

# Office Hour Tracker - Sprint 2 Retrospective

2024/10/25

1. Dates of sprint: Mon, Oct 7, 2024 – Fri, Oct 18, 2024

2. Links:

a. Project management tool Page:

<https://tamu-team-office-tracker.atlassian.net/jira/software/projects/SCRUM/boards/1>

b. Github Repo:

[et-tran50/CSCE\\_606\\_Office\\_Hours\\_Tracker](https://github.com/et-tran50/CSCE_606_Office_Hours_Tracker)

c. Heroku deployment: <https://office-hours-tracker-a63f1f6d64ad.herokuapp.com/>

3. Team member contributions (an overview of team member capacity and effort done in the sprint. A table / chart of team members' percent of fair share effort):

Story	Status	Story Points	Assignee
As a <scrum master>, I want to <arrange daily meetings and remove difficulties encountered by teammates>, so that <the developers will have reasonable progress during the sprint>	Completed	2	Chen-Hung
As a <Student/TA>, I want to see <already registered when marking attendance repeatedly>, so that <a new entry is not added to same time slot.>	Completed	2	Gourangi
As a <student>, I want to <select from a dropdown of course>, so that <I know which course I am marking attendance for>	Completed	2	Ethan

As a <Admin/TA/Student>, I want to <know my role after login>, so that <I can check if I am registered correctly>	Completed	2	Bryson
As a <product owner>, I want to <prioritize the story that should be addressed in this sprint>, so that <the client is able to use the product after this sprint.>	Completed	2	Franklin
As a <TA> I want to <show I am here>, so that <system can record the attendance of me>	Completed	2	Nazaro
As an <Admin>, I want to <download the attendance data from TA and student>, so that <I know the number of person joined the slot>	Completed	2	Nhat

#### 4. Sprint goal: Chen-Hung Wu

Meet the minimum usage requirements so the client can begin offering it to students for testing. This includes differentiating roles (admin/TA/student), allowing students to check in based on their courses, and enabling admins to select the time range for downloading attendance data.

#### 5. Sprint Achievement: Chen-Hung Wu

During this sprint, the following user stories were successfully implemented:

- Differentiate roles (admin/TA/student): Once a user logs in, we will compare their login email with the admin database and the TA database. If it matches neither, they will be classified as a student and directed to the corresponding page.
- Admin can download specific data: First, we separate the attendance data of TAs and students into two tables. Second, the admin can select a time range and download the specific statistical information.

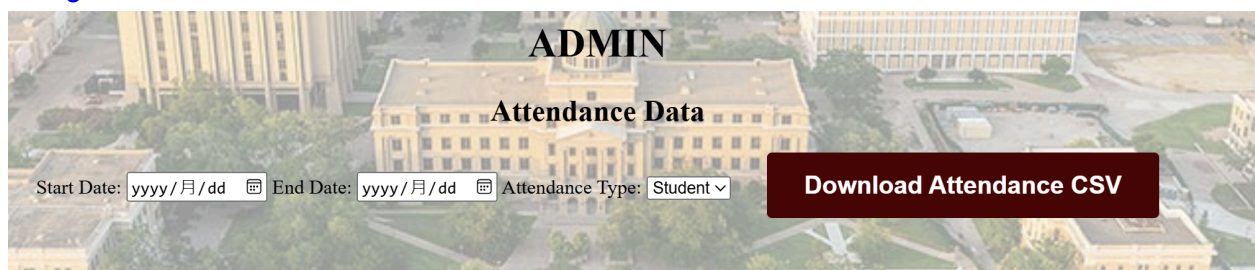
- Students can check in based on their courses: Add a menu that allows students to select which course session they want to check in for after logging in.

## 6. Sprint backlog item and status Wei Chien Kao

Below is the story included in the Sprint 2 plan. The change after the sprint starts is mainly about revising the implementation of the stories in code to make the development workflow smoother. The affected stories will be marked (\*) after index.

1) (\*) As an <Admin>, I want to <download the attendance data from TA and student>, so that <I know the number of person joined the slot>

Change/Reason: The actual implementation is changed during the sprint because we find creating two buttons for TA and student download will conflict with the option of choosing data range. The current UI is shown below. Information in three boxes have to be specified before hitting the download button.



**Point (2) - Nhat**

**Estimated Time: 3 hr**

**Task:**

- Create two download buttons to download the attendance information of TA and student
- Add flexible choice such that admin can choose to download the attendance based on specific course or time range

2) As a <Student>, I want to <select from a dropdown of course>, so that <I know which course I am marking attendance for>

**Point (2) - Ethan**

**Estimated Time: 3 hr**

**Task:**

- We make the student choose which course they are signing in for before they sign in, only until after the attendance button will appear.
- Connect the button so that student's attendance can be saved correctly into the database.

3) As a <TA> I want to <show I am here>, so that <system can record the attendance of me>

**Point (2) - Nazar**

**Estimated Time: 3 hr**

**Task:**

- i) Create the TA home page
- ii) Connect the button so that TA's attendance can be saved correctly into the database.
- iii) Add some CSS to different pages to make it more attractive

4)(\*) As a <Admin/TA/Student>, I want to <know my role after login>, so that <I can check if I am registered correctly>

[Change/Reason: The current implementation doesn't store the TA/admin information into file, instead, certain files are queried to determine roles.](#)

**Point (2) - Bryson, Nazar**

**Estimated Time: 3 hr**

**Task:**

- i) The database need to mark the identification for admin/TA
- ii) When logging in, the app can query certain file to know the role of user (Admin or TA or Student)
- iii) The view will show different welcome message based on their role

5) (\*) As a <Student/TA>, I want to <see an error message when marking attendance repeatedly>, so that <I mark my attendance once in every timeslot.>

[Change/Reason: To ensure consistency, we move the database creation \(TAattendance\) to Gorangi.](#)

**Point (2) - Gourangi**

**Estimated Time: 3 hr**

- i) Add a check mechanism before students mark their attendance. They should only be able to mark attendance once every hour.
- ii) The Students should see "Attendance already marked for the time slot" if they are already marked attendance at the specified timeslot
- iii) Make sure that they are able to mark attendance for any other time slot.
- iv) Create the database based on the diagram

6) As a <product owner>, I want to <prioritize the story that should be addressed in this sprint>, so that <the client is able to use the product after this sprint.>

**Point (2) - Franklin**

**Estimated Time: 3 hrs**

- i) Discuss with teammates about the story that should be put in this sprint
- ii) Write rspec tests to ensure that each component works normally.

7) As a <scrum master>, I want to <arrange daily meetings and remove difficulties encountered by teammates>, so that <the developers will have reasonable progress during the sprint>

## Point (2) - Chen-Hung

### Estimated Time: 3hrs

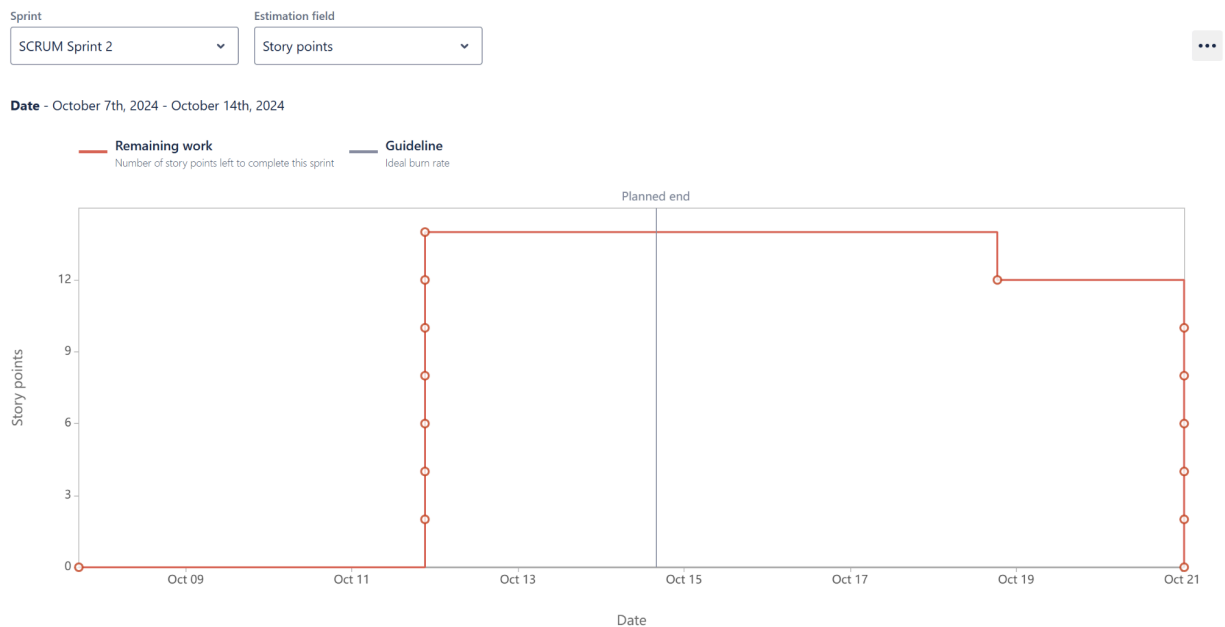
i) Hold regular stand-up meetings to know about people's progress and to remove obstacles in the development process

ii) Write cucumber tests to ensure that the functions are working properly.

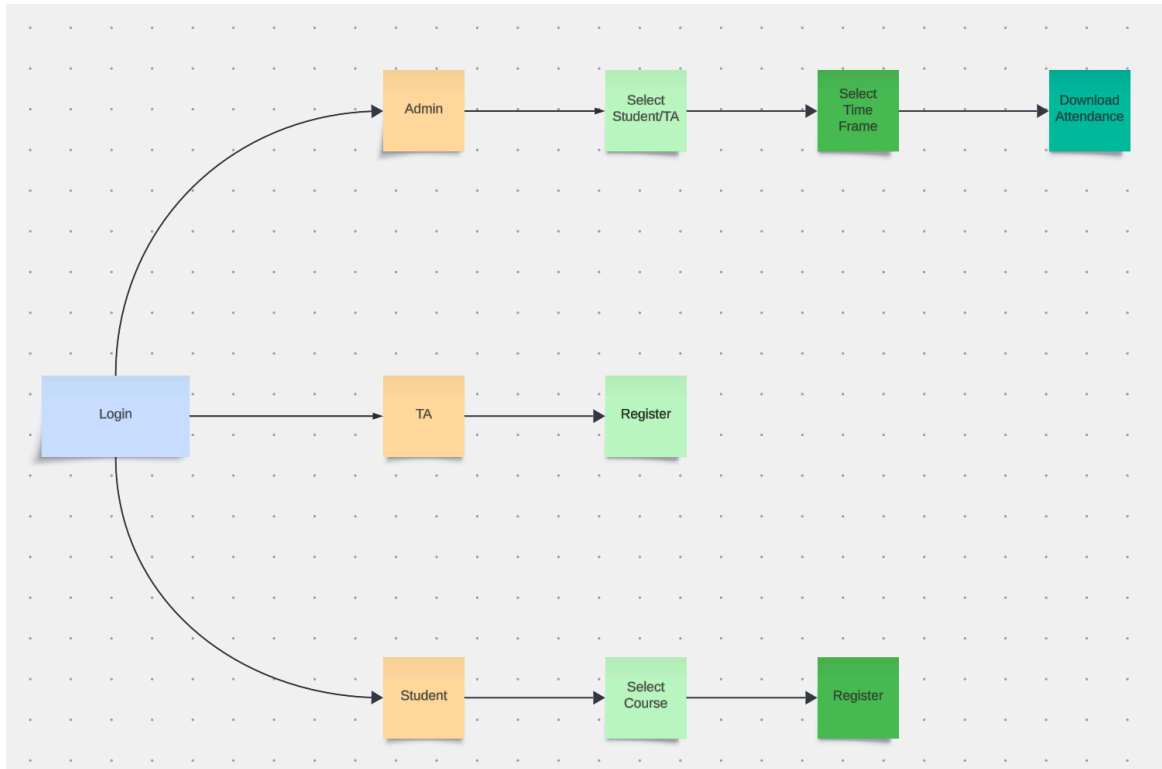
iii) Arrange weekly meetings with TA

## 7. burndown(A burndown chart for the sprint.):

Most people turn in their work close to the end of the sprint, so we update the status of each story very late. Our team will add a new rule to turn in work early at the team working agreement.



## 8. Design Diagram for the sprint Chen-Hung Wu



## 9. Documentations of changes: Nhat Nguyen

Following is the burnup report of the story points throughout the sprint:

**Sprint Duration:** Mon, Oct 7, 2024 – Fri, Oct 18, 2024

### 1. **Sprint Initiation:** The sprint began with the addition of the following tasks:

- Admin CSV download panel
- Course selection during sign in
- TA Clock in
- Role assignment account permissions setup
- Implementing front end safety for students marking attendance

### 2. **Progress Updates:**

- As issues were completed, the total completed story points increased from 54 (sprint 1) to 76 (sprint 2, 14 points + 8) by the end of the sprint.
- Several tasks were added and their estimates were updated throughout the sprint, including:
  - Rspec Tests (3 points)

- Cucumber Tests (3 points)
- Database for User (2 points)

### 3. Scope Changes:

- The total scope increased over time as new tasks were added, reaching a maximum of 76 story points at sprint completion.
- Adjustments were made to estimates for several tasks, reflecting the team's understanding of the required effort as work progressed.
- **Issue Completions:** There were some issues. Login permissions weren't being properly applied.
- **Estimate Updates:** Estimates for several issues were updated throughout the sprint to reflect a more accurate assessment of the work required.

Report: SCRUM Sprint 2

\*Issue added after sprint start

Scope changes log

[View in issue navigator](#)

Date	Key	Summary	Issue type	Epic	Details of scope change	Change in estimation
2024-10-11	SCRUM-24	As an <Admin>, I want to <download the...	Story		Issue added to sprint	-
2024-10-11	SCRUM-25	As a <student>, I want to <select from a ...	Story		Issue added to sprint	-
2024-10-11	SCRUM-26	As a <TA> I want to <show I am here>, s...	Story		Issue added to sprint	-
2024-10-11	SCRUM-27	As a <Admin/TA/Student>, I want to <kn...	Story		Issue added to sprint	-
2024-10-11	SCRUM-28	As a <Student/TA>, I want to see <alread...	Story		Issue added to sprint	-
2024-10-11	SCRUM-29	As a <product owner>, I want to <prioriti...	Story		Issue added to sprint	-
2024-10-11	SCRUM-30	As a <scrum master>, I want to <arrange ...	Story		Issue added to sprint	-

- Completion:** By the end of the sprint, all initially planned and newly added tasks were completed, demonstrating effective progress and adaptability throughout the sprint.

## 10. Evaluation of code and test quality Gourangi Sanjay Taware

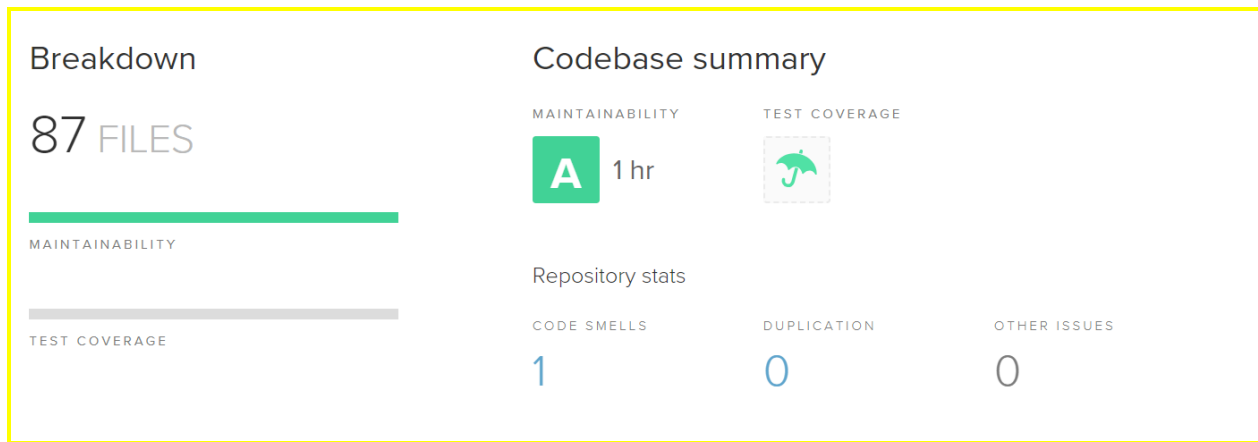
### a. Coverage

- Coverage report generated for Cucumber Features, RSpec to /home/user/office-hours-tracker/coverage.
- Line Coverage: 96.71% (206 / 213) (the data given by rspec)

### b. Code Climate Smells

### c. Code Climate Style

The image below shows the climate report of the current codebase, the maintainability is at level A, and there are no code smells.



## 11. Customer meeting Chen-Hung Wu

Demo date:

2024/10/23 13:45 - 14:00 @Annex 240 (group discussion) + zoom call w/ client

We demoed the Heroku site, and the client agreed that it met her initial expectations. The format of the student attendance data is fine, but she requested that the TA attendance data include names. She also suggested adding charts or tables to better present the data and asked for confirmation that the downloadable data is in CSV format. Additionally, she wants us to ensure the app works smoothly on mobile, as most students will be accessing the system through their phones. We also asked if a TA clock-out feature was necessary, and she confirmed that it was not.

## 12. BDD & TDD Wei Chien Kao

<BDD>

7 scenarios are created/updated in this sprint.

**home\_page.feature**

Feature: Home Page

As a user (student/TA/prof)

I want to see the home page and about page

So that I can get information about the Office Hours Tracker

Scenario: Home Page Elements

Given I am on the page "Home"

Then I should see "Welcome"

And I should see "Login with Google"



## admin\_page.feature

Feature: Admin Page

As a Admin

I want to login and see the Admin page

Scenario: Admin logs in with mocked Google OAuth account

Given I am logged in with name "name\_admin" and email "admin@admin.com"

Given I am on the page "Home"

When I click "Login with Google"

Then I should see "Howdy Admin, name\_admin!"

Scenario: Download student attendance data

Given I am logged in with name "name\_admin" and email "admin@admin.com"

Given I am on the page "Home"

When I click "Login with Google"

And I am on the page "Admin"

When I set the start date to "2024-10-01"

And I set the end date to "2024-10-18"

And I select "Student" from the admin "Attendance Type" dropdown

And I click "Download Attendance CSV"

Then I should receive a CSV file

And the CSV file should contain the correct attendance data

Scenario: Admin logs out

Given I am on the page "Home"

When I click "Login with Google"

When I click link "Logout"

Then I should see "Office Hours Tracker"

## student\_page.feature

Feature: Student Page

As a user (student)

I want to login and see the student page

So that I can register I am here

Scenario: When first logging in, the button should not be visible

Given I am logged in with name "name\_student" and email "student@student.com"

Given I am on the page "Home"  
When I click "Login with Google"  
Then I should not see "CHECK IN FOR"  
Then I should see "No selection" in the "course\_number" dropdown and it is selected

Scenario: Student can select specific courses

Given I am on the page "Home"  
Given I am logged in with name "name\_student" and email "student@student.com"  
When I click "Login with Google"  
Then I select "ENGR 102" from the "course\_number" dropdown  
Then I should see "CHECK IN FOR ENGR 102" on the button with id "mark-attendance-btn"

## ta\_page.feature

Feature: TA Page

As a TA

I want to login and see the ta page

Scenario: TA successfully marks attendance

Given I am logged in with name "Gourangi" and email "gourangitaware@tamu.edu"  
Given I am on the page "Home"  
When I click "Login with Google"  
Then I should see "Howdy TA, Gourangi!"  
When I mark my attendance  
Then I should see "Ta attendance marked successfully!"

<TDD>

14 new tests are created in this sprint, and the high level view is described below.

## users\_controller\_spec.rb

- **#authorize\_access when current user is valid:**
  - **Test:** Allows access.
    - **Purpose:** Ensures the valid users are allowed to login.
- **#authorize\_access when current user is invalid:**
  - **Test:** Redirects to the root path with an alert.
    - **Purpose:** Ensures that invalid users are not allowed to login.
- **#current\_user\_valid? when current user exists and action is showTA:**

- **Test:** Calls `ta_email?` method.
    - **Purpose:** Confirms that the `ta_email?` method is called when the action is `showTA`.
- **#current\_user\_valid? when current user exists and action is showAdmin:**
  - **Test:** Calls `admin_email?` method.
    - **Purpose:** Ensures that the `admin_email?` method is called when the action is `showAdmin`.
- **#current\_user\_valid? when current user exists and action is neither showTA nor showAdmin:**
  - **Test:** Returns true.
    - **Purpose:** Ensures that any other action allows access when the user exists.
- **#current\_user\_valid? when current user is nil:**
  - **Test:** Returns false.
    - **Purpose:** Confirms that when no user is present, the method returns false.
- **#ta\_email? for valid TA email:**
  - **Test:** Returns true for valid TA email.
    - **Purpose:** Ensures that the system correctly recognizes a valid TA email.
- **#ta\_email? for invalid TA email:**
  - **Test:** Returns false for invalid TA email.
    - **Purpose:** Confirms that invalid TA emails are rejected as expected.
- **#admin\_email? for valid admin email:**
  - **Test:** Returns true for valid admin email.
    - **Purpose:** Verifies that the system correctly identifies a valid admin email.
- **#admin\_email? for invalid admin email:**
  - **Test:** Returns false for invalid admin email.
    - **Purpose:** Ensures that invalid admin emails are properly rejected.
- **#ta\_emails:**
  - **Test:** Calls `read_emails_from_csv` with the correct file path.
    - **Purpose:** Ensures that the TA emails are fetched from the correct CSV file.
- **#admin\_emails:**
  - **Test:** Calls `read_emails_from_csv` with the correct file path.
    - **Purpose:** Ensures that admin emails are read from the correct CSV file.
- **#read\_emails\_from\_csv:**

- **Test:** Reads emails from the CSV file.
  - **Purpose:** Ensures that emails are correctly read from the provided CSV file.

ta\_attendance\_spec.rb

### TaAttendance class

- **Test:** Checks that TaAttendance belongs to a user.
  - **Purpose:** Ensures that the TaAttendance model is correctly associated with the user model through a belongs\_to relationship.