COSC2440 Software Architecture Design and Implementation

Assignment 1: build a console app

Code Review Report

Name: Han sang yeob

Student number : S3821179

Date: 4/10/2021

1. Overview

This report is a review of the code which is for the assignment in COSC2440 : build a console app. The application is built for the admin who needs to enroll the student in the school with the specific course and semester. For functional requirements, it will have the list of students and the course which is already embedded in the list. For the admin(or user), they are possible to enroll the student with a specific course and semester which will be saved in .csv format. Furthermore, there is also a test using Junit to ensure that this application will work with the other data for further usage.

2. Use case diagram

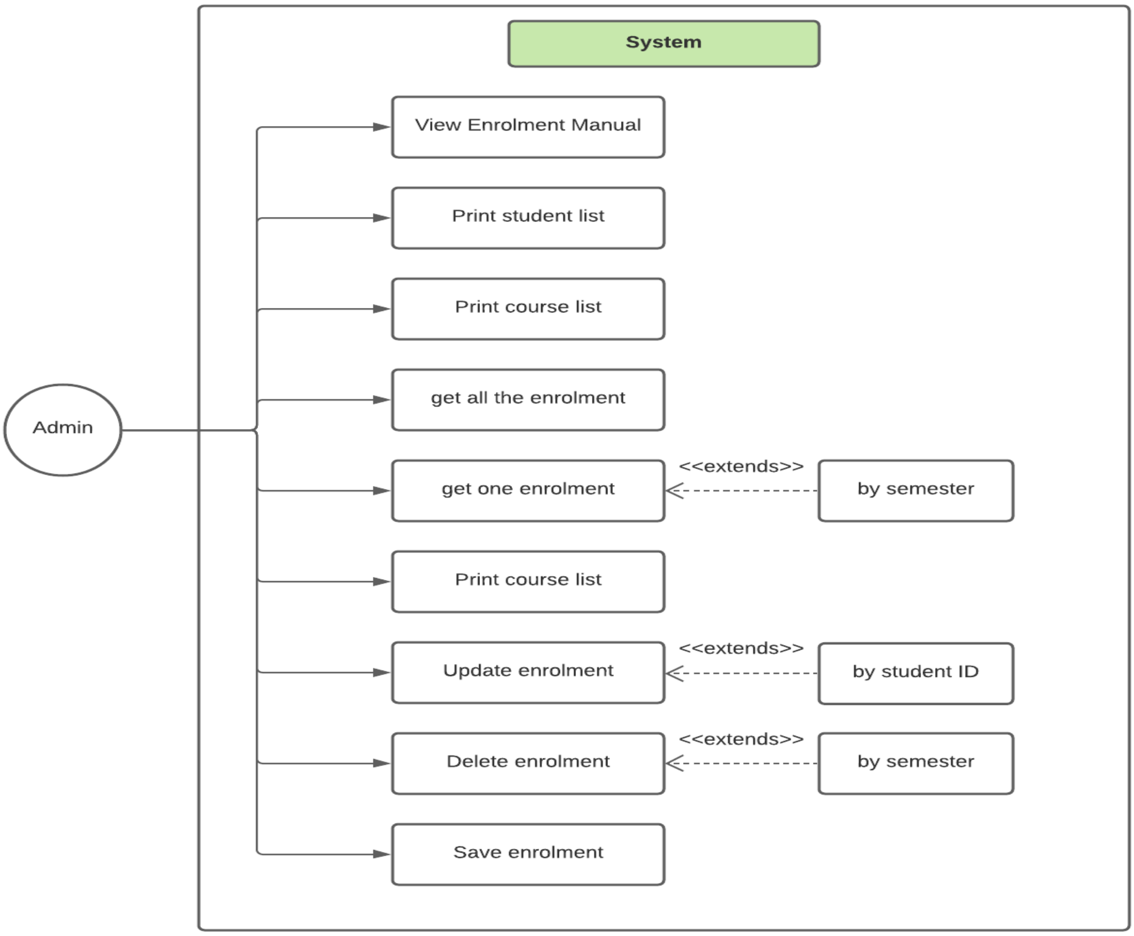

Figure 1. *Use case diagram*

For the use case, it will have only one actor, admin. The admin will be able to control the system with the console with a variable function. First of all, when the system starts, the system will print out the manual for the user pressing 0 – 8 number with the current student and course list which they can allocate. The system is able to print the student and course list again for enhancing the usage during enrolment. For the get all the enrolment, it will print the current enrolment state with all the information including which semester, course, and student ID. However, the get one enrolment, the user needs to type the semester “2021A” or “2021B” to see the enrolment at the state. Also, the CRUD function is coded to help the maintenance of the system. The user is able to update the student enrolment by typing their student ID and delete the enrollment information by using the semester. Lastly, when all the process is done the user can save the current enrolment list to .csv format.

3. Method

1. `saveEnrolmentToFile()`

This method used 'FileWiter' to get the data from the current StudentEnrolment list and save it to CSV format inside the Try–catch exception function. For enhancing the readability, the list of enrolment will be separate by using '/' differentiating 'semester', 'course name' and 'student id' as well as sending the next data to other lines using lineseparator(). The IOException will catch the possible error and display it in the console. For example, if the user tries to save the data while the CSV file is opened on the window bar, the error will occur.

2. `enrollStudent()`

This method used Scanner function to get the input from the user. This function will ask the student id, name of the course, and semester to get the enrolment information. Once the user typed the exciting information in the current list, the information will be saved in the studentEnrolment list. If not, the system will print out a null value whenever the user tries to get the data from the list.

3. printStudents()

This method will print out what is inside the student list. The 3 student information is already existing in the list and the user cannot add or delete this information.

4. `printCourses()`

This method will print out what is existing in the course list. The 3 course information is already existing in the list and the user cannot add or delete this information.

5. `getOne()`

This method used Scanner function to get the data from user input. Once this method is initialized, it will ask the user which semester that user wants to get the data. The user needs to type either '2021A' or '2021B'. once the user typed the valued input, the system will print out every enrolment list from the data including the student id and name of the course.

6. `getAll()`

This method will get all the information from the current student enrolment list. However, if the list is empty and does not possess the information to show it to the user, it will recommend enrolling the new information.

7. `enterStudentIdPrompt()`

This method is for `updateStudent(String studentId).` If the user tuped correct student id, it will return the Strint value to `updateStduent()` for updating the enrolment.

8. `updateStudent(String studentId)`

This method will get the boolean data from the `enterStudentIdPrompt().`

Once the valued data is sent, it will pass the data to `updateInfo()`. If the input data has been updated successfully, it will show the user message.

9. `getStudentById(String studentId)`

This method is used in `updateInfo(String studentId)`, for it's method to get the valued inforamtion from the user.

10. `updateInfo(String studentId)`

This method is to update enrolment information. Once this method is initialized, it will print printStudentUpdateManual() for suggesting guidance to the user. In this method, the user can update the course name or semester by using the student ID. Once updating is finished, the user needs to press '0' to end this method and update.

11. `printStudentUpdateManual()`

This method is called from updateInfo(). It contains the information to give guidance to the user who is using updateInfo(String studentId) method.

3. Class diagram

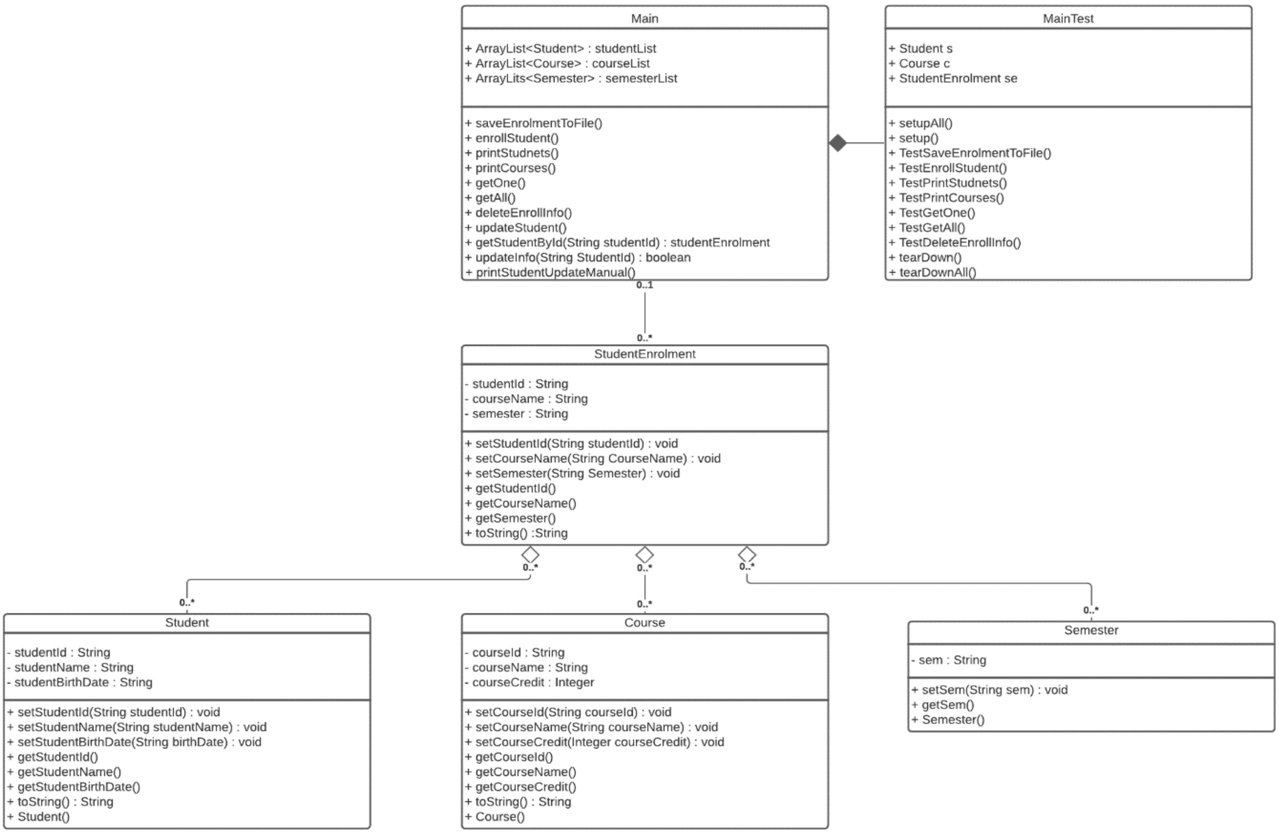

Figure 2. *Class diagram*