Class Scheduler Administrator Manual

1. Introduction

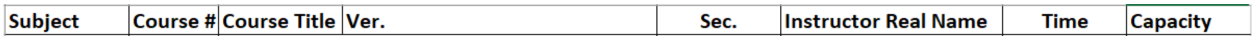
The Class Scheduler allows you to automatically assign classes to rooms based on where they are on campus vs how far away they are from the professor's office and their major's central building. It does this by taking in class and room data for the UMBC Campus and assigns classes to rooms based on their class's professor, subject, and capacity requirements. Our software is programmed in python and includes its own user friendly GUI that allows the user to submit xlsx data, generate a schedule, view the generated schedule, provide alternatives, view room usage, and report errors. Our program outputs the generated schedule into an xlsx file for future use.

2. System Overview

2.1 Background

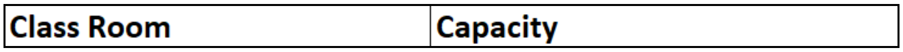
The system administrator(admin) is in charge of maintaining the system and making sure that user issues get solved in a timely manner. They must track warnings and check comments left from the users as the program is used. To run the program, an input file must be created of necessary classes and available rooms. The input file should follow the format of a sample file included with the program. The program will then take in this file and a name for the output file in order to generate the schedule. The scheduler takes into account the room requirements of the course and a room's proximity to the professor's office in order to minimize the travel distance for the professor. Once the schedule is generated, the user can view the scheduled courses and look at alternative time slots for the courses that could not be scheduled. This information is stored in an output file whose name is specified by the user. The scheduler also generates a file that contains statistics about the schedule, such as how many hours a room is in use. This covers the day to day operation of the program.

As for maintenance, the system admin has a few different responsibilities starting with making sure that users are properly formatting the input files. The input and output files should both be excel workbooks (.xlsx extension). The input file should contain three sheets: Schedule, Capacity, and Coords. Schedule should have columns in this format:



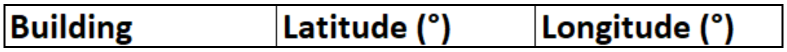
Subject	Course #	Course Title	Ver.	Sec.	Instructor Real Name	Time	Capacity

Capacity should have columns in this format:



Class Room	Capacity

Coords should have columns in this format:



Building	Latitude (°)	Longitude (°)

Additionally, the admin should be checking the "user_comments.csv" file that is generated when comments have been submitted by users to make sure any logged issues are resolved. Along with user comments, the admin should be checking the warnings file. This is where the program will inform the user of any issues including any system and user errors.

2.2 Hardware and Software Requirements

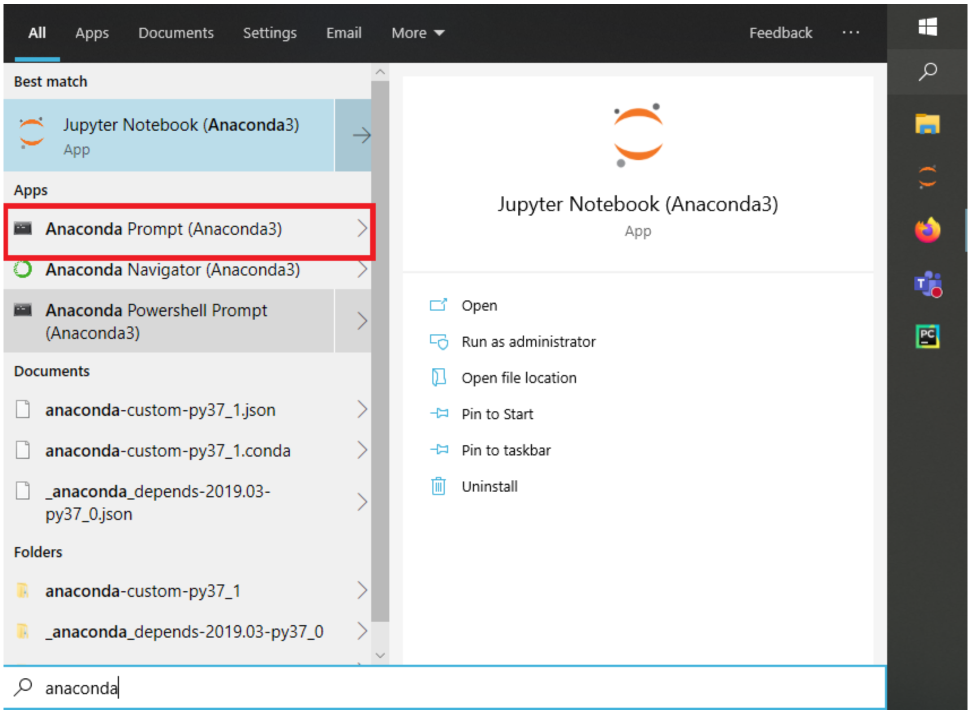
The system must :

- Have an operating system of Windows 8 or newer, 64-bit macOS 10.13+, or Linux, including Ubuntu, RedHat, CentOS 6+, and others.
- Have a system architecture of Windows- 64-bit x86, 32-bit x86; MacOS- 64-bit x86; Linux- 64-bit x86, 64-bit Power8/Power9.
- Minimum 5 GB disk space to download and install(if downloading the recommended python environment Anaconda).
- Be able to run python3 with all required modules.
- Have a window manager to utilize the GUI

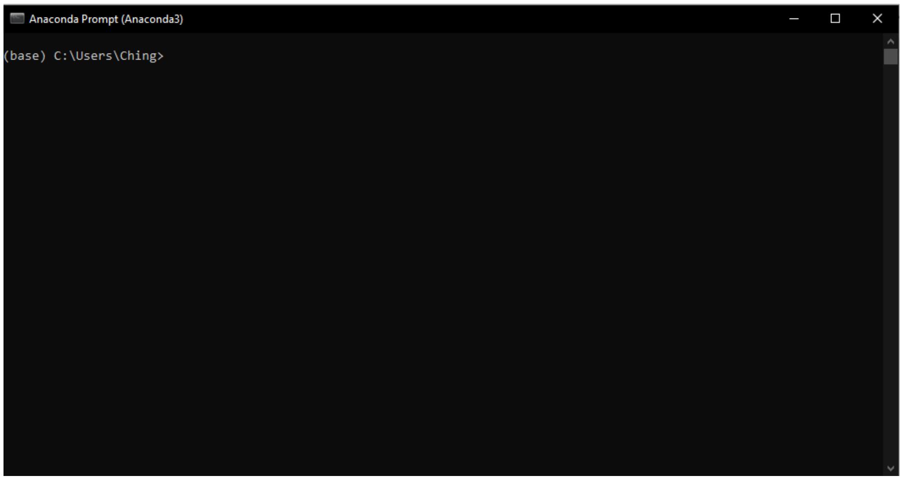
3. Administrative Procedures

3.1 Installation:

- Download a Python3 environment, we recommend Anaconda, but you're welcome to download any Python3 environment: https://docs.anaconda.com/anaconda/install/
- **Required Libraries:** pandas, matplotlib, xlsxwriter, tkinter, sys, os, time, numpy, and daytime, pillow, copy
    - Once Anaconda completes its installation, install the libraries by opening up the Anaconda prompt, which you can access by searching up Anaconda on the Windows search bar.

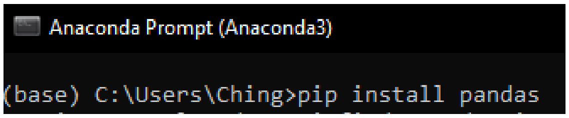

All
Apps
Documents
Settings
Email
More
Feedback
...
Best match
Jupyter Notebook (Anaconda3)
App
Apps
Anaconda Prompt (Anaconda3)
Anaconda Navigator (Anaconda3)
Anaconda Powershell Prompt (Anaconda3)
Documents
anaconda-custom-py37_1.json
anaconda-custom-py37_1.conda
_anaconda_depends-2019.03-py37_0.json
Folders
anaconda-custom-py37_1
_anaconda_depends-2019.03-py37_0
Jupyter Notebook (Anaconda3)
App
Open
Run as administrator
Open file location
Pin to Start
Pin to taskbar
Uninstall
anaconda

- After clicking on the option, the following prompt should open up:



```
(base) C:\Users\Ching>
```

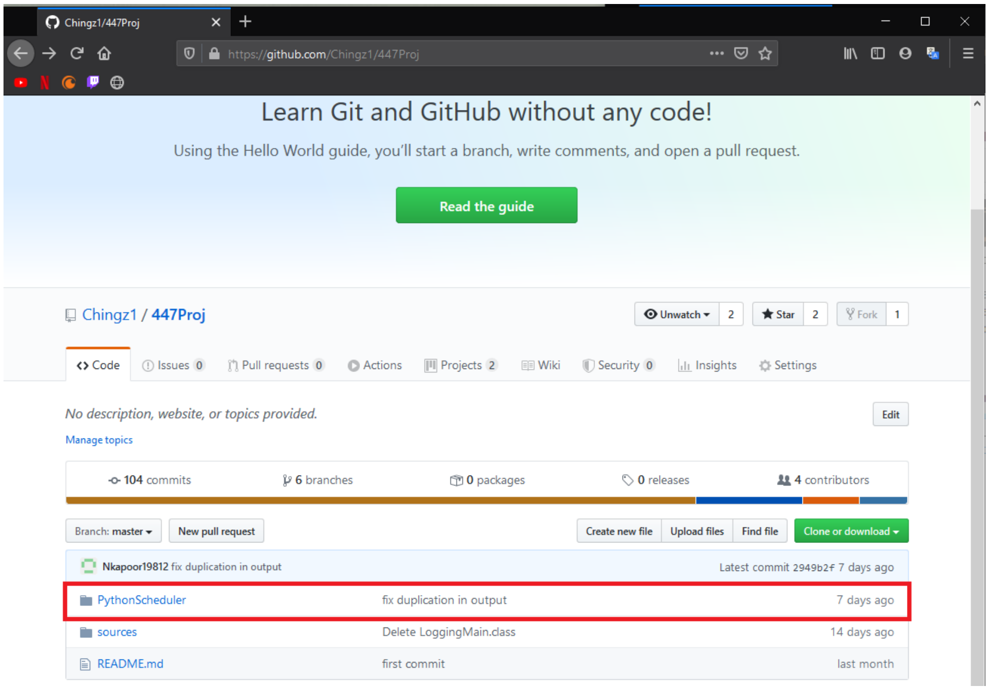
- On the command line type the follow command type: “pip install pandas”



```
(base) C:\Users\Ching>pip install pandas
```

- Perform the same command for the other required libraries, mentioned above, in the same format: pip install *library_name*

- After successfully installing all of the required libraries, download our scheduling software from Github and save the files to a location that you can easily access.
    - The link to our Github: https://github.com/Chingz1/447Proj
    - Download all of the contents in the PythonScheduler folder



Learn Git and GitHub without any code!

Using the Hello World guide, you'll start a branch, write comments, and open a pull request.

Chingz1 / 447Proj

No description, website, or topics provided.

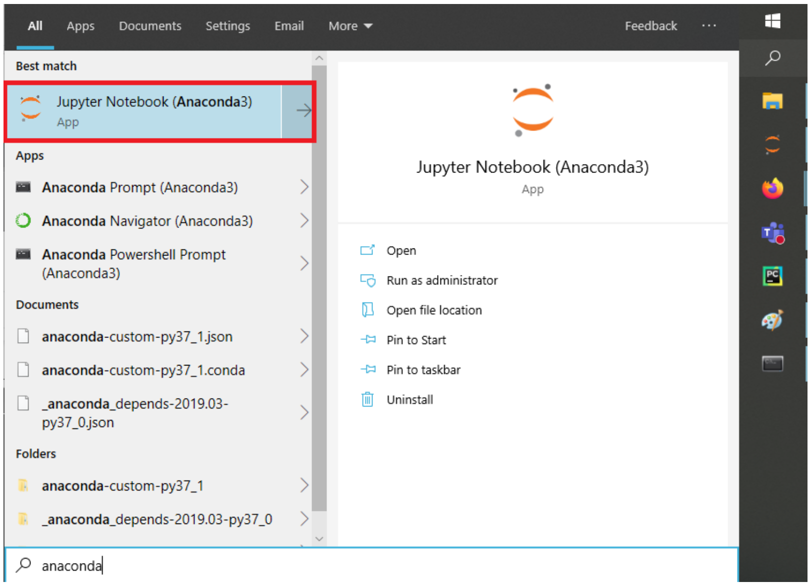
104 commits · 6 branches · 0 packages · 0 releases · 4 contributors

Nkapoor19812 fix duplication in output · Latest commit 2949b2f 7 days ago

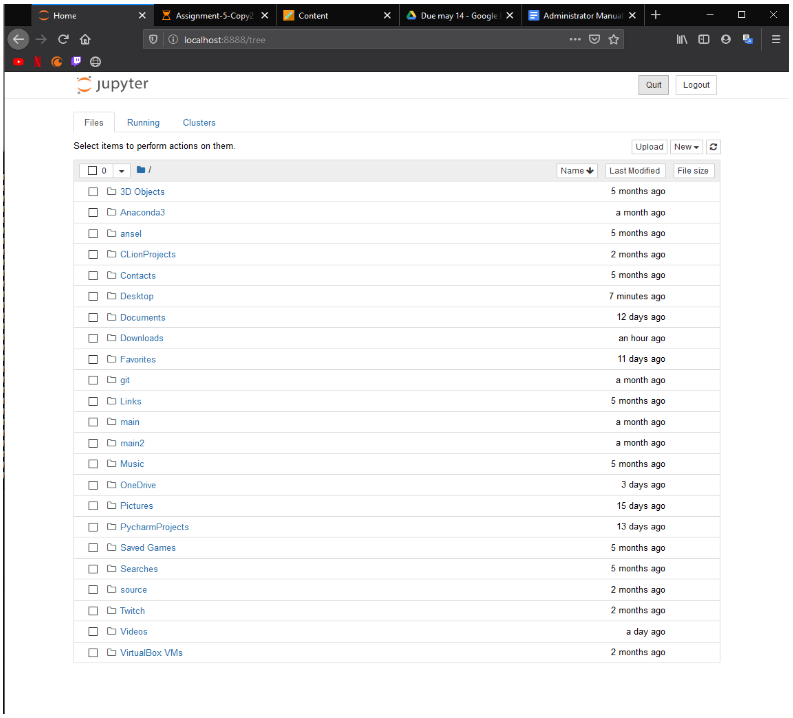
PythonScheduler	fix duplication in output	7 days ago
sources	Delete LoggingMain.class	14 days ago
README.md	first commit	last month

3.2 Routine Task:

- Once the scheduling software is downloaded, open up Jupyter Notebook which comes prepackaged with Anaconda. Jupyter Notebook can be accessed in the same manner as Anaconda, simply search it up on the Windows search bar.
  - Here is a screenshot example:


All
Apps
Documents
Settings
Email
More
Feedback
Best match
Jupyter Notebook (Anaconda3)
App
Apps
Anaconda Prompt (Anaconda3)
Anaconda Navigator (Anaconda3)
Anaconda Powershell Prompt (Anaconda3)
Documents
anaconda-custom-py37_1.json
anaconda-custom-py37_1.conda
_anaconda_depends-2019.03-py37_0.json
Folders
anaconda-custom-py37_1
_anaconda_depends-2019.03-py37_0
Jupyter Notebook (Anaconda3)
App
Open
Run as administrator
Open file location
Pin to Start
Pin to taskbar
Uninstall
anaconda

  - Open Jupyter notebook is opened, a browser page should open up displaying an interface similar to the one seen below:

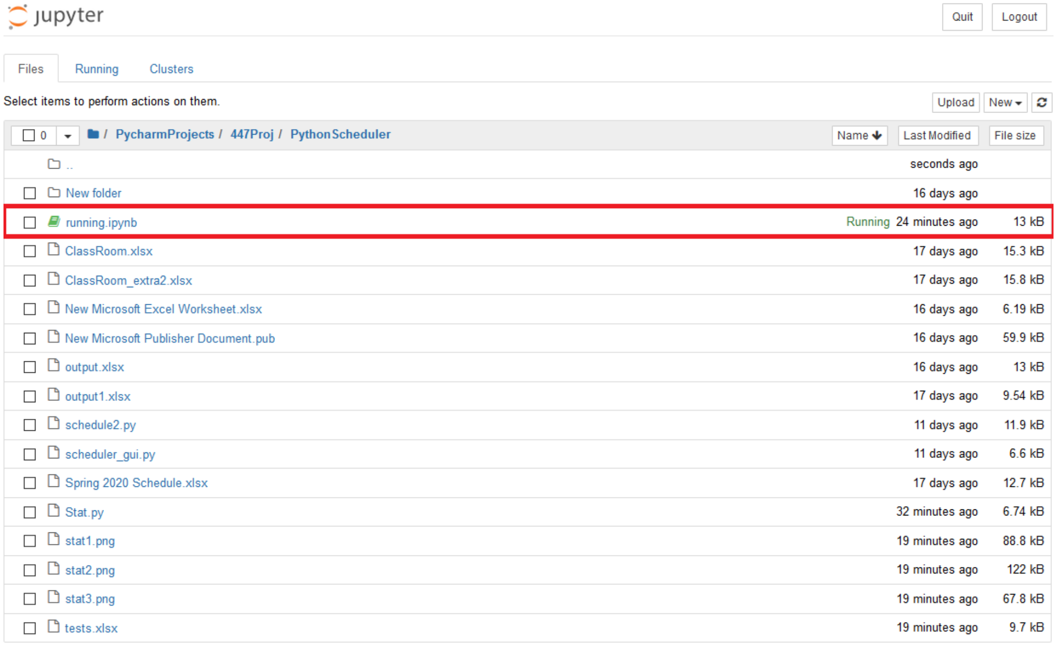


Select items to perform actions on them.

Name	Last Modified	File size
3D Objects	5 months ago	
Anaconda3	a month ago	
ansel	5 months ago	
CLionProjects	2 months ago	
Contacts	5 months ago	
Desktop	7 minutes ago	
Documents	12 days ago	
Downloads	an hour ago	
Favorites	11 days ago	
git	a month ago	
Links	5 months ago	
main	a month ago	
main2	a month ago	
Music	5 months ago	
OneDrive	3 days ago	
Pictures	15 days ago	
PycharmProjects	13 days ago	
Saved Games	5 months ago	
Searches	5 months ago	
source	2 months ago	
Twitch	2 months ago	
Videos	a day ago	
VirtualBox VMs	2 months ago	

  - Then find the location to where you saved the scheduler software you downloaded from Github. By default it should be under the *Downloads* folder.

- The folder should contain all of the files from our Github repository, if not return to the Github page and redownload the folder. An example of the folder's content is seen here:

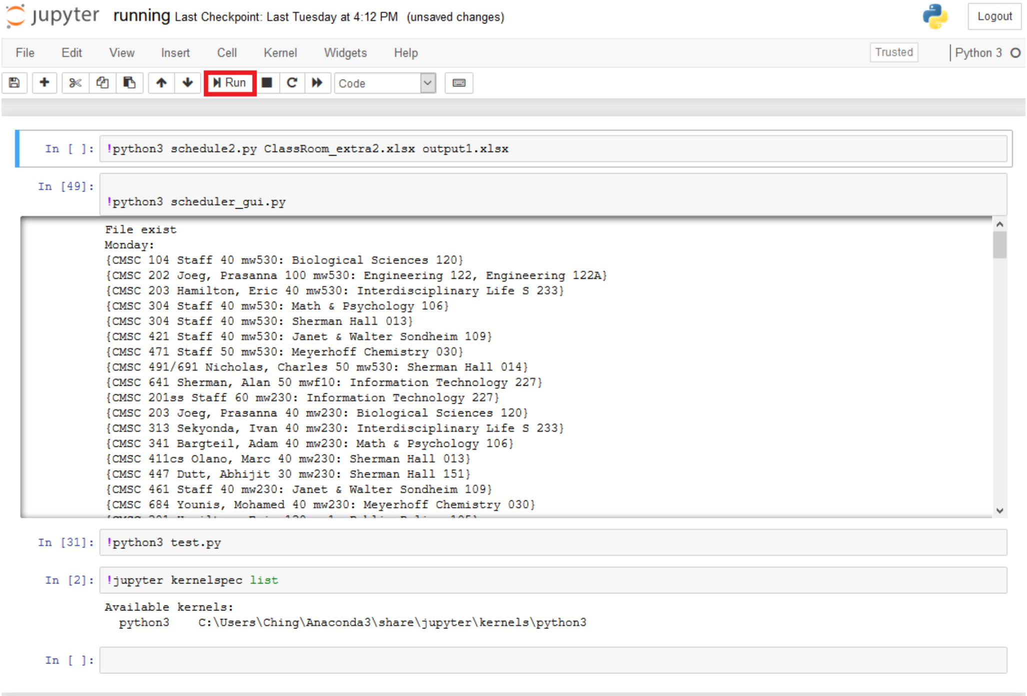


Select items to perform actions on them.

□ 0 / PycharmProjects / 447Proj / PythonScheduler

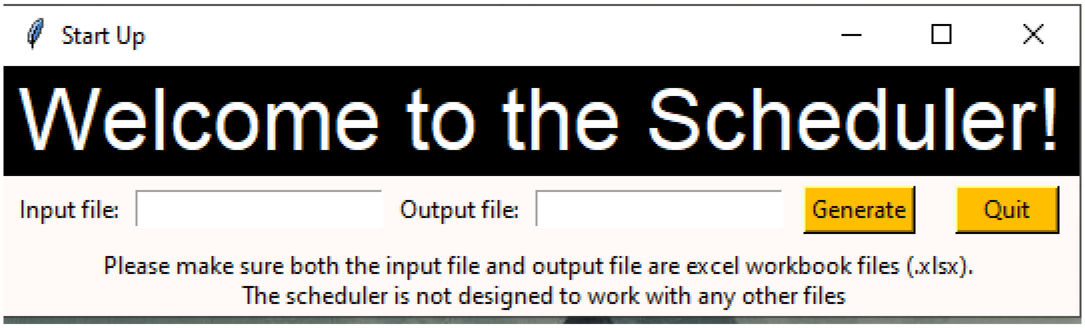
	Name	Status	Last Modified	File size
	..		seconds ago	
□	New folder		16 days ago	
□	running.ipynb	Running	24 minutes ago	13 kB
□	ClassRoom.xlsx		17 days ago	15.3 kB
□	ClassRoom_extra2.xlsx		17 days ago	15.8 kB
□	New Microsoft Excel Worksheet.xlsx		16 days ago	6.19 kB
□	New Microsoft Publisher Document.pub		16 days ago	59.9 kB
□	output.xlsx		16 days ago	13 kB
□	output1.xlsx		17 days ago	9.54 kB
□	schedule2.py		11 days ago	11.9 kB
□	scheduler_gui.py		11 days ago	6.6 kB
□	Spring 2020 Schedule.xlsx		17 days ago	12.7 kB
□	Stat.py		32 minutes ago	6.74 kB
□	stat1.png		19 minutes ago	88.8 kB
□	stat2.png		19 minutes ago	122 kB
□	stat3.png		19 minutes ago	67.8 kB
□	tests.xlsx		19 minutes ago	9.7 kB

- Once you're inside the folder, open up the running.ipynb file and click run. The first cell should output the generated schedule. Then click on run again to bring up the software's GUI.



```
In [ ]: !python3 schedule2.py ClassRoom_extra2.xlsx output1.xlsx

In [49]:
        !python3 scheduler_gui.py

        File exist
        Monday:
        {CMSC 104 Staff 40 mw530: Biological Sciences 120}
        {CMSC 202 Joeg, Prasanna 100 mw530: Engineering 122, Engineering 122A}
        {CMSC 203 Hamilton, Eric 40 mw530: Interdisciplinary Life S 233}
        {CMSC 304 Staff 40 mw530: Math & Psychology 106}
        {CMSC 304 Staff 40 mw530: Sherman Hall 013}
        {CMSC 421 Staff 40 mw530: Janet & Walter Sondheim 109}
        {CMSC 471 Staff 50 mw530: Meyerhoff Chemistry 030}
        {CMSC 491/691 Nicholas, Charles 50 mw530: Sherman Hall 014}
        {CMSC 641 Sherman, Alan 50 mwf10: Information Technology 227}
        {CMSC 201ss Staff 60 mw230: Information Technology 227}
        {CMSC 203 Joeg, Prasanna 40 mw230: Biological Sciences 120}
        {CMSC 313 Sekyonda, Ivan 40 mw230: Interdisciplinary Life S 233}
        {CMSC 341 Bargteil, Adam 40 mw230: Math & Psychology 106}
        {CMSC 411cs Olano, Marc 40 mw230: Sherman Hall 013}
        {CMSC 447 Dutt, Abhijit 30 mw230: Sherman Hall 151}
        {CMSC 461 Staff 40 mw230: Janet & Walter Sondheim 109}
        {CMSC 684 Younis, Mohamed 40 mw230: Meyerhoff Chemistry 030}

In [31]: !python3 test.py

In [2]: !jupyter kernelspec list

        Available kernels:
          python3    C:\Users\Ching\Anaconda3\share\jupyter\kernels\python3

In [ ]:
```

- The scheduler's GUI should pop up prompting the user to enter the input file and output file's name.



Please make sure both the input file and output file are excel workbook files (.xlsx).
The scheduler is not designed to work with any other files

  - Enter the name of your input file in the directed field with the extension .xlsx, so for example for the file *myfile*, you would enter *myfile.xlsx* into the field. Then do the same for the output file field with the same .xlsx requirement. After both fields are filled on click on generate.
  - After the schedule generates, there should be two main options: view schedule and view statistics. Choose the option that best suits your task. If the view schedule or view statistic options output empty data, please recheck if you've made an error when entering the filename. If issues persist, please refer to the troubleshooting section.

- 3.3 Periodic Administration
  - The software will generate user comments in the "user_comments.csv" file. Every now and then, the developers should address them. Once they're addressed, the comments should be deleted from the file.
- 3.4 User Support
  - The software has a contract support function where a user will be allowed to send a message regarding their issues to the developers. More information will be provided below in the troubleshooting section.

4. Troubleshooting

4.1 Common Error Messages:

Error messages are written in "Warnings.txt". The most common error messages show up when a class's subject in the input data is not recognized. This may occur if the subject was misspelled, and you can check what class has the misspelling by checking the warning file. If this error occurs, the class will still be put in the schedule, however it will likely end up getting a lower priority than other classes. If a subject is spelled correctly, then it needs to be mapped to a building by changing a building's subject to the unrecognized subject.

If the user has another serious error message or failure, we implemented a feature that allows the user to report it. On the GUI, there is a button to 'Contact Us'. The 'Contact Us' button will allow the user to send a message with their comment describing their error that developers can later review.

4.2 Limitations:

One limitation of our project is we do not authenticate the users. We did not implement this feature because it was not required but it does limit the security of our application. This will not affect the end user’s experience of our application but it does introduce the possibility of vulnerabilities.

Another limitation is that only one subject can currently be mapped to a building, so you can only have as many subjects as buildings. This specific location of this limitation is in schedule2.py. This limitation could not be fixed because of time constraints. This will affect the end user if they want to extend the scheduling program to have more subjects.

The user of the program needs to specify what building a professor should be in as well as what building a subject goes. Otherwise, the weight generation function will not be able to determine a proper weight and the resulting schedule may not be optimal.

Appendix A - Document Contributions

Introduction: Dylan
System Overview: Anna and Nimish
Administrative Procedures: Ching
Troubleshooting: Kaya