# **STUDENT MANAGEMENT SYSTEM**

**A PROJECT REPORT**

*Submitted by*

**M NAMRATHA [Reg No:-RA2111003010240]**

**Y VIJAYRAM [Reg No:-RA2111003010232]**

*Submitting to*

**Dr. M. Eliazer**

(Asst Professor, Department of Computer Science and Engineering)

**BACHELOR OF TECHNOLOGY**

**Logo, company name

Description automatically generated**

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING FACULTY OF ENGINEERING AND TECHNOLOGY SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR- 603 203

**June 2022**

1. Objective

Schools and Universities are the foundation of knowledge and an educational body upon which students rely. Therefore, they need to maintain a proper database of its students to keep all the updated records and easily share information with students.

Most schools and Universities count on an advanced software tool known as **'Student Information System (SIS)'** to keep all their student records like Name, Roll Numbers, Phone Numbers, Percentage

Over the recent years, the performance and efficiency of the education industry have been enhanced by using the **Student Management System**. This tool has productively taken over the workload of the Class in charge with its well-**organized, easy, and reliable adding, Deleting, Searching, Updating**

**"Student Management System (SMS) is a Code that is designed to Add, Delete, Update Display Send arch all the data generated by a School or college, including the grades of a student, their name Phone Number, Percentage”**

1. Problem Definition

The school / College Management must handle records for the number of students and maintain their records and update their records regularly. But it takes more time if we do it on paper or in books Hence we need an upgradation this code which we wrote can easily Add, Delete, Update, Search, and display the information of the students whenever required

Existing System

All the details of the student are Maintained in a Single Record

So Searching and Updating the Details is a tedious task Also it has more time and space complexity

Proposed System

The Code which written can easily Add, Delete, Update, Search, and Display the information of the students whenever required

Easy to Handle / Fast and Convenient

**Flow Chart**

**If** Choice 3 is Selected Data is deleted using a linked list

Select the Choice

If Choice-1 Selected For Insert using linked list data is loaded

If Choice 5 is Selected All the available present is displayed using this choice

If Choice 4 is Selected If data is present then print “Record Found” If not print “Record Not Found

If Choice 2 is Selected, if data is present then it will update

Choices are assigned against Operations

Write Function for the update, Display, Delete, Insert, Search Separately

Define the value of the Phone number, Roll Number, Percentage, Name

Declare struct Student

4) Algorithm

# **Problem** – Design a **STUDENT MANAGEMENT SYSTEM** Tool that helps Teachers to Update, Add, Delete, and Display details of Students

**Step 1** − START

**Step 2** − Declare struct Student

**Step 3** − Define the value of the Phone number, Roll Number, Percentage, Name

**Step 4** – Write Function for the update, Display, Delete, Insert, Search Separately

**Step 5** – Choices are assigned against Operations

**Step 6** – Select the Choice

**Step 7**– If Choice-1 Selected For Insert using linked list data is loaded

**Step 8 –** If Choice 2 is Selected, if data is present then it will update

**Step 9 – If** Choice 3 is Selected Data is deleted using a linked list

**Step 10** – If Choice 4 is Selected If data is present then print “Record Found” If not print “Record Not Found”

**Step 11** – If Choice 5 is Selected All the available present is displayed using this choice

**Step 12 –** Go to Step 6

**Step 10** − STOP

* **Search** − Search an item in a data structure.
* **Insert** − Insert item in a data structure.
* **Update** − Update an existing item in a data structure.
* **Delete** − delete an existing item from a data structure.
* **Display** – Display All the Items present in Data Structure

5) Source Code:-

**#include<stdlib.h>**

**#include<string.h>**

**#include<stdio.h>**

**struct Student**

**{**

**int rollnumber;**

**char name[100];**

**char phone[100];**

**float percentage;**

**struct Student \*next;**

**}\* head;**

**void insert(int rollnumber, char\* name, char\* phone, float percentage)**

**{**

**struct Student \* student = (struct Student \*) malloc(sizeof(struct Student));**

**student->rollnumber = rollnumber;**

**strcpy(student->name, name);**

**strcpy(student->phone, phone);**

**student->percentage = percentage;**

**student->next = NULL;**

**if(head==NULL){**

**head = student;**

**}**

**else{**

**student->next = head;**

**head = student;**

**}**

**}**

**void search(int roll number**

**{**

**struct Student \* temp = head;**

**while(temp!=NULL){**

**if(temp->rollnumber==rollnumber){**

**printf("Roll Number: %d\n", temp->rollnumber);**

**printf("Name: %s\n", temp->name);**

**printf("Phone: %s\n", temp->phone);**

**printf("Percentage: %0.4f\n", temp->percentage);**

**return;**

**}**

**temp = temp->next;**

**}**

**printf("Student with roll number %d is not found !!!\n", rollnumber);**

**}**

**void update(int rollnumber)**

**{**

**struct Student \* temp = head;**

**while(temp!=NULL){**

**if(temp->rollnumber==rollnumber){**

**printf("Record with roll number %d Found !!!\n", rollnumber);**

**printf("Enter new name: ");**

**scanf("%s", temp->name);**

**printf("Enter new phone number: ");**

**scanf("%s", temp->phone);**

**printf("Enter new percentage: ");**

**scanf("%f",&temp->percentage);**

**printf("Updation Successful!!!\n");**

**return;**

**}**

**temp = temp->next;**

**}**

**printf("Student with roll number %d is not found !!!\n", rollnumber);**

**}**

**void Delete(int rollnumber)**

**{**

**struct Student \* temp1 = head;**

**struct Student \* temp2 = head;**

**while(temp1!=NULL){**

**if(temp1->rollnumber==rollnumber){**

**printf("Record with roll number %d Found !!!\n", rollnumber);**

**if(temp1==temp2){**

**head = head->next;**

**free(temp1);**

**}**

**else{**

**temp2->next = temp1->next;**

**free(temp1);**

**}**

**printf("Record Successfully Deleted !!!\n");**

**return;**

**}**

**temp2 = temp1;**

**temp1 = temp1->next;**

**}**

**printf("Student with roll number %d is not found !!!\n", rollnumber);**

**}**

**void display()**

**{**

**struct Student \* temp = head;**

**while(temp!=NULL){**

**printf("Roll Number: %d\n", temp->rollnumber);**

**printf("Name: %s\n", temp->name);**

**printf("Phone: %s\n", temp->phone);**

**printf("Percentage: %0.4f\n\n", temp->percentage);**

**temp = temp->next;**

**}**

**}**

**int main()**

**{**

**head = NULL;**

**int choice;**

**char name[100];**

**char phone[100];**

**int rollnumber;**

**float percentage;**

**printf("1 to insert student details\n2 to search for student details\n3 to delete student details\n4 to update student details\n5 to display all student details");**

**do**

**{**

**printf("\nEnter Choice: ");**

**scanf("%d", &choice);**

**switch (choice)**

**{**

**case 1:**

**printf("Enter roll number: ");**

**scanf("%d", &rollnumber);**

**printf("Enter name: ");**

**scanf("%s", name);**

**printf("Enter phone number: ");**

**scanf("%s", phone);**

**printf("Enter percentage: ");**

**scanf("%f", &percentage);**

**insert(rollnumber, name, phone, percentage);**

**break;**

**case 2:**

**printf("Enter roll number to search: ");**

**scanf("%d", &rollnumber);**

**search(rollnumber);**

**break;**

**case 3:**

**printf("Enter roll number to delete: ");**

**scanf("%d", &rollnumber);**

**Delete(rollnumber);**

**break;**

**case 4:**

**printf("Enter roll number to update: ");**

**scanf("%d", &rollnumber);**

**update(rollnumber);**

**break;**

**case 5:**

**display();**

**break;**

**}**

**} while (choice != 0);**

**}**

**6) Output Screen**

**Adding the Student**

**Text

Description automatically generated**

**Displaying All students**

**Text

Description automatically generated**

**Deleting Roll no :-240**

**Text

Description automatically generated**

**As 240 is deleted after searching**

**Text

Description automatically generated**