|  |
| --- |
| HDSDEV |
| Requirements Specification (RS) |
| Insert your project title here |

|  |
| --- |
| Alexandru Georgescu  07/01/24 |

Requirements Specification (RS)

Document Control

Revision History

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Date** | **Version** | **Scope of Activity** | **Prepared** | **Reviewed** | **Approved** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Distribution List

|  |  |  |
| --- | --- | --- |
| **Name** | **Title** | **Version** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Related Documents

|  |  |
| --- | --- |
| **Title** | **Comments** |
|  |  |
|  |  |

**Table of Contents**

[Requirements Specification (RS) 1](#__RefHeading___Toc596428_381736926)

[Document Control 1](#__RefHeading___Toc596430_381736926)

[Revision History 1](#__RefHeading___Toc596432_381736926)

[Distribution List 1](#__RefHeading___Toc596434_381736926)

[Related Documents 1](#__RefHeading___Toc596436_381736926)

[1 Introduction 4](#__RefHeading___Toc596438_381736926)

[1.1 Purpose 4](#__RefHeading___Toc596440_381736926)

[1.2 Project Scope 4](#__RefHeading___Toc596442_381736926)

[1.3 Definitions, Acronyms, and Abbreviations 5](#__RefHeading___Toc596444_381736926)

[2 User Requirements Definition 6](#__RefHeading___Toc596446_381736926)

[3 Requirements Specification 6](#__RefHeading___Toc596448_381736926)

[3.1 Functional requirements 6](#__RefHeading___Toc596450_381736926)

[3.1.1 Top Level Use Case Diagram 7](#__RefHeading___Toc596452_381736926)

[3.1.2 Requirement 1: Register User 8](#__RefHeading___Toc596454_381736926)

[3.1.3 Requirement 2: Login 10](#__RefHeading___Toc596456_381736926)

[3.1.4 Requirement 3: Manage Entries 12](#__RefHeading___Toc596458_381736926)

[3.1.5 Requirement 4: Customize UI 13](#__RefHeading___Toc596460_381736926)

[3.1.6 Requirement 5: View Quotes 15](#__RefHeading___Toc596462_381736926)

[3.1.7 Requirement 6: Provide Feedback 16](#__RefHeading___Toc596464_381736926)

[3.2 Non-Functional Requirements 18](#__RefHeading___Toc596466_381736926)

[3.2.1 Performance/Response time requirement 18](#__RefHeading___Toc596468_381736926)

[3.2.2 Availability requirement 18](#__RefHeading___Toc596470_381736926)

[3.2.3 Robustness requirement 18](#__RefHeading___Toc596472_381736926)

[3.2.4 Security requirement 19](#__RefHeading___Toc596474_381736926)

[3.2.5 Reliability requirement 19](#__RefHeading___Toc596476_381736926)

[3.2.6 Maintainability requirement 19](#__RefHeading___Toc596478_381736926)

[3.2.7 Portability requirement 19](#__RefHeading___Toc596480_381736926)

[4 Interface requirements 19](#__RefHeading___Toc596482_381736926)

[4.1 GUI 19](#__RefHeading___Toc596484_381736926)

[4.2 Application Programming Interfaces (API) 21](#__RefHeading___Toc596486_381736926)

[5 System Architecture 22](#__RefHeading___Toc596488_381736926)

[6 System Evolution 23](#__RefHeading___Toc596490_381736926)

# Introduction

## Purpose

The purpose of this document is to outline the requirements for the SoulJournal web application. SoulJournal is a digital journaling platform that aims to provide a holistic experience for self-reflection, personal growth and mental well-being. The application will serve as a central hub where users can **CRUD**(Create, Read, Update, Delete) journal entries, access inspirational quotes and also adjust the interface of the web application. The intended customers for SoulJournal are individuals seeking a convenient and accessible tool to care for their emotional well-being through the practice of journaling.

## Project Scope

The **SoulJournal** application will facilitate the following:

* User registration and account creation
* **CRUD**(Create, Read Update, Delete) of journal entries
* Integration with the Quotables API to fetch and display inspirational quotes
* **UI** (User Interface) customization(themes, fonts, layouts)

**Out of Scope:**

* In the initial version, the application may not support advanced quote personalization based on sentiment analysis

**Motivation for this Project:**

The motivation behind **SoulJournal** is to provide a modern and accessible platform for individuals to engage in the practice of journaling, promoting self-reflection and mental well-being

**Project Objectives:**

**For users:**

* Create and manage personal journal entries
* Access a collection of inspirational quotes
* Customize the application’s appearance and layout

**Project Expectations**

Upon completion, the **SoulJournal** web application will serve as a functioning digital journaling platform.

**Project Assumptions and Restrictions:**

* Due to time constraints, the scope of user reasearch and feedback gathering during the planning phase may be limited;
* The project timeline and milestones may impose restrictions on the number of features that can be implemented.

**Project Risks and Contingency Plan:**

The main risk to the project is the time constraint for implementing all desired features and completing comprehensive documentation. To mitigate this risk, feature prioritization may be necessary to ensure timely delivery of the web application.

## Definitions, Acronyms, and Abbreviations

**User:** An individual who interacts with the **SoulJournal** application, either as a registered user or an unregistered visitor.

**Registered User**: A user who has created an account and can access the full functionality of the application, including creating and managing journal entries, accessing quotes and customizing the application’s interface.

**Journal Entry:** A written record of thoughts, experiences or reflections created by a registered user within the **SoulJournal** application.

**Quote**: An inspirational or motivational statement fetched from the integrate **ZenQuotes** API and displayed within the application.

**API:** Application Programming Interface, a set of protocols and tools for building software applications that allows different software components to communicate and interact with each other.

**UI**: User Interface, the visual and interactive elements of the **SoulJournal** application that allow users to interact with its features and functionality.

**UX**: User Experience, the overall experience and satisfaction a user derives from interacting with the **SoulJournal** application, including its usability, design, and functionality.

**Encryption:** The process of converting data into a coded format to protect its confidentiality and prevent unauthorized access.

……..

# User Requirements Definition

This section outlines the objectives and requirements for the **SoulJournal** application from the customer’s perspective, detailing what users want and need from the system.

**Objectives and Requirements**

1. **Ease of Use:**
   * Users want an easy-to-navigate interface that will require minimal training.
   * Ability to customize their journaling experience to match their preferences, such as themes, fonts, and layouts.
2. **Accessibility:**

* Users want the ability to access their journals across various devices including desktops, laptops, and smartphones.

1. **Privacy and Security:**

* Users require their journal entries to be securely stored and protected from unauthorized access.

1. **Functionality:**
   * Users want efficient ways to search, categorize, and tag journal entries for easy retrieval.
2. **Inspiration and Motivation:**
   * Users want something that will give them a slight idea on what to write about. Users will receive daily inspiratonal quotes from the ZenQuotes API to motivate and inspire their journaling practice. This feature provides a fresh quote each day, hekping to spark creativity and reflection.

When the primary focus is on user needs, certain technical requirements must be also met to ensure the objectives are achieved:

* **Web Browser:** The **SoulJournal** application will be accessible through a a modern web browser, ensuring cross-platform compatibility.
* **Internet Connection**: Users will require an active internet connection to access the **SoulJournal** application, as it will be hosted on a remove server.
* **Computing Device:** Users can access the **SoulJournal** application from various computed devices, such as desktop computers, laptops or smart-phones.
* **Supported Browsers (**examples):
  + Google Chrome
  + Mozilla Firefox
  + Safari
  + Microsoft Edge

# Requirements Specification

The following section outlines all the functional requirements that **SoulJournal** web application should acoompliash and it also describes the main details of the systems and the way the users will interact with the system. We will also investigate non-functional requirements

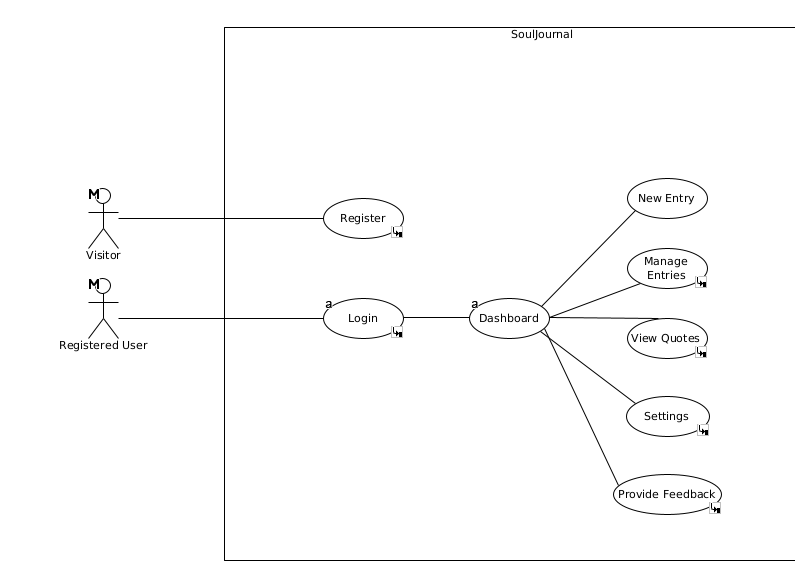
## Functional requirements

The functional requirement describe the core features and functionalities that the SoulJournal system must provide. The requirements are ranked in terms of priority (1 being the highest priority).

1. **User Registration and Authentication**
   * Non-registered users should be able to create a new account by providing necessary information (email, password).
   * Registered users should be able to securely log in to the web application.
2. **Journal Entry Management**
   * Users should be able to create new journal entries.
   * Users should be able to save journal entries securely.
   * Users should be able to retrieve and view their previous journal entries.
   * users should be able to tag or categorize their journal entries for better organization.
3. **Quote Integration**
   * The application should integrate with the **Quotable** API and fetch and display inspirational quotes.
   * Users should be able to filter or search for quotes based on specific topics
4. **User Interface and Customization**
   * The application should provide a user interface **focused** on the writing experience
   * Users should be able to customize the application’s appearance, such as themes, fonts and layouts.
5. **User Feedback and Suggestions**
   * Users should be able to submit feedback, suggestions or report issues within the application.

### Top Level Use Case Diagram

The following diagram outlines the actors and how they interact with the **SoulJourney** web application (**Figure 3.1**).

Figure 3.1.1.1: Top-Level Use-Case Diagram for **SoulJournal**

### Requirement 1: Register User

**Use Case: Register Visitor(Figure 3.1.2.1.)**

Figure 3.1.2.1.: Register User

|  |  |
| --- | --- |
| **Name** | **Register Customer** |
| **Description** | A non-registered user registers with the system |
| **Priority** | This is a **high** priority requirement. |
| **Scope** | This allows new users to create an account and access the full functionality of the **SoulJournal** application. |
| **Precondition** | The user is not registered with the system. |
| **Flow Description** | |
| **Activation** | This use case starts when an unregistered visitor attempts to create a new account. |
| **Main Flow** | 1. The user selects the option register. 2. The system displays the registration form. 3. The user enters their personal information(email, password). 4. The user submits the registration form. 5. The system validates the provided information. 6. The system creates a new user account. 7. The system logs the user in. |
| **Alternate Flow** | |
| **Title** | **Description** |
| **The user cancels the registration process** | A1.1. The user cancels the registration process.  A1.2. The use case terminates |
| **Exceptional Flow** | |
| **Title** | **Description** |
| **The provided information (email/password) is invalid** | E1.1. The system displays an error message. (**Title**)  E1.2. The user is prompted to correct the information.  E1.3. The use case continues from main flow 3. |
| **Post-Condition (For Successful Main Flow)** | |
| **Title** | **Description** |
| **Termination** | The system displays registration is successful and the user is given the option to go to the Login section in the SoulJournal application |
|  |  |

### Requirement 2: Login

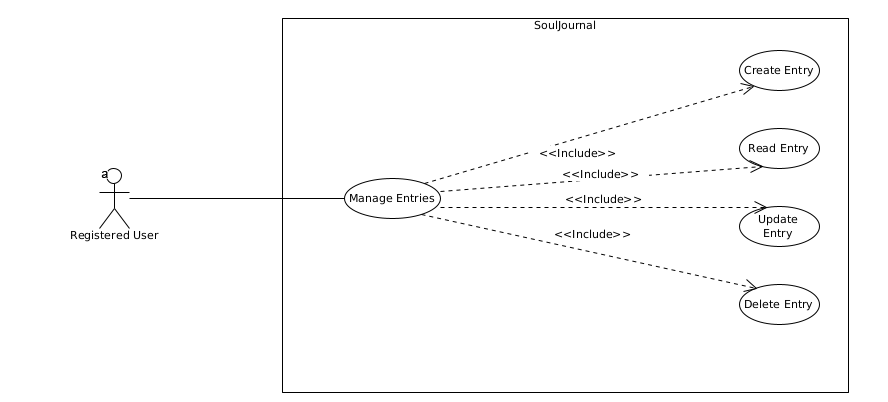
**Use Case: Login Registered User(Figure 3.1.3.1.)**

Figure 3.1.3.1.: Login

|  |  |
| --- | --- |
| **Name** | **Login User** |
| **Description** | A registered user logs in the web application |
| **Priority** | This is a **high** priority requirement. |
| **Scope** | This use case describes the process of a registered user authenticating themselves by providing their credentials to gain access to the SoulJournal application. |
| **Precondition** | The user has a registered account |
| **Flow Description** | |
| **Activation** | This use case starts when a registered user attempts to log in to the application. |
| **Main Flow** | 1. The system displays the login form. 2. The user enters their information (email, password). 3. The user submits the login form. 4. The system validates the provided information. 5. The system logs the user in. |
| **Alternate Flow** | |
| **Title** | **Description** |
| **N/A** | N/A |
| **Exceptional Flow** | |
| **Title** | **Description** |
| **The provided information (email/password) is invalid** | E1.1. The system displays an error message.(**Title**)  E1.2. The user is prompted to enter the correct information.  E1.3. The use case continues from main flow 2. |
| **Post-Condition (For Successful Main Flow)** | |
| **Title** | **Description** |
| **Termination** | The user is logged in to the SoulJournal application. |
|  |  |

### Requirement 3: Manage Entries

**Use Case: Manage Entries(Figure 3.1.4.1.)**

Figure 3.1.4.1.: Manage Entries

|  |  |
| --- | --- |
| **Name** | **Manage Entries** |
| **Description** | A registered user can **C.R.U.D.** (Create, Read, Update or Delete) their journal entries. |
| **Priority** | This is a **high** priority requirement. |
| **Scope** | This use case enables registered users to **C.R.U.D. (**Create, Read, Update or Delete)their personal journal entries within the **SoulJournal** application. |
| **Precondition** | The user is logged in to the SoulJournal application |
| **Flow Description** | |
| **Activation** | This use case starts when a logged-in user wants to manage their journal entries. |
| **Main Flow** | 1. The system displays the user’s journal entries. 2. The user selects an action (create, read, update, or delete). 3. The system executes the selected action on the journal entry. 4. The system updates the display of journal entries. |

### Individual Flow Descriptions for Manage Entries

**Create Entry**

|  |  |
| --- | --- |
| **Name** | **Create Entry** |
| **Main Flow** | 1. The user selects the “Create” action. 2. The system displays a new, blank journal entry form. 3. The user fills out the form and submits it. 4. The system saves the new journal entry. 5. The system updates the display to include the new entry. |
| **Alternate Flow** | |
| **User Cancels Creation** | A1.1. The user cancels the action.  A1.2. The system discards any input.  A1.3. The use case terminates. |
| **Exceptional Flow** | |
| **Title** | **Description** |
| **Error Saving Entry** | E1.1. The system encounters an error saving the entry.  E1.2. The system displays an error message. (**Title**)  E1.3. The user is prompted to try again or cancel.  E1.4. If the user retries, the use case continues at step 3 of the main flow.  E1.5. If the user cancels, the system discards any input and the use case terminates. |

**Read Entry**

|  |  |
| --- | --- |
| **Name** | **Read Entry** |
| **Main Flow** | 1. The user selects the “Read” action. 2. The system displays the selected journal entry in read-only mode. |
| **Alternate Flow** | |
| **User Exits Read Mode** | A1.1. The user exits the read mode.  A1.2. The use case terminates. |
| **Exceptional Flow** | |
| **Title** | **Description** |
| **Error Loading Entry** | E1.1. The system encounters an error loading the entry.  E1.2. The system displays an error message. **(****Title)**  E1.3. The user is prompted to try again or cancel.  E1.4. If the user retries, the use case continues at step 2 of the main flow.  E1.5. If the user cancels, the use case terminates. |

**Update Entry**

|  |  |
| --- | --- |
| **Name** | **Update Entry** |
| **Main Flow** | 1. The user selects the “Update” action. 2. The system displays the selected journal entry in edit mode. 3. The user modifies the entry and submits the changes. 4. The system saves the updated entry. 5. The system updates the display to reflect the changes. |
| **Alternate Flow** | |
| **User Cancels Creation** | A1.1. The user cancels the action.  A1.2. The system discards any changes.  A1.3. The use case terminates. |
| **Exceptional Flow** | |
| **Title** | **Description** |
| **Error Saving Entry** | E1.1. The system encounters an error saving the updated entry.  E1.2. The system displays an error message. **(****Title)**  E1.3. The user is prompted to try again or cancel.  E1.4. If the user retries, the use case continues at step 3 of the main flow.  E1.5. If the user cancels, the system discards any changes and the use case terminates. |

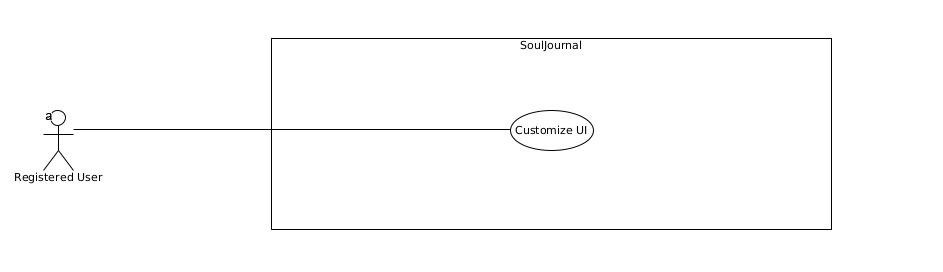
**Delete Entry**

|  |  |
| --- | --- |
| **Name** | **Delete Entry** |
| **Main Flow** | 1. The user selects the “Delete” action. 2. The system prompts the user to confirm the deletion. 3. The user confirms the deletion. 4. The system deletes the journal entry. 5. The system updates the display to remove the deleted entry. |
| **Alternate Flow** | |
| **User Cancels Deletion** | A1.1. The user cancels the action.  A1.2. The use case terminates. |
| **Exceptional Flow** | |
| **Title** | **Description** |
| **Error Deleting Entry** | E1.1. The system encounters an error deleting the entry.  E1.2. The system displays an error message. **(****Title)**  E1.3. The user is prompted to try again or cancel.  E1.4. If the user retries, the use case continues at step 3 of the main flow.  E1.5. If the user cancels, the use case terminates. |

|  |  |
| --- | --- |
| **Post-Condition (For Successful Main Flow)** | |
| **Title** | **Description** |
| **Termination** | The user’s journal entries are updated according to the selected action. |
|  |  |

### Requirement 4: Customize UI

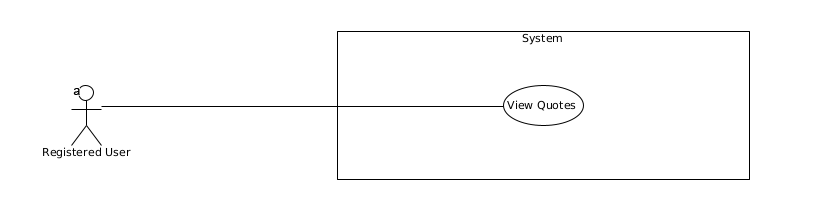
**Use Case: Customize UI(Figure 3.1.5.1)**

Figure 3.1.5.1: Customize UI

|  |  |
| --- | --- |
| **Name** | **Customize UI (User Interface)** |
| **Description** | A registered user can customize the appearance of the SoulJournal application, such as themes and fonts. |
| **Priority** | This is a **medium** priority requirement. |
| **Scope** | This use case describes the process of a registered user customizing the appearance of the SoulJournal application by adjusting themes, fonts, and other UI elements. |
| **Precondition** | The user is logged in to the SoulJournal application |
| **Flow Description** | |
| **Activation** | This use case starts when a logged-in user wants to customize the application's user interface. |
| **Main Flow** | 1. The system displays the customization options. 2. The user selects the desired customization settings. 3. The user applies the changes. 4. The system updates the application’s user interface. |
| **Alternate Flow** | |
| **Title** | **Description** |
| **N/A** | N/A |
| **Exceptional Flow** | |
| **Title** | **Description** |
| **There is an error applying the customization settings.** | E1.1. The system displays an error message. **(****Title)**  E1.2. The user is prompted to try again or cancel the changes.  E1.3. The use case continues at step 2 of the main flow. |
| **Post-Condition (For Successful Main Flow)** | |
| **Title** | **Description** |
| **Termination** | The application’s user interface is updated according to the user’s customization settings. |
|  |  |

### Requirement 5: View Quotes

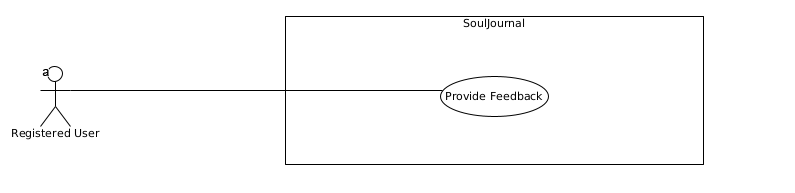
**Use Case: View Quotes (Figure 3.1.6.1)**

Figure 3.1.6.1: View Quotes

|  |  |
| --- | --- |
| **Name** | **View Quotes** |
| **Description** | This allows registered users to view inspirational quotes fetched from the ZenQuotes API. |
| **Priority** | This is a **medium** priority requirement. |
| **Scope** | This use case describes the process of a registered user customizing the appearance of the SoulJournal application by adjusting themes, fonts, and other UI elements. |
| **Precondition** | The user is logged in to the SoulJournal application |
| **Flow Description** | |
| **Activation** | This use case starts when a logged-in user wants to view inspirational quotes. |
| **Main Flow** | 1. The system fetches quotes from the ZenQuotes API. 2. The system displays the fetched quotes. 3. The user views the quotes. |
| **Alternate Flow** | |
| **Title** | **Description** |
| **The user searches for quotes based on specific topics** | A1.1. The system displays the filtered or searched quotes.  A1.2. The use case continues at step 3 of the main flow. |
| **Exceptional Flow** | |
| **Title** | **Description** |
| **There is an error fetching quotes from the ZenQuotes API** | E1.1. The system displays an error message. **(****“There is an error displaying the quotes”)**  E1.2. The user is prompted to try again.  E1.3. The use case continues at step 1 of the main flow. |
| **Post-Condition (For Successful Main Flow)** | |
| **Title** | **Description** |
| **Termination** | The user has viewed the inspirational quotes. |
|  |  |

### Requirement 6: Provide Feedback

**Use Case: Provide Feedback (Figure 3.1.7.1)**

Figure 3.1.7.1: Provide Feedback

|  |  |
| --- | --- |
| **Name** | **Provide Feedback** |
| **Description** | This use case describes the process of a user providing feedback, suggestions, or reporting issues about the SoulJournal application. |
| **Priority** | This is a **low** priority requirement. |
| **Scope** | This use case allows registered users to submit feedback, suggestions, or report issues regarding the SoulJournal application. |
| **Precondition** | The user is logged in to the SoulJournal application |
| **Flow Description** | |
| **Activation** | This use case starts when a logged user wants to provide feedback, suggestions, or report an issue |
| **Main Flow** | 1. The system displays the feedback form. 2. The user enters their, feedback, suggestions, or issue details. 3. The user user submits the feedback or issue report. 4. The system displays a confirmation message |
| **Alternate Flow** | |
| **Title** | **Description** |
| **The user cancels the feedback submission** | A1.1. The system discards any entered information  A1.2. The use case terminates. |
| **Exceptional Flow** | |
| **Title** | **Description** |
| **There is an error processing the feedback or issue report.** | E1.1. The system displays an error message. **(****Title)**  E1.2. The user is prompted to try again.  E1.3. The use case continues at step 2 of the main flow. |
| **Post-Condition (For Successful Main Flow)** | |
| **Title** | **Description** |
| **Termination** | The user’s or visitor’s feedback, suggestions, or issue report has been submitted and processed by the system. |
|  |  |

## Non-Functional Requirements

### Performance/Response time requirement

The deployed SoulJournal web application must be capable of handling simultaneous users without compromising performance. The following response time targets should be met:

1. **User Authentication:**
   * Research indicates that users expect login processes to be quick and seamless. A response time of 3 seconds is generally considered acceptable to prevent user frustration and abandoment. Studies have shown that 53% of users will abandon a website that takes more than 3 seconds to load. (**[Hobo](https://www.hobo-web.co.uk/your-website-design-should-load-in-4-seconds/" \l ":~:text=53%25 of mobile site visits,a negative Google ranking factor.)**)
2. **Page Loading and Rendering of Journal Entries:**
   * For content-rich pages, such as those displaying journal entries, a 5-second load time is considered the upper limit for acceptable performance. Studies also have shown that an additional two second delay in load time resulted in abandonment rates of up to 87%. (**[Hobo](https://www.hobo-web.co.uk/your-website-design-should-load-in-4-seconds/" \l ":~:text=53%25 of mobile site visits,a negative Google ranking factor.)**)

### Availability requirement

The SoulJournal application will be deployed on a cloud platform that will meet the stipulated performance and response time. End-users will require a device with an internet connection to access the system.

### Robustness requirement

The deployed SoulJournal web application must handle errors and unexpected conditions. The system should be designed to recover from issues without compromising availability.

### Security requirement

All user passwords will be hashed and salted using a cryptographic algorithm before storing them in the database.

All journal entries and sensitive user data will be encrypted before being stored in the database.

### Reliability requirement

The selected cloud hosting platform for SoulJournal must consistently meet the specified availability and performance/response time criteria.

### Maintainability requirement

The web application source code should be easy to access and modify in case of updating the features and to fix the issues.

### Portability requirement

The web application will be available to access from different devices (computer, laptop, smart-phone).

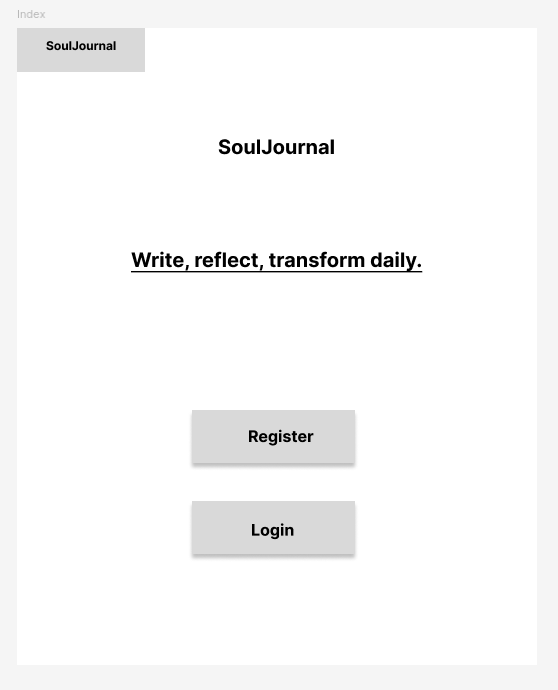
# Interface requirements

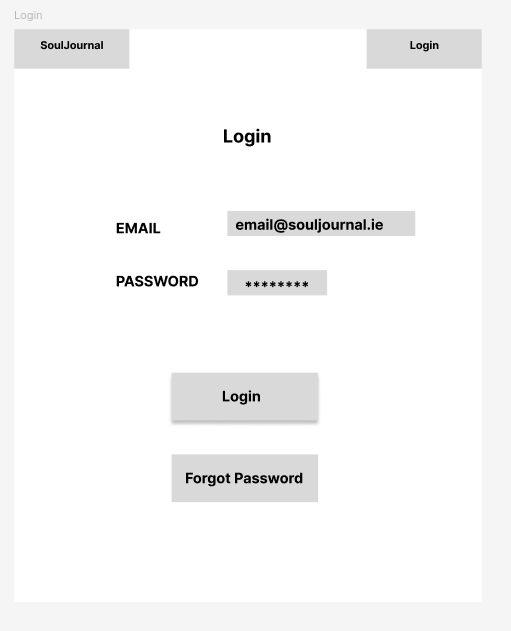
## GUI

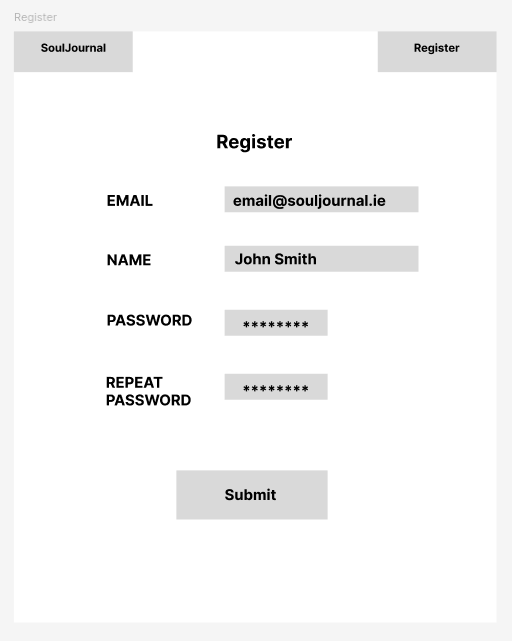
The **SoulJournal** web application will adopt a mobile-first design approach for its graphical user interface. This approach ensures a consistent and optimized user experience across various devices, with a primary focus on smaller screens such as smartphones.

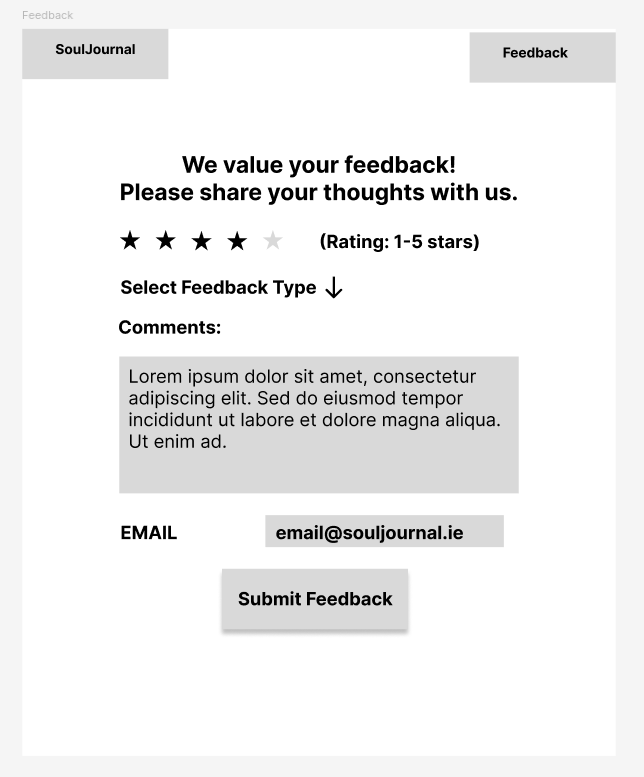
Simple wireframes will be used to illustrate the GUI for each core function of the **SoulJournal** system. These wireframes will serve as a starting point for the visual design and layout of the application.

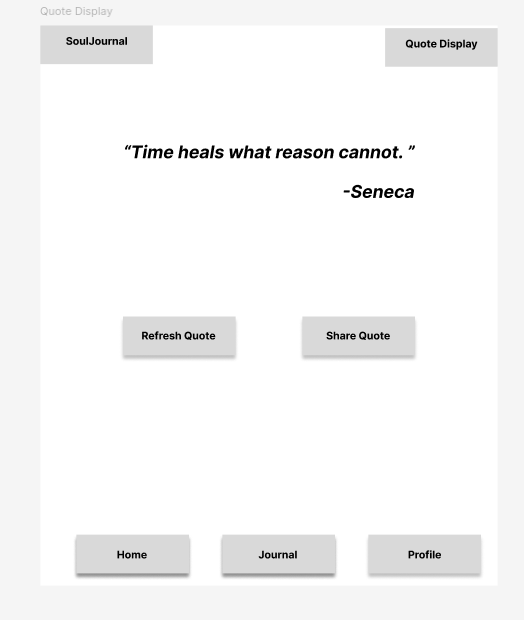
1. **User Login:** The login interface will have a simple form where users can login with their email and password. (Fig. 2)
2. **User Registration:** The registration interface will feature a simple form where new users can enter their personal information (email, password) to create an account. Appropriate form validation and error handling will be implemented. (Fig. 3)
3. **Feedback**: A dedicated interface will allow users to submit feedback, suggestions, or report issues regarding the SoulJournal web application. (Fig. 4)
4. **Quote Display:** Inspirational quotes fetched from the ZenQuotes API will be prominently displayed on the main dashboard or landing page of the application. (Fig. 5)
5. **Journal Entry Creation**: The interface for creating a new journal entry will feature a clean and distraction-free writing space, with a prominent text editor occupying the majority of the screen. Minimal UI elements, such as a toolbar for formatting options and a button to save the entry, will be present. (Fig. 6)

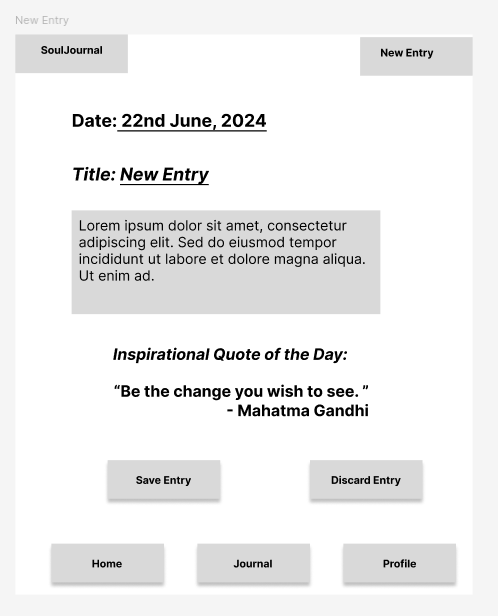
Figure 1: Index Page

Figure 2: User Login

Figure 3: User Registration

Figure 4: Feedback

Figure 5: Quote Display

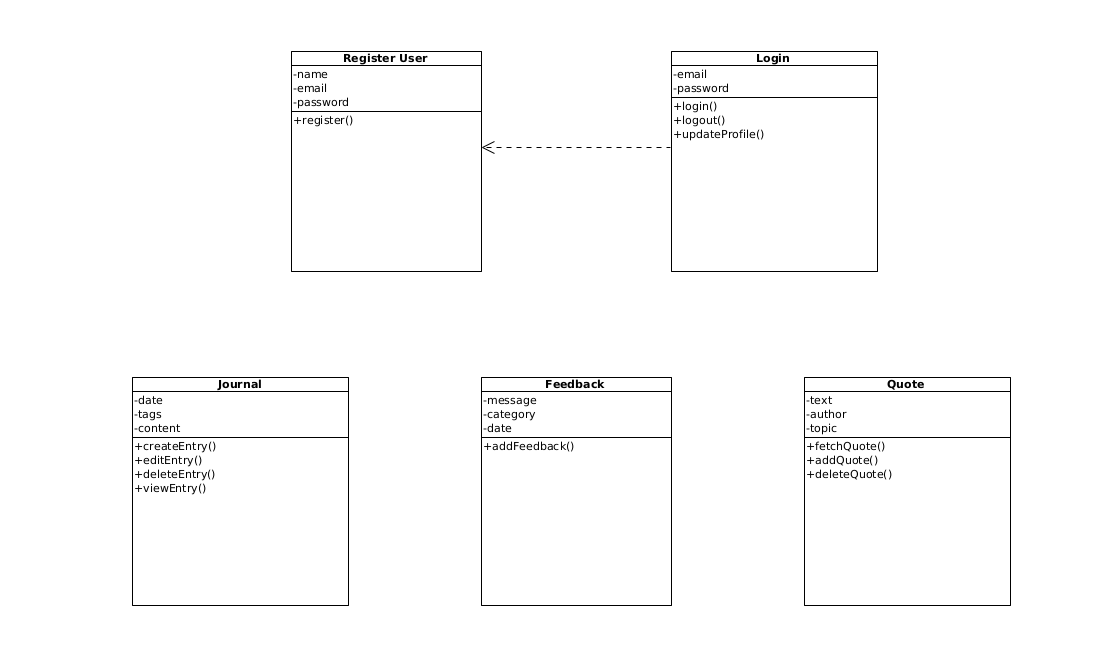
Figure 6: New Entry

## Application Programming Interfaces (API)

The **SoulJournal** web application will integrate with the **ZenQuotes** API:

* The API will be used to fetch and display inspirational quotes within the **SoulJournal** web application.
* The API provides a collection of quotes categorized by various topics, allowing the application to retrieve relevant quotes based on user prefferences.

# System Architecture

Figure 5.1: System Architecture

We’ve decided to separate the web application into five classes:

* User
* Login
* Journal
* Feedback
* Quote

Each class has a well-defined responsibility:

* **User** class handles user registration
* **Login** class manages authentication and profile updates
* **Journal** class deals with journal entry operations
* **Feedback** class deals with feedback.
* **Quote** class deals with the fetch of quotes.

Each class encapsulates its own data (attributes) and behavior (operations).

# System Evolution

At the moment there are no future features to be implemented, but this will be taken into consideration.