# Report:
# GRADE SHEET MANAGEMENT SYSTEM

LTTS
GLOBAL ENGINEERING ACADEMY

L&T Technology Services

Version Number:
Team Members :
Team No:
Module: Model Based System Engineering

# DECLARATION

I hereby declare that the attached documents are correct and valid to the best of my knowledge.

**BALAJI S B**

# ABSTRACT:

Grade sheet management is a project that manages and stores grade information electronically according to student's score. The system helps both students and examination manager to keep a constant track of all semester grades and marks. It allows both the admin and the student to search for the desired marksheet.
The project titled Grade Sheet Management System is Grade Management software for monitoring and controlling the transactions in an Examination cell. The project “Grade sheet Management System” is developed in C, which mainly focuses on basic operations in an Examination cell like adding new grade sheets, and searching grade sheets and displaying all the sheets.
This project “GRADE SHEET MANAGEMENT SYSTEM” gives us the complete information about the Examination cell. We can enter the record of new sheets and retrieve the details of sheets available in the Examination cell. We can issue the sheets to the students and maintain their records.

# IMPLEMENTATION:

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

struct  Subject
{
   char code[50];
   char name[50];
   char grade[10];
   int grade_point;
   int credit;
};

struct Student
{
   char name[50];
   char rollno[50];
   char date[50];
   int sem;
   int total_subjects;
   double sgpa;
   struct Subject subject[50];
};

int returnGradePoint(char grade[]){
   if (!strcmp(grade, "O"))
   {
      return 10;
   }
   else if (!strcmp(grade, "A"))
   {
      return 9;
   }
   else if (!strcmp(grade, "B"))
   {
      return 8;
   }
   else if (!strcmp(grade, "C"))
   {
      return 7;
```

```
    }
    else if (!strcmp(grade, "D"))
    {
        return 6;
    }
    else if (!strcmp(grade, "E"))
    {
        return 5;
    }
    else
    {
        return 0;
    }
}

// function to generate mark sheet
void gen_ms(struct Student student)
{
    printf("\n\n");
    printf("\t ABC College of Technology");
    printf("\n\t ------------------");
    printf("\n Name:%s \t Roll no.:%s", student.name, student.rollno);
    printf("\n Date:%s \t Semester:%d", student.date, student.sem);
    printf("\n-----------------------------------------");
    printf("SUBJECT CODE \t\t");
    printf("SUBJECT NAME \t\t");
    printf("GRADE \t\t");
    printf("GRADE POINT \t\t");
    printf("TOTAL CREDITS \t\t");
    printf("-----------------------------------------");
    printf("\n\n");
}
void gen_ms_mid(struct Student student)
{
    printf("\n                                    ");
    for(int i=0; i<student.total_subjects; i++){
        struct Subject sub = student.subject[i];
        printf("%s\t\t", sub.code);
        printf("%s\t\t", sub.name);
        printf("%s\t\t", sub.grade);
        printf("%d\t\t", sub.grade_point);
        printf("%d\t\t", sub.credit);
```

```
        printf("\n");
    }
}

void gen_ms_foot(double sgpa)
{
    printf("\n");
    printf("-----------------------------------------\n");
    printf("SGPA:%.2f\n", sgpa);
    printf("-----------------------------------------\n");
}

// Save data to a file

void saveData(FILE *fp, struct Student student){
    fprintf(fp, "%s\n", student.name);
    fprintf(fp, "%s\n", student.rollno);
    fprintf(fp, "%s\n", student.date);
    fprintf(fp, "%d\n", student.sem);
    fprintf(fp, "%d\n", student.total_subjects);
    for(int i=0; i<student.total_subjects; i++){
        fprintf(fp, "%s %s %s %d %d\n", student.subject[i].code, student.subject[i].name, student.subject[i].grade,
student.subject[i].grade_point, student.subject[i].credit);
    }
    fprintf(fp, "%.2f\n\n", student.sgpa);
}

int readData(FILE *fp, struct Student students[]){
    int idx=0;
    char line[200];

    while(fgets(line, 200, fp)){
        line[strlen(line) - 1] = 0;
        strcpy(students[idx].name, line);

        fgets(line, 200, fp);
        line[strlen(line) - 1] = 0;
        strcpy(students[idx].rollno, line);

        fgets(line, 200, fp);
        line[strlen(line) - 1] = 0;
        strcpy(students[idx].date, line);
```

```
        fscanf(fp, "%d", &students[idx].sem);
        fscanf(fp, "%d", &students[idx].total_subjects);

        for (int i = 0; i < students[idx].total_subjects; i++)
        {
            fscanf(fp, "%s", students[idx].subject[i].code);
            fscanf(fp, "%s", students[idx].subject[i].name);
            fscanf(fp, "%s", students[idx].subject[i].grade);
            fscanf(fp, "%d", &students[idx].subject[i].grade_point);
            fscanf(fp, "%d", &students[idx].subject[i].credit);

        }
        fgets(line, 200, fp);

        fgets(line, 200, fp);

        students[idx].sgpa = (double)atof(line);

        fgets(line, 200, fp);
        idx++;
    }
    return idx;
}

void displayAllStudents(struct Student students[], int std_count){
    for(int idx=0; idx<std_count; idx++){
        gen_ms(students[idx]);
        gen_ms_mid(students[idx]);
        gen_ms_foot(students[idx].sgpa);
    }
}

void searchStudent(struct Student students[], int std_count, char rollno[]){
    int isFound = 0;
    for(int idx=0; idx<std_count; idx++){
        if(!strcmp(students[idx].rollno, rollno)){
            isFound = 1;
            gen_ms(students[idx]);
            gen_ms_mid(students[idx]);
            gen_ms_foot(students[idx].sgpa);
            break;
```

```
        }
    }
    if(!isFound){
        printf("No Record Found\n");
    }
}

int main(){
    struct Student student;
    struct Student students[50];
    int opt, total_students;

    FILE *fp = fopen("./student.txt", "a+");

    total_students = readData(fp, students);

    printf("\t==========  GRADE SHEET ==========");
    printf("\n");
    printf("\nSelect your option");
    printf("\n1. Generation of Grade sheet");
    printf("\n2. Display total Grade sheets");
    printf("\n3. Search Grade sheet");
    printf("\n4. Exit");
    printf("\n");
    printf("\n\t==========  GRADE SHEET ==========");
    printf("\n");
    printf("Enter an option and continue\t");
    scanf("%d", &opt);
    fgetc(stdin);

    switch(opt){
        case 1:
            printf("\nEnter the Name: ");
            fgets(student.name, 50, stdin);
            student.name[strlen(student.name) - 1] = 0;

            printf("\nEnter the Roll no.: ");
            fgets(student.rollno, 50, stdin);
            student.rollno[strlen(student.rollno) - 1] = 0;


            strcpy(student.date, __DATE__);
```

```
    printf("\nEnter the Semester: ");
    scanf("%d", &student.sem);

    printf("\nEnter the Number of subjects: ");
    scanf("%d", &student.total_subjects);
    fgetc(stdin);

    double value=0, tot_credit=0;
    for(int i=0; i<student.total_subjects; i++){
      printf("\n\n");

      printf("Enter the Subject code: ");
      fgets(student.subject[i].code, 20, stdin);
      student.subject[i].code[strlen(student.subject[i].code) - 1] = 0;

      printf("Enter the Subject Name: ");
      fgets(student.subject[i].name, 20, stdin);
      student.subject[i].name[strlen(student.subject[i].name) - 1] = 0;

      printf("Please enter the Grade(in Caps): ");
      fgets(student.subject[i].grade, 10, stdin);
      student.subject[i].grade[strlen(student.subject[i].grade)  - 1] = 0;

      printf("Please enter the Credit points: ");
      scanf("%d", &student.subject[i].credit);
      fgetc(stdin);

      student.subject[i].grade_point = returnGradePoint(student.subject[i].grade);

      value += student.subject[i].credit * student.subject[i].grade_point;
      tot_credit += student.subject[i].credit;
    }
    student.sgpa = (double)value / tot_credit;
    gen_ms(student);
    gen_ms_mid(student);
    gen_ms_foot(student.sgpa);
    saveData(fp, student);
    break;

  case 2:
    displayAllStudents(students, total_students);
```

```
            break;
        case 3:;
            char rollNo[50];
            printf("\nEnter the Roll no.: ");
            fgets(rollNo, 50, stdin);
            rollNo[strlen(rollNo) - 1] = 0;
            searchStudent(students, total_students, rollNo);
            break;

    }
    fclose(fp);
    return 0;
}
```

# RESULT:

The desired output is obtained for the above code and executed correctly.