

Table of Contents

Abstract	i
1. Introduction	1
1.1 Background and objectives	1
1.2 Overview of the structure of the report	1
2. Project Description	3
2.1 Purpose and scope	3
2.2 Problem Description	3
2.3 Functionality and Features	3
3. Methodology	4
3.1 Development Process	4
3.2 Tools and Resources	4
3.3 Development Environment	4
4. Results and Discussion	5
4.1 Presentation of Project Outcomes and Achievements	5
4.2 Comparison of Actual Results with Expected Outcomes	5
4.3 Discussion of Challenges Faced and Lessons Learned	6
5. Conclusion	7
6. Future Work	8
6.1 Suggestion for Future Enhancements	8
6.2 Ideas for Additional Improvements	8
7. References	9
8. Appendices	10
8.1 Source Code	10
8.2 Output	26

Abstract

This report gives an explanation about a Grade Sheet Management System, which is a software application that streamlines the process of managing student grade sheets at educational institutions. This application, written in the C programming language, allows users to generate, display, search, and delete grade sheets. The system has a user-friendly interface that allows users to enter and save student information like as registration numbers, names, faculties, semesters, topic titles, and grades obtained. The application computes grades based on the marks earned and generates a full grade sheet with detailed information about each student's academic performance. File handling technologies are used to effectively store and retrieve student records. The system intends to improve the efficiency and accuracy of grade sheet management operations, resulting in increased organizational effectiveness within educational institutions. This study summarizes the Grade Sheet Management System, including its architecture, functions, implementation details, and prospective areas for future improvement.

1. Introduction

The Grade Sheet Management System project automates grade sheet management in educational institutions, addressing the issues that come with it. Traditionally, grade sheet management was based on error-prone systems such as paper records and spreadsheets, resulting in inefficiencies and mistakes. This project intends to streamline grade sheet administration operations, increase productivity, and boost organizational performance.

1.1 Background and objectives

In academic administration, the management of student grade sheets is critical to ensuring smooth and efficient operations within educational institutions. Traditionally, this procedure was primarily reliant on manual means, such as paper-based records and spreadsheets. The limits of manual grade sheet management, such as inaccuracies, inefficiencies, and logistical issues, have become more apparent as educational institutions grow in size and complexity.

In light of these issues, the Grade sheet Management System project seeks to transform grade sheet management processes through automation and technology. The project's goal is to streamline grade sheet management operations, increase productivity, and improve organizational effectiveness inside educational institutions by creating a user-friendly software application.

This report was prepared to meet the following objectives:

- Give an overview of the Grade sheet Management System.
- Describe the requirement for automated grade sheet management.
- Discover the system's purpose and relevance.
- Explain system functionalities.
- Address possibilities for future growth and development.

1.2 Overview of the structure of the report

- a) Introduction:** Provides background information on the Grade sheet Management System project, outlining its objectives and setting the report's context.
- b) Background:** Examines the issues of manual grade sheet management, emphasizing the need for automated alternatives.

- c) **Objectives:** Outline the particular goals that will guide the development of the Grade sheet Management System.
- d) **Methodology:** Discusses the methodology and procedures utilized to construct the system.
- e) **System Overview:** Provides a detailed description of the Grade sheet Management System, including its architecture, functionalities, and user interface.
- f) **Results and Discussion:** The system testing and evaluation results are presented, as well as a discussion of the findings and insights.
- g) **Future Work:** Explores potential areas for system refinement and development.
- h) **Conclusion:** Summarizes the report's important findings and contributions, emphasizes the importance of the Grade sheet Management System, and provides concluding remarks.
- i) **References:** A list of all sources cited in the report.
- j) **Appendices:** This section contains any additional auxiliary information, such as code snippets, sample grade sheets, or user manuals.

2. Project Description

2.1 Purpose and scope

The Grade sheet Management System (GMS) aims to transform how educational institutions maintain and process student grade sheets. The project's scope includes the creation of a sophisticated software program that automates different aspects of grade sheet management, such as generation, display, search, and deletion. The system intends to reduce administrative operations, increase efficiency, and improve grade sheet management accuracy in educational institutions.

2.2 Problem Description

Traditionally, grade sheet management in educational institutions has been done manually using paper-based records and spreadsheets. However, these methods are prone to mistakes, inefficiencies, and logistical issues. Administrators and teaching personnel frequently struggle with time-consuming tasks including manually inputting student data, computing grades, and accessing grade sheets as needed. Furthermore, paper-based records are easily lost, damaged, or misplaced, complicating grade sheet management.

The Grade sheet Management System tackles these issues by offering a centralized and automated solution to reduce errors, streamline administrative processes, and increase overall organizational effectiveness in educational institutions.

2.3 Functionality and Features

The Grade sheet Management System provides a variety of functionalities and features to satisfy the needs of educational institutions:

- **Automated Grade sheet Generation:** The system generates grade sheets using student data, removing the need for manual data entry.
- **Grade sheet Display:** A user-friendly interface allows users to quickly view and access grade sheets for individual students or entire courses.
- **Search Functionality:** The system has a powerful search feature that allows users to easily access grade sheets based on particular criteria such as registration number.
- **Secure Deletion:** The system can securely erase grade sheets as needed to remove unnecessary files.

3. Methodology

3.1 Development Process

- **Requirements Gathering:** First of all, I gathered all the information related to the problem and what is required to solve the problem.
- **Design:** In the next step, I visualized a general system architecture such as the user interface and other system components. Then I sketched the design in a normal paper.
- **Implementation:** Now, I translated the design into code. This involved writing the code using C programming language in IDE called Dev-C++.
- **Testing:** After writing the code, I tested the application to check if all the components work properly. Through testing, I could make sure that the system meets all the requirements and solves the problem in an efficient manner.
- **Deployment:** Finally, after making sure that there are no bugs or errors, the system is ready to be used in the organization.

3.2 Tools and Resources

- **IDE (Integrated Development Environment):** The project used **Dev-C++** IDE to write, edit, and debug code.
- **Compiler:** It used Mingw port of GCC (GNU Compiler Collection) as its compiler.
- **Libraries:** The application used libraries such as the Standard C Library for input and output `<stdio.h>`. For dealing with strings, it used `<string.h>` library. For terminating program, it used `<stdlib.h>`

3.3 Development Environment

- **Platform:** The Grade sheet Management System is designed for desktop platforms, including Windows and macOS.
- **Operating System:** For development of the application, Windows Operating System was used.

4. Results and Discussion

4.1 Presentation of Project Outcomes and Achievements

The creation of the Grade sheet Management System (GMS) has yielded notable outcomes and accomplishments, hence contributing to the improvement of grade sheet management practices in educational institutions. Key results include:

- **Automation of Grade sheet Management:** The GMS effectively automates important grade sheet management functions such as generation, display, search, and deletion. This automation has simplified administrative operations, decreasing manual work and saving time for administrators and faculty members.
- **Enhanced Efficiency and Accuracy:** By automating data entry and computation methods, the GMS has enhanced grade sheet processing efficiency and accuracy. The system ensures that grade sheet data is accurate and dependable, while also minimizing errors and conflicts.
- **Enhanced Accessibility:** Administrators, faculty members, and students may easily obtain grade sheet information thanks to the GMS's user-friendly design. This accessibility promotes informed decision-making and academic planning, resulting in improved organizational effectiveness.

4.2 Comparison of Actual Results with Expected Outcomes

The actual results obtained from the GMS implementation closely match the predicted outcomes specified during the project design phase. Specifically:

- **Automation:** The GMS successfully automates grade sheet management duties, as expected. The system efficiently generates grade sheets based on student data, displays grade sheets for individual students or entire classes.
- **Efficiency and Accuracy:** The GMS has showed improved efficiency and accuracy in grade sheet processing, which meets expectations. By automating data entry and computation methods, the system reduces errors and discrepancies, resulting in dependable grade sheet data.
- **Accessibility:** The GMS improves access to grade sheet information, as planned. The user-friendly design allows users to easily retrieve grade sheets, making information retrieval and decision-making more efficient.

4.3 Discussion of Challenges Faced and Lessons Learned

Several problems emerged during the project, providing great learning experiences:

- **Technical Challenges:** During the implementation phase, some technical issues arose, such as troubleshooting complex code and optimizing system performance. These issues required extensive problem solving and asking help from friends and looking for solutions on the internet.
- **User Feedback:** Without proper feedback from other users, the system could not be trusted for solving the problem. That is why, I asked many of my friends to run their code in their computer and ask their feedback.
- **Limited Time:** Due to limited time, I was not able to add more features that could make the system even better and more efficient.

By encountering these problems, I was able to learn the following lessons:

- **Adaptability:** Projects are frequently met with unexpected challenges or changes in needs. Being flexible and quick in our response to these issues is critical for overcoming obstacles and ensuring project success.
- **Time Management:** Effective time management is essential for fulfilling project deadlines and milestones. Effective task planning, scheduling, and prioritization can help us manage your time more efficiently.
- **Continuous Improvement:** Reflecting on project successes and failures might yield useful insights for future projects. Embracing a culture of continuous development, learning from prior events, and putting those lessons into practice can lead to improved results in future attempts.

5. Conclusion

In conclusion, the Grade sheet Management System project is a successful attempt to use technology to improve academic administration processes. Its adoption has established a solid foundation for improving grade sheet administration methods, increasing organizational efficiency, and ultimately promoting educational institutions' performance.

The successful completion of the GMS project is a significant step toward improving academic administration processes. The system's capacity to automate grade sheet management operations, increase productivity, and improve accuracy illustrates its potential to address common difficulties confronting educational institutions.

Furthermore, the GMS's success demonstrates the revolutionary potential of technology in the field of education. By using technology to optimize administrative operations, educational institutions can better distribute resources, minimize manual workload, and improve overall education quality.

6. Future Work

6.1 Suggestion for Future Enhancements

- **Basic User Authentication:** Create a simple user authentication system in which users must input a username and password to access the grade sheet management system. This can be accomplished by using basic file handling techniques in C to store and validate user credentials.
- **Grade Average Calculation:** Add functionality to calculate each student's average grade depending on their performance in various subjects. Display the calculated average with specific topic grades to provide a complete overview of student achievement.
- **Basic Data Validation:** Implement basic data validation checks to ensure that user inputs, such as student names and subject marks, are within valid ranges. For example, ensure that marks are between 0 and 100, and student names do not contain special characters.

6.2 Ideas for Additional Improvements

In addition to future developments, there are various ideas for more features or changes that could further enhance the functioning and usability of the GMS:

- **Basic Editing Options:** Implement basic editing options for grade sheets, such as the ability to modify or update the marks for individual subjects. Provide prompts to select the student and subject to edit, and then allow the user to enter the new marks.
- **Export to Text File:** Enable users to export grade sheet data to a text file for easy backup or sharing. Allow the user to specify the filename and location for the exported file, and then save the grade sheet data in a simple text format.
- **Subject Addition:** Allow users to add new subjects to the system. Prompt the user to enter the name of the subject and any other relevant details, and then include it in the list of subjects for which grades can be recorded.

7. References

[2] “Project Description” retrieved from:

<https://www.linkedin.com/pulse/what-7-disadvantages-manual-system-richard-breitmeyer/>

[4] “Results and Discussion” retrieved from:

<https://wpschoolpress.com/benefits-advantages-student-management-system/>

8. Appendices

8.1 Source Code

```
#include<stdio.h>

#include<stdlib.h>

#include<windows.h>

/* *****Structures***** */

// *****Sub Structures***** //

struct subject {

    char subjectName[20];

    int marks;

} subject;

struct marks {

    int totalMarks;

    float percentage;

};

// *****MAIN STRUCTURE***** //

typedef struct student {

    int registrationNum;

    char name[20];

    char faculty[20];

    int semester;

    struct subject subj[5];

    struct marks mrk;

} student;
```

```

/* *****Function Prototypes***** */

int home();

void confirmationDialogue(int,int,char *,char *);

void getStudentInfo(struct student *std);

char* calculateGrade(int marks);

void saveToFile(struct student *std);

void readFromFile();

int generateGradesheet(int,int,struct student *std);

int showGradesheet();

int searchGradesheet();

int deleteGradesheet();

/* *****Utility Functions Prototypes***** */

void gotoxy(int,int);

void printHorizontalLine(int,int,int);

void printVerticalLine(int,int,int);

void printRows(int,char *,int [],float);

/* *****Main Program***** */

int main() {

    char ch;

    int x=30,y=15;

    do {

        system("cls");

        y=home();

        y+=1;

        confirmationDialogue(x,++y,"Do you want to perform another
operation?[y/n]: ",&ch);

```

```

        } while(ch=='y' || ch=='Y');

        return 0;

    }

    /* *****Function Definitions***** */

    // Function to move cursor to specified position

    void gotoxy(int x,int y) {

        COORD coord;

        coord.X=x;

        coord.Y=y;

        SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), coord);

    }

    // Function to print horizontal line

    void printHorizontalLine(int x,int y,int width) {

        int i;

        for(i=0; i<width; i++) {

            gotoxy(x++,y);

            printf("-");

        }

    }

    // Function to print vertical line

    void printVerticalLine(int x,int y,int height) {

        int i;

        for(i=0; i<height; i++) {

            gotoxy(x,++y);

            printf("|");

        }

    }

```

```
}
```

// Function to display starting point of program

```
int home() {  
    system("Color 07");  
    struct student std;  
    char ch;  
    int x=30,y=5;  
    int choice;  
    int width=46;  
    printHorizontalLine(30,5,width);  
    gotoxy(x,++y);  
    printf("|*****Gradesheet Management System*****|");  
    printHorizontalLine(30,7,width);  
    y+=2;  
    gotoxy(x,y);  
    printf("1. Generate Gradesheet");  
    gotoxy(x,++y);  
    printf("2. Show all gradesheet");  
    gotoxy(x,++y);  
    printf("3. Search gradesheet");  
    gotoxy(x,++y);  
    printf("4. Delete gradesheet");  
    gotoxy(x,++y);  
    printf("5. Exit program");  
    printHorizontalLine(30,++y,width);  
    y+=2;  
}
```

```

gotoxy(x,y);

printf("Which operation do you want to perform?[1-5]: ");

scanf("%d",&choice);

fflush(stdin);

system("cls");

switch(choice) {

    case 1:

        system("Color 02");

        getStudentInfo(&std);

        system("cls");

        y=generateGradesheet(40,5,&std);

        y+=1;

        confirmationDialogue(30,++y,"Do you want to save the
gradesheet?[y/n]: ",&ch);

        if(ch=='y' || ch=='Y') {

            gotoxy(30,++y);

            saveToFile(&std);

        }

        break;

    case 2:

        system("cls");

        system("Color 07");

        y=showGradesheet();

        break;

    case 3:

        system("cls");

```



```

        system("Color 03");

        y=searchGradesheet();

        break;

    case 4:

        system("cls");

        system("Color 04");

        y=deleteGradesheet();

        break;

    case 5:

        exit(0);

    default:

        gotoxy(30,5);

        printf("Invalid choice");

    }

    return y;

}

```

// Function to ask whether to continue program

```

void confirmationDialogue(int x,int y,char *dialogue,char *ch) {

    gotoxy(x,y);

    printf("%s",dialogue);

    scanf(" %c",ch);

}

```

// Function to get student information

```

void getStudentInfo(struct student *std) {

    struct student std1;

    FILE *fptr;

```

```

int i,x=30,y=5;

fptr=fopen("student.txt","ab+");

if(fptr==NULL) {

    printf("Error opening student.txt");

    exit(1);

}

std->mrk.totalMarks=0;

gotoxy(x,y);

printf("Enter student details:");

y+=2;

gotoxy(x,y);

printf("Name: ");

fgets(std->name,sizeof(std->name),stdin);

gotoxy(x,++y);

printf("NCCS reg. num: ");

scanf("%d",&std->registrationNum);

while(fread(&std1,sizeof(std1),1,fptr)) {

    while(std->registrationNum==std1.registrationNum) {

        y+=1;

        gotoxy(x,++y);

        printf("Record of student already exists!!");

        gotoxy(x,++y);

        printf("Enter NCCS reg. num again: ");

        scanf("%d",&std->registrationNum);

        // Reset file pointer to check reg No from the beginning

        rewind(fptr);
    }
}

```

```

        }

    }

    fclose(fptr);

    getchar(); // consume the new line character

    y+=1;

    gotoxy(x,++y);

    printf("Faculty: ");

    fgets(std->faculty,sizeof(std->faculty),stdin);

    gotoxy(x,++y);

    printf("Semester(In number): ");

    scanf("%d",&std->semester);

    getchar(); // Consume the new line character

    // Input marks
    for(i=0; i<5; i++) {

        y+=2;

        gotoxy(x,y);

        printf("Enter subject name: ");

        fgets(std->subj[i].subjectName,sizeof(std->subj[i].subjectName),stdin);

        gotoxy(x,++y);

        printf("Enter marks: ");

        scanf("%d",&std->subj[i].marks);

        getchar(); // consume new line character

        std->mrk.totalMarks+=std->subj[i].marks;

    }

    std->mrk.percentage=(float)std->mrk.totalMarks/500*100;

}

```

// Function to calculate grade

```
char* calculateGrade(int marks) {  
    if(marks>=90 && marks<=100) {  
        return "A+";  
    } else if(marks>=80) {  
        return "A";  
    } else if(marks>=70) {  
        return "B+";  
    } else if(marks>=60) {  
        return "B";  
    } else if(marks>=50) {  
        return "C";  
    } else {  
        return "NG";  
    }  
}
```

// Function to save data to file

```
void saveToFile(student *std) {  
    FILE *fptr;  
    int i;  
    fptr=fopen("student.txt","ab");  
    if(fptr==NULL) {  
        system("cls");  
        printf("Error opening student.txt");  
    }  
}
```

```

if(fwrite(std,sizeof(student),1,fptr)) {
    printf("Successfully saved to file");
} else {
    printf("Error writing into file");
}
fclose(fptr);
}

```

// Function to generateGradesheet

```

int generateGradesheet(int x,int y,struct student *std) {
    char ch;
    int i;
    gotoxy(x,y);
    printf("National College Of Computer Studies");
    x+=8;
    gotoxy(x,++y);
    printf("Paknajol, Kathmandu");
    x-=18;
    y+=2;
    printHorizontalLine(x,y,56);
    printVerticalLine(x,y,19);
    x+=3;
    gotoxy(x,++y);
    printf("Name: %s",std->name);
    gotoxy(x,++y);
    printf("NCCS Reg. No: %d",std->registrationNum);
    gotoxy(x,++y);
}

```

```

printf("Faculty: %s",std->faculty);

gotoxy(x,++y);

printf("Semester: %d",std->semester);

printHorizontalLine(x-2,++y,55);

printVerticalLine(x+52,y-5,19);

gotoxy(x+1,++y);

printf("Subject Name");

gotoxy(x+18,y);

printf("Marks Obtained");

gotoxy(x+36,y);

printf("Grade Obtained");

printHorizontalLine(x-2,++y,54);

printVerticalLine(x+16,y-2,14);

printVerticalLine(x+33,y-2,14);

for(i=0; i<5; i++) {

    // Subject Name

    gotoxy(x+1,++y);

    printf("%s",std->subj[i].subjectName);

    // Marks obtained

    gotoxy(x+24,y);

    printf("%d",std->subj[i].marks);

    gotoxy(x+42,y);

    // Grade obtained

    printf("%s",calculateGrade(std->subj[i].marks));

    printHorizontalLine(x-2,++y,54);

}

```

```

    printHorizontalLine(x-2,++y,54);

    // Percentage

    gotoxy(x+1,++y);

    printf("Percentage");

    gotoxy(x+40,y);

    printf("%.2f%%",std->mrk.percentage);

    printHorizontalLine(x-3,++y,56);

    return y;
}

// Function to show grade sheet

int showGradesheet() {

    FILE *fptr;

    int x=40,y=5,flag;

    struct student std;

    fptr=fopen("student.txt","rb");

    if(fptr==NULL) {

        printf("Error opening student.txt");

        exit(1);

    }

    flag=fread(&std,sizeof(std),1,fptr);

    if(flag<1) {

        gotoxy(30,y);

        printf("No gradesheet found!!!");

        fclose(fptr);

        return y;

    } else {

```

```

        generateGradesheet(x,y,&std);

        printHorizontalLine(0,y+25,120);

        y+=26;

        while(fread(&std,sizeof(std),1,fptr)) {

            generateGradesheet(x,y,&std);

            printHorizontalLine(0,y+25,120);

            y+=26;

        }

    }

    fclose(fptr);

    return y;

}

```

// Function to search grad esheet by registration No

```

int searchGradesheet() {

    FILE *fptr;

    struct student std;

    int regNo;

    int x=40,y=5,flag=0;

    gotoxy(x-10,y);

    printf("Enter NCCS registration number of the student: ");

    scanf("%d",&regNo);

    y+=2;

    fptr=fopen("student.txt","rb");

    if(fptr==NULL) {

        printf("Error opening student.txt");

        exit(1);

    }

}

```



```

    }

    while(fread(&std,sizeof(std),1,fptr)) {

        if(std.registrationNum==regNo) {

            y=generateGradesheet(x,++y,&std);

            flag=1;

            break;

        }

    }

    gotoxy(x,++y);

    if(flag==0) {

        printf("Record not found!!");

    }

    fclose(fptr);

    return y;

}

```

// Function to delete grade sheet by registration No

```

int deleteGradesheet() {

    FILE *fptr1,*fptr2;

    struct student std;

    int regNo,err=0,flag=0;

    int x=40,y=5;

    gotoxy(x-10,y);

    printf("Enter NCCS registration number of the student: ");

    scanf("%d",&regNo);

    y+=1;

    fptr1=fopen("student.txt","rb");

```

```

fptr2=fopen("temp.txt","wb");

if(fptr1==NULL) {

    printf("Error opening student.txt");

    exit(1);

}

if(fptr2==NULL) {

    printf("Error opening student.txt");

    exit(1);

}

while(fread(&std,sizeof(std),1,fptr1)) {

    if(std.registrationNum!=regNo) {

        fwrite(&std,sizeof(std),1,fptr2);

    } else {

        flag=1;

    }

}

if(flag==0) {

    gotoxy(x,++y);

    printf("Gradesheet not found!!!!");

    fclose(fptr1);

    fclose(fptr2);

    return y;

}

fclose(fptr1);

fclose(fptr2);

err=remove("student.txt");

```

```

gotoxy(x-5,++y);

if(err!=0) {

    printf("Error deleting student.txt");

    fclose(fp1);

    fclose(fp2);

    return y;

}

err=rename("temp.txt","student.txt");

if(err!=0) {

    printf("Error deleting record");

    fclose(fp1);

    fclose(fp2);

    return y;

} else {

    printf("Successfully deleted record");

}

return y;

}

```

8.2 Output

```
-----  
|*****Gradesheet Management System*****|  
-----  
1. Generate Gradesheet  
2. Show all gradesheet  
3. Search gradesheet  
4. Delete gradesheet  
5. Exit program  
-----  
  
Which operation do you want to perform?[1-5]:
```

Fig 8.1: Home Interface

```
Enter student details:  
  
Name: Siddhartha Shakya  
NCCS reg. num: 1288  
  
Faculty: BIM  
Semester(In number): 1  
  
Enter subject name: English  
Enter marks: 89  
  
Enter subject name: Mathematics  
Enter marks: 92  
  
Enter subject name: Business  
Enter marks: 81  
  
Enter subject name: C Programming  
Enter marks: 100  
  
Enter subject name: FIT  
Enter marks: 91
```

Fig 8.2: Option 1 (Input Details)

National College Of Computer Studies
Paknajol, Kathmandu

Name: Siddhartha Shakya
NCCS Reg. No: 1288
Faculty: BIM
Semester: 1

Subject Name	Marks Obtained	Grade Obtained
English	89	A
Mathematics	92	A+
Business	81	A
C Programming	100	A+
FIT	91	A+
Percentage		90.60%

Do you want to save the gradesheet?[y/n]: y
Successfully saved to file

Do you want to perform another operation?[y/n]:

Fig 8.3: Option 1 (Generate Report)

Enter NCCS registration number of the student: 1288

Successfully deleted record

Do you want to perform another operation?[y/n]:

Fig 8.4: Option 4 (Delete Record)

Enter NCCS registration number of the student: 1288

National College Of Computer Studies
Paknajol, Kathmandu

Name: Siddhartha Shakya		
NCCS Reg. No: 1288		
Faculty: BIM		
Semester: 1		
Subject Name	Marks Obtained	Grade Obtained
English	89	A
Mathematics	92	A+
Business	81	A
C Programming	100	A+
FIT	91	A+
Percentage		90.60%

Fig 8.5: Option 3 (Search Record)