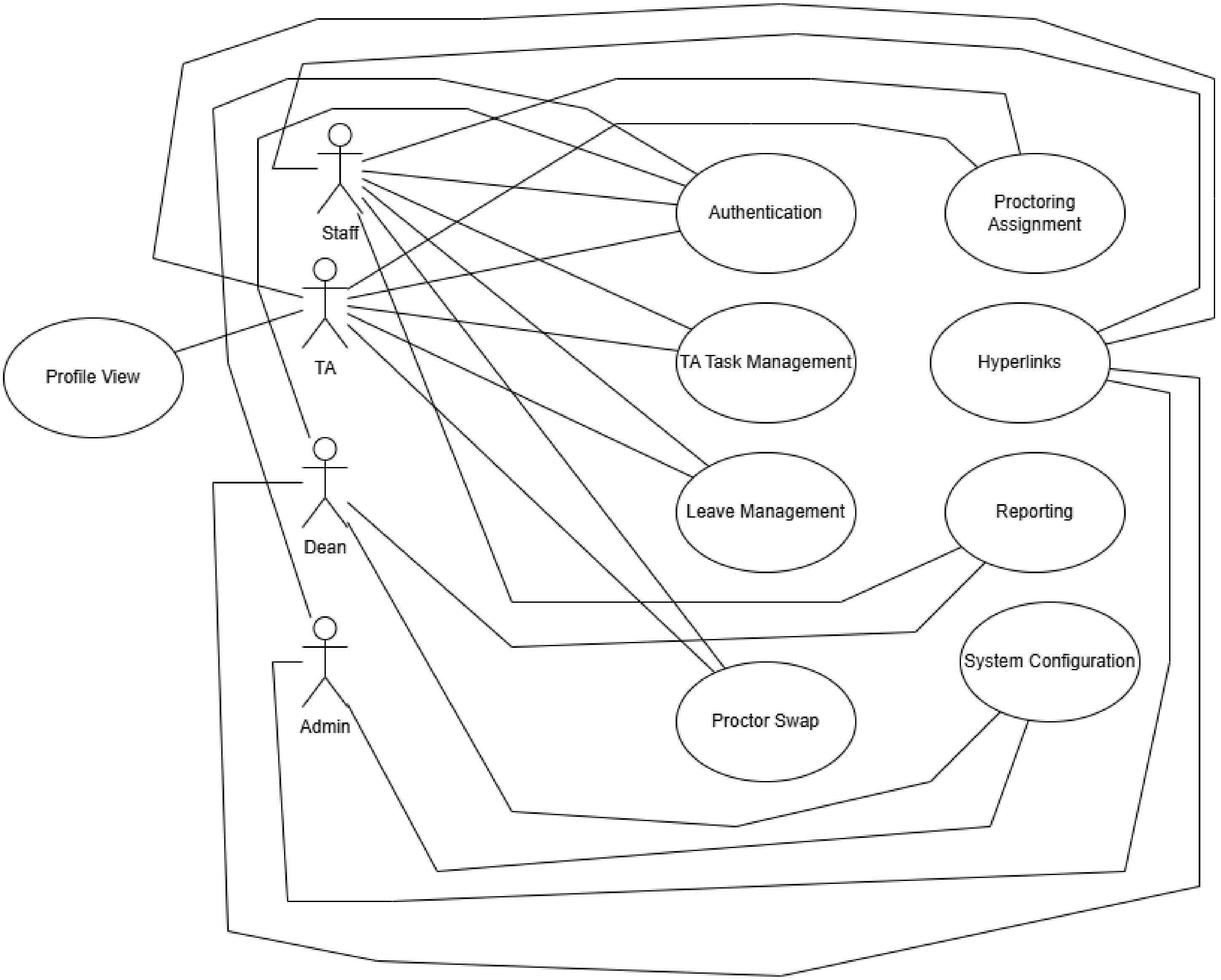
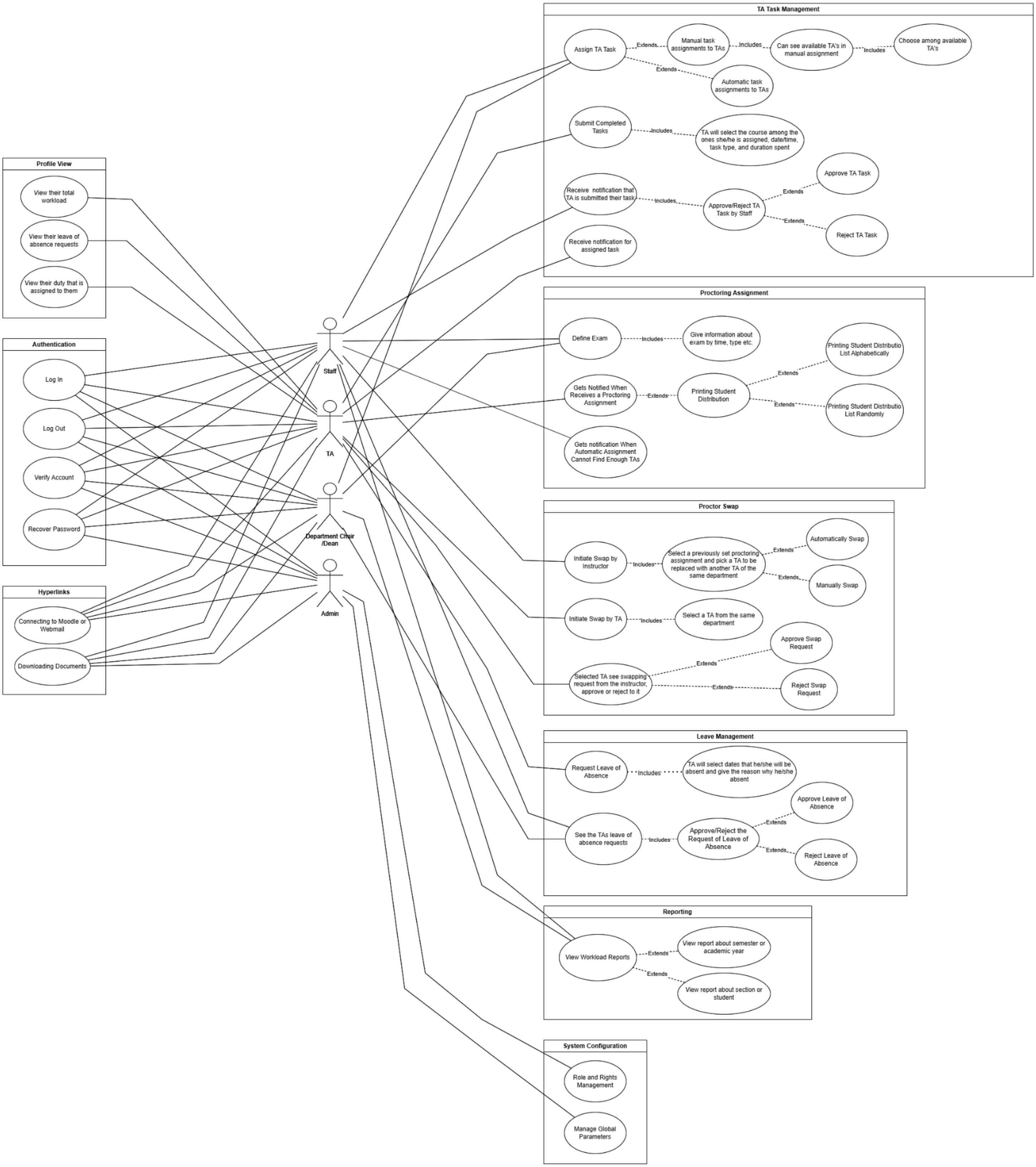
Table of Contents

1. Use Case Diagram

Level 0



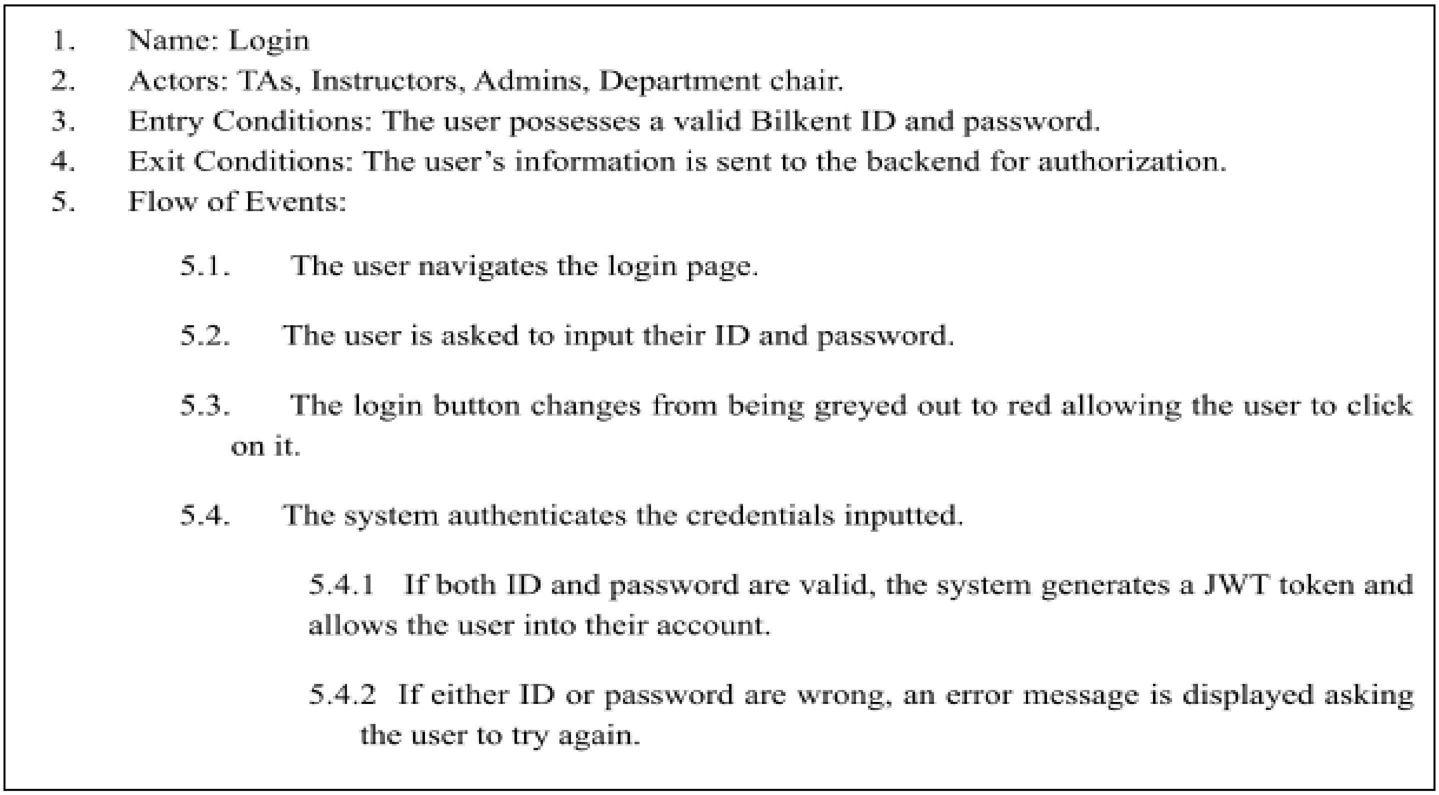
Level 1


Profile View
View their total workload
View their leave of absence requests
View their duty that is assigned to them
Authentication
Log In
Log Out
Verify Account
Recover Password
Hyperlinks
Connecting to Moodle or Webmail
Downloading Documents
Staff
TA
Department Chair /Dean
Admin
TA Task Management
Assign TA Task
Extends
Manual task assignments to TAs
Includes
Can see available TA's in manual assignment
Includes
Choose among available TA's
Extends
Automatic task assignments to TAs
Submit Completed Tasks
Includes
TA will select the course among the ones she/he is assigned, date/time, task type, and duration spent
Receive notification that TA is submitted their task
Includes
Approve/Reject TA Task by Staff
Extends
Approve TA Task
Extends
Reject TA Task
Receive notification for assigned task
Proctoring Assignment
Define Exam
Includes
Give information about exam by time, type etc.
Gets Notified When Receives a Proctoring Assignment
Extends
Printing Student Distribution
Extends
Printing Student Distributio List Alphabetically
Extends
Printing Student Distributio List Randomly
Gets notification When Automatic Assignment Cannot Find Enough TAs
Proctor Swap
Initiate Swap by Instructor
Includes
Select a previously set proctoring assignment and pick a TA to be replaced with another TA of the same department
Extends
Automatically Swap
Extends
Manually Swap
Initiate Swap by TA
Includes
Select a TA from the same department
Selected TA see swapping request from the instructor, approve or reject to it
Extends
Approve Swap Request
Extends
Reject Swap Request
Leave Management
Request Leave of Absence
Includes
TA will select dates that he/she will be absent and give the reason why he/she absent
See the TAs leave of absence requests
Includes
Approve/Reject the Request of Leave of Absence
Extends
Approve Leave of Absence
Extends
Reject Leave of Absence
Reporting
View Workload Reports
Extends
View report about semester or academic year
Extends
View report about section or student
System Configuration
Role and Rights Management
Manage Global Parameters

2. Use Case Textual Descriptions

2.1 Authentication

2.1.1 Login



1. Name: Login
2. Actors: TAs, Instructors, Admins, Department chair.
3. Entry Conditions: The user possesses a valid Bilkent ID and password.
4. Exit Conditions: The user's information is sent to the backend for authorization.
5. Flow of Events:

    5.1. The user navigates the login page.

    5.2. The user is asked to input their ID and password.

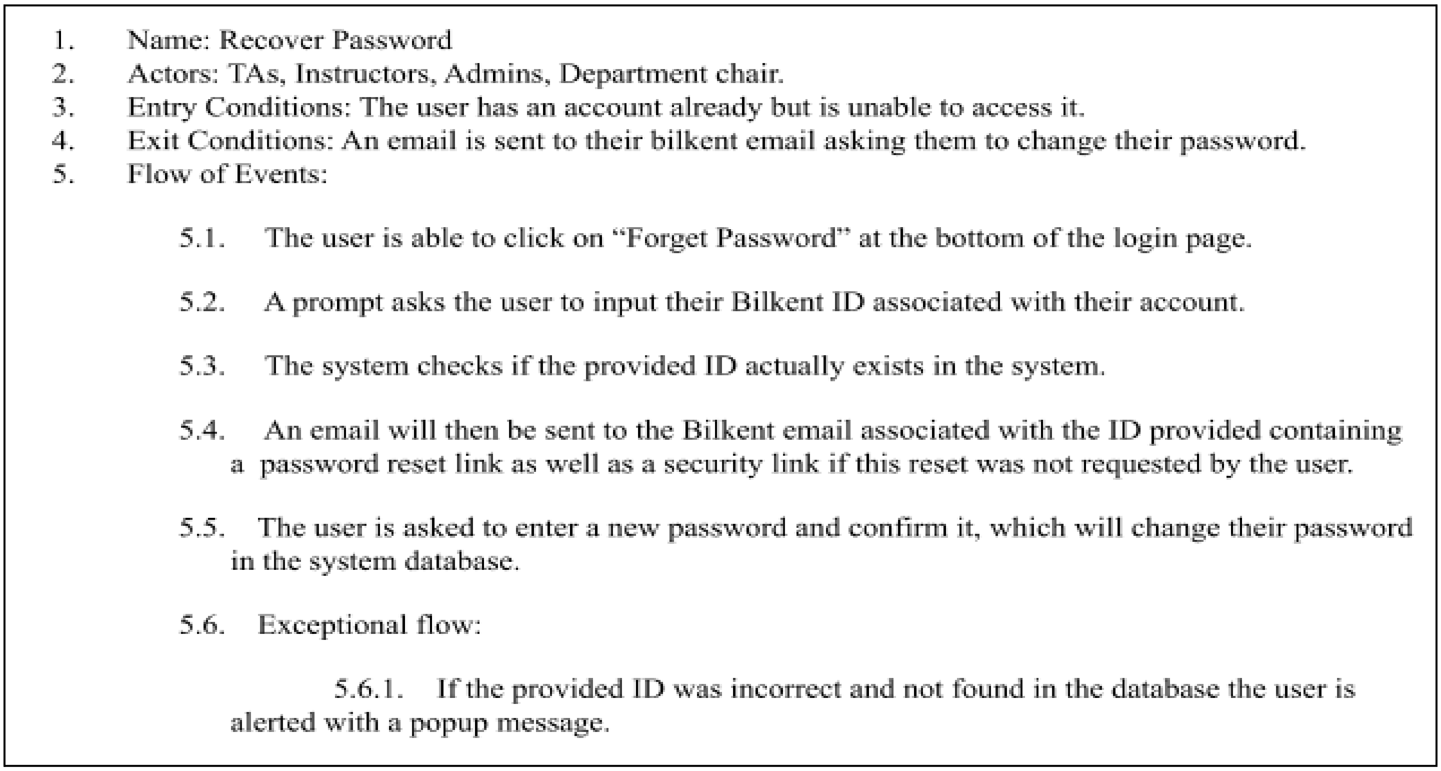
    5.3. The login button changes from being greyed out to red allowing the user to click on it.

    5.4. The system authenticates the credentials inputted.

        5.4.1 If both ID and password are valid, the system generates a JWT token and allows the user into their account.

        5.4.2 If either ID or password are wrong, an error message is displayed asking the user to try again.

2.1.2 Recover Password



1. Name: Recover Password
2. Actors: TAs, Instructors, Admins, Department chair.
3. Entry Conditions: The user has an account already but is unable to access it.
4. Exit Conditions: An email is sent to their bilkent email asking them to change their password.
5. Flow of Events:

    5.1. The user is able to click on “Forget Password” at the bottom of the login page.

    5.2. A prompt asks the user to input their Bilkent ID associated with their account.

    5.3. The system checks if the provided ID actually exists in the system.

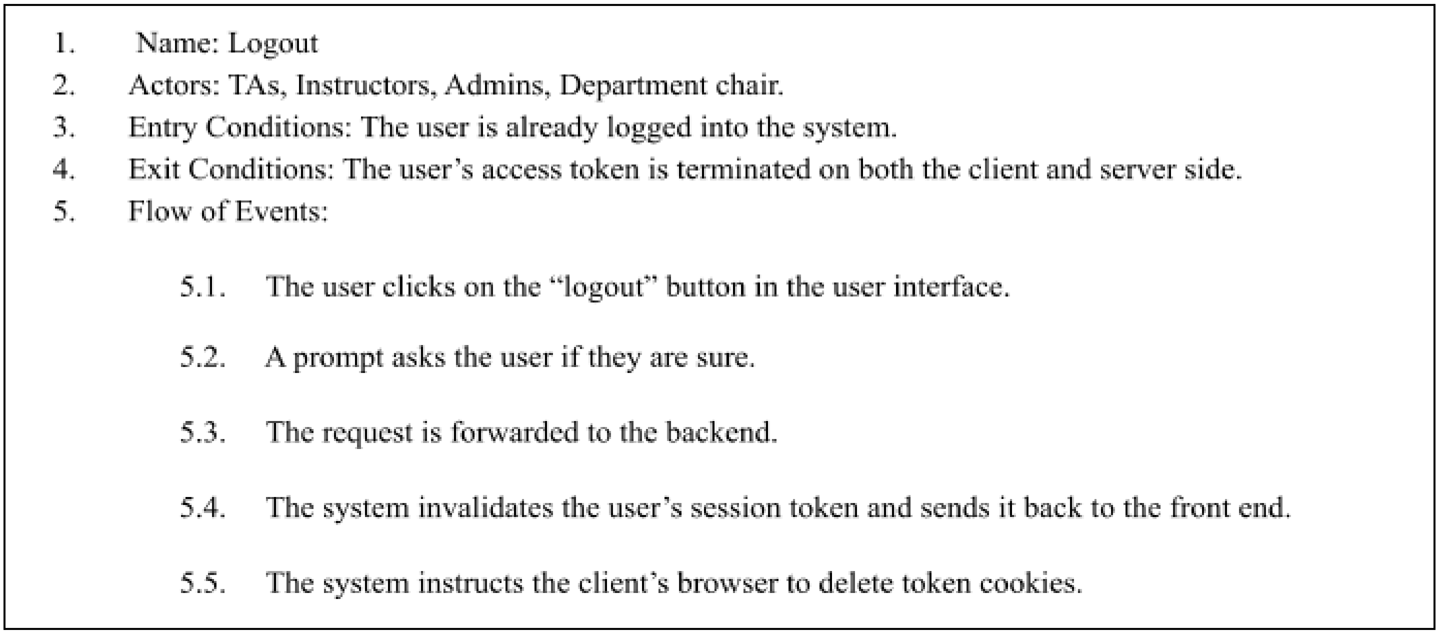
    5.4. An email will then be sent to the Bilkent email associated with the ID provided containing a password reset link as well as a security link if this reset was not requested by the user.

    5.5. The user is asked to enter a new password and confirm it, which will change their password in the system database.

    5.6. Exceptional flow:

        5.6.1. If the provided ID was incorrect and not found in the database the user is alerted with a popup message.

2.1.3 Logout



1. Name: Logout
2. Actors: TAs, Instructors, Admins, Department chair.
3. Entry Conditions: The user is already logged into the system.
4. Exit Conditions: The user’s access token is terminated on both the client and server side.
5. Flow of Events:

    5.1. The user clicks on the “logout” button in the user interface.

    5.2. A prompt asks the user if they are sure.

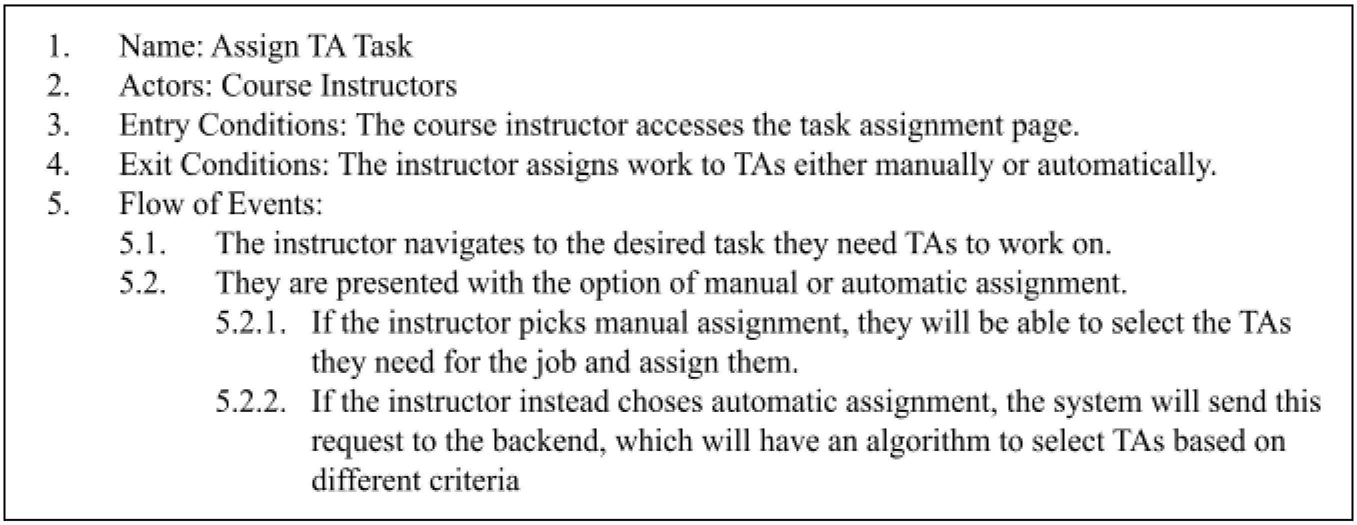
    5.3. The request is forwarded to the backend.

    5.4. The system invalidates the user’s session token and sends it back to the front end.

    5.5. The system instructs the client’s browser to delete token cookies.

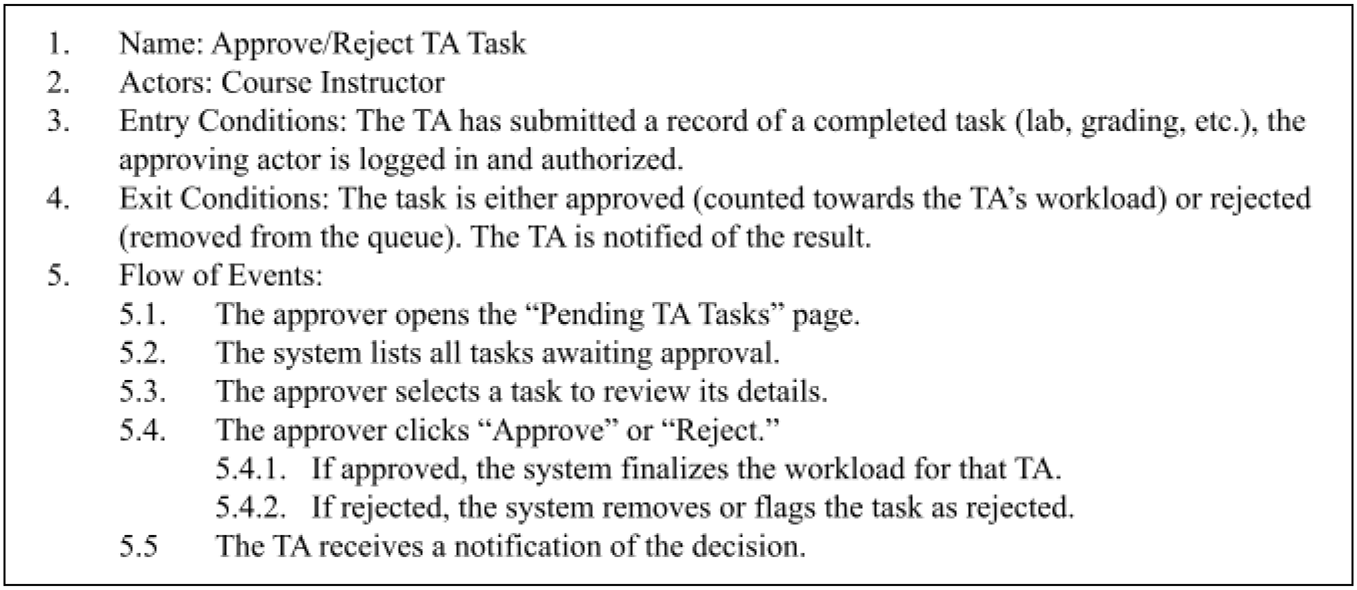
2.2 TA Task Management

2.2.1 Assign TA Task



1. Name: Assign TA Task
2. Actors: Course Instructors
3. Entry Conditions: The course instructor accesses the task assignment page.
4. Exit Conditions: The instructor assigns work to TAs either manually or automatically.
5. Flow of Events:
    - 5.1. The instructor navigates to the desired task they need TAs to work on.
    - 5.2. They are presented with the option of manual or automatic assignment.
        - 5.2.1. If the instructor picks manual assignment, they will be able to select the TAs they need for the job and assign them.
        - 5.2.2. If the instructor instead choses automatic assignment, the system will send this request to the backend, which will have an algorithm to select TAs based on different criteria

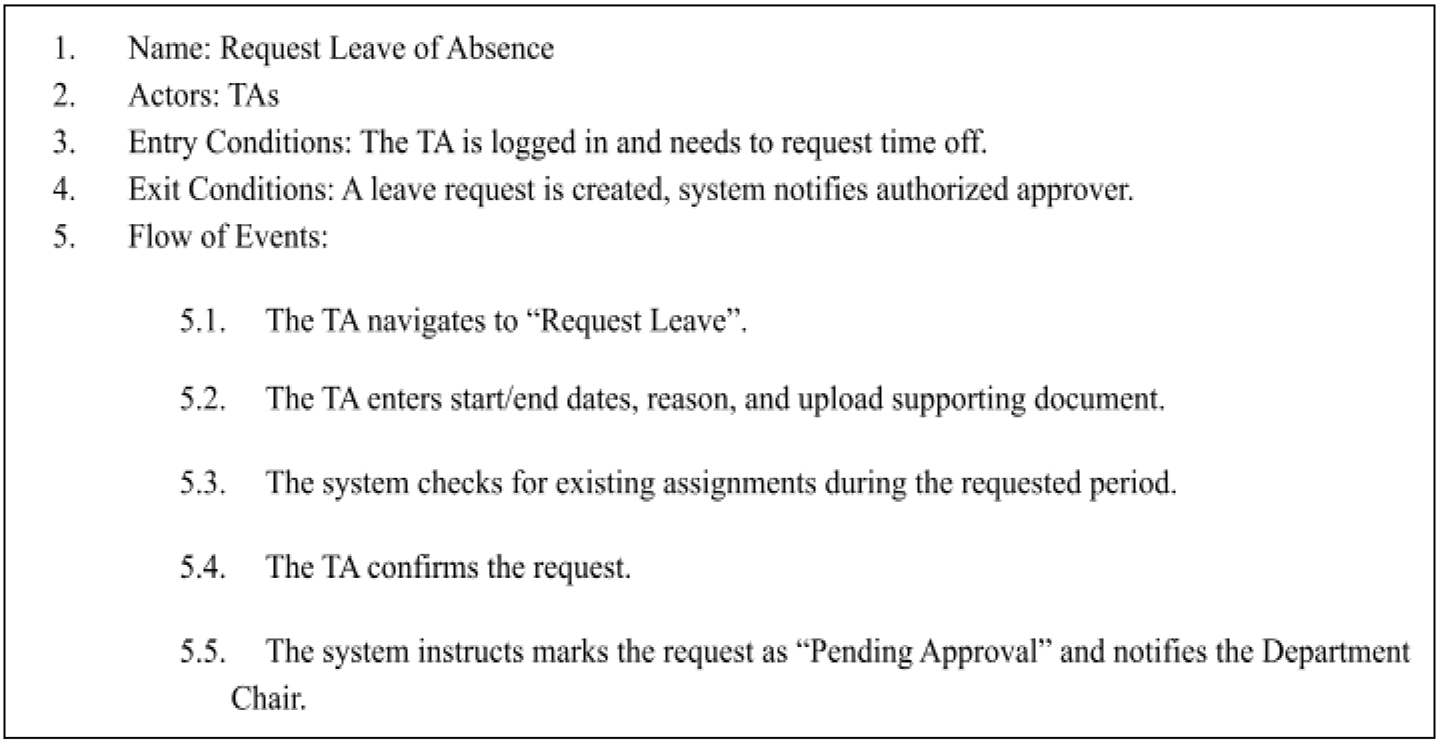
2.2.2 Approve/Reject TA Task



1. Name: Approve/Reject TA Task
2. Actors: Course Instructor
3. Entry Conditions: The TA has submitted a record of a completed task (lab, grading, etc.), the approving actor is logged in and authorized.
4. Exit Conditions: The task is either approved (counted towards the TA's workload) or rejected (removed from the queue). The TA is notified of the result.
5. Flow of Events:
    - 5.1. The approver opens the "Pending TA Tasks" page.
    - 5.2. The system lists all tasks awaiting approval.
    - 5.3. The approver selects a task to review its details.
    - 5.4. The approver clicks "Approve" or "Reject."
        - 5.4.1. If approved, the system finalizes the workload for that TA.
        - 5.4.2. If rejected, the system removes or flags the task as rejected.
    - 5.5 The TA receives a notification of the decision.

2.3 Leave Management

2.3.1 Request Leave of Absence



1. Name: Request Leave of Absence
2. Actors: TAs
3. Entry Conditions: The TA is logged in and needs to request time off.
4. Exit Conditions: A leave request is created, system notifies authorized approver.
5. Flow of Events:

   5.1. The TA navigates to “Request Leave”.

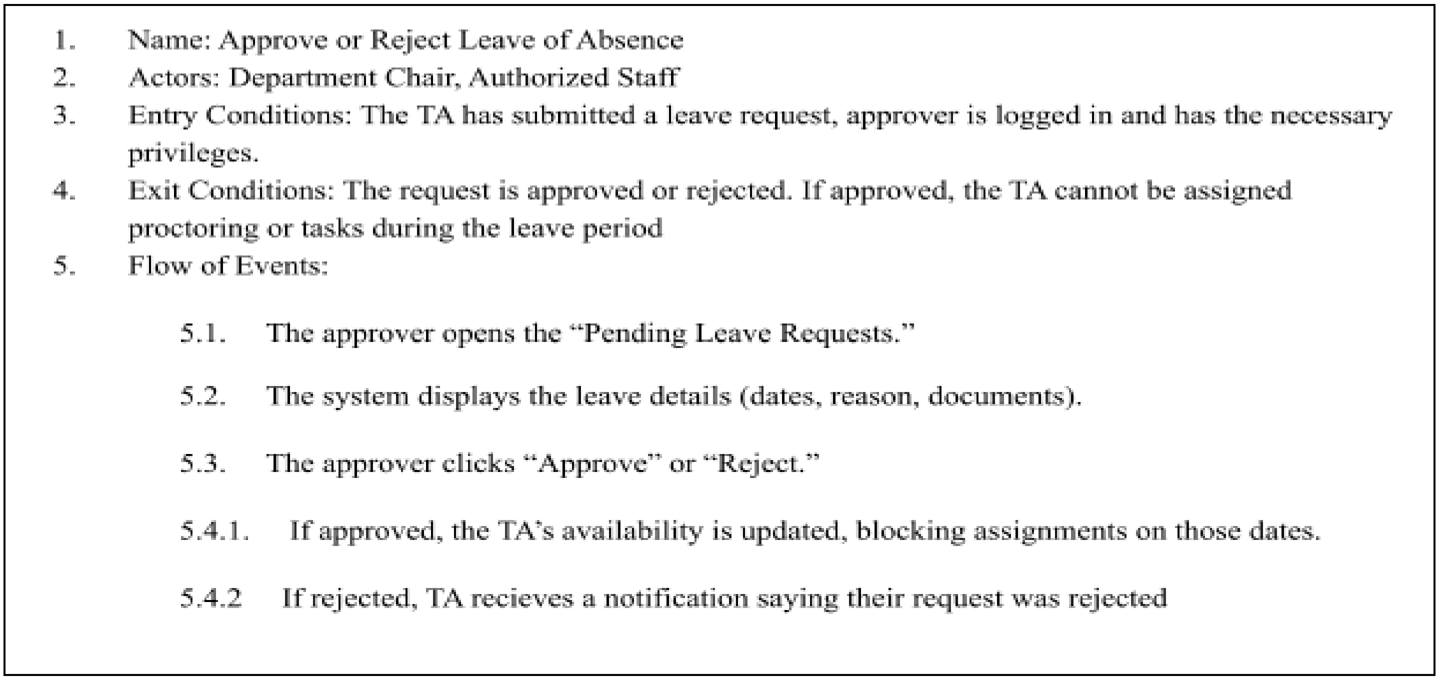
   5.2. The TA enters start/end dates, reason, and upload supporting document.

   5.3. The system checks for existing assignments during the requested period.

   5.4. The TA confirms the request.

   5.5. The system instructs marks the request as “Pending Approval” and notifies the Department Chair.

2.3.2 Approve Leave of Absence



1. Name: Approve or Reject Leave of Absence
2. Actors: Department Chair, Authorized Staff
3. Entry Conditions: The TA has submitted a leave request, approver is logged in and has the necessary privileges.
4. Exit Conditions: The request is approved or rejected. If approved, the TA cannot be assigned proctoring or tasks during the leave period
5. Flow of Events:

   5.1. The approver opens the “Pending Leave Requests.”

   5.2. The system displays the leave details (dates, reason, documents).

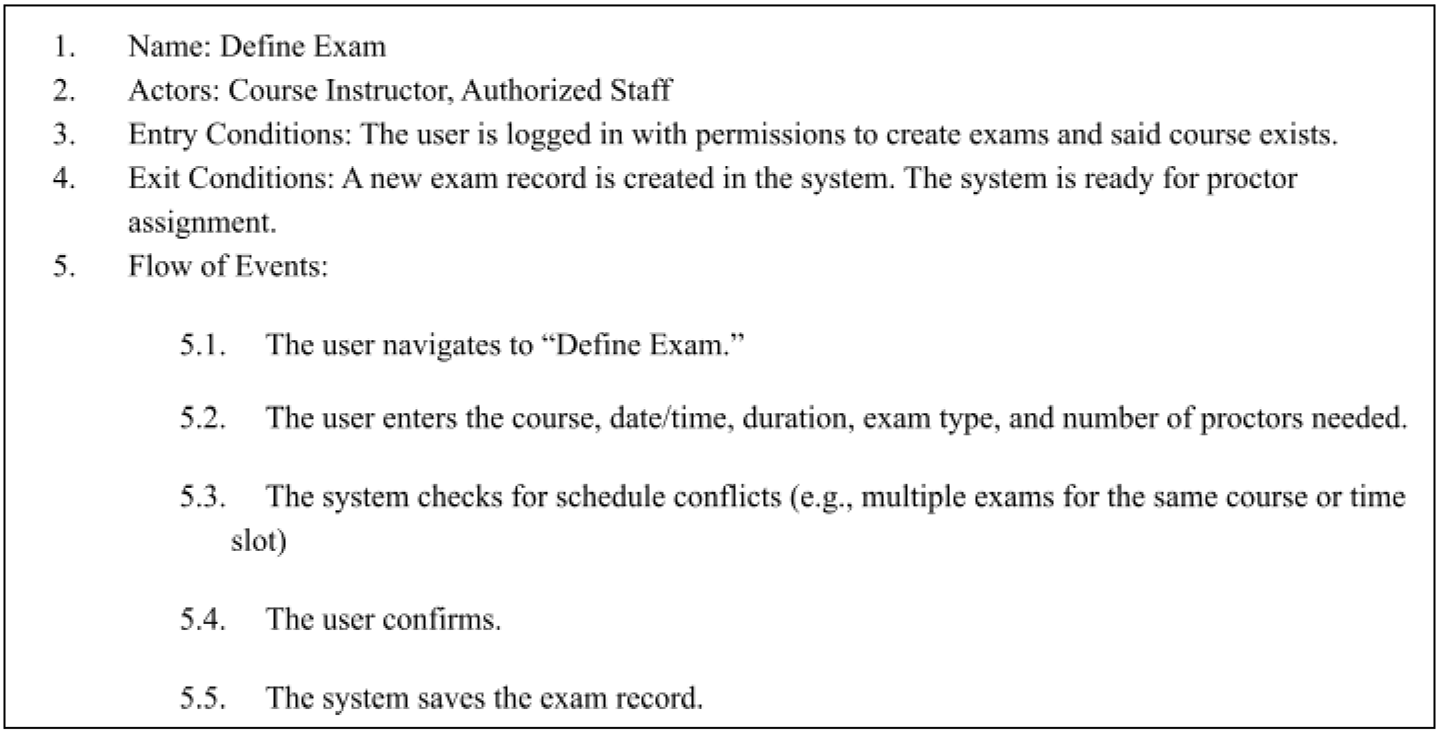
   5.3. The approver clicks “Approve” or “Reject.”

   5.4.1. If approved, the TA’s availability is updated, blocking assignments on those dates.

   5.4.2 If rejected, TA recieves a notification saying their request was rejected

2.4 Proctoring Assignment

2.4.1 Define Exam



1. Name: Define Exam
2. Actors: Course Instructor, Authorized Staff
3. Entry Conditions: The user is logged in with permissions to create exams and said course exists.
4. Exit Conditions: A new exam record is created in the system. The system is ready for proctor assignment.
5. Flow of Events:

   5.1. The user navigates to “Define Exam.”

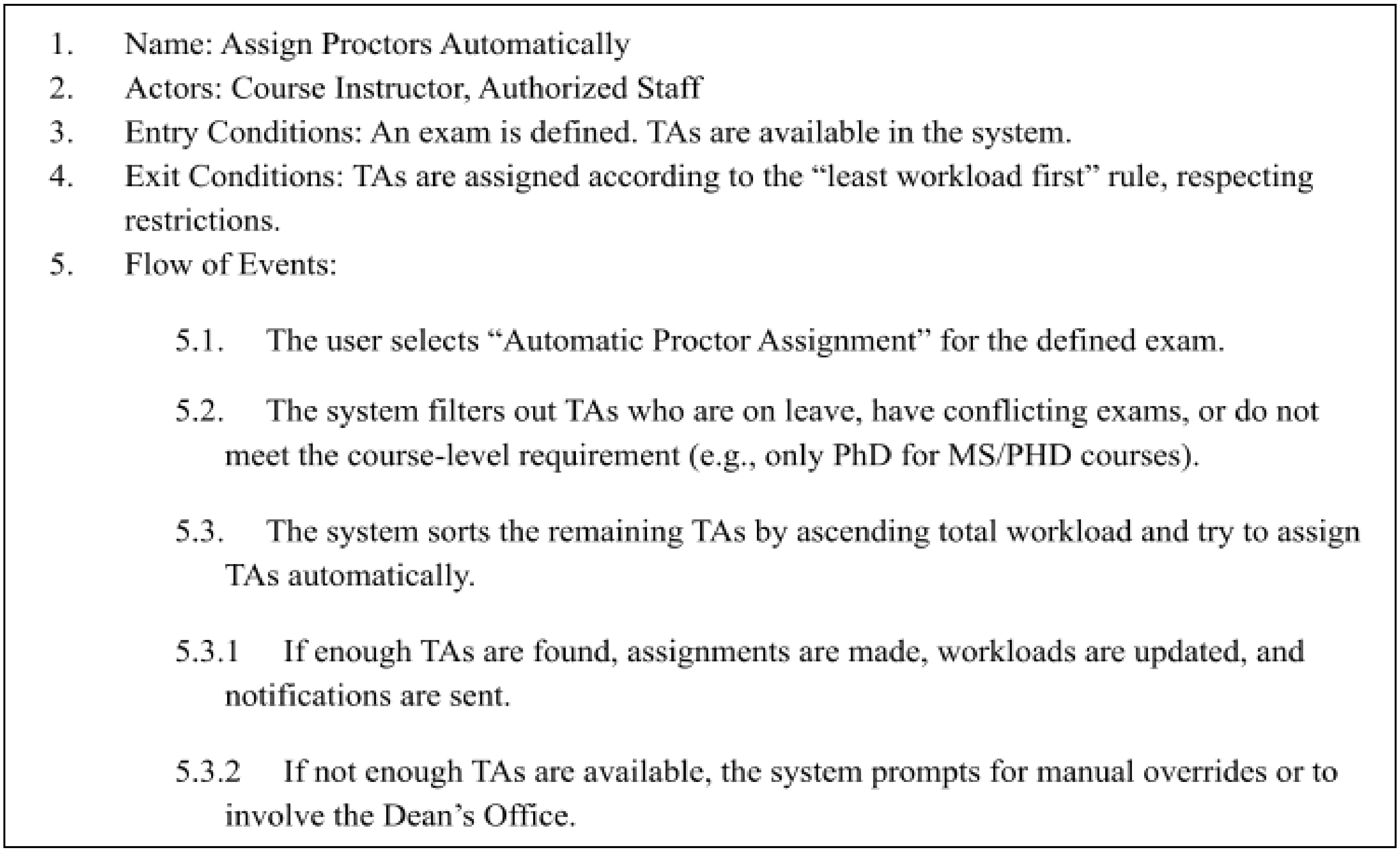
   5.2. The user enters the course, date/time, duration, exam type, and number of proctors needed.

   5.3. The system checks for schedule conflicts (e.g., multiple exams for the same course or time slot)

   5.4. The user confirms.

   5.5. The system saves the exam record.

2.4.2 Assign Proctors Automatically



1. Name: Assign Proctors Automatically
2. Actors: Course Instructor, Authorized Staff
3. Entry Conditions: An exam is defined. TAs are available in the system.
4. Exit Conditions: TAs are assigned according to the “least workload first” rule, respecting restrictions.
5. Flow of Events:

    5.1. The user selects “Automatic Proctor Assignment” for the defined exam.

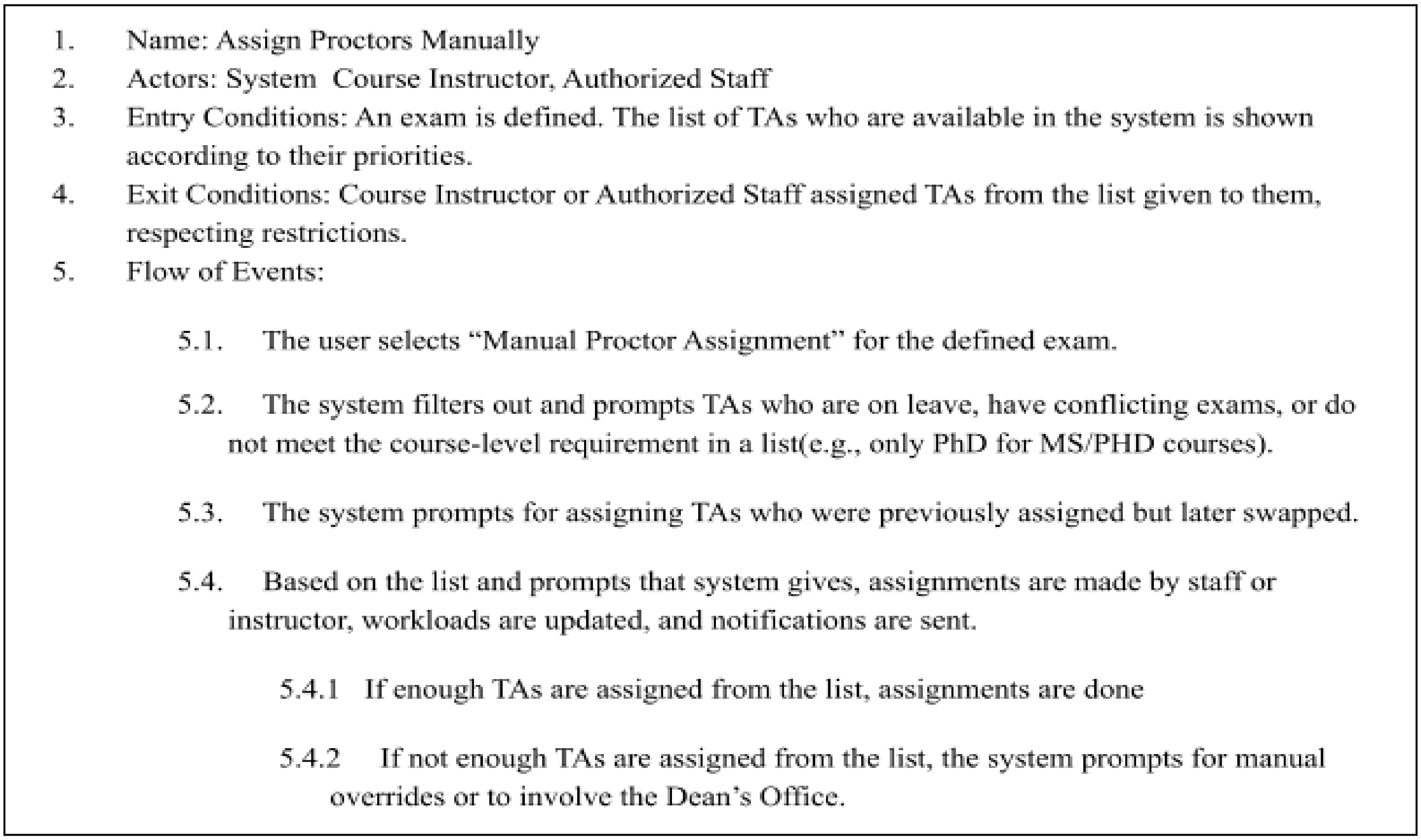
    5.2. The system filters out TAs who are on leave, have conflicting exams, or do not meet the course-level requirement (e.g., only PhD for MS/PHD courses).

    5.3. The system sorts the remaining TAs by ascending total workload and try to assign TAs automatically.

    5.3.1 If enough TAs are found, assignments are made, workloads are updated, and notifications are sent.

    5.3.2 If not enough TAs are available, the system prompts for manual overrides or to involve the Dean’s Office.

2.4.3 Assign Proctors Manually



1. Name: Assign Proctors Manually
2. Actors: System Course Instructor, Authorized Staff
3. Entry Conditions: An exam is defined. The list of TAs who are available in the system is shown according to their priorities.
4. Exit Conditions: Course Instructor or Authorized Staff assigned TAs from the list given to them, respecting restrictions.
5. Flow of Events:

    5.1. The user selects "Manual Proctor Assignment" for the defined exam.

    5.2. The system filters out and prompts TAs who are on leave, have conflicting exams, or do not meet the course-level requirement in a list(e.g., only PhD for MS/PHD courses).

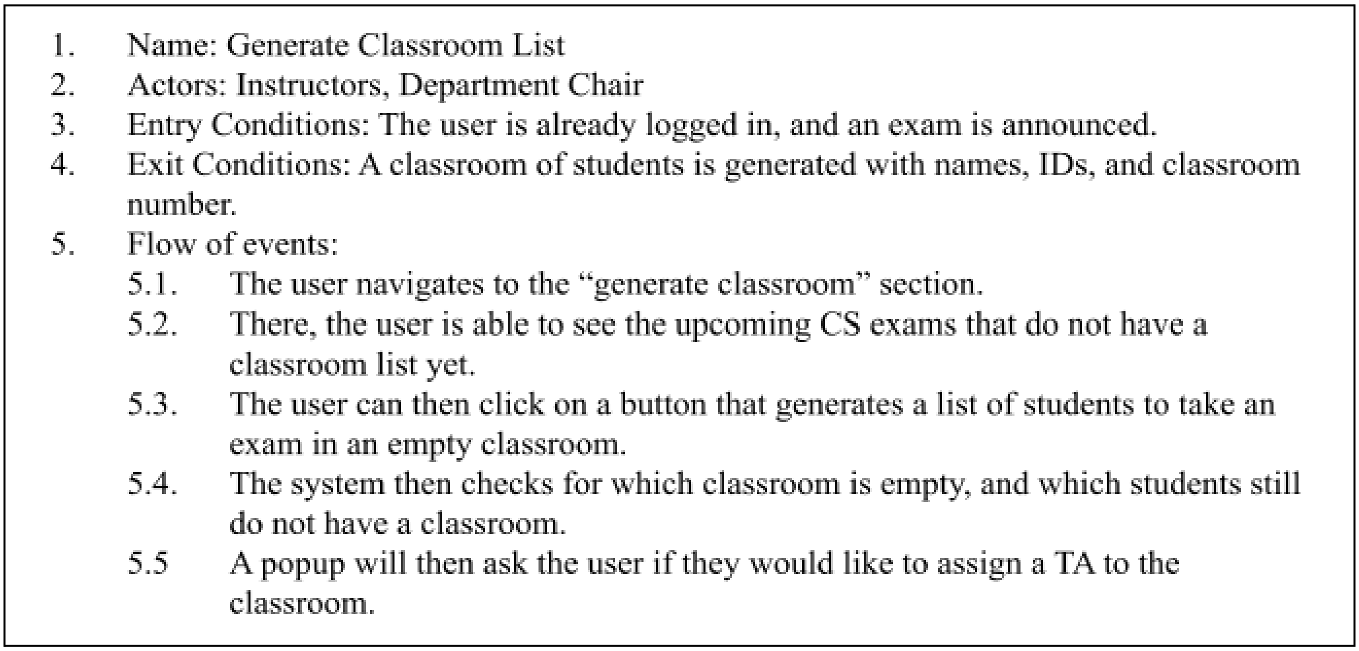
    5.3. The system prompts for assigning TAs who were previously assigned but later swapped.

    5.4. Based on the list and prompts that system gives, assignments are made by staff or instructor, workloads are updated, and notifications are sent.

        5.4.1 If enough TAs are assigned from the list, assignments are done

        5.4.2 If not enough TAs are assigned from the list, the system prompts for manual overrides or to involve the Dean's Office.

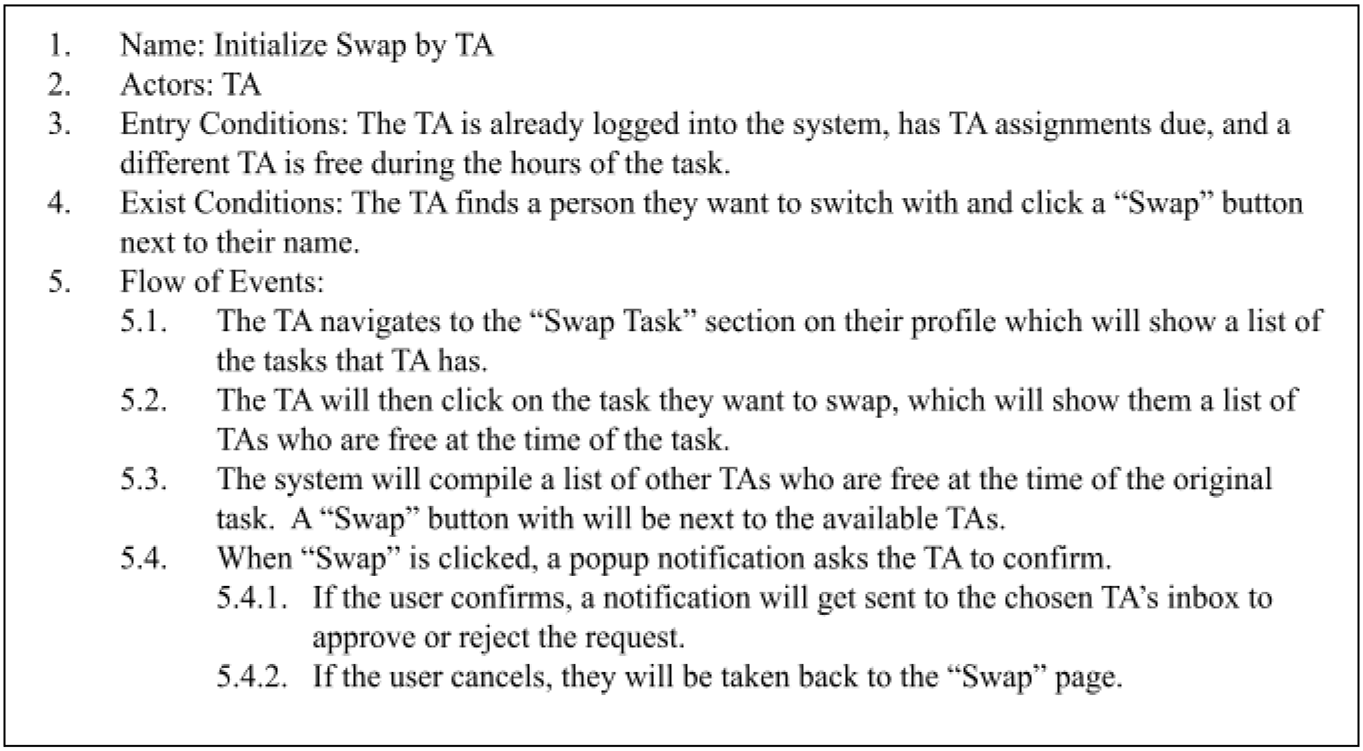
2.4.4 Generate Classroom Lists



1. Name: Generate Classroom List
2. Actors: Instructors, Department Chair
3. Entry Conditions: The user is already logged in, and an exam is announced.
4. Exit Conditions: A classroom of students is generated with names, IDs, and classroom number.
5. Flow of events:
    5.1. The user navigates to the "generate classroom" section.
    5.2. There, the user is able to see the upcoming CS exams that do not have a classroom list yet.
    5.3. The user can then click on a button that generates a list of students to take an exam in an empty classroom.
    5.4. The system then checks for which classroom is empty, and which students still do not have a classroom.
    5.5 A popup will then ask the user if they would like to assign a TA to the classroom.

2.5 Proctor Swap

2.5.1 Initiate Swap by TA



1. Name: Initialize Swap by TA
2. Actors: TA
3. Entry Conditions: The TA is already logged into the system, has TA assignments due, and a different TA is free during the hours of the task.
4. Exist Conditions: The TA finds a person they want to switch with and click a “Swap” button next to their name.
5. Flow of Events:

    5.1. The TA navigates to the “Swap Task” section on their profile which will show a list of the tasks that TA has.

    5.2. The TA will then click on the task they want to swap, which will show them a list of TAs who are free at the time of the task.

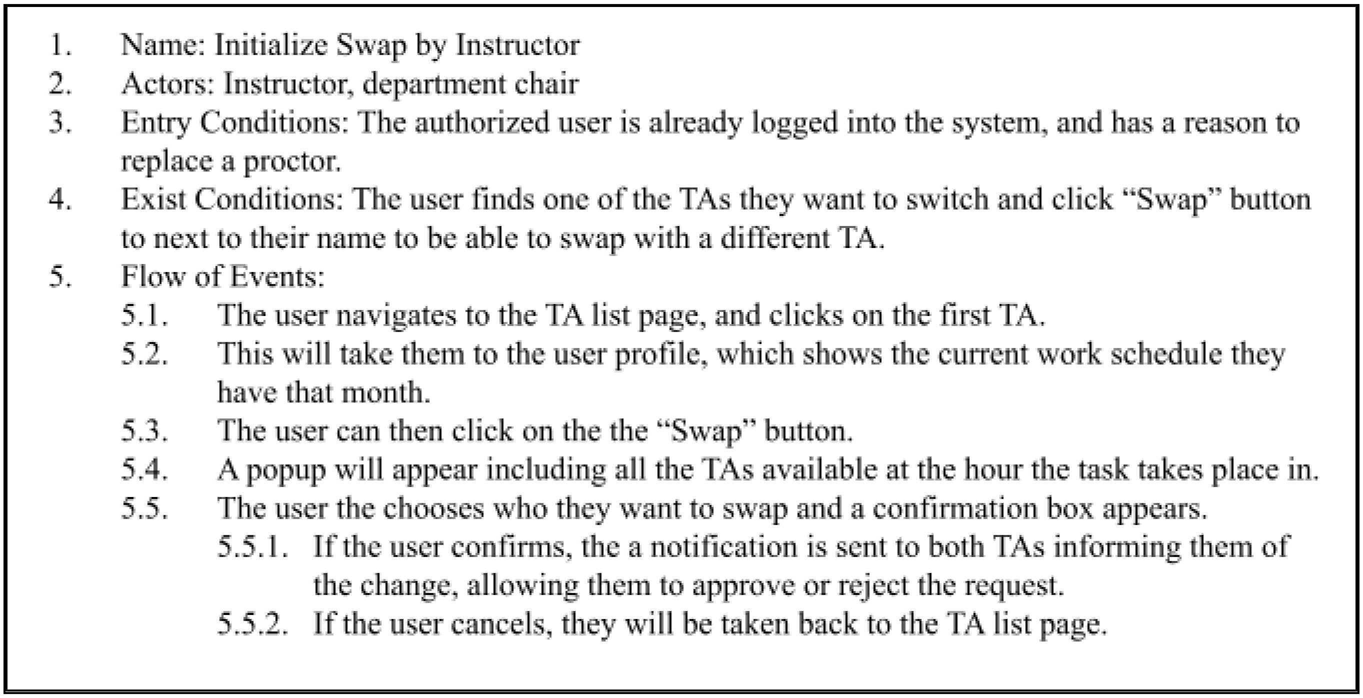
    5.3. The system will compile a list of other TAs who are free at the time of the original task. A “Swap” button with will be next to the available TAs.

    5.4. When “Swap” is clicked, a popup notification asks the TA to confirm.

        5.4.1. If the user confirms, a notification will get sent to the chosen TA’s inbox to approve or reject the request.

        5.4.2. If the user cancels, they will be taken back to the “Swap” page.

2.5.2 Initiate Swap by Instructor (Staff)



1. Name: Initialize Swap by Instructor
2. Actors: Instructor, department chair
3. Entry Conditions: The authorized user is already logged into the system, and has a reason to replace a proctor.
4. Exist Conditions: The user finds one of the TAs they want to switch and click “Swap” button to next to their name to be able to swap with a different TA.
5. Flow of Events:

    5.1. The user navigates to the TA list page, and clicks on the first TA.

    5.2. This will take them to the user profile, which shows the current work schedule they have that month.

    5.3. The user can then click on the the “Swap” button.

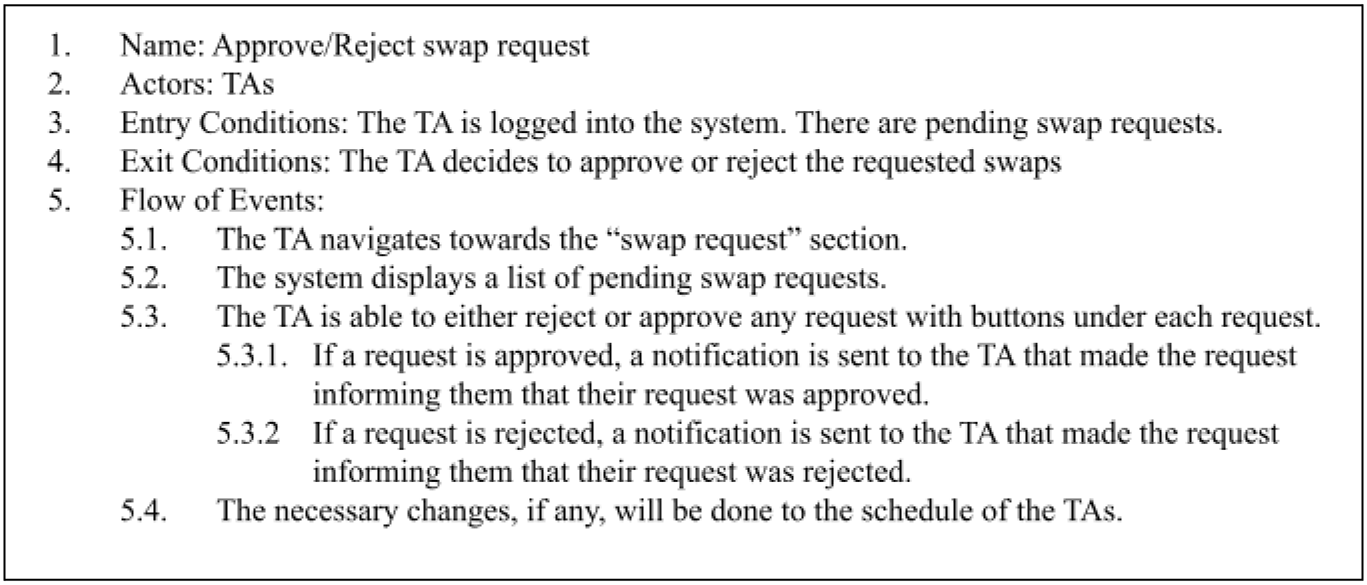
    5.4. A popup will appear including all the TAs available at the hour the task takes place in.

    5.5. The user the chooses who they want to swap and a confirmation box appears.

        5.5.1. If the user confirms, the a notification is sent to both TAs informing them of the change, allowing them to approve or reject the request.

        5.5.2. If the user cancels, they will be taken back to the TA list page.

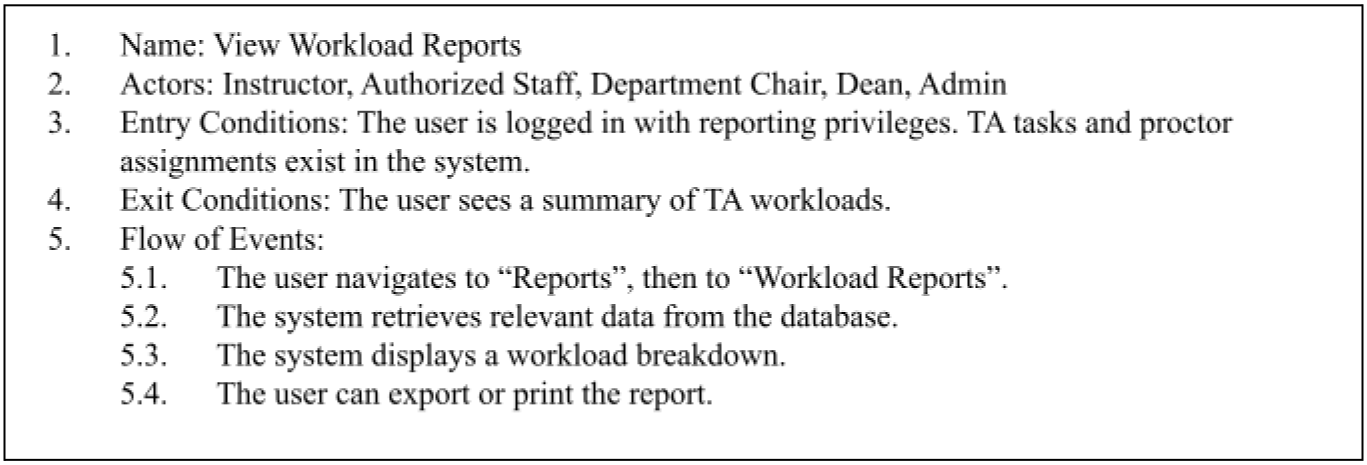
2.5.3 Approve or reject swap



1. Name: Approve/Reject swap request
2. Actors: TAs
3. Entry Conditions: The TA is logged into the system. There are pending swap requests.
4. Exit Conditions: The TA decides to approve or reject the requested swaps
5. Flow of Events:
    5.1. The TA navigates towards the “swap request” section.
    5.2. The system displays a list of pending swap requests.
    5.3. The TA is able to either reject or approve any request with buttons under each request.
        5.3.1. If a request is approved, a notification is sent to the TA that made the request informing them that their request was approved.
        5.3.2 If a request is rejected, a notification is sent to the TA that made the request informing them that their request was rejected.
    5.4. The necessary changes, if any, will be done to the schedule of the TAs.

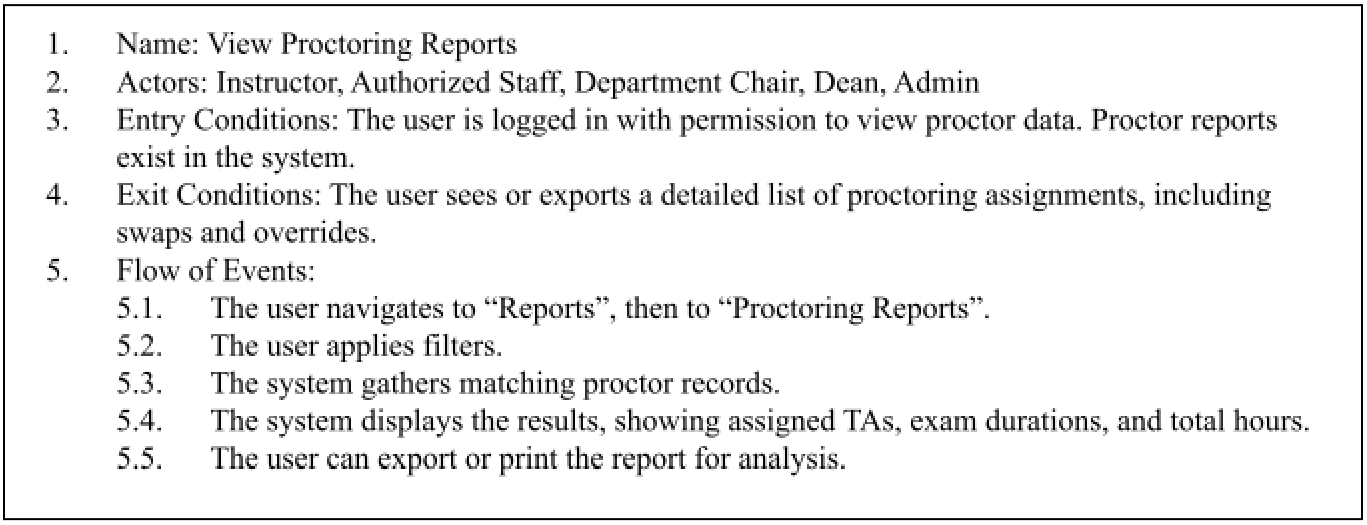
2.7 Reporting

2.7.1 View Workload Reports



1. Name: View Workload Reports
2. Actors: Instructor, Authorized Staff, Department Chair, Dean, Admin
3. Entry Conditions: The user is logged in with reporting privileges. TA tasks and proctor assignments exist in the system.
4. Exit Conditions: The user sees a summary of TA workloads.
5. Flow of Events:
    5.1. The user navigates to “Reports”, then to “Workload Reports”.
    5.2. The system retrieves relevant data from the database.
    5.3. The system displays a workload breakdown.
    5.4. The user can export or print the report.

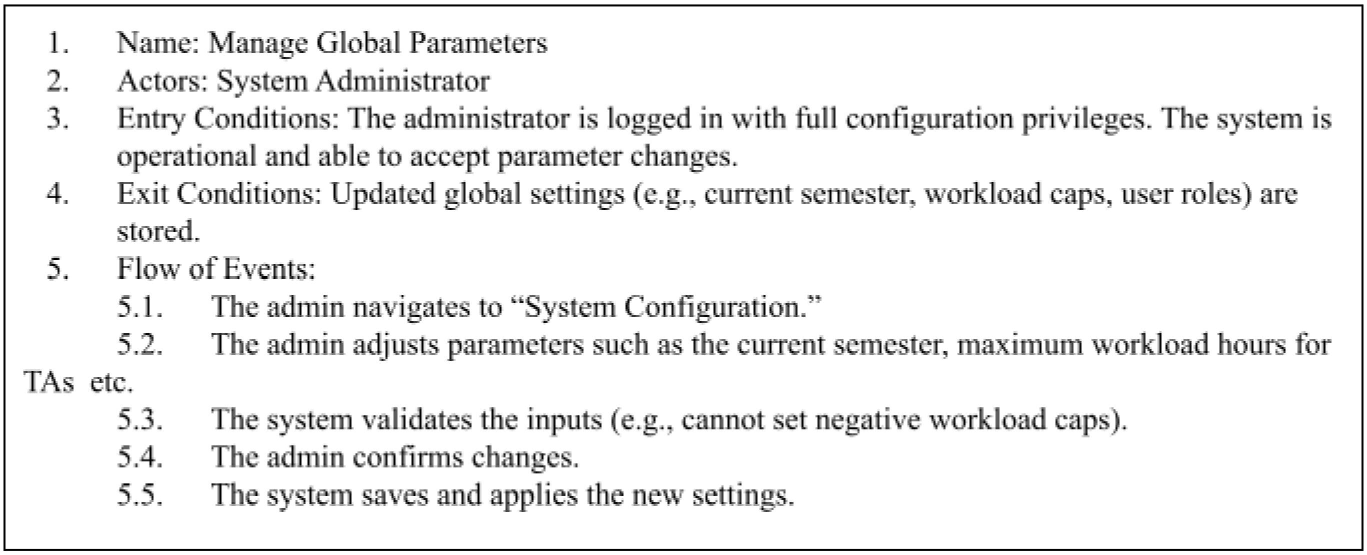
2.7.2 View Proctoring Reports



1. Name: View Proctoring Reports
2. Actors: Instructor, Authorized Staff, Department Chair, Dean, Admin
3. Entry Conditions: The user is logged in with permission to view proctor data. Proctor reports exist in the system.
4. Exit Conditions: The user sees or exports a detailed list of proctoring assignments, including swaps and overrides.
5. Flow of Events:
   - 5.1. The user navigates to “Reports”, then to “Proctoring Reports”.
   - 5.2. The user applies filters.
   - 5.3. The system gathers matching proctor records.
   - 5.4. The system displays the results, showing assigned TAs, exam durations, and total hours.
   - 5.5. The user can export or print the report for analysis.

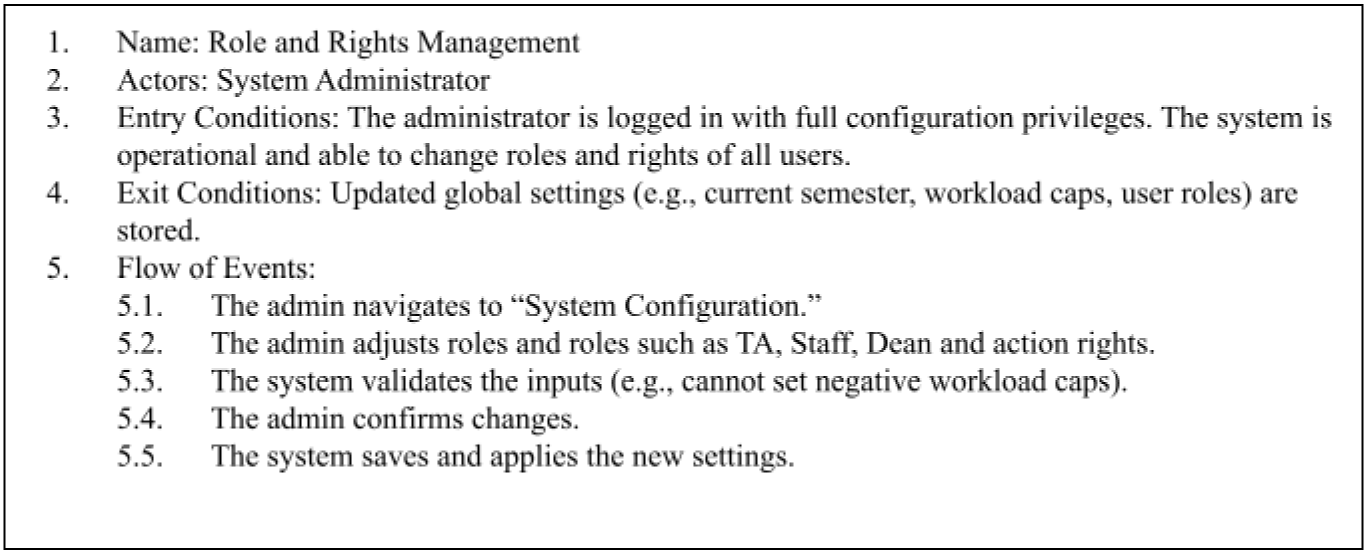
2.8 System Configuration

2.8.1 Manage Global Parameters



1. Name: Manage Global Parameters
2. Actors: System Administrator
3. Entry Conditions: The administrator is logged in with full configuration privileges. The system is operational and able to accept parameter changes.
4. Exit Conditions: Updated global settings (e.g., current semester, workload caps, user roles) are stored.
5. Flow of Events:
   - 5.1. The admin navigates to “System Configuration.”
   - 5.2. The admin adjusts parameters such as the current semester, maximum workload hours for TAs etc.
   - 5.3. The system validates the inputs (e.g., cannot set negative workload caps).
   - 5.4. The admin confirms changes.
   - 5.5. The system saves and applies the new settings.

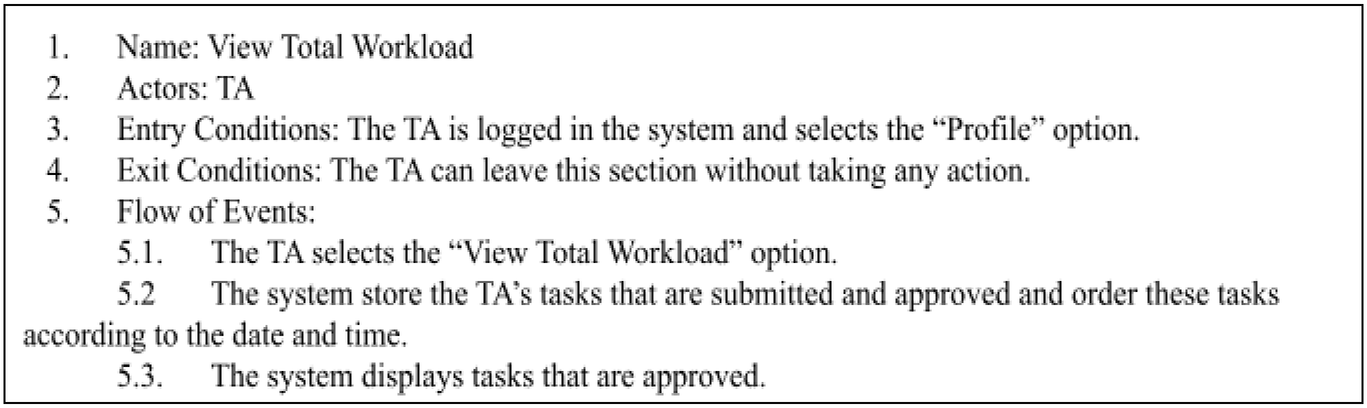
2.8.2 Role and Rights Management



1. Name: Role and Rights Management
2. Actors: System Administrator
3. Entry Conditions: The administrator is logged in with full configuration privileges. The system is operational and able to change roles and rights of all users.
4. Exit Conditions: Updated global settings (e.g., current semester, workload caps, user roles) are stored.
5. Flow of Events:
    5.1. The admin navigates to “System Configuration.”
    5.2. The admin adjusts roles and roles such as TA, Staff, Dean and action rights.
    5.3. The system validates the inputs (e.g., cannot set negative workload caps).
    5.4. The admin confirms changes.
    5.5. The system saves and applies the new settings.

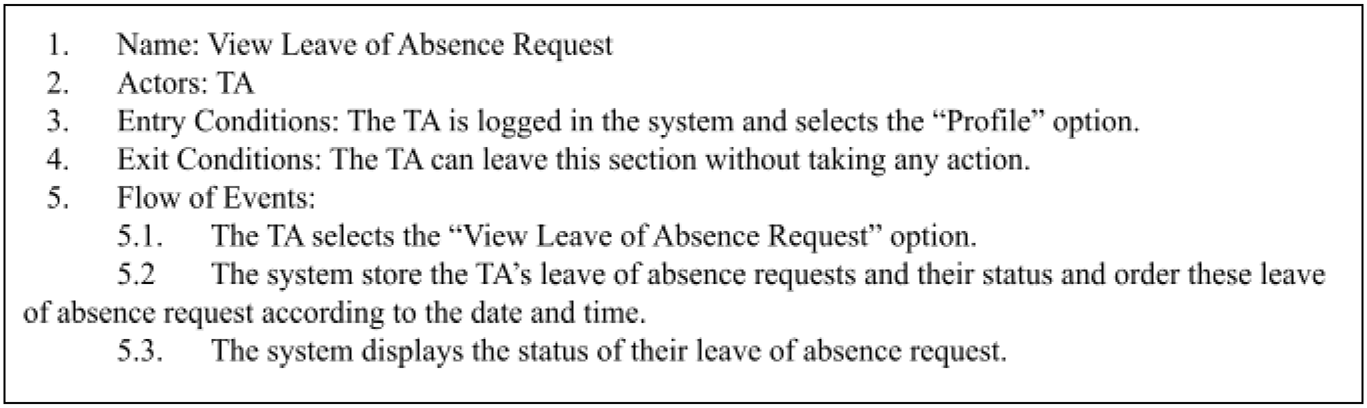
2.9 Profile View

2.9.1 View Total Workload



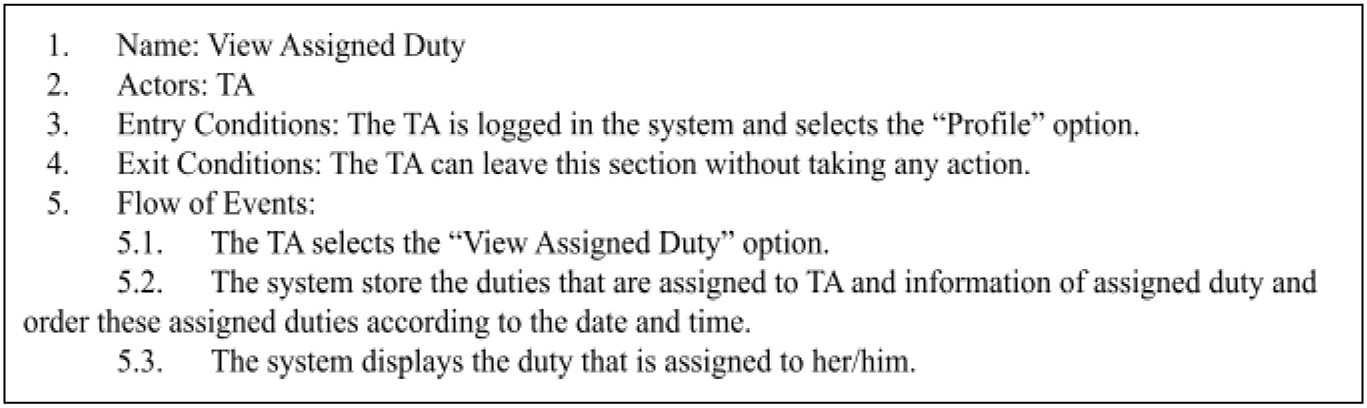
1. Name: View Total Workload
2. Actors: TA
3. Entry Conditions: The TA is logged in the system and selects the “Profile” option.
4. Exit Conditions: The TA can leave this section without taking any action.
5. Flow of Events:
    5.1. The TA selects the “View Total Workload” option.
    5.2 The system store the TA’s tasks that are submitted and approved and order these tasks according to the date and time.
    5.3. The system displays tasks that are approved.

2.9.2 View The Leave of Absence Request



1. Name: View Leave of Absence Request
2. Actors: TA
3. Entry Conditions: The TA is logged in the system and selects the “Profile” option.
4. Exit Conditions: The TA can leave this section without taking any action.
5. Flow of Events:
    5.1. The TA selects the “View Leave of Absence Request” option.
    5.2 The system store the TA’s leave of absence requests and their status and order these leave of absence request according to the date and time.
    5.3. The system displays the status of their leave of absence request.

2.9.3 View The Assigned Duty



1. Name: View Assigned Duty
2. Actors: TA
3. Entry Conditions: The TA is logged in the system and selects the “Profile” option.
4. Exit Conditions: The TA can leave this section without taking any action.
5. Flow of Events:
   5.1. The TA selects the “View Assigned Duty” option.
   5.2. The system store the duties that are assigned to TA and information of assigned duty and order these assigned duties according to the date and time.
   5.3. The system displays the duty that is assigned to her/him.

3. Technology Stack

The technology stack we are going to use in the project consists of React for the frontend, Express.js for the backend, and MySQL (mandatory) for the database.

3.1 React

React.js is a library included in JavaScript that helps users create frontend pages for their applications and projects. Its dynamic and responsive user interface capabilities are good for delivering an interactive experience in the TA Management System. It allows changes to be made to the page design and the data used to be implemented efficiently with any minor updates. Since it will be a nice framework for designing frontends, we will use it with all its convenience and popularity.

3.2 Express.js

Express.js is a framework included in Node.js that allows developers to create backend functionalities efficiently. It is chosen for backend development because of its lightweight and flexible framework. With Express.js, handling routes, managing HTTP requests, and integrating middleware is streamlined, making backend development more organized and less time-consuming. Express.js provides support that simplifies the implementation of complex business logic (such as automated proctor assignments and conflict resolution) and integrates smoothly with the frontend services.Since it enables smooth communication between the frontend and database, we'll use Express.js to ensure our project's backend operates with optimal performance and reliability.

3.3 MySQL

MySQL is the mandatory relational database system known for its reliability, performance, and ease of use. By using MySQL, we can store and retrieve data efficiently while maintaining the integrity of our application's data. It supports structured queries and has a strong community. MySQL's mature ecosystem and proven scalability make it an ideal choice for ensuring data integrity and performance in this application.