QAIWAN INTERNATIONAL UNIVERSITY - ONLINE COURSE REGISTRATION

CARWAN JAAF

QAIWAN INTERNATIONAL UNIVERSITY

PSZ 19:16 (Pind. 1/13)

UNIVERSITI TEKNOLOGI MALAYSIA

DECLARATION OF THESIS / UNDERGRADUATE PROJECT REPORT AND COPYRIGHT

Author's full name : Karwan Jaaf

Date of Birth : 22/6/2001

Title : QAIWAN INTERNATIONAL UNIVERSITY - ONLINE COURSE REGISTRATION

Academic Session :

I declare that this thesis is classified as:

☐ **CONFIDENTIAL** (Contains confidential information under the Official Secret Act 1972) *

☐ **RESTRICTED** (Contains restricted information as specified by the organization where research was done) *

☑ **OPEN ACCESS** I agree that my thesis to be published as online open access (full text)

I acknowledged that Universiti Teknologi Malaysia reserves the right as follows:

The thesis is the property of Universiti Teknologi Malaysia

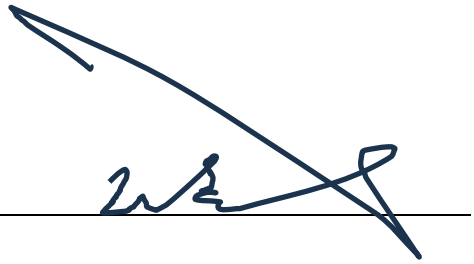
The Library of Universiti Teknologi Malaysia has the right to make copies for the purpose of research only.

The Library has the right to make copies of the thesis for academic exchange.

	Certified by:
SIGNATURE OF STUDENT	**SIGNATURE OF SUPERVISOR**
QU192SCSJ01	Mr. Karzan
MATRIX NUMBER	**NAME OF SUPERVISOR**
Date: 11 Jan 2023	Date: 11 Jan 2023

NOTES : If the thesis is CONFIDENTIAL or RESTRICTED, please attach with the letter from the organization with period and reasons for confidentiality or restriction

“I hereby declare that we have read this thesis and in my opinion this thesis is sufficient in term of scope and quality for the award of the degree of Bachelor of Computer Science (Software Engineering)”

Signature : 

Name of Supervisor : Mr. Karzan

Date : 20 MAY 2022

QAIWAN INTERNATIONAL UNIVERSITY - ONLINE COURSE REGISTRATION

CARWAN JAAF

A thesis submitted in fulfilment of the
requirements for the award of the degree of
Bachelor of Computer Science (Software Engineering)

School of Computing
Faculty of Engineering
Qaiwan International University

December 2022

DECLARATION

I declare that this thesis entitled *"Qaiwan International University - Online Course Registration"* is the result of my own research except as cited in the references. The thesis has not been accepted for any degree and is not concurrently submitted in

DEDICATION

This thesis is dedicated to my father, who taught me to never quit on the thing that you truly believe is possible to fix, with everything in life. It is also dedicated to my mother, who taught me that those who help you without asking them to help are the ones that should be close within your quarters.

ACKNOWLEDGEMENT

In preparing the thesis for the system, I took some thoughts off of students in Qaiwan International University to fully find the understanding of what they wish to see in the Online Course Registration system for QIU. I also took some advice from some lecturers. I would like to thank my supervisor (Mr. Karzan) for always believing in me and always explaining what I have missed or need to fix in my writings. Without Mr. Karzan I don't believe I could have gone this far, with all his support. I would also like to thank the students that have given me the feedback and support on this project.

ABSTRACT

The Qaiwan International University - Online Course Registration (QIU - OCR) project addresses the challenges faced by students at Qaiwan International University regarding the outdated course registration process. Currently, students are required to physically visit the campus at specific times for course registration, leading to various issues, such as missed opportunities due to timing constraints or inability to attend during vacations and pandemic situations. To address this problem, the project proposes the implementation of a modernized online course registration system. The chosen methodology for the project is the Waterfall Model, known for its incremental and systematic approach to software development. With clear requirements and objectives, this model allows for the creation of a comprehensive schedule and application design, ensuring predictability and effective project management. The proposed solution involves developing a user-friendly mobile application exclusively accessible to QIU students through their assigned QIU email accounts for enhanced security. Upon logging into the application, students will have the convenience of selecting courses available for their respective semesters and departments. They can also view their registered courses or drop courses if necessary. The QIU - OCR project aims to streamline the course registration process, offering students a more efficient and flexible way to manage their academic choices. By eliminating the need for physical presence on campus, this modern solution enhances accessibility and adaptability, ensuring a smoother registration experience for all Qaiwan International University students.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

QIU	-	Qaiwan International University
OCR	-	Online Course Registration
UI	-	User Interface

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Introduction

The project title is called Qaiwan International University - Online Course Registration (QIU - OCR). The reason this topic has been picked is because of the situations that many students have been a part of for almost for how long they have been in Qaiwan International University for. The main point of the project is for students to register for their courses online rather than on campus. The registering system in Qaiwan International University today isn't much modern like it should be, it still requires students to come to the campus to register for their courses. The overview is to allow students to access this website or application with their login address and password and pick which course they wish to pick for the ongoing semester.

1.2 Problem Background

The problem relating to this project is inside the Qaiwan International University itself, where some minor issues exist within the university system. However, one particular problem stands out among others, which is the registration system in QIU. To go into more detail, the problem arises from requiring QIU students to visit the campus at specific times and locations for course registration. This causes significant challenges, such as students not being able to arrive on time and missing their registration window or being unable to attend due to vacation commitments. Moreover, this problem was exacerbated during the pandemic when students were still mandated to come to the campus, risking their health and safety, to register for courses alongside other students. There is also the concern that future health crises may result in similar quarantine situations, impacting the next registration cycle.

1.3 Project Aim

The project aim of this project could be to understand the factors that influence the success of an online course registration system.”It acts as a focal point for your research and provides your readers with clarity as to what your study is all about.”(DiscoverPHDS, 2016) This could involve studying the user experience of students and instructors, the technical infrastructure and features of the system, and the policies and procedures surrounding course registration. The project could aim to identify best practices for designing and implementing an effective online course registration system, as well as to identify challenges and limitations of such systems and potential solutions to these problems.

1.4 Project Question

Is this project idea likely to be effective in the QIU system, or is it destined for failure?

1.5 Project Objectives

Some possible project objectives for a study on online course registration could include:

- To identify best practices for designing and implementing an effective online course registration system.
- To design and develop the factors that influence the success of an online course registration system, including the user experience of students, technical infrastructure, and challenges and limitations of online course registration systems and potential solutions to these problems.
- To test and evaluate the effectiveness of different online course registration systems or approaches to the project.

1.6 Project Scope

The scope of online course registration refers to the range of activities and processes involved in enrolling in and completing an online course. This includes identifying and selecting a course that meets the student's needs and interests, enrolling in the course by completing necessary forms, accessing course materials. Overall, the scope of online course registration bounds the various steps and activities involved in successfully completing an online course and earning credit.

1.7 Report Organization

Chapter 1 explains the reasoning of the whole project, and Chapter 2 is a literature review which shows the comparison between other similar systems. This report is organized in a way that the reader can easily read through the whole report and understand each step of what happened and how it has happened. It is organized in a formal way so that it is not all over that place to confuse the reader.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Numerous systems share the same idea that the QIU - OCR project aims to address. However, the purpose of this literature review is to gain deeper insights into the current issues faced by these systems, their unique contexts, and to propose future improvements based on real-world experiences. While many applications have their advantages and disadvantages, any shortcomings encountered serve as opportunities to fortify the QIU - OCR project by avoiding the pitfalls of its predecessors.

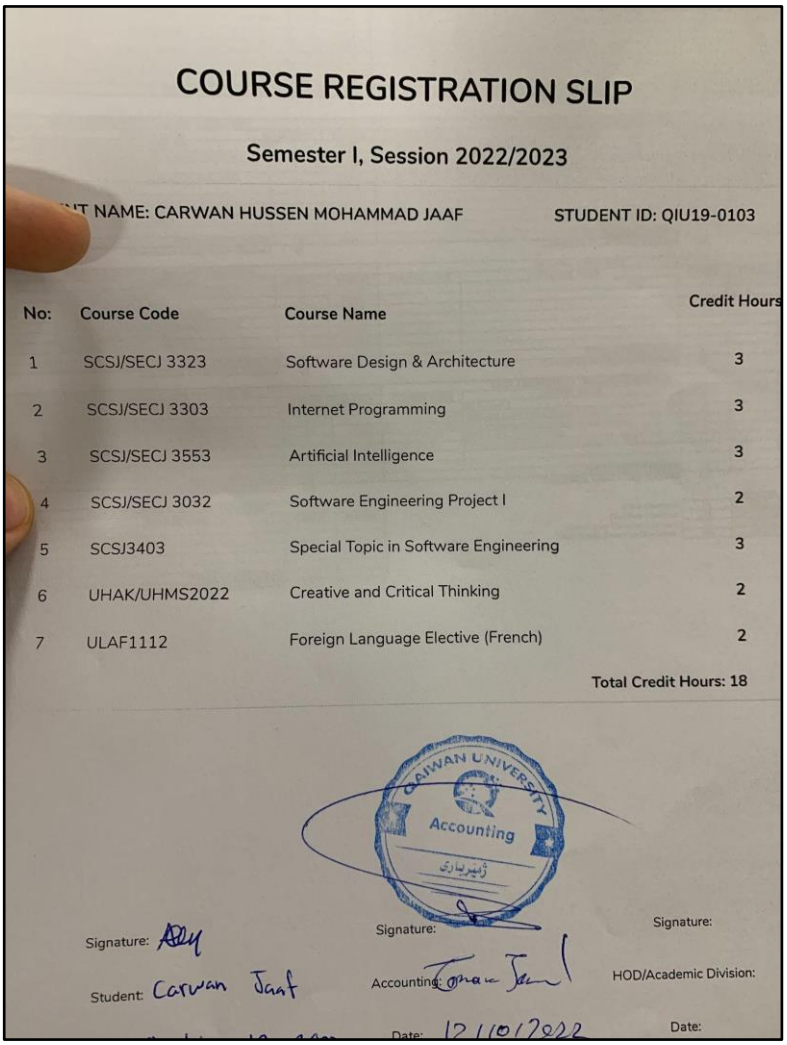
2.2 Problem Formulation

In recent years, the education systems have had it easy due to all of the web-based technologies that students nowadays use. To supplement their current teaching techniques, many educational facilities have now provided online courses, online homework, and many more. This provides an increase of how many students can be present in the course and also adults can join in too, to study. Everything has become remote based rather than written or printed information.

2.2.1 Project Domain

COVID-19 has brought significant disruptions to university closures and education across Europe. This didn’t only affect the educational systems but also the dramatic loss of human life, food systems, border & traveling limitations, and the qurantines that caused social interactions. Despite best efforts to set up a supportive remote learning experience, evidence is emerging to show that school closures have resulted in actual learning losses (Donnelly R., 2021).

As of today, Qaiwan International University still requires students to visit the campus just to choose their courses and sign their name, consenting that they accept the terms and services. The students in QIU have different majors from one another, where there's biomedical engineering, software engineering, human resources, and etc… While each major has different course differentiation between each other, the courses also have prerequisites for them which might be for a certain semester in the QIU students life, or a prerequisite of finishing another course for the second course to be available to the student. There are many prerequisites for courses for all the majors in QIU. Even during the pandemic, they still required students to come and register for their courses, while knowing how dangerous Covid-19 was. This method is still manual and has a variety of shortcomings sooner or later. Additionally, QIU still continues the registration process knowing that there might be another pandemic. The course registration slip that they give to the students is shown in Figure 2.1.



COURSE REGISTRATION SLIP

Semester I, Session 2022/2023

NT NAME: CARWAN HUSSEN MOHAMMAD JAAF STUDENT ID: QIU19-0103

No:	Course Code	Course Name	Credit Hours
1	SCSJ/SECJ 3323	Software Design & Architecture	3
2	SCSJ/SECJ 3303	Internet Programming	3
3	SCSJ/SECJ 3553	Artificial Intelligence	3
4	SCSJ/SECJ 3032	Software Engineering Project I	2
5	SCSJ3403	Special Topic in Software Engineering	3
6	UHAK/UHMS2022	Creative and Critical Thinking	2
7	ULAF1112	Foreign Language Elective (French)	2

Total Credit Hours: 18

QAIWAN UNIVERSITY Accounting

Signature: Signature: Signature:

Student: Carwan Jaaf Accounting: HOD/Academic Division:

Date: 12/10/2022 Date:

Figure 2.1 Registration Form for the Students

#	Student Name
1	Hevi Ahmed Safaa Ibrahim
2	Noor Nabeel Ibrahim
3	Ayoub Mohammed Abdulkarim
4	Paiwand Hadi H.Saeed
5	Safin Yasin Abubaker
6	Schanga Ayub Josef
7	Zhir Aras Mohammed
8	Ahmed Azad Jasim
9	Arivan Shamal M.zyad
10	Carwan Hussen Mohammad Jaaf
11	Dilan Barham Hussein Faraj
12	Mohammed Awat Mustafa
13	Mussa Sarkawt Abdul
14	Nahro Aso Othman
15	Ravyar Barzan Othman
16	Rebaz Awat Mustafa
17	Zhewar Omer Mohammed
18	Zhin Abdalqadir Sabir
19	Zhiwar Anwar Karim
20	Bezhan Jamal Mohammed
21	Danyar Omar Abdalla
22	Mohammed Saad Abdulwahhab
23	Las Sirwan M.Rashid

Figure 2.2 List of Students of Software Engineering Semester 5

Figure 2.2 shows the list of students in software engineering just in semester 5 that have to manually fill out the registration forms. There are many more students in QIU in different departments and different semester students.

This method takes time since students have to manually register for courses which is a very time consuming process, especially if students have to search through all of the available options and then manually add them to their registration form. This can take hours or even days depending on the number of classes they are trying to register for. This also might be difficult for those who don’t have the time to make it on time to register for courses and depending on how quickly a student registers for the classes, there may be limited availability for certain courses. This could lead to students not being able to get into the classes they need in order to complete their studies or have a more difficult time finding the perfect schedule that works with their lifestyle and other commitments. This can also lead to a student having to settle for classes that are not of their first choice.

2.3 Description of Related Studies

The Concordia University web application and the Ignatius Ajuru University of Education web application are two similar systems to the planned online course registration project. The below system identifies and assesses the proposed systems to aid in the discovery of features and limitations of the proposed system.

2.3.1 Concordia University Web Application (Case Study 1)

Concordia University's web application is an online platform that allows prospective students to apply for admission to the university. Through this platform, applicants can register for classes, pay tuition fees, check their admissions status and access other services related to their studies. The system also provides information about Concordia's academic programs and courses as well as providing a variety of resources such as student support services, library databases and research tools. It is a public research university located in Montreal, Quebec, Canada. The university offers a wide range of undergraduate and graduate programs across its three campuses; Sir George Williams Campus, Loyola Campus and Downtown Campus. Concordia University also provides an online portal for students to access their course information, register for classes and view grades. This web application is available on all platforms including Windows, Mac OS X and iOS devices. It allows users to search for courses by keyword or subject area as well as view descriptions of each course before registering.

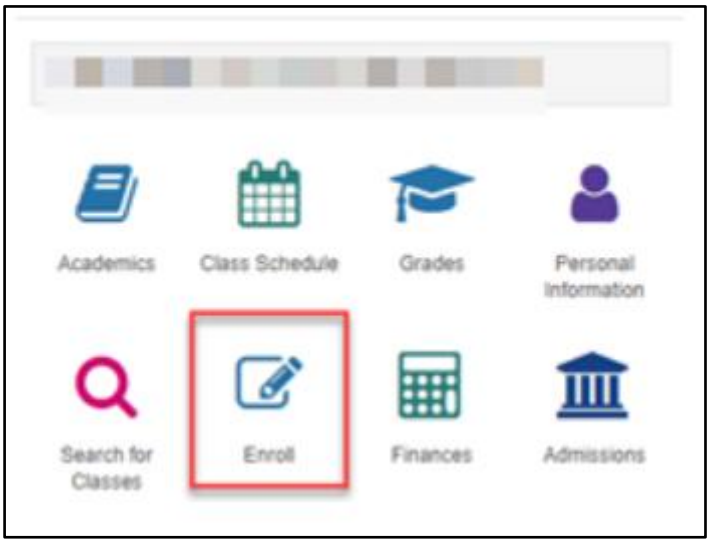


Figure 2.3 Concordia's Dashboard Page

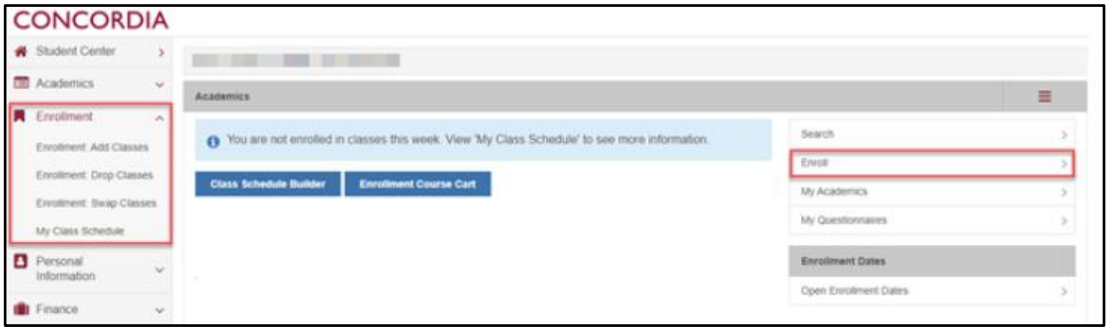


Figure 2.4 Concordia's Home Page

In figure 2.3, it is a simple UI for the students to easily navigate through the features of the system. The user can then select the kind of activity he or she wishes to engage in from the menu.

In figure 2.4, it shows many options for enrollments, either adding a new course, dropping a course, swapping a course, or either checking their class schedule. Each option has their different types of services that will help the student or user.

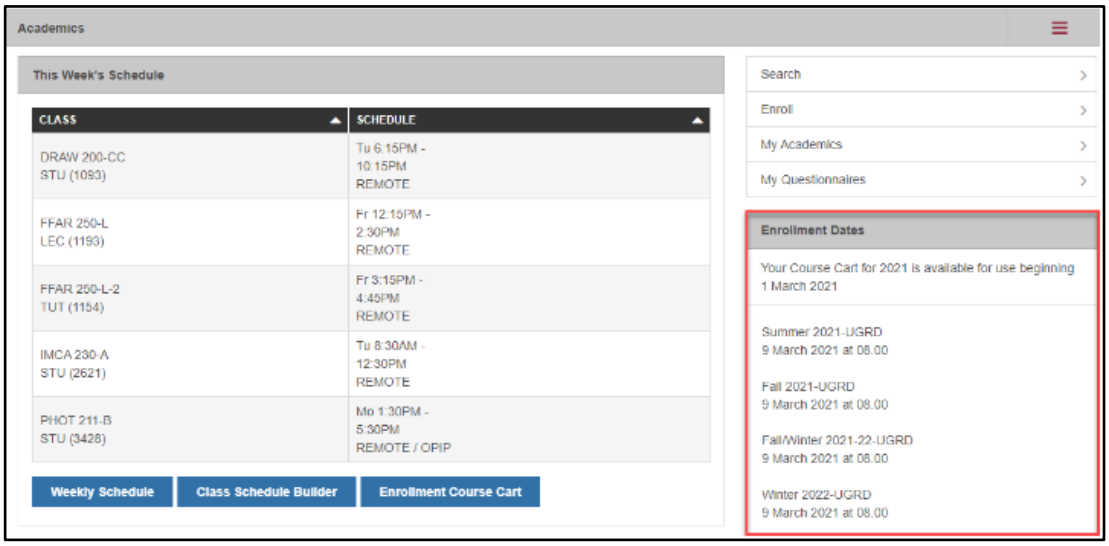


Figure 2.5 Enrollment Appointment Dates

Figure 2.5 shows the enrollment appointment dates where they can begin to register for their first day of that term. Their individual date will be shown on the enrollment dates box on the right side of the figure.

2.3.2 Ignatius Ajuru University of Education Web Application (Case Study 2)

The Ignatius Ajuru University of Education web application is an online platform designed to provide students, faculty and staff with access to resources related to the university. It includes a student portal that allows users to view course information, register for classes, manage their finances and access other services. The faculty portal provides professors with the ability to post syllabi and assignments, grade work and communicate with students. Additionally, the web application includes a library catalog where users can search for books or articles related to their studies or research interests. It also provides links to important university offices such as Admissions and Financial Aid. It offers users an easy-to-use interface for accessing academic resources such as course syllabi, exam schedules and results, library catalogues, announcements, campus events and more. The website also includes links to essential student services such as financial aid applications and scholarships. Additionally, the site features helpful tools for career planning such as job postings and internships.

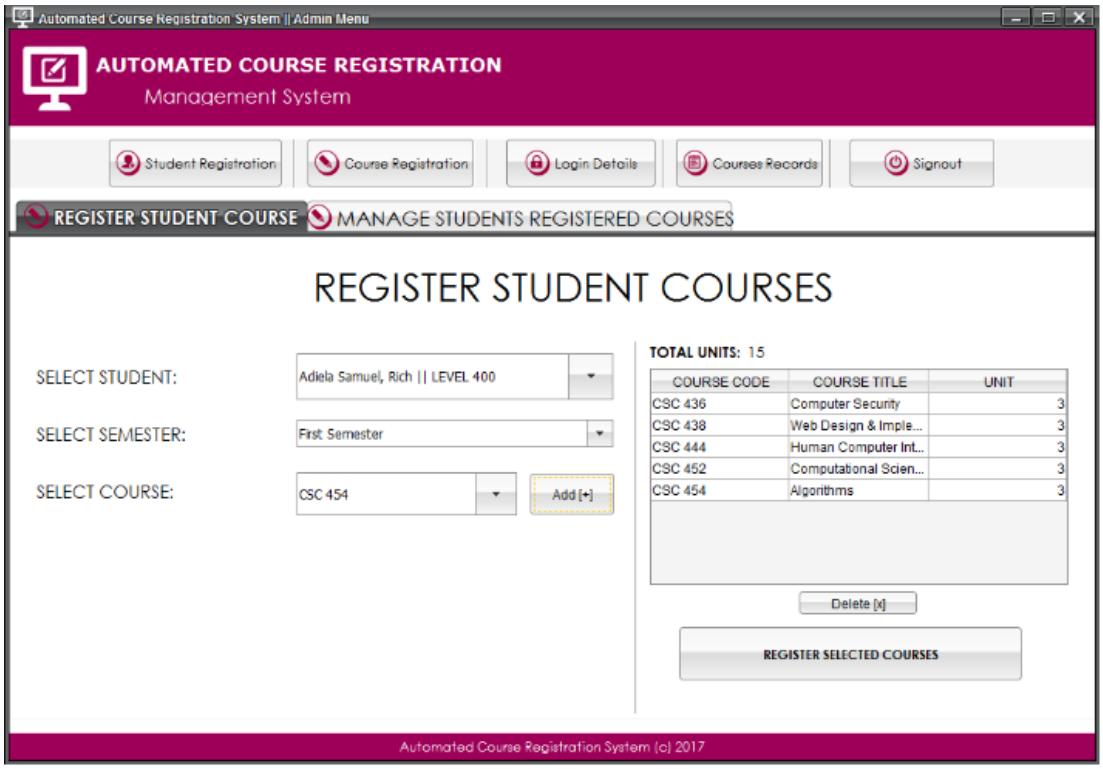


Figure 2.6 Register Student Courses

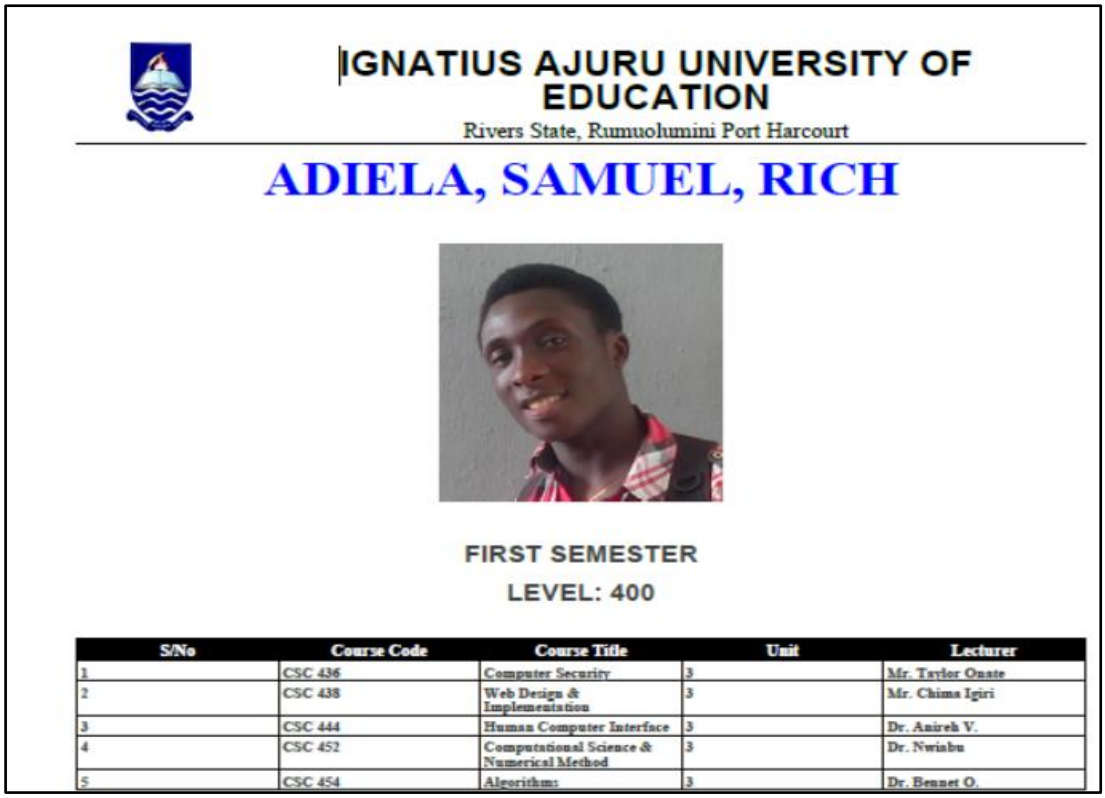


Figure 2.7 Student Information Page

In figure 2.6 it shows the students courses that they can register for following with how many credits each course has with them, and in figure 2.7 it shows the students information in a single page to see what the student has registered for and in which semester the student is in.

2.4 Comparison between Systems

Concordia University Web Application:

The Concordia University web application is an online portal for students, faculty, and staff to access information about the university. It provides a range of features including course registration, student accounts management, financial aid information, library resources and more. It also offers services such as career planning and development advice. The platform is easy to use with intuitive navigation tools that make it simple to find the information you need.

Ignatius Ajuru University of Education web application:

The Ignatius Ajuru University of Education web application is a comprehensive online platform that offers students access to their course information, registration process, payment system and other school-related activities. It also provides faculty members with the resources they need to plan classes and manage student records. Students can use this portal to check their grades, pay fees and apply for scholarships or financial aid. The website also allows users to search for courses, view upcoming events and even connect with other students on campus.

Properties	Concordia University	Ajuru University	Qaiwan International University
User	Students, Faculty, Staff	Students, Faculty	Students, Staff, Faculty
Register	Students need to be registered	Students can register themselves	Emails only available
Login	Students and Staff can login	Students can only login	Students, Staff, Admins can login
Description (Course)	Don’t have description for courses	Don’t have description for courses	Has description for courses

Internet Access	Needs	Needs	Needs
Portability	No	No	Yes

Table 2.1 Comparison Table of Similar Systems

The Concordia University web application and the Ignatius Ajuru University of Education web application are both used for student registration, course selection, payment of tuition fees and other school-related activities. Both applications provide access to a variety of services such as academic advice, library resources and career counseling.

The key difference between the two applications is in terms of user interface design. The Concordia University web application has a modern design that provides easy navigation with intuitive menus and links. It also offers an integrated search feature which allows users to quickly find information they need without having to scroll through pages or click on multiple tabs. In addition, it supports mobile devices so students can access the site from any device including tablets and smartphones.

2.5 Chapter Summary

This chapter's purpose is to show literature reviews overview of the current knowledge on a topic by summarizing what has been published in journal articles within the last few years. It also showed the similar online course registration systems that are close to the proposed system of mine named QIU - OCR. The similarities were compared and contrasted between each other and showed their strengths and weaknesses of their systems.

CHAPTER 3

SYSTEM DEVELOPMENT METHODOLOGY

3.1 Introduction

In this chapter, it will explain the methods and approaches that were chosen for building the QIU - OCR system. Methodology for system development involves a series of steps that are followed to develop an information system. It typically involves activities such as gathering requirements from stakeholders, analyzing user needs, designing a system architecture, testing the system design and then implementing it in stages. This process is often referred to as the systems development life cycle (SDLC). The purpose of this methodology is to ensure that all aspects of the system are considered during its development in order to create an effective solution for the user.

System development methodology is a framework used to structure, plan, and control the process of developing an information system. Commonly used methodologies include Waterfall Model, Agile Methodology, Rapid Application Development (RAD), Iterative Development Processes and Extreme Programming (XP). Each of these approaches has its own advantages and disadvantages depending on the project requirements.

3.2 Methodology Choice and Justification

The methodology that will be chosen for the QIU - OCR is going to have to be the Waterfall Model. The reasoning of why the Waterfall Model fits best for the QIU - OCR system is because the Waterfall Model is a popular software development methodology that follows an incremental approach to system development. It is best used when there are clear requirements and objectives, and the system can be broken down into distinct stages. “Requirements are completed early in the project, enabling the team to define the entire project scope, create a complete schedule, and design the

overall application."(R. Sherman, 2015). This model works well for projects where the customer needs a predictable delivery date, as each stage of the process must be completed before moving on to the next one. The Figure below will show how the project will be managed throughout the waterfall process, shown in Figure 3.1.

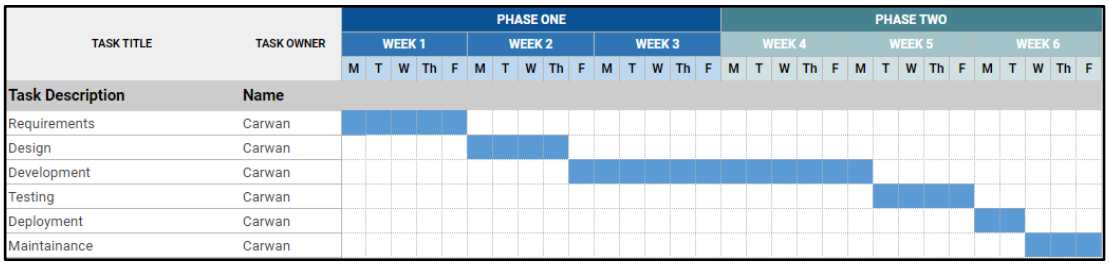


Figure 3.1 Gantt Chart Waterfall Process

It is often used in systems with well-defined requirements and specifications, as it provides an orderly progression of activities that can be easily monitored and controlled. This model works best when the requirements are clear and fixed, making it ideal for projects such as developing embedded systems or large applications where changes are not expected once the initial design is completed. The Waterfall Model allows teams to move through each phase of the project linearly, meaning they can identify potential problems early on before they become too costly to fix later in the development cycle.

Additionally, this model allows for more control over changes in scope or design during the project's lifecycle, since any changes must be approved by stakeholders at each stage. The Waterfall Model also helps ensure that developers create quality code with fewer bugs due to its emphasis on testing throughout every phase of development.

Advantages:

1. The Waterfall Model is easy to understand and use, making it a popular choice for software development projects.

2. It allows for more control over the project since each phase of its development must be completed before moving onto the next one, thus preventing costly errors or delays due to changes in requirements later on in the process.

3. By having clearly defined phases with specific deliverables, there is less chance of scope creep occurring during the course of development.

4. This model also enables team members to focus on their individual tasks without worrying about other aspects of the project that may still need attention at a later stage.

5. It is easy to manage due to its rigid structure, as progress can be measured in terms of completed phases.

Disadvantages:

1. Once a phase has been completed, it is difficult to go back and make changes without disrupting the entire project timeline and budget.

2. There is no room for iteration, so any issues that arise during development may not be able to be addressed until later phases of the project are complete.

3. The Waterfall Method does not allow for feedback from users or stakeholders until after development has already begun, which can lead to costly rework if their expectations have changed since then.

4. It can be difficult to accurately estimate how long each phase will take since there is no way of knowing what challenges may arise along the way.

5. It doesn’t allow for any sort of flexibility or adaptation to changes in technology or customer needs.

3.3 Waterfall Model Phases

This part will show the phases of the Waterfall Model step by step, also will be shown in Figure 3.1 below.

1. Requirements Gathering and Analysis: This is the first step in the waterfall model, where stakeholders identify project requirements and analyze them. This includes gathering information from customers, clients, users, etc., to understand their needs and expectations from the project.

2. Design: In this phase, a high-level design of how the software will work is created based on customer requirements gathered in the previous stage. The design phase also involves deciding on what technologies will be used for development purposes such as programming language, database type or user interface elements like buttons or forms.

3. Implementation/Coding: After designing what needs to be done in terms of functionality and features, developers start coding following best practices for writing clean code that can be easily maintained and updated.

4. Testing: After the code is written, it needs to be tested against user requirements and any bugs or errors need to be identified and fixed. This phase also includes integration testing which ensures that all components are working together as expected.

5. Deployment: Once the software has been tested successfully, it can then be deployed for use in a production environment such as a web server or cloud service.

This step involves setting up necessary infrastructure like databases and servers so that the software is available for end users to access in real time.

6. Maintenance: Even after deployment of the software, maintenance activities continue on an ongoing basis including bug fixes, updates, security patches etc., depending on customer needs or changes in the market. This phase is an ongoing process to ensure that the software works as expected and any issues are identified and fixed quickly.

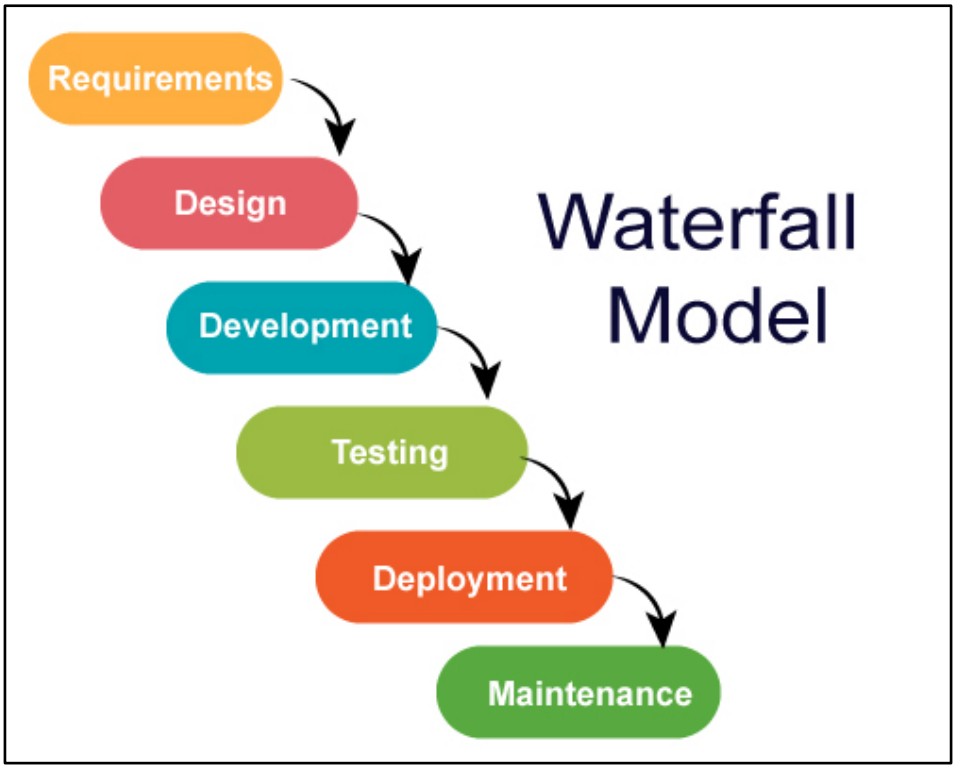


Figure 3.2 Waterfall Model Phases

3.4 Results and Analysis of the Survey

"A survey can be used to investigate the characteristics, behaviors, or opinions of a group of people."(K. Cherry, 2020). The survey was made for the students in

Qaiwan International University checking whether or not they feel comfortable with the current registration system or do they wish for a new change with the old registration system, and what would they prefer in the new course registration system. The survey was sent to a majority of students that were picked by random to even out the probability of the answers.

The following figures will show the results of all the questions that were asked in the form:

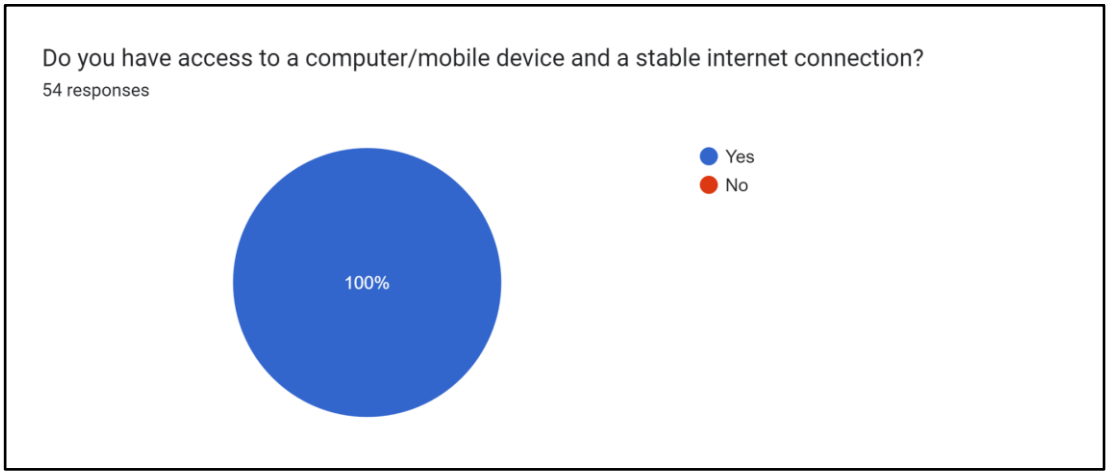


Figure 3.4 Question 2

This question was to figure out whether or not the students in Qaiwan International University have access to any mobile or computer device so they can access the Online Course Registration system in the future.

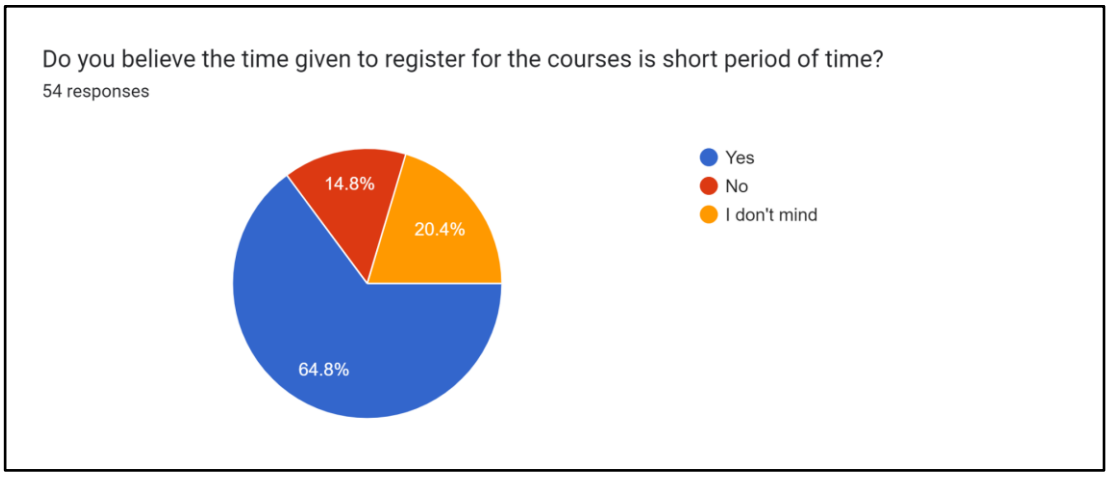


Figure 3.5 Question 3

This question was for those who have experienced the course registration system in QIU campus and seeing if they found it not having enough time to register for their courses when the notification was sent to them.

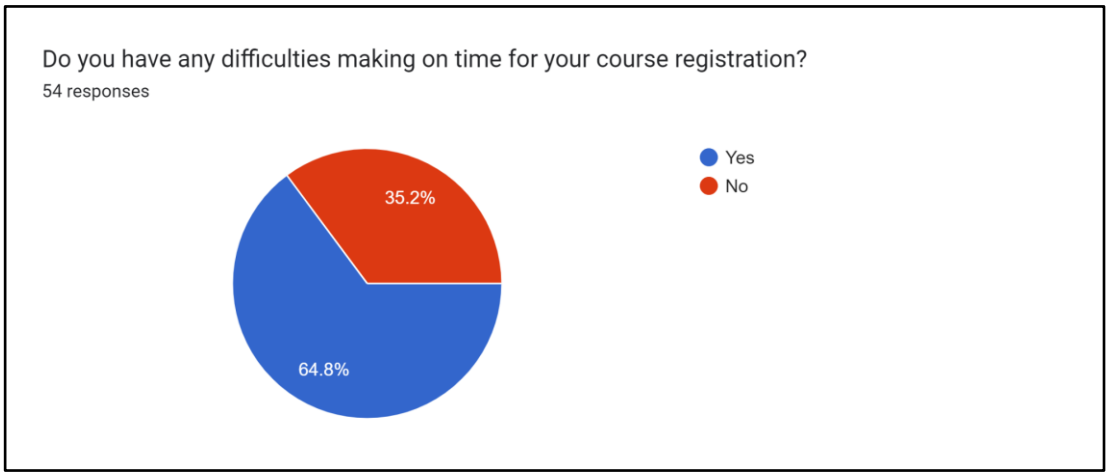


Figure 3.6 Question 4

This question was to determine if the students might have any difficulties during the registration day. Seeing if they aren't able to make it on time or are just out on a vacation of some sort.

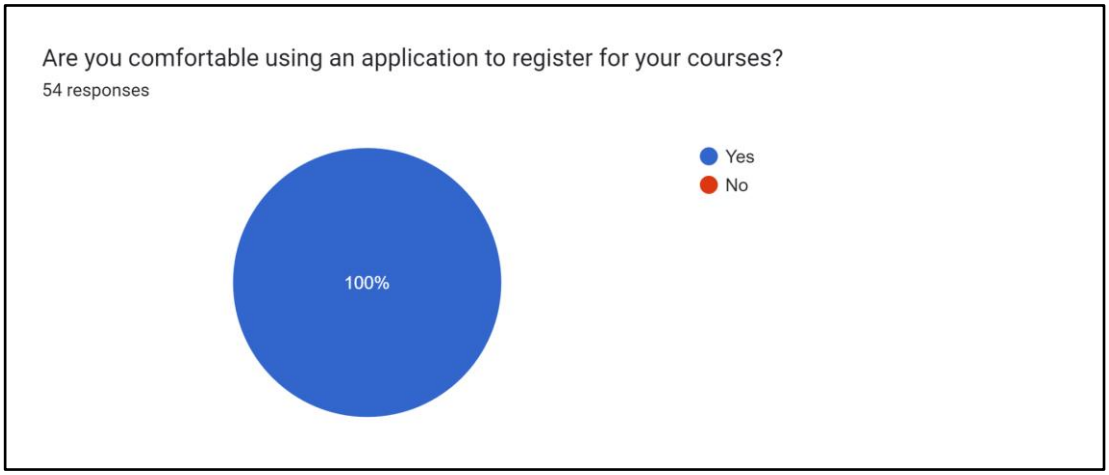


Figure 3.7 Question 5

This question is trying to figure out if the students feel comfortable using an application so they can register for their courses online.

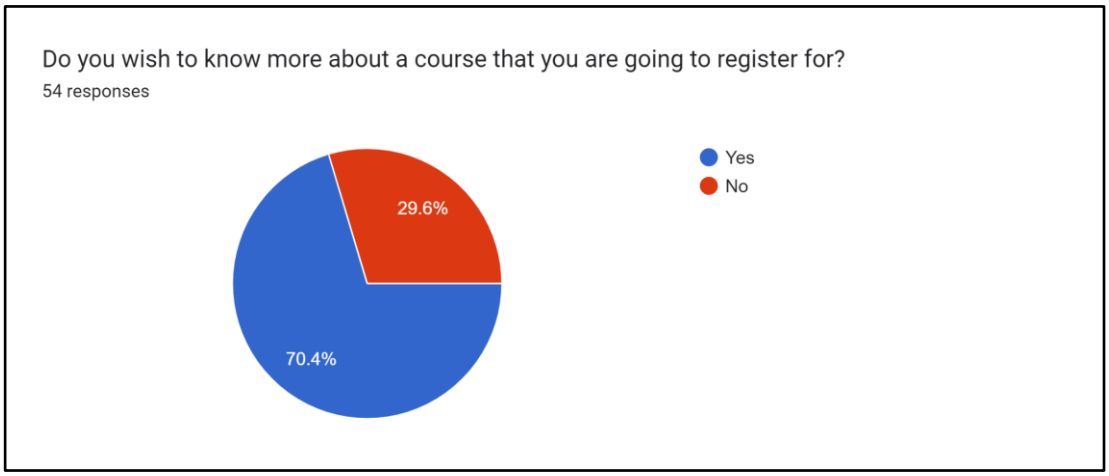


Figure 3.9 Question 7

This question is to find out what they would prefer on the application features, seeing if they want to know more information about a subject without finding out when they take the course.

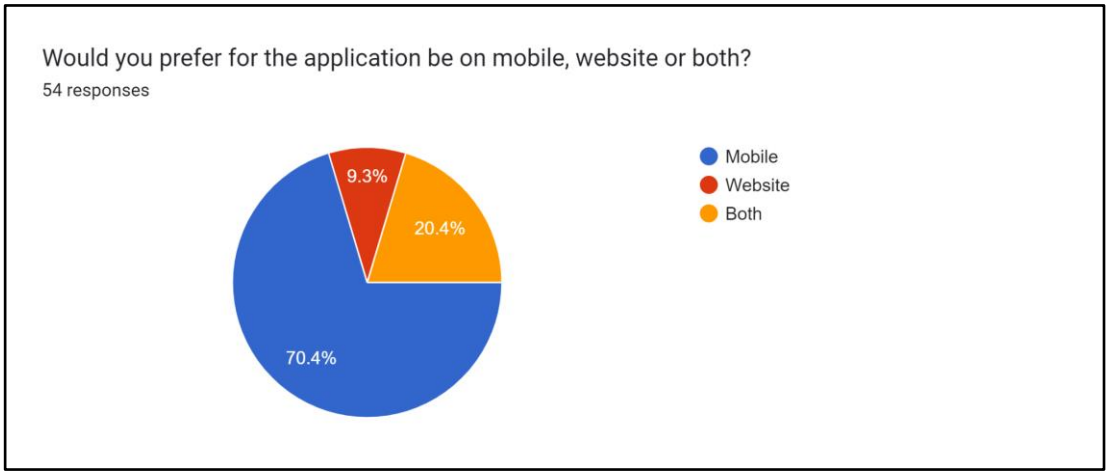


Figure 3.10 Question 8

This question was to see if they prefer the application to be on mobile, on a website or both, and the majority has picked for it to be on mobile.

3.5 Survey Result

The survey shows that this application will be effective for the Qaiwan International University system, students will have use for this application for their course registration process.

3.6 Chapter Summary

In a short note methodology is a set of principles, techniques and processes used to develop software solutions. The Waterfall Model is an example of a methodology which uses a sequential design process in which progress flows from one phase to the next in order. It follows the traditional engineering model whereby progress is seen as flowing downwards (like a waterfall) through the phases of Requirements Gathering, Design, Build/Test, Deployment & Maintenance. This approach works well for smaller projects where requirements are relatively easily understood and changes are not expected during development.

CHAPTER 4

REQUIREMENT ANALYSIS AND DESIGN

4.1 Introduction

This chapter explains the requirement analysis and design of the OCR - QIU application. Requirement analysis and design are activities that are typically carried out as part of the software development process, with the goal of understanding and specifying the requirements for an application. Requirement analysis involves identifying and documenting the needs and constraints of the application, including the functional and non-functional requirements. This might involve conducting user interviews, gathering requirements from stakeholders, and analyzing existing systems to understand how they meet the needs of users. Design, on the other hand, refers to the process of developing a plan or blueprint for the application. This might include creating prototypes or mockups of the user interface, as well as specifying the technical architecture of the system. The design process should take into account the requirements identified during the analysis phase, as well as any relevant design patterns or best practices.

The chapter will also contain the prototype of the application, the main functions of the application will consist of the login, home page, courses, description of the course, and the registration form. It will also include various diagrams such as use case diagrams, sequence diagrams, and activity diagrams.

4.2 Requirement Analysis

According to Simplilearn (2022), requirements analysis is the process of understanding and documenting the needs and expectations of a new product through regular communication with the stakeholders and end-users. This process helps to define the requirements for the product, resolve any conflicts or discrepancies, and ensure that the final product meets the needs of the users. It is an important step in the development process as it helps to ensure that the product is designed and built to meet the needs of the stakeholders and end-users.

4.2.1 Functional Requirements

Functional requirements are the specific capabilities, behaviors, and characteristics that a system must have in order to meet the needs of the user. They describe what the system is supposed to do and how it is supposed to behave. Functional requirements are an important part of the software development process, as they help to define the scope of the project and ensure that the final product meets the needs of the user. Overall, functional requirements help to ensure that the system is fit for purpose and meets the needs of the user. Use case diagrams are a type of graphical representation used to capture the functional requirements of a system.

4.2.1.1 Use Case Diagram

Use case diagrams are useful for showing the functional requirements of a system because they provide a high-level view of the system and its capabilities. They can help to identify the key functions that the system must be able to perform, and the actors that will use those functions. This information can be used to define the functional requirements of the system in more detail.

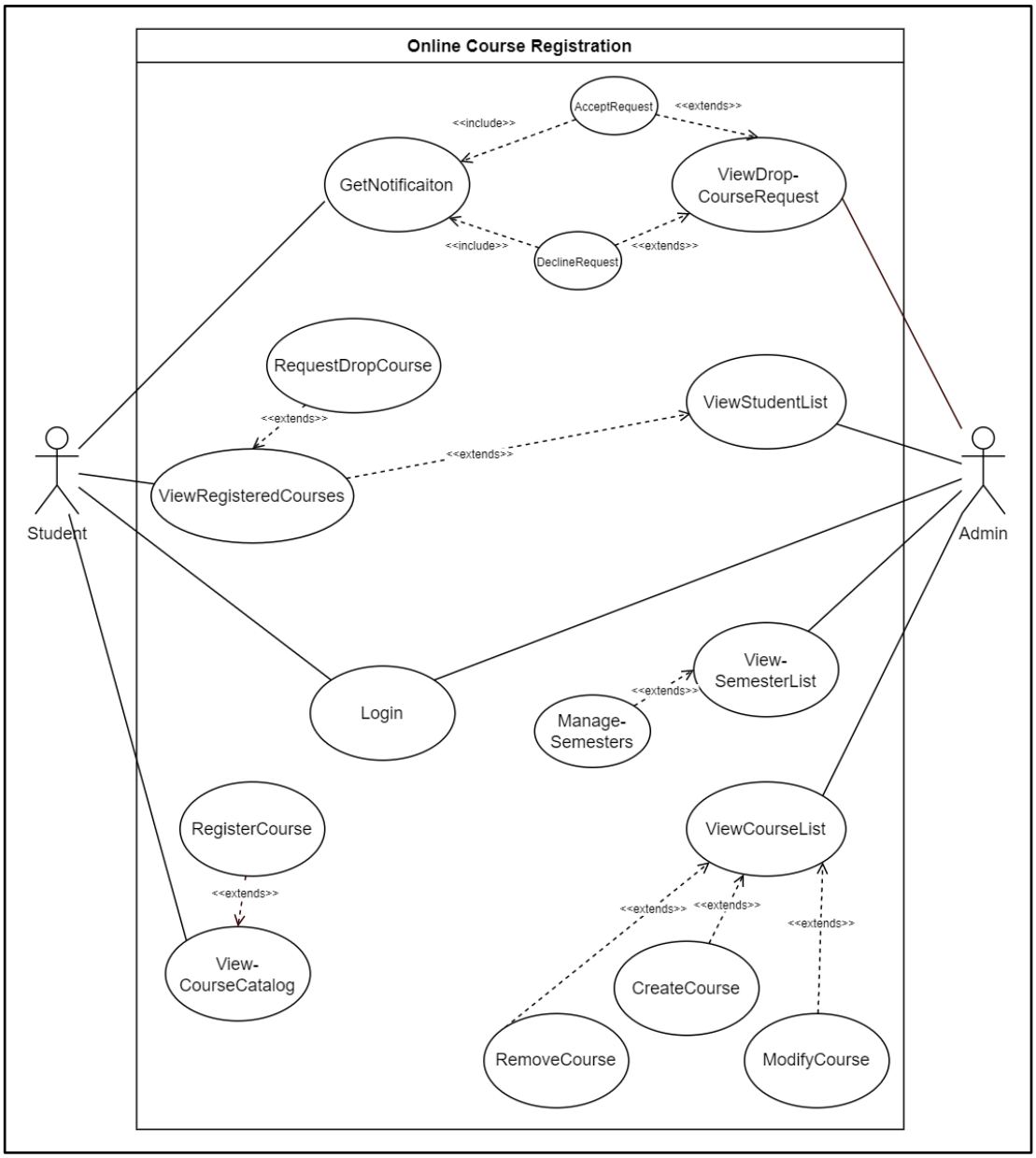


Figure 4.1 Use Case Diagram for OCR

4.2.1.2 Use Case Description

The system has a total of 9 functions inside the system and each 9 of them will be described in detailed below:

- UC001 - Login

  This use case allows the user to login into the system with their emails that have been given to them.

- UC002 - DisplayCourses

  This use case allows the student to see all the available courses that are for their major.

- UC003 - RegisterCourse

  This use case allows the student to pick any course that is available for them and register for it.

- UC004 - ViewRegisteredCourses

  This use case allows students to view their courses that they have registered for.

- UC005 - DropCourses

  This use case lets the students drop any course they feel, but only when allowed by a superior in the university if it is a wise move to drop the course for the situation the student is in.

- UC006 - ShowNotifications

  This use case will show the student a notification on their email about when the user has successfully been created.

- UC007 - CreateNewCourse

  This use case allows the administrators in the system to create a new course to the system for the students.

- UC008 - ModifyCourse

  This use case allows the administrators to modify a previous course in the system, either edit the name, credit, description or the lecturer.

- UC009 - RemoveCourse

  This use case allows the administrators to remove a course off the system, only when asked to remove it from their superiors.

4.2.1.3 Sequence Diagram

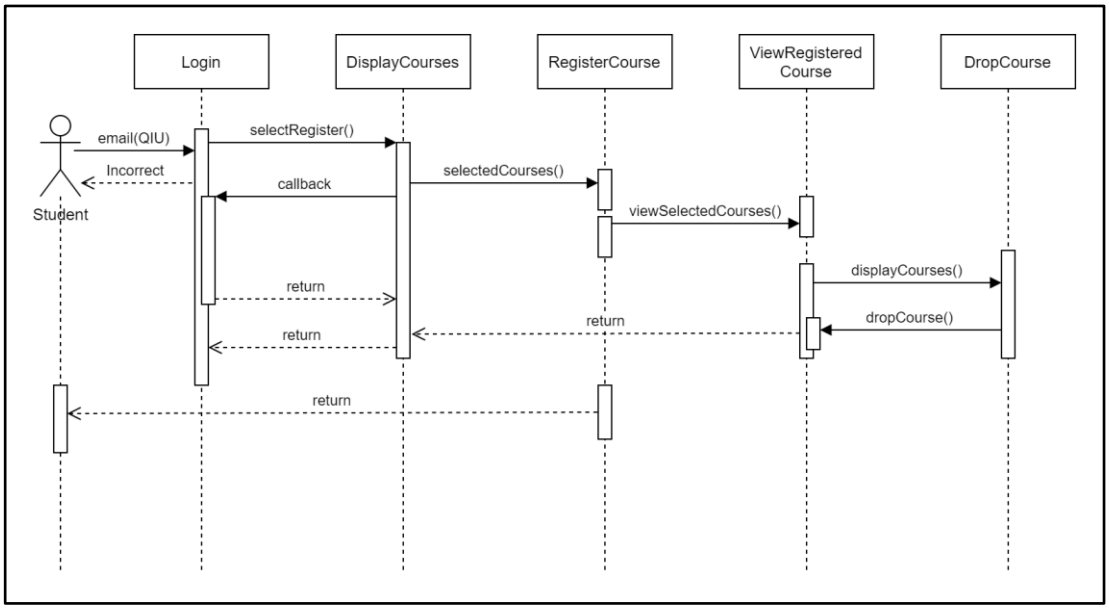


Figure 4.2 Sequence Diagram for Student OCR

4.2.1.4 Activity Diagram

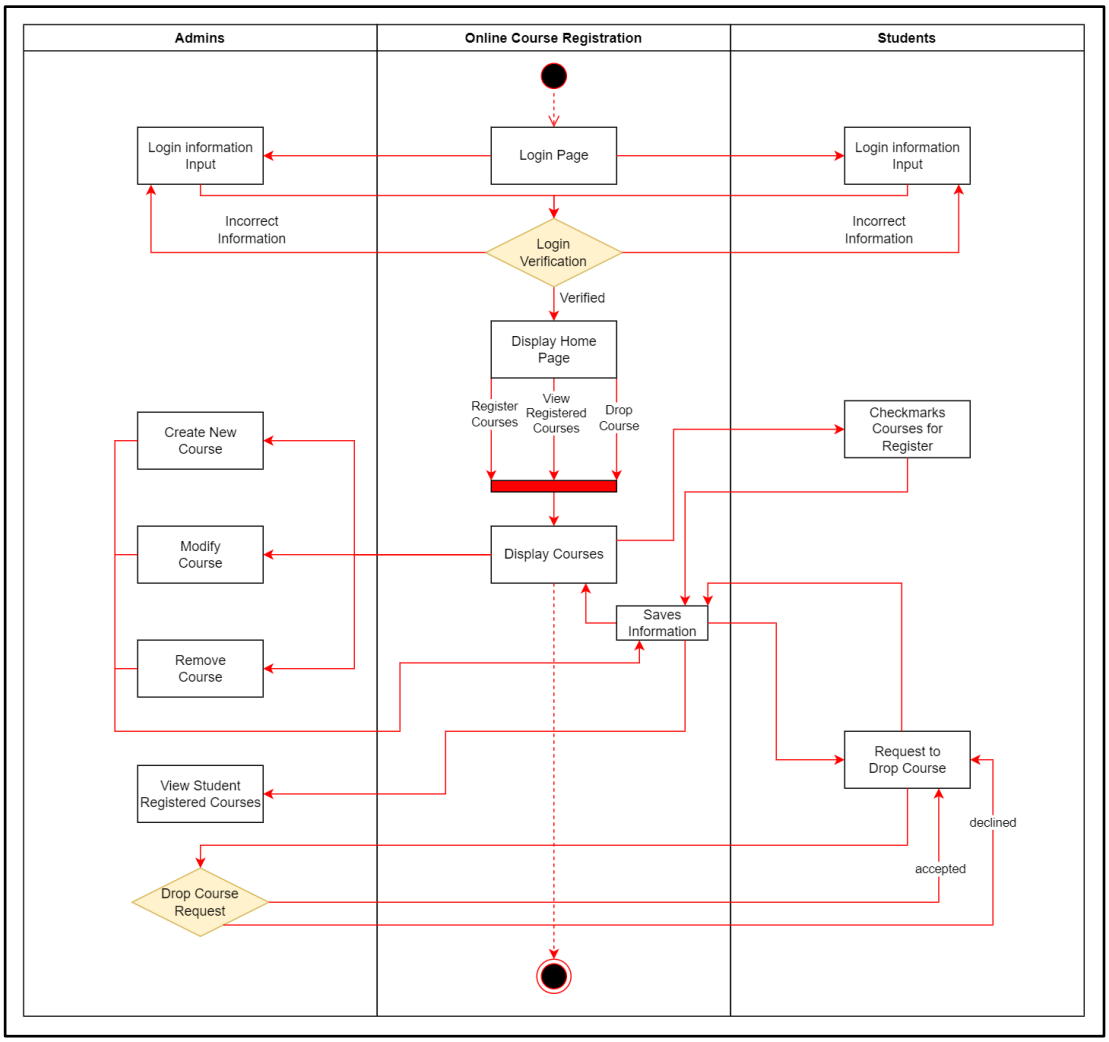


Figure 4.3 Activity Diagram for OCR

4.2.2 Non-Functional Requirements

Non-functional requirements are the characteristics of a system that do not relate directly to a specific function or feature, but rather to the overall performance, reliability, and usability of the system. “Nonfunctional requirements (NFR) must be addressed early in the software development cycle to avoid the cost of revisiting those requirements or re-factoring at the later stages of the development cycle.” (S. Farhat, 2009). In the context of the OCR - QIU system, some potential non-functional requirements might include:

1. **Performance**: The system should be able to handle a high volume of users and transactions without experiencing significant delays or downtime.
2. **Reliability**: The system should be reliable and consistently available to users.
3. **Scalability**: The system should be able to handle an increasing number of users as the user base grows.
4. **Security**: The system should be secure, protecting user data from unauthorized access or tampering.
5. **Usability**: The system should be easy to use and navigate, with a user-friendly interface.
6. **Accessibility**: The system should be accessible to users with disabilities, complying with relevant accessibility standards.
7. **Compatibility**: The system should be compatible with a range of devices.

4.3 Requirement Design

As stated in Honolulu (2012), "design requirements means those specifications and design criteria contained in the Contract that specify the minimum acceptable technical standards and define the limits within which the design of the Project shall be developed and conducted."

4.3.1 Class Diagram

Class diagrams are a type of graphical representation used to model the structure and relationships of a system. They are used to visualize the static structure of a system, including the classes (representing the objects or components in the system), their attributes (properties or characteristics), and their relationships (how they interact with each other).

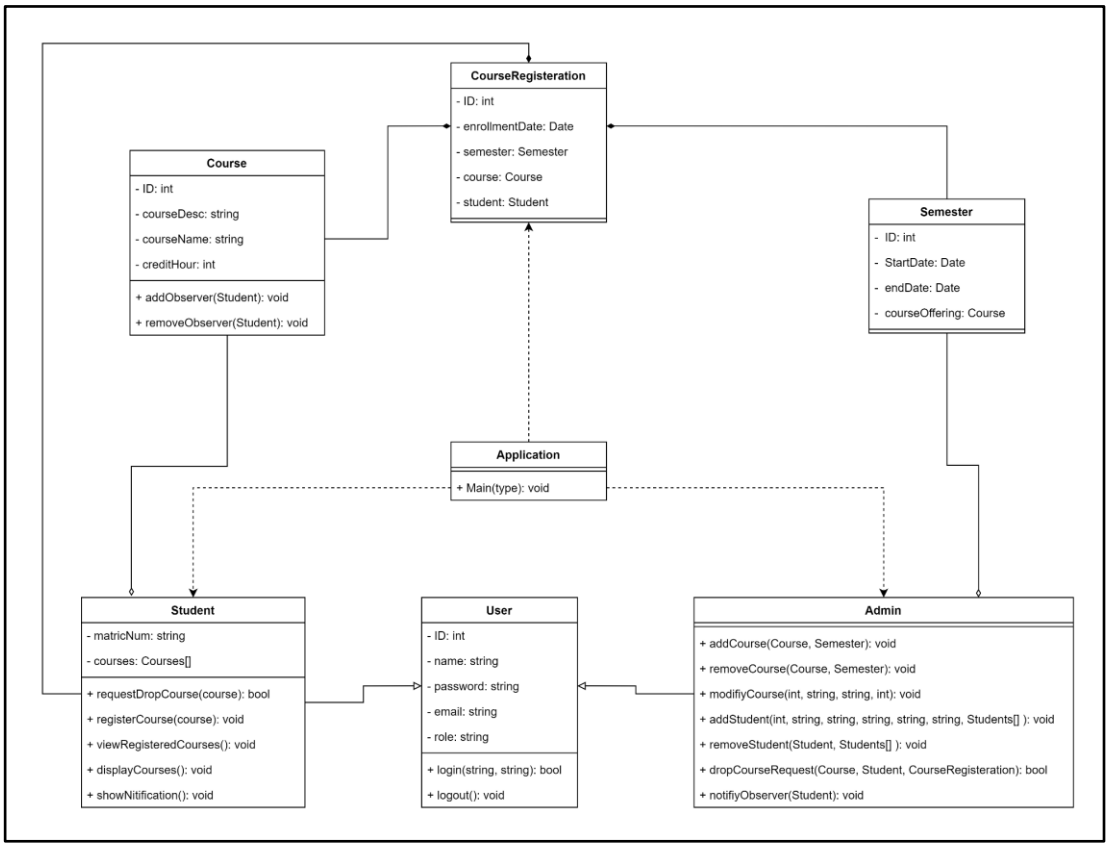


Figure 4.4 Class Diagram for OCR

4.4 Interface Design

The interfaces that will be shown will show the view of the student and also the administrators view on the system. The reason why an interface design prototype is so important as Beaubien, S (2021) says is that “it allows designers to quickly test out multiple ideas before committing to having developers code them. It also reduces the need for written explanations of a design. A prototype is worth a thousand static pages with documentation on how they’re supposed to work.” Down below the figures will show the student side of the application.

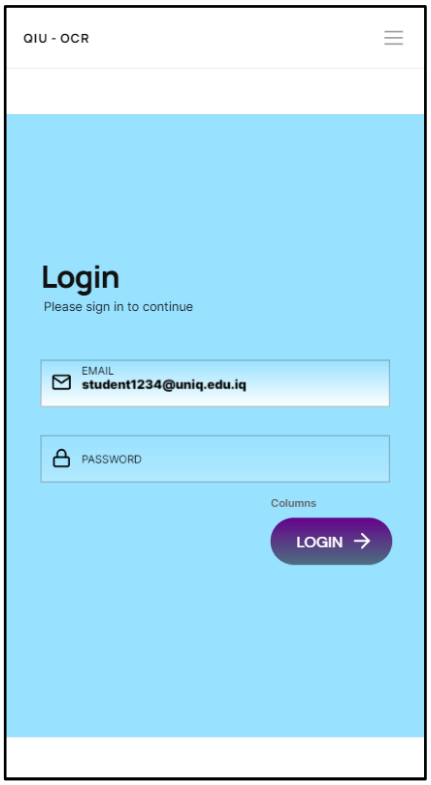


Figure 4.5 Login Page of OCR

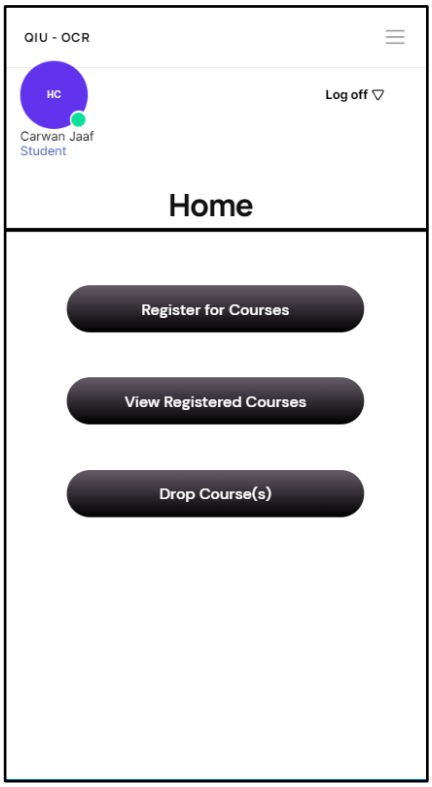


Figure 4.6 Home Page of OCR

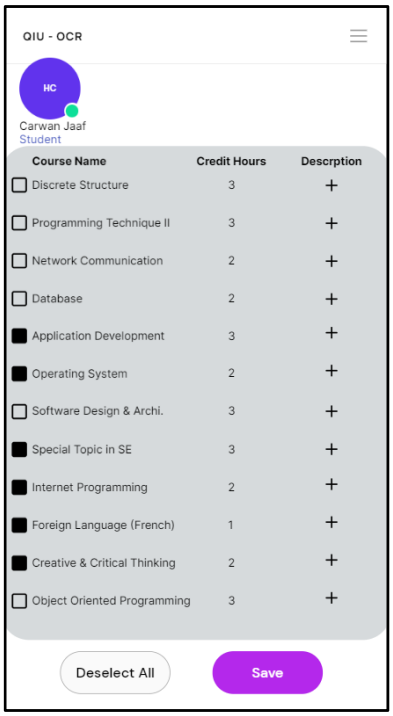


Figure 4.7 Register of OCR

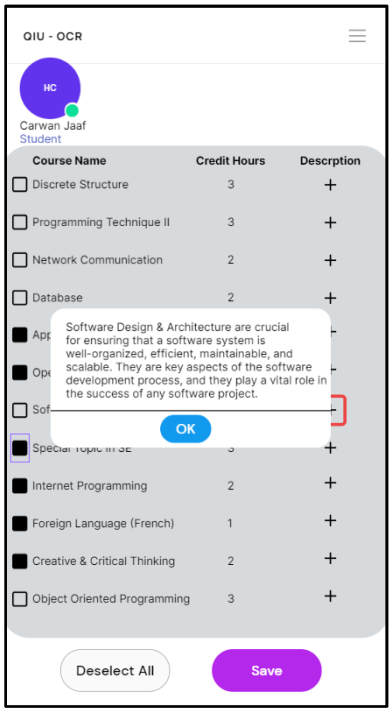


Figure 4.8 Description for Courses

4.5 Chapter Summary

This chapter explained the requirement analysis and design of the OCR application. It also has explained the functional and the non-functional requirements inside the system, including many various diagrams, making it visualizably easier to understand

CHAPTER 5

IMPLEMENTATION AND TESTING

5.1 Introduction

This chapter focuses on the implementation of the OCR Application based on the system development approach, requirements analysis, and design described in the previous chapters of the paper. It also provides a comprehensive overview of the outcomes from different tests conducted during the implementation phase, also including the backend and frontend code.

5.2 Coding of System Main Functions

The following code segment below shows the create_account component which allows the user to create an account if they don't have an account already. It will give the user data fields that need to be all filled in with the correct prerequisites, they will need to input their full name, age, gender, email, username, password, student code, and department.

```
      class CreateAccountScreen extends StatefulWidget {
       static String routeName = '/create_account';
       const CreateAccountScreen({Key? key}) : super(key: key);

       @override
       State<CreateAccountScreen> createState() =>
_CreateAccountScreenState();
      }

      class _CreateAccountScreenState extends State<CreateAccountScreen> {
       final _formKey = GlobalKey<FormState>();
       final TextEditingController _nameController =
TextEditingController();
       final TextEditingController _ageController =
TextEditingController();
```

```
        final TextEditingController _emailController =
TextEditingController();
        final TextEditingController _passwordController =
TextEditingController();
        final TextEditingController _confairmPasswordController =
            TextEditingController();
        final TextEditingController _usernameController =
TextEditingController();
        final TextEditingController _studentCodeController =
TextEditingController();
        String _gender = '';
        String _studentDepartment = '';



       @override
        Widget build(BuildContext context) {
          return Scaffold(
            extendBodyBehindAppBar: true,
            appBar: AppBar(
              backgroundColor: Colors.transparent,
              elevation: 0,
              title: const Text(
                "Create Account",
                style: TextStyle(fontSize: 24, fontWeight: FontWeight.bold),
              ),
            ),
            body: Container(
              width: MediaQuery.of(context).size.width,
              height: MediaQuery.of(context).size.height,
              decoration: BoxDecoration(
                gradient: LinearGradient(
                  colors: [
                    hexStringToColor("CB2B93"),
                    hexStringToColor("9546C4"),
                    hexStringToColor("5E61F4"),
                  ],
                  begin: Alignment.topCenter,
                  end: Alignment.bottomCenter,
                ),
              ),
              child: SingleChildScrollView(
                child: Padding(
                  padding: const EdgeInsets.fromLTRB(20, 120, 20, 0),
                  child: Form(
                    key: _formKey,
```

```
                    child: Column(
                      children: <Widget>[
                        inputField(
                          "Enter Full Name",
                          Icons.person,
                          false,
                          _nameController,
                          TextInputType.name,
                          validator: (value) {
                            if (value == null || value.isEmpty) {
                              return 'Please enter your name';
                            }
                            if (value.length < 4) {
                              return 'Name must be at least 3 characters';
                            }
                            if (RegExp(r"^[a-zA-Z]+(([',. -][a-zA-Z ])?[a-zA-
Z]*)*$")
                                    .hasMatch(value) ==
                                false) {
                              return 'Please enter a valid name';
                            }

                            return null;
                          },
                        ),
                        const SizedBox(height: 20),
                        inputField(
                          "Enter Age",
                          Icons.calendar_today,
                          false,
                          _ageController,
                          TextInputType.number,
                          validator: (value) {
                            if (value == null || value.isEmpty) {
                              return 'Please enter your age';
                            }
                            if (int.parse(value) < 18) {
                              return 'You must be at least 18 years old';
                            }
                            return null;
                          },
                        ),
                        const SizedBox(height: 20),
                        DropdownButtonFormField<String>(
                          decoration: InputDecoration(
                            labelText: "Gender",
```

```
                            prefixIcon: const Icon(Icons.transgender),
                            filled: true,
                            errorBorder: OutlineInputBorder(
                              borderRadius: BorderRadius.circular(20),
                              borderSide: const BorderSide(
                                  color: Colors.yellow,
                                  width: 2.0), // change as needed
                            ),
                            focusedErrorBorder: OutlineInputBorder(
                              borderRadius: BorderRadius.circular(20),
                              borderSide: const BorderSide(
                                  color: Colors.yellow,
                                  width: 2.0), // change as needed
                            ),
                            errorStyle:        const        TextStyle(color:
Colors.yellow),
                            fillColor: Colors.white,
                            border: OutlineInputBorder(
                              borderRadius: BorderRadius.circular(20),
                            ),
                          ),
                          onChanged: (String? newValue) {
                            setState(() {
                              if (newValue == null ||
                                  newValue.isEmpty ||
                                  newValue == '') {
                                return;
                              }
                              _gender = newValue;
                            });
                          },
                          validator: (value) {
                            if (value == null || value.isEmpty) {
                              return 'Please select your gender';
                            }
                            return null;
                          },
                          items: <String>['Male', 'Female', 'Other']
                              .map<DropdownMenuItem<String>>((String   value)
{
                            return DropdownMenuItem<String>(
                              value: value,
                              child: Text(value),
                            );
                          }).toList(),
                        ),
```

```
                        const SizedBox(height: 20),
                        inputField(
                          "Enter Email",
                          Icons.email,
                          false,
                          _emailController,
                          TextInputType.emailAddress,
                          validator: (value) {
                            if (value == null || value.isEmpty) {
                              return 'Please enter your email';
                            }
                            if  (RegExp(r"^[a-zA-Z0-9.]+@[a-zA-Z0-9]+\.[a-zA-
Z]+")
                                    .hasMatch(value) ==
                                false) {
                              return 'Please enter a valid email';
                            }
                            return null;
                          },
                        ),
                        const SizedBox(height: 20),
                        inputField(
                          "Enter Username",
                          Icons.person_outline,
                          false,
                          _usernameController,
                          TextInputType.name,
                          validator: (value) {
                            if (value == null || value.isEmpty) {
                              return 'Please enter your username';
                            }
                            return null;
                          },
                        ),
                        const SizedBox(height: 20),
                        inputField(
                          "Enter Password",
                          Icons.lock,
                          true,
                          _passwordController,
                          TextInputType.text,
                          validator: (value) {
                            if (value == null || value.isEmpty) {
                              return 'Please enter your password';
                            }
                            if (value.length < 8) {
```

```
                            return 'Password must be at least 8 characters';
                          }
                          if   (RegExp(r"^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)[a-
zA-Z\d]{8,}$")
                                  .hasMatch(value) ==
                              false) {
                            return  'Password  must  contain  at  least  one
uppercase letter, one lowercase letter and one number';
                          }
                          return null;
                        },
                      ),
                      const SizedBox(height: 20),
                      inputField(
                        "Confirm Password",
                        Icons.lock,
                        true,
                        _confairmPasswordController,
                        TextInputType.text,
                        validator: (value) {
                          if (value == null || value.isEmpty) {
                            return 'Please enter your Confirm password';
                          }
                          if (value.length < 8) {
                            return 'Password must be at least 8 characters';
                          }
                          if   (RegExp(r"^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)[a-
zA-Z\d]{8,}$")
                                  .hasMatch(value) ==
                              false) {
                            return  'Password  must  contain  at  least  one
uppercase letter, one lowercase letter and one number';
                          }
                          if (value != _passwordController.text) {
                            return 'Password does not match';
                          }
                          return null;
                        },
                      ),
                      const SizedBox(height: 20),
                      inputField(
                        "Enter Student Code",
                        Icons.vpn_key,
                        false,
                        _studentCodeController,
                        TextInputType.text,
```

```
                validator: (value) {
                  if (value == null || value.isEmpty) {
                    return 'Please enter your student code';
                  }
                  if (value.length < 8) {
                    return 'Student code must be at least 8
characters';
                  }
                  return null;
                },
              ),
              const SizedBox(height: 20),
              DropdownButtonFormField<String>(
                decoration: InputDecoration(
                  labelText: "Student Department",
                  prefixIcon: const Icon(Icons.school),
                  filled: true,
                  errorBorder: OutlineInputBorder(
                    borderRadius: BorderRadius.circular(20),
                    borderSide: const BorderSide(
                        color: Colors.yellow,
                        width: 2.0), // change as needed
                  ),
                  focusedErrorBorder: OutlineInputBorder(
                    borderRadius: BorderRadius.circular(20),
                    borderSide: const BorderSide(
                        color: Colors.yellow,
                        width: 2.0), // change as needed
                  ),
                  errorStyle:        const        TextStyle(color:
Colors.yellow),
                  fillColor: Colors.white,
                  border: OutlineInputBorder(
                    borderRadius: BorderRadius.circular(20),
                  ),
                ),
                onChanged: (String? newValue) {
                  setState(() {
                    if (newValue == null ||
                        newValue.isEmpty ||
                        newValue == '') {
                      return;
                    }
                    _studentDepartment = newValue;
                  });
                },
```

```
                            validator: (value) {
                              if (value == null || value.isEmpty) {
                                return 'Please select your department';
                              }
                              return null;
                            },
                            items: <String>[
                              'IT',
                              'Software Engineering',
                              'Human Resource',
                              'Business',
                              'Network and Security',
                            ].map<DropdownMenuItem<String>>((String value) {
                              return DropdownMenuItem<String>(
                                value: value,
                                child: Text(value),
                              );
                            }).toList(),
                          ),
                          const SizedBox(height: 20),
                          Container(
                            width: MediaQuery.of(context).size.width,
                            height: 50,
                            margin: const EdgeInsets.fromLTRB(0, 10, 0, 20),
                            decoration: BoxDecoration(
                              borderRadius: BorderRadius.circular(90),
                            ),
                            child: ElevatedButton(
                              onPressed: () {
                                signUpUser();
                              },
                              style: ButtonStyle(
                                backgroundColor:
                                    MaterialStateProperty.resolveWith((states)
{
                                  if   (states.contains(MaterialState.pressed))
{
                                    return Colors.black26;
                                  }
                                  return Colors.white;
                                }),
                                shape:

MaterialStateProperty.all<RoundedRectangleBorder>(
                                  RoundedRectangleBorder(
                                    borderRadius: BorderRadius.circular(30),
```

```
                            ),
                          ),
                        ),
                        child: const Text(
                          'CREATE ACCOUNT',
                          style: TextStyle(
                            color: Colors.black87,
                            fontWeight: FontWeight.bold,
                            fontSize: 16,
                          ),
                        ),
                      ),
                    ),
                  ),
                ],
              ),
            ),
          ),
        ),
      ),
    );
  }
}
```

Figure 5.1 Account Creation

Also some part of the code segment is linked with Firebase to allow it to create the user into the database with all the information the user has inputted. Firebase's real-time database ensures that course registration data is synchronized in real time across devices. Additionally, Firebase's scalability and ease of use make it an ideal choice for developing and maintaining an efficient and reliable online course registration application. Down below will show how the code links with the Firebase database.

```
void signUpUser() async {
if (_formKey.currentState!.validate()) {
context.read<FirebaseAuthMethods>().signUpWithEmail(
user: {
'name': _nameController.text,
'age': int.parse(_ageController.text),
'email': _emailController.text,
'password': _passwordController.text,
'username': _usernameController.text,
'studentCode': _studentCodeController.text,
'gender': _gender,
```

```
'studentDep': _studentDepartment,
},
password: _passwordController.text,
context: context,
).then((value) => context.router.replace(const LoginRoute()));
}
}
```

Figure 5.2 Firebase Connection

5.3 Interfaces of System Main Functions

Video provides a powerful way to help you prove your point. When you click Online Video, you can paste in the embed code for the video you want to add. You can also type a keyword to search online for the video that best fits your document. To make your document look professionally produced, Word provides header, footer, cover page, and text box designs that complement each other. For example, you can add a matching cover page, header, and sidebar.

5.4 Testing

The purpose of system testing for an online course registration application is to ensure that the system functions correctly, meets the specified requirements, and performs reliably. It validates the application's functionality, usability, performance, security, integration, error handling, and stability. By conducting system testing, any defects or issues can be identified and addressed before the application is deployed, resulting in a reliable software product. Therefore having two types of tests, black box testing, and white box testing, each has its own different style of testing the application.

5.4.1 Black box Testing

Black box testing is a software testing technique where the internal structure, design, and implementation of the system being tested are not known to the tester. Instead, the testers focus on validating the system's functionality and behavior based on its inputs and expected outputs.

Black box testing is important for the online course registration application because it helps ensure that the application functions correctly from a user's perspective. It allows testers to assess the application's usability, functionality, and compatibility without knowledge of the underlying code or system architecture. This approach helps identify potential issues, such as incorrect input handling, missing features, or inconsistencies in the application's behavior.

Test Case ID	TC001	TC002	TC003	TC004
Full name	Carwan Jaaf	John Paul	Carwan Jaaf	Jack Black
Age	22	18	15	26
Email	jaafcarwan @gmail.com	jp@gmail.com	cj@gmail.com	bj@gmail/h hh
Username	Carwan	John	Carzz	Jack
Password	carwan	Qwerty1	Qq12345	Jacky123
Confirm Password	carwan658	Qwerty1	Qq12345	Jacky123
Student Code	QIU09257	QIU84724	37582	HFD36593
Department	Software Engineering	Biomedical	Human Resource	Software Engineering
Semester	Semester 1	Semester 2	Semester 1	Semester 1
Expected Result	**Actual Result**			
Error message input data is filled out or not	✓	✓	✓	✓
Error message user is above 18	✓	✓	✗	✓
Error message	✓	✓	✓	✗

email is available				
Error message password has 1 capital, 1 non-capital, and a number	✗	✓	✓	✓
Error message confirm password is identical to password	✗	✓	✓	✓
Error message student code is 8 letters/digits	✓	✓	✗	✓

Figure 5.3 Black Box Tests

5.4.1.1 System Flow

The system flow of the online course registration starts with the user authentication, typically the user does not have an account so for this, they can click on the create account tab which will take them to a page where they can input all of the data displayed to them. After they complete creating an account the next part is to login with their email and password of their choosing. When they login, it will take them to the home screen which will display their full name, student department, and student code. Also giving the user three functionalities which are, course catalog, registered courses, and drop course(s). If they click the course catalog, it will display all available courses for their department and let the user be able to choose and save it. If they choose registered courses, it will show their courses that they have registered for, if they haven't registered for any course then it won't show them anything. Finally if they choose to click drop course(s), it will allow them the courses that they have registered and will allow them to drop the course they want to choose, but it won't instantly drop the course but it will notify the admin of the application so they will see if they accept it or not.

5.5 Chapter Summary

This chapter focuses on implementing the OCR Application based on the system development approach and conducting system testing for the online course registration application. It highlights the importance of black box testing for validating functionality from a user's perspective. The system flow includes user authentication, account creation, login, and access to features such as course catalog, registered courses, and drop course(s). Overall, the chapter emphasizes the significance of testing and provides an overview of the implemented application and its functionalities.

CHAPTER 6

CONCLUSION

6.1 Introduction

It is important to identify and document both functional and non-functional requirements for a system in order to ensure that the final product meets the needs of the user and is fit for purpose. Functional requirements describe the specific capabilities, behaviors, and characteristics that the system must have, while non-functional requirements describe the overall performance, reliability, and usability of the system. Diagrams provide a high-level overview of the system and its capabilities, and can help to identify potential design issues or areas for improvement. Comparing the requirements and diagrams of a system with those of similar systems can also be helpful in the development process. This can provide insight into best practices and identify areas where the system can be improved or differentiated from its competitors. Overall, finding and documenting the functional and non-functional requirements, creating diagrams, and comparing the system to others can help to ensure that the system is well-designed and effective.

6.2 Project Results and Analysis

There are several results that have come upon this project, one can be the effectiveness of the online course registration system in terms of user satisfaction, efficiency, and accuracy. Also the impact it has on the students outcomes to examine how well the system will be for the students retention rates. Finally, the integration with the other systems allows one to see what other mistakes the other systems have made and for this new OCR system to not make the same mistake as the previous systems.

6.3 Future Works

For future work on the system, to enhance the user experience might be adding a new option to allow the student to either get a notification either via SMS, phone call, or mobile banner. This will allow the user to feel more open to the system and feel safe when using it. Also maybe a feature to see which lecturer is teaching what lecture and read more about the lecturer to see if the student feels comfortable registering for the lecture. Finally the last feature might be an option to click forget password and it directly sends a notification to the QIU emails that are given to the QIU students and allows the student to easily restore their account.

REFERENCES

Simplilearn. “What Is Requirement Analysis: Applications, Techniques and Tools Used [2022 Edition].” Simplilearn.com, Simplilearn, 18 Nov. 2022, https://www.simplilearn.com/what-is-requirement-analysis-article. .

“206.195.188.174.” HONOLULU RAIL TRANSIT PROJECT, Honolulu, 2012, http://206.195.188.174/docushare/dsweb/Get/Document-16396/Special_Provisions_Chapters_1-7.pdf.

Beaubien, Sean, et al. “An Awesome Guide to Prototyping in User Interface Design.” CareerFoundry, 5 Aug. 2021, https://careerfoundry.com/en/blog/ui-design/the-value-of-prototyping-in-ui-design/#:~:text=It%20allows%20designers%20to%20quickly,they're%20supposed%20to%20work.

University, S. Farhat Nova Southeastern, et al. “Refining and Reasoning about Nonfunctional Requirements: Proceedings of the 47th Annual Southeast Regional Conference.” ACM Other Conferences, 1 Mar. 2009, https://dl.acm.org/doi/abs/10.1145/1566445.1566497.

Cherry, Kendra. “How Surveys Are Used in Psychology to Collect Data.” Verywell Mind, Verywell Mind, 7 Apr. 2020, https://www.verywellmind.com/what-is-a-survey-2795787#:~:text=A%20survey%20can%20be%20used,religion%2C%20ethnicity%2C%20and%20income.

Sherman, Rick. “Waterfall Methodology.” Waterfall Methodology - an Overview | ScienceDirect Topics, 2015, https://www.sciencedirect.com/topics/computer-science/waterfall-methodology.

“Aims and Objectives - Guide for Thesis and Dissertations.” DiscoverPhDs, 7 Jan. 2021, https://www.discoverphds.com/advice/doing/research-aims-and-objectives.

“Course Registration.” Concordia University, https://www.concordia.ca/students/registration.html#undergraduate.

Home - Ignatius Ajuru University of Education. https://iauoe.edu.ng/.

“When to Use Waterfall vs. Agile.” Macadamian, 17 May 2019, https://www.macadamian.com/learn/when-to-use-waterfall-vs-agile/.

Appendix B Software Requirement Specification

1. INTRODUCTION

1.1 **Purpose**

This SRS describes the Qaiwan International University - Online Course Registration (QIU - OCR) system. The reason this topic has been picked is because of the situations that many students have been a part of for almost as long as they have been at Qaiwan International University. The main point of the project is for students to register for their courses online rather than on campus. The registration system at Qaiwan International University today isn't much modern as it should be, it still requires students to come to the campus to register for their courses. The overview is to allow students to access this website or application with their login address and password and pick which course they wish to pick for the ongoing semester.

1.2 **Scope**

The software product is focused on making the students' life easier in a way that they won't need to go all their way to the campus just for a simple signature and a stamp which can be easily solved with a program. That will allow them to choose the courses that they wish and the program will see if the student is approved or not. Once the program accepts the student, it will send it over to the managers of the students for them to double check just in case of any situation. The program will be on a mobile application when implemented perfectly. This program aims to satisfy the students' college life and the employees' jobs.

1.3 **Definitions, Acronyms, and Abbreviations**

OCR - Online Course Registration

QIU - Qaiwan International University

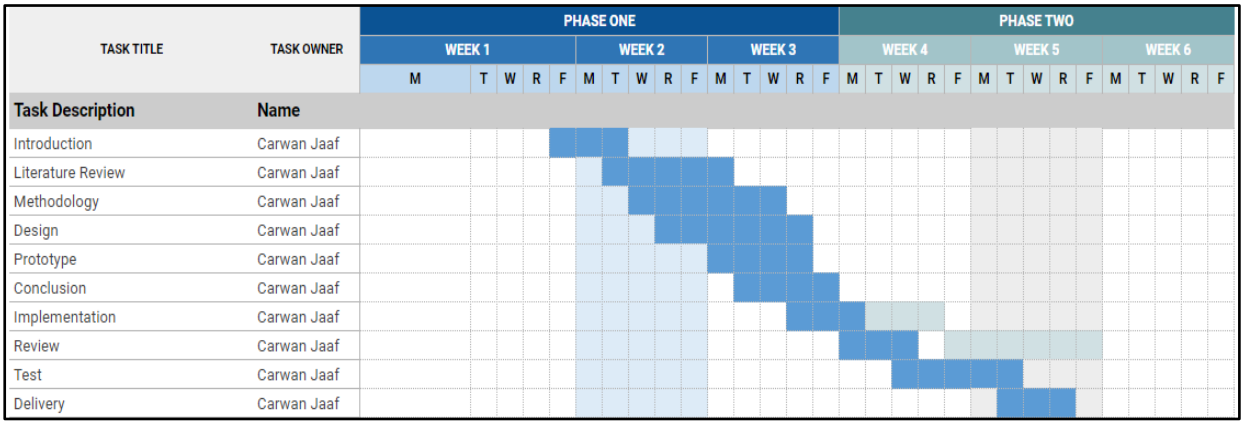


Figure 6.1 Gantt Char

1.4 **References**

Beaubien, Sean, et al. “An Awesome Guide to Prototyping in User Interface Design.” CareerFoundry, 5 Aug. 2021, https://careerfoundry.com/en/blog/ui-design/the-value-of-prototyping-in-ui-design/#:~:text=It%20allows%20designers%20to%20quickly,they're%20supposed%20to%20work.

“Difference between Hardware and Software.” GeeksforGeeks, 2 Dec. 2022, https://www.geeksforgeeks.org/difference-between-hardware-and-software/.

1.5 **Overview**

An overview of this SRS document includes additional information such as a list of requirements, a glossary of terms, and any supplementary documentation. This document will help the software development team to have a clear understanding of the requirements of the system which they are going to build and also it will help in communication with the client, stakeholders, and end-users during the software development life cycle. of this SRS document. It also includes diagrams of the systems module to visualize for the reader.

2. OVERALL DESCRIPTION

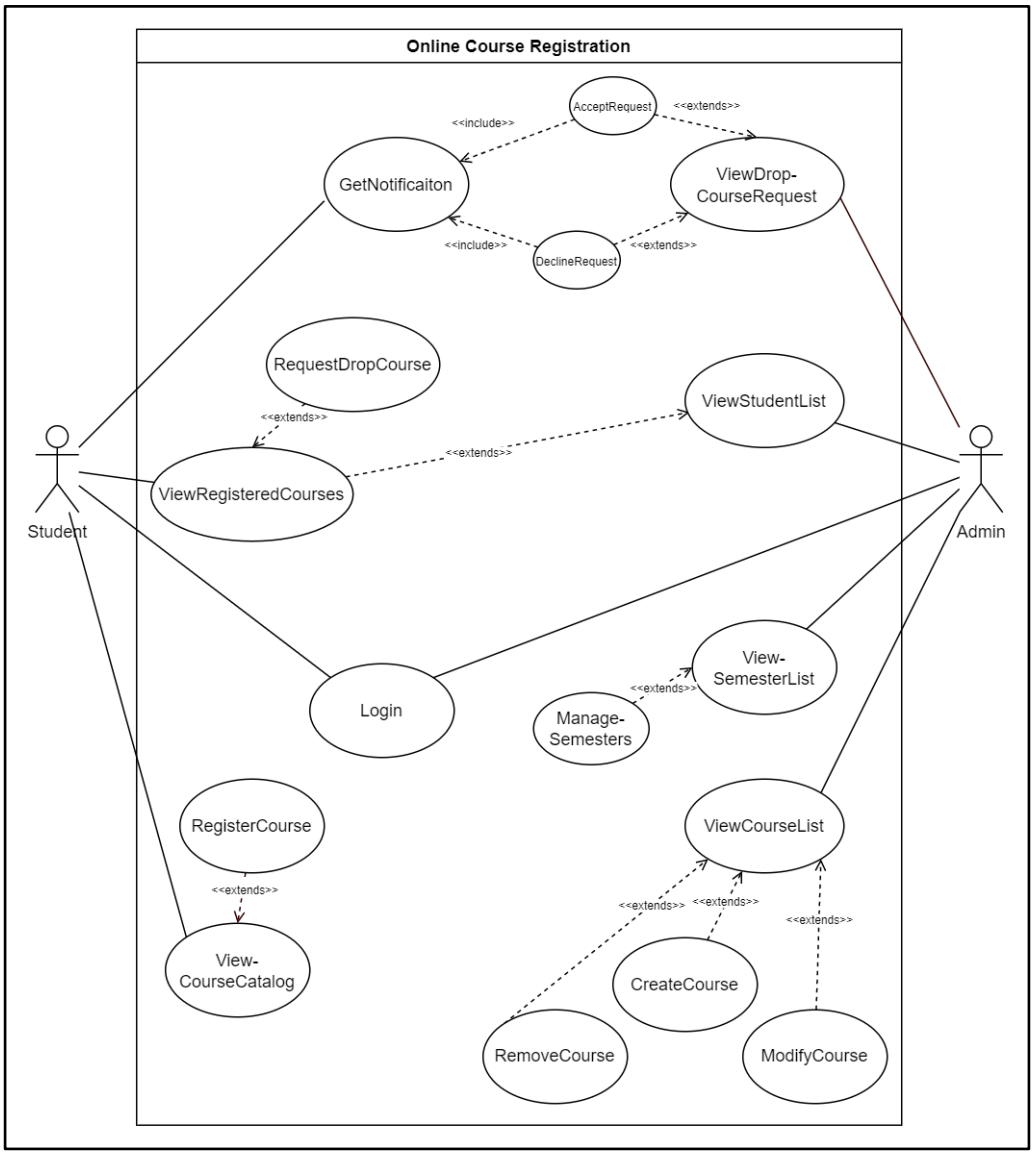


Figure 2.1: Use Case Diagram of QIU - OCR

2.1 **Product Perspective**

An online course registration system can be viewed from several perspectives, depending on the role of the stakeholders involved. From a student's perspective, an online course registration system is an application that allows them to browse for available courses and have the option

to register for the available courses when opened to students. Also from the department administrator's perspective, the system can be useful to allow them to manage their course registration online, view how many students have enrolled for which course, and other details. They can also find the revenue generated from the courses, and also to manage the user accounts.

2.1.1 System Interfaces

The Qaiwan International University Online Course Registration (QIU-OCR) system is a comprehensive and user-friendly software that allows students and administrators to manage courses and schedules effectively.

The system's login feature is restricted to QIU students and administrators, ensuring that only authorized users have access to the system. The display courses functionality allows students to view the available courses for their specific major, making it easier for them to plan their schedules. The register courses functionality allows students to select and register for the courses that are most suitable for them, and the view registered courses feature allows them to keep track of their enrolled classes. Additionally, students have the option to drop courses they no longer wish to attend.

The admin side of the system also has a range of functionalities, such as the ability to create new courses and make changes to existing ones. This includes adding course name, credit, code, description, and lecturer information, as well as modifying course name, lecturer, code, and credit. Admins also have the ability to remove courses, however, this requires approval.

The QIU-OCR system is a comprehensive and effective solution that meets the needs of both students and administrators. It allows students to plan their schedules, manage their registered courses and provide the necessary functionalities for administrators to manage the courses and the students in the University.

2.1.2 User Interfaces

The first time the user opens the application, they will be given a login page screen with no registering near it, the reason there won't be any registering for the system is because the QIU student will be given their login information beforehand, and their email address will be their emails. If the student writes their login information incorrectly, the system will display to the user that their login information is incorrect and will ask them to please try again. If the user keeps getting the login information incorrectly, they will have to contact an admin on the system and see what their login information is with proof that it is their account and not a fraud. When the user logs into the system, they will be met with the homepage of the system and will be given many different options, such as, register for courses, view registered courses, and drop courses. The last two options are for the students who have already registered for courses, but the first option “register for courses” will be for those who are looking to register for courses. When the student clicks on register for courses, they will be met with a page of courses that are available for their major and requirements. They can scroll through the courses and choose which courses they prefer and read the descriptions of each course. Once they pick the courses for that semester all they will have to do next is click save and it will send them to a confirmation page where they will accept and agree to the contract.

2.1.3 Hardware Interfaces

There are several hardware interfaces that can be used for the QIU - online course registration mobile application. These include:

1. Push notifications: Can be used to send reminders or updates about upcoming classes or deadlines.

2. Touchscreen: Allows for easy navigation and input of information in the application.
3. Network connectivity: Allows for communication with the server to retrieve and submit course information, such as servers or databases.
4. Scanners: can be used to digitize paper documents, such as registration forms or ID cards, for electronic submission.
5. Printer: can be used for printing student records, transcripts or other documents.

2.1.4 **Software Interfaces**

An online course registration system can use a variety of software interfaces to provide functionality and allow users to interact with the system. Such as:

1. User interface (UI): A graphical user interface (GUI) that allows students to navigate the system, search for classes, and register for courses.
2. Application programming interface (API): A set of routines, protocols, and tools for building software and integrating it with other systems. APIs can be used to allow other systems, such as a student information system to interact with the course registration system.
3. Database interface: A way for the system to interact with a database, such as a relational database management system (RDBMS), that stores information about courses, students, and registration data.
4. Communication interfaces: A way for the system to send and receive email, SMS to students about their registration status or upcoming available classes.

2.1.5 **Memory**

The OCR system will consist of many memory allocations for a safety reason, and each memory will have its own job on what it will save or not save temporally, such as:

1. Random Access Memory (RAM): This is the memory that the system uses to temporarily store information that is currently being used or processed. This type of memory is fast, but it is also volatile, meaning it is lost when the system is closed.
2. Database memory: this is the memory allocated to the database server, which stores all the data related to the registration system.

2.2 **Product Functions**

The online course registration system has the following product functions:

1. Display courses: Allows students to view a list of available courses, including course information such as the course name, description, schedule, and prerequisites.
2. Register for courses: Allows students to select courses and add them to their schedule.
3. View registered courses: Allows students to view the courses they have registered for, including the course schedule and any other relevant details.
4. Drop courses: Allows students to remove courses from their schedule, which may have certain restrictions like deadlines.
5. Create new course: Allows authorized users, such as lecturers or administrators, to create new courses and add them to the course catalog.
6. Modify course: Allows authorized users to edit the information of existing courses such as course schedules or course description
7. Remove course: Allows authorized users to remove a course from the course catalog, usually when it's not available anymore or when the class is not enough to run.

2.3 **User Characteristics**

The online course registration system is typically used by a variety of user groups, each with their own unique characteristics. Some examples of user characteristics include:

1. Students: They are the main users of the system and typically use it to search for and register for courses, view their registered courses, and drop courses. They may have varying levels of computer literacy and experience with online systems.
2. Lecturers: They may use the system to view their course schedules, and access student information. They may have varying levels of computer literacy and experience with online systems.
3. Administrators: They use the system to manage the course catalog, approve course registrations, and generate reports on enrollment and course demand. They may have a high level of computer literacy and experience with online systems.

2.4 **Constraints**

The online course registration system has a variety of constraints that can affect its design and functionality. Some examples include:

1. Technical: limitations in the hardware and software platforms used by the system, which can affect its scalability, performance, and user experience.
2. Budget: which can limit the resources available for the development and maintenance of the system.
3. Security: to ensure that sensitive information such as personal data is protected against unauthorized access or breaches.
4. Legal: compliance with regulations and laws related to data protection and privacy.
5. Scalability: Ensuring that the system can handle a large number of concurrent users and requests, without causing performance issues.

6. Data Management: Ensuring that the data stored in the system is accurate, consistent and reliable, also ensuring data privacy and security, also backup and recovery of data.

3. SPECIFIC REQUIREMENTS

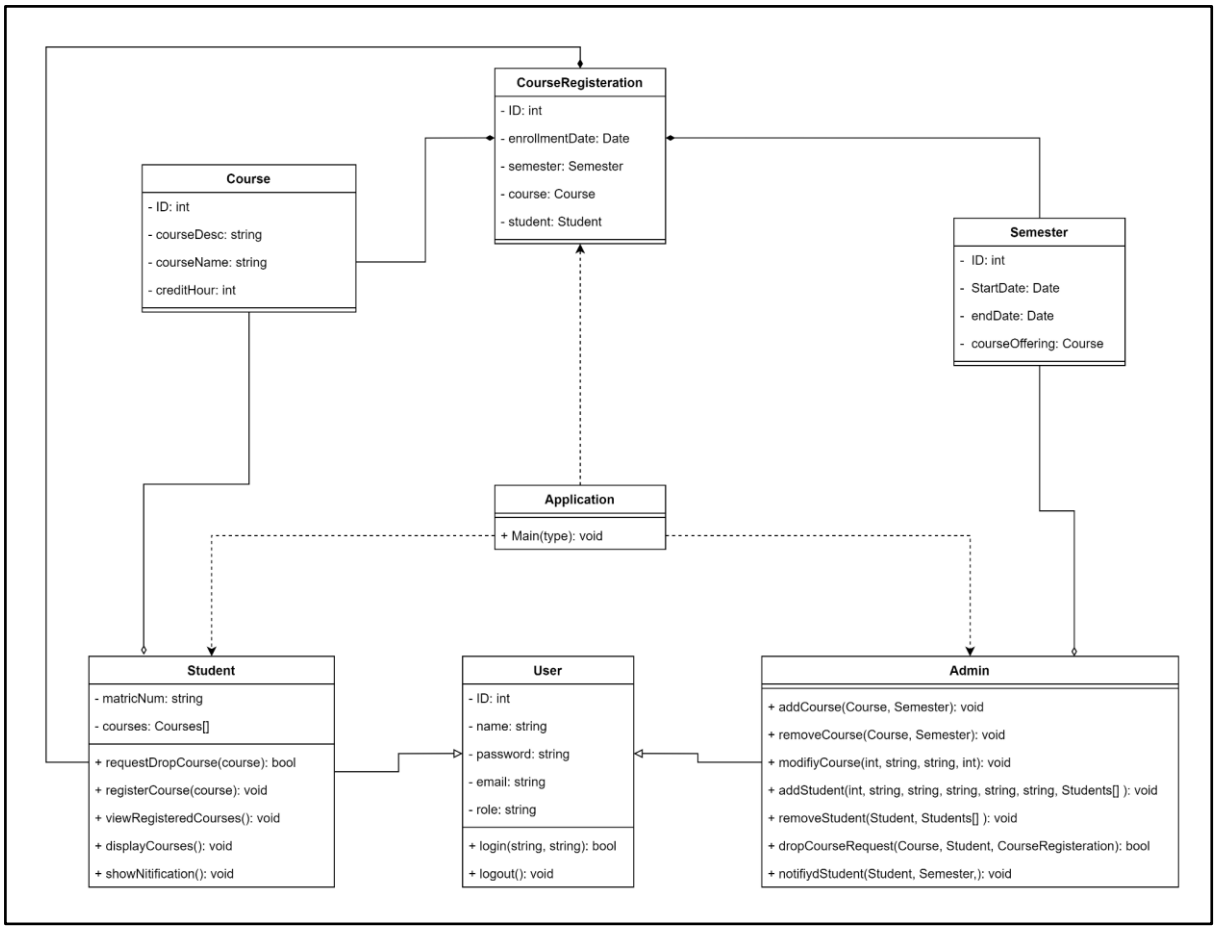


Figure 3.1: Domain Model of QIU - OCR

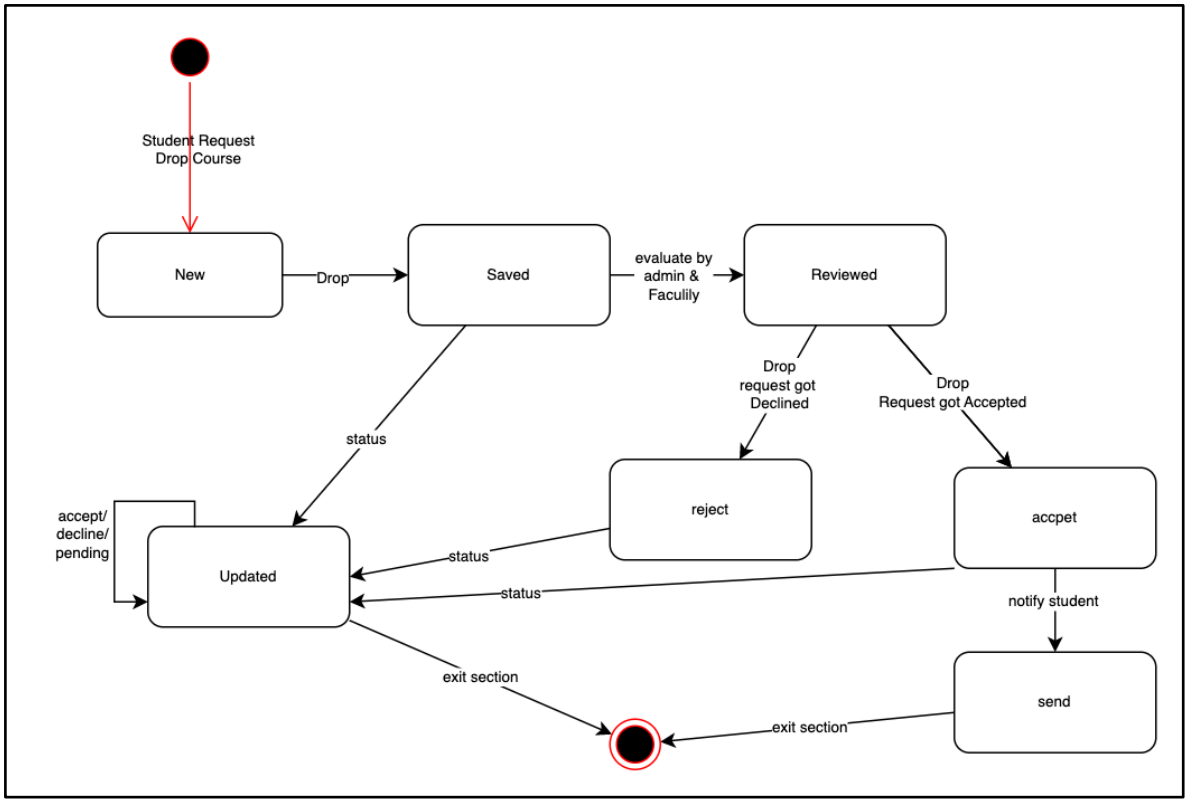


Figure 3.2: State Machine Diagram of Course

3.1 System Features

3.1.1 Module Student System

State briefly the functional requirements (use cases) that are available in this module. Better to include the diagram of the specific module (or the example of Customer Support System – by subsystem, see example below) from the overall use case diagram in Figure 2.1.

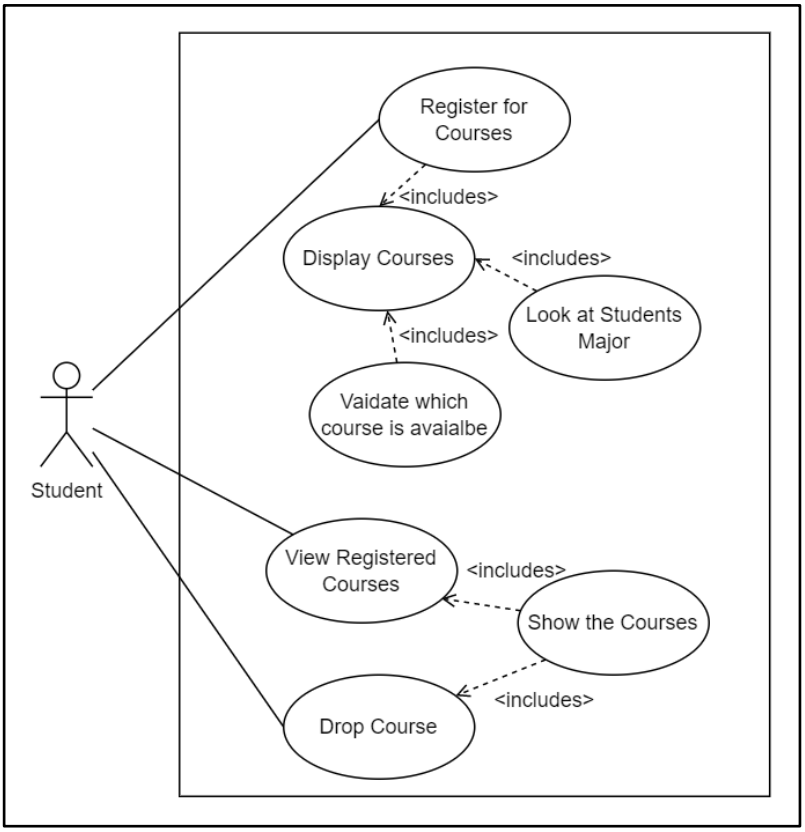


Figure 3.2: Student System

3.1.1.1 UC001: Use Case Register for Courses

Use Case Name:	Register for Courses
Scenario:	Student wants to register for their courses
Triggering Event:	Email notification alerted all the QIU students that the courses are available to register to
Breif Description:	When the student clicks on the Register for Courses tab, it will check the student's major and see what courses are available for them
Actors:	Student
Preconditions:	The student must meet certain requirements, such as being admitted to the institution, being in good academic standing, or having completed certain prerequisites.
Postconditions:	The student will receive a confirmation that they have successfully registered for the selected courses.
Flow of Events:	1. Login into the application
	2. Click Register for Courses
	3. Checkmark the courses that they want
	4. Click Save button to register for the marked courses
	5. Fill out the consent form
Exception Conditions:	1. The Login information is incorrect
	2. The course cannot be chosen because of some other requirements the student needs

FIgure 3.1: Use Case Description for Register for Courses

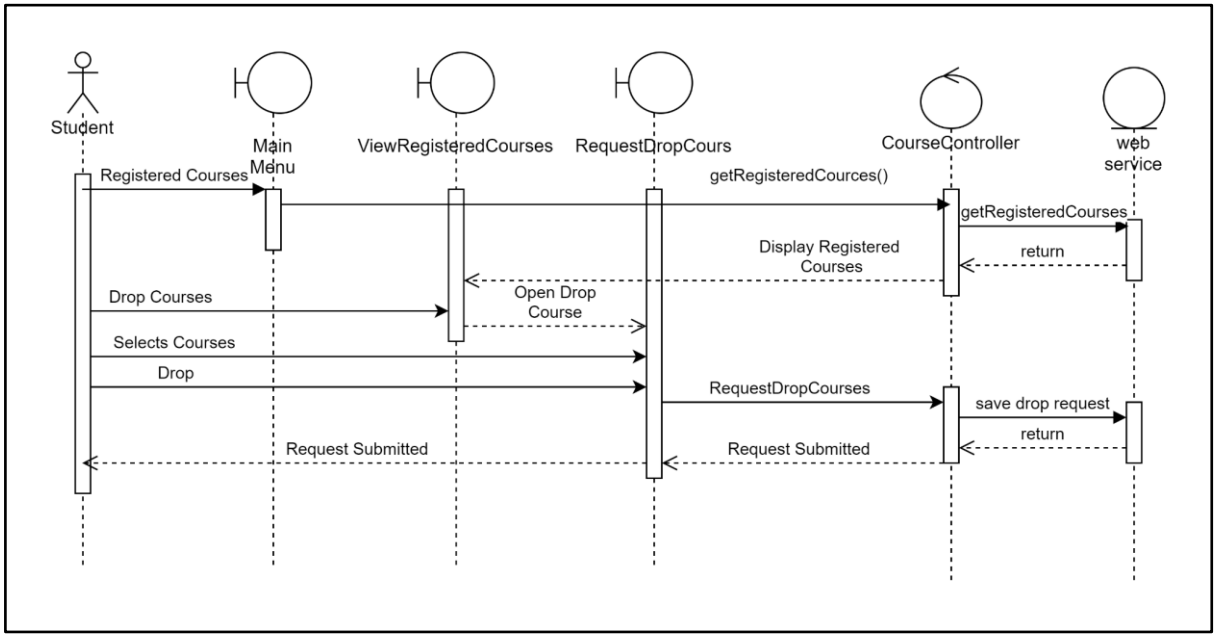


Figure 3.3: System Sequence Diagram of Register for Courses

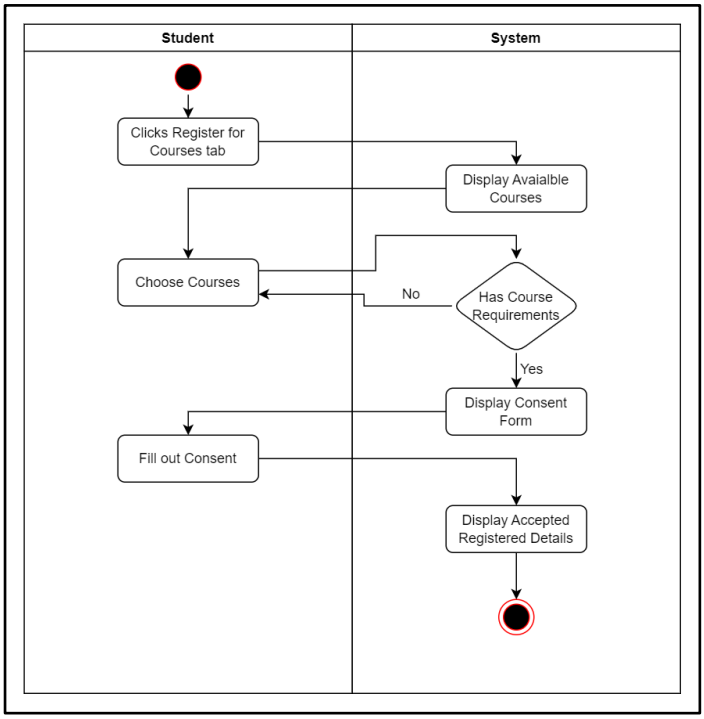


Figure 3.4: Activity Diagram of Register for Courses

3.1.1.2 UC002: Use Case View Registered Courses

Use Case Name:	View Registered Courses
Scenario:	Student wants to see their registered courses
Triggering Event:	They want to see what courses they have registered for
Breif Description:	When the student clicks on the View Registered Courses tab, it will show the courses that the student has registered for
Actors:	Student
Preconditions:	The student must of already registered for courses inorder for the function to work.
Postconditions:	The student will be shown a list of courses that they have registered for.
Flow of Events:	1. Login into the application
	2. Click View Registered Courses
Exception Conditions:	1. The Login information is incorrect
	2. The course cannot be chosen because of some other requirements the student needs

Table 3.2: Use Case Description for View Registered Courses

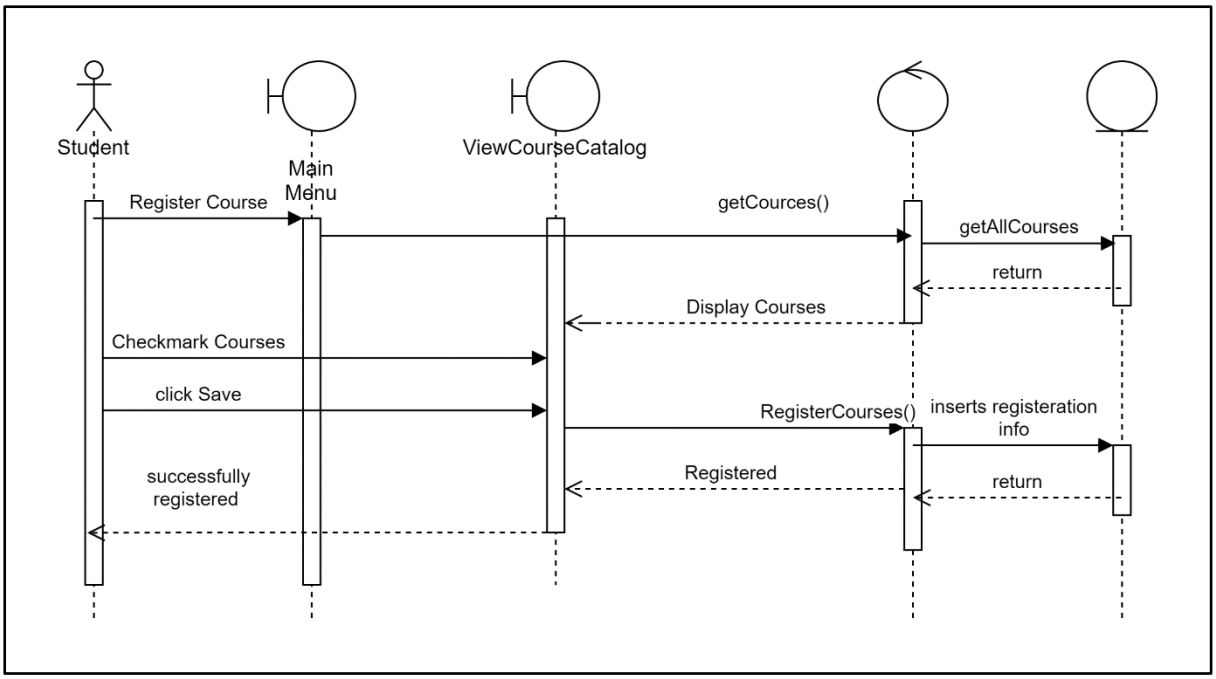


Figure 3.5: System Sequence Diagram of View Registered Courses

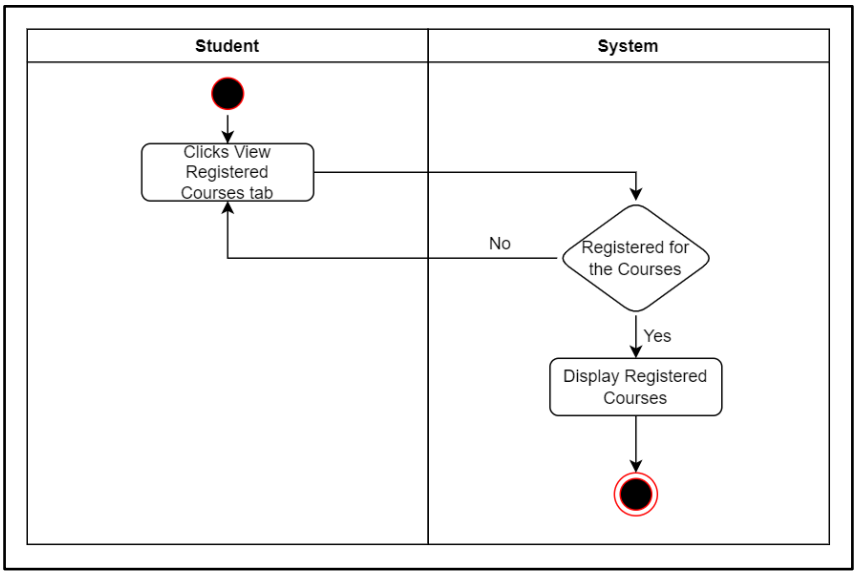


Figure 3.6: Activity Diagram of View Registered Courses

3.1.1.3 UC003: Use Case Drop Courses

Use Case Name:	Drop Courses
Scenario:	Student wants to drop a course in their scheduale because it is to hard for them
Triggering Event:	They want to drop a course
Breif Description:	When the student clicks on the Drop Courses tab, it will show the courses that the student has registered for and let the student choose which one they wish to drop
Actors:	Student, Admin
Preconditions:	The student must of already registered for courses inorder for the function to work.
Postconditions:	The student has removed the course off their scheduale
Flow of Events:	1. Login into the application
	2. Click Drop Courses
	3. Choose the course that wants to be dropped
	4. Click Save
Exception Conditions:	1. The Login information is incorrect
	2. The course cannot be dropped due to its importance

Figure 3.2: Use Case Description for Drop Courses

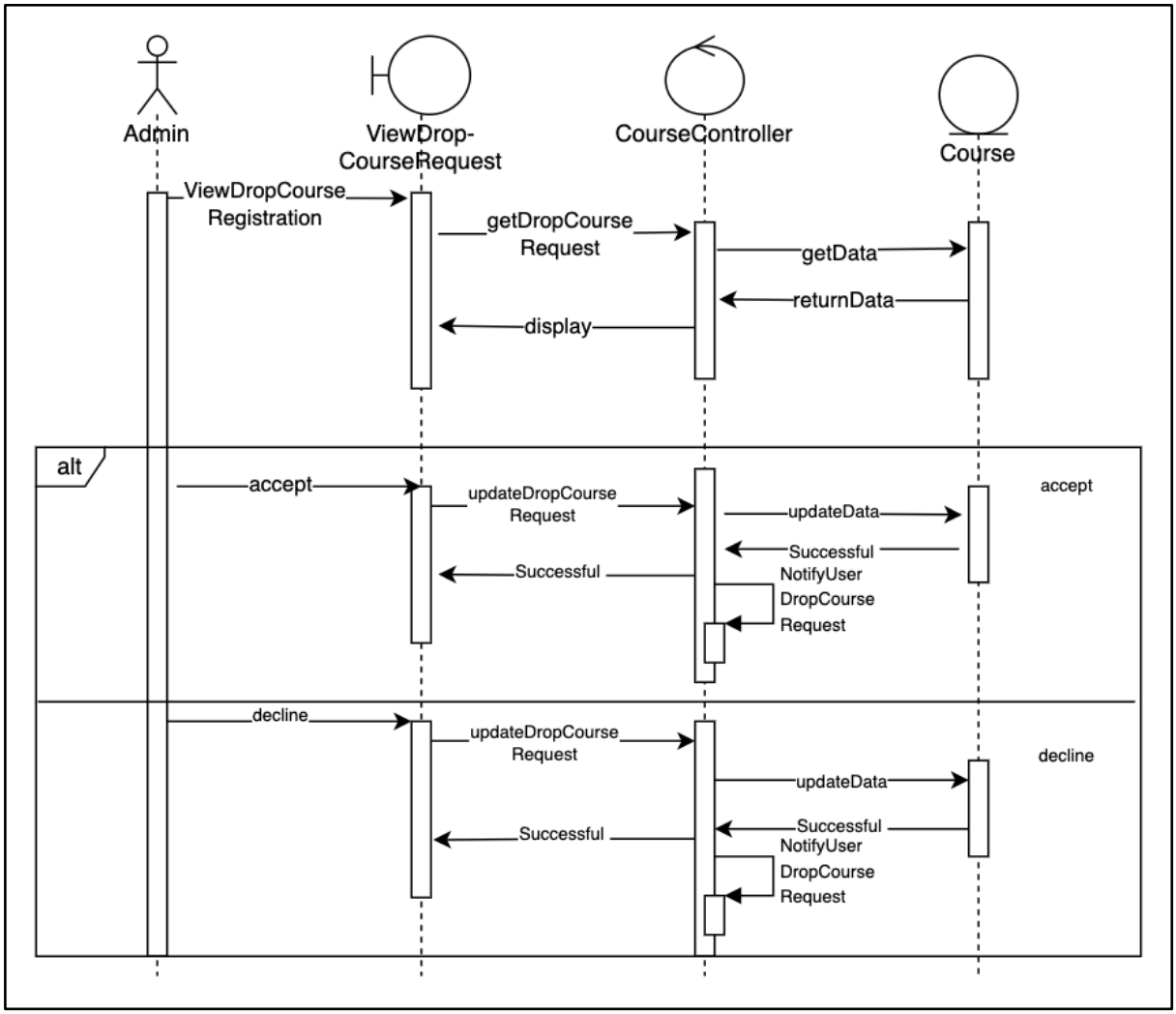


Figure 3.5: System Sequence Diagram of Drop Courses

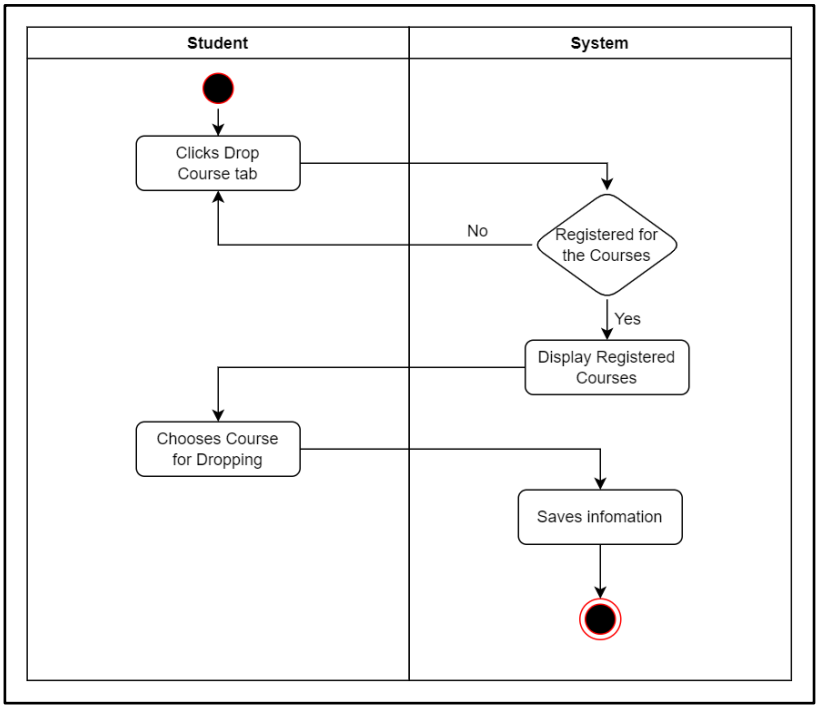


Figure 3.6: Activity Diagram of Drop Courses

3.2 **Performance Requirements**

This non-functional requirement can be related to several functional requirements, but one example could be "The system shall allow students to register for courses online." If the response time for user requests is slow, students may become frustrated and give up on registering for courses, thus the system would not be able to fulfill the functional requirement of allowing students to register for courses online.

3.3 **Design Constraints**

One constraint that an organization may impose on an online course registration system is that it must adhere to certain data security and privacy standards. For example, the system may be required to comply with regulations in order to protect personal information of students. This constraint would likely be related to non-functional requirements such as data encryption, access controls, and regular security testing.

3.4 **Software System Attributes**

1. Attractiveness: Users may request that the system has an attractive and visually pleasing design. They may also want the system to be easy to navigate and have a user-friendly interface.
2. Ease of Use: Users may request that the system is easy to use and understand, with clear and concise instructions and minimal input required. They may also want the system to be self-explanatory.
3. Speed: Users may request that the system is fast, with minimal wait times and quick response times.
4. Flexibility: Users may request that the system allows for flexibility when it comes to handling changes, like dropping or adding courses

1. INTRODUCTION

1.1 **Purpose**

This SDD describes design of the system and how it will interact with the user using various diagrams such as, use case diagram to show the functionality, package diagram that shows overall package of the system, class diagram of different modules inside the system, and the sequence diagrams of the same modules that have been chosen for the class diagrams. The SDD will also preview the system's functionality as a prototype.

1.2 **Scope**

The software product is focused on making the students' life easier in a way that they won't need to go all their way to the campus just for a simple signature and a stamp which can be easily solved with a program. That will allow them to choose the courses that they wish and the program will see if the student is approved or not. Once the program accepts the student, it will send it over to the managers of the students for them to double check just in case of any situation. The program will be on a mobile application when implemented perfectly. This program aims to satisfy the students' college life and the employees' jobs.The goal of this software product is to streamline the process of course registration for students by eliminating the need for them to physically go to campus for signatures and stamps. The program will allow students to select the courses they want and determine if they are approved or not. If approved, the program will then forward the registration to the relevant managers for final review. The plan is to make this program available on a mobile app for easy access. The ultimate objective of this program is to improve the student experience and make the job of college staff more efficient.

1.3 **Definitions, Acronyms and Abbreviation**

Definitions of all terms, acronyms and abbreviation used are to be defined here

QIU - Qaiwan International University

OCR - Online Course Registration

SDD - Software Design Document

DBMS - Database Management System

1.4 **References**

"What Is a Data Dictionary?" What Is a Data Dictionary? | UC Merced Library, https://library.ucmerced.edu/data-dictionaries#:~:text=A%20Data%20Dictionary%20is%20a,part%20of%20a%20research%20project.

1.5 **Overview**

The current version of the system allows students to catalog the courses that are available for registering, registering for courses, dropping courses, getting notified when courses are available for registering, and an admin panel to look after the system.

2. SYSTEM ARCHITECTURAL DESIGN

2.1 **Architecture Style and Rationale**

Model-View-Controller (MVC) is a software architectural pattern that separates an application into three main components: the model, the view, and the controller.

The model represents the data and the business logic of the application. It is responsible for maintaining the state of the application and performing operations on the data.

The view is responsible for rendering the user interface (UI) of the application. It presents the data to the user and allows them to interact with the UI.

The controller is responsible for handling the user's input and interacting with the model and the view. It receives requests from the user, processes them, and then updates the model and the view as needed.

In the context of an online course registration system, the model might include data about the courses being offered, the students who are registering for the courses, and the registration information. The view could be the web pages or forms that allow students to browse and select courses, and the controller could handle the submission of registration forms and the validation of registration information, including checking for availability and prerequisites. MVC is a popular architectural pattern because it separates the concerns of the application into distinct components, making it easier to develop and maintain. It also allows different team members to work on different parts of the application concurrently, which can improve development efficiency.

2.2 **Architecture Model**

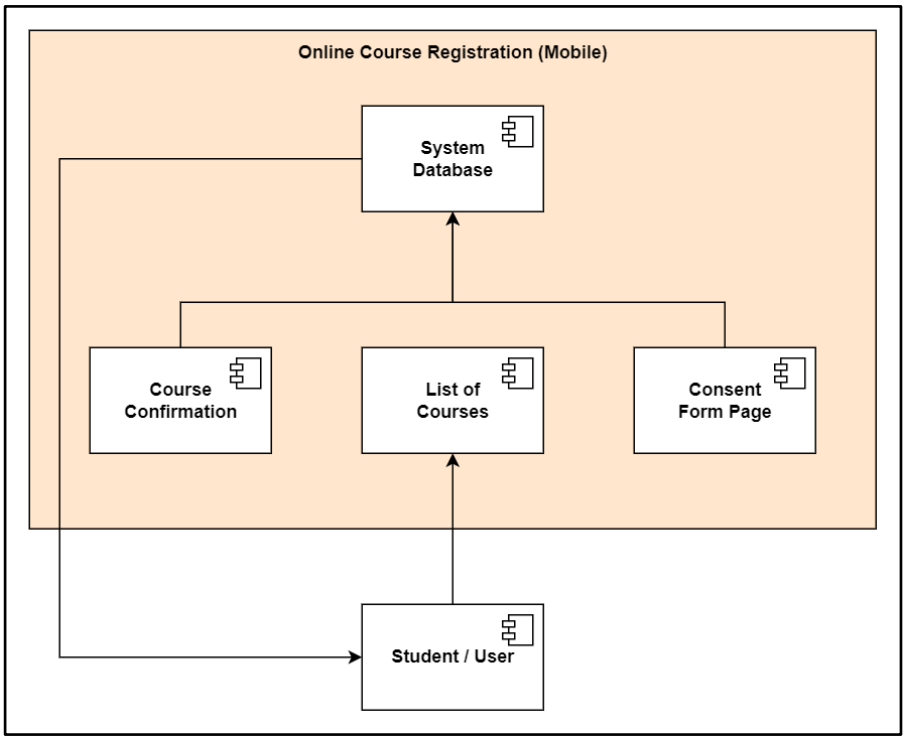


Figure 2.1: Component Model of OCR (Student)

2.3 **Use Case Diagram**

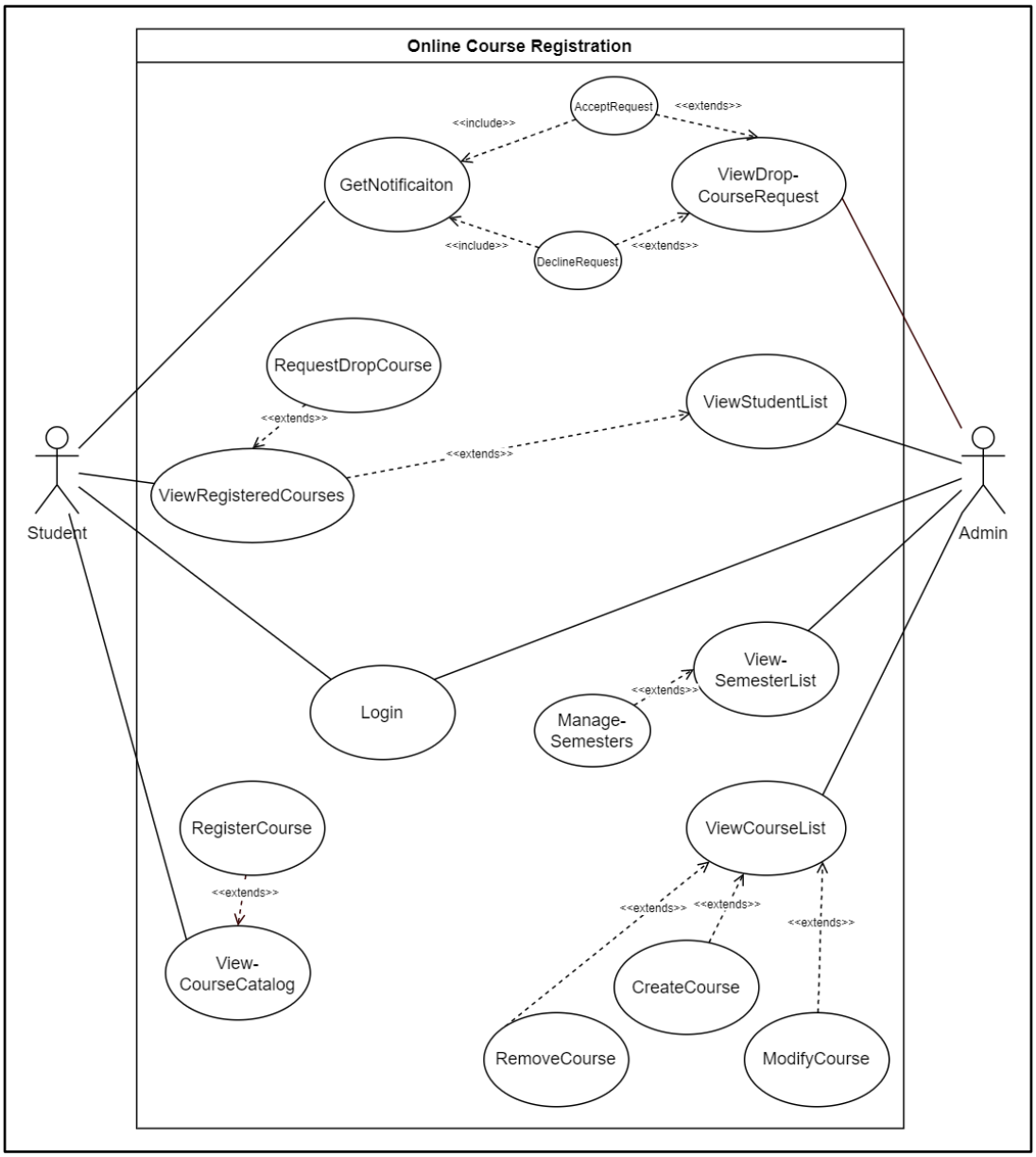


Figure 2.2: Use Case Diagram of OCR

3. DETAILED DESCRIPTION OF COMPONENTS

3.1 **Complete Package Diagram**

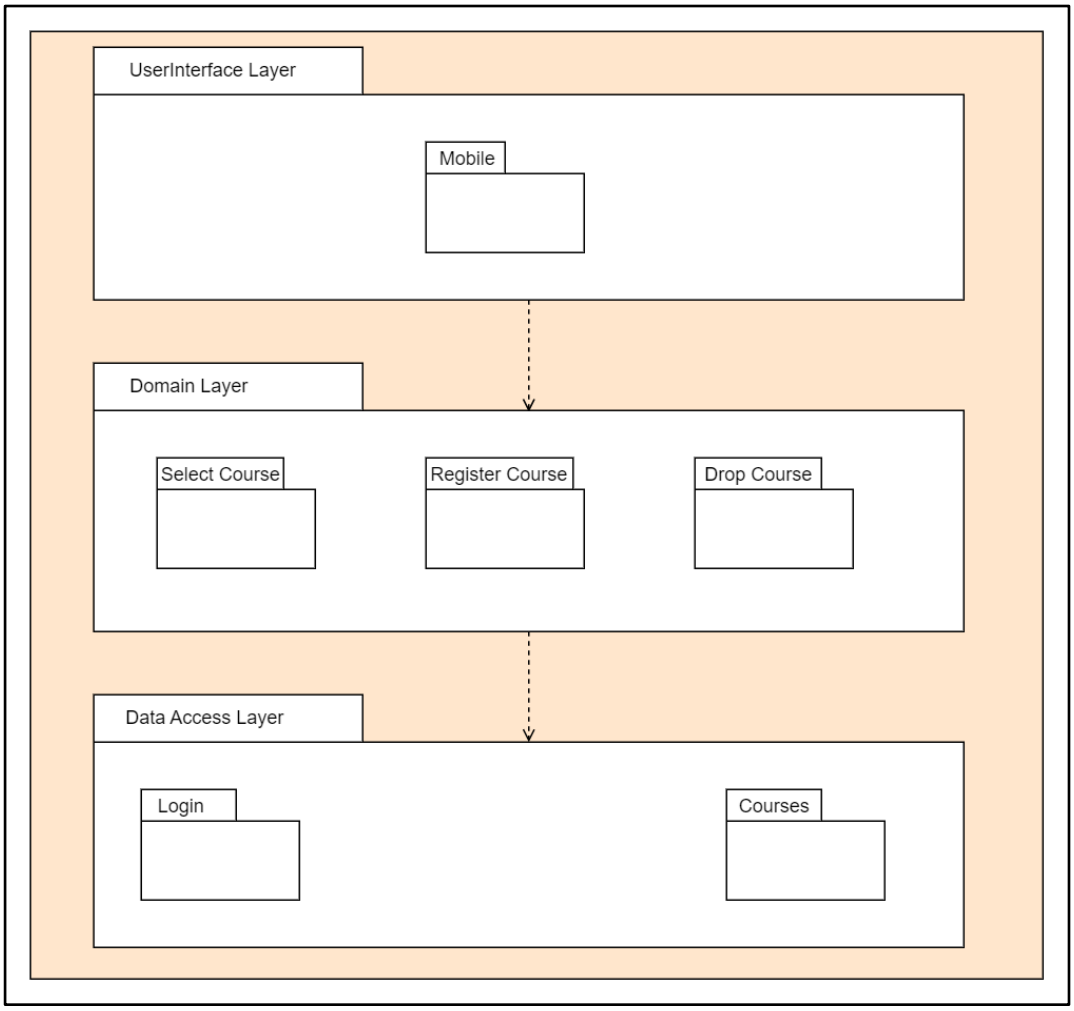


Figure 3.1: Subsystem of OCR

3.2 **Detailed Description**

An online course registration system mobile application is a software program that allows students to register for classes and manage their schedules using their mobile devices.

3.2.1 Subsystem User

3.2.1.1 Class Diagram

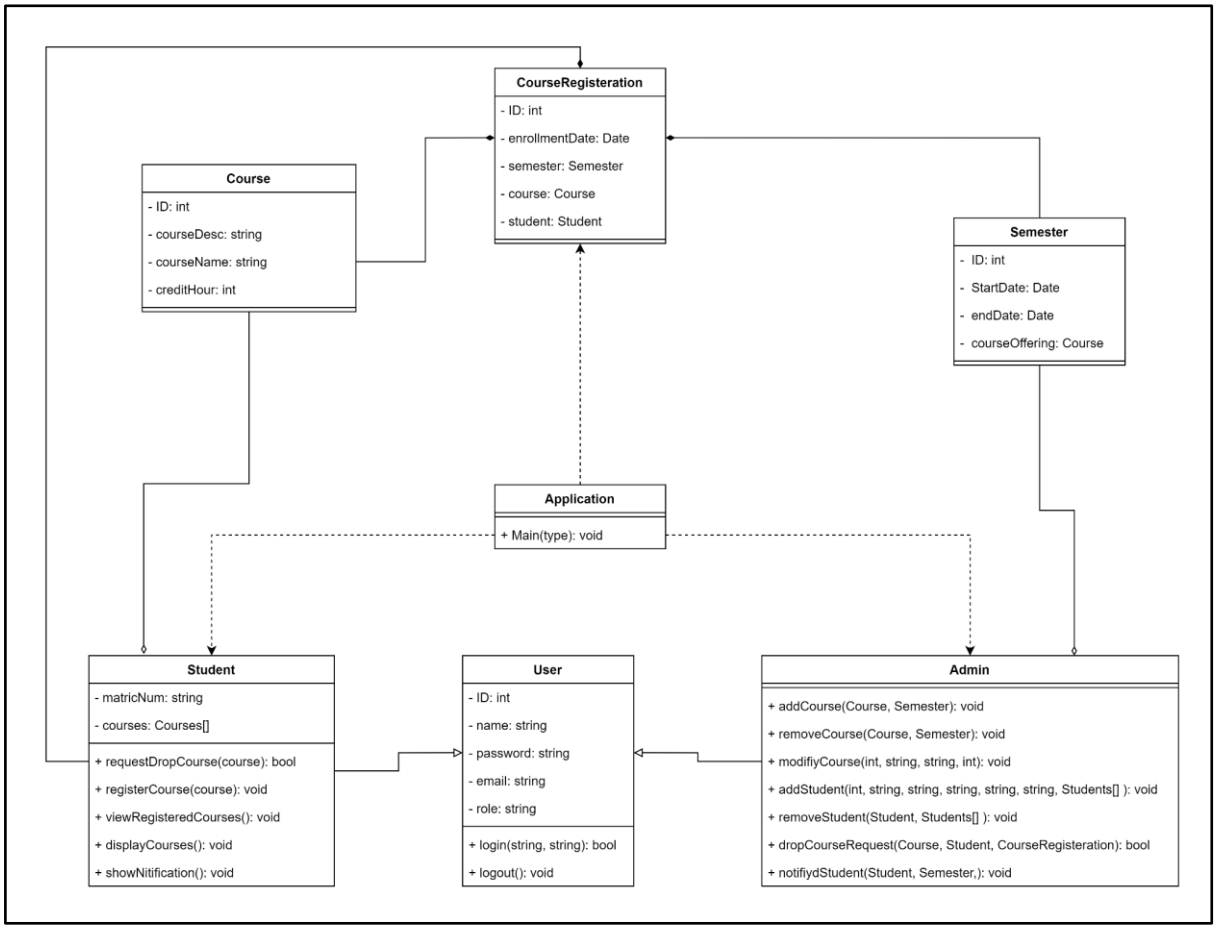


Figure 3.2: Class diagram for OCR (Student)

3.2.1.2 **Sequence Diagrams**

a) SD001: Sequence diagram for Register for Course

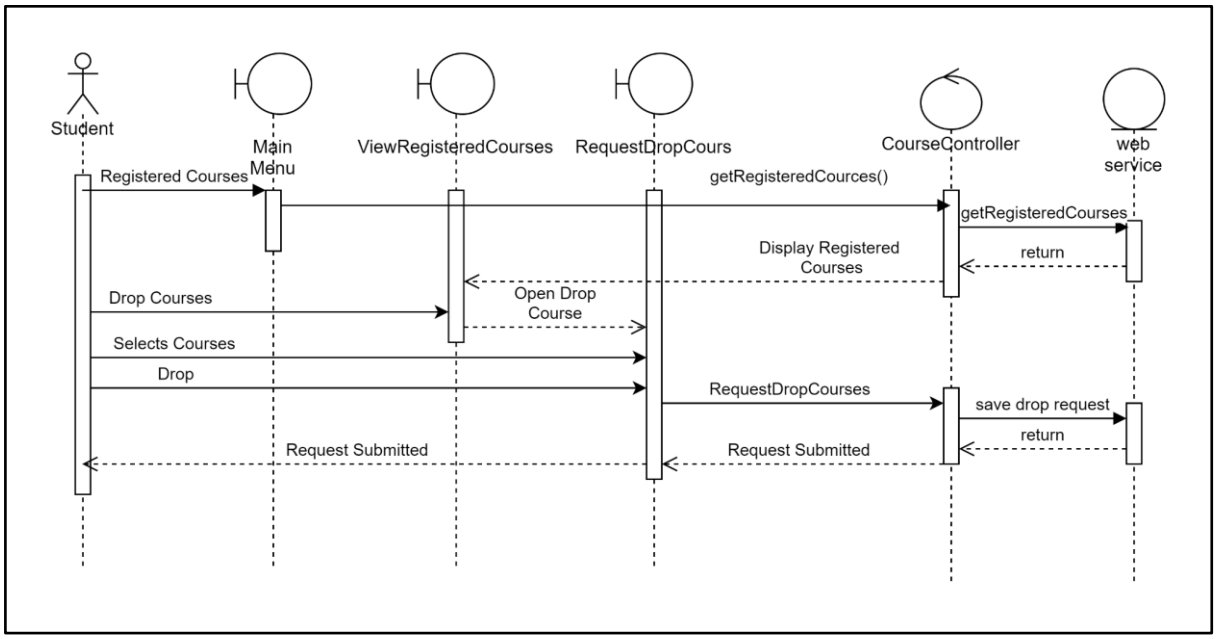


Figure 3.3: Sequence Diagram of <Register for Course>

b) SD002: Sequence diagram for View Registered Courses

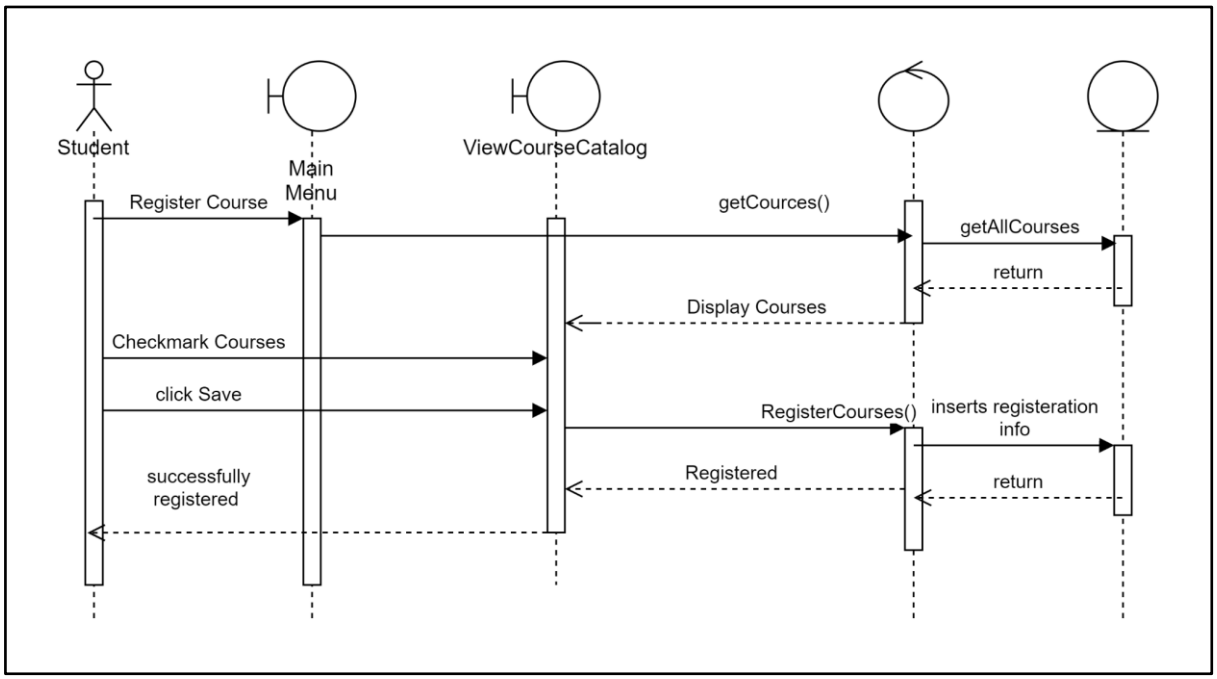


Figure 3.4: Sequence Diagram of <View Registered Courses>

c) SD003: Sequence diagram for Drop Course

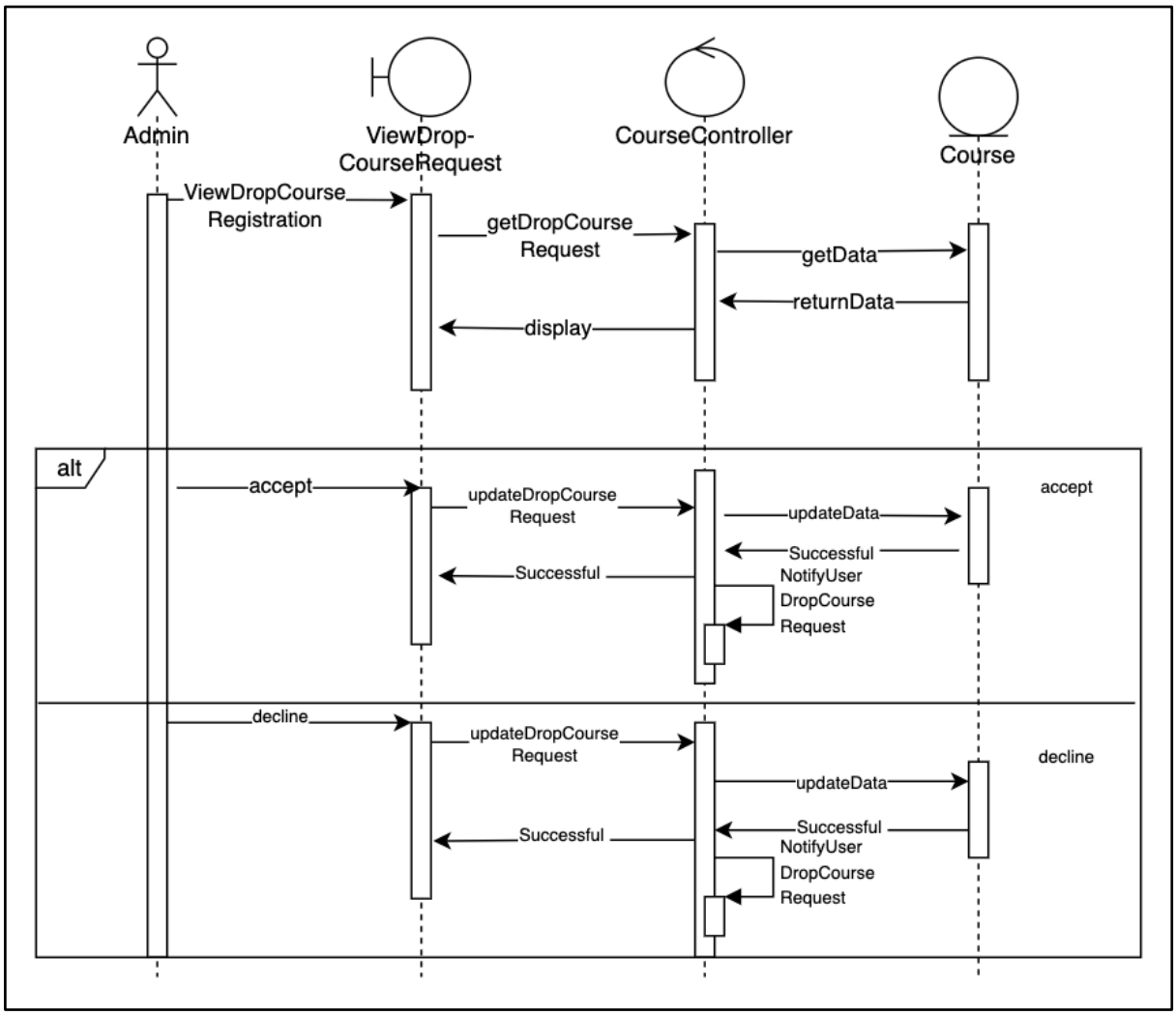


Figure 3.5: Sequence Diagram of <Drop Course>

4. DATA DESIGN

4.1 **Data Description**

An online course registration mobile application would have a variety of data structures that are used to store, process, and organize information. The information domain of the system includes data about the courses available for registration, the students who are registering for courses from the registration process.

1. Course Catalog: The course catalog would be stored in a database, with each course represented as a record. The fields in the record would include the course code, title, description, prerequisites, and credit hours.
2. Students: Student data would be stored in a separate database, with each student represented as a record. The fields in the record would include the student's name, ID, major, and contact information.

All these data structures would be managed by a database management system (DBMS) that would handle the storage, processing, and retrieval of data. The DBMS would also ensure that the data is consistent, secure, and available to authorized users.

5. USER INTERFACE DESIGN

5.1 **Overview of User Interface**

When a user opens the application, they will be presented with a login page where they will enter their login information which has been provided to them by Qaiwan International University. The login information will consist of their email address, which is their email. In case the user enters the login information incorrectly, the system will prompt them to try again. If the user continues to enter the login information incorrectly, they will need to contact an admin for assistance and provide proof of their identity. Upon successful login, the user will be taken to the homepage of the system where they will have access to various options such as registering for courses, viewing their registered courses and dropping courses. The option to register for courses is intended for students who are yet to register for classes, while the other two options are for students who have already registered. When the student clicks on the option to register for courses, they will be presented with a list of available classes that match their major and requirements. They can scroll through the list, read the course descriptions, and choose the classes they wish to register for. Once they have selected their classes for the semester, they will save their choices and be taken to a confirmation page where they will need to accept the terms and conditions before the registration is finalized.

5.2 **Screen Images**

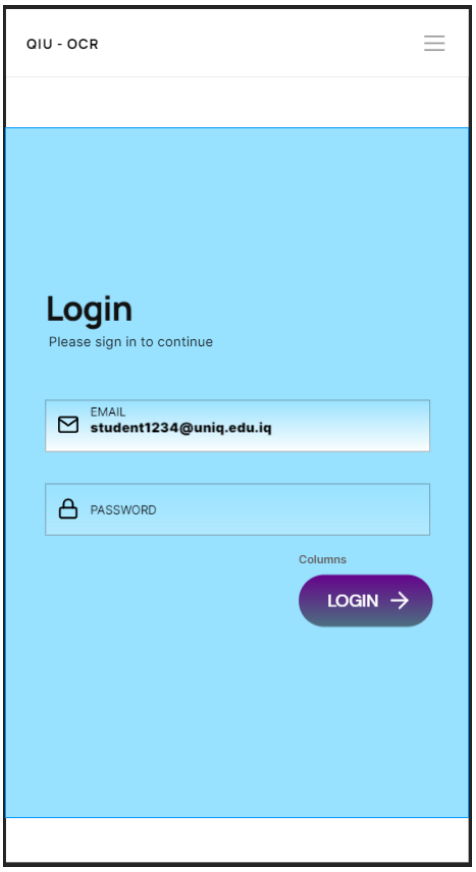


Figure 5.1: Login Page

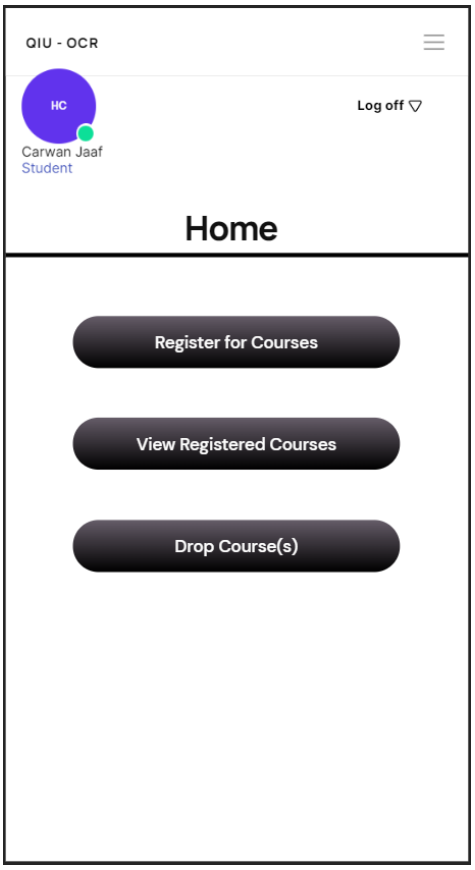


Figure 5.2: Home Page

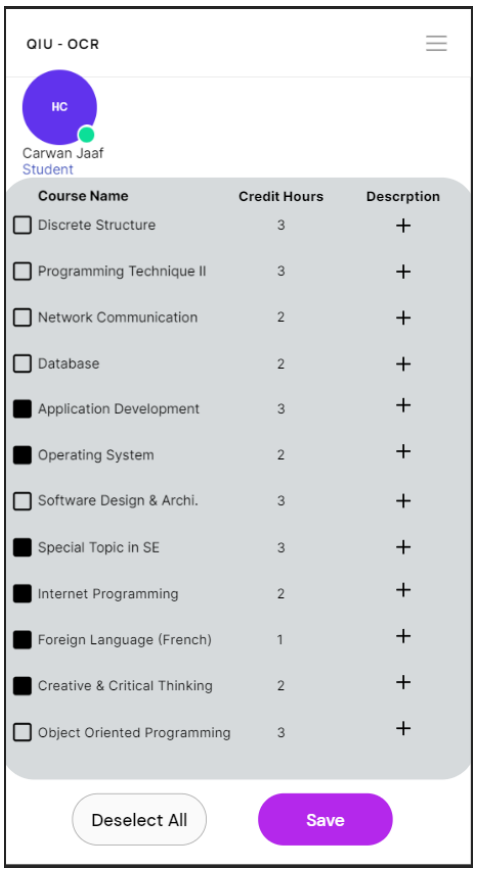


Figure 5.3: Register for Courses

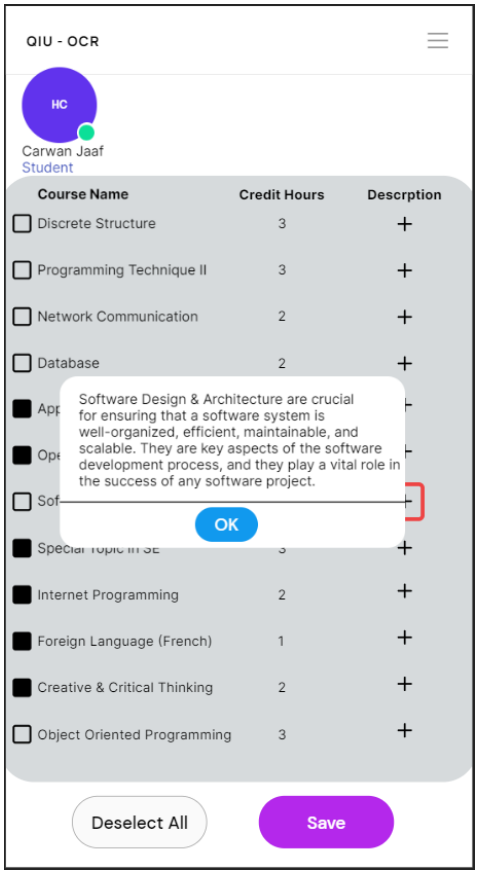


Figure 5.4: Course Description

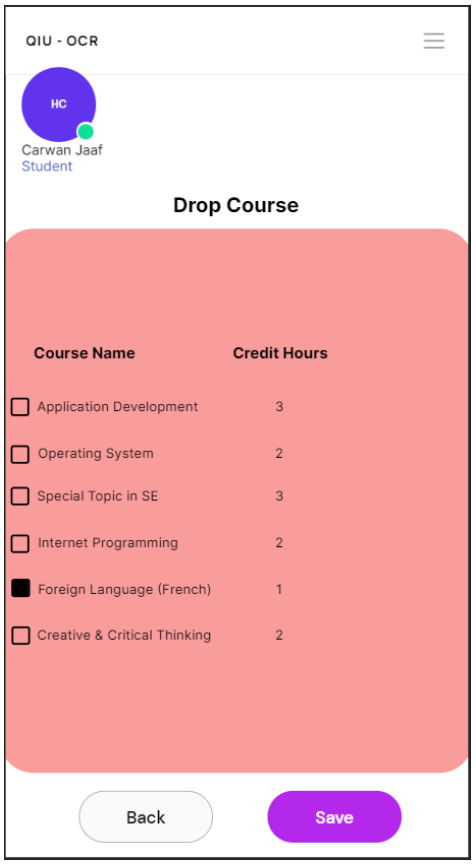


Figure 5.5: View Registered Courses **Figure 5.6: Drop Course(s)**

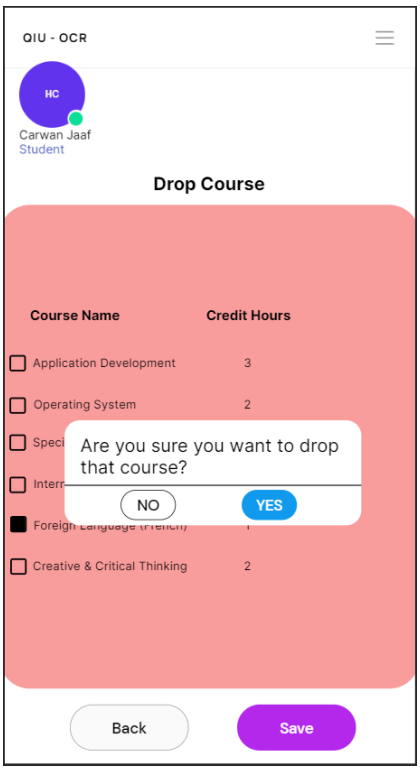


Figure 5.7: Confirmation for Dropping Course

6. REQUIREMENTS MATRIX

	ViewRegisteredCou	RequestDropCourse	ViewCourseCatalog	ViewDropCourseReq	ViewStudentList	ViewCourseList	EditCourse

	r s e s			u e s t			
UC002, SD002			X				
UC003, SD001	X						
UC004, SD001	X	X					
UC005, SD005						X	X
UC006, SD005						X	X
UC007, SD005						X	

1. INTRODUCTION

1.1 **Purpose**

The purpose of this software testing documentation (STD) for an online course registration system is to provide the necessary information about the testing activities that have been conducted on the system. It includes details such as test descriptions, test results, and any issues that were identified and resolved during testing. The intended audience for this documentation would be the development team, and any stakeholders involved in the testing and development of the online course registration system.

1.2 **Scope**

The software product is focused on making the students' life easier in a way that they won't need to go all their way to the campus just for a simple signature and a stamp which can be easily solved with a program. That will allow them to choose the courses that they wish and the program will see if the student is approved or not. Once the program accepts the student, it will send it over to the managers of the students for them to double check just in case of any situation. The program will be on a mobile application when implemented perfectly. This program aims to satisfy the students' college life and the employees' jobs.The goal of this software product is to streamline the process of course registration for students by eliminating the need for them to physically go to campus for signatures and stamps. The program will allow students to select the courses they want and determine if they are approved or not. If approved, the program will then forward the registration to the relevant managers for final review. The plan is to make this program available on a mobile app for easy access. The ultimate objective of this program is to improve the student experience and make the job of college staff more efficient.

1.3 **Definitions, Acronyms and Abbreviation**

STD - Software Testing Documentation

QIU - Qaiwan International University

OCR - Online Course Registration

1.4 **System Overview**

SOFTWARE TESTING DOCUMENTATION IS TYPICALLY ORGANIZED INTO SEVERAL SECTIONS THAT PROVIDE A DETAILED OVERVIEW OF THE TESTING PROCESS. IT BEGINS WITH A TEST PLAN, WHICH OUTLINES THE OVERALL APPROACH TO TESTING, INCLUDING THE SCOPE OF TESTING, TESTING OBJECTIVES, AND THE RESOURCES REQUIRED.

TEST CASES, DATA AND EXPECTED RESULTS

1.5 **Test TC001 for Module <Name of Module1>: <Name of Use Case (UC001)>**

This test contains the following test cases:

2.2.1.UC001: Use Case <login>

Use case: <Login>
ID: UC001
Actors:Admin, Student
Preconditions: System must be connected to the network.
Flow of events: 1. Users enter into the login page. 2. System request for email and password 3. User insert the email and password 4. If the email and password are correct, the system verifies the user account. 5. System display the page based on user type: Student or Admin
Exception Flow: 1. If the email or password is wrong, flow of events exception 2 is performed.
Postconditions: After a successful login the system will view the page based on user type.

Table 2.1: Use Case Description for <login>

2.2.2.UC002: Use Case <Register Courses>

Use case: <Register Courses>
ID: UC002
Actors: students
Preconditions: The student has logged in to the system and the student must meet certain requirements, such as being admitted to the institution, being in good academic standing, and having completed certain prerequisites.
Flow of events: 1. Click View Course Catalog 2. Click Register for Courses 3. Checkmark the courses that they want 4. Click Save button to register for the marked courses 5. Course/Courses Registered
Postcondition: The prospective student has successfully finished filling the course registering page and can proceed.

Table 2.2: Use Case Description for <Register Courses>

2.2.3. UC003: Use Case <View Registered Courses>

Use case: <View Registered Courses>
ID: UC003
Actors: Student
Preconditions: The student must be logged in and have already registered for courses in order for the function to work.
Flow of events: 1. Click View Registered Courses. 2. View Registered Courses.
Postconditions: The student will be shown a list of courses that they have registered for.

Table 2.3: Use Case Description for <View Registered Courses>

2.2.4.UC004: Use Case <Request Drop Course>

Use case: <Request Drop Course>
ID: UC004
Actors: Student, Admin
Preconditions: The student must be logged in and ask permission on dropping the course incase of any downfall in the long-run of the student.
Flow of events: 1. Click View Registered Courses. 2. Click Drop Course Request. 3. Choose the course that wants to be dropped. 4. Click Drop.
Postconditions: The student has requested to drop selected courses.

Table 2.4: Use Case Description for <Request Drop Course>

2.2.6.UC005 Use Case <Create New Course>

Use case: <Create New Course>
ID: UC005
Actors: Admin
Preconditions: The admin needs to be logged in and gets permission from the department administrators to add a new course for the students department(s).
Flow of events: 1. Click view course list. 2. Click add course 3. Enters course information 4. Click save
Postconditions: The system will have a new course in the course catalog and it will notify the student with an email notification.

Table 2.5: Use Case Description for <Create New Course>

2.2.7 UC006: Use Case <Modify Course>

Use case: <Modify Course>
ID: UC006
Actors: Admin
Preconditions: The admin must be logged in and needs to get permission from the department administrators to modify a pre-existing course.
Flow of events: 1. Click view course list 2. Select a course and click Modify Course 3. Update desired information 4. Click save
Postconditions: The course will be modified to the admin and department administrators liking.

Table 2.6: Use Case Description for <Modify Course>

2.2.8 UC007: Use Case <Remove Course>

Use case: <Remove Course>
ID: UC007
Actors: Admin
Preconditions: The admin needs to be logged in and needs to get permission from the department administrators to remove an existing course.
Flow of events: 1. Click view course list 2. Select a course and click Delete Course 3. Confirm Delete
Postconditions: The system will have one less course in its course catalog.

Table 2.7: Use Case Description for <Remove Course>

2. TEST APPROACH ANALYSIS

2.1 **Data Description**

The major data or systems entities are stored into a relational database named as… comprises 4 entities.

Entity Name	**Description**
student	This entity contains prospective student information, such as matric number
user	This entity contains prospective users information, such as name, email,role, and password
enrolled_courses	This entity links student to courses
course	This entity contains courses information, such as description, name,and credit hour
message	This entity contains message information
notifications	This entity links message to student
semester	This entity contains semester information, such as stating date and ending date and available courses
available_courses	This entity links semester to course

2.2 **Data Dictionary**

Entity: <user**>**

Attribute Name	**Type**	**Description**
Name	string	name of user

Entity: <user**>**

Attribute Name	**Type**	**Description**
Email	string	email of user’s account

Entity: <user**>**

Attribute Name	Type	Description
ID	int	id of user's account

Entity: <user**>**

Attribute Name	Type	Description
Password	string	password of user's account

Entity: <student**>**

Attribute Name	Type	Description
ID	int	id of student account

Entity: <student**>**

Attribute Name	Type	Description
matric	string	matric number of student

Entity: <course**>**

Attribute Name	Type	Description
ID	int	ID of course

Entity: <course**>**

Attribute Name	Type	Description
Name	string	name of the course

Entity: <course**>**

Attribute Name	Type	Description
description	string	description of the course

Entity: <course**>**

Attribute Name	Type	Description
credit	int	credit hour of the course

Entity: <enrolled_courses**>**

Attribute Name	Type	Description
ID	int	enrollment ID

Entity: <enrolled_courses**>**

Attribute Name	Type	Description
student_ID	int	it is a foreign key it stores student ID number

Entity: <enrolled_courses**>**

Attribute Name	Type	Description
course_ID	int	it is a foreign key it stores course ID number

Entity: <message**>**

Attribute Name	Type	Description
ID	int	ID of the message

Entity: <message**>**

Attribute Name	Type	Description
message	string	content of the message

Entity: <notifications**>**

Attribute Name	Type	Description
ID	int	id of notification

Entity: <notification**>**

Attribute Name	Type	Description
message_ID	int	it is a foreign key it stores message ID number

Entity: <notification**>**

Attribute Name	Type	Description
student_ID	int	it is a foreign key it stores student ID number

Entity: <semester**>**

Attribute Name	Type	Description
ID	int	ID of the semester

Entity: <semester**>**

Attribute Name	Type	Description
starting_date	Date	the date of when the semester starts

Entity: <semester**>**

Attribute Name	Type	Description
starting_date	Date	the date of when the semester starts

Entity: <semester**>**

Attribute Name	Type	Description
end_date	Date	the date of when the semester ends

Entity: <available_courses**>**

Attribute Name	Type	Description
ID	int	ID of the available courses

Entity: <available_courses**>**

Attribute Name	Type	Description
course_ID	int	it is a foreign key it stores course ID number

Entity: <available_courses**>**

Attribute Name	Type	Description
semester_ID	int	it is a foreign key it stores semester ID number

ADDITIONAL MATERIAL

Table-based Design

Condition/Action	TC001	TC002	TC003	TC004
QIU staff approve the application	T	T	F	F
Faculty admissions approve the application	T	F	T	F
The application will be accepted and offered	x		x	
The application will get a rejection letter				x

Table 9.3: Limited Entry Decision Table

APPENDIX A. TRACEABILITY MATRIX

	ViewRegisteredCourses	RequestDropCourse	ViewCourseCatalog	ViewDropCourseRequest	ViewStudentList	ViewCourseList	EditCourse
UC002, SD002			X				
UC003, SD001	X						
UC004, SD001	X	X					
UC005, SD005						X	X
UC006, SD005						X	X
UC007, SD005						X	

QAIWAN INTERNATIONAL UNIVERSITY	QAIWAN INTERNATIONAL UNIVERSITY	Form No.:LIB003 Edition: 01 Effective Date: 01/ 06/ 2024 Page (s): 03
	FINAL THESIS SUBMISSION FORM	

Section 1 (To be completed by Student)
(Please tick (√) where applicable)

Library
Qaiwan International University

Submission of Final Copies of Thesis

I Carwan Jaaf (name) hereby submit:
the final copies of my thesis for the degree of [✓] Bachelor / [ ] Master / [ ] Doctor of Philosophy

The title of the thesis is:

Qaiwan International University
Course registration

Hence, I declare that:

[✓] i) My thesis has been reviewed by my Main Supervisor / HoD / Dean / Faculty

[✓] ii) I had submitted the CD of my softcopy of the thesis (the contents are similar with the hardcopy of the thesis) in a single PDF file.

[✓] iii) I had submitted the Degree Scroll Information Form.

Student Name: Carwan Jaaf

Department: S.E

Signature:

Date: 27/8/2024

ENDORSEMENT BY MAIN SUPERVISOR

Section 2 (To be completed by Main Supervisor)

Supervisor's Name: W. Kar Zan

Department: S.I.E

I have examined the thesis of Mr. /Mrs. /Ms.: Carman Jacif

I hereby confirm that all corrections and amendments made to the thesis have been rectified by the candidate.

Dr Shib

(Main Supervisor's Signature & Stamp)

27/8/2024

(Date)

ENDORSEMENT BY Librarian

Section 3 (To be completed by Librarian)

Librarian's Name: Dilkhwan Othman

☑ Approved ☐ Not Approved

I hereby confirm that all corrections and amendments made to the thesis have been rectified by the candidate.

(Librarian's Signature & Stamp)

27/8/2024

(Date)

STUDENT'S DECLARATION

Section 4 (To be completed by Student)

Student Thesis's Copyright and University's Intellectual Property

"The copyright to a thesis belongs to the student. However, as a condition of being awarded the degree, the student hereby grants to the University, a free, ongoing, non-exclusive right to use the relevant work and/or thesis for the University's teaching, research and promotional purposes as well as free and the non- exclusive right to retain, reproduce, display and distribute a limited number of copies of the thesis, together with the right to require its publication for further research and archival use."

I declare that the contents presented in this thesis are my own which was done at Qaiwan International University unless stated otherwise. The thesis has not been previously submitted for any other degree.

I also declare that my thesis has been reviewed by the Main Supervisor / HoD / Dean / Faculty the comments are as stated in Section 2 (page 2).

Name of Student: Carwan Saaf

Signature of Student: .....

IC. No. / Passport No.: .....

Date: 27/8/2024