CS162

Group Project

A Course Management System

Group 2

Members

1. 22125003 - Trần Công Lâm Anh - 22TT1
2. 22125036 - Nguyễn Kim Khanh - 22TT1
3. 22125063 - Bùi Danh Nghệ - 22TT1
4. 22125090 - Nguyễn Ngọc Duy Tân - 22TT1

Introduction to system

The system will run and display in the console.

There are two types of users: students and staff.

Students can interact with the system to: register for a course, view their scoreboards,…

Staff can work with the system: organize a course, add students to courses,…

Both users will have accounts with an able-changing password.

The data will be stored in a folder and file.

Convention

Struct: PascalCase (must be named succinctly and easy to remember)

Function: camelCase (the name of main function could describe the task)

Variable: camelCase (not too long and complicated)

File

All of files used to creating data for demo were first created via Excel, and then exported as Text (tab delimited) (*.txt), hence it could be much easier to transport between *.txt and *.xlsx)

There are:

- Files for all students that were admitted in each school year ([Name School year].txt)
- Files for all courses that are supposed to be created in the semester (dshp.txt)
- Files contains list of students registered for the each course ([ID of course]_[Name class of course].txt)
- Files for the lecturer to input the result, and hence could be read back to update result of the student after finishing the course ([ID of course]_[Name class of course]_mark.txt)

Link

https://docs.google.com/spreadsheets/d/1v7ZHwGI3eSAy2ckVsYheStUG4jBN51k-tiyO_HuOwSQ/htmlview

https://docs.google.com/spreadsheets/d/1FKrLkoy6Dy1qq732hrOyGt2fTDlE7ApcbtvfsVqj7xA/edit

DATA STRUCTURE

```
struct Date;
struct Class;
struct SchoolYear;
struct StuInCourse;
struct Student;
struct Semester;
struct Staff;
```

struct **Date**

```
struct Date {
    string day;
    string month;
    string year;
};
```

Struct Date includes 3 **strings** storing **day**, **month** and **year** of a date.

struct **Class**

```
struct Class {
    //detail
    string name;
    int requiredCredits; //22TT1-165 tin chi de tot nghiep
    int numStu = 50;

    //node
    Student* stuHead = NULL;
    Class* classNext = NULL;
    SchoolYear* inSY = NULL;
};
```

Describe:

Used to create Linked list of all classes that created in the starting of new school year

Information: *Used to store the main detail (mainly in the form of string, integer,...) of the structure itself*

string **name**: the name of class

int **requiredCredits**: the number of credits to graduate.

int **numStu**: the number of students (currently) in a class.

Main nodes: *Used to exploit the information from other structs and make ease when building functions*

Student* **stuHead**: the linked list's head of students

Class* **classNext**: point to the next **Class**.

SchoolYear* **inSY**: present school year the class is taken in

struct **SchoolYear**

```
struct SchoolYear {
    //detail
    string name; //ex: 2022-2023

    //node
    Class* classHead = NULL;
    //  Course* courseHead = NULL;
    SchoolYear* yearNext = NULL;

    //array
    Semester sm[3];
};
```

Describe:

SchoolYear, created in the form of Linkedlist, will contain the data of all classes and courses created by the staff of the school.

Information:

string **name**: name of created school year *(ex: 2022-2023)*

Semester **sm[3]**: an array of struct **Semester**

Main nodes:

Class* **classHead**: the head of a linked list including all of classes in that school year

SchoolYear* **yearNext**: points to the next **SchoolYear**

struct StuInCourse

```
struct StuInCourse {
    //detail
    float totalM;
    float finalM;
    float midM;
    float otherM;
    float GPA;
    string courseID;
    Course* infoCourse = NULL;

    //node
    Student* stuInClass = NULL; //point to the node Student in a Class
    StuInCourse* stuNext = NULL; //next of linkedlist
    StuInCourse* pStuCourseNext = NULL; // that student in another Course
};
```

Describe:

The structure is aimed at building a **LinkedList** data structure, in which every **Student** registering for every single **Course** will be considered as a node **StuInCourse** in that **Course**.

This node contains the information of marks of the **Student** in that **Course** and many nodes that could utilize more detailed data from other structures.

Information: *Mainly used to store data of mark, grade (float, string) of the **Student** in that **Course***

float **totalM**: average score *(0-10 point scale)*

float **finalM**: Final exam's score *(0-10 point scale)*

float **midM**: Midterm exam's score *(0-10 point scale)*

float **otherM**: other component points like plus points, exercise points,... *(0-10 point scale)*

` float **GPA**: standard GPA *(0-4.0 point scale)*

string **courseID**: his/her course's ID

Main nodes:

Course* **infoCourse**: point back to his/her **Course** which stores all the information about the **Course**

Student* **stuInClass**: point back to his/her node **Student** for retrieving his/her information

StuInCourse* **stuNext**: point to the next node for the next **Student** in that **Course**.

StuInCourse* **pStuCourseNext**: point to the node **StuInCourse** of that student but in another **Course** that also registered by that **Student**

struct **Student**

```
struct Student {
    //detail
    string No;
    string StuID;
    string firstName;
    string lastName;
    string gender;
    Date dateOfBirth;
    string socialID;
    string className;
    string password = "123"; //mac dinh
    string curriculum;

    int accumCredits = 0;
    float accumScore = 0;

    //node
    Class* inClass = NULL;
    Student* stuNext = NULL; //next of linkedlist
    StuInCourse* pStuCourseHead = NULL;
};
```

Describe:

The structure is used for storing all information needed for the student, which are mostly retrieved by many other structures and functions.

Information:

string **No**: the ordinal number in the class of students *(ex: 1,2,...)*
string **stuID**: ID of the Student in that School *(ex: 22125001)*
string **firstName**: student's first name
string **lastName**: student's last name
string **gender**: student's gender *("Male", "Female")*
Date **dateOfBirth**: student's date of birth *(take advantage of struct **Date**: day, month, year)*
string **socialID**: his/her number on ID card.
string **className**: his/her class's name.

string **password**: his/her password for the login. *(Default: 123)*

string **curriculum**: the program he/she is learning **(Computer Science, Information Technology,...)**

int **accumCredits**: accumulated credits he/she has achieved so far

float **accumScore**: average score he/she has accumulated so far

Main nodes:

Class* **inClass**: point to the node that stores his/her **Class**, hence one could retrieve all the information about his/her **Class**

Student* **stuNext**: point to the next node of the next **Student** in the **Class**

StuInCourse* **pStuCourseHead**: the head of a Linked List of all the **StuInCourse** nodes (each node represent single Course that has been ever registered by him/her)

struct Semester

```
struct Semester {
    //detail
    Date start;
    Date end;
    string syName;
    bool state = 0; //if 1 - accessible
    int num;

    //node
    Course* courseHead = NULL;
    SchoolYear* inSY = NULL;
};
```

Describe:

Semester structure contains basic information such as the period time as well as all the courses of any semester in a given school year.

Used mainly in struct SchoolYear *(appears as Array)*

Detail:

Date **start**: starting date of this semester *(day, month, year)*

Date **end**: ending date of this semester *(day, month, year)*

string **syName**: name of the school year which this semester is in

bool **state**: check whether this is the current semester *(1: true; 0: false)*

int **num**: the ordinal number of this semester *(ex: semester 1 -> num = 1)*

Main nodes:

Course* **courseHead**: points to the head of the **Course** Linked list storing all courses' created in this semester.

SchoolYear* **inSY**: points to the school year which this semester is in.

struct **Staff**

```
struct Staff {
    //elements
    string ID; //username == ID
    string name;
    string password = "10diemLy"; //mac dinh
    Staff* staffNext = NULL;
```

Describe:

Used to build Staff Linkedlist in which each node stores basic information of the staff

Includes myriads of functions available only for the staff

Information:

string **ID**: similar to the student's ID, each staff member has his/her own ID.

string **name**: Staff's name

string **password**: Staff's password for the login *(Default: "10diemLy")*

Main nodes:

Staff* **staffNex**t: points to the next **Staff** in the linked list.

Staff's function on system:

```
    //Main functions
    void exportCourseToCSV(Course* course); //19
    void importScoreboard(Course* course); //20
    void viewScoreboard(Course* course); //21
    void updateRes(Course* course); //22
    void viewScoreClass(SchoolYear* sy, int numSm, Class* cl); //24

    //Supplementary Functions
    void addStuFromDSDKHP(Course* course, Student* s);
    void createCourseFromCSV(SchoolYear* sy, int numSm);
    bool deleteACourse(Course*& courseHead, string courseID);
    bool addAStudentInCourse(Course*& a, Student* newStu);
    bool removeAStudentFromCourse(Course*& a, string ID);

    //Change and update course's information
    void viewOptions();
    void viewCourseInfo(Course* course);
    void updateCourseInfo(Course* course);

    Class* findClass(SchoolYear*& sy, string nameClass);
    void addStuToClass(Class*& cl, Student*& stu);
    void createAllClassesFromCSV(SchoolYear*& sy, string syname, InfoStu*& info);
    void createAllCourse(SchoolYear* sy);
};
```

struct InfoStu

```
struct InfoStu {
    Student* stuInClass = NULL;
    Class* infoClass = NULL;
    InfoStu* InfoStuNext = NULL;
    SchoolYear* sy = NULL; //*

    //Main functions
    void viewScoreBoard(SchoolYear* sy, int numSm); //24

    //Supplementary functions
    void viewCourses(Semester s); // View all the courses that the student participates in.
    bool checkCourseName(Course*& course, string s);
    bool checkIfExist(Course*& c);
    void importStuToCourseCSV(string courseName);
    void countCredit(Course* c, int& cre, int& numCourse);
    bool addAndSortByID(Course*& c);
    void printListCourse(Course* c, int cre, int numCourse);
    void selectCourse();
    void viewSchedule(Semester sm, Student *s);
};
```

Describe:

Used to build Linkedlist whose nodes represent each student of the school so far.

Contain various functions that only students could use.

Main nodes: *Since its take advantage of all other struct, it use nodes to retrieve needed information*

Student* **StuInClass**: point to the node Student in his/her Class (retrieving student's data)

Class* infoClass: point to the node Class of that student (retrieving information of the student's class)

Infostu* **InfoStuNext:** point to the next node in InfoStu Linked List

SchoolYear* **sy**: point to the node SchoolYear in which that student was admitted to the school