Use Case Diagram Textual Description

Use Case: Request TAs From Dean For Proctoring

1. Name: Request TAs From Dean For Proctoring
2. Participating actor: Instructor
3. Entry Condition:
   - The instructor has determined there are not enough available TAs for proctoring assignments.
4. Exit Condition:
   - The request for additional TAs has been submitted to the department dean's office.
   - The request has been logged for further reviews of activities.
5. Flow of Events:
   1. The instructor identifies a shortage of available TAs for proctoring.
   2. The instructor fills out and submits a request for additional TAs to the dean's office.
   3. The dean's office reviews the request and processes it based on availability.
   4. The instructor receives confirmation or denial of the request status.
6. Special / Quality Requirements:
   - Requests should be properly logged and tracked.
   - The request process should be efficient to avoid delays.
   - Notifications should be sent to the instructor about the status of the request.

Use Case: View Task With Class Info

1. Name: View Task With Class Info

2. Participating actor: All actors
3. Entry Condition:
    - There are tasks assigned or scheduled for a specific actors and course.
4. Exit Condition:
    - The actor has successfully viewed the task details.
5. Flow of Events:
    1. The actor navigates to the task management section.
    2. The system displays a list of tasks that are visible to associated actor.
    3. The actor selects a task to view its details.
    4. The system displays task details, including:
        - Task name, type, and date
        - Responsible TAs (if assigned)
        - List of students attending the class
    5. The actors reviews the task details.
6. Special / Quality Requirements:
    - The system should provide an intuitive interface for easy navigation.
    - Task information should be up-to-date and accurate.
    - The system should support filtering and searching for specific tasks.

Use Case: Adding TAs to Blacklist

1. Name: Adding TAs to Blacklist
2. Participating actor: Instructor

3. Entry Condition:
    - The instructor identifies a TA they want to blacklist.
4. Exit Condition:
    - The TA is successfully added to the instructor's blacklist.
5. Flow of Events:
    1. The instructor navigates to the TA management section.
    2. The instructor selects a TA to be blacklisted.
    3. The instructor confirms the action.
    4. The system updates the TA’s status to blacklisted.
    5. When selecting TAs for proctoring, blacklisted TAs are marked in red.
    6. The dean’s office & secratary may also notice blacklisted TAs when reviewing TA assignments.
6. Special / Quality Requirements:
    - The system should ensure that blacklisted TAs are visually indicated.
    - The blacklist should be accessible only to authorized users.
    - The instructor should be able to manage their blacklist easily.

Use Case: Removing TAs from Blacklist

1. Name: Removing TAs from Blacklist
2. Participating actor: Instructor
3. Entry Condition:
    - There are blacklisted TAs in the system.
4. Exit Condition:
    - The TA is successfully removed from the instructor's blacklist.

5. Flow of Events:
    1. The instructor navigates to the TA blacklist management section.
    2. The instructor selects a TA and remove from the blacklist.
    3. The system updates the TA’s status
    4. The TA is no longer marked as blacklisted in future proctoring selections.
6. Special / Quality Requirements:
    - The system should provide a simple and secure process for removing TAs from the blacklist.
    - Changes should be reflected immediately in the proctoring selection process.
    - Only authorized users should be able to modify the blacklist.

Use Case: Viewing TA Status / Workload

1. Name: Viewing TA Status / Workload
2. Participating actor: Instructor
3. Entry Condition:
    - TAs workload data is available in the system for instructor.
4. Exit Condition:
    - Successfully viewed the workload details of TAs.
5. Flow of Events:
    1. The instructor navigates to the TA workload section.
    2. The system displays an option to view either:
        - The instructor’s own TAs

        - Other instructors’ TAs.

    3. The instructor selects a category.
    4. The system displays workload details, including:
        - Total working hours of the TAs
        - Tasks they have worked on
        - Comments related to their performance (if available)
    5. The instructor reviews the information.

6. Special / Quality Requirements:
    - Workload data should be accurate and updated in real-time.
    - The interface should be user-friendly for quick access to information.

Use Case: Viewing Other TAs

1. Name: Viewing Other TAs
2. Participating actor: Instructor
3. Entry Condition:
    - Authorization is granted to view other instructors’ TAs.
4. Exit Condition:
    - The instructor has successfully viewed the workload details of TAs assigned to other instructors.
5. Flow of Events:
    1. The instructor navigates to the TA workload section.
    2. The instructor selects the option to view other instructors’ TAs.
    3. The system verifies authorization.
    4. If authorized, the system displays the workload details of other instructors’ TAs.

    5. The instructor reviews the workload information.

6. Special / Quality Requirements:
    - The system should restrict access based on permissions.
    - Data should be displayed in a structured and readable format.
    - Privacy policies should be upheld when sharing TA workload details.

Use Case: Viewing Personal TAs

1. Name: Viewing Personal TAs
2. Participating actor: Instructor
3. Entry Condition:
    - The instructor has assigned TAs and their workload data in the system.
4. Exit Condition:
    - The instructor has successfully viewed the workload details of their own TAs.
5. Flow of Events:
    1. The instructor navigates to the TA workload section.
    2. The instructor selects the option to view their own TAs.
    3. The system retrieves and displays workload details, including:
        - Total working hours of each TA
        - Tasks they have worked on
        - Any related comments or feedback
    4. The instructor reviews the information and may take necessary actions.
6. Special / Quality Requirements:
    - The system should ensure that only the instructor’s own TAs are displayed.

   - Workload data should be accurate and updated in real-time.
   - The system should allow filtering based on time period or task type.

Use Case: Create Task (e.g., Exams, Quizzes, Recitations, etc.)

1. Name: Create Task
2. Participating actor: Instructor
3. Entry Condition:
   - The instructor has decided to create tasks (e.g. exams, quiz, recitations) according to the course curriculum.
4. Exit Condition:
   - The task has been successfully created and stored in the system.
   - Notifications and proctoring request may be sent to relevant TAs in case requested.
5. Flow of Events:

   1. The instructor navigates to the task creation section.
   2. The instructor selects or create the type of task (e.g., exam, quiz, recitation).
   3. The instructor enters/chooses task details, such as:
      - Task type
      - Date and time
      - Classroom location (if assigned right away)
      - Requested TAs (if requested right away)
      - Sections that are involved
   4. The instructor submits the task creation form.

5. The system saves the task and notifies relevant parties (e.g., TAs).

6. Special / Quality Requirements:
   - Should be completed in less than 1 minute to ensure efficiency.

Use Case: Creating Task Type

1. Name: Creating Task Type
2. Participating actor: Instructor
3. Entry Condition:
   - The instructor has decided to create new task types, so that TAs can choose when entering their workload.
4. Exit Condition:
   - A new task type has been successfully created and stored in the system.
5. Flow of Events:

   1. The instructor navigates to the task type management section.
   2. The instructor selects the option to create a new task type.
   3. The instructor enters the essential infos such as task type name and description, how long can it take etc.
   4. The instructor submits the form.
   5. The system saves the new task type, making it available for selection when creating tasks for the instructor him/herself and also TAs to choose when entering workload.

6. Special / Quality Requirements:
    - The system should prevent duplicate task types from being created.
    - The task type creation process should be completed in less than 30 seconds to minimize delay.

Use Case: Answer Request

1. Name: Answer Request
2. Participating actor: Instructor
3. Entry Condition:
    - There are pending requests from TAs in the system.
4. Exit Condition:
    - The request has been processed (either approved or rejected), TA is notified.
    - The system should log all decisions for those requests for further reviews.
5. Flow of Events:

    1. The instructor navigates to the request management section.
    2. The instructor views the list of pending requests.
    3. The instructor reviews each request's details (type, date, urgency, etc.).
    4. The instructor selects a request and decides to approve or reject it.
    5. The system updates the request status and notifies the TA.

6. Special / Quality Requirements:
    - Requests should be clearly categorized and easy to filter.
    - The processing request (including notifying TAs) in system should take less than 5 seconds per request for efficiency.

Use Case: Approve/Reject TA Workload

1. Name: Approve/Reject TA Workload
2. Participating actor: Instructor
3. Entry Condition:
   - A TA has submitted their workload for approval.
4. Exit Condition:
   - The TA's workload has been either approved or rejected.
   - The system should log all workload decisions for tracking purposes.
5. Flow of Events:

   1. The instructor navigates to the TA workload approval section.
   2. The instructor selects a pending workload request.
   3. The instructor reviews workload details, including task type, submission date, and relevant comments.
   4. The instructor decides whether to approve or reject the request.
   5. The system updates the workload status and notifies the TA.

6. Special / Quality Requirements:
   - The approval/rejection decision should be applied to system in less than 1 minute per request.

Use Case: Approve/Reject TA Availability

1. Name: Approve/Reject TA Availability
2. Participating actor: Instructor
3. Entry Condition:
   - A TA has submitted an availability request for approval.
4. Exit Condition:
   - The TA's availability request has been either approved or rejected.
   - The system should log all availability decisions for future reference.
5. Flow of Events:
   1. The instructor navigates to the TA availability request section.
   2. The instructor selects a pending request.
   3. The instructor reviews request details, including urgency, time period, and justification.
   4. The instructor decides whether to approve or reject the request.
   5. The system updates the request status and notifies the TA.
6. Special / Quality Requirements:
   - Urgent requests should be highlighted for quick attention.
   - The process of approvel or denial should be applied within 20 seconds to ensure responsiveness.

Use Case: Dismiss TA from Assignment

1. Name: Dismiss TA from Assignment
2. Participating actor: Instructor
3. Entry Condition:
   - A TA is currently assigned to a task (exam, quiz, recitation, etc.), and instructor decided to dismiss the TA from a particular task.
4. Exit Condition:
   - The TA has been dismissed from the assignment.
   - The TA has been notified of the dismissal.
   - The dismissal action should be logged for future reference.
5. Flow of Events:

   1. The instructor navigates to the task management section.
   2. The instructor selects a task and views the assigned TAs.
   3. The instructor chooses a TA to dismiss from the assignment.
   4. The system removes the TA from the task and updates the records.
   5. The TA receives a notification about the dismissal.

6. Special / Quality Requirements:
   - The TA should be notified within 5 seconds after dismissal.
   - The entire dismissal process should take less than 30 seconds to complete.

Use Case: View Personal Task Info

1. Name: View Personal Task Info
2. Participating Actor: TA
3. Entry Condition:
   - TA opened the Dashboard page and clicked to Tasks option.
   - There are tasks entered to the system.
4. Exit Condition:
   - TA clicked to another option or changed to another page.
5. Flow of Events:
   1. TA opens the Dashboard page.
   2. TA clicks to Tasks option.
   3. Tasks are shown.

Use Case: Print Classroom Details

1. Name: Print Classroom Details
2. Participating Actor: TA, Instructor
3. Entry Condition:
   - Actor clicked the print button after choosing an exam.
   - There are exams created.
4. Exit Condition:
   - Actor canceled the print.
   - Classroom details are successfully printed.
5. Flow of Events:
   1. Actor opens the Exams page.

    2. Actor selects an exam.
    3. Actor clicks to print button.
    4. Classroom details are printed.

6. Special / Quality Requirements:
    - After user clicks to print button, system should ask user to either select a destination to save classroom details as a file or select a printer to print classroom details.

Use Case: (Un)Locking Swaps for Yourself

1. Name: (Un)Locking Swaps for Yourself
2. Participating Actor: TA
3. Entry Condition:
    - TA opened the Exams page.
    - There are exams which TA is assigned to.
4. Exit Condition:
    - Swaps are locked or unlocked.
5. Flow of Events:
    1. TA opens the Exams page.
    2. TA selects a task from his/her tasks.
    3. Clicks to switch button to turn the locked option on or off.
6. Special / Quality Requirements:
    - System should ensure that no other TA is allowed to send a swap request to this TA for the particular task if swaps are locked.

Use Case: View Weekly Schedule

A) View Weekly Schedule of TAs

1. Name: View Weekly Schedule of TAs
2. Participating Actor: Dean’s Office, Administrator Assistant, Insurctor
3. Entry Condition:
    - Actor selected one of TAs and directed to the Profile page.
4. Exit Condition:
    - Actor changes to another page.
5. Flow of Events:
    1. Actor opens the TA list.
    2. Actor selects one of the TAs.
    3. Weekly schedule of the selected TA is displayed.

B)View Weekly Schedule of yourself

1. Name: View Weekly Schedule of TAs
2. Participating Actor: Insurctor, TA
3. Entry Condition:
    - Actor opens the Profile page.
4. Exit Condition:
    - Actor changes to another page.
5. Flow of Events:
    1. Actor opens the Profile page
    2. Weekly schedule is displayed

Use Case: Inform Dean’s Office about Selected TAs

1. Name: Inform Dean’s Office about Selected TAs
2. Participating Actor: Administrator Assistant
3. Entry Condition:
   - Actor selected TAs among the ones who applied for paid proctoring.
   - There are enough TAs to select.
4. Exit Condition:
   - Actor canceled to inform after selecting TAs.
   - Dean’s office is informed.
5. Flow of Events:
   1. Actor opens the Dashboard page.
   2. Actor selects enough TAs among the ones who applied for paid proctoring.
   3. Actor clicks to Notify Dean’s Office button.
   4. Dean’s Office is notified about which TAs are selected for paid proctoring.
6. Special / Quality Requirements:
   - System should immediately send a notification to the Dean’s Office of the faculty.

Use Case: Inform All Department TAs about New Paid Proctoring

1. Name: Inform All Department TAs about New Paid Proctoring
2. Participating Actor: Administrator Assistant

3. Entry Condition:
    - Actor opened the Dashboard page.
    - There is at least one paid proctoring requested.
4. Exit Condition:
    - TAs are informed.
5. Flow of Events:
    1. Actor opens the Dashboard page.
    2. Actor selects the paid proctoring among the ones that are available to inform the TAs.
    3. Actor clicks the Inform TAs button.
    4. TAs of the department are notified that there is a new paid proctoring available.
6. Special / Quality Requirements:
    - System should update the available paid proctoring tasks for each TA in the department.

Use Case: Assign TAs to Other Faculties Proctoring

1. Name: Assign TAs to Other Faculties Proctoring
2. Participating Actor: Dean's Office
3. Entry Condition:
    - Actor opened the TA Requests page.
    - There are TAs selected by Administrator Assistants for a particular exam created by another faculty.
4. Exit Condition:
    - TAs are assigned to other faculty's exam.

5. Flow of Events:
    1. Actor opens the TA Requests page.
    2. Actor selects an exam created by another faculty.
    3. Actor approves the TA selection made by Administrator Assistants for this exam.
    4. TAs assigned to exams created by other faculty.
6. Special / Quality Requirements:
    - After the assignment is made, Dean's Office that created the exam should get notified and exam details for other actors should be updated immediately.

Use Case: Approve/Reject TA Request Coming from Other Faculties

1. Name: Approve/Reject TA Request Coming from Other Faculties
2. Participating Actor: Dean's Office
3. Entry Condition:
    - Actor opened the Dashboard page.
    - There are TA requests pending.
4. Exit Condition:
    - Actor rejects or accepts the request.
5. Flow of Events:
    1. Actor opens the Dashboard page.
    2. Actor clicks to Received Request option.
    3. Actor selects the desired request coming from other faculties and approves or rejects it.

Use Case: Approve/Reject Other Department TA Request from Instructor

1. Name: Approve/Reject Other Department TA Request from Instructor
2. Participating Actor: Dean’s Office
3. Entry Condition:
   - Actor opened the Dashboard page.
   - There are TA requests pending.
4. Exit Condition:
   - Actor rejects or accepts the request.
5. Flow of Events:
   1. Actor opens the Dashboard page.
   2. Actor clicks to Received Request option.
   3. Actor selects the desired request coming from instructors and approves or rejects it.

Use Case: Dismiss TA from Assignment

1. Name: Dismiss TA from Assignment
2. Participating Actor: Dean’s Office
3. Entry Condition:
   - Actor opened the Exams page.
   - There are exams created.
   - There are TAs assigned to the exam selected.
4. Exit Condition:

    - TA dismissed from assignment.

5. Flow of Events:
    1. Actor opens the Exams page.
    2. Actor selects the desired exam.
    3. Actor selects the TA to dismiss.
    4. Actor clicks to dismiss button.
    5. Selected TA gets dismissed.
6. Special / Quality Requirements:
    - TA should be dismissed and the exam details for other actors like TA and instructors should be updated immediately.

Use Case: Inform Administrator Assistants about Requested TAs

1. Name: Inform Administrator Assistants about Requested TAs
2. Participating Actor: Dean’s Office
3. Entry Condition:
    - Actor opened the TA Requests page.
    - There are exams that TAs are requested for.
4. Exit Condition:
    - Selected Administrator Assistant got notified.
5. Flow of Events:
    1. Actor opens the TA Requests page.
    2. Actor selects an exam.
    3. Actor selects a department.

    4. Actor chooses the number TAs requested from the particular department.
    5. Actor clicks to Send Request button.
    6. Administrator Assistant of the desired department gets notified.

6. Special / Quality Requirements:
    - Administrator Assistant of the selected department should get a notification immediately.

Use Case: Approve/Reject Swap of TAs between Different Faculties/Departments

1. Name: Approve/Reject Swap of TAs between Different Faculties/Departments
2. Participating Actor: Dean’s Office
3. Entry Condition:
    - Actor opened the Dashboard page.
    - There are swap requests pending.
4. Exit Condition:
    - Actor rejects or accepts the request.
5. Flow of Events:
    1. Actor opens the Dashboard page.
    2. Actor clicks to Received Request option.
    3. Actor selects the desired swap request and approves or rejects it.

Use Case: Request TAs from Other Faculties

1. Name: Request TAs from Other Faculties

2. Participating Actor: Dean’s Office
3. Entry Condition:
    - Actor opened the TA Requests page.
    - There are exams that TAs are requested for by instructors.
4. Exit Condition:
    - Selected faculty’s Dean’s Office got notified.
5. Flow of Events:
    1. Actor opens the TA Requests page.
    2. Actor selects an exam.
    3. Actor selects another faculty.
    4. Actor chooses the number TAs requested from the particular faculty.
    5. Actor clicks to Send Request button.
    6. Dean’s Office of the desired department gets notified.
6. Special / Quality Requirements:
    - Dean’s Office of the selected faculty should get a notification immediately.

Use Case: Set Current Term

1. Name: Set Current Term
2. Participating Actor: Admin
3. Entry Condition:
    - The admin is authenticated and selects “Database” option.
4. Exit Condition:

- The academic term is successfully set and reflected throughout the system.

5. Flow of Events:

    1. The admin navigates to the "Set Globals" section from the admin database page.
    2. The system displays the currently active term and an option to set it.
    3. The admin enters the new semester and clicks the set button.
    4. The system saves the new term as the active term.
    5. All term-related functionalities (course associations, proctoring tasks, workload entries, etc.) are now bound to the new term.

6. Special / Quality Requirements:

    - The term change must be logged and tracked.
    - The updated term should be applied system-wide immediately without requiring a system restart (for new actions that started after term change).

Use Case: Set TA Proctoring Task Cap

1. Name: Set TA Proctoring Task Cap
2. Participating Actor: Admin
3. Entry Condition:

   - The admin is logged in with the appropriate permissions and selects “Database” option.

4. Exit Condition:
   - The system successfully saves and enforces the new proctoring task cap across future TA assignments.
5. Flow of Events:
   1. The admin navigates to the "Set Globals" section from the admin database page.
   2. The admin enters a new cap value (e.g., maximum number of proctoring assignments per TA).
   3. The system validates the input to ensure it is a non-negative integer.
   4. The admin confirms the changes.
   5. The system updates the global settings and applies the cap to TA assignment logic.
   6. Future TA assignments are automatically restricted based on the configured cap.
6. Special / Quality Requirements:
   - The cap change must be logged and tracked.
   - The cap can be adjusted at any time but should not retroactively unassign TAs unless explicitly triggered.

Use Case: Generate Term / Annual Report

1. Name: Generate Term / Annual Report

2. Participating Actor: Admin

3. Entry Condition:

   - The admin is authenticated and selects “Logs” option.

4. Exit Condition:

   - The system generates the requested report and presents it in a downloadable format.

5. Flow of Events:

   1. The admin accesses the "Print Report" button from the logs page.

   2. The system askes admin to choose between Term Report or Annual Report.

   3. The system gathers and processes the relevant data from the database.

   4. The report is generated in a structured format (PDF).

   5. The system offers options to download, print, or share the report securely.

6. Special / Quality Requirements:

   - The report format should be clear, structured, and professionally styled.

   - Only users with reporting permissions should access this feature.

   - Generated reports must be logged and tracked.

Use Case: View Application Logs

1. Name: View Application Logs

2. Participating Actor: Admin

3. Entry Condition:
    - The admin is authenticated and selects “Logs” option.
4. Exit Condition:
    - The admin has successfully viewed the logs and optionally exported them for review or investigation.
5. Flow of Events:
    1. The admin navigates to the "Logs" section from the admin dashboard.
    2. The system retrieves and log entries, including timestamps, actor IDs, and action summaries.
    3. The admin reviews the logs and optionally exports them in a file format.
    4. To export file admin can use the print logs button at the end of the listed logs.
6. Special / Quality Requirements:
    - Logs must be securely stored and tamper-proof to ensure accountability.
    - The log system should capture timestamps, actor identifiers, action types, and status (success/failure).

Use Case: Change Database

1. Name: Change Database
2. Participating Actor: Admin

3. Entry Condition:
    - The admin is authenticated and authorized to manage system-wide data such as students, courses, classes, instructors, TAs, and exams and admin selects "Database" option.
4. Exit Condition:
    - The database is successfully modified by creating, updating, or deleting records.
5. Flow of Events:
    1. The admin navigates to the "Database" section.
    2. The system offers two options for modifying data:
    3. Manual changes through the system interface.
    4. Automatic creation or update via Excel file upload.
    5. The admin selects the appropriate method and the entity to work on (e.g., TAs, instructors).
    6. The system validates the data and applies changes to the database.
    7. The system confirms the operation and logs the action.
6. Special / Quality Requirements:
    - Both manual and Excel-based changes must be validated before applying.
    - Admin actions must be logged and traceable.

Use Case: Create/Update Automatically via Excel

1. Name: Create/Update Automatically via Excel
2. Participating Actor: Admin
3. Entry Condition:
   - The admin is authenticated and has prepared an Excel file containing new or updated records for system entities such as students, courses, TAs, or exams and enters “Database” section.
4. Exit Condition:
   - The data from the Excel file is validated and successfully inserted or updated in the database.
5. Flow of Events:
   1. The admin navigates to the “Dump New Data” section under the database interface.
   2. The admin prepares and uploads the Excel file using the format.
   3. The system parses and validates the uploaded file:
      - Checks for correct column headers.
      - Validates data types and required fields constraints.
   4. If validation passes, the system updates the database records accordingly.
   5. If errors exist, the system presents a report detailing error reasons.
6. Special / Quality Requirements:
   - Excel templates must be standardized and available for download within the interface.
   - Validation must run before any changes are committed to prevent partial updates or corrupt data.

- The system must log every successful import along with a timestamp and admin identity.

Use Case: Change (Create/Update/Remove) Manually

1. Name: Change (Create/Update/Remove) Manually
2. Participating Actor: Admin
3. Entry Condition:
   - The admin is authenticated and intends to make manual changes to system records such as class information, students, courses, instructors, TAs, or exams and enters the “Database” section.
4. Exit Condition:
   - The selected record type is successfully created, updated, or removed through the system interface.
5. Flow of Events:
   1. The admin navigates to the “Find Data” section.
   2. The system displays a list of editable data categories:
      - Class Information
      - Students

    - Courses
    - Instructors
    - TAs
    - Exams

3. The admin selects one of the categories to manage.
4. The system displays existing Data and provides options to remove records that are no longer needed.
5. To update existing records, the admin can select a data segment and navigate to "View Data Details" section.
6. The admin can see and update the existing data from this part of the page.
7. To add new data to the database, admin navigate to "Create Data" section.
8. Admin can select data type to add and fill the information boxes to add the database.
9. The system validates each operation before saving changes.
10. A confirmation message is shown, and the changes are reflected in the database.

6. Special / Quality Requirements:

    - All changes should be logged with action details, actor identity, and timestamps.
    - Only authorized admins should be allowed to access this feature.
    - Validation rules must prevent accidental data corruption or deletion.
    - Critical operations (especially deletions) should include confirmation prompts.

Use Case: Change Class Information

1. Name: Change Class Information

2. Participating Actor: Admin

3. Entry Condition:

    - The admin is authenticated and selects "Class Information" as the data type from the dropdown menu in the "Database" section.

4. Exit Condition:

    - Class data is successfully created, updated, or removed in the system.

5. Flow of Events:

    1. The admin selects "Class Information" from the dropdown list.

    2. A list of current class records is shown in the top-left "Find Data" panel.

    3. To delete a class, the admin clicks the trash bin icon next to the record.

    4. To update a class, the admin selects a record. It is loaded into the "View Data Details" section.

    5. The admin modifies fields such as course code.

    6. After editing, the admin clicks "Update" to save the changes.

    7. To add a new class, the admin navigates to the "Create Data" section.

    8. The admin selects “Class Information” as the data type, fills in the relevant fields, and clicks "Create".

    9. The system validates the data and saves it to the database.

    10. A confirmation message appears after each successful operation.

6. Special / Quality Requirements:

    - Changes should reflect instantly in the "Find Data" panel.

    - All changes should be logged with action details, actor identity, and timestamps.

Use Case: Change Students

1. Name: Change Students

2. Participating Actor: Admin

3. Entry Condition:

    - The admin is authenticated and selects "Student" as the data type in the "Database" section.

4. Exit Condition:

    - Student records are successfully created, updated, or removed.

5. Flow of Events:

    1. The admin selects "Student" from the dropdown list.

    2. The top-left panel lists existing students.

    3. Admin deletes a student using the trash bin icon.

4. Admin selects a student to update; their details load into the "View Data Details" section.
5. Admin edits attributes like name, ID, or department and clicks "Update".
6. To add a new student, the admin uses the "Create Data" section.
7. Admin selects "Student" as the data type, fills in the fields, and clicks "Create".
8. The system validates the data and adds the new student.
9. Confirmation messages indicate successful operations.

6. Special / Quality Requirements:
   - Student IDs must be unique.
   - All changes should be logged with action details, actor identity, and timestamps.

Use Case: Change Courses

1. Name: Change Courses
2. Participating Actor: Admin
3. Entry Condition:
   - The admin is authenticated and selects "Course" as the data type in the "Database" section.
4. Exit Condition:

    - Course records are successfully added, updated, or removed.

5. Flow of Events:

    1. Admin selects "Course" in the dropdown menu.
    2. Existing courses are listed in the top-left panel.
    3. Admin deletes a course using the trash icon.
    4. To edit a course, the admin selects it, makes edits in the "View Data Details", and updates.
    5. To add a new course, the admin uses the "Create Data" panel.
    6. Selects “Course”, enters course details, and creates the record.
    7. System validates input and confirms successful operations.

6. Special / Quality Requirements:

    - All changes should be logged with action details, actor identity, and timestamps.

Use Case: Change Instructors

1. Name: Change Instructors
2. Participating Actor: Admin
3. Entry Condition:

- The admin selects "Instructor" from the dropdown list.

4. Exit Condition:
   - Instructor data is modified as intended (create/update/delete).
5. Flow of Events:
   1. The admin selects "Instructor" as the data type.
   2. The left panel displays current instructors.
   3. Admin uses the trash bin to remove records.
   4. Admin selects a record, makes changes in "View Data Details", and clicks "Update".
   5. New instructors can be created through the "Create Data" panel by entering their details and submitting.
   6. System saves and confirms the update.
6. Special / Quality Requirements:
   - Email fields must be validated.
   - All changes should be logged with action details, actor identity, and timestamps.

Use Case: Change TAs

1. Name: Change TAs
2. Participating Actor: Admin
3. Entry Condition:

- Admin selects "TA" from the dropdown list in the "Database" interface.

4. Exit Condition:
    - TA records are created, updated, or deleted successfully.
5. Flow of Events:
    1. Admin chooses "TA" as the data type.
    2. The left panel displays the current list of TAs.
    3. Admin removes TAs using the trash icon.
    4. To edit, the admin selects a TA; their data loads in "View Data Details".
    5. Admin modifies fields (e.g., full-time status, department), and clicks "Update".
    6. To add a TA, admin selects TA from the "Create Data" dropdown, fills in required fields, and submits.
    7. The system validates and confirms the operation.
6. Special / Quality Requirements:
    - TA emails and IDs must be unique.
    - All changes should be logged with action details, actor identity, and timestamps.

Use Case: Change Exams

1. Name: Change Exams

2. Participating Actor: Admin

3. Entry Condition:

   - The admin selects "Exam" from the data type list.

4. Exit Condition:

   - Exam records are added, updated, or removed through the UI.

5. Flow of Events:

   1. Admin selects "Exam" from the data type dropdown.

   2. The top-left shows scheduled exams.

   3. Admin removes an exam using the delete icon.

   4. Admin clicks a record to view and edit its details in "View Data Details".

   5. Admin updates exam details (date, course, duration) and clicks "Update".

   6. To add a new exam, admin goes to "Create Data", selects “Exam”, fills fields, and clicks "Create".

   7. The system processes and confirms the action.

6. Special / Quality Requirements:

   - All changes should be logged with action details, actor identity, and timestamps.

Use Case: View Student Information

1. Name: View Student Information

2. Participating Actor: Dean’s Office, Administrator Assistant, Admin

3. Entry Condition:
   - The actor has logged into the system with permission to view student data.
   - Student records exist in the system.
4. Exit Condition:
   - The actor has accessed and reviewed the relevant student information and completed the process.
5. Flow of Events:
   - The actor views the database section.
   - The actor selects the “Student Information” part.
   - The system presents a list of students, including ID numbers, names, and departments and other relevant information.
   - The actor selects a specific student to view more detailed information The system displays the requested student's details.
   - The actor exits the student information screen.
6. Special / Quality Requirements:
   - Only authorized users can view student data.
   - The system should support searching/filtering by name, ID, or department.

Use Case: View TA Information

1. Name: View TA Information
2. Participating Actor: Dean’s Office, Administrator Assistant, Admin

3. Entry Condition:
   - The actor is logged into the system with privileges to view TA data.
   - The system has valid TA records.
4. Exit Condition:
   - The actor has successfully viewed or filtered the TA information needed.
5. Flow of Events:
   1. The actor views the database section.
   2. The actor selects the "TA" part
   3. The system displays a list of TAs, showing key information.
   4. The system retrieves and presents matching TA profiles based on search.
   5. The actor selects a TA to see detailed information.
   6. The actor reviews or closes the details.
6. Special / Quality Requirements:
   - Access to TA data should be restricted to authorized staff.
   - The system should support searching/filtering by name, ID, or department.

Use Case: View Instructor Information

1. Name: View Instructor Information
2. Participating Actor: Dean's Office, Administrator Assistant, Admin
3. Entry Condition:
   - Instructor records exist in the system.

- The actor has permission to view instructor data.

4. Exit Condition:

   - The actor has viewed or filtered the instructor information and returns to the previous menu.

5. Flow of Events:

   1. The actor views the database section.
   2. The actor selects the “Instructor” part.
   3. The system shows a list of instructors (e.g., name, department, courses taught).
   4. The actor can search or filter by specific instructor name or department.
   5. The system displays detailed information about the instructor based on search (e.g., contact details, assigned TAs, office hours).
   6. The actor reviews the details as needed.
   7. The actor exits the module.

6. Special / Quality Requirements:

   - The system must enforce proper authorization for viewing instructor data.
   - The system should support searching/filtering by name, ID, or department.

Use Case: View Course Information

1. Name: View Course Information
2. Participating Actor: Dean’s Office, Administrator Assistant, Admin
3. Entry Condition:

- The system contains valid course records (e.g., course code, name, department).
- The actor has sufficient permissions to view course details.

4. Exit Condition:

   - The actor has finished reviewing the course details.

5. Flow of Events:

   1. The actor views the database section
   2. The actor selects the “Course Information” part.
   3. The system displays a list of courses with serached information
   4. The actor may filter the courses by department, level, or semester.
   5. The actor selects a course to see further information
   6. The system displays the course details.
   7. The actor reviews the information and either exits or returns to the course list.

6. Special / Quality Requirements:

   - The system must enforce proper authorization for viewing Course data.
   - The system should support searching/filtering by Course name, department etc.

Use Case: View Classroom Information

1. Name: View Classroom Information
2. Participating Actor: Dean’s Office, Administrator Assistant, Admin
3. Entry Condition:

   - Classroom data is stored in the system (room numbers, capacity, building, equipment).
   - The actor is authorized to view classroom usage and details.

4. Exit Condition:
    - The actor has accessed the classroom information.
5. Flow of Events:
    1. The actor views the database section.
    2. The actor selects the “Classroom Information” from the dropbox.
    3. The system shows a list of classrooms .
    4. The actor selects a classroom, and the system displays relevant details
    5. The actor reviews or prints the classroom details.
    6. The actor exits the module.
6. Special / Quality Requirements:
    - The system should check for proper authorization for viewing classroom data.
    - The system should support searching/filtering by classroom name.

Use Case: View Exam Information

1. Name: View Exam Information
2. Participating Actor: Dean’s Office, Administrator Assistant, Admin
3. Entry Condition:
    - Exam schedules exist in the system.
    - The actor is authorized to see exam details.
4. Exit Condition:
    - The actor has viewed the exam information and closes the view.
5. Flow of Events:

1. The actor views the database viewing part.
2. The actor navigates to the “Exam Information” section.
3. The system lists exams with key details (e.g., date, course code).
4. The actor can filter by date range, course, or exam type.
5. The actor selects an exam to view additional info (location, assigned TAs, number of enrolled students).
6. The system displays the chosen exam’s details.
7. The actor reviews the information and exits.

6. Special / Quality Requirements:
   - The system should check for proper authorization for viewing exam information.

Use Case: Assign TAs to Proctoring Task

1. Name: Assign TAs to Proctoring Task
2. Participating Actor: Instructor, Administrator Assistant
3. Entry Condition:
   - A proctoring is ready to be assigned.
   - TAs exist in the system with their availability/workload info recorded and are not in the blacklist.
   - Instructor decides to assign a TA to the proctoring task.
4. Exit Condition:
   - TA is assigned to a proctoring task with either manual or automatic assignment or notifies the actor if no suitable assignment is possible.

5. Flow of Events:
    1. An event is selected which needs TA assignment.
    2. Required TA number is displayed to the user.
    3. The actor selects either the “Automatic Assignment” or “Manual Assignment "option when creating or scheduling a task.
    4. Flow moves on with one of these two actions.

Use Case: Automatic Assignment

1. Name: Automatic Assignment
2. Participating Actor: Instructor, Administrator Assistant
3. Entry Condition:
    - TAs with requested priorities exists in the system.
    - There are enough TAs
    - Actor decides to do automatic assignment.
4. Exit Condition:
    - The system automatically assigns TA(s) or notifies the actor if no suitable assignment is possible.
5. Flow of Events:
    1. The actor selects the “Automatic Assignment” option when creating or scheduling a task.

   2. The system retrieves TAs' schedules, workload, blacklists, and availability constraints and sorts them based on required criteria.
   3. An assignment algorithm matches the best-fit TA(s) to the task.
   4. The system assigns those TA(s) automatically with best match. (No schedule conflict, lowest workload, no blacklist status)
   5. The system sends assignment notifications to the TAs and the instructor.
   6. If the system cannot assign TAs (e.g., none are available), the instructor is notified to seek alternatives. (Paid proctoring from other departments)

6. Special / Quality Requirements:
   - The algorithm should respect instructor blacklists, time conflicts, and maximum workload constraints.
   - The system notifies all involved parties of the outcome.
   - Assignments should be logged for traceability.

Use Case: Manual Assignment

1. Name: Manual Assignment
2. Participating Actor: Instructor, Administrator Assistant
3. Entry Condition:
   - A task (e.g., proctoring duty) needs to be assigned to TA(s).
   - There are TAs in the system for the instructor to pick from.
4. Exit Condition:

- Necessarry amount of TA(s) is/are successfully assigned manually, or the instructor cancels the process.

5. Flow of Events:
    1. The instructor creates or opens the task to be assigned.
    2. The instructor selects “Manual Assignment” instead of automatic assignment.
    3. The system displays a list of available TAs, including workload, schedule, and blacklist status.
    4. The instructor chooses the desired TA(s) and confirms.
    5. The system updates the task with the assigned TA(s).
    6. Notifications are sent to the assigned TAs regarding their new task.
6. Special / Quality Requirements:
    - Any conflict or blacklist warning should be highlighted before final confirmation.
    - The system should provide real-time info about TAs’ current obligations.
    - Changes are recorded in the system logs for transparency.

Use Case: Login

1. Name: Login
2. Participating Actor: Instructor, Administrator Assistant, Dean’s Office, TA
3. Entry Condition:
    - User has valid log in credentials (email & password).
4. Exit Condition:
    - The user's login data is sent to the backend for further authorization process.

5. Flow of Events:

    1. Actor enters the system’s login page.
    2. Actor inputs the email & password combination.
    3. Actor clicks on login to submit credentials.
    4. System authenticates and either welcomes the user or prints an error message displaying failed authentication.

Use Case: Change Password

1. Name: Change Password
2. Participating Actor: Instructor, Administrator Assistant, Dean’s Office, TA
3. Entry Condition:
    - The user is authenticated and selects the change password option from profile.
4. Exit Condition:
    - The user may navigate away from the section without taking any action.
5. Flow of Events:
    1. Actor selects profile.
    2. Actor chooses change password option from profile.
    3. Actor can update their password by writing down their email and new password.
    4. The password is changed when the actor confirms the final change.

Use Case: Forgot (Recover) Password

1. Name: Forgot Password
2. Participating Actor: Instructor, Administrator Assistant, Dean’s Office, TA
3. Entry Condition:
   - The actor has an existing account but cannot recall their password.
4. Exit Condition:
   - The actor resets their password successfully by sending the required data to their registered emails.
5. Flow of Events:
   1. Actor enters the system’s login page.
   2. The Actor navigates to the login page and clicks on the "Forgot Password".
   3. The system checks if the provided email exists in the user database.
   4. The system sends the details to recovering password to the email.

Use Case: Logout

1. Name: Logout
2. Participating Actor: Actor: Instructor, Administrator Assistant, Dean’s Office, TA
3. Entry Condition:
   - The user is currently logged into the system with an active session and decides to log out.
4. Exit Condition:

- User successfully logged out.

5. Flow of Events
    1. The user clicks the "Logout" button in the user interface.
    2. System receives the logout request.
    3. The system ends user’s session and instructs client’s browser to delete tokens.
    4. Confirmation message is shown and the user is directed to the login page.

Use Case: View Notifications

1. Name: View Notifications
2. Participating Actor: Actor: Instructor, Administrator Assistant, Dean’s Office, TA
3. Entry Condition:
    - The user is currently logged into the system with an active session.
    - User checks notification tab which is located in the main dashboard
4. Exit Condition:
    - User moves from the dashboard into some other part in which they can’t see their notification.
5. Flow of Events
    - An event occurs, such as swap request, an assignment to a task etc.
    - A notification message is generated with the relevant information.
    - Notifications are sent to the necessary people.
    - The user gets a personalized notification which specifies the task or the request.
    - Received notifications are logged with timestamps.

- The system may require recipients to take some actions according to the delivered notification.

Use Case: Apply Available Paid Proctoring

1. Name: Apply Available Paid Proctoring
2. Participating actor: TA
3. Entry Condition:
    - At least one unpaid proctoring duty is available in the system.
4. Exit Condition:
    - TA’s application is submitted to the system (pending approval by Dean’s Office).
    - Administrator Assistant is notified
5. Flow of Events:
    1. TA navigates to paid proctoring page
    2. System displays a list of open duties (filtered by TA’s eligibility).
    3. TA selects a duty and clicks "Apply."
    4. System records the application as "Pending Approval".
6. Special / Quality Requirements:
    - Applications must appear in the Administrator Assistant’s dashboard within 5 minutes.

- TA's schedule must temporarily block the proctoring slot until approval/rejection.

Use Case: Approve/Reject Proctoring Swap

1. Name: Approve/Reject Proctoring Swap
2. Participating actor: TA
3. Entry Condition:
   - TA has at least one pending proctoring swap request.
   - The requested proctoring slot is still active (not expired or canceled).
   - TA is eligible to approve/reject (no conflicting assignments).
4. Exit Condition:
   - Swap request is approved or rejected.
   - All parties (requester, receiver, instructor) are notified.
   - Schedules and dashboards are updated in real-time.
5. Flow of Events:
   1. TA navigates to "Received Swap Requests" section in dashboard.
   2. System displays pending requests with Proctoring duty details (course, time, current assignee).
   3. TA selects a request and chooses:
   4. "Approve":
   5. System validates no schedule conflicts for both TAs.
   6. Updates both TAs' schedules and dashboards.

    7. "Reject":
    8. System records rejection reason (optional).
    9. System triggers notifications to both sides.

6. Special / Quality Requirements:

    - Schedule and dashboard of the requester and receiver should be updated accordingly

Use Case: Approve/Reject Proctoring Task

1. Name: Approve/Reject Proctoring Task
2. Participating actor: TA
3. Entry Condition:

    - TA has at least one pending proctoring task assignment from an instructor.
    - The proctoring task is not expired (within allowable response period).

4. Exit Condition:

    - Proctoring task is approved or rejected by the TA.
    - Instructor and TA receive notification of the decision.
    - All relevant schedules and dashboards are updated.

5. Flow of Events:

    1. TA navigates to dashboard then to received requests
    2. TA selects the proctoring request from all requests he/she received
    3. TA approves or rejects the proctoring request

    4. TA and instructor are notified

6. Special / Quality Requirements:

    - Schedule of the TA should be updated accordingly
    - The page dashboard/proctorings of the TA should be updated accordingly
    - Proctoring information in the exam section of the instructor should be updated accordingly

Use Case: Withdraw Request Before Answer

1. Name: Withdraw Request Before Answer
2. Participating actor: TA
3. Entry Condition:

    - TA has at least one pending request.
    - The request is within the allowable withdrawal period.
    - The request is not yet approved/rejected by the recipient (instructor/admin).

4. Exit Condition:

    - Pending request is successfully withdrawn and removed from the recipient’s queue.

5. Flow of Events:

    1. TA navigates to "Pending Requests" in the dashboard.
    2. TA selects a request and clicks "Cancel"

3. System validates the request is still withdrawable and removes the request from all queues.
4. System notifies TA

6. Special / Quality Requirements
   - Pending requests page of TA is updated

Use Case: Sending Reminder Mail In Every 3 Hours

1. Name: Sending Reminder Mail In Every 3 Hours
2. Participating actor: TA, System
3. Entry Condition:
   - TA is assigned to an active proctoring task.
   - Proctoring task has not yet started or is ongoing.
4. Exit Condition:
   - Proctoring task is completed or canceled.
   - Final reminder is sent if task ends mid-cycle.
5. Flow of Events:
   - System checks for active proctoring tasks every 3 hours.
   - Repeat until the proctoring task is completed.

Use Case: Withdraw Swap Request

1. Name: Withdraw Swap Request
2. Participating actor: TA

3. Entry Condition:
   - TA has at least one pending swap request.
4. Exit Condition:
   - Swap request is successfully withdrawn and removed from the receiver's queue.
   - TA’s “Pending Requests” is updated accordingly.
5. Flow of Events:
   1. TA navigates to “Pending Requests” on “Dashboard”
   2. TA selects the particular swap request from all requests he/she sent
   3. TA withdraws his/her swap request
   4. System validates the request is still withdrawable and removes the request from the receiver’s requests
6. Special / Quality Requirements
   - The swap request is within the allowable withdrawal period.
   - The swap request is not yet approved/rejected by the receiver TA.

Use Case: Withdraw Workload Approval Request

1. Name: Withdraw Workload Approval Request
2. Participating actor: TA
3. Entry Condition:
   - TA has submitted a workload approval request that is still pending instructor review.

4. Exit Condition:

   - Workload request is removed from the instructor's approval queue.

5. Flow of Events:

   1. TA navigates to “Pending Requests” on “Dashboard”
   2. TA selects the particular workload approval request from all requests he/she sent
   3. TA withdraws his/her request
   4. System validates the request is still withdrawable and removes the request from all approval queues.

6. Special / Quality Requirements

   - The request is within the allowable withdrawal period.
   - The request has not yet been approved/rejected by the instructor.

Use Case: Send Swap Request

1. Name: Send Swap Request
2. Participating actor: TA
3. Entry Condition:

   - Initiator TA has at least one assigned proctoring duty.

4. Exit Condition:

   - Both TAs and instructor are notified.
   - Both TAs' dashboards reflect pending swap status.

5. Flow of Events:

    1. TA navigates to “Exams” page.
    2. TA selects one of his/her own proctoring task
    3. TA either selects one exam and a proctor in that exam or selects only a TA.
    4. TA sends the swap request to the selected TA

6. Special / Quality Requirements

    - Neither TA has swap restrictions.

Use Case: Send Mutual Swap Request

1. Name: Send Mutual Swap Request
2. Participating actor: TA
3. Entry Condition:

    - Initiator TA has at least one assigned proctoring duty.
    - Selected exam has at least one eligible proctor

4. Exit Condition:

    - Both TAs' dashboards reflect pending swap status.
    - Both TAs and instructor are notified.

5. Flow of Events:

1. TA navigates to exams page
2. TA selects the particular exam and a proctor of that exam
3. TA also selects one of his/her own proctoring task
4. TA sends the swap request

6. Special / Quality Requirements:
   - Both sides should have available schedules for the swap
   - Neither TA has swap restrictions.

Use Case: Send One Sided Swap Request

1. Name: Send One Sided Swap Request
2. Participating actor: TA
7. Entry Condition:

   - Initiator TA has at least one assigned proctoring duty.

8. Exit Condition:

   - Swap request is recorded in the system.
   - Both TAs' dashboards reflect pending swap status.
   - Both TAs and instructor are notified.

9. Flow of Events:

   1. TA navigates to “Proctoring Tasks” on “Dashboard”
   2. TA selects the particular proctoring task he/she has

3. TA selects an eligible TA
4. TA sends the swap request

10. Special / Quality Requirements:
    - The other side should have available schedule for the swap
    - Sender does not have swap restriction

Use Case: Send Notification

1. Name: Send Notification
2. Participating actor: TA, instructor, dean, administrator assistant, admin
3. Entry Condition:
    - Actor sends request.
    - Actor receives request.
    - Actor’s request gets accepted.
    - Actor’s request gets rejected.
4. Exit Condition:
    - Notification is successfully delivered to relevant parties.
5. Flow of Events:
    1. System detects the sending/approval/rejection of a request

Use Case: Set Unavailable Time

1. Name: Set Unavailable Time
2. Participating actor: TA
3. Entry Condition:
    - TA has not exceeded the maximum unavailability requests per semester
4. Exit Condition:
    - Unavailability request is submitted for approval.
    - TA’s dashboard reflects pending status.
    - Administrator Assistant and TA receives notification.
5. Flow of Events:
    1. TA navigates to profile page
    2. TA clicks the set unavailable time button
    3. TA enters the start date, end date and description
    4. TA sends the request

Use Case: Send Workload Approval Request

1. Name: Send Workload Approval Request
2. Participating actor: TA
3. Entry Condition:
    - TA has not exceeded the maximum allowed workload hours

4. Exit Condition:

   - Received requests page of instructor is updated
   - TA's "Pending Requests" page on "Dashboard" is updated
   - Both instructor and TA are notified

5. Flow of Events:

   1. TA navigates to "Tasks" page on "Dashboard"
   2. TA enters the details of the task such as "Task Type", "Time spent" and "Details".
   3. TA clicks the send button

6. Special / Quality Requirements

   - Time spent is less than maximum time cap allowed by the instructor