Software Requirements Specification

for

Insight Write

Version 1.4 approved

Prepared by Miguel Gonzalez Torres, Tony Lau, Hoang Le,

Keyvan Mahmoodzadeh Kani, Will May, Hayk Vardapetyan,

Jian Verdad, Niyusha Zarnegar

California State University - Los Angeles

May 10th, 2023

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Will	3/7/24	Formatting edits	1.0
Jian	4/30/24	Update Sections 1, 3 & 5	1.1
Will	5/2/2024	Update Sections 4.2 - 4.4 and 6	1.2
Dempsey	5/3/24	Update Sections 2	1.3
Dempsey	5/2/24	Sections 4 and table of content hyperlinks	1.4

1. Introduction

1.1 Purpose

The introduction to the Software Requirements Specifications (SRS) document for our mental health-focused web journaling application provides an overarching view of the software's intended functionalities and user benefits. It clarifies that the document aims to detail what the application will accomplish, specifically designed to support users in tracking and enhancing their mental well-being through journaling. This version of the SRS outlines comprehensive requirements, excluding implementation specifics reserved for the Software Design Document (SDD).

It covers all software aspects, ensuring developers and stakeholders are aligned on the project scope and objectives. This document will cover the requirements up to planned release 24.2 and covers all requirements of that release.

1.2 Intended Audience and Reading Suggestions

- The customers and Users: Focus on the introduction and interface
- Developers: Begin with introduction and understand the app’s goal
- Testers: Begin with the requirements
- Product Managers: Begin with the introduction and for an overview of the project
- Marketing and sales: Begin with user interface to learn about user features
- UI/UX Designers: Begin with introduction and user interfaces

1.3 Product Scope

The software will allow users to:

- Reflect and express their emotions, thoughts, and plans through journaling

Benefits of the software

- Privacy and Security
- Memory Preservation
- Emotional Release and Stress Reduction
- Self Reflection
- Planning and Time Management
- Access to Local Weather Data

1.4 Definitions, Acronyms, and Abbreviations

Django: Django is a high-level Python web framework that enables rapid development of secure and maintainable websites.

Bootstrap: Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development.

OpenWeather: OpenWeather is an application for servicing weather data that can be connected through an API.

API: Application Programming Interface, a set of rules that allows different software applications to communicate with each other.

2. Overall Description

2.1 System Analysis - **Hoang Le - (Dempsey)**

- Offering users a complete platform for expressive journaling
- Creating an efficient account register, sign-in/log-out management interface
- Designing an overall dashboard page aids users on their navigation through each application section; includes: journal entries, mindfulness resources, and google API calendar.
- Journal entries that allow customization with the Text Formatting Buttons, Image attachment, and linking hyperlinks (URL links) for the journal entries sections.

- Having encryption (https) and access control management help to ensure the users' privacy and personal data
- Adding features such as inspiring quotations, daily prompts, weather, up-to-date widget calendars, and meditation tabs to practice mindfulness to improve user experience.

Potential Technical Barriers

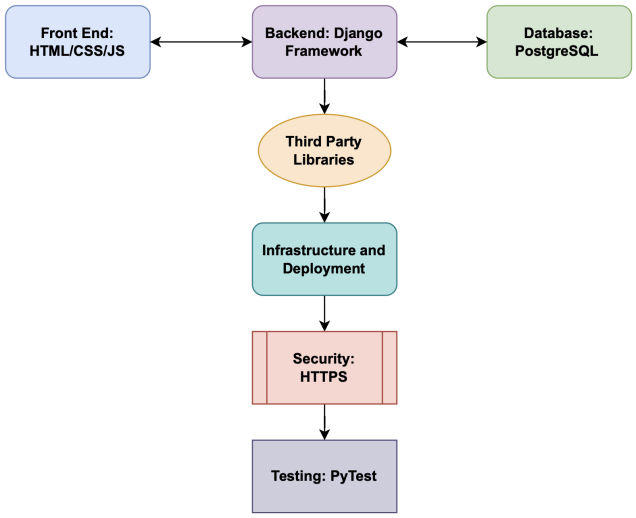
- Working with diverse computer programming language tools: HTML/CSS/JS, Bootstrap, Django Web Framework, and PostgreSQL API
- Implementing secure authorization using Google login and other logins with AllAuth API
- Implementing database for frontend elements and resolving issues around Docker

Technical Barrier Solutions

- Having members of the team to do research, and apply their strength into different development processes, and researching and sharing about development processes.

2.2 Product Perspective - (Hoang Le)

<DFD-0 data flow diagram> - (Will)

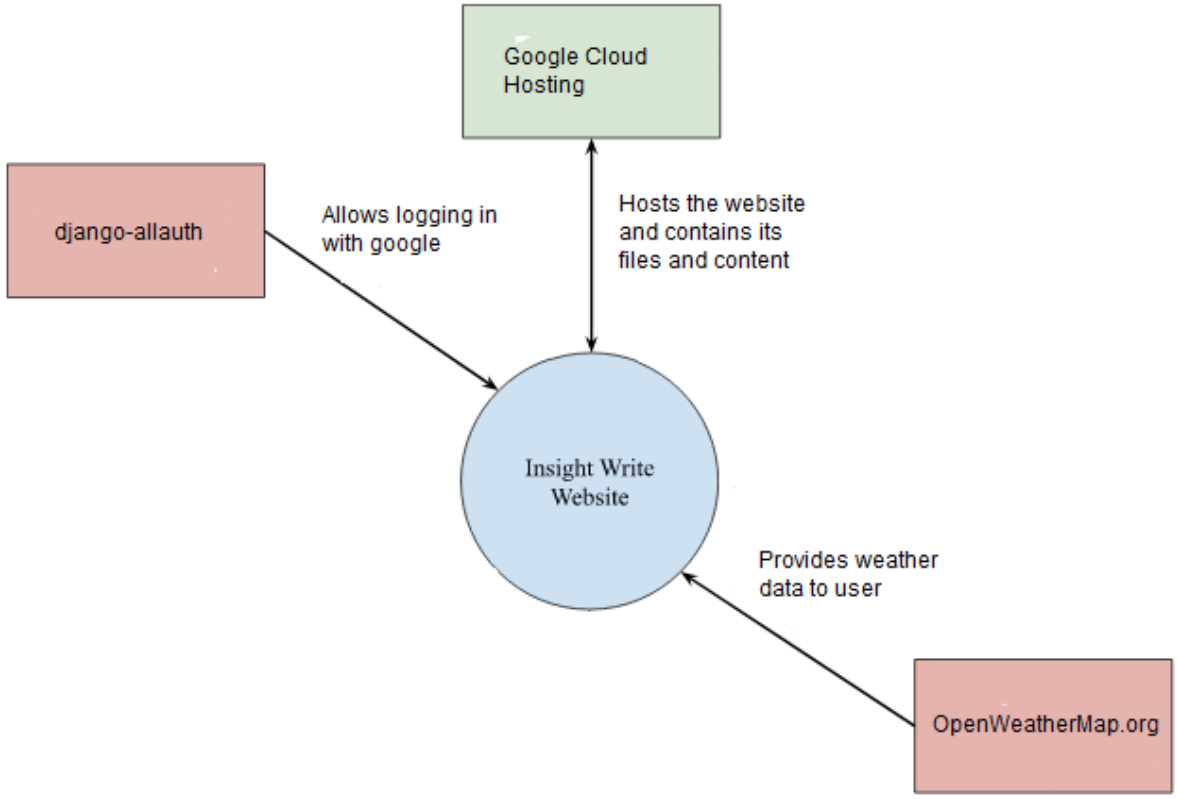


In this section, we will provide an overview of how our Website compares to other products, its dependability, market position and its potential as a component in bigger system:

We are envisioning that our Journal Website is intended to be a stand-alone platform as a solution for well-being and personal reflections, therefore, it is not meant to be a part of a bigger system. Our website will provide users with a full and personalized digital journaling experience, operate independently while interacting with other external systems to improve its functionality and users' accessibility.

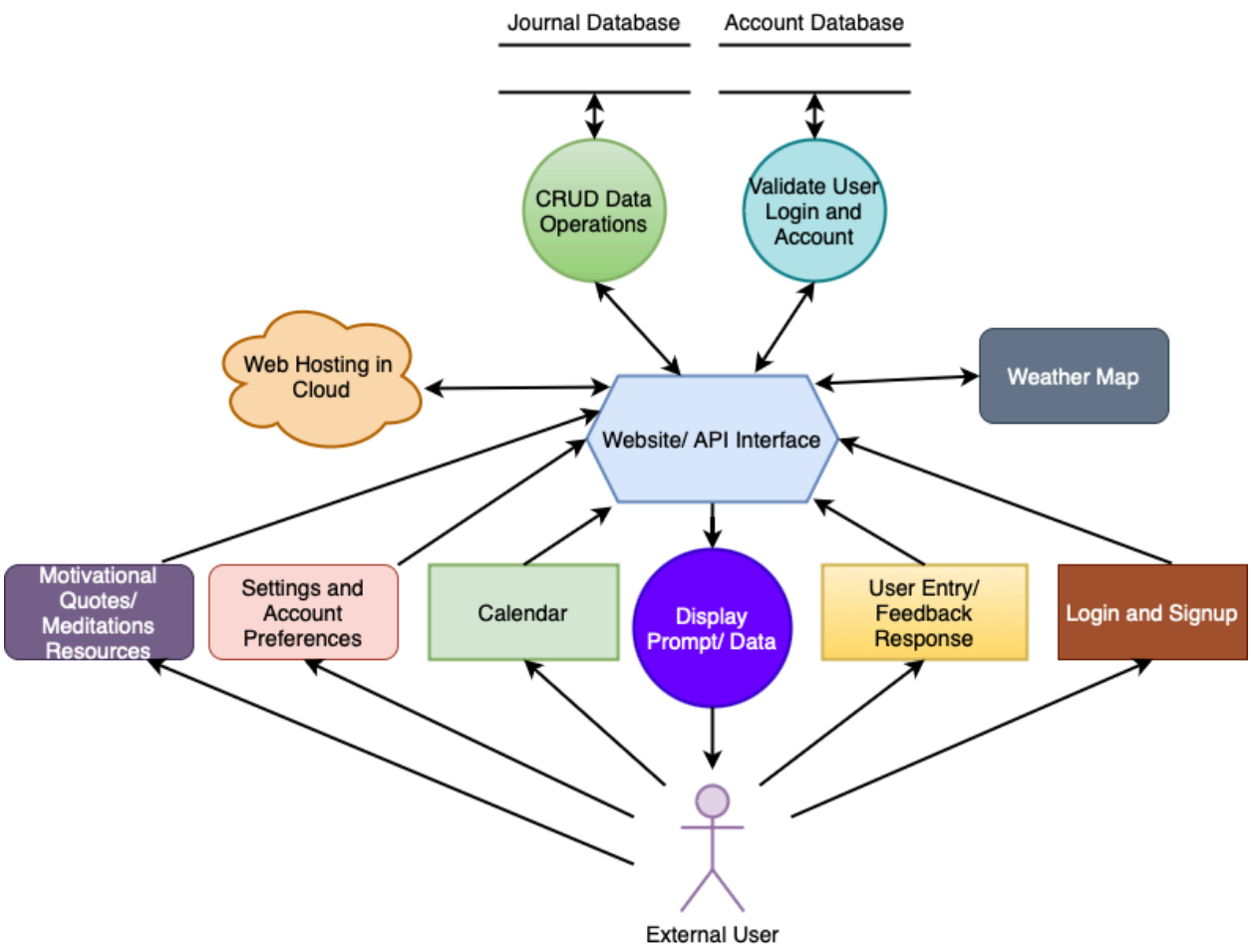
When compared with other applications, there are stand out points that our website brings which are the included features in the journal page to contribute for users to have more control interaction, having an user-friendly design, additional benefit with the meditation tab for convenience when user wants to journal and practice mindfulness and finally prioritizing over user privacy.

Context Model (Hoang Le):



2.3 Product Functions - (Hoang Le)

<DFD-1 data flow diagram> - (Will)



Major functions:

User Registration

- Sign-Up: Enable users to create accounts securely.
- Login: Enable users that have a google account or an account saved to the application's database to login to their personal account.

User Login with Google Allauth

- Login: Enable users to login with their google accounts.

Journal Functionality:

- Create Journal Entries: Journal entries created by users to document their thoughts, experiences, and emotions.
- Edit and Update Entries: Edit and Update existing entries to reflect any changes or updates made in the future
- Delete Entries: Getting rid of unwanted or outdated journal entries.
- Text Formatting Buttons: Include Clear, Bold, Italic, Underline, Headers, Bullet Points, Number Lists, Check Lists, Link, Tag.
- Journal Title & Date: Give the option to provide a title and date for their journal.
- Save Entries Display: Saved entries will be displayed underneath.
- Sidebar: The sidebar is collapsible by choice, for a more focused display.

Journal Web Page Features:

- Word-count: Helps users keep track of their writing process
- Time bar: Displaying the current day in the week and time clock
- Calendar: Small move-around widget functional calendar
- Weather: Third-party OpenWeatherMap API to access current weather for users in most locations around the world, many countries and cities.
- Dark/Light Mode: Users can switch between 2 modes (dark and light), depending on their preference, mainly for visibility and eye health.
- Daily prompt: Inspirational prompt will be randomized each day to help individuals who are new to journaling, to get them started. Within 24 hours of the day, users will be able to make changes to their answers, which will be saved.

Motivational Features:

- Motivational Quotes: Giving access to a library of uplifting quotations

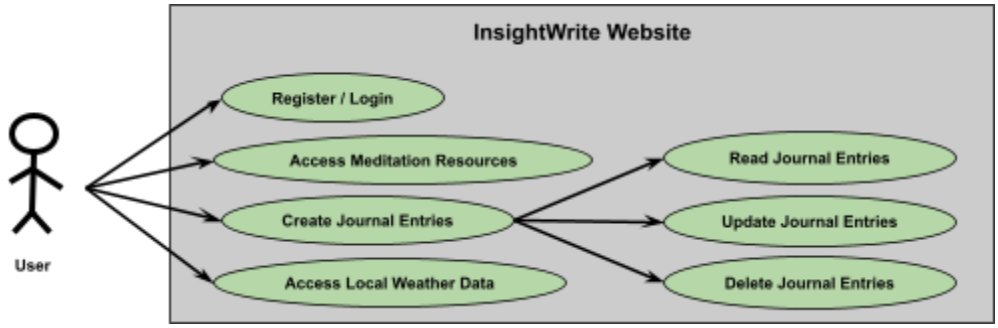
Meditation Section:

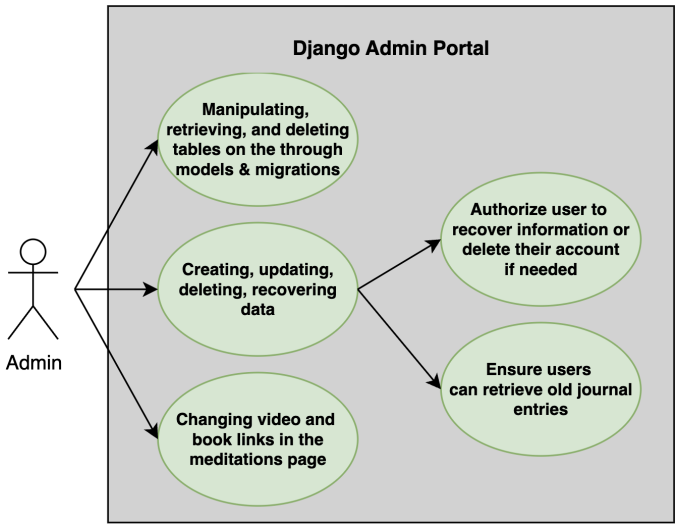
- Meditation Tab: Navigate into a specialized meditation section for mindfulness and relaxation. Recommends videos and books for the user to partake in.

Accessibility:

- Navigation Bar Theme: Aiding user with their navigation process
- User-Friendly Interface: Making sure the interface is easy to use for users of all technical backgrounds
- Encryption and Data Security: Setting a strong security measure in place to protect user personal information and also encrypt journal entries to increase privacy
-

Use-Case Diagram:





2.4 User Classes and Characteristics

Teenagers (Most frequent): Expected to very frequently use this website due to their struggle to control their feelings more than others. Should have enough technical expertise to use a simple website such as ours. Simple features that don't require inputs directly from the user such as the motivational quotes are meant to positively impact these users.

Adults (Very frequent): More technically experienced and educated than the teenagers. Will make more use features to aid the responsible such as the meditation section for a mental health aid and the calendar to keep track of important dates.

Elders (Somewhat frequent): Likely not to be as technically experienced as the other user classes, which impacts the frequency of this user class using our project. Will likely find a good use in recording moments and memories in their lives, whether from the past or the current time.

2.5 Operating Environment

Currently, we have decided that the operating environment of the website will be using the Windows OS and will be accessible by any hardware using Windows OS.

The environment we expect the software to most often be used in is the user's room to achieve a similar experience of writing in a diary while offering more than what a diary can do. As a website, the software can be used anywhere with a device that can access an internet browser with all of the website's features available.

2.6 Design and Implementation Constraints

Upkeep costs: Hosting a website costs money and attempting to test the code without already paying for a website could lead to new issues stemming from the code attempting to work with the website.

Memory constraints: Server storage could potentially be an issue based on the service we decide to use.

Safety and security considerations: Our project will be under more scrutiny than others due to its functions being focused on the aspect of privacy. May require more time to implement to avoid a major breach in security. Also, supporting data backups of the website will add to the upkeep costs.

2.7 User Documentation

The online guide on how to use the software will be on the website itself, likely on the front page or login page. We may have multiple guides dedicated to different sections of the website if we believe their functions are too complicated to group up in one document.

2.8 Assumptions and Dependencies

The application uses Google Allauth Authentication for user sign up and the Google Calendar API. The sign up, login, and calendar features are available to users without Google accounts, but the full functionality of what the application may offer only extends to Google Account Users.

2.9 Apportioning of Requirements

Currently, we believe that all the requirements must be implemented in the first publicly released version of the system.

3. External Interface Requirements

3.1 User Interfaces

The user interface for this product will be a website utilizing the user's keyboard and mouse to interact. As such, it will require the installation of a browser and will be supported by all major browsers. Regardless of the user's device, having any browser application will allow the user to access the web application.

Sample website screenshots currently not available.

- Web page layout designs will remain thematically consistent across all web pages.
- UI design will be documented in a separate file once development has begun and stabilized.
- Web page styling will be limited to what CSS3 tags are supported by major browser applications.

Styling will also incorporate accessibility to individuals with disabilities such as dyslexia, color blindness, and users of screen readers. These are covered in the following issues:

- Poor color contrast
- Use of color alone to give information
- Lack of text alternatives ("alt text") on images
- Mouse-only navigation (lack of keyboard navigation)

Web Pages:

- Home page
- Login page
- Signup Page
- Journaling dashboard
- Meditation dashboard
- Weather Page

3.2 Hardware Interfaces

This is a web application and will not utilize any hardware interfaces besides the user's browser-viewing electronic device.

3.3 Software Interfaces

Because we are creating and designing a web application from scratch ourselves, we do not have a “customer” with software or system requirements besides ourselves. Thus, this section will instead go over some of the necessities for our desired products, such as APIs, in a mutable list of software and technologies that our product may utilize for certain features.

- Google Cloud Web Hosting
- Google Account Authentication
- Railway Cloud Deployment Platform
- Django (Python Web Framework)
- OpenWeather Software Application

3.4 Communications Interfaces

The InsightWrite application will communicate with the following interfaces:

- **Google Authentication**
- **OpenWeather**

4. Requirements Specification

4.1 Functional Requirements

4.1.1 Journal Management

4.1.1.1 Entry Creation and Editing

- Requirement: The system shall allow users to create, edit, and save journal entries
- Rationale: To enable users to document their thoughts, experiences, and emotions digitally
- Who: All registered users
- Why: Users need a flexible and accessible way to express and reflect on their daily lives, enhancing their mental well-being.

4.1.1.2 Text Formatting

- Requirement: The system shall provide text formatting tools including bold, italic, underline, headers, and lists.
- Rationale: To enhance readability and organization of journal entries.
- Who: All users
- Why: Users require tools to emphasize important parts of their entries and to structure their content effectively.

4.1.1.3 Checklist Creation

- Requirement: The system shall enable users to create checklists within their journal entries
- Rationale: To assist users in task management and organization within their entries
- Who: All users
- Why: Users benefit from integrating task management directly into their journaling process to track personal goals and daily tasks.

4.1.1.4 Image Attachment

- Requirement: The system shall allow users to attach images to their journal entries
- Rationale: To enrich the journaling experience by adding visual context.
- Who: All users
- Why: Users often need to capture and reflect on visual memories or add illustrations to their journal entries.

4.1.1.5 Link Integration

- Requirement: The system shall allow users to add hyperlinks to their journal entries
- Rationale: To enable users to reference external resources and provide additional helpful materials that they want to include
- Who: All users
- Why: Users may want to link to external content for reference or further reading, mentioning music list, etc… within their entries

4.1.1.6 Entry Deletion

- Requirement: The system shall allow users to delete their journal entries
- Rationale: To provide users with control over their data and to support data management
- Who: All users

- Why: Users need the flexibility to manage their entries, including removing outdated or unwanted content

4.1.2 Validate User Login and Account

4.1.2.1 Application User Non-API Signup

- Requirement: The system shall allows users to sign up for an account using a non-API signup methods
- Rationale: To provide straightforward registration process for users in a traditional sense
- Who: New users
- Why: Enable new users to create accounts and access applications features

4.1.2.2 Application User All-Auth Signup

- Requirement: The system shall provide a signup mechanism using All-Auth that supports Google authentication providers
- Rationale: To simplify the registration process by allowing users to sign up using their existing Google email, reducing the barrier to entry.
- Who: New users
- Why: Offering another signup option to increase the accessibility of the application and help leading to higher user acquisition rates.

4.1.2.3 Application User Non-API login

- Requirement: The system shall allow users to log in using a username and password without relying on external API integrations.
- Rationale: To ensure that users can access their accounts independently of third-party service availability.
- Who: Existing users.
- Why: Maintaining a non-API login ensures that the application remains functional for users even when third-party services are down or when users prefer not to use social media accounts for login purposes.

4.1.2.3 Application User All-Auth API login

- Requirement: The system shall allow users to log in using their Google accounts through All-Auth API integrations.
- Rationale: To provide a convenient and fast login process using existing Google email, enhancing user experience and security.
- Who: Existing users who prefer to use their Google account to log in

- Why: Utilizing Google logins can reduce the time it takes for users to access the application.

4.1.3 Weather Map

4.1.3.1 Weather Display

- Requirement: The system shall display current weather information as an exclusive feature inside the journal page showing the right weather icon such as rain, cloud, sunny,... to aid with visual aid on current weather based on users' input on location.
- Rationale: To provide users with real-time weather updates that influence their journaling context and mood.
- Who: All users
- Why: Access to weather information can enhance the journaling experience by allowing users to note the environmental context of their entries, which can affect mood and activities.

4.1.3.2 Weather Data Retrieval

- Requirement: The system shall retrieve weather information using a third-party API using OpenWeatherMap.org forecast website
- Rationale: To ensure accurate and real-time weather data is available.
- Who: All users
- Why: Leveraging reliable external data sources ensures that the application provides accurate and up-to-date weather information, enhancing user trust and engagement.

4.1.4 Meditations Resources

4.1.4.1 Display Related Content and Information

- Requirement: Hyperlinks for images, resources, and various other content will be properly displayed out towards the user. Requires no backend functionality, only front end.
- Rationale: The web page content by no means affects the overall system or any other dependencies. Simply a link to a page that is front-end development only. Easily managed and edited without interacting with any other part of the application.
- Who: Authenticated users and non-users.

- Why: As a well rounded application, the application has created a webpage for user related content to explore outside of the main functionality. Users will interact with the application to find resources all over the rest of the web.

4.1.5 Settings and Account Preferences

- Requirement: The system shall allow users to customize settings and account preferences.
- Rationale: To provide users with control over their personal settings and preferences to enhance their user experience.
- Who: All registered users.
- Why: Enabling personalization of the platform increases user satisfaction and retention by allowing them to tailor the application according to their needs.

4.1.6 Accessing User Calendar

4.1.6.1 The system shall allow registered users to access their personal calendar from the home page or dashboard to effectively plan and manage events.

Child Requirements:

4.1.6.1.1 The system shall provide a clickable link or button on the home page or dashboard for accessing the user calendar.

- Who: Registered users
- Why: To provide easy access to the calendar feature, facilitating efficient event planning and management.

4.1.6.1.2 The system shall authenticate users before granting access to their personal calendar to ensure security and privacy.

- Who: System
- Why: To protect user data and ensure that only authorized individuals can view and manage the calendar.

4.1.7 Managing User Calendar Events

4.1.7.1 The system shall enable users to view, create, update, and delete (CRUD) events within their personal calendar.

Child Requirements:

4.1.8.1.1 The system shall display a visual representation of the user's calendar with existing events for easy viewing and planning.

- Who: Registered users
- Why: To provide users with a clear overview of their scheduled events, facilitating effective planning and management.

4.1.7.1.2 The system shall allow users to create new events by providing necessary details such as title, date, time, and description.

- Who: Registered users
- Why: To enable users to add new events to their calendar, ensuring accurate scheduling and organization.

4.1.7.1.3 The system shall permit users to update existing events by modifying event details or rescheduling dates and times.

- Who: Registered users
- Why: To empower users to make changes to their scheduled events, ensuring flexibility and adaptability in their calendar management.

4.1.7.1.4 The system shall provide users with the capability to delete events they no longer wish to keep in their calendar.

- Who: Registered users
- Why: To allow users to remove unnecessary or outdated events from their calendar, maintaining relevance and organization.

4.2 External Interface Requirements

Incorporating these external interface requirements, the application can provide a smooth and secure experience for users, whether they choose to create an account through the website or log in using a Google account via the API.

Registration Interface:

- User-friendly registration form with fields such as username, email address, password, etc.
- Validation for input fields to ensure data integrity (e.g., email format validation, password strength requirements).

Login Interface:

- Option to log in using a registered email and password combination.
- Option to log in using a Google account via Google API authentication.

- Error handling for incorrect login attempts.

Google API Integration:

- Integration of Google Sign-In and Google Calendar API for seamless authentication using Google accounts.
- Handling of AllAuth tokens for secure communication between the application and Google services.

User Profile Management:

- Interface for users to update their profile information (e.g., username, email, password).

Session Management:

- Secure session management to keep users authenticated throughout their browsing session.
- Implementation of cookies or tokens for session persistence.

Security Measures:

- Encryption of sensitive data (e.g., passwords) during transmission and storage.
- Protection against common security threats such as CSRF (Cross-Site Request Forgery) and XSS (Cross-Site Scripting).
- Implementation of HTTPS protocol for secure communication.

User Feedback and Support:

- Feedback mechanism for users to report issues or provide suggestions.
- Support interface with FAQs, documentation, or a help desk for user assistance.

Terms of Service and Privacy Policy:

- Clear presentation of terms of service and privacy policy for users to review and agree upon during registration.

Logging and Analytics:

- Logging mechanism to record user activities and system events for troubleshooting and analysis.
- Integration with analytics tools for monitoring user engagement and system performance.

4.3 Logical Database Requirements

The application can effectively manage user authentication and journaling functionalities while maintaining data integrity and relational consistency by what is outlined below:

User Database Requirements:

- **User Table:** Contains information about registered users.
    - Fields: User ID (primary key), Username, Email, Password (hashed), Registration Date, Last Login Date, etc.
- **Session Table:** Stores active user sessions.
    - Fields: Session ID (primary key), User ID (foreign key), Session Token, Expiry Timestamp, etc.
- **Profile Table:** Optional table to store additional user profile information.
    - Fields: User ID (foreign key), Full Name, Profile Picture URL, Bio, etc.\

Journal Entry Database Requirements:

- **Entry Table:** Stores journal entries created by users.
    - Fields: Entry ID (primary key), User ID (foreign key), Title, Content, Creation Date, Last Modified Date, etc.
- **Category Table:** Optional table to categorize journal entries.
    - Fields: Category ID (primary key), Category Name, Description, etc.
- **Tag Table:** Optional table to tag journal entries for organization.
    - Fields: Tag ID (primary key), Tag Name, Description, etc.
- **Entry-Category Mapping Table:** If implementing a many-to-many relationship between entries and categories.
    - Fields: Entry ID (foreign key), Category ID (foreign key).
- **Entry-Tag Mapping Table:** If implementing a many-to-many relationship between entries and tags.
    - Fields: Entry ID (foreign key), Tag ID (foreign key).

Database Relationships:

- **User to Session Relationship:** One-to-Many relationship between users and sessions (one user can have multiple active sessions).
- **User to Entry Relationship:** One-to-Many relationship between users and journal entries (one user can create multiple entries).
- **Entry to Category Relationship:** Many-to-Many relationship between entries and categories (an entry can belong to multiple categories, and a category can have multiple entries).
- **Entry to Tag Relationship:** Many-to-Many relationship between entries and tags (an entry can have multiple tags, and a tag can be associated with multiple entries).

Data Integrity Constraints:

- Enforce foreign key constraints to maintain referential integrity between related tables.
- Use unique constraints/indexes to ensure uniqueness of certain fields (e.g., usernames, email addresses).
- Implement cascading actions (e.g., cascade delete) for appropriate relationships to maintain data consistency.

4.4 Design Constraints

Outlining imposed or self imposed constraints of the application design

Technical constraints: Programming Languages and Frameworks:

- Languages: HTML/CSS, Python, Javascript, SQL
- Frameworks and Management Systems: Google Cloud, Docker, Railway, Bootstrap, Django
- Development Requirements:
  - Coverage (version 7.4.0),
  - Django (Version 4.2.9),
  - Gunicorn (Version 21.2.0),
  - Psycopg2-binary (Version 2.9.9),
  - Pytest-Django (Version 4.7.0),
  - Pytest (Version 7.4.4),
  - Wait-for-it (Version 2.2.2),
  - Django-Allauth (Version 0.61.1)

Financial constraints: The application is designed around free services or promotional access for all design aspects. Design will remain limited to the free or promotional services available to use for each developer.

Technical Stack Constraints: Using the Django backend framework requires a few constraints and proper linking of static files including: HTML, IMG, CSS, and some Javascript files. All other dependencies should be free to download and easily to moderately accessible with free licenses.

Organizational constraints: The application is developed within the scope of a shared repository on GitHub. Developers access the application and its development through Github

Projects. This various project views (Roadmap, Product Backlog, etc) inform the development and timelines of the application.

Performance constraints:

- Database: Database constraints (efficient database storage and retrieval) will be accessed during testing, but no current constraints exist at the moment
- Google Cloud Web Hosting: The application is dependent on the Google Cloud hosting platform which may include considerations related to website speed, load times, and server performance. These factors will impact the user experience.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

No user terminals need to be supported for this application.

The number of simultaneous users to be supported:

- Initial User Base: 25 - 50 initial simultaneous users.
- Growth Potential
- Infrastructure scalability

Amount and type of information to be handled:

- User Authentication Data
- Journal Entries
- Meditation Resources
- Weather Data
- User Settings

Dynamic numerical requirements imposed on the entire system (may include):

- Transaction Processing Time
- Task Completion Time
- Data Processing Speed
- Concurrency Limits
- Data Storage and Retrieval Time
- Error Handling

5.2 - Safety Requirements

Data Privacy and Security: Ensure that user data, including journal entries and personal information, is securely stored, transmitted, and processed to prevent unauthorized access.

User Authentication and Authorization: Implement secure authentication mechanisms to verify user identities and restrict access to sensitive features and data.

Backup and Disaster Recovery: Implement regular data backup procedures and disaster recovery plans to prevent data loss in case of system failures, cyber attacks, or natural disasters. Ensure that backups are securely stored and easily retrievable.

Error Handling and Logging: Implement robust error handling mechanisms to detect and respond to system failures, errors, and exceptions gracefully. Log error details for troubleshooting and auditing purposes.

User Feedback and Reporting: Provide mechanisms for users to report safety-related issues, such as security vulnerabilities or inappropriate content. Establish procedures for handling and addressing user feedback promptly and effectively.

Performance and Scalability: Ensure that the application can handle expected user load and traffic without experiencing performance degradation or service interruptions. Monitor system performance metrics and scale resources as needed to maintain optimal performance.

Accessibility: Ensure that the application is accessible to users with disabilities, complying with accessibility standards such as WCAG (Web Content Accessibility Guidelines). Provide alternative means of access for users who may have difficulty using standard interfaces.

Compliance with Legal and Regulatory Requirements: Ensure compliance with relevant laws, regulations, and industry standards related to safety, privacy, security, and accessibility. Stay updated on changes to regulations and adapt the application accordingly.

5.3 Security Requirements

Data Encryption: Ensure that sensitive data, including user journal entries, passwords, and personal information, is encrypted both in transit and at rest to prevent unauthorized access or interception. Use industry-standard encryption algorithms and protocols.

Access Control: Implement role-based access control mechanisms to restrict access to sensitive features and data based on user roles and permissions. Users should only have access to the data and functionality necessary for their roles.

User Authentication: Require strong user authentication methods, such as multi-factor authentication (MFA) or biometric authentication, to verify user identities securely. Passwords should be stored securely using hashing and salting techniques.

Data Minimization: Collect and store only the minimum amount of data necessary for the application's functionality. Minimize the collection of personally identifiable information (PII) and sensitive data to reduce the risk of data breaches and privacy violations.

Data Retention and Deletion: Define policies and procedures for data retention and deletion to ensure that user data is not retained longer than necessary. Allow users to delete their account and associated data easily.

Security Auditing and Monitoring: Implement logging and monitoring mechanisms to track user activities, system events, and security incidents. Regularly review audit logs for suspicious activities and potential security threats.

Secure Communication: Use secure communication protocols such as HTTPS to encrypt data transmission between the web application and users' devices. Avoid transmitting sensitive data over unsecured channels.

Compliance with Regulations: Ensure compliance with relevant data protection regulations such as GDPR, CCPA, HIPAA (Health Insurance Portability and Accountability Act), or other industry-specific regulations. Adhere to privacy principles such as transparency, consent, and data subject rights.

Security Assessments and Penetration Testing: Conduct regular security assessments, vulnerability scans, and penetration tests to identify and remediate security vulnerabilities in the application. Engage third-party security experts if necessary.

Security and Privacy Certifications: Depending on the nature of the application and its target market, obtaining security and privacy certifications may be necessary. Examples include ISO 27001 certification for information security management or SOC 2 compliance for data privacy and security.

5.4 Software Quality Attributes

The application needs to complete the following quietly of life requirements where the application:

- Must adapt to the screen dynamic of each device (whether they use a mobile or computer/laptop device)
- Ensure it records all information in the Railway database to always display & be available to the user. The information retrieval process should be quick for the user to continue journaling or meditating
- Grammar check each page via some dictionary extension or other means
- Must not crash on random occasions when the user is consistently using the application

- It should be reliable in offering a smooth writing experience regardless of what E-Pen is used to write with
- The page dynamic of each section has cohesive animations to prevent the user from feeling annoyed by the interface
- The closing and opening are efficient and not slow

5.5 Business Rules

The following rules must be set to operate a safe and ethical application:

Ensure that users' journal entries are kept confidential and not shared with third parties without explicit consent.

Establish clear guidelines regarding the responsible use of data analytics to provide insights to users, ensuring transparency in how their data is utilized and empowering them with control over their information

Uphold integrity in the analysis process, avoiding any bias or manipulation of data that could compromise the trust users place in the application

All user journal entries must be securely stored and encrypted to ensure the confidentiality and privacy of user data

a.) Access to these entries should be restricted to authorized personnel only, and the user must explicitly approve any sharing or disclosure of user journal content

6. Legal and Ethical Considerations

Data Security and Privacy:

Legal Concerns: Complying with data protection laws

- GDPR Compliance

Issue: Collecting and analyzing user data, particularly emotional states can raise privacy concerns. Users should be aware of how or why their data is being used when writing down their thoughts and personal insights

Justification: Ensure clear communication on how their data will be used, and implement security measures to protect the data. Obtain consent before processing user data in any way, shape, or form. Users can opt in or opt out through our terms of service for their data they wish to remain private

Issue: Ensure that users have control over their data

Justification: Allow users to easily access journal entries, delete journal entries, as well as any other relevant data, and make the whole process intuitive and clear. Limit internal data access.

Content Influence:

Legal Concerns: Complying with consumer protection laws.

- State to State Legal Compliance
- CCPA in California, where the current platform is hosted and developed

Issue: External content or resources provided to users based on mood can manipulate or influence their emotions.

Justification: Allow users to have control over recommendations by customizing preferences and settings, and also making it clear that these recommendations may not accurately reflect the user's emotional state. Features will try to encourage positive behaviors and choices rather than exploiting vulnerabilities or stuff that can trigger negative emotions. Provide disclaimers that the content provided does not is not intended to be the only source of information users are expected to view or engage with, rather that the curated content is a small sample window of content that viewers can engage with no biases towards a viewpoint, practice, or advice.

Appendix A: Glossary

API: Application Programming Interface, a set of rules that allows different software applications to communicate with each other.

Bootstrap: Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development.

CCPA: California Consumer Privacy Act, a state statute intended to enhance privacy rights and consumer protection for residents of California, United States.

Django: Django is a high-level Python web framework that enables rapid development of secure and maintainable websites.

Encryption: The process of encoding information in such a way that only authorized parties can access it.

GDPR: General Data Protection Regulation, a regulation in EU law on data protection and privacy in the European Union (EU) and the European Economic Area (EEA).

Google Cloud Hosting: Google Cloud Hosting refers to the services provided by Google Cloud Platform (GCP) for hosting applications and websites on Google's infrastructure.

Gunicorn: Gunicorn is a Python WSGI HTTP Server for UNIX, used to run Python web applications.

HTTPS: Hypertext Transfer Protocol Secure, an extension of HTTP that is used for secure communication over a computer network, and is widely used on the Internet.

OpenWeather: OpenWeather is an application for servicing weather data that can be connected through an API.

OS: Operating System, a software that manages computer hardware resources and provides common services for computer programs.

PostgreSQL: PostgreSQL is a powerful, open-source object-relational database system.

Pytest: Pytest is a testing framework for Python that makes it easy to write simple tests.

Pytest-Django: Pytest-Django is a plugin for Pytest that provides a set of useful fixtures and marks for testing Django applications.

Psycopg2-binary: Psycopg is a PostgreSQL adapter for the Python programming language.

Railway: A database system designed to manage data related to railway operations, such as scheduling, ticketing, and maintenance.

UI: User Interface, the means by which a user interacts with a computer, website, or application.

URL: Uniform Resource Locator, a reference to a web resource that specifies its location on a computer network.

UX: User Experience, the overall experience of a person using a product such as a website or computer application.