Shape

Description automatically generated with medium confidence

Further programming

Assignment 1 – Build console app

Course: Further Programming

Course code: COSC2440

Student name: Bui Quang An

Student ID: S3877482

Lecturers: Mr. Vu Thanh Minh

**Table of Contents**

1. Overview ……………………………………………………………………………. 2
2. Flow description ……………………………………………………………………. 2
3. Table of methods …………………………………………………………………… 4
4. Diagrams……………………………………………………………………………. 7
5. **Overview**

The Student Enrolment Management Application is the system which helps users manage the information of courses, students, and the enrolments developed by using Java. There are 3 parts which are “Manage courses”, “Manage enrollments” and “Manage students”. Inside these parts will have other various functions helping users manage information conveniently.

1. **Flow description**

When users run the program, they will see the introduction which is represented the application and the main menu on the below. There will be four choices for users including “Manage courses”, “Manage enrollments”, “Manage students”, and “Quit” to stop the program.

Text

Description automatically generated

*Figure 1: The main menu*

To begin with, when users choose “Manage courses”, they will see another menu which has four other choices: “View courses”, “View courses in one semester”, “View courses student learn in one semester” and “Back”. If users choose view courses, they will observe the list of all courses had in the database. On the other hand, if they want to see available courses in one specific semester, they can enter number 2 and then type the semester they want to know. Furthermore, if they enter number 3, type the student ID, and type the semester, the program will demonstrate all courses that student ID learn in that semester. Also, when users decide to see list of courses in one semester or list of all courses that student learns in one semester, they can save the list and write it on new CSV file added on the directory called “data” when they stop the program. Finally, if users enter number 4, the program will go back to the main menu.

Text

Description automatically generated

*Figure 2: Course menu*

Secondly, when users choose “Manage enrollments”, once again, they will see the enrollments’ menu which has four other choices: “View all enrollments”, “View all enrollments in one semester”, “View one enrollment in one semester” and “Back”. If users choose “View all enrollments”, they can view the list of all enrollments had in the system. After that, the program will ask users to decide if they want to update the enrolment. If they enter “y”, the program will ask users to add or drop enrollments and by entering the student ID, course ID and semester, they can add or drop the enrollment in the system. On the other hand, if they choose number 2 and type the semester, the program will list all enrollments in that semester. Moreover, the program will illustrate the enrolment if users choose number 3 and type the student ID, course ID, and semester. Finally, the program will go back to the main menu if users enter number 4.

Text

Description automatically generated

*Figure 3: Enrollment menu*

Thirdly, choosing the “Manage students” will show the student’s menu with three options: “View all students”, “View all students in one course” and “Back”. As the previous two, “View all students” will demonstrate list of the students who have enrolled in the system. If users choose number 2 and type the course ID with the semester, users will see all students who enrolled that course in that semester. After that, the program will ask users to save the report and if users choose “y”, the program will save and write it in the new CSV file added on the directory called “data” when they stop the program. Finally, entering number 3 and users can go back to the main menu.

Text

Description automatically generated

*Figure 4: Student menu*

1. **Table of methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method name** | **Return type** | **Params** | **Description** |
| addEnrolment | boolean | studentID: String, courseID: String, semester: String | Add new enrolment based on studentID, courseID, and semester |
| deleteEnrolment | boolean | studentID: String, courseID: String, semester: String | Delete enrolment based on studentID, courseID, and semester |
| getOneEnrolment | StudentEnrolment | studentID: String, courseID: String, semester: String | Get one enrolment in the system based on studentID, courseID and semester |
| getAllEnrolments | ArrayList<StudentEnrolmen> |  | Return all enrolments in the system |
| getAllCourses | ArrayList<Course> |  | Return all courses in the system |
| getAllStudents | ArrayList<Student> |  | Return all students in the system |
| getStudentByID | Student | studentID: String | Return student in the system based on student ID |
| getCourseByID | Course | courseID: String | Return course in the system based on course ID |
| getSemestersInOneCourse | ArrayList<String> | courseID: String | Get list of semesters that course ID is available |
| populateStudents | void |  | Process the data in csv file to get students information |
| populateCourses | void |  | Process the data in csv file to get courses information |
| populateEnrolments | void |  | Process the data in csv file to get enrolments information |
| populateData | void |  | Call the populateStudents, populateCourses and populateEnrolments methods when users run the program |

1. **Diagrams**
2. Use case diagram:Diagram, schematic

   Description automatically generated
3. Class diagram

Diagram, schematic

Description automatically generated

Link to the github repository: <https://github.com/quangan186/COSC2440-A1>