Gaucho Time

Team: WeWriteCode

Authors

Josh Andrews: jaa@ucsb.edu
Irene Evans: Irene_evans@ucsb.edu
Alan Roddick: aroddick@ucsb.edu
Jay Patwardhan: jaypatwardhan@ucsb.edu
Grant Clark: gclark@ucsb.edu

Front-End Github: https://github.com/Joshandrews43/gs_scheduler_frontend
Travis-CI: https://travis-ci.org/Joshandrews43/cs48-wewritecode
Trello: https://trello.com/b/9bJcj514/cs-48-project
Vision Statement

The Problem to be Solved

When it is time to select classes for the next upcoming quarter at UCSB, many Gauchos are very stressed out figuring out how to best schedule their classes. There are multiple factors that come into play here: not wanting schedule conflicts, wanting classes at the right times, not wanting 8am classes, etc.. The list goes on and on. As a result, students spend countless amounts of time trying to put together a best fitting schedule that they would be happy with for the next quarter. However, this comes at the price of them cutting time out of their day to make a schedule, that they could be using to do homework, study, relax, hang out with friends, or any other activity of their desired choice. Our website will solve this problem by creating a list of schedules for students that best fit their needs and preferences in a process that will take the student a maximum time of a minute or two, taking away their burden and stress of having to brute force a schedule that they want next quarter. Essentially, a student will input the classes they are planning to take next quarter and select their preferences such as the time of day they want their classes to be, whether they want their classes spaces out or not, etc.. Then, the website will generate a list of many schedules created for the student that best fit their preferences from which the student can then pick a schedule out of and sign up for those classes on UCSB GOLD, saving them time, energy and stress.

Why is the Problem Important?

Every quarter when it is time to select courses for the upcoming quarter, students go into a frenzy to make their schedules and spend a lot of time creating a schedule for them that they could be using for other activities. This website will help students have many schedules made for them in a matter of a couple minutes instead of manually making one schedule for them in a matter of at least an hour or two which is what students have to do now.

How is the Problem Solved Today?

Today, the problem is solved by students manually creating a schedule for them on a piece of paper through guess and check. Students write and erase classes until they create a desirable
schedule. We want to introduce a much more efficient and faster method.

**Outcome:**

The outcome of the project will be a website that allows students to enter the classes they want for next quarter alongside preferences about their schedule, and the website will give back the student a list of many schedules for the student to choose from. The student can then conveniently choose one of the schedules they like the most and choose the corresponding classes on UCSB GOLD during their pass time.

**Project Milestones**

1. Setup a very basic webpage on heroku with things for the user to interact with that works with fake course information to form a fake schedule.
2. Scrape the data from https://my.sa.ucsb.edu/public/curriculum/coursesearch.aspx into actual data our webpage can use and interact with.
3. Make sure frontend and backend are properly communicating with scraped, real time information.
4. Implement functionality to allow a user to filter their suggested schedules by a variety of filter options, such as time distribution, average grade received, average ratings from RateMyProfessor, and others.
5. Complete webpage styling.
6. Implement functionality to allow users to optimize their scheduling, by finding GEs and classes that best meet their requirements.
Improvements on Final Turn-In From Draft Project

There were some improvements we made from the Draft Project to our Final Turn-In, to improve upon what we needed to. First, we were missing an appendix in the Draft Project but we made sure to include that at the end of our final turn-in, which includes all the technologies that we used for the project. Secondly, We were missing explicitly stating our use cases in the Draft Project, but we made sure to include 10 use cases, all of which have been completed, to our final project turn-in. In the Draft Project, we also did not explicitly write down our sprints on the document; we only had a link to our task breakdown, but we added thorough information about all three sprints here. We lastly also have a very updated and revised Vision Statement for our final project turn-in.
System Architecture

Task Breakdown
https://docs.google.com/document/d/1yCRpZX8MoZ4xPlvWFiDSYQbj7dvAw3AaoCMBEqepWds/edit?usp=sharing

Trello Link
https://trello.com/invite/b/9bJcj514/51830fca7d7ea26aabaa85dd6f7521a8/cs-48-project
Other UML Diagrams (Relating to Structures We Used)

- Dropdown Input
  - Quarters
  - Subjects
  - Courses

- HandleChange()

- Selected Courses
  - course name

- deleteCourse()

- Schedules
  - parseCourses()
  - momentInterval
  - numberOfDays
  - event

- GOLDMiner
  - available quarters
  - available subjects per quarter
  - available subjects
  - available courses
  - all lecture/section times and dates

- GOLDPan
  - scheduling algorithm
  - filtering algorithm

- Data Storage
  - stores list of schedules
Schedule
- courses: List<Course>
  - getCourse(): List<Course>
  - setCourses(courses: List<Course>): void
  - getLecture(): List<Lecture>

Lecture
- sections: List<Section>
  - getSections(): List<Section>
  - setSections(sections: List<Sections>): void
  - addSection(section: Section): void

Course
- courseid: String
- lectures: List<Lecture>
  - getCourse(): String
  - setCourses(courses: List<Course>): void
  - getLecture(): List<Lecture>
  - addLecture(lecture: Lecture): void

Session
- days: List<String>
- time: Time
  - getTime(): Time
  - setTime(time: Time): void

Time
- start: TimeObj
- end: TimeObj
  - getStart(): TimeObj
  - setStart(time: TimeObj): void
  - getEnd(): TimeObj
  - setEnd(time: TimeObj): void

Lecture
- sections: List<Section>

Course
- courses: List<Course>
- lectures: List<Lecture>
Sequence Diagrams

1. <<create>> a new GoldMiner
2. Get the most recent quarter(s)
3. <<create>> a new GoldMineController to get data from one quarter
4. Mine the quarter

For each subject in the quarter, a new Callable is created with a new GoldMiner, and called asynchronously amongst n-threads.

4.1. <<create>> a new GoldMiner Callable for every unique subject area
4.2. Get all the course information of a specific subject and quarter (asynchronous with other subjects)
4.3. Return a subject object representing all courses of the subject and quarter in a JSON format.
4.4. Combine all subjectObj returns into one object
4.5. Store the combined quarter object into the file system.

From here, the application continues to run to accept HTML requests.
1: POST /api/v1/schedules

1.1: generate()

1.1.1: splitRequest(request)
1.1.2: findViableSchedules()
1.1.3: sort()

1.1.4: ScheduleResponse

1.1.5: RESPONSE: ScheduleResponse
Sprint Planning

**Sprint 1 (4/15-4/26):**
Frontend - Jay, Irene, Josh
Backend - Alan, Grant

- Learn React
- Scrape course information from
  [https://my.sa.ucsb.edu/public/curriculum/coursesearch.aspx](https://my.sa.ucsb.edu/public/curriculum/coursesearch.aspx)
- Investigate scraping utilities and implementation details. *(2 days)*
- Begin using chromedriver and implementing Miner class *(1 day)*
- Implement Miner class v1 (no multithreading) *(3 days)*
- Implement Miner class v2 (support for multithreading)
- Get very basic-level webpage functional for course viewing
  - Create JSON format sample with types
  - Create a basic functional website hosted with create-react-app
  - Create components for inputs using styled components
    - Create components for dropdowns
    - Create components for text inputs
    - Create component for generate buttons
  - Populate inputs with dummy data
  - Create separate file and read from file to emulate what we will be doing

**Sprint 1 Retrospective:** One issue we discussed to fix was that the scraping of the course information was taking around forty minutes for one quarter alone, which was not ideal nor convenient. So in order to fix it, our back-end team implemented multithreading to get the scraping down to less than ten minutes which performed much better. On the front-end, one thing that came up was that the user was able to select classes but not delete, so we fixed that by adding a delete button next to each selected course which would remove the course from the array of selected courses when pressed

**Sprint 2 (4/29-5/10):**
Frontend - Jay, Irene, Josh
Backend - Alan, Grant

- Create Generate Schedules button
- Add title
- When add course button is clicked, the class should appear on the right of the webpage
• Create option to delete added courses
• Outline server architecture in Java objects
• Create viable schedules with only mandatory classes and merge changes to master
• Create BruteForceScheduler.java for schedule generation
• Style webpage
• Outline server architecture in Java objects *(5 days)*

• Add onto / refactor POJOs for scheduling *(1 day)*
• Help create BruteForceScheduler.java for schedule generation *(1 day)*
• Create viable schedules with only mandatory classes and merge changes to master

**Sprint 2 Retrospective:** When the “Generate Schedules” button was pressed, the right things were not completely disappearing to make room for the list of schedules output, and the schedule templates with the highlighted classes were not showing up yet which needed to be implemented, all of which ultimately got done. On the backend, the course generating algorithm was having bugs where it was not correctly handling courses with no sections, so we fixed that so our app could work for all courses. We also needed to implement the filters menu on the front end to work with the filtering algorithms that the backend had created, which got done.

**Sprint 3 (5/13 - 5/24):**
**Frontend** - Jay, Irene, Josh  
**Backend** - Alan, Grant

• Filter schedules by times
• Output all viable schedules when generate schedules button is clicked
• Make small demo of GoldMiner utility to illustrate the scraping process.
• Begin work on architecture for filtering and adding optional courses. *(2 days)*
- Create a TimeFilter class to filter time and add interface and abstract classes for Filters to allow for easier parsing from JSON into POJOs. *(1 day)*
- Add algorithm for allowing for optional courses in the scheduling process. *(1 days)*
- Add Full versions of course representation to allow for creating the frontend general.json *(2 days)*
- Add GeneralController and GoldVault classes for controlling general.json flow to frontend. *(1 days)*

**Sprint 3 Retrospective:** We wanted to make some changes just to keep refining such as keeping on revising and updating the user interface. We also wanted to reformat where some items on our homepage were placed to make it more user friendly, which we did complete doing.
Challenges Faced/Overcome:

We had challenged spanning from the backend to the front-end that we had to overcome. On the front-end a challenge was that most of the back-end people had never used React before, so we had to spend a lot of time in sprint one learning how to use React in order to use that for the project. On the backend, the back-end team had to learn how technologies ChromeDriver and Selenium Webdriver worked for the scraping utility. The backend spent a lot of time learning how these technologies worked to make an efficient and effective scraper. The back-end team also had to experiment a lot with which type of algorithm would work best, such as a genetic algorithm, or not a non-genetic algorithm etc.. Once the scraper was made, it was however also taking around forty minutes to scrape information for one quarter alone which was not practical at all, so that backend had to learn and implement multithreading to fix this problem.

Missing/Remaining Features and Functionality:

We hope to add a feature which color codes all the classes on the schedule display output to make it nicer to read. We also hope to add a filter by Rate my Professor option that displays the classes with the highest RateMyProfessor ratings at the top. We also want to show the available space in each class in the output of class schedules, and not display the schedules with full classes. On top of having the RateMyProfessor filter, we would also like to display the average RateMyProfessor rating for each class whether the RateMyProfessor filter is selected or not, as students our team has talked to have said this would be a very useful feature.
Test Files

GoldMinerTest.java
This class tests the GoldMiner.java class’s public non-asynchronous methods by outputting the scraping results to a file. The limitations of this, is that this class can only really test if the scraping utility is actually scraping, rather than testing for the accuracy of the data it actually scraped. As a result, testing for accuracy had to be done manually by cross-referencing the information scraped with the data provided by UCSB.

BruteForceSchedulerTest.java
This class tests the BruteForceScheduler.java class’s public methods. It “sends” a list of courses to the algorithm and generates a large number of schedules. To test, we simply checked for the number of schedules it generated, as trying to test each schedule for non-conflicts would involve rewriting our non-conflicting algorithms, which would defeat the purpose of the test entirely. In order to check for non-conflicts, we waited until the frontend could display schedules, and then we ironed-out any issues after that point.

DayFilterTest.java
This class tests the DayFilter.java class’s getFitness() method. We essentially gave the filter a schedule that does not conflict, and tested to see if it could properly provide us a value between 0 and 1 that accurately represents how well it fits our DayFilter.

GapsFilterTest.java
This class tests the GapsFilter.java class’s getFitness() method. We again gave the filter a schedule that does not conflict, and tested to see if it could properly provide us a value between 0 and 1 that accurately represents how well it fits our GapsFilter.
Requirements:

Use Cases

Use Case #1: GOLDMiner
Actors: GOLDVault, EventListener, Course Webpage
Pre-Condition: Server is running and course webpage is updated.
Basic Path:
1. New Quarter Path:
   1.1. EventListener detects a new quarter is listed on the Course Webpage.
   1.2. GOLDMiner scrapes all new data from the site, and stores it in the GOLDVault for later processing.
2. Quarter Update Path:
   2.1. EventListener detects an update to an ongoing or future quarter on the Course Webpage.
   2.2. GOLDMiner scrapes any new information and updates the GOLDVault. May also simply scrape all data, and replace the data contained in the GOLDVault.
Alternative Paths:
   1. If GOLDMiner is unable to scrape the data from the website, keep the current scraped data. GOLDMiner will try to scrape the data again after an update to the Course Webpage, at which point it will either follow the Quarter Update Path, or fail to scrape again. After (n) failed attempts, the system will attempt to notify developers working on the project via email of the issue.

Postcondition: Information has been correctly scraped.

https://trello.com/c/jdOAC2Gi/8-goldminer-java-scraper

Use Case #2: Display the Correct Courses to Select From
Actors: Student, GOLDMiner, Front End
Precondition: The input dropdown menus have been created for the student to select courses from.

Basic Path:

1. The Front End will extract the right information from GOLDMiner pertaining to which subjects and courses are available per quarter.
2. The Front End will then display that corresponding information in the dropdown menus for the student to view and select relevant classes from.

Alternate Path:

1. If there are no courses available for a subject, then instead of any courses displaying or instead of just an empty dropdown, the dropdown menu will display an item labeled “No Courses”.
2. If the “No Courses” item is shown, the student will not be able to add that item as a course as part of the desired courses.

Postcondition: All of the correct offered subject and courses will be viewable for the student in the dropdown menus for the three latest quarters.

https://trello.com/c/33jItMxM/26-be-able-to-add-appropriate-classes-on-schedule-template

Use Case #3: Main Screen of the Website

Actors: Student, Front End

Precondition: The student is on the website.

Basic Path:

- Title of the website.
- Multiple input dropdown menus exist.
- An add course button allowing the student to add the course.
- “Generate Schedules” button
- Dropdown menu for filter options.

Postcondition: Classes correctly display when chosen and correctly get deleted when wanting to delete.

https://trello.com/c/ONQ3RZEC/10-initial-site-displaymain-screen
Use Case #4: GOLDPan Scheduler

Actors: Student, Backend
Precondition: The student has selected their desired classes and clicked “Generate Schedules”.
Basic Path:
1. GOLDPan receives information about the desired classes.
2. GOLDPan’s algorithm creates and stores the list of viable schedules for the student.
3. The information about the schedules is then sent elsewhere to be used.
Postcondition: The resulting information about the schedules is now ready to be displayed in schedule templates on the front-end to be viewed by the user.

https://trello.com/c/igOCnPNl/15-goldpan-schedule-filter

Use Case #5: GOLDPan Filtering Algorithm

Actors: Student, Backend
Precondition: The student has selected their desired classes and has selected a filter option to filter the schedules by.
Basic Path:
1. The student selects filter options on top of their desired classes.
2. That information is then sent to GOLDPan, and the unfiltered list of schedules is made using GOLDPan Scheduler.
3. The GOLDPan filtering algorithm then uses that list of schedules and determines the correct order of the schedules according to the selected filters.
4. That information is then stored and sent elsewhere to be used.
Postcondition: The resulting information of the order of the schedules is ready to be displayed as schedule templates on the front end for the user to be able to view.
Use Case #6: Input Dropdown Menus
Actors: Student, Frontend
Precondition: The student is on the website and ready to pick classes.
Basic Path:
1. There are three dropdown menus: one for quarter, one for subject, one for class.
2. When the student clicks on any of them, the correct options are shown in the dropdown.
3. When the student clicks on an option from the dropdown, the selected option is shown at the top of the dropdown.
Alternate Path:
- If no options to display in a dropdown in either the subjects or courses, then there will be one item in the dropdown saying “No *subjects/courses*”
Postcondition: The student can see all the input options for quarters, subjects and courses in the dropdowns and be able to select from them.

Use Case #7: SpringBoot Server
Actors: Frontend, Backend, Server
Description: Server to manage all requests on the frontend and backend.
Precondition: Front and back end are up and running.
Basic Path:
1. If a POST request is made by the backend or frontend, then the request should go through to the front end or backend respectively and information should be correctly passed.
2. If a GET request is made by the frontend or backend to the backend or frontend respectively, the information should be received correctly.
3. If a PUT request is made, the correct objects should be updated with the right information.

Postcondition: Information can be correctly be passed between the backend and the frontend.

https://trello.com/c/vlc2dWIu/16-set-up-rest-server-with-spring-boot

Use Case #8: Display of Selected Courses
Actors: Student, Frontend
Precondition: The student has selected a course from the input dropdown and is ready to add the course to desired courses.
Basic Path:
1. The student selects a specific course from the dropdown input menu and clicks on the “Add Course” button.
2. Each added course name shows in a wrapped button below the dropdown inputs.
3. Each button with the added course names also has functionality to be deleted from the list of desired courses if student wants to un-add the course.
Postcondition: Student has added all desired courses and all the desired courses are displayed as wrapped buttons.

https://trello.com/c/lMnuoLEf/19-create-selectedcourse-component

Use Case #9: Display of List of Schedules
Actors: Student, Frontend
Precondition: The student has selected all desired courses and filters and has hit the “Generate Schedules” button.
Basic Path:
1. After “Generate Schedules” button is pressed, the list of schedules is displayed top to bottom.
2. Each shaded interval in the schedule displays the class name and time.
3. The schedules are enumerated.
Postcondition: The student is able to view the list of schedules generated for them top to bottom.
https://trello.com/c/EAKUw1B5/11-schedule-display-feature

Use Case #10: Delete a selected course

Actors: Student, Frontend

Precondition: The student has added at least one desired course.

Basic Path:
1. The student decides they want to delete one or more selected courses
2. The student clicks on the X next to the course name
3. The course that was clicked on successfully disappears

Postcondition: The student should no longer see the deleted course on the screen

https://trello.com/c/ONQ3RZEC/10-make-the-initial-site-display
Appendix

Technologies Used

1) Frontend:
   a) React.js used to build the front-end.

2) Backend:
   a) Java used to write the back-end.
   b) Selenium Webdriver and Chromedriver for scraping utility
   c) UCSB Curriculum Course Search website for information on courses

3) Communication Between Frontend and Backend:
   a) Use of Spring Boot server to send requests between front and back end.
   b) Postman for debugging and testing requests
Commits

Link to commits, frontend:
https://github.com/Joshandrews43/gs_scheduler_frontend/commits/master

Backend:
https://github.com/Joshandrews43/cs48-wewritecode/commits/master

Front End:

Commits on Jun 4, 2019

fixed

Joshandrews43 committed 7 days ago 380fd69

Commits on Jun 2, 2019

added clear filters and title

Joshandrews43 committed 9 days ago 797b898

Commits on May 24, 2019

added loading indicator

Joshandrews43 committed 18 days ago b0e2a84

Commits on May 23, 2019

removed log

Joshandrews43 committed 19 days ago

working filters / filters are ugly

Joshandrews43 committed 19 days ago 80d30f9
Commits on May 22, 2019

Merge pull request #8 from Joshandrews43/close ...
Joshandrews43 committed 20 days ago
various fixes / partial filters
Joshandrews43 committed 20 days ago

Commits on May 21, 2019

fixed duplicates
Joshandrews43 committed 21 days ago

Commits on May 20, 2019

fixed
Joshandrews43 committed 22 days ago
working calendar
Joshandrews43 committed 22 days ago
new ui no calendar test
Joshandrews43 committed 22 days ago
Merge pull request #7 from Joshandrews43/Jay2 ...
Joshandrews43 committed 22 days ago

Latest Version With Multiple Schedules in Output
Jay Patwardhan committed 22 days ago
Merge pull request #6 from Joshandrews43/alan ...
Joshandrews43 committed 22 days ago
Joshandrews43 committed 22 days ago

alphabetically sort subjects / fall data

aroddick committed 22 days ago

Commits on May 16, 2019

Added Filter Dropdown

Jay Patwardhan authored and Jay Patwardhan committed 26 days ago

Latest Working Update

Jay Patwardhan committed 26 days ago

Broken Calendar Version

Jay Patwardhan committed 26 days ago

Merge branch 'master' into Jay2

Jay Patwardhan committed 26 days ago

Latest

Jay Patwardhan committed 26 days ago

Commits on May 15, 2019

Sections Parse

Jay Patwardhan authored and Jay Patwardhan committed 27 days ago

Commits on May 14, 2019

fixed generate button
### Commits on May 8, 2019

<table>
<thead>
<tr>
<th>Commit Message</th>
<th>Author</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Merge pull request #5 from Joshandrews43/josh</td>
<td>Joshandrews43</td>
<td>28 days ago</td>
</tr>
<tr>
<td>added partial date parsing</td>
<td>Joshandrews43</td>
<td>28 days ago</td>
</tr>
<tr>
<td>Merge branch 'master' of <a href="https://github.com/Joshandrews43/gs_schedule">https://github.com/Joshandrews43/gs_schedule</a>...</td>
<td>Joshandrews43</td>
<td>May 8, 2019</td>
</tr>
<tr>
<td>Merge pull request #4 from Joshandrews43/Jay2...</td>
<td>Joshandrews43</td>
<td>May 8, 2019</td>
</tr>
</tbody>
</table>

### Commits on May 7, 2019

<table>
<thead>
<tr>
<th>Commit Message</th>
<th>Author</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Latest Version of my Project</td>
<td>Jay Patwardhan</td>
<td>May 8, 2019</td>
</tr>
<tr>
<td>Rough Draft Table</td>
<td>Jay Patwardhan</td>
<td>May 7, 2019</td>
</tr>
</tbody>
</table>

### Commits on May 3, 2019

<table>
<thead>
<tr>
<th>Commit Message</th>
<th>Author</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Most Recent Fixed Code with working Disabled Add Course Button when G...</td>
<td>Jay Patwardhan</td>
<td>May 3, 2019</td>
</tr>
<tr>
<td>Disabled Button Is Working Now</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Button Disable

Jay Patwardhan authored and Jay Patwardhan committed on May 3

created post request method

Joshandrews43 committed on May 3

Commits on May 2, 2019

current attempt #2

Jay Patwardhan authored and Jay Patwardhan committed on May 2

Working on Disabling Button

Jay Patwardhan authored and Jay Patwardhan committed on May 2

Commits on May 1, 2019

May1

Jay Patwardhan committed on May 1
Commit: Branch: master

Commits on Apr 29, 2019

- [J](https://github.com/JayPatwardhan/commit)
  - Jay Patwardhan committed on Apr 29
  - [7fa1a37](https://github.com/JayPatwardhan/commit)

- [Jay2](https://github.com/JayPatwardhan/commit)
  - Jay Patwardhan committed on Apr 29
  - [663b29b](https://github.com/JayPatwardhan/commit)

- [Merge pull request #3 from Joshandrews43/josh](https://github.com/Joshandrews43/commit)
  - Joshandrews43 committed on Apr 29
  - Verified [6cb34dd](https://github.com/Joshandrews43/commit)
  - changed gradient
  - Joshandrews43 committed on Apr 29
  - [7cb52b3](https://github.com/Joshandrews43/commit)

Commits on Apr 27, 2019

- [classes](https://github.com/Joshandrews43/commit)
  - Joshandrews43 committed on Apr 27
  - [9cb2bb7](https://github.com/Joshandrews43/commit)

Commits on Apr 26, 2019

- [Merge pull request #2 from Joshandrews43/josh](https://github.com/Joshandrews43/commit)
  - Joshandrews43 committed on Apr 26
  - Verified [cb3048d](https://github.com/Joshandrews43/commit)

- [Merge branch 'master' into josh](https://github.com/Joshandrews43/commit)
Joshandrews43 committed on Apr 26

**my stuff**

Joshandrews43 committed on Apr 26

Merge pull request #1 from Joshandrews43/Jay2 ...

Joshandrews43 committed on Apr 26

**tes**

Jay Patwardhan committed on Apr 26

**friday spring 1**

Jay Patwardhan committed on Apr 26

Commits on Apr 24, 2019

**bug test**

Jay Patwardhan authored and Jay Patwardhan committed on Apr 24

Commits on Apr 22, 2019

**first commit**

Jay Patwardhan committed on Apr 22

Commits on Apr 17, 2019

**lifted up form state**
Commits on Apr 15, 2019

*Initial commit from Create React App*

Commits on May 22, 2019

*Updated algorithm to account for user specified gaps*
Commits on Jun 8, 2019

**fixed day option in test**
arooddick committed 3 days ago

**Fixed check for section in schedule when a course in schedule may not...**
arooddick committed 3 days ago

**Fixed error when only one schedule was generated**
arooddick committed 3 days ago

Commits on Jun 4, 2019

**Added filters to generate request and removed FilterRequest**
Grant Clark committed 7 days ago

**Added filters to generate request and removed FilterRequest**
Merge branch 'master' of github.com:Joshandrews43/cs48-wewritecode

Grant Clark committed 7 days ago

Fixed scheduling conflicts

Grant Clark committed 7 days ago

Changed the option name

arooddick committed 7 days ago

Improved logic of gaps filter again

arooddick committed 8 days ago

Commits on Jun 3, 2019

Refactored Gaps Filter

arooddick committed 8 days ago

fixed gaps filter test

arooddick committed 8 days ago

refined gaps filter

arooddick committed 8 days ago

Commits on May 31, 2019

Merge branch 'scheduler'

Grant Clark committed 11 days ago
Commit on May 28, 2019

- **Fixed failing build**
  - Grant Clark committed 11 days ago

- **Removed unimplemented availability filter**
  - Grant Clark committed 11 days ago

- **Changed Gaps Options**
  - Grant Clark committed 11 days ago

Commits on May 28, 2019

- **Merge pull request #30** from Joshandrews43/scheduler ...
  - aroddick committed 14 days ago

- **Merge pull request #29** from Joshandrews43/scheduler_aroddick ...
  - aroddick committed 14 days ago

- **Fixed tests**
  - aroddick committed 14 days ago

- **Added Scrape mapping**
  - Grant Clark committed 14 days ago

- **Fixed but with conflicting schedules**
  - aroddick committed 14 days ago

Commits on May 24, 2019

- **Merge pull request #28** from Joshandrews43/scheduler ...

- **Merge pull request #27** from Joshandrews43/scheduler ...

- **Verified**

- **Verified**
aroddick committed 19 days ago

Merge pull request #27 from Joshandrews43/scheduler_aroddick ...

aroddick committed 19 days ago

Fixed refresh of days array

aroddick committed 19 days ago

Commits on May 23, 2019

Merge pull request #26 from Joshandrews43/scheduler ...

aroddick committed 19 days ago

Merge pull request #25 from Joshandrews43/scheduler_aroddick ...

aroddick committed 19 days ago

Added check for null option

aroddick committed 19 days ago

Commits on May 22, 2019

Removed git attributes

aroddick committed 20 days ago

Reverted changes

aroddick committed 20 days ago

testing .gitattributes

aroddick committed 20 days ago
Commits on May 21, 2019

Merge pull request #22 from Joshandrews43/scheduler ...
aroDDick committed 21 days ago

Merge pull request #21 from Joshandrews43/scheduler_aroDDick ...
aroDDick committed 21 days ago

Fixed tests
aroDDick committed 21 days ago

Merge pull request #20 from Joshandrews43/scheduler ...
aroDDick committed 21 days ago

Merge pull request #19 from Joshandrews43/scheduler_aroDDick ...
aroDDick committed 21 days ago
Added fix for course with no sections
aroddick committed 21 days ago

Merge pull request #18 from Joshandrews43/scheduler ...
aroddick committed 21 days ago

Merge pull request #17 from Joshandrews43/scheduler_aroddick ...
aroddick committed 21 days ago

Merge pull request #16 from Joshandrews43/scheduler_aroddick ...
aroddick committed 21 days ago

Fixed bug where if a course was not added after another course was ad...
aroddick committed 21 days ago

Removed extra imports
Grant Clark committed 21 days ago

Fixed build error
Grant Clark committed 21 days ago

Merge scheduler -> gclark_scheduler
Grant Clark committed 21 days ago

Merge branch 'scheduler' of github.com:Joshandrews43/cs48-wewritecode...
Grant Clark committed 21 days ago

logs added to gitignore
Grant Clark committed 21 days ago

Small changes

Grant Clark committed 21 days ago

Merge pull request #15 from Joshandrews43/scheduler ...

aroddick committed 22 days ago

Merge pull request #14 from Joshandrews43/scheduler_aroddick ...

aroddick committed 22 days ago

Fixed assertion

aroddick committed 22 days ago

Refactored gaps filter and fixed errors

aroddick committed 22 days ago

Commits on May 20, 2019

Merge pull request #13 from Joshandrews43/scheduler ...

Joshandrews43 committed 22 days ago

Verified d77221c

Merge pull request #12 from Joshandrews43/scheduler_aroddick ...

Joshandrews43 committed 22 days ago

Verified d6d291f

Fixed bug where data would not be cleared between requests

aroddick committed 22 days ago

merge
<table>
<thead>
<tr>
<th>Author</th>
<th>Message</th>
<th>Commit ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>aroddick</td>
<td>Added GapsFilterTests</td>
<td>6567c86</td>
</tr>
<tr>
<td>aroddick</td>
<td>Fixed Travis-CI Build failing</td>
<td>F7d8fbr</td>
</tr>
<tr>
<td>aroddick</td>
<td>Added functionality for updating client courses and general information</td>
<td>9bb1932</td>
</tr>
<tr>
<td>Grant Clark</td>
<td>First implementation of gaps filter and added tests</td>
<td>d4edc73</td>
</tr>
<tr>
<td>Grant Clark</td>
<td>Added test for exception</td>
<td>4ad6d18</td>
</tr>
<tr>
<td>aroddick</td>
<td>Merge pull request #11 from Joshandrews43/scheduler_aroddick</td>
<td>109a44f</td>
</tr>
</tbody>
</table>

Commits on May 17, 2019

- **Fixed Travis-CI Build failing**
  - Grant Clark committed 25 days ago

Commits on May 16, 2019

- **Added functionality for updating client courses and general information**
  - Grant Clark committed 26 days ago

- **Added test for exception**
  - aroddick committed 27 days ago

- Merge pull request #11 from Joshandrews43/scheduler_aroddick
Refactored code to maintain readability and match form throughout code

arooddick committed 27 days ago

Commits on May 15, 2019

Merge pull request #10 from Joshandrews43/scheduler_aroddick ...

arooddick committed 27 days ago

Added check for minimize or maximize option and tested output

arooddick committed 27 days ago

c6c895b

changed the options in the day filter

arooddick committed 27 days ago

be816c3

Modified TimeFilter to work properly; Capped POST response to 50 by d...

Grant Clark committed 27 days ago

8ec68b5

Added logs to .gitignore

Grant Clark committed 27 days ago

daa1854

Added log

Grant Clark committed 27 days ago

f7d1de4

Attempt at displaying schedules on backend

Grant Clark committed 27 days ago

a68b54a

Attempt at displaying schedules on backend

Grant Clark committed 27 days ago
<table>
<thead>
<tr>
<th>Commit</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>d09874a</td>
<td><strong>Added GET and PUT requests for ScheduleController</strong></td>
</tr>
<tr>
<td>Grant Clark committed 27 days ago</td>
<td></td>
</tr>
<tr>
<td>e11d07a</td>
<td><strong>Added TODOs in BruteForceScheduleTest.java for further tests to make</strong></td>
</tr>
<tr>
<td>Grant Clark committed 28 days ago</td>
<td></td>
</tr>
<tr>
<td>7ba8a6c</td>
<td><strong>Added Optional course selection and more Unit tests</strong></td>
</tr>
<tr>
<td>Grant Clark committed 28 days ago</td>
<td></td>
</tr>
</tbody>
</table>

Commits on May 14, 2019

<table>
<thead>
<tr>
<th>Commit</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>3860b72</td>
<td><strong>pulled from scheduler</strong></td>
</tr>
<tr>
<td>aroddick committed 28 days ago</td>
<td></td>
</tr>
<tr>
<td>5809c3e</td>
<td><strong>Fixed failing test</strong></td>
</tr>
<tr>
<td>Grant Clark committed 28 days ago</td>
<td></td>
</tr>
<tr>
<td>a0e07d1</td>
<td><strong>Added Optional Course Functionality</strong></td>
</tr>
<tr>
<td>Grant Clark committed 28 days ago</td>
<td></td>
</tr>
<tr>
<td>18a448c</td>
<td><strong>Added filter parsing functionality</strong></td>
</tr>
<tr>
<td>Grant Clark committed 28 days ago</td>
<td></td>
</tr>
</tbody>
</table>

Commits on May 13, 2019

<table>
<thead>
<tr>
<th>Commit</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>d0d9b3a</td>
<td><strong>merged with scheduler</strong></td>
</tr>
<tr>
<td>aroddick committed 29 days ago</td>
<td></td>
</tr>
<tr>
<td>7297008</td>
<td><strong>cleaned up bruteforce scheduler</strong></td>
</tr>
</tbody>
</table>
aroddick committed 29 days ago

**Added ability to get from Filters**

Grant Clark committed 29 days ago

**Added TimeFilter**

Grant Clark committed 29 days ago

Commits on May 9, 2019

**Merge pull request #9 from Joshandrews43/scheduler_aroddick ...**

aroddick committed on May 9

**Implemented DayFilter and added a test for it**

aroddick committed on May 9

**Formatted isConflicting function**

aroddick committed on May 9

Commits on May 8, 2019

**GET mapping test removed**

Grant Clark committed on May 8

**Enabled CORS**

Grant Clark committed on May 8

**GET Test**
<table>
<thead>
<tr>
<th>Commit Message</th>
<th>Author</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Switch consume/produces type</td>
<td>Grant Clark</td>
<td>May 8</td>
</tr>
<tr>
<td>Merge branch 'gclark_scheduler'</td>
<td>Grant Clark</td>
<td>May 8</td>
</tr>
<tr>
<td>sout</td>
<td>Grant Clark</td>
<td>May 8</td>
</tr>
<tr>
<td>Merge branch 'master' into scheduler_aroddick</td>
<td>aroddick</td>
<td>May 8</td>
</tr>
<tr>
<td>Merge branch 'scheduler' into scheduler_aroddick</td>
<td>aroddick</td>
<td>May 8</td>
</tr>
<tr>
<td>Merge branch 'scheduler' into scheduler_aroddick</td>
<td>aroddick</td>
<td>May 8</td>
</tr>
<tr>
<td>Added jaxb dependency</td>
<td>aroddick</td>
<td>May 8</td>
</tr>
<tr>
<td>Merge branch 'gclark_scheduler' into scheduler</td>
<td>Grant Clark</td>
<td>May 8</td>
</tr>
</tbody>
</table>

Commits on May 8, 2019

<table>
<thead>
<tr>
<th>Commit Message</th>
<th>Author</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>courseld -&gt; courseID</td>
<td>Grant Clark</td>
<td>May 8</td>
</tr>
</tbody>
</table>
Grant Clark committed on May 8

committing logs

c77debd

aroddick committed on May 8

Merge branch 'scheduler' of github.com:Joshandrews43/cs48-wewritecode...

8503bfa

Grant Clark committed on May 8

Fixed Time Object

4fc97b8

aroddick committed on May 8

Removed import from Application

82e6b3c

Grant Clark committed on May 8

GenerateSchedules complete

419e75a

Grant Clark committed on May 8

Updated generateSchedules POST; Issue: Request and Response List<Filter>

848218c

Grant Clark committed on May 8

Resolving merge conflicts

b466259

Grant Clark committed on May 8

Resolving conflicts

1550182

Grant Clark committed on May 8

Switching from in-memory database to java objects

23db316
Switching from in-memory database to java objects

Added another course to the test

Brute force algorithm completed and created a test

Added getters and setters for Schedule objects and it's derivatives a...

Progress on database layout before attempting to remove database layout

Added in-memory database base implementation

Merge branch 'gclark_scheduler' into scheduler

Adding precursors for InMemoryDB
Grant Clark committed on May 3

Added some helper methods for ScheduleManager subject to change

aroddick committed on May 3

Merge pull request #6 from Joshandrews43/scheduler_aroddick ...

aroddick committed on May 3

fixed quarterName string to get the quarter from file

aroddick committed on May 3

Refactored code worked on from aroddick_scheduler and added @author a... ...

Grant Clark committed on May 3

Conflict resolution

Grant Clark committed on May 3

Somewhat finalized project architecture

Grant Clark committed on May 3

merge

aroddick committed on May 3

Renamed UpdateController

aroddick committed on May 3

Commits on May 2, 2019

Continued work on class structure
Grant Clark committed on May 2

Continued work on class structure  

Grant Clark committed on May 2

More class layout stuff

Grant Clark committed on May 2

More class layout stuff

Grant Clark committed on May 2

Starting to work out specifics on class structure and layout

Grant Clark committed on May 2

Added test in UpdateControllerTest

aroddick committed on May 2

Commits on May 1, 2019

Moved converting methods into RequestController

aroddick committed on May 1

Created new class UpdateController to deal with updating quarters

aroddick committed on May 1

Commits on May 1, 2019

Adjusted Spring classes to use GSON interface and JSON implementation

Grant Clark committed on May 1
Commits on Apr 30, 2019

Merge branch 'gclark_scheduler' into scheduler ...

Grant Clark committed on Apr 30

Added other outline files

Grant Clark committed on Apr 30

Merge pull request #5 from Joshandrews43/scheduler_aroddick ...

aroddick committed on Apr 30

uncommented miner

aroddick committed on Apr 30

Set up GoldPan utility

aroddick committed on Apr 30

Commits on Apr 29, 2019

Created Java classes for algorithm to generate schedules

aroddick committed on Apr 29

Broke down update quarter request into helper functions

aroddick committed on Apr 29

Added some guide/outline files for filtering

Grant Clark committed on Apr 29

Made changes to ScheduleOutlinePOST.json
Grant Clark committed on Apr 29

**Added new outline files**

8ba32a5

Grant Clark committed on Apr 29

**Refactored packages**

9069ca0

Grant Clark committed on Apr 29

**Added functionality for formatted json and raw json file creation**

e64eb55

Grant Clark committed on Apr 29

**Removed CommandLineRunner from Application.java**

7dc9f3b

Grant Clark committed on Apr 29

**Removed command-line arguments**

8038ab9

Grant Clark committed on Apr 29

Merge pull request #4 from Joshandrews43/aroddick_scheduleAlgorithm ...

aroddick committed on Apr 29

deleted extra folder

Verified 5555a32

aroddick committed on Apr 29

removed second controller and started POST request for generating sched...

1fe4450

aroddick committed on Apr 29

Merge pull request #3 from Joshandrews43/gclark_miner ...

7adc5e6
Joshandrews43 committed on Apr 29

**Modified travis.yml to get rid of un-needed junk**

Grant Clark committed on Apr 29

**Modified GoldMinerTests to cut down on testing time**

Grant Clark committed on Apr 29

**Attempt to get travis-ci to work with chromedriver (again)**

Grant Clark committed on Apr 29

**Trying to get travis-ci working work chromedriver again**

Grant Clark committed on Apr 29

**Trying to get chromedriver working with travis-ci**

Grant Clark committed on Apr 29

**Attempt to fix GoldMiner class from giving IllegalStateException in t...**

Grant Clark committed on Apr 29

**Fixed missing </dependencies> tag**

Grant Clark committed on Apr 29

**Removed duplicate gson dependency**

Grant Clark committed on Apr 29

**Resolving conflicts with main function**

Grant Clark committed on Apr 29
Resolving conflicts

<table>
<thead>
<tr>
<th>Grant Clark committed on Apr 29</th>
</tr>
</thead>
<tbody>
<tr>
<td>Added logging and Multi-threading</td>
</tr>
<tr>
<td>Grant Clark committed on Apr 29</td>
</tr>
</tbody>
</table>

Commits on Apr 26, 2019

| Merge pull request #2 from Joshandrews43/aroddick_server | aroddick committed on Apr 26 |

Commits on Apr 25, 2019

| Attempt at multi-threading |
| Grant Clark committed on Apr 25 |

Commits on Apr 24, 2019

| removed junit from script |
| aroddick committed on Apr 24 |

| Added TempMain class, Deleted extraneous files, modified Miner to non... |
| Grant Clark committed on Apr 24 |

| Changed post request algorithm and removed dummy scraper |
| aroddick committed on Apr 24 |

Commits on Apr 24, 2019

| Beginning conversion of Miner class to non-static |
Commits on Apr 22, 2019

Updated algorithm to check if the client and server has same number of... ...
arooddick committed on Apr 22

Created winter2019.json
arooddick committed on Apr 22

added script
arooddick committed on Apr 22

changed format of general.json
arooddick committed on Apr 22

updated request for an array of quarters and added response with gene... ...
arooddick committed on Apr 22

Changed layout of general.json
arooddick committed on Apr 22

Finished Miner_v01
Grant Clark committed on Apr 22

Updated method to compare test file with post request and return upda... ...
Grant Clark committed on Apr 24
Aroddick committed on Apr 22

Added dummy json files to work with
e804877

Aroddick committed on Apr 22

Most of Miner class implemented
f626ca

Grant Clark committed on Apr 22

Changed package name
62a09f7

Aroddick committed on Apr 22

removed json from import
3864a90

Aroddick committed on Apr 22

forcing spring boot to use gson http message converter
0b767f0

Aroddick committed on Apr 22

added gson to dependencies
155824f

Aroddick committed on Apr 22

initial commit in new branch with server skeleton
85fdde3

Aroddick committed on Apr 22

Aroddick committed on Apr 22

Commits on Apr 18, 2019

Added chromedriver + Modified Miner and MinerTest classes

Grant Clark committed on Apr 18

a3874af
Commits on Apr 17, 2019

**Added Miner and MinerTest classes; Modified pom.xml to add dependencies**

Grant Clark committed on Apr 17

**new branch: gclark_miner**

Grant Clark committed on Apr 17

**merge**

Joshandrews43 committed on Apr 17

**Merge branch ‘master’ of https://github.com/Joshandrews43/cs48-schedule...**

Joshandrews43 committed on Apr 17

**initial commit**

Joshandrews43 committed on Apr 17

Commits on Apr 15, 2019

**Added the matrix with the languages we’re using**

aroddick committed on Apr 15

Commits on Apr 7, 2019

**Update README.md**

Joshandrews43 committed on Apr 7

**sample heroku app**

Joshandrews43 committed on Apr 7
<table>
<thead>
<tr>
<th>Message</th>
<th>Author</th>
<th>Date</th>
<th>Commit Hash</th>
</tr>
</thead>
<tbody>
<tr>
<td>Added the headings to the README file and linked the license</td>
<td>aroddick</td>
<td>Apr 3</td>
<td>d84f5bf</td>
</tr>
<tr>
<td>Changed the copyright year and name</td>
<td>aroddick</td>
<td>Apr 3</td>
<td>3597048</td>
</tr>
<tr>
<td>Merge pull request #1 from Joshandrews43/add-license-1</td>
<td>Joshandrews43</td>
<td>Apr 3</td>
<td>Verified 0a9bc29</td>
</tr>
<tr>
<td>Create LICENSE</td>
<td>Joshandrews43</td>
<td>Apr 3</td>
<td>Verified 1ad04cb</td>
</tr>
<tr>
<td>Create .travis.yml</td>
<td>Joshandrews43</td>
<td>Apr 3</td>
<td>Verified 46025d9</td>
</tr>
<tr>
<td>Update README.md</td>
<td>Joshandrews43</td>
<td>Apr 3</td>
<td>Verified 357a7ce</td>
</tr>
<tr>
<td>Initial commit</td>
<td>Joshandrews43</td>
<td>Apr 3</td>
<td></td>
</tr>
</tbody>
</table>