NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

CZ2002 MyStars Report
SS14 Group 3

Lim Ming Aun Ashton U1922375J
Neo Guat Kwan U1921843D
Chong Zhe Ming U1920757K
Peng Wei Xing U1921133E
Li Yibai U1923270E

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	Lab Group	Signature /Date
PENG WEIXING	CZ2002	SS14	[illegible] 24/11/2020
CHONG ZHE MING	CZ2002	SS14	[illegible] 24/11/2020
LIM MING AUN ASHTON	CZ2002	SS14	[illegible] 24/11/2020
NEO GUAT KWAN	CZ2002	SS14	[illegible] 24/11/2020
LI YIBAI	CZ2002	SS14	[illegible] 24/11/2020

Demonstration of MyStars - https://www.youtube.com/watch?v=AqQC8HajPaY

Design Considerations

1. Approach Taken

Entity-Control-Boundary (ECB) pattern:

The design implements the ECB pattern. An actor will only interact with the boundary class for reasons such as navigating the interface and entering input. The boundary class then passes on any user command or input to the controller class where it will communicate with the entities for logic handling and data gathering, forming 3 layers within the application. From Figure 1, it can be seen that each type of class has a distinct role in the system. This segregation helps to further reinforce the concept of encapsulation since boundary classes will not have access to entities and vice versa. This also reduces tight coupling as boundary classes will not be linked to entities and will not be affected by any changes made to entities.

	Interaction			
	Actor	Boundary	Control	Entity
Actor	Yes	Yes	No	No
Boundary	Yes	As whole/part	Yes	No
Control	No	Yes	Yes	Yes
Entity	No	No	Yes	Yes

Figure 1. Table showing Interaction between the classes in the ECB pattern

Data Handling

In designing the application, a consideration was that no database application is allowed, so data is stored and retrieved in a binary format using a serialization implementation. In order to query data efficiently, objects are stored hierarchically using array lists, i.e. faculties hold courses which hold indexes and so on. For example, *Course* class keeps a list of *Index* classes while each *Index* also has a reference to their respective *Course*. This way, we can quickly access the list of indexes under a specific course without searching all indexes in the system. However, one drawback of such design is that classes are tightly coupled with each other.

Application Extensibility:

The notification implementation aims to be extensible by allowing for multiple modes of communication. There is an Interface *INotification* for different messaging API to implement in order to integrate into the system. In our program, send() method needs to be implemented to send out notifications via different APIs. We have implemented a sample SMS API (*SMSNotification*) in our program to demonstrate this extensibility.

2. Principles Used

The application applies concepts such as data encapsulation, SOLID design principles, loose coupling and high cohesion, with consideration for extensibility, reusability and maintainability. The implementation largely considers S, O, L and D in SOLID. The I (Interface Segregation Principle) however is not applicable for our design.

Single Responsibility Principle (SRP):

Each class should have one responsibility to achieve low coupling. An example would be a class for printing objects in different formats, components across the application are then able to tap on the printing capabilities of that class, reducing the need to change code in multiple places. In the design, classes have distinct roles to play for example the *AdminController* and *StudentController* class will only provide functions that carry out their respective users' operations. By having two controller classes for each user type rather than a single one for both will ensure that changes made to student functions will not affect admin functions.

Open-Closed Principle (OCP):

OCP essentially follows the idea that modules should be open for extension but closed for modification. Interfaces and abstractions facilitate this idea. In the design, both *Student* and *Admin* classes inherit from the parent class *User*, which contains attributes such as username and password required by both classes. If a new type of user needs to be introduced into the system, for example a *Staff* class that can only change courses but cannot add students, it can be extended from the *User* class without any changes made to the *User* class. Similarly, the design also contains *StudentView* and *AdminView* which implements the *UserView* interface. A new view class can be extended from the interface to cater to the *Staff* class without having to disturb any other parts of the code. This way, the *User* class and *UserView* interface has demonstrated that they are open for extension but closed for modification.

Liskov Substitution Principle (LSP):

Generally, LSP states that if all superclasses were replaced with its subclasses, the program or application should still function. This principle is demonstrated by the function *verifyLogin(User user, String password)* where the base class, *User*, needs to provide the function with its stored password. If a *Student* or *Admin* class, both of which are subclasses of the *User* class, is passed into the function, they are also able to provide the stored password. As such, the *verifyLogin()* can still function properly when the base class is substituted with the derived class.

Interface Segregation Principle (ISP)

The only interface that is present in the design is *UserView*, which is implemented by *StudentView* and *AdminView*. However, there was no need to segregate the interface further as both classes implement all the functions in the interface.

Dependency Inversion Principle (DIP):

This principle works on 2 parts:

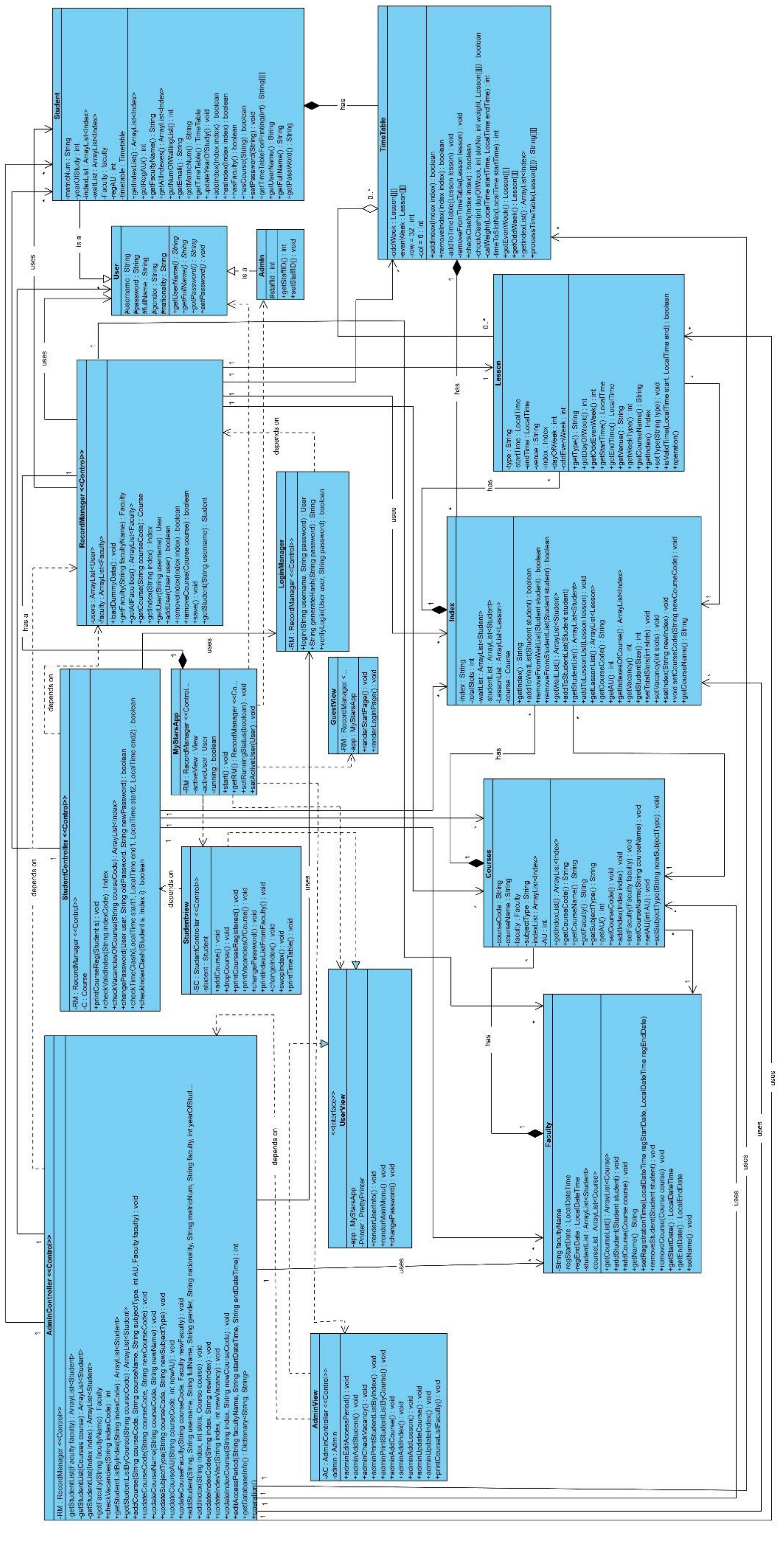
1. High-level modules should not depend on low-level ones where instead, both should depend on abstractions.
2. Abstractions should not depend on details; details should depend on abstractions.

One example of this could be wrapping the Email and SMS modules in an abstraction layer that the application code interfaces with, instead of directly communicating with those packages. The benefit of this is that the application code does not need to change every time the packages change. The wrapper also allows a developer to create a custom way to communicate with the main application.

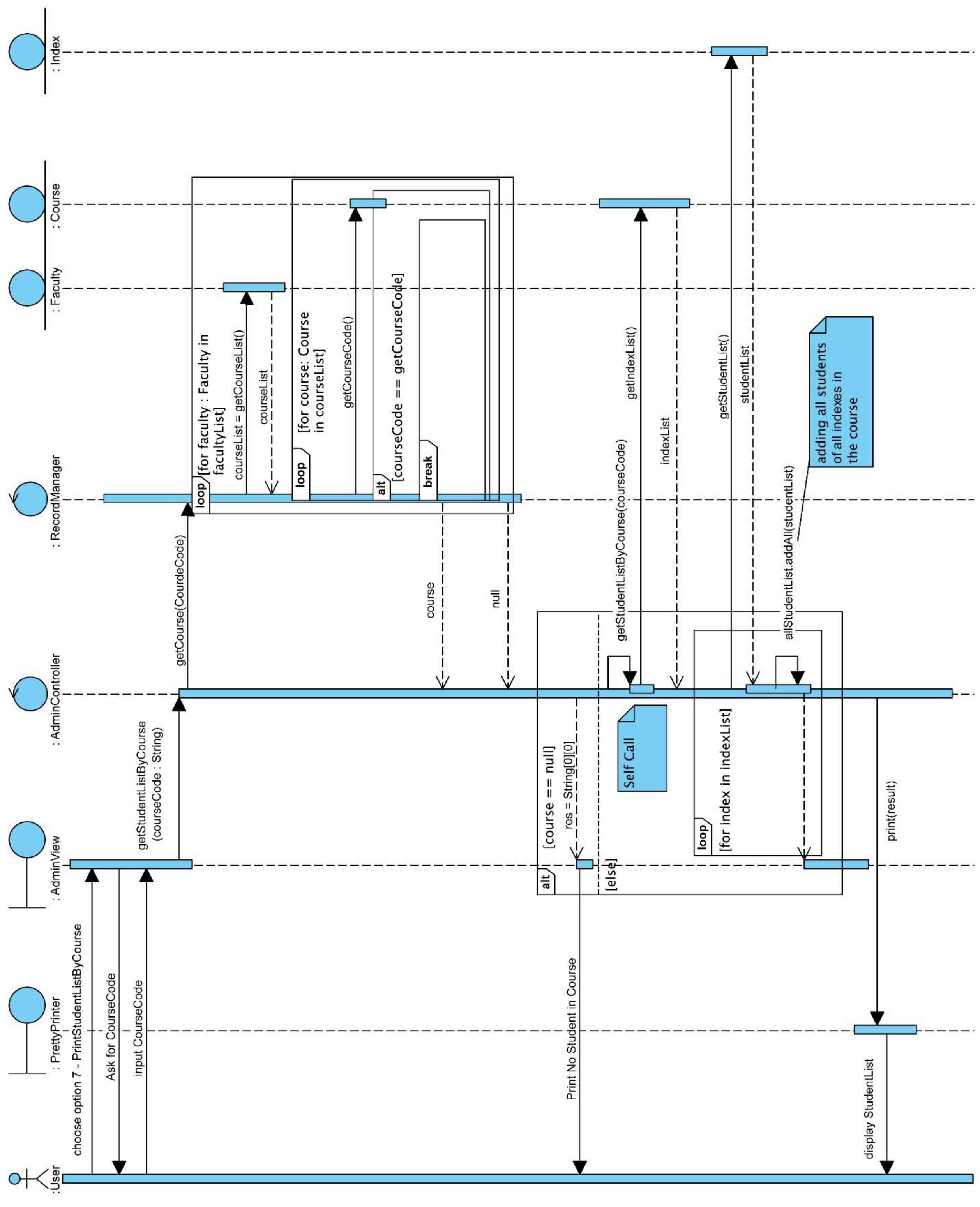
3. Assumptions

1. Administrators do not require an email. Not all notification methods are necessary. An example of sending out an email blast is coded but not used.
2. Administrators are system administrators rather than staff under faculties, so they are able to modify data across faculties.

Detailed UML Class Diagram



Sequence Diagram (Print Student List By Course)

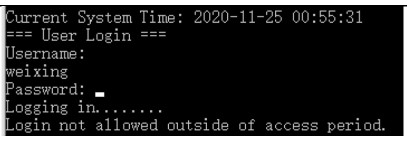


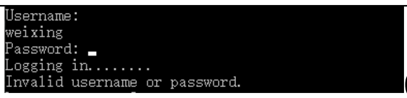
:User
: PrettyPrinter
: AdminView
: AdminController
: RecordManager
: Faculty
: Course
: Index
choose option 7 - PrintStudentListByCourse
Ask for CourseCode
input CourseCode
getStudentListByCourse (courseCode : String)
getCourse(CourdeCode)
loop
[for faculty : Faculty in facultyList]
courseList = getCourseList()
courseList
loop
[for course: Course in courseList]
getCourseCode()
alt
[courseCode == getCourseCode]
break
course
null
alt
[course == null]
res = String[0][0]
Print No Student in Course
[else]
getStudentListByCourse(courseCode)
getIndexList()
indexList
Self Call
loop
[for index in indexList]
getStudentList()
studentList
allStudentList.addAll(studentList)
adding all students of all indexes in the course
print(result)
display StudentList

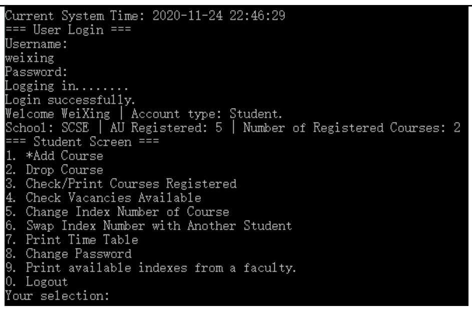
Testing

Student User

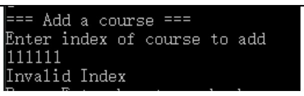
Test Cases	Test Result
Login	

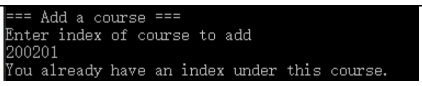
Login before or after allowed period:	
	`Current System Time: 2020-11-25 00:55:31`
	`=== User Login ===`
	`Username:`
	`weixing`
	`Password:`
	`Logging in........`
	`Login not allowed outside of access period.`

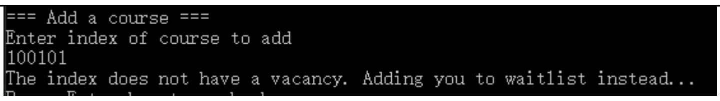
Login with wrong password	
	`Username:`
	`weixing`
	`Password:`
	`Logging in........`
	`Invalid username or password.`
	(password is hidden)

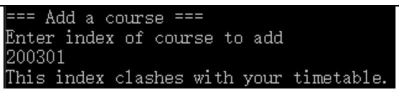
Successful Login	
	`Current System Time: 2020-11-24 22:46:29`
	`=== User Login ===`
	`Username:`
	`weixing`
	`Password:`
	`Logging in........`
	`Login successfully.`
	`Welcome WeiXing | Account type: Student.`
	`School: SCSE | AU Registered: 5 | Number of Registered Courses: 2`
	`=== Student Screen ===`
	`1.  *Add Course`
	`2.  Drop Course`
	`3.  Check/Print Courses Registered`
	`4.  Check Vacancies Available`
	`5.  Change Index Number of Course`
	`6.  Swap Index Number with Another Student`
	`7.  Print Time Table`
	`8.  Change Password`
	`9.  Print available indexes from a faculty.`
	`0.  Logout`
	`Your selection:`

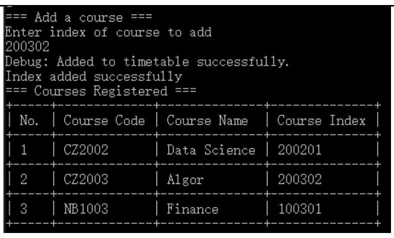
Add a course (do by add an index in this course)	

Invalid index number	
	`=== Add a course ===`
	`Enter index of course to add`
	`111111`
	`Invalid Index`

Enter registered index again	
	`=== Add a course ===`
	`Enter index of course to add`
	`200201`
	`You already have an index under this course.`

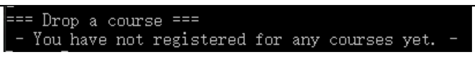
Enter an index with 0 vacancy	
	`=== Add a course ===`
	`Enter index of course to add`
	`100101`
	`The index does not have a vacancy.  Adding you to waitlist instead...`

New index clashes with registered indexes	
	`=== Add a course ===`
	`Enter index of course to add`
	`200301`
	`This index clashes with your timetable.`

Successful add	
	`=== Add a course ===`
	`Enter index of course to add`
	`200302`
	`Debug: Added to timetable successfully.`
	`Index added successfully`
	`=== Courses Registered ===`
	`+-----+-------------+--------------+--------------+`
	`| No. | Course Code | Course Name  | Course Index |`
	`+-----+-------------+--------------+--------------+`
	`| 1   | CZ2002      | Data Science | 200201       |`
	`+-----+-------------+--------------+--------------+`
	`| 2   | CZ2003      | Algor        | 200302       |`
	`+-----+-------------+--------------+--------------+`
	`| 3   | NB1003      | Finance      | 100301       |`
	`+-----+-------------+--------------+--------------+`

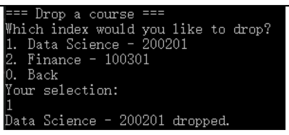
Drop a course (do by drop user's index in this course)	

User has not registered any course yet.



```
=== Drop a course ===
 - You have not registered for any courses yet. -
```

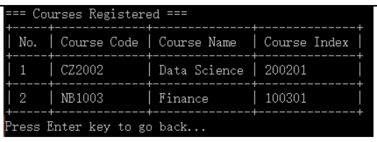
Success



```
=== Drop a course ===
Which index would you like to drop?
1. Data Science - 200201
2. Finance - 100301
0. Back
Your selection:
1
Data Science - 200201 dropped.
```

Check/Print courses registered

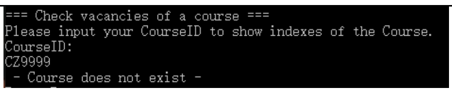
Success



```
=== Courses Registered ===
+-----+--------------+--------------+--------------+
| No. | Course Code  | Course Name  | Course Index |
+-----+--------------+--------------+--------------+
| 1   | CZ2002       | Data Science | 200201       |
+-----+--------------+--------------+--------------+
| 2   | NB1003       | Finance      | 100301       |
+-----+--------------+--------------+--------------+
Press Enter key to go back...
```

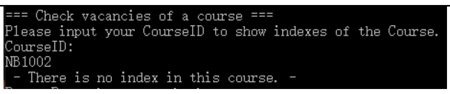
Check vacancies available of course

Invalid course code entered.



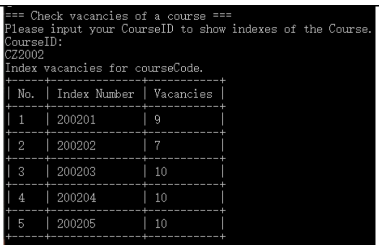
```
=== Check vacancies of a course ===
Please input your CourseID to show indexes of the Course.
CourseID:
CZ9999
 - Course does not exist -
```

Course does not have index yet.



```
=== Check vacancies of a course ===
Please input your CourseID to show indexes of the Course.
CourseID:
NB1002
 - There is no index in this course. -
```

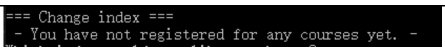
Success



```
=== Check vacancies of a course ===
Please input your CourseID to show indexes of the Course.
CourseID:
CZ2002
Index vacancies for courseCode.
+-----+--------------+-----------+
| No. | Index Number | Vacancies |
+-----+--------------+-----------+
| 1   | 200201       | 9         |
+-----+--------------+-----------+
| 2   | 200202       | 7         |
+-----+--------------+-----------+
| 3   | 200203       | 10        |
+-----+--------------+-----------+
| 4   | 200204       | 10        |
+-----+--------------+-----------+
| 5   | 200205       | 10        |
+-----+--------------+-----------+
```

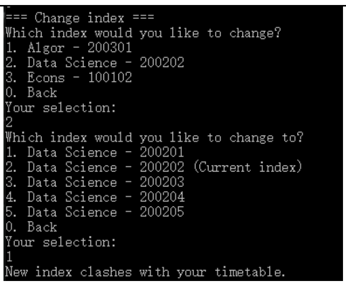
Change index number of course

No registered course yet.



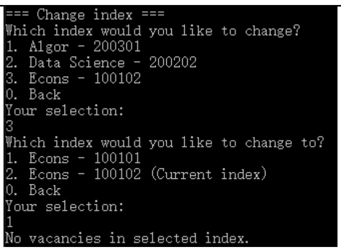
```
=== Change index ===
 - You have not registered for any courses yet. -
```

New index clashes with timetable

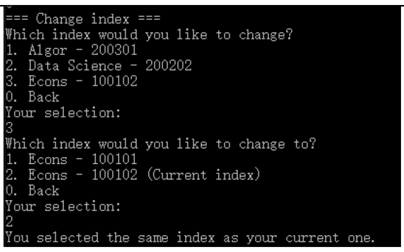


```
=== Change index ===
Which index would you like to change?
1. Algor - 200301
2. Data Science - 200202
3. Econs - 100102
0. Back
Your selection:
2
Which index would you like to change to?
1. Data Science - 200201
2. Data Science - 200202 (Current index)
3. Data Science - 200203
4. Data Science - 200204
5. Data Science - 200205
0. Back
Your selection:
1
New index clashes with your timetable.
```

No vacancies in new index / same index selected

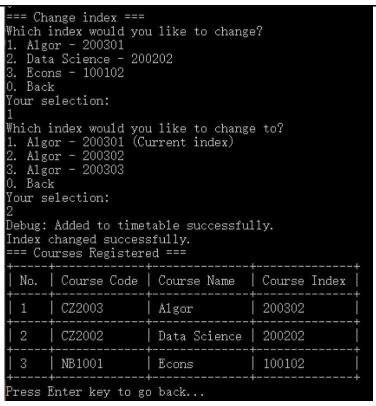


```
=== Change index ===
Which index would you like to change?
1. Algor - 200301
2. Data Science - 200202
3. Econs - 100102
0. Back
Your selection:
3
Which index would you like to change to?
1. Econs - 100101
2. Econs - 100102 (Current index)
0. Back
Your selection:
1
No vacancies in selected index.
```



```
=== Change index ===
Which index would you like to change?
1. Algor - 200301
2. Data Science - 200202
3. Econs - 100102
0. Back
Your selection:
3
Which index would you like to change to?
1. Econs - 100101
2. Econs - 100102 (Current index)
0. Back
Your selection:
2
You selected the same index as your current one.
```

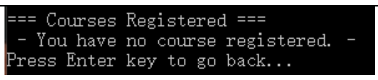
Success



```
=== Change index ===
Which index would you like to change?
1. Algor - 200301
2. Data Science - 200202
3. Econs - 100102
0. Back
Your selection:
1
Which index would you like to change to?
1. Algor - 200301 (Current index)
2. Algor - 200302
3. Algor - 200303
0. Back
Your selection:
2
Debug: Added to timetable successfully.
Index changed successfully.
=== Courses Registered ===
+-----+-------------+--------------+--------------+
| No. | Course Code | Course Name  | Course Index |
+-----+-------------+--------------+--------------+
| 1   | CZ2003      | Algor        | 200302       |
+-----+-------------+--------------+--------------+
| 2   | CZ2002      | Data Science | 200202       |
+-----+-------------+--------------+--------------+
| 3   | NB1001      | Econs        | 100102       |
+-----+-------------+--------------+--------------+
Press Enter key to go back...
```

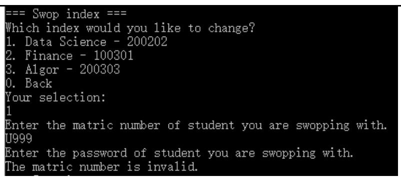
Swap Index Number with Another Student

Student does not have any courses registered.

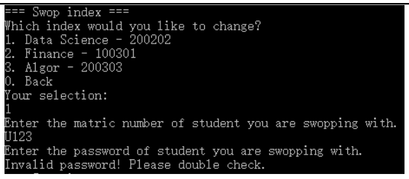


```
=== Courses Registered ===
 - You have no course registered. -
Press Enter key to go back...
```

Invalid partner’s matric number / password entered

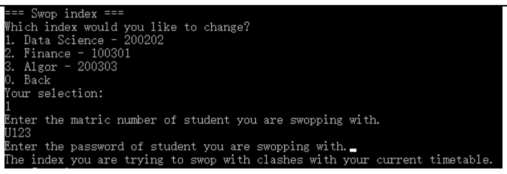


```
=== Swop index ===
Which index would you like to change?
1. Data Science - 200202
2. Finance - 100301
3. Algor - 200303
0. Back
Your selection:
1
Enter the matric number of student you are swopping with.
U999
Enter the password of student you are swopping with.
The matric number is invalid.
```



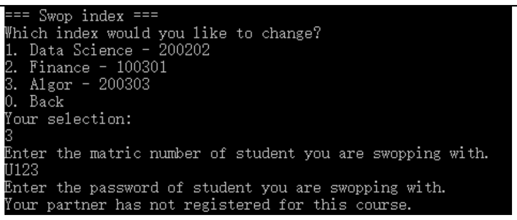
```
=== Swop index ===
Which index would you like to change?
1. Data Science - 200202
2. Finance - 100301
3. Algor - 200303
0. Back
Your selection:
1
Enter the matric number of student you are swopping with.
U123
Enter the password of student you are swopping with.
Invalid password! Please double check.
```

Partner’s index clashes with student’s timetable



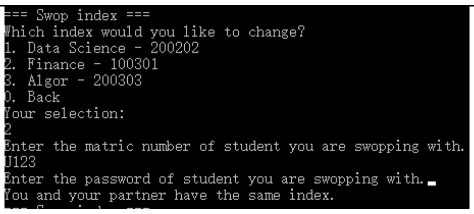
```
=== Swop index ===
Which index would you like to change?
1. Data Science - 200202
2. Finance - 100301
3. Algor - 200303
0. Back
Your selection:
1
Enter the matric number of student you are swopping with.
U123
Enter the password of student you are swopping with.
The index you are trying to swop with clashes with your current timetable.
```

Partner does not have the index of the course student is swapping.



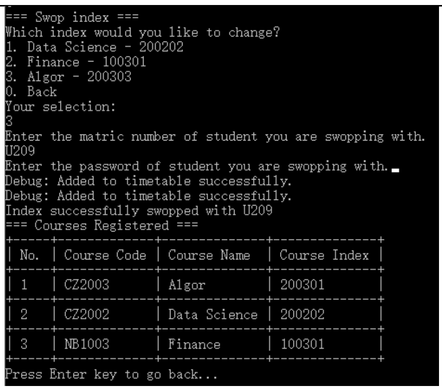
```
=== Swop index ===
Which index would you like to change?
1. Data Science - 200202
2. Finance - 100301
3. Algor - 200303
0. Back
Your selection:
3
Enter the matric number of student you are swopping with.
U123
Enter the password of student you are swopping with.
Your partner has not registered for this course.
```

Student and the partner have the same index for the course.



```
=== Swop index ===
Which index would you like to change?
1. Data Science - 200202
2. Finance - 100301
3. Algor - 200303
0. Back
Your selection:
2
Enter the matric number of student you are swopping with.
U123
Enter the password of student you are swopping with.
You and your partner have the same index.
```

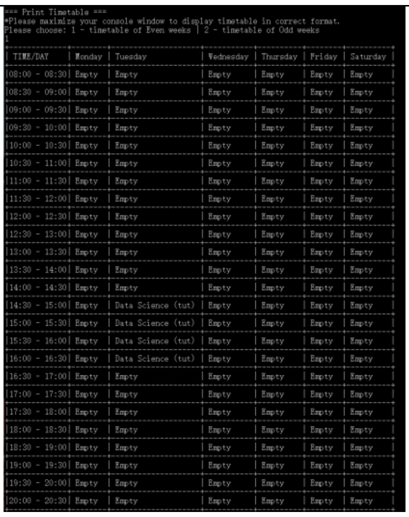
Success



```
=== Swop index ===
Which index would you like to change?
1. Data Science - 200202
2. Finance - 100301
3. Algor - 200303
0. Back
Your selection:
3
Enter the matric number of student you are swopping with.
U209
Enter the password of student you are swopping with.
Debug: Added to timetable successfully.
Debug: Added to timetable successfully.
Index successfully swopped with U209
=== Courses Registered ===
+-----+-------------+--------------+--------------+
| No. | Course Code | Course Name  | Course Index |
+-----+-------------+--------------+--------------+
| 1   | CZ2003      | Algor        | 200301       |
+-----+-------------+--------------+--------------+
| 2   | CZ2002      | Data Science | 200202       |
+-----+-------------+--------------+--------------+
| 3   | NB1003      | Finance      | 100301       |
+-----+-------------+--------------+--------------+
Press Enter key to go back...
```

Print timetable

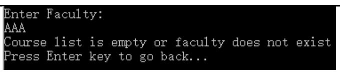
Success



```
=== Print Timetable ===
*Please maximize your console window to display timetable in correct format.
Please choose: 1 - timetable of Even weeks | 2 - timetable of Odd weeks
1
+-------------+--------+--------------------+-----------+----------+--------+----------+
| TIME/DAY    | Monday | Tuesday            | Wednesday | Thursday | Friday | Saturday |
+-------------+--------+--------------------+-----------+----------+--------+----------+
|08:00 - 08:30| Empty  | Empty              | Empty     | Empty    | Empty  | Empty    |
+-------------+--------+--------------------+-----------+----------+--------+----------+
|08:30 - 09:00| Empty  | Empty              | Empty     | Empty    | Empty  | Empty    |
+-------------+--------+--------------------+-----------+----------+--------+----------+
|09:00 - 09:30| Empty  | Empty              | Empty     | Empty    | Empty  | Empty    |
+-------------+--------+--------------------+-----------+----------+--------+----------+
|09:30 - 10:00| Empty  | Empty              | Empty     | Empty    | Empty  | Empty    |
+-------------+--------+--------------------+-----------+----------+--------+----------+
|10:00 - 10:30| Empty  | Empty              | Empty     | Empty    | Empty  | Empty    |
+-------------+--------+--------------------+-----------+----------+--------+----------+
|10:30 - 11:00| Empty  | Empty              | Empty     | Empty    | Empty  | Empty    |
+-------------+--------+--------------------+-----------+----------+--------+----------+
|11:00 - 11:30| Empty  | Empty              | Empty     | Empty    | Empty  | Empty    |
+-------------+--------+--------------------+-----------+----------+--------+----------+
|11:30 - 12:00| Empty  | Empty              | Empty     | Empty    | Empty  | Empty    |
+-------------+--------+--------------------+-----------+----------+--------+----------+
|12:00 - 12:30| Empty  | Empty              | Empty     | Empty    | Empty  | Empty    |
+-------------+--------+--------------------+-----------+----------+--------+----------+
|12:30 - 13:00| Empty  | Empty              | Empty     | Empty    | Empty  | Empty    |
+-------------+--------+--------------------+-----------+----------+--------+----------+
|13:00 - 13:30| Empty  | Empty              | Empty     | Empty    | Empty  | Empty    |
+-------------+--------+--------------------+-----------+----------+--------+----------+
|13:30 - 14:00| Empty  | Empty              | Empty     | Empty    | Empty  | Empty    |
+-------------+--------+--------------------+-----------+----------+--------+----------+
|14:00 - 14:30| Empty  | Empty              | Empty     | Empty    | Empty  | Empty    |
+-------------+--------+--------------------+-----------+----------+--------+----------+
|14:30 - 15:00| Empty  | Data Science (tut) | Empty     | Empty    | Empty  | Empty    |
+-------------+--------+--------------------+-----------+----------+--------+----------+
|15:00 - 15:30| Empty  | Data Science (tut) | Empty     | Empty    | Empty  | Empty    |
+-------------+--------+--------------------+-----------+----------+--------+----------+
|15:30 - 16:00| Empty  | Data Science (tut) | Empty     | Empty    | Empty  | Empty    |
+-------------+--------+--------------------+-----------+----------+--------+----------+
|16:00 - 16:30| Empty  | Data Science (tut) | Empty     | Empty    | Empty  | Empty    |
+-------------+--------+--------------------+-----------+----------+--------+----------+
|16:30 - 17:00| Empty  | Empty              | Empty     | Empty    | Empty  | Empty    |
+-------------+--------+--------------------+-----------+----------+--------+----------+
|17:00 - 17:30| Empty  | Empty              | Empty     | Empty    | Empty  | Empty    |
+-------------+--------+--------------------+-----------+----------+--------+----------+
|17:30 - 18:00| Empty  | Empty              | Empty     | Empty    | Empty  | Empty    |
+-------------+--------+--------------------+-----------+----------+--------+----------+
|18:00 - 18:30| Empty  | Empty              | Empty     | Empty    | Empty  | Empty    |
+-------------+--------+--------------------+-----------+----------+--------+----------+
|18:30 - 19:00| Empty  | Empty              | Empty     | Empty    | Empty  | Empty    |
+-------------+--------+--------------------+-----------+----------+--------+----------+
|19:00 - 19:30| Empty  | Empty              | Empty     | Empty    | Empty  | Empty    |
+-------------+--------+--------------------+-----------+----------+--------+----------+
|19:30 - 20:00| Empty  | Empty              | Empty     | Empty    | Empty  | Empty    |
+-------------+--------+--------------------+-----------+----------+--------+----------+
|20:00 - 20:30| Empty  | Empty              | Empty     | Empty    | Empty  | Empty    |
+-------------+--------+--------------------+-----------+----------+--------+----------+
```

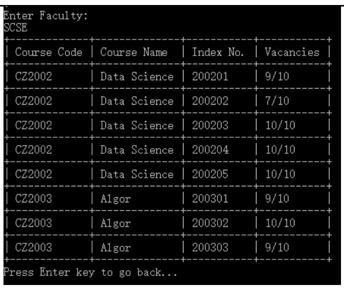
Print available indexes from a faculty

Invalid faculty / Faculty with no course created yet



```
Enter Faculty:
AAA
Course list is empty or faculty does not exist
Press Enter key to go back...
```

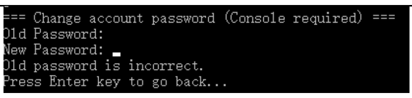
Success



```
Enter Faculty:
SCSE
+-------------+--------------+-----------+-----------+
| Course Code | Course Name  | Index No. | Vacancies |
+-------------+--------------+-----------+-----------+
| CZ2002      | Data Science | 200201    | 9/10      |
+-------------+--------------+-----------+-----------+
| CZ2002      | Data Science | 200202    | 7/10      |
+-------------+--------------+-----------+-----------+
| CZ2002      | Data Science | 200203    | 10/10     |
+-------------+--------------+-----------+-----------+
| CZ2002      | Data Science | 200204    | 10/10     |
+-------------+--------------+-----------+-----------+
| CZ2002      | Data Science | 200205    | 10/10     |
+-------------+--------------+-----------+-----------+
| CZ2003      | Algor        | 200301    | 9/10      |
+-------------+--------------+-----------+-----------+
| CZ2003      | Algor        | 200302    | 10/10     |
+-------------+--------------+-----------+-----------+
| CZ2003      | Algor        | 200303    | 9/10      |
+-------------+--------------+-----------+-----------+
Press Enter key to go back...
```

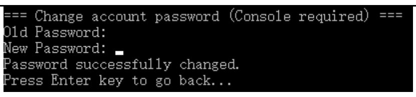
Change password

Wrong old password



```
=== Change account password (Console required) ===
Old Password:
New Password:
Old password is incorrect.
Press Enter key to go back...
```

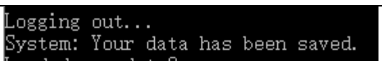
Success



```
=== Change account password (Console required) ===
Old Password:
New Password:
Password successfully changed.
Press Enter key to go back...
```

Logout

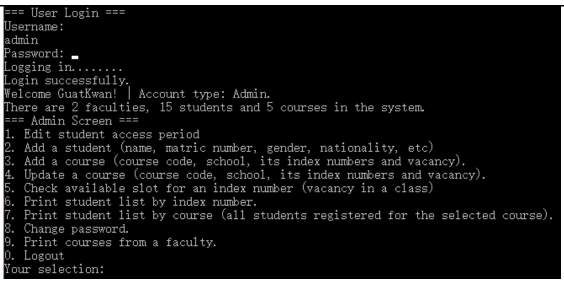
Success



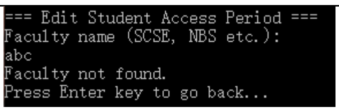
```
Logging out...
System: Your data has been saved.
```

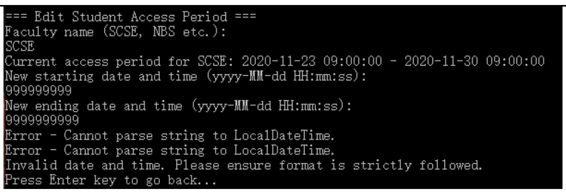
Admin User

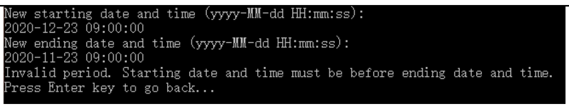
Test Cases	Test Results
Login	

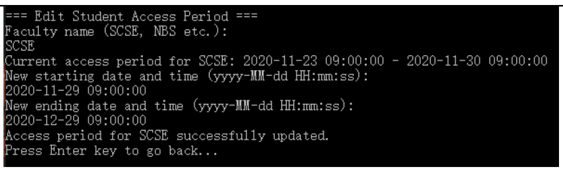
Success	
	`=== User Login ===`
	`Username:`
	`admin`
	`Password:`
	`Logging in........`
	`Login successfully.`
	`Welcome GuatKwan! | Account type: Admin.`
	`There are 2 faculties, 15 students and 5 courses in the system.`
	`=== Admin Screen ===`
	`1. Edit student access period`
	`2. Add a student (name, matric number, gender, nationality, etc)`
	`3. Add a course (course code, school, its index numbers and vacancy).`
	`4. Update a course (course code, school, its index numbers and vacancy).`
	`5. Check available slot for an index number (vacancy in a class)`
	`6. Print student list by index number.`
	`7. Print student list by course (all students registered for the selected course).`
	`8. Change password.`
	`9. Print courses from a faculty.`
	`0. Logout`
	`Your selection:`

Edit student access period	

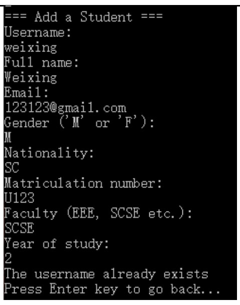
Invalid faculty name input.	
	`=== Edit Student Access Period ===`
	`Faculty name (SCSE, NBS etc.):`
	`abc`
	`Faculty not found.`
	`Press Enter key to go back...`

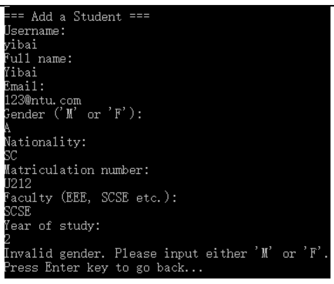
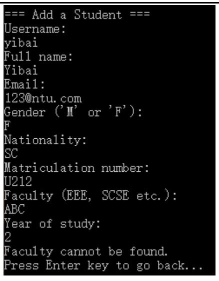
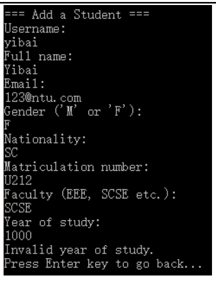
Invalid date and time format input.	
	`=== Edit Student Access Period ===`
	`Faculty name (SCSE, NBS etc.):`
	`SCSE`
	`Current access period for SCSE: 2020-11-23 09:00:00 - 2020-11-30 09:00:00`
	`New starting date and time (yyyy-MM-dd HH:mm:ss):`
	`999999999`
	`New ending date and time (yyyy-MM-dd HH:mm:ss):`
	`9999999999`
	`Error - Cannot parse string to LocalDateTime.`
	`Error - Cannot parse string to LocalDateTime.`
	`Invalid date and time. Please ensure format is strictly followed.`
	`Press Enter key to go back...`

Input start time later than end time.	
	`New starting date and time (yyyy-MM-dd HH:mm:ss):`
	`2020-12-23 09:00:00`
	`New ending date and time (yyyy-MM-dd HH:mm:ss):`
	`2020-11-23 09:00:00`
	`Invalid period. Starting date and time must be before ending date and time.`
	`Press Enter key to go back...`

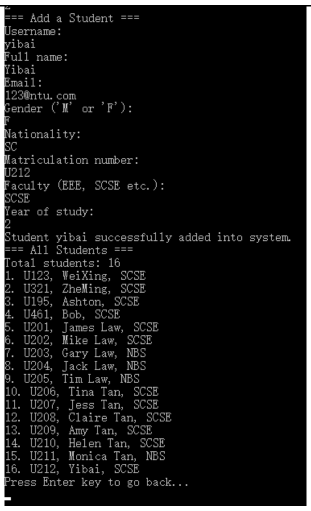
Success	
	`=== Edit Student Access Period ===`
	`Faculty name (SCSE, NBS etc.):`
	`SCSE`
	`Current access period for SCSE: 2020-11-23 09:00:00 - 2020-11-30 09:00:00`
	`New starting date and time (yyyy-MM-dd HH:mm:ss):`
	`2020-11-29 09:00:00`
	`New ending date and time (yyyy-MM-dd HH:mm:ss):`
	`2020-12-29 09:00:00`
	`Access period for SCSE successfully updated.`
	`Press Enter key to go back...`

Add a student (name, matric number, gender, nationality, etc)	

Add an already existed student	
	`=== Add a Student ===`
	`Username:`
	`weixing`
	`Full name:`
	`Weixing`
	`Email:`
	`123123@gmail.com`
	`Gender ('M' or 'F'):`
	`M`
	`Nationality:`
	`SC`
	`Matriculation number:`
	`U123`
	`Faculty (EEE, SCSE etc.):`
	`SCSE`
	`Year of study:`
	`2`
	`The username already exists`
	`Press Enter key to go back...`

Invalid gender information / faculty / year of study input	
	`=== Add a Student ===`
	`Username:`
	`yibai`
	`Full name:`
	`Yibai`
	`Email:`
	`123@ntu.com`
	`Gender ('M' or 'F'):`
	`A`
	`Nationality:`
	`SC`
	`Matriculation number:`
	`U212`
	`Faculty (EEE, SCSE etc.):`
	`SCSE`
	`Year of study:`
	`2`
	`Invalid gender. Please input either 'M' or 'F'.`
	`Press Enter key to go back...`
	
	`=== Add a Student ===`
	`Username:`
	`yibai`
	`Full name:`
	`Yibai`
	`Email:`
	`123@ntu.com`
	`Gender ('M' or 'F'):`
	`F`
	`Nationality:`
	`SC`
	`Matriculation number:`
	`U212`
	`Faculty (EEE, SCSE etc.):`
	`ABC`
	`Year of study:`
	`2`
	`Faculty cannot be found.`
	`Press Enter key to go back...`
	
	`=== Add a Student ===`
	`Username:`
	`yibai`
	`Full name:`
	`Yibai`
	`Email:`
	`123@ntu.com`
	`Gender ('M' or 'F'):`
	`F`
	`Nationality:`
	`SC`
	`Matriculation number:`
	`U212`
	`Faculty (EEE, SCSE etc.):`
	`SCSE`
	`Year of study:`
	`1000`
	`Invalid year of study.`
	`Press Enter key to go back...`

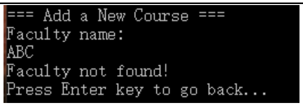
Success



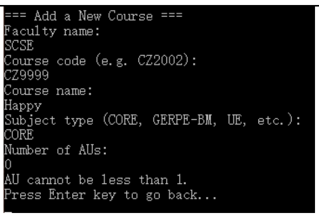
```
=== Add a Student ===
Username:
yibai
Full name:
Yibai
Email:
123@ntu.com
Gender ('M' or 'F'):
F
Nationality:
SC
Matriculation number:
U212
Faculty (EEE, SCSE etc.):
SCSE
Year of study:
2
Student yibai successfully added into system.
=== All Students ===
Total students: 16
1. U123, WeiXing, SCSE
2. U321, ZheMing, SCSE
3. U195, Ashton, SCSE
4. U461, Bob, SCSE
5. U201, James Law, SCSE
6. U202, Mike Law, SCSE
7. U203, Gary Law, NBS
8. U204, Jack Law, NBS
9. U205, Tim Law, NBS
10. U206, Tina Tan, SCSE
11. U207, Jess Tan, SCSE
12. U208, Claire Tan, SCSE
13. U209, Amy Tan, SCSE
14. U210, Helen Tan, SCSE
15. U211, Monica Tan, NBS
16. U212, Yibai, SCSE
Press Enter key to go back...
```

Add a course (course code, school, its index numbers and vacancy).

Invalid faculty / AU input

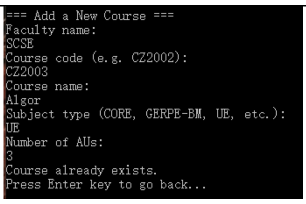


```
=== Add a New Course ===
Faculty name:
ABC
Faculty not found!
Press Enter key to go back...
```



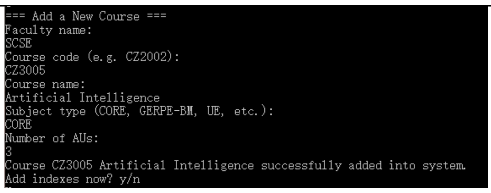
```
=== Add a New Course ===
Faculty name:
SCSE
Course code (e.g. CZ2002):
CZ9999
Course name:
Happy
Subject type (CORE, GERPE-BM, UE, etc.):
CORE
Number of AUs:
0
AU cannot be less than 1.
Press Enter key to go back...
```

Add an already existed course



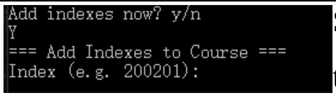
```
=== Add a New Course ===
Faculty name:
SCSE
Course code (e.g. CZ2002):
CZ2003
Course name:
Algor
Subject type (CORE, GERPE-BM, UE, etc.):
UE
Number of AUs:
3
Course already exists.
Press Enter key to go back...
```

Success



```
=== Add a New Course ===
Faculty name:
SCSE
Course code (e.g. CZ2002):
CZ3005
Course name:
Artificial Intelligence
Subject type (CORE, GERPE-BM, UE, etc.):
CORE
Number of AUs:
3
Course CZ3005 Artificial Intelligence successfully added into system.
Add indexes now? y/n
```

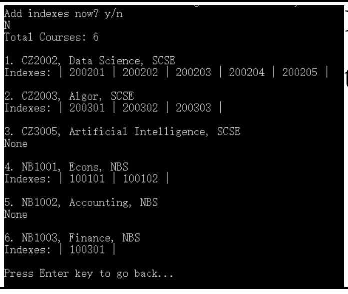
Choose Y



```
Add indexes now? y/n
Y
=== Add Indexes to Course ===
Index (e.g. 200201):
```

Start adding index to this course, which will be implemented in detail later.

Choose N

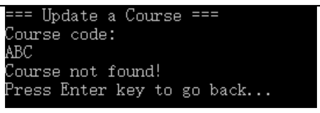


```
Add indexes now? y/n
N
Total Courses: 6

1. CZ2002, Data Science, SCSE
Indexes: | 200201 | 200202 | 200203 | 200204 | 200205 |

2. CZ2003, Algor, SCSE
Indexes: | 200301 | 200302 | 200303 |

3. CZ3005, Artificial Intelligence, SCSE
None

4. NB1001, Econs, NBS
Indexes: | 100101 | 100102 |

5. NB1002, Accounting, NBS
None

6. NB1003, Finance, NBS
Indexes: | 100301 |

Press Enter key to go back...
```

Print all courses with their indexes

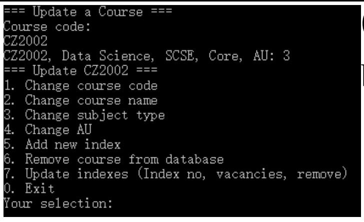
Update a course (course code, school, its index numbers and vacancy).

Invalid course code input



```
=== Update a Course ===
Course code:
ABC
Course not found!
Press Enter key to go back...
```

Success

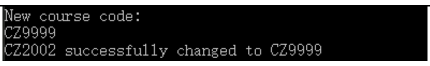


```
=== Update a Course ===
Course code:
CZ2002
CZ2002, Data Science, SCSE, Core, AU: 3
=== Update CZ2002 ===
1. Change course code
2. Change course name
3. Change subject type
4. Change AU
5. Add new index
6. Remove course from database
7. Update indexes (Index no, vacancies, remove)
0. Exit
Your selection:
```

(Following 1-6 operations will based on course CZ2002)

1. Change course code

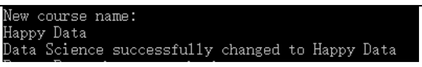
Success



```
New course code:
CZ9999
CZ2002 successfully changed to CZ9999
```

2. Change course name

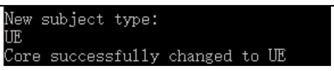
Success



```
New course name:
Happy Data
Data Science successfully changed to Happy Data
```

3. Change subject type

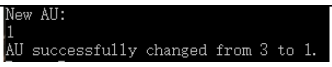
Success



```
New subject type:
UE
Core successfully changed to UE
```

4. Change AU

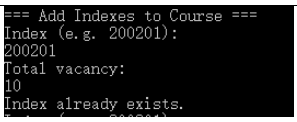
Success



```
New AU:
1
AU successfully changed from 3 to 1.
```

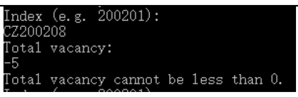
5. Add new Indexes

Add an already existed index.



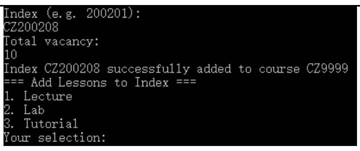
```
=== Add Indexes to Course ===
Index (e.g. 200201):
200201
Total vacancy:
10
Index already exists.
```

Invalid index vacancy input.



```
Index (e.g. 200201):
CZ200208
Total vacancy:
-5
Total vacancy cannot be less than 0.
```

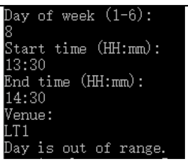
Success



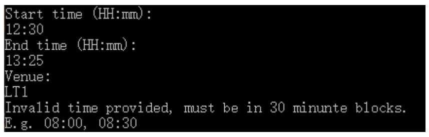
```
Index (e.g. 200201):
CZ200208
Total vacancy:
10
Index CZ200208 successfully added to course CZ9999
=== Add Lessons to Index ===
1. Lecture
2. Lab
3. Tutorial
Your selection:
```

Start adding lessons to this index

Invalid day / time input.

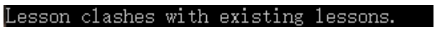


```
Day of week (1-6):
8
Start time (HH:mm):
13:30
End time (HH:mm):
14:30
Venue:
LT1
Day is out of range.
```



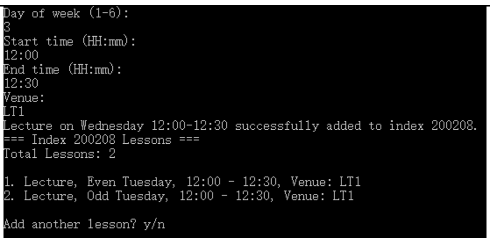
```
Start time (HH:mm):
12:30
End time (HH:mm):
13:25
Venue:
LT1
Invalid time provided, must be in 30 minunte blocks.
E.g. 08:00, 08:30
```

Clashed lesson input

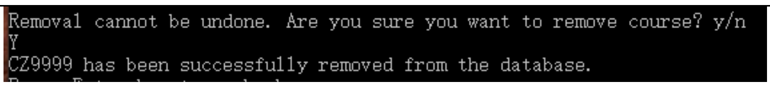
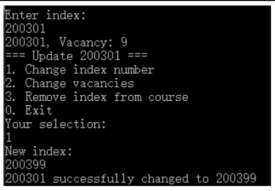
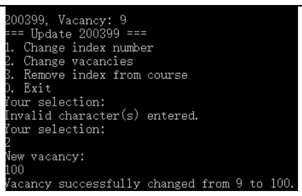
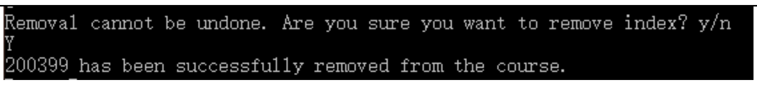
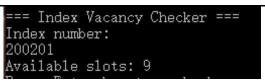
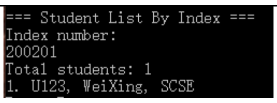
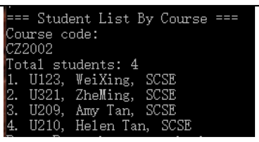


```
Lesson clashes with existing lessons.
```

Success

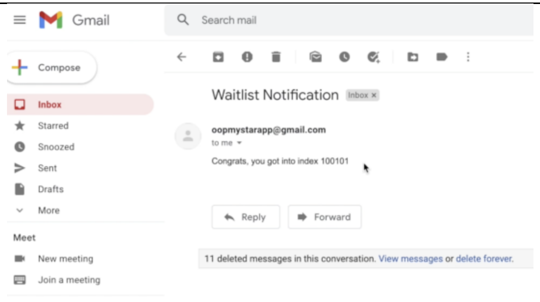


```
Day of week (1-6):
3
Start time (HH:mm):
12:00
End time (HH:mm):
12:30
Venue:
LT1
Lecture on Wednesday 12:00-12:30 successfully added to index 200208.
=== Index 200208 Lessons ===
Total Lessons: 2

1. Lecture, Even Tuesday, 12:00 - 12:30, Venue: LT1
2. Lecture, Odd Tuesday, 12:00 - 12:30, Venue: LT1

Add another lesson? y/n
```

		Choose Y will restart the process of adding lessons to this index; Choose N will print all courses with their indexes like in previous case ("add a course")
6.Remove course from database		
Success	 Removal cannot be undone. Are you sure you want to remove course? y/n Y CZ9999 has been successfully removed from the database.	
7. Update indexes (CZ9999 (ie. Renamed CZ2002) is removed, so we switched to update indexes in course CZ2003)		
Update Index number and vacancy	 Enter index: 200301 200301, Vacancy: 9 === Update 200301 === 1. Change index number 2. Change vacancies 3. Remove index from course 0. Exit Your selection: 1 New index: 200399 200301 successfully changed to 200399  200399, Vacancy: 9 === Update 200399 === 1. Change index number 2. Change vacancies 3. Remove index from course 0. Exit Your selection: Invalid character(s) entered. Your selection: 2 New vacancy: 100 Vacancy successfully changed from 9 to 100.	
Removal from the course	 Removal cannot be undone. Are you sure you want to remove index? y/n Y 200399 has been successfully removed from the course.	
Check available slot for an index number (vacancy in a class)		
Success	 === Index Vacancy Checker === Index number: 200201 Available slots: 9	
Print student list by index number.		
Success	 === Student List By Index === Index number: 200201 Total students: 1 1. U123, WeiXing, SCSE	
Print student list by course (all students registered for the selected course).		
Success	 === Student List By Course === Course code: CZ2002 Total students: 4 1. U123, WeiXing, SCSE 2. U321, ZheMing, SCSE 3. U209, Amy Tan, SCSE 4. U210, Helen Tan, SCSE	
"Print courses from a faculty", "Change password", "Logout" is the same as the last 3 operations in the Student part.		

E-mail Notification System

When a student user successfully drops an index, the program will check if this dropped index's wait list contains any students. If so, it will register the first student in the queue with this index and send him/her a notification email. Also if the admin adds more vacancies into an index, the waitlist check will be triggered as well.

Test Cases	**Test Results**
Student A adds in an index with zero vacancies. When Student B drops the same index, Student A will be removed from the waitlist and an email notification will be sent to him.	 Gmail Search mail Compose Inbox Starred Snoozed Sent Drafts More Meet New meeting Join a meeting Inbox × Reply Forward Waitlist Notification oopmystarapp@gmail.com to me Congrats, you got into index 100101 11 deleted messages in this conversation. View messages or delete forever.