



**ÇANKAYA UNIVERSITY  
FACULTY OF ENGINEERING  
COMPUTER ENGINEERING DEPARTMENT**

**Project Report**  
**Version 1**

**CENG 407**

Innovative System Design and Development I

**<202416>**

**Notewiz: AI-Assisted Note-Taking Application**

Eray Ateş 202011023

Halit Emir Turan 202111068

Zeynep Persil 202011031

Güneş Ölçer 201911046

Advisor: Dr. Öğr. Üyesi Faris Serdar TAŞEL



# Table of Contents

---

Table of Contents	ii
Abstract	6
1. Introduction	7
1.1 Motivation.....	7
1.2 Problem Statement.....	7
1.3 Background or Related Work.....	7
1.4 Solution Statement.....	8
1.5 Contribution.....	8
2. Literature Search	9
2.1 Library Research.....	9
2.2 Internet Research.....	9
3. Summary	10
3.1 Summary of Conceptual Solution.....	10
3.2 Technology Used.....	11
4. Software Requirements Specification	13
4.1 Introduction.....	13
4.1.1 Purpose.....	13
4.1.2 Scope of Project.....	13
4.1.3 Glossary.....	14
4.1.4 Overview of Document.....	15
4.2 Overall Description.....	15
4.2.1 Product Perspective.....	16
4.2.2 Development Methodology.....	16
4.2.3 User Characteristic.....	17
4.2.3.1 Participants.....	17
4.2.3.2 Admin.....	17
4.2.4 Operation.....	18
4.2.5 Functional Requirements.....	19
4.2.5.1 NoteWiz User Authentication and Account Management.....	19
4.2.5.1.1 Log In and Sign up.....	19
4.2.5.1.2 Forgot/Change password.....	21
4.2.5.2 Note Creation and Management.....	22
4.2.5.3 AI-powered question asking.....	24
4.2.5.4 Task Management.....	26
4.2.5.5 Cross-Platform Accessibility.....	26
4.2.5.6 Note Sharing.....	26
4.2.5.7 Document Upload and Processing.....	27
4.2.5.8 Note Creation and Visualization.....	27
4.2.5.9 User Authentication and Security.....	28
4.2.6 Constraints.....	28
4.2.7 Assumptions and Dependencies.....	28

4.3 Requirements Specification.....	30
4.3.1 External Interface Requirements.....	30
4.3.1.1 User interfaces.....	30
4.3.1.2 Hardware interfaces .....	34
4.3.1.2.1 Output Devices.....	35
4.3.1.2.2 Network Connectivity.....	35
4.3.1.3 Software interfaces .....	35
4.3.1.4 Communications interfaces .....	35
4.3.2 Functional Requirements .....	36
4.3.3 Software System attributes.....	36
4.3.3.1 Portability .....	36
4.3.3.2 Usability .....	36
4.3.3.3 Adaptability .....	36
4.3.3.4 Security.....	36
4.3.3.5 Reliability.....	37
4.3.3.6 Availability.....	37
4.3.3.7 Maintainability.....	37
4.3.4 Safety Requirement.....	37
5. Software Design Description.....	38
5.1 Introduction.....	38
5.1.1 Purpose.....	38
5.1.2 Scope.....	39
5.1.3 Glossary.....	40
5.1.4 Overview of document.....	41
5.2 Architecture design (You can use your sections and headers).....	41
5.2.1 Front-end layer.....	41
5.2.2 Back-end Layer.....	41
5.2.3 Database Layer.....	41
5.2.4 Integration and Modularity.....	41
5.3 Data structure design (You can use your sections and headers).....	42
5.4 System Design.....	43
5.4.1 Uml Diagram.....	43
5.4.2 Decomposition Description.....	43
5.4.2.1 Database Layer.....	44
5.4.2.2 User Interface Logic and Workflows.....	45
5.4.2.2.1 Sign up and Log in.....	45
5.4.2.2.2 Profile Management.....	48
5.4.2.2.3 Main Menu.....	49
5.4.2.2.4 Note Creation and Management .....	51
5.4.2.2.4.1 Note Creation and Management.....	51
5.4.2.2.4.2 Note Visualization.....	53
5.4.2.2.5 Note Sharing.....	54
5.4.2.2.6 Document Upload and Processing.....	56

5.4.2.2.7 AI Chat Page (AI-powered question asking).....	58
5.4.2.2.8 Forgot Password Pages.....	59
5.4.2.3 Admin Pages.....	61
5.4.2.3.1 User Record Page.....	61
5.4.2.3.2 User Access Control Page.....	64
5.4.2.4 Settings Page.....	66
5.5 Interface design.....	68
5.5.1 Login Page.....	68
5.5.2 Sign Up Page.....	69
5.5.3 User Profile Management Page.....	71
5.5.4 AI Chat Page.....	73
5.5.5 Note Creation and Management.....	74
5.5.6 Note Visualization.....	75
5.5.7 Note Sharing .....	76
5.5.8 Document Upload and Processing.....	77
5.5.9 Admin Pages.....	78
5.5.9.1 User Record Page.....	78
5.5.9.2 User Access Control Page.....	79
5.5.10 Forgot Password Pages.....	80
5.5.11 Settings Page.....	81
5.6 Help system design.....	82
6. Conclusions.....	84
6.1 Summary of Work.....	84
6.2 Key Learnings.....	84
6.3 Future Work and Directions.....	84
6.4 Open Problems.....	85
6.5 Advantages and Disadvantages.....	85
Acknowledgement.....	86
References.....	86
Appendices.....	87

## **Abstract**

Efficient note-taking and task management remain significant challenges for students, professionals, and individuals in various domains. Traditional methods and many existing applications fail to integrate intelligent features such as real-time collaboration, automated summarization, and seamless cross-platform functionality. This project aims to address these issues through the development of NoteWiz, an AI-powered note-taking and productivity application. NoteWiz combines state-of-the-art artificial intelligence technologies with an intuitive user interface to offer features such as automated note summarization, personalized content generation, and document processing. Leveraging frameworks like Firebase for real-time data synchronization and React for responsive design, the application ensures robust performance, data security, and accessibility. Key outcomes of the project include an AI-driven platform that enhances productivity, streamlines task management, and fosters collaboration across devices. This research emphasizes the importance of integrating cutting-edge AI tools into productivity software to meet the demands of modern users and maximize efficiency.

**Key words:** Artificial Intelligence (AI), Note-Taking Application, Productivity Tools, Real-Time Collaboration, NLP (Natural Language Processing)

## **1. Introduction**

### **1.1 Motivation**

As computer science students, our shared interest in AI and productivity software inspired us to conceptualize NoteWiz. We identified a growing demand for smart, AI-integrated applications that simplify workflows and improve task management. Leveraging our experience in React and Firebase, we aimed to build a platform that would deliver value to its users while staying compliant with modern technological advancements. This project provided an opportunity to deepen our understanding of software engineering and user experience design. Additionally, to enhance our skill set, we explored AI-focused APIs and security protocols to ensure NoteWiz meets the highest standards of functionality and data protection. By combining advanced AI capabilities with productivity tools, we aim to deliver a solution that meets diverse user needs while encouraging innovation.

### **1.2 Problem Statement**

Efficient note-taking and task management are critical for individuals seeking to enhance their productivity in professional and academic settings. However, many existing tools lack advanced features like AI assistance, real-time collaboration, and seamless cross-platform accessibility. Moreover, maintaining data security and user privacy is a significant challenge with contemporary productivity applications. The limitations of traditional note-taking methods, combined with the need for intuitive and intelligent systems, highlight the necessity of developing a robust solution. NoteWiz addresses these gaps by providing an AI-powered platform that integrates smart features such as automated summarization, document processing, and collaborative note-sharing. The project focuses on overcoming challenges related to scalability, data synchronization, and user-friendly design to ensure a reliable and impactful solution for its users.

### **1.3 Background or Related Work**

The challenge of effective note-taking and productivity management has been addressed by various applications and platforms. Notion, for instance, is a highly versatile tool widely used for its ability to combine note-taking, task management, and knowledge sharing. It offers collaborative features and customization options, allowing users to adapt the platform to their specific workflows. However, Notion primarily relies on manual input and lacks advanced AI capabilities like automated summarization or intelligent task prioritization.

Similarly, other platforms such as Microsoft OneNote and Evernote provide robust note-taking features with cross-platform synchronization. These tools excel in organizing and storing information but fall short in leveraging artificial intelligence to enhance productivity. Google Keep focuses on simplicity and real-time collaboration but offers limited customization and lacks the depth needed for complex task management or advanced content generation.

Previous research and developments in AI-integrated productivity tools highlight the potential of features such as natural language processing (NLP) for summarization and intelligent recommendations. Studies from Google and Microsoft emphasize the importance of AI in improving user engagement and efficiency. Despite these advancements, current solutions often lack a seamless combination of AI-driven insights, security, and an intuitive user experience.

NoteWiz builds on the strengths of these platforms while addressing their shortcomings. By integrating advanced AI functionalities like personalized content creation and real-time collaboration, NoteWiz aims to fill the gaps in existing systems. Its modular architecture also allows compatibility with third-party tools, enabling users to import and integrate data from applications like Notion, Evernote, or OneNote.

## 1.4 Solution Statement

NoteWiz aims to bridge this gap by offering a comprehensive AI-powered platform for note-taking and task management. Using Firebase for real-time data synchronization and React for creating responsive user interfaces, the application provides advanced features such as automated note summarization, document processing, and real-time collaboration. The integration of cutting-edge AI APIs ensures intelligent content generation and customization, enabling users to manage their workflows efficiently. Security measures, including two-factor authentication and encryption, protect user data while ensuring seamless cross-platform functionality.

## 1.5 Contribution

NoteWiz extends the capabilities of current productivity tools by integrating advanced artificial intelligence and modern design principles to address the limitations of existing solutions. Unlike traditional note-taking applications, which primarily rely on user-driven input, NoteWiz incorporates AI-powered features such as:

- **Automated Summarization:** Allowing users to quickly generate concise summaries of lengthy notes or documents.
- **Personalized Content Generation:** Enabling the creation of tailored suggestions, templates, and task prioritization based on user preferences and past behaviors.
- **Real-Time Collaboration:** Facilitating seamless sharing and editing of notes among multiple users, ensuring effective teamwork and knowledge exchange.

NoteWiz also builds on the modular and scalable architectures of platforms like Notion and Evernote by introducing AI APIs and Firebase-based real-time synchronization. These technologies ensure cross-platform compatibility, enabling users to access their data securely from multiple devices.

In addition, NoteWiz enhances data privacy and security through robust encryption and two-factor authentication, addressing a critical concern for modern productivity tools. Its

architecture allows for the seamless integration of new AI models and third-party tools, ensuring adaptability and continuous improvement.

By combining these features, NoteWiz not only improves the efficiency of note-taking and task management but also sets a new benchmark for intelligent, user-centric productivity solutions. This project contributes to the ongoing development of AI-powered applications and demonstrates the potential of emerging technologies in addressing real-world challenges.

## 2. Literature Search

For the development of NoteWiz, a comprehensive review of literature was conducted to identify the technological foundations, user needs, and gaps in existing solutions for AI-powered note-taking applications. This review combined library research to examine academic sources and internet research to evaluate practical implementations, market trends, and technical challenges.

### 2.1 Library Research

Library research involved accessing reliable academic databases, including IEEE Xplore, ACM Digital Library, and Google Scholar, to gather insights into AI-powered tools and their role in productivity solutions. Keywords such as “AI-powered note-taking applications,” “intelligent productivity tools,” and “machine learning in note-taking” were used to explore a variety of relevant studies.

One key focus of this research was on the potential of AI technologies like natural language processing (NLP) and machine learning to improve the note-taking experience. Studies such as *"GazeNoter: Co-Piloted AR Note-Taking via Gaze Selection of LLM Suggestions to Match Users' Intentions"* demonstrated how large language models (LLMs) like GPT and BERT can enhance real-time note-taking, summarization, and context-aware tagging. Research on summarization methods highlighted the effectiveness of combining **extractive summarization**—which selects key sentences directly from the text—with **abstractive summarization**, which generates new sentences to convey the essence of content. This hybrid approach was identified as a critical feature for improving user experience in NoteWiz.

The research also emphasized gaps in existing applications, such as the lack of adaptive AI features, insufficient real-time collaboration tools, and security vulnerabilities in cloud-based systems. These findings shaped the vision for NoteWiz, emphasizing the integration of advanced AI features, secure storage solutions, and seamless multi-device synchronization.

### 2.2 Internet Research

Internet research complemented the library findings by focusing on the practical challenges, competitive landscape, and emerging trends in AI-powered note-taking tools. Official documentation from React and Firebase provided technical guidance for developing scalable

and responsive applications. Additionally, industry reviews and reports from platforms like Zapier and Time offered insights into the strengths and weaknesses of existing solutions, including Microsoft OneNote, Evernote, and Notion.

Competitor analysis revealed that while these tools are popular, they lack advanced AI capabilities. Their features are often limited to basic text recognition and tagging, with minimal support for real-time collaboration or intelligent summarization. Emerging AI-powered tools that automate tasks like meeting summarization and action-item generation were identified as key market trends.

Technical challenges encountered in real-time collaboration and AI integration were explored through forums like Stack Overflow and GitHub. Firebase's real-time database was chosen for its ability to ensure seamless multi-device synchronization, while OpenAI's APIs were considered for their advanced NLP capabilities. Internet research also highlighted the importance of implementing robust security measures, including encryption and multi-factor authentication, to address privacy concerns.

By integrating insights from academic research with practical applications, NoteWiz was designed to meet user needs while addressing market gaps. Its features include adaptive AI-driven suggestions, robust real-time collaboration tools, and secure cloud-based synchronization, positioning it as a comprehensive solution for modern productivity challenges.

### **3. Summary**

#### **3.1 Summary of Conceptual Solution**

The NoteWiz project aims to provide a comprehensive, AI-powered note-taking solution designed to improve productivity and collaboration. The NoteWiz project integrates cutting-edge technology to achieve these goals. The application addresses key challenges such as inefficient summarization, disorganized content management, real-time collaboration, and security concerns through the following solutions:

##### **AI-Powered Summarization:**

- Extractive Summarization: Key phrases or sentences are extracted directly from the text, providing fast and effective previews.
- Abstractive Summarization: New sentences are created, original content is rephrased and condensed for clarity.
- A hybrid model that combines both techniques provides flexibility based on user needs.

##### **AI-Powered Content Creation and Management:**

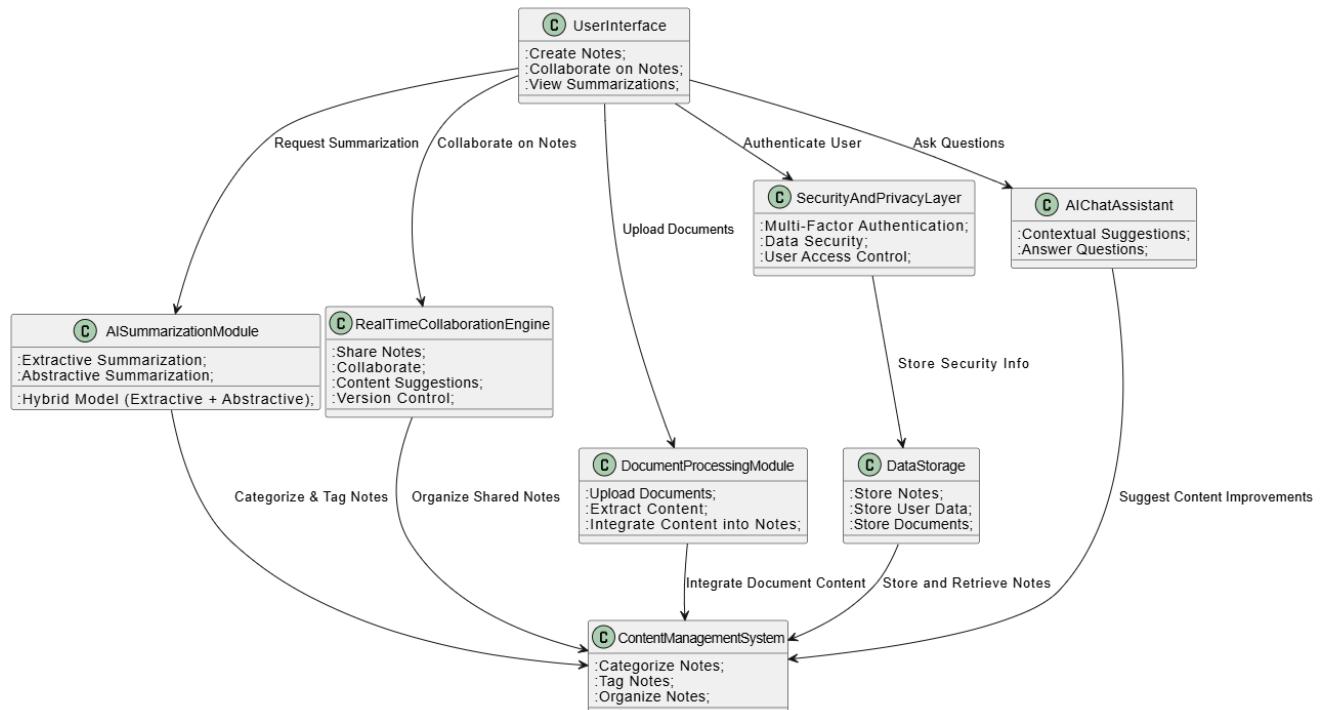
- Context-aware tagging and categorization to streamline note organization.
- Integration of an AI chat assistant for real-time question answering and contextual suggestions.
- Cover creation relevant to note content.

**Real-time Collaboration:** Users can collaborate on shared notes with AI-powered insights such as content suggestions and automatic version control.

**Document Processing and Integration:** The system allows users to upload documents (e.g. PDFs, Word files), extract relevant content, and seamlessly integrate it into their notes.

**Security and Privacy:** Multi-factor authentication protects user accounts and sensitive data.

Shapes and activity diagrams outline workflows such as note creation, sharing, and document processing, and provide clarity in user interactions and system responses.



*figure 1-Interaction Diagram Between Components*

### 3.2 Technology Used

Mention also the technology used to build the solution, such as java, .net, oracle, MySQL ...  
 Also include a block diagram of your solution.

The solution leverages modern technologies and frameworks to create a scalable, efficient, and secure platform:

**Front-end Development:** React: Used to create a responsive and intuitive user interface with support for mobile and desktop platforms.

Material Design principles are applied for consistency and usability.

**Back-end Development:** Firebase: Provides real-time data synchronization, cloud storage, and user authentication. Provides seamless cross-device data updates and secure database interactions.

**AI:** OpenAI APIs: Enable natural language processing tasks such as summarization, tagging, and question answering.

Machine learning models, including transformer-based architectures such as GPT, are used for text analysis and AI interactions.

**Database Management:** Firebase Realtime Database: Ensures fast, reliable, and scalable data processing.

**Security:** Industry-standard encryption and role-based access control mechanisms protect user data.

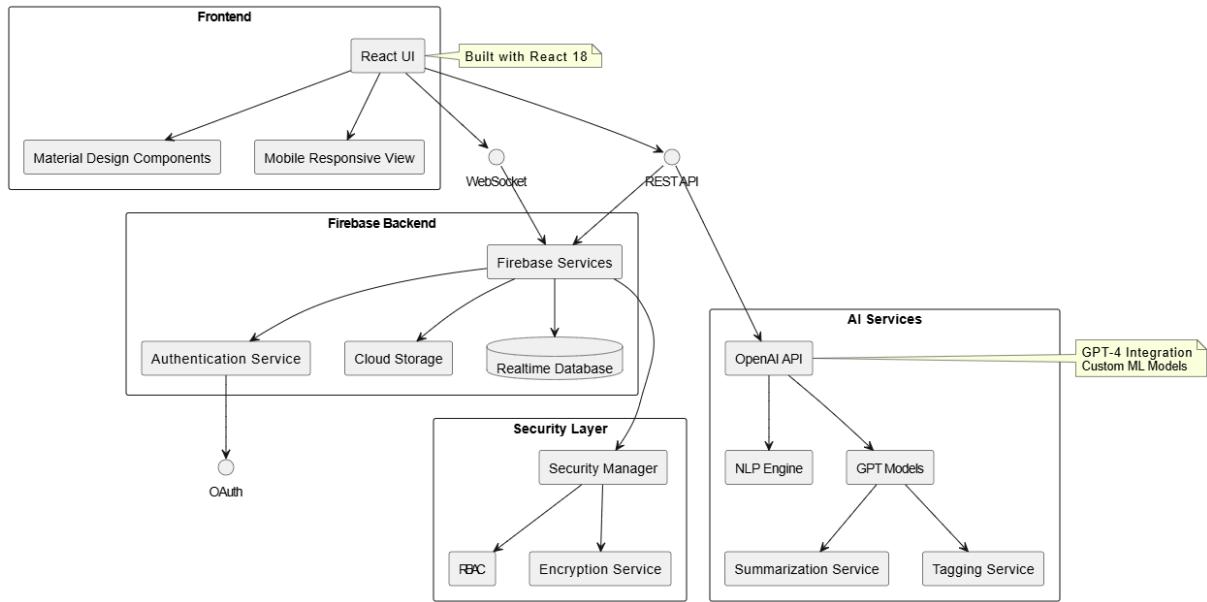


Figure2 -Technical Architecture

## 4. Software Requirements Specification

### 4.1 Introduction

#### 4.1.1 Purpose

The purpose of this document is to outline the software requirements for NoteWiz, a state-of-the-art AI-powered note-taking, planning, and organization tool. NoteWiz aims to increase user productivity and ease of working with various documents by integrating advanced AI functions such as AI questioning, task planning, and more into a distraction-free, minimalist interface. This document details the key requirements for developing, implementing, and using the application, ensuring that user needs are met effectively, while leveraging professionally integrated AI APIs to help users achieve the best results in their chosen fields and maximize the efficiency of note-taking and task management.

#### 4.1.2 Scope of Project

The NoteWiz project aims to address the limitations of traditional note-taking tools by incorporating artificial intelligence (AI) technologies that support smart summarization, real-time collaboration, and personalized content creation. This app is designed for professionals, students, and individuals who need a smart, organized system to manage notes, tasks, and projects. The scope of this project includes designing, developing, and deploying the app on mobile and tablet platforms. It will support functions such as note creation, AI-powered question asking, generating answers by combining multiple AI APIs, task scheduling, sharing notes with users, and cross-device syncing. NoteWiz aims to provide an all-in-one platform that eliminates the need for external productivity tools.

### **What NoteWiz will do:**

- **Note Creation:** Allows users to create and organize notes with rich text formatting.
- **Interaction with AI:** NoteWiz allows the user to click on the AI Tool item, label the desired area on the note page, direct it to the artificial intelligence and ask questions, and if the user approves the answer, it can be added to the labeled area as a pop-up item or added directly to the note as text.
- **Task Scheduling:** Helps users organize their tasks, set reminders, and track deadlines.
- **Cross-Device Synchronization:** Ensures that users can access their notes and tasks from any device.
- **Collaborative Note Sharing:** Users can share notes with each other.

### **What NoteWiz will not do:**

- **Replace Professional Tools:** While powerful, NoteWiz will not replace advanced project management tools or specialized software for professional use.
- **Personalized AI Models:** While the app provides generalized AI-driven tools, it does not generate fully personalized AI models specific to individual users.

**Application Benefits and Goals:** NoteWiz is targeted at individuals, teams, and organizations who require an intelligent, user-friendly note-taking and task management system. The application aims to:

- Improve accessibility to AI-powered productivity tools for all users.
- Provide personalized content and summarization based on user inputs.
- Enable efficient task management through seamless integration with note-taking functions.
- Support collaborative note sharing for other users
- Ensure data security and user privacy through robust encryption and privacy controls.

NoteWiz will also focus on providing a flexible, scalable solution that can be customized to suit the specific needs of different user groups, from students to corporate teams, while maintaining a simple, user-friendly interface.

#### **4.1.3 Glossary**

This subsection should provide the definitions of all terms, acronyms, and abbreviations required to properly interpret the SRS. This information may be provided by reference to one or more appendixes in the SRS or by reference to other documents. Example:

#### **1.3 Glossary**

Term	Definition

AI (Artificial Intelligence)	A branch of computer science that aims to create machines capable of performing tasks that would normally require human intelligence, such as learning, reasoning, and problem-solving.
NLP (Natural Language Processing)	A field of AI that focuses on the interaction between computers and human language, enabling machines to understand, interpret, and generate human language.
API (Application Programming Interface)	A set of rules and protocols that allows one software application to interact with another, enabling the integration of third-party services and functionalities.
OCR	Technology that converts different types of documents, such as scanned paper documents or images, into editable and searchable data.
Cross-Device Synchronization	The process of ensuring that data is updated and available across multiple devices in real-time, allowing users to seamlessly continue their work on different platforms.
REQ	REQ is an abbreviation for the word requirement.

#### 4.1.4 Overview of Document

The second part of this document provides a detailed description of the functionalities of **NoteWiz**, an AI-powered note-taking and productivity application. Informal requirements are outlined, setting the context for the technical specification provided in the **Requirements Specification** chapter.

The **Requirements Specification** section is designed for software developers and provides a detailed technical breakdown of the key features of the application, such as AI-powered question asking, real-time collaboration, task scheduling, and cross-device synchronization.

Both sections describe the functionality of the same product, but are targeted towards different audiences. The first part offers a high-level overview of the application, while the second part delves into the technical details necessary for developers to understand and implement the system's functionalities.

#### 4.2 Overall Description

This section of the SRS should describe the general factors that affect the product and its requirements. This section does not state specific requirements. Instead, it provides a background for those requirements, which are defined in detail in Section 3 of the SRS, and makes them easier to understand.

#### **4.2.1 Product Perspective**

NoteWiz is an AI-powered note-taking and productivity tool designed to increase user productivity through advanced features such as content summarization, task planning, real-time collaboration, and personalized content creation. The app targets professionals, students, and individuals who need an organized and intelligent system for managing notes, tasks, and projects. By integrating artificial intelligence (AI), NoteWiz provides features such as AI-powered question asking and automatic task management, ensuring a productive and distraction-free workspace. The project is divided into several components, each aimed at improving different aspects of productivity:

**Note Creation:** Allows users to create and organize notes with rich text formatting.

**AI-powered question asking:** NoteWiz allows the user to ask a question by labeling any area on the note page and directing it to the AI, and if the user approves the answer, it is added as a drop-down item in the labeled area or added directly to the note as text.

**Task Scheduling:** Helps users organize their tasks, set reminders, and track deadlines.

**Cross-Device Synchronization:** Ensures that users can access their notes and tasks from any device.

**Collaborative Note Sharing:** NoteWiz Ensures users can share notes with each other.

#### **4.2.2 Development Methodology**

For the development of NoteWiz, the team has chosen Agile methodology, focusing on an iterative and incremental approach to development. Agile ensures continuous feedback, which is essential for improving the application as it evolves. The development process will be divided into short cycles, or sprints, each lasting approximately 2 weeks. Every sprint will have defined tasks with specific goals to be achieved.

Each sprint will follow these key phases:

**Sprint Planning:** Tasks are identified and prioritized based on importance and customer requirements.

**Development & Testing:** The development team works on implementing features and functionalities, while the testing team ensures that all aspects of the app meet quality standards.

**Review & Feedback:** At the end of each sprint, the work completed is reviewed and feedback is provided to ensure the project is on track.

**Deployment:** Modules that are fully tested and functional are deployed to the live environment.

Scrum methodology is particularly suitable for **NoteWiz** due to its ability to adapt to changes quickly, ensuring timely delivery and the ability to incorporate ongoing feedback from stakeholders.

#### **4.2.3 User Characteristic**

##### **4.2.3.1 Participants**

Participants must be professionals, students, or individuals who are seeking to organize and manage their work using **NoteWiz**.

Participants should have basic knowledge of note-taking, task management, and AI functionalities.

Participants must be able to use mobile or tablet devices for accessing the application.

##### **What can users do?**

###### **User Registration and Login:**

The system allows users to sign up by creating an account.

Users can log in to the system using their credentials.

Users can log out securely when their session is complete.

###### **Profile Management:**

Users can update their profile information (e.g., name, email, etc.).

Users can change their password to ensure account security.

###### **Note Create and Management:**

Users can create new notes to store information.

Existing notes can be edited by the user.

Users can delete notes they no longer need.

Notes can be organized for better usability and access.

###### **Document Submission:**

Users can upload and send documents to the system.

###### **Theme Switching:**

Users can toggle between dark mode and light mode based on their preference.

###### **Interactions with Other Users:**

Users can send notes to other users within the system.

Users can add other users as friends.

Users can remove friends from their friend list.

###### **Interaction with AI Assistant:**

Users can communicate and interact with an AI assistant integrated into the system for various tasks.

##### **4.2.3.2. Admin**

Admins must have an advanced understanding of how **NoteWiz** works, including its AI functionalities and real-time collaboration features.

Admins should be familiar with the system's backend for managing user permissions, content sharing, and system configurations.

Admins must have knowledge of security protocols and user authentication processes.

Admins should be able to assist users with troubleshooting and guide them through advanced features.

### **What can Admin do?**

#### **Admin Registration and Login:**

The system allows users to sign up by creating an account.

Users can log in to the system using their credentials.

Users can log out securely when their session is complete.

#### **Authentication and Record Review:**

The system authenticates user identities to ensure security.

Administrators or authorized personnel can review user records for maintenance or oversight.

#### **Log Management:**

The system generates and manages logs for user activities and system operations.

#### **System Maintenance:**

The system performs maintenance tasks to ensure smooth operations, such as optimizing performance and cleaning up unnecessary data.

## **4.2.4 Operations**

#### **User Registration and Login:**

Users can register and log in to the system using their email or Google accounts. The system supports secure authentication via an external authentication API.

#### **Note Creation and Management:**

Users can create, edit, organize, and delete notes. Notes can be categorized and tagged for easier retrieval.

#### **Document Upload and Processing:**

Users can upload documents (e.g. PDFs, Word files, and images) for AI-powered questioning. Unsupported formats will trigger an error message with appropriate feedback.

#### **Task Scheduling and Management:**

Users can create tasks, set deadlines, and receive notifications for upcoming deadlines. Tasks can be prioritized based on urgency.

#### **Data Update and Deletion:**

Users can update or delete their personal information entered during registration. They can also delete notes, tasks, and uploaded documents when no longer needed.

#### **Real-Time Collaboration:**

Users can invite collaborators to shared notes and tasks, enabling real-time editing and commenting.

**Profile Management:**

Users can update their mood status, view the last note or task they interacted with, and manage their preferences from the profile section.

**AI-powered question asking:**

NoteWiz allows the user to ask a question by labeling any area on the note page and directing it to the AI, and if the user approves the answer, it is added as a drop-down item in the labeled area or added directly to the note as text.

**Cross-Device Synchronization:**

Data, including notes, tasks, and settings, will be synced across devices in real-time for seamless access.

**Note Visualization in Operations:**

When a user creates or updates a note, the application generates an automatic cover image that serves as a visual representation. Users can view these covers in the notes overview section and use them for quick identification.

## 4.2.5 Functional Requirements

### 4.2.5.1 NoteWiz User Authentication and Account Management

#### 4.2.5.1.1 Log In and Sign up

The login and registration feature allows authorized users to securely access NoteWiz. This feature provides a personalized and secure experience by verifying user credentials before granting access.

**Sequence of Actions:**

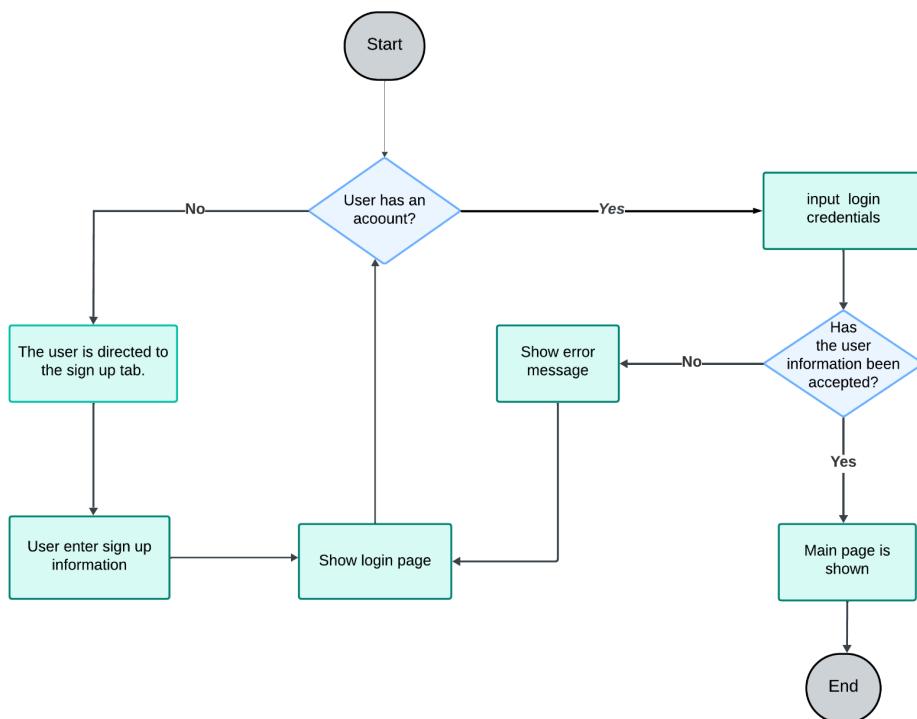
1. System: Check if the user has an account.
  - Yes: Proceed to the login process.
  - No: Direct the user to the sign-up tab.
2. User: Enters sign-up information (if not already registered).
  - System: Validate the input data for accuracy and completeness (e.g., valid email format, strong password).
  - System: Check if the email is already associated with an existing account.
  - System: If validation passes, create a new user account and send a confirmation email or message to the provided address.
  - System: If validation fails, display an error message and prompt the user to correct the input.
  - If everything is correct, system display the login page
3. System: Displays the login page.
4. User: Inputs login credentials.
5. The system checks if the credentials are correct
  - If the credentials are correct, show the main page.

- o If incorrect, display an error message.

### Associated Functional Requirements:

- REQ-1: Users should be able to sign up with their email and a password.
- REQ-2: Passwords must meet security criteria, including minimum length, inclusion of special characters, and resistance to dictionary attacks.
- REQ-3: The system should provide error messages that guide users in correcting any issues with their input (e.g., “Email already in use” or “Password is too weak”).
- REQ-4: The system must ensure that email addresses are unique and not already in use.
- REQ-5: The system should require users to verify their email address by sending a confirmation link or code.
- REQ-6: Users should input valid login credentials to access the main page.
- REQ-7: The system should display an error message if login credentials are invalid.
- REQ-8: Users should be able to log out whenever they want
- REQ-9: Logged in users should stay logged in unless they logged out (closing the app should not log out the users). The login screen should be skipped for already logged in users.
- REQ-10: Optional fields for user profile information (e.g., phone number, date of birth) should be clearly marked and not mandatory for account creation.

ACTIVITY DIAGRAM FOR USER SIGN UP AND LOGIN



*figure 3-activity diagram*

#### **4.2.5.1.2 Forgot/Change password**

The system will allow users to securely update their passwords to ensure account security. This feature will allow users to regularly update their credentials or recover access in case they forget their password.

##### **Process Sequence:**

###### **Change Password**

1. **System:** Displays the "Change Password" option in the account settings.
2. **User:** Taps on "Change Password".
3. **System:** Prompts the user to enter their current password, new password, and the new password again for confirmation.
4. **User:** Enters the current password and new password.
5. **System:** Verifies that the entered current password matches the password stored in the system.
6. **System:** If the entered password does not match and the verification attempt fails multiple times, temporarily locks the account for security reasons and notifies the user about the lockout.
7. **System:** Ensures the new password meets the required security criteria (e.g., minimum length, special characters).
8. **System:** If the verification is valid, updates the password and displays a confirmation message.
9. **System:** If the verification fails, displays an error message indicating which criteria were not met (e.g., password too short, missing characters) and prompts the user to try again.
10. **System:** Optionally suggests a strong password or provides password strength feedback to the user during the process.

###### **Forgot Password**

1. **System:** Displays the "Forgot Password?" option on the login page.
2. **User:** Taps on "Forgot Password?".
3. **System:** Prompts the user to enter their registered email address for recovery.
4. **User:** Enters the registered email address.
5. **System:** Sends a password reset link or verification code to the provided email address.
6. **User:** Clicks on the link or enters the verification code received.
7. **System:** Verifies the link/code and prompts the user to enter a new password.
8. **User:** Enters the new password and confirms it.
9. **System:** Verifies that the new password meets the required security criteria.
10. **System:** If valid, updates the password and displays a confirmation message. If invalid, prompts the user to try again.
11. **System:** Optionally suggests a strong password or provides feedback to ensure the

password is secure.

### **Relevant Functional Requirements:**

REQ-1: The system should verify the user's current password before allowing any changes to the password.

REQ-2: The system must provide password strength feedback during both the password change and recovery process to ensure users create secure passwords.

REQ-3: The system should prevent the last X passwords from being reused repeatedly to enhance security and prevent reuse of old passwords.

REQ-5: If multiple attempts to change the password fail, the system should temporarily lock the account for a specified period of time and notify the user to prevent unauthorized access.

REQ-6: The system should allow the user to securely reset their password through email or SMS verification if they are locked out due to unsuccessful attempts.

REQ-8: The system should provide the user with a secure way to reset their password if they forget it, using a verification link or code sent to their registered email address.

REQ-5: Users must be able to reset their password securely by following the password recovery steps provided by the system.

REQ-7: The system should display a confirmation message once the password has been successfully updated, both for password changes and password recovery.

#### **4.2.5.2 Note Creation and Management**

NoteWiz app allows users to easily create, manage, and edit notes. It supports features like adding media attachments and asking questions about notes with AI collaboration. Notes are securely shared and accessible across devices.

##### **Process Sequence:**

- Process Order:
- System: Displays the option to "Create a New Note" in the UI.
- User: Clicks "Create a New Note."
- System: Prompts the user to enter a title, content, and optionally add tags or categories.
- User: Enters the title and content of the note.
- System: Saves the note to the user's account and displays a confirmation message saying "The note was created successfully."
- User: Adds optional additional formatting or attachments to the note (e.g., images, links).
- System: Saves additional changes and updates the note in the user's list.
- System: Creates an image for the cover of the user's note by the AI that is related to the note's content.
- User: Can ask the AI questions about the note or attachments they added to the note.
- System: Responds to the user's question with the AI assistant.

- User: If the user approves the AI assistant's answer, they can add it to the area labeled as a pop-up item or add it directly to their note as text
- System: If the user approves the note, the system adds the AI assistant's answer to the note as a pop-up item
- System: If the user approves the note, the system adds the AI assistant's answer to the note as text
- System: Allows the user to view, edit, or delete the note from the note list. User: If the user decides to share the note, they tap the "Share" button.
- System: Prompts the user to choose who to share the note with.
- User: Shares the desired note with the selected recipients.
- System: Sends the note to the selected recipients.
- System: If the user decides to delete a note, a confirmation message ("Are you sure you want to delete this note?") is displayed.
- User: Confirms the deletion and the system removes the note from the account.

### **Relevant Functional Requirements:**

REQ-1: The system should allow users to create a new note by providing a title and content.

REQ-2: Users should be able to edit existing notes, including the ability to change the title, content.

REQ-3: Users should be able to add attachments (e.g., images, links) to notes.

REQ-4: The system should be able to respond to user questions with AI support.

REQ-5: The system should allow users to delete notes with a confirmation prompt to prevent accidental deletion.

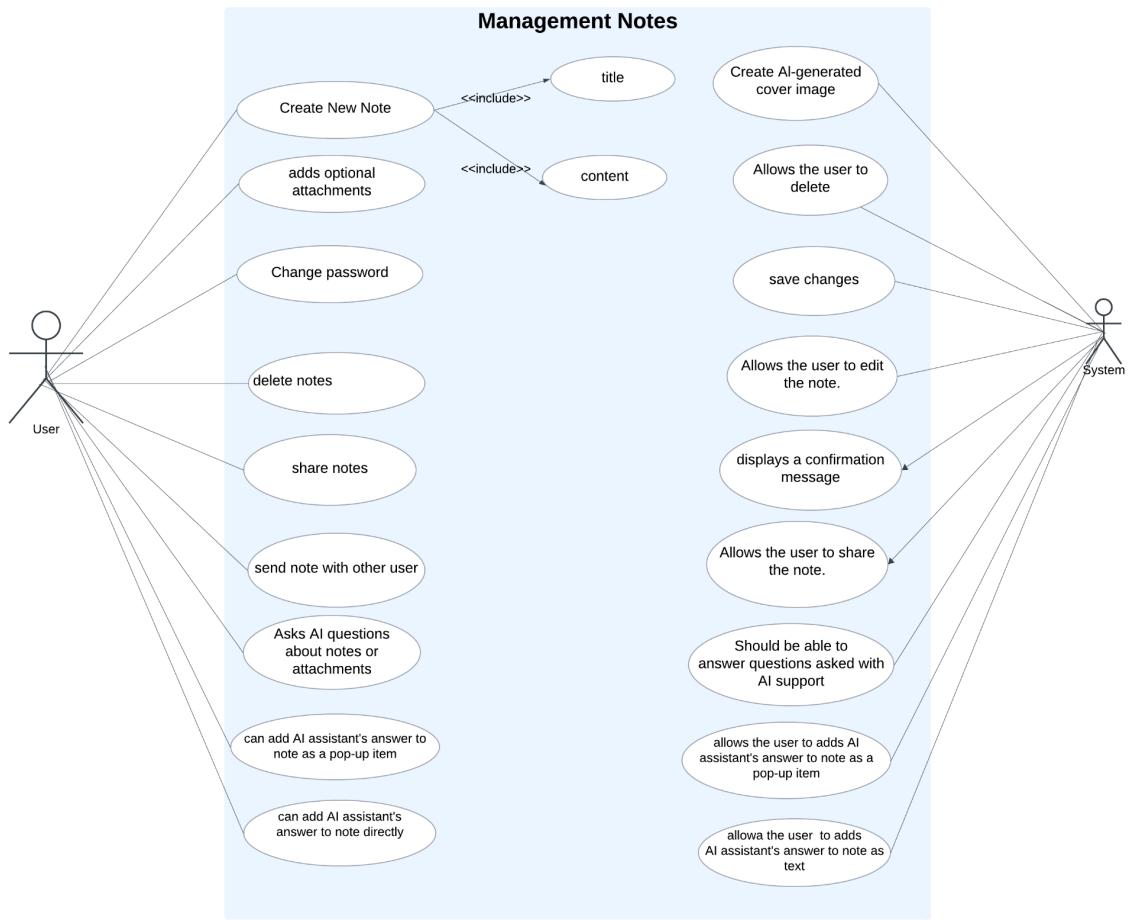
REQ-6: Notes should be securely stored in the user's account and accessible from multiple devices after logging in.

REQ-7: The system should dynamically update the user's list of notes after notes are created, edited, or deleted.

REQ-8: The system should allow users to share their notes with other users via email, link, or app with appropriate privacy controls.

REQ-9: The system should provide users with control over sharing settings (e.g., view-only or editable access).

REQ-10: Users should be notified when their notes are successfully shared or if there are any problems with the sharing process.



*figure 4-management note diagram*

#### 4.2.5.3 AI-powered question asking

NoteWiz allows the user to ask a question by labeling any area on the note page and directing it to the AI, and if the user approves the answer, it is added as a drop-down item in the labeled area or added directly to the note as text.

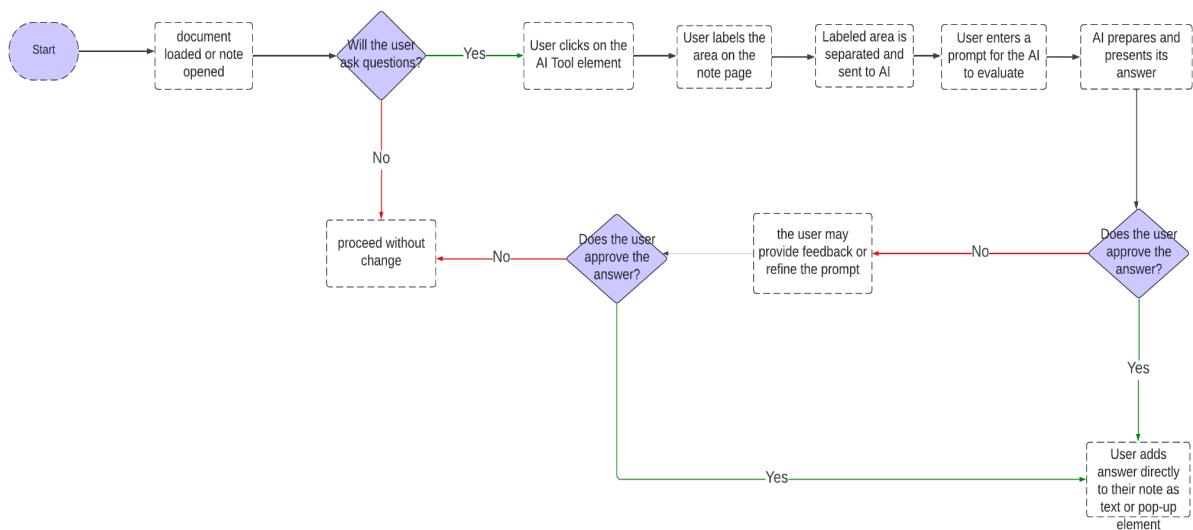
##### Process Sequence:

1. **System:** Displays the "AI Tool" option on the note page.
2. **User:** Clicks on the "AI Tool" option.
3. **System:** Allows the user to label a specific area on the note page.
4. **User:** Labels the desired area for AI input.
5. **System:** Separates the labeled area from the rest of the note and prepares it for AI processing.
6. **User:** Enters a prompt or query related to the labeled content.
7. **System:** Processes the prompt using the AI and generates a response.

8. **System:** Displays the AI-generated response to the user.
9. **User:** Reviews the response.
  - o If satisfied, the user approves the response.
  - o If not satisfied, the user may provide feedback or refine the prompt.
10. **System:** Upon user approval:
  - o Allows the user to add the response as a pop-up element in the labeled area.
  - o Or directly integrates the response as text into the note.
11. **System:** Saves the updated note and notifies the user that the changes have been saved successfully.

### **Relevant Functional Requirements:**

- REQ-1:** The system must provide an option for users to label specific areas on the note for AI processing.
- REQ-2:** The system must allow users to input prompts or queries for AI evaluation based on the labeled content.
- REQ-3:** The AI must generate relevant responses based on user input and present them in an easily readable format.
- REQ-4:** Users must be able to review and approve AI responses before adding them to the note.
- REQ-5:** The system should support adding AI responses as either pop-up elements or inline text within the note.
- REQ-6:** All updates made to the note must be saved automatically, ensuring no data loss.
- REQ-7:** The system should allow users to refine their prompts and request updated responses from the AI if needed.
- REQ-8:** The system must notify users of successful updates to their notes after changes are made.



*figure 5-AI-powered question asking*

#### **4.2.5.4 Task Management**

The system enables users to create, edit, and prioritize tasks efficiently. Tasks can be associated with deadlines and categorized for better organization. Users can integrate tasks with a calendar and set reminders to track their progress.

##### **Process Sequence:**

1. **User:** Creates or edits a task.
2. **System:** Saves the task and updates the task list.
3. **User:** Sets a deadline and a reminder for the task.
4. **System:** Integrates the task into the calendar and triggers reminders at the set times.

##### **Relevant Functional Requirements:**

**REQ-1:** The system must support task creation and editing with deadlines.

**REQ-2:** Users must be able to categorize and prioritize tasks.

**REQ-3:** The system should provide calendar integration for task tracking.

**REQ-4:** Users should receive reminders for tasks based on their settings.

#### **4.2.5.5 Cross-Platform Accessibility**

Users can access their accounts from multiple devices, ensuring seamless data synchronization and continuity across platforms.

##### **Process Sequence:**

1. User: Logs into their account on a new device.
2. System: Authenticates the user's credentials.
3. System: Synchronizes all user data (notes, tasks, settings) from the cloud to the device.
4. User: Makes changes (e.g., edits a note or completes a task).
5. System: Automatically syncs these changes to all other logged-in devices in real time.

##### **Relevant Functional Requirements:**

- **REQ-1:** The system must synchronize user data across multiple devices in real time.
- **REQ-2:** The system must provide secure, encrypted connections during data synchronization.

#### **4.2.5.6 Note Sharing**

Users can share notes with others, enabling seamless collaboration on projects or information sharing.

Users can send their notes directly to other users through the application without creating a separate copy.

### **Process Sequence:**

1. **User:** Selects a note and clicks the "Send Note" button.
2. **System:** Prompts the user to choose a recipient from their contact list or enter the recipient's email/username.
3. **User:** Confirms the recipient and sends the note.
4. **System:** Sends the note to the recipient via the app's internal messaging system.
5. **Recipient:** Receives the note in their inbox and can view it within their account.

### **Relevant Functional Requirements:**

**REQ-1:** The system must allow users to send notes directly to other users within the application. **REQ-2:** The system should notify recipients of incoming notes.

**REQ-3:** Sent notes should retain all content, including attachments and formatting.

### **4.2.5.7 Document Upload and Processing**

Enables users to upload documents such as PDFs, DOCX files, or images for analysis. Uploaded documents can be summarized, have text extracted, or be used for Q&A purposes.

### **Process Sequence:**

1. **User:** Uploads a document.
2. **System:** Processes the document for Q&A.
3. **User:** Reviews the processed output and integrates it into notes or tasks if desired.

### **Relevant Functional Requirements:**

- **REQ-1:** The system must support uploads of common document formats (PDF, DOCX, images).
- **REQ-2:** Documents should be processed for Q&A.
- **REQ-3:** Users must be able to integrate processed content into their notes.

### **4.2.5.8 Note Creation and Visualization**

For every new note, the system generates a unique visual representation (e.g., a color-coded or image-based cover). These covers help users quickly identify notes in the list view.

### **Process Sequence:**

1. **User:** Creates a new note.
2. **System:** Automatically generates a visual cover for the note based on its content or metadata.
3. **System:** Displays the cover in the notes list view.

### **Relevant Functional Requirements:**

**REQ-1:** The system must generate visual covers for each new note.

**REQ-2:** Covers must be customizable or regenerable based on user preference.

**REQ-3:** Covers should be displayed alongside notes in the list view.

#### 4.2.5.9 User Authentication and Security

Users can log in securely using their email or Google accounts. The system encrypts and securely stores user data to prevent unauthorized access.

##### Process Sequence:

5. **User:** Logs in using email or Google account.
6. **System:** Authenticates the credentials and grants access.
7. **System:** Encrypts and securely stores user data.

#### 4.2.6 Constraints

##### 1. Regulatory Policies:

- The application must comply with data privacy regulations, such as **GDPR** (General Data Protection Regulation) and **CCPA** (California Consumer Privacy Act), ensuring user data is stored, processed, and shared securely.
- Compliance with app store policies (e.g., Google Play Store and Apple App Store) is required for successful distribution.

##### 2. Hardware Limitations:

- The application is designed to function efficiently on devices with a minimum of **2 GB RAM** and requires stable internet connectivity for real-time synchronization and AI features.
- The app must be compatible with smartphones and tablets running Android or iOS, with no specific support for older or unsupported operating systems.

##### 3. Interfaces to Other Applications:

- The software integrates with external APIs (e.g., Firebase for authentication and real-time database) and AI APIs for summarization and question answering.
- Compatibility with standard file formats (e.g., PDF, DOCX) for document uploads and processing is a must.

##### 4. Parallel Operation:

- The system supports concurrent operations, such as simultaneous note editing and sharing by multiple users. However, conflicts are managed using a "last-save-wins" mechanism or versioning.

##### 5. Audit Functions:

- Limited audit functionality is included for note history tracking. Actions such as creating, editing, and sharing notes are logged for user reference.

##### 6. Control Functions:

- User control over privacy settings (e.g., visibility of shared notes) and AI interactions is included to enhance security and customization.

**7. Higher-Order Language Requirements:**

- The application is developed using **JavaScript (React)** for the front-end and **Firebase** for the back-end, ensuring efficient development and scalability.

**8. Reliability Requirements:**

- The system must maintain **99% uptime**, with regular backups to prevent data loss.
- The AI services must provide a response within **2 seconds** for a seamless user experience.

**9. Criticality of the Application:**

- As a productivity tool, **Notewiz** is not life-critical but must ensure data integrity and usability to avoid disruptions to user workflows.

**10. Safety and Security Considerations:**

- Two-factor authentication and role-based access control (RBAC) are mandatory to safeguard user data.
- End-to-end encryption is implemented to secure data during storage and transfer.

#### **4.2.7 Assumptions and Dependencies**

##### **Operating System Availability:**

It is assumed that the target devices will run **Android 8.0+** or **iOS 12+** to ensure compatibility with the app's features.

The availability of desktop versions is assumed to depend on React's compatibility with popular web browsers.

##### **Internet Connectivity:**

A stable internet connection is assumed for real-time synchronization, AI services, and cloud storage operations. Limited offline functionality is provided for note creation only.

##### **Third-Party APIs:**

The application assumes continued support and availability of **Firebase** for authentication and database services and **AI APIs** for summarization and question answering. Changes or outages in these services may affect application performance.

##### **User Proficiency:**

The target users are assumed to have basic familiarity with mobile applications and note-taking tools. The design accommodates intuitive navigation for ease of use.

##### **Hardware Dependencies:**

The application depends on the availability of hardware supporting features like file uploads (e.g., cameras, file browsers) and sufficient storage capacity for caching and temporary data storage.

### **Regulatory Changes:**

It is assumed that there will be no significant changes to the regulatory policies (e.g., GDPR) that could require a major redesign of data handling processes.

### **Development Environment:**

The development assumes stable support for the selected frameworks and libraries (e.g., React, Firebase) during the project lifecycle.

### **User Base and Scalability:**

Initial user load is expected to be manageable within Firebase's free-tier limits. Increased usage might require upgrading to higher-tier plans.

## **4.3 Requirements Specification**

### **4.3.1 External Interface Requirements**

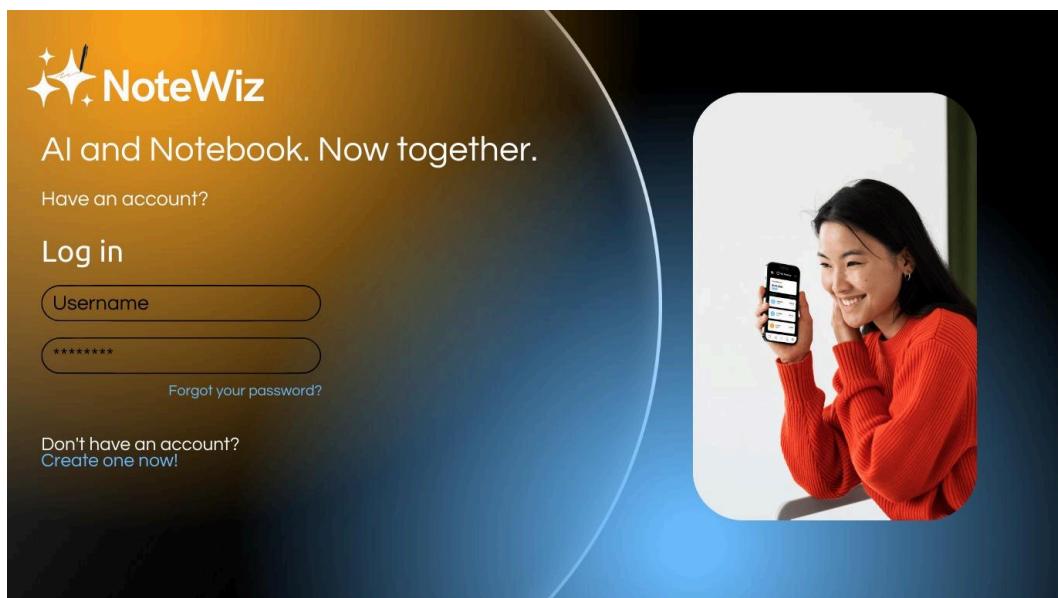
#### **4.3.1.1. User interfaces**

The user interface (UI) is designed to be intuitive, simple, and easy to navigate. Key UI components include:

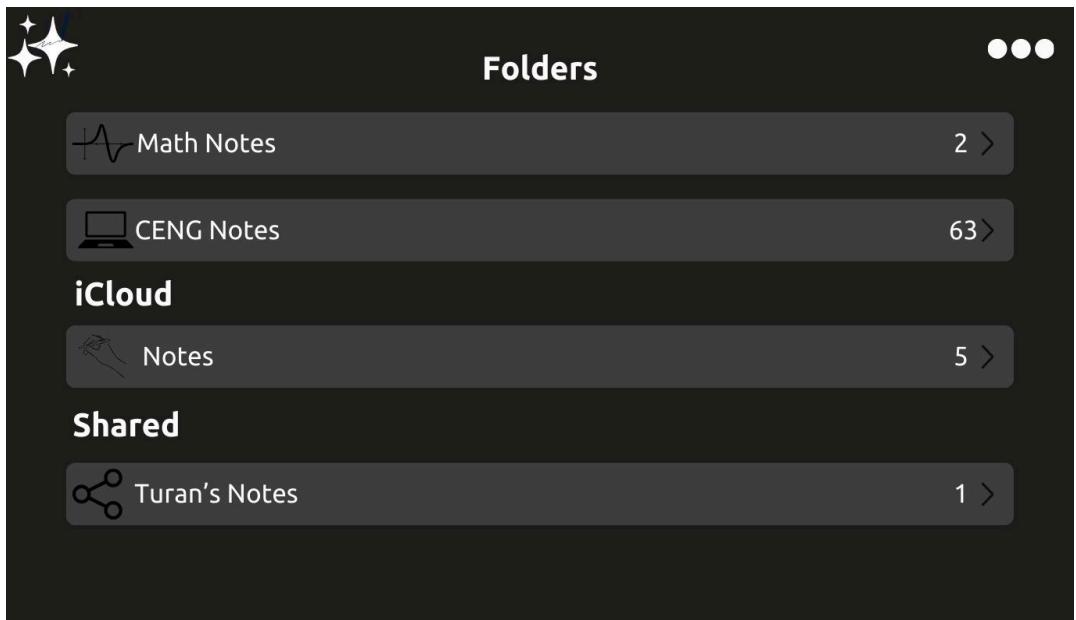
- **Home Screen:** Provides quick access to recently opened notes, tasks, and collaborative projects. *Figure 7, Figure 8, Figure 9*
- **Login:** allows users to securely log in to their NoteWiz account using their credentials (email and password). *Figure 6*
- **Sign Up:** New users can quickly sign up for an account and create a new user profile by entering their email and password. *Figure 6*
- **Forgot/Change Password :** If a user forgets their password, they can easily recover it by receiving a password reset link via email. This feature provides secure access to the application while preventing unauthorized account access. *Figure 6*
- **Note Creation:** Users can easily create and edit notes with a minimalist text editor. *Figure 8, Figure 9*
- **Task Management Interface:** Users can create, update, and track tasks with an integrated calendar and reminder system.
- **Collaborative Note Sharing:** Users can share notes with each other. *Figure 7*
- **Settings:** Allows users to customize the app's appearance (e.g., Light/Dark Mode), manage their account settings, and adjust notification preferences. *Figure 10*
- **Interaction with AI interface**

1. NoteWiz allows the user to click on the AI Tool item, label the desired area on the note page, direct it to the artificial intelligence and ask questions, and if the user approves the answer, it can be added to the labeled area as a pop-up item or added directly to the note as text. **Figure 11, Figure 12**
- **Upload Document:** Allows users to upload various document types (e.g., PDF, DOCX, images) for processing. The uploaded files can be used for AI-powered summarization, note creation, or question-answering. **Figure 12**
- **Note Visualization:** When users create a new note, the system automatically generates a visual cover for the note, which is displayed as a thumbnail on the notes list screen. This visual element improves the user's ability to quickly identify and organize notes. **Figure 12**

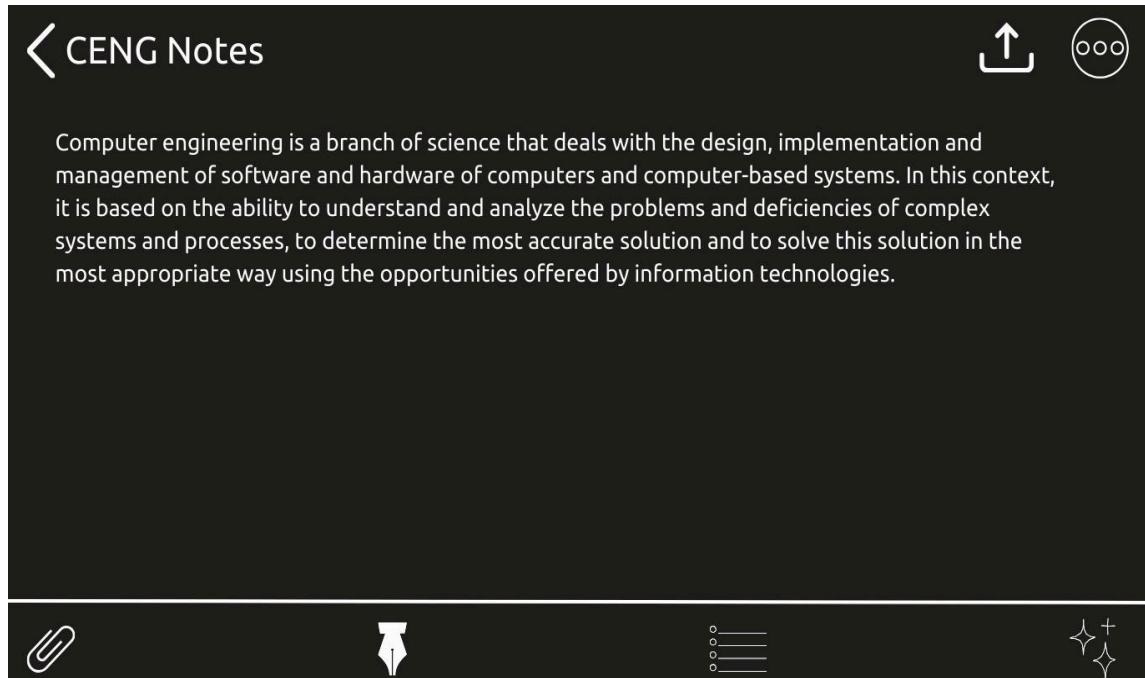
These features will be seamlessly integrated into the **NoteWiz** interface, providing users with powerful tools for quick content review and detailed question answering. The goal is to make it as intuitive as possible for users to access these advanced AI features while maintaining a clean and distraction-free experience.



**Figure 6-Log In-Sign up-Forgot your password**



**Figure 7- Home Page -Also showm note sharing**



**Figure 8-Home screen- create note and other option**

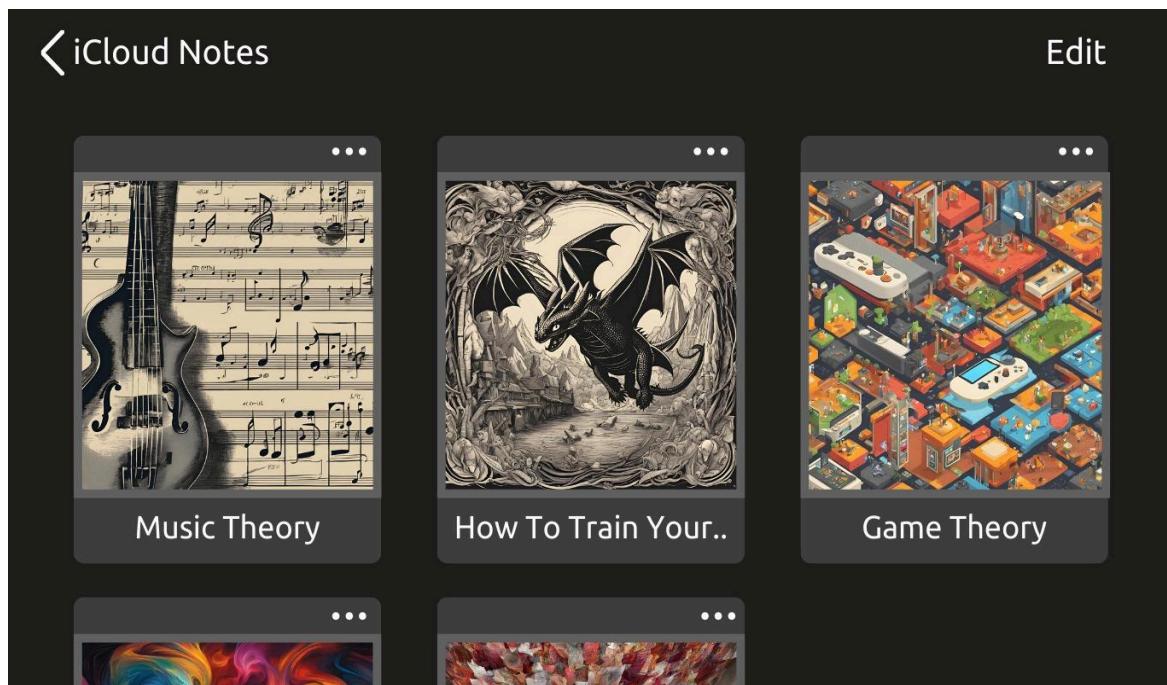


Figure 9- Home Page -Note Visualization-Note creation

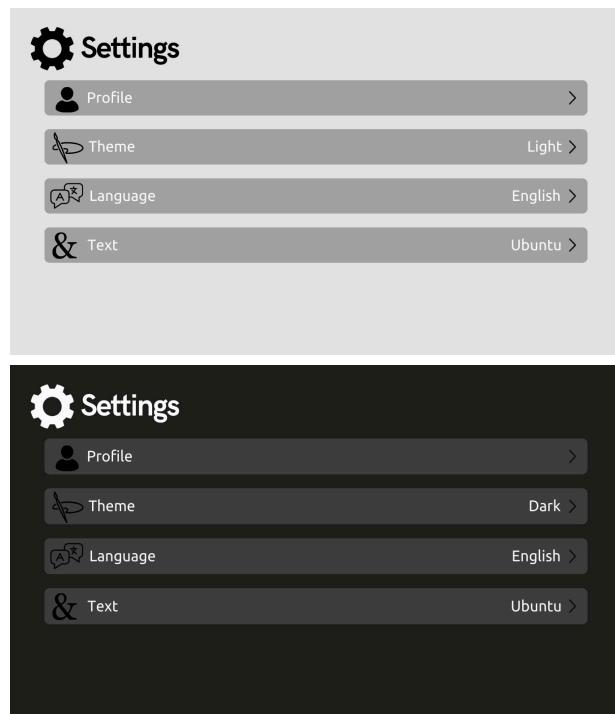
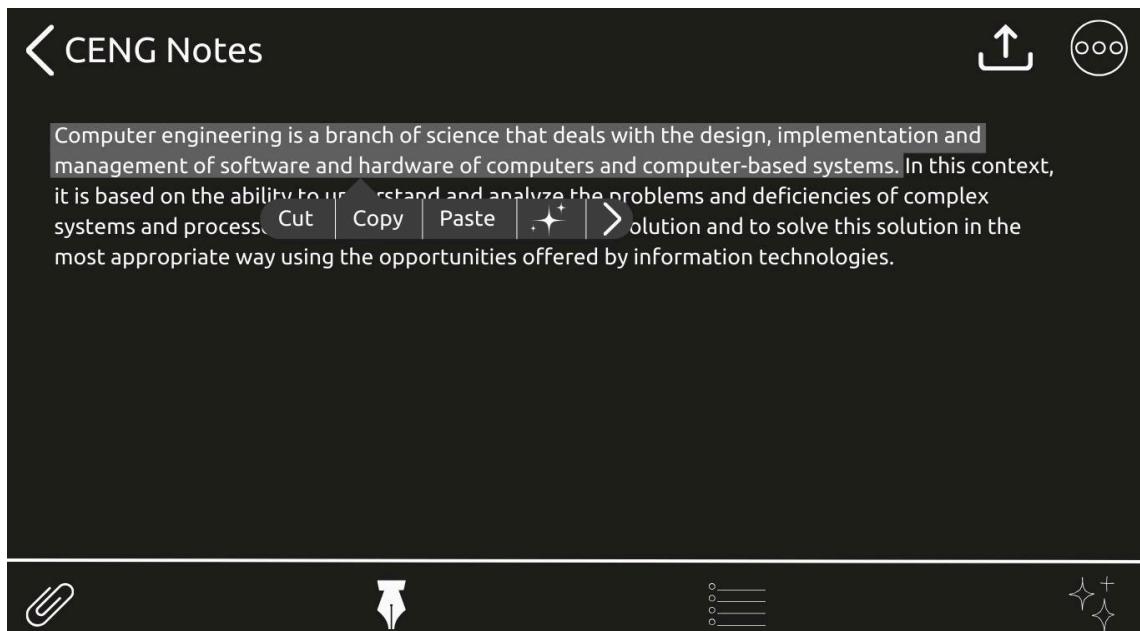
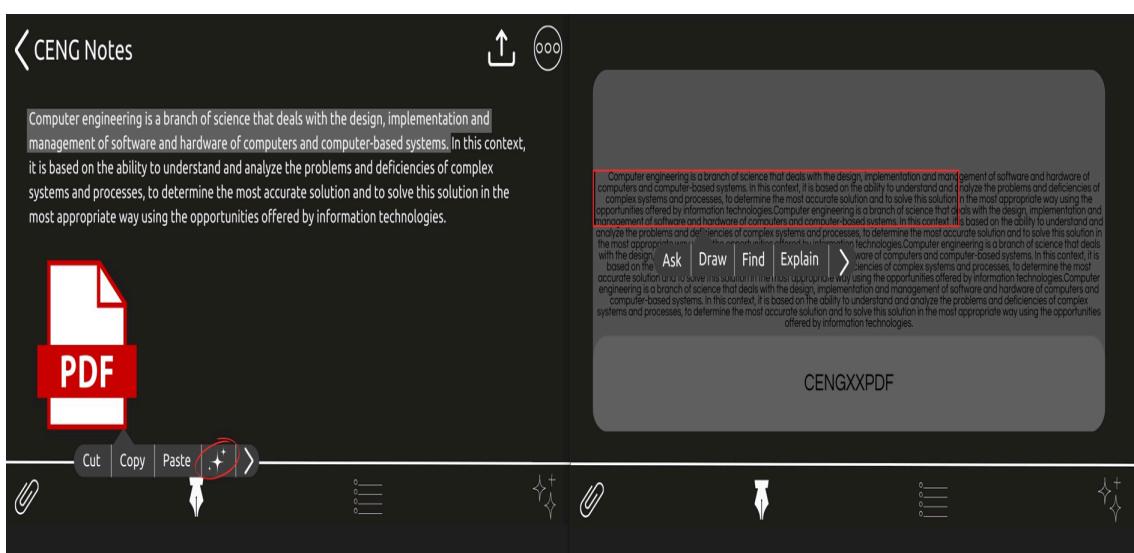


Figure 10 - Settings-theme selection



**Figure 11-Asking questions directly from notes**



**Figure 12- Asking questions about the document in the note**

#### 4.3.1.2 Hardware interfaces

This application is a mobile app and will be coded for Android. However, supported updates for IOS will be added later.

#### **4.3.1.2.1 Output Devices**

Display: SenseAI should be able to render the user interface on the mobile device's screen, utilizing the available screen resolution and aspect ratio.

#### **4.3.1.2.2 Network Connectivity**

NoteWiz will require access to the device's internet connection to communicate with external services.

#### **4.3.1.3. Software interfaces**

The app will interact with multiple third-party services and software interfaces, including:

- **AI APIs:** Integrated APIs for summarizing content, answering questions, creating visuals.
  1. A user interface that analyzes data from users and provides feedback and suggestions. This interface is used to integrate with AI APIs, such as answering questions and creating a cover image based on the content of the note. API interfaces enhance the user experience by extending the functionality of NoteWiz. They are necessary for client and server communication.
- **Cross-Platform Syncing:** The app will sync notes and tasks across devices using cloud services to ensure seamless user experience.
- **Authentication API:** For secure user sign-in via Google or email/password login.
- **Operating System (OS):** The app is built using the Flutter framework and relies on Android and iOS as primary operating systems. The minimum versions required for Android devices are Android 10 (API level 29) and iOS 13.0 for iOS devices. These versions ensure that the app has access to modern system APIs required for critical features.
- **Database Interface:** An interface where user data is stored and queried. This interface securely stores user interactions and other important information, and provides quick access when needed. The database interface is critical for managing and analyzing user data.
  1. Each note's metadata will include a generated cover image stored as part of the note's record. This image will be used to visually display the note in the application interface.
- **User Interface:** This is the interface through which users interact with NoteWiz. This interface takes various inputs from the user, such as text input. It will be developed with a user-friendly design, so that users can easily communicate with NoteWiz.

#### **4.3.1.4. Communications interfaces**

**NoteWiz** will use secure communication protocols to ensure the confidentiality and integrity of user data. Key communication features include:

- **Real-Time Syncing:** Firebase will handle real-time updates, ensuring data such as notes, tasks, and projects are synced immediately across devices.

- **RESTful APIs:** For communication between the app and backend servers, facilitating efficient data exchange.
- **User Authentication:** **Google Authentication API** will be used for secure login and account management.
- **Third-Party Integrations:** The app will integrate with external APIs to enhance functionality.

The device should have access to a stable internet connection, Wi-Fi, or mobile data, to use all the functionalities.

#### **4.3.2 Functional Requirements**

This section is explained in detail in section 4.2.5 Functional Requirements.

#### **4.3.3 Software System attributes**

##### **4.3.3.1. Portability**

**NoteWiz** will be designed to operate seamlessly across various devices and operating systems.

- **Optimization for App Stores:** The system will be optimized to meet Google Play Store requirements.
- **Responsive Design:** Compatibility with various screen sizes and resolutions will be ensured to deliver a consistent user experience on phones, tablets, and other devices.
- **Testing Across Devices:** Comprehensive testing will be conducted to ensure the application works efficiently on different hardware configurations and operating systems.

##### **4.3.3.2. Usability**

The user interface should be intuitive and easy for users to learn to use. The design should be simple and clear, with easy-to-navigate menus and options.

##### **4.3.3.3. Adaptability**

The app should be able to integrate new AI models or third-party services in future updates without major overhauls to the existing codebase.

##### **4.3.3.4. Security**

User data security and privacy are top priorities for **NoteWiz**. The system will implement advanced measures to protect sensitive user data against both accidental and malicious threats.

- **Two-Factor Authentication:** Users will have the option to enable two-factor authentication for enhanced account security.

- Role-Based Access Control: Access to sensitive user data will be restricted to authorized personnel only.
- Regular verification processes to ensure data integrity.

#### 4.3.3.5 Reliability

**NoteWiz** is a productivity application designed to enhance note-taking and collaboration. As such, reliability is a critical aspect to ensure consistent user satisfaction. The system must maintain sustainable service without losing user data, including notes, tasks, and collaborative projects. To achieve this:

- **Regular Backups:** Automatic and scheduled data backup mechanisms will be implemented to prevent data loss.
- **Error Recovery:** In the event of system errors, automatic recovery processes will ensure users can continue their activities without significant interruption.

#### 4.3.3.6 Availability

The system should be available 24/7 and include recovery and restart mechanisms for disaster recovery situations. High availability (HA) infrastructure will be used to minimize user waiting time.

#### 4.3.3.7 Maintainability

- **User Experience Preservation:** Updates will be designed to avoid disrupting ongoing user activities, ensuring data integrity and a seamless experience.

### 4.3.4 Safety Requirements

- **Data Security:**  
All user data must be encrypted during transmission and storage. The app must comply with data privacy laws and regulations, including GDPR and CCPA.
- **Privacy:**  
The app should allow users to delete their data upon request, and it must not store sensitive data for longer than necessary. User conversations will be deleted after a specified period.

## **5 Software Design Description**

### **5.1 Introduction**

#### **5.1.1 Purpose**

The purpose of this Software Design Document is to provide details of the project titled “NoteWiz: AI-Powered Note-Taking and Productivity App”.

The target audience is professionals, students, and individuals looking for advanced tools to take notes and increase productivity. NoteWiz is designed to revolutionize the note-taking experience by combining the latest AI technologies, offering note sharing and cross-platform accessibility.

The aim of the NoteWiz project is to create a comprehensive application that integrates features such as creating and editing notes, planning tasks efficiently, and interacting with an AI assistant. NoteWiz allows users to seamlessly manage their workflow by syncing data across devices. It also allows users to share notes in real-time. NoteWiz uses AI to assist users with summarization, personalized content creation, and automated task management, while providing a smooth and distraction-free interface.

The app is designed to include features such as AI-powered note management, task reminders, and visual note displays for intuitive organization. Participants can interact with NoteWiz via user-friendly interfaces on mobile devices.

Integration of cutting-edge APIs ensures robust AI interactions for note management, while cloud-based synchronization facilitates seamless cross-platform usage. Security features like encryption and two-factor authentication protect user data and privacy.

NoteWiz also includes customization options like light and dark themes to cater to user preferences. The app can be accessed using a variety of devices, making it versatile and adaptable for various user needs. The diagrams and technical details in this document provide more information about the app's functionality, system architecture, and user flow.

#### **5.1.2 Scope**

This document provides a complete explanation of the design and functionality of NoteWiz: AI-Powered Note-Taking and Productivity App.

The development of NoteWiz leverages modern tools and frameworks to deliver an intuitive, feature-rich app for enhanced productivity and collaboration. The React framework was chosen because it enables the app to run smoothly on Android devices. React's support for creating responsive UIs ensures an optimal experience across screen sizes and resolutions.

The backend of NoteWiz uses Firebase for real-time data synchronization, user authentication, and cloud storage. Firebase's reliable infrastructure ensures secure and

efficient data processing, while the real-time database provides instant cross-device updates for tasks, notes, and collaboration projects.

NoteWiz incorporates advanced AI APIs for functions such as smart summarization, personalized content creation, cover art generation, and AI-powered question answering. These APIs allow the app to process user inputs and provide meaningful, context-aware suggestions. Additionally, the system integrates third-party tools to enable scalable and efficient AI interactions.

For visual and functional consistency, Material Design principles are applied throughout the interface. This design approach provides users with a clean, minimalist layout and an easy-to-navigate environment.

The app also supports document uploading and processing, allowing users to extract and summarize information from various file formats, such as PDF and Word documents. The AI assistant makes it easy to seamlessly integrate this content into notes or tasks.

To ensure security and privacy, NoteWiz implements industry-standard encryption protocols, two-factor authentication.

In summary, NoteWiz combines user-friendly design principles with cutting-edge AI technologies to deliver an all-in-one platform for note-taking, task management, and productivity enhancement, delivering a superior experience across multiple devices.

### 5.1.3 Glossary

Term	Definition
<b>Alum</b>	Graduate of Jacksonville State University Computing and Information Sciences Department
<b>BDE</b>	Borland Database Engine
<b>CI</b>	Configuration Item
<b>CIS</b>	Computing and Information Sciences Department
<b>Html</b>	Hyper text markup language
<b>IEEE</b>	Institute of Electrical and Electronic Engineers

<b>QA</b>	Quality assurance
<b>SCMP</b>	Software Configuration Management Plan
<b>SDD</b>	Software Design Document
<b>SEI</b>	Software Engineering Institute, Pittsburgh, Pa
<b>SQAP</b>	Software Quality Assurance Plan
<b>SRS</b>	Software Requirement Specification
<b>Web Site</b>	A place on the world wide web
<b>OS</b>	Operating System
<b>AI (Artificial Intelligence)</b>	A branch of computer science that aims to create machines capable of performing tasks that would normally require human intelligence, such as learning, reasoning, and problem-solving.
<b>NLP (Natural Language Processing)</b>	A field of AI that focuses on the interaction between computers and human language, enabling machines to understand, interpret, and generate human language.
<b>API (Application Programming Interface)</b>	A set of rules and protocols that allows one software application to interact with another, enabling the integration of third-party services and functionalities.

*Table 2 Glossary of SDD*

#### 5.1.4 Overview of document

The remaining sections and their contents are listed below.

**Section 2** is the Architectural Design, which defines the project development phase. It also includes the class diagram of the system and the architectural design of the simulation, which defines the actors, exceptions, basic sequences, priorities, preconditions, and postconditions. In addition, this section includes the activity diagram of the scenario generator.

**Section 3** is the Use Case Implementation. In this section, the block diagram of the system designed according to the use cases in the SRS document is shown and explained.

**Section 4** is about the Environment. In this section, we showed the sample frames of the environment from the prototype and explained the scenario.

## **5.2 Architecture design**

NoteWiz's architecture is designed to deliver a smooth and efficient user experience while ensuring scalability, security, and reliability. It is structured into three main layers: front-end, back-end, and database.

### **5.2.1 Front-end layer**

The front-end layer uses the React framework to create a responsive and visually appealing user interface. It ensures compatibility across devices such as smartphones. This layer facilitates user interactions such as note creation, task planning, AI-powered question answering, and document processing. The user interface is designed for simplicity and accessibility by offering intuitive navigation and customization options such as light and dark themes.

### **5.2.2 Back-end Layer**

The back-end layer serves as the system's core, handling user requests, executing business logic, and managing secure interactions with the database. Firebase services are utilized for real-time synchronization, ensuring seamless cross-device data consistency. The back-end also includes AI integrations that process user inputs for features like summarization, question answering, and task recommendations. This layer is designed to handle high traffic and large data volumes efficiently, ensuring consistent performance and scalability.

### **5.2.3 Database Layer**

The database layer securely stores core data such as user profiles, notes, tasks, and document metadata. Firebase's real-time database ensures fast and reliable data retrieval and multi-device syncing. Security measures, including data encryption and controlled access protocols, safeguard user information and privacy.[3]

### **5.2.4 Integration and Modularity**

APIs connect the three layers, ensuring seamless communication and enabling modularity. This design supports future feature enhancements, such as new AI capabilities or additional integrations, without requiring significant architectural changes. The architecture ensures that NoteWiz remains a reliable and scalable productivity solution as user needs evolve.

### 5.3 Data structure design

This section outlines the data structures designed to manage user information within the Notewiz application. The data structure has been carefully crafted to ensure scalability, security, and usability while supporting a user-centric experience.

The design of the data structure facilitates core functionalities such as authentication, profile management, security features, user preferences, and subscription management. It has also been optimized for future expansions, data backups, and security compliance to ensure robust system performance and reliability.

The table below provides a detailed overview of the primary data fields representing user information within the application. Each field is described with its name, data type, and maximum size. This design ensures data consistency, system scalability, and secure handling of user information.

Attribute Name	Attribute Type	Attribute Size
UserID	String	20
Username	String	30
EmailAddress	String	50
Password	String (hashed)	64
CreatedAt	DateTime	-
UpdatedAt	DateTime	-
ProfilePicture	String (URL)	255
Bio	String	150
IsActive	Boolean	1
Roles	String	50
LastLogin	DateTime	-
PhoneNumber	String	15
TwoFactorEnabled	Boolean	1
DateOfBirth	DateTime	-
SubscriptionType	String	20
LoginAttempts	Int	-
Preferences	JSON	-
DeletedAt	DateTime	-
DeviceInfo	String	255
LastPasswordChange	DateTime	-
APIKeys	String	255

## 5.4 System Design

### 5.4.1 UML Diagram

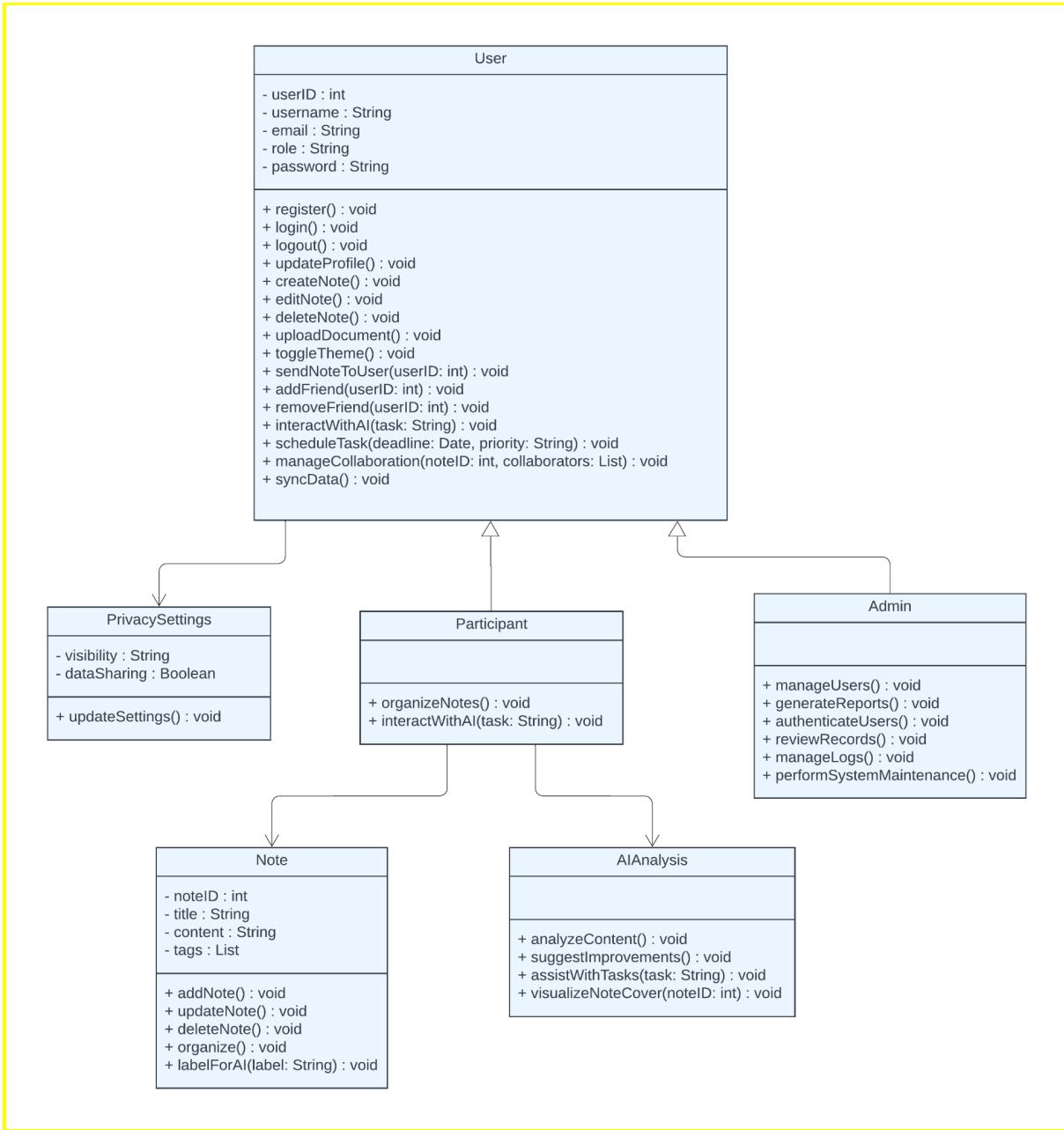


Figure13- System Design

### 5.4.2 Decomposition Description

For each module provide a flowchart or activity diagram and describe the flow of the module.

### **5.4.2.1 Database Layer**

**Summary:** This activity diagram outlines the process for managing notes in the system, including user actions and system responses.

**Actors:**

- User

**Precondition:**

- User must be logged into the system.

**Basic Sequence:**

**Start:**

- The user logs into the system.

**Create a Note:**

- The user creates a new note.
- The system saves the note to the database.

**Edit a Note:**

- If the user edits the note, the system updates it in the database.
- **Completion Check:**
- The system ensures all notes are up-to-date.

**Exception:**

- Database Connection Failure:
  - If the database connection is unstable, the system displays an error message: "Connection issues."
  - The system analyzes database integrity and prompts the user to retry.

**Outcome:**

- Notes are securely stored, updated, or managed in the system database.

**Post Conditions:**

- All notes are up-to-date and stored in the database.

**Priority:**

- High

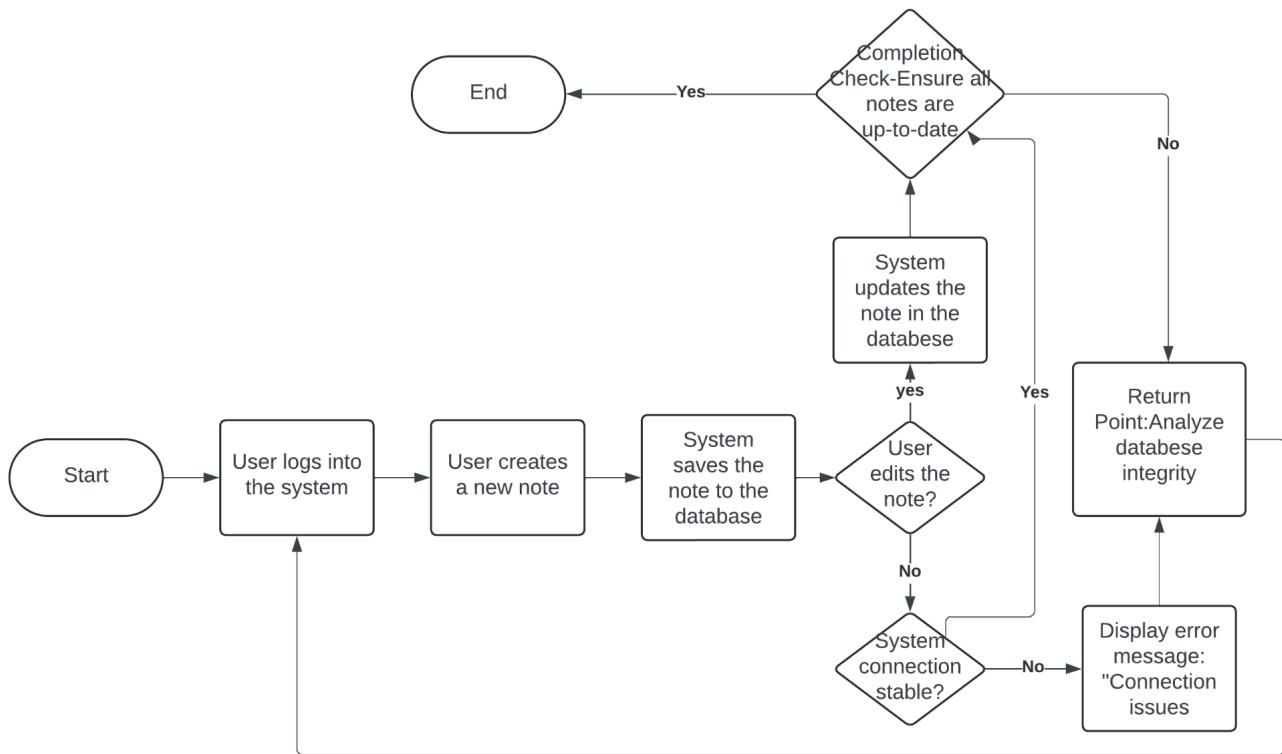


Figure-14 Database Layer

#### 5.4.2 2 User Interface Logic and Workflows

##### 5.4.2.2.1 Sign up and Log in

**Summary:** This activity diagram outlines the steps for a user to sign up and log in to the system. It includes actions for both participants and system responses.

##### Actors:

-Participant: The user interacting with the system.

##### Precondition:

-The user must have the program running.

##### Basic Sequence:

- Start:

- The user starts the program.
- **Sign Up:**
  - The user selects the "Sign Up" option.
  - The system displays the registration form.
  - The user enters required details (e.g., username, password, email).
  - The system validates the inputs.
  - If inputs are valid, the system saves the user's data to the database.
  - A success message is displayed to the user.
- **Login:**
  - The user selects the "Login" option.
  - The system displays the login form.
  - The user enters their username and password.
  - The system validates the credentials against the database.
  - If credentials are valid, the user is granted access to the system.
  - If invalid, an error message is displayed.
- **Exit:**
  - The user can choose to exit the system at any time by selecting the "Exit" option.
  - The system logs the user out and terminates the session.

**Exception:**

**Database Connection Failure:**

- If the database connection fails during sign-up or login, an error message is displayed.
- The user is prompted to try again later.

**Post Conditions:**

- The user's account is successfully created and stored in the database (if signing up).
- The user is logged into the system (if logging in).

**Priority:**

-High

## ACTIVITY DIAGRAM FOR USER SIGN UP AND LOGIN

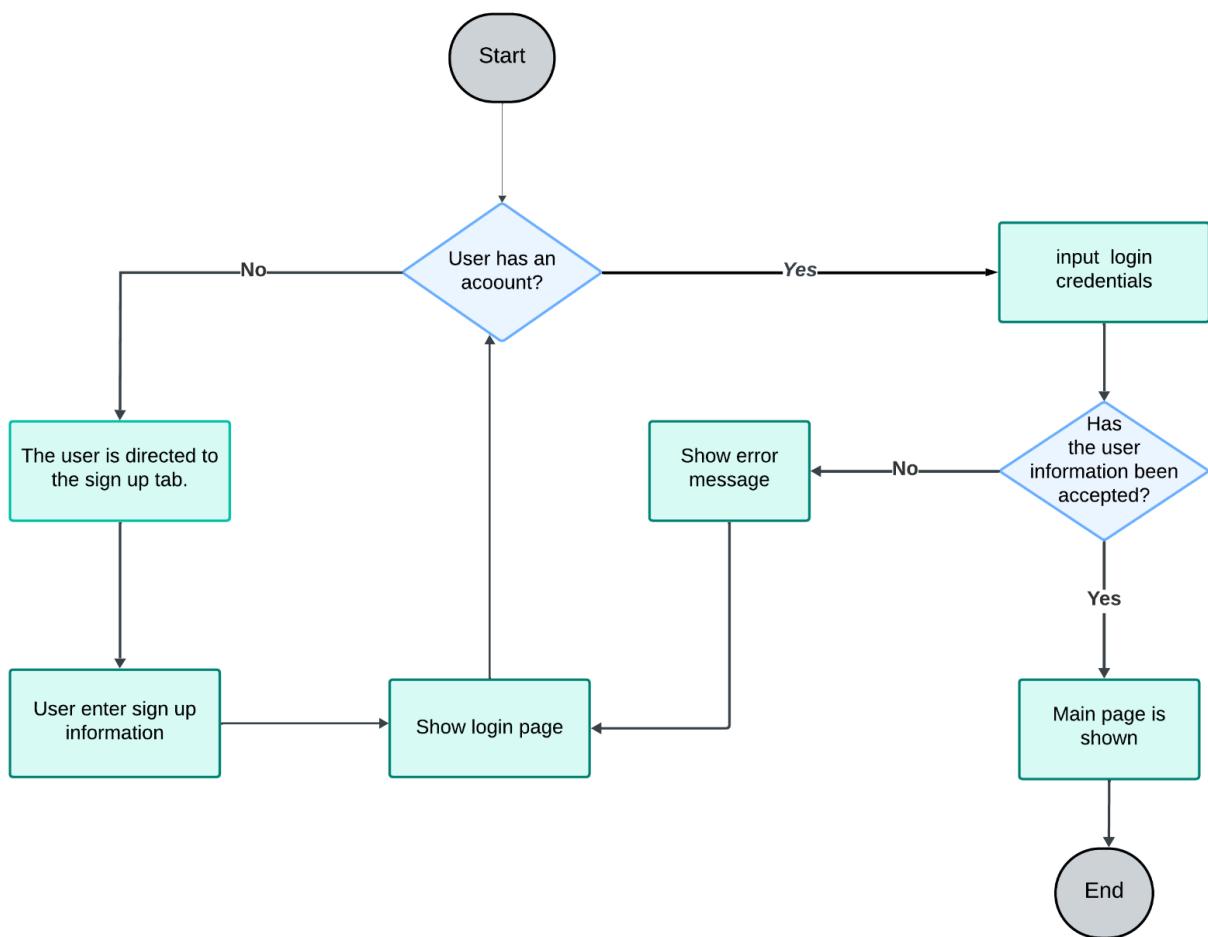


Figure-15 Sign Up and Login

#### **5.4.2.2.2 Profile Management**

**Summary:** This system is used by participant and admin.

Participant can login, register, and update personal information and exit from the system. In addition to this, admin can delete an account, approve participant accounts and add a new admin.

**Actor:** Participant, admin **Precondition:** User must run the program.

**Basic Sequence:**

- User must register if s/he doesn't have an account.
- User shall login to the system by entering his/her username and password.
- User can update his/her personal information by selecting the update button from the user menu.
- Admin can delete a user account by selecting the delete button from the admin menu.
- Admin can approve a user account which is registered recently by selecting approve from admin menu.
- Admin can add a new admin by selecting add new admin button from admin menu.
- User can exit from the system by selecting the exit button.

**Exception:** Database connection can be failed.

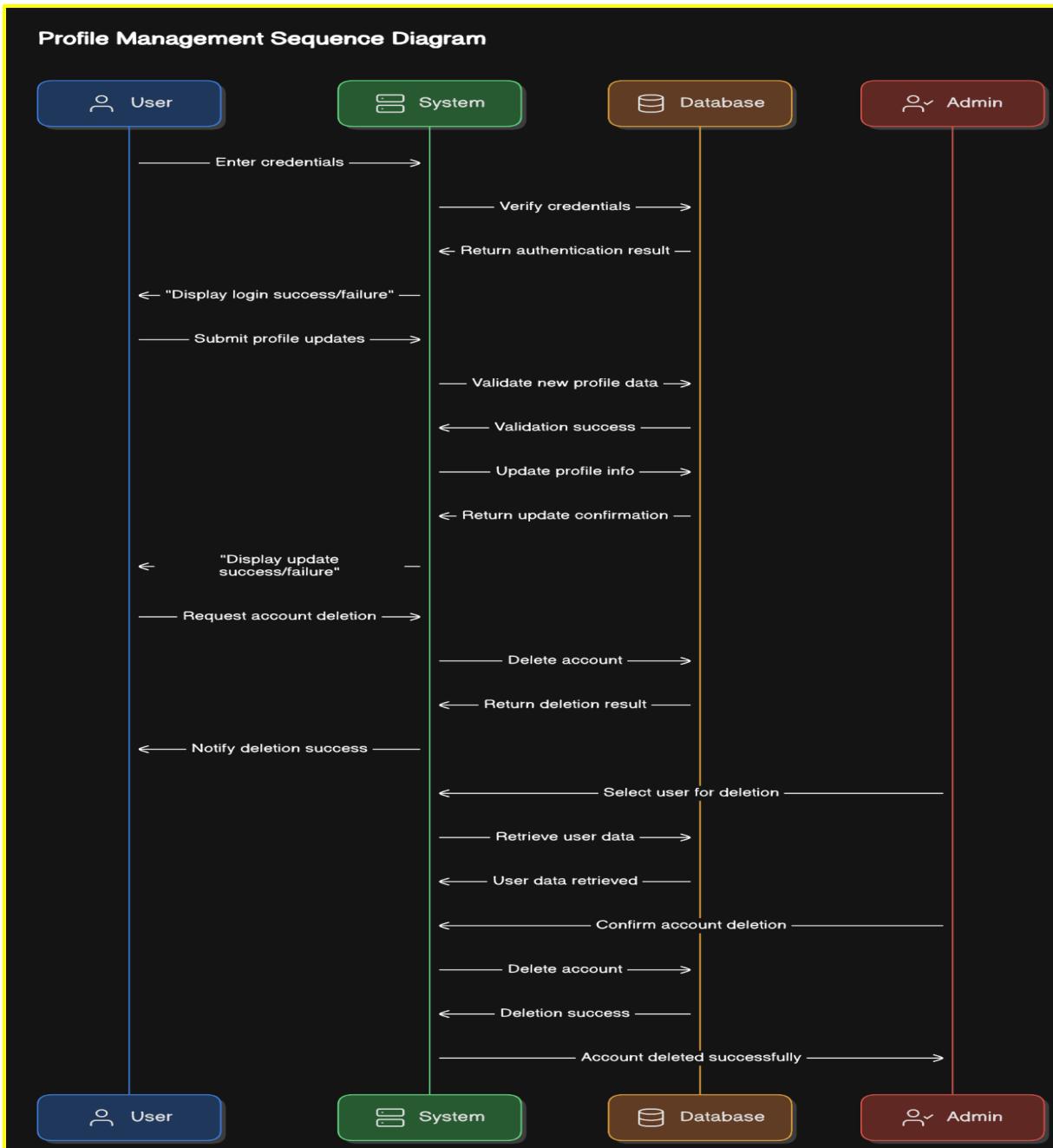


Figure-16 Profile Management

#### 5.4.2.2.3 Main Menu

##### Summary:

This activity diagram outlines the interactions available in the main menu for participants and admins. The menu provides access to core functionalities of the system.

## **Actors:**

- **Participant:** A regular user of the system.
- **Admin:** A user with elevated privileges.

## **Precondition:**

- The user must be logged into the system.

## **Basic Sequence:**

- **Start:**
  - The user accesses the main menu after logging in.
- **Participant Options:**
  - The participant can select from the following options:
    - "Create Note": Opens the note creation interface.
    - "View Notes": Displays a list of saved notes.
    - "Update Profile": Opens the profile management screen.
    - "Exit": Logs the participant out of the system.
    - "Share Notes" can observe the note sharing screen.
- **Admin Options:**
  - The admin can select from the following additional options:
    - "Approve Accounts": Displays pending account registrations for approval.
    - "Delete Account": Allows the admin to remove a user account.
    - "Add New Admin": Opens the interface to register a new admin account.
- **Exit:**
  - The user selects the "Exit" option to log out and return to the login screen.

## **Exception:**

- **Menu Load Failure:**
  - If the main menu fails to load, an error message is displayed, and the user is prompted to restart the application.

## **Post Conditions:**

- The selected action is successfully completed, or the user exits the system.

## **Priority:**

- Medium

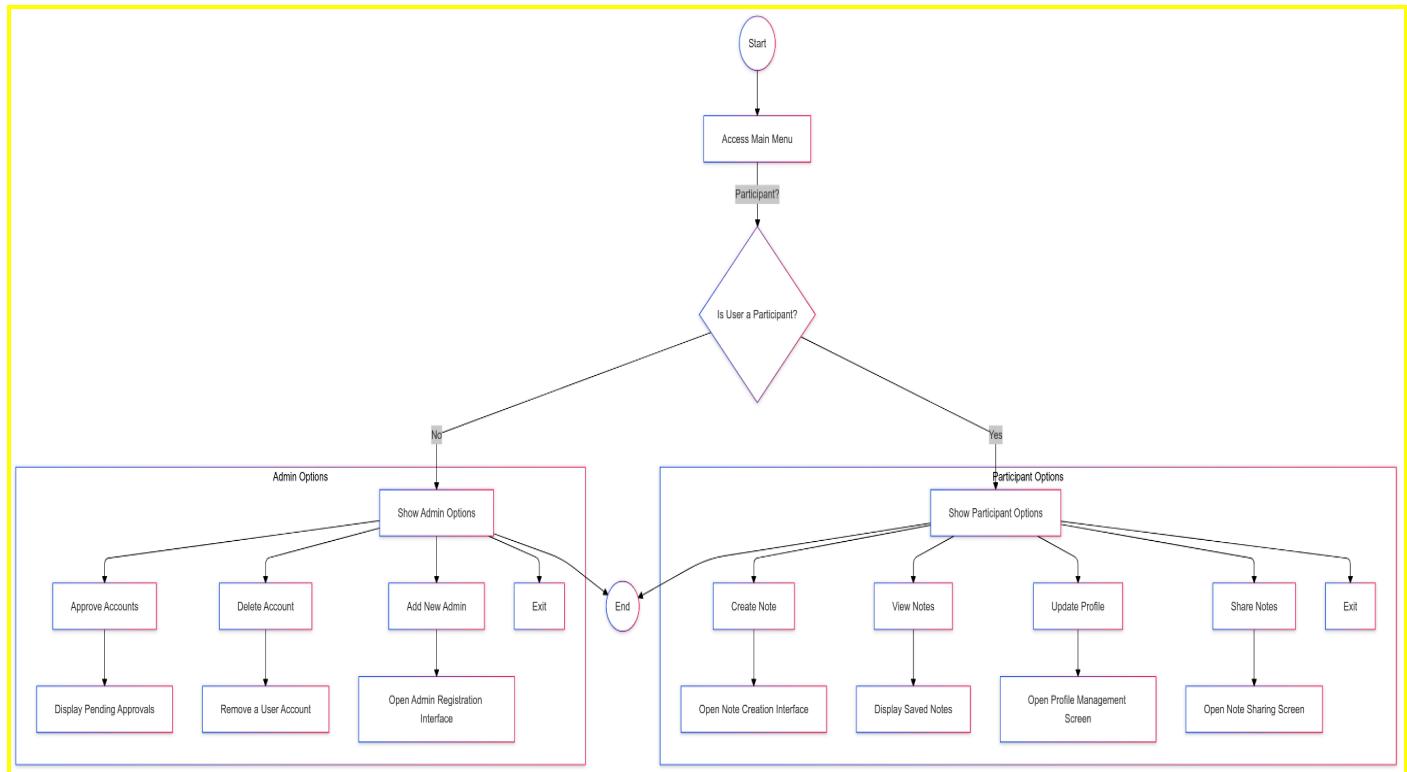


Figure-17 Main Menu

#### 5.4.2.2.4 Note Creation and Management

##### 5.4.2.2.4.1 Note Creation and Management

This diagram describes the note creation process, allowing users to create, edit, and manage notes with AI-assisted features. The system supports rich content creation including media attachments and AI-powered interactions.

**Actors:** User: A regular user of the system creating and managing notes. System: The application handling note creation and AI interactions.

**Precondition:** The user must be logged into the system and have accessed the note creation interface.

##### Basic Sequence:

- **Start:** The user initiates note creation from the main menu.
- **Note Creation Process:**
  - "Enter Title": Provides a name for the note.
  - "Add Content": Types or pastes the main note content.
  - "Add Attachments": Includes images, links, or other media.
  - "AI Interactions": Asks questions about note content and receives AI assistance.

- **AI Integration Features:**
  - "Generate Cover Image": Creates a relevant cover image based on note content.
  - "Answer Questions": Responds to user queries about note content.
  - "Add AI Responses": Incorporates AI answers as pop-up items or inline text.
- **Note Management:**
  - "Save Note": Stores the note in the user's account.
  - "Share Note": Distributes the note to selected recipients.
  - "Edit Note": Modifies existing note content.
  - "Delete Note": Removes the note with confirmation.
- **Exit:**
  - The user completes note creation and returns to the note list view.

**Exceptions:** Save Failure: If note saving fails, an error message appears, and the system preserves the content for retry.

**AI Service Unavailability:** If AI features are temporarily unavailable, the system continues with basic note functionality.

**Post Conditions:** The note is successfully created and stored in the user's account, or the process is cancelled.

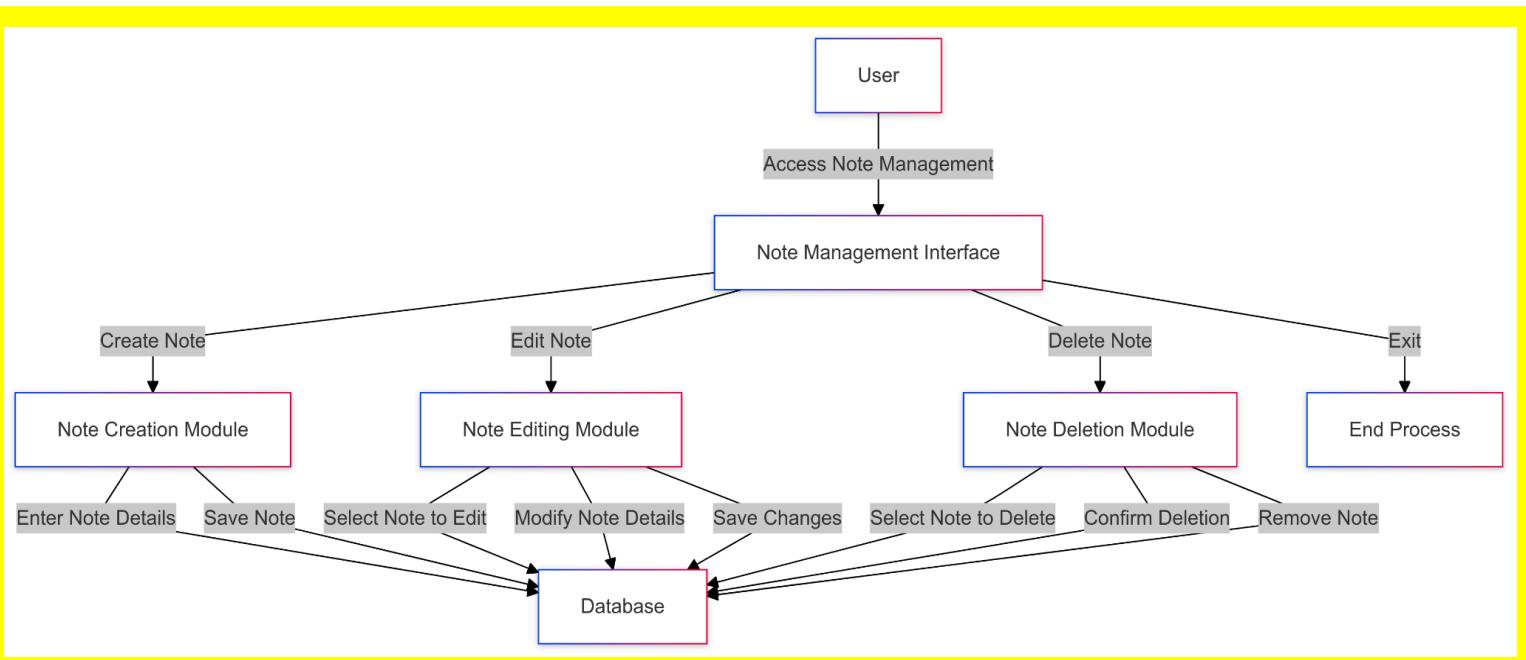


Figure-17 Note Creation and Management

#### **5.4.2.2.4.2 Note Visualization**

**Summary:** This diagram outlines the automatic cover generation process for notes, enhancing visual organization and note identification through AI-generated imagery and visual elements.

**Actors:** User: The note creator

**System:** The application's cover generation and display components

**Precondition:**

- A note must be created or in the process of being created
- The system's AI image generation service must be available

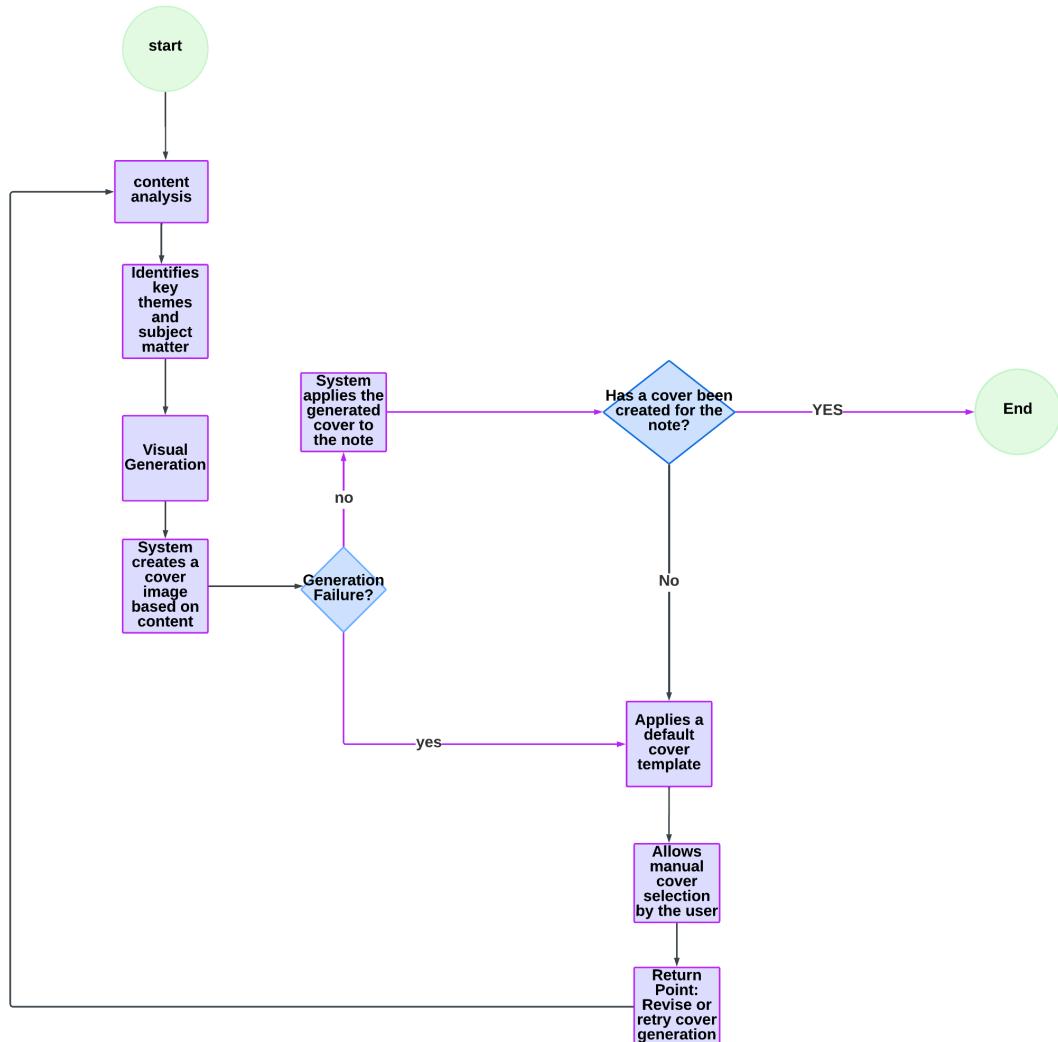
**Basic Sequence:**

- **Start:** The process begins when a new note is created or saved.
- **Cover Generation Process:**
  - Content Analysis
  - System analyzes note title, content, and metadata
  - Identifies key themes and subject matter
  - Determines appropriate visual elements
  - Visual Generation
  - System creates a unique cover image based on:
    - Note content analysis
    - User-added tags or categories
  - Display Integration
  - System applies the generated cover to the note
- **Exit:** The cover is finalized and associated with the note in the system.

**Exceptions:**

**Generation Failure:** If cover generation fails, the system:

- Applies a default cover template
- Logs the failure for system monitoring
- Allows manual cover selection by the user



*Figure-18 Note Visualization*

#### 5.4.2.2.5 Note Sharing

**Summary:** This diagram outlines the process of sharing notes with other users, enabling collaboration and efficient knowledge exchange.

##### Actors:

- User: The individual sharing the note
- System: The application's note sharing service

##### Precondition:

- The user must be logged into the system.
- The note to be shared must exist in the user's account.

## Basic Sequence:

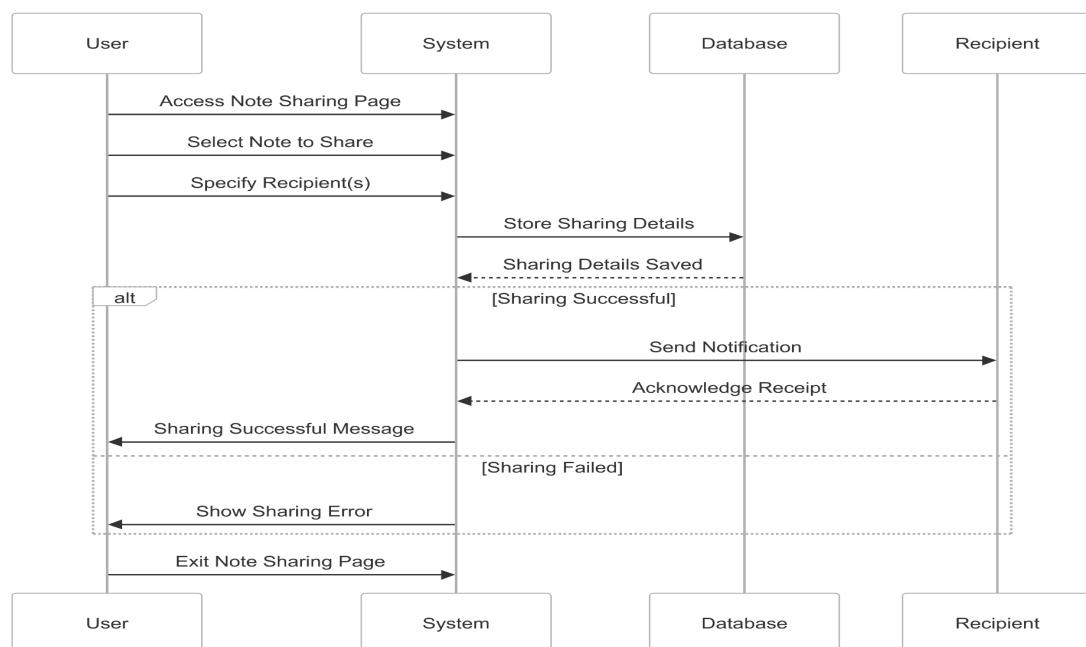
- **Start:** The user selects a note to share from their note list.
- **Share Options:** The system provides sharing options:
  - Share via email
  - Generate a sharable link
  - Share to a group or specific users within the system
- **Select Recipients:** The user specifies the recipients or sharing method.
- **Permission Setting:**
  - The user sets permissions (e.g., view-only, edit, comment).
- **Confirmation:** The system confirms the note-sharing action.
- **Notification:**
  - Recipients receive a notification or email with the shared note link.
- **Exit:** The process concludes, and the user returns to the note management page.

## Exceptions:

- **Sharing Failure:** If sharing fails due to network or system issues:
  - The system displays an error message.
  - The user can retry or cancel the sharing process.

## Post Conditions:

- The note is successfully shared with the specified recipients.
- Recipients can access the note based on the permissions set by the user.



*Figure-19 Note Sharing*

#### **5.4.2.2.6 Document Upload and Processing**

**Summary:** This diagram describes the process of uploading and processing documents to extract and integrate content into notes.

##### **Actors:**

- User: The individual uploading the document
- System: The application's document processing component

##### **Precondition:**

- The user must be logged into the system.
- The document to be uploaded must be accessible from the user's device.

##### **Basic Sequence:**

- **Start:** The user selects the "Upload Document" option.
- **File Selection:**
  - The user browses and selects a document (e.g., PDF, Word file).
  - The system validates the file format.
- **Content Extraction:**
  - The system processes the document to extract text, images, and metadata.
  - Extracted content is displayed for user review.
- **Integration:**
  - The user chooses how to integrate the content (e.g., as a new note, append to an existing note).
  - The system processes the integration.
- **Confirmation:** The system confirms successful document upload and integration.
- **Exit:** The user returns to the document management interface.

##### **Exceptions:**

- **Invalid File Format:**
  - The system rejects unsupported file formats and displays an error message.
- **Processing Failure:**
  - The system logs the failure and prompts the user to retry.

##### **Post Conditions:**

- The document content is successfully extracted and integrated into the note.

- The processed content is available in the user's note list.

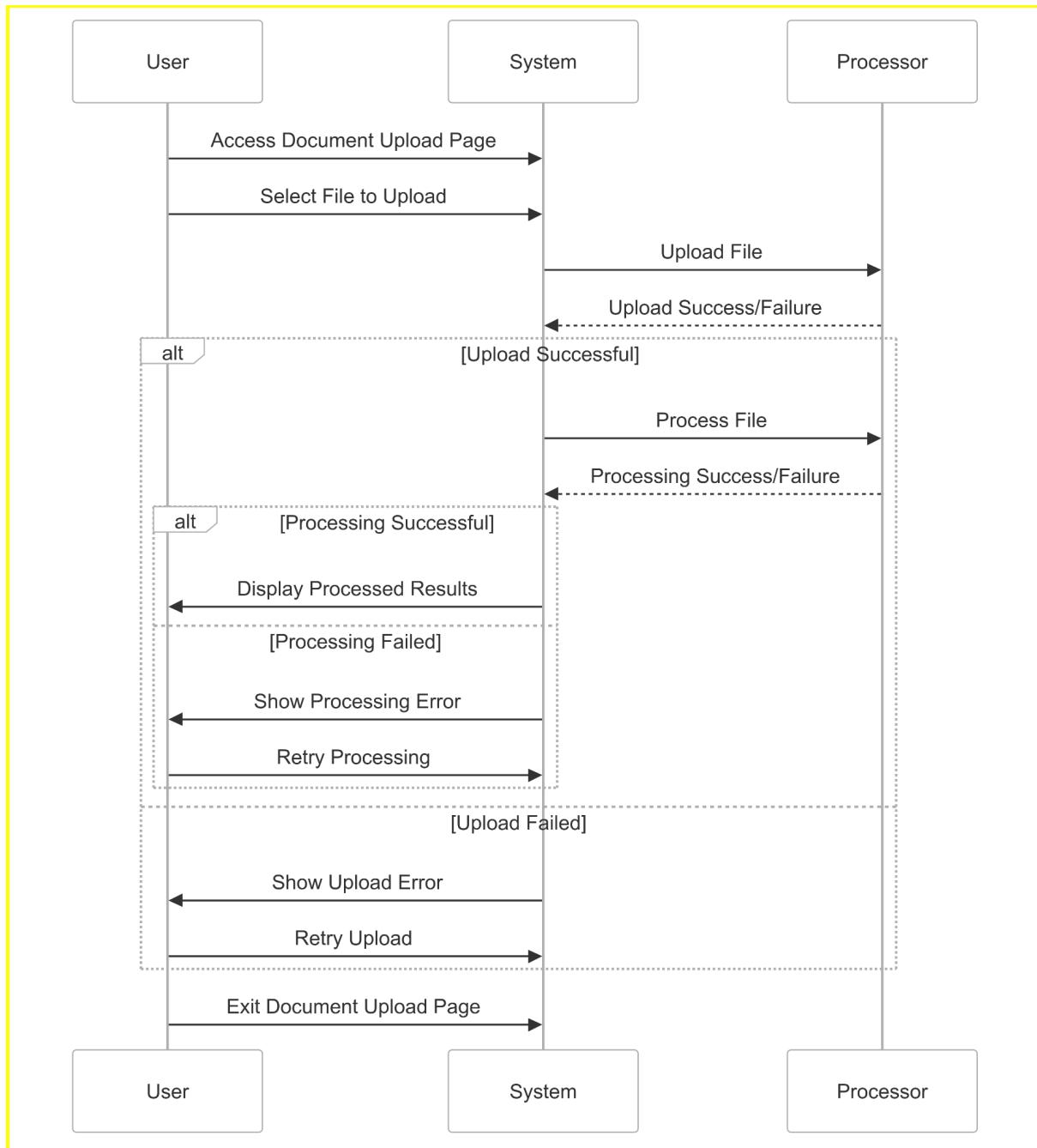


Figure-20 Document Upload and Processing

#### **5.4.2.2.7 AI Chat Page (AI-powered question asking)**

**Summary:** This diagram illustrates the interaction between users and the AI chat feature for asking questions and receiving context-aware answers.

##### **Actors:**

- User: The individual interacting with the AI chat
- System: The AI-powered question-answering module

##### **Precondition:**

- The user must be logged into the system.
- The AI service must be active and operational.

##### **Basic Sequence:**

- **Start:** The user navigates to the AI chat page.
- **Question Input:** The user enters a question into the chat interface.
- **Processing:**
  - The system analyzes the question using NLP algorithms.
  - Relevant context from the user's notes is retrieved, if applicable.
- **Response Generation:**
  - The AI generates an answer based on the question and available context.
- **Display Response:**
  - The system displays the AI's response to the user.
- **Follow-up Interaction:**
  - The user can ask follow-up questions or clarify the query.
  - If the user approves the answer, it can be added as a pop-up
- **Exit:** The user closes the chat or navigates to another feature.

##### **Exceptions:**

- **Service Unavailability:**
  - If the AI service is down, the system displays a fallback message and logs the issue.
- **Ambiguous Query:**
  - If the query is unclear, the system prompts the user for clarification.

##### **Post Conditions:**

- The user receives an accurate and relevant response to their question.
- The AI interaction log is updated for future enhancements.

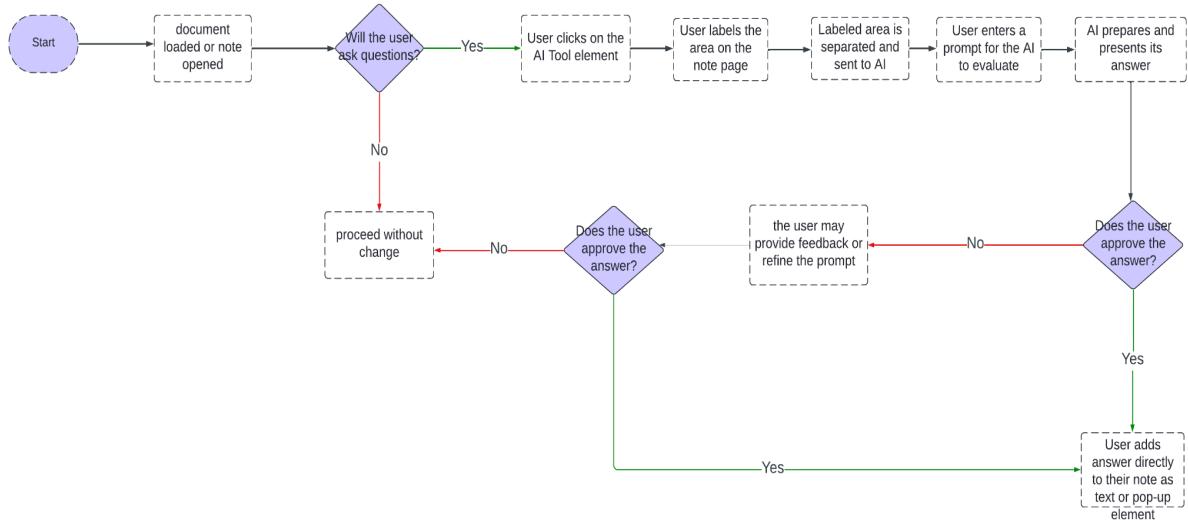


Figure-21 AI Chat Page(AI-powered question asking)

#### 5.4.2.2.8 Forgot Password Pages

**Summary:** This diagram outlines the process for users to reset their password if forgotten.

##### Actors:

- User: The individual requesting a password reset
- System: The password management module

##### Precondition:

- The user must have a registered email address in the system.

##### Basic Sequence:

- **Start:** The user selects the "Forgot Password" link on the login page.
- **Email Input:**
  - The user enters their registered email address.
  - The system validates the email format and checks its existence in the database.
- **Email Verification:**
  - The system sends a password reset link to the user's email.
- **Link Access:**
  - The user clicks the link and is redirected to the password reset page.
- **New Password Entry:**
  - The user enters and confirms a new password.
  - The system validates the password's strength and matches both entries.
- **Confirmation:**
  - The system updates the password and notifies the user of success.

- **Exit:** The user returns to the login page to access the system with the new password.

### Exceptions:

- **Invalid Email:**
  - The system rejects unregistered email addresses and displays an error.
- **Expired Link:**
  - If the reset link is expired, the system prompts the user to request a new one.

### Post Conditions:

- The user's password is successfully reset.
- The system ensures secure access with the updated credentials.

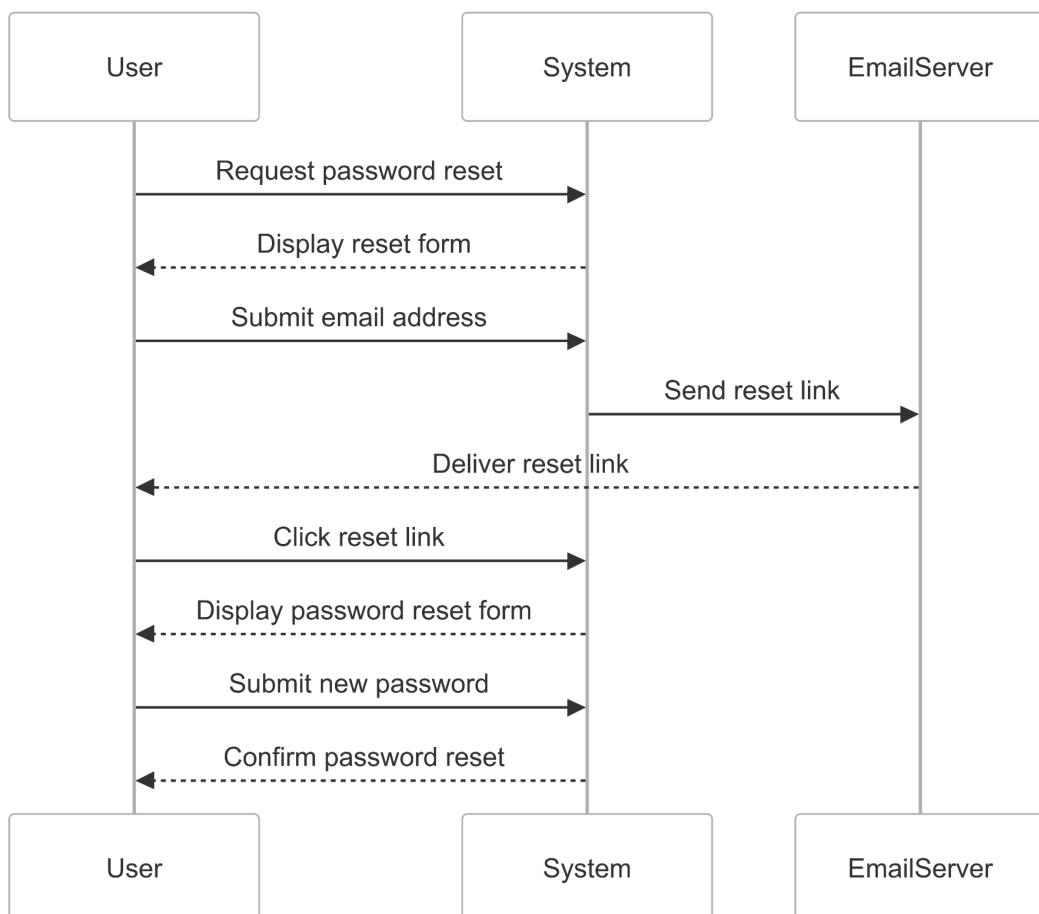


Figure 22 -Forgot Password

### **5.4.2.3 Admin Pages**

**Summary:** These diagrams describe the functionalities available to administrators for managing users and system configurations.

#### **5.4.2.3.1 User Record Page**

**Summary:** This diagram details the process for viewing and managing user records.

**Actors:**

- Admin: The individual managing user records
- System: The user management module

**Precondition:**

- The admin must be logged into the system with appropriate permissions.

**Basic Sequence:**

- **Start:** The admin navigates to the User Record page.
- **View Records:**
  - The system displays a list of registered users with details.
- **Search/Filter:**
  - The admin can search or filter user records based on criteria.
- **Action Options:**
  - The admin selects an action (e.g., view details, delete user, export data).
- **Confirmation:** The system processes the action and confirms completion.
- **Exit:** The admin logs out or navigates to another page.

**Exceptions:**

- **Access Denied:**
  - The system blocks unauthorized access attempts and logs the incident.

**Post Conditions:**

- User records are effectively managed with the selected actions.

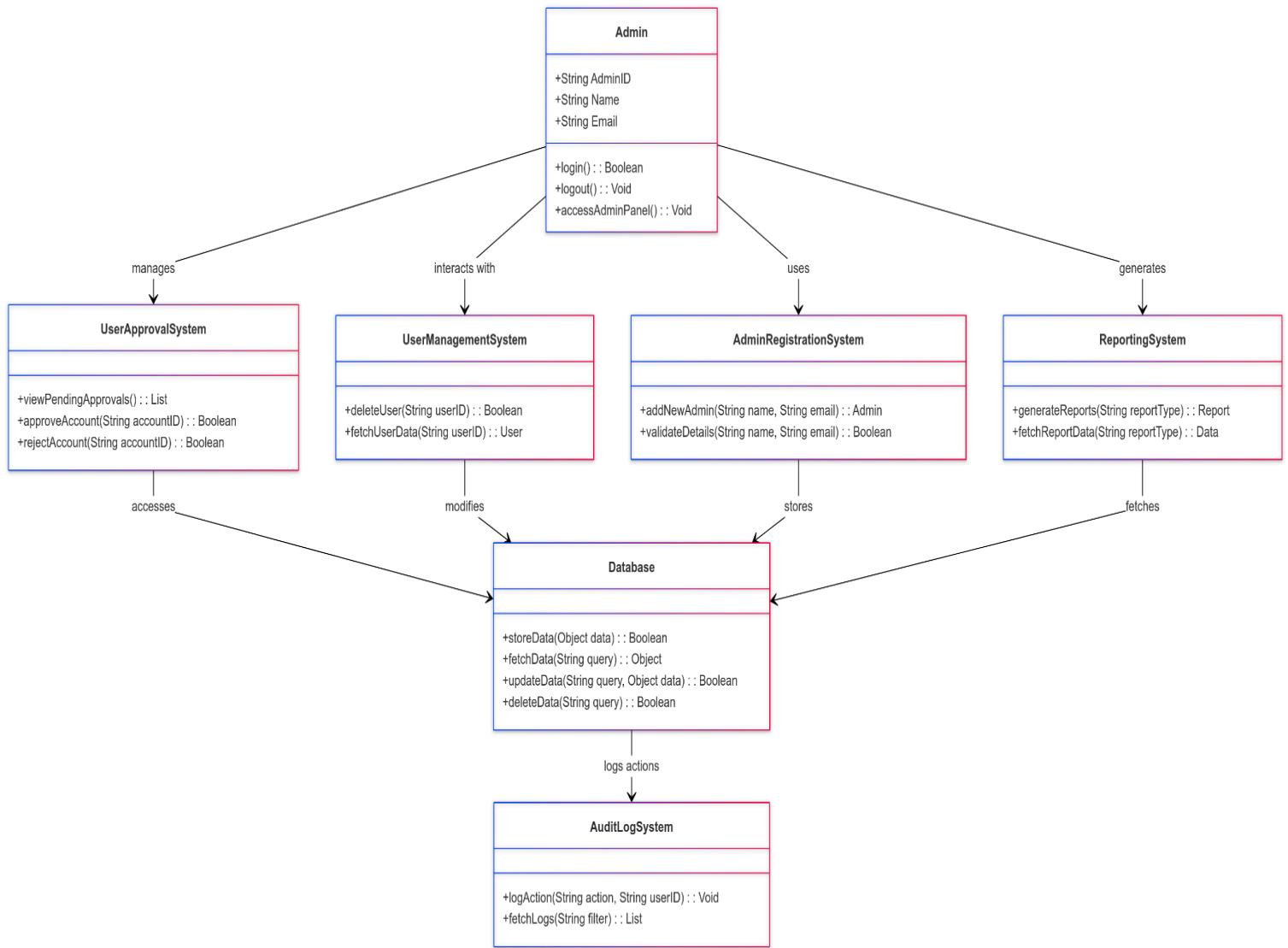


Figure-12 User Record Page

#### 5.4.2.3.1 User Record Page

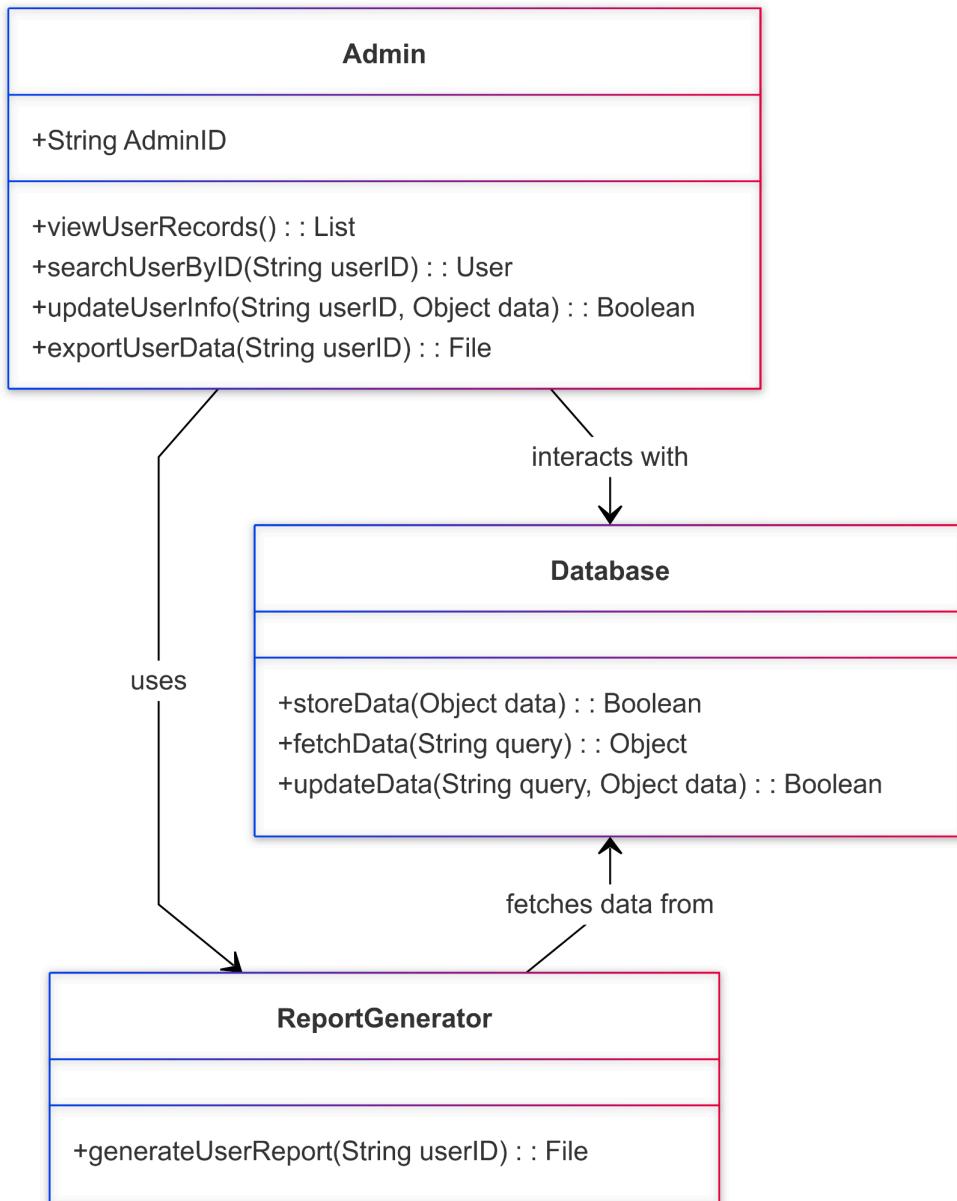


Figure-23 User Record Admin

**Summary:** This diagram details the process for viewing and managing user records.

**Actors:**

- **Admin:** The individual managing user records
- System: The user management module

**Precondition:**

- The admin must be logged into the system with appropriate permissions.

**Basic Sequence:**

- **Start:** The admin navigates to the User Record page.
- **View Records:**
  - The system displays a list of registered users with details.
- **Search/Filter:**
  - The admin can search or filter user records based on criteria.
- **Action Options:**
  - The admin selects an action (e.g., view details, delete user, export data).
- **Confirmation:** The system processes the action and confirms completion.
- **Exit:** The admin logs out or navigates to another page.

**Exceptions:**

- **Access Denied:**
  - The system blocks unauthorized access attempts and logs the incident.

**Post Conditions:**

- User records are effectively managed with the selected actions.

#### 5.4.2.3.2 User Access Control Page

**Summary:** This diagram outlines the access control management process, allowing system administrators to manage user roles, add/remove users, and control system access permissions.

**Actors:** The system administrator

**System:** The application's access control management components

**Precondition:**

- Administrator must be logged into the system
- Administrator must have sufficient privileges
- System access control service must be available

**Basic Sequence:**

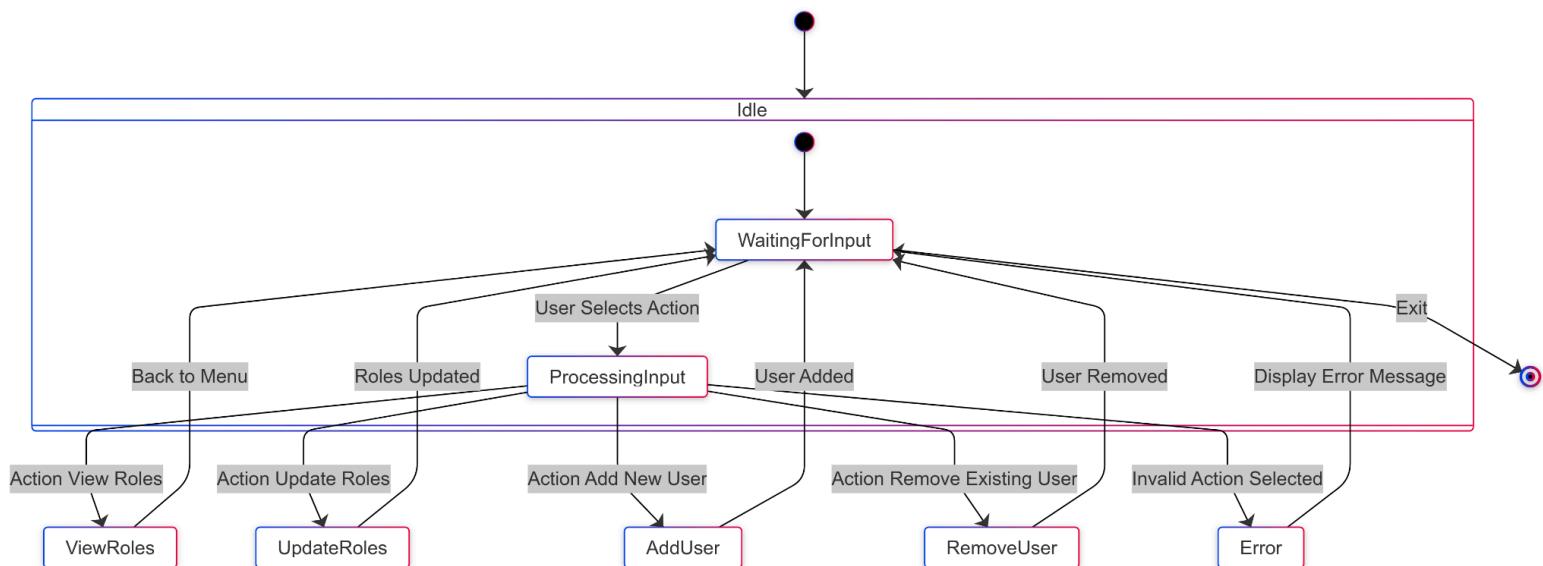
- **Start:** The process begins when an administrator accesses the User Access Control page.
- **Access Control Process**
  - **Role Management**

- System displays current user roles and permissions
- Administrator can view existing role assignments
- System allows role updates and modifications
- **User Management**
  - System enables adding new users to the system
  - Administrator can remove existing users
  - System processes user access changes
- **Action Processing**
  - System validates administrator actions
  - Updates are applied to user permissions
  - System confirms successful changes
- **Exit:** The changes are finalized and saved in the system.

Exceptions: Action Failure: If action processing fails, the system:

- Displays an error message
- Returns to waiting state
- Allows administrator to retry action or return to menu

Figure-14 User Access Control



#### 5.4.2.4 Settings Page

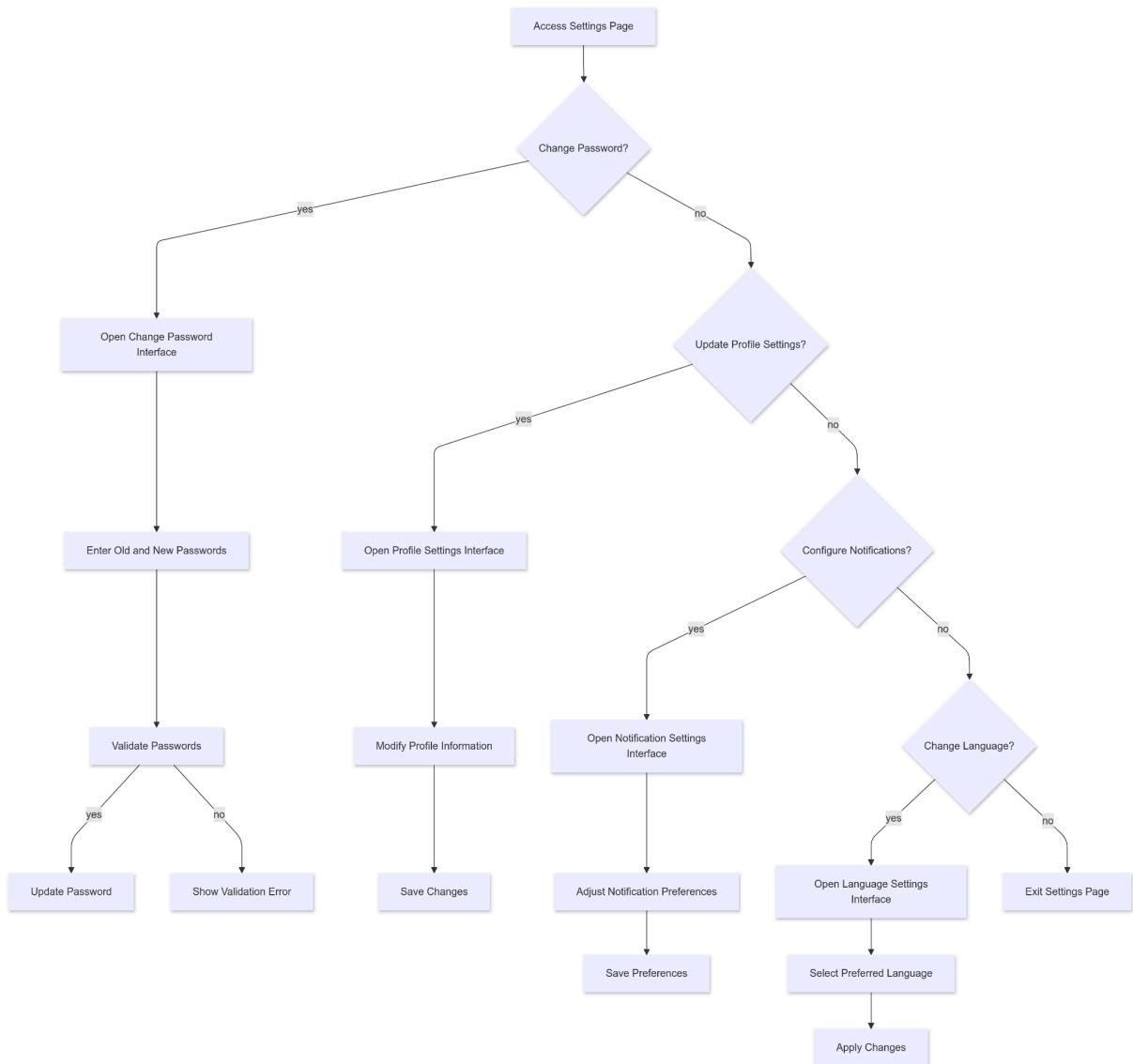


Figure-24 Settings Page

**Summary:** This diagram describes the process for managing user-specific settings and system configurations, enabling personalized and optimized application usage.

#### Actors:

- **User:** The individual modifying settings for their account.
- **System:** The application's settings management module.

#### Precondition:

- The user must be logged into the system.

### **Basic Sequence:**

- **Start:** The user navigates to the Settings page from the main menu.
- **View Settings:**
  - The system displays available settings categories (e.g., Account, Preferences, Notifications, Privacy).
- **Modify Settings:**
  - The user selects a category and makes desired changes.
  - Examples:
    - Account settings: Change username, email, or password.
    - Preferences: Switch between light and dark mode, language options.
    - Notifications: Enable or disable specific notifications.
    - Privacy: Adjust data sharing permissions or account visibility.
- **Save Changes:**
  - The system validates inputs and saves the changes.
- **Confirmation:** A success message is displayed to confirm that settings have been updated.
- **Exit:** The user exits the settings page and returns to the main menu.

### **Exceptions:**

- **Validation Error:**
  - If inputs are invalid, the system displays an error message and prompts the user to correct them.
- **Save Failure:**
  - If changes fail to save due to a network issue, the system notifies the user and logs the error.

### **Post Conditions:**

- User settings are successfully updated and stored in the system.

## 5.5 Interface design

### 5.5.1 Login Page

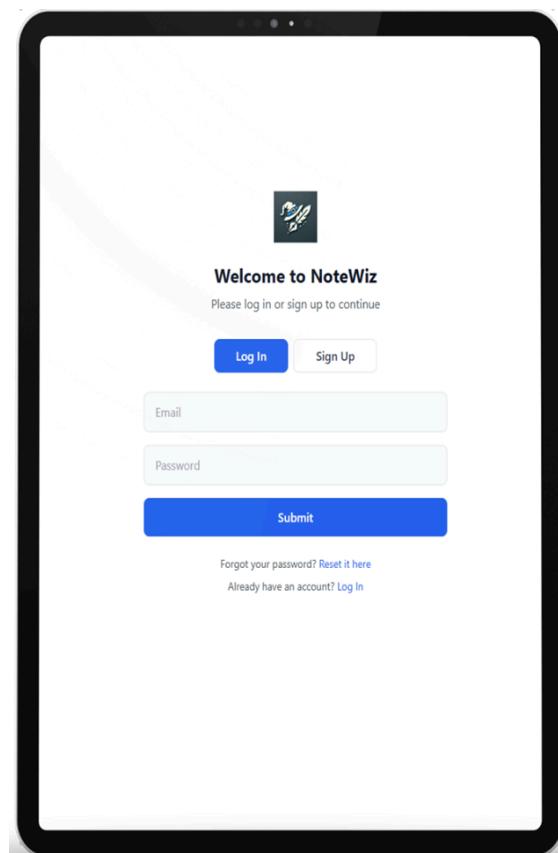


Figure 24 - Login Page UI

Title	Login
E-mail Field	Field where the user will enter his/her e-mail address. Accepts data only in e-mail format.

Password Field	The field where the user will enter the password. The characters are displayed hiddenly.
Remember Me Checkbox	Provides the option for the user to remember their login information.
Sign In Button	The main action button that allows the user to log in.
Google Login Icon	Icon that allows you to log in with a Google account.
Sign Up Link	Referral link for user to create account
Forgot Password Link	Allows the user to go to the reset screen when they forget their password.

### 5.5.2 Sign Up Page

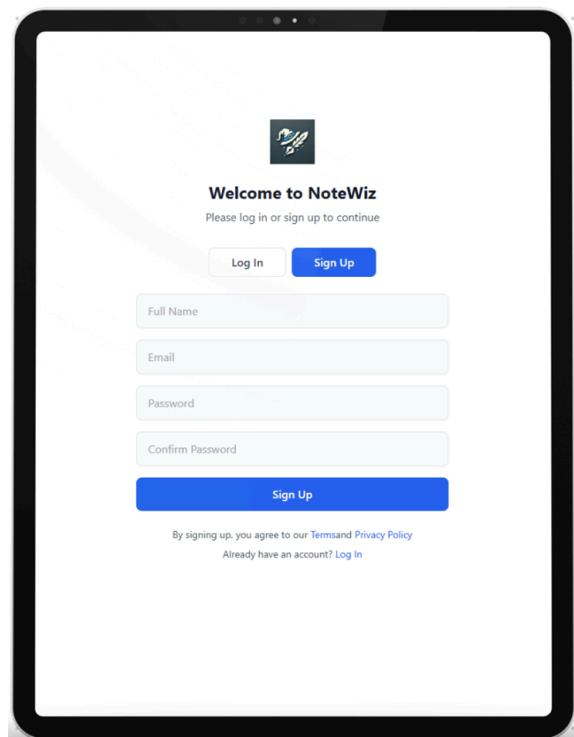


Figure 25- Sign Up Page UI

Title	Sign Up
-------	---------

Username Field	Textbox for entering the user's username.
E-mail Field	Field where the user will enter his/her e-mail address. Accepts data only in e-mail format.
Password Field	The field where the user will enter the password. The characters are displayed hiddenly.
Checkbox	Checkbox labeled " <i>I have read and agree to the term of service</i> ". Users must check this box before signing up.
Sign Up Button	Main button for account creation, highlighted in purple.
Google Sign-Up Button	Secondary button with Google icon for signing up via Google authentication.
Sign In Link	Text link for existing users to navigate to the login page. Label: " <i>You already have an account? Sign in</i> "

### 5.5.3 User Profile Management Page

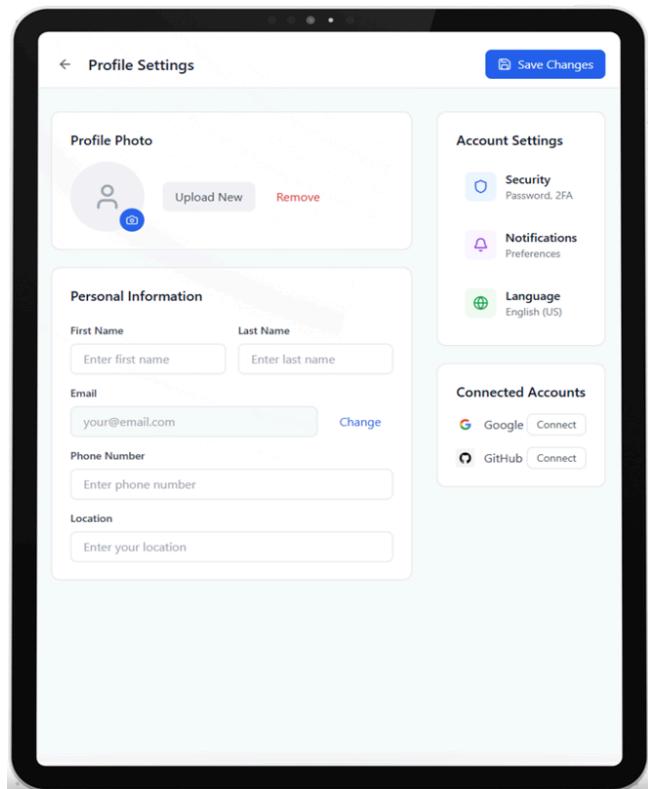


Figure 26 - User Profile Management Page UI

<b>Title</b>	User Profile Management
<b>Profile Picture Field</b>	Option to upload, view, or update the user's profile picture.
<b>Name Field</b>	Field for editing the user's display name.
<b>Email Field</b>	Editable field to update the user's email address. Email format validation is applied.
<b>Password Field</b>	Option to reset or change the user's password.

<b>Save Changes Button</b>	Button to save any modifications made to the user profile.
<b>Cancel Button</b>	Button to discard all changes and return to the previous menu.
<b>Security Button</b>	Button to go to security settings.
<b>Notification Button</b>	Button to go to notification settings.
<b>Language Button</b>	Button to go to language settings.
<b>Connected Accounts Field</b>	Field for connect other accounts with NoteWiz.

#### 5.5.4 AI Chat Page

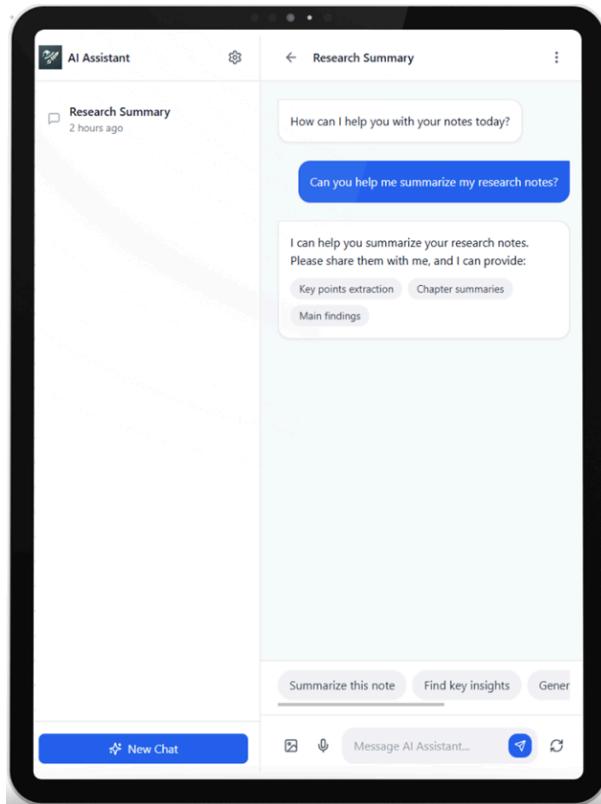


Figure 27-AI Chat Page UI

Title	AI Chat
<b>Chat Input Field</b>	A text field where the user can type questions or input queries.
<b>Submit Button</b>	Button to submit the query to the AI system for processing.
<b>Response Display</b>	Area where the system displays the AI's responses or suggestions.
<b>Follow-up Options</b>	Buttons or options for refining the question or asking follow-up queries.
<b>Exit Button</b>	Button to leave the chat interface and return to the main application menu.

## 5.5.5 Note Creation and Management

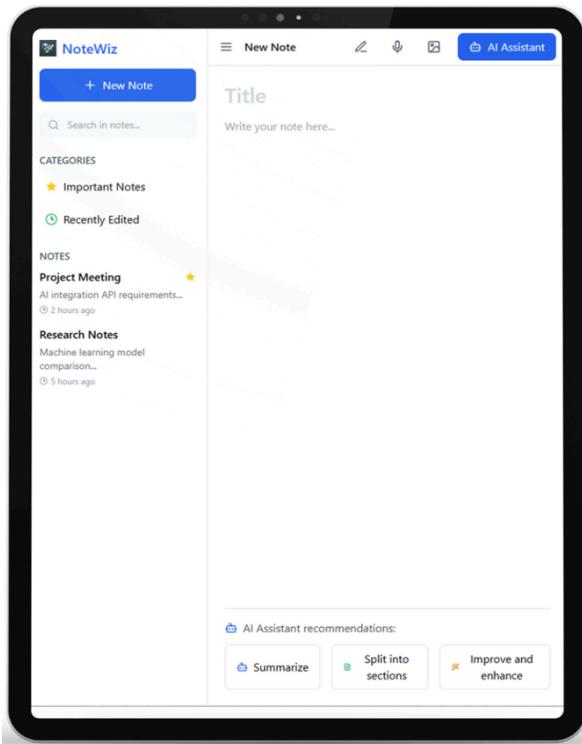


Figure -28 Note Creation and Manage Page UI

Title	Note Creation and Management
Title Field	Field where the user specifies the title of the note.
Content Field	Text area for entering the main content of the note.
Add Attachment Button	Button for uploading attachments, such as images, documents, or links.
AI Assistance Button	Button to enable AI-powered suggestions, including summaries or content generation.
Save Button	Button to save the note to the user's account.

<b>Cancel Button</b>	Button to discard the note and return to the previous screen.
----------------------	---

### 5.5.6 Note Visualization

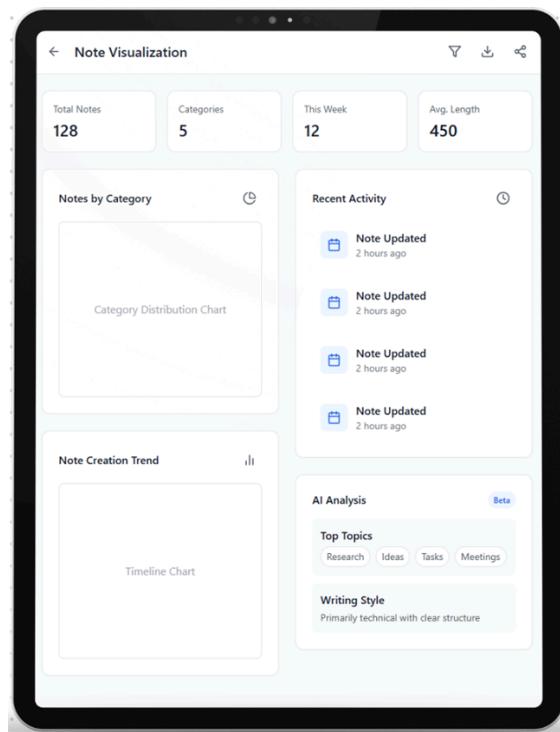


Figure 28 - Note Visualization Page UI

Title	Note Visualization
<b>Note List</b>	Displays all saved notes in a list format, showing titles, creation dates, and tags.
<b>Preview Pane</b>	Area for previewing the content of a selected note without fully opening it.
<b>Edit Button</b>	Button to open a note for editing.
<b>Delete Button</b>	Button to delete a selected note. Includes a confirmation prompt before action is executed.

### 5.5.7 Note Sharing

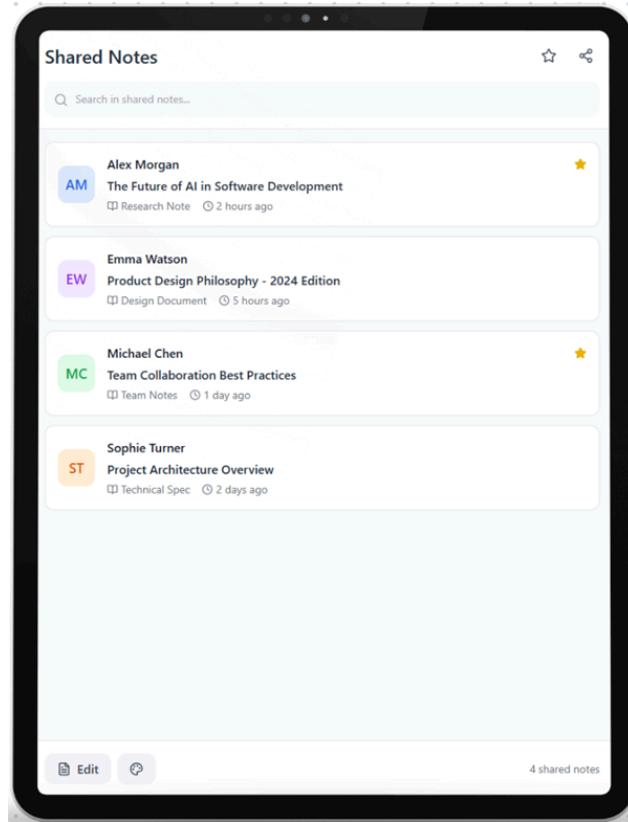


Figure 29 - Note Sharinn Page UI

Title	Note Sharing
Share via Email	Enables users to send notes directly to specific email addresses.
Generate Link	Creates a sharable link that can be distributed to other users.
Set Permissions	Allows the user to define access permissions (e.g., view-only, edit).
Share Confirmation	Notifies the user that the note has been successfully shared.

### 5.5.8 Document Upload and Processing

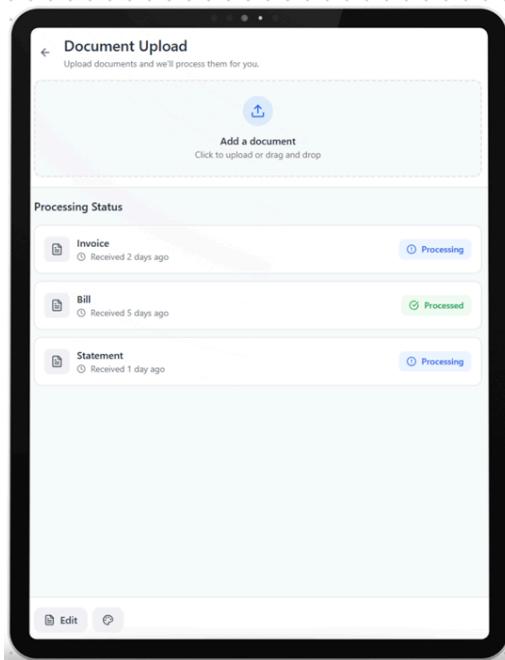


Figure 30 -Document Upload and Processing Page UI

Title	Document Upload and Processing
<b>Upload Button</b>	Button to select and upload a document file (e.g., PDF, Word).
<b>Processing Status</b>	Displays the progress of document processing (e.g., extracting content, generating summaries).
<b>Integration Options</b>	Provides options to incorporate extracted content into a new or existing note.
<b>Confirmation Message</b>	Notifies the user upon successful document upload and processing.

## 5.5.9 Admin Pages

### 5.5.9.1 User Record Page

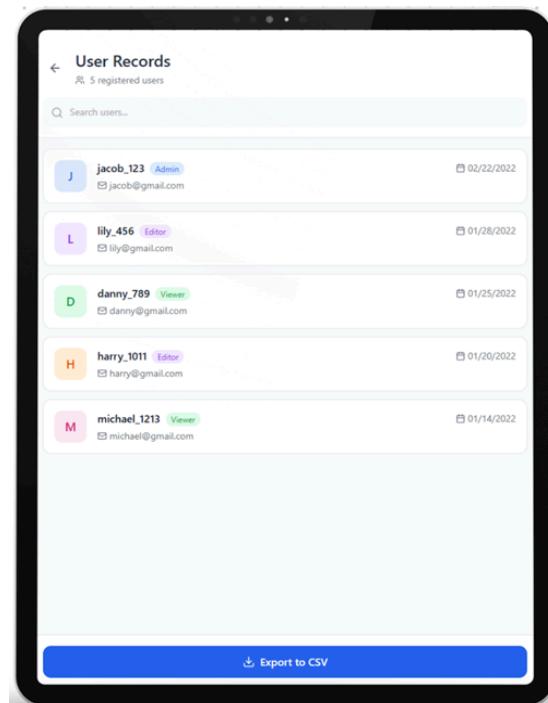


Figure 31 - User Record Page UI

Title	User Record Management
<b>User List</b>	Displays all registered users along with details such as username, email, and account status.
<b>Search Field</b>	Field to search for users by name or email.
<b>Delete Button</b>	Option to delete a user account permanently. Includes confirmation prompts.
<b>Export Button</b>	Button to export user records in a downloadable format.

### 5.5.9.2 User Access Control Page

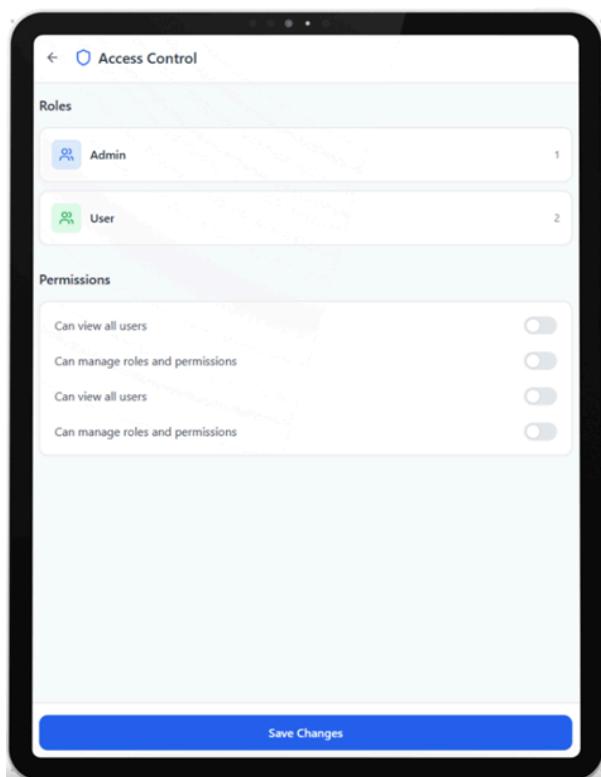


Figure 32 - User Access Control Page UI

Title	Access Control
<b>Role Management</b>	Allows admins to assign or modify user roles, such as admin or regular user.
<b>Add User Button</b>	Button to register a new user in the system.
<b>Remove User Button</b>	Option to remove a user from the system with necessary confirmations.

<b>Save Changes Button</b>	Button to apply and save all changes to user roles and permissions.
----------------------------	---

### 5.5.10 Forgot Password Pages

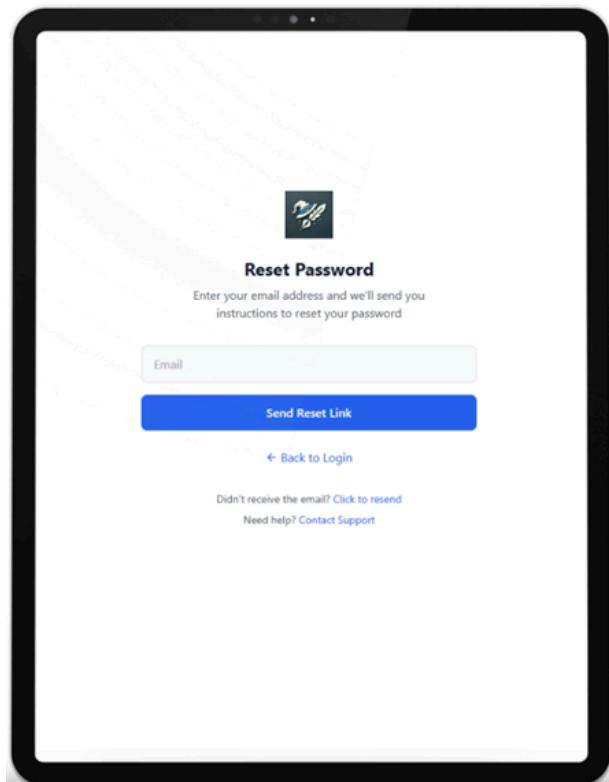


Figure 32 -Forgot Password Page UI

<b>Title</b>	<b>Forgot Password</b>
--------------	------------------------

<b>New Password Field</b>	Field where the user will enter new password.
<b>Verify Password Field</b>	Field where the user will verify new password.
<b>Reset Password Button</b>	Button for resetting password.
<b>Explanation Part</b>	Explain the new password constraints.

### 5.5.11 Settings Page

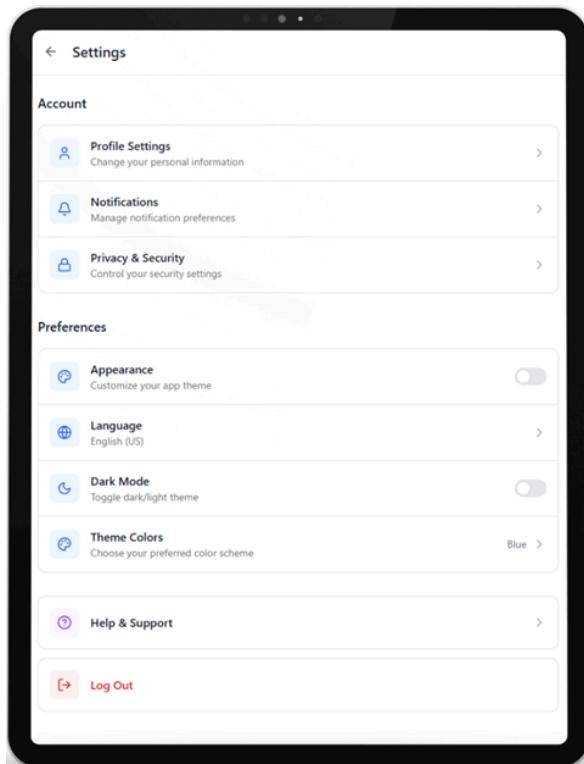


Figure 33- Settings Page UI

Title	Settings
<b>Account Settings</b>	Allows users to update their username, email, or password.
<b>Theme Preferences</b>	Toggle between light, dark and other themes for the application.

<b>Notification Settings</b>	Options to enable or disable specific notifications (e.g., email, in-app alerts).
<b>Privacy Settings</b>	Controls for data sharing permissions and visibility settings.
<b>Save Changes Button</b>	Button to confirm and save updates to user settings.

## 5.6 Help system design

The **Help System** for Notewiz is designed to provide users with comprehensive guidance on using the application effectively. The system ensures that users can easily navigate through the platform, resolve common issues, and maximize productivity with minimal external support. The following components form the core of the Help System:

1. **Interactive Tutorials**
  - When users first log in, they are guided through an interactive onboarding tutorial. This tutorial introduces the key features of Notewiz, such as creating notes, using AI-assisted tools, and managing tasks.
2. **Help Center / Knowledge Base**
  - The Help Center contains a detailed knowledge base with categorized articles, step-by-step instructions, and frequently asked questions (FAQs). Topics include:
    - How to create, edit, and organize notes
    - Using AI tools for summarization or suggestions
    - Managing account settings and preferences
    - Troubleshooting common issues
3. **Searchable User Manual**
  - A digital user manual, accessible from within the app, allows users to search for specific topics or commands. This manual includes visual aids like screenshots and diagrams to enhance comprehension.
4. **Chatbot / AI Assistance**
  - An AI-powered chatbot is integrated into the platform to provide instant answers to common queries. The chatbot can guide users step-by-step and redirect them to the appropriate help articles or tutorials when necessary.
5. **Customer Support Contact**
  - Users can submit support tickets directly within the app for unresolved issues. A response is typically provided within 24-48 hours.

## **6. Feedback Mechanism**

- A feedback form is included to allow users to report bugs, suggest features, or share their experiences with the app. This feedback is actively monitored and incorporated into future updates.

## 6. Conclusions

### 6.1 Summary of Work

This project focused on the design and development of NoteWiz, an AI-powered note-taking and productivity application. The core achievements include the integration of advanced AI features for automated summarization, personalized content generation, and task management. Leveraging modern frameworks like React for the front-end and Firebase for real-time synchronization, we built a robust, secure, and scalable platform. Key functionalities such as cross-platform compatibility, real-time collaboration, and document processing were implemented to enhance user experience and address gaps identified in existing solutions. The iterative Agile development process ensured continuous improvement and adaptability throughout the project lifecycle.

### 6.2 Key Learnings

Through this project, several important conclusions were drawn:

1. **AI Integration in Productivity Tools:** Incorporating artificial intelligence significantly enhances usability, providing personalized and efficient solutions for note-taking and task management.
2. **User-Centric Design:** An intuitive interface combined with features like dark mode, automated covers, and easy navigation improves engagement and accessibility across diverse user groups.
3. **Security and Scalability:** Secure authentication and data encryption ensure user trust, while modular architecture allows for easy scalability and future enhancements.
4. **Limitations in Current Tools:** Existing applications lack sufficient AI capabilities and seamless integration, which NoteWiz addresses effectively.

### 6.3 Future Work and Directions

While NoteWiz has achieved its primary objectives, several opportunities for future development exist:

- **Enhanced AI Features:** Integrating more sophisticated natural language processing models to support deeper contextual understanding and more precise summarization.
- **Offline Functionality:** Expanding the app to support offline note creation and syncing updates once connected to the internet.
- **Third-Party Integrations:** Adding support for seamless integration with other productivity tools like Trello, Slack, or Google Workspace.
- **Collaborative Features:** Enhance real-time note-sharing and task management functionalities with live editing and commenting.

## **6.4 Open Problems**

The following challenges remain unresolved and provide avenues for future exploration:

1. AI Bias and Accuracy: Ensuring that AI responses are unbiased and contextually accurate remains a critical challenge.
2. Real-Time Collaboration Conflicts: Managing simultaneous edits across devices and users requires more sophisticated conflict resolution mechanisms.
3. Privacy Concerns: Balancing advanced AI functionalities with strict privacy standards, especially for sensitive user data, needs continuous evaluation.
4. Performance Optimization: Addressing latency issues during high traffic or resource-intensive AI interactions.

By addressing these challenges, NoteWiz can further solidify its position as a pioneering productivity tool in an increasingly digital workspace.

## **6.5 Advantages and Disadvantages**

### **Advantages:**

- Combines advanced AI functionalities with everyday productivity needs.
- Offers a clean, distraction-free interface designed for intuitive use.
- Ensures robust data security through encryption and two-factor authentication.
- Provides cross-platform synchronization, enhancing usability across devices.

### **Disadvantages:**

- Reliance on an active internet connection for AI features and synchronization.
- Current limitations in localized support and advanced personalization.
- Limited scope of collaborative tools compared to dedicated project management software.

## Acknowledgement

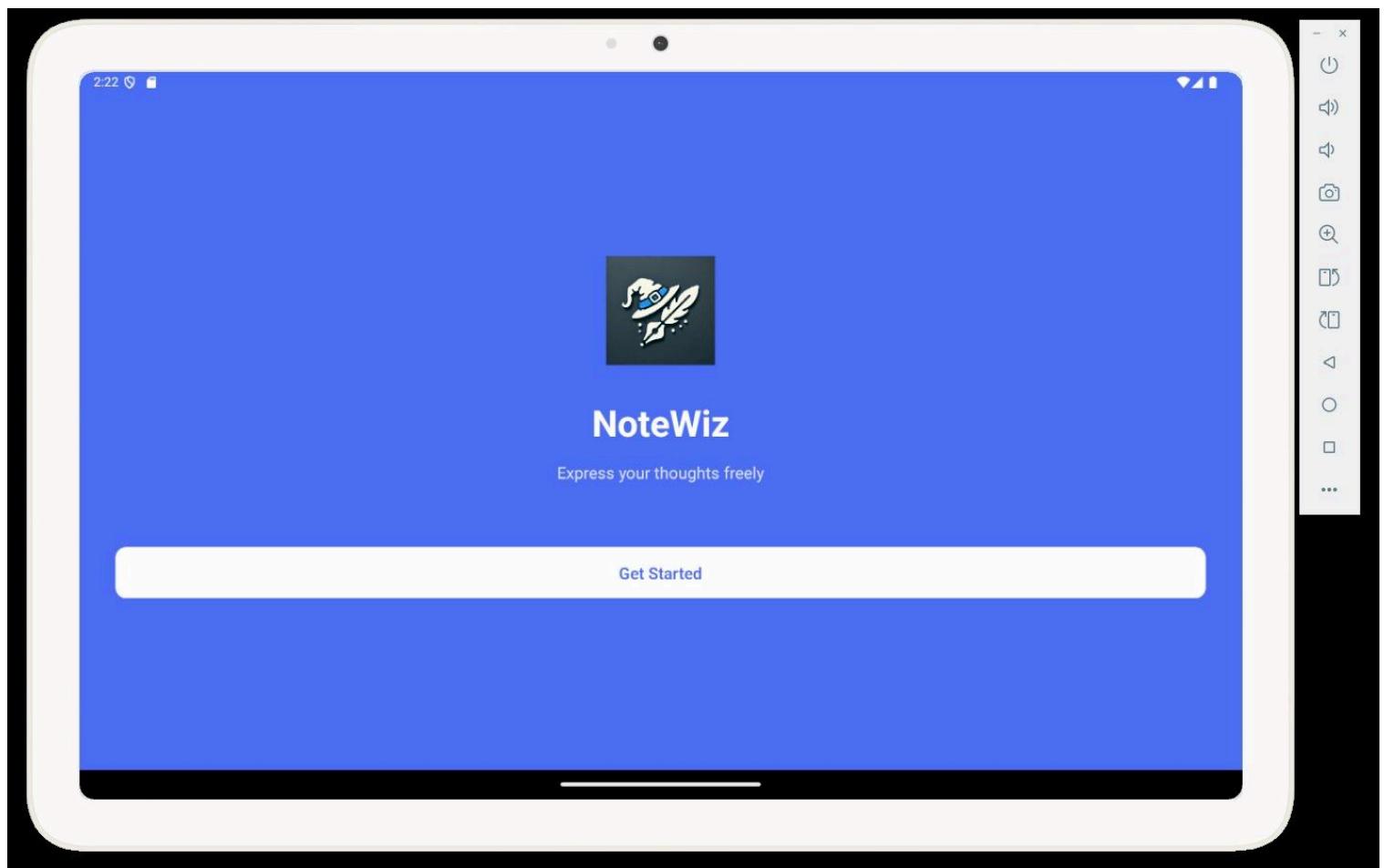
We would like to express our sincere gratitude to everyone who supported us throughout the development of the Notewiz project. First of all, we would like to sincerely thank our project advisor, Assoc. Prof. Faris Serdar Taşel, for his invaluable guidance, encouragement, and insightful feedback that greatly contributed to the completion of this project.

## References

1. T. T. Chamiss, "Mobile App Security: Safeguarding User Data and Privacy," *Cardinal Peak Blog*. [Online]. Available: <https://www.cardinalpeak.com/blog/mobile-app-security-safeguarding-user-data-and-privacy>. [Accessed: Nov. 22, 2024].
2. R. Gopikrishnan, et al., "AI-powered productivity tools: Enhancing efficiency and automation in business and personal productivity," *Journal of Artificial Intelligence in Business*, vol. 25, no. 3, pp. 112–128, 2021.
3. *Evernote, Microsoft OneNote, Notion: Comparative Analysis, Productivity App Review by Tech Insights*. [Online]. Available: <https://www.techinsights.com>. [Accessed: 2020].
4. J. Smith and T. Clark, "The role of NLP and reinforcement learning in improving task management tools," *International Journal of Productivity and AI*, vol. 30, no. 4, pp. 45–59, 2022.
5. Otter.ai, "Speech-to-text and transcription solutions for meetings." [Online]. Available: <https://www.otter.ai>. [Accessed: 2022].
6. L. Jackson and E. Morris, "Leveraging AI to create immersive virtual classrooms for enhanced student engagement," *Educational Technology Research and Development*, vol. 38, no. 5, pp. 455–474, 2021.
7. *Slack, Asana: Integration of AI for Efficient Task Management, Productivity Tools Review*. [Online]. Available: <https://www.productivitytoolsreview.com>. [Accessed: 2021].
8. "BERT, GPT: NLP Techniques for Task Automation in Productivity Apps," *Journal of Computational Linguistics*, vol. 12, no. 1, pp. 78–93, 2020.
9. "TensorFlow, PyTorch: Frameworks for Machine Learning in AI Applications," *AI and Machine Learning Technology Review*, vol. 45, no. 2, pp. 234–256, 2020.
10. Firebase, "Real-time Data Synchronization for Productivity Apps." [Online]. Available: <https://firebase.google.com>. [Accessed: 2021].
11. "Google Cloud and AWS: Cloud Infrastructure for Scalable AI Solutions," *Cloud Computing in AI*, vol. 18, no. 6, pp. 123–142, 2021.

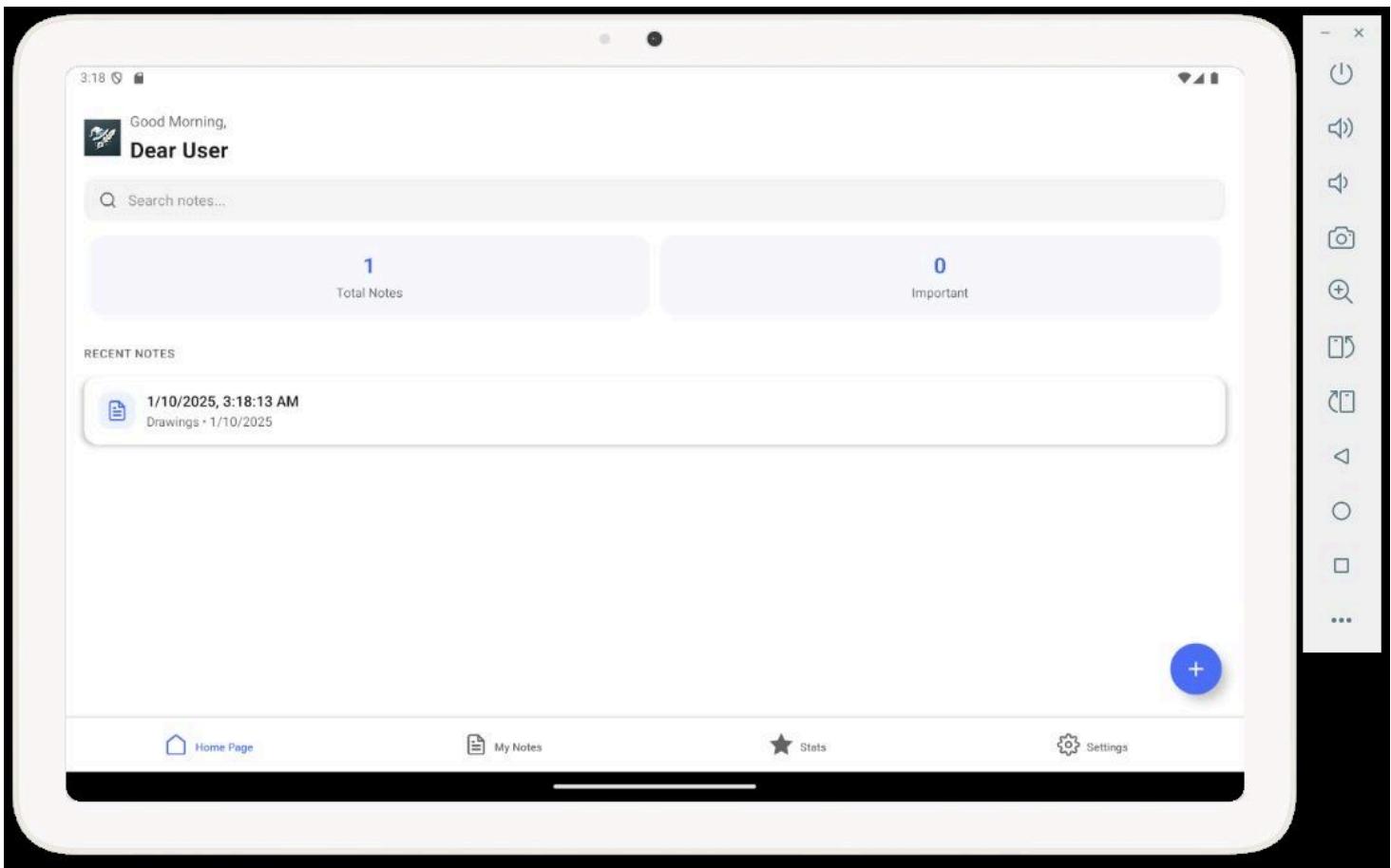
12. Zapier, "AI Note-Taking Apps Review: Automating Tasks for Enhanced Productivity." [Online]. Available: <https://zapier.com/blog/ai-note-taking/>. [Accessed: 2022].
13. Time Magazine, "AI in Meeting Productivity: Tools that Enhance Efficiency." [Online]. Available: <https://time.com>. [Accessed: 2021].
14. "GazeNoter: Co-Piloted AR Note-Taking via Gaze Selection of LLM Suggestions to Match Users' Intentions," *ARXIV: AI and Augmented Reality for Note-Taking*. [Online]. Available: <https://arxiv.org/abs/2305.01759>. [Accessed: 2023].
15. *Notion, Evernote, Microsoft OneNote: Comparative Analysis of Productivity Apps, Tech Insights*. [Online]. Available: <https://www.techinsights.com>. [Accessed: 2020].
16. OCR: Optical Character Recognition," *IEEE Dataport*. [Online]. Available: <https://ieee-dataport.org/keywords/ocr-optical-character-recognition>. [Accessed: Dec. 3, 2024].
17. [2] A. Jo, "How to Build a Native OpenAI Integration," *Paragon Blog*. [Online]. Available:<https://www.useparagon.com/blog/how-to-build-a-native-openai-integration>. [Accessed: Dec. 5, 2024].

## Appendices



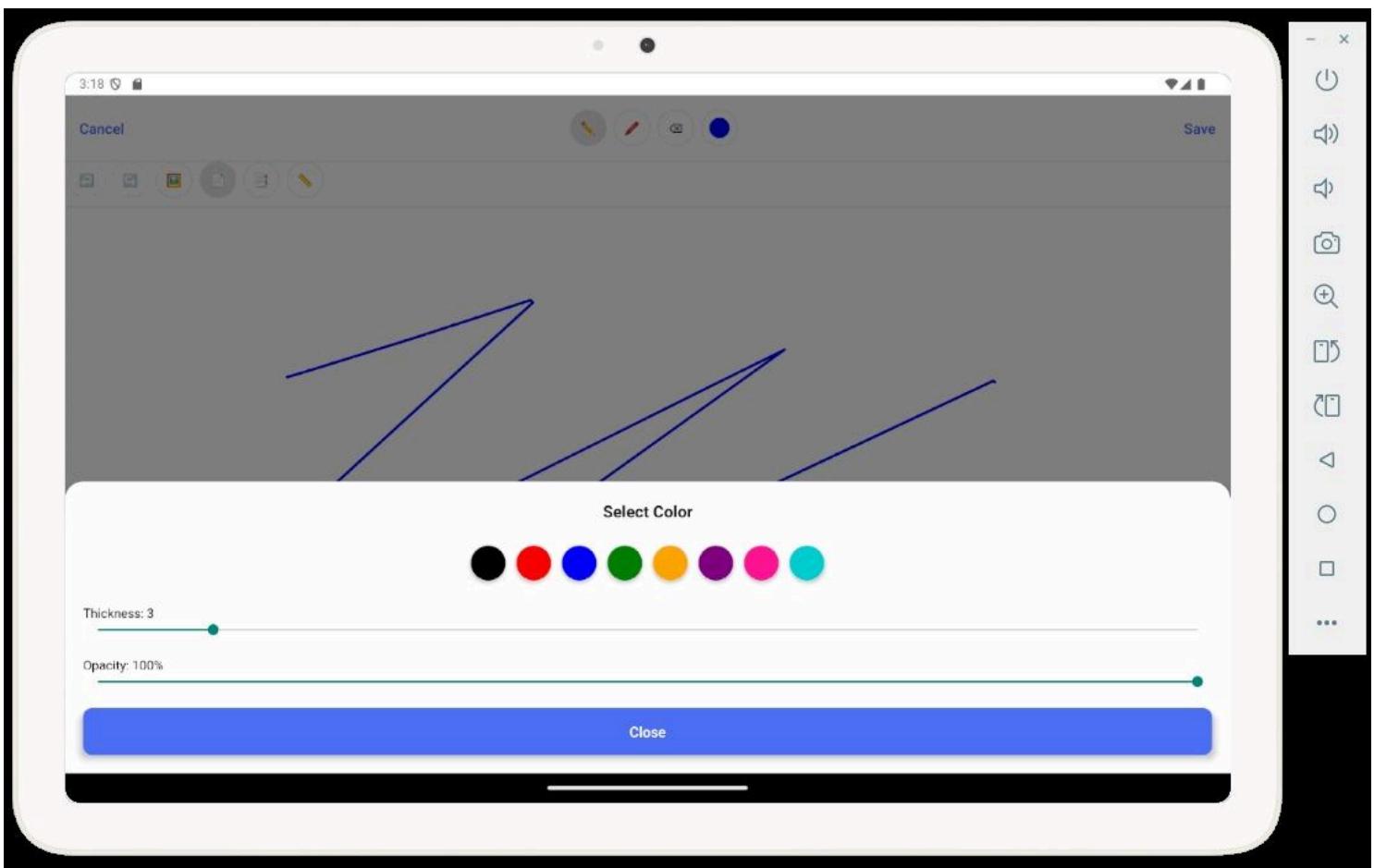
***Figure 1.1: Welcome Page***

This figure shows the introductory screen of the NoteWiz application. The design is minimalist, featuring the app's logo and the tagline "Express your thoughts freely," along with a "Get Started" button to initiate user interaction.



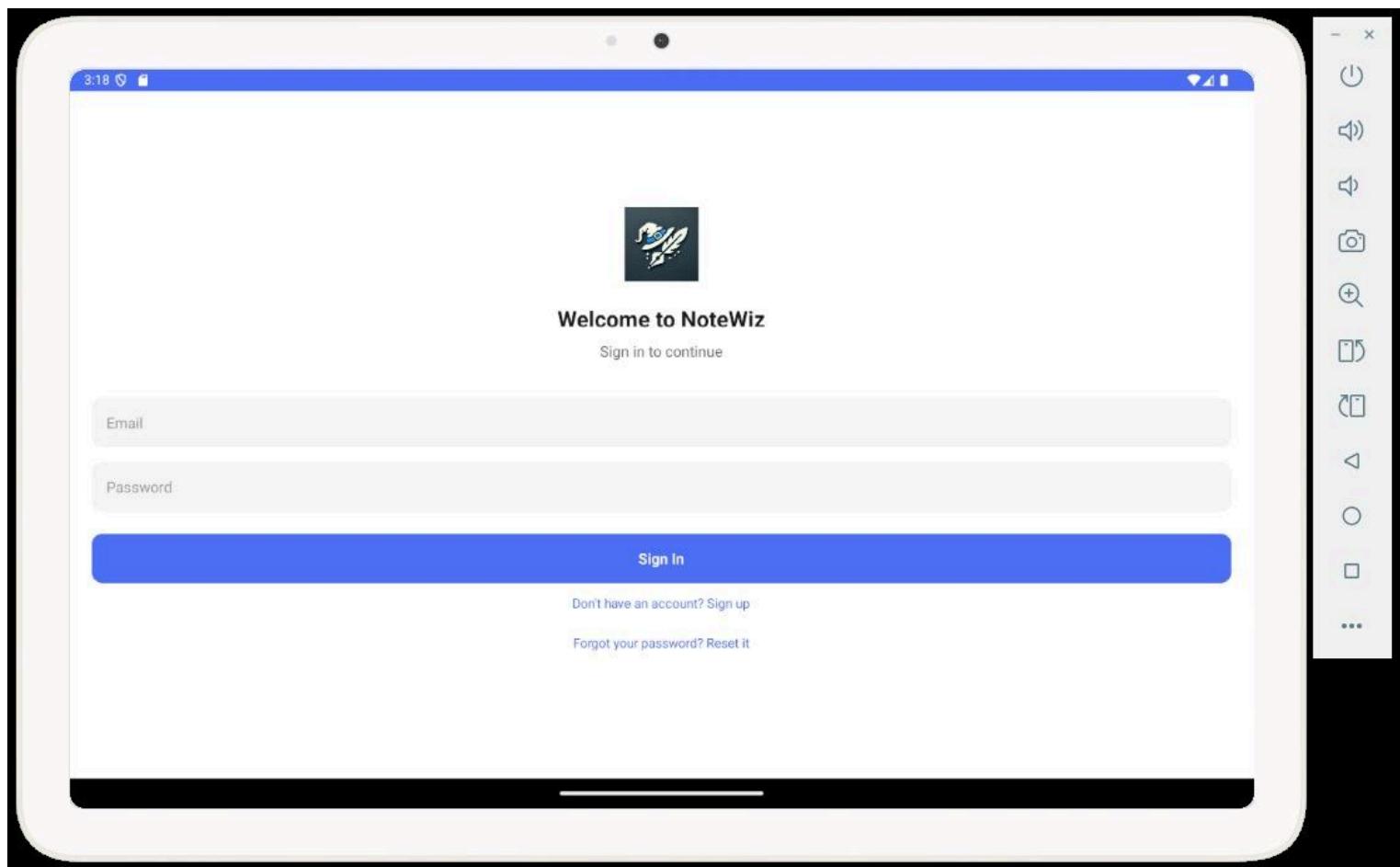
**Figure 1.2: Home Page**

The home page displays an overview of the user's notes and key statistics. Users can search notes, view the total number of notes, and access recent notes for quick navigation.



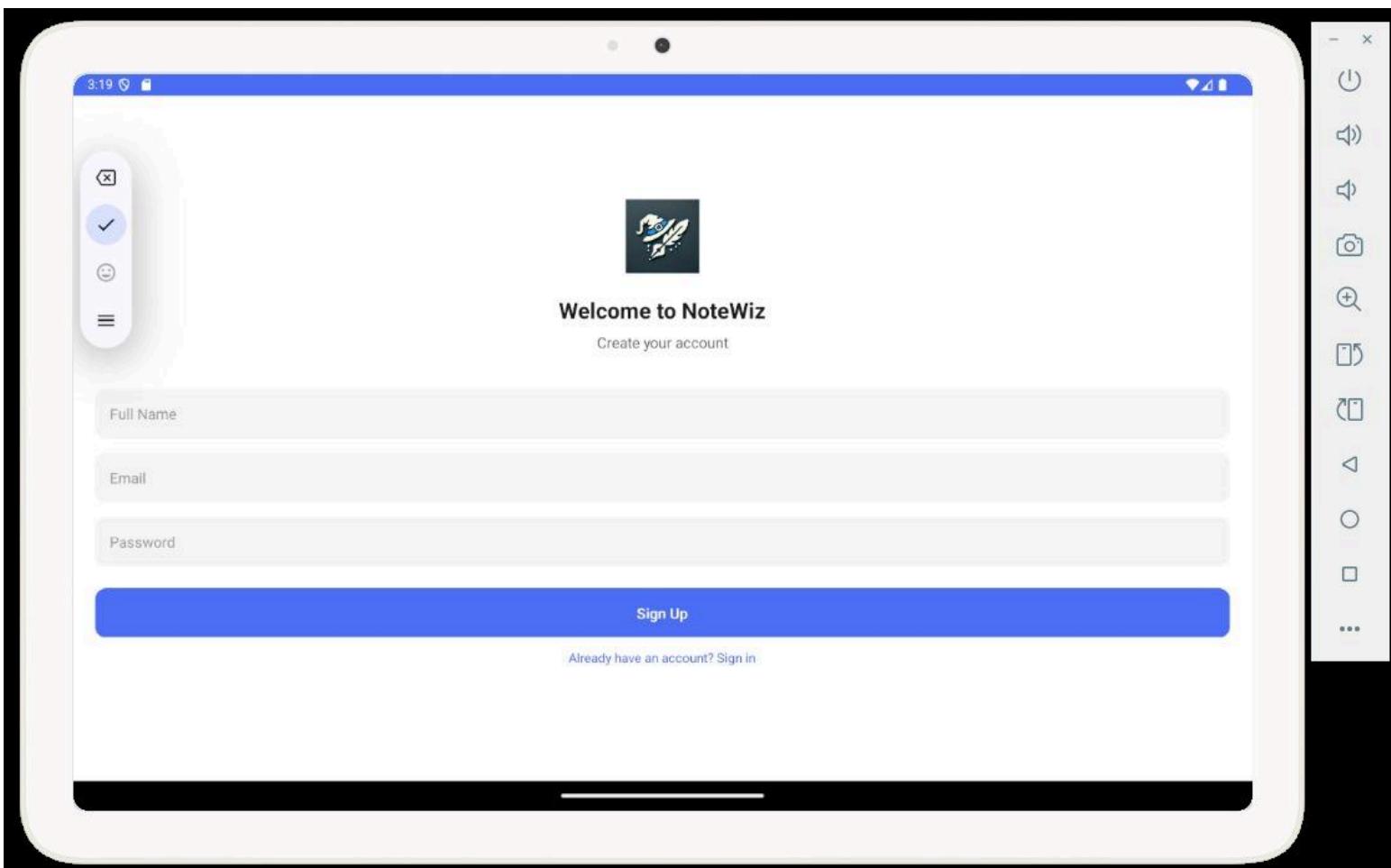
**Figure 1.3: Drawing Interface**

This figure illustrates the note-drawing feature in NoteWiz. Users can select colors, adjust pen thickness, and set opacity, providing a customizable drawing experience for visual notes.



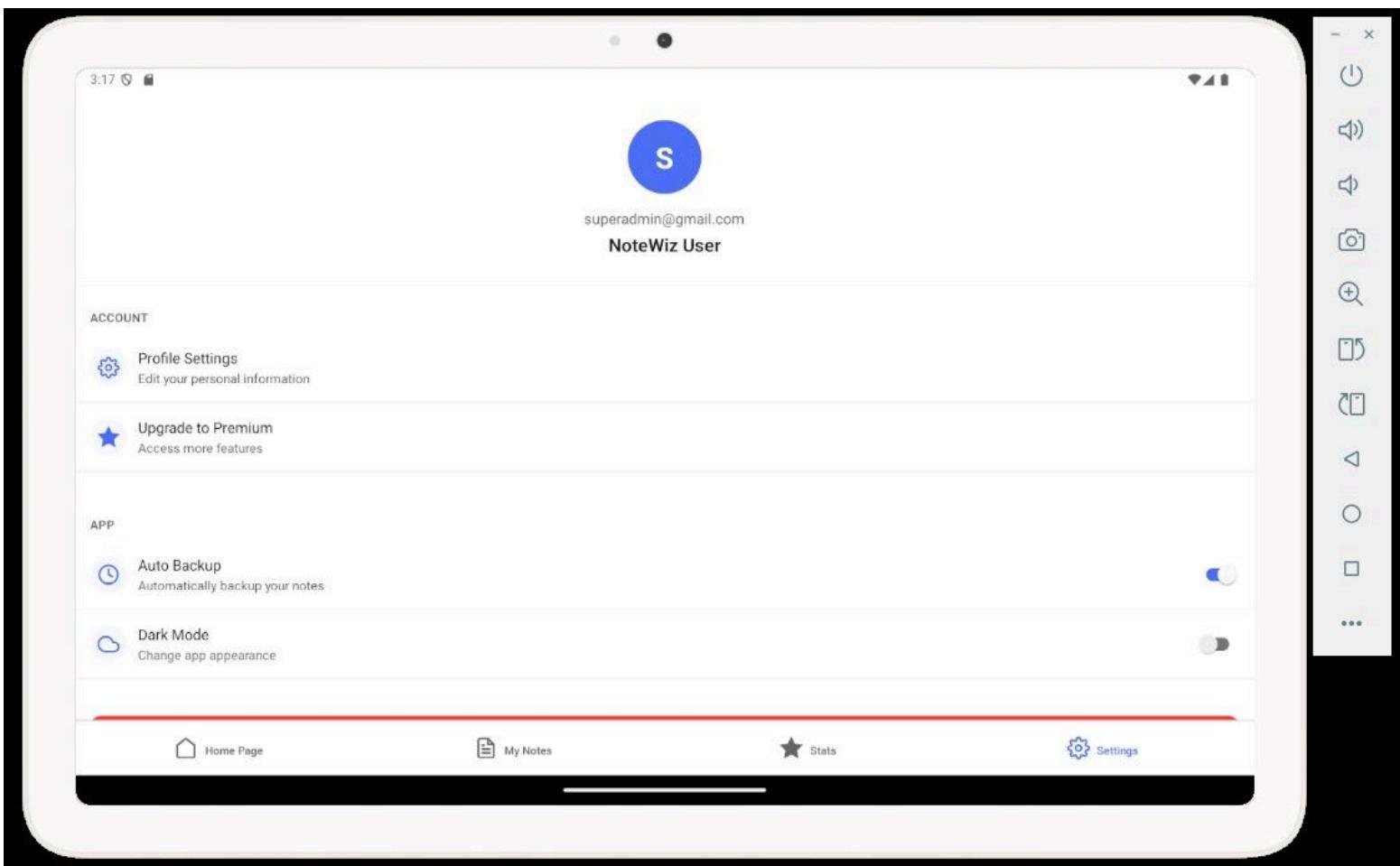
**Figure 1.4: Sign-In Page**

The sign-in page is where users log into their accounts. It includes fields for email and password, along with options to sign up or reset forgotten passwords.



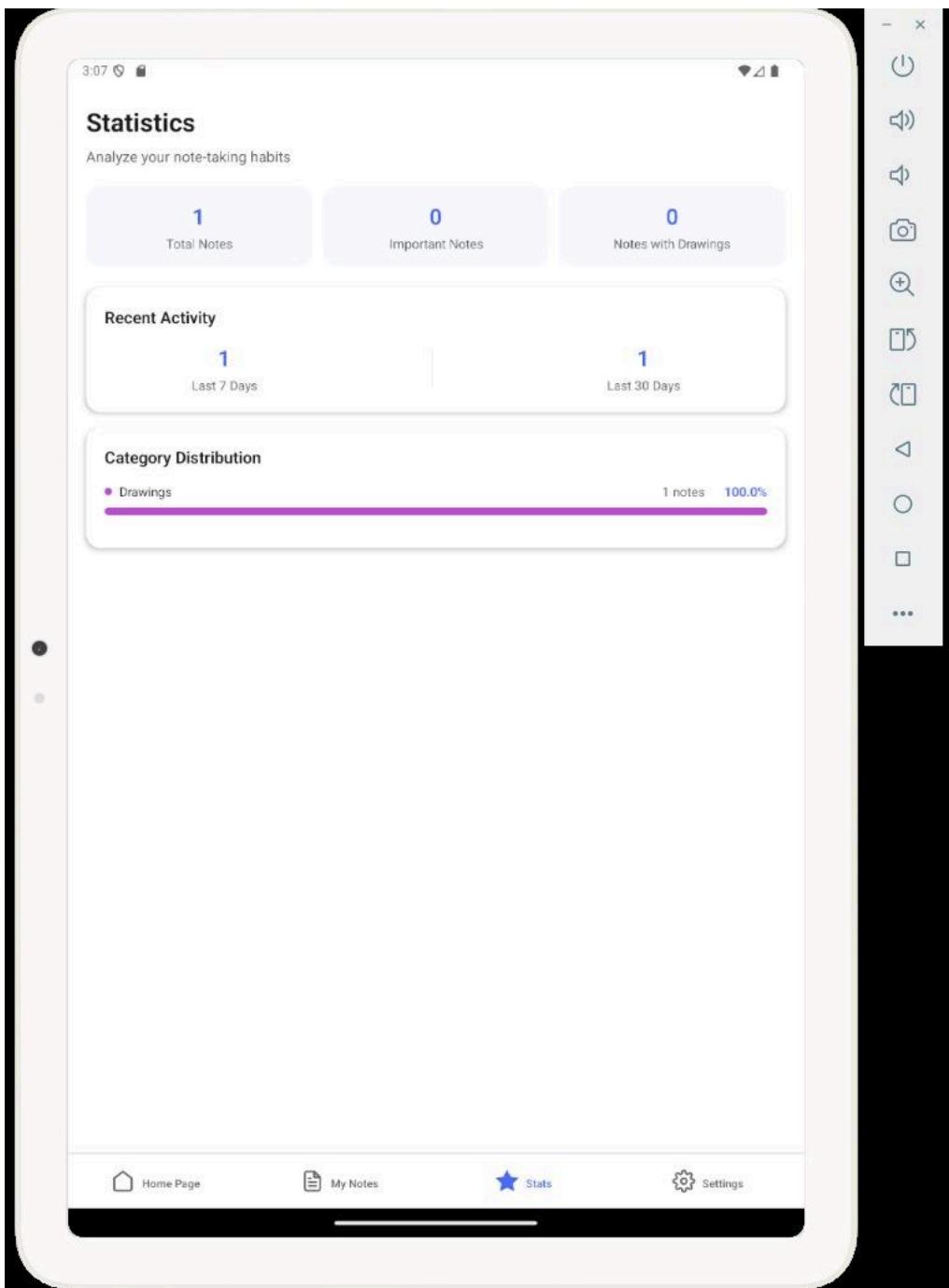
**Figure 1.5: Sign-Up Page**

This page is for new users to create an account. It requires input for full name, email, and password. A clear, user-friendly design ensures ease of use.



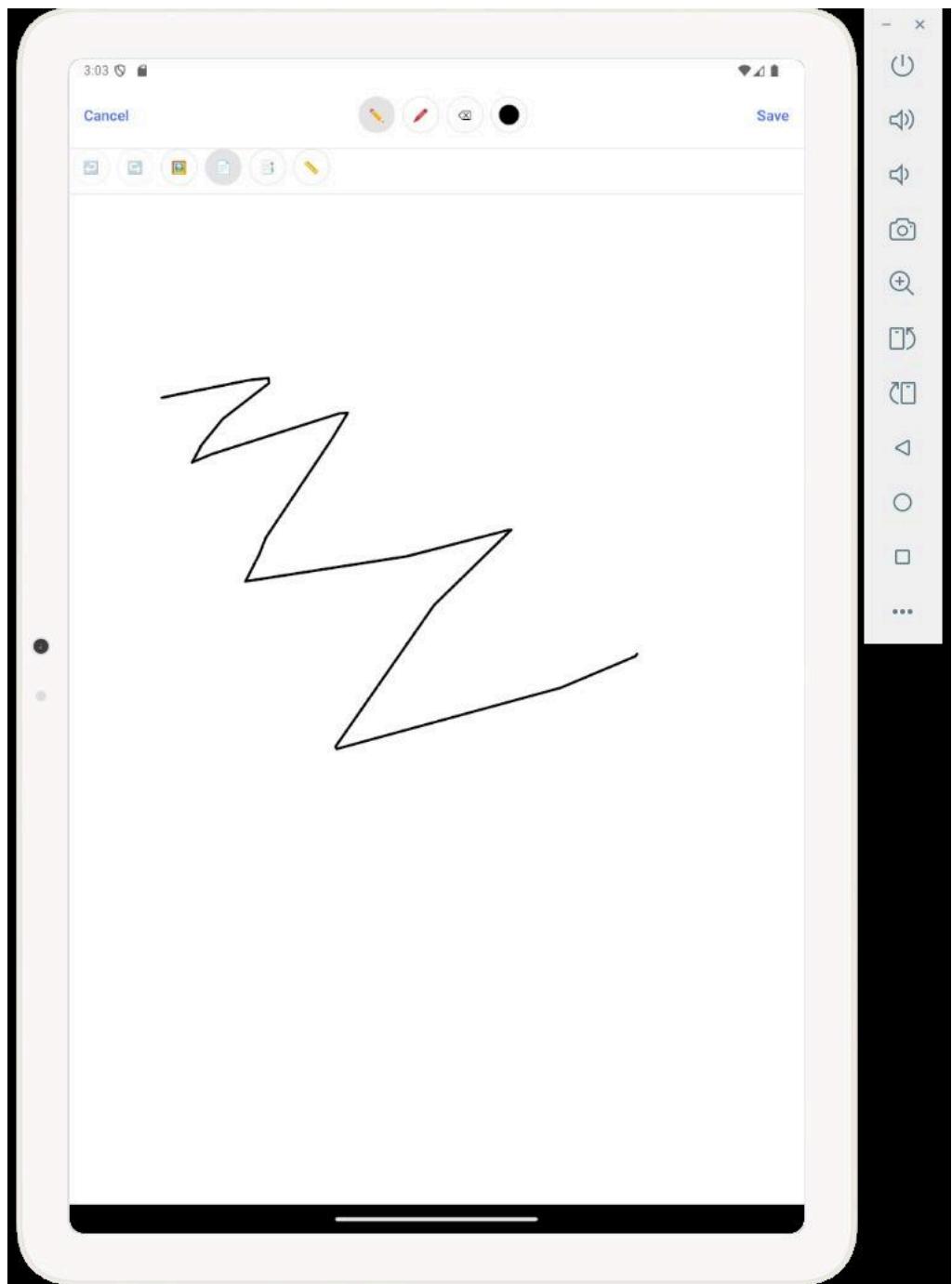
**Figure 1.6: User Settings**

The settings page allows users to manage their accounts, including profile settings, app preferences, and additional features like enabling dark mode or upgrading to premium.



**Figure 1.7: Statistics Page**

The statistics page helps users analyze their note-taking habits. It includes metrics such as total notes, important notes, recent activity, and category distribution for better organization.



**Figure 1.8: Drawing Page**

This figure shows the core functionality of the drawing interface in the NoteWiz application. Users can freely sketch their ideas or diagrams using the drawing tools available at the top of

the interface. This feature supports visual note-taking and enhances creativity, making it suitable for diverse applications like brainstorming or design.

## Current State Assessment

The prototype successfully demonstrates the core functionalities and user interface of the application. Its simplicity and functionality prioritize user experience. However, the following areas present significant opportunities for further development:

1. **Feature Enrichment:** Advanced features such as AI-based note summarization and task management can be incorporated into the application.
2. **Performance and User Experience:** The speed and responsiveness of features, particularly the drawing tools, can be optimized.
3. **Inclusivity:** Multi-language support and customization options for different user needs can be developed further.