**Simplifying Online Course Selection**

**A Redesigned Course Management System for UCSI**



Arash Karimpourghannadi

[1002057456@ucsiuniversity.edu.my](mailto:1002057456@ucsiuniversity.edu.my)

# Background

This is the year 2023 and in this digital age, colleges and universities are expected to provide an efficient way for students to do their course selection completely online. The current Course Management System solution at UCSI University is famously known for its limitations and the overall bad experience that students tend to have when using it. To address these issues, a complete redesign of the system would be necessary to enhance the overall user experience.

Unfortunately, there are quite a lot of challenges that the students and even the faculty at UCSI University face when using the current online course selection system. The main complaint from the students is the lack of an intuitive and user-friendly interface. The current Course Management System's web page uses IIS (Internet Information Services) version 2 which appears outdated simply because it is an outdated option. This makes it very difficult for the students to navigate the available courses trying to find the ones suitable for them. The lack of any filtering options makes it even worse, as the students are forced to go through the list of all available courses trying to find the appropriate courses.

The current system also lacks detailed descriptions of the courses and instead only represents the names of the courses and their timing. This brings extra issues for the students, as without access to the basic information about each course and its prerequisites, it becomes very hard to make smart and calculated choices regarding their courses which could lead to the students regretting their choices later, as they learn this information about their courses. A much more informative system is needed to ensure that the students can make well-informed decisions regarding their course selections.

Another issue with the current system is the lack of online course recommendations. The course recommendations at the UCSI University are only given by the faculty in a face-to-face manner which has its own advantages like being more reliable or personal, but this would make it much harder for the students to get the recommendations they were looking for. This brings extra issues for the students.

In addition, canceling and retaking courses using the current system is very inefficient and extremely confusing. This leaves the students confused when they try to modify their course schedules, which leads them to seek offline help. That is, they must physically go to UCSI University and talk to the faculty for their issues to be resolved. Fixing this issue by creating a new more user-friendly interface for managing such challenges during the course selection could improve the overall experience of the students by magnitudes.

The current Course Management System at UCSI University uses the outdated IIS (Internet Information Services) version 2 from the Microsoft Corporation. This introduces its own set of limitations, such as compatibility issues with newer software and potential security vulnerabilities because of using outdated software which is not being regularly updated. This highlights the need for a newer and more updated system to ensure a secure online course selection process.

To summarize, the weaknesses of the current Course Management System are as follows:

1. Lack of a clear and user-friendly interface.
2. Insufficient information about the available courses.
3. Lack of online course recommendations.
4. Inefficient and confusing course modification process.

Because of all these limitations, the faculty at UCSI University have come to understand that a new Course Management System is clearly needed. By completely redesigning the process, UCSI University can simplify the whole process a lot and improve the user experience. UCSI University hopes to help the students make more well-informed decisions regarding their course selection by providing them with a simpler, more informative, and user-friendly Course Management System.

# Requirement Analysis & Design

### Understanding Business Needs and Processing Requirements

To understand the needs and requirements of the new Course Management System at UCSI University, I asked some of the students regarding their experiences with the old Course Management System and their perspective on the matter. This was to help me understand the problems with the current system better and to find some new ideas about how to improve the system.

### System Requirements

The system requirements for redesigning the Course Management System at UCSI University can be categorized into two separate groups, functional and non-functional. The functional requirements are simply the features that this new Course Management System should include. In simple words, what should this new system have:

User registration and authentication: This allows the students and faculty members to create official accounts to be able to access the system. This is an important step as each person must authenticate himself or herself before he or she can access the system or change anything in it.

Course browsing and filtering: This is so the students would be able to search and filter through the courses that are offered using subject, timing, and prerequisites. Knowing the prerequisites is important as the students should know if there are any courses that they need to take first before being able to choose their desired course. For example, if a student wants to take course B but course A is a requirement for course B, then he or she should see course A but not B, as course B is technically not available to him.

Detailed information about each course: We need to provide very detailed descriptions of each course, such as basic course outlines, course prerequisites and hours. This will help the students to make better and more well-informed decisions.

Online course recommendations: While this one is not mandatory; it is a very useful feature that most students are wishing for. To accomplish this, we need an algorithm that can suggest courses based on each student’s history, career goals and personal preferences.

Course modification and scheduling: Facilitate easy course modification, including adding, dropping, and swapping courses, with a user-friendly interface and clear instructions.

The non-functional requirements are simple anything regarding the performance, security, and usability of the system, including:

Performance: This is to make sure that the new Course Management System can handle many users who are using the system at the same time. This is because all the students in the university will access the time at the same time which can easily overload the system. And thus, we need to make sure that the system can handle such traffic in the first place.

Security: We need to take the security measures needed to protect the user data and to prevent unauthorized access to the system.

Usability: We need to design a user-friendly interface which looks good, instead of looking outdated. It needs to be easy to navigate, and accessible through both personal computers and phones.

### System Design Models

Multiple design models were developed to show the system’s functionality and its requirements:

Use Case Diagram: The following is a visual representation of the system's interactions with the different actors. That is, the students, the faculty members, and the administrators. It shows the main use cases and the relationships between them, highlighting the main functionalities provided by the system. The following diagram visually represents the interactions between the actors (Student, Faculty, and Administrator) and the various use cases of the system. It demonstrates the main functionalities of the system, such as registering a new account, logging in and authentication, browsing the list of courses, viewing course details, selecting desired courses, modifying course selections, and viewing the final schedule.

A screenshot of a computer screen

Description automatically generated

Since the Use Case Diagram is a bit complicated, I also drew the diagram separately for each actor. There’s no difference between the following three diagrams and the one above, except that the following diagrams are specific to each actor in the system. Faculty, Student and Administrator.

A screenshot of a computer

Description automatically generated

As shown in the diagrams, the only difference between the User and Faculty is browsing the list of courses. This is because the list of courses is based on the individual needs of each student, and not a long list that shows every course available at UCSI University.

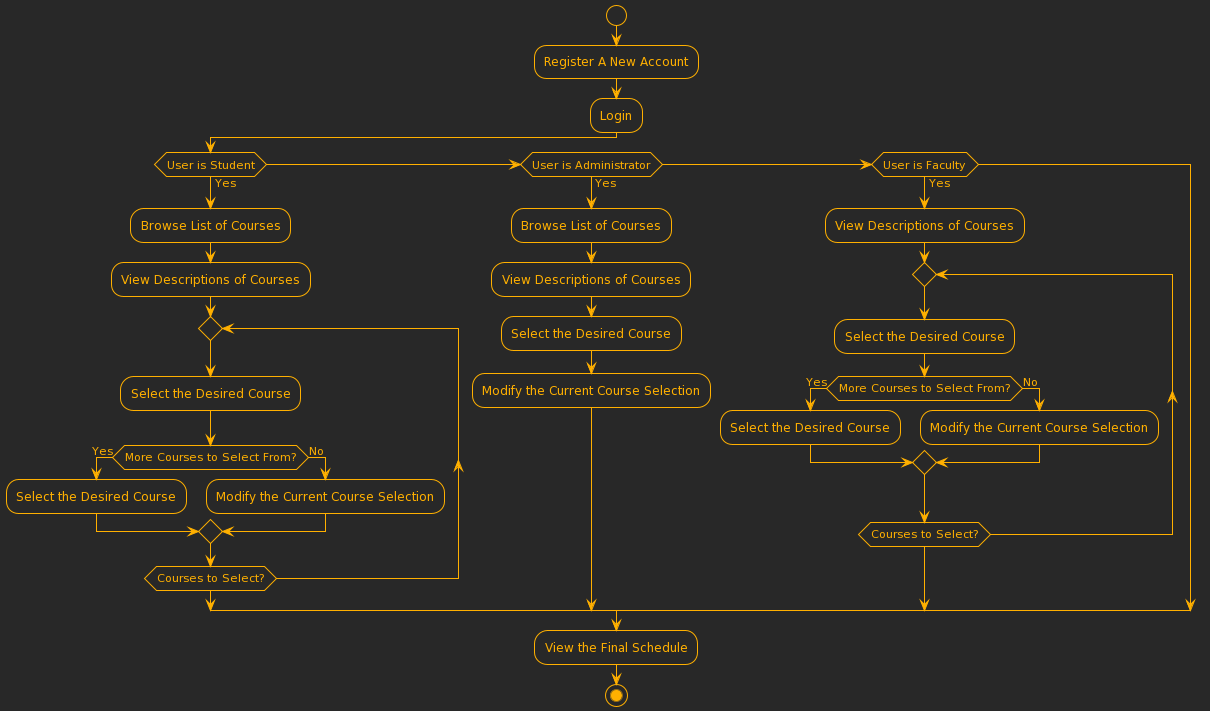
A screenshot of a computer screen

Description automatically generated

A screenshot of a computer screen

Description automatically generated

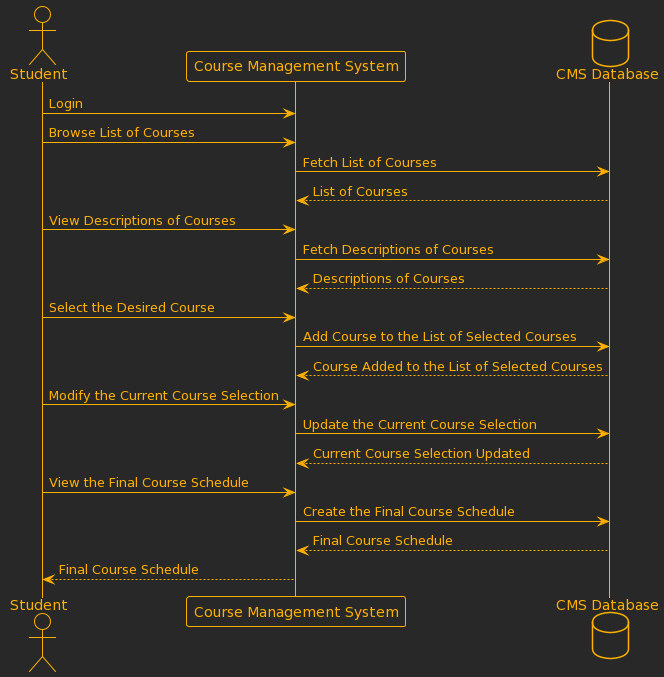
Activity Diagram: The following diagram is a graphical representation of the workflow and the activities that are part of the course selection process. It shows actions in a step-by-step fashion including any decision points, providing an overview of the steps taken by different actors to accomplish their goals. It shows the exact sequence of steps, such as browsing the list of courses, viewing course descriptions, selecting desired courses, and modifying the current course selection. Finally, it concludes with viewing the final schedule.



Sequence Diagram: The following is a dynamic model that represents the interactions and the message exchanges between the system objects. It shows the order of events during three separate scenarios. This Sequence Diagram show the interactions with the Course Management System’s database, which includes such steps as retrieving descriptions of courses, adding new desired courses to the list of selected courses, modifying, and updating the current course selection, and generating the final schedule from the list of the selected courses. As Sequence Diagrams deal with very specific scenarios, I had to come up with three separate scenarios.

1. Student Course Selection
2. Faculty Course Assignment
3. Administrator Course Management

The first Sequence Diagram showcases the “Student Course Selection” scenario, where the actor is the student and the interactions involve the student logging in, browsing through the list of available courses, selecting his or her desired course, and finally modifying their course selection.



In the Sequence Diagram above, the "Student" actor interacts with the Course Management System (CMS) to select their desired courses. Here are the steps in this scenario:

1. The “Student” logs in to the Course Mangemange System.
2. The “Student” browses through the list of available courses.
3. The Course Mangemange System fetches the list of available courses.
4. The “Student” views the descriptions of a specific course.
5. The Course Mangemange System fetches the descriptions of the specific course.
6. The “Student” selects a desired course.
7. The Course Mangemange System adds the course to the list of courses selected by the “Student”.
8. The “Student” modifies their course selection.
9. The Course Mangemange System updates the list of selected courses.
10. The “Student” views their final course schedule.
11. The Course Mangemange System generates the final course schedule.
12. The Course Mangemange System provides the final course schedule to the “Student”.

The second Sequence Diagram showcases the “Faculty Course Assignment” scenario, where the actor is the “Faculty” and the interactions involve the faculty members logging in, viewing the descriptions of the available courses, and finally modifying their course selection.

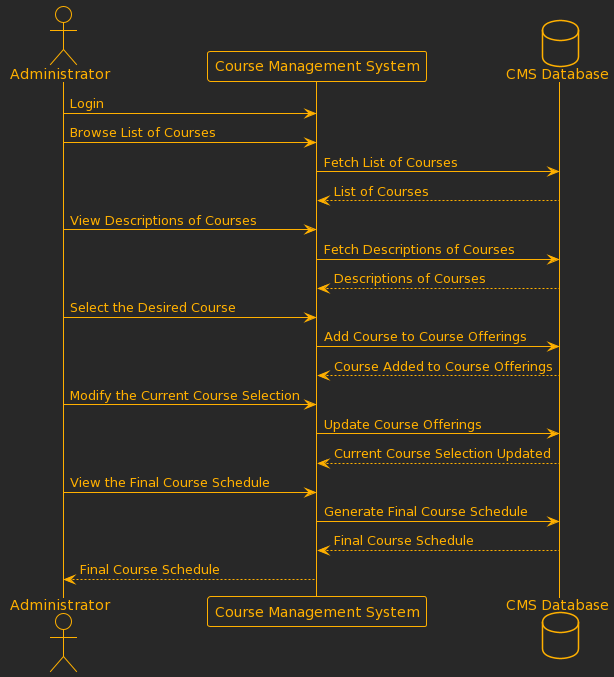
A diagram of course management system

Description automatically generated

In the Sequence Diagram above, the "Faculty" actor interacts with the Course Management System (CMS) for course assignment. Here are the steps in this scenario:

1. The “Faculty” member logs in to the CMS.
2. The “Faculty” member views the descriptions of the courses.
3. The CMS fetches the descriptions of the courses from the CMS Database.
4. The CMS Database provides the descriptions of the courses to the CMS.
5. The “Faculty” member selects the desired course for assignment.
6. The CMS adds the course to the “Faculty” member's assignment in the CMS Database.
7. The CMS Database confirms that the course has been added to “Faculty” member's assignment.
8. The faculty member modifies their current course selection or assignment if needed.
9. The CMS updates the “Faculty” member's course assignment in the CMS Database.
10. The CMS Database confirms that “Faculty” member's current course selection has been updated.
11. The “Faculty” member views the final course schedule.
12. The CMS generates the final course schedule by accessing the CMS Database.
13. The CMS Database provides the final course schedule to the CMS.
14. The CMS provides the final course schedule to the “Faculty” member.

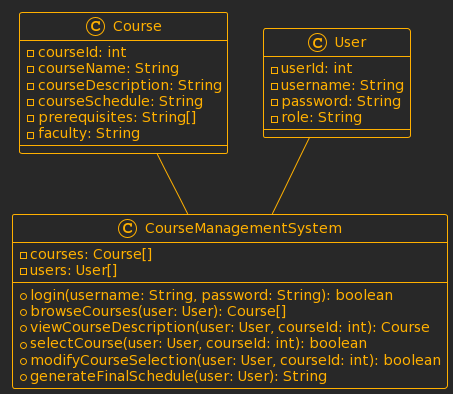
And finally, the third Sequence Diagram showcases the “Administrator Course Management” scenario, where the actor is the “Administrator” and the interactions involve the administrators logging in, browsing through the list of courses, viewing the descriptions of the available courses, selecting courses, and finally modifying their course selection.



In the Sequence Diagram above, the "Administrator" interacts with the Course Management System (CMS) for managing courses. Here are the steps in this scenario:

1. The "Administrator" logs in to the CMS.
2. The "Administrator" browses the list of courses available.
3. The CMS fetches the list of courses from the CMS Database.
4. The CMS Database provides the list of courses to the CMS.
5. The "Administrator" views the descriptions of the courses.
6. The CMS fetches the descriptions of the courses from the CMS Database.
7. The CMS Database provides the descriptions of the courses to the CMS.
8. The "Administrator" selects the desired course to add to the course offerings.
9. The CMS adds the course to the course offerings in the CMS Database.
10. The CMS Database confirms that the course has been added to the course offerings.
11. The "Administrator" modifies the current course selection or offerings if needed.
12. The CMS updates the course offerings in the CMS Database.
13. The CMS Database confirms that the current course selection has been updated.
14. The "Administrator" views the final course schedule.
15. The CMS generates the final course schedule by accessing the CMS Database.
16. The CMS Database provides the final course schedule to the CMS.
17. The CMS provides the final course schedule to the "Administrator".

Design Class Diagram: The following is a structural model that identifies the main classes in the Course Management System and the relationships that they have with each other. It showcases the attributes, methods (functions), and the associations between the classes.



The Design Class Diagram above shows the structure and the classes in the Course Management System and their relationships with each other. The are three classes available. They are “Course”, “User”, and “CourseManagementSystem”.

The “Course” class represents a specific course and includes attributes such as courseId, courseName, courseDescription, courseSchedule, prerequisites, and faculty. These attributes hold information about the courses. The “User” class represents users of the system, including Faculty, Student, and Administrator. It includes attributes such as userId, username, password, and role. The role attribute differentiates between the roles of users within the system. Role is what tells the system if a user is a “Student”, a “Faculty” member, or an “Administrator”. While generally char is used for such an attribute, I used a String to keep things simple for future developers. Keeping the code simple to understand can help future developers to make changes to the system easier and faster which will result in the Content Management System to be updated more regularly. The “CourseManagementSystem” class is the class that includes all the main logic and operations. It contains the collections courses and users to store the courses and users of the system. The class provides methods such as login, browseCourses, viewCourseDescription, selectCourse, modifyCourseSelection, and generateFinalSchedule to make it possible for the users to interact with the system. The Course and User classes are associated with the CourseManagementSystem class because the courses and the users are managed within the system.

### Storyboard of User Interfaces

At this stage, we need to focus on the user interface design for the system. The goal here is to make a very simple and clear interface that doesn’t have the issues that come with the current system, to help students, faculty, and administrators to browse the courses and select their desired courses with ease.

In the process of redesigning the user interface, many things were considered. The chosen color scheme is very minimalist, with green indicating selected courses and blue or red used for unselected courses depending on their availability and prerequisites. The interface adopts a minimal theme to make sure that everything is shown in a clear way. The design also supports accessibility to make sure all the users can access the Course Management System and use it without any external help. The new Course Management System has multiple pages, each with a different purpose.

Login Page: The login page should be quite plain and straightforward for all users. This can be achieved by designing this page in a clean and minimal way. After the users authenticated themselves successfully, users will be automatically redirected to the course browsing page.

Course Browsing and Selection Page: Users can browse and select courses from a card view or calendar view. In the card view, courses are presented as individual cards, displaying the name of the course on top followed by the code for the course, name of its lecturer, the timing of the classes, and a brief description about the course. Also, if there are other courses that are considered a prerequisite for the course, then the courses will be listed at the end of each card as highlighted links.

A screenshot of a computer

Description automatically generated

Selected courses are in green, while unselected courses are displayed in blue or red based on their availability and prerequisites. Keeping different courses in different colors makes it much easier for anyone to distinguish between different types of courses just by glancing at the page.

A green screen with white text

Description automatically generated

Available courses will be in blue unless they can’t be selected.

A blue screen with white text

Description automatically generated

That can happen for two reasons. First is when the timing of classes clashes with other courses which are already selected, and second, if the course has prerequisites that need to be taken first. In both cases the reason for the card not being selectable will be a highlighted link to direct the user to appropriate courses.

A screenshot of a phone

Description automatically generated

The calendar view presents courses in a visual format, showing their timing and availability. Users can select courses directly from the calendar view, with the same color scheme applied to indicate availability and selection status in the card view. This view is especially helpful because the timing of different courses can easily clash with each other and seeing the list of courses based on the timing of their classes can be extremely helpful.

Course Modification Page: This page will be very similar to course browsing. The big difference here is the sorting of the available courses. The card list should start with the cards that the user wanted to select, followed by the courses he or she already selected, and finally any other available course. This is to make it simpler for the user to find a solution to their issues. Users can deselect courses as easily as they selected them. By clicking on them, they get removed from the schedule. If a course cannot be selected due to timing conflicts with already chosen courses, it is highlighted in red to prevent accidental selection. Clear instructions and notifications guide users through the modification process.

Final Course Schedule Page: The final course schedule page provides users with an overview of their selected courses. Courses are displayed in a structured format, including the timetable. Different courses will be in different random colors to make them easily distinguishable.

Accessibility: Accessibility was a priority during the design process. The interfaces were created with intuitive navigation and a minimal theme, to ensure all the users can easily understand and interact with the Course Management System. Accessibility features, such as appropriate color contrast, alternative text for the images, and keyboard navigation support, were implemented to make sure that all users, including those with disabilities, can effectively engage with the system.

Project Plan

Now it’s time to outline the project plan for the development of the Course Management System. This plan will guide the developers through the different phases that are required to create the Course Management System from beginning to finish.

Project Goals: The main goal of this new Course Management System is to create a better user experience for the users by simplifying the course selection process. This redesigned Course Management System’s goal is to address the existing issues regarding the current Course Management System and to create a simple yet clear platform for the students, the faculty members, and the administrators.

Project Approach: To make sure that the development process will be handled professionally, we will use the Software Development Life Cycle (SDLC) methodology. SDLC methodology is the main approach which is used by programmers during their development process. It allows for requirements gathering, detailed system design, iterative development and testing, and user feedback through each phase. By following the SDLC guidelines, we can effectively manage the project.

Project Phases: The Course Management System project has the following phases:

1. Requirements Gathering: In this phase, we will need to talk to the real users of the old Course Management System to identify the needs and preferences of different students, faculty members, and administrators. This will give us ideas to shape the new system's features and functionalities.
2. System Design: Based on the requirements we gathered in the previous phase; the system architecture will be designed. This phase will focus on creating an intuitive user interface, course management workflows, and of course integrating necessary security measures.
3. Development: This phase is where the actual coding and development of the Course Management System will take place. This is where following coding standards matter as it ensures that the system is scalable, and maintainable.
4. Testing: Careful testing needs to be done to make sure that the system is working the way it was intended and to ensure that it is reliable code.
5. Deployment: After the testing is done and the Course Management System has been approved by the UCSI University, it will be finally deployed. This includes such things as server setup, and configuration. In simple words, it will be uploaded to UCSI servers and will finally be available for the users to test it.
6. Training and Documentation: Classes and training sessions are needed at this point to teach the students, and especially the faculty members, and administrators, how to use the new system the way it was intended to be used. A complete and detailed documentation will be needed to provide the users with a reference guide.
7. System Maintenance and Support: After it’s all done, there will still be a need for ongoing system maintenance and support to make sure that the new Course Management System will remain up to date, and secure. Plus, that’s the only way to make sure that the Course Management System will meet evolving user needs.

Project Timeline: The project will start on August 1st, 2023, and has an estimated duration of six months. The following Gantt chart illustrates the project timeline, including its major milestones:

A screenshot of a computer

Description automatically generated

This Gantt chart above provides a visual representation of the project schedule, the activities involved in it, and its milestones. It highlights the start and end dates of each phase and the path that should be followed by the developers for the goals of the system to be accomplished. This path shows the sequence of activities that need to be followed.

A screenshot of a document

Description automatically generated

The project includes two significant milestones, the “project kickoff” and “development completion”. The project kickoff marks the official start of the project, where everyone involved will gather and start taking the first steps toward creating the new Course Management System. Development completion marks the end of the development phase. The next phase after it would be the testing and deployment.

This project plan provides a detailed overview of the development process of the Course Management System. It serves as a visual tool to track progress and ensure completion of project milestones in a timely fashion. This project plan sets the stage for the subsequent phases of the project, ensuring effective coordination between the different developers involved.

Looking ahead, this newly redesigned Course Management System at the UCSI University is just the start for different types of future improvements. With a strong foundation, the system can be improved upon easily and efficiently. This ongoing development ensures that UCSI University will provide the students and faculty with the latest technology for better learning and teaching experiences.