My (Musical) Life

Design Document

Team Number: sddec20-13
Adviser & Client: Dr. Henry Duwe

Team Members:
Christian Hernandez - Project Manager
Chaz Clark - iOS Developer
Daksh Goel - Backend Developer
Vignesh Krishnan - Frontend Developer
Vatsal Bhatt - Backend Developer

Team Email: sddec20-13@iastate.edu
Team Website: http://sddec20-13.sd.ece.iastate.edu/

Executive Summary

Development Standards & Practices Used

***Please not that these standards and practices are what we plan on using. We are still in the research/planning phase of our project. Therefore, we have not used some practices yet. Also, we have not begun development*

- Development Standards
  - Commented Code
  - Quality
  - Efficiency
  - Apple Developer Standards
  - Waterfall Design
- Practices
  - Test code regularly
  - Agile Development
- Engineering Standards
  - Quality
  - Performance
  - Safety

Summary of Requirements

- **Functional requirements**
  - User Data (from their mobile device)
    - Location
    - Weather
    - Schedule
  - A music Streaming service account
  - Music Recommendations
  - Predictive Mood
  - Volume Control

- **Economical requirements**
  - Apple developer account
  - Online Cloud service/platform
  - A music Streaming service account

- **Environmental requirements**
  - Network reception in user's mobile device
  - iOS device (iPhone)
  - Xcode (Apple's integrated development software)
  - Macbook (Apple's laptop computer device that runs Xcode software)
- **Apple Design Guideline Requirements**
  - Consistency

    - Feedback
    - Direct manipulation
    - User control

Applicable Courses from Iowa State University Curriculum

- S E 185 - Problem Solving in Software Engineering
- CPR E 185 - Introduction to Computer Engineering and Problem Solving I
- COM S 227 - Object-Oriented Programming
- COM S 228 - Introduction to Data Structures
- COM S 309 - Software Development Practices
- CPR E 310 - Theoretical Foundations of Computer Engineering
- COM S 311 - Introduction to the Design and Analysis of Algorithms
- S E 319 - Construction of User Interfaces
- S E 329 - Software Project Management
- S E 339 - Software Architecture and Design
- COM S 363 - Introduction to Database Management Systems
- ENGL 314 - Technical Communication

New Skills/Knowledge acquired that was not taught in courses

- Swift
- iOS Development
- Machine Learning
- Using Spotify's API
- Amazon Web Services

Table of Contents

List of figures/tables/symbols/definitions

1 Introduction

1.1 Acknowledgement

We would like to first thank Dr. Henry Duwe for meeting with us weekly and providing us with guidance and advice as we being our senior design project. Dr. Duwe has done an amazing job in regards to helping us set up this project by giving us small assignments to complete each week for him. We would also like to thank the Electronics Technology Group for providing us with a website and a Git project. Lastly, we would like to thank the TAs and the professors (Dr. Lotfi Ben-Othmane and Dr. Daji Qiao) for their guidance and help thus far.

1.2 Problem and Project Statement

Have you ever been in a sad mood and listened to sad music despite it not helping and making you even sadder? Do you typically play high-tempo music when heading to the gym and while working out at the gym? Do you listen to softer, calmer music as you study for your next exam at the library? If you answered yes to some of the questions, *My (Musical) Life* app will be perfect for you! For people who love music, this app will be great to use on the daily.

Our app will use data from multiple different, possible sources (location, pulse, calendar, weather, schedule, time of day, etc.) to determine which song is the best to pipe directly into your ears. The app will require little user input, and the music suggestions will improve as the user continues to use the app. Having a Fitbit or Apple Watch may improve your experience with this app. Additionally, the app will require a Spotify account. Overall, as long as the app is open on your phone, the app will continue to play music based on the different sources listed above.

1.3 Operational Environment

The end product of our project is an iOS mobile application, *My (Musical) Life*, which could be installed and used by anyone owning Apple's mobile device, namely, iPhone. Upon installation and registration, our app will require some permissions from the user including access to their mobile device's location and some user data. Our app is only supported by iPhone's Operating System and will be not available for use in mobile devices running Android.

1.4 Requirements

The requirements of our project are as follows:

- **Functional requirements**
  - User Data (from their mobile device)
    - Location
    - Weather
    - Schedule
  - A music Streaming service account
  - Music Recommendations

    - Predictive Mood
    - Volume Control

- **Economical requirements**
    - Apple developer account
    - Online Cloud service/platform
    - A music Streaming service account

- **Environmental requirements**
    - Network reception in user's mobile device
    - iOS device (iPhone)
    - Xcode (Apple's integrated development software)
    - Macbook (Apple's laptop computer device that runs Xcode software)
- **Apple Design Guideline Requirements**
    - Consistency
    - Feedback
    - Direct manipulation
    - User control

1.5 Intended Users and Uses

An intended user for our iOS mobile application, *My (Musical) Life*, will be someone who loves to listen to music and wants to listen to their favorite music with minimal to no user input at specific times and during specific events of their day. Our app will create personalized playlists for each user depending on their mood, location, and schedule. Other factors including time of the day, and weather will also play a role in creating these playlists. The main purpose of the app is to start playing music in a user's mobile device without their input, during specific times of the day when the user would possibly be wanting to listen to music. An example use case will be a student wanting to listen to soft and calm music while studying in the library. Our app will determine that the user i.e., the student is in the library through the location of their mobile device and will start playing soft and calm music in their mobile device, while having 5 other playlist recommendations in case the user did not like the music that is being played currently.

1.6 Assumptions and Limitations

Assumptions:

1. The user will have an existing account with Spotify music streaming service or will be willing to create one for the use of the application.
2. The user will have an existing Spotify library to choose songs from, otherwise songs may be chosen from random spotify playlist for specific mood.
3. The user will have an existing Google account or will be willing to create one for the use of the application.

4. The user is comfortable with the private data and permissions that the application requires to provide the most intelligent song selections.

Limitations:

1. The application will only be available on iOS devices as that is the platform that our client uses.
2. Users with streaming services other than Spotify or local music storage will not be able to integrate their music library.
3. Users without a Google account will not be able to use features that use unique personal data for selecting songs.
4. There will be minimal user input and songs will only be able to be played automatically with a skip feature, not chosen specifically.
5. The user will need to allow device location permissions at all times to receive location based song selections in real time.
6. The user will need to allow bluetooth device connection permissions at all times to receive device based song selections in real time.

1.7 Expected End Product and Deliverables

The *My (Musical) Life* iOS application will be the final deliverable to our client and it will be commercialized on the Apple App Store.

Description:

My (Musical) Life is a music streaming application that changes your music based on what you do every day. This is a great app for listeners who love having music playing throughout the day, almost like a soundtrack for their life! Finding the right song, playlist, or genre for a particular activity can be difficult and time consuming. *My (Musical) Life* can take the pain out of choosing music by predicting and playing the songs you like. Powered by Spotify, *My (Musical) Life* will select your favorite songs that fit your daily activities. Whether it be working out at the gym, working in your office, studying in the library, or taking a road trip, never worry about changing the song again. *My (Musical) Life* will recommend and play music that best fits your daily routine. Using your location, calendar, and Bluetooth connectivity data, it will select the genre of music that best matches your mood for a specific activity. As you provide feedback on selections made by *My (Musical) Life*, it will build a personalized music profile that will only play the songs that you love.

2. Specifications and Analysis

2.1 Proposed Approach

The proposed approach to our app is to develop an iOS application for our users to interact with. The iOS application will then communicate to various third-party APIs. In order to play music, our app will use Spotify's API and stream music from there. The main part of our application deals with mood prediction. Currently, we are researching possible ways to do this. One approach we are looking into is a third-party Machine Learning implementation. This would allow us to build a predictive model for each user. However, this would most likely require lots of data and we are not sure if we will have access to that much data. The other approach would be to come up with a predictive algorithm. This would not use machine learning and therefore, would not be as specific to each user.

For our data storage we plan to use a hybrid of on-device storage and AWS. The on-device storage will store user-login, and settings. Our AWS backend will build a profile of the type of music the users listen to, which will help us recommend music. We will also use the backend to recommend the next song(s) to the user. This will either be ML or a predictive algorithm (as stated above).

So far, we have implemented a POC application that uses the spotify API to play music and search for music. Additionally, we have all been learning Swift as most of us are brand-new to iOS development. We have been spending most of our time doing research on several APIs, data storage, and ways that we can predict mood.

2.2 Design Analysis

So far, we have developed a high-level system level diagram (see below). However, parts of the backend may change as we do more research. This is because we are still unsure of how we will implement mood prediction. Apart from this one module, we are confident in our design. In addition, one of the requirements of our application is that there should be limited user input. This may impact some of the data that we can collect. One of the strengths of our design is that we will utilize as many integrations that iOS allows us.

2.3 Development Process

To satisfy the requirements of our client and the senior design course we are currently following a Waterfall method. We have completed the Planning & Requirements stage and currently in the System & Software development design stage. This helps the team gain a high-level perspective on the project. Once software development comes around, we will begin to adopt an agile approach towards the project. This will allow the team to implement continuous client feedback into our application.

2.4 Conceptual Sketch

Our System-level diagram contains multiple modules that will deliver the necessary requirements. *My (Musical) Life* will consist of 5 modules. Local Storage, AWS Database, AWS Cloud Processing, Third-party API, and User Feedback. The application is designed to take minimal user input and provide a nice automated experience. To do so there will be a minimal user interface. The user will input password and login credentials that will be stored locally on the device. This data will be used to access their third-party accounts such as their google calendars and schedule. The local storage data will also be used to deliver audio streaming service. Our largest requirement is to predict what type of sounds the user wants to listen to and deliver it. Part of this processing will be done on Amazon web services. We are storing detailed user models in the database to build a personalized profile for each user. This will serve as a key factor when we try to predict the users sounds. To minimize the amount of on device processing we will utilize AWS cloud functions to perform more of the heavy-duty computation in prediction.

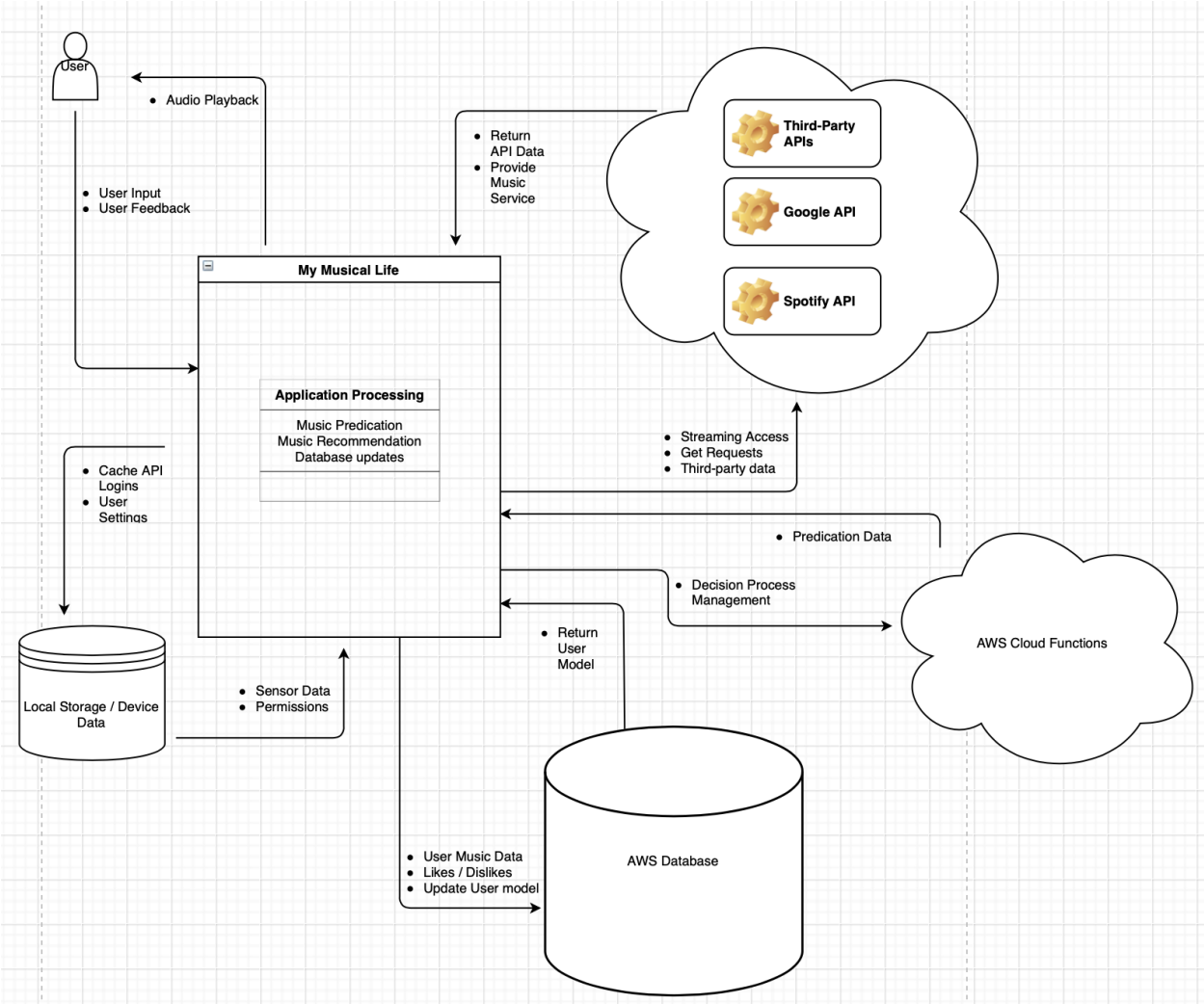

Figure 1

3. Statement of Work

3.1 Previous Work And Literature

As for previous work that has been done, there are apps that have been created that do something similar to the app we plan on making. However, our app will be different than the three that we will mention.

First, there is an app by the name of *MusicFit.* The goal of this app is to generate music based on the user's body movements through the iPhone's sensors [3]. Ultimately, this app will generate music based on the user's change of pace while working out. The app will only generate music based on four genres (techno, electro, idm, chill) [3]. *My (Musical) Life* will be different because our app will generate music based on a couple of factors (calendar, time of day, weather, connectivity, location, etc.). Our app will predict what type of music the user would like to listen to based on those factors. Additionally, our app will not be limited to only four genres.

Next, another similar application that exists is *Musicovery.* For this app, "recommendations of tracks, artists, genres, and playlists are personalized in real time to each listener, according to his music preferences, listening behavior, and listening history" [2]. Again, this is different from *My (Musical) Life* since our app will generate music based on different factors. Furthermore, *Musicovery* seemed to have been an app at some point, but now it is only a website. We could not find it in the app store.

Lastly, the last application that seemed to be similar to our project is *Songza*. Songza would use date, time, and past listening history to generate "playlists based on predictions about the user's mood and/or activity at the time" [1]. However, this app would allow the user to also search for playlists based on genres, mood, and artists. Adding on, the app was shut down, but it was integrated into Google Play Music. The difference between *My (Musical) Life* and *Songza* (now integrated in Google Play Music) is the fact that *My (Musical) Life* is planned to be an app with minimal to no user input. The app is planned to be extremely simple to use for the user. Overall, *Songza* seems to have the most similar idea to *My (Musical) Life,* but with the plan of making our app play music based on location, connectivity, weather, (potentially) heart rate, etc., *My (Musical) Life* will be different than Songza.

3.2 Technology Considerations

- **Spotify API**
  - **Strengths**: Spotify has tutorials and many webpages with how to integrate their API into an app.
  - **Weaknesses:** Each member of the team has not had much experience using Spotify's API. Therefore, there will be a learning curve.
- **AWS**
  - **Strengths:** AWS seems to be simple to use and integrated into an app. There are many tutorials that can be followed to use AWS.
  - **Weaknesses:** Some members of the team do not have experience with using AWS
- **Google API**

    - **Strengths:** There are many tutorials one can follow with Google API. Additionally, this API seems to be relatively simple to integrate into an app.
    - **Weaknesses:** Experience with using Google API.
- **iOS**
    - **Strengths:** There are many nice tools to use while developing iOS applications. Apple provides some tutorials for iOS development as well.
    - **Weaknesses:** Apple has many restrictions on iOS apps. There are many guidelines we must follow. Also, one out of the five team members has experience with iOS development.

3.3 Task Decomposition

Below are the current tasks:

- Research/Learning APIs, Swift User Interfaces, Spotify API, AlamoFire, AWS, Machine Learning, etc.
- Create a database to store data
- Have the app connect to multiple APIs (Spotify, Google, and any Third-Party APIs)

3.4 Possible Risks And Risk Management

As we develop the app, there will be multiple issues that we will run into. The first issue is the user not having a Spotify Premium account. This may make it difficult to access all types of music for the user. Trying to decide how to make the app function without the need of a Premium account will be a hurdle.

Next, trying to figure out how to predict the user's mood based on factors such as location, weather, time, connectivity, etc. and generate the right music will be difficult. Currently, the team is attempting to decide on whether to use Machine Learning or an algorithm to combat this issue. We have not come to any conclusions yet.

Developing an iOS app will create some roadblocks for the team. Apple has many restrictions and guidelines one must follow. Creating an app in iOS rather than Android is a bit more difficult. Only one out of the five team members is familiar with iOS development. Thus, there will be a learning curve for the four other team members.

Obtaining user data, such as accessing the user's calendar, location, etc. may be a difficult task for the team. There may be some restrictions on the amount of data we can receive from each user.

3.5 Project Proposed Milestones and Evaluation Criteria

Overall, the most important milestone for the team is to have a fully functional app by December 2020. Another key milestone is, by May 2020, the team plans on having a working

app but not finalized. Lastly, the last key milestone the team currently has is having a claim on the way the team plans on making a prediction of the mood in order to generate music.

As for a smaller milestone, by the middle of March, research regarding tools, software, APIs, etc. should be completed. It is expected that the team begins developing the app during the month of March.

Tests will include many types of tests. Currently, we plan on using Unit Testing, System Testing, Regression Testing, Stress Testing, etc. Since we have not begun developing the app, we have not settled on our test cases yet.

3.6 Project Tracking Procedures

The group will use GitLab to track progress throughout the course of this and next semester. GitLab has a section called "Issues" where one can add issues to an "Issues Board." Thus, this is the project management tool the team will use throughout the duration of the project.

3.7 Expected Results and Validation

The desired outcome is to have a fully functional app by December 2020. *My (Musical) Life* app will be in the app store free for users to use. The application will predict the user's mood based on many factors and generate music based on that mood. Additionally, the application will be bug-free.

We will confirm the application will work at a high level by running an extensive list of tests (System Tests, Unit Tests, Stress Tests, Regression Tests, etc). Additionally, it is important the app satisfies our client (Dr. Duwe) and other potential users.

4. Project Timeline, Estimated Resources, and Challenges

4.1 Project Timeline

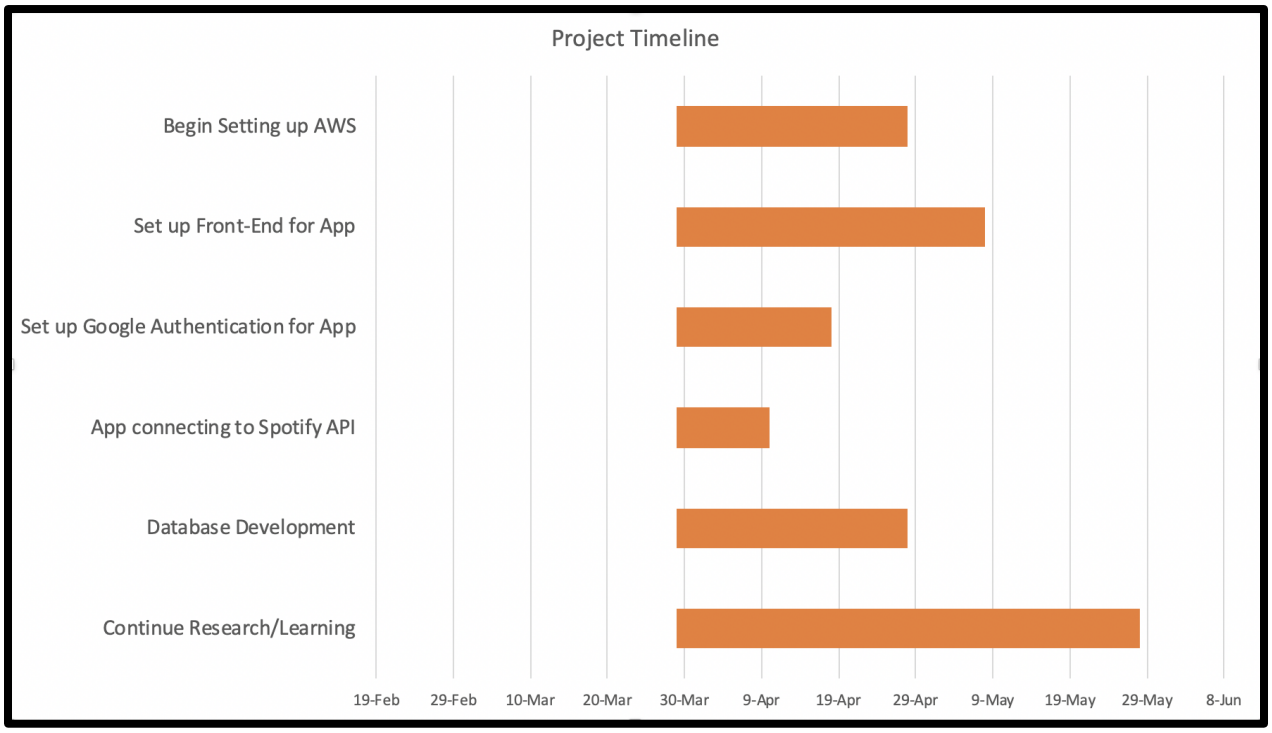

Figure 2: Gantt Chart of Project Timeline

The Gantt chart above shows some of the short term goals that we have for this project. We are now slowly moving to the phase that primarily consists of design and development. However, these are some core pieces we must cover in the next couple of weeks. First, we would like to begin setting up AWS for our app. The team is projecting that this may take up to 30 days to complete. Next, setting up the front-end will take about 40 days to complete. The app will not consist of many pages, but the initial set up of the front-end may take awhile. Thus, 40 days seems to be reasonable. Going forward, we need to set up Google authentication for the app. Thus, this will involve having the users logging into his/her google account via the app. This should take about 20 days to complete. Additionally, the app will need to connect to Spotify and use Spotify. Thus, connecting to Spotify's API is something else the team must accomplish. As mentioned, users will need a Spotify account. This should take about 12 days to complete. Setting up the database is another task that the team will begin working on. Setting up the database can be difficult and take a while to accomplish. Thus, this may take 30 days or more to complete. Lastly, research and learning will take 60 days to complete. Research and learning may take longer as we discover new technologies and software to potentially use during the development process.

4.2 Feasibility Assessment

Overall, the expectation of this project is to be an iOS app that our client (Dr. Duwe) will be able to download and use. Realistically speaking, by the end of the Spring semester (May 2020), there will be issues with the app. However, by the end of the Fall semester (December 2020), the goal is for the app to be fully functional. The goal is for the app to meet all of the requirements listed at the beginning of this document. The application should also meet the expectations of Dr. Duwe.

As for challenges the team may run into, the team may run into challenges using new technologies we do not have experience. Thus, we will have to allocate time for learning new technologies. Another challenge is time. We are given a year to create this app; therefore, given our class schedules and other obligations, this will be a challenge. Additionally, if we do decide to use Machine Learning, that will be a challenge. No one on the team has experience with Machine Learning. There will be more challenges throughout the development of this project.

4.3 Personnel Effort Requirements

Task	Description	Projected Effort
Research/Learning of APIs, Swift User Interfaces, Spotify API, AlamoFire, AWS, Machine Learning, etc.	This task simply includes research and learning any new technologies needed for the project.	This takes quite a bit of effort. Most likely a month is a good timeframe for this task.
Create a database to store data locally and in the cloud	We need to set up a database to store information locally and in the cloud. Currently, we plan on storing data in both places.	Setting up databases can be difficult and confusing. Ultimately, a month will be enough time to set up a fully database.
App(Frontend) connects to APIs (Google API, Spotify API, Third-Party APIs)	After the research/learning phase, we would like to begin to connect to these APIs. That is, we need to integrate some of these APIs into our app and learn how to use them.	This should only take about two weeks to accomplish. With the right training and tutorials, this can be done in a short time frame.
Use Lambda functions in AWS to update DB	Write push, pull function to update and read our database	This should take about 2-3 weeks depending on how many tables we use.
Connect backend to Spotify API	Backend needs to query /recommendations endpoint in order to get the next song(s)	This should take about one week. However, building the logic for recommendation may take about a month
Implement Google login	Frontend will use google to authenticate users	This should take about 2 weeks.

4.4 Other Resource Requirements

Our Project has a few resource requirements in order to function.

Device: Since we have chosen to utilize the newest Swift frameworks available our app needs to be running on an Apple device that is running iOS 13 or higher.

Accounts: A Spotify account will be required to use the app in order to enable streaming data to the user.

Cellular Data or Wifi: A constant connection will need to be enabled in order for the program to function.

4.5 Financial Requirements

Currently, our app does not require any financial resources.

5. Testing and Implementation

5.1 Interface Specifications

Since we have not begun developing, we are unsure about what the interface specifications will look like right now.

5.2 Hardware and software

- Testflight
    - This allows us to distribute our ios builds among our testers without submitting a build to the app store
- PostMan
    - This will allow us to send and retrieve data to our custom-built APIs providing us a detailed view of our JSON packets

5.3 Functional Testing

We will be using Apple's built-in UI testing to ensure UIkit components are responsive

5.4 Non-Functional Testing

Once the app is developed, we will perform some performance, usability, security, and compatibility tests.

5.5 Process

We have not tested anything yet since we have not begun developing. Therefore, this section is not applicable to us at the moment.

5.6 Results

We are still in the early stages of our project (research/learning phase). Therefore, we have not begun the development of our app yet which means tests have not been created yet.

6. Closing Material

6.1 Conclusion

In conclusion, we have not begun developing our app at the moment. We are still currently in the planning phase and working with Dr. Duwe to make sure we are on the right track. The planning phase also includes lots of research and learning. Overall, the goal is to create an app that is fully-functional by December 2020. We want our adviser/client (Dr. Duwe) to be satisfied with the final product. Therefore, we are making sure to follow the tasks and advice given to us by Dr. Duwe in order to stay on track. Additionally, our research currently involves tools and technologies we can potentially use for this app. Ultimately, we use the best tools, technologies, APIs, etc. for our app. All in all, there are lots of resources for our team to explore, and we are attempting to explore all options right now.

6.2 References

[1] J. Crook, "Google Will Shut Down Songza App, Songza.com To Fold Into Google Play Music," *TechCrunch*, 02-Dec-2015. [Online]. Available: https://techcrunch.com/2015/12/02/google-will-shut-down-songza-app-songza-com-to-fold-into-google-play-music/. [Accessed: 24-Feb-2020].

[2] "Musicovery B2B," *Musicovery B2B*. [Online]. Available: http://b2b.musicovery.com/. [Accessed: 24-Feb-2020].

[3] "MusicFit," *App Store*, 02-Jan-2018. [Online]. Available: https://apps.apple.com/us/app/musicfit/id1186085097. [Accessed: 24-Feb-2020].

6.3 Appendices

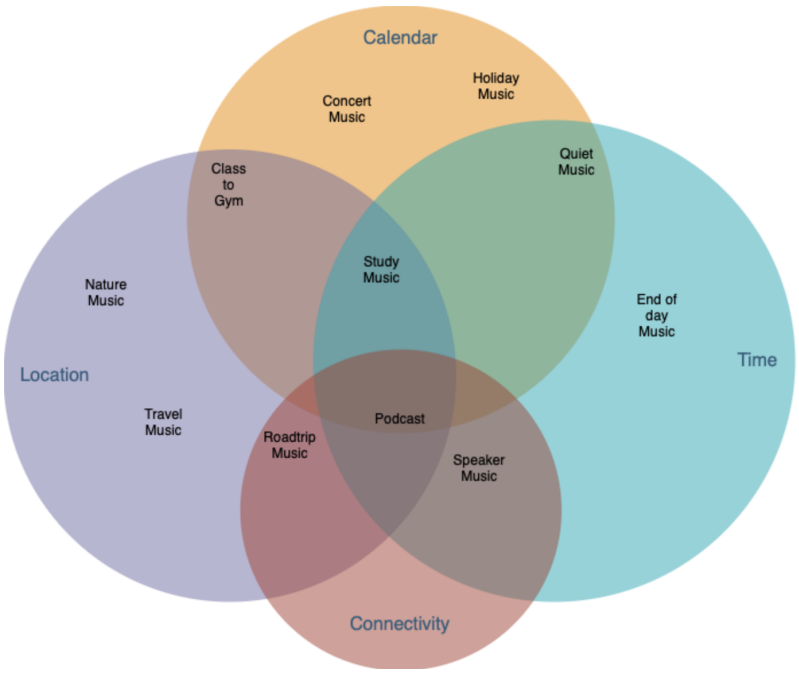

Figure 3: Use Case Diagram