

**CSCI3100 Software Engineering  
High-Level Design Documentation  
Group B7**

**CU advanced Course Selection System**

Junwoo Yun, 1155124295  
Taewhei Kim, 1155114082  
Jounghoon Kim, 1155096482  
Taehun Lee, 1155123419  
HyunJun Yoo, 1155100531

Feb 1, 2023

Version 1.0

Department of Computer Science and Engineering, The Chinese University of Hong Kong

# **1 INTRODUCTION**

## **1.1 Project Overview**

This project is a web-based application for course selection system which enables students to browse and select courses offered by the university, and admin to manage the student and courses. It would be an ultimate replacement of CUHK's course selection system, CUSIS.

Our system follows a client-server architecture. Clients interact with the user through UI and sending API calls to the server. Server receives API calls, query and receive data from the SQL database. It would have additional features such as course recommendation and review pages which would enlarge the user experience.

## **1.2 System Features**

The main will contain login features. After login, the application will display two main pages for users - course browsing page and the profile page.

In the course browsing page, users can search eligible courses based on course ID, name, time, department etc. The course information includes course ID, course name, time, place, department, instructor and capacity.

Eligible courses (for example, courses with no capacity will not be able to be selected) can be selected, and added by users.

The profile page will display courses selected by users and also will contain features to drop selected courses by pressing the drop button.

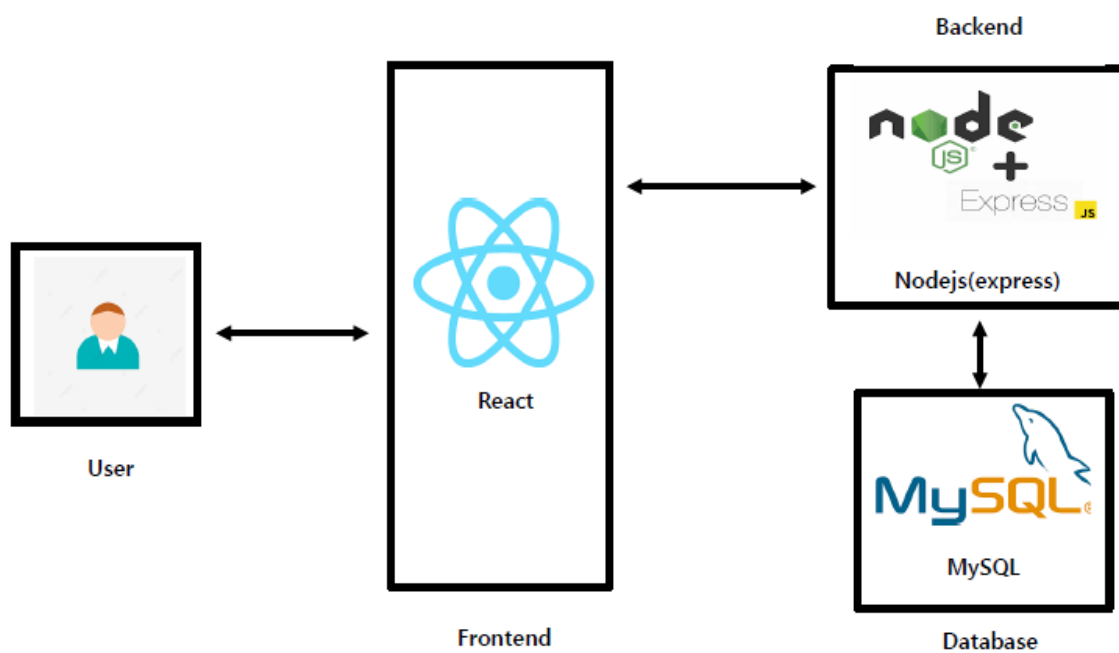
There will be two types of users - normal users and admin users. Normal users can do all the basic functionalities mentioned above, and admin users can view all course/user information and add or delete courses/users if needed. For this feature, admin users will have an additional page to display such information. Our application's advanced functionalities include course recommendation system, concurrency control and course review system with user-friendly and appealing UI.

## 2 SYSTEM ARCHITECTURE

### 2.1 Technologies

1. SQL Database management system (mySQL)
2. User Interface (ReactJS, HTML, Bootstrap)
3. Data transfer technology (Hypertext Transfer Protocol)
4. Programming languages (Javascript)
5. Web Application Services
6. Operating systems

### 2.2 Architecture Diagram



### Components

1. User
2. Client
3. Server
4. Database

## Communication

- User ↔ Frontend
  - Communicate through HTTPS to render the web browser and receive user input actions by the front-end server.
- Frontend ↔ Backend
  - API request and response throughs HTTPS
- Backend ↔ Database
  - Communicate through Sequelize ORM (search, query, store)

## Features

- Authentication
- CRUD action (User management, Add/Delete User)
- User action(Search, Select, history, Drop Courses, Log out)
- Schedule display
- Course Recommendation system
- Concurrency Control
- Course review system

## 2.3 System Components

### 1. User

Individual users of the software are required to pass through authentication to use the features of the software. The users will interact with the software with a web browser rendered by the web application server.

### 2. Front-end

Render UI determining each part of the software application and how it will look to users. To implement a front-end component, ReactJS is chosen for the application as components can be modularized and reused, which is highly productive, especially for large systems. As it also

allows to implement complex UI and lots of open-source libraries compare to others, allowing developers to build efficient and user-friendly application

### 3. Back-end

Provide data when there is a request from the front-end. Process the data in the context of the application, communicate with the database organising the information. In the Back-end part, we would use NodeJS and ExpressJS to set up the server side to respond with a client side. It also queries and receives data from the database and stores user/admin input to the database.

### 4. Database

Collection of structured data such as user and course information, accessed by database management systems. We would use MySQL to store the data, with predefined schema necessary for our components and features since all the functions are fixed and no such additional feature would be added without updates.