*A scheduling tool based on LSU course offering but with additional features and improvements to enhance user’s experience in planning for their next semester.*

*G20 Summit*

Project Portfolio

*November 15, 2023*

[Introduction 2](#_Toc151034331)

[The G20 Summit Team 3](#_Toc151034332)

[System Requirements 4](#_Toc151034333)

[Requirements 4](#_Toc151034334)

[Epics 4](#_Toc151034335)

[Epic #1 4](#_Toc151034336)

[User Stories 4](#_Toc151034337)

[User Story #1 4](#_Toc151034338)

[User Story #2 4](#_Toc151034339)

[Project Management 5](#_Toc151034340)

[Continuity of Operations Plan (COOP) 5](#_Toc151034341)

[Project Plan 0](#_Toc151034342)

[System Architecture Design and Development 0](#_Toc151034343)

[System Implementation 0](#_Toc151034344)

[Project Postmortem <Postmortem> 0](#_Toc151034345)

[Project Wins 0](#_Toc151034346)

[Root Cause Analysis 0](#_Toc151034347)

[Lessons Learned 0](#_Toc151034348)

[System Design 1](#_Toc151034349)

[System Architecture 1](#_Toc151034350)

[Component Design 1](#_Toc151034351)

[Data Flow 1](#_Toc151034352)

[System Components 2](#_Toc151034353)

[Component 1: Course Recommendation 2](#_Toc151034354)

[Component 2: Course Catalog, Degree Audit, Meta Data 2](#_Toc151034355)

[Component 3: User Preference 2](#_Toc151034356)

[Design Pattern 2](#_Toc151034357)

[System Implementation 4](#_Toc151034358)

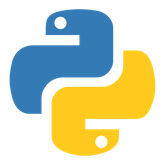
# Introduction

The current LSU schedule system and database is considered archaic for the modern web. It sorts classes by only the semester and department, which forces students to find their desired courses within a list of others that are most likely inconsequential for them. Due to the confusing layout, it has led to many frustrations within the student body and possible results not selecting the best courses for their specific needs.

This updated scheduling system will solve the aforementioned issues by including a robust sorting system for all classes, a simple web interface that anyone can use, and the ability to view specific course and/or instructor information.

It’s made primarily with C++ including libraries such as nlohmann/json.hpp and along with another programming language, Python, for the back-end. The front-end used HTML, CSS, and Javascript with jQuery and selectize.js for better user experience.

A hexagon with a white c and plus symbol

Description automatically generated  

Core Features:

* Filter and sort available courses based on user’s selected criteria
* An intuitive web interface that makes selecting classes simple
* Automatically updates available courses based on class time selected

Viable Features:

* The ability to view current or prior course syllabus for each courses
* Denote courses with specific requirements such as a group project or a lab involved
* Recommend courses based on the course flowchart for each major

Stretch Features

* Estimate the course’s quality based on RateMyProfessor score
* Implement degree audit functionality

# The G20 Summit Team

**GitHub Link:** [*https://github.com/hkaiserteaching/csc3380-fall-2023-project-group-20*](https://github.com/hkaiserteaching/csc3380-fall-2023-project-group-20)

**Preston Saxon** – Team Leader; UML Diagram, Degree Audit Scraping (backend)

**Alexander Leake** – Linking Python file with main C++ file, Printing courses (backend)

**Jacob Rogers** – Catalog Scraper and Data Generation (backend)

**Joel Rogers** – Course Recommendation, main C++ file (backend)

**Vivian San** – User interface (front-end)

# System Requirements

## Requirements

Required dependencies: “nlohmann-json,” “python2”

## Epics

### Epic #1

*As a student, I want to easily schedule and plan my classes for next semester, so I can manage my time quickly without opening up many tabs (degree audit, calendar, course offering, etc.).*

## User Stories

### User Story #1

*As a student, I want to know the outline of the course I’m signing up for, so I can consider how I should approach the course or rethink my decision in selecting it.*

### User Story #2

*As a student, I want to search for courses in my major, so I can enroll in relevant class that goes toward my degree credit.*

# Project Management

## Continuity of Operations Plan (COOP)

If team member is sick and cannot fulfill his or her duties, their responsibilities regarding the project will be accounted for by the next member with the most time on their hands or a later date once available. If a team member cannot attend a meeting in-person or through Discord, they will be messaged about the content of the meeting. The absent member will still be in contact with the rest of the team concerning the current developments on the project and any new requirements that must be met for their specific job.

## Project Plan

### System Architecture Design and Development

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **Activity** | **Pre #** | **Estimated**  **Effort** | **Actual**  **Effort** | **Estimated**  **Start Date** | **Estimated**  **Finish Date** | **Actual**  **Start Date** | **Actual**  **Finish Date** |
| 1 | Obtain student’s input based on preferences | 1 | 10% | 10% | 9/20/23 | 10/7/23 | 11/9/23 | 11/15/23 |
| 2 | User interface that allows student to interact in selections | 1 | 5% | 5% | 9/20/23 | 10/7/23 | 11/9/23 | 11/15/23 |
| 3 | Gather student’s degree audit | 3 | 30% | 30% | 9/20/23 | 10/9/23 | 11/9/23 | 11/15/23 |
| 4 | Recommendation based on student’s preferences and completed courses from degree audit | 4 | 20% | 20% | 9/20/23 | 11/3/23 | 11/9/23 | 11/15/23 |
| 5 | Obtain LSU database – course catalog | 1 | 15% | 15% | 9/20/23 | 10/9/23 | 11/9/23 | 11/15/23 |
| 6 | Obtain LSU database – course metadata (syllabus, reqs., etc) | 2 | 15% | 15% | 9/20/23 | 10/9/23 | 11/9/23 | 11/15/23 |
| 7 | Display selected courses based on time | 5 | 5% | 5% | 9/20/23 | 11/3/23 | 11/9/23 | 11/15/23 |

### System Implementation

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **Activity** | **Pre #** | **Estimated**  **Effort** | **Actual**  **Effort** | **Estimated**  **Start Date** | **Estimated**  **Finish Date** | **Actual**  **Start Date** | **Actual**  **Finish Date** |
| 1 | User interface development | 2 | 10% | 5% | 9/20/23 | 11/3/23 | 11/10/23 | 11/15/23 |
| 2 | Backend development | 1 | 50% | 75% | 9/20/23 | 11/3/23 | 10/30/23 | 11/15/23 |
| 3 | LSU database integration | 1 | 20% | 30% | 9/20/23 | 11/3/23 | 11/10/23 | 11/15/23 |
| 4 | Optimization | 5 | 6.7% | 15% | 9/20/23 | 11/3/23 | 11/10/23 | 11/15/23 |
| 5 | Debugging | 3 | 6.7% | 20% | 9/20/23 | 11/3/23 | 11/10/23 | 11/15/23 |
| 6 | Testing | 4 | 6.7% | 5% | 9/20/23 | 11/3/23 | 11/10/23 | 11/15/23 |

## Project Postmortem <Postmortem>

### Project Wins

[Provide a bulleted list of at least 3 positive aspects of the project.]

### Root Cause Analysis

[Provide a bulleted list of at least 3 negative aspects of the project. For each negative, provide the answer to the three successive “Why” questions. ]

### Lessons Learned

[For each negative aspect identified in the Root Cause Analysis, provide a mitigation strategy (i.e., what process should be introduced) to ensure that the problem is not repeated in subsequent projects.]

# System Design

A scheduling tool to enhance user experience by providing course recommendation based on information that the algorithm obtain from both the user’s input as well as the retrieved data from LSU’s databases (course offering, degree audit, metadata) to efficiently create a schedule for next semester.

## System Architecture

Utilizing C++ with JSON and Python to gather information from LSU’s databases in order to reorganize them and optimize the performance of the system so that it allows for better user interaction.

### Component Design

Created by: *Joel Rogers*

A diagram of a course

Description automatically generated

### Data Flow

Created by: *Joel* *Rogers*

A diagram of a course

Description automatically generated

## System Components

### Component 1: Course Recommendation

Generates recommended courses based on user’s selection of preferences. It takes data from the degree audit, courses for the degree, and the meta data along with those courses. Based on their degree audit, completed courses will be removed to narrow down the choice of courses. Any preferences the user selected such as the amount of workload, attendance, etc. will be ranked.

### Component 2: Course Catalog, Degree Audit, Meta Data

These three are separate components but they function similarly. Members working on these components are extracting the information from the LSU databases to reorganize and function together into one program. Once they are redesigned and organized, the data are compared to obtain user preferences based on the information stored from input.

### Component 3: User Preference

Obtained user’s selected preferences such as the amount of workload, whether they want group projects, if attendance is optional, etc. We’re also obtaining other information, which are the completed courses from the user’s degree audit, in order to further narrow down possible courses the program can recommend in the course recommend component.

## Design Pattern

Created by: *Preston Saxon*

A diagram of a computer

Description automatically generated

# System Implementation

|  |  |  |
| --- | --- | --- |
| **Architectural Component** | **Programming Language(s) %** | **Team Member(s) %** |
| Degree audit database | C++ (81.9%)  Python (1.1%)  Other (1.4%) | Preston Saxon (100%) |
| Linking Python file with main C++ file, Printing courses (Gather Course Data) | C++ (81.9%)  Python (1.1%)  Other (1.4%) | Alexander Leake (100%) |
| Recommendation Main cPP | C++ (99.9%)  Python (1.1%)  Other (1.4%) | Joel Rogers (100%) |
| Catalog Scraper and Data Generation (Gather Course Data) | Python (98%)  Other (2%) | Jacob Rogers (100%) |
| Student I/O | Javascript (15.6%), Other (1.4%) | Vivian San (100%) |