YorkU Scheduling Tool

Group #3

Zayn Ali, Mirza Baig, Ayokunlemi Fakuade, Caleb Jones, Ahmet Karaca, Mateo Lobato

YorkU Scheduling Tool Vision Statement

‘YorkU Scheduling Tool’ is a web application designed to optimize the experience of building your course schedule for an academic session. The site will generate schedules based on their courses and other preferences. It will also streamline other aspects of the course enrollment process to allow users to get all the information they need in one place.

This tool is meant to be used exclusively by York University students during their course selection. The student will be able to search for courses which are available, and they will be able to access all public information about the course. Then they will allocate the time for each activity in the course or use suggested times. Based on all information the user has provided, schedules will be algorithmically generated.

Outside of timing, our site aims to incorporate other considerations for students when building the schedules. One of these is peer opinions on the course and their experiences with the instructor, which is currently accomplished through word-of-mouth or searching on social media. Another is social features, which would allow users to directly compare their schedules with friends to easily ensure common courses or breaks.

For compatibility and ease of access, we plan to make a web application suitable for personal computers and phone browsers. Web development also creates an opportunity for the use of libraries for interactive and smooth user interfaces.

The purpose of this web app is to create an alternative to the existing process of building a schedule, which requires students to navigate through many different resources. Our all-in-one tool aims to reduce the existing stresses and frustrations associated with course enrolment at York University.

This success of our application will be considered a success based on two metrics. The first and most important is that users find our system more convenient to use than the existing ones. The second metric is that we hope to reduce the time it takes to build a schedule from start to finish by 30% on average.

Big User Stories:

Plan With Friends

As a student, I want to easily plan my schedule in parallel with my friends. I want to see their schedules compared directly with mine.

Priority: high Cost: 10 days

Create Specific Schedule

As a student, I want to be able to build a schedule based on the courses I add and various other personal preferences.

Priority: high Cost: 9 days

Course and Professor Insights

As a student, I want to access insights from my peers on instructors and course content. I would also like to leave my own insights for future peers taking the same courses/instructors.

Priority: high Cost: 9 days

User Stories Iteration 1:

Personal Scheduling Preferences

As a student, I'd like to be able to block out times for courses,

Priority: high Cost: 0.5 days

Calendar Interface

Be able to display, at a glance, a user's weekly schedule.

Priority: high Cost: 0.5 days

Account System

Create an account to save user information between sessions. Also fetch information when logging in.

Priority: high Cost: 0.4days

Social Connections

Allow users to be able to connect with their friends. Keep a record of this in the account.

Priority: medium Cost: 0.35 days

Schedule Statistics

Display statistics on how long a user may spend per day in class and between classes.

Priority: low Cost: 0.15 days

Leaving Ratings

Allow users to leave ratings on courses for future students to view and see reviews from peers.

Priority: medium Cost: 0.20days

User Stories Iteration 2:

Better Time Selection

As I student I want to it to be easy to map out times on my own when I am adding a course.

Priority: high Cost: 0.5 day

time average option

As a user I want the time average feature to be an option rather than something manual.

Priority: low Cost: 1 day

More schedule options

As a student I want to be able to create schedule times and custom activities.

Priority: medium Cost: 0.5 day

View friend schedules

As a student I would like to see my friends schedules or common courses.

Priority: medium Cost: 0.5 day

Improve friend features

As a user I want the ability to remove friends as well as accept and deny pending friend requests on the UI.

Priority: medium Cost: 0.5 day

Viewing reviews

As a student I want the UI to have lists of reviews for each course available.

Priority: high Cost: 0.5 day

Working with the PostgreSQL and the Database:

After some time, it was necessary to implement a working database to supersede the stub one created previously.

An SQL script was designed to create and structure a PostgreSQL database named ystTestDB, which supports a course review and scheduling platform. The script begins by ensuring that any existing database with the same name is removed using DROP DATABASE IF EXISTS, preventing conflicts with previous versions. It then creates a fresh instance of the database and establishes a connection to it with \c ystTestDB.

Three tables—reviews, courses, and accounts—are defined to store user-generated reviews, course information, and account details. The reviews table contains attributes like review_id as the primary key, author, date, course, and review-specific details such as description, difficulty, and quality. The courses table assigns a unique courseCode as its primary key, stores course title and sections in a JSONB format for flexible data storage, and maintains an array of review IDs linked to it. Additionally, it tracks aggregate difficulty and quality ratings. The accounts table manages user credentials (username_email, password), course preferences (coursePrefs stored as JSONB), and social interactions through friends, requests, and reviews, all stored as arrays for efficient querying. This structured design ensures that the platform can effectively manage user interactions, course data, and review functionalities.