



COMP4801 Final Year Project [2023/24]  
Detailed Project Plan

## Re-design of HKU's Course Enrollment System with More Features

Supervisor: Dr. Tam, Anthony T.C.

Students: Zhang Zhijie (UID: 3033093416)  
Xiong Hongruifeng (UID: 3035771921)

## 1 Project Background

Course enrollment is a perennially hot topic among students at HKU as each semester approaches. Students require resources to plan their course selections in advance, allowing them to manage their studies effectively. Access to course information such as timings, content, and instructors is crucial for students to arrange their schedules wisely.

Currently, there are several online resources available for students to refer to. The respective faculty websites provide detailed information about each course, including course descriptions, schedules, and instructor details. The school's course selection system also maintains a comprehensive list of available courses, offering basic information and links to course websites. This system also allows students to add the courses they would like to enroll in only before the start of the enrollment period.

However, these existing tools have their limitations, leading to student complaints. The first problem is the poor user interaction in the school's course enrollment system. Many students find the course selection procedures cumbersome and experience slow response times. An inefficient design significantly impacts the user experience. For instance, adding a course to the "course shopping cart," which stores selected courses for submission, often involves multiple jumps. Users need to look for their wanted courses from a very long list with a small font size. To view the course details, they must click the select button to check. After that, they must click another button to add the course to the "shopping cart." This process flow can consume significant time, especially when the network is slow, or the system is during peak hours. Also, this process will be repeated for each intentional course.

Another problem is the school system will not notify the students if their course selection is valid until the start of the enrollment period. For example, a student cannot add a course since he/she has not fulfilled the pre-requisite yet. The failure to send an enrollment request can cause big trouble for students because students lose an opportunity for an early submission in a competitive enrollment battle.

## 2 Project Objective

By developing this integrated system, we aim to streamline the course selection process, enhance user experience, and provide students with a more efficient and user-friendly platform to plan their studies effectively. This system can be divided into three parts: a login system, a course selection platform, and a course information system.

First, the system will feature a new UI design, including an improved visual rendering of the shopping cart. Unlike the school system, the shopping cart will not be contained in a list. We will use a calendar view to store the selected course and use different colors to annotate the status of each class (enrolled, pending approval, etc.). Also, the estimated quota prediction will be provided in the course list. This system should be able to display if the pre-requisite has been fulfilled during the selection process.

This system should also be able to bear the high blasts of user traffic. At the beginning of the enrollment period, there usually are many users to be online and send the HTTP requests to the server. To enhance the user experience during peak periods, we will incorporate high-concurrency mechanisms like flash sale systems. This will reduce waiting times and ensure an accurate participation quota.

Last, this system includes a recommender system that gives students course selection suggestions. Each student needs to view hundreds of courses before the enrolment. Aiming to reduce this tedious process, the system will try to recommend the courses which the student is likely to gain good grade in and are in line with the student's interest.

## 3 Project Methodology

### 3.1 Front-end

For the front-end development, we will utilize Vue 3 as the primary framework for constructing the web pages. This framework is built upon the foundations of HTML, CSS, and JavaScript. By employing Vue 3 and external UI components like Vue-Cal and Ant Design, we aim to create a visually appealing and user-friendly web application that provides students with efficient access to course information and management.

The user interface will consist of three main views. The initial page presented to users will be the login page. Once logged in, users will be directed to the system's main page. The navigation bar will be positioned on the left side of the page, while the main content will be displayed on the right. Students can switch between the course calendar and course list views through buttons located in the navigation bar. This switching mechanism will be implemented using Vue's router mechanism.

To create other components of the webpage, such as layouts, buttons, and tables, we will import Ant Design. This UI library offers a wide range of pre-built components that will enhance the overall aesthetics and functionality of the application. In the course list views, students will be able to browse through course information and details, and they will have the option to add or remove courses from their shopping cart. For the class timetable, we will import Vue-Cal. This component will visually represent the course calendar using color blocks to indicate the status of each course. Clicking on a block will trigger a floating window that allows students to view the details of the selected course.

### 3.2 Back-end

For the back-end, we will develop a web application utilizing Java Spring Boot. The back-end communicates with the front-end and maintains the database.

With an API to the front-end, the back-end will provide student information and course information to the front-end. When handling the enrollment request from the front-end, the back-end will need to check the time clash and pre-requisite. Multi-threading will be implemented to fulfill the high concurrency requirement.

The database contains the student information, the course information and the enrollment information. The student information includes the identity information and the course history. The course information includes course code, course title, course description, course provider, course feedback and provided schedule. The enrollment information includes the students who enrolled in each course. The student information and the course information are static and will be prepared before the course enrollment period, while the enrollment information will be maintained during the course enrollment period.

The back-end will also calculate the correlation table for the recommender algorithm in advance and send it to the front-end.

### 3.3 recommender algorithm

The recommender algorithm aims to find out the potential high-grade and in line with interest courses for the students.

The algorithm consists of two parts. The first part, based on an existing large language model, will calculate the correlation between two courses relying on their course title and course description. A table contains the correlation of every two courses that will be produced by the backend. The second part is located at front-end. With the assumption that students can get higher grade in a course similar to their previous well perform courses, the algorithm read the course history of student and find out the best performed courses and gives out recommendation according to the correlation table obtained from back-end. Other factors like the course feedback, provided time and so on will also be considered.

#### 4 Project Schedule and Milestones

Milestone	Due date
Learn necessary skill	30/09/2023
Front end	01/01/2023
Back end	01/01/2023
Stress test	01/04/2024
Recommender algorithm	01/04/2024