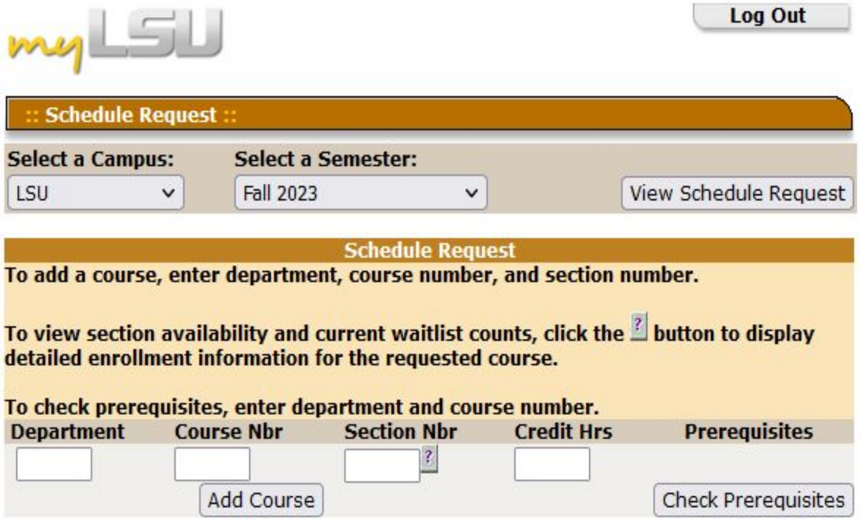
G20 Scheduler

Preston Saxon, Alexander Leake,
Jacob Rogers, Joel Rogers, Vivian San

Main Overview

- LSU's current scheduling system is a hassle to work with
- Have to use multiple sources of information on mylsu just to know what to schedule (degree audit, personal schedule, schedule booklet, schedule request, course flowcharts)
- Provides minimal information on classes themselves
- The G20 Scheduler will condense the scheduling process into a single interface that is both convenient, and efficient for students



:: Schedule Request ::

Select a Campus: LSU

Select a Semester: Fall 2023

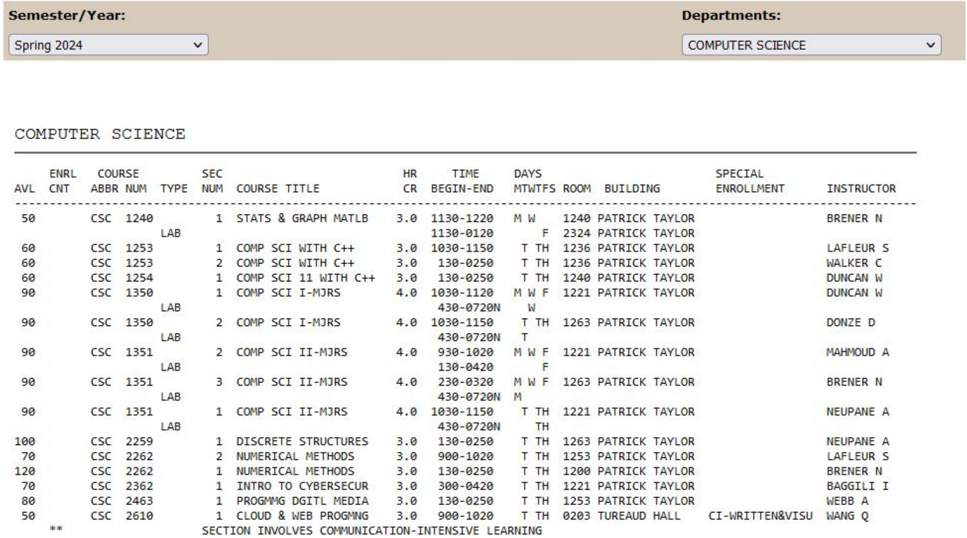
Schedule Request

To add a course, enter department, course number, and section number.

To view section availability and current waitlist counts, click the ? button to display detailed enrollment information for the requested course.

To check prerequisites, enter department and course number.

Department	Course Nbr	Section Nbr	Credit Hrs	Prerequisites



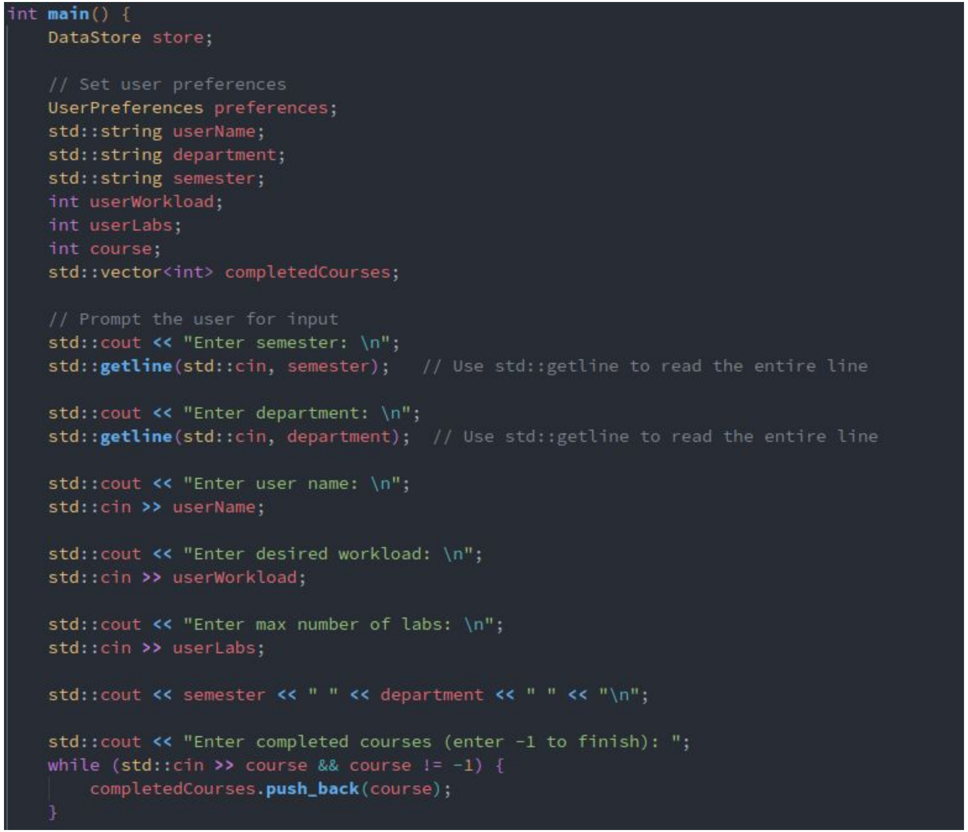
Semester/Year: Spring 2024

Departments: COMPUTER SCIENCE

```
COMPUTER SCIENCE
____________________________________________________________________________________________________________

     ENRL  COURSE          SEC                        HR    TIME       DAYS                        SPECIAL
AVL  CNT   ABBR NUM  TYPE  NUM  COURSE TITLE          CR  BEGIN-END   MTWTFS ROOM  BUILDING        ENROLLMENT      INSTRUCTOR
------------------------------------------------------------------------------------------------------------------------------
 50        CSC  1240          1  STATS & GRAPH MATLB   3.0  1130-1220  M W    1240 PATRICK TAYLOR                   BRENER N
                     LAB                                    1130-0120      F  2324 PATRICK TAYLOR
 60        CSC  1253          1  COMP SCI WITH C++     3.0  1030-1150   T TH  1236 PATRICK TAYLOR                   LAFLEUR S
 60        CSC  1253          2  COMP SCI WITH C++     3.0   130-0250   T TH  1236 PATRICK TAYLOR                   WALKER C
 60        CSC  1254          1  COMP SCI 11 WITH C++  3.0   130-0250   T TH  1240 PATRICK TAYLOR                   DUNCAN W
 90        CSC  1350          1  COMP SCI I-MJRS       4.0  1030-1120  M W F  1221 PATRICK TAYLOR                   DUNCAN W
                     LAB                                     430-0720N   W
 90        CSC  1350          2  COMP SCI I-MJRS       4.0  1030-1150   T TH  1263 PATRICK TAYLOR                   DONZE D
                     LAB                                     430-0720N  T
 90        CSC  1351          2  COMP SCI II-MJRS      4.0   930-1020  M W F  1221 PATRICK TAYLOR                   MAHMOUD A
                     LAB                                     130-0420      F
 90        CSC  1351          3  COMP SCI II-MJRS      4.0   230-0320  M W F  1263 PATRICK TAYLOR                   BRENER N
                     LAB                                     430-0720N M
 90        CSC  1351          1  COMP SCI II-MJRS      4.0  1030-1150   T TH  1221 PATRICK TAYLOR                   NEUPANE A
                     LAB                                     430-0720N    TH
100        CSC  2259          1  DISCRETE STRUCTURES   3.0   130-0250   T TH  1263 PATRICK TAYLOR                   NEUPANE A
 70        CSC  2262          2  NUMERICAL METHODS     3.0   900-1020   T TH  1253 PATRICK TAYLOR                   LAFLEUR S
120        CSC  2262          1  NUMERICAL METHODS     3.0   130-0250   T TH  1200 PATRICK TAYLOR                   BRENER N
 70        CSC  2362          1  INTRO TO CYBERSECUR   3.0   300-0420   T TH  1221 PATRICK TAYLOR                   BAGGILI I
 80        CSC  2463          1  PROGMMG DGITL MEDIA   3.0   130-0250   T TH  1253 PATRICK TAYLOR                   WEBB A
 50        CSC  2610          1  CLOUD & WEB PROGMNG   3.0   900-1020   T TH  0203 TUREAUD HALL   CI-WRITTEN&VISU  WANG Q
     **                    SECTION INVOLVES COMMUNICATION-INTENSIVE LEARNING
```

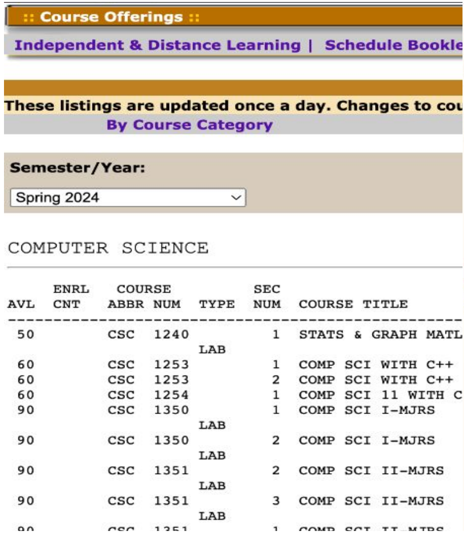
Main Files & Technology Used

- **generatedata.py**
  - Scrapes data from LSU course mainframe using python library BeautifulSoup for easy parsing of html using lxml
  - Appends course data, course metadata, and degree audit data into .json files for the engine to use
- **project.cpp**
  - Asks users for their semester, department, username, max workload, max amount of labs, and already completed courses
  - Embedded python used to call generatedata.py with user variables



```
int main() {
    DataStore store;

    // Set user preferences
    UserPreferences preferences;
    std::string userName;
    std::string department;
    std::string semester;
    int userWorkload;
    int userLabs;
    int course;
    std::vector<int> completedCourses;

    // Prompt the user for input
    std::cout << "Enter semester: \n";
    std::getline(std::cin, semester);   // Use std::getline to read the entire line

    std::cout << "Enter department: \n";
    std::getline(std::cin, department);  // Use std::getline to read the entire line

    std::cout << "Enter user name: \n";
    std::cin >> userName;

    std::cout << "Enter desired workload: \n";
    std::cin >> userWorkload;

    std::cout << "Enter max number of labs: \n";
    std::cin >> userLabs;

    std::cout << semester << " " << department << " " << "\n";

    std::cout << "Enter completed courses (enter -1 to finish): ";
    while (std::cin >> course && course != -1) {
        completedCourses.push_back(course);
    }
```

Integrations



:: Course Offerings ::

Independent & Distance Learning | Schedule Bookle

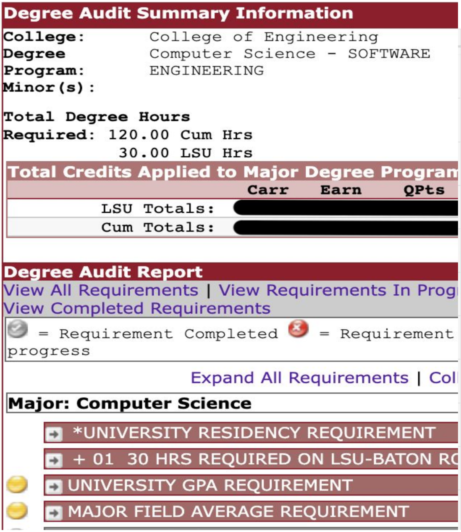
These listings are updated once a day. Changes to cou

By Course Category

Semester/Year:

Spring 2024

```
COMPUTER SCIENCE

     ENRL    COURSE             SEC
AVL  CNT   ABBR  NUM   TYPE    NUM    COURSE TITLE
--------------------------------------------------------
 50        CSC   1240            1    STATS & GRAPH MATL
                       LAB
 60        CSC   1253            1    COMP SCI WITH C++
 60        CSC   1253            2    COMP SCI WITH C++
 60        CSC   1254            1    COMP SCI 11 WITH C
 90        CSC   1350            1    COMP SCI I-MJRS
                       LAB
 90        CSC   1350            2    COMP SCI I-MJRS
                       LAB
 90        CSC   1351            2    COMP SCI II-MJRS
                       LAB
 90        CSC   1351            3    COMP SCI II-MJRS
                       LAB
 90        CSC   1351            1    COMP SCI II-MJRS
```



Degree Audit Summary Information

```
College:       College of Engineering
Degree         Computer Science - SOFTWARE
Program:       ENGINEERING
Minor(s):

Total Degree Hours
Required: 120.00 Cum Hrs
           30.00 LSU Hrs
```

Total Credits Applied to Major Degree Program

	Carr	Earn	QPts
LSU Totals:	[redacted]		
Cum Totals:	[redacted]		

Degree Audit Report

View All Requirements | View Requirements In Prog

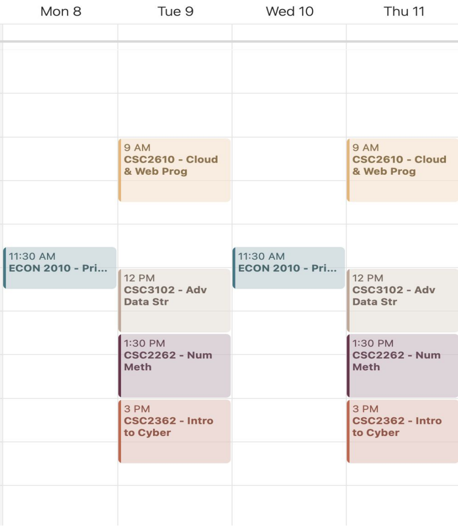
View Completed Requirements

● = Requirement Completed ☒ = Requirement progress

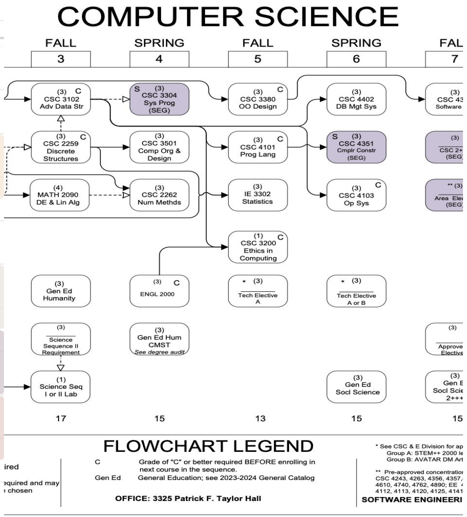
Expand All Requirements | Col

Major: Computer Science

- *UNIVERSITY RESIDENCY REQUIREMENT
- + 01 30 HRS REQUIRED ON LSU-BATON RC
- ● UNIVERSITY GPA REQUIREMENT
- ● MAJOR FIELD AVERAGE REQUIREMENT



Mon 8	Tue 9	Wed 10	Thu 11
	9 AM CSC2610 - Cloud & Web Prog		9 AM CSC2610 - Cloud & Web Prog
11:30 AM ECON 2010 - Pri...		11:30 AM ECON 2010 - Pri...	
	12 PM CSC3102 - Adv Data Str		12 PM CSC3102 - Adv Data Str
	1:30 PM CSC2262 - Num Meth		1:30 PM CSC2262 - Num Meth
	3 PM CSC2362 - Intro to Cyber		3 PM CSC2362 - Intro to Cyber



FLOWCHART LEGEND

ired

equired and may
chosen

C Grade of "C" or better required BEFORE enrolling in next course in the sequence.

Gen Ed General Education; see 2023-2024 General Catalog

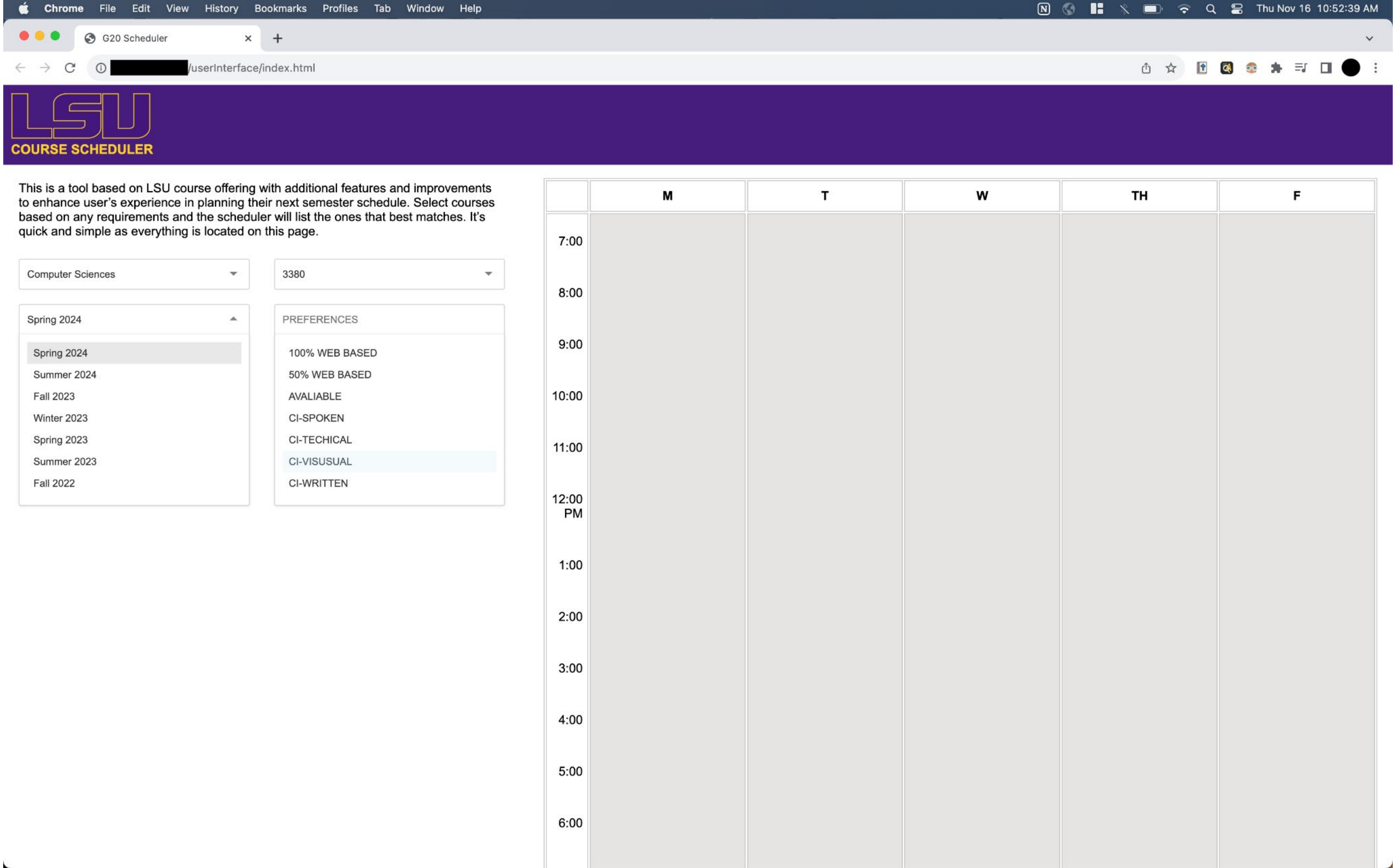
OFFICE: 3325 Patrick F. Taylor Hall

* See CSC & E Division for ap
Group A: STEM++ 2000 le
Group B: AVATAR DM Art

** Pre-approved concentratio
CSC 4243, 4263, 4356, 4357,
4610, 4740, 4762, 4890; EE
4112, 4113, 4120, 4125, 4141

SOFTWARE ENGINEERI

Course Offerings ✚ Degree Audit ✚ Calendar ✚ Flowchart



Chrome File Edit View History Bookmarks Profiles Tab Window Help
Thu Nov 16 10:52:39 AM
G20 Scheduler
[redacted]/userInterface/index.html
LSU
COURSE SCHEDULER

This is a tool based on LSU course offering with additional features and improvements to enhance user's experience in planning their next semester schedule. Select courses based on any requirements and the scheduler will list the ones that best matches. It's quick and simple as everything is located on this page.

Computer Sciences

3380

Spring 2024

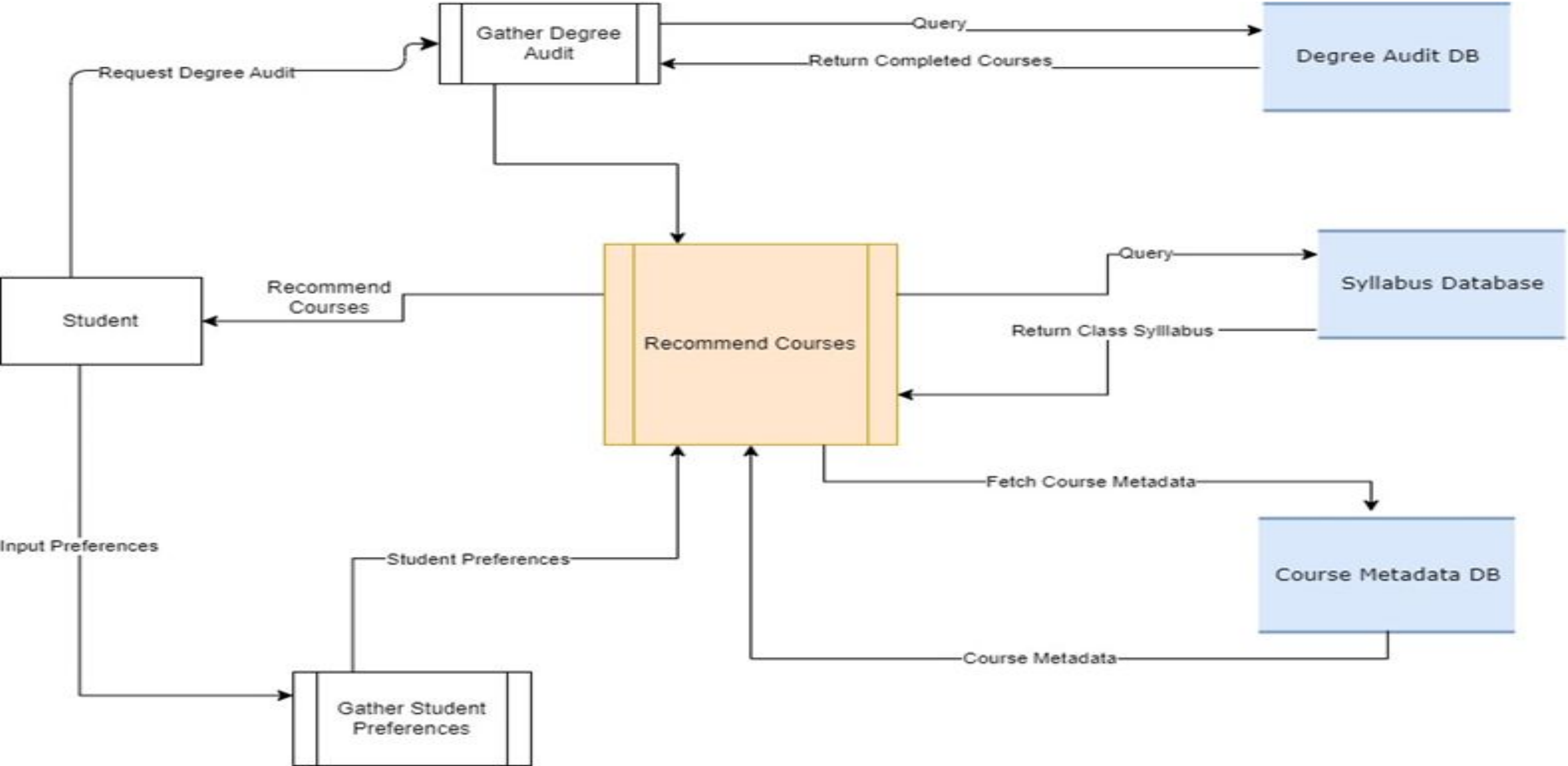
- Spring 2024
- Summer 2024
- Fall 2023
- Winter 2023
- Spring 2023
- Summer 2023
- Fall 2022

PREFERENCES

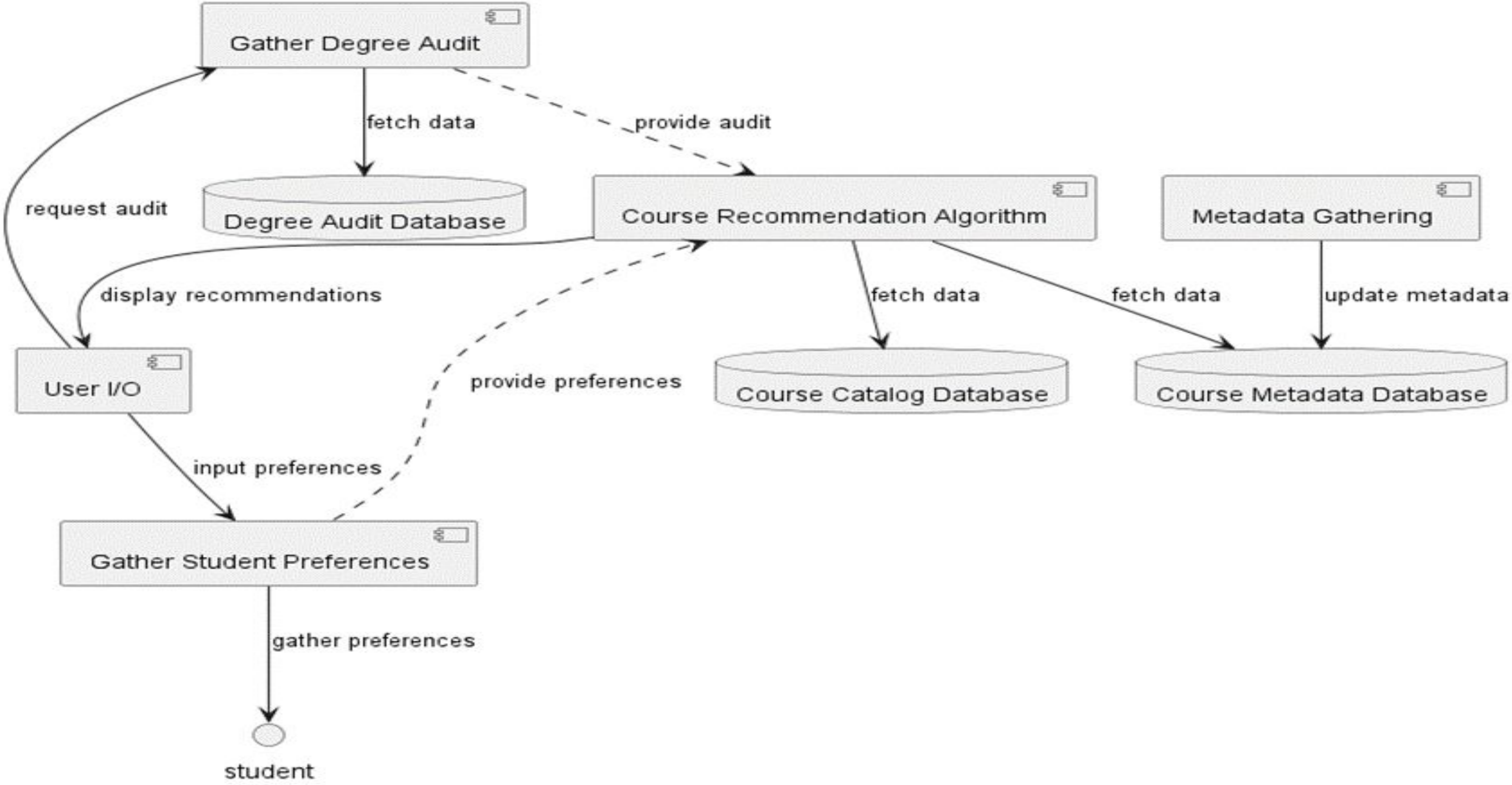
- 100% WEB BASED
- 50% WEB BASED
- AVALIABLE
- CI-SPOKEN
- CI-TECHICAL
- CI-VISUSUAL
- CI-WRITTEN

	M	T	W	TH	F
7:00					
8:00					
9:00					
10:00					
11:00					
12:00 PM					
1:00					
2:00					
3:00					
4:00					
5:00					
6:00					

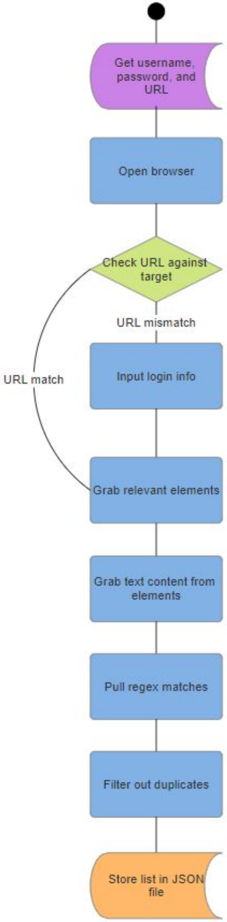
How will it work?



What it will do in practice



Preston


Get username, password, and URL

Open browser

Check URL against target

URL mismatch

Input login info

URL match

Grab relevant elements

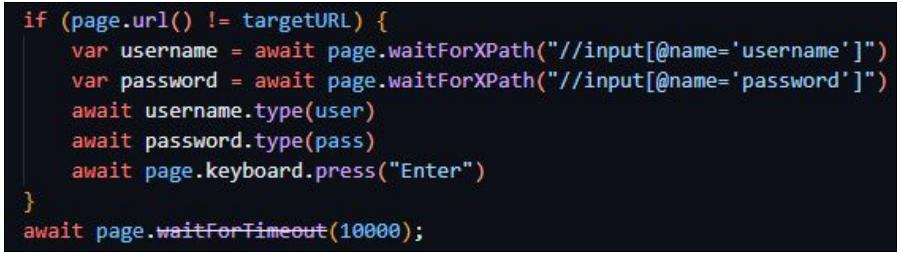
Grab text content from elements

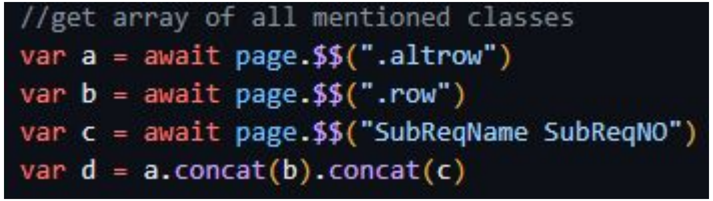
Pull regex matches

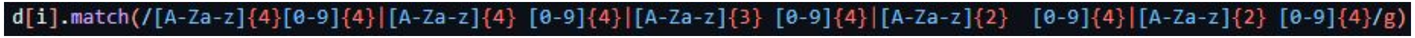
Filter out duplicates

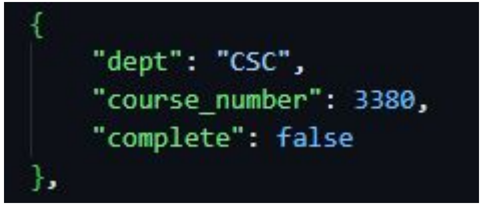
Store list in JSON file

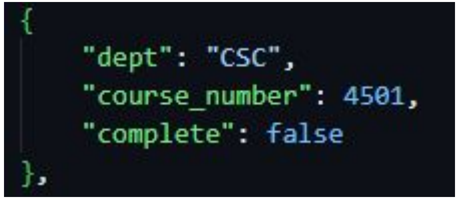
Puppeteer


```
if (page.url() != targetURL) {
    var username = await page.waitForXPath("//input[@name='username']")
    var password = await page.waitForXPath("//input[@name='password']")
    await username.type(user)
    await password.type(pass)
    await page.keyboard.press("Enter")
}
await page.~~waitForTimeout~~(10000);
```


```
//get array of all mentioned classes
var a = await page.$$(".altrow")
var b = await page.$$(".row")
var c = await page.$$("SubReqName SubReqNO")
var d = a.concat(b).concat(c)
```

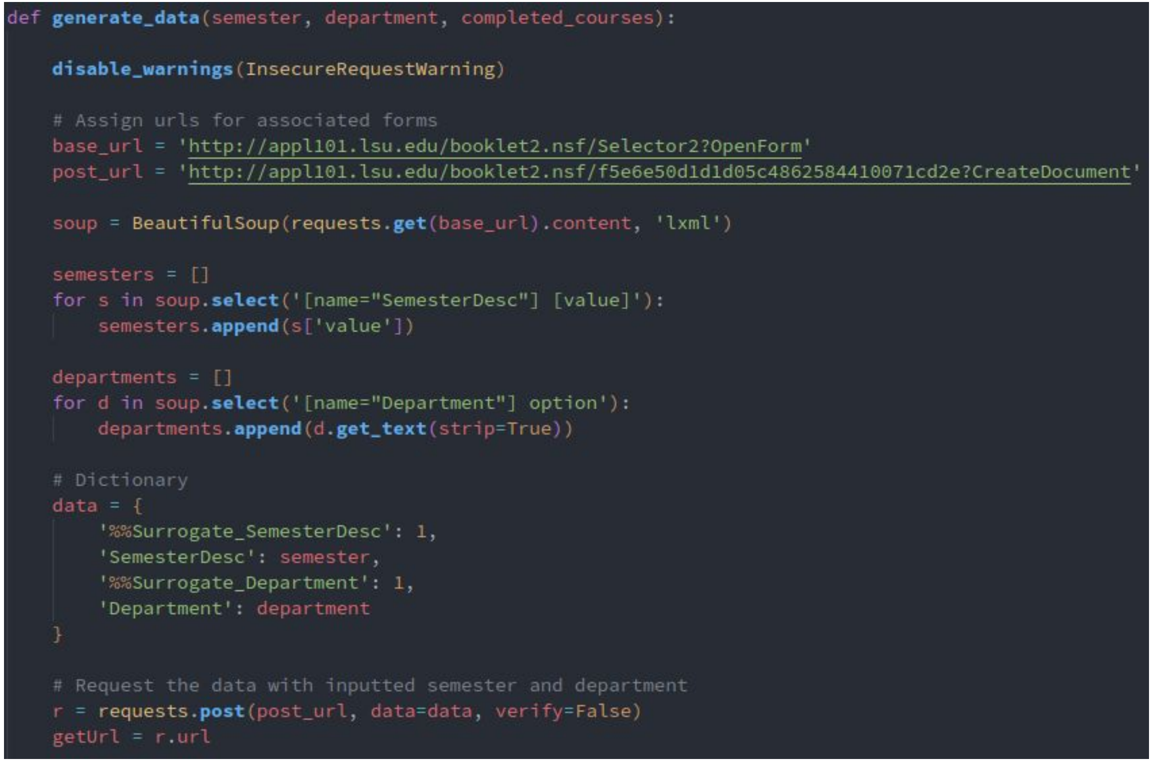

```
d[i].match(/[A-Za-z]{4}[0-9]{4}|[A-Za-z]{4} [0-9]{4}|[A-Za-z]{3} [0-9]{4}|[A-Za-z]{2}  [0-9]{4}|[A-Za-z]{2} [0-9]{4}/g)
```


```
{
    "dept": "CSC",
    "course_number": 3380,
    "complete": false
},
```


```
{
    "dept": "CSC",
    "course_number": 4501,
    "complete": false
},
```

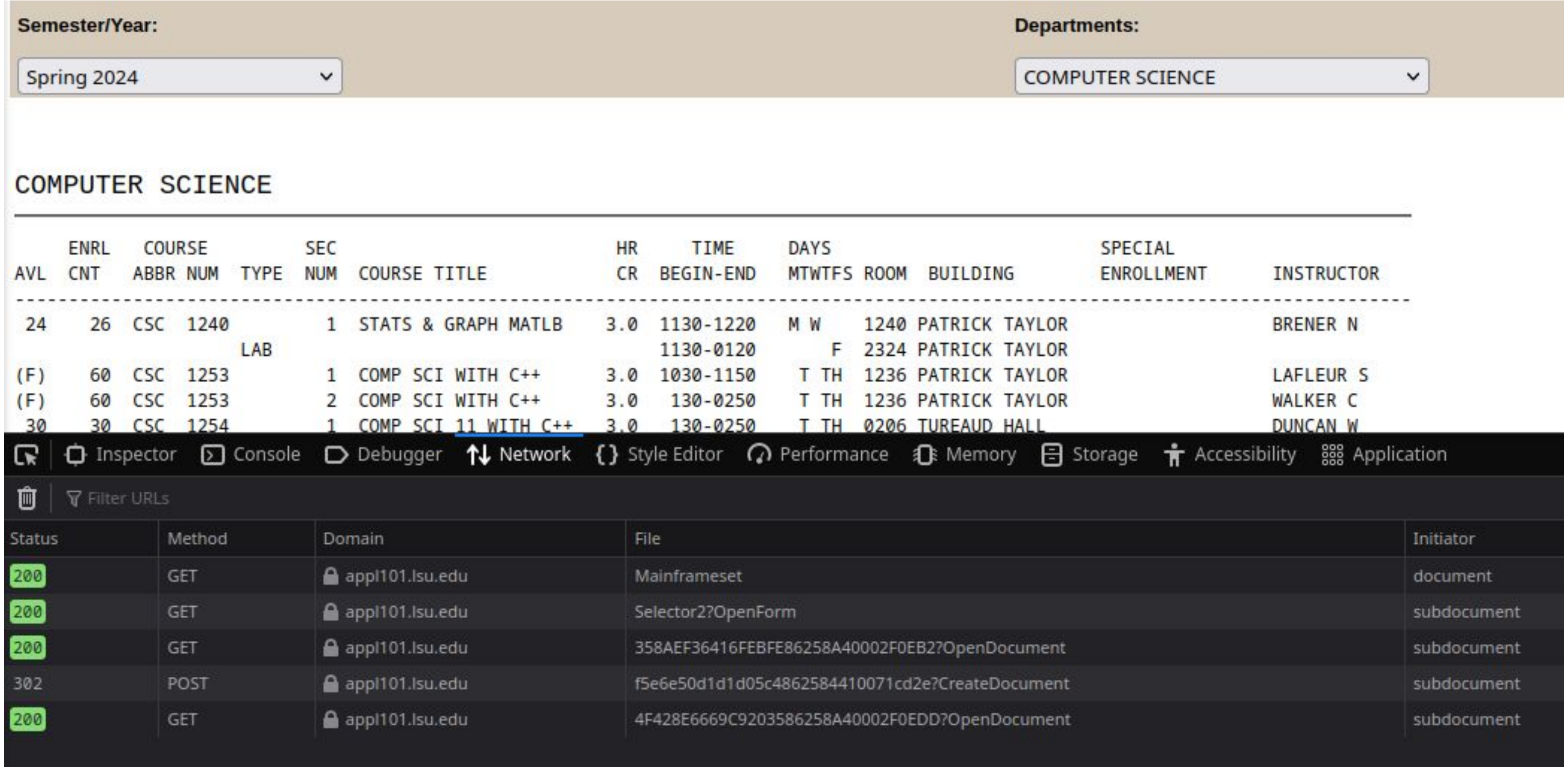
Jacob

- Wrote the data generator and embedded it within the main C++ project file
- Function `generate_data` takes 3 inputs:
    - string semester
    - string department
    - list completed_courses
- Outputs 3 files:
    - course_data.json
    - course_meta.json
    - degree_audit_data.json



```
def generate_data(semester, department, completed_courses):

    disable_warnings(InsecureRequestWarning)

    # Assign urls for associated forms
    base_url = 'http://appl101.lsu.edu/booklet2.nsf/Selector2?OpenForm'
    post_url = 'http://appl101.lsu.edu/booklet2.nsf/f5e6e50d1d1d05c4862584410071cd2e?CreateDocument'

    soup = BeautifulSoup(requests.get(base_url).content, 'lxml')

    semesters = []
    for s in soup.select('[name="SemesterDesc"] [value]'):
        semesters.append(s['value'])

    departments = []
    for d in soup.select('[name="Department"] option'):
        departments.append(d.get_text(strip=True))

    # Dictionary
    data = {
        '%%Surrogate_SemesterDesc': 1,
        'SemesterDesc': semester,
        '%%Surrogate_Department': 1,
        'Department': department
    }

    # Request the data with inputted semester and department
    r = requests.post(post_url, data=data, verify=False)
    getUrl = r.url
```

Jacob



Semester/Year: Spring 2024

Departments: COMPUTER SCIENCE

COMPUTER SCIENCE

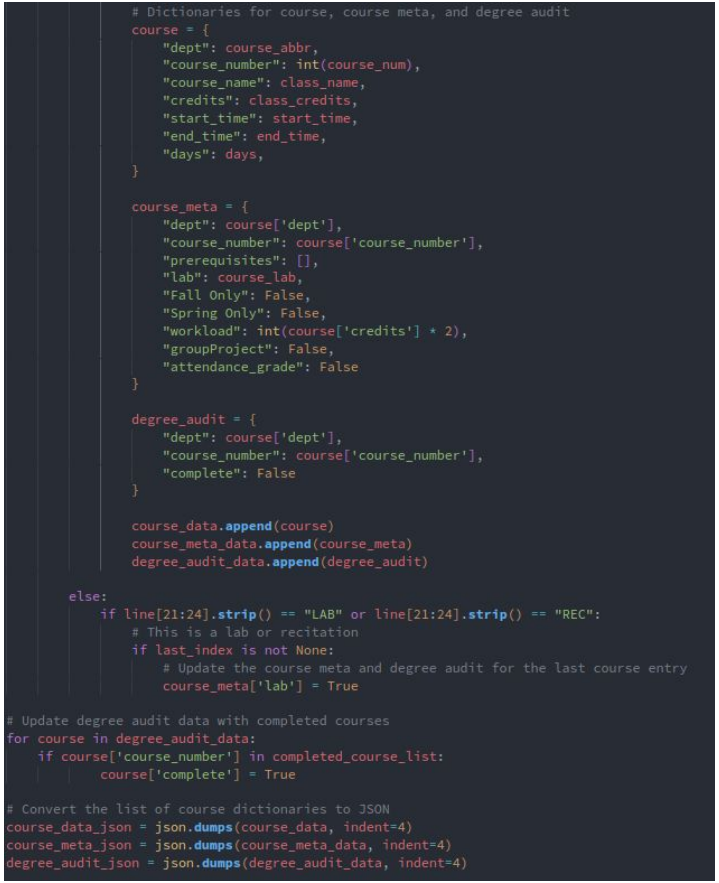
```
      ENRL   COURSE          SEC                             HR     TIME     DAYS                                SPECIAL
AVL   CNT   ABBR  NUM   TYPE  NUM  COURSE TITLE               CR   BEGIN-END  MTWTFS ROOM  BUILDING                ENROLLMENT      INSTRUCTOR
-------------------------------------------------------------------------------------------------------------------------------------------
 24    26   CSC   1240          1  STATS & GRAPH MATLB       3.0  1130-1220  M W    1240 PATRICK TAYLOR                          BRENER N
                        LAB                                        1130-0120      F 2324 PATRICK TAYLOR
(F)    60   CSC   1253          1  COMP SCI WITH C++         3.0  1030-1150   T TH  1236 PATRICK TAYLOR                          LAFLEUR S
(F)    60   CSC   1253          2  COMP SCI WITH C++         3.0   130-0250   T TH  1236 PATRICK TAYLOR                          WALKER C
 30    30   CSC   1254          1  COMP SCI 11 WITH C++      3.0   130-0250   T TH  0206 TUREAUD HALL                            DUNCAN W
```

Status	Method	Domain	File	Initiator
200	GET	appl101.lsu.edu	Mainframeset	document
200	GET	appl101.lsu.edu	Selector2?OpenForm	subdocument
200	GET	appl101.lsu.edu	358AEF36416FEBFE86258A40002F0EB2?OpenDocument	subdocument
302	POST	appl101.lsu.edu	f5e6e50d1d1d05c4862584410071cd2e?CreateDocument	subdocument
200	GET	appl101.lsu.edu	4F428E6669C9203586258A40002F0EDD?OpenDocument	subdocument

Example of hidden data requests made by the course catalog, in this example the POST request is made after selecting to display the courses for Computer Science in the semester Spring 2024.

Jacob

- Course data is sorted into dictionaries after whitespace parsing, and then appended to lists before being dumped into a .json
- Some classes have labs after the line they are split from, so the course meta data must be updated for the previous entry in course_meta
- If the completed_course_list has any classes within degree_audit, they are marked as completed



```
                # Dictionaries for course, course meta, and degree audit
                course = {
                    "dept": course_abbr,
                    "course_number": int(course_num),
                    "course_name": class_name,
                    "credits": class_credits,
                    "start_time": start_time,
                    "end_time": end_time,
                    "days": days,
                }

                course_meta = {
                    "dept": course['dept'],
                    "course_number": course['course_number'],
                    "prerequisites": [],
                    "lab": course_lab,
                    "Fall Only": False,
                    "Spring Only": False,
                    "workload": int(course['credits'] * 2),
                    "groupProject": False,
                    "attendance_grade": False
                }

                degree_audit = {
                    "dept": course['dept'],
                    "course_number": course['course_number'],
                    "complete": False
                }

                course_data.append(course)
                course_meta_data.append(course_meta)
                degree_audit_data.append(degree_audit)

        else:
            if line[21:24].strip() == "LAB" or line[21:24].strip() == "REC":
                # This is a lab or recitation
                if last_index is not None:
                    # Update the course meta and degree audit for the last course entry
                    course_meta['lab'] = True

# Update degree audit data with completed courses
for course in degree_audit_data:
    if course['course_number'] in completed_course_list:
            course['complete'] = True

# Convert the list of course dictionaries to JSON
course_data_json = json.dumps(course_data, indent=4)
course_meta_json = json.dumps(course_meta_data, indent=4)
degree_audit_json = json.dumps(degree_audit_data, indent=4)
```

Joel

Main logic(cpp)

CourseRecommendationEngine is designed to revolutionize how students approach their academic planning, particularly in managing their course workload for the semester.

The industry standard of 2 hours per credit hour is multiplied by a scalar after looking at that particular courses number of assignments, labs, group assignments, attendance requirement, and prior student feedback. This gives a truer estimation of the time you can expect to spend outside of class for that class.

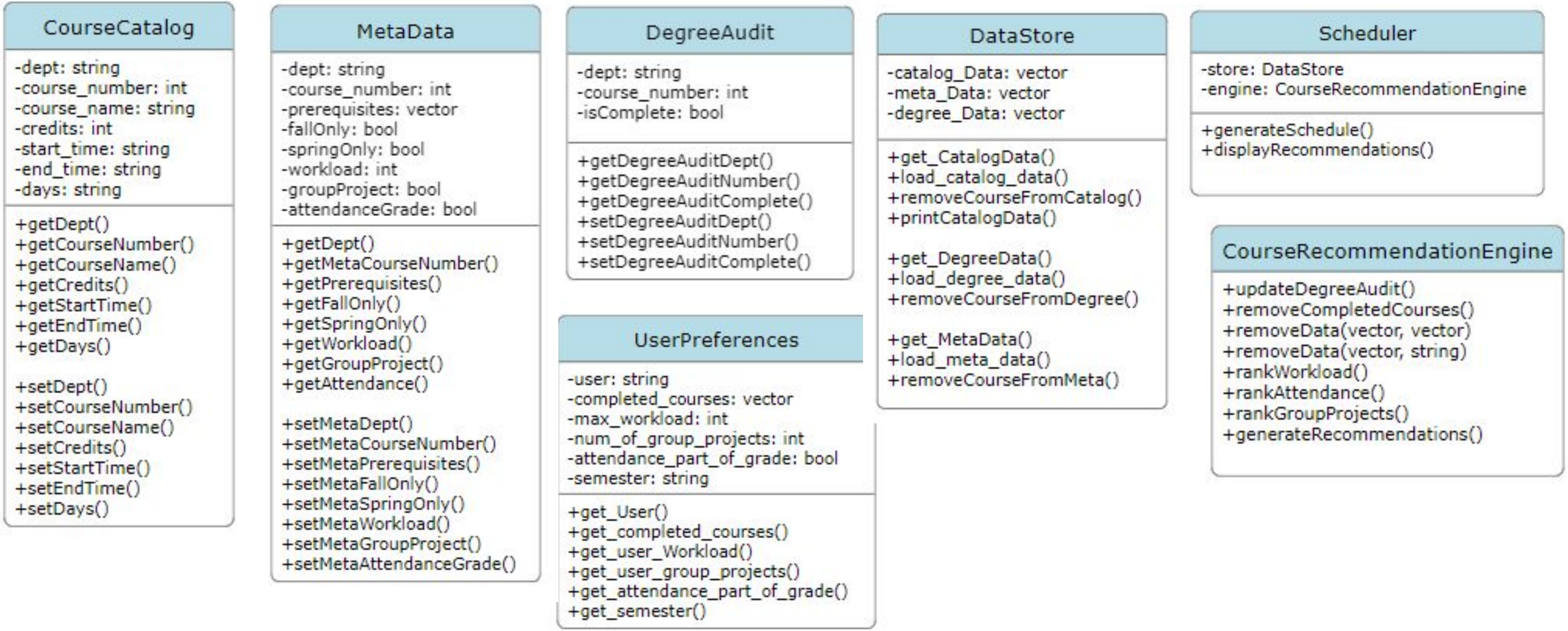
When using the engine you select a desired workload and can select a preferred amount of labs, group projects, attendance requirements. The engine will try to fit a schedule to match that criteria.

Alex

- Linking the Python file to the main C++ file that contained the project.
    - Created a py objection the main file after initializing the python file
    - Iterated through the list that was the python object

- Printed Courses
    - The python file is provided user input in the form of a text block.
    - After that block is translated into a list of courses, I call on this text block in the C++ file and print each course. This list of courses is the actual schedule that the user submits.

Component Diagrams



CourseCatalog

-dept: string
-course_number: int
-course_name: string
-credits: int
-start_time: string
-end_time: string
-days: string

+getDept()
+getCourseNumber()
+getCourseName()
+getCredits()
+getStartTime()
+getEndTime()
+getDays()

+setDept()
+setCourseNumber()
+setCourseName()
+setCredits()
+setStartTime()
+setEndTime()
+setDays()

MetaData

-dept: string
-course_number: int
-prerequisites: vector
-fallOnly: bool
-springOnly: bool
-workload: int
-groupProject: bool
-attendanceGrade: bool

+getDept()
+getMetaCourseNumber()
+getPrerequisites()
+getFallOnly()
+getSpringOnly()
+getWorkload()
+getGroupProject()
+getAttendance()

+setMetaDept()
+setMetaCourseNumber()
+setMetaPrerequisites()
+setMetaFallOnly()
+setMetaSpringOnly()
+setMetaWorkload()
+setMetaGroupProject()
+setMetaAttendanceGrade()

DegreeAudit

-dept: string
-course_number: int
-isComplete: bool

+getDegreeAuditDept()
+getDegreeAuditNumber()
+getDegreeAuditComplete()
+setDegreeAuditDept()
+setDegreeAuditNumber()
+setDegreeAuditComplete()

UserPreferences

-user: string
-completed_courses: vector
-max_workload: int
-num_of_group_projects: int
-attendance_part_of_grade: bool
-semester: string

+get_User()
+get_completed_courses()
+get_user_Workload()
+get_user_group_projects()
+get_attendance_part_of_grade()
+get_semester()

DataStore

-catalog_Data: vector
-meta_Data: vector
-degree_Data: vector

+get_CatalogData()
+load_catalog_data()
+removeCourseFromCatalog()
+printCatalogData()

+get_DegreeData()
+load_degree_data()
+removeCourseFromDegree()

+get_MetaData()
+load_meta_data()
+removeCourseFromMeta()

Scheduler

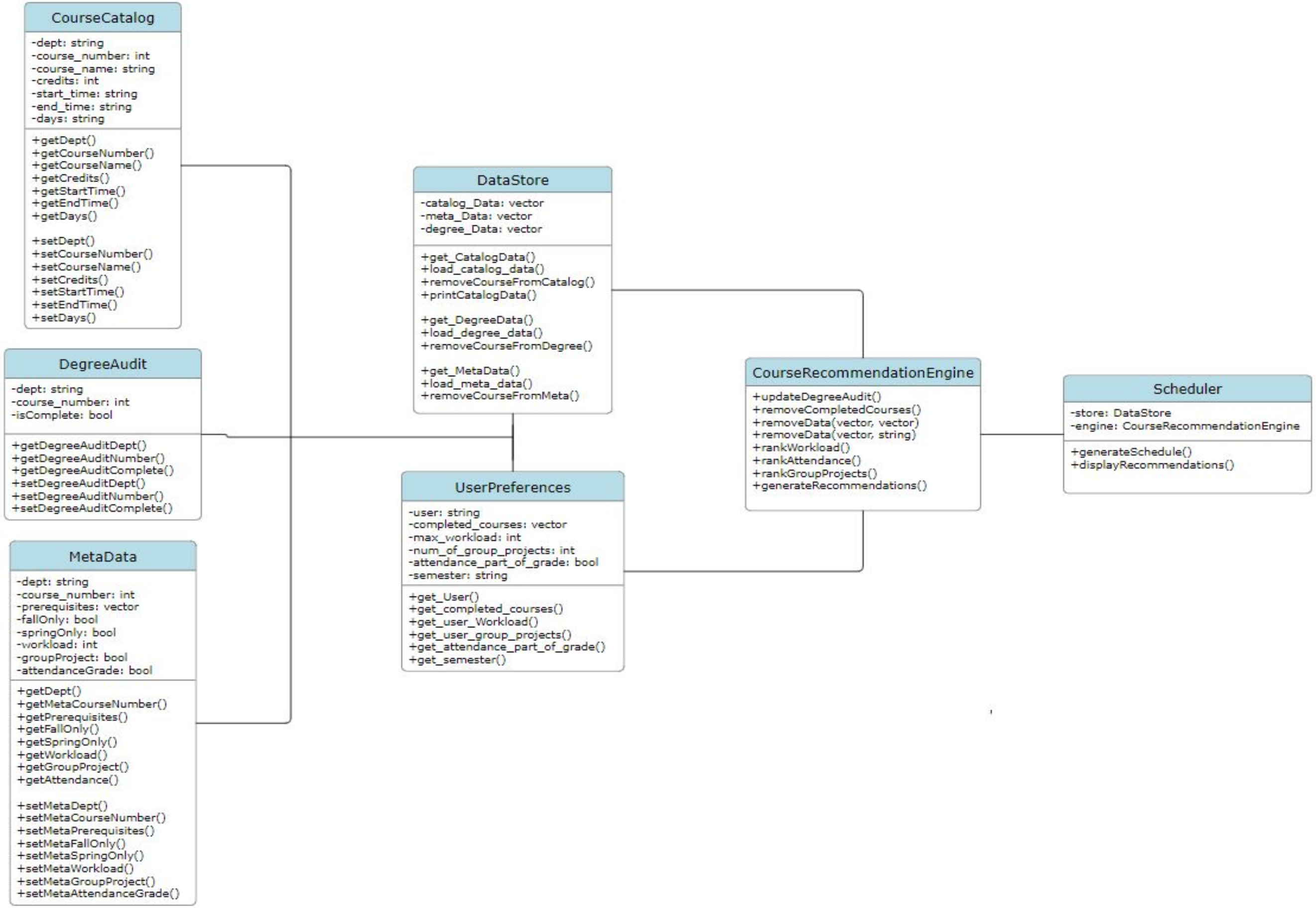
-store: DataStore
-engine: CourseRecommendationEngine

+generateSchedule()
+displayRecommendations()

CourseRecommendationEngine

+updateDegreeAudit()
+removeCompletedCourses()
+removeData(vector, vector)
+removeData(vector, string)
+rankWorkload()
+rankAttendance()
+rankGroupProjects()
+generateRecommendations()

Component Class UML



CourseCatalog

-dept: string
-course_number: int
-course_name: string
-credits: int
-start_time: string
-end_time: string
-days: string

+getDept()
+getCourseNumber()
+getCourseName()
+getCredits()
+getStartTime()
+getEndTime()
+getDays()

+setDept()
+setCourseNumber()
+setCourseName()
+setCredits()
+setStartTime()
+setEndTime()
+setDays()

DegreeAudit

-dept: string
-course_number: int
-isComplete: bool

+getDegreeAuditDept()
+getDegreeAuditNumber()
+getDegreeAuditComplete()
+setDegreeAuditDept()
+setDegreeAuditNumber()
+setDegreeAuditComplete()

MetaData

-dept: string
-course_number: int
-prerequisites: vector
-fallOnly: bool
-springOnly: bool
-workload: int
-groupProject: bool
-attendanceGrade: bool

+getDept()
+getMetaCourseNumber()
+getPrerequisites()
+getFallOnly()
+getSpringOnly()
+getWorkload()
+getGroupProject()
+getAttendance()

+setMetaDept()
+setMetaCourseNumber()
+setMetaPrerequisites()
+setMetaFallOnly()
+setMetaSpringOnly()
+setMetaWorkload()
+setMetaGroupProject()
+setMetaAttendanceGrade()

DataStore

-catalog_Data: vector
-meta_Data: vector
-degree_Data: vector

+get_CatalogData()
+load_catalog_data()
+removeCourseFromCatalog()
+printCatalogData()

+get_DegreeData()
+load_degree_data()
+removeCourseFromDegree()

+get_MetaData()
+load_meta_data()
+removeCourseFromMeta()

UserPreferences

-user: string
-completed_courses: vector
-max_workload: int
-num_of_group_projects: int
-attendance_part_of_grade: bool
-semester: string

+get_User()
+get_completed_courses()
+get_user_Workload()
+get_user_group_projects()
+get_attendance_part_of_grade()
+get_semester()

CourseRecommendationEngine

+updateDegreeAudit()
+removeCompletedCourses()
+removeData(vector, vector)
+removeData(vector, string)
+rankWorkload()
+rankAttendance()
+rankGroupProjects()
+generateRecommendations()

Scheduler

-store: DataStore
-engine: CourseRecommendationEngine

+generateSchedule()
+displayRecommendations()