






### **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 the 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
Lionel Wong Zhi Neng	CZ2002	SS14	 24/11/2020
Glenn Tay	CZ2002	SS14	 24/11/2020
Lim Khian Heng, Alston	CZ2002	SS14	 24/11/2020
Amadeus Koh Ying Jie	CZ2002	SS14	 24/11/2020
Satini Sankeerthana	CZ2002	SS14	 24/11/2020

## Design Considerations

### Approach Taken :

To design MySTARS, we considered the approach from both the perspective of a Student and the Admin. We thought about what boundary, control and entity classes, as well as interfaces, superclasses, abstract classes and enumerations, to have. We decided to have 3 boundary classes that are responsible for displaying options to the user: StarsMgr, StudentMenu and AdminMenu. StarsMgr asks the user for his/her username and password, StudentMenu displays the available options that a student can choose from, and AdminMenu displays the available options for the Admin. Our control classes are StudentMgr and AdminMgr, both of which verifies the user logging in, and handles the logic of the different inputs by the student/admin respectively. Our entity classes are Course, Index, Lab, Lecture, Tutorial, Lesson, Student, Venue, and Timetable. Lesson is an abstract class which contains protected attributes like DayOfWeek, timeslot and Venue, which are inherited by the child classes: Lab, Lecture and Tutorial. The control class AdminMgr has a dependency on another class - Email which uses its sendEmail() method to send email notifications.

### Principles Used :

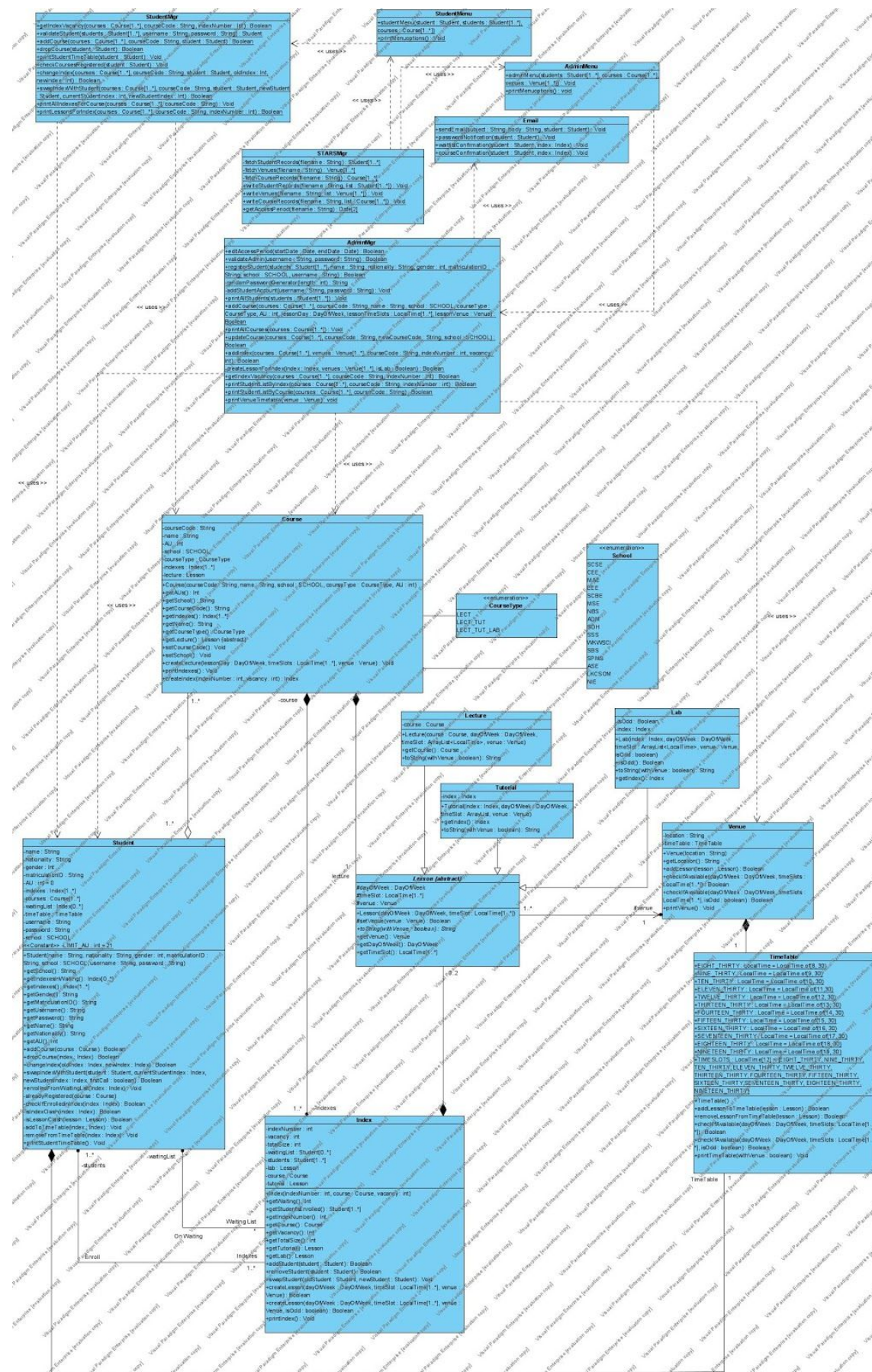
1. Single Responsibility Principle: Each class has a unique responsibility.
2. Open-closed principle: The 'Lesson' abstract class is Open for extension by lecture, lab and tutorial but closed for modification by the Index class
3. Dependency-injection principle: Higher level modules like AdminMgr and StudentMgr are independent of the low level modules like Index, lab, tutorial and lecture
4. Encapsulation and Information Hiding : Classes like STARSMgr, AdminMgr and StudentMgr contain private methods that can only be accessible by their respective classes. Hence, the logic of these functions are not made known to other classes thus achieving Encapsulation and Information Hiding.
5. Abstraction: we have Lesson as an abstract class with Tutorial, Lab and Lecture as its child classes because they have common attributes: DayOfWeek, timeslot and venue.
6. Liskov Substitution Principle: The class Lesson and its subclasses - Lecture, Tutorial and Lab classes are coded in such a way that the objects of the Lecture can be substituted by either of its subclasses.

Assumptions Made :

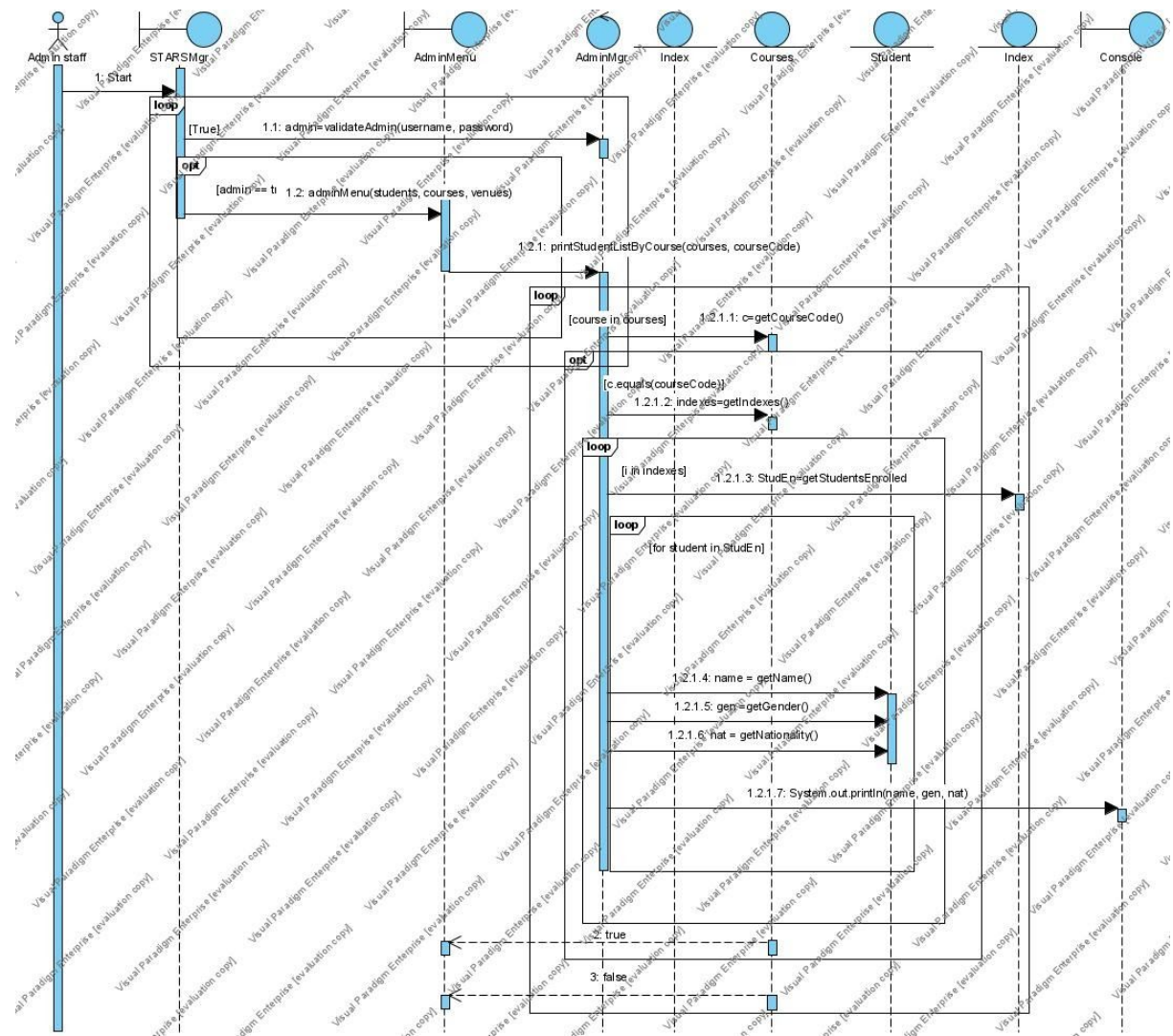
1. Admins do not have access to edit the class schedule and venue(applies to lectures, lab and tutorials).
2. Admins have only 1 shared account that has a fixed username and password
3. All students have the same access time for STARS
4. Students are allowed to drop a course after registering for it. Students are allowed to have no courses.
5. All indexes for the same course will have the same lecture timeslot
6. Each course can only have 1 lecture per week.
7. Lesson timings are divided into blocks of 1 hour from 8.30am to 8.30pm (12 blocks)

Youtube link: <https://www.youtube.com/watch?v=ISiRZ0cot40>

### Detailed UML Class Diagram



## Detailed UML Sequence Diagram





## Test cases:

### 1. Student Login

Invalid username/password:

```
=====LOGIN=====
***Do not exit by closing terminal***
Log in to Admin/Student Account. Press Enter to Quit.
Username: lwong020
Password:
Invalid username/password!
```

Logging in before/after access period:

```
=====LOGIN=====
***Do not exit by closing terminal***
Log in to Admin/Student Account. Press Enter to Quit.
Username: lwong020
Password:
Sorry, students not allowed to log in now.
Access period is from Tue, 24 Nov 2020, 12:00 am to Tue, 24 Nov 2020, 11:59 pm
```

Editing student access period:

```
=====EDIT STUDENT ACCESS PERIOD=====
Enter the start date in the format (dd/mm/yyyy hh:mm) - 24 hour clock: 23/11/2020 22:00
Enter the end date in the format (dd/mm/yyyy hh:mm) - 24 hour clock: 01/12/2020 23:59
Start date set to: Mon, 23 Nov 2020, 10:00 pm
End date set to: Tue, 01 Dec 2020, 11:59 pm
```

### 2. Add a Student

Adding a student:

```
=====ADD A STUDENT=====
Enter the student's name: lionel
Enter the student's nationality: sg
Enter the student's gender (1 for Male/0 for Female): 1
Enter the student's matriculation ID: u9182859f
(1) SCSE
(2) CEE
(3) MAE
(4) EEE
(5) SCBE
(6) MSE
(7) NBS
(8) ADM
(9) SoH
(10) SSS
(11) WKWSCl
(12) SBS
(13) SPMS
(14) ASE
(15) LKCSOM
(16) NIE
Select the student's school: 1
Enter the student's username: lwong020
DEBUG: setDebug: JavaMail version 1.6.2
Sending email to lwong020@e.ntu.edu.sg...
```

## Email sent to student after creation of student:

### Login Details for AY2020-2021 SEMESTER 1 STAR WARS



ntustarssystem@gmail.com  
Mon 11/23/2020 10:50 PM  
To: #LIONEL WONG ZHI NENG#

Dear lionel  
Matric ID: u9182859f

Please take note of the login details for the upcoming AY2020-2021 STAR WARS:

Username: lwong020  
Password: FRDXr1@o

Regards  
STARS System

## Printing student list after adding new student:

```
Dear lionel
Matric ID: u9182859f

Please take note of the login details for the upcoming AY2020-2021 STAR WARS:

Username: lwong020
Password: FRDXr1@o

Regards
STARS System
.
250 2.0.0 OK 1606143036 a21sm13537759pjh.3 - gmail
DEBUG SMTP: message successfully delivered to mail server
QUIT
221 2.0.0 closing connection a21sm13537759pjh.3 - gmail
Sent email successfully....
Password is FRDXr1@o

List of Students:
lionel (Male) sg u9182859f SCSE
```

## Adding an already added student:

```
=====ADD A STUDENT=====
Enter the student's name: lionel
Enter the student's nationality: sg
Enter the student's gender (1 for Male/0 for Female): 1
Enter the student's matriculation ID: u28383838f
(1) SCSE
(2) CEE
(3) MAE
(4) EEE
(5) SCBE
(6) MSE
(7) NBS
(8) ADM
(9) SoH
(10) SSS
(11) WKWSCI
(12) SBS
(13) SPMS
(14) ASE
(15) LKCSOM
(16) NIE
Select the student's school: 1
Enter the student's username: lwong020
Student's matriculation ID or username already exists! Please try again.
```

Entering string into gender:

```
Enter the student's gender (1 for Male/0 for Female): male
Please enter an integer: █
```

Entering integer out of range of possible values:

```
Enter the student's gender (1 for Male/0 for Female): 2
Enter the student's gender (1 for Male/0 for Female): 3
```

### 3. Add a Course

Adding a course and printing courses added:

```
=====ADD A COURSE=====
Enter the course code: CZ2001
Enter the course name: Algorithms
(1) SCSE
(2) CEE
(3) MAE
(4) EEE
(5) SCBE
(6) MSE
(7) NBS
(8) ADM
(9) SoH
(10) SSS
(11) WKWSCI
(12) SBS
(13) SPMS
(14) ASE
(15) LKCSOM
(16) NIE
Select the course's school: '1
Please enter an integer: 1
(1)Lecture only
(2)Lecture and Tutorial
(3)Lecture, Tutorial and Lab
Select the course type: 1
Enter the number of AUs for this course: 3
Select time and location for the lecture:
Enter the day of the lecture (1 for Monday to 5 for Friday): 1
Enter the duration (number of hours) of the lecture: 3
Select the starting time slot
(1) 08:30
(2) 09:30
(3) 10:30
(4) 11:30
(5) 12:30
(6) 13:30
(7) 14:30
(8) 15:30
(9) 16:30
(10) 17:30
(11) 18:30
```

```
(12) 19:30
Select the starting time slot (1 - 12): 1
Select a venue (1-9):
1) LT1
2) LT2
3) LT3
4) TR1
5) TR2
6) TR3
7) SWLab
8) HWLab
9) Project Lab
1

List of all Courses:
CZ2001: Algorithms SCSE Number of AUs: 3
```

Updating a course code and school:

```
=====UPDATE COURSE CODE/SCHOOL=====
Enter the course code: CZ2001
Enter the new course code (re-enter same course code to leave it unchanged): CZ2002
(1) SCSE
(2) CEE
(3) MAE
(4) EEE
(5) SCBE
(6) MSE
(7) NBS
(8) ADM
(9) SoH
(10) SSS
(11) WKWSCI
(12) SBS
(13) SPMS
(14) ASE
(15) LKCSOM
(16) NIE
Select the school (select same school to leave school unchanged): 15
Course code changed from CZ2001 to CZ2002
```



Adding a course with course code that already exists:

```
=====ADD A COURSE=====
Enter the course code: CZ2002
Enter the course name: OODP
(1) SCSE
(2) CEE
(3) MAE
(4) EEE
(5) SCBE
(6) MSE
(7) NBS
(8) ADM
(9) SoH
(10) SSS
(11) WKWSCl
(12) SBS
(13) SPMS
(14) ASE
(15) LKCSOM
(16) NIE
Select the course's school: 1
(1)Lecture only
(2)Lecture and Tutorial
(3)Lecture, Tutorial and Lab
Select the course type: 1
Enter the number of AUs for this course: 1
Select time and location for the lecture:
Enter the day of the lecture (1 for Monday to 5 for Friday): 1
Enter the duration (number of hours) of the lecture: 1
Select the starting time slot
(1) 08:30
(2) 09:30
(3) 10:30
(4) 11:30
(5) 12:30
(6) 13:30
(7) 14:30
(8) 15:30
(9) 16:30
(10) 17:30
(11) 18:30
(12) 19:30
```

```
Select the starting time slot (1 - 12): 1
Select a venue (1-9):
1) LT1
2) LT2
3) LT3
4) TR1
5) TR2
6) TR3
7) SWLab
8) HWLab
9) Project Lab
1
Course code already exists!
Failed to add course.. Please try again!
```

Adding new index to course:

```
=====ADD NEW INDEX=====
Enter the course code to add new index to: CZ2002
Enter the new index number: 10001
Enter the vacancy (>0): 30
Index 10001 created for course CZ2002
```

#### 4. Register for course

Registering for a course:

```
=====ADD NEW COURSE=====
Enter Course Code: CZ2002
Index: Index Number/Vacancy/Length of Waitlist
10001/30/0
Please select an index: 10001
Successfully enrolled in CZ2002
```

Registering for a course with no vacancies:

```
=====ADD NEW COURSE=====
Enter Course Code: CZ2003
Index: Index Number/Vacancy/Length of Waitlist
10001/0/0
Please select an index: 10001
Sorry, there are no more vacancies left for index 10001
Do you want to enter the waiting list (y/n)? y
You've been added to the waiting list!
```

Email sent to student after entering waitlist:

Waitlist Confirmation for AY2020-2021 SEMESTER 1 STAR WARS



[ntustarssystem@gmail.com](mailto:ntustarssystem@gmail.com)

Tue 11/24/2020 12:40 AM

To: #LIONEL WONG ZHI NENG#

Dear lionel

This is to inform you that you have been placed in the waitlist for the following course and index:

Course: CZ2003 CGV

Index: 10001

Status: Onwait

Regards

STARS System

Registering for course already registered for:

```
=====ADD NEW COURSE=====
Enter Course Code: CZ2002
Sorry, you're already enrolled in this course/on the waiting list.
```

Registering for a course that does not exist:

```
=====ADD NEW COURSE=====
Enter Course Code: CZ2009
Course does not exist! Please try again.
```

## 5. Check for vacancy in a class

Checking for vacancy of a particular course index:

```
=====CHECK VACANCIES AVAILABLE=====
Enter the course code: CZ2002
Enter the index number: 10001
Vacancy remaining: 29/30
```

Checking for vacancy of an index that does not exist for the course:

```
=====CHECK VACANCIES AVAILABLE=====
Enter the course code: CZ2002
Enter the index number: 9999
Course does not have the index inputed! Please try again.
```

## 6. Day/Time clash with other course

Registering for course that clashes with other courses in timetable:

```
=====ADD NEW COURSE=====
Enter Course Code: CZ2001
Index: Index Number/Vacancy/Length of Waitlist
10001/30/0
Please select an index: 10001
Sorry, index clashes in timetable..
```

## 7. Waitlist notification

Student dropping course for another student on waitlist to register for course:

```
=====DROP COURSE=====
Courses enrolled:
1) 10001 CZ2003 CGV Registered
Select a number (1) to drop or Enter 0 to exit: 1
Successfully dropped CZ2003 - CGV
```

Confirmation of registered course from waitlist:

Course registered for AY2020-2021 SEMESTER 1 STAR WARS



ntustarssystem@gmail.com

Tue 11/24/2020 12:57 AM

To: #LIONEL WONG ZHI NENG#

Dear lionel

Matric ID: u9182859f

This is to confirm that you have registered in Semester 1, AY2020-21 the following course:

Course details: CZ2003 CGV

Index: 10001

Status: Registered

Please do check your degree audit for your Outstanding AUs.

Please view your timetable from Print/Check Course Registered.

Both links are available at StudentLink->Academic Matters->Course Registration website.

Regards

STARS System

Confirmation of course added from waitlist:

```
=====PRINT COURSE REGISTERED=====
Courses Registered for lionel:
CZ2002 Algorithms Index: 10001
CZ2003 CGV Index: 10001
```

Confirmation of new timetable:

=====STUDENT TIMETABLE=====			
Timetable for Student lionel			
MONDAY			
08:30: Odd Week - CZ2002 LEC LT1	Even Week - CZ2002 LEC LT1		
09:30: Odd Week - CZ2002 LEC LT1	Even Week - CZ2002 LEC LT1		
10:30: Odd Week - CZ2002 LEC LT1	Even Week - CZ2002 LEC LT1		
11:30: Odd Week - NIL	Even Week - NIL		
12:30: Odd Week - NIL	Even Week - NIL		
13:30: Odd Week - NIL	Even Week - NIL		
14:30: Odd Week - NIL	Even Week - NIL		
15:30: Odd Week - NIL	Even Week - NIL		
16:30: Odd Week - NIL	Even Week - NIL		
17:30: Odd Week - NIL	Even Week - NIL		
18:30: Odd Week - NIL	Even Week - NIL		
19:30: Odd Week - NIL	Even Week - NIL		

FRIDAY			
08:30: Odd Week - NIL	Even Week - NIL		
09:30: Odd Week - NIL	Even Week - NIL		
10:30: Odd Week - NIL	Even Week - NIL		
11:30: Odd Week - NIL	Even Week - NIL		
12:30: Odd Week - NIL	Even Week - NIL		
13:30: Odd Week - NIL	Even Week - NIL		
14:30: Odd Week - NIL	Even Week - NIL		
15:30: Odd Week - NIL	Even Week - NIL		
16:30: Odd Week - NIL	Even Week - NIL		
17:30: Odd Week - NIL	Even Week - NIL		
18:30: Odd Week - NIL	Even Week - NIL		
19:30: Odd Week - CZ2003 LEC LT1	Even Week - CZ2003 LEC LT1		

## 8. Print student list by index number, course

Printing student list by course:

```
=====PRINT STUDENT LIST BY COURSE=====
Enter the course code: CZ2002
lionel (Male) sg
```

Print student list by index:

```
=====PRINT STUDENT LIST BY INDEX NUMBER=====
Enter the course code: CZ2002
Enter the index number: 10001
lionel (Male) sg
```

Printing student list of non-existent course:

```
=====PRINT STUDENT LIST BY COURSE=====
Enter the course code: CZ2009
Course does not exist! Please try again.
```

Printing student list of non-existent index of a course:

```
=====PRINT STUDENT LIST BY INDEX NUMBER=====
Enter the course code: CZ2002
Enter the index number: 9999
Course does not have the index inputed! Please try again.
```

Print venue timetable:

```
=====PRINT VENUE=====
1) LT1
2) LT2
3) LT3
4) TR1
5) TR2
6) TR3
7) SwLab
8) HwLab
9) Project Lab
Please select a venue number: 1
Timetable for Venue LT1
MONDAY
08:30: Odd Week - CZ2002 LEC          Even Week - CZ2002 LEC
09:30: Odd Week - CZ2002 LEC          Even Week - CZ2002 LEC
10:30: Odd Week - CZ2002 LEC          Even Week - CZ2002 LEC
11:30: Odd Week - NIL                 Even Week - NIL
12:30: Odd Week - NIL                 Even Week - NIL
13:30: Odd Week - NIL                 Even Week - NIL
14:30: Odd Week - NIL                 Even Week - NIL
15:30: Odd Week - NIL                 Even Week - NIL
16:30: Odd Week - NIL                 Even Week - NIL
17:30: Odd Week - NIL                 Even Week - NIL
18:30: Odd Week - NIL                 Even Week - NIL
19:30: Odd Week - NIL                 Even Week - NIL
TUESDAY
```

Print all courses available:

```
=====COURSES=====

List of all Courses:
CZ2002: Algorithms LKCSOM Number of AUs: 3
CZ2003: CGV SCSE Number of AUs: 3
CZ2001: OODP SCSE Number of AUs: 3
```

Print all indexes of a course:

```
=====INDEXES OF A COURSE=====
Enter the course code: CZ2002
Index: Index Number/Vacancy/Length of Waitlist
10001/29/0
10002/30/0
```

View lesson timings and information of specific index:

```
=====VIEW LESSON TIMINGS FOR INDEX=====
Enter the course code: CZ2002
Enter the index number: 10001

Index 10001 for Course CZ2002
Vacancy remaining: 29/30
Length of waiting list: 0
Lecture: MONDAY 08:30 - 11:30 LT1
```

## 9. Changing index of a course

Changing index successfully:

```
=====CHANGE INDEX NUMBER OF COURSE=====
Enter Course Code: CZ2002
Enter Current Index Number: 10001
Enter Index Number you would like to change to: 10002
Index successfully changed from 10001 to 10002.
```



Changing index but timetable clashes:

```
=====CHANGE INDEX NUMBER OF COURSE=====
Enter Course Code: CZ2004
Enter Current Index Number: 10001
Enter Index Number you would like to change to: 10002
Sorry, new Index clashes with current timetable..
Failed to change index number. Please try again!
```

Changing index but no vacancies for slot:

```
=====CHANGE INDEX NUMBER OF COURSE=====
Enter Course Code: CZ2004
Enter Current Index Number: 10001
Enter Index Number you would like to change to: 1003
Sorry, new Index has no more vacancy..
Failed to change index number. Please try again!
```

## 10. Swapping index of a course with another student

Swapping successfully:

```
=====SWAP INDEX NUMBER WITH ANOTHER STUDENT=====
Please input your course code: CZ2004
Please input your current index: 10001
Please input your partner's index: 10003
Please input your partner's username: DAVID
Please input your partner's password:
Swap successful!
```

Unsuccessful swap due to clash in timetable:

```
=====SWAP INDEX NUMBER WITH ANOTHER STUDENT=====
Please input your course code: CZ2004
Please input your current index: 10001
Please input your partner's index: 10002
Please input your partner's username: CHARLIE
Please input your partner's password:
Sorry, new Index clashes with current timetable..
Swap is unsuccessful!
```

References:

Eugenp. (n.d.). Eugenp/tutorials. Retrieved November 24, 2020, from  
<https://github.com/eugenp/tutorials/tree/master/core-java-modules/core-java-security-2>