Sprint 2 Retrospective

Contents

Team Roles

Product Owner: Abel Gizaw
Scrum Master: Wahib Sabir Kapdi

Dates of Sprint

7th October 2024 - 20th October 2024

Information about Team and Contributions

Team Member	Points	Role	Contribution
Abel Gizaw	1	Product Owner	Abel handled all the client meetings, conveyed the requirements to the team, and provided the acceptance criteria for all the stories.
Wahib Sabir Kapdi	3	Scrum Master	Wahib coordinated with the team and the product owner and ensured the developers were clear on the evolving requirements. He also took care of documenting the efforts and conducting sprint ceremonies. He worked on displaying the room details and set up the header navigation bar.
Navya Unnikrishnan	3	Developer	Navya worked on setting up the schedules, which included creating/deleting new schedules as per the client's requirements. She also followed the TDD and BDD approach to develop the feature.
Pavithra Gopalakrishnan	2	Developer	Pavithra worked on setting up the time slots view, which included displaying the available time slots for classes in a particular schedule. She also ensured all the test cases passed.
Yuqi Fan	2	Developer	Yuqi worked on trying out various algorithms for the constraint optimization algorithm. He did PoCs on them and collaborated with Colby to identify the optimal algorithm for our use case.
Navya Priya Nandimandalam	2	Developer	Navya developed the Instructor view for schedules, displaying instructor details and some of their class preferences. She

			also made sure to cover all test cases and meet the acceptance criteria.
Colby Endres	2	Developer	Colby worked on trying out different algorithms for the constraint optimization algorithm and collaborated with Yuqi to identify the optimal algorithm for our use case. He also tested the optimal algorithm with data similar to our actual data and ensured it covered most of the scenarios.

Sprint Goal

Our sprint goal was to develop key views (schedules, rooms, instructors, time slots) and populate tables with real data provided by the client. This data serves as the foundation for running our constraint optimization algorithm. With this in mind, we focused on carefully parsing the input and populating the tables according to our database design. Simultaneously, we refined the optimization algorithm to ensure it aligns perfectly with our use case and conducted dry runs using test data that closely mirrors the actual data provided by the client.

Sprint Achievements

- The navigation header bar was developed
- Room view was implemented with filters and sorting capabilities, displaying the availability of rooms for teaching courses.
- Instructor View was implemented, displaying some of the teaching preferences
- A time slot view was implemented, displaying the time slots available to teach each of the courses along with the days.
- The feature was designed to enable the creation and deletion of schedules. Schedules serve as the framework around which rooms, time slots, and instructors are assigned.
- PoCs on constraint optimization algorithms were conducted on test data. We finalized one of the algorithms that is optimal for our data.

Sprint Backlog Items and Status

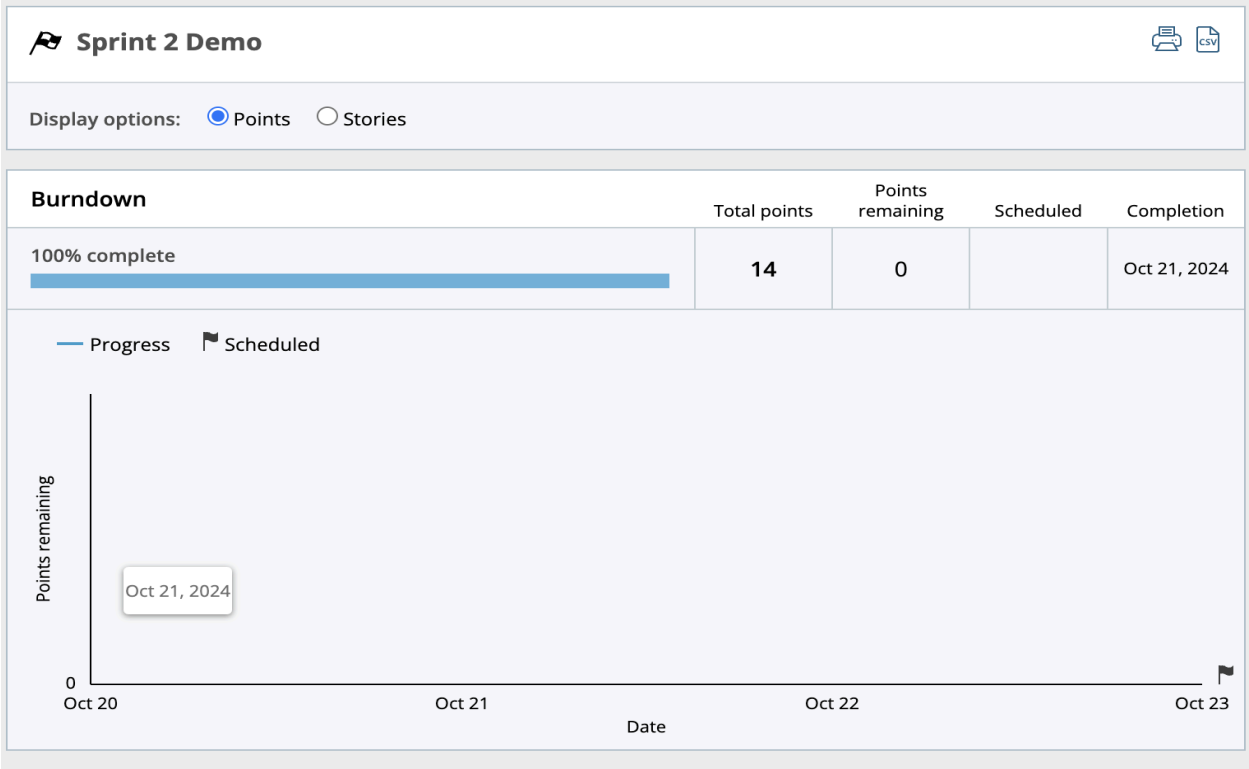
Change	Story/Bug	Status
+	User should be able to see the Header Bar and Logout form	Completed

+	User should be able to upload data as csv file	Completed
+	Link Github and Pivotal Tracker	Completed
+	User should be able to see the Time Slot View	Completed
+	User should be able to see the Room View	Completed
+	User sees the landing page with all the generated schedules	Completed
+	Research Constraint Optimization Algorithm	Completed
+	User should be able to see the instructors View	Completed
-	User should be able to see the courses View	New- Rolled over

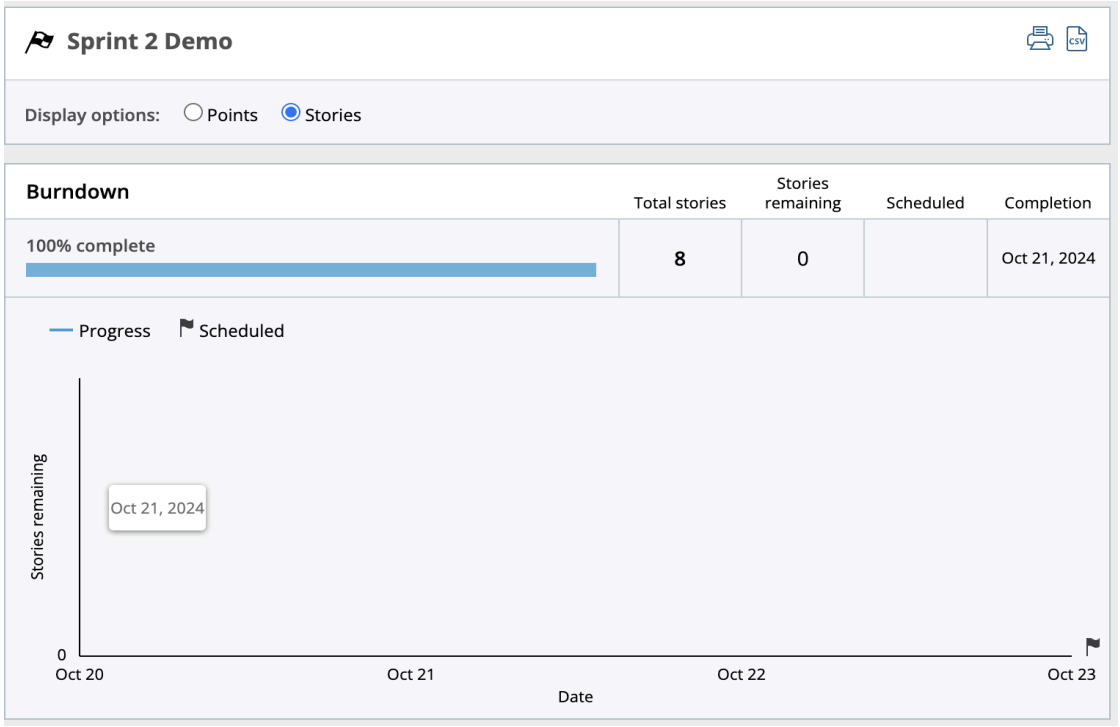
One of the stories (‘User should be able to see the courses View’) could not be completed due to the unavailability of data.

Burndown and Burnup Chart

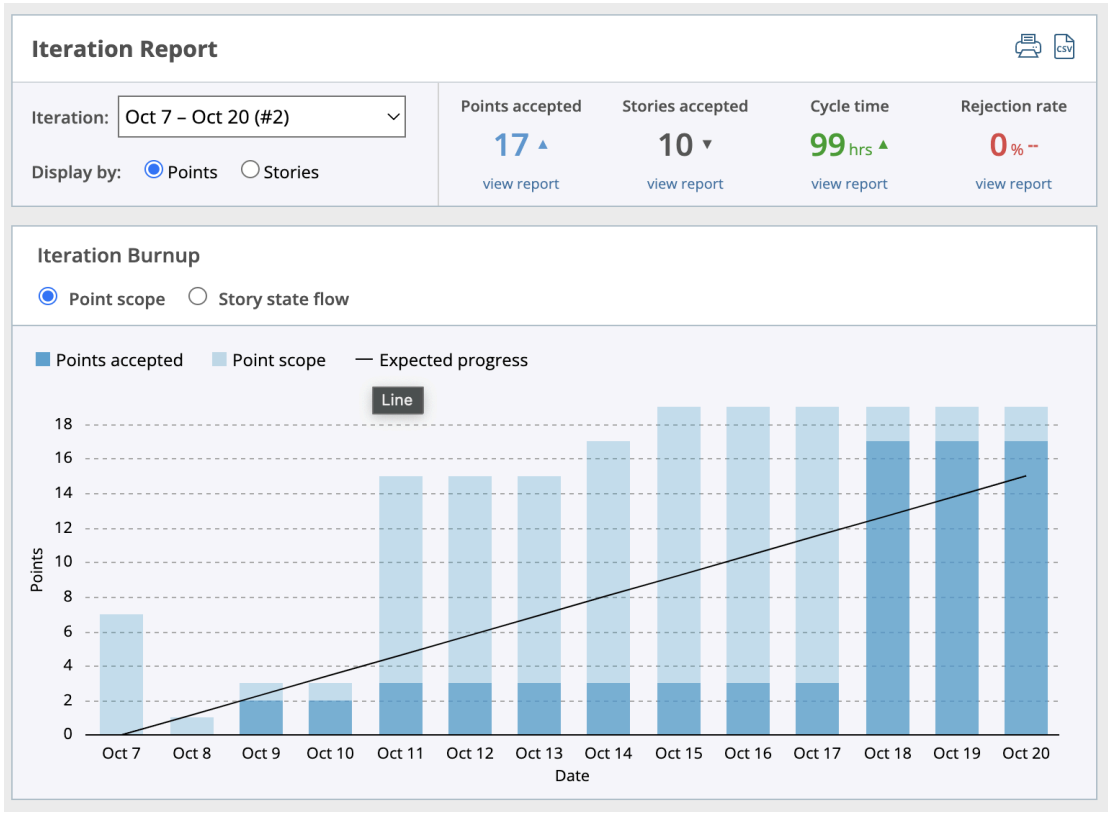
Point-based burndown chart for this sprint



Story based burndown chart for this sprint



Sprint burnup chart



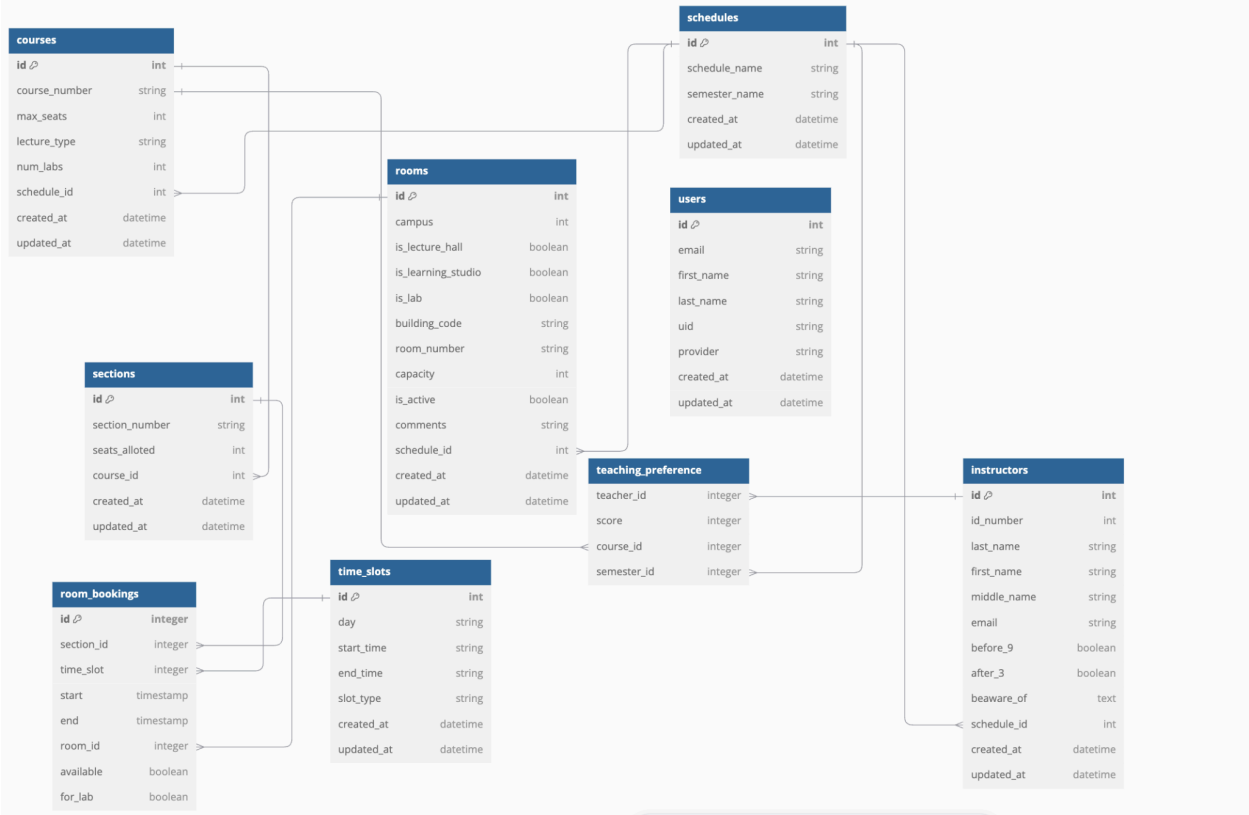
Design Diagram

UI Design

- No changes were made
- https://www.figma.com/design/CRRqUd8c0q8BnKWkPlY1pS/AggieAssign?node-id=1-2377&node-type=frame

DB Design

- Based on the actual data provided by the client, the contents of the tables were updated



Documentation of changes

DB Design

- Based on the actual data provided by the client, the table structure was updated.

Mock-ups and storyboards

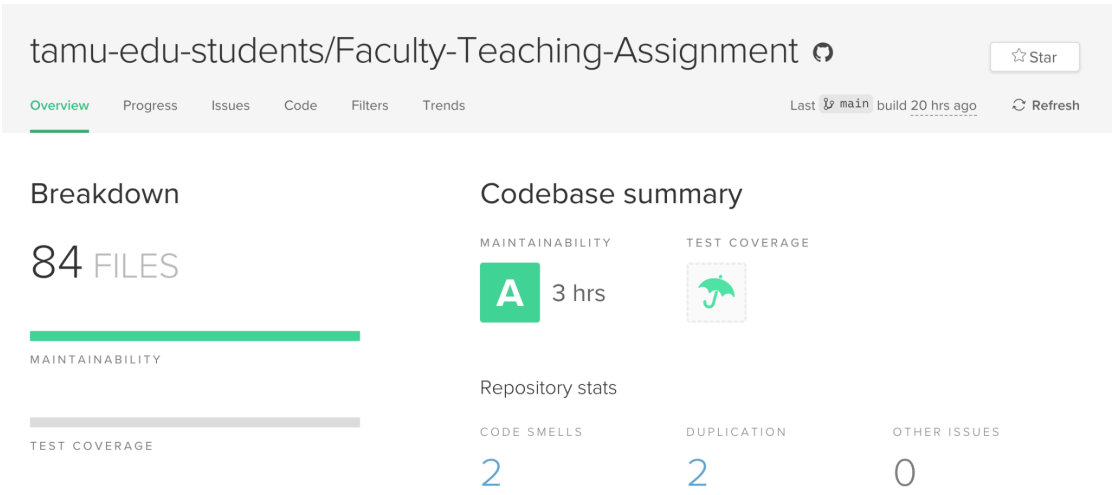
- There was no update.

Team working agreement

- Updated with respect to deleting stale branches and writing test cases for corner cases.

Evaluation of code and test quality

Code climate



Rubocop

```
96 files inspected, 79 offenses detected, 57 offenses autocorrectable

Tip: Based on detected gems, the following RuboCop extension libraries might be helpful:
  * rubocop-capybara (https://rubygems.org/gems/rubocop-capybara)
  * rubocop-factory_bot (https://rubygems.org/gems/rubocop-factory_bot)
  * rubocop-rspec (https://rubygems.org/gems/rubocop-rspec)
  * rubocop-rspec_rails (https://rubygems.org/gems/rubocop-rspec_rails)

The following RuboCop extension libraries are installed but not loaded in config:
  * rubocop-rails

You can opt out of this message by adding the following to your config (see https://docs.rubocop.org/rubocop/extensions.html#extensio
n-suggestions for more options):
  AllCops:
    SuggestExtensions: false
```

Coverage

Unit Tests (rspec)

```
navyan@Navyas-MacBook-Pro-2 Faculty-Teaching-Assignment-Sprint2 % bundle exec rspec
session[:user_id] = 1
........................................................

Finished in 0.72223 seconds (files took 5.76 seconds to load)
56 examples, 0 failures

Coverage report generated for RSpec to /Users/navyan/Downloads/Faculty-Teaching-Assignment-Sprint2/coverage.
Line Coverage: 95.02% (210 / 221)
```

Integration Tests (cucumber)

```
19 scenarios (19 passed)
107 steps (107 passed)
0m0.713s

Share your Cucumber Report with your team at https://reports.cucumber.io

Command line option:    --publish
Environment variable:   CUCUMBER_PUBLISH_ENABLED=true
cucumber.yml:           default: --publish

More information at https://cucumber.io/docs/cucumber/environment-variables/

To disable this message, specify CUCUMBER_PUBLISH_QUIET=true or use the
--publish-quiet option. You can also add this to your cucumber.yml:
default: --publish-quiet

Coverage report generated for Cucumber Features, RSpec to /Users/navyan/Downloads/Faculty-Teaching-Assignment-Sprint2/coverage.
Line Coverage: 98.17% (215 / 219)
```

Coverage of almost 90% was achieved in Sprint 2.

Tests Covered as part of Sprint 2

Cucumber test cases:

```
# features/csv_upload.feature
Feature: CSV Upload
  As a user
  I want to upload a CSV file
  So that I can process data from the file

  Background:                       # features/csv_upload.feature:7
    Given I am logged in as a user # features/step_definitions/OAuth_Authorization_steps.rb:34

  Scenario: Successfully upload a valid CSV file        # features/csv_upload.feature:10
    Given I am on my profile page                       # features/step_definitions/csv_upload_steps.rb:5
    When I attach a valid CSV file to the upload form   # features/step_definitions/csv_upload_steps.rb:9
    And I click the "Submit" button                     # features/step_definitions/common_steps.rb:3
    Then I should see "CSV file uploaded successfully." # features/step_definitions/common_steps.rb:7

  Scenario: Upload an invalid CSV file                   # features/csv_upload.feature:16
    Given I am on my profile page                        # features/step_definitions/csv_upload_steps.rb:5
    When I attach an invalid CSV file to the upload form # features/step_definitions/csv_upload_steps.rb:13
    And I click the "Submit" button                      # features/step_definitions/common_steps.rb:3
    Then I should see "Cannot parse CSV file"            # features/step_definitions/common_steps.rb:7

  Scenario: Upload without selecting a file          # features/csv_upload.feature:22
    Given I am on my profile page                    # features/step_definitions/csv_upload_steps.rb:5
    When I do not attach any file to the upload form # features/step_definitions/csv_upload_steps.rb:17
    And I click the "Submit" button                  # features/step_definitions/common_steps.rb:3
    Then I should see "Please upload a CSV file."    # features/step_definitions/common_steps.rb:7
```

```
Feature: Instructors Page

  Scenario: User should not be able to reach an invalid schedule room                  # features/instructors.feature:2
    Given I am logged in as a user with first name "Test"                              # features/step_definitions/welcome.rb:3
    And a schedule exists with the schedule name "Sched 1" and semester name "Fall 2024" # features/step_definitions/rooms_steps.rb:12
    When I visit the instructor page for id "9a9a9a9"                                  # features/step_definitions/instructors_steps.rb:22
    Then I should see "Schedule not found."                                            # features/step_definitions/common_steps.rb:7

  Scenario: Successfully retrieving instructors for a valid schedule                   # features/instructors.feature:8
    Given a schedule exists with the schedule name "Sched 1" and semester name "Fall 2024" # features/step_definitions/rooms_steps.rb:12
    And I am logged in as a user with first name "Test"                                # features/step_definitions/welcome.rb:3
    And the following instructors exist:                                               # features/step_definitions/instructors_steps.rb:3
      | id_number | first_name | last_name | middle_name | email            |
      | 1001      | John       | Doe       | A           | john@example.com |
      | 1002      | Jane       | Smith     | B           | jane@example.com |
    When I visit the instructors page for "Sched 1"                                    # features/step_definitions/instructors_steps.rb:17
    Then I should see "John Doe"                                                       # features/step_definitions/common_steps.rb:7
    And I should see "Jane Smith"                                                      # features/step_definitions/common_steps.rb:7

  Scenario: Upload instructors data                                                    # features/instructors.feature:20
    Given a schedule exists with the schedule name "Sched 1" and semester name "Fall 2024" # features/step_definitions/rooms_steps.rb:12
    And I am logged in as a user with first name "Test"                                # features/step_definitions/welcome.rb:3
    And I am on the details page for "Sched 1"                                         # features/step_definitions/instructors_steps.rb:35
    When I attach a valid "instructor_file" with path "spec/fixtures/instructors/instructors_valid.csv" # features/step_definitions/instructors_steps.rb:39
    And I click the "Upload Instructor Data" button                                    # features/step_definitions/common_steps.rb:3
    Then I should see "Instructors successfully uploaded."                             # features/step_definitions/common_steps.rb:7
```

```
Feature: Rooms Page

  Scenario: User should not be able to reach an invalid schedule room                  # features/rooms.feature:2
    Given I am logged in as a user with first name "Test"                              # features/step_definitions/welcome.rb:3
    And a schedule exists with the schedule name "Sched 1" and semester name "Fall 2024" # features/step_definitions/rooms_steps.rb:12
    When I visit the rooms page for id "9a9a9a9"                                       # features/step_definitions/rooms_steps.rb:42
    Then I should see "Schedule not found."                                            # features/step_definitions/common_steps.rb:7

  Scenario: User should be able to see room view                                       # features/rooms.feature:8
    Given I am logged in as a user with first name "Test"                              # features/step_definitions/welcome.rb:3
    And a schedule exists with the schedule name "Sched 1" and semester name "Fall 2024" # features/step_definitions/rooms_steps.rb:12
    And the following rooms exist for that schedule:                                   # features/step_definitions/rooms_steps.rb:16
      | campus | building_code | room_number | capacity | is_active | is_lab | is_learning_studio | is_lecture_hall |
      | CS     | BLDG1         | 101         | 30       | true      | true   | true               | true            |
      | GV     | BLDG2         | 102         | 50       | true      | true   | true               | true            |
      | CS     | BLDG3         | 102         | 50       | false     | true   | true               | true            |
    When I visit the rooms page for "Sched 1"                                          # features/step_definitions/rooms_steps.rb:8
    Then I should see "Campus"                                                         # features/step_definitions/common_steps.rb:7
    And I should see "Building Code"                                                   # features/step_definitions/common_steps.rb:7
    And I should see "Show Active Rooms"                                               # features/step_definitions/common_steps.rb:7

  Scenario: User should be able to filter only the active rooms                        # features/rooms.feature:21
    Given I am logged in as a user with first name "Test"                              # features/step_definitions/welcome.rb:3
    And a schedule exists with the schedule name "Sched 1" and semester name "Fall 2024" # features/step_definitions/rooms_steps.rb:12
    And the following rooms exist for that schedule:                                   # features/step_definitions/rooms_steps.rb:16
      | campus | building_code | room_number | capacity | is_active | is_lab | is_learning_studio | is_lecture_hall |
      | CS     | BLDG1         | 101         | 30       | true      | true   | true               | true            |
      | GV     | BLDG2         | 102         | 50       | true      | false  | false              | true            |
      | CS     | BLDG3         | 102         | 50       | false     | false  | true               | true            |
    When I visit the rooms page for "Sched 1"                                          # features/step_definitions/rooms_steps.rb:8
    And I click "Show Active Rooms"                                                    # features/step_definitions/common_steps.rb:11
    Then I should see the following rooms:                                             # features/step_definitions/rooms_steps.rb:36
      | Building Code | Room Number | Capacity |
      | BLDG1         | 101         | 30       |
      | BLDG2         | 102         | 50       |
    And I should not see "BLDG3"                                                       # features/step_definitions/common_steps.rb:15

  Scenario: Upload rooms data                                                          # features/rooms.feature:37
    Given a schedule exists with the schedule name "Sched 1" and semester name "Fall 2024" # features/step_definitions/rooms_steps.rb:12
    And I am logged in as a user with first name "Test"                                # features/step_definitions/welcome.rb:3
    And I am on the details page for "Sched 1"                                         # features/step_definitions/instructors_steps.rb:35
    When I attach a valid "room_file" with path "spec/fixtures/rooms/rooms_valid.csv"  # features/step_definitions/instructors_steps.rb:39
    And I click the "Upload Room Data" button                                          # features/step_definitions/common_steps.rb:3
    Then I should see "Rooms successfully uploaded."                                   # features/step_definitions/common_steps.rb:7
```

```
Feature: Create a new schedule
  As a user
  I want to create a new schedule
  So that I can assign teachers to courses efficiently

  Scenario: Creating a schedule successfully                # features/schedule_create.feature:6
    Given I am logged in as a user with first name "Test"   # features/step_definitions/welcome.rb:3
    And I am on the new schedule page                       # features/step_definitions/schedules_steps.rb:18
    When I fill in "Schedule name" with "Fall 2024 Schedule" # features/step_definitions/common_steps.rb:19
    And I fill in "Semester name" with "Fall 2024"          # features/step_definitions/common_steps.rb:19
    And I click the "Save Schedule" button                  # features/step_definitions/common_steps.rb:3
    Then I should see "Schedule was successfully created"   # features/step_definitions/common_steps.rb:7
    And I should see "Fall 2024 Schedule"                   # features/step_definitions/common_steps.rb:7
    And I should see "Fall 2024"                            # features/step_definitions/common_steps.rb:7

Feature: Delete a schedule
  As a user
  I want to delete a schedule I don't want
  So that I can remove it from the system

  Scenario: Deleting a schedule successfully                 # features/schedule_delete.feature:6
    Given I am logged in as a user with first name "Test"    # features/step_definitions/welcome.rb:3
    And I have created a schedule called "Fall 2024 Schedule" # features/step_definitions/schedules_steps.rb:26
    And I am on the schedules index page                     # features/step_definitions/schedules_steps.rb:22
    When I click the "Delete" button for "Fall 2024 Schedule" # features/step_definitions/schedules_steps.rb:30
    Then I should see "Schedule was successfully deleted"    # features/step_definitions/common_steps.rb:7
    And I should not see "Fall 2024 Schedule"                # features/step_definitions/common_steps.rb:15

Feature: Schedules page
    As a scheduler
    So that I can see the schedules I have worked on
    I want to see the list of my schedules on the landing page and be able to search them

  Background: schedules in database      # features/schedules.feature:6
    Given the following schedules exist: # features/step_definitions/schedules_steps.rb:3
      | schedule_name     | semester_name |
      | Test Schedule 1   | Fall 2024     |
      | Another Schedule  | Spring 2024   |

  Scenario: User sees landing page with generated schedules # features/schedules.feature:13
    Given I am logged in as a user with first name "Test"   # features/step_definitions/welcome.rb:3
    When I visit the schedules index page                   # features/step_definitions/schedules_steps.rb:8
    Then I should see "Test Schedule 1"                     # features/step_definitions/common_steps.rb:7
    And I should see "Another Schedule"                     # features/step_definitions/common_steps.rb:7

  Scenario: User should be able to see each schedule's details # features/schedules.feature:19
    Given I am logged in as a user with first name "Test"      # features/step_definitions/welcome.rb:3
    When I visit the schedules index page                      # features/step_definitions/schedules_steps.rb:8
    And I click on the card for "Test Schedule 1"              # features/step_definitions/schedules_steps.rb:12
    Then I should see "Schedule Name: Test Schedule 1"         # features/step_definitions/common_steps.rb:7
    And I should see "Semester:Fall 2024"                      # features/step_definitions/common_steps.rb:7
    And I should see "Some details about the schedule"         # features/step_definitions/common_steps.rb:7
    But I should not see "Another Schedule"                    # features/step_definitions/common_steps.rb:15
```

```
  Scenario: Search for an existing schedule              # features/schedules.feature:28
    Given I am logged in as a user with first name "Test" # features/step_definitions/welcome.rb:3
    When I visit the schedules index page                # features/step_definitions/schedules_steps.rb:8
    When I search for "Test Schedule"                    # features/step_definitions/schedules_steps.rb:37
    Then I should see "Test Schedule 1"                  # features/step_definitions/common_steps.rb:7
    But I should not see "Another Schedule"              # features/step_definitions/common_steps.rb:15

  Scenario: Search for a non-existing schedule           # features/schedules.feature:35
    Given I am logged in as a user with first name "Test" # features/step_definitions/welcome.rb:3
    When I visit the schedules index page                # features/step_definitions/schedules_steps.rb:8
    When I search for "ABCD"                             # features/step_definitions/schedules_steps.rb:37
    Then I should not see "Test Schedule 1"              # features/step_definitions/common_steps.rb:15
    And I should not see "Another Schedule"              # features/step_definitions/common_steps.rb:15

Feature: Welcome Page

  Scenario: Logged-in user is redirected to their profile with a welcome notice # features/welcome.feature:3
    Given I am logged in as a user with first name "Test"                       # features/step_definitions/welcome.rb:3
    When I visit the welcome page                                               # features/step_definitions/welcome.rb:9
    Then I should be on my profile page                                         # features/step_definitions/OAuth_Authorization_steps.rb:21
    And I should see "Welcome back, Test!"                                      # features/step_definitions/common_steps.rb:7
```

Unit Test Cases Covered:

Instructors Controller

- incorrect schedule id: Checks the invalid case
- without added instructors: When no instructors are added to a schedule
- with added instructors: Happy case, when instructors are added to a schedule

Rooms Controller

- incorrect schedule id: Checks the invalid case
- without added rooms: When no rooms are added to a schedule
- with added rooms: Happy case, when rooms are added to a schedule
- without any filters or sorting: When no filters or sort order is chosen
- with active filter: When filters are added
- with sorting by building_code in ascending order: Test sort by building_code
- with sorting by capacity in descending order: Test sort by capacity

SchedulesController

- GET #index: populates an array of schedules
- GET #show: assigns the requested schedule and renders the template
- GET #new: assigns a new schedule
- POST #create: saves new schedule to DB
- DELETE #destroy: delete schedule from DB
- POST #upload_rooms: upload rooms for a particular schedule
- POST #upload_instructors: upload instructors for a particular schedule

TimeSlotsController

- assigns all time slots when no filters are applied: display all time slots
- filters time slots by day: filter by day
- filters time slots by type: filter by type
- filters time slots by day and type: filter by day and type

Customer Meeting

1) 1:45 PM October 16, 2024
   Place: PETR 102B

   Customer Meeting Details and Feedback

   Presented and showcased the constraint optimization algorithm using test data and clarified doubts related to the input files. Also captured additional requirements for the views.

   Constraint Optimization Algorithm
   - Check if multiple schedules can be generated using the same data, out of which client can pick any one.

   Additional Requirements
   - Rooms csv can be uploaded for every schedule
   - Instructuor preferences and details will be available in the same csv

- Add Block Time Slots feature in the Predefined Courses Screen, ensuring no classes can be scheduled during this time.

2) 1:45 PM October 23, 2024
   Place: PETR 102B

   Customer Meeting Details and Feedback - Sprint 2 MVP Demo

   Demonstrated Landing page, views - schedule, instructors, time slots, and CSV file upload. Discussed regarding the constraint optimization algorithm.

   Feedback
   - The client expressed interest in a feature to manually add/remove individual time slots from the view
   - The client put forward a potential FERPA issue for publicly-hosted identifying data
     - No changes are needed on our side, the professor will run the application locally
     - We need to add build instructions
   - Constraint Optimization Algorithm
     - Potential Issue where the room/class/time assignment prevents all professors from teaching was discussed.
     - We need to test this scenario with our algorithm
     - Introduce nondeterminism in the scheduling algorithm so that many schedules can be generated.
     - Unify ILP/bipartite matching steps into one cost function
       - Discussed deriving cost function that considers wasted rooms and professor happiness.

Links

Deployed App: https://faculty-teaching-assignment-31f5f9c405bc.herokuapp.com/
GitHub Repository: https://github.com/tamu-edu-students/Faculty-Teaching-Assignment
Pivotal Tracker: https://www.pivotaltracker.com/n/projects/2721604
Slack: https://tamu.slack.com/archives/C07PA043PA7
Team Working agreement:
https://github.com/tamu-edu-students/Faculty-TeachingAssignment/blob/documentation/documentation/Fall2024/Team%20Working%20Agreement.md
Code Climate Report:
https://codeclimate.com/github/tamu-edu-students/Faculty-TeachingAssignment