

Office Hour Tracker - Sprint 3 Retrospective

2024/11/5

1. Dates of sprint: Mon, Oct 21, 2024 – Fri, Nov 1, 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 an <Admin>, I want to <see attendance graphs>, so that I can understand attendance trends.	Completed	2	Chen-Hung
As a <product owner>, I want to <prioritize the story that should be addressed in this sprint>, so that <the client can use the product after this sprint.>	Completed	2	Gourangi
As an <Admin>, I want to <see student attendance in UI>, so that I can have a record of attendance for analysis in UI.	Completed	2	Ethan
As an <admin> I want to <upload a Admin/TA email file>, so that <I can change the Admins/TA if needed.>	Completed	2	Bryson
As an <admin> I want to <change the available courses>, so that <courses information can be updated each semester>	Completed	2	Franklin

As a <User/TA>, I want to <make the GUI more intuitive>, so that <users can navigate the application easily and have a positive user experience>	Completed	1	Nazaro
As an <Admin>, I want to <see the list of TA checked in>, so that <the admin can use it to process payroll>	Completed	1	Nazaro
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	Nhat

4. Sprint goal:

To enhance the functionality and usability of the application by implementing critical admin features, improving the user interface, and ensuring accurate data management for attendance and course information, allowing the client to effectively utilize the product by the end of this sprint.

5. Sprint Achievement:

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

- Mass role and course updating: the admin can now upload a csv file to create new permissions for emails and course information
- Admins can see attendance trends: added a configurable bar graph that displays student attendance over a window of time for admins to gain insight
- Improved CSV download: the admin can now download csv files containing either student attendances or TA attendances

6. Sprint backlog item and status

1.As an <admin> I want to <upload a Admin/TA email file>, so that <I can change the Admins/TA if needed.>

Point (2) - Bryson

Tasks:

- Allow the admin to add individual TA account

- b. Implement functionality for the admin to upload a file containing TA emails, which will replace the existing TA email file.

Estimated Time: 3 hr

2. As an <admin> I want to <change the available courses>, so that <courses information can be updated each semester>

Point (2) - Franklin

Tasks:

- a. Display the list of courses stored in the Course database on a dedicated subpage in the admin account.
- b. Enable CRUD (Create, Read, Update, Delete) functionality for the course database directly on the subpage. This allows admins to easily add, modify, or remove courses as needed.

Estimated Time: 3 hr

3. As an <Admin>, I want to <see student attendance in UI>, so that I can have a record of attendance for analysis in UI.

Point (2) - Ethan

Tasks:

- a. Display attendance data in the UI.
- b. Add course selection options (all or specific courses) for the admin when viewing the attendance CSV.

Estimated Time: 3 hr

4. As an <Admin>, I want to <see attendance graphs>, so that I can understand attendance trends.

Point (2) - Chen-Hung

Tasks:

- a. Display students' attendance trends on a monthly and weekly basis.
- b. Add course selection options (all or specific courses) for the admin when viewing attendance data

Estimated Time: 3 hr

5. As a <User/TA>, I want to <make the GUI more intuitive>, so that <users can navigate the application easily and have a positive user experience>

Point (1) - Nazaro

Tasks:

- a. Modify the Background image to make the text readable, Modify the UI of the form in the Admin page.
- b. Modify the CSS in the newly added page, so that each page looks consistent.

Estimated Time: 1.5 hr

6. (*) As an <Admin>, I want to <see the list of TA checked in>, so that <the admin can use it to process payroll>

Point (1) - Nazaro

Tasks:

- a. Modify the TA CSV file to contain the following column (name of the TA, check in date and time)

Estimated Time: 1.5 hr

Change/Reason: The person doing this task accidentally makes a list of people appending to specific time slots instead of showing people's check-in time. Should be rectified in the next sprint.

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

Point (2) - Gourangi

Tasks:

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

Estimated Time: 3 hrs

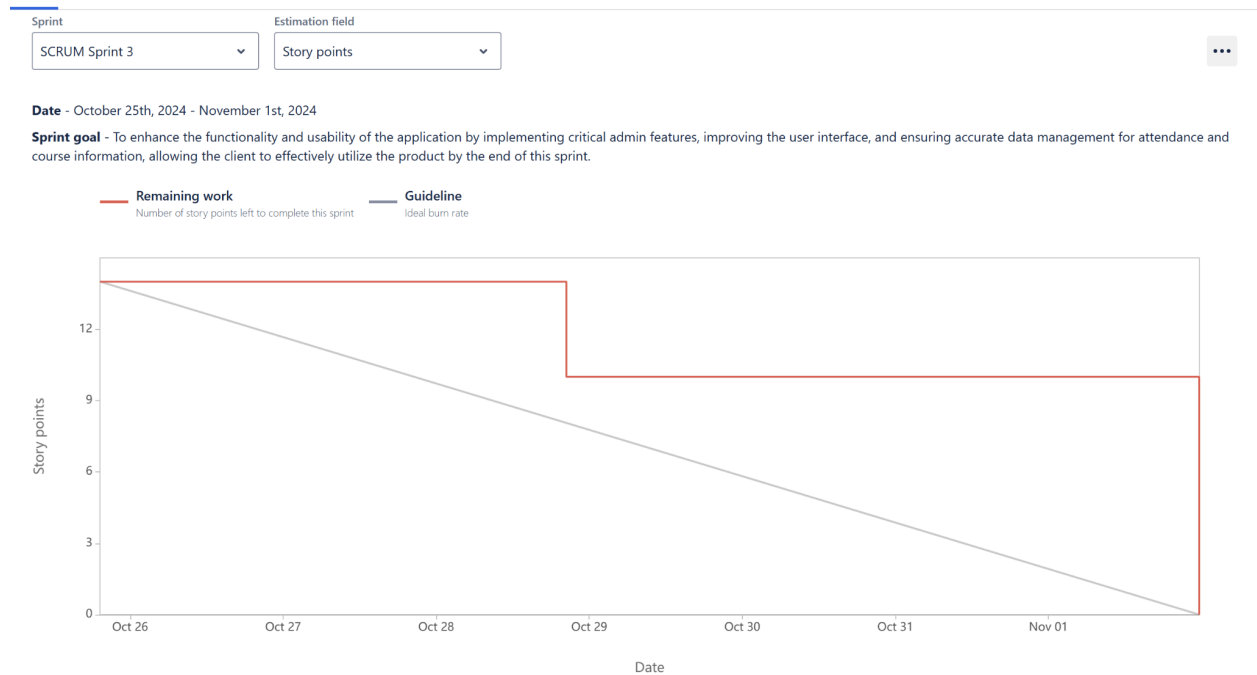
8) As a <scrum master>, I want to <arrange daily meetings and remove difficulties encountered by teammates>, so that <the developers will have reasonable progress during the sprint>

Point (2) - Nhat

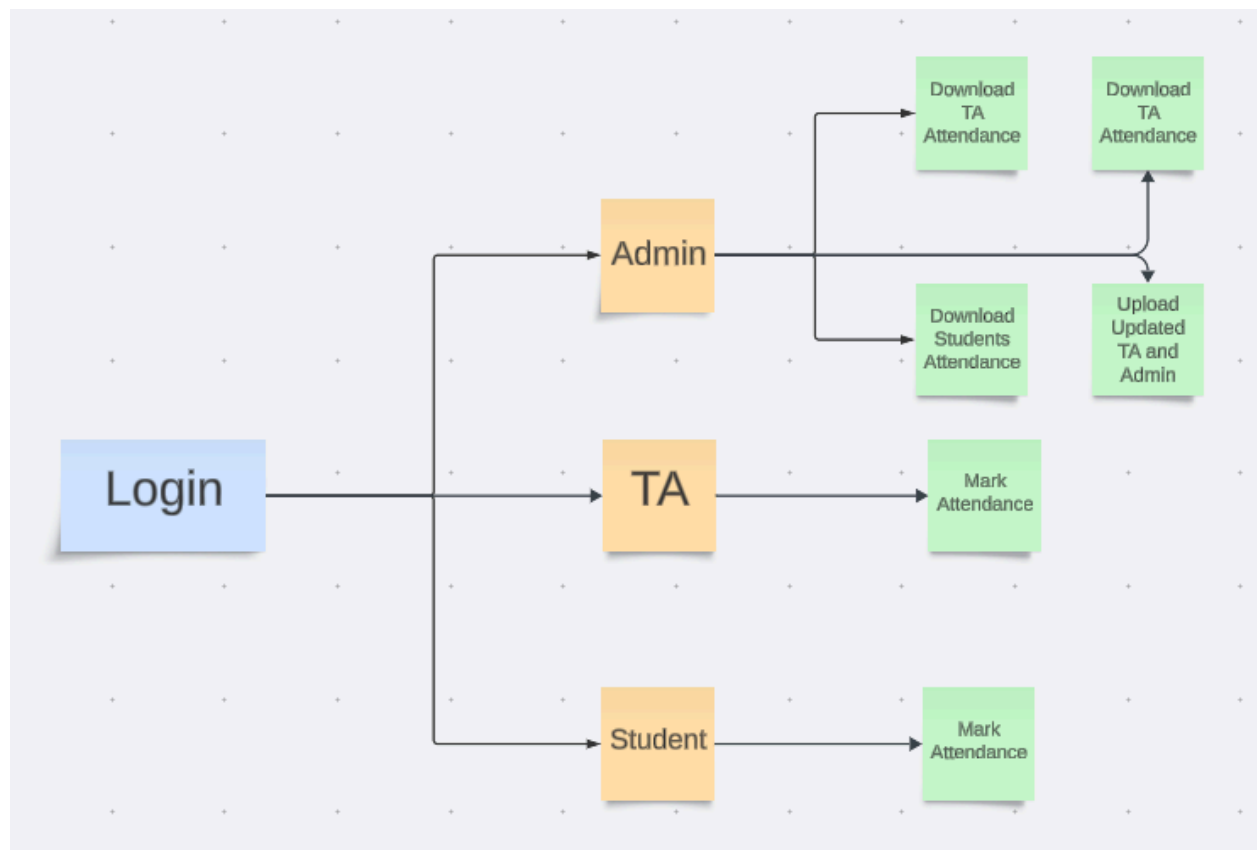
Estimated Time: 3hrs

- a. Hold regular stand-up meetings to learn about people's progress and to remove obstacles in the development process
- b. Write cucumber tests to ensure that the functions are working properly.
- c. Arrange weekly meetings with the TA

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

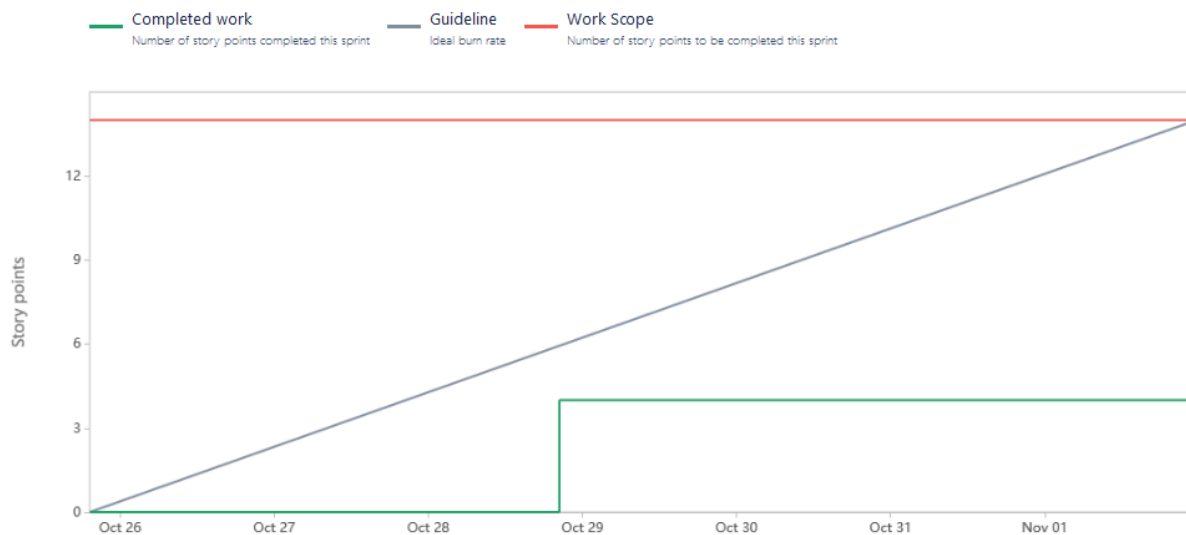


8. Design Diagram for the sprint



9. Documentations of changes:

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



Sprint Duration: Mon, Oct 25, 2024 – Fri, Nov 1, 2024

- Sprint Initiation:** The sprint began with the addition of all tasks:
 - Allow the admin to add TA accounts & upload classes
 - Create a graph for the admin to view the trend of items
 - Make the CSS of the text easier to read
 - Update the download CSV with an option to download TA CSV data
- Progress Updates:**
 - As issues were completed, the total completed story points increased from 52 (sprint 2) to 66 by the end of the sprint.
- Scope Changes:**
 - The total scope increased over time as new tasks were added, reaching a maximum of 66 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:** Multiple issues, including Rspec tests, Cucumber tests, download function, GUI styling, and user database creation, were completed during the sprint.
 - **Estimate Updates:** Estimates for several issues were updated throughout the sprint to reflect a more accurate assessment of the work required.

Mon, Oct 28 2024, 8:25pm	Issue completed	SCRUM-31 As an <admin> I want to <upload a Admin/TA email file>, so that <I can change the Admins/TA if needed.>	0 → 2	14
Mon, Oct 28 2024, 8:25pm	Issue completed	SCRUM-32 As an <admin> I want to <change the available courses>, so that <courses information can be updated each semester>	2 → 4	14
Fri, Nov 01 2024, 11:39pm	Issue completed	SCRUM-33 As an <admin> I want to <view> the attendance history table straight on the app so that I do not have to download a new CSV file every day.	4 → 6	14
Fri, Nov 01 2024, 11:39pm	Issue completed	SCRUM-34 As an admin, I want to view some graphs showing trends in attendance.	6 → 8	14
Fri, Nov 01 2024, 11:39pm	Issue completed	SCRUM-35 As a <User/TA>, I want to <make the GUI more intuitive>, so that <users can navigate the application easily and have a positive user experience>	8 → 9	14
Fri, Nov 01 2024, 11:39pm	Issue completed	SCRUM-36 As an <Admin>, I want to <see the list of TA checked in>, so that <the admin can use it to process payroll>	9 → 10	14
Fri, Nov 01 2024, 11:39pm	Issue completed	SCRUM-37 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.>	10 → 12	14
Fri, Nov 01 2024, 11:39pm	Issue completed	SCRUM-38 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>	12 → 14	14

4. **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

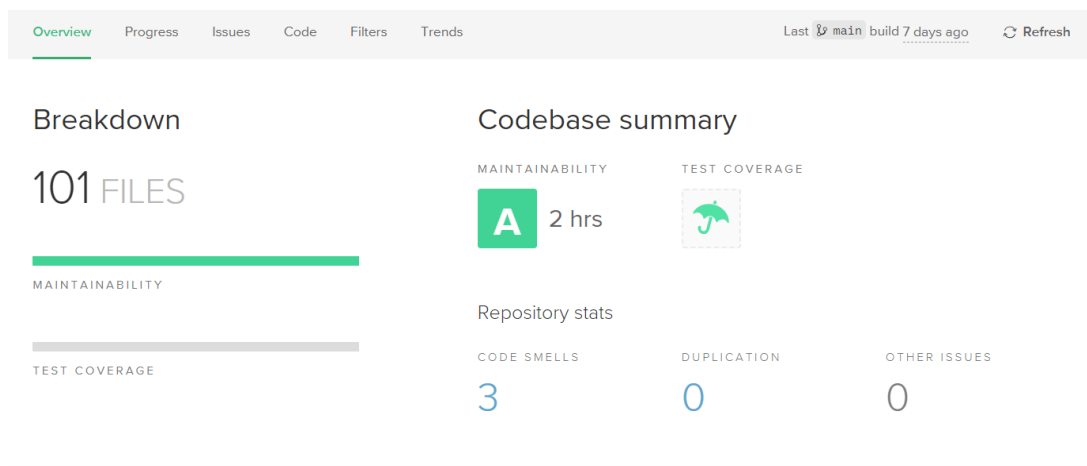
a. Coverage

- i. Coverage report generated for Cucumber Features, RSpec to /home/user/office-hours-tracker/coverage.
- ii. Line Coverage: 92.62% (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.



11. Customer meeting

Demo date:

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

We demonstrated the Heroku site to the client, who was pleased with the UI and the recent feature additions. The client provided feedback specifically on the TA CSV download functionality, suggesting improvements to include TA names in the CSV output. Based on this input, we'll prioritize completing the TA CSV download feature and ensure TA names are incorporated as requested.

Additionally, we'll address any existing bugs and handle edge cases in preparation for the upcoming sprint. The client has approved our plan moving forward and is aligned with the approach for testing functionality enhancements in the next sprint.

2024/11/8 via email: We have one more meeting set up with the client for complete end to end testing of the application.

12. BDD & TDD

<BDD>

Feature: Admin Page

- **As an Admin**
 - I want to log in and view the Admin page

Scenario 1: Admin logs in with mocked Google OAuth account

- Given the following courses exist:

| course_number | course_name | instructor_name | start_date | end_date |

| ENGR 102 | Engineering Lab I - Computation | Niki Ritchey | 2024-08-19 | 2024-12-31 | ENGR 200 | Dummy Course 1 | Dummy Professor | 2024-08-19 | 2024-12-31 |

- Given I am logged in with:
 - Name: "name_admin"
 - Email: "admin@admin.com"
- And I am on the page "Home"

- When I click "Login with Google"
- Then I should see "Howdy, name_admin!"
- When I am on the page "Admin"
 - And I set:
 - Start date: "2024-10-01"
 - End date: "2024-10-18"
 - And I select "Student" from the "Attendance Type" dropdown
 - And I click "Download Data"
- Then I should receive a CSV file
 - And the CSV file should contain the correct attendance data

Scenario 2: Admin logs out

- Given the following courses exist:

| course_number | course_name | instructor_name | start_date | end_date

| ENGR 102 | Engineering Lab I - Computation | Niki Ritchey | 2024-08-19 | 2024-12-31 | ENGR
200 | Dummy Course 1 | Dummy Professor | 2024-08-19 | 2024-12-31 |

- Given I am logged in with:
 - Name: "name_admin"
 - Email: "admin@admin.com"
- And I am on the page "Home"
- When I click "Login with Google"
- And I click the link "Logout"
- Then I should see "Office Hours Tracker"

Scenario 3: Attendance form submits correctly and updates histogram

- **Given** the following courses exist:

| course_number | course_name | instructor_name | start_date | end_date |

| ENGR 102 | Engineering Lab I - Computation | Niki Ritchey | 2024-08-19 | 2024-12-31

- **Given** I am logged in with:
 - **Name:** "name_admin"
 - **Email:** "admin@admin.com"
- **And** I am on the page "Home"
- **And** I click "Login with Google"
- **When** I select "ENGR 102" from the "course_id" dropdown within the ".centered-table" section
- **Then** the histogram should be updated with the correct attendance data

<TDD> :

GET #upload_email

- **Test:** Returns a successful response.
- **Purpose:** Verifies that the admin can access the upload email page and the view renders correctly.
- **Test:** Assigns `@current_user`.
- **Purpose:** Ensures that the correct user is assigned upon accessing the upload email page.

#valid_email? method

- **Test:** Returns `true` for valid email addresses.
- **Purpose:** Confirms that the `valid_email?` method correctly identifies valid email formats.
- **Test:** Returns `false` for invalid email addresses.
- **Purpose:** Validates that invalid email formats are correctly flagged as invalid.

POST #upload_emails (when valid email provided)

- **Test:** Sets a success flash message without modifying the actual file.
- **Purpose:** Ensures that an email is successfully added to admin emails and that success feedback is shown.

POST #upload_emails (when invalid email provided)

- **Test:** Does not append the email and sets an alert flash message.
- **Purpose:** Ensures that invalid email formats are rejected, and appropriate feedback is displayed.

POST #upload_emails (when valid CSV file uploaded)

- **Test:** Overwrites emails in the target file and sets a success flash message.
- **Purpose:** Confirms that valid CSV uploads successfully replace the target file's contents and provide success feedback.

POST #upload_emails (when invalid file format uploaded)

- **Test:** Does not overwrite emails and sets an alert flash message.
- **Purpose:** Ensures only CSV files are accepted and invalid formats trigger an alert.

POST #upload_emails (when no email or file provided)

- **Test:** Sets an alert flash message.
- **Purpose:** Ensures that either an email or file is required for processing, with feedback for missing data.

#overwrite_emails method

- **Test:** Overwrites the target file with emails from the provided CSV file.
- **Purpose:** Confirms that `overwrite_emails` correctly updates the file with new emails only, ensuring accuracy in file updates.