# Second Iteration Demo

**Team HGCG**

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

**Summary**

1. The demo was on 12/3/2018, everything goes quiet smoothly and nothing unexpected arose during the demo.
2. In first iteration, we implemented user cases(details for each user case in **Implemented User Cases** section):

* [A0]
* [A1]
* [A2]
* [U0]

Since first iteration, we implemented addition user cases(details for each user

case in **Implemented User Cases** section), which are listed below:

* [U1]
* [U2]
* [U3]

No changes to these implemented user cases so far.

1. In our CI implementation, during each push to the repository, the CI server would:

* Set up the environment (make sure the correct version of Python is installed).
* Install all the dependencies.
* Start the Django server.
* Run all unit tests (written with Python unittest module).

Any one of the above step failed Travis would prevent the changes from being made to the repository.

For this iteration, new tests were added for each of our newly implemented user stories. The tests now also include the generation of a code coverage report with coverage.py.

1. Github: <https://github.com/JiahongHe/Personal-Facial-Identification-System>

**Implemented User Cases**

1. **Administrator**
   1. **Title**: [**A0**] Add new user to the system
      * **Actor(s)**: The System (Djingo)
      * **Description**:
        + As an administrator, I want the ability to add new users to the system, so that they can access it. Once a user is added, they will remain in the system even when it is turned off and on again
      * **Basic flow**:
        + The system successfully added user to the system.
      * **Alternate flows**:
        + While the system adds user to the system, user’s name or password or photo, or song list error. The system failed to add the new user to the system
        + Once restart the system, the newly added user’s information disappeared.
   2. **Title**: [**A1**] Remove a user from the system
      * **Actor(s)**: The System (Django)
      * **Description**:
        + 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. Once a user is removed, their information is deleted and will not return. Once removed, a user can no longer access the system
      * **Basic flow**:
        + The system successfully removed user from the system.
      * **Alternate flows**:
        + Once the user removed from the system, this user still can login to the system with his/her removed password to get the removed information.
   3. **Title**: [**A2**] Update user information in the system
      * **Actor(s)**: The System (Djingo)
      * **Description**:
        + 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. The conditions of satisfaction are be able to change a user’s name, password, photo, song list in the system.
      * **Basic flow**:
        + The system successfully updated user information.
      * **Alternate flows**:
        + While the system updates user information, user’s name or password is not found or incorrect
        + While the system updates user information, user’s photo or user’s song list is not found or its format is incorrect

1. **Users**
   1. **Title**: [**U0**] System can identify user through camera
      * **Actor(s)**: The System (Django), User
      * **Description**:
        + 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. Once I look at the camera, the system can correctly recognized my face. Using facial recognition to get my personal preferences
      * **Basic flow**:
        + User can be successfully identified by the system. Then, the system grab information about this recognized user’s, and play his/her favourite song.
      * **Alternate flows**:
        + While the system identifies user through camera, user is not staying still, so that the camera cannot capture a clear image of user’s face.
        + While the system grabs relevant information about the user, the user is not exist, which the information is not found. Then, the system categorize this user as unknown.
        + While the system grabs relevant information about the user, this user’s song is not playable, which might be an audio file error.
   2. **Title**: [**U1**] Update user’s preferences
      * **Actor(s)**: The System (Django), Users
      * **Description**:
        + As a user, I want to be able to update my preferences such as full name, photo, and songs, so that I can control how the system responds when I look at it. My conditions of satisfaction are be able to change my first and last name, photo, and song list in the system.
      * **Basic flow**:
        + User’s preferences can be successfully updated and stored into the system.
      * **Alternate flows**:
        + While the user updates his/her preferences, user failed to update any of the preferences.
   3. **Title**: [**U2**] Protected personal information with secured password in the system
      * **Actor(s)**: The System (Django), User
      * **Description**:
        + 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. 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
      * **Basic flow**:
        + User’s personal information are secured in the system, which is protected by a secured password.
      * **Alternate flows**:
        + While the system protects user’s personal information, user failed to login to the system by his/her secured password
        + While the system protects user’s personal information, users can access and update other users’ personal information and preferences.
        + While the system protects user’s personal information, user’s password is not stored securely.
   4. **Title**: [**U3**] Update user’s password
      * **Actor(s)**: The System (Django), Users
      * **Description**:
        + 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. Once I have created my account with the system, I can be able to update my password whenever I want.
      * **Basic flow**:
        + User’s password can be successfully updated and stored into the system.
      * **Alternate flows**:
        + While the user updates his/her password, user failed to update the password, which still need to use the old password to login to the application.