter by date range or duty type.

3. Entry Condition: The actor has a valid user account.
4. Exit Condition: The actor views their schedule.
5. Quality Requirements: The schedule should be visually clear and exportable. All filter and sort options should update the table in 1 second.

Use Case Name: Manage Leave Request

1. Participating Actor: TA
2. Flow of Events:

   2.1 TA logs into the system and navigates to the leave management section.

   2.2 TA views current and past leave requests.

2.3 TA can create new requests, modify pending requests, or cancel requests.

3. Entry Condition: TA has a valid user account.
4. Exit Condition: Leave request is created, modified, or canceled.
5. Quality Requirements: The system should prevent overlapping leave requests.

Use Case Name: View Approved Leaves

1. Participating Actor: Instructor, Authorized Staff, Department Chair
2. Flow of Events:

   2.1 The actor navigates to the leave management section.

   2.2 The system displays all approved leaves, filterable by date and TA.

   2.3 The actor can view details of each leave.

3. Entry Condition: The actor has appropriate permissions.
4. Exit Condition: Approved leaves are displayed.
5. Quality Requirements: The view should highlight leaves that affect upcoming duties. Filters and sorting must update the table within 1 second.

Use Case Name: Submit Swap Proposal

1. Participating Actor: TA

2. Flow of Events:

    2.1 TA selects a duty they want to swap.

    2.2 TA chooses another TA to propose the swap to.

    2.3 TA submits the proposal with optional details/reason.

    2.4 The system notifies the selected TA.

3. Entry Condition: TA has an assigned duty eligible for swapping.
4. Exit Condition: Swap proposal is sent to another TA.
5. Quality Requirements: The system should verify eligibility of both TAs. It should prevent more than 3 pending swap requests for the same duty.

Use Case Name: Manage User Roles and Permissions

1. Participating Actor: Department Chair, System Administrator
2. Flow of Events:

    2.1 The actor navigates to the user management section.

    2.2 The actor selects a user to modify.

    2.3 The actor assigns or modifies roles and permissions.

    2.4 The system updates the user's access rights.

3. Entry Condition: The actor has administrator privileges.
4. Exit Condition: User's roles and permissions are updated.

5. Quality Requirements: The system should enforce role separation and least privilege.

2.5 Assignment Management Package

Use Case Name: Assign Proctors

Participating Actor: Authorized Staff
Flow of Events:

1. Staff selects exams and triggers proctor assignment.
2. Can assign manually (drag/drop) or use auto-fill.
3. System checks all constraints and fills proctor slots.
4. Staff reviews and confirms assignment.

Entry Condition: An upcoming exam is scheduled.
Exit Condition: TA proctors are assigned.
Quality Requirements:

- No duplicate TA-to-exam conflict.
- Auto-fill must complete within 5 seconds.
- Result must be editable before saving.

Use Case Name: Assign Manually

1. Participating Actor: Instructor, Authorized Staff

2. Flow of Events:

   2.1 The actor navigates to the assignment section.

   2.2 The actor selects a specific duty, course, or exam.

   2.3 The actor views eligible TAs and their current workload.

   2.4 The actor manually selects TAs for the assignment.

   2.5 The system validates the selection and confirms the assignment.

3. Entry Condition: The actor has appropriate permissions.
4. Exit Condition: TAs are assigned to the specified duty.
5. Quality Requirements: The system should prevent assignments that violate constraints and warn users. The interface should clearly display TA availability and workload.

Use Case Name: Request TA Reassignment

1. Participating Actor: Instructor
2. Flow of Events:

   2.1 The instructor navigates to the TA assignments for their course.

   2.2 The instructor selects a TA assignment they wish to change.

   2.3 The instructor submits a reassignment request with justification.

2.4 The system forwards the request to the Authorized Staff or Department Chair.

3. Entry Condition: The instructor has an assigned TA.
4. Exit Condition: Reassignment request is submitted.
5. Quality Requirements: The system should track reassignment frequency and reasons.

Use Case Name: Override Restrictions

Participating Actor: Authorized Staff

Flow of Events:

1. The staff attempts an assignment that violates one or more constraints (e.g., same-day conflict, department mismatch).

2. The system blocks the assignment and prompts for override.

3. Staff confirms override, provides optional justification.

4. System logs the override and proceeds with the assignment.

Entry Condition: The user is attempting a restricted assignment.

Exit Condition: Assignment is completed and restriction override is logged.

Quality Requirements:

- Must log user ID, timestamp, and override reason.
- Only users with override permissions can proceed.
- Should not affect other valid assignments.

Use Case Name: Break MS/PhD Restrictions

Participating Actor: Authorized Staff

Flow of Events:

1. The system detects that the assigned TA is a graduate student with limited duty scope.
2. Staff chooses to override this restriction.
3. Assignment proceeds with a flag indicating MS/PhD exception.

Entry Condition: Target TA is flagged as MS/PhD restricted.

Exit Condition: Assignment saved with override flag.

Quality Requirements:

- Warning must be displayed before allowing override.
- Override count should be recorded per semester.

Use Case Name: Consider TAs from Other Departments

Participating Actor: Authorized Staff

Flow of Events:

1. While assigning, no eligible TAs are found in the current department.

2. System suggests available TAs from other departments.

3. Staff reviews and selects cross-department TAs.

4. Assignment saved and marked as cross-department.

Entry Condition: No in-department TA meets criteria.

Exit Condition: TA from another department is assigned.

Quality Requirements:

- Should show department mismatch notice.

- Assignment summary must indicate cross-department status.

Use Case Name: Allow Consecutive/Same Day Assignments

Participating Actor: Authorized Staff

Flow of Events:

1. System detects multiple assignments for the same day or consecutive time blocks.

2. Staff explicitly allows scheduling.

3. Assignment proceeds, system marks exception.

Entry Condition: Time conflict exists on duty being assigned.
Exit Condition: Assignment is confirmed.
Quality Requirements:

- System must track which assignments were exceptions.

Use Case Name: Check TA Restrictions

Participating Actor: System (automatic) / Authorized Staff
Flow of Events:

1. System checks for TA overload, back-to-back duty, department match, etc.

2. Violations (if any) are shown to the actor.

3. Options to override (if allowed) are presented.

Entry Condition: TA is selected for assignment.

Exit Condition: Decision is made (assigned or not).

Quality Requirements:

- Must check: workload cap, department, exam conflicts.
- Output shown within 2 seconds.

Use Case Name: Process TA Assignments

1. Participating Actor: Department Chair, Authorized Staff
2. Flow of Events:

   2.1 The actor navigates to the TA assignment processing section.

   2.2 The system displays courses needing TA assignments.

   2.3 The actor can manually assign or use automatic assignment.

   2.4 The system validates assignments against constraints.

   2.5 The actor confirms finalized assignments.

3. Entry Condition: Courses requiring TAs must be defined in the system.
4. Exit Condition: TAs are assigned to courses.
5. Quality Requirements: The system must ensure balanced TA workloads before confirmation. The final assignment list should be exportable for record-keeping.

Use Case Name: View/Distribute Student List

1. Participating Actor: Instructor, TA
2. Flow of Events:

   2.1 The actor navigates to the course section.

   2.2 The actor selects a specific course.

   2.3 The system displays the student list.

   2.4 For instructors, an option to distribute students among TAs is available.

3. Entry Condition: The actor is associated with the course.
4. Exit Condition: Student list is viewed or distributed.
5. Quality Requirements: Distribution should be balanced and configurable. The interface should display student details clearly for easy assignment.

Use Case Name: Post List Announcement

1. Participating Actor: Authorized Staff
2. Flow of Events:

   2.1 The actor navigates to the announcement section.

   2.2 The actor creates a new announcement regarding TA assignments.

   2.3 The actor uploads or links to assignment lists.

2.4 The actor publishes the announcement.

3. Entry Condition: TA assignments are finalized.
4. Exit Condition: Assignment announcement is published.
5. Quality Requirements: Announcements should be accessible to all relevant users. The system must notify all affected TAs and instructors when an announcement is posted.

Use Case Name: Determine Workload Cap

Participating Actor: Department Chair or Authorized Staff

Flow of Events:

1. Actor accesses workload settings per semester.
2. Sets max allowable hours per TA (may vary by TA type or program).
3. System updates scheduling constraints accordingly.

Entry Condition: The current semester is active.

Exit Condition: New workload cap is stored and enforced.

Quality Requirements:

- Applied in real-time to new assignments.
- Supports per-TA or global workload policies.

2.6 Reporting & User Management Package

Use Case Name: Generate Workload Reports

1. Participating Actor: Authorized Staff, Department Chair
2. Flow of Events:

   2.1 The actor navigates to the reporting section.

   2.2 The actor selects the type of report (workload, assignments, duty logs).

   2.3 The actor specifies parameters (date range, courses, TAs).

   2.4 The system generates and displays the report.

   2.5 The actor can export the report in various formats.

3. Entry Condition: The actor has permission to access reports.
4. Exit Condition: Report is generated and displayed/exported.
5. Quality Requirements: Reports should be accurate, comprehensive, and exportable. The system must generate reports within 5 seconds. Reports should be exportable in multiple formats (PDF, Excel, CSV).

Use Case Name: View User Duties and Permissions

1. Participating Actor: Authorized Staff, Department Chair
2. Flow of Events:

2.1 The actor navigates to the user management section.

2.2 The actor selects a user to view.

2.3 The system displays the user's current duties, roles, and permissions.

3. Entry Condition: The actor has appropriate permissions.
4. Exit Condition: User information is displayed.
5. Quality Requirements: The interface should clearly distinguish between different roles. Users should only see permissions relevant to their access level.

Use Case Name: Manage User Roles and Permissions

1. Participating Actor: Department Chair, System Administrator
2. Flow of Events:

   2.1 The actor navigates to the user management section.

   2.2 The actor selects a user to modify.

   2.3 The actor assigns or modifies roles and permissions.

   2.4 The system updates the user's access rights.

3. Entry Condition: The actor has administrator privileges.
4. Exit Condition: User's roles and permissions are updated.
5. Quality Requirements: The system should enforce role separation and least privilege.

Use Case Name: Send Personal Invitation

1. Participating Actor: Authorized Staff, Department Chair
2. Flow of Events:

   2.1 The actor navigates to the user management section.

   2.2 The actor selects the option to invite a new user.

   2.3 The actor enters the invitee's email and role.

   2.4 The system generates and sends an invitation.

3. Entry Condition: The actor has permission to invite users.
4. Exit Condition: Invitation is sent to potential users.
5. Quality Requirements: Invitations should have secure, time-limited links, with validity duration of 2 days. Invitees should be able to accept the invitation with a single step.

Use Case Name: New Coordinator/Exam or Same Day Assignment

1. Participating Actor: Department Chair, Authorized Staff
2. Flow of Events:

   2.1 The actor navigates to the assignments section.

   2.2 The actor creates a new coordinator position or same-day assignment.

   2.3 The actor specifies details and requirements.

2.4 The actor assigns a TA or leaves it open for application.

3. Entry Condition: The actor has appropriate permissions.
4. Exit Condition: New position/assignment is created.
5. Quality Requirements: The system should check for scheduling conflicts and the interface should clearly indicate open/filled positions.

Use Case Name: Manage/View Roles and Permissions

1. Participating Actor: Department Chair, System Administrator
2. Flow of Events:

   2.1 The actor navigates to the role management section.

   2.2 The actor can view existing roles and their permissions.

   2.3 The actor can create new roles or modify existing ones.

   2.4 The system updates the role configuration.

3. Entry Condition: The actor has administrator privileges.
4. Exit Condition: Roles and permissions are updated.
5. Quality Requirements: The system should prevent circular dependencies in role hierarchies. Only users with high-level permissions should modify roles.

2.7 TA Notifications Package

Use Case Name: Notification Email

Participating Actor: System (TA Management System), All Users
Flow of Events:

1. Triggered by system events (approvals, swaps, assignments).
2. The system composes a message, including context (e.g., approval, rejection, proctor assignment).
3. Sends an email via the Email API.
4. Displays a notification in-app.

Entry Condition: A triggering event occurs (e.g., TA swap approval, new assignment).
Exit Condition: The email is delivered, and the notification is shown in the UI.

Quality Requirements:

- Email must be sent within 3 seconds of the event.
- Messages must include proper details and links.
- Users must be able to mark as read.

2.8 TA Assignment Package

Use Case Name: Swap Proctors

Participating Actor: Authorized Staff
Flow of Events:

1. Authorized staff selects a proctoring duty.
2. Initiates swap between eligible TAs.
3. Validates availability and assignment constraints.
4. Confirms swap, and system updates records.

Entry Condition: The user is logged in and has assignment privileges.
Exit Condition: Proctoring duty is swapped.

Quality Requirements:

- Prevent conflicts or workload limit violations. Send real-time notifications to both TAs.

Use Case Name: Override Restrictions

Participating Actor: Authorized Staff
Flow of Events:

1. Staff attempts to assign TA to an already full slot.
2. System detects conflict and prompts for override.
3. Staff confirms override, optionally adds justification.
4. Assignment is forced through.

Entry Condition: The selected assignment violates TA restriction(s).
Exit Condition: Assignment is saved, restriction logs updated.

Quality Requirements:

- Only high-permission users may override. Must log reason and timestamp.

Use Case Name: Assign Automatically

Participating Actor: Department Chair / Authorized Staff
Flow of Events:

1. Actor selects “Auto-Assign” for a course.
2. System runs assignment logic (availability, workload, preferences).
3. Assigns TAs and displays results.
4. Actors can accept, revise, or override assignments.

Entry Condition: Course offering is created, unassigned TAs exist.
Exit Condition: TA assignments are generated.

Quality Requirements:

- Must finish within 5 seconds. All assignments must satisfy constraints unless overridden.

Use Case Name: View Weekly Lecture Schedule

Participating Actor: TA
Flow of Events:

1. TA navigates to their schedule.
2. System fetches upcoming lectures, exams, and duties.
3. Displays schedule on calendar view.

Entry Condition: TA is authenticated.
Exit Condition: Weekly calendar is shown.

Quality Requirements:

- Should load in under 1 second. Must highlight overlapping/conflicting duties.

2.9 Exam & Distribution Management

Use Case Name: Print List (Alphabetical/Randomized)

Participating Actor: Authorized Staff

Flow of Events:

1. Actor selects a course or exam.

2. Chooses between alphabetical or randomized print order.

3. System fetches and formats the student list.

4. Exported as printable PDF.

Entry Condition: A course or exam with enrolled students exists.
Exit Condition: A PDF is generated for print.

Quality Requirements:

- Generation time is under 3 seconds and output must contain headers and be paginated.

Use Case Name: View Student Distribution for an Exam

Participating Actor: Authorized Staff, Instructor
Flow of Events:

1. Actor selects an upcoming exam.

2. System displays student distribution across classrooms.

3. Includes sectioning, seating arrangements, and proctoring.

Entry Condition: An exam is scheduled and students are assigned.

Exit Condition: Distribution layout is shown.

Quality Requirements:

- Load under 2 seconds.
- Allow export to PDF/Excel.
- Group by classroom or section.

Use Case Name: View Scheduled Exams

Participating Actor: Authorized Staff, Instructor

Flow of Events:

1. Actor navigates to the exams section.
2. System lists all exams with date/time, course, location, and proctors.
3. Actor may click into an exam to view full details.

Entry Condition: Actor is authenticated.

Exit Condition: List of scheduled exams is displayed.

Quality Requirements:

- Pagination or infinite scroll if >20 exams.

- Filter by date, course, or room.

2.10 Access Control

Use Case Name: Set Current Semester

Participating Actor: Admin

Flow of Events:

1. Admin selects "Set Semester" in the dashboard.

2. Picks semester code (e.g., Spring 2025).

3. System updates all semester-based filters and logic.

Entry Condition: Admin is logged in.

Exit Condition: Semester context is updated.

Quality Requirements:

- Must not disrupt live session filters.

- Confirmation should appear in under 2 seconds.

2.11 Data Access & Table Viewing

Use Case Name: View General Tables

Participating Actor: All Roles (Admin, Staff, Instructor)
Flow of Events:

1. Actor selects “General Tables” from the sidebar.
2. System loads tables such as Course Catalog, Department List, Room List, etc.
3. Actor can sort, filter, or download.

Entry Condition: Actor is logged in.
Exit Condition: Tables are rendered and interactable.
Quality Requirements:

- Must support column filtering.
- Load under 2 seconds for up to 500 rows.

Use Case Name: View Private Tables

Participating Actor: Authorized Staff, Department Chair
Flow of Events:

1. Actor selects "Private Tables" such as TA evaluations, workload audits, etc.
2. System fetches restricted data.

3. Actor can view or export based on permissions.

Entry Condition: Staff role has permission.
Exit Condition: Private data is shown in table form.
Quality Requirements:

- Must show "Access Denied" if unauthorized.
- Encryption at rest for sensitive rows.

2.12 Data Import

Use Case Name: Import TAs

Participating Actor: Admin
Flow of Events:

1. Admin uploads file with TA data (e.g., name, email, program).
2. System validates and parses file.
3. TA users are created and stored.

Entry Condition: A valid .xlsx file is selected.
Exit Condition: TA users are added to system.
Quality Requirements:

- Must validate required fields (email, ID).

- Duplicate check must run automatically.
- Allow max 500 TAs per import.

Use Case Name: Import Staff

Participating Actor: Admin

Flow of Events:

1. Admin uploads staff details (name, email, department, role).
2. System validates, assigns role, and creates staff user.

Entry Condition: File with staff data is uploaded.

Exit Condition: Staff user accounts are created.

Quality Requirements:

- Only valid roles (Instructor, Authorized, Chair) allowed.
- Confirmation email must be sent.

Use Case Name: Import Excel Data

Participating Actor: Admin

Flow of Events:

1. Admin uploads a combined Excel file (courses, offerings, users).
2. System detects and splits relevant parts.
3. Runs import logic for each component.

Entry Condition: Excel file is uploaded.

Exit Condition: Course, TA, and staff data are imported.

Quality Requirements:

- Detects file type by header format.
- Reports success/failure line-by-line.
- Upload must complete in under 10 seconds for 1000+ rows.

Conclusion: We designed each of these packages to organize related functionality within the TA Management System, making the system more modular and easier to understand.

3. Tech Stack

Backend:

Initially, we planned to use Flask as our main backend framework, but we switched to **Django** because it offers a more structured framework with built-in features like authentication, **ORM**, and an admin panel, reducing reliance on third-party integrations and enforcing security. For handling "Excel-based data imports," we consider using the **Pandas** library, which provides powerful tools for reading and analyzing spreadsheet data. Although not currently implemented, external tools such as **Google OR-Tools** might be integrated in the future to manage "proctoring restrictions," as it supports advanced constraint programming. Thus, these enhancements will allow the system to scale efficiently and adapt to more complex assignment rules as institutional needs evolve.

Frontend:

The frontend of the application will be developed using **React**, within a **Next.js** framework for server-side-rendering and routing capabilites. The project uses **Typescript** to ensure type safety and catch errors during development, and **Tailwind CSS** for utility-first, responsive and maintainable styling. For UI design, we are leveraging the **Shadcn** library, which provides a collection of accessible and customizable UI components. Moreover, React's component-based architecture allows for modular and reusable code, facilitating easier maintenance and scalability. Combined, this stack enables rapid development, a better developer experience, and a modern, performant user interface.

Database:

For data management, we will use **MySQL**. It offers efficient handling of complex queries, supports relational data structures, and ensures data integrity through constraint mechanisms. MySQL is also well-suited for working with data imported from Excel files, making the process of importing, updating, and managing structured data straightforward and reliable.

Version Control:

Our development workflow will be managed using **Git**. Git enables precise tracking of all changes, supports collaborative development through branching and merging, and allows for integration of code across team members. It is a widely adopted version control system that the entire team is comfortable working with, making it a practical and reliable choice for managing our project.