Design Considerations

Approach Taken:

We aimed to mimic previous designs of student course registration systems; to take onboard what was considered ‘good’ design practices and improve on the areas that were missing

Assumptions Made:

- Every student, no matter which faculty they are from, has the same access period. Unlike the actual STARS system where different students may have different access periods.
-Every lab/lecture/tutorial slot is assumed to be only on either odd, even or weekly basis.

OO Design Principles

As taught in the course, the basis of good design aims to achieve loose coupling and high cohesion between the different classes, whilst keeping in mind a design with high reusability (can be reused in other projects), extensibility (can be extended/inherited if one needs to extend functionality) and maintainability (if need to change, only minimum effort is needed). While constructing the application, we also tried to fulfil some of the SOLID Design Principles:

Single Responsibility Principle: Each class should only have a single Responsibility, so that if changes need to be made, only that class needs to be changed, in general. This would also help achieve cohesion. For our design, we tried to follow the ECB architecture, so that each class would mainly be responsible for reading/storing data, managing logic and I/O respectively.

Open-Closed Principle: “A module should be open for extension but closed for modification”. We want to be able to change what the modules do, without changing the source code of the modules. We want to reduce modification and allow extension. This means that a certain class allows for more inheritance from it, but its methods are kept consistent so that existing classes that inherit/implement it would not be disrupted. For instance, we added an abstract class, NotificationManager, which can be extended from, if say the future functionalities to be added

are additional modes of notifications like SMS, WhatsApp, etc. are used instead of just email. This may be done without modifying the NotificationManager code.

Dependency Injection Principle: Exploring the dependency injection principle, the principle states that high level modules should not depend upon low level modules. Rather, both should depend upon abstractions.
We tried to create an interface called UserUI so that 'high-level' users would not depend directly on the actual 'low-level' boundary classes.

Use of Object-Oriented Concepts

Abstraction is the process of identifying essential characteristics (i.e. attributes & behaviour) of an object that distinguish it from all other kinds of objects.
-Implementation: entity classes like Student and Staff, who both inherit from User superclass, have a different set of attributes. E.g. Students have matriculation number recorded, but Staff members do not

Encapsulation: data in the object is private data, so other objects cannot get information about its state. Access to private data can be obtained through the public methods of the object's class (e.g. get methods); but an object has to send a request to another to get its data.
-Implementation: in general, most if not all of the attributes in the entity classes are private. In fact, most have their setter methods removed, so that they can only be modified during initialization when calling the constructor

Information hiding hides the implementation details for the methods from others. Other objects know what this object's methods will do but they (the other objects) don't know how those object's methods are implemented.
-Implementation: Many of the methods implemented are public, in fact some are even static; so other objects can call these methods but don't know how those object's methods are implemented.

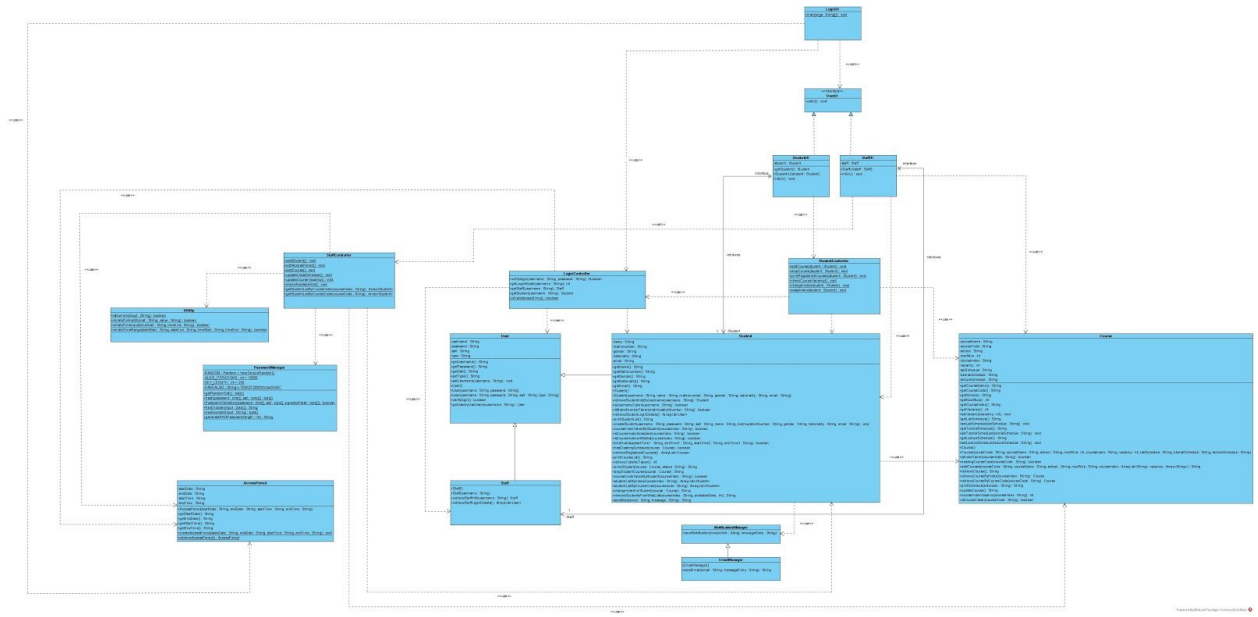
Inheritance: If you subsequently define a new class, you don't have to design it from scratch; you can inherit some of the properties & behaviours of an existing class. One can create new classes without extensive duplication of code, by simply reusing the parent class' code.
-Implementation: a Superclass 'User' was defined and 2 subclasses, 'Student' and 'Staff' inherited attributes such as username, password and methods such as getUserbyUsername().

Polymorphism: An object can send the same message to different objects from different subclasses to ask each receiving object to do sth. The objects would then perform the corresponding operation according to the message, but they may do things differently.
-Implementation: the interface 'UserUI' was defined with an abstract method initUI() to initialiseUI. 2 classes that realise the interface, 'StudentUI' and 'StaffUI', both call initUI() to initialiseUI, but both implement it differently.

Applying Entity-Control-Boundary (ECB) architecture

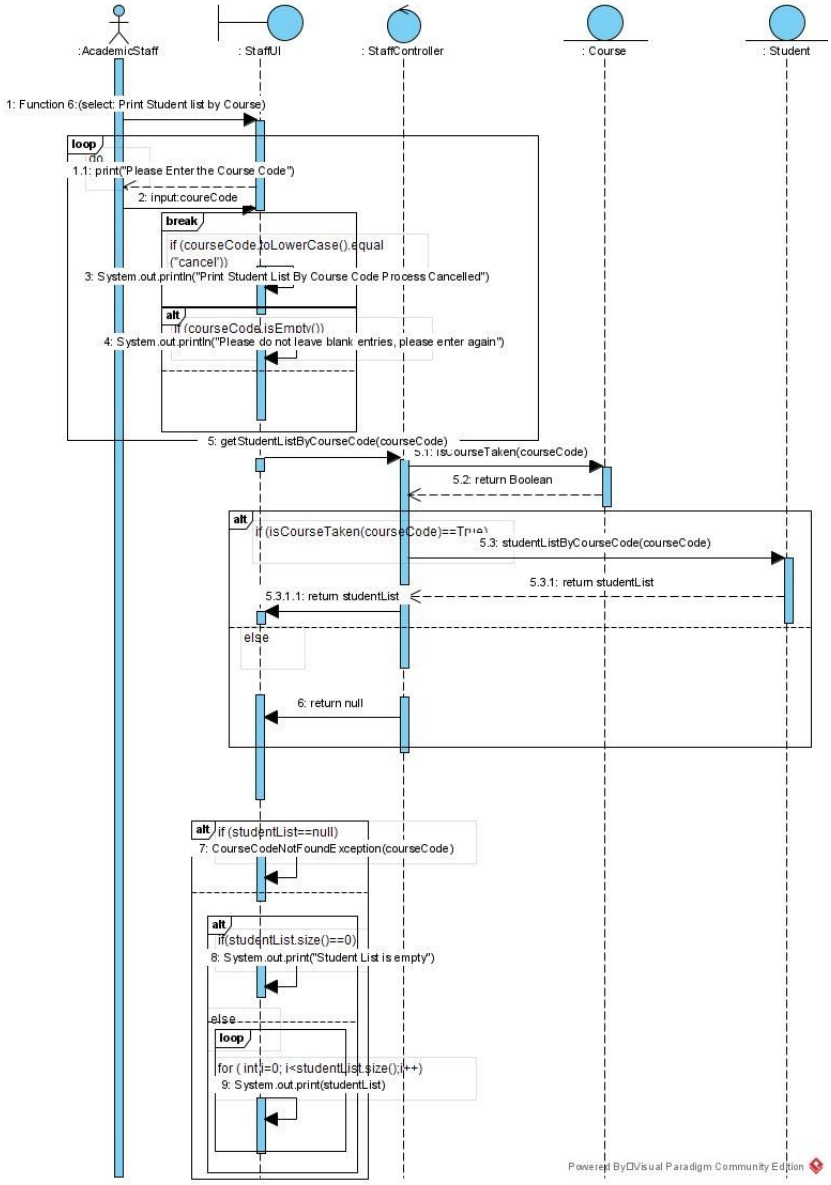
Classes created were placed into these packages titled: Entity, Control, Boundary, Helper and Exceptions
-Boundary classes are for input/output, e.g. StaffUI and StudentUI displaying options to the staff/student.
-Entity classes are for storing data/info in its attributes, e.g. Course stores data such as no. of vacancies, course code; and Student stores data such as student's name, gender, nationality in the attributes.
-Control Class: Holds Logic to coordinate program flow and realise use case (functional usage). E.g. StudentController and StaffController contains logic for some of the use case functions
-Helper Class: Contains Utility functions for various portions of the application, but were decided to neither be Entity nor Control classes.
-Exception Class: Custom exceptions for the program to be thrown in the function where it occurs and caught in the boundary class object
Different class stereotypes should interact with each other in a defined manner. E.g. boundary classes should not have any relationship with entity classes, while control classes should have a relationship with both boundary and entity classes.

a) A detailed UML Class Diagram for the application (exported as an image)



Please refer to the image in the project folder:
(Class Diagram\CZ2002_MYSTARS_ClassDiagram.jpg)

b) A detailed UML Sequence Diagram (exported as an image)



Please refer to the image in the project folder
(Sequence Diagram\CZ2002_MYSTARS_SequenceDiagram.jpg)

YouTube link:
https://www.youtube.com/watch?v=OgFeOGxTtqw&feature=youtu.be&ab_channel=TesterTest

Test Cases

1. Student Login

1a. Login before allowed period (dates)

```
====MySTARS====
1. Login
2. Exit
1
Username:
a
Enter password :
Processing....
Sorry you are not allowed to access the portal now!
Please log in at your specified access period!

Current date is                 2020-11-22

Your access date is from        2020-11-28 to 2021-08-28

Current time is                 00:13
Your access time is from        10:08 to 22:08
====MySTARS====
1. Login
2. Exit
```

1b. Login after allowed period (dates)

```
Enter the number of your choice: 0
====MySTARS====
1. Login
2. Exit
1
Username:
a
Enter password :
Processing....
Sorry you are not allowed to access the portal now!
Please log in at your specified access period!

Current date is                 2020-11-22

Your access date is from        2020-08-08 to 2020-08-28

Current time is                 00:16
Your access time is from        08:08 to 18:08
====MySTARS====
1. Login
2. Exit
```

1c. Wrong password

```
====MySTARS====
1. Login
2. Exit
1
Username:
a
Enter password :
Processing....

Login Unsuccessful.
Username or password is incorrect or is not registered in the system
====MySTARS====
1. Login
2. Exit
```

2. Add a student

2a. Add a new student

```
Please enter student's name:
Mindy
Please enter student's matric number:
U829819J
Please enter student's email:
cz2002.student@gmail.com
Please enter student's gender: ('Male' or 'Female')
Female
Please enter student's nationality:
Singaporean
====================Student List====================
Username: Zteh24 Matriculation No: U123456E Name: Zen
Username: Kyle256 Matriculation No: U224466E Name: Kyle
Username: Erric23 Matriculation No: U987987E Name: Erric
Username: JustinT99 Matriculation No: U123890E Name: Justin
Username: a Matriculation No: U1234567H Name: Tester
Username: bob123 Matriculation No: U2020202H Name: Bob
Username: q Matriculation No: U2022222B Name: Quentin
Username: c Matriculation No: U2002020X Name: Cheryl
Username: Aaa12 Matriculation No: U123123E Name: Aaron
Username: Ate123 Matriculation No: U232323H Name: Atricia
Username: Ump45 Matriculation No: U363478F Name: Umpire
Username: Ink412 Matriculation No: U389489I Name: Inklay
Username: Rek23 Matriculation No: U939139J Name: Rekky
Username: Hal30 Matriculation No: U1921932K Name: Halberg
Username: Joe123 Matriculation No: U929910M Name: Joe
Username: Min20 Matriculation No: U829819J Name: Mindy

Add Student Operation Completed!! Press the "ENTER" key to be directed back to STARS main menu!
```

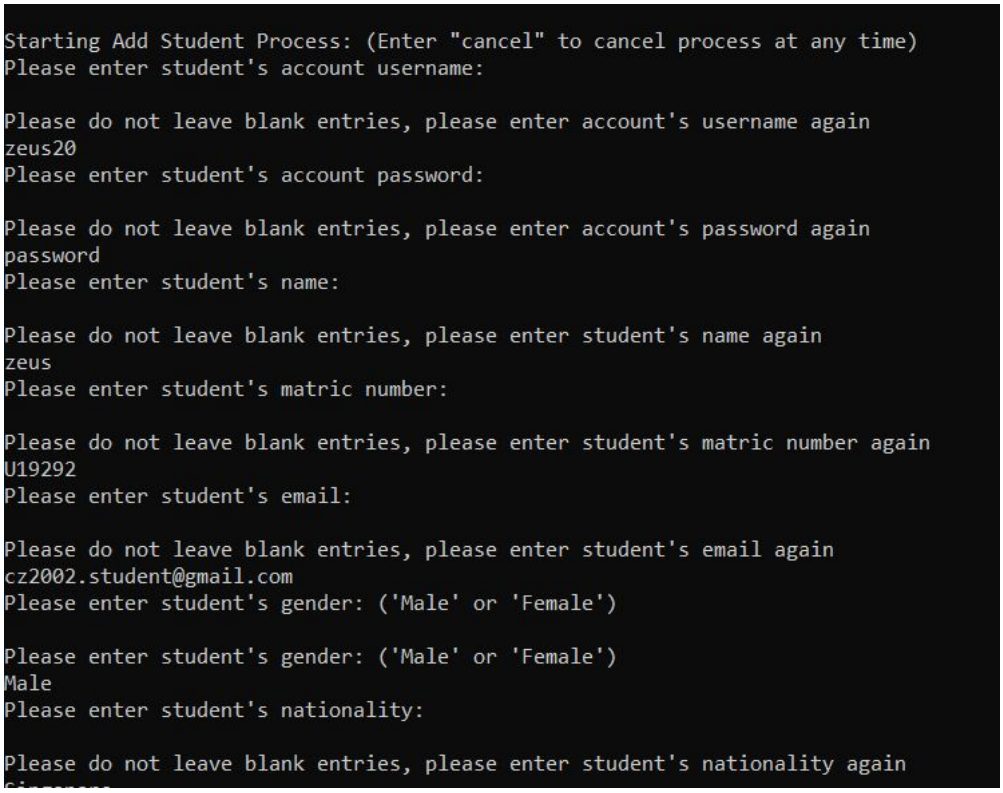
2b. Add an existing student

Checks if the username and matriculation number exists in the database as these two have to be unique for every student in the database.

```
Starting Add Student Process: (Enter "cancel" to cancel process at any time)
Please enter student's account username:
Zteh24
Username already exists, please input a different account's username
Zeus20
Please enter student's account password:
password
Please enter student's name:
Zeus
Please enter student's matric number:
U123456E
Matric number already exists, please input a different matric number
```

2c. Invalid data entries

Ensures that the admin does not leave any of the fields blank when creating a new student account.



3. Add a course

   3a. Add a new course (with combination of (ii) from above)

```
Please enter the course code:
et0001
Please enter the name for this course:
ENTERPRISE & INNOVATION
Please enter the number of AUs for this course:
1
Please enter School of the Course:
SCSE
Please enter the list of indexes that the course has (Eg. '64001,64003'):
10300
You have entered the following indexes:
10300
Are you sure to proceed with these indexes? ('Y' or 'N')
y
Please enter vacancy space for each class index:
Class vacancy for index number 10300 :
10
=====SUMMARY=====
Course Code:            ET0001
Course Name:            ENTERPRISE & INNOVATION
AU:             1
School:         SCSE
---Index List---
1. 10300                Vacancy: 10
Are you sure to proceed with adding course? ('Y' or 'N')
y
Courses added successfully.

Press the "ENTER" key to be directed back to STARS main menu!

Course Code: CZ2001, Course Name: ALGORITHMS, Course Index: 10124
Course Code: CZ2001, Course Name: ALGORITHMS, Course Index: 10186
Course Code: CZ2002, Course Name: OBJECT ORIENTED DESIGN & PROGRAMMING, Course Index: 10126
Course Code: CZ2002, Course Name: OBJECT ORIENTED DESIGN & PROGRAMMING, Course Index: 10200
Course Code: CZ2005, Course Name: OPERATING SYSTEMS, Course Index: 18001
Course Code: CZ2005, Course Name: OPERATING SYSTEMS, Course Index: 18002
Course Code: CZ2005, Course Name: OPERATING SYSTEMS, Course Index: 10239
Course Code: CZ2007, Course Name: INTRODUCTION TO DATABASES, Course Index: 10255
Course Code: EE8087, Course Name: LIVING WITH MATHEMATICS, Course Index: 35001
Course Code: HE9091, Course Name: PRINCIPLES OF ECONOMICS, Course Index: 00551
Course Code: HE9091, Course Name: PRINCIPLES OF ECONOMICS, Course Index: 00552
Course Code: HE9091, Course Name: PRINCIPLES OF ECONOMICS, Course Index: 00553
Course Code: ET0001, Course Name: ENTERPRISE & INNOVATION, Course Index: 10268
Course Code: ET0001, Course Name: ENTERPRISE & INNOVATION, Course Index: 10269
Course Code: ET0001, Course Name: ENTERPRISE & INNOVATION, Course Index: 10300


1. Add Course
2. Update Class Schedule of a Course
3. Update course vacancy:
0. Exit

Enter the number of your choice:
```

3b. Add an existing course

```
Starting Add Course Process: (Enter "cancel" to cancel process)
Please enter the course code:
ET0001
Please enter the name for this course:
ENTERPRISE & INNOVATION
Please enter the number of AUs for this course:
1
Please enter School of the Course:
SCSE
Please enter the list of indexes that the course has (Eg. '64001,64003'):
10268
You have entered the following indexes:
10268
Are you sure to proceed with these indexes? ('Y' or 'N')
Y
Please enter vacancy space for each class index:
Class vacancy for index number 10268 :
10
=====SUMMARY=====
Course Code:            ET0001
Course Name:            ENTERPRISE & INNOVATION
AU:             1
School:         SCSE
---Index List---
1. 10268                Vacancy: 10
Are you sure to proceed with adding course? ('Y' or 'N')
Y
Courses could not be added, the index entered exists already.

Press the "ENTER" key to be directed back to STARS main menu!
```

3c. Invalid data entries

```
Starting Add Course Process: (Enter "cancel" to cancel process)
Please enter the course code:

Please enter the name for this course:

Please enter the number of AUs for this course:

Invalid Input, please enter again
```

4. Register student for a course

4a. Add a student to a course index with available vacancies.

```
Starting Add Course Process: (Enter "cancel" to cancel process)
Please enter course index:
10200
=====SUMMARY=====
Course Code:            CZ2002
Course Name:            Object Oriented Design & Programming
AU:             3
School:         SCSE
Index:          10200
---Lecture Schedule---
Week:WEEKLY     Friday          11:30 - 12:30    Location:LT2A
---Lab Schedule---
Week:ODD        Friday          12:30 - 14:30    Location:SPL
---Tutorial Schedule---
Week:WEEKLY     Monday          09:30 - 10:30    Location:LHN-TR+17
============================
Are you sure to proceed with adding course Info? ('Y' or 'N')
y
Courses added successfully.

Press the "ENTER" key to be directed back to the previous menu!
```

4b. Add a student to a course index with 0 vacancies in Tut / Lab.

```
Starting Add Course Process: (Enter "cancel" to cancel process)
Please enter course index:
10268
=====SUMMARY=====
Course Code:            ET0001
Course Name:            ENTERPRISE & INNOVATION
AU:             1
School:         SCSE
Index:          10268
---Lecture Schedule---
No schedule found.
---Lab Schedule---
No schedule found.
---Tutorial Schedule---
No schedule found.
============================
Are you sure to proceed with adding course Info? ('Y' or 'N')
y
Course Vacancy is full. You have been put on waitlist for the course.

Press the "ENTER" key to be directed back to the previous menu!
```

4c. Register the same course again

```
Starting Add Course Process: (Enter "cancel" to cancel process)
Please enter course index:
10186
=====SUMMARY=====
Course Code:            CZ2001
Course Name:            Algorithms
AU:             3
School:         SCSE
Index:          10186
---Lecture Schedule---
Week:WEEKLY     Friday          10:30 - 11:30    Location:ONLINE
---Lab Schedule---
Week:EVEN       Monday          14:30 - 16:30    Location:HWLAB1
---Tutorial Schedule---
Week:WEEKLY     Tuesday         11:30 - 12:30    Location:TR+16
============================
Are you sure to proceed with adding course Info? ('Y' or 'N')
y

Course adding failed. Selected course is taken by you!! Press the "ENTER" key to be directed back to the previous menu!
```

4d. Invalid data entries (eg wrong student ID / course code, etc)

```
Starting Add Course Process: (Enter "cancel" to cancel process)
Please enter course index:
91203809123

There are no records of course index entered. Press the "ENTER" key to be directed back to the previous menu!
```

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

5a. Check for vacancy in course index

```
Starting Check Course Vacancy Process: (Enter "cancel" to cancel process)
Please enter course index:
00553
Course vacancy of index '00553' is 10

Press the "ENTER" key to be directed back to the previous menu!
```

5b. Invalid data entries (eg course code, class code etc)

```
Starting Check Course Vacancy Process: (Enter "cancel" to cancel process)
Please enter course index:
aosdjpaosj

There are no records of course index entered. Press the "ENTER" key to be directed back to the previous menu!
```

6. Day/Time clash with other course

6a. Add a student to a course index with available vacancies.

```
Starting Check Course Vacancy Process: (Enter "cancel" to cancel process)
Please enter course index:
10127
Course vacancy of index '10127' is 10

Press the "ENTER" key to be directed back to the previous menu!


Welcome to STARS (Student): Zen
1. Add Course
2. Drop Course
3. Check/Print Courses Registered
4. Check Vacancies Available
5. Change Index Number of Course
6. Swap Index Number with Another Student
0. Exit

Enter the number of your choice: 1

Starting Add Course Process: (Enter "cancel" to cancel process)
Please enter course index:
10127
=====SUMMARY=====
Course Code:            CZ2002
Course Name:            Object Oriented Design & Programming
AU:             3
School:         SCSE
Index:          10127
---Lecture Schedule---
Week:WEEKLY     Monday          13:30 - 14:30    Location:LT2
---Lab Schedule---
No schedule found.
---Tutorial Schedule---
No schedule found.
============================
Are you sure to proceed with adding course Info? ('Y' or 'N')
Y

Course adding failed. Selected course clashes with your schedule!! Press the "ENTER" key to be directed back to the previous menu!
```

7. Waitlist notification

7a(i). Add studentA to a course index with 0 vacancies

```
Starting Add Course Process: (Enter "cancel" to cancel process)
Please enter course index:
10268
=====SUMMARY=====
Course Code:            ET0001
Course Name:            ENTERPRISE & INNOVATION
AU:             1
School:         SCSE
Index:          10268
---Lecture Schedule---
No schedule found.
---Lab Schedule---
No schedule found.
---Tutorial Schedule---
No schedule found.
============================
Are you sure to proceed with adding course Info? ('Y' or 'N')
y
Course Vacancy is full. You have been put on waitlist for the course.

Press the "ENTER" key to be directed back to the previous menu!
```

7a(ii). Drop studentB from the same course index

```
Starting Drop Course Process: (Enter "cancel" to cancel process)
Please enter course index:
10268
=====SUMMARY=====
Course Code:            ET0001
Course Name:            ENTERPRISE & INNOVATION
AU:             1
School:         SCSE
Index:          10268
---Lecture Schedule---
No schedule found.
---Lab Schedule---
No schedule found.
---Tutorial Schedule---
No schedule found.
============================
Are you sure to proceed with dropping this course index? ('Y' or 'N')
y
Course dropped successfully.

Press the "ENTER" key to be directed back to the previous menu!
```

7a(iii). Display studentA timetable

(Before)

```
=====Registered Courses=====

Course:CZ2001   AU:3    Index No.:10186 Status:ACCEPTED
Course:CZ2005   AU:3    Index No.:18002 Status:ACCEPTED
Course:CZ2007   AU:3    Index No.:10255 Status:ACCEPTED
Course:ET0001   AU:1    Index No.:10268 Status:WAITLIST
Total AU: 10
============================
```

(After)

```
Enter the number of your choice: 3
=====Registered Courses=====

Course:CZ2001   AU:3    Index No.:10186 Status:ACCEPTED
Course:CZ2005   AU:3    Index No.:18002 Status:ACCEPTED
Course:CZ2007   AU:3    Index No.:10255 Status:ACCEPTED
Course:ET0001   AU:1    Index No.:10268 Status:ACCEPTED
Total AU: 10
============================
```

8. Print student list by index number, course

8a. Print list by (i) Course

```
Please enter the course code to print student list:
CZ2001
=====Student list=====
Name:Tester     Gender:Male     Nationality:American
Name:Zen        Gender:Male     Nationality:Singaporean
Name:Kyle       Gender:Male     Nationality:Singaporean
Name:Mindy      Gender:Female   Nationality:Singaporean
Name:Erric      Gender:Male     Nationality:Singaporean
Name:Rekky      Gender:Female   Nationality:Swedish
Name:Halberg    Gender:Male     Nationality:Singaporean
Name:Umpire     Gender:Male     Nationality:French
Name:Joe        Gender:Female   Nationality:Singaporean
Press the "ENTER" key to be directed back to the previous menu!
```

8a. Print list by (ii) index

```
Please enter the course index to print student list:
10186
=====Student list=====
Name:Tester     Gender:Male     Nationality:American
Name:Zen        Gender:Male     Nationality:Singaporean
Name:Kyle       Gender:Male     Nationality:Singaporean
Name:Mindy      Gender:Female   Nationality:Singaporean
Name:Erric      Gender:Male     Nationality:Singaporean
Press the "ENTER" key to be directed back to the previous menu!
```

8b. Invalid data entries (eg course code, index code etc)

```
Enter the number of your choice: 6
Please enter the course code to print student list:
jdosajd

There are no records of course code entered. Press the "ENTER" key to be directed back to the previous menu!
```