Department of Computer Engineering, Bilkent University
CS319 Object Oriented Software Engineering
Group 11
NFR, Various Diagrams and Mockups Report - D2
Section 1 - Eray Tüzün
16/03/2025

Furkan Mert Aksakal 22003191
Furkan Komaç 22102165
Alper Biçer 22003097
Erkan Can Arslan 22103948
Deniz Yazıcı 21902557

Contents

1. Non-Functional Requirements

1.1 Usability Requirements

Role-Specific Dashboards: Each user (TA, Instructor, Secretary, Admin, etc.) will see a role-specific dashboard relevant to their job.

Responsive UI: Frontend will be fully responsible across different devices such as desktop, tablet and mobile platforms.

Real-Time Notifications: Duty assignment, approval, and swap request notifications will be shown in real-time (≤ 2 seconds) with a visual cue (e.g., badge icon).

Duty Swap System: TAs will have an easy-to-use interface to request, accept, or reject proctor duty swaps. All swap interactions will be preceded by confirmation dialogs for clarity.

1.2 Security/Safety Requirements

Authentication: All users must authenticate via secure login (username/password). Passwords will be securely stored.

Authorization: Role-Based Access Control (RBAC) is where users are restricted to seeing only features/data depending on their role.

Data Protection: Frontend-backend communications will be encrypted with HTTPS. Sensitive user data (e.g., personal data, logs) will be restricted access.

Audit Logging:Admins can see logs of significant system activities (e.g., duty creation, user management, data imports). Logs cannot be edited and are stored securely.

Account Management: Only admins allowed to CRUD (create / load / edit /delete) users. Forgotten passwords will be reset securely via email verification and token-based authentication.

1.3 Performance Requirements

API Response Time: 95% of all backend requests will respond within 2 seconds under normal load conditions**.**

Concurrent Users: The system must be able to handle a minimum of 200 concurrent users without a performance drop.

Bulk Import Processing: Excel imports of data (e.g., TA list, classroom data) of 10,000 records or less will be processed within 30 seconds.
Page Load Time: All core pages (e.g., dashboard, duty management) will load in under 5 seconds on standard wireless connections.

1.4 Reliability Requirements

System Uptime: The backend system (Django + MySQL) will have ≥ 99.5% uptime, if there is no scheduled maintenance work.

Data Backups: Routine every-day automated back-ups of core data (responsibilities, users, logs) will be preserved, and the recovery should be possible within 24 hours when data is missing.

Conflict Prevention: Scheduling conflicts (concurrent responsibilities, double bookings) will be prevented by checking for them upon assigning responsibilities.

Failover Handling: In case of API or database failure, the system will notify users and retry automatically or provide fallbacks.

1.5 Maintainability Requirements:

Modular Architecture: Django's MTV pattern and React component structure make it possible for code to be easily maintained. Every module (e.g., duty management, logging) is isolated so that updates are made separately.

Code Documentation: Backend APIs will be documented using Swagger, while frontend components will be provided with inline comments and JSDoc as and when needed.

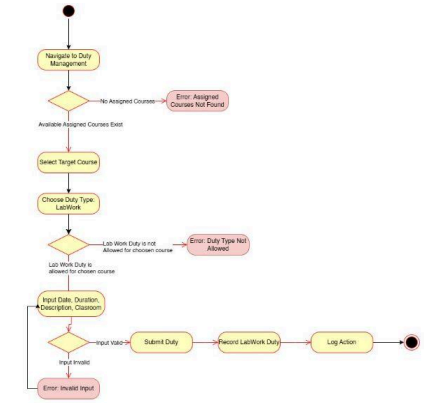
Version Control: Source code will be stored on GitHub with branching (development, staging, production) and pull request workflows.

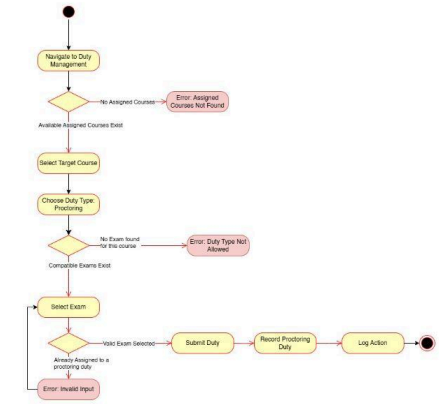
Extensibility: The system will accommodate future extensions of new duty types, roles, and reporting functionality with minimal interruption to existing functionality.

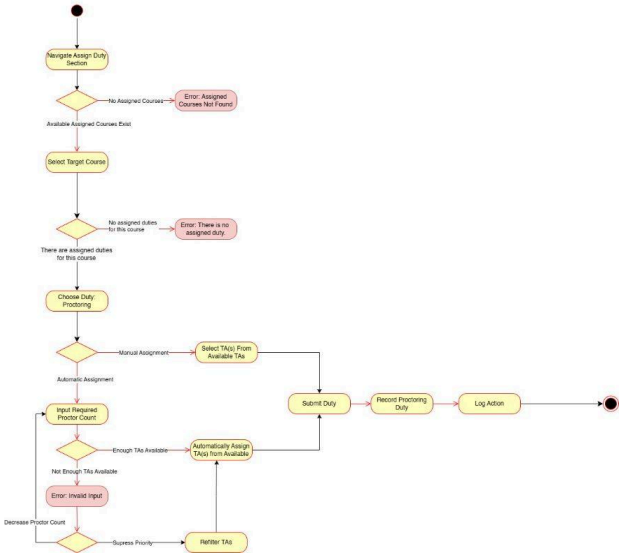
2. Activity Diagrams

For better quality: click here

1. Activity Diagram: Creating the lab work duty.
2. Activity Diagram: Creating the proctoring duty.
3. Activity Diagram: Assign proctoring duty.



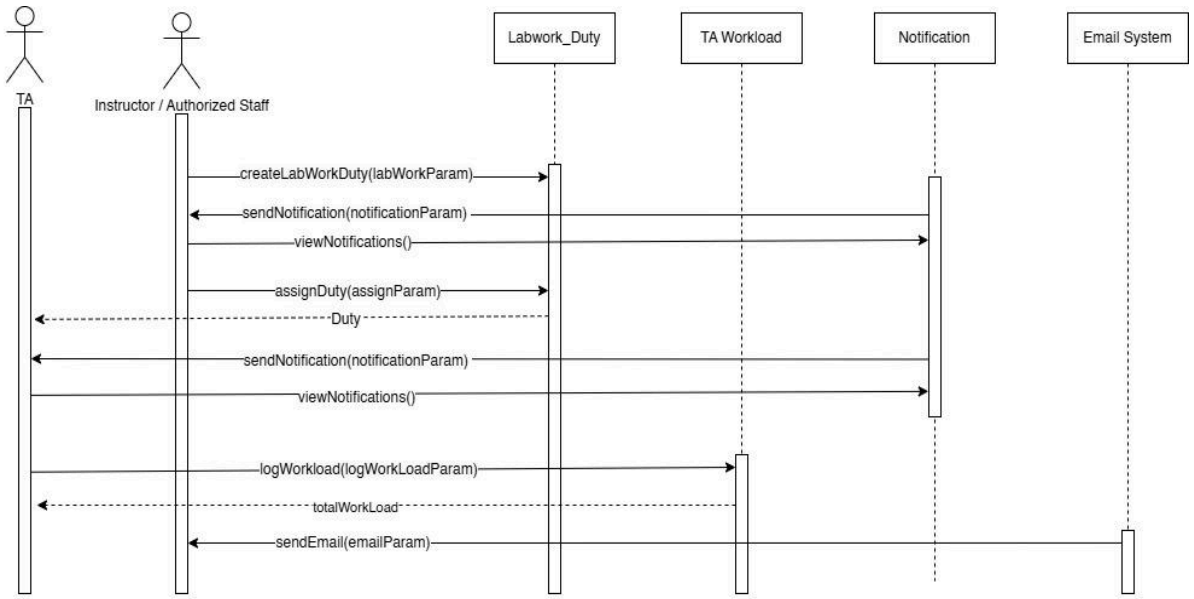


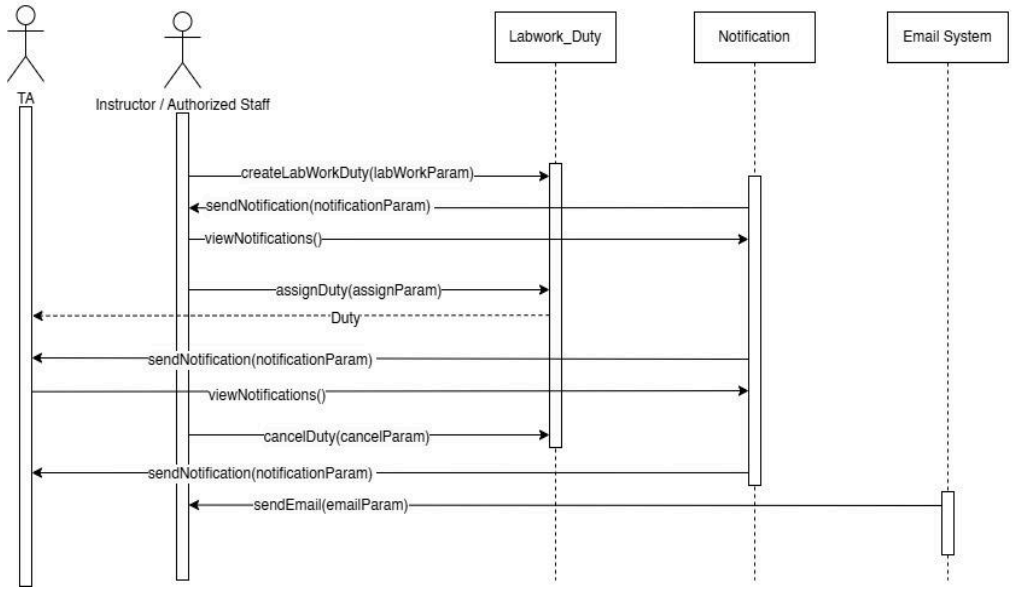


3. Sequence Diagrams

For better quality: [click here]

1. Sequence Diagram: Creating, assigning the lab work duty and entering duty workload.
2. Sequence Diagram: Creating, assigning and canceling lab work duty.





Note: First sequence diagram illustrates the process of creating a lab work duty, assigning it to a TA, and marking the corresponding workload. Notifications are sent after both creation and assignment of the duty, and are visible to the TA. Once the TA marks the workload, the total workload is fixed and a confirming email is sent. This diagram does not cover cancellation or modification of duties. Second sequence diagram shows the creation, assignment, and subsequent cancellation of a lab work task. Both creation and cancellation operations trigger notifications, which are visible to the TA. After cancellation, an informatory email is sent to the TA. Workload logging procedure is not shown in this diagram.

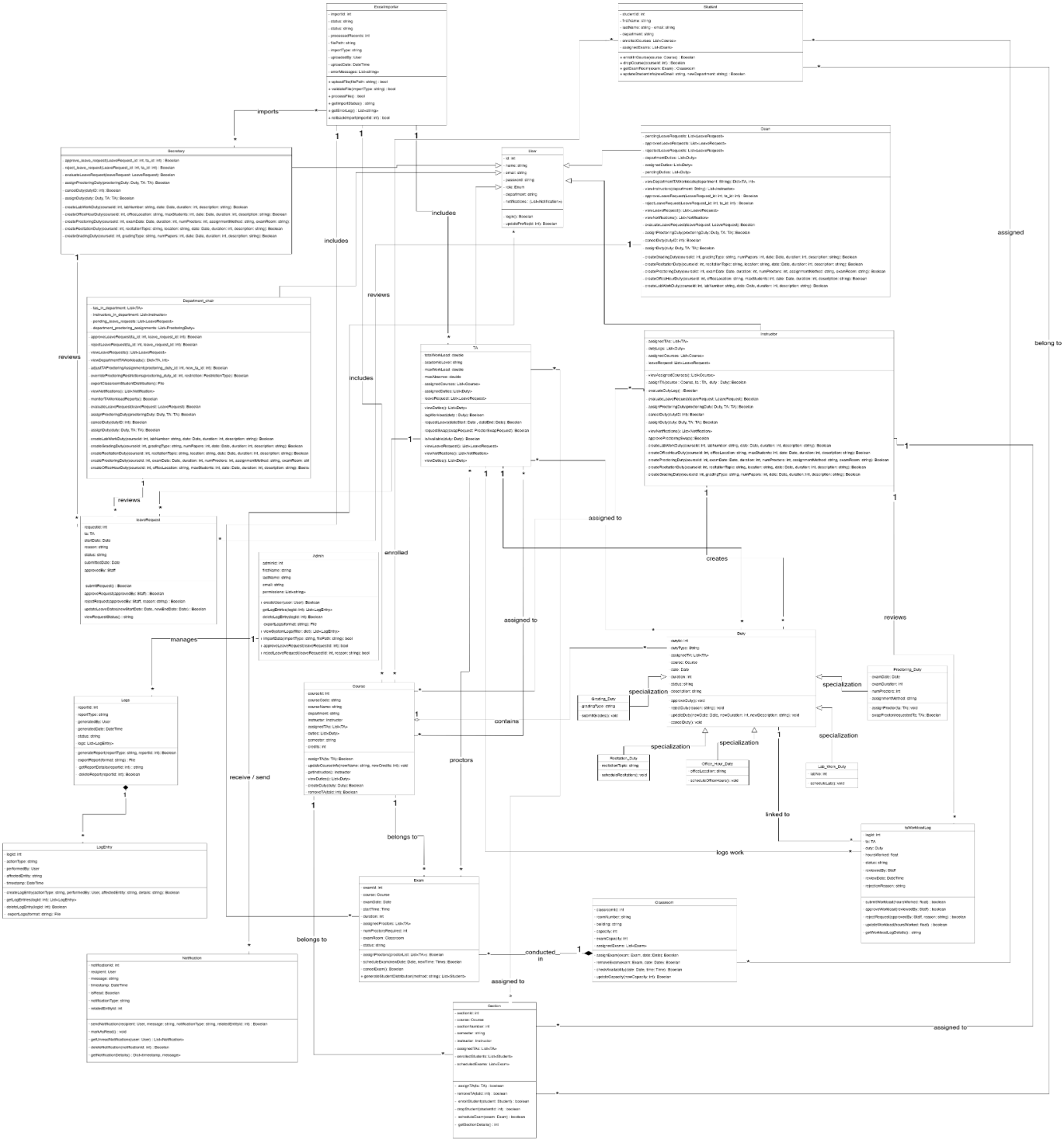
Authorized Staff means Dean, Department Chair, Secretary besides Instructor.

Parameters in the Sequence Diagram:

- **labWorkParam:** courseId: int, labNumber: string, date: Date, duration: int, description: string
- **assignParam:** duty: Duty, TA: TA
- **notificationParam:** recipient: User, message: string, notificationType: string, relatedEntityId: int
- **logWorkLoadParam:** duty: Duty
- **emailParam:** receiver: String, sender: String, message: String, attachment: File
- **cancelParam:** dutyID: int

4. Class Diagram

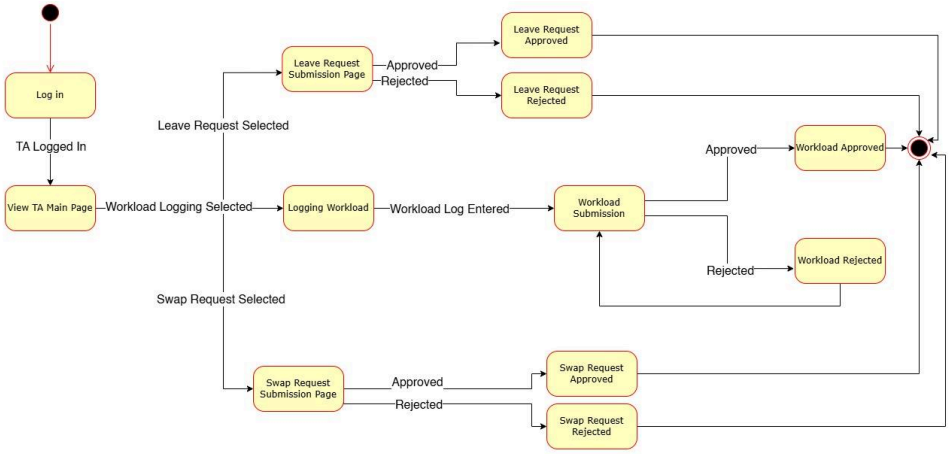
For better quality: [click here]

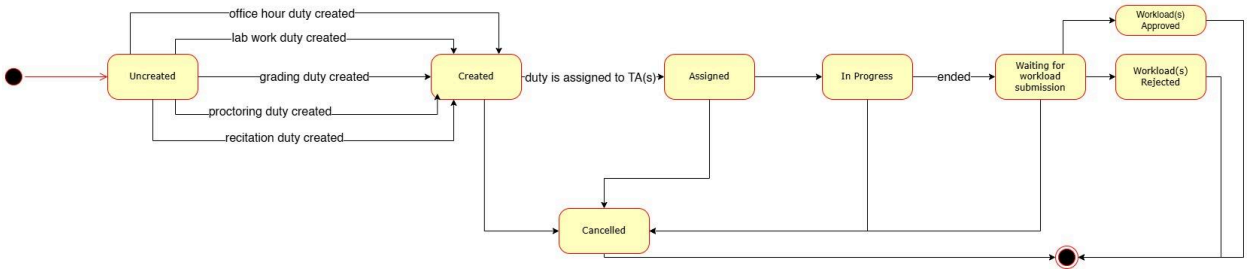


5. State Diagrams

For better quality: [click here]

1. State Diagram: All states of the user TA.
2. State Diagram: All states of a duty.





Note: First diagram shows the workflow available to a TA when logged into the system. The TA can visit the main page and select one of three options: leave request, workload logging, or swap request. Each of these requests gets approved or denied, leading to respective end states. If the workload log gets denied, the TA can submit the log again. Second diagram illustrates the lifecycle of different responsibilities like office hour, lab work, grading, proctoring, and recitation. Responsibilities get uncreated before becoming created, then afterwards either assigned to TAs or canceled. Assigned responsibilities get in progress before waiting for workload submission. Submitted workloads are approved or denied. Canceled responsibilities do not proceed to workload submission and are therefore eliminated from the lifecycle. This chart does not have duty editing or reassignment after being created.

6. Mockups

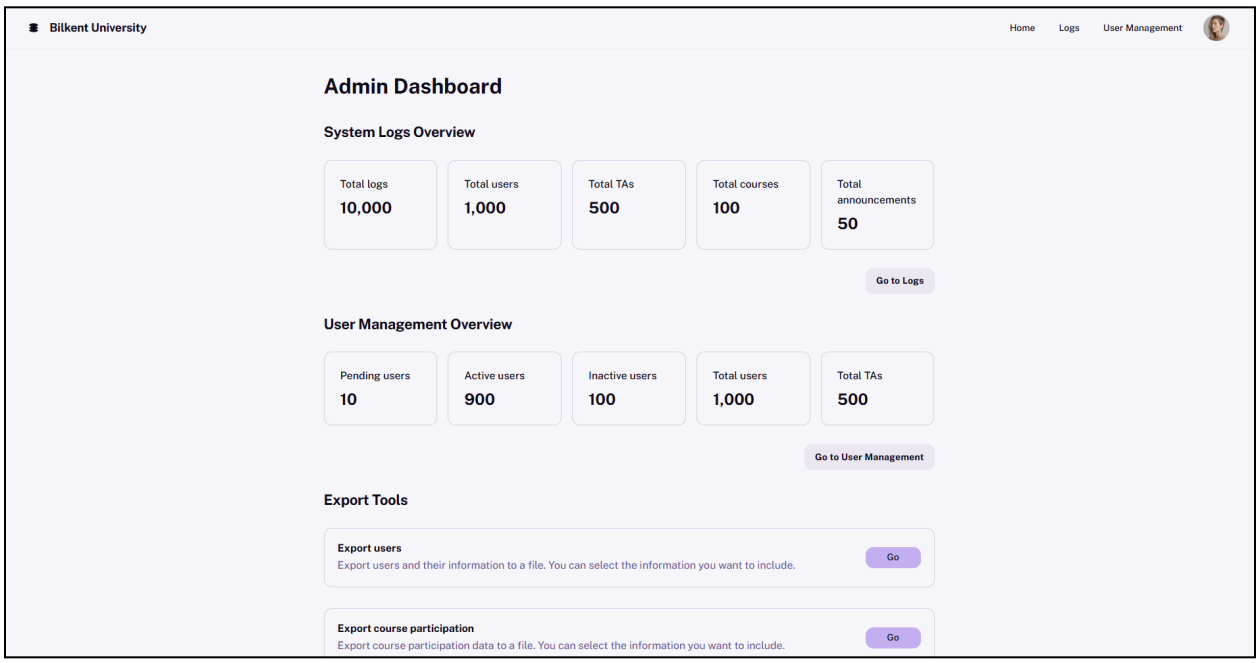


Figure 1: Admin Home Page

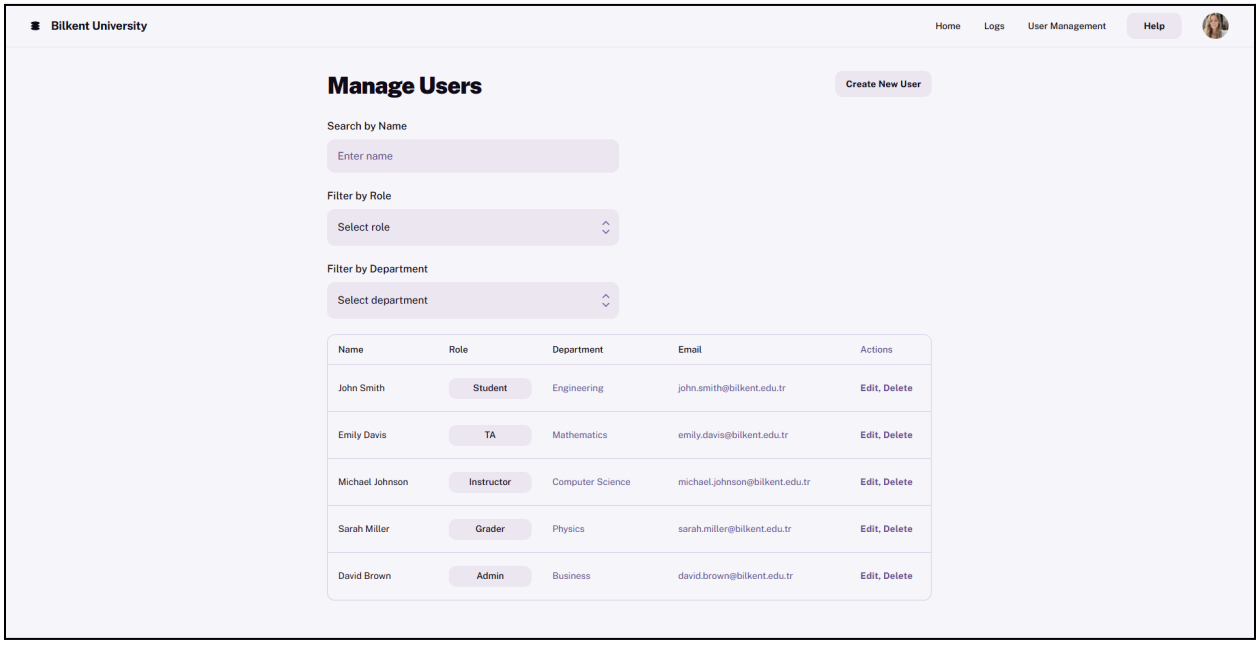


Figure 2: Admin Manage User Page

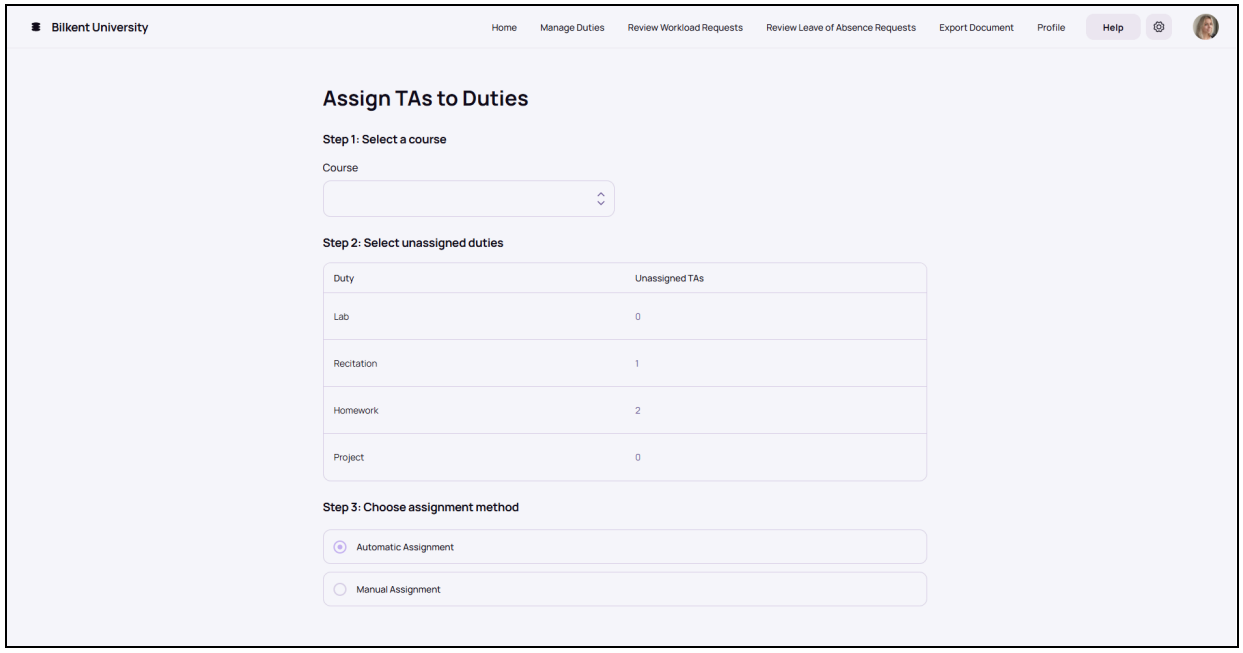


Figure 3: Assign Duty Main Page

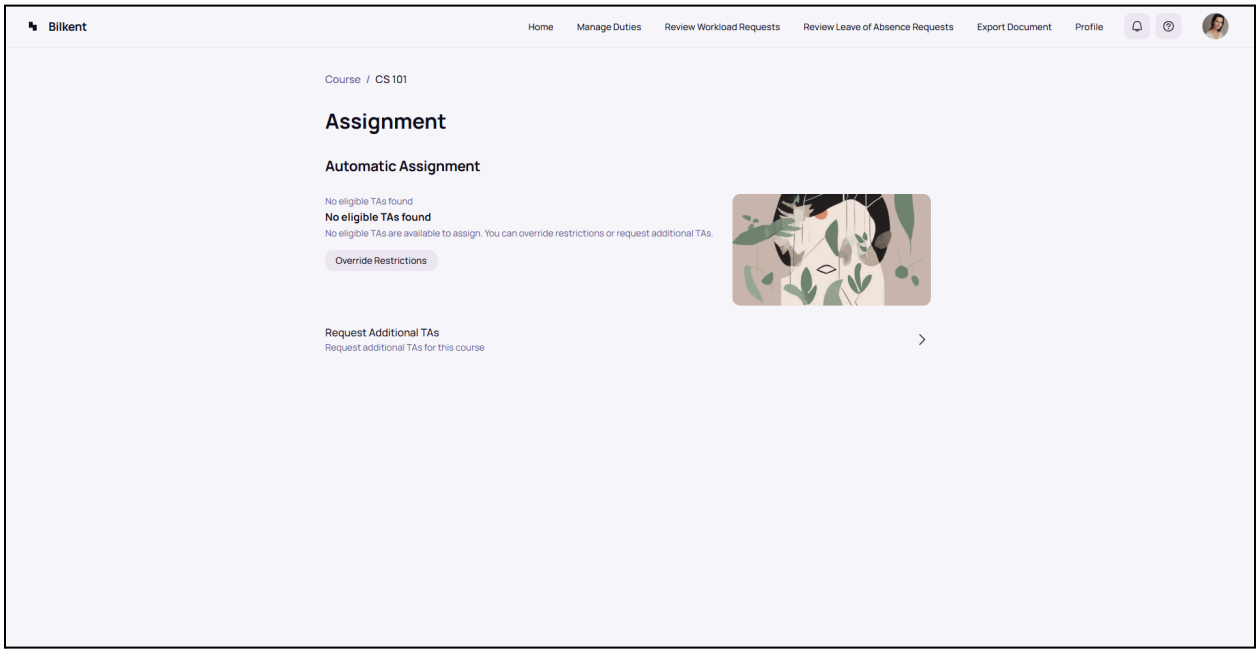


Figure 4: Automatic Assignment Fail Page

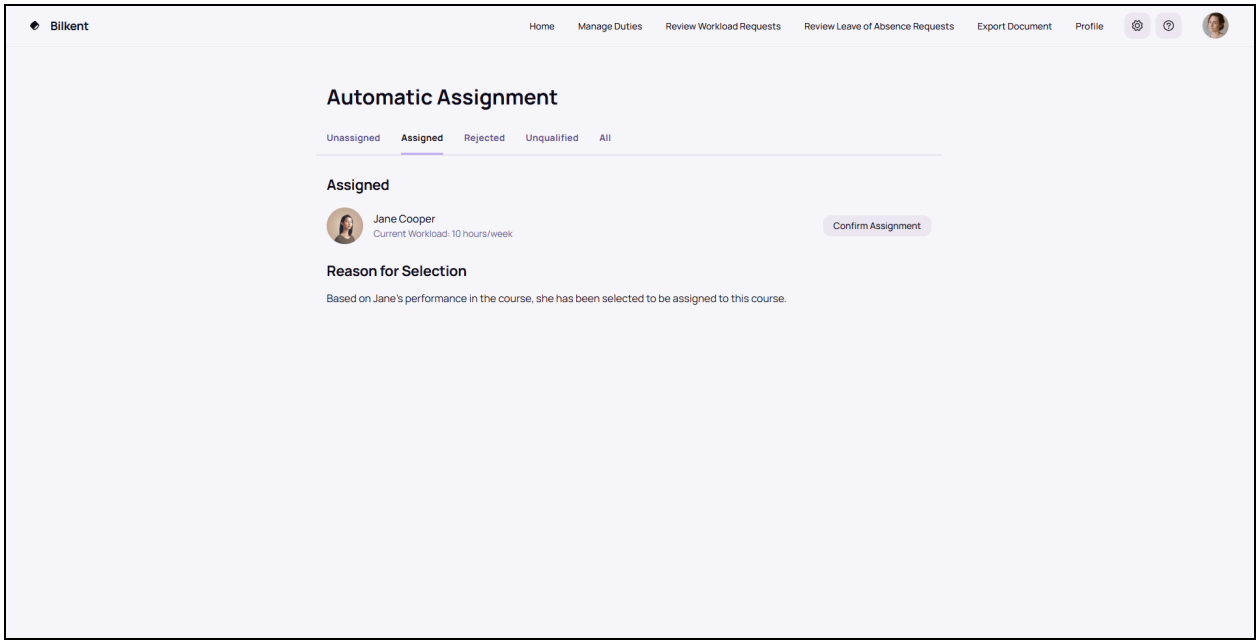


Figure 5: Automatic Assignment Success Page

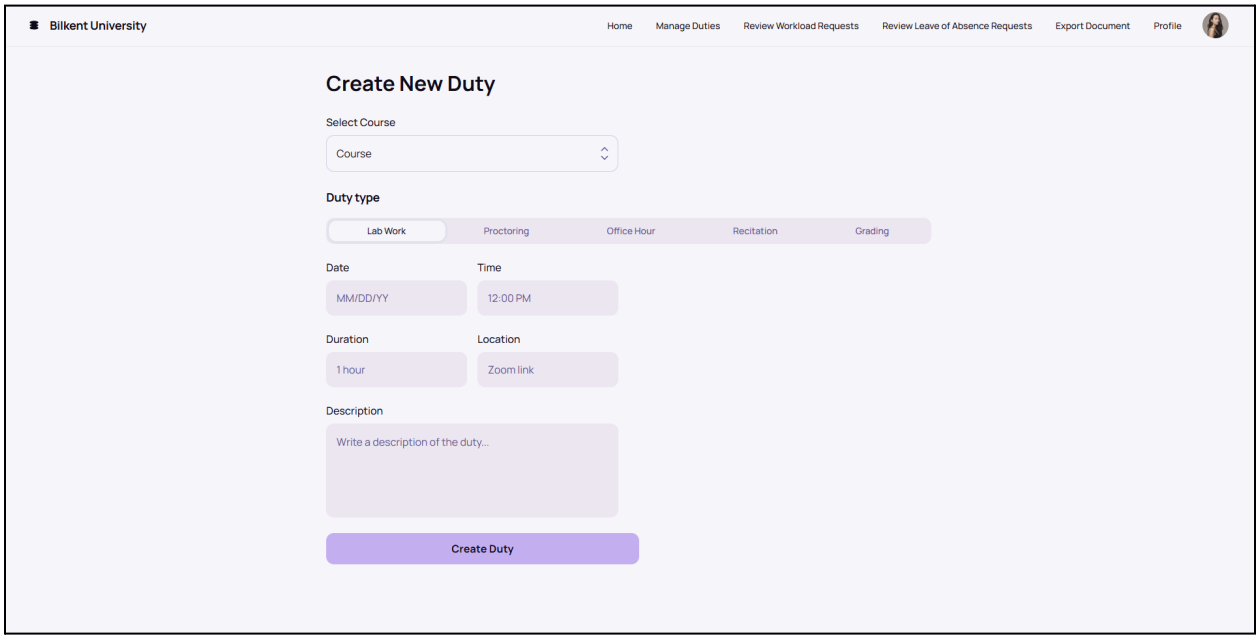


Figure 6: Create Duty Page

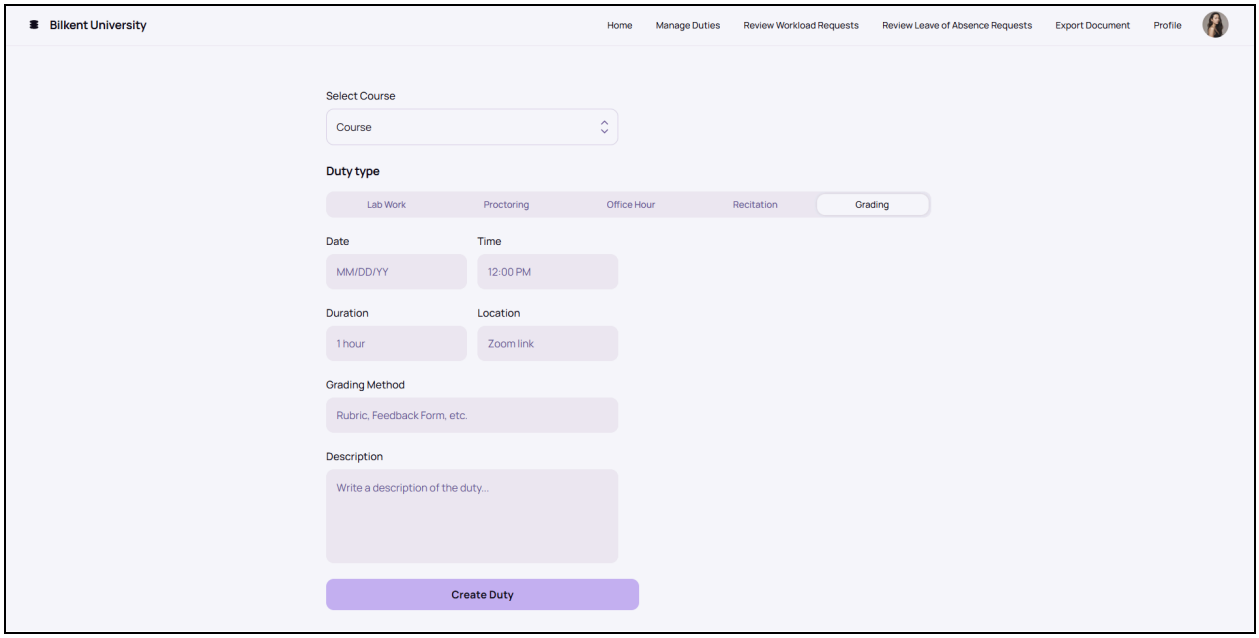


Figure 7: Create Grading Session Page

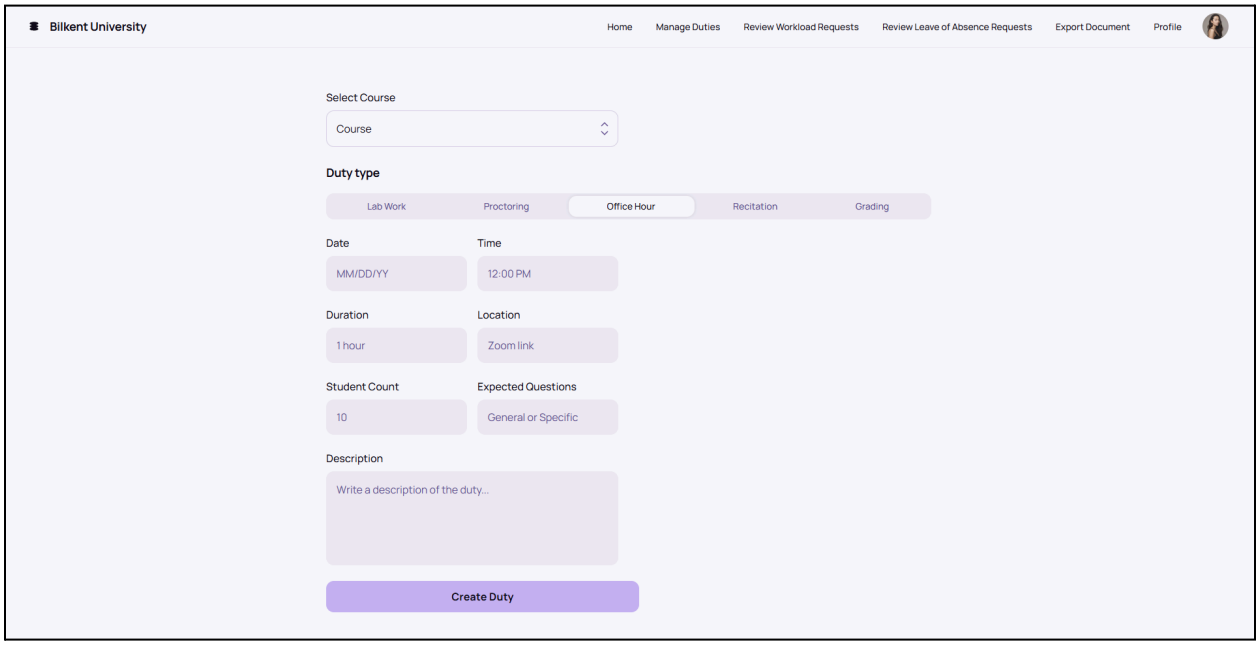


Figure 8: Create Office Hour Page

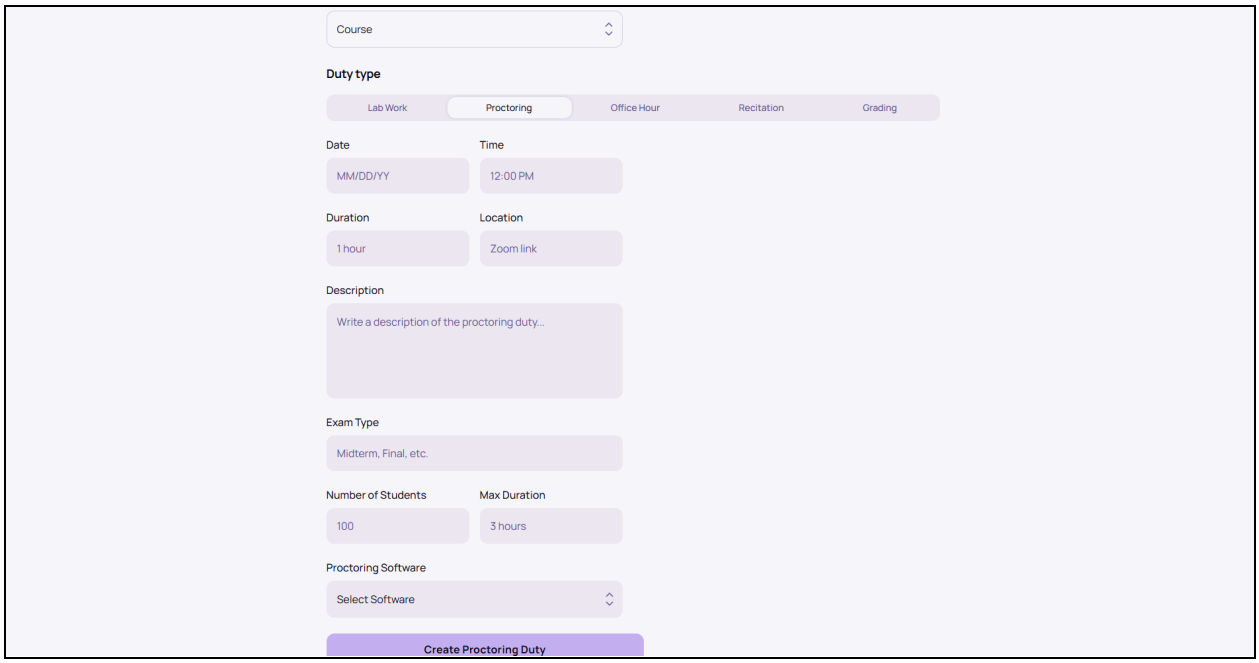


Figure 9: Create Proctoring Page

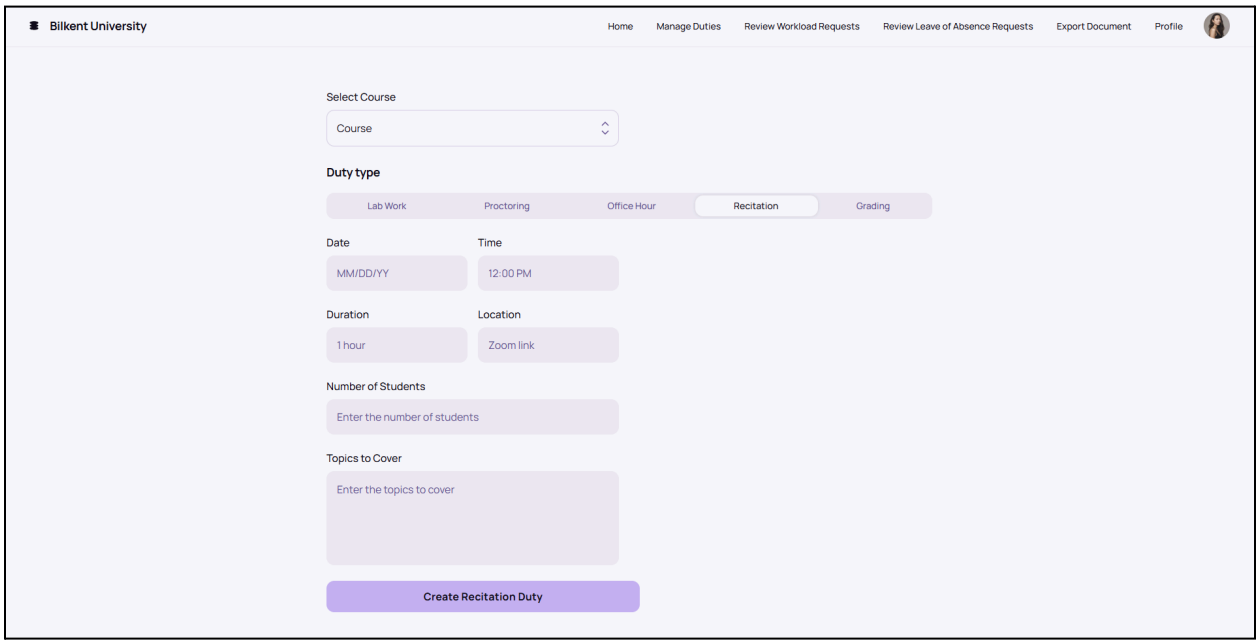


Figure 10: Create Recitation Page

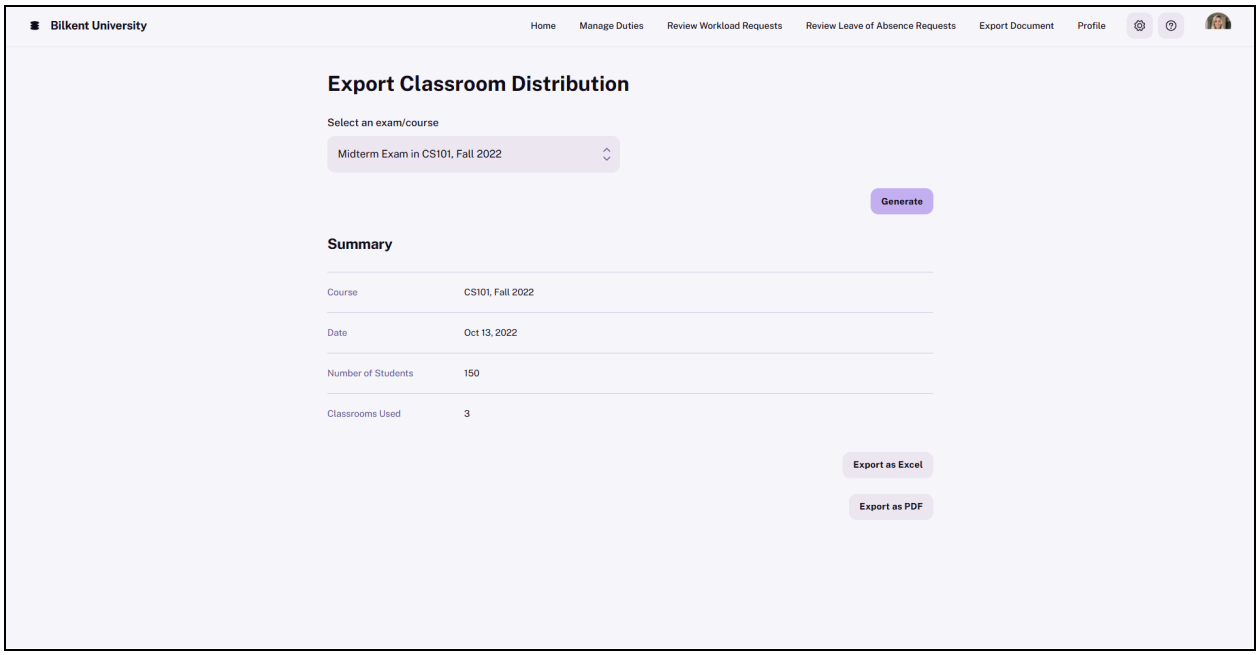


Figure 11: Export Distribution Page

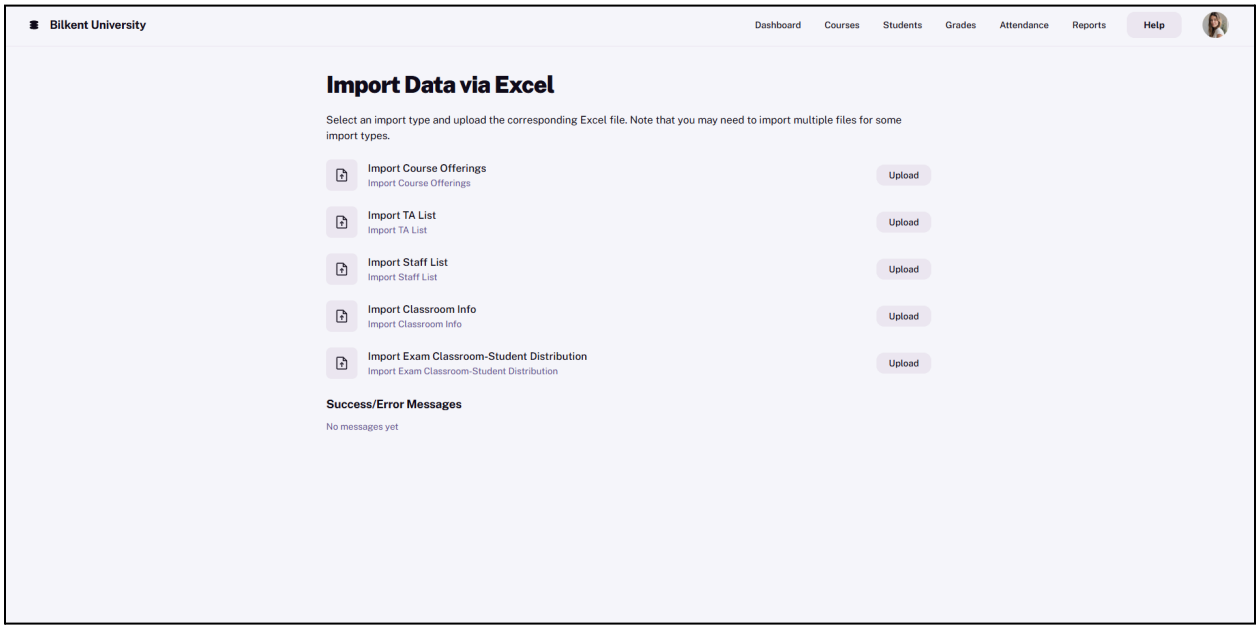


Figure 12: Import Excel Page

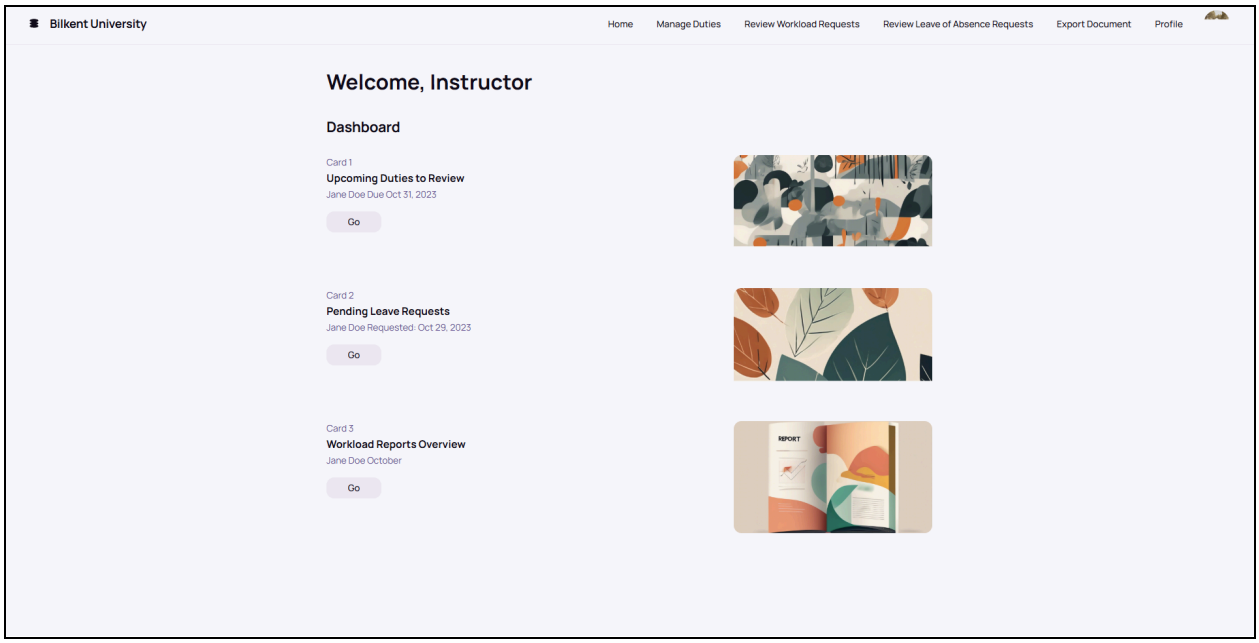


Figure 13: Instructor Home Page

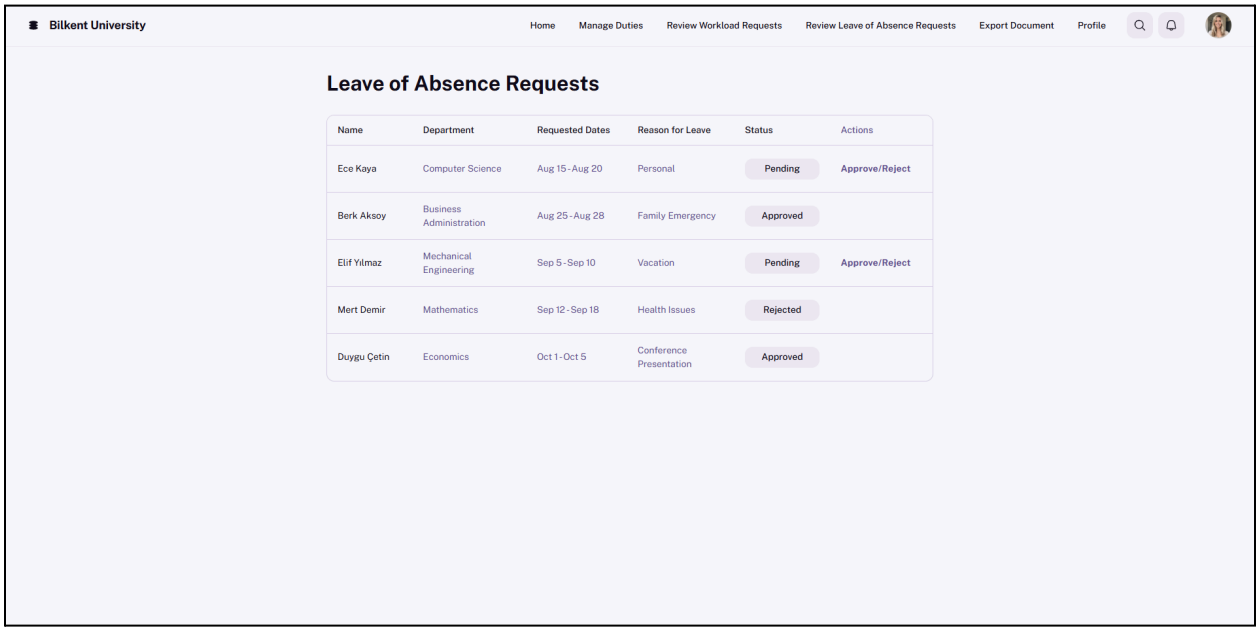


Figure 14: Leave of Absence Review Page

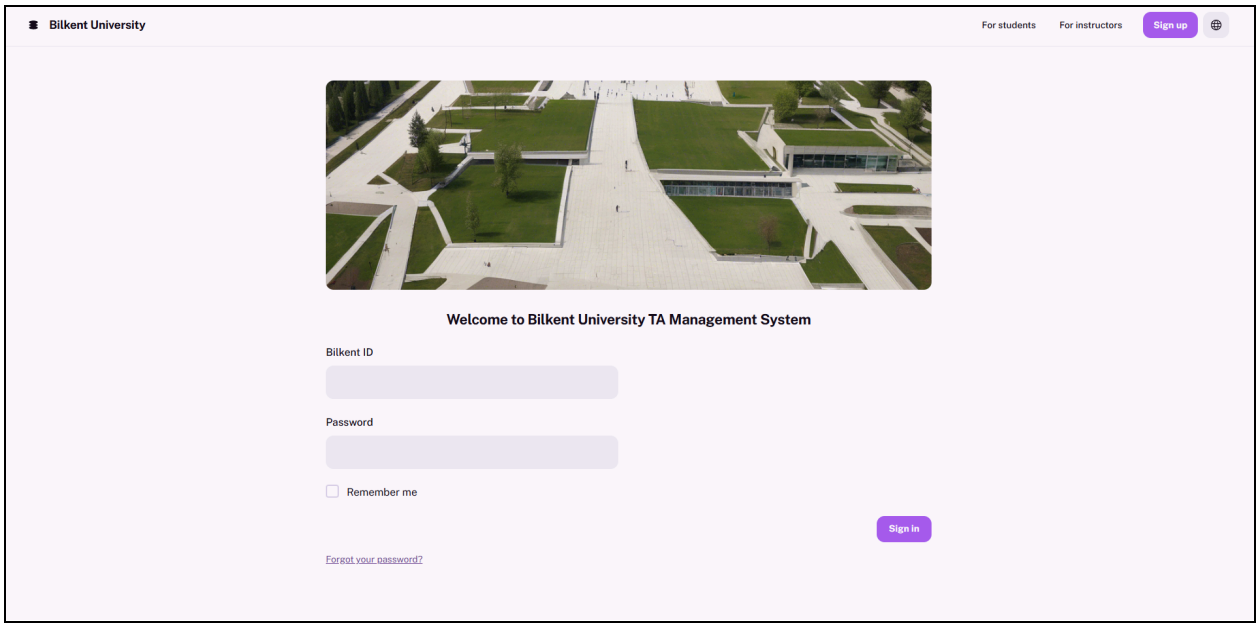


Figure 15: Login Page

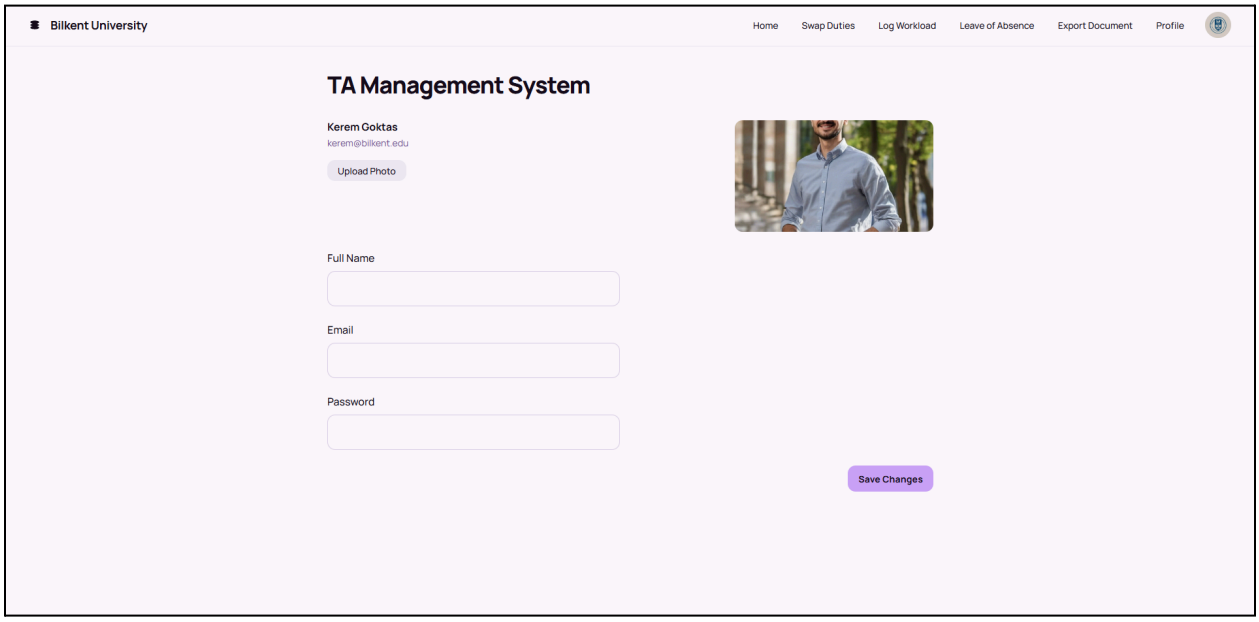


Figure 16: Manage Profile Page

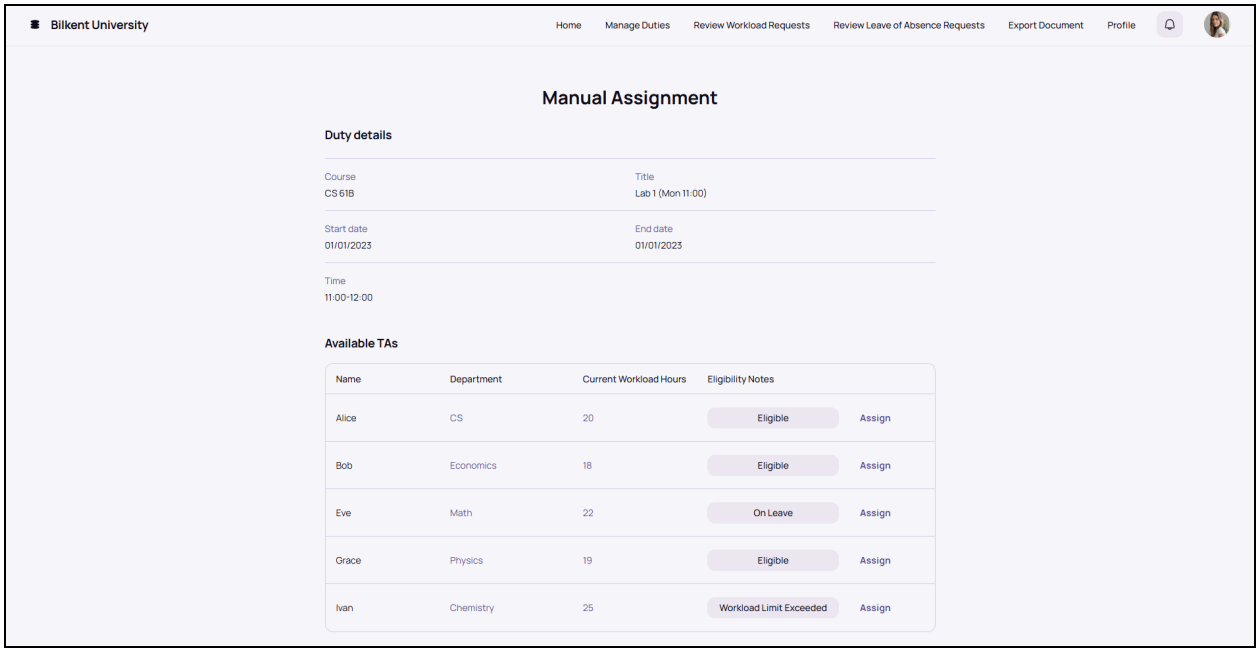


Figure 17: Manual Assign Page

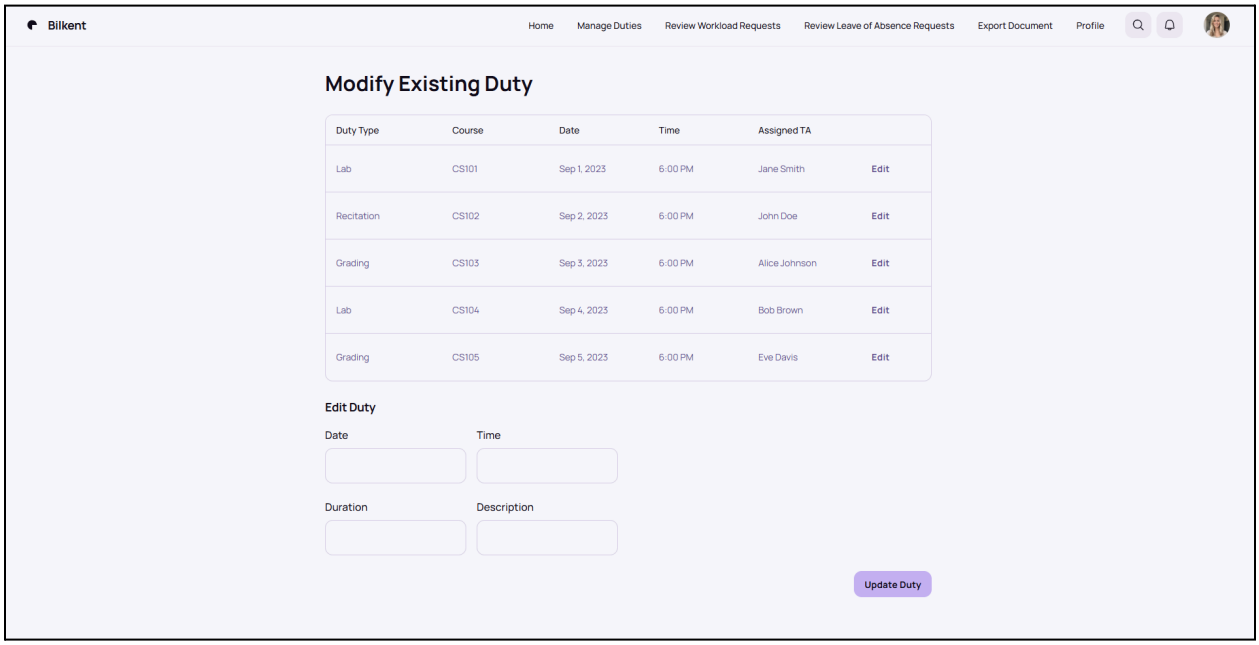


Figure 18: Modify Duty Page

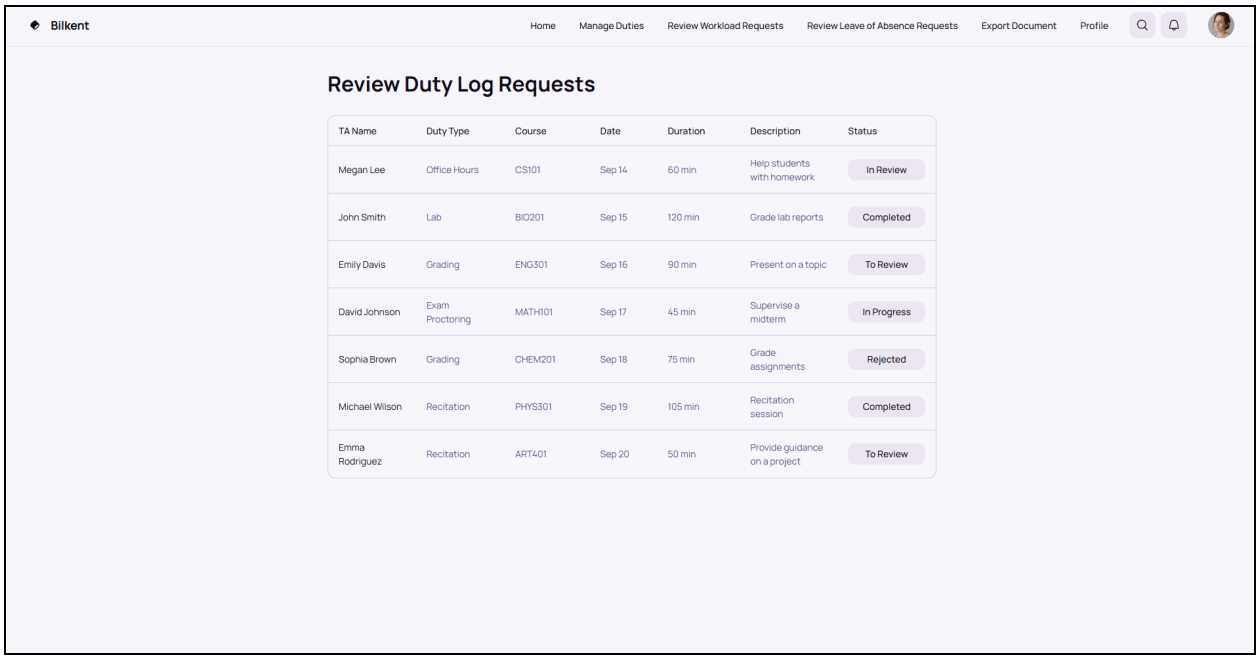


Figure 19: Review Duty Logs Page

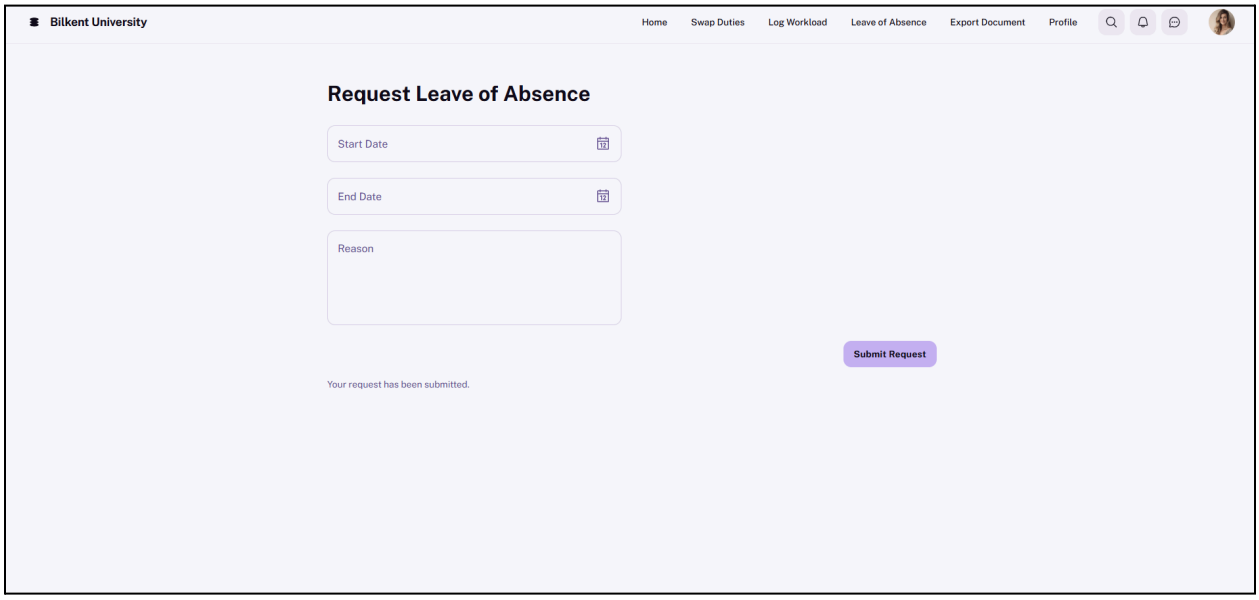


Figure 20: TA Absence Request Page

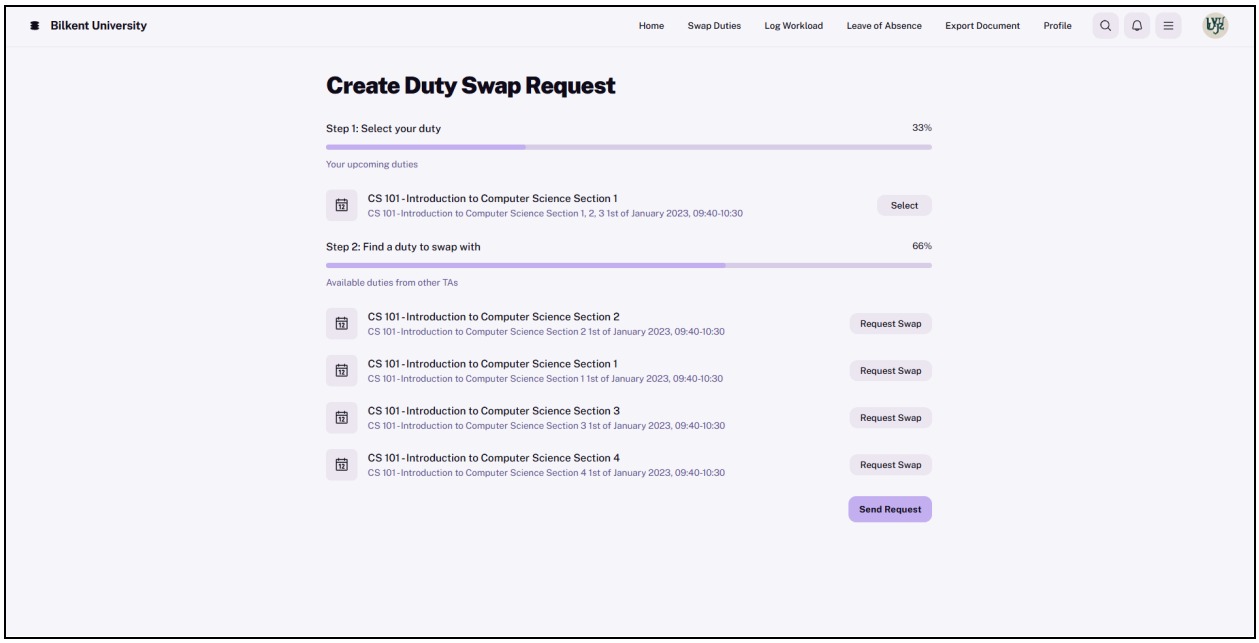


Figure 21: TA Create Swap Page

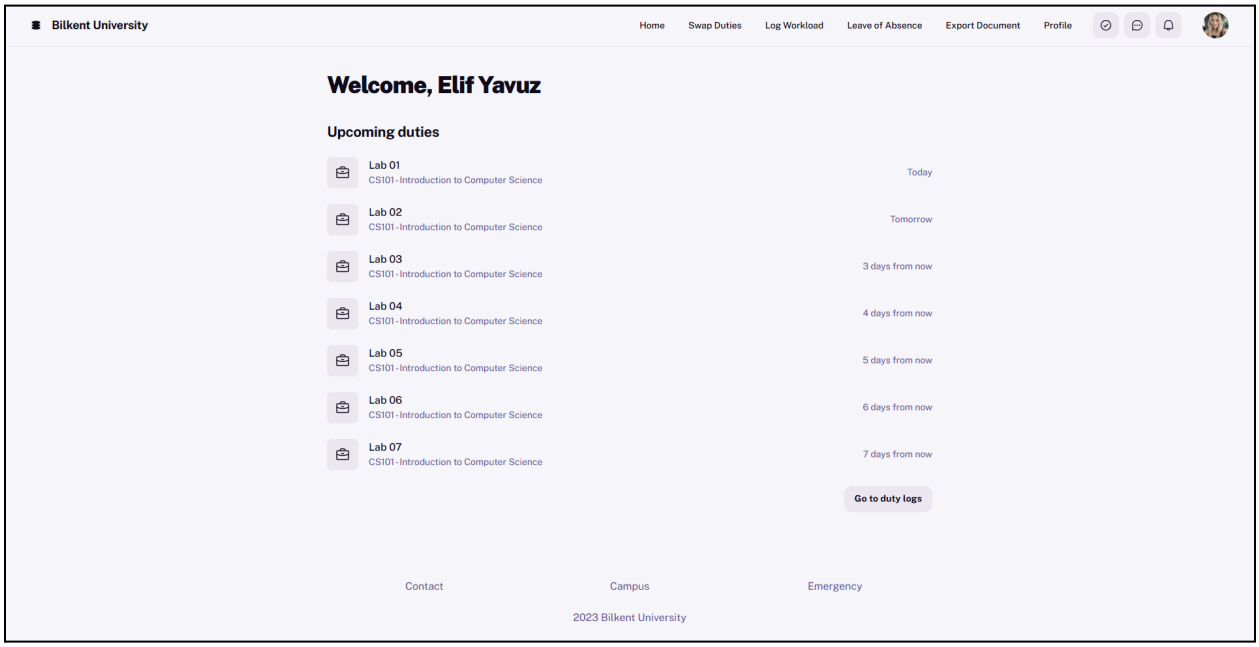


Figure 22: TA Home Page

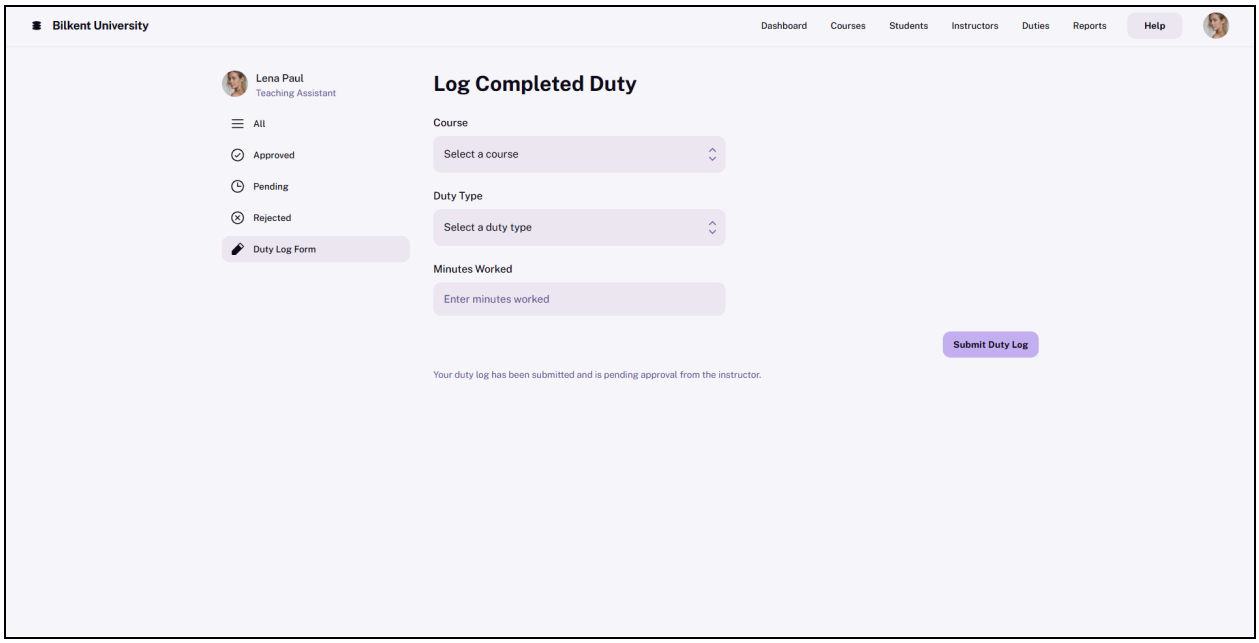


Figure 23: TA Before Logging Duty Page

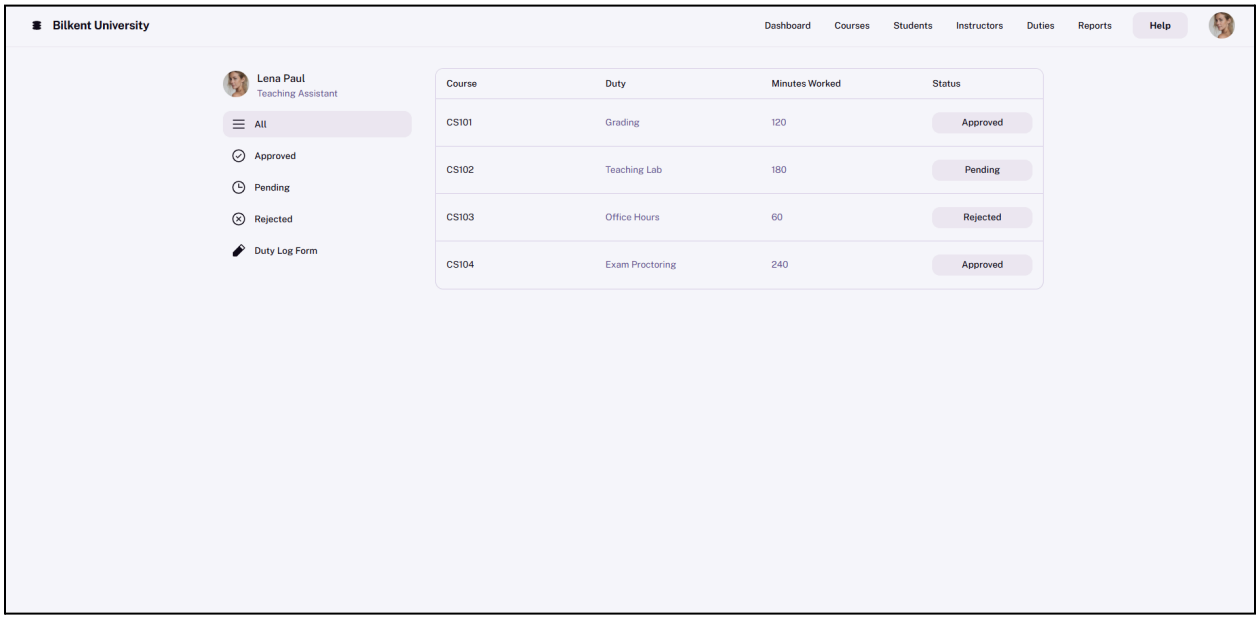


Figure 24: TA Log Duty Page

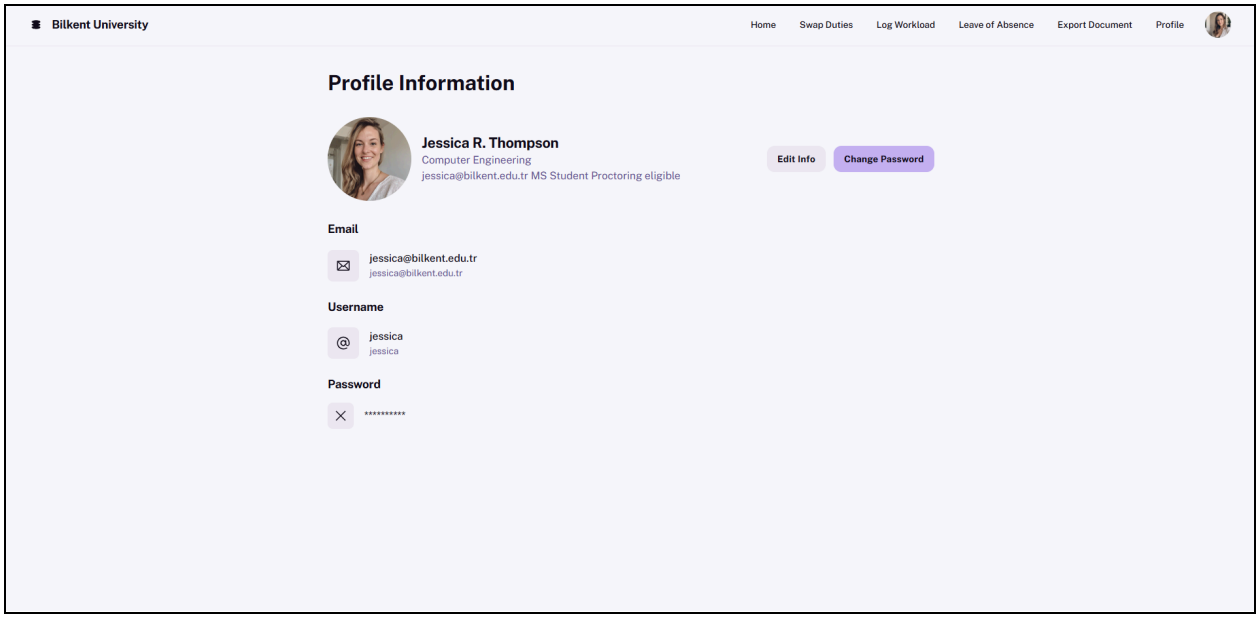


Figure 25: TA Profile Info Page

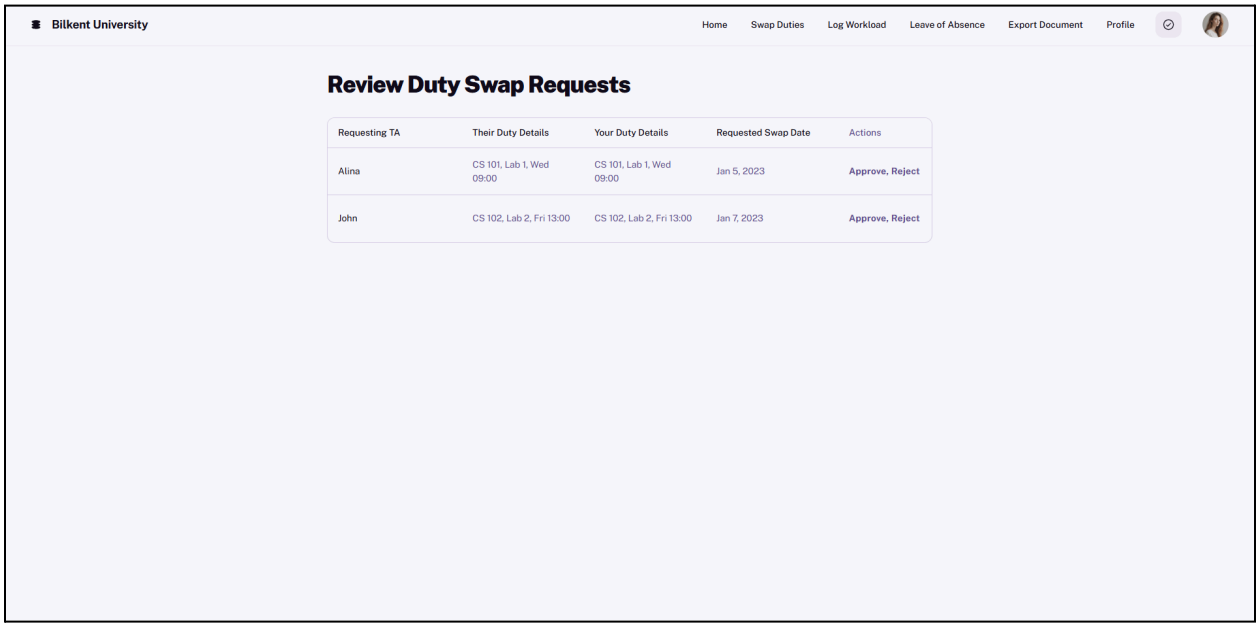


Figure 26: TA Review Swaps Page