Tunku Abdul Rahman University College

BACS2023
Object-Oriented Programming

Assignment

Programme : RSD2

Tutorial Group : Group 1

Date Submitted to Tutor : 10 December 2021

Team Members:

No	Student Name	Student ID	Contribution (%)	Signature
1.	GUI HUI CHYI	20WMR12153	25	gui
2.	DESMOND ONG KHAI YANG	20WMR12046	25	desmond
3.	FONG ZHI JUN	21WMR00063	25	Jun
4.	TEOH ZI CONG	20WMR12048	25	TEO
		Total	100	-

Table of Content

Task Allocation **2**

UML Class Diagram **4**

Description of Assignment Idea **5**

General Description 5

Individual Description Based On Task Assigned 6

Course Module by Gui Hui Chyi 6

Staff Module by Teoh Zi Cong 8

Student Module by Fong Zhi Jun 9

Enroll, Payment Module by Desmond Ong Khai Yang 11

Screen Capture/Screenshot **12**

GUI HUI CHYI 12

TEOH ZI CONG 18

FONG ZHI JUN 23

DESMOND ONG KHAI YANG 30

Concepts of Object-Oriented Programming Applied **39**

GUI HUI CHYI 39

TEOH ZI CONG 42

FONG ZHI JUN 44

DESMOND ONG KHAI YANG 47

1. Task Allocation

No.	Team Member	Task(s) Allocated
1.	GUI HUI CHYI	Menu ● Main menu section Course Module ○ Curriculum ○ Co-curriculum ● Applying CRUD functions (MySQL) ● Generate Course Report
2.	TEOH ZI CONG	Menu ● Staff Login and Input Validation Method Staff Module ● Staff , Full time Staff and Part time Staff ● Apply CRUD functions ● Generate Staff Report
3.	FONG ZHI JUN	Menu ● Student Login & Input Validation Method Student Module ● Student, LocalStudent & InternationalStudent Class ● Various CRUD functions in StudentService Class including generating Student Report Helper Module ● AES Encryption & Decryption Method ● Email Method ● Generate Pie Chart Method
4.	Desmond Ong Khai Yang	Menu ● Registration, and input validation method Enrollment module ● View and Select Course Payment Module ● Calculate cost CRUD in payment class/sql ● Generate Receipt / invoice

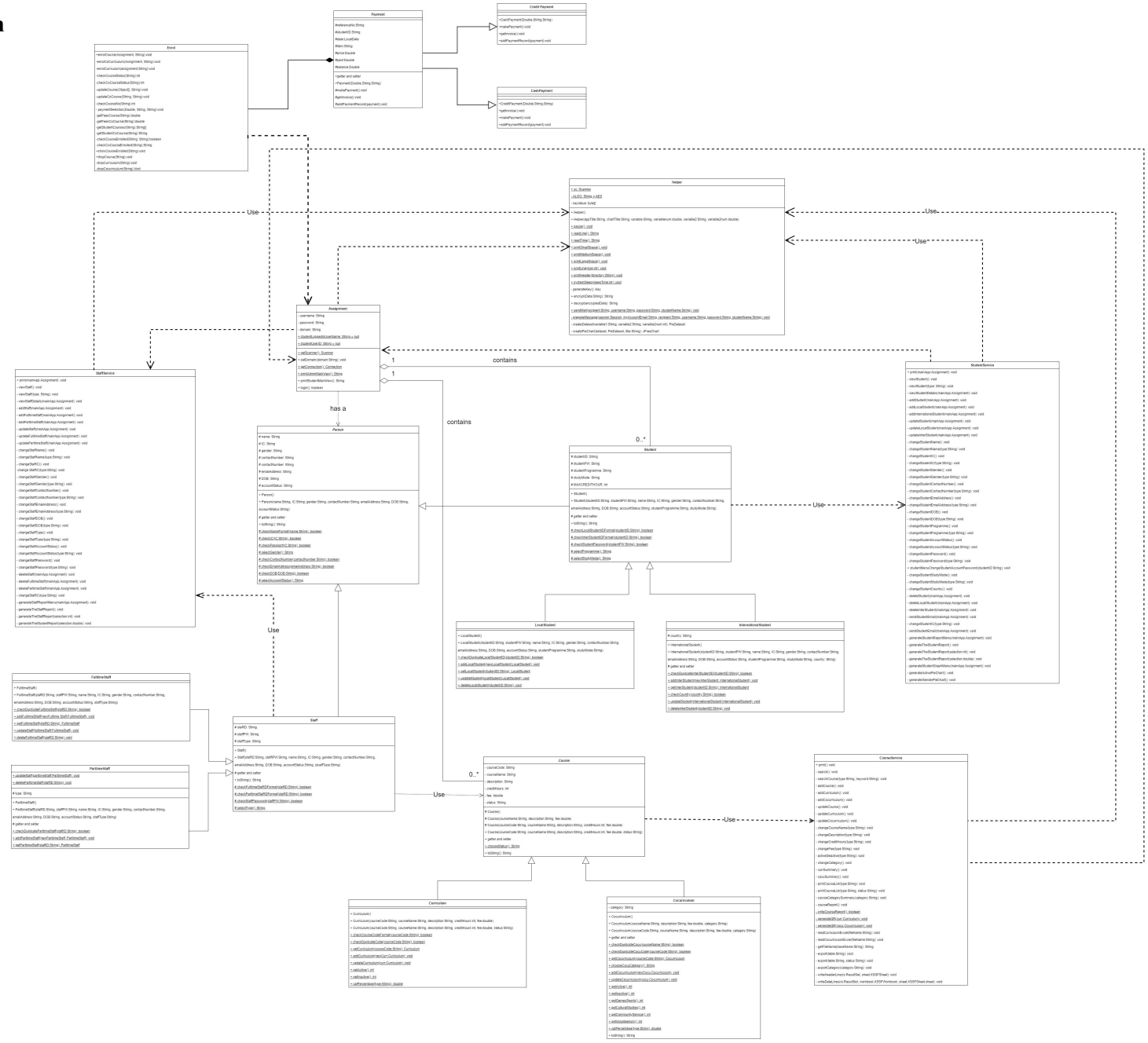
Declaration

We confirm that we have read and shall comply with all the terms and conditions of TAR University College's plagiarism policy.

We declare that this assignment is free from all forms of plagiarism and for all intents and purposes is my own properly derived work.

Signature :	gui	desmond	Jun	TEO
Name :	GUI HUI CHYI	DESMOND ONG KHAI YANG	FONG ZHI JUN	TEOH ZI CONG
Date :	10 December 2021	10 December 2021	10 December 2021	10 December 2021

2. UML Class Diagram



3. Description of Assignment Idea

General Description

We had decided to opt for a college enrolment system named "ANAVRIN" to implement the concepts of object-oriented programming, including Inheritance, Polymorphism, and Encryption into the Java program. In addition, we also make use of our database skill, MySQL to store, retrieve and execute Staff, Student, Course, and Enrolment information.

ANAVRIN Enrolment System is a college application meants for academic staff and undergraduate students. Thus, the user interface is designed to be user-friendly for novice users to IT savvy to operate on it. For instance, we have incorporated features such as the quit (Q) option is available to allow for internal locus of control. When users input invalid data, an error message will be prompted instead of the system crashing and tips will be provided to guide users for correct input. Moreover, there is a header in each of the sections so students and staff will have a clear view, for example, 'Admin Main > Add Course', to indicate to the User which function they are on.

The application allows the creation of new courses and adding of staff and student records as well as registration of courses and students enrollment. Users will be separated into 2 types which are Students and Administrators where they have respective account privileges. There will be an administrator mode for academic staff and user mode for students. Administrator mode can perform CRUD on courses, staff and students. Beside, staff can use admin mode to generate reports to monitor the system status such as percentage of available courses and reports of inactive students' accounts. User mode on the other hand can perform their daily tasks such as view timetable and enroll for new courses.

Assumptions: A student should take no more than 21 credits of courses. A student is allowed to enroll in one co-curriculum only. All the course status is available by default. A student cannot enroll in an unavailable course. They will also see their own respective interface based on the functions that they are allowed to perform on the application.

Individual Description Based On Task Assigned

Course Module by Gui Hui Chyi

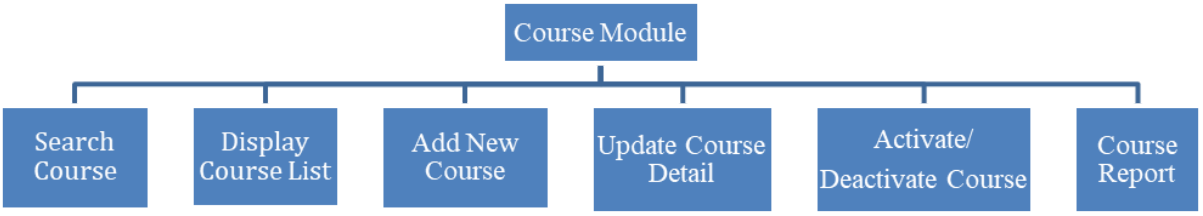


Figure 3.1 Function of Course Module

1. **Search Course**
   The Search Course function allows the staff and students to search for a course. By entering the keyword of the course name, a list of course records which fulfill the searching requirement will be displayed.

2. **Display Course List**
   The Display Course List function allows the staff to view the course list of curriculum and co-curriculum courses. There is some filtering provided such as shows all, shows available courses, shows unavailable courses and shows courses by their category. This function also supports the export of course lists to Excel files, allowing the staff to save the details of an export operation for future use.

3. **Add New Course**
   When there are new courses introduced to the college, the staff can add the course information such as course code, course name, course description, credit hours, and tuition fee. For curriculum course, the course code must be in a format of four letters followed by four digits (i.e. BACS2023). For co-curriculum courses, the course code is auto generated by the system with format of COCU9999. The course code is uneditable to avoid duplicated code. Whenever a new course is added, a QR Code will also be generated automatically. The QR Code will contain information such as course code, course name, course description, credit hours, fee, status and category (for co-curriculum). The Add New Course function also supports the import of data from Excel files, allowing the staff to add a list of courses simultaneously.

4. **Update Course Detail**
   Update Course Detail function is available in the Course Module. To perform updating, a staff member is required to enter the course code of a particular course. All the course details can be changed except for the course code. Any changes made will also be updated in the particular course's QR code.

5. **Activate/ Deactivate Course**
   Activate/ Deactivate Course function is also available in the Course Module. When the course is no longer offered to students, a staff can deactivate the course. After confirmation is made, the particular course is deactivated successfully. Now, the course is no longer available for enrolling, its status will change from "Available" to "Unavailable".

6. **Course Report**

   Report Generation is also available in the Course Module to allow the staff to view the course summary. The Course Report has the data including total course, the breakdown of course by course availability and course category. Only the staff is allowed to generate the report. The staff can also save the report as a text file.

Staff Module by Teoh Zi Cong

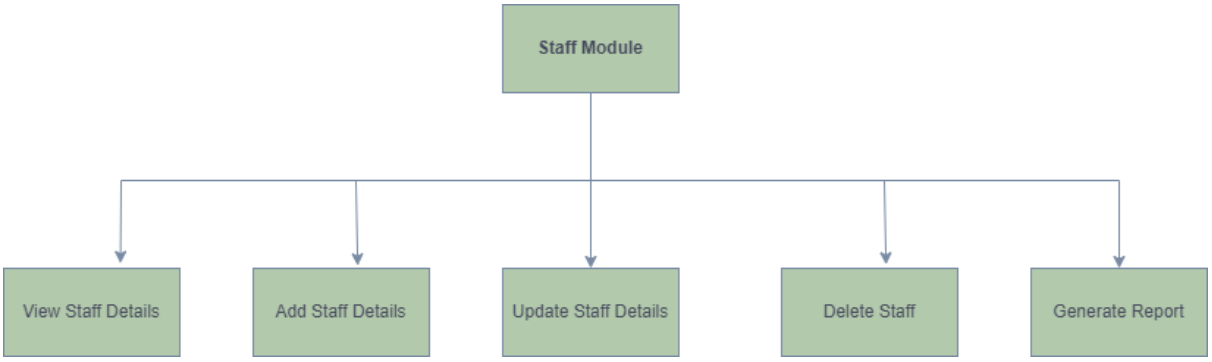


Figure 3.2 Function of StaffModule

1. **<u>View Staff Details</u>**
   This module allows staff to view the staff details . Staff can access this module by using staffID to view the details of the full time staff and the part time staff . The system will present an error message when the staff typo the staffID and need the staff to type the correct staff ID to access this module.

2. **<u>Add Staff Details</u>**
   This module allows staff to add a new staff member . Staff need to enter the staff information to complete the step . For example , staffID , name , gender , contact number and so on . After staff complete the staff information and it will present the message to staff , Staff need to enter 'Y' to add the staff member into the records.

3. **<u>Update Staff Details</u>**
   This module is to let the staff update staff details. Sometimes staff want to change their information and this module can provide them with the latest information .

4. **<u>Delete Staff</u>**
   This module is to delete the staff from the Full time Staff or Part time Staff .

5. **<u>Generate Report</u>**
   Staff can use this module to generate reports of the staff details such as full time staff list or the part time staff list .

Student Module by Fong Zhi Jun

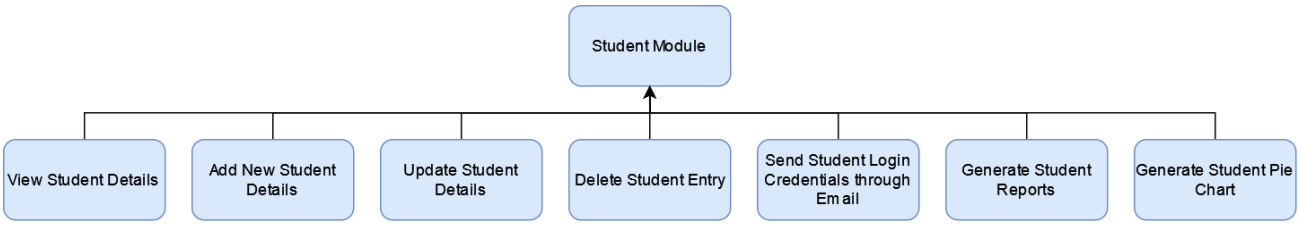


Figure 3.3 Function of Course Module

1. **<u>View Student Details</u>**

This function is for staff to check and view specific student details regardless of local or international students through entering the correct sequence of student ID. The system will prompt for error if an incorrect student ID sequence is entered and ask for retry.

2. **<u>Add New Student Details</u>**

Staff can enter new student details through this function by entering details such as student name, IC, contact number, email address and so on. Staff will be able to specify which type of student either local or international students before proceeding to enter the details. System will also perform a student ID duplication check to prevent duplication of student ID and various input validation checks in place to assist staff in entering correct input. For example, no digits and symbols are allowed in name input, correct email sequence check, and student ID character sequence check like 10 characters for local and 11 characters for international students such as 22WMR99999 and 22IWMR77777. Besides, student passwords will be encrypted into string before saving inside the database for security purposes.

3. **<u>Update Student Details</u>**

As the function implies, staff are able to update various students' details in case of typo in the student details or keeping student account details up to date. It can be done through entering the correct student ID to access and update those. Otherwise, students can also update their student account password from the student account view.

4. **<u>Delete Student Details</u>**

This function is meant for obsolete student account deletion such as when the student withdraws from the college or the student has departed. Otherwise, the staff are strongly advised to update student status from active to inactive for good bookkeeping practices.

5. **<u>Send Student Login Credentials through Email</u>**

Once a student account is created and error free in those student particulars, staff can use this function to send the student login credentials to his / her private email address so students can start login into the system and perform daily tasks such as view the current timetable.

6. **<u>Generate Student Reports</u>**

Staff can use this function to generate various useful reports such as list out all the students details, active or inactive students accounts list in a txt format. The report lists will then be generated and saved under the same java source code folder.

7. **Generate Student Pie Chart**

This function will generate pie charts based on 2 staff input selection whether active and inactive student accounts or total of students gender pie chart. This will give staff a clear visual view of current student status.

Enroll, Payment Module by Desmond Ong Khai Yang

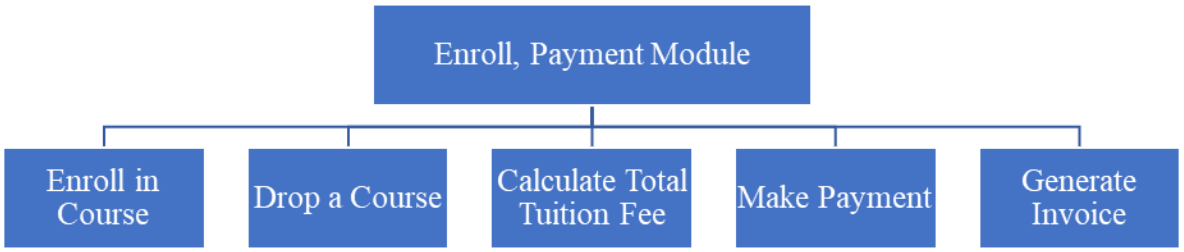


Figure 3.4 Function of Enroll, Payment Module

1. **<u>Enrol Course</u>**
   This module allows students to enroll in multiple courses including the co curricular courses. The student may first login to their account. Then they could select their desired courses from a list of courses by inputting the course code. They will be prompted to make payment for the tuition fee. After which the student will be enrolled to its course.

2. **<u>Drop a Course</u>**
   This module allows students to drop their courses when they need to. The system will allow the students to drop their course if they are enrolled. The student will have to enter the course code of the course they want to drop in order to drop the course

3. **<u>View Courses</u>**
   This module allows the students to view their respective courses and co curricular courses they are currently enrolled in. They will have to first login to their account to view their courses.

4. **<u>Make Payment</u>**
   This module allows the user to make payment for their courses. They can choose from two payment options; with cash or with bank cards;. The payment module will calculate the balance and will update the database accordingly.

5. **<u>Generate Invoice</u>**
   Students can use this module to generate an Invoice for their purchase and payment. This will allow the user to view their invoice as well as save it to a file.

4. Screen Capture/Screenshot

No.	Team Member	Sample Run (Screen Capture)
1.	GUI HUI CHYI	

Main Menu:

After the staff login successfully, he will be brought to the main menu page. The staff can select any options from 1 to 4.

```
╔═══════════════════════════════════════════════════════════════════════════════════════╗
║                                                                                       ║
║                                          Welcome to                                   ║
║                                                                                       ║
║                                  _   _     __      _______  _____ _   _               ║
║                                 /\   | \ | |   /\ \    / /  __ \|_   _| \ | |        ║
║                                /  \  |  \| |  /  \ \  / /| |__) | | | |  \| |        ║
║                               / /\ \ | . ` | / /\ \ \/ / |  _  /  | | | . ` |        ║
║                              / ____ \| |\  |/ ____ \  /  | | \ \ _| |_| |\  |        ║
║                             /_/    \_\_| \_/_/    \_\/   |_|  \_\_____|_| \_|        ║
║                                                                                       ║
║                                Anavrin College Enrollment System                      ║
║                                                                                       ║
╚═══════════════════════════════════════════════════════════════════════════════════════╝

╔═══════════════════════════════════════════════════════════════════════════════════════╗
║ Admin Main                                                                            ║
╚═══════════════════════════════════════════════════════════════════════════════════════╝
What would you like to do?
[1] Staff
[2] Student
[3] Course
[4] Logout
Choice                                                  : |
```

Course Module:

The staff can input '3' to enter the Course module. The staff can choose from 1 to 6 on what he would like to do.

```
╔═══════════════════════════════════════════════════════════════════════════════════════╗
║                                          Welcome to                                   ║
║                                  _   _     __      _____  ____ _   _                 ║
║                                 /\   | \ | |   /\ \    / /  __ \|_   _| \ | |        ║
║                                /  \  |  \| |  /  \ \  / /| |__) | | | |  \| |        ║
║                               / /\ \ | . ` | / /\ \ \/ / |  _  /  | | | . ` |        ║
║                              / ____ \| |\  |/ ____ \  /  | | \ \ _| |_| |\  |        ║
║                             /_/    \_\_| \_/_/    \_\/   |_|  \_\____|_| \_|        ║
║                                Anavrin College Enrollment System                      ║
╚═══════════════════════════════════════════════════════════════════════════════════════╝

╔═══════════════════════════════════════════════════════════════════════════════════════╗
║ Admin Main > Course                                                                   ║
╚═══════════════════════════════════════════════════════════════════════════════════════╝
What would you like to do?
[1] Search Course
[2] Add Course
[3] Update Course
[4] Display Course List
[5] View Course Summary Report
[6] Return to Main Menu
Choice                                                  :
```

If the staff enters options other than 1 to 6, an error message is shown to alert the staff and prompt him to re-enter his option.

```
What would you like to do?
[1] Search Course
[2] Add Course
[3] Update Course
[4] Display Course List
[5] View Course Summary Report
[6] Return to Main Menu
Choice                                                  : 8
Please enter valid option.
```

1. Add New Nook Function

If the staff entered 2, he will be brought to the Add Course page. The staff is required to choose the course type of a course to be added. The two course types are Curriculum and Co-curriculum.

```
Enter Course Type
[1] Curriculum
[2] Cocurriculum
[3] Import Curriculum From Excel
[4] Import Cocurriculum From Excel
Enter choice (Q to quit)                    : 1
```

For instance, the staff enters '1' to add a new Curriculum course.

- **Add Curriculum**

As shown in the figure below, Curriculum with course code (BACS2023) already exists. If the staff attempts to add this existing curriculum again, an error message will be prompted to avoid duplicated curriculum.

```
Admin Main > Add Curriculum
```

Course Code	Course Name	Course Description	Credit Hours	Tuition Fee(RM)	Status
BACS2023	OBJECT-ORIENTED PROGRAMMING	OO IN JAVA	3	777.00	AVAILABLE
BAIT1173	IT FUNDAMENTALS	FOCUSES ON THE BASICS OF COMPUTING	3	777.00	AVAILABLE
BAIT2004	FUNDAMENTAL COMPUTER NETWORKS	NETACAD CISCO FOR BEGINNER	4	1036.00	AVAILABLE
BAMS1623	DISCRETE MATHEMATICS	IMPROVE PROGRAMMING SOLVING ABILITY	3	777.00	UNAVAILABLE
BMIT2164	COMPUTER NETWORKS	NETACAD CISCO	4	1036.00	AVAILABLE

Total Curriculum Course: 5

```
Enter new course code (Q to quit)                : bacs2023
Course Code BACS2023 already exist!
```

Now, the staff will add a new curriculum.

```
Enter new course code (Q to quit)                : bait1023
Enter course name                                : web design and development
Enter course description                         : building a website using html, css and javascript
Enter credit hours                               : 3
Enter course tuition fee                         : RM 777
```

Later, the system will list down all the course details entered just now to ask for confirmation. If the staff confirms to add the course, he can enter 'y' (case-insensitive). A success message is shown after the staff confirms to add the course. The course record is added.

```
New Curriculum Details:
Course Code                          : BAIT1023
Course Name                          : WEB DESIGN AND DEVELOPMENT
Description                          : BUILDING A WEBSITE USING HTML, CSS AND JAVASCRIPT
Credit Hours                         : 3
Tuition Fee                          : RM 777.00

Confirm? y/n                         : y
New curriculum added successfully.
QR Code for Course BAIT1023 is created/updated successfully!
```

Whenever a course is created, the course's QR Code will be generated at the same time. Each QR Code consists of the information of course code, course name, description, credit hours, fee, status and category(for co-curriculum). The QR Code is stored in the folder named Course-QRCode.

Course Code: BAIT1023
Course Name: WEB DESIGN AND DEVELOPMENT
Description: BUILDING A WEBSITE USING HTML, CSS AND JAVASCRIPT
Credit Hours: 3
Tuition Fee: RM 1036.00
Status: UNAVAILABLE

QR Code for curriculum (BAIT1023)

Staff can also import courses from excel files by choosing option ‘3’ and ‘4’.

Example (Import Curriculum from Curriculum.xlsx):

	A	B	C	D	E	F
1	Course Code	Course Name	Description	Credit Hours	Fee	Status
2	BAIT2164	ADVANCED COMPUTER NETWORKS	ADVANCED TOPICS IN COMPUTER NETWORKING	4	1036	AVAILABLE
3	BACS2063	DATA STRUCTURES AND ALGORITHMS	LEARNING METHOD OF ORGANIZING DATA IN VER	3	777	AVAILABLE
4	AACS1074	PROGRAMMING CONCEPTS AND DESIGN I	C++ PROGRAMMING	4	1036	AVAILABLE
5	ABDM1183	INTRODUCTION TO BUSINESS	THE FOUNDATIONS OF BUSINESS	3	777	AVAILABLE

```
Admin Main > Add Course > Import Curriculum From Excel

Enter Excel File Name (Q to quit):                : Curriculum
QR Code for Course BAIT2164 is created/updated successfully!
QR Code for Course BACS2063 is created/updated successfully!
QR Code for Course AACS1074 is created/updated successfully!
QR Code for Course ABDM1183 is created/updated successfully!
Curriculum import successfully!
```

Course Code	Course Name	Course Description	Credit Hours	Tuition Fee(RM)	Status
AACS1074	PROGRAMMING CONCEPTS & DESIGN	C++ PROGRAMMING	4	1036.00	AVAILABLE
ABDM1183	INTRODUCTION TO BUSINESS	THE FOUNDATIONS OF BUSINESS	3	777.00	AVAILABLE
BACS2023	OBJECT-ORIENTED PROGRAMMING	OO IN JAVA	3	777.00	AVAILABLE
BACS2063	DATA STRUCTURES AND ALGORITHMS	LEARNING METHOD OF ORGANIZING DATA IN VERTUAL SYSTEM	3	777.00	AVAILABLE
BAIT1023	WEB DESIGN AND DEVELOPMENT	BUILDING A WEBSITE USING HTML, CSS AND JAVASCRIPT	3	1036.00	UNAVAILABLE
BAIT1083	VISUAL PROGRAMMING	DEVELOPING WINDOWS-BASED APPLICATIONS	3	777.00	AVAILABLE
BAIT1173	IT FUNDAMENTALS	FOCUSES ON THE BASICS OF COMPUTING	3	777.00	AVAILABLE
BAIT2004	FUNDAMENTAL COMPUTER NETWORKS	NETACAD CISCO FOR BEGINNER	4	1036.00	AVAILABLE
BAIT2164	ADVANCED COMPUTER NETWORKS	ADVANCED TOPICS IN COMPUTER NETWORKING	4	1036.00	AVAILABLE
BAMS1623	DISCRETE MATHEMATICS	IMPROVE PROGRAMMING SOLVING ABILITY	3	777.00	UNAVAILABLE
BMIT2164	COMPUTER NETWORKS	NETACAD CISCO	4	1036.00	AVAILABLE

Course Code	Course Name	Course Description	Credit Hours	Tuition Fee(RM)	Status
AACS1074	PROGRAMMING CONCEPTS & DESIGN	C++ PROGRAMMING	4	1036.00	AVAILABLE
ABDM1183	INTRODUCTION TO BUSINESS	THE FOUNDATIONS OF BUSINESS	3	777.00	AVAILABLE
BACS2023	OBJECT-ORIENTED PROGRAMMING	OO IN JAVA	3	777.00	AVAILABLE
BACS2063	DATA STRUCTURES AND ALGORITHMS	LEARNING METHOD OF ORGANIZING DATA IN VERTUAL SYSTEM	3	777.00	AVAILABLE
BAIT1023	WEB DESIGN AND DEVELOPMENT	BUILDING A WEBSITE USING HTML, CSS AND JAVASCRIPT	3	1036.00	UNAVAILABLE
BAIT1083	VISUAL PROGRAMMING	DEVELOPING WINDOWS-BASED APPLICATIONS	3	777.00	AVAILABLE
BAIT1173	IT FUNDAMENTALS	FOCUSES ON THE BASICS OF COMPUTING	3	777.00	AVAILABLE
BAIT2004	FUNDAMENTAL COMPUTER NETWORKS	NETACAD CISCO FOR BEGINNER	4	1036.00	AVAILABLE
BAIT2164	ADVANCED COMPUTER NETWORKS	ADVANCED TOPICS IN COMPUTER NETWORKING	4	1036.00	AVAILABLE
BAMS1623	DISCRETE MATHEMATICS	IMPROVE PROGRAMMING SOLVING ABILITY	3	777.00	UNAVAILABLE
BMIT2164	COMPUTER NETWORKS	NETACAD CISCO	4	1036.00	AVAILABLE

Courses from Curriculum.xlsx are added successfully.

Exception Handling

For credit hours, exception handling is applied to catch all invalid data entries such as the alphabet, number greater than 4 and number less than 1 (Assuming credit hours is between 1 and 4).

```
Enter credit hours                                  : a
Please enter valid number between 1 and 4.
Enter credit hours                                  : 0
Please enter valid number between 1 and 4.
Enter credit hours                                  : 8
Please enter valid number between 1 and 4.
Enter credit hours                                  : 3
```

The exception handling is also implemented in course tuition fees. Invalid data entries such as alphabet and number less or equal to zero will prompt an error message (Assume no curriculum is free).

```
Enter course tuition fee                            : RM 0
Please enter valid number.
Enter course tuition fee                            : RM a
Please enter valid number.
Enter course tuition fee                            : RM 1036
```

2. Search Book Function

Now, the staff quits this Add Curriculum page, and goes to the Search Curriculum page.

```
Enter new course code (Q to quit)                   : Q

What would you like to do?
[1] Search Course
[2] Add Course
[3] Update Course
[4] Display Course List
[5] View Course Summary Report
[6] Return to Main Menu
Choice                                              : 1
```

When the staff enters the course name keyword, he is able to find the course record he added just now.

```
Admin Main > Search Curriculum

Enter keyword (Q to quit)                   : web
```

Course Code	Course Name	Course Description	Credit Hours	Tuition Fee(RM)	Status
BAIT1023	WEB DESIGN AND DEVELOPMENT	BUILDING A WEBSITE USING HTML, CSS AND JAVASCRIPT	3	1036.00	UNAVAILABLE

1 course with "WEB"

3. Update Course Detail Function

The admin enters '3' in the Course Menu to update the course.

```
What would you like to do?
[1] Search Course
[2] Add Course
[3] Update Course
[4] Display Course List
[5] View Course Summary Report
[6] Return to Main Menu
Choice                                              : 3
```

The staff then needs to choose which information to update. The staff can update any information except for the course code to avoid course code duplicated.

In this case, the staff changes the curriculum course "BAIT1023" tuition fee from RM 777 to RM

1036. Thus, the staff chooses option ‘4’.

```
What would you like to do?
[1] Change Course Name
[2] Change Course Description
[3] Change Credit Hours
[4] Change Tuition Fee
[5] Deactivate/Active Course
[6] Return to Main Menu
Enter Choice                                    : 4
```

In the end, the staff is required to confirm the edit. If the staff enters ‘y’ (case-insensitive) then the course is updated successfully.

```
Enter the Course Code (Q to Quit): bait1023
Enter new Course Tuition Fee : RM 1036
This Course Tuition Fee will be updated to Course BAIT1023: RM 1036.00

Confirm? y/n: y
Course Tuition Fee Changed Successfully!
QR Code for Course BAIT1023 is created/updated successfully!
```

4. Deactivate/ Activate Course

All Course status is Available by default. To deactivate/activate a course, the staff can choose option ‘5’.

```
What would you like to do?
[1] Change Course Name
[2] Change Course Description
[3] Change Credit Hours
[4] Change Tuition Fee
[5] Deactivate/Active Course
[6] Return to Main Menu
Enter Choice                                    : 5
```

Let say the staff wants to deactivate curriculum (BAIT1023)

```
Enter the Course Code (Q to Quit): BAIT1023
Status
[1] Activate
[2] Deactivate
Update Status:                                  : 2
This Course Status will be updated to Course BAIT1023: UNAVAILABLE

Confirm? y/n: Y
Course Status Changed Successfully!
QR Code for Course BAIT1023 is created/updated successfully!
```

5. Course Report

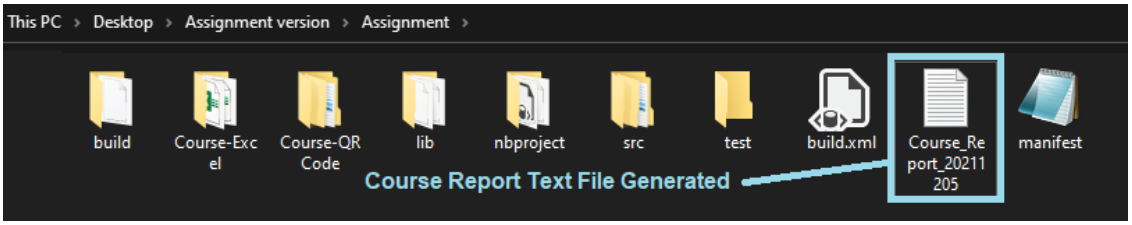
To view the course report, the staff can choose option ‘5’.

```
What would you like to do?
[1] Search Course
[2] Add Course
[3] Update Course
[4] Display Course List
[5] View Course Summary Report
[6] Return to Main Menu
Choice                                          : 5
```

```
ANAVRIN COLLEGE ENROLLMENT SYSTEM
COURSE SUMMARY REPORT

Generated On: 2021-12-05 14:58:21

Curriculum Course
By Course Availability:-
● Available Course                    : 5 (71.43%)
● Unavailable Course                  : 2 (28.57%)

Total                                 : 7

Cocurriculum Course
By Course Availability:-
● Available Course                    : 1 (25.00%)
● Unavailable Course                  : 3 (75.00%)

By Course Category:-
● Games/Sports                        : 1 (25.00%)
● Cultural Studies                    : 1 (25.00%)
● Community Service                   : 1 (25.00%)
● Volunteerism                        : 1 (25.00%)

Total                                 : 4

Grand Total Course                    : 11
```

The course report is generated, and the staff can choose to save this report as a text file.

```
Do you want to save the report? y/n: y
Course Report generated successfully!
```



6. Display Course List

To view the course list, the staff can choose option '4'.

```
What would you like to do?
[1] Search Course
[2] Add Course
[3] Update Course
[4] Display Course List
[5] View Course Summary Report
[6] Return to Main Menu
Choice                                    : 4
```

The staff member can choose whether he wants to view the curriculum list or co-curriculum list.

```
What would you like to do?
[1] Curriculum Course List
[2] Cocurriculum Course List
Enter Choice (Q to quit)                  :
```

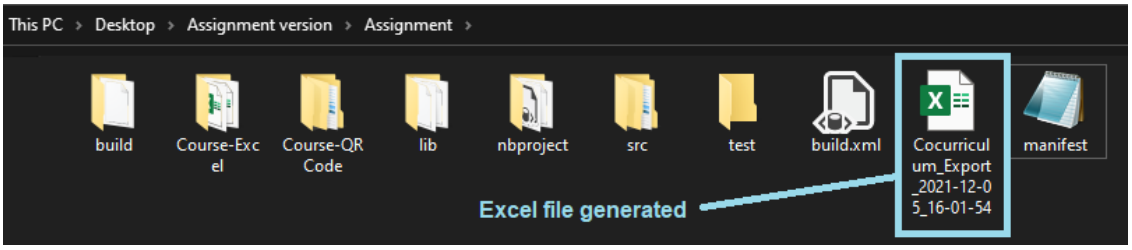
Example (Curriculum)
The staff can view the course list by status selection, including shows all, shows active course and shows inactive course.

```
View Curriculum Course List By Status
Status
[1] Shows All
[2] Shows Available Course Only
[3] Shows Unavailable Course Only
[4] Return to Main Menu
Enter Choice                                    :
```

Example (Co-curriculum)
The staff can view the course list by status selection, including shows all, shows active course and shows inactive course as well as by category (e.g. Games/Sports, Cultural Studies, Community Service, Volunteerism).

```
View Cocurriculum Course List By Filtering
Filtering
[1] Shows All
[2] Shows Available Course Only
[3] Shows Unavailable Course Only
[4] Shows Course Filtering By Category
[5] Return to Main Menu
Enter Choice                                    :
```

After the staff chooses his filtering selection, the course records fulfilling the filtering will be listed down. The staff can choose to export the course records into an excel file.



2.	TEOH ZI CONG

1. **Staff Module Menu**

```
Admin Main > Staff

What would you like to do?
[1] View Staff Details
[2] Add New Staff Details
[3] Update Staff Details
[4] Delete Staff
[5] Generate Staff Reports
[6] Return to Previous Menu
Choice
```

Staff will provide 6 options for users to choose.

```
╔═══════════════════════════════════════════════════════════
‖ Admin Main > Staff
╚═══════════════════════════════════════════════════════════
What would you like to do?
[1] View Staff Details
[2] Add New Staff Details
[3] Update Staff Details
[4] Delete Staff
[5] Generate Staff Reports
[6] Return to Previous Menu
Choice                                                : 7
Please enter valid option.
```

If the user types other than 1 to 6 and it will present an error message “Please enter a valid option” to remind the user to enter the correct input.

2. View Staff Details

```
╔═══════════════════════════════════════════════════════════
‖ Admin > Staff: View StaffDetails
╚═══════════════════════════════════════════════════════════

‖Staff ID          ‖ Staff Name                          ‖ Staff Status

Fulltime Staff List

‖FT001             ‖ WEISHENG                            ‖ Activate

‖FT002             ‖ JANNY                               ‖ Activate

Parttime Staff List

‖PT0001            ‖ ZICONG                              ‖ Activate

‖PT0002            ‖ ZIJUN                               ‖ Activate

‖PT0003            ‖ HUICHYI                             ‖ Activate

‖PT0005            ‖ JACELYN                             ‖ Activate

Total Staff Count: 6
```

If the user wants to view the staff details , users need to enter “1” for the option to view the staff details . There are two staff details to be shown such as “Full Time Staff “ and the “Part Time Staff”.

```
Enter the Staff ID in order to view the staff details (Q to Quit): FT004
Staff Record not found. Please double check the input.
```

If the user enters a staff ID that does not appear in the staff details, it will present an error message “ Staff Record Not Found . Please double check the input” to remind the users to enter the correct input .

```
Enter the Staff ID in order to view the staff details (Q to Quit): FT001

Fulltime Staff Details:
Staff ID                              : FT001
Staff  Password                       : ws012355
Staff Name                            : WEISHENG
Staff IC                              : 998120-01-3889
Staff Gender                          : MALE
Staff Contact Number                  : 016-832-9901
Staff Email Address                   : weisheng@gmail.com
Staff DOB                             : 01-12-1998
Staff Account Status                  : Activate
Staff Staff Type                      : FullTime

Enter the Staff ID in order to view the stffdetails (Q to Quit): PT0001

Parttime Staff Details:
Staff ID                              : PT0001
Staff Password                        : zicong99
Staff Name                            : ZICONG
```

Users can enter the Staff ID to view the staff details.

3. Add Staff Details

```
Enter staff type:
[1] Fulltime Staff
[2] Parttime Staff
```

The system will present the message about the staff details such as “Fulltime Staff “ and the “Part time Staff” to provide the user to select what staff type that they want to add .

```
Admin > Staff: Add Fulltime Staff Details

| Staff ID      | Staff Name                                  | Staff Status |
| FT001         | WEISHENG                                    | Activate     |
| FT002         | JANNY                                       | Activate     |

Admin > Staff: Add Parttime Staff Details

| Staff ID      | Staff Name                                  | Staff Status |
| PT0001        | ZICONG                                      | Activate     |
| PT0002        | ZIJUN                                       | Activate     |
| PT0003        | HUICHYI                                     | Activate     |
| PT0005        | JACELYN                                     | Activate     |

Enter new staff ID Eg:FT001 (Q to quit)          : FT001
StaffID FT001 already exist!
```

The staff record will appear to prevent the user entering the same Staff ID with others .

```
Enter Staff name                                    : minteck
Enter Staff IC     eg:999999-99-9999                : 21
Invalid input. Please check your input. eg:999999-99-9999.
Enter Staff IC     eg:999999-99-9999                : 010201-21-0032
Gender :
                [1] Male
                [2] Female
Choice:                                             : 1
Enter staff contact number eg: 011-9999999          : 32121
Invalid input. Please double check your number input.
Enter staff contact number eg: 011-9999999          : 010-2312342
Enter Staff date of birth   eg:dd-mm-yyyy           : 21213
Invalid input. Please double check your DOB input.
Enter Staff date of birth   eg:dd-mm-yyyy           : 20-03-2001
Enter Staff email address                           : 2
Invalid input. Please double check your email address input.
Enter Staff email address                           : |
```

If the users did not follow the format to enter the information, the system will present the error message to users.

```
Staff Password will be same the as staff ID for the first time registration: FT003


New Staff Details:
Staff ID:                           : FT003
Staff Password:                     : FT003
StaffName:                          : MINTECK
Staff IC:                           : 010293-21-2318
Staff Gender:                       : Male
Staff Contact Number:               : 018-8291921
Staff Email Address:                : minteck@gmail.com
Staff DOB:                          : 20/06/2001
Staff Type:                         : FullTime
Staff Account Status:               : Active
==================================================================================
Are you confirm to Save? y/n        : y
New fulltime staff record added successfully.
```

Staff ID	Staff Name	Staff Status
FT001	WEISHENG	Activate
FT002	JANNY	Activate
FT003	MINTECK	Active

The new Staff is created when users complete their staff information.

4. Update Staff Details

```
╔═══════════════════════════════════════════════════════════════
‖ Admin > Staff: Update Staff Details > FullTime Staff
╚═══════════════════════════════════════════════════════════════
What would you like to do?
[1] Update Staff Name
[2] Update Staff IC
[3] Update Staff Gender
[4] Update Staff Contact Number
[5] Update Staff Email Address
[6] Update Staff DOB
[7] Update Staff Account Status
[8] Update Staff Account Password
[9] Update Staff Type
[0] Return to Previous Menu
```

The system will provide 9 options for users to update the staff information.

```
Enter the Fulltime Staff ID in order to change the name (Q to Quit): FT001
Existing Staff Name is :WEISHENG


Enter New Staff Name: CHANGSHENG
Existing staff name WEISHENG will be updated to CHANGSHENG

Confirm? y/n: Y
Fulltime Staff Name Changed Successfully!
```

If users want to update the staff details can enter the input to perform the function . For example , if users want to change their staff name, they can press “1” as the input to change the name .

5. **Delete Staff Details**

```
Enter the Staff ID in order to delete the account
```

The system requires users to enter the staff ID to perform delete .

```
Enter the Staff ID in order to delete the account (Q to Quit): PT0005
Staff's ID number: PT0005 , JACELYN will be deleted.

Confirm to proceed? y/n: Y
Parttime Staff account successfully deleted!
```

Users enter the staff ID to delete the staff details .

6. **Generate Report**

```
Select type of report to be generated:
[1] Generate ALL
[2] Generate Fulltime Staff List Only
[3] Generate Parttime Staff List Only
[0] Return to Previous Menu
Choice                                          : 1

Staff Report generated successfully!
```

The system will provide 3 options for users to generate the staff report . The users can enter the input according to what they want to view . After the users enter their choice and the Staff reports will be saved as txt format in the Assignment file.

For option “1” :

ANAVRIN COLLEGE ENROLLMENT SYSTEM
STAFF LIST REPORT

Generated On: 2021-12-10 18:19:16

FullTime Staff

Staff ID	Staff Name	IC	Gender	Contact Number	Email Address	DOB	Account Status	Staff Type
FT001	CHANGSHENG	998120-01-3889	MALE	016-832-9901	weisheng@gmail.com	01-12-1998	Activate	FullTime
FT002	JANNY	002821-11-2814	FEMALE	018-289-1099	janny@gmail.com	21-08-2000	Activate	FullTime
FT003	MINTECK	010293-21-2318	Male	018-8291921	minteck@gmail.com	20/06/2001	Active	FullTime

PartTime Staff

Staff ID	Staff Name	IC	Gender	Contact Number	Email Address	DOB	Account Status	Staff Type
PT0001	ZICONG	010502-11-2844	Male	016-291-3092	zicong@gmail.com	2001-05-10	Activate	PartTime
PT0002	ZIJUN	010113-21-3011	Male	017-432-7833	zijun@gmail.com	2000-03-22	Activate	PartTime
PT0003	HUICHYI	010221-88-1033	Female	012-718-4912	huichyi@gmail.com	1999-10-02	Activate	PartTime
PT0005	JACELYN	010122-11-3411	Female	018-221-9931	jacelyn@gmail.com	2002-01-09	Activate	PartTime

Total: 7 staff

For option “2” :

ANAVRIN COLLEGE ENROLLMENT SYSTEM
STAFF LIST REPORT

Generated On: 2021-12-10 18:19:18

FullTime Staff

Staff ID	Staff Name	Staff IC	Gender	Contact Number	Email Address	DOB	Account Status	Staff Type
FT001	CHANGSHENG	998120-01-3889	MALE	016-832-9901	weisheng@gmail.com	01-12-1998	Activate	FullTime
FT002	JANNY	002821-11-2814	FEMALE	018-289-1099	janny@gmail.com	21-08-2000	Activate	FullTime
FT003	MINTECK	010293-21-2318	Male	018-8291921	minteck@gmail.com	20/06/2001	Active	FullTime

Total: 3 staff

For option “3” :

ANAVRIN COLLEGE ENROLLMENT SYSTEM
STAFF LIST REPORT

Generated On: 2021-12-10 18:19:20

PartTime Staff

Staff ID	Staff Name	Staff IC	Gender	Contact Number	Email Address	DOB	Account Status	Staff Type
PT0001	ZICONG	010502-11-2844	Male	016-291-3092	zicong@gmail.com	2001-05-10	Activate	PartTime
PT0002	ZIJUN	010113-21-3011	Male	017-432-7833	zijun@gmail.com	2000-03-22	Activate	PartTime
PT0003	HUICHYI	010221-88-1033	Female	012-718-4912	huichyi@gmail.com	1999-10-02	Activate	PartTime
PT0005	JACELYN	010122-11-3411	Female	018-221-9931	jacelyn@gmail.com	2002-01-09	Activate	PartTime

Total: 4 staff

3.	FONG ZHI JUN

Staff

1. **Student Module Menu**

Staff will be greeted by 7 options upon successfully logged in from the admin main menu.

```
╔═══════════════════════════════════════════════════════════════════════════════
‖ Admin Main > Student                                                         ‖
╚═══════════════════════════════════════════════════════════════════════════════

What would you like to do?
[1] View Student Details
[2] Add New Student Details
[3] Update Student Details
[4] Delete Student Entry
[5] Send Student Login Credentials Through Email
[6] Generate Student Reports
[7] Generate Student Graph
[9] Return to Previous Menu
Choice                                        :
```

If options other than 1 to 9, symbols and characters are entered, the system will prompt and guide staff to enter a correct input.

```
[9] Return to Previous Menu
Choice                                        : 10
Please enter valid option.
```

```
[9] Return to Previous Menu
Choice                                        : v
Incorrect input. Please enter digits 1 - 9 only.
```

2. **View Student Details**

This function can be entered through choosing the first option. Staff can notice a header: View student details in the above to keep track of which section he / she is currently located at right now. Total student counts will be displayed along with the student ID, name and account status.

```
Admin > Student: View Student Details

Student ID          Student Name                          Student Status

Local Student List

20WBR77777          ELAINE MAY                            Active

22WMR88888          CHEAH CHENG LONG                      Active

19WMR00050          XIAO CHENG LEI                        Inactive

International Student List

20IWBR22222         JOHN WICK                             Active

21IWMR00001         CHARLIE DOE                           Active

22IWMR12345         BILLIE EILISH                         Inactive

Total Student Count: 6

Enter the Student ID in order to view the student details (Q to Quit):
```

If the student ID entered does not exist or incorrect student ID sequence, the system will prompt the student record not found and double check the input. Else, entering: Q will quit the section and go back to the student module menu.

```
Enter the Student ID in order to view the student details (Q to Quit): 20WMR77711
Student Record not found. Please double check the input.
```

3. Add Student Details

Header will display which section the staff is in currently, retrieve existing data from the database and display the student's ID, name and account status regardless of local or international student names. Then, the staff can choose which type of student staff intend to create.

```
Admin > Student: Type of Student

Student ID          Student Name                          Student Status

Local Student List
```

```
Enter student type:
[1] Local Student
[2] International Student
Enter choice (Q to quit)                    :
```

If staff mistakenly intend to create existing student ID, the system will prompt the student ID already existed.

```
Enter new student ID Eg:21WMR00001 (Q to quit)   : 20WBR77777
StudentID 20WBR77777 already exist!
```

Same applies to incorrect student ID sequences.

```
Enter new student ID Eg:21WMR00001 (Q to quit)   : aa123qwe
Invalid input. Student ID has 2 digits, followed by 3 alphabets and 5 digits, eg = 21WMR00001
```

Various input validations in names, IC, emails and so on is as below.

```
Enter new student ID Eg:21WMR00001 (Q to quit)   : 21WMR66666
Enter Student name                               : Ali 1
Invalid input. No symbols or digits is allowed.
Enter Student name                               : Ali
Enter Student IC     eg:999999-99-9999           : 789451x-44
Invalid input. Please check your input. eg:999999-99-9999.
Enter Student IC     eg:999999-99-9999           : 021010-45-7898
```

```
Gender :
                [1] Male
                [2] Female
Choice:                                          : s
Incorrect input. Please enter digits 1 - 2 only.
Choice:                                          : 1
Enter student contact number eg: 011-9999999     : 017a-45641
Invalid input. Please double check your number input.
Enter student contact number eg: 011-9999999     : 017-1234567
Enter Student date of birth   eg:dd/mm/yyyy      : 10-10-2000
Enter Student email address                      : testing.com
Invalid input. Please double check your email address input.
Enter Student email address                      : testing@gmail.com
```

After the staff finished inputting all the new student details and keen to save the inputted student details, the student passwords will be encrypted into an unreadable string for security purposes, then saved into the database.

password
HSAgZhz6RjM92LIrYFC0wA==
wvyS5Z46TjLzAsAKXxozfQ==
2B0TJOWhPLzjAZUYkivhcA==

4. Update Student Details

The header will print out the “update student details” and staff can pick from 1 to 10 to update respective student details. Input validations from previous section: adding student details are also applied in this section.

```
╔═══════════════════════════════════════════════════════════════════
 ‖ Admin > Student: Update Student Details > Local Student
 ╚═══════════════════════════════════════════════════════════════════

What would you like to do?
[1] Update Student Name
[2] Update Student IC
[3] Update Student Gender
[4] Update Student Contact Number
[5] Update Student Email Address
[6] Update Student DOB
[7] Update Student Programme
[8] Update Student Account Status
[9] Update Student Account Password
[10] Update Student Study Mode
[0] Return to Previous Menu
Choice                                          :
```

5. Delete Student Details

A tips and delete confirmation message will be prompted for the staff before attempting to delete any student account.

```
                                        Tips:
The delete process is INRREVERSIBLE, please proceed with caution.
You can update the student account status to inactive instead of deleting it.


Enter the Student ID in order to delete the account (Q to Quit):
```

```
Enter the Student ID in order to delete the account (Q to Quit): 20WBR77777
Student's ID number: 20WBR77777 , ELAINE MAY will be deleted.

Confirm to proceed? y/n:
```

6. Send Student Login Credentials Through Email

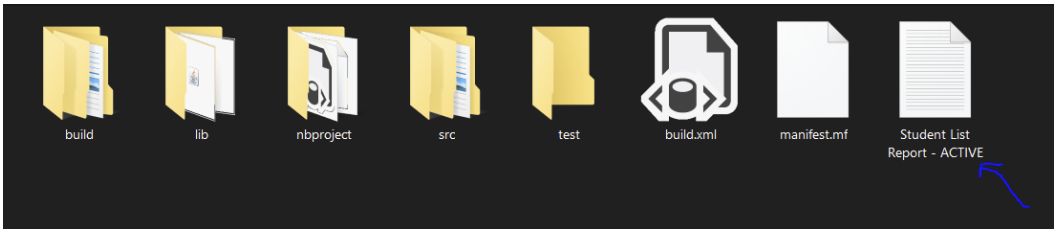
Staff will be prompted to enter the correct student ID in order to send student login credentials to that particular student.

```
Total Student Count: 6

Enter the Student ID to proceed sending the Student Login Credentials email (Q to Quit): 22WMR88888
Local Student :CHEAH CHENG LONG's email address is fongzhijun@gmail.com

Press Y to proceed (Q to quit):
```

7. Generate Student Report

Staff can pick from 1 to 3 of the report menu to generate desired reports. For example, the first option is to generate all students from both local and international students, second option is to generate active students only and third is for inactive students only. Successfully generated text files will be stored inside the source file folder.

```
Select type of report to be generated:
[1] Generate ALL
[2] Generate Active Students Account List Only
[3] Generate Inactive Students Account List Only
[0] Return to Previous Menu
Choice                                          : 2

Student Report generated successfully!

Choice                                          :
```



Student List Report - ACTIVE - Notepad

File Edit Format View Help

ANAVRIN COLLEGE ENROLLMENT SYSTEM
STUDENT LIST REPORT

Generated On: 2021-12-07 13:50:19

Local Students

Student ID	Student Name	Student IC	Gender	Contact Number	Email Address	DOB	Programme	Account Status
20WBR77777	ELAINE MAY	991010-78-6541	Female	018-7894562	elaine@hotmail.com	10/10/1999	Active	REI
22WMR88888	CHEAH CHENG LONG	001010-08-7894	Male	012-1234561	fongzhijun@gmail.com	10/10/2000	Active	RMM

International Students

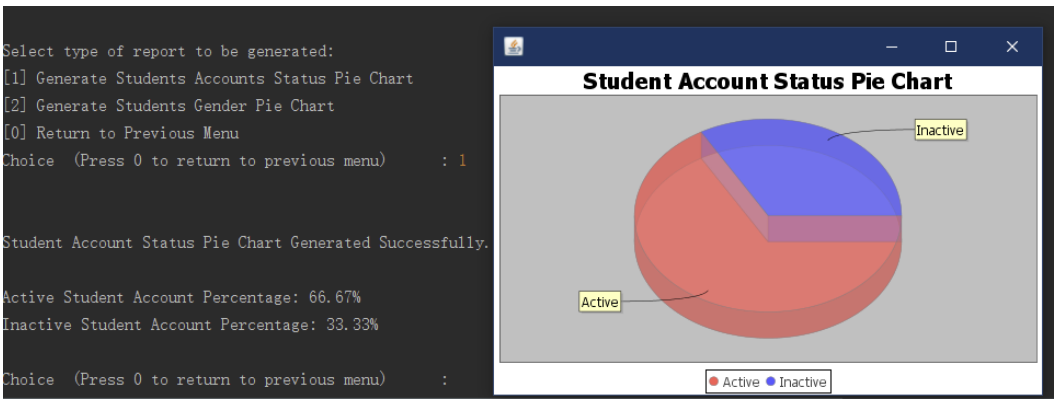
Student ID	Student Name	Student IC	Gender	Contact Number	Email Address	DOB	Programme	Account Status
20IWBR22222	JOHN WICK	A78945612	Male	018-7894562	fongzhijun@gmail.com	20/07/1998	Active	REI
21IWMR00001	CHARLIE DOE	Z12345678	Male	012-7894561	charliedoe@hotmail.com	10/10/1999	Active	RMM

Total: 4 students

8. Generate Student Pie Chart

Staff can choose which type of pie chart to be displayed by entering digit 1 or 2, then a java program will pop out and display the pie chart.

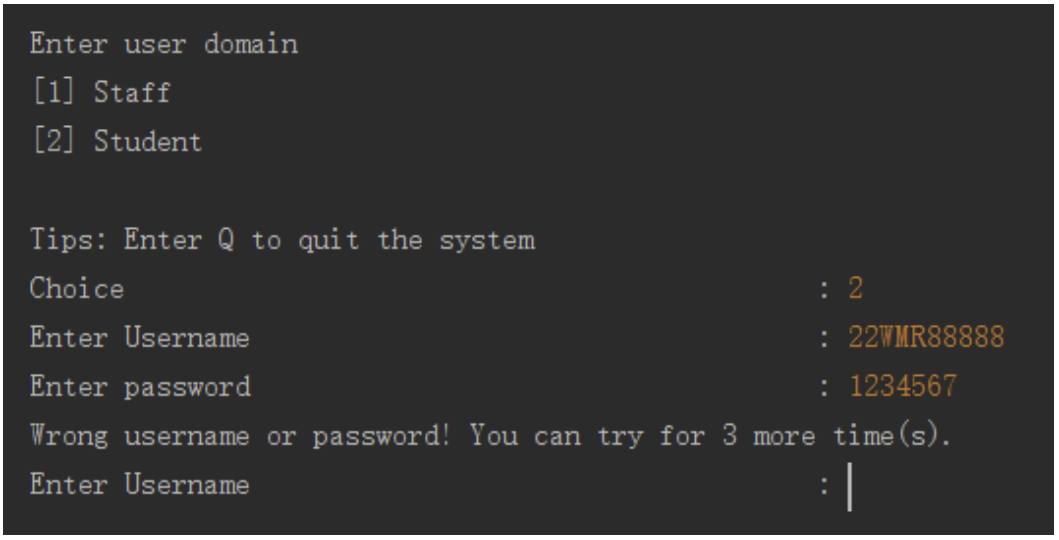
```
╔═══════════════════════════════════════════════════════════════════════════
 ‖ Admin > Student: Generate Student Pie Chart
 ╚═══════════════════════════════════════════════════════════════════════════

Select type of report to be generated:
[1] Generate Students Accounts Status Pie Chart
[2] Generate Students Gender Pie Chart
[0] Return to Previous Menu
Choice  (Press 0 to return to previous menu)      :
```



<u>Student</u>

1. **Log in Menu**

Students will be required to press select 2 to log in as student. If the student username and password don't match, the system will prompt the wrong username and password message. Otherwise, if logged in successfully, the student will be greeted with the student menu view with the student name stated and able to select those functions.



```
╔═══════════════════════════════════════════════════════════
║ Student Main
╙═══════════════════════════════════════════════════════════

Welcome CHEAH CHENG LONG

What would you like to do?
[1] Enroll Into A Course
[2] Drop From A Course
[3] Show Courses Enrolled
[4] Show Timetable
[5] Change Your Password
[6] Logout
Choice                                          :
```

4. DESMOND ONG KHAI YANG

1. <u>Student Enrolment module</u>

```
                          Welcome to

                _   _     __      _______  _____ _   _
          /\   | \ | |   /\ \    / /  __ \|_   _| \ | |
         /  \  |  \| |  /  \ \  / /| |__) | | | |  \| |
        / /\ \ | . ` | / /\ \ \/ / |  _  /  | | | . ` |
       / ____ \| |\  |/ ____ \  /  | | \ \ _| |_| |\  |
      /_/    \_\_| \_/_/    \_\/   |_|  \_\_____|_| \_|

            Anavrin College Enrollment System
```

```
Student Main
```

```
Welcome DESMOND ONG KHAI YANG

What would you like to do?
[1] Enroll Into A Course
[2] Drop From A Course
[3] Show Courses Enrolled
[4] Change Your Password
[6] Logout
Choice                                          :
```

For the student module, students are presented with 3 options, the enrol option, drop option, and show courses option. And the fourth option is done by Zhi Jun which allows the student to change their password. Student can choose from option 1-4 which

corresponds to their individual module.

2. Enrolment Module

```
                                Welcome to

                 _   _    __      _______  _____ _   _
              /\  | \ | |  /\ \    / /  __ \|_   _| \ | |
             /  \ |  \| | /  \ \  / /| |__) | | | |  \| |
            / /\ \| . ` |/ /\ \ \/ / |  _  /  | | | . ` |
           / ____ \| |\  |/ ____ \  /  | | \ \ _| |_| |\  |
          /_/    \_\_| \_/_/    \_\/   |_|  \_\_____|_| \_|

                 Anavrin College Enrollment System
```

```
Student > Enrol
```

```
What would you like to Enrol in? :
[1] Curriculum
[2] Cocurriculum
[0] Exit
=====================================================
Choice                                              :
```

In the enrollment module, students will be prompted to choose between enrolling in curriculum courses or join co curricular activity. Each option is represented by a number that the user can input to select. The number 0 is designated for exit.

```
Student > Enrol
```

```
What would you like to Enrol in? :
[1] Curriculum
[2] Cocurriculum
[0] Exit
=====================================================
Choice                                              :a
Please select a valid option
What would you like to Enrol in? :
[1] Curriculum
[2] Cocurriculum
[0] Exit
=====================================================
Choice                                              :9
Please select a valid option
What would you like to Enrol in? :
[1] Curriculum
[2] Cocurriculum
[0] Exit
=====================================================
Choice                                              :#
Please select a valid option
What would you like to Enrol in? :
[1] Curriculum
[2] Cocurriculum
[0] Exit
=====================================================
Choice                                              :
```

If the user enters invalid input, the system will prompt the user to input the correct options

```
                                Welcome to

                 _   _     __      _______  _____ _   _
                /\   | \ | |   /\ \    / /  __ \|_   _| \ | |
               /  \  |  \| |  /  \ \  / /| |__) | | | |  \| |
              / /\ \ | . ` | / /\ \ \/ / |  _  /  | | | . ` |
             / ____ \| |\  |/ ____ \  /  | | \ \ _| |_| |\  |
            /_/    \_\_| \_/_/    \_\/   |_|  \_\_____|_| \_|

                    Anavrin College Enrollment System

 Student > Enrol > curriculum

Course Code ||Course Name                    ||Description
   BACS2023 ||OBJECT-ORIENTED PROGRAMMING    ||OO IN JAVA
   BAIT1083 ||VISUAL PROGRAMMING             ||DEVELOPING WINDOWS-BASED APPLICATIONS
   BAIT2004 ||FUNDAMENTAL COMPUTER NETWORKS  ||NETACAD CISCO FOR BEGINNER
   BAMS1623 ||DISCRETE MATHEMATICS            ||IMPROVE PROGRAMMING SOLVING ABILITY
   BMIT2164 ||COMPUTER NETWORKS              ||NETACAD CISCO
   BSEX4269 ||SEX EDUCATION                  ||PRACTICE SAFE SEX, CHILD CONTROL
Select a course [Course Code] [Q to exit]:
```

In the curricular course menu, a list of all the courses will be displayed to the user to choose from. The user can choose to input the course code of the course to enroll, or they can input Q to return/exit.

```
Course Code ||Course Name                    ||Description
   BACS2023 ||OBJECT-ORIENTED PROGRAMMING    ||OO IN JAVA
   BAIT1083 ||VISUAL PROGRAMMING             ||DEVELOPING WINDOWS-BASED APPLICATIONS
   BAIT2004 ||FUNDAMENTAL COMPUTER NETWORKS  ||NETACAD CISCO FOR BEGINNER
   BAMS1623 ||DISCRETE MATHEMATICS            ||IMPROVE PROGRAMMING SOLVING ABILITY
   BMIT2164 ||COMPUTER NETWORKS              ||NETACAD CISCO
   BSEX4269 ||SEX EDUCATION                  ||PRACTICE SAFE SEX, CHILD CONTROL
Select a course [Course Code] [Q to exit]: BAIT1083
you are already enrolled in this course
Select a course [Course Code] [Q to exit]: abcde
Please Select a valid Course
Select a course [Course Code] [Q to exit]: BAMS2164
Please Select a valid Course
Select a course [Course Code] [Q to exit]: BAMS1623
Course unavailable
Select a course [Course Code] [Q to exit]:
```

```
Select a course [Course Code] [Q to exit]: BACS2023
Course is available,Press <Enter> to continue...
```

There are multiple validations done to prevent unwanted input. The first one is if the user has already enrolled in the course, they can't enrol in the course again. Secondly, any unrecognized input will also be disregarded. Lastly, a Course that has its status set to Unavailable cannot be enrolled into.
If a course is available, the user can continue to payment module..

```
                              Welcome to

                 _   _     __      _______  _____ _   _
                /\  | \ |  /\ \    / /  __ \|_   _| \ | |
               /  \ |  \| | /  \ \  / /| |__) | | | |  \| |
              / /\ \| . ` | / /\ \ \/ / |  _  /  | | | . ` |
             / ____ \| |\  |/ ____ \  /  | | \ \ _| |_| |\  |
            /_/    \_\_| \_/_/    \_\/   |_|  \_\_____|_| \_|

                    Anavrin College Enrollment System
```

```
Student > Enrol > Enrol Cocurriculum
```

```
Curriculum ||Course Name                    ||Description
   COCU0001 ||XIANGQI / GO                   ||CHINESE ELEPHANT CHESS
   COCU0002 ||WEIQI                          ||CHINESE ENCIRCLING GAME
   COCU0003 ||BADMINTON                      ||FAST-PACES RACKET SPORT
   COCU0005 ||TAEKWONDO                      ||KOREAN TRADITIONAL SELF-DEFENSE ART
   COCU0006 ||CHINESE CALLIGRAPHY            ||CHINESE ANCIENT ART FORM WRITING
   COCU0007 ||FIRST AID UNIT                 ||SOCIETY THAT PROVIDE FIRST AID SERVICE
   COCU0008 ||COMMUNITY SOCIOLOGY            ||SHARE LOVE WITH OTHERS AND HELP THOSE IN
   COCU0009 ||KARATE-DO                      ||MARTIAL ART PECULIAR

Please Select a Cocuriculum activity:
```

On the other hand, users can also choose to enroll in co curricular courses. Like enrolling in a curriculum, the user can enrol using the course code .

But if the user has already enrolled in a curricular course he/she won't be allowed to access this page until they finish or dropped their course.

```
Student > Enrol > Enrol Cocurriculum
```

```
Curriculum ||Course Name                    ||Description
   COCU0001 ||XIANGQI / GO                   ||CHINESE ELEPHANT CHESS
   COCU0002 ||WEIQI                          ||CHINESE ENCIRCLING GAME
   COCU0003 ||BADMINTON                      ||FAST-PACES RACKET SPORT
   COCU0005 ||TAEKWONDO                      ||KOREAN TRADITIONAL SELF-DEFE
   COCU0006 ||CHINESE CALLIGRAPHY            ||CHINESE ANCIENT ART FORM WRI
   COCU0007 ||FIRST AID UNIT                 ||SOCIETY THAT PROVIDE FIRST A
   COCU0008 ||COMMUNITY SOCIOLOGY            ||SHARE LOVE WITH OTHERS AND H
   COCU0009 ||KARATE-DO                      ||MARTIAL ART PECULIAR


Please Select a Cocuriculum activity:
Please select a valid cocu Activity
Please Select a Cocuriculum activity: 10
Please select a valid cocu Activity
Please Select a Cocuriculum activity: Cocu0008
Done!Press <Enter> to continue...
```

Validations are also done to prevent unwanted input that would jeopardize the whole system.

2. Payment

```
                          Welcome to

                _   _    __      ______  _____ _   _
           /\   | \ |  |  /\ \    / /  __ \|_   _| \ | |
          /  \  |  \| | /  \ \  / /| |__) | | | |  \| |
         / /\ \ | . ` | / /\ \ \/ / |  _  /  | | | . ` |
        / ____ \| |\  |/ ____ \  /  | | \ \ _| |_| |\  |
       /_/    \_\_| \_/_/    \_\/   |_|  \_\_____|_| \_|

              Anavrin College Enrollment System
```

```
Student > Enrol > Payment
```

```
Please select payment method
[1] cash
[2] card
Choice                                          :
```

After the student has enrolled in a course be it co curricular or main course, they will be shown a payment screen where they can make their payment via cash or credit. As usual standard validation are enforced here.

```
          Tuition Payment
================================
              BMIT2164     1036.00


Enter payment amount: 500
Not enough payment...
Enter payment amount: a
please enter numerals.
Enter payment amount: 1000
Not enough payment...
Enter payment amount: 2000.00
Payment Successful!
```

If the user selects cash, they will be presented with a simple bill detailings their expenses, they choose to input the amount they are going to pay. If the pay is insufficient they will have to add more money until there is enough. Invalid characters are also prevented from the system.

```
Please select payment method
[1] cash
[2] card
Choice                                                  :2
        Tuition Payment
================================
            BSEX4269      69.00


Enter card Number: asdsadab
Please Enter valid card no (10 digit)..
12202927101
Please Enter valid card no (10 digit)..
1020291782
Payment Successful!Press <Enter> to continue...
```

If payment is done using a card, credit card numbers are required. Standard validation follows suit.

3. Invoice generation

```
========================
INVOICE
              2021-12-13
========================
ITEM                FEEs
====                ====
BMIT2164         1036.00



========================
Paid             2000.00
Method           Cash
Change           964.00
Press <Enter> to continue...

save Invoice to file? (y/n):
y
Saved to file.
Payment record added successfully.
Done! Press <Enter> to continue...
```

An invoice is generated upon successful payment. The invoice details include item name, its price , the amount paid, payment method, and the balance as well as the date.
When the invoice is generated it will prompt the wether they want to save the invoice as a file. If the answer is “Y” a file will be generated that includes the invoice.

Invoice generated

4. View Course module

```
                              Welcome to

                 _   _     __     _______  _____ _   _
                /\   | \ | |   /\ \    / /  __ \|_   _| \ | |
               /  \  |  \| |  /  \ \  / /| |__) | | | |  \| |
              / /\ \ | . ` | / /\ \ \/ / |  _  /  | | | . ` |
             / ____ \| |\  |/ ____ \  /  | | \ \ _| |_| |\  |
            /_/    \_\_| \_/_/    \_\/   |_|  \_\_____|_| \_|

                    Anavrin College Enrollment System

 Student > Show Course Enrolled

==================================================================================
STUDENT ID : 20WMR12046
==================================================================================
Main Curriculum
==================================================================================
BACS2023 || OBJECT-ORIENTED PROGRAMMING    || OO IN JAVA
BAIT2004 || FUNDAMENTAL COMPUTER NETWORKS  || NETACAD CISCO FOR BEGINNER
BAIT1083 || VISUAL PROGRAMMING             || DEVELOPING WINDOWS-BASED APPLICATIONS
BMIT2164 || COMPUTER NETWORKS              || NETACAD CISCO
BSEX4269 || SEX EDUCATION                  || PRACTICE SAFE SEX, CHILD CONTROL
==================================================================================
Cocurriculum
==================================================================================
COCU0008 || COMMUNITY SOCIOLOGY            || SHARE LOVE WITH OTHERS AND HELP THOSE IN NEED
Press <Enter> to continue...
```

In this module, user can view their currently enrolled courses, and its details.

5. Drop Course Module

```
                              Welcome to

                 _   _     __     _______  _____ _   _
                /\   | \ | |   /\ \    / /  __ \|_   _| \ | |
               /  \  |  \| |  /  \ \  / /| |__) | | | |  \| |
              / /\ \ | . ` | / /\ \ \/ / |  _  /  | | | . ` |
             / ____ \| |\  |/ ____ \  /  | | \ \ _| |_| |\  |
            /_/    \_\_| \_/_/    \_\/   |_|  \_\_____|_| \_|

                    Anavrin College Enrollment System

 Student > DROP COURSE

Select courses to drop
[1] Curriculum
[2] Cocurriculum
[0] Back
Choice                                    : 1
```

For this module, students can choose to drop their course on either curricular or co curricular courses.

```
                              Welcome to

                 _   _    __      _______  _____ _   _
                /\  | \ | |  /\ \    / /  __ \|_   _| \ | |
               /  \ |  \| | /  \ \  / /| |__) | | | |  \| |
              / /\ \ | . ` | / /\ \ \/ / |  _  /  | | | . ` |
             / ____ \| |\  |/ ____ \  /  | | \ \ _| |_| |\  |
            /_/    \_\_| \_/_/    \_\/   |_|  \_\_____|_| \_|

                   Anavrin College Enrollment System
```

```
Student > Drop-A-Course> Drop curriculum
```

```
WELCOME TO DROP-A-COURSE (Curriculum)!
Enter 'i'  to  view your current enrolled courses.
Enter 'q'  to  exit.
Enter the course code of the course you wish to drop:i
```

The user can drop from their course by entering the course code. Furthermore the user can quick access view the view enrolled course module for referencing by inputting the letter "i" .

```
==========================================================================================
STUDENT ID : 20WMR12046
==========================================================================================
Main Curriculum
==========================================================================================
BAIT2004 || FUNDAMENTAL COMPUTER NETWORKS  || NETACAD CISCO FOR BEGINNER
BAIT1083 || VISUAL PROGRAMMING             || DEVELOPING WINDOWS-BASED APPLICATIONS
BMIT2164 || COMPUTER NETWORKS              || NETACAD CISCO
BSEX4269 || SEX EDUCATION                  || PRACTICE SAFE SEX, CHILD CONTROL
==========================================================================================
Cocurriculum
==========================================================================================
COCU0008 || COMMUNITY SOCIOLOGY            || SHARE LOVE WITH OTHERS AND HELP THOSE I
Press <Enter> to continue...

Enter the course code of the course you wish to drop:Bcs1234
You are not enrolled in this course.
Enter the course code of the course you wish to drop:1234567
You are not enrolled in this course.
Enter the course code of the course you wish to drop:BMIT2164
you are already enrolled in this course
Course Removed
Done! Press <Enter> to continue...
```

With reference to the table, the student can safely drop out of their courses if they wish to do so. Validation are also enforced for safer input. If they wish to drop before they are enrolled, they will not be permitted.

```
                              Welcome to

                 _   _     __      _______  _____ _   _
                /\   | \ | |   /\ \    / /  __ \|_   _| \ | |
               /  \  |  \| |  /  \ \  / /| |__) | | | |  \| |
              / /\ \ | . ` | / /\ \ \/ / |  _  /  | | | . ` |
             / ____ \| |\  |/ ____ \  /  | | \ \ _| |_| |\  |
            /_/    \_\_| \_/_/    \_\/   |_|  \_\_____|_| \_|

                   Anavrin College Enrollment System
```

```
Student > Drop-A-Course> Drop Cocurriculum
```

```
WELCOME TO DROP-A-COURSE (Cocurriculum)!
[1] drop from your current cocu activity
[2] Exit
1
Done!Press <Enter> to continue...
```

Since there is one co curricular course for each student they can only drop from one course activity. If they wish to drop before they are enrolled, they will not be permitted.

5. Concepts of Object-Oriented Programming Applied

No.	Team Member	Concepts Applied
1.	GUI HUI CHYI	

1. Inheritance

This concept is shown by the Curriculum and Co-curriculum classes.

Since both the classes have common attributes and functions like course code, course name, description, credit hours, fee, and status, as well as getter and setter methods, having a common parent class prevents duplication of these attributes and methods. Thus, these similar attributes between Curriculum and Co-curriculum are declared under the Course class.

```
public abstract class Course {
    private String courseCode;
    private String courseName;
    private String description;
    private int creditHours;
    private double fee;
    private String status;
```

Both Curriculum and Co-curriculum classes extend the User class and inherit all the data members as well as getter and setter methods from Course class. All data members, getter, and setter methods from Course class can be accessible by Curriculum and Co-curriculum classes. They also define sets of their own attributes in their own class. For example, Co-curriculum class has a data member of Category and methods such as getCategory and setCategory.

One thing to mention is that Course class is also declared as an abstraction class so that no instance of Course class can be created.

- **Course (Curriculum/ Co-curriculum)**

By using the extends keyword, Curriculum and Co-curriculum become a child class to the Course class and it will inherit all the data members and methods of the Course class.

```
public class Curriculum extends Course
public class Cocurriculum extends Course
```

Example (Co-curriculum):

```
public class Cocurriculum extends Course {

    private String category;
```

```java
public String getCategory() {
    return this.category;
}

public void setCategory(String category) {
    this.category = category;
}
```

The constructors of Course class must be also assigned with a **protected modifier** keyword in order for Curriculum and Co-curriculum to **inherit** or **reuse** the getter and setter methods in Course class. By doing so, only subclasses such as Curriculum and Co-curriculum can use the method and variable, other non-related classes are prevented from trying to use it.

```java
protected Course() {
}
protected Course(String courseName, String description, double fee) {
    this.courseName = courseName;
    this.description = description;
    this.fee = fee;
    this.creditHours = 2;
    this.status = "AVAILABLE";
}
protected Course(String courseCode, String courseName, String description, int creditHours, double fee) {
    this.courseCode = courseCode;
    this.courseName = courseName;
    this.description = description;
    this.creditHours = creditHours;
    this.fee = fee;
    this.status = "AVAILABLE";
}
protected Course(String courseCode, String courseName, String description, int creditHours, double fee, String status)
    this.courseCode = courseCode;
    this.courseName = courseName;
    this.description = description;
    this.creditHours = creditHours;
    this.fee = fee;
    this.status = status;
}
```

A **super** keyword is used in the Curriculum and Co-curriculum classes constructor to call the parent class constructor (Course) while creating a Curriculum/Co-curriculum object. The constructor is **protected**, so the subclass can access it.

```java
public Curriculum(String courseCode, String courseName, String description, int creditHours, double fee) {
    super(courseCode, courseName, description, creditHours, fee);
}

public Curriculum(String courseCode, String courseName, String description, int creditHours, double fee, String status) {
    super(courseCode, courseName, description, creditHours, fee, status);
}

public Cocurriculum(String courseName, String description, double fee, String category) {
    super(courseName, description, fee);
    this.category = category;
}

public Cocurriculum(String courseCode, String courseName, String description, int creditHours, String category, double fee, String status) {
    super(courseCode, courseName, description, creditHours, fee, status);
    this.category = category;
}
```

2. Polymorphism

- **Method Overloading**

This concept allows overloading of methods i.e. two methods with the same name but different parameters will behave differently. In our case, we have implemented this at several places, one of them being – printCourseList(String, String) and printCourseList(String). In the first case, the system will print the course of a particular

course status (Available/Unavailable), hence two String parameters are required. The first String parameter is to take in the course type (Curriculum/Co-curriculum), while the second String parameter is to pass in the course status (Available/Unavailable). On the other hand, the latter function displays the course list for all courses under the same course type (Curriculum/Co-curriculum), so no status is needed.

```
private void printCourseList(String type, String status) {...43 lines }

private void printCourseList(String type) {...41 lines }
```

Now, when I call one of these methods, the provided set of parameters identifies the method which has to be called.

In the following code snippet, I call the method only with a String. At the compile time, the Java compiler binds this method call to the printCourseList(String type) method.

```
case "1":
    printCourseList("Curriculum");
    System.out.print("Do you want to export data into Excel? y/n: ");
```

Now, I call the printCourseList method with two String parameters. The compiler binds the method call to the other printCourseList(String type, String status) method.

```
case "2":
    printCourseList("Curriculum", "AVAILABLE");
    System.out.print("Do you want to export data into Excel? y/n: ");
```

- **Method Overriding**

Polymorphism allows us to manipulate or override the methods of parent class in a child class. For example, the Course class is the superclass of the Co-curriculum class. Both classes provide an implementation of the toString() method. The Course class only allows the display of course code, course name, description, credit hours, fee and status. The toString method of Co-curriculum adds another display for category. Hence, the toString() methods of Co-curriculum will override the toString() method of Course class.

Example (Course class toString() method)

```
@Override
public String toString() {
    return String.format("%12.12s || %-30s || %-70s || %-12s || %-15.2f || %-12s",
            courseCode, courseName, description, creditHours, fee, status);
}
```

Example (Co-curriculum class toString() method)

```
@Override
public String toString() {
    return String.format("%12.12s || %-30s || %-50s || %-20s || %-12s || %-15.2f || %-12s",
            getCourseCode(), getCourseName(), getDescription(), category, getCreditHours(), getFee(), getStatus());
}
```

As a result, when Co-curriculum class's toString() method is called, the output will be different from Course class's toString() methods even though Co-curriculum class has inherited Course class's toString() method.

3. Encapsulation

Encapsulation is used to keep values and inner workings of an object hidden within the class and prevent illegal direct access by other classes. In our program, we had shown encapsulation in our program where all class data is declared as private.

Other classes can only access the data through the public method of the class, the so-called getter and setter methods. The method performs the function and returns the result. This way, the internal functioning is hidden from the class which called the method ensuring data protection.

For example, for a Co-curriculum entity, to obtain a course category, we would have to use getCategory() to retrieve it. To change a particular cocuriculum's category, we need to use the setCategory(String).

Example (Co-curriculum):

The Co-curriculum class uses a **private** attribute to store the information about the course category. The access modifier **private** makes the **category** attribute inaccessible for other classes within the same or other package.

```
public class Cocurriculum extends Course {

    private String category;
```

To get the information of the category, we need to call one of the **public** methods. The get method, getCategory() provides read access to the **category** attribute. The getCategory method will return the value of the variable category after being called.

```
public String getCategory() {
    return this.category;
}
```

To change the information of the category, a setter method is needed. The set method, setCategory(String) takes a String parameter (**category**) and assigns it to the **category** variable. The **this** keyword is used to refer to the current object.

```
public void setCategory(String category) {
    this.category = category;
}
```

2.	TEOH ZI CONG

1. **<u>Inheritance</u>**

The inheritance is shown by the Staff . As you can see the code is using the extend code to perform the inheritance such as FulltimeStaff extends Staff and the ParttimeStaff extends Staff. Inheritance is a concept that acquires the properties from parent class to child class to let the class inherit attributes and methods from another class.

Example 1 :

```
public class FulltimeStaff extends Staff{

 public class ParttimeStaff extends Staff{

abstract class Staff extends Person{

    protected String staffID;
    protected String staffPW;
    protected String staffType;
```

FulltimeStaff and the ParttimeStaff classes inherit from their parent class which is Staff and Person class. Use protected for the code is to prevent other classes except the parent class and child class from using this methods .

Example 2 :

```
protected static String selectStaffType() {
```

The inheritance allows the user to reuse the parent methods through invoking those methods. So the FulltimeStaff and the ParttimeStaff class can invoke the selectStaffType methods from the superclass to put the same input inside the different class .

2. **<u>Encapsulation</u>**

Encapsulation is the process of encapsulating data into a single unit. It's the mechanism that connects code and the data it works with. Encapsulation can also be thought of as a protective shield that prevents data from being accessed by code outside the shield.

Example 1:

```
public void setStaffType(String staffType){
    this.staffType = staffType;
}

public String getStaffType(){
    return staffType;
}
```

The getmethod , getStaffType() is to provide access to the StaffType attribute and this method will return the value of the StaffType after being called. And the setter method such as setStaffType is to change the information of the staffType.

3. Polymorphism

Polymorphism in Java refers to an object's ability to take multiple forms. To put it another way, polymorphism in Java allows us to perform the same action in multiple ways. Polymorphism is a feature of Java's object-oriented programming language that allows a single task to be completed in multiple ways.

Example 1 :

```java
private void viewStaff() {

private void viewStaff(String type) {

private void addFulltimeStaff(Assignment mainApp) {
    String staffName;
    String staffIC;
    String staffGender;
    String staffContactNumber;
    String staffEmailAddress;
    String staffDOB;
    String staffStatus = "Active";
    String staffPW;
    String staffType;
    String[] type ={"FullTime" , "PartTime"};

    Helper.printHeader("Admin > Staff: Add Fulltime Staff Details");
    while (true) {
        viewStaff("Fulltime");
```

The StaffService class has two methods , the first is there are no parameters and retrieves both parttime and fulltime staff account details for display . The second is takes a string of staff type like “Parttime” and “Fulltime” to retrieve the staff details.

3. FONG ZHI JUN

1. Encapsulation

Encapsulation is a mechanism of restricting direct access of the object variables or methods from external interference, and can be used to hide both data members and functions associated with an instantiated object. Usage of getters and setters will help link the data or variables with code that manipulates it and as a only means of access to prevent misuse of object or variable and reduce chances of bugs happening and protecting sensitive data.

Example 1:

```java
public class InternationalStudent extends Student{
    private String country;
```

```java
public String getCountry() {
    return country;
}
```

```java
public void setCountry(String country) {
    this.country = country;
}
```

For example, inside the InternationalStudent.java class has a private variable string Country that is meant for constructing the International Student object. With private visibility modifiers, other classes other than InternationalStudent cannot directly access it aside utilizing its setters and getters.

Example 2:

```java
private static final String ALGO = "AES";
private byte[] keyValue;
```

```java
private Key generateKey() throws Exception{
    Key key = new SecretKeySpec(keyValue, ALGO);
    return key;
}
```

```java
public String encrypt(String Data) throws Exception{
    Key key = generateKey();
    Cipher c = Cipher.getInstance(ALGO);
    c.init(Cipher.ENCRYPT_MODE, key);
    byte[] encVal = c.doFinal(Data.getBytes());
    String encryptedValue = new BASE64Encoder().encode(encVal);
    return encryptedValue;
}
```

With regards to the AES encryption method located inside Helper.java, it also has the private modifier: generateKey with the intention that only Helper class's encrypt method can access it and avoid methods from other classes tampering with the encryption and decryption key string. With encapsulation, it can protect sensitive data and hide data.

2. Inheritance

Inheritance is a mechanism that is like child classes inherits the attributes from the parent class by using the code: extends, which promotes code reusability of parent class method, adding unique properties besides adhering to the hierarchical classifications. Protected modifier also applies here as it permits its subclass methods to access into it only.

Example 1:

```java
public class LocalStudent extends Student{
```

```java
public class InternationalStudent extends Student{
```

```java
public LocalStudent(String studentID, String studentPW, String name, String IC, String gender, String contactNumber, String emailAddress, String DOB,
                    String accountStatus, String studentProgramme, String studyMode){
    super(studentID,studentPW, name, IC, gender, contactNumber, emailAddress, DOB, accountStatus, studentProgramme, studyMode);
}
```

```java
abstract class Student extends Person{
```

```java
protected String studentID;
protected String studentPW;
protected String studentProgramme;
protected String studyMode;
```

LocalStudent and InternationalStudent classes inherit all of the protected variables and methods from its parent class which is Student and Person abstract classes. Super keywords exist explicitly in order to invoke superclass constructor. Reason for these 2 classes is chosen to inherit is for better categorization of classes, a clear view of hierarchy classes as in Person and Student is general classes and these 2 subclasses will become more specific and concrete classes. Both subclasses can inherit those personal particulars like names, gender from parent class, but InternationalStudent class becomes unique by adding in a private variable: Country as we need to capture this data to differentiate it from local students. Inheritance as a powerful function will assist and make things easier that it can achieve through extends and super keywords and is flexible enough for us to add on extra variables.

Example 2:

```java
protected static String selectProgramme() {
```

```java
protected static String selectStudyMode() {
```

```
String newInput = LocalStudent.selectProgramme();
```

```
String newInput = InternationalStudent.selectProgramme();
```

Besides, inheritance will also allow us to reuse the parent methods through invoking those methods instead of duplicating the same method over and over again inside subclasses, making the code feel abundant. For instance, the LocalStudent and InternationalStudent subclasses can invoke the selectProgramme and selectStudyMode methods from the superclass instead of put duplicately inside the 2 subclasses.

3. Polymorphism

Polymorphism means "many forms", it allows the same name methods to operate differently by giving certain parameters.

<u>Example 1:</u>

```
private void viewStudent() {...41 lines }

private void viewStudent(String type) {...39 lines }
```

```
private void addLocalStudent(Assignment mainApp) {
    String studentName;
    String studentIC;
    String studentGender;
    String studentContactNumber;
    String studentEmailAddress;
    String studentDOB;
    String studentStatus = "Active";
    String studentPW;
    String studentProgramme;
    String[] programme = {"RMM", "RSD", "REI"};
    String studyMode;

    Helper.printHeader("Admin > Student: Add Local Student Details");
    while (true) {
        viewStudent("Local");
        Helper.printLine(59);
```

```
Admin > Student: Add Local Student Details

| Student ID      | Student Name                    | Student Status |
| 20WBR77777      | ELAINE MAY                      | Active         |
| 22WMR88888      | CHEAH CHENG LONG                | Active         |
| 19WMR00050      | XIAO CHENG LEI                  | Inactive       |

Total Student Count: 3

Enter new student ID Eg:21WMR00001 (Q to quit)  :
```

In StudentService class, it consists of this 2 methods which the first one is without any parameters is to retrieve both local and international student account details for display during admin CRUD operations, on the other hand the viewStudent method that will receive string of student type like either "Local" or "Inter" to retrieve local or international student details that will be displayed during specific course of actions like showing only local students details during update local student name and where international student details is not required.

<u>Example 2:</u>

```
public static void updateStudent(LocalStudent localStudent) {
```

```
public static void updateStudent(InternationalStudent internationalStudent) {
```

```
if (confirm.equals("Y")) {
    localStudent.setStudentProgramme(newInput);
    LocalStudent.updateStudent(localStudent);
    System.out.println("Local student's Programme changed successfully!");
```

```
if (confirm.equals("Y")) {
    internationalStudent.setStudentProgramme(newInput);
    InternationalStudent.updateStudent(internationalStudent);
    System.out.println("International student's Programme changed successfully!");
```

Polymorphism permits the same method name to be invoked throughout the class family hierarchy, with each same naming method inside each class can possess its own implementation and generate different outcomes. Above 2 pictures are regarding 2 methods with the same name but receiving different objects as parameters. First method exists under LocalStudent class while the other one is under InternationalStudent class. Both method purposes are used to update the student details with regards to either local or international student type. With the dynamic binding feature, JVM will search and invoke the first found method under the subclass and go up to the parent class until it is found. Hence the dynamic binding can help to invoke the correct method and update either local or international student details.

4.	DESMOND ONG KHAI YANG

1. Inheritance

   Inheritance are a unique feature of OOP, it allows reusability of codes in programming without having to rewrite it. Methods and attributes that belong to the parent class or superclass can be inherited to its subclasses like a parent passing its knowledge and genes to its children. In this case it's the subclasses that receives the knowledge and the similarity to its parent class.

```
public class CashPayment extends Payment{

public class CreditPayment extends Payment {

public CreditPayment(Double price, String item, String studentID) {
    super(price, item, studentID);
}

public CreditPayment(Double price, String item, String studentID) {
    super(price, item, studentID);
}
```

   Example 1

   For this example, I have 2 subclasses, CashPayment and CreditPayment that extend the parent class which is the abstract Payment class. The subclasses can fully inherit the method and attributes of the parent class. Using the super attribute we can initialize the attributes of the subclasses using the same constructor as the superclass

   Example 2

```
public static void addPaymentRecord(Payment payment) {
```

In this Example, inheritance is showcased with its ability to share method with its child classes. The classes are able to reuse method from the parent class without the need to rewrite the code that causes more work and confusion. In this instance, the method addPaymentRecord is invoked by both the subclasses but it is only written once.

2. Polymorphism
   Polymorphism allows the reuse of a method name to perform different functions sort of like a swiss army knife. It is one but come with many different function and form.

```
@Override
public void makePayment() {
```

   Example 1
   In this example, method overriding is considered one of the main method of polymorphism in java. By defining the same method as the parent class, this method is able to provide functions that are more catered to the class in ways such as paying with Cash is different from payment using Credit. As they require different features and functions.

3. Abstraction
   Abstraction is unique in OOP as it allows the body and class to be generalized and only provide the necessary information. It is useful to partially define a class that act as a guidelines towards other subclasses

```
abstract class Payment {
    protected String referenceNo;
    protected String studentID;
    protected LocalDate date;
    protected String item;
    protected double price;
    protected double paid;
    protected double balance;
    protected String remark;
```

   Example 1
   This abstract class is an example of abstraction. It provides the attributes and methods that its subclasses can follow and use. As a guideline without affecting other part of the program.

```
abstract void makePayment();
abstract void getInvoice();
```

   Example 2
   This abstract method makes it so that its child classes can use it to implement method that are more specialized to its specific requirements.

4. Encapsulation
   Encapsulation provides the ability for a program to have total control over its own field and variables and decides who gets to access what. And how to access its data. This allow sensitive field or method to be hidden from some while accessible by others.

```
public double getPrice() {
    return price;
}
public void setPrice(double price) {
    this.price = price;
}
        protected double price;
```

Example 1

In this example, the variable price is only accessible and modifiable by its subclasses. In order for classes and method outside of the class to access the modifier, we have provided control in the form of getter and setter which specify exactly how they can use the attributes