Assignment 1: Build a console app

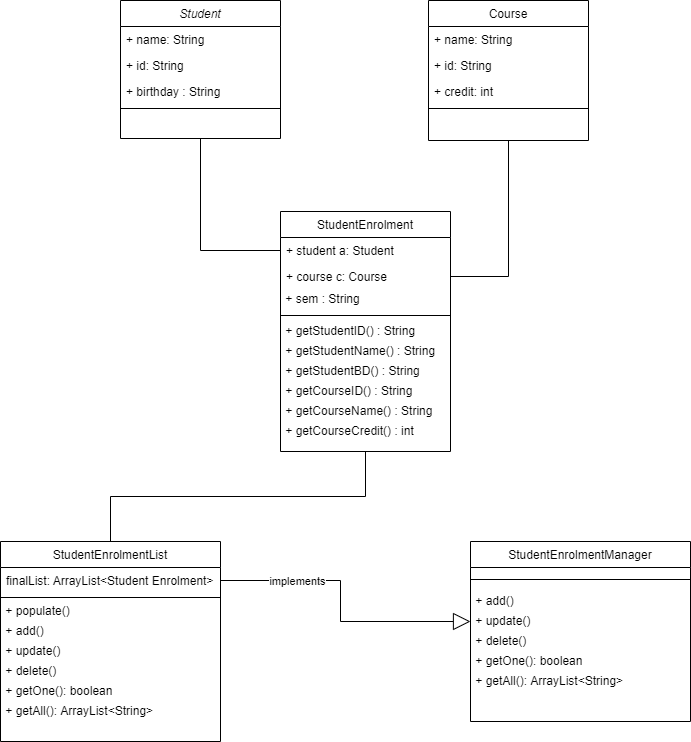
COSC2440 - Further Programming

Nguyen Hoang Vinh Hung – S3911246

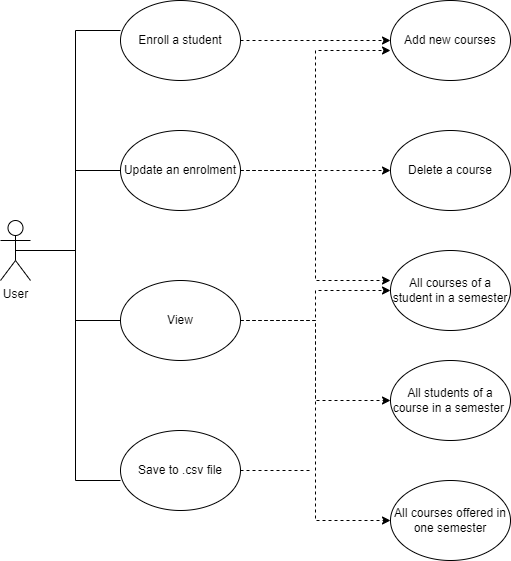
# **Report**

For the program of this assignment, I have created five classes: Main, Student, Course, StudentEnrolment and StudentEnrolmentList. The Main class is in charge of controlling the system, perform I/O functions such as read files and user’s input, as well as exporting file. The StudentEnrolment class is an association class between the Student class and the Course class. The StudentEnrolmentList class contains a list that stores all of the StudentEnrolment and implemented with the StudentEnrolmentManager interface.

**Class Diagram**

****

**Use case diagram**

****

**StudentEnrolmentList class**

This class is where the enrolment list (ArrayList<StudentEnrolment> finalist) is stored and processed. There are 6 methods created in this class:

1. Populate(StudentEnrolment se): this method would simply add the records in the inputted file into the list of Student Enrolments.
2. Add(StudentEnrolment se): this method would take user’s input and add new enrolments to the list. It would also check if student were already enrolled with the same course and throws an exception.
3. Update(Student s, Course c, String sem, String action): provided with a student, a course, semester and an action (add or delete), this method will perform the action to the record in the EnrolmentList.
4. Delete(String sid, String cid, String sem): this method will remove a record that has the same sid, cid and sem in the EnrolmentList.
5. getOne(Student s, Course c, String sem): this method will return a Boolean value true if the EnrolmentList contains a record of same student s, course c and semester and vice-versa.
6. getAll(Object o): this method returns an ArrayList of all of the records that contains the object o (Student or Course).

**Main class**

This class will act as the main system of the program with two methods:

1. Main():

First, I created Hash Maps (Map<String, Student>, Map<String, Course>) of students and courses so that a student or course will be linked with its id for better data extraction. I also declared a new StudentEnrolmentList object that contains the final enrolment list.

Then the program would let the user choose whether to use their own file or go with the default file. File Scanner is used to read each line of the file and then populate data into the student map, course map and final enrolment list.

After that, a menu of functions will be printed and user could choose options from 0 to 5. The first function would be “Enroll a student for one semester” and the system will ask user to input student id, course id and semester. Using the add() method, the program would check if the student is already enrolled with that course and add new enrolment to the list or send out an error message.

The second function is updating an enrolment of a student in one semester. User will input the student id and semester. Then the system will use the getAll() method to get all enrolments of that student and use a for loop to filter out the records that has the same semester as input. Those records would be printed and user could choose add or delete. After choosing the action and course id, the system will parse in those input to the update() method and perform the action to the enrolment list.

The third and fourth function are quite the same since it will ask user to input student id or course id and semester. Then it will use the getAll() method to get all of the records with the same id and semester and print out. After that, user can choose whether to save those records to csv file using the export() method.

The fifth function is displaying all courses offered in one semester. It will take in a semester from user input, then use a for loop to iterate through the enrolment list and extract courses in that semester. The courses would be added to a hash map first so there will be no duplicates and then converted to an array list. User will also be asked to save the records to csv file or not.

The Last function would let the user exit the program when they are done editing and export the enrolment list to a csv file called “FinalStudentEnrolment.csv” using export() method.

1. Export():

This method will use the OpenCSV library and write the desired data into a csv file.