UCS503 SOFTWARE ENGINEERING

PROJECT REPORT

EMOSIC
(Music Playlist Generator on the basis of Facial Expression)

Submitted by:
Aarti Sharma (101683032)
Ajay (101503015)
Aditya Bisht (101553001)
COE-1

Submitted to:
Dr. Sanmeet Bhatia
Assistant Professor
Computer Science and Engineering Department,
Thapar University, Patiala

COMPUTER SCIENCE ENGINEERING DEPARTMENT
THAPAR UNIVERSITY, PATIALA-147004, PUNJAB
INDIA
November 2017

TABLE OF CONTENTS

Abstract

Manual segregation of a playlist and annotation of songs, in accordance with the current emotional state of a user, is labor intensive and time consuming. The software developed by us and the algorithm used in it, aspires to reduce the overall computational time and the cost of the designed system. It also aims at increasing the accuracy of the designed system. The facial expression recognition module of the proposed algorithm is validated by testing the system against FER2013 dataset. The overall accuracy of the emotion recognition algorithm, for user independent dataset is 72%.

Introduction

Music plays an important role in an individual's life. It is an important source of entertainment and is often associated with a therapeutic role. With the advent of technology and contiguous advancements in multimedia, sophisticated music players have been designed and have been enriched with numerous features, including volume modulation, genre classification etc. Although, these features successfully addressed the requirements of an individual, a user sporadically suffered through the need and desire of browsing through his playlist, according to his mood and emotions. Using traditional music players, a user had to manually browse through his playlist and select songs that would soothe his mood and emotional experience. This task was labor intensive and an individual often faced the dilemma of landing at an appropriate list of songs. Emotions are synonymous with the aftermath of an interplay between an individual's cognitive gauging of an event and the corresponding physical response towards it. Among the various ways of expressing emotions, including human speech and gesture, a facial expression is the most natural way of relaying them. A facial expression is a discernible manifestation of the emotive state, cognitive activity, motive, and psychopathology of a person. Homo sapiens have been blessed with an ability to interpret and analyze an individual's emotional state. However, machines were deprived of a complex brain like a human's, which could recognize, distinguish, and perceive different emotions accurately. Ergo, an urge to craft sophisticated intelligent systems, equipped with such skills persisted. The field of Facial Expression recognition (FER) synthesized algorithm that excelled in furnishing such demands. FER enabled the computer systems to monitor an individual's emotional state effectively and react appropriately.

Emosic (Emotion+Music) is a web based software that automatically detects the mood of the user and generates a playlist of songs which is suitable for the current mood. In this software, the image is captured using webcam and that image is passed under different stages to detect the mood or emotion of the user. The web app is developed in such a way that it can analyze the image properties and determine the mood of the user and yield a list of songs from a playlist in conformance with the user's emotional state.Any user can access this hosted platform-independent software from their computers using an Internet browser. The software would be ready for the user to use after an online registration. The scope of this software is not just limited to web based software but we can later develop an android application.

Dataset

The dataset used for training LBPHclassifier is FER2013.

The data consists of 48x48 pixel grayscale images of faces. The faces have been automatically registered so that the face is more or less centered and occupies about the same amount of space in each image. The task is to categorize each face based on the emotion shown in the facial expression in to one of seven categories (0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral). train.csv contains two columns, "emotion" and "pixels". The "emotion" column contains a numeric code ranging from 0 to 6, inclusive, for the emotion that is present in the image. The "pixels" column contains a string surrounded in quotes for each image. The contents of this string a space-separated pixel values in row major order. test.csv contains only the "pixels" column and your task is to predict the emotion column.

The training set consists of 28,709 examples. The public test set consists of 3,589 examples.

This dataset was prepared by Pierre-Luc Carrier and Aaron Courville, as part of an ongoing research project.

Source:

https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/data

General Constraints, Assumptions and Dependencies

The following list presents the constraints, assumptions, dependencies or guild lines that are imposed implementation of Generation of music playlist on the basis of Emotion detection(Emosic).

- It will take 4-6 seconds to give the result.
- It will not be able to generate a playlist of very large amount of songs because it will take a lot of memory to load larger list hence it will decrease the performance of site.
- It requirse a good computation power because it will process the pixel of an image and map them to corresponding emotion.
- A person can't increase size of its favorite songs playlist more than 100.
- A user should have basic knowledge of using internet and browser.
- The webcam should be at least of 2 megapixel and more than this.
- It will require dedicated server for very high computation performance.
- The central database server and backup database servers should be updated regularly as user changes the playlist or add some new songs.

Methodology

In this project, the proposed algorithm revolves around an automated music recommendation system that generates a a customized playlist in accordance with a user's facial expressions. A User's facial expression helps the music recommendation system to decipher the current emotional state of a user and recommend a corresponding subset of the original playlist. It is composed of three main modules: Facial expression recognition module, Audio emotion labeled database and a System integration module.

Facial Expression Recognition
The input image to the system can be captured using a web cam. This image undergoes image enhancement where the **Haarcascade Automatic face finder** (Cascade Classifier). All RGB and gray scale images are converted into 48x48 pixel unlisted arrays stored as .npy files. This preprocessed image is fed into the face detection block. Face detection is carried out using Automatic face finder (Cascade Classifier) algorithm.
The facial image obtained from the face detection stage forms an input to the emotion recognition stage. Training and classification is carried out using **LBPH Face Recognition** algorithm to classify faces among 4 classes of emotions : happy, angry, sad , neutral.

LBPH FaceRecognizer:

It treats data as a vector somewhere in a high-dimensional image space. We all know high-dimensionality is bad, so a lower-dimensional subspace is identified, where (probably) useful information is preserved. **Local binary patterns (LBP)** is a type of visual descriptor used for classification in computer vision. LBP is the particular case of the Texture Spectrum model proposed in 1990.LBP was first described in 1994. It has since been found to be a powerful feature for texture classification; it has further been determined that when LBP is combined with the Histogram of oriented gradients (HOG) descriptor, it improves the detection performance considerably on some datasets.

Code Implemented

i. Face Detection: HaarFilters Automatic face finder (Cascade Classifier)

```
faceDet = cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
faceDet_two = cv2.CascadeClassifier("haarcascade_frontalface_alt2.xml")
faceDet_three = cv2.CascadeClassifier("haarcascade_frontalface_alt.xml")
faceDet_four = cv2.CascadeClassifier("haarcascade_frontalface_alt_tree.xml")


def detect_faces():
   files = glob.glob("pictures\\*") #Get list of all images with emotion
   frame = cv2.imread(files[0]) #Open image
   gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY) #Convert image to grayscale

   face = faceDet.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=10,
                     minSize=(5, 5), flags=cv2.CASCADE_SCALE_IMAGE)
   face_two = faceDet_two.detectMultiScale(gray, scaleFactor=1.1,
            minNeighbors=10, minSize=(5, 5), flags=cv2.CASCADE_SCALE_IMAGE)
   face_three = faceDet_three.detectMultiScale(gray, scaleFactor=1.1,
            minNeighbors=10, minSize=(5, 5), flags=cv2.CASCADE_SCALE_IMAGE)
   face_four = faceDet_four.detectMultiScale(gray, scaleFactor=1.1,
            minNeighbors=10, minSize=(5, 5), flags=cv2.CASCADE_SCALE_IMAGE)

   if len(face) == 1:
      facefeatures = face
   elif len(face_two) == 1:
      facefeatures = face_two
   elif len(face_three) == 1:
      facefeatures = face_three
   elif len(face_four) == 1:
      facefeatures = face_four
   else:
      facefeatures = ""
      print('no face detected')

   for (x, y, w, h) in facefeatures: #get coordinates and size of rectangle containing face
      print ("face found in file")
      gray = gray[y:y+h, x:x+w] #Cut the frame to size
      print(gray)
      filenumber=0
      try:
         out = cv2.resize(gray, (48, 48)) #Resize face so all images have same size
         cv2.imwrite("pictures\\%s.jpg" % filenumber, out) #Write image
      except:
          print('hello') #If error, pass file
```

ii. Emotion Detection: LBPH Face Recognizer

```
fishface = cv2.face.LBPHFaceRecognizer_create()#Initialize fisher face classifier

data = {}

def get_data(): #Define function to get file list, randomly shuffle it and split 80/20
    images=np.load('data_train.npy')
    labels= np.load('labels_train.npy')
    images_test = np.load('data_test.npy')
    images_test = images_test[:300]
    labels_test = np.load('labels_test.npy')
    labels_test = labels_test[:300]
    images=images.reshape([-1, SIZE_FACE, SIZE_FACE, 1])
    images_test = images_test.reshape([-1, SIZE_FACE, SIZE_FACE, 1])
    return images,labels,images_test,labels_test

def run_recognizer():
    training_data, training_labels, prediction_data, prediction_labels = get_data()

    print("training fisher face classifier")
    print("size of training set is:", len(training_labels), "images")
    fishface.train(training_data, np.asarray(training_labels))
    print("predicting classification set: ",len(prediction_labels))
    cnt = 0
    correct = 0
    incorrect = 0
    for image in prediction_data:
        pred, conf = fishface.predict(image)
        print(EMOTIONS[pred])
        if pred == prediction_labels[cnt]:
            correct += 1
            cnt += 1
        else:
            incorrect += 1
            cnt += 1
    return ((100*correct)/(correct + incorrect))

#Now run it
metascore = []
for i in range(0,1):
    correct = run_recognizer()
    print("got", correct, "percent correct!")
    metascore.append(correct)

print("\n\nend score:", np.mean(metascore), "percent correct!")
```

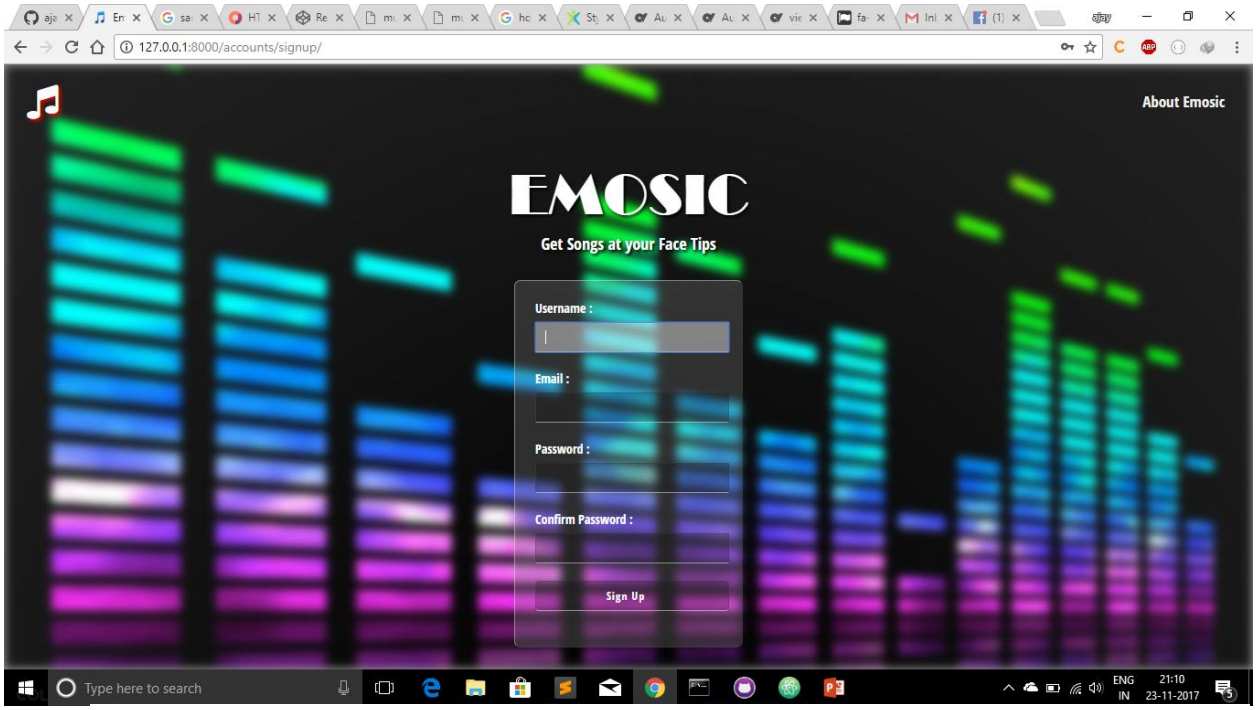
External Interface Requirements

The following list presents the external interface requirements:

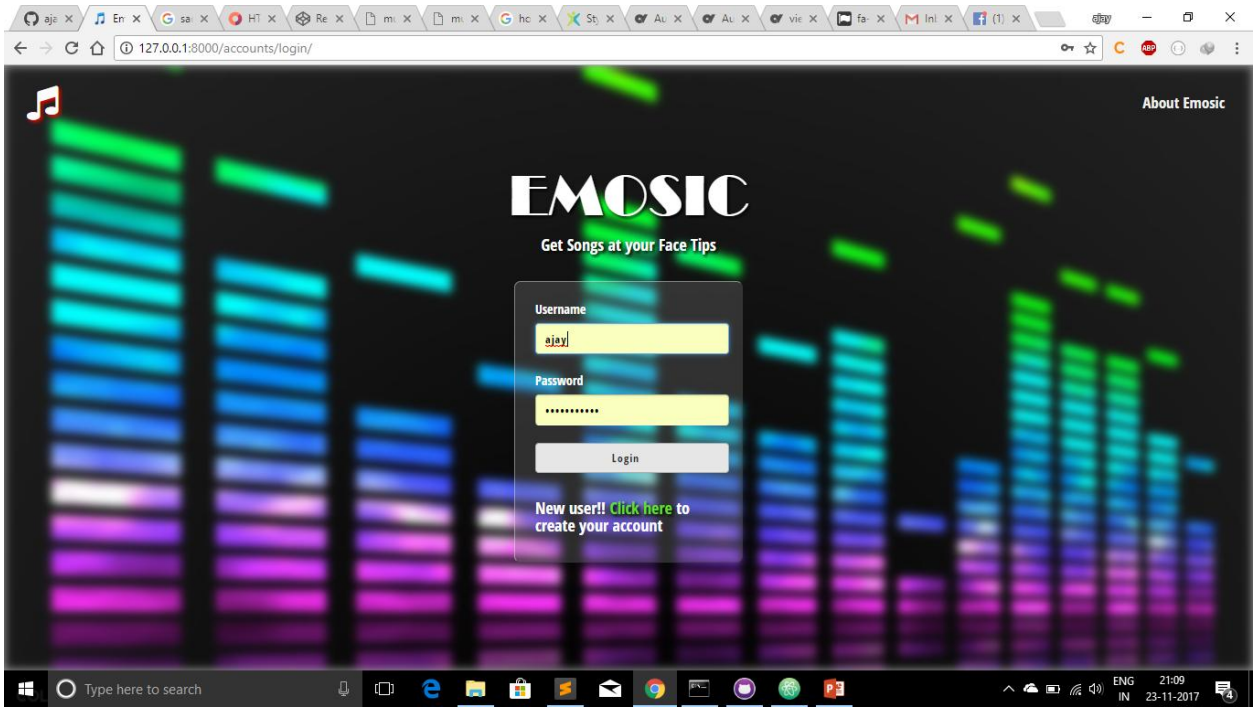
1) The product requires a good quality webcam in case of pc and laptop or a good front facing camera in case of smartphone and tablets. By good quality we mean a camera of at least 2 megapixels.
2) The device on which the product is used should also be having a good quality sound system to listen to the output sound of the music.
3) The hardware and operating system requires a screen resolution not more than 320 x 240 pixels

User Interface and Product Functionalities

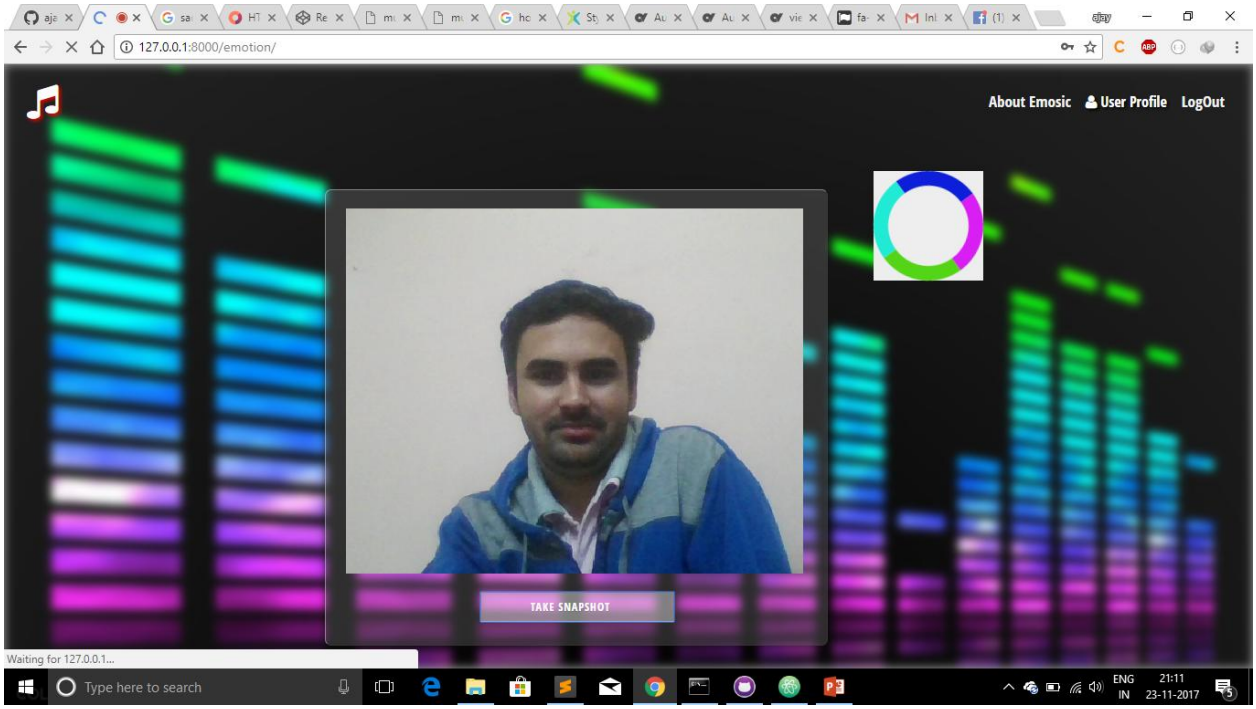
- Sign up/Login
  The user must be registered in order to use the software, unregistered user can't access the music player.



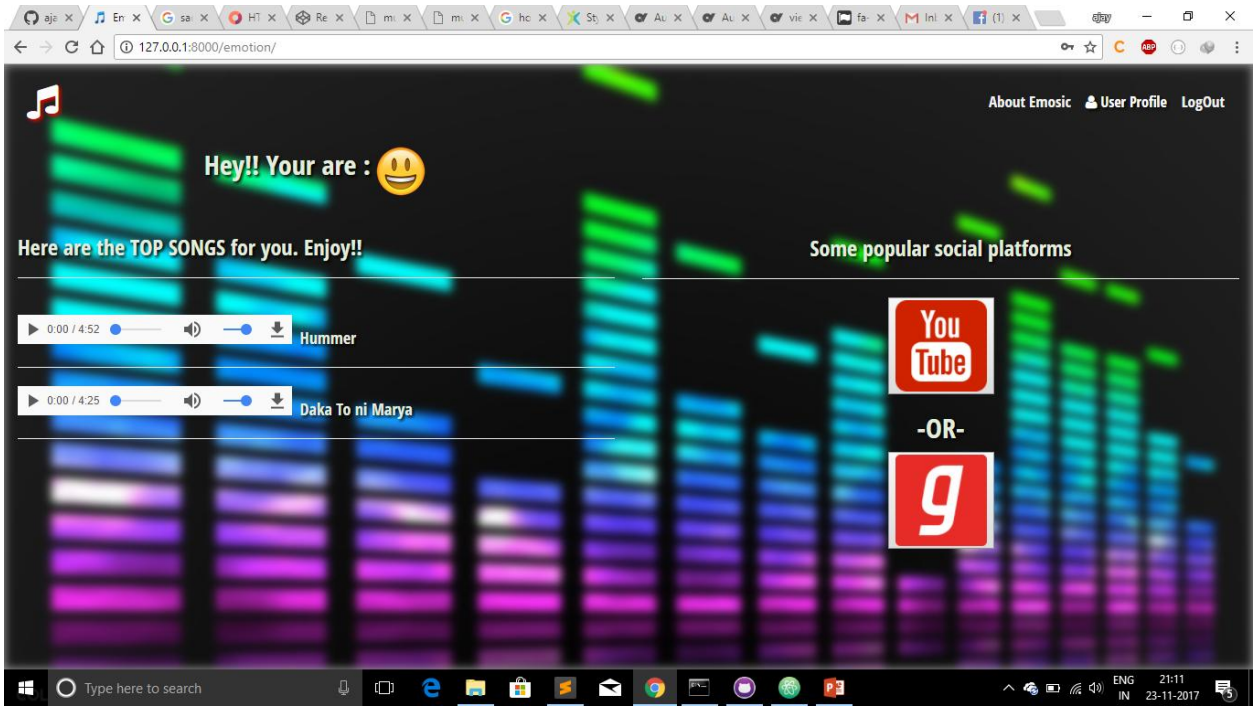
- Login User can log in to the system by entering valid username and password.



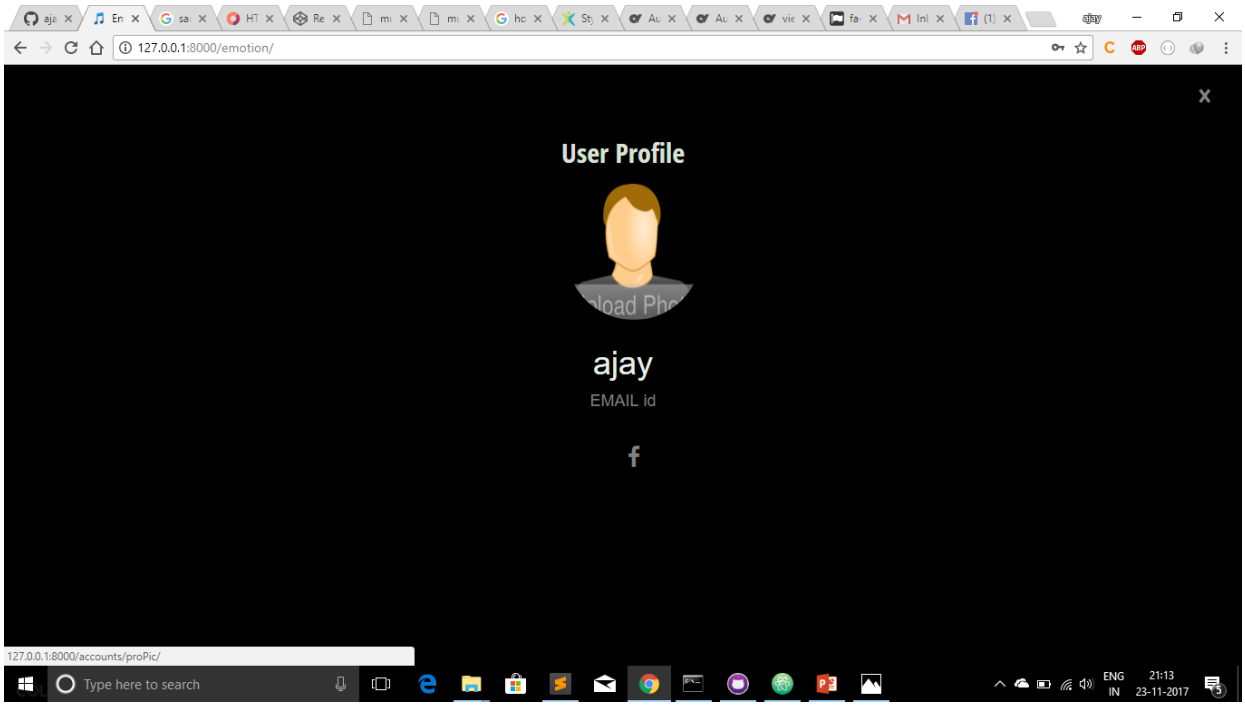
- Scan image
  The software will capture the image of the user using webcam and detect the user's current emotion.



- Emotion Recognition and Playlist generation
  The software would analyze the image properties and determine the mood of the user and yield a list of songs from a playlist in conformance with a user's emotional state.

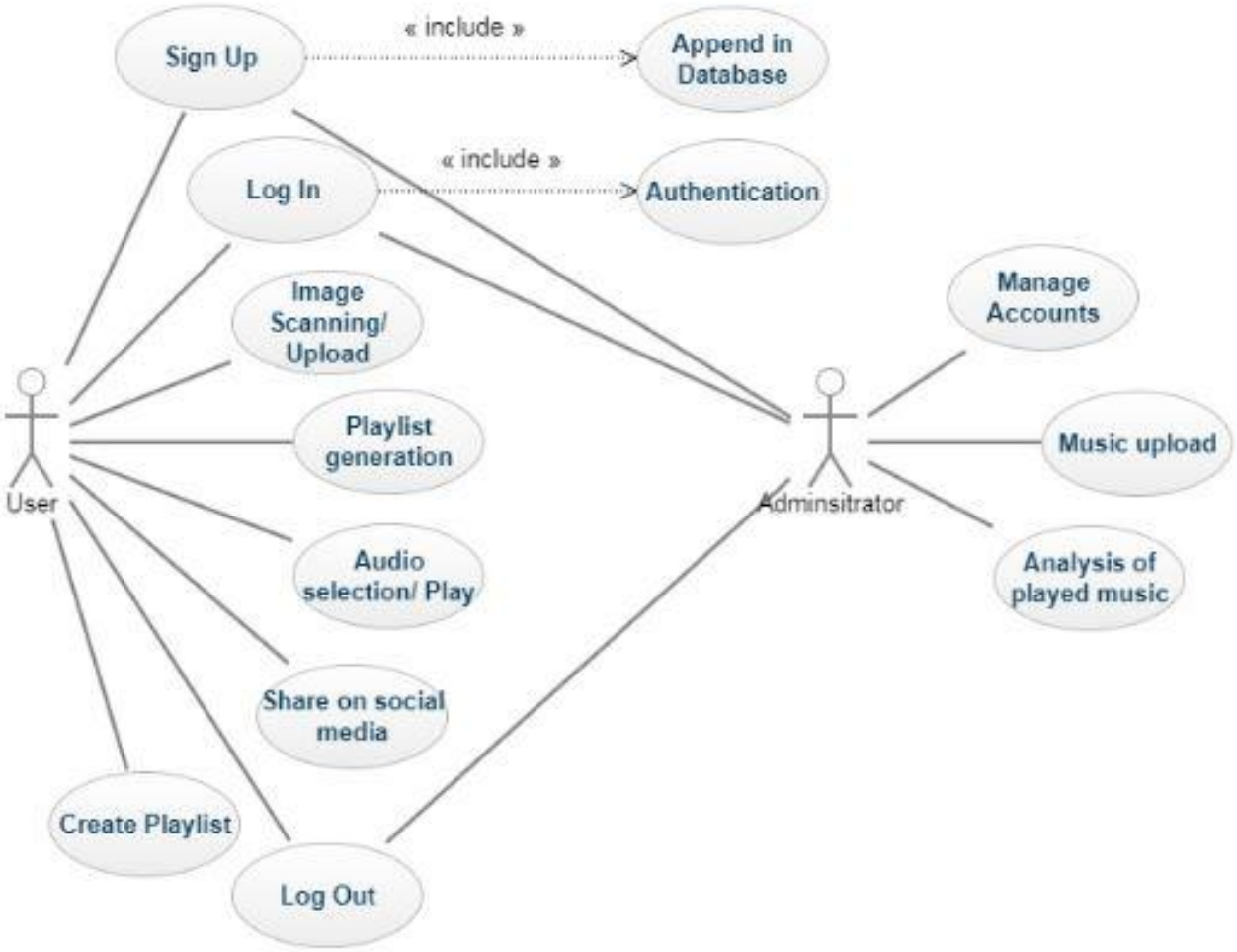


- User’s Profile



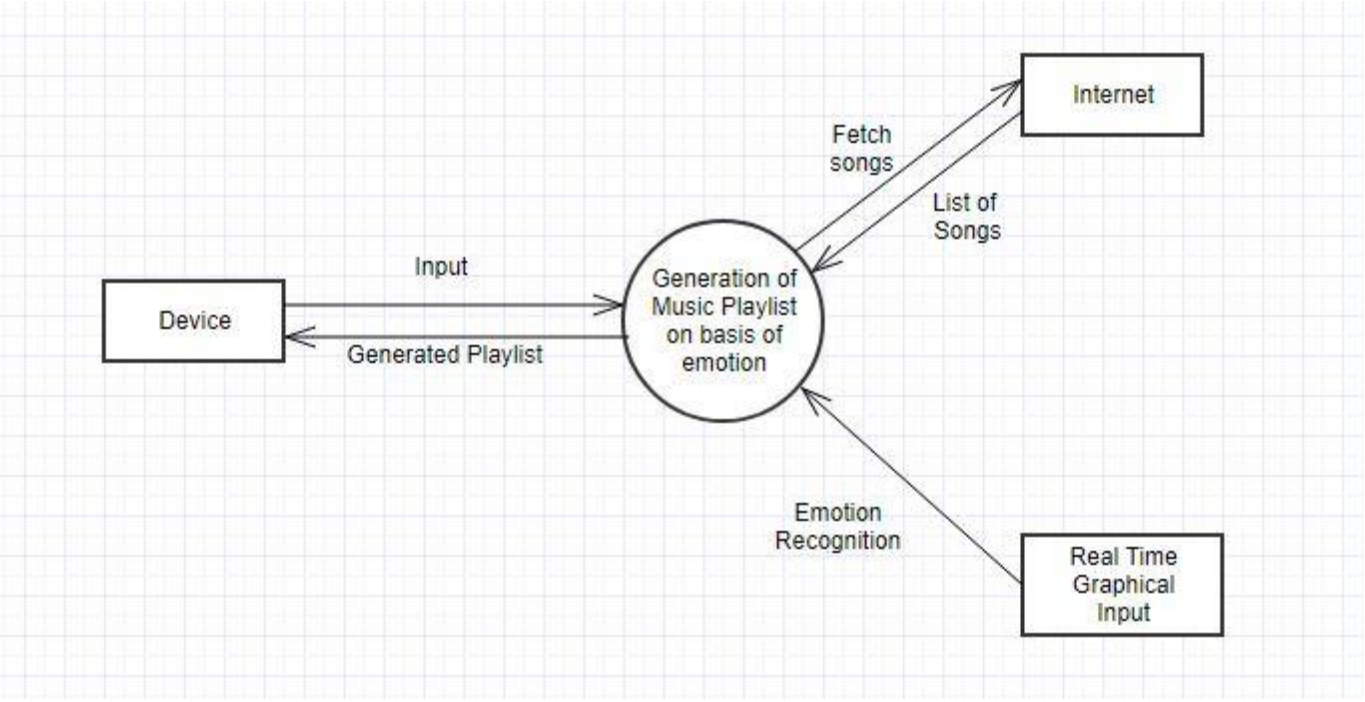
Appendix:

1. **Use Case Diagram**

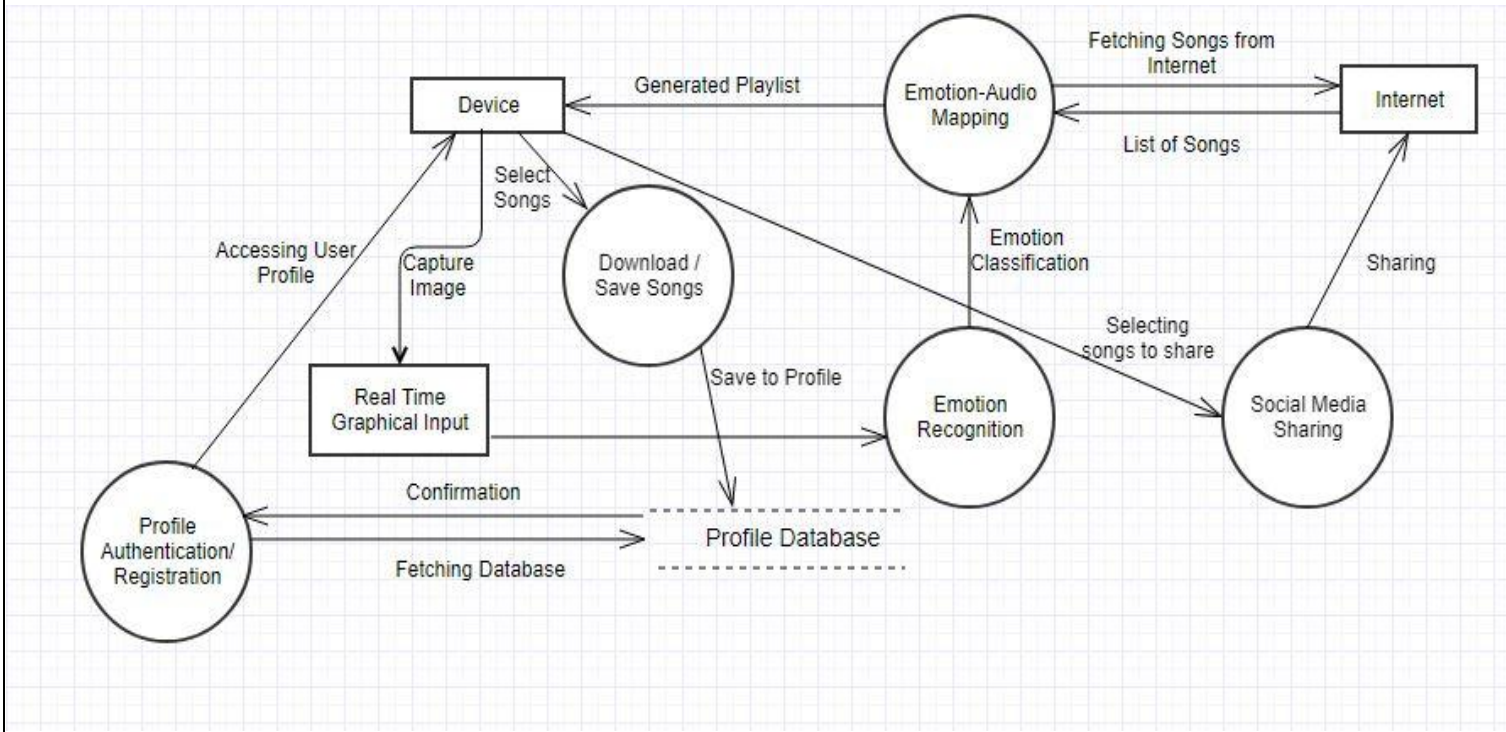


2. Data Flow Diagram

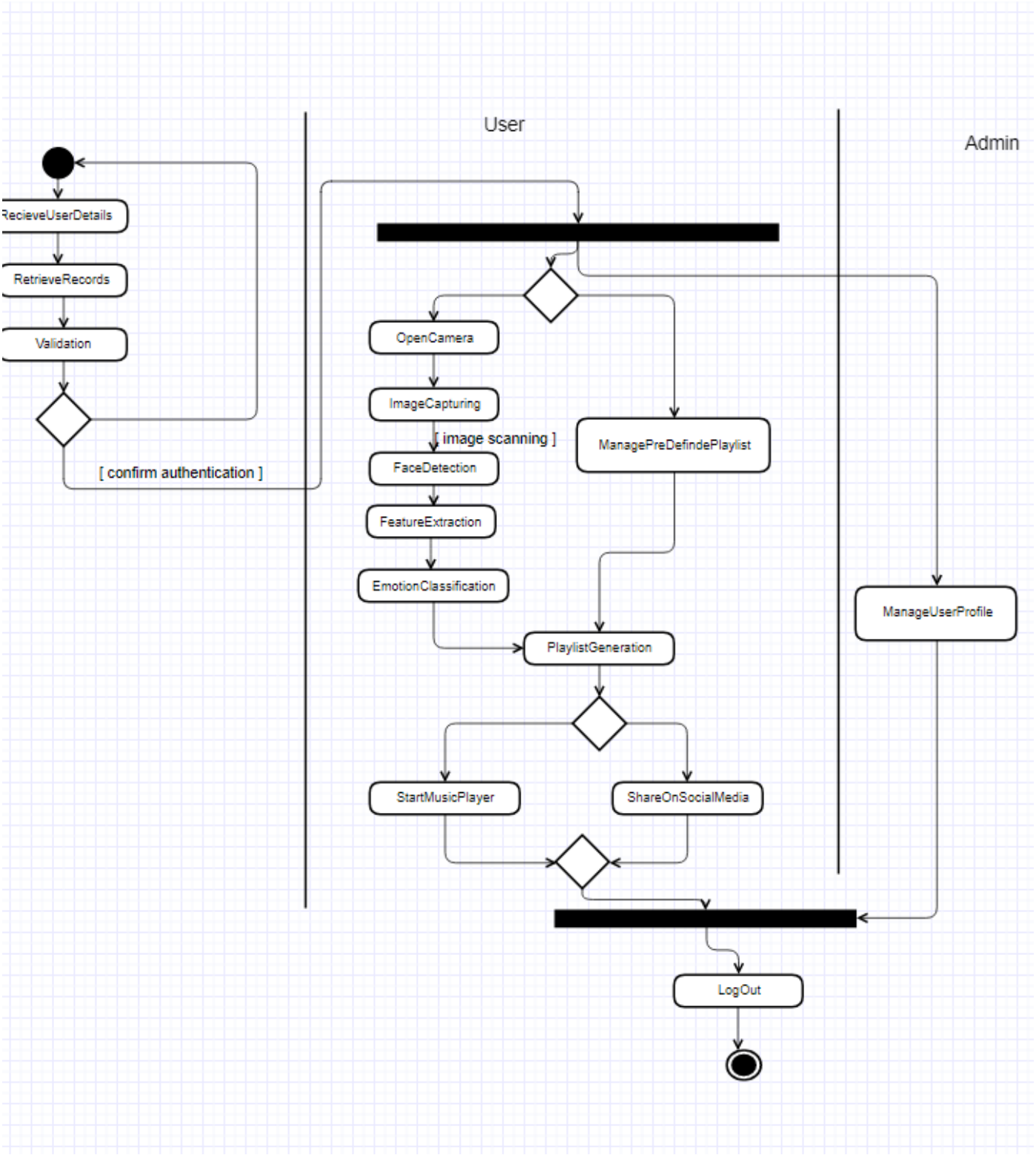
Level 0:



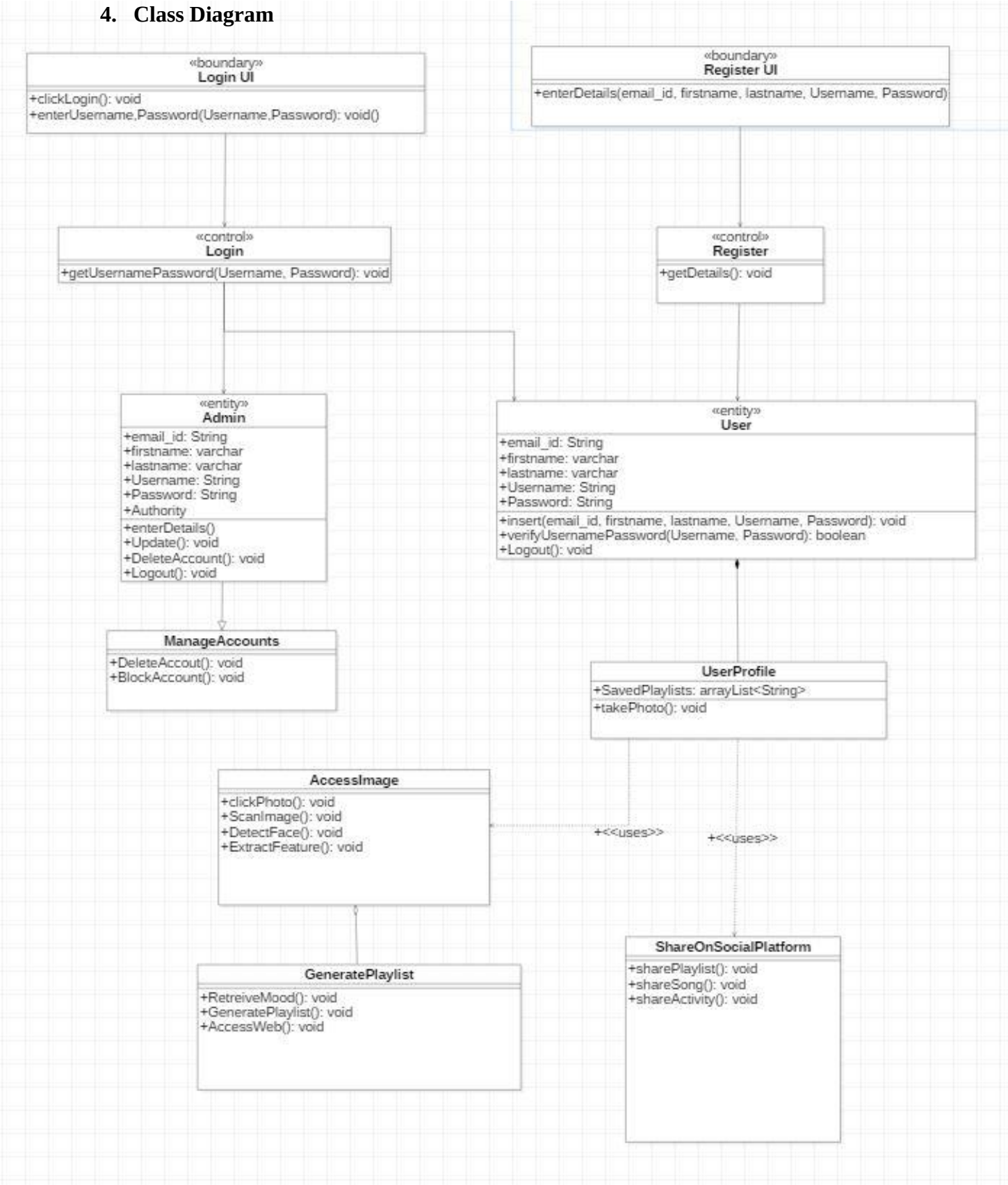
Level 1:



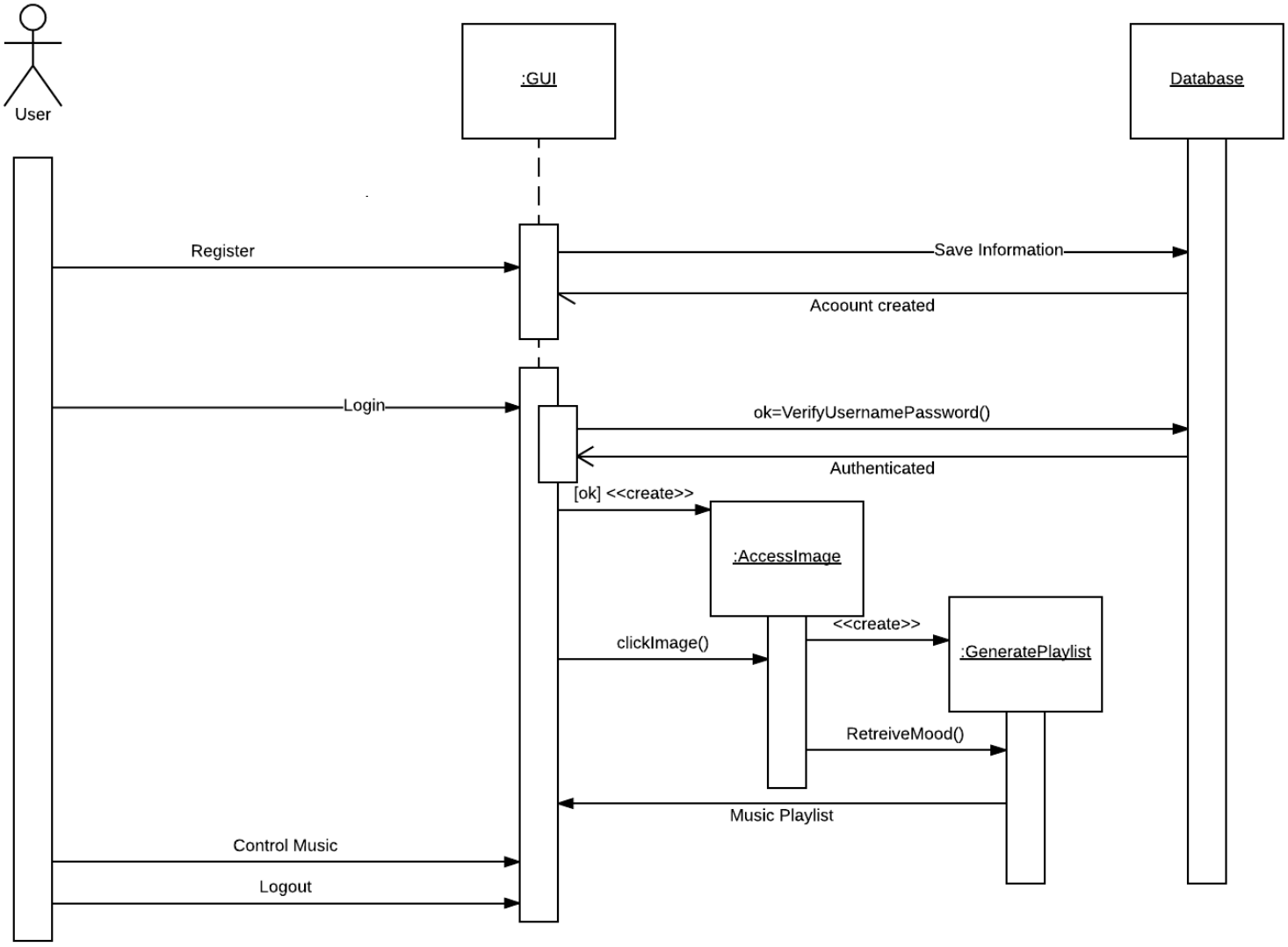
3. Activity Diagram



4. Class Diagram



5. Sequence Diagram



6. Collaboration Diagram