



Senior Design Project

Moodio

Final Report

Ahmet Akif Uğurtan, Mehmet Taha Çetin, Ulaş İş, Esra Nur Ayaz, Ahmet Ensar

Supervisor: Cevdet Aykanat

Jury Members: H. Altay Güvenir, Hamdi Dibekliolu

Innovation Expert: Duygu Gözde Tümer

May 9, 2019

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Senior Design Project course CS491/2.

1. Introduction	2
2. System Overview	3
3. Final Architecture and Design	3
3.1. Overview	3
3.2. Subsystem Decomposition	4
4. Impact of Engineering Solutions	5
4.1. Global Impact	5
4.2. Social Impact	6
5. Contemporary Issues	6
5.1. Face Recognition	6
5.2. Natural Language Processing	6
5.3. Cloud Platforms	6
5.4. React Native	7
6. Use of Libraries and Resources	7
7. New Tools and Technologies Used	8
7.1. React Native	8
7.2. Google Cloud	8
7.3. Microsoft Azure	9
7.4. Spotify API	9
7.5. Node JS / NPM	9
7.6. Git / GitHub	9
8. User Manual	10
8.1 Login Screen	10
8.2 Homepage	11
8.3 Camera Screen	12
8.4 Audio Screen	13
9. Bibliography	14

1. Introduction

With the increasing number of songs in the world, people are in a trouble of discovering new music and choosing what to listen. They especially tend to listen to music according to their mood and emotions since music is seen as a relief and escape from the real world. Moodio is an application that recommends people music playlists according to their moods. It will recognize the mood of people from their voice and face using the microphone and the camera of the phone. Some researches shows that facial expressions play an important role in recognition of emotions and are used in the process of non-verbal communication as well as to identify people. They are very important in daily emotional communication just next to the tone of voice. They are also indicator of feelings, allowing a man to express an emotional state. People can immediately recognize an emotional state of person. Therefore, the facial expressions and voice tone can be considered as a best way to determine the mood of people.

With the help of facial expressions and voice tone, we developed an Android and iOS application called Moodio that will suggest the best music playlist according to the mood of users. By using Moodio, the users will be able to login to the application through Spotify, take their pictures and record their audio by the application and then Moodio determines the mood of the user by the picture and audio and will recommend a playlist to the user considering the current music taste of the user obtained from Spotify as well as the mood of the user. And then, the playlist is saved to the user's account and user will be directed to Spotify application and they will be able to listen to those playlists.

In this report, we will present the final status of our project. We will describe the system overview. Then, we will talk about the final architecture and design of Moodio. After that, we will discuss impact of engineering solutions implemented in our project. We then talk about the contemporary issues related to our project. Moreover, we will describe how we made use of which resources as well as which new tools and technologies we used while developing Moodio. Then, we will talk about possible future developments that can be made to improve our project. Finally, we will give the user's manual for Moodio to describe how it is used to new users.

2. System Overview

Imagine an ordinary day of people. Listening to music is something people do often in an ordinary day in order to escape the real world. However, people spend their time searching for more than few minutes to listen two minutes songs. Using the current applications on the market, they have to search a songs one by one manually and create playlist based on their moods. When we consider all these process of finding songs which people listen to for a few minutes, it is too cumbersome.

Moodio is offers a quick solution to this process. Moodio detects the mood each of user with the help of images and audio records that is obtained using microphone and camera of the mobile phone. Then according to the mood of user, it recommends the playlist quickly.

Each user who have spotify account can use Moodio. If they don't have spotify account, they can also create one from the login screen with the help of spotify application.

After logging in the application, each user will be presented with a list of their recommended songs, their favorite singers based on the songs listened by user and their playlists. Also at the bottom of the page, there are two options which camera and audio, are provided for the user to determine their mood by the application. User can prefer either camera screen or audio screen to detect their own mood. In camera screen user can take a picture of their own face and create the playlist based on the mood that is determined by the Moodio. After the taking photo process, Moodio find the correct mood and create the playlist in the Spotify account of user automatically. Also according to the mood of user and the singers who is preferred by user most frequently, Moodio offers playlist to user. Moreover, user can see the all playlist that is created based on their mood.

3. Final Architecture and Design

3.1. Overview

In the subsystem decomposition, the subsystem structure of Moodio is described. Moodio follows a Client-Server architectural style. Nearly all work is done in client side, therefore we have a simple server and a rather complicated client. All needed data is

gathered in the client side and then send to cloud services and Spotify. We only use server for login process to Spotify.

3.2. Subsystem Decomposition

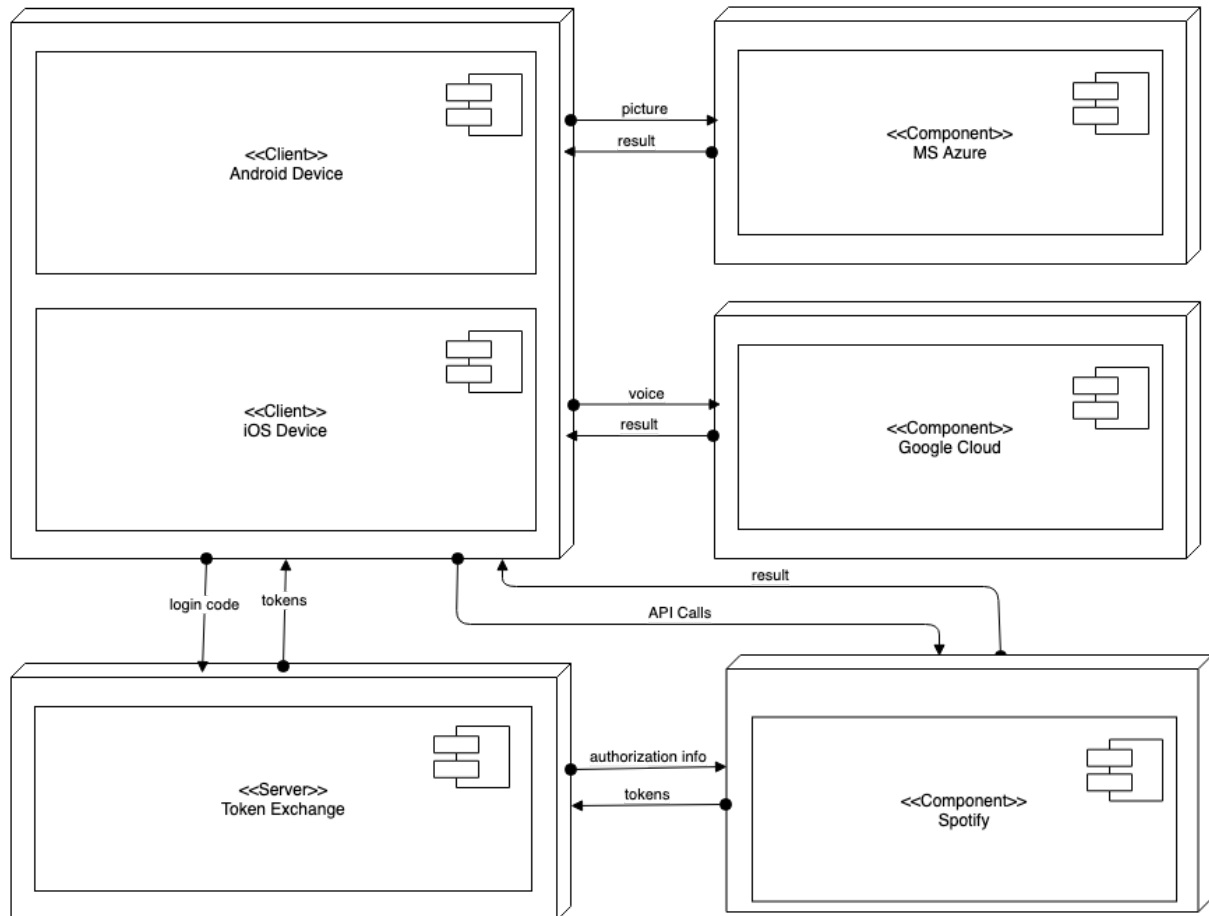


Figure 1: Subsystem Decomposition Diagram

In the authorization process, user logs in and authorize access to Moodio. This generates a unique code for the user. The server combines this info with application's secret key and produces tokens to be used in API calls made by client to Spotify. Then, it returns these tokens alongside with other tokens for Google and Microsoft APIs.

Client side makes API calls using images and voices to retrieve emotion of the user. Emotions are detected from facial analysis of the images and textual analysis of the recordings. The API calls return confidence levels for each emotion and client side converts these confidence levels to valence and arousal values.

<i>mood</i>	<i>valence</i>	<i>arousal</i>	<i>mood</i>	<i>valence</i>	<i>arousal</i>
<i>sleepy</i>	0.01	-1.00	<i>bored</i>	-0.35	-0.78
<i>tired</i>	-0.01	-1.00	<i>annoyed</i>	-0.44	0.76
<i>afraid</i>	-0.12	0.79	<i>enraged</i>	-0.18	0.83
<i>angry</i>	-0.40	0.79	<i>excited</i>	0.70	0.71
<i>calm</i>	0.78	-0.68	<i>melancholy</i>	-0.05	-0.65
<i>relaxed</i>	0.71	-0.65	<i>satisfied</i>	0.77	-0.63
<i>content</i>	0.81	-0.55	<i>distressed</i>	-0.71	0.55
<i>depressed</i>	-0.81	-0.48	<i>uncomfortable</i>	-0.68	-0.37
<i>discontent</i>	-0.68	-0.32	<i>worried</i>	-0.07	-0.32
<i>determined</i>	0.73	0.26	<i>amused</i>	0.55	0.19
<i>happy</i>	0.89	0.17	<i>apathetic</i>	-0.20	-0.12
<i>anxious</i>	-0.72	-0.80	<i>peaceful</i>	0.55	-0.80
<i>good</i>	0.90	-0.08	<i>contemplative</i>	0.58	-0.60
<i>pensive</i>	0.03	-0.60	<i>embarrassed</i>	-0.31	-0.60
<i>impressed</i>	0.39	-0.06	<i>sad</i>	-0.81	-0.40
<i>frustrated</i>	-0.60	0.40	<i>hopeful</i>	0.61	-0.30
<i>disappointed</i>	-0.80	-0.03	<i>pleased</i>	0.89	-0.10

Figure 2:: Valence Arousal Model [1]

Client side selects five artists from user top artists according to the pdf where n is the ranking of the artist.

$$P(n) = 0.5^n, n = 1, 2, 3, \dots$$

Then, client side generates a playlist using Spotify API with these information and creates the playlist in user's account.

4. Impact of Engineering Solutions

4.1. Global Impact

Currently, there are not many applications that recommends people new songs other than the music applications, in our case Spotify, especially regarding the mood of the user. There are some built-in playlists on Spotify that regards a specific mood, but none of them considers the music taste of the user. Moreover, Moodio determines the mood of the user at real time by their image and voice. These features of Moodio makes it impactful for people all around the people, which means every people in the world can use it for getting music according to their music taste, and of course their mood.

4.2. Social Impact

Detecting people's mood through the picture and audio makes user take a selfie and record their speech which makes it entertaining to use. Detecting the mood from these can make the application more interesting rather than just user saying it. Furthermore, we do not share this information from the user with anyone and do not keep them at all, which makes it more secure for users. With Moodio, users can compare their playlists generated by using their mood and music taste. It makes people discover new music and increase social interaction. Moodio offers users a more fun and entertaining way of discovering new music.

5. Contemporary Issues

5.1. Face Recognition

Face recognition, detecting the emotion and mood from the face is a really actual topic currently. In our application, we detect the emotion from the face of the user. We used Microsoft Azure Face API for emotion detection through face and it is a quite recent technology. Moreover, face recognition is also really a contemporary issue that we have worked with.

5.2. Natural Language Processing

Natural Language Processing (NLP) is used for analyzing human (natural) languages and it became really popular recently and is a contemporary issue. We used a specific aspect of NLP, which is sentiment analysis from text, which gives an emotion value between -1 and 1 according to the sentiment of the text. For detecting the sentiment of the text, we used Natural Language API of Google Cloud and detected the sentiment value of user's converted speech to text.

5.3. Cloud Platforms

We used Google Cloud and Microsoft Azure as cloud platform services in our project. Cloud platforms are getting very popular and they are really contemporary technologies. As a result, we had some issues while using them but in the end, we managed to get emotion detection through face, speech-to-text and sentiment analysis through these

technologies and their API's. Therefore, Google Cloud and Microsoft Azure are really helpful for these kind of applications.

5.4. React Native

React Native is a relatively new technology and as a result of this, it is really difficult to find a solution to encountered problems and to find resources and documents about React Native. Moreover, the synchronization between the platforms is also another issue with React Native. It is still in development process not complete yet.

6. Use of Libraries and Resources

"react": "^16.8.6" : It creates user interfaces by using Javascript. We used it for creating React Native apps.

"react-art": "^16.8.6" : It is a JavaScript library for drawing vector graphics using React. We used it for creating React Native apps.

"react-dom": "^16.8.6" : It serves as the entry point to the DOM and server renderers for React. We used it for creating React Native apps.

"react-native": "^0.59.6" : It lets you build mobile apps using JavaScript. It uses the same design as React, letting you compose a rich mobile UI from declarative components. We used it for creating React Native apps for both iOS and Android.

"react-native-audio": "^4.3.0" : It records audio in iOS or Android React Native apps. We used it to record user's speech which will be sent to Google Cloud Speech API for analyzing.

"react-native-camera": "^1.13.1" : It takes photo in iOS or Android React Native apps. We used it to take a selfie which will be sent to Microsoft Azure API for analyzing.

"react-native-cookies": "^3.3.0" : It is a cookie manager library for react native. We used it to login and logout users' profile.

"react-native-elements": "^1.1.0" : It serves better user interface components for better user experience. We used it for better user experience with the application's interface.

"react-native-fs": "^2.13.3" : It serves native filesystem access for react-native applications to read or write files on local storage of mobile phones.

"react-native-gesture-handler": "^1.1.0" : It provides native-driven gesture management APIs for building best possible touch-based experiences in React Native. This is a necessary library to use other libraries.

"react-native-progress": "^3.6.0" : It provides progress indicators and spinners for React Native using ReactART. We used it to show progress icon during delay of response time for better user experience.

"react-native-sound": "^0.10.12" : It is a React Native module for playing sound clips on iOS, Android, and Windows. We used it to play recorded speech of users.

"react-native-vector-icons": "^6.4.2" : It is icon library to use png icons in the application. We used it for icons such as home, camera, and microphone in bottom bar of the application.

"react-native-web": "^0.11.2" : It is a web browser inside the application. We used it to open Spotify login page.

"react-navigation": "^3.9.1" : It provides navigation between the screens of the application. We used it to navigate from one screen to other.

"rn-fetch-blob": "^0.10.15" : It provides the fetch process to send binary large objects in POST requests. We used it to send binary image data to Microsoft Azure API.

"rn-spotify-sdk": "^1.2.7" : It provides user interface and functions to access Spotify API in React Native applications. We used it to get the user specific data from Spotify API.

7. New Tools and Technologies Used

7.1. React Native

Since we wanted to develop Android and iOS applications, we used React Native to do that, to develop a cross-platform application. React Native is developed by Facebook. By React Native, you can develop applications with the help of web of web technologies and it is based on JavaScript. React Native creates native elements through your code. However, React Native is quite new technology and it is open source and still being developed and improved. Therefore, it is difficult to find documentation and sources for it that makes it harder for developers who are new to React Native.

7.2. Google Cloud

Since our application gets audio of the user and detects the mood from it, we used Google Cloud API's for that function. Google Cloud is a cloud platform service developed by Google. We first get the audio and convert it to a text by speech-to-text. Then, we detect the emotion from that text with sentiment analysis. For converting speech to text, we used Cloud

Speech-to-Text API. Then, for detecting the mood of the user from that text, we used sentiment analysis from Google Cloud's Natural Language API. The both API's are under AI&Machine Language Products in Google Cloud.

7.3. Microsoft Azure

Moodio also needs to do emotion analysis from the picture of the user. For that purpose, we use Face API from Microsoft Azure. Microsoft Azure is a cloud platform service developed by Microsoft. Through Face API, we used emotion analysis that gives eight different probability values for eight different emotion types. Then, we get a valence and arousal value by using these values to give it to Spotify API.

7.4. Spotify API

Since our application is highly dependent on Spotify, we use Spotify for Developers for different number of functions of our application Moodio. First, we need it for logging in the user to Spotify with authorization and permissions so that every user is logged in to the system by their Spotify accounts. More importantly, we need Spotify API for creating playlists for user according to the mood and the music taste of the user. We give the emotion value of the user to Spotify API, and then it creates a playlist according to those emotion and the music taste. Then, we save that playlist to that user's account.

7.5. Node JS / NPM

Node.js is an open-source, cross-platform JavaScript run-time environment that executes JavaScript code server-side. npm is a package manager for the JavaScript programming language. It is the default package manager for the JavaScript runtime environment Node.js. It consists of a command line client, also called npm, and an online database of public and paid-for private packages, called the npm registry. The registry is accessed via the client, and the available packages can be browsed and searched via the npm website

7.6. Git / GitHub

Git is a version control system for tracking changes in computer files and coordinating work on those files among multiple people. It is primarily used for source code management in software development. GitHub is a web-based hosting service for version control using git. It

is mostly used for computer code. It offers all of the distributed version control and source code management (SCM) functionality of Git

8. User Manual

8.1 Login Screen

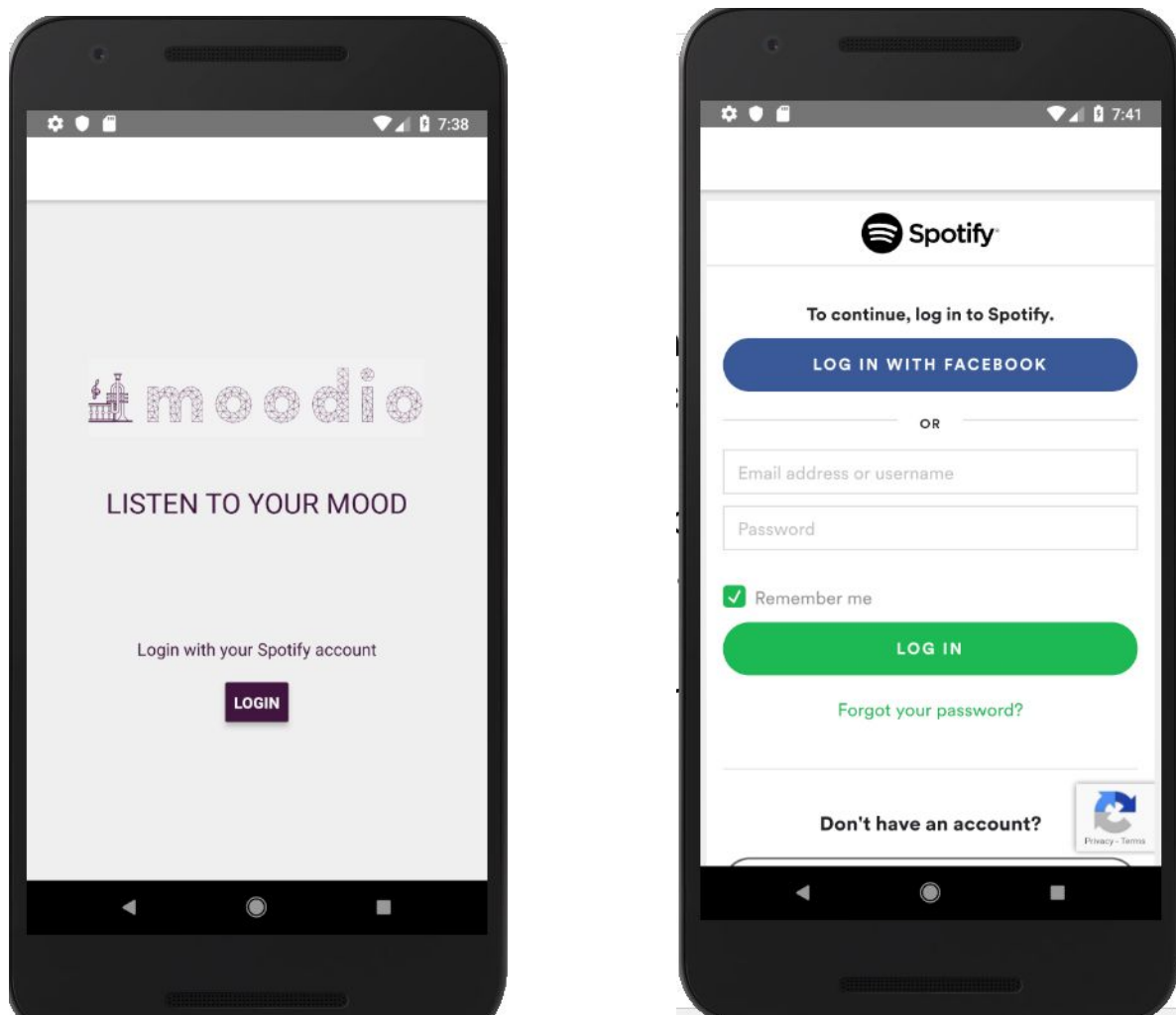


Figure 3: Login Page

In this page, user is asked to login to the system via his/her Spotify account or Facebook account and ask permission to allow to access Spotify account of user

8.2 Homepage

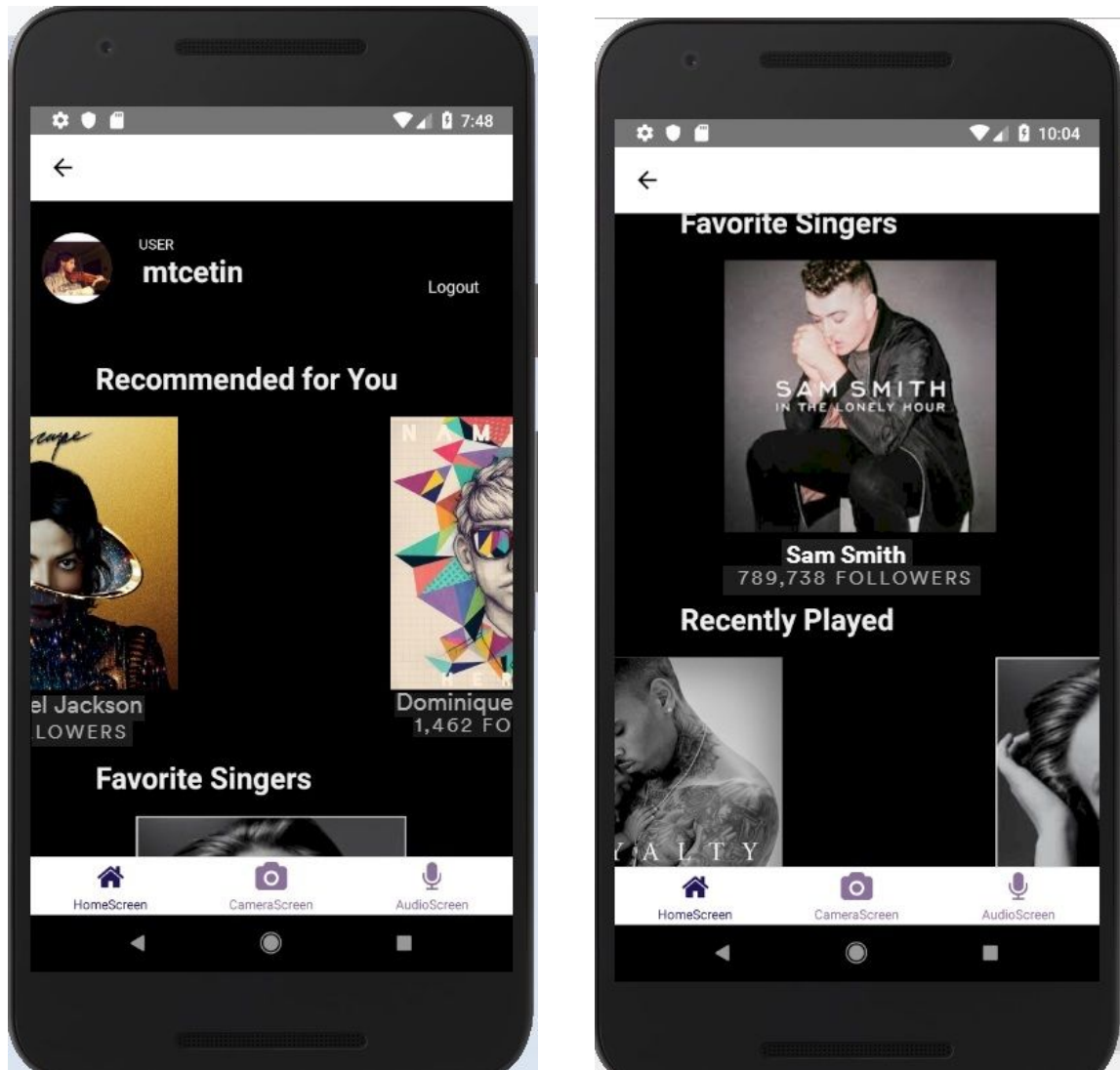


Figure 4: Homepage

In this page, user can see the recommended playlist that is based on the mood and the singers who is preferred by user most frequently. Also user can make a selection for determining the mood either using camera screen or audio screen.

8.3 Camera Screen

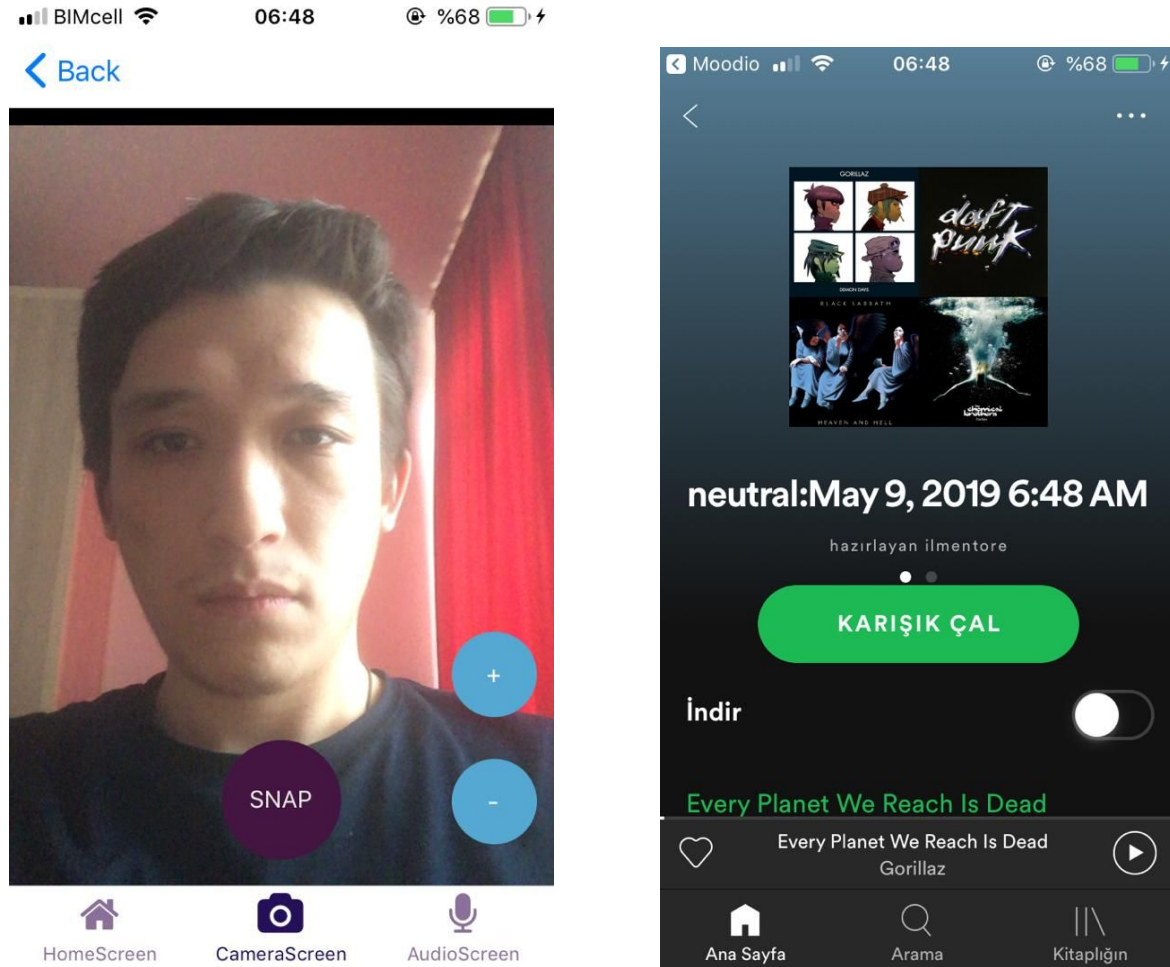


Figure 5: Camera Screen

In this screen, user can take a picture of their own face in order to detect the current mood. After this process, Moodio finds the correct mood of user based on the picture and create a playlist according to the mood of him/her in the Spotify account of user automatically.

8.4 Audio Screen

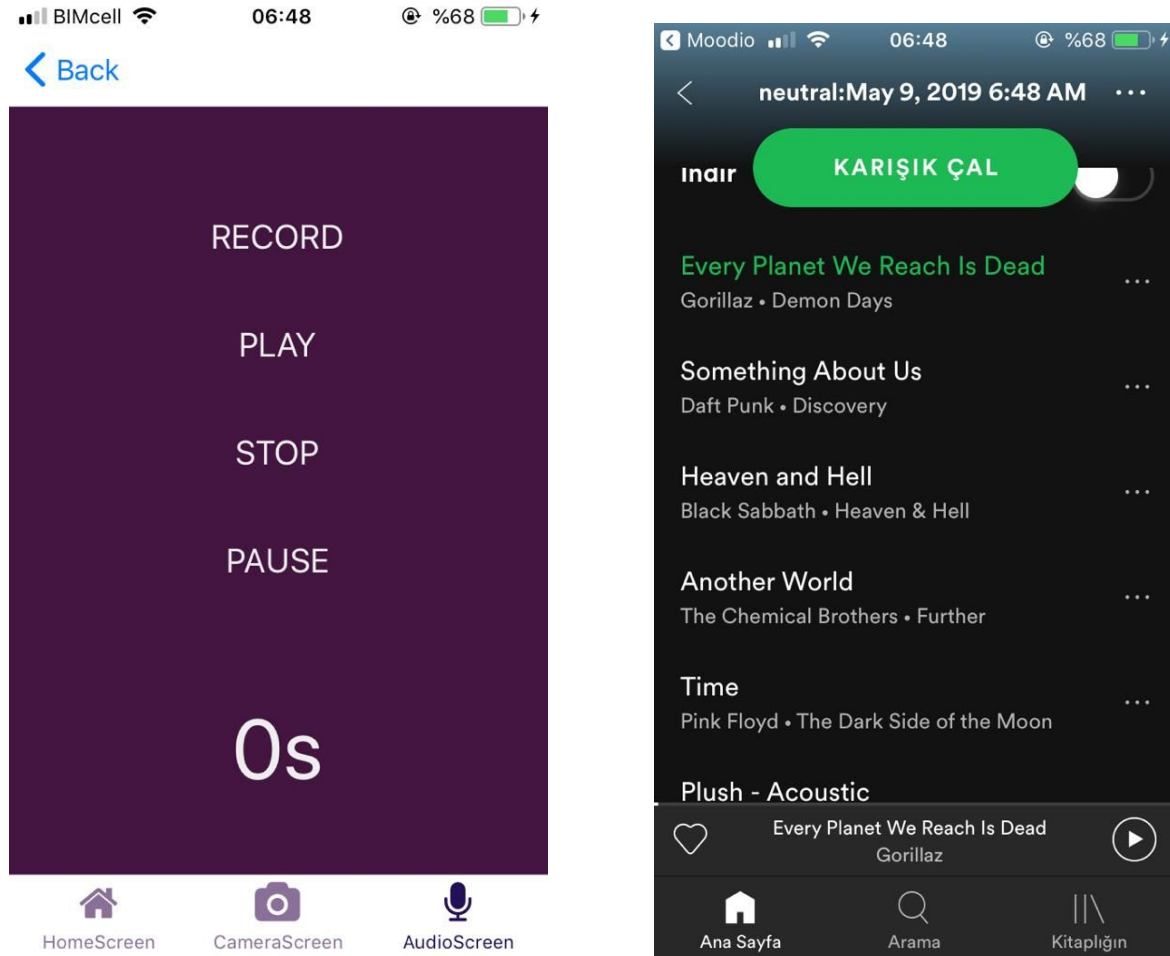


Figure 6: Audio Screen

In this screen, user can take a record their own voice in order to detect the current mood. After this process, Moodio finds the correct mood of user based on the audio and create a playlist according to the mood of him/her in the Spotify account of user automatically.

9. Bibliography

- [1] G. Paltoglou and M. Thelwall, "Seeing Stars of Valence and Arousal in Blog Posts," *IEEE Transactions on Affective Computing*, vol. 4, no. 1, pp. 116–123, 2013.
- [2] <https://facebook.github.io/react-native/>
- [3] <https://github.com/>
- [4] <https://github.com/facebook/react/tree/master/packages/react-dom>
- [5] <https://github.com/jsierles/react-native-audio>
- [6] <https://github.com/react-native-community/react-native-camera>
- [7] <https://github.com/joeferraro/react-native-cookies>
- [8] <https://github.com/react-native-training/react-native-elements>
- [9] <https://github.com/itinance/react-native-fs>
- [10] <https://github.com/kmagiera/react-native-gesture-handler>
- [11] <https://github.com/oblador/react-native-progress>
- [12] <https://github.com/zmxv/react-native-sound>
- [13] <https://github.com/oblador/react-native-vector-icons>
- [14] <https://github.com/necolas/react-native-web>
- [15] <https://facebook.github.io/react-native/docs/navigation>
- [16] <https://github.com/joltup/rn-fetch-blob>
- [17] <https://developer.spotify.com>
- [18] <https://azure.microsoft.com/tr-tr/>
- [19] <https://cloud.google.com/vision/>
- [20] <https://nodejs.org/en/docs/>
- [21] <https://cloud.google.com/natural-language>