Office Hour Tracker - Sprint 1 Retrospective

2024/10/11

1. Dates of sprint: Mon, Sep 23, 2024 – Fri, Oct 04, 2024

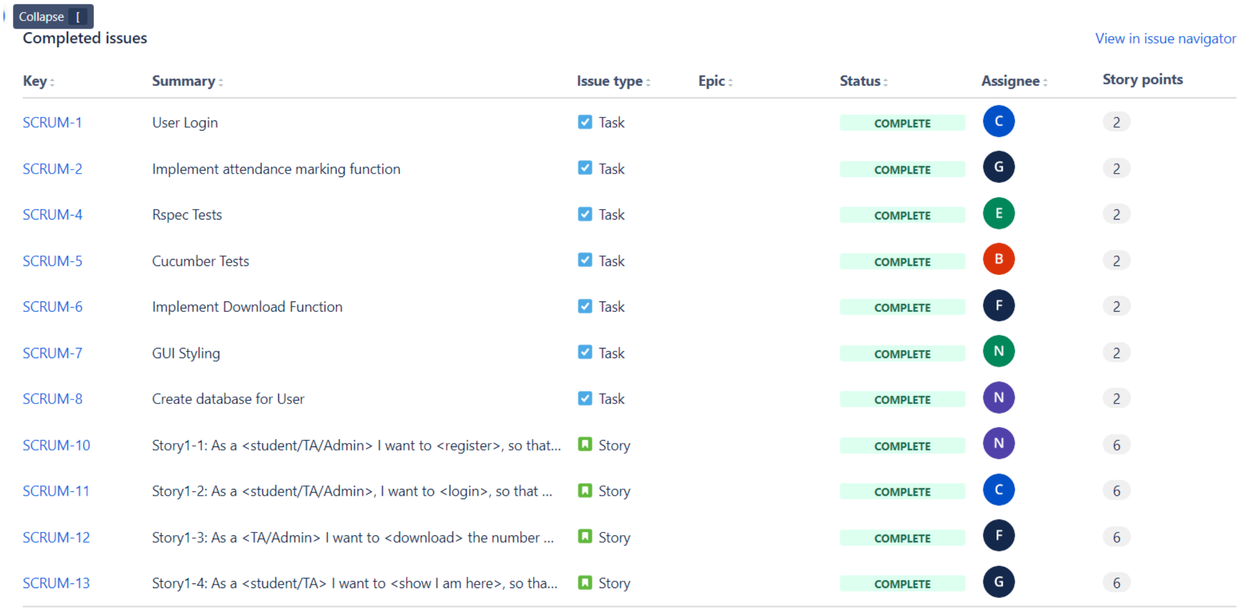
2. Links:
   a. Project management tool Page: https://tamu-team-office-tracker.atlassian.net/jira/software/projects/SCRUM/boards/1
   b. Github Repo: https://tamu-team-office-tracker.atlassian.net/jira/software/projects/SCRUM/boards/1
   c. Heroku deployment: https://office-hours-tracker-a63f1f6d64ad.herokuapp.com/

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):

Issue ID	Issue Title	Status	Story Points	Assignee
SCRUM-1	User Login	Completed	2	Chen
SCRUM-2	Implement attendance marking function	Completed	2	Gourangi
SCRUM-4	Rspec Tests	Completed	2	Ethan
SCRUM-5	Cucumber Tests	Completed	2	Bryson
SCRUM-6	Implement Download Function	Completed	2	Franklin
SCRUM-7	GUI Styling	Completed	2	Nazaro

SCRUM-8	Create database for User	Completed	2	Nhat

All user stories and tasks for Sprint 1.



Collapse [

Completed issues

View in issue navigator

Key	Summary	Issue type	Epic	Status	Assignee	Story points
SCRUM-1	User Login	☑ Task		COMPLETE	C	2
SCRUM-2	Implement attendance marking function	☑ Task		COMPLETE	G	2
SCRUM-4	Rspec Tests	☑ Task		COMPLETE	E	2
SCRUM-5	Cucumber Tests	☑ Task		COMPLETE	B	2
SCRUM-6	Implement Download Function	☑ Task		COMPLETE	F	2
SCRUM-7	GUI Styling	☑ Task		COMPLETE	N	2
SCRUM-8	Create database for User	☑ Task		COMPLETE	N	2
SCRUM-10	Story1-1: As a <student/TA/Admin> I want to <register>, so that...	Story		COMPLETE	N	6
SCRUM-11	Story1-2: As a <student/TA/Admin>, I want to <login>, so that ...	Story		COMPLETE	C	6
SCRUM-12	Story1-3: As a <TA/Admin> I want to <download> the number ...	Story		COMPLETE	F	6
SCRUM-13	Story1-4: As a <student/TA> I want to <show I am here>, so tha...	Story		COMPLETE	G	6

4. Sprint goal:

"Enable secure user login via Google OAuth2, implement session management, and deliver key office hour tracking functionalities, including attendance marking and CSV export, to streamline the office hour logging process."

5. Sprint Achievement:
During this sprint, the following user stories were successfully implemented:

- **Google OAuth2 Authentication**: Configured Google OAuth2 to allow secure login for users via their Google accounts.
- **Session Management**: Implemented session handling, enabling users to log in and out, and ensuring session persistence and security.
- **Attendance Marking**: Developed the functionality for PTs to mark attendance during office hours, ensuring accurate logging of student visits.
- **CSV Export Feature**: Provided the ability to export the logged attendance data to a CSV file, allowing for easy data analysis and reporting.

6. Sprint backlog item and status

(Indicate backlog items added after the sprint has been started with + in the list. Indicate backlog items modified during the sprint with * in the list. Write a short description of the changes made and the reason behind the changes. Indicate backlog items that were removed from the sprint (i.e. you were not able to implement all the stories that you initially chose for this iteration) with - in the list. Write a short explanation of why they were unable to be completed.)

Below shows the story in the sprint 1 plan. We added new backlog items (indicated by “+”) and revise a story (indicated by “*”) to reflect the actual effort done by everyone.

1) (*) As a <student/TA/Admin> I want to <register>, so that <system knows who I am>.

Changes/Reasons: We merge this task with the login, for register and login means the same in OAuth

- a) Assigned students: Chen
- b) Tasks:
  - i) Incorporate third-party software to be able to register a new user

2) As a <student/TA/Admin>, I want to <login>, so that <I can enter the system>.

- a) Assigned students: Chen
- b) Tasks:
  - i) Incorporate third-party software to be able to log in.

3) As a <TA/Admin> I want to <download> the number of students per hour so that <we can use the records for scheduling purposes>.

- a) Assigned students: Wei
- b) Tasks:
  - i) Create a database to be able to store the number of students who clicked “I am here” for each time shift.
  - ii) Export that information upon request to an Excel spreadsheet

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

- a) Assigned students: Gourangi
- b) Tasks:

i) Create the student home page

ii) Create a button on the student home page that increments the time slot the student is currently in inside of the database whenever they click “I am here”

5) (+) As a <student>, I want <each page to have a well-designed and intuitive UI>, so that <I can easily understand the purpose and content of each page>.

Changes/Reason: We make the UI design a separate task to ensure that the style between each page is consistent.

a) Assigned Students: Nazar
b) Tasks:
   i) Design UI for login and student page.

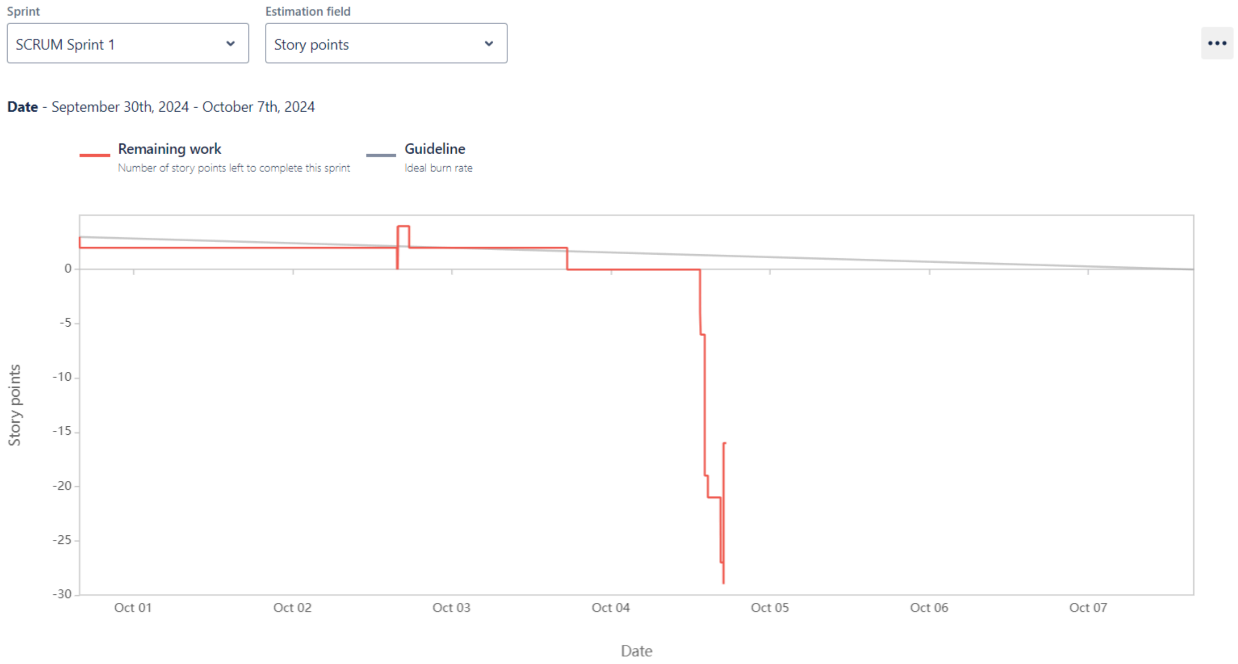
6. (+) As a <student>, I want the < the navigation between pages to be seamless>, so that <I have a smooth and enjoyable user experience>.

Changes/Reason: Our team has designated the scrum master and product owner to primarily handle documentation and testing. To support their efforts, we have assigned a dedicated person to assist with testing to ensure quality.

a) Assigned Students: Nhat, Ethan, Bryson
b) Tasks:
   i) Set up Cucumber tests for end-to-end navigation flows
   ii) Implement RSpec tests to validate individual page components and functionality
   iii) Integrate Code Climate to monitor code quality and maintainability

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

we discovered that the tasks on Jira were organized as individual tasks rather than as user stories. To align with a story-based approach, we restructured the presentation, to group tasks under relevant user stories. This shift caused an unusual pattern in the burndown chart.

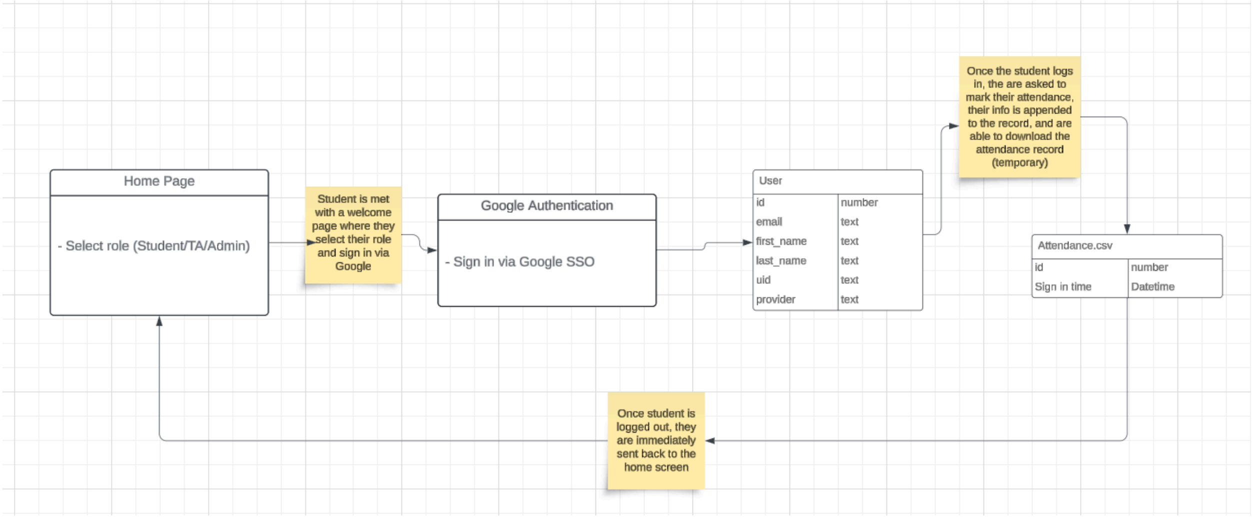


Sprint: SCRUM Sprint 1

Estimation field: Story points

Date - September 30th, 2024 - October 7th, 2024

8. Design Diagram for the sprint



Student is met with a welcome page where they select their role and sign in via Google

User

id	number
email	text
first_name	text
last_name	text
uid	text
provider	text

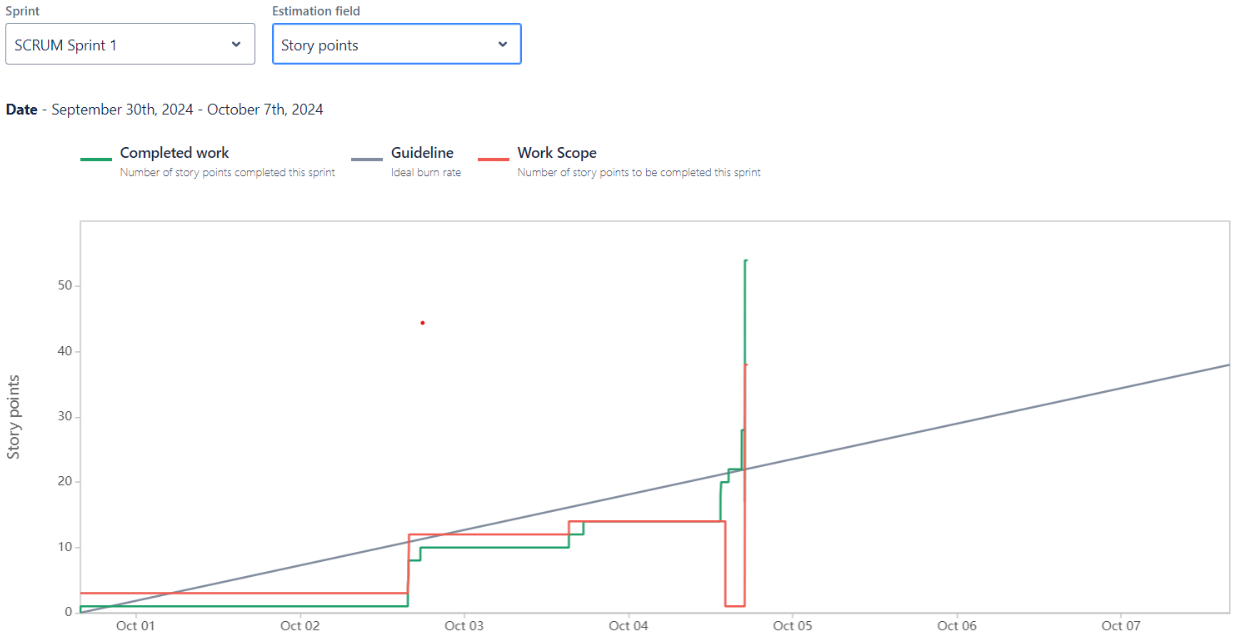
Once the student logs in, the are asked to mark their attendance, their info is appended to the record, and are able to download the attendance record (temporary)

Attendance.csv

id	number
Sign in time	Datetime

Once student is logged out, they are immediately sent back to the home screen

9. Documentations of changes:
Following is the burnup report of the story points throughout the sprint:



Sprint
SCRUM Sprint 1

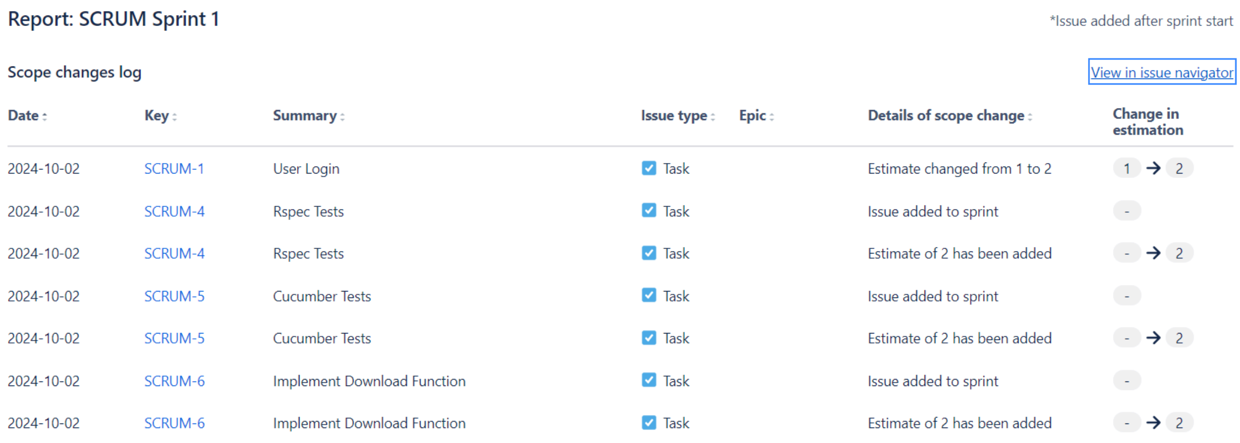
Estimation field
Story points

Date - September 30th, 2024 - October 7th, 2024

Sprint Duration: Mon, Sep 23, 2024 – Fri, Oct 04, 2024

1. **Sprint Initiation:** The sprint began with the addition of the following tasks:
   - User Login
   - Attendance Marking Function
   - GUI Styline
   - Download Attendance Functionality
2. **Progress Updates:**
   - As issues were completed, the total completed story points increased from 0(initiation) to 54 by the end of the sprint.
   - Several tasks were added and their estimates were updated throughout the sprint, including:
     - Rspec Tests
     - Cucumber Tests
     - Database for User
3. **Scope Changes:**
   - The total scope increased over time as new tasks were added, reaching a maximum of 54 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 user login, attendance marking function, 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.



Report: SCRUM Sprint 1

*Issue added after sprint start

Scope changes log

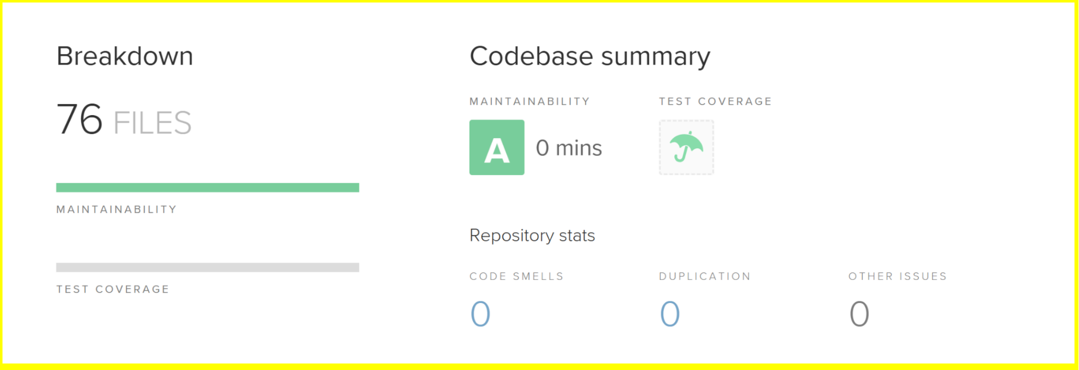
View in issue navigator

Date	Key	Summary	Issue type	Epic	Details of scope change	Change in estimation
2024-10-02	SCRUM-1	User Login	☑ Task		Estimate changed from 1 to 2	1 → 2
2024-10-02	SCRUM-4	Rspec Tests	☑ Task		Issue added to sprint	-
2024-10-02	SCRUM-4	Rspec Tests	☑ Task		Estimate of 2 has been added	- → 2
2024-10-02	SCRUM-5	Cucumber Tests	☑ Task		Issue added to sprint	-
2024-10-02	SCRUM-5	Cucumber Tests	☑ Task		Estimate of 2 has been added	- → 2
2024-10-02	SCRUM-6	Implement Download Function	☑ Task		Issue added to sprint	-
2024-10-02	SCRUM-6	Implement Download Function	☑ Task		Estimate of 2 has been added	- → 2

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: 100.0% (77 / 77)
    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.



Breakdown

76 FILES

MAINTAINABILITY

TEST COVERAGE

Codebase summary

MAINTAINABILITY	TEST COVERAGE
A 0 mins	

Repository stats

CODE SMELLS	DUPLICATION	OTHER ISSUES
0	0	0

11. Customer meeting

Demo date:
2024/10/9 13:05 - 14:30 @Annex 245 (group discussion) + zoom call w/ client

a. We demoed the heroku site and the client loved the frontend UI so far. We then clarified the full scope of the project regarding the classes that will be using it. Based on this, we will be continuing on with our admin pages and TA pages as we planned, but we will modify the student page to select the specific class the student is there to get help for. We also planned on giving the client a version of the application to be able to test at the end of sprint 2, with the improvements to the functionality being added modularly in sprints 3 and 4. The client has given an OK for approval.
b. Client did not suggest any improvements.
c. For internal improvement, we want to fix internal bugs and add in the next set of features. New features will include code handling for admins/TAs and the expanded database to track the check-in button.

12. BDD & TDD

<BDD>

```
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'm on the page "Home"
        Then I should see "Welcome"
        And I should see "Login with Google"


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: Student logs in successfully
  Given I'm on the page "Home"
  When I click "Login with Google"
  Then I should see "Howdy, Test!"
  When I click "I AM HERE!"
  Then I should see "Attendance marked successfully!"

Scenario: Student logs in successfully
  Given I'm on the page "Home"
  When I click "Login with Google"
  When I click the download link
  Then the download link should point to the correct file path

Scenario: Student logs out
  Given I'm on the page "Home"
  When I click "Login with Google"
  When I click link "Logout"
  Then I should see "Office Hours Tracker"
```

<TDD>
There are five Rspec files in this sprint, and the tests in each file are described below.

`welcome_controller_spec.rb`

- GET #index when user is logged in:
    - Test: Redirects to the user's show page with a welcome notice.
        - Purpose: Simulates a logged-in user visiting the index page and expects it to redirect to the user's show page, with a flash notice saying welcome back.
- GET #index when user is not logged in:
    - Test: Renders the index view.
        - Purpose: Ensures that when no user is logged in, only the index page is rendered.

`users_controller_spec.rb`

- GET #show when user exists:
    - Test: Assigns the requested user to `@current_user`.

        - Purpose: Verifies that the correct user is assigned to `@current_user` when the user exists.
    - Test: Renders the show template.
        - Purpose: Ensures the `show` view can be seen when the user exists.
- GET #show when user does not exist:
    - Test: Raises an `ActiveRecord::RecordNotFound` error.
        - Purpose: Confirms that an error is shown when trying to show a non-existent user.

`sessions_controller_spec.rb`

- GET /auth/google_oauth2/callback on successful login:
    - Test: Logs the user in and sets the session, redirecting to the user path.
        - Purpose: Ensures a successful Google login saves the user's session and redirects to the user's show page.
- GET /auth/google_oauth2/callback on failed login:
    - Test: does not log the user in and redirects to the welcome page with an error message
        - Purpose: Verifies that a failed login prevents setting the session and redirects with an alert message.
- GET /logout:
    - Test: logs the user out and redirects to the welcome page
        - Purpose: Confirms that logging out clears the session and redirects to the welcome page.

`attendances_controller_spec.rb`

- GET #export_csv:
    - Test: Sends the attendance report as a CSV file.
        - Purpose: Checks that whether a person can download the csv data correctly
- POST #mark when user exists:
    - Test: marks the attendance and redirects with a notice
        - Purpose: Ensures that the attendance record is created when the user tries to mark attendance and that the flash notice confirms success.
- POST #mark when user does not exist:
    - Test: Does not mark attendance and redirects with an alert
        - Purpose: Verifies that attempting to mark attendance for a non-existent user fails and redirects with an alert.

`application_controller_spec.rb`

- GET #index when user is not logged in:
  - Test: Redirects to the welcome page with an alert message.
    - Purpose: Tests that the `require_login` filter redirects unauthenticated users to the welcome page and displays an alert.
- GET #index when user is logged in:
  - Test: Confirms `current_user` is set correctly.
    - Purpose: Ensures the `current_user` method returns the correct user when logged in.
  - Test: Checks if `logged_in`? returns `true`.
    - Purpose: Ensures that the `logged_in`? method correctly identifies a logged-in user.