

Trường Đại học Khoa Học Tự Nhiên, ĐHQG – HCM

Khoa Công nghệ Thông tin



## PROJECT REPORT

### Programming Techniques

**Teacher**

: Đinh Bá Tiến  
Trương Phước Lộc  
Đỗ Nguyên Kha  
Lê Thanh Tùng

**Class**

: 21CLC09

**Group**

: 7

**Members**

: 21127604 – Nguyễn Lâm Hải  
21127347 – Đặng Hoàng Long  
21127661 – Fa Ngọc Uyển Nhi  
21127697 – Lê Thị Minh Thư

Trường Đại học Khoa Học Tự Nhiên, ĐHQG – HCM

Khoa Công nghệ Thông tin



## PROJECT REPORT

### Programming Techniques

<b>Teacher</b>	: Đinh Bá Tiến Trương Phước Lộc Đỗ Nguyên Kha Lê Thanh Tùng
<b>Class</b>	: 21CLC09
<b>Group</b>	: 7
<b>Members</b>	: 21127604 – Nguyễn Lâm Hải 21127347 – Đặng Hoàng Long 21127661 – Fa Ngọc Uyển Nhi 21127697 – Lê Thị Minh Thư

## **ACKNOWLEDGEMENTS**

First and foremost, we would like to give our utmost appreciation to our teachers, which includes Dr. Dinh Ba Tien, Mr. Truong Phuoc Loc , MS. Do Nguyen Kha, Dr. Le Thanh Tung whom we have learnt a lot from. They have provided us with invaluable advice and guidance. They have helped tremendously to the successful completion of the project.

Besides, we would like to thank all the teachers on the previous course of “Introduction to Programming”, the basic knowledge of coding and algorithms we have learnt from them have made the learning process of this course much easier for all of us.

At last but not in least, we would like to thank everyone who helped and motivated us to work on this project.

## LIST OF CONTENTS

I. PREFACE .....	5
II. CONTENTS .....	6
II.1. IDEA .....	6
II.1.1. Project deployment.....	6
II.1.2. System design.....	8
II.1.3. Implementation.....	9
II.1.4. Distribution of the tasks .....	10
III. WORKING PROCESS .....	11
III.1. General .....	11
III.1.1. Login .....	11
III.1.2. Change password.....	11
III.1.3. Logout .....	11
III.2. Staff.....	12
III.2.1. Create school year .....	12
III.2.2. Create semester .....	12
III.2.3. Create classes .....	12
III.3. Student .....	13
III.3.1. Enroll in courses.....	13
III.3.2. Remove Enrolled Courses.....	15
III.3.3. View list of courses .....	15
III.3.4. View the scoreboard.....	15
III.4. Delete function .....	16
III.5. Store information.....	17

III.6.	Read information .....	17
IV.	Courses Registration System Introduction .....	18
IV.1.	Function tree .....	18
IV.2.	General.....	18
IV.2.1.	Change his/her password.....	18
IV.2.2.	View his/her profile.....	18
IV.3.	Staff.....	18
IV.3.1.	Create a new school year.....	18
IV.3.2.	Create a new semester .....	19
IV.3.3.	Create new classes for 1 <sup>st</sup> year students .....	19
IV.3.4.	View the list of classes .....	19
IV.3.5.	Add students to a class .....	20
IV.3.6.	View the list of students in a class .....	20
IV.3.7.	Create the list of courses .....	20
IV.3.8.	Update the information of courses .....	21
IV.3.9.	Remove a course from the list of courses .....	21
IV.3.10.	View the list of courses.....	22
IV.3.11.	View the list of students in a course .....	22
IV.3.12.	Import Scoreboard .....	23
IV.3.13.	Update Scoreboard.....	23
IV.4.	Student .....	23
IV.4.1.	Enroll in courses.....	23
IV.4.2.	Remove a course that was enrolled in.....	24
IV.4.3.	View the list of courses .....	24
IV.4.4.	View their scoreboard .....	25

V. Work evaluating .....	26
V.1. Percentage rating.....	26
V.2. Members rate others and themselves .....	26
V.2.1. 21127697 - Lê Thị Minh Thư .....	26
V.2.2. 21127347 - Đặng Hoàng Long.....	27
V.2.3. 21127661 - Fa Ngọc Uyển Nhi .....	27
V.2.4. 21127604 - Nguyễn Lâm Hải.....	28
VI.Reference source .....	29

## LIST OF PICTURES

Figure 1. Idea Map.....	8
Figure 2. Function tree.....	18
Figure 3. List of students .....	20
Figure 4 - View the list of courses.....	22
Figure 5 - View the list of students who enroll in the course.....	22
Figure 6 - Scoreboard CSV file .....	23
Figure 7 - Update scoreboard .....	23
Figure 8 - Enroll in courses .....	24
Figure 9 - View the list of courses in which were enrolled.....	25
Figure 10 - View scoreboard .....	25

## LIST OF TABLES

Table 1. Distribution of the tasks .....	11
Table 2. The list of classes.....	20
Table 3. The list of students in a class.....	21
Table 4. The list of courses.....	21
Table 5. The list of courses.....	22
Table 6. The list of courses.....	24
Table 7. The list of courses.....	24
Table 8. Percentage rating .....	26
Table 9. Le Thi Minh Thu's evaluation board .....	26
Table 10. Dang Hoang Long's evaluation board .....	27
Table 11. Fa Ngoc Uyen Nhi's evaluation board.....	28
Table 12. Nguyen Lam Hai's evaluation board .....	28

## **I. PREFACE**

This Project Report has been written as a mandatory requirement of the final project of the CS162 – Programming Techniques course in the academic year 2021-2022.

As a part of the CS162 – Programming Techniques course, we are required to do a final project task. The final project is about implementing our coding and algorithmic knowledge to create a course registration system.

This Report will cover many aspects of our work progress throughout the whole timeline, which includes: our idea and approach to the given task, weekly reports of our work, functionalities of our product and individual assessment at the end of the project.

The information presented in this Project Report is our own knowledge gained during the time we worked on the project. Moreover, our assessments are entirely objective and not forced in any way.

Group members

21127347	-	Đặng Hoàng Long
21127604	-	Nguyễn Lâm Hải
21127661	-	Fa Ngọc Uyển Nhi
21127697	-	Lê Thị Minh Thư



## II. CONTENTS

### II.1. IDEA

#### II.1.1. Project deployment

##### II.1.1.1. General:

- There are 2 types of users in the system:
  - + Academic staff members.
  - + Students
- The users need to log in to the system to:
  - + View their profile
  - + Change password
  - + Do their own task
- After using, they log out of the system.

##### II.1.1.2. Academic staff:

- At the beginning of the year:
  - + Create a school year.
  - + Create several classes for 1st year students.
  - + Add new 1st year students to 1st-year classes (*import a CSV file containing all students in a specific class to the system, instead of adding one by one: No, Student ID, First name, Last name, Gender, Date of Birth, Social ID.*)
  - + Create a semester: 1, 2, or 3, school year, start date, end date. Choose the school year that the newly created semester belongs to. The created semester will be the current / the default semester for all the below actions.
  - + Create a course registration session: start date, end date.
  - + Add a course to this semester: course id, course name, teacher name, number of credits, the maximum number of students in the course (default 50), day of the week, and the session that the course will be performed (MON / TUE / WED / THU / FRI / SAT, S1 (07:30), S2 (09:30), S3(13:30) and S4 (15:30)). A course will be taught in 2 sessions in a week.
  - + View the list of courses.

- + Update course information.
- + Delete a course.
- At the end of the semester:
  - + Export a list of students in a course to a CSV file
  - + Import the scoreboard of a course. A scoreboard will have at least the following columns: No, Student ID, Student Full Name, Total Mark, Final Mark, Midterm Mark, Other Mark. (An academic staff member will export the list of students in a course, send the CSV file to the teacher, the teacher will enter student results in this file, send it back to the staff, and then the staff will import the scoreboard to the system)
  - + View the scoreboard of a course.
  - + Update a student's result.
  - + View the scoreboard of a class, including final marks of all courses in the semester, GPA in this semester, and the overall GPA.

#### **II.1.1.3. Students**

- Enroll in a course. If 2 sessions of the new course are conflicted with existing enrolled course sessions, students can not enroll in it. They can enroll in at most 5 courses in a semester.
- View a list of enrolled courses.
- Remove a course from the enrolled list.
- View their list of courses.
- At any time, an academic staff member can:
- View their scoreboards.

## II.1.2. System design

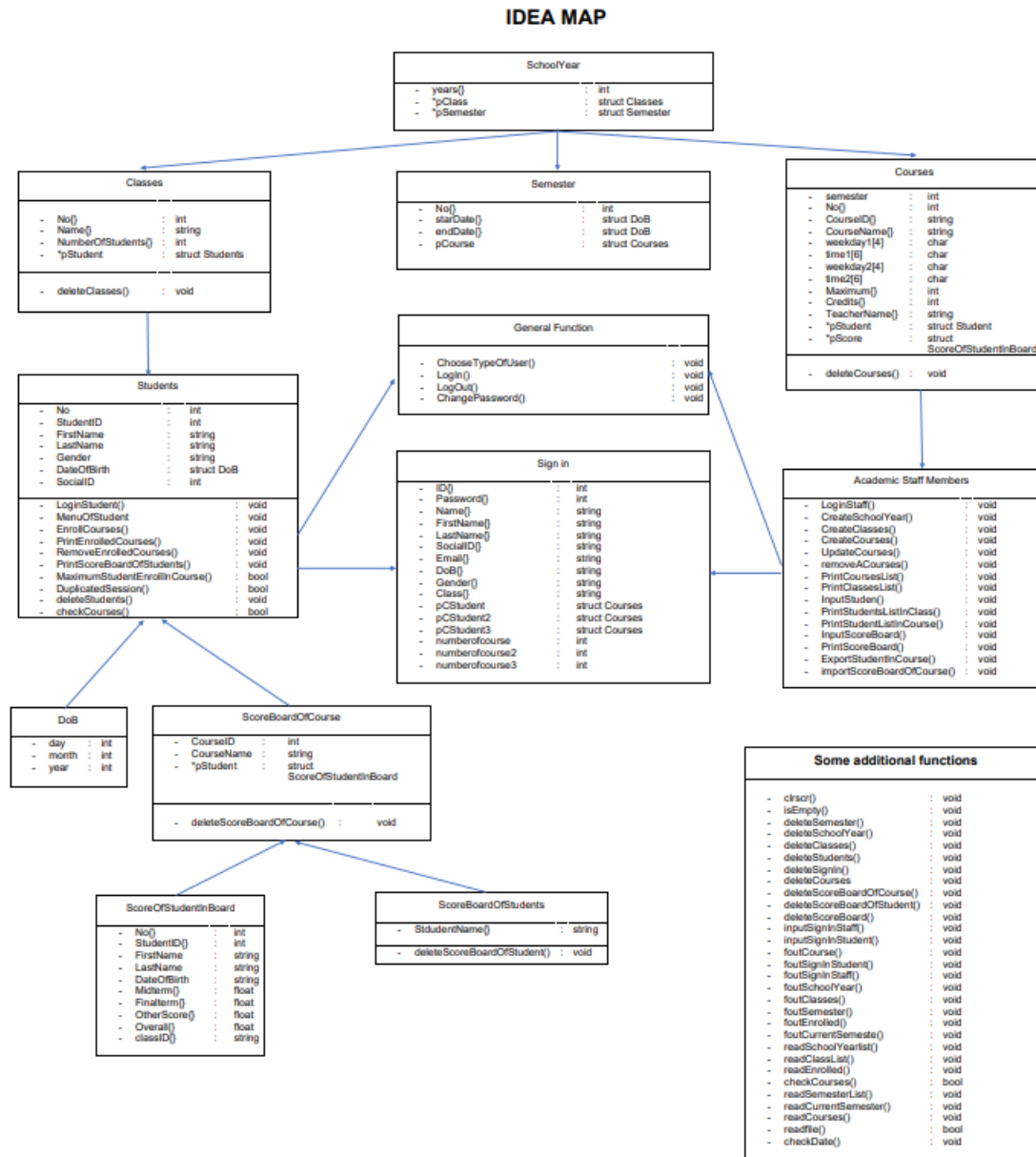


Figure 1. Idea Map

Link Google Drive: <https://drive.google.com/drive/folders/1ZrFJwUC-pABHfN198ycHw44LnVCABNQN?usp=sharing>

### **II.1.3. Implementation**

#### **II.1.3.1. General function**

- ChooseTypeOfUser(int& type): the users need to choose “type of user” to log in (For example: you are a student, you need to choose “student” for the “type of user”).
- Login(SignIn\*& pHead): the users need to log in to use.
- ChangePassword(string& Password): the user can change the password of the account.
- Logout(): the users log out of the system.

#### **II.1.3.2. Academic staff function**

- CreateSchoolYear(int& Year): the staff create a new school year.
- CreateSemester(int& Semester): the staff create a new semester.
- CreateClasses(Classes\* pHead): the staff create new classes for the 1<sup>st</sup> year students.
- CreateCourses(Courses\*& pHead, Courses \*&pCurr): the staff create a courses list for the student to enroll.
- UpdateCourses(Courses\*& pHead): the staff can update the information of the courses list.
- PrintCoursesList(Courses\* pHead): the staff can view the list of courses.
- PrintClassesList(Classes\* pHead): the staff can view the list of classes.
- InputStudent(Students \*& pHead, ifstream &studentInput): the staff add students and their information to the class.
- PrintStudentsListInClass(Students\* pHead): the staff view the list of the students in a class.
- PrintStudentListInCourse(Students\* pHead, Courses\*pH): the staff view the list of the students who enrolled in the course.
- InputScoreBoard(ScoreBoardOfCourse\*& pHead): the staff create the scoreboard of the students of the course.
- PrintScoreBoard(ScoreBoardOfCourse\* pHead): the staff view the scoreboard.

### II.1.3.3. Students function

- EnrollCourses(Courses\*& pHead, Courses\*& pStudents): students enroll in courses.
- PrintEnrolledCourses(Courses\* pHead): students view the list of courses that they enrolled in.
- RemoveEnrolledCourses(Courses\*& pHead): students can remove a course that was enrolled before.
- PrintScoreBoardOfStudents(ScoreBoardOfStudent\*&pHead): students can view their scoreboard.

### II.1.4. Distribution of the tasks

Student ID	21127347	21127604	21127661	21127697
Full Name	Đặng Hoàng Long	Nguyễn Lâm Hải	Fa Ngọc Uyển Nhi	Lê Thị Minh Thư
Tasks	Task 12	Task 7	Task 18	Task 13
	Log Out	Task 8	Task 20	Task 14
	Change Password	Task 9	Task 25	Task 15
	Task 1	Task 10	Task 26	Task 16
	Task 6	Task 11	Cover page of the report	Task 24
	Task 2	Task 17	Format the report	Tasks distribution
	Task 3	Task 19	Idea map	Project Deployment

	Task 21	Task 22		CSV files
	Idea for each function in the report	Main function		Idea for each function in the report
	Acknowledgements	Idea for each function in the report		Course Registration System Introduction
	Preface	Record User manual video		Edit User manual video

*Table 1. Distribution of the tasks*

### III. WORKING PROCESS

#### III.1. General

##### III.1.1. Login

- Function: void LogIn(SignIn\* pHead)
- Idea:
  - + Use the credentials from a struct called SignIn, go through the whole SignIn linked list until there's a correct ID - password combination, else the login process will restart again.

##### III.1.2. Change password

- Function: ChangePassword(string& Password)
- Idea:
  - + Prompt the user to input the new password.
  - + Change to old password to the new password using assign.

##### III.1.3. Logout

- Function: void LogOut(SignIn \*pStaff, SignIn\* pStudent, SchoolYear \*&pHead, SchoolYear \*&pCurr)
- Idea:
  - + Clear screen.

- + Go back to the General Menu.

#### **III.1.4. General menu**

- Function: void GeneralMenu(SignIn \*pStaff, SignIn\* pStudent, SchoolYear \*&pHead, SchoolYear \*&pCurr).
- Idea:
  - + User will input the current date and time.
  - + User then will choose the type of current user (Staff, student) to access the system.
  - + If enter wrong username or password, user must enter again or exit the system.

### **III.2. Staff**

#### **III.2.1. Create school year**

- Function: void CreateSchoolYear(SchoolYear \*&pHead, SchoolYear \*&pCurr)
- Idea:
  - + Initialize a new schoolyear pointer in SchoolYear struct.
  - + Prompt the user to input new school year (example: 2020-2021).
  - + Then add classes to that school year, also by the user's input.
  - + Finally add students to each class with CSV file.

#### **III.2.2. Create semester**

- Function: void CreateSemester(SchoolYear \*pHead).
- Idea:
  - + Initialize a new semester pointer in SchoolYear struct.
  - + Input semester number, semester start and end date.
  - + Then create courses in each semester.

#### **III.2.3. Create classes**

- Function: void CreateClasses(Classes\*& pHead, Classes\*& pCurr, int &i).

- Idea:
  - + This function is already included in create school year function.
  - + Initialize a new class pointer in the Classes struct.
  - + Input number of students and input students.

#### **III.2.4. Input student**

- Function: void InputStudent(Students \*& pHead, ifstream &studentInput).
- Idea:
  - + Access the csv file of “classname” + “.csv” at specified school-year folder.
  - + For each student allocate a Students node.
  - + Run until reach eof(studentInput).

#### **III.2.5. Menu of staff**

- Function: void MenuOfStaff(SignIn \*pStaff, SignIn\* pStudent, SchoolYear \*&pHead, SchoolYear \*&pCurr).
- Idea:
  - + Have all functions of staff.
  - + Change password
  - + View profile info
  - + Create school-year, classes, semester and courses
  - + Access Scoreboard
  - + Log out.
  - + Using switch case and label. For each case (except for log out) users will move to another menu(such as create school-year, semester, v.v) or back to the menu of staff.

### **III.3. Student**

#### **III.3.1. Enroll in courses**

- Function: void EnrollCourses(Courses\*& pHead, Courses\*& pStudents)
- Idea:



- + Print out the list of courses that staff created before for students to know the course, time and day to study.
- + Ask students the number of courses that they want to register for.
- + The students are required to type in the cardinal number for the system to determine the course that they want to register. The cardinal number will be stored in a variable called “no”.
- + The system will store the cardinal number in the variable “No” of the pointer pStudents (the first node of student course linked list). (pStudents→No = no)
- + Then, the system will compare the pStudents→No to variable “No” of all the pointers in courses linked list. (using pCurrent to scan through all the pointers in the courses linked list)
  - If pStudents is a null pointer and pCurrent→No = pStudents→No, then the system will copy all the values of the pointer pCurrent and paste to the pStudents.
  - If the pStudents is not a null pointer, we use a pointer pCur to create a new node in the Student courses linked list. If pCur→No = pCurrent→No, the system will copy all the values of pointer pCurrent and paste to the pCur. Scanning through all the nodes in the Student courses linked list to check whether pCur→session and pCur→Day is duplicated, if pCur→session and pCur→Day are duplicated, the system will ask the student to choose another course.
  - After picking all the courses, the system prints out a menu including continue enrolling, view the list of courses, back to the menu and ask them to choose one task.
  - If the students choose “Continue Enrolling”, the system will ask them to type in the number of courses that they want to enroll again (if they have enrolled 5 courses, the system will inform them to stop)
  - If the students choose “Back to the Menu” the system will show them the main menu.

### **III.3.2. Remove Enrolled Courses**

- Function: void RemoveACourses(Courses\*& pHead)
- Idea:
  - + First, the system prints out the list of courses that students enrolled in before.
  - + Then, the system requires students to type in an integer (which is the cardinal number of the course that they want to remove) and store it in a variable called “no”.
  - + Then, the system scans through the courses in the Student courses linked list to find out the course that the students want to remove by comparing the variable “No” of each node to the variable “no” (using pCur to scan).
  - + Using a pPrev as the previous node of the node pCur.
  - + Remove pCur if pCur→No = no and then, use pPrev to link to the next node of pCur.
  - + If pHead→No = no, use a pointer called pTemp to mark the old pHead, assign pHead = pHead→pNext and delete pTemp. The pPrev and pCur are assigned to pHead.
  - + After picking the course, The function will automatically return to the menu-screen.

### **III.3.3. View list of courses**

- Function: void PrintScoreBoardOfStudents(ScoreBoardOfStudent\*& pHead).
- Idea:
  - + Using library <iomanip> for design user interface as a table to show the Scoreboard. (setw, setfill,...)
  - + Print out the No, Course ID, Course Name, Day, Session, Teacher Name.

### **III.3.4. View the scoreboard**

- Function: void PrintScoreBoardOfStudents(ScoreBoardOfStudent\*& pHead).
- Idea:
  - + Using library <iomanip> for design user interface as a table to show the Scoreboard. (setw, setfill,...)

- + Print out the No, Course ID, Course Name, Midterm Score, Final term Score, Other score, Overall.

### **III.3.5. Menu of student**

- Function: void MenuOfStudent(SignIn \*pStaff, SignIn\* pStudent, SchoolYear \*&pHead, SchoolYear \*&pCurr).
- Idea:
  - + Have all functions of student:
  - + Change password
  - + View profile info
  - + Enroll courses (Only if the current time is within the current courses registration session)
  - + View enrolled courses.
  - + Cancel the courses.
  - + Log out.
  - + Using switch case and label. For each case (except for log out) the user will move to another screen or back to the menu of the student.

### **III.4. Delete function**

- Function:
  - + void deleteSemester(Semester \*&pHead);
  - + void deleteSchoolYear(SchoolYear \*&pHead);
  - + void deleteClasses(Classes \*&pHead);
  - + void deleteStudents(Students \*&pHead);
  - + void deleteSignIn(SignIn \*&pHead);
  - + void deleteCourses(Courses \*&pHead);
  - + void deleteScoreBoardOfCourse(ScoreBoardOfCourse \*&pHead);
  - + void deleteScoreBoardOfStudent(ScoreBoardOfStudent \*&pHead);
  - + void deleteScoreBoard(ScoreBoardOfCourse \*&pBoard, ScoreBoardOfStudent \*&pStudent);
- Idea:

- + Delete every allocated memory of the linked list until it points to nullptr.

### **III.5. Store information**

- Function:
  - + void foutCourses(SchoolYear \*pHead);
  - + void foutSignInStudent(SignIn \*pHead);
  - + void foutSignInStaff(SignIn \*pHead);
  - + void foutSchoolYear(SchoolYear \*pHead);
  - + void foutClasses(SchoolYear\*pHead);
  - + void foutSemester(SchoolYear\*pHead);
  - + void foutEnrolled(SchoolYear \*pHead,SignIn \*pStudent);
  - + void foutCurrentSemester(SchoolYear \*pHead);
- Idea:
  - + Store any information of the project at a folder called SchoolYear□ name and some text file to recall it.
  - + Use ofstream variable to store information at a txt file.

### **III.6. Read information**

- Function:
  - + void readSchoolYearlist(SchoolYear \*&pHead,SchoolYear \*&pCurr);
  - + void readClassListAndStudent(SchoolYear \*pHead);
  - + void readClassList(SchoolYear \*pHead);
  - + void readEnrolled(SchoolYear \*pHead,SignIn \*pStudent);
  - + bool checkCourses(Courses \*pStudents, Courses \*pC);
  - + void readSemesterList(SchoolYear \*pHead);
  - + void readCurrentSemester(SchoolYear \*pHead);
  - + void readCourses(SchoolYear \*pHead);
  - + bool readfile();
- Idea:
  - + If readfile is true then the program will read any existed information.

- + Use ifstream variable to read information from a text file or a csv file.

## IV. Courses Registration System Introduction

### IV.1. Function tree

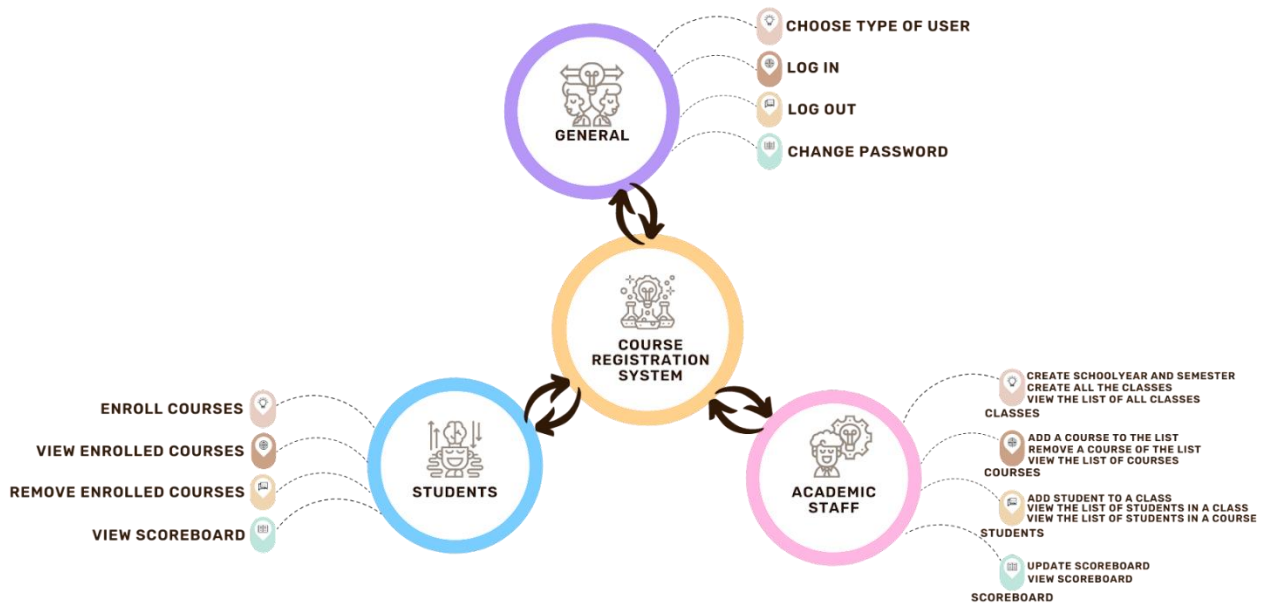


Figure 2. Function tree

### IV.2. General

#### IV.2.1. Change his/her password

- The users can change their password so it's much easier for them to remember.
- Example:

+ Current password: uIl21!Ms

+ New password: 20030920

#### IV.2.2. View his/her profile

- The users can view their profile, including: ID, Class,...
- Example:

+ Class: 21CLC09

+ Student ID: 21127697

### IV.3. Staff

#### IV.3.1. Create a new school year

- The staff can type in a new school year to create a new school year.
- Example: 2020 - 2021, 2021 - 2022...,

#### **IV.3.2. Create a new semester**

- The staff can type in a new semester to create a new semester.
- Example: Semester 1 of School Year 2020\_2021, Semester 2 of School Year 2020\_2021...

#### **IV.3.3. Create new classes for 1<sup>st</sup> year students**

- The staff can type in the Class ID to add a new class to the list of classes.
- Example: 21CLC09, 21CTT03, 21CNTN...

#### **IV.3.4. View the list of classes**

- The staff can view the list of all classes that were added before.
- Example:

21CLC01	21CTT01
21CLC02	21CTT02
21CLC03	21CTT03
21CLC04	21CTT04
21CLC05	21CTT05
21CLC06	21CTT06
21CLC07	21CTT07
21CLC08	21CTT08
21CLC09	21CTT09

21CLC10	21CTT10
---------	---------

Table 2. The list of classes

#### IV.3.5. Add students to a class

- The staff can add students and their information to a class by importing a CSV file to the folder that contains the program.
- Example:

No	Student ID	Name	Social ID	Date of Birth	Gender
1	21127212	Pham Do Hoai An	123456789	3/1/2003	Female
2	21127581	Tran Ngoc Viet Anh	123456790	3/2/2003	Male
3	21127580	Phan Lam Anh	123456791	3/3/2003	Male
4	21127007	Nguyen Quoc Anh	123456792	3/4/2003	Male
5	21127013	Nguyen Phu Minh Bao	123456793	3/5/2003	Male
6	21127602	Nguyen Hoang Duy	123456794	3/6/2003	Male
7	21127206	Pham Dang Son Ha	123456795	3/7/2003	Male
8	21127604	Nguyen Lam Hai	123456796	3/8/2003	Male
9	21127038	Vo Phu Han	123456797	3/9/2003	Male
10	21127278	Nguyen Trong Hieu	123456798	3/10/2003	Male
11	21127307	Tong Gia Huy	123456799	3/11/2003	Male
12	21127055	Huynh Nguyen Minh Hu	123456800	3/12/2003	Male
13	21127618	Nguyen Khang Hy	123456801	3/13/2003	Male
14	21127071	Nguyen Cong Khanh	123456803	3/15/2003	Male
15	21127517	Nguyen Anh Khoa	123456804	3/16/2003	Male
16	21127632	Nguyen Cao Khoi	123456805	3/17/2003	Male
17	21127635	Nguyen Khanh Anh Kiet	123456806	3/18/2003	Male
18	21127347	Dang Hoang Long	123456808	3/20/2003	Male
19	21127532	Tran Thu Minh	123456809	3/21/2003	Male
20	21127303	Do Nhat Minh	123456810	3/22/2003	Male
21	21127728	Nguyen Cao Nhat Nam	123456811	3/23/2003	Male
22	21127367	Do The Nghia	123456812	3/24/2003	Male
23	21127384	Duong Hanh Nhi	123456813	3/25/2003	Female
24	21127661	Fa Ngoc Uyen Nhi	123456814	3/26/2003	Female
25	21127664	Tran Dai Nien	123456815	3/27/2003	Male

Figure 3. List of students

#### IV.3.6. View the list of students in a class

- The staff can view the list of students in a class.
- Example: (ảnh màn hình console)

#### IV.3.7. Create the list of courses

- The staff can create the list of courses by typing in the information of a course.
- Example:

No	Course ID	Course Name	Credits	Day	Session	Teacher Name
1	MTH00001	Discrete Mathematics	3	MON	07h30	Nguyễn Kim Ngọc
2	CSC10002	Programming Techniques	4	SAT	15h30	Đinh Bá Tiến

Table 3. The list of students in a class

#### IV.3.8. Update the information of courses

- The staff can change the information of a course in the list of courses.
- Example: Discrete Mathematics
  - + Old session: 07h30
  - + New session: 09h30

#### IV.3.9. Remove a course from the list of courses

- The staff can delete a course from the list of courses.
- Example:
  - + Old list:

No	Course ID	Course Name	Credits	Day	Session	Teacher Name
1	MTH00001	Discrete Mathematics	3	MON	07h30	Nguyễn Kim Ngọc
2	CSC10002	Programming Techniques	4	SAT	15h30	Đinh Bá Tiến

Table 4. The list of courses

- + New list:

No	Course ID	Course Name	Credits	Day	Session	Teacher Name
----	-----------	-------------	---------	-----	---------	--------------



1	MTH00001	Discrete Mathematics	3	MON	07h30	Nguyễn Kim Ngọc
---	----------	-------------------------	---	-----	-------	-----------------

Table 5. The list of courses

#### IV.3.10. View the list of courses

- The staff can view the list of courses to check if it's okay or not.
- Example:

```

*****STAFF*****
*
* 1.Create school year and classes *
* 2.Create semesters and courses *
* 3.View the list of all classes *
* 4.View the list of all courses,update and delete courses *
* 5.View the list of all student in a class *
* 6.View the list of all student in a course *
* 7.Back *
*****
Your Choice: 4
No Course ID Course Name Credit Day 1 Session 1 Day 2 Session 2 Teacher
-----
1 MTH1 MTH1 3 MON 09:30 TUE 09:30 t
2 MTH2 MTH2 2 MON 09:30 TUE 07:30 N
3 MTH3 MTH3 3 WED 07:30 WED 09:30 K
4 MTH4 MTH4 5 THU 09:30 THU 13:30 O
Press any key to continue . . .

```

View the list of courses that you have added

Figure 4 - View the list of courses

#### IV.3.11. View the list of students in a course

- The staff can view the list of students who enrolled in this course.
- Example:

```

No Course ID Course Name Credit Day 1 Session 1 Day 2 Session 2 Teacher
-----
1 MTH1 MTH1 3 MON 09:30 TUE 09:30 t
2 MTH2 MTH2 2 MON 09:30 TUE 07:30 N
3 MTH4 MTH4 5 THU 09:30 THU 13:30 O
Input number of course you want to see student: 1
-----LIST OF STUDENTS IN COURSE-----
No StudentID SocialID LastName FirstName Gender DateofBirth
1 21127883 2123456791 Tran Van C Male 1/3/2003
Press any key to continue . . .

```

Choose a course to view the list of students who enrolled in that course

Figure 5 - View the list of students who enroll in the course

### IV.3.12. Import Scoreboard

- The staff can create the scoreboard for students in a course by importing a CSV file.
- Example:

	A	B	C	D	E	F	G	H	I	J
1	No	Student ID	First Name	Last Name	Date Of Bi	Class	Mid Term	Final Term	Other Scor	Overall
2	1	21127003	C	Tran Van	1/3/2003	21CLC09	8	8	0	8
3	2	21127004	D	Tran Thi	1/4/2003	21CLC09	9	8	0	9.4
4	3	21127005	E	Le Van	1/5/2003	21CLC10	7	7.5	0	7.3
5	4	21127006	E	Le Thi	1/6/2003	21CLC10	3	5	0	4.2

Figure 6 - Scoreboard CSV file

### IV.3.13. Update Scoreboard

- The staff can update the scoreboard of students in the course when there are changes.
- Example:

```
*****STAFF*****
*****
* 1.Create school year and classes *
* 2.Create semesters and courses *
* 3.View the list of all classes *
* 4.View the list of all courses,update and delete courses *
* 5.View the list of all student in a class *
* 6.View the list of all student in a course *
* 7.Back *
*****

Your Choice: 2

School Year list:
2021-2022
Input school year this semester belong to (Example: 2021-2022): 2021-2022
Semester list:
Empty
Input semester number: 1
Input start date (DD MM YYYY): 01 01 2022
Input end date (DD MM YYYY): 10 10 2022
No Course ID Course Name Credit Day 1 Session 1 Day 2 Session 2 Teacher
-----
Input course's ID: MTH1
Input name of course: MTH1
Input weekday of session 1: MON
Input time of session 1: 07:30
Input weekday of session 2: TUE
Input time of session 2: 09:30
Input maximum number of students (Default: 50):
Input Number of credits: 4
Input teacher's name: A
Continue adding courses? (y = yes, n = no): y
No Course ID Course Name Credit Day 1 Session 1 Day 2 Session 2 Teacher
-----
1 MTH1 MTH1 4 MON 07:30 TUE 09:30 A
Input course's ID: MTH2
Input name of course: MTH2
Input weekday of session 1: TUE
Input time of session 1: 07:30
Input weekday of session 2: TUE
Input time of session 2: 09:30
Input maximum number of students (Default: 50):
Input Number of credits: 5
Input teacher's name: 
```

Figure 7 - Update scoreboard

## IV.4. Student

### IV.4.1. Enroll in courses

- The students can enroll in at most 5 courses.
- Example:

No	Course ID	Course Name	Credit	Day 1	Session 1	Day 2	Session 2	Teacher
1	MTH1	MTH1	3	MON	09:30	TUE	09:30	t
2	MTH2	MTH2	2	MON	09:30	TUE	07:30	N
3	MTH4	MTH4	5	THU	09:30	THU	13:30	O
Input number of courses you want to pick: 1								
No	Course ID	Course Name	Credit	Day 1	Session 1	Day 2	Session 2	Teacher
1	MTH1	MTH1	3	MON	09:30	TUE	09:30	t
Continue enroll? (y = yes, n = no)								
y								
No	Course ID	Course Name	Credit	Day 1	Session 1	Day 2	Session 2	Teacher
1	MTH1	MTH1	3	MON	09:30	TUE	09:30	t
2	MTH2	MTH2	2	MON	09:30	TUE	07:30	N
3	MTH4	MTH4	5	THU	09:30	THU	13:30	O
Input number of courses you want to pick: 2								
Time of courses conflicted.								
Press any key to continue . . .								
No	Course ID	Course Name	Credit	Day 1	Session 1	Day 2	Session 2	Teacher
1	MTH1	MTH1	3	MON	09:30	TUE	09:30	t
Continue enroll? (y = yes, n = no)								
y								
No	Course ID	Course Name	Credit	Day 1	Session 1	Day 2	Session 2	Teacher
1	MTH1	MTH1	3	MON	09:30	TUE	09:30	t
2	MTH2	MTH2	2	MON	09:30	TUE	07:30	N
3	MTH4	MTH4	5	THU	09:30	THU	13:30	O
Input number of courses you want to pick: █								

Figure 8 - Enroll in courses

#### IV.4.2. Remove a course that was enrolled in

- The students can remove any course that they enrolled in before.
- Example:
  - + Old list:

No	Course ID	Course Name	Credits	Day	Session	Teacher Name
1	MTH00001	Discrete Mathematics	3	MON	07h30	Nguyễn Kim Ngọc
2	CSC10002	Programming Techniques	4	SAT	15h30	Đinh Bá Tiến

Table 6. The list of courses

- + New list:

No	Course ID	Course Name	Credits	Day	Session	Teacher Name
1	MTH00001	Discrete Mathematics	3	MON	07h30	Nguyễn Kim Ngọc

Table 7. The list of courses

#### IV.4.3. View the list of courses

- The students can view the list of courses that they enrolled to check if it's okay or not.
- Example:



No	Course ID	Course Name	Credit	Day 1	Session 1	Day 2	Session 2	Teacher
1	MTH1	MTH1	3	MON	09:30	TUE	09:30	t
3	MTH4	MTH4	5	THU	09:30	THU	13:30	o

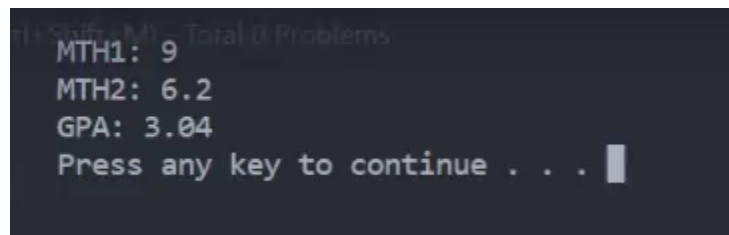
Press any key to continue . . .

After enrolling in courses, you can view the list of courses that you have enrolled

Figure 9 - View the list of courses in which were enrolled

#### IV.4.4. View their scoreboard

- The students can view their scoreboard to know their result after studying in the course.
- Example:



```

MTH1: 9
MTH2: 6.2
GPA: 3.04
Press any key to continue . . .

```

Figure 10 - View scoreboard

### IV.5. Product

**IV.5.1. User Manual Video:** <http://surl.li/bxfjv>

**IV.5.2. Link GitHub:** <http://surl.li/bxfsx>

## V. Work evaluating

### V.1. Percentage rating

Student ID	Full Name	Tasks Percentage
21127347	Đặng Hoàng Long	20%
21127604	Nguyễn Lâm Hải	40%
21127661	Fa Ngọc Uyển Nhi	20%
21127697	Lê Thị Minh Thư	20%

Table 8. Percentage rating

### V.2. Members rate others and themselves

#### V.2.1. 21127697 - Lê Thị Minh Thư

Student ID	Full Name	Score	Comment
21127347	Đặng Hoàng Long	9	<ul style="list-style-type: none"><li>– Finish all the tasks well.</li><li>– Being excellent at English.</li></ul>
21127604	Nguyễn Lâm Hải	10	<ul style="list-style-type: none"><li>– Active in trading information and working.</li><li>– Finish all the tasks effectively.</li><li>– Hard-working.</li></ul>
21127661	Fa Ngọc Uyển Nhi	9	<ul style="list-style-type: none"><li>– Finish all the tasks well.</li><li>– Active in giving ideas.</li></ul>
21127697	Lê Thị Minh Thư	9	<ul style="list-style-type: none"><li>– Finish all the tasks well.</li><li>– Active in finding information.</li></ul>

Table 9. Lê Thị Minh Thư's evaluation board

### V.2.2. 21127347 - Đặng Hoàng Long

Student ID	Full Name	Score	Comment
21127347	Đặng Hoàng Long	8	<ul style="list-style-type: none"> <li>– Average coding skills</li> <li>– Not proactive in taking tasks</li> <li>– Slow work progress.</li> </ul>
21127604	Nguyễn Lâm Hải	10	<ul style="list-style-type: none"> <li>– Diligent.</li> <li>– Excellent coding skills</li> <li>– Sometimes handles too much work.</li> </ul>
21127661	Fa Ngọc Uyển Nhi	9	<ul style="list-style-type: none"> <li>– Finished assigned tasks satisfactorily</li> <li>– Provided valuable ideas for the project.</li> </ul>
21127697	Lê Thị Minh Thư	9.5	<ul style="list-style-type: none"> <li>– Showed good leadership skills</li> <li>– Finished assigned tasks satisfactorily</li> <li>– Provided valuable ideas for the project.</li> </ul>

*Table 10. Dang Hoang Long's evaluation board*

### V.2.3. 21127661 - Fa Ngọc Uyển Nhi

Student ID	Full Name	Score	Comment
21127347	Đặng Hoàng Long	9	<ul style="list-style-type: none"> <li>– Actively working in a team</li> <li>– Completing work on time</li> <li>– Being responsible to the team.</li> </ul>
21127604	Nguyễn Lâm Hải	10	<ul style="list-style-type: none"> <li>– Very active in teamwork</li> <li>– Has the skills to handle situations quickly and flexibly</li> <li>– The one to cover the entire project</li> <li>– Completes the work on time.</li> </ul>

21127661	Fa Ngọc Uyển Nhi	9	<ul style="list-style-type: none"> <li>– Actively participates in team work</li> <li>– Complete the work on time, get along</li> <li>– But the ability to code has many limitations that need to be overcome.</li> </ul>
21127697	Lê Thị Minh Thư	9	<ul style="list-style-type: none"> <li>– Reasonable assignment of work</li> <li>– Promoting the work of the team</li> <li>– Completing the work on time</li> <li>– But the ability to code has many limitations that need to be overcome more.</li> </ul>

Table 11. Fa Ngọc Uyển Nhi's evaluation board

#### V.2.4. 21127604 - Nguyễn Lâm Hải

Student ID	Full Name	Score	Comment
21127347	Đặng Hoàng Long	9	<ul style="list-style-type: none"> <li>– Actively working in a team</li> <li>– Completing work on time</li> <li>– Being excellent at English.</li> </ul>
21127604	Nguyễn Lâm Hải	9	<ul style="list-style-type: none"> <li>– Not being informative enough when explaining how the functions should be done.</li> <li>– Cannot finish some functions properly</li> </ul>
21127661	Fa Ngọc Uyển Nhi	9	<ul style="list-style-type: none"> <li>– Finished assigned tasks very well</li> <li>– Some ideas are invaluable to the project</li> </ul>
21127697	Lê Thị Minh Thư	9	<ul style="list-style-type: none"> <li>– Good leadership skills</li> <li>– Good coding skill but some functions are pretty hard to understand</li> </ul>

Table 12. Nguyen Lam Hai's evaluation board

## **VI. Reference source**

<http://surl.li/bxcxk>

<http://surl.li/bxczo>