****

**Green University of Bangladesh**

**Department of Computer Science and Engineering (CSE)**

**Faculty of Sciences and Engineering**

**Semester: (Spring, Year:2021), B.Sc. in CSE (Day/Eve)**

**Course Title:CSE Lab**

**Course Code: CSE 106 Section:DD**

**Lab Project Name: STUDENT MANAGEMENT SYSTEM**

**Student Details**

| **Name** | **ID** |
| --- | --- |
| **MD ABU UBAIDA JUBAER SAYEB** | **213902113** |

**Submission Date: 9/10/2022**

**Course Teacher’s Name: Md. Sultanul Islam Ovi**

**[For Teachers use only: Don’t Write Anything inside this box]**

| **Lab Project Status**  **Marks: ………………………………… Signature: .....................**  **Comments: .............................................. Date: ..............................** |
| --- |

Table of Contents

**Chapter 1 Introduction**

1.1 Introduction

1.2 Design Goals/Objective

**Chapter 2 Design/Development/Implementation of the Project**

2.1 Section (Choose the name of this section as appropriate with your project)

2.2 Section (Choose the name of this section as appropriate with your project)

2.2.1 Subsection

**Chapter 3 Performance Evaluation**

3.1 Simulation Environment/ Simulation Procedure

3.2 Results and Discussions

**Chapter 4 Conclusion**

4.1 Introduction

4.1 Practical Implications

4.2 Scope of Future Work

**References**

# Chapter 1 Introduction

## Introduction

Our main focus is to design a unique Student Management system System that will improve Data management in Institutes experience for both Students and the Administration authorities. The whole system will run on the internet. The system is written in C using data structure. Users will have the ability to log in from any place with internet connection. After that they will be able to do various tasks that are designed for them. Student Management System is software which is helpful for students as well as the school authorities. In the current system all the activities are done manually.Our Student Management System deals with the various activities related to the students.

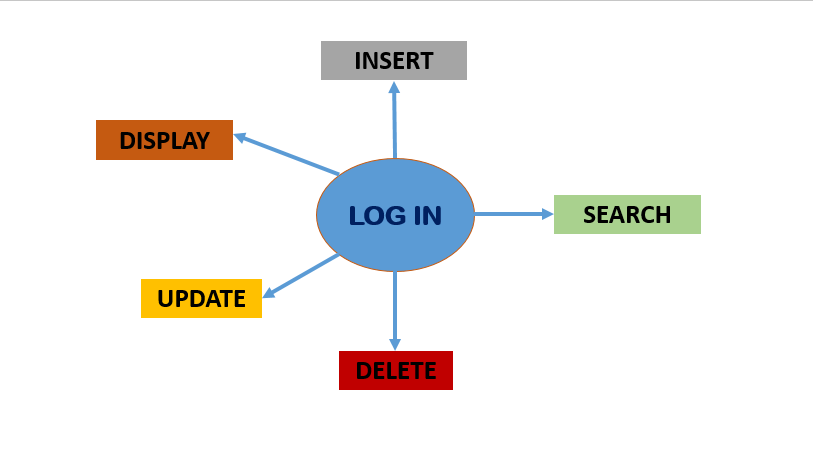
## Design Goals/Objective

* **Student Management system is a management information system for education establishments to manage student data.**
* **It provides capabilities for registering students in courses,Admitting students, tracking student attendance and Submitting students documents.**
* **Ensure data integrity, privacy, and security in an open-access environment.**

# Chapter 2

# Design/Development/Implementation of the Project

## Front Screen Design



## PSEUDO CODE

## HEADER FILES & VARIABLES

| **#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;** |
| --- |

## BINARY SEARCH FUNCTION

| **//binary search**  **struct Student\* middle(struct Student\* start,struct Student\* last)**  **{**  **if (start == NULL)**  **return NULL;**  **struct Student\* slow = start;**  **struct Student\* fast = start -> next;**  **while (fast != last)**  **{**  **fast = fast -> next;**  **if (fast != last)**  **{**  **slow = slow -> next;**  **fast = fast -> next;**  **}**  **}**  **return slow;**  **}**  **struct Student\* binarySearch(struct Student \*head, int rollnumber)**  **{**  **struct Student\* start = head;**  **struct Student\* last = NULL;**  **do**  **{**  **struct Student\* mid = middle(start, last);**  **if (mid == NULL)**  **return NULL;**  **if (mid -> rollnumber == rollnumber)**  **return mid;**  **else if (mid -> rollnumber < rollnumber)**  **start = mid -> next;**  **else**  **last = mid;**  **} while (last == NULL ||**  **last != start);**  **return NULL;**  **}** |
| --- |

## INSERT

| **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){**  **// if head is NULL**  **// set student as the new head**  **head = student;**  **}**  **else{**  **// if list is not empty**  **// insert student in beginning of head**  **student->next = head;**  **head = student;**  **}**  **}** |
| --- |

## SEARCH

| **void search(int rollnumber)**  **{**  **struct Student \* temp = head;**  **if(binarySearch(head,rollnumber)==NULL)**  **{**  **printf("Data not found");**  **}**  **else**  **{**  **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;**  **}**  **}** |
| --- |

## UPDATE

| **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);**  **}** |
| --- |

## DELETE

| **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){**  **// this condition will run if**  **// the record that we need to delete is the first node**  **// of the linked list**  **head = head->next;**  **free(temp1);**  **}**  **else{**  **// temp1 is the node we need to delete**  **// temp2 is the node previous to temp1**  **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);**  **}** |
| --- |

## DISPLAY

| **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;**  **}**  **}** |
| --- |

## SORTING

| **void sort()**  **{**  **struct Student \*curNode,\*nextNode;**  **curNode = head;**  **while(curNode!=0)**  **{**  **nextNode = curNode->next;**  **while(nextNode!=0)**  **{**  **if(curNode->rollnumber>nextNode->rollnumber)**  **{**  **swap1(&curNode->rollnumber, &nextNode->rollnumber);**  **swap2(curNode->name, nextNode->name);**  **swap2(curNode->phone, nextNode->phone);**  **swap3(&curNode->percentage, &nextNode->percentage);**  **}**  **nextNode = nextNode->next;**  **}**  **curNode = curNode->next;**  **}**  **printf("\nRecord is Now Sorted\n");**  **}** |
| --- |

## SWAP

| **void swap1( int \*a, int \*b)**  **{**  **int temp;**  **temp = \*a;**  **\*a = \*b;**  **\*b = temp;**  **}**  **void swap3(float \*x, float \*y)**  **{**  **float temp;**  **temp = \*x;**  **\*x = \*y;**  **\*y = temp;**  **}**  **void swap2(char \*str1, char \*str2)**  **{**  **char \*temp = (char \*)malloc((strlen(str1) + 1) \* sizeof(char));**  **strcpy(temp, str1);**  **strcpy(str1, str2);**  **strcpy(str2, temp);**  **free(temp);**  **}** |
| --- |

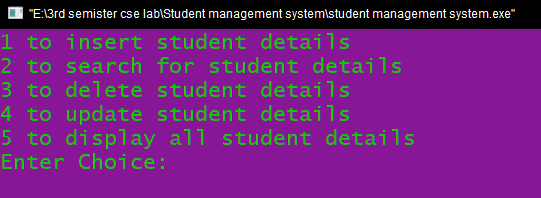
## MAIN FUNCTION

| **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);**  **}** |
| --- |

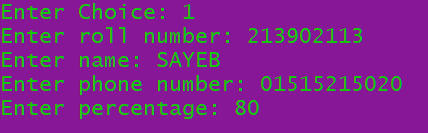
# Chapter 3 Performance Evaluation

## OUTPUT

**ENTER CHOICE**



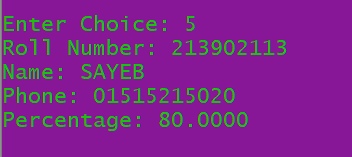
**INSERT**

****

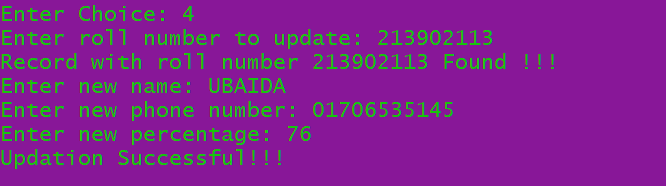
**SEARCH**

## 

**DISPLAY**

****

**UPDATE**

****

**DELETE**

****

## Results and Discussions

**Analysis and Outcome**

**By using student management system we can easily insert a students information,we can search the inserted student information whenever we want ,if we have given the wrong information then we change it and update it,we can display all the information,if the student leaves then we can delete those informations**

# Chapter 4 Conclusion

**Student management systems make faculty jobs more accessible by giving them an easy place to find and sort information. This system allows teachers and student managers to follow with their student engagement. The idea is to create a scenario that makes the lives of administration and teachers easier.**

## Scope of Future Work

**In future whenever we need the students details we can get from here,it is easy to get student data for a institution with a huge amount of student,we just need to search the student ID and we will get the whole information**

# References

1. **Used google to get the basic structure of my code.**
2. **Got the idea for slideshare.com**