# BẢNG ĐÁNH GIÁ THÀNH VIÊN
## Nhóm 3

| Số thứ tự | MSSV | Họ và tên | Công việc | Số commit | Số giờ làm | % đóng góp |
|---|---|---|---|---|---|---|
| 1 | 20125010 | Trần Bảo Lợi | 1. Create login method<br>2. Add create user when add a new student<br>3. Get and save user list<br>4. Enroll in a course<br>5. count current course at this semester of user<br>6. function to check if the course id is available<br>7. A part of check is a course conflicted<br>8. Get course from course Id<br>9. View list of enrolled a course<br>10. Remove a course from enrolled list<br>11. View list of class<br>12. Link other function to user menu<br>13. Function to change user info<br>14. Test and fix other function to make it fitted to use<br>15. Get path link to current semester<br>16. Covert string to char* to use in some case<br>17. Get course list of user<br>18. Get student info from user<br>19. Get current date | 38 | 49 giờ | 25% |
| 2 | 20125034 | Võ Bùi Phú Hưng | 1. Create a semester: 1, 2, or 3, school year, start date, end date.<br>2. Choose the school year that the newly created semester belongs to. | 27 | 35 - 36 giờ | 25% |

| | | | 3.     Save the new semester to file.<br>4.     Load the semester list of one school year from file.<br>5.     Create a course registration session: start date, end date.<br>6.     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.<br>7.     Save the new course to file.<br>8.     Load the course list of one semester from file.<br>9.     View list of courses.<br>10.    Update course information.<br>11.    Delete a course.<br>12.    Create .csv files about semester and course. | | | |
| 3 | 20125006 | Nguyễn Hoàng Khang | 1. Task management<br>2. View list of class in a course.<br>3. Export list of students in a course to a CSV file<br>4. 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)<br>5. View the scoreboard of a course. | 25 | 30 - 32 giờ | 25% |

| | | | 6. Update a student result.<br>7. View the scoreboard of a class, including final marks of all courses in the semester, GPA in this semester, and the overall GPA.<br>8. View his/her scoreboard.<br>9. Implement clear screen function.<br>10. Implement press to continue function. | | | |
|---|---|---|---|---|---|---|
| 4 | 1751067 | Nguyễn Minh Hiếu | 0. Create some initial text file like schoolyear.txt (store previous school years)<br>Classes.txt (store previous years classes), add some struct for later usage, ...<br>1. Create a school year (2020-2021, for example):<br>    a. Load School years from txt file for later usage. (linked-list)<br>    b. Check if the school year user want to create is existed.<br><br>    c. Create school year. (write to file and attach to linked-list)<br><br>    d. Delete School year List after usage<br>2. Create several classes for 1st year students. For example: class 20APCS1, class 20APCS2, class 20CLC1â€¦, class 20CLC11, class 20VPâ€¦<br>    a. Load all the classes from txt file.(linked-list)<br>    b. In each class, load the student list.<br>    c. Check if the class user want to create is existed.<br>    d. Create new class. (write to file and attach to linked-list)<br><br>    e. Delete class list.<br>    f. Delete student list. | 25 | 35 - 38 giờ | 25% |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | 3. Add new 1st year students to 1st-year classes.<br>      1. add one by one:<br>            a. check if class is exist and return the class.<br>            b. create student (check valid input, etc..) then write to csv file in append mode and attach to student list.<br>      2. import student to class by csv:<br>            a. check if class is exist and return the class.<br>            b. check if file exists.<br>            c. load student into class. (delete the older list and create new list).<br>4. Support other teammates (debug + fix some bugs). | | | |