TEST REPORT

YUNUS EMRE AVCI-111044040

GENERAL INFORMATION ABOUT PROJECT:

➢*This project will be done by one person.*

➢*Program implement a student automation system which will carry out some tasks about courses, students and instructors.*

➢*Program executes different modes: "Student mode" and "Instructor mode".*

➢*When the program executes, it wants to user name and password on the beginning.*

➢*Program can create own database file that token necessary information from "csv" files.*

➢*If student wants to save courses which enrolled they save to csv files as "<student number>_courses.csv".*

➢ *If instructors want to save their opened courses, program can create csv file as "<name_surname>_openCourses.csv".*

➢*Program make operations according to users that can be a student or an instructor.*

➢*Functions take datas from database and update database*

➢*Students can operations like that; adding course, removing course and listing their enrolled courses.*

➢*Instructors can operations like that;*

**opening a course,*
**finalizing a course,*
**assigning grades to students,*
**listing of the opened courses of them,*
**listing grades of enroll student at chosen course.*

DIFFERENCES

➢*There are some changes about functions prototypes. I realized that most of functions have uncessary and unused parameters. So I have to remove parameters to efficient project.*

➢*There are changes about some structures . Somewhere I need some same information so I added new variable to some structures.*

➢*To avoid from code repetitons and to increase productivity I added new functions to Project.*

STRUCTURES

>***Course structure***

```
typedef struct{
        char courseID[TC];                      /*Course's ID           */
        char courseName[MAX_NAME];              /*Course's name         */
        char courseInstructor[MAX_NAME];/*course's instructor's information*/
        char prequisites[MAX_COURSE][MAX_COURSE];/*pre. Of course      */
        int isOpen;                             /*course is open or not */
        char mode;                              /*add/remove is open or not */
        int numOfPrequisites;                   /*number of course's prequisites */
        double grades;                          /*Student's grades      */
}course_t;
```

>***Student structure***

```
typedef struct{
        char TCnum[TC];                 /*Student's TC number   */
        char name[MAX_NAME];            /*Student's name        */
        char surname[MAX_NAME];         /*Student's surname     */
```

```
        char studentNum[STU_NUM];       /*Student number              */

        char startYear[TC];             /*Student's start year        */

        course_t enrolCourse[MAX_COURSE];  /*Enrolled course          */

        course_t previousCourse[MAX_COURSE];
                                          /*Student's previous courses*/
        int numOfEnCourse;              /*Number of enrolment course*/

        int numOfPreCourse;             /*Number of previous course */
}student_t;
```

>***Instructor structure***

```
typedef struct{

        char TCnum[TC];                 /*Instructor's TC number      */

        char name[MAX_NAME];            /*Instructor's name           */

        char surname[MAX_NAME];         /*Instructor's surname        */

        course_t openCourse[MAX_COURSE];/*Instructor's open courses*/

        int numOpCourse;                /*Number of Instructor's open course*/
}instructor_t;
```

>***User structure***

```
typedef struct{
        char userID[TC];                /*User's ID                   */

        char userType[MAX_NAME];        /*User'stype(student,instructor)*/

        char userName[MAX_NAME];        /*User's name                 */

        char password[PASSWORD];        /*User's password             */
}user_t;
```

FUCTION PROTOTYPES

➔***Read student's information from csv file and write to database***

*int readStudentStruct(int *sizeOfStu);*

➔***Read instructor's information from csv file and write to database***

*int readInstructorStrcut(int *sizeOfIns);*

➔***Read user's information from csv file and write to databese***

*int readUserStruct(int *sizeOfuser);*

➔***Read courses's information from csv file and write to databese.***

*int readCourseStruct(course_t *course, int *sizeOfcourse);*

➔***Lists the available courses for the student and enables the student to choose courses to enroll***

*int addCourse(student_t *student);*

➔***Shows the course list of the student and enables the student to remove a course from the list.***

*int removeCourse(student_t *student);*

➔***Shows the enrolled courses list of the student. Asks the student. The list will be saved into file if student want to save***

void listCourses(student_t student);

➔***Shows the list of the courses which are given by the instructor and to open a course for enrollment.***

*int openCourse(instructor_t *instructor, course_t *course);*

➔***Shows the list of the courses which are given by the instructor and the instructor finish the course***

*int finalizeCourse(instructor_t *instructor);*

➔***Shows the list of the courses which are given by the instructor and enrolment student graded for student's course.***

*int assignGrade(instructor_t *instructor);*

➔***Lists the students which are enrolled to the course with their grades.***

*void listGrades(instructor_t *instructor);*

➔***Shows the list of the courses which are given by the instructor and are open***

*void listOpCourse(instructor_t *instructor);*

➔***Function to find automation system user***

*int getUser(char *userID, user_t *user);*

➔***Function to find instructor if user is instructor***

*int getIns(char *password, instructor_t *instructor);*

➔***Function to find student if user is student***

*int getStudent(char *userID, student_t *student);*

➔***Function to find course that will be made an operation about it***

*int getCourse(char *corsID,course_t *course);*

➔***Function to list all course***

*void listCourse(course_t *course);*

➔***Function to list instructors opened courses***

*int listOpenedCourse(instructor_t *instructor);*

➔***Function that indicate two mode that make some operations according to users***

*void menu(int userType, char *userID, int sizeOfcourse);*

➔***Function to control binary files was created or not***

int controlBinFiles(int sizeOfcourse);

FLOW CHART OF PROJECT