|  |  |
| --- | --- |
|  | ASSESSMENT 1: Build a console app |
|  |  |
|  | Nguyen Hoang Linh – s3880313  Instructor: Minh Vu Thanh  COSC2440  RMIT University Vietnam, Semester 2021A |

—

Table of Contents

[Class diagram](#_Toc68271677) 2

[report](#_Toc68271677) 3

—

# Class diagram

Graphical user interface

Description automatically generated

—

# Main menu

# Text Description automatically generated

—

# Report

This report will describe my Student Enrolment System which has been illustrated in the class diagram and menu picture above. The system has three classes which are Student, Course, Student Enrolment, and an interface Student Enrolment Manager. The first two classes Student and Course contain Student ID, Student name, birth date and Course ID, course name, number of credits, respectively. These two classes hold the information of a student and a course themselves. Student Enrolment is an associate class. It has three main properties which are Student, Course and Semester. Moreover, Student Enrolment has an array list to keep track of all the enrollments. The interface Student Enrolment Manager is responsible for taking care of that list with five methods: add, update, delete, get One, and get All. The main function is called a Student Enrolment System. This main function should present a menu with nine options for an academic assistant or user to select as in the picture shown above. Before running the program, the system should ask the user to input their CSV file data, else it will run the default.csv file to populate enrolments. The enrolments will be stored into enrolment List array list. For the Student and Course information, it will be collected into the student List and course List which are the list I created to help holding lists of students and courses for the purpose of displaying. Now, the main menu should run, and we will go through each option to get a deeper understanding of how each feature works.

1. **Create a student information.**

* The system will ask the user to enter student information which include Student ID, student name, and student’s birthdate.
* After getting user input, the system will create a student object with that three information and add it into student List inside the main function.
* The system will print out “Created student successfully!” after creating a new student and will pop back into the main menu.

1. **Create a course information.**

* The system will ask the user to enter course information which include Course ID, course name, and number of credits.
* After getting user input, the system will do exactly the following process like creating a student information but refers to the Course information.
* The system will print out “Created course successfully!” after creating a new student and will pop back into the main menu.

1. **Enroll a student for 1 semester.**

* Firstly, the system will print out the student List for users to pick for the enrollment. After printing out the list, the system will ask for a student ID in the list that users want to enroll. If the student that they want to enroll is not in the list, users might enter “QUIT” to go back to the menu and choose option 1 to create a student first. The same happens with the system print out course List and then asking for a course ID. If the user input correct student ID and course ID, the system will ask for the semester that they might want to enroll that student.
* When the system has got the student ID, course ID, and semester from the user input, it will trace back to get the student object and course object that includes those ID. After that, it will create a new Student Enrollment object and use the method “Add” to add the new enrollment into an array list inside the Student Enrollment class.
* The system will print “Enrolled successfully!” when the actions are done and return to the main menu for continued action.

1. **Print all courses of a student in a semester. Update or add or delete courses from that list.**

* The system will ask the student ID and semester from the user input. After that, it will check in the enrollment List to see if an enrollment has that student ID and semester, the course will add to a new course list. The system will print out that course list as all courses of a student in a semester.
* The system will ask users to add, update or delete courses from that list. If a user wants to add a new course, the system will take user input of new course information then create a new course and add it to the list. The same with update and delete. After adjusting the list, the system will print out successfully and return to the main menu for further action.

1. **Print all courses for 1 student in 1 semester.**

* The system will ask the student ID and semester the same as option 4 and print out all courses of a student and semester from user input student tID and semester. This feature will allow users to save output as a CSV file.

1. **Print all students of 1 course in 1 semester.**

* The system will ask for the course ID and semester. After that, it will check the enrollment list to get the student information who has enrolled in that course and in that semester. The system will display the list of students and allow users to save as a CSV file.

1. **Print all courses offered in 1 semester.**

* The system will ask for the semester. After taking the semester that users want, it will check in the enrollment list those courses in that semester and add them to a list. There will be a remove duplicated function to remove duplicated courses. All courses offered in that semester will be printed and will be able to be saved as a CSV file.

1. **Get information of one enrollment.**

* The system will ask for the student ID and course ID that the user wants to get the enrollment. The system will display the information of that enrollment using get One function in the Student Enrollment class. This function is also available to save output as a CSV file.

1. **Display all enrollments.**

* This option will display all the enrollments using get All function in the Student Enrollment class. The user can save the output as a CSV file for this feature as well.

Else, the system will print out “Goodbye! Thank you for using the system” as a user finishes their tasks. If the user input is not valid as in the menu options, the system will basically print out “Invalid input” and display the menu again. That is all about my student enrollment system.