# SEOScribe

## Content Writing Assistant

**Authors**

Muhammad Ansar Ullah

Umair Karim

Najibullah

**Supervisor**: Dr. Sajid Anwar

Final Year Project Report submitted in partial fulfillment of the requirements for the Degree of BSSE

im | sciences

INSTITUTE OF MANAGEMENT SCIENCES, PESHAWAR PAKISTAN
Session: 2020-2024

# Certificate of Approval

I, certify that I have read the report titled: **SeoScribe Content Writing Assistant**, by **Mohammad Ansar Ullah, Umair Karim and Najibullah**, and in my opinion, this work meets the criteria for approving the report submitted in partial fulfillment of the requirements for BSSE at Institute of Management Sciences, Peshawar.

Supervisor: Dr Sajid Anwar

Professor

Signature: _____________

Coordinator BSSE: Mr. Hamood Ur Rehman

Assistant Professor

Signature: ______________

Coordinator FYP: Mr. Omar Bin Samin

Lecturer

Signature: ______________

# Declaration

We, **Mohammad Ansar Ullah, Umair Karim and Najibullah**, hereby declare that the Final Year Project Report titled: **SeoScribe Content Writing Assistant** to FYP Coordinator and R&DD by us is our own original work. We are aware of the fact that in case, our work is found to be plagiarized or not genuine, FYP Coordinator and R&DD has the full authority to cancel our Final Year Project and We will be liable to penal action.

Mohammad Ansar Ullah

BSSE

Session: 2020-2024

Umair Karim

BSSE

Session: 2020-2024

Najibullah
BSSE
Session: 2020-2024

## Dedication

We dedicate this Final Year Project to our parents and teachers, who have always supported and helped us in every aspect of life.

## Acknowledgement

All the praise to Allah that induced the man with intelligence, knowledge, and wisdom. Peace and blessing of Allah be upon the Holy Prophet who exhort his followers to seek for knowledge from cradle to grave.

Foremost, we would like to express our sincere gratitude to our supervisor Dr. Sajid Anwar for his continuous support, patience, motivation, enthusiasm, and immense knowledge. His guidance helped us throughout the project. Last, but not the least, we would like to thank our parents for supporting us morally and spiritually throughout our life.

# Abstract

The problem in the digital content creation landscape is the high cost and complexity of existing SEO optimization tools. Current premium tools, such as AISEO and Surfer SEO, are primarily AI-driven and come with hefty subscription fees ranging from $50 to $1000 per month, making them inaccessible to small businesses, independent content creators, and students. This creates a significant barrier for those who need to optimize their content for better visibility and engagement but cannot afford such expensive tools. Hence, we came up with the solution, SEOscribe: Content Writing Assistant, designed to democratize access to essential SEO tools. SEOscribe is an affordable, user-friendly platform built using existing technologies, APIs, and Python libraries. It offers a powerful yet straightforward approach to content optimization by providing real-time content analysis and SEO best practices recommendations. Unlike other tools that heavily rely on AI, SEOscribe empowers authors to maintain control over their work while optimizing it for search engines. Key functionalities of SEOscribe include keyword generation, semantically related keyword and LSI (Latent Semantic Indexing) generation, real-time keyword monitoring, SEO and readability analysis, and keyword density monitoring. Additionally, it provides real-time tips and iterative improvements through a user-friendly graphical interface. By focusing on on-page content optimization and excluding off-page and technical SEO intricacies, SEOscribe ensures that content creators can enhance their work without being overwhelmed by complex SEO tasks.

# Table of Contents

# List of Figures

# List of Tables

# Chapter 1: Introduction

## 1.1 Overview

SEOscribe: Content Writing Assistant is made to make the process of creating and optimizing digital content easier. There's a growing need for a dependable, affordable, and user-friendly tool in the digital realm. By automating the processes involved in creating digital content, SEOscribe meets this need.

Utilizing existing technologies, APIs, and Python libraries, SEOscribe is a straightforward but powerful software. It seeks to offer an easy-to-use and free solution for SEO optimization and content creation. While SEOscribe provides content enhancement recommendations based on SEO best practices, authors are free to choose how to apply them. This contrasts with AISEO and Surfer Seo etc., some existing premium tools, which is primarily AI-driven and may lack a human touch.

During the production or revision process, real-time content analysis guarantees adherence to SEO best practices. Finding relevant terms is made easier with simplified keyword analysis, which increases content visibility.

SEOscribe is a comprehensive content writing assistant, promoting active content improvement alongside authors' workflows. It represents a shift toward accessible SEO excellence for all, marking the start of a transformative journey in content creation.

## 1.2 Project Motivation

The reason behind the creation of SEOscribe: Content Writing Assistant is our dissatisfaction with the exorbitant monthly fees of the SEO tools available today, which can range from $50 to $1000. As students of software engineering, we understand the importance of SEO in the current digital era, yet many people cannot afford the many expensive tools available.

This inspired us to found SEOscribe since we think everyone should have access to these crucial SEO tools, not just those with big expenditures. SEOscribe promises to offer features comparable to those of current solutions at a price up to 70% less. Our mission is to democratize SEO tools so small businesses, independent content creators, and students may afford them.

## 1.3 Project Scope and Glossary

### *1.3.1 Scope*

#### *1.3.1.1 In-scope*

The scope of SEOscribe is distinctly focused on content optimization within the domain of content writing. SEOscribe positions itself as a specialized content writing assistant, honing in on the following key functionalities:

1) **Keyword Generation:**

   Create versions of the input focus keywords to utilize in headlines and throughout the article.

2) **Semantically Related Keywords and LSI Generation:**

   Generate relevant keywords and LSI terms using the focal keyword.

3) **Text Editor with Real-time Keyword Monitoring:**

   Provide a simple text editor with basic word processing capabilities, as well as real-time keyword usage monitoring and highlighting.

4) **SEO and Readability Analysis:**
5) Evaluate content for SEO conformance and readability level, and make recommendations for improvement**.**
6) **Keyword Density Monitoring:**

   Monitor keyword usage throughout the article and utilize color coding to demonstrate conformity (green: recommended, red: overused).

7) **Additional Functionalities:**

   - **Real-time tips:** Display SEO and readability tips in a user-friendly sidebar for simple adoption.
   - **Iterative Improvements:** Analyze material as users make modifications and offer updated ideas.
   - **Graphical Interface:** Utilize a sidebar for intuitive presentation of suggestions.

#### *1.3.1.2 Out Scope*

Our project deliberately excludes engagement with off-page SEO and technical SEO intricacies,

- **Off-Page SEO-related features:** The tool does not include functionalities related to off-page SEO strategies, such as backlink building and social media promotion.
- **Technical SEO-related features:** Excludes features related to website structure, page speed optimization, and other technical aspects of SEO.
- **Keyword Research tools:** The tool does not provide functionalities for in-depth keyword research or competitor analysis

### *1.3.2 Glossary*

1) **SEO (Search Engine Optimization):**

   The practice of optimizing web content to enhance its visibility and ranking on search engine results pages.

2) **NLP (Natural Language Processing):**

   - A branch of artificial intelligence focused on the interaction between computers and human language, enabling machines to understand, interpret, and generate human-like text.

3) **API (Application Programming Interface):**

   A set of rules and protocols that allows different software applications to communicate with each other, facilitating the integration of external services and data sources.

4) **NLTK (Natural Language Toolkit):**

   A Python library used for natural language processing tasks, including tokenization, stemming, and semantic analysis.

5) **Llama 3 API:**

   An API (Application Programming Interface) provided by Meta for accessing the Llama 3 language model, enabling developers to integrate its capabilities into their applications.

6) **Flesch-Kincaid Readability Tests:**

   A set of readability tests designed to evaluate the ease of readability in written content, considering factors such as sentence length and syllable count.

7) **Gunning Fog Index:**

A readability metric that measures the complexity of written content based on sentence length and the use of complex words.

8) **Graphical Interface:**

The visual representation and interaction layer of a software application, often created using technologies such as JavaScript, HTML, and CSS, providing users with an intuitive way to interact with the application

## 1.4 Objectives

Our primary objectives center around the development of SEOscribe, a cutting-edge content optimization tool. We adopt a comprehensive approach to achieve the following goals:

**Development:**

- **SEOscribe Implementation:** Deliver a user-friendly, scalable web application, focusing on advanced content optimization functionalities to bridge the gap between premium tools and the needs of students, individual creators, and small businesses.
- **Scalability:** Ensure the platform is robust and scalable, aligning with long-term growth and usability criteria.

**User Empowerment:**

- **Content Optimization:** Equip users with real-time keyword monitoring, readability analysis, and color-coded suggestions, meeting the criteria of enhancing user-driven content optimization.
- **Intuitive Design:** Provide an intuitive interface, encouraging active participation and fostering a sense of ownership, aligning with user engagement and empowerment criteria.

**Efficiency in Content Management:**

- **Streamlined Processes:** Design SEOscribe to streamline content creation and optimization, ensuring efficiency and ease of use, meeting the criteria for productivity and user satisfaction.

- **Real-time Analysis:** Incorporate real-time text analysis and intuitive editing features, enhancing overall content management, aligning with efficiency and workflow optimization criteria.

**Performance and Accessibility:**

- **Fast and Responsive:** Ensure swift interactions and real-time updates, leveraging modern web technologies, meeting the performance and speed criteria.
- **User Accessibility**: Focus on accessibility for users with varying technical skills, providing a smooth experience, aligning with inclusivity and user accessibility criteria.

## 1.5 Tools

1) **Python Programming Language:**

Python serves as the backbone of SEOscribe due to its versatility and extensive libraries relevant to natural language processing (NLP), text analysis, and content optimization. The following Python libraries play a pivotal role:

**NLTK (Natural Language Toolkit):**

Utilized for NLP tasks, such as tokenization, Readability analysis and other natural language processing features.

Contributes to the generation of NLP keywords and semantic analysis of content.

2) **Web Framework:**

Flask or Django can be used as web frameworks for handling HTTP requests, routing, and managing the application structure.

3) **Llama 3 API:**

The Llama 3 API serves as a powerful tool for enhancing content analysis by tapping into Llama 3 language model. Key functionalities include:

Keywords and LSI generation:  API is utilized to generate variation of the focus keyword provided by the user

4) **Readability Metrics Tools:**

**Flesch-Kincaid Readability Tests:**

Measures the ease of readability in content by assessing sentence length and syllable count.

Contributes to content optimization by providing readability suggestions.

**Gunning Fog Index:**

Gauges the readability of content by considering sentence length and the use of complex words.

Offers insights to improve content comprehension.

5) **Graphical Interface:**

**JavaScript and HTML/CSS:**

Employed to create an interactive and user-friendly graphical interface.

Integrates seamlessly with the Python backend to present suggestions through a graphical

sidebar, enhancing user experience.

6) **Version control**

Git and GitHub can be used for tracking changes and progress in code

By leveraging these tools, SEOscribe aims to create a robust content optimization assistant that not only analyzes text but also provides meaningful and actionable insights to content creators. The integration of various tools ensures a comprehensive approach to achieve the defined objectives

# Chapter 2 Background Study

## 2.1 Related Work

The landscape of SEO content optimization tools is diverse, with numerous solutions tailored to different user needs. These tools range from AI-powered writing assistants to comprehensive SEO platforms that cater to bloggers, small business owners, and professional SEO experts alike. Among these, SEMrush Writing Assistant, Yoast SEO, Surfer SEO, and AISEO stand out as prominent tools that have gained considerable traction in the industry due to their distinct features and functionalities.

The market for SEO content optimization tools is saturated with a plethora of options, each offering a unique set of features and functionalities. These tools cater to a broad spectrum of users, ranging from bloggers and small business owners to experienced SEO professionals. Several notable tools stand out in the competitive landscape, offering comprehensive solutions for content optimization. These include:

### *2.1.1 SEMrush Writing Assistant*

The SEMrush Writing Assistant is a prominent tool in the field of SEO content optimization, designed to assist content creators in crafting SEO-friendly and engaging copy. This AI-powered tool seamlessly integrates with widely used platforms such as Google Docs, MS Word, and WordPress, offering real-time feedback on crucial aspects like readability, tone of voice, originality, and SEO compliance. SEMrush employs a combination of natural language processing (NLP) techniques and machine learning algorithms to provide these insights, ensuring that content adheres to SEO best practices while resonating with the target audience.

The SEMrush Writing Assistant works by analyzing the content in real-time, comparing it against a vast database of web content and SEO guidelines. The tool provides a comprehensive analysis of the content's SEO performance, including keyword usage, meta tags, and internal linking structure. Additionally, it offers suggestions for improving the readability of the content, such as adjusting sentence length, reducing passive voice, and enhancing the overall tone. The tool's ability to offer AI-powered features like paraphrasing, summarizing, and content expansion further streamlines the writing process, allowing users to generate high-quality content efficiently.

From a technical standpoint, SEMrush integrates various APIs and Python libraries to function seamlessly across different platforms. The tool's backend is built using a combination of Python, JavaScript, and other web technologies, ensuring compatibility with popular content management systems (CMS) like WordPress. The use of cloud-based infrastructure allows SEMrush to process large volumes of data in real-time, providing users with instant feedback on their content. Furthermore, SEMrush's integration with Google Analytics and other SEO tools enables users to track the performance of their content over time, making it a comprehensive solution for content optimization [1], [2].

### *2.1.2 Yoast SEO*

Yoast SEO is one of the most widely used SEO plugins, particularly within the WordPress community. It offers a user-friendly interface that guides writers toward on-page SEO best practices. Yoast SEO meticulously analyzes content for crucial factors such as title tags, meta descriptions, internal linking, and keyword usage. By following Yoast's tailored recommendations, content creators can ensure that their WordPress posts adhere to established SEO principles, potentially increasing their visibility in search engine results.

Yoast SEO operates by evaluating the content's SEO performance in real-time, providing users with a color-coded assessment of various SEO factors. The tool uses NLP and keyword analysis algorithms to assess the content's relevance to the target keywords and suggests improvements where necessary. It also evaluates the content's readability, offering suggestions to enhance sentence structure, paragraph length, and overall clarity. The tool's emphasis on keyword optimization and meta tag management makes it an essential tool for content creators looking to improve their on-page SEO [3].

From a technical perspective, Yoast SEO is built using PHP, JavaScript, and MySQL, making it highly compatible with WordPress. The plugin integrates with WordPress's existing infrastructure, allowing users to manage their content's SEO directly from the WordPress dashboard. Yoast SEO also leverages the WordPress REST API to provide real-time feedback and updates, ensuring that users can optimize their content as they write. Additionally, Yoast SEO offers integration with Google Search Console, enabling users to monitor their website's SEO performance and identify areas for improvement [4], [5].

### *2.1.3 Surfer SEO*

Surfer SEO stands out in the SEO landscape for its data-driven approach to content optimization. Unlike other tools that focus primarily on keyword analysis, Surfer SEO delves deeper into the factors that influence a webpage's ranking in search engine results. By analyzing the top-ranking pages for a target keyword, Surfer SEO generates a comprehensive content quality score, which assesses various factors such as word count, headings, images, and internal linking. This score provides valuable insights into what Google might deem high-quality content, allowing writers to tailor their content to closely resemble successful competitors.

Surfer SEO's core functionality revolves around its SERP analyzer, which evaluates the top-ranking pages for a given keyword and extracts data on various on-page SEO factors. The tool uses machine learning algorithms and NLP techniques to analyze the content structure, keyword density, and other SEO elements of these pages. Surfer SEO then provides users with actionable insights on how to optimize their content to match or exceed the quality of the top-ranking pages. The tool also offers a content editor that integrates these insights directly into the writing process, allowing users to optimize their content in real-time [6].

Technically, Surfer SEO is built on a robust backend infrastructure that leverages cloud computing and big data technologies. The tool's ability to analyze large volumes of data from search engine results pages (SERPs) in real-time requires a highly scalable and efficient architecture. Surfer SEO utilizes Python, JavaScript, and various machine learning frameworks to process and analyze data, providing users with real-time feedback on their content's SEO performance. The tool's integration with popular CMS platforms like WordPress further enhances its usability, allowing content creators to optimize their content directly within their preferred writing environment [7], [8].

### *2.1.4 AISEO*

AISEO is a comprehensive suite of AI-powered SEO tools designed to empower content creators with advanced features for content generation and optimization. AISEO leverages cutting-edge NLP and machine learning technologies to generate content in various formats, conduct thorough keyword research, and provide on-page optimization suggestions. The tool's focus on AI-driven solutions makes it particularly appealing to content creators seeking to streamline their workflow and enhance their SEO performance.

AISEO operates by integrating multiple AI models that are trained on vast datasets of web content and SEO guidelines. These models are capable of generating high-quality content in different formats, from blog posts to product descriptions, with minimal input from the user. The tool also offers advanced keyword research capabilities, allowing users to identify high-value keywords and optimize their content accordingly. Additionally, AISEO provides real-time on-page optimization suggestions, helping users to refine their content's SEO performance as they write [9].

The technical implementation of AISEO involves the use of advanced machine learning frameworks such as TensorFlow and PyTorch, which enable the tool's AI models to learn and improve over time.

The tool's backend is built using Python and JavaScript, ensuring compatibility with various web platforms. AISEO also leverages cloud-based infrastructure to provide scalable and efficient content generation and optimization services. The tool's integration with popular CMS platforms and SEO tools further enhances its functionality, making it a valuable asset for content creators looking to improve their SEO performance [10].

## 2.2 Limitations of Previous Works

Despite the wide range of features offered by these tools, they are not without limitations. One of the most significant drawbacks is their cost, with many of these tools requiring expensive subscriptions that can range from $50 to $1000 per month. This makes them inaccessible to smaller content creators, students, and small businesses that cannot afford such expenses. Furthermore, these tools often have a broader scope that includes features beyond content optimization, such as technical SEO and off-page SEO strategies, which may not be relevant to all users.

In contrast, our proposed system, SEOscribe, is specifically tailored for content optimization and is offered as a free tool. By focusing solely on content creation and optimization, SEOscribe provides a more accessible and user-friendly solution for those who need to improve their SEO performance without the financial burden of expensive subscriptions. This targeted approach ensures that users can benefit from advanced content optimization features without being overwhelmed by unnecessary functionalities.

## 2.3 Comparison Table

| **Tool Name** | **Description** | **Features** | **Limitations** |
|---|---|---|---|
| SEMrush Writing Assistant | A cloud-based tool that provides real-time SEO recommendations and suggestions as users compose their content. | Real-time SEO recommendation<br>Keyword suggestions<br>Tone of voice suggestions<br>Target audience insights | Can be expensive<br>Recommendations may be overly conservative |
| Yoast SEO | A popular WordPress plugin that offers a range of SEO optimization features, including meta tag generation, image optimization, and keyword analysis. | Meta tag generation<br>Image optimization<br>Keyword analysis<br>Sitemap generation<br>Content optimization suggestions | Difficult to configure for complex websites<br>Interface can be cluttered and overwhelming |
| Surfer SEO | A content optimization tool that offers features such as on-page analysis, competitor comparison, and real-time editing by utilizing data-driven insights, aiming to improve organic search rankings and user experience. | On-page analysis for keyword usage and content structure.<br>Competitor comparison for strategic insights.<br>Real-time editing with immediate impact assessment. | relatively expensive for individual users or small businesses, limiting accessibility to certain budget constraints.<br>Users might experience a learning curve, particularly if they are new to SEO |
| AISEO | An AI-powered content optimization tool. It provides real-time SEO recommendations during content creation, emphasizing personalized content suggestions based on AI insights. AISEO focuses on enhancing user engagement through user-friendly content and context-aware suggestions. | AI-powered content suggestions for optimization.<br>Real-time SEO recommendations during content creation.<br>Emphasis on user-friendly content and engagement. | AISEO's effectiveness is reliant on the accuracy and relevance of its AI suggestions<br>Users might face limitations in customizing AI-generated suggestions to align with specific content creation preferences<br>Relying solely on AI suggestions may result in a lack of human touch and creativity |

Table 2.1 Comparison Table

# Chapter 3 System Requirements, Architecture, and Design

## 3.1 Software Requirements Specification (SRS)

### *3.1.1 Introduction*

#### *3.1.1.1 Purpose*

This Software Requirement Specification (SRS) document defines the functional and non-functional requirements for the development of SEOscribe: Content Writing Assistant, a web application designed to empower content creators with real-time feedback and valuable suggestions for optimizing their written content.

#### *3.1.1.2 Scope*

This SRS covers the following aspects of SEOscribe:

**Functional requirements:** Describing the specific features and functionalities of the tool, including keyword generation, keyword usage monitor, LSI suggestions, real-time text analysis, SEO and readability analysis, basic text processing features like cut, copy, paste, text formatting and image and links insertion.

**Non-functional requirements:** Outlining performance, security, usability, and other quality attributes of the application.

**User interface (UI) requirements:** Specifying the design and interaction elements of the graphical interface.

**External interface requirements:** Detailing how the tool will interact with external services and APIs.

#### *3.1.1.3 Definitions, Acronyms, and Abbreviations*

- SEO: Search Engine Optimization
- NLP: Natural Language Processing
- LSI: Latent Semantic Indexing
- API: Application Programming Interface
- UI: User Interface

- UX: User Experience

### *3.1.2 Overview*

This SRS document provides a comprehensive roadmap for the development of SEOscribe. It outlines the features, functionalities, and technical specifications of the application, ensuring a clear understanding of its intended functionalities and desired user experience.

### *3.1.3 General Description*

#### *3.1.3.1 Product Perspective*

SEOscribe positions itself as a user-friendly and free web application for content creators seeking to optimize their written content for search engines. It aims to bridge the gap between existing premium tools and the needs of students, individual creators, and small businesses by offering effective content optimization functionalities at an accessible price point.

#### *3.1.3.2 Product Functions*

##### 3.1.3.2.1 Core Functionalities

- **Keyword Generation:** Generate variations of the input focus keyword for use in headings and throughout the content.
- **Semantically Related Keywords and LSI Generation:** Provide relevant keywords and LSI terms based on the focus keyword.
- **Text Editor with Real-time Keyword Monitoring:** Offer a user-friendly text editor with basic word processing functionalities and with real-time keyword usage monitoring and highlighting.
- **SEO and Readability Analysis**: Analyze the content for SEO adherence and readability level, providing suggestions for improvement.
- **Keyword Usage Monitoring and Color Coding:** Monitor keyword usage throughout the content and indicate adherence through color coding (green: recommended, red: overused).

#### 3.1.3.2.2 Additional Functionalities

- **Real-time Suggestions:** Display SEO and readability suggestions in a user-friendly sidebar for easy implementation.
- **Iterative Improvements:** Continuously analyze content as users make changes and provide updated suggestions.
- **Graphical Interface:** Utilize a sidebar similar to Surfer SEO for intuitive presentation of suggestions.

### *3.1.3.3 User Characteristics*

The primary target users of SEOscribe include:

- **Content Creators:** Individuals who create content for websites, blogs, and social media platforms.
- **Students:** Individuals learning about SEO and content creation in academic settings.
- **Small Businesses:** Businesses with limited budgets seeking to improve their online presence through content optimization.

### *3.1.3.4 General Constraints*

- **Budgetary limitations:** Development costs of the tool should be kept within a specific budget.
- **Accessibility:** The tool must be affordable and accessible to a wide range of users.
- **Ease of use:** The UI should be user-friendly and intuitive, requiring minimal technical expertise.

### *3.1.3.5 Assumptions and Dependencies*

- Users have basic access to web browsers and an internet connection.
- The application will be optimized for desktop browsers initially.
- Mobile app development is ruled out as it hinders efficiency and ease of use.

### *3.1.4 Specific Requirements*

#### *3.1.4.1 Functional Requirements*

##### 3.1.4.1.1 Log In

- FR1: The app must display a login button on the main screen once the user opens the app.
- FR2: Clicking the login button must display a login form.
- FR3: The login form must include fields for email address and password.
- FR4: The system must validate the entered email against registered email addresses.
- FR5: The system must validate the entered password against the stored password.
- FR6: Upon successful login, the user must be redirected to the SeoScribe dashboard.
- FR7: Upon unsuccessful login due to invalid email, the system must display an error message prompting re-entry.
- FR8: Upon unsuccessful login due to invalid password, the system must display an error message prompting re-entry of password.

##### 3.1.4.1.2 Sign Up

- FR9: The app must display a sign-up button on the main screen once the user opens the app.
- FR10: Clicking the sign-up button must display a sign-up form.
- FR11: The sign-up form must include fields for email address, username, and password.
- FR12: The system must validate email uniqueness against the database.
- FR13: The system must validate password strength with a minimum length of 8 characters.
- FR14: The sign-up process must ensure accepting the terms and conditions before creating an account through an agreement checkbox.
- FR15: Upon successful sign-up, the system must create a new user account in the database.

- FR16: Upon successful sign-up, the system must redirect the user to the login page.
- FR17: Unsuccessful sign-up due to an existing email must prompt the user to input a different email address.
- FR18: Unsuccessful sign-up due to a weak password must prompt for a password longer than 8 characters.
- FR19: Unsuccessful sign-up due to missing agreement must prompt the user to agree to the terms and conditions.

##### 3.1.4.1.3 Create a New Project

- FR20: The system must allow the user to create a new project only if they are logged in.
- FR21: The system must display a "Create Project" button on the writing assistant dashboard.
- FR22: Clicking the "Create Project" button must prompt the user to name the project.
- FR23: The system must provide a description box for the user to enter a project description (optional).
- FR24: The system must allow the user to input the project name and description.
- FR25: The system must create a project entry in the database upon clicking the "Next" button.
- FR26: The system must redirect the user to the keyword generation page after creating the project.
- FR27: If the user cancels the project creation, the system must redirect them to the dashboard without creating a project

##### 3.1.4.1.4 Open an Existing Project

- FR28: The system must allow the user to open an existing project only if they are logged in.
- FR29: The system must display previously created projects on the writing assistant dashboard.
- FR30: Clicking on a project must load it in the text editor.

- FR31: If the system fails to load the project, it must display an error message.

#### 3.1.4.1.5 Keyword Generation

- FR32: The system must allow the user to generate keywords only if they are logged in.
- FR33: The system must display a "Keyword Tool" button on the homepage.
- FR34: Clicking the "Keyword Tool" button must redirect the user to the keyword generation page.
- FR35: The system must prompt the user to enter a focus keyword.
- FR36: The system must generate at least long-tail keywords and LSI keywords based on the entered focus keyword.
- FR37: If the entered keyword is longer than 60 characters, the system must display a warning prompting the user to reduce the character length to 60.
- FR38: If the API fails to respond, the system must display an error message asking the user to check their internet connection.
- FR39: The system must display the generated long-tail and LSI keywords.

#### 3.1.4.1.6 Text Editor

- FR40: The system must allow the user to open a created project in the text editor only if they have completed the steps in UC03 and UC05.
- FR41: The system must redirect the user to the keyword generation page after clicking the "Next" button on the project creation window.
- FR42: After generating keywords, the system must display a "Next" button to proceed to the text editor.
- FR43: The system must update the project entry in the database with the generated keywords.
- FR44: The system must display the keywords in the sidebar of the text editor.
- FR45: If the user cancels the operation, the system must revert to the writing assistant dashboard and discard any changes.

#### 3.1.4.1.7 Input Text

- FR46: The system must allow the user to input text into the text editor only if they have completed the steps in UC03 and UC05.
- FR47: The system must accept text input from the user.
- FR48: The system must identify the text as either body or heading and format it accordingly.
- FR49: If the system fails to accept input, it must display an exception error message.
- FR50: The system must display the input text and format it accordingly.

#### 3.1.4.1.8 Edit Text

- FR51: The system must allow the user to perform text editing operations only if they have text inside the text editor.
- FR52: The system must allow the user to select text.
- FR53: The system must provide options for various editing operations such as font selection, bold, italic, underline, ordered list, numbered list, alignment, and spacing.
- FR54: The system must apply the selected editing operation to the selected text.
- FR55: The system must display the results of the applied editing operation.
- FR56: If the user deselects the text, the system must revert to the editor without applying the operation.

#### 3.1.4.1.9 SEO Analysis

- FR57: The system must allow the user to get SEO analysis of the text only if they have completed the steps in UC03, UC04, and UC05.
- FR58: The system must scan the headings for long-tail keywords.
- FR59: The system must highlight the matched keywords in the sidebar as green.
- FR60: The system must scan the paragraphs for LSI keywords.
- FR61: The system must highlight the matched keywords in the sidebar as green.
- FR62: The system must perform steps FR58 to FR61 continuously as the user makes changes to the text.

- FR63: If the system finds no matches for a keyword, the keyword color in the sidebar must remain unchanged.

#### 3.1.4.1.10 Readability Analysis

- FR64: The system must allow the user to get readability analysis only if they have completed the steps in UC03, UC04, and UC05.
- FR65: The system must identify long sentences and difficult vocabulary.
- FR66: The system must highlight the identified issues in the sidebar under the readability section.
- FR67: The system must scan the text continuously and highlight resolved issues as green and unresolved issues as red.
- FR68: If a corresponding issue is not fit on the Gunning Fog Index readability metrics, the issue color must remain unchanged in the sidebar.

#### 3.1.4.1.11 Keyword Density Monitor

- FR69: The system must allow the user to monitor keyword density only if they have text in the text editor and the text is formatted with incorporated keywords.
- FR70: The system must display a "Check Keyword Density" button in the sidebar.
- FR71: Clicking the "Check Keyword Density" button must display a drop-down menu with all the keywords and LSI keywords.
- FR72: The system must allow the user to select a keyword from the drop-down menu.
- FR73: The system must scan the text and detect occurrences of the selected keyword.
- FR74: The system must calculate keyword density based on occurrences/total word count * 100.
- FR75: If the corresponding keyword is not found in the text, the keyword color must remain unchanged.
- FR76: The system must highlight keywords with density greater than 2% as red in the text editor.

#### 3.1.4.1.12 Paraphrasing

- FR77: The system must allow the user to paraphrase text only if they have an internet connection.
- FR78: The system must display a "Paraphraser" button on the homepage.
- FR79: Clicking the "Paraphraser" button must redirect the user to the paraphraser page.
- FR80: The system must allow the user to input text to be paraphrased.
- FR81: The system must provide a "Paraphrase" button to initiate the paraphrasing process.
- FR82: The system must display the paraphrased text as results.
- FR83: If the user cancels the operation, the system must revert to the text editor.

#### 3.1.4.1.13 Image Insertion

- FR84: The system must allow the user to insert an image at the cursor position only if they are on the text editor page.
- FR85: The system must provide an "Insert Image" icon on the toolbar.
- FR86: Clicking the "Insert Image" icon must open a prompt providing an option to browse for an image.
- FR87: The system must allow the user to select an image and click "Insert".
- FR88: The system must prompt the user to a window asking to modify the image size.
- FR89: The system must provide an option to insert alt text for the image on the same window.
- FR90: The system must allow the user to input the image size and alt text.
- FR91: If the system fails to insert the image, it must display an error message "Image insertion failed, try again".
- FR92: The system must insert the image at the cursor position.

#### 3.1.4.1.14 Hyperlink Insertion

- FR93: The system must allow the user to insert a hyperlink only if they are on the text editor page and have text inside the editor.
- FR94: The system must allow the user to select text.
- FR95: The system must provide a "Hyperlink" icon on the toolbar.
- FR96: Clicking the "Hyperlink" icon must open a window prompting the user to provide a link.
- FR97: The system must allow the user to input the link and click "Done".
- FR98: The system must associate the link with the selected text.
- FR99: If the user cancels the operation, the system must revert back to the editor.
- FR100: The system must display the selected text as anchor text for the link.

### *3.1.4.2 Non-Functional Requirements*

#### 3.1.4.2.1 Performance

- NFR1: The system shall respond to user interactions (such as clicks, typing, button presses) within 1 second on average. This includes loading times for pages, data updates, and feedback messages.
- NFR2: The API shall respond to keyword generation requests within 5 seconds with at least 90% availability.

#### 3.1.4.2.2 Usability

- NFR3: The user interface shall be intuitive and easy to navigate for users of all technical skill levels.
- NFR4: All functionalities shall be clearly labeled and explained within the interface or readily accessible through contextual help features.
- NFR5: The learning curve for mastering the basic features of the app should be minimal and achievable within 20 minutes of use.
- NFR6: The system shall provide clear error messages and instructions for resolving any encountered issues.

- NFR7: The interface should be intuitive and easy to use, with clear visual cues i.e., green for keyword that is present and red for no presence and for keyword highlights and green for resolved, red for present readability issues.

##### 3.1.4.2.3 Security

- NFR8: The system shall implement industry-standard security measures to protect user data and ensure data privacy
- NFR9: The system should inform the user through email within 5 minutes of unauthorized access and data breaches.

##### 3.1.4.2.4 Integration

- NFR10: The system must successfully integrate with the Llama3 API for keyword generation.

##### 3.1.4.2.5 Ease of Use

- NFR11: The UI should be intuitive and easy to use, with clear instructions and error messages.

##### 3.1.4.2.6 Reliability

- NFR12: The application should be available and functional and must handle at least 100 users at a time.
- NFR13: Downtime frequency should be less than 5%.

##### 3.1.4.2.7 Maintainability

- NFR14: The code should be modular and well-documented to facilitate future updates and bug fixes.

##### 3.1.4.2.8 Portability

- NFR15: The SeoScribe app shall be compatible with all major web browsers including Chrome, Edge, Firefox.

### *3.1.5 User Interface (UI) Requirements*

#### *3.1.5.1 Screen Layouts*

##### 3.1.5.1.1 Login/Signup page:

- Display the platform logo and brand name prominently.

- Login form: Provide dedicated fields for email and password with a large "Log In" button.
- Signup form: Offer fields for email, username (optional), and password with a clear "Sign Up" button.

##### 3.1.5.1.2 Home Page:

- Prominent display of primary features “Create new project” and list of already created projects
- Clear call to action for users to start a new project.
- Access to account settings and user profile.

##### 3.1.5.1.3 Keyword Generation Page:

- Input field for focus keyword.
- Display of next button to navigate to the text editor

##### 3.1.5.1.4 Text Editor Page:

- User-friendly text editor with features like font size, formatting options, and spell-checking.
- Real-time keyword usage highlighting and color coding.
- Sidebar at the right-side displaying sections, keywords, readability, SEO Score, headings schema, keyword density check and word count.

#### *3.1.5.2 Input Fields*

- Clearly labeled input fields with appropriate placeholders and instructions.
- Input validation to ensure data accuracy and prevent errors.

#### *3.1.5.3 Output Displays*

- Visually appealing and informative presentation of analysis results and suggestions.

#### *3.1.5.4 Error Messages*

- Informative and user-friendly error messages to guide users in correcting errors.
- Offer suggestions for troubleshooting or provide links to help resources.

#### *3.1.5.5 Help and Support Features*

- Comprehensive help documentation or a knowledge base for users to access.
- Option to contact support for further assistance.

### *3.1.6 External Interface Requirements*

#### *3.1.6.1 Llama 3 API*

- Function: Accessing the Llama 3 language model for keywords generation.
- Input: Focus keyword provided by the user.
- Output: Variation of the keyword and LSI keywords
- Output: Readability scores and suggestions for improvement.

#### *3.1.6.2 Web Browser Compatibility*

- The application should be compatible with major web browsers (Chrome, Firefox, Edge).
- Ensure consistent functionality and user experience across different browsers and devices.

## 3.2 Data Flow diagrams

### *3.2.1 DFD Level 0*

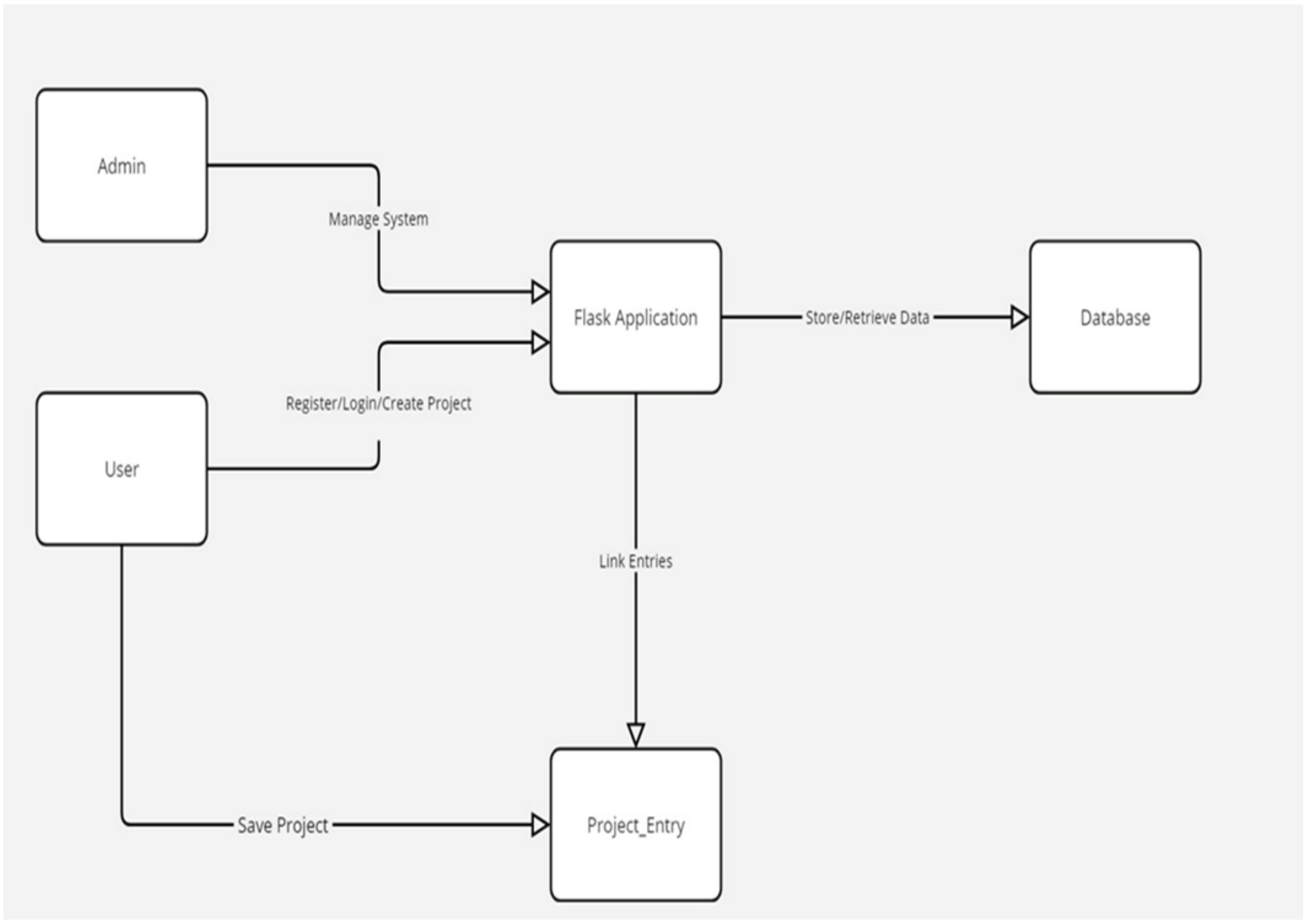


Figure 3.1 DFD Level 0

### 3.2.2 DFD Level 1

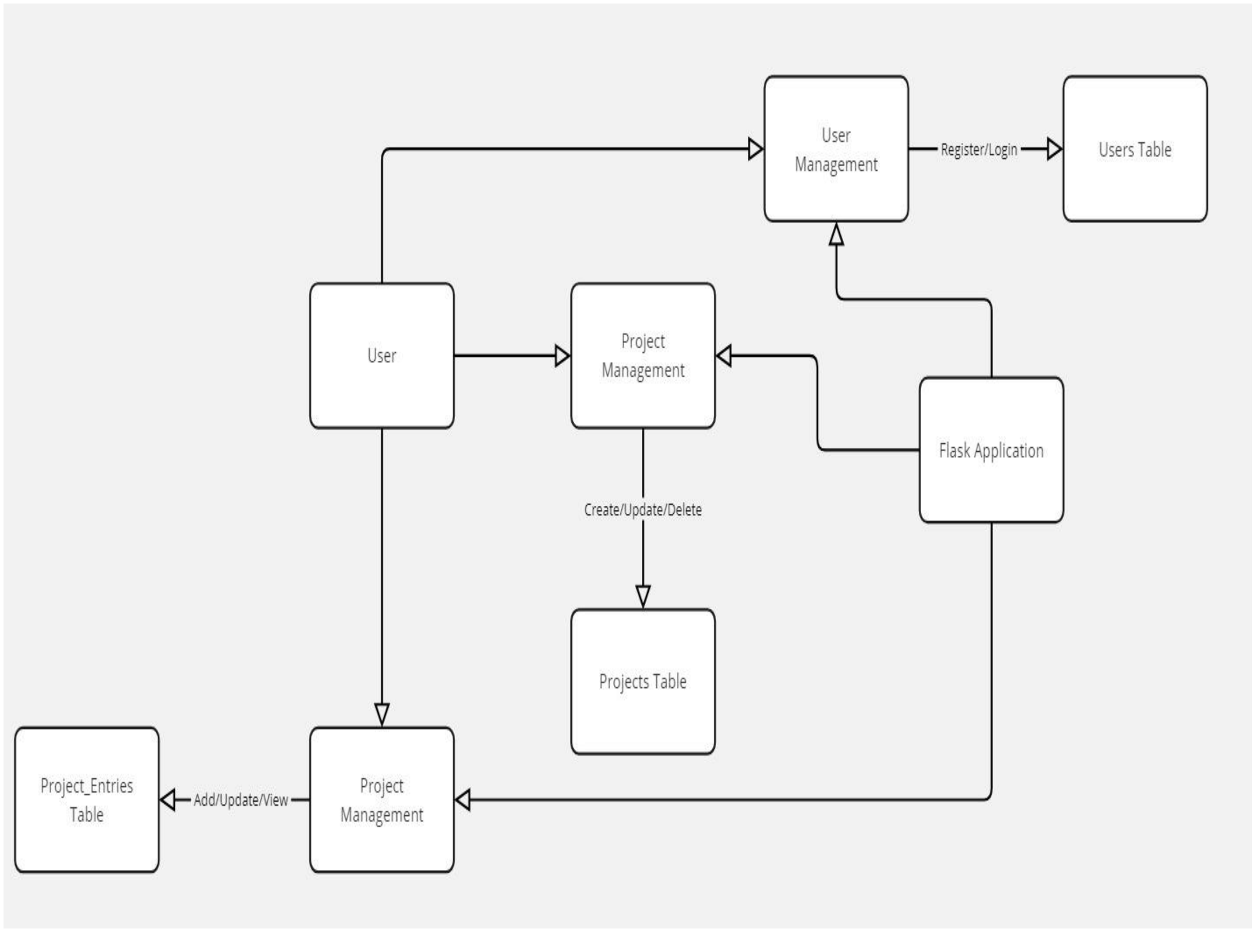


Figure 3.2 DFD Level 1

## 3.3 Use Case Diagram

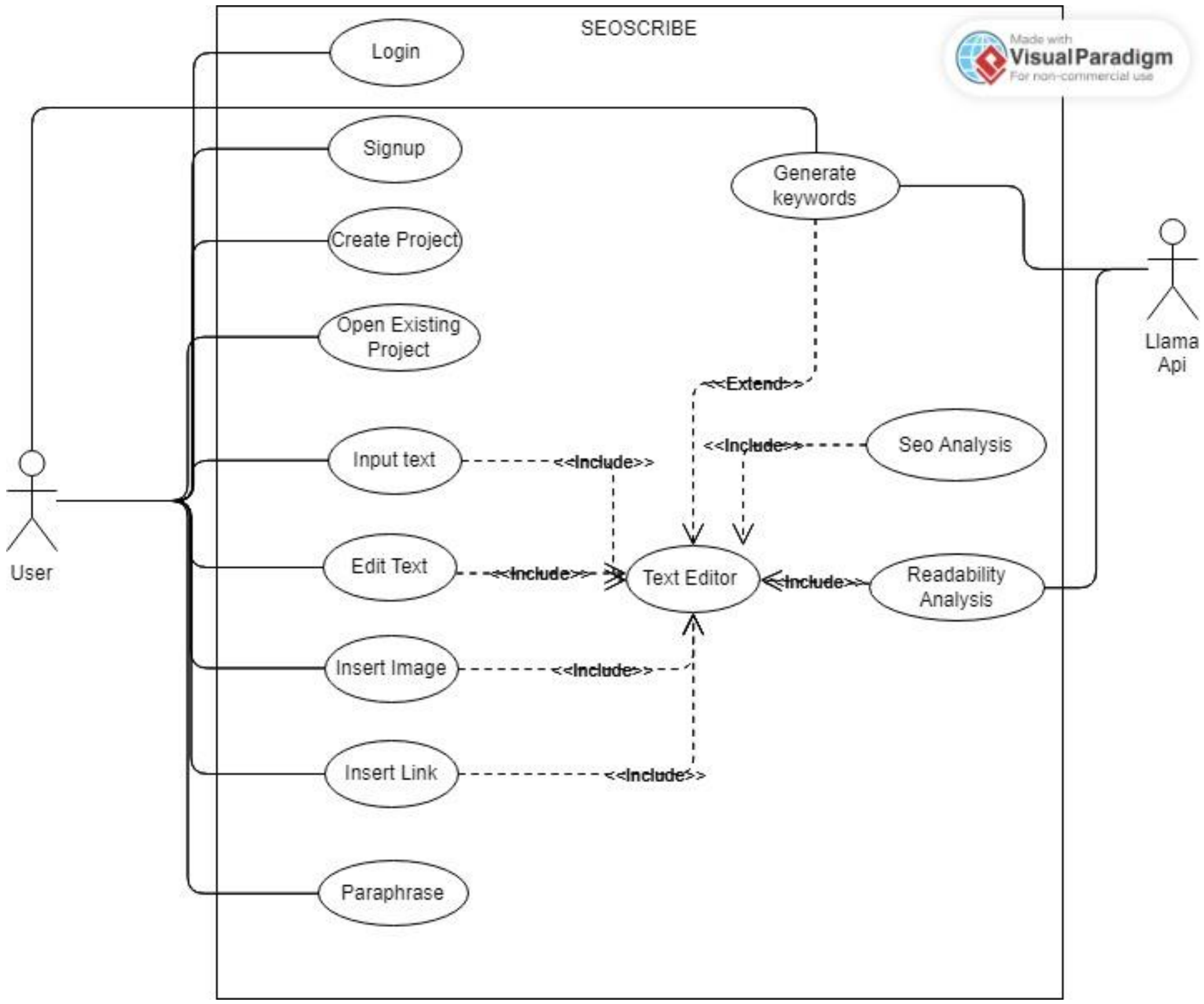


Figure 3.3 Use Case Diagram

## 3.4 Fully dressed Use Cases

### *3.4.1 UC01 Log In*

| | |
|---|---|
| **UC01** | Log In |
| **Actor** | User |
| **Goal** | To log in to SeoScribe app |
| **Precondition** | 1. User must have an internet connection<br>2. User must have a registered account in the app |
| **Basic Flow** | 1. User opens SeoScribe app<br>2. User clicks on login button<br>3. User is redirected to log in page and prompted to enter email and password<br>4. User enter his email and password<br>5. System validates the entered credentials with the database |
| **Alternate Flow** | 1. User enter email and password<br>1.1. System crosschecks the credentials with the database<br>1.2. Email is not found<br>1.3. System displays an error 'invalid email' prompting user to renter his email<br>2. User enter email and password<br>2.1. System crosschecks the credentials with the database<br>2.2. Password is not matched with associated email<br>2.3. System display an error 'invalid password' prompting user to re-enter his password |
| **Postcondition** | 1. User is redirected to SeoScribe dashboard |

Table 3.1 UC01

### *3.4.2 UC02 Sign up*

| | |
|---|---|
| **UC02** | Sign Up |
| **Actor** | User |
| **Goal** | To create an account in the app to access the features |
| **Precondition** | 1. User must be connected to the internet |
| **Basic Flow** | 1. User opens the app<br>2. User clicks on the sign up button<br>3. User is redirected to sign up page and is asked to enter his email, password and username<br>4. User input his credentials<br>5. User clicks sign up<br>6. System creates an entry in database and stores user's password and username against the corresponding email |
| **Alternate Flow** | 1. User enter an invalid email<br>2. System displays an error message (invalid email) |
| **Postcondition** | 1. User account is created in database<br>2. User is redirected to login page |

Table 3.2 UC02

### *3.4.3 UC03 Create a New Project*

| | |
|---|---|
| **UC03** | Create a New Project |
| **Actor** | User |
| **Goal** | To create a new project |
| **Precondition** | 1. User must be logged in |
| **Basic Flow** | 1. User clicks writing assistant on the homepage<br>2. User clicks create project on the writing assistant dashboard<br>3. System prompts the user to name the project<br>4. User is displayed the description box to enter a description of project (Optional)<br>5. User inputs project name and description (Optional)<br>6. User clicks next |
| **Alternat Flow** | 1. User cancels the project creation<br>2. User is redirected to the dashboard |
| **Postcondition** | 1. User is redirected to keyword generation page<br>2. A project entry is created in the database |

Table 3.3 UC03

### *3.4.4 UC04 Open an Existing Project*

| | |
|---|---|
| **UC04** | Open an Existing Project |
| **Actor** | User |
| **Goal** | To open an existing project |
| **Precondition** | 2. User must be logged in<br>3. User must have created a project previously |
| **Basic Flow** | 7. User clicks writing assistant on the homepage<br>8. System displays previous projects<br>9. User clicks on a project<br>10. System Loads the project in text editor |
| **Alternat Flow** | 1. system fails to load the project(exception)<br>2. an error message is displayed |
| **Postcondition** | 3. User is redirected text editor |

Table 3.4 UC04

### *3.4.5 UC05 Keyword Generation*

| | |
|---|---|
| **UC05** | Keyword Generation |
| **Actor** | User |
| **Goal** | To generate keywords |
| **Precondition** | 1. User must be logged in<br>2. User must have an internet connection |
| **Basic Flow** | 1. User clicks on the keyword tool on homepage<br>2. User is redirected to the keyword generation page<br>3. User is prompted to to enter the focus keyword<br>4. User enters the keyword<br>5. System generate at least long tail keywords and LSI keywords |
| **Alternate Flow** | 1. User enters keyword longer than 60 characters<br>1.1. System displays a warning prompting to reduce character length to 60<br>2. Api fails to respond<br>2.1. An error is displayed asking user to check his internet connection |
| **Postcondition** | 1. Long tail and LSI keywords are generated and displayed |

Table 3.5 UC05

### 3.4.6 UC06 Text Editor

| | |
|---|---|
| **UC06** | Text Editor |
| **Actor** | User |
| **Goal** | To open a created project in the text editor |
| **Precondition** | 1. User must have an internet connection<br>2. User must have created a project |
| **Basic Flow** | 1. On the project creation window user clicks next<br>2. System must redirect the user to the keyword generation page<br>3. User follows steps in the UC 05<br>4. After generating keywords user is displayed a next button<br>5. User clicks next |
| **Alternate Flow** | 1. user cancels the operation<br>2. system reverts back to the writing assistant dashboard<br>3. changes made are discarded |
| **Postcondition** | 1. Project entry in database is updated with the now generated keywords<br>2. User is redirected to the text editor<br>3. Keywords are displayed in the sidebar |

Table 3.6 UC06

### 3.4.7 UC07 Input text

| | |
|---|---|
| **UC07** | Input text |
| **Actor** | User |
| **Goal** | To input text into the text editor |
| **Precondition** | 1. User have completed steps in UC03 and UC05<br>2. User is on text editor page |
| **Basic Flow** | 1. User writes into the text editor<br>2. System accepts the text input<br>3. System identify text as either body or heading<br>4. System formats the text accordingly |
| **Alternate Flow** | 1. System fails to accept input (exception)<br>2. system displays an exception error |
| **Postcondition** | 1. Text is displayed and formatted accordingly |

Table 3.7 UC07

### *3.4.8 UC08 Edit Text*

| | |
|---|---|
| **UC08** | Edit Text |
| **Actor** | User |
| **Goal** | Perform text editing operations |
| **Precondition** | 3. User has text inside the text editor<br>4. User is on text editor page |
| **Basic Flow** | 5. User selects text<br>6. User selects an editing operation (i.e., font selection, bold, italic, underline, ordered list, numbered list, align, spacing etc)<br>7. The selected operation is applied on the selected text<br>8. System displays the results |
| **Alternate Flow** | 3. User deselect the text<br>4. System reverts back to the editor |
| **Postcondition** | 2. Selected operation is applied to the text |

Table 3.8 UC08

### *3.4.9 UC09 SEO Analysis*

| | |
|---|---|
| **UC09** | SEO Analysis |
| **Actor** | User |
| **Goal** | To get SEO analysis of the text |
| **Precondition** | 1. Steps in UC 03,04 and 05 must be completed |
| **Basic Flow** | 1. User input text into the editor<br>2. System scans the headings for long tail keywords<br>3. System highlights the keywords matched as green in the sidebar<br>4. System scans the paragraphs for LSI keywords<br>5. System highlights matching keywords as green in the sidebar<br>6. User makes changes to text inside the text editor<br>7. System performs steps 2 to 5 in UC 09 |
| **Alternate Flow** | 1. System finds no matches for a keyword<br>a. Keyword colour in sidebar remains unchanged |
| **Postcondition** | 1. Keywords found in text are highlighted as green to indicate presence in text |

Table 3.9 UC09

### *3.4.10 UC010 Readability Analysis*

| | |
|---|---|
| **UC010** | Readability analysis |
| **Actor** | User |
| **Goal** | To get readability analysis |
| **Precondition** | 2. Steps in UC 03,04 and 05 must be completed<br>3. User has text in the editor |
| **Basic Flow** | 1. User paste text into the text editor<br>2. System identifies long sentences and difficult vocabulary<br>3. System highlights the identified issues as light red in the sidebar under readability section<br>4. User makes changes to the corresponding sentence or vocabulary<br>5. System scans the text<br>6. System highlights the resolved issues as green in sidebar<br>7. User types in the text editor<br>8. System performs steps 2 to 6 in UC010 |
| **Alternate Flow** | 1. User makes changes to a highlighted issue (sentence or vocabulary)<br>a. Corresponding issue is not fit on gunning fox index readability metrics<br>b. Issue colour remains unchanged in sidebar |
| **Postcondition** | 1. Resolved issues are highlighted as green in sidebar<br>2. Unresolved issues remain red in sidebar |

Table 3.10 UC10

### 3.4.11 UC011 Keyword Density Monitor

| **UC09** | Keyword density monitor |
|---|---|
| **Actor** | User |
| **Goal** | To remove keyword stuffing |
| **Precondition** | 1. User has text in the text editor<br>2. Text is formatted<br>3. User has incorporated keywords into the text |
| **Basic Flow** | 1. User clicks check keyword density button in the sidebar<br>2. System displays a drop-down menu displaying all the keywords and LSI<br>3. User selects a keyword<br>4. System scans the text<br>5. System detects occurrences of the keyword throughout the text<br>6. System calculates keyword density based on occurrences/ total word count * 100 |
| **Alternate Flow** | 1. Corresponding keyword is not found in text<br>2. Keyword colour remain unchanged |
| **Postcondition** | 1. System highlights keywords with density greater than 2% as red in the text editor |

Table 3.11 UC11

### *3.4.12 UC12 Paraphrasing*

| **UC12** | Paraphrasing |
|---|---|
| **Actor** | User |
| **Goal** | To paraphrase text |
| **Precondition** | 1. User must have an internet connection |
| **Basic Flow** | 1. User clicks paraphraser on homepage<br>2. User is redirected to the paraphraser page<br>3. User inputs some text<br>4. User clicks on the paraphrase button<br>5. System paraphrases the text |
| **Alternate Flow** | 1. User cancels the operation<br>2. System reverts back to the editor |
| **Postcondition** | 1. System displays results |

Table 3.12 UC12

### 3.4.13 UC13 Image Insertion

| | |
|---|---|
| **UC13** | Image insertion |
| **Actor** | User |
| **Goal** | To insert an image at cursor position |
| **Precondition** | 1. User is on the text editor page |
| **Basic Flow** | 1. User selects the insert image icon from the toolbar<br>2. System opens a prompt providing option to browse for image<br>3. User selects an image and click insert<br>4. System prompts user to a window asking to modify image size<br>5. System provides option on the same window to insert alt text for the image<br>6. User inputs the image size and alt text |
| **Alternate Flow** | 1. System fails to insert image<br>2. An error is displayed "image insertions failed, try again" |
| **Postcondition** | 1. System inserts the image at the cursor position |

Table 3.13 UC13

### *3.4.14 UC14 Hyperlink insertion*

| | |
|---|---|
| **UC14** | Hyperlink insertion |
| **Actor** | User |
| **Goal** | Insert a hyperlink |
| **Precondition** | 1. User is on the text editor page<br>2. User has text inside the editor |
| **Basic Flow** | 1. User selects text<br>2. User clicks hyperlink icon from toolbar<br>3. System opens a window prompting user to provide a link<br>4. User inputs the link and clicks done<br>5. System associates the link with the selected text |
| **Alternate Flow** | 1. User cancels the operation<br>2. System reverts back to editor |
| **Postcondition** | 1. Selected text is displayed as anchor text for the link |

Table 3.14 UC14

## 3.5 System Sequence Diagram (SSD)

**Log In:**

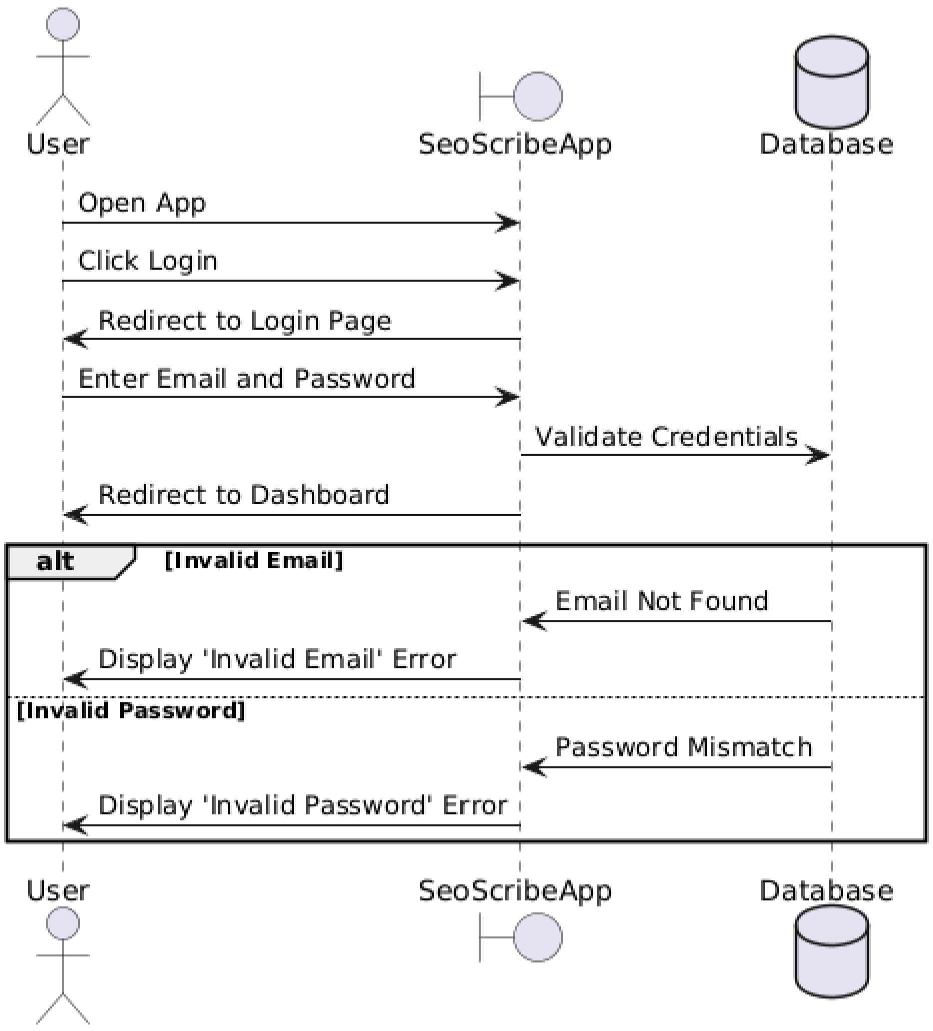


Figure 3.4 SSD - Login

**Sign up:**

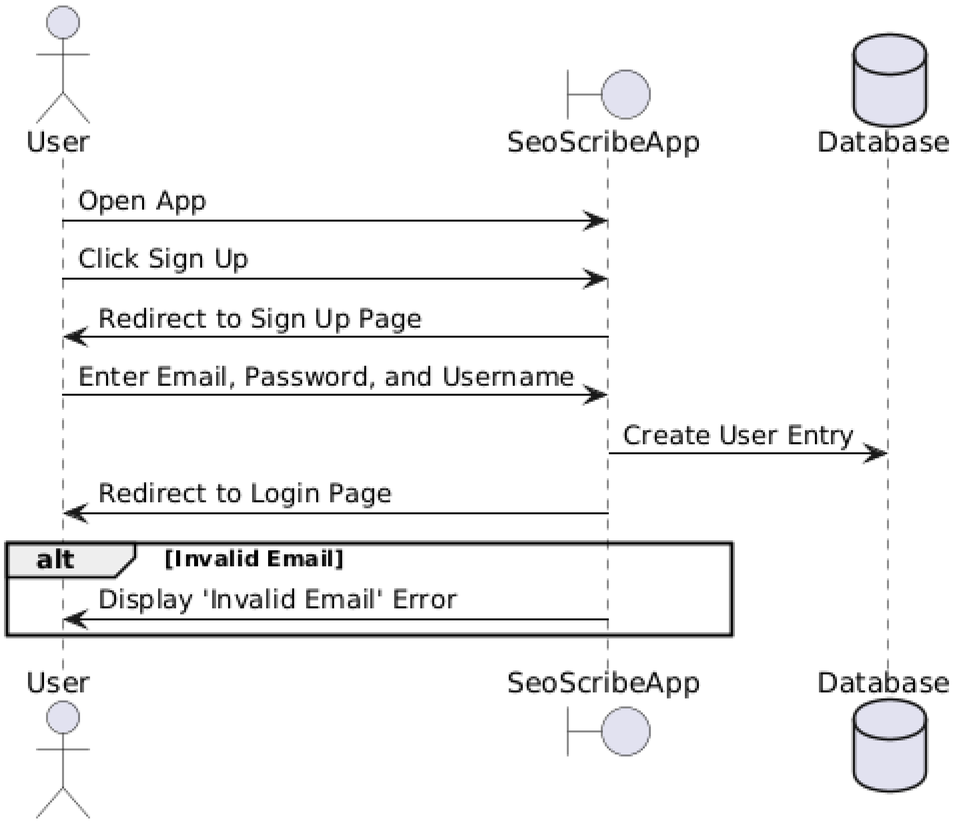


Figure 3.5 SSD - Sign Up

**Project Creation:**

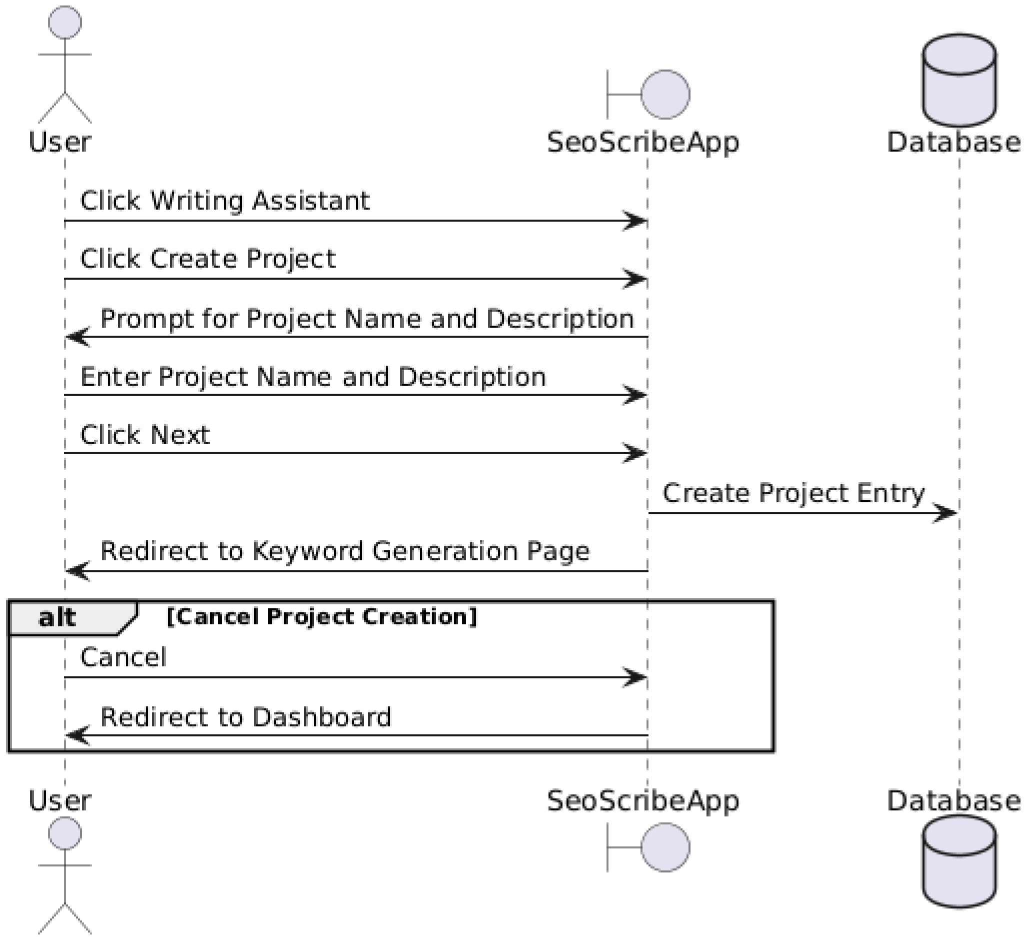


Figure 3.6 SSD - Create Project

**Open Project:**

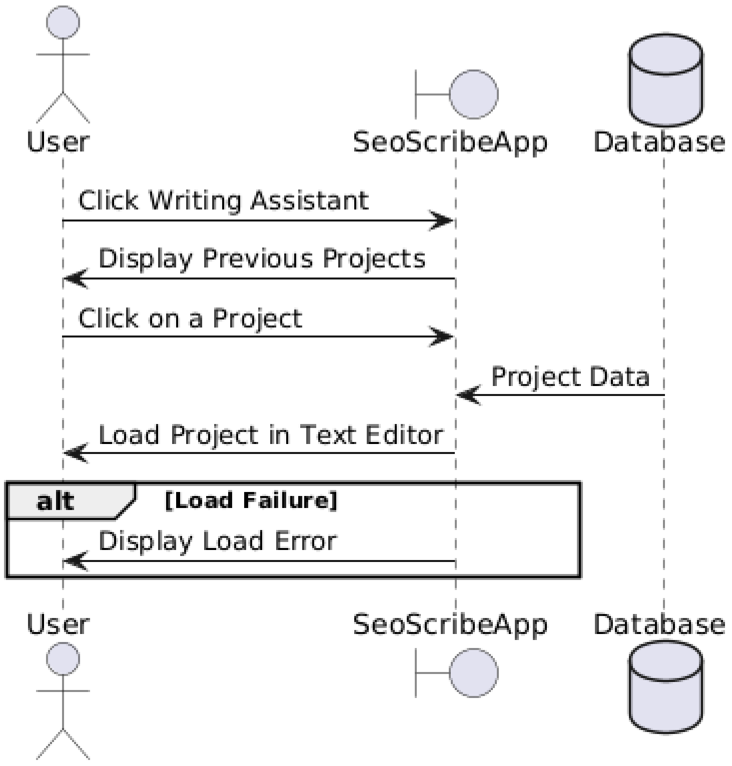


Figure 3.7 SSD - Open Project

**Generate Keywords:**

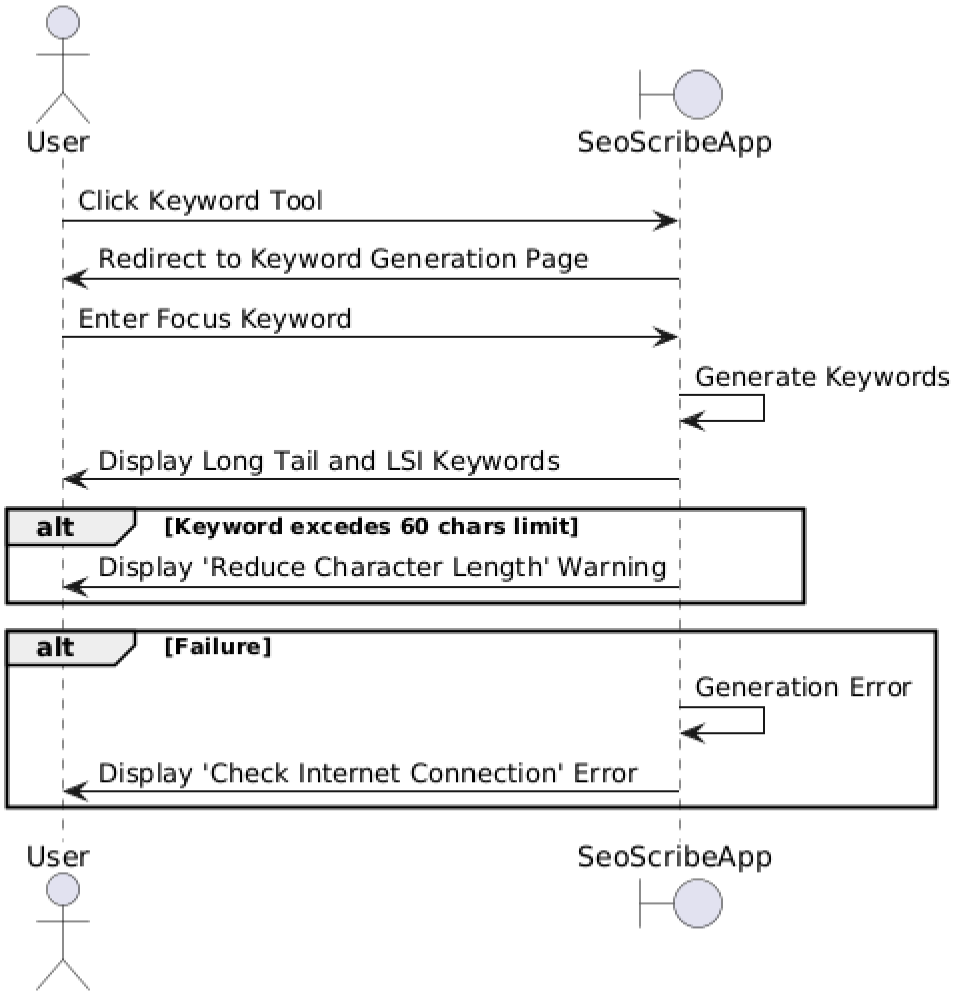

User
SeoScribeApp
Click Keyword Tool
Redirect to Keyword Generation Page
Enter Focus Keyword
Generate Keywords
Display Long Tail and LSI Keywords
alt
[Keyword excedes 60 chars limit]
Display 'Reduce Character Length' Warning
alt
[Failure]
Generation Error
Display 'Check Internet Connection' Error
User
SeoScribeApp

Figure 3.8 SSD - Generate Keywords

**Text Editor:**

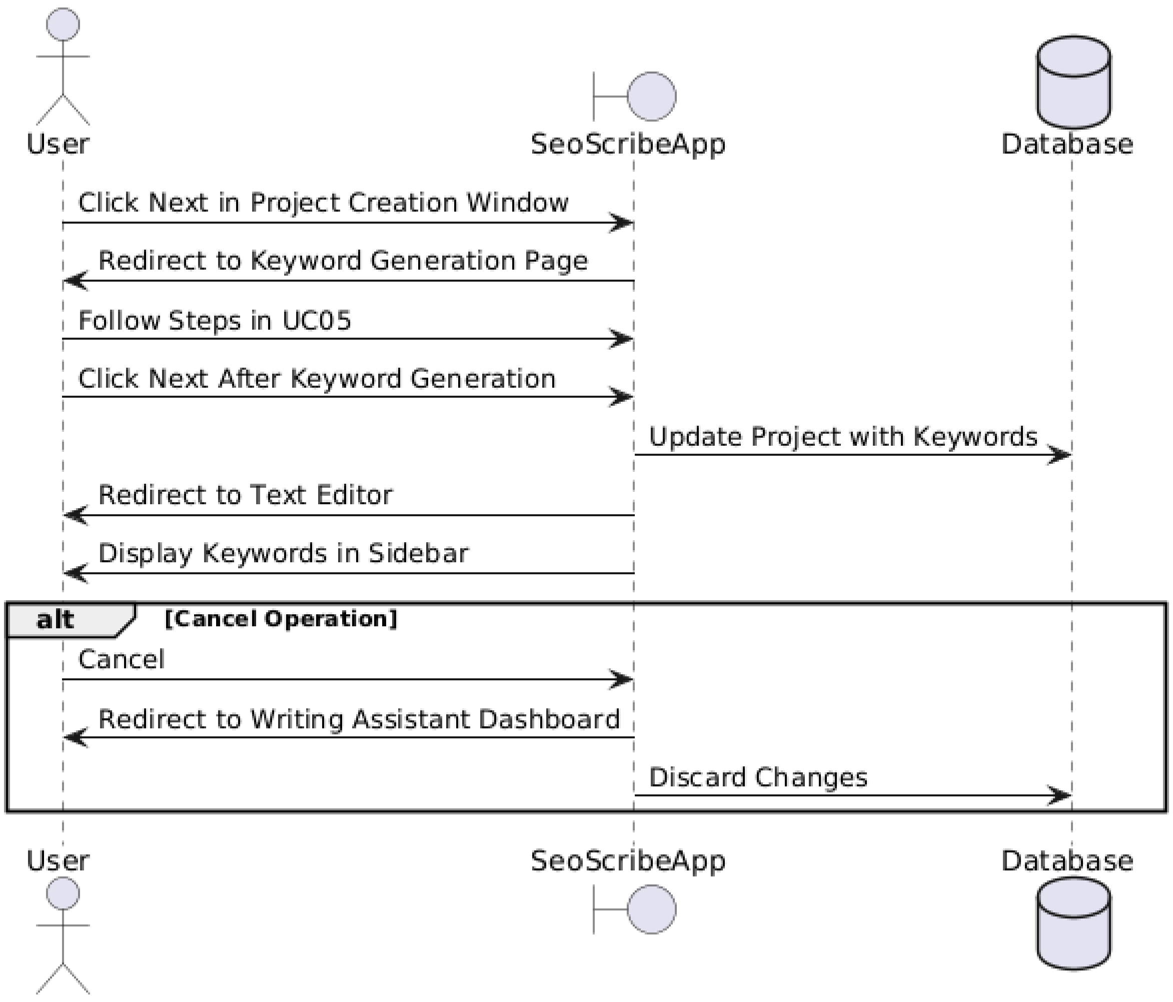

User
SeoScribeApp
Database
Click Next in Project Creation Window
Redirect to Keyword Generation Page
Follow Steps in UC05
Click Next After Keyword Generation
Update Project with Keywords
Redirect to Text Editor
Display Keywords in Sidebar
alt
[Cancel Operation]
Cancel
Redirect to Writing Assistant Dashboard
Discard Changes
User
SeoScribeApp
Database

Figure 3.9 SSD - Text Editor

**Input Text:**

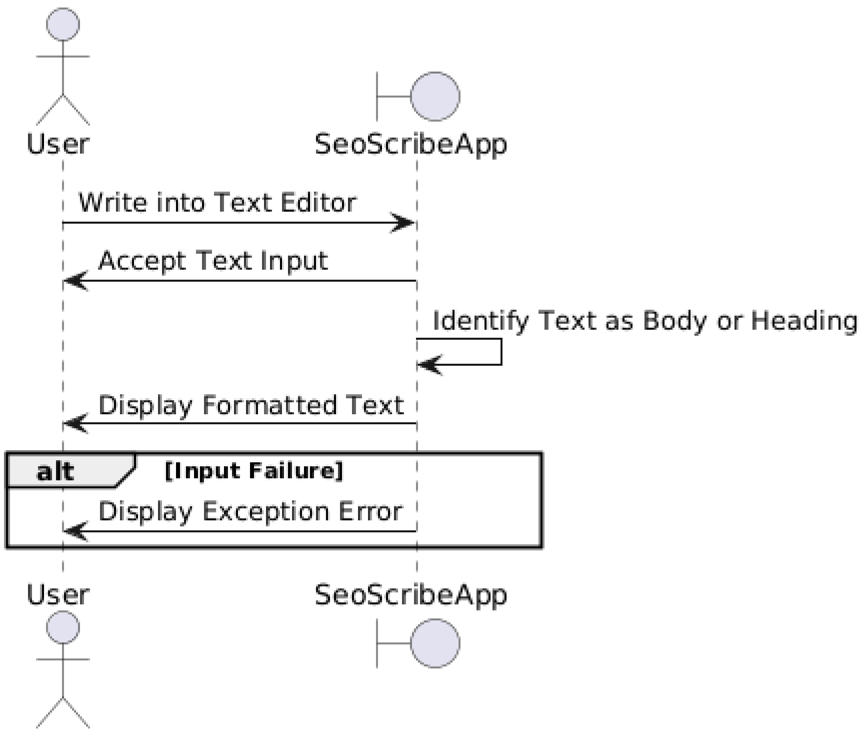


Figure 3.10 SSD - Input Text

**Edit Text:**

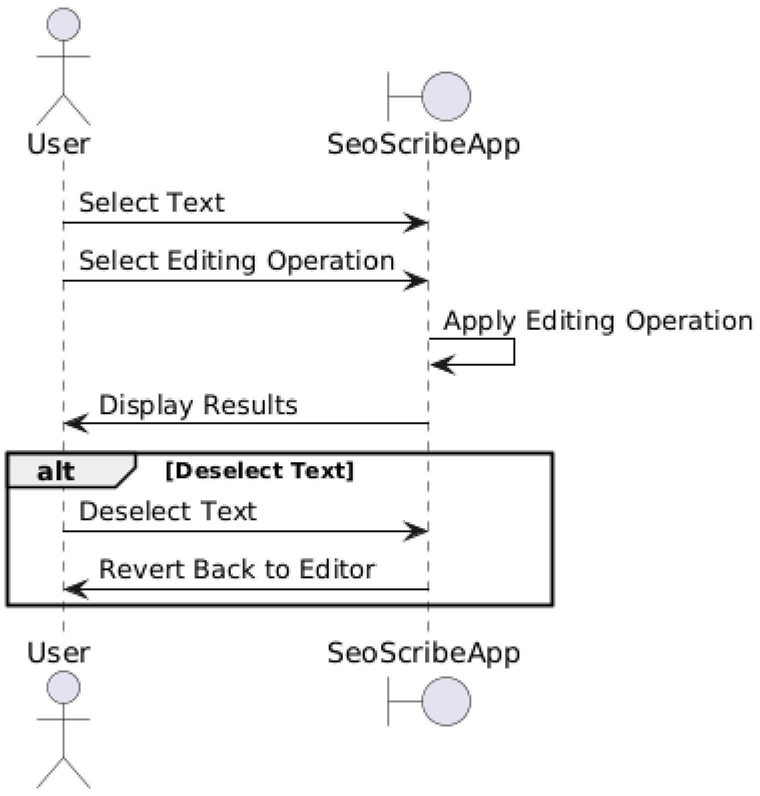


Figure 3.11 SSD - Edit Text

**Seo Analysis:**

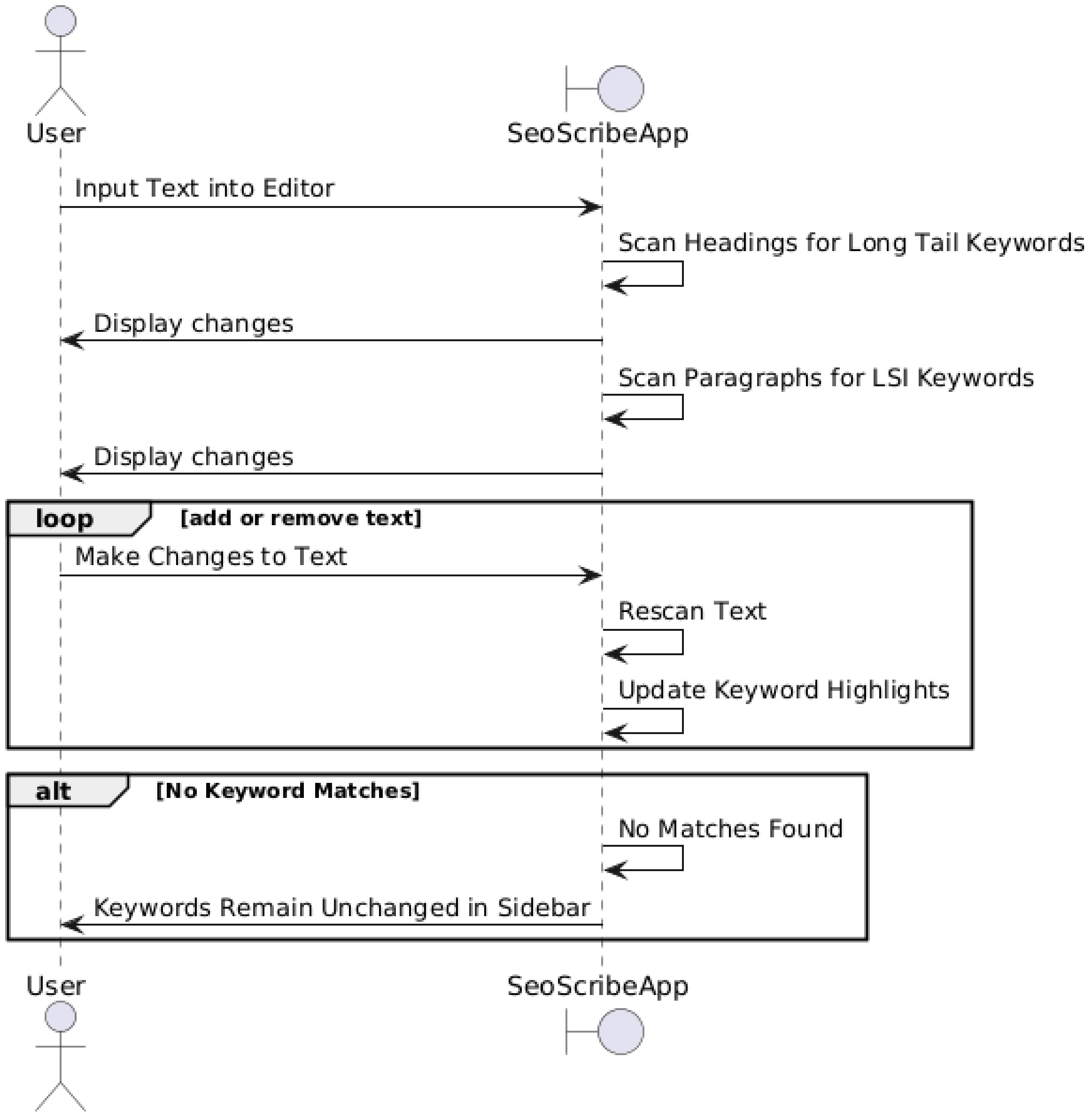


Figure 3.12 SSD - Seo Analysis

**Readability Analysis:**

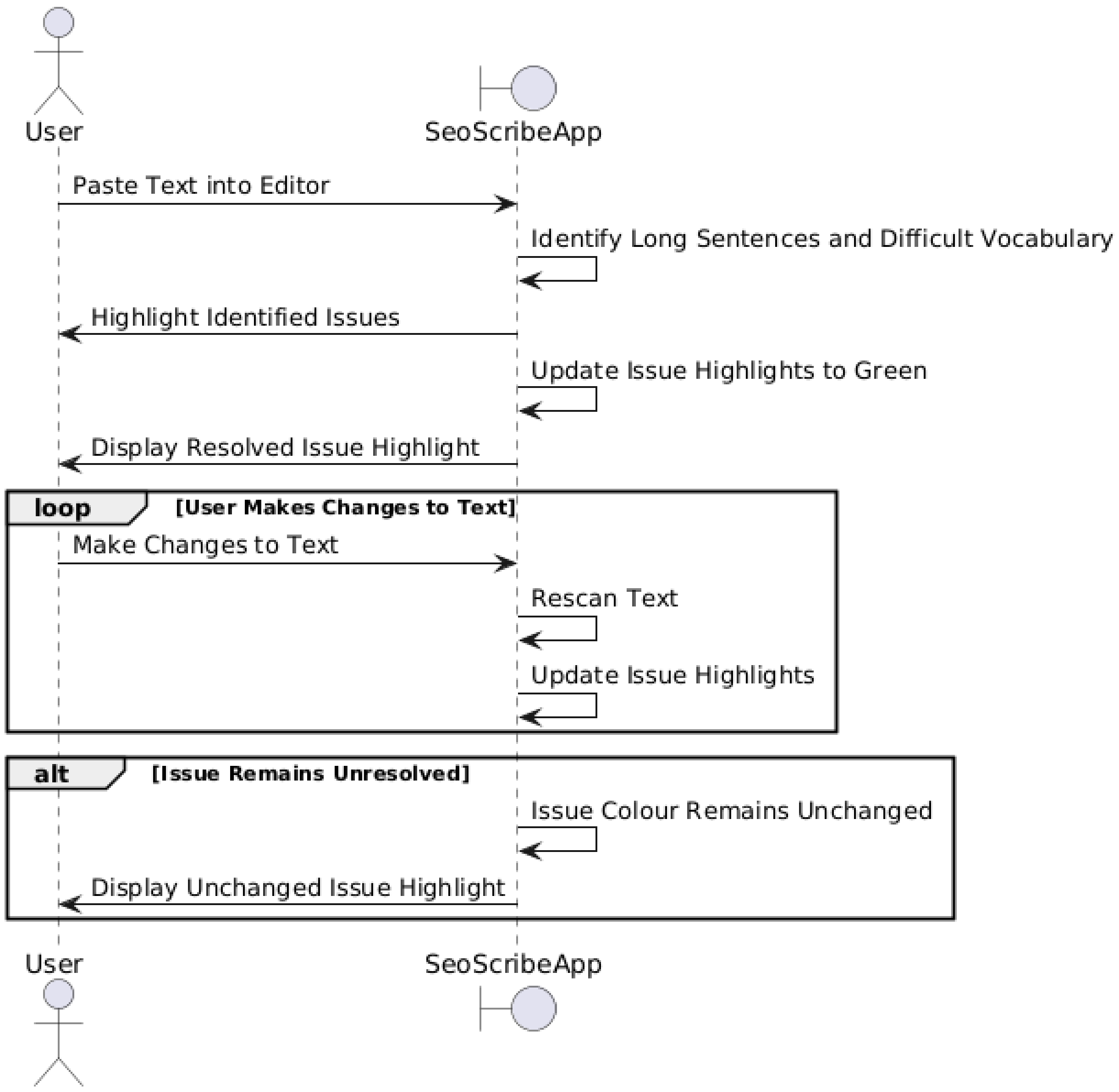

User
SeoScribeApp
Paste Text into Editor
Identify Long Sentences and Difficult Vocabulary
Highlight Identified Issues
Update Issue Highlights to Green
Display Resolved Issue Highlight
loop
[User Makes Changes to Text]
Make Changes to Text
Rescan Text
Update Issue Highlights
alt
[Issue Remains Unresolved]
Issue Colour Remains Unchanged
Display Unchanged Issue Highlight
User
SeoScribeApp

Figure 3.13 SSD - Readability Analysis

**Check Keyword Density:**

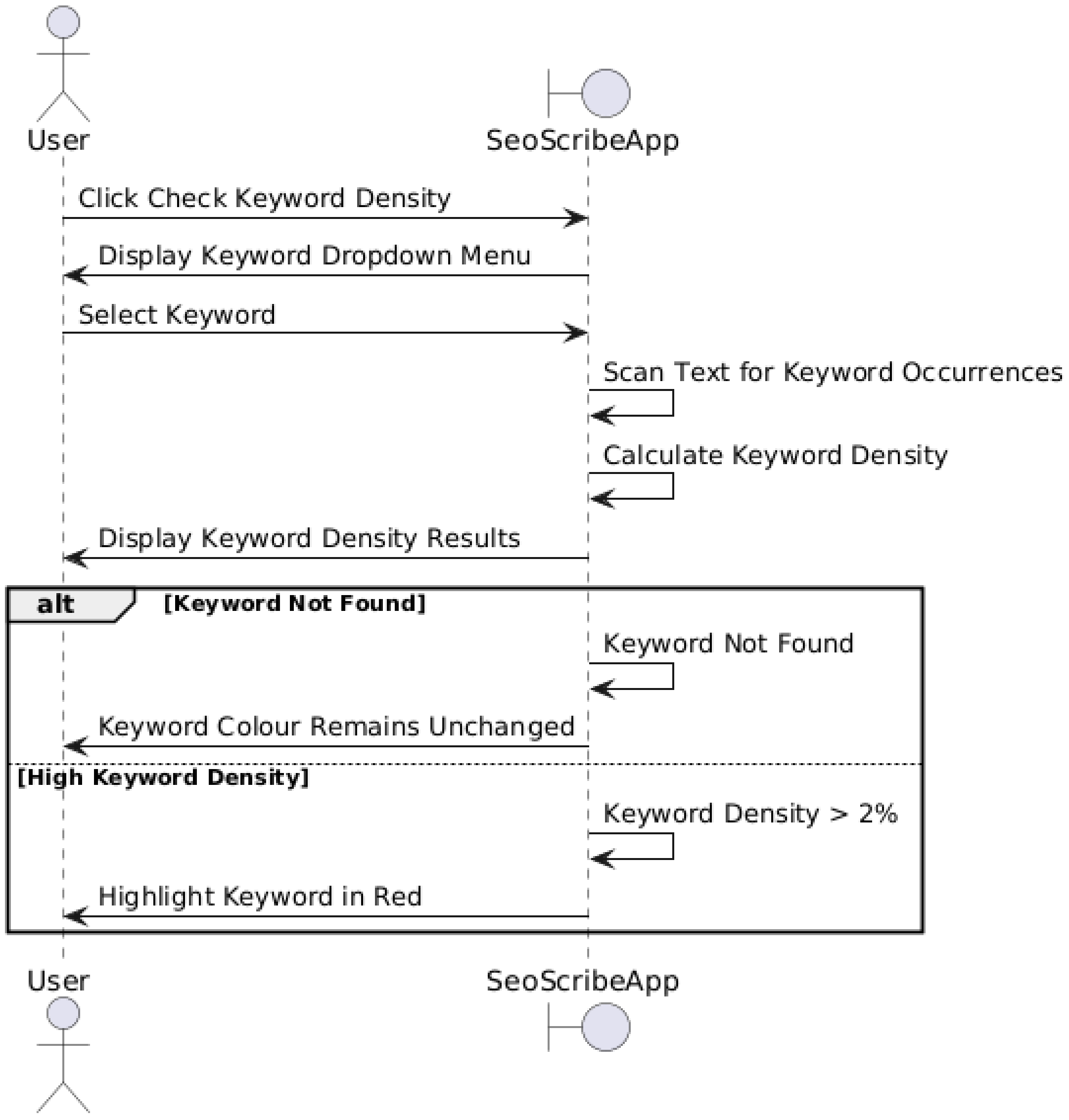

User
SeoScribeApp
Click Check Keyword Density
Display Keyword Dropdown Menu
Select Keyword
Scan Text for Keyword Occurrences
Calculate Keyword Density
Display Keyword Density Results
alt
[Keyword Not Found]
Keyword Not Found
Keyword Colour Remains Unchanged
[High Keyword Density]
Keyword Density > 2%
Highlight Keyword in Red
User
SeoScribeApp

Figure 3.14 SSD - Check Keyword Density

**Paraphrase:**

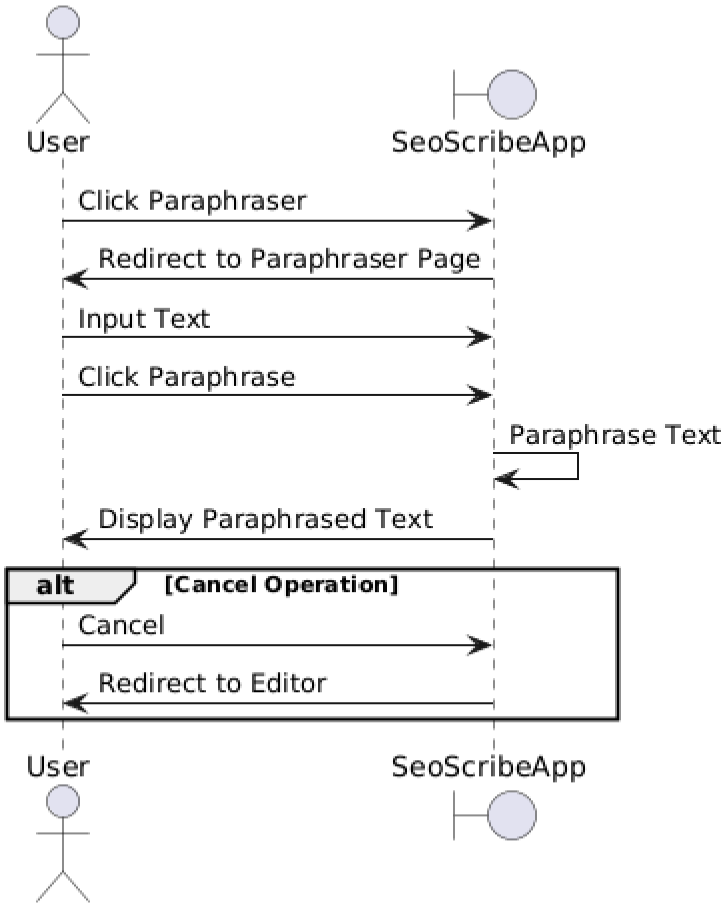


Figure 3.15 SSD - Paraphrase

**Insert Image:**

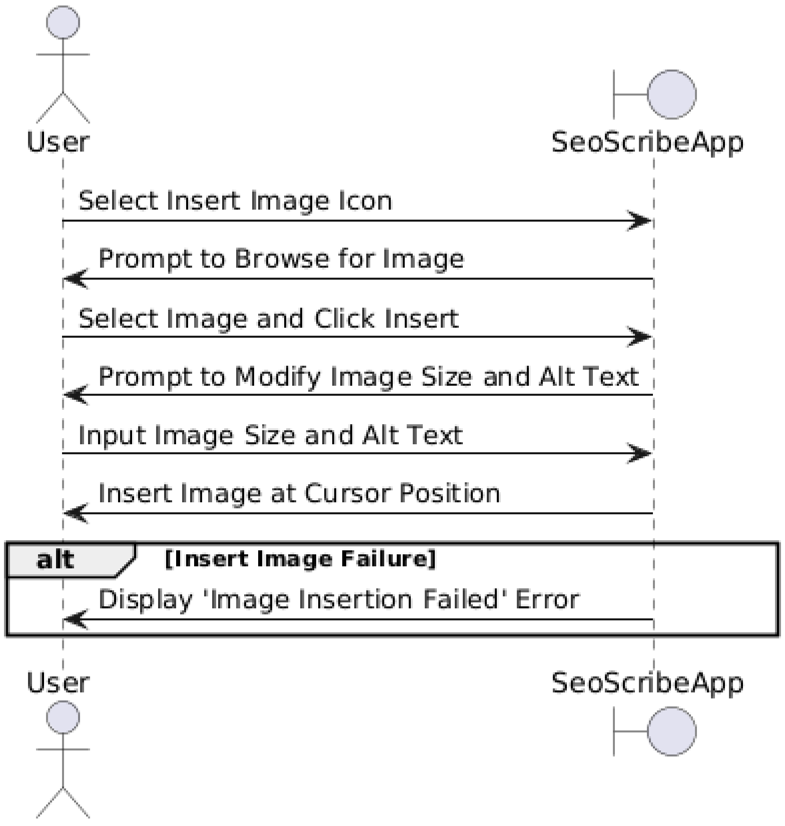

User
SeoScribeApp
Select Insert Image Icon
Prompt to Browse for Image
Select Image and Click Insert
Prompt to Modify Image Size and Alt Text
Input Image Size and Alt Text
Insert Image at Cursor Position
alt
[Insert Image Failure]
Display 'Image Insertion Failed' Error
User
SeoScribeApp

Figure 3.16 SSD - Insert Image

**Insert Link:**

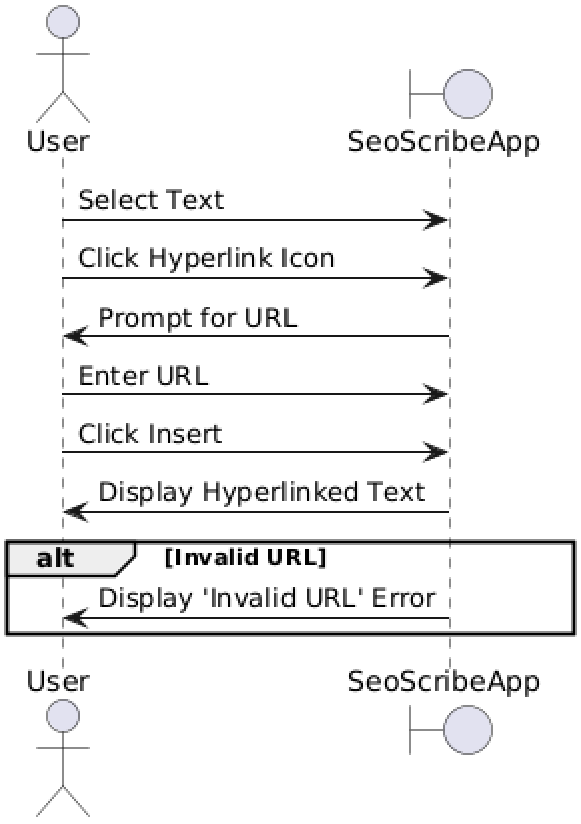


Figure 3.17 SSD - Insert Link

## 3.6 Domain Model

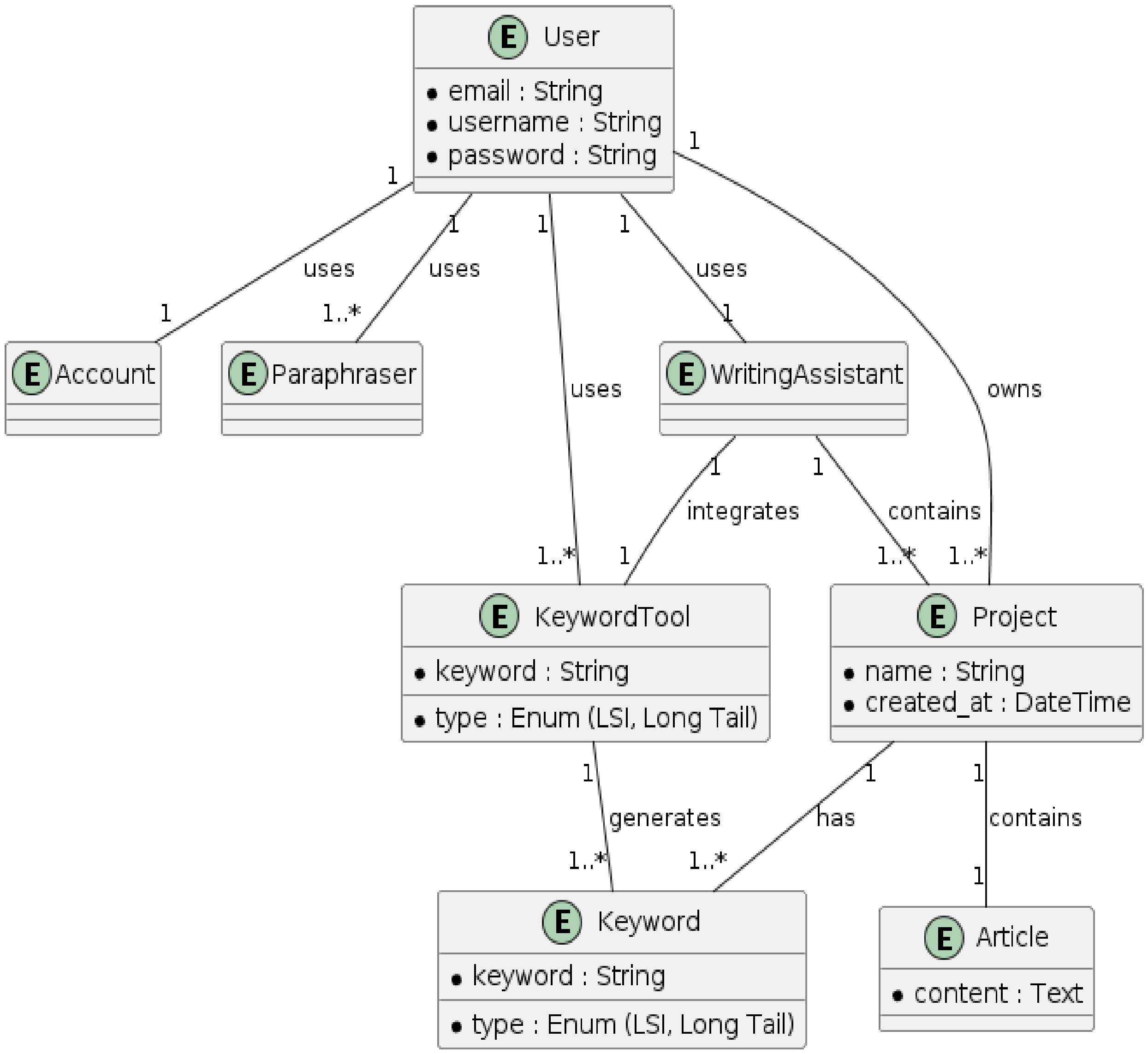


Figure 3.18 Domain Model

## 3.7 Software Development Plan

### *3.7.1 Software Architecture*

#### *3.7.1.1 Architecture Description*

The Three-Tier Architecture provides a structured and scalable approach for SEOscribe. It accommodates the frontend, backend, and storage requirements efficiently. the system is classified into two main components i.e., client, that requests services or resources on the presentation layer (browser), and the server provides those requested services or resources to the client.

**Presentation Layer (Frontend):**

**Responsibility:** Handles user interface and user interactions.

**Components:**

- HTML, CSS, JavaScript for the user interface.

**Application Layer (Backend):**

**Responsibility:** Manages application logic, processes requests, and communicates with the data layer.

**Components:**

- Python backend using a web framework (Flask).
- NLP libraries for text analysis and semantic understanding.
- Llama 3 API integration for advanced natural language processing.

**Data Layer (Storage):**

**Responsibility:** Manages data storage and retrieval.

**Components:**

- Database (SQLite) for storing textual content, user data, and related information.

### 3.7.1.2 *Flow Chart*

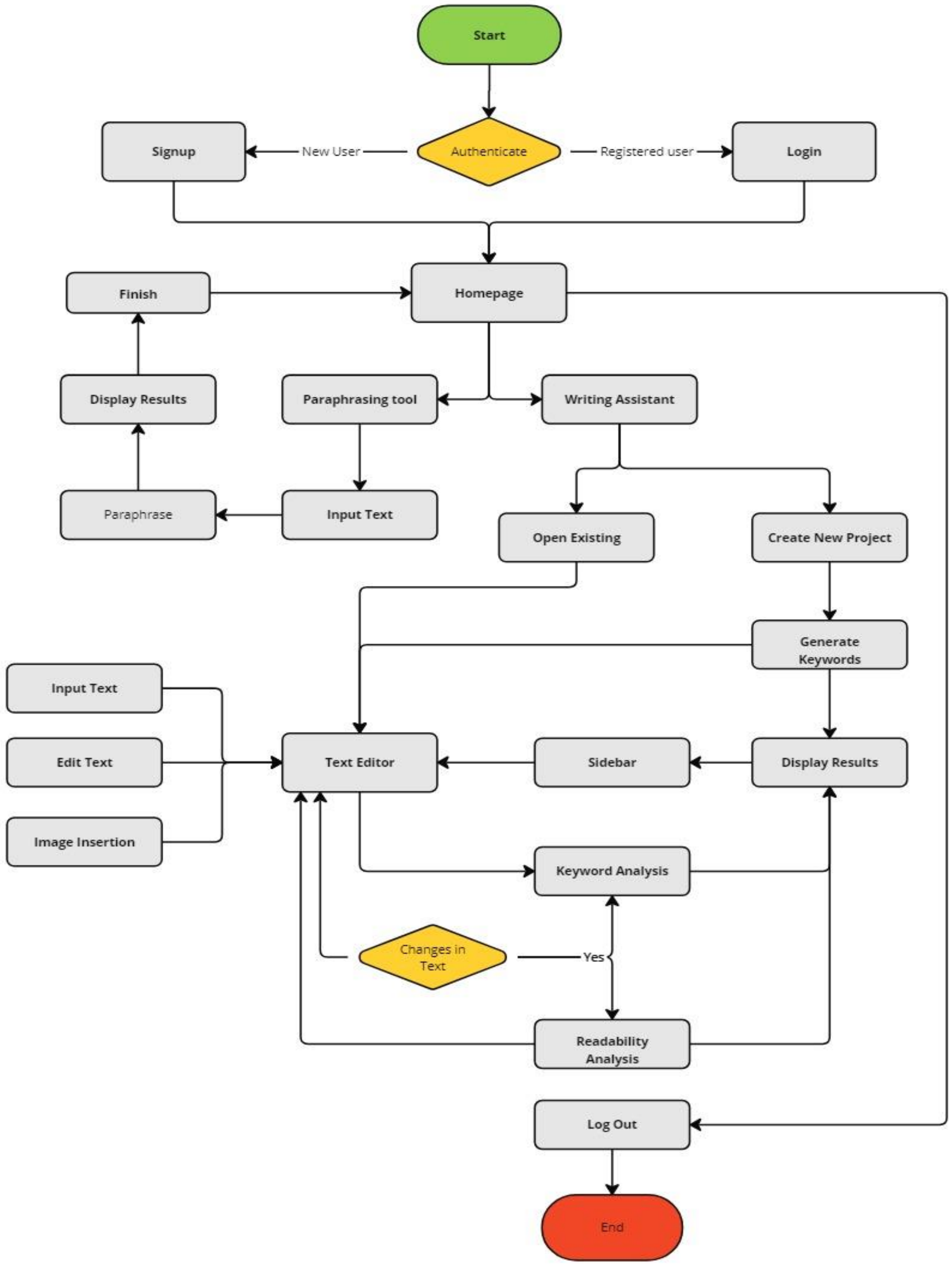


Figure 3.19 Flow Chart

#### 3.7.1.3 *Architecture Diagram*

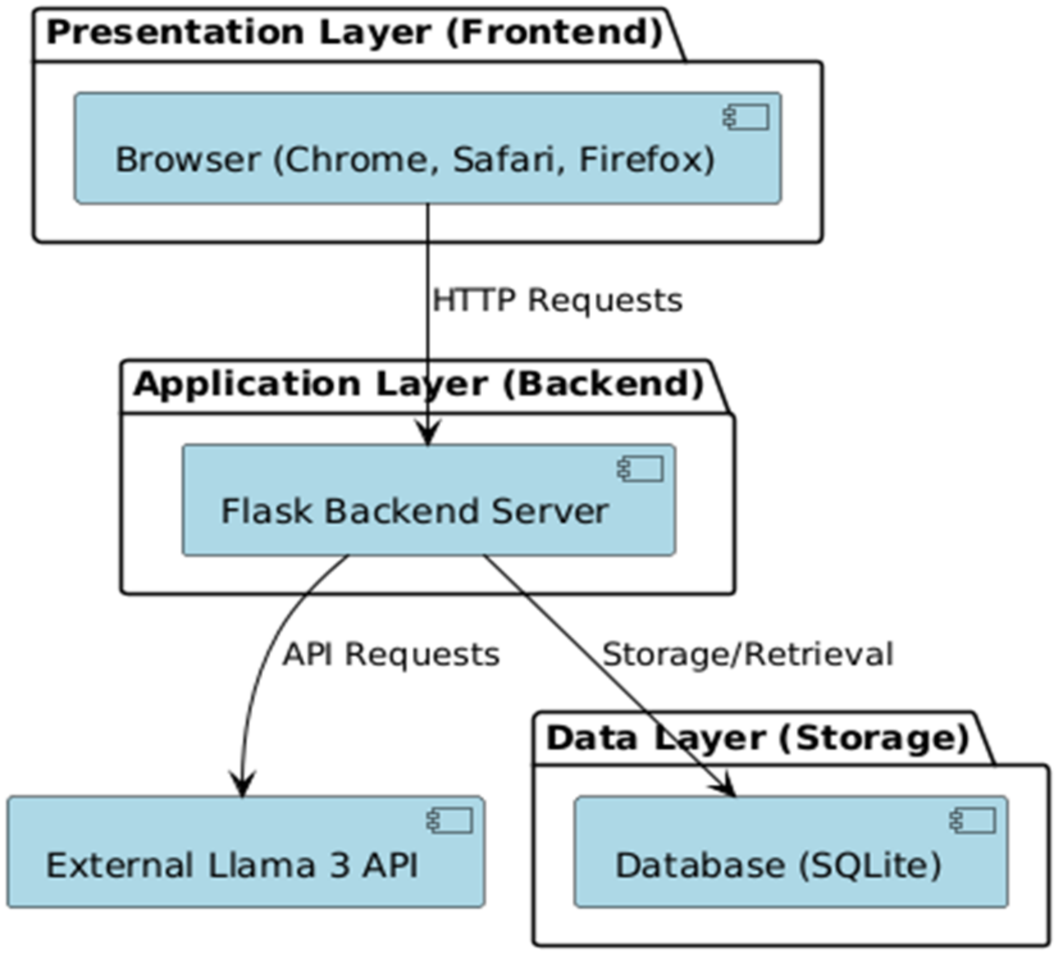


Figure 3.20 Architecture Diagram

### 3.7.2 *Development iterations*

Our project is divided into the following four iterations along with documentation tasks

#### *3.7.2.1 Iteration 1:*

**Core Functionalities Implementation – 1**

In the first iteration we plan on developing the initial features including keyword generation feature and the paraphrasing tool.

**Tasks:**

- Complete prerequisites i.e., Set up the project repository on GitHub, establish a virtual environment for Python, set up the framework (Flask).
- Create UI for homepage, Keyword generation and paraphrasing pages.
- Implement backend logic for the said functionalities.

**Documentation**

- Project Introduction
- Scope
- Objectives
- Background Study
- Use cases
- Sequence Diagrams
- SRS Document
- Software Development Plan

#### *3.7.2.2 Iteration 2:*

**Core Functionalities Implementation – 2**

In the second iteration, we will focus on implementing the writing assistant, user authentication, and SQLite database integration.

**Tasks:**

- Implement user authentication: Create login and signup functionality.
- Develop the writing assistant feature
- Integrate SQLite database: Set up the database, create models for users and projects, and ensure proper data storage and retrieval.

**Documentation:**

- Software Architecture and Flow chart
- Domain Model
- Data flow Diagrams
- Database Schema

#### *3.7.2.3 Iteration 3:*

**Module Integration**

In the third iteration, we will integrate all modules into a cohesive system, including keyword generator integration with the writing assistant, user permissions, and more.

**Tasks:**

- Integrate keyword generator with the writing assistant: Allow keywords to be fetched and displayed in the text editor.
- Implement user permissions: Define user roles and access controls.

**Documentation:**

- Adopted methodology
- Implementation Documentation
- Test Plan and Test cases

### *3.7.2.4 Iteration 4:*

**Testing, Finalizing and Reviewing Documentation**

In the final iteration, we will focus on comprehensive testing, finalizing the project, and reviewing all documentation. Testing will be done using Test cases and Flask integration with pytest using VSCode.

**Tasks:**

- Conduct extensive testing: Write and execute test cases for all functionalities using pytest.
- Fix bugs and issues identified during testing.
- Finalize the project: Ensure all features are working as expected and the system is stable.
- Review and update documentation: Make sure all documentation is accurate and complete.

# Chapter 4 Implementation

## 4.1 Adopted Methodology

Given the nature of our project, which involves the development and integration of multiple features such as a writing assistant, keyword tool, paraphraser, and user authentication, Agile methodology seemed highly suitable.

### Reasons for Choosing Agile:

1. **Iterative Development:**
   Our project includes several distinct features that need to be developed and integrated over time. Agile's iterative approach allows us to break down the project into manageable iterations or sprints. This means we can focus on developing and refining individual features in each iteration, which is particularly useful as we need to integrate and adjust different components like the writing assistant and keyword tool.
2. **Flexibility:**
   One of the core benefits of Agile is its flexibility. As our project progresses, requirements may evolve or new needs might emerge. Agile allows us to adapt to these changes without significant disruption. For instance, if we receive feedback indicating a need for additional functionality or adjustments in the writing assistant or keyword tool, we can easily incorporate these changes in subsequent iterations.
3. **Continuous Feedback:**
   Agile emphasizes regular feedback and review. This means that after each iteration, we can review the developed features, gather feedback, and make necessary improvements. This continuous feedback loop ensures that our features align closely with user needs and expectations, enhancing the overall quality of our project.
4. **Collaboration:**
   Agile promotes close collaboration among team members, which is crucial for our project. Working together frequently and sharing progress helps us identify and resolve issues early, improve the quality of our work, and maintain a shared vision for the project. This collaborative environment supports effective integration of different modules, such as combining the keyword tool with the writing assistant.

5. **Incremental Integration:**

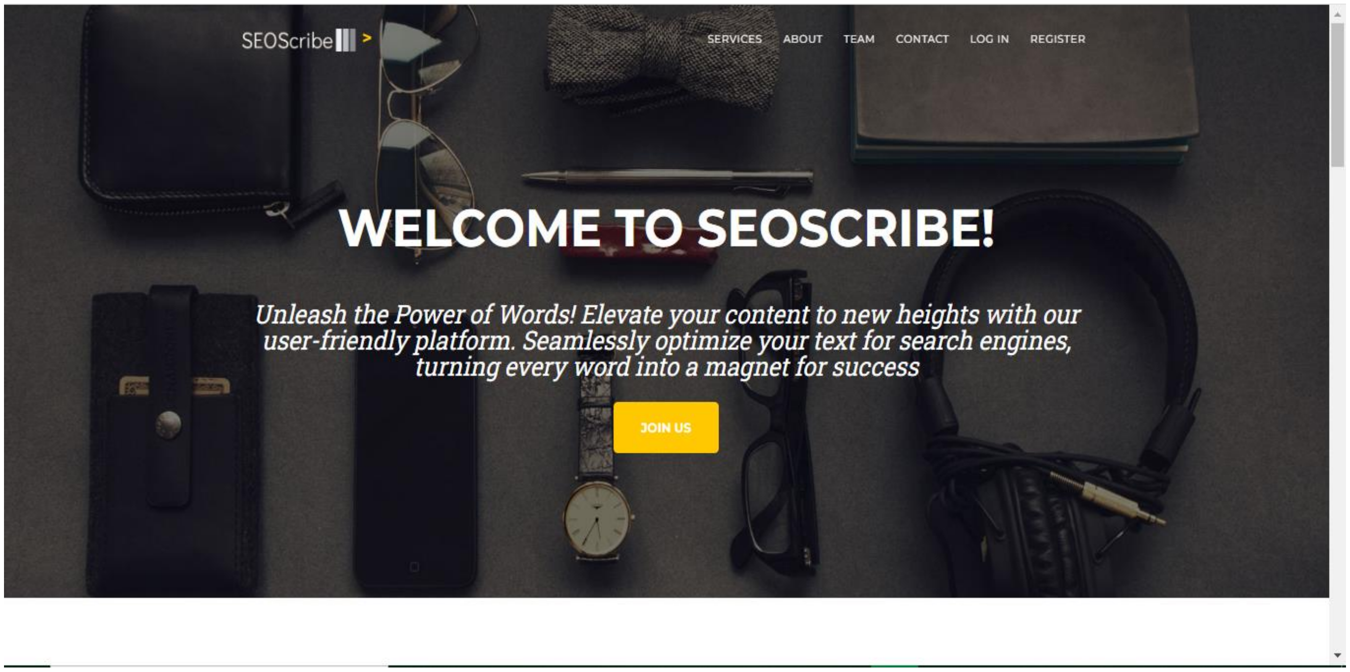
With Agile, we can incrementally integrate features, ensuring that each component works well with the others. This approach is particularly beneficial for our project as we need to integrate the writing assistant with the keyword generator and other features seamlessly. Each iteration allows us to test and integrate components, minimizing risks and ensuring a cohesive system.

Overall, Agile methodology's principles of iterative development, flexibility, and continuous feedback align well with our project's needs. It supports our goal of developing, integrating, and refining features effectively while accommodating changes and ensuring high-quality outcomes.

## 4.2 Interface Details

### *4.2.1 Homepage*

Upon opening the app, user is displayed the homepage. From the homepage user can navigate to different operation such as **Register/Login** and **Services.**



*Unleash the Power of Words! Elevate your content to new heights with our user-friendly platform. Seamlessly optimize your text for search engines, turning every word into a magnet for success*

Figure 4.1 Home Screen Interface

### *4.2.2 Sign up*

In order to create an account in the app, user must navigate to the registration page either through clicking the **join us** button or by clicking the **Register** in the navbar. Upon clicking the mentioned buttons user is redirected to the account creation page and user must fill out the form i.e., Username, email and password. Upon Successful completion user is redirected to the Login page.

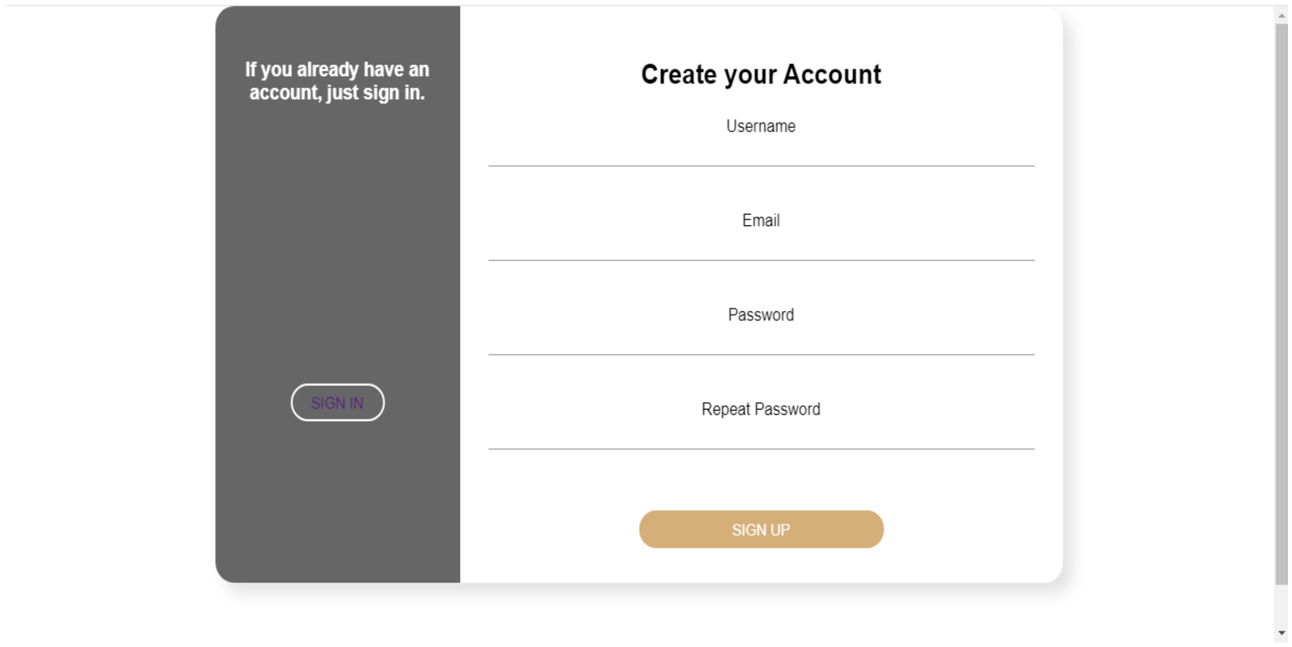


Figure 4.2 Sign Up Interface

### *4.2.3 Log In*

In order to Login, user must navigate to the Login page through clicking the **Login** button in the navbar. Upon clicking the user is redirected to the Login page and user must fill out the form i.e., email and password. Upon successful completion user is redirected to the homepage with a welcome message. (User can use the toggle button to switch between Signup and Login on the User Authentication page)

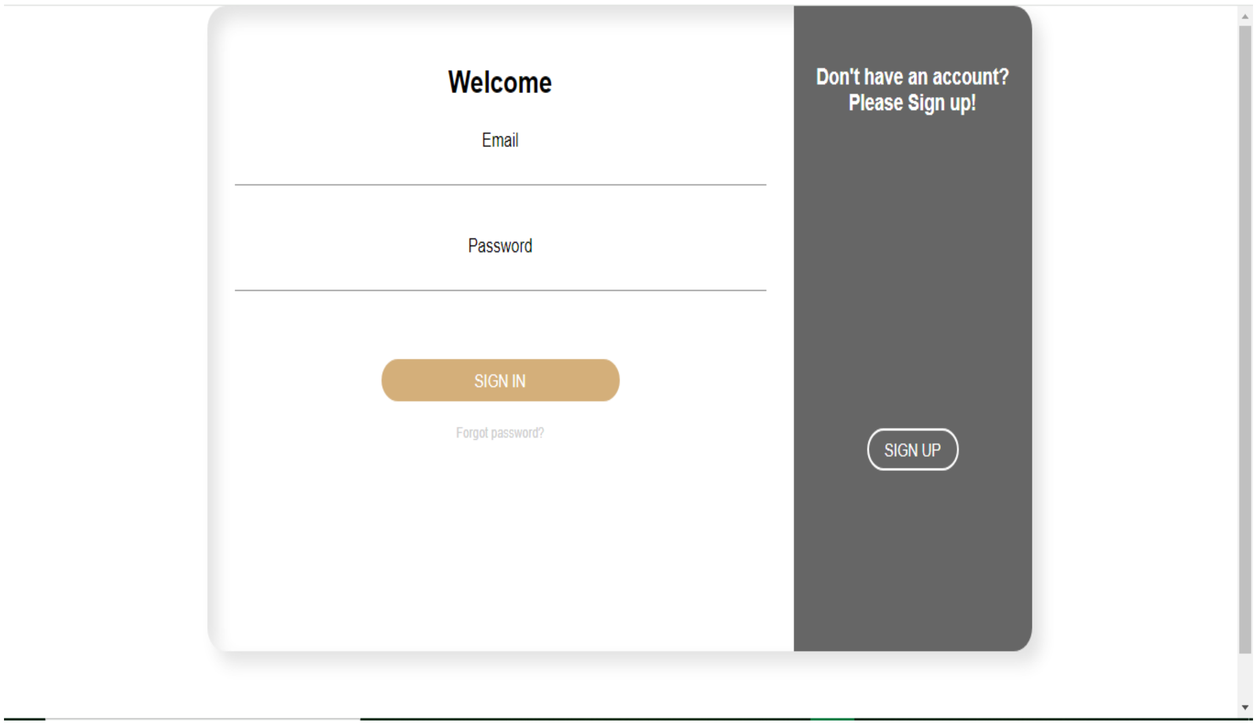
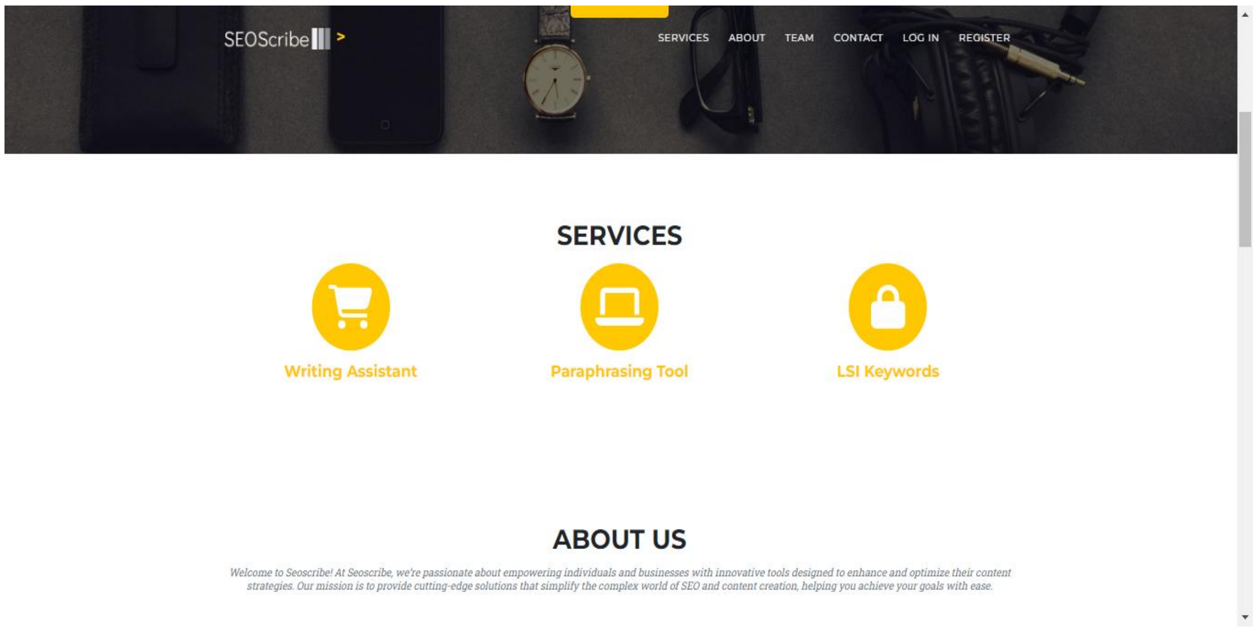


Figure 4.3 Login Interface

### *4.2.4 Services*

On the home page user can navigate to the **Services** in order to use the apps features. The app provides 3 features as of now, **Writing Assistant, Paraphraser** and **Keyword Generation Tool**. In order to the use a service user must click on the desired service and will be redirected to the corresponding page.



Welcome to Seoscribe! At Seoscribe, we're passionate about empowering individuals and businesses with innovative tools designed to enhance and optimize their content strategies. Our mission is to provide cutting-edge solutions that simplify the complex world of SEO and content creation, helping you achieve your goals with ease.

Figure 4.4 Services Interface

### *4.2.5 Paraphraser*

In order to use the paraphraser feature user must navigate to the paraphraser page by clicking the **Paraphraser** button under **Services**. Upon clicking, user is redirected to the paraphraser page and a form is displayed to the user. User can input a paragraph or a sentence in the input field. Upon clicking the **Paraphrase** button system paraphrase the text and the result is displayed.

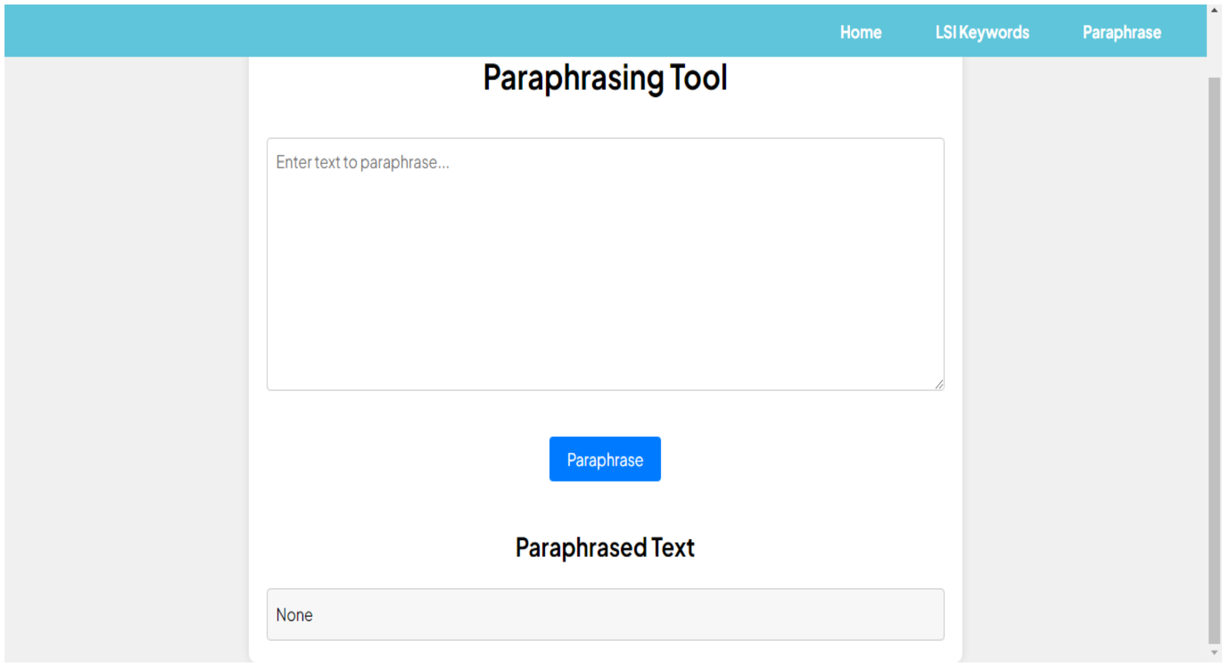
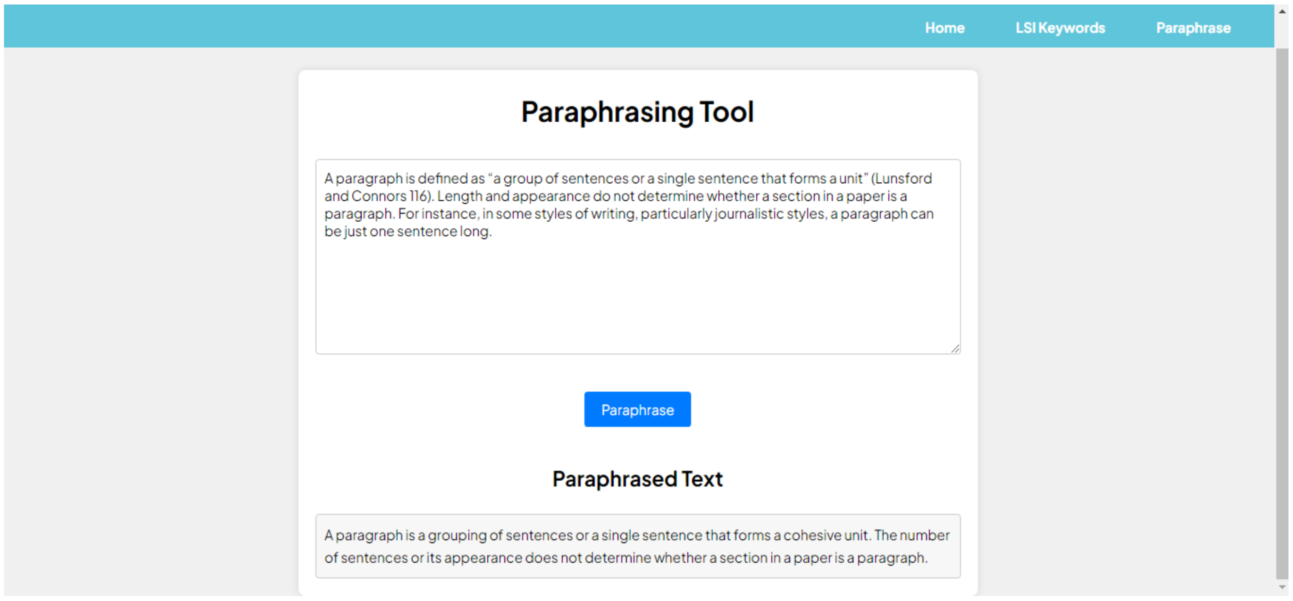


Figure 4.5 Paraphraser UI (Blank)



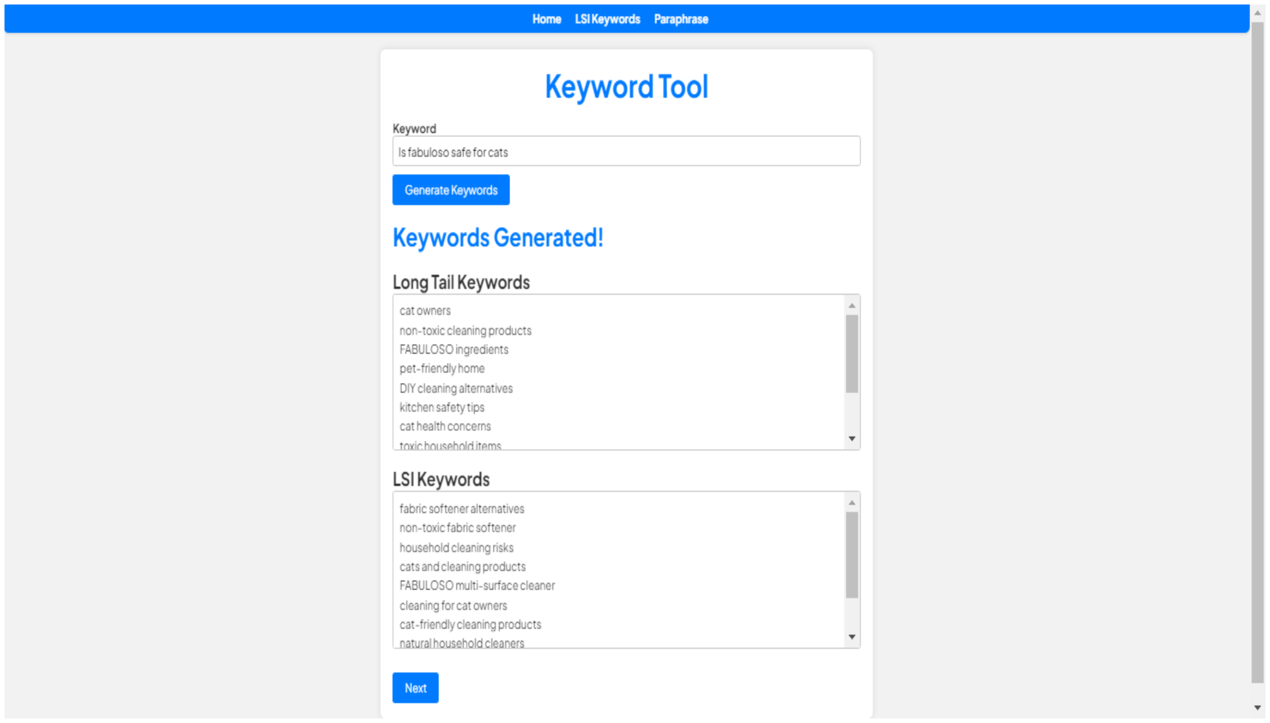
A paragraph is defined as "a group of sentences or a single sentence that forms a unit" (Lunsford and Connors 116). Length and appearance do not determine whether a section in a paper is a paragraph. For instance, in some styles of writing, particularly journalistic styles, a paragraph can be just one sentence long.

A paragraph is a grouping of sentences or a single sentence that forms a cohesive unit. The number of sentences or its appearance does not determine whether a section in a paper is a paragraph.

Figure 4.6 Paraphraser UI (With Data)

### *4.2.6 Keyword Generation*

In order to use the **keyword Tool**, user must navigate to the desired page by clicking the **keyword Tool** button under the **Services**. Upon clicking user is redirected to the desired page. User can input the keyword in the input field. Upon clicking the **Generate** button, system will generate keywords and the results would be displayed.



cat owners
non-toxic cleaning products
FABULOSO ingredients
pet-friendly home
DIY cleaning alternatives
kitchen safety tips
cat health concerns
toxic household items

fabric softener alternatives
non-toxic fabric softener
household cleaning risks
cats and cleaning products
FABULOSO multi-surface cleaner
cleaning for cat owners
cat-friendly cleaning products
natural household cleaners

Figure 4.7 Keyword Tool UI

### *4.2.7 Writing Assistant*

In order to use the writing Assistant User must Click the **Writing Assistant** button under the **Services**. Upon Clicking user is redirected to the **Project creation window**.

#### *4.2.7.1 Project Creation Window*

In the project creation window user is displayed previous projects (if have any) and user have the option to **Edit** or **Delete** any project Entry. Moreover, user is displayed a button **Create New Project**.

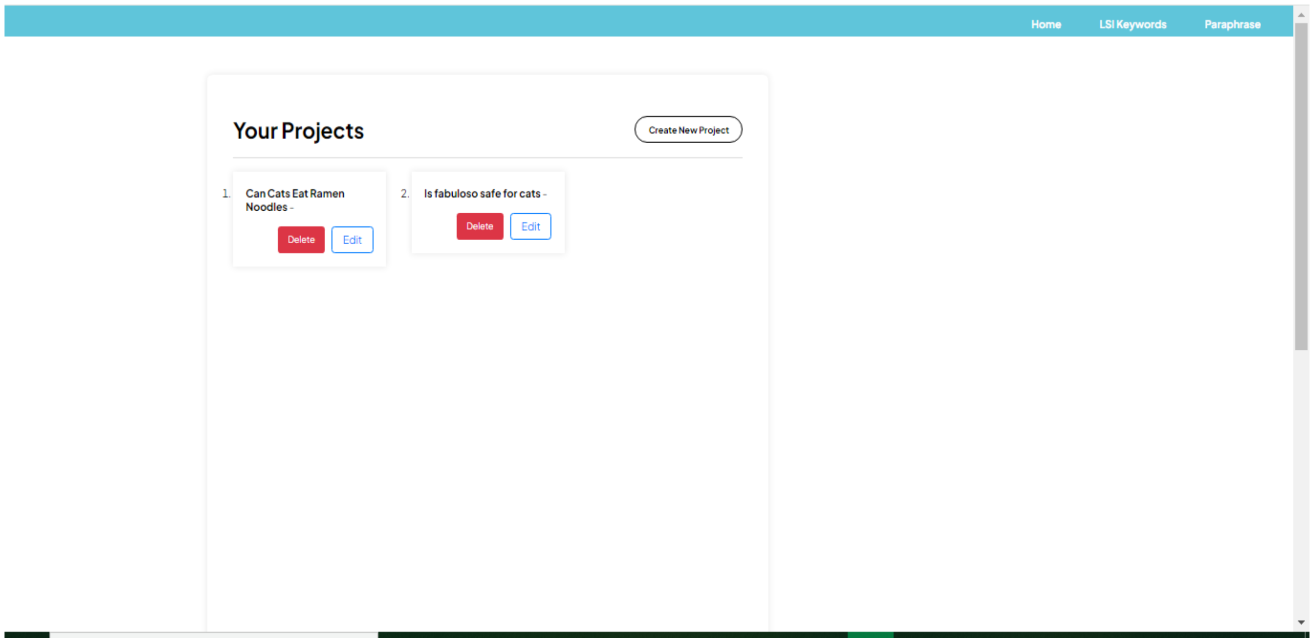


Figure 4.8 Project Creation Window

#### *4.2.7.3 Create New Project Prompt*

Upon Clicking **Create New Project** button, user is displayed a prompt to enter the project title and provide a description (Optional). Upon Clicking **save,** system creates a Project entry and proceeds to keyword generation page.

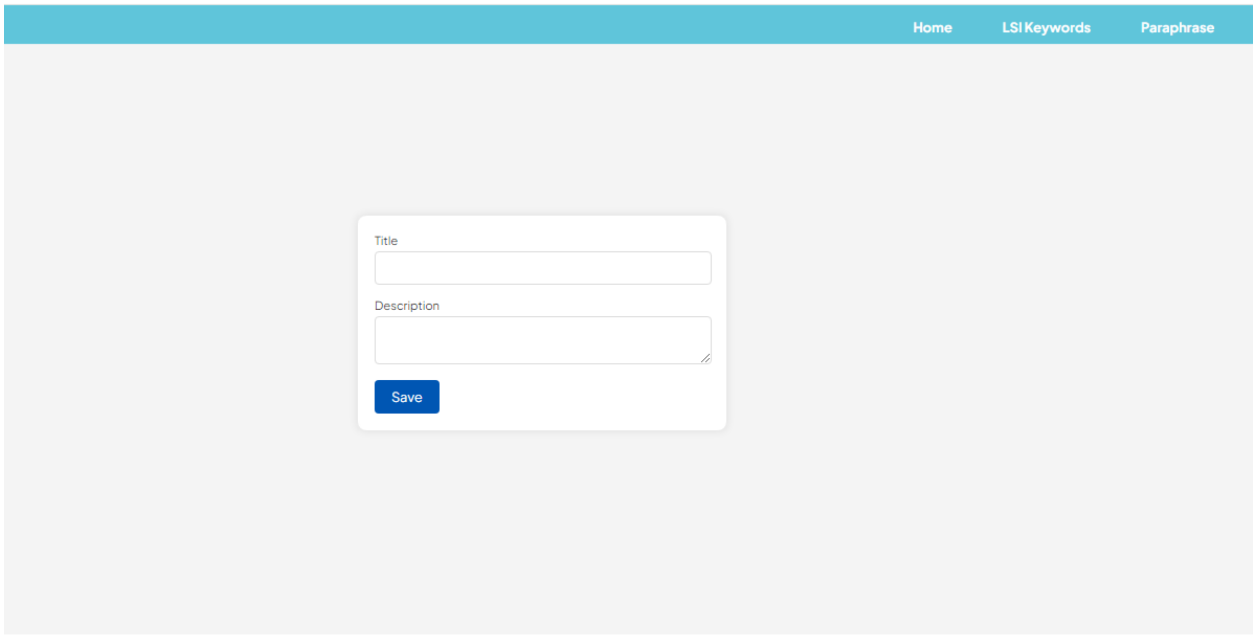
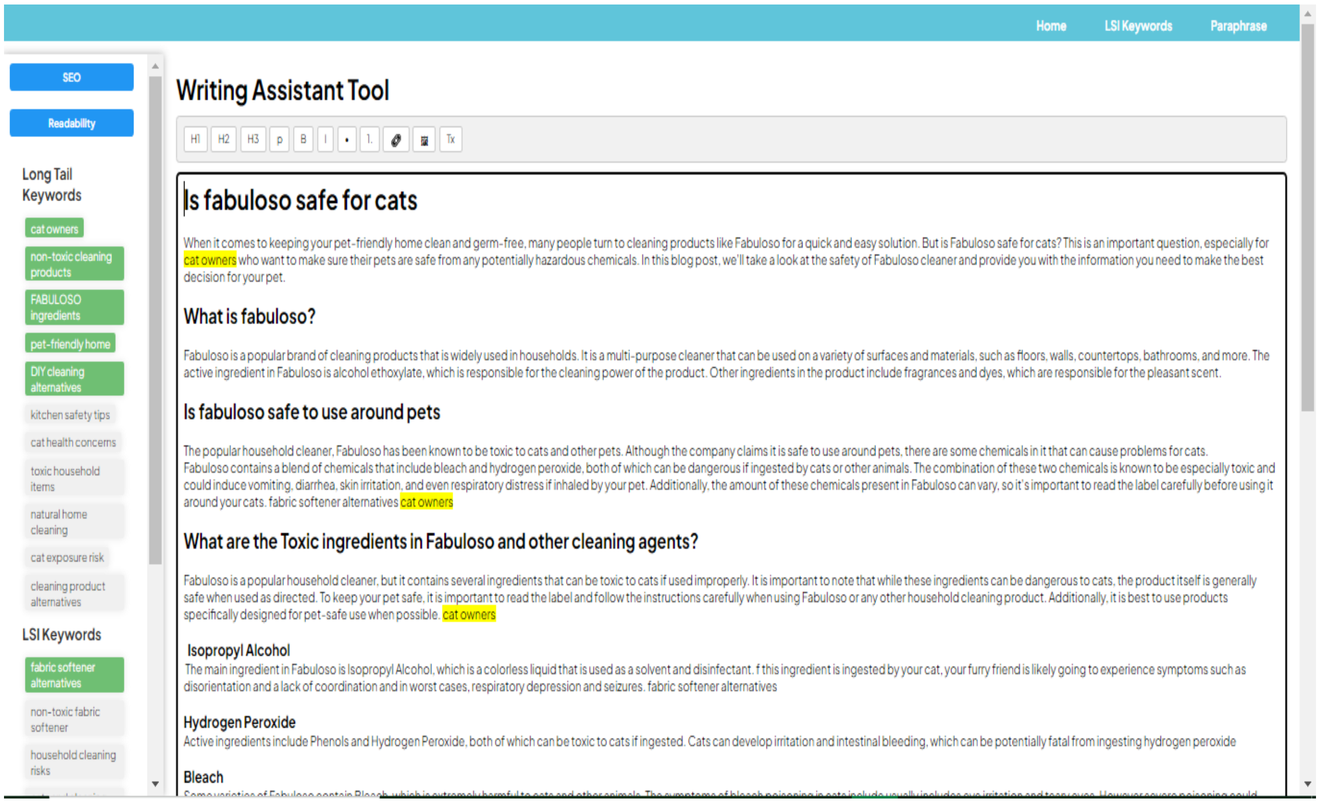


Figure 4.9 New Project Prompt

### 4.2.7.4 *Text Editor*

After Completing the keywords generation and clicking the **Next** button. User is redirected to the **Text Editor** page. Moreover, user is displayed a sidebar with the generated keywords displayed in blocks under their respective section (Long tail and LSI). User can input text int the Text Editor. Editing Operations can be selected from the **Top Bar** of the Editor. User can switch between SEO and Readability Analysis through the toggle button in the sidebar. Upon Clicking the **Get Suggestions** button System will display SEO and Readability suggestions.



Writing Assistant Tool

Is fabuloso safe for cats

When it comes to keeping your pet-friendly home clean and germ-free, many people turn to cleaning products like Fabuloso for a quick and easy solution. But is Fabuloso safe for cats? This is an important question, especially for cat owners who want to make sure their pets are safe from any potentially hazardous chemicals. In this blog post, we'll take a look at the safety of Fabuloso cleaner and provide you with the information you need to make the best decision for your pet.

What is fabuloso?

Fabuloso is a popular brand of cleaning products that is widely used in households. It is a multi-purpose cleaner that can be used on a variety of surfaces and materials, such as floors, walls, countertops, bathrooms, and more. The active ingredient in Fabuloso is alcohol ethoxylate, which is responsible for the cleaning power of the product. Other ingredients in the product include fragrances and dyes, which are responsible for the pleasant scent.

Is fabuloso safe to use around pets

The popular household cleaner, Fabuloso has been known to be toxic to cats and other pets. Although the company claims it is safe to use around pets, there are some chemicals in it that can cause problems for cats. Fabuloso contains a blend of chemicals that include bleach and hydrogen peroxide, both of which can be dangerous if ingested by cats or other animals. The combination of these two chemicals is known to be especially toxic and could induce vomiting, diarrhea, skin irritation, and even respiratory distress if inhaled by your pet. Additionally, the amount of these chemicals present in Fabuloso can vary, so it's important to read the label carefully before using it around your cats. fabric softener alternatives cat owners

What are the Toxic ingredients in Fabuloso and other cleaning agents?

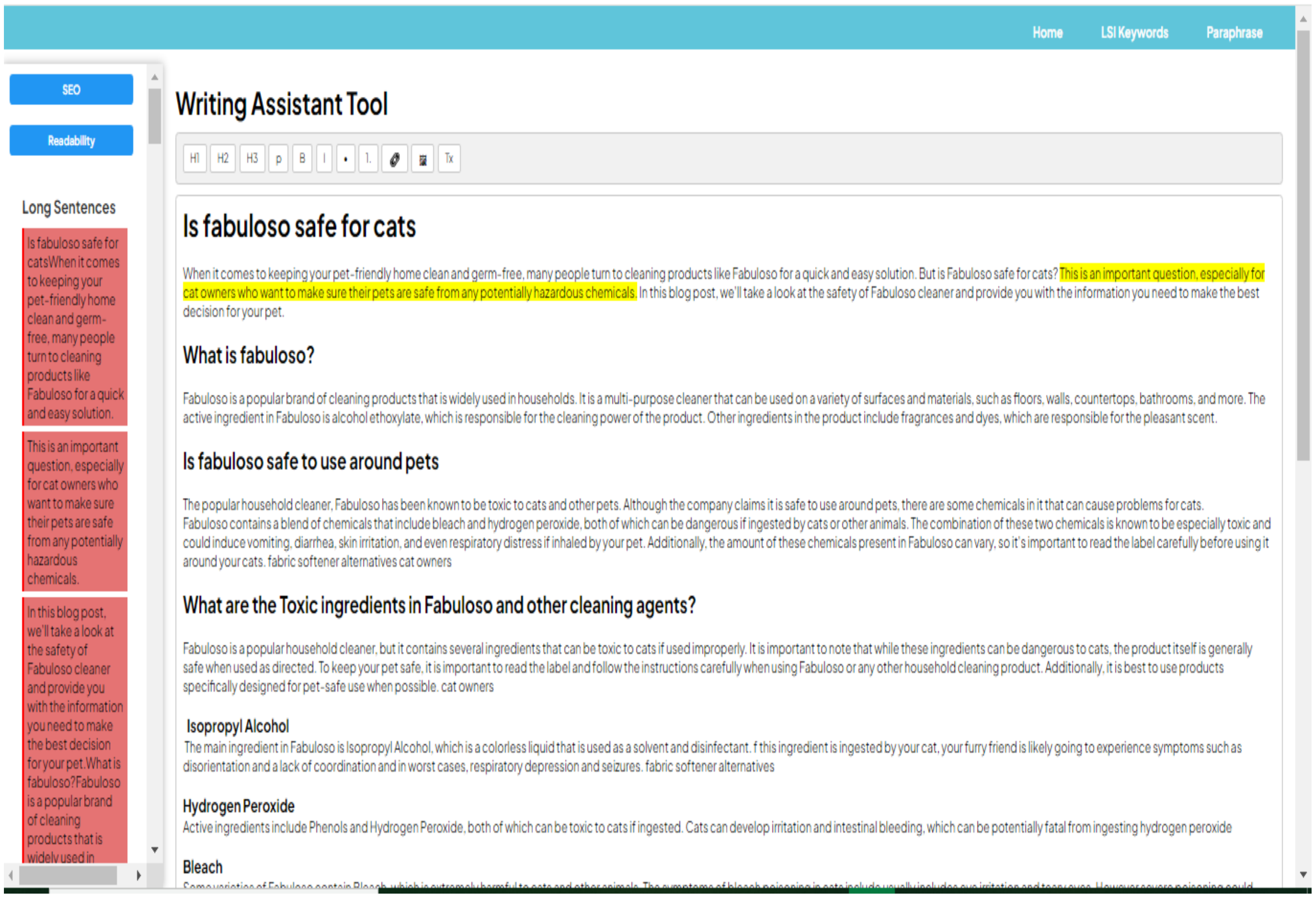
Fabuloso is a popular household cleaner, but it contains several ingredients that can be toxic to cats if used improperly. It is important to note that while these ingredients can be dangerous to cats, the product itself is generally safe when used as directed. To keep your pet safe, it is important to read the label and follow the instructions carefully when using Fabuloso or any other household cleaning product. Additionally, it is best to use products specifically designed for pet-safe use when possible. cat owners

Isopropyl Alcohol
The main ingredient in Fabuloso is Isopropyl Alcohol, which is a colorless liquid that is used as a solvent and disinfectant. f this ingredient is ingested by your cat, your furry friend is likely going to experience symptoms such as disorientation and a lack of coordination and in worst cases, respiratory depression and seizures. fabric softener alternatives

Hydrogen Peroxide
Active ingredients include Phenols and Hydrogen Peroxide, both of which can be toxic to cats if ingested. Cats can develop irritation and intestinal bleeding, which can be potentially fatal from ingesting hydrogen peroxide

Bleach
Some varieties of Fabuloso contain Bleach, which is extremely harmful to cats and other animals. The symptoms of bleach poisoning in cats include usually includes eye irritation and teary eyes. However severe poisoning could

Figure 4.10 Editor UI -1



Long Sentences

Is fabuloso safe for catsWhen it comes to keeping your pet-friendly home clean and germ-free, many people turn to cleaning products like Fabuloso for a quick and easy solution.

This is an important question, especially for cat owners who want to make sure their pets are safe from any potentially hazardous chemicals.

In this blog post, we'll take a look at the safety of Fabuloso cleaner and provide you with the information you need to make the best decision for your pet.What is fabuloso?Fabuloso is a popular brand of cleaning products that is widely used in

Writing Assistant Tool

Is fabuloso safe for cats

When it comes to keeping your pet-friendly home clean and germ-free, many people turn to cleaning products like Fabuloso for a quick and easy solution. But is Fabuloso safe for cats? This is an important question, especially for cat owners who want to make sure their pets are safe from any potentially hazardous chemicals. In this blog post, we'll take a look at the safety of Fabuloso cleaner and provide you with the information you need to make the best decision for your pet.

What is fabuloso?

Fabuloso is a popular brand of cleaning products that is widely used in households. It is a multi-purpose cleaner that can be used on a variety of surfaces and materials, such as floors, walls, countertops, bathrooms, and more. The active ingredient in Fabuloso is alcohol ethoxylate, which is responsible for the cleaning power of the product. Other ingredients in the product include fragrances and dyes, which are responsible for the pleasant scent.

Is fabuloso safe to use around pets

The popular household cleaner, Fabuloso has been known to be toxic to cats and other pets. Although the company claims it is safe to use around pets, there are some chemicals in it that can cause problems for cats. Fabuloso contains a blend of chemicals that include bleach and hydrogen peroxide, both of which can be dangerous if ingested by cats or other animals. The combination of these two chemicals is known to be especially toxic and could induce vomiting, diarrhea, skin irritation, and even respiratory distress if inhaled by your pet. Additionally, the amount of these chemicals present in Fabuloso can vary, so it's important to read the label carefully before using it around your cats. fabric softener alternatives cat owners

What are the Toxic ingredients in Fabuloso and other cleaning agents?

Fabuloso is a popular household cleaner, but it contains several ingredients that can be toxic to cats if used improperly. It is important to note that while these ingredients can be dangerous to cats, the product itself is generally safe when used as directed. To keep your pet safe, it is important to read the label and follow the instructions carefully when using Fabuloso or any other household cleaning product. Additionally, it is best to use products specifically designed for pet-safe use when possible. cat owners

Isopropyl Alcohol

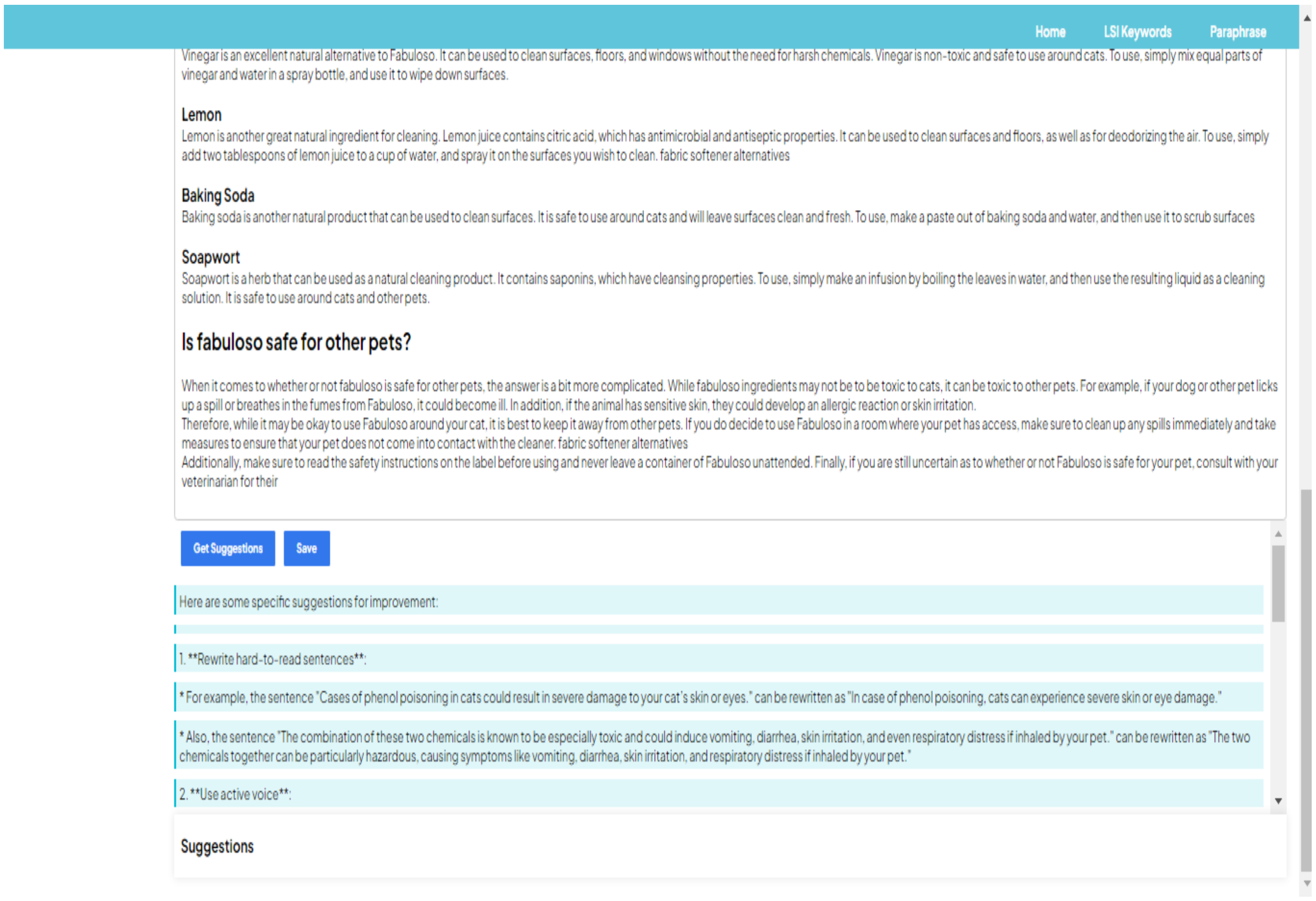
The main ingredient in Fabuloso is Isopropyl Alcohol, which is a colorless liquid that is used as a solvent and disinfectant. f this ingredient is ingested by your cat, your furry friend is likely going to experience symptoms such as disorientation and a lack of coordination and in worst cases, respiratory depression and seizures. fabric softener alternatives

Hydrogen Peroxide

Active ingredients include Phenols and Hydrogen Peroxide, both of which can be toxic to cats if ingested. Cats can develop irritation and intestinal bleeding, which can be potentially fatal from ingesting hydrogen peroxide

Bleach

Figure 4.11 Editor UI -2



Vinegar is an excellent natural alternative to Fabuloso. It can be used to clean surfaces, floors, and windows without the need for harsh chemicals. Vinegar is non-toxic and safe to use around cats. To use, simply mix equal parts of vinegar and water in a spray bottle, and use it to wipe down surfaces.

**Lemon**

Lemon is another great natural ingredient for cleaning. Lemon juice contains citric acid, which has antimicrobial and antiseptic properties. It can be used to clean surfaces and floors, as well as for deodorizing the air. To use, simply add two tablespoons of lemon juice to a cup of water, and spray it on the surfaces you wish to clean. fabric softener alternatives

**Baking Soda**

Baking soda is another natural product that can be used to clean surfaces. It is safe to use around cats and will leave surfaces clean and fresh. To use, make a paste out of baking soda and water, and then use it to scrub surfaces

**Soapwort**

Soapwort is a herb that can be used as a natural cleaning product. It contains saponins, which have cleansing properties. To use, simply make an infusion by boiling the leaves in water, and then use the resulting liquid as a cleaning solution. It is safe to use around cats and other pets.

**Is fabuloso safe for other pets?**

When it comes to whether or not fabuloso is safe for other pets, the answer is a bit more complicated. While fabuloso ingredients may not be to be toxic to cats, it can be toxic to other pets. For example, if your dog or other pet licks up a spill or breathes in the fumes from Fabuloso, it could become ill. In addition, if the animal has sensitive skin, they could develop an allergic reaction or skin irritation.
Therefore, while it may be okay to use Fabuloso around your cat, it is best to keep it away from other pets. If you do decide to use Fabuloso in a room where your pet has access, make sure to clean up any spills immediately and take measures to ensure that your pet does not come into contact with the cleaner. fabric softener alternatives
Additionally, make sure to read the safety instructions on the label before using and never leave a container of Fabuloso unattended. Finally, if you are still uncertain as to whether or not Fabuloso is safe for your pet, consult with your veterinarian for their

Here are some specific suggestions for improvement:

1. **Rewrite hard-to-read sentences**:

* For example, the sentence "Cases of phenol poisoning in cats could result in severe damage to your cat's skin or eyes." can be rewritten as "In case of phenol poisoning, cats can experience severe skin or eye damage."

* Also, the sentence "The combination of these two chemicals is known to be especially toxic and could induce vomiting, diarrhea, skin irritation, and even respiratory distress if inhaled by your pet." can be rewritten as "The two chemicals together can be particularly hazardous, causing symptoms like vomiting, diarrhea, skin irritation, and respiratory distress if inhaled by your pet."

2. **Use active voice**:

**Suggestions**

Figure 4.12 Editor UI -3

## 4.3 Testing

### *4.3.1 Test Plan*

#### *4.3.1.1 Introduction*

This test plan outlines the test levels and testing techniques for the Seoscribe project, focusing on ensuring the quality and functionality of the application. Testing will be conducted using manual test cases and pytest within the VSCode environment.

#### *4.3.1.2 Test Levels*

##### 4.3.1.2.1 Unit Testing

**Objective:** Verify the functionality of individual units/components of the application.

**Scope:**

- Testing individual functions, methods, and classes within the project.
- Ensuring that each unit performs as expected in isolation.

**Techniques:**

- **Type:** White Box Testing
- **Method:** Manual Test Cases, Automated Testing using Pytest
- **Tools**: Pytest, VSCode

##### 4.3.1.2.2 Integration Testing

**Objective:** Ensure that the integrated units/components work together as intended.

**Scope:**

- Testing the interactions between different modules and components.
- Verifying data flow and communication between integrated units.

**Techniques:**

- **Type:** Gray Box Testing

- **Method:** Manual Test Cases, Automated Testing using Pytest
- **Tools:** Pytest, VSCode

##### 4.3.1.2.3 System Testing

**Objective:** Validate the entire system's compliance with the specified requirements.

**Scope:**

- Testing the complete and integrated application to evaluate the end-to-end system specifications.
- Ensuring that the system functions as a whole according to the requirements.

**Techniques:**

- **Type:** Black Box Testing
- **Method:** Manual Test Cases, Automated Testing using Pytest
- **Tools:** Pytest, VSCode

#### *4.3.1.3 Testing Techniques*

##### 4.3.1.3.1 Manual Testing

**Purpose:** To provide a human perspective on the application's functionality and usability.

**Approach:**

- Develop detailed test cases for each test level.
- Execute test cases manually to verify the expected outcomes.
- Document and report any defects or issues found during testing.

##### 4.3.1.3.2 Automated Testing with Pytest

**Purpose:** To ensure consistent, repeatable, and efficient testing across the application.

**Approach:**

- Write automated test scripts using pytest for unit, integration, and system testing.
- Execute automated tests regularly to catch regressions and ensure stability.

**Tools:** Pytest, VSCode

### *4.3.2 Test Cases*

#### *4.3.2.1 Login*

| **TC ID** | LogIn_01 |
|---|---|
| **Test Case Name** | Log In to SeoScribe App |
| **Objective** | To verify that a registered user can log in to the SeoScribe app successfully. |
| **Precondition** | User must have an active internet connection.<br>User must have a registered account in the app. |
| **Actor** | User |

| **Test Step** | **Action** | **Test Data** | **Expected Result** | **Test Result** |
|---|---|---|---|---|
| 1 | **Open App**: Launch the SeoScribe app | None | SeoScribe app is launched | |
| 2 | **Navigate to Login**: Click on the login button on the home screen. | None | User is redirected to the login page | |
| 3 | Enter Test data | Email: user@example.com<br>Password: validpassword12 | Email and password are entered | |
| 4 | **Submit**: Click on the 'Log In' button | None | System checks the credentials against the database | |
| 5 | **Validate**: Verify that the system checks the credentials against the database | None | Credentials are validated against the database | |
| 6 | **Check Redirect**: Ensure that the user is redirected to the SeoScribe dashboard upon successful login. | None | User should be successfully logged in and redirected to the SeoScribe dashboard | |

Table 4.1 TC01

#### 4.3.2.1.1 Login 02

| | |
|---|---|
| **TC ID** | LogIn_02 |
| **Test Case Name** | Log In with Invalid Email |
| **Objective** | To verify that the system handles invalid email input correctly. |
| **Precondition** | User must have an active internet connection.<br>User must have a registered account in the app. |
| **Actor** | User |

| **Test Step** | **Action** | **Test Data** | **Expected Result** | **Test Result** |
|---|---|---|---|---|
| 1 | **Open App**: Launch the SeoScribe app | None | SeoScribe app is launched | |
| 2 | **Navigate to Login**: Click on the login button on the home screen. | None | User is redirected to the login page | |
| 3 | Enter Test data | Email: invalid@example.com<br>Password: validpassword12 | Email and password are entered | |
| 4 | **Submit**: Click on the 'Log In' button | None | System checks the credentials against the database | |
| 5 | **Validate**: Verify that the system checks the credentials against the database | None | Credentials are validated against the database | |
| 6 | **Check Error:** Ensure that the system displays an error message: ‘invalid email’. | None | System should display an error message indicating an invalid email | |

Table 4.2 TC01.2

#### 4.3.2.1.2 Login03

| | |
|---|---|
| **TC ID** | LogIn_03 |
| **Test Case Name** | Log In with Invalid Password |
| **Objective** | To verify that the system handles invalid password input correctly |
| **Precondition** | • User must have an active internet connection.<br>• User must have a registered account in the app. |
| **Actor** | User |

| **Test Step** | **Action** | **Test Data** | **Expected Result** | **Test Result** |
|---|---|---|---|---|
| 1 | **Open App**: Launch the SeoScribe app | None | SeoScribe app is launched | |
| 2 | **Navigate to Login**: Click on the login button on the home screen. | None | User is redirected to the login page | |
| 3 | Enter Test data | Email: test@example.com<br>Password: invalidpassword123 | Email and password are entered | |
| 4 | **Submit**: Click on the 'Log In' button | None | System checks the credentials against the database | |
| 5 | **Validate**: Verify that the system checks the credentials against the database | None | Credentials are validated against the database | |
| 6 | **Check Error:** Ensure that the system displays an error message: ‘invalid password’ | None | System should display an error message indicating an invalid password | |

Table 4.3 TC01.3

### *4.3.2.2 Sing up*

| | |
|---|---|
| **TC ID** | SignUp_01 |
| **Test Case Name** | Sign Up to SeoScribe App |
| **Objective** | To verify that a new user can create an account in the SeoScribe app successfully. |
| **Precondition** | User must be connected to the internet |
| **Actor** | User |

| Test Step | Action | Test Data | Expected Result | Test Result |
|---|---|---|---|---|
| 1 | **Open App**: Launch the SeoScribe app | None | SeoScribe app is launched | |
| 2 | **Navigate to Sign Up:** Click on the sign-up button on the home screen. | None | User is redirected to the sign-up page | |
| 3 | Enter Test data | Email: user@example.com Password: validpassword123 Username: validusername | Email, password, and username are entered | |
| 4 | **Agree to Terms:** Check the agreement box | None | Agreement box is checked | |
| 5 | **Submit:** Click on the 'Sign Up' button | None | System creates an entry in the database | |
| 6 | **Check Redirect:** Ensure that the user is redirected to the SeoScribe dashboard upon successful sign up | None | User should be successfully signed up and redirected to the SeoScribe dashboard | |

Table 4.4 TC02

#### 4.3.2.2.1 Sign Up 02

| | |
|---|---|
| **TC ID** | SignUp_02 |
| **Test Case Name** | Sign Up with Invalid Email |
| **Objective** | To verify that the system handles invalid email input during the sign-up process. |
| **Precondition** | User must be connected to the internet |
| **Actor** | User |

| Test Step | Action | Test Data | Expected Result | Test Result |
|---|---|---|---|---|
| 1 | **Open App:** Launch the SeoScribe app | None | SeoScribe app is launched | |
| 2 | **Navigate to Sign Up:** Click on the sign up button on the home screen. | None | User is redirected to the sign up page | |
| 3 | Enter Test data | Email: invalid@gmail.com<br>Password:kpwd<br>Username: validusername | Email, password, and username are entered | |
| 4 | **Submit:** Click on the 'Sign Up' button | None | System displays an error message "invalid email" | |
| 5 | **Check Error:** Ensure that the user is prompted to enter a stronger password | None | System displays an error message "invalid email" and does not proceed with account creation | |

Table 4.5 TC02.1

#### *4.3.2.3 Create a New Project*

| | |
|---|---|
| **TC ID** | CreateProject_01 |
| **Test Case Name** | Create a New Project in SeoScribe App |
| **Objective** | To verify that a logged-in user can create a new project successfully in the SeoScribe app. |
| **Precondition** | User must be logged in. |
| **Actor** | User |

| **Test Step** | **Action** | **Test Data** | **Expected Result** | **Test Result** |
|---|---|---|---|---|
| 1 | **Navigate to Writing Assistant:** Click on the writing assistant on the homepage | None | User is redirected to the writing assistant dashboard | |
| 2 | **Click Create Project:** Click on the 'Create Project' button | None | System prompts the user to name the project | |
| 3 | **Enter Project Name**: User enters a valid project name | Project Name: MyFirstProject | Project name is entered successfully | |
| 4 | **Enter Project Description** (Optional): User enters a description for the project | Description: My first project description | Description is entered successfully (optional) | |
| 5 | **Click Next:** Click on the 'Next' button | None | System creates an entry in the database | |
| 6 | **Check Redirect:** Ensure that the user is redirected to the keyword generation page upon successful project creation | None | User should be successfully redirected to the keyword generation page | |

Table 4.6 TC03

#### 4.3.2.4 Open an Existing Project

| | |
|---|---|
| TC ID | OpenProject_01 |
| **Test Case Name** | Open an Existing Project in SeoScribe App |
| **Objective** | To verify that a logged-in user can open an existing project successfully in the SeoScribe app |
| **Precondition** | User must be logged in.<br>User must have created a project previously. |
| **Actor** | User |

| Test Step | Action | Test Data | Expected Result | Test Result |
|---|---|---|---|---|
| 1 | **Navigate to Writing Assistant:** Click on the writing assistant on the homepage | None | User is redirected to the writing assistant dashboard | |
| 2 | **Display Projects:** System displays previous projects | None | User sees a list of previously created projects | |
| 3 | **Select Project:** User clicks on an existing project | Project Name: MyFirstProject | System loads the project in the text editor | |
| 4 | **Validate:** Verify that the project loads correctly in the text editor | None | Project content is displayed in the text editor | |

Table 4.7 TC04

#### *4.3.2.5 Keyword Generation*

| | |
|---|---|
| **TC ID** | KeywordGeneration_01 |
| **Test Case Name** | Generate Keywords in SeoScribe App |
| **Objective** | To verify that a logged-in user can generate keywords successfully in the SeoScribe app. |
| **Precondition** | User must be logged in.<br>User must have an internet connection. |
| **Actor** | User |

| **Test Step** | **Action** | **Test Data** | **Expected Result** | **Test Result** |
|---|---|---|---|---|
| 1 | **Navigate to Keyword Tool:** Click on the keyword tool on the homepage | None | User is redirected to the keyword generation page | |
| 2 | **Enter Focus Keyword:** User is prompted to enter the focus keyword | Keyword: SEO Tools | Focus keyword is entered successfully | |
| 3 | **Click Generate:** Click on the 'Generate' button | None | System generates long tail keywords and LSI keywords | |
| 4 | **Validate:** Verify that the generated keywords are displayed | None | Long tail and LSI keywords are displayed | |

Table 4.8 TC05

#### 4.3.2.6 *Text Editor*

| | |
|---|---|
| **TC ID** | TextEditor_01 |
| **Test Case Name** | Open Text Editor in SeoScribe App |
| **Objective** | To verify that a logged-in user can access the text editor after creating a project. |
| **Precondition** | User must have an internet connection.<br>User must have created a project. |
| **Actor** | User |

| Test Step | Action | Test Data | Expected Result | Test Result |
|---|---|---|---|---|
| 1 | **Complete Project Creation:** On the project creation window, user clicks 'Next' | None | System redirects the user to the keyword generation page | |
| 2 | **Follow Keyword Generation**: User follows the steps in UC05 to generate keywords | None | Keywords are generated and displayed | |
| 3 | **Navigate to Text Editor:** After generating keywords, user clicks 'Next' | None | System redirects the user to the text editor | |
| 4 | **Validate**: Verify that the text editor opens with the project details and keywords in the sidebar | None | Text editor opens with the project details and keywords | |

Table 4.9 TC06

#### *4.3.2.7 Input Text in Text Editor*

| | |
|---|---|
| **TC ID** | InputText_01 |
| **Test Case Name** | Input Text in Text Editor in SeoScribe App |
| **Objective** | To verify that a logged-in user can input and format text in the text editor. |
| **Precondition** | User must have completed steps in UC03 and UC05.<br>User is on the text editor page. |
| **Actor** | User |

| **Test Step** | **Action** | **Test Data** | **Expected Result** | **Test Result** |
|---|---|---|---|---|
| 1 | **Input Text:** User writes into the text editor | Text: This is a sample text. | System accepts the text input | |
| 2 | **Validate Input:** Verify that the system identifies text as either body or heading | None | Text is identified and formatted accordingly | |
| 3 | **Format Text:** System formats the text accordingly | None | Text is displayed and formatted as either body or heading | |

Table 4.10 TC07

#### *4.3.2.8 Edit Text*

| | |
|---|---|
| **TC ID** | EditText_01 |
| **Test Case Name** | Edit Text in Text Editor in SeoScribe App |
| **Objective** | To verify that a logged-in user can perform text editing operations in the text editor. |
| **Precondition** | User has text inside the text editor.<br>User is on the text editor page. |
| **Actor** | User |

| **Test Step** | **Action** | **Test Data** | **Expected Result** | **Test Result** |
|---|---|---|---|---|
| 1 | **Select Text:** User selects text | Text: sample text | Text is selected successfully | |
| 2 | **Apply Editing Operation:** User selects an editing operation (e.g., bold, italic) | Operation: Bold | The selected operation is applied to the selected text | |
| 3 | **Validate Editing:** Verify that the system displays the results of the applied operation | None | Text is displayed with the applied formatting | |

Table 4.11 TC08

#### *4.3.2.9 SEO Analysis*

| | |
|---|---|
| **TC ID** | SEOAnalysis_01 |
| **Test Case Name** | Perform SEO Analysis in SeoScribe App |
| **Objective** | To verify that a logged-in user can perform SEO analysis of the text in the text editor. |
| **Precondition** | Steps in UC03, UC04, and UC05 must be completed. |
| **Actor** | User |

| Test Step | Action | Test Data | Expected Result | Test Result |
|---|---|---|---|---|
| 1 | **Input Text:** User inputs text into the editor | Text: This is a sample text for SEO analysis. | System accepts the text input | |
| 2 | **Scan Headings:** System scans the headings for long tail keywords | None | System highlights the matching keywords as green in the sidebar | |
| 3 | **Scan Paragraphs:** System scans the paragraphs for LSI keywords | None | System highlights matching keywords as green in the sidebar | |
| 4 | **Validate:** Verify that keywords found in text are highlighted as green | None | Keywords are highlighted as green to indicate presence in text | |

Table 4.12 TC09

### *4.3.2.10 Readability Analysis*

| | |
|---|---|
| **TC ID** | ReadabilityAnalysis_01 |
| **Test Case Name** | Perform Readability Analysis in SeoScribe App |
| **Objective** | To verify that a logged-in user can perform readability analysis of the text in the text editor. |
| **Precondition** | Steps in UC03, UC04, and UC05 must be completed.<br>User has text in the editor. |
| **Actor** | User |

| Test Step | Action | Test Data | Expected Result | Test Result |
|---|---|---|---|---|
| 1 | **Input Text:** User pastes or input text into the text editor | Text: This is a complex sentence that needs readability analysis. | System accepts the text input | |
| 2 | **Scan Text**: System identifies long sentences and difficult vocabulary | None | System highlights the identified issues as light red in the sidebar under readability section | |
| 3 | **Validate:** Verify that identified issues are highlighted as light red | None | Issues are highlighted as light red in the sidebar | |
| 4 | **Resolve Issues:** makes changes to the corresponding sentence or vocabulary | None | System scans the text and highlights resolved issues as green in sidebar | |
| 5 | **Validate Resolved Issues:** Verify that resolved issues are highlighted as green | None | Resolved issues are highlighted as green in the sidebar | |

Table 4.13 TC10

### *4.3.2.11 Paraphrasing*

| | |
|---|---|
| **TC ID** | Paraphrasing_01 |
| **Test Case Name** | Paraphrase Text in SeoScribe App |
| **Objective** | To verify that a logged-in user can paraphrase text in the SeoScribe app. |
| **Precondition** | User must have an internet connection. |
| **Actor** | User |

| **Test Step** | **Action** | **Test Data** | **Expected Result** | **Test Result** |
|---|---|---|---|---|
| 1 | **Navigate to Paraphraser:** User clicks paraphraser on homepage | None | User is redirected to the paraphraser page | |
| 2 | **Input Text:** User inputs some text | Text: This is a sample text for paraphrasing. | Text is entered successfully | |
| 3 | **Click Paraphrase:** Click on the 'Paraphrase' button | None | System paraphrases the text and displays the result | |
| 4 | **Validate:** Verify that the paraphrased text is displayed | None | Paraphrased text is displayed | |

Table 4.14 TC11

### *4.3.2.12 Image Insertion*

| | |
|---|---|
| **TC ID** | ImageInsertion_01 |
| **Test Case Name** | Insert Image in Text Editor in SeoScribe App |
| **Objective** | To verify that a logged-in user can insert an image at the cursor position in the text editor. |
| **Precondition** | User is on the text editor page. |
| **Actor** | User |

| **Test Step** | **Action** | **Test Data** | **Expected Result** | **Test Result** |
|---|---|---|---|---|
| 1 | **Select Insert Image:** User selects the insert image icon from the toolbar | None | System opens a prompt providing an option to browse for an image | |
| 2 | **Browse Image:** User selects an image and clicks insert | Image: sample.jpg | System prompts user to a window asking to modify image size | |
| 3 | **Input Image Details:** User inputs the image size and alt text | Size: 500x500, Alt Text: Sample Image | System provides option on the same window to insert alt text for the image | |
| 4 | **Insert Image:** User confirms and inserts the image | None | System inserts the image at the cursor position | |
| 5 | **Validate:** Verify that the image is inserted at the cursor position | None | Image is inserted at the cursor position | |

Table 4.15 TC12

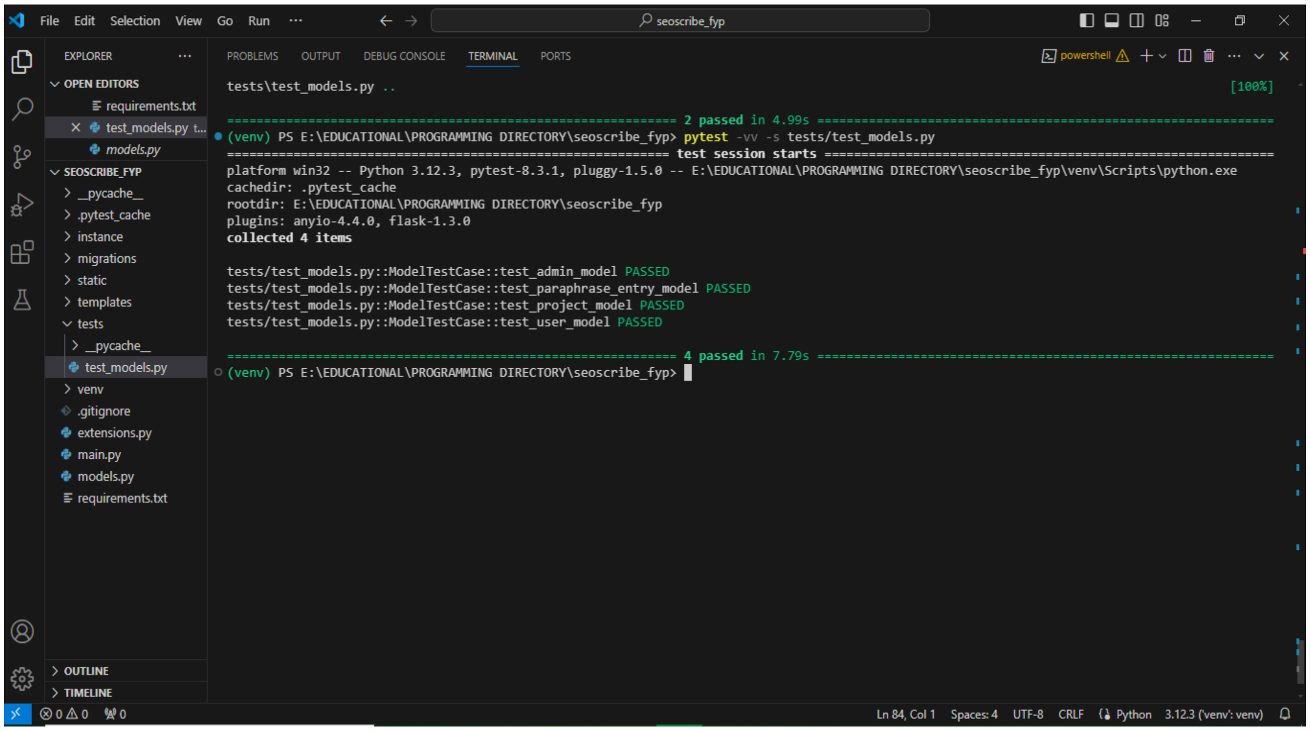
### *4.3.2.13 Hyperlink Insertion*

| | |
|---|---|
| **TC ID** | HyperlinkInsertion_01 |
| **Test Case Name** | Insert Hyperlink in Text Editor in SeoScribe App |
| **Objective** | To verify that a logged-in user can insert a hyperlink in the text editor. |
| **Precondition** | User is on the text editor page.<br>User has text inside the editor. |
| **Actor** | User |

| **Test Step** | **Action** | **Test Data** | **Expected Result** | **Test Result** |
|---|---|---|---|---|
| 1 | **Select Text:** User selects text | Text: click here | Text is selected successfully | |
| 2 | **Select Hyperlink Icon:** User clicks hyperlink icon from toolbar | None | System opens a window prompting user to provide a link | |
| 3 | **Input Link:** User inputs the link and clicks done | Link: https://example.com | System associates the link with the selected text | |
| 4 | **Validate:** Verify that the selected text is displayed as anchor text for the link | None | Selected text is displayed as anchor text for the link | |

Table 4.16 TC13

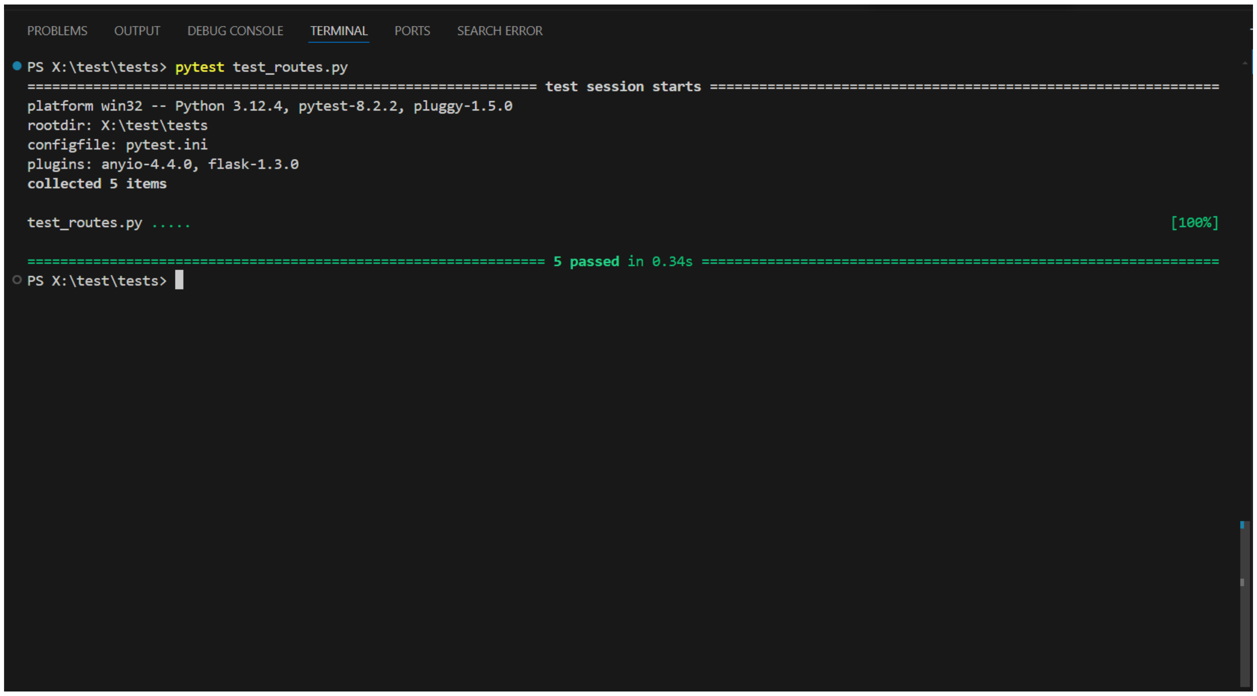
### 4.3.3 Unit Testing Using PYTEST



```
tests\test_models.py ..                                                                                   [100%]

============================================================ 2 passed in 4.99s ============================================================
(venv) PS E:\EDUCATIONAL\PROGRAMMING DIRECTORY\seoscribe_fyp> pytest -vv -s tests/test_models.py
=========================================================== test session starts ===========================================================
platform win32 -- Python 3.12.3, pytest-8.3.1, pluggy-1.5.0 -- E:\EDUCATIONAL\PROGRAMMING DIRECTORY\seoscribe_fyp\venv\Scripts\python.exe
cachedir: .pytest_cache
rootdir: E:\EDUCATIONAL\PROGRAMMING DIRECTORY\seoscribe_fyp
plugins: anyio-4.4.0, flask-1.3.0
collected 4 items

tests/test_models.py::ModelTestCase::test_admin_model PASSED
tests/test_models.py::ModelTestCase::test_paraphrase_entry_model PASSED
tests/test_models.py::ModelTestCase::test_project_model PASSED
tests/test_models.py::ModelTestCase::test_user_model PASSED

============================================================ 4 passed in 7.79s ============================================================
(venv) PS E:\EDUCATIONAL\PROGRAMMING DIRECTORY\seoscribe_fyp>
```

Figure 4.13 Unit Test

### 4.3.4 Routes Testing



```
PS X:\test\tests> pytest test_routes.py
============================================================= test session starts =============================================================
platform win32 -- Python 3.12.4, pytest-8.2.2, pluggy-1.5.0
rootdir: X:\test\tests
configfile: pytest.ini
plugins: anyio-4.4.0, flask-1.3.0
collected 5 items

test_routes.py .....                                                                                                                    [100%]

============================================================== 5 passed in 0.34s ==============================================================
PS X:\test\tests>
```

Figure 4.14 Routes Test

# Chapter 5 Results and Discussions

## 5.1 Achieved Results

Following is the detailed summarization of key achievements iteration wise.

### *5.1.1 Iteration 1: Core Functionalities Implementation – 1*

1) **Homepage:** Homepage was designed and implemented. It includes a navbar for easy navigation and different section displaying services, and about page and contact us page.
2) **Paraphraser**: The Paraphraser tool was designed and implemented. It allows the user to input a paragraph or sentence and paraphrase that sentence.
3) **Keyword Tool**: Keyword generation tool was developed to allow user to generate variations of a focus keyword. The tool uses a web scraper to fetch search results based on the keyword and extract keywords from the fetched data using the en-core-web-mb model.

### *5.1.2 Iteration 2: Core Functionalities Implementation – 2*

1) **User Authentication:** Login and Signup pages were created along with their backend logic. The user authentication feature allows users to create accounts and login in to their accounts.
2) **Writing Assistant:** Project Management features were implemented along with the Text Editor and sidebar. Project Management allows user to create, delete and edit projects. Text editor allows users to input text and apply several editing operations. Sidebar displays SEO and Readability suggestions to the user.
3) **SQLite Integration:** SQLite database was integrated to store user authentication data, user projects and keywords. Models were created for user, project and project entry.

### *5.1.3 Iteration 3: Modules Integration*

1) **Keyword Too1 integration with Writing Assistant:** Keywords generation functionality was integrated with the writing assistant. Thus, allowing users to generate keyword for a project.
2) **User Permissions and services restrictions:** User roles and access permissions were defined. Services permissions were set thus allowing only logged in user to access the writing assistant feature.

### *5.1.4 Iteration 4: Testing, Finalizing and Reviewing Documentation*

1) **Integration and System Testing:** Integration and system testing was carried out using PYTEST flask extension. System routes were tested along with unit testing and end to end testing was done to validate the system as whole.
2) **Documentation:** Documentation was finalized, testing results were added. Documentation was extensively reviewed and proofread to check for any mistakes or compilation errors.

## 5.2 Summary of key features

1) **User Friendly Interface:** A user-friendly Interface was developed focusing on usability. Warning prompts were integrated to guide the user in different operations. Navbar was added to allow users to find their way through the website.
2) **Paraphraser Tool:** A paraphraser tool was developed to allow user to paraphrase blocks of text. User only have to either write into the input field or just simply paste the text and system will paraphrase the given input withing seconds.
3) **Keyword Generation Tool:** It allows user to generate variations and LSI for a focus keyword. User inputs the focus keyword and system will generate variation and LSI for that given keyword.
4) **Writing Assistant:** A user-friendly writing assistant was developed to allow users to get SEO and readability Suggestions as they write their content.
5) **Dynamic Keyword Checker and Suggestions:** The writing assistant provides a dynamic and real time suggestions as the user writes their content; suggestions are updated incase user makes any changes to the text.

## 5.3 Future Work

1) **Competitive Analysis:** Develop functionalities to analyze competitors' content, identify weak point in their content and provide suggestions based on that.
2) **Outline Generation**: Develop a feature that can analyze competitor content and generate an outline based on keyword use, frequency and SEARP rank.
3) **Enhanced UI:** Continuous refining of the UI to make sure it is as user friendly and aesthetically pleasing as possible**.**
4) **Content Score Meter**: Implement a Content rating system, that can analyze the content and provide an SEO and readability Score based on the suggestions and keywords usage.

# Chapter 6 Conclusion

The successful conclusion of the SEOscribe: Content Writing Assistant project marks a significant milestone in digital content optimization. This project was executed in four meticulously planned iterations, each contributing critical functionalities. The first iteration saw the implementation of the homepage, paraphraser tool, and keyword generation tool. The second iteration focused on user authentication, writing assistant features, and SQLite database integration. The third iteration was dedicated to integrating the keyword tool with the writing assistant and defining user permissions. Finally, the fourth iteration concentrated on rigorous integration and system testing, as well as finalizing and reviewing documentation.

Looking forward, we have identified several areas for future enhancement, including competitive content analysis, outline generation, UI refinement, and the implementation of a content score meter. These improvements will ensure that SEOscribe remains relevant and continues to meet the evolving needs of its users.

In summary, SEOscribe has successfully achieved its initial objectives, establishing a robust foundation for future growth and innovation. Our dedication to user engagement, security, and continuous improvement will drive the platform's ongoing success, setting a new standard in the field of content creation and optimization.

# References

[1] SEMrush, "SEMrush Writing Assistant: SEO Writing Tool for Optimizing Content," Accessed: Aug. 20, 2024. [Online]. Available: https://www.semrush.com

[2] S. Smith, "How to Use SEMrush Writing Assistant to Improve Your Content," Content Marketing Institute, Sep. 12, 2023. [Online]. Available: https://contentmarketinginstitute.com

[3] Yoast SEO, "Yoast SEO: The #1 WordPress SEO Plugin," Accessed: Aug. 20, 2024. [Online]. Available: https://yoast.com/wordpress/plugins/seo/

[4] D. Johnson, "Optimizing Your WordPress Site with Yoast SEO," WP Beginner, Jul. 20, 2023. [Online]. Available: https://www.wpbeginner.com

[5] R. Brown, "Advanced SEO Techniques with Yoast SEO," Search Engine Journal, Jun. 14, 2023. [Online]. Available: https://www.searchenginejournal.com

[6] Surfer SEO, "Surfer SEO: Data-Driven SEO Tool for Content Optimization," Accessed: Aug. 20, 2024. [Online]. Available: https://surferseo.com

[7] J. Williams, "A Comprehensive Guide to Using Surfer SEO," SEO Blog, Apr. 10, 2023. [Online]. Available: https://seoblog.com

[8] M. Allen, "Leveraging Surfer SEO for Effective Content Strategy," Moz Blog, May 8, 2023. [Online]. Available: https://moz.com/blog

[9] AISEO, "AISEO: AI-Powered Tools for Content and SEO," Accessed: Aug. 20, 2024. [Online]. Available: https://aiseo.ai

[10] P. Davis, "Maximizing SEO with AISEO's AI Tools," AI Trends, Feb. 28, 2024. [Online]. Available: https://aitrends.com