Subject Code :	**COSC2440**
Subject Name:	**Further Programming**
Location:	**SGS (Sai Gon South Campus)**
Title of assignment:	**Assignment 1**
Student name:	**Luu Duy Toan**
Student number:	**S3777213**
Teachers name:	**Minh Vu Thanh**
Assignment due date:	**6/4/2022**
Date of submission:	**4/4/2022**
Number of pages:	**6**

I. Introduction:

This is the application of the enrollment system. In this system, there are 3 classes: StudentsEnrolment, Students, Courses, and the main classes called Main. Moreover, there is an interface in this system.

```java
public class StudentsEnrolment extends Main {
    Students students;
    Courses courses;
    String semester;


    public Students getStudents() { return students; }

    public Courses getCourses() { return courses; }

    public String getSemester() { return  semester; }

    public StudentsEnrolment(Students students, Courses courses, String semester){
        this.students = students;
        this.courses = courses;
        this.semester = semester;
    }

    @Override
    public String toString(){
        return students.getId()+
                "," + students.getName()+
                "," + students.getId()+
                "," + courses.getCourseId()+
                "," + courses.getCourseName()+
                "," + courses.getNumberOfCredit()+
                "," + semester + "\n";

    }

}
```

StudentsEnrolment

```java
import java.io.IOException;

public interface StudentsEnrolmentManager {
      void Add(String sID);
      void Update();
      void Delete(String sID) throws IOException;
      void GetOne() throws IOException;
      void GetAll();
}
```

Interface StudentsEnrolmentManager

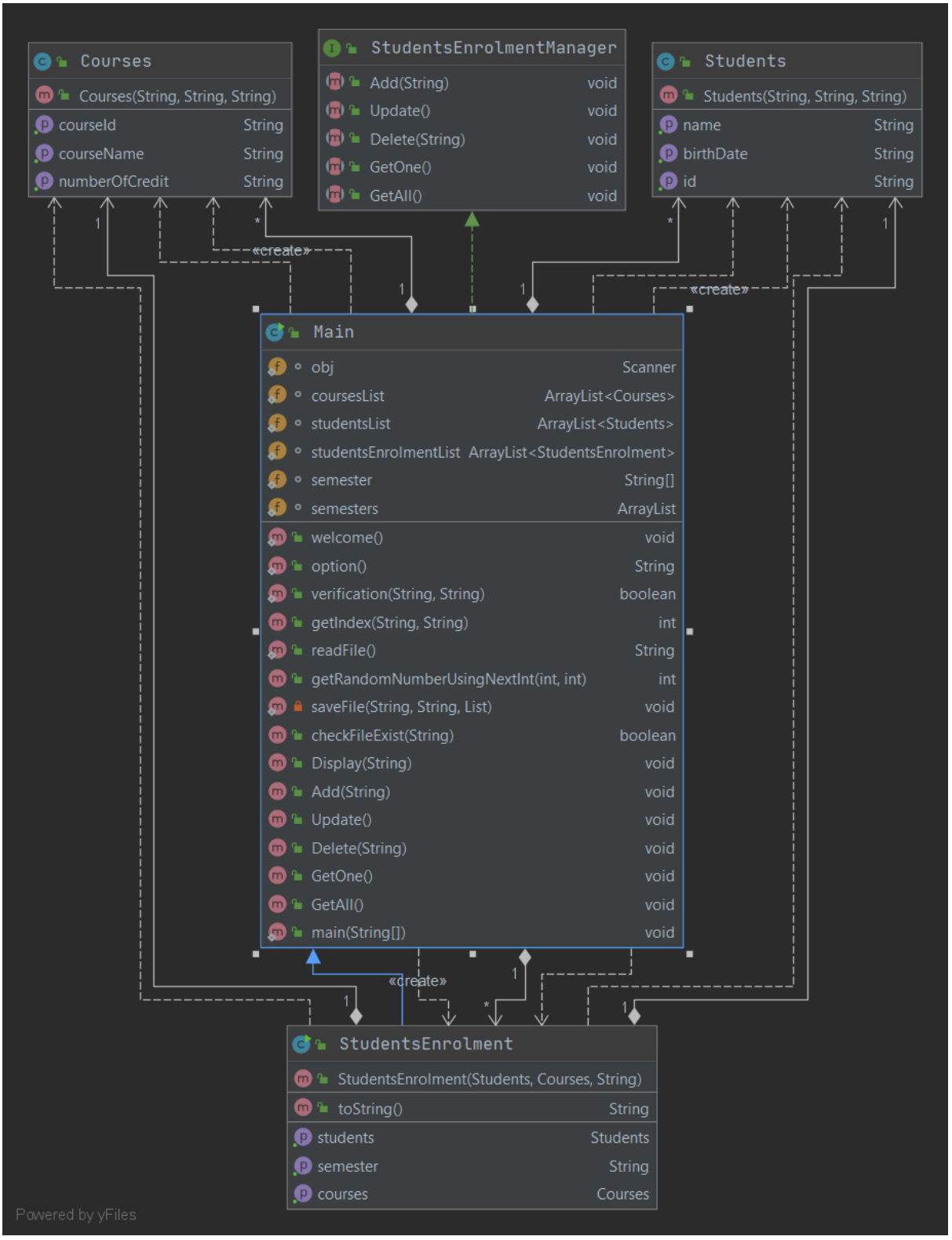
```java
public class Courses{
    private final String CourseId;
    private final String CourseName;
    private final String numberOfCredit;

    public Courses(String CourseID, String CourseName, String numberOfCredit){

        this.CourseId = CourseID;
        this.CourseName = CourseName;
        this.numberOfCredit = numberOfCredit;
    }

    public String getCourseId() { return CourseId; }

    public String getCourseName() { return CourseName; }

    public String getNumberOfCredit() { return numberOfCredit; }

}
```

Course

```java
public class Students  {
    private final String Sid;
    private final String StudentName;
    private final String birthDate;

    //Setter Getter
    public  Students(String Sid, String StudentName, String birthDate){
        this.Sid = Sid;
        this.StudentName = StudentName;
        this.birthDate = birthDate;

    }
    public String getId() { return Sid; }

    public String getName() { return  StudentName; }

    public String getBirthDate() { return birthDate; }



}
```

Student

Class Diagram:



II. Function:

Main functions:

- GetAll():
    - This is the function to print all information of enrollment

    - The data will include: Sid, student name, birthdate, course name, course id, credit, and semester.
    - Users just need to enter this function, the application will print out data.
- GetOne():
    - This is the function to print specific information:
        - Students in one specific course.
        - Courses enrolled of specific students.
        - Courses available in specific semesters.
    - The data will include:
        - Sid, student name.
        - Course ID, course name.
        - Course ID, course name.
    - After entering this function, users need to pick an option of the specific field of data they want to print out.
    - This function also allows users to save their file.
- Add():
    - This function allows users to add information to the enrollment list.
    - Users need to enter the Sid of the student that they want to add more courses.
    - The application will check if student ID and course ID is available in the list. Moreover, it will check if that student has enrolled in that course or not. If not, users will be able to add.
- Delete():
    - This function allows users to delete courses of specific students.
    - Users need to enter the Sid of the student that they want to add more courses.
    - The application will check if student ID and course ID is available in the list. Moreover, it will check if that student has enrolled in that course or not. If yes, users will be able to delete.
- Update():
    - This function give users the ability to delete or add courses of one specific student.

Side functions:

- Verification(): This function will check if a student has enrolled in that course or not. It will return false.
- GetIndex():This function will return the index of students that users want to delete.
- CheckFileExits(): This will check if the file's name is in the system or not. It will avoid the collapse of the same name.
- SaveFile():This function will save the file with name and data imputed by users.

III. GitHub

For the whole process, there are 12 commits to the git repository.
Here is the link to access git repository:
https://github.com/ThomasLuuu/StudentManagement_A1