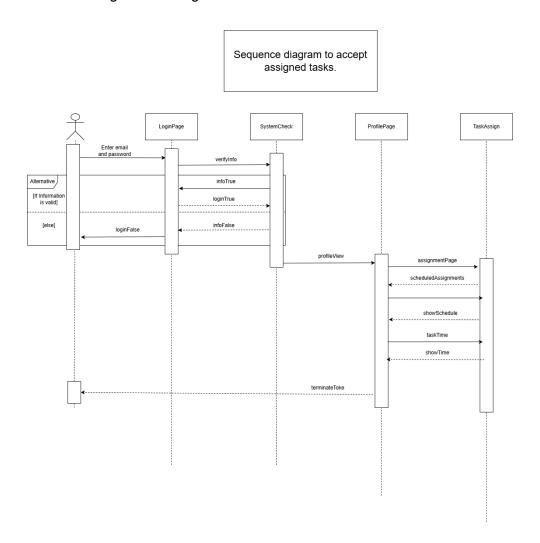
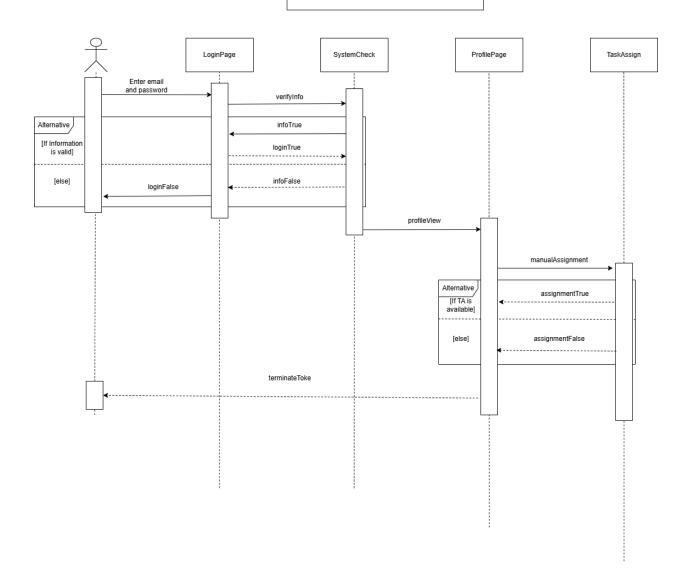
1. Sequence Diagram	
2.Activity Diagram	4
3.Class Diagram	7
4.NFR	8
5.MOCKUPS	10
6.STATE DIAGRAMS	24

1. Sequence Diagram

1.1 State Diagram to Assign Task

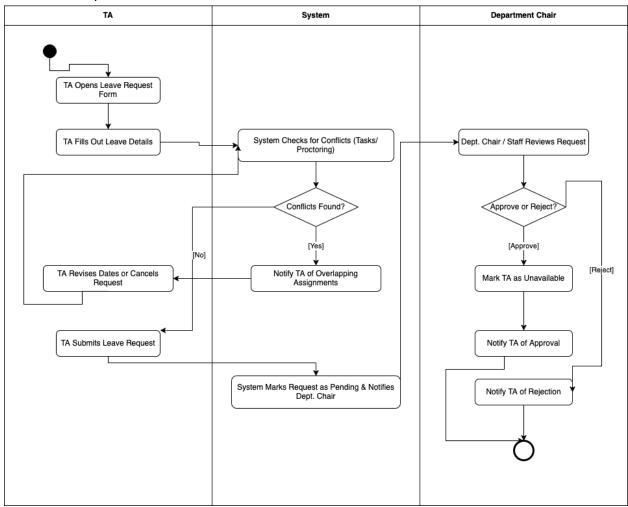


Sequence diagram to assign scheduled tasks.

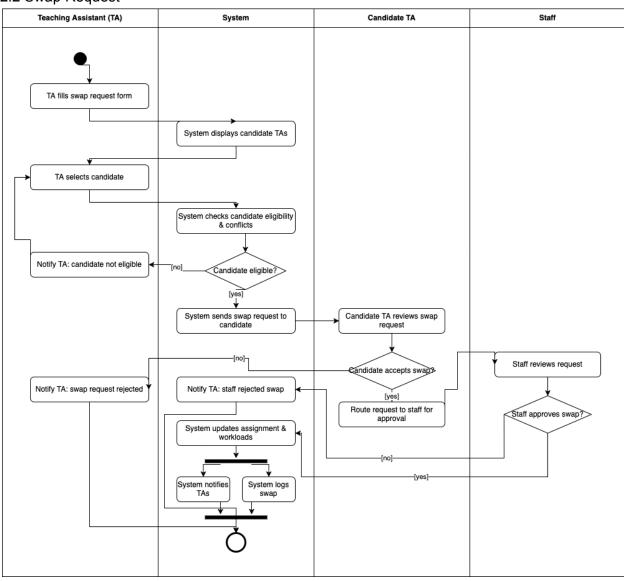


2.Activity Diagram

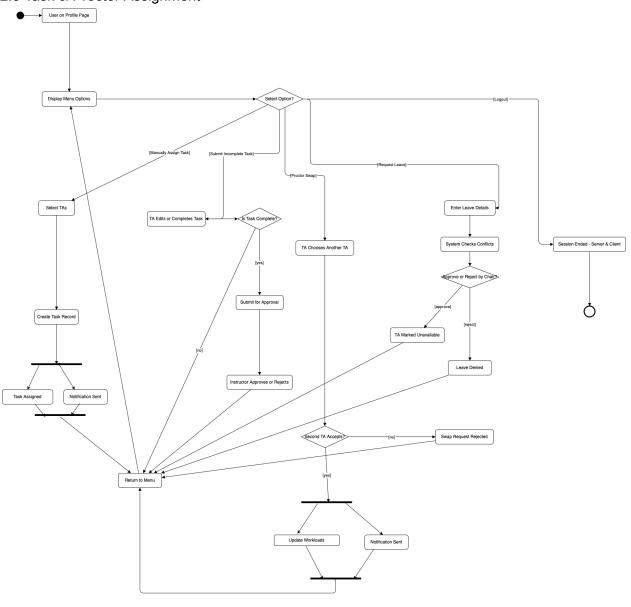
2.1 Leave Request



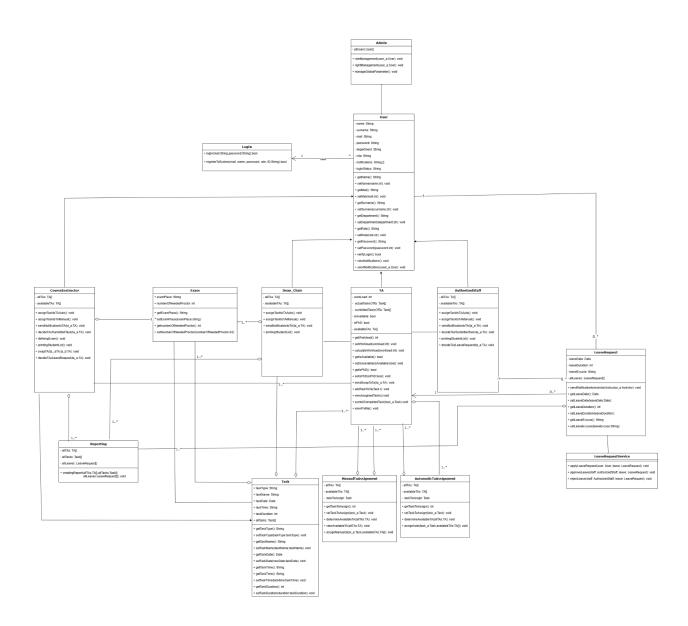
2.2 Swap Request



2.3 Task & Proctor Assignment



3.Class Diagram



4 NFR

Non-Functional Requirements

Maintainability

Consistent Coding Standards:

The project's code will use strict coding standards. For instance, variable names will follow camelCase conventions, and API endpoints will use lower-case with hyphens or underscores consistently. Clear commenting will be used across the codebase.

Modular Architecture:

The codebase will be divided by functionalities such as TA Task Management, Proctoring, Leave Management, Reporting, and System Configuration. This approach will ensures that individual components can be updated or replaced without affecting the entire system.

• Readable Structure:

Code organization will be enforced by placing UI components in a dedicated "components" folder and page layouts in a "pages" folder, with uniform indentation (e.g., 4 spaces per indent). This consistent structure will ensure that the code is easily maintainable by all team members.

Usability

• User-Centric Design:

The TA Management System will feature a clean, intuitive user interface to simplify task submissions, leave requests, and proctoring assignments. Interactive elements (such as buttons and menus) will have subtle animations triggered on hover or click to provide immediate visual feedback.

• Streamlined Workflows:

User interactions such as task assignments, leave requests, and exam definitions are designed to require minimal steps. This simplicity will reduce user errors and enhance overall productivity.

Reliability

• Robust Error Handling:

Critical operations (task deletions, leave approvals, assignment overrides) must handle errors gracefully, logging errors automatically and attempting immediate recovery within 2 seconds.

Data Integrity:

All operations interacting with the MySQL database will ensure data consistency, employing transactions where needed, so that TA workloads, assignments, and reports are accurate and reliable.

Resilience

Even in the event of unexpected errors or server issues, the system will provide informative error messages and redirect users to safe fallback pages to prevent data loss or system crashes.

Performance

Response Time:

The application is expected to have a response time of under 5 seconds for most user interactions, ensuring a smooth experience even during complex operations like automatic proctor assignment and real-time report generation.

Scalability:

The system must scale efficiently to support at least 200 concurrent users (based on department size) with the ability to handle up to 300 users as the department grows over the next two academic years.

Security

Data Encryption: All sensitive data (personal information, assignment data, reports) must be encrypted.

- Access Control: Implement Role-Based Access clearly distinguishing permissions for TA, Staff, Dean, and Admin roles.
- Auditability: All critical actions must be logged comprehensively with timestamps and user identifiers.

Portability and Compatibility

- Cross-Browser Compatibility: The system must support current versions of Chrome, Firefox, Safari, and Edge browsers.
- **Device Compatibility:** The system will be fully responsive, maintaining functionality and usability on desktop platforms.

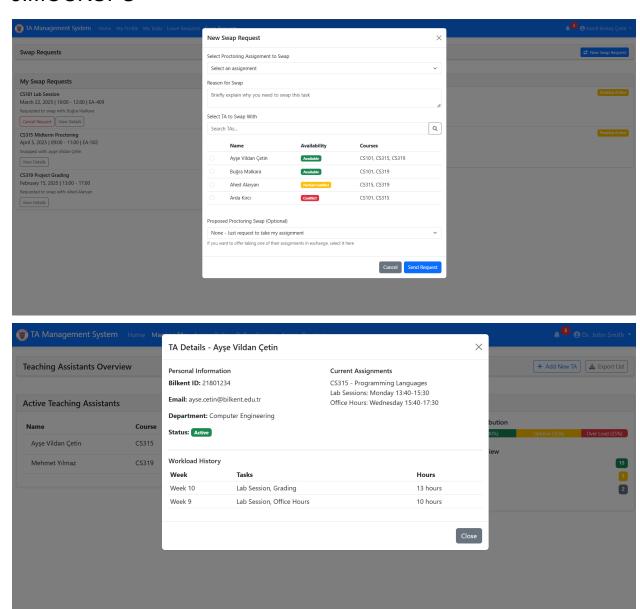
Auditability and Logging

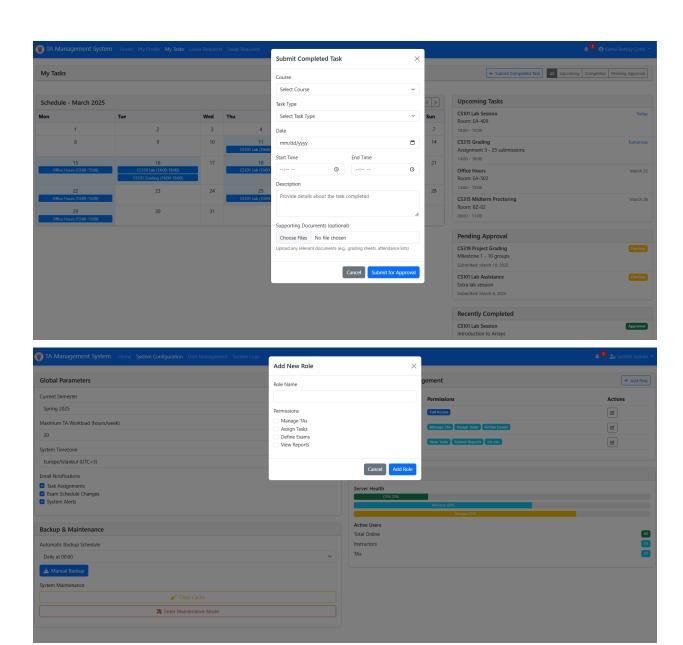
• **Comprehensive Logging:** All significant user actions (assignments, proctor swaps, leave requests, administrative modifications) will be logged in detail.

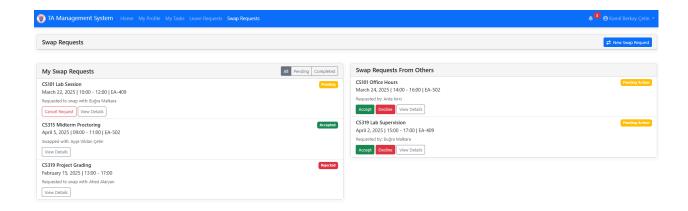
Compliance Requirements

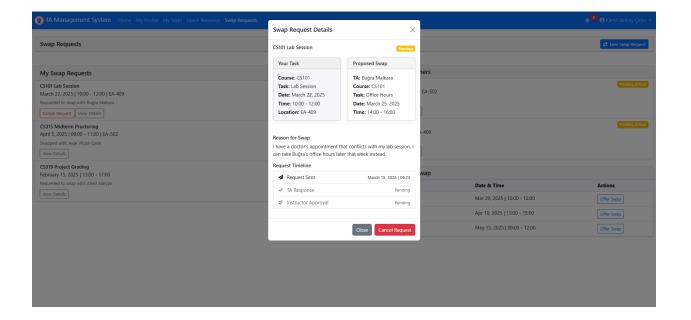
• **Data Protection Compliance:** The system must adhere strictly to Turkish data protection regulations (KVKK) and international standards (GDPR) regarding the handling of personal data.

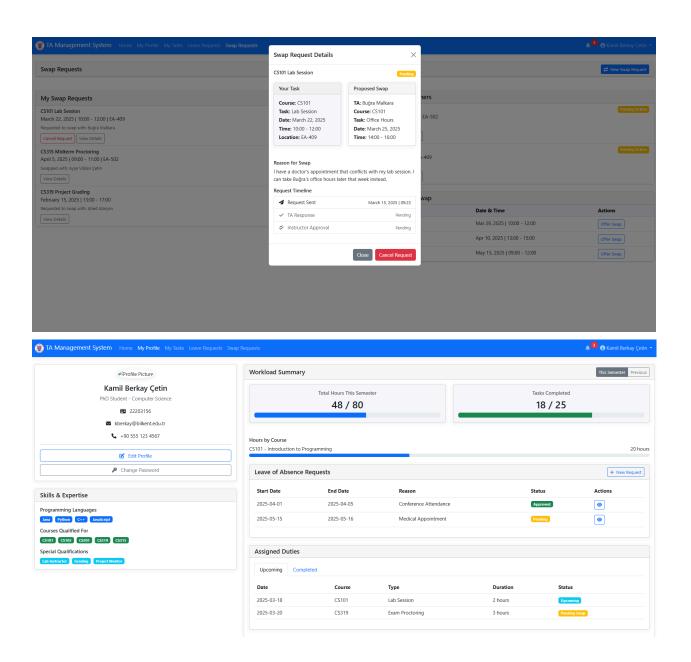
5.MOCKUPS

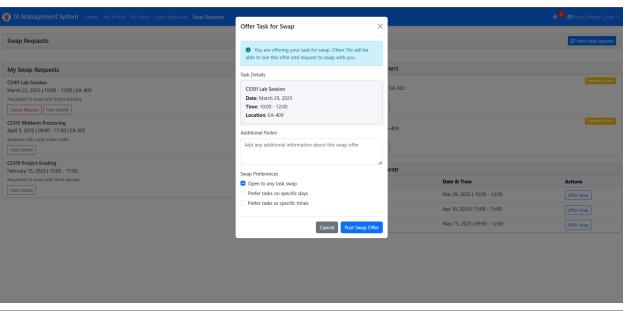


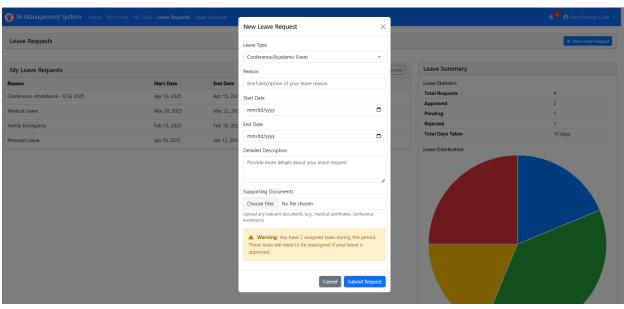


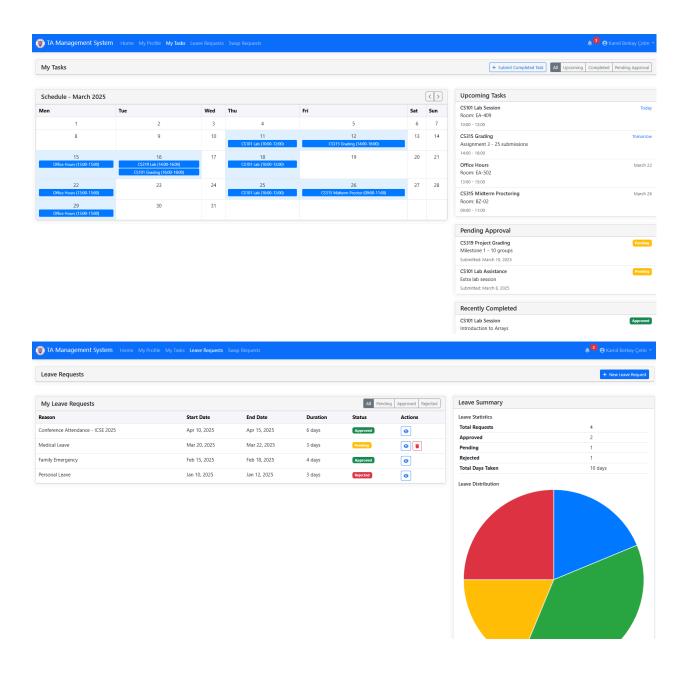


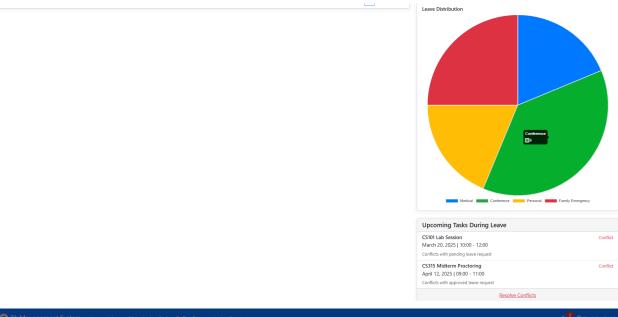


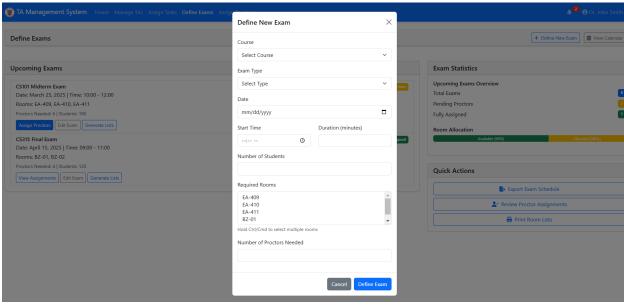


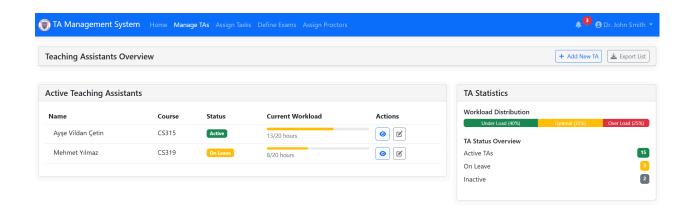


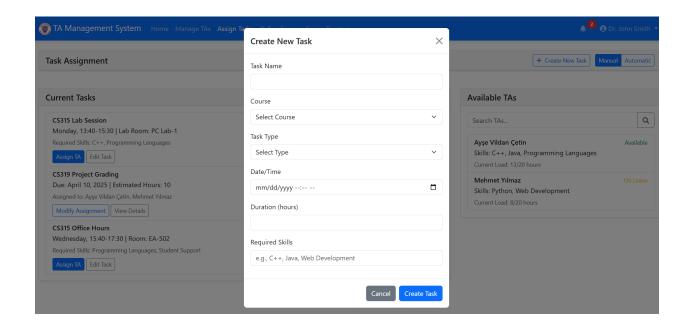


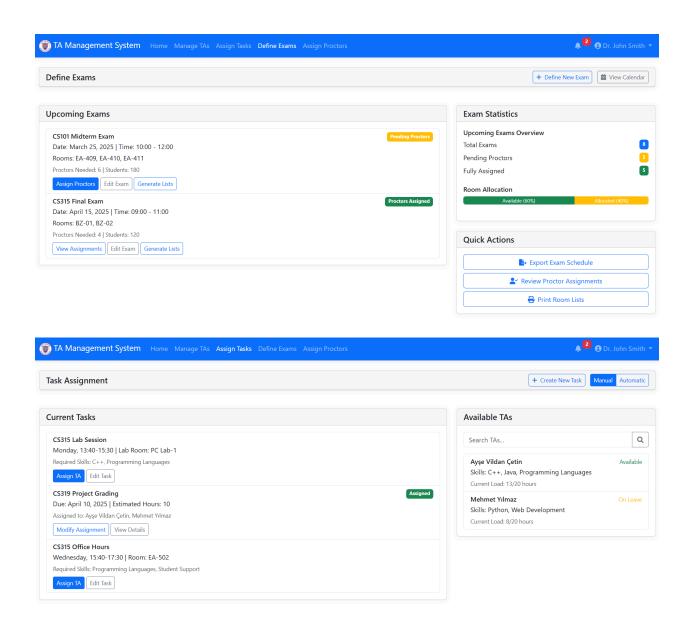


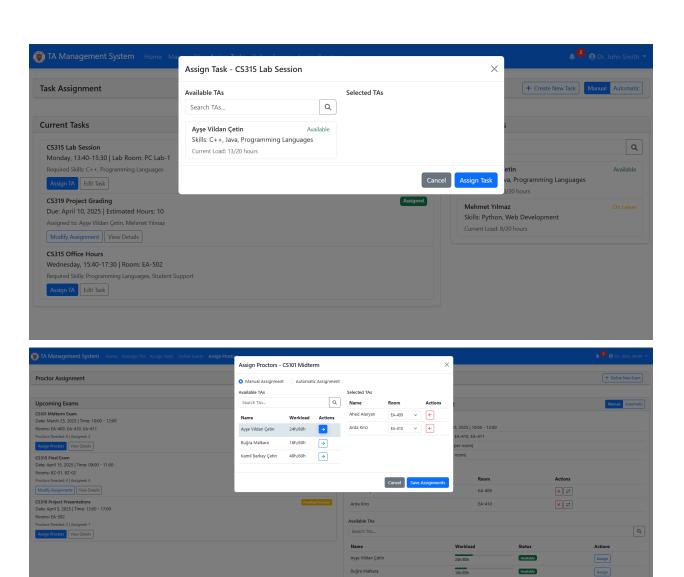


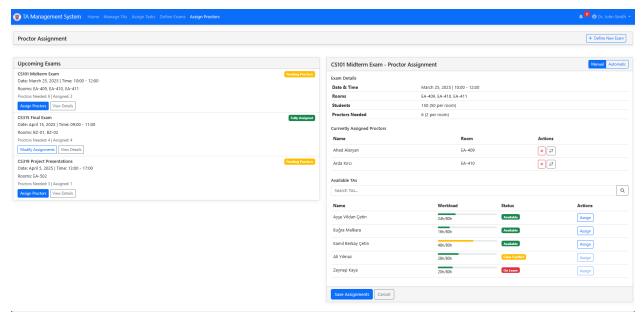


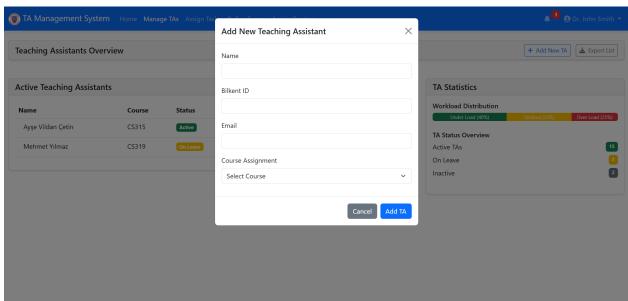


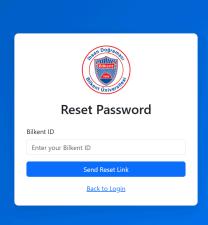




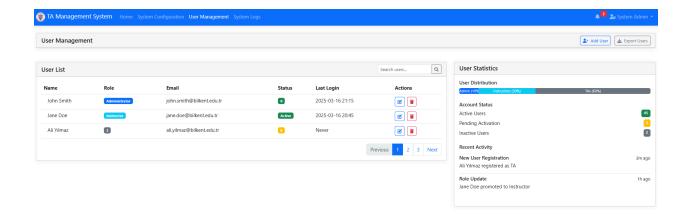


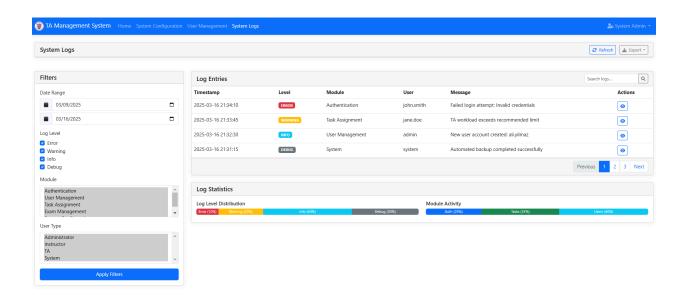


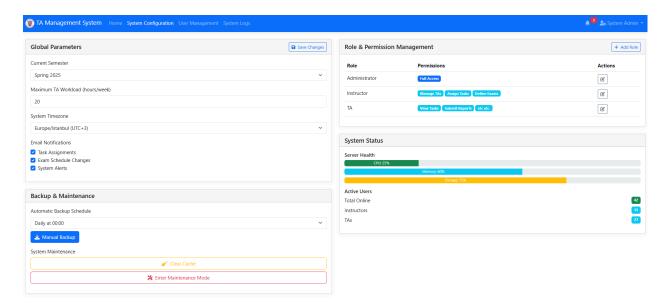


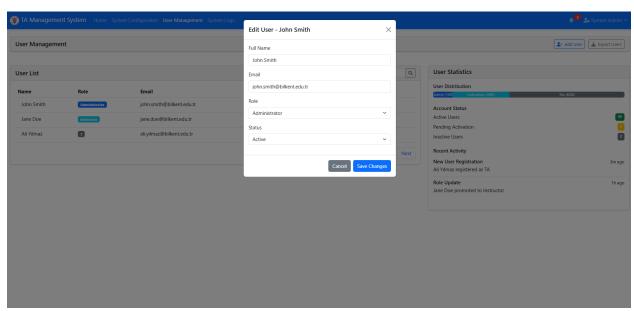






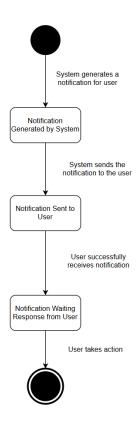




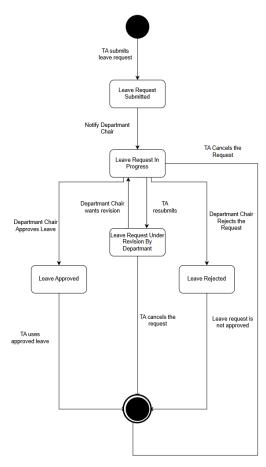


6.STATE DIAGRAMS

6.1 State Diagram for Notification System



State Diagram for Leave Request



State Diagram for Swap Request

