Sprint 3 plan

Content

Team Roles

Product Owner: Colby Endres
Scrum Master: Navya Priya Nandimandalam

Sprint Goal

Our Sprint 3 goal is to run the Constraint Optimization Algorithm on the actual data provided by the client and alongside this implementation, we will also look into running the algorithm to satisfy soft constraints like the instructor's preferences. Additionally, we are working on completing the remaining skeleton UIs and fixing some bugs identified in the previous sprint.

This includes the following tasks

- Create Course View
- Create a schedule room/time slot matrix for visualizing the allotment.
- Add predefined courses and block the time for these to ensure no other class can get scheduled at the same time.
- Parse and update instructor preferences data to try the algorithm on soft constraints
- Bugs - Fix the filtering of time slots in the time slot view

User Stories this sprint

For this sprint, we have decided on roughly 3 points per person.
Here is the table containing the user stories and who they are assigned to.

Story Title	Points	Assigned To
User should be able to edit the added rooms in a schedule by uploading a new csv ● Add tests ● Create a room time slot matrix to represent a schedule in the view.	1	Wahib Sabir Kapdi
User should be able to see the Courses Constraints View ● Test Cases ● Parse data, create course&sections table and populate the tables ● Create a view to display courses	3	Navya Priya Nandimandalam, Abel Gizaw
User should directly land onto the schedules view ● Change redirection, to land on schedules page	1	Colby Endres

User should be able to view the Schedule Room/ Time Slot Matrix ● Test Cases ● Create a matrix UI to display the schedule view ● Read the schedule table into the matrix ● Populate view	2	Pavithra Gopalakrishnan
User should be able to add Predefined Courses and Instructors and Block out Time Slots ● Test Cases ● Remove the blocked time slots and courses from solver ● Display the blocked time slots and predefined courses	2	Wahib Sabir Kapdi
User should be able to navigate to the previous pages using Back Buttons ● Test Cases ● Add back buttons on all pages	1	Navya Unnikrishnan
Instructor Preference - Parse data and populate table ● Test Cases ● Parse the preferences CSV ● Extract courses from the csv ● Create a preferences table ● Add data to the table	2	Navya Unnikrishnan
PoC - Soft constraints with the algorithm ● Modify the algorithm to test the nice-to-have features	1	Abel Gizaw
User should be able to generate schedule by uploading data ● Test cases ● Run the algorithm on the actual data	3	Yuqi Fan, Colby Endres

Backlog

Change	Story	Points
+	User should be able to copy rooms from another schedule	2
+	User should be able to download his generated schedules as csv	2
+	User should be able to hide courses from the generator	2
+	User should be able to lock courses with time slots - After generating schedule	2

Links

Deployed App: https://faculty-teaching-assignment-31f5f9c405bc.herokuapp.com/
GitHub Repo: 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