

Software Requirement Specification (SRS)

Student Result Management System

1. Introduction

1.1 Purpose

The purpose of this document is to define the requirements for the **Student Result Management System (SRMS)**. This system allows **admins, teachers, and students** to manage student academic records efficiently.

1.2 Scope

The SRMS will:

- Allow **admins** to manage users and results.
- Allow **teachers** to enter and update student marks.
- Allow **students** to view their results.
- Store student records in a `.txt` file.

1.3 Definitions, Acronyms, and Abbreviations

- **SRMS** – Student Result Management System
- **Admin** – System Superuser
- **Teacher** – User responsible for entering student marks
- **Student** – User who can view their results

1.4 References

- C Programming Language for File Handling
 - IEEE SRS Standard Document Template
-

2. Overall Description

2.1 Product Perspective

SRMS is a standalone software designed for managing student academic results, ensuring role-based access control.

2.2 User Classes and Characteristics

User Role	Description
Admin	Full access to manage users, add/delete students, and oversee results.
Teacher	Can enter and update student marks but cannot delete records.
Student	Can only view their results.

2.3 Operating Environment

- OS: Windows/Linux
- Programming Language: C
- File Storage: `.txt` file for database

2.4 Design and Implementation Constraints

- The system must use **C file handling** instead of databases.
- User authentication must be **stored securely in a text file**.

2.5 Assumptions and Dependencies

- The system assumes valid user credentials are pre-registered in `users.txt`.
 - The system assumes `.txt` file handling is sufficient for data storage.
-

3. Functional Requirements

3.1 User Authentication

- The system must allow users to log in with a username and password.
- Roles should be assigned based on credentials from `users.txt`.

3.2 Admin Functions

- Add/Delete teachers and students.
- Manage student results.

- View reports of all students.

3.3 Teacher Functions

- Enter or update student marks.
- Generate student performance reports.

3.4 Student Functions

- View personal results (total, percentage, pass/fail status).

3.5 File Handling Requirements

- Student records should be stored in `students.txt`.
 - User credentials should be stored in `users.txt`.
-

4. Non-Functional Requirements

4.1 Performance Requirements

- The system should process login requests within **2 seconds**.
- The system should retrieve and display student results in **less than 1 second**.

4.2 Security Requirements

- Passwords must be stored **in an encrypted format**.
- Users must have **restricted access** based on roles.

4.3 Usability Requirements

- The system should have a **menu-driven interface** for easy navigation.
- Error messages should be **clear and informative**.

4.4 Maintainability and Scalability

- The `.txt` file format should be structured for easy modifications.
 - Future versions may **upgrade to a database system (e.g., MySQL)**.
-

5. Appendices

- **Appendix A:** Example of `users.txt` and `students.txt` format.

users.txt (Example)

```
admin,admin123,admin  
teacher1,teachpass,teacher  
student1,studpass,student
```

students.txt (Example)

```
101,John Doe,2023,Class 10,Math:80,Science:90,English:85  
102,Jane Smith,2023,Class 10,Math:75,Science:95,English:88
```

6. Conclusion

This document outlines the requirements for the **Student Result Management System**. It ensures role-based access control, authentication, and result management using **C file handling**. Future enhancements may include **database integration** and **GUI improvements**.