RESULT PROCESSING SYSTEM

SOFTWARE ENGINEERING LAB

Submitted by:

1. **Kazi Md Nur Alam Jowel**
   6th semester
   Registration no: 20502004836

2. **Md Asikur Rahman**
   6th semester
   Registration no: 20502004860

Submitted to:

MD. Habibur Rahman
Faculty Member
Computer Science and Engineering

Chapter-1

RESULT PROCESSING SYSTEM

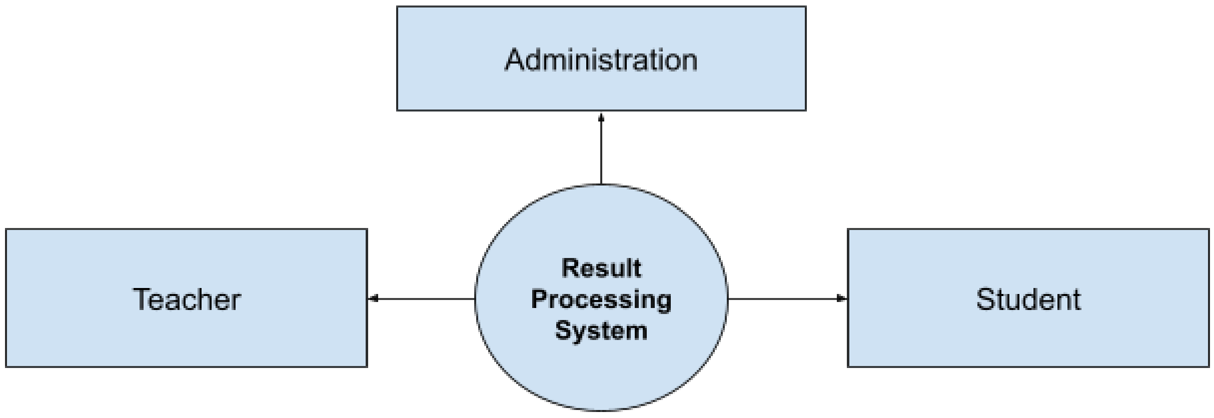
1 Introduction

This document lays out a project plan for developing the “**Result Processing System**”.

In the current result processing system, most of the work is done manually using paperwork, which makes the entire process time-consuming and error-prone. Manually managing records for a large number of students often leads to piles of files, making it hard to organize and retrieve data when needed.

This manual approach also increases the risk of losing important information due to human error, misplacement, or damage. Moreover, there's often a mismatch between the amount of work involved and the available manpower, which causes delays in processing results and affects their accuracy.

Our proposed software aims to solve these issues by automating the entire result processing workflow. It will efficiently store and manage student records, automatically calculate percentages and grade points, and significantly reduce the time and effort needed to publish accurate results. With this system, the result declaration process becomes faster, more reliable, and much easier to handle.



1.1 Problem Definition

In many institutions, the current process of handling exams and processing student results is still done manually, involving a lot of paperwork. This makes it difficult and time-consuming to manage large volumes of student data, often leading to stacks of files that are hard to organize and maintain.

With everything being handled by hand, there's always a risk of losing important information due to errors, misplacement, or damage. On top of that, the amount of work required usually doesn't match the available manpower, which leads to delays and mistakes when publishing results.

Our system is designed to solve these issues by digitizing the entire result processing workflow. It will automatically store student records, calculate percentages and grade points, and help administrators generate results quickly and accurately. This will not only reduce the need for manual effort but also speed up the entire result declaration process.

1.2 Purpose

The purpose of the **Result Processing System** is to create an efficient result publishing that enables administrators, teachers, and students to manage their results, publish results, and view results, select, and purchase items effortlessly. Key purposes are:

- The purpose of this project is to create a simple and reliable Result Processing System that makes it easier for students, teachers, and administrators to manage academic results.
- Instead of relying on manual work and paperwork, this system will handle everything digitally — from storing student records to calculating grades.

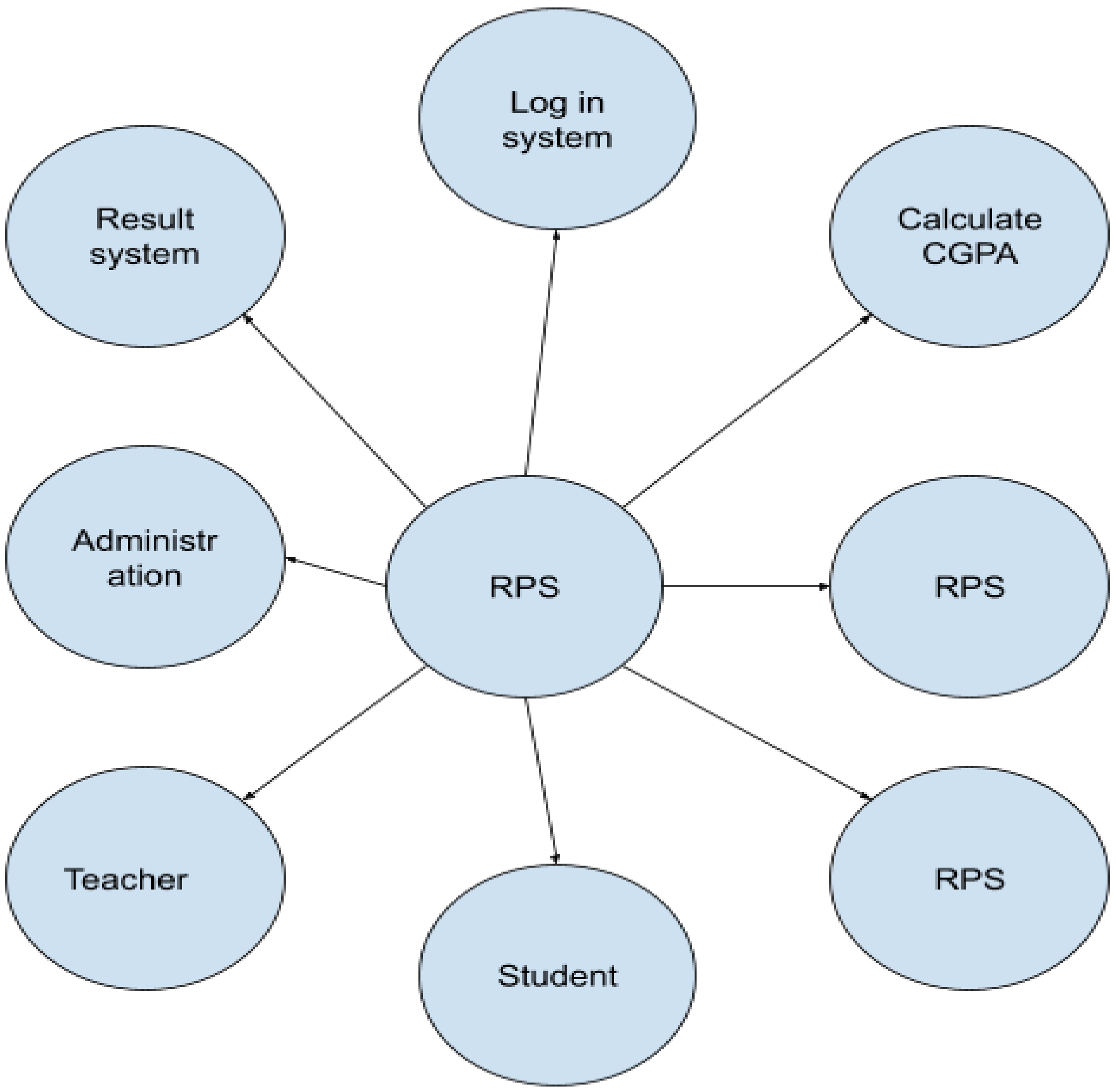
- The aim is to save time, reduce errors, and make the whole process of managing and publishing results faster and more efficient for everyone involved.

1.3 Scope

The **Result Processing System** is designed to handle the full lifecycle of academic result management in an educational institution. It will provide a digital platform to simplify and automate the tasks associated with student assessments, grading, and result generation.

- Student, teacher, and admin log in with role-based access
- Exam data entry and management
- Subject-wise mark entry by authorized personnel
- Automatic grade and GPA/CGPA calculation
- Generation of individual and batch result sheets
- Export of results in PDF/CSV format
- Result view portal for students
- Admin panel for user and data control

Context Diagram of an RPS:



1.4 User and Literature Survey

User:

i) University/College Administrators

ii) Faculty Members (Teachers)

iii) Examination Controllers

iv) Students

v) IT Support Staff

Literature Survey:

There are various academic management systems already in use, such as **EduPage**, **Fedena**, and **OpenSIS**, but many of them are either too complex for smaller institutions or lack features specifically focused on efficient result processing. Studies and user feedback suggest that systems with automated grade calculation, flexible data entry, and customizable reporting are more effective and widely adopted.

Modern web-based technologies like **Django**, **React**, and **Flask** are often used to create responsive and scalable result processing systems. These technologies help ensure that the system is fast, easy to use, and can handle large volumes of student data while allowing for future upgrades and feature expansion.

1.5 Definition and Abbreviations

RPS – Result Processing System

UI – User Interface

UX – User Experience

API – Application Programming Interface

CRUD – Create, Read, Update, Delete

DBMS – Database Management System

CGPA – Cumulative Grade Point Average

GPA – Grade Point Average

SQL – Structured Query Language

Admin Panel – A secured section of the system for managing users, results, and system settings

Semester – A half-year term in an academic session

Session – The academic year or period for which results are managed

Marksheet – A document or digital record containing the subject-wise marks and grades of a student

1.6 Books and Academic References

- Pressman, R. S., & Maxim, B. R. (2014). *Software Engineering: A Practitioner's Approach* (8th ed.). McGraw-Hill Education.

- Sommerville, I. (2016). *Software Engineering* (10th ed.). Pearson Education.

1.7 Overview

This project focuses on building a full-stack **Result Processing System** using Python (Django/Flask) for the backend and a simple, responsive frontend (HTML, CSS, JavaScript or React), with **MySQL** as the database. The system includes modules for student record management, marks entry, automatic grade calculation, result generation, and admin control.

The goal is to create a secure, fast, and user-friendly platform to streamline the result processing workflow for schools, colleges, and universities. It is designed to reduce manual work, prevent data loss, and ensure accurate and timely result publication.

Overall Description

2.1 Product Perspective

The **Result Processing System (RPS)** is a web-based application designed to digitize and simplify the process of managing student results. It is a modular, full-stack system developed using Python (Django or Flask) for the backend, with a responsive frontend built using HTML, CSS, and JavaScript (or optionally React). MySQL is used as the database to store all academic records securely.

While the system functions as a standalone platform, it is built with scalability in mind, allowing future integration with mobile applications or other institutional software like student portals or attendance tracking systems. The system is ideal for schools, colleges, and universities looking to replace manual result processing with a faster, more accurate, and more efficient digital solution.

2.1.1 System Interfaces

- **User Interface:**
  The user interface is designed to be clean, responsive, and role-based. It is built using standard web technologies like **HTML**, **CSS**, and **JavaScript** (or **React.js** if used). Different users—such as **administrators**, **teachers**, **students**, and **examination controllers**—will see different dashboards and features based on their roles, ensuring that each user only has access to what they need.
- **Hardware Interfaces:**
  The system can be accessed through any standard device like a **desktop**, **laptop**, or **tablet** with an internet connection. No special hardware is required, though basic peripherals like **printers** may be used for printing mark sheets or result reports.

- **Software Interfaces:**

  The front end communicates with the backend through **RESTful APIs**. The backend is developed using **Python (Django or Flask)** and is connected to a **MySQL** database to store all academic records. The architecture is modular and service-oriented, making it easy to maintain and extend in the future.