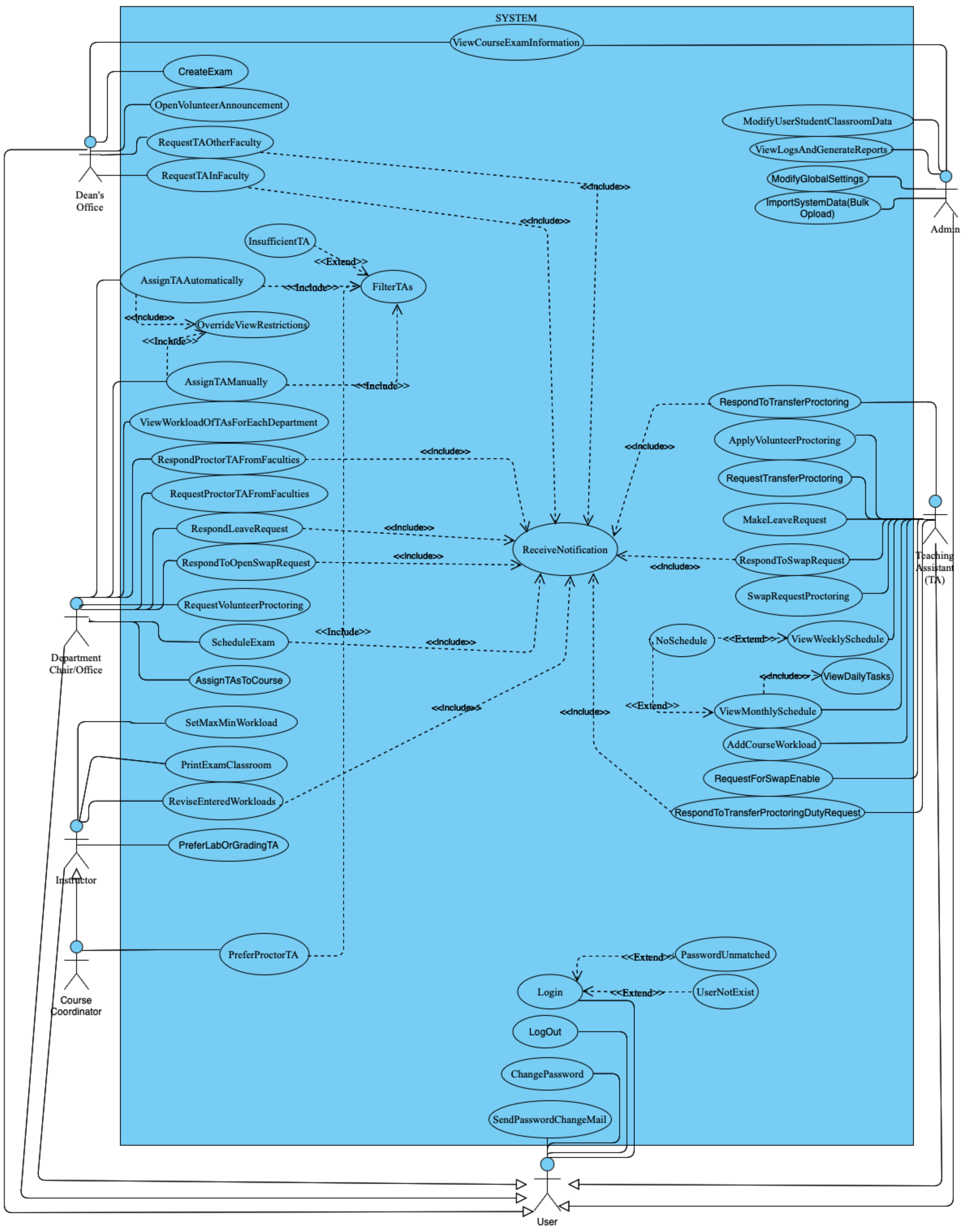
BILKENT UNIVERSITY
COMPUTER SCIENCE DEPARTMENT
CS 319 OBJECT-ORIENTED SOFTWARE ENGINEERING
DELIVERABLE 1 ITERATION 2
SPRING 2025
GROUP 2 SECTION 1
08.04.2025

Full Name	ID
Perhat Amanlyyev	22201007
Emiralp İlgen	22203114
Anıl Keskin	22201915
İlmay Taş	22201715
Simay Uygur	22203328

1.USE CASE DIAGRAM



The link: visual_paradigm

2. USE CASE TEXTUAL DESCRIPTION

1.
Unique name: Login
Participating actors: All actors
Entry conditions: The user entered the correct ID and password.
Exit conditions: The user successfully logs in and is directed to the main page.
Flow of events:

1. The user enters their ID and password in separate provided blocks.
2. The user clicks the login button.
3. The user enters successfully and is directed to the main page.

Extensions:

- If the password is incorrect, execute “PasswordUnmatched”. An error message that explains the situation will be displayed and the user will be directed to the same login page.
- If the user account does not exist, execute “UserNotExist”. An error message that explains the situation will be displayed and the user will be directed to the same login page.

Special/quality requirements: Session timeout after 15 minutes of inactivity (aligned with GDPR). Password complexity enforced (e.g., 8+ characters, special symbols).

2.
Unique name: SendPasswordChangeMail
Participating actors: All Actors
Entry conditions: The user clicks “Forgot Password?” on the login page.
Exit conditions: An email containing a link to reset the password is sent successfully.
Flow of events:

1. The user forgets the password.
2. The user clicks the “Forgot Password?”
3. The user is navigated to the page where the User should write his email.
4. The system checks the correctness of the email.
5. The system generates a unique link for resetting the password.
6. The system sends the mail to the given email.

Special/quality requirements: Password reset links must expire after 15 minutes. Email delivery latency ≤5 seconds (critical for user trust).

3.
Unique name: LogOut
Participating actors: All Actors
Entry conditions: The user is currently logged into the system.

Exit conditions: The user session is terminated, and the user is redirected to the login page.
Flow of events:

1. The user clicks the “Log Out” button or link.
2. The system invalidates the current session token.
3. The system redirects the user to the login page (or home page if configured).

Special/quality requirements: Session tokens are invalidated in 5 seconds (no reuse allowed).

4.
Unique name: RespondToSwapRequest
Participating actors: Teaching Assistant(TA)
Entry conditions: TA gets the notification for swapping proctoring from another TA.
Exit conditions: TA either approves or declines the request successfully.
Flow of events:

1. TA gets the notification for swapping.
2. TA decides to approve or decline and clicks respectively.

Special/quality requirements: Swap operations shall be executed atomically, ensuring that both TAs’ schedules are either fully updated or completely rolled back, and real-time notifications shall be delivered via WebSocket within 2 seconds of any related event.

5.
Unique name: ViewMonthlySchedule
Participating actors: Teaching Assistant (TA)
Entry conditions: TA requests to view his monthly schedule, and the system navigates the user to the schedule page. TA’s course schedule must be present in the system.
Exit conditions: The monthly schedule is displayed.
Flow of events:

1. The user clicks the monthly schedule button.
2. The system retrieves the data from the database.
3. The monthly schedule for the current and next month is displayed to the TA.

Extension:

- If the system finds no scheduled tasks, it triggers "No Schedule". An empty page with the message explaining that the courses are not yet loaded.

Special/quality requirements: Schedule loading time ≤1 second (caching for frequently accessed data).

6.
Unique name: ViewWeeklySchedule

Participating actors: Teaching Assistant (TA)
Entry conditions: TA requests to view his weekly schedule, and the system navigates the user to the schedule page.
Exit conditions: The weekly schedule is displayed.
Flow of events:

1. The user clicks the schedule button.
2. The system retrieves the data from the database.
3. The schedule is displayed to the TA. The weekly course schedule and labs will be displayed. Exam proctoring sessions will not be shown.

Special/quality requirements: Schedule loading time ≤1 second (caching for frequently accessed data).

7.
Unique name: ViewDailyTasks
Participating actors: Teaching Assistant (TA)
Entry conditions: The user is logged in and has assigned tasks or proctoring duties for the day.
Exit conditions: The user views all tasks scheduled for the current day.
Flow of events:

1. The user clicks on a specific day in the monthly schedule.
2. The system retrieves tasks (lectures, labs, proctoring, office hours) from the database.
3. The system displays the list of tasks with time and location.
4. The user can click each task for more details.

8.
Unique name: ViewLogsAndGenerateReports
Participating actors: Admin
Entry conditions: Admin requests to see the logs by clicking the Log button.
Exit conditions: Close the page by clicking the main page or another button.
Flow of events:

1. The user clicks the Logs button on the Admin main page.
2. The page is directed to a new page containing all logs of the processes done by all actors.
3. By clicking the courses, a course can be selected and the related workload of that course can be seen also it can be filtered by semester, student, etc and its report can be made in PDF and printed.
4. By clicking the proctoring duties, all proctoring logs can be seen.
5. The page can be closed by clicking the main page.

9.
Unique name: MakeLeaveRequest
Participating actors: Teaching Assistant (TA)
Entry conditions: TA is active and has tasks in the current semester. TA wants to report his/her leave for a long time or a few days for exam duty.
Exit conditions: Request is sent to the Department Office.
Flow of events:

1. The user clicks to fill out the leave report.
2. The user enters the start date and end date for their leave and they can select whether it is permanent or temporary.
3. The user can attach the file (PDF or any image type) of their medical report or reason for their leave.
4. The user can write a message that can explain their causes in the text.
5. The submit button is clicked and the form that is filled is sent to the Department Office.

Special/quality requirements: TA medical reports (attached to leave requests) must be encrypted and automatically deleted 6 months after approval.

10.
Unique name: RespondLeaveRequest
Participating Actors: Department Chair/Office
Entry Conditions: A leave request has been submitted.
Exit Conditions: Request is either approved or rejected.
Flow of Events:

1. The actor accesses all leave requests on a page.
2. The system displays pending requests.
3. The actor reviews the request details, attached file, and excuse message that is written by the TA.
4. The actor selects approval or rejection. If approved, the system will check for date conflicts with exams/proctoring duties and will show warning about the conflicts to the actor and at the same time TA will be removed from those tasks and another TA can be assigned to these duties.
5. The system updates the request status and notifies the TA about the update.

11.
Unique name: CreateExam
Participating actors: Dean's Office
Entry conditions: The uncertainty of the exams during that period.
Exit conditions: The scheduling of the exams.
Flow of events:

1. Determines the exams to be held, their number, and their timings.
2. After clicking done, all information is saved in the database and can be seen by the instructors and the department’s office.

12.
Unique name: PreferProctorTA
Participating actors: Course Coordinator
Entry conditions: The occurrence of that course's exam is determined by the Dean’s Office. The course coordinator has that course offered and wants to have preference over some TAs of his department.
Exit conditions: The sending of the message.
Flow of events:

1. The number of TAs is specified by the course coordinator, and if TAs are available, preferred and non-preferred TAs are indicated.
2. The button to send the preferences is clicked and that information is saved in the database and can be seen by the Department’s office.

13.
Unique name: FilterTAs
Participating actors: Instructor, Department Chair/Office
Entry conditions: For the instructor, the TA preferring button is clicked and TA course instructors are from the same department. For the department office/chair, automatically or manually assigning proctoring is clicked.
Exit conditions: All restrictions are determined and TAs aligning those are listed.
Flow of events:

1. The instructor selects the proctoring duties of his courses. The department’s office selects the proctoring duties of the faculty.
2. The instructor selects the amount that this duty needs and he can flag the TAs of their department to show which are more preferred and less preferred. The department’s office selects manual assignments.
3. Both ways of assigning TAs are filtered by the restrictions and preferences of instructors.

Extension:

- If there is not enough TAs that satisfy the conditions “InsufficientTA” is triggered. A pop up will be displayed that explains there is not enough TAs and asking the user whether they want to override restrictions. If yes, all restrictions applied will be displayed so that actor can select to eliminate among restrictions.

14.
Unique name: AssignTAManually

Participating actors: Department's Office/Chair
Entry conditions: Clicking the manually assign button of that department's proctoring duties.
Exit conditions: Clicking done in the assigning page and duties assigned to each selected TA without conflict.
Flow of events:

1. The exam proctoring is selected.
2. After selecting the necessary restrictions, the TAs are listed.
3. Department Office/Chair gets the TAs request considering PreferProctorTA and PreferLabOrGradingTA and manually chooses TAs from the list.
4. After choosing the TAs Department Office/Chair approves the decision.
5. The assigned TAs' workload is updated for proctoring and schedules of TAs' are updated.
6. Notifications are sent to all relevant parties.

15.
Unique name: RequestProctorTAFromFaculties
Participating actors: Department Chair/Office
Entry conditions:

1. There are unassigned proctoring duties for an exam.
2. The Department Office has determined that there are not enough available TAs within their department or for other reasons.

Exit conditions:

1. The request for additional TAs is sent to the Dean's Office.
2. The Dean's Office forwards the request to other departments.

Flow of events:

1. The Department Office sends a request to the Dean's Office for additional TAs.
2. The Dean's Office receives the request and determines how many TAs are needed.
3. The Dean's Office forwards the request to relevant departments.
4. Other departments receive the request and review availability.
5. If other departments accept, they begin searching for available TAs.
6. The Dean's Office monitors the process and updates the requesting department.

16.
Unique name: AssignTAAutomatically
Participating actors: Department Chair/Office
Entry conditions:

1. An exam requires proctoring, or TAs should be assigned for grading and general tasks of the course.

2. The system attempts to find the required number of available TAs.

Exit conditions:

1. TAs are successfully assigned to the exam, grading, or other tasks.
2. Classrooms for the exam are recorded for proctoring.
3. Notifications are sent to all relevant parties.
4. The assigned TAs' workload is updated for proctoring and schedules of TAs' are updated.

Flow of events:

1. The system searches for available TAs based on predefined rules.
2. If enough TAs are found, proceed to the assignment.
3. If not enough TAs are available, the system prompts the user to override restrictions (e.g., consecutive days, MS/PhD limits).
4. If still insufficient, the system sends a request to other departments via the Dean's office for proctoring.
5. Once enough TAs are secured, the system requests assignment details.
6. After confirmation, the system finalizes assignments.
7. Automatic email notifications are sent to TAs, the coordinator, and relevant staff.
8. The system updates the workload of assigned TAs.

Special/quality requirements: Assignment completed in ≤30 seconds for 500+ TAs. The algorithm handles 1,000+ concurrent assignments during finals.

17.

Unique name: OpenVolunteerAnnouncement

Participating actors: Dean's Office

Entry conditions: A request for more TAs is sent by the Department's office/chair.

Exit conditions: Volunteer announcement made and becomes open for requests by TAs.

Flow of events:

1. The request of how many TAs are displayed in the Dean's office.
2. Create a new volunteering announcement by selecting the departments that can see those.
3. Announcement becomes only available to the allowed departments' TAs.

18.

Unique name: ApplyVolunteerProctoring

Participating actors: Teaching Assistant (TA)

Entry conditions: Volunteer proctoring applications must be open and the TA wants to apply for an available volunteer proctoring request.

Exit conditions: The volunteer proctoring application is successfully submitted to the Dean's Office.

Flow of events:

1. The actor views the open volunteer proctoring request.
2. The actor clicks on the apply option.
3. The application is submitted to the Dean's Office.

19.
Unique name: RequestTAOtherFaculty
Participating actors: Dean's Office
Entry conditions: The Department Office must have previously applied to the Dean's Office to request volunteer TAs from other faculties if no one is found within the faculty and there are still unassigned proctoring duties for the exam.
Exit conditions: The request is forwarded to the other faculty.
Flow of events:

1. A message is sent indicating from which department and how many TAs are requested from other faculties.
2. If the other faculties accept, the relevant departments announce the exam proctoring duty as a voluntary position, similar to a job posting, to find volunteers.

20.
Unique name: RequestTAInFaculty
Participating actors: Dean's Office
Entry conditions: The Department Office must have previously applied to the Dean's Office to request TAs from other departments if no one is found within the department and there are still unassigned proctoring duties for the exam.
Exit conditions: The request is forwarded to the other departments in the same faculty.
Flow of events:

1. A message is received indicating how many TAs are requested from the relevant departments.
2. If other departments accept, the relevant departments search for people for the exam proctoring duty.

21.
Unique name: ReviseEnteredWorkloads
Participating actors: Instructor, Course Coordinator
Entry conditions: The instructor has previously entered workloads for courses and wants to make changes.
Exit conditions: Workloads are successfully revised and saved.
Flow of events:

1. The instructor selects the workload revision option.
2. The system displays the previously entered workloads.

3. The instructor modifies the workloads.
4. The system validates and updates the workloads.
5. The confirmation screen is displayed to the instructor and returned to the workload lists.

22.
Unique Name: ModifyUsersStudentsClassrooms
Participating Actors: Admin
Entry Conditions: The admin wants to update information related to users, students, or classrooms. The system must have existing records for the selected entity.
Exit Conditions: The selected user, student, or classroom information is successfully updated and saved in the system.
Flow of Events:

1. The admin selects an entity to modify (User, Student, or Classroom).
2. The system displays the current details of the selected entity.
    a. If a user is selected, the system shows role, department, and access permissions.
    b. If a student is selected, the system displays course enrollments, academic status, and personal details.
    c. If a Classroom is selected, the system shows capacity, assigned courses, and available time slots.
3. The admin modifies the necessary fields.
4. The system validates the changes to prevent invalid updates.
5. The updated information is saved.
6. A confirmation message is displayed.

23.
Unique name: SetWorkloadLimit
Participating Actors: Instructor
Entry Conditions: The workload limit is subject to change.
Exit Conditions: Maximum and minimum workloads are successfully set.
Flow of Events:

1. Workload settings are selected by the user.
2. The system displays current limits.
3. The instructor adjusts limits.
4. The system saves and applies the new values.
5. Confirmation is displayed.

24.
Unique name: ViewWorkloadOfTAsForEachDepartment

Participating Actors: Department Chair/Office
Entry Conditions: The actor needs workload data.
Exit Conditions: Workload data is displayed.
Flow of Events:

1. Staff navigates to the workload section.
2. The system retrieves and displays workload information.

25.
Unique name: ViewTA_IBAN
Participating actors: Department Chair/Office
Entry conditions: The Teaching Assistant(TA) has entered IBAN details.
Exit conditions: IBAN details are displayed.
Flow of events:

1. The actor selects a TA from the listed ones.
2. The system displays stored IBAN details.

Special/quality requirements: IBAN details must be stored with AES-256 encryption.

26.
Unique name: ChangePassword
Participating actors: All Actors
Entry conditions: The user wants to change their password.
Exit conditions: Either the password is successfully updated or no change is made due to failures or a change of mind.
Flow of events:

1. The user enters a current password.
2. The user enters a new password.
3. The system validates and either updates the password or not.

Special/quality requirements: Password history check (no reuse of last 3 passwords).

27.
Unique name: AddCourseWorkload
Participating actors: Teaching Assistant(TA)
Entry conditions: TA needs to submit workload details.
Exit conditions: The workload form is either submitted or the mission is aborted.
Flow of events:

1. The TA fills out the workload form.
2. The system validates and saves it.

28.
Unique name: RequestTransferProctoring

Participating actors: Teaching Assistant(TA)
Entry conditions: The TA wants to transfer their proctoring duty to another TA having less than 20 workload hours.
Exit conditions: The request to transfer the proctoring duty has been successfully submitted.
Flow of events:

1. TA selects another TA to take over the proctoring duty.
2. TA submits the transfer request.
3. The system records the request and forwards it to the other TA.

Special/quality requirements: Notifications for requests must be delivered in ≤5 seconds via WebSocket with email fallback to ensure timely user awareness and system responsiveness.

29.
Unique name: RespondToTransferProctoring
Participating actors: Teaching Assistant(TA)
Entry conditions: The TA receives a notification regarding a request to transfer proctoring duty from another TA.
Exit conditions: The TA either approves or declines the request.
Flow of events:

1. The TA receives a notification about the proctoring transfer request.
2. The TA decides whether to approve or decline the request and clicks the respective option.

30.
Unique name: ViewCourseExamInformation
Participating actors: Admin, Dean’s Office
Entry conditions:

1. The actor is logged into the system.
2. The actor requests listed exams.
3. The actor selects one of the provided exams.

Exit conditions: The requested course exam information is displayed.
Flow of events:

1. The actor (or authorized staff) navigates to the “Exam Management” section.
2. The system displays a list of available courses or exam offerings.
3. The actor selects a specific course/exam to load.
4. The system asks for the necessary exam information (date, time,
5. The actor is authorized to make changes (add/edit/delete):
    a. The actor can edit the exam details (e.g., date, time, assigned TAs)
    b. The actor can delete an exam record or create a new one if needed.

6. The actor confirms or cancels any changes.
7. The system saves the changes (if confirmed) and returns to the exam management overview.

31.
Unique name: OverrideViewRestrictions
Participating actors: Department Chair/Office
Entry conditions:

1. The system has been unable to find enough available TAs due to restrictions (e.g., PhD/MS level restrictions, leave status, course enrollment, and exam conflicts).
2. The user (admin or exam coordinator) is prompted to review and potentially override these restrictions.

Exit conditions:

1. The user either approves the override of one or more restrictions or cancels the request.
2. The system updates the TA assignment process based on the user's decision (allowing overrides).

Flow of events:

1. The system attempts to assign TAs to the exam but encounters restrictions (e.g., only PHD students are available for MS/PHD courses, or TA conflicts with exam schedules).
2. The system notifies the user (admin or coordinator) that there are not enough available TAs due to these restrictions.
3. The user is presented with options to override the restrictions, including
    a. Allowing TAs with consecutive or same-day assignments.
    b. Breaking MS/PHD restrictions to assign MS students to MS/PHD level courses.
    c. Sending requests to other departments via the Dean’s office for additional TAs.
4. The user chooses whether to apply these overrides.
5. If the overrides are accepted, the system proceeds with the assignment process using the newly relaxed restrictions.
6. If the overrides are rejected, the system either retries with stricter criteria or notifies the user that no assignments were made.

32.
Unique name: ReceiveNotification
Participating actors: All Actors
Entry conditions:

1. The user is logged into the system.
2. There are pending notifications for the user.

Exit conditions: The notification is marked as read or remains pending if the user closes it.

Flow of events:

1. The user clicks on the “Notifications” icon or link.
2. The system retrieves unread and reads notifications from the database.
3. The user selects a notification to view its details.
4. Some notifications may direct the user to take action depending on their role.
5. The system marks the notification as read.
6. The user can close or delete the notification.

Special/quality requirements: Notifications must delivered in ≤3 seconds (real-time system).

33.

Unique name: RequestVolunteerProctoring

Participating actors: Department Office/Chair

Entry conditions:

1. An exam is scheduled, and not enough TAs are available to proctor.

Exit conditions: The department asks the Dean’s Office for volunteer proctoring of other departments.

Flow of events:

1. The actor opens the exam management page.
2. The actor selects “Request Volunteer Proctoring.”
3. The system prompts for the number of TAs needed.
4. The system sends the volunteer request to the Dean’s Office.
5. A confirmation message is displayed to the user.

34.

Unique name: PrintExamClassroom

Participating actors: Instructor, Course Coordinator

Entry conditions: The user opens the page with listed exams and clicks the printing option.

Exit conditions: The user downloads the PDF of that classroom.

Flow of events:

1. The user selects a classroom and clicks the Print option.
2. The system retrieves the relevant classroom exam details from the database.
3. The system generates a PDF containing the exam details for that classroom.
4. The system provides a Download button for the user.
5. The user downloads the PDF.

Special/quality requirements: PDF generation ≤10 seconds for 200+ students (async processing).

35.
Unique name: PreferLabOrGradingTA
Participating actors: Instructor
Entry conditions: Being an instructor with a course that brings a workload like a lab or grading.
Exit conditions: Approve the preference list.
Flow of events:

1. The number of TAs and type of work (lab or grading) specified, and if TAs are available, preferred and non-preferred TAs are indicated.
2. The button to send the preferences is clicked and that information is saved in the database and can be seen by the Department's office.

36.
Unique Name: ScheduleExam
Participating Actors: Department Chair/Office
Entry Conditions:

1. Courses and exams must be available.

Exit Conditions:

1. All exams must be successfully planned and scheduled.
2. Exam halls and proctors must be assigned.

Flow of Events:

1. Exam dates and times are planned for each course.
2. The prepared exam schedule is approved.

37.
Unique name: RequestForSwapEnable
Participating actors: Teaching Assistant (TA), Department Chair/Office
Entry conditions:

1. A TA is assigned to an exam proctoring duty.
2. The TA wants to swap their duty with another TA but is not authorized to do so directly.

Exit conditions:

1. The department chair/office approves or rejects the request to enable the swap option for the TA.
2. If approved, the swap option is activated, allowing the TA to request the swap.

Flow of events:

1. The TA attempts to initiate a swap with another TA but is unable to do so due to restrictions.
2. The TA submits a request to the department chair/office to open the swap option.
3. The system forwards the swap request to the department chair/office for review.
4. The department chair/office reviews the request and decides whether to approve or reject it.
5. If approved, the system allows the TA to swap the duty with another TA.
6. If rejected, the TA is notified, and no swap option is made available.

38.
Unique name: RespondToOpenSwapRequest
Participating actors: Department Chair/Office
Entry conditions:
1. A TA has submitted a request to swap their proctoring duty with another TA, but they do not have the authority to initiate the swap.
2. The department chair/office has received the swap request and must approve or reject it.

Exit conditions:
1. The department chair/office has responded to the request, either approving or rejecting it.
2. If approved, the system enables the TA to swap duties with another TA.
3. If rejected, the TA is notified, and the swap request is not granted.

Flow of events:
1. The TA submits a swap request to the department chair/office.
2. The department chair/office receives the request and reviews it.
3. The department chair/office decides whether to approve or reject the request.
4. If the request is approved, the system activates the swap functionality, allowing the TA to proceed with swapping duties with another TA.
5. If the request is rejected, the department chair/office notifies the TA that the swap request has been denied, and no changes are made to the assignment.

39.
Unique name: RespondTAFromFaculties
Participating actors: Department Chair/Office
Entry conditions:
1. The RequestTAInFaculty has been sent by the Dean's Office to other departments requesting TAs for proctoring duties.
2. The relevant departments have reviewed the request and are ready to respond.

Exit conditions:
1. The other departments have either accepted or rejected the request.

2. The system updates the status of the request, notifying the Dean’s Office of the decision.

Flow of events:

1. The system forwards the request for TAs to the relevant departments.
2. The departments review the request for TAs and check their available pool of proctors.
3. Each department decides whether they can provide the required TAs.
    a. If a department accepts, they begin searching for available TAs.
    b. If a department rejects, they notify the Dean's Office that they cannot fulfill the request.
4. The system updates the status of the request and notifies the Dean’s Office of the decisions made by the departments.
5. The Dean's Office can take further actions (e.g., seeking additional TAs from other departments or adjusting the request).

Special/quality requirements: Auto-reminders for pending requests after 24 hours.

40.

Unique name: SwapRequestProctoring

Participating actors: Teaching Assistant (TA)

Entry conditions: The TA is assigned to at least one exam duty. The TA wants to see existing or potential swaps. The TA requests the system to swap proctoring duties with another TA. The system navigates to the list of proctoring duties page.

Exit conditions: TA sends the request for swapping to another TA successfully. The system displays the swap request interface.

Flow of events:

1. TA clicks the swap list button and is directed to another page
2. The system displays only TAs available during the exam time and satisfying the exam requirements and also the proctoring with the pending status of the old request if no response is given.
3. TA can choose from the options.
4. The system sends a request to the chosen TA for approval.

Special/quality requirements: Notifications for requests must be delivered in ≤5 seconds via WebSocket and email. Also, during final exams, 300 TAs simultaneously request swaps. The system must handle ≥50 swap requests/minute without downtime (load-balanced backend).

41.

Unique name: ModifyGlobalSettings

Participating actors: Admin

Entry conditions: The global system settings exist in the database (e.g., current semester, TA workload limits, proctoring restrictions). The admin wants to update or modify one or more system settings.
Exit conditions: The selected global parameters are successfully updated in the system. The system logs the changes for auditing purposes. If required, notifications are sent to affected users (e.g., if the current semester is updated, instructors and TAs are notified).
Flow of events:

1. The admin navigates to the System Settings page.
2. The system displays editable global parameters (e.g., current semester, TA workload limits, proctoring restrictions).
3. The admin modifies the required settings.
4. The system validates the changes and checks for potential conflicts (e.g., ensuring that workload limits do not exceed system constraints).
5. The admin confirms the changes.
6. The system saves and applies the updated settings.
7. The system logs the changes for security and tracking.

42.
Unique name: ImportSystemData (Bulk Upload)
Participating actors: Admin
Entry conditions: The admin is logged in and has the necessary permissions. The Excel file containing student, faculty, or course data follows the required format. The system has existing records that may need updating or new records that need to be added.
Exit conditions: The data is successfully processed and added/updated in the system. The system validates the uploaded file, ensuring correct formatting and preventing duplicates or conflicts. The system logs the import process for tracking and troubleshooting.
Flow of events:

1. The admin navigates to the Data Import section.
2. The system prompts the admin to upload an Excel file.
3. The admin selects and uploads the Excel file containing new or updated data.
4. The system validates the file to ensure it follows the correct format, structure, and data integrity rules.
5. If the file is valid, the system parses and processes the data:
    a. New records are added to the database.
    b. Existing records are updated as needed.
6. If the file contains errors, the system generates an error report and prompts the admin to fix the issues.

7. Once successful, the system logs the import activity for future reference.

Special/quality requirements: Bulk uploads validated for schema compliance (e.g., no duplicate TA IDs). The system shall complete bulk import operations of up to 10,000 records within 10 seconds.

43.
Unique name: AssignTAsToCourse
Participating actors: Department Chair/Office
Entry conditions: A course requires TAs and TAs should be assigned for any assignments that are to be made throughout the semester.
Exit conditions: TAs (randomly or the preferred ones by instructors, if any) are successfully assigned to courses meeting the desired number. Notifications are sent to all relevant parties.
Flow of events:

1. The system searches for available TAs based on predefined rules.
2. If enough TAs are found, the Department Chair will assign TA(s) to that course.
3. If not enough TAs are available, the system prompts the user to override restrictions (e.g., consecutive days, MS/PhD limits).
4. If still insufficient, the system sends a request to other departments via the Dean’s office for TAs.
5. Once enough TAs are secured and assignments are made, the system now allows viewing course details showing assigned TAs.
6. After confirmation, the system finalizes assignments.
7. Automatic email notifications are sent to TAs, the coordinator, and relevant staff.

Special/quality requirements: Assignment completed in ≤30 seconds for 500+ TAs. The algorithm handles 1,000+ concurrent assignments during finals.

3. TECH STACK

Backend: Spring Boot

- Fast and Easy Development: Minimal configuration, allowing rapid development.
- Microservice Compatibility: Ideal for modular and scalable applications.
- Robust Ecosystem: Integrates well with Spring Security, Spring Data, and Spring Cloud.
- REST API Support: Provides structured and efficient RESTful services.
- Easy Integration: Compatible with databases like PostgreSQL and MySQL.

Frontend: React.js + TypeScript

- Component-Based Architecture: Enables modular and reusable UI components.
- Rich Ecosystem: A vast collection of libraries accelerates development.
- High Performance: Virtual DOM ensures efficient rendering.
- TypeScript Support: Enhances type safety, debugging, and code maintainability.

Why Do We Use Apache Tomcat?

- It is a lightweight and reliable Java-based web server.
- It can integrate with the Apache HTTP Server if needed.
- It is included by default in Spring Boot as an embedded server, requiring no additional configuration.

How They Work Together

- React Frontend Deployment: The built React application (static files) will be hosted on Apache Tomcat.
- Backend Communication: API requests from the front end will be proxied through Apache Tomcat to the Spring Boot backend.
- Security and Optimization: SSL management, caching, and request forwarding will be handled by Apache Tomcat.

This architecture ensures that the Spring Boot and React integration provides a scalable, high-performance, and secure deployment strategy.