

Sprint 4 MVP Report

1. Deployed App

Link: <https://ea-block-scheduler-4fec886e389.herokuapp.com>

2. Sprint Goal

By the end of this sprint, our deployed app will be able to generate schedules with typical blocks and we hope to have a fully functional application with limited features. We also plan to have a user form to be able to show what classes a student has completed.

#38 application makes possible blocks with conditions (8 points)

- Develop algorithm to apply block generating conditions for typical blocks made by admin: 4 hours (Junhyuk Lee)

#39 generate possible blocks (8 points)

- Blocks are generated automatically through algorithm: 6 hours (Junhyuk Lee and Ryann Lu)

#90 Create Database Schema (5 points)

- Creating a way for class data to be stored more efficiently: 4 hours

#91 Fix UI for header and footer (5 points)

- Upgrading the style and making the headers and footers look more like current A&M pages: 2 hours (Chengyan Tsai)

#92 Add UI for landing page (5 points)

- Upgrading the style for the landing page and adding more functionality: 2 hours (Chengyan Tsai)

#93 Fix UI for dashboard (2 points)

- Making sure all buttons on the landing page work and link to correct pages: 1 hour (Chengyan Tsai)

#94 Add corequisite and type column to course table for schedule generation (3 points)

- Populate corequisite and type automatically based on course name: 6 hours (Ryann Lu)

#95 Create form for completed courses (8 points)

- Be able to select completed courses so that the algorithm can tailor a schedule that fits that student: 6 hours (Ryann Lu)

#96 Bug Fix - Dashboard (6 points)

- Handle invalid user login: 3 hours (Adithi Srinath)

3. Sprint Backlog

#36 Set Typical Classes Grouping on the Web (5 points)

#40 User pick rest of classes (5 points)

4. Use of BDD

All stories have Cucumber feature(s) defined with several scenarios each, all Cucumber scenarios are passing and all steps are defined

5. Use of TDD

All models and controllers have specs with at least one example/expectation,

all examples are passing

6. Test Coverage

Test coverage generated by SimpleCov is at least 90%. > Run bundle exec cucumber and rspec to see the result - 94.5% currently.

7. Code quality

Code Climate Report

8. Code Style

All Rubocop cops are enabled and report at most 1 style offense per file. > Run rubocop to see the result

9. Use of Project Tracker

Taiga User Stories:

<https://tree.taiga.io/project/aaronjones05-block-scheduler/taskboard/flamboyant-manatee>

10. Presentation

See the deployed app.