



School of Computer Science and Engineering
CE2002 Project Report
My Student Automated Registration System
(MySTARS)

Prepared by:

SE2 Assignment Group 5

Deng Ziyang U1923770A

Elayne Tan Hui Shan U1921730C

Joe Marc Marzo Valdez U1822276G

Phang Yan Feng Benito U1922513L

Renganathan Ramasamy U1922494L

Table of Contents

1. Introduction.....	3
2. Design Considerations	3
2.1 Approach Taken	3
2.2 Principles Used.....	3
2.2.1 Single Responsibility Principle	3
2.2.2 Open-Closed Principle.....	3
2.2.3 Dependency Inversion Principle.....	4
2.3 Assumptions	4
3. The UML Diagram	5
3.1 Boundary	5
3.2 Control.....	5
3.3 Entity Classes	6
4. Sequence Diagram	8
5. Test Cases	9
6. Declaration of Original Work for CE/CZ2002 Assignment	15

1. Introduction

My Student Automated Registration System (MySTARS) is a university application developed for each School's academic staff and undergraduate students. It allows creation of courses, adding student records, and registration of courses and students.

The purpose of this report is to showcase the object-oriented programming concepts and design considerations that were used to create the application. Assumptions were also included which serves as the scope of the application developed. The class diagram of the application is included together with the sequence diagram to show relationships and object interactions. Lastly, test cases were included to ensure that the application meets the requirements.

2. Design Considerations

2.1 Approach Taken

In order to make the application "easy to maintain and modify", the team applied Object-Oriented Programming (OOP) concepts in the design and the implementation. In the design, classes are separated into different packages namely control, boundary and entity, based on their respective responsibilities for better clarity. Inheritance and composition were also applied when applicable to reduce code repetition and improve reusability of our code.

2.2 Principles Used

2.2.1 Single Responsibility Principle

SRP states that a class should only have one responsibility to ensure cohesion. This is exemplified in the various classes used in the project. For example, the responsibility of the UI classes is to handle any IO between the user and the program for each respective class. On the other hand, the entity classes, such as Student and Admin class, are only responsible for holding the attributes of each class respectively. This ensures easy readability, easy maintainability and only one reason for any class to change, hence resulting in a non-rigid program.

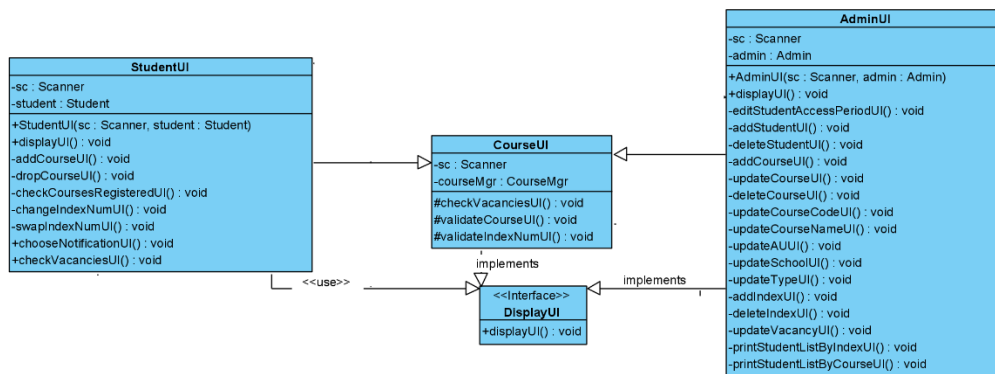
2.2.2 Open-Closed Principle

Open-Closed Principle states that entities should be open for extension but closed for modification which means these entities can allow its methods to be extended i.e. add new functionalities without modifying its source code, hence enabling loose coupling of objects. This is exemplified in the relationship between abstract class *User* and *Student* class. Student

implements User and can add new functionalities such as checkValidStudent() without modification to User class. Moreover, future extensions will also be possible such as in the case when the user is a staff that is also taking courses in the university.

2.2.3 Dependency Inversion Principle

This principle states that modules which provide complex logic (high-level modules) should be easily reused and should be unaltered by changes in low-level modules which provide utility features. High-level modules should not depend on low-level modules. We used this principle for our UI classes. Interface DisplayUI() is used to link and act as an intermediate for the higher-level UI and the lower-level UI, hence any changes made in the lower-level UI does not affect the higher-level UI. This also improves readability of our code.



2.3 Assumptions Made

1. Need not to consider multi-users concurrent log-in.
2. Need not to consider pre-requisite conditions when registering course.
3. Notification will be via external API (javax.mail)
4. The passwords for login will be stored in a flat file in hashed format and not clear text.
5. External source implies pre-existing records and can be loaded from the file(s).
6. Admin, Course and Student records are to be stored in files.
7. Updating student details will require the admin to delete and add new student.
8. The databases are pre-loaded with some information, generated from TestCases.txt using GenerateTestCases.java, which should be run first before running the main application.
9. The admin will not enter invalid data on purpose to cause errors.
10. The student's password field is not hidden on purpose during addition of new student by the admin to facilitate the admin. We assume that the admin is responsible enough.

11. We do not consider prerequisites required for registering of a new course.
12. We will accept “illogical” situations in real life scenarios, such as a 20AU course, for the purposes of testing.

3. The UML Diagram

3.1 Boundary

Boundary Classes are used to model the interaction between a system and its environment or actors. In the UML, the boundary classes receive input from a user—an admin or a student. The boundary classes are MainUI, LoginUI, and CourseUI. Inheritance is applied in StudentUI and Admin UI where they are subclasses of CourseUI as shown in Figure 1. This enables implementation of common methods thereby encouraging code reuse.

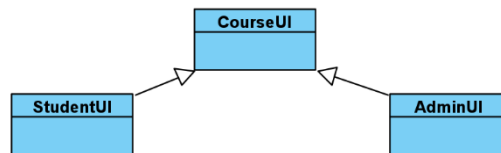


Figure 1. Inheritance of CourseUI.

At the start of the execution of the program, the user will input the selection shown by the MainUI. If the user is a student, the user will select the “Student” option and the LoginUI will be displayed. After a successful login, the StudentUI will be displayed where the user can choose its desired actions.

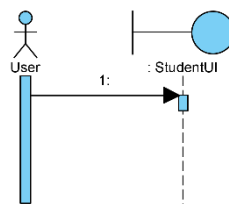


Figure 2. Interaction between user and StudentUI.

3.2 Control

Control classes are used to model control behaviour specific to one or more use cases. In other words, they run the use-case realizations.

In the UML, the realisation of the selections in StudentUI, LoginUI, and AdminUI are in the Control classes StudentMgr, LoginMgr, and AdminMgr respectively. For example, as shown in Figure 3, if a student wants to register for a course, the StudentUI will call on the StudentMgr to realise it by using addCourse method.

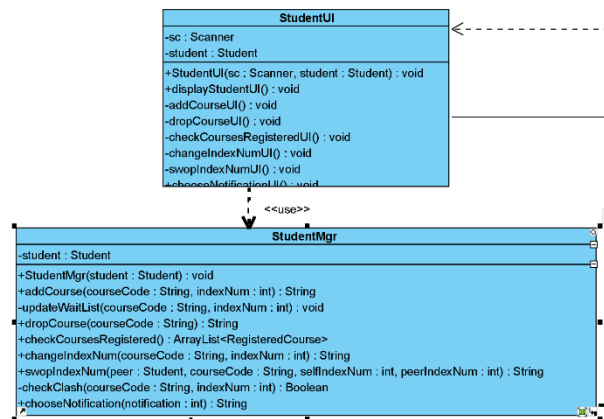


Figure 3: StudentMgr implements StudentUI.

Inheritance is also applied. The StudentMgr and the AdminMgr are subclasses of CourseMgr as shown in Figure 4. Again, this enables implementation of common methods thereby encouraging code reuse.

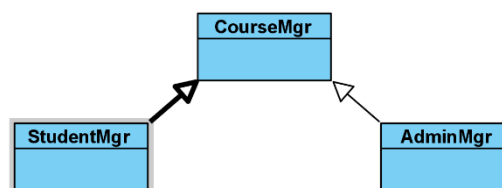


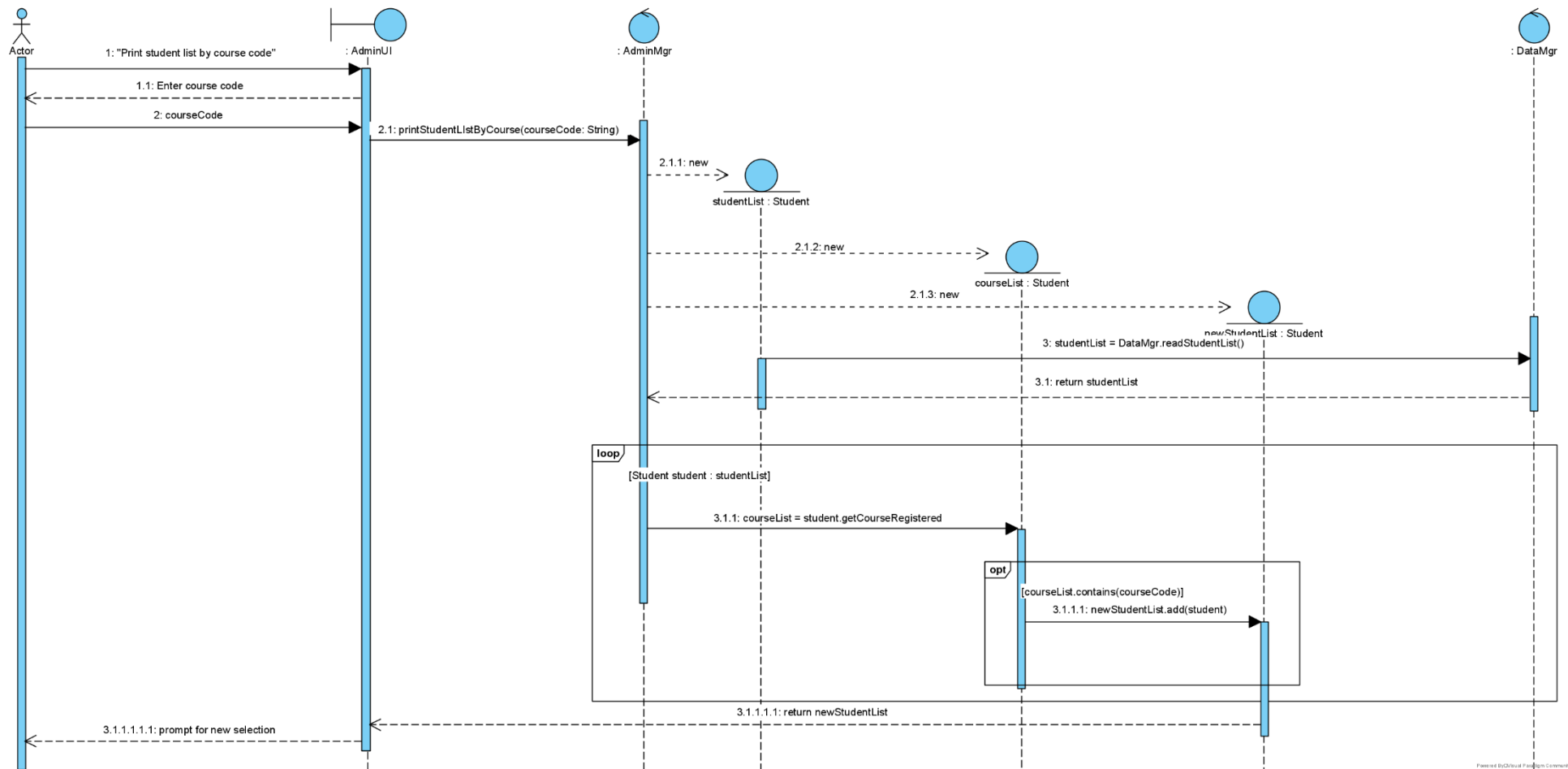
Figure 4: Inheritance of CourseMgr.

3.3 Entity Classes

Entity classes are where we store data or information in its attributes which will be used by the application or the system. Several of the Entity classes in the UML are Student and Admin which are subclasses of the User interface.

An illustration of the use of entity classes is when a student adds a course. The array list coursesRegistered will store all these courses which can then be used by the StudentMgr for checkCoursesRegistered.

4. Sequence Diagram



5. Test Cases

1. Student Login

	Test Case	Expected Outcome
a	Login before allowed period (dates)	<pre> MainUI [Java Application] C:\Program Files\Java\jdk-14.0.2\bin\javaw.exe (Nov 25, 2020, 3:37:12 PM) Username: TEST001 Password: Invalid Access Time! Username: </pre>
b	Login after allowed period (dates)	<pre> MainUI [Java Application] C:\Program Files\Java\jdk-14.0.2\bin\javaw.exe (Nov 25, 2020, 10:41:06 AM) Welcome to STARS Planner! Please select: [1] Login as Admin/Student [2] Quit Application 1 Please select: [1] Login as Admin [2] Login as Student [3] Back [4] Quit Application 2 Logging in as Student... Username: TEST001 Password: Successful login! Welcome to STARS firstname1 Please select: [1] Add course [2] Drop course [3] Check courses registered [4] Change course index number [5] Swap course index number [6] Choose notification method [7] Check vacancies available [8] Quit Application </pre>
c	Wrong password	<pre> Welcome to STARS Planner! Please select: [1] Login as Admin/Student [2] Quit Application 1 Please select: [1] Login as Admin [2] Login as Student [3] Back [4] Quit Application 2 Logging in as Student... Username: TEST001 Password: Wrong password! Username: </pre>

2. Add a student

	Test Case	Expected Outcome
a	Add a new student	<pre> Adding New Student... Student Username: JOEM0001 Student Password: password1 First Name: Joe Last Name: Marc Matric Number: U1822276G Gender: Male Nationality: Filipino Mobile Number: 98457553 Email: joem0001@e.ntu.edu.sg Access Period Start: Year: 2020 Month: 11 Day: 26 Hour: 09 Minute: 30 Access Period End: Year: 2020 Month: 11 Day: 27 Hour: 09 Minute: 30 New Student JOEM0001 created. </pre>

b	Add an existing student	<pre> MainUI [Java Application] C:\Program Files\Java\jdk-14.0.2\bin\javaw.exe (Nov 25, 2020, 12:09:56 PM) Please select: [1] Edit Student Access Period [2] Add New Student [3] Delete Student [4] Print Student List By Course [5] Print Student List By Index [6] Add Course [7] Check Index Vacancy [8] Update Course [9] Quit Application 2 Adding New Student... Student Username: JOEM0001 Student Password: password1 First Name: Joe Last Name: Marc Matric Number: U1822276G Matric Number already exists, please try again. Matric Number: </pre>
c	Invalid data entries	<pre> MainUI [Java Application] [5] Print Student List By Index [6] Add Course [7] Check Index Vacancy [8] Update Course [9] Quit Application 2 Adding New Student... Student Username: JOEM0001 Student Password: password1 First Name: Joe Last Name: Marc Matric Number: U1822276G Matric Number already exists, please try again. Matric Number: U1822376G Gender: Others Gender: Male Nationality: Filipino Mobile Number: JOE Invalid, please try again. Mobile Number: </pre>
d	Delete a student	<pre> MainUI (1) [Java Application] C:\Program Files\Java\jdk-14.0.2\bin\javaw.exe (Nov 25, 2020, 7:17:40 PM) [4] Print Student List By Course [5] Print Student List By Index [6] Add Course [7] Check Index Vacancy [8] Update Course [9] Logout [0] Quit Application 3 Deleting Student... Student Matric Number: U0000002A Student U0000002A deleted Please select: [1] Edit Student Access Period [2] Add New Student [3] Delete Student [4] Print Student List By Course [5] Print Student List By Index [6] Add Course [7] Check Index Vacancy [8] Update Course [9] Logout [0] Quit Application 1 Editing Student Access Period... Student Matric Number: U0000002A Student does not exist, please try again. </pre>

3. Add a course

	Test Case	Expected Outcome
a	Add a new course (with combination of (ii) from above)	<pre> MainUI [Java Application] 6 Adding New Course... Course code: MS1015 Course Name: Materials Science AU: 3 School: MSE Course Type: Core Exam Date:Year: 2021 Month: 05 Day: 01 Hour: 09 Minute: 30 New Course MS1015 added Please select: [1] Edit Student Access Period [2] Add New Student [3] Delete Student [4] Print Student List By Course [5] Print Student List By Index [6] Add Course [7] Check Index Vacancy [8] Update Course [9] Quit Application </pre>
b	Add an existing course	<pre> Adding New Course... Course code: MS1015 Course code already exists, please try again. Course code: </pre>
c	Invalid data entries	<pre> Adding New Course... Course code: MS1015 Course code already exists, please try again. Course code: MS4651 Course Name: Polymer Technology AU: III Invalid, please try again. AU: 3 School: MSE Course Type: IDK Exam Date:Year: IDK Invalid, please try again. Year: </pre>
d	Drop Course	<pre> MainUI (1) [Java Application] C:\Program Files\Java\jdk-14.0.2\bin\javaw.exe (Nov 25, 2020, 8:26:05 PM) [7] Check vacancies available [8] Logout [9] Quit Application 2 [2] Drop course selected. Enter course code: CE2001 Course dropped successfully. Welcome to STARS firstname1 Please select: [1] Add course [2] Drop course [3] Check courses registered [4] Change course index number [5] Swap course index number [6] Choose notification method [7] Check vacancies available [8] Logout [9] Quit Application 3 [3] Check courses registered selected. Student TEST001's Registered Courses Welcome to STARS firstname1 </pre>

4. Register student for a course

	Test Case	Expected Outcome
a	Add a student to a course index with available vacancies.	<pre>[1] Add course selected Enter course code: CE2001 Enter index number: 100000 Invalid index, please try again. Enter index number: 10010 Course successfully registered.</pre>
b	Add a student to a course index with 0 vacancies in Tut / Lab.	<pre>Enter course code: CE3000 Enter index number: 10009 Course successfully registered.</pre>
c	Register the same course again	<pre>MainUI (1) [Java Application] C:\Program Files\Java\jdk-14.0.2\bin\javaw.exe (Nov 25, 2020, 5:59:24 PM) [9] Quit Application 3 [3] Check courses registered selected. Student TEST001's Registered Courses CE2001 ALGORITHMS AU: 3 School: SCSE Type: Core Index Number: 10020 CE3000 NAME AU: 3 School: SCSE Type: UE Exam Date: 23/12/021 Index Number: 10009 Welcome to STARS firstname1 Please select: [1] Add course [2] Drop course [3] Check courses registered [4] Change course index number [5] Swap course index number [6] Choose notification method [7] Check vacancies available [8] Logout [9] Quit Application 1 [1] Add course selected Enter course code: CE2001 Enter index number: 10020 Failed to register for course - Course clashes with other registered courses.</pre>
d	Invalid data entries (eg wrong student ID / course code, etc)	<pre>[1] Add course selected Enter course code: CE2001 Enter index number: 100000 Invalid index, please try again.</pre>
e	Maximum AU reached	<pre>[1] Add course selected Enter course code: CE2000 Enter index number: 10009 Failed to register for course - Reached maximum number of AUs.</pre>

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

	Test Case	Expected Outcome
a	Check for vacancy in course index	<pre>MainUI [Java Application] C:\Program Files\Java\jdk-14.0.2\bin\javaw.exe (Nov 25, 2020, 3:56:14 PM) Successful login! Welcome to STARS firstname2 Please select: [1] Add course [2] Drop course [3] Check courses registered [4] Change course index number [5] Swap course index number [6] Choose notification method [7] Check vacancies available [8] Quit Application 7 [7] Check vacancies available selected. Enter course code: CE2005 Enter index number: 10029 Course code: CE2005 Index number: 10029 Number of vacancies: 9</pre>

b	Invalid data entries (eg course code, class code etc)	MainUI [Java Application] C:\Program Files\Java\jdk-14.0.2\bin\javaw.exe (Nov 25, 2020, 3:56:14 PM) Enter course code: CZ2 Course does not exist, please try again. Enter course code: CE2001 Enter index number: 12234 Enter index number: 10019 Course code: CE2001 Index number: 10019 Number of vacancies: 10
c	Swap index number with another student	MainUI [Java Application] /Library/Java/JavaVirtualMachines/jdk-14.0.2.jdk/Contents/Home/bin/java (25 Nov 2020, 9:07:11 pm) [7] Check vacancies available [8] Logout [9] Quit Application 5 [5] Swap course index number selected. Enter peer's username: TEST001 Enter course code: CE2001 Enter your index number: 10020 Enter peer's index number: 10019 Index number successfully swapped. Welcome to STARS Elayne

6. Day/Time clash with other course

	Test Case	Expected Outcome
a	Add a student to a course index with available vacancies.	MainUI [Java Application] /Library/Java/JavaVirtualMachines/jdk-14.0.2.jdk/Contents/Home/bin/java (25 Nov 2020, 9:07:11 pm) [3] Check courses registered [4] Change course index number [5] Swap course index number [6] Choose notification method [7] Check vacancies available [8] Logout [9] Quit Application 1 [1] Add course selected Enter course code: CS1105 Enter index number: 55082 Failed to register for course x Course clashes with other registered courses. Welcome to STARS firstname1

7. Waitlist notification

	Test Case	Expected Outcome
a(i)	Add studentA to a course index with 0 vacancies It should be noted that notification will only be sent via the notification method (email etc) went course registration is successful.	MainUI [Java Application] /Library/Java/JavaVirtualMachines/jdk-14.0.2.jdk/Contents/Home/bin/java (25 Nov 2020, 9:07:11 pm) [3] Check courses registered [4] Change course index number [5] Swap course index number [6] Choose notification method [7] Check vacancies available [8] Logout [9] Quit Application 1 [1] Add course selected Enter course code: CE3000 Enter index number: 10009 Course added to waitlist. Welcome to STARS Elayne Please select: [1] Add course [2] Drop course [3] Check courses registered [4] Change course index number [5] Swap course index number [6] Choose notification method [7] Check vacancies available [8] Logout [9] Quit Application

a(ii)	Drop studentB from the same course index	<pre> MainUI [Java Application] C:\Program Files\Java\jdk-14.0.2\bin\javaw.exe [2] Drop course [3] Check courses registered [4] Change course index number [5] Swap course index number [6] Choose notification method [7] Check vacancies available [8] Logout [9] Quit Application 2 [2] Drop course selected. Enter course code: CE3000 Course dropped successfully. Student gTAN074 has been informed. Sent email to ETAN074@E.NTU.EDU.SG Welcome to STARS firstname1 Please select: [1] Add course [2] Drop course [3] Check courses registered [4] Change course index number [5] Swap course index number [6] Choose notification method [7] Check vacancies available [8] Logout [9] Quit Application </pre>
a(iii)	Display studentA timetable	<pre> MainUI [Java Application] C:\Program Files\Java\jdk-14.0.2\bin\javaw.exe (25 Nov 2020, 9:07:11 pm) Status: Registered Index Number: 10019 Type: LEC/STUDIO, Day: FRI, Venue: LT10, Group: CE2, Remarks: Type: LEC/STUDIO, Day: FRI, Venue: LT11, Group: CE2, Remarks: Type: TUT, Day: FRI, Venue: TR+9, Group: SSR1, Remarks: Teaching Wk2-13 Type: LAB, Day: FRI, Venue: HMLAB1, Group: SSR1, Remarks: Teaching Wk2,4,6,8,10,12 CE3000 TESTTEST AU: 3 School: SCSE Type: GERPE Exam Date: 26/12/2020 Status: Waitlist Index Number: 10009 Type: LAB, Day: FRI, Venue: HMLAB2, Group: CE4, Remarks: CE2005 OPERATING SYSTEMS AU: 3 School: SCSE Type: Core Exam Date: 26/12/2020 Status: Registered Index Number: 10029 Type: LEC/STUDIO, Day: FRI, Venue: ONLINE, Group: CE2, Remarks: Type: LEC/STUDIO, Day: FRI, Venue: ONLINE, Group: CE2, Remarks: Type: TUT, Day: FRI, Venue: TR+44, Group: SE1, Remarks: Teaching Wk2-13 Type: LAB, Day: FRI, Venue: SWLAB3, Group: SE1, Remarks: Teaching Wk2,4,6,8,10,12 </pre>

8. Print student list by index number, course



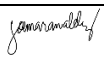


	Test Case	Expected Outcome
a	Print list by (i) Course (ii) index	<pre> Printing Student List By Course... Course code: CE2001 firstname1 lastname1 F Singaporean firstname2 lastname2 F Singaporean Elayne Tan F Singaporean Please select: 5 Printing Student List By Index... Course code: CE2001 Index: 10019 firstname1 lastname1 F Singaporean firstname2 lastname2 F Singaporean Please select: </pre>
b	Invalid data entries (eg course code, index code etc)	<pre> MainUI (1) [Java Application] C:\Program Files\Java\jdk-14.0.2\bin\javaw.exe (Nov 25, 2020, 8:36:38 PM) 5 Printing Student List By Index... Course code: IDK Course does not exist, please try again. Course code: CE2001 Index: IDK Invalid, please try again. Index: 10 Index does not exist, please try again. Index: 10019 firstname1 lastname1 F Singaporean </pre>

6. 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
Deng Ziyang	CE2002	SE2	 25/11/20
Elayne Tan Hui Shan	CE2002	SE2	 25/11/20
Joe Marc Marzo Valdez	CE2002	SE2	 25/11/20
Phang Yan Feng Benito	CE2002	SE2	 25/11/20
Renganathan Ramasamy	CE2002	SE2	 25/11/20

Important notes:

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

Demonstration video link: <https://youtu.be/rZzN6h-ra5o>