|  |  |
| --- | --- |
| Academic plan amendment system  **Course Title: Human computer interaction**  **Supervisor name: Dr. Mohammad Abushariah** | **Zaid Omar Ali Abu Shukidem (2222121).**  **Own Ayman Alfaouri (2214008).**  **Adam Ahmad joblo (0229172).**  **Osama Tayseer Abualwafa (0208935).**  **Khalid Mutaz Zabalawi (0223281).** |

Contents

[Introduction: 2](#_Toc167557534)

[1.1 Need Analysis and Description 2](#_Toc167557535)

[1.2 Project Constraints 2](#_Toc167557536)

[1.3 System Environment 2](#_Toc167557537)

[1.4 Project Software and Hardware Requirements 2](#_Toc167557538)

[1.4.1 Project Software Requirements 2](#_Toc167557539)

[1.4.2 Project Hardware Requirements 2](#_Toc167557540)

[1.5 Project Schedule 3](#_Toc167557541)

[2.0 Project Background and Existing Technologies 3](#_Toc167557542)

[3.0 Software Requirements Document 3](#_Toc167557543)

[3.1 Targeted Users 3](#_Toc167557544)

[3.2 Requirements Gathering and Customer Feedback Techniques 3](#_Toc167557545)

[3.3 Functional Requirements 4](#_Toc167557546)

[3.4 Non-Functional Requirements 5](#_Toc167557547)

[3.5 Usability and User Experience Goals 6](#_Toc167557548)

[3.5.1 Usability Goals: 6](#_Toc167557549)

[3.5.2 User Experience Goals: 7](#_Toc167557550)

[4.0 System Design Document 7](#_Toc167557551)

[4.1 Low Fidelity Prototyping : 7](#_Toc167557552)

[4.2 Medium Fidelity Prototyping 16](#_Toc167557553)

[4.3 Database Design 22](#_Toc167557554)

[22](#_Toc167557555)

[5.0 System Implementation 23](#_Toc167557556)

[5.1 High-Fidelity Prototyping 23](#_Toc167557557)

[6.0 System Testing and Installation 26](#_Toc167557558)

[6.1 Heuristic Evaluation 26](#_Toc167557559)

[6.2 Cooperative Evaluation 27](#_Toc167557560)

[7.0 Conclusion and Future Work 28](#_Toc167557561)

[7.1 Overall Weaknesses 28](#_Toc167557562)

[7.2 Overall Strengths 28](#_Toc167557563)

[7.3 Future Work 29](#_Toc167557564)

[8.0 References 29](#_Toc167557565)

# Introduction:

## 1.1 Need Analysis and Description

The Academic Plan Amendment System is designed to address the cumbersome process of modifying study plans within the University of Jordan. By transitioning from manual paperwork to a web-based platform, this system aims to enhance efficiency, accuracy, and user satisfaction among university administrators, faculty members, and students.

## 1.2 Project Constraints

The project faces constraints such as budget limitations, time constraints for development and implementation, and the need for seamless integration with existing university databases and systems.

## 1.3 System Environment

The system operates within the university's IT infrastructure, leveraging web technologies for accessibility across devices and platforms. It interfaces with databases containing academic program data and user information.

## 1.4 Project Software and Hardware Requirements

Software requirements include a robust web application framework, database management systems, and security protocols. Hardware requirements encompass servers, networking infrastructure, and devices for user access.

### 1.4.1 Project Software Requirements

* + Pencil project Application
  + Visual studio

### Project Hardware Requirements

* + Computer desks + Computers/Laptops
  + Website hosting server

## 1.5 Project Schedule

The project follows a structured schedule encompassing phases such as requirements analysis, design, development, testing, deployment, and ongoing maintenance. Key milestones and deadlines are established to ensure timely delivery.

# 2.0 Project Background and Existing Technologies

The project builds upon existing technologies in web development, database management, and security protocols. It leverages industry best practices to deliver a reliable and scalable solution for academic plan amendments.

# 3.0 Software Requirements Document

## 3.1 Targeted Users

The system caters to university administrators, faculty members, and students, each with distinct roles and permissions within the platform.

## 3.2 Requirements Gathering and Customer Feedback Techniques

Requirements are gathered through techniques such as questionnaires, surveys, literature review, and consultation with academic and industry sources. Customer feedback drives the design and functionality of the system.

## 3.3 Functional Requirements

1. **User Authentication:**
   * Users must authenticate using unique credentials (username/password or other authentication methods).
   * Different user roles (administrators, faculty members, students) with distinct permissions are implemented for access control.
2. **Study Plan Submission and Review Workflows:**
   * Faculty members can submit proposed amendments to academic study plans through a designated interface.
   * Review workflows are established to allow for collaboration among faculty members, enabling them to review and provide feedback on proposed changes.
3. **Collaboration Tools for Faculty Members:**
   * The system provides collaboration tools such as commenting, version tracking, and document sharing for faculty members to collaborate effectively during the review process.
4. **Approval Mechanisms for Department Heads and Deans:**
   * Department heads have the authority to approve or reject proposed amendments to department-specific study plans.
   * Deans oversee the overall academic planning and have the final approval authority for all amendments.
5. **Reporting Functionalities:**
   * The system generates reports on the status of submitted amendments, including pending, approved, and rejected changes.

## 3.4 Non-Functional Requirements

1. **Security Measures:**
   * Implement robust authentication mechanisms (e.g., multi-factor authentication) to secure user access.
   * Utilize data encryption protocols (e.g., SSL/TLS) to protect sensitive information during transmission and storage.
2. **Scalability**:
   * Design the system to handle a growing number of users and data without compromising performance.
   * Implement scalable infrastructure and architecture to support increased user loads and data volume.
3. **Reliability**:
   * Minimize system downtime through regular maintenance, updates, and backups.
   * Conduct thorough testing (e.g., load testing, stress testing) to ensure system reliability under varying conditions.

## 3.5 Usability and User Experience Goals

### 3.5.1 Usability Goals:

1. **Efficiency**:
   * Optimize workflows and user interactions for quick task completion.
   * Streamline processes by reducing unnecessary steps.
2. **Learnability**:
   * Design an intuitive interface for easy learning by new users.
   * Provide clear instructions and guidance throughout the system.
3. **Effectiveness**:
   * Ensure users achieve their goals accurately within the system.
   * Validate inputs and offer meaningful feedback to prevent errors.
4. **Memorability**:
   * Design the interface for easy recall even after periods of inactivity.
   * Maintain consistency in design elements and layout.
5. **Error Prevention**:
   * Implement techniques like input validation and user-friendly error messages.
   * Minimize user errors and confusion in forms and interfaces.

### 3.5.2 User Experience Goals:

1. **Accessibility**:
   * Ensure the system is usable by all, including those with disabilities (e.g., screen readers, keyboard navigation).
   * Adhere to accessibility standards (e.g., WCAG) for an inclusive experience.
2. **Satisfaction**:
   * Strive for a positive overall user experience that encourages regular use.
   * Gather user feedback to continually improve and meet user needs.
3. **Inclusivity**:
   * Design for diverse user backgrounds, languages, and cultural preferences.
   * Foster a sense of belonging and fairness through inclusive design practices.

# System Design Document

## Low Fidelity Prototyping :

A paper with writing on it

Description automatically generated

Figure 1: Log In page

A paper with writing on it

Description automatically generated

Figure 2: Main Dashboard

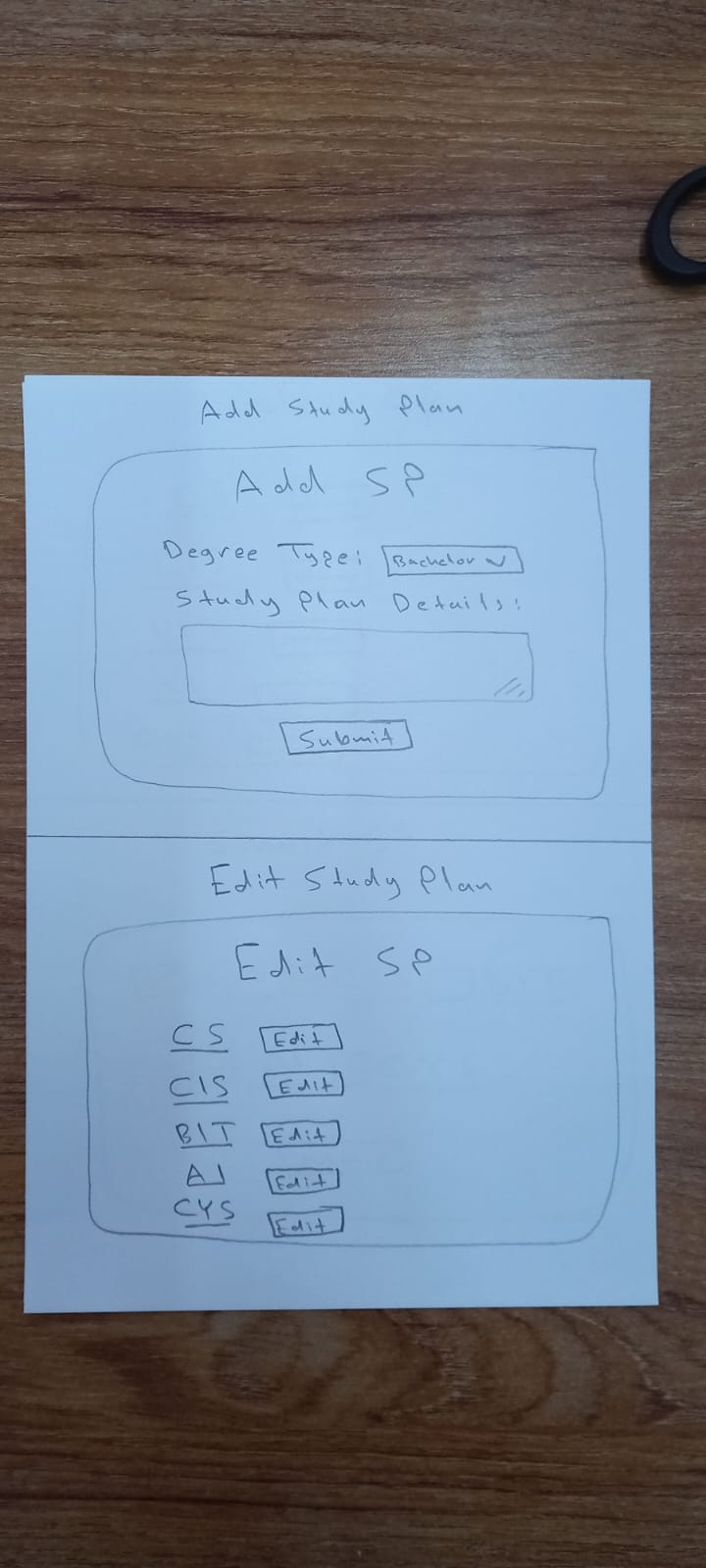


Figure 3: Add Study Plan

A white paper with black text

Description automatically generated

Figure 4: Edit study Plan

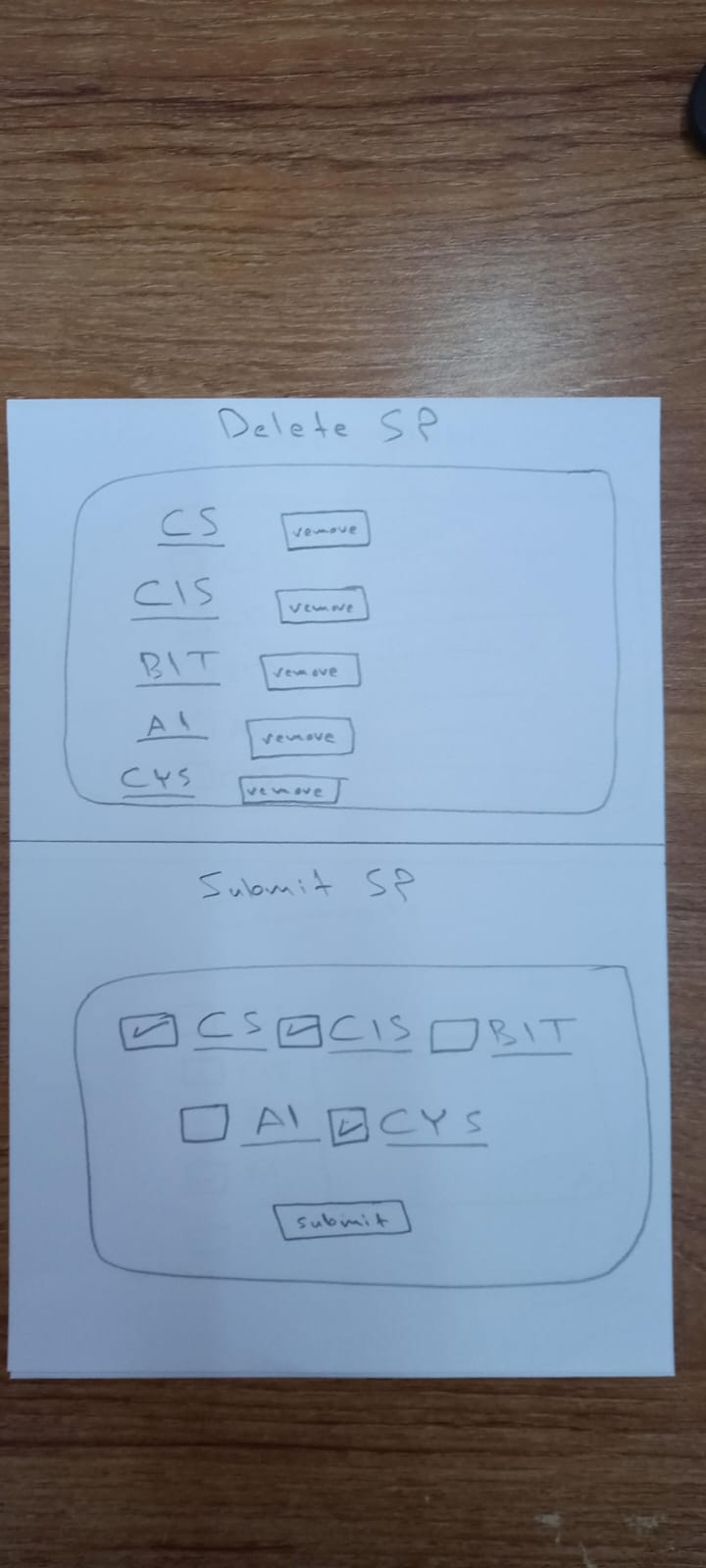


Figure 5: Delete Study Plan

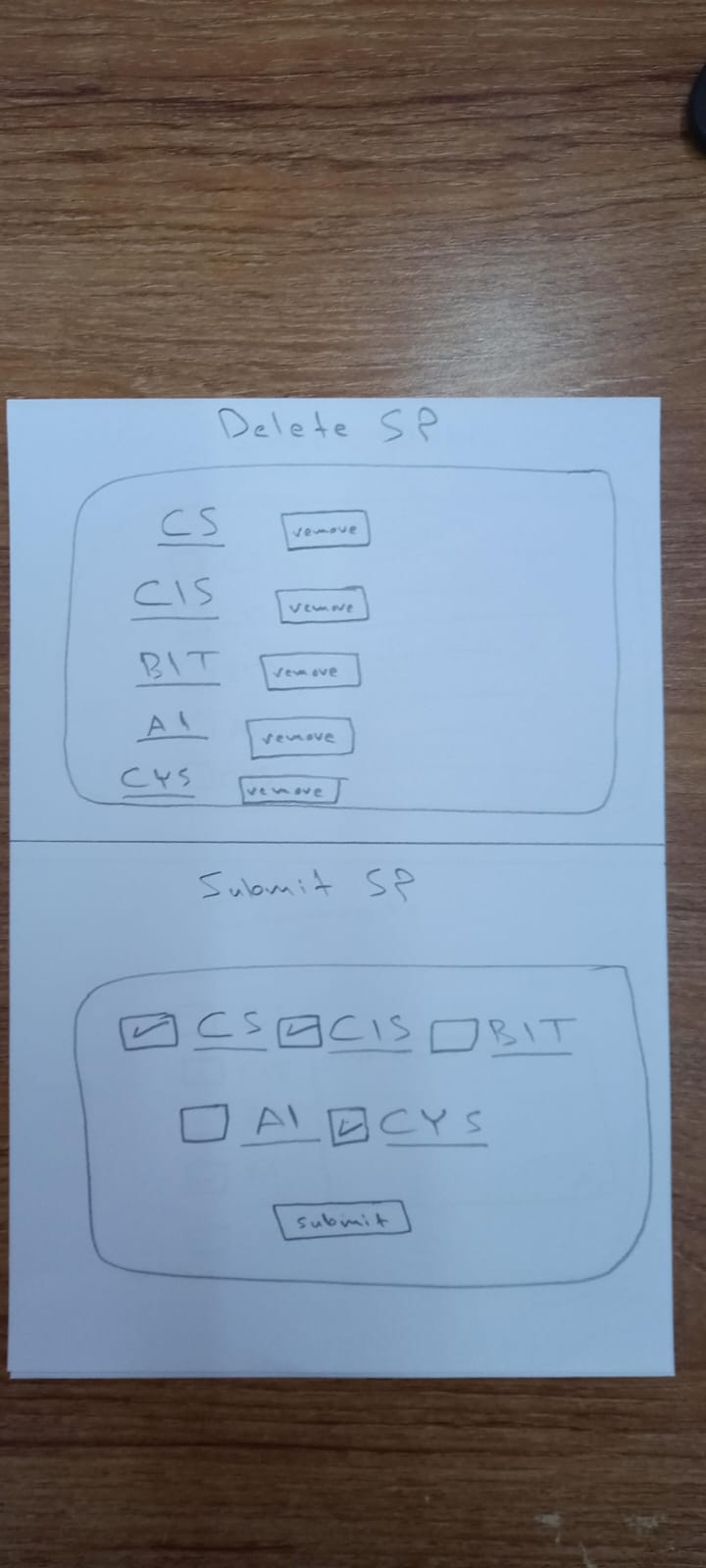


Figure 6: Submit Stady Plan

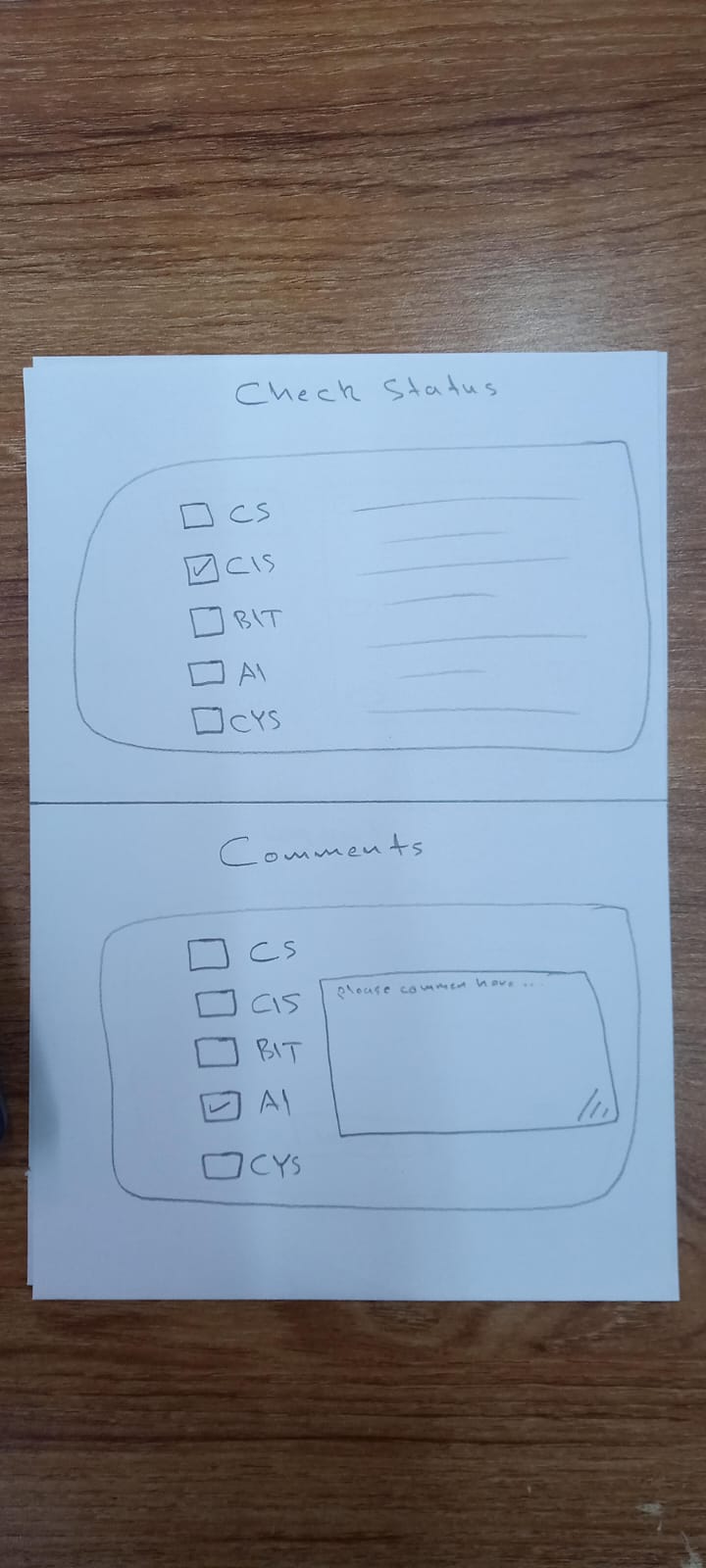


Figure 7: Check Status

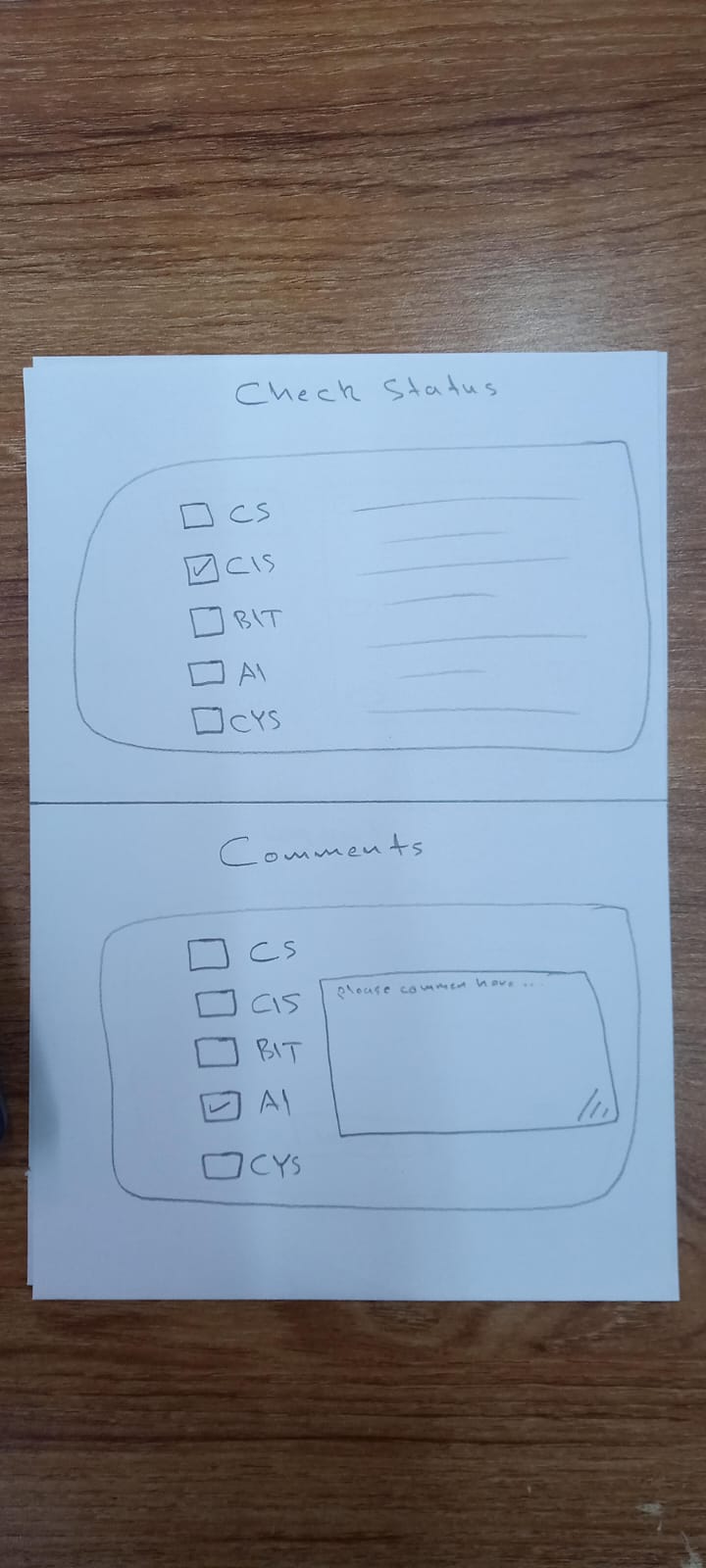


Figure 8: Comment

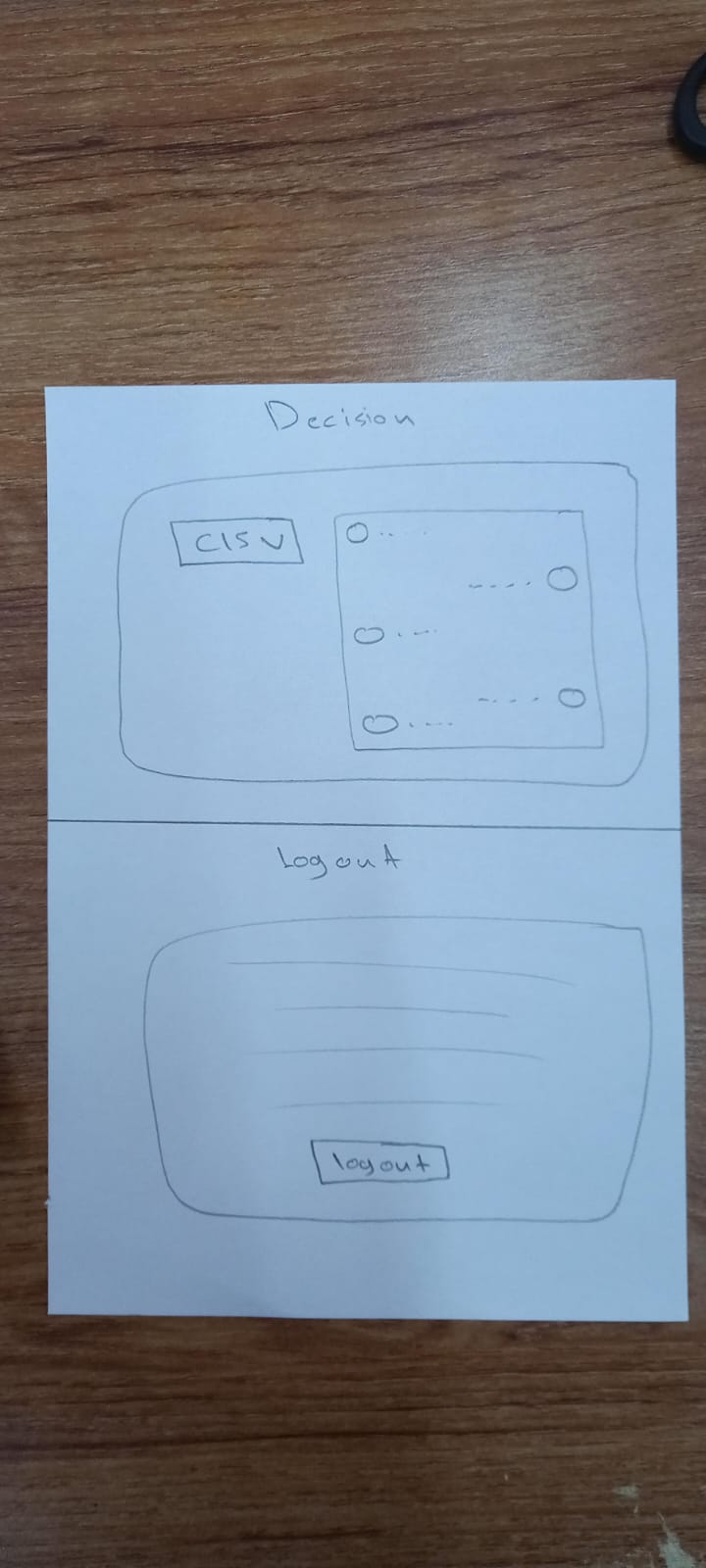


Figure 9: Decision

A paper with a drawing on it

Description automatically generated

Figure 10: Logout

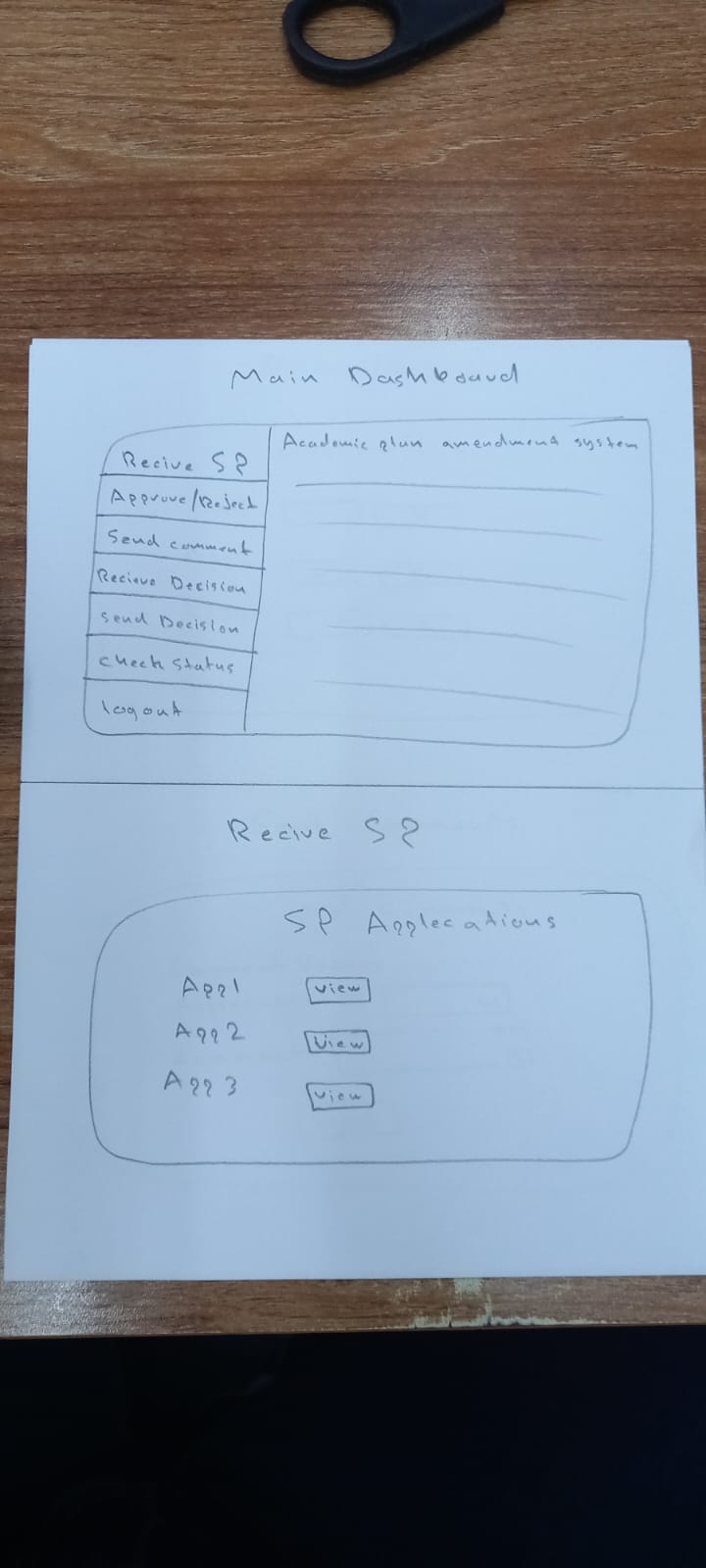


Figure 11: Main DB

A paper with writing on it

Description automatically generated

Figure 12: Receive SP

A paper with writing on it

Description automatically generated

Figure 13: Approve/Reject

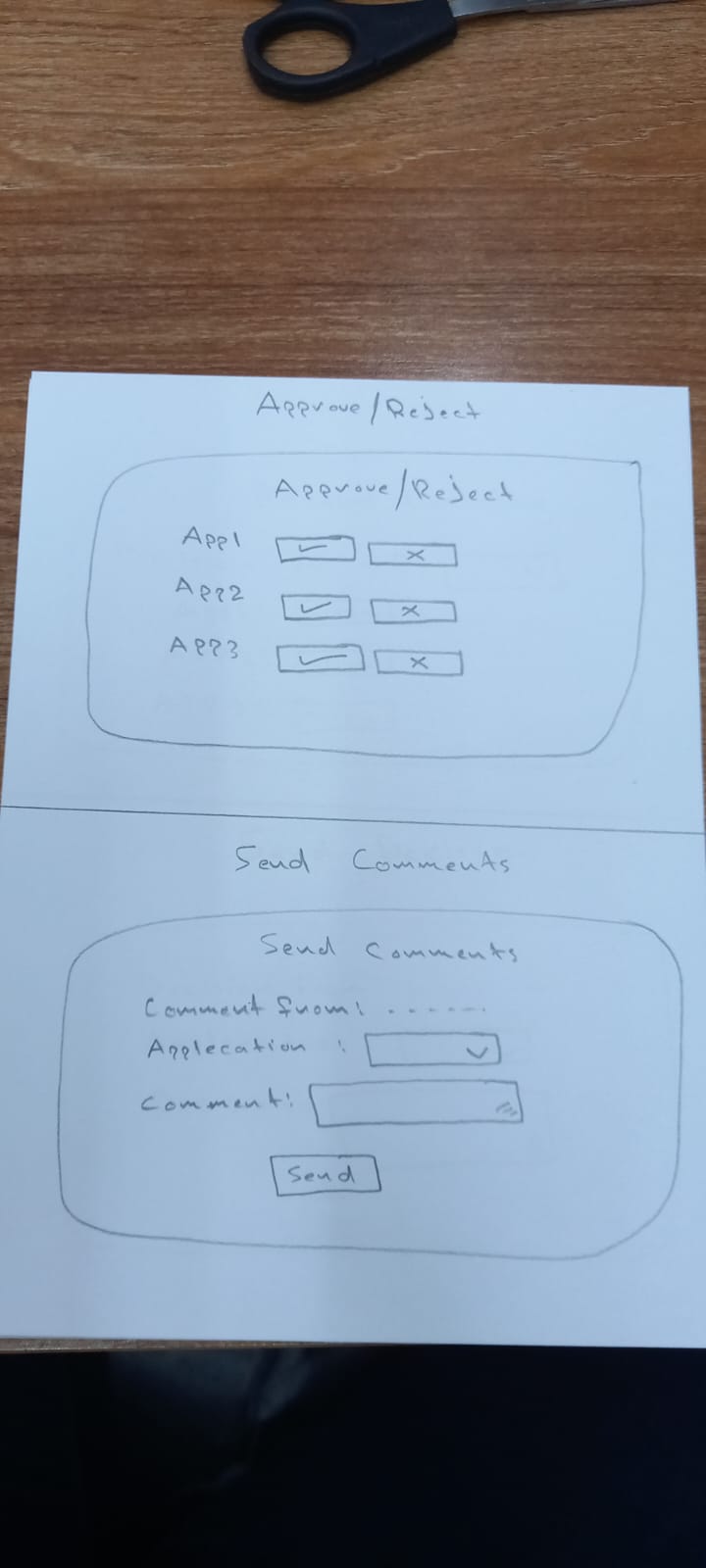


Figure 14:Send Comment

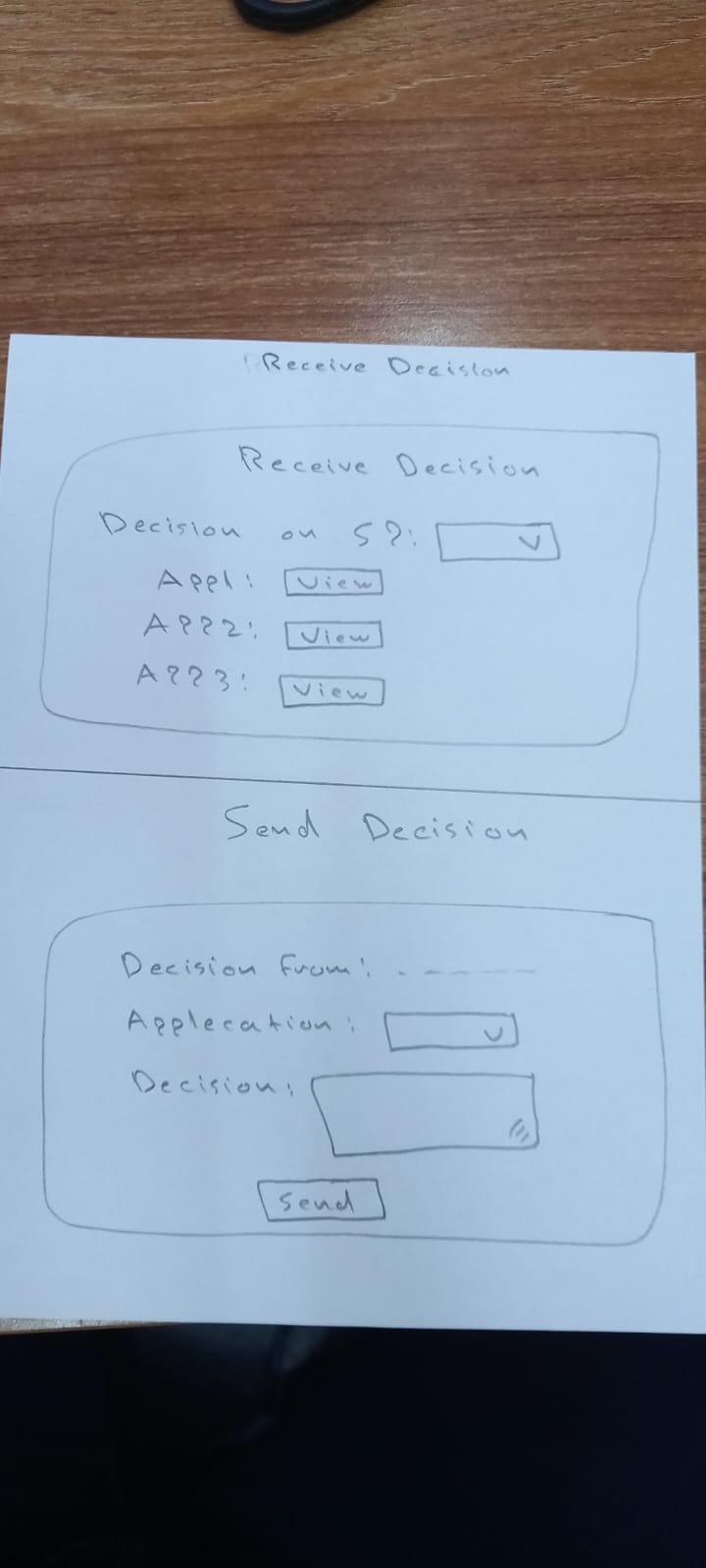


Figure 15: Receive Decision

A paper with writing on it

Description automatically generated

Figure 16: Send Decision

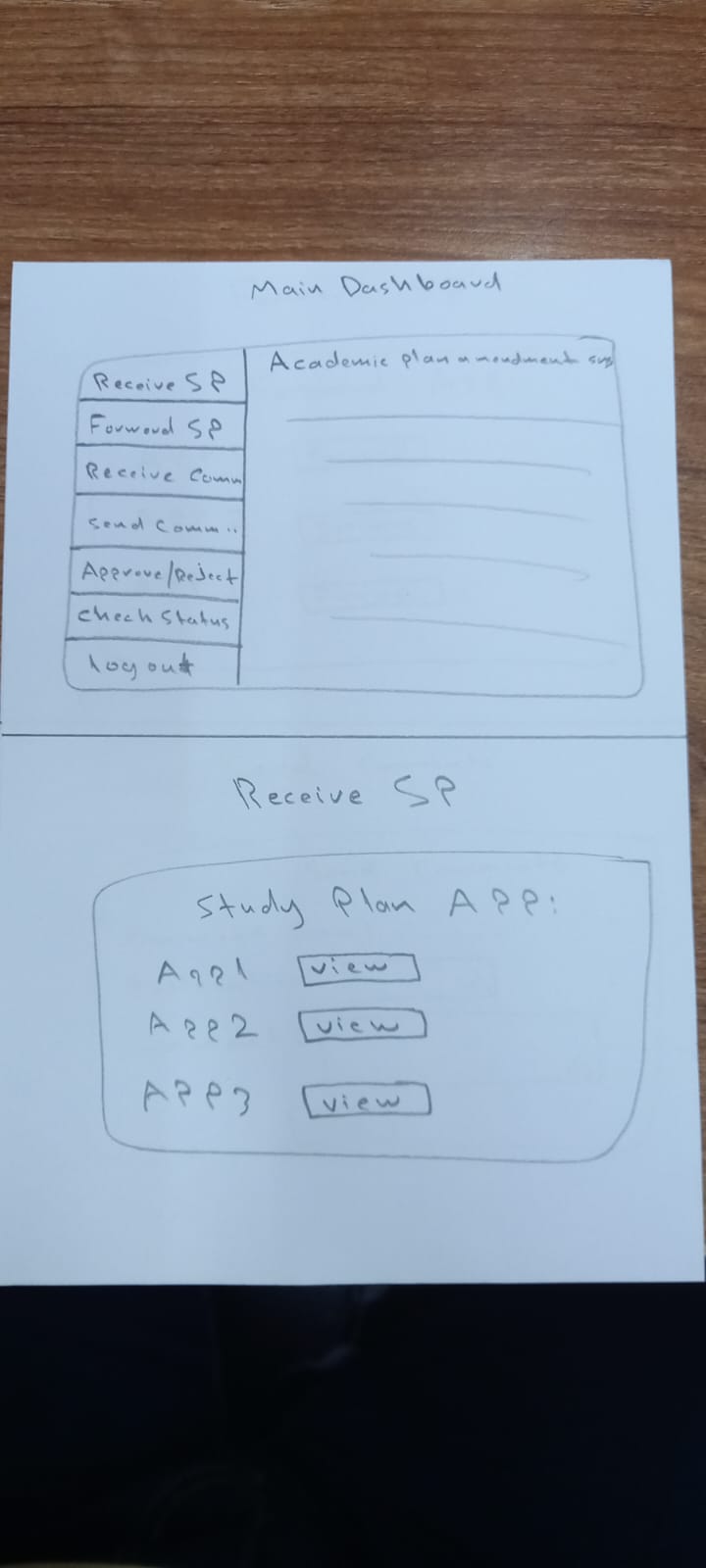


Figure 17: Main DB

A paper with writing on it

Description automatically generated

Figure 18: Receive SP

A paper with a drawing on it

Description automatically generated

Figure 19: Forward SP

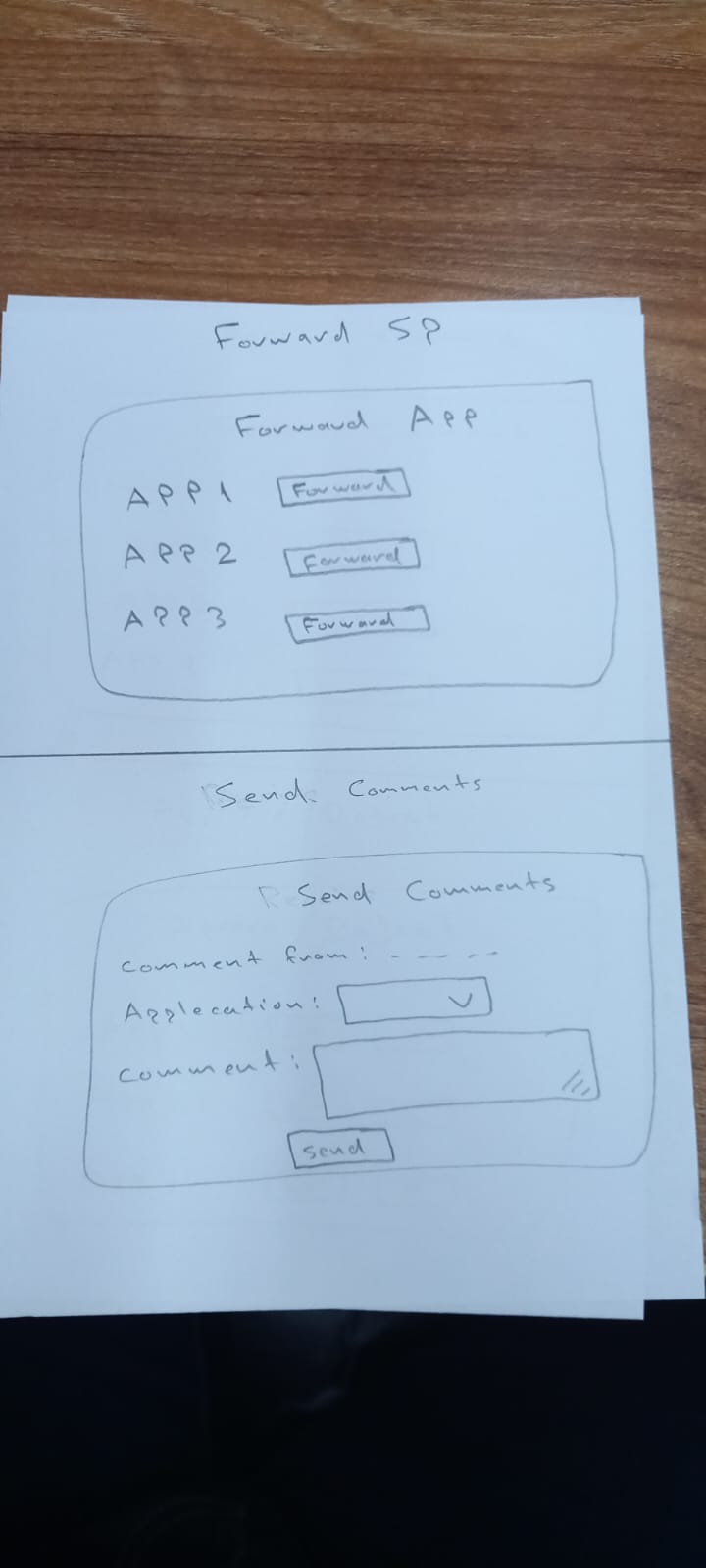


Figure 20: Send Comment

A paper with writing on it

Description automatically generated

Figure 21: Receive Comment

A paper with writing on it

Description automatically generated

Figure 22: Approve/Reject

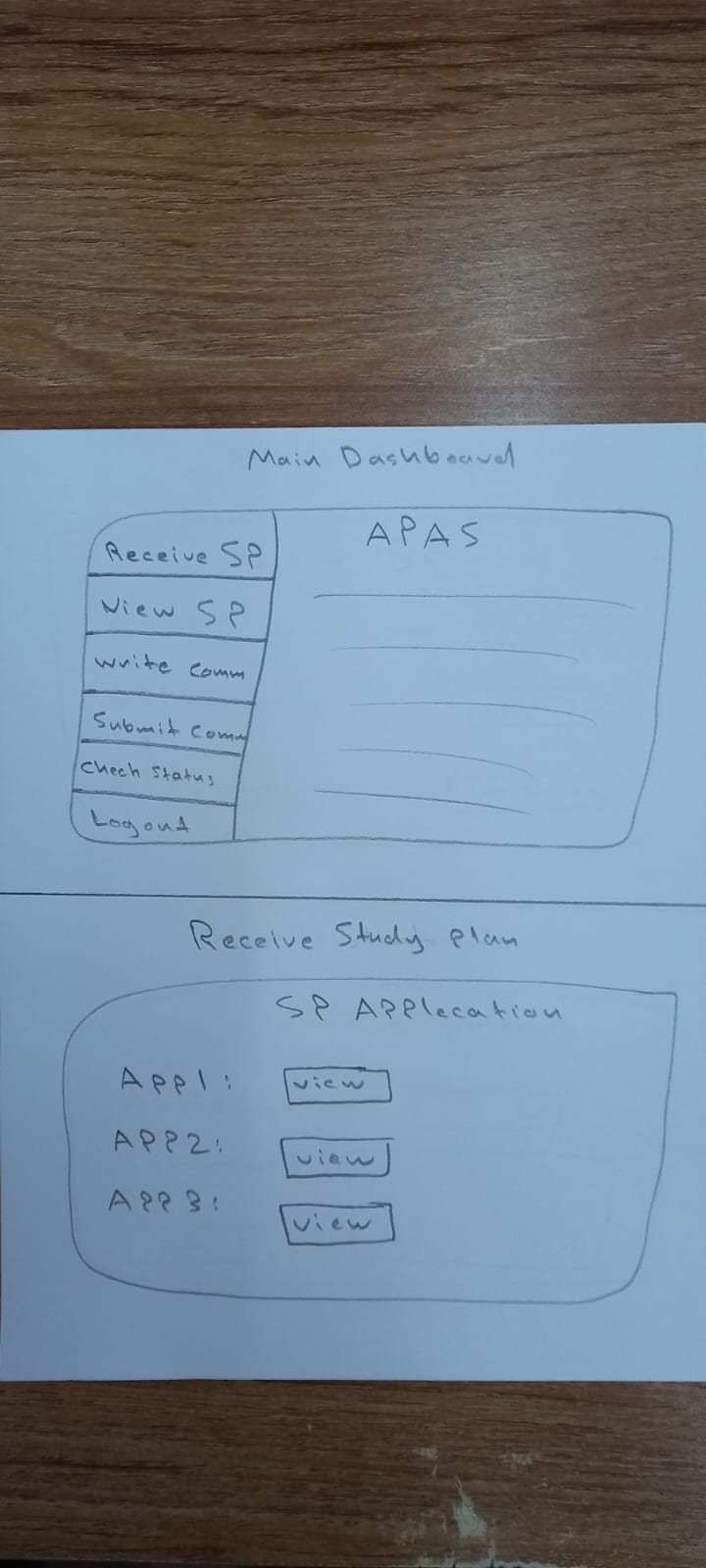


Figure 23: Main DB

A white paper with writing on it

Description automatically generated

Figure 24: Receive SP

A paper with writing on it

Description automatically generatedA paper with writing on it

Description automatically generated

Figure 25: Write Comment

Figure 26: View SP

# 4.2 Medium Fidelity Prototyping

A close-up of a dashboard

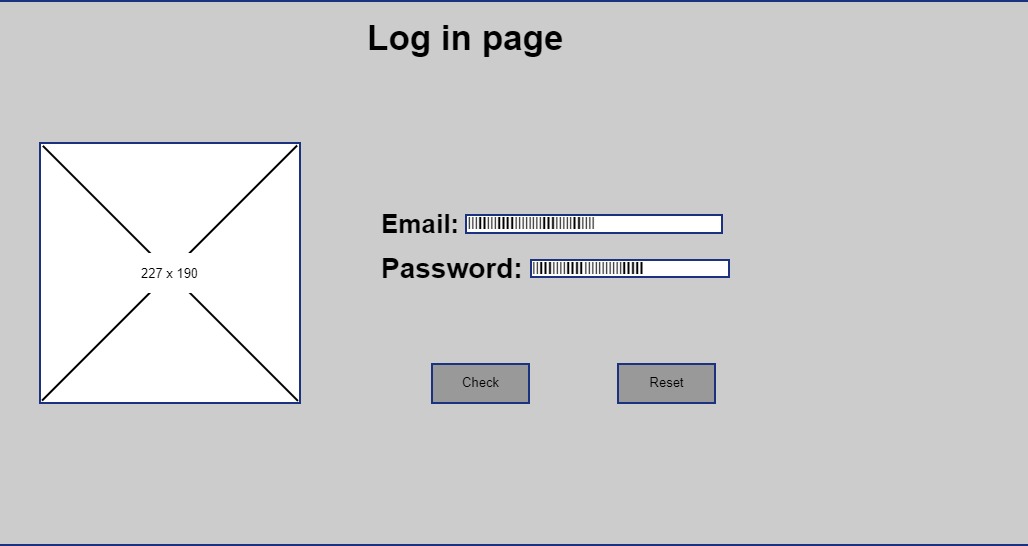
Description automatically generated

Figure 27: Main DB

Figure 28: Log In Page

A screenshot of a computer program

Description automatically generatedA screenshot of a study plan

Description automatically generated

Figure 29:Add Sp

Figure 30: Submit SP

Figure 31: Delete SP

A box with text on it

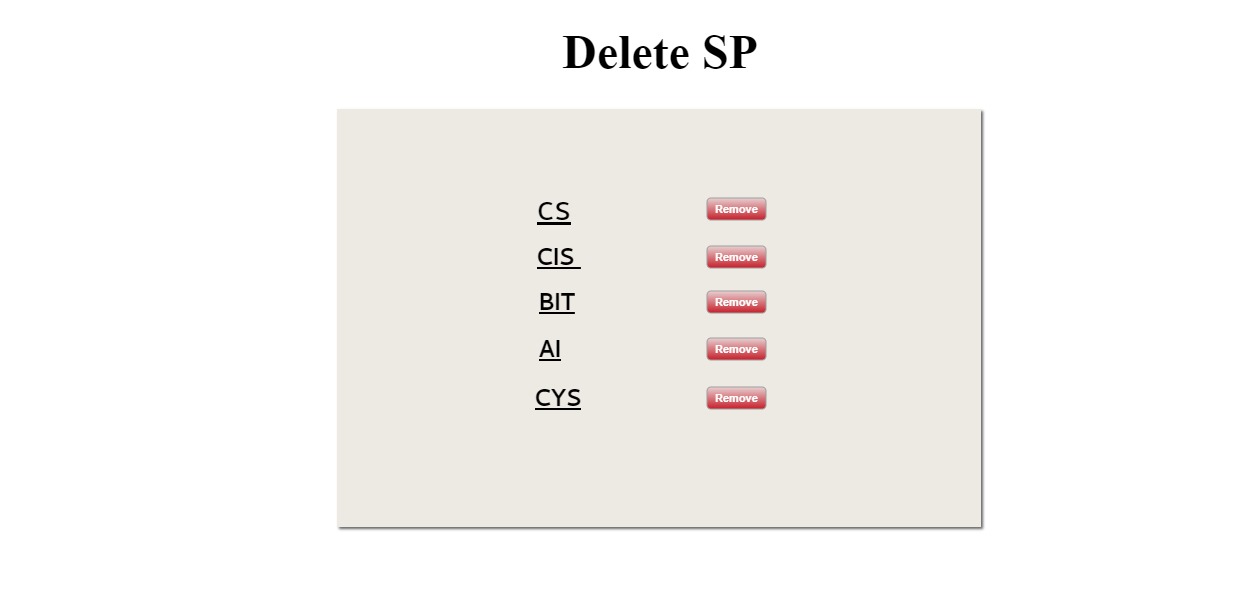
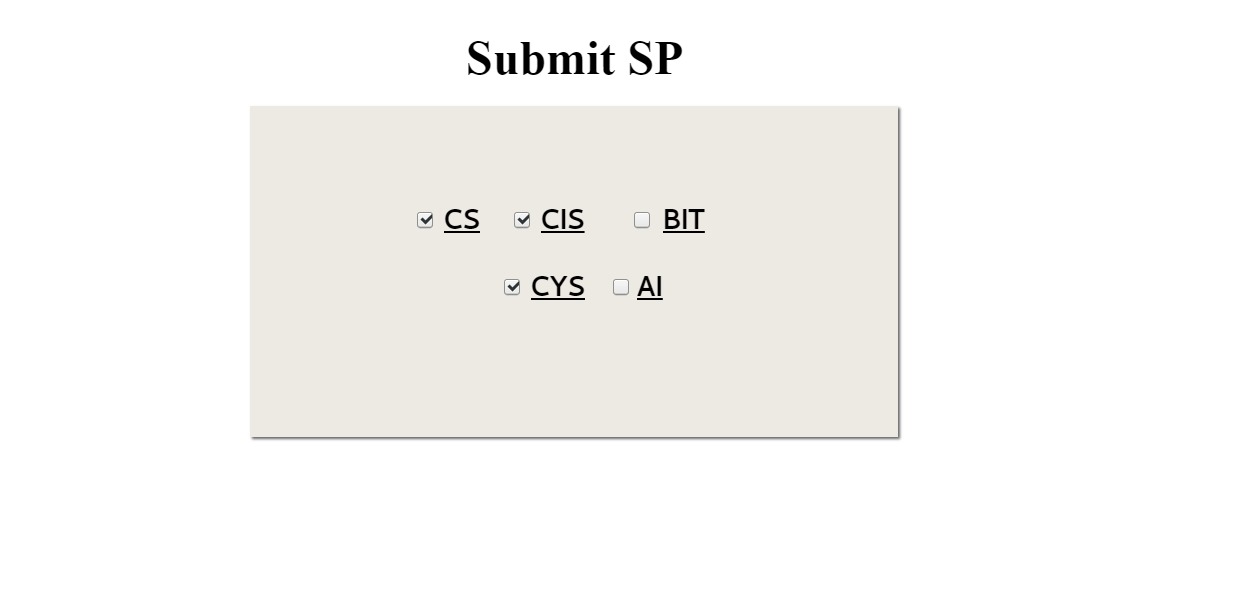
Description automatically generated

Figure 32: Comments

A close-up of a checklist

Description automatically generated

Figure 33: Check Status

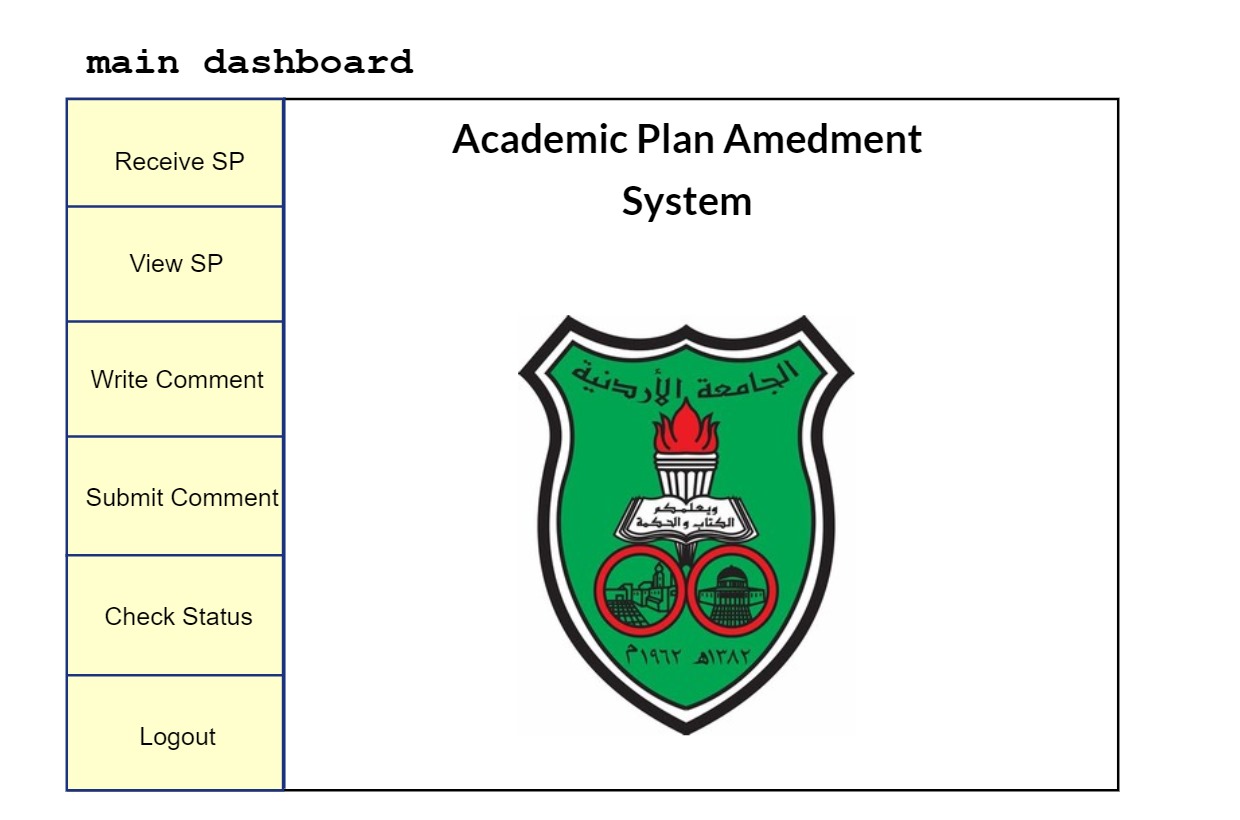


Figure 34: Main DB

A screenshot of a computer screen

Description automatically generated

Figure 35: Forward SP

A close-up of a form

Description automatically generated

Figure 36: Approve/Reject

A close-up of a box

Description automatically generated

A screenshot of a computer

Description automatically generated

Figure 37: Receive Comment

Figure 38: Send Comments

Figure 39: Main DB

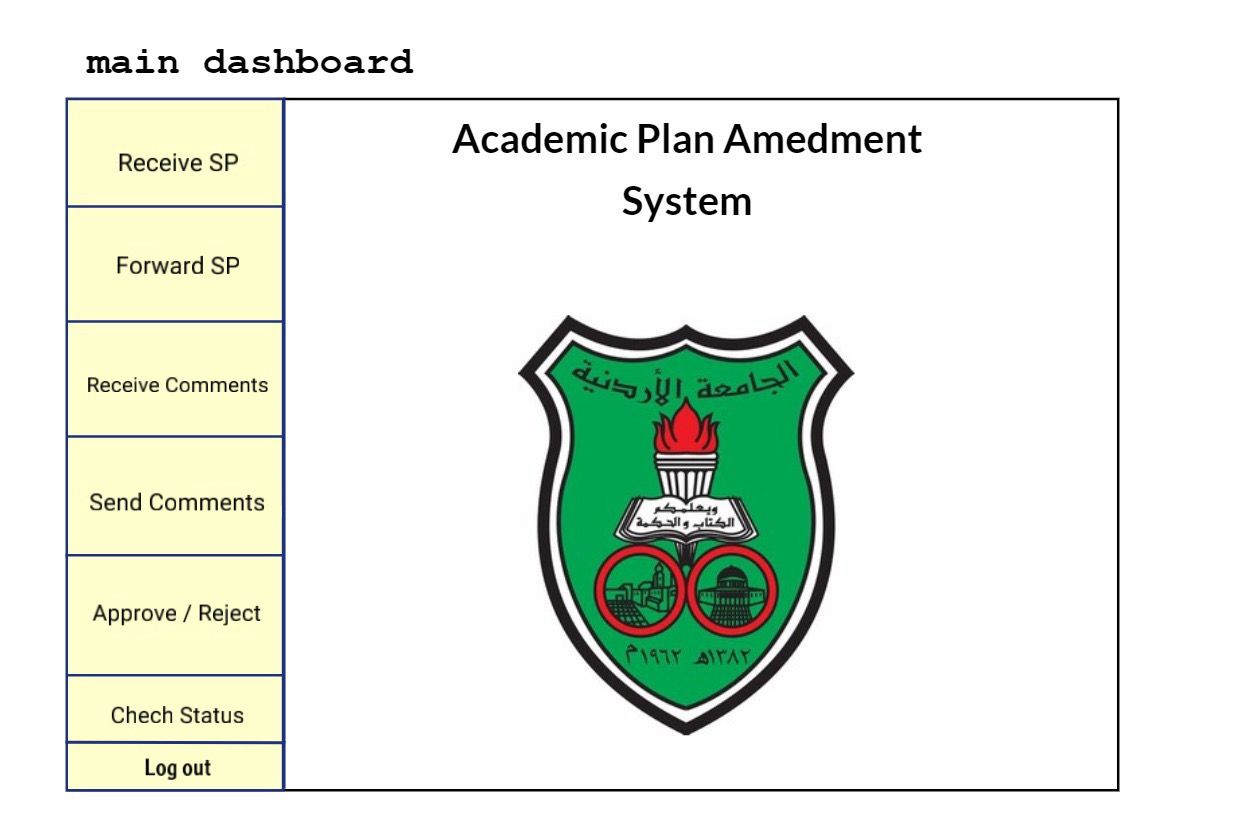
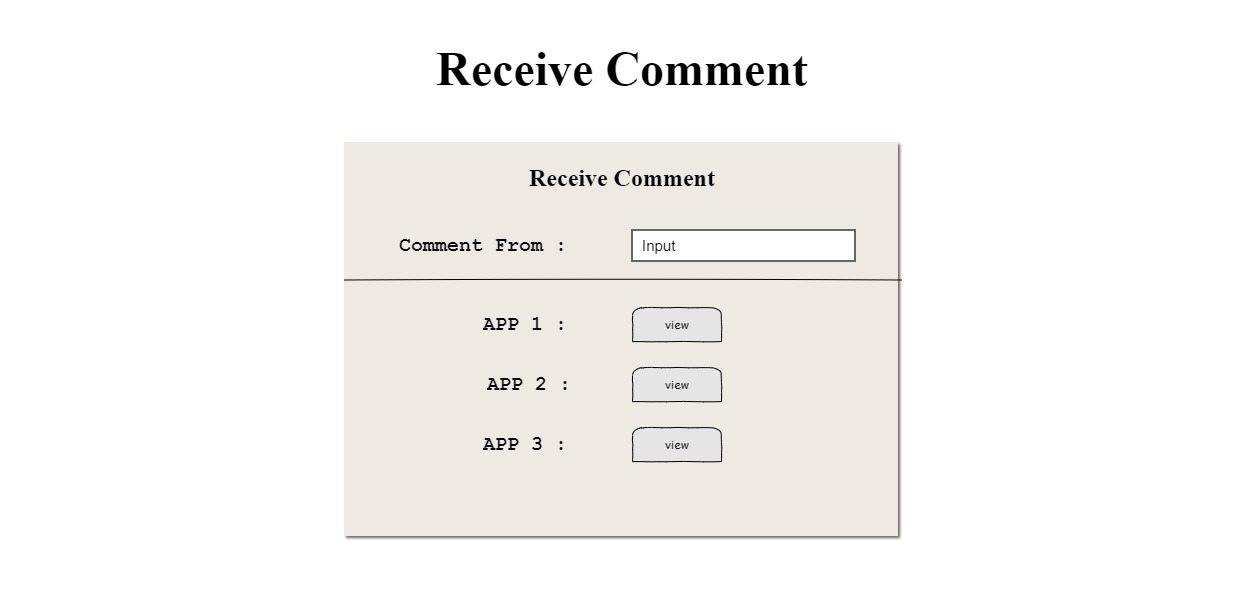


Figure 40: Receive Comment

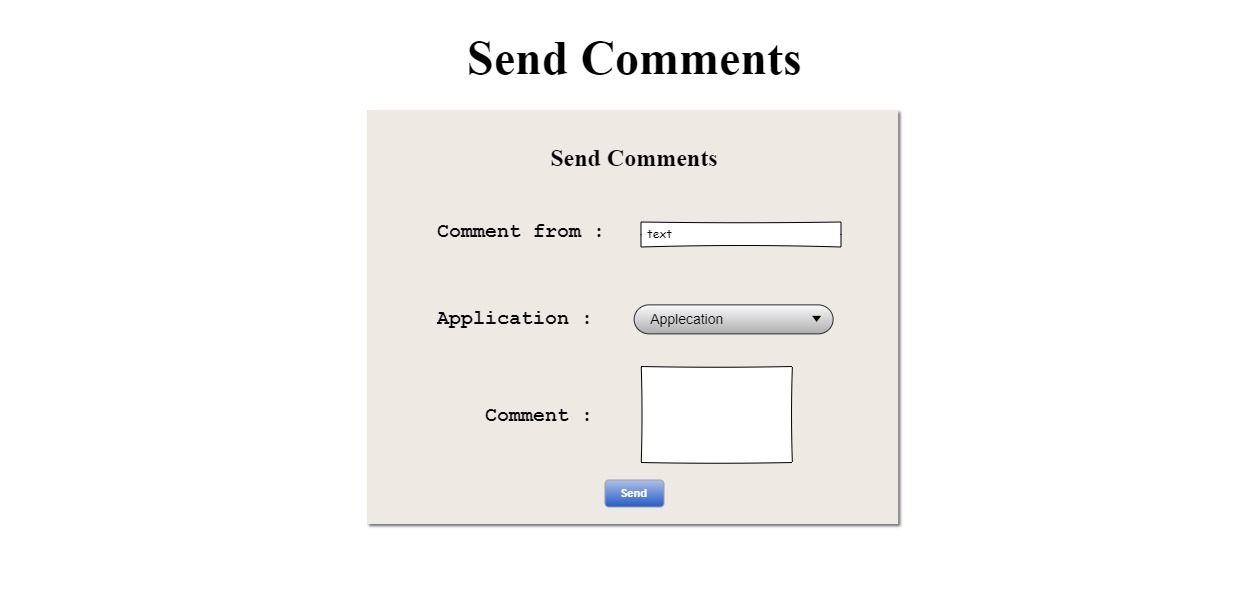


Figure 41: Send Comments



Figure 42: Approve/Reject

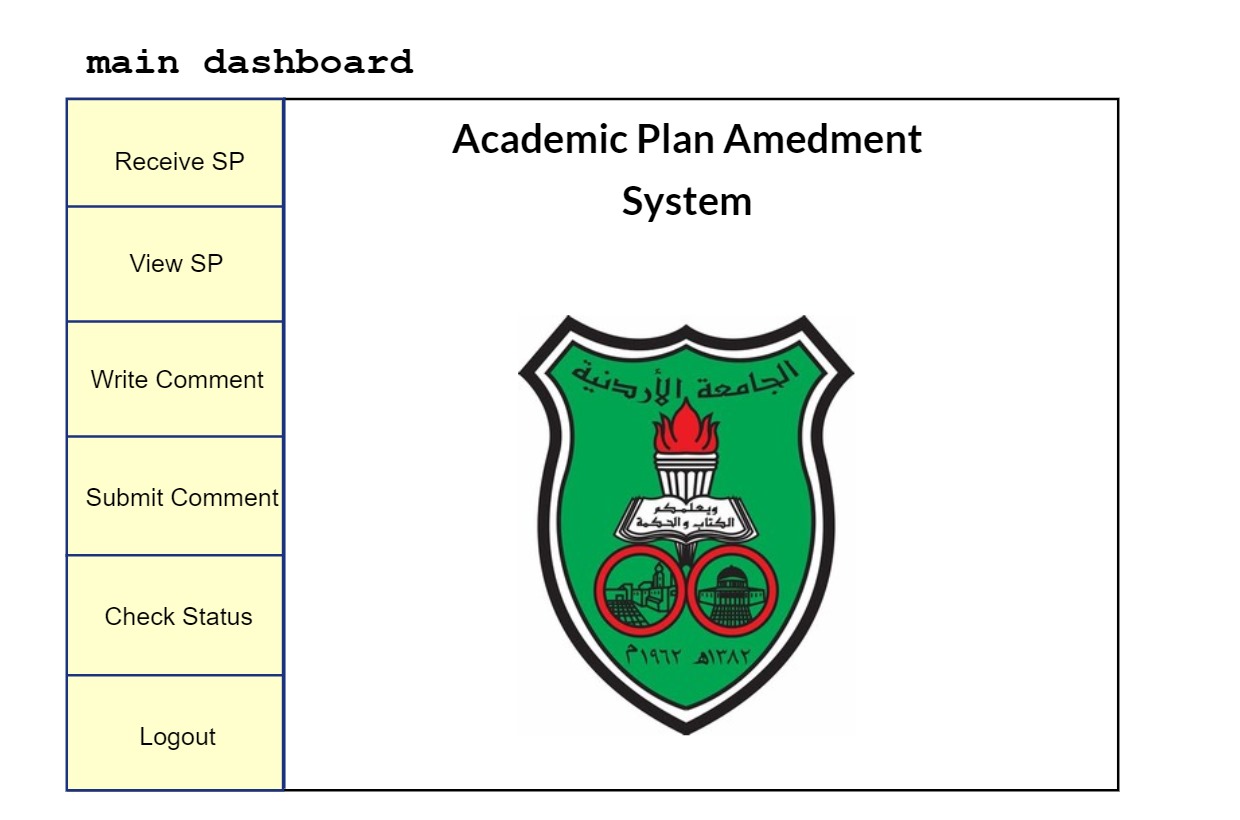


Figure 43: Main DB

A screenshot of a computer

Description automatically generated

A close-up of a checklist

Description automatically generated

Figure 44: Send Comments

Figure 45: Check Status

# 4.3 Database Design

# 

# 5.0 System Implementation

## 5.1 High-Fidelity Prototyping

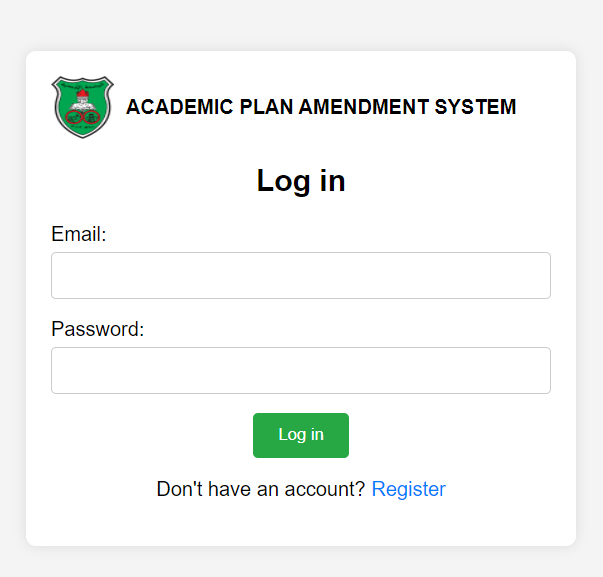


Figure 46: Log In Page

A screen shot of a registration form

Description automatically generated

Figure 47: Register

A green shield with red and black text

Description automatically generated

Figure 48: Main DB

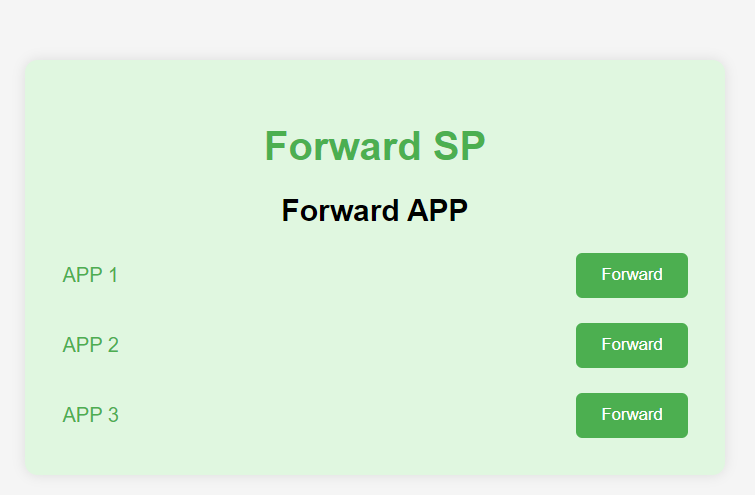
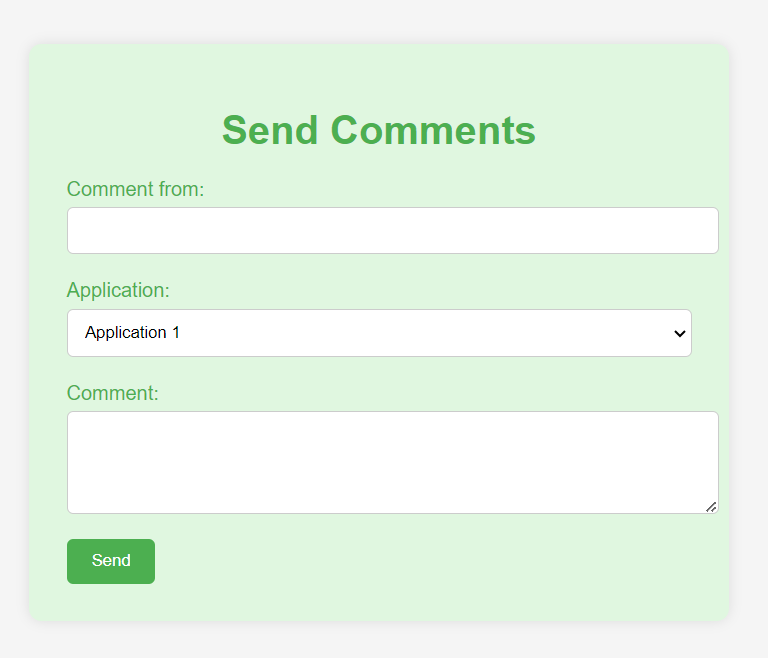


Figure 49: Send Comments

Figure 50: Forward Sp

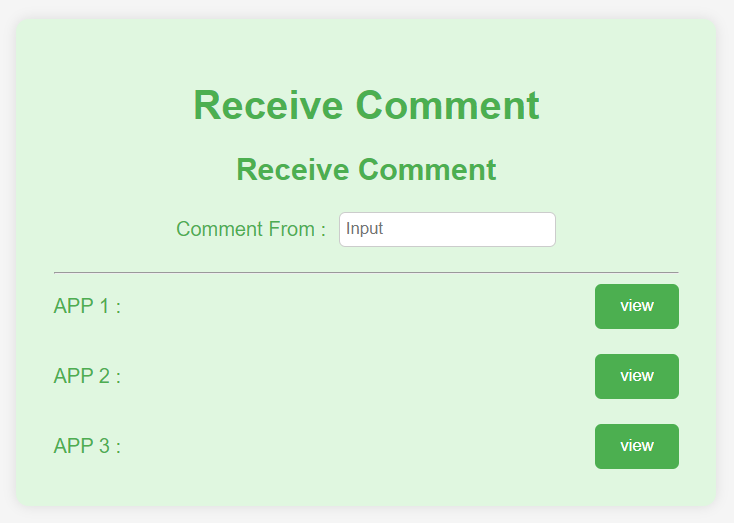


Figure 51: Receive Comments

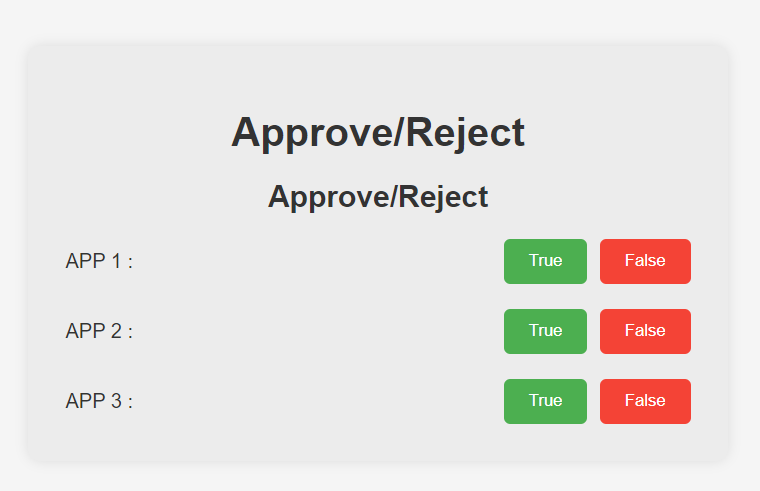


Figure 52: Approve/Reject

# 6.0 System Testing and Installation

## 6.1 Heuristic Evaluation

## 6.2 Cooperative Evaluation

# 7.0 Conclusion and Future Work

## 7.1 Overall Weaknesses

* Budget Constraints: Limited financial resources may restrict the project's ability to employ advanced technologies, hire additional expertise, or extend development timelines, potentially compromising the quality and scope of the final system.
* Security Risks: Despite planned security measures, any web-based system is vulnerable to cyber threats. Ensuring robust protection against unauthorized access and data breaches is a continuous challenge.
* Integration Challenges: Seamlessly integrating the new system with existing university databases and IT infrastructure can be complex and time-consuming, potentially leading to data consistency issues and system incompatibilities.

## 7.2 Overall Strengths

* Increased Efficiency: By automating the academic plan amendment process, the system significantly reduces the time and effort required for administrative tasks, enhancing overall operational efficiency.
* Improved Accuracy: Transitioning from manual to digital processes minimizes human errors, leading to more accurate and reliable management of academic plans and amendments.
* User-Centric Design: The system's focus on usability and user experience goals, such as efficiency, learnability, and accessibility, ensures a positive and inclusive experience for all users.

## 7.3 Future Work

* Enhanced Features and Functionalities: Continuously update and expand the system with new features based on user feedback, technological advancements, and evolving university needs to maintain relevance and effectiveness.
* Scalability Improvements: Invest in scalable infrastructure and optimize the system's architecture to handle increasing user loads and data volumes, ensuring consistent performance and reliability.
* Advanced Security Measures: Implement advanced security protocols, regular audits, and proactive threat detection mechanisms to safeguard the system against evolving cyber threats and ensure data protection.
* User Training and Support: Develop comprehensive training programs and support resources to help users transition smoothly to the new system, addressing resistance and enhancing user adoption.
* Mobile Optimization: Enhance the system's mobile accessibility and functionality to cater to users who prefer accessing the platform via smartphones and tablets, ensuring a seamless experience across all devices.
* Regular User Feedback and Iteration: Establish mechanisms for continuous user feedback and iterative improvements, allowing the system to evolve in response to user needs and preferences, thereby maintaining high levels of satisfaction and usability.

# 8.0 References

Uploaded lecture notes in Humen Computer Interaction, Jordan University.