Senior Design Project
Moodio
Project Analysis Report

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

Supervisor: Cevdet Aykanat
Jury Members: H. Altay Güvenir, Hamdi Dibeklioğlu
Innovation Expert: Duygu Gözde Tümer

Nov 12, 2018
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

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 show that facial expressions play an important role in the 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 an indicator of feelings, allowing a man to express an emotional state. People can immediately recognize an emotional state of a person. Therefore, the facial expressions and voice tone can be considered as the best way to determine the mood of people. With the help of facial expressions and voice tone, Moodio will suggest the best music playlist according to the mood of users.

This report will contain information about the Moodio, its constraints, professional and ethical issues, functional and nonfunctional requirements.

2. Overview

Moodio is basically an application that provides people the music list according to their mood. The main goal of this application is that it helps people while they are searching for music. It will also offer several features such as finding most searched moods.

What makes Moodio different from the other music application is that Moodio will help users in the stage of music playlist selection. Although the other music applications suggest the music playlist according to the music preferences of users, Moodio will offer the music playlist according to the mood of users determined from their voice and face.

Moodio will be an application that includes 2 main features. The first feature will be the determination of mood from the voice and face using the microphone and the camera of the phone. The second feature of the Moodio will search the music using the mood of people. In order to handle these processes, we will use image processing and machine learning in this project.

In the initial stage of the project, image analysis will be used in order to recognize the mood of users from their face. This will be done based on OpenCV face detection library and Kaggle facial expression of emotion dataset. Furthermore, Warsaw Set of Emotional Facial Expression Pictures (WSEFEP), the Amsterdam Dynamic Facial Expression Set (ADFES), and the Radboud Faces Database (RaFD) will also be used in order to obtain more accurate results. Another thing that we will handle in the initial stage of the project is that voice recognition will be used to recognize the mood of users. This will be done based on the Ryerson Audio-Visual Database of Emotional Speech and Song.

The next stage of the project, machine learning algorithms will be used in order to train the category of music playlists. Also, Spotify API will be used for the classification of songs as a dataset. The data that trained by us will be kept in the database in MySQL.

3. Requirements

3.1. Functional Requirements

- The user must signup and login by using her/his Spotify Application.
- The user can search emotion categories to choose a specific category.
- The user can display the most chosen emotion categories.
- The user can receive a recommended music playlist based on her/his music taste on Spotify and emotions by selecting an emotion category.
- The user can receive a recommended music playlist based on her/his music taste on Spotify and emotions by taking a selfie through the application or talking to the application in order to detect the emotion category.
- The user can listen to the recommended playlist created by Moodio through a Spotify redirection from the application.
- The user can like or dislike a recommended playlist in order to train ai better via the feedback system.
- The user can still share, listen, search music through Spotify not via Moodio.

3.2. Non-functional Requirements

3.2.1. User-friendliness

The user interface must be simple and straightforward yet elegant. It will not be complex since Moodio targets people from any age group and background. Our ultimate goal is Moodio to become an app that even people who do not have had much interaction with smartphones should be able to use it.

3.2.2. Efficiency

Moodio aims to use phone CPU and battery as little as possible to allow users to listen to music for a longer period of time. The response time of Moodio cannot be much to increase user-friendliness and interaction, so the algorithms working behind must be efficient and well-thought.

3.2.3. Scalability

Although Moodio will begin its life with a small number of users, it is expected to reach a large audience. Therefore, Moodio must be scalable to handle lots of users.

3.2.4. Security

Images and audios of users must be kept securely.

3.2.5. Reliability

Moodio must have a very low downtime and must not crash easily because Moodio will be developed to become a part of user’s daily life.

3.2.6. Extensibility

At first, an Android application will be developed for Moodio but it should be easy to develop iOS and Web applications. Platform-specific tools should not be used much.

3.3.Pseudo Requirements

- The application will be mobile app and will have a server. The application will be available for Android and iOS.
- We will have a client-server architecture.
- We will use Git for version controlling and tracking.
- We will build the mobile application using React Native.
- Machine learning part will be done through Python using machine learning libraries such as Scikit-learn. However, we can also use Matlab for machine learning and definitely for image processing.

4.System Models

4.1 Scenarios

➢ Use Case #1

4.1.1 Use Case: Search categories based on emotions

Primary Actor: User

Stakeholders and Interests:

- The user wants to search a category about emotions.
- The system shows categories which include searching key.

Pre-conditions:

- The user must be in the search view.

Post-conditions:

• User found the category what he/she searched for.

Entry-conditions:

• The user must enter a key to search in the input box and click the search button.

Exit conditions:

• User clicks the back button.

• User clicks a category and displays songs.

Success Scenario Event Flow:

1. The user enters the search key and clicks the search button.

2. The user chooses and clicks a category among categories filtered by key in order to display songs.

Alternative Event Flows:

1. User clicks the back button and returns to the main menu.

➢ Use Case #2

4.1.2 Use Case: Login

Primary Actor: User

Stakeholders and Interests:

• The user wants to login in order to listen to music based on emotions.

• Spotify API approves for authorization.

Pre-conditions:

• The user must open the application from his/her phone.

Post-conditions:

• The user logins and use the application according to his/her interest.

Entry-conditions:

• The user must enter his/her Spotify username and password so Moodio can use Spotify APIs.

Exit conditions:

• The user terminates Moodio by clicking back button.

• Login authorization is successful.

Success Scenario Event Flow:

1. The user enters username and password and clicks the login button.
2. Spotify API rejects login and the user
3. Spotify API approves credentials and authorizes Moodio.
4. The user successfully logins.

Alternative Event Flows:

1. The user terminates Moodio application.

➢ Use Case #3

4.1.3 Use Case: Display the most preferred categories

Primary Actor: User

Stakeholders and Interests:

• The user wants to see which emotion category other users preferred most.

Pre-conditions:

• The user must be in the main menu.

Post-conditions:

• The user displays categories and chooses one of them.

Entry-conditions:

• The user must click display categories button.

Exit conditions:

- The user chooses a category to listen to songs.
- The user clicks the back button.

Success Scenario Event Flow:

1. The user clicks the display categories button to display the most preferred emotion categories.
2. The user chooses one of the categories in order to listen in this category.

Alternative Event Flows:

1. The user clicks the back button and returns to the main menu.

➢ Use Case #4

4.1.4 Use Case: Display recommended list based on facial expression

Primary Actor: User

Stakeholders and Interests:

- The user wants to discover the music list based on her/his mood using the picture of her/his face

Pre-conditions:

- The user should be login
- The user must be in the search with facial expression view

Post-conditions:

- The user displays music list and chooses the recommended music list based on the mood that determines with the facial expressions of user

Entry-conditions:

- The user must take a picture of his/her own face.

Exit conditions:

- The user chooses a recommended music list to listen

• The user clicks the back button.

Success Scenario Event Flow:

1. The user clicks the discover music list button in order to discover the music list according to his/her current mood using camera

2. The user take a picture of his/her own face in order to discover recommended music list to his/her current mood

Alternative Event Flows:

1. The user clicks the back button and returns to the main menu.

➢ Use Case #5

4.1.5 Use Case: Display recommended list based on selected category

Primary Actor: User

Stakeholders and Interests:

• The user wants to discover the music list based on her/his mood either through picture of his/her own face using phone camera or voice using the internal microphone of the phone

Pre-conditions:

• The user should be login

• The user must be in the main menu

Post-conditions:

• The user selects either discover music list through facial expression view or sound expression view.

Entry-conditions:

• The user must select the display recommended list button

Exit conditions:

• The user chooses either facial expression or sound expression button

• The user clicks the back button.

Success Scenario Event Flow:

1. The user clicks the discover music list button in order to select the searching method that he/she want to discover music list

Alternative Event Flows:

1. The user clicks the back button and returns to the main menu.

➢ Use Case #6

4.1.6 Use Case: Display recommended list based on sound expression

Primary Actor: User

Stakeholders and Interests:

• The user wants to discover the music list based on her/his mood using the voice record

Pre-conditions:

• The user should be login

• The user must be in the search with sound expression view

Post-conditions:

• The user displays music list and chooses the recommended music list based on the mood that determines using the voice of user

Entry-conditions:

• The user must record his/her voice using the inner microphone of phone

Exit conditions:

• The user chooses a recommended music list to listen

• The user clicks the back button.

Success Scenario Event Flow:

1. The user clicks the discover music list button in order to discover the music list according to his/her current mood using voice record

2. The user records his/her own voice to discover recommended music list to his/her current mood

Alternative Event Flows:

1. The user clicks the back button and returns to the main menu.

➢ Use Case #7

4.1.7 Use Case: Listen recommended music list

Primary Actor: User

Stakeholders and Interests:

• The user wants to listen the music list based on her/his mood

Pre-conditions:

• The user should be login

• The user must be in the discover music list view

• The user must be determine his/her mood either using camera or voice record

Post-conditions:

• The user displays music list and chooses the recommended music list based on the mood to listen

Entry-conditions:

• The user must click the play button of the recommended music list

Exit conditions:

• The user terminates the recommended music list

• The user clicks the back button.

Success Scenario Event Flow:

1. The user records his/her voice or takes picture of his/her own face in order to determine the current mood

2. The user clicks the recommended music list button to listen the songs

3. Spotify API allows to play the recommended music list

4. The user successfully play the recommended list.

Alternative Event Flows:

1.The user clicks the back button and returns to the discover music list view

➢ Use Case #8

4.1.8 Use Case: Like & Dislike Recommended Music List

Primary Actor: User

Stakeholders and Interests:

• The user evaluates the recommended music list according to his/her interests.

Pre-conditions:

• The user must be listen to the recommended music list

Post-conditions:

• The user listens the music list and evaluates the recommended music list.

Entry-conditions:

• The user must click evaluate button of the recommended music list

Exit conditions:

• The user evaluates the recommended music list

Success Scenario Event Flow:

1. The user listens the recommended music list

2. The user clicks the like & dislike button in order to evaluate the recommended music list

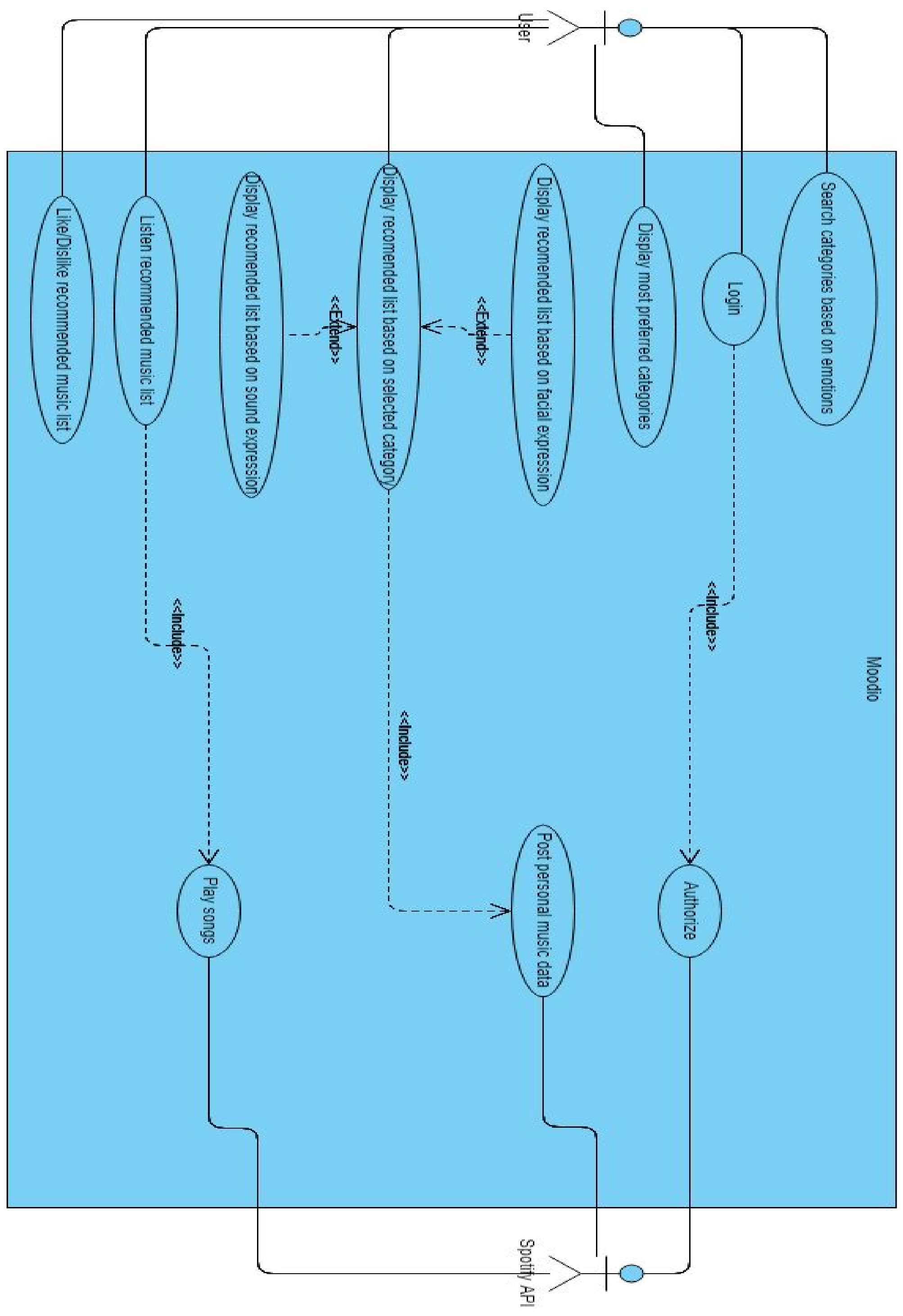
Alternative Event Flows:

1.The user clicks the back button before he/she finish listening to the recommended list

2.The user evaluates the music list before leaving the recommended list.

4.2. Use Case Model

4.2.1 Use Case Diagram


User
Moodio
Search categories based on emotions
Login
Display most preferred categories
Display recomended list based on facial expression
<<Extend>>
Display recommended list based on selected category
<<Extend>>
Display recomended list based on sound expression
Listen recommended music list
Like/Dislike recommended music list
<<Include>>
<<Include>>
<<Include>>
Authorize
Post personal music data
Play songs
Spotify API

4.2.2 Use Case Descriptions:

➢**Login:** Login to the app

➢**Search Categories:** Search the music categories based on emotions

➢**Display the most preferred:** Display the most preferred music list by the users

➢**Display recommended music list(facialexpression):** Display the recommended music list based on the facial expression the user

➢**Display the recommended list based on category:** Display the music list based on category using the one of the searching methods which are facial expression and voice record

➢**Display recommended music list(voice record):** Display the recommended music list based on the voice record of user

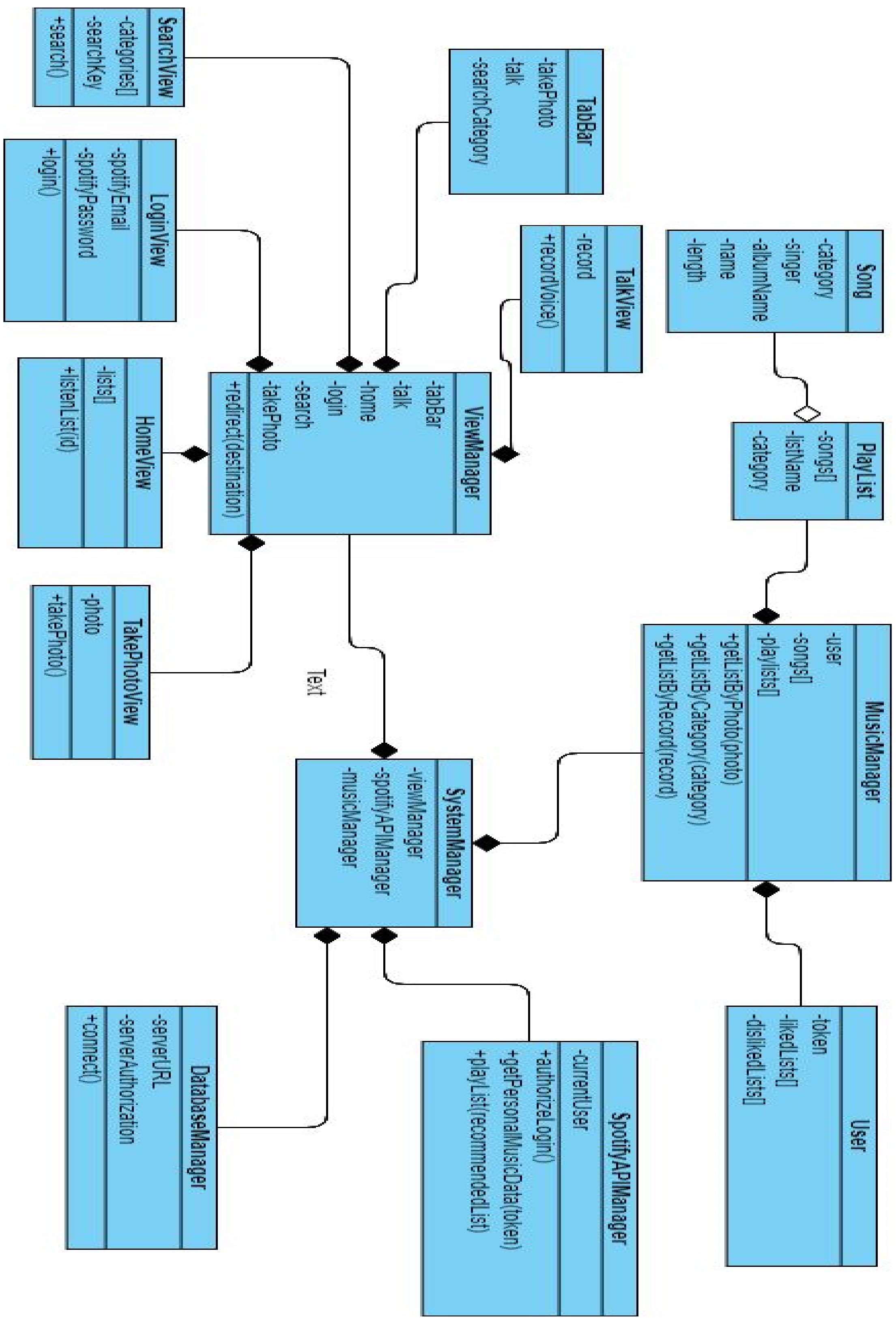
➢**Listen to the recommended music list:** Listen to music list after discovering the musics according to the current mood of user

➢**Play songs:** Play the songs of recommended music list.

➢**Authorize:** Spotify API authorize the account using the username and password for the Moodio

➢**Post personal music data:** Spotify API post the each users music data

4.3. Object and Class Model



4.3.1 Explanation of Class Diagram

➢**SystemManager:** System manager act as wrapper class which organizes and arranges the all execution of the Moodio. This class responsible for the initiating the necessary objects and making the procedural call that is needed by the app. Also System Manager keeps track of the screens,musics and current user session through by communicating View Manager,Music Manager and Spotify API manager.

➢**ViewManager:** View Manager Class basically responsible for the displaying screens of app.It keeps track of currently displayed screen and updates the display according to the instructions. This class includes objects of each screen that is used in Mooido. The basic role of this class is that View Manager connects the each screen of app to each other.

➢**SpotifyAPIManager:** Spotify API Manager basically responsible for the user session and authorization of the Mooido to allow the user login to app. This class keeps track of login information of the each user.

➢**User:** This class keeps liked/disliked music list of each users. It’s not responsible to the keeps track of the login information of users such as email and password.

➢**Music Manager:** Music Manager class keeps track of each users recommended playlist based on the moods. Also this class communicates with the System Manager in order to get the voice records and face picture of the users. Using all these data, this class keeps track of music list based on the given mood which is determined using picture or voice record by the users.

➢**TalkView:** Talk View class keeps track of the voice that recorded by user

➢**TakePhotoView:** Take Photo View Class basically keeps track of the picture that is taken by the user

➢**SearchView:** This class keeps track of the categories that is determined by the spotify and responsible for the making search according to the given key by the user.

➢**HomeView:** This class keeps track of the list determined by the either facial expressions or sound expressions of the user. Also provides user to play the recommended music list

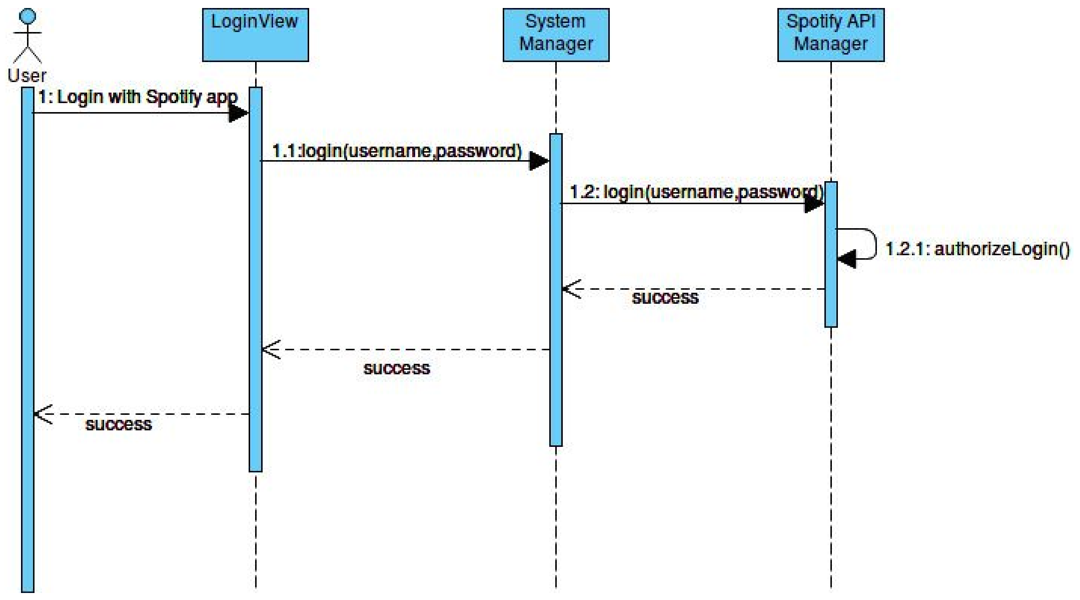
➢**Database Manager:** This class manages the sql queries and creates database connections.

➢**Song:** This class keeps track of category, singer, name, length and album name of each song

4.4. Dynamic Models

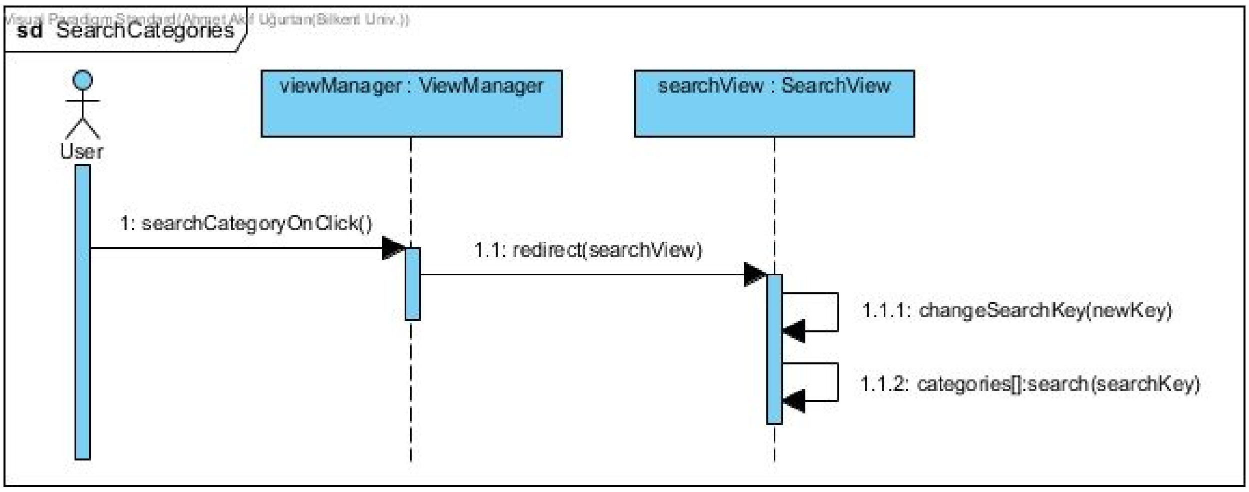
4.4.1. Sequence Model

4.4.1.1. Login Diagram



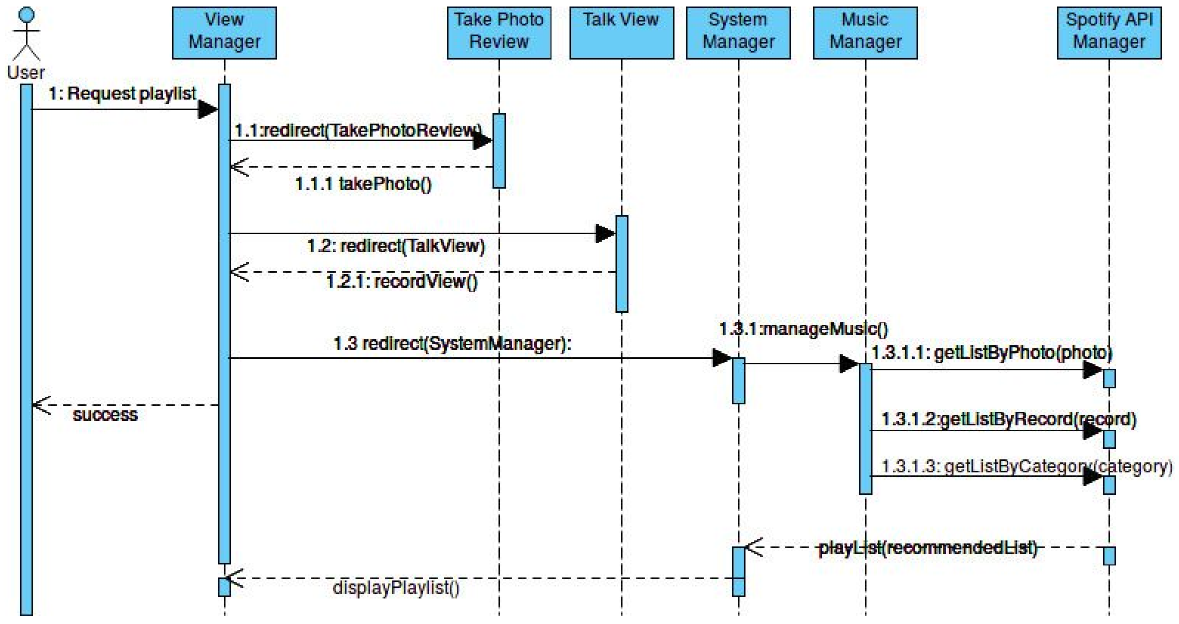
The user opens the app and sees the login screen. Then, the user logs in to the application through his/her Spotify account by entering username and password. Then, Spotify API authorizes login and returns success and failure. If success, it redirects to the main page, if fail, it redirects to the login screen.

4.4.1.2. Search categories based on emotions


Visual Paradigm Standard(Ahmet Akif Uğurtan(Bilkent Univ.))
sd SearchCategories
User
viewManager : ViewManager
searchView : SearchView
1: searchCategoryOnClick()
1.1: redirect(searchView)
1.1.1: changeSearchKey(newKey)
1.1.2: categories[]:search(searchKey)

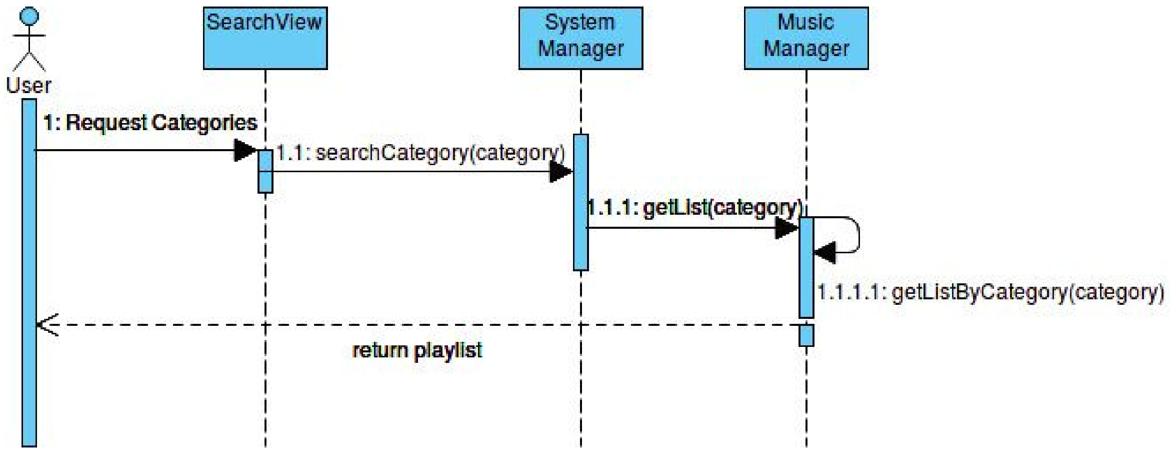
User clicks the search icon on the upper tab bar. The system sends the requests to the ViewManager. ViewManager processes the requests and redirect the view to the searchView. The user enters the key he/she wanted to search and the system updates searchKey. The user clicks the search button and displays the list of categories filtered by searchKey.

4.4.1.3. Get Playlist Recommendation Diagram



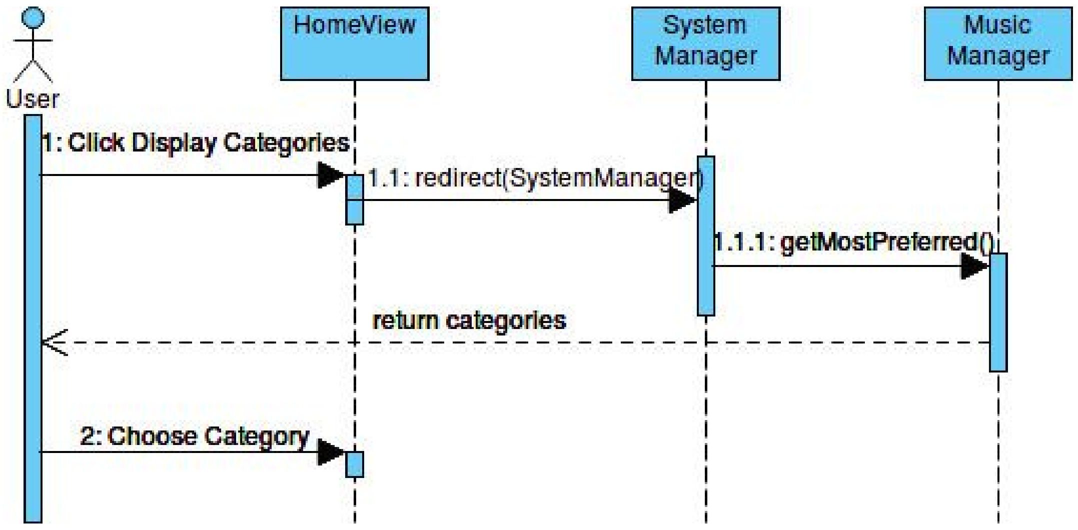
In this sequence, the user first is redirected to take photo view to take his/her photo and then redirects to Talk View and record voice. Then, this information goes to the Spotify API and these information is used with the previous data from the user and a playlist is created. Then, this playlist is displayed to the user.

4.4.1.4. Search Category Diagram



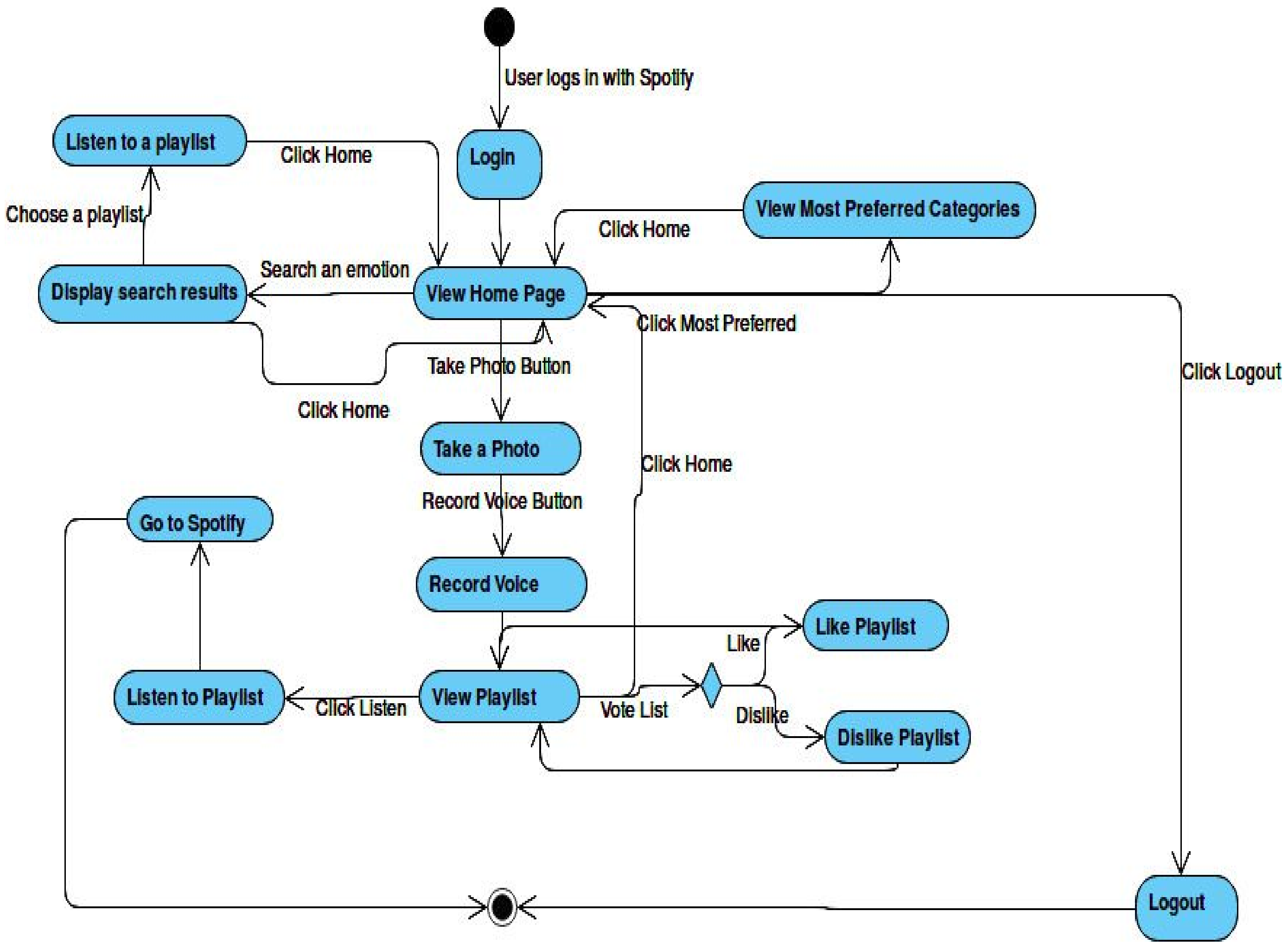
In this diagram, user clicks on search bar and searches for a category. The system manager sends this specific category based on emotions. Then music manager returns a list based on that category.

4.4.1.5. Display Most Preferred Categories Diagram


User
HomeView
System Manager
Music Manager
1: Click Display Categories
1.1: redirect(SystemManager)
1.1.1: getMostPreferred()
return categories
2: Choose Category

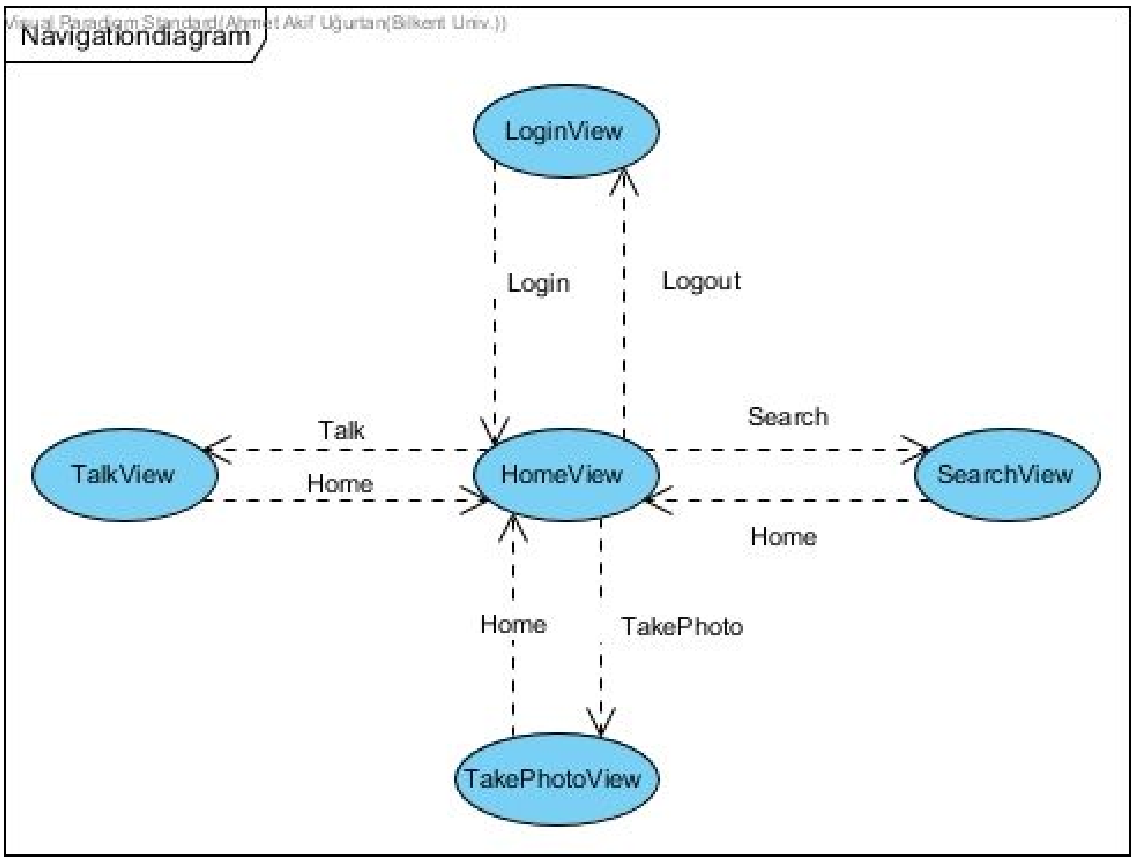
In this diagram, user clicks “Display Most Preferred Categories” button in main menu. Then, the system redirects to the music manager to get the most preferred categories, then returns that playlist to the user. Then, the user chooses a category to view the playlists on that category.

4.4.2. Activity Model



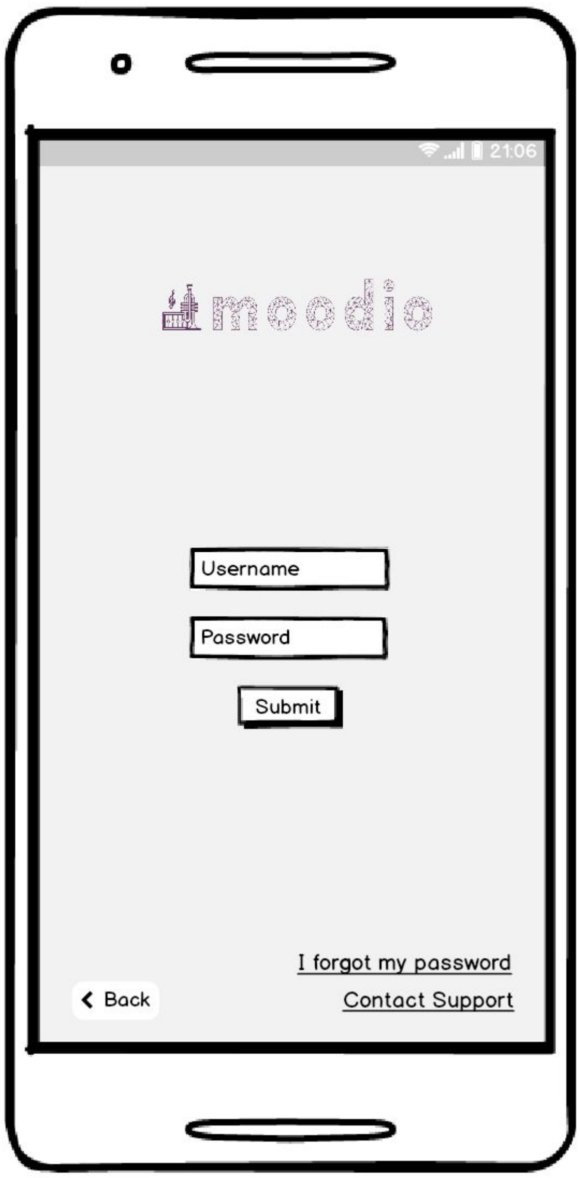
5. User Interface

5.1. Navigational Path



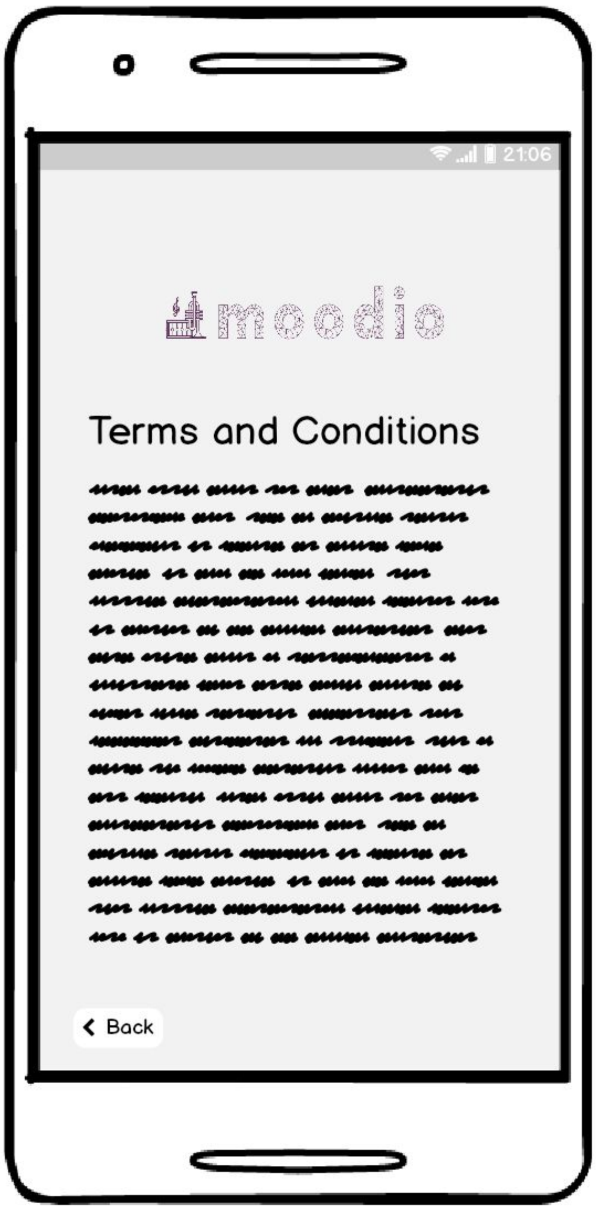
5.2. Mockups

5.2.1. Login page



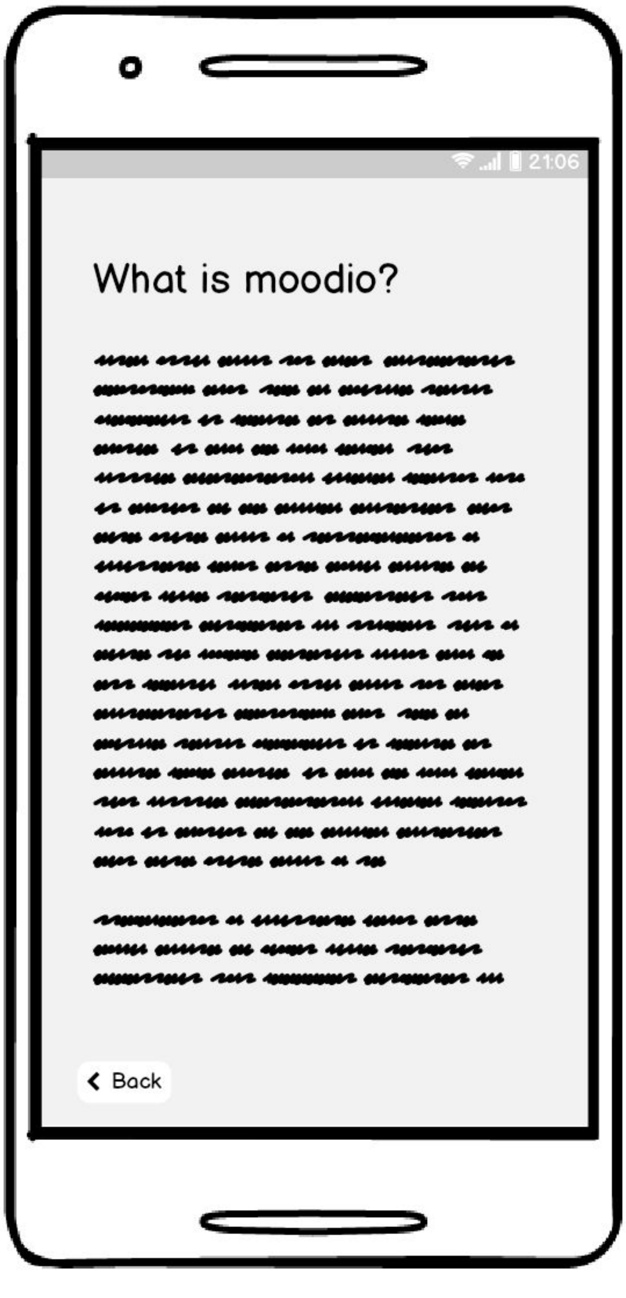
If user already has an account, s/he can sign in using username and password. If user has forgotten her/his password, s/he can claim her/his account providing some information. S/he also can contact to developer team in case of a problem.

5.2.2. Terms and Conditions



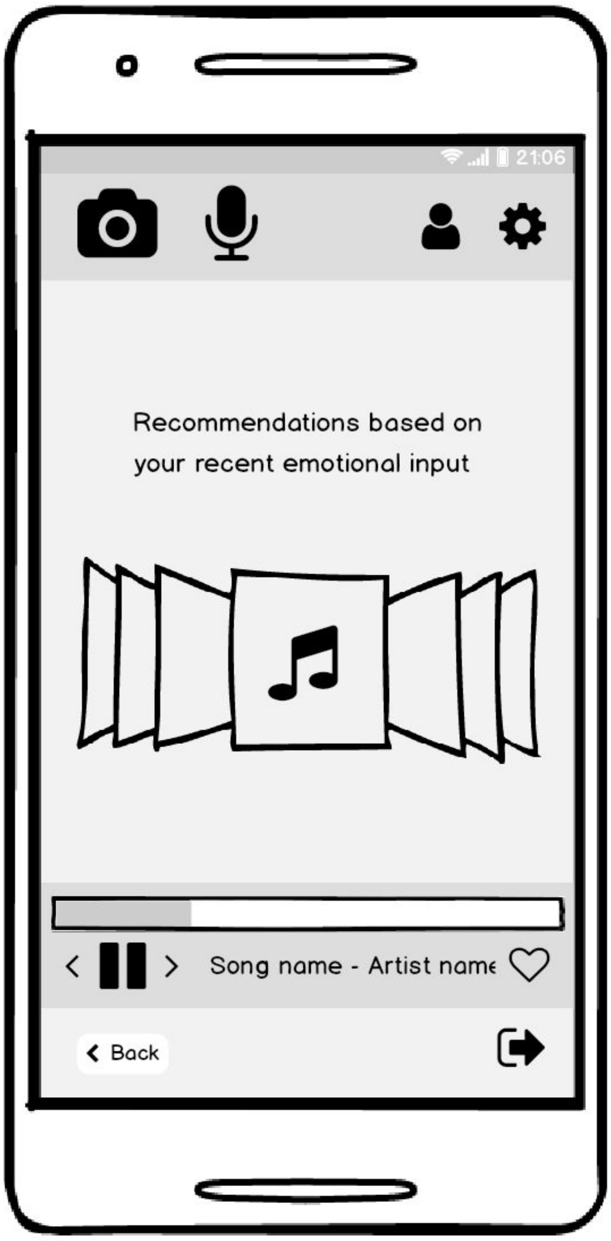
Moodio heavily uses and relies on user data, therefore it is essential to inform users and lighten users of Moodio’s policies regarding security.

5.2.3. Help page



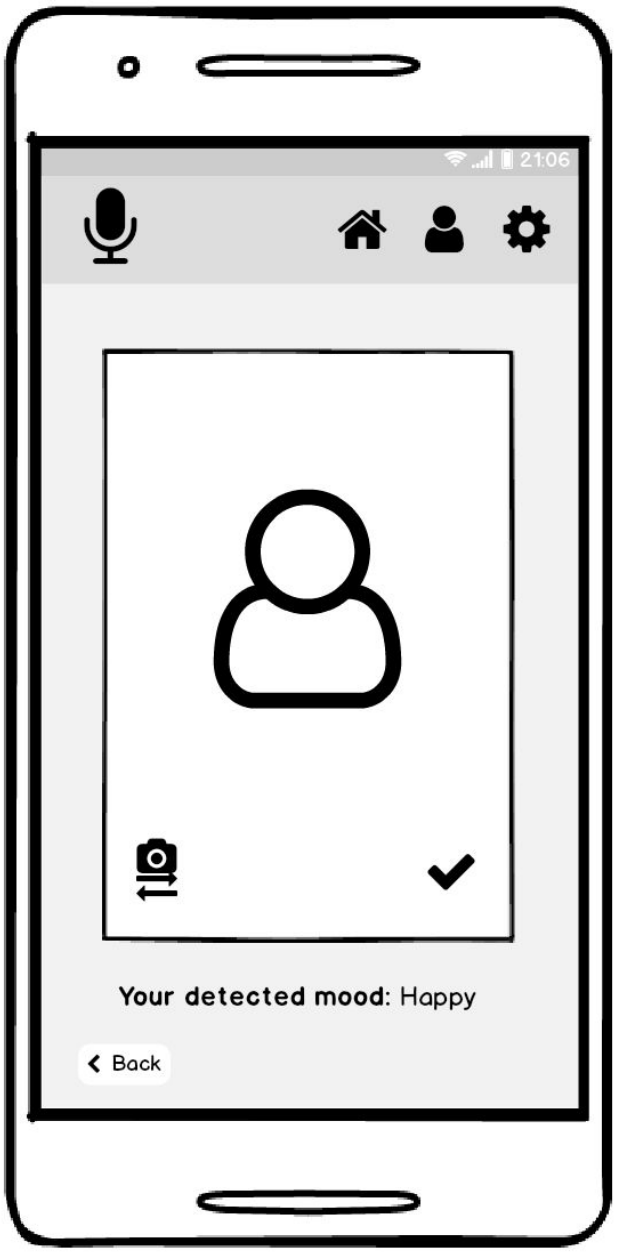
Help page can be accessed through starting page where users choose to sign in or sign up and settings.

5.2.4. Homepage



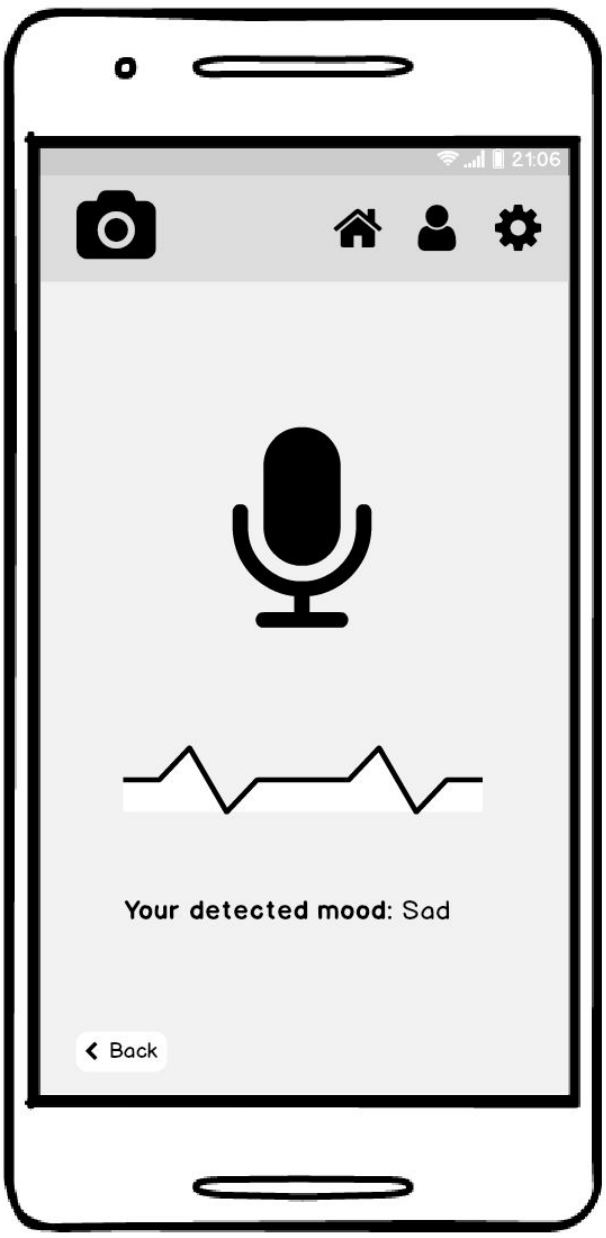
After user signs in, s/he is welcomed to the service with recommendations based on her/his recent emotional input. S/he can choose from and play recommended songs using this page. If s/he does not like one, s/he can skip it. Otherwise, s/he is expected to mark it using the heart button. Using camera and microphone buttons on the top left of the page, s/he enters new input. On the top right of the page, there exist buttons to reach profile and settings pages. They are simple pages with only showing existing information or links to change it.

5.2.5. Mood detection using visual



User can take a picture of her/him and Moodio will analyze it to recommend songs based on this analysis combined with previously gathered data. Currently detected mood will be shared with user. User can go back to homepage using the button on the top.

5.2.6. Mood detection using audio



User can record her/his voice and Moodio will analyze it to recommend songs based on this analysis combined with previously gathered data. Currently detected mood will be shared with user. User can go back to homepage using the button on the top.

6. Reference

[1] Spotify API:

Developer.spotify.com. (2018). *Features | Spotify for Developers*. [online] Available at: https://developer.spotify.com/discover/ [Accessed 14 Oct. 2018].

[2] Emotion recognition from speech and song:

Livingstone, S., & Russo, F. (2018). *The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS)*. *Zenodo*. Retrieved 14 October 2018, from https://zenodo.org/record/1188976#.W73QUitzCUl

[3] The dataset for emotion recognition from image:

Facial Expression of Emotion. (2018). *Kaggle.com*. Retrieved 14 October 2018, from https://www.kaggle.com/sivlemx/facial-expression-of-emotion

[4] Emotion recognition from facial expression

Tarnowski, P., Kołodziej, M., Majkowski, A., & Rak, R. J. (2017). Emotion recognition using facial expressions. *Procedia Computer Science, 108*, 1175-1184. doi:10.1016/j.procs.2017.05.025