EE / CPR E/ SE 491 - Weekly Report #4

Dates: 3/2/2020 - 3/15/2020

Group Number: sddec20-13

Project Title: My (Musical) Life

Client & Advisor: Dr. Henry Duwe

Team Members:

Christian Hernandez - Project Manager

Chaz Clark - iOS Developer

Daksh Goel - Backend Developer

Vatsal Bhatt - Backend Developer

Vignesh Krishnan - Frontend Developer

Weekly Summary

Throughout the past two weeks, the team has continued research into APIs, frameworks, tools, databases, etc. that the team plans on using. The team also worked towards a diagram displaying how bins will be created, a database design diagram, a user feedback diagram, and an onboarding diagram. Additionally, the team created a table that consists of a list of APIs, the data type the APIs will return, and a brief description of how the APIs will be useful for the team. Additionally, the team met with its advisor once during the week to discuss updates. The plan is to begin building a prototype of the application soon.

Past Week Accomplishments

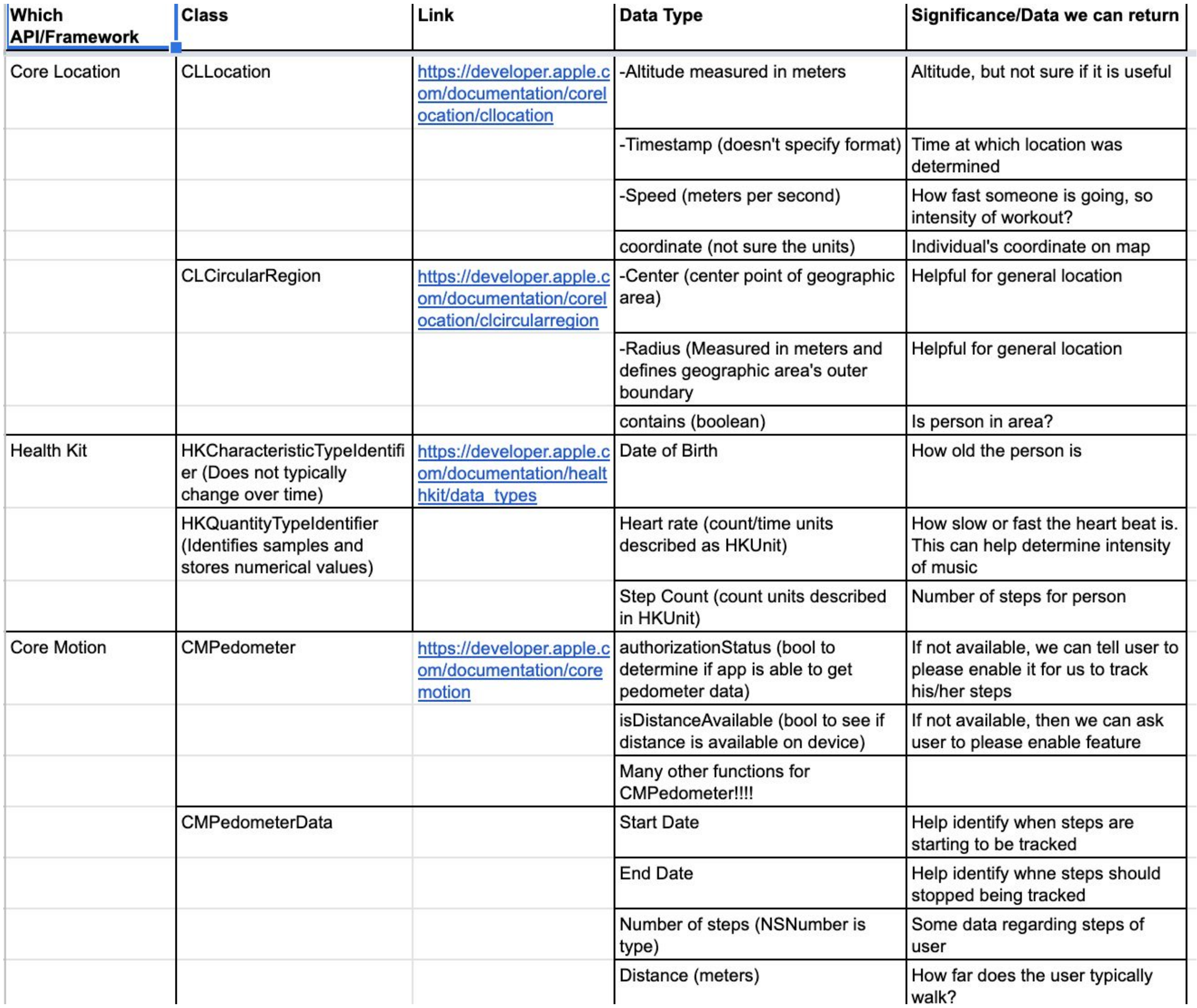
- Christian

- Continue to keep track issues in GitLab and create new ones as needed
- Research in DynamoDB and Lambda
    - Christian is not too experience with using AWS, thus research into AWS was necessary in order to understand how the team can use DynamoDB and Lambda
    - DynamoDB
        - DynamoDB is a “key-value and document database that delivers single-digit millisecond performance at any scale” (“DynamoDB: Everything You Need to Know About Amazon Web Service's NoSQL Database”).
        - DynamoDB is serverless and fast (“DynamoDB: Everything You Need to Know About Amazon Web Service's NoSQL Database”).
        - Offers a “free tier” (“DynamoDB: Everything You Need to Know About Amazon Web Service's NoSQL Database”).
    - Lambda
        - “AWS Lambda lets you run code without provisioning or managing servers. You pay only for the compute time you consume” (“AWS Lambda”).
        - AWS Lamda does not require administration (“AWS Lambda”).
        - No servers that needs to be managed (“AWS Lambda”).
        - There is a “free tier” offered (“AWS Lambda”).
        - We will only have to pay for the compute time (“AWS Lambda”).

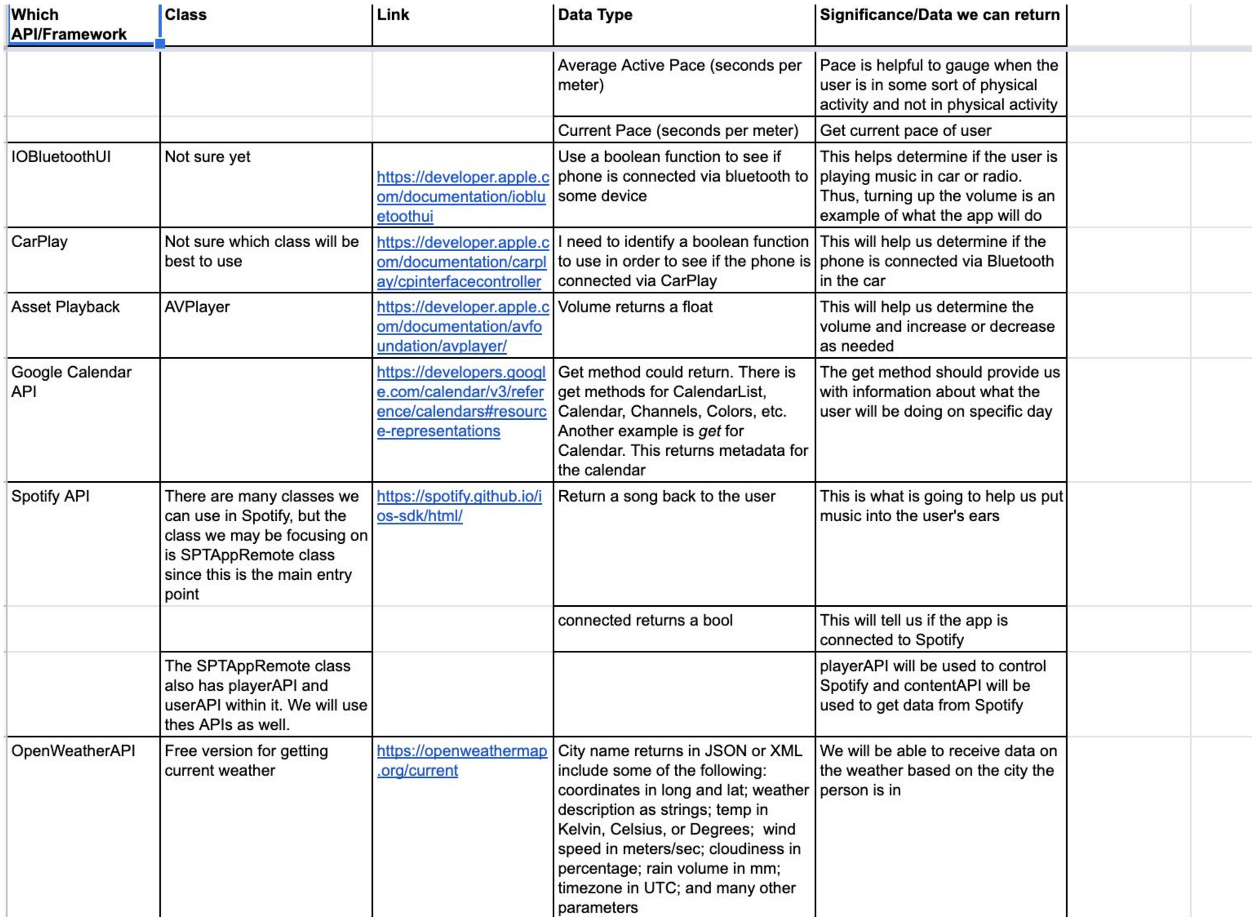
- Lambda can “retrieve and transform data from Amazon DynamoDB” (“AWS Lambda”).
- AWS Lamda keeps the code secure by encrypting it at rest and stores in Amazon S3 (“AWS Lambda”).

- Created a table dedicated to a list of APIs we will be using and information regarding the APIs
  - This table consists of some example methods/calls we will be using for our app. Each API has numerous classes that include many methods that we can use. I decided to grab the ones that I know that we will use for sure. However, please note, there many other methods within these APIs we can use.
  - The following APIs (and one framework) are what the team currently plans on using
    - Apple Developer APIs
      - Core Location
      - Health Kit
      - Core Motion
      - IOBluetoothUI
      - CarPlay
      - Asset Playback
    - Google
      - Google Calendar API

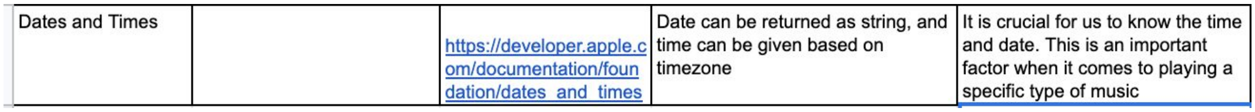
- Spotify
    - Spotify API for iOS development
- OpenWeatherAPI
- Apples Dates and Times framework
        - Below is are screenshots of what the table looks like



Which API/Framework	Class	Link	Data Type	Significance/Data we can return
Core Location	CLLocation	https://developer.apple.com/documentation/corelocation/cllocation	-Altitude measured in meters	Altitude, but not sure if it is useful
			-Timestamp (doesn't specify format)	Time at which location was determined
			-Speed (meters per second)	How fast someone is going, so intensity of workout?
			coordinate (not sure the units)	Individual's coordinate on map
	CLCircularRegion	https://developer.apple.com/documentation/corelocation/clcircularregion	-Center (center point of geographic area)	Helpful for general location
			-Radius (Measured in meters and defines geographic area's outer boundary	Helpful for general location
			contains (boolean)	Is person in area?
Health Kit	HKCharacteristicTypeIdentifier (Does not typically change over time)	https://developer.apple.com/documentation/healthkit/data_types	Date of Birth	How old the person is
	HKQuantityTypeIdentifier (Identifies samples and stores numerical values)		Heart rate (count/time units described as HKUnit)	How slow or fast the heart beat is. This can help determine intensity of music
			Step Count (count units described in HKUnit)	Number of steps for person
Core Motion	CMPedometer	https://developer.apple.com/documentation/coremotion	authorizationStatus (bool to determine if app is able to get pedometer data)	If not available, we can tell user to please enable it for us to track his/her steps
			isDistanceAvailable (bool to see if distance is available on device)	If not available, then we can ask user to please enable feature
			Many other functions for CMPedometer!!!!	
	CMPedometerData		Start Date	Help identify when steps are starting to be tracked
			End Date	Help identify whne steps should stopped being tracked
			Number of steps (NSNumber is type)	Some data regarding steps of user
			Distance (meters)	How far does the user typically walk?

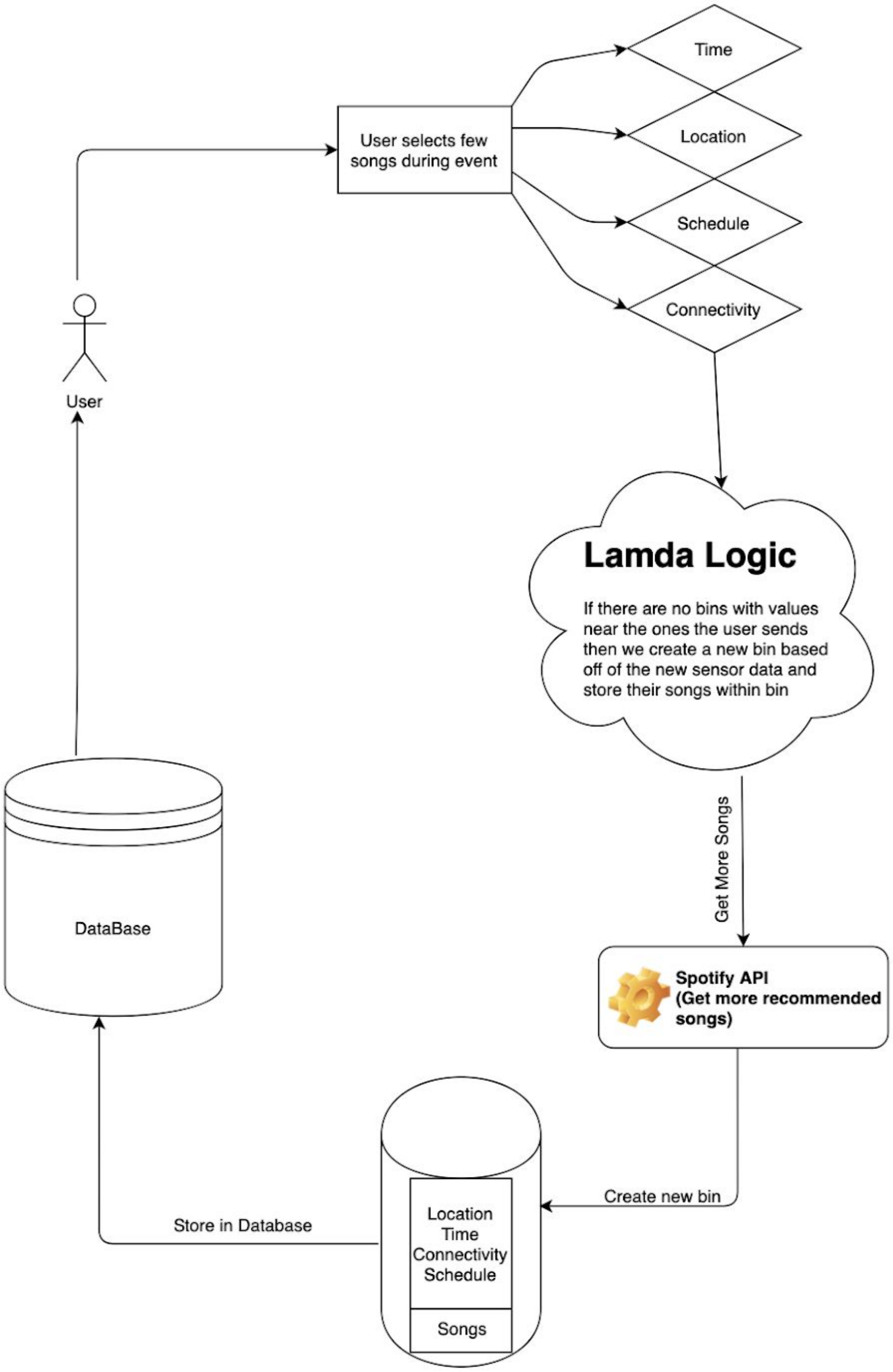


Which API/Framework	Class	Link	Data Type	Significance/Data we can return
			Average Active Pace (seconds per meter)	Pace is helpful to gauge when the user is in some sort of physical activity and not in physical activity
			Current Pace (seconds per meter)	Get current pace of user
IOBluetoothUI	Not sure yet	https://developer.apple.com/documentation/iobluetoothui	Use a boolean function to see if phone is connected via bluetooth to some device	This helps determine if the user is playing music in car or radio. Thus, turning up the volume is an example of what the app will do
CarPlay	Not sure which class will be best to use	https://developer.apple.com/documentation/carplay/cpinterfacecontroller	I need to identify a boolean function to use in order to see if the phone is connected via CarPlay	This will help us determine if the phone is connected via Bluetooth in the car
Asset Playback	AVPlayer	https://developer.apple.com/documentation/avfoundation/avplayer/	Volume returns a float	This will help us determine the volume and increase or decrease as needed
Google Calendar API		https://developers.google.com/calendar/v3/reference/calendars#resource-representations	Get method could return. There is get methods for CalendarList, Calendar, Channels, Colors, etc. Another example is *get* for Calendar. This returns metadata for the calendar	The get method should provide us with information about what the user will be doing on specific day
Spotify API	There are many classes we can use in Spotify, but the class we may be focusing on is SPTAppRemote class since this is the main entry point	https://spotify.github.io/ios-sdk/html/	Return a song back to the user	This is what is going to help us put music into the user's ears
			connected returns a bool	This will tell us if the app is connected to Spotify
	The SPTAppRemote class also has playerAPI and userAPI within it. We will use thes APIs as well.			playerAPI will be used to control Spotify and contentAPI will be used to get data from Spotify
OpenWeatherAPI	Free version for getting current weather	https://openweathermap.org/current	City name returns in JSON or XML include some of the following: coordinates in long and lat; weather description as strings; temp in Kelvin, Celsius, or Degrees; wind speed in meters/sec; cloudiness in percentage; rain volume in mm; timezone in UTC; and many other parameters	We will be able to receive data on the weather based on the city the person is in



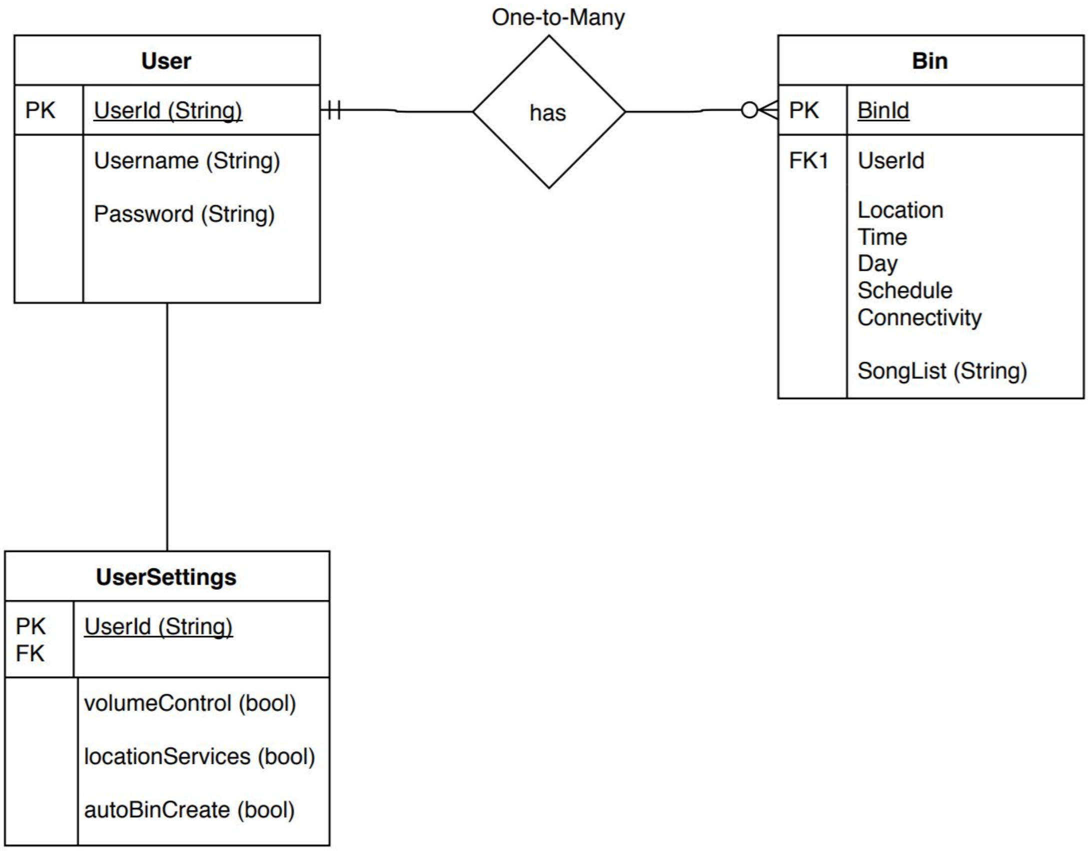
Dates and Times		https://developer.apple.com/documentation/foundation/dates_and_times	Date can be returned as string, and time can be given based on timezone	It is crucial for us to know the time and date. This is an important factor when it comes to playing a specific type of music

- Chaz
  - Built “Bin Creation Diagram”

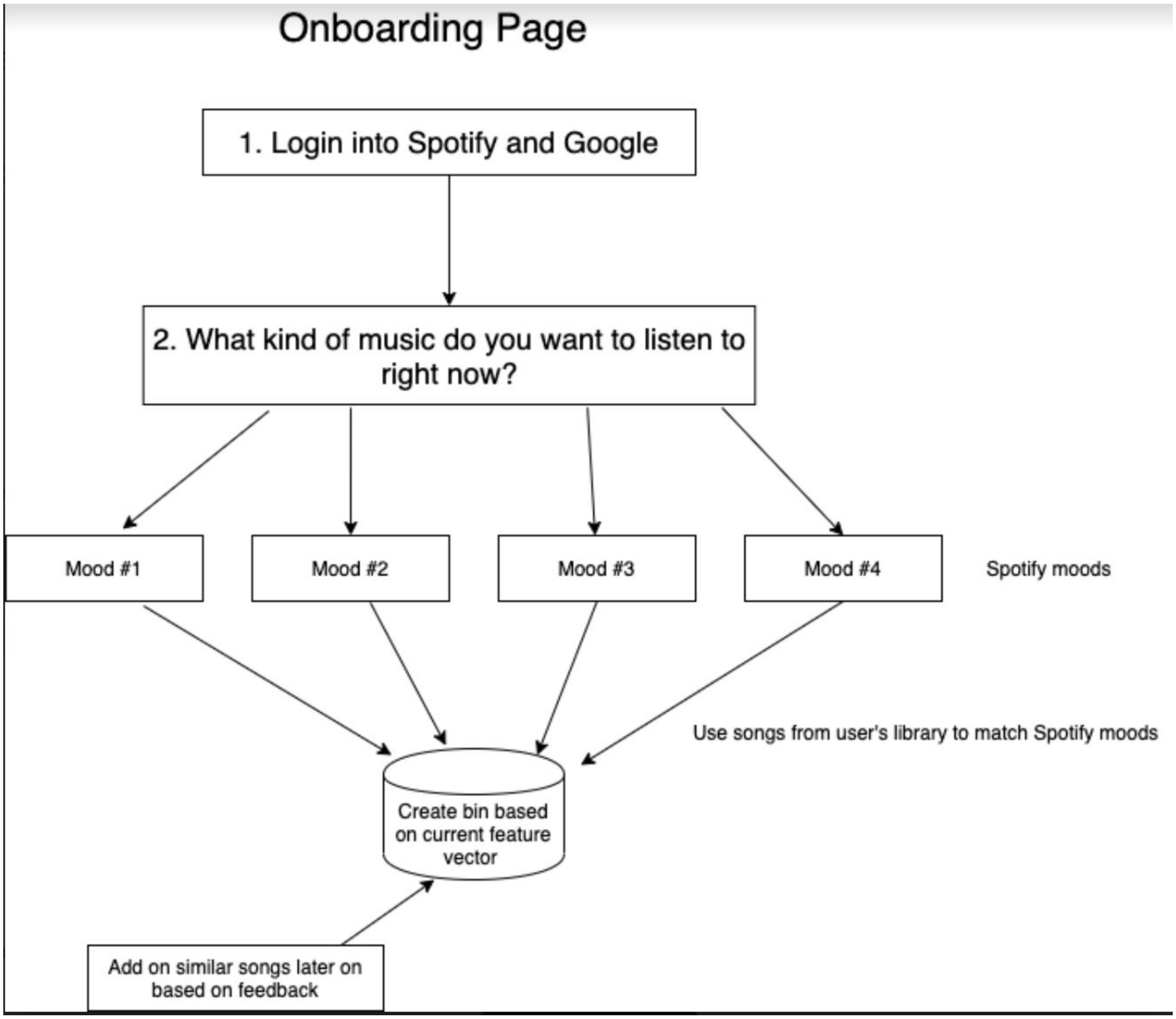


If there are no bins with values near the ones the user sends then we create a new bin based off of the new sensor data and store their songs within bin

- Daksh
    - Learning AWS (Dynamo DB, Lamba)
        - Created a User table and a controller method to get data
    - Built initial design for database

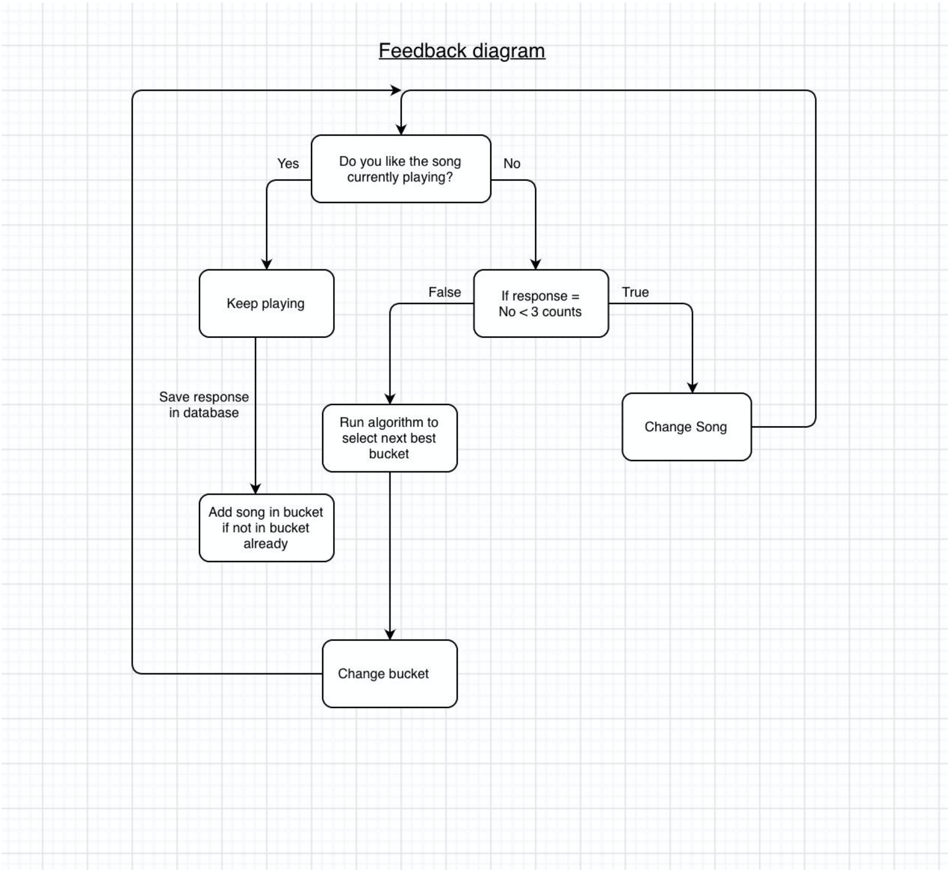


- Vignesh
    - Built onboarding diagram with initial steps on how we will onboard a user and create their initial bins


Onboarding Page
1. Login into Spotify and Google
2. What kind of music do you want to listen to right now?
Mood #1
Mood #2
Mood #3
Mood #4
Spotify moods
Use songs from user's library to match Spotify moods
Create bin based on current feature vector
Add on similar songs later on based on feedback

- Vatsal
    - Designed a diagram on User Feedback Response, given, on of the features of our application. It serves as a decision-making process on what action to take next based on the user response

      Here is the diagram:



Feedback diagram
Yes
Do you like the song currently playing?
No
Keep playing
False
If response = No < 3 counts
True
Save response in database
Run algorithm to select next best bucket
Change Song
Add song in bucket if not in bucket already
Change bucket

- Continued learning AWS and how to integrate it’s services (DynamoDB and Lambda).

Pending Issues

There are currently no pending issues.

Individual Contributions

NAME	INDIVIDUAL CONTRIBUTIONS	HOURS THIS WEEK	HOURS CUMULATIVE
Christian Hernandez	1. Conducted research	10	16

	into APIs we will need to use for our app 2. Created a table with descriptions of APIs and what we expect to get from the APIs 3. Contributed to updating GitLab 4. Conducted some research into AWS (Dynamo DB and Lambda)		
Chaz Clark	1. Built Bin Creation Diagram 2. Researched into Apple APIs to retrieve appropriate data 3. Designed Bin associations & attributes	6	12
Daksh Goel	1. Build DB design 2. Learning DynamoDB and Lamda 3. Researched common practices for dbs	6	12
Vatsal Bhatt	1. Designed User Feedback Response diagram 2. Came up with action prediction depending on user feedback 3. Continued learning AWS DynamoDB and Lambda	6	12
Vignesh Krishnan	1. Built onboarding	6	12

	diagram 2. Initial steps to how user will be onboarding and bins created		

Comments and extended discussions (Optional)

Plans for the Upcoming Week

- Christian
  - Follow DynamoDB and AWS Lambda tutorials
  - Assist with research and the beginning stages of development in both the frontend and backend
  - Update GitLab update to reflect the current status
  - Begin learning how to integrate APIs into the app
- Chaz
  - Dive deeper into Bin Creation models
  - Test Apple API’s
  - Build proof of concepts
- Daksh
  - Continue to learn AWS services
  - Build functions to interact with our DB
  - Continue to research Spotify and iOS integration

- Vatsal
    - Dive deeper into User feedback response actions.
    - Build a AWS Lambda and DynamoDB pipeline as a prototype
    - Grow Swift coding skills
- Vignesh
    - Work on music prediction (decision making) algorithm prototype
    - Dig deeper into Music bin creation models
    - Begin learning on how to integrate third party APIs in Swift

Summary of Weekly Advisor Meeting

For this reporting period, we only met with Dr. Duwe once due to the recent events with Covid-19. For our second meeting, we provided Dr. Duwe with updates via email.

For our first meeting, we discussed using feature vectors as a way to pass information into our bins. The bins will be used to store music for the user. During that meeting, Dr. Duwe asked us to create a figure to display the process of how the bins will work. As a result, Chaz created a diagram to display the process of how the bins will work. Next, Dr. Duwe advised us to begin to think about building a prototype. He wants us each to begin researching something different. For example, 1 or 2 people digging into ML, 1 person researching APIs, and so on. Dr. Duwe also wanted the team to explicitly list the APIs the team plans on using and what type of data the APIs will return. Therefore, Christian created a table with a list of APIs that the team will use along with the type of data the APIs will return. The table includes brief descriptions of the significance of the results from the APIs, as well. Lastly, Dr. Duwe advised the team to begin thinking of a backup plan. For example, what will the team do if the user does not like it? What is the risk the user will not like it? Overall, the meeting was successful, and the team plans on taking Dr. Duwe’s advice. Thus, the team will begin building the prototype and building a backup plan as soon as possible.

References

“AWS Lambda.” Amazon, aws.amazon.com/lambda/.

“DynamoDB: Everything You Need to Know About Amazon Web Service's NoSQL Database.” Amazon, 2015, aws.amazon.com/dynamodb/.