Anxiety Knot

by

DiCarlo, Hornung, Kokenge, Lu

Submitted to
the Faculty of the School of Information Technology
in Partial Fulfillment of the Requirements for
the Degree of Bachelor of Science
in Information Technology/Cybersecurity

Gabriel DiCarlo	4/24/2022 Date
Jacob Hornung	4/24/2022 Date
Nicholas Kokenge	4/24/2022 Date
Oliver Lu	4/24/2022 Date
Yahya Gilany, Faculty Advisor	4/24/2022 Date

University of Cincinnati
College of
Education, Criminal Justice, and Human Services

April 2022

Table of Contents

List of Illustrations

Figures:

Tables:

Abstract

Anxiety Knot's primary purpose is to address rising levels of anxiety present in each new generation. Our digital journaling application facilitates users to address anxiety issues through Cognitive Behavioral Therapy techniques. This method includes helping users understand their negative thinking patterns, identify inconvenient situations more clearly and respond in an effective manner. This process involves the implementation of notable features such as mood input via icons and ratings, mood history of user and journal entries to assist user reflection on their current and past feelings. Prompted entries are introduced to ask users various questions to assist in reflecting on a current or past event during the entry. If users desire to input a quick entry, without question prompts, freeform journals provide the option to write quickly and freely enter thoughts and feelings into a single text area.

Introduction

Project Summary:

Anxiety Knot is an anti-anxiety journaling web application that provides users a self-guided journey, to better assist in understanding their personal thoughts and feelings relating to anxiety. Additionally, this will challenge their point of views in relation to their anxieties to help overcome and manage those worries. Users will metaphorically untangle their anxieties - hence Anxiety Knot- and given tips and advice to overcome their difficulties after a set number of journal entries.

Problem Statement:

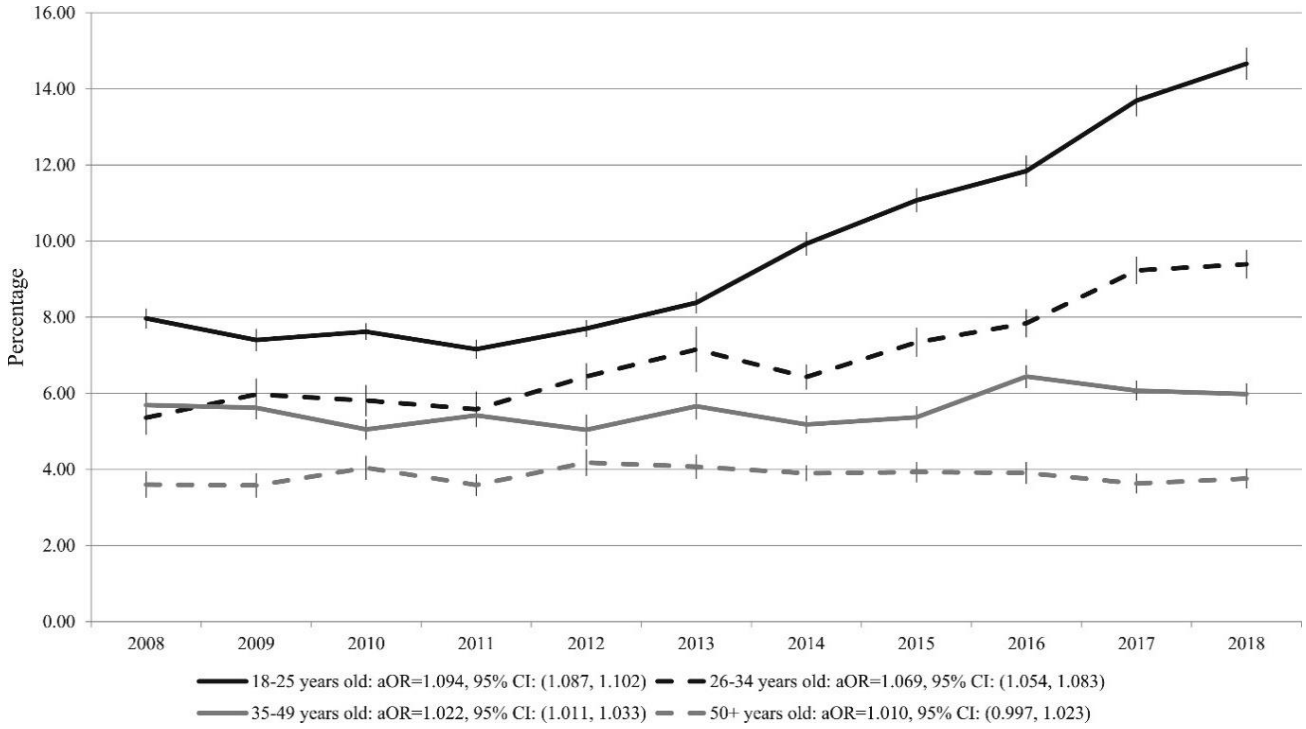
With each passing generation, individuals begin to feel less in control of their mental health, increasing the unlikelihood of openly discussing any psychological dilemmas. Yet inversely, younger age groups yearn to learn more about their own mental health. However, they find it difficult to improve their mental health and do not know where to find the help or information on doing so. This problem has significantly increased with the ongoing Covid pandemic, with over half of adults surveyed by the University of California San Francisco showing re-entry anxiety; feeling isolated, fearful of succumbing to the Covid virus and struggling to have a similar life to that of which they had before the pandemic.

Before the pandemic, anxiety has been a growing issue throughout the past decade. According to “Trends in anxiety among adults in the United States, 2008–2018: Rapid increases among young adults” published by the *Journal of Psychiatric Research* and shown in figure 1, anxiety has nearly doubled in 18-25 years old. Whereas ages 26- to 49-year-old respondents

witnessed a noticeable increase. The only age group that did not see an increase in anxiety were ages fifty and older, which remained stable.

Figure 1: This line graph presents a rapid increase of anxiety among young adults. Projections indicate that there is a growing trend of increased anxiety over the past years. The level of impact greatly differs depending on age group.

Figure 1: Trends in Anxiety

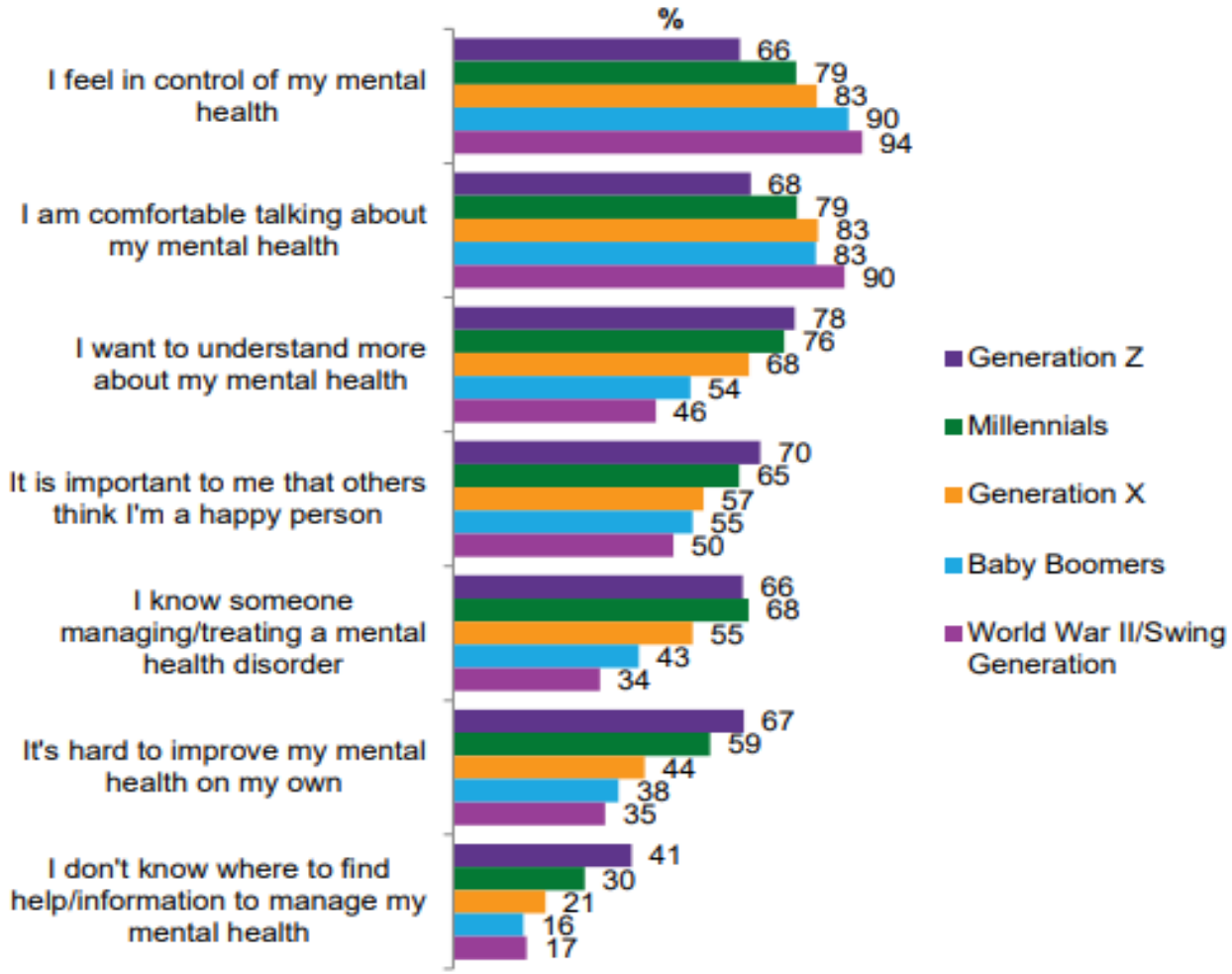


Due to the high rise in anxiety amongst those under age 50, another movement has been growing; those feeling unable to control their mental health desire to learn and manage their psychological wellbeing. As shown in Figure 2 below, only 66% of Generation Z feels in control of their mental health as opposed to 94% of the Swing Generation. Additionally, 78% of Generation Z wants to understand more about their mental health while only 46% of the Swing

Generation does. If not solved, this trend will only continue as it has for the past few decades as shown in the data below.

Figure 2: Bar graph measures in percentage the number of respondents (based on generation) that feel in control, discuss, and know where to reach out for mental health disorders.

Figure 2: Attitudes Toward Mental Health, By Generation, May 2020.



Solution:

Anxiety Knot is a journaling Web application that provides a self-guided journey to help the user cope with their stresses and anxieties. Users will log-in, able to keep an archive of written journal entries, which are guided by prompts inspired from an anti-anxiety journal based on cognitive behavioral therapy. A separate space will be provided to users that will allow them to have unprompted journaling. The user will be given a list of resources for professional help and advice.

To help monitor a user's mood patterns, this product implements two mood tracking tools. The first tool tracks how the user feels before and after journaling. The second tool displays a visual diagram showing how the user feels on a weekly basis; viewed over time via bar graph within in the application. To help promote the applications continual usage, users will receive in-app awards/medals displayed in the form of a knot icon being un-tied as they progress through journal entries. Additionally, to ensure users receive the knowledge and help needed, popup tips/advice will be provided to assist with their anxiety.

By offering users these tools and through the process of helping them identify their own cognitive distortions – they will be able to challenge their own thinking. As a result, they can understand more about their own anxiety. This will help with the trend we see across generations, where their mental health continues to get worse.

Project Source:

The inspiration for this project came from publish retail anti-anxiety journaling notebook which includes questions to utilize Cognitive Behavioral Therapy (CBT). This product influenced the team in a positive manner, as we decided to apply the journaling notebook tools toward family and friends, such as identifying cognitive distortions. Requirement analysis continues to be actively conducted by all team members to discuss ways to improve application primary goal.

Discussion

Project Objectives/Goals:

This section highlights the main objectives and goals for what Anxiety Knot will achieve for audiences and interactions will persist throughout the application. Our primary goal is for users to complete journal entries, obtain a better understanding of their own negative thinking patterns and self-reflect in how to break these patterns.

- After a set number of journal entries, the user will better identify and cope with their stresses and anxieties via journaling prompts.
- The user will learn to think differently by challenge their negative thoughts and emotions moving forward.
- Creation of an unprompted journaling feature allows users to freely include their thought patterns within a single text field, a freeform journal.
- Design of a prompted journaling feature allows users to answer a series of created questions. These prompt questions ask about their day, or an event that occurred, to assist identifying specific thoughts and moods experienced, and if the occurrence could have been approached differently.
- The app will include optional user notifications to encourage consistent practice.

- Users will be able to reflect on mental health growth or progress, by reviewing their past journal entries and posts.

- Compare how they are feeling before and after journal entries via a brief survey that asks them how they are feeling.

- Search and view past journal entries, as to recognize mood input patterns and other events within entries.

- All elements are measured by the in-app weekly mood tracker and feeling assessments before and after journal entries.

- Mood tracker information can be viewed via in-app graphs.

- Mood tracker will ask the user how they are feeling on a rating scale using star icons.

- Users reserve the right to delete or previously created entries; either out of painful events or mistakes made within their entry.

Project Scope:

Here we define our project's scope, which includes the features and functions of our application. This list also incorporates the range of effort required to finish the project.

- Allow users to journal their thoughts and unravel their anxieties and stresses by utilizing the in-app journal.

- Include a mood tracking component to monitor current mood inputs and display input history by use of daily and weekly mood prompts.

- Incorporate a timeline to suggest tips and advice to the user on how to deal with their mental health related issues. Tips display in conjunction with consistent use or measurable progress.

- Have an archive of previous journal entries to encourage self-reflection by creating a designated save area in the Web application.

- Challenge user thinking and responses by recognizing cognitive distortions using the in-app journaling prompts.

- In terms of scope, the mood tracking application is designed to help users understand and cope with anxieties; mood tracker works only when paired with journaling.

- The concept of this application works based on willingness to truthfully express thoughts and feelings.

- Regarding constraints, this application does not offer medical advice or direct recommendations. However, links to professional help sites are provided.

Quick Project Timeline:

As depicted in the project timeline, the first semester involves the processes of identifying and establishing Anxiety Knot's foundation; contract, chosen technologies, research, and application development. The final deliverable incorporates solidifying application development through modifying previous features and including innovative quality assessment testing enhancements to improve user experience and overall functionality.

Task #	Task Name	Duration	Start Date	End Date
1	Finished contract	4 Weeks	8/23/21	9/20/21
2	Finalize technologies we are using	1 Week	9/20/21	9/27/21
3	Setting up GitHub repo with project board	1 Week	9/27/21	10/4/21
4	Researching Angular and creating front-end project stub	1 Week	9/27/21	10/4/21
5	Researching Node and Express and creating back-end stub with MongoDB	1 Week	10/4/21	10/11/21
6	Linking front-end and back-end subscriptions and setup hosting	1 Week	10/4/21	10/11/21

7	Allow for unprompted journaling	2 Weeks	10/11/21	10/25/21
8	Include prompted/guided journal entries	2 Weeks	10/11/21	10/25/21
9	Ability to save and delete prompted/guided journal entries	1 Week	10/25/21	11/1/21
10	Ability to edit prompted/guided journal entries	1 Week	11/1/21	11/8/21
11	Tracking journal entries and setting up achievement system	2 Weeks	11/8/21	11/22/21
12	Tips and tricks provided to users	2 Weeks	11/22/21	12/6/21
13	Tracking user's feelings before and after journaling	1 Week	12/6/21	12/13/21
14	Weekly mood tracking for users	2 Weeks	12/6/21	12/20/21
15	Styling for user interface	2 Weeks	1/10/22	1/24/22
16	Enhance user experience by providing them with more insights on their journaling habits	2 Weeks	1/24/22	2/7/22
17	Enhance app accessibility on mobile devices	7 Weeks	2/7/22	4/2/22
18	Remove any unnecessary code and deploy to webservice for IT Expo.	1 Week	4/3/22	4/11/22

Technologies Used:

This segment summarizes technologies chosen to develop the Anxiety Knot Web application. We chose to use a typical MongoDB, Express, Angular, and Node (MEAN) for full-stack Web application development. Thanks to several group members prior familiarity of the Angular framework, simplifying our decision to use this platform. Additionally, the accessibility of detailed resources and other guidelines for this technology stack, was deemed extremely beneficial to the group as opposed to using a combination of other frameworks or libraries.

1. **Front-end:** Angular (HTML, CSS, and TypeScript)
    a. Team had more familiarity with Angular over alternatives (React and Vue).
    b. Unanimously agreed more tasks can be completed by using a larger framework versus using vanilla JavaScript.
2. **Back-end:** Node.js with Express.js framework
    a. There were a lot of resources for using Node.js and Express.js for a back-end when paired with Angular
3. **Database:** MongoDB with Mongoose
    a. MongoDB contains a friendly user interface, easy to work with when paired with Mongoose as opposed to MongoDB's in-house drivers.
    b. A downloadable client application provided easy access to view and modify database information.
4. **Hosting:** Amazon Web Services (AWS)
    a. AWS Simple Storage Service (S3) offers simplistic steps to seamlessly upload front-end to hosting environment.
    b. AWS Elastic beanstalk (EB) provides simple steps to deploy and scale Nodejs web application and services.

c. Plenty of other available resources to manage environments, with the benefit of providing non-paid services.

5. **UI/UX Design:** Figma

    a. Figma prove to be more user-friendly over alternatives like Adobe XD or InVision.

6. **Repository and Project Board:** GitHub

    a. Being among the most popular Git services, it was only natural to use this service.

    b. All team members are familiar with GitHub's services and functionalities.

7. **Source Code Editor:** Microsoft Visual Studio Code

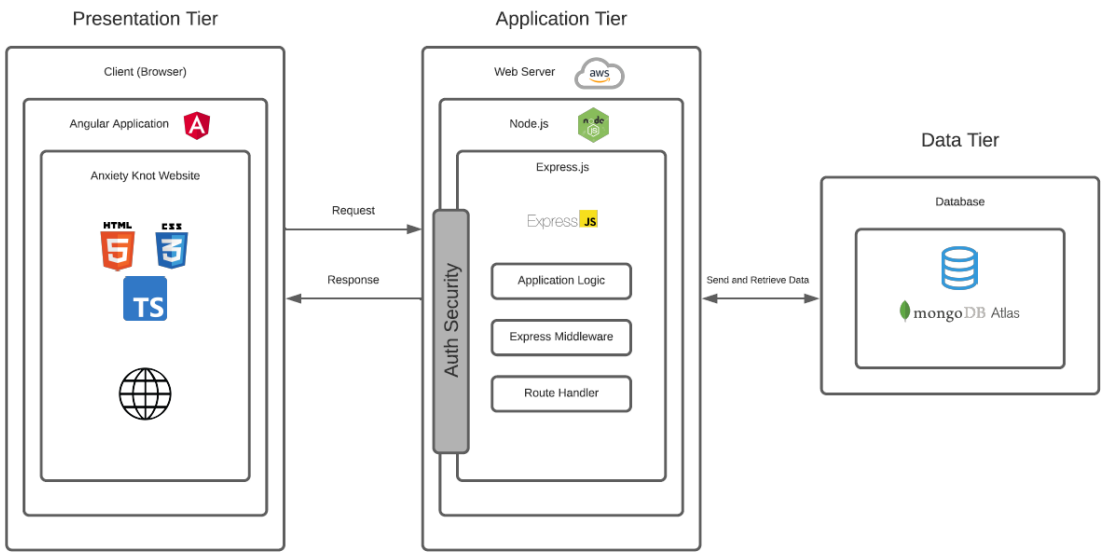
    a. Contains an extraordinary open user interface and universally accessible to numerous application plugins, enhancing the project's scope.

Technical Architecture Diagram:

Anxiety Knot's technical architecture consists of three tiers: presentation, application, and data. The presentation tier houses front-end client and browser interaction of the application. Matching with the presentation tier, the Angular framework manages HTTP requests, while displaying responses/results from our application tier. Producing our front-end web app, requires optimizing core Angular functions, HTML, CSS, and TypeScript. Our presentation tier communicates with the application tier, where the application's back end is hosted using AWS. The back-end consists of a Node.js server using Express.js to oversee the application logic, middleware, and route handling. Next, the application tier forwards requests and receives data between itself and the corresponding data tier. Lastly, the data tier is our logical database where we used Mongo DB Atlas.

Figure 3: Anxiety Knot's structure is set up into 3 parent tiers. *Presentation* represents all interaction through the front end. *Application* passes request and response information passing through *presentation* and *data*. *Data* tier logically stores all user created posts, which are fetched and viewable to the client (using created account).

Figure 3: Technical Architecture Illustration



User Personas:

This table summarizes multiple user personas for our target audience. We believe these users are the ideal types of individuals that will benefit the most from our application. As a team, we feel that our application will most aid people who follow similar personas as listed below:

Table 1: User Profile

User Persona: 1	
	Title: High School Student
	Name: Julie Roberts
	Age: 16
	Gender: Female
Behavior	- Goes to school and works a part time job - Technology is second nature; she grew up with smartphones - Trying to finish high school with an associate degree as well
Pain	- Prone to anxiety attacks - Pressure from school and social life can be overwhelming
Needs & Goals	- Methods for dealing with anxiety and pressure - Does not want to seek professional help - School counselor is not enough

User Persona: 2	
	Title: College Freshman
	Name: Alexander Smith
	Age: 19
	Gender: Male
Behavior	- First year at college hundreds of miles away from home - In a relationship with his high school girlfriend - Introverted and shy
Pain	- Hard to make friends during covid pandemic while being shy and introverted - Navigating college campus and classes are stressful - Girlfriend troubles with long distance

Needs & Goals	- Wants to create a social life for himself - Wants to maintain his high school relationship - Wants to do well in classes and succeed in academic life

User Persona: 3	
	Title: Recent College Graduate
	Name: Olivia Herst
	Age: 23
	Gender: Female
Behavior	- Recently graduated college and moved to a new city on the other side of the country - Just landed a new job as a social worker - Planning a wedding with her fiancé - Outgoing, extroverted, and motivated
Pain	- Social work clients project their problems onto Olivia - Living in a new city while juggling a brand-new job all while planning a wedding
Needs & Goals	- Wants to setup a good life for themselves in a new city with minimal stress - Looking for a way to document and cope with this journey

Use Cases:

Defined in this portion, we have our applications use cases. These are scenarios in which our application receives an external request, usually from a user, and responds to it. Below are the most common and crucial use cases to the flow of our application.

Table 2: Use Cases

Use Case ID	001
Use Case Name	Create Journal Entry
End Objective	Being able to have a saved journal entry and have a logged mood
User/Actor	Anxiety Knot user
Trigger	The user selects “Create Journal Entry”
Frequency of Use	100% of users will need to create journal entries frequently
Preconditions	User is logged into the app
Basic Flow	The user selects “Create Journal Entry” The user selects a mood icon for current mood The journal entry page is displayed with a journaling prompt at the top The user completes the journal and saves The user selects a mood icon for new mood The user is taken back to the home page
Alternate Flow	The user selects “Quick Journal” The journal entry page is displayed with no prompt The user can make a journal entry and save it The user is returned to the home page
Postconditions	The user has created and saved a journal entry

Use Case ID	002
Use Case Name	Find and view past journal entry
End Objective	View a specific journal entry saved from a previous journaling session
User/Actor	Anxiety Knot user
Trigger	The user selects “View Journal History”
Frequency of Use	All users will want to view previous journal entries, some will want to do this weekly
Preconditions	User is logged into the app and has multiple journal entries
Basic Flow	The user selects the “View Journal History Option”

	The user selects the month which they would like to view the history of The user begins to scroll through the timeline of journal entries The user finds the journal they are looking for and clicks on it That journal entry opens The user can read the entry and edit it if desired
Alternate Flow	The user selects the “View Journal History” option The user chooses the search icon The user searches for a title of or phrase within the entry The user can see a list of all Journal entries containing the search The user selects the desired entry The user can now read or edit the entry
Postconditions	The journal entry has been selected and the user can view and re-read it

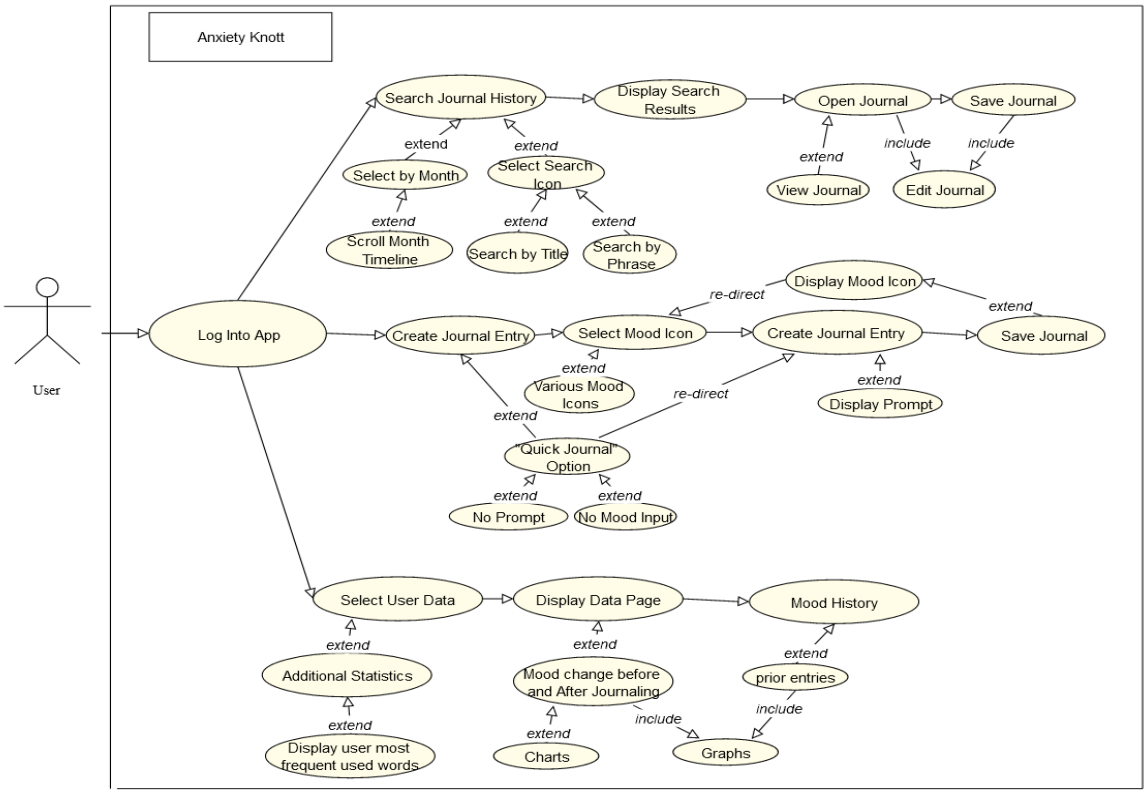
Use Case ID	003
Use Case Name	Mood statistics
End Objective	View historical data of mood before and after journaling and over time
User/Actor	Anxiety Knot user
Trigger	The user selects “User Data”
Frequency of Use	70% of users will want to periodically measure their progress using this data
Preconditions	User is logged into the app and has multiple journal and mood entries
Basic Flow	The user selects the “User Data” The data page is displayed The user can see graphs and charts showing their mood change before and after journaling The user can see a graph of their mood history over time The user can select a period to view data from
Alternate Flow	The user selects the “User Data” The user selects additional statistics The user is shown their most frequently used words
Postconditions	The user has been able to view and understand their data

Use Case Diagram:

The overarching design goal of Anxiety Knot is for users to login into their created account and perform a variety of tasks shown below. One of these tasks involves the ability to create one or various journal entries, while incorporating the option to display their current mind-set using mood emojis. Provide the capability to search, view and modify created journal entries. Offer a data section displaying both a chart and graph history of the user's previous moods gathered from their prior journal entries. Finally, because our application is not meant to substitute medical professionals, a resources webpage will contain links to credible mental health Web sites.

Figure 4: Once a user logins into the application, they will be given distinctive ways to search for journal history, create an entry, and access information in relation to their input history.

Figure 4: Use Case Illustration



Testing Plan:

Overview

In this section there is a breakdown of our testing strategy, and our end goal is to verify that the application is working properly. We will outline our approach to testing throughout the development cycle, a list of use features that we will be testing and why we chose them. Additionally, we will outline pass/fail conditions and results of our tests along with a change management plan.

Methodology

Our test approach to is to systematically break down each feature and component of our application and design a unit test to ensure that it works properly, using test driven development via the Jasmine test framework which the Angular Command Line Interface (CLI) downloads automatically when testing. To test the application at the end of each development cycle, this approach allows us to evaluate and integrate each feature developed. On top of these tests, we decided to add integration tests to perform after the unit testing, which ensures data transmission persists and distinct application components are working simultaneously. Lastly, user acceptance testing (UAT) is conducted as the final phase of our application testing process, ensuring components and features perform to user expectations. In doing so, we can confirm our application supports the experiences of our user demographic and deploy to our hosting production environment.

Scope

The chosen use cases and features encompass the core functionality of our application and are crucial to ensure our product functions correctly as designed. These use case scenarios follow the main flows a user will encounter while using the application. All passing tests signify our application is working as intended. Failed tests provide insight as to further adjust, focus on these areas, greater enhancing our application's integrity and improve user experience.

Use Cases/Features

1. Log Into App
2. Create Journal Entry
3. Select Mood Icon
4. Create Unprompted "Quick Journal"
5. Save Journal
6. Browse Journal History
7. Search Journal History
8. Edit Journals in Journal History
9. View User Frequent Word Usage
10. View User Mood History Data
11. View Mood Change Data Surrounding Journaling Sessions

Objectives

This section highlights each objective for our application that we plan to accomplish and to be acceptable by our designated completion date.

1. All major features and use cases need to be accounted for.
2. All bugs need to be resolved before the IT (Information Technology) Expo.
3. App should be able to manage bad data or requests without crashing.
4. Test results should be logged with meaningful data; used to improve the application.
5. Tests should be limited in scope to only the feature they are assessing.
6. When adding a new feature, old unit tests and dependencies should work properly before moving on.
7. All use cases must account for all the user roles.

Test Logs and Procedures

Item #	Test Case #	Expected Output	Actual Output	Pass/ Fail	Reason	Date
1	1.1	User logs in, output is homepage navigation route	User logs in, output is homepage navigation route	Pass	Included a routing path to send users to home page upon login.	2/5/22
2	2.1	User clicks 'create' and supplies mood, output is journal entry textbox with prompt	Mood is displayed after user creates journal entry	Fail	This is a feature we decided to change	2/5/22
3	3.1	When user clicks 'create' they are given choice to select mood, where output reflects mood choice	Mood is displayed after user creates journal entry	Fail Pass	This is a feature we decided to change	2/5/22 3/21/22

3	3.2	When user selects mood, output reflects updated database	Upon clicking mood, journal is placed into database	Pass	N/A	2/5/2022
4	4.1	When user clicks ‘quick journal’ output is journal entry with textbox	When user clicks “add journal entry” output is journal entry with textbox	Pass	Shown on home page for users to access after login.	2/5/2022
5	5.1	When user enters text in journal and clicks ‘save’, output should reflect updated database with entry	Output is reflected in updated database	Pass	Post and user info stored in schemas designed for update and storage within MongoDB	2/5/2022
5	5.2	When a user tries to save a journal with no text output expected to be an error	When a user tries to save a journal with no text output throws an error telling them to fill journal	Pass	We also have the ‘submit’ button set as inactive until the required form fields are set.	2/5/2022
5	5.3	Mood Icon displays on “Save Journal” page, clicking icon redirects user back to "Select Mood” page.	User selects mood from a five-star rating system	Pass	User is redirected to their Journal Book after mood selection submission or cancelation.	2/5/2022
6	6.1	When a user goes to ‘journal history’, output expected to be list of past journal entries that	When a user goes to ‘journal history’, output expected to be list of past journal entries that	Pass	‘journal history’ is now displayed within a	2/5/2022

		can be scrolled through	can be scrolled through		scrollable 'journal book' webpage.	
7	7.1	When a user goes to 'journal history' and clicks on search bar, output expected to focus on bar and allow user to search through previous entries in database	There is no search functionality	Fail	Feature yet to be added	2/5/2022
8	8.1	User should be able to click on past journal entries and edit them with 'edit' button, output should be that journal entry with its data retained	User able to click on past journal entries and edit them with 'edit' button, output is journal entry with its data retained	Pass	Successfully have past journal entries displayed with their retained journal data displaying during edits.	2/5/2022
9	9.1	When user clicks 'history' button they should see their frequent word usage displayed from aggregated journal entries	No word usage data is gathered yet	Fail	As of now, feature decided to not be implemented.	2/5/2022
9	9.2	When user clicks 'journal history' button but there are no previous journal entries, the user is prompted to start journaling	User is not prompted, but given options at top of page	Pass	Feature changed	2/5/2022
10	10.1	When user clicks 'mood history' button they should see their prior mood history	User clicks 'tracker' button they see prior mood history entries displayed in graph format	Pass	Feature changed	2/5/2022

		entries displayed in graph format				
10	10.2	When user clicks ‘user data’ button but there is no data, user should be prompted to quick journal	No additional information is available other than mood	Fail	Feature yet to be added	2/5/2022
11	11.1	When user clicks ‘user data’ and clicks on ‘mood chart’ they should see how their mood has changed over time	Hard-coded data is displayed when user clicks into these features	Pass	Chart functionality is there	2/5/2022
12	12.1	When a user logs in, the application should authenticate them	When a user logs in, the application authenticates them	Pass	Authentication by providing JSON Web Token to users on signup and/or login. Token validations happen upon login.	2/5/2022
12	12.2	Authenticated users are shown the content they are authorized to see	Authenticated users are shown the content they are authorized to see	Pass	Authenticate users by having their user account ID stored in their posts within “creator:” field.	2/5/2022
13	13.1	When a journal entry is posted that information should be visible on the database and stored properly	When a journal entry is posted that information should be visible on the database and stored properly	Pass	Post ID, and schema fields and text display in order in accordance to designed	2/5/2022

					schema. Entries include user account ID.	

Testing Review

During the testing section, we discovered a method in which our application should flow and operate. Previously, we were unsure of the specifics to how a potential user would navigate throughout the app or the exact details of everything that would be displayed to them. However, as we move forward, we are now able to better implement our knowledge on the various aspects of our application, allocate time more wisely and reflect on the applications functionality, such as archived journal section and elements within it. So, as we include additional sections or features, we can dedicate more time to thinking about how they will operate. Throughout the user acceptance testing process, we were able to find areas to enhance the overall user experience of our application and discovered items of which we were not previously aware.

Change Management Plan

In the event of required changes - to further enhance the functionality and integrity of the Anxiety Knot application - we developed a set of guidelines listed below.

1. Any group member, user, or instructor can request a change.
    a. Under the circumstances that the change is within scope and will not cause significant delays of tasks or milestone progression.

2. If a change requested is agreed upon by team majority or our advisor sees it as applicable.
    a. We will move forward with the change.
    b. If the change is core to the functionality of our app and how users operate it will be designated as acceptable, and that will be focused on first.
    c. Otherwise, if the change improves a user's experience it should be considered.
3. Pros to our change management plan.
    a. Open ended in terms of team member(s) requesting a change.
    b. Each member should feel comfortable proposing any ideas or changes they want to request to the project.
4. Cons to our change management plan.
    a. Team members disagree on a change and there is a split decision or indecisiveness.
5. If all team members are not available to decide on the change:
    a. A selected team member who will impact the most will be designated to make the final decision.
    b. A different team member can oversee changes for different areas of our application and scope.
6. If the change is critical to the functionality of our app, that change will:
    a. Be highlighted and made a priority.

b. If the severity of the change is critical or urgent, multiple team members will directly assist to ensure change functionality.

7. If a change has a significant impact on our application -out of scope- or team members are split on the decision.
    a. We will outsource the decision and get thoughts on our process via our advisor.

Budget

The table below is the estimated costs of building our application, as well as the ongoing costs of its maintenance. We included the labor costs of app development and the cost of outsourcing minor graphic design for the app. Included in the development costs are one-time expenses of computers, operating systems, and graphic design software. The ongoing annual expenses include the cost of labor for application maintenance, amount of Internet for development and maintenance, and ongoing hosting cost. The reason we chose to include these costs is because we believe they encompass all expenses needed to develop, maintain, and host our application.

Table 3: Project Budget

		Rate Per/Hr.	Time (Hours)	Costs	Ongoing Annual		
					Rate Per/Hr.	**Work Effort (Hours)**	**Support Cost**
Labor - IT	Software Devs: 3 Network/Admin: 1	30.00	756 Hrs.	$ 22680	30.00	104	$ 3120

Labor - External	Graphic Design (logo & badges)	30.00	5 Hrs.	$ 150	-	––	$ 0
Software - External	Windows 10: 4 Adobe Suite: 2	-		$ 790	-	-	$ 0
Hardware - External	Desktop Computer: 3 Laptop: 1	-		$ 5600	-	-	$ 0
Misc.	AWS Hosting (free until app released) MongoDB (Free Tier) VS Code	- - -		0 0 0	- - -	- - -	$ 2981.88
TOTAL				$ 21660			$ 5061.88

Problems Encountered and Analysis of Problems Solved

A retrospective discussing the biggest hurdles faced by our team during the development of our application and what we did to overcome them. We look back and discuss the issues of hosting, coordinating schedules, time and project management, and issues with keeping our repo organized and up to date.

One problem faced was the proper creation of our front-end and back-end hosting services. Initially, it was decided to use Microsoft's Azure services. However, complications arose when it became unclear on how to properly create the necessary virtual services. This problem was furthered by the need to use GitHub Actions to build and deploy our front-end and back-end code, relying on a specific YAML file configured for Microsoft Azure. We were able to solve these issues by abandoning Azure and GitHub and directing our resources to AWS (Amazon Web Services). Amazon Web Services offered a more convenient, easy, and direct method to build our front/back-end services, while providing the ability to build and deploy our application's code within the same environment.

Another difficulty was determining the proper times and days to hold our team meetings. This is due to everybody having different schedules because of internships, full-time job(s), participating in other courses, or other external obligations. Although, we easily discovered the best times and dates for everyone to virtually gather and collaborate on current/future progression. Group text messaging is another method we decided to use for easier communication convenience or for important necessities.

We struggled with time management throughout the semester but began to see significant improvement on it in the latter half of the semester. Because we were all trying to learn the technologies being used as well as apply what we were learning, while also managing other course loads, it was difficult to keep the application on schedule.

While working on features we ran into a few issues where there would be multiple branches that were many commits ahead of the Main branch which made it difficult to merge them. We began making more regular commits and merging more often, which addressed this problem.

We tried to meet often and with regularity during the initial weeks but were able to meet twice a week once everyone could coordinate their schedules. This helped solve the problem of everyone being up to date with the project as well as helping us with time management and coordinating who would work on each feature as the project evolved.

Conclusion

Fall Semester 2021

Throughout the duration of the fall semester, as a team we learned the importance of effective project, time management, and the value of attending scheduled meetings - even if only for standups. Effective communication became especially important in making sure everyone was understanding the task at hand and getting their work done effectively. Additionally, we learned about the importance of monitoring technical debt – making sure it did not exceed a certain limit of which it inhibited us from getting other crucial work done. Also, the importance of focusing on a minimum viable product before worrying about additional features. Sometimes we must watch out for scope creep and be wary of what we can handle.

There were some abilities and skills each member acquired throughout this experience. Although half of the team were familiar with Angular, not one of us were anywhere near expert level programmers. However, we were able to enhance this skillset through various online Websites and videos. Additionally, expand our TypeScript vs. JavaScript knowledge, take part in using host services -AWS- and create website schematics via design programs (Figma) for UI/UX development.

Spring Semester 2022

As team Anxiety Knot continues to dedicate themselves to improving the project and accomplishing our goals, we are looking forward to fully implementing all our features into the application. This includes adding emoji icons, user account creation and login, front-end and back-end security/routing, user achievements, and improving the overall application responsiveness. Of course, we will dedicate ourselves to cleaning up and reorganizing our

application's coding to make it more comprehensive and easier to navigate before our final presentation. Meanwhile, our group will continue to contact and set goals for ourselves to completely finalize our project. Ultimately, everyone in group 34 better understood the importance of project management, the benefit of working together to combat against persistent complications, and the satisfaction of deploying a solid product while continuing to accomplish our goals.

References

Reynolds, Brandon R. “Feeling Re-Entry Anxiety? Expert Advice on Navigating Covid-19 Reopenings.” *Feeling Re-Entry Anxiety? Expert Advice on Navigating COVID-19 Reopenings | UC San Francisco*. The Regents of The University of California, May 24, 2020. Last modified May 24, 2020. Accessed August 30, 2021. https://www.ucsf.edu/news/2021/05/420581/feeling-re-entry-anxiety-expert-advice-navigating-covid-19-reopenings.

Gilbert, Marissa. “MANAGING STRESS AND MENTAL WELLBEING: INCL IMPACT OF COVID-19 - US - JULY 2020,” July 2020. Accessed August 30, 2021. https://reports-mintel-com.uc.idm.oclc.org.

Goodwin, Renee D., Andrea H. Weinberger, June H. Kim, Melody Wu, and Sandro Galea. “Trends in Anxiety among Adults in the United States, 2008–2018: Rapid Increases among Young Adults.” *Journal of Psychiatric Research* 130 (August 14, 2020): 441–446.

Tamir, Hod. “The Anti-Anxiety Notebook: Cognitive Behavior Therapy and Other Essentials” *Therapy Notebooks*. Accessed April 24, 2022. https://shop.therapynotebooks.com/products/the-anti-anxiety-therapy-notebook-retail.