<MoodBlend>

Software Design Specification

<22/05/2022>

Aydın Duygu-150118981

Buse Apaydın-150119694

Elif Beril Yılmaz-150119748

Hamza Kavak-150118886

Zahit Erdem Güzel-150119906

Prepared for
CSE3044 Software Engineering Term Project

Table of Contents

1. Introduction

The metropolitan life that people live in today is a stressful life, but it causes changes in emotions such as excitement, happiness, etc. This change in the state of emotions is reflected in the facial shapes and facial expressions of people. Nowadays, with the development of artificial intelligence technology, it has become even easier to analyze the state of emotions. Mood Blend application makes music suggestions according to people's emotion status analysis and recommends the song they need to the user. This project analyzes the emotion status according to the expression on people's faces and recommends the appropriate song to the user.

1.1. Purpose

Personalized playlists have been in our lives for a long time, but these lists are mostly blended lists that reflect our overall likes. We have designed the MoodBlend application for a system that suggests songs based on the mood of people and sorts these songs correctly. The main purpose of Mood Blend application is a mobile application that suggests the appropriate song according to people's mood. Thanks to this application, people can find the song that best suits their mood. Thanks to the song that Mood Blend offers to the user, a person can better understand the mood and feel the feeling that they are better and more comfortable.

1.2. Statement of scope

Mood blend application is a mobile application that suggests the appropriate song according to the user's mood. This software application analyzes the emotional state of people's facial expressions and determines their emotional state. It recommends the most appropriate song to the user according to the specified emotion state. When the users first enters the application, they encounters the splash screen. The application asks for permission to take a photo of the users. If the photo taken by the user is not suitable, the application will send an alert to the user to take the photo again. If the photo taken is appropriate, emotion status analysis is performed from the user's facial expression. The system makes song suggestions according to the perceived emotion state according to the determined facial expression. As a result, thanks to Mood Blend, the user will feel better and understand the mood better, finding the most suitable song for their emotional state.

1.3. Software context

The task of the project is to analyze the facial expressions of people, determine their feelings and suggest the most appropriate song to the user. The Mood Blend application allows you to conveniently recommend music that accompanies people's emotional states.

Since this application is a mobile application, it needs API level 24 (Android 7.0) or higher for Android. The application will use the HTTP protocol for API requests communication over the internet. No database is required for the project. With the help of Spotify API, the data comes from the Spotify database.
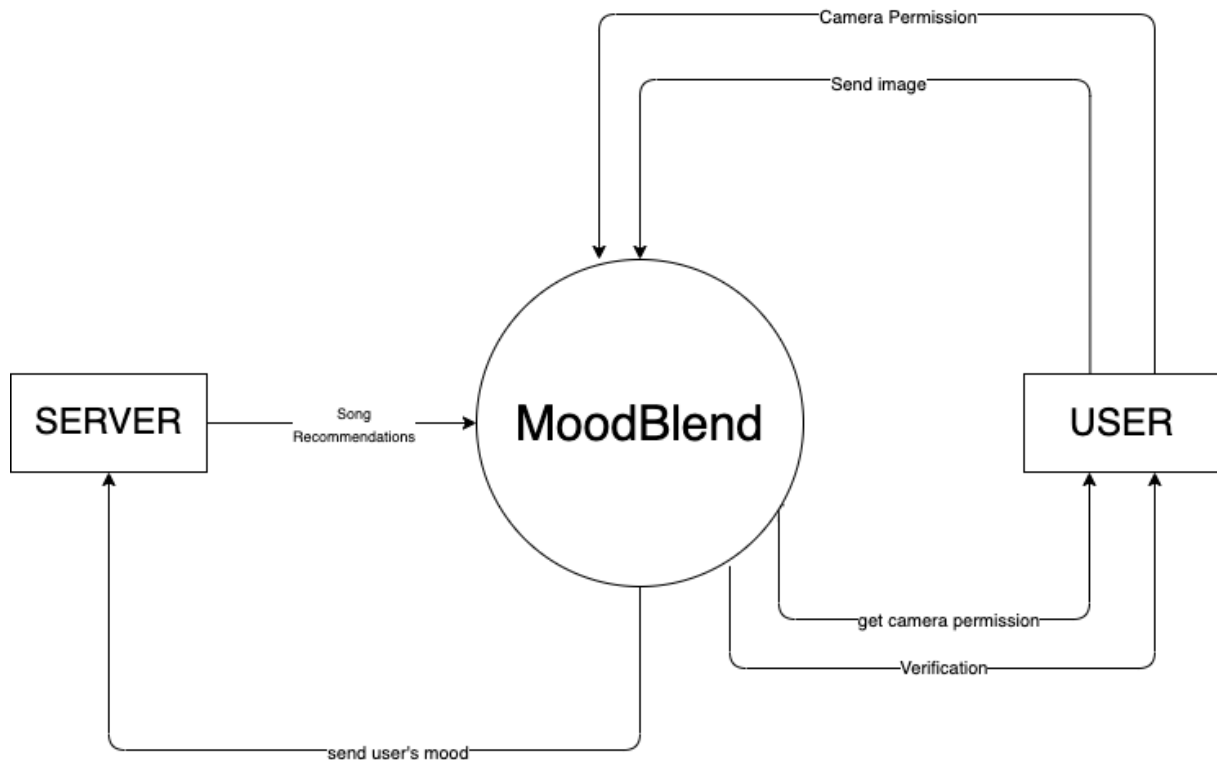

Figure 1. Software Context Diagram

1.4. Major constraints

- An internet connection is required for the system to work.
- A smartphone or tablet is required for the mobile application to work because MoodBlend is a mobile application.

- Having a mobile application working correctly depends on the version of the android that is suitable for Android 7.0 (API level 24) and above for Android.
- Integrated development environment (IDE) is used for the backend. Detection model is created with Pyhton. Codes are written via Java.

1.5. Definitions

- **Convolutional Neural Networks**
  - An algorithm which is a famous machine learning approach in image detection to recognize emotions.

- **Artificial Neural Networks**
  - Usually simply called neural networks (NNs), are computing systems inspired by the biological neural networks that constitute animal brains.

- **Rectifier Linear Units (Relu):**
  - The rectified linear activation function or Relu for short is a piecewise linear function that will output the input directly if it is positive, otherwise, it will output zero.

- **Pycharm**
  - It is an IDE that is an integrated development environment where you can make programs using python language.

- **Processing narrative (PSPEC)**
  - When a user first creates his/her account with SplitPay, a new UserAccount object is created. This object is responsible for storing information unique to the SplitPay user.

1.6. Acronyms and Abbreviations

MOODBLEND	Project Name
GUI	Graphical User Interface
USER	The person who receives a music recommendation
UI	User Interface
ANDROID	Android; It is a free operating system based on Linux being developed for mobile devices by Google and the Open Handset Alliance.
IDE	Integrated Development Environment
UC	Use Case
OS	Operating System
App	Application
PSPEC	Project Specification
HTTP	HTTP is an application-level communication protocol for hype media information systems that are distributed from a source and are open to common use.
API 24	It is a unique value that enables communication and interaction between the application and the operating system with level 24

Table 1. Acronyms and Abbreviations

1.7. References

- https://dergipark.org.tr/en/download/article-file/1798929
- https://www.ijirt.org/master/publishedpaper/IJIRT143701_PAPER.pdf
- https://ijarcce.com/wp-content/uploads/2021/07/IJARCCE.2021.10682.pdf
- https://www.ijream.org/papers/SSJ2019004.pdf
- https://iopscience.iop.org/article/10.1088/1757-899X/912/6/062007/pdf
- https://www.frontiersin.org/articles/10.3389/fpsyg.2021.760060/full
- https://open.spotify.com/

2. Design Consideration

2.1. Design Assumptions and Dependencies

- As a new update arrives to the application, we assume that these updates will be shaped according to the wishes of the users. Otherwise, these should be fixed with updates.
- We assume that all the symbols, images and buttons in the application, in short, the application designs are understandable by the users. Otherwise, they should be fixed with updates.
- If the system sends a warning message to users, they should receive these messages and instructions. We assume that users should use the system in appropriate conditions, conveniently, easily and reliably.
- No database is required for the MoodBlend application. With the help of Spotify API, the data comes from the Spotify database via webserver.
- Operating System does not matter.

2.2. General Constraints

- **Hardware or software environment**

  o This application will work on android phones and tablets. Since this application is a mobile app, it needs Android 7.0 (API Level 24) or higher. These devives must have an internet connection and a working camera to run this app.

- **End-user environment**

  o Moodblend app available to all people with an Android phone.

- **Interface/protocol requirements**

  o The application will use the HTTP protocol for communication over the internet. The server will be connected to the internet via Wifi or 3G.

- **Data repository and distribution requirements**

  - There is no database maintained by the developers in this application.

- **Security requirements (or other such regulations)**

  - The user does not need to provide any special information to use the application. The photo he took remains on the user's device and is not shared with third parties.

- **Network communications**

  - It is absolutely necessary to be able to recommend songs to the user with a server connection.

- **Financial constraints**

  - We need a hosting to make our server to be active all the time. We also need a special domain to access to server before accessing Spotify api. Also we need to have Google Play account to publish our app.

2.3. System Environment

The application is coded with Java language on Android Studio and includes XML design. The main working mechanism and machine learning model of the application were developed using the Python programming language on Pycharm. To use the application, it is necessary to use an android supported device. There is no specific hardware requirement for the system.

2.4. Development Methods

We will use Convolutional Neural Networks algorithm which is a famous machine learning approach in image detection to recognize emotions. CNN algorithm is aclassification algorithm like logistic regression. It is frequently used in image classification, object identification, image and captioning problems. Besides implementing vanilla Artificial Neural Networks, we will use convolution and pooling layers and then We will implement max pooling as pooling algorithm

on specified image dataset. We will use Rectifier Linear Units (Relu) and softmax functions for normalization. Secondly we will develop a mobile application rather than a desktop computer program. We will use Java and Python programming languages while doing these.

3. Architectural and component-level design

3.1. System Structure

Moodblend has two separate component which is firstly user-side mobile app and server-side with spotify api.

User side also has two part which are android user interface written with java and motion detected model created python.

With the model prepared here, the user's face is scanned and the mood is determined. Then, the determined mode is sent to the API prepared on the server side, and there is a song return according to the user's mode. There is no database record that we need to keep in all these transactions. Songs will be pulled from previously prepared playlists on spotify and presented to the user.

3.1.1. Architecture diagram
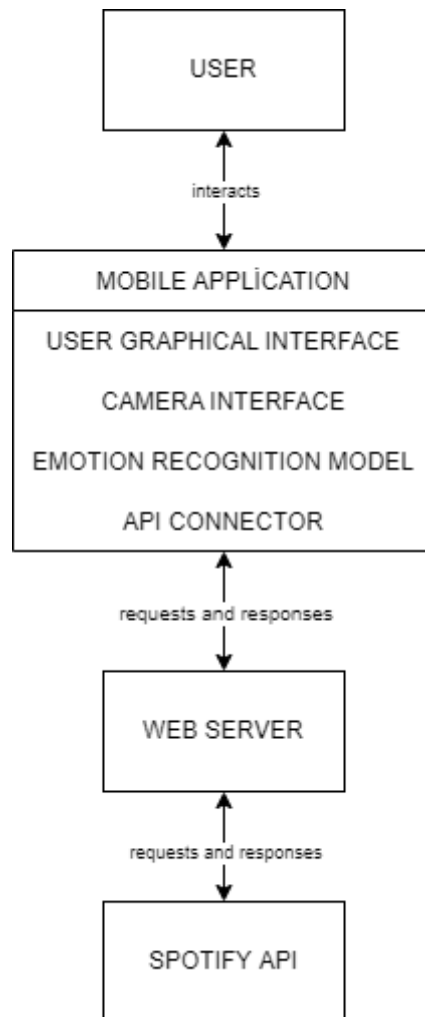

3.2. Description for Component n

3.2.1. Onboarding Component

It is the welcome screen that the user will see once.

3.2.1.1. Processing narrative (PSPEC) for Onboarding Component

This component aims to provide a brief information to the user who will use the application for the first time and gets the necessary permissions to use it in the next process.

3.2.1.2. Onboarding Component interface description.
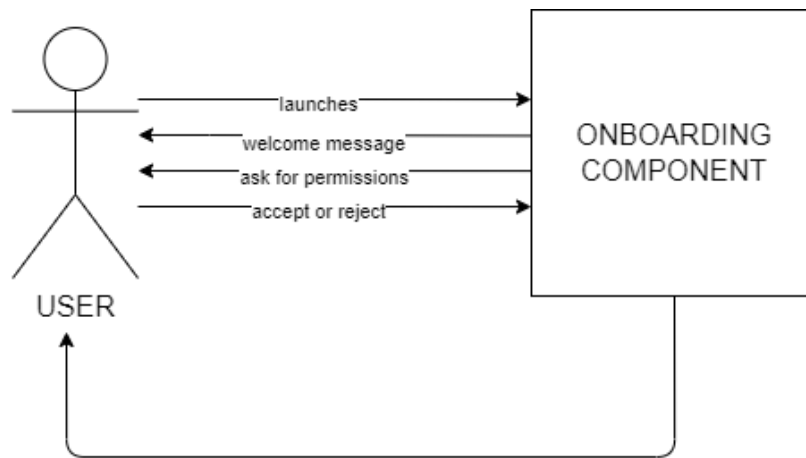
3.2.1.3. Onboarding Component processing detail

The first time the user opens the application after installing it, it will open and will not appear on subsequent launches. It contains a very short introductory text on how to use the application Also here the user will be asked for camera permission. If it does not, the transition to the next main screen cannot be achieved.
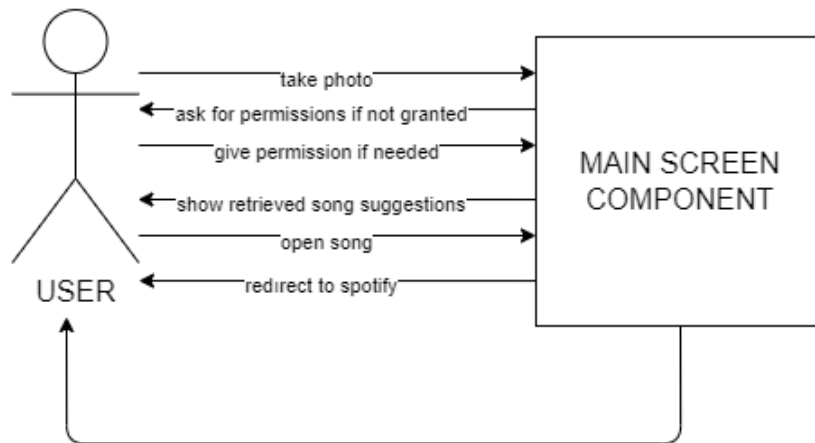
3.2.2. Main Screen Component

It is home page in application.

3.2.2.1. Processing narrative (PSPEC) for Main Screen Component

This component checks if the user is connected to the internet, and checks the camera permission again. When these conditions are met, the camera screen opens properly when the button is pressed.

3.2.2.2. Main Screen Component interface description.

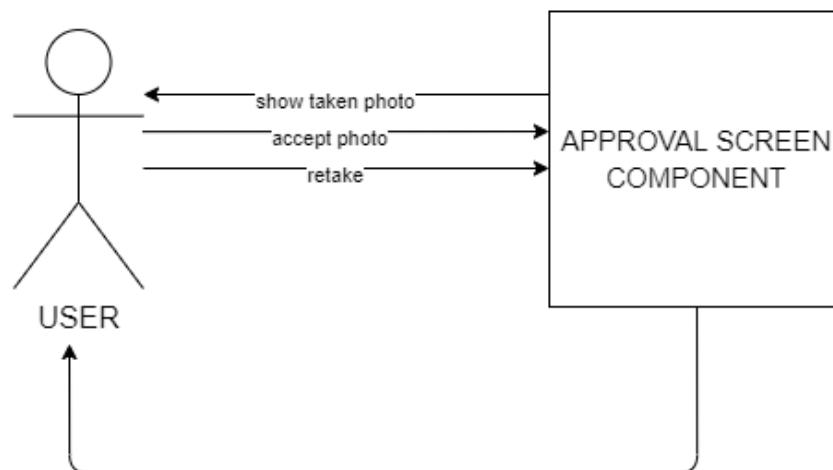### 3.2.2.3. Main Screen Component processing detail

The user comes to this screen from the onboard screen, and will start the operations here. When the button is clicked, it is checked whether the necessary permissions are available, if there are no permissions, a warning message is displayed to the user and permission is requested again.

3.2.3. Approval Screen Component

3.2.3.1. Processing narrative (PSPEC) for Approval Screen Component

This component controls taking pictures and managing them.

3.2.3.2. Approval Screen Component interface description.
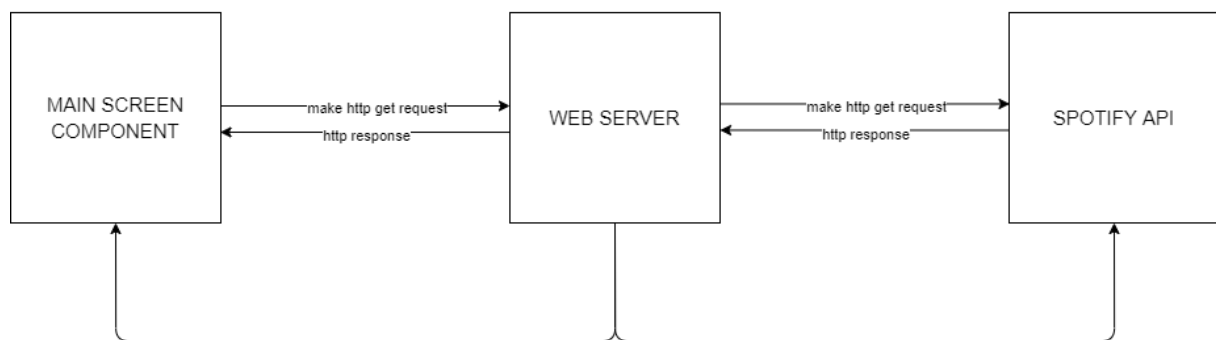

3.2.3.3. Approval Component processing detail

Here, the selfie is taken and confirmed or the reshoot process is repeated. When it is confirmed, the picture is processed and its mode is found. Then the music recommendation process continues automatically.

3.2.4.   Web server Component

3.2.4.1.   Processing narrative (PSPEC) for Web server Component

This component communicates with the server according to the result produced from the model and waits for a response from the server. This component brings together the user and the song suitable for user's mode and allows the user to listen.

3.2.4.2.   Web server Component interface description.

3.2.4.3.   Web server Component processing detail

The user approved the photo he took and after the current mode of the user is determined with the help of our model, we send the result produced from here to the server via API. According to the mode of the user sent to the API, one of the previously determined songs is selected and shown to the user. If the user wants to click on this song, he can listen to it on spotify.

4. Restrictions, limitations, and constraints

The MoodBlend project has some restrictions, limitations, and constraints.

- This project can only be used by users using an android phone.
- Also these devices must use Android 7.0 (API level 24 operating systems).Other systems which have operating systems as Windows Mobile, Blackberry, etc. can not use the app. Our application is not available for iOS users.
- An internet connection is required to run the application. The project will not work without it. In the section recommending songs to the user, song suggestions are made via Spotify. The server will be connected to the internet via Wifi or 3G.
- There is a time limitation for writing project code. We have only 2 months to complete our project.

5. Conclusion

Moodblend is a mobile application for all people. This application is an application that analyzes the emotional state for people who experience intense emotions and recommends them to the user with the music they need.

It is written only for devices using android. The user's mode is determined by the developed python model with the help of Tensorflow, then it makes a suggestion to the user through the server according to the specified mode. In the process of making a suggestion here, the server communicates with spotify and pulls the song directly from the spotify database and directs it there when it wants to rest.