**MESS (Musical Entrance Security System)**

**Team: HGCG**

1. Tvisha Gangwani (trg2128)
2. Jiawen Li (jl5303)
3. Evan Ziebart (erz2109)
4. Jiahong He (jh3863)

**Topic:**

Personal identification system that plays “your favorite song” when you walk in.

**Summary:**

The system implements facial recognition to identify people entering a room, such as an office or a school, and then plays their favorite song out loud. The system will be run on a Raspberry Pi connected to a Pi Camera pointed at a room or building entrance. The Raspberry Pi can optionally be connected to a speaker for louder sound output.

An administrator interacts with the system through a command line interface, and will have an ability to add, modify, and delete users from tracking in the system. This data is stored in a database of user information which will contain names, pictures, and favorite songs. The administrator can specify the pictures of users which the system uses to identify them and manages song lists for the users.

To interface with the system, a user needs to approach the connected camera and look into it for long enough that the system can identify who they are. The Raspberry Pi runs a facial recognition module that implements deep learning to identify which user in the database is currently in view of the camera. It then accesses the user’s data and commands the system to play their song. Additionally, the system will have the ability to recognize a person who is not in the database and record their picture or send an alert to an administrator.

**Technology used:**

1. Raspian (a version of Linux)
2. Python
3. Raspberry PI
4. PI cam
5. Speaker
6. Database- SQL
7. Python library:<https://github.com/ageitgey/face_recognition>
8. Github

**Use cases and target users:**

1. Employees of big conglomerations
2. Children entering school
3. Home parties
4. Weddings

**User Stories**

* **Administrator**
* [A0]: As an administrator, I want the ability to add new users to the system, so that they can access it. My conditions of satisfactions are:
  + - Once a user is added, they will remain in the system even when it is turned off and on again
* [A1]: As an administrator, I want to remove a user from the system, in order to prevent them from accessing the system or to save space. My conditions of satisfaction are:
  + Once a user is removed, their information is deleted and will not return
  + Once removed, a user can no longer access the system
  + [A2]: As an administrator, I want an ability to update user data in the system, such as name, password, photo, and songs, so that I can control how the system treats a user and fix a user’s errors. My conditions of satisfaction are:
    - Able to change a user’s name in the system
    - Able to change a user’s password in the system
    - Able to change a user’s photo in the system
    - Able to change a user’s song list in the system
  + [A3]: As an administrator, I want to be able to specify custom behavior for edge and error cases in the system, so that I can handle unexpected situations. My conditions of satisfaction are:
    - Can specify a default noise and action when a person is not known to the system
    - Can specify a default noise and action when a user’s song cannot be played
* **Users**
  + [U0]: As a user, I want the system to correctly identify me when I look into the camera, so that it can greet me and play my song. My conditions of satisfaction are:
    - Once I look at the camera, the system can correctly recognized my face
    - Using facial recognition to get my personal preferences
  + [U1]: As a user, I want to be able to update my preferences such as name, photo, and songs, so that I can control how the system responds when I look at it. My conditions of satisfaction are:
    - Able to change my name in the system
    - Able to change my photo in the system
    - Able to change my song list in the system
  + [U2]: As a user, I want my information in the system to be password protected, so that others cannot change my information to things I don’t want. My conditions of satisfaction are:
    - I have an ability to login to the system using my password
    - Other users cannot access my information or update my preferences without knowing my password
    - My password is stored securely
  + [U3]: As a user, I want the ability to update my password in the system, so that I can change it into something I can easily remember. My conditions of satisfaction are:
    - Able to update my password

**Unit Tests**

1. Story: [A0]

Inputs: run add user command with creation info. as arguments

Outputs: users list should now include new user info.

1. Story: [A1]

Inputs: run remove user command with user id as argument

Outputs: users list should now not include user info.

1. Story: [A2]

Inputs: run update user command with updated values as arguments

Outputs: accessing user attributes should now include updated values

1. Story: [A3]

Inputs: run command to change default noise for unknown user

Outputs: image of unknown user should trigger the default noise

1. Story: [A3]

Inputs: add user with null song attribute; run command to change default noise for user

Outputs: image of the user should trigger the default noise

1. Story: [U0]

Inputs: A user’s data is in the system; the user looks into the camera

Outputs: The user’s specified song should play

1. Story: [U1]

Inputs: The user runs an update info command with updated values as arguments

Outputs: The user’s attributes now display as the updated values

1. Story: [U2]

Inputs: At login, enter a user id and correct password

Outputs: The system reports the user is now logged in

1. Story: [U2]

Inputs: At login, enter a user id and incorrect password

Outputs: The user is not logged in, and the system reports incorrect password

1. Story: [U2]

Inputs: User attempts to run administrator commands to add/remove users

Outputs: Command fails with error message

1. Story: [U2]

Inputs: User attempts to update data for another user

Outputs: Command fails with error message

1. Story: [U3]

Inputs: User is logged in; enters a command to change their password; logout

Outputs: Entering new password at login logs the user in; entering old password at login fails to log the user in

**Potential Technical challenges:**

We could probably have some problem storing the pictures in a MySQL database. To solve this, we would potentially store images in directories on the local file system and store references to the images in the database. Also, as the hardware of Raspberry Pi is not the best in terms of computational power, running the database system locally on the Raspberry Pi might be slow. A potential solution for this problem is to use cloud databases like Google Firebase.