



# Testing Diploma Final Project

# 1 Project Objective

The project objective is to test EDGES Reservation software written in C Language using CUnit Framework applying all the techniques/Standards we studied through the Diploma.

## 2 SW Users Description

The software is designed to deal with 2 types of users

### 1. Admin Mohamed Tarek:

This user type is used to manage the Software and can do the following:

- a. Register new user account to the Software
- b. Delete user account from the Software
- c. Check all available Courses Registrations
- d. Print all user's information including login credentials
- e. Print Specific user information including login credentials

This user Type should login at the startup with a secret Token defined in the File **Sec.h**

### 2. Normal User (Student)

This user type is used to interact with the Software and can do the following:

#### a. **Create Account**

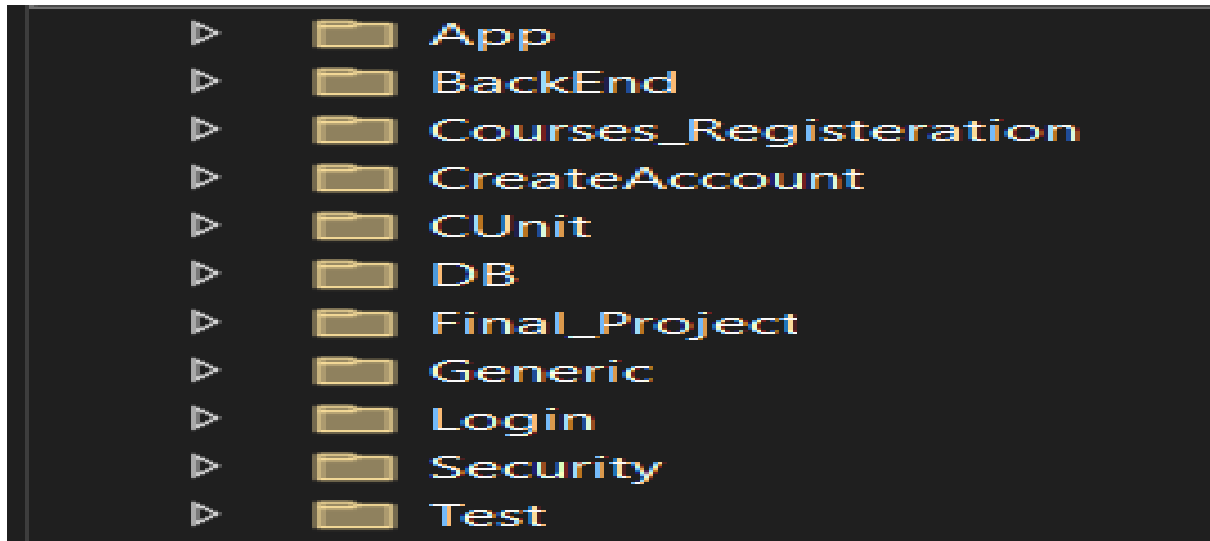
- i. Entering his personal info and login credentials to have an account on the Software

#### b. **Login to Account** (after successful login user can:)

- i. Logout from his Account
- ii. change his password
- iii. see his info (not including username and password)
- iv. Reserve a course from an announced list of courses
- v. Show his Enrolled Courses

### 3 SW Components

The software consists of Multiple SW Components each consists of .c/.h files in a specific folder in the Source code project provided to you:



#### 1. APP

This Component Contains the Main App Logic that mainly Detects the User Type (illustrated in section 2)

##### a. Admin User:

- i. Verifies the credentials of the Admin

Note: To Login as an Admin use the token: **10203040**  
you can change it in the file **Sec.h**

- ii. If successful verification it goes to Admin Page

##### b. Customer User:

- i. Load the User Page to (Login/Create Account)

##### c. Wrong Entry / Admin not verified:

- i. Terminate the Software

## 2. BackEnd

This component is the main logic of the application which is responsible for Running the Logic of:

### a. Admin Page

The home page after Admin Successful verification

```
Welcome Mohamed Tarek
Welcome To Edges Software
Enter your choice and enter Q for Quit
Enter A to add a new user
Enter D to Delete user
Enter C to Check Courses Registrations
Enter P to Print all users
Enter U to Print one user
Enter H for Help
```

### b. Customer Page

The home page if the user type is Customer to Login or Create Account

```
Welcome To Edges Software
Enter your choice and enter Q for Quit
Enter L to Login to your account
Enter C to Create a new account
Enter H for Help
```

### **c. Student Page**

Running the Home Page for the Student after a Successful Login

```
Welcome AhmedJimmy  
Welcome To Your Home Page  
Enter your choice and enter 0 To Logout from your current Account  
Enter C to change your password  
Enter i to see your info  
Enter R To Reserve a course  
Enter S To Show Your Enrolled Courses  
Enter H for Help
```

Also The Backend Component Also responsible for getting the choices from the user, Checking it and verifying it's correctness, then Adding/Removing/Invoking the Data to/from Database Component if needed.

### **3. Course Registration:**

This Component is used by The Backend Component to make a reservation for a student, Show the List of available courses and show the list of courses that a specific student is registered to.

### **4. Create Account**

This component is used by The Backend component to take a form from the User/Admin to create an account after the verification of the Input form.

and if verified it adds / removes the Account from the Database.

### **5. CUnit**

This Component is the CUnit Library Header/Source files

## **6. DB**

This is the Data Base Manager Component which is responsible for Receiving Requests from the Backend Component and making CRUD operations to the Database itself

Note: **CRUD** (**C**reate/**R**ead/**U**ppdate/**D**eleate)

it also contains the DB info which contains the definition of the DB Variables which contains the Data elements (Users info, Login Credentials, ..etc.).

## **7. Final Project**

This Folder Contains the output file of building the whole project

## **8. Generic**

This Component contains the header files contains the declaration of the Data Types used in our Software and the info about the size of users allowed in the SW, the number of Available courses and some other important definitions

## **9. Login**

This component is used by the Backend component to handle the login functionality in the Software it performs the following:

- a. **Detecting User Type**
- b. **Verifying the Admin Token to Login**
- c. **Verifying the User Attempt to Login**

## **10. Security**

This Component Contains the Definition of the Admin Security Token used for Admin Login Verification

## 11. Tests

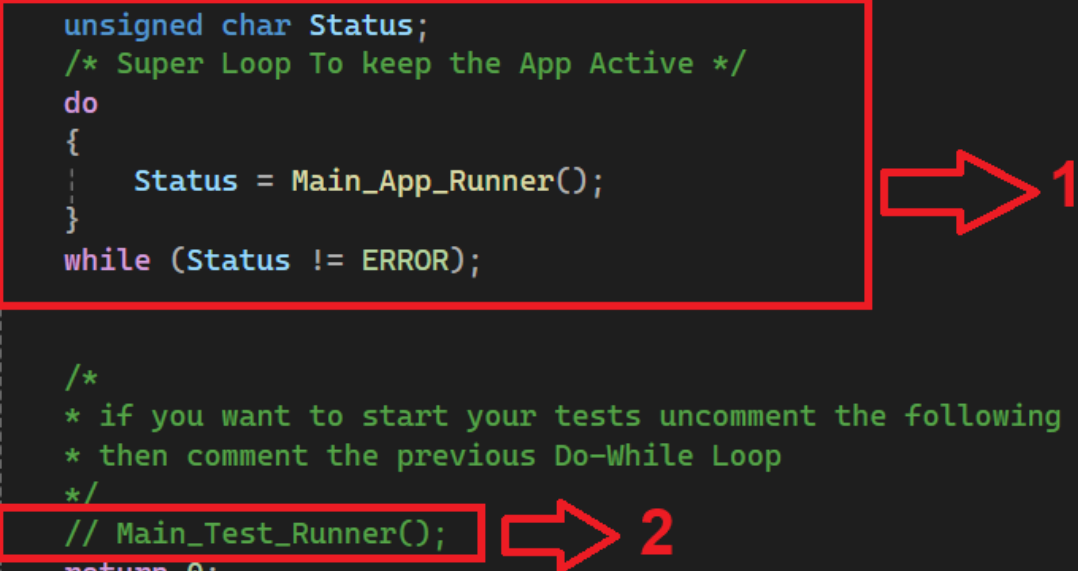
This Component is used to perform the testing on the Application Software it contains initial test cases and test suites.

## 12. Main.c

This is the File that contains the main function that starts the Application or the Test

```
// Main test runner
int main()
{
    /* initialize The DB with some initial Data */
    DBM_initDB();
    /* Variable to hold the status of the Main Application */
    unsigned char Status;
    /* Super Loop To keep the App Active */
    do
    {
        Status = Main_App_Runner();
    }
    while (Status != ERROR);

    /*
     * if you want to start your tests uncomment the following function
     * then comment the previous Do-While Loop
     */
    // Main_Test_Runner();
    return 0;
}
```

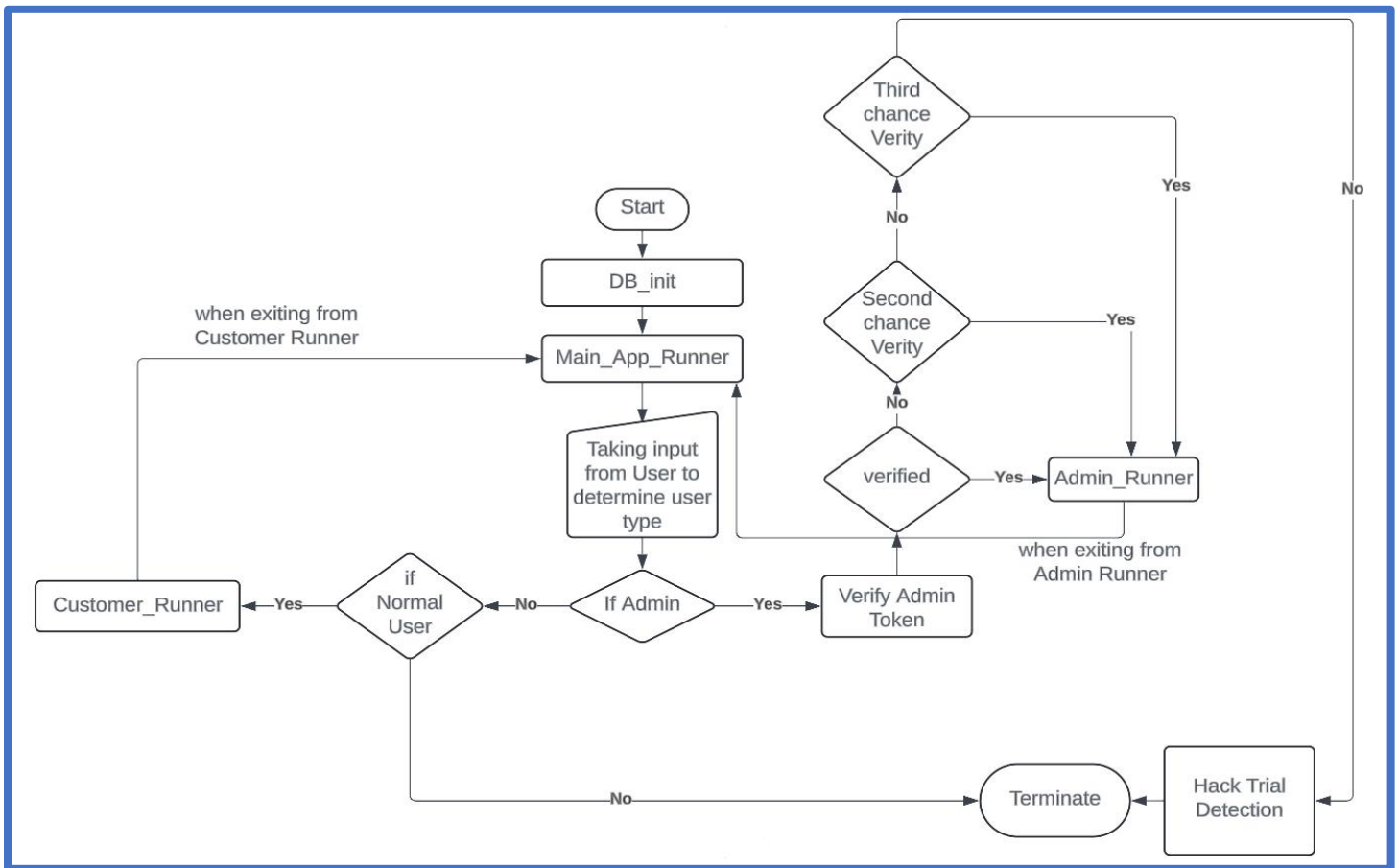


\*) if you want to run the Application itself you need to keep Section one in the previous Picture and comment section 2 (leave the file as it is)

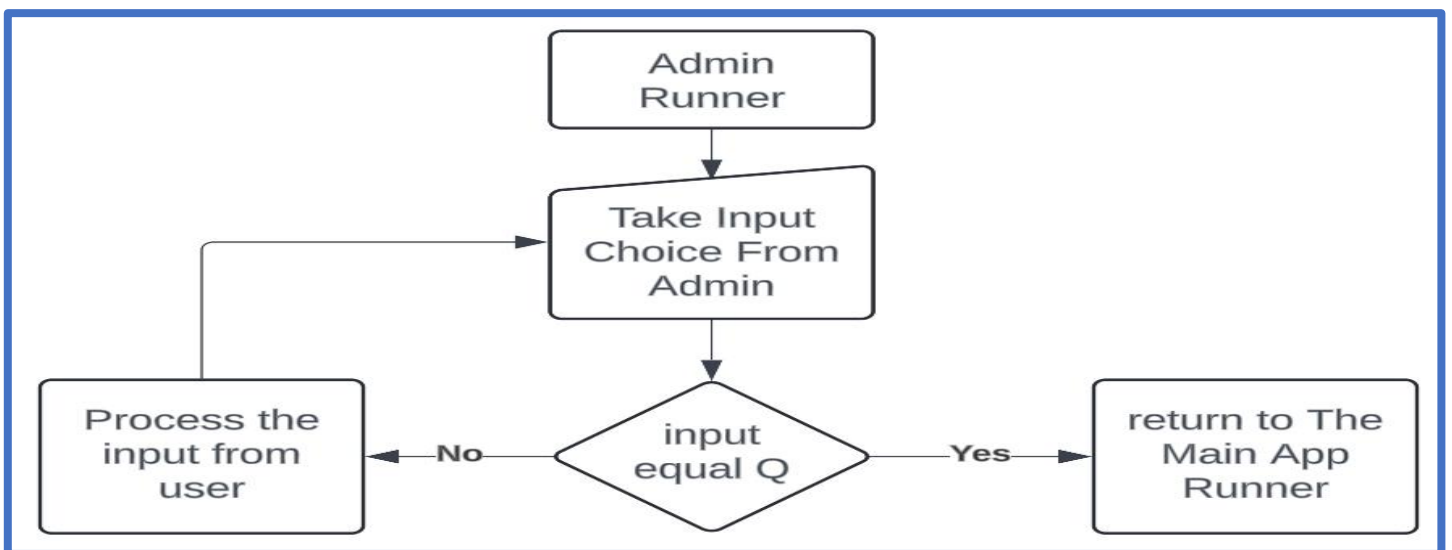
\*) if you want to start the tests you should comment Section one and uncomment section 2

## 4 Important Diagrams

### 1. Flow Chart for the Main App Runner Function

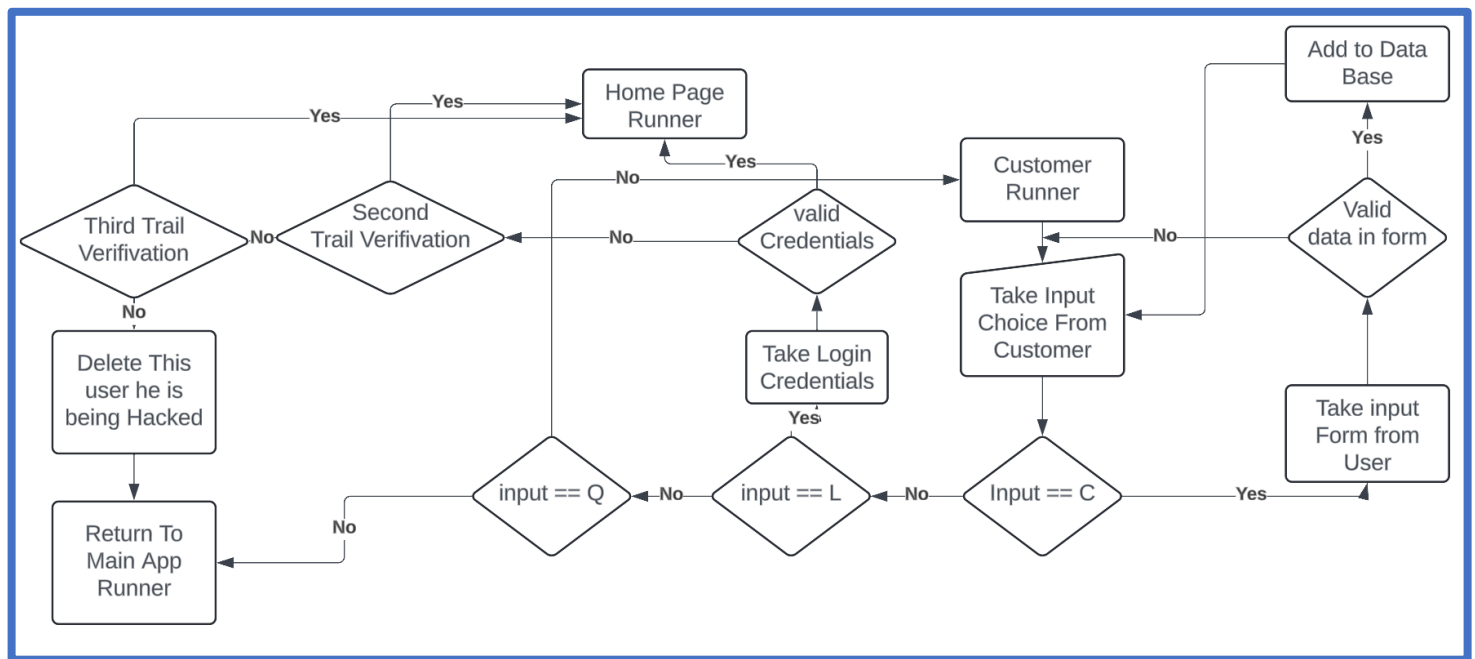


### 2. Flow Chart for the Admin Runner

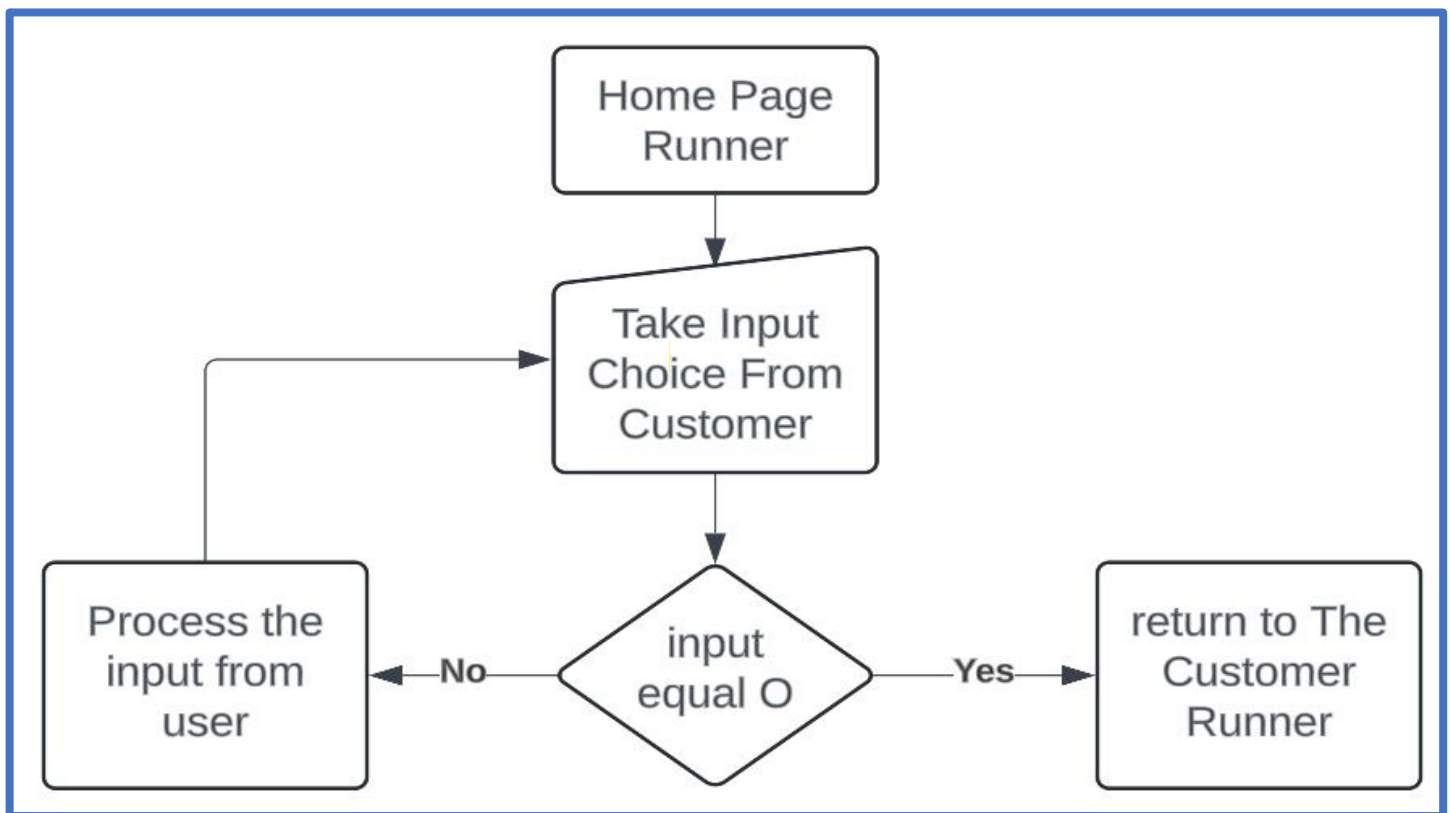




### 3. Flow Chart for the Customer Runner



### 4. Flow Chart For the Home Page Runner



## 5 Important Notes

- The Function **DB\_init** function is used to initialize the Database with 3 Users and make some registrations.
- The Software will never Terminate unless you do either 1 of 3:
  - a. At the main App Runner, you enter value rather than (1,2)
  - b. If you choose an admin and entered the token 3 consecutive times wrong
  - c. If you force closed the terminal 😞
- The Database is alive in the separate Run you can go as a user then log out and go as an admin and the Database should be correct and updated.
- **To login as an Admin:**
  - a. Start the App
  - b. Enter 0
  - c. Press Enter
  - d. Enter the Token: **10203040**
  - e. Press Enter
- **To login as a user you have initially 3 users to use:**
  - a. Username: **AdminUser1**  
Password: **Edges123**
  - b. Username: **AdminUser2**  
Password: **Edges123456**
  - c. Username: **AdminUser3**  
Password: **Edges\_123**

## 6 Important Instructions

- You can test whatever you want but you **Mustn't** update the source code itself at any condition
- You Can Add test code for testing purposes but never update or delete any line of source code
- The Folder Tests that contains the test cases and test suite is your workspace you should update and add tests in as you want
- Run the SW First to explore it and understand the Software itself

## 7 Deliverables

1. You should deliver the project source code updated with the test cases and test suites.
2. You should deliver the test case report for each test cases implemented which describe
  - a. test conditions
  - b. test steps
  - c. test inputs
  - d. expected outputs
  - e. actual outputs
  - f. prerequisites
  - g. Technique used
  - h. and any another information you would like to mentionyou can use the provided template or you own template

3. For the found any bugs, you should deliver a bug report containing
  - a. which test case failed?
  - b. what is the expected output vs the Actual output?
  - c. how to reproduce the issue from the beginning of the SW
  - d. the root cause for these bugs
  - e. suggested fix
4. you should deliver a test completion report contains:
  - a. How many test cases you executed?
  - b. How many test cases failed?
  - c. The coverage achieved for all these techniques:
    - i. (MC/DC-Branch-Statement) coverage
    - ii. State transition coverage
    - iii. Boundary value analysis coverage

**Thank You**  
**EDGES For Training Team**