Software Design Document

For

PROBee

EDA KOCAMAN, DİLA TÜMÜR, YUSUF VURAL

ÇANKAYA UNIVERSITY | SENG272

Table of Contents

1. Introduction

1.1. Purpose

The Software Design Document (SDD) serves as a detailed blueprint for the development of PROBee, the graduation project management platform for the Çankaya University Software Engineering Department. It provides a comprehensive overview of the system's architecture, component interactions, database design, user interface layout, security measures, performance considerations, testing strategy, deployment plan, and maintenance procedures. By documenting these aspects, the SDD ensures alignment with the project requirements outlined in the Software Requirements Specification (SRS) and facilitates effective communication, informed decision-making, and risk mitigation throughout the software development lifecycle. Moreover, the SDD serves as a valuable reference for future enhancements, modifications, and upgrades to PROBee, fostering its long-term sustainability, adaptability, and usability within the academic environment.

1.2. Scope

Outlines the detailed design and architecture of PROBee, the graduation project management platform for the Çankaya University Software Engineering Department. It covers the specific functionalities, technical components, and system interactions required to build the features described in the Software Requirements Specification (SRS). This includes designing the user interface, database structure, security measures, integration with external systems, and performance optimization. The SDD sets the boundaries for the design process and serves as a guide for future updates and improvements to PROBee.

1.3. Definitions, Acronyms and Abbreviations

PROBee: Graduation Project Development Application used by the Çankaya University Software Engineering Department.

MVP: Minimum Viable Product - The basic version of the PROBee application with essential features.

SENG: Software Engineering Department at Çankaya University.

SRS: Software Requirement Specification

API: Application Programming Interface

1.4. References

- IEEE SRS template standards.
- Çankaya University Software Engineering Department. (2023-2024). SENG 491-492 Graduation Project I-II Information Meeting [Course Material].
- Graduation Project Development Application (PROBee Documentation).

1.5. Overview

Provides a detailed blueprint of the architecture and design of PROBee, a web application developed for managing graduation projects within the Çankaya University Software Engineering Department. This document outlines the system's structure, including its main components, their interactions, and the technologies employed. It serves as a guide for developers, ensuring the system is constructed to meet the functional and non-functional requirements outlined in the Software Requirements Specification (SRS). By following this document, stakeholders can gain insight into the design decisions, system behavior, and the integration of components to achieve the goals of PROBee.

2. Overall Architecture

2.1. System Architecture

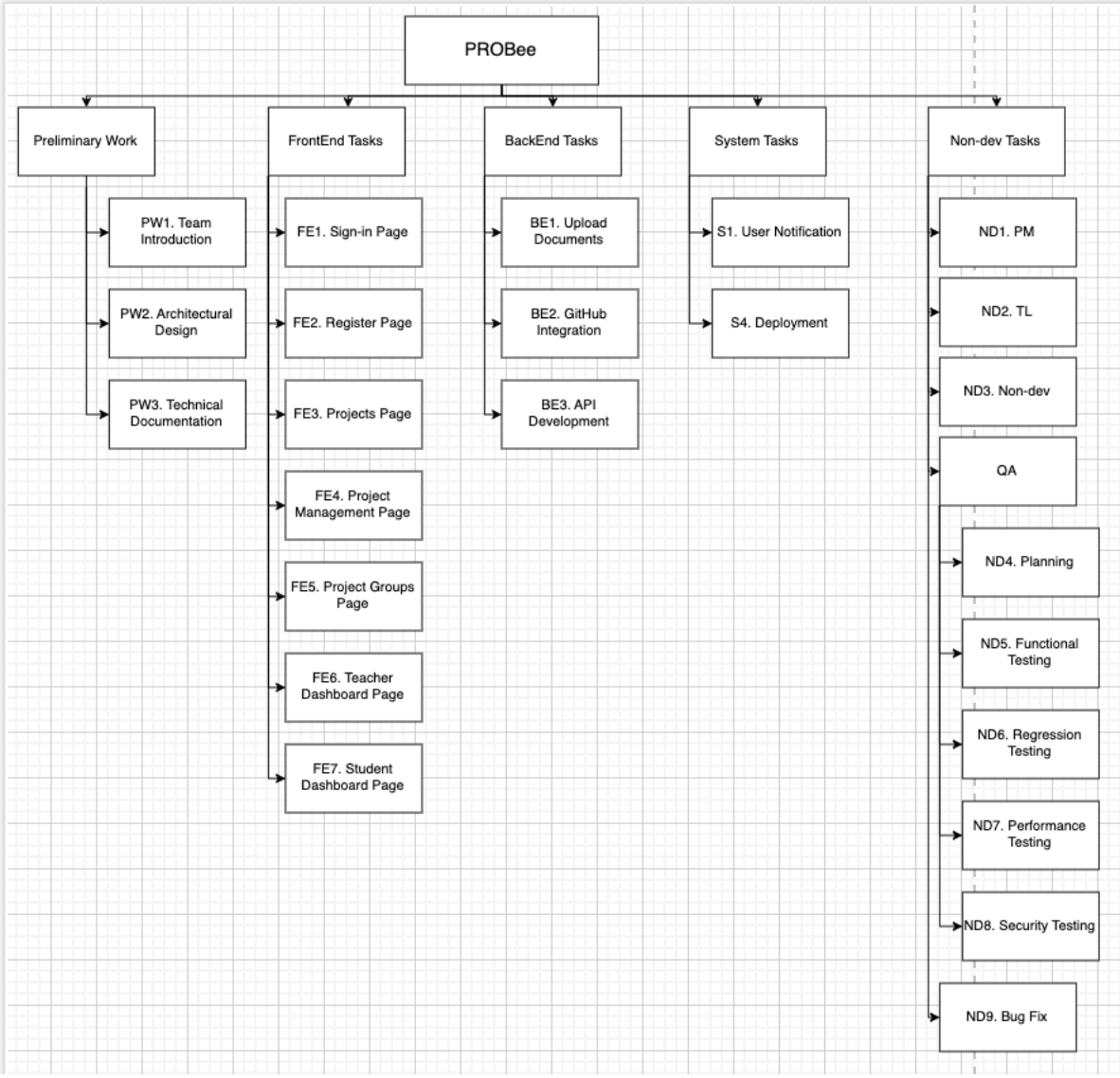

Image1:WBS

2.2. Activity Diagram

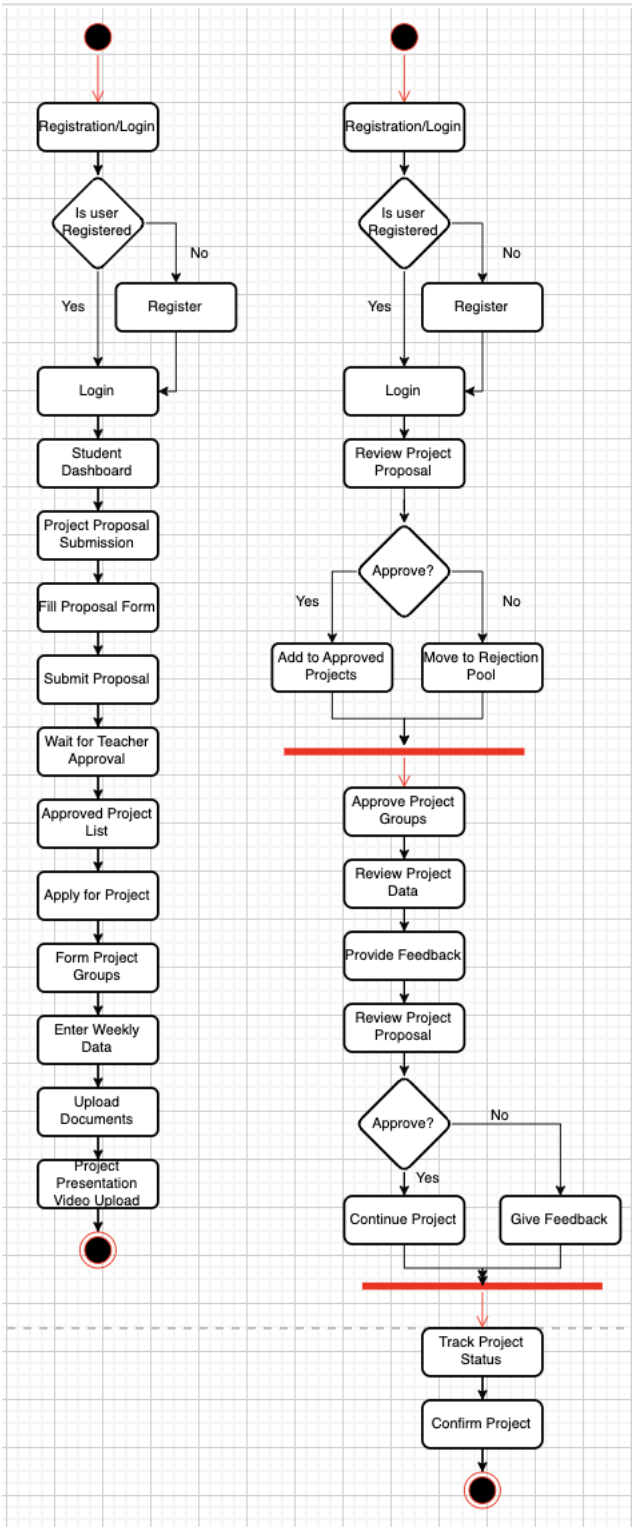

Image2: Activity Diagram

2.3. Use Case Diagram

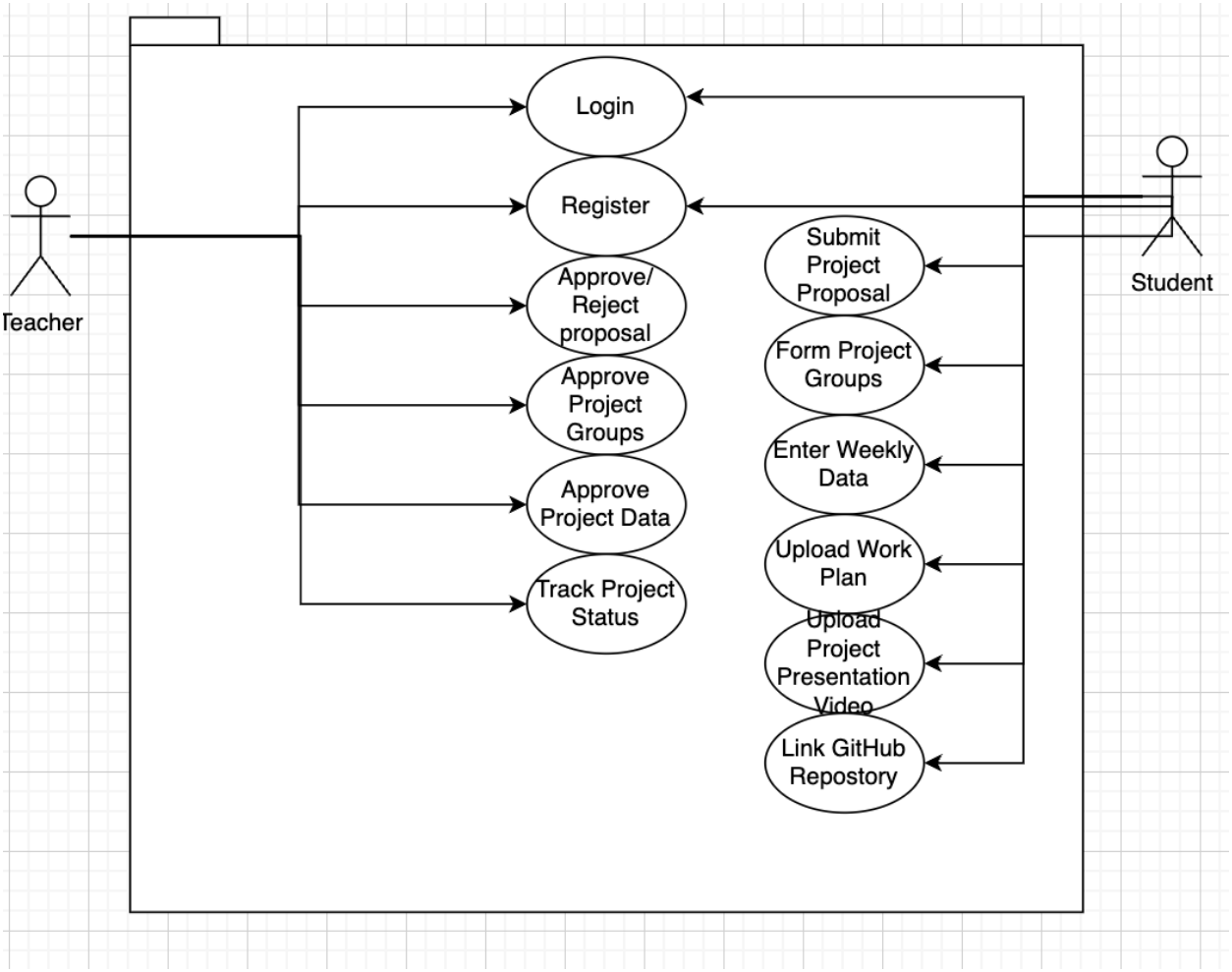

Image3: Use Case Diagram

2.4. Sequence Diagram

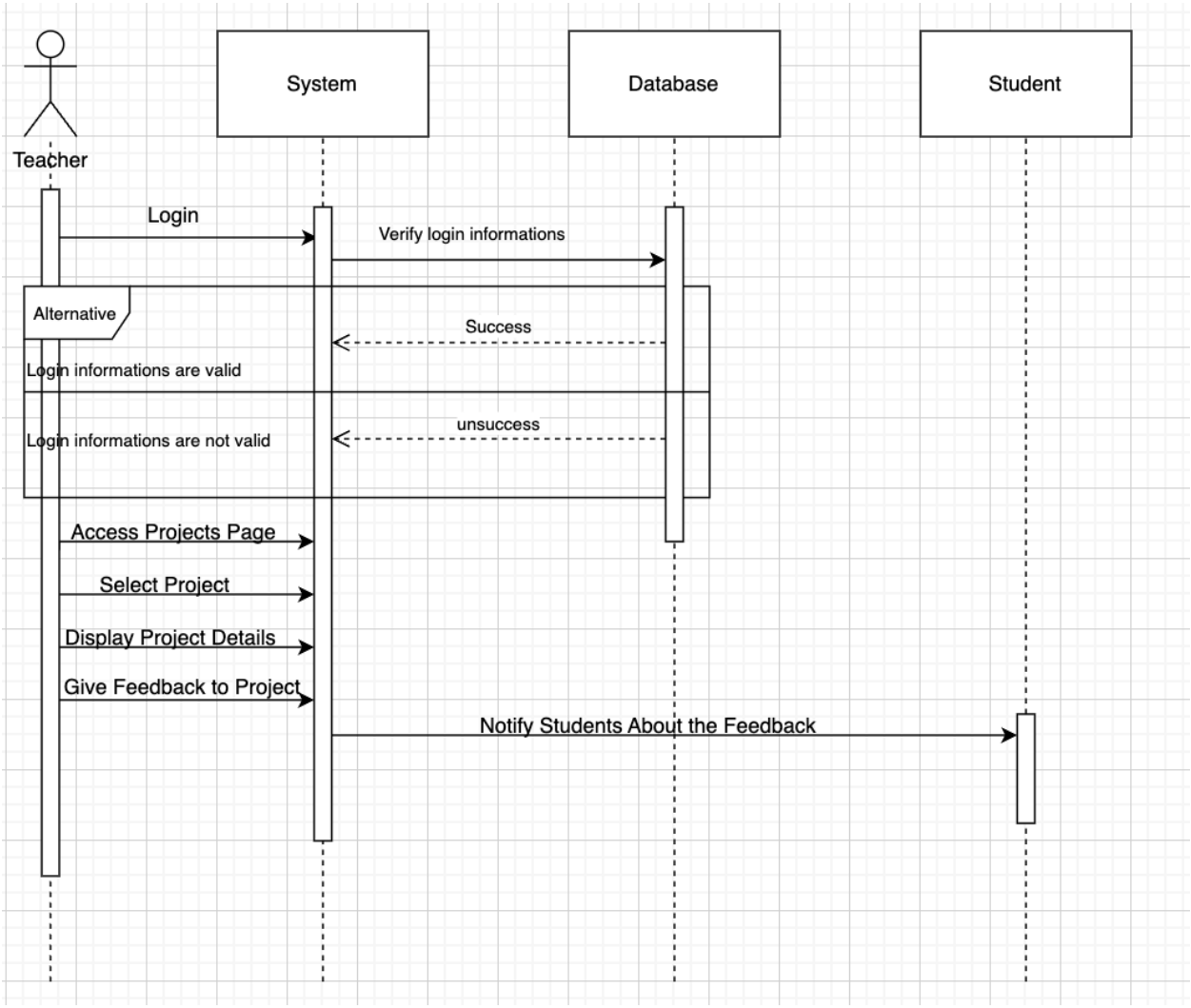

Image4: Sequence Diagram

2.5. High-Level Architecture Overview

The high-level architecture of PROBee is designed to provide a scalable, secure, and efficient platform for managing graduation projects within the Çankaya University Software Engineering Department. At its core, PROBee follows a client-server architecture, where the client-side interacts with the server-side components to access and manipulate project-related data. The architecture consists of several layers, each serving a specific purpose and responsible for different aspects of the application's functionality.

2.6. Components and Modules

The architecture of PROBee comprises several components and modules, each fulfilling specific roles and functionalities within the system. These components interact seamlessly to facilitate the management of graduation projects and ensure the smooth operation of the application. The key components and modules include:

User Management Module:

- Responsible for handling user authentication, registration, and profile management.
- Provides role-based access control (RBAC) to differentiate between administrators, students, advisors, and guests.
- Ensures the security and integrity of user accounts and credentials.

Project Proposal Management Module:

- Allows students to submit project proposals, including project details, objectives, and proposed methodologies.
- Facilitates the review and approval process by administrators and advisors.
- Records the status and history of project proposals, including approval decisions and feedback.

Team Formation Module:

- Enables students to form project teams and apply for projects collectively.
- Manages the assignment of project teams to approved proposals.
- Supports communication and collaboration among team members and advisors.

Project Monitoring Module:

- Facilitates regular monitoring of project progress, including milestones, tasks, and deliverables.
- Provides mechanisms for entering and tracking project monitoring data, such as meeting minutes and progress reports.
- Allows advisors to review and provide feedback on project status and performance.

Documentation and Reporting Module:

- Supports the upload and management of project-related documents, including Software Requirement Specifications (SRS) and Software Design Documents (SDD).
- Enables students to submit progress reports, work plans, and project outputs for evaluation.
- Generates reports and statistics on project activities, achievements, and outcomes.

Integration with External Tools Module:

- Integrates with external services and APIs, such as version control systems like GitHub, to enhance collaboration and versioning capabilities.
- Provides seamless integration with communication tools (e.g., email notifications) and task management platforms to streamline project workflows.
- Ensures interoperability with existing academic and industry tools used by stakeholders.

Security and Compliance Module:

- Implements security measures to protect user data, prevent unauthorized access, and ensure compliance with data protection regulations.
- Enforces encryption, access controls, and secure communication protocols to mitigate security risks.
- Regularly audits and monitors system activity to detect and respond to potential security incidents.

User Interface (UI) Module:

- Provides an intuitive and user-friendly interface for interacting with PROBee's features and functionalities.
- Utilizes responsive design principles to ensure compatibility across different devices and screen sizes.
- Enhances user experience through interactive elements, visualizations, and navigation aids.

2.7. Technology Stack

PROBee, a Graduation Project Development Application, operates through server and database communication over the internet, facilitating seamless interaction between users and the system. Database technologies selected for PROBee prioritize adaptability, initially opting for PostgreSQL due to its ability to accommodate diverse data types. The hardware ecosystem encompasses Android or iOS smartphones with internet connectivity, serving as primary access points for users. Server deployment relies on conventional personal computers with tailored configurations to meet the application's demands. Frontend technologies employed for constructing the user interface and delivering pertinent information leverage cross-platform Object-Oriented Programming (OOP) frameworks, ensuring a cohesive and user-friendly experience across various platforms.

3. Detailed Design

3.1. User Authentication and Authorization

User authentication and authorization in PROBee are crucial components ensuring secure access to the system's features and functionalities. This process involves verifying the identity of users and granting appropriate permissions based on their roles within the system. Below are the key aspects of user authentication and authorization in PROBee:

Authentication:

- Users are required to register and log in using a username and password combination.
- Upon registration, users' credentials are securely stored in the system's database using industry-standard encryption algorithms.
- Authentication mechanisms may include additional layers of security, such as two-factor authentication (2FA) or email verification, to enhance account security.
- Sessions are managed securely to maintain user authentication state throughout their interactions with the application.

Authorization:

- User roles and permissions are defined within the system, including roles such as admin, student, advisor, and guest.
- Each role is associated with specific permissions dictating access to various features and data within the application.
- Authorization checks are performed at each access point to ensure that users only have access to resources and functionalities relevant to their roles.
- Role-based access control (RBAC) mechanisms are implemented to enforce these permissions, allowing administrators to manage roles and permissions dynamically.

Secure Communication:

- All communication between the client and server, including authentication requests and session management, is encrypted using industry-standard protocols such as HTTPS.
- Secure token-based authentication, such as JSON Web Tokens (JWT), may be utilized to transmit authentication credentials securely between the client and server.
- Measures are implemented to prevent common security vulnerabilities such as cross-site scripting (XSS), cross-site request forgery (CSRF), and session hijacking.

3.2. Project Proposal Management

Project proposal management within PROBee is a pivotal process facilitating the submission, review, and approval of project proposals by students and administrators. This functionality streamlines the initial phase of project development, ensuring transparency and efficiency. Key components of project proposal management in PROBee include:

Submission Process:

- Students can submit project proposals through the PROBee platform, providing detailed information about the proposed project, including objectives, methodologies, and expected outcomes.
- The submission process includes uploading relevant documents, such as project outlines, timelines, and resource requirements, to aid in the evaluation process.

Review and Approval:

- Upon submission, project proposals undergo review by administrators or designated faculty members.
- Review criteria may include feasibility, relevance to academic objectives, and alignment with departmental guidelines.
- Administrators have the authority to approve or reject project proposals based on the review outcomes, providing feedback or suggestions for improvement when necessary.
- Approved proposals are marked as such within the system, indicating readiness for the next phase of project development.

Status Tracking:

- Throughout the review process, students can track the status of their project proposals within the PROBee platform.
- Clear indicators inform students of the progress of their submissions, including pending review, under review, approved, or rejected status.

Feedback and Communication:

- PROBee facilitates communication between students and administrators regarding project proposals.
- Students may receive feedback on their submissions, including suggestions for revisions or clarifications, helping them refine their proposals before resubmission if necessary.

Documentation and Archiving:

- Approved project proposals are archived within the system for future reference and documentation purposes.
- Archived proposals serve as a repository of past projects, providing valuable insights and examples for future students undertaking similar endeavors.

3.3. Project Calls

Project calls in PROBee facilitate the selection and allocation of projects to students and advisors, streamlining the process of project assignment and team formation. This functionality ensures efficient coordination and alignment between project proposals and student interests. Key features of project calls in PROBee include:

Inclusion of Approved Projects:

- Approved projects are included in project calls, making them visible to students and advisors for selection.
- Project calls provide a comprehensive list of available projects, including details such as project objectives, requirements, and desired skill sets.

Application Process:

- Students and advisors can apply for projects of interest through the PROBee platform.
- The application process may involve submitting preferences, qualifications, and motivations for selecting specific projects.

Project Acceptance Forms:

- Students who are assigned to projects are required to submit project acceptance forms, indicating their commitment to participating in the project.
- Acceptance forms serve as formal agreements between students and project advisors, ensuring clarity and accountability.

Approval Workflow:

- Project selection forms submitted by students and advisors undergo approval by administrators or designated faculty members.
- Approval ensures that project assignments align with departmental guidelines, student capabilities, and project requirements.

Team Formation:

- Upon approval of project selection forms, students are organized into project teams based on their assigned projects.
- Team formation may involve matching students with complementary skills and expertise to enhance collaboration and project success.

3.4. Team Setup

The Team Setup functionality in PROBee allows students to form project teams, facilitating organized collaboration. Team leaders create teams and invite members, who accept invitations through the system. Teams apply for projects by submitting selection forms, which administrators review for approval. The process ensures compliance with departmental guidelines and enables effective project management through structured roles and responsibilities.

3.5. Guest Access

Guest Access allows external users, such as students from other departments, to interact with PROBee by registering as guests. They can view and submit project proposals, apply for projects, and participate in limited project activities. This functionality ensures inclusivity while maintaining system security through restricted access and robust authentication.

3.6. Project Monitoring

Project Monitoring tracks and manages the progress of projects, ensuring they stay on schedule and meet objectives. Teams provide regular updates, and advisors review and provide feedback during scheduled meetings. The system tracks milestones and generates status reports, facilitating effective oversight and timely interventions to support project success.

3.7. Collaboration Tools

Collaboration Tools in PROBee enhance team communication and coordination by providing features like an "I need" section for requesting help, integrated communication tools, and task management systems. These tools streamline workflows, facilitate resource sharing, and ensure effective collaboration among team members and advisors.

3.8. Documentation and Reporting

Documentation and Reporting functionalities enable students to upload project-related documents, such as SRS and SDD, and generate progress reports. The system maintains a repository of documents, allowing easy access and tracking. Automated reporting features provide insights into project activities, ensuring transparency and accountability.

3.9. Project Presentation and Evaluation

Project Presentation and Evaluation features allow students to upload presentation materials and project videos. Advisors can review, evaluate, and provide feedback on presentations. This functionality supports the assessment of project outcomes and facilitates the evaluation process through structured reporting and feedback mechanisms.

3.10. Integration with GitHub

Integration with GitHub enhances version control and collaboration by linking project repositories to the PROBee system. This integration allows teams to manage their code, track changes, and collaborate effectively on software development projects, ensuring a streamlined workflow and comprehensive project management.

3.11. Task Management

Task Management features provide tools for assigning and tracking project tasks, including Gantt charts and Kanban boards. These tools help teams visualize project timelines, monitor task progress, and ensure timely completion of project activities, facilitating efficient project management and coordination.

4. Database Design

4.1. Data Model

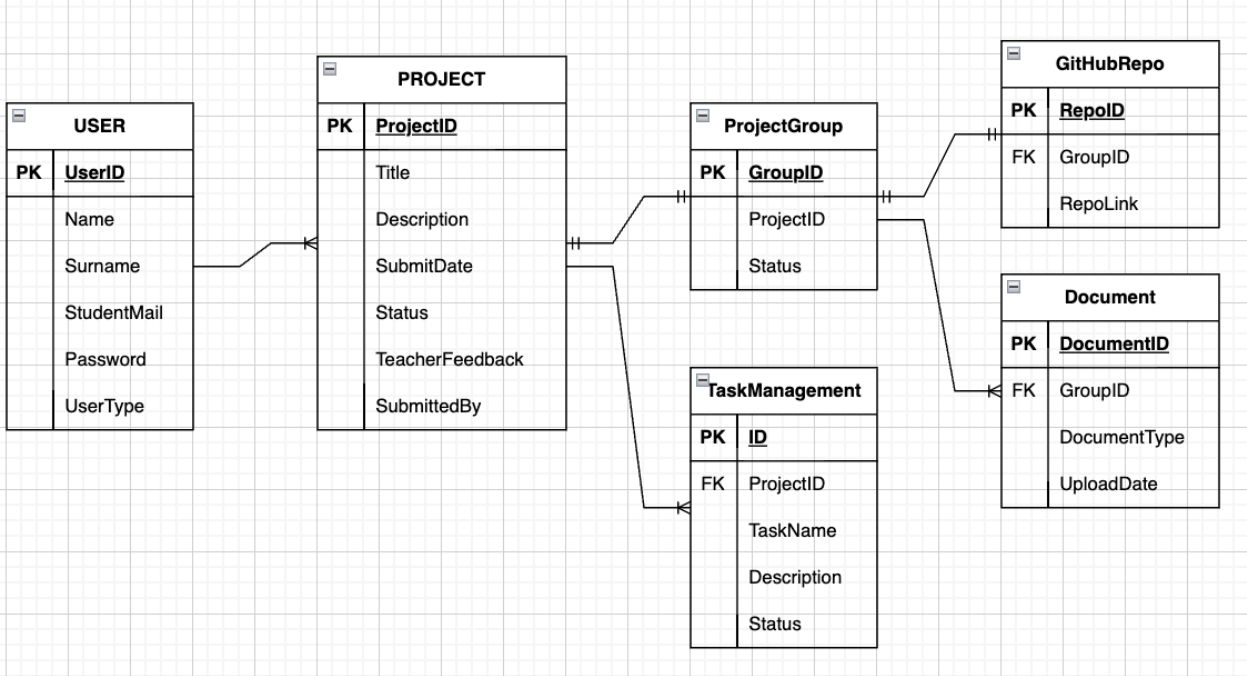

Image5:Data Model

5. User Interface Design

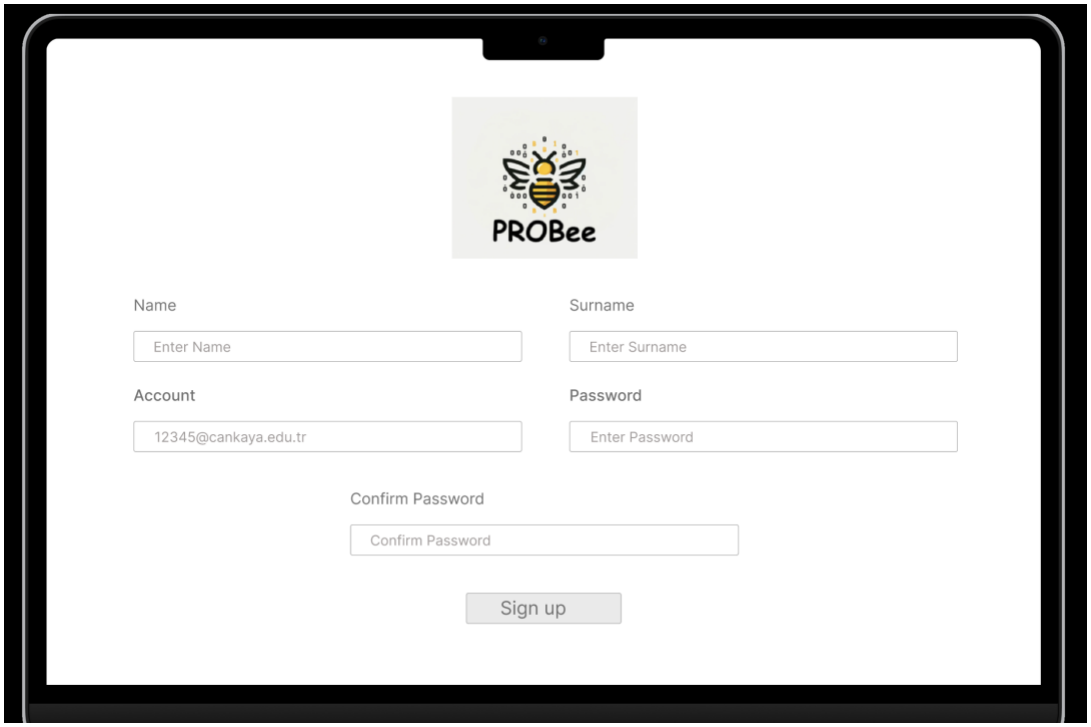

Image7: Register page

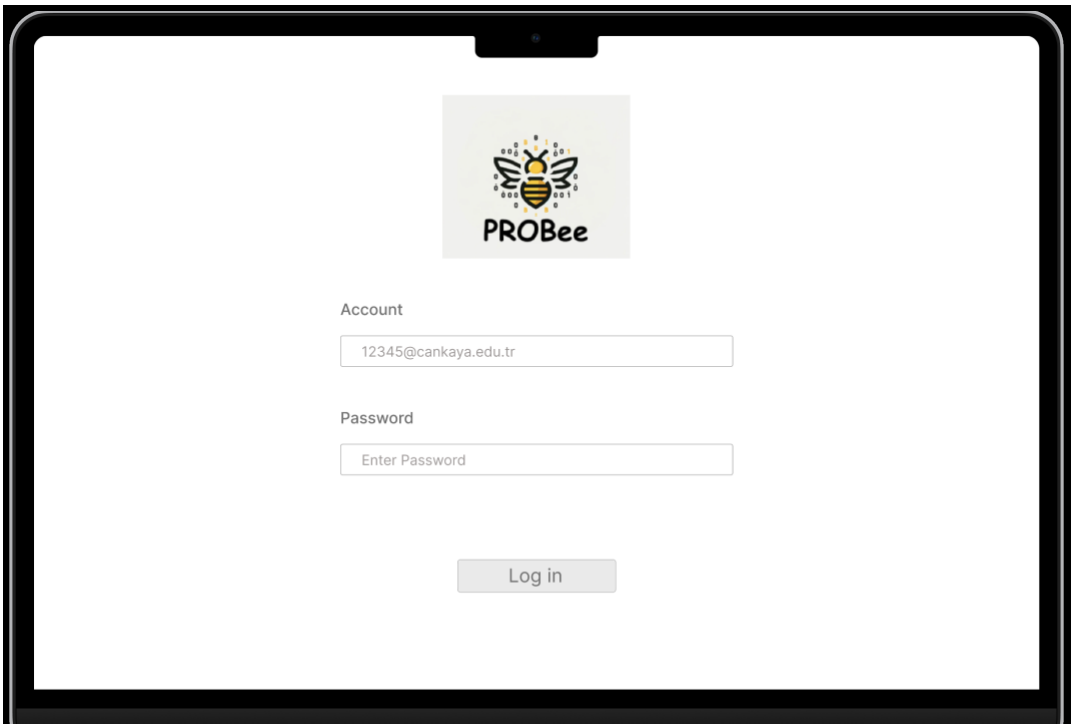

Image8: Login Page

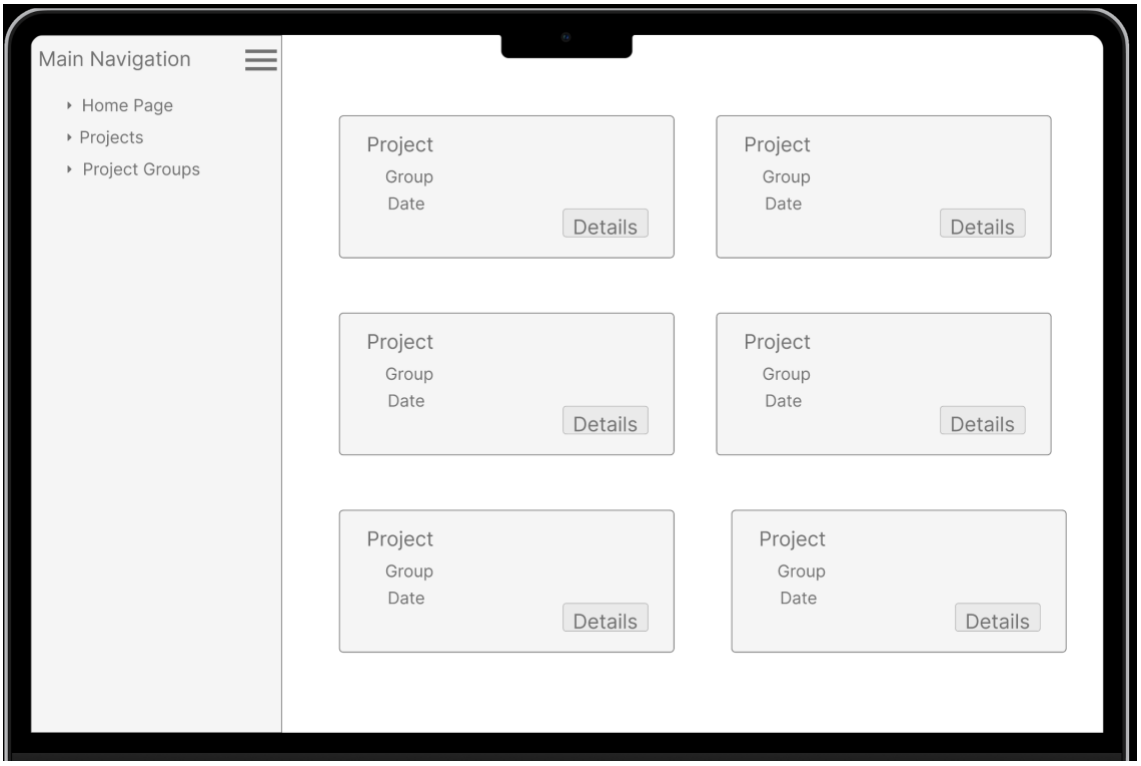

Image9: Projects page

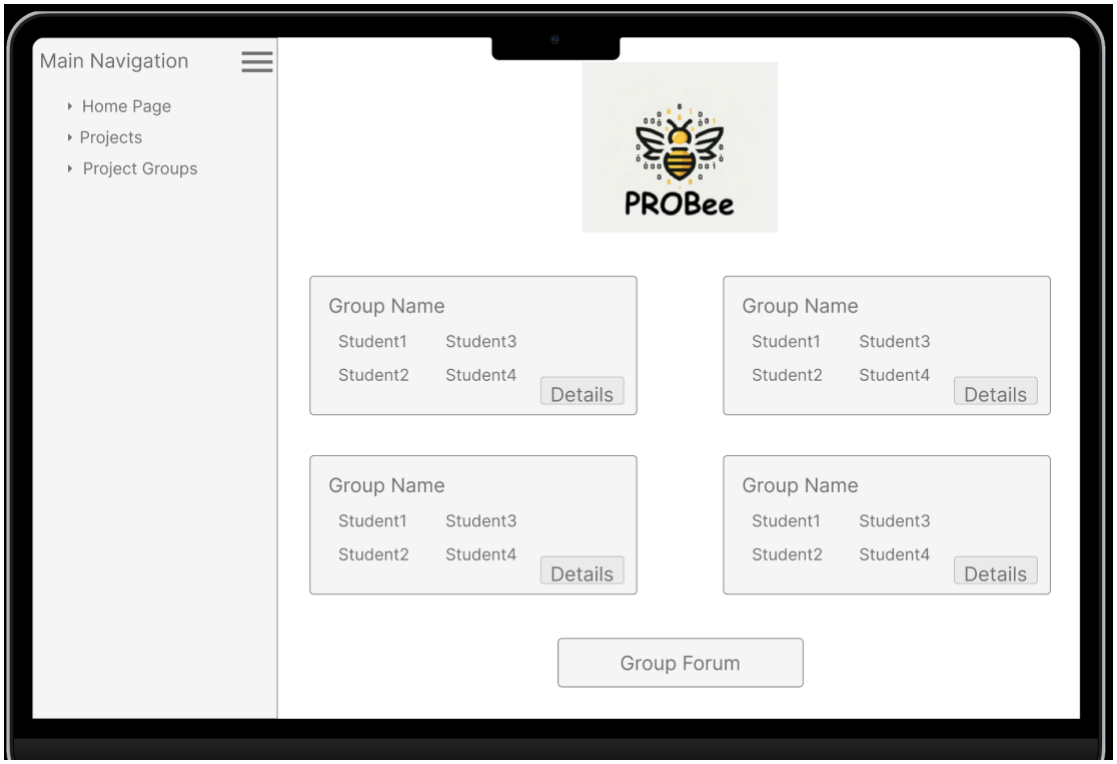

Image10: Groups Page

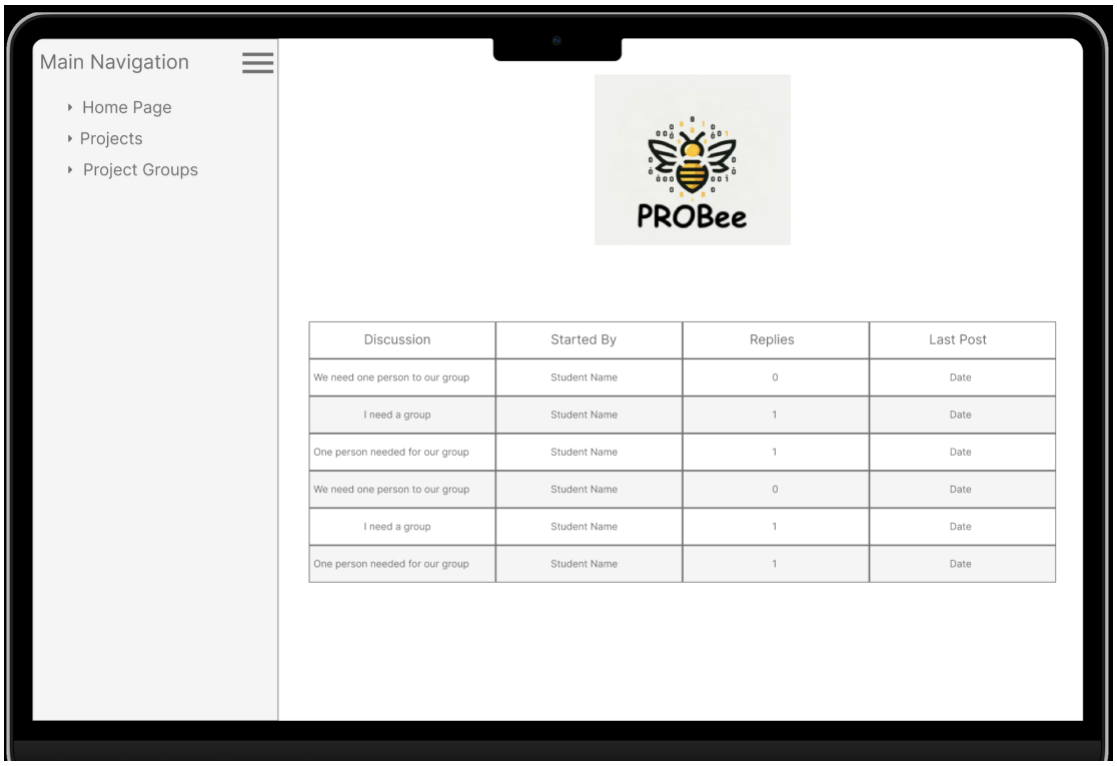

Image11: Student Forum

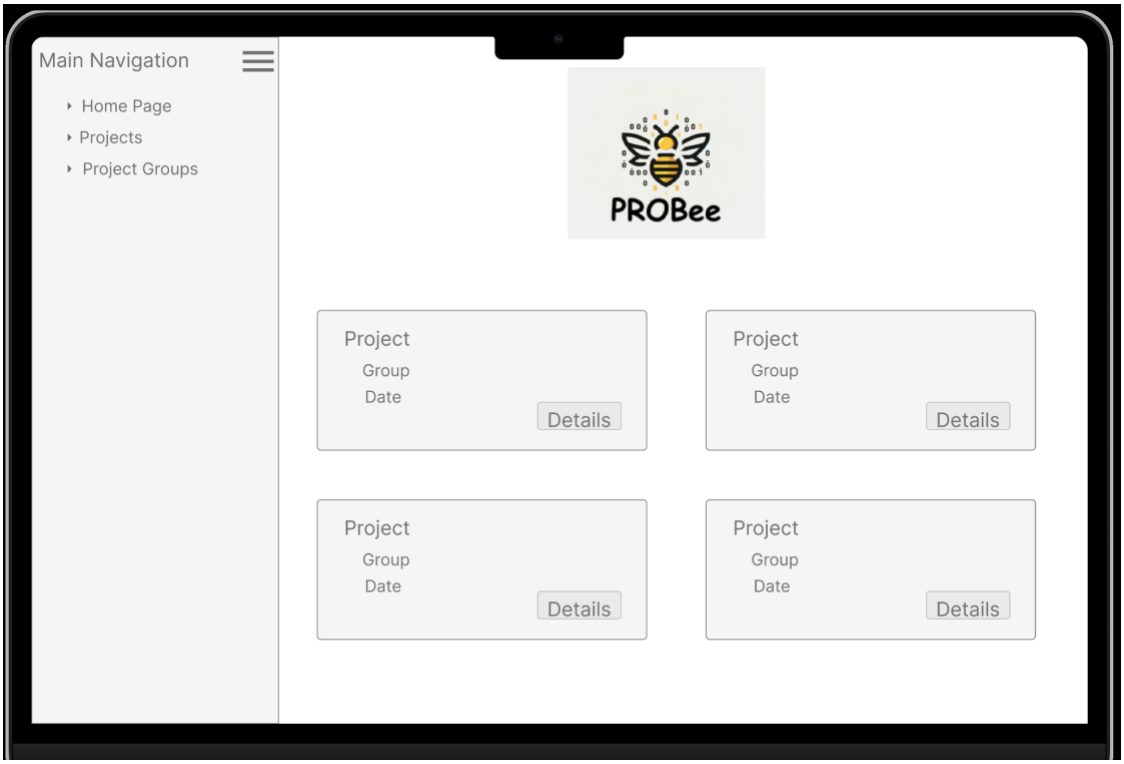

Image12: Teacher Homepage

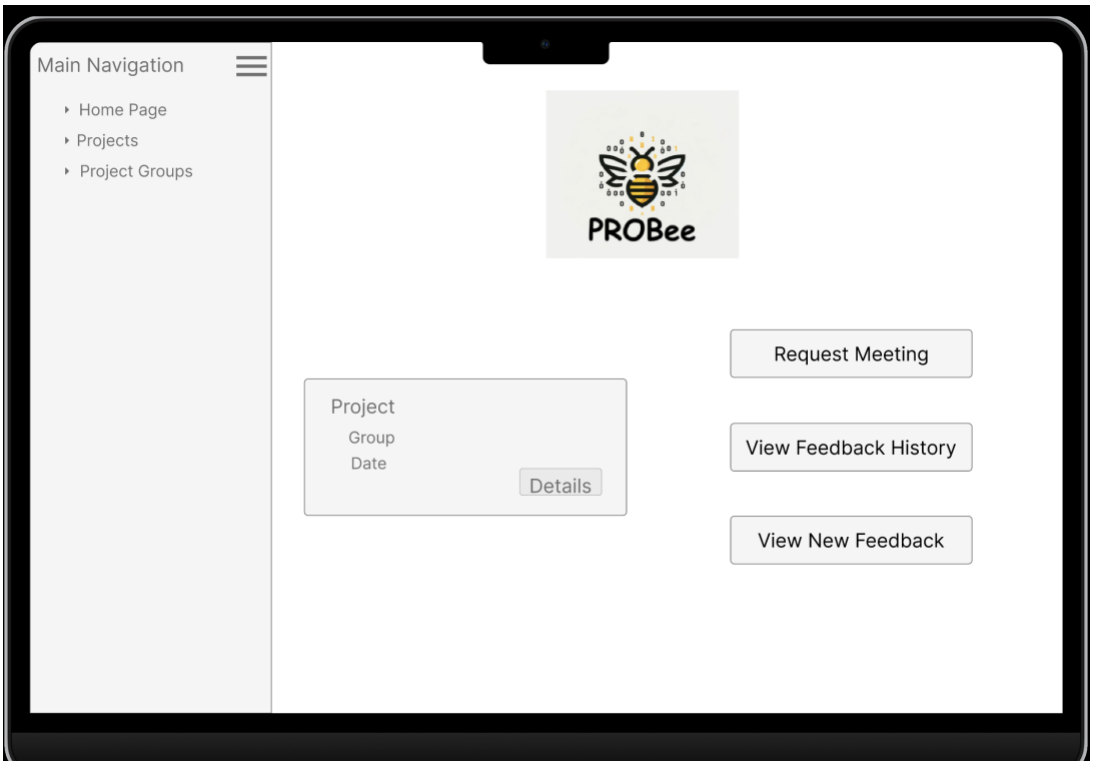

Image13: Student Homepage

6. Security Design

6.1. Authentication Mechanism

- PROBee uses secure login methods, potentially incorporating multi-factor authentication to verify user identities.

6.2. Access Control

- The system enforces role-based access control (RBAC), ensuring users can only access functionalities and data relevant to their roles.

6.3. Data Encryption

- All sensitive data is encrypted both in transit and at rest using industry-standard algorithms to maintain confidentiality and integrity.

6.4. Secure Communication

- Communication between clients and servers is secured using protocols like HTTPS and TLS, ensuring data exchanges are protected from unauthorized access.

7. Performance Design

7.1. Response Time Optimization

- Techniques such as database query optimization and caching are employed to minimize system response times

7.2. Throughput Management

- The system is designed to handle high transaction volumes efficiently, maintaining performance during peak usage periods.

7.3. Concurrency Handling

- PROBee supports multiple concurrent users and transactions through strategies like load balancing and efficient resource management.

7.4. Scalability Strategies

- The system can scale both vertically (enhancing hardware capabilities) and horizontally (adding more servers) to accommodate growing user demands.

8. Testing Strategy

8.1. Unit Testing

- Individual components and functions are tested in isolation to ensure they work as expected.

8.2. Integration Testing

- The interaction between different modules and components is tested to identify issues with how they work together.

8.3. System Testing

- The entire system is tested as a whole to ensure it meets the specified requirements and functions correctly.

8.4. User Acceptance Testing

- The system is tested by end-users to ensure it meets their needs and expectations before final deployment.