EDGES

OUR EDGE IS YOUR SUCCESS

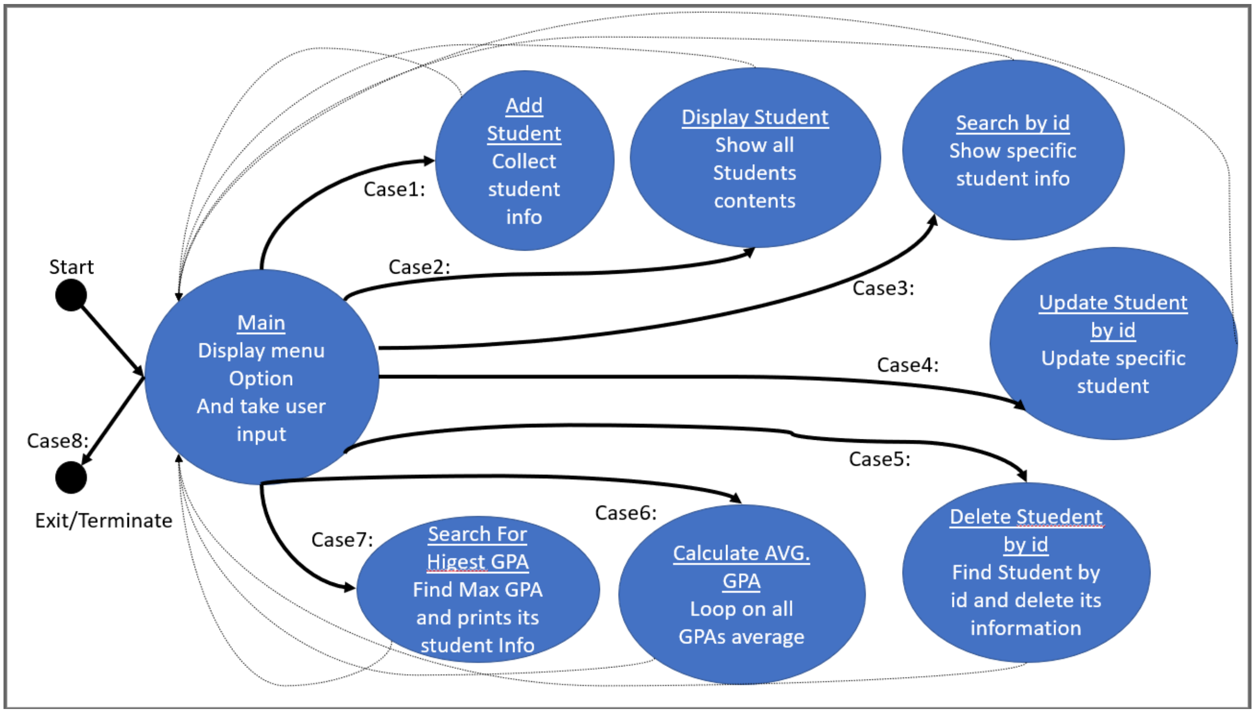
Edges For Training Standard Embedded Systems Diploma C Project

Student Management System

The Student Management System is a C program designed to manage student information efficiently. It allows the user to perform various operations such as adding new students, displaying all students, searching for a student by ID, updating student details, calculating the average GPA, and finding the student with the highest GPA.

This system utilizes fundamental programming concepts including conditions, loops, strings, structs, and pointers, besides the ability to use data structures efficiently like the linked list.

State Machine



Students Data

All student’s data shall be stored consecutively in a linked list.

Each node of the list is a structure containing the structure of the data needed to be stored of each student and a pointer that points to the next node in the linked list.

```
struct student
{
    int id;             /* Student ID */
    char name[50];      /* Student Name */
    int age;            /* Student Age */
    float gpa;          /* Student GPA */
};

/* Define a structure to represent a node in a linked list */
struct node
{
    struct student data;
    struct node *next;
};

struct node *head = NULL;
```

Function Descriptions and Techniques Used:

int main(void)

The main function serves as the entry point of the program. It continuously displays the menu options to the user, takes the user's choice as input, and calls the corresponding function based on the user's selection. The menu loop continues until the user decides to exit the program.

Menu Options:

- **Add a Student:** Prompts the user for new student details and adds them to the system.
- **Display All Students:** Shows a list of all stored student records.
- **Search for a Student by ID:** Allows the user to find a specific student using their unique ID.
- **Update Student Information**: Facilitates the updating of details for an existing student identified by their ID.
- **Delete a Student:** Removes a student's record based on their ID.
- **Calculate Average GPA**: Computes and displays the average GPA of all students.
- **Find Student with Highest GPA**: Identifies and displays the student with the highest GPA.
- **Exit:** Terminates the program.

*void addStudent(const struct student *const ptr)*

This function collects new student details from the user and adds them to the student linked list.

Description:

- It adds a new student to the linked list. It first checks if the student's ID already exists.
- If the ID exists, it prints an error message and returns. If the ID does not exist, it attempts to allocate memory for a new node.
- If memory allocation fails, it prints an error message and returns.
- If memory allocation is successful, it copies the student data to the new node and adds the node to the end of the linked list.

void displayStudents(void)

This function displays the details of all students currently in the system.

Description:

- Displays all students in the linked list.
- If the list is empty, it prints a message indicating no students are present. Otherwise, it traverses the list and prints the details of each student.

void searchStudentByID(int id)

This function searches for a student by their ID and displays their details if found.

Description:

- It searches for a student by their ID in the linked list.
- If the student is found, it prints their details.
- If the student is not found, it prints a message indicating the student was not found.

void updateStudent(int id)

This function updates the details of a student with the given ID.

Description:

- Updates the details of a student with a given ID in the linked list.
- If the student is found, then update the student's information (student name, age, gpa).
- If the student is not found, it prints a message indicating the student was not found.

float calculateAverageGPA(void)

This function calculates and returns ethe average GPA of all students.

Description:

- Calculates and returns the average GPA of all students in the linked list.
- It traverses the list to sum the GPAs of all students and counts the number of students.
- If the list is empty, it returns 0.0.
- Otherwise, it computes and returns the average GPA by dividing the total sum of GPAs by the number of students.

void searchHighestGPA(void)

This function finds and displays the student with the highest GPA.

Description:

- Searches for the student with the highest GPA in the linked list.
- It traverses the list, keeping track of the student with the highest GPA encountered.
- If the list is empty, it prints a message indicating that there are no students.
- After finding the student with the highest GPA, it prints the details of that student.

void deleteStudent(int id)

This function deletes a student from the linked list by their ID.

Description:

- Iterates through the linked list of students starting with the Head of it to find the student with the given ID.
- If the student is found, it adjusts the pointers to remove the node from the list, frees the memory allocated for the node, and prints a success message.
- If the student is not found, it prints a message indicating the student was not found.

Video to demonstrate the project behavior and outputs:

https://youtu.be/SlGfWfT15eI

Thank You

Edges For Training Team

Engineer / Mohamed Tarek