Sprint 4 plan

Content

Team Roles

Product Owner: Yuqi Fan
Scrum Master: Pavithra Gopalakrishnan

Sprint Goal

Our Sprint 4 goal is to integrate the Constraint Optimization Algorithm using the imported data and the existing application. This algorithm will be further tuned using soft constraints. Additionally, we are working on fully fleshing out some UI elements to satisfy our client's feature requests.

This includes the following tasks

- Add exotic courses to the scheduler
- Ensure that instructors can see only their courses
- Room bookings should only reflect available instructors
- Allowing for the exporting of generated schedules
- Hosting of the algorithm on heroku
- Bugs - Fix tab altering when adding a room booking

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	Time	Points	Assigned To
User should be able to lock courses with time slots after generating schedules ● Add tests ● Select courses to lock ● Update database to make times/rooms.professors unavailable	2 days	2	Wahib Sabir Kapdi, Navya Priya Nandimandalam
PoC - soft constraints with the algorithm ● Test Cases ● Modify the algorithm to test the nice-to-have features	1 day	1	Abel Gizaw
User should be able to block out time slots ● Fix merge conflicts	2 days	2	Pavithra Gopalakrishnan
User should be able to generate the Schedule by uploading data	3 days	3	Yuqi Fan, Colby Endres

● Test Cases ● Add constraints to require courses to be scheduled in supported rooms ● Modify temporal constraints to require courses to be scheduled in supported rooms ● Modify temporal constrains so that we automatically block out times for each room. Currently, all rooms are assumed to support all timeslots, which isn't the case in practice ● Consider removing bipartite matching and merge professors into the ILP. This will require significant modification to the existing problem setup ● Write RSpec and Cucumber tests			
Fix room booking bug ● Test Cases	1 day	1	Pavithra Gopalakrishnan
User should only be able to add allowed instructors to a room booking. ● View available instructors for that time ● Select instructor and attach them to a time and course ● Block their availability so that they can't be double booked ● Test Cases	3 days	3	Abel Gizaw
User should be able to hide courses from the generator ● Select a course to hide ● Update the db to make the course unavailable ● Ensure that the algorithm is synced to this ● Test Cases	2 days	2	Wahib Sabir Kapdi
User should be able to see only his generated schedules. ● Connect schedules to users ● Check user login information before loading schedules ● Test Cases		2	Navya Unnikrishnan
Hosting the algorithm on heroku ● Push algorithm to heroku ● Make sure that the integration works	2 days	2	Yuqi Fan

• Test process			
User should be able to download their generated schedules as csv • Access correct table(s) from db • Convert the table into excel sheet format • Test cases	2 days	2	Navya Priya Nandimandalam
Add exotic courses • Select a time and room • Input exotic courses information • Add to db as a locked course • Test Cases	1 day	1	Wahib Sabir Kapdi
Handle similar course names between courses table and preferences table • Parse stacked courses and try to find them separately in the courses table	1 day	1	Navya Unnikrishnan

Backlog

Change	Story	Points
+	Users should be able to copy rooms from another schedule	2
+	Look into drag n drop functionality in rails	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