A New Home: Final Phase

INSO 4101: Introduction to Software Engineering

Informative part

1. Name, Place, Date

A New Home by Team Codewalkers, UPR Mayagüez Puerto Rico, Spring Semester 2021

2. Situation

Twice a year, students at the University of Puerto Rico (UPR) go through the often-frustrating process of enrolling into their classes. In the University of Puerto Rico Mayagüez (UPRM) system, the enrollment process favors students with good grades, athletes and others by letting them start the enrollment process before other students, which often causes students to miss out on classes they need to take on the upcoming semester. Many of these classes are core requirements without which students may not be able to progress on their academic careers. This has been a constant problem for UPR students for years now and as a result students have had their graduation dates delayed by one or more semesters. This issue is further worsened by the fact that once students are several years into their majors classes are often given only one section per semester, if the class is even offered. This is an issue that needs to be addressed given the amount of students that are affected by this problem, whereas students in private colleges in Puerto Rico don't suffer from this problem.

3. Needs

For enrollment decision making, students need easy and convenient ways to access their information regarding their major, regardless of what major it is, in order to make the enrollment process easier. As such, there is a need for a unified solution which is capable of not only enrolling students, but also providing University faculty knowledgeable data on what courses are in high demand. Such data can be accessed by the University faculty in order to better prepare for each semester in order to accommodate the student body's needs. This functionality is missing from the UPR's own enrollment system which leads students to having to move through various platforms, therefore there is a need for domain description, a requirements prescription, software design and an implementation for a platform is needed to solve all of these problems.

4. Idea

We will develop a software architecture, requirement prescription, domain description and an implementation to address the students' needs. The domain description needs to be easily accessible by the student body, yet also provide faculty the ability to see student demand and, if needed, open or close more sections for each class. In the platform's current state, data has been added manually for the purposes of serving as a demo of sorts, but once the platform has been fully developed and tested we would hope for it to eventually replace the already existing student portal used in UPRM. A mock user and curriculum has been added to the platform in order to demonstrate how it would eventually look and function.

Presently, the platform has been built using a simple python script that functions as a server and nothing else besides vanilla HTML, CSS, and Javascript, with several JSON files serving as a "database". The decision to do this and not use any pre existing platforms or other tools was for us to take it mostly as a learning experience for these programming languages. In the future, we wish to iron out any and all issues we currently have in the platform before moving forward with development and making it a full fledged platform that can be used by University faculty and students for years to come.

5. Scope, span

The scope of this project is for students to tell the faculty which classes they intend to take during the upcoming semester according to their curriculum and/or interest. The span is that the faculty will be able to have insight into course demand in order to better prepare for the enrollment process.

Descriptive part

1. Rough Domain Sketches

The enrollment process in the UPR is a very stressful moment for many students. The reason as to why most students do not enjoy this process is because it prioritizes students with better grades, athletes, and certain other categories of students by allowing them to start their enrollment process days before the rest of the students have their chance. This leads to courses being full before a majority of students even gain access to the enrollment system.

There have been many ideas and suggestions to make the enrollment process a better one but, as many people know, the UPR is underfunded, the enrollment system that is being used at the moment is vastly outdated, and there appears to be no intent to change it in the near future.

The enrollment process in UPRM has been the cause of a crisis in the academic life of students for many years. Due to this, many students prefer to enroll in a private university with the hopes of finishing their degrees in a timely fashion. At that point students who are far into their degrees just want it, no matter how they get it. You go to any UPRM department and say "I couldn't enroll in X course due to lack of space." or "The system does not let me enroll in X course due to prerequisites.". So that brings lots of questions, what are they doing wrong? What can they do to improve their system? Is the university prepared each semester for the enrollment process when the date comes? Most of the time the answers to these questions are not what the student wants to hear, and so the university needs feedback from students in order to make sure that they are comfortable with the enrollment process in UPRM.

2. Requirements

Students should be able to view and interact with their curriculum so they may plan for their next academic terms. The system should also provide feedback to the departments as to how many students will take said courses in the next semester so that they can see the demand and adjust the offer accordingly. The user interface should be able to compete with today's standards with the user interface by keeping it simple and straightforward yet versatile. The system should be able to catalog departments differently and provide useful information to said departments.

3. Domain Phenomenon

Domain Terminology:

- **UPR** - abbreviation for University of Puerto Rico, a public university system consisting of 11 campuses around the island.
- **UPRM** - abbreviation for the University of Puerto Rico, Mayagüez. A campus of the University of Puerto Rico.
- **Semester** - a half-year term in a school or university, typically lasting between fifteen and eighteen weeks.

- **Student** - a person who is studying at a university or other place of higher education.
- **University faculty** - a division within a university or college comprising one subject area or a group of related subject areas, possibly also delimited by level.
- **Enrollment** - the action of enrolling or being enrolled into a course.
- **Department** - a division of a university or school faculty devoted to a particular academic discipline.
- **Campus** - the grounds and buildings of a university, college, or school.
- **Curriculum** - the classes a student has to take to get their degree.
- **Course** - class that the student enrolls in to take during a specific semester.
- **Section** - one (or several) classes for any given course

Domain Entities:

Atomic:

- **Student** - UPRM student looking to let his department know he/she wants to take a specific course next semester.
- **Department** - a specific department of UPRM who wants to know the exact number of students who want to take a given course.
- **Curriculum** - the classes a student has to take to finish their degree.

Composed:

- **Curriculums** - List of curriculum, each item in the list will be the curriculum of a specified degree.

Domain Events:

A. **From a student's perspective:**
    - The curriculum has been displayed to the student.
    - Courses suggestions for the next academic term are shown.
    - All the courses given in a determined term are shown.
    - The student is communicating with the faculty members of the departments of the university.

B. **From a faculty member's perspective:**
    - The faculty has been shown how many students want to take a specific course(demand).
    - The faculty is communicating better with the student indirectly.
    - The faculty is showing the classes that will be given next term.

- The faculty is opening and closing either sections or courses for the next academic term.

Domain Behavior:

- Student logs into the system using their UPR credentials
- Student looks up his curriculum.
- Student look and select for courses based on their interests and curriculum.
- Student see all the courses that are going to be given in the future.
- System determines if student meets the requirements to enroll in the course
- Student enrolls in a class and updates the student demand for that particular course
- Faculty member log into web app using their UPR credentials
- Faculty member sees student demand for each course
- Faculty members uses web app to updates available courses based on student demand by opening/closing courses and/or sections

4. Domain Requirements

A. For students:

- Easy access to the student's curriculum
- Show detailed progress on a student's curriculum
- Be able to see all the courses that are going to be given in a academic term.
- Be able to see course demand and to notify University faculty of a student's desire to enroll on a specific course
- Show whether a student meets requirements in order to enroll a course
- Be able to show demand for a course, whether it is being offered or not
- Be able to enroll in courses directly in the platform

B. For faculty members:

- Show student demand for any given course
- Be able to add or remove courses, and sections, based on demand
- Be able to update course validation

5. Interface requirements

Shared Phenomena and Concept Identification

- Student
- Department
- Curriculum

These three phenomena and concepts are the most important part of our domain and will be included in our machine. In our web application the students and department will be our users. Students will use their college emails to create an account in order to be able to let their department know what courses they want to enroll in. Student users will have different privileges when compared to the faculty users.

6. Software Architecture Design

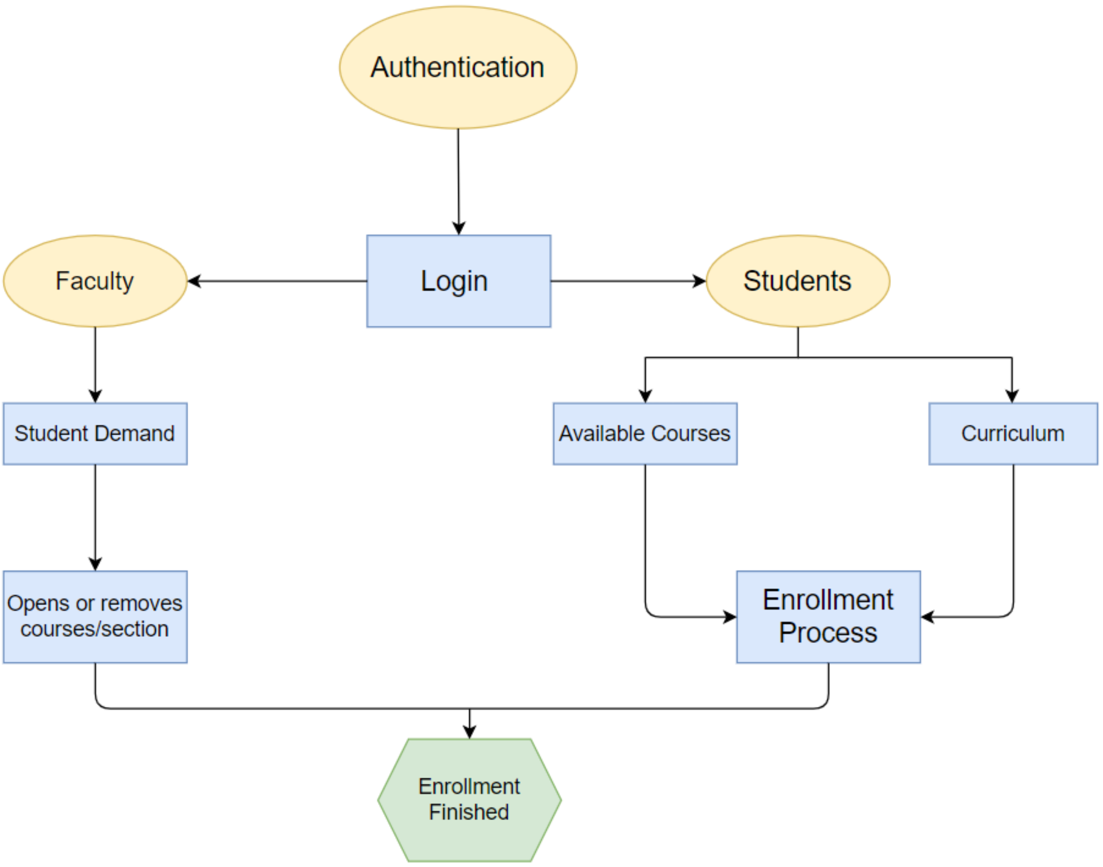


Figure 2: Rough sketch of the interface for the web application

- The login page will receive institutional credentials and detect if the user is a faculty member or a student.
- If a student, it will provide access to their curriculum and also allow them to select the courses they want to take next semester.
- If faculty, the user will have access to the list of courses and how many students want to take it, in order to open as many sections as they can for the students to enroll.

7. Machine Requirements

A. Performance: Since the app is made with the very simple HTML/CSS and Javascript the performance of the website is pretty good but since we are using JSON files to store data, our storage is limited.

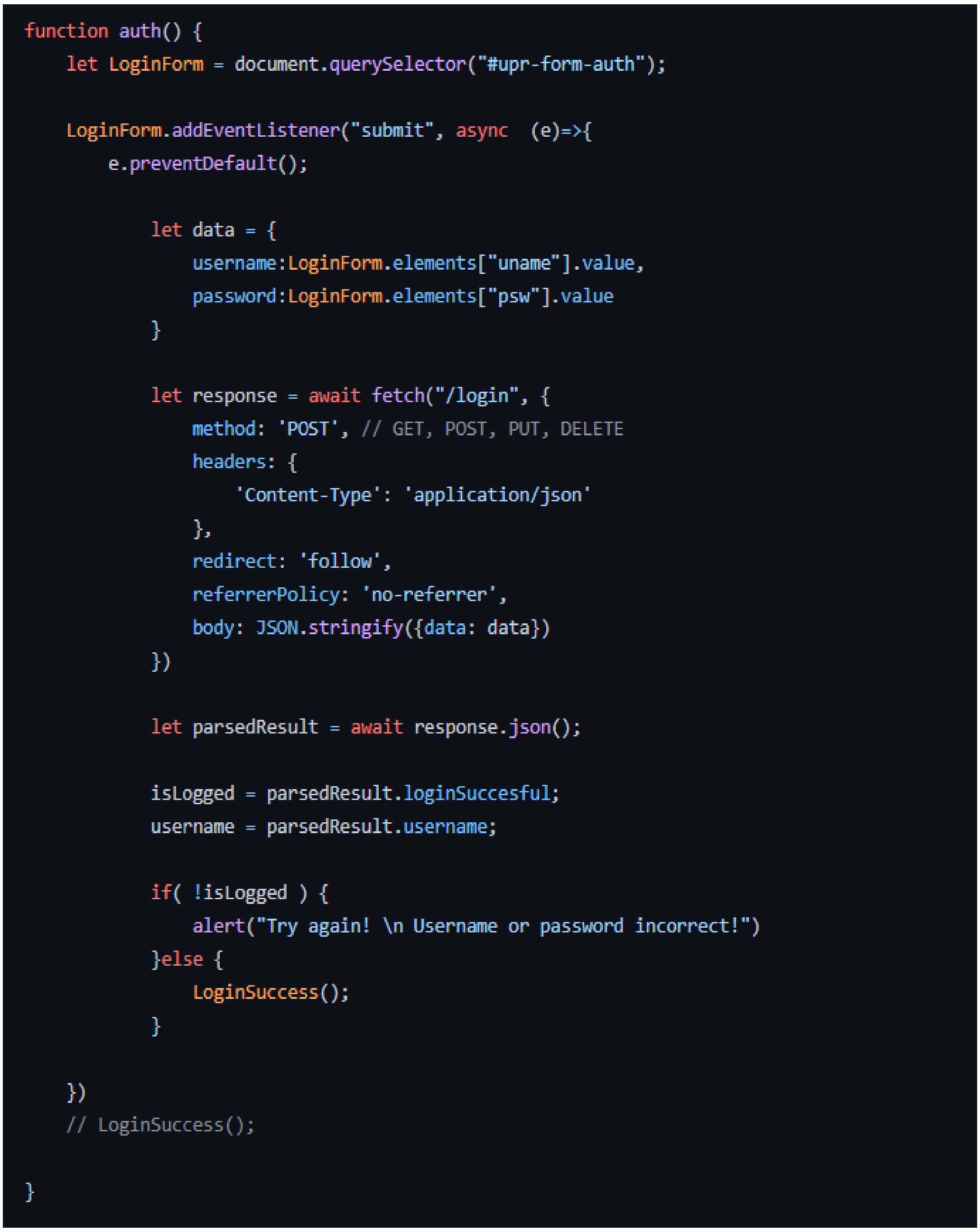
B. Dependability: Our application would only allow users to register with an email corresponding to the UPR to make sure that the right people use the application.

C. Maintenance: After each semester, the database corresponding to the courses students want to take would be reset.

D. Platform: Since it is a web application we are using HTML/CSS and Javascript so that it is compatible with most web browsers.

8. Software components

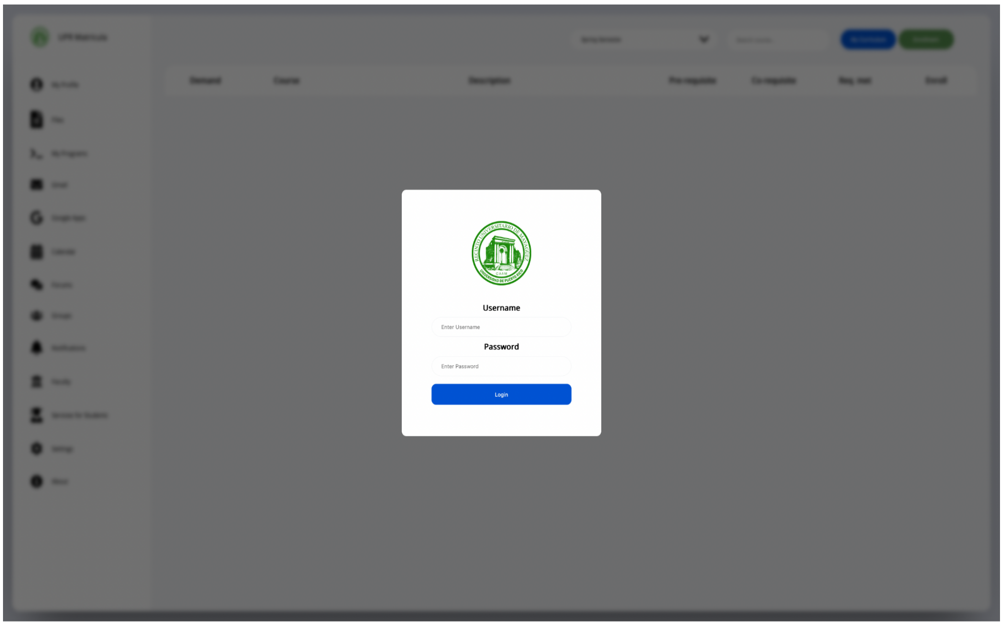
Selected fragments of implementation:

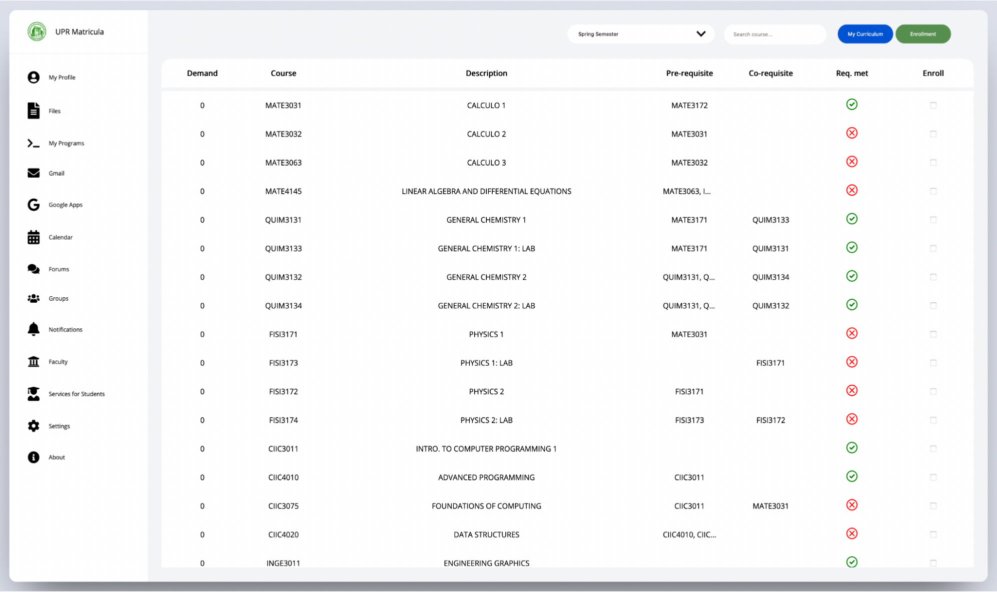


```
function auth() {
    let LoginForm = document.querySelector("#upr-form-auth");

    LoginForm.addEventListener("submit", async  (e)=>{
        e.preventDefault();

            let data = {
                username:LoginForm.elements["uname"].value,
                password:LoginForm.elements["psw"].value
            }

            let response = await fetch("/login", {
                method: 'POST', // GET, POST, PUT, DELETE
                headers: {
                    'Content-Type': 'application/json'
                },
                redirect: 'follow',
                referrerPolicy: 'no-referrer',
                body: JSON.stringify({data: data})
            })

            let parsedResult = await response.json();

            isLogged = parsedResult.loginSuccesful;
            username = parsedResult.username;

            if( !isLogged ) {
                alert("Try again! \n Username or password incorrect!")
            }else {
                LoginSuccess();
            }

    })
    // LoginSuccess();

}
```

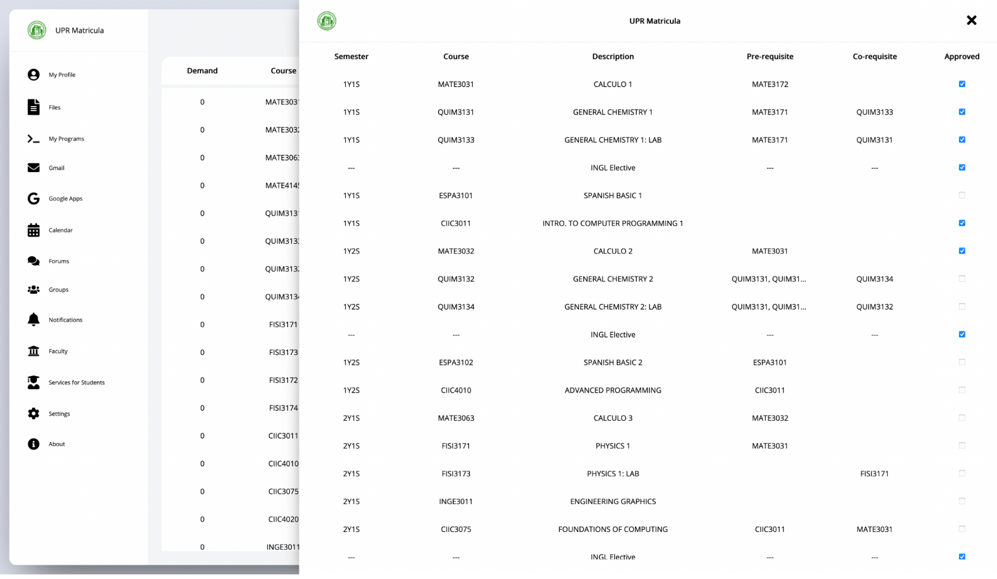
For the back-end portion of our system, a function called *auth* was implement which takes the username and the password of the user and validates it via the UPRM accounts database and validates the account, authenticating the user based on whether they are a student or faculty member based on their respective credentials. Letting the users freely use the system within their authorization.

9. Sketch of the current User Interface (UI)





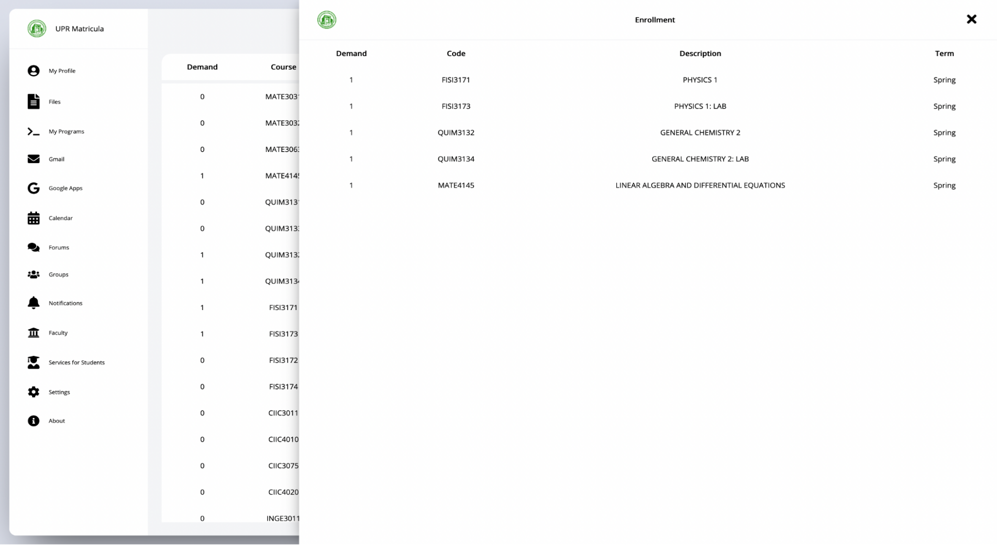
Demand	Course	Description	Pre-requisite	Co-requisite	Req. met	Enroll
0	MATE3031	CALCULO 1	MATE3172		✓	□
0	MATE3032	CALCULO 2	MATE3031		☒	□
0	MATE3063	CALCULO 3	MATE3032		☒	□
0	MATE4145	LINEAR ALGEBRA AND DIFFERENTIAL EQUATIONS	MATE3063, I...		☒	□
0	QUIM3131	GENERAL CHEMISTRY 1	MATE3171	QUIM3133	✓	□
0	QUIM3133	GENERAL CHEMISTRY 1: LAB	MATE3171	QUIM3131	✓	□
0	QUIM3132	GENERAL CHEMISTRY 2	QUIM3131, Q...	QUIM3134	✓	□
0	QUIM3134	GENERAL CHEMISTRY 2: LAB	QUIM3131, Q...	QUIM3132	✓	□
0	FISI3171	PHYSICS 1	MATE3031		☒	□
0	FISI3173	PHYSICS 1: LAB		FISI3171	☒	□
0	FISI3172	PHYSICS 2	FISI3171		☒	□
0	FISI3174	PHYSICS 2: LAB	FISI3173	FISI3172	☒	□
0	CIIC3011	INTRO. TO COMPUTER PROGRAMMING 1			✓	□
0	CIIC4010	ADVANCED PROGRAMMING	CIIC3011		✓	□
0	CIIC3075	FOUNDATIONS OF COMPUTING	CIIC3011	MATE3031	☒	□
0	CIIC4020	DATA STRUCTURES	CIIC4010, CIIC...		☒	□
0	INGE3011	ENGINEERING GRAPHICS			✓	□



Demand	Course
0	MATE303
0	MATE303
0	MATE306
0	MATE414
0	QUIM313
0	QUIM313
0	QUIM313
0	QUIM313
0	FISI3171
0	FISI3173
0	FISI3172
0	FISI3174
0	CIIC3011
0	CIIC4010
0	CIIC3075
0	CIIC4020
0	INGE3011

UPR Matricula

Semester	Course	Description	Pre-requisite	Co-requisite	Approved
1Y1S	MATE3031	CALCULO 1	MATE3172		☑
1Y1S	QUIM3131	GENERAL CHEMISTRY 1	MATE3171	QUIM3133	☑
1Y1S	QUIM3133	GENERAL CHEMISTRY 1: LAB	MATE3171	QUIM3131	☑
---	---	INGL Elective	---	---	☑
1Y1S	ESPA3101	SPANISH BASIC 1			□
1Y1S	CIIC3011	INTRO. TO COMPUTER PROGRAMMING 1			☑
1Y2S	MATE3032	CALCULO 2	MATE3031		☑
1Y2S	QUIM3132	GENERAL CHEMISTRY 2	QUIM3131, QUIM31...	QUIM3134	□
1Y2S	QUIM3134	GENERAL CHEMISTRY 2: LAB	QUIM3131, QUIM31...	QUIM3132	□
---	---	INGL Elective	---	---	☑
1Y2S	ESPA3102	SPANISH BASIC 2	ESPA3101		□
1Y2S	CIIC4010	ADVANCED PROGRAMMING	CIIC3011		□
2Y1S	MATE3063	CALCULO 3	MATE3032		□
2Y1S	FISI3171	PHYSICS 1	MATE3031		□
2Y1S	FISI3173	PHYSICS 1: LAB		FISI3171	□
2Y1S	INGE3011	ENGINEERING GRAPHICS			□
2Y1S	CIIC3075	FOUNDATIONS OF COMPUTING	CIIC3011	MATE3031	□
---	---	INGL Elective	---	---	☑



Demand	Course
0	MATE303
0	MATE303
0	MATE306
1	MATE414
0	QUIM313
0	QUIM313
1	QUIM313
1	QUIM313
1	FISI3171
1	FISI3173
0	FISI3172
0	FISI3174
0	CIIC3011
0	CIIC4010
0	CIIC3075
0	CIIC4020
0	INGE3011

Enrollment

Demand	Code	Description	Term
1	FISI3171	PHYSICS 1	Spring
1	FISI3173	PHYSICS 1: LAB	Spring
1	QUIM3132	GENERAL CHEMISTRY 2	Spring
1	QUIM3134	GENERAL CHEMISTRY 2: LAB	Spring
1	MATE4145	LINEAR ALGEBRA AND DIFFERENTIAL EQUATIONS	Spring

Figure 3: Sketch for the UI of the web application

10. Progress Report

Roles

Edimar Valentín Kery - Frontend/Backend(Hybrid)

Pablo G. Sepulveda Cosme - Frontend

Kelvin S. González Aquino - Backend

Kristinne N. Negrón Terón - Backend

Fernando Méndez González - Frontend/Backend(Hybrid)

For the organization of this project we started by meeting up through the Matrix chat room via voice calls in the earlier days of the semester. Since then we have moved on to using a WhatsApp group chat given it convenience, and a mixture of Discord and Google Meet for group meetings. For Phase 1 we had a lot to fix within our project as we felt we had missed the mark on our team proposal. In this phase we thought about what programming languages we wanted to use and came to the initial idea that Python coupled with HTML and CSS would be our best bet given that almost all of the team members are familiar with them due to previous classes. We also made some advancements in the front-end portion of the application by creating a rough sketch of the user interface which was made using Figma, a web based vector graphics editor and prototyping tool.

For Phase 2 we initially started working on the frontend part of the web application. Since we wanted to work with HTML/CSS, we thought of using Python's Flask framework which we thought would serve our needs. For Phase 2 we got to build a very rough home page of the web application and other necessary pages like the about page, the contact page and the login/register pages.

For the Final Phase, we decided to scratch what we had built so far in order to go back to basics and to ensure that all group members can contribute, given the different levels of familiarity of the developers. A rough and simple Python server was developed in order to be able to run with just the simplest of requirements. From then on the application was developed using just HTML, CSS, and Javascript.

Analytical part

1. Concept analysis:

- The web application will provide the user with information crucial for the upcoming semester.
    - For the students, it will give them as users the most convenient way to plan their class schedule regardless of their date of enrollment.
    - For the faculty members, it would give them insight into their students' academic needs and would give them ample time to plan accordingly.
- The web application will provide options for students and the moment to choose courses for next semester.
    - The student will be able to choose beforehand courses that they need to take according to their curriculum.
- The faculty will have the authorization to see how many students want to take a specific course.
    - This will help the faculty decide how many sections of an established course they are going to open for students to enroll into their classes without problem.
- The web application will work with institutional credentials
    - Using institutional credentials will make the use of the software safer because only people that have access to the institutional services will be able to use it.

2. Validation

The validation process for this project will be carried out by the stakeholders which consist of the following:

- Marko Schutz (Professor) - The professor will be reviewing each phase of the project and will be giving feedback for us to improve the project.
- Developers - Each member of the team will give feedback throughout the development process on what they think must be improved.
- Users - We plan on adding a link to our GitHub repository in order to be able to receive end user feedback, as well as be able to keep track of bugs and feature requests that might be submitted

For validation purposes, the first is to have the evaluator (professor) see the work the developers have done, for his approval for release. As we (developers) receive approval, our task is to confirm that our web application works as intended in order for it to be released and, if a bug or an error is to be found, the work will be re-evaluated by the professor. Issues may arise due to specific courses, students either meeting or not pre/co-requirements, problems with newly created accounts, the wrong curriculum showing up, or any other bug or error that was not fixed accordingly by the developers. When it comes to resolving these issues, if the web application was already released, the developers will work behind the scenes in order to minimise the overall effect on users, restart the validation process, and push the updates to the appropriate servers. If any other errors or bugs arise, the process will start again as many times as needed.

3. Verification

Checking/Testing

- Before commiting a newly implemented task, the code will be put through various debugging techniques to make sure the program runs as expected. Also, each member of the team will review the system to check if there are parts of it that can be optimized or fixed.
- We will also make use of the feedback given by the users to ensure that the web application runs smoothly.
- The documentation of the web application will be essential for the web application to not lose its purpose.
- For verification purposes, there will be walkthroughs in the system made by different users, the developers included, to test for bugs, look for errors and optimization options.
- See experts if we as developers get stuck while fixing bugs or making updates to the application.
- Consistently testing the main functions of the application to ensure that it works optimally. If not, then proceed to fixing errors/bugs.