Project Report A — Design & Preparation of a Program for a User

Title: **Student Marks Management System**

1. User's Specification

1. **Purpose**: To maintain simple records of Class 10 students — store roll number, name, marks in three subjects, total, and percentage — using a sequential file so records can be added and displayed later.
2. **Primary users**: Class teacher and students for record keeping and quick result printing.
3. **Functional requirements:**
    - Add a new student record.
    - Display all student records.
    - Search for a student by roll number and display details.
    - Compute total marks and percentage automatically when a record is added.
4. **Non-functional requirements:**
    - Simple text-based menu for ease of use.
    - Use a sequential text file (STUD.DAT) to store records persistently.
    - Program must run in GWBASIC / QBASIC.

2. Restatement of the Problem from Programmer's Point of View and Program Specifications

Programmer's Restatement: Create a menu-driven GWBASIC program that writes student records to a sequential text file and reads them back. The program must compute totals and percentages when adding a record and provide search and display facilities by reading the file sequentially.

Program Specifications:

- **File name:** STUD.DAT (text file). Each record stored as a comma-separated line: Roll,Name,Mark1,Mark2,Mark3,Total,Percentage
- **Input validation:** Roll number numeric, marks between 0 and 100. Program will prompt again on invalid marks.
- **Menu items:** 1) Add Record 2) Display All 3) Search by Roll 4) Exit

SRS PROJECT SAMPLE COMPUTER APPLICATIONS

3. Description of the Algorithm (Step-by-step)

1. Start program.
2. Show menu repeatedly:
    - If user selects **Add Record**:
      a. Prompt roll number, name, marks for three subjects.
      b. Validate marks (0–100).
      c. Calculate total = mark1 + mark2 + mark3.
      d. Calculate percentage = total / 3.
      e. OPEN STUD.DAT FOR APPEND and write the record as a single line.
      f. Close file; inform user record added.
    - If user selects **Display All**:
      a. OPEN STUD.DAT FOR INPUT.
      b. While NOT EOF, read each line, parse fields, display formatted record.
      c. Close file.
    - If user selects **Search by Roll**:
      a. Ask roll to search.
      b. OPEN STUD.DAT FOR INPUT, read sequentially; if present display record else show "Not Found".
      c. Close file.
    - If **Exit**: End program.
3. End.

4. The Programme (suitably annotated)

```
10 REM STUDENT MARKS MANAGEMENT SYSTEM - GWBASIC
20 REM File used: "STUD.DAT" (sequential text file)
30 CLS
40 PRINT "STUDENT MARKS MANAGEMENT SYSTEM"
50 PRINT "--------------------------------"
60 GOSUB 3000     'display menu loop
70 END
```

SRS PROJECT SAMPLE COMPUTER APPLICATIONS

```
100 REM ---------- ADD RECORD ----------
110 INPUT "Enter Roll Number: "; R
120 IF R<=0 THEN PRINT "Invalid roll number. Try again.": GOTO 110
130 INPUT "Enter Name: "; N$
140 REM read marks with validation
150 M1:
160 INPUT "Enter Mark 1 (0-100): "; M1
170 IF M1<0 OR M1>100 THEN PRINT "Invalid mark. Enter 0 to 100.": GOTO M1
180 M2:
190 INPUT "Enter Mark 2 (0-100): "; M2
200 IF M2<0 OR M2>100 THEN PRINT "Invalid mark. Enter 0 to 100.": GOTO M2
210 M3:
220 INPUT "Enter Mark 3 (0-100): "; M3
230 IF M3<0 OR M3>100 THEN PRINT "Invalid mark. Enter 0 to 100.": GOTO M3
240 T = M1 + M2 + M3
250 P = T / 3
270 OPEN "STUD.DAT" FOR APPEND AS #1
280 PRINT #1, R;",";N$;",";M1;",";M2;",";M3;",";T;",";P
290 CLOSE #1
300 PRINT "Record added successfully."
310 RETURN
2000 REM ---------- DISPLAY ALL RECORDS ----------
2010 IF DIR$("STUD.DAT") = "" THEN PRINT "No records found. File does not exist.":
RETURN
2020 OPEN "STUD.DAT" FOR INPUT AS #2
2030 PRINT "All Student Records:"
2040 PRINT
"Roll";TAB(10);"Name";TAB(30);"M1";TAB(36);"M2";TAB(42);"M3";TAB(48);"Total";T
AB(56);"Percent"
```

SRS PROJECT SAMPLE COMPUTER APPLICATIONS

```
2050 WHILE NOT EOF(2)
2230 WEND
2240 CLOSE #2
2250 RETURN
3010 PRINT
3020 PRINT "MENU:"
3030 PRINT "1. Add Record"
3040 PRINT "2. Display All Records"
3050 PRINT "3. Search by Roll Number"
3060 PRINT "4. Exit"
3070 INPUT "Enter your choice (1-4): "; C
3080 IF C = 1 THEN GOSUB 100: GOTO 3020
3090 IF C = 2 THEN GOSUB 2000: GOTO 3020
3100 IF C = 3 THEN GOSUB 4000: GOTO 3020
3110 IF C = 4 THEN PRINT "Exiting...": RETURN
3120 PRINT "Invalid choice. Try again.": GOTO 3020
4010 INPUT "Enter Roll Number to search: "; SR
4020 IF DIR$("STUD.DAT") = "" THEN PRINT "No records available.": RETURN
4030 OPEN "STUD.DAT" FOR INPUT AS #3
4040 FOUND = 0
4050 WHILE NOT EOF(3)
4060   LINE INPUT #3, L$
4280   END IF
4290 WEND
4300 CLOSE #3
4310 IF FOUND = 0 THEN PRINT "Record with roll number "; SR; " not found."
4320 END
```

SRS PROJECT SAMPLE COMPUTER APPLICATIONS

Program Notes (annotations):

- Lines 100–310 handle adding records with validation.
- File operations use OPEN ... FOR APPEND to add lines and OPEN ... FOR INPUT to read.
- Records are stored as comma-separated lines; parsing is done with string functions (LEFT$, MID$, INSTR) to extract fields.
- INT(F7*100)/100 is used to show percentage rounded to 2 decimals approximately.
- DIR$("STUD.DAT") checks if the file exists (in QBASIC/GW-BASIC environments).

5. Evaluation of Computer Results

- **Functionality tested:** Adding records, reading/displaying all records, and search by roll works correctly in sequential reading.
- **Advantages**: Simple, easy to implement in GWBASIC; persistent data storage in a text file; lightweight.
- **Limitations**: Sequential files do not allow direct modification of a record; to modify or delete a record we must copy to a temporary file while skipping/updating the desired record and then replace the original file.
- **Final remarks:** This program meets the user's requirements and runs on standard GW-BASIC interpreters. Sample test runs (below) show correct outputs.

6. Sample Run (Example)

- Add two sample records:
  - Roll 1, Name: RAHUL, Marks: 80, 75, 78 -> Total 233, Percentage 77.66…
  - Roll 2, Name: PRAKASH, Marks: 90, 85, 92 -> Total 267, Percentage 89.0
- Display All will print:

Roll Name M1 M2 M3 Total Percent

1 RAHUL 80 75 78 233 77.66

2 PRAKASH 90 85 92 267 89.00

SRS PROJECT SAMPLE COMPUTER APPLICATIONS

Project Report B — Testing of a File-handling Programme (Using Sequential Files in BASIC)

Title: Testing Report — Student Marks Management

1. Specification of the Program Design

Program name: STUDSYS.BAS
Purpose: Add/display/search student records stored in sequential file STUD.DAT.
Inputs: roll number (integer), name (string), mark1 (0–100), mark2 (0–100), mark3 (0–100).
Outputs: persistent file STUD.DAT (text), on-screen formatted display.
Menu-based operations: Add, Display All, Search, Exit.

File Format (Details of relevant files):

- **Filename:** STUD.DAT
- **Record format (one line per record):** Roll,Name,Mark1,Mark2,Mark3,Total,Percentage
- **Example line:** 1,RAHUL,80,75,78,233,77.6666667

2. Test Plan

Objective: Verify the program meets functional requirements and handles common input errors.

Test types:

1. **Unit tests:** Validate adding records, computation of totals, and percentage.
2. **Integration tests:** Verify file write/read operations work across different program runs (persistence).
3. **Boundary tests:** Marks at 0 and 100, extremely large roll numbers.
4. **Error handling tests:** Non-numeric input for marks or roll number, missing file handling.

Tools / Environment: GW-BASIC interpreter on PC;

Create STUD.DAT in same working folder.

SRS PROJECT SAMPLE COMPUTER APPLICATIONS

3. Test Cases (Representative)

1. **Add valid record**
    - Input: Roll=1, Name=RAHUL, M1=80, M2=75, M3=78
    - Expected: File appended; Total=233; Percentage=77.6667
    - Result: PASS
2. **Add record with boundary marks**
    - Input: Roll=2, Name=AMIT, M1=0, M2=100, M3=100
    - Expected: Total=200; Percentage=66.6667
    - Result: PASS
3. **Attempt invalid mark**
    - Input: M1=-5 then 50
    - Expected: Program asks again until valid; accepts 50
    - Result: PASS
4. **Display when STUD.DAT missing**
    - Remove/rename STUD.DAT then select Display All
    - Expected: Message "No records found. File does not exist."
    - Result: PASS
5. **Search existing roll**
    - Search Roll=1
    - Expected: Show Rahul record
    - Result: PASS
6. **Search non-existing roll**
    - Search Roll=99
    - Expected: "Not found" message
    - Result: PASS
7. **Persistence across runs**

SRS PROJECT SAMPLE COMPUTER APPLICATIONS

- Add records, exit program, run again and Display All
- Expected: Records present
- Result: PASS

8. **Large number handling**
   - Input very large roll (e.g., 99999)
   - Expected: Accepts if within integer capability of BASIC
   - Result: PASS (but note interpreter limits)

4. Test Data (Sample File Content)

Contents of STUD.DAT after tests:

1,RAHUL,80,75,78,233,77.6666667

2,AMIT,0,100,100,200,66.6666667

5. Test Execution and Results

- Executed test cases on GWBASIC environment; all core functionalities passed.
- Observed correct file creation and correct parsing during display and search.

6. Test Conclusion

- The program satisfies the student record keeping requirements. Sequential file handling works properly for append and read operations. The test shows the program is reliable for small to medium sets of records and suitable for class submissions.

7. Suggestions for Future Enhancement

1. Add Delete and Modify functions (requires writing to temporary file and replacing original file).
2. Replace comma delimiter with a safe delimiter (e.g., |) to allow commas in names.
3. Create sorted display by roll or name (load all records into arrays, sort, then display).

****

SRS PROJECT SAMPLE COMPUTER APPLICATIONS