

EnrollEase

(Smart Class Registration)

Erick Vega

Senior Project

Brigham Young University – Idaho

Winter 2024

EnrollEase is a web application dedicated to help students simplify their class registration process and optimize the time they spend between classes.

Section 1

INTRODUCTION

Purpose

Simplify the class registration process at the beginning of each semester through a modern and simple web interface by generating unique set of classes optimized to reduce walking distance and gaps between classes.

Scope

This is a system which facilitates a student creating a schedule for a semester. This program allows the students to place the required class in a semester plan and sends the semester plan helping the use their university's registration system to register their classes. It also allows the student to auto-generate a plan considering the location of a class, the time between classes, and other student-specified preferences.

The system does not provide authentication services and does not verify pre-requisites for any course.

Overview

By the author's own experience as a college student, registering for classes can take an unnecessary enormous amount of time, especially if factors like, distance and gaps between classes or professor's ratings are taking in account. Currently, adding classes to a schedule is a try-an-error activity, when students want to add a class, they usually look up the professor's name in sites like RateMyProfessor.com to evaluate other student's previous experiences with such professor. Once a preferred professor is chosen, they have to make sure that the class does not overlap with any other class, and finally they add the class to their schedule. The surprise is that following this approach can leave little room for schedule customization. What if a student chose a class in the south of his campus and fifteen minutes after that class, he has to walk ten minutes to the north side of campus to finally return to the south side of campus, that is a painful experience in terms of time. The only known way to create an optimal schedule with all the features mentioned above, is manually. This web application targets the common pain points associated with time-consuming registrations and suboptimal schedules. By leveraging technology to automate the process, EnrollEase ensures that choosing and registering for classes takes only a few minutes, significantly reducing the hours currently spent on creating a "smart" schedule. The app's intuitive interface allows students to log in, select desired classes, set schedule preferences, and generate optimal schedules effortlessly. EnrollEase is designed for any student globally who frequently faces the challenge of selecting classes. The application is compatible across major web browsers and mobile devices. The goal is to offer a streamlined solution for students to register with just a single button click. The project's success is defined by its ability to provide at least one optimal schedule when a student selects a set of classes, prioritizing non-overlapping classes, and minimum gap and walking time between classes.

Standards

EnrollEase will follow the MVC Architecture for its development. MVC Architecture stands for the components Model, View, and Controller. This architecture has proven to be simple and efficient for a basic web application and meet the needs of the current scope of the project.

This project will use Bootstrap 5.0 as its front-end framework and Django for its back-end framework, both have proven to collaborate tightly on web development and are compatible with Python which is the main programming language used for our application.

Section 2

DEFINITIONS

Web Interface: Allows the user to interact with content and software running on a remote server through a Web browser.

System: The EnrollEase system defined in the Scope.

Optimal Schedule: A collection of courses optimized as defined in the Scope.

Walk Time: The distance between classrooms as measured in how long it takes the average person to walk.

Gap Score: Numeric value attached to an Optimal Schedule computed by averaging the time between courses.

Preferences: The student's preferred way of taking courses, including walk time, professor evaluation, and gap between classes.

Professor Score: A numeric value attached to an Optimal Schedule computed by the Rate My Professor scores of the instructors of the suggested courses.

REQUIREMENTS

1. The **Web Interface** shall utilize the student's university log-in page for authentication.
2. The **Web Interface** shall include the following in the menu of option: Student Profile, My Schedule, Schedule Generator, and Log-out.
3. The **Web Interface** shall present informative error messages when is unable to perform an action.
4. The **Web Interface** shall accept a list of classes from the user.
5. The System shall find non-overlapping classes for an **Optimal Schedule**.
6. The **System** shall compute **Walk Time** for an **Optimal Schedule**.
7. The **System** shall compute **Gap Score** for an **Optimal Schedule**.
8. The **System** shall take available courses from the student's university website.
9. The **System** shall suggest a set **Optimal Schedule** from an **Optimal Schedule** list.

STRECH REQUIREMENTS

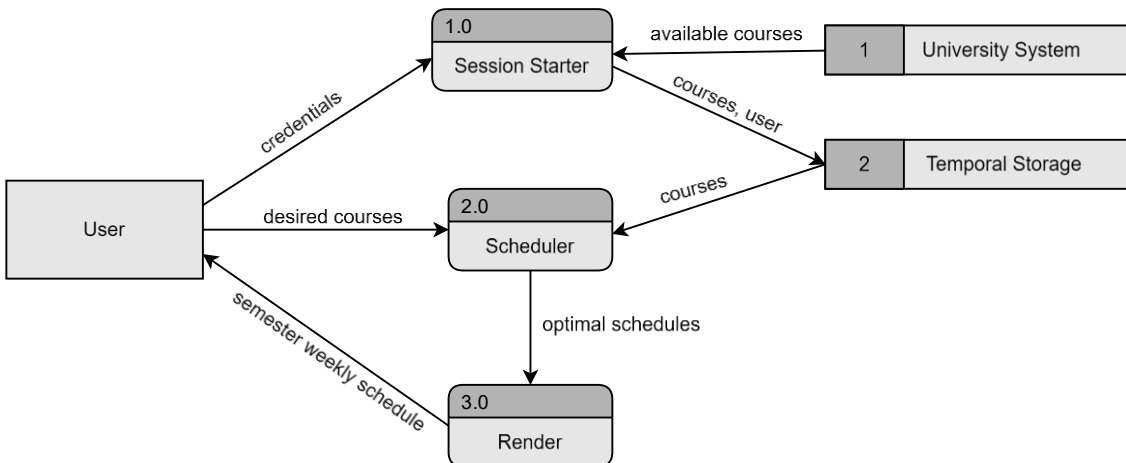
1. The **System** shall allow the **student** to edit and update **Preferences**.
2. The **System** shall generate an **Optimal Schedule** based on **Preferences**.
3. The **System** shall retain the **Student's Preferences**.
4. The **System** will compute **Professor Score** for a **Semester Plan**.
5. The **Web Interface** shall contain a print **Optimal Schedule** feature.
6. The **Optimal Schedule** view will contain a direct registration feature.

Section 3

DESIGN OVERVIEW OF THE PRODUCT

Workflow

When the user accesses the application through the web, initially, is taken to a log-in page, these **credentials** will allow the web-scrapper in the **Session Starter (1.0)** to grab all the available courses from the **University System (1)**. When the user chooses a set of **desired courses**, the **Scheduler (2.0)** computes all the possible combinations of courses that the student could take and sorts them by Walk Time and Gap Score. Once the computation is done, the top **optimal schedules** are sent back to the user for revision through the **Render (3.0)** which is the Web Interface.



Data at Rest

User saved schedules, preferences, and data from the University's registration system will be stored in a SQLite3 database.

Data on the Wire

Data will be processed mainly by the Scheduler (see 2.0 in the Data Flow Diagram), this will make sure that the data is consumed and sent to the right unit of local storage. Session data like available courses will be deleted by the Browser.

Resources

Python: Main language of the application supported by Django, the back-end framework.

JavaScript: Handles dynamic changes of the HTML structure of the web interface.

HTML: Structures the content of the interface.

Selenium: Web-scrapper used to take user to the university log-in page and get all available courses.

Bootstrap: Front-end framework with predefined styling features for HTML content.

SQLite3: Stores course, section, student, and optimal schedules information.

Heroku: Platform as a service that enables to build, run, and operate the application in the cloud.

Visual Studio Code: IDE used for testing and development of the application.

User Interface

The user interface is simple and straightforward. The user can see his current selected optimal schedule or generate new ones. Initially, the student selects a set of courses needed to be taken on the **Choose Courses** section semester and click **Generate Schedule**, then a set of courses (in this case 5) will be presented where the user would be able to select it based on his preferences and the information provided in **Schedule Details**. The interface will allow the user to navigate through the different optimal schedules with arrow buttons.

The screenshot displays the EnrollEase web application. The header includes the EnrollEase logo, navigation links (Home, My Schedule, Schedule Generator, Donate), a search bar, and a user profile link. The main content area shows a message: "We found 5 schedules for you!". Below this is a calendar view with columns for Monday, Tuesday, Wednesday, Thursday, and Friday. The calendar shows a schedule for a student, with classes listed for each day. The classes are: BUS100_01 Class Title (1015-1115) on Monday, Tuesday, Wednesday, and Thursday; CSE382_01 Class Title (1245-1345) on Monday, Wednesday, and Friday; and CHLD210_02 Class Title (1400-1500) on Monday, Wednesday, and Friday. The interface also includes buttons for "Choose Courses", "Generate Schedule!", and "Schedule Details".

	Monday	Tuesday	Wednesday	Thursday	Friday
7 AM					
8 AM					
9 AM					
10 AM	BUS100_01 Class Title 1015-1115	ED444_01 Class Title 1015-1115	BUS100_01 Class Title 1015-1115	ED444_01 Class Title 1015-1115	
11 AM					
12 PM					
1 PM	CSE382_01 Class Title 1245-1345		CSE382_01 Class Title 1245-1345		CSE382_01 Class Title 1245-1345
2 PM	CHLD210_02 Class Title 1400-1500		CHLD210_02 Class Title 1400-1500		CHLD210_02 Class Title 1400-1500
3 PM					
4 PM					
5 PM					

Section 4

VERIFICATION

Demo

The application will work as designed when a student not related to this project is able to:

1. Type the URL of the EnrollEase website and open the site without errors.
2. Log in into his BYU-Idaho account.
3. Select courses for his/her next semester.
4. See at least one optimal schedule with a list of courses.
5. Register successfully to all the courses of the optimal schedule in the University system.

Testing

1. The application is accessible through the web by more than one device.
2. The user is redirected to BYU-Idaho log-in and back to the application on first interaction.
3. The courses suggested on the Optimal Schedule are valid and real, therefore they can be added into the actual University registration system without any errors.
4. A set of optimal schedules contains courses which locations on campus are distributed optimally, this will be proven with a map of the campus.
5. The time from logging to registering is less than 5 minutes.