CZ2002: Object Oriented Design and Programming

Project Title: Course Registration IT System - NTU STARS Planner

Submission Date: 25 November 2020

Lab Group: Lab SS3, Assignment Group 3

Names of group members:

Name:	**Matriculation Number:**
Alvin Tan De Jun	U1922616C
Ernest Ang Cheng Han	U1921310H
Cai Xinrui	U1921389A
Ng Chi Hui	U1922243C
Rajkumar Snehaa	U1921000H

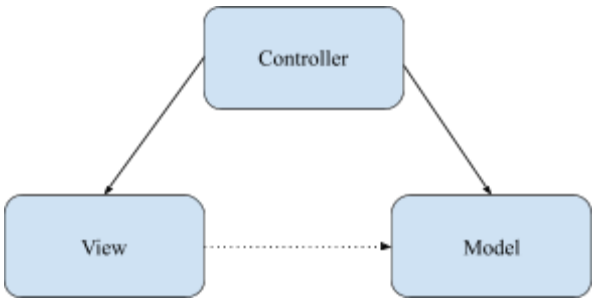
1. Introduction

For our project, we developed a course registration IT system based on Nanyang Technological University's **Student Automated Registration System Planner** (STARS Planner). This application is a console-based programme designed with the functions of allowing students to plan and register for their courses as well as allow the university staff to administer the entire course registration process. Please refer to the "README.txt" file for an explanation on how to run the program.

2. Design Considerations

2.1 Design Pattern

2.1.1 Model-View-Controller (MVC)



The architectural pattern that we applied in our project is **MVC**, where we separate the program into 3 components: model, view and controller (Ashish Shukla, 2014). Each component is built to handle specific development aspects of the application, allowing us to achieve **high cohesion** and **low coupling** as classes are separated into their individual responsibilities. **Model** is responsible for representing the data/state of a particular object, **View** handles the input/output, **Controller** handles the coordination between the View and Model by updating/retrieving information from the Model and displaying/getting input or data from the View.

2.1.2 Singleton - Restriction of the instantiation of a class to one object

We applied this design pattern under the "**EmailNotification**" class. As "**EmailNotification**" is responsible for sending email to a recipient, it also needs to set the login information for the account where all the emails will be sent from. "**EmailNotification**" contains a static method setEmailAccount that allows the client to configure the email account used to send the email to the students. This configuration should persist for every object using "**EmailNotification**", hence we applied the Singleton pattern.

2.2 Design Principles Used

2.2.1 Single Responsibility Principle (SRP)

This design principle is applied across all classes, modules and functions where each existing class assumes only one responsibility. This helps to achieve a low coupling effect such that when one class requires changes, there will not be a need to do multiple changes on other classes. For

instance, the **Student** class manages all the related attributes such as name, userId, nationality, accessTime of a single student and only contains attributes and methods relevant to a student. The **TimeTable** class's responsibility is to only manage all the index numbers that a particular Student object is registered or in the waitlist for.

2.2.2 Open Closed Principle (OCP)

Our group used the Open Closed Principle in the implementation of some of our modules, allowing for further extension in the future while being closed for modification. For example, the **INotification** interface has the abstract method send() which sends a message to the recipient. The **EmailNotification** class implements the **INotification** interface and provides the functionality of our application to be able to notify students on the status of their course registration via e-mail. Similarly, through the **INotification** interface, further notification platforms such as text messaging could be integrated to adapt the functionality of our application without changing existing code. Additionally, **AdminUi**, **StudentUi**, **LoginUi** extends the **Ui** class, and extends the functionality of our application to be able to provide user interfaces for different users. In the future, through extension of the Ui class, the STARS application will be able to extend its functionality to other types of users such as Teaching Assistants.

2.2.3 Dependency Inversion Principle (DIP)

Our group also implemented the Dependency Inversion Principle in parts of our program. One example is that our **StudentController** class depends on an interface named **Iloginable** to allow for further login check during swapping of index numbers with peers in the same course. **Iloginable** is then implemented by the **LoginManager**. The presence of interface **Iloginable** makes the two classes **LoginManager** and **StudentController** independent of each other, and the changes in logic in either class will not affect another class.

2.2.4 Dependency Injection

Dependency injection refers to the technique in which an object receives other objects that it depends on. This is implemented by constructor injection in **AdminController**, **StudentController** and **LoginController**. An example is shown in the code snippet on the right, where the dependencies are first instantiated before passing into the constructors of **AdminController**, **StudentController** and **LoginController**. This further promotes decoupling

```
IStorageManager storageManager = new StorageManager();
ILoginInfoFileManager loginInfoFileManager = new LoginInfoFileManager();
ILoginable loginManager = new LoginManager(loginInfoFileManager, storageManager);

LoginController lc = new LoginController(loginManager);
LoginInfo providedLoginInfo = lc.run();
switch (providedLoginInfo.getAccountType()) {
case ADMIN:
    AdminController ac = new AdminController(storageManager, loginInfoFileManager);
    ac.run();
    break;
case STUDENT:
    StudentController sc = new StudentController(providedLoginInfo.getUserId(), storageManager, loginManager);
    sc.run();
    break;
default:
    assert false : "Invalid accountType returned from LoginController!";
}
```

as the client using the dependencies does not need to know about the concrete classes being passed into the constructor. The dependencies could be instantiated and easily configured with specific configurations before passing it to the client. Furthermore, this promotes the ease of unit testing as the dependencies could be separately tested.

2.3 Design Considerations - OO Concepts

2.3.1 Abstraction:

Abstraction is the ability to define and differentiate different objects and classes based on their essential attributes and behaviors. Abstraction is the hiding of implementation details and showing how it's used to the user of the method. For example, the **ILoginable** interface provides the method containing the logic to verify the login information. Clients that depend on **ILoginable** do not need to know how the verification process works and just have to interact with the simple interface without any knowledge about the internal implementation.

2.3.2 Encapsulation:

We have used encapsulation to hide the values or state of a structured data object inside a class, so as to prevent unauthorized parties' direct access to them and define functionality in one logical particular place in our application. For instance, **Student** class holds private attributes like name, userId, which are not directly accessible to the public. The attributes can only be retrieved and modified through the getter and setter methods.

2.3.3 Inheritance:

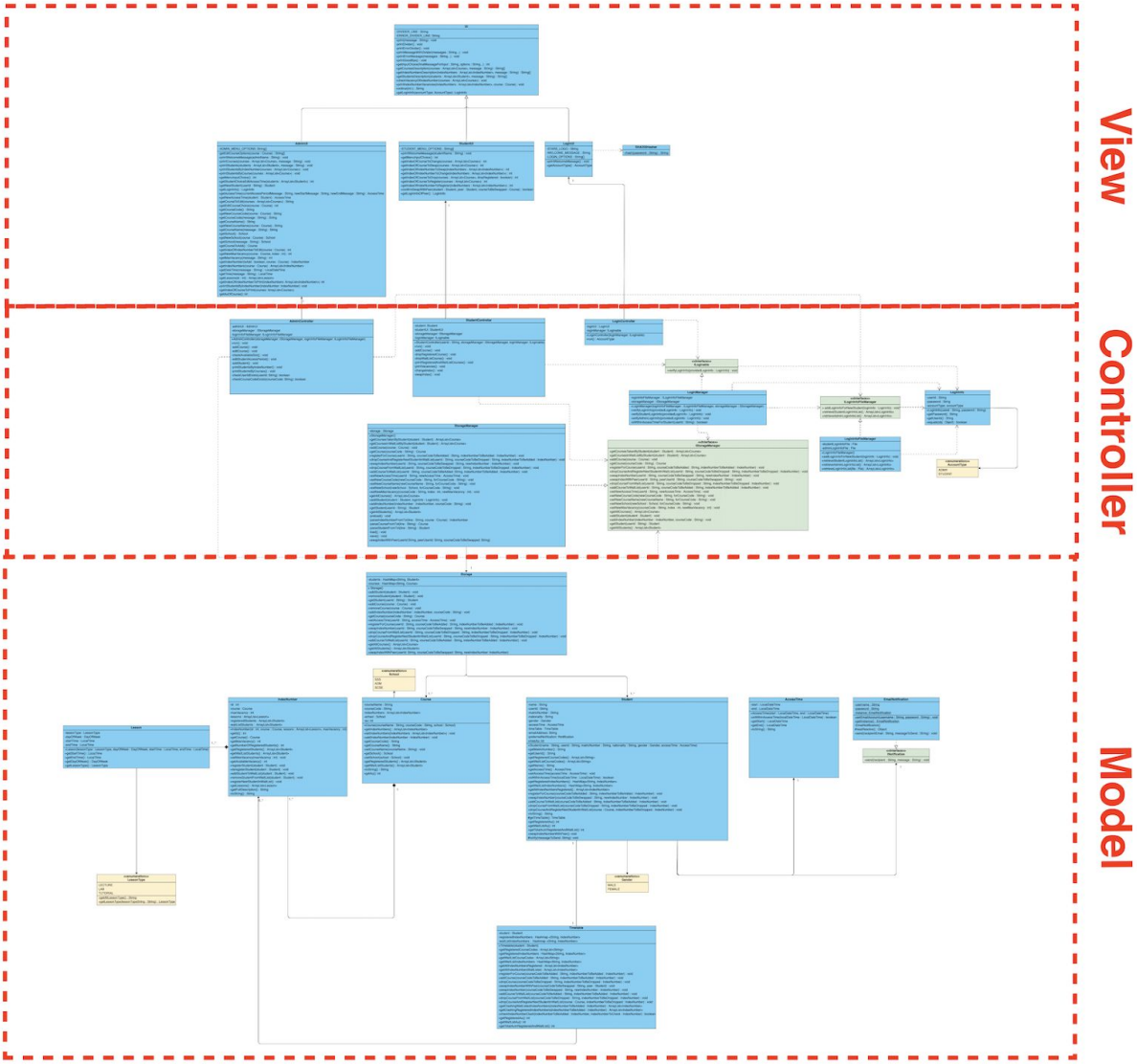
Inheritance is an OO feature that allows us to derive new classes from existing classes through inheritance of parent class attributes and behaviours. This allows for code reuse and reduces effort for implementing new classes. Also, the derived classes can be further extended to allow for more capabilities. For instance, **Ui** is the parent class of **AdminUi**, **StudentUi** and **LoginUi**. **Ui** contains some common methods that are shared amongst the child classes, such as printMessageWithDivider and printErrorMessage. All child classes use the inherited methods from the Ui parent class, thereby reducing code repetition. This also promotes the OCP principle as mentioned earlier, and allows for further extensibility of the **Ui** class in future, for e.g. if the program were to be extended for "Professor", there could be a "ProfessorUi" class.

3. Assumptions

- Course code of each course should be unique
- UserId of each student should be unique

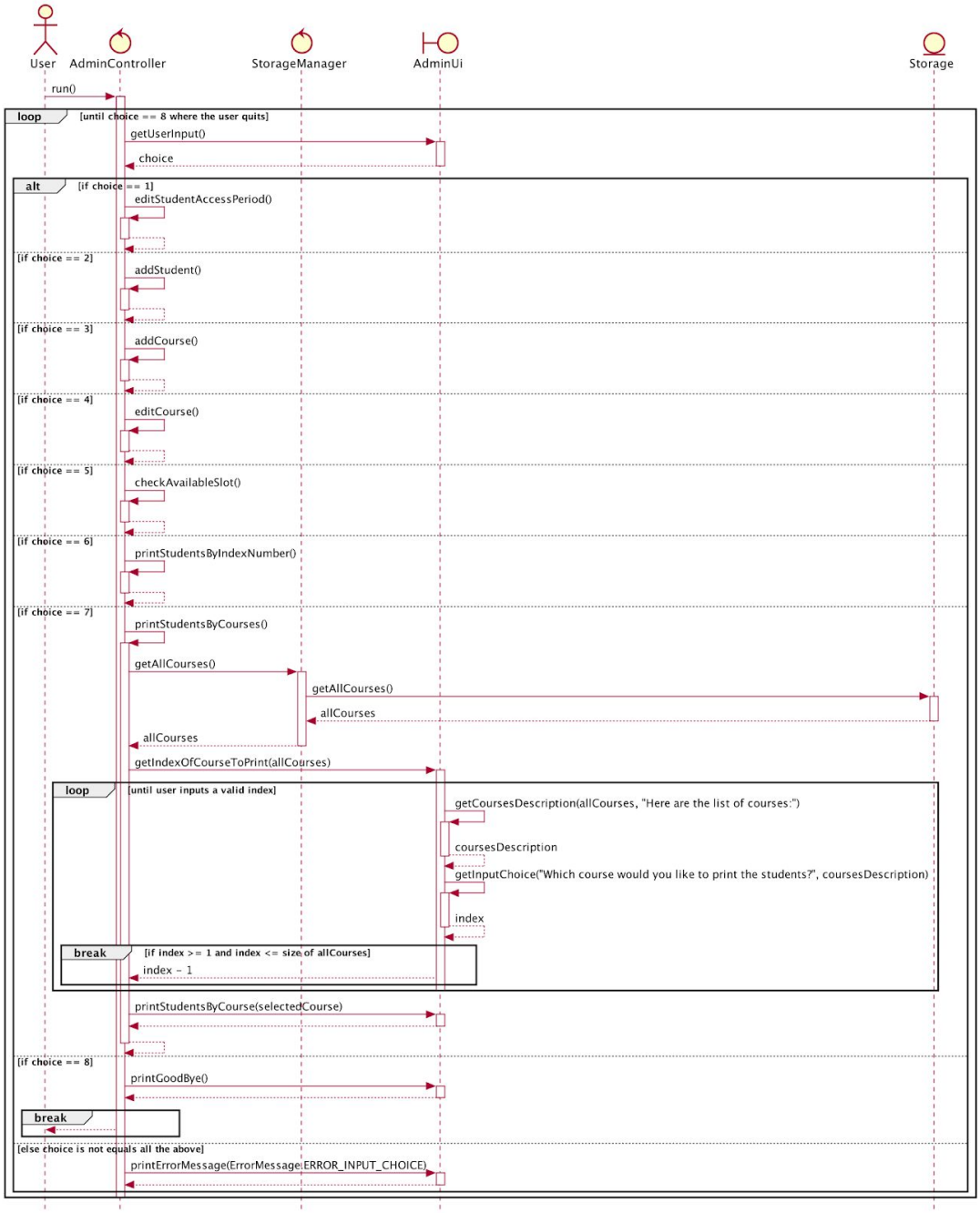
- Students are not able to register for a course (even if there is a vacancy) if it clashes with another course that is already registered or in the wait list. To register for the course, he/she must drop the course(s) that clashes first.
- The limit to the number of courses each student can register for is 21AU.
- Only one user will be logging in at any one time.

4. UML Class Diagram



Please refer to the file named ‘UML Class.png’ in our folder for a closer and more highly defined look at the Class diagram.

5. UML Sequence Diagram ("Print Student list by Course")



Please refer to the file named 'UML Sequence.png' in our folder for a closer and more highly defined look at the Sequence diagram.

6. Test Cases

1. **Student Login**	
a.) Login before allowed period (dates)	b.) Login after allowed period (dates)

```
 _________________________ __________  _________
/   _____/\__    ___/  _  \\______   \/   _____/
\_____  \   |    | /  /_\  \|       _/\_____  \
/        \  |    |/    |    \    |   \/        \
/_______  /  |____|\____|__  /____|_  /_______  /
        \/                 \/       \/        \/

Welcome to NTU Stars Planner. Please login.
------------------------------------------------------------
1. Student Login
2. Admin Login
------------------------------------------------------------
Enter your choice:
1
Enter user id:
Student19
Enter password:

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Error Message:
You are not authorised to access STARS! Your access period is:
10-12-2020 23:59 to 31-12-2020 23:59
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

```
 _________________________ __________  _________
/   _____/\__    ___/  _  \\______   \/   _____/
\_____  \   |    | /  /_\  \|       _/\_____  \
/        \  |    |/    |    \    |   \/        \
/_______  /  |____|\____|__  /____|_  /_______  /
        \/                 \/       \/        \/

Welcome to NTU Stars Planner. Please login.
------------------------------------------------------------
1. Student Login
2. Admin Login
------------------------------------------------------------
Enter your choice:
1
Enter user id:
Student25
Enter password:

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Error Message:
You are not authorised to access STARS! Your access period is:
10-10-2020 23:59 to 31-10-2020 23:59
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

c.) Wrong password

```
 _________________________ __________  _________
/   _____/\__    ___/  _  \\______   \/   _____/
\_____  \   |    | /  /_\  \|       _/\_____  \
/        \  |    |/    |    \    |   \/        \
/_______  /  |____|\____|__  /____|_  /_______  /
        \/                 \/       \/        \/

Welcome to NTU Stars Planner. Please login.
------------------------------------------------------------
1. Student Login
2. Admin Login
------------------------------------------------------------
Enter your choice:
1
Enter user id:
Student25
Enter password:

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Error Message:
Wrong UserID and/or Password. Please try again!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

2. Add a Student

a.) Add a new student

```
------------------------------------------------------------
Welcome to the Admin panel!
1. Edit student access period.
2. Add a student.
3. Add a course.
4. Edit a course.
5. Check available slot for an index number.
6. Print student list by index number.
7. Print student list by course.
8. Quit.
------------------------------------------------------------
Enter your choice:
2
Enter user id:
Student19
Enter password:

Enter the name:
Hazel Tan
Enter the matric number:
U1921000H
Enter the nationality:
Singaporean
Enter the email address:
HTAN001@e.ntu.edu.sg
Enter the start date and time for the access period in "dd-MM-yyyy HH:mm" format (e.g. 10-12-2020 23:59):
10-12-2020 23:59
Enter the end date and time for the access period in "dd-MM-yyyy HH:mm" format (e.g. 10-12-2020 23:59):
31-12-2020 23:59
Enter the gender (MALE, FEMALE):
FEMALE
Serialization Done!!
------------------------------------------------------------
Added Hazel Tan!
```

b.) Add an existing student

```
------------------------------------------------------------
Welcome to the Admin panel!
1. Edit student access period.
2. Add a student.
3. Add a course.
4. Edit a course.
5. Check available slot for an index number.
6. Print student list by index number.
7. Print student list by course.
8. Quit.
------------------------------------------------------------
Enter your choice:
2
Enter user id:
Student1
Enter password:

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Error Message:
User id already exists! Please enter a new user id.
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

c.) Invalid Data Entries

```
------------------------------------------------------------
Welcome to the Admin panel!
1. Edit student access period.
2. Add a student.
3. Add a course.
4. Edit a course.
5. Check available slot for an index number.
6. Print student list by index number.
7. Print student list by course.
8. Quit.
------------------------------------------------------------
Enter your choice:
2
Enter user id:
Student20,
Enter password:

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Error Message:
Please enter a valid user id! (no special characters e.g. *, !, {)
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Enter user id:
```

3. Add a Course

a.) Add a new course

1.)

```
------------------------------------------------------------
Welcome to the Admin panel!
1. Edit student access period.
2. Add a student.
3. Add a course.
4. Edit a course.
5. Check available slot for an index number.
6. Print student list by index number.
7. Print student list by course.
8. Quit.
------------------------------------------------------------
Enter your choice:
3
Enter course name:
Object-Oriented Design and Programming
Enter course code:
CZ2002
Enter the school offering the course (SCSE, SSS):
SCSE
Enter the 1st index number (Enter Q if you are done):
10126
Enter the maximum vacancy of this index:
4
1st Lesson for index 10126
Enter the type of lesson LECTURE, LAB, TUTORIAL (Enter Q if you are done):
LECTURE
1. Monday
2. Tuesday
3. Wednesday
4. Thursday
5. Friday
6. Saturday
7. Sunday
Enter the day of the week of the lesson (e.g. 1 for Monday, 2 for Tuesday):
4
Enter the start time (in 24 hours format e.g. 0830):
0830
Enter the end time (in 24 hours format e.g. 2341):
0930
2nd Lesson for index 10126
```

2.)

```
2nd Lesson for index 10126
Enter the type of lesson LECTURE, LAB, TUTORIAL (Enter Q if you are done):
LECTURE
1. Monday
2. Tuesday
3. Wednesday
4. Thursday
5. Friday
6. Saturday
7. Sunday
Enter the day of the week of the lesson (e.g. 1 for Monday, 2 for Tuesday):
2
Enter the start time (in 24 hours format e.g. 0830):
1430
Enter the end time (in 24 hours format e.g. 2341):
1530
3rd Lesson for index 10126
Enter the type of lesson LECTURE, LAB, TUTORIAL (Enter Q if you are done):
TUTORIAL
1. Monday
2. Tuesday
3. Wednesday
4. Thursday
5. Friday
6. Saturday
7. Sunday
Enter the day of the week of the lesson (e.g. 1 for Monday, 2 for Tuesday):
3
Enter the start time (in 24 hours format e.g. 0830):
0930
Enter the end time (in 24 hours format e.g. 2341):
1030
4th Lesson for index 10126
Enter the type of lesson LECTURE, LAB, TUTORIAL (Enter Q if you are done):
LAB
1. Monday
2. Tuesday
3. Wednesday
4. Thursday
5. Friday
```

3.)

```
5. Friday
6. Saturday
7. Sunday
Enter the day of the week of the lesson (e.g. 1 for Monday, 2 for Tuesday):
1
Enter the start time (in 24 hours format e.g. 0830):
1430
Enter the end time (in 24 hours format e.g. 2341):
1630
5th Lesson for index 10126
Enter the type of lesson LECTURE, LAB, TUTORIAL (Enter Q if you are done):
Q
Enter the 2nd index number (Enter Q if you are done):
10127
Enter the maximum vacancy of this index:
4
1st Lesson for index 10127
Enter the type of lesson LECTURE, LAB, TUTORIAL (Enter Q if you are done):
LECTURE
1. Monday
2. Tuesday
3. Wednesday
4. Thursday
5. Friday
6. Saturday
7. Sunday
Enter the day of the week of the lesson (e.g. 1 for Monday, 2 for Tuesday):
4
Enter the start time (in 24 hours format e.g. 0830):
0830
Enter the end time (in 24 hours format e.g. 2341):
0930
2nd Lesson for index 10127
Enter the type of lesson LECTURE, LAB, TUTORIAL (Enter Q if you are done):
LECTURE
1. Monday
2. Tuesday
3. Wednesday
4. Thursday
5. Friday
```

4.)

```
5. Friday
6. Saturday
7. Sunday
Enter the day of the week of the lesson (e.g. 1 for Monday, 2 for Tuesday):
2
Enter the start time (in 24 hours format e.g. 0830):
1430
Enter the end time (in 24 hours format e.g. 2341):
1530
3rd Lesson for index 10127
Enter the type of lesson LECTURE, LAB, TUTORIAL (Enter Q if you are done):
TUTORIAL
1. Monday
2. Tuesday
3. Wednesday
4. Thursday
5. Friday
6. Saturday
7. Sunday
Enter the day of the week of the lesson (e.g. 1 for Monday, 2 for Tuesday):
3
Enter the start time (in 24 hours format e.g. 0830):
1330
Enter the end time (in 24 hours format e.g. 2341):
1430
4th Lesson for index 10127
Enter the type of lesson LECTURE, LAB, TUTORIAL (Enter Q if you are done):
LAB
1. Monday
2. Tuesday
3. Wednesday
4. Thursday
5. Friday
6. Saturday
7. Sunday
Enter the day of the week of the lesson (e.g. 1 for Monday, 2 for Tuesday):
1
Enter the start time (in 24 hours format e.g. 0830):
1430
Enter the end time (in 24 hours format e.g. 2341):
```

5.)

```
Enter the start time (in 24 hours format e.g. 0830):
1430
Enter the end time (in 24 hours format e.g. 2341):
1630
5th Lesson for index 10127
Enter the type of lesson LECTURE, LAB, TUTORIAL (Enter Q if you are done):
Q
Enter the 3rd index number (Enter Q if you are done):
Q
Serialization Done!!
------------------------------------------------------------
Added CZ2002 Object-Oriented Design and Programming!
1. CZ1007 DATA STRUCTURES
2. CZ1103 INTRODUCTION TO COMPUTATIONAL THINKING & PROGRAMMING
3. CZ1101 ENGINEERING MATHEMATICS I
4. CZ2002 Object-Oriented Design and Programming
5. HE1001 MICROECONOMIC PRINCIPLES
6. HE1002 MACROECONOMIC PRINCIPLES
------------------------------------------------------------
```

b.) Add an existing course

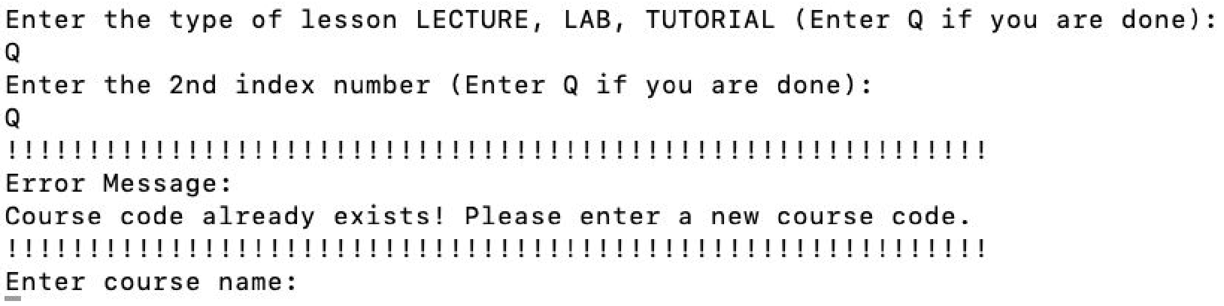
1.)

```
------------------------------------------------------------
Welcome to the Admin panel!
1. Edit student access period.
2. Add a student.
3. Add a course.
4. Edit a course.
5. Check available slot for an index number.
6. Print student list by index number.
7. Print student list by course.
8. Quit.
------------------------------------------------------------
Enter your choice:
3
Enter course name:
Object Oriented Design and Programming
Enter course code:
CZ2002
Enter the school offering the course (SCSE, SSS):
SCSE
Enter the 1st index number (Enter Q if you are done):
10126
```

2.)

```
------------------------------------------------------------
Enter the maximum vacancy of this index:
5
1st Lesson for index 10126
Enter the type of lesson LECTURE, LAB, TUTORIAL (Enter Q if you are done):
LECTURE
1. Monday
2. Tuesday
3. Wednesday
4. Thursday
5. Friday
6. Saturday
7. Sunday
Enter the day of the week of the lesson (e.g. 1 for Monday, 2 for Tuesday):
4
Enter the start time (in 24 hours format e.g. 0830):
0830
Enter the end time (in 24 hours format e.g. 2341):
0930
```

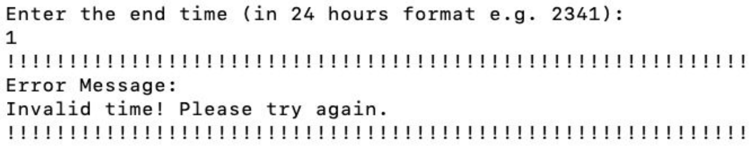
3.)



```
Enter the type of lesson LECTURE, LAB, TUTORIAL (Enter Q if you are done):
Q
Enter the 2nd index number (Enter Q if you are done):
Q
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Error Message:
Course code already exists! Please enter a new course code.
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Enter course name:
```

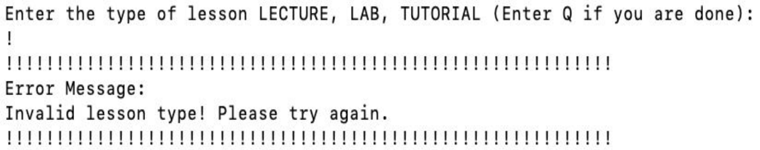
c.) Invalid Data Entries

1.) Invalid time input



```
Enter the end time (in 24 hours format e.g. 2341):
1
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Error Message:
Invalid time! Please try again.
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

2.) Invalid lesson input

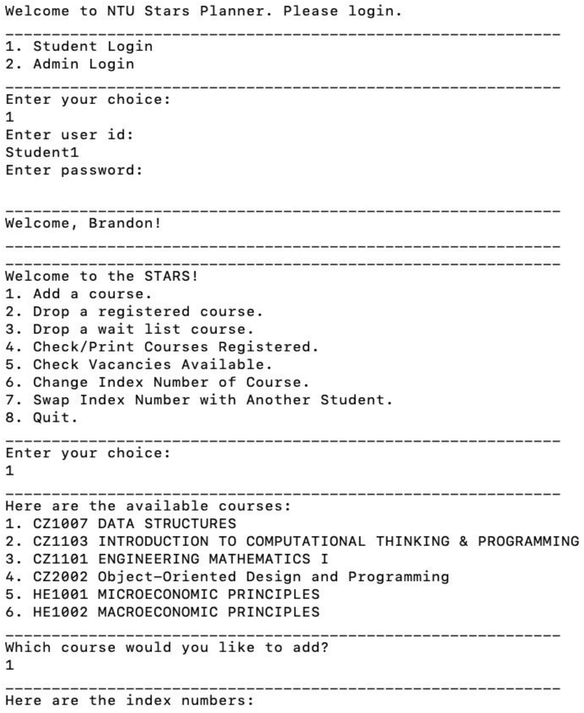


```
Enter the type of lesson LECTURE, LAB, TUTORIAL (Enter Q if you are done):
!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Error Message:
Invalid lesson type! Please try again.
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

4. Register Student For Course

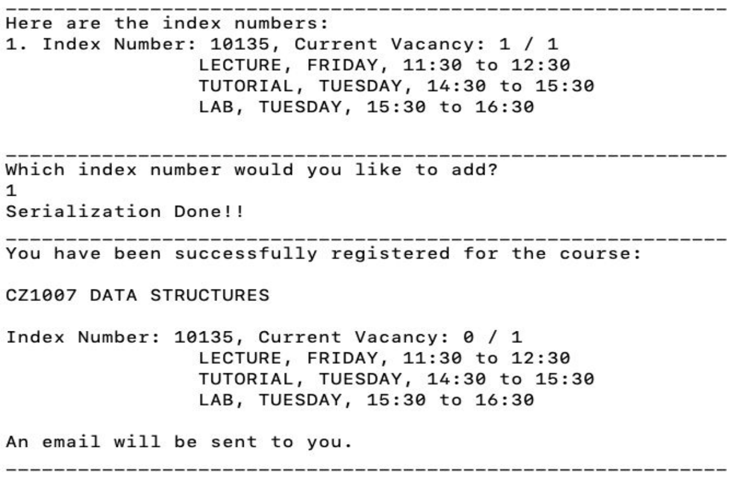
a.) Add a student to a course index with available vacancies.

1.)



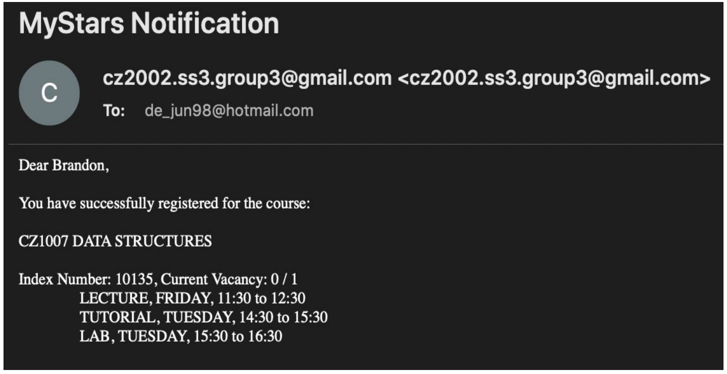
```
Welcome to NTU Stars Planner. Please login.
------------------------------------------------------------
1. Student Login
2. Admin Login
------------------------------------------------------------
Enter your choice:
1
Enter user id:
Student1
Enter password:

------------------------------------------------------------
Welcome, Brandon!
------------------------------------------------------------
------------------------------------------------------------
Welcome to the STARS!
1. Add a course.
2. Drop a registered course.
3. Drop a wait list course.
4. Check/Print Courses Registered.
5. Check Vacancies Available.
6. Change Index Number of Course.
7. Swap Index Number with Another Student.
8. Quit.
------------------------------------------------------------
Enter your choice:
1
------------------------------------------------------------
Here are the available courses:
1. CZ1007 DATA STRUCTURES
2. CZ1103 INTRODUCTION TO COMPUTATIONAL THINKING & PROGRAMMING
3. CZ1101 ENGINEERING MATHEMATICS I
4. CZ2002 Object-Oriented Design and Programming
5. HE1001 MICROECONOMIC PRINCIPLES
6. HE1002 MACROECONOMIC PRINCIPLES
------------------------------------------------------------
Which course would you like to add?
1
------------------------------------------------------------
Here are the index numbers:
```

2.)



```
------------------------------------------------------------
Here are the index numbers:
1. Index Number: 10135, Current Vacancy: 1 / 1
                LECTURE, FRIDAY, 11:30 to 12:30
                TUTORIAL, TUESDAY, 14:30 to 15:30
                LAB, TUESDAY, 15:30 to 16:30


------------------------------------------------------------
Which index number would you like to add?
1
Serialization Done!!
------------------------------------------------------------
You have been successfully registered for the course:

CZ1007 DATA STRUCTURES

Index Number: 10135, Current Vacancy: 0 / 1
                LECTURE, FRIDAY, 11:30 to 12:30
                TUTORIAL, TUESDAY, 14:30 to 15:30
                LAB, TUESDAY, 15:30 to 16:30

An email will be sent to you.
------------------------------------------------------------
```

3.)



MyStars Notification

cz2002.ss3.group3@gmail.com <cz2002.ss3.group3@gmail.com>
To: de_jun98@hotmail.com

Dear Brandon,

You have successfully registered for the course:

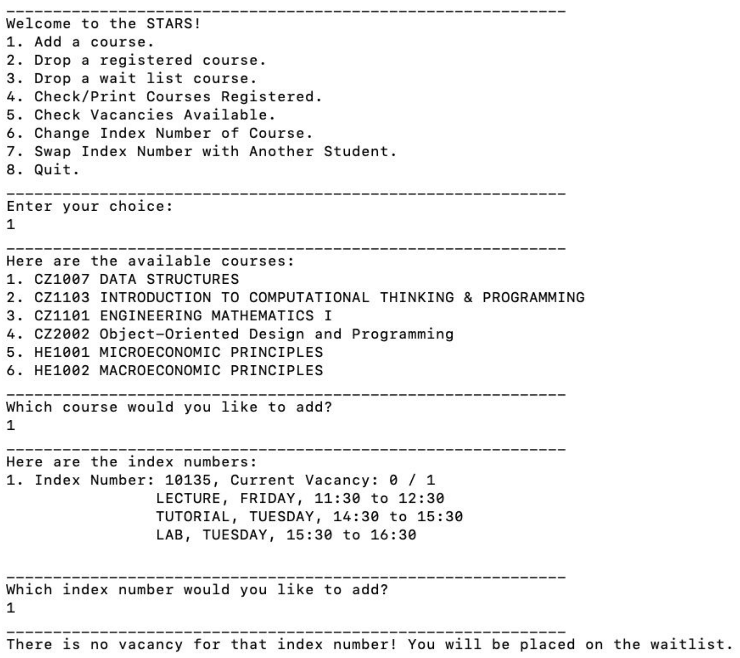
CZ1007 DATA STRUCTURES

Index Number: 10135, Current Vacancy: 0 / 1
LECTURE, FRIDAY, 11:30 to 12:30
TUTORIAL, TUESDAY, 14:30 to 15:30
LAB, TUESDAY, 15:30 to 16:30

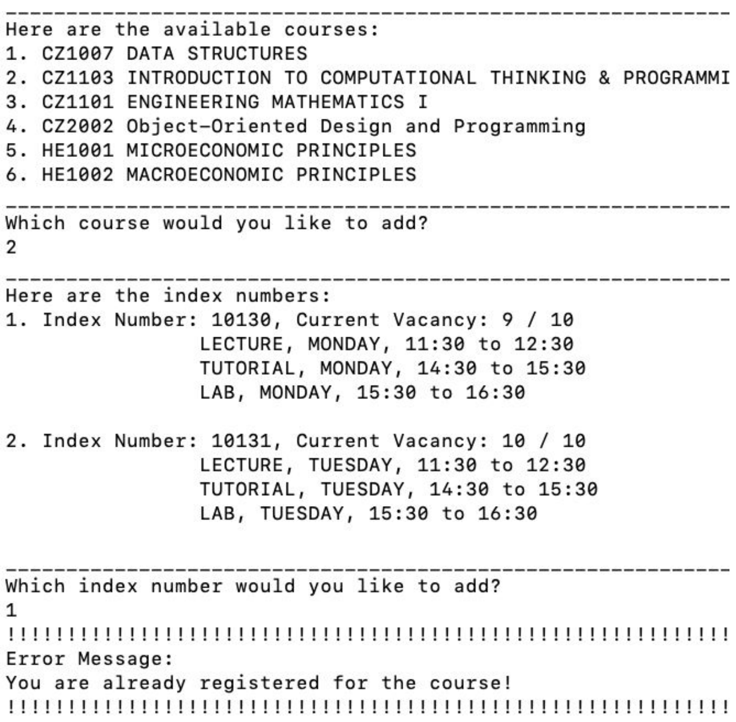
b.)Add a student to a course index with 0 vacancies in Tut /

c.)Register the same course again

Lab.

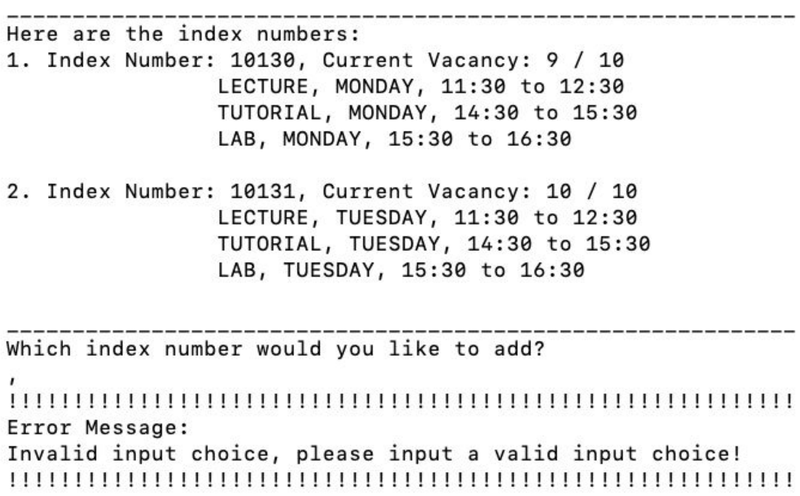


```
------------------------------------------------------------
Welcome to the STARS!
1. Add a course.
2. Drop a registered course.
3. Drop a wait list course.
4. Check/Print Courses Registered.
5. Check Vacancies Available.
6. Change Index Number of Course.
7. Swap Index Number with Another Student.
8. Quit.
------------------------------------------------------------
Enter your choice:
1
------------------------------------------------------------
Here are the available courses:
1. CZ1007 DATA STRUCTURES
2. CZ1103 INTRODUCTION TO COMPUTATIONAL THINKING & PROGRAMMING
3. CZ1101 ENGINEERING MATHEMATICS I
4. CZ2002 Object-Oriented Design and Programming
5. HE1001 MICROECONOMIC PRINCIPLES
6. HE1002 MACROECONOMIC PRINCIPLES
------------------------------------------------------------
Which course would you like to add?
1
------------------------------------------------------------
Here are the index numbers:
1. Index Number: 10135, Current Vacancy: 0 / 1
                LECTURE, FRIDAY, 11:30 to 12:30
                TUTORIAL, TUESDAY, 14:30 to 15:30
                LAB, TUESDAY, 15:30 to 16:30


------------------------------------------------------------
Which index number would you like to add?
1
------------------------------------------------------------
There is no vacancy for that index number! You will be placed on the waitlist.
```



```
------------------------------------------------------------
Here are the available courses:
1. CZ1007 DATA STRUCTURES
2. CZ1103 INTRODUCTION TO COMPUTATIONAL THINKING & PROGRAMMI
3. CZ1101 ENGINEERING MATHEMATICS I
4. CZ2002 Object-Oriented Design and Programming
5. HE1001 MICROECONOMIC PRINCIPLES
6. HE1002 MACROECONOMIC PRINCIPLES
------------------------------------------------------------
Which course would you like to add?
2
------------------------------------------------------------
Here are the index numbers:
1. Index Number: 10130, Current Vacancy: 9 / 10
                LECTURE, MONDAY, 11:30 to 12:30
                TUTORIAL, MONDAY, 14:30 to 15:30
                LAB, MONDAY, 15:30 to 16:30

2. Index Number: 10131, Current Vacancy: 10 / 10
                LECTURE, TUESDAY, 11:30 to 12:30
                TUTORIAL, TUESDAY, 14:30 to 15:30
                LAB, TUESDAY, 15:30 to 16:30


------------------------------------------------------------
Which index number would you like to add?
1
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Error Message:
You are already registered for the course!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

d.)Invalid data entries (eg wrong student ID / course code, etc)



```
------------------------------------------------------------
Here are the index numbers:
1. Index Number: 10130, Current Vacancy: 9 / 10
                LECTURE, MONDAY, 11:30 to 12:30
                TUTORIAL, MONDAY, 14:30 to 15:30
                LAB, MONDAY, 15:30 to 16:30

2. Index Number: 10131, Current Vacancy: 10 / 10
                LECTURE, TUESDAY, 11:30 to 12:30
                TUTORIAL, TUESDAY, 14:30 to 15:30
                LAB, TUESDAY, 15:30 to 16:30


------------------------------------------------------------
Which index number would you like to add?
,
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Error Message:
Invalid input choice, please input a valid input choice!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

5.) Check available slot in a class (vacancy in a class)

a.)Check for vacancy in course index	**b.) Invalid data entries**

```
------------------------------------------------------------
Welcome, Brandon!
------------------------------------------------------------
------------------------------------------------------------
Welcome to the STARS!
1. Add a course.
2. Drop a registered course.
3. Drop a wait list course.
4. Check/Print Courses Registered.
5. Check Vacancies Available.
6. Change Index Number of Course.
7. Swap Index Number with Another Student.
8. Quit.
------------------------------------------------------------
Enter your choice:
5
------------------------------------------------------------
Here are the list of courses:
1. CZ1007 DATA STRUCTURES
2. CZ1103 INTRODUCTION TO COMPUTATIONAL THINKING & PROGRAMMING
3. CZ1101 ENGINEERING MATHEMATICS I
4. CZ2002 Object-Oriented Design and Programming
5. HE1001 MICROECONOMIC PRINCIPLES
6. HE1002 MACROECONOMIC PRINCIPLES
------------------------------------------------------------
Which course do you want to check?
2
------------------------------------------------------------
Here are the vacancies for CZ1103 INTRODUCTION TO COMPUTATIONAL THINKING & PROGRAMMING
Index Number: 10130, Current Vacancy: 9 / 10
Index Number: 10131, Current Vacancy: 10 / 10
```

```
------------------------------------------------------------
Here are the list of courses:
1. CZ1007 DATA STRUCTURES
2. CZ1103 INTRODUCTION TO COMPUTATIONAL THINKING & PROGRAMMING
3. CZ1101 ENGINEERING MATHEMATICS I
4. CZ2002 Object-Oriented Design and Programming
5. HE1001 MICROECONOMIC PRINCIPLES
6. HE1002 MACROECONOMIC PRINCIPLES
------------------------------------------------------------
Which course do you want to check?
,
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Error Message:
Invalid input choice, please input a valid input choice!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
------------------------------------------------------------
Here are the list of courses:
1. CZ1007 DATA STRUCTURES
2. CZ1103 INTRODUCTION TO COMPUTATIONAL THINKING & PROGRAMMING
3. CZ1101 ENGINEERING MATHEMATICS I
4. CZ2002 Object-Oriented Design and Programming
5. HE1001 MICROECONOMIC PRINCIPLES
6. HE1002 MACROECONOMIC PRINCIPLES
------------------------------------------------------------
Which course do you want to check?
```

6.Day/Time clash with other course

a.)Add a student to a course index with available vacancies.

```
------------------------------------------------------------
Here are the available courses:
1. CZ1007 DATA STRUCTURES
2. CZ1103 INTRODUCTION TO COMPUTATIONAL THINKING & PROGRAMMING
3. CZ1101 ENGINEERING MATHEMATICS I
4. CZ2002 Object-Oriented Design and Programming
5. HE1001 MICROECONOMIC PRINCIPLES
6. HE1002 MACROECONOMIC PRINCIPLES
------------------------------------------------------------
Which course would you like to add?
3
------------------------------------------------------------
Here are the index numbers:
1. Index Number: 10137, Current Vacancy: 10 / 10
                LECTURE, TUESDAY, 11:30 to 12:30
                TUTORIAL, TUESDAY, 14:30 to 15:30
                LAB, TUESDAY, 15:30 to 16:30


------------------------------------------------------------
Which index number would you like to add?
1
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Error Message:
The index number you have selected clashes with the timetable of the courses you are registered for! Please try again.
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

7.) Waitlist notification

a.) Add studentA to a course index with 0 vacancies	b.)Drop studentB from the same course index

```
------------------------------------------------------------
Welcome to the STARS!
1. Add a course.
2. Drop a registered course.
3. Drop a wait list course.
4. Check/Print Courses Registered.
5. Check Vacancies Available.
6. Change Index Number of Course.
7. Swap Index Number with Another Student.
8. Quit.
------------------------------------------------------------
Enter your choice:
1
------------------------------------------------------------
Here are the available courses:
1. CZ1007 DATA STRUCTURES
2. CZ1103 INTRODUCTION TO COMPUTATIONAL THINKING & PROGRAMMING
3. CZ1101 ENGINEERING MATHEMATICS I
4. CZ2002 Object-Oriented Design and Programming
5. HE1001 MICROECONOMIC PRINCIPLES
6. HE1002 MACROECONOMIC PRINCIPLES
------------------------------------------------------------
Which course would you like to add?
1
------------------------------------------------------------
Here are the index numbers:
1. Index Number: 10135, Current Vacancy: 0 / 1
                LECTURE, FRIDAY, 11:30 to 12:30
                TUTORIAL, TUESDAY, 14:30 to 15:30
                LAB, TUESDAY, 15:30 to 16:30

------------------------------------------------------------
Which index number would you like to add?
1
------------------------------------------------------------
There is no vacancy for that index number! You will be placed on the waitlist.
```

```
------------------------------------------------------------
Welcome, Brandon!
------------------------------------------------------------
------------------------------------------------------------
Welcome to the STARS!
1. Add a course.
2. Drop a registered course.
3. Drop a wait list course.
4. Check/Print Courses Registered.
5. Check Vacancies Available.
6. Change Index Number of Course.
7. Swap Index Number with Another Student.
8. Quit.
------------------------------------------------------------
Enter your choice:
2
------------------------------------------------------------
Here are your registered courses:
1. CZ1007 DATA STRUCTURES
2. CZ1103 INTRODUCTION TO COMPUTATIONAL THINKING & PROGRAMMING
3. HE1001 MICROECONOMIC PRINCIPLES
------------------------------------------------------------
Which course would you like to drop?
1
Serialization Done!!
------------------------------------------------------------
You have successfully dropped the course:

CZ1007 DATA STRUCTURES

Index Number: 10135, Current Vacancy: 0 / 1
                LECTURE, FRIDAY, 11:30 to 12:30
                TUTORIAL, TUESDAY, 14:30 to 15:30
                LAB, TUESDAY, 15:30 to 16:30

An email will be sent to you
------------------------------------------------------------
```

c.)Display studentA timetable

```
------------------------------------------------------------
Welcome to the STARS!
1. Add a course.
2. Drop a registered course.
3. Drop a wait list course.
4. Check/Print Courses Registered.
5. Check Vacancies Available.
6. Change Index Number of Course.
7. Swap Index Number with Another Student.
8. Quit.
------------------------------------------------------------
Enter your choice:
4
------------------------------------------------------------
Here are the courses you are registered for:

1. CZ1007 DATA STRUCTURES
        Index Number: 10135, Current Vacancy: 0 / 1
                LECTURE, FRIDAY, 11:30 to 12:30
                TUTORIAL, TUESDAY, 14:30 to 15:30
                LAB, TUESDAY, 15:30 to 16:30

2. CZ1103 INTRODUCTION TO COMPUTATIONAL THINKING & PROGRAMMING
        Index Number: 10130, Current Vacancy: 8 / 10
                LECTURE, MONDAY, 11:30 to 12:30
                TUTORIAL, MONDAY, 14:30 to 15:30
                LAB, MONDAY, 15:30 to 16:30

3. HE1001 MICROECONOMIC PRINCIPLES
        Index Number: 10130, Current Vacancy: 8 / 10
                LECTURE, WEDNESDAY, 11:30 to 12:30
                TUTORIAL, WEDNESDAY, 14:30 to 15:30

------------------------------------------------------------
------------------------------------------------------------
Here are the courses on your wait list:


------------------------------------------------------------
```

8. Print student list by index number, course

a.) Print list by Course

b.)Print list by Index

```
------------------------------------------------------------
Welcome to the Admin panel!
1. Edit student access period.
2. Add a student.
3. Add a course.
4. Edit a course.
5. Check available slot for an index number.
6. Print student list by index number.
7. Print student list by course.
8. Quit.
------------------------------------------------------------
Enter your choice:
7
------------------------------------------------------------
Here are the list of courses:
1. CZ1007 DATA STRUCTURES
2. CZ1103 INTRODUCTION TO COMPUTATIONAL THINKING & PROGRAMMING
3. CZ1101 ENGINEERING MATHEMATICS I
4. CZ2002 Object-Oriented Design and Programming
5. HE1001 MICROECONOMIC PRINCIPLES
6. HE1002 MACROECONOMIC PRINCIPLES
------------------------------------------------------------
Which course would you like to print the students?
5
------------------------------------------------------------
Course: HE1001 MICROECONOMIC PRINCIPLES

Registered Students:
Name: Brandon, Gender: MALE, Nationality: Singaporean
Name: Eugene, Gender: MALE, Nationality: Malaysian

Wait List Students:
```

```
------------------------------------------------------------
Welcome to the Admin panel!
1. Edit student access period.
2. Add a student.
3. Add a course.
4. Edit a course.
5. Check available slot for an index number.
6. Print student list by index number.
7. Print student list by course.
8. Quit.
------------------------------------------------------------
Enter your choice:
6
------------------------------------------------------------
Here are the list of courses:
1. CZ1007 DATA STRUCTURES
2. CZ1103 INTRODUCTION TO COMPUTATIONAL THINKING & PROGRAMMING
3. CZ1101 ENGINEERING MATHEMATICS I
4. CZ2002 Object-Oriented Design and Programming
5. HE1001 MICROECONOMIC PRINCIPLES
6. HE1002 MACROECONOMIC PRINCIPLES
------------------------------------------------------------
Which course would you like to print the students?
2
------------------------------------------------------------
Here are the list of index numbers:
1. Index Number: 10130, Current Vacancy: 8 / 10
                LECTURE, MONDAY, 11:30 to 12:30
                TUTORIAL, MONDAY, 14:30 to 15:30
                LAB, MONDAY, 15:30 to 16:30

2. Index Number: 10131, Current Vacancy: 9 / 10
                LECTURE, TUESDAY, 11:30 to 12:30
                TUTORIAL, TUESDAY, 14:30 to 15:30
                LAB, TUESDAY, 15:30 to 16:30


------------------------------------------------------------
Which index number would you like to print the students?
1
------------------------------------------------------------
Course: CZ1103 INTRODUCTION TO COMPUTATIONAL THINKING & PROGRAMMING

Index Number: 10130, Current Vacancy: 8 / 10
Registered Students:
Name: Brandon, Gender: MALE, Nationality: Singaporean
Name: Eugene, Gender: MALE, Nationality: Malaysian

Wait List Students:
```

Additional Test Cases

9. Swop index number with another student

1.Student 1's Index Number

```
You have been successfully registered for the course:

CZ1103 INTRODUCTION TO COMPUTATIONAL THINKING & PROGRAMMING

Index Number: 10130, Current Vacancy: 9 / 10
                LECTURE, MONDAY, 11:30 to 12:30
                TUTORIAL, MONDAY, 14:30 to 15:30
                LAB, MONDAY, 15:30 to 16:30

An email will be sent to you.
```

2.Student 2's Index Number

```
You have been successfully registered for the course:

CZ1103 INTRODUCTION TO COMPUTATIONAL THINKING & PROGRAMMING

Index Number: 10131, Current Vacancy: 9 / 10
                LECTURE, TUESDAY, 11:30 to 12:30
                TUTORIAL, TUESDAY, 14:30 to 15:30
                LAB, TUESDAY, 15:30 to 16:30

An email will be sent to you.
```

3.Swapping of Index

```
------------------------------------------------------------
Which course would you like to change?
1
Enter user id:
Student1
Enter password:

------------------------------------------------------------
Confirm swap for CZ1103 INTRODUCTION TO COMPUTATIONAL THINKING & PROGRAMMING with peer?

Your index number:
Index Number: 10131, Current Vacancy: 9 / 10
                LECTURE, TUESDAY, 11:30 to 12:30
                TUTORIAL, TUESDAY, 14:30 to 15:30
                LAB, TUESDAY, 15:30 to 16:30

Your peer Brandon's index number:
Index Number: 10130, Current Vacancy: 9 / 10
                LECTURE, MONDAY, 11:30 to 12:30
                TUTORIAL, MONDAY, 14:30 to 15:30
                LAB, MONDAY, 15:30 to 16:30

------------------------------------------------------------
Enter (Y for yes, N for no):
Y
------------------------------------------------------------
Your Index Number: 10131 for CZ1103 INTRODUCTION TO COMPUTATIONAL THINKING & PROGRAMMING has been successfully changed to: 10130
An email will be sent to you and your peer.
```

10. Student hits Maximum AU Ceiling while registering for courses

```
------------------------------------------------------------
Here are the available courses:
1. CZ1008 Software Engineering
2. CZ1007 DATA STRUCTURES
3. CZ1103 INTRODUCTION TO COMPUTATIONAL THINKING & PROGRAMMING
4. CZ1101 ENGINEERING MATHEMATICS I
5. HE1001 MICROECONOMIC PRINCIPLES
6. HE1002 MACROECONOMIC PRINCIPLES
------------------------------------------------------------
Which course would you like to add?
1
------------------------------------------------------------
Here are the index numbers:
1. Index Number: 123, Current Vacancy: 4 / 4
                LAB, MONDAY, 12:30 to 14:30


------------------------------------------------------------
Which index number would you like to add?
1
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Error Message:
You have exceeded the maximum AU allowed! Please drop your
registered courses or wait list courses before trying again.
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

7. Video Link

Link: https://youtu.be/wbPBIBWeg7s

8. Bibliography

Ashish Shukla. (2014, 09 28). *Developing MVC applications using SOLID principles*. https://www.codeproject.com/Articles/822791/Developing-MVC-applications-using-SOLID-principles

----------END OF REPORT----------

Attached a scanned copy with the report with the filled details and signatures.

Declaration of Original Work for CE/CZ2002 Assignment

We hereby declare that the attached group assignment has been researched, undertaken, completed and submitted as a collective effort by the group members listed below.

We have honored the principles of academic integrity and have upheld Student Code of Academic Conduct in the completion of this work.

We understand that if plagiarism is found in the assignment, then lower marks or no marks will be awarded for the assessed work. In addition, disciplinary actions may be taken.

Name	Course (CE2002 or CZ2002)	Lab Group	Signature/Date
Ernest Ang Cheng Han	CZ2002	SS4	24/11/2020
Alvin Tan De Jun	CZ2002	SS4	24/11/2020
Rajkumar Snehaa	CZ2002	SS4	24/11/2020
Ng Chi Hui	CZ2002	SS4	24/11/2020
Cai Xinrui	CZ2002	SS4	24/11/2020