CART-351-2252-A
André Neder de Almeida | 40208953
Hugo Waterfall | 40245720
Owen Hill | 40272473

Project #3 – An Explorable Networked-Space

“Journal: Mood Analyzer”

In short, **Journal: Mood Analyzer** is a website that allows users to write and submit journal entries on whatever topics they wish, which will then be analyzed by an artificial intelligence to determine the sentimentality of each entry (from negative to positive). Additionally, the website features a collective display of all user-submitted entries, which are altered to reflect each of their sentimentality values.

As a group, we spent a lot of time brainstorming ideas for this project, and despite coming up with many proposals that seem crazy in retrospect, we decided to keep it as simple as possible, while making it visually appealing and interactive. Determined to work with data obtained from strings, we set out to develop a system in which users would type an entry and receive data stored in MongoDB, which would also be manipulated by an API.

The journal idea was simple, but after revisiting ML5, an AI library for P5 JavaScript, we decided to take this one step further by creating a system in which the user would input a journal entry and have the mood of that entry analyzed to determine how positive it is on a scale from 0% to 100%.

From there, we expanded quite a bit, but the main focus of the program was allowing users to submit and read entries. In the beginning, we used our previous projects as a base, but eventually decided to build up from a new template, where we split the website into three core pages: Home, Append and Collection.

Home is the simplest of the three pages and functions as a landing page to give users a basic idea of the website. It was especially designed to be intriguing, drawing in potential users to the **Journal: Mood Analyzer** experience.

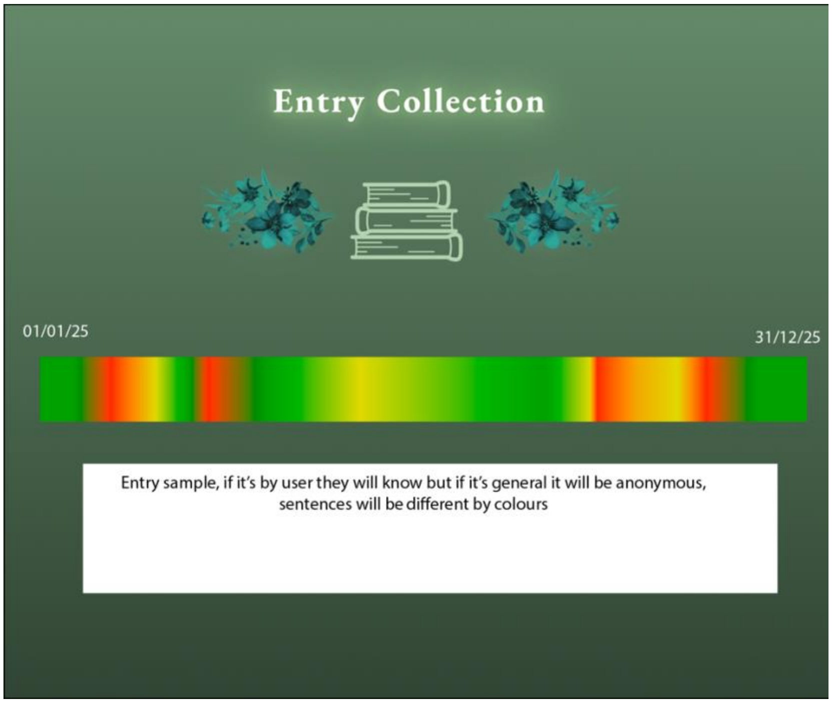
The Append page gives users a form to write and submit their entries. We might have overcomplicated our development by adding extra steps to the process, but it definitely paid off by enhancing the user experience. First, the user would come up with an inspiring message and click the “Begin” button to access the form. Then, they would type their name, their entry and click submit. After a brief pause for the website to load, the page will then display the positive percentage and an option to add another entry. All these elements are already present in the HTML and are triggered to display with JavaScript events.

When clicking submit, the JS and the ML5 model process the entry and assign values to the whole message, as well as assigning values to each sentence fragment (delimited by a comma or period). All this information is routed to Python, which stores the username, an array of sentiment values, and the date and time of submission. Upon submission, Python also stores the username in a cookie, which allows the site to remember their name in case the user wants to enter another message. While we initially planned on having more variables to display different things once a message was submitted, we diverted our efforts to the display of entries in the Collection page.

Since the very beginning, we knew that Collection was going to be the page where users could read past entries, but we had no idea how to display them. While this might have been the most challenging part of the project to tackle, the final product ended up being a pleasant and concise way of visualizing user entries as a collective. To achieve this, we implemented two routes in this page’s display: one through which users can view *all* the entries, and another to view entries from a particular username. The latter option supplies users with a form to enter a username, which is used in filtering entries. Both options will have a similar display: a single-row table that functions as a timeline, with equally distributed cells representing individual entries - each coloured depending on that entry’s positivity rating. Above the table are the dates of the first and last entries, from left to right, and at the center is the date of whichever entry the user is currently hovering over.

If the user clicks on an entry, the contents are displayed, accompanied by the positivity level. To further highlight this positivity value, the entry’s textbox changes depending on the mood. Not only does the colour of the inner shadow change to reflect the sentiment, but the text is filled with different fonts to represent the particular emotion of each sentence fragment. The colour is mapped from red to yellow and green, where red indicates 0% positivity, yellow indicates 50%, and green indicates 100%. This data is retrieved with Python and routed with IDs to JavaScript, where it is processed and parsed.

Our main struggles with developing this page had less to do with figuring out what needed to be implemented, but more to do with each of our busy, conflicting schedules, which was made even more complicated by end-of-term deadlines. We were initially forced to set some ideas aside to develop the cleanest, most polished version of our project, then went back and implemented whatever we could to give the Collection page the nice charm it has now.


Entry Collection
01/01/25
31/12/25
Entry sample, if it's by user they will know but if it's general it will be anonymous, sentences will be different by colours

Early sketch of Collection using Adobe XD

On the surface, the scope of this project seemed quite large and felt almost unattainable due to all the responsibilities that a student can have at the end of a semester. That being said, each of us was able to coordinate effectively and work in our own time to accomplish our tasks. And, after each commit on GitHub, the project looked better and better. It was hard working with shared files because of the fear that we would override each other’s changes, but we got around that by frequently duplicating and merging files together to preserve each of our work.

ML5 should have been easy to implement, and a great way to measure the sentiment of texts, but the model we used showed its age. A lack of recent updates to its system made the model’s results unreliable at times, producing some numbers with excessive decimals, or even extremely positive or negative percentages when the text was clearly showing the opposite.

Overall, this project represents a balance between experimentation and restraint, combining AI-driven analysis with a clear focus on user interaction and data visualization. Despite technical limitations, such as ML5’s occasional unreliable sentiment readings, and the logistical challenges of coordinating work at the end of the semester, we were able to adapt and refine our approach. Through iteration, collaboration, and careful integration of our individual contributions, the project evolved into a cohesive system that successfully connects user input, machine learning, and visual feedback, while also reflecting the realities and compromises of working on a complex group project.

Group work coordination

During our first brainstorming session, we delegated work and deliberated on which libraries we could use to enhance the project. In fact, one of the strongest aspects of this project was the way we delegated and coordinated work that played to each member's technical strengths, as well as taking measures to accommodate each of our schedules.

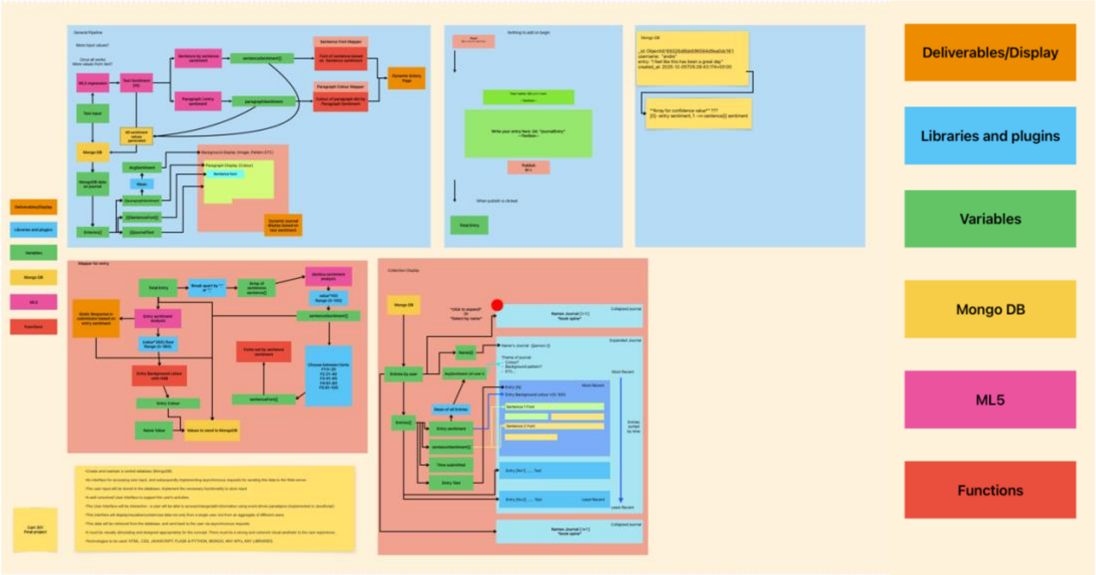
Hugo was a massive help, taking on the early stages of the website and providing the crucial framework upon which the project was built. While the planning for the project was a group effort, Hugo helped by organizing the project files and laying out each of the different pages of the website. He designed the UI, creating a visually cohesive colour scheme and design. He carefully planned out the big picture elements, such as customizing the navigation bar and modifying all the images, such that they not only blended well with the design but were also small enough to prevent loading lag. Special attention was given to the smaller details as well, including a specialized scroll bar and browser tab logo. His efforts were focused on creating a distinct, polished website, well-suited to handle the contributions of both Owen and André, with the goal of tying the project together through a common aesthetic.

André helped with setting up the backend to the website, the MongoDB communication, and the JavaScript for the display. Throughout the project, he was crucial in ensuring communication between members to make sure people were working to their strengths and to avoid redundant work. André also worked on the Collection page towards the end to ensure that the project was presentable. Despite of all the planning there was no clear visual reference for what the page was going to look like, other than how the colours were going to be mapped. So, with a screenshot of the empty in Adobe XD, a sketch was quickly made as a guide to be replicated in the front-end and even then, a few more visual elements were added such as the inner shadow for the text box. The first thing to be done in the front-end was receiving data from MongoDB and converting it to a 0 to 100 scale, rounding it and then mapping it from 0 to 50 and 50 to 100 for the colour scheme. Other than that, it was mainly the job of doing changes in the JavaScript to ensure that only certain elements were being displayed.

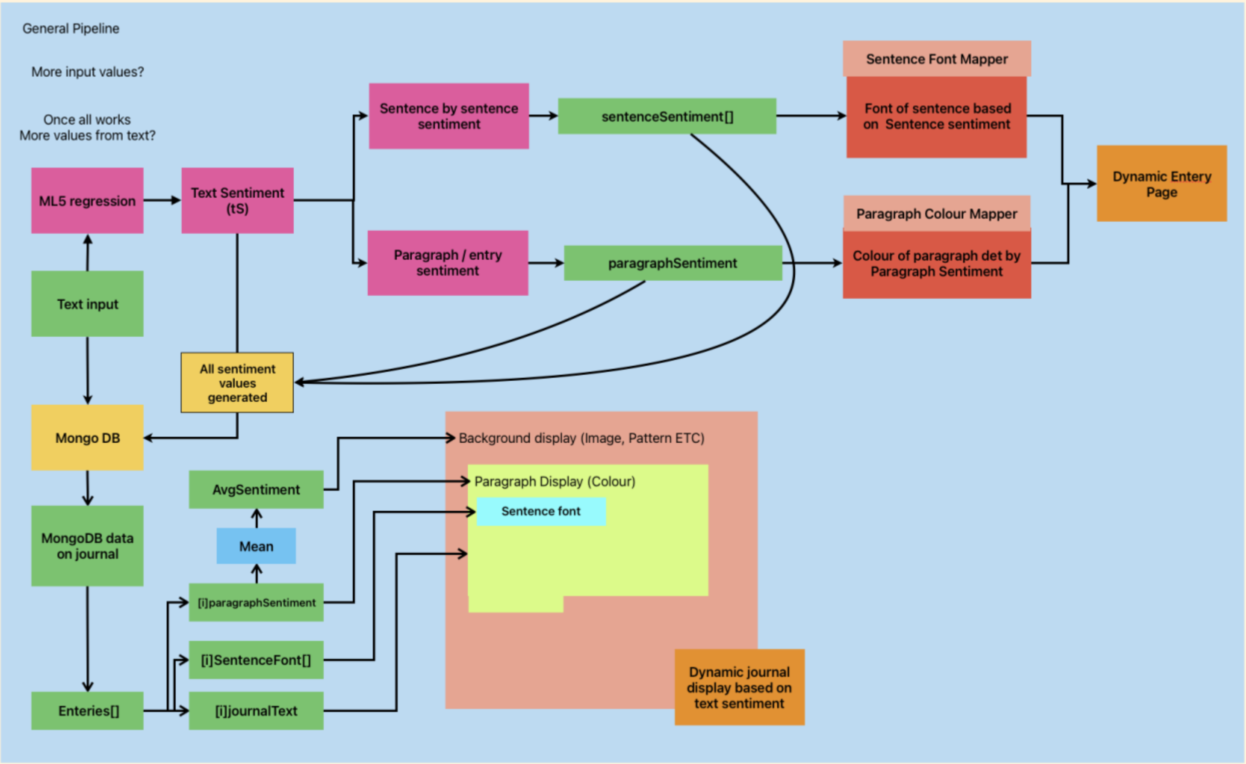
Owen worked on setting up the ML5 integration as well as handling the sentiment values and how they would manipulate the various entries. He designed the various helper functions used in the display that would allow for various things to be changed dynamically, with their modular nature assisting in the ease of implementation. This also allows for these modules to be used by group members in future projects. Most of the planning for Owen’s part was done in Apple Freeform, outlined in the planning flowchart walkthrough section below.

Planning Flowchart Walkthrough

Aswell some of the planning was done in a flowchart:

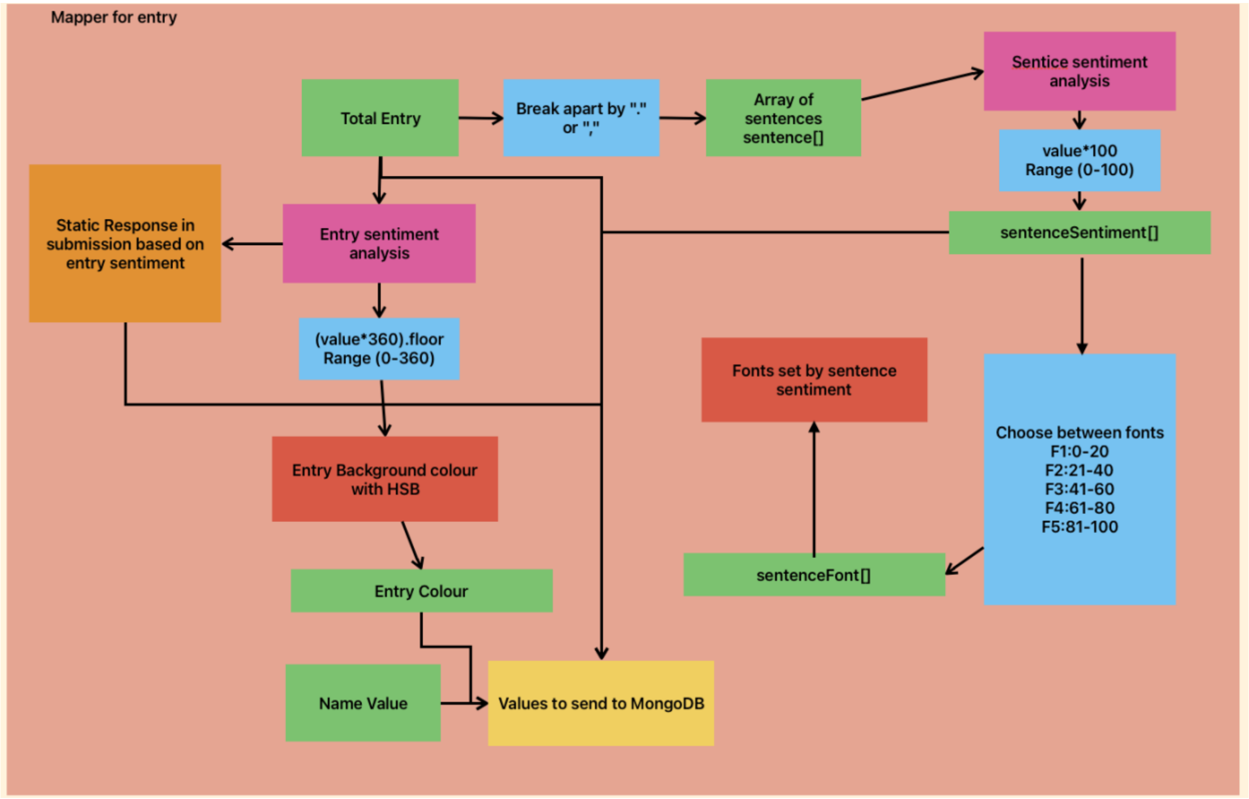


Segment 1) Rough pipeline:



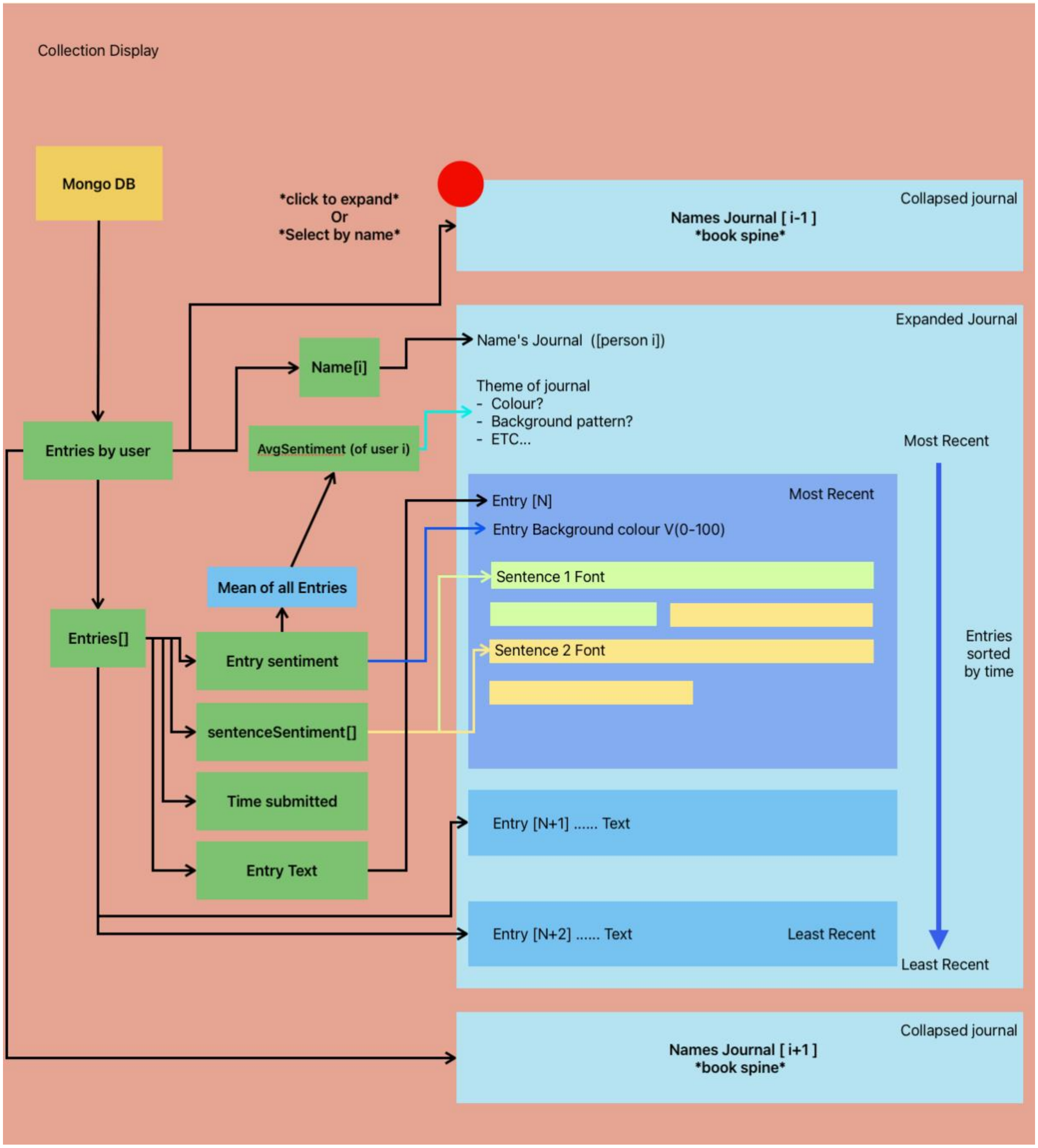
This was made to starting to plan out what would be done with the sentiment values, and to plan what to do with displaying the entries, although the final display changed, this still was useful to plan how different helper functions would be implemented

Segment 2) Entry mapper:


Mapper for entry
Total Entry
Break apart by "." or ","
Array of sentences sentence[]
Sentice sentiment analysis
value*100 Range (0-100)
sentenceSentiment[]
Static Response in submission based on entry sentiment
Entry sentiment analysis
(value*360).floor Range (0-360)
Fonts set by sentence sentiment
Choose between fonts F1:0-20 F2:21-40 F3:41-60 F4:61-80 F5:81-100
Entry Background colour with HSB
sentenceFont[]
Entry Colour
Name Value
Values to send to MongoDB

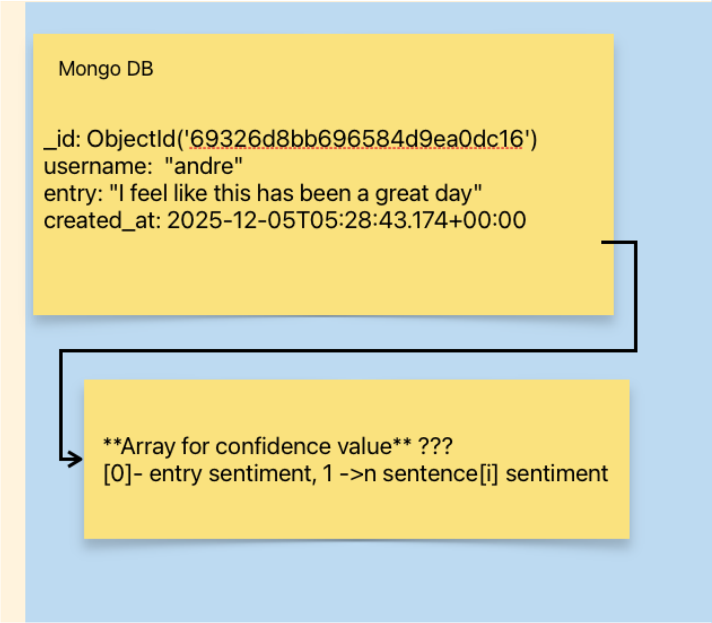
This was made to figure out what values would be sent to mongo DB and how they would be found. These functions would be implemented as helper functions through a JS object and can be used in a modular way with the entries display.

Segment 3) Collection display:



Again, though not the final design for the collective display as the total collection is selected through a dynamic bar coloured by the sentiment of the entry, this design was used for the colour and font of each entry. This flow chart also describes several ways that the website could be further developed to further utilize the sentiment values.

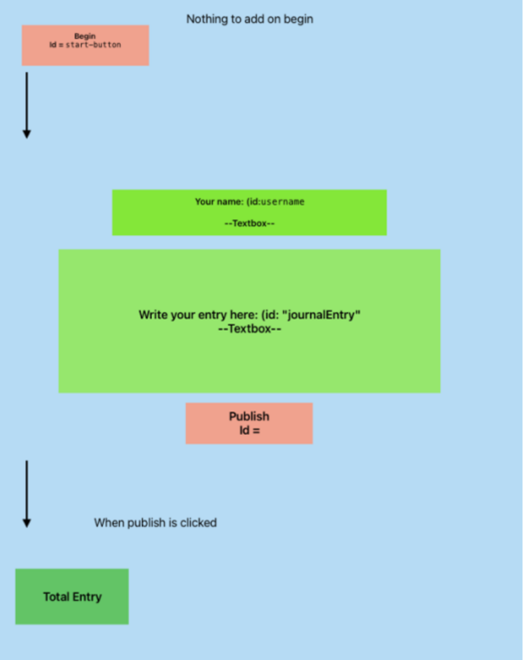
Segment 4/5) Planning for inputs



Mongo DB

```
_id: ObjectId('69326d8bb696584d9ea0dc16')
username:  "andre"
entry: "I feel like this has been a great day"
created_at: 2025-12-05T05:28:43.174+00:00
```

Array for confidence value ???
[0]- entry sentiment, 1 ->n sentence[i] sentiment



These smaller sections were made to help keep track of different aspects of both data and the HTML pages. These didn’t get fully fleshed out but still helped in planning

Most of our planning was mostly done with google docs for live sharing:

https://docs.google.com/document/d/12XXb9qSrXZeEwKFECrPOhi8s_V4sHmYJYxbxjEUe1_Q/edit?usp=sharing