TEAM MEMBERS

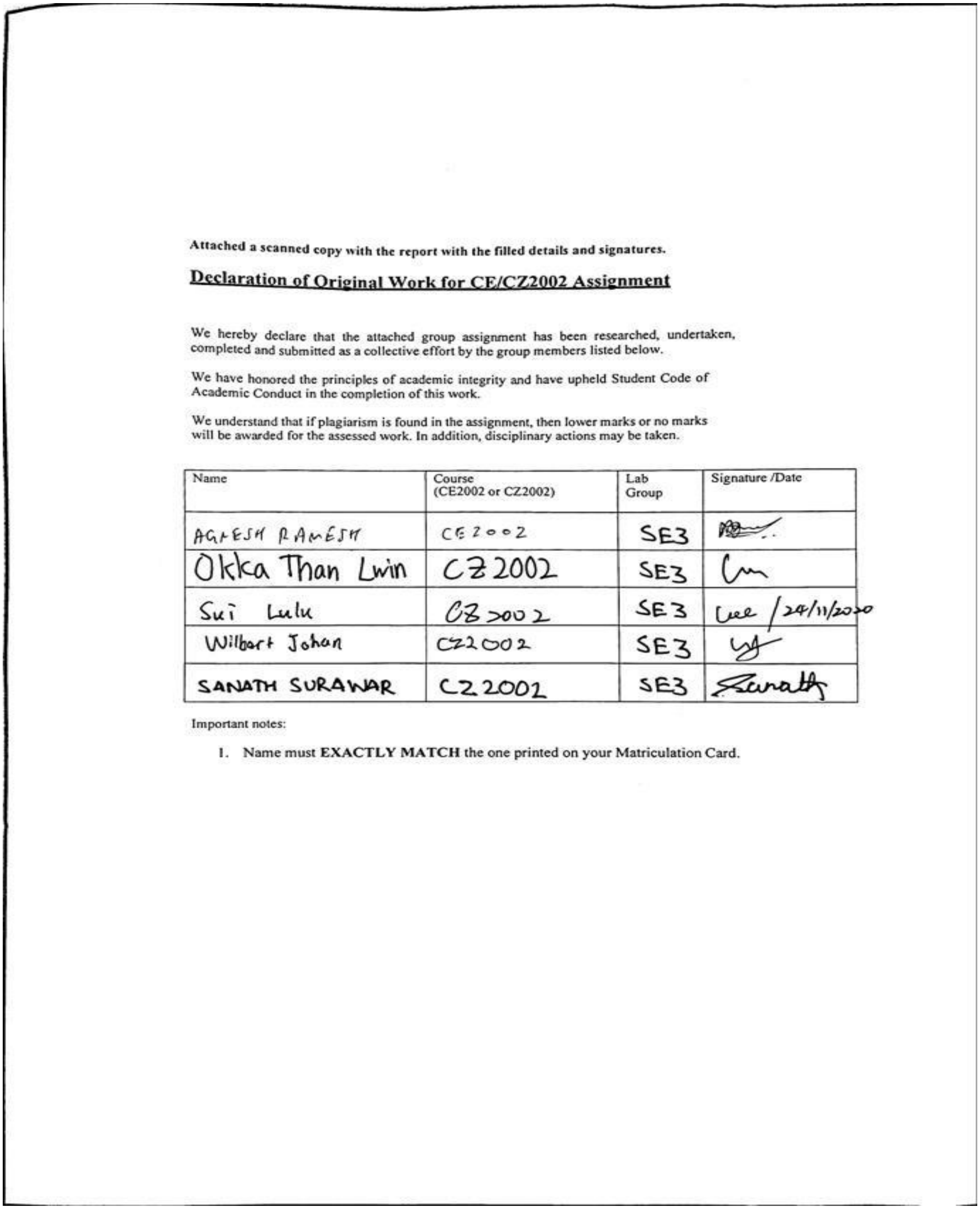
Name	Matriculation Number
Surawar Sanath Sachin	**U1922592K**
Agnesh	**U1921141K**
Sui Lulu	**U1921987A**
Okka Than Lwin	**U1920751J**
Wilbert	**U1922167K**

SE3 GROUP 7

CZ2002 Final Report

Nanyang Technological University

DECLARATION OF ORIGINAL WORK FORM



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
AGNESH RAMESH	CE2002	SE3	
Okka Than Lwin	CZ2002	SE3	
Sui Lulu	CZ2002	SE3	Lee / 24/11/2020
Wilbert Johan	CZ2002	SE3	
SANATH SURAWAR	CZ2002	SE3	Sanath

Important notes:

1. Name must **EXACTLY MATCH** the one printed on your Matriculation Card.

Table of Contents

INTRODUCTION

My STudent Automated Registration System (MySTARS) is a console-based application designed and developed for both staff and students to manage registration of courses. This application encompasses the key features such as creation of new courses, registration of courses and addition of student records.

This report covers the object-oriented programming (OOP) concepts and key design considerations used to implement the application. The design will also be represented in a UML Class Diagram and UML Sequence Diagram for one of the features, showing the relationships and interactions between the objects. Moreover, several test cases have been included to ensure that the requirements of the application that had been set initially, are met.

DESIGN CONSIDERATIONS

a. Method of Approach

- This project was a comprehensive application of OO concepts, both in terms of ensuring proper design and efficiency of code.
- The architectural style that we have adopted is thus the Object-Oriented Architecture, which is a newer form of the call-and-return architecture

b. Design considerations and principles

In our project design, SOLID design principles are extensively applied to make sure the project is easy to maintain and modify the codes with minimal cost by minimizing the impact of changes. The SOLID design principles are namely Single Responsibility Principle, Open/Closed Principle, Liskov Substitution Principle, Interface Segregation Principle, and Dependency Injection Principle. Detailed explanation and implementation examples will be elaborated one by one.

b. 1 Single Responsibility Principle

- This principle means that there should never be more than ONE reason for a class to change, which means high cohesion. To achieve this objective, we try to make all our classes to bear only one responsibility.
- For example, the Menu class only print the menu of the actions that our users can take and then pass on the input of the user to the respective controller classes or the application classes. Since the Menu class has no access to the actual functionality to the respective classes, but merely uses them, any modification done on other classes or functions will not affect the Menu class and vice versa.
- As such, the ripple effect of changes on the codes will be extensively reduced and the minimal modification efforts will be required in the entire STAR system.

b.2 Open/Closed Principle

- This principle means that a module should be open for extension but closed for modification. To achieve this, we make sure subclasses can only extend the functionalities of the superclass but cannot modify the source codes in our project.
- For example, the Student class and Admin class are both extended from the User superclass. However, the User class is closed for the two subclasses to modify it, though the two classes make modification. For instance, the Student class has the extended functions such as setStudent_id, which is not present in the User class.
- Following this principle in our project allows us great flexibility in the future when additional needs arise. For example, if the university needs a class of Deans where this group of staff can access all information on the Admin class, the User class can be easily extended again for a new subclass without great influence on many other classes.

b.3 Liskov Substitution Principle

- This principle means that subtypes must be substitutable for their base types. In terms of class, the derived class is substitutable for its base class if its pre-conditions are no stronger than the base class method, and its post-conditions are no weaker than the base class method.
- In our project, this principle is best illustrated by the method of dropCourse in the fileController. This dropCourse is also used in the studentController and the adminController with no greater expectation and providence than what is in the fileController.

b.4 Interface Segregation Principle

- This principle means that many specific interfaces are better than one general purpose interface so that classes should not depend on interfaces that they do not use.
- To avoid those fat interfaces, we use the MenuUI interface to display menu but the studentMenuUI and the adminMenuUI to display the specific menu for each group. This segregation makes sure fat interfaces are avoided.

b.5 Dependency Injection Principle

- This principle has a two-part meaning. Firstly, a high-level module should not depend upon low level modules, but both should depend upon abstractions. Secondly, abstractions should not depend upon details, but details should depend upon abstractions. In our project, many of our classes depend on abstraction, the abstract interface of Serializable.
- This dependence is beneficial for us to store necessary information in the binaryio file easily. For example, the User class is the higher-level module, and the Student class is the lower-level module. Both classes depend on the abstraction of Serializable. Thus, using abstraction of Serializable allows us to change behaviors and future code evolutions easily.

c. Use of OO concept

c.1 Abstraction

- Abstraction is constantly applied in our project. Besides the User superclass and its Student and Admin subclasses discussed in the Open/Closed Principle, another good illustration is the menuUI and its implementation of studentMenuUI and adminMenuUI.
- Using such interface implementation, we can apply the method displayMenu() more easily in the implementation class using object reference.

c.2 Encapsulation

- Encapsulation or information hiding is essential in our project as the STAR system needs to deal with many private and confidential information such as emails and passwords. This information should be encrypted and not accessible to other classes.
- For example, in the notificationController class, the sender_email and sender_pw attributes are built to be private so that such private information should be encapsulated within the class. Thus, critical information will not be accessible to classes such as fileController and data safety is granted.

c.3 Inheritance

- Inheritance is applied in our project to enhance code reuse. For example, the Student and Admin classes inherit from the User class and use the methods available inside the User class like setUsername() and setPassword() so that repetition is minimized.
- This method overriding make sure that no new methods must be created every time a subclass is created.

c.4 Polymorphism

- Polymorphism allows us to perform a single action in different ways. This is a useful concept in our STAR project as there are some occasions where the same methods are called to perform the tasks slightly different in some classes.
- For example, the function below shows that the menuUI object reference is referred to different types as studentMenuUI and adminMenuUI. Such polymorphism allows you to define one interface of menuUI and have multiple implementations.

```
if(acc.equals("student")) {

menuUI studentmenu=new studentMenuUI();

studentmenu.displayMenu(username);

}

else if(acc.equals("admin")) {

menuUI adminmenu=new adminMenuUI();

adminmenu.displayMenu(username); }
```

c.5 Composition

- Composition is best illustrated in our Course-Index-Schedule association. This is because an index will not exist without a course. Similarly, a schedule will not exist without an index.
- This ‘whole-part’ relationship allows us to reuse code by modeling the “has-a” association between these objects. For example, getWaitlist() method is reused in both Course and Index.

c.6 Generalization and Interface Realization

- The Student class and the Admin class are two specific classifiers of their generalization, the User class. Similarly, the studentMenuUI and the adminMenuUI are the realization of the menuUI interface.
- Such usage of different relationship between the classes and interfaces allows a clearer understanding of the entire system and easy reuse of codes.

d. Data structure

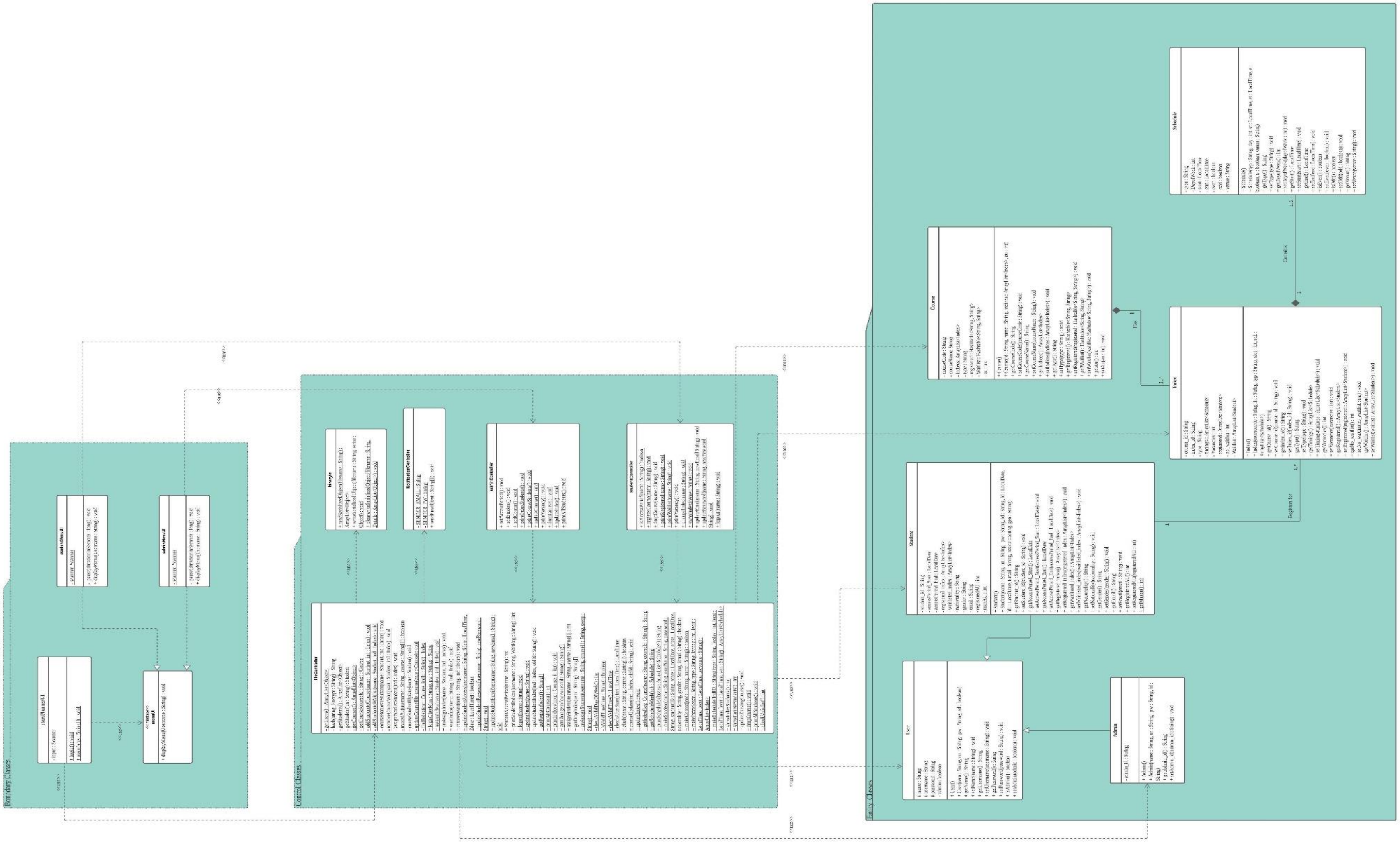
- To read objects or write objects to a file, we use object serialization provided by Java. The data about each user is then stored and updated in a file called binaryio. Such binary file IO makes sure our data are stored and accessed as a sequence of bytes. Binary files are chosen because they are more efficient and the speed of access of data is faster.
- Additionally, as the data is stored using numeric formats, these files will also take up less memory spaces. This advantage is crucial in our project as the STAR system is needed to every member in a university and the huge amount of data resulted will require such efficient files.

e. Assumptions & Limitations

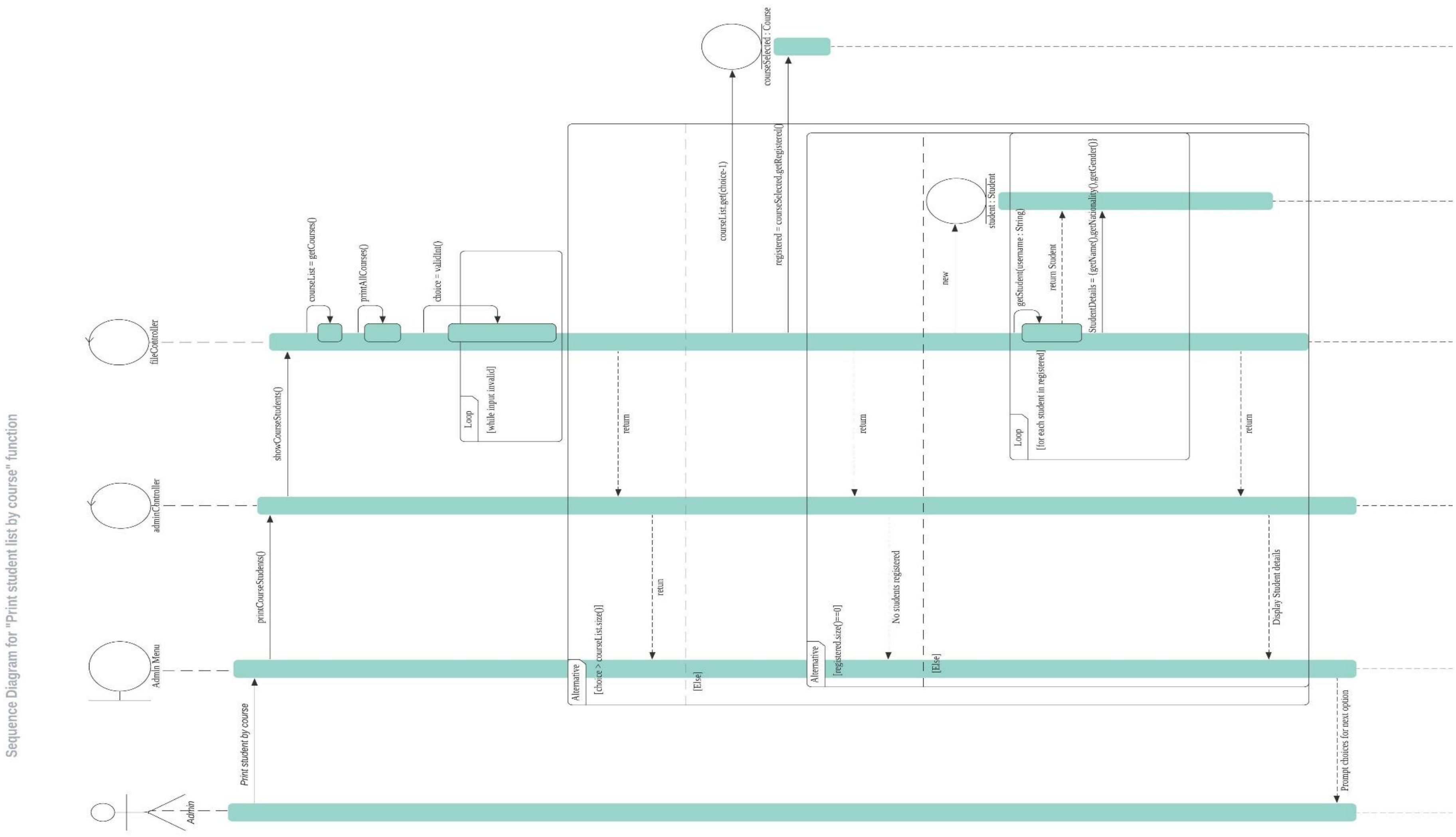
In our project, some of the assumptions and limitations are listed as follows.

- Multi-users concurrent login is not considered.
- Pre-requisite conditions when registering courses are not omitted.
- The major and school of each student is not accounted for. Therefore, all students can take all possible courses available on STAR.
- There are only two level of authorities in our STAR system, namely Admin and Student. There is discrepancy from the real life where more levels of authority are needed to handle the STAR properly.

UML CLASS DIAGRAM

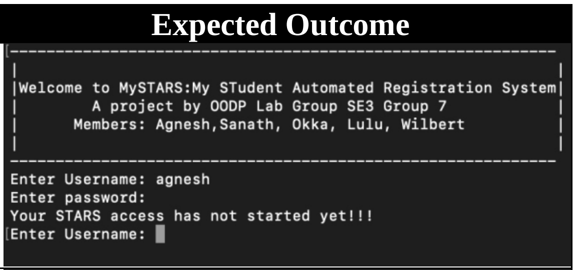
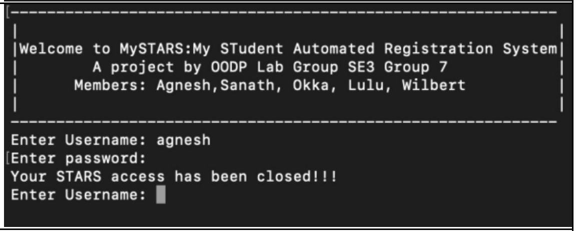
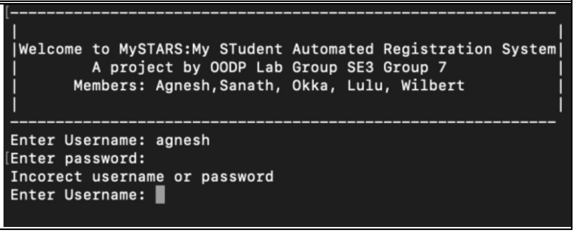


UML SEQUENCE DIAGRAM

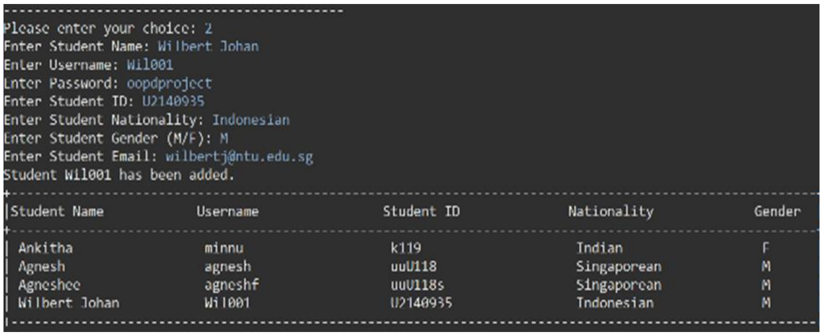
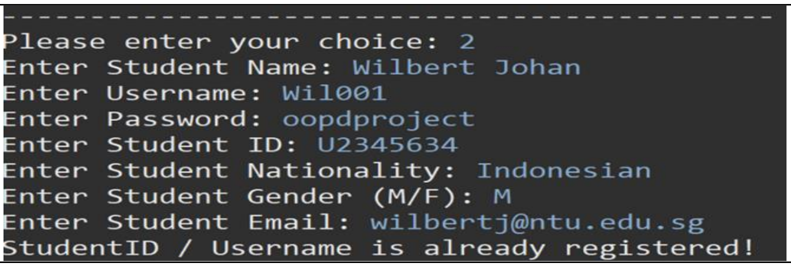


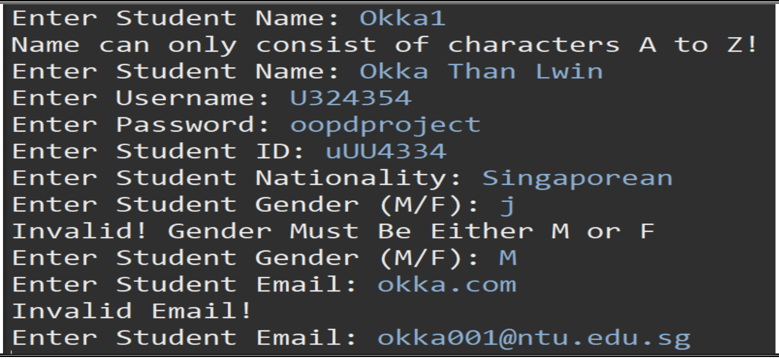
TEST CASES

Student Login

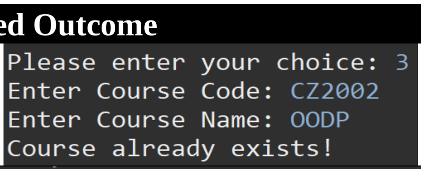
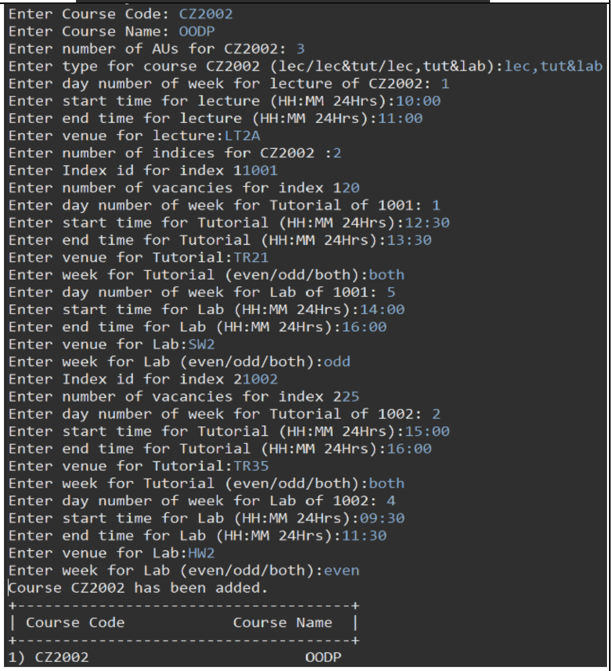
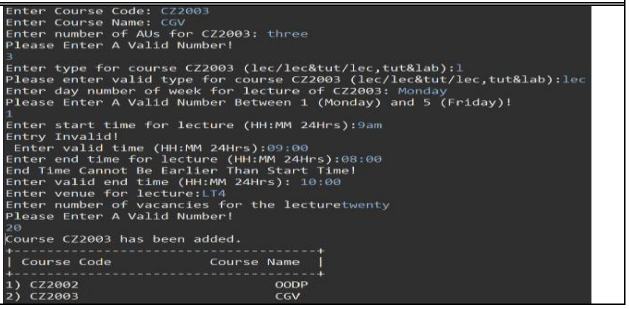
	Test Case	Expected Outcome
a	Login before allowed period (dates)	 -- | | |Welcome to MySTARS:My STudent Automated Registration System| | A project by OODP Lab Group SE3 Group 7 | | Members: Agnesh,Sanath, Okka, Lulu, Wilbert | | | -- Enter Username: agnesh Enter password: Your STARS access has not started yet!!! Enter Username:
b	Login after allowed period (dates)	 -- | | |Welcome to MySTARS:My STudent Automated Registration System| | A project by OODP Lab Group SE3 Group 7 | | Members: Agnesh,Sanath, Okka, Lulu, Wilbert | | | -- Enter Username: agnesh Enter password: Your STARS access has been closed!!! Enter Username:
c	**Wrong password**	 -- | | |Welcome to MySTARS:My STudent Automated Registration System| | A project by OODP Lab Group SE3 Group 7 | | Members: Agnesh,Sanath, Okka, Lulu, Wilbert | | | -- Enter Username: agnesh Enter password: Incorect username or password Enter Username:

Add a student

	Test Case	Expected Outcome
a	Add a new student	 -- Please enter your choice: 2 Enter Student Name: Wilbert Johan Enter Username: Wil001 Enter Password: oopdproject Enter Student ID: U2140935 Enter Student Nationality: Indonesian Enter Student Gender (M/F): M Enter Student Email: wilbertj@ntu.edu.sg Student Wil001 has been added. +-- |Student Name Username Student ID Nationality Gender +-- | Ankitha minnu k119 Indian F | Agnesh agnesh uuU118 Singaporean M | Agneshee agneshf uuU118s Singaporean M | Wilbert Johan Wil001 U2140935 Indonesian M |--
b	Add an existing student	 --- Please enter your choice: 2 Enter Student Name: Wilbert Johan Enter Username: Wil001 Enter Password: oopdproject Enter Student ID: U2345634 Enter Student Nationality: Indonesian Enter Student Gender (M/F): M Enter Student Email: wilbertj@ntu.edu.sg StudentID / Username is already registered!

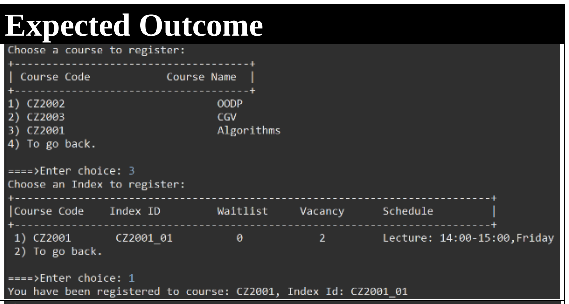
c	**Invalid data entries**		 Enter Student Name: Okka1 Name can only consist of characters A to Z! Enter Student Name: Okka Than Lwin Enter Username: U324354 Enter Password: oopdproject Enter Student ID: uUU4334 Enter Student Nationality: Singaporean Enter Student Gender (M/F): j Invalid! Gender Must Be Either M or F Enter Student Gender (M/F): M Enter Student Email: okka.com Invalid Email! Enter Student Email: okka001@ntu.edu.sg

Add a course

Test Case	Expected Outcome
Add an existing course	 Please enter your choice: 3 Enter Course Code: CZ2002 Enter Course Name: OODP Course already exists!
Add a new course **(with combination of (ii) from above)**	 Enter Course Code: CZ2002 Enter Course Name: OODP Enter number of AUs for CZ2002: 3 Enter type for course CZ2002 (lec/lec&tut/lec,tut&lab):lec,tut&lab Enter day number of week for lecture of CZ2002: 1 Enter start time for lecture (HH:MM 24Hrs):10:00 Enter end time for lecture (HH:MM 24Hrs):11:00 Enter venue for lecture:LT2A Enter number of indices for CZ2002 :2 Enter Index id for index 11001 Enter number of vacancies for index 120 Enter day number of week for Tutorial of 1001: 1 Enter start time for Tutorial (HH:MM 24Hrs):12:30 Enter end time for Tutorial (HH:MM 24Hrs):13:30 Enter venue for Tutorial:TR21 Enter week for Tutorial (even/odd/both):both Enter day number of week for Lab of 1001: 5 Enter start time for Lab (HH:MM 24Hrs):14:00 Enter end time for Lab (HH:MM 24Hrs):16:00 Enter venue for Lab:SW2 Enter week for Lab (even/odd/both):odd Enter Index id for index 21002 Enter number of vacancies for index 225 Enter day number of week for Tutorial of 1002: 2 Enter start time for Tutorial (HH:MM 24Hrs):15:00 Enter end time for Tutorial (HH:MM 24Hrs):16:00 Enter venue for Tutorial:TR35 Enter week for Tutorial (even/odd/both):both Enter day number of week for Lab of 1002: 4 Enter start time for Lab (HH:MM 24Hrs):09:30 Enter end time for Lab (HH:MM 24Hrs):11:30 Enter venue for Lab:HW2 Enter week for Lab (even/odd/both):even Course CZ2002 has been added. +-----------------------------------+ | Course Code Course Name | +-----------------------------------+ 1) CZ2002 OODP
Invalid data entries	 Enter Course Code: CZ2003 Enter Course Name: CGV Enter number of AUs for CZ2003: three Please Enter A Valid Number! 3 Enter type for course CZ2003 (lec/lec&tut/lec,tut&lab):l Please enter valid type for course CZ2003 (lec/lec&tut/lec,tut&lab):lec Enter day number of week for lecture of CZ2003: Monday Please Enter A Valid Number Between 1 (Monday) and 5 (Friday)! 1 Enter start time for lecture (HH:MM 24Hrs):9am Entry Invalid! Enter valid time (HH:MM 24Hrs):09:00 Enter end time for lecture (HH:MM 24Hrs):08:00 End Time Cannot Be Earlier Than Start Time! Enter valid end time (HH:MM 24Hrs): 10:00 Enter venue for lecture:LT4 Enter number of vacancies for the lecturetwenty Please Enter A Valid Number! 20 Course CZ2003 has been added. +-----------------------------------+ | Course Code Course Name | +-----------------------------------+ 1) CZ2002 OODP 2) CZ2003 CGV

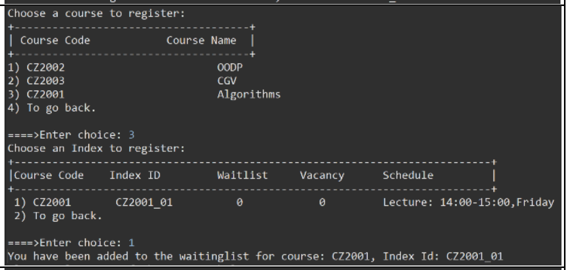
Register Student for a course

<table>
<tr><th></th><th>Test Case</th><th>Expected Outcome</th></tr>
<tr><td><b>a</b></td><td>Add a student to a course index with available vacancies.</td><td>



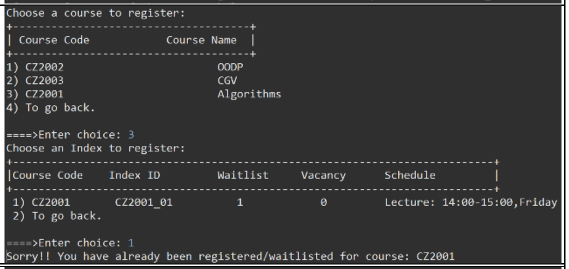
```
Choose a course to register:
+-----------------------------------+
| Course Code            Course Name  |
+-----------------------------------+
1) CZ2002                         OODP
2) CZ2003                         CGV
3) CZ2001                         Algorithms
4) To go back.

====>Enter choice: 3
Choose an Index to register:
+--------------------------------------------------------------------------+
|Course Code    Index ID         Waitlist     Vacancy      Schedule         |
+--------------------------------------------------------------------------+
 1) CZ2001       CZ2001_01          0           2         Lecture: 14:00-15:00,Friday
 2) To go back.

====>Enter choice: 1
You have been registered to course: CZ2001, Index Id: CZ2001_01
```

</td></tr>
<tr><td><b>b</b></td><td>Add a student to a course index with 0 vacancies in Tut / Lab.</td><td>



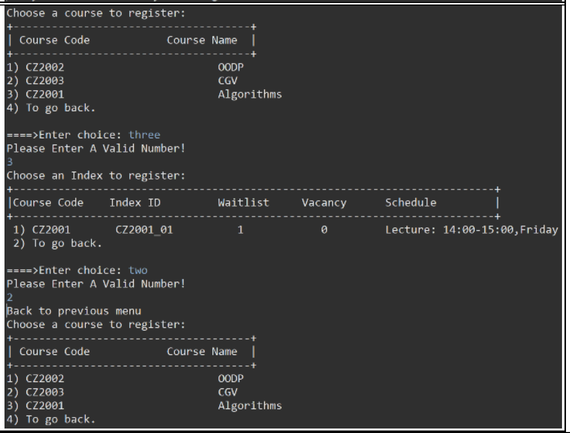
```
Choose a course to register:
+-----------------------------------+
| Course Code            Course Name  |
+-----------------------------------+
1) CZ2002                         OODP
2) CZ2003                         CGV
3) CZ2001                         Algorithms
4) To go back.

====>Enter choice: 3
Choose an Index to register:
+--------------------------------------------------------------------------+
|Course Code    Index ID         Waitlist     Vacancy      Schedule         |
+--------------------------------------------------------------------------+
 1) CZ2001       CZ2001_01          0           0         Lecture: 14:00-15:00,Friday
 2) To go back.

====>Enter choice: 1
You have been added to the waitinglist for course: CZ2001, Index Id: CZ2001_01
```

</td></tr>
<tr><td><b>c</b></td><td>Register the same course again</td><td>



```
Choose a course to register:
+-----------------------------------+
| Course Code            Course Name  |
+-----------------------------------+
1) CZ2002                         OODP
2) CZ2003                         CGV
3) CZ2001                         Algorithms
4) To go back.

====>Enter choice: 3
Choose an Index to register:
+--------------------------------------------------------------------------+
|Course Code    Index ID         Waitlist     Vacancy      Schedule         |
+--------------------------------------------------------------------------+
 1) CZ2001       CZ2001_01          1           0         Lecture: 14:00-15:00,Friday
 2) To go back.

====>Enter choice: 1
Sorry!! You have already been registered/waitlisted for course: CZ2001
```

</td></tr>
<tr><td></td><td>Invalid data entries
(e.g. wrong student ID / course code, etc.)</td><td>

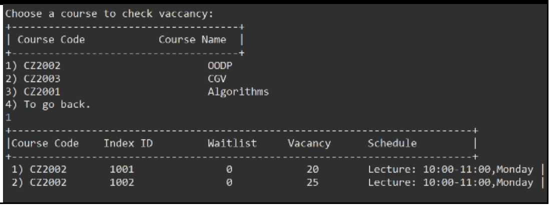


```
Choose a course to register:
+-----------------------------------+
| Course Code            Course Name  |
+-----------------------------------+
1) CZ2002                         OODP
2) CZ2003                         CGV
3) CZ2001                         Algorithms
4) To go back.

====>Enter choice: three
Please Enter A Valid Number!
3
Choose an Index to register:
+--------------------------------------------------------------------------+
|Course Code    Index ID         Waitlist     Vacancy      Schedule         |
+--------------------------------------------------------------------------+
 1) CZ2001       CZ2001_01          1           0         Lecture: 14:00-15:00,Friday
 2) To go back.

====>Enter choice: two
Please Enter A Valid Number!
2
Back to previous menu
Choose a course to register:
+-----------------------------------+
| Course Code            Course Name  |
+-----------------------------------+
1) CZ2002                         OODP
2) CZ2003                         CGV
3) CZ2001                         Algorithms
4) To go back.
```

</td></tr>
</table>

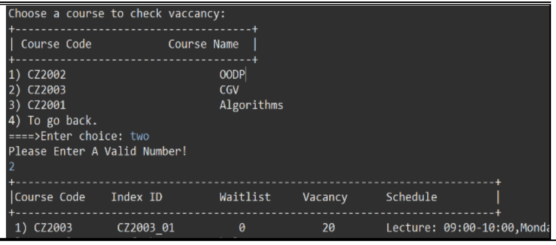
Check available slot in a class(vacancy in a class)

<table>
<tr><th></th><th>Test Case</th><th>Expected Outcome</th></tr>
<tr><td><b>a</b></td><td>Check for vacancy in course index</td><td>



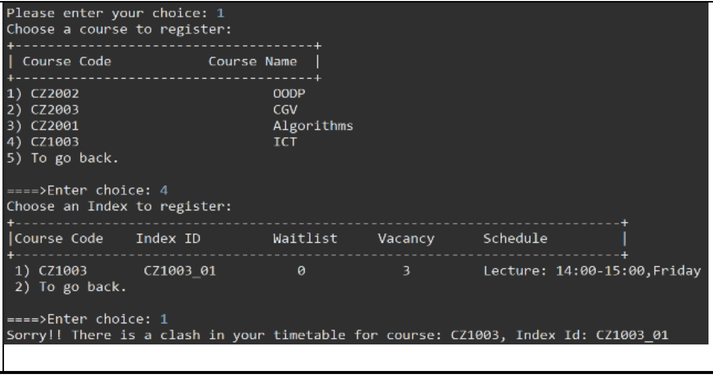
```
Choose a course to check vaccancy:
+-----------------------------------+
| Course Code            Course Name  |
+-----------------------------------+
1) CZ2002                         OODP
2) CZ2003                         CGV
3) CZ2001                         Algorithms
4) To go back.
1
+--------------------------------------------------------------------------+
|Course Code    Index ID         Waitlist     Vacancy      Schedule         |
+--------------------------------------------------------------------------+
 1) CZ2002       1001               0           20        Lecture: 10:00-11:00,Monday |
 2) CZ2002       1002               0           25        Lecture: 10:00-11:00,Monday |
```

</td></tr>
</table>

b	**Invalid data entries (e.g. course code, class code etc.)**	

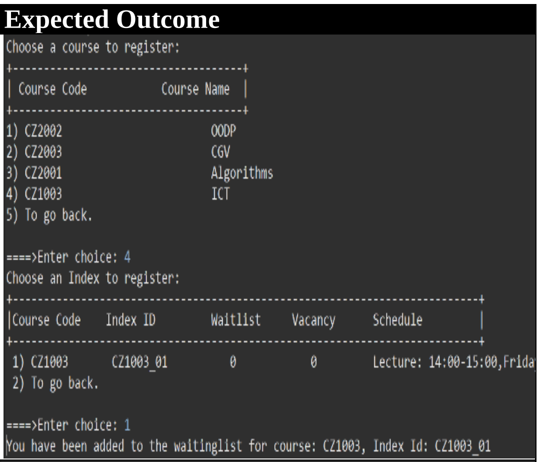
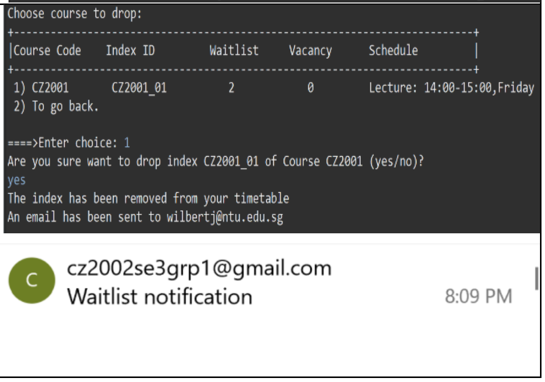
```
Choose a course to check vaccancy:
+-----------------------------------+
| Course Code           Course Name  |
+-----------------------------------+
1) CZ2002                       OODP
2) CZ2003                       CGV
3) CZ2001                       Algorithms
4) To go back.
====>Enter choice: two
Please Enter A Valid Number!
2
+------------------------------------------------------------------------+
|Course Code    Index ID        Waitlist     Vacancy      Schedule        |
+------------------------------------------------------------------------+
 1) CZ2003       CZ2003_01         0            20        Lecture: 09:00-10:00,Monda
```

Day/Time clash with another course

	Test Case	Expected Outcome
a	**Add a student to a course index with available vacancies.**	

```
Please enter your choice: 1
Choose a course to register:
+-----------------------------------+
| Course Code           Course Name  |
+-----------------------------------+
1) CZ2002                       OODP
2) CZ2003                       CGV
3) CZ2001                       Algorithms
4) CZ1003                       ICT
5) To go back.

====>Enter choice: 4
Choose an Index to register:
+------------------------------------------------------------------------+
|Course Code    Index ID        Waitlist     Vacancy      Schedule        |
+------------------------------------------------------------------------+
 1) CZ1003       CZ1003_01         0            3         Lecture: 14:00-15:00,Friday
 2) To go back.

====>Enter choice: 1
Sorry!! There is a clash in your timetable for course: CZ1003, Index Id: CZ1003_01
```

Waitlist Notification

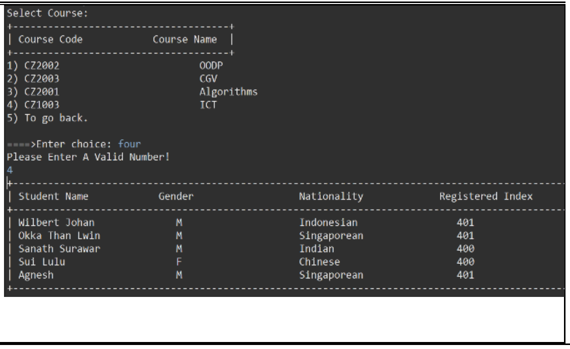
Test Case	Expected Outcome
Add studentA to a course index with 0 vacancies	
Drop studentB from the same course index	

```
Choose a course to register:
+-----------------------------------+
| Course Code           Course Name  |
+-----------------------------------+
1) CZ2002                       OODP
2) CZ2003                       CGV
3) CZ2001                       Algorithms
4) CZ1003                       ICT
5) To go back.

====>Enter choice: 4
Choose an Index to register:
+------------------------------------------------------------------------+
|Course Code    Index ID        Waitlist     Vacancy      Schedule        |
+------------------------------------------------------------------------+
 1) CZ1003       CZ1003_01         0            0         Lecture: 14:00-15:00,Frida
 2) To go back.

====>Enter choice: 1
You have been added to the waitinglist for course: CZ1003, Index Id: CZ1003_01
```

```
Choose course to drop:
+------------------------------------------------------------------------+
|Course Code    Index ID        Waitlist     Vacancy      Schedule        |
+------------------------------------------------------------------------+
 1) CZ2001       CZ2001_01         2            0         Lecture: 14:00-15:00,Friday
 2) To go back.

====>Enter choice: 1
Are you sure want to drop index CZ2001_01 of Course CZ2001 (yes/no)?
yes
The index has been removed from your timetable
An email has been sent to wilbertj@ntu.edu.sg
```

cz2002se3grp1@gmail.com
Waitlist notification
8:09 PM

<table>
<tr>
<td><b>Display studentA timetable</b></td>
<td>



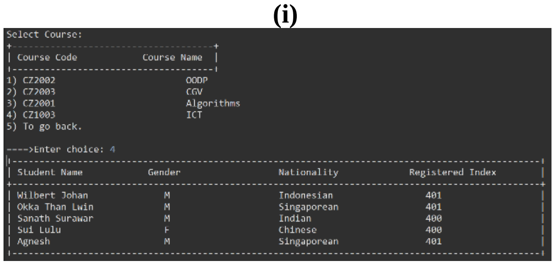
```
Select Course:
+-----------------------------------+
| Course Code           Course Name  |
+-----------------------------------+
1) CZ2002                    OODP
2) CZ2003                    CGV
3) CZ2001                    Algorithms
4) CZ1003                    ICT
5) To go back.

====>Enter choice: four
Please Enter A Valid Number!
4
+------------------------------------------------------------------------------------------------
| Student Name           Gender                 Nationality            Registered Index
+------------------------------------------------------------------------------------------------
| Wilbert Johan            M                    Indonesian               401
| Okka Than Lwin           M                    Singaporean              401
| Sanath Surawar           M                    Indian                   400
| Sui Lulu                 F                    Chinese                  400
| Agnesh                   M                    Singaporean              401
+------------------------------------------------------------------------------------------------
```

</td>
</tr>
</table>

Print student list by index number, course

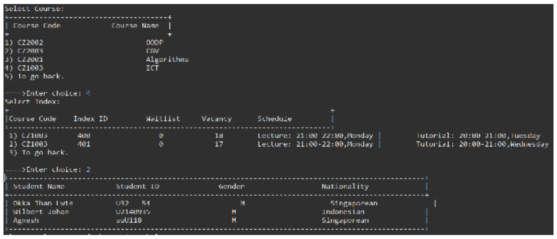
<table>
<tr><th></th><th>Test Case</th><th>Expected Outcome</th></tr>
<tr>
<td><b>a</b></td>
<td>Print list by
(i) Course
(ii) index</td>
<td>

(i)



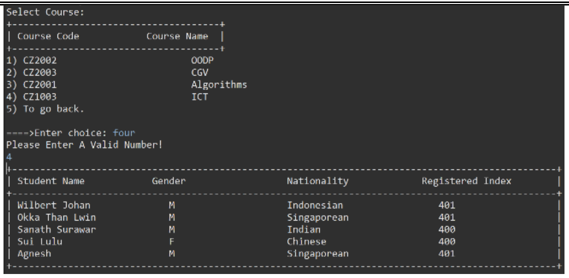
```
Select Course:
+-----------------------------------+
| Course Code           Course Name  |
+-----------------------------------+
1) CZ2002                    OODP
2) CZ2003                    CGV
3) CZ2001                    Algorithms
4) CZ1003                    ICT
5) To go back.

====>Enter choice: 4
+------------------------------------------------------------------------------------------------+
| Student Name           Gender                 Nationality            Registered Index           |
+------------------------------------------------------------------------------------------------+
| Wilbert Johan            M                    Indonesian               401                      |
| Okka Than Lwin           M                    Singaporean              401                      |
| Sanath Surawar           M                    Indian                   400                      |
| Sui Lulu                 F                    Chinese                  400                      |
| Agnesh                   M                    Singaporean              401                      |
+------------------------------------------------------------------------------------------------+
```

(ii)



```
Select Course:
+-----------------------------------+
| Course Code           Course Name  |
+                                   +
1) CZ2002                    OODP
2) CZ2003                    CGV
3) CZ2001                    Algorithms
4) CZ1003                    ICT
5) To go back.

---->Enter choice: 4
Select Index:
+                                                                     +
|Course Code    Index ID         Waitlist     Vacancy      Schedule   |
+---------------------------------------------------------------------
 1) CZ1003       400               0           18       Lecture: 21:00 22:00,Monday |       Tutorial: 20:00 21:00,Tuesday
 2) CZ1003       401               0           17       Lecture: 21:00-22:00,Monday |       Tutorial: 20:00-21:00,Wednesday
 3) To go back.

---->Enter choice: 2
+------------------------------------------------------------------------------------+
| Student Name           Student ID             Gender                  Nationality    |
+------------------------------------------------------------------------------------+
| Okka Than Lwin         U32   54                  M                      Singaporean     |
| Wilbert Johan          U2140935               M                      Indonesian     |
| Agnesh                 uuU118                 M                      Singaporean    |
+------------------------------------------------------------------------------------+
```

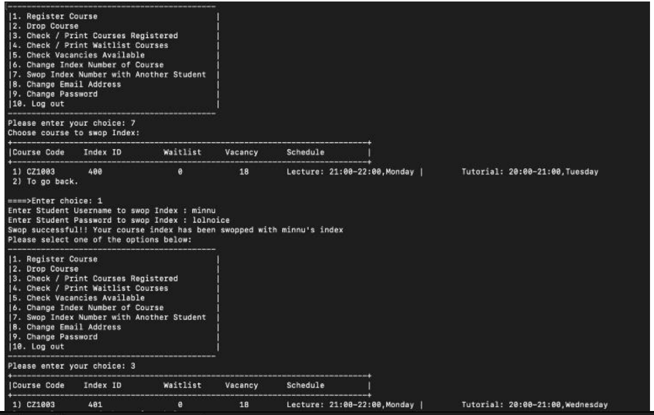
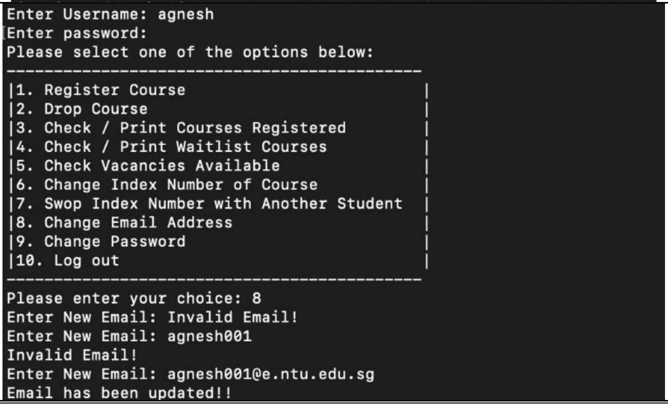
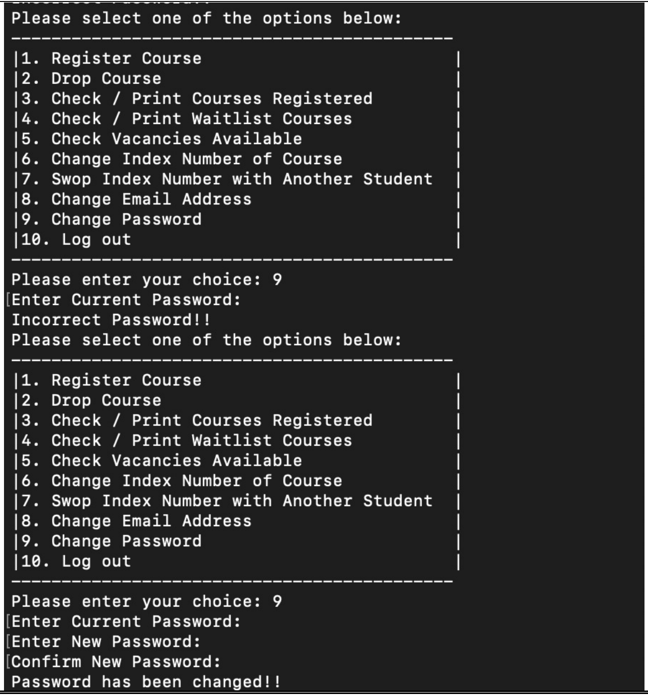
</td>
</tr>
<tr>
<td><b>b</b></td>
<td><b>Invalid data entries
(e.g. course code, index code etc.)</b></td>
<td>



```
Select Course:
+-----------------------------------+
| Course Code           Course Name  |
+-----------------------------------+
1) CZ2002                    OODP
2) CZ2003                    CGV
3) CZ2001                    Algorithms
4) CZ1003                    ICT
5) To go back.

---->Enter choice: four
Please Enter A Valid Number!
4
+------------------------------------------------------------------------------------------------+
| Student Name           Gender                 Nationality            Registered Index           |
+------------------------------------------------------------------------------------------------+
| Wilbert Johan            M                    Indonesian               401                      |
| Okka Than Lwin           M                    Singaporean              401                      |
| Sanath Surawar           M                    Indian                   400                      |
| Sui Lulu                 F                    Chinese                  400                      |
| Agnesh                   M                    Singaporean              401                      |
+------------------------------------------------------------------------------------------------+
```

</td>
</tr>
</table>

Additional Provisions

	Test Case	Expected Outcome
a	Swap indexes with another student	
b	Change registered email	
c	**Change password for student**	

```
-------------------------------------------
|1. Register Course                        |
|2. Drop Course                            |
|3. Check / Print Courses Registered       |
|4. Check / Print Waitlist Courses         |
|5. Check Vacancies Available              |
|6. Change Index Number of Course          |
|7. Swop Index Number with Another Student |
|8. Change Email Address                   |
|9. Change Password                        |
|10. Log out                               |
-------------------------------------------
Please enter your choice: 7
Choose course to swop Index:
+--------------------------------------------------------------------------+
|Course Code    Index ID        Waitlist     Vacancy      Schedule         |
+--------------------------------------------------------------------------+
 1) CZ1003      400             0            18           Lecture: 21:00-22:00,Monday |        Tutorial: 20:00-21:00,Tuesday
 2) To go back.

====>Enter choice: 1
Enter Student Username to swop Index : minnu
Enter Student Password to swop Index : lolnoice
Swop successful!! Your course index has been swopped with minnu's index
Please select one of the options below:
-------------------------------------------
|1. Register Course                        |
|2. Drop Course                            |
|3. Check / Print Courses Registered       |
|4. Check / Print Waitlist Courses         |
|5. Check Vacancies Available              |
|6. Change Index Number of Course          |
|7. Swop Index Number with Another Student |
|8. Change Email Address                   |
|9. Change Password                        |
|10. Log out                               |
-------------------------------------------
Please enter your choice: 3
+--------------------------------------------------------------------------+
|Course Code    Index ID        Waitlist     Vacancy      Schedule         |
+--------------------------------------------------------------------------+
 1) CZ1003      401             0            18           Lecture: 21:00-22:00,Monday |        Tutorial: 20:00-21:00,Wednesday
```

```
Enter Username: agnesh
Enter password:
Please select one of the options below:
-------------------------------------------
|1. Register Course                        |
|2. Drop Course                            |
|3. Check / Print Courses Registered       |
|4. Check / Print Waitlist Courses         |
|5. Check Vacancies Available              |
|6. Change Index Number of Course          |
|7. Swop Index Number with Another Student |
|8. Change Email Address                   |
|9. Change Password                        |
|10. Log out                               |
-------------------------------------------
Please enter your choice: 8
Enter New Email: Invalid Email!
Enter New Email: agnesh001
Invalid Email!
Enter New Email: agnesh001@e.ntu.edu.sg
Email has been updated!!
```

```
Please select one of the options below:
-------------------------------------------
|1. Register Course                        |
|2. Drop Course                            |
|3. Check / Print Courses Registered       |
|4. Check / Print Waitlist Courses         |
|5. Check Vacancies Available              |
|6. Change Index Number of Course          |
|7. Swop Index Number with Another Student |
|8. Change Email Address                   |
|9. Change Password                        |
|10. Log out                               |
-------------------------------------------
Please enter your choice: 9
Enter Current Password:
Incorrect Password!!
Please select one of the options below:
-------------------------------------------
|1. Register Course                        |
|2. Drop Course                            |
|3. Check / Print Courses Registered       |
|4. Check / Print Waitlist Courses         |
|5. Check Vacancies Available              |
|6. Change Index Number of Course          |
|7. Swop Index Number with Another Student |
|8. Change Email Address                   |
|9. Change Password                        |
|10. Log out                               |
-------------------------------------------
Please enter your choice: 9
Enter Current Password:
Enter New Password:
Confirm New Password:
Password has been changed!!
```

Link to Demonstration Video: https://youtu.be/97qAMecTb4k