|  | **ÇANKAYA UNIVERSITY**  **COMPUTER ENGINEERING DEPARTMENT** |  |
| --- | --- | --- |

**CENG 407**

**Software Design Description**

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

**Prepared by: Eray Ateş, Halit Emir Turan, Zeynep Persil, Güneş Ölçer**

**Date: 25/12/2024**

Table of Contents

[**1 Introduction 1**](#_1y810tw)

[**1.1 Purpose 1**](#_4i7ojhp)

[**1.2 Scope 1**](#_2xcytpi)

[**1.3 Definitions, acronyms, and abbreviations 1**](#_1ci93xb)

[**1.4 References 1**](#_3whwml4)

[**1.5 Overview 1**](#_2bn6wsx)

[**2 Architecture 1**](#_qsh70q)

[**2.1**](#_3as4poj) **General Design Approach of the Project** [**2**](#_3as4poj)

[**2.2**](#_1pxezwc) **Architectural Design** [**2**](#_j8sehv)

[**2.2.1 Front-end Layer**](#_49x2ik5)**……………………………………………………………………………………………..**

[**2.2.2 Back-end Layer**](#_2p2csry)

### [2.2.3 Database Layer](#_gjdgxs)

[**2.2.4 Integration and Modularity**](#_23ckvvd)

[**3 System Design 2**](#_338fx5o)

[**3.1 UML Diagram**](#_ihv636)

[**3.2 Decomposition Description**](#_32hioqz)

[**3.2.1 Database Layer**](#_1hmsyys)

[**3.2 2 User Interface Logic and Workflows**](#_41mghml)

## [3.2.2.1 Sign up and Log in](#_30j0zll)

[**3.2.2.2 Profile Management**](#_37m2jsg)

[**3.2.2.3 Main Menu**](#_3cqmetx)

[**3.2.2.4 Note Creation and Management**](#_34g0dwd)

[**3.2.2.4.1 Note Creation and Management**](#_43ky6rz)

[**3.2.2.4.2 Note Visualization**](#_2iq8gzs)

### [3.2.2.5 Note Sharing…………………………………………………………………………..25](#_1fob9te)

### [3.2.2.6 Document Upload and Processing………………………………………...25](#_3znysh7)

### [3.2.2.7 AI Chat Page (AI-powered question asking)](#_2et92p0)

### [3.2.2.8 Forgot Password Pages](#_42ddq1a)

### [3.2.3 Admin Page](#_2afmg28)

### 3.2.3.1 User Record Page

### *3.2.3.2* User Access Control Page

[**3.2.4 Settings Page**](#_2nusc19)

**3.3 User Interface**

### [3.3.1 Login Page](#_2s8eyo1)

[**3.3.2 Sign Up Page**](#_haapch)

[**3.3.3 User Profile Management page**](#_1gf8i83)

### [3.3.4 AI Chat Page](#_17dp8vu)

### [3.3.5 Note Creation and Management](#_3rdcrjn)

### [3.3.5.1 Note Creation and Management](#_26in1rg)

### [3.3.5.2 Note visualization](#_1tuee74)

### [3.3.6 Note Sharing…………](#_2hio093)

### [3.3.7 Document Upload and Processing](#_1ksv4uv)

### [3.3.8 Admin Pages](#_44sinio)

### [3.3.8.1 User Record Page](#_wnyagw)

### *3*[*.3.8.2* User Access Control Page](#_45jfvxd)

### [3.3.9 Forgot Password Pages](#_3gnlt4p)

**3.3.10 Settings Page**

[**3**](https://docs.google.com/document/d/10Ip_r-EpnMTDK8pbAEiBnaQ0e1CXckSmUDyDe8mV0EE/edit?tab=t.0#heading=h.k3h0vvogx9xd)**.4 Software System attributes...................................................................................27**

**3.4.1. Portability ..............................................................................................................27**

**3.4.2. Usability .................................................................................................................27**

**3.4.3. Adaptability ...........................................................................................................27**

**3.4.4. Security............................................................................................................27**

**3.4.5 Reliability…………………………………………………………………27**

**3.4.6 Availability…………………………………………………………….28**

**3.4.7 Maintainability……………………………………………………………….28**

[**4. References**](#_243i4a2)**………………………………………………………………………………………………………………..**

# Introduction

## 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.

## 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.[2]

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.[3]

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.

## Definitions, acronyms, and abbreviations

| Term | Description |
| --- | --- |
| OS | Operating System |
| IP | Internet Protocol Address |
| SSD | Solid State Disk |
| 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. |

## References

See 7. References section.

## Overview

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.

# Architecture

## General Design Approach of the Project

The NoteWiz project focuses on providing an intuitive and productivity-enhancing user experience. A user-centric design ensures that the platform is simple, accessible, and tailored to the needs of students and professionals. With a clean, minimalist interface, users can effortlessly navigate features such as note creation, task planning, document processing, and AI-powered question answering, making interactions seamless and efficient.

The integration of advanced AI APIs increases the accuracy and variety of features, enabling intelligent summarization and content personalization. Security and data privacy are top priorities, and robust encryption measures protect user information and encourage trust.

Visually appealing elements such as auto-generated note covers and customizable themes enhance the usability and aesthetic appeal of the platform. Regular testing and iterative feedback ensure continuous improvement and feature optimization.

Analytics are used to increase user satisfaction by improving AI recommendations and streamlining workflows. The modular design allows for adaptability and future feature integrations, ensuring NoteWiz remains innovative and in sync with user needs. NoteWiz offers an engaging platform for cross-device note sharing and cross-platform accessibility.

## Architectural 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.

### 2.2.1 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.

### 2.2.2 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.

### 2.2.3 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]

### 2.2.4 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.

# SYSTEM DESIGN

## UML Diagram

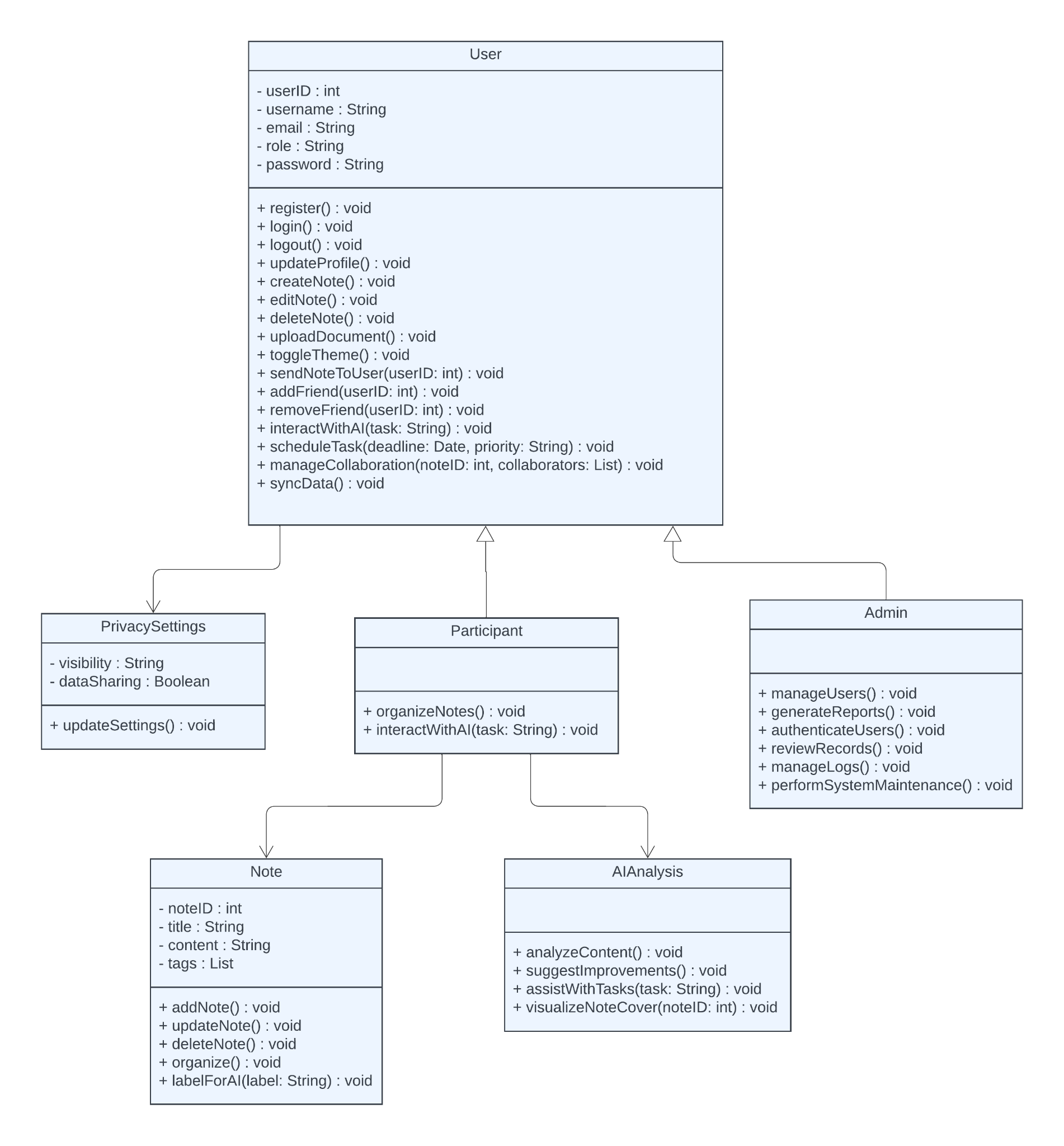


Figure1- System Design

## Decomposition Description

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

### 3.2.1 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**:

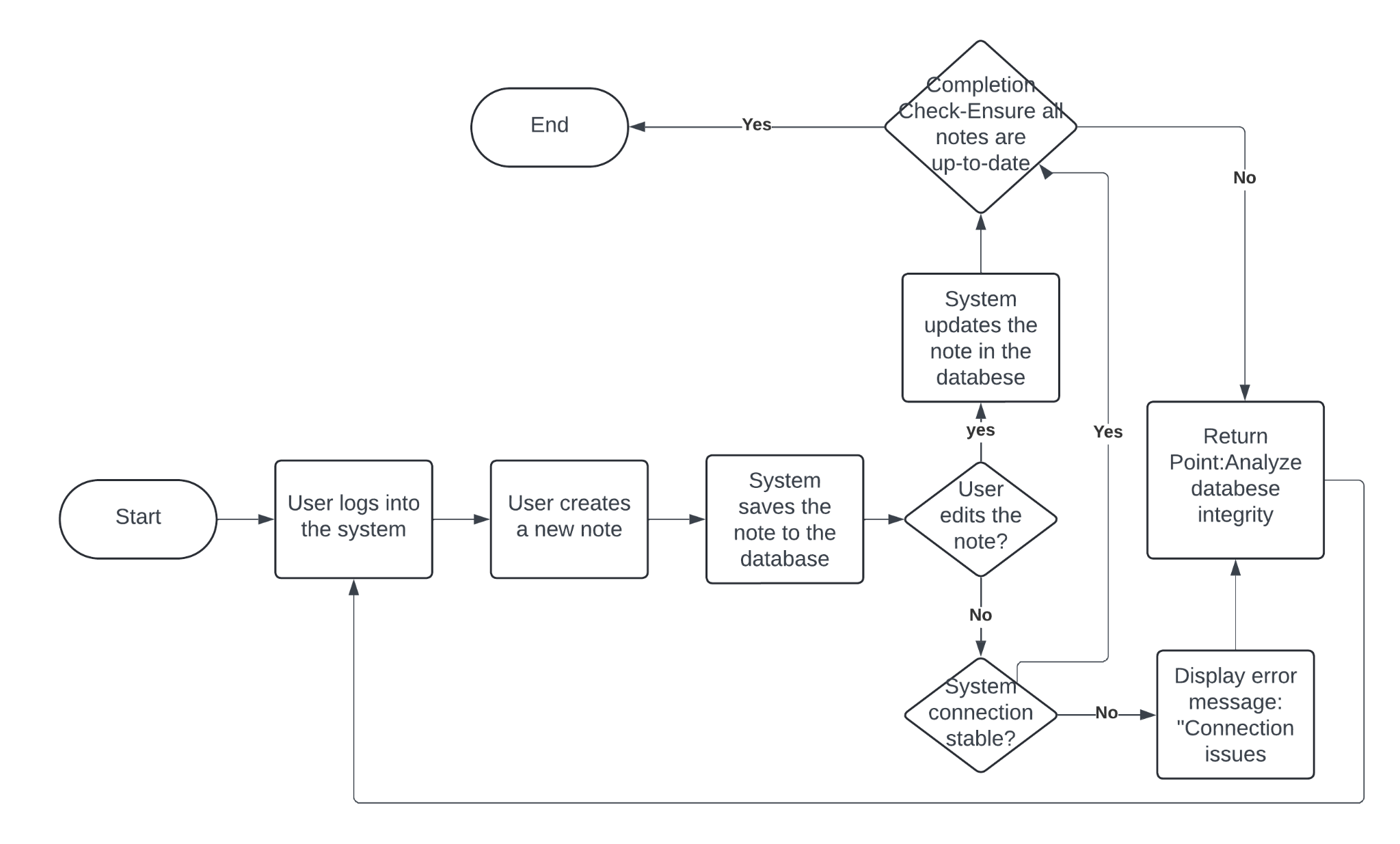
* Notes are securely stored, updated, or managed in the system database.

**Post Conditions**:

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

**Priority**:

* High

Figure-2 Database Layer

## 3.2 2 User Interface Logic and Workflows

### 3.2.2.1 [Sign up and Log in](#_30j0zll)

#### **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

Figure-3 Sign Up and Login

### 3.2.2.2 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-4 Profile Management

## 3.2.2.3 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:

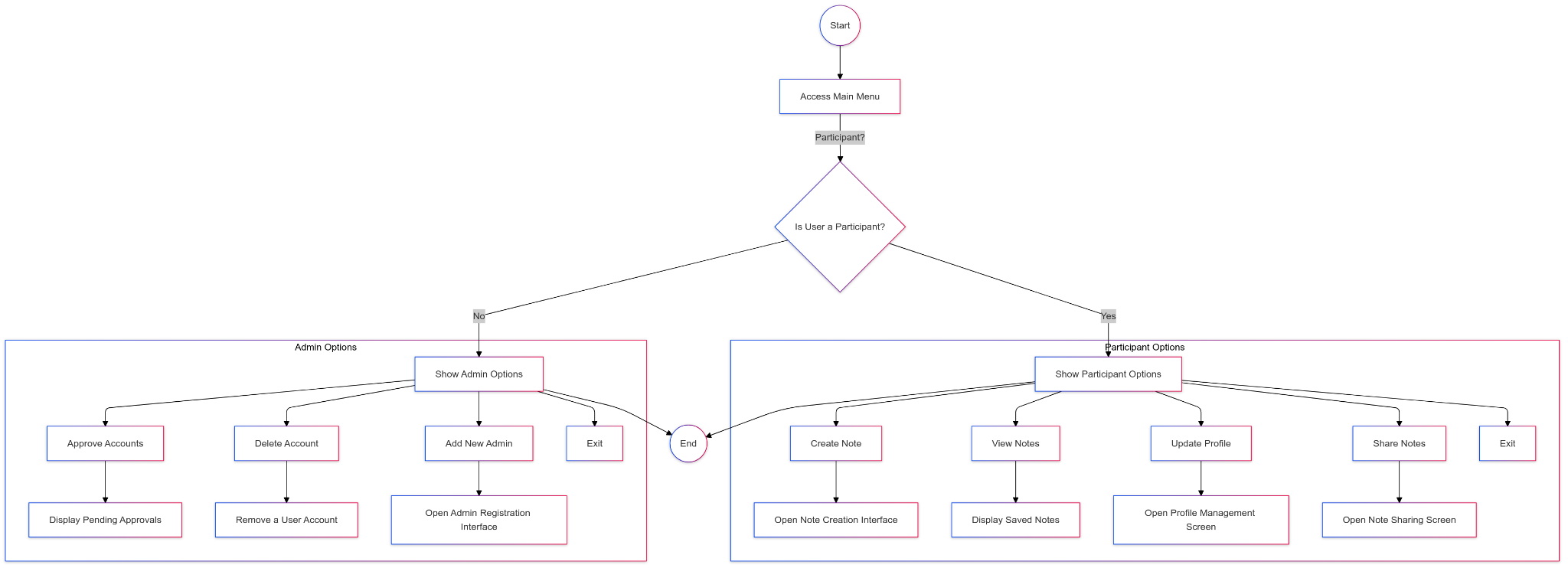
* **Menu Load Failure**:
  + If the main menu fails to load, an error message is displayed, and the user is prompted to restart the application.

### Post Conditions:

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

### Priority:

* Medium



### Figure-5 Main Menu

### 3.2.2.4 Note Creation and Management

### 3.2.2.4.1 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.

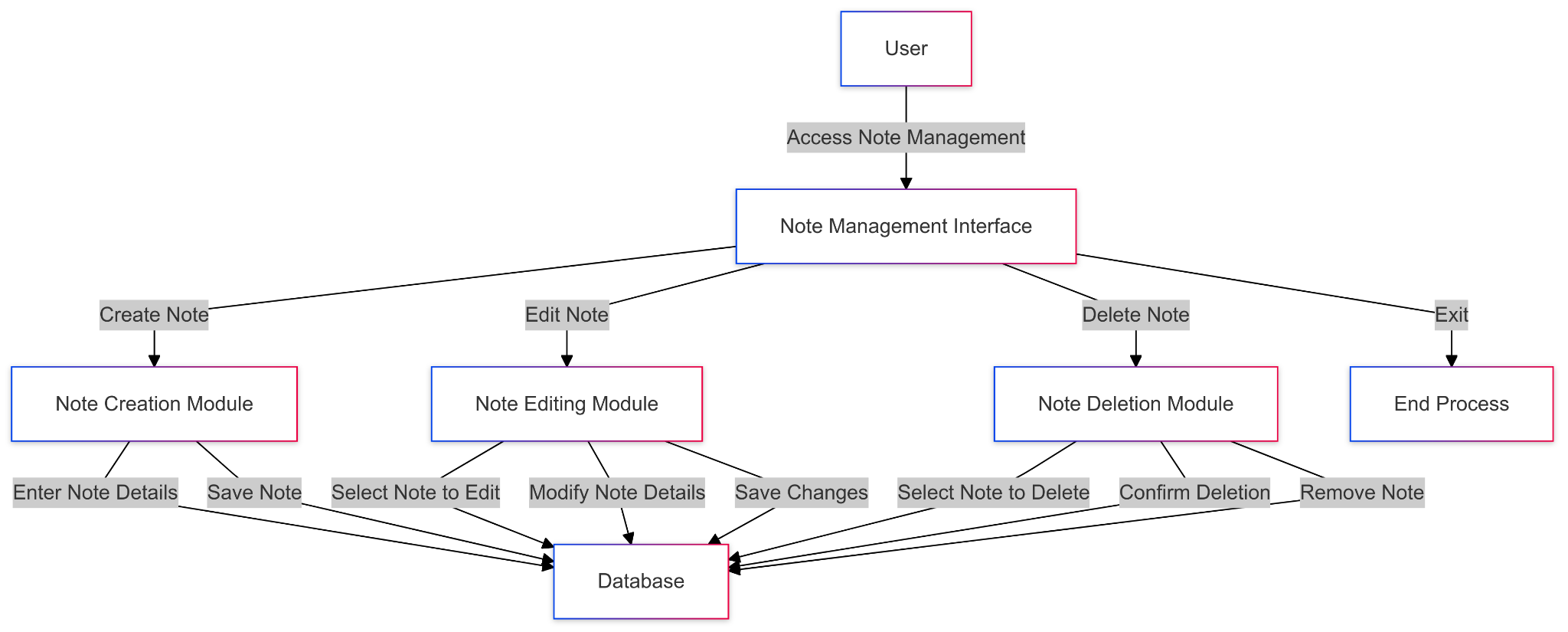
****

Figure-6 Note Creation and Management

### 3.2.2.4.2 [Note Visualization](https://docs.google.com/document/d/10Ip_r-EpnMTDK8pbAEiBnaQ0e1CXckSmUDyDe8mV0EE/edit?tab=t.0#heading=h.zh51quhqdr60)

**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-7 Note Visualization

### [3.2.2.5 Note Sharing](#_1fob9te)

**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.

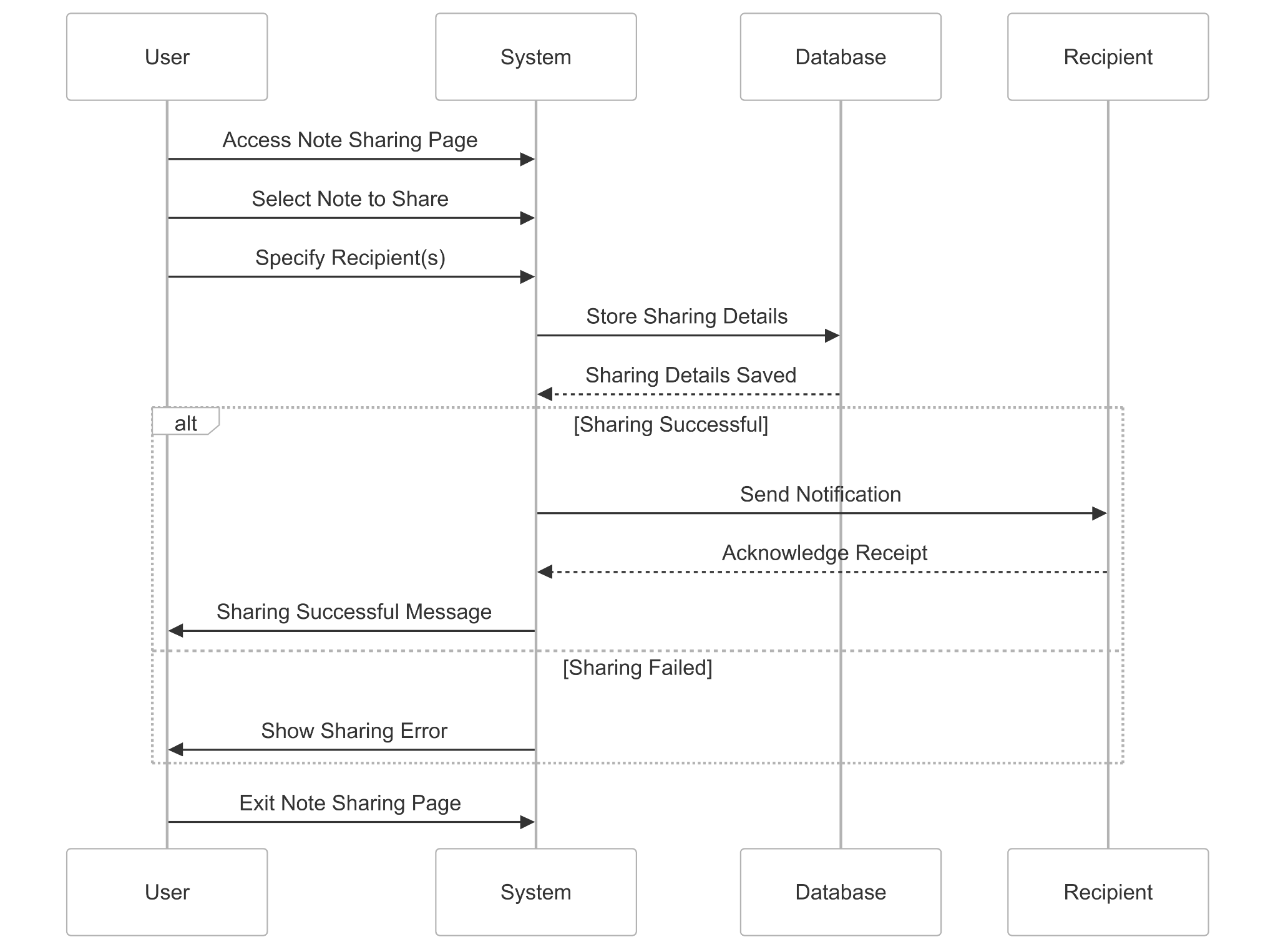


Figure-8 Note Sharing

### [3.2.2.6 Document Upload and Processing](#_3znysh7)

**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.

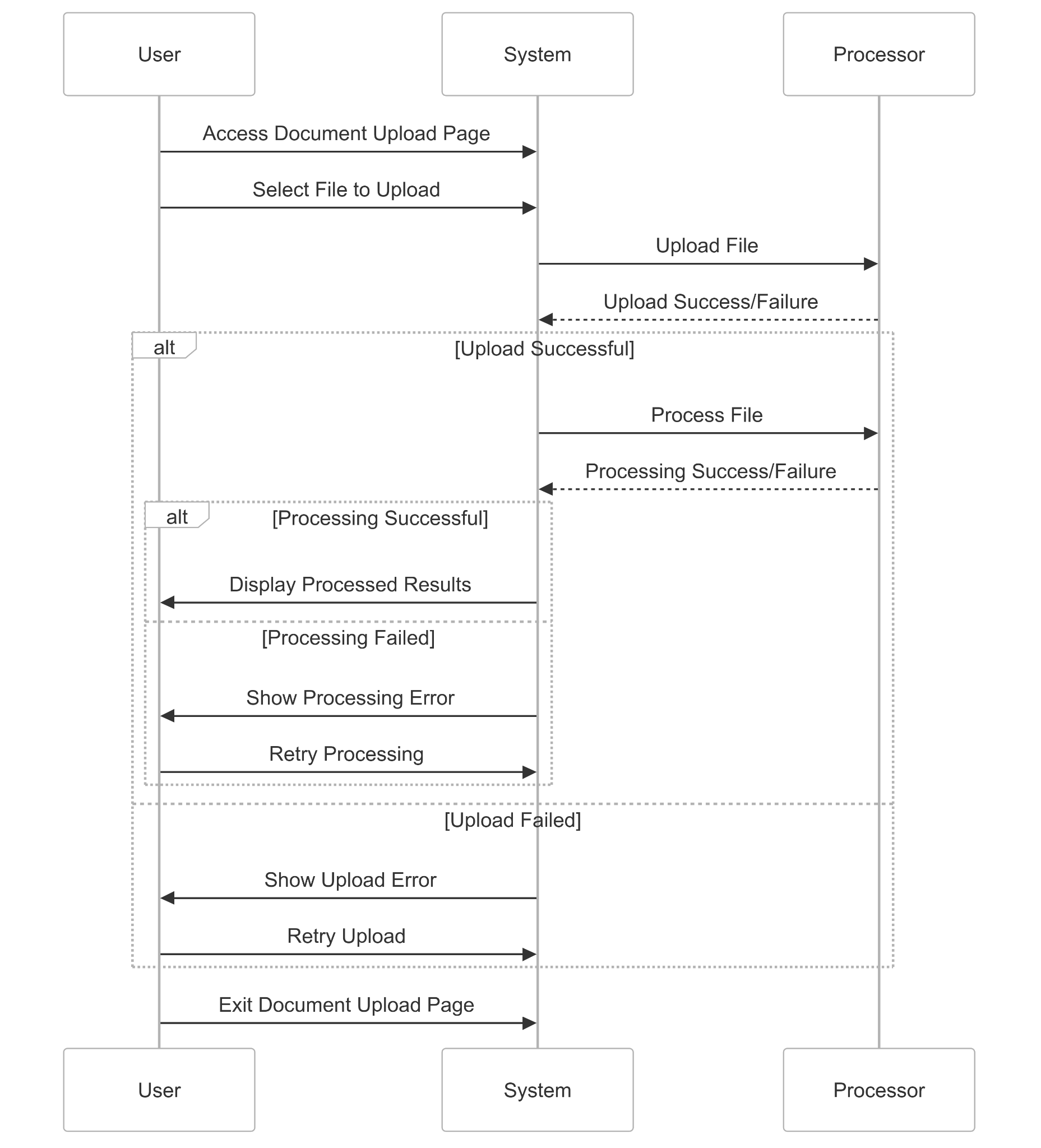
****

Figure-9 Document Upload and Processing

### [3.2.2.7 AI Chat Page (AI-powered question asking)](#_1baon6m)

**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.

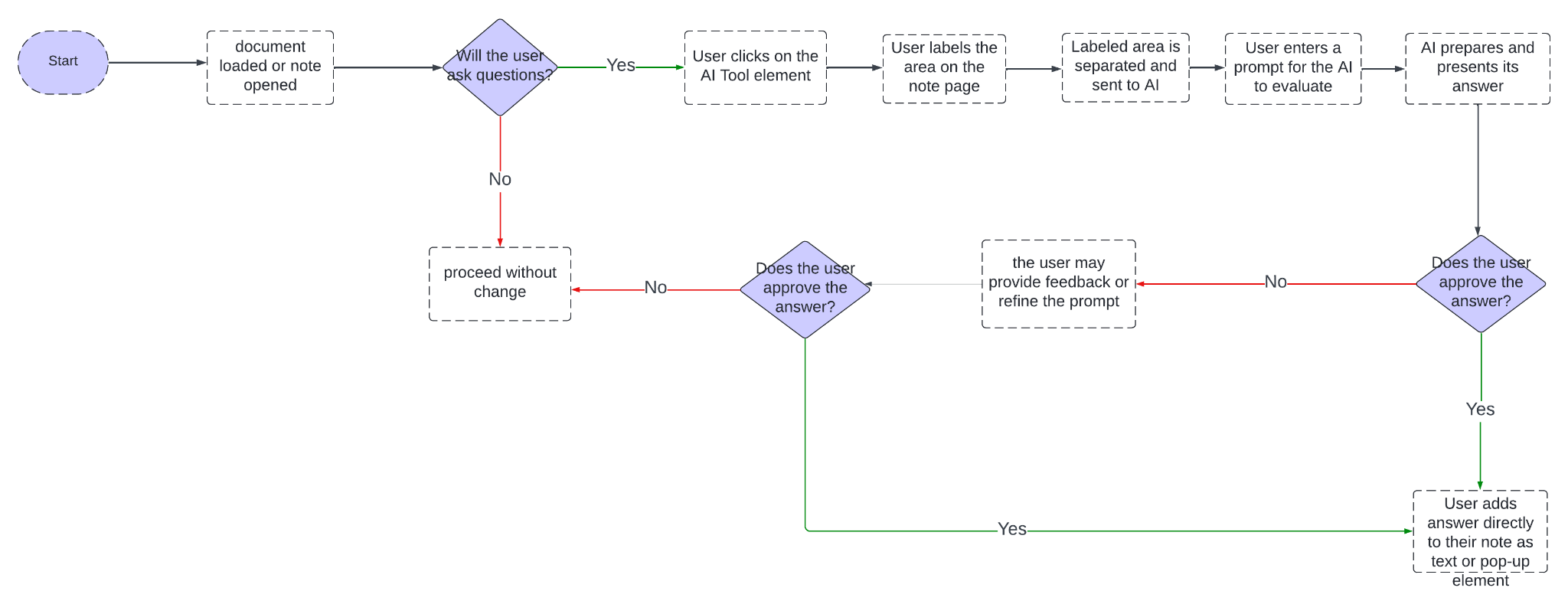


Figure-10 AI Chat Page(AI-powered question asking)

### [3.2.2.8 Forgot Password Pages](#_42ddq1a)

**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.

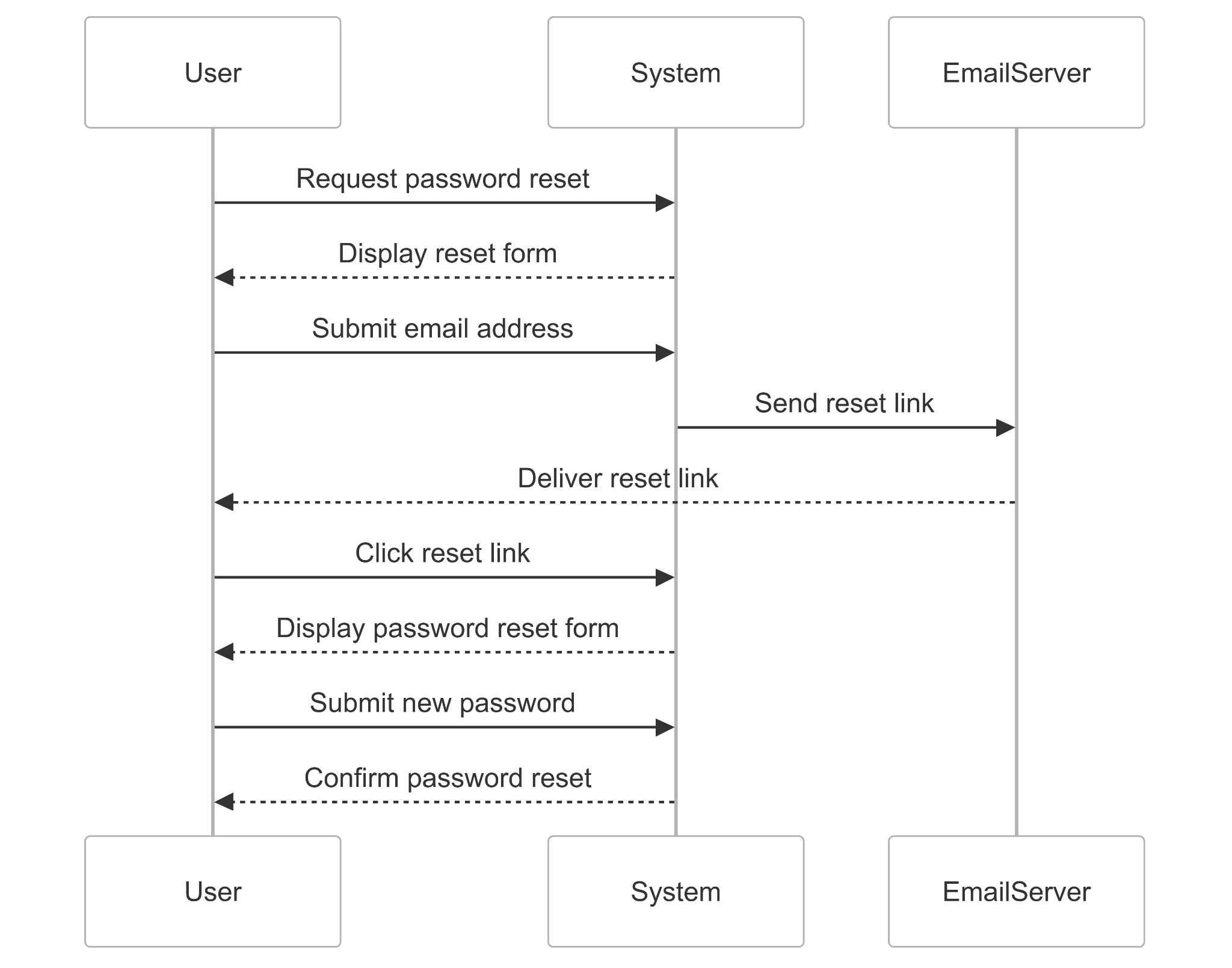


Figure-11 Forgot Passoword

## 3.2.3 Admin Pages

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

#### **3.2.3.1 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:**

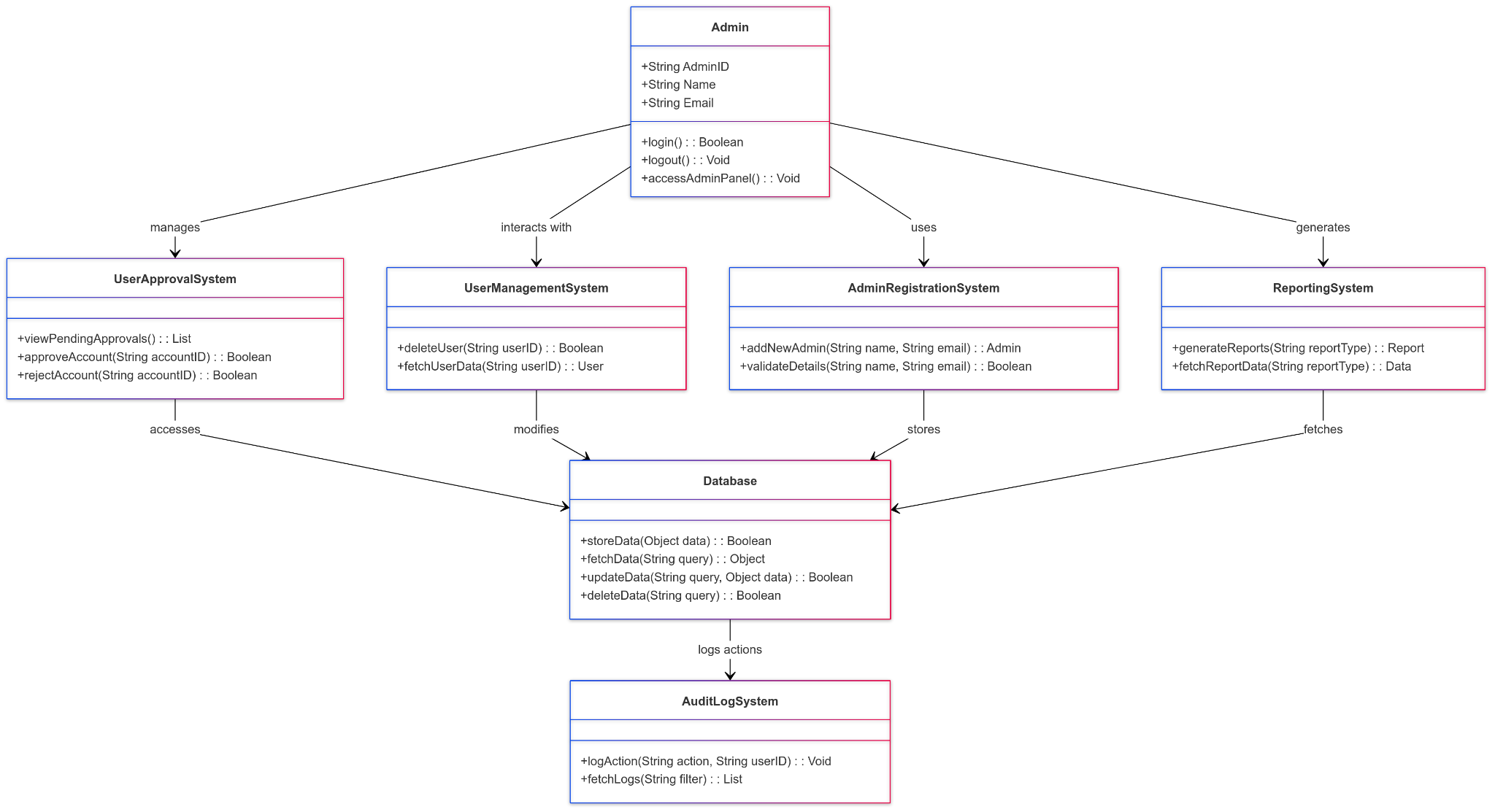
* **Start:** The admin navigates to the User Record page.
* **View Records:**
  + The system displays a list of registered users with details.
* **Search/Filter:**
  + The admin can search or filter user records based on criteria.
* **Action Options:**
  + The admin selects an action (e.g., view details, 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

### 3.2.3.1 User Record Page

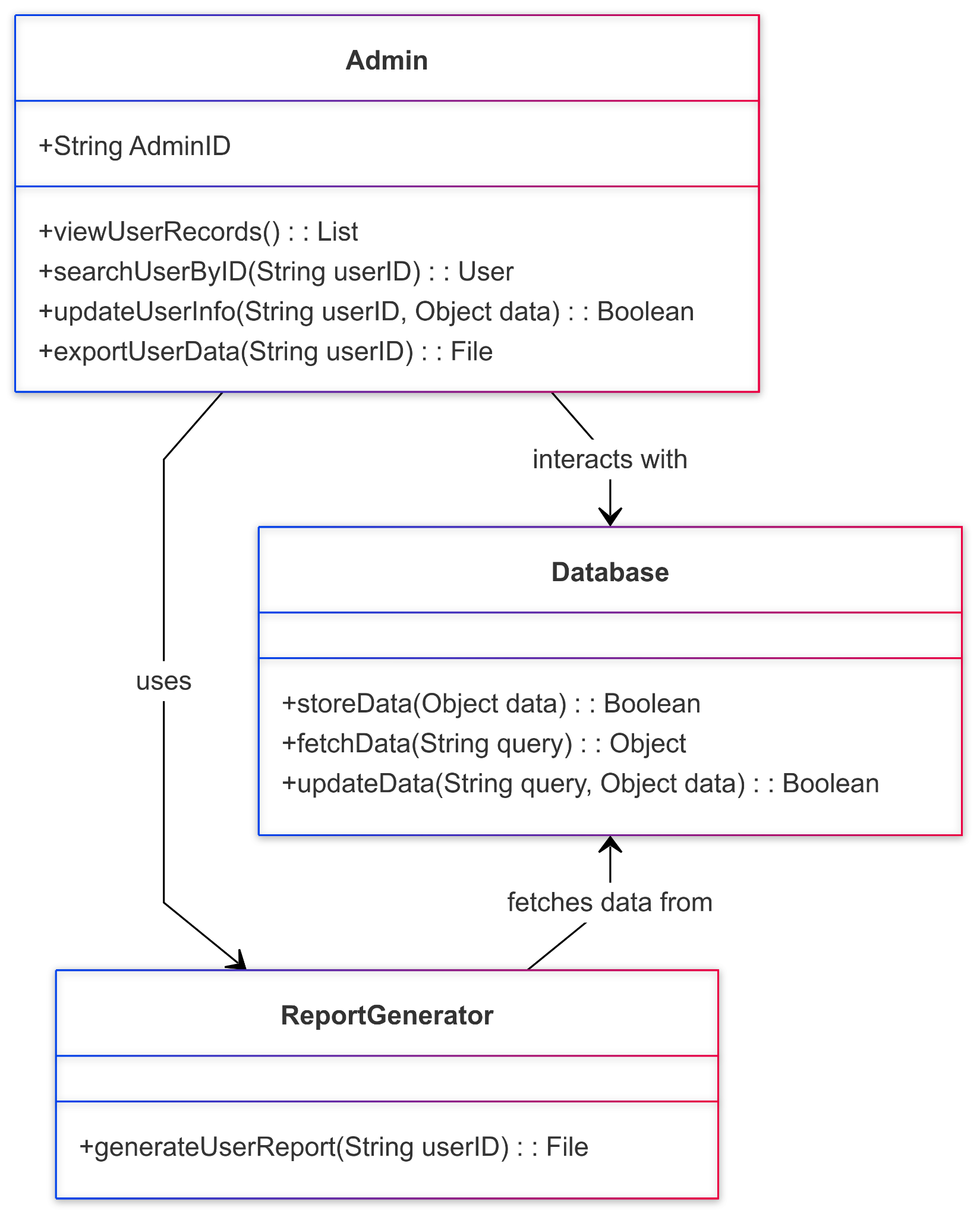


Figure-13 User Record Admin

**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.

### 3.2.3.2 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**:

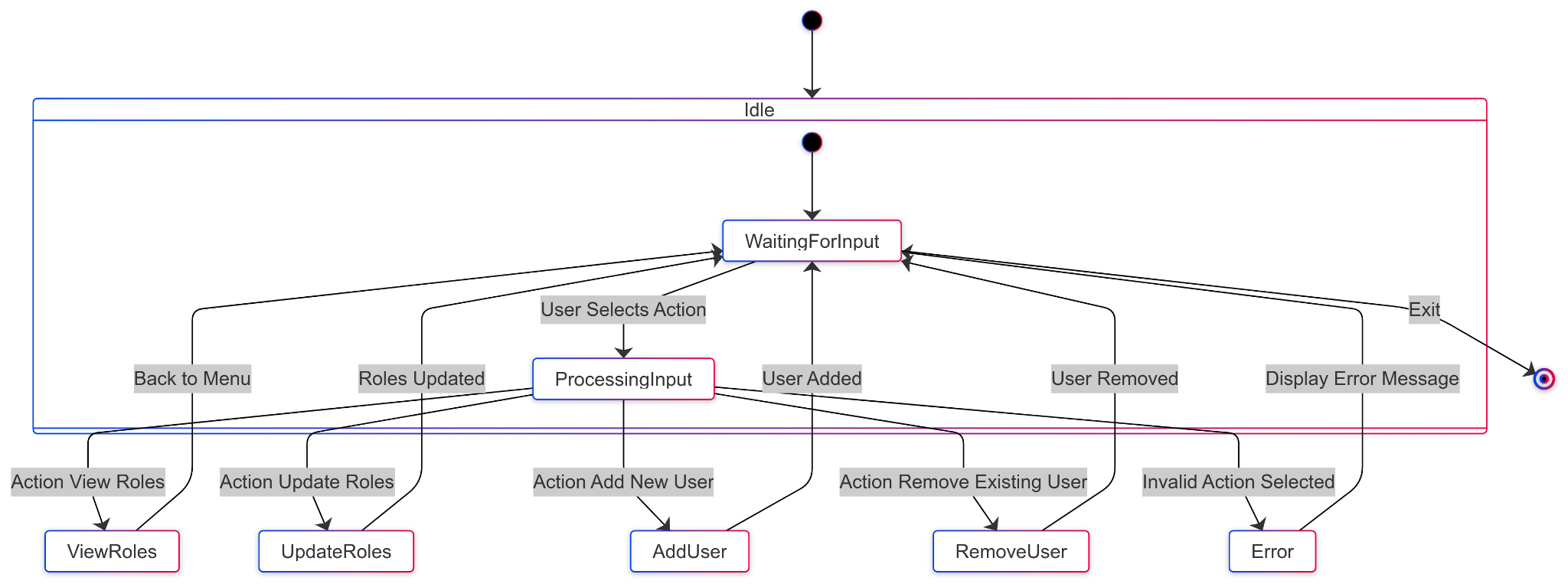
* Administrator must be logged into the system
* Administrator must have sufficient privileges
* System access control service must be available

**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

## 3.2.4 Settings Page

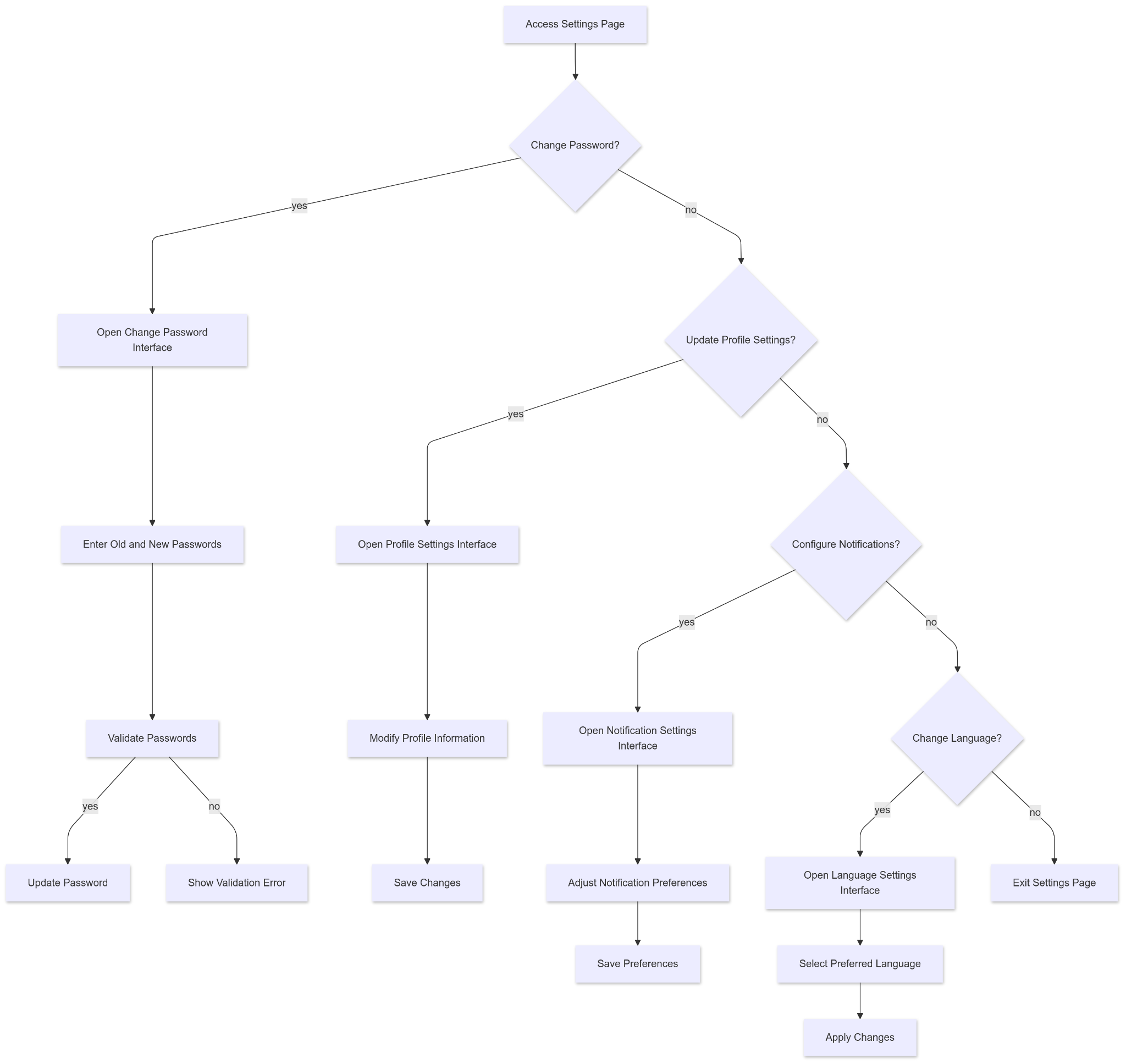


Figure-15 Settings Page

**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.

# User Interface

### 3.3.1 Login Page

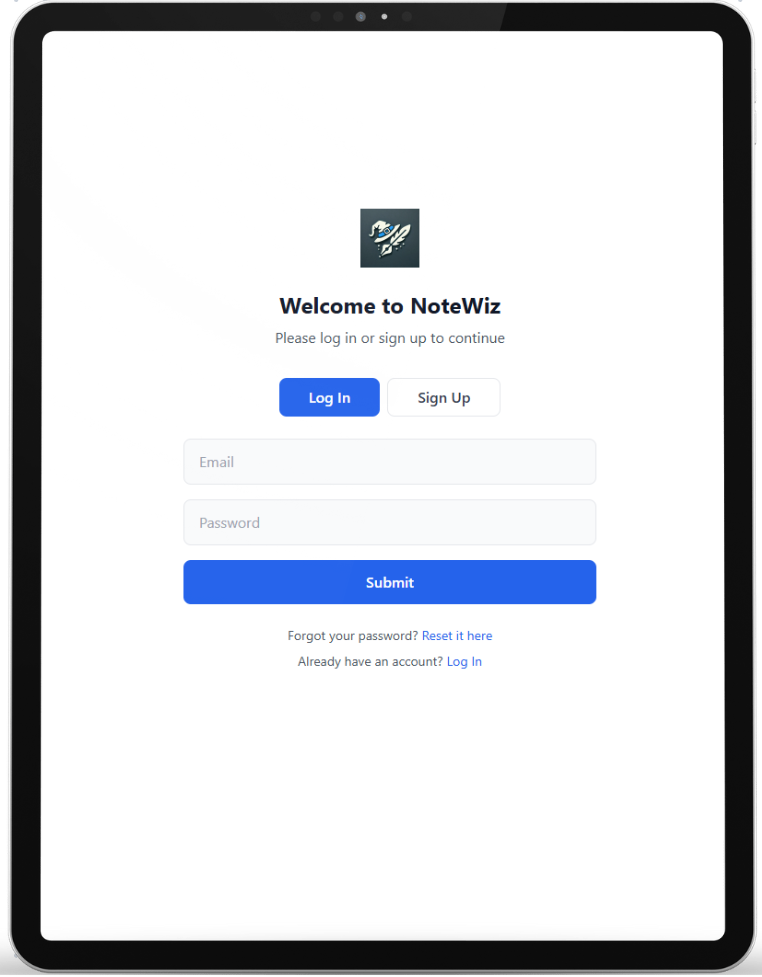


Figure -16 Login Page UI

| 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. |

## 

### 3.3.2 Sign Up Page

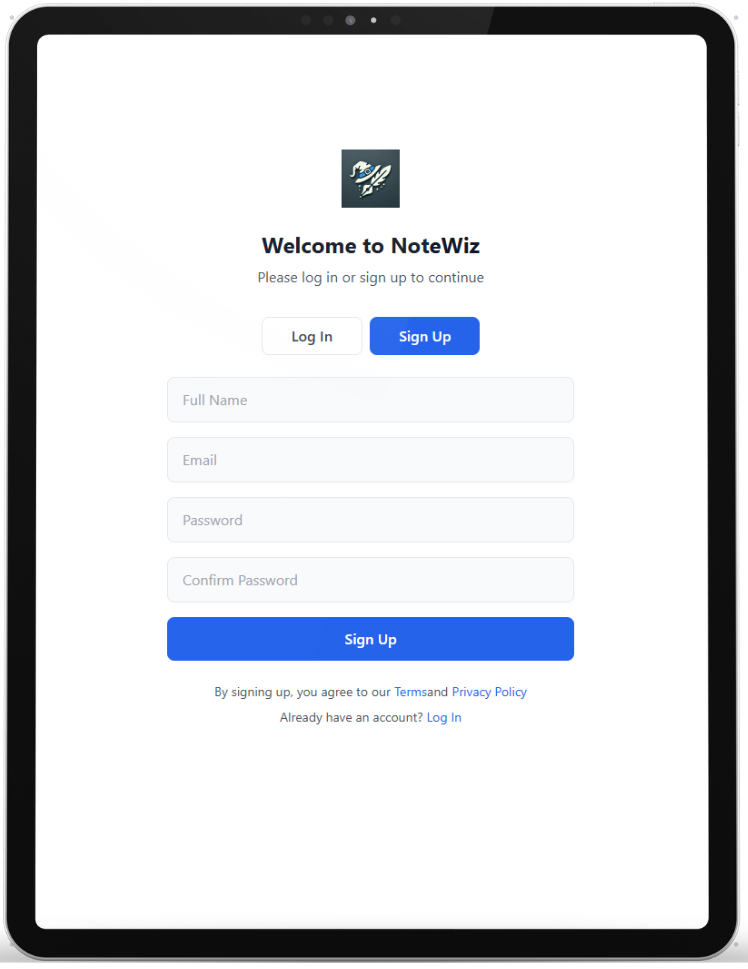


Figure -17 Sign Up Page UI

| 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 have read and agree to the term of service"*. 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: *"You already have an account? Sign in"* |

### 3.3.3 User Profile Management page

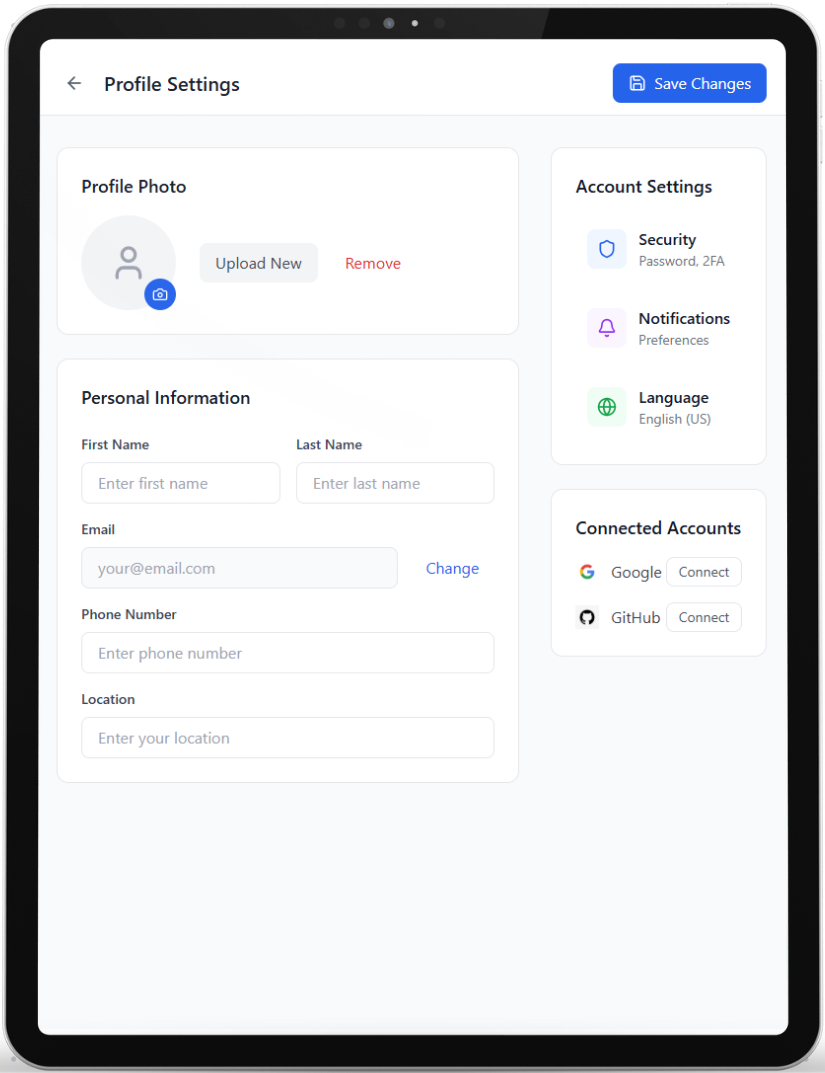


Figure -18 User Profile Management Page UI

| **Title** | User Profile Management |
| --- | --- |
| **Profile Picture Field** | Option to upload, view, or update the user’s profile picture. |
| **Name Field** | Field for editing the user’s display name. |
| **Email Field** | Editable field to update the user’s email address. Email format validation is applied. |
| **Password Field** | Option to reset or change the user’s password. |
| **Save Changes Button** | Button to save any modifications made to the user profile. |
| **Cancel Button** | Button to discard all changes and return to the previous menu. |
| **Security Button** | Button to go to security settings. |
| **Notification Button** | Button to go to notification settings. |
| **Language Button** | Button to go to language settings. |
| **Connected Accounts Field** | Field for connect other accounts with NoteWiz. |

### 3.3.4 AI Chat Page

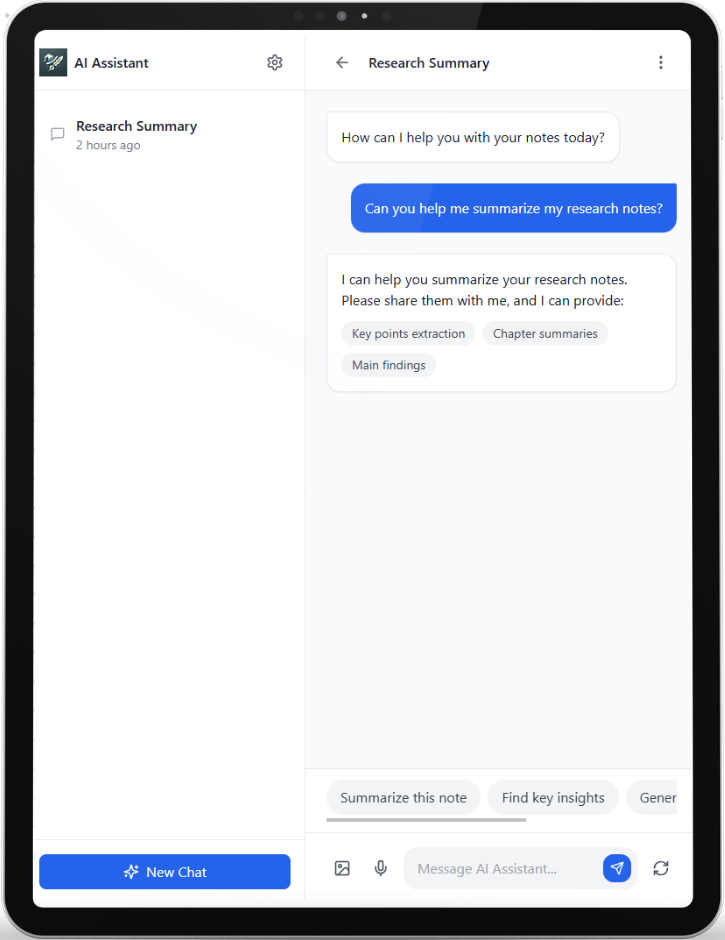


Figure -19 AI Chat Page UI

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

### 3.3.5 [Note Creation and Management](https://docs.google.com/document/d/10Ip_r-EpnMTDK8pbAEiBnaQ0e1CXckSmUDyDe8mV0EE/edit?tab=t.0#heading=h.xqwnf42cyjj0)

### 3.3.5.1 [Note Creation and Management](https://docs.google.com/document/d/10Ip_r-EpnMTDK8pbAEiBnaQ0e1CXckSmUDyDe8mV0EE/edit?tab=t.0#heading=h.xqwnf42cyjj0)

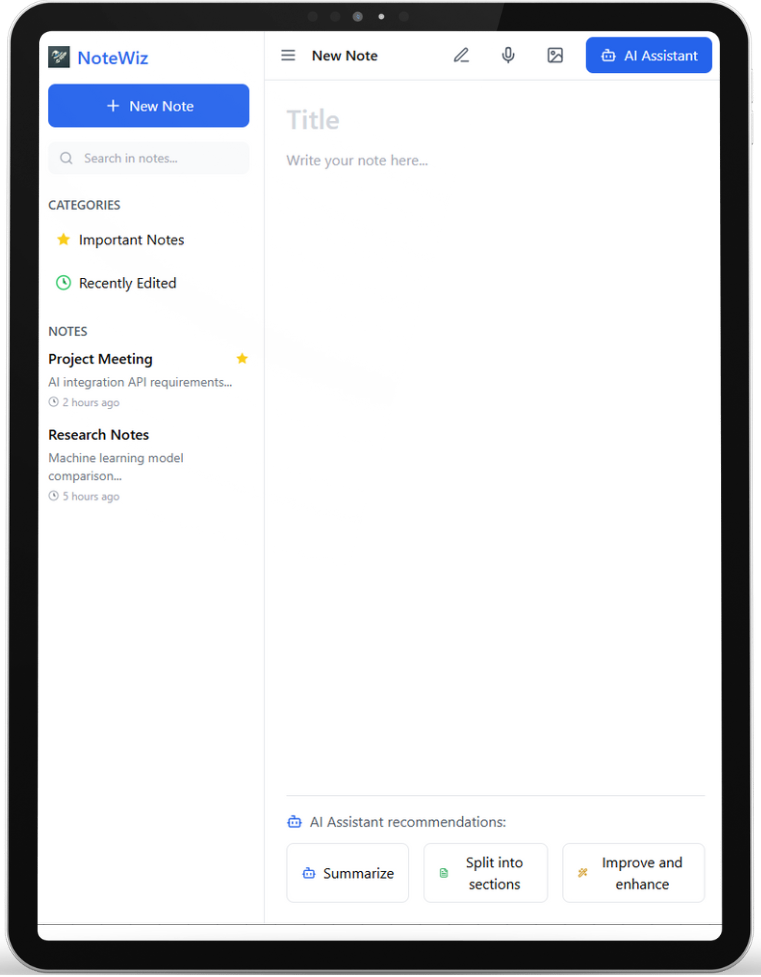


Figure -20 Note Creation and Manage Page UI

| **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. |
| **Cancel Button** | Button to discard the note and return to the previous screen. |

### 3.3.5.[2 Note](https://docs.google.com/document/d/10Ip_r-EpnMTDK8pbAEiBnaQ0e1CXckSmUDyDe8mV0EE/edit?tab=t.0#heading=h.xqwnf42cyjj0) visualization

### 

Figure -21 Note Visualization Page UI

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

### 

### 

### [3.3.6 Note Sharin](https://docs.google.com/document/d/10Ip_r-EpnMTDK8pbAEiBnaQ0e1CXckSmUDyDe8mV0EE/edit?tab=t.0#heading=h.1kwnpmv1pdo0)g

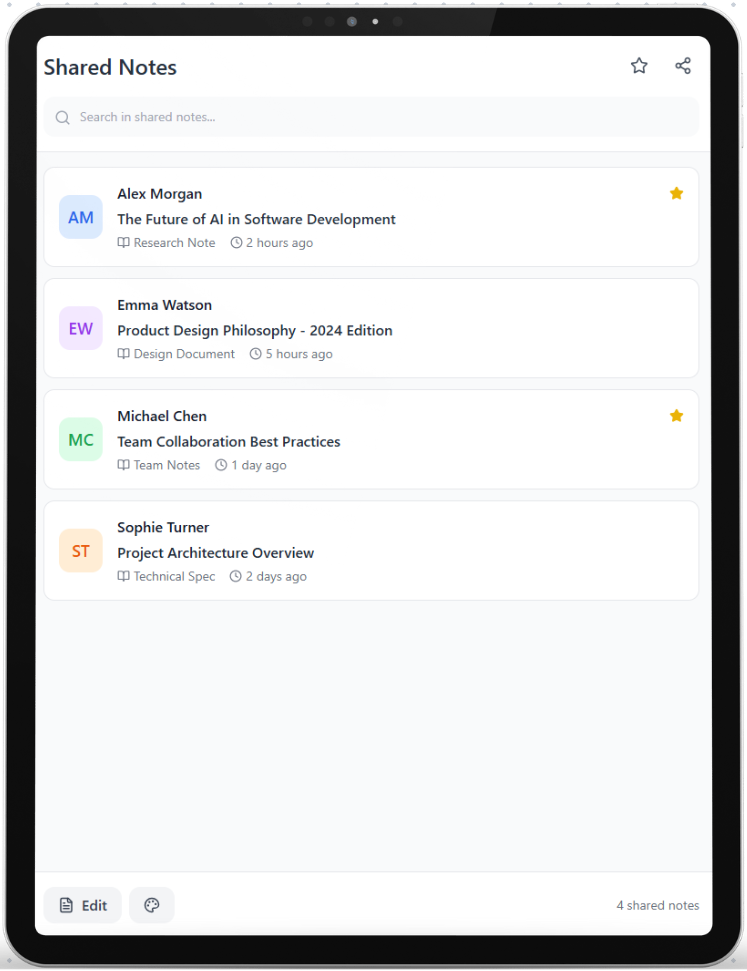


Figure -22 Note Sharinn Page UI

| **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. |

### 

### [3.3.7 Document Upload and Processing](#_1ksv4uv)

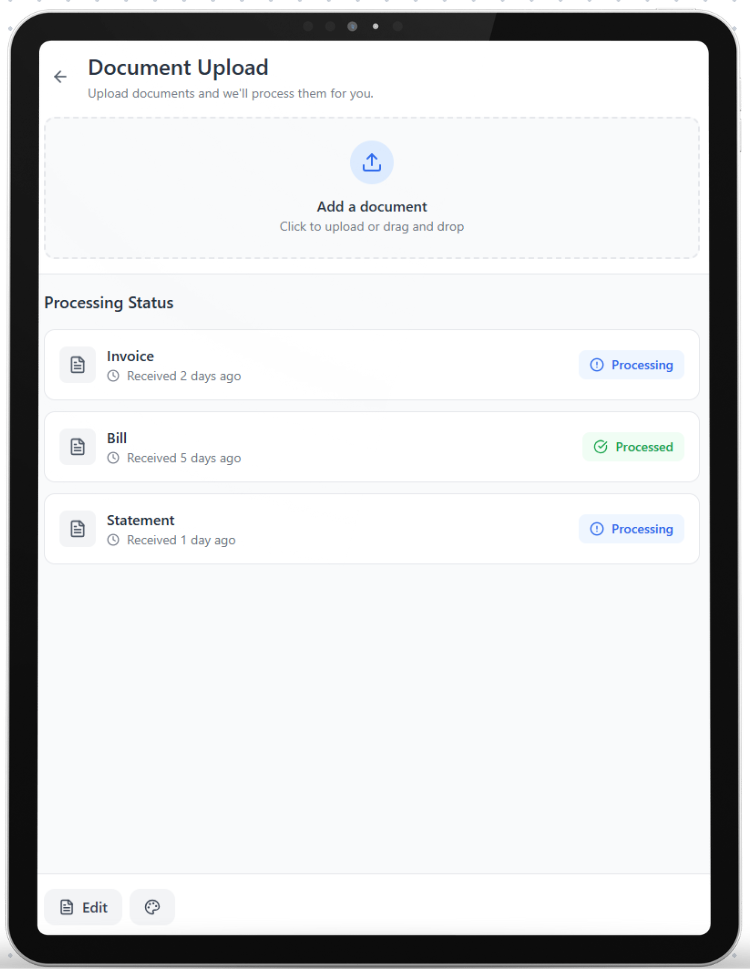


Figure -23 Document Upload and Processing Page UI

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

### [3.3.8 Admin Pages](#_44sinio)

### [3.3.8.1 User Record Page](#_wnyagw)

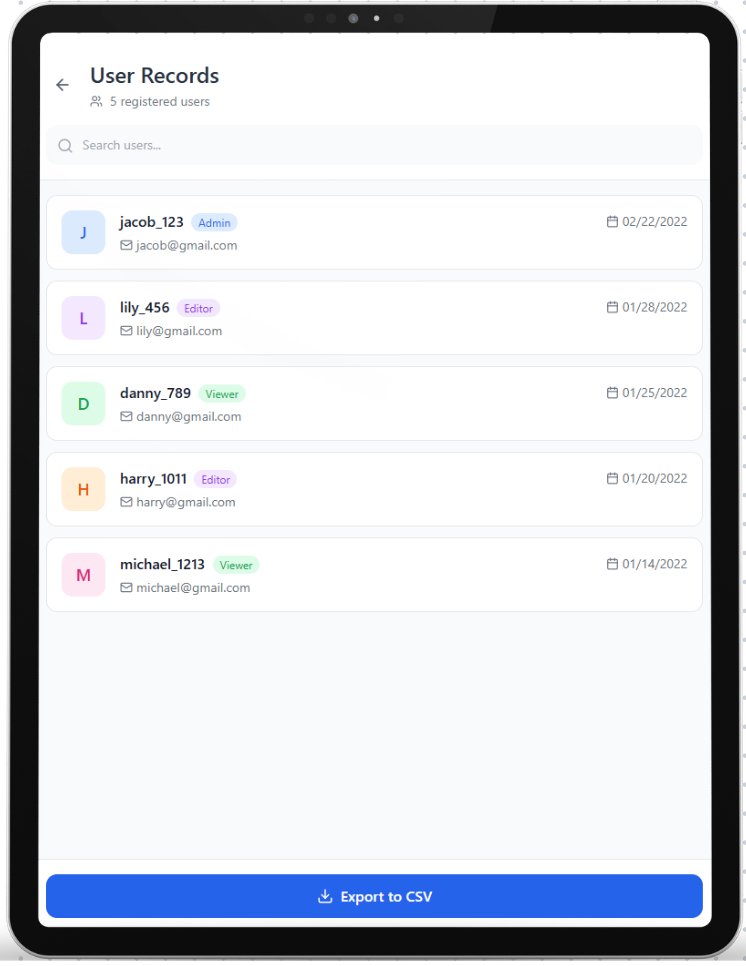


Figure -24 User Record Page UI

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

### *3*[*.3.8.2* User Access Control Page](#_45jfvxd)

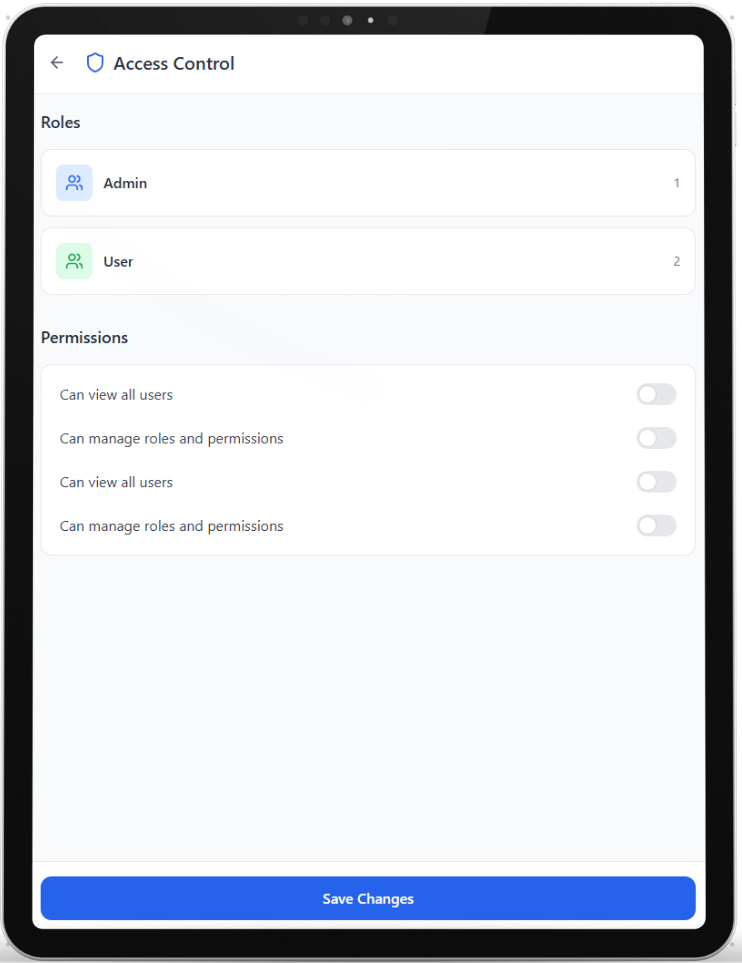


Figure -25 User Access Control Page UI

| **Title** | Access Control |
| --- | --- |
| **Role Management** | Allows admins to assign or modify user roles, such as admin or regular user. |
| **Add User Button** | Button to register a new user in the system. |
| **Remove User Button** | Option to remove a user from the system with necessary confirmations. |
| **Save Changes Button** | Button to apply and save all changes to user roles and permissions. |

### 

### [3.3.9 Forgot Password Pages](#_3gnlt4p)

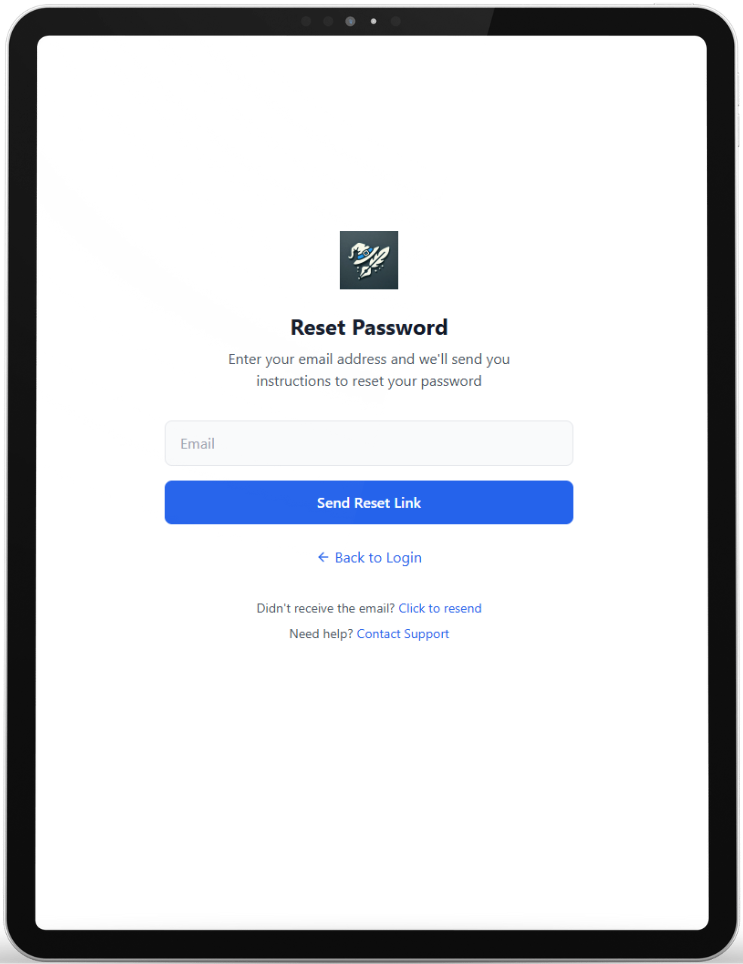


Figure -26 Forgot Password Page UI

| Title | Forgot Password |
| --- | --- |
| New Password Field | Field where the user will enter new password. |
| Verify Password Field | Field where the user will verify new password. |
| Reset Password Button | Button for resetting password. |
| Explanation Part | Explain the new password constraints. |

## 

**3.3.10 Settings Page**

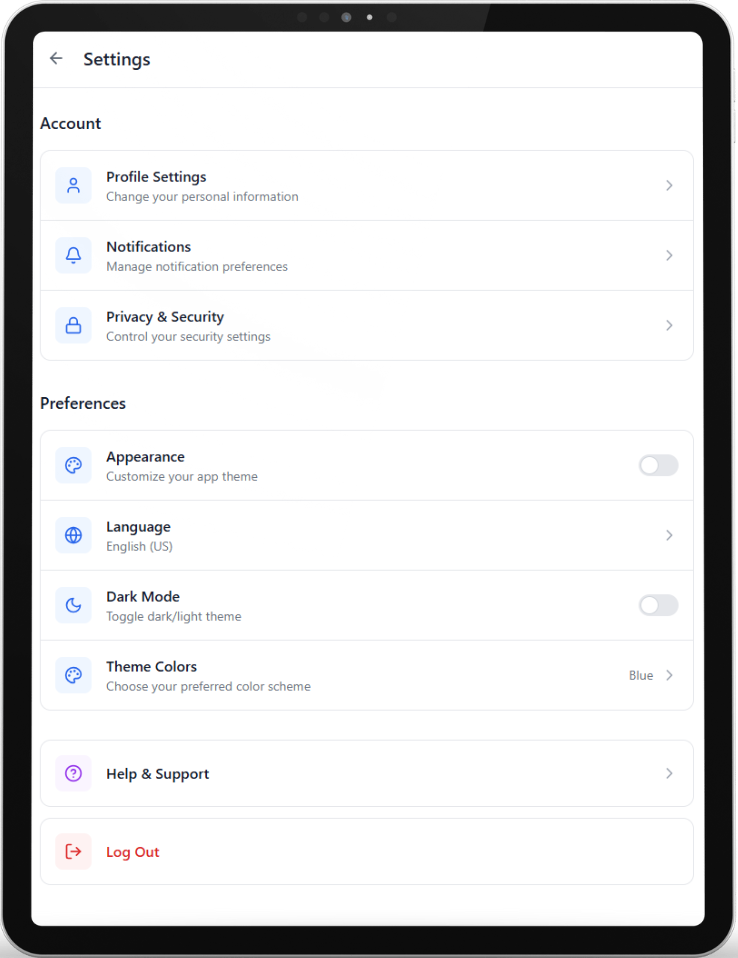


Figure -27 Settings Page UI

| **Title** | **Settings** |
| --- | --- |
| **Account Settings** | Allows users to update their username, email, or password. |
| **Theme Preferences** | Toggle between light, dark and other themes for the application. |
| **Notification Settings** | Options to enable or disable specific notifications (e.g., email, in-app alerts). |
| **Privacy Settings** | Controls for data sharing permissions and visibility settings. |
| **Save Changes Button** | Button to confirm and save updates to user settings. |

## 3.4 Software System attributes

### 3.4.1. 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.

### 3.4.2. 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.

### 3.4.3. 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.

### 3.4.4. 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.

### 3.4.5 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.

### 3.4.6 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.

### 3.4.7 Maintainability

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

# 4 References

| [1] | "React - A JavaScript library for building user interfaces," React Official Documentation, [Online]. Available:<https://react.dev/>. [Accessed 27 Dec. 2024]. |
| --- | --- |

[2] Firebase Realtime Database, Google Firebase Documentation. [Online]. Available: https://firebase.google.com/docs/database/. [Accessed: 27-Dec-2024].