

BACH KHOA UNIVERSITY
COMPUTER SCIENCE & ENGINEERING FACULTY



PROGRAMMING FUNDAMENTAL

ASSIGNMENT
SCHOOL MANAGEMENT

Instructor: Tran Tuan Anh

Group 2:

Huynh Kien Van – 1552423 (Leader)

Nguyen Huu Canh – 1652066

Bui Nguyen Vu - 1552437

Nguyen Vu Thanh Nguyen – 1652437

Tran Minh Hung - 1652271

INDEXES

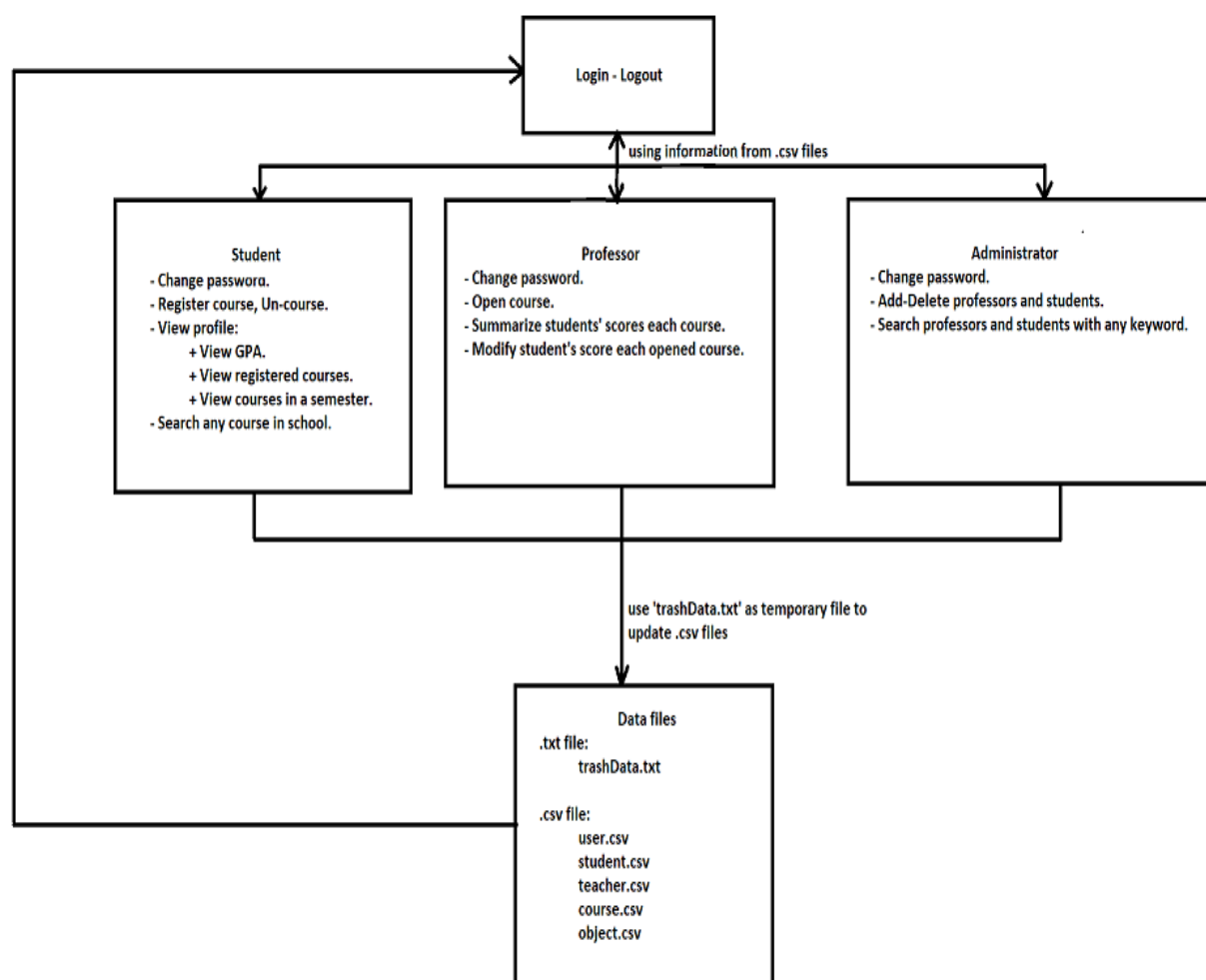
I, Introduction.....	4
II, Flowchart.....	5
III, Completed tasks.....	6
IV, Additional functions.....	7
V, Problems.....	7

I, Introduction:

University Management System is a large database system which can be used for managing activities. Management System allows users to store almost their information electronically, including information on students, properties, teaching meteorites.

In this assignment, we are going to implement a part of mentioned functions above. There are three types of user: administrator, teacher and student. Administrators are people who can add or remove other types of user to or out of the database. Teachers can grade or modify students' scores, open courses and summarize students' scores. Students only register courses, un-course or view their own profile such as: view Grade Point Average (GPA), registered courses or view courses within semester.

II, Flowchart:



III, Completed tasks:

The following list contains tasks that we have solved and the corresponding score of each task.

Login – Logout:

- 2.1.1. There are three types of user. (1.0 point)
- 2.1.2. When login failed, user can login again. (0.5 point)
- 2.1.3. Each user can logout. (0.5 point)

Administrator:

- 2.2.1. Add-Delete professor and student. (0.5 point)
- 2.2.2. Change password. (0.5 point)

Professor:

- 2.3.1. Change password. (0.5 point)
- 2.3.2. Open course. (0.5 point)
- 2.3.3. Summarize students' score with each opened course. (0.5 point)
- 2.3.4. Modify student scores with each opened course. (0.5 point)

Student:

- 2.4.1. Change password. (0.5 point)
- 2.4.2. Register course - Un-course. (1.0 point)
- 2.4.3. View profile:
 - View GPA (The result is only true with a specific semester).(??? point)
 - View course during the semester. (0.5 point)
 - View registered courses. (random order) (??? Point)

Functional Accessory:

- 3.1. Search by name. (1.0 point)
- 3.2. Delete by name. (1.0 point)

IV, Advance functions:

In this assignment, we implemented some function with advance utility for advance purposes of part III(**Functional Accessory**):

- **bool checkDigit(std::string str)** : In option menu in **Administrator.cpp**, **Student.cpp**, **Professor.cpp**, the user is forced to type with the provided options to control the program runs well.
- **bool checkSubStr()**: functions with that name defined in **commonFunc.cpp** used to check substring of the string. If the string A contains string B, it returns **true**. Otherwise, it returns **false**.
- When a student register a course, his/her score will be '**NULL**' to differ them from others(They only register or still learning the course. They have not finish the course yet). If they finished their course, their Professors would mark them and they can not un-course anymore.
- When student un-course the course, it depends on the Score value of the subject they want to retrieve. If the Score value is '**NULL**', it means that they **can retrieve** their course. Otherwise, they finished their course so that they can not un-course.

V, Problems:

The following list is the problems that we can not solve.

Student:

2.4.3. View profile:

- View GPA (Cannot see all semester the same time, only works with a specific semester).
- View registered courses. (Cannot show the result in ascending order).

More problems:

+) When input/type to the command line, the user have to concern about the case insensitive. For example: The User types '**Van A**' or substrings of '**Nguyen Van A**' to search '**Nguyen Van A**'. If the User types '**van a**', it returns another result.