# Software Requirements Specification for Student Result Management System

**Version:** 1.0

## Table of Contents

1. Introduction  
   1.1. Purpose  
   1.2. Scope  
   1.3. Definitions, Acronyms, and Abbreviations
2. Overall Description  
   2.1. Product Perspective  
   2.2. Product Functions  
   2.3. User Characteristics  
   2.4. Constraints  
   2.5. Assumptions and Dependencies
3. System Models  
   3.1. Context Diagram  
   3.2. Data Flow Diagrams (DFD)  
   3.2.1. Level 1 DFD  
   3.2.2. Level 2 DFDs  
   3.3. Use Case Diagram  
   3.4. Entity-Relationship Diagram (ERD)
4. Interface Requirements  
   4.1. User Interfaces  
   4.2. Hardware Interfaces  
   4.3. Software Interfaces
5. Functional Requirements  
   5.1. User Management (FR-01)  
   5.2. Student Information Management (FR-02)  
   5.3. Subject Information Maintenance (FR-03)  
   5.4. Marks Information Maintenance (FR-04)  
   5.5. Report Generation (FR-05)
6. Non-Functional Requirements  
   6.1. Security  
   6.2. Performance  
   6.3. Maintainability  
   6.4. Portability  
   6.5. Reliability

### 1. Introduction

This document provides a detailed description of the requirements for the Student Result Management System. It outlines the purpose, scope, functionalities, and constraints of the system. The intended audience for this document includes developers, testers, project managers, and end-users.

#### 1.1 Purpose

The primary purpose of the Student Result Management System is to automate the process of managing student results at a university. This system will provide a centralized platform for entering, storing, managing, and accessing student academic data, thereby simplifying and speeding up the result preparation and distribution process.

#### 1.2 Scope

The system will manage information about:

* **Students:** Details of students enrolled in various undergraduate programs.
* **Subjects:** Information about subjects offered in different semesters.
* **Marks:** Marks obtained by students in various subjects.
* **Users:** Different user roles with specific access rights (Administrator, Data Entry Operator, Teacher, Student).
* **Reports:** Generation of performance reports and semester-wise mark sheets.

#### 1.3 Definitions, Acronyms, and Abbreviations

|  |  |
| --- | --- |
| **Term** | **Description** |
| **SRS** | Software Requirements Specification |
| **UG** | Undergraduate |
| **DBA** | Database Administrator |
| **DFD** | Data Flow Diagram |
| **ERD** | Entity-Relationship Diagram |

### 2. Overall Description

#### 2.1 Product Perspective

The Student Result Management System is a self-contained application that will replace the manual process of result management. It will be a web-based application accessible to authorized users within the university network.

#### 2.2 Product Functions

The system will support the following key functions:

* **User Authentication:** Secure login for all user types.
* **Student Information Management:** Add, update, and view student details.
* **Subject Management:** Add, update, and view subject information for each semester.
* **Marks Management:** Enter and update marks for students in different subjects.
* **User Account Management:** Create, modify, and delete user accounts.
* **Report Generation:** Generate various reports like student performance reports and semester-wise mark sheets.
* **System Reset:** A feature for the administrator to clear all data from the database.

#### 2.3 User Characteristics

|  |  |
| --- | --- |
| **User Role** | **Responsibilities & Expertise** |
| **Administrator** | Overall system management, user account maintenance, report generation, system reset. Requires basic computer literacy. |
| **Data Entry Operator** | Manages student and subject information. Requires basic data entry skills. |
| **Teacher** | Enters and updates student marks. Requires basic computer literacy. |
| **Student** | Views their personal information, academic records, and results. Requires basic computer literacy. |

#### 2.4 Constraints

* The system will be developed using Visual Studio 2010 and SQL Server 2008.
* The application must run on Windows XP/Vista/7.
* The system must be password-protected to ensure data security.

#### 2.5 Assumptions and Dependencies

* The number of semesters and subjects per semester is fixed.
* The underlying database will be managed and backed up by the university's DBA.
* All users will have valid login credentials provided by the Administrator.

### 3. System Models

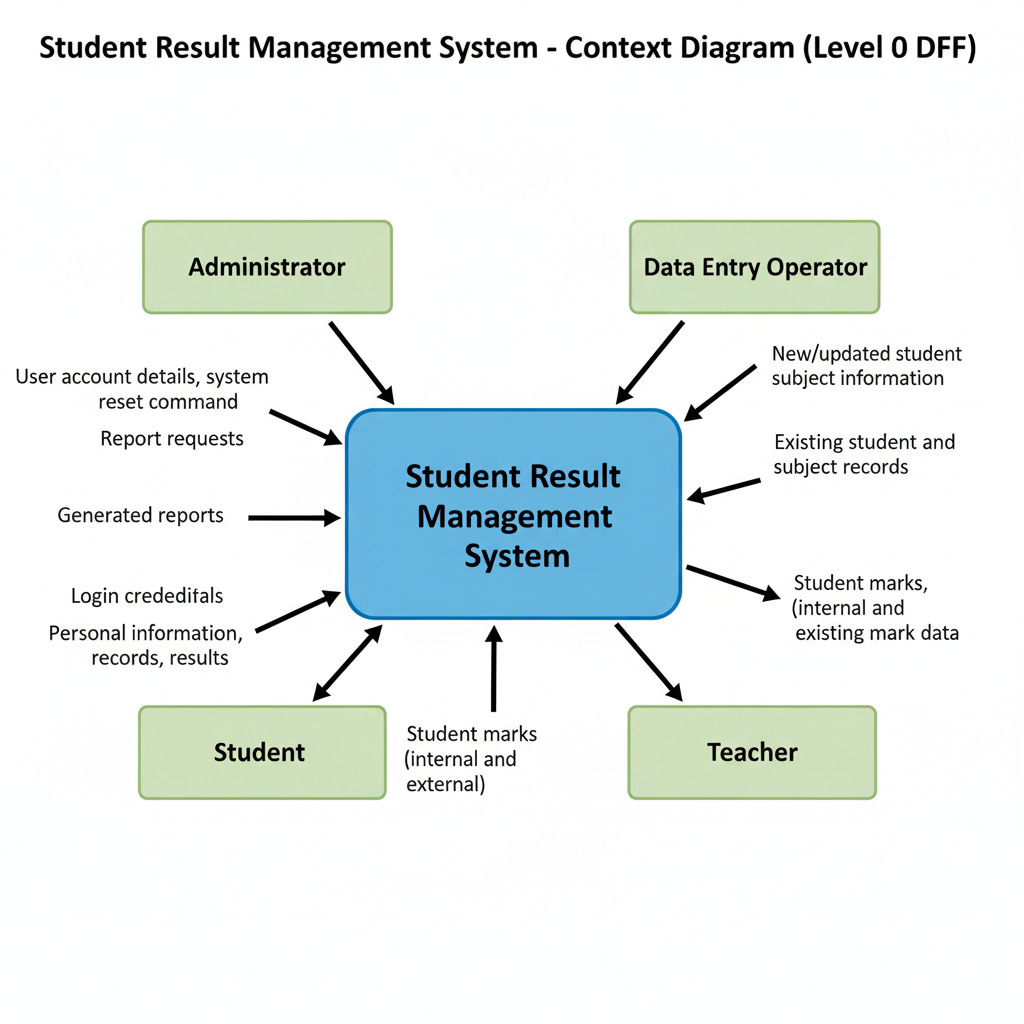
#### 3.1 Context Diagram

The context diagram shows the interaction of external entities with the system. The main entities are Administrator, Data Entry Operator, Teacher, and Student.

#### 3.2 Data Flow Diagrams (DFD)

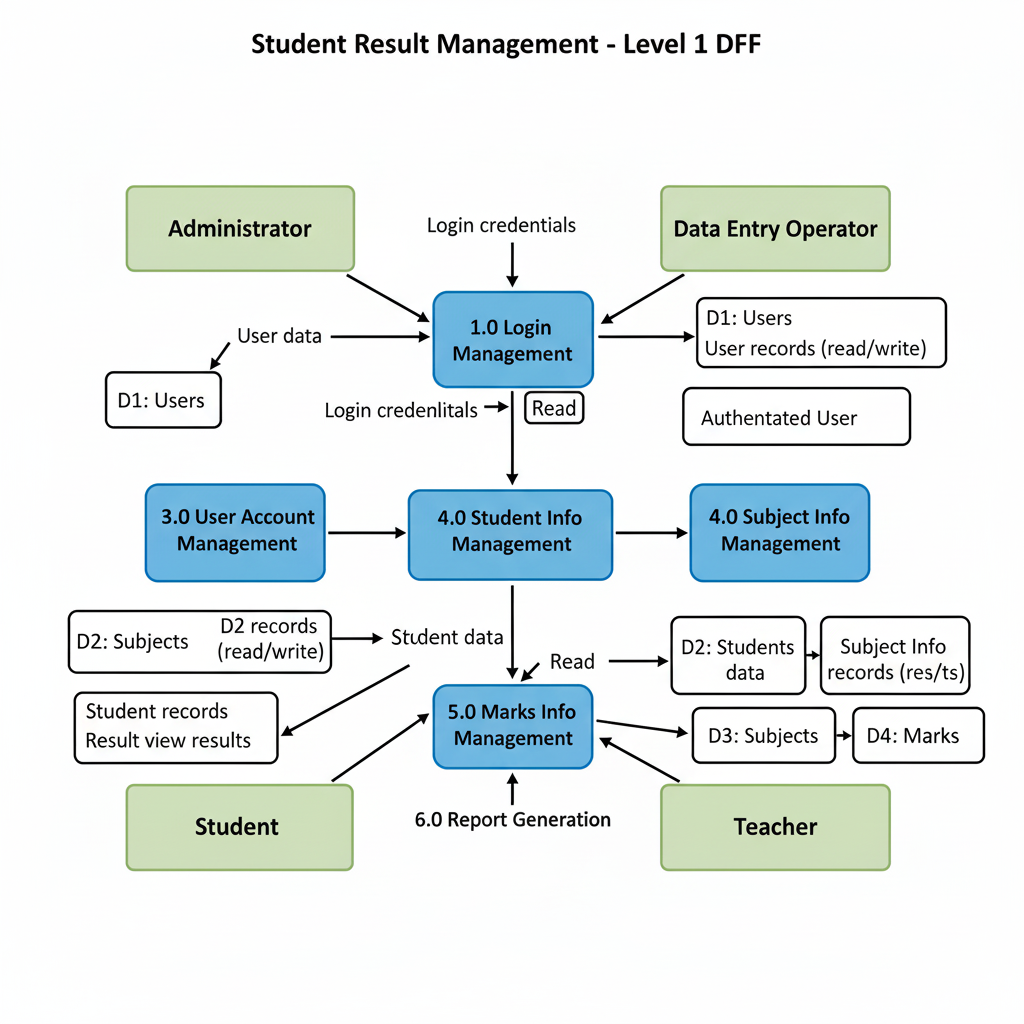
##### 3.2.1 Level 0 DFD

This DFD breaks down the main process into sub-processes like Login, User Account Management, Subject Info Management, Student Info Management, Marks Info Management, and Report Generation, showing the data flow between them.



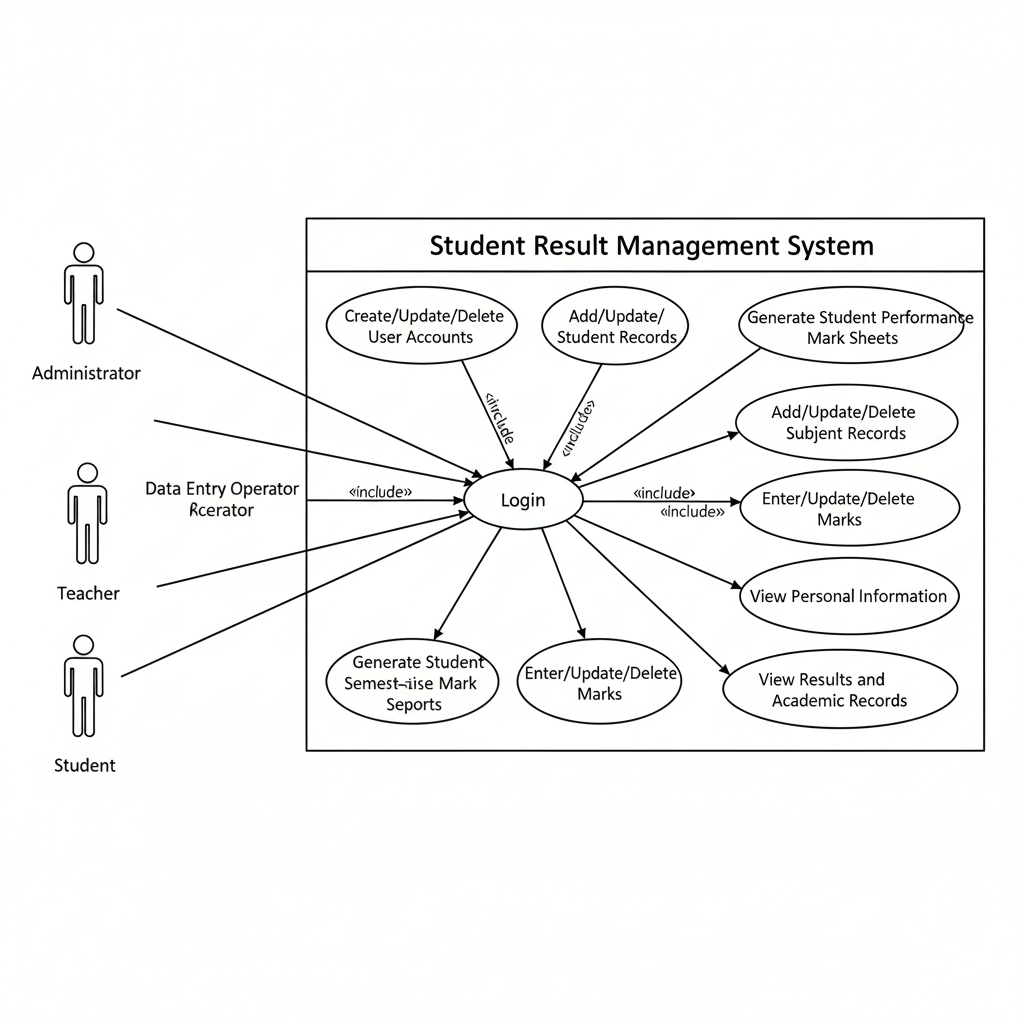
##### 3.2.2 Level 1 DFDs

* **Login:** Shows how different users log in to the system.
* **User Account Management:** Details the process of creating, updating, and deleting user accounts by the administrator.
* **Student Information Management:** Illustrates how student data is entered, updated, and viewed.
* **Subject Information Management:** Shows the process of managing subject details.
* **Marks Information Management:** Details how marks are entered and processed for students.



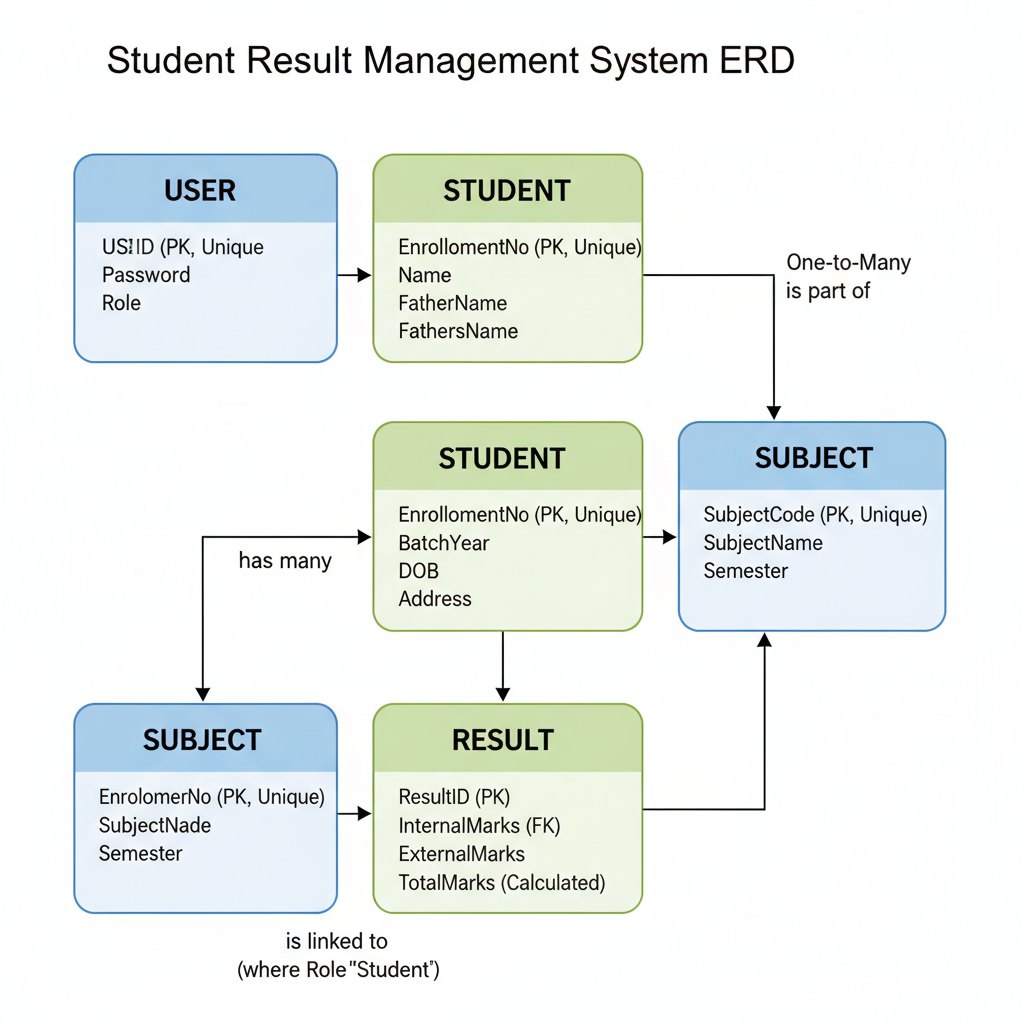
#### 3.3 Use Case Diagram

This diagram provides a high-level view of the functionalities available to different actors (users) in the system.



#### 3.4 Entity-Relationship Diagram (ERD)

The ERD illustrates the database schema, showing the relationships between entities like User, Student, Subject, and Result.



### 4. Interface Requirements

#### 4.1 User Interfaces

* **Login Screen:** Fields for User ID, Password, and Role (dropdown).
* **Administrator Dashboard:** Links to Maintain User Accounts, Generate Reports, and Reset System.
* **Data Entry Operator Dashboard:** Links to Manage Student Information and Manage Subject Information.
* **Teacher Dashboard:** Links to Manage Marks Information.
* **Student Dashboard:** Links to View Personal Information, View Academic Record, and View Result.
* **Data Entry Forms:** Consistent and user-friendly forms for entering and editing data with clear labels and validation messages.
* **Report Views:** Clean and printable layouts for all generated reports.

#### 4.2 Hardware Interfaces

* **Processor:** Intel P4, 2GHz or higher
* **RAM:** Minimum 1GB
* **Hard Disk:** Minimum 20GB of free space

#### 4.3 Software Interfaces

* **Operating System:** Windows XP/Vista/7
* **Development Environment:** Visual Studio 2010
* **Database:** SQL Server 2008

### 5. Functional Requirements

#### 5.1 User Management (FR-01)

* **FR-01.1:** The Administrator shall be able to create, view, update, and delete user accounts.
* **FR-01.2:** Each user shall have a unique User ID and a password.
* **FR-01.3:** The system shall support four user roles: Administrator, Data Entry Operator, Teacher, and Student.

#### 5.2 Student Information Management (FR-02)

* **FR-02.1:** The Data Entry Operator shall be able to add, update, and delete student records.
* **FR-02.2:** Each student record shall include Enrollment No, Name, Batch Year, Father's Name, DOB, and Address.
* **FR-02.3:** The Enrollment number shall be unique for each student.

#### 5.3 Subject Information Maintenance (FR-03)

* **FR-03.1:** The Data Entry Operator shall be able to add, update, and delete subject details.
* **FR-03.2:** Each subject record shall include Subject Code, Subject Name, and Semester.
* **FR-03.3:** The Subject Code shall be unique.

#### 5.4 Marks Information Maintenance (FR-04)

* **FR-04.1:** The Teacher shall be able to enter, update, and delete marks for students in their assigned subjects.
* **FR-04.2:** The system shall store internal and external marks.
* **FR-04.3:** The system shall automatically calculate the total marks (Internal + External).

#### 5.5 Report Generation (FR-05)

* **FR-05.1:** The Administrator shall be able to generate student performance reports.
* **FR-05.2:** The Administrator shall be able to generate semester-wise mark sheets for all students.
* **FR-05.3:** Students shall be able to view their own results and academic records.

### 6. Non-Functional Requirements

#### 6.1 Security

* The system shall be password-protected.
* Access to different modules shall be restricted based on user roles.

#### 6.2 Performance

* The system should respond to user requests within 3 seconds under normal load.
* The database should be able to handle records for at least 1000 students without significant degradation in performance.

#### 6.3 Maintainability

* The application shall be designed in a modular way to allow for easy incorporation of new requirements.

#### 6.4 Portability

* The system should be easily deployable on any machine that meets the specified hardware and software requirements.

#### 6.5 Reliability

* The system should be available 99% of the time during university working hours.
* The system shall have mechanisms for data validation to ensure data integrity.