Student number: s3751881

Student name: Tran Dang Bao Nhi.

**Student Management System technical documentation**

1. **Introduction:**

The Student Enrollment Management system (acronyms SEM) is designed to support academic assistant to manage enrollments system of a school. The interface of this SEM is built with console application. This report is technical documentation of the system which will explain further design assumptions, patterns and tell in detail how the system operates.

1. **Assumption:**

There are some assumptions during the design process:

1. If user have specific file of enrollment to upload, their file type must in csv type in order to run the system, other file type is not supported.
2. When add an enrollment, user only can choose from available student in the file which will be print out as menu option with index. When user wants to select a student, they can input the index represents that student. After the valid student index is received, the system will print all available courses and the user continues to choose the course by inputting the course index. After the course, user enters the semester in type of string with the example printing on the screen for the semester format. Each of the index will be checked and validated, in any step if one of the indices inputted from user is wrong the program will asks them to input again.
3. If the user enters any invalid option that is out of the menu range from [0-8] which is displayed in the console interface, the program will print “Invalid option. Please try again ”the string “Press any key to continue.” and if user presses key like enter or space or any key the program will print option and let user inputs their option again. It will keep user in a loop of enter option. Whenever user chooses option number 8, the application will be shut down.
4. Any action will be one way, user cannot undo or redo the action. They must be careful when choosing one option to execute.
5. **Program flow:**
6. When the user clicks run the main class, there will be a question prompts and asks whether user wants to import a file or not. If the user any other key than “y” in the instruction, the program will automatically load the default file and work with it.
7. In the menu the program will show all essential functions for user to choose, user can have option of their choice by input the number of their choice in the console next to a question.
8. They can use the add, update, delete features by choosing the index shown in the screen. After the function receive an index the validate function will be called to check the input, if user choose any object that doesn’t exist, there will be a message shows the specific error for user and user are asked to try again. Or if they want to get out of the function, they can press any key.
9. There are options that utilizes academic assistant to have insights about enrollments or students or courses if they need to. There will be function to print all available courses in a semester and it will call the GetCourse() method in Academic Assistant class. In addition, base on the scope there is function to display all students of specific course in a semester which call the GetStudents() method. Lastly, there is functions to print all student enrollments in a specific semester, which will get course of specific student’s index and semester input.
10. When there is any error, the program will inform user about that specific error. User will be asked for further action with confirmation - “y” stands for yes in msot of cases.
11. **Class diagram:**

The class diagram serves as the designer’s interpretation and some decisions made during the design stage of how the system should be constructed, how classes interact with each other and what attributes and functionalities should the classes have. There are 9 classes in the system which serve different purposes and present distinguishable objects in the application.

First, the class “Helper” is designed without any attribute, the main purpose of this class is only supporting the system file management with 2 methods:

1. ReadFile(param: String fileName) from specific directory to use for passing data into other method for further operations.
2. WriteFile(String fileName, String buffer) to write file for import command.

Student class has 3 attributes to set identification for student. With set of many getters and setters to take out the property for further usage.

Course class is a design of course object which set identification for a course. With set of many getters and setters to take out the property for further usage.

StudentEnrollment class is composed of Student and Course classes that take objects from those classes to construct new object which is Student Enrollment object with additional string semester.

Then, there is an interface to act as manager for those Student Enrollment, the “StudentEnrollmentManager” specifies what class implements it must do and “StudentEnrollmentList” implements this interface class to execute user’s request. This class has defined 5 methods which reference data types that are add, update, delete, get one and get all. The main will use this class as an interface to communicate with Student Enrollment List class.

Next, the “Student Enrollment List” (SEL) implements the Student Enrollment Manager interface and do specific operations that it must do like add enrollment, update enrollment, delete enrollment from enrollments list. The SEL class has 1 attribute which is student enrollments list as it mainly deals with student enrollment object.

Additionally, the Academic Assistant class is created to mainly support academic assistant from the school with methods that execute operations such as add, update, read, delete specified enrollment which the user wants to interpret or add to the enrollment management system, load enrollment from file or run command to export particular list. The class have 3 attributes which are students – List<> type, courses – List<> type and an interface instance Student Enrollments Manager to communicate with the SEL class. There are methods that serve the scope functional requirements such as getAll() to print all available enrollments in the system, method to get all students in one course of specific semester GetStudent(int course index, semester), method to get all courses that are available n semester getCourses(String semester). There is a special method acts as filter - filterEnrollment() that is implemented in above methods which take out list of necessary enrollments that match with the user’s query.

Finally, the main class is where all of the necessary methods and classes come together to operate the system. The menu option and essential results will be printed in here, this acts as the console app’s interface between user and system.

Diagram

Description automatically generated

1. **Use Case diagram:**

Diagram

Description automatically generated

To identify and organize the system requirements, the use case diagram is shown to represent all the possible system’s features and relationship between actors and their actions based on the stakeholder’s demand.

Mainly, the academic assistant or administrator will be those who using the system most of the time. They have core use cases such as view, add, update, delete enrolment. They can upload file with populated enrolment to do further actions. After query Course or Student or Enrollment they want, they need to export the csv file with the results.