UPR Grader Project Proposal

ICOM 4009/INSO 4101: Introduction to Software Engineering

Prof. Marko Schütz-Schmuck

Team: DLMS

I. Informative Part

Group Members:

José F. Rivera Rivera

Jaime L. Miranda Ramírez

Joey R. Hernández Perez

Génesis M. Torres Pinto

Gabriela Cardona Blas

Darielis M. Morales Rodríguez

Group Name: DLMS

Location: University of Puerto Rico at Mayagüez

Date: January 2021 – onward

Partners:

- **Front-end developers:** Front-end web development is the practice of converting data to a graphical interface so that users can view and interact with that data.
    - Jaime L. Miranda Ramírez
    - Génesis M. Torres Pinto
    - Gabriela Cardona Blas
- **Back-end developers:** A back-end web developer is responsible for server-side web application logic and integration of the work front-end developers do. Back-end developers usually write the web services and APIs used by front-end developers and mobile application developers.

- José F. Rivera Rivera
- Joey R. Hernández Perez
- Darielis M. Morales Rodríguez

Current Situation

Although currently the university's student community has several platforms, most are usually a bit archaic or not so accessible, for example, Putty. In it you can see the enrolled classes and their information, the GPA, the classes taken or to take and several other functions. Although Putty has these functions, the system is usually quite unstable and one in which you sometimes must wait a long time for maintenance/updates among other things to be able to observe your GPA at the end of a semester. On the other hand, at the moment there is no platform where you can place your grades by yourself and get the GPA either for a specific semester or the general one.

Needs

There is a need to optimize and provide an efficient service to the university's student community. Essentially, to provide a user-friendly application where students can navigate through it and observe the courses already taken or to take, observe their grades or add them, in the same way to be able to observe and calculate their general GPA, semester GPA, or major GPA without the need to install complicated third-party software.

Ideas

The main idea is based on meeting the needs of the university's student community and in turn making this procedure more bearable. We are going to develop a web-based application where students can "log in" with a username and password so that they can perform various functions in a faster and more accessible way. The students will be able to choose their curriculum from a list so that they can see all the courses they need to take while having the option to create their own semester by manually adding the different courses they complete. Students will be able to see what classes they already have taken, and which ones they are missing, they will be able to observe and add grades to their classes and in this way be able to

see their GPA for a particular semester, general GPA, or major GPA. Also, they will be able to see UPR Campuses where these courses were taken.

Scope

Meet the needs of the university's student community in a fast, effective and user-friendly way. Also, improve the overall experience in the process for all the university's student community.

Span

To improve the process by making it a user-friendly experience and less-time consuming process for all the university's student community.

Synopsis

When you have an archaic, slow and quite unstable system, the user experience during the process becomes a bad and undesirable one. For this reason, there is a need to optimize the system, improving and providing an efficient service for the university's student community. In our case, it will provide a better experience through a web-based application. The website will allow the student to choose a curriculum so they can keep track of their academic progress. Also, the website will facilitate the process of showing students their GPA, either for a specific semester, the general one, or major GPA. Also, it will be able to post your grades to the corresponding classes without having to wait for a system to place it for you long after you have finished your course or without having periods that deny access to the information. On the other hand, you will be able to know which classes you have and have not taken.

II. Descriptive Part

Rough Domain Sketch of UPR GPA and Class Grades Viewer

Students from all of the University of Puerto Rico campuses have complained when using Putty to access valuable information from the university's server. Some of these tasks include: student enrollment, viewing their enrolled classes and the information pertaining to these enrolled classes, viewing their GPA and classes taken, or to take, and many more. Now, the problem may not be necessarily Putty, but at times the system is under maintenance, or when

student enrollment time comes, it is under heavy usage and it can be real slow or even crash many times. For a student that just wants to view their current GPA, courses enrolled, or courses taken and their respective grades, this might be a problem or just too much trouble. The current process for a student to view their GPA has too many steps in students' opinions. They would need to use Putty, or an equivalent program, to access the university's server, and log in as a student. Then, a selection would need to be made depending on the student's particular need at the time. In the case of viewing the GPA, the student selects "View other information". Once view other information is selected, the student would select the curriculum option. After pressing this option, the student enters personal information that includes student ID, permanent access code, social security number, and their birth date. If the information entered is correct, the student will see their curriculum courses and the grades obtained for the courses that have been taken. More information about credits completed, credits missing, general GPA, major GPA, and more, can also be found in that screen.

By UPR GPA and Class Grades Viewer, it is meant a structure of entities, functions, events, and behaviors that include, but are not limited to:

- **Students:**
    a. Looking for their GPA.
    b. Looking for grades on particular classes.
    c. Adding their own grades to their classes.
    d. Looking for courses taken or to be taken.
    e. UPR Campuses where these courses were taken.
    f. Calculating GPA based on inputted grades or from a starting number of credits and GPA.
- **University of Puerto Rico's:**
    a. Campuses
    b. 4.0 GPA scale
    c. Courses offered by campus
    d. Terms and courses offered by term.
    e. Academic programs

Domain Narrative

Every University uses some kind of metric to evaluate their students. At the University of Puerto Rico the Grade-Point Average (GPA) 4.00 scale is used for those purposes. What this means is that the highest GPA possible for a student to have is 4.0. Each student will have their own GPA based on the grades obtained in the courses they have taken. The grades that are used in the UPR system are A, B, C, D, and F. In order for students to view their current GPA, they need to either order a transcript from the Registrar's office, remember the grades obtained from previous courses and calculate it themselves, or go into Putty and enter the SSH command followed by the university's respective campus server address. From there, students can do all sorts of things; including viewing their GPA and their grades. Depending on the UPR campus, students can be enrolled in undergraduate or graduate programs.

Domain Terminology

1. A *University* is a higher education and research institution in which students receive academic degrees from the university in different academic disciplines. A university can have different campuses.

   Type: University

2. A *Student* is the entity enrolled in some academic program offered by the university. The student can be considered an undergraduate or graduate depending on the academic program in which the student is currently enrolled. This entity is capable of looking into its current GPA, list of courses taken with their corresponding credit quantity and grade obtained, calculating possible GPAs by adding possible courses and their respective outcomes, and updating their GPA by adding new courses and their final outcome.

   Type: Student

   Function:

   - **Type:** course, grade
   - **value**: add_course: Course x Grade

3. An *Academic Program* is defined as a combination of requirements and courses leading to a degree to be obtained by the student. Each academic program is composed of Major courses and General courses.

   Type: Academic_Program

4. An *Undergraduate Student* is a student that is enrolled in a higher education degree program, either a Bachelor's degree program or an Associate's degree program.
5. A *Graduate Student* is someone who has previously completed a Bachelor's degree program and is currently enrolled in a, more specialized, graduate program.
6. The *Grade Point Average (GPA)* is a number that represents the students performance on the courses the student has taken. The GPA number ranges from 0.00 to 4.00, where 4.00 is the highest score to achieve. This number is calculated using the number of credits corresponding to the approved courses taken by the student and the honor points from the approved courses resulting grades. There are two types of GPA, general GPA and major GPA.

   Type: Float

   Function: Calculate GPA.

   - **Type: List of courses**
   - **value**: calculate_gpa: List of courses_taken-> **Float**
7. *Major GPA* corresponds to the GPA calculated using only the approved courses that correspond to the student's major.

   Type: Float
8. *General GPA* corresponds to the GPA calculated using all the courses approved by the student.

   Type: Float
9. A *Credit* is the unit of time for a course where the total amount of credits for a course correspond to the total number of contact hours taken in class.

   Type: Integer
10. A *Grade* corresponds to the letter grade obtained at the end of the semester for a respective course. Grades translate into honor points which are then used to calculate the student's GPA. The letter grades are A, B, C, D, and F. Each grade has a corresponding weight associated with it (4 ,3, 2, 1, 0), respectively, which will be later used to calculate honor points.

    Type: String
11. Honor points are the result of multiplying the weight corresponding to the course's final grade and the amount of credits of the respective course.

Type: Integer

12. A *course* corresponds to a class to be taken by the student. Each course has a specific ID formed by concatenating the department's ID and the course code, a specific number of credits, and results in a grade for the student.

    Type: Course

13. *Courses Taken* are courses that a specific student has taken. This entity includes course details, grade obtained, and term taken.

    Type: List of courses

14. A *department* offers different academic programs that are related to a topic or domain.

    Type: String

Requirements

As a student, I want to be able to look for my current GPA standing, courses taken with their respective grades and credits, and be able to calculate possible GPA outcome scenarios given more courses taken and their respective outcomes.

Rough sketch requirements as user stories

As a UPR student, I want to be able to look for my current GPA without having to go through the problems that using Putty and accessing the university's server might cause. To avoid these problems, I want to be able to see my current GPA instantly without using Putty, and calculate possible GPAs by adding more courses and their respective credits and grades to the GPA calculation. Also, I would want to see the grades obtained in each of the courses previously taken.

Requirements as User Stories

1. Users shall be able to register for an account.
2. Users shall be able to log in once an account has been created.
3. Users shall be able to keep session alive even when exiting the application, unless decided otherwise or the user logs out.
4. Users shall be able to navigate application views which include, but are not limited to, profile, curriculum, enrolled courses, settings, and logout, once logged in.
5. Users shall see their name, current GPA, academic program, and UPR campus on the profile view.
6. Users shall be able to add or remove courses to their curriculum.

7. Users shall be able to add a custom course.
8. Users shall be able to see an update on their GPA, if the value actually changed, when they add a new course.
9. Users shall be able to add or remove courses to enrolled courses view.
10. Users shall be able to modify, add, and delete grades to enrolled courses respective evaluations.
11. Users shall be able to modify settings that include, but are not limited nor constrained to, university campus and academic program.
12. Users shall be able to log out.

Software Design

a. Software Architecture

   i. The aim of the software to be developed is to solve the problem that arises from students having to use Putty and connect to the university's server to view their GPA. With that in mind, the team has decided to build a Web application that will allow students to view their GPA, add courses and their respective grades, view their grades obtained from added courses, and basically view their academic program's curriculum graded as they take courses and add them to their website account.

   ii. To that aim, and given the skills and knowledge that each team member has, the team decided to use the Python programming language and Django as the Web framework. Python was chosen as the programming language since most team members had some knowledge on Python already. On the other hand, Django offered some great documentation and tutorials, while also a team member had some previous experience using the tool as well. This Web framework provided some tutorials on connecting with databases as well, and since the system to be built needs to store user data, it only boosted its chances to get used. Since user data needs to be stored, a database management system needed to be added to our software architecture. To that end, the team decided to use PostgreSQL, since some members of the team had already used the mentioned DBMS. A question arises, and is that where does the Web application store user data. The most viable solution is using serverless computing, in which the cloud provider runs the server

and manages the server resource allocation. The cloud provider to be used has not yet been determined.

iii. The Web application can be divided into different views that when joined together form the complete web-application. Sketches and descriptions for these views can be seen below.

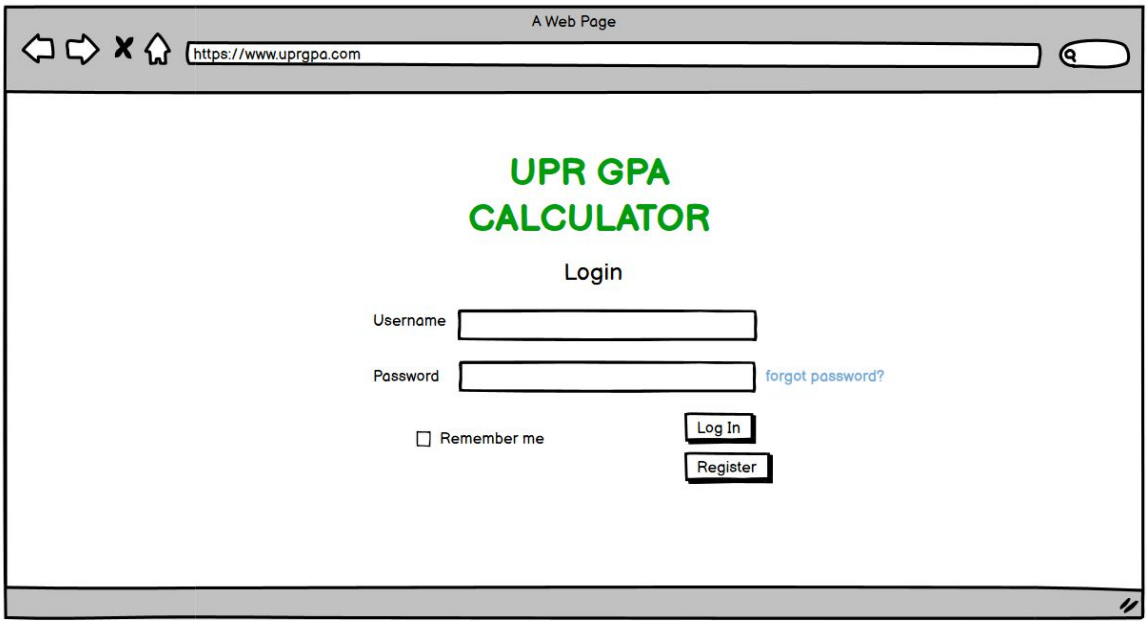

Image 1. **Login View -** User logs in if account has been previously created, if not, selects register to create a new account.

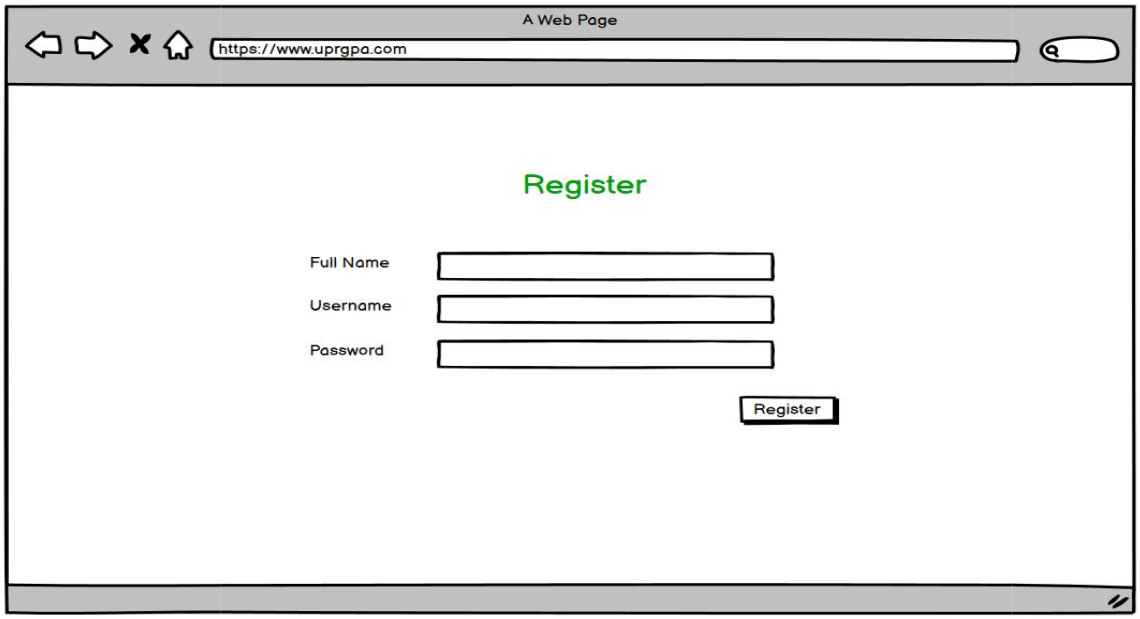

Image 2. **Register View -** User creates new account to be able to use the Web-Application.

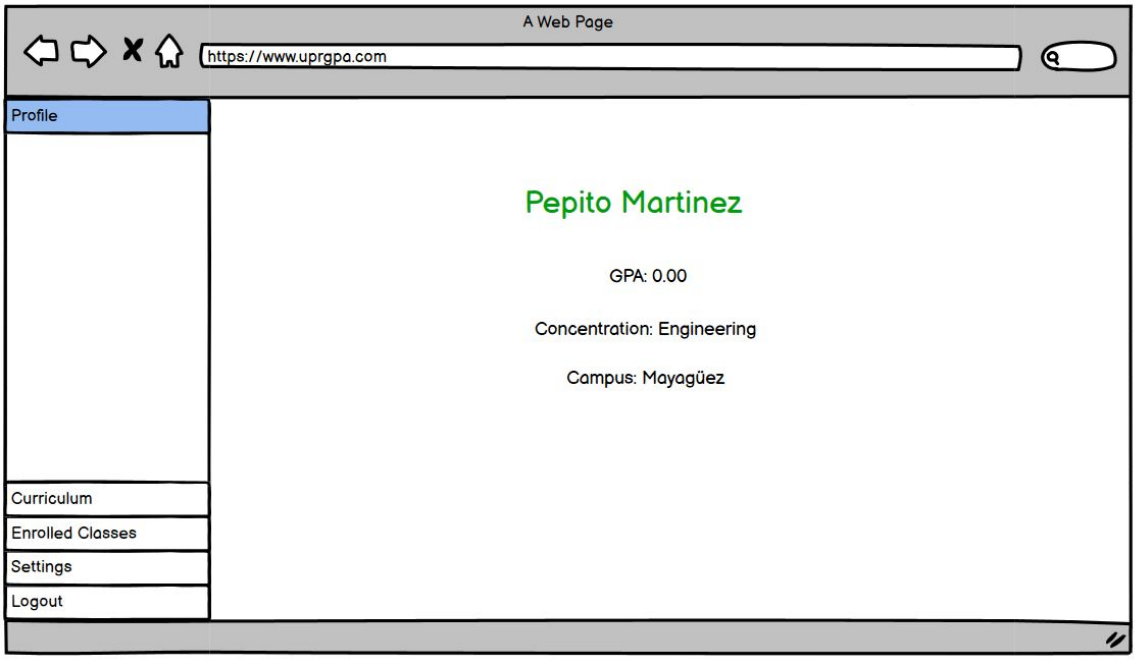

Image 3. **Profile View -** User sees details about his profile.

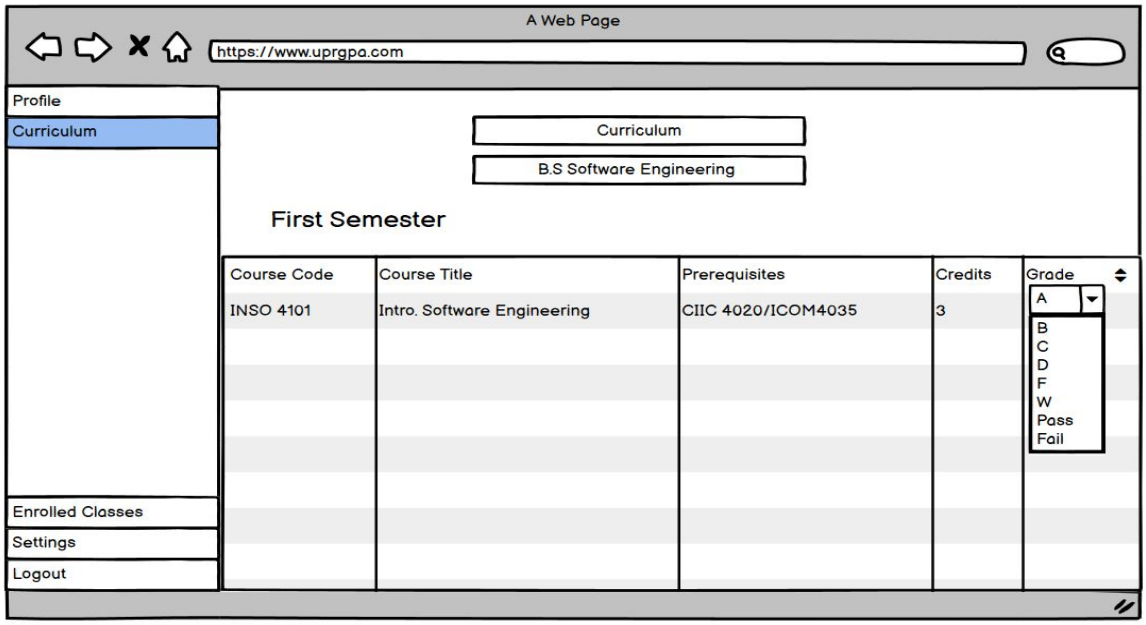

Image 4. **Curriculum View -** Users add, modify, delete, and see their courses taken and their resulting grade.

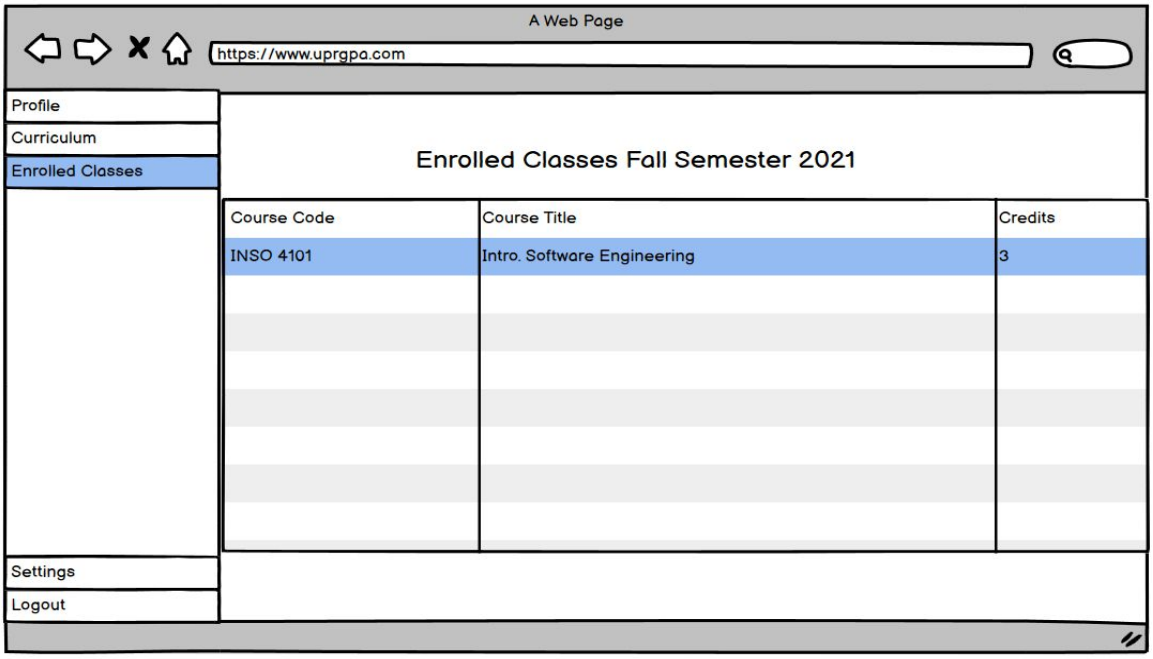

Image 5. **Enrolled Courses View -** Users can see, add, modify, and delete courses currently enrolled in.

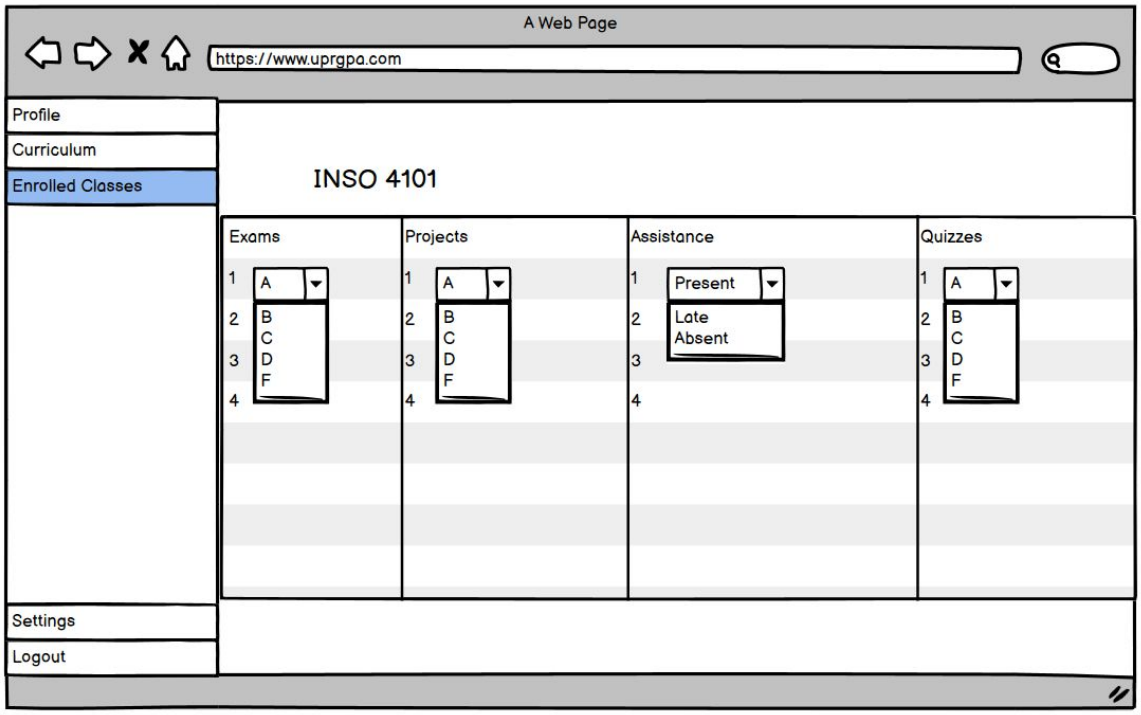

Image 6. **Course Details View -** Users can see a detailed view of a particular course and add evaluation results obtained in that course.

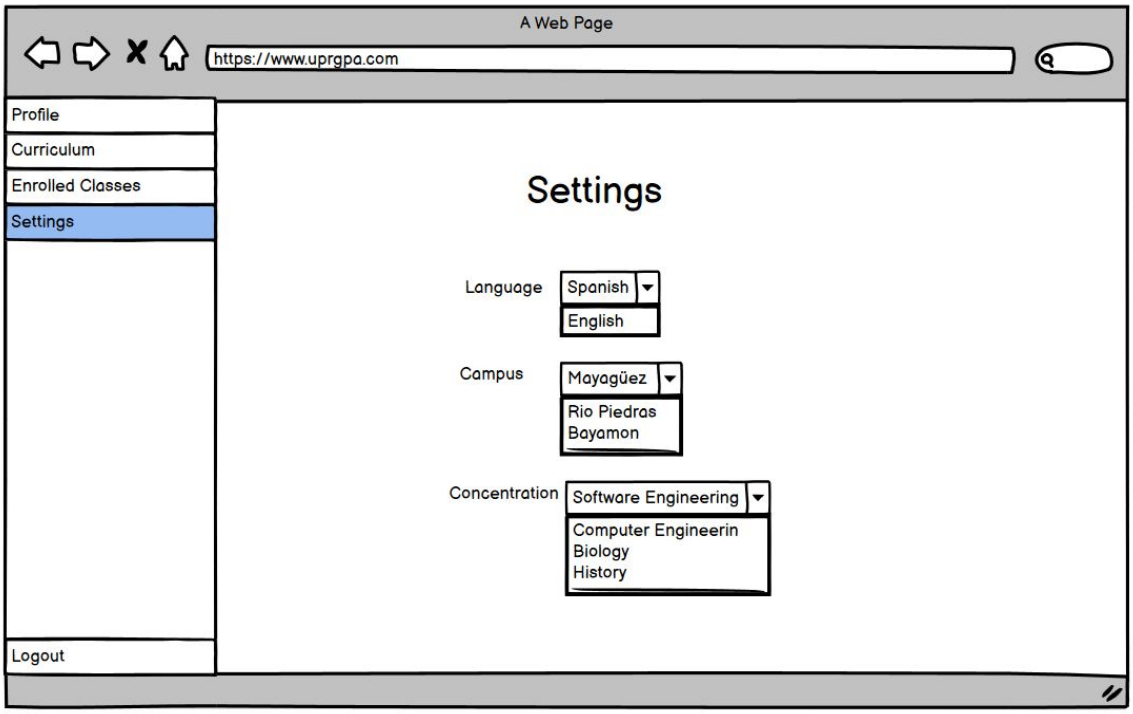

Image 7. **Settings View -** Users can see current settings and modify them.

Selected Fragments of Implementation

Initial implementation fragments have been implemented to set up the Web application skeleton. This includes downloading and installing latest versions of software tools to be used, which, at the moment, include Python 3.9, PostgreSQL 13.1, Django 3.1.6, and PyCharm Enterprise edition. Also, the project was created and pushed to Git origin. Finally, the first issue pertaining to the application skeleton and some initial work to set up Django was pushed as the teams first issue.

III. Analytic Part

Concept Analysis

A system that helps students keep track of their classes, grades, and GPA consists of a set of classes that correspond to a specific curriculum, a set of students that each have their own classes taken as well as their GPA, and a set of curriculums that have their own pertaining classes.

Verification

The current system that the university and its community have is one that causes many problems and it is not active all-year round. Students can access Putty and obtain certain information such as GPA, classes taken, or grades obtained, but as mentioned above, the system is not always available. There are times when students need quick access to this up to date information when searching for internship and coop opportunities. Besides this, students also need this when applying for different scholarships or grants. The current system offers a command line interface that can prove complicated to first time users. For instance, a graphical user interface can improve the user experience by allowing a much friendlier view of the data. For these reasons we need a more reliable and simple system to get this information.

Validation

The proposed system can be exceptionally useful especially in our current position where we depend on only one archaic system that is already being overburdened by the enrollment process. It has been evident that every time the enrollment process starts, Putty instantly crashes. Also, the vast majority of students agree that there is a necessity for a simpler, more reliable, and modern solution.