

**NANYANG
TECHNOLOGICAL
UNIVERSITY**

CE/CZ2002: Object-Oriented Design & Programming

Building an OO Application






Lab Group: SS12

Assignment Group: Group 2

Team Members:

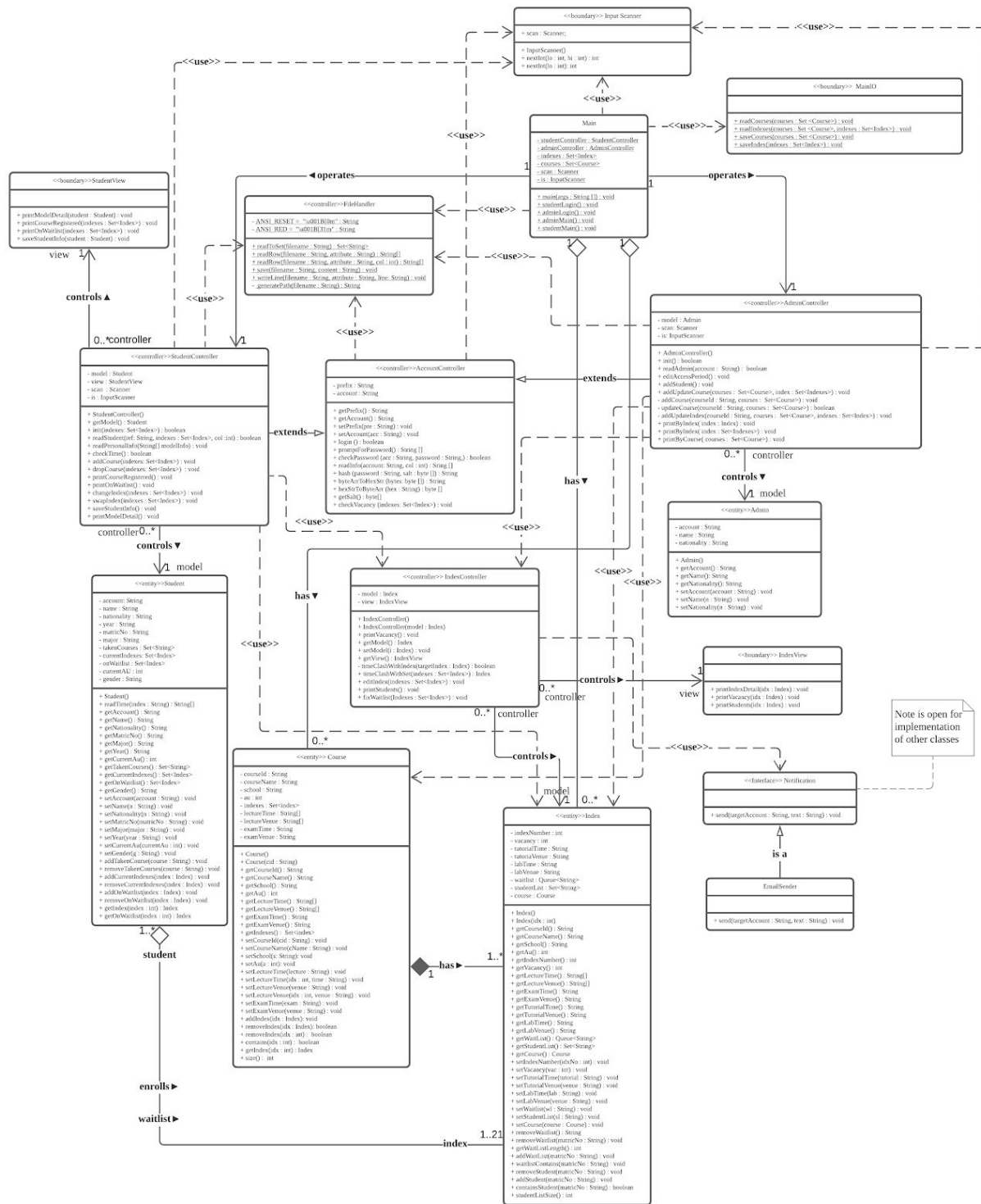
Bian Hengwei (U1923732B), He Yinan (U1922693C), Yin Jiarui (U1921776G),
Lim Qian Hui (U1922364J), Tan Min, Tricia (U1922619J)

Declaration of Original Work for CE/CZ2002 Assignment

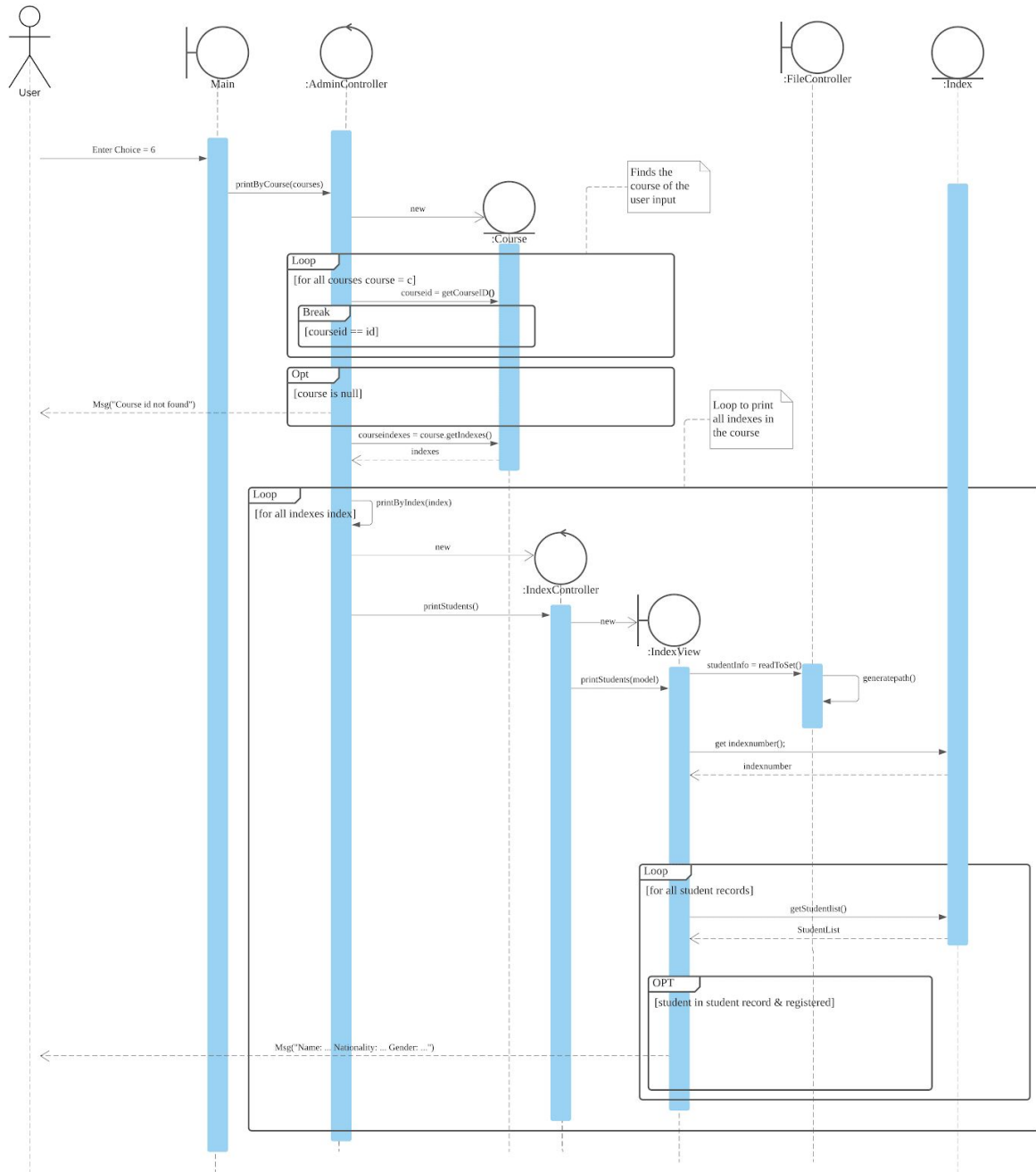
Name	Course Lab (CE2002 or CZ2002)	Lab Group	Signature/Date
Bian Hengwei	CZ2002	ss12	 25/11/2020
He Yinan	CZ2002	ss12	 25/11/2020
Yin Jiarui	CZ2002	ss12	 25/11/2020
Lim Qian Hui	CZ2002	ss12	 25/11/2020
Tan, Min Tricia	CZ2002	ss12	 25/11/2020

PROGRAM DEMO LINK: <https://youtu.be/NH9xD7jQ4CA>

UML Class Diagram



UML Sequence Diagram



Design Considerations

Overall Design Flow

In our design, we used the Model View Controller design pattern to avoid rigidity and immobility. In our application, we divide all code into 5 subpackages: app, boundary, controller, model, and view. App contains only the Main file of the project, which is used to run the whole MYSTARS system. Boundary contains all classes that deal with interactions with the user and file system. Controller contains control classes that operate the entities and controls the view to print messages. For example, the “Student Controller” class carries the logic needed for implementing all the functions, such as "Add Course", "Drop Course", "Change Index Number of Course", when a user login as a student. “Student Controller” will call “Student” which is a Model, and get all the data stored in a particular student object. It will call “Student View” to display the information of this student object. Models deal with all the data stored inside an entity and only contain methods that modify the data (setter and getter). For example, the “Student” class is a Model. It contains all the necessary get and set methods to take in and return data of students such as “getAccount”, “getCurrentAu”, “setAccount”, “setCurrentAu”. Last but not least, the view takes in the model and prints all information stored inside. For example, “StudentView” class to display students’ information.

In our MYSTARS system, we have 2 types of user, administrator and student. We designed a model class for each of them to store their data read from the csv files. Since we need to perform a number of operations on the two types of users, we created Controllers for the student class and the administrator class, which handles all requests from their menu (in the main). Considering that student information and admin information both need to read data from the csv files, and they both perform a series of similar operations like login, read information, use their View classes to produce output to the console and the csv files etc. We wrote their shared methods in one superclass called AccountController which handles the shared methods of the two controllers.

The course class and index class are used to store and handle course data and index data. Course is a collection of indices, the collection of indices have some shared attributes of indices like the course name, the lecture time and venue, etc which will be stored in course class. Indices store

the individual data like vacancies and index numbers. Since we need to perform a lot of operations on index, including checking index clashes, fixing waitlists if there is vacancy, we created an index controller, which handles all index operations and generates output through Index View class.

We created some additional boundary classes to deal with different kinds of inputs and outputs. They are divided by their functionalities and they all have only one or two functions to keep their responsibility simple (Single Responsibility Principle). We have a Notification interface that provides a “send()” method to send notification to students. It is implemented by EmailSender class which sends email to the student NTU email account, this allows easy extension as adding other methods of notification only requires creating a class that implements Notification. The FileHandler class tries to read from and write to databases. The MainIO class constructs the courses and indices collection for the main program. The InputScanner class tries to deal with all simple numeric inputs and eliminate any invalid inputs.

In conclusion, since every class does its own work, making our program structure clear.

Principles

1. Single Responsibility Principle (SRP)

We designed our classes such that each class only has one responsibility. We have split up the bigger classes into three classes. Model class only contains the setters and getters, it is responsible for the modification and retrieval of data of the object. View class contains all the reusable print functions, for example, StudentView class contains functions to print student information and functions to print the all courses a particular student is registered etc. The Controller class handles all the functions that are logic intensive, these methods are built using the model and view class, for example, StudentController class contains addCourse method and readStudent method.

2. Open Closed Principle (OCP)

In our design, the Notification Interface makes use of the Open Closed Principle. It has a send method that takes in a target account and a piece of text. The interface is then implemented by EmailSender class, which implements the send method and sends an

email to the target account. It is open for extension in the way that we can add more classes to implement the interface, for example, a WhatsAppSender class that sends WhatsApp messages. It is closed for modification such that the new classes would not affect any of the previous classes.

3. Dependency Injection Pattern (DIP)

The higher level classes in our design are independent of the lower level classes. For example, AccountController is a higher level class compared to StudentController and AdminController. AccountController is not dependent on these two classes and it can be reused quite simply should there be a new user type.

Assumptions made

- All courses have 2 lectures. Courses with one two hours lecture could be split into two continuous one hour lectures.
- All labs are either on odd or even week.
- Email sent only when student move from waitlist to registered
- AU of all courses are between 1 and 10.
- The maximum AU students can register or put onto waitlist is 21. Sum of number of AU of registered course and number of AU of waitlisted course cannot exceed 21.

Testing

Admin Functions

Function	Input	Expected Output
Login	<u>Successful login</u> Account: admin1 Password: pwd1	<pre> Account: admin1 Password: Logged in successfully. Current account: admin1 Please select one of the functions: 1. Edit Student Access Period 2. Add Student 3. Add/Update A Course 4. Check Available Slot For An Index Number 5. Print Student List By Index Number 6. Print Student List By Course 7. Exit </pre>
	<u>Unsuccessful login</u> Account: wrongacc OR Password: wrongpwd	<pre> Account: wrongacc Password: Login failed. Check account and password and try again. Logging in as admin... (prompt enter to exit) Account: </pre>

Edit Student Access Period	<p>Your choice: 1</p> <p>School: SCSE</p> <p>Year: 2</p> <p>Start date: 20201124</p> <p>End date: 20201130</p> <p>Start time: 0800</p> <p>End time: 2359</p>	<pre> Your choice: 1 ----- Edit student access period ----- You can set the access period for students from any school and any year School: SCSE Year: 2 Start date (YYYYMMDD): 20201124 End date (YYYYMMDD): 20201130 Start time (HHMM): 0800 End time (HHMM): 2359 Access time for student in SCSEY2 successfully changed </pre>
Add New Student	<p><u>Successful add student</u></p> <p>You choice: 2</p> <p>Account: student17</p> <p>Password: pwd17</p> <p>Student name: John</p> <p>Student nationality :sg</p> <p>Student matric number: U17</p> <p>Student major (school): SCSE</p> <p>Student year: Year 2</p> <p>Student gender: 4 (Invalid gender)</p> <p>Student gender: m</p> <p>(test case with invalid entries)</p>	<pre> Your choice: 2 ----- Add student ----- Account: student16 Password: Student account does not exist Please enter student information Student name: Tom Student nationality: sg Student matric number: U16 Student major (school): SCSE Student year: Year 2 Student gender (M/F): 4 Invalid gender Student gender (M/F): m Initializing student... New student account generated Saving student... Saving student... Student information successfully saved Student saved </pre>
	<p><u>Existing student account</u></p> <p>Your choice: 2</p> <p>Account: student2 (<i>already exists</i>)</p> <p>Password: pwd2</p>	<pre> Your choice: 2 Adding new student to student list... Account: student2 Password: Current account already exists Exiting... </pre>
Add Course	<p><u>Adding a new course</u></p> <p>Your choice: 3</p> <p>Option: 1</p> <p>Course ID: CZ2006</p> <p>Course name: SOFTWARE ENGINEERING</p> <p>School: SCSE</p> <p>AU: 3</p> <p>Option: 2</p> <p>(Since the course is new, no index is found.)</p>	<pre> Option: 1 ----- Add a new course ----- Course ID: CZ2006 Course name: SOFTWARE ENGINEERING School of the course: SCSE AU: 3 Course successfully added, please select option 2 and 3 to add more details. Select an option: 1. Add a new course 2. Modify indexes of existing course 3. Modify other information of existing course 4. Exit Option: 2 ----- Modify indexes of existing course ----- Course ID: CZ2006 Choose an option: 1. Display all indexes 2. Add new index 3. Modify existing index 4. Remove current index 5. Exit Option: 1 ----- Display all indexes ----- No index added </pre>

	<p><u>Unsuccessful adding a new course</u> Your choice: 3 Option: 1 Course ID: CZ2001 (This course already exists)</p>	<pre> Your choice: 3 ----- Add/Update a course ----- Select an option: 1. Add a new course 2. Modify indexes of existing course 3. Modify other information of existing course 4. Exit Option: 1 ----- Add a new course ----- Course ID: CZ2001 Error: this course already exists </pre>
	<p><u>Successful adding a new course</u> Your choice: 3 Option: 1 Course ID: CZ3003 School of the course: SCSE AU: r (Invalid input) AU: 3 (test case with invalid entries)</p>	<pre> Option: 1 ----- Add a new course ----- Course ID: CZ3003 Course name: SSAD School of the course: SCSE AU: r Invalid input Input: 3 Course successfully added, please select option 2 and nd 3 to add more details. </pre>
Update Course	<p><u>Adding indices to new course</u> <u>'CZ2006'</u> Your choice: 3 Option: 1 Index number: 20061 Updated information is reflected</p>	<pre> Choose an option: 1. Display all indexes 2. Add new index 3. Modify existing index 4. Remove current index 5. Exit Option: 2 ----- Add new index ----- Index number: 20061 Index added Choose an option: 1. Display all indexes 2. Add new index 3. Modify existing index 4. Remove current index 5. Exit Option: 1 ----- Display all indexes ----- 20061 </pre>

	<p><u>Updating index information to new index '20061' in 'CZ2006'</u> Your choice: 3 Option: 2</p> <p><u>Vacancy</u> Option: 3 New vacancy: 50</p> <p><u>Tutorial</u> Option: 4 Tutorial time: TUE1030-1130 Tutorial venue: THE WAVE</p> <p><u>Lab</u> Lab time: EVEWSED1030-1230 Lab venue: SWLAB2</p> <p><u>Check for updated information</u> Option : 1 The updated information is reflected</p>	<pre> Choose an option: 1. Print current index information 2. Change index number 3. Change vacancy 4. Set tutorial 5. Set lab 6. Exit Option: 3 ----- Change vacancy ----- New vacancy: 10 Finished waitlist operation Choose an option: 1. Print current index information 2. Change index number 3. Change vacancy 4. Set tutorial 5. Set lab 6. Exit Option: 4 ----- Set tutorial ----- New tutorial time (MON1030-1130): TUE1030-1130 New tutorial venue: THE WAVE Successfully saved Choose an option: 1. Print current index information 2. Change index number 3. Change vacancy 4. Set tutorial 5. Set lab 6. Exit Option: 5 ----- Set lab ----- New lab time (ODDMON1030-1230): EVEWED1030-1230 New lab venue: SWL2 Successfully saved ----- Print current index information ----- Printing index: 20061 Course ID: CZ2006 Course Name: SOFTWARE ENGINEERING School: SCSE AU: 3 Vacancy: 10 Lecture Time: Lecture Venue: Exam Time: Exam Venue: Tutorial Time: TUE1030-1130 Tutorial Venue: THE WAVE Lab Time: EVEWED1030-1230 Lab Venue: SWL2 </pre>
--	--	---

	<p><u>Updating lecture time of existing course 'CZ2002'</u></p> <p>Option 3: add/update a course</p> <p>Option 3: modify other information of existing course</p> <p>Option 5</p> <p>Option 0: change the first lecture timing</p> <p>New lecture time: THU0830-0930</p> <p>New lecture venue: LT1</p>	<pre> Please select one of the functions: 1. Edit student access period 2. Add student 3. Add/Update a course 4. Check available slot of an index number 5. Print student list by index number 6. Print student list by course 7. Exit Your choice: 3 ----- Add/Update a course ----- Select an option: 1. Add a new course 2. Modify indexes of existing course 3. Modify other information of existing course 4. Exit Option: 3 ----- Modify other information of existing course ----- Course ID: CZ2002 Select an option: 1. Change course id 2. Change course name 3. Change school 4. Change AU 5. Change lecture time 6. Change exam time 7. Continue Option: 5 ----- Change lecture time ----- Lecture Time: 0. MON0830-0930 1. TUE0830-0930 Select index to edit (or empty lecture to delete) Option: 1 New lecture time (MON1030-1130): THU0830-0930 New lecture venue: LT1 Successfully saved </pre>
Check Availability for Index	<p><u>Checking vacancy for index '20013'</u></p> <p>Your choice: 4</p> <p>Enter index to be checked: 20013</p>	<pre> Your choice: 4 please enter the index you want to check: 20013 Course id: CZ2001 Index: 20013 Vacancy: [0/3] Length of Wait list: 2 Would you like to check vacancy for another index? (y/n) </pre>
	<p><u>Checking vacancy for non-existent index</u></p> <p>Enter index to be checked: 20014 (does not exist)</p>	<pre> please enter the index you want to check: 20014 Index not found Would you like to check vacancy for another index? </pre>
Print student list by Index	<p><u>Printing student list for index '20013'</u></p> <p>Your choice: 5</p> <p>Enter index Id: 20013</p>	<pre> Your choice: 5 ----- Print student list bu index number ----- Enter the index Id: 20013 Printing students taking 20013 Name: Tom Nationality: sg Gender: m Name: name1 Nationality: sg Gender: m Name: Alice Nationality: sg Gender: f </pre>
	<p><u>Printing student list for non-existent index</u></p> <p>Enter index Id: 20014 (does not exist)</p>	<pre> Your choice: 5 Enter the index Id: 20014 Index not found </pre>

Print student list by course	<u>Printing student list for course 'CZ2002'</u> Your choice: 6 Enter course Id: CZ2002	<pre> Your choice: 6 ----- Print student list by course ----- Enter the course Id: CZ2002 Printing students taking 20023 Name: Violet Nationality: cn Gender: f Name: Diana Nationality: us Gender: m Printing students taking 20022 Name: name1 Nationality: sg Gender: m Printing students taking 20021 Name: Tom Nationality: sg Gender: m Name: Jasper Nationality: sg Gender: m Name: Jack Nationality: sg Gender: f </pre>
	<u>Printing student list for non-existent course</u> Enter course Id: CZ1015 (does not exist)	<pre> Your choice: 6 Enter the course Id: CZ1015 Course id not found </pre>

Students Functions

Function	Input	Expected Output
Student Login	<u>Successful login</u> Account: student12 Password: pwd12	<pre> Account: student12 Password: Logged in successfully. Current account: student12 Student Name: Kejie Matriculation Number: U12 Major: SCSE Year: Y3 Number of AUs taken: 0 </pre>
	<u>Unsuccessful login</u> Account: student1 Password: wrongpwd (incorrect password or account)	<pre> Account: student1 Password: Login failed. Check account and password and try again. Logging in as student... (press enter to exit) Account: █ </pre>
	<u>Unsuccessful login</u> Account: student2 Password: pwd2 (Login not between 10:00 and 12:00, not during assigned time)	<pre> Account: student2 Password: Logged in successfully. Current account: student2 Your assigned time is: 10:00 - 12:00 from 22/11/2020 to 01/12/2020 Try access the system at your assigned access time Courses successfully saved. Indexes successfully saved. </pre>
Add Course (the test cases are generated using different student accounts)	<u>Course registered</u> Your choice: 1 Index: 20051	<pre> Your choice: 1 ----- Add course ----- Please enter the index you want to add: 20051 You have successfully added index 20051 Registered Courses: Course: CZ2005 05, Index: 20051, School: SCSE </pre>
	<u>Course already registered</u> Your choice: 1 Index: 20051	<pre> Your choice: 1 ----- Add course ----- Please enter the index you want to add: 20051 This course is already registered </pre>

	<u>Course already registered, student add a different index for the same module</u> Your choice: 1 Index: 20052	<pre> Your choice: 1 ----- Add course ----- Please enter the index you want to add: 20052 This course is already registered </pre>
	<u>Taken course before</u> Your choice: 1 Index: 20013	<pre> Your choice: 1 ----- Add course ----- Please enter the index you want to add: 20013 You have taken this course before. </pre>
	<u>Time clash</u> Checks exam, tutorial, lecture and lab time clash	<pre> ----- Add course ----- Please enter the index you want to add: 20021 This Index has a time clash with CZ2005 which you are currently registered </pre>
	<u>No Vacancy, add to waitlist</u> Your choice: 1 Index: 20021	<pre> Your choice: 1 ----- Add course ----- Please enter the index you want to add: 20021 You have been successfully added onto waitlist of index 20021 </pre>
	<u>Invalid index</u> Your choice: 1 Index: 1005	<pre> Your choice: 1 ----- Add course ----- Please enter the index you want to add: 1005 Index entered is invalid. </pre>
Drop Course	<u>Course registered dropped</u> Your choice: 2 Type of course: 1 Index: 20051	<pre> Your choice: 2 ----- Drop course ----- Please choose the index to drop from below: Registered Courses: Course: CZ2005 0S, Index: 20051, School: SCSE Courses on WaitList: Course: CZ2002 00DP, Index: 20021, School: SCSE please choose type of course to drop: 1. Registered Course 2. Course on WaitList 1 Index: 20051 ----- Drop registered course ----- Finished waitlist operation 20051 is successfully dropped </pre>
	<u>Invalid index</u> Your choice: 2 Type of course: 1 Index: 10005	<pre> Your choice: 2 ----- Drop course ----- Please choose the index to drop from below: Not registered to any course yet Courses on WaitList: Course: CZ2002 00DP, Index: 20021, School: SCSE please choose type of course to drop: 1. Registered Course 2. Course on WaitList 1 Index: 10005 ----- Drop registered course ----- Index cannot be found among registered indexes. You are not registered to 10005 </pre>

	<u>Invalid index on waitlist</u> Your choice: 2 Type of course: 2 Index: 1003	<pre> Your choice: 2 ----- Drop course ----- Please choose the index to drop from below: Not registered to any course yet Courses on WaitList: Course: CZ2002 00DP, Index: 20021, School: SCSE please choose type of course to drop: 1. Registered Course 2. Course on WaitList 2 Index: 1003 ----- Drop waitlisted course ----- Index cannot be found among waitlisted indexes. You are not registered to 1003 </pre>
	<u>Course on waitlist dropped</u> Your choice: 2 Type of course: 2 Index: 20021	<pre> Your choice: 2 ----- Drop course ----- Please choose the index to drop from below: Not registered to any course yet Courses on WaitList: Course: CZ2002 00DP, Index: 20021, School: SCSE please choose type of course to drop: 1. Registered Course 2. Course on WaitList 2 Index: 20021 ----- Drop waitlisted course ----- 20021 is successfully dropped </pre>
Check/Print Course Registered and on Wait list	<u>Course registered and course registered on waitlist</u> Your choice: 3	<pre> Your choice: 3 ----- Check/Print courses registered and on waitlist ----- Registered Courses: Course: CZ2005 0S, Index: 20051, School: SCSE Courses on WaitList: Course: CZ2002 00DP, Index: 20021, School: SCSE </pre>
	<u>No course registered and no course registered on waitlist</u> Your choice: 3	<pre> Your choice: 3 ----- Check/Print courses registered and on waitlist ----- Not registered to any course yet Not registered on any waitlist </pre>
Change Index Number of Course	<u>Successful Change</u> Your choice: 4 Current Index: 20051 New Index: 20053 Confirm the change: y (confirm change)	<pre> ----- Change index number of course ----- Please enter current index: 20051 Please enter new index: 20053 ----- Current Index Information ----- Printing index: 20051 Course ID: CZ2005 Course Name: 0S School: SCSE AU: 3 Vacancy: 0 Lecture Time: WED2100-2200 THU2100-2200 Lecture Venue: LT2A Exam Time: 12030830-1030 Exam Venue: LT3 Tutorial Time: FRI2100-2200 Tutorial Venue: TR15 Lab Time: EVEFRI0130-0330 Lab Venue: SWL2 ----- New Index Information ----- Printing index: 20053 Course ID: CZ2005 Course Name: 0S School: SCSE AU: 3 Vacancy: 1 Lecture Time: WED2100-2200 THU2100-2200 Lecture Venue: LT2A Exam Time: 12030830-1030 Exam Venue: LT3 Tutorial Time: FRI0930-1130 Tutorial Venue: TR03 Lab Time: EVEFRI1430-1630 Lab Venue: HWLAB1 Please enter y to confirm the change, enter n to cancel. y You have successfully changed from index 20051 to index 20053 </pre>
	<u>Unsuccessful Change</u> unconfirm change	<pre> Please enter y to confirm the change, enter n to cancel. n Changing of indexes cancelled. </pre>

	<u>Unsuccessful Change</u> Your choice: 4 Current Index: 20053 New Index: 20052 (no vacancy)	<pre> Your choice: 4 ----- Change index number of course ----- Please enter current index: 20053 Please enter new index: 20052 The new Index do not have vacancy. </pre>
	<u>Unsuccessful Change</u> Your choice: 4 Current Index: 20023 (invalid current index)	<pre> Your choice: 4 ----- Change index number of course ----- Please enter current index: 20023 You have not registered for the current index you entered. </pre>
	<u>Unsuccessful Change</u> Your choice: 4 Current Index: 20053 New Index: 10005 (invalid new index)	<pre> Your choice: 4 ----- Change index number of course ----- Please enter current index: 20053 Please enter new index: 10005 The new index you entered is not valid. </pre>
	<u>Unsuccessful Change</u> Your choice: 4 Current Index: 20053 New Index: 20021 (Index has a time clash)	<pre> Your choice: 4 ----- Change index number of course ----- Please enter current index: 20023 Please enter new index: 20021 This Index has a time clash with CZ2005 which you are currently registered </pre>
Swap Index With Another Student	<u>Successful Swap</u> Your choice: 5 Current Index: 20053 Peer's Account: student4 Peer's Password: pwd4 Peer's Index: 20051	<pre> Your choice: 5 ----- Swap index number with another student ----- Your index number: 20051 Trying peer's login... Checking password... Logging in as student... (Press enter to exit) Account: student4 Password: Logged in successfully. Current account: student4 Peer's index number: 20052 Indexes successfully switched </pre>
	<u>Unsuccessful Swap</u> Your choice: 5 Current Index: 20053 Peer's Account: student15 Peer's Password: pwd15 Peer's Index: 20021 (Peer is not registered that index)	<pre> Your choice: 5 ----- Swap index number with another student ----- Your index number: 20053 Trying peer's login... Checking password... Logging in as student... (Press enter to exit) Account: student15 Password: Logged in successfully. Current account: student15 Peer's index number: 20021 Index cannot be found among registered indexes. Peer is not registered for 20021 </pre>

	<p><u>Unsuccessful Swap</u> Your choice: 5 Current Index: 20053 (from CZ2005) Peer's Account: student15 Peer's Password: pwd15 Peer's Index: 20011 (from CZ2001) (Two indices entered are not from the same course)</p>	<pre> 7. Press Your choice: 5 ----- Swap index number with another student ----- Your index number: 20053 Trying peer's login... Checking password... Logging in as student... (Press enter to exit) Account: student15 Password: Logged in successfully. Current account: student15 Peer's index number: 20011 The two indexes entered are not from the same course. </pre>
	<p><u>Unsuccessful Swap</u> Incorrect peer password and account, reach 3 attempts</p>	<pre> Logging in as student... (Press enter to exit) Account: student11 Password: Login failed. Check account and password and try again. 2 Logging in as student... (Press enter to exit) Account: student11 Password: Login failed. Check account and password and try again. 3 You have exceeded 3 tries. </pre>
	<p><u>Unsuccessful Swap</u> Your choice: 5 Current Index: 20053 Peer's Account: student2 Peer's Password: pwd2 (Not peer's assigned time)</p>	<pre> Your choice: 5 ----- Swap index number with another student ----- Your index number: 20053 Trying peer's login... Checking password... Logging in as student... (Press enter to exit) Account: student2 Password: Logged in successfully. Current account: student2 Your assigned time is: 10:00 - 12:00 from 22/11/2020 to 01/12/2020 Try access the system at your assigned access time Peer login unsuccessful. </pre>
Check Vacancies Available	<p><u>Successful Check</u> Your choice: 6 Index: 20021</p>	<pre> Your choice: 6 ----- Check vacancies available ----- please enter the index you want to check: 20021 Course id: CZ2002 Index: 20021 Vacancy: [0/3] Length of Wait list: 1 Would you like to check vacancy for another index? (y/n) </pre>
	<p><u>Successful Check</u> Your choice: 6 Index: 20021 Check Vacancy for another index: y Index: 20051</p>	<pre> Your choice: 6 ----- Check vacancies available ----- please enter the index you want to check: 20021 Course id: CZ2002 Index: 20021 Vacancy: [0/3] Length of Wait list: 1 Would you like to check vacancy for another index? (y/n) y please enter the index you want to check: 20051 Course id: CZ2005 Index: 20051 Vacancy: [1/3] Length of Wait list: 0 Would you like to check vacancy for another index? (y/n) </pre>