Software Requirements Specification

for

Result System of Internal Exams

Version 1.0 approved

Prepared by Group B

University of Rajshahi

16 Feb 2022

Group B members:

- **Mohammad Hasibul Hasan**   **1810676124**
- **Shahadat Hossain**   **1810676134**
- **Al Amin Kawsar**   **1810576130**
- **Abdur Rahim Sheikh**   **1810576141**
- **Abdullah Al Mamun**   **1810576118**
- **Johirul Islam**   **1810876145**
- **Humayun Ahmad Rajib**   **1810276140**
- **Sojib Mia**   **1810876123**
- **Umme Habiba Rea**   **1812076108**
- **Prithu Rani Roy**   **1812076138**
- **Bappi Kumar**   **1611176112**
- **Devit Chandra Roy**   **1710776108**

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1.   Introduction

1.1   Purpose

The purpose of the project is to make a viable result system for varsity internal exams which will provide all information regarding different level admin roles in the result system and also reduce system traffic.

1.2   Document Conventions

This document uses the following conventions.

DB	Database
DDB	Distributed Database
ER	Entity Relationship

1.3   Intended Audience and Reading Suggestions

This project is a prototype for the result management system of a University's internal examination and it is restricted within the university premises. This has been implemented under the guidance of a university professor. This project is useful for different layers of the administrator team within the university and as well as to the students.

1.4   Project Scope

The purpose of the online result management system is to ease the result system and to create a convenient and easy-to-use interface system for both administrators and students. The system is based on a relational database with students' information provided by the top level administrator and results' information system managed by the individual department offices. As well as, students' access to check result history and get in printable format.

1.5   References

Websites Link:
   1. *IEEE Recommended Practice for Software Requirements Specification (IEEE-STD-830-1998). Available at http://ieeexplore.ieee.org*
   2. *Study from Geeks for geeks website.*
       https://www.geeksforgeeks.org/software-requirement-specification-srs-format/
   1. *Further study from https://www.javatpoint.com/software-requirement-specifications*

2. *Sample SRS study from*
    https://www.cse.msu.edu/~chengb/RE-491/Papers/SRS-BECS-2007.pdf

Books references:
1. *Software Engineering : A practitioner's approach Ed. By Pressman, Roger.*

2.   Overall Description

2.1   Product Perspective

A result processing system provides the following information.

- ***Student Information***

    Including information about name, ID, department name, attached hall, session, semester, contact number.

- ***Marks and result information***
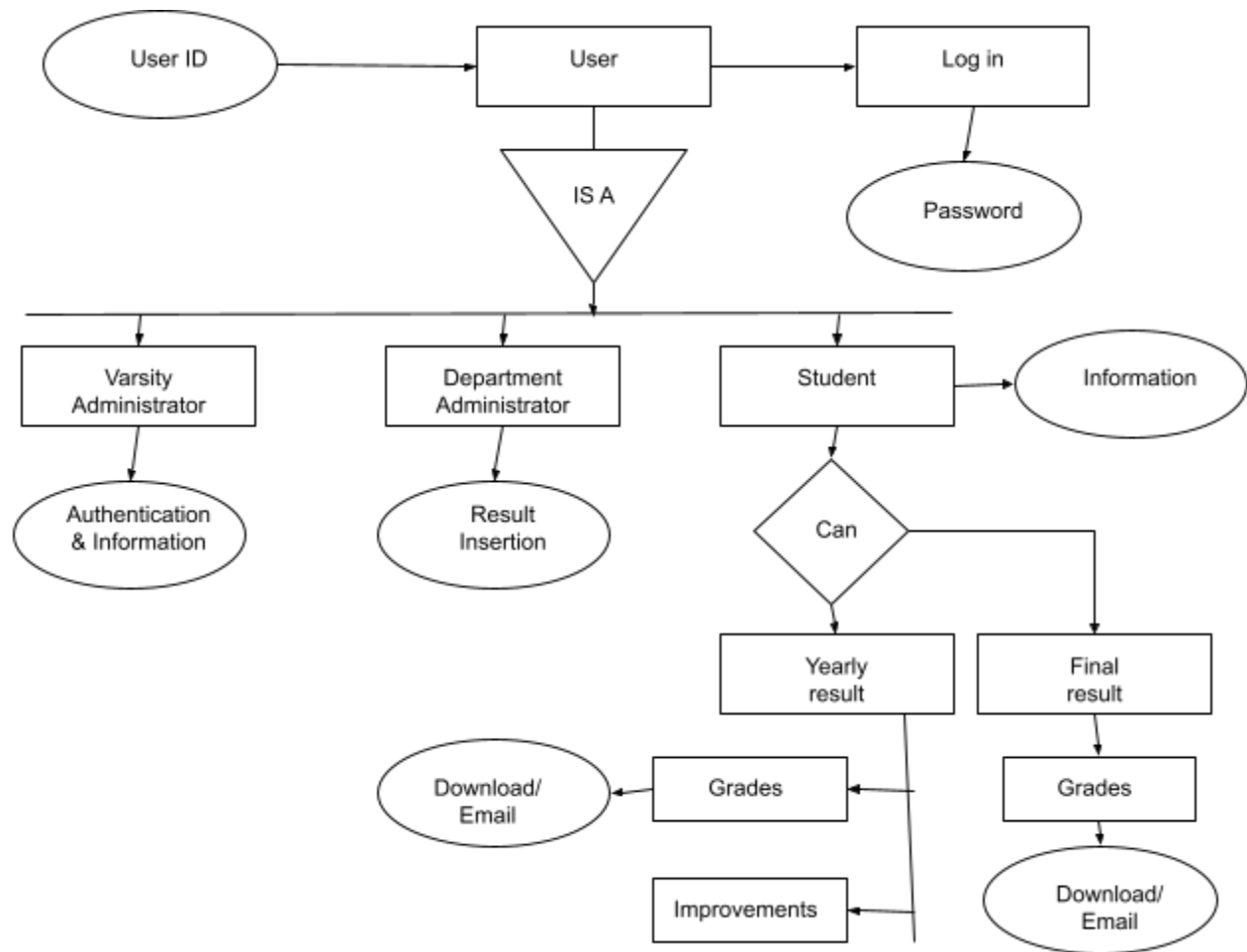
    It includes semester, courses enrolled, marks and grading for distinctive courses.

- ***Credits and improvements***

    It includes acquired and lost credits information, improvement courses, left improvement credits information

2.2   Product Features

The major features of Result System for Internal Exams as shown in below Entity-Relationship model (ER model)

E-R Model

## 2.3 User Classes and Characteristics

Students as users should be able to retrieve his/her result history with enrolled courses, acquired credits with grades and marks and improvement information. The system will also support different level admin functionality such as marks insertion by department teachers and new student information insertion by University administrators.

The students should be able to do the following functions:

- *Login with ID*
- *Mark details*
- *Improvement details*
- *Improvement credits*
- *Download pdf of result*
- *Total credits*
- *Result graphs*

The Admins should have following management functionalities:
- *Department teachers*
  - *Login with ID*
  - *Mark insertion*
  - *Improvement result*
  - *Yearly result*
  - *Download result sheet*
  - *Final result*
- *University administrators*
  - *Login with ID*
  - *Approvement of result publication*
  - *New student information insertion*
  - *Maintenance*
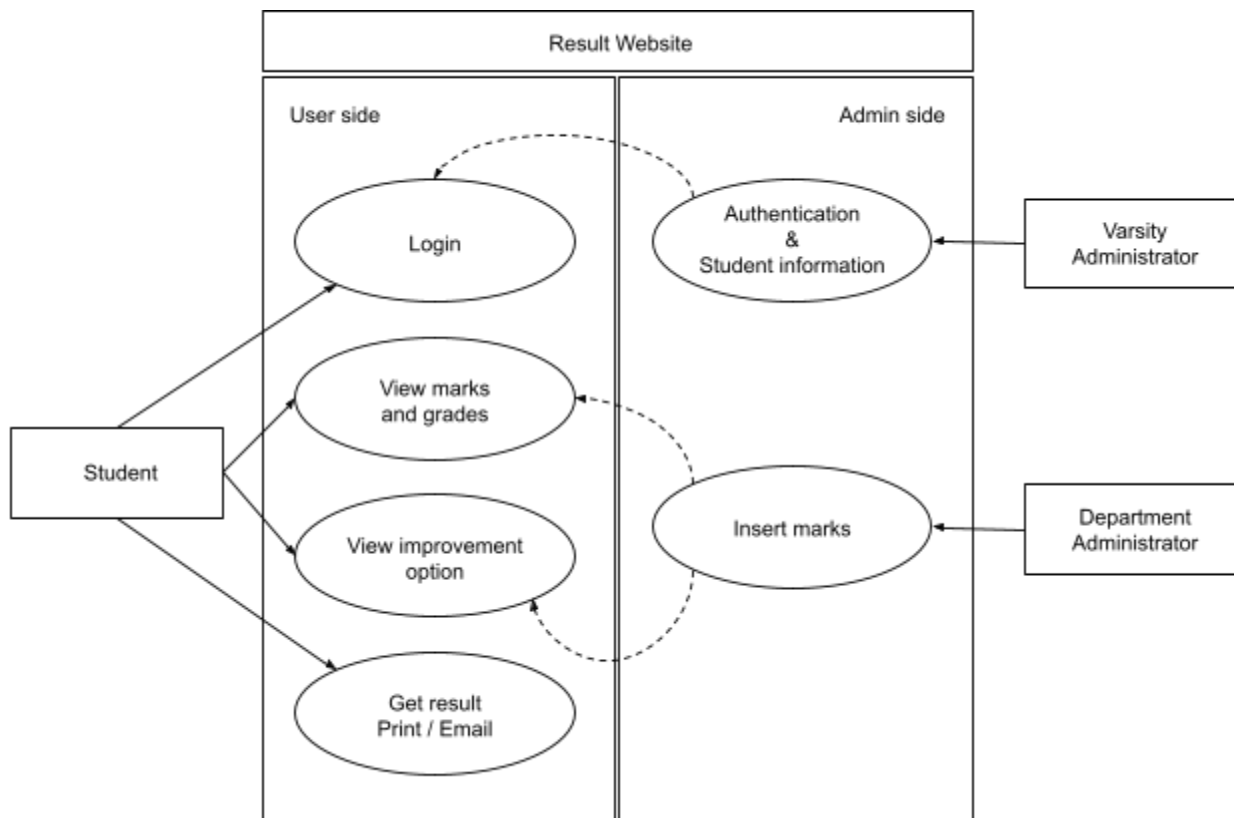
UML Diagram

2.4   Operating Environment

- *distributed database*

- *client/server system*

- *Operating system: Windows, Android*

- *database: sql + database*

- *platform: Kotlin/Java/PHP*

2.5   Design and Implementation Constraints

1. *The global schema, fragmentation schema, and allocation schema.*
2. *SQL commands for above queries/applications*
3. *How the response for application 1 and 2 will be generated. Assuming these are global queries. Explain how various fragments will be combined to do so.*
4. *Implement the database at least using a centralized database management system.*

2.6   Assumptions and Dependencies

Let us assume that this is a result system and it is used in the following application:

- *Providing information and authentication for newly admitted student*
- *Inserting course marks and grading information*
- *View marks and grades and downloadable version of the result*

3.   System Features

3.1   System Feature 1

3.1.1   Description and Priority

The main objective of the system is easy management of internal results. The system will provide all information regarding their role in the result system and also reduce system traffic . System users can easily maintain their information . The system has high security and scalability.

    3.1.2   Stimulus/Response Sequences

There are two user students and admins. Admin have all aceess of system and also can update any data of system. Student can see his/her results and other information.

    3.1.3   Functional Requirements

- *User Story-1: Student*
    1. *Logging in with ID and its verification*
    2. *Apply for the Examination/Improvement.*
    3. *See result details/Improvement Details*
    4. *Download/Get email the pdf of the result.*

- *User Story-2: Department Teacher*
    1. *Logging in with ID and its verification*
    2. *Mark Insertion in the database.*
    3. *Improvement Result*
    4. *Download Result sheet(Final/Yearly)*

- *User Story-3: University administrators*
    1. *Logging in with ID and its verification*
    2. *Approvement of result publication*
    3. *New Student information insertion.*
    4. *Old Student information Deletion.*
    5. *Control Student Database.*
    6. *Maintenance.*

4. External Interface Requirements

4.1 User Interfaces

- *Front-end software: HTML, Javascript, Android studio*
- *Back-end software: SQL, PHP, Kotlin*

4.2 Hardware Interfaces

- *Windows, Android*
- *A browser which supports CGI, HTML & Javascript.*

4.3   Software Interfaces

Software used	Description
Operating system	We have used web platform as operating system to cover a wide variety of device ranges as well as android platform which is easily reachable by our user base
Database	To save information about student, course and result
HTML, Javascript, PHP	To implement the project in a web platform we have chosen these languages for its more interactive support
Kotlin	To implement the project in an android platform we have chosen kotlin language for its more interactive support

4.4   Communications Interfaces

This project supports all types of web browsers and android devices.

5.   Other Nonfunctional Requirements

5.1   Performance Requirements

The steps involved to perform the implementation of the result database are as listed below.

A. E-R diagram

The E-R Diagram constitutes a technique for representing the logical structure of a database in a pictorial manner. This analysis is then used to organize data as a relation, normalizing relation and finally obtaining a relation database.

B. Normalization

The basic objective of normalization is to reduce redundancy which means that information is to be stored only once. Storing information several times leads to wastage of storage space and increase in the total size of the data stored.

If a database is not properly designed it can give rise to modification anomalies. Modification anomalies arise when data is added to, changed or deleted from a database table. Similarly, in traditional databases as well as improperly designed

relational databases, data redundancy can be a problem. These can be eliminated by normalizing a database.

Normalization is the process of breaking down a table into smaller tables. So that each table deals with a single theme. There are three different kinds of modifications of anomalies and formulating the first, second and third normal forms (3NF) is considered sufficient for most practical purposes. It should be considered only after a thorough analysis and complete understanding of its implications.

5.2   Safety Requirements

If there is extensive damage to a wide portion of the database due to catastrophic failure, such as a disk crash, the recovery method restores a past copy of the database that was backed up to archival storage (typically tape) and reconstructs a more current state by reapplying or redoing the operations of committed transactions from the backed up log, up to the time of failure.

5.3   Security Requirements

Security systems need database storage just like many other applications. However, the special requirements of the security market mean that vendors must choose their database partner carefully.

5.4   Software Quality Attributes

***AVAILABILITY:** The result system should be available 24 hours as well as in high traffic when publicating a result.*

***CORRECTNESS:** The system must provide flawless data and accurate in terms of roundness*

***MAINTAINABILITY:** The administrators should maintain correct schedules of updating results.*

***USABILITY:** The process of publication should satisfy a maximum number of customers' needs.*