

## **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	Lab Group	Signature	Date
Bryson Teo Yuan Harn	CZ2002	SS13		24/11/2020
Lee Chia Zhe	CZ2002	SS13		24/11/2020
Liu Zhixuan	CZ2002	SS13		24/11/2020
Tan Jia Qing	CZ2002	SS13		24/11/2020
Wong Jia Wen	CZ2002	SS13		24/11/2020

Youtube demo link: <https://youtu.be/5zKnigMFHb8>

# 1 Design Consideration

When designing the program, we kept in mind the basic design principles such as low coupling and high cohesion in our program to avoid rigidity, immobility and fragility in our program. Other than those basic design principles, our program also had considered the 5 SOLID design principles. On top of that, we had also shown the use of Object-Oriented concepts in our program.

## 1.1 Assumption

- Each student will have unique account username and unique matriculation number.
- Each course and index will have unique course code and index number.
- If a student's registered AU has already reached maximum AU(21) when his or her index at the head of the waitlist queue is polled, it will be pushed to the end of the queue to avoid exceeding the maximum AU.
- A student can't add an index where the period of the index clashes with any period of the other indexes registered by the student.
- A student can't add an index of a course which is already in the waitlist or registered list.
- If two students want to swap indexes, the two indexes must belong to the same course code and they must register for the index beforehand and no clash should occur after swapping.
- Students can't drop an index which he/she does not register for.
- Admins can have full access to MySTARS at any period of time. Students can have full access to MySTARS during the add/drop period, other than that students can only perform limited operations.
- Two different courses can have the same index. Each course has unique course code. The index is uniquely identified by course code and index.

## 1.2 Class Stereotypes

### Entity

Entity is classes which are used to store data and attributes which are used by a program application. In our design, the entities included are User, Admin, Student, Period, Course, Index, Class. All these entities store important information which will then be required by the application.

## **Boundary**

Boundary is the interface to the external. In our design, there is one boundary, MySTARS. This boundary is used by the user to access the my star system. For admins, they will use this boundary to manage the system. For students, they will use it to do add/drop of courses during the add/drop period and to check the courses they had registered for the rest of the time.

## **Control**

Control is the object that is intermediate to entity and boundary, they serve in managing the interaction between them. In our design, the control that we had used included AccountManager, RegistrationManager and Email. AccountManager is used to manage user accounts of MySTARS. RegistrationManager is used to manage add/drop and swapping of courses, indexes or classes by students. Email is used to notify the student after successfully registering for a course in MySTARS.

## **1.3 Basic Design Principles**

### **Low Coupling**

We had taken low coupling into consideration when designing our program. We try to achieve low coupling as much as possible by linking classes together only when necessary. Every class will only provide general information to other class. They will process the data themselves if they need.

### **High Cohesion**

We had also taken high cohesion into consideration when designing our program. We avoided creating a class which performs many operations required by the whole program or do something else that should not be done by the class. For example, a student can perform many operations on the index such as add, drop and swap. With high cohesion in mind, we created a new class, RegistrationManager which works specifically on these operations to avoid either the students or index class to perform methods which shouldn't belong to them.

## **1.4 Usage of SOLID Design Principles**

### **Single Responsibility Principle(SRP)**

Single responsibility principle states that every class should only have a single responsibility for the function. All classes in our application have been designed to have only one responsibility. For instance, although the login() method is implemented in MySTARS class, the method itself will only get the user input password but will not validate it. The validation is done by the AccountManager class. The reason for this is that in our program design, MySTARS class should only be responsible to communicate with

external systems. Therefore, the responsibility of validating an account or creating a new login account is with AccountManager class.

## **Open-Closed Principle**

Open closed principle states that a module should open for extension but closed for modification. This principle is illustrated in the RegisterNotification interface and Email class. The RegisterNotification interface is closed for modification, but it is opened for different implementations to change the way of sending notifications without actually changing the source code. This interface is realized by Email class, which has overridden the sendRegisterNotification() method so that users will receive notifications via Email. If we wish to add Whatsapp, SMS or other ways to notify the users when they are allocated for a course, we could just implement the interface and override the method.

## **Liskov Substitution Principle**

Liskov Substitution Principle states the relationship between superclass and subclasses. The subclass objects must be substitutable into superclass objects in a program without changing any properties of the program. In our program, the only inheritance relationship available is between User with Admin and Student class. However, there isn't any class which uses the User class. Therefore, we do not violate this principle.

## **Interface Segregation Principle**

This principle states that many client specific interfaces are better than a general one. With this principle in mind, we avoided creating FAT interfaces whenever applicable. We implement two interfaces in our project that is the TimePeriod interface and the RegisteredNotification interface. The RegisteredNotification has only one abstract method and is impossible to be a FAT interface. While the TimePeriod interface has three abstract methods, we did not split the interface into two or more interfaces which is more specific as the only two classes that implemented this interface need all the three methods. Since there is no class which is forced to implement methods which they do not use, we do not need to split them.

## **Dependency Injection Principle**

This principle states that high level should not depend upon low level modules and both should depend upon abstraction. Abstraction should not depend upon details. Details should depend upon abstractions. One of the example in our code is the implementation of the Email class. It implements the RegisterNotification interface with sendRegisterNotification method. The Index class send the notification by using the RegisterNotification, and Email instance will be pass in to specify how the notification will be send(in this case only Email are implemented). This allows us to implement different types of notification such

as Whatsapp and SMS and doesn't need to modify the code in the Index class. Instead, we just need to change the instance we pass to the correspond method.

## 1.5 Usage of OO Concepts

### Abstraction

One example of abstraction is between the User class and Course class. Both classes have their own set of attributes and functions. All these identify uniquely the User objects and Course objects from all other objects.

### Encapsulation

We had shown encapsulation in our program where all class data is declared as private. Direct access of a class data by other classes is not possible. Other classes can only access the data through the public method of the class, the so-called getter and setter.

### Inheritance

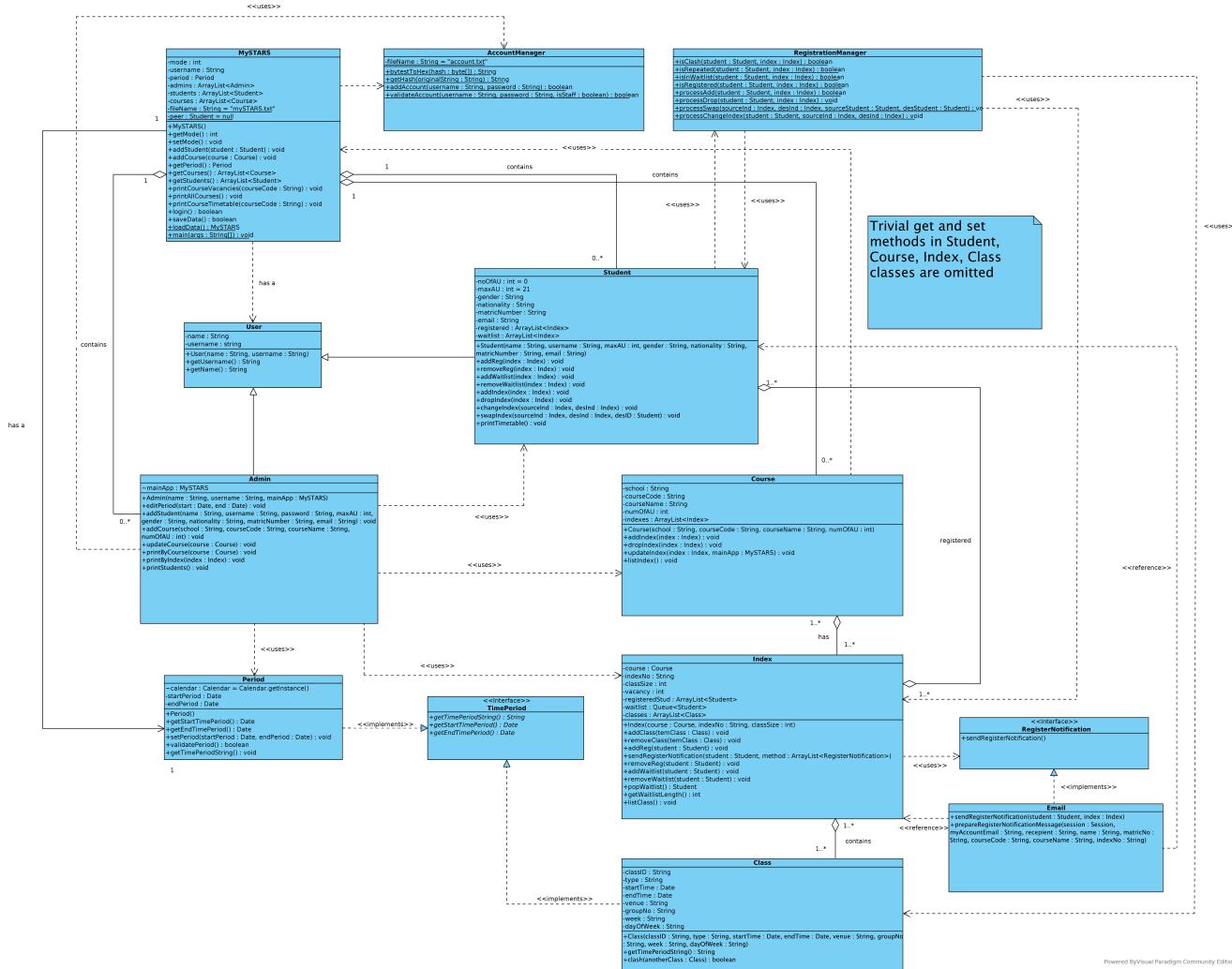
This concept is shown by the Student and Admin classes. The similar attributes between Student and Admin are declared under the User class. Both of these classes extend the User class and they also define sets of their own attributes in their own class.

### Polymorphism

One of the example in our code is RegisterNotification interface and the Email class. The Index class is depends on the RegisterNotification, and calling the sendRegisterNotification abstract method. Hence, this will cater to any class that implement this interface. For now, only Email class is implemented. But various sending method could be added in the future and used by the Index class, without changing the code in the Index class.

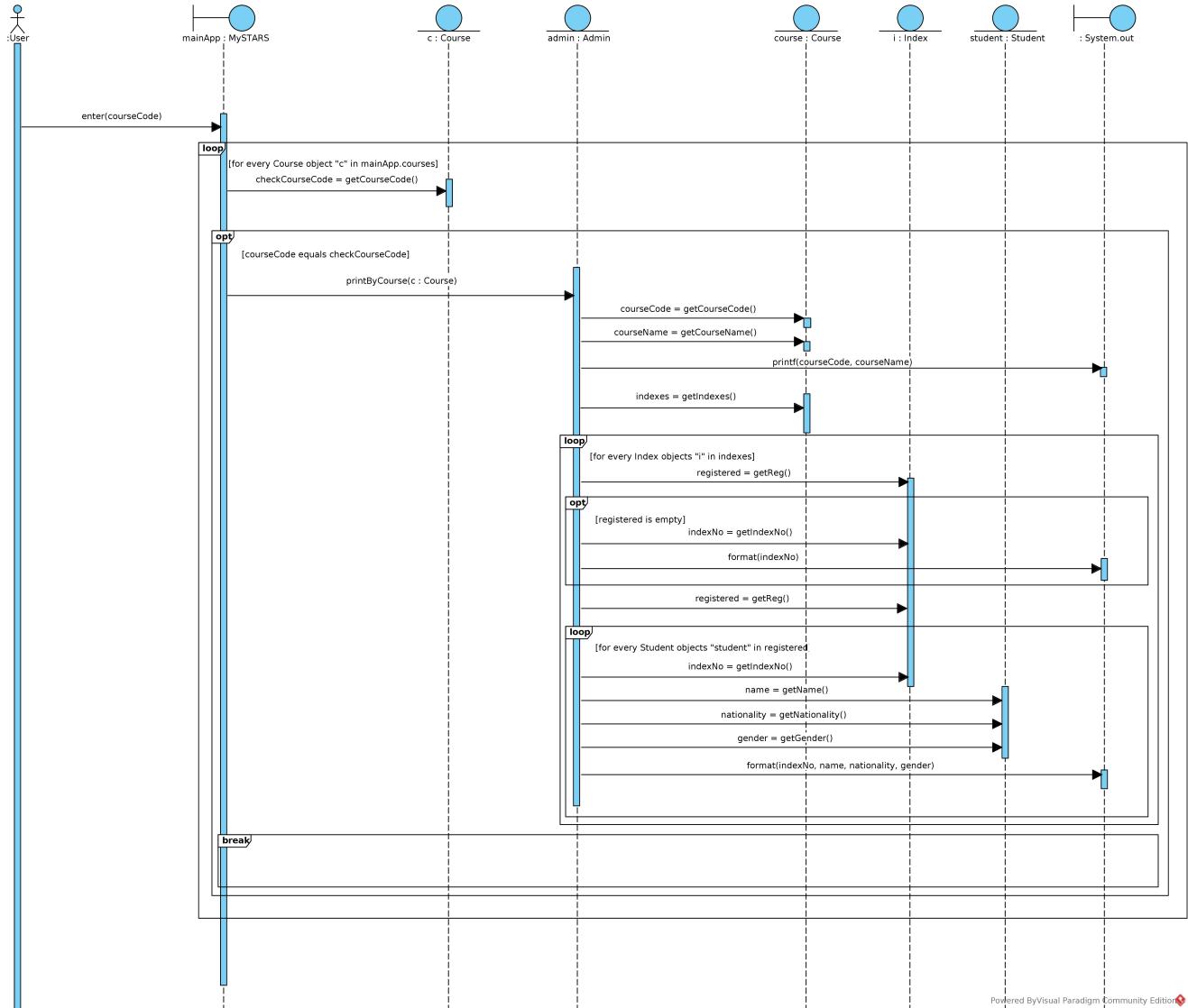
## 2 Detailed UML Class Diagram

Following is the class diagram of the program. Zoom in and wait the picture to render the high resolution image.



### 3 Detailed UML Sequence Diagram of stated function

Following is the sequence diagram of print student list by course. Zoom in and wait the picture to render the high resolution image.



## 4 Testing

### 4.1 Student login

```
MySTARS@6ae40994
(1)Admin      (2)Student
Mode: 2
Username: andy001
Password:

// ----- Welcome to MySTARS! -----
You are not allowed register course now, the registration period is:
Start period: 01 Dec 2020(Tue) 00:00
End period: 31 Dec 2020(Thu) 00:00

(1) Check/print registered course
(2) Check all courses
(3) Check vacancy of course
(4) Check timetable of course
(5) Exit
Your choice: 
```

(a) Not in registration period

```
MySTARS@6ae40994
(1)Admin      (2)Student
Mode: 2
Username: andy001
Password:

// ----- Welcome to MySTARS! -----
(1) Add Course
(2) Drop Course
(3) Check/print courses registered
(4) Check all courses
(5) Check vacancy of course
(6) Check timetable of course
(7) Change index number of course
(8) Swap index number with another student
(9) Exit
Your choice: 
```

(b) During registration period

Figure 1: Student login during different period

```
MySTARS@6ae40994
(1)Admin      (2)Student
Mode: 2
Username: andy001
Password:
Wrong username or password! You can try for 3 more time(s).
Username: andy001
Password:
Wrong username or password! You can try for 2 more time(s).
Username: andy001
Password:
Wrong username or password! You can try for 1 more time(s).
Username: andy001
Password:
Invalid Credential!
```

Figure 2: Student key in wrong password

## 4.2 Add student

```
// ----- Welcome to MySTARS! -----
(1) Edit registration period
(2) Check registration period
(3) Add student
(4) Add course
(5) Update course
(6) Check all courses
(7) Check vacancy of course
(8) Check timetable of course
(9) Print student list
(10) Print student list by course
(11) Print student list by index number
(12) Quit
Your choice: 3
Please enter student's name: Wesley Yap
Please enter student's username: wes1001
Please enter student's password: IloveJava!
Please enter student's maximum AU: 24
Please enter student's gender (Female/Male): Male
Please enter student's nationality: Singaporean
Please enter student's matriculation number: U20457288
Please enter student's email account: wes1001@e.ntu.edu.sg
Student added!

Student List
+-----+
|   Name    | Username | Matric No | Nationality | Gender |
+-----+
| Andy Lau  | andy001  | U1920001A  | Singaporean | Male   |
| Bryson Teo | bryson002 | U1920002A  | Malaysian   | Male   |
| Cody Hong  | cody003  | U1920003A  | Malaysian   | Male   |
| Daron Lim   | daron004  | U1920004A  | American    | Male   |
| Emily Kim   | emily005  | U1920005A  | Korean      | Female |
| Fatima Tan  | fatima006  | U1920006A  | Singaporean | Female |
| Gavin Chong  | gavin007  | U1920007A  | Singaporean | Male   |
| Hayley Tan  | hayley008  | U1920008A  | American    | Female |
| Isaac Newton | isaac009  | U1920009A  | British     | Male   |
| Jay Chou    | jay010   | U1920010A  | Taiwanese   | Male   |
| Kingston Wong | kingston011 | U1920011A  | Singaporean | Male   |
| Lucas Tan   | lucas012  | U1920012A  | Singaporean | Male   |
| Mandy Wong   | mandy013  | U1920013A  | Singaporean | Female |
| Nathan Lim   | nathan014  | U1920014A  | Singaporean | Male   |
| Ogei Chong   | ogei015   | U1920015A  | Japanese    | Female |
| Wesley Yap  | wes1001   | U20457288  | Singaporean | Male   |
+-----+
```

(a) Add a new student

```
// ----- Welcome to MySTARS! -----
(1) Edit registration period
(2) Check registration period
(3) Add student
(4) Add course
(5) Update course
(6) Check all courses
(7) Check vacancy of course
(8) Check timetable of course
(9) Print student list
(10) Print student list by course
(11) Print student list by index number
(12) Quit
Your choice: 3
Please enter student's name: Andy Lau Tak
Please enter student's username: andy001
Student with this username already exist!
```

(b) Add an existing student

Figure 3: Add a student

```
// ----- Welcome to MySTARS! -----
(1) Edit registration period
(2) Check registration period
(3) Add student
(4) Add course
(5) Update course
(6) Check all courses
(7) Check vacancy of course
(8) Check timetable of course
(9) Print student list
(10) Print student list by course
(11) Print student list by index number
(12) Quit
Your choice: 3
Please enter student's name: Yvonne Tee
Please enter student's username: yvonne023
Please enter student's password: yvonne023
Please enter student's maximum AU: qw
Please enter integer!
Please key in again: 24
Please enter student's gender (Female/Male): F
Invalid input! Please enter Female / Male !
Please enter student's gender (Female/Male): Female
Please enter student's nationality: Singaporean
Please enter student's matriculation number: U2037712K
Please enter student's email account: yvonne023@e.ntu.edu.sg
Student added!
```

Figure 4: Invalid data entries

## 4.3 Add course

```
// ----- Welcome to MySTARS! -----
(1) Edit registration period
(2) Check registration period
(3) Add student
(4) Add course
(5) Update course
(6) Check all courses
(7) Check vacancy of course
(8) Check timetable of course
(9) Print student list
(10) Print student list by course
(11) Print student list by index number
(12) Quit
Your choice: 4
School List: NBS, CBE, CEE, SCSE, EEE, MSE, MAE, ADM, SoH, SoSS, WKWSCI, SBS, SPMS, ASE, LKCSoM, NIE, RSIS
Please choose school of the course: soh
Please enter the course code: LJ9001
Please enter the course name: Japanese Language Level 1
Please enter the number of AU: 3
Course added!
+-----+
| Course Code | Course Name | School | AU |
+-----+
| CZ2001 | Algorithms | SCSE | 3 |
| CZ2002 | Object-oriented Design & Programming | SCSE | 3 |
| HE9091 | Principles of Economics | SSS | 3 |
| BU8201 | Business Finance | NBS | 3 |
| LJ9001 | Japanese Language Level 1 | SOH | 3 |
+-----+
```

(a) Add a new course

```
// ----- Welcome to MySTARS! -----
(1) Edit registration period
(2) Check registration period
(3) Add student
(4) Add course
(5) Update course
(6) Check all courses
(7) Check vacancy of course
(8) Check timetable of course
(9) Print student list
(10) Print student list by course
(11) Print student list by index number
(12) Quit
Your choice: 4
School List: NBS, CBE, CEE, SCSE, EEE, MSE, MAE, ADM, SoH, SoSS, WKWSCI, SBS, SPMS, ASE, LKCSoM, NIE, RSIS
Please choose school of the course: soh
Please enter the course code: LJ9001
This course code already exist!
```

(b) Add an existing course

Figure 5: Add a course

```
// ----- Welcome to MySTARS! -----
(1) Edit registration period
(2) Check registration period
(3) Add student
(4) Add course
(5) Update course
(6) Check all courses
(7) Check vacancy of course
(8) Check timetable of course
(9) Print student list
(10) Print student list by course
(11) Print student list by index number
(12) Quit
Your choice: 4
School List: NBS, CBE, CEE, SCSE, EEE, MSE, MAE, ADM, SoH, SoSS, WKWSCI, SBS, SPMS, ASE, LKCSoM, NIE, RSIS
Please choose school of the course: computer
Please choose school from the list shown!
School List: NBS, CBE, CEE, SCSE, EEE, MSE, MAE, ADM, SoH, SoSS, WKWSCI, SBS, SPMS, ASE, LKCSoM, NIE, RSIS
Please choose school of the course: SCSE
Please enter the course code: CZ2006
Please enter the course name: Software Engineering
Please enter the number of AU: w
Please enter integer!
Please key in again: 3
Course added!
+-----+
| Course Code | Course Name | School | AU |
+-----+
| CZ2001 | Algorithms | SCSE | 3 |
| CZ2002 | Object-oriented Design & Programming | SCSE | 3 |
| HE9091 | Principles of Economics | SSS | 3 |
| BU8201 | Business Finance | NBS | 3 |
| LJ9001 | Japanese Language Level 1 | SOH | 3 |
| CZ2007 | Introduction to Database | SCSE | 3 |
| CZ2006 | Software Engineering | SCSE | 3 |
+-----+
```

Figure 6: Invalid data entries

## 4.4 Student register course

```
// ----- Welcome to MySTARS! -----
(1) Add Course
(2) Drop Course
(3) Check/print courses registered
(4) Check all courses
(5) Check vacancy of course
(6) Check timetable of course
(7) Change index number of course
(8) Swap index number with another student
(9) Exit
Your choice: 5
Please enter the course code: LJ9001

LJ9001: Japanese Language Level 1
+-----+
| Index No | Class Size | Vacancy | Waitlist |
+-----+
| 18054    | 1          | 1        | 0        |
+-----+
```

// ----- Welcome to MySTARS! -----
(1) Add Course
(2) Drop Course
(3) Check/print courses registered
(4) Check all courses
(5) Check vacancy of course
(6) Check timetable of course
(7) Change index number of course
(8) Swap index number with another student
(9) Exit
Your choice: 1
Please enter the index number: 18054
Preparing to send email to jwong097@e.ntu.edu.sg
Email sent successfully to jwong097@e.ntu.edu.sg!
Successfully registered LJ9001 Japanese Language Level 1
Index: 18054

(a) Register course index with available vacancy

```
// ----- Welcome to MySTARS! -----
(1) Add Course
(2) Drop Course
(3) Check/print courses registered
(4) Check all courses
(5) Check vacancy of course
(6) Check timetable of course
(7) Change index number of course
(8) Swap index number with another student
(9) Exit
Your choice: 5
Please enter the course code: LJ9001

LJ9001: Japanese Language Level 1
+-----+
| Index No | Class Size | Vacancy | Waitlist |
+-----+
| 18054    | 1          | 0        | 0        |
+-----+
```

// ----- Welcome to MySTARS! -----
(1) Add Course
(2) Drop Course
(3) Check/print courses registered
(4) Check all courses
(5) Check vacancy of course
(6) Check timetable of course
(7) Change index number of course
(8) Swap index number with another student
(9) Exit
Your choice: 1
Please enter the index number: 18054
Successfully registered LJ9001 Japanese Language Level 1
Index: 18054 into waitlist

(b) Register course index with 0 vacancy

Figure 7: Student register for a course

```
// ----- Welcome to MySTARS! -----
(1) Add Course
(2) Drop Course
(3) Check/print courses registered
(4) Check all courses
(5) Check vacancy of course
(6) Check timetable of course
(7) Change index number of course
(8) Swap index number with another student
(9) Exit
Your choice: 1
Please enter the index number: 18054
You have already taken this course!
```

(a) Register the same course again

```
// ----- Welcome to MySTARS! -----
(1) Add Course
(2) Drop Course
(3) Check/print courses registered
(4) Check all courses
(5) Check vacancy of course
(6) Check timetable of course
(7) Change index number of course
(8) Swap index number with another student
(9) Exit
Your choice: 1
Please enter the index number: 134
The index number does not exists!
```

(b) Index number not exists

Figure 8: Invalid data entries

## 4.5 Check available slot(vacancy) in a class

```
// ----- Welcome to MySTARS! -----
(1) Add Course
(2) Drop Course
(3) Check/print courses registered
(4) Check all courses
(5) Check vacancy of course
(6) Check timetable of course
(7) Change index number of course
(8) Swap index number with another student
(9) Exit
Your choice: 5
Please enter the course code: CZ2002

CZ2002: Object-oriented Design & Programming
+-----+
| Index No | Class Size | Vacancy | Waitlist |
+-----+
| 10126    | 10          | 9       | 0       |
| 10127    | 10          | 10      | 0       |
+-----+
```

(a) Check for vacancy in course index

```
// ----- Welcome to MySTARS! -----
(1) Add Course
(2) Drop Course
(3) Check/print courses registered
(4) Check all courses
(5) Check vacancy of course
(6) Check timetable of course
(7) Change index number of course
(8) Swap index number with another student
(9) Exit
Your choice: 5
Please enter the course code: qwert
The course code doesn't exists!
```

(b) Invalid data entries

Figure 9: Check available slot(vacancy) in a class

## 4.6 Day/Time clashes with other course

```
MySTARS@6ae40994
(1)Admin      (2)Student
Mode: 2
Username: jwong097
Password:

// ----- Welcome to MySTARS! -----
(1) Add Course
(2) Drop Course
(3) Check/print courses registered
(4) Check all courses
(5) Check vacancy of course
(6) Check timetable of course
(7) Change index number of course
(8) Swap index number with another student
(9) Exit
Your choice: 1
Please enter the index number: 10125
Preparing to send email to jwong097@e.ntu.edu.sg
Email sent successfully to jwong097@e.ntu.edu.sg!
Successfully registered CZ2001 Algorithms
Index: 10125

// ----- Welcome to MySTARS! -----
(1) Add Course
(2) Drop Course
(3) Check/print courses registered
(4) Check all courses
(5) Check vacancy of course
(6) Check timetable of course
(7) Change index number of course
(8) Swap index number with another student
(9) Exit
Your choice: 1
Please enter the index number: 10126
This index's timeslot clashes with your other registered indexes / indexes in waitlist!
```

Figure 10: Add a student to a course index with available vacancies, but clashes with his/her registered/in-waitlist courses

## 4.7 Waitlist notification

```

Mode: 2
Username: wjwen5
Password:
// ----- Welcome to MySTARS! -----
(1) Add Course
(2) Drop Course
(3) Check/print courses registered
(4) Check all courses
(5) Check vacancy of course
(6) Check timetable of course
(7) Change index number of course
(8) Swap index number with another student
(9) Exit
Your choice: 5
Please enter the course code: LJ9001

LJ9001: Japanese Language Level 1
+-----+
| Index No | Class Size | Vacancy | Waitlist |
+-----+
| 18054    | 1           | 0       | 0       |
+-----+

// ----- Welcome to MySTARS! -----
(1) Add Course
(2) Drop Course
(3) Check/print courses registered
(4) Check all courses
(5) Check vacancy of course
(6) Check timetable of course
(7) Change index number of course
(8) Swap index number with another student
(9) Exit
Your choice: 1
Please enter the index number: 18054
Successfully registered LJ9001 Japanese Language Level 1
Index: 18054 into waitlist

```

```

MySTARS@1a93a7ca
(1)Admin      (2)Student
Mode: 2
Username: jwong097
Password:
// ----- Welcome to MySTARS! -----
(1) Add Course
(2) Drop Course
(3) Check/print courses registered
(4) Check all courses
(5) Check vacancy of course
(6) Check timetable of course
(7) Change index number of course
(8) Swap index number with another student
(9) Exit
Your choice: 2
Please enter the index number: 18054
Preparing to send email to wjwen5@gmail.com
Email sent successfully to wjwen5@gmail.com!
Successfully deregistered LJ9001 Japanese Language Level 1
Index: 18054

```

(a) Add student A to course index with 0 vacancies

(b) Drop another student B from the same course index

```

Mode: 2
Username: wjwen5
Password:
// ----- Welcome to MySTARS! -----
(1) Add Course
(2) Drop Course
(3) Check/print courses registered
(4) Check all courses
(5) Check vacancy of course
(6) Check timetable of course
(7) Change index number of course
(8) Swap index number with another student
(9) Exit
Your choice: 3
Registered courses timetable:
+-----+
| Weekday | Time   | Course Code | Index | Class Type | Group Number | Venue | Week |
+-----+
| MONDAY  | 09:30-11:00 | LJ9001    | 18054 | TUT        | 01          | SEM6  | BOTH  |
| TUESDAY |
| WEDNESDAY|
| THURSDAY|
| FRIDAY   |
| SATURDAY|
| SUNDAY   |
+-----+

Registered Courses:
+-----+
| Course Code | Course Name   | Index |
+-----+
| LJ9001     | Japanese Language Level 1 | 18054 |
+-----+

```

(c) Show student A timetable

Figure 11: Student in waitlist registered for the course when some student drop the same course

## 4.8 Print student list by course/index number

```
// ----- Welcome to MySTARS! -----
(1) Edit registration period
(2) Check registration period
(3) Add student
(4) Add course
(5) Update course
(6) Check all courses
(7) Check vacancy of course
(8) Check timetable of course
(9) Print student list
(10) Print student list by course
(11) Print student list by index number
(12) Quit
Your choice: 10
Please enter the course code: CZ2002

Registered student of CZ2002: Object-oriented Design & Programming
+-----+
| Index No | Student Name | Nationality | Gender |
+-----+
| 10126   | Cody Hong      | Malaysian   | Male    |
|          | Hayley Tan     | American    | Female   |
|          | Ogei Chong     | Japanese    | Female   |
| 10127   |
+-----+
```

(a) By course

```
// ----- Welcome to MySTARS! -----
(1) Edit registration period
(2) Check registration period
(3) Add student
(4) Add course
(5) Update course
(6) Check all courses
(7) Check vacancy of course
(8) Check timetable of course
(9) Print student list
(10) Print student list by course
(11) Print student list by index number
(12) Quit
Your choice: 11
Please enter the index number: 10126

Registered student of Index 10126 - CZ2002: Object-oriented Design & Programming
+-----+
| Student Name | Nationality | Gender |
+-----+
| Cody Hong    | Malaysian   | Male    |
| Hayley Tan   | American    | Female   |
| Ogei Chong   | Japanese    | Female   |
+-----+
```

(b) By index number

Figure 12: Print student list

```
// ----- Welcome to MySTARS! -----
(1) Edit registration period
(2) Check registration period
(3) Add student
(4) Add course
(5) Update course
(6) Check all courses
(7) Check vacancy of course
(8) Check timetable of course
(9) Print student list
(10) Print student list by course
(11) Print student list by index number
(12) Quit
Your choice: 10
Please enter the course code: df
The course name does not exists!

// ----- Welcome to MySTARS! -----
(1) Edit registration period
(2) Check registration period
(3) Add student
(4) Add course
(5) Update course
(6) Check all courses
(7) Check vacancy of course
(8) Check timetable of course
(9) Print student list
(10) Print student list by course
(11) Print student list by index number
(12) Quit
Your choice: 11
Please enter the index number: 123
The index number does not exists!
```

Figure 13: Invalid data entries

## 4.9 Course Registered Notification Email

Course Registration

 ntumystars@gmail.com <ntumystars@gmail.com>  
1:05 PM 

To: wjwen5@gmail.com

  
**NTU MySTARS**

**Your registered course has been confirmed!**

Name: Wong Jia Wen  
Matric No: U2023455L

We are pleased to inform you that you have been allocated the following course in coming semester:

Course Code	Course Name	Index
U9001	Japanese Language Level 1	18054

Email generated automatically by system  
Digital signature does not required

Figure 14: Email sent automatically to notify student who have been allocated course