*Music Matcher*

Team Name: Ray Huang, Amy Paul, Mario Verdugo, Amir Yazdani

Project Proposal

### **Introduction**

The project is a music taste-based app for making connections with people! You connect a music streaming service that you use, either Spotify or Apple Music, and the app uses machine learning to connect you to people with similar tastes in the music they listen to. Our group theorizes that the best friendships come from people who have the same taste in music. The motivation behind this project lies in testing this theory and attempting to develop friendships to the users of the app. What is novel about the app is that users are connected through purely music preference. Most connection apps base matching through other preferences. In addition, the app will be usable on either Apple or Android mobile devices for wider compatibility.

As for challenges, we will primarily be using two API’s: Spotify and Apple Music to rely on data and music preferences. Relying so heavily on APIs out of the team’s control will most likely present unexpected challenges such as data retrieval not yielding exactly what we expect, and connecting via http calls is often more convoluted than it seems at first. Also, for development we plan to use an app development framework called Flutter. This does present a slight challenge, as though several group members have found it interesting, none of the team has any prior experience with that framework. However, the team does have quite a bit of background in app and web development, which should prove invaluable in learning to use Flutter effectively.

### **2. Customer Value**

Since we are developing an app on mobile devices, the primary customers are the app users, the people who want to find connections based on music taste by our mobile app, their input is crucial for making the app as good as it can be for them. The secondary stakeholders are the central/state/local governments, the general public, and the competing connection apps such as Tinder, Bumble, or Hinge. The users of the app want it to work as intended; that is, when they make an account, they will be able to make meaningful matches with people they find interesting. They want to make connections with people, which is why they’re trying out our app. The desired overall experience would be to be able to see other peoples music preferences who are in their area and connect with the people who each user finds interesting. If the other user finds that specific person interesting as well, they can chat with each other and develop a friendship.

Our SMART user stories are as follows:

1. As a potential user, I want to see the marketing page so that I can see what features the app has.
   1. Case 1: The user is not logged in.
      1. Given the user is not logged in, when the user visits the app, then the user should see the marketing page and should not be able to access the app’s features.
   2. Case 2: The user is logged in.
      1. Given the user is logged in, when the user visits the app, then the user should see their homepage and should be able to access the app’s features.
2. As a user, I want to be able to log in to the app so that I can use the app’s features.
   1. Case 1: The user is not logged in.
      1. Given the user is not logged in, when the user clicks on the Log In button on the marketing page, then they should be able to input their username and password and ensure that they are redirected to their dashboard.
   2. Case 2: The user is logged in.
      1. Given the user is logged in, when the user visits the app, then they should be redirected to their dashboard and the Log In button should be replaced with a Log Out button.
3. As a user, I want to be able to log out of the app so that I can keep my account from being accessed by someone else.
   1. Case 1: The user is logged in.
      1. Given the user is logged in, when the user clicks on the Log Out button, then ensure they no longer have access to their account until they log in again and ensure that they are redirected to the marketing page and ensure that the Log Out button is replaced with a Log In button.
4. As a user, I want my personal information to be stored securely so that bad actors cannot infiltrate it.
   1. Case 1: A user has made an account and someone is trying to access it unauthorized.
      1. Given that someone is unauthorized, when they try to access an account, then ensure they are unable to do so.
5. As a new user, I want to be able to create an account so that I can use the app’s features.
   1. Case 1: A user does not yet have an account.
      1. Given that a user does not have an account, when they click on the Create Account button, then ensure they can enter a username and password to create an account.
   2. Case 2: A user has an account.
      1. Given that a user already has an account, when they try to make one with the same email address, then ensure that they are unable to do so.
6. As a user, I want to be able to connect my Spotify and/or Apple Music accounts so that other users can see my music taste and the algorithm can select potential matches for me.
   1. Case 1: Using a Spotify account.
      1. Given that a user wants to use a Spotify account, when they input their account information, ensure that the app has access to their Spotify data and ensure that the algorithm uses the data to suggest potential matches.
   2. Case 2: Using an Apple Music account.
      1. Given that a user wants to use an Apple Music account, when they input their account information, ensure that the app has access to their Apple Music data and ensure that the algorithm uses the data to suggest potential matches.
7. As a user, I want the algorithm to suggest connections with people who match my music taste so that I can make new friends.
   1. Case 1: The user is logged in and has connected a music streaming service account.
      1. Given that the user has connected a music streaming service account, when the user logs in, then ensure that the algorithm checks their music taste and ensure that the algorithm uses that data to suggest accounts with similar tastes.
8. As a user, I want to be able to select and communicate with matches so that I can learn more about them and become friends.
   1. Case 1: User has connected a music streaming service account.
      1. Given that the user has connected a music streaming service account, when they select another user, then ensure that person becomes one of their matches and ensure that they can message with that person in the app.
   2. Case 2: User has not connected a music streaming service account.
      1. Given that the user has not connected a music streaming service account, when they visit their homepage, then ensure that the user cannot see any other profiles.
9. As a user, I want to be able to see and match with other people who aren’t recommended to me.
   1. Case 1: User has connected a music streaming service account.
      1. Given that the user has connected a music streaming service account, when they visit their homepage, then ensure they can see unrecommended people near them and ensure that they can match with those people.
   2. Case 2: User has not connected a music streaming service account.
      1. Given that the user has not connected a music streaming service account, when they visit their homepage, then ensure that the user cannot see any other profiles.
10. As a user, I want to be able to see my suggested matches on my homepage.
    1. Case 1: User has connected a music streaming service account.
       1. Given that the user has connected a music streaming service account, when the user visits their homepage, then ensure that they see suggested matches and ensure that those matches are based on similar music preferences.
    2. Case 2: User has not connected a music streaming service account.
       1. Given that the user has not connected a music streaming service account, when they visit their homepage, then ensure that the user cannot see any other profiles.
11. As a user, I want to be able to get some weekly or monthly recommendations based on what my matches are listening to.
    1. Case 1: The user is logged in and has connected a music streaming service account.
       1. Given the user is logged in, when the user visits the app, then the user should see the recommendations from the homepage.
    2. Case 2: User has not connected a music streaming service account.
       1. Given that the user has not connected a music streaming service account, when they visit their homepage, then ensure that the user cannot see any other profiles.
12. As a user, I want to be able to leave comments or share stories about a song.
    1. Case 1: The user is logged in and has connected a music streaming service account.
       1. Given the user is logged in, when the user visits the page of a song, then the user should see comments from other users and leave his/her own comments.
    2. Case 2: User has not connected a music streaming service account.
       1. Given that the user has not connected a music streaming service account, when they visit their homepage, then ensure that the user cannot see any other profiles.
13. As a user I want to be able to request one of my candidate profiles to communicate with
    1. Case 1: The user has already been matched with the candidate profile
       1. Given that the user has already matched with a candidate profile, when they select that profile, then they may choose to request to communicate with the candidate profile.
14. As a user I want to be able to communicate with a matched profile
    1. Case 1: The user has requested the matched profile to communicate with and they have accepted
       1. Given the user has requested the matched profile to communicate with and they have accepted, when the user tries to communicate with that profile, then a platform for chat will be provided to the two users.
    2. Case 2: The user has requested the matched profile to communicate with and they have not yet accepted.
       1. Given that the user has requested the matched profile to communicate with and they have not yet accepted, when the user tries to communicate with that profile, then ensure a chat platform is not provided to the user.
    3. Case 3: The user has requested to communicate with a non-matched profile.
       1. Given that the user has requested to communicate with a non-matched profile, when the user attempts to communicate with a non-matched profile, then remind the user to send a request to the candidate profile, first.

### **3. Problem Definition**

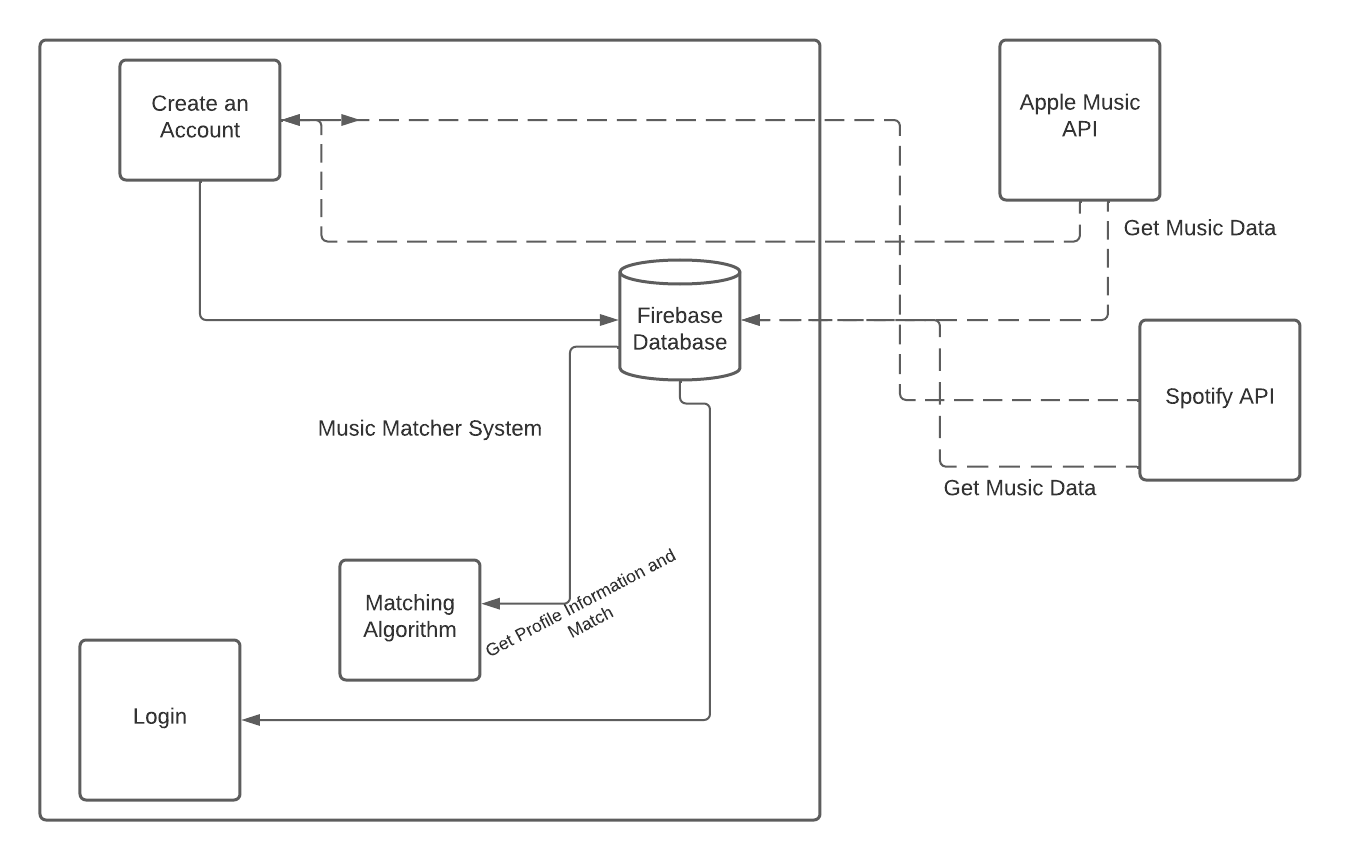
Developing new friendships can at times be difficult especially in current years given the global pandemic. Being able to develop friendships in a safe and unique way gives more opportunity to create long lasting relationships. Because of this, we have decided to address the problem of making meaningful friendships at a time when making friends in person is still difficult and finding things you have in common can be more difficult still. The system will benefit the user by providing the opportunity to connect with people with similar interests and preferences in music. Providing the opportunity of meeting people of the same tastes is a very personalized service, which means we are trying to understand every single user’s preferences and needs, and it obviously will help support users’ desired overall experience.

Outside of the team, we have tested the idea on several family members and friends. They have all been accepting of the idea, and a couple even offered to test it out once we get the app running. The real outside customer, though, is anyone who likes music and would like to reach out and make friends with similar tastes. To check whether the customers get their desired benefits, a well-designed survey can be conducted to know users' experiences and feedback on the product. Also, the records of users actions can provide insights into how much time users spend their time on and how users interact with the product.

Our customer-centric measures of success are as follows:

1. The time users spend on the app
2. The times users open the app
3. The number of matches users started a conversation with among all matches
4. The number of messages users send to their matches
5. The time a conversation lasts (i.e. the time difference between the first message and the latest message)
6. The number of times users click on the lists/songs recommended

### **4. System Description**



The system’s main elements are the backend codebase, consisting of the machine learning algorithm to suggest matches based on music taste as well as general backend code; the frontend codebase, consisting of the UI; the account database, which holds user account information; and the external services, which are Spotify and Apple Music and shall be interacted with by us via API calls.

### **5. Solution Approach**

Our app will provide its users with the opportunity to find people with similar taste in music. Users authorize the app to connect to their Spotify and/or Apple Music profiles. The system will use APIs provided by these platforms to access various user data, such as most recent/liked music tracks, favorite artists, etc. Based on user data, the system will create a profile and keep and update in a database. Ultimately, our database will associate each user with their music profile and other identifying information, such as GPS location, age, etc. In the backend, the system will conduct machine learning solutions such as clustering to identify similar profiles. After finding profiles with adequate similarity, the system will recommend the user with the ability to connect to these profiles. The user will also be able to filter recommendations based on their locations. Recommended users can then try to communicate through a chat interface. Users will also be shown a selection of users that are close to their location but not recommended by the algorithm to go through. If time permits, we will also try to gather feedback from user experience and tune our ML solution to recommend better candidate profiles as our app evolves. We are planning to use Flutter as the framework for our app. In addition, we will use the Spotify and Apple Music APIs to connect to them as external services, and Google Firebase for our account database. At this point, we do not anticipate using any other platforms or tools, but this might change in the future. As for testing, Flutter has testing packages available that we plan to look into and utilize in order to make testing easier and more complete. We plan to make sure each part of our code has unit tests to verify that everything is working properly. The adequacy of that test strategy shall be evaluated by thoroughly integration testing from the user’s point of view; if the app works for the user as intended, we shall know our testing strategy worked.

### **6. Project Management**

*Changelog Link:* [*https://docs.google.com/spreadsheets/d/1H8TjAgGXaTTaaAIvyOuuYWwSE5\_7cy0iwTG5hW02yTw/edit?usp=sharing*](https://docs.google.com/spreadsheets/d/1H8TjAgGXaTTaaAIvyOuuYWwSE5_7cy0iwTG5hW02yTw/edit?usp=sharing)

We are planning to use an Agile development process with Scrum. The minimal working system that we plan to build initially is an app with a very basic UI that enables users to login, link their spotify account, and see people within their area.

Our feature wish list is as follows:

* A user can login and create an account
* Each account has linked a music service that shows each individual's music preferences.
* Each user can see other users in a range that they set
* Other users that are shown in their area are shown in order of most compatible to least compatible
* You can like or move on from other users that you are interested in
* If two users like each other they can chat with each other privately

For team coordination, work items will be created and weighted based on the amount of time each person thinks their tasks will take them. Each person will have work items that add up to around 4 hours per week. Total hours worked per week will be adjusted based on the amount of work that needs to be done. We will meet once a week over zoom, which will act as Weekly Scrum meetings. As such, during the meetings we will go over what we have done during the week, if we are stuck on anything, and what we will do for the next week.

### **Team**

Our team is fairly varied in interests and background. Ray (Zirui) is a current Ph.D. student majoring in Transportation and minoring in CS. His research interest is traffic simulation, traffic predictions, and travel demand management strategies. He has experience in developing traffic simulators and applying machine learning algorithms in research, and in prototyping multimodal trip planner and cycling routing engine in industry during internship. But he doesn’t have experience in building a mobile app.

Amy is a current master’s student studying Computer Science. She worked as a web software developer for about a year and a half, and thus has quite a bit of applicable experience for this project. In addition, she has some machine learning and cybersecurity research experience. She has never built a full mobile app nor used Flutter, but both should be easy to pick up.

Amir is a current PhD student with the ECE department and is minoring in CS. His research is focused on machine learning solutions for wireless communication problems. He has experience with developing discrete event simulators and machine/deep learning algorithms. This is Amir’s first experience with software engineering.

Mario is a current master’s student at the University of Arizona studying Computer Science. He is currently working at Microsoft as a software engineer. He has some experience building Android apps, but has never used flutter.

Each person in the group will act as a developer during the duration of the project. However, we will rotate who is scrum master each week, and during non-Weekly Scrum meetings, someone will act as the stakeholder/user to help develop a backlog and prioritize work items.

### **Constraints and Risks**

The ethical and legal constraint for our app shall be that each user will need to be over 18 to use it. Even though this isn’t explicitly a dating app, we still want to minimize any nefarious activity and restricting users to being over 18 will help with this. Overall, we shouldn’t have any data risks. We will have access to the data, services, and resources we need through the use of Flutter, Firebase, and APIs. All these services have plenty of available resources to give us all the data and tools we would want.