Course Enrollment

Window Application

April 9th, 2022

PoKai Huang 000429977

# **Contents**

[Contents 2](#_Toc100345085)

[1. Introduction 4](#_Toc100345086)

[1.1 Project Overview 4](#_Toc100345087)

[1.2 Project Goals and Objectives 4](#_Toc100345088)

[2. General Design Constraints 4](#_Toc100345089)

[2.1 Software Environment 4](#_Toc100345090)

[2.2 User Need 4](#_Toc100345091)

[3. Non-functional Requirement 5](#_Toc100345092)

[3.1 Operational Requirements 5](#_Toc100345093)

[3.2 Performance Requirements 5](#_Toc100345094)

[3.3 Security 5](#_Toc100345095)

[4. Functional Requirement 6](#_Toc100345096)

[4.1 Application Overview 6](#_Toc100345097)

[5. Project Plan 7](#_Toc100345098)

[5.1 Project Plan Summary 7](#_Toc100345099)

[5.2 Project Schedule 8](#_Toc100345100)

[5.3 Project Tools for Application Development 9](#_Toc100345101)

[6. Project Design 10](#_Toc100345102)

[6.1 Application Architecture 10](#_Toc100345103)

[6.2 Data Storage 11](#_Toc100345104)

[6.3 Class Diagram 12](#_Toc100345105)

[6.4 Active Diagrams 13](#_Toc100345106)

[6.5 Wireframes 15](#_Toc100345107)

[7. Test Result 17](#_Toc100345108)

[7.1 User Testing 17](#_Toc100345109)

[7.1.1 Get All Student 17](#_Toc100345110)

[7.1.2 Get All Courses Information 18](#_Toc100345111)

[7.1.3 Get Prerequisites Courses for Student 19](#_Toc100345112)

[7.1.4 Get Student Uncompleted Courses 20](#_Toc100345113)

[7.1.5 Get Non-Prerequisite Courses for Student 21](#_Toc100345114)

[8. User Guide 22](#_Toc100345115)

[8.1 Installation 22](#_Toc100345116)

[9. Conclusion 23](#_Toc100345117)

[9.1 Lessons Learned 23](#_Toc100345118)

[9.2 Known Issues 23](#_Toc100345119)

[9.3 Future Improvements 23](#_Toc100345120)

[10. References 24](#_Toc100345121)

# 1. Introduction

## 1.1 Project Overview

This document presents a detail description of the Course Enrollment WinApp. It will explain the purpose and features of the application, the interfaces and what this app will do, the constraints under which it must operate and how Course Enrollment WinApp will react to external stimuli. This document is intended for both the stakeholders and the developers of the Course Enrollment WinApp and will be proposed to our course instructor George for its approval.

## 1.2 Project Goals and Objectives

In this project, our team want to create a windows application named Course Enrollment. Course Enrollment will be designed to helps users to manage their class, course, and view their validity of course next term. Users can easily manage the student and courses. Moreover, user can easily view how many course the student can take.

There are other project management applications in the market, but most of them demand a heavy monetary subscription. We want to make our project management application free and friendly to our users.

# 2. General Design Constraints

## 2.1 Software Environment

We will use the Visual Studio 2019 to develop the application for this project. We aim to design our Course Enrollment WinApp by using WinUI and run-on Window device.

## 2.2 User Need

Users need an application that can add students, courses and manage student to add the proper course. For courses, some courses have prerequisites to sign up. For student, some student needs different courses to choose after first semester.

# 3. Non-functional Requirement

## 3.1 Operational Requirements

The application will run simple, there won’t be any login just launch the app then will start it.

This application will need internet access at minimum 3G speed to save the data.

Overall, this app will be easy as most users are already familiar with management systems.

## 3.2 Performance Requirements

* Users will be able to access their account 99% of the time without failure.
* For better use of this application, it will be recommended that users should try to use it solo at first (by assigning tasks to themselves) than working in a group.
* The application is user friendly and can be used by different aged groups for different tasks.

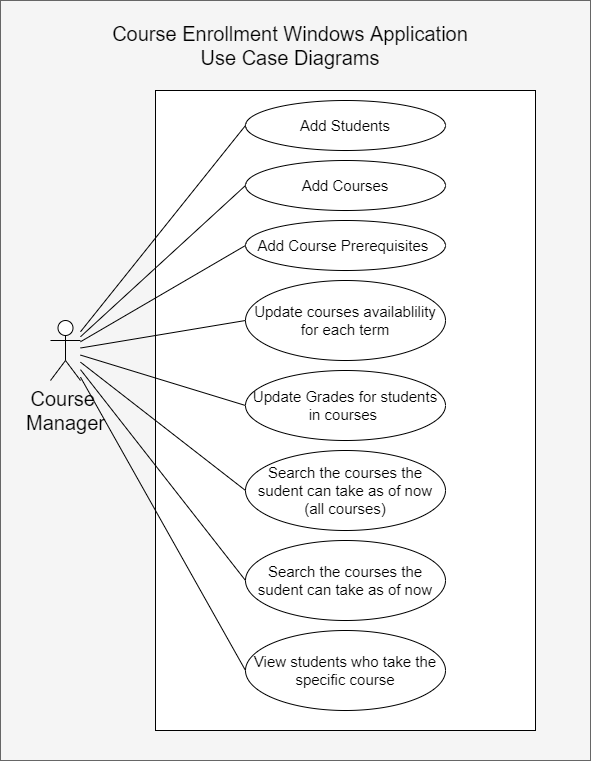
## 3.3 Security

* All data generated from the application, such as user’s projects and tasks should be stored in the azure backend only
* Only application developers and maintainers should have access to the backend.
* Unauthorized data export from the backend is prohibited.

# 4. Functional Requirement

## 4.1 Application Overview

* User can add student (Done in Microsoft Azure database)
* User can add course (Done in Microsoft Azure database)
* User can find student by name
* User can manage student’s scores for each course (Done in Microsoft Azure database)
* User can manage course’s requirement (Done in Microsoft Azure database)
* User can add courses prerequisite (Done in Microsoft Azure database)
* User can view the course suggestion feature for a given student and a given year/term
* Output of the system

Figure 4.1 Use Case Diagram

# 5. Project Plan

## 5.1 Project Plan Summary

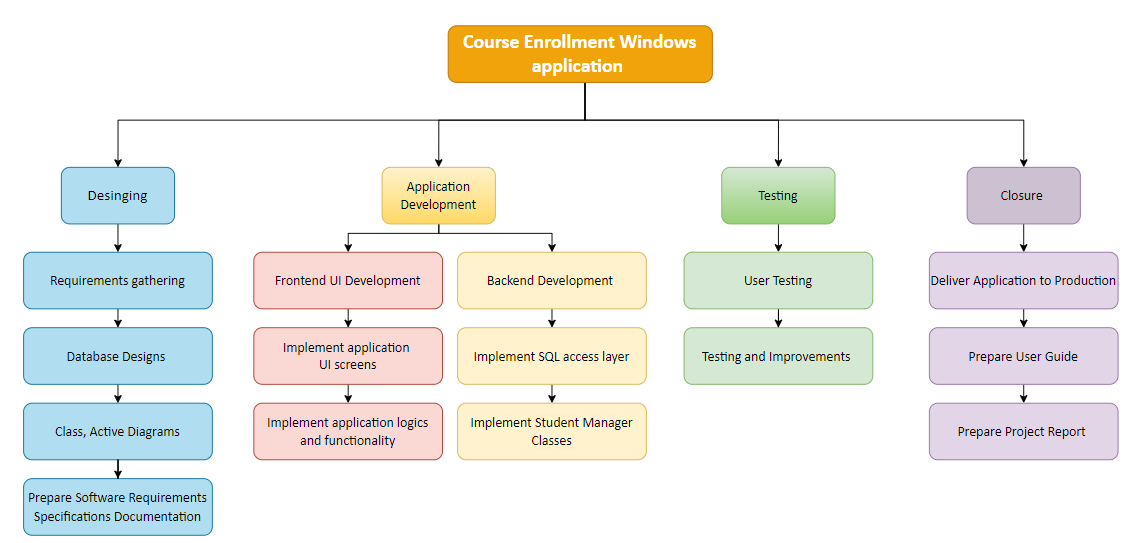
 Our project was divided into 4 development phases/milestones: the **Designing**, **Implementing**, **Testing** and **Closing** phases. The above figure is the working breakdown structure diagram.

Figure 5.1 WBS Diagram

In the Designing phase (from Feb 12 to Mar 8), our team has completed the following works: **Requirements gathering**, **Database Design**, **Class Diagram** & **Active Diagram Design** and **Software Requirement Specifications (SRS).**

In the Implementation phase (from Mar 8 to Mar 26), our team aimed to first complete the backend development, such as implementing the Microsoft Azure Database access layer, logic, and functionalities. After it was completed mostly, we moved on to implement the Student Manager Class, such as data storage and searching data. We did some of the UI design parts with Win UI 3. However, due to the time limit, we focus more on the backend implementation.

In the Testing phase (from Mar 27 to Apr 7), we conducted user testing and collected feedback from test subjects, then fixed the error we found.

In the Closure phase (Apr 9), the team completed our first version of the Course Enrollment Windows application on Apr 9 and the user guide on the same day.

## 5.2 Project Schedule

Please refer to APPENDIX 1.1 for our project Gantt chart. It illustrates our project development phases/milestones, planning schedule, duration, and date of completion of each task. You may also find a downloadable version of our Gantt chart at the following link:

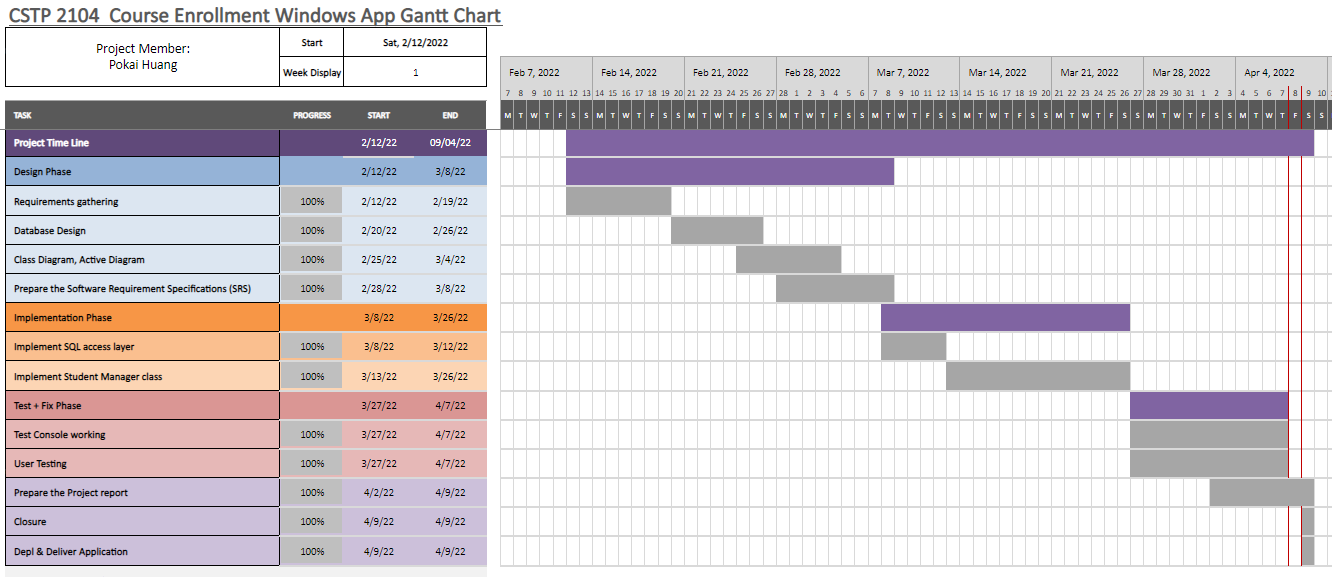
<https://1drv.ms/x/s!AgSWbnXU4oIwtTw6WVSkRmwBQhNm?e=diwKhf>

Figure 5.2 Project Planning Gantt Chart

## 5.3 Project Tools for Application Development

In this project, we used Visual Studio 2019 as our primary code editor. Visual Studio 2019 is developed by Microsoft. It is a free and open-source code editor. It runs on cross platforms like Windows, Mac, and Linux.

We also used Microsoft Azure and GitHub as support development tools. Microsoft Azure is a public cloud computing platform—with solutions including Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) that can be used for services such as analytics, virtual computing, storage, networking, and much more. For this project, it helped our project for setting up database and manage data structures.

As for versioning control tool, our team used GitHub to deposit our application source codes.

Below is a summary specification of the Course Enrollment application.

|  |  |
| --- | --- |
| **Hardware Platform** | Windows PC |
| **Software Platform** | Windows 11 |
| **Program Language Used** | C# |
| **Development Tools** | Visual Studio 2019  Microsoft Azure  GitHub |

*Table 5.1 Application summary*

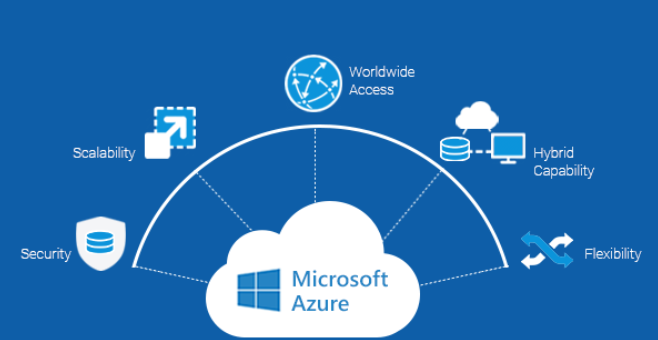
# 6. Project Design

## 6.1 Application Architecture

Figure 6.1 Architecture Design Use Case Diagram

Figure 6.1 shows the system architecture of our application. Users interact with the application via the mobile frontend, from which the users can manage their courses and students. The frontend communicates with an Azure SQL database backend.

## 6.2 Data Storage

 Since we are using Azure SQL database, it is a fully managed platform as a service (PaaS) database engine that handles most of the database management functions such as upgrading, patching, backups, and monitoring without user involvement.

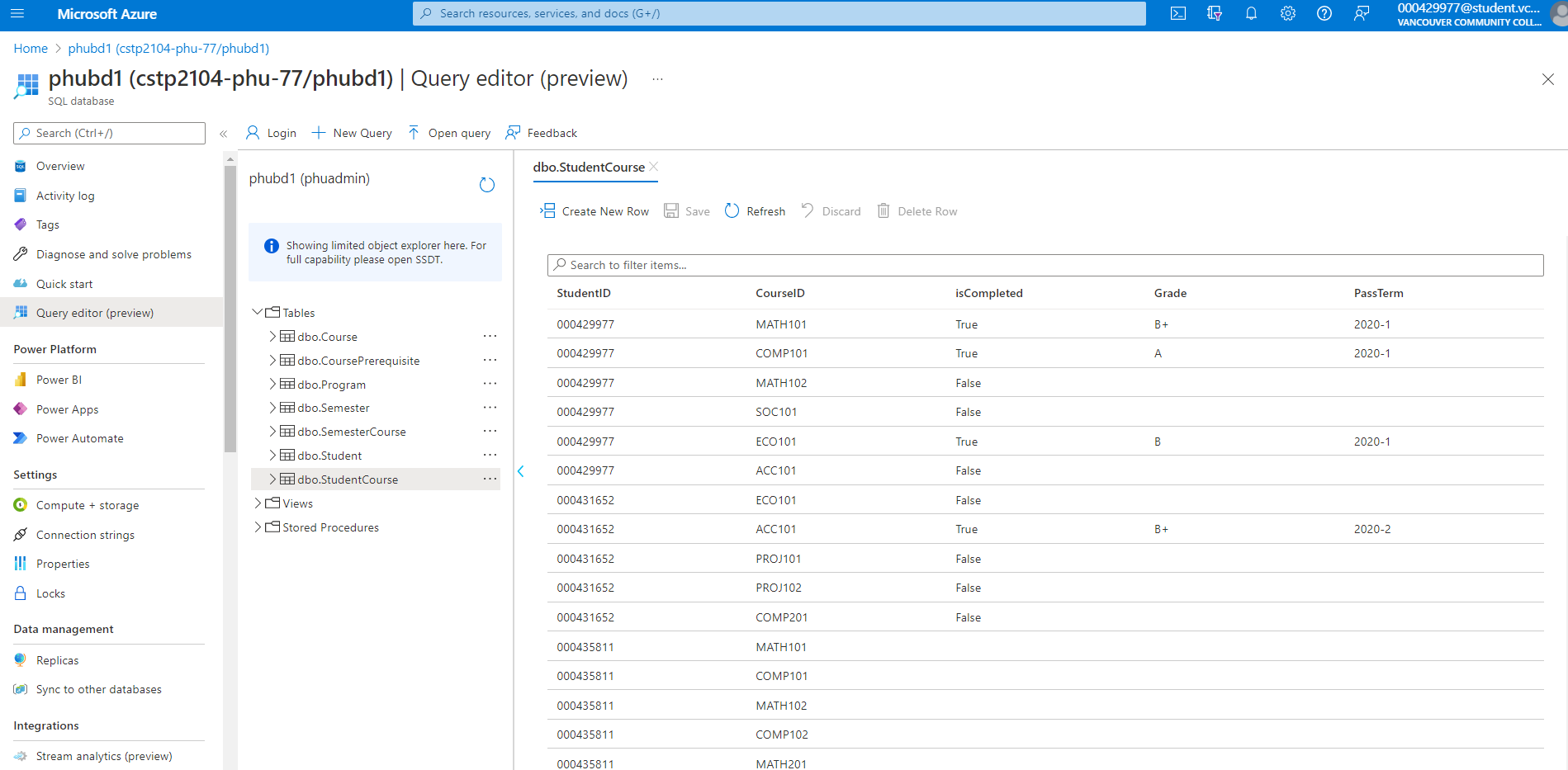
Figure 6.2 Microsoft Azure - 1

Figure 6.2 Microsoft Azure - 2

## 6.3 Class Diagram

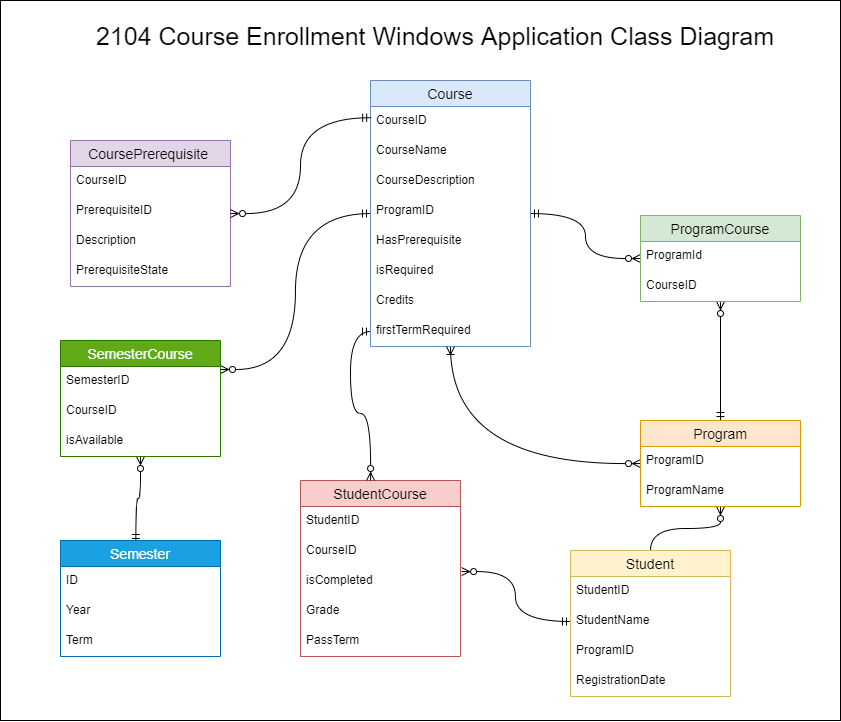


Figure 6.3 Class Diagram

Figure 6.3 shows a class diagram of the Student, Program, and Course and the Semester and more classes.

The Program class contain the program ID and Name then has relationships with ProgramCourse class and Student Class.

The Course class contains all courses information such as course ID, Name, Descriptions, Credits and so on. It also includes Course Prerequisites class to manage the course which need another course to complete first. Moreover, it also connects the student class with StudentCourse class to make sure each student is completed the course and grad and pass term.

The Semester class contain the ID, Year, Terms and has relationships with SemesterCourse to let user has an ability to manage the courses availability.

## 6.4 Active Diagrams

Due to the time limit, we could not go over all works on the application. However, we still trying give all active diagrams as our plans.

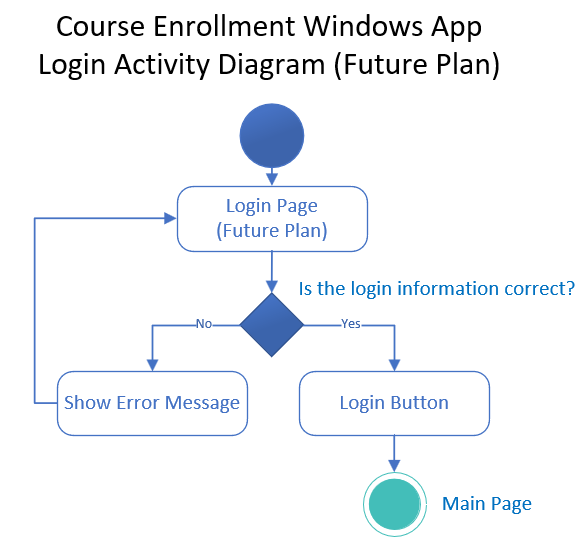
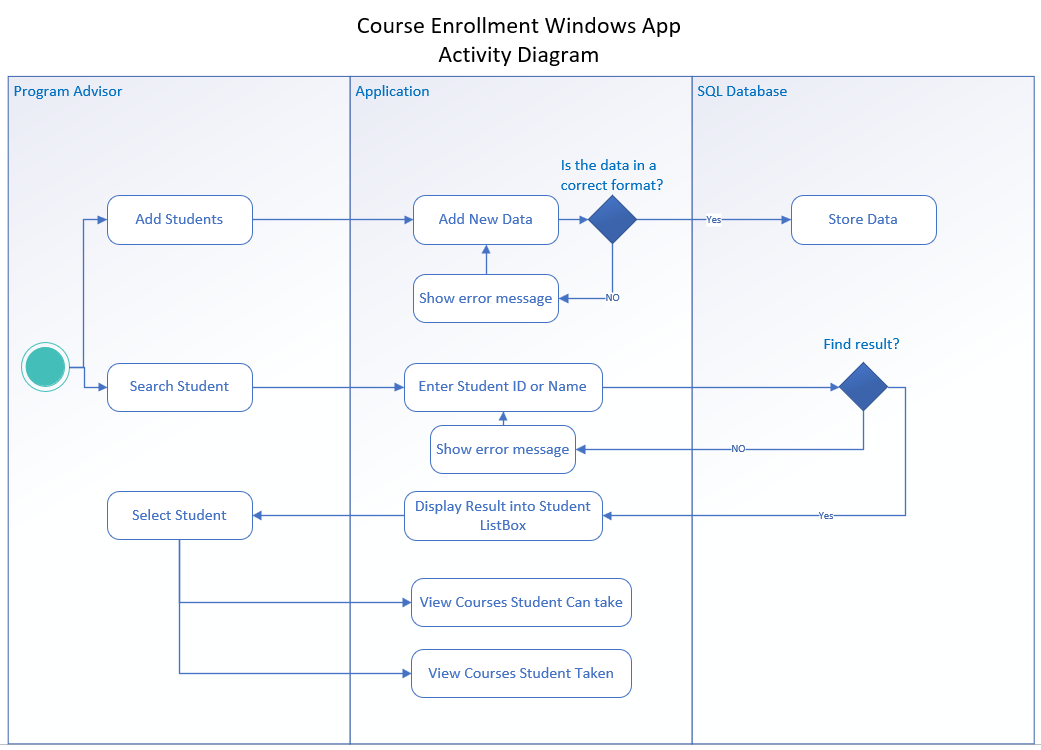


Figure 6.4.1 Login Active Diagram (Future Plan)

Figure 6.4.2 Active Diagram

## 6.5 Wireframes

In this project, we use Figma to design our wireframes. Figma is a powerful design tool that helps us to create anything: websites, applications, logos, and much more. To view our prototype on figma, you can access [HERE](https://www.figma.com/file/EPrFU7aiFByvYZxXCvOZPV/Wireframes).

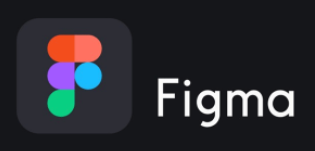


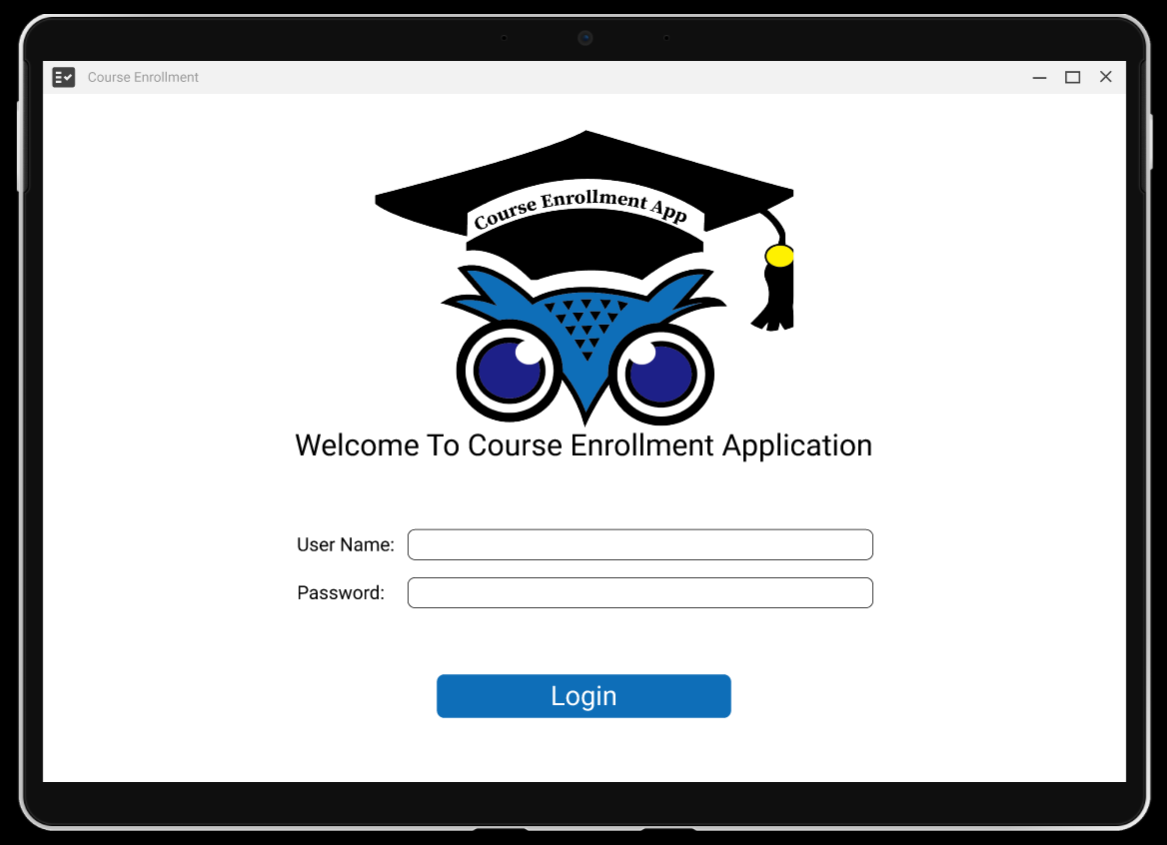
Figure 6.5.1 Figma Logo

Also, for our project’s logo, we are using Adobe illustrator app to design. Illustrator lets you create precise, editable vector graphics that stay sharp when scaled to any size. Use flexible shape and drawing tools to create logos, icons, and other illustrations that look equally good on a business card, flyer, or billboard. Edit and customize text in many ways to create striking typography.

一張含有 文字 的圖片

自動產生的描述

Figure 6.5.2 Illustrator Logo

一張含有 桌 的圖片

自動產生的描述Figure 6.5.3 Wireframe Login Page

Figure 6.5.4 Wireframe Main Page

# 7. Test Result

## 7.1 User Testing

To examine our application in more depth, we did the above tests:

### 7.1.1 Get All Student

Graphical user interface, text, application

Description automatically generated with medium confidence

Figure 7.1.1 Testing result1

### 7.1.2 Get All Courses Information

Text

Description automatically generated with low confidence

Figure 7.1.2 Testing result2

### 7.1.3 Get Prerequisites Courses for Student

Graphical user interface, text, application, email

Description automatically generated

Figure 7.1.3 Testing result3

### 7.1.4 Get Student Uncompleted Courses

Graphical user interface, text, application, email

Description automatically generated

Figure 7.1.4 Testing result4

### 7.1.5 Get Non-Prerequisite Courses for Student

Text

Description automatically generated with medium confidence

Figure 7.1.5 Testing result5

# 8. User Guide

## 8.1 Installation

For installing our application, user can go to GitHub to clone the repository into their machine and open the solution file. Due to the time limit, we could not done the UI implementation so if the user wants to test the project, should use the FinalAConsole and run it.

Text

Description automatically generated

Figure 8.1 Use FinalAConsole to run

# 9. Conclusion

## 9.1 Lessons Learned

This project has been a meaningful learning experience for our team. We have learned many valuable skills from applying project management skills, drafting technical reports, and developing a windows application.

During the design phase, we had little idea of how to manage a

windows application project, such as defining the project scope, constructing a schedule, and creating the database on Microsoft Azure. Fortunately, our instructor has taught us step by step. In addition, it helped us to have more understanding of database structure to manage our application.

In the application development phase, we learned about making a

proper backend system and made it work. Understanding my weakness is the backend. The instructor gave us intense classes to provide us with a full understanding of it.

Finally, we have learned to utilize some useful windows application development tools such as Visual Studio 2019 and Microsoft Azure. Now I have more understanding of using databases to store data and use it with windows applications.

## 9.2 Known Issues

Unfortunately, due to the time limitations, I could not complete the UI part and the Year – Term. However, I will keep going to do it to make it as my portfolio in the future.

## 9.3 Future Improvements

* Complete UI part.
* Make the login system.
* More complete functionality of searching system.

# 10. References

* <https://www.w3schools.com/sql/sql_alter.asp>
* <https://www.youtube.com/watch?v=AEjwvfa-Aj0&list=PLhPyEFL5u-i3CoTQC9wPhB9uECUxTxkiu&index=3>
* <https://www.completecsharptutorial.com/ado-net/first-ado-net-program-in-c.php>
* <https://zetcode.com/csharp/sqlite/>
* <https://sqliteonline.com/syntax/alter_table/>
* <https://www.youtube.com/watch?v=8NdJaztrNk8>