Nov 2021

Mastering embedded system online diploma by Eng. Keroles Shenouda  
[www.learn-in-depth.com](http://www.learn-in-depth.com)

First Term (Final Project 2)

Student Management System

Eng. Mazen Talaat

My profile: <https://bit.ly/3DUmnyz>

Table of contents

[Problem Statement 2](#_Toc88862769)

[Approach 2](#_Toc88862770)

[Idea 2](#_Toc88862771)

[main.c 3](#_Toc88862772)

[SYS\_API.h 4](#_Toc88862773)

[SYS\_API.c 5](#_Toc88862774)

[1. Add student details manually 5](#_Toc88862775)

[2. Add student details from file 6](#_Toc88862776)

[3. Find the student by the given roll number 7](#_Toc88862777)

[4. Find the student by the given first name 7](#_Toc88862778)

[5. Find the student registered in a course 8](#_Toc88862779)

[6. Count number of students 8](#_Toc88862780)

[7. Delete a student 9](#_Toc88862781)

[8. Update a student 10](#_Toc88862782)

[9. Print data 11](#_Toc88862783)

[Students.txt 11](#_Toc88862784)

[Implemented code output 12](#_Toc88862785)

# Problem Statement

A simple software for student information management system which can perform the following operations:

1. Store first name of the student.
2. Store last name of the student.
3. Store unique roll number for every student.
4. Store GPA for every student.
5. Store courses registered by the student.

# Approach

The idea is to form an individual functions for every operation. All the functions are unified to form software.

1. Add student details from file.
2. Add student details manually.
3. Find the student by the given roll number.
4. Find the student by the given first name.
5. Find the student registered in a course.
6. Count number of students.
7. Delete a student by the given roll number.
8. Update a student by the given roll number.
9. Print all student’s data.
10. Exit the program.

# Idea

The software will consist of 4 files main.c, SYS\_API.c, SYS\_API.h and, Students.txt.

The main.c will contain the interface and calling the function, SYS\_API.c will contain the global variables and the body of the functions, SYS\_API.h will contain the definitions and the prototypes, Students.txt will contain the data.

1. We will have array of struct of 50 elements which is the max number of students.
2. Each struct contains the student details.
3. We will have a global index which refers to the number of students.
4. The index is initialized with zero and can go up to 50.

# main.c

The main.c is just a simple file which consists of:

* Infinite while loop until the choice = 10.
* Dprint and Dscan function(a macro to prinf and fflush to fix eclipse console bug).
* Calling the program functions through switch cases.

Text

Description automatically generated

# SYS\_API.h

The SYS\_API.h consist of:

* Macros to make things easier.
* Struct sinfo which contains the details of a student.
* st[max\_num\_of\_students] which is an array of 50 of struct sinfo.
* FILE \*fp which is a pointer to a typedef FILE to manage reading a file.
* Functions prototypes with their usage.

Diagram

Description automatically generated

Text

Description automatically generated

# SYS\_API.c

The SYS\_API.h consist of:

* Global variable num\_of\_students which is the index of the array st[] to track the data.
* Functions body.

## Add student details manually

* The function will start by checking the list if it is full.
* Takes from user the roll number and checks if it is duplicated.

1. If yes, it will print an error.
2. If no, it will print success and add the student details to the location where the index is pointing to.

* The number of students is then incremented by 1.
* Printing the total, max and remaining number of students through the function tot\_s().

A picture containing text

Description automatically generated

## Add student details from file

* The function will start by checking the list if it is full.
* Reads from the file the details, saves them, and checks if the roll number is duplicated.

1. If yes, it will print that the roll number is already taken and continue to the next line in the file.
2. If no, it will print success and increment the number of students by 1 (index).

* If it reached the EOF(end of file) , a flag will be raised.
* Printing the total, max and remaining number of students through the function tot\_s().

Text

Description automatically generated with low confidence

## Find the student by the given roll number

* The function will start by checking the list if it is empty.
* Searches for the roll number received from user.

1. If found, it will print that the details and return.
2. If not, it will print an error.

Text

Description automatically generated

## Find the student by the given first name

* The function will start by checking the list if it is empty.
* Searches for the first name received from user.

1. If found, it will print that the details and raise a flag.
2. If not, it will print an error.

* It will continue the search until reaching the end of the list.

Text

Description automatically generated

## Find the student registered in a course

* The function will start by checking the list if it is empty.
* Searches for the course ID received from user.

1. If found, it will print that the details and raise a flag.
2. If not, it will print an error.

* It will continue the search until reaching the end of the list and prints the total number of students enrolled.

Text

Description automatically generated

## Count number of students

* The function will print the total number of students by using the index\_number\_of\_students.
* Print the max number by using the macro max\_number\_of\_students.

Text, letter

Description automatically generated

## Delete a student

It consists of two functions: remove\_gap and del\_s.

del\_s:

* The function will start by checking the list if it is empty.
* Searches for the roll number received from user.

1. If found, it will decrease the number of students, calls the function remove\_gap and return.
2. If not, it will print an error.

remove\_gap:

* The function will shift the elements starting from the index of the removed student to the start.

Diagram

Description automatically generated with low confidence

Text

Description automatically generated

## Update a student

* The function will start by checking the list if it is empty.
* Searches for the roll number received from user.

1. If found, it will ask the user to choose what to update and print success if so.
2. If not, it will print an error.

A picture containing graphical user interface

Description automatically generated

## Print data

* The function will start by checking the list if it is empty.
* Loop through each element and print the data.

Graphical user interface, text, application

Description automatically generated

# Students.txt

The format is:

Roll\_Number First\_Name Last\_Name GPA C1\_ID C2\_ID C3\_ID C4\_ID C5\_ID

Text, letter

Description automatically generated

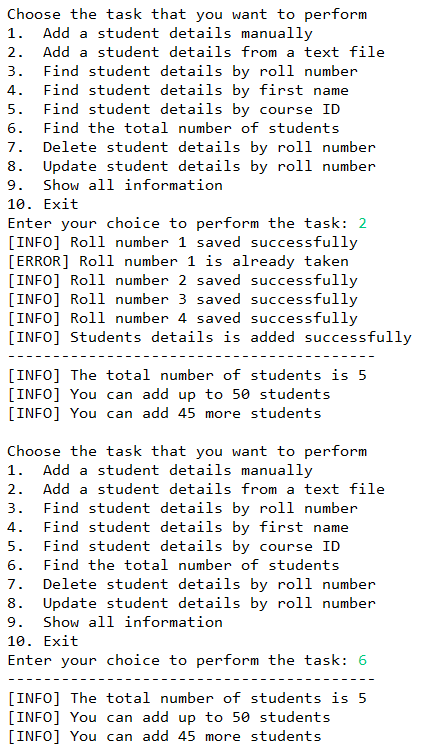
# Implemented code output

Text

Description automatically generated

Text

Description automatically generated



Text

Description automatically generated

Text

Description automatically generated

Text

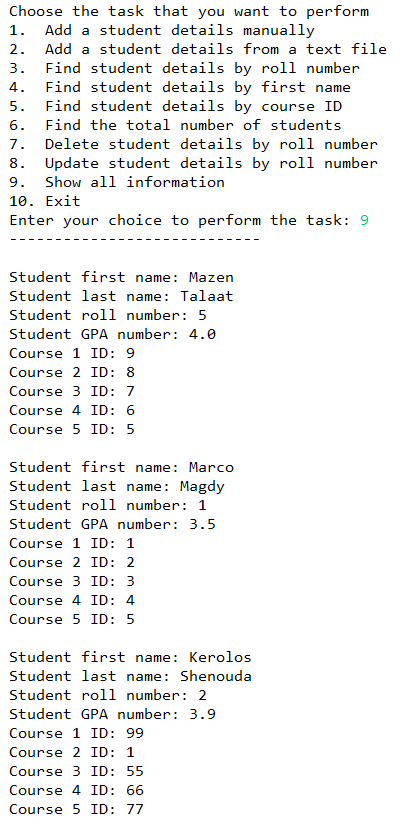
Description automatically generated

Text

Description automatically generated

Text

Description automatically generated



Text, letter

Description automatically generated

Text

Description automatically generated

