SOFTWARE REQUIREMENT SPECIFICATION DOCUMENT
STUDENT RESULT MANAGEMENT SYSTEM

Introduction:

The requirements and specifications for the are provided in this paper in full. SRMS stands for Student Result Management System. The SRMS is intended to promote effective management and monitoring of academic records and outcomes for students in educational establishments for students. It manages the marks of all the students across multiple departments. The students can view all their marks of all the years at the same time. It also converts the marks into corresponding grades and it will generate the final report.

Purpose:

This Software Requirements Specification (SRS) document's main goal is to:

• Specify the Student Result Management System's goals and objectives.

• Clearly state the software's functional and non-functional needs.

• Foster communication among the development team, educational institutions, users, and users with reference to the capabilities and functionality of the system.

• Act as a base for the creation, creation, testing, and deployment of SRMS.

This document is to specify the requirements of the student result management web application.

The main purpose of the project is to:

1. Store the information of the users.
2. Store subjects of the various departments.
3. Store the marks obtained by the student in each subject.
4. Generate the report.

Scope:

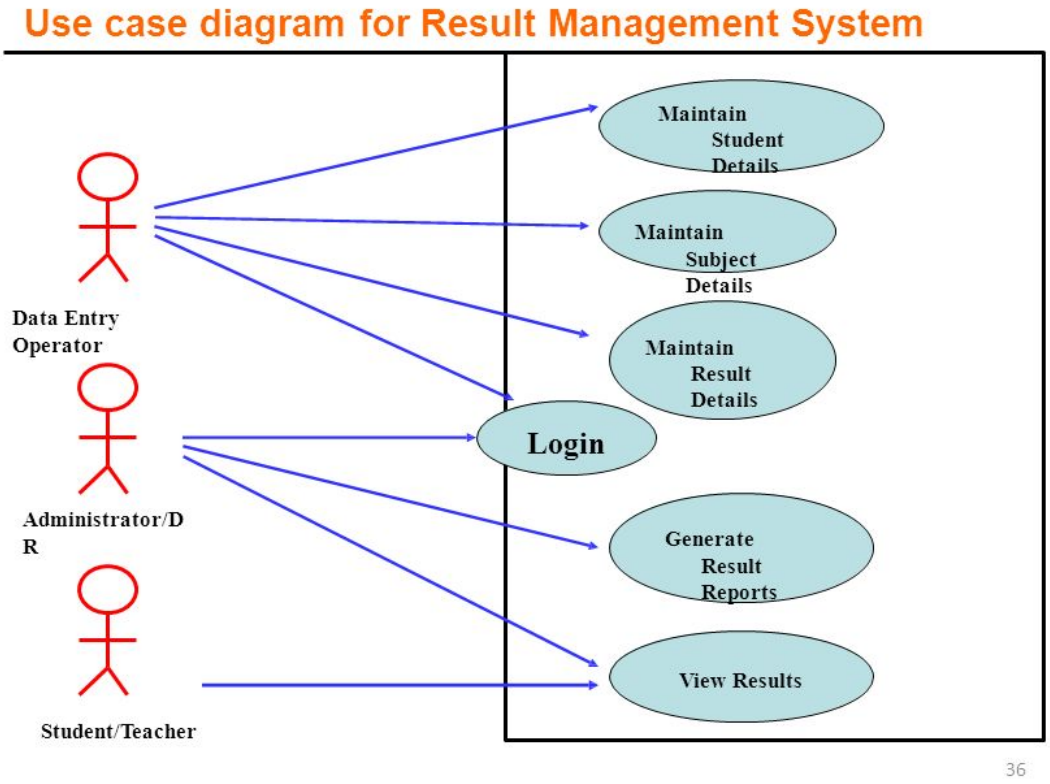
The following essential aspects are what the Student Result Management System strives to offer:

• For administrators, teachers, and students, role-based access management and user authentication.

• Managing student profiles, which contain personal data and enrollment information.

• Recording and archiving course data, such as topics, curricula, and grading standards.

• The input and computation of test, assignment, and assessment results for students.



Overall description:

A complete web-based tool called the Student Result Management System (SRMS) was created with the goal of streamlining and improving the management, processing, and distribution of student academic results inside educational institutions. In order to store and retrieve academic records, test results, and other data, the system intends to offer administrators, teachers, and students a productive and user-friendly platform of performance analytics.

Assumptions and Constraints:

The following presumptions and limitations govern the creation of the Student Result Management System:

• Modern web technologies will be used in the system's development to guarantee compatibility with common web browsers.

• The hosting environment's hardware infrastructure will support the system's performance and security needs. Sensitive student information will be stored in the system's database, and adequate security measures will be put in place to secure it.

The SRMS will abide by applicable data protection legislation and privacy laws, guaranteeing the confidentiality of user data.

Features and requirements:

Functional requirements:

Access Control and User Authentication: Users must be able to sign in using special usernames and passwords.

• Role-based access control must be offered by the system, allowing for the separation of administrators, teachers, and students.

• While teachers and students should have access to their individual features, administrators should have full access to the system's functionality.

User profiles including contact information, personal information, and enrollment information should be able to be created and updated by users.

Administrators should be able to handle student records in the system, including admissions, transfers, and graduations.

The ability to develop and manage courses, including their titles, descriptions, prerequisites, and credit hours, should be available to teachers and administrators.

• The system must enable the pairing of students with specific courses and preserve correct enrollment data.

• **Assessment and Grading:**

Exam, quiz, assignment, and project scores must be submitted by teachers and updated as needed.

• The system ought to compute and save cumulative grades in accordance with predetermined grading standards.

The ability to develop and publish individual student result reports based on assessment results should be available to teachers and administrators.

• Students should have secure login access to their individual result reports.

• **Analytics and Progress Tracking:** The system should offer features that allow teachers and administrators to monitor students' progress over the course of several terms or semesters.

• It should be possible to uncover trends and patterns in student performance using statistical analysis and visualizations.

• Notifications, emails, and other forms of communication between teachers, administrators, and students should be supported by the system.

Tracking of Attendance: Teachers should be able to record students' attendance in courses and other events.

• The system ought to keep track of attendance and offer information on attendance rates.

Student registration for courses should be possible during designated registration periods.

• When allocating courses to students, the system ought to prevent registration conflicts and enforce prerequisites.

Data management and reporting: • Administrators should be able to provide a variety of reports, including class rosters, enrollment data, and GPA distributions.

• For external analysis and reporting, the system should support exporting data in common formats.

Administration of the system: Administrators must be able to control user accounts, roles, and permissions.

• The system ought to offer resources for configuring academic periods, grading schemes, and assessment standards.

Backup and Data Security:

To prevent data loss, the system should back up data on a regular basis. Sensitive data should be stored securely, and suitable access controls should be in place. An overview of the main attributes and capabilities of a student result management system is given by these functional requirements. Based on the unique requirements and objectives of your project, modify and add to these specifications.

Non-Functional requirements:

Performance:

The system must offer quick user interaction responses, with a page load time of no more than two seconds.

• It should manage user interactions amongst concurrent users effectively, supporting at least 100 concurrent users without noticeably degrading performance. Ability to scale

• The system needs to be built to handle future increases in user volume and student enrollment.

• Within a single academic year, it should be able to handle a 20% growth in the number of enrolled students.

The system must be reliable, with a 99.9% uptime requirement and announced periodic maintenance windows.

• It must automatically restore after system failures or crashes without erasing any data.

Security:

User data, including personal data and test results, must be encrypted and stored in a secure manner.

Usability: The user interface should be simple to use and straightforward, needing little in the way of teacher, administrator, or student training.

• The system must follow accessibility standards so that people with impairments can navigate and communicate successfully.
The system should be responsive on many device types (desktops, tablets, and smartphones) and compatible with current web browsers like Chrome, Firefox, and Safari.

• It must support the widespread operating systems used by educators, administrators, and students.

• Data Integrity: The system should guarantee that test results and student records are accurate and consistent.

• Data entry errors or incomplete entries should be prevented through data validation methods.

Data Backup and Recovery:

To avoid data loss in the event of hardware failure or other emergencies, the system must regularly automate data backups.

• An audit record of significant user actions, such as data revisions or access attempts, should be kept by the system.

• Audit logs should be accessible to administrators for examination and analysis.

• Adherence to Regulations:

• The system must abide by applicable privacy and data protection legislation, such as HIPAA or GDPR.

• It ought to offer ways for users to control their consent to the processing of their personal data.

Support and Upkeep:

• The system should have a clear maintenance strategy that includes frequent updates and bug corrections and feature upgrades.

• There should be guaranteed response times for user support during business hours.24 hours for urgent matters.

• Proofreading:

• There should be thorough user documentation and instruction materials available.educators, administrators, and students.

External interface specifications:

1. User Interfaces: The system must have an intuitive web-based user interface that is accessible through popular web browsers (Chrome, Firefox, and Safari).

- Desktops, tablets, and smartphones are just a few examples of diverse screen sizes that the user interface must be responsive to.

2. Authentication and Authorization: To enable single sign-on for users, the system must interact with a third-party authentication provider (such as LDAP or OAuth).
- To regulate user access to system features and permissions, role-based access control must be developed.

3. External Systems Integration: To synchronize student enrolment data, the system must interface with the institution's Student Information System (SIS).
- Email server integration will enable users to get notifications and alerts.

4. Communication Interfaces: To notify students and administrators automatically, the system must employ conventional email protocols (SMTP).
In order to integrate with external applications and facilitate data interchange and reporting, it must offer RESTful APIs.

5. Data Import and Export: The system must handle importing student data from CSV or Excel files, including profiles and enrolment data.
- Administrators will be able to export result reports and other pertinent data in formats like PDF and Excel.

6. Reporting Interfaces: The system must have an integrated reporting module that can provide predefined reports such as class rosters, GPA distributions, and attendance summaries.

7. Payment Gateways, if Any:
- If the system accepts fee payments, it must link with a safe payment gateway to speed up the processing of online payments.
- To protect payment data, payment interfaces must adhere to industry security requirements (such as PCI DSS).

8. Third-Party Services: To track user engagement and system performance, the system may integrate with third-party analytics and monitoring services (such as Google Analytics).
- Consider social media integration for sharing system upgrades and accomplishments.

9. SMS Notifications (Optional): By integrating with SMS gateways, the system may be able to text users with vital notifications and alerts

10. Learning Management System (LMS) Integration: If applicable, the system must link with the institution's LMS to synchronize information about the courses and student enrollments.

Glossary and references:

• **Student Result Management System (SRMS):** This software programme is being created to manage the academic performance of students in educational institutions. Software Requirements Specification the description of the functional requirements, or specification as well as the SRMS's non-functional criteria.

• Use Case: An account of user interactions with the ARMS that illustrates how the system will be used to accomplish particular objectives.

• User Role: A predetermined set of access rights and permissions that specify the a user may carry out within the system.

• Assessment Score: The quantitative or qualitative rating given to a student's performance on a performance in a certain test or task.

• Grading Criteria: The predetermined criteria applied to student evaluation final grades and evaluate performance. Application Programming Interface (API) is a collection of tools and protocols that enables various software.

M.HARINI
STUDENT RESULT SOFTWARE MANAGEMENT

.

.