

**Sprint 1 Retrospective Review**

Team 18: Chunao Liu, Anurag Shah, Jenna Zhang, Yierpan Abuduwaili, Michelle He, Jingyuan Yang

**What Went Well?**

In general, we have a basic UI setup for most of our Front-end feeds, and each UI component has the correct action handler to send and receive requests from the backend. Basic user information can be displayed correctly and requests for adding new users or groups are available. The backend server is deployed on an AWS EC2 instance and ports have been configured so that our backend app can accept or send HTTP requests from and to the open internet. Django Models for user, group, and pictures are functioning and can interact with the local MySQL server correctly using ORM.

* + User Story #1(Welcome Page)

As a user, I would like to see an introduction page with the logo of the APP when I open the program

| # | Description | Estimated Time | Owner |
| --- | --- | --- | --- |
| 1 | Create UI page with the WhiteBoard logo | 1 hrs | Yierpan |
| 2 | Checks if the user is logged in(remember me) or not and redirects the user to different Camera pages | 3 hrs | Yierpan |
| 3 | Debug and test algorithm using unit tests | 3 hrs | Yierpan |

**Completed:**

App starts and functions normally, the splash screen will show during the loading.

* + User Story #2(Login Page)
* As a user, I would like to be able to sign in to my account.

| # | Description | Estimated Time | Owner |
| --- | --- | --- | --- |
| 1 | Create a UI page with a login form that allows users to enter their username and password | 1 hr | Jenna |
| 2 | Create a User entity Class on the back-end which contains all info of a user | 1 hr | Jenna |
| 3 | Implement login processing functions on the back-end to receive login requests from the front-end, communicate with the database, and send login results back to the front-end. | 3 hrs | Jenna |
| 4 | Set up the connection between the server and the database | 2 hrs | Jenna |
| 4 | Implement login authentication on the back-end | 2 hrs | Jenna |
| 5 | Debug and test algorithm using unit tests | 4 hrs | Jenna |

**Completed:**

Each component of the login UI page functions well. Backend login function is implemented and communicates normally with frontend and the database.

* + User Story #3

As a user, I do not need to log in again if I’ve logged in before and did not log out.

| # | Description | Estimated Time | Owner |
| --- | --- | --- | --- |
| 1 | Create a “remember me” checkbox on Login Page | 1 hr | Jenna |
| 2 | Create a login session for each user (both front-end and back-end). | 2 hrs | Jenna |
| 3 | Debug and test algorithm using unit tests | 4 hrs | Jenna |

**Completed:** A “remember me” checkbox is created on the Login Page and functioning well. The device won’t ask users to login again if they are active within a month. The device remembers the user’s account info if the user has logged in before.

* + User Story #4  
    As a user, I would like to have a way to reset my password.

| # | Description | Estimated Time | Owner |
| --- | --- | --- | --- |
| 1 | Create a Forgot Password Button where users can click to reset their passwords | 1 hr | Jenna |
| 2 | Create a pop-up window that prompts users to enter an email linked to their accounts. | 1 hr | Jenna |
| 3 | Implement reset password functions on the back-end to receive reset password requests from the front-end and send an email with a reset password link to the given email address. | 5 hrs | Jenna |
| 4 | Create an HTML page where users can enter and submit their new passwords. | 1 hr | Jenna |
| 5 | Implement update password functions on the back-end which receive new passwords from the user and send updated password queries to the database | 1 hr | Jenna |
| 6 | Debug and test algorithm using unit tests | 4 hrs | Jenna |

**Completed:** A “forgot password” button is created on the Login Page and functioning well. It prompts the user to enter their emails and check if the given email is valid and registered.

* + User Story #5(Create Account)

As a user, I would like to be able to create my own account.

| # | Description | Estimated Time | Owner |
| --- | --- | --- | --- |
| 1 | Create a Sign-Up button on the Login Page that redirects to the Registration page | 1 hr | Michelle |
| 2 | Create a UI page with a registration form with allows users to enter their username, email address, password, and password confirmation | 2 hrs | Michelle |
| 3 | Create error messages that show when the email address is in an invalid form and when the password and the confirmed password are different respectively | 1 hr | Michelle |
| 4 | In the back-end, create a registration processing function to check the username is not already in use by other users and check the email address is not already linked to an account. Sends the successful/error message respectively and adds the users to the database if successful | 3 hrs | Michelle |
| 5 | Display the message that tells the users to either change a username or change an email address if creating an account is not successful | 2 hrs | Michelle |
| 6 | Create a popup window to notify that the account is successfully created and send an email for confirmation | 2 hrs | Michelle |
| 7 | Debug and test algorithm using unit tests | 4 hrs | Michelle |

**Completed:**

The text fields to input username, email address, password, and confirmation password has been created. Error messages correctly display when the email address is in an invalid format or when two passwords are different.

* + User Story #6(Team page)

As a user, I would like to manage and see my team’s information.

| # | Description | Estimated Time | Owner |
| --- | --- | --- | --- |
| 1 | Create a list of teams that the user is in | 2 hrs | Jingyuan Yang |
| 2 | The list of teams should include the name of the team, the coordinator, and a short description of the team depending on whether a team has one | 3 hrs | Jingyuan Yang |
| 3 | Create a button that leads users to a page that allows users to create a team | 3 hrs | Jingyuan Yang |
| 4 | Create a team page should allow the user to input team name, team description, and set the user to the coordinator, creates a library that contains the code for the team | 4 hrs | Jingyuan Yang |
| 5 | Users should be able to move to the page specific to a team by clicking the team that shows team members, contact information or the members and coordinator, and the code shared amongst the team, user should be able to share code to the code library of the team  If the user is the coordinator of the team, the coordinator should have an ADD button that adds members to the team | 8 hrs | Jingyuan Yang |
| 6 | Create a button to allow users to go back to the Team list, team list page should have a button that takes the user back to the Camera page(Home page) | 4 hrs | Jingyuan Yang |
| 7 | Test the functionality, debug and test algorithm using unit tests | 4 hrs | Jingyuan Yang |

**Completed:**

User can see a list of their team member as a contact list on the phone

Name, email, and Avatar picture can be retrieved from the backend server using fetch and displayed in the list.

* + User Story #7(Account page)

As a user, I would like to be able to see and edit my account information.

| # | Description | Estimated Time | Owner |
| --- | --- | --- | --- |
| 1 | Create UI page with user profile picture, username, email address and their team information | 2 hrs | Yierpan |
| 2 | Users should be able to change their user profile picture, username, email address and password(change password send request to backend implemented for resetting password in Forgot Password) | 3 hrs | Yierpan |
| 3 | Users should be able to see the teams they belong to and be able to be redirected to the Team Page | 2 hrs | Yierpan |
| 4 | Create UI button so users should be able to go to their team page or the Camera Page | 2 hrs | Yierpan |
| 5 | Test the functionality, debug and test algorithm using unit tests | 4 hrs | Yierpan |

**Completed:**

Account page UI implemented so Users will be able to see their account information and will be able to edit their user info.

* + User Story #8(Camera page)

| # | Description | Estimated Time | Owner |
| --- | --- | --- | --- |
| 1 | This page will be the root page of this app. Just like a normal camera app, the middle area of the screen would display the real time view captured by the camera. | 6 hrs | Yierpan |
| 2 | There should be a collapsible sidebar on the upper left corner that shows different functions and redirects user to different pages(Save, Library, Team, Account, Logout) | 8 hrs | Yierpan |
| 3 | If user is not logged in, they should only be able to see a login button, different UI, this function is connected with introduction page | 1 hr | Yierpan |
| 4 | Create a pop of window after the picture is taken, user needs to select Accept to send the picture to OCR and compile or Decline to retake the photo | 4 hrs | Yierpan |
| 5 | Users can either take a photo of the code or import a photo of the code from their local picture library and send it to backend compile | 2 hrs | Yierpan |
| 6 | Users should be able to click on the keyboard icon on the upper right corner of the page and be redirected to the Text Editor Page | 1 hr | Yierpan |
| 7 | Create a UI that shows the result sent back from compiler | 2 hrs | Yierpan |
| 8 | Test the functionality, debug and test algorithm using unit tests | 4 hrs | Yierpan |

**Completed:**

Camera UI completed, pictures can be taken and sent to backend, image picker function implemented so users can pick pictures from their gallery. Page redirects and drop down menu, sidebar and topbar functions well, will be further implemented to other UI pages.

* User Story #9(Text Editor Page)

As a user, I would like to type the code in the text editor and run it.

| # | Description | Estimated Time | Owner |
| --- | --- | --- | --- |
| 1 | Create UI page with a text editor that allow users to directly type their code | 1 hr | Michelle |
| 2 | Create a sidebar that include access to additional features if the users are logged in | 1 hr | Michelle |
| 3 | Create a run button that sends the code to the back end | 3 hrs | Michelle |
| 4 | Create a popup window that asks the user whether they confirm to submit the code | 2 hrs | Michelle |
| 5 | In the back end, create a run algorithm that accepts the code of the users, compiles the code, and runs the code. If it compiles successfully, the algorithm sends the run results back; if not, it sends the error back | 5 hrs | Michelle |
| 6 | Display the compilation results and run results | 2 hrs | Michelle |
| 7 | Debug and test algorithm using unit tests | 4 hrs | Michelle |

**Completed:**

A text editor that allows users to type their code is created. A sidebar displays more features. The run button functions correctly by popping up the confirmation message window and then sending the code typed.

User Story #11(Backend networking)

| # | Description | Estimated Time | Owner |
| --- | --- | --- | --- |
| 1 | Write a script that Infinitely listens to specific ports and receive requests from the front end | 2 hrs | Chunao |
| 2 | Upon receiving a sign-in request, authenticate the user’s information and send the result back to the user | 3 hrs | Chunao |
| 3 | Upon receiving a sign-up request, create a new entry in our database and send the status to the front-end | 3 hrs | Chunao |

**Completed:**

Although most of these are done by Django, we do have a running server that listens to a specific port and is able to detect and react to HTTP requests of user log-in, sign-up, or user info display.

* User Story #12(Compilation Script)

| # | Description | Estimated Time | Owner |
| --- | --- | --- | --- |
| 1 | Write a script that will take a piece of code, compile and run it with GCC. | 2 hrs | Anurag |
| 2 | Upon compilation and runtime, fetch any output, including stack trace and terminal output, and pack them into a JSON file | 2 hrs | Anurag |

**Completed:**

We have a functioning Script that can run individually. Given a picture or code, it can automatically compile it and run the code. It extracts the output from stdout and stderr, and is able to pipe that back into python. It works for code that compiles and code that doesn’t compile. Packing the output remains but has been moved to a different part of the architecture.

* User Story #13(Back-end Database)

As a developer, I would like to deploy a MySQL DBMS with specific tables to store any user and group’s information

| # | Description | Actual Hours Spend | Owner |
| --- | --- | --- | --- |
| 1 | Deploy a MySQL server on our back-end server | 2 hrs | Chunao |
| 2 | Create one default schema for all tables | 2 hrs | Chunao |
| 3 | Create a table of User Info in the default schema | 2 hrs | Chunao |
| 4 | Create a table of Group Info in the default schema | 3 hs | Chunao |
| 5 | Create a table of Output Info in the default schema | 3 hs | Chunao |
| 6 | Create a table of Authentication info in the default schema | 2 hs | Chunao |
| 7 | Write a trigger that when a new user signs up, a private group should be created and will become the default group of the user | 3 hs | Chunao |
| 8 | Debug and Unit tests to ensure the database works correctly | 5 hs | Chunao |

**Completed:**

Local MySQL server is functioning and corporate well with the Django App. All tables have been created and rows can be written. Basic Front-end Back-end communication is achieved via HTTP requests.

* User Story 14 (OCR for Typeform Data)

As a user, I would like the application to process my image of typeform code, and convert it to the relevant textual representation.

| # | Description | Estimated Time | Owner |
| --- | --- | --- | --- |
| 1 | Setup the OCR pipeline, and setup functions the backend can call for OCR. | 2 Hours | Anurag |
| 2 | Setup a pre-trained OCR model, and configure it to work with images of typeform text (printed or screenshot) of the correct size. | 4 Hours | Anurag |
| 3 | Setup the preprocessing suite for the OCR model in (2). | 4 Hours | Anurag |
| 4 | Create the end of the OCR pipeline, with the postprocess filter, a (for now) redundant filter for language detection, and pipe it into the Compiler (as in user story 12) | 2 Hours | Anurag |

**Completed:**

A script has been written that can automatically send a picture to the OCR functions. This allows OCR to be developed and deployed separately and higher team efficiency. The OCR functions as expected and is able to process photographs taken from the phone and manually entered as inputs.

* User story 15 (OCR for Handwritten Data)  
  As a user, I would like the application to process my image of handwritten code, be able to distinguish it from typeform code, and convert it to the relevant textual representation.

| # | Description | Estimated Time | Owner |
| --- | --- | --- | --- |
| 1 | Create a dataset of images with handwritten and typeform text, in various image sizes and resolutions. | 3 Hours | Anurag |
| 3 | Create a dataset of handwritten characters for the full ASCII set, with at least 100 images for each character. | 3 Hours | Anurag |

* **Completed:**

The two datasets for this portion were created from a variety of online datasets.

**What Didn’t Go Well?**

User Story #2:

Currently, the HTTP basic login authentication is used on the back-end, which is not secure.

User Story #3:

There’s no login session created for the user each time they login. The local storage stores user’s account info in the local storage, which can be insecure. It is more ideal to implement session authentication.

User Story #4:

Sending a password reset link via email is not implemented on the backend. When users click forgot password and enter their emails, the backend should be able to generate an email with password resetting link to the given email address. However, for now, only the frontend checks if the given email is valid or not. The backend does nothing yet.

User Story #5:

Sending the information gained from users to the backend and the registration processing function in the backend is not complete. The function should check whether the username is already in use by other users and whether the email address is already linked to an account, should add the new user to the database and send a success message to the frontend.

User Story #8:

The static picture taken by the camera is very zoomed in using the EXPO camera, it will be fixed during the next sprint by implementing RNCamera instead of EXPO camera, there is a post on EXPO website saying that they haven’t found a solution for the zoomed in image problem.

User Story #9:

A function that processes the code sent needs to be implemented. Still need to figure out how to display an error message precisely in the same place where the code is typen.

User Story #10:

The backend was not yet set up for file and photo transfer. So the Library page would be implemented after setting up the backend.

User Story #12

| # | Description | Estimated Time | Owner |
| --- | --- | --- | --- |
| 3 | Upon finalizing the output package, send it back to the correct User’s device in the same thread | 2 hrs | Chunao |
| 4 | Unit tests and Debug to ensure the correctness of the compilation and runtime | 3 hrs | Chunao |
| 5 | Unit tests and Debug to ensure the package is correct and it will be sent to the correct User’s device | 3 hrs | Chunao |

Most of these works are unfinished as picture serialization and transfer proved to be much harder than expected, and setting up the server also took longer than expected. We have an OCR wrapper function ready, but we still haven’t been able to connect it to the pipeline. This will be the first priority in sprint 2. Finally, unit tests for the compiler were not created, but the compiler was tested manually on a variety of inputs, and produced the expected results.

User Story #15:

There were several issues with architecture design for the models. Using a classic CNN based classifier for Handwritten vs Typeform data was not producing the desirable result, not even with the introduction of feature reconstruction or residual layers. Further, tasks on the compilation side, and integrating the OCR pipeline with the backend, were given priority with only 2 people working on the backend side; this meant that points 4, 5, 6 were sidelined. These three were stretch goals for the sprint regardless, so we do not think those 3 are a major issue.

**How should you improve?**

Overall, everyone participated well as a team in the first sprint, and

all members met the 10-hour minimum contribution every week. Our communication, especially early on, was very good, and while responsibilities shifted between people occasionally, there were never any conflicts or disagreements over this.

The main problem we had this sprint was everyone was getting busy at the end of the sprint because of the midterms and projects. We couldn’t meet and combine our components at the end of the sprint, even though it wasn’t included in our plan for sprint 1, but we could’ve done better if we had more time and better communications. So, for improvement, we will make better plans and reschedule some of our meetings to have better communication in sprint 2.

Some of our initial goals and estimates were not perfectly aligned to changes we encountered during the sprint, and thus the final product after the three weeks. Improvement here would involve better division of tasks between front and back end, more realistic and conservative estimates of time, and building in some lee-way in tasks in order to accommodate changing ideas and priorities.

Finally, we should aim for better integration between individual units. Part of this was, of course, that the project was new, and we were building individual units for most of our goals. But it is essential for an application like this to have even basic communication between its various parts, and we must dedicate time just to integrating app pages or larger units between each other.