Green University of Bangladesh

Department of Computer Science and Engineering (CSE)
Semester: (Fall, Year: 2024), B.Sc. in CSE (Day)

Student Result Management System

Course Title: Database System Lab
Course Code CSE 210-CSE(181)
Section: 223 D3

Students Details

Name	ID
Md Arif Billah Mubin	223002008
-	-

Submission Date: 1-06-2024
Course Teacher's Name: Naznin

[For teachers use only: Don't write anything inside this box]

Lab Project Status	
Marks:	**Signature:**
Comments:	**Date:**

Contents

Chapter 1

Introduction

1.1 Overview

The Student Result Management System is a Java-based application designed to efficiently manage student records and grades. The system utilizes MySQL for database management, ensuring secure and reliable data storage. It features distinct interfaces for administrators and students, streamlining tasks such as adding, updating, deleting student records, and viewing grades. The project aims to enhance accuracy, data organization, and overall academic administration.

1.2 Motivation

The motivation behind the Student Result Management System is to leverage MySQL and Java technologies to create an efficient, reliable, and user-friendly solution for managing student records and grades.
The key drivers include:

1.Time-saving: Automating administrative tasks reduces the time spent on managing student data.
2.Accuracy and Reliability: Ensuring precise and consistent data management, minimizing human errors.
3.Data Organization: Structuring student information in an accessible and organized manner.
4.Performance Analysis: Facilitating the analysis of student performance through detailed records and reports.
5.Enhanced Communication: Improving the flow of information between administrators, faculty, and students. [1].

1.3 Problem Definition

1.3.1 Problem Statement

Managing student records and grades manually is time-consuming, prone to errors, and inefficient. Traditional methods lack accuracy, proper data organization, and effective communication channels between administrators, teachers, and students. The need for an automated, reliable system to handle these tasks is essential. Our project addresses this problem by developing a Student Result Management System that uses MySQL for secure and organized data storage, ensuring efficient management and accurate performance analysis.

1.3.2 Complex Engineering Problem

The Student Result Management System involves several complex engineering aspects that must be addressed:

Table 1.1: Summary of the attributes touched by the mentioned projects

Name of the P Attributess	**Explain how to address**
P1: Depth of knowledge required	Requires knowledge of Java programming, MySQL database management, and system design.
P2: Range of conflicting requirements	Balances user-friendliness with advanced features for both admins and students.
P3: Depth of analysis required	Needs detailed analysis for data accuracy and efficient database-query handling.
P4: Familiarity of issues	Must handle well-known issues like data security, user authentication, and data integrity.
P5: Extent of applicable codes	Adheres to coding standards for Java and SQL to maintain quality and readability.
P6: Extent of stakeholder involvement and conflicting requirements	Involves input from admins, teachers, and students, each with unique needs.
P7: Interdependence	Requires seamless integration of the user interface, database, and business logic.

1.4 Design Goals/Objectives

The main goal of the Student Result Management System is to create an easy-to-use, efficient, and scalable application for managing student records and grades. Objectives include:

1.User-Friendly Interface: Ensure the system is simple to navigate for both administrators and students.
2.Efficient Data Management: Streamline the processes of adding, updating, deleting,

and viewing student records.
3.Accurate Grade Tracking: Provide reliable and precise grade entry and calculation.
4.Effective Communication: Enhance information flow between administrators, teachers, and students.
5.Scalability: Design the system to support future growth and additional features.

1.5 Application

The Student Result Management System is used in educational institutions to manage student records and grades efficiently. It helps administrators handle tasks like adding, updating, and deleting student records, while students can easily view their grades and performance. This system enhances data accuracy, reduces administrative workload, facilitates performance analysis, and improves communication between students, teachers, and administrators.

Chapter 2

Design/Development/Implementation of the Project

2.1 Introduction

The Student Result Management System is a project aimed at making it easier to manage student records and grades in schools and colleges. Using Java for the main application and MySQL for the database, the system provides a reliable way for administrators to handle student data and for students to access their grades.

Manual management of student records is often slow and prone to mistakes. This project automates these tasks, saving time and reducing errors. The system combines software development, database management, and user-friendly design to create an effective solution. [2] [3] [4].

2.2 Project Details

The Student Result Management System is divided into two main interfaces: Admin and Student. Each interface provides specific functionalities to streamline the management of student records and grades. Below are the detailed descriptions of each interface and their respective functionalities.

2.2.1 Admin Interface

The Admin Interface is designed for school or college administrators to manage student records efficiently. The following features are included:

1.Add New Student

Description: Allows administrators to add new students to the database.

Functionality: Input fields for student name, roll number, class, and other relevant details.

2.Add Student Result

Description: Enables administrators to input and save student grades for various subjects.

Functionality: Input fields for roll number, subject, marks obtained, and grade.

3.Delete Student

Description: Allows the removal of student records from the database.

Functionality: Search and select student records to delete based on roll number.

4.See All Students' Results

Description: Provides a comprehensive view of all students' academic performance.

Functionality: Display table with student details and grades, sortable and searchable.

4.View Registered Students

Description: Lists all students currently registered in the system.

Functionality: Display table with student details, sortable and searchable.

5.Logout

Description: Securely logs out the administrator from the system.

Functionality: Ends the session and redirects to the login screen.

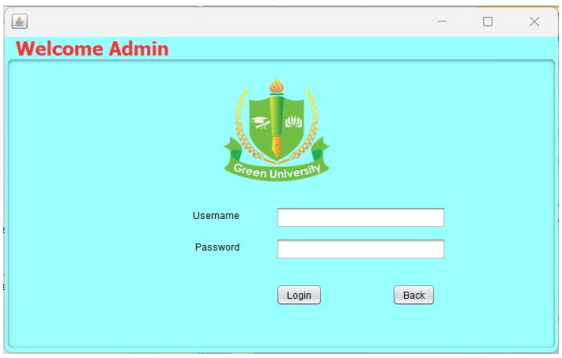

Figure 2.1: Admin Interface

2.2.2 Student Interface

The Student Interface is designed for students to access their academic information conveniently. The following feature is included:

1.Search Student by Roll Number

Description: Allows students to search for their academic results using their roll number.

Functionality: Input field for roll number; displays student details and grades upon submission.

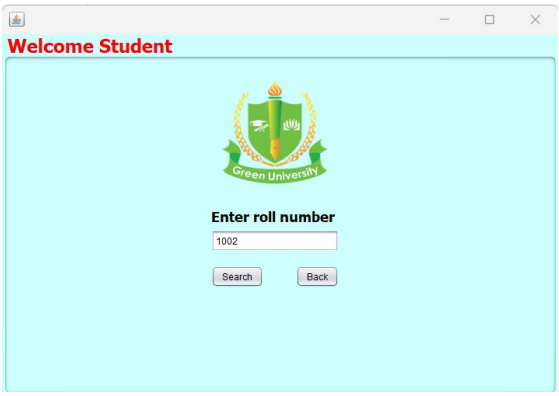

Figure 2.2: Student Interface

2.3 Implementation

This section provides detailed information on the implementation of the Student Result Management System, including workflow, tools and libraries used, and specific implementation details with screenshots and programming code.

2.3.1 Workflow

The workflow of the Student Result Management System involves several steps from logging in to managing and viewing student records. Here is an overview of the workflow:

User Authentication: Admin and student login interfaces for secure access.

Admin Operations: Adding, updating, and deleting student records; managing student results; viewing all registered students and their results.

Student Operations: Students can search for their results using their roll number.

The workflow

Tools and libraries

Implementation details (with screenshots and programming codes)

Each subsection may also include subsubsections.

2.3.2 Tools and Libraries

The development of the Student Result Management System involves various tools and libraries:

Java: Core programming language used for application development.

MySQL: Database management system for storing student records and results.

JDBC: Java Database Connectivity API for connecting and executing queries on MySQL database.

Swing: Java library for building graphical user interfaces.

JUnit: Testing framework for Java to ensure code quality and reliability.

2.3.3 Implementation Details

Below are the implementation details, including screenshots and code snippets, for key features of the project.

Database Creation and Admin Interface:

Database Table Creation:

The Student Result Management System uses MySQL to store and manage student records and results. Below are the SQL scripts for creating the necessary tables.

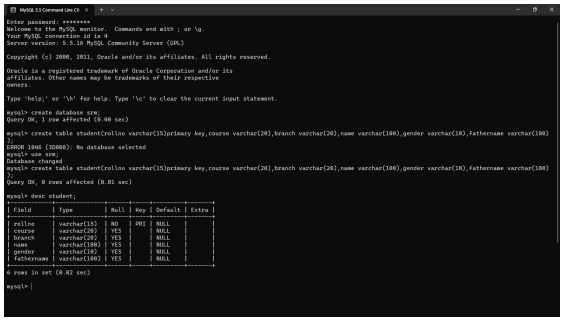

Figure 2.3: Schema of the students

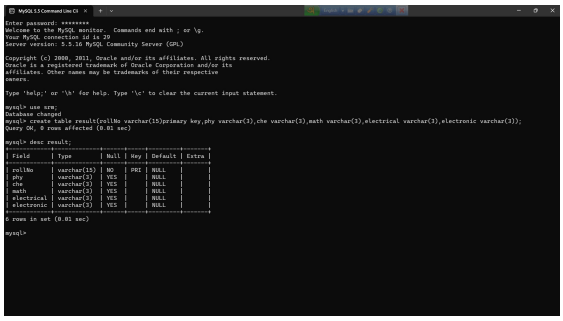

Figure 2.4: Schema of the students

Admin Interface:

Add New Student:
Code Snippet:

```
private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    String rollNo=jTextField1.getText();
    String phy=jTextField2.getText();
    String che=jTextField3.getText();
    String math=jTextField4.getText();
    String electrical=jTextField5.getText();
    String electronic=jTextField6.getText();
    try{
        Class.forName("com.mysql.jdbc.Driver");
        Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/srm","root","27432288");
        Statement st=con.createStatement();
        ResultSet rs=st.executeQuery("select *from student where rollNo='"+rollNo+"'");
        if(rs.next()){
            st.executeUpdate("insert into result (rollNo,phy,che,math,electrical,electronic) values"
                    + "('"+rollNo+"','"+phy+"','"+che+"','"+math+"','"+electrical+"','"+electronic+"')");
            JOptionPane.showMessageDialog(null, "Successfully Updated");
            setVisible(false);
            new insertNewResult().setVisible(true);
        }
        else
        {
            JOptionPane.showMessageDialog(null, "Roll number doess not exist in our database");
            setVisible(false);
            new insertNewResult().setVisible(true);
        }
    }
    catch(Exception e)
    {
      JOptionPane.showMessageDialog(null, "connection error");

    }
}
```

Figure 2.5: Algorithms

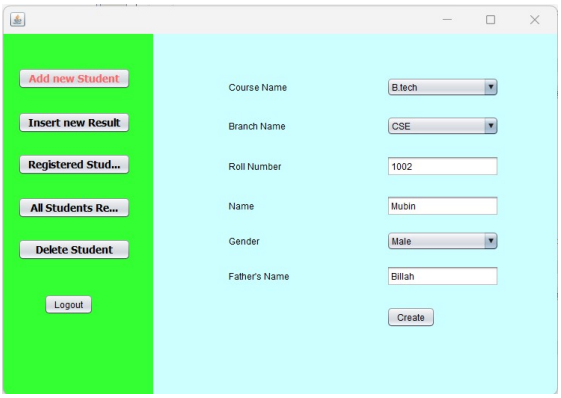

Figure 2.6: Interface

Add Student Result:
Code Snippet:

```
private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    String course=(String)jComboBox1.getSelectedItem();
    String branch=(String)jComboBox2.getSelectedItem();
    String rollNo=jTextField1.getText();
    String name=jTextField2.getText();
    String gender=(String)jComboBox3.getSelectedItem();
    String fatherName =jTextField3.getText();
    try{
        Class.forName("com.mysql.jdbc.Driver");
        Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/srm","root","27432288");
        Statement st=con.createStatement();
        st.executeUpdate("insert into student(course,branch,rollNo,name,gender,fatherName) values"
        + "('"+course+"','"+branch+"','"+rollNo+"','"+name+"','"+gender+"','"+fatherName+"')");
        JOptionPane.showMessageDialog(null, "Successfully Updated");
        setVisible(false);
        new adminHome().setVisible(true);
        st.close();
    }
    catch(Exception e)
    {
      JOptionPane.showMessageDialog(null, "This roll number is already exist");
      setVisible(false);
      new adminHome().setVisible(true);
    }
}
```

Figure 2.7: Algorithms

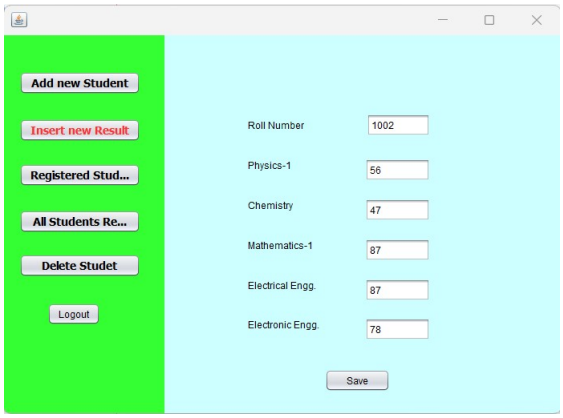

Figure 2.8: Interface

View All Students:
Code Snippet:

Figure 2.9: Algorithms

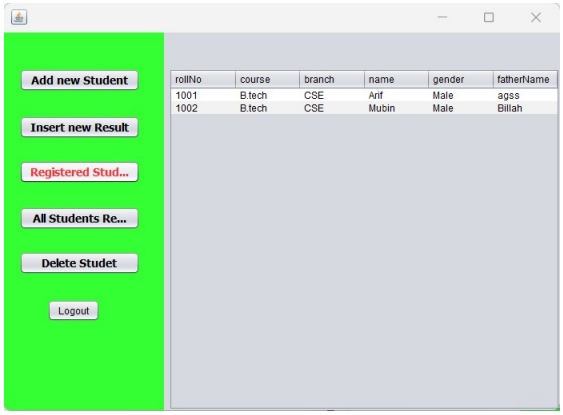

Figure 2.10: Interface

View All Students Results:

Code Snippet:

```
private void formComponentShown(java.awt.event.ComponentEvent evt) {
    // TODO add your handling code here:
    try
    {
        Class.forName("com.mysql.jdbc.Driver");
        Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/srm","root","27432288");
        Statement st=con.createStatement();
        ResultSet rs=st.executeQuery("select *from result");
        jTable1.setModel(DbUtils.resultSetToTableModel(rs));
    }
    catch(Exception e)
    {
        JOptionPane.showMessageDialog(null, "Connection error");
    }
    jTable1.setEnabled(false);
}
```

Figure 2.11: Algorithms

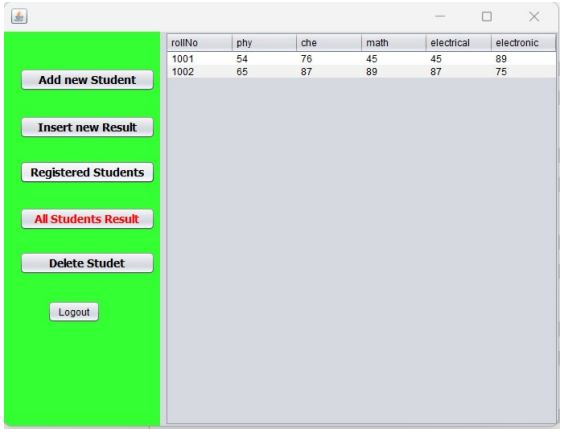

Figure 2.12: Interface

Student Interface

Search Student by Roll Number:
Code Snippet:

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    String rollNo=jTextField1.getText();
    try
    {
        Class.forName("com.mysql.jdbc.Driver");
        Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/srm","root","27432258");
        Statement st=con.createStatement();
        ResultSet rs=st.executeQuery("select *from Student inner join result where student.rollNo='"+rollNo+"' and result.rollNo='"+rollNo+"'");
        if(rs.next())
        {
            setVisible(false);
            new studentHome(rollNo).setVisible(true);
        }
        else
            JOptionPane.showMessageDialog(null, "Incorrect roll number");
    }catch(Exception e)
    {
        JOptionPane.showMessageDialog(null, "Connection error");
    }
}
```

Figure 2.13: Algorithms

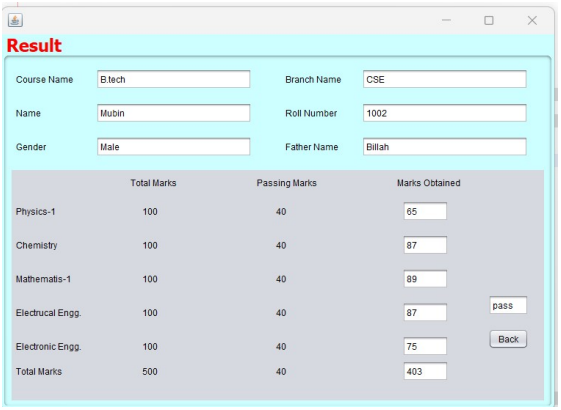

Figure 2.14: Interface

Chapter 3

Conclusion

3.1 Discussion

The project aims to streamline the process of managing student records and grades in educational institutions. Key components include the Admin Interface, which allows administrators to add, update, delete, and view student records and results, and the Student Interface, which enables students to view their academic information by searching with their roll number. The system uses Java for application logic and MySQL for database management, ensuring data accuracy and efficient handling. Through detailed workflows, code implementations, and database structures, this project demonstrates the successful development of a user-friendly, efficient, and scalable solution for academic management.

3.2 Limitations

Despite its benefits, the Student Result Management System has some limitations:

1.Scalability Issues: As the number of records grows, the system might face performance bottlenecks without proper optimization.

2.User Interface: The current interface, while functional, might not be very intuitive or visually appealing, requiring improvements for better user experience.

3.Security: Basic security measures are implemented, but the system could be vulnerable to more sophisticated attacks and thus needs enhanced security protocols.

4.Feature Limitations: The system currently handles only basic student records and result management. Advanced features like attendance tracking, detailed analytics, and automated notifications are not included.

5.Dependency on Database: The system heavily relies on MySQL. Any issues with the database server could disrupt the entire system's functionality.

3.3 Scope of Future Work

The Student Result Management System can be enhanced in several ways in the future:

1.Improved User Interface: Develop a more intuitive and visually appealing interface.

2.Enhanced Security: Implement advanced security features to protect data from sophisticated attacks.

3.Scalability: Optimize the system to handle a larger number of records efficiently.

4.Additional Features: Add functionalities such as attendance tracking, detailed performance analytics, and automated notifications.

5.Mobile Application: Create a mobile version of the system to increase accessibility for users on the go.

References

[1] Omid C Farokhzad and Robert Langer. Impact of nanotechnology on drug delivery. *ACS nano*, 3(1):16–20, 2009.

[2] Uthayasankar Sivarajah, Muhammad Mustafa Kamal, Zahir Irani, and Vishanth Weerakkody. Critical analysis of big data challenges and analytical methods. *Journal of Business Research*, 70:263–286, 2017.

[3] Douglas Laney. 3d data management: controlling data volume, velocity and variety. gartner, 2001.

[4] MS Windows NT kernel description. http://web.archive.org/web/20080207010024/http://www.808multimedia.com/winnt/kernel.htm. Accessed Date: 2010-09-30.