

EE / CPR E/ SE 491 - Weekly Report #3

*Dates:* 2/17/2020 - 3/1/2020

*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**

Over the past two weeks our team transitioned into designing how we will approach our prediction process. We developed high level diagrams on how our modules will interact with each other to formulate music predictions for our users. We have been doing extensive research into online tools and platforms that can help us achieve this process. Amazon Web Services is a tool we found that will be very useful in this decision making process.

### **Past Week Accomplishments**

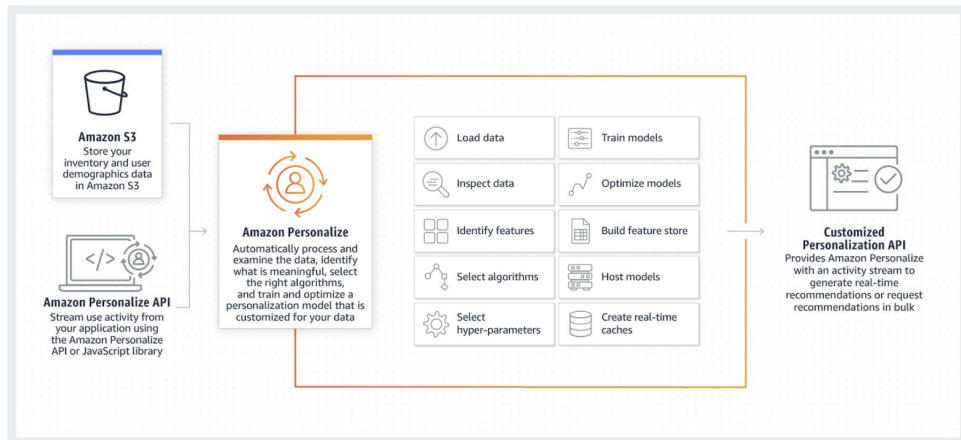
- Christian
  - Continue to keep track issues in GitLab and create new ones as needed
  - Research into Swift User Interfaces

- Apple has recently come out with SwiftUI. This can be used across all platforms
  - This is in contrast to using Appkit for MacOS, UIKit for iOS and tvOS
- SwiftUI can only be used on devices with iOS 13 or later
- It is recommended to use SwiftUI for newer apps, but knowledge and training of UIKit are just as important since the older apps with an iOS version of below 13 were built with UIKit (“SwiftUI vs. UIKit”).
- Since SwiftUI is beta software, it may be difficult to use (Bower).
- Found multiple tutorials to follow to build a simple UI, such as
  - <https://developer.apple.com/library/archive/referencelibrary/GettingStarted/DevelopiOSAppsSwift/BuildABasicUI.html>
- Swift Networking Research
  - The two frameworks that the team can use are Alamofire or URLSession
  - Alamofire Notes
    - Open-source (Van Der Lee 1)
    - Layer on top of URLSession (Van Der Lee 1)
    - It is said that Alamofire can save time and easier to use (Van Der Lee 1)
    - Saves a lot of extra code (Van Der Lee 1)
    - Smaller framework (Van Der Lee 1)
  - URLSession

- Part of Foundation framework (Van Der Lee 1)
- Provides an API “for downloading data from and uploading data to endpoints indicated by URLs” (“NSURLSession” 1).
- Can perform background downloads when the app is not running (“NSURLSession” 1).
- “An object that coordinates a group of related, network data-transfer tasks” (“NSURLSession” 1).
- This website (along with many others) provide a tutorial one can follow for URLSession
  - <https://www.raywenderlich.com/3244963-urlsession-tutorial-getting-started>
- Overall, based on opinions from others, URLSession seems to be a bit more difficult to use
  - Continued to update website with required documents
- Chaz
  - Researched into Amazon Web Services
    - Amazon Web Services has many useful services that will help our team reach its goal in predicting music for our users. One of these services is [Amazon Personalize](#). This service uses Amazon S3 file storage, Lambda code hosting, and machine learning to build a detailed profile about each user to better predict what a user wanted and what they will do in future events.

- Assisted in High Level Prediction Process Design
  - We began to design a layout of how we will implement our components in predicting user music. The design we created utilizes users feedback and sensor data as input parameters for prediction. This data will be sent to our AWS component of our application where we will use ML or decision trees to predict a users song based off of what song bucket they fall into. We will store their output into a database to build a detailed record on what they like and call our Spotify API to retrieve a song they might want to listen to next.

**How it works**



- Daksh

- Helped design a high-level process for prediction
  - Designed a system that created “buckets” that will help us choose which music to recommend. These buckets will be stored on our AWS db and we will build a “profile” for each user
- Research on Mood Prediction models

- Researched possible ways for us to predict mood. Went through research papers on mood prediction and found a couple different ways others have gone about it. One paper did mood prediction based on lyrics of songs. However, it seems that most of these previous research papers will not be able to aid us because our input parameters may not be sufficient to use open source models.
  
- Vatsal
  - Research into Amazon S3 and Personalize
    - Looked into and read about how Amazon personalize (a MachineLearning service) could possibly be used in our project as a music recommendation method.
    - Read on how to incorporate Amazon S3 (cloud storage service) with Amazon personalize if we decide to move forward with this method.
  - Helped the team to design a high-level process for prediction
    - Designed a system diagram involving mood buckets and a high level data transfer sketch for us to visualise the music recommendation method.
  - Swift and API integration
    - Did some tutorials to learn how to integrate third party APIs into swift, especially Spotify API
  
- Vignesh
  - Decision Making Algorithm Research

- Brute force tree-based algorithm for playing genre/playlist based on mood, if use cases are explicitly defined
- Updating user profile with different use cases as they arise and therefore updating tree-based algorithm's cases

### ML Algorithm Research

- Need existing data to implement training/testing data set prediction model
- Could compile training set of overlapping use cases for a user and the resulting music genre that they played from that use case

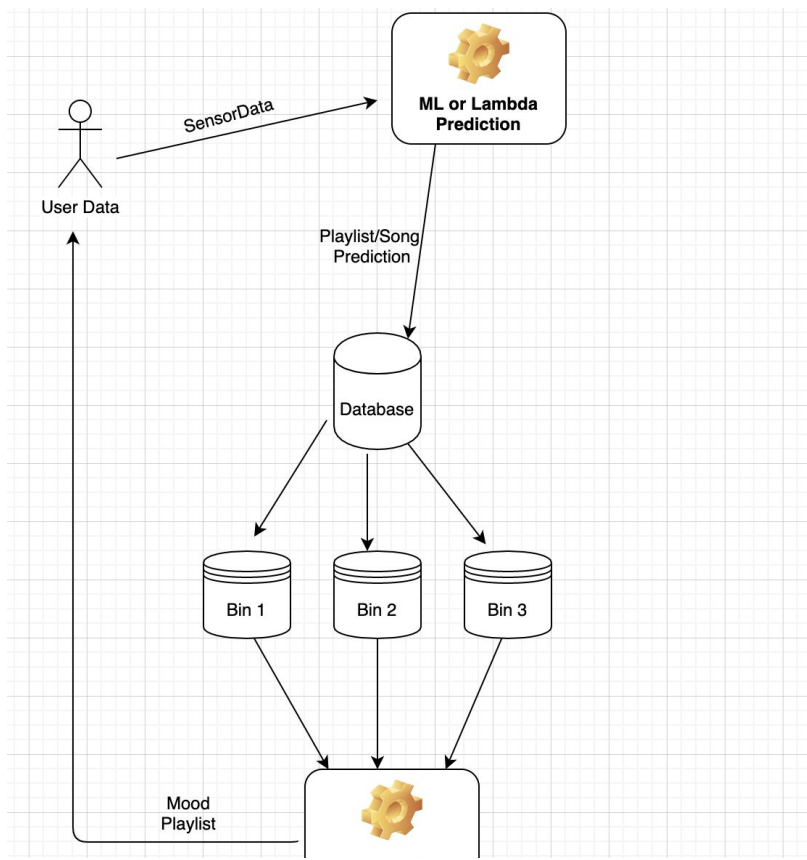
Helped the team to design a high-level process for prediction

- Designed a system diagram involving “mood buckets” and a high level data transfer sketch for us to visualise the music recommendation method.

### Swift Tutorials

- Continued on swift UI tutorials to learn how to constrain objects for UI

- Prediction Diagram



- As Chaz noted earlier, the team discussed the design of the prediction process.

The picture placed above is what the team designed recently. Please note that this is a draft, but this is what the team envisions how the prediction will take place.

Sensor data from the user will be sent to the ML or Lambda Prediction. Currently, the team is leaning towards using Amazon Personalized to generate the prediction. Then, the result from the ML or Lambda Prediction will be the Song/Playlist Prediction. This prediction will head to a database and be stored into their respective bins. The bins will ultimately be playlists for the user. The user will have the ability to modify these bins. Then, the Spotify API will play one of these playlists for the user.

### **Pending Issues**

There are currently no pending issues.

### **Individual Contributions**

<b>NAME</b>	<b>INDIVIDUAL CONTRIBUTIONS</b>	<b>HOURS THIS WEEK</b>	<b>HOURS CUMULATIVE</b>
Christian Hernandez	<ol style="list-style-type: none"> <li>1. Research UI for Swift</li> <li>2. Research how to perform networking in Swift               <ol style="list-style-type: none"> <li>a. Alamofire vs. URLSession</li> </ol> </li> <li>3. Contributed to the delegation of tasks and assisted with</li> </ol>	8	15

	<p>updated issues in GitLab</p> <p>4. Uploaded Design Document to website</p>		
Chaz Clark	<ol style="list-style-type: none"> <li>1. Amazon Personalize Research.</li> <li>2. High level diagram for Decision making process</li> <li>3. Collected Apple iOS development tutorials and documentation to guide team through future development</li> </ol>	6	12
Daksh Goel	<ol style="list-style-type: none"> <li>1. Research on ML mood prediction</li> <li>2. High Level diagram for mood prediction on AWS</li> <li>3. Learn Swift</li> </ol>	6	12
Vatsal Bhatt	<ol style="list-style-type: none"> <li>1. Research on Amazon Personalize and S3</li> <li>2. High Level diagram for mood prediction on AWS</li> <li>3. Swift and API integration</li> </ol>	6	12
Vignesh Krishnan	<ol style="list-style-type: none"> <li>1. Research on decision making algorithm</li> <li>2. Research on ML algorithm for predicting mood</li> <li>3. Swift practice</li> </ol>	6	12



## Comments and extended discussions (Optional)

### Plans for the Upcoming Week

- Christian
  - Continue research with Swift network, specifically Alamofire
    - Based off of research, Alamofire seems to be the best framework to use when it comes to networking in Swift
  - Continue research and begin tutorials surround UI in Swift
    - SwiftUI is new and recommended to use for newer apps. Thus, following tutorials that teach one how to use SwiftUI
    - It is recommended to learn and follow tutorials teaching UIKit as well. Thus, the goal is to get experience using UIKit
  - Continue to learn how to develop in Swift using the book that Chaz provided the team
  - Learn more about Amazon Personalization. This may potentially be what the team uses to help with determining the mood and playlist to generate
- Chaz
  - Build small test project using Amazon Personalize to see how useful the service can be

- Design a more detailed diagram on our prediction model and how we can design our database
- Daksh
  - Continue to learn Swift and create a connection with Spotify
  - Modify our prediction design, based on feedback we received from Dr. Duwe
- Vatsal
  - Continue research and try examples for Amazon S3, and DynamoDB
  - Continue to learn Swift UI design and API integration
  - Decide whether Amazon personalize would be a viable option for music recommendation
- Vignesh
  - Decide what information will go into our tables for use case prediction
  - Continue to research into back up brute force decision making algorithm that doesn't use any ML prediction
  - Continue with Swift tutorials
  - Research how we can use Spotify song API data in our prediction model

### **Summary of Weekly Advisor Meeting**

Our meeting with Dr. Duwe was canceled on 2/22. However, we did meet with Dr. Duwe on 2/28. During this meeting, the largest topic discussed was around the topic of whether we would predict the mood or the song for the user. After a lengthy conversation with Dr. Duwe, we ultimately decided it would be best to predict the song rather than the mood. This is because predicting mood can be quite difficult. Therefore, the approach we decided we would take is to

create bins for the user based on user input. These “bins” will ultimately be playlists for the user. These bins can be altered by the user as well. Then, the bins will be played based on the machine learning we implement or a brute force algorithm we create. The user will have the freedom to change the bin if he/she chooses to do so. Ultimately, based on the factors, such as the user calendar, location, etc., the application will predict which bin of music to play.

Additionally, Dr. Duwe would like for us to construct a table that consists of sensor inputs and the data they provide. For example, what metadata can we receive from the user’s calendar? Thus, we will create this table for Dr. Duwe by our next meeting on Friday, March 6th.

## References

Bower, Graham. “Why SwiftUI Might Be the Biggest Thing to Come out of WWDC.” Cult of Mac, 7 June 2019, [www.cultofmac.com/629988/swiftui-wwdc-2019/](http://www.cultofmac.com/629988/swiftui-wwdc-2019/).

“SwiftUI vs. UIKit.” Raywenderlich.com, [www.raywenderlich.com/4919757-your-first-ios-and-swiftui-app/lessons/2](http://www.raywenderlich.com/4919757-your-first-ios-and-swiftui-app/lessons/2).

“URLSession.” URLSession - Foundation | Apple Developer Documentation, Apple Inc., [developer.apple.com/documentation/foundation/urlsession](https://developer.apple.com/documentation/foundation/urlsession).

Van Der Lee, Antoine. “Alamofire vs URLSession: A Comparison for Networking in Swift.” SwiftLee, 8 May 2019, [www.avanderlee.com/swift/alamofire-vs-urlsession/](http://www.avanderlee.com/swift/alamofire-vs-urlsession/).

