

**Final Project for SW Engineering Class CSC
648-848 Section 01 Summer 2023**

Team 2
Zenful

Suzanne Heikal (sheikal@mail.sfsu.edu)

Kimon Monokandilos

Braden Dalit

Jada Campbell

Lucas Soto

Edward Chen

<http://34.219.66.26/>

08/05/23

History Table:

| Milestone's Versions | Date |
|----------------------|----------|
| M5V2 | 08/05/23 |
| M4V2 | 08/05/23 |
| M4V1 | 07/27/23 |
| M3V2 | 08/01/23 |
| M3V1 | 07/24/23 |
| M2V2 | 07/24/23 |
| M2V1 | 06/28/23 |
| M1V2 | 06/26/23 |
| M1V1 | 06/12/23 |

Table of contents

| | |
|---|-----------|
| Product Summary | 6 |
| Milestone Documents M1-M4 | 8 |
| Milestone 1..... | 8 |
| Executive Summary..... | 8 |
| Main Use Cases..... | 9 |
| Main Use Cases Diagram..... | 9 |
| List of Main Data Items and Entities..... | 14 |
| Initial List of Functional Requirements..... | 15 |
| General User..... | 15 |
| Registered User..... | 15 |
| Admin..... | 15 |
| Home Page..... | 15 |
| Registration..... | 15 |
| Login..... | 15 |
| Journal..... | 16 |
| To-do List..... | 16 |
| Features..... | 16 |
| List of Non-Functional Requirements..... | 17 |
| System Requirements..... | 17 |
| Coding Standards..... | 17 |
| Content and Appearance..... | 17 |
| Security..... | 20 |
| Marketing..... | 20 |

| | |
|--|-----------|
| User Data..... | 20 |
| Performance and Scalability..... | 21 |
| Environmental..... | 21 |
| Competitive Analysis..... | 22 |
| Features of Competitors..... | 22 |
| Competitive Analysis Summary and Weakness..... | 24 |
| High-level System Architecture and Technologies Used..... | 25 |
| Milestone 2..... | 26 |
| Data Definitions..... | 26 |
| General User..... | 26 |
| Registered User..... | 26 |
| Admin..... | 26 |
| Home Page..... | 26 |
| Registration..... | 27 |
| Login..... | 27 |
| Journal..... | 27 |
| Features..... | 27 |
| Functional Requirements..... | 29 |
| Priority 1..... | 29 |
| Priority 2..... | 30 |
| Priority 3..... | 30 |
| UI Mockups and StoryBoards..... | 31 |
| High-Level DAtabase Architecture and Organization..... | 37 |
| Database Requirements..... | 37 |
| Entity Attributes, Relationships, and Domains..... | 37 |

| | |
|---|-----------|
| Entity Relationship Diagram..... | 39 |
| DBMS..... | 39 |
| Media Storage..... | 39 |
| Search/Filter..... | 39 |
| High-Level APIs and Main Algorithms..... | 40 |
| Search Algorithm..... | 40 |
| Filter Algorithm..... | 40 |
| High-Level UML Diagrams..... | 41 |
| High-Level Application Network and Deployment Diagram..... | 42 |
| Key Risks for Project..... | 43 |
| Project Management..... | 45 |
| Vertical Prototype Feedback..... | 46 |
| Milestone 3..... | 47 |
| Data Definitions..... | 47 |
| General User..... | 47 |
| Registered User..... | 47 |
| Admin..... | 47 |
| Home Page..... | 47 |
| Registration..... | 48 |
| Login..... | 48 |
| Journal..... | 48 |
| Features..... | 48 |
| Functional Requirements..... | 50 |
| Priority 1..... | 50 |
| Priority 2..... | 51 |

| | |
|---|-----------|
| Priority 3..... | 51 |
| Wireframes Mockups..... | 52 |
| Registration..... | 52 |
| Login..... | 53 |
| Starting a Journal..... | 54 |
| High-Level Database Architecture and Organization..... | 56 |
| ERD..... | 56 |
| EER..... | 57 |
| UML..... | 58 |
| Deployment and Network Diagram..... | 59 |
| Horizontal Prototype Feedback | 60 |
| Milestone 4..... | 61 |
| Product Summary..... | 61 |
| Final Functions..... | 61 |
| URL..... | 62 |
| Usability Test Plan..... | 63 |
| Usability Effectiveness..... | 64 |
| Questionnaire..... | 66 |
| QA Test Plan..... | 67 |
| Code Review..... | 72 |
| Best Practices for Security..... | 76 |
| Assets to Protect..... | 76 |
| Data Validation Include..... | 76 |
| Self-Check:Non-Functional Specs..... | 77 |
| System Requirements..... | 77 |

| | |
|---|-----------|
| Coding Standards..... | 77 |
| Content and Appearance..... | 77 |
| Security..... | 80 |
| Marketing..... | 81 |
| User Data..... | 81 |
| Performance and Scalability..... | 82 |
| Environmental..... | 82 |
| Post Analysis..... | 84 |
| Team Member Contributions..... | 86 |

Product Summary

Our product is Zenful. It is a relaxing journaling app with added features, such as music, for a more personal and positive experience. What makes this app special is the safe space it provides. You can create a journal, with your eyes only, and you can listen to relaxing music while doing it! The best way to market this product would be for us to utilize social media. Social media is a great tool for advertising products that have to do with productivity. With an app like ours, it is best promoted on social media. It would start with ads on Tik Tok. This is where you would find most of the users who are looking for something relaxing yet productive. The ad would include a sample of the templates we offer, as well as all the possibilities of the app. Then we would also do small skits that would get people interested. Some skits include someone wanting to organize their thoughts in one place that they could protect. Another way to market this product would be to market it on Instagram where people are influencers and would need to make journal content ideas, etc. Our main focus will be people who need to be able to write out what they need to get done, or just simply for relaxing purposes. We should also focus on people who are going to do content because this is an app with the purpose of letting people decide what they need to write out in an electronic way that is private and secure.

Home Page

1. The home page shall provide a description on Zenful.
2. The home page shall show the user where to start.

General User

1. General users shall be able to view the home page.
2. General users shall be able to register.
3. General users shall be able to access a guest page.

Registered User

1. Registered users shall be able to login.
2. Registered users shall have access to Zenful once they are logged in.

Registration

1. Users shall need to provide their email address during registration.
2. Users shall need to provide their username during registration.
3. Users shall need to provide their password during registration.

Login

1. Registered users shall be able to login.
2. Registered users shall need to provide their username and password to login.
3. Registered users shall have the option to logout.

Admin

1. The admin shall be able to access and modify the database.
2. The admin shall be provided with full data access.
3. The admin shall need to login before accessing admin permissions.

Journal

1. Registered users shall be able to create a new private (accessible only to the registered user and no-one else) journal entry.
2. Registered users shall be able to see all of their existing private journal entries.

Features

1. Registered users shall be able to play music through Zenful.
2. Registered users shall be able to pause currently playing music.
3. Registered users shall be able to skip currently playing music.

URL: <http://34.219.66.26/>

Executive Summary

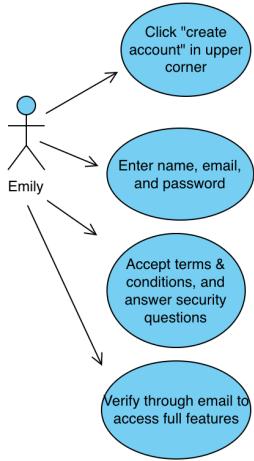
Are you feeling stressed, lonely, or lacking productivity? It seems as though it's either or, being productive with your tasks or relaxing. We have apps that help you relax, and we have apps that keep you to stay on top of your tasks. But there is never something that can help us do both. Why is it that one has to be over the other? Why can't both go together? It is said that stress might come from not taking action over something that you can have some control over. Why not create an application that focuses on this quote and takes it to another level, focusing on relaxation and productivity. This way, you can ease your stress in a double hit; one with being productive, and the other one by taking things slow and reflecting on where you stand mentally.

With our application, we can do both. This project is about a peaceful, mental-health positive app called "Zenful" where you can both be productive and have a relaxing session of journaling. The web app will feature a collection of calming sounds for you to play while you either make a productive journal about your different days. The importance of this application is that it is centered around you. This app takes both things that seem so parallel, relaxing and being productive, and it merges them into two. You never have to feel as though you are being unproductive while you are relaxing when you are Zenful. Everyone can create an account within the application, and you will come back to your progress as it is saved. You can also pick your own music if you choose. The application will feature organized ways to write journals. With your journal, you write the date and a title to also keep your progress in check.

This is unique because it encompasses a safe space journal, and a music feature, both centered around comfort to help the user ease, as they continue to do what is productive for them. The relaxation advantages aspect of our application, mixed with our user-friendly features, topping it off with a productivity boost, makes our app the center of you.

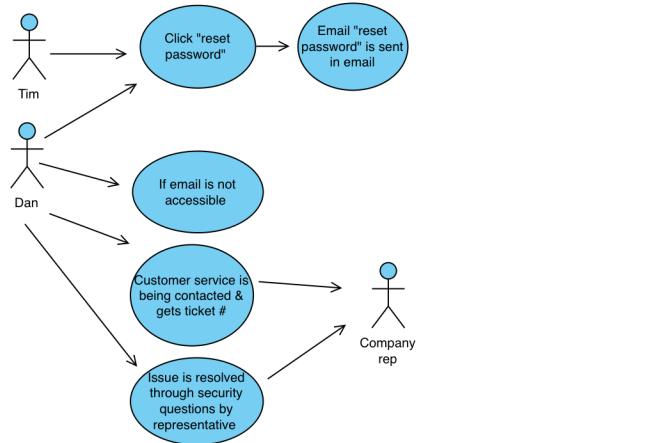
Main Use Cases

1. Use Case (Creating an account)



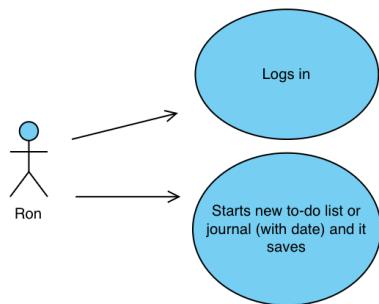
Emily loves to be productive but is having a hard time because it stresses her out. She found Zenful, and was intrigued by how you can both have a relaxing time, but be productive. She is taken to the front page where she is met by an introduction, and in the corner is a “login/create account” here. All she has to do is put her email, her name, sign terms & conditions, answer security questions, and put a password. She cannot access full features until verifying with her email.

2. Use Case (Customer support with account access issues)



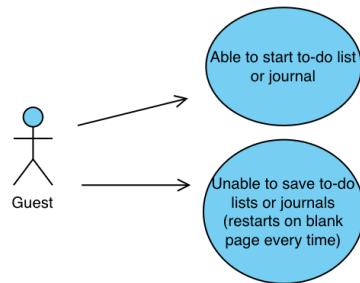
Dan is having issues with his login. He could've sworn his password was right, but it is not giving him access to his account. He is worried because his email isn't working, so he cannot get a "reset password" message in there. His friend, Tim, also forgot his password, but he had his email so he had an option to reset the password. He, however, has to contact customer service with the correct email address. He gets a ticket # and the company representative assists him (will ask about his security questions).

3. Use Case (Using to-do list/journal as a registered account)



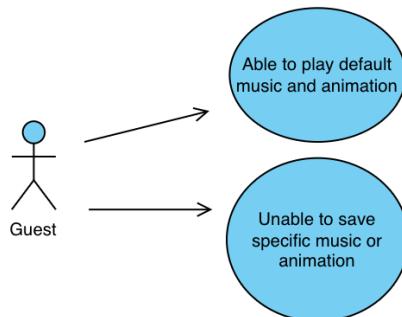
Ron is a registered user in Zenful. He hasn't logged into the site for a long time because he was busy with his job, where he worked 12 hours straight. His job is draining and he needs an escape. He wants to go inside his journals in this app to see how he was feeling before he joined this new job. He needs to log in. He opens his page and sees all his saved journals and saved lists.

4. Use Case (Using to-do list/journal as a guest)



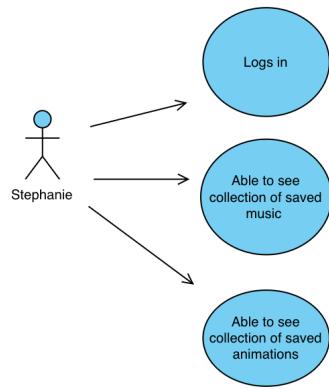
A user wants to try out an app just for a short term homework assignment that will take them 2 hours to do. It will take them 2 hours because they usually get very distracted. A simple task takes them a long time because they have ADHD. This user is able to write a to-do list, listen to music, and write a journal. This user, however, will lose all progress once the page is refreshed because they are not a registered user of Zenful.

5. Use Case (User using sounds/animation as a guest)



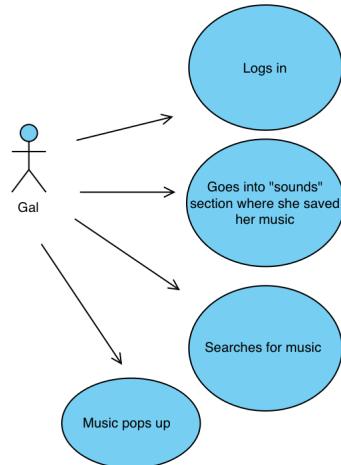
A user wants to play specific animation, as well as music while they color in their book they purchased from a store. They can listen to music and have animation on the site, but their progress will not be saved because they are not a registered user with Zenful. Every now and then, default animations and music might be discontinued so this user has no collection, and can only use the ones available now while they finish their coloring activity.

6. Use Case (User using sounds/animation as a registered account)



Stephanie has been a loyal user for a long time in Zenful. She enjoys being on the app because of the peace it brings her, while she stays productive but also relaxes. She had a rough day, and she was remembering how last month was a very relaxing month. She decides to log in with the intention of reading her past journal and listening to a saved song that is no longer accessible. She searches up the date of the month, and finds her past journal. She starts reading it, along with the song in the background and is awokened by hope.

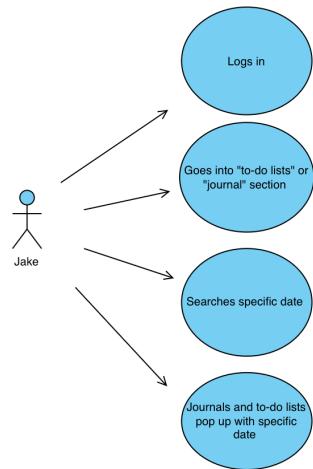
7. Use Case (Searching sounds for registered account)



Gal has too many saved sounds and animations. She always does this, she hoards things because of the emotion they hold and now her Zenful account is very overwhelming. She

wants to be able to search up ones that are from a while back, and possibly get rid of some. She is a registered user, so she logs in. All she needs to do is search up the key words and they will pop up. She now can clear her space and focus on the most important memories.

8. Use Case (Searching dates for to-do list/journal as a registered account)



Jake is a registered user and has many journal entries. He wants to find the one that was on his birthday. He wants to do this because he doesn't remember what some of his friends got him due to some drinks he had later that night. He knows he received all the gifts, but we forgot who gave him what. He loves his friends and wants to send them all thank you cards. He logs in, and types in the date of his birthday, and sees his journal entry. Now he knows who got him what, and was able to reminisce.

List of Main Data Items & Entities

General User: can access only the Home page. Needs to login/register to use Zenful.

Registered User: can access Journal and To-do list. Needs to login to use Zenful.

Admin: can access all data and modify database. Needs to login to access data.

Home Page: accessible to all users. Contains buttons to sign-in and register.

Registration: accessible to all users. Registers a new user into the system.

Login: accessible to all users. Logs in registered users.

Journal: accessible only to the specific registered user that created the journal, and the admin.

To-do List: accessible only to the specific registered user that created the to-do list, and the admin.

Features: includes animations and music. Accessible to all registered users, and the admin.

Initial List of Functional Requirements

1. General User

- 1.1. The system shall display the Home Page for all general users.
- 1.2. The system shall provide the option for all general users to register.

2. Registered User

- 2.1. The system shall provide the option for all registered users to login.
- 2.2. The system shall provide the registered user access to Zenful once they are logged in.

3. Admin

- 3.1. The system shall require the Admin to login before accessing Admin permissions.
- 3.2. The system shall provide the Admin with all data access.
- 3.3. The system shall allow the Admin to access and modify the database.

4. Home page

- 4.1. The system shall display step-by-step instructions to use Zenful.
- 4.2. The system shall display log-in and register options for the user.

5. Registration

- 5.1. The system shall record the user's first name.
- 5.2. The system shall record the user's last name.
- 5.3. The system shall record the user's email address.
- 5.4. The system shall record the user's username.
- 5.5. The system shall record the user's password.
- 5.6. The system shall display whether the user's username is available.
- 5.7. The system shall display whether the user's email is available.
- 5.8. The system shall display whether the password matches the confirmed password.
- 5.9. The system shall display that the user needs to choose a different email if their email is unavailable.
- 5.10. The system shall display that the user needs to choose a different username if their username is unavailable.
- 5.11. The system shall display that the password and confirmed password fields must match in order to register.
- 5.12. The system shall generate a unique user identification number for each registered user.
- 5.13. The system shall allow the user to edit their registration information before submitting the registration form.

6. Login

- 6.1. The system shall allow a registered user to login.
- 6.2. The system shall require a registered user's username to login.
- 6.3. The system shall require a registered user's password to login.
- 6.4. The system shall allow the registered user to select "forgot password".

- 6.5. Forgot Password: The system shall allow the registered user to enter their email address in order to receive an email with instruction to reset their password.
- 6.6. The system shall record the current signed-in registered user status.
- 6.7. The system shall allow a registered user to logout.
- 6.8. The system shall allow a registered user to change their password.
- 6.9. The system shall allow a registered user to view and edit their account details.

7. **Journal**

- 7.1. The system shall record journal entries.
- 7.2. The system shall display existing journal entries for the registered user.
- 7.3. The system shall allow a registered user to edit their journal entries.
- 7.4. The system shall allow a registered user to search for a journal entry.
- 7.5. The system shall allow a user to select a journal entry for editing.
- 7.6. The system shall allow a registered user to create a new journal entry.
- 7.7. The system shall allow a registered user to delete a journal entry.
- 7.8. The system shall allow a registered user to add a 4 digit passcode to a journal entry for privacy.

8. **To-do List**

- 8.1. The system shall record to-do lists.
- 8.2. The system shall display existing to-do lists for the registered user.
- 8.3. The system shall allow a registered user to edit their to-do lists.
- 8.4. The system shall allow a registered user to mark tasks as complete or incomplete on their to-do lists.
- 8.5. The system shall allow a registered user to search for a to-do list.
- 8.6. The system shall allow a user to select a to-do list.
- 8.7. The system shall allow a registered user to create a new to-do list.
- 8.8. The system shall allow a registered user to delete a to-do list.

9. **Features**

- 9.1. The system shall display an animation if animation is toggled on by a registered user.
- 9.2. The system shall allow a registered user to toggle their animation on or off.
- 9.3. The system shall allow a registered user to choose from available animations.
- 9.4. The system shall allow a registered user to upload their own animations.
- 9.5. The system shall allow a registered user to play music through Zenful.
- 9.6. The system shall allow a registered user to pause music.
- 9.7. The system shall allow a registered user to listen to available music.
- 9.8. The system shall allow a registered user to upload their own music.
- 9.9. The system shall record uploaded music.

List of Non-Functional Requirements

1. System Requirements

- 1.1. Web app should be available on personal computers and laptops.
- 1.2. Web app should be able to service a few thousand clients.
- 1.3. Will take about 100 GB of bandwidth a month.
- 1.6. Web app will be easy for anyone to use.
- 1.7. If there is a delay on using site user should only experience a few seconds delay.
- 1.8. There should be enough storage for users to upload music to the website.
- 1.9. There should be enough storage for users to have hundreds of notes and or journal pages.
- 1.12. If there is maintenance there will be a banner on the website for all to see.

2. Coding Standards

- 2.1. All code is written in JavaScript.
- 2.2. All code will have notes attached for easy maintenance on scheduled days.

3. Content and Appearance

- 3.1. Landing page should have a neutral background.
- 3.2. The forefront of the page should be a highlight reel for the sites features.
- 3.3. There will be the site name and or logo in left hand upper corner.
- 3.4. Middle top will have maintenance schedule.
- 3.5. The upper right hand side should have the login/signup option.
- 3.6. The highlight reel will be in the middle of the page.
- 3.7. Site will have a neutral and cool toned theme for the website until user chooses their own theme for their usability.
- 3.8. There will also be a setting for light or dark.

3.9. There will be an option for logging in on the upper right hand side of the page with links to account settings.

3.10. Under the logo will be the header on the left hand side: Notes, Journals, Music, Avatar, Themes, Account.

3.11. On the bottom of the page shall be links to about us,faqs,contact us,terms of use, privacy policy.

3.12. The side should have a menu option to flip between the music section and the current page.

3.13. The companies information: name, address, establishment, last time site was updated should be on the bottom of the page.

3.14. Above companies information should be the Faq,About Us, Contact Us, Terms of use,Privacy Policy.

3.15. When in the notes tab, there will be title of note, date, saving function on the left hand side.

3.16. The title and date will be larger than the saving function.

3.17. Under the title of the note will be the file options, edit, insert, format, and help.

3.18. File options, edit, insert,format, and help should be slightly smaller than the title and date.

3.19. Under file options should be the Font, Zoom, Text Style, Text Size, Highlighter, Template, and Print function.

3.20. On the left hand side there will be a area where the titles of notes are listed.

3.21. On the right hand side there will be the Account function.

3.22. The header from the dashboard will be visible in the middle of the top of the page.

3.23. At the bottom of the page is the links to About Us, Faqs, Contact Us, Terms of Use, Privacy Policy. The company information and last update of site/feature.

3.24. In the bottom right should be the animated gif/animation.

3.25. The music player should be in the bottom left.

3.26. The background should be user pick and in the foreground should be the ‘paper’ the user is typing on.

3.27. When in the journaling tab, there will be a date at the top left hand side.

3.28. There will be the choice to add a title or leave just the date.

3.29. There will be the files, insert, format, edit, and help functions under the date.

3.30. The date should be larger than the files, insert , format, edit, and help functions.

3.31.Under files, insert, format, edit, and help should be font, font size, zoom, and print function.

3.32. On the right hand side should be the account function as well as the passcode toggle.

3.33. At the bottom of the page should be the About Us, Faqs, Contact Us, Terms of Use, Privacy Policy, Company Information and last update of site/feature.

3.34. In the bottom right should be the animated gif/animation.

3.35. The music player should be in the bottom left of the screen.

3.36. The background should be the users pick.

3.37. In the foreground should be the ‘paper’ the user is typing on.

3.38. When in the music tab there should be a mini player in the top left hand side.

3.39. It should have a play, pause, and back and forth function.

3.40. On the top left under the player should be the functions: Playlist(s), Browse, Just for You, Trending.

3.41. On the main page should be the pages that are toggled.

3.42. The page should show the background that the user has chosen.

3.43. At the bottom of the page should be the About Us, Faqs, Contact Us, Terms of Use, Privacy Policy, Company Information and last update of site/feature.

3.44. In the right hand side should be the account function in the upper right.

3.45. In the bottom right should be the animated gif/animation.

4. Security

- 4.1 The user will need to login in order to access any of the files on the server.
- 4.2. This will enable them to ensure their information is kept private and not open for strangers to engage with their work.
- 4.3. When toggled on the user will be prompted to enter a four number code for their passcode.
- 4.4. Users data should be stored on site storage.

5. Marketing

5.1. To market this product we will provide information on the key aspects of the site such as the ability to have a little animated friend that will help keep users motivated to use the app.

5.2. We will also tote about how you can add your music to the library and ensure that they can customize the page in preset themes.

5.3. The big point is that the app is able to have music, an animated gif or animation in the corner, customizable page themes, journal and note taking and the ability to passcode notes.

6. User Data

6.1 User data that will be collected is as follows: music genres liked by user, user age, user gender.

6.2. This information is gathered to help further the understanding of what the users enjoy and what can be added in the future for more user engagement.

6.3. This information is only used for the furthering of user enjoyment and is not sold to any outside parties.

6.4. User information is tracked through cookies.

6.5. User's have all rights to their privacy information and can opt to decline accepting trackers.

6.6. If a user decides to decline having their information stored, their information will be placed on a list of do not track.

6.7. Any change to the privacy policy will be sent out in an email to all users for them to accept again and or decline.

6.8. As well as a notice placed on the website header.

6.9. Last update to privacy policy will be placed at the bottom of the page.

6.10. The user has the right to request a copy of what information we have collected from them at any time.

6.11. Registered user will be able to accept or decline terms of conditions upon registering.

6.12. User will have the right to retain the information and usage data.

6.13. User shall have access to privacy policy upon registration

6.14. User will have access to privacy policy in a link at the footer of the page.

7. Performance and Scalability

7.1. Web app will be reliable at all times and few crashes.

7.2. There will be a fast and efficient switch between page options.

7.3. Web app should be able to handle 10 times its capacity.

7.4. The website should be up at all times unless there is maintenance.

7.5 Peak loads are expected to be about 5000 users.

8. Environmental

8.1. The environment should be compatible with web browsers of all kinds.

8.2. The environment should support a filter with dates for journal/to-do list

8.3. The environment should support a filter with animation/sounds

8.4. The environment should support a stable internet connection.

Competitive Analysis

| Feature/Company | Spotify | Apple notes | Mimi cute notes | Pandora | CollaNotes |
|-----------------|---|--|--|---|--|
| Strength | <ul style="list-style-type: none"> • Can upload music • Access to thousands of music • Access to other people's playlist • Light/dark mode • Save playlists • Make playlists • Suggests songs based on previous plays • Connect to Facebook | <ul style="list-style-type: none"> • Can use Apple pen • On Apple products • No limit • Lock feature • Can scan work in • Light/dark mode • Can title notes • Can draw notes • No ads | <ul style="list-style-type: none"> • Cute • 100+ themes • Onboard ing message • Can request new themes • Sticky note appearan ce of all notes to choose from • Can title notes • Customi ze notes backgrou nd • Customi ze app theme • Passcode lock • Can change font • No ads | <ul style="list-style-type: none"> • On multiple platfor ms • Listen to music you like • Keep collecti ons • Podcast s • For you section • Personalizes what plays next | <ul style="list-style-type: none"> • Import notes • Import files • Folders • Annotation • Light/dark mode • Records audio • Translation • Collab with others • Digital planner • Can add links • Can present on app • Can use stylus • No ads |
| Weakness | <ul style="list-style-type: none"> • Limited skips in free mode • Playlists confusing • No notes • No customization of background • Not a planner • Ads | <ul style="list-style-type: none"> • Only on Apple • Can't customize background • No music • Hard to insert files • Not a planner | <ul style="list-style-type: none"> • No music • No animation • Limited to backgrounds • Author makes • Font can be hard to read initially • No stylus use • Can't insert | <ul style="list-style-type: none"> • No notes • Can't customize background • Can't upload music • Limited skips in free mode • Not a planner • Ads | <ul style="list-style-type: none"> • No music • No customization of backgrounds • Apple only • No lock |

| | | | | | |
|--------------|--|---|---|--|--|
| | | | <ul style="list-style-type: none"> • files • Not a planner | | |
| Pricing | <ul style="list-style-type: none"> • Free and paid • Premium individual - \$9.99 a month • Duo-12.99 a month • Family- \$15.99 a month • Student - \$4.99 a month | <ul style="list-style-type: none"> • Free | <ul style="list-style-type: none"> • Free • Sticker library - \$1.99 a month / \$25.99 a year | <ul style="list-style-type: none"> • Free and premium • Pandora plus - \$4.99 a month / \$54.89 a year • Pandora premium - \$9.99 a month / \$109.89 a year • Pandora premium family- \$14.99 a month/ \$64.89 a year • Student premium - \$4.99 a month / \$59.88 a year • Military premium - \$7.99 a month / \$87.79 a year | <ul style="list-style-type: none"> • Free |
| Social media | <ul style="list-style-type: none"> • Music app • Twitter • Facebook | <ul style="list-style-type: none"> • Notes app • None | <ul style="list-style-type: none"> • Notes app • Twitter | <ul style="list-style-type: none"> • Music app • Twitter • Facebook • | <ul style="list-style-type: none"> • Notes app • Twitter • Facebook |
| Onboarding | <ul style="list-style-type: none"> • Given info first then shown how to sign up • Tour after sign up | <ul style="list-style-type: none"> • None | <ul style="list-style-type: none"> • Welcome note on homepage | <ul style="list-style-type: none"> • Given info then shown to sign up | <ul style="list-style-type: none"> • option of beginner tips |

| Feature | Spotify | Apple Notes | Mimi - Cute Notes | Pandora | Zenful |
|----------------|---------|-------------|-------------------|---------|--------|
| Journalling | - | + | + | - | ++ |
| Music Playlist | ++ | - | - | ++ | ++ |
| Customization | - | - | + | - | +++ |
| Animation | - | - | + | - | ++ |

Analysis

Zenful is a user-friendly web app that will allow for the personal note and journaling of individuals. When landing on the front page users will be given information about how the website works and all the key features like allowing them to upload music, customize background, passcode feature, and animated friend. Users will then be prompted to sign up for an account in order to start the service. Music section on Zenful will allow users to use the preselected music or make their own playlists of their own music. Users will be able to customize their backgrounds from preselected backgrounds. There will be a light and dark mode as well. Zenful will allow users to insert pdfs, insert pictures, draw, and create a journal book that has a passcode feature. There will be preset templates as well as a blank page for notes and journals.

Weakness: No social media aspect with others. No planner feature. Unable to present with a website. Unable to use a stylus. Unable to change font.

High-level System Architecture & Technologies Used

Server Host: Google Cloud 18.0.0

Operating System: Ubuntu 23.04

Database: MySQL 8.0

Web Server: NGINX 1.25.1

Server-Side Language: Javascript (1.11)

Additional Technologies:

Web Framework: React 18.2.0

IDE: VS Code Codespaces

SSL Certification: Lets Encrypt (Cert Bot)

Data Definitions

General User

The General User does not pertain to any entity. In order to become a “User” entity, the General User needs to register. The General User is essentially anyone who visits the Zenful web application for the first time, or does not have an account. The General User has access only to the Home Page, which contains step-by-step instructions on using Zenful. If the General User decides that they would like to use Zenful (Journal, Features) then they will need to complete Registration. In order to complete Registration, the General User will need to provide Zenful with a first name, last name, username, e-mail address, and password. Once the General User finishes the Registration process, they will become a Registered User.

Registered User

The Registered User pertains to the “User” entity, with attributes: Email(varchar), UserID(integer), Name(varchar), and Password(varchar). The Registered User is basically a General User that has completed the Registration process. The Registered User is now able to Login to use Zenful. In order to Login, the Registered User will need to provide their username and password, which they created during Registration. Upon logging in, the Registered User will have access to Zenful: a Journal and additional Features, which include animations and music. The Registered User is also able to log out at any moment.

Admin

The Admin also pertains to the “User” entity, with attributes: Email(varchar), UserID(integer), Name(varchar), and Password(varchar). The Admin differs from both the Registered User and General User in several ways. First, The Admin is a member of the Zenful team, so not anyone can become The Admin. Second, The Admin has access to all data on Zenful, including permissions to modify the database. The Admin can modify user accounts, Journals, and Zenful features (animations and music). Zenful takes user privacy very seriously, and The Admin only has access to data in order to ensure a smooth web application performance for all Zenful users.

Home Page

The Home Page does not pertain to any entity. The Home Page is accessible to all General Users who visit the Zenful web application. The Home Page serves the purpose to provide General Users with essential information: 1) What is Zenful? Zenful is your new source of relaxation, supporting not only productivity, but also peace. How so, you ask? Zenful will provide its users with a Journal along with Features like music and animations for peace. 2) What do I need to do to start using Zenful? First, complete Registration to create a new account. Second, Login to

Zenful. Third, Access the Journal and Features (music and animations) by using the corresponding links.

Registration

Registration is the process during which a User entity is formed. Registration is the process that General Users need to complete in order to become Registered Users. In order to complete Registration, General Users will first need to click on the “Register” link at the top right corner of our Zenful web application. General Users will then need to fill out a registration form, providing their first name, last name, username, e-mail address, and password. The General User will also need to confirm their password, accept the Terms of Service and Privacy Policy, and click submit. If the username the General User provided is not already existing in our database, the e-mail address is valid, and the password matches the confirmed password, the General User will complete Registration and become a Registered User.

Login

Login is the process a User entity needs to complete in order to become logged in. Login is the process that Registered Users need to complete in order to access Zenful. In order to Login, Registered Users will first need to click on the “Login” link at the top right corner of our Zenful web application. During Login, Registered Users need to provide their username along with their password by filling in the Login Form and clicking submit. If the username and password match exist in the database, the Registered User will be logged in. If the Registered User does not remember their password, they will be provided with a “Forgot Password” link in order to retrieve their password.

Journal

The Journal pertains to the “Journal” entity, with attributes: Date created(date), Title(varchar), and description(text). The Journal is only accessible once a Registered User has logged into Zenful. Upon clicking the Journal link on Zenful, a Registered User will be provided with a journal where they can write down anything they decide to. The Journal’s aim is to provide the Registered User with a quick opportunity during work to introspect, and note or work through their thoughts and feelings.

Relevant Data: document files (.pdf, .txt), image files (.jpg, .png, .jpeg, .gif).

Features

Zenful Features pertains to the “Customize” entity, with attributes: Music(text URL), Background(text URL), Name(varchar), and Animation(text URL). Zenful Features currently

include: animations and music. Similarly to the Journal, Zenful Features are only accessible once a Registered User has logged into Zenful. Upon clicking the Features link on Zenful, a Registered User will be provided with two sections: music, and animations. Under the music section, the Registered User will be provided with options for music to play. Music will be in the relaxing theme, namely: raindrops, lo-fi, and study/meditation sounds. Under the animations section, the Registered User will be provided with an on/off toggle for animations, along with options for animations. If the Registered User chooses to toggle animations on, they will see the animation play on a loop in the corner of their screen on the Journal section.

Relevant Data: music audio files (.mp3), animation video files (.mp4, .mov, .avi), animation image files (.jpg, .png, .jpeg, .gif).

Functional Requirements

Priority 1

Home Page

- 4.1. The system shall display step-by-step instructions to use Zenful.
- 4.2. The system shall display log-in and register options for the user.

General User

- 1.1. The system shall display the Home Page for all general users.
- 1.2. The system shall provide the option for all general users to register.

Registered User

- 2.1. The system shall provide the option for all registered users to login.
- 2.2. The system shall provide the registered user access to Zenful once they are logged in.

Registration

- 5.1. The system shall record the user's first name.
- 5.2. The system shall record the user's last name.
- 5.3. The system shall record the user's email address.
- 5.4. The system shall record the user's username.
- 5.5. The system shall record the user's password.
- 5.13. The system shall allow the user to edit their registration information before submitting the registration form.
- 5.12. The system shall generate a unique user identification number for each registered user.
- 5.6. The system shall display whether the user's username is available.
- 5.7. The system shall display whether the user's email is available.
- 5.8. The system shall display whether the password matches the confirmed password.
- 5.9. The system shall display that the user needs to choose a different email if their email is unavailable.
- 5.10. The system shall display that the user needs to choose a different username if their username is unavailable.
- 5.11. The system shall display that the password and confirmed password fields must match in order to register.

Login

- 6.1. The system shall allow a registered user to login.
- 6.2. The system shall require a registered user's username to login.
- 6.3. The system shall require a registered user's password to login.
- 6.7. The system shall allow a registered user to logout.
- 6.4. The system shall allow the registered user to select "forgot password".
- 6.5. Forgot Password: The system shall allow the registered user to enter their email address in order to receive an email with instructions to reset their password.
- 6.8. The system shall allow a registered user to change their password.
- 6.9. The system shall allow a registered user to view and edit their account details.

Admin

- 3.3. The system shall allow the Admin to access and modify the database.
- 3.2. The system shall provide the Admin with all data access.
- 3.1. The system shall require the Admin to login before accessing Admin permissions.

Journal

- 7.6. The system shall allow a registered user to create a new private (accessible only to the registered user and no-one else) journal entry.
- 7.3. The system shall allow a registered user to edit their private journal entries.
- 7.1. The system shall record private journal entries.
- 7.2. The system shall display existing private journal entries for the registered user.
- 7.7. The system shall allow a registered user to delete a private journal entry.

Features

- 9.1. The system shall display an animation if animation is toggled on by a registered user.
- 9.2. The system shall allow a registered user to toggle their animation on or off.
- 9.3. The system shall allow a registered user to choose from available public animations.
- 9.5. The system shall allow a registered user to play public music through Zenful.
- 9.6. The system shall allow a registered user to pause music.

Priority 2**Login**

- 6.6. The system shall record the current signed-in registered user status.

Journal

- 7.4. The system shall allow a registered user to search for a private journal entry.
- 7.5. The system shall allow a user to select a private journal entry for editing.

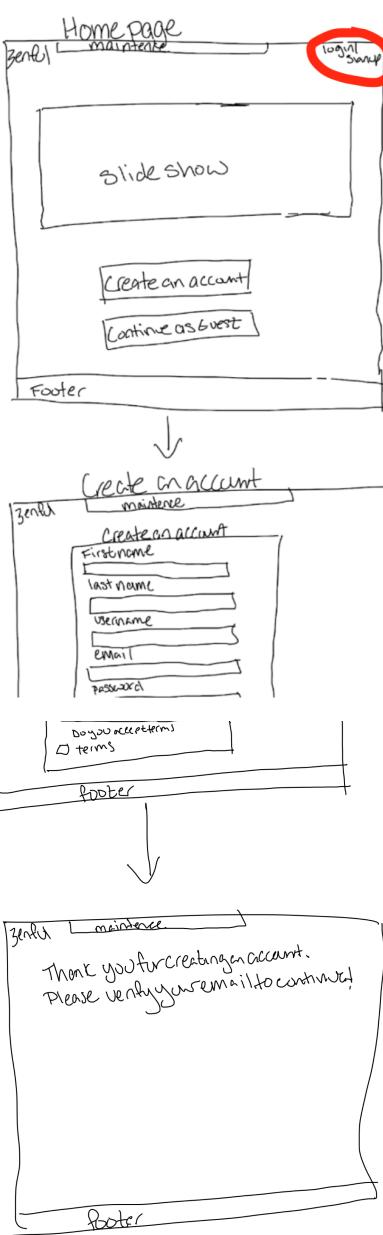
Priority 3**Journal**

- 7.8. The system shall allow a registered user to add a 4 digit passcode to a private journal entry for additional security.

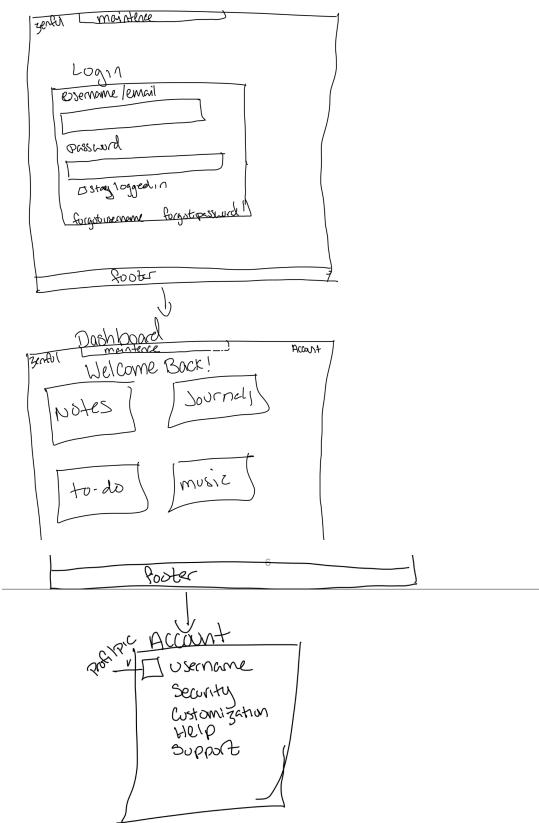
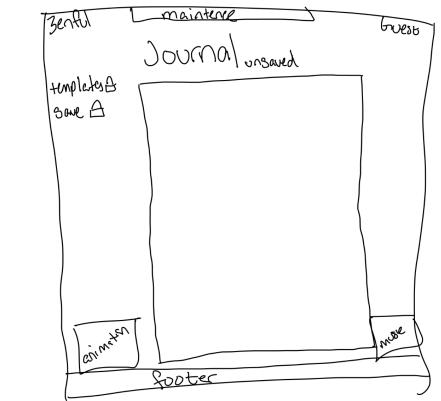
Features

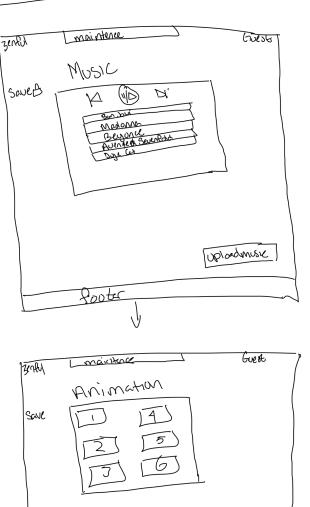
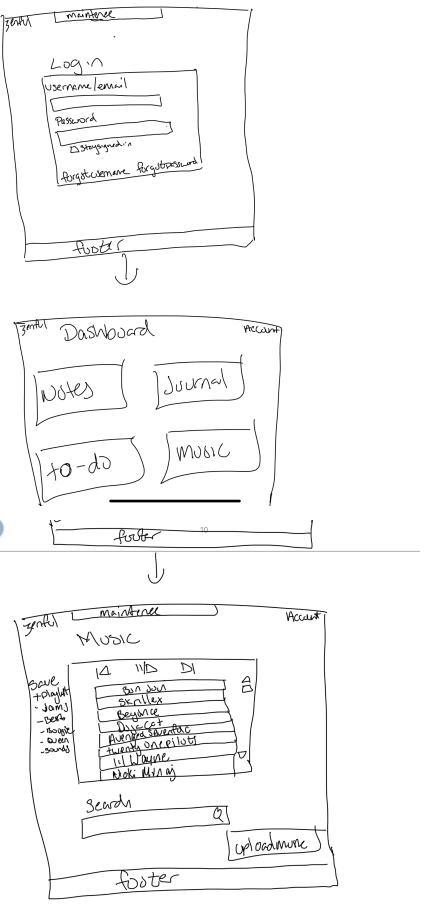
- 9.4. The system shall allow a registered user to upload their own private animations.
- 9.8. The system shall allow a registered user to upload their own private music.
- 9.9. The system shall record uploaded music.

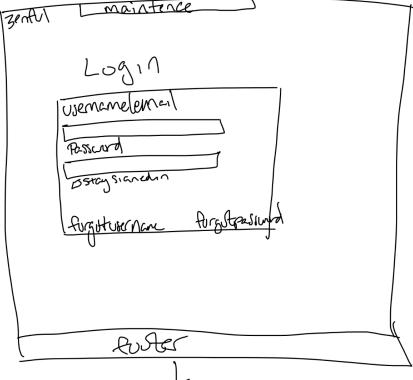
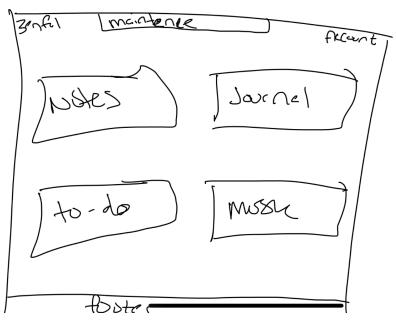
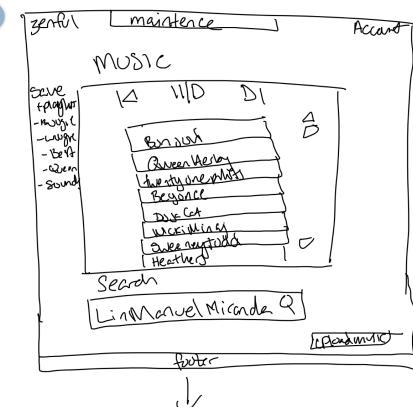
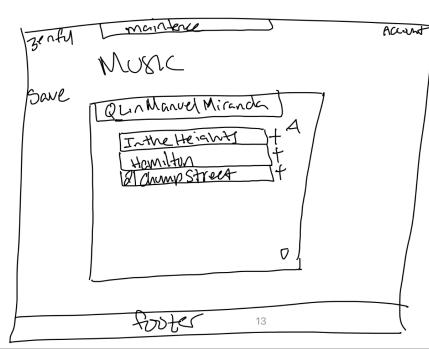
UI Mockups and Storyboards

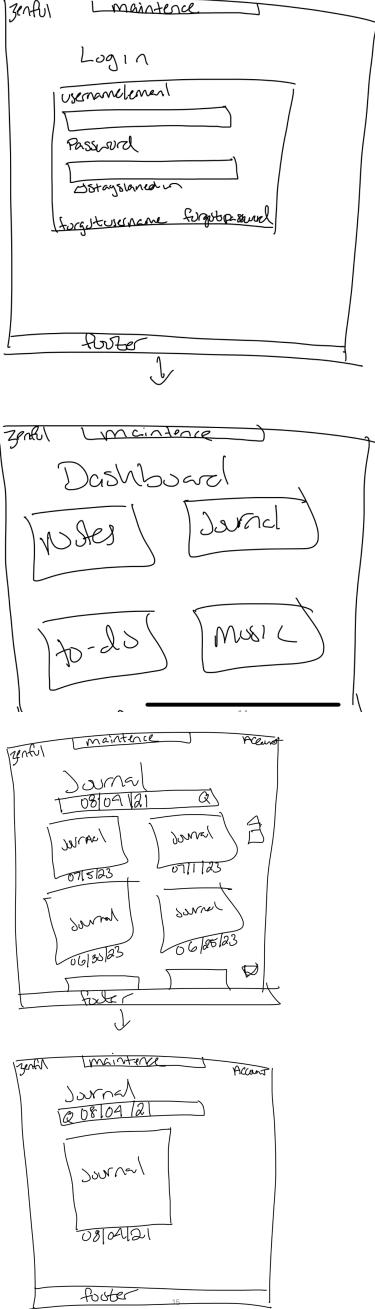
| Mock-up (UI) | Description (User Case) |
|--|---|
| <p><u>Case one</u></p>  <p>The storyboard illustrates a user flow for account creation:</p> <ul style="list-style-type: none"> Home page: Shows a "slide show" button and "Create an account" and "Continue as guest" links. A red circle highlights the "login/signup" link in the top right corner. Create account: A form with fields for Firstname, Lastname, Username, Email, and Password. It includes checkboxes for "Do you accept terms?" and "terms". Confirmation: A message stating "Thank you for creating an account. Please verify your email to continue!" | <p>Emily loves to be productive but is having a hard time because it stresses her out. She found Zenful, and was intrigued by how you can both have a relaxing time, but be productive. She is taken to the front page where she is met by an introduction, and in the corner is a “login/create account” here. All she has to do is put her email, her name, sign terms & conditions, answer security questions, and put a password. She cannot access full features until verifying with her email.</p> |

| Mock-up (UI) | Description (User Case) |
|------------------------|---|
| <p><u>Case two</u></p> | <p>Dan is having issues with his login. He could've sworn his password was right, but it is not giving him access to his account. He is worried because his email isn't working, so he cannot get a "reset password" message in there. His friend, Tim, also forgot his password, but he had his email so he had an option to reset the password. He, however, has to contact customer service with the correct email address. He gets a ticket # and the company representative assists him (will ask about his security questions).</p> |

| Mock-up (UI) | Description (User Case) |
|---|--|
| <p><u>Case three</u></p>  <p>The wireframe illustrates a user flow starting from a login screen, moving to a dashboard, and finally to a profile page.</p> <ul style="list-style-type: none"> Login Screen: Shows fields for "username / email" and "password", along with links for "Forgot login?" and "Forgot password?". Dashboard: Displays a "Welcome Back!" message, navigation links for "Notes", "Journal", "to-do", and "music", and a "Recent" section. Profile Page: Shows a profile picture placeholder, a "username" field, and links for "Security", "Customization", "Help", and "Support". | <p>Ron is a registered user in Zenful. He hasn't logged into the site for a long time because he was busy with his job, where he worked 12 hours straight. His job is draining and he needs an escape. He wants to go inside his journals in this app to see how he was feeling before he joined this new job. He needs to log in. He opens his page and sees all his saved journals and saved lists.</p> |
| <p><u>Case Four</u></p>  <p>The wireframe shows a guest user interface for a journal entry.</p> <ul style="list-style-type: none"> Header: Shows "Zenful" and "guest" status. Left Sidebar: Includes "templated", "save A", and "cancel" buttons. Main Area: A large empty box labeled "Journal unsaved". Bottom Navigation: Includes "more" and "footer" sections. | <p>A user wants to try out an app just for a short term homework assignment that will take them 2 hours to do. It will take them 2 hours because they usually get very distracted. A simple task takes them a long time because they have ADHD. This user is able to write a to-do list, listen to music, and write a journal. This user, however, will lose all progress once the page is refreshed because they are not a registered user of Zenful.</p> |

| Mock-up (UI) | Description (User Case) |
|---|---|
| <p><u>Case Five</u></p>  | <p>A user wants to play specific animation, as well as music while they color in their book they purchased from a store. They can listen to music and have animation on the site, but their progress will not be saved because they are not a registered user with Zenful. Every now and then, default animations and music might be discontinued so this user has no collection, and can only use the ones available now while they finish their coloring activity.</p> |
| <p><u>Case Six</u></p>  | <p>Stephanie has been a loyal user for a long time in Zenful. She enjoys being on the app because of the peace it brings her, while she stays productive but also relaxes. She had a rough day, and she was remembering how last month was a very relaxing month. She decides to log in with the intention of reading her past journal and listening to a saved song that is no longer accessible. She searches up the date of the month, and finds her past journal. She starts reading it, along with the song in the background and is awakened by hope.</p> |

| Mock-up (UI) | Description (User Case) |
|---|--|
| <p>Case Seven</p>   | <p>Gal has too many saved sounds and animations. She always does this, she hoards things because of the emotion they hold and now her Zenful account is very overwhelming. She wants to be able to search up ones that are from a while back, and possibly get rid of some. She is a registered user, so she logs in. All she needs to do is search up the key words and they will pop up. She now can clear her space and focus on the most important memories.</p> |
|   | |

| Mock-up (UI) | Description (User Case) |
|---|---|
| <p><u>Case eight</u></p>  <p>The wireframes illustrate a user flow:</p> <ul style="list-style-type: none"> Login Screen: Shows fields for 'username' and 'password' with placeholder text 'usernameplaceholder' and 'passwordplaceholder'. Buttons for 'Forgot Username' and 'Forgot Password' are at the bottom. Dashboard: Shows four main sections: 'Notes' (with a note icon), 'Journal' (with a journal icon), 'To-Do's' (with a list icon), and 'Music' (with a music note icon). Journal View: Shows a list of journal entries with dates: '08/09/21', '07/28/23', '07/11/23', and '06/16/23'. Each entry has a small icon next to it. Account Page: Shows a single journal entry for '08/09/21' with the title 'Journal'. | <p>Jake is a registered user and has many journal entries. He wants to find the one that was on his birthday. He wants to do this because he doesn't remember what some of his friends got him due to some drinks he had later that night. He knows he received all the gifts, but we forgot who gave him what. He loves his friends and wants to send them all thank you cards. He logins in, and types in the date of his birthday, and sees his journal entry. Now he knows who got him what, and was able to reminisce.</p> |

High Level Database Architecture & Organization

Database Requirements:

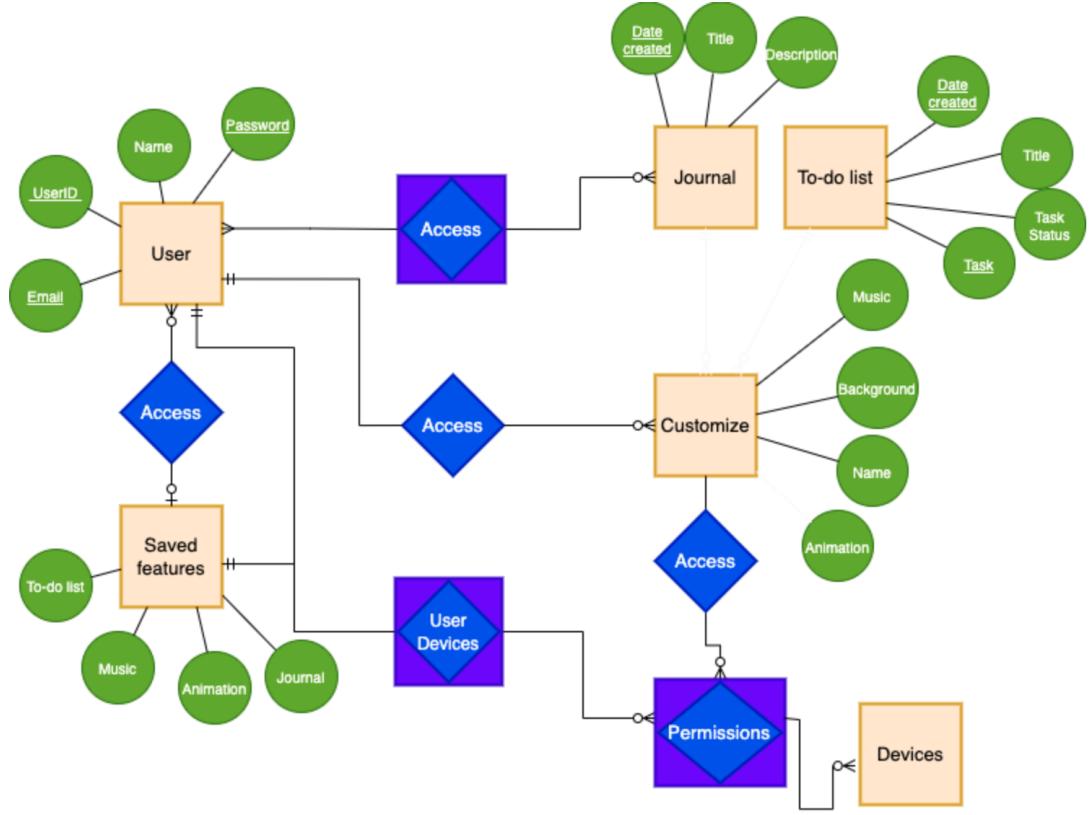
1. Users will choose if they will sign up for the site.
2. If they decide not to sign up they will see a preview of the site.
3. If they choose to sign up the user will be taken to the sign up page.
4. They will be prompted for email, password, and username.
5. They will accept the terms and conditions.
6. If denied, the session will end.
7. They will be taken to customization pages.
8. They will choose from a preset set of themes.
9. They will then move to music where they upload music or choose from the preset music.
10. They will then choose the animation they want.

Entity Attributes, Relationships, and Domains:

| Entity | Attributes | Relationships | Domains |
|------------|----------------------------------|--|--|
| User | Email, UserID, Name | <u>User accesses Journal</u> Many-to-Many <u>User accesses To-do List</u> Many-to-Many <u>User accesses Saved Music</u> One-to-Many <u>User accesses Customization</u> One-and-only-one | <u>Email</u> varchar(100), not null, valid email address <u>UserID</u> Primary key, integer, not null <u>Name</u> varchar(100), not null <u>Password</u> varchar(100), not null, valid password (fulfilling requirements, ex) 1 special character, 1 number |
| Journal | Date created, Title, Description | <u>Journal is customized</u> One-and-only-one | <u>Date created</u> Date, not null <u>Title</u> varchar(100) <u>Description</u> text |
| To-do List | Date created, Title, Task | <u>To-do List is customized</u> One-and-only-one | <u>Date created</u> Date, not null <u>Title</u> |

| | | | |
|----------------|-----------------------------|---|--|
| | | | varchar(100) <u>Task Status</u> Integer, 0 (not done) or 1 (done) <u>Task</u> text |
| Customize | Music, Background Animation | <u>Customize access</u> One-to-Many | <u>Music</u> Text (URL) <u>Background</u> Text (URL) <u>Name</u> varchar(100) <u>Animation</u> Text (URL) |
| Saved Features | To-do List, Music, Journal | <u>Saved Features acc</u> <u>permissions</u> One-and-only-one | <u>To-do List</u> Text <u>Music</u> Text (URL) <u>Animation</u> Text(URL) <u>Journal</u> Text |

Entity Relationship Diagram:



The **DBMS** we will be using is MySQL because everyone in the team is familiar with it, and because of its widely used features around the globe.

Media storage will be used in a file system(mysql), with audio(mp3), as well as images(mostly png).

Search/Filter architecture and implementation:

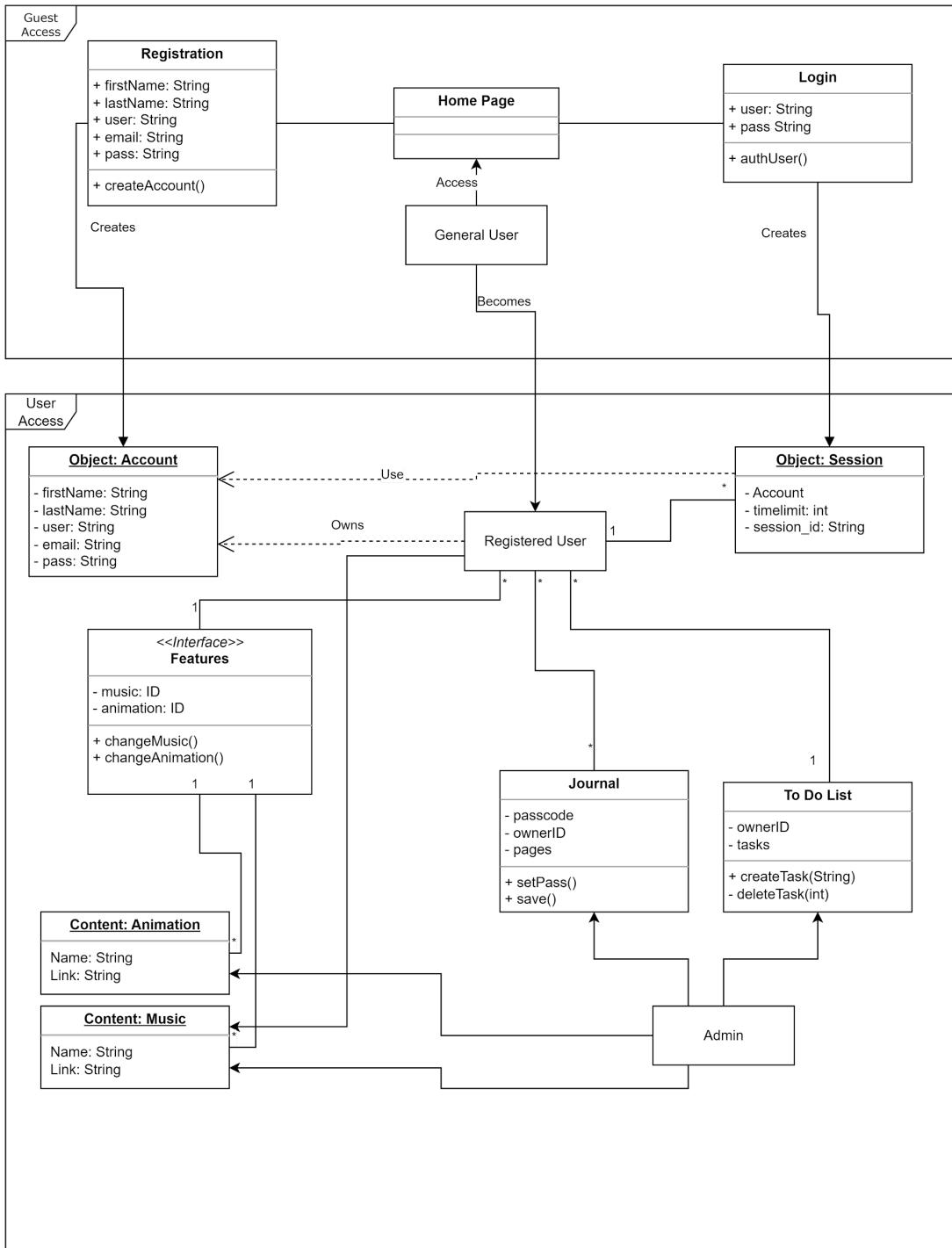
In order to search for a particular item, whether that be animation or music, the user is taken to a search bar alongside his “saved” features. He types in the string of the animation/music, and finds it being popped up. For dates in terms of journals/to-do lists, it will incorporate integers, not strings, of the particular date the journal was created. In terms of the filter, it will annihilate every single other string(if animation/music) or int(journal/to-do list) if it does not contain the keys given by the user.

High Level APIs and Main Algorithms

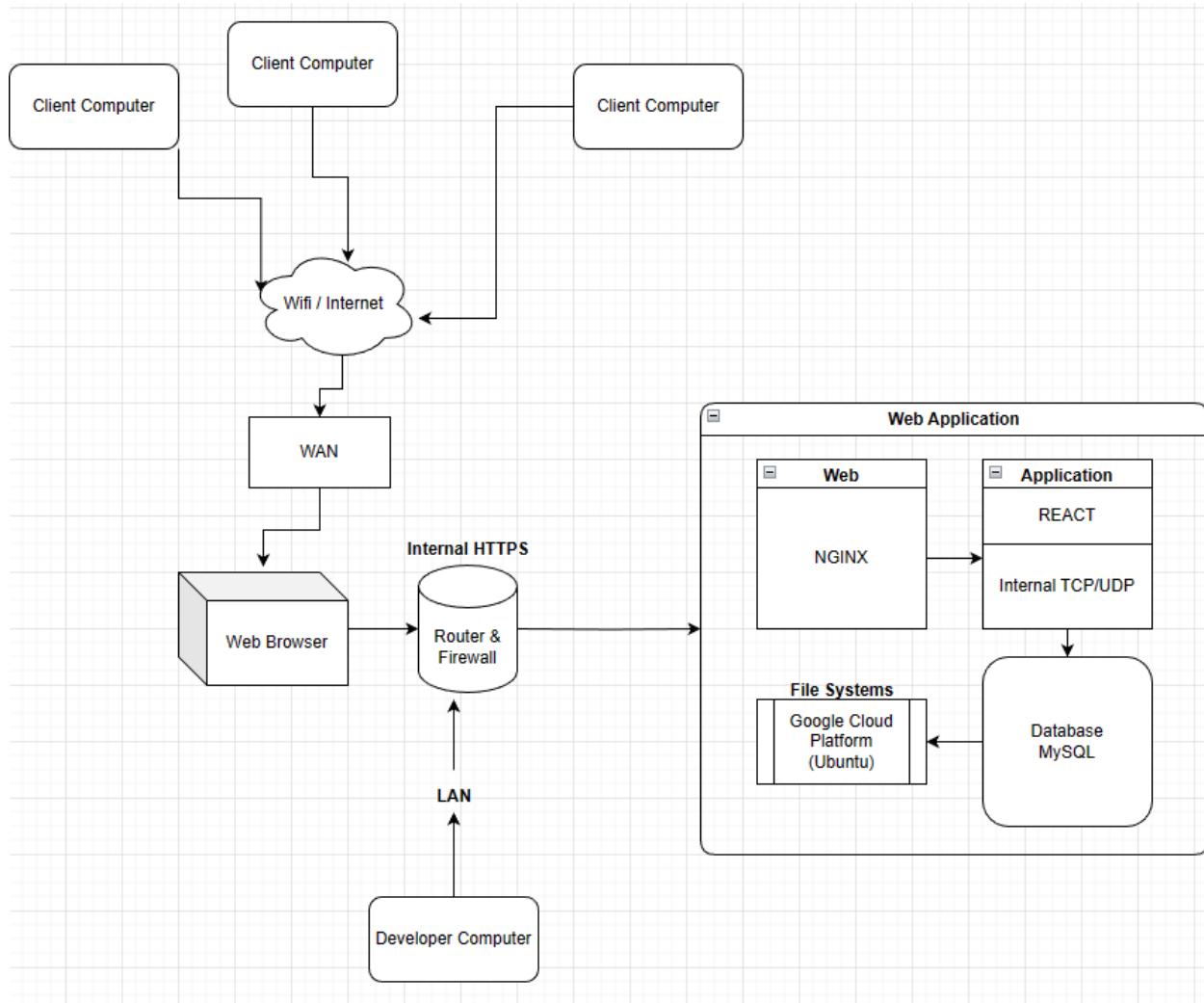
Search Algorithm - Users are able to post their own music as well as enjoy other user's public music. So in order for users to access other user's music, there must be a search function. Search will automatically list music and sort by most recently popular. When searching for a music or a user, the algorithm/api will provide a list of music and users with titles related to the search keywords, and still sorted by most recently popular.

Filter Algorithm - Users are able to search what date they did their to-do list or journal. This can be by the search bar they have, and they can filter out the specific date.

High Level UML Diagrams



High Level Application Network and Deployment Diagrams



Key Risks for Project

| Risk | Description | How To Solve |
|----------------------------|--|---|
| Skills Risk | <p>Most of the team seems to be skilled in javascript and has had at least a little experience with databases. However, what we lack in is experience with cloud computing and getting a web application to run on a web server and allow any user to access the web app without creating a manual instance.</p> | <p>We should teach each other and not let anyone fall behind in terms of all aspects of the code, and implementation of the project overall.</p> |
| Schedule Risk | <p>Multiple people work full-time and it would conflict with the schedule of making everything work smoothly, since everyone works at different times.</p> | <p>We should meet twice a week to touch base on everything, so that we are all on the same base and are doing what we need to do in terms of tasks.</p> |
| Technical risks | <p>There is no current security implementation, making the data accessible easily.</p> | <p>We need to implement a strong base word and not let anyone except the team + CEO have access to it.</p> |
| Teamwork risks | <p>Sometimes miscommunication lets other people a different way. This might put the group back.</p> | <p>In order to solve this, we have to check in with each other and it will be clear what is the correct direction, and the one that is misunderstood.</p> |
| Legal/content risks | <p>Musical content may be copyrighted and also there</p> | <p>In our app, we should offer copyright free music to pick</p> |

| | | |
|--|---|---|
| | <p>might be issues with data privacy and enforcement.</p> | <p>that is optimal for the user's experience, with different flavors.</p> |
|--|---|---|

Project Management

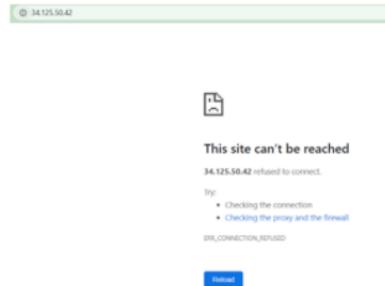
Trello has helped assist us on managing what the team needs to do, and when it needs to be done by. The group was able to log in, and see we have different tasks for different people. Under there, was a set of dates where we needed to have it done by. The person can also comment on how far they are in terms of the work so that everyone could be updated as well. This helped us in terms of the confusion that would surround the team sometimes on what needed to be done, by who, and by when. The way we used it was a silent update amongst each individual to all of us, the whole group. This also helped with those who wanted to double check sections after they were done. Sometimes people were confused about a section, and so seeing who had what allowed for more communication in terms of assistance or just getting things done.

The organizational aspect is always important when being in a big group, with different people who have different schedules. We have multiple people here working full time, and so being able to have an organization tool really helped. To help compliment the organizational tool of Trello, simple communication was made through Discord, along with announcements to get the team on board with how things were doing. Having an organizational platform, and having a form of communication to disclose all of the updates of the group really made us work on these things smoothly as a group.

Vertical Prototype Feedback

Feedback for vertical prototype:

- Unable to launch your website to test the vertical prototype. Please check if the server is up and running for us to grade.



Data Definitions

General User

The General User does not pertain to any entity. In order to become a “User” entity, the General User needs to register. The General User is essentially anyone who visits the Zenful web application for the first time, or does not have an account. The General User has access only to the Home Page, which contains step-by-step instructions on using Zenful. If the General User decides that they would like to use Zenful (Journal, Features) then they will need to complete Registration. In order to complete Registration, the General User will need to provide Zenful with a first name, last name, username, e-mail address, and password. Once the General User finishes the Registration process, they will become a Registered User.

Registered User

The Registered User pertains to the “User” entity, with attributes: Email(varchar), UserID(integer), Name(varchar), and Password(varchar). The Registered User is basically a General User that has completed the Registration process. The Registered User is now able to Login to use Zenful. In order to Login, the Registered User will need to provide their username and password, which they created during Registration. Upon logging in, the Registered User will have access to Zenful: a Journal and additional Features, which include animations and music. The Registered User is also able to log out at any moment.

Admin

The Admin also pertains to the “User” entity, with attributes: Email(varchar), UserID(integer), Name(varchar), and Password(varchar). The Admin differs from both the Registered User and General User in several ways. First, The Admin is a member of the Zenful team, so not anyone can become The Admin. Second, The Admin has access to all data on Zenful, including permissions to modify the database. The Admin can modify user accounts, Journals, and Zenful features (animations and music). Zenful takes user privacy very seriously, and The Admin only has access to data in order to ensure a smooth web application performance for all Zenful users.

Home Page

The Home Page does not pertain to any entity. The Home Page is accessible to all General Users who visit the Zenful web application. The Home Page serves the purpose to provide General Users with essential information: 1) What is Zenful? Zenful is your new source of relaxation, supporting not only productivity, but also peace. How so, you ask? Zenful will provide its users with a Journal along with Features like music and animations for peace. 2) What do I need to do to start using Zenful? First, complete Registration to create a new account. Second, Login to

Zenful. Third, Access the Journal and Features (music and animations) by using the corresponding links.

Registration

Registration is the process during which a User entity is formed. Registration is the process that General Users need to complete in order to become Registered Users. In order to complete Registration, General Users will first need to click on the “Register” link at the top right corner of our Zenful web application. General Users will then need to fill out a registration form, providing their first name, last name, username, e-mail address, and password. The General User will also need to confirm their password, accept the Terms of Service and Privacy Policy, and click submit. If the username the General User provided is not already existing in our database, the e-mail address is valid, and the password matches the confirmed password, the General User will complete Registration and become a Registered User.

Login

Login is the process a User entity needs to complete in order to become logged in. Login is the process that Registered Users need to complete in order to access Zenful. In order to Login, Registered Users will first need to click on the “Login” link at the top right corner of our Zenful web application. During Login, Registered Users need to provide their username along with their password by filling in the Login Form and clicking submit. If the username and password match exist in the database, the Registered User will be logged in. If the Registered User does not remember their password, they will be provided with a “Forgot Password” link in order to retrieve their password.

Journal

The Journal pertains to the “Journal” entity, with attributes: Date created(date), Title(varchar), and description(text). The Journal is only accessible once a Registered User has logged into Zenful. Upon clicking the Journal link on Zenful, a Registered User will be provided with a journal where they can write down anything they decide to. The Journal’s aim is to provide the Registered User with a quick opportunity during work to introspect, and note or work through their thoughts and feelings.

Relevant Data: document files (.pdf, .txt), image files (.jpg, .png, .jpeg, .gif).

Features

Zenful Features pertains to the “Customize” entity, with attributes: Music(text URL), Background(text URL), Name(varchar), and Animation(text URL). Zenful Features currently

include: animations and music. Similarly to the Journal, Zenful Features are only accessible once a Registered User has logged into Zenful. Upon clicking the Features link on Zenful, a Registered User will be provided with two sections: music, and animations. Under the music section, the Registered User will be provided with options for music to play. Music will be in the relaxing theme, namely: raindrops, lo-fi, and study/meditation sounds. Under the animations section, the Registered User will be provided with an on/off toggle for animations, along with options for animations. If the Registered User chooses to toggle animations on, they will see the animation play on a loop in the corner of their screen on the Journal section.

Relevant Data: music audio files (.mp3), animation video files (.mp4, .mov, .avi), animation image files (.jpg, .png, .jpeg, .gif).

Functional Requirements

Priority 1

Home Page

- 4.1. The home page shall display step-by-step instructions to use Zenful.
- 4.2. The home page shall display log-in and register options for the user.

General User

- 1.1. General users shall be able to view the home page.
- 1.2. General users shall be able to register.

Registered User

- 2.1. Registered users shall be able to login.
- 2.2. Registered users shall have access to Zenful once they are logged in.

Registration

- 5.1. Users shall need to provide their first name during registration.
- 5.2. Users shall need to provide their last name during registration.
- 5.3. Users shall need to provide their email address during registration.
- 5.4. Users shall need to provide their username during registration.
- 5.5. Users shall need to provide their password during registration.
- 5.13. Users shall be able to edit their registration details before submitting the registration form.
- 5.12. Each registered user shall have a unique user identification number once registered.
- 5.6. Users shall be able to see whether their chosen username is available during registration.
- 5.7. Users shall be able to see whether their chosen email is available during registration.
- 5.8. Users shall be able to see whether their confirmed password matches their password.
- 5.9. Users shall need to choose a different email address if the one they provide is unavailable.
- 5.10. Users shall need to choose a different username if the one they provide is unavailable.
- 5.11. Users shall be able to see that their password and confirm password fields must match in order to complete registration.

Login

- 6.1. Registered users shall be able to login.
- 6.2. Registered users shall need to provide their username to login.
- 6.3. Registered users shall need to provide their password to login.
- 6.7. Registered users shall have the option to logout.
- 6.4. Registered users shall be able to select “forgot password” if needed.
- 6.5. Forgot Password: Registered users shall need to enter their email address in order to receive an email with instructions to reset their password.
- 6.8. Registered users shall be able to change their password.

6.9. Registered users shall be able to view and edit their account details upon logging in.

Admin

- 3.3. The admin shall be able to access and modify the database.
- 3.2. The admin shall be provided with full data access.
- 3.1. The admin shall need to login before accessing admin permissions.

Journal

- 7.6. Registered users shall be able to create a new private (accessible only to the registered user and no-one else) journal entry.
- 7.3. Registered users shall be able to edit their private journal entries.
- 7.1. Registered users shall have their private journal entries saved in Zenful for the user's individual viewing.
- 7.2. Registered users shall be able to see all of their existing private journal entries.
- 7.7. Registered users shall be able to delete a private journal entry.

Features

- 9.1. An animation shall be displayed if the animation setting is toggled on by a registered user.
- 9.2. Registered users shall be able to toggle their animation on and off.
- 9.3. Registered users shall be able to choose from available public animations.
- 9.5. Registered users shall be able to play public music through Zenful.
- 9.6. Registered users shall be able to pause currently playing music.

Priority 2

Login

- 6.6. Registered users shall have a signed-in user status.

Journal

- 7.4. Registered users shall be able to search for a private journal entry.
- 7.5. Registered users shall be able to select a private journal entry for editing.

Priority 3

Journal

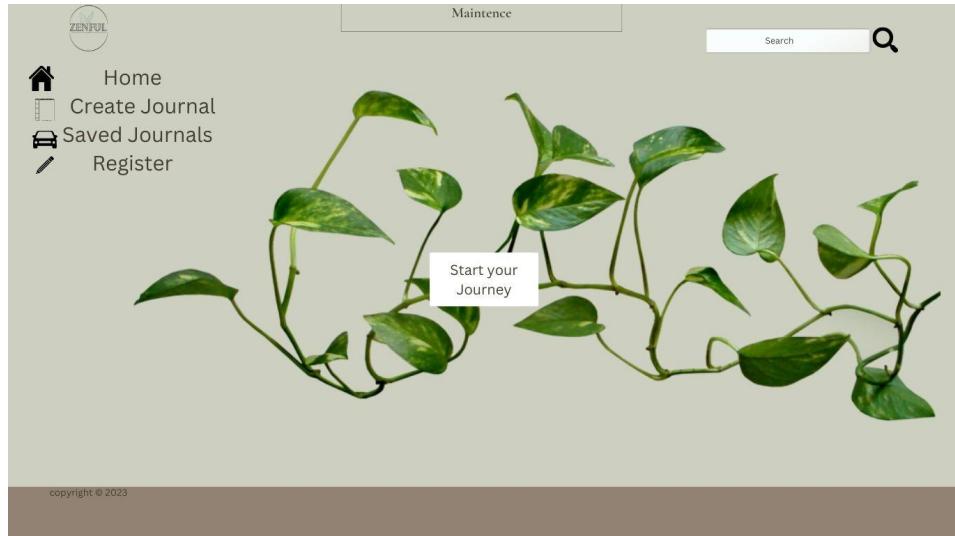
- 7.8. Registered users shall be able to add a 4 digit passcode to their private journal entries for additional security.

Features

- 9.4. Registered users shall be able to upload their own private animations.
- 9.8. Registered users shall be able to upload their own private music.
- 9.9. Registered users shall have their uploaded private music saved for their use only.

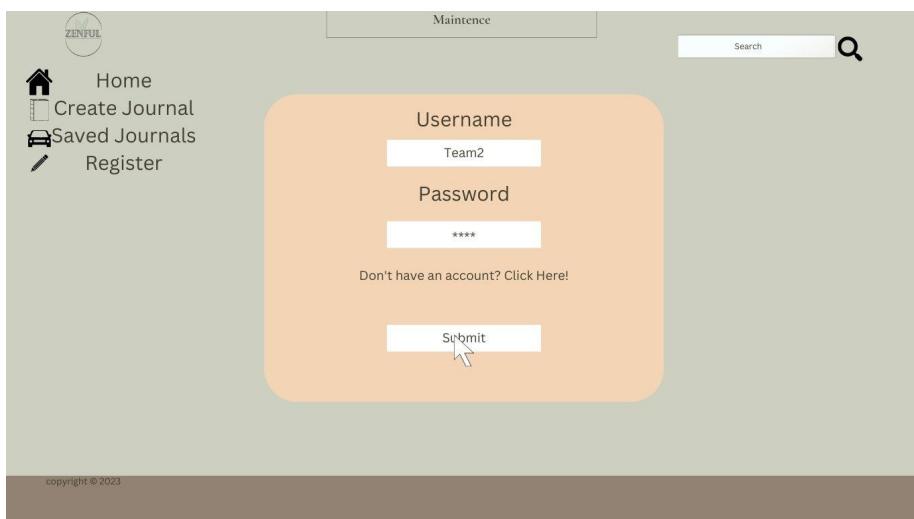
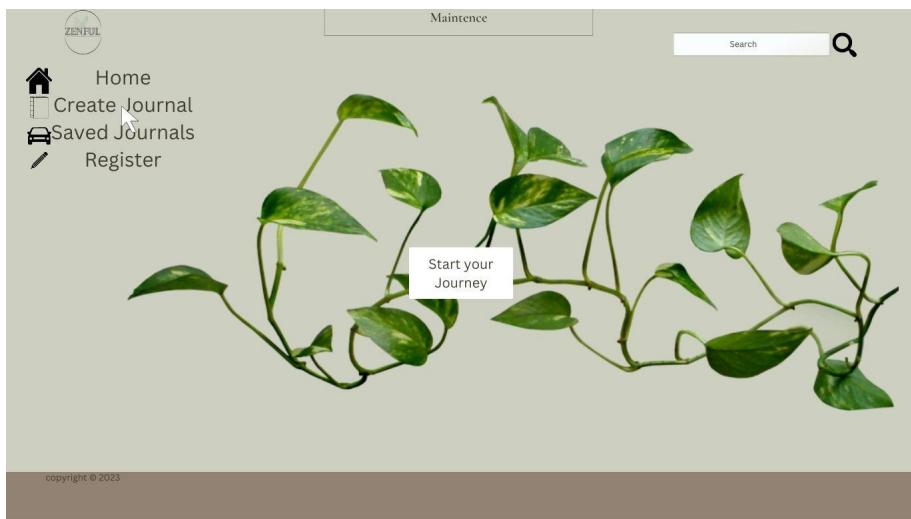
Wireframes Mockups

3.1 Registration

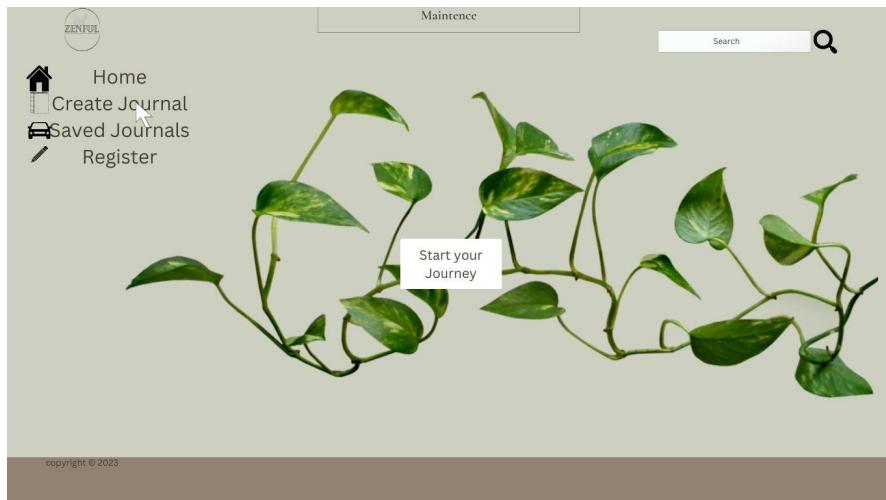


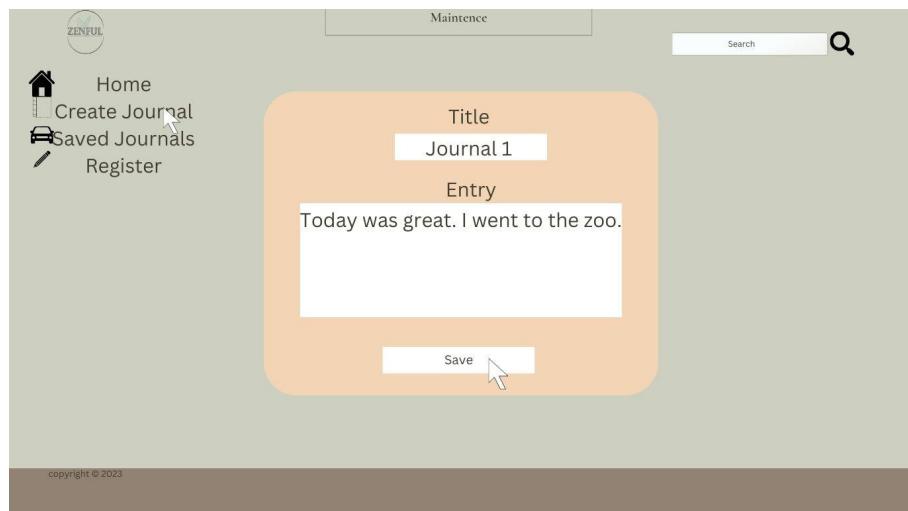
The wireframe shows a top navigation bar with the word "Maintence". On the left is a circular logo with "ZENFUL" text. On the right is a search bar with a magnifying glass icon. Below the navigation is a registration form. The form has an orange rounded rectangle background. It contains fields for "Username" (with value "Team2"), "Email" (with value "Team2@team2.com"), "Password" (with placeholder "****"), and "Confirm Password" (with placeholder "****"). Below these fields are two checkboxes: one for "Are at Least 13 years old" and another for "Terms and Conditions". At the bottom is a "Submit" button with a cursor icon hovering over it. To the left of the form, there is a vertical sidebar with icons and text: a house icon for "Home", a document icon for "Create Journal", a car icon for "Saved Journals", and a pencil icon for "Register". A cursor icon is also visible near the "Register" link. At the bottom left of the page is a small copyright notice: "copyright © 2023".

3.2 Log-in



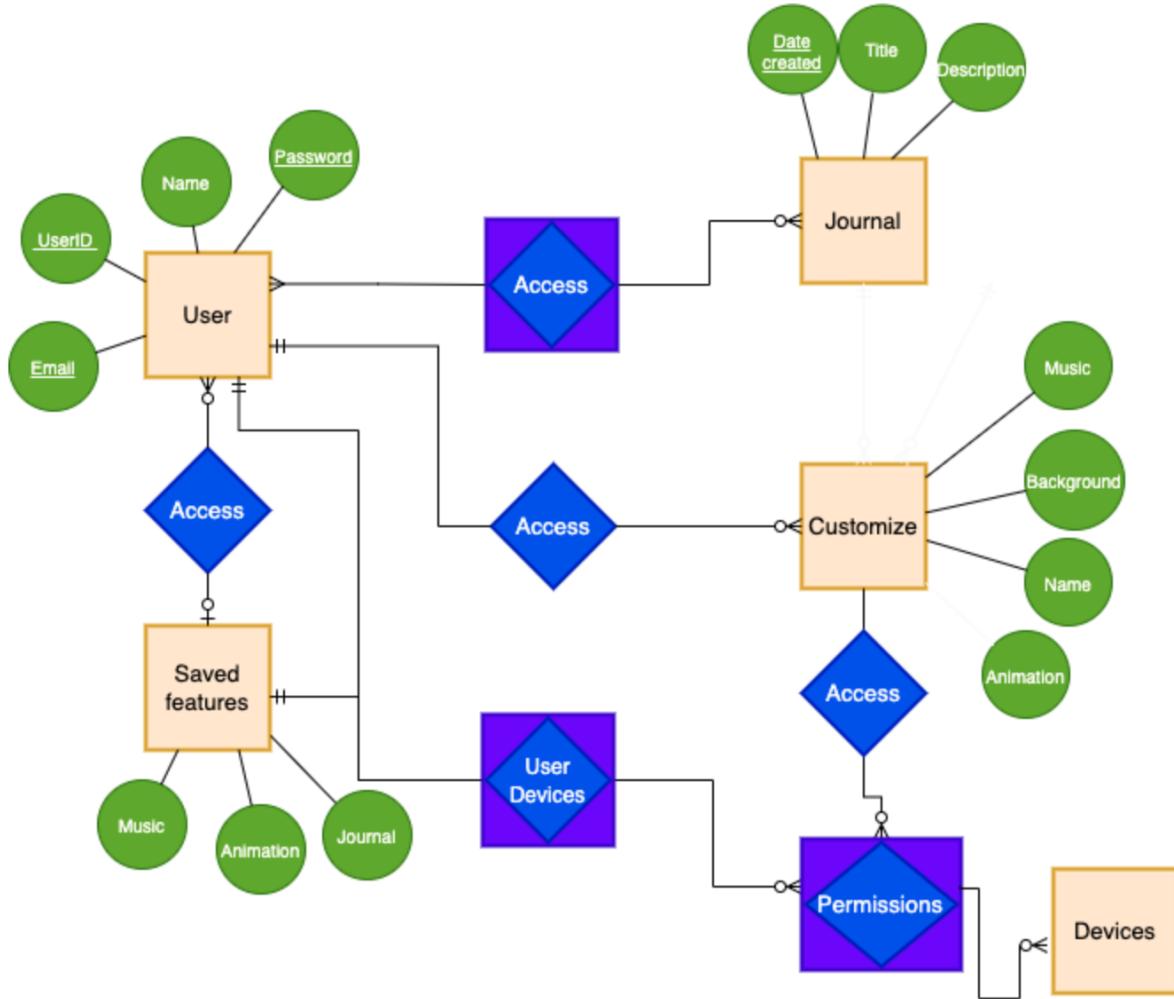
3.3 Starting a Journal



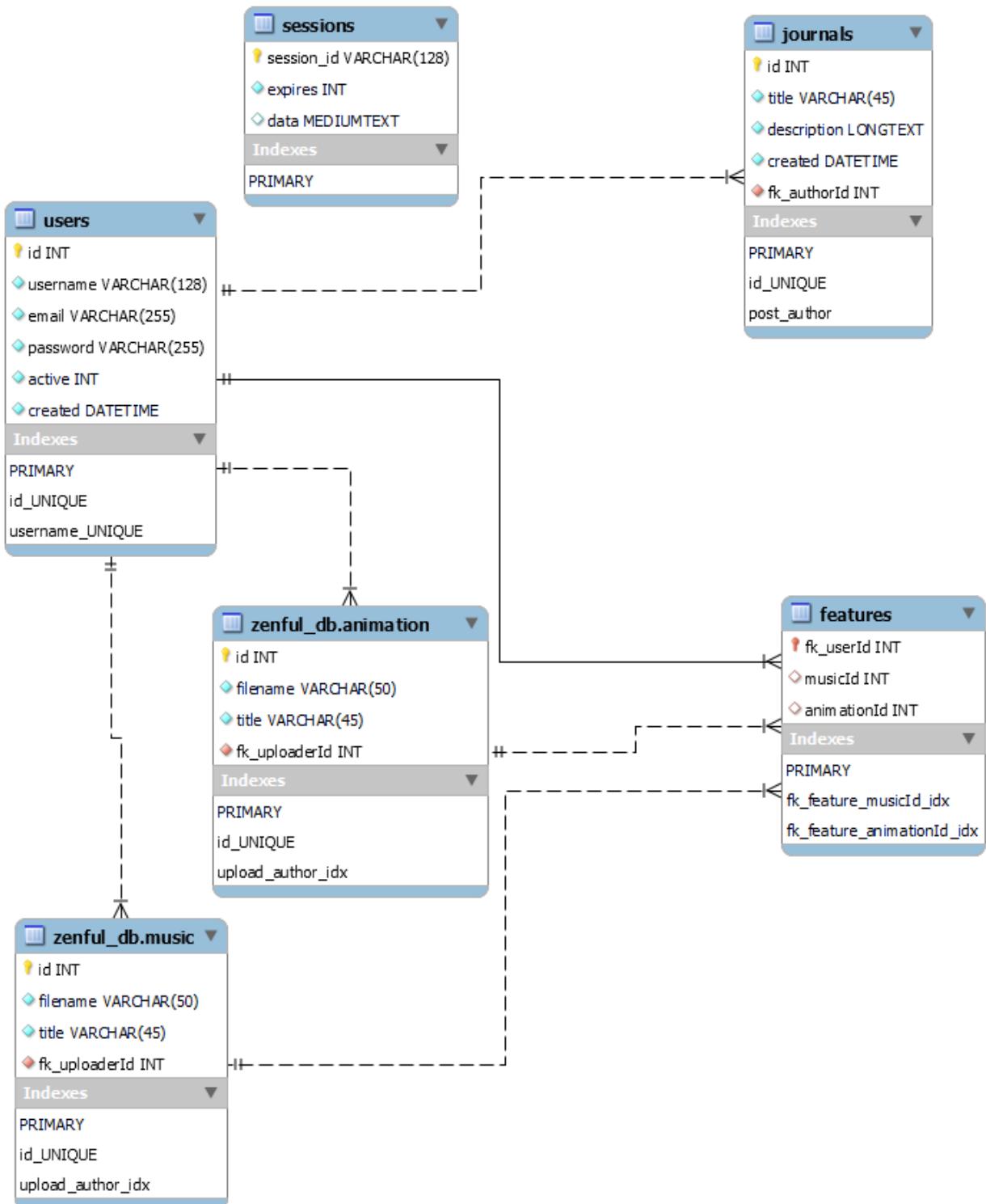


High Level Database Architecture and Organization

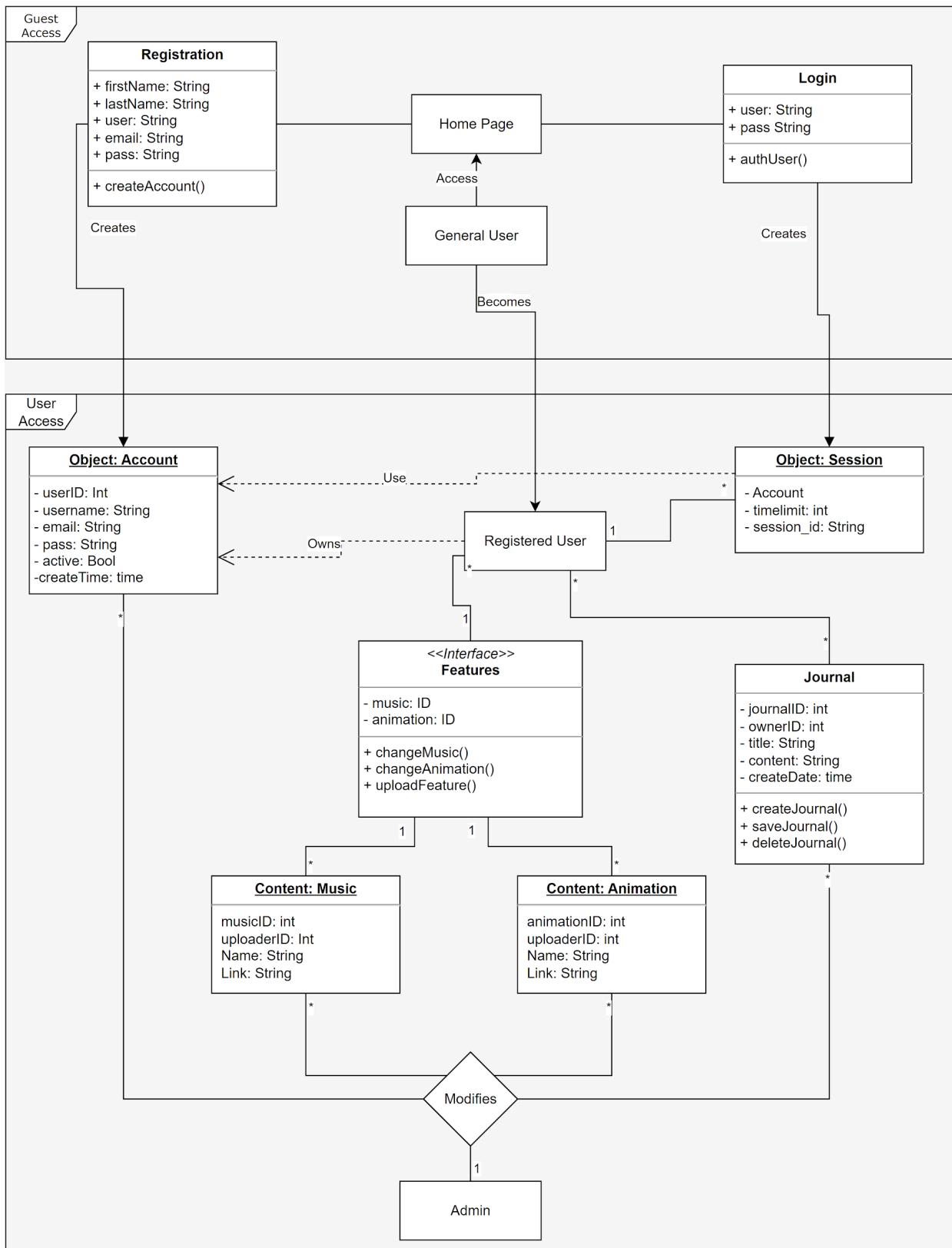
ERD:



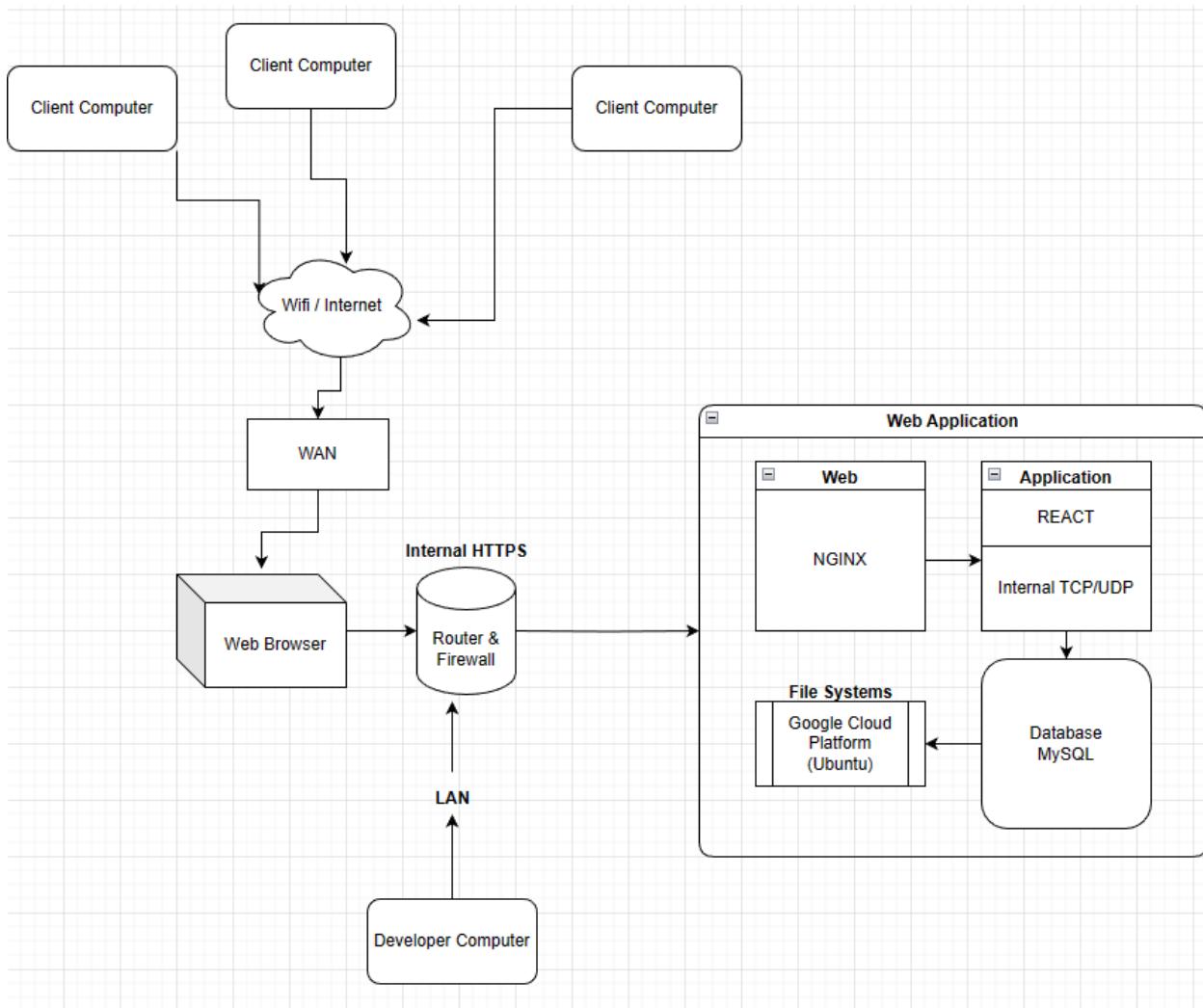
EER:



UML:



Deployment and Network Diagram:



Horizontal Prototype Feedback

- What's the site about? Add to home page, confusion with purpose of site.
- Flash on create journal needs to be moved lower down.
- Re-think search bar. only have it for journal page for user to search for their journal entries(?) maybe ask jose to be sure
- Something to show a user is signed in ex) hello, Jose diff button does not suffice, need something written at the top or something
- When user creates a new journal, route them to saved journals so they can see it.
- Need a hook on home page for what this is, or maybe screenshots of designs of journals so it's clear what it's for before registration.
- View journal not view post.
- Footer not implemented
- Link to terms and conditions
- Maybe change plants or add a hook. Initial thoughts of user: weed!

Product Summary

Our product is Zenful. It is a journaling app with added features, such as music, for a more personal and positive experience. The best way to market this product would be for us to utilize social media. Social media is a great tool for advertising products that have to do with productivity and an app like ours where you can make a journal into anything you need it to be is best promoted on social media. It would start with ads on Tik Tok. This is where you would find most of the users who are looking for something productivity related. The ad would include a sample of the templates we offer as well as all the possibilities of the app. Then we would also do small skits that would get people interested. Some skits include someone wanting to organize their thoughts in one place that they could protect. Another way to market this product would be to market it on Instagram where people are influencers and would need to make journal content ideas, etc. Our main focus will be people who need to be able to write out what they need to get done but we should also focus on people who are going to do content because this is an app with the purpose of letting people decide what they need to write out in an electronic way that is secure.

Home Page

1. The home page shall display step-by-step instructions to use Zenful.

General User

1. General users shall be able to view the home page.
2. General users shall be able to register.

Registered User

1. Registered users shall be able to login.
2. Registered users shall have access to Zenful once they are logged in.

Registration

1. Users shall need to provide their email address during registration.
2. Users shall need to provide their username during registration.
3. Users shall need to provide their password during registration.

Login

1. Registered users shall be able to login.
2. Registered users shall need to provide their username and password to login.
3. Registered users shall have the option to logout.

Admin

1. The admin shall be able to access and modify the database.
2. The admin shall be provided with full data access.
3. The admin shall need to login before accessing admin permissions.

Journal

1. Registered users shall be able to create a new private (accessible only to the registered user and no-one else) journal entry.
2. Registered users shall be able to see all of their existing private journal entries.

Features

1. Registered users shall be able to play public music through Zenful.
2. Registered users shall be able to pause currently playing music.

URL: <http://34.219.66.26/>

Usability Test Plan

1. Journals are the main feature of Zenful, and the main purpose of usage, this along with other features, are to provide a better experience for writing than other sites.

- a. **Journal Creation Steps:**

- i. User registers a new account or logs into an existing one.
- ii. User navigates to sidebar on the left to find the “Create Journal” button.
- iii. User provides a title and description for their new journal entry.
- iv. User clicks submit button to save their new journal entry.

- b. **Journal Viewing Steps:**

- i. User registers a new account or logs into an existing one.
- ii. User needs to have created at least one journal entry in order to view existing journals.
- iii. User navigates to sidebar on the left to find the “Saved Journals” button.
- iv. User views their journals.

- c. **Journal Management Steps:**

- i. Users must be registered and logged in, access to task is from a selected journal on the Saved Journals page, journal must be modified and saved.

- d. **Journal Deletion Steps:**

- i. User registers a new account or logs into an existing one.
- ii. User needs to have created at least one journal entry in order to delete a journal entry.
- iii. User navigates to sidebar on the left to find the “Saved Journals” button.
- iv. User clicks on “view journal” on one of their journals.
- v. User clicks “delete” button.
- vi. Journal is removed.

2. Music is being played in the background of the user’s choosing, to keep them focused and motivated in the writing process, enhancing the experience.

- a. **Music Toggle Steps:**

- i. User registers a new account or logs into an existing one.
- ii. User navigates to the bottom of the page.
- iii. User clicks play button to toggle music on.
- iv. User has the ability to click pause to toggle music off.

- b. **Music Selection Steps:**

- i. User registers a new account or logs into an existing one.
- ii. User navigates to the bottom of the page.
- iii. User clicks next and previous buttons to switch the current song.

3. Animation serves as a simulant when compared to a static background, keeping users interested in the journaling process and in turn stay on the site.

- a. **Animation Toggle Steps:**

- i. User registers a new account or logs into an existing one.
 - ii. User navigates to the bottom of the page.
 - iii. User clicks on button to turn on animation.
 - iv. User clicks off button to turn off animation.
- b. Animation Selection:**
- i. User registers a new account or logs into an existing one.
 - ii. User navigates to the bottom of the page.
 - iii. User clicks next and previous buttons to switch animations.
4. Users can upload music and animations to better fit their interests, and the customizability gives individual users a sense of value.
- a. Upload Music:Users must be registered and logged in (**TODO**)
 - b. Upload Animation:Users must be registered and logged in (**TODO**)
5. Once a user has accumulated enough usage on Zenful, they will need a search function to find the specific journal entry they are looking for, or to use their own uploaded music or animation.
- a. Date Search Steps:**
- i. User registers a new account or logs into an existing one.
 - ii. User navigates to their “Saved Journals”
 - iii. User searches a date ex) “August 2” in the search bar at the top right corner of the page.
 - iv. User sees results for journals made on “August 2nd”.
- b. Title Search Steps:**
- i. User registers a new account or logs into an existing one.
 - ii. User navigates to their “Saved Journals”
 - iii. User searches a title ex) “Finals Week” in the search bar at the top right corner of the page.
 - iv. User sees results for journals with the title “Finals Week”.
- c. Content Search Steps:**
- i. User registers a new account or logs into an existing one.
 - ii. User navigates to their “Saved Journals”
 - iii. User searches a description ex) “Stressful” in the search bar at the top right corner of the page.
 - iv. User sees results for journals containing the word “Stressful” in the description.

Usability effectiveness & efficiency table

| Test/use case | % completed | errors | comments |
|------------------|-------------|--------|-----------|
| Journal creation | 100% | none | completed |

| | | | |
|---------------------|------|---------------------|--|
| Journal viewing | 100% | none | completed |
| Journal management | 100% | none | completed |
| Journal deletion | 0% | Not implemented yet | Button and deletion functionality has not been added yet |
| Music toggle | 100% | none | completed |
| Music selection | 100% | none | completed |
| Animation toggle | 0% | Not implemented yet | Animation section and functionality has not been added yet |
| Animation selection | 0% | Not implemented yet | Animation section and functionality has not been added yet |
| Upload music | 0% | Not implemented yet | Uploading music functionality has not been added yet |
| Upload animation | 0% | Not implemented yet | Uploading animation functionality has not been added yet |
| Date search | 50% | No visible errors | Everything seems to be working in the back-end, but not showing up in the front-end yet. There might be validation logic that deals with displaying user's journals that is conflicting with the search function as a whole. |
| Title search | 50% | No visible errors | Everything seems to be working in the back-end, but not showing up in the front-end yet. There might be validation |

| | | | |
|----------------|-----|-------------------|--|
| | | | logic that deals with displaying user's journals that is conflicting with the search function as a whole. |
| Content search | 50% | No visible errors | Everything seems to be working in the back-end, but not showing up in the front-end yet. There might be validation logic that deals with displaying user's journals that is conflicting with the search function as a whole. |

QUESTIONNAIRE:

<https://docs.google.com/forms/d/12SDDy8Zm-JNKMNsIP8p--L-R9HINp7jworVlhE0ZXvg/edit>

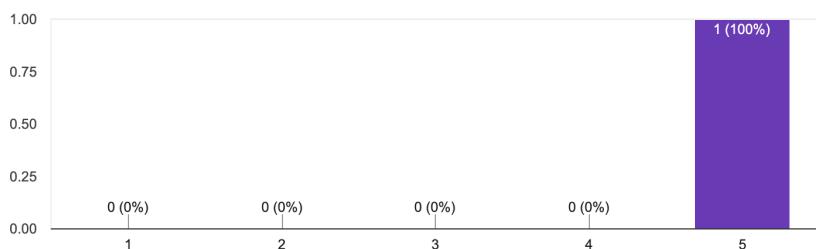
Responses:

Journals

I was able to create a new journal entries.

1 response

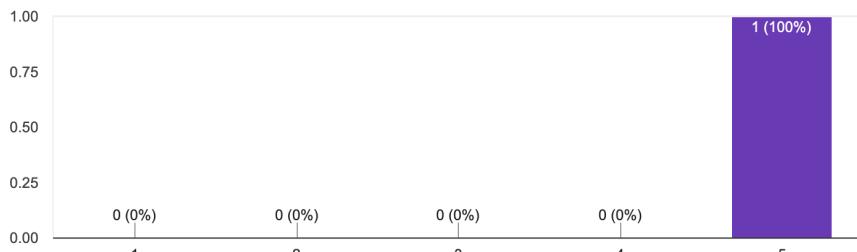
 Copy



I was able to view my previously created journal entries.

1 response

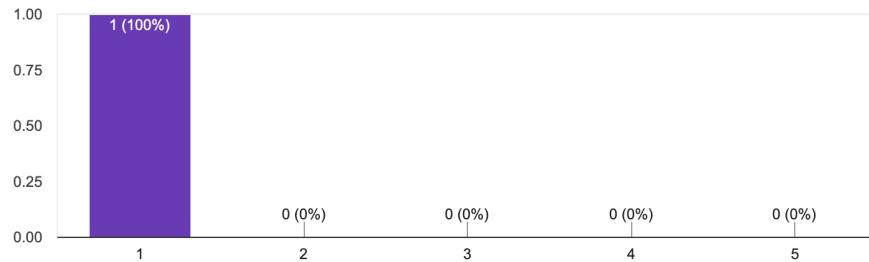
 Copy



I was able to modify my existing journal entries.

 Copy

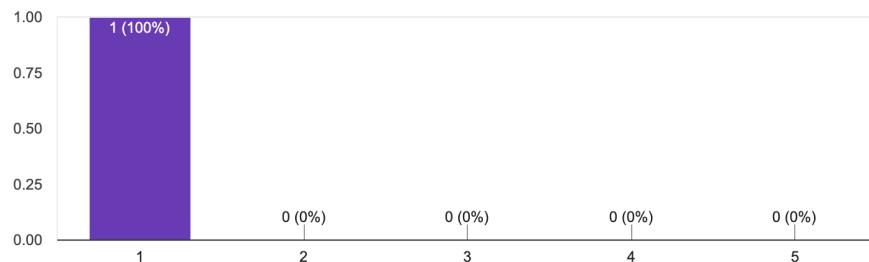
1 response



I was able to delete my journal entries.

 Copy

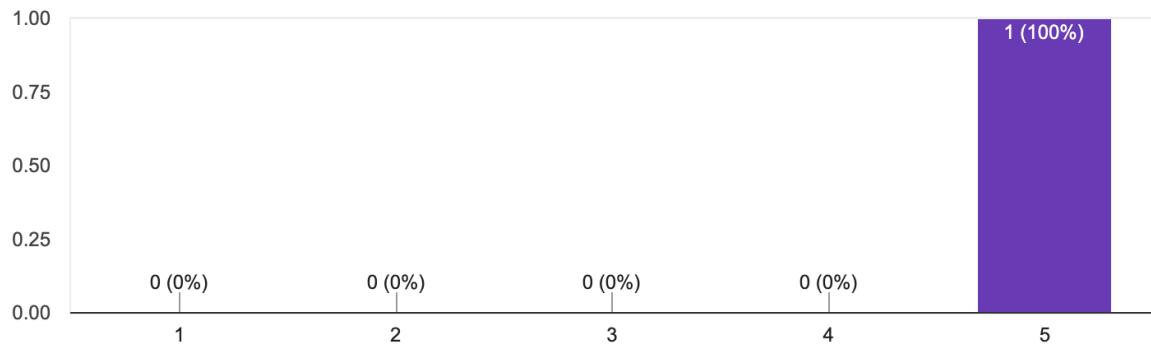
1 response



I was able to toggle the music that is playing.

 Copy

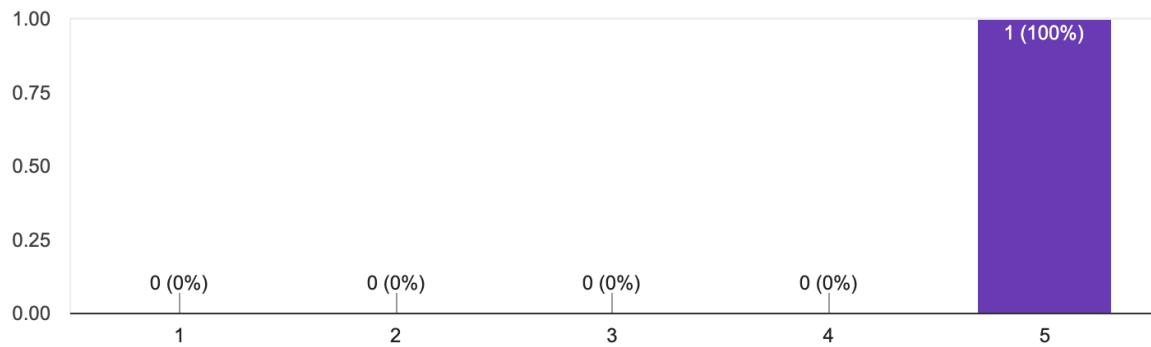
1 response



I was able to change the music that is playing.

 Copy

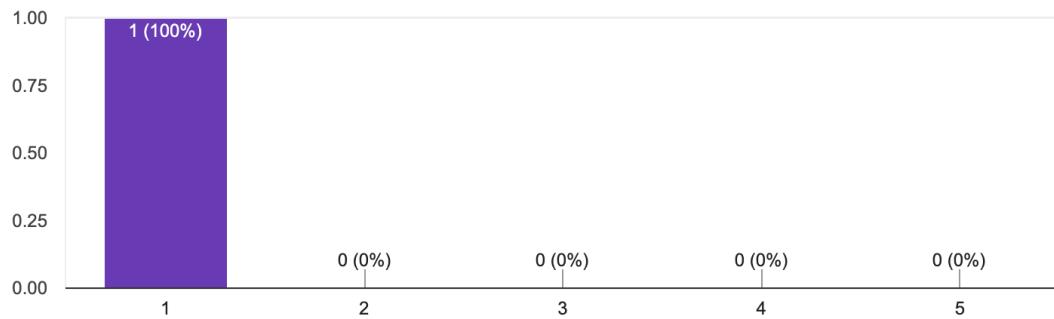
1 response



I was able to toggle the animation playing in the background.

 Copy

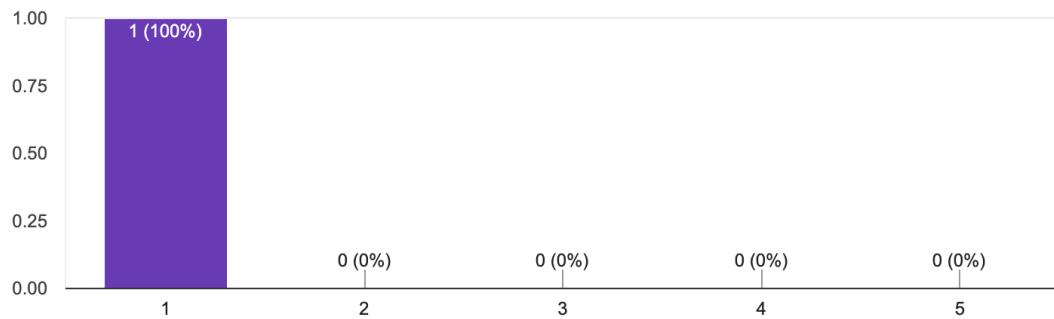
1 response



I was able to change the animation that is playing.

 Copy

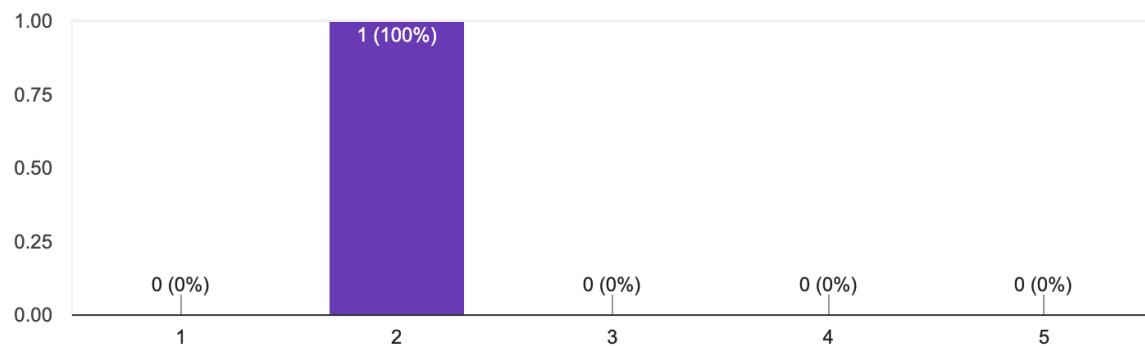
1 response



I was able to upload a music track to use.

 Copy

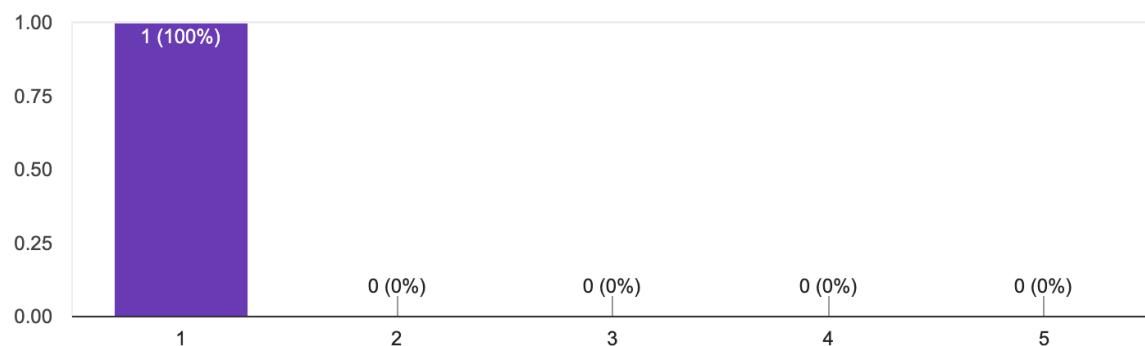
1 response



I was able to upload an animated file to use.

 Copy

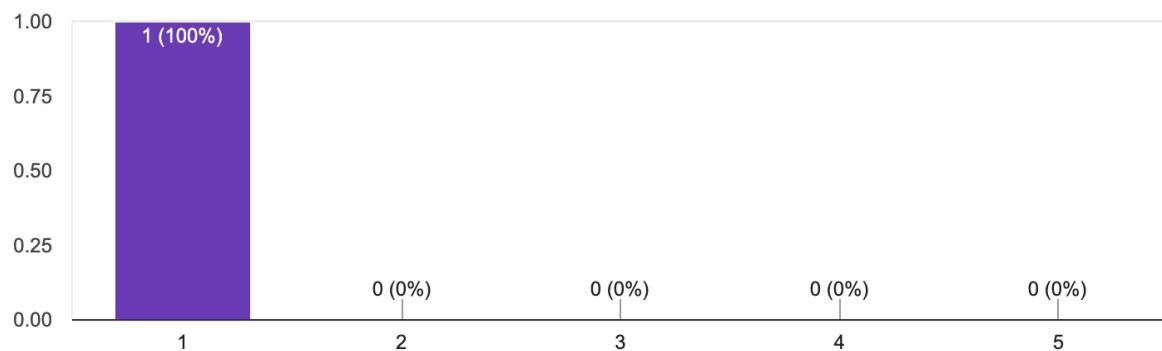
1 response



I was able to search for journals by the contents in it.

 Copy

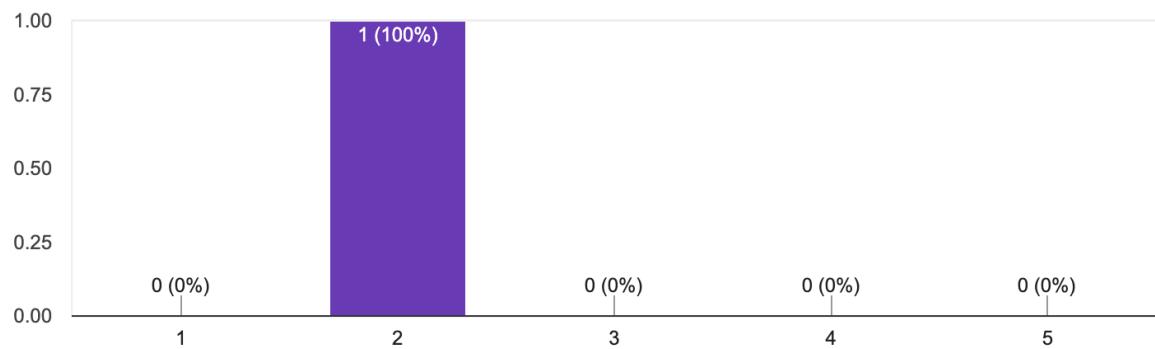
1 response



I was able to search for journals by the date.

 Copy

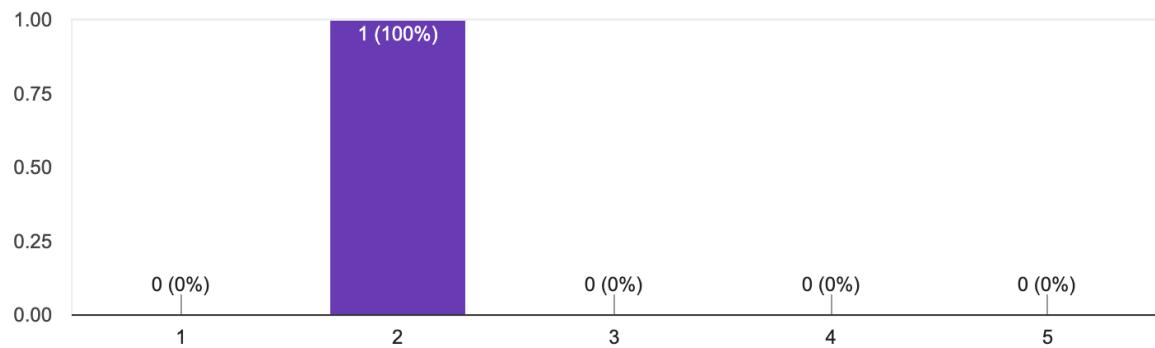
1 response



I was able to search for journals by the title.

 Copy

1 response



QA Test Plan

Test objectives:

System Requirements

HW and SW setup:

HW: Macbook(macOS 13), Windows(11), Iphone 11

SW: Chrome(16), Safari(16)

URL: <http://34.219.66.26/>

Feature to be tested - Check Basic Loading Latency, Check Heavy Loading Latency, Check Upload Latency, Check Creation Latency, and Check Deletion Latency.

QA Test plan:

| System Requirements | Description | Test Input | Expectations | Result |
|---------------------|-----------------------------|--|---|--------|
| #1 | Check Basic Loading Latency | Load Created Journals with 0 journals. | Load completes within expected timeframe. | PASS |
| #2 | Check Heavy Loading Latency | Load Created Journals with 100 journals | Load completes within expected timeframe. | PASS |
| #3 | Check Upload Latency. | Upload music file and animation file of same size. | Upload completes within expected timeframe. | PASS |
| #4 | Check Creation Latency | Create a new user account and journal entry from that account. | Creation completes within expected timeframe. | PASS |
| #5 | Check Deletion Latency | Delete journal entries and then the user account | Deletion completes within expected timeframe. | FAIL |

Test objectives:

Coding Standards

HW and SW setup:

HW: Macbook(macOS 13), Windows(11), Iphone 11

SW: Chrome(16), Safari(16)

URL: <http://34.219.66.26/>

Feature to be tested - Check Code Documentation, Check Code Warnings, Check Code Utilization, Check Database Utilization, Check Database Warnings

QA Test plan:

| Coding Standard s | Description | Test Input | Expectations | Result |
|-------------------|----------------------------|------------|---|--------|
| #1 | Check Code Documentation | N/A | Frontend and Backend is documented and understandable | PASS |
| #2 | Check Code Warnings | N/A | No formatting errors detected | PASS |
| #3 | Check Code Utilization | N/A | No redundant code utilized | PASS |
| #4 | Check Database Utilization | N/A | No redundant tables utilized. | PASS |
| #5 | Check Database Warnings | N/A | No query errors detected | PASS |

Test objectives:

Security

HW and SW setup:

HW: Macbook(macOS 13), Windows(11), Iphone 11

SW: Chrome(16), Safari(16)

URL: <http://34.219.66.26/>

Feature to be tested - Access Unauthorized Journals, Access Journals without Account, Access Own User Profile, Access Own Journals, Access Deleted Journals

QA Test plan:

| Security | Description | Test Input | Expectations | Result |
|----------|---------------------------------|--|--------------------------------|--------|
| #1 | Access Unauthorized Journals | Load journals page with random ID while signed in. | Redirected to Created Journals | PASS |
| #2 | Access Journals without Account | Load random journals as an guest | Redirected to Login Page | FAIL |
| #3 | Access Own User Profile | Load user profile page while signed in. | Directed to own profile | PASS |
| #4 | Access Own Journals | Load journals from the created journals page. | Directed to Journal | PASS |
| #5 | Access Deleted Journals | Load journals from history after deletion. | Redirected to Created Journals | FAIL |

Test objectives:

User Data

HW and SW setup:

HW: Macbook(macOS 13), Windows(11), Iphone 11

SW: Chrome(16), Safari(16)

URL: <http://34.219.66.26/>

Feature to be tested - Create Journal Entry, Upload Music, Upload Animation, Delete Journal, Delete User Account

QA Test plan:

| User Data | Description | Test Input | Expectations | Result |
|-----------|----------------------|---------------------------------------|--|--------|
| #1 | Create Journal Entry | Randomly generated 500 character text | Journal entry is created and accessible by the creator. | PASS |
| #2 | Upload Music | music.mp3 | Music is saved and usable by the uploader. | FAIL |
| #3 | Upload Animation | animate.mp4 | Animation is saved and usable by the uploader | FAIL |
| #4 | Delete Journal | N/A | Journal entry is deleted and deallocated. | FAIL |
| #5 | Delete User Account | N/A | User account is deleted and all associated data i.e. Journals/Music/ Animation | PASS |

Test objectives:

Environmental

HW and SW setup:

HW: Macbook(macOS 13), Windows(11), Iphone 11

SW: Chrome(16), Safari(16)

URL: <http://34.219.66.26/>

Feature to be tested - Check Website Scaling, Check Browser Compatibility, Check Mobile Compatibility, Check Upload Verification, Check Journal Verification

| Environmental | Description | Test Input | Expectations | Result |
|---------------|-----------------------------|--|---|--------|
| #1 | Check Website Scaling | N/A | Site scales proportionate to display size | PASS |
| #2 | Check Browser Compatibility | N/A | Site loads without issue | PASS |
| #3 | Check Mobile Compatibility | N/A | Site loads without issue | PASS |
| #4 | Check Upload Verification | Upload invalid formats such as music and animation. | Site rejects invalid upload files | PASS |
| #5 | Check Journal Verification | Create new and modify existing journal entries with invalid texts. | Site rejects invalid characters in journals | PASS |

Code Review

Our coding style utilizes the Prettier extension in VSCode to maintain a consistent syntax. Prettier syntax has its own set of rules that include the following: Egyptian brackets, minimal horizontal line lengths by splitting lines up where possible, 2 space indents, vertical indents for logical blocks, functions declared above uses, and else without line breaks. Prettier enforces the majority of these rules by itself, while things such as functions declared above the use of said functions, are a choice by us as a team.

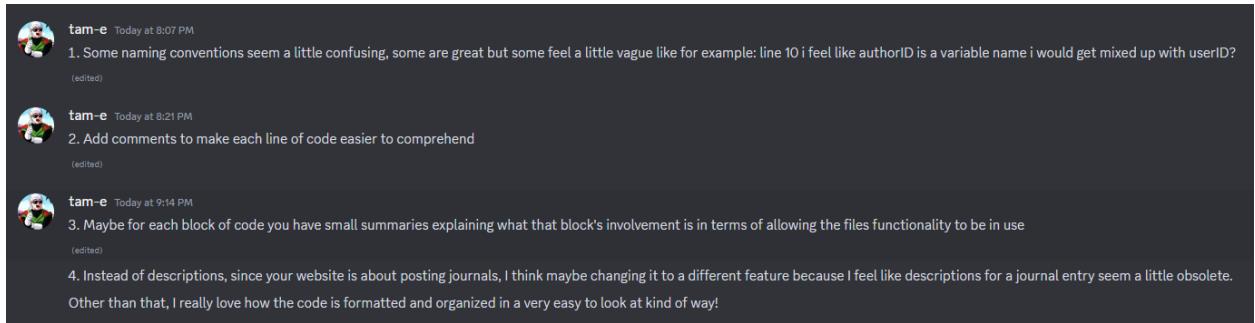
Review sent to other team:

```
1 var db = require("../config/db");
2 const { successPrint } = require("../helpers/debug/debugprinters");
3
4 const PostModel = {};
5
6 PostModel.create = (title, description, fk_authorId) => {
7   let baseSQL =
8     "INSERT INTO journals (title, description, created, fk_authorId) VALUE (?, ?, now(), ?);";
9   return db
10    .execute(baseSQL, [title, description, fk_authorId])
11    .then(([results, fields]) => {
12      return Promise.resolve(results && results.affectedRows);
13    })
14    .catch((err) => Promise.reject(err));
15 };
16
17 PostModel.getNRecentPosts = (numberOfPosts) => {
18   let baseSQL = `SELECT id, title, description, created FROM journals ORDER BY created DESC LIMIT 16`;
19   return db
20    .execute(baseSQL, [numberOfPosts])
21    .then(([results, fields]) => {
22      return Promise.resolve(results);
23    })
24    .catch((err) => Promise.reject(err));
25 };
26
27 PostModel.getJournalById = (journalId) => {
28   let baseSQL = `SELECT u.username, j.title, j.description, j.created
29   FROM users u
30   JOIN journals j
31   ON u.id=j.fk_authorId
32   WHERE j.id=?`;
33
34   return db
35    .execute(baseSQL, [journalId])
36    .then(([results, fields]) => {
37      return Promise.resolve(results);
38    })
39    .catch((err) => Promise.reject(err));
40 };
41
```

```

1  var express = require("express");
2  var router = express.Router();
3  const { errorPrint, successPrint } = require("../helpers/debug/debugprinters");
4  var multer = require("multer");
5  var crypto = require("crypto");
6  var PostError = require("../helpers/error/PostError");
7  var PostModel = require("../models/Posts");
8
9  var storage = multer.diskStorage({
10    destination: function (req, file, cb) {
11      cb(null, "public/images/uploads");
12    },
13    filename: function (req, file, cb) {
14      let fileExt = file.mimetype.split("/")[1];
15      let RandomName = crypto.randomBytes(22).toString("hex");
16      cb(null, `${RandomName}.${fileExt}`);
17    },
18  });
19
20  var uploader = multer({ storage: storage });
21
22  router.post("/createPost", uploader.single("chooseImage"), (req, res, next) => {
23    let title = req.body.title;
24    let description = req.body.description;
25    let fk_userId = req.session.userId;
26
27    return PostModel.create(
28      title,
29      description,
30      fk_userId
31    )
32    .then((postWasCreated) => {
33      if (postWasCreated) {
34        successPrint("if true");
35        req.flash("success", "Your post was created successfully!");
36        res.redirect("/");
37      } else {
38        res.json({
39          status: "OK",
40          message: "post was not created",
41          redirect: "/createjournal",
42        });
43      }
44    );
45  });

```



tam-e Today at 8:07 PM
1. Some naming conventions seem a little confusing, some are great but some feel a little vague like for example: line 10 i feel like authorID is a variable name i would get mixed up with userID?
(edited)

tam-e Today at 8:21 PM
2. Add comments to make each line of code easier to comprehend
(edited)

tam-e Today at 9:14 PM
3. Maybe for each block of code you have small summaries explaining what that block's involvement is in terms of allowing the files functionality to be in use
(edited)

4. Instead of descriptions, since your website is about posting journals, I think maybe changing it to a different feature because I feel like descriptions for a journal entry seem a little obsolete.
Other than that, I really love how the code is formatted and organized in a very easy to look at kind of way!

I chose this code for review because we as a team want to strengthen our functionality with journals as it is one of our main features with our app. And with this review we can apply the suggestions to the rest of our code.

Team member's code for review:

In the code that was sent, I do think that we need to add more comments because of other people in this team looking at the code. This is because we might add things, and not everyone on this team has knowledge on what that certain code does. This way, they will be able to add their own code and not be as confused. I also believe that we should add better naming conventions too because sometimes “post” might not be as clear as “journal”.

```

1  var db = require("../config/db");
2
3  const MusicModel = {};
4
5  MusicModel.create = (title, description, fk_authorId) => {
6    let baseSQL =
7      "INSERT INTO journals (title, description, created, fk_authorId) VALUE (?, ?, now(), ?);";
8    return db
9      .execute(baseSQL, [title, description, fk_authorId])
10     .then(([results, fields]) => {
11       return Promise.resolve(results && results.affectedRows);
12     })
13     .catch((err) => Promise.reject(err));
14   };
15
16 MusicModel.search = (searchTerm, userId) => [
17   // created FROM journals WHERE fk_authorId = ? AND (title LIKE ? OR description LIKE ?);
18   [
19     let baseSQL =
20       "SELECT id, title, description, concat_ws(' ', title, description) AS haystack \
21         FROM journals \
22         having haystack like ?;";
23     let sqlReadySearchTerm = "%" + searchTerm + "%";
24     return db
25       .execute(baseSQL, [userId, sqlReadySearchTerm])
26       .then(([results, fields]) => {
27         return Promise.resolve(results);
28       })
29       .catch((err) => Promise.reject(err));
30   ];
31
32 MusicModel.getMusicById = (journalId) => {
33   let baseSQL = `SELECT u.username, j.title, j.description, j.created
34   FROM users u
35   JOIN music m
36   ON u.id=j.fk_authorId
37   WHERE j.id=?`;
38
39   return db
40     .execute(baseSQL, [journalId])
41     .then(([results, fields]) => {
42       return Promise.resolve(results);
43     })
44     .catch((err) => Promise.reject(err));
45   };
46

```

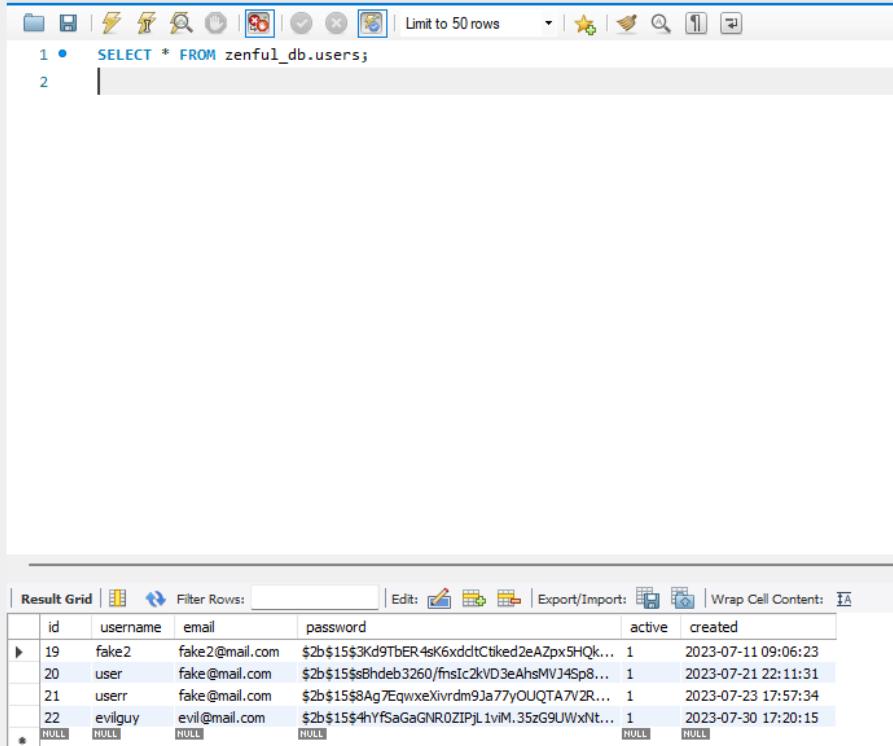
Review notes:

- Need to add comments for functionality so that readers can comprehend them.
- SQL statements are referencing the journals table in certain cases when they should be referencing the music table.
- Last change is that I would change fk_authorId to fk_userId as the user poster may not be the author

Best Practices for Security

Assets To Protect:

- Passwords
- Journals (People could put very personal or identifiable information there)
- Database (Using Bcrypt, when a user is creating an account the password is automatically hashed when being inserted into the database).



The screenshot shows a MySQL Workbench interface. At the top, there's a toolbar with various icons. Below it is a query editor window containing the following SQL code:

```
1 •  SELECT * FROM zenful_db.users;
```

Below the query editor is a results grid titled "Result Grid". The grid has columns: id, username, email, password, active, and created. It displays five rows of data:

| | id | username | email | password | active | created |
|---|------|----------|----------------|--|--------|---------------------|
| ▶ | 19 | fake2 | fake2@mail.com | \$2b\$15\$Kd9TbER4sK6xdctCtiked2eAZpx5HQk... | 1 | 2023-07-11 09:06:23 |
| | 20 | user | fake@mail.com | \$2b\$15\$shdeb3260/fnsLc2kVD3eAhsMVJ4Sp... | 1 | 2023-07-21 22:11:31 |
| | 21 | userr | fake@mail.com | \$2b\$15\$8Ag7EqwxeXivrdm9Ja77yOUQTA7V2R... | 1 | 2023-07-23 17:57:34 |
| | 22 | evilguy | evil@mail.com | \$2b\$15\$4hYfsaGaGNR0ZPj1viM.35zG9UWxNt... | 1 | 2023-07-30 17:20:15 |
| * | NULL | NULL | NULL | NULL | NULL | NULL |

Data Validation Include:

- Username when registering (Regex: not digit, is alphanumeric character, min of 2 characters long) `^[\D\w{2,}]$`
- Email exists when registering
- Password exists when registering (No regex as of yet, password is encrypted when saved to Database using bcrypt)
- Username and Password are of an existing user when logging in and that they match.
- User must be logged in before they can create journal entries.
- Search: Checks if there are journal entries that exist to search, contents of search match a journal entry (Date, Title).

Self-Check: Non-Functional Specs

1. System Requirements

- 1.1. Web app should be available on personal computers and laptops. **DONE**
- 1.2. Web app should be able to service a few thousand clients. **DONE**
- 1.3. Will take about 100 GB of bandwidth a month. **DONE**
- 1.6. Web app will be easy for anyone to use. **DONE**
- 1.7. If there is a delay on using site user should only experience a few seconds delay. **DONE**
- 1.8. There should be enough storage for users to upload music to the website. **ISSUE: Music cannot be upload, because it is given**
- 1.9. There should be enough storage for users to have hundreds of notes and or journal pages. **DONE**
- 1.12. If there is maintenance there will be a banner on the website for all to see. **DONE**

2. Coding Standards

- 2.1. All code is written in JavaScript. **DONE**
- 2.2. All code will have notes attached for easy maintenance on scheduled days. **DONE**

3. Content and Appearance

- 3.1. Landing page should have a neutral background. **DONE**
- 3.2. The forefront of the page should be a highlight reel for the sites features. **ISSUE: Made guest page instead**
- 3.3. There will be the site name and or logo in left hand upper corner. **DONE**
- 3.4. Middle top will have maintenance schedule. **DONE**
- 3.5. The upper right hand side should have the login/signup option. **ISSUE: We decided to have login/signup on the left sidebar of our web-app.**
- 3.6. The highlight reel will be in the middle of the page. **ISSUE: Made guest page instead**

3.7. Site will have a neutral and cool toned theme for the website until the user chooses their own theme for their usability. **DONE**

3.8. There will also be a setting for light or dark. **ISSUE: No time to implement advanced features, might do it in future**

3.9. There will be an option for logging in on the upper right hand side of the page with links to account settings. **ISSUE: We decided to have login/signup on the left sidebar of our web-app.**

3.10. Under the logo will be the header on the left hand side: Notes, Journals, Music, Avatar, Themes, Account. **ISSUE: Only journals, accounts and music work together, while others not implemented**

3.11. On the bottom of the page shall be links to about us,faqs,contact us,terms of use, privacy policy. **DONE**

3.12. The side should have a menu option to flip between the music section and the current page. **DONE**

3.13. The companies information: name, address, establishment, last time site was updated should be on the bottom of the page. **DONE**

3.14. Above companies information should be the Faq,About Us, Contact Us, Terms of use,Privacy Policy. **DONE**

3.15. When in the notes tab, there will be title of note, date, saving function on the left hand side. **DONE**

3.16. The title and date will be larger than the saving function. **ISSUE: Note is saved automatically, no need for a saving button.**

3.17. Under the title of the note will be the file options, edit, insert, format, and help. **ISSUE: No time to implement advanced features, could do it in future**

3.18. File options, edit, insert,format, and help should be slightly smaller than the title and date. **ISSUE: No time to implement advanced features, could do it in future**

3.19. Under file options should be the Font, Zoom, Text Style, Text Size, Highlighter, Template, and Print function. **ISSUE: No time to implement advanced features, could do it in future**

3.20. On the left hand side there will be a area where the titles of notes are listed. **ISSUE: No time to implement, might do it in future**

3.21. On the right hand side there will be the Account function. **ISSUE: We decided to have all functions on the left sidebar.**

3.22. The header from the dashboard will be visible in the middle of the top of the page. **DONE**

3.23. At the bottom of the page is the links to About Us, Faqs, Contact Us, Terms of Use, Privacy Policy. The company information and last update of site/feature. **DONE**

3.24. In the bottom right should be the animated gif/animation. **ISSUE: No time, might in the future**

3.25. The music player should be in the bottom left. **ISSUE: Upon discussion we decided to have the music player at the bottom rather than bottom left.**

3.26. The background should be user pick and in the foreground should be the ‘paper’ the user is typing on. **ISSUE: Background needs to be picked by our software engineering team rather than the user.**

3.27. When in the journaling tab, there will be a date at the top left hand side. **ISSUE: Date is given in every journal under title**

3.28. There will be the choice to add a title or leave just the date. **ISSUE: Date is given in every journal under title**

3.29. There will be the files, insert, format, edit, and help functions under the date. **ISSUE: No time to implement edit, files, insert, format, and help functions.**

3.30. The date should be larger than the files, insert , format, edit, and help functions. **ISSUE: We are currently implementing edit, no time to implement files, insert, format, and help functions.**

3.31.Under files, insert, format, edit, and help should be font, font size, zoom, and print function. **ISSUE: No time to implement all of these advanced features, could do it in future**

3.32. On the right hand side should be the account function as well as the passcode toggle. **ISSUE: We decided to have all of our functions on the left sidebar.**

3.33. At the bottom of the page should be the About Us, Faqs, Contact Us, Terms of Use, Privacy Policy, Company Information and last update of site/feature. **DONE**

3.34. In the bottom right should be the animated gif/animation. **ISSUE: no time to implement, could do it in future**

3.35. The music player should be in the bottom left of the screen. **ISSUE: Upon discussion we decided to have the music player at the bottom rather than bottom left.**

3.36. The background should be the users pick. **ISSUE: Background needs to be picked by our software engineering team rather than the user.**

3.37. In the foreground should be the ‘paper’ the user is typing on. **ISSUE: Currently we just have a form for the user to type into rather than a paper background image.**

3.38. When in the music tab there should be a mini player in the top left hand side. **ISSUE: Upon discussion we decided to have the music player at the bottom rather than bottom left.**

3.39. It should have a play, pause, and back and forth function. **DONE**

3.40. On the top left under the player should be the functions: Playlist(s), Browse, Just for You, Trending. **ISSUE: No time to implement these advanced features, could do it in future**

3.41. On the main page should be the pages that are toggled. **DONE**

3.42. The page should show the background that the user has chosen. **ISSUE: Background needs to be picked by our software engineering team rather than the user.**

3.43. At the bottom of the page should be the About Us, Faqs, Contact Us, Terms of Use, Privacy Policy, Company Information and last update of site/feature. **DONE**

3.44. In the right hand side should be the account function in the upper right. **ISSUE: We decided to have all of our functions in the left sidebar.**

3.45. In the bottom right should be the animated gif/animation. **ISSUE: Not implemented, could do it in the future**

4. Security

4.1 The user will need to login in order to access any of the files on the server. **DONE**

4.2. This will enable them to ensure their information is kept private and not open for strangers to engage with their work. **DONE**

4.3. When toggled on the user will be prompted to enter a four number code for their passcode. **ISSUE: No time to implement this functionality, could do it in the future**

4.4. Users data should be stored on site storage. **DONE**

5. Marketing

5.1. To market this product we will provide information on the key aspects of the site such as the ability to have a little animated friend that will help keep users motivated to use the app. **DONE**

5.2. We will also tote about how you can add your music to the library and ensure that they can customize the page in preset themes. **ISSUE: Music works, however, may not have the time to have the page be customizable with preset themes.**

5.3. The big point is that the app is able to have music, an animated gif or animation in the corner, customizable page themes, journal and note taking and the ability to passcode notes.

ISSUE: Music (DONE), animation (ISSUE: animations not implemented), customizable page themes (ISSUE: may not have the time), journal (DONE), passcode (ISSUE: may not have the time).

6. User Data

6.1 User data that will be collected is as follows: music genres liked by user, user age, user gender. **ISSUE: We are saving usernames, emails, and passwords only for the user entity.**

6.2. This information is gathered to help further the understanding of what the users enjoy and what can be added in the future for more user engagement. **ISSUE: We are saving usernames, emails, and passwords only for the user entity.**

6.3. This information is only used for the furthering of user enjoyment and is not sold to any outside parties. **ISSUE: We are saving usernames, emails, and passwords only for the user entity.**

6.4. User information is tracked through cookies. **DONE**

6.5. User's have all rights to their privacy information and can opt to decline accepting trackers. **ISSUE: Users have a checkbox for Terms of Service and Privacy Policy. They can either accept it and create their account, or decline it, and not create an account.**

6.6. If a user decides to decline having their information stored, their information will be placed on a list of do not track. **ISSUE: Users have a checkbox for Terms of Service and Privacy Policy. They can either accept it and create their account, or decline it, and not create an account.**

6.7. Any change to the privacy policy will be sent out in an email to all users for them to accept again and or decline. **ISSUE: This functionality is important, but might need to be implemented in the future, as we are presently preoccupied with urgent functionalities.**

6.8. As well as a notice placed on the website header. **DONE**

6.9. Last update to privacy policy will be placed at the bottom of the page. **ISSUE: This functionality is important, but might need to be implemented in the future, as we are presently preoccupied with urgent functionalities.**

6.10. The user has the right to request a copy of what information we have collected from them at any time. **ISSUE: This functionality is important, but might need to be implemented in the future, as we are presently preoccupied with urgent functionalities.**

6.11. Registered users will be able to accept or decline terms of conditions upon registering. **DONE**

6.12. Users will have the right to retain the information and usage data. **DONE**

6.13. User shall have access to privacy policy upon registration **DONE**

6.14. User will have access to privacy policy in a link at the footer of the page. **DONE**

7. Performance and Scalability

7.1. Web app will be reliable at all times and few crashes. **DONE**

7.2. There will be a fast and efficient switch between page options. **DONE**

7.3. Web app should be able to handle 10 times its capacity. **DONE**

7.4. The website should be up at all times unless there is maintenance. **DONE**

7.5 Peak loads are expected to be about 5000 users. **DONE**

8. Environmental

8.1. The environment should be compatible with web browsers of all kinds. **DONE**

8.2. The environment should support a filter with dates for journal/to-do list **ISSUE: Search bar works but does not filter out all journals**

8.3. The environment should support a filter with animation/sounds **DONE**

8.4. The environment should support a stable internet connection. **DONE**

Post Analysis

This Software Engineering class has been, by far, the most enriching experience as a Computer Science student and as a person. Everything was real and raw, not so much a class where assignments are “due” and you turn them in and call it a day. This was more so going through the mechanics of a fully functional team, with different skills and personalities, ultimately to deliver a product for a client under a specific deadline. Having taken it in the summer, I think the experience was more raw because of the stricter deadlines, as well as the pressure further highlighting everyone’s true personalities and set of skills.

Our team has dealt with many difficulties. These include: M2, schedule conflicts, as well as the set of skills. It mixed in to create a cocktail that was less than ideal, but nevertheless, everyone pushed through to the best of their abilities. M2 was distressing because as a team, we had to deliver a product and we were not ready for it. There was a lack of communication, and a lack of urgency. The personal deadline was not met, and there was no communication to sense where some even were. We had a great idea, but when the team is not harmonic, the product will not be met. As a team lead, I had to send distressing messages to everyone because things were not being done, nor were they being communicated about. That night, everyone could clearly see that if we did not pick it up, we could fail. If we are talking about a real-case scenario, it would be put at the same urgency as: our client would not be happy, there would have to be some terminations, and or a loss of money. We solved this through reminding ourselves of the consequences. We solved this also through an improved way of communication, because it should be common courtesy that if you are not finished with something, to alert the team so that we know how to tackle the delayed task. There was a lack of sleep that night and we did what we had to do. An issue I’ve stated above was also schedules. Everyone had conflicting life schedules and it was not ideal in terms of meeting/communicating. The conflicting life schedules also made some complete tasks a little later than the personal set deadline. This was, again, less than ideal, but we solved this by meeting at a specific time 2 times a week with a summarized agenda I would create, and I would send summarized daily messages at 6am so that everyone would see it before their jobs. Furthermore, another conflict we had was our skill set. There were people in this group that were not familiar with any coding language, as well as some having no knowledge in databases, along with other very vital Software concepts. Adding on, none of us knew how to deploy. This set the team back in terms of time, and also quality. We solved this by teaching everyone eachother’s tasks, with demonstrations and explanations. Others did what they could in the time that they had with what they knew how.

In terms of what I could’ve done differently or more as a team lead is a question I have been thinking about this entire semester, almost everyday. You can put your absolute all into this class, but you can never be perfect at something. I think this is the basis of what makes a great leader... is always wanting to grow. I feel as though I should have asked for everyone’s life schedule before assuming everyone had 24/7 open for this class. This might be also true for a

real-life work scenario, because everyone is different, with different cultures and different schedules. I think it's important to stop expecting yourself, your habits, and schedules from other people. However, that is no excuse not to deliver a product on time, because I've learned as a team lead, it's also important to put your foot down when it needs to be. It's important to be compassionate but also firm. To address these challenges, as a team, once we fixed our communication, our harmony improved. Ever since the first email, I have stated that the #1 thing is communication, and here again, as this class ends, I remain firm in that fact. Everyone gave it their best with what they had, and I hope this experience has been enriching for all. Here's to the culmination of this phase and to the lessons learned that will resonate in our future pursuits!

Team Member Contributions

| Name | Score |
|--------------------|-------|
| Braden Dalit | 2.4 |
| Kimon Monokandilos | 2.2 |
| Lucas Soto | 2 |
| Jada Campbell | 1.7 |
| Edward Chen | 1.7 |