

CS3714 Apple Mobile Software Development

Semester Project Report

Spring 2025

MoodMuse: Your Daily Emotion Journal with Music & Memories

Moon Yejin
Mvula Nchimi
Nawthale Soham

Department of Computer Science
Virginia Tech
Blacksburg, VA 24061

Date: 03.31.2025
Team Number: Team 9
Instructor: Prof. Osman Balci

EXECUTIVE SUMMARY

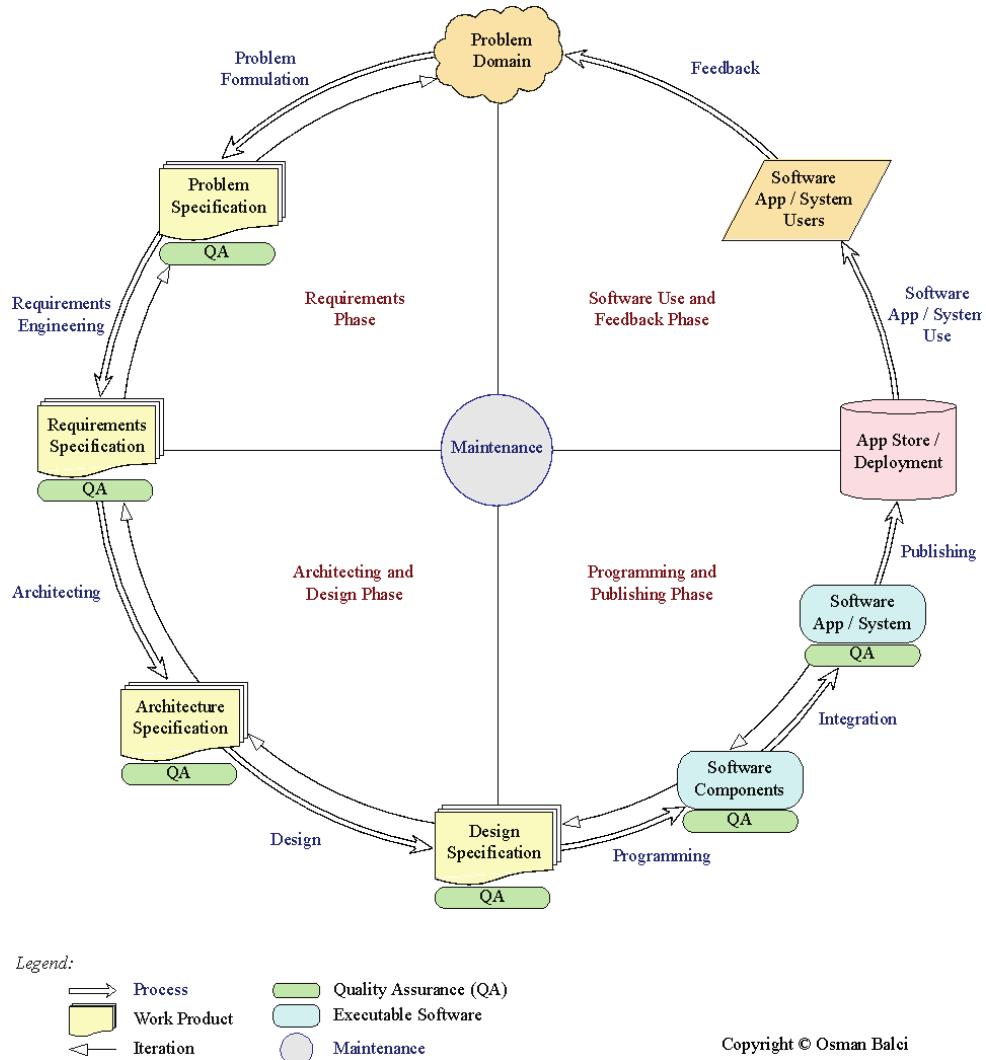
For the semester project, Team 9 (Soham Nawthale, Yejin Moon, Nchimi Mvula) presents MoodMuse. MoodMuse is a daily journaling app aiming to combat mental health issues among young students. It uses Multimedia Features to allow the user to express their emotions in many forms like songs, voice recordings, artwork, and photos. This report outlines the effort we took to achieve desired results. Section 1 and 2 describes the software design cycle and overall goal we followed throughout our development process. Section 3-6 describes the requirements and features we came up with to implement in our app. Section 7 describes our actual product; there are screenshots showing all the features we implemented. The rest of the report describes our efforts, takeaways, and resources used in the project.

TABLE OF CONTENTS

EXECUTIVE SUMMARY.....	ii
1. SOFTWARE LIFE CYCLE.....	1
2. OUR OBJECTIVE.....	1
3. PROBLEM SPECIFICATION.....	2
3.1 WHAT IS THE PROBLEM?.....	2
3.2 EXPECTED MOBILE SOFTWARE-BASED SOLUTION.....	2
3.3 LEARNED AND NEW FEATURES PLANNED TO SOLVE THE PROBLEM.....	2
4. REQUIREMENTS SPECIFICATION.....	3
4.1 FUNCTIONAL REQUIREMENTS.....	3
4.2 NON-FUNCTIONAL REQUIREMENTS.....	3
5. ARCHITECTURE SPECIFICATION.....	4
6. DESIGN SPECIFICATION.....	4
6.1 USER INTERFACE DESIGN SPECIFICATION.....	4
6.2 DATABASE DESIGN SPECIFICATION.....	4
6.3 API DATA MODEL DESIGN SPECIFICATION.....	5
7. DELIVERED SOFTWARE FUNCTIONALITY.....	5
7.1 IMPLEMENTED LEARNED FEATURES.....	5
7.2 IMPLEMENTED NEW FEATURES.....	6
7.3 FUNCTIONALITY DESCRIPTION.....	6
8. CONCLUSIONS.....	6
9. SUBMISSION INSTRUCTIONS.....	7
10. PERCENTAGES OF CONTRIBUTION.....	8
11. CALCULATION OF GRADES.....	9
12. TEAM MEMBER RULES.....	9
13. GRADING SHEET.....	10
REFERENCES.....	11
APPENDIX A: MEETING MINUTES.....	12

1. SOFTWARE LIFE CYCLE

A good software engineer develops software by following the software life cycle shown below.



A programmer (hacker or ad-hoc developer) develops software by looking at the problem and directly coding in an IDE. This approach is known as the *Build-and-Fix Approach*, which must never be used!

2. OUR OBJECTIVE

The objective of our team project is to **demonstrate how capable the team members are** in engineering an Apple mobile software application to solve a problem. The app is created for the purpose of showing how *learned* and *new* complex functionalities and features the team members are capable of developing. The objective is **not** to develop an app for publishing it at the App Store.

3. PROBLEM SPECIFICATION

3.1 What is the Problem?

Worldwide, people struggle with dealing with their emotions, and they end up bottling up the way they feel. This approach is detrimental as one day it could lead to a bigger issue with mental health. One of the best ways to handle emotions is through journaling. Traditional journaling apps are somewhat limited and don't allow you to express how you feel through different mediums. Mediums such as text, video, music, images, etc, are often not all included in journaling apps, limiting user expression.

3.2 Expected Mobile Software-Based Solution

1. API's we will use:
 - a. Zen Quotes API
 - b. An API that gives daily motivational quotes.
 - c. <https://zenquotes.io>
2. Functionality:
 - a. Music Kit (new feature)
 - i. can add a song as an entry to the journal
 - ii. list of songs that they have added
 - iii. can create a playlist of songs based on your mood
 - b. PDF kit (new feature)
 - i. generate a daily PDF journal combining text, images, and music in the app.
 - c. Swift Database
 - i. store songs
 - ii. store journal entries
 - iii. and photos
 - iv. audio from speech entries
 - d. Image slider
 - i. list of images that the user has added to the database
 - e. Vision kit (Camera)
 - i. take /upload images/videos from photo library
 - f. Speech to Text
 - i. to generate text for journal using a person's voice
 - g. Painting and Drawing
 - i. user can draw
 - h. Authentication
 - i. use faceID and passwords to protect your journal entries.
 - i. Audio
 - i. audio stored from the user when they are making a journal entry
 - ii. audio can be replayed from any day

3.3 Learned and New Features Planned to Solve the Problem

The software application we plan to develop is expected to implement the following learned features to be able to solve the problem stated above:

	Topic	Learned Feature
<input checked="" type="checkbox"/>	API (Required)	<ol style="list-style-type: none"> 1. Enable the user to enter a search query, 2. Obtain complex JSON data from an API for the search query, 3. Process the JSON data safely by using Optionals, and 4. Display results using the data extracted from the JSON data.
<input checked="" type="checkbox"/>	SwiftData Database (Required)	<ol style="list-style-type: none"> 1. Create a JSON file containing example content with minimum 5 items, 2. Read the example content with ≥ 5 items from the JSON file, 3. Create a SwiftData database with the example content with many objects and their relationships, 4. Enable the user to perform CRUD operations on the database, and 5. Enable the user to perform complex searches on the database.
<input checked="" type="checkbox"/>	Action Sheet	Display a list of options to select one and process the selected option
<input type="checkbox"/>	Animation	Create an animated image such as Now Playing
<input checked="" type="checkbox"/>	Audio	Play a long audio file with Play and Pause controls
<input checked="" type="checkbox"/>	Audio	Play a short sound file such as the click sound, applaud sound, or Hokie gobble sound
<input checked="" type="checkbox"/>	Authentication	Authenticate user with Password, Face ID or Touch ID
<input type="checkbox"/>	Barcode	Create a barcode of type Code 128, PDF417, Aztec or Quick Response (QR)
<input type="checkbox"/>	Barcode	Enable the user to scan a barcode, query an API with the barcode value, obtain JSON data from the API, process the JSON data, and display results
<input type="checkbox"/>	Barcode	Enable the user to scan a QR barcode and display its corresponding website externally in default web browser
<input type="checkbox"/>	Chart	Create and Display a Bar Chart or Line Chart
<input checked="" type="checkbox"/>	Color Picker	Enable the user to color the UI background with Color Picker
<input checked="" type="checkbox"/>	Context Menu	Long press Text or Image to display a menu of options to select one
<input checked="" type="checkbox"/>	Copy and Paste	Copy phone number, email address, website URL, or image from app and paste it on iPhone, iPad, Mac laptop, or Mac desktop running under the same Apple ID
<input checked="" type="checkbox"/>	Dark or Light Mode	Enable the user to select Dark or Light mode
<input checked="" type="checkbox"/>	Email	Tap email address to open the Mail app externally to send email
<input checked="" type="checkbox"/>	Email	Send email within the app
<input type="checkbox"/>	Game	Create a game with drag gesture to teach a particular topic (e.g., geography)
<input checked="" type="checkbox"/>	Image Slider	Create an Image Slider (a.k.a. Slide Show) with indicator dots, the background color of which can be changed by the user
<input checked="" type="checkbox"/>	Image Slider	Create an Image Slider (a.k.a. Slide Show) with Timer

<input checked="" type="checkbox"/>	Images, Grid of	Display a grid of images where each image can be tapped to display information
<input type="checkbox"/>	Map	Display a map of type Standard, Satellite, Hybrid or Globe with an annotation pin
<input type="checkbox"/>	Map	Display directions for driving or walking on a map type of Standard, Satellite, or Hybrid
<input type="checkbox"/>	Map	Display the map of a given address using forward geocoding
<input type="checkbox"/>	Map	Navigate to another view by tapping an annotation pin's location name
<input type="checkbox"/>	Map	Obtain user's current location and show it on a map type of Standard, Satellite, or Hybrid
<input type="checkbox"/>	Map	Show / remove a location's name above its annotation pin by tapping the pin
<input type="checkbox"/>	Map	Show many locations marked with annotation pins on a Standard type of map
<input type="checkbox"/>	Phone Call	Tap phone number to display the Call Interface
<input checked="" type="checkbox"/>	Photo	Take a photo using camera or pick one from Photos library, store/retrieve it in from document directory and SwiftData database, and display it
<input type="checkbox"/>	Radio	Stream and play radio broadcast with Play and Pause controls
<input checked="" type="checkbox"/>	ScrollView	Display horizontally scrollable list of selectable options with icon and caption
<input checked="" type="checkbox"/>	Speech-to-text	Convert user's speech (voice) to text simultaneously
<input checked="" type="checkbox"/>	Text-to-speech	Convert user-entered text to speech (voice)
<input type="checkbox"/>	Text Message	Compose and send text message (SMS) within the app
<input type="checkbox"/>	Timer	Create a Timer that displays hours, minutes, seconds, and milliseconds
<input checked="" type="checkbox"/>	Video	Take a video using camera or pick one from Photos library, store/retrieve it in from document directory and SwiftData database , and enable the user to play it with controls
<input checked="" type="checkbox"/>	Video – YouTube	Display the YouTube video player with its controls as embedded within a WebView page.
<input checked="" type="checkbox"/>	Voice	Record user's voice using the microphone, store/retrieve it in from document directory and SwiftData database, and enable the user to play it.
<input type="checkbox"/>	Web	Display a website externally in default web browser
<input type="checkbox"/>	Web	Display web browser as part of UI or in another view internally in the app

We also plan to implement the following new features to provide an enhanced solution to the problem stated above.

- PDFKit
 - [PDFKit](#): Create PDF by converting multimedia data in your app.
- Spotify API

4. REQUIREMENTS SPECIFICATION

This section specifies the Functional and Non-Functional Requirements under which our software application will be developed.

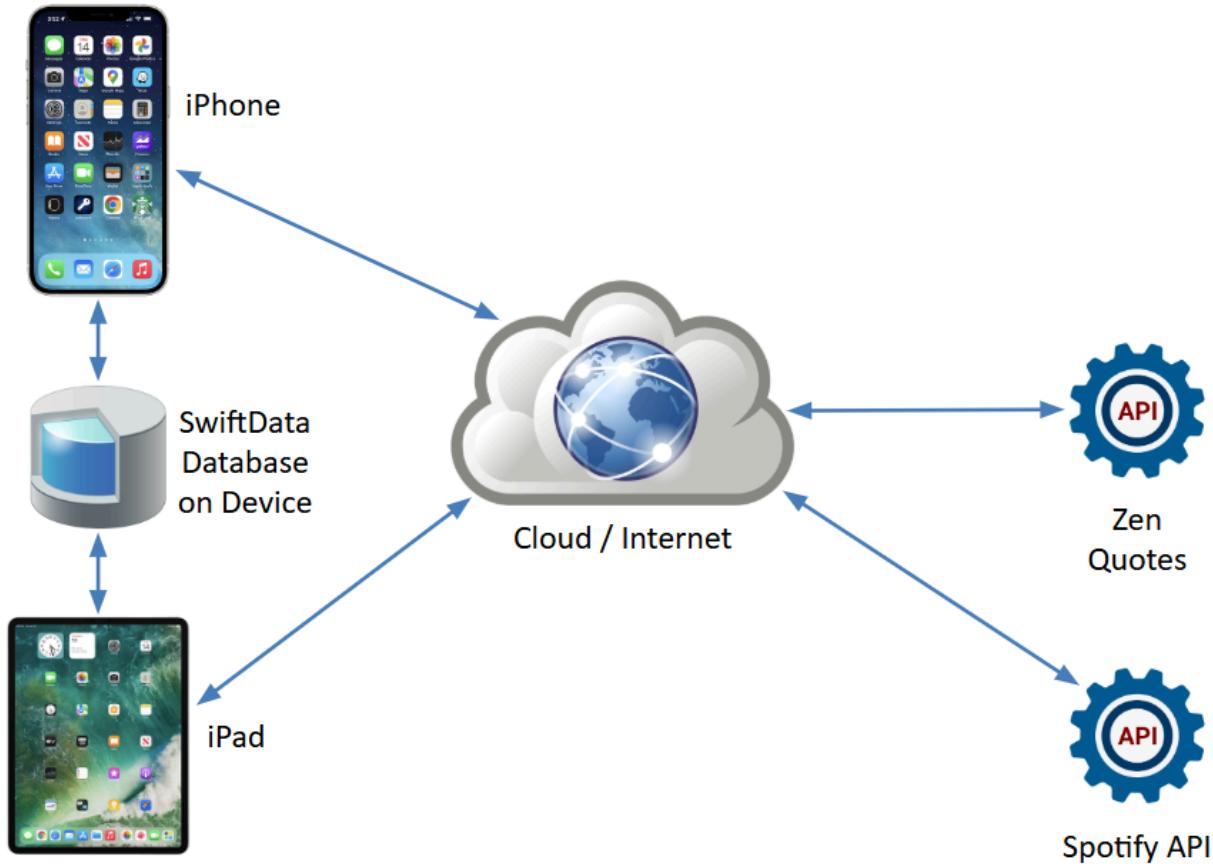
4.1 Functional Requirements

1. The app shall allow users to create journal entries using text, images, and voice recordings.
2. The app shall allow users to add songs to journal entries and play them later.
3. The app shall generate a daily PDF journal that combines text and images.
4. The app shall provide an image slider to view stored pictures in the journal.
5. The app shall authenticate users using Face ID or password before allowing journal access.
6. The app shall convert user speech into text for journal entry creation.
7. The app shall create playlists based on a user's Spotify data.
8. The app shall allow users to create playlists based on their emotional state.
9. The app shall allow users to search journal entries based on name, date, mood, and notes.

4.2 Non-Functional Requirements

1. The app shall be usable on all iOS devices running iOS 18.0 or later without requiring additional adjustments.
2. The app shall support both light mode and dark mode allowing users to view the app in different color.
3. The app shall generate journal log using PDFKit in a readable and properly formatted layout suitable for printing and digital sharing.
4. The app shall respond to user input within 10 seconds for operations including saving a journal log or fetching API data.
5. The app shall launch in less than 5 seconds on devices running iOS 18 or newer device.
6. All software documentation shall be provided in PDF or Word format and included as part of the project submission.

5. ARCHITECTURE SPECIFICATION



Zen Quotes API

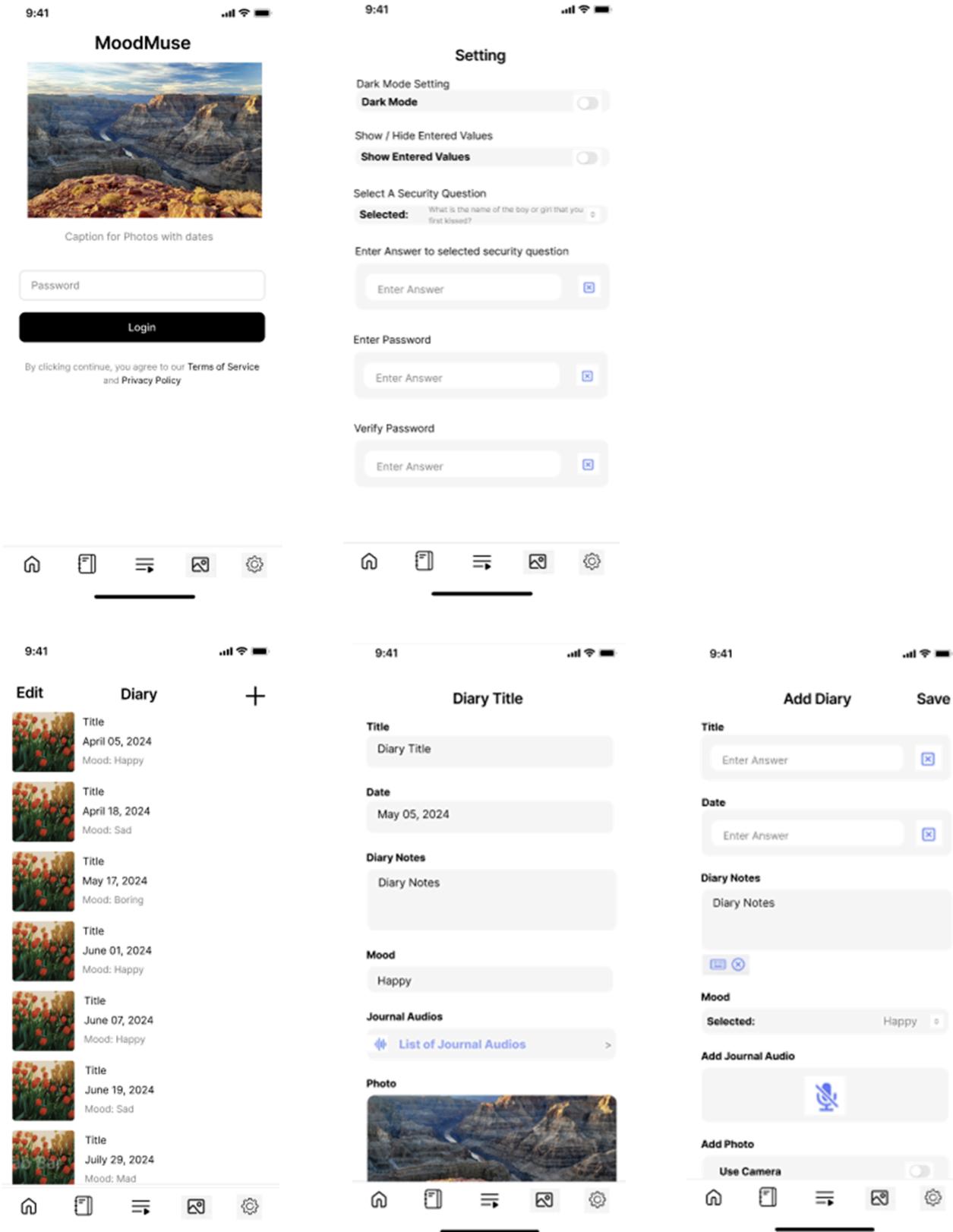
- An API that gives daily motivational quotes.
- <https://zenquotes.io>

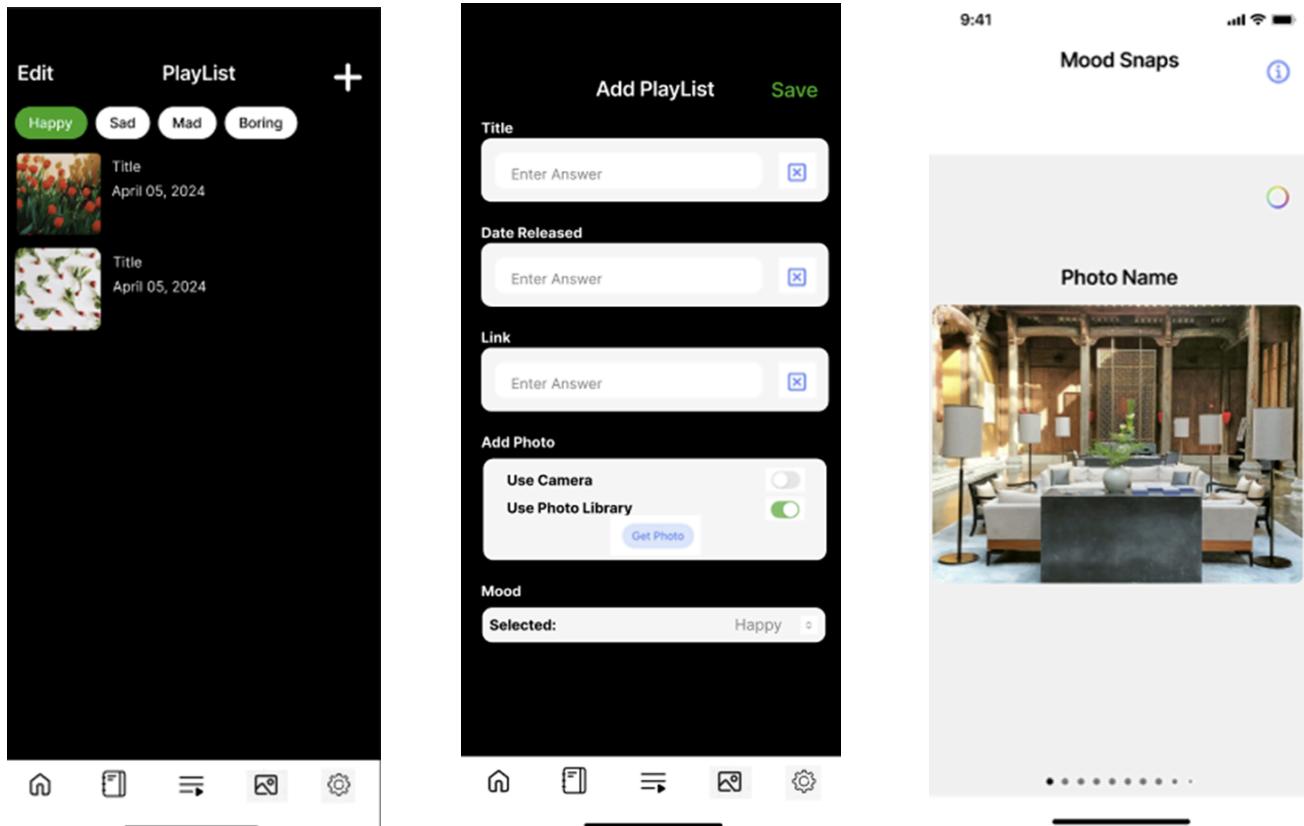
Spotify API

- An API allows users to access songs and playlists.
- <https://developer.spotify.com/documentation/web-api>

6. DESIGN SPECIFICATION

6.1 User Interface Design Specification





6.2 Database Design Specification

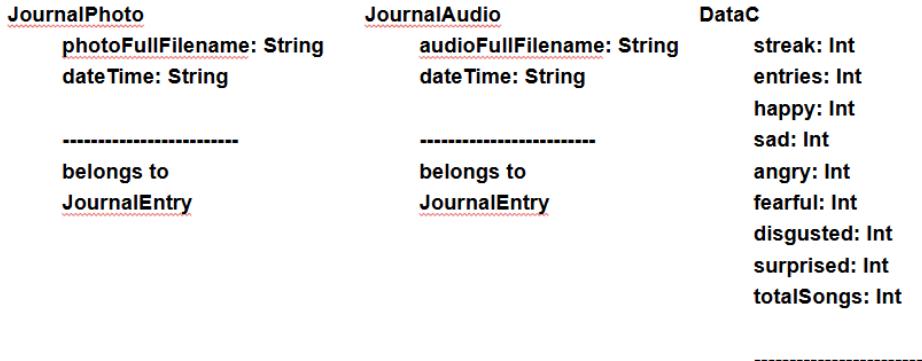
JournalEntry

```
Title: String
Date: String
Notes: String
Mood: String
SpeechToTextNotes: String
JournalAudios: JournalAudio
JournalSong: Song
dataC: DataC
```

has a Songs
has many JournalAudios
has many JournalPhotos
has a DataC

Song

```
id: String
name: String
albumName: String
imageURL: String
duration: Int
popularity: Int
artist: String
mood: String
```



6.3 API Data Model Design Specification

```

response from Zen quotes API:
{
  a = "Zig Ziglar";
  h = "<blockquote>&ldquo;The greatest of all mistakes is to do nothing because you think you can only do a little. &rdquo; &mdash; <footer>Zig Ziglar</footer></blockquote>";
  q = "The greatest of all mistakes is to do nothing because you think you can only do a little. ";
}
////////////////////////////////////////////////////////////////// SPOTIFY API////////////////////////////////////////////////////////////////

playlist api response: {
  href = "https://api.spotify.com/v1/users/j2c1d3eko80t4eeby1vqdkjfn/playlists?offset=0&limit=50&locale=en-US,en;q%3D0.9";
  items = (
    {
      collaborative = 0;
      description = "Created with MoodMuse";
      "external_urls" = {
        spotify = "https://open.spotify.com/playlist/23eN9bEQHIDBj9Vn0CXXv6";
      };
      href = "https://api.spotify.com/v1/playlists/23eN9bEQHIDBj9Vn0CXXv6";
      id = 23eN9bEQHIDBj9Vn0CXXv6;
      images = (
        {
          height = "<null>";
          url = "https://i.scdn.co/image/ab67616d00001e0208cceaa061752f69c1eff6069";
          width = "<null>";
        }
      );
      name = "MoodMuse: Happy";
      owner = {
        "display_name" = Skogisawesome;
        "external_urls" = {
          spotify = "https://open.spotify.com/user/j2c1d3eko80t4eeby1vqdkjfn";
        };
        href = "https://api.spotify.com/v1/users/j2c1d3eko80t4eeby1vqdkjfn";
        id = j2c1d3eko80t4eeby1vqdkjfn;
        type = user;
        uri = "spotify:user:j2c1d3eko80t4eeby1vqdkjfn";
      };
    }
  );
}

```

```

"primary_color" = "<null>";
public = 1;
"snapshot_id" = "AAAABJIFuQY/IzTokzD95kaaQT4CkGt7";
tracks = {
    href = "https://api.spotify.com/v1/playlists/23eN9bEQHIDBj9Vn0CXXv6/tracks";
    total = 3;
};
type = playlist;
uri = "spotify:playlist:23eN9bEQHIDBj9Vn0CXXv6";
},
... (other playlist item)
);
limit = 50;
next = "<null>";
offset = 0;
previous = "<null>";
total = 12;
}

////////////////////////////////////////////////////////////////SPOTIFY API////////////////////////////////////////////////////////////////

api response: {
    tracks = {
        href = "https://api.spotify.com/v1/search?offset=0&limit=5&query=hello&type=track&locale=en-US,en;q%3D0.9";
        items = (
            {
                album = {
                    "album_type" = album;
                    artists = (
                        {
                            "external_urls" = {
                                spotify = "https://open.spotify.com/artist/0eDvMgVFoNV3TpwtrVCoTj";
                            };
                            href = "https://api.spotify.com/v1/artists/0eDvMgVFoNV3TpwtrVCoTj";
                            id = 0eDvMgVFoNV3TpwtrVCoTj;
                            name = "Pop Smoke";
                            type = artist;
                            uri = "spotify:artist:0eDvMgVFoNV3TpwtrVCoTj";
                        }
                    );
                    "available_markets" = (
                        KH,
                        CM,
                        GQ,
                        ... (lot of available markets)
                        LY,
                        XK
                    );
                    "external_urls" = {
                        spotify = "https://open.spotify.com/album/2MDU46hcBn3u94s46BOSdv";
                    };
                    href = "https://api.spotify.com/v1/albums/2MDU46hcBn3u94s46BOSdv";
                    id = 2MDU46hcBn3u94s46BOSdv;
                    images = (
                        {
                            height = 640;
                            url = "https://i.scdn.co/image/ab67616d0000b27346e1307c35579c3483ea7b03";
                            width = 640;
                        },
                        .. other images with lower pixel values
                    );
                    "is_playable" = 1;
                    name = "Shoot For The Stars Aim For The Moon (Deluxe)";
                    "release_date" = "2020-07-20";
                    "release_date_precision" = day;
                    "total_tracks" = 34;
                    type = album;
                    uri = "spotify:album:2MDU46hcBn3u94s46BOSdv";
                }
            }
        )
    }
}

```

```

};

artists = (
{
  "external_urls" = {
    spotify = "https://open.spotify.com/artist/0eDvMgVFoNV3TpwtrVCoTj";
  };
  href = "https://api.spotify.com/v1/artists/0eDvMgVFoNV3TpwtrVCoTj";
  id = 0eDvMgVFoNV3TpwtrVCoTj;
  name = "Pop Smoke";
  type = artist;
  uri = "spotify:artist:0eDvMgVFoNV3TpwtrVCoTj";
},
{
  "external_urls" = {
    spotify = "https://open.spotify.com/artist/31W5EY0aAly4Qieq6OFu6I";
  };
  href = "https://api.spotify.com/v1/artists/31W5EY0aAly4Qieq6OFu6I";
  id = 31W5EY0aAly4Qieq6OFu6I;
  name = "A Boogie Wit da Hoodie";
  type = artist;
  uri = "spotify:artist:31W5EY0aAly4Qieq6OFu6I";
}
);
"available_markets" = (
  AR,
  AU,
  BR,
  BG,
  ... lots of other available markets
  CG,
  IQ,
  XK
);
"disc_number" = 1;
"duration_ms" = 190534;
explicit = 1;
"external_ids" = {
  isrc = USUM72014370;
};
"external_urls" = {
  spotify = "https://open.spotify.com/track/2r6OAV3WsYtXuXjvJ1lIDI";
};
href = "https://api.spotify.com/v1/tracks/2r6OAV3WsYtXuXjvJ1lIDI";
id = 2r6OAV3WsYtXuXjvJ1lIDI;
"is_local" = 0;
"is_playable" = 1;
"name" = "Hello (feat. A Boogie Wit da Hoodie)";
popularity = 74;
"preview_url" = "<null>";
"track_number" = 30;
type = track;
uri = "spotify:track:2r6OAV3WsYtXuXjvJ1lIDI";
},
... (4 other song items)
);
limit = 5;
next = "https://api.spotify.com/v1/search?offset=5&limit=5&query=hello&type=track&locale=en-US,en;q%3D0.9";
offset = 0;
previous = "<null>";
total = 106;
};

}
}

```

7. DELIVERED SOFTWARE FUNCTIONALITY

Our delivered software application implements the following features:

7.1 Implemented Learned Features

Our delivered software application implements the following learned features:

Topic	Learned Feature	Tutorials
<input checked="" type="checkbox"/> API <small>(Required)</small>	1. Enable the user to enter a search query, 2. Obtain complex JSON data from an API for the search query, 3. Process the JSON data safely by using Optionals, and 4. Display results using the data extracted from the JSON data.	Countries, Barcode, MusicAlbums, Recipes, Movies, Companies, NationalParks
<input checked="" type="checkbox"/> SwiftData Database <small>(Required)</small>	1. Create a JSON file containing example content with minimum 5 items, 2. Read the example content with ≥ 5 items from the JSON file, 3. Create a SwiftData database with the example content with many objects and their relationships, 4. Enable the user to perform CRUD operations on the database, and 5. Enable the user to perform complex searches on the database.	Communicate, Countries, PhotosVideos, MusicAlbums, Recipes, Movies, Companies, NationalParks, VTQuest
<input checked="" type="checkbox"/> Action Sheet	Display a list of options to select one and process the selected option	TravelGuide
<input type="checkbox"/> Animation	Create an animated image such as Now Playing	HokieCheer
<input checked="" type="checkbox"/> Audio	Play a long audio file with Play and Pause controls	HokieCheer
<input type="checkbox"/> Audio	Play a short sound file such as the click sound, applaud sound, or Hokie gobble sound	HokieCheer, JigsawPuzzle
<input checked="" type="checkbox"/> Authentication	Authenticate user with Password, Face ID or Touch ID	PhotosVideos
<input type="checkbox"/> Barcode	Create a barcode of type Code 128, PDF417, Aztec or Quick Response (QR)	Barcode
<input type="checkbox"/> Barcode	Enable the user to scan a barcode, query an API with the barcode value, obtain JSON data from the API, process the JSON data, and display results	Barcode
<input type="checkbox"/> Barcode	Enable the user to scan a QR barcode and display its corresponding website externally in default web browser	Barcode
<input checked="" type="checkbox"/> Chart	Create and Display a Bar Chart or Line Chart	Companies

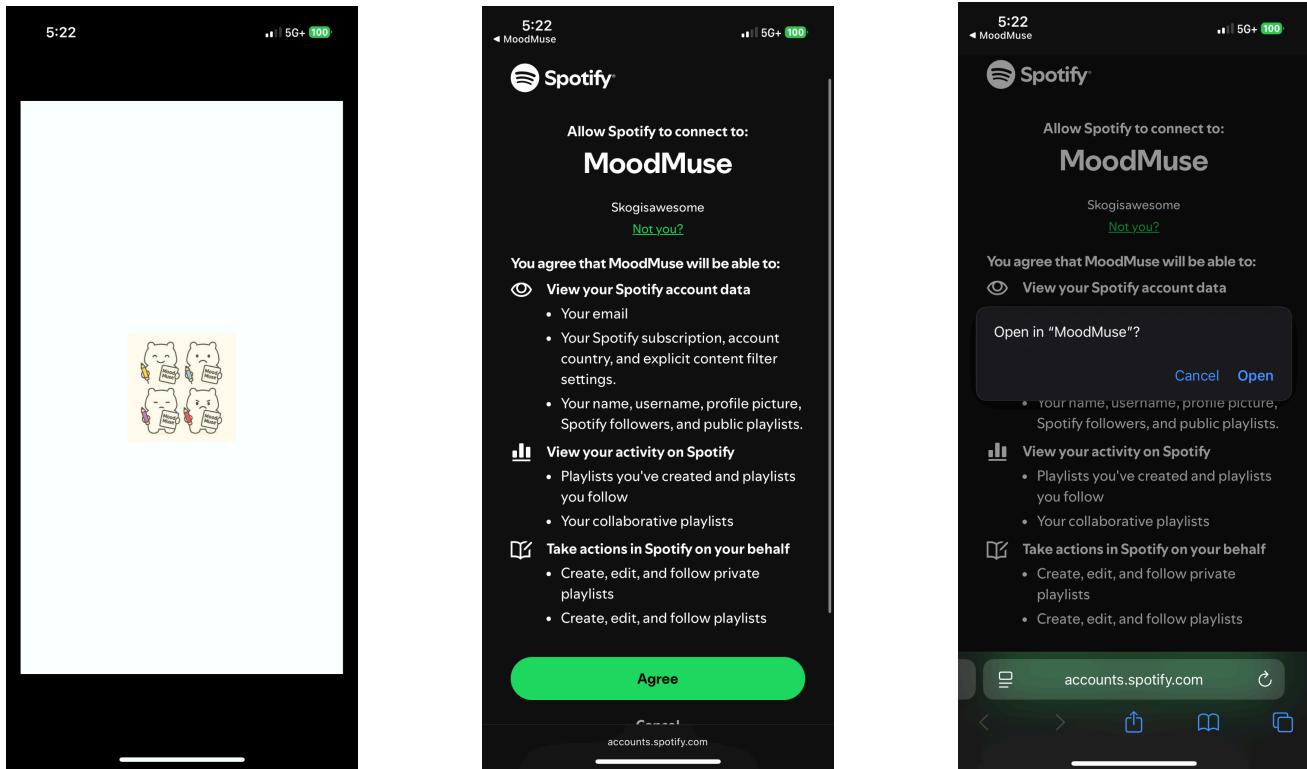
<input checked="" type="checkbox"/>	Color Picker	Enable the user to color the UI background with Color Picker	TravelGuide, Blacksburg
<input type="checkbox"/>	Context Menu	Long press Text or Image to display a menu of options to select one	TravelGuide, Blacksburg
<input type="checkbox"/>	Copy and Paste	Copy phone number, email address, website URL, or image from app and paste it on iPhone, iPad, Mac laptop, or Mac desktop running under the same Apple ID	Blacksburg, Communicate, Countries, Movies, Companies
<input checked="" type="checkbox"/>	Dark or Light Mode	Enable the user to select Dark or Light mode	All of the tutorial apps except the first two
<input type="checkbox"/>	Email	Tap email address to open the Mail app externally to send email	Blacksburg, Communicate
<input type="checkbox"/>	Email	Send email within the app	Communicate
<input type="checkbox"/>	Game	Create a game with drag gesture to teach a particular topic (e.g., geography)	JigsawPuzzle
<input checked="" type="checkbox"/>	Image Slider	Create an Image Slider (a.k.a. Slide Show) with indicator dots, the background color of which can be changed by the user	TravelGuide, Blacksburg
<input checked="" type="checkbox"/>	Image Slider	Create an Image Slider (a.k.a. Slide Show) with Timer	Countries, PhotosVideos, Barcode, MusicAlbums, Recipes, Movies, Companies, NationalParks
<input type="checkbox"/>	Images, Grid of	Display a grid of images where each image can be tapped to display information	Countries, PhotosVideos
<input type="checkbox"/>	Map	Display a map of type Standard, Satellite, Hybrid or Globe with an annotation pin	TravelGuide, Blacksburg, Countries, PhotosVideos, Companies, VTQuest, NationalParks
<input type="checkbox"/>	Map	Display directions for driving or walking on a map type of Standard, Satellite, or Hybrid	Blacksburg, VTQuest
<input type="checkbox"/>	Map	Display the map of a given address using forward geocoding	VTQuest
<input type="checkbox"/>	Map	Navigate to another view by tapping an annotation pin's location name	Blacksburg, PhotosVideos, VTQuest
<input type="checkbox"/>	Map	Obtain user's current location and show it on a map type of Standard, Satellite, or Hybrid	Blacksburg, VTQuest
<input type="checkbox"/>	Map	Show / remove a location's name above its annotation pin by tapping the pin	Blacksburg, PhotosVideos, VTQuest
<input type="checkbox"/>	Map	Show many locations marked with annotation pins on a Standard type of map	Blacksburg, PhotosVideos, VTQuest
<input type="checkbox"/>	Phone Call	Tap phone number to display the Call Interface	Blacksburg
<input checked="" type="checkbox"/>	Photo	Take a photo using camera or pick one from Photos library, store/retrieve it in/from document directory and SwiftData database, and display it	PhotosVideos, MusicAlbums, Companies, NationalParks
<input type="checkbox"/>	Radio	Stream and play radio broadcast with Play and Pause controls	HokieCheer
<input type="checkbox"/>	Scroll View	Display horizontally scrollable list of selectable options with icon and caption	TravelGuide
<input type="checkbox"/>	Speech-to-text	Convert user's speech (voice) to text simultaneously	Communicate, NationalParks
<input checked="" type="checkbox"/>	Text-to-speech	Convert user-entered text to speech (voice)	Communicate
<input type="checkbox"/>	Text Message	Compose and send text message (SMS) within the app	Communicate
<input type="checkbox"/>	Timer	Create a Timer that displays hours, minutes, seconds, and milliseconds	JigsawPuzzle
<input type="checkbox"/>	Video	Take a video using camera or pick one from Photos library, store/retrieve it in/from document directory and SwiftData database , and enable the user to play it with controls	PhotosVideos
<input type="checkbox"/>	Video – YouTube	Display the YouTube video player with its controls as embedded within a WebView page.	Blacksburg, Movies
<input checked="" type="checkbox"/>	Voice	Record user's voice using the microphone, store/retrieve it in/from document directory and SwiftData database, and enable the user to play it.	Communicate, NationalParks
<input checked="" type="checkbox"/>	Web	Display a website externally in default web browser	TravelGuide, Blacksburg, Countries, Barcode, MusicAlbums, Recipes, Movies, Companies, NationalParks
<input type="checkbox"/>	Web	Display web browser as part of UI or in another view internally in the app	TravelGuide

7.2 Implemented New features

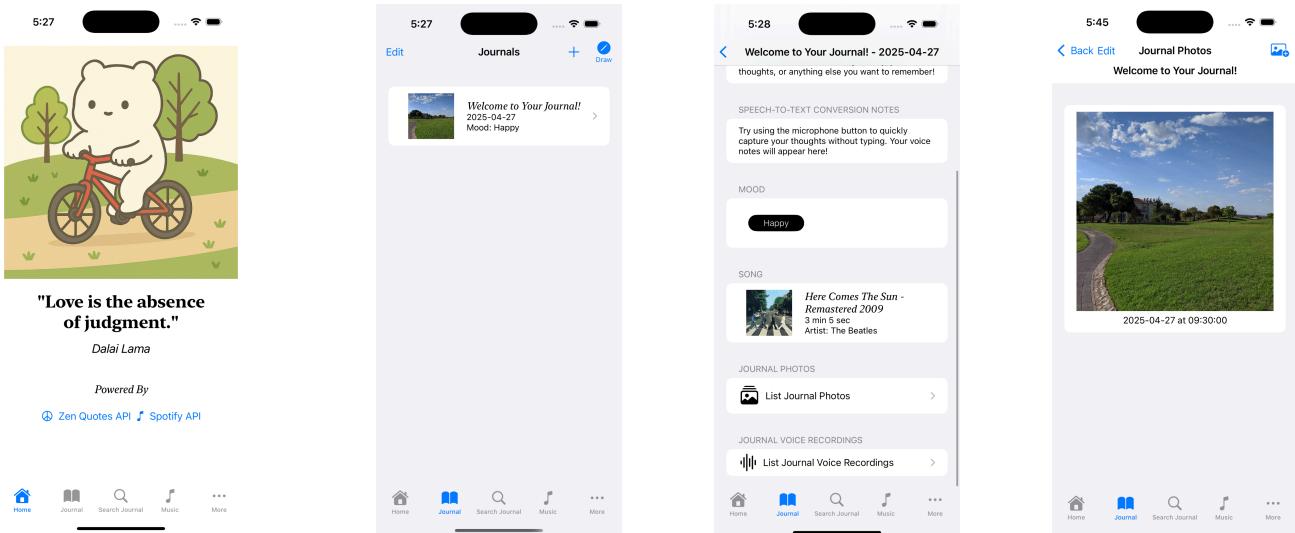
Our delivered software application implements the following features:

1. We incorporated painting and drawing into our app to allow users to express their emotions through drawings. On the drawing page, you can change colors and drawing instruments, and if you wish, your drawings can be stored in the Photos app.
2. We incorporated the PDF kit in our app. This allows users to get a pdf representation of their journal entry. The pdf kit page you can search a journal by its title. Then you can download it and choose where to store it.
3. We incorporated Spotify API. We use the spotify api to manage all music related requests. This includes searching for a song, adding the song to a playlist, creating the playlist, viewing all the playlist. You will first notice that we use the spotify API when you open the app for the first time and it will ask you to authenticate and allow MoodMuse to use your data.
4. We incorporated Charts in our app. We use this feature to give users a visual representation of statistics in the app. When you are on the profile page, you will see several statistics. We implemented 2 chart styles to show users what type of mood is most frequently logged in the app.

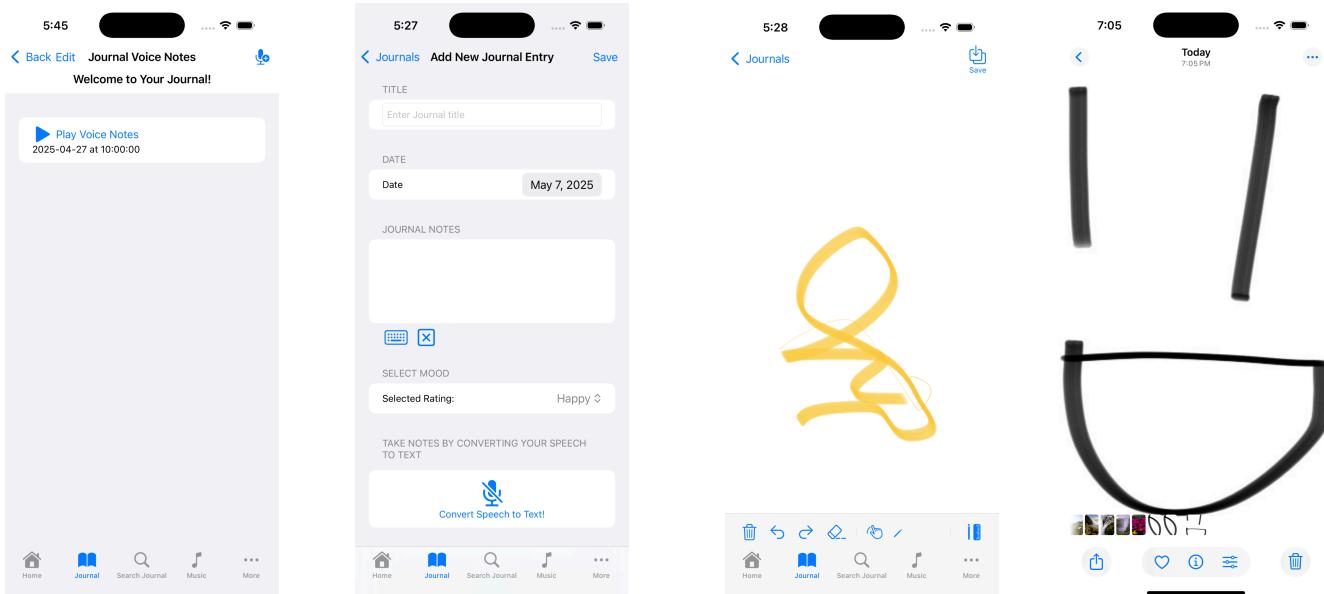
7.3 Functionality Description



When you open the app for the first time, the app displays the following launchImage (designed by Yejin). Once the app loads, it redirects you to your native browser to login/sign up for spotify. The app heavily relies on your spotify account for any of the music features. The website redirects you back to the app.

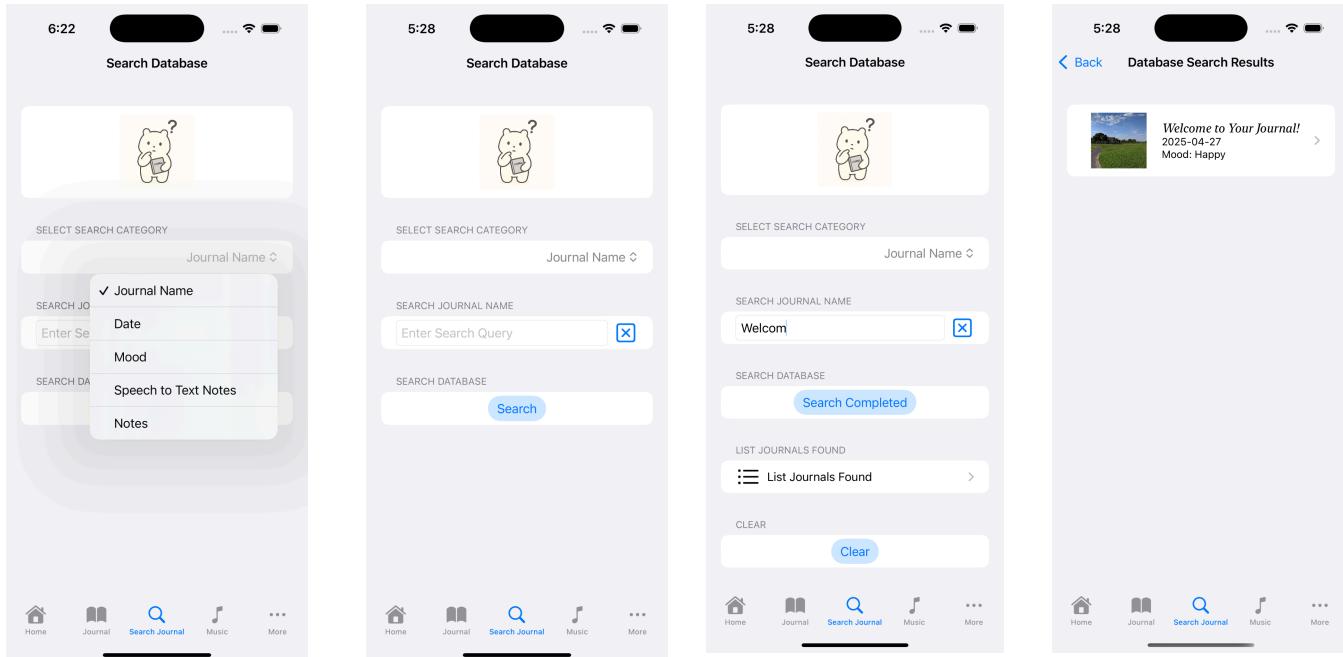


- When you open the app, the second start screen is the home page, as seen in the first view above, with the image of the mood muse bear. Below the image is a quote from the Zen Quotes API that is constantly changing.
- The next view is the journals page with a list of the journal entries. At the top, you have a button to edit the entries (delete), add an entry, and make drawings based on your mood
- The next view is the journal details page, where you can see the information pertaining to a journal entry. As a part of this page, you can access a list of photos specific to the entry as well as journal audios specific to the entry (seen below in the leftmost image). You can also add photos and voice notes to the entry.

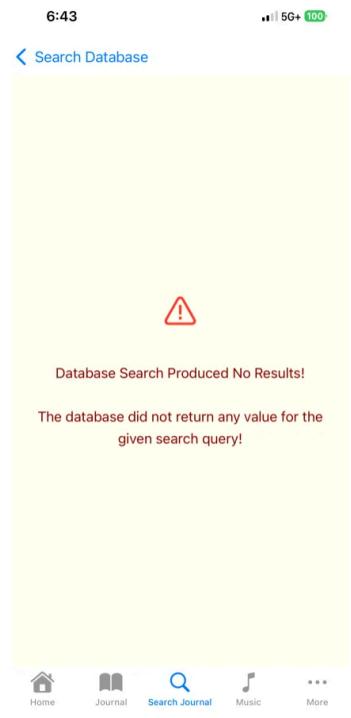


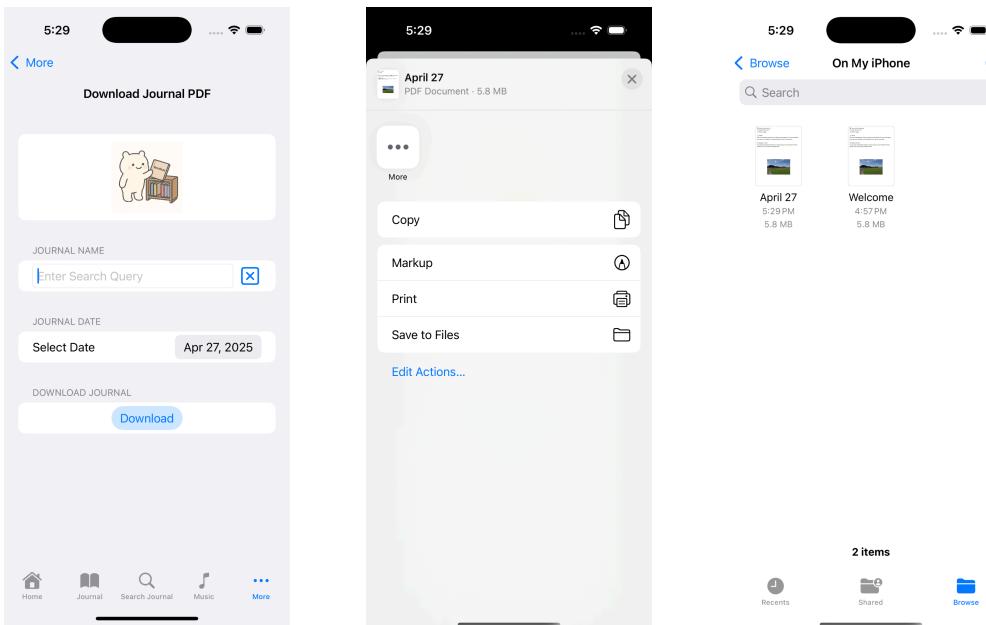
- The next view is the add Journal Entry page, where you can add an entry record notes with text-to-speech and add photos and images. As a part of this page, you can also search for a song to add to the entry.

- The next view is the drawing page, where you can doodle your thoughts in different colors and with different drawing tools. You can also save these drawings to your camera roll upon completion. The rightmost image above shows another drawing I made being stored in the camera roll.

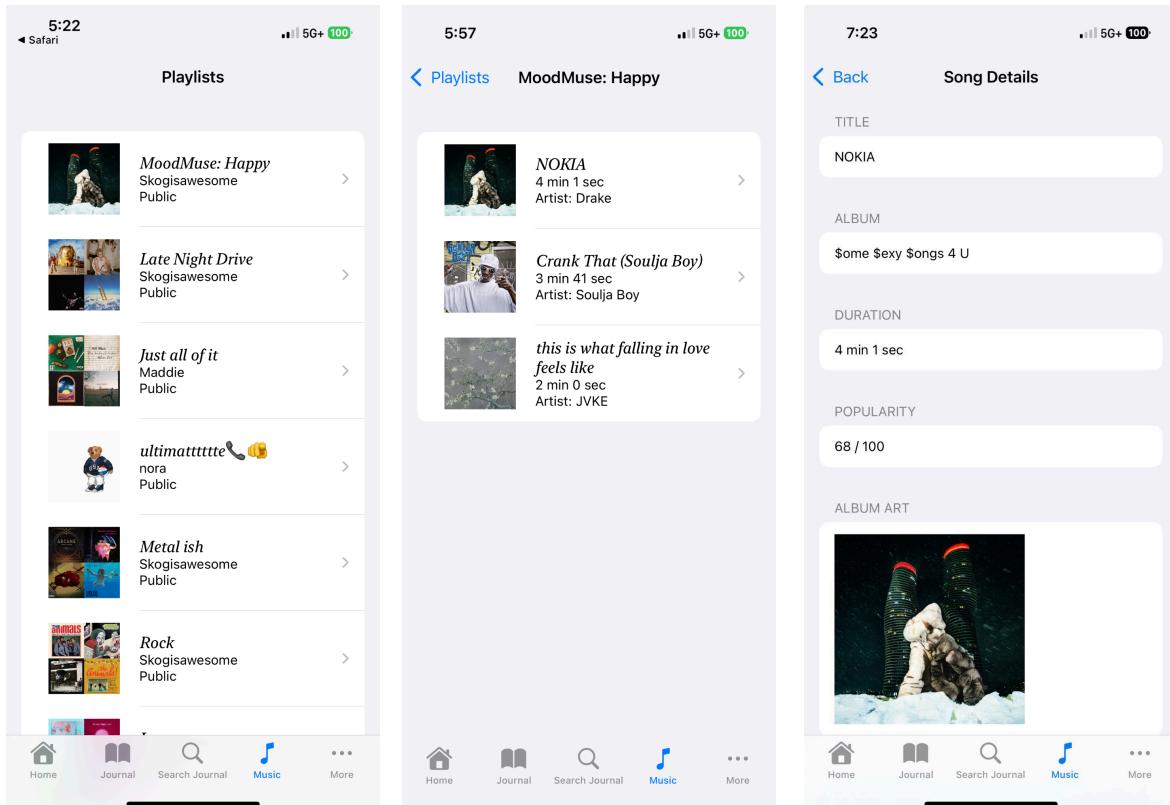


- The first view above shows the search database page for the journal, allowing you to search through entries for name, date, mood, speech-to-text notes, or notes. Each uses `localizedStandardContains(searchQuery)` to find journals that contain the query.
- The next 3 views show the process of picking a category, entering a query, and then seeing the list of journal entries.
- !!!!!If no journal is found, the view on the right will show highlighting that the relevant entries could not be found.

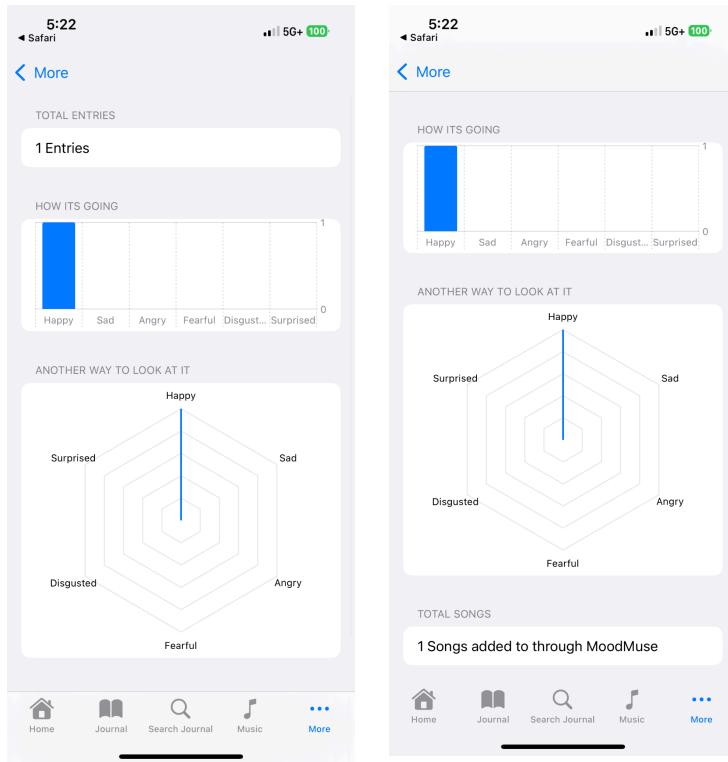




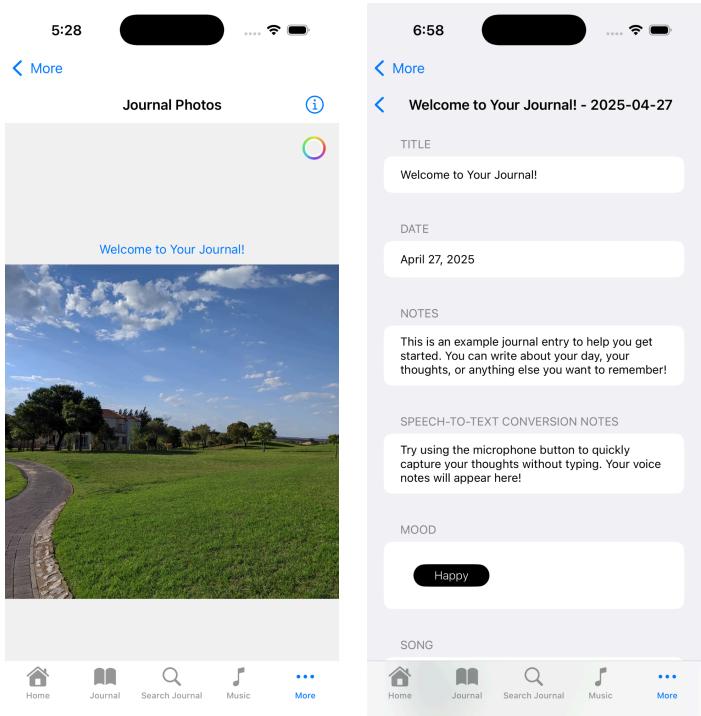
- The first view above is the download journals page, where you can enter the name of the PDF/journal and select the date of the journal and download it. This will then allow you to save, copy, markup, or print the generated pdf. As you can see in the next view when you press save to files, the entries have been added to files with whatever name you decide.



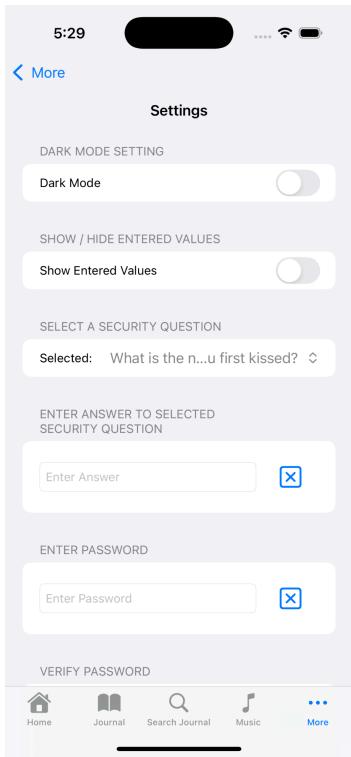
When you tap the music tab at the bottom, this is what you see. You will first see a list of all the playlist in your spotify account. This includes playlists that MoodMuse will create on your behalf. When you click on a playlist, you will be taken to a view that shows a list of songs in the playlist. When you click on a song, it takes you to song details.



This is the view you will see when you click on the profile page. It displays the statistics on your journaling. It shows the number of entries in different moods. and it displays the information in graphs.



- The page on the right is the Journal Photos page where an image slider will be displaying your photos. Upon clicking on the title in blue it'll take you to the journal details page as seen in the next view.



- The last view is the settings page, where you can make a password and personalize your app experience.

8. CONCLUSIONS

MoodMuse development was a comprehensive and rewarding experience for us, as it compelled us to merge acquired and novel Apple mobile development techniques to create a deep, emotionally supportive journaling app. With APIs such as ZenQuotes and Spotify, implementing features such as PDF printing, painting, and personalized music playlists, we were able to finish a multimedia journaling app that allows users to record and reflect on their emotions in meaningful ways.

MoodMuse is a testament to our ability as novice software engineers to develop intuitive, efficient, and robust mobile applications. While this app was not developed for deployment in the commercial marketplace, it is proof of our capacity to leverage the capabilities of leading-edge iOS functionality to craft a product that is innovative, responsive, and user-focused.

9. SUBMISSION INSTRUCTIONS

Please carefully follow the instructions below in submitting your project:

- Step 1: Microsoft Word or PDF version of this report file is acceptable.
- Step 2: Create a new folder named **CS3714 Project Team N** where N = your team number. Do not replace the spaces in the filename with anything else.
- Step 3: Place the following in the created folder:
 - a. This file: *CS3714 Team N Semester Project Report.docx* or *.pdf* (Replace N with your team number. Do not change this filename and do not remove the spaces in the filename!)
 - b. Entire **Xcode project** folder containing the project files and executable file
- Step 4: Right click the folder and select Compress to generate the ZIP file **CS3714 Project Team N.zip**.
- Step 5: Verify the ZIP file: Unzip the compressed file on another computer and make sure that the ZIP file is properly created.
- Step 6: Upload your ZIP file to your account in **OneDrive**, right click the uploaded file and select **Share** from the pop-up menu, enter **balci@vt.edu**, add a note as **CS3714 Project Team N Submission**, and click **Send** before the due date.
- Step 7: Only one team member should submit the project on behalf of the team **by copying the team members**. **Late submissions are not allowed.**
- Step 8: If your project app is a multi-user app or an app that requires demonstration to evaluate its functionality, then get an appointment with Dr. Balci before the due date to give a hands-on demo so that your app can be graded properly.

10. PERCENTAGES OF CONTRIBUTION

We hereby certify that the list of contributions and the corresponding percentages of contribution specified below truly reflect the actual contributions of the team members.

(Write your name as your signature)

<i>Student Name & List of Contributions</i>	<i>% Contributed</i>	<i>Signature</i>
Soham Nawthale's Contributions: 1. Spotify API integration 2. ZenQuotes API integration 3. Profile page 4. Music page	33.33%	Soham Nawthale
Nchimi Mvula's Contributions: 1. Database creation outside DataC and certain elements within Song. 2. Journal Page 3. Drawing page	33.33%	Nchimi Mvula
Yejin Moon's Contributions: 1. PDF kit integration 2. Basic app features: authentication, settings, launch image, 3. Artwork used in the app 4. Search Database, Photos page, Database creation	33.33%	Yejin Moon
Sum of Percentages:	100%	

11. CALCULATION OF GRADES

Grade Calculation.xls						
	A	B	C	D	E	F
1		Grade Given for the Project	Student 1 Percentage of Contribution	Student 2 Percentage of Contribution	Student 3 Percentage of Contribution	
2	Example:	86	40%	25%	35%	100%
3						
4	Student 1	91.73	= B2 + B2 * (C2 - 33.333%)			
5	Student 2	78.83	= B2 + B2 * (D2 - 33.333%)			
6	Student 3	87.43	= B2 + B2 * (E2 - 33.333%)			
7						

Additional percentage of contribution cannot exceed 10%.

If a situation arises where a student is doing more than 10% extra work, Dr. Balci must be informed immediately.

12. TEAM MEMBER RULES

1. It is the professional obligation of each team member **not** to drop or withdraw from the course after committing to be a team member.
2. Each team member is expected to contribute equally.
3. You shall submit percentages of contribution together with a list of each student's individual contributions with signatures of all team members. In case of disagreement, you shall submit it separately with your rationale. (Write your name as representing your signature.)
4. Grades shall be determined based on the percentages of contribution.
 - a. If you contribute more than your equal share, then your grade shall be increased based on the extra percentage of contribution, which cannot be more than 10%.
 - b. If you contribute less than your equal share, then your grade shall be decreased accordingly.
 - c. The extra percentage of contribution shall not be more than 10%. Dr. Balci shall be notified immediately if a situation arises where a student needs to contribute more than 10% extra.
5. A team member who does not cooperate with other team members for conducting the project with equal contribution shall be penalized. **Doing more work than agreed upon by the team and claiming extra contribution shall not be acceptable.** Cooperation is essential!

13. GRADING SHEET

CS3714 Semester Project Grading: Team N			
	Points	Earned	Notes
Requirements Specification	6		Minimum 9 Functional Requirements (3 per team member) Minimum 6 Non-Functional Requirements (2 per team member)
Architecture Specification	4		Revised diagram in CS3714 Team Presentation 1.pptx with description of each API
Software Design Specification	20		(10 pts): Minimum 12 annotated easy-to-read balsamiq.com UI wireframes (4 per team member) (6 pts): SwiftData database classes (judged based on quantity and complexity) (4 pts): JSON file data structure for each API with used parts marked
Penalty for Missing Required Features			(-5) Warning messages displayed upon app launch (-10) For each app crash during testing for grading (-15) No SwiftData database used for app data storage and retrieval (-15) No SwiftData database complex searches implemented properly (-15) No API is used with search query entered by the user (-10) Empty list (e.g., favorites) with no example content of at least 5 items (-5) App name is too long and truncated (-5) No app icon, (-5) no launch image, or (-5) if reused from course apps (-10) Number of project team meeting minutes < 5 and/or their content lacks sufficient detail (-X) Other missing features typically implemented in tutorials
Software Features			
Quantity of Features	20		Number of features properly implemented from the list of App Features Learned in Course Tutorials (.../StudentsOnly/Project/LearnedFeatures.html) + new features if implemented
Variety of Features	20		Variety of features properly implemented from the list of App Features Learned in Course Tutorials (.../StudentsOnly/Project/LearnedFeatures.html) + new features if implemented
Complexity of Features	20		Difficulty of creating each feature. Does it take minimum 72 hours (for 3 students) to develop the app? Complexity is judged with respect to minimum 72 hours time requirement.
Software Documentation	5		Meaningful and informative documentation, similar to the course tutorials, showing that you understand what the Swift and SwiftUI code is doing.
Project Report Quality	5		(-5) Instructions in red are not deleted in the final version of the report. Title page, table of contents, styles, formatting, hyperlinks, etc.
Total Points	100	0.00	
Team Members:	% Cont.	Grade	Minimum time required: 8 hrs/week × 4 weeks × 3 students = 72 hrs
1. Student 1 Name	33.333%	0.00	
2. Student 2 Name	33.333%	0.00	
3. Student 3 Name	33.333%	0.00	

REFERENCES

- Apple, Inc. (2025), “SwiftUI,” <https://developer.apple.com/xcode/swiftui/>
- Apple, Inc. (2025), “Xcode,” <https://developer.apple.com/xcode/>
- Balci, O. (2025), “CS3714 Apple Mobile Software Development Course Website,” <https://manta.cs.vt.edu/cs3714>
- Reddit, “RadialGraphShape,” https://www.reddit.com/r/SwiftUI/comments/femg90/how_to_create_a_radar_graph/
- Swift (2025), “Swift Programming Language,” <https://swift.org/>
- Apple Inc. (2024). *PDFKit*. Apple Developer Documentation. <https://developer.apple.com/documentation/pdfkit>
- Figma. (n.d.). *Figma: The collaborative interface design tool*. <https://www.figma.com/>

Sample Photo

Pexels. (n.d.). Delighted friends having lunch in cafe [Photograph]. Pexels. <https://www.pexels.com/photo/delighted-friends-having-lunch-in-cafe-4349791/>

Pexels. (n.d.). Person writing in journal with pen and glasses [Photograph]. Pexels. <https://www.pexels.com/photo/person-writing-in-journal-with-pen-and-glasses-30706600/>

Pexels. (n.d.). White ceramic mug on white wooden table [Photograph]. Pexels. <https://www.pexels.com/photo/white-ceramic-mug-on-white-wooden-table-374885/>

Pexels. (n.d.). Silhouette of two person walking on seashore during golden hour [Photograph]. Pexels. <https://www.pexels.com/photo/silhouette-of-two-person-walking-on-seashore-during-golden-hour-1533720/>

Pexels. (n.d.). Person using MacBook Pro on table [Photograph]. Pexels. <https://www.pexels.com/photo/person-using-macbook-pro-on-table-1181244/>

Sample Audio

Pixabay. (n.d.). Fast Epic – 15 sec [Audio file]. Pixabay. <https://pixabay.com/music/main-title-fast-epic-15-sec-108425/>

Pixabay. (n.d.). Moonlight Sonata (Beethoven) – 28 sec [Audio file]. Pixabay. <https://pixabay.com/music/solo-piano-moonlight-sonata-335629/>

Pixabay. (n.d.). Uplifting Classical - Moments of Triumph [Audio file]. Pixabay.
<https://pixabay.com/music/modern-classical-uplifting-classical-moments-of-triumph-303175/>

Pixabay. (n.d.). Background Music Soft Piano [Audio file]. Pixabay.
<https://pixabay.com/music/piano-background-music-soft-piano-108428/>

Pixabay. (n.d.). Silent Evening (Calm Piano) [Audio file]. Pixabay.
<https://pixabay.com/music/piano-silent-evening-calm-piano-108429/>

List References to the APIs.

Spotify, Inc. (2025), “Spotify Web API Website,” <https://developer.spotify.com/documentation/web-api>

Zen Quotes (2025), “Zen Quotes Website,” <https://zenquotes.io>

APPENDIX A: MEETING MINUTES

Meeting Number 1

Date & Duration:	4/4/25 1 hour
Location:	Zoom
Members Present:	All
Members Absent:	None
Discussions:	Discussing Ideas for the proposal, introductions
Decisions Made:	Come up with a list of ideas, form a discord group for communication
Work Assignments:	Come up with Ideas
Minutes Prepared By:	Soham Nawthale

Meeting Number 2

Date & Duration:	4/16/25
Location:	Zoom
Members Present:	All
Members Absent:	None
Discussions:	Assigning tasks for proposal
Decisions Made:	Nchimi does 3.1, Soham does 3.2, Yejin does 3.3
Work Assignments:	Nchimi does 3.1, Soham does 3.2, Yejin does 3.3
Minutes Prepared By:	Soham Nawthale

Meeting Number 3

Date & Duration:	4/23/25 8pm for 1 hour
Location:	Zoom
Members Present:	All
Members Absent:	none
Discussions:	Assigning tasks for in Demo
Decisions Made:	Soham Works on APIs, Grace and Nchimi create database and basic structure
Work Assignments:	Soham Works on APIs, Grace and Nchimi create database and basic structure
Minutes Prepared By:	Soham Nawthale

Meeting Number 4

Date & Duration:	5/5/2025 Multiple Hours 3-5
Location:	Lab/Zoom
Members Present:	All
Members Absent:	none

<i>Discussions:</i>	Working on individual parts to bump up the score with more advanced features.
<i>Decisions Made:</i>	Soham works on finishing up the Spotify API, Grace implements PDF kit, and Nchimi implements painting and drawing.
<i>Work Assignments:</i>	Soham worked on APIs, Grace worked on PDF kit, Nchimi worked on painting/drawing
<i>Minutes Prepared By:</i>	Nchimi Mvula

Meeting Number 5

<i>Date & Duration:</i>	5/7/2025
<i>Location:</i>	Lab/Zoom
<i>Members Present:</i>	All
<i>Members Absent:</i>	none
<i>Discussions:</i>	Finishing up the project
<i>Decisions Made:</i>	Soham finish profile page, Grace help with profile page database, Nchimi organize some slide information into the report.
<i>Work Assignments:</i>	Soham finished the profile page, Grace helped in completing the profile page in terms of the database, Nchimi wrapped up the documentation, All wrapped up the documentation.
<i>Minutes Prepared By:</i>	Yejin Moon