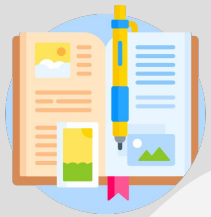


Insight Write

CS 3337-03 202301 Group #7

Team Members: Miguel Gonzalez Torres, Tony Lau, Hoang Le, Keyvan Mahmoodzadeh Kani, Will May, Hayk Vardapetyan, Jian Verdad, Niyusha Zarnegar.

Introduction

Team Member Names and Contributions:

- **Full Team Contributions:** Documentation, Software Design, Project Planning
- **Miguel Gonzalez Torres:** Backend Development(python)
- **Tony Lau:** Backend Development (Python) & Testing
- **Hoang Le:** Front-End Development (HTML/CSS/JS)
- **Keyvan Mahmoodzadeh Kani**: Backend (Python) Development & Railway Database
- **Will May:** Front-End Development (HTML/CSS/JS)
- **Hayk Vardapetyan:** Backend Google Login Integration
- **Jian Verdad**: Backend Development (Python) & Railway Database
- **Niyusha Zarnegar:** Database and Server Setup & Testing

- Link to application (if web based):
- Link to GitHub: Insight Write Repository

Miguel Gonzalez Torres

Our Vision!

Who is the Customer

- Individuals who are interested in personal growth and self-expression

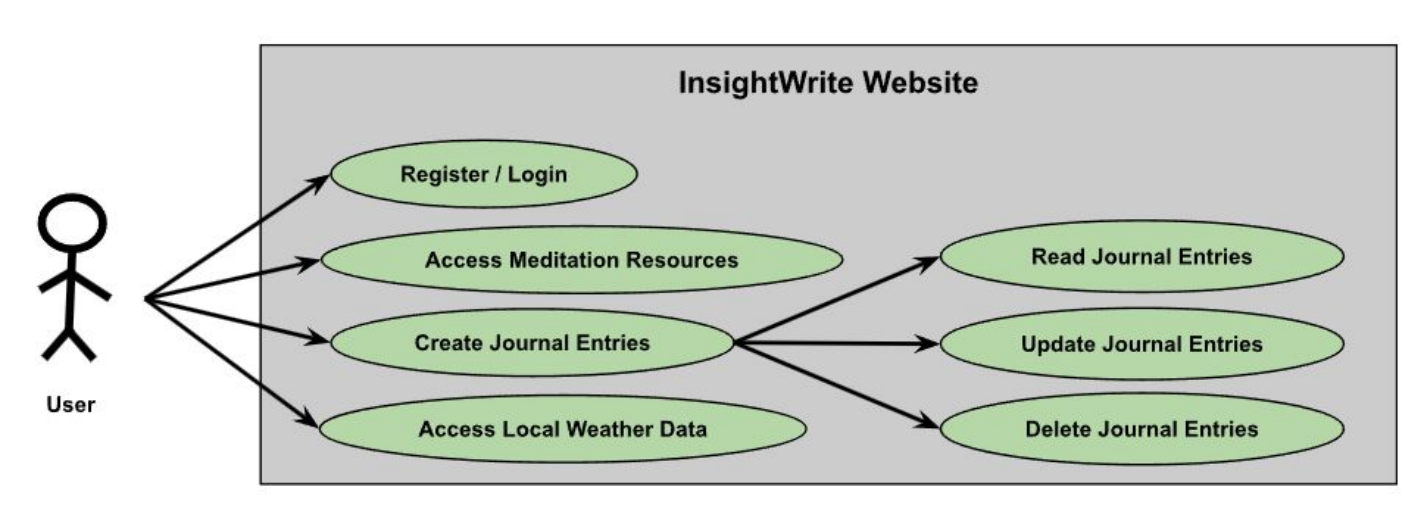
Problem

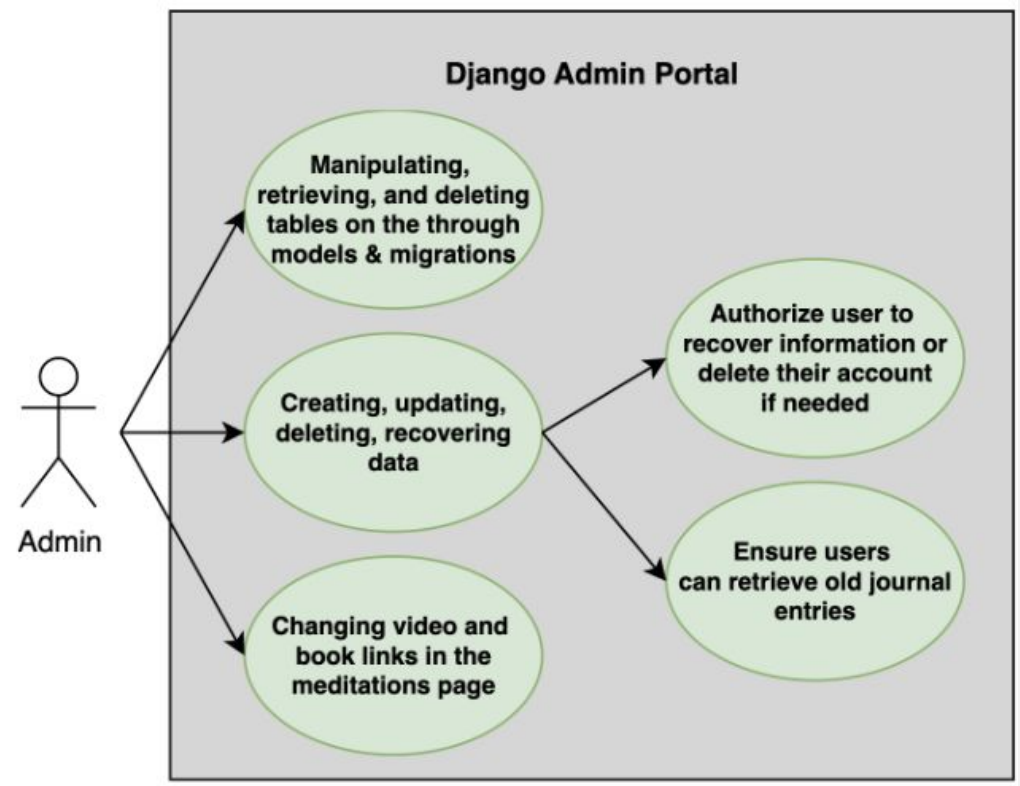
- Lack of existing applications which allows the user to freely express themselves
- Many users feel disconnect in their journaling journey cause of a lack of feedback

Digital Solution

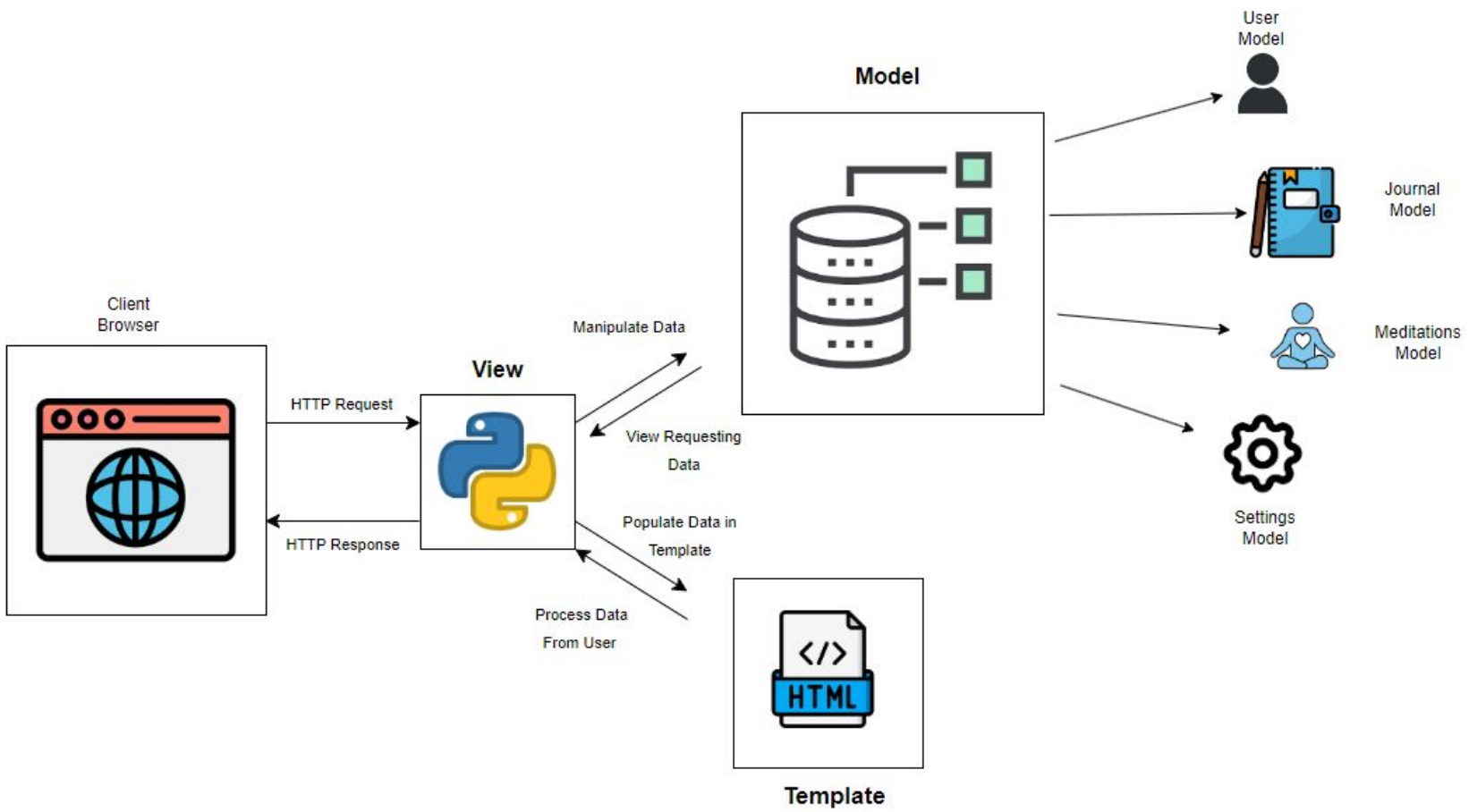
- Our application: A platform with a focus on allowing our user to express themselves through journaling.
  - Facilitates personal growth
  - Provides a private space to reflect
  - Improve mental health

Application Requirements





Architectural Design



Non functional requirements that favor this architecture for our web application include extensive security measures in place by MVT / MVC architectures for authentication and encryption, as well as allowing an easy way to be compatible with other third party systems.

Technologies Used & Implemented

Hoang Le

Frontend:

- **JavaScript**, **HTML**, and **CSS** → Building user interface

Backend:

- **Python**
  - **Django Framework:** Provides structured web application development
  - **Settings, Models, Views, URLs:** Components to configure, interact with the database, handle logic, and map web requests

Database:

- **PostgreSQL** on **Railway** → Store and save users' accounts and journals data.

Website Hosting:

- **Google Cloud**

Authentication and Authorization:

Django Allauth → Login with Google

API Integration:

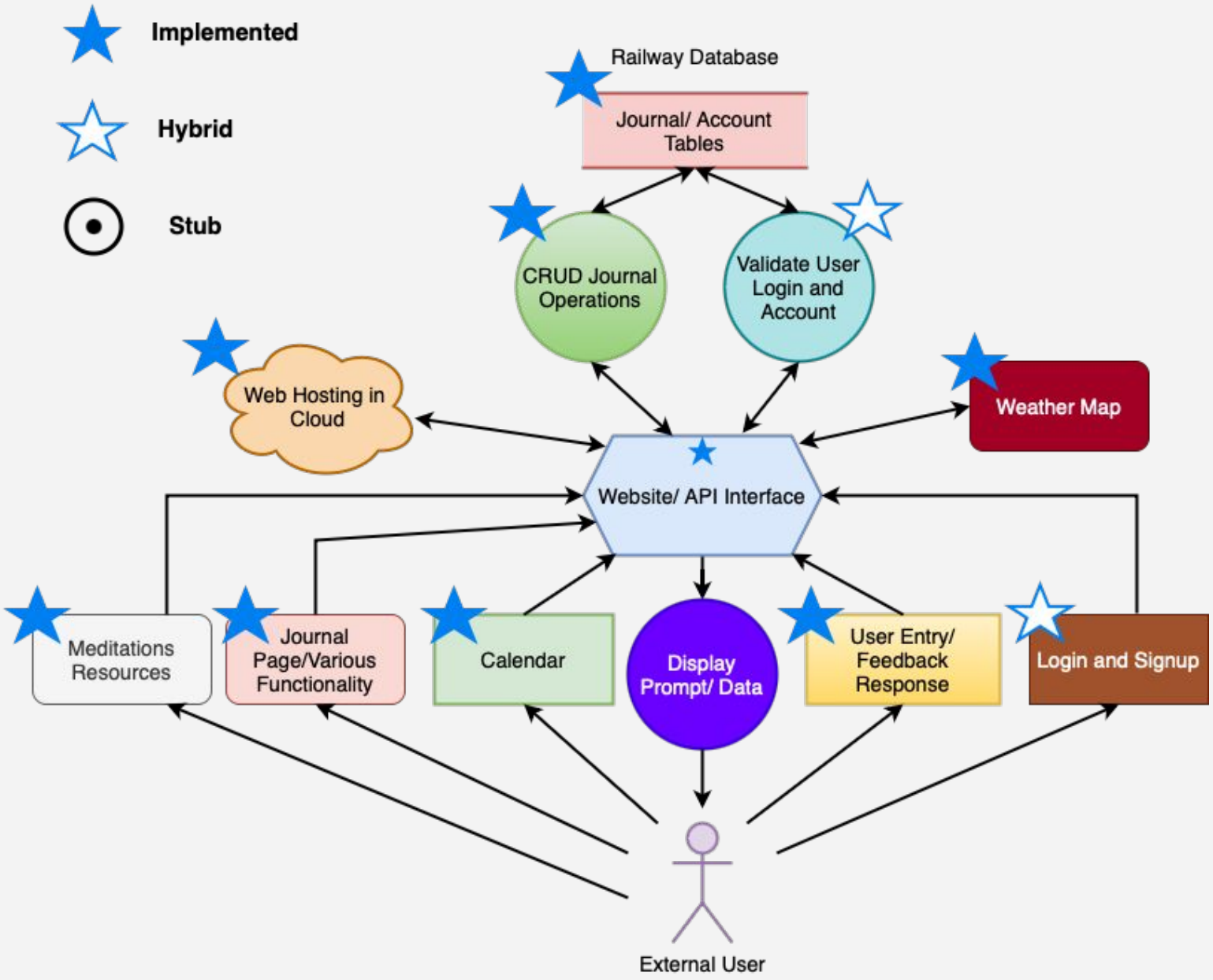
OpenWeatherMap API → Retrieve weather information

- **Third Party Libraries:**
  - Bootstrap → Navigation bar
  - Boxicons, Ionicons → Website Icons
  - Font Awesome → Animations
- **Resources:**
  - Pexels → Background images for our website

Processes - Detailed Design

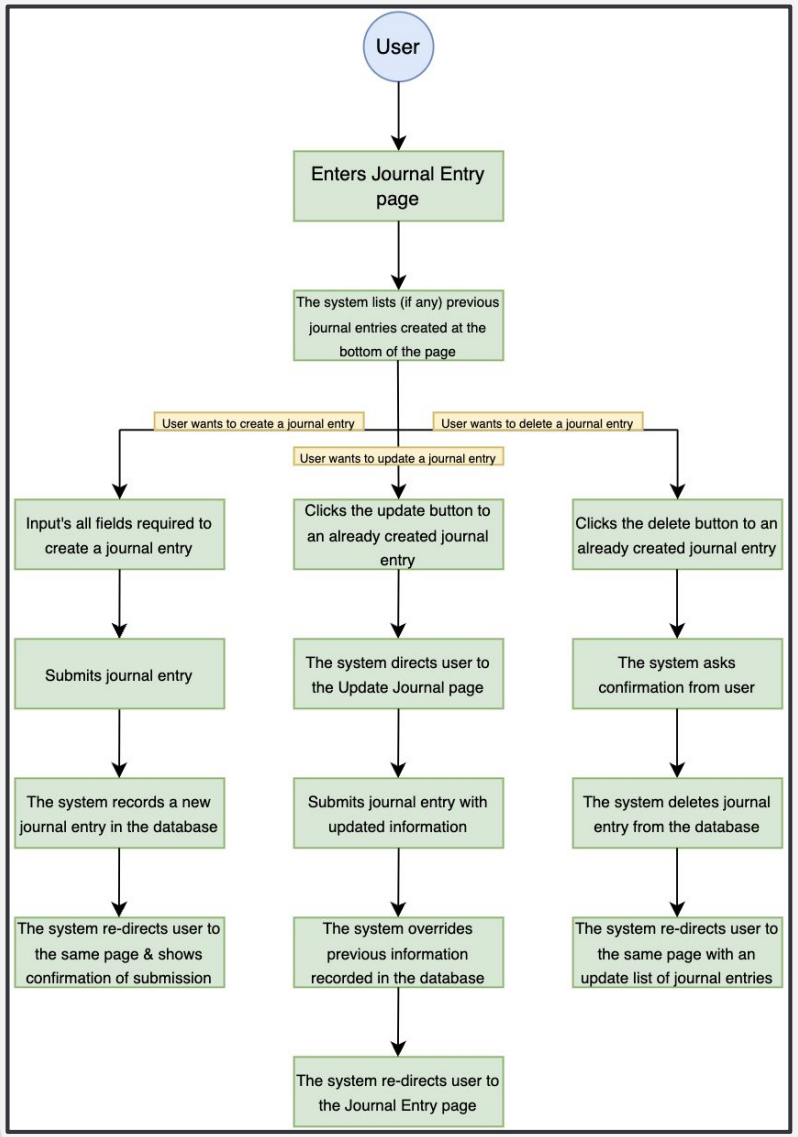
Will May

DFD Level 1 Diagram of Insight Write

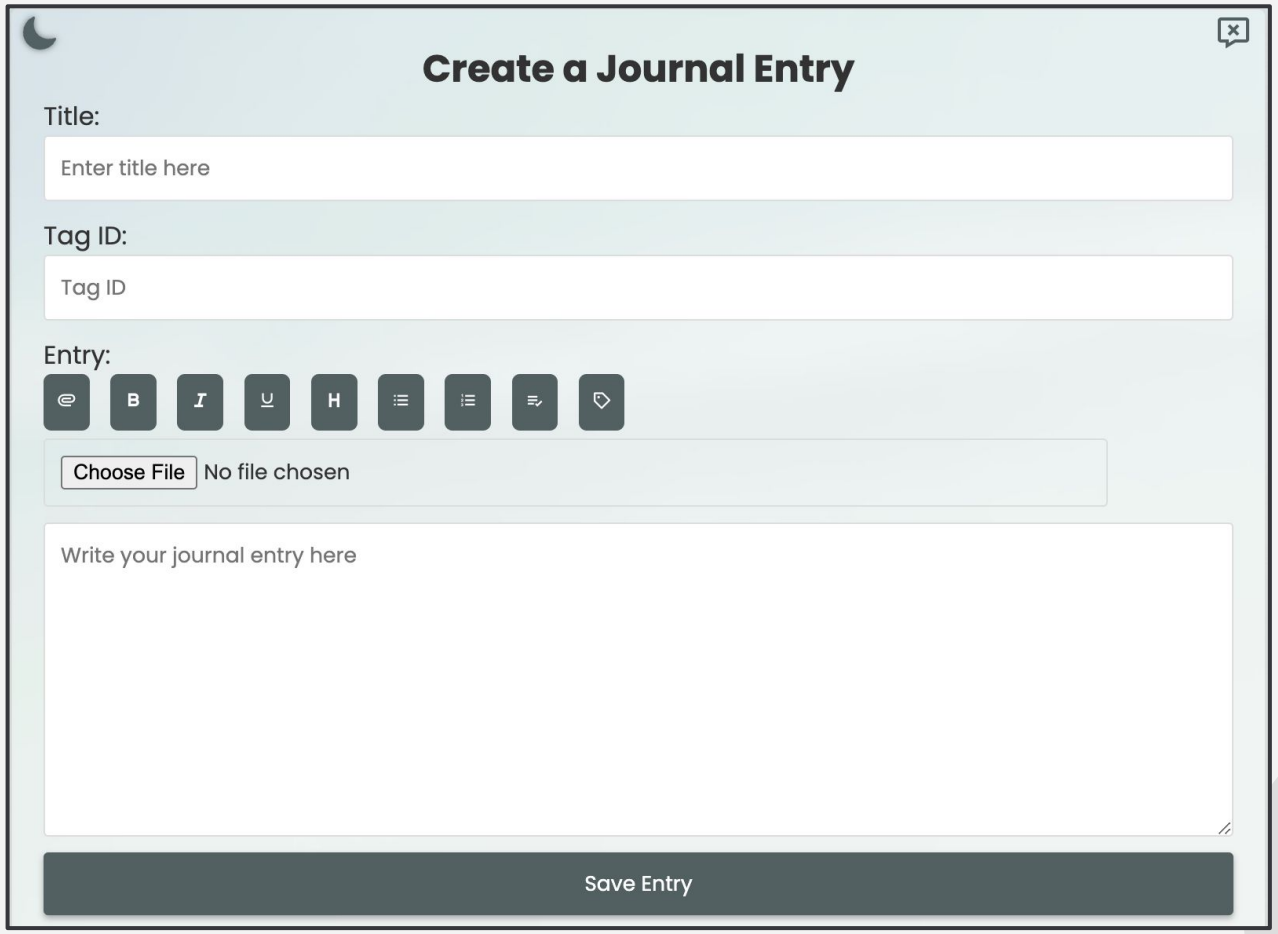


Keyvan M Kani

Behavioral Diagram of Journal Entry



Journal Entry Management:
Handles the creation, retrieval, update, and deletion (CRUD) operations for user journal entries.



Lessons Learned

- Docker Issues
- Agile Development was very helpful in communications
- How to better define features and stories
- Learned more about Railway, PostgreSQL, and Google Cloud
- Organizing the repository
- Locally hosting webpages
- The intricacies of how frontend and backend interact
- Integrating APIs
- Whether the data requires a new database or should be put in an existing one

Tony Lau

Column Roadmap (24.0, 24.1, 24.2, Future Enhancements)

24.0	24.1	24.2	Future
. Selection of Website Hosting - Platform: Github Pages . UX/UI Design . Change Management Plan . Unit Testing and Test Plan - PyTest . Frameworks and Tech Stack: - HTML/CSS/JS - Bootstrap - Django Web Framework - MongoDB API - NLTK API	Software Test Plan (STP) Software Requirements Document (SRD) Software Design Document (SDD) Journaling Webpage Login Webpage Setting Web Page Design Database Design	Framework Changes - Django Web Framework to Railway - PostgreSQL Databases - Google Cloud Mostly done with Journal webpage Finalized Signup/Login webpage Dashboard Webpage Meditation Webpage Database Implementation Calendar and adding events Soft API Integration: - Django allauth - OpenWeatherMap.org Journal CRUD operations - Designs, fonts, colors, etc. - Mini-draggable calendar - Text-Formatting (Bold, Italics, Hyperlink, etc.) Basic Javascript on all webpages Basic Python operations with database	Finalize Journal webpage (ex: Tags) Finalize Calendar (ex: email notifications) Unique data verification Encryption/Security Email Verification Setting Web Page (including User Accessibility features) Autosave Data and Information Save images and gifs to database Setting Editing User Profile Page Recovering Trashed Journal Entries

Demo

Q & A