**ÇANKAYA UNIVERSITY**

**COMPUTER ENGINEERING DEPARTMENT**

**CENG 407**

**SOFTWARE REQUIREMENTS SPECIFICATION**

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

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

**Date: 06/12/2024**

**Table of Contents**

[**1. INTRODUCTION.....................................................................................................................**](#_aydznsek4hxp)**4**

[**1.1 Purpose**](#_9zpqoq9gd5ky) **.................................................................................................................................4**

[**1.2 Scope of Project....................................................................................................................**](#_vuz1rm799x92)**4**

[**1.3 Glossary…............................................................................................................................**](#_2xhip8e3zgp7)**5**

[**1.4 References ............................................................................................................................6**](#_vqf1pqc7q4b)

[**1.5 Overview of the Document..................................................................................................**](#_36z3uydicq33)**6**

[**1.6 Version History………….....................................................................................................7**](#_9fpxq93uyvh7)

[**2. OVERALL DESCRIPTION.....................................................................................................7**](#_uvd87d2tqno)

[**2.1 Product Perspective.............................................................................................................**](#_u5m369gdwnyt)**7**

[**2.1.1. Development Methodology ..........................................................................................**](#_5leqd7yogale)**7**

[**2.2 User Characteristic..............................................................................................................**](#_97irm35a438m)**8**

[**2.2.1. Participants…................................................................................................................**](#_b2e4z13ofpwk)**8**

[**2.2.1.1 What can User do?...................................................................................................**](#_idijcwjhoqjf)**8**

[**2.2.2. Admin…………….........................................................................................................9**](#_vmldodgfne3f)

[**2.2.1.1 What can Admin do?.....................................................................................**](#_8lu8yuhkiu95)**.........9**

[**2.3 Operations…………………………………………………………………………………..**](#_v0wc3dh4wjpe)**9**

[**3. REQUIREMENTS SPECIFICATION..................................................................................**](#_tvtylbd1z8mt)**10**

[**3.1 External Interface Requirements.....................................................................................**](#_ljrxorq2hx46)**10**

[**3.1.1. User interfaces.............................................................................................................10**](#_se5qphe3sge6)

[**3.1.2. Hardware interfaces ..................................................................................................14**](#_uxcv07fwbl5n)

[**3.1.2.1 Output Devices**](#_1hmsyys)**……………………………………………………………………15**

[**3.1.2.2 Network Connectivity**](#_41mghml)**…………………………………………………………….15**

[**3.1.3. Software interfaces**](#_7fadcjtsin6y) **.....................................................................................................15**

[**3.1.4. Communications interfaces**](#_76pv9xbkcqa2) **.......................................................................................15**

[**3.2 Functional Requirements ..................................................................................................16**](#_v863aznau7uz)

[**3.2.1. NoteWiz User Authentication and Account Management**](#_m6d3n11rs1dh)**………..……………….17**

[**3.2.1.1 Log In and Sign up**](#_l29b122gr4ba)**…………………………………………………………….17**

[**3.2.1.2 Forgot/Change password**](#_7ixo5slvyxpr)**……………………………………………………...19**

[**3.2.2 Note Creation and Management**](#_xqwnf42cyjj0)**……………………………………………………..21**

[**3.2.3 AI-powered question asking**](#_ri9b3w6qiz94)**………………………………………………………….22**

[**3.2.4 Task Management**](#_6oj5m6tj2wx6)**…………………………………………………………………….24**

[**3.2.5 Cross-Platform Accessibility**](#_xkkejpy73azk)**…………………………………………………………24**

[**3.2.6 Note Sharing**](#_1kwnpmv1pdo0)**…………………………………………………………………………..25**

[**3.2.7 Document Upload and Processing…………………………………………………...25**](#_v241qnz7zrxj)

[**3.2.8 Note Creation and Visualization……………………………………………………..26**](#_zh51quhqdr60)

[**3.2.9 User Authentication and Security**](#_t85ydcz5lu3u)**…………………………………………………...27**

[**3.3 Software System attributes................................................................................................**](#_k3h0vvogx9xd)**27**

[**3.3.1. Portability .....................................................................................................................**](#_qamvjohp5lcs)**27**

[**3.3.2. Usability**](#_i0iwbasmdmtn) **........................................................................................................................27**

[**3.3.3. Adaptability**](#_q751qibs0qiu) **..................................................................................................................27**

[**3.3.4. Security**](#_6pvcwl9eydla)**..........................................................................................................................27**

[**3.3.5 Reliability**](#_f3b0w3rmnfhc)**………………………………………………………………………………27**

[**3.3.6 Availability**](#_7zoxo45mfuow)**…………………………………………………………………………….28**

[**3.3.7 Maintainability**](#_cby0f7a5il8m)**……………………………………………………………………….28**

[**3.4 Safety Requirement ............................................................................................................**](#_ayol408bys0h)**28**

[**4. REFERENCES…….................................................................................................................**](#_5tuyu5l6i4rr)**28**

#### **1.INTRODUCTION**

In recent years, advances in artificial intelligence (AI) and natural language processing (NLP) have led to the emergence of innovative applications in various fields such as education, productivity, and personal organization. One of these areas that has benefited greatly from AI integration is digital note-taking. While traditional note-taking tools aim to help users organize their thoughts and create structured content, the new generation of AI-powered applications have broken through these boundaries.

Note-taking applications now not only help create content, but also offer the ability to ask questions with AI support. Users can access their accounts from multiple devices and share their notes with other users. Leveraging these innovations, the NoteWiz project aims to use professionally integrated AI APIs to help users achieve the most successful results in their chosen fields. This approach allows comparing various AI models and determining the most suitable model and output. The API-based solution increases the scalability of the project, improves the user experience, and provides a flexible infrastructure.

A versatile and multifunctional note-taking tool designed for enhanced productivity and collaboration, NoteWiz aims to increase user productivity by combining advanced AI functions such as AI support, task scheduling, and multi-device access features in a distraction-free, minimalist interface.

#### **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.

#### **1.2 Scope of the 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.
* **İnteraction 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.

#### **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. |

#### **1.4 References**

See 4 references section.

#### **1.5 Overview of the 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.

#### **1.6 Version History**

| 1.0 | Initial Release |
| --- | --- |
| 1.1 | Corrections and additions have been made |

# **2. OVERALL DESCRIPTION**

#### **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.

#### **2.1.1. 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:

1. **Sprint Planning**: Tasks are identified and prioritized based on importance and customer requirements.
2. **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.
3. **Review & Feedback**: At the end of each sprint, the work completed is reviewed and feedback is provided to ensure the project is on track.
4. **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.

#### **2.2 User Characteristic**

##### **2.2.1. 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.

##### **2.2.1.1 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.

#### **2.2.2. 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.

##### **2.2.2.1 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.

#### **2.3 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.

# **3. REQUIREMENTS SPECIFICATION**

#### **3.1 External Interface Requirements**

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

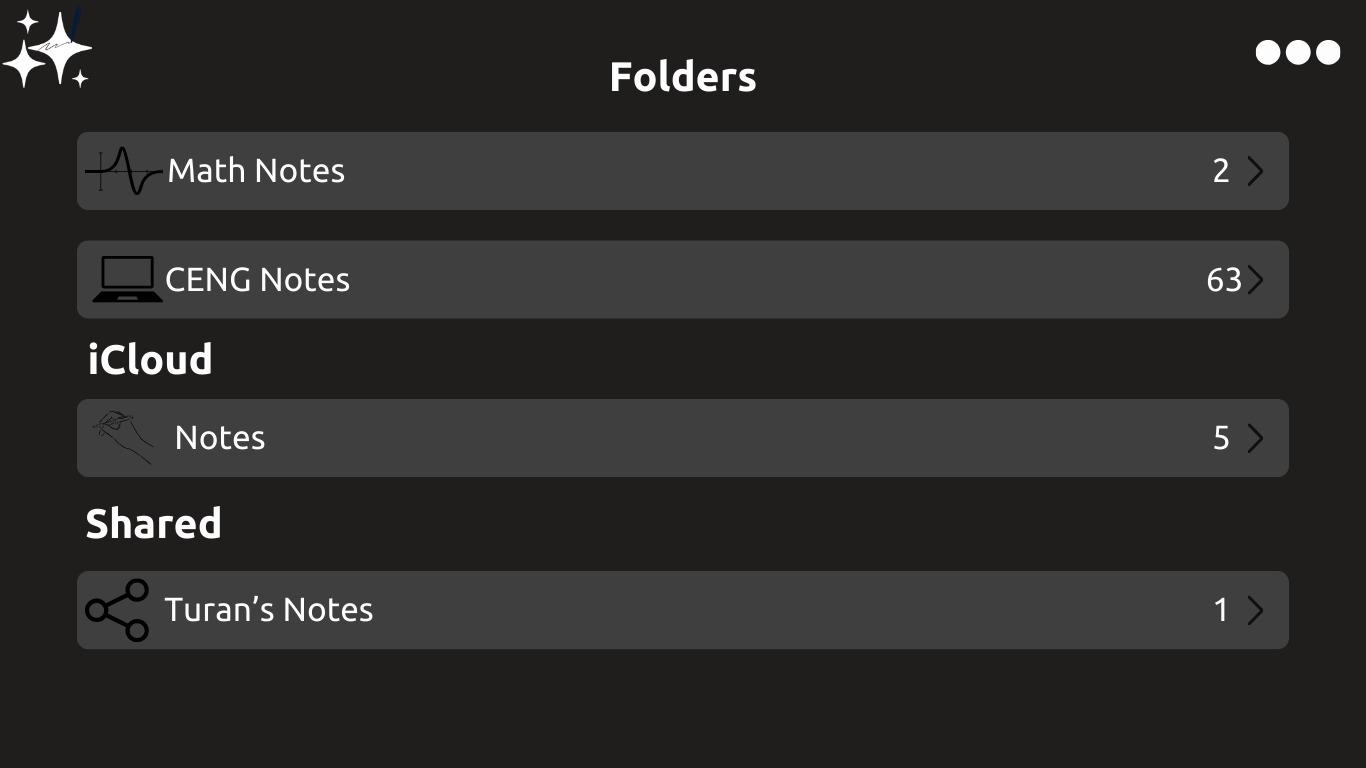
* **Home Screen**: Provides quick access to recently opened notes, tasks, and collaborative projects. ***Figure 2, Figure 3, Figure 4***
* **Login:** allows users to securely log in to their NoteWiz account using their credentials (email and password).***Figure 1***
* **Sign Up:** New users can quickly sign up for an account and create a new user profile by entering their email and password.***Figure 1***
* **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 1***
* **Note Creation**: Users can easily create and edit notes with a minimalist text editor. ***Figure 4,Figure 3***
* **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 2***
* **Settings**: Allows users to customize the app's appearance (e.g., Light/Dark Mode), manage their account settings, and adjust notification preferences. ***Figure 5***
* **Interaction with AI interface**

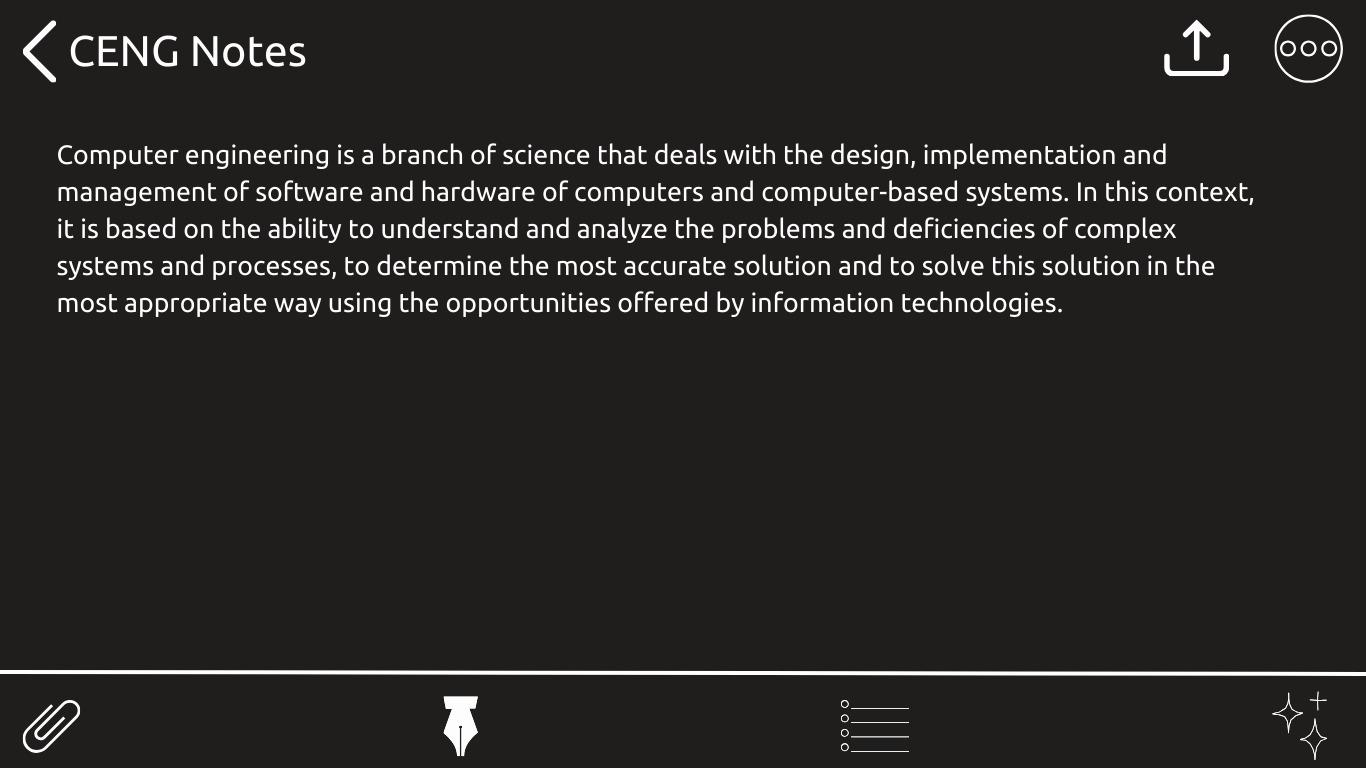
1. NoteWiz allows the user to click on the AI ​​Tool item, label the desired area on the note page, direct it to the artificial intelligence and ask questions, and if the user approves the answer, it can be added to the labeled area as a pop-up item or added directly to the note as text. ***Figure 6,Figure 7***

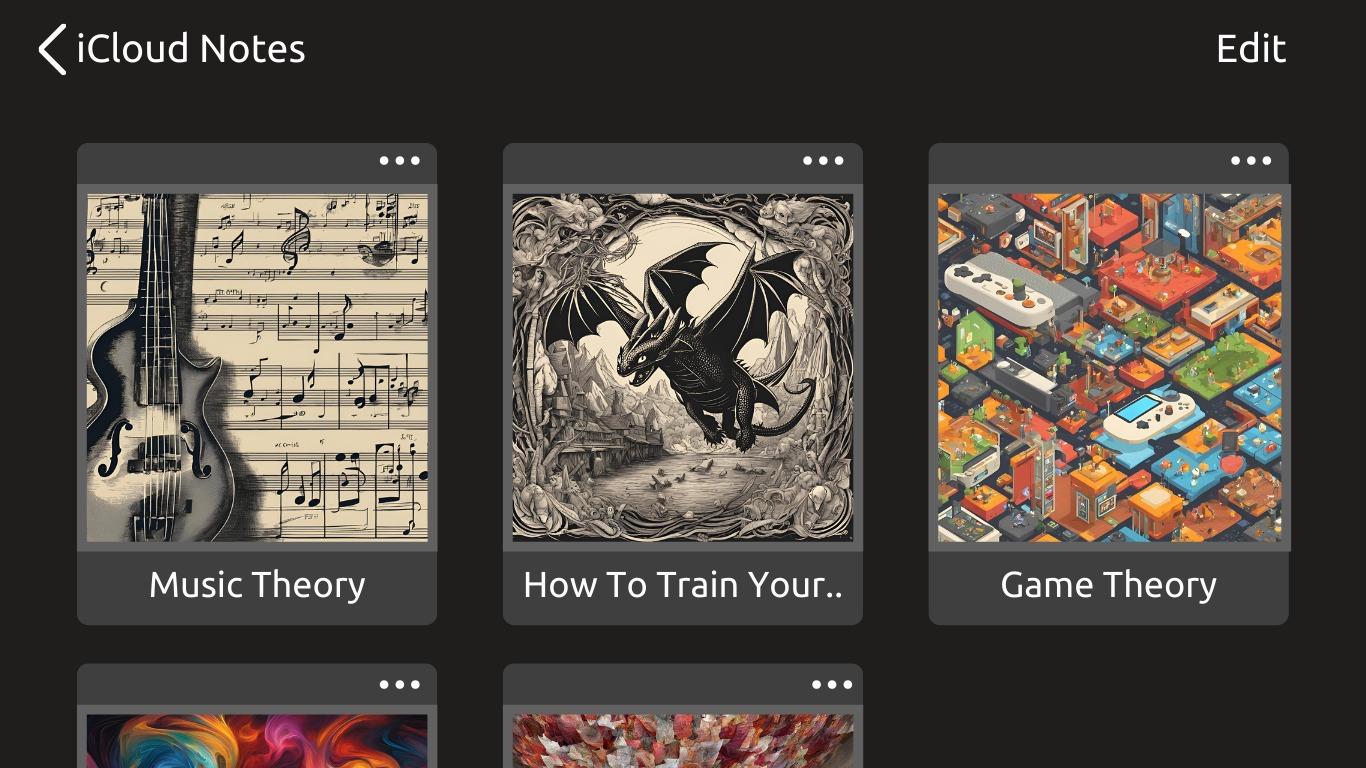
* **Upload Document:** Allows users to upload various document types (e.g., PDF, DOCX, images) for processing. The uploaded files can be used for AI-powered summarization, note creation, or question-answering. ***Figure 7***
* **Note Visualization:** When users create a new note, the system automatically 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 4***

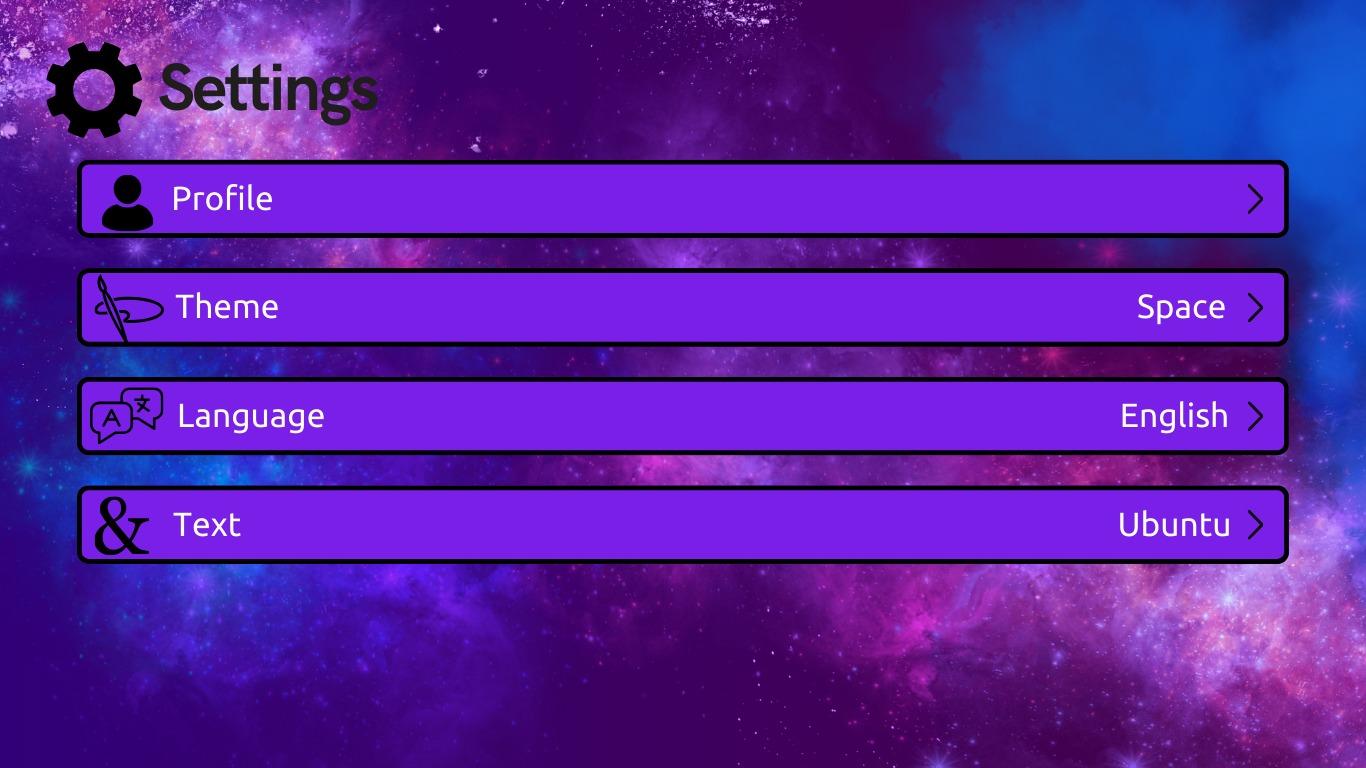
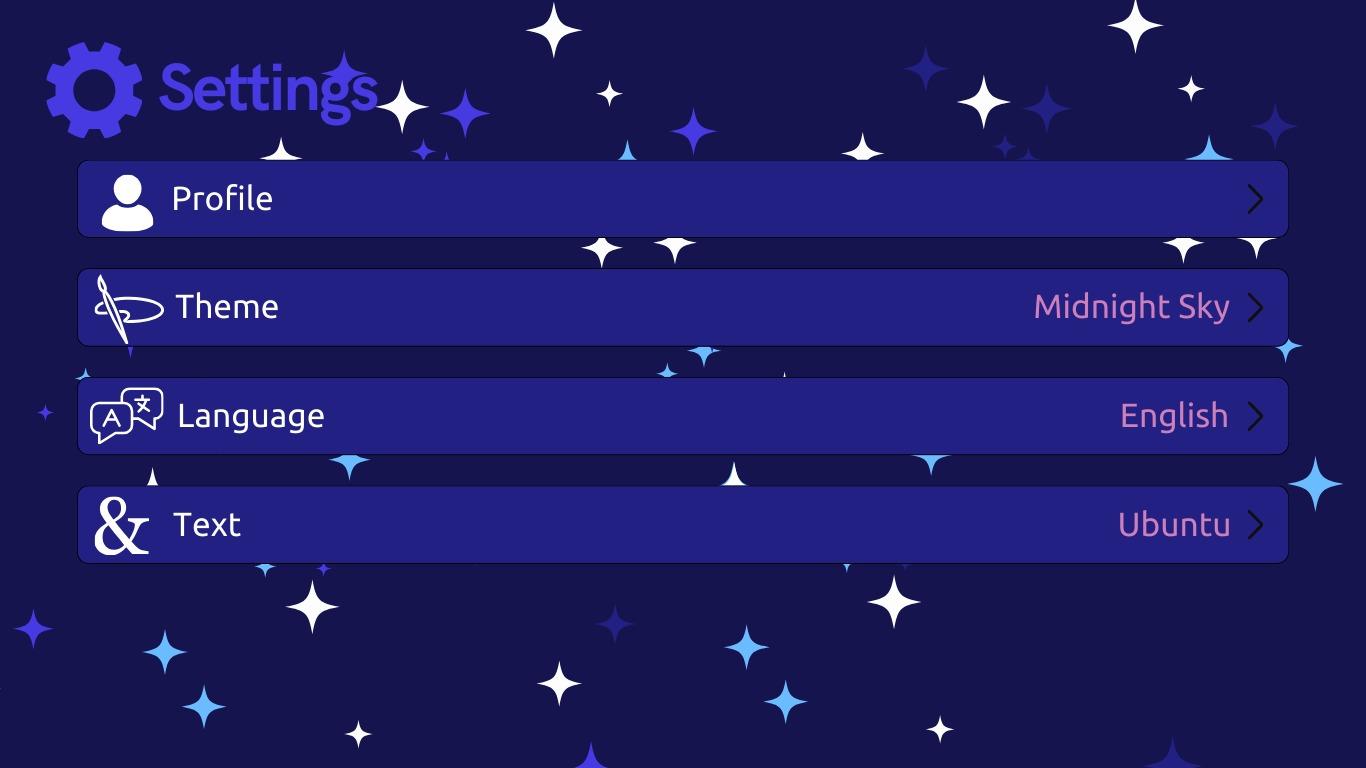
These features will be seamlessly integrated into the **NoteWiz** interface, providing users with powerful tools for quick content review and detailed question answering. The goal is to make it as intuitive as possible for users to access these advanced AI features while maintaining a clean and distraction-free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

** *Figure 1-Log In-Sign up-Forgot your password***

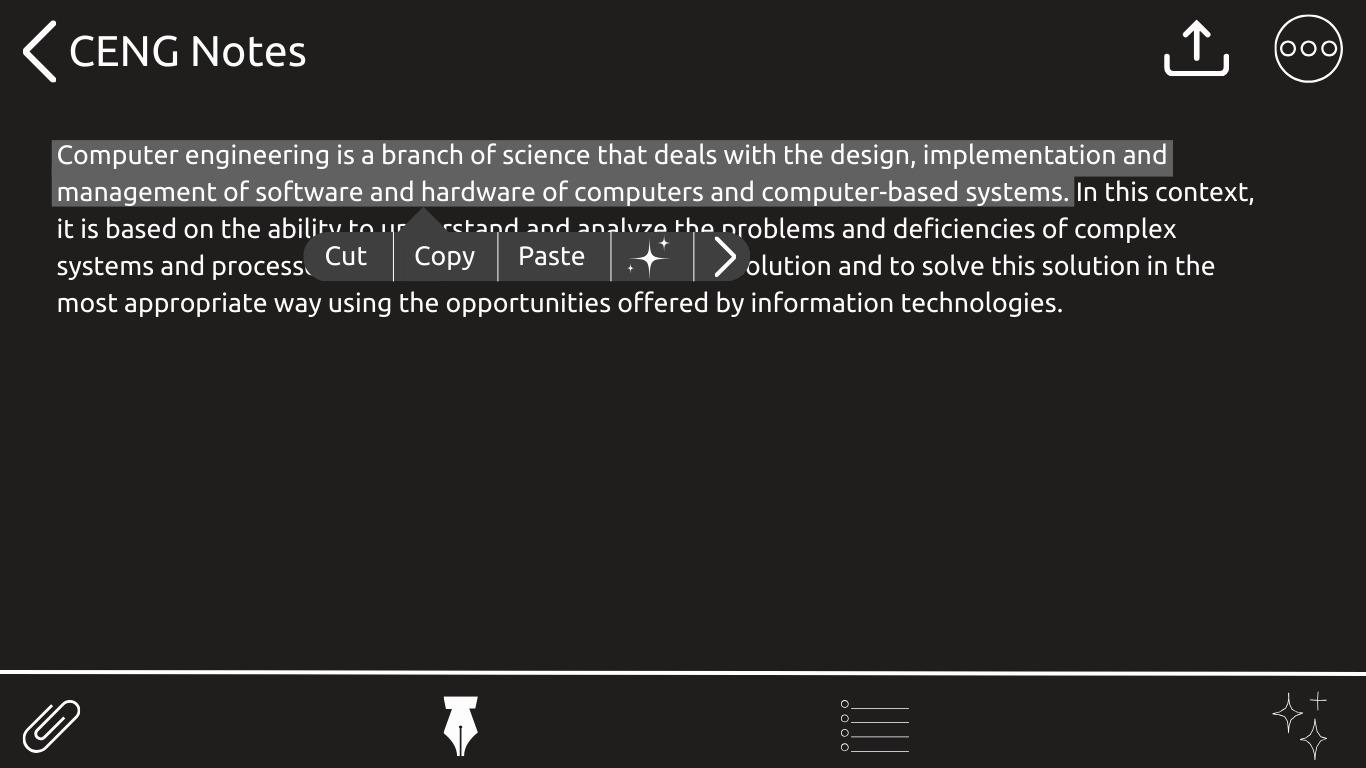
*** Figure 2- Home Page -Also showm note sharing***

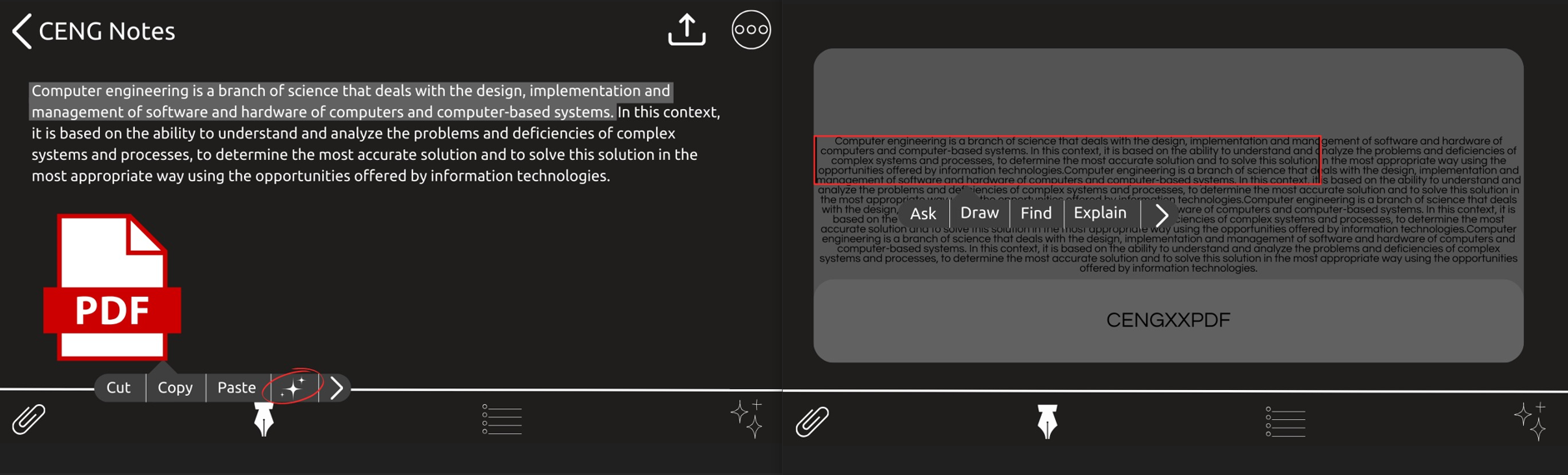
***Figure 3-Home screen- create note and other option***

** *Figure 4- Home Page -Note Visualization-Note creation***

**** *Figure 5 - Settings-theme selection***

#### 

 ***Figure 6-Asking questions directly from notes***

******

***Figure 7- Asking questions about the document in the note***

#### **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.

##### **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.

##### **3.1.2.2 Network Connectivity**

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

#### **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.

##### **3.1.4. Communications interfaces**

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

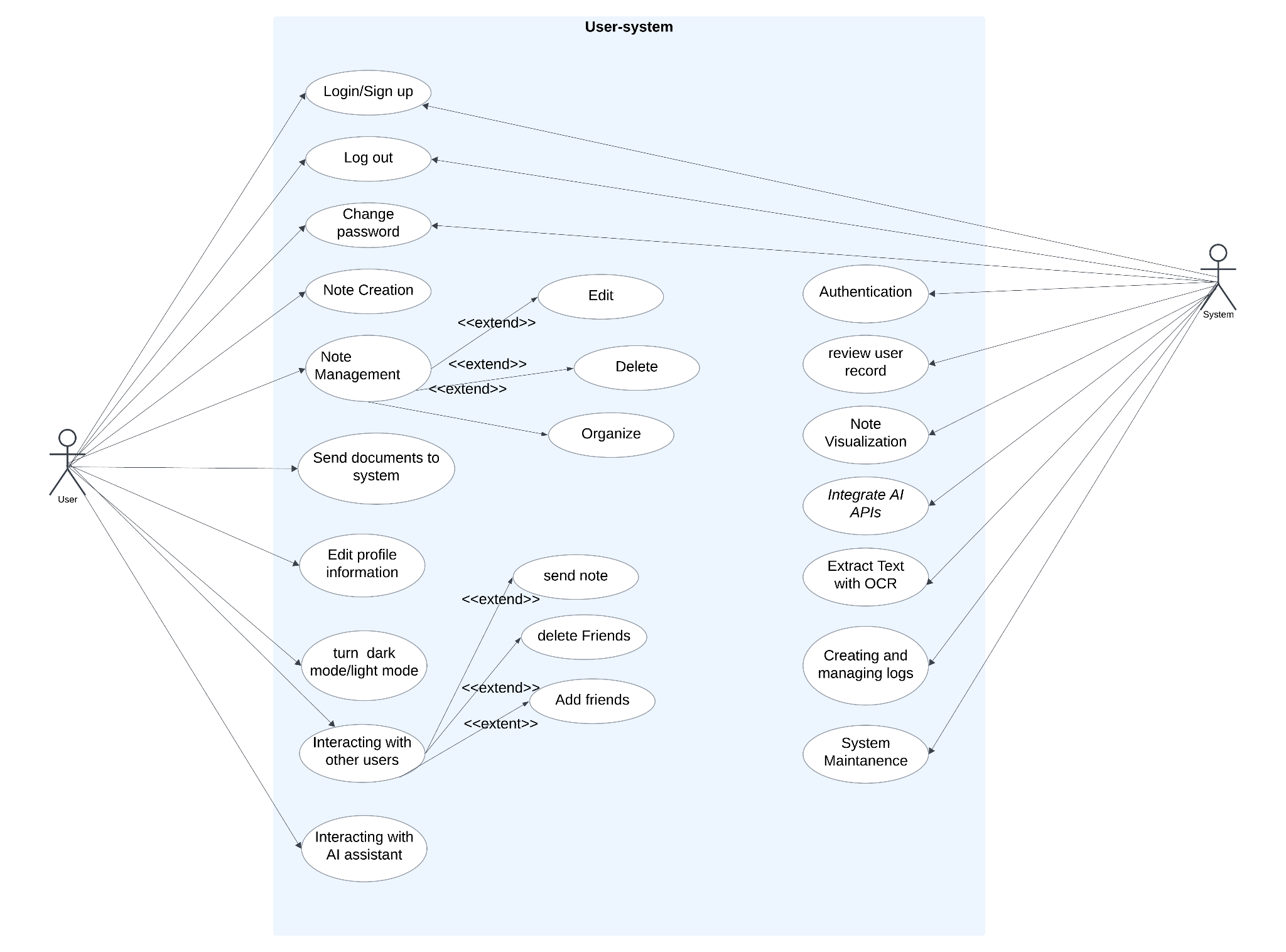
* **Real-Time Syncing**: Firebase will handle real-time updates, ensuring data such as notes, tasks, and projects are synced immediately across devices.
* **RESTful APIs**: For communication between the app and backend servers, facilitating efficient data exchange.
* **User Authentication**: **Google 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.

#### **3.2 Functional Requirements**

In this section, the functional requirements for the NoteWiz project are outlined based on the system's features and the primary services it offers. The following list is derived from the provided use case diagram, showing the users and their operations at a high level.

* The system shall allow users to sign up and log in securely using email and password credentials.
* The system shall provide a "Forgot Password" option to allow users to reset their password.
* Users shall have the ability to log out of the application.
* Users shall be able to edit their profile information, such as name, email, and profile picture.
* Users shall have the option to switch between dark mode and light mode to customize their viewing experience.
* The system shall allow users to create new notes.
* Users shall have the ability to:
  + Edit existing notes.
  + Delete notes they no longer need.
  + Organize notes into folders or categories.
* The system shall provide a search functionality to find notes by title or content.
* Users shall be able to send documents or files to the system and attach them to specific notes.
* The system shall allow users to interact with other users by:
  + Sending notes to friends.
  + Adding friends to their contact list.
  + Deleting friends from their contact list.
* The system shall include an AI assistant to assist users in managing their notes.
* Users shall be able to interact with the AI assistant to:
  + Retrieve summarized content from notes.
  + Get suggestions for organizing notes.
  + Ask general questions related to note-taking or system usage.
* The system shall allow administrators to:
  + Authenticate user actions for security purposes.
  + Review user records to resolve any issues.
  + Manage and create system logs to monitor activities.
* The system shall support regular system maintenance to ensure stability and security.



##### **3.2.1 NoteWiz User Authentication and Account Management**

##### **3.2.1.1 Log In and Sign up**

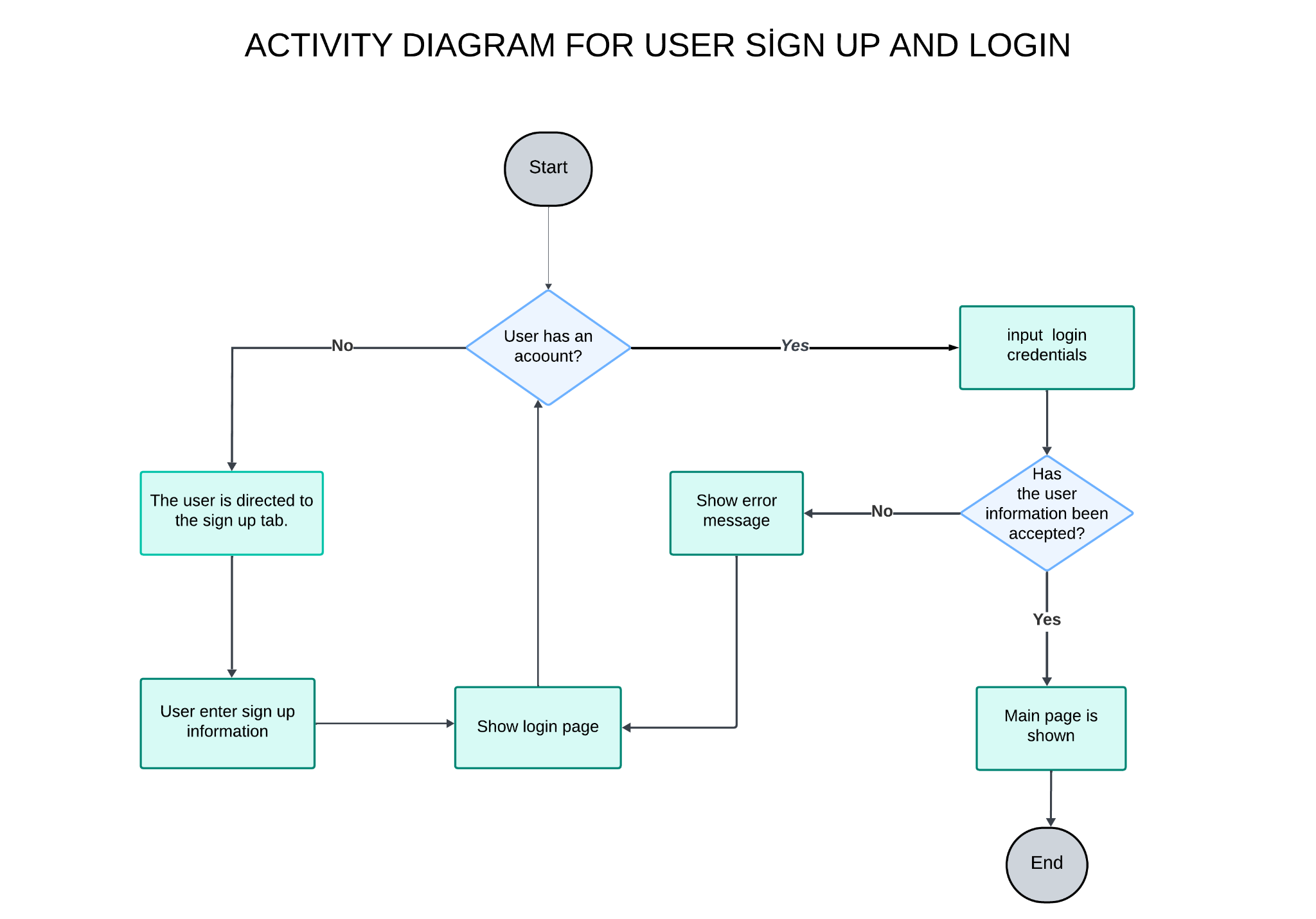
The login and registration feature allows authorized users to securely access NoteWiz. This feature provides a personalized and secure experience by verifying user credentials before granting access.

**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.
   * 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.

****

##### **3.2.1.2 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.

#### **3.2.2 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.

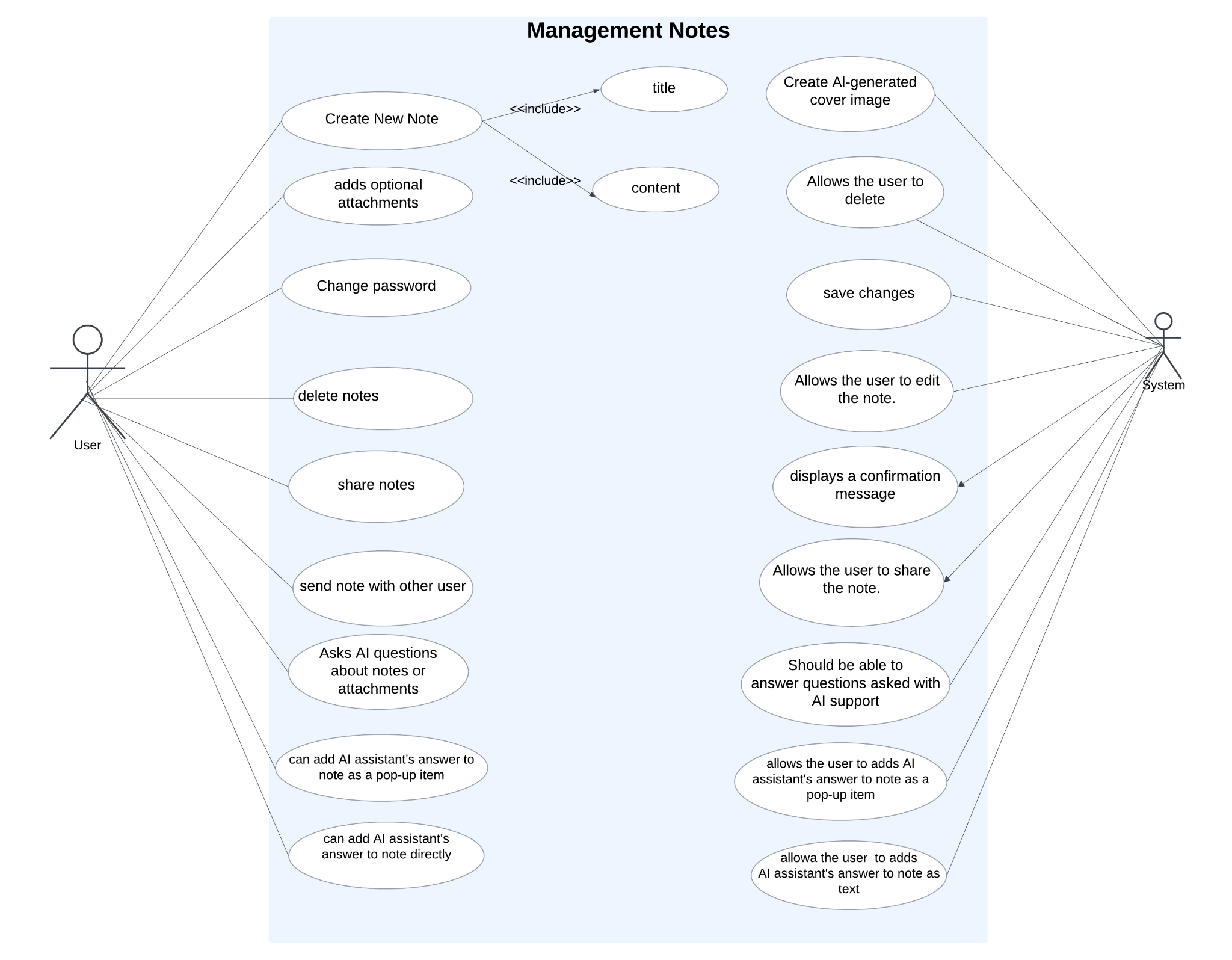
REQ-6: Notes should be securely stored in the user's account and accessible from multiple devices after logging in.

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.



#### **3.2.3 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.
   * If satisfied, the user approves the response.
   * If not satisfied, the user may provide feedback or refine the prompt.
10. **System:** Upon user approval:
    * Allows the user to add the response as a pop-up element in the labeled area.
    * 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.

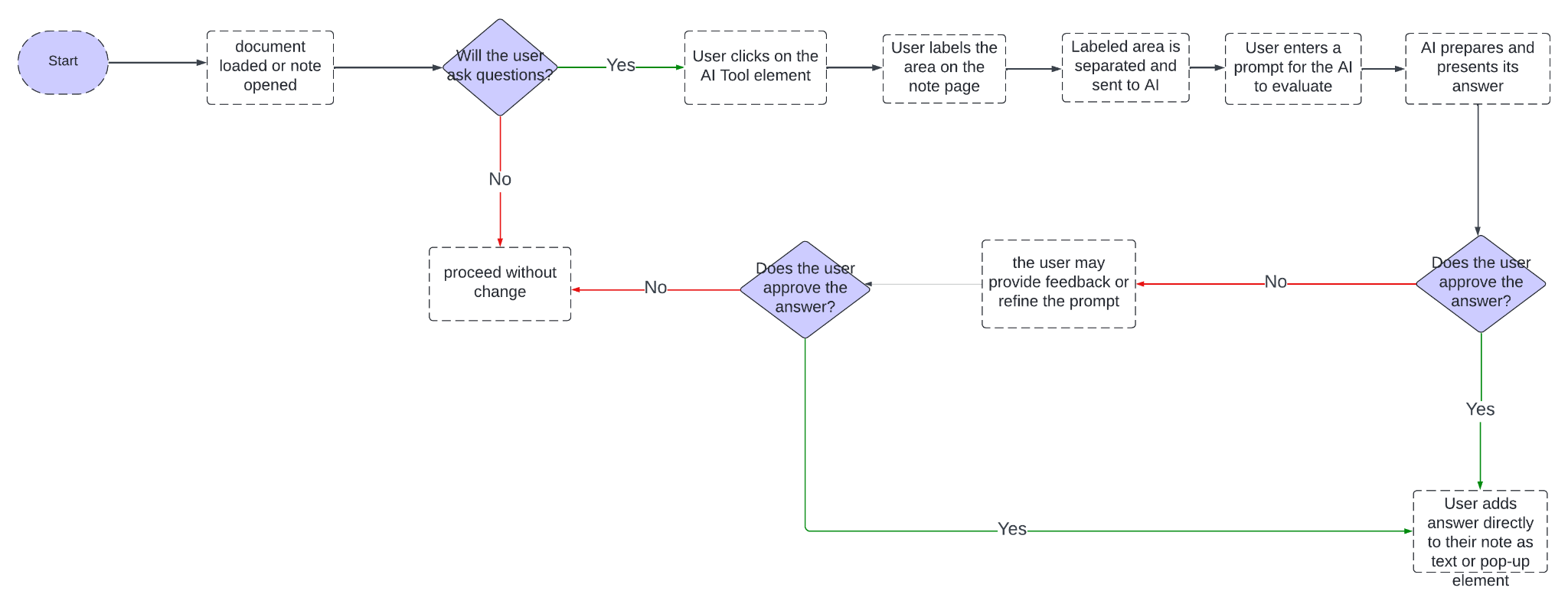
**REQ-4:** Users must be able to review and approve AI responses before adding them to the note.

**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.



#### **3.2.4 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.

##### **3.2.5 Cross-Platform Accessibility**

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

#### Process Sequence:

### User: Logs into their account on a new device.

### System: Authenticates the user’s credentials.

### System: Synchronizes all user data (notes, tasks, settings) from the cloud to the device.

### User: Makes changes (e.g., edits a note or completes a task).

### 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.

#### **3.2.6 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.

##### **3.2.7 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.

##### **3.2.8 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.

##### **3.2.9 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:**

1. **User:** Logs in using email or Google account.
2. **System:** Authenticates the credentials and grants access.
3. **System:** Encrypts and securely stores user data.

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

* **REQ-1:** The system must support authentication via email and Google accounts.
* **REQ-2:** All user data must be encrypted both in transit and at rest.
* **REQ-3:** The system should log out users after a period of inactivity for security purposes.

#### **3.3 Software System attributes**

##### **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.

##### **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.

##### **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.

##### **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.

##### **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.

##### **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.

##### **3.3.7 Maintainability**

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

#### **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.

# **4.REFERENCES**

| [1] | OCR: Optical Character Recognition," IEEE Dataport, [Online]. Available: <https://ieee-dataport.org/keywords/ocr-optical-character-recognition>. [Accessed: Dec. 3, 2024]. |
| --- | --- |

[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: 05-Dec-2024].

[3] 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: 22-Nov-2024].