COMP4801 - Final Year Project
Detailed Project Plan

Web Application for Planning Semester Schedule and AI-driven Chatbot

Supervisor: Prof. Tam Anthony

Member: Chan Yuen Kwan

Tsang Ching Yu

Wong Ying Pui

Table of Contents

1. Background

As each academic year approaches, students face the task of planning and organizing their semester schedules. This process, however, is not without a multitude of challenges. With an ever-expanding curriculum, varying prerequisites, and the desire to balance academic commitments with personal and extra-curricular responsibilities, students often find it daunting to create an efficient and personalized course schedule.

Central to this process is the HKU Portal, the centralized platform that also serves as the student course planning system at the University of Hong Kong. It offers mainly two functions: “Course Information” and “Enrollment Add Classes”.

The “Course Information” function serves as a search tool, where students can input the course name, course code, or the specific curriculum. It retrieves matching courses and presents comprehensive information relevant to the course, including but not limited to course schedule, prerequisites, and assessment methods.

For “Enrollment Add Classes”, the user first select the appropriate semester, after which in the “Search by My Requirement” page, the system provides a simplistic view of all courses offered in the chosen semester, grouped by their “Course Categories” as specified in the user’s curriculum, e.g. “Disciplinary Core”, “UG5 Requirement”, and allows the user to add classes to the “Temporary Course List”.

In this view, users can only view the name, code, unit and offering semester. Detailed information of the course, such as class time and location, is not visible until the student clicks on the course and goes to the “Course Detail” page. There is no pre-selection or recommendation of schedule for students. Students have to manually select each course they desire one by one, having planned the schedule on their own beforehand. The curriculum-based categorization may also be confusing as the system displays all courses under the category label, regardless of whether or not the user has fulfilled the prerequisite. The information displayed on the “Course Detail” page, as compared with the “Course Information” result, is quite lacking, failing to offer insights into course prerequisites, instructor information and other crucial elements. Students often need to cross check information between the two pages. Moreover, a calendar view of the schedule is not available until the student starts enrolling into the courses. The lack of visualization, where information is presented in a plain text block, results in a cluttered and challenging-to-digest format, further complicating the course planning process for students.

Overall, it is evident that there is a significant need for a better course planning tool.

2. Project Objective

The objective of this project is to develop a web application for course selection scheduling, integrated with an AI-powered chatbot. The application aims to provide comprehensive support to undergraduate students at HKU, assisting them in planning their course registration and offering individualized academic and career guidance. Ultimately, the project seeks to explore the possibility for integration with the current HKU Portal system, with the goal of incorporating these features for enhanced functionality and an improved user experience.

3. Features design

3.1. Timetable Planner

The Timetable Planner feature will be designed to assist students in planning their course registration at HKU. It will provide a visual representation of the course schedule, allowing students to easily review and optimize their selections. By considering factors such as time conflicts, course popularity, prerequisites, and constraints, the Timetable Planner will suggest alternative course sections and resolve conflicts to improve overall schedule efficiency.

3.2. Course Recommendation Engine and Conflict Resolution

A course recommendation engine will be integrated, automating the generation of a semester-specific course list. This process will be guided by users' individual course history, major or minor prerequisites, and tailored preferences. This feature will enable users to efficiently select courses, optimizing their time. Moreover, the system will possess the capability to address conflicts, such as full classes or scheduling conflicts, by offering alternative solutions to users.

3.3. AI-Powered Chatbot

An AI-powered chatbot will be developed, serving as an intelligent virtual assistant for students. It will provide students with real-time support, answering some Frequently Asked Questions (FAQs) about course enrollment and course information, and giving out career advice. This feature will be designed to enhance overall user experience and provide students with a quick and convenient access to information and assistance.

4. Methodology

This chapter shows the implementation and technique for this project.

4.1. Platform setup/ software involved

The system is a web application built with **Python**, a language renowned for its extensive libraries and packages conducive to AI chatbot development, as well as its capabilities in web frameworks [1].

For the frontend, **React** will be utilized. This framework boasts a wealth of libraries for crafting UI components. The utilization of Virtual DOM enables the system to update efficiently without affecting the entire structure, resulting in reduced processing time, particularly during timetable generation [2].

For the backend, **Django**, a Python web framework, will be utilized. It offers a range of valuable features for development. This encompasses a robust user authentication and permission system, bolstering security. Additionally, Django provides an administration site by default, facilitating easier web maintenance for administrators [3]. Furthermore, the framework excels in serializing and serving data as JSON [3], which proves invaluable when client-side code takes charge of data rendering.

For AI-Powered Chatbot, the core of the chatbot relies on the OpenAI API, which provides access to GPT-3.5, incorporating cutting-edge natural language processing (NLP) capabilities into the web-based chatbot. GPT-3.5 is chosen as the AI model for the project due to its state-of-the-art NLP capabilities like comprehending natural language queries and generating human-like text responses, making it ideal for such conversational applications.

4.2. UI design

Figma will serve as the platform for GUI design, while **React** will be employed for the actual implementation of the user interface (UI).

The system necessitates a login interface. After login, users will be able to review their academic progress, including the requisite credits for their degree and a recommended study plan generated by the system for reference. Additionally, a timetable generator with customizable input will be provided, allowing users to specify preferences while generating a timetable.

The system will also feature an entry to an AI-driven chatbot. The chatbot's UI will mimic a traditional chatroom, affording users the ability to input text at the bottom.

An exclusive user interface (UI) will be provided to administrators, allowing them to update the class timetable, modify degree syllabi, add new courses and other possible data to the database. Recognizing that re-coding is not the most efficient approach, this admin UI serves as a streamlined tool for managing database updates effectively.

4.3. Data collection

The majority of the data essential for the system is static data or data that only necessitates updating annually.

This encompasses users' information, such as their program of study, major(s), and minor(s), as well as their academic history and corresponding grades. These details are derived from the "**Degree Audit**" uploaded by the user.

Furthermore, course-specific information is imperative, including schedules, timings, descriptions, and enrollment prerequisites. This information is extracted from the **Handbook of Faculty** and **Class Timetable** provided by HKU SIS.

FAQs, including both questions and answers, also form a vital component for the chatbot. These are sourced from various departments' websites within HKU, encompassing **SIS, AAO, and various faculties**. Due to the potential volume of FAQ content, prioritization will be crucial during chatbot training. AAO will receive the highest priority, given its extensive coverage of inquiries that students typically have when registering for courses.

To establish the foundation for the recommendation engine, data will be gathered from past course evaluation obtained from the **course effectiveness profiles**. However, collecting the number of enrollments of each course for immediate recommendations during enrollment period requires collaboration with HKU, which is not currently feasible during this stage of the project, alternative approaches will be explored to provide recommendations based on available data.

4.4. Database

MongoDB will be utilized. It serves as a NoSQL database management application. It stores data in documents utilizing a JSON-like structure for representation and interaction. MongoDB excels in handling intricate, nested data structures and eliminates the need for crafting complex SQL queries during data retrieval, resulting in significant time savings in development [4]. Moreover, MongoDB provides a streamlined approach for text-based data

search, making it particularly well-suited for our system which is heavily reliant on textual information.

The data will be collected and stored in .xlsx, and then imported into the database. The example data structure is designed as below. The structure is subject to change.

User

```
{
  _id: ObjectId(),
  email: String,
  program: [
     {
      programName: String,
      type: String, //major or minor
     },
   ],
  course_history:[
    {
      courseID: String,
      grade: String,
    },
    {
      courseID: String,
      grade: String,
    },
    // Additional history
  ]
}
```

Course Schedules

```
{
  _id: ObjectId(),
  availabilityID: String,
  courseID: String,
  semester: String,
  sections: [
    {
      sectionID: String,
      time: String,
      location: String
    },
    {
      sectionID: String,
      time: String,
      location: String
    },
    // Additional sections as needed
  ]
}
```

FAQs

```
{
  _id: ObjectId(),
  FAQsType: String,
  question: String,
  answer: String
}
```

Course Information

```
{
  _id: ObjectId(),
  courseID: String,
  title: String,
  description: String,
  credits: Number,
  prerequisites: [String]
}
```

5. Project Schedule and Milestones

Date	Milestones
September	• Explore GPT-3.5 Chatbot feasibility
October	• UI/UX design • Data collection and preprocessing • Database structure setup • Web setup • Basic project setup for frontend and backend
November	• Basic implementation of core features • Timetable planner implementation • Chatbot development
1 - 15 December	• Recommendation algorithm design
16 December – 31 January	• Preparation for first presentation (8-12 Jan) • Complete Preliminary implementation and Detailed interim report (21 Jan) • Integrate backend and frontend of application
Febuary - March	• Functionality and performance testing • Improve and refine the system
April	• Preparation for final presentation (15-19 Apr) • Complete final report (23 Apr) • Project exhibition (26 Apr)

6. Reference

[1] “Find, install and publish python packages with the python package index,” PyPI. [Online]. Available: https://pypi.org/ [accessed Sep. 25, 2023].

[2] “Reactjs reactdom,” GeeksforGeeks. [Online]. Available: https://www.geeksforgeeks.org/reactjs-reactdom/ [accessed Sep. 27, 2023].

[3] “Django overview,” Django Project. [Online]. Available: https://www.djangoproject.com/start/overview/ [accessed Sep. 27, 2023].

[4] “The developer Data Platform,” MongoDB. [Online]. Available: https://www.mongodb.com/ [accessed Sep. 27, 2023].