



Tribhuvan University

Faculty of Humanities and Social Sciences

RESULT MANAGEMENT SYSTEM

A PROJECT REPORT

Submitted to:

Department of Computer Application

Thames International College

Old Baneswor, Kathmandu

In partial fulfillment of the requirements for the Bachelors in Computer Application

Submitted by:

Rohit Yadav

BCA 4th Semester

University SN: 6-2-450-69-2022

Symbol No: 45002107

March 10, 2025

Roshan Rapacha Sunuwar

BCA 4th Semester

University SN: 6-2-450-71-2022

Symbol No: 45002109

March 10, 2025

Under the Supervision of

Mr. Prabin Maharjan



Tribhuvan University
Faculty of Humanities and Social Sciences
Thames International College
Old Baneswor, Kathmandu

Bachelor in Computer Applications (BCA)

SUPERVISOR'S RECOMMENDATION

I hereby recommend that this project prepared under my supervision by **Rohit Yadav and Roshan Rapacha Sunuwar** entitled “**Result Management System** ” in the Partial Fulfillment of requirement for the degree of Bachelor in Computer Application is recommended for that final evaluation.

Prabin Maharjan
Project Supervisor
Thames International College



Tribhuvan University
Faculty of Humanities and Social Sciences
Thames International College

LETTER OF APPROVAL

This is to certify that this project prepared by **Rohit Yadav and Roshan Rapacha Sunuwar** entitled “**Result Management System**” in partial fulfillment of the requirements for the degree of Bachelor in Computer Application has been evaluated. In our opinion it is satisfactory in the scope and quality as a project for the required degree.

Signature of Supervisor <hr/> <p>Mr. Prabin Maharjan (Supervisor) Faculty at SOIT School of Information Technology (SOIT) Old Baneswor, Kathmandu</p>	Signature of HOD <hr/> <p>Mr. Surendra Malla (HOD) Head of Department School of Information Technology (SOIT) Old Baneswor, Kathmandu</p>
Signature of Internal Examiner <hr/> <p>Internal Examiner Mr. Pesal Rai Faculty at SOIT</p>	Signature of External Examiner <hr/> <p>External Examiner Mr. Ramesh Singh Saud Tribhuvan University</p>

ABSTRACT

The Result Management System serves as a crucial tool in the academic landscape, streamlining the process of managing and generating student results efficiently. As the need for systematic and accurate result processing continues to grow, the necessity for an intuitive and efficient platform becomes evident. The documentation abstraction highlights the creation of the Result Management System, emphasizing its role in simplifying and automating result-related operations.

The system functions as a centralized platform where the admin is the sole user, responsible for performing CRUD (Create, Read, Update, Delete) operations and generating student results. It ensures that academic records are managed systematically, minimizing errors and improving efficiency. The platform is designed to be user-friendly, enabling the admin to handle results effortlessly, from data entry to final result publication.

In essence, the documentation abstraction encapsulates the purpose of the Result Management System, focusing on enhancing the accessibility, accuracy, and effectiveness of academic result processing through a structured and efficient digital solution.

Keyword: Java, MySQL, Database, XAMPP

ACKNOWLEDGEMENT

I would like to express my sincere appreciation and gratitude to all those who have contributed to the completion of the Result Management System project. This endeavor has been a significant journey of learning and growth, and I am thankful for the support and assistance that made it possible.

First and foremost, I am deeply grateful to my project supervisor **Prabin Maharjan** for their invaluable guidance, mentorship, and insightful feedback throughout the project's duration. Their expertise and encouragement kept me focused and motivated, ensuring the project's success.

I extend my thanks to **Thames International College** for providing the necessary resources, including access to libraries, software, and other facilities, which were crucial in the development and implementation of the project.

I also want to acknowledge the unwavering support of my friends and peers who provided encouragement and helpful discussions whenever needed. Their insights contributed significantly to shaping the project's direction. My heartfelt appreciation goes to my family for their constant encouragement, understanding, and patience. Their belief in my abilities was a driving force behind the completion of this project.

In the end, this project's successful completion stands as a testament to the combined efforts of those who supported and guided me. I am truly grateful for their contributions, and I look forward to applying the knowledge and skills gained from this project in my future endeavors.

Yours Sincerely,

Rohit Yadav

Roshan Rapacha Sunuwar

TABLE OF CONTENTS

SUPERVISOR'S RECOMMENDATION

LETTER OF APPROVAL

ABSTRACT	i
ACKNOWLEDGEMENT	ii
LIST OF ABBREVIATIONS	vi
LIST OF FIGURES	vii
LIST OF TABLES	viii
CHAPTER: 1 INTRODUCTION	1
1.1 Introduction	1
1.2 Problem Statement	2
1.3 Objectives	2
1.4 Scope and Limitation	2
1.4.1 Scope	3
1.4.2 Limitations	3
1.5 Report Organization	4
CHAPTER: 2 BACKGROUND STUDY AND LITERATURE REVIEW	5
2.1 Background Study.....	6
2.2 Literature Review	7

CHAPTER: 3 SYSTEM ANALYSIS AND DESIGN	8
3.1 System Analysis	8
3.1.1 Requirement Analysis	11
3.1.2 Feasibility Analysis	12
3.1.3 Data Modeling (ER-Diagram)	13
3.1.4 Process Modeling (DFD)	15
3.2. System Design	16
3.2.1. Architectural Design	16
3.2.2. System Flowchart	17
3.2.3. Database Schema Design	18
3.2.4. Interface Design (UI Interface)	20
3.2.5. Physical DFD	22
CHAPTER: 4 IMPLEMENTATION AND TESTING	23
4.1. Implementation	23
4.1.1. Tools Used (CASE tools, Programming language, Database platforms)	24
4.1.2. Implementation Details of Modules (Description of procedures)	24
4.2. Testing	25
4.2.1. Test Cases for Unit Testing	26
4.2.2. Test Cases for System Testing	31

CHAPTER: 5 CONCLUSION AND FUTURE RECOMMENDATIONS	32
5.1. Lesson Learnt / Outcome	32
5.2. Conclusion	33
5.3. Future Recommendations	33
REFERENCES	34
APPENDIX: SYSTEM SCREENSHOTS	

LIST OF ABBREVIATIONS

CRUD	Create, Read, Update and Delete
DFD	Data Flow Diagram
ERD	Entity Relationship Diagram
MySQL	My Structured Query Language
RMS	Result Management System
UI	User Interface

LIST OF FIGURES

Figure 3.1: Waterfall Methodology for RMS	8
Figure 3.2: Use Case Diagram of RMS	10
Figure 3.3: Gantt chart for RMS	12
Figure 3.4: Entity Relational Diagram for RMS	13
Figure 3.5: Level 0 DFD for RMS	14
Figure 3.6: Level 1 DFD for RMS	15
Figure 3.7: Architectural Design of RMS	16
Figure 3.8: System Flow chart of RMS	17
Figure 3.9: Database Schema of RMS	18
Figure 3.10: UI Home page of RMS	19
Figure 3.11: Login page of RMS	19
Figure 3.12: Dashboard of RMS	20
Figure 3.13: Physical DFD of RMS	22

LIST OF TABLES

Table 3.1: Gantt chart Table for RMS	11
Table 4.1: Test case for User Login of RMS.....	25
Table 4.2: Test case for User Registration of RMS.....	26
Table 4.3: Test Case for Student Registration Successful	26
Table 4.4: Test Case for Student Registration Failure of RMS	27
Table 4.5: Test Case for Successful Marks Entry of RMS	27
Table 4.6: Test Case for Marks Entry Failure of RMS	28
Table 4.7: Test Case for Successful Detail Deletion of RMS	28
Table 4.8: Test Case for Detail Deletion Failure of RMS	28
Table 4.9: Test Case for Successful Marks Deletion of RMS	29
Table 4.10: Test Case for Marks Deletion Failure of RMS	29
Table 4.11: Test Case for Successful Detail Update of RMS	29
Table 4.12: Test Case for Detail Update Failure of RMS	30
Table 4.13: Test Case for Successful Marks Update of RMS	30
Table 4.14: Test Case for Marks Update Failure of RMS	31

CHAPTER 1:

Introduction

1.1 Introduction

In the modern academic environment, ensuring efficient result management is crucial for maintaining accuracy and transparency. The Result Management System (RMS) is designed as an administrative tool to handle student records, manage grades, and generate academic results efficiently. This project focuses exclusively on administrative functionalities, including CRUD (Create, Read, Update, Delete) operations on student records and automated result processing.

The primary objective of the Result Management System is to establish a centralized platform for storing student details, subject-wise marks, and final grades. The system will streamline result computation, reduce manual errors, and generate performance reports, ensuring that administrators can efficiently manage academic records with ease.

In a dynamic academic setting, where time is of the essence, convenience and accessibility are crucial in enhancing efficiency and productivity. The Result Management System, designed especially for the Bachelor in Computer Application (BCA) program, provides a streamlined and efficient platform for managing student records and results. It simplifies tasks by enabling essential operations like adding, updating, and retrieving student data with ease.

With the introduction of automation, this system aims to enhance the efficiency of result processing, allowing for real-time updates, structured data management, and accurate computation of academic results. The system's interface will be user-friendly, specifically tailored for administrators to ensure a seamless experience in managing student performance records.

1.2 Problem Statement

Traditional result management systems often involve manual data handling, leading to errors, inefficiencies, and delays in publishing results. Administrators face challenges in maintaining accurate student records, computing grades manually, and generating performance reports efficiently. Additionally, the lack of a structured digital system makes it difficult to update or modify records without inconsistencies.

The Result Management System addresses these issues by providing a structured, digital platform for administrators. By integrating automated result computation, secure student record management, and efficient report generation, the system ensures that administrators can manage academic data effortlessly, reducing workload and improving accuracy in result processing.

1.3 System Objectives

It should define why the system is being developed for a particular environment.

Functional objectives

- To provide an efficient and automated platform for academic result management, exclusively for administrators.
- To enable CRUD operations for maintaining student records, subject marks, and final grades.

Organisational objectives

- To ensure structured data storage, reducing errors in result management.

1.4 Scope and Limitation

1.4.1 Scope

The scope of the Result Management System encompasses the design, development, and deployment of an administrative platform for handling academic results. The key components of the project include:

- Student Record Management: Storing and managing student information, including personal details and course enrollment.

- **CRUD Operations:** Allowing administrators to add, update, delete, and retrieve student records efficiently.
- **Automated Result Computation:** Systematically computing student grades based on subject-wise marks and predefined grading criteria.

1.4.2 Limitations

While the Result Management System enhances efficiency, certain limitations exist:

- The system is restricted to administrative use and does not provide access for students or teachers.
- No real-time notification feature for publishing results is available.
- The system does not include functionalities for student complaints or result re-evaluation requests.

1.5 Report Organization

Introduction

This chapter introduces the system, its objectives, limitations, and the rationale behind its development.

Background Study and Literature Review

This chapter discusses previous research and existing result management systems, analyzing their strengths and limitations.

System Analysis and Design

This chapter focuses on the functional and non-functional requirements of the system, feasibility analysis, data models, and architectural design.

Implementation and Testing

This chapter details the technologies used, system implementation process, and testing methodologies applied to ensure system reliability.

Conclusion and Future Enhancements

This chapter summarizes the project findings, outcomes, and potential enhancements for future versions of the system.

CHAPTER 2:

BACKGROUND STUDY AND LITERATURE REVIEW

2.1 Background Study

The process of managing academic results has long been a crucial aspect of educational institutions, ensuring accurate assessment and record-keeping for students. However, traditional result management methods have heavily relied on manual processes, leading to inefficiencies, errors, and challenges in maintaining accurate academic records.

Historically, result management involved physical record-keeping, manual grade computation, and report generation. These processes were time-consuming, error-prone, and often resulted in inaccuracies, affecting the reliability and transparency of student assessments. Administrative staff had to manually handle large volumes of student records, leading to difficulties in data retrieval, modifications, and report generation.

With advancements in technology and the growing need for efficient academic management, automated result management systems have become essential. These systems streamline operations by digitizing student records, automating grade computation, and enabling quick access to academic performance reports. By leveraging technology, institutions can enhance transparency, accuracy, and efficiency in managing student results.

Modern result management systems aim to revolutionize how student academic records are handled. These systems integrate software applications, databases, and online platforms to simplify result entry, grade computation, and report generation. By automating these tasks, institutions can reduce administrative workload, minimize errors, and improve the overall efficiency of academic record-keeping.

The transition from manual result management to automated systems represents a significant improvement in educational administration. The inefficiencies and challenges of traditional methods have highlighted the necessity of adopting digital solutions to enhance result accuracy, data security, and administrative efficiency. The upcoming Result Management System project aims to capitalize on these advancements, providing an

efficient and user-friendly platform exclusively for administrators to manage student results effectively.

The Result Management System is designed to be used solely by administrators. The admin is responsible for creating, updating, and maintaining student academic records, ensuring that all data is accurate and properly stored. The system will support result entry, modifications, and report generation while restricting access to unauthorized users, ensuring data integrity and security.

2.2 Literature Review

Managing student results efficiently is essential for academic institutions. Many result management systems (RMS) have been developed to improve record-keeping, automate grade calculations, and simplify administrative work. This review examines two existing systems, highlighting their features, strengths, and areas that need improvement.

Online Student Result Management System

The "Online Student Result Management System" is designed to provide secure and structured access to student academic records. It helps administrators store, retrieve, and modify student information easily. One key benefit is its ability to calculate grades automatically, which reduces human errors and ensures accuracy. Additionally, it includes user authentication features to prevent unauthorized access. [3] However, it has some limitations. For example, it does not offer customizable reporting features, making it difficult for administrators to generate specialized reports. Moreover, it lacks advanced data visualization tools that could help analyze student performance trends. [2]

Automated Academic Performance Management System

Another widely used system is the "Automated Academic Performance Management System," which aims to make result processing more efficient. This system provides real-time data updates and automated report generation, making academic record management more streamlined. It stores data in a structured database and follows predefined grading rules, ensuring consistency. [1] Despite these benefits, there are some challenges. The system has limited administrative controls, which restricts modifications to grading

parameters. Additionally, exporting reports in different formats is not always possible, making it less flexible for different institutional needs. [3]

Need for an Improved Result Management System

Since administrators are the main users of RMS, an ideal system should be easy to use and provide full control over result entry, modification, and report generation. Security is also crucial to prevent unauthorized access and ensure data integrity. Many current systems lack flexibility in reporting and data visualization, which are essential for monitoring academic performance.

To address these issues, a new result management system is needed. This system should not only fix the shortcomings of existing solutions but also be aligned with the operational needs of educational institutions. A well-designed RMS will help administrators manage student results accurately, efficiently, and securely.

CHAPTER 3:

SYSTEM ANALYSIS AND DESIGN

3.1 System Analysis

This system is designed following the Waterfall software development lifecycle, emphasizing a linear and sequential development process. During the requirement analysis phase, both functional and non-functional requirements are thoroughly gathered, documented, and finalized before moving on to the next phase. System design is developed in a comprehensive and upfront manner, with detailed architectural and technical specifications established. The implementation phase follows, where the development team writes the code according to the pre-defined design. After implementation, the system enters the testing phase, where all features are tested for correctness, functionality, and quality. Once the system passes testing, it is deployed in one complete release. Maintenance follows after deployment, where the system is updated or patched based on user feedback and changing requirements, but no further development occurs until the next project phase.

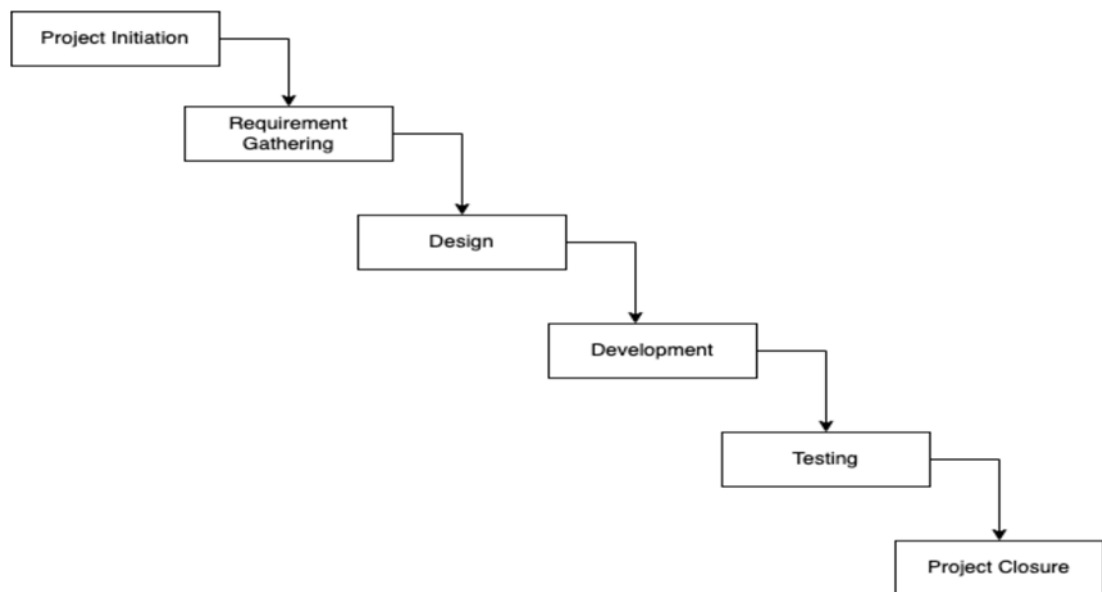


Figure 3.1: Waterfall Methodology for Result Management System

3.1.1 Requirement Analysis

To ensure the successful development of the Result Management System, functional and non-functional requirements have been identified and studied.

i. Functional Requirements:

- Admin Login: The system allows only the admin to log in securely using valid credentials.
- Student Record Management: The admin can add, update, and delete student records.
- Result Entry: The admin can input student results, including marks and grades.
- Result Modification: The system allows modification and updating of stored results.
- Report Generation: The system generates reports based on students' academic performance.
- Data Backup : The system provides backup data security.

USECASE DIAGRAM

In the system, there is a single user, the Admin, who performs various actions through different processes or use cases. The first process is Login, where the Admin accesses the system by logging in with their credentials. Once logged in, the Admin can proceed with managing student records. The Add Student Details process allows the Admin to input information for a new student, such as their name, enrollment number, and other relevant data. If needed, the Admin can also delete a student's record through the Remove Student Details process. In case any student details need to be updated, the Admin can modify the existing information through the Update Student Details process. To view student information, the Admin can use the Display Student Details process, which shows all the relevant student records.

Additionally, the Admin manages student marks. Using the Add Marks process, the Admin inputs marks for a student in a specific subject or exam. If there is a need to delete or correct any marks, the Remove Marks and Update Marks processes allow for removing or editing the marks, respectively. To view the marks of a student, the Admin can use the Display Marks process. Finally, the Admin can end the session and log out of the system through the Logout process.

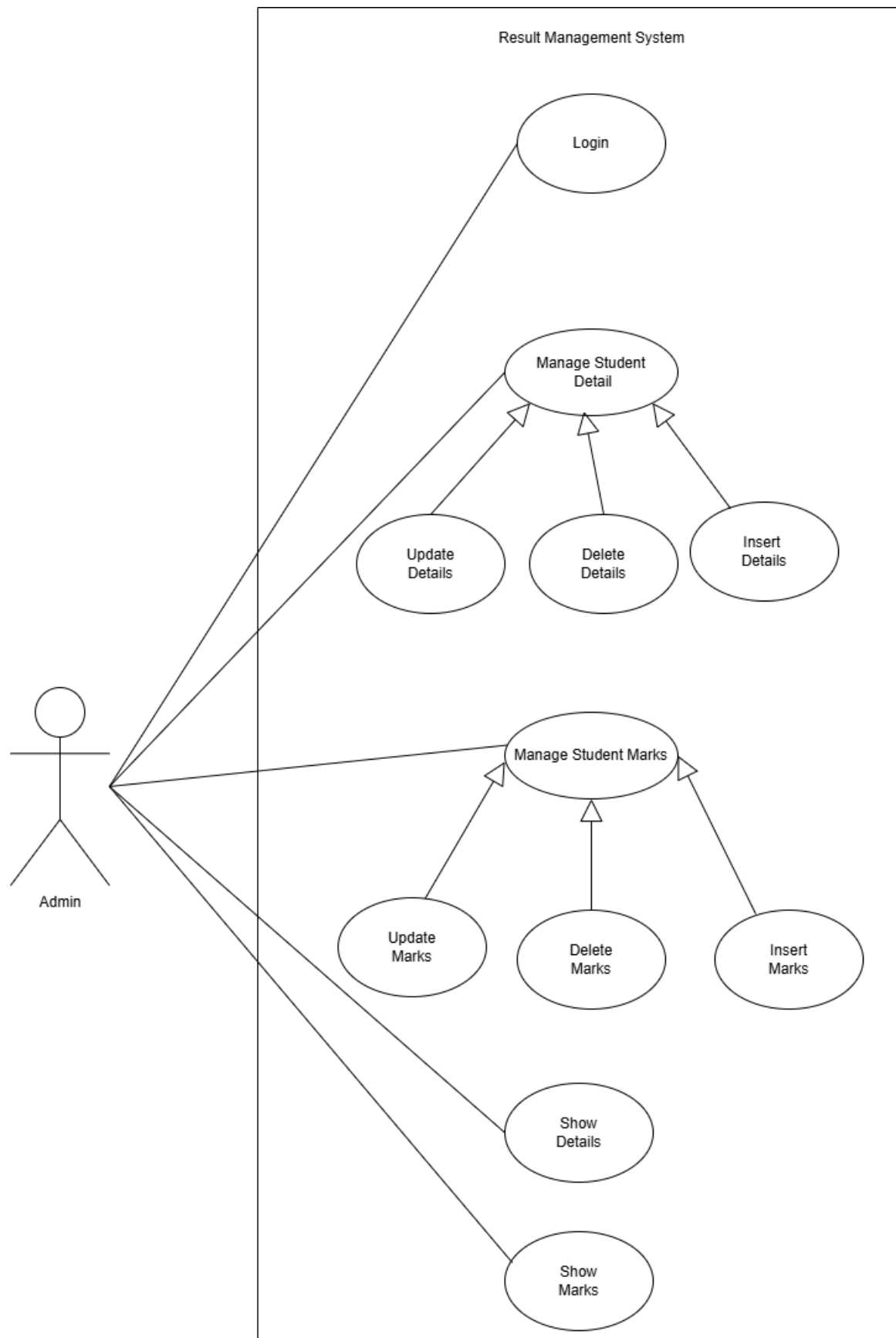


Figure 3.2 : Use Case Diagram of Result Management System

ii. Non-functional Requirements:

- Usability: The system must have an intuitive and user-friendly interface.
- Performance: It should provide fast and accurate responses when processing results.
- Reliability: The system must ensure high availability and minimal downtime.
- Security: Only authorized administrators can access and manage student records.

3.1.2 Feasibility Analysis

A feasibility study was conducted to determine the practicality of implementing the system.

a) Technical Feasibility

The system is developed using Java Swing for the front-end and MySQL (hosted on XAMPP) for the database, ensuring compatibility with widely used technologies.

b) Operational Feasibility

The Result Management System enhances the efficiency of academic record management. It simplifies administrative tasks, reduces errors, and provides easy access to student performance data.

c) Economic Feasibility

The system is cost-effective since it utilizes open-source tools such as Java and MySQL, eliminating the need for expensive software or hardware.

d) Schedule Feasibility

The system is completed within scheduled time and do not exceed the scheduled time.

Task Name	Duration
Project Initiation	2 weeks
Requirement Gathering	2 weeks
Design	2 weeks
Development	6 weeks
Testing	1 week
Project Closure	1 week
Documentation	14 weeks

Table 3.1: Gantt Chart of Result Management System

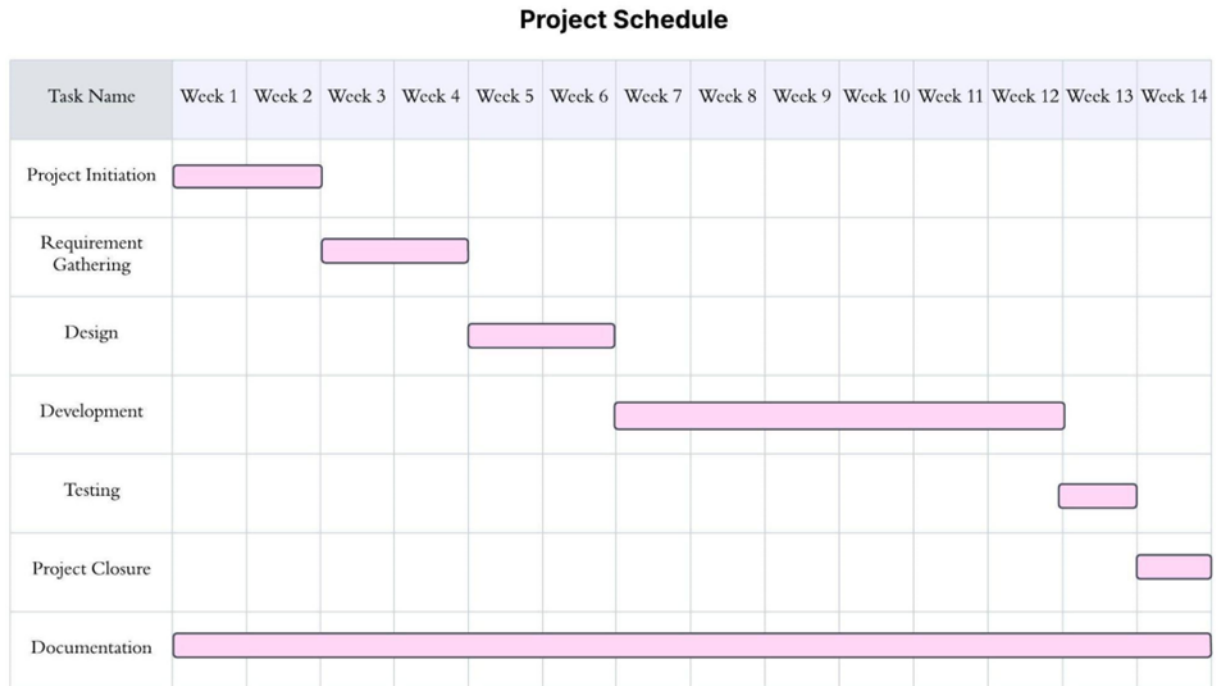


Figure 3.3 : Gantt Chart of Result Management System

3.1.3 Data Modeling (ER Diagram)

The Entity-Relationship (ER) diagram represents the core entities in the Result Management System:

- Admin (Attributes: ID, Username, Password)
- Students (Attributes: Student_ID, Name, Course, Address, Contact, Semester)
- Marks (Attributes: Student_ID, Subject_ID, Marks, Semester)
- Subjects (Attributes: Subject_ID, Subject_Name, Semester, Subject_Type)

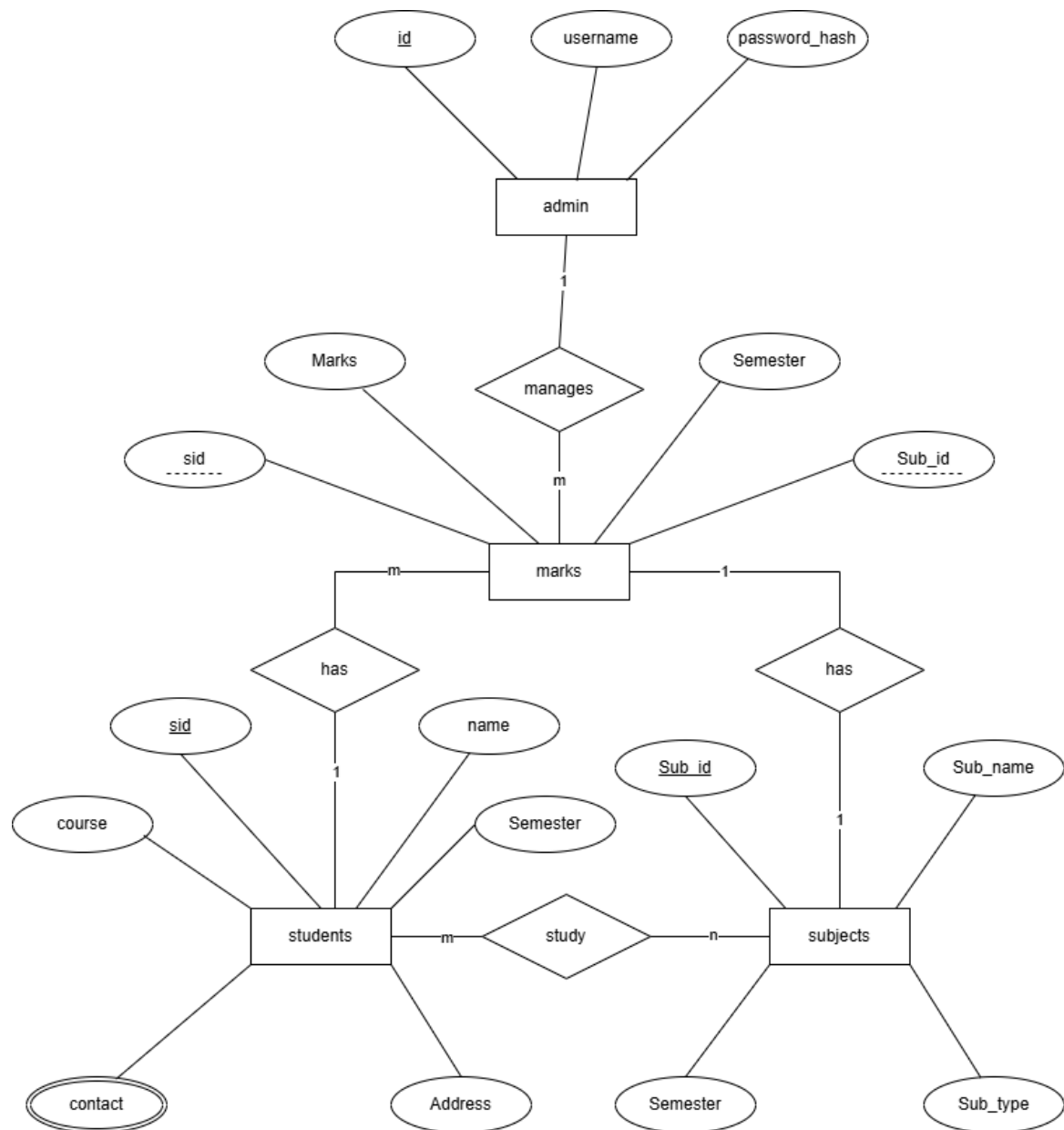


Figure 3.4 : E-R Diagram of Result Management System

3.1.4 Process Modeling (DFD)

Level 0 DFD (Context Diagram)

Inputs: Admin login request, student record updates, result entries.

Outputs: Student performance reports, updated result data.

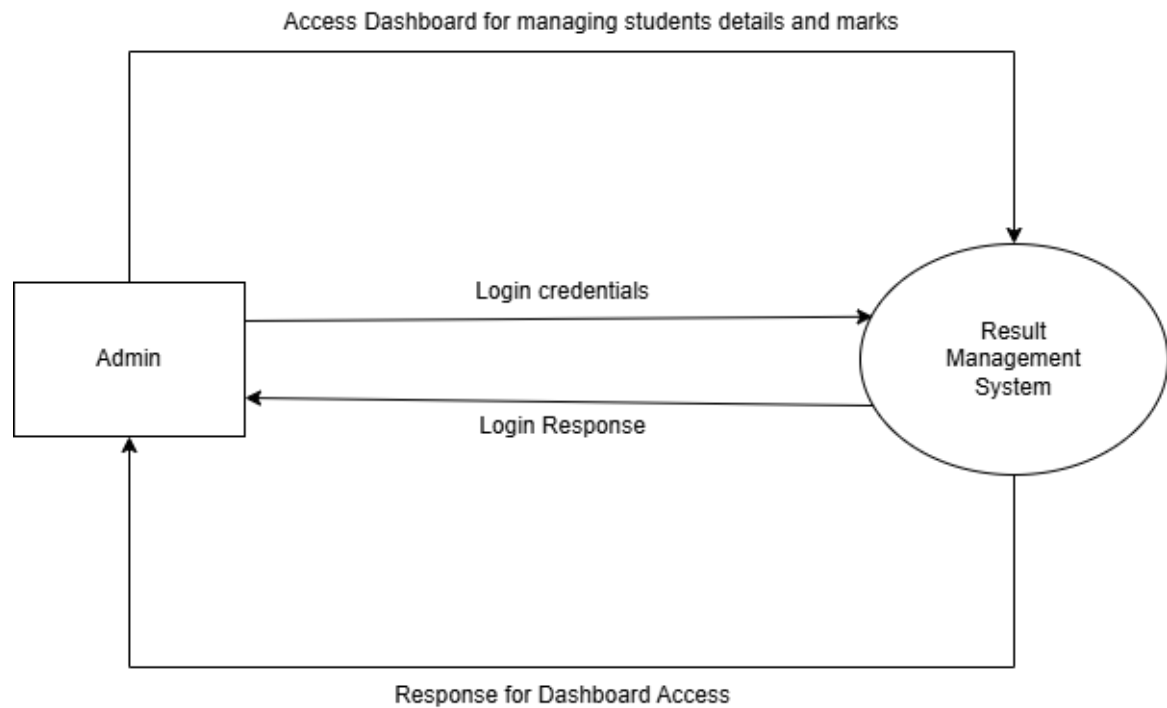


Figure 3.5 : Level 0 DFD of Result Management

Level 1 DFD

The system consists of processes such as:

- Login authentication.
- Managing student records.
- Managing result entries.
- Generating result.

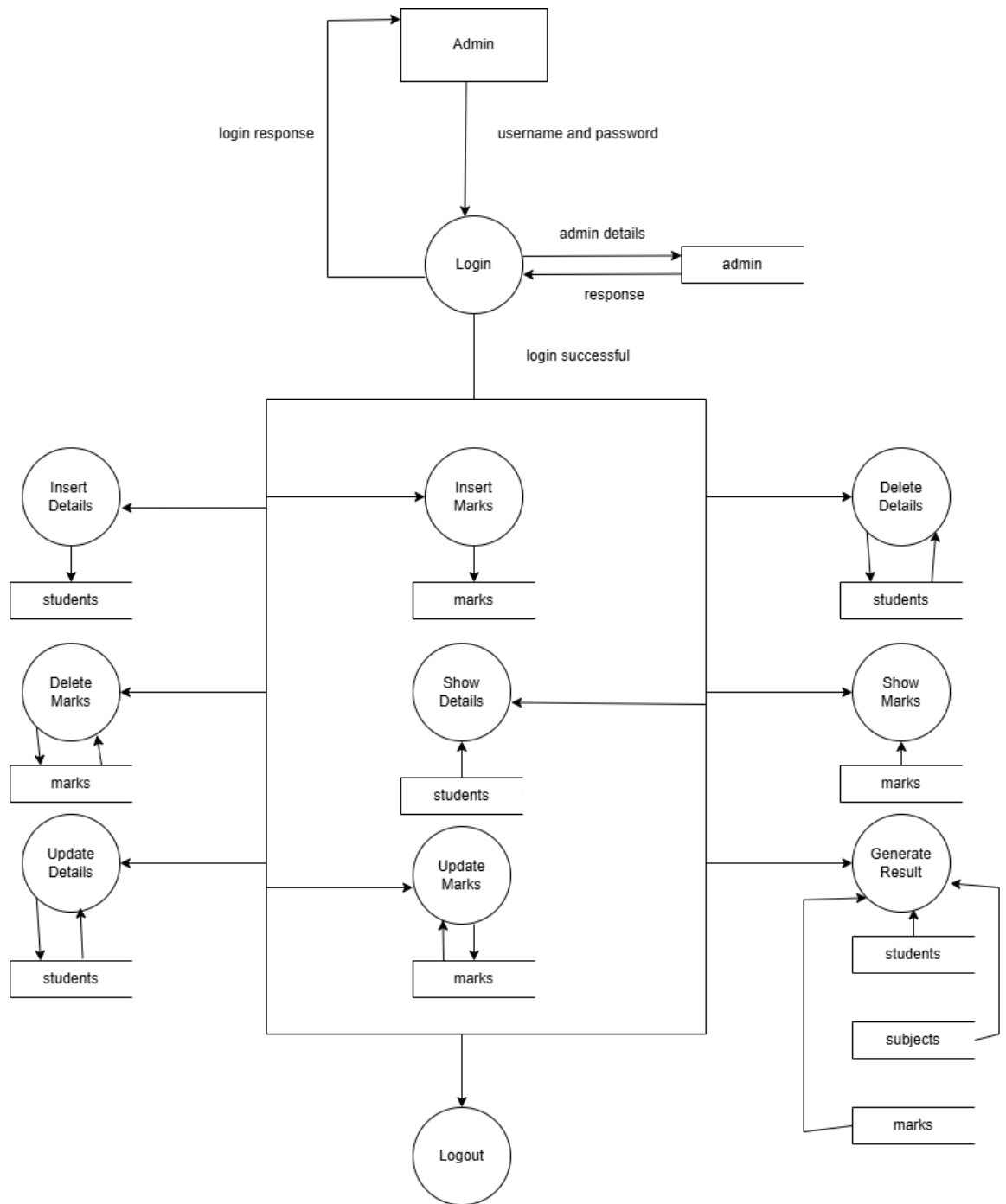


Figure 3.6 : Level 1 DFD of Result Management System

3.2 System Design

The system design phase ensures that all functional and non-functional requirements are met effectively.

3.2.1 Architectural Design

The system follows a two-tier architecture, consisting of:

- User Interface: Java Swing-based desktop application.
- Database Server: MySQL hosted on XAMPP.

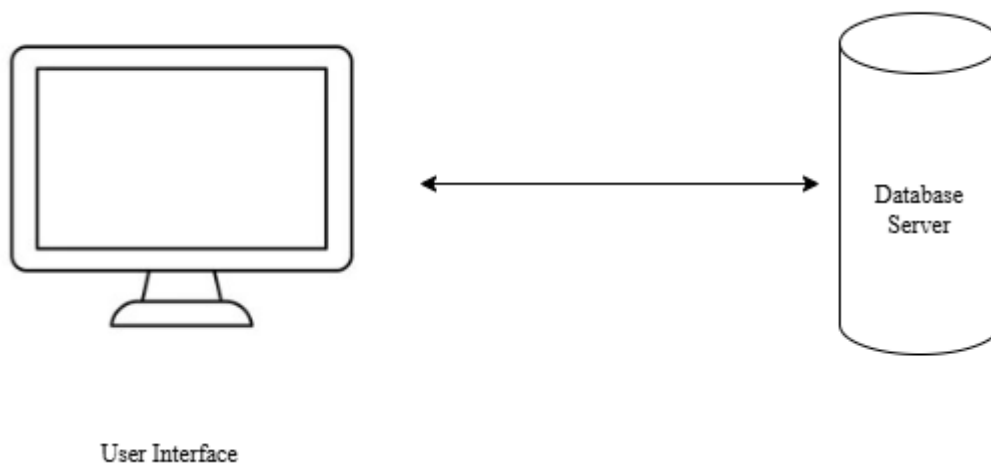


Figure 3.7 : Architectural Design of Result Management System

3.2.2 System Flowchart

The flowchart outlines the workflow of the Result Management System:

- The admin logs in to the system.
- If authentication is successful, the admin accesses the dashboard.
- The admin manages student records (add/update/delete).
- The admin enters or updates results for students.
- The system generates reports based on student performance.
- The admin logs out after completing tasks.

System Flowchart of Result Management System

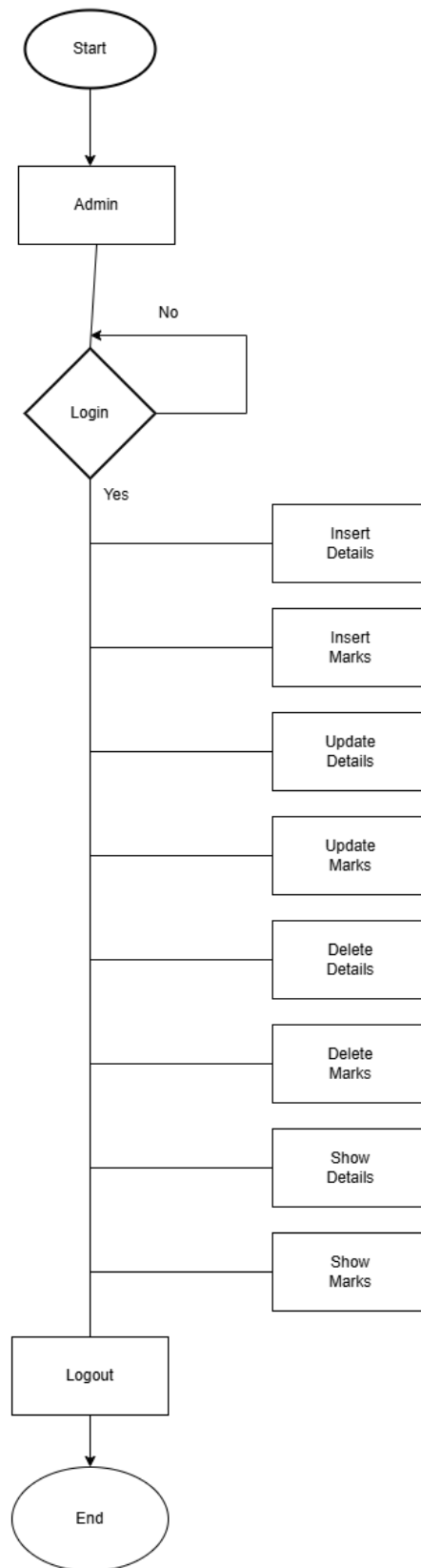


Figure 3.8 : System Flow Chart of Result Management System

3.2.3 Database Schema Design

The figure below is the database schema design of the Result Management System.

Database schema design is used to show the basic structure of the system. In the Result Management System, there are four tables in the database, each of them having its own fields where their ID is the Primary Key, and if that ID is used in another table, it becomes a Foreign Key. Foreign keys are connected to another table with a line. Each entity has a data type, and the foreign key in the schema is represented by an arrow, as shown in the diagram.

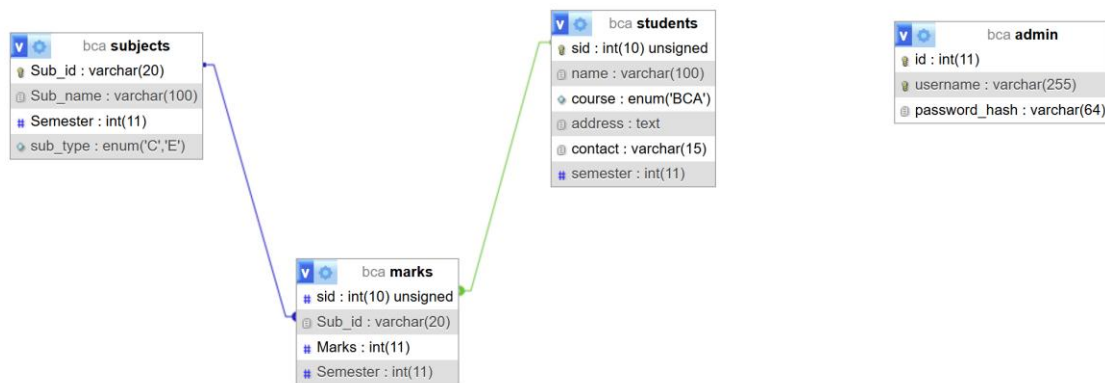


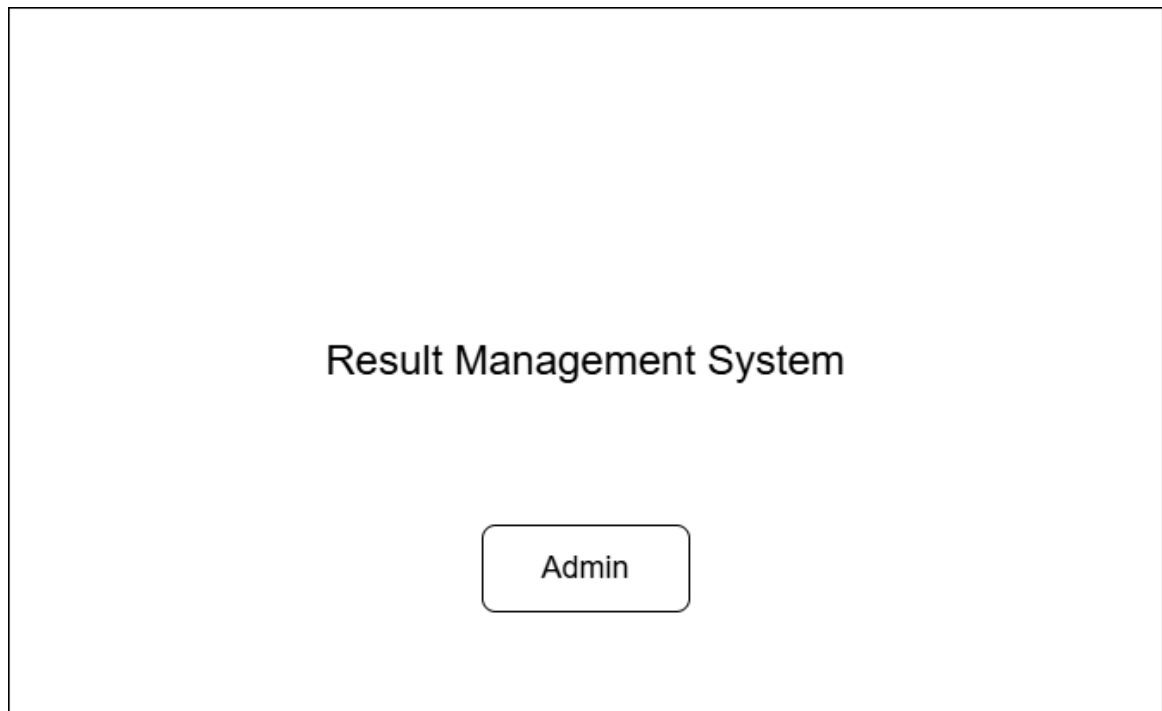
Figure 3.9 : Database Schema Design of Result Management System

3.2.4 Interface Design (UI Interface)

The interface design consists of:

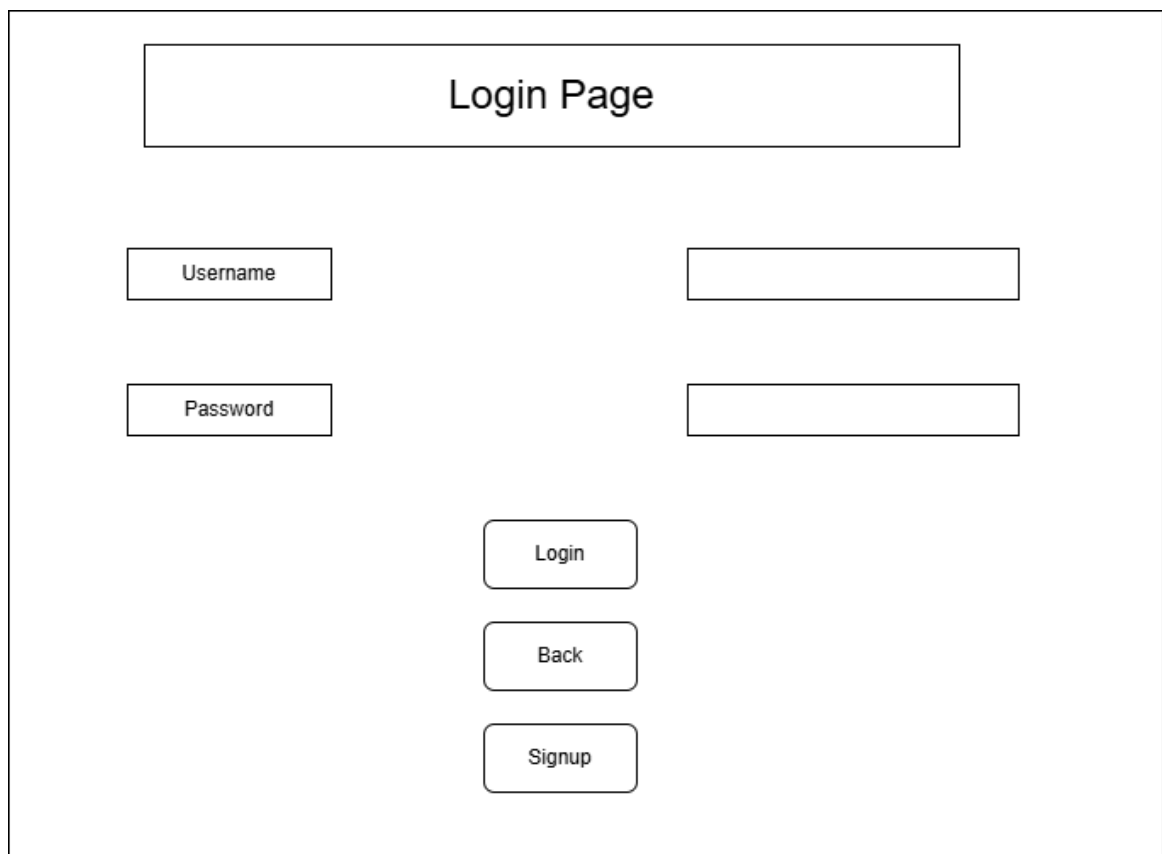
- Home Page: First page of the system.
- Login Page: Secure authentication for admin.
- Dashboard: Displays system features such as student record management and result entry.

This structured system ensures efficient and error-free management of academic results, improving administrative workflow and data accuracy.



The image shows a simple web page layout. At the top center, the text "Result Management System" is displayed in a large, bold, black font. Below this text, centered on the page, is a single rounded rectangular button with the word "Admin" written inside it in a black font.

Figure 3.10 : Home Page of Result Management System



The image shows a login page layout. At the top center, there is a rectangular box containing the text "Login Page". Below this box, there are two rows of input fields. The first row has a label "Username" in a small rectangular box followed by a larger empty rectangular input field. The second row has a label "Password" in a small rectangular box followed by a larger empty rectangular input field. Below these input fields, there are three rounded rectangular buttons stacked vertically in the center. The buttons are labeled "Login", "Back", and "Signup" from top to bottom, all in a black font.

Figure 3.11 : Login Page of Result Management System

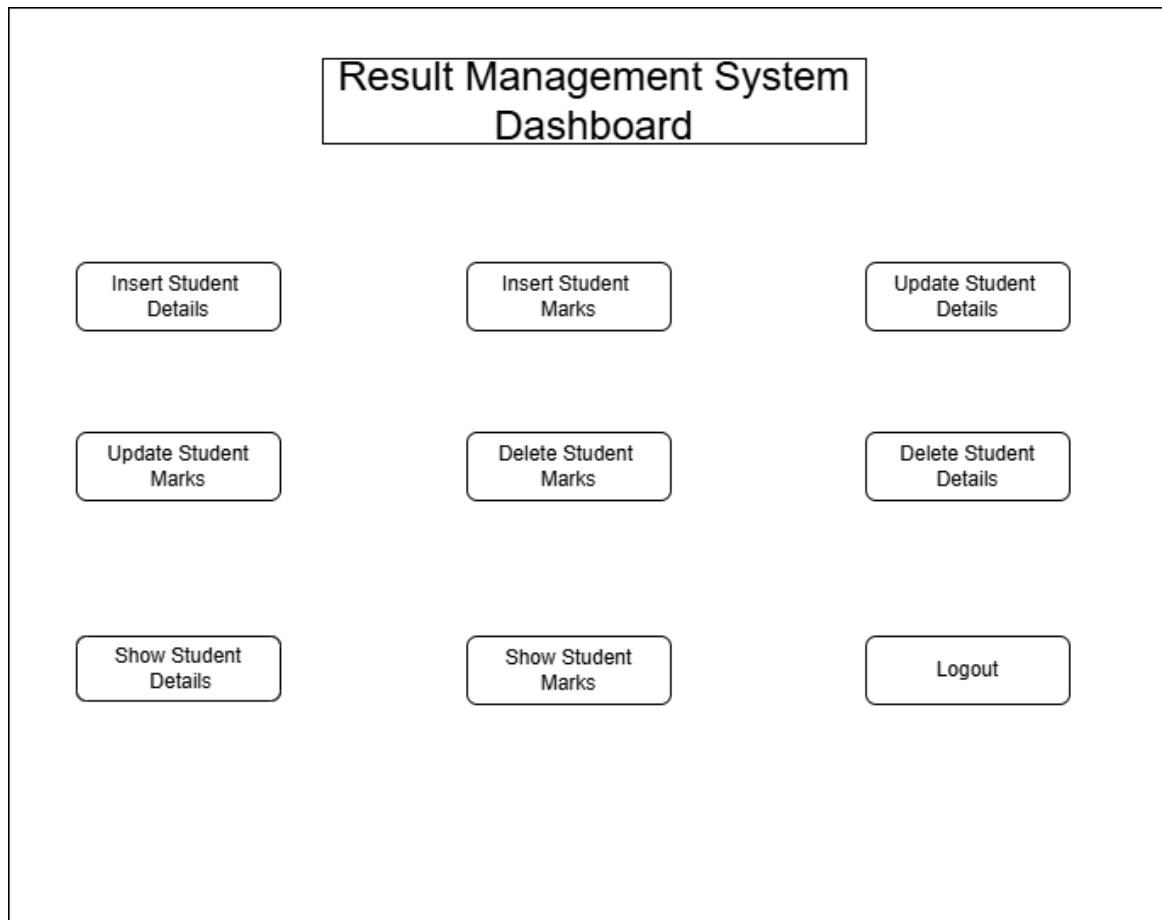


Figure 3.12 : Dashboard of Result Management System

3.2.5 Physical DFD

Here, admin is the user who registers and login to the system, if it is successful then admin' information is stored in the admin database and then admin can perform tasks such as insert details, insert marks, update details, update marks, delete details, delete marks, show marks, show details and generate result. All the marks and details of students are stored in the marks and students database respectively. Similarly subjects database contains the information of each subject.

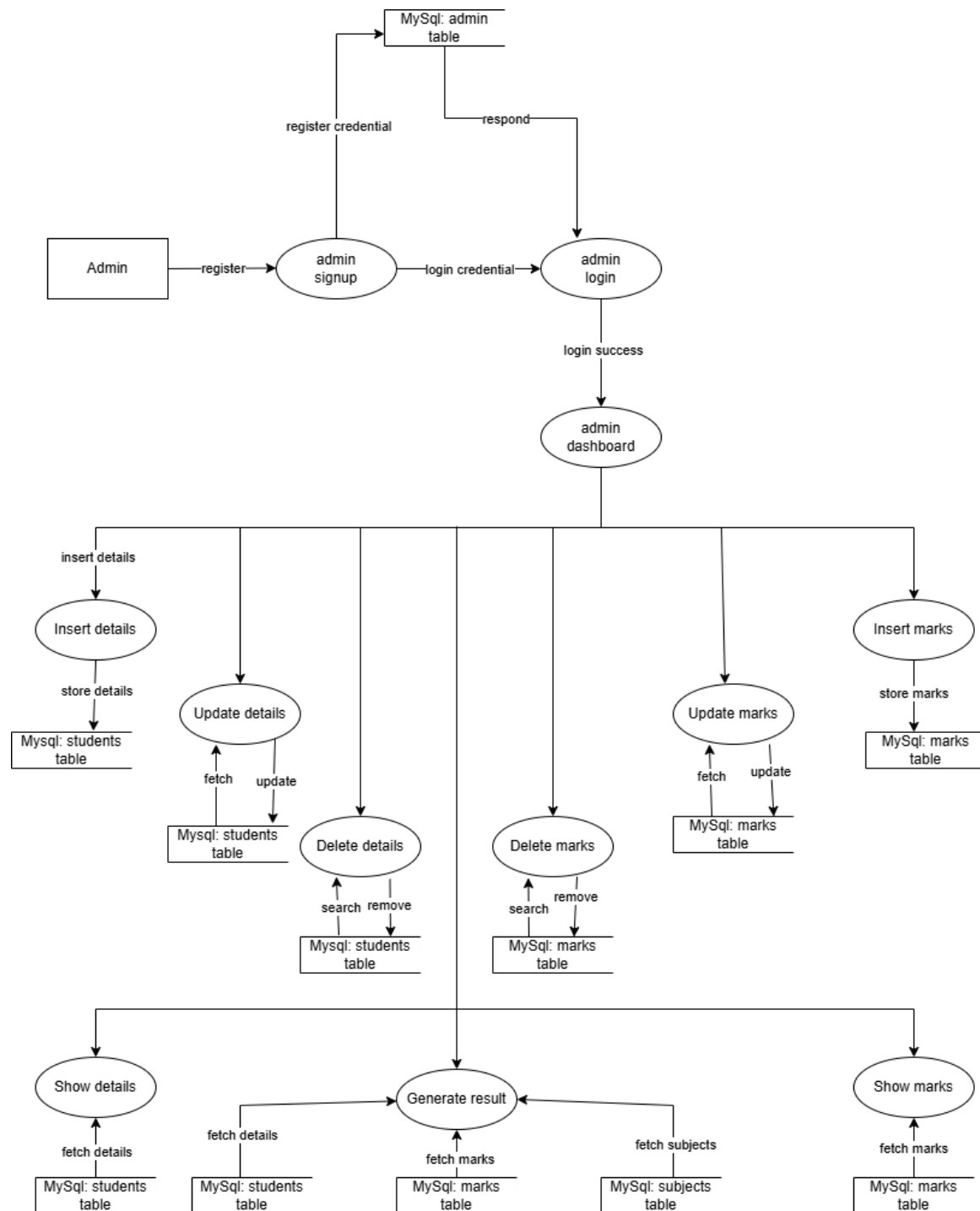


Figure 3.13 : Physical DFD of Result Management System

CHAPTER 4:

IMPLEMENTATION AND TESTING

4.1 Implementation

4.1.1 Tools Used

Following are the tools and frameworks used for the development of the Result Management System:

Front-End Tools

Java Swing

Java Swing is used for designing the graphical user interface (GUI) of the Result Management System. It helps in creating forms, buttons, tables, and input fields for user interaction.

Back-End Tools

Java (JDK)

Java is used as the primary programming language for handling logic and data operations in the Result Management System.

It manages CRUD (Create, Read, Update, Delete) operations related to students, subjects, and marks.

Database Tools

MySQL

MySQL is used as the database management system to store student details, subjects, marks, and generated reports.

It enables efficient storage, retrieval, and management of academic records.

XAMPP

XAMPP is used as a local server environment to manage MySQL database services during development and testing.

Development Environment

NetBeans IDE

NetBeans is used for writing, debugging, and executing Java Swing-based applications. It simplifies GUI development and database connectivity using Java Database Connectivity (JDBC).

Documentation Tools

MS Office

This is used for writing and editing the documentation of sponsorship management system.

Draw.io

This is used to generate diagrams for system analysis and design of sponsorship management system. Diagrams were created using this tool in order to save time since all components are available with drag and drop functions.

Version Control:

- **Git:** Used for version control to manage code changes and collaborate efficiently.
- **GitHub:** Used to store the code repository and manage branches for different development tasks.

4.1.2 Implementation Details of Modules

The Result Management System consists of several modules that help manage student records and results. Below are the details of each module:

1. Student Management Module

- Allows the user to add, update, delete, and view student records.
- Stores details such as Student ID, Name, Course, Contact, Semester in the database.

2. Marks Management Module

- Allows entry and modification of student marks for each subject.
- Stores Student ID, Subject ID, Marks, and Semester in the database.

3. Result Generation Module

- Fetches student marks from the database and generates a result report.
- Calculates total marks, percentage, and pass/fail status based on predefined criteria.
- Displays results in a tabular format and allows printing or exporting reports.

4. Login Module

- Provides authentication to the user through username and password validation.
- Ensures only authorized users can access the system.

4.2 Testing

The system has undergone rigorous testing to ensure its accuracy and reliability.

4.2.1 Test Cases for Unit Testing

Each module was tested individually to validate its correctness. Below are the test cases:

Admin Login

Table 4.1 : Test Case for User Login of Result Management System

S.No	Test Name	Input	Expected Output	Actual Output	Test Result
1.	Open Application	Run the project from Netbeans (Click “Run”)	RMS Home Page	RMS Home Page	Pass
2.	Enter Username and Invalid Password	Username: rohit@gmail.com Password: 33075	Login Failed and invalid username or password is shown	Login Failed	Pass
3.	Enter Valid Username and Password and click login button	Username: rohit@gmail.com Password: 12345	Login Successful and redirect to dashboard	Redirect to dashboard	Pass

User Registration

Table 4.2 : Test Case for User Registration of Result Management System

S.No	Test Name	Input	Expected Output	Actual Output	Test Result
1.	Open Application	Run project from Netbeans (Click “Run” on AdminRegister.java)	RMS Register Page	RMS Register Page	Pass
2.	Enter Invalid Username and Password and click submit button	Username: Rohit123 Password: Rohit123	Username as Valid Email format	Registration Failed	Pass
3.	Enter Valid Username and Password and click submit button	Username: rohit@gmail.com Password: 12345	Registration Successful	Registration Successful	Pass

4.2.2 Test Cases for System Testing

In system testing, whole system is tested as below:

Test Case for Student Registration Successful

Table 4.3 : Test Case for Student Registration Success of Result Management System

Test Case 1	Successful Student Registration
Test Data	SID = 1 Name = Rohit Yadav Course = BCA Address = Lahan Contact = 9814795068 Semester = 1
Expected Result	A message should be displayed saying “Details Saved Successfully”
Test Result	Student Registration Success

Test Case for Student Registration Failure of Result Management System

Table 4.4 : Test Case of Student Registration Failure of RMS

Test Case 1	Student Registration Failure
Test Data	SID = 1 Name = Rohit Yadav Course = BCA Address = Contact = 9814795068 Semester = 1
Expected Result	A message should be displayed saying “All fields must be filled”
Test Result	A message “All fields must be filled” is displayed

Test Case for Successful Marks Entry

Table 4.5 : Test Case for Successful Marks Entry of RMS

Test Case 2	Successful Marks Entry
Test Data	SID = 1 Computer Fundamental & Applications = 60 Society and Technology = 68 English-I = 78 Mathematics-I = 98 Digital Logic = 56
Expected Result	A message should be displayed saying “Marks Saved Successfully”
Test Result	A message “Marks Saved Successfully” is displayed

Test Case for Marks Entry Failure

Table 4.6 : Test Case for Marks Entry Failure

Test Case 2	Marks Entry Failure
Test Data	SID = 1 Computer Fundamental & Applications = 60 Society and Technology = English-I = 78 Mathematics-I = 98 Digital Logic = 56
Expected Result	A message should be displayed saying “Please fill all fields”
Test Result	A message “Please fill all fields” is displayed

Test Case for Successful Student Details Deletion

Table 4.7 : Test Case for Successful Student Details Deletion

Test Case 3	Successful Details Deletion
Test Data	SID = 1 Semester = 1
Expected Result	A message should be displayed saying “Record Deleted Successfully”
Test Result	A message “Please fill all fields” is displayed

Test Case for Student Details Deletion Failure

Table 4.8 : Test Case for Student Details Deletion Failure

Test Case 3	Details Deletion Failure
Test Data	SID = Semester =
Expected Result	A message should be displayed saying “Enter valid SID and select Semester”
Test Result	A message “Enter valid SID and select Semester” is displayed

Test Case for Successful Student Marks Deletion

Table 4.9 : Test Case for Successful Student Marks Deletion

Test Case 4	Successful Details Deletion
Test Data	SID = 1 Semester = 1
Expected Result	A message should be displayed saying “Marks Deleted Successfully”
Test Result	A message “Marks Deleted Successfully” is displayed

Test Case for Student Marks Deletion Failure

Table 4.10 : Test Case for Student Marks Deletion Failure

Test Case 4	Marks Deletion Failure
Test Data	SID = Semester =
Expected Result	A message should be displayed saying “Enter valid SID and select Semester”
Test Result	A message “Enter valid SID and select Semester” is displayed

Test Case for Successful Student Details Update

Table 4.11 : Test Case for Successful Student Details Update

Test Case 5	Successful Details Update
Test Data	SID = 1 Name = Rohit Yadav (Yadav added as update) Course = BCA Address = Lahan Contact = 9814795068 Semester = 1
Expected Result	A message should be displayed saying “Record Successfully Updated”
Test Result	A message “Record Successfully Updated” is displayed

Test Case Student Details Update Failure

Table 4.12 : Test Case for Student Details Update Failure

Test Case 5	Details Update Failure
Test Data	SID = 1 Name = Course = BCA Address = Lahan Contact = 9814795068 Semester = 1
Expected Result	A message should be displayed saying “All fields must be filled”
Test Result	A message “All fields must be filled” is displayed

Test Case Successful Student Marks Update

Table 4.13 : Test Case for Successful Student Marks Update

Test Case 6	Successful Marks Update			
Test Data	SID	Sub_id	Marks	Semester
	1	1	45	1
	1	2	67 (Updated)	1
	1	3	56	1
	1	4	70	1
	1	5	67	1
Expected Result	A message should be displayed saying “Marks Successfully Updated”			
Test Result	A message “Marks Successfully Updated” is displayed			

Test Case Student Marks Update Failure

Table 4.14 : Test Case for Student Marks Update Failure

Test Case 6	Marks Update Failure			
Test Data	SID	Sub_id	Marks	Semester
	1	1	45	1
	1	2		1
	1	3	56	1
	1	4	70	1
	1	5	67	1
Expected Result	A message should be displayed saying “Marks field is empty”			
Test Result	A message “Marks field is empty” is displayed			

CHAPTER 5:

CONCLUSION AND FUTURE RECOMMENDATIONS

5.1. Lessons Learned / Outcome

Every project provides an opportunity to learn and gain knowledge in different aspects. During the development of the Result Management System (RMS), we gained hands-on experience with various technologies and tools essential for software development.

Technical Skills Acquired

- **Java Swing** – Developed the graphical user interface (GUI) for the RMS application, implementing various UI components and event handling mechanisms.
- **XAMPP** – Used XAMPP to configure and manage the Apache server, PHP, and MySQL for local database connectivity and testing.
- **MySQL** – Designed and optimized database schemas, executed SQL queries, and managed data effectively to ensure efficient system performance.
- **Draw.io** – Created Entity-Relationship (ER) diagrams, schema diagrams, and Data Flow Diagrams (DFD) for system design and documentation.
- **Microsoft Word** – Prepared project documentation, including proposals, reports, and system manuals.

Through this project, we strengthened our ability to integrate multiple technologies to build a functional and efficient Result Management System.

5.2. Conclusion

The Result Management System (RMS) has been successfully developed with predefined objectives. The system allows an admin user to efficiently manage student records, subject details, and marks while generating reports. This system is designed to handle CRUD operations (Create, Read, Update, Delete) and provides an intuitive Graphical User Interface (GUI) using Java Swing.

Key functionalities of this system include:

- Student & Subject Management: Admin can add, update, and delete student and subject records.
- Marks Entry & Modification: Admin can enter and modify marks for students.
- Result Generation: The system automatically calculates total marks, percentage, and pass/fail status for students.
- Secure Authentication: The system ensures secure login for the admin user.

Overall, the Result Management System enhances efficiency in handling student academic records and ensures accurate and timely result generation.

5.3. Future Recommendations

Although the Result Management System meets its current objectives, it can be improved by adding additional features in the future. Some recommended enhancements include:

- Multi-User Access
- Implement separate roles for students, teachers, and admins to allow students to view their results online.
- Automated Report Generation
- Add a PDF/Excel export feature for generating and printing result reports.
- Graphical Performance Analysis
- Include graphs and charts to visualize student performance trends.
- Email Notifications
- Enable automated email notifications to inform students about their results.
- Cloud Database Integration
- Upgrade from a local MySQL database to a cloud-based system for remote access.

These enhancements will further improve the usability and functionality of the system, making it more scalable and user-friendly.

REFERENCES

- [1] J. Doe and M. Lee, "Enhancing Academic Record Management," *Journal of Educational Technology*, vol. 45, no. 3, pp. 122-135, 2021.
- [2] P. Johnson, R. Smith, and K. Taylor, "Automating Student Performance Tracking," *Academic Computing Review*, vol. 36, no. 2, pp. 78-92, 2019.
- [3] A. Smith and T. Brown, "Secure Online Student Records: Challenges and Solutions," *International Journal of Information Systems*, vol. 27, no. 1, pp. 45-60, 2020.
- [4] L. Williams and H. Clark, "Data Analytics in Education: Improving Result Management Systems," *Educational Data Review*, vol. 40, no. 4, pp. 89-104, 2022.
- [5] MySQL Reference Manual. Available: <https://dev.mysql.com/doc/>. [Accessed: Mar. 9, 2025].
- [6] NetBeans IDE Guide. Available: <https://netbeans.apache.org/kb/docs/>. [Accessed: Mar. 9, 2025].
- [7] I. Sommerville, *Software Engineering*, 9th ed. Boston, MA, USA: Addison-Wesley, 2010.
- [8] A. Silberschatz, H. Korth, and S. Sudarshan, *Database System Concepts*, 6th ed. New York, NY, USA: McGraw Hill, 2010.
- [9] J. A. Hoffer, *Modern System Analysis and Design*. India: McGraw Hill, 2010.
- [10] K. Arnold, J. Gosling, and D. Holmes, *The Java Programming Language*, 4th ed. Boston, MA: Addison-Wesley, 2005.

APPENDIX: SYSTEM SCREENSHOTS

➤ Database Overview

▪ Admin Database

The screenshot shows the phpMyAdmin interface for the 'bca' database, specifically the 'admin' table. The 'Table structure' tab is active, displaying the following table structure:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	username	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
3	password_hash	varchar(64)	utf8mb4_general_ci		No	None			Change Drop More

Below the table structure, there are options to 'Check all', 'With selected', 'Browse', 'Change', 'Drop', 'Primary', 'Unique', 'Index', 'Spatial', and 'Fulltext'. There is also a 'Print' button and a 'Propose table structure' button. A 'Move columns' button is also present. A 'Normalize' button is also present. A 'Add' button is present with a dropdown menu showing '1' and 'column(s)' and a 'Go' button. Below this, there is an 'Indexes' section with a table showing the indexes for the 'admin' table:

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Rename Drop	PRIMARY	BTREE	Yes	No	id	0	A	No	
Edit Rename Drop	username	BTREE	Yes	No	username	0	A	No	

At the bottom, there is a 'Create an index on' section with a dropdown menu showing '1' and 'columns' and a 'Go' button. There is also a 'Partitions' button.

▪ Marks Database

The screenshot shows the phpMyAdmin interface for the 'bca' database, specifically the 'marks' table. The 'Table structure' tab is active, displaying the following table structure:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	sid	int(10)		UNSIGNED	No	None			Change Drop More
2	Sub_id	varchar(20)	utf8mb4_general_ci		Yes	NULL			Change Drop More
3	Marks	int(11)			Yes	NULL			Change Drop More
4	Semester	int(11)			Yes	NULL			Change Drop More

Below the table structure, there are options to 'Check all', 'With selected', 'Browse', 'Change', 'Drop', 'Primary', 'Unique', 'Index', 'Spatial', and 'Fulltext'. There is also a 'Print' button and a 'Propose table structure' button. A 'Move columns' button is also present. A 'Normalize' button is also present. A 'Add' button is present with a dropdown menu showing '1' and 'column(s)' and a 'Go' button. Below this, there is an 'Indexes' section with a table showing the indexes for the 'marks' table:

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Rename Drop	fk_subid	BTREE	No	No	Sub_id	5	A	Yes	

At the bottom, there is a 'Create an index on' section with a dropdown menu showing '1' and 'columns' and a 'Go' button. There is also a 'Partitions' button.

▪ Students Database

The screenshot shows the phpMyAdmin interface for the 'students' table. The table structure is as follows:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	sid	int(10)		UNSIGNED	No	None			Change Drop More
2	name	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
3	course	enum('BCA')	utf8mb4_general_ci		No	BCA			Change Drop More
4	address	text	utf8mb4_general_ci		No	None			Change Drop More
5	contact	varchar(15)	utf8mb4_general_ci		No	None			Change Drop More
6	semester	int(11)			No	None			Change Drop More

Below the table structure, there is an 'Indexes' section showing a primary index on the 'sid' column:

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Rename Drop	PRIMARY	BTREE	Yes	No	sid	1	A	No	

At the bottom, there is a 'Create an index on' section with a dropdown for '1' column(s) and a 'Go' button.

▪ Subjects Database

The screenshot shows the phpMyAdmin interface for the 'subjects' table. The table structure is as follows:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	Sub_id	varchar(20)	utf8mb4_general_ci		No	None			Change Drop More
2	Sub_name	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
3	Semester	int(11)			No	None			Change Drop More
4	sub_type	enum('C','E')	utf8mb4_general_ci		No	None			Change Drop More

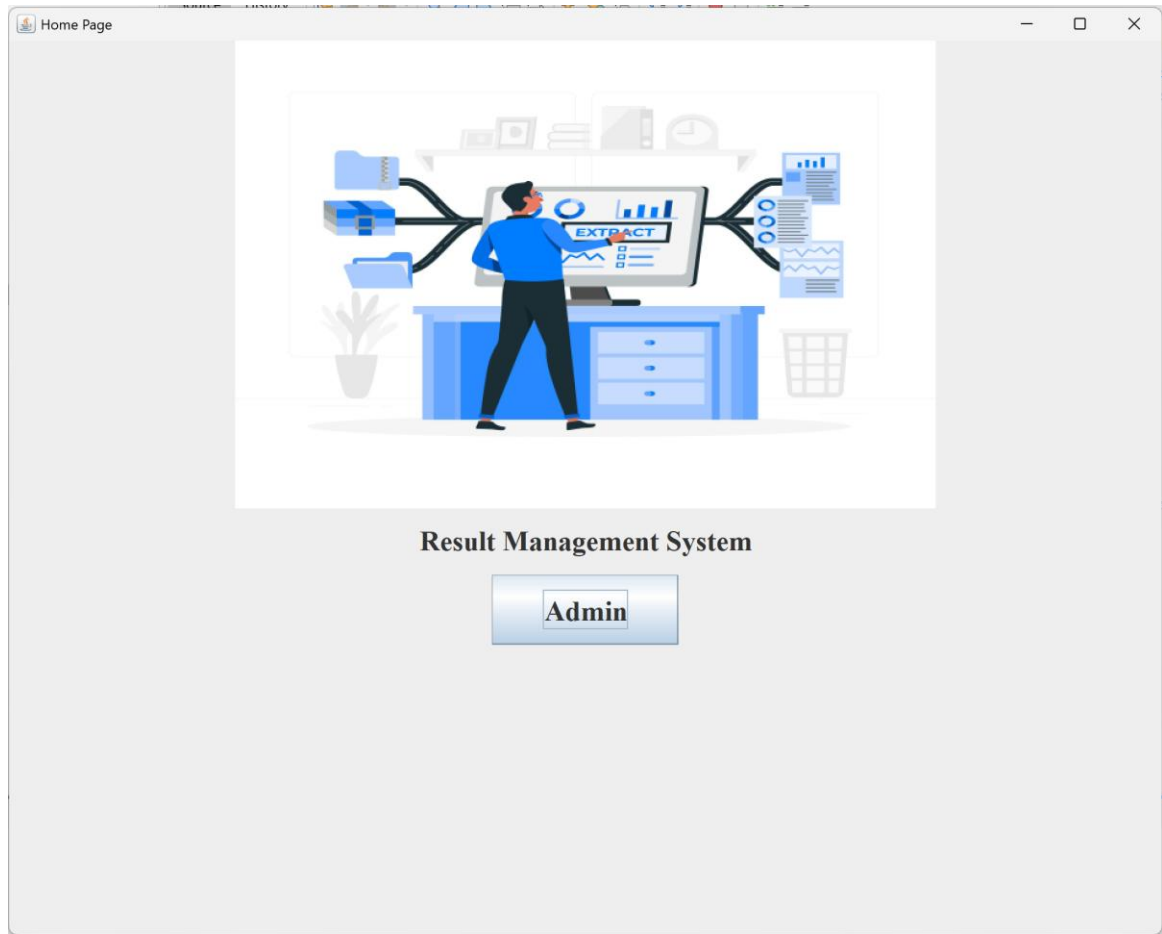
Below the table structure, there is an 'Indexes' section showing a primary index on the 'Sub_id' column:

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Rename Drop	PRIMARY	BTREE	Yes	No	Sub_id	51	A	No	

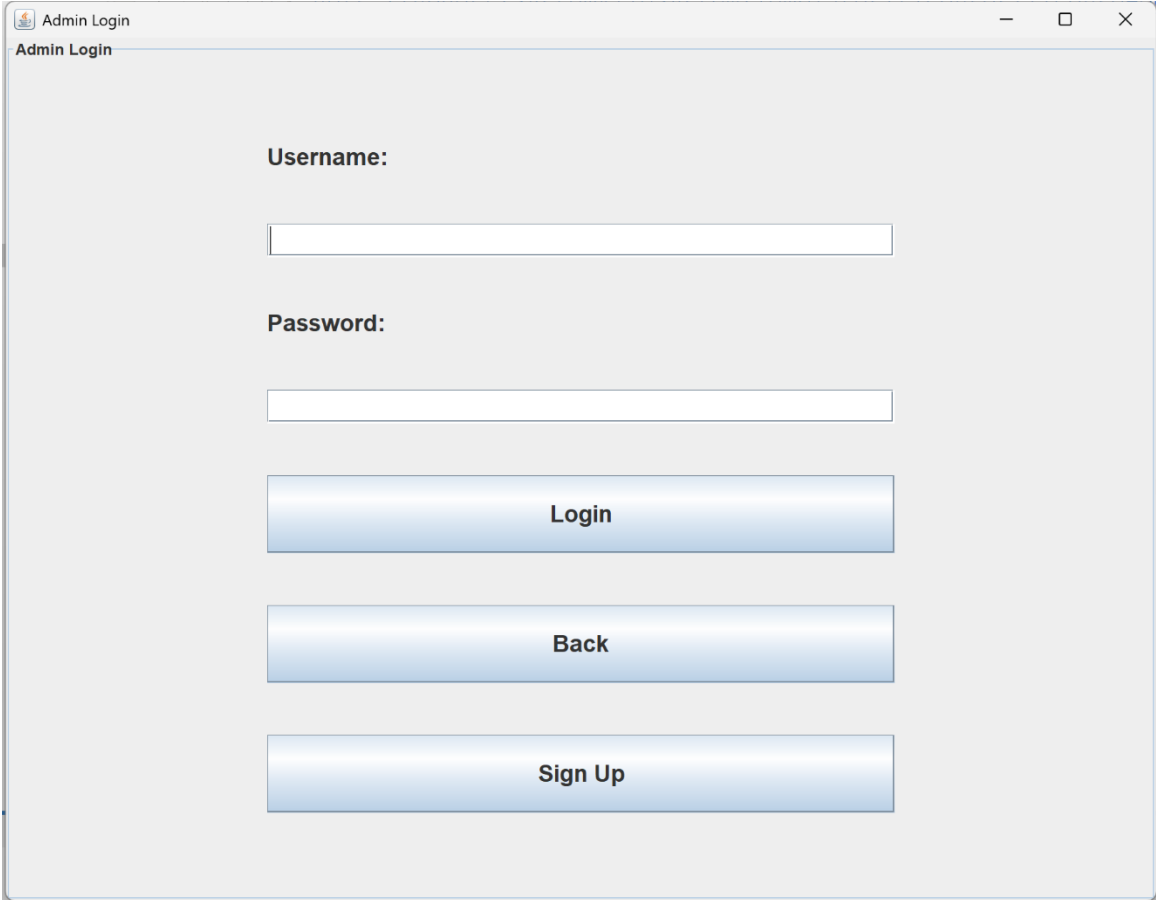
At the bottom, there is a 'Create an index on' section with a dropdown for '1' column(s) and a 'Go' button. Below this, there is a 'Partitions' section.

➤ System Overview

▪ Home Page

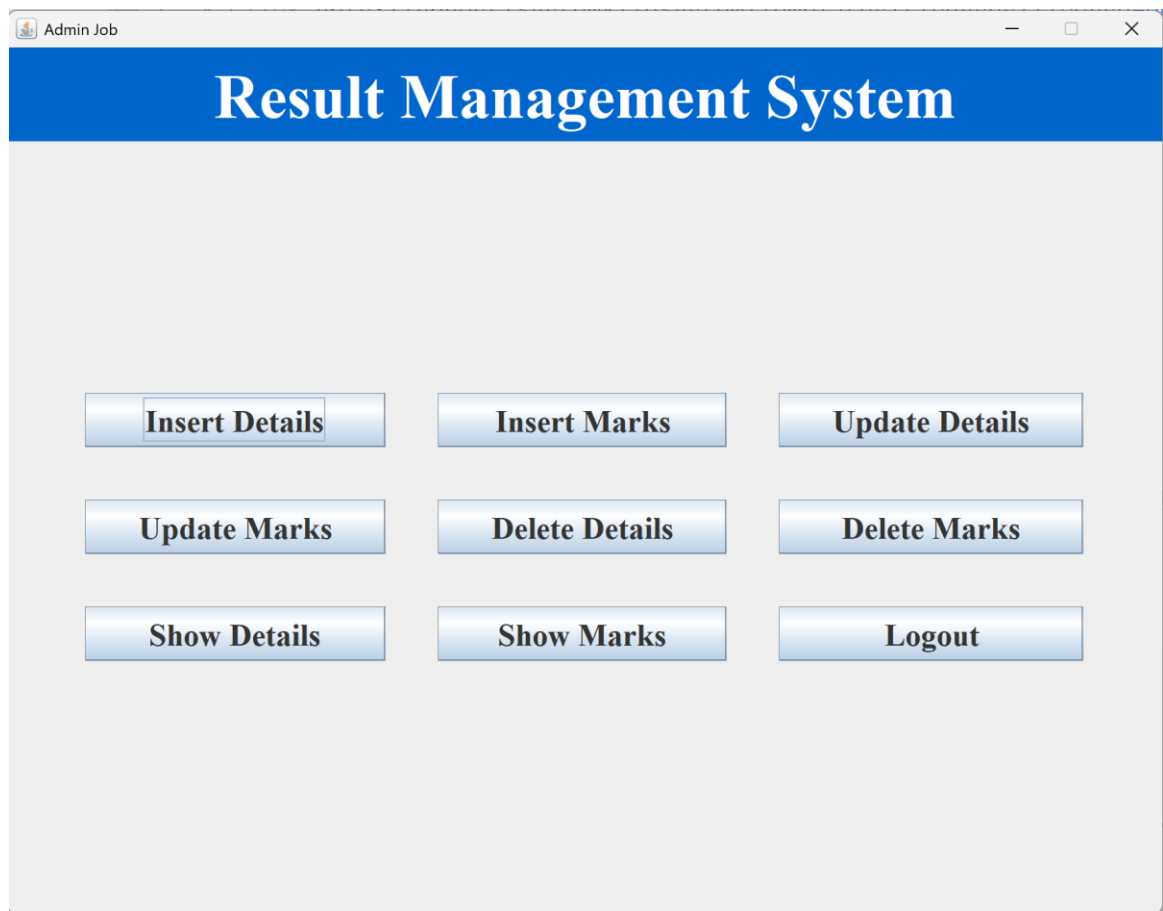


▪ Login Page

A screenshot of a web application window titled "Admin Login". The window has a light gray background and a standard Windows-style title bar with minimize, maximize, and close buttons. The main content area contains a login form with the following elements:

- A label "Username:" followed by a white text input field.
- A label "Password:" followed by a white text input field.
- A blue button with a gradient and the text "Login".
- A blue button with a gradient and the text "Back".
- A blue button with a gradient and the text "Sign Up".

- **Dashboard**



▪ Insert Details

Insert Details

SID:

Name:

Course:

BCA

Address:

Contact:

Semester:

1

Back

Save

▪ Insert Marts

First Semester Marks Entry

SID:

Computer Fundamentals & Applications:

Society and Technology:

English-I:


Mathematics-I:

Digital Logic:

Save

Back

▪ Update Details

 Update Student Record

Student ID:

Search

Name:

Course:

Address:


Contact:

Semester:

Update

Back

▪ Update Marks

 Update Marks

Select Semester:

Select Semester

Enter Student ID (SID):

Search

Update

Back

▪ Delete Marks


Delete Marks Record

Enter Student ID (SID):

Enter Semester (optional):

SID	Sub_ID	Semester
1	1	1
1	2	1
1	3	1
1	4	1
1	5	1
75		

Success

 Record deleted successfully!

OK

Search

Delete

Back

▪ Delete Details


Delete Record

Enter Student ID (SID):

Select Semester:

SID	Name	Contact	Semester
1	Rohit	98134567890	1

Success

 Record deleted successfully!

OK

Search

Delete

Back

▪ Show Marks

Marksheet Generator

Enter Student ID (SID):

1

Enter Semester:

1

Subject	Marks
Computer Fundamentals and Applications	67
Society and Technology	89
English I	90
Mathematics I	89
Digital Logic	90

Name: Rohan, Course: BCA

Current Semester: 1

Percentage: 85.00%, Grade: A, GPA: 4.00

Search

Save PDF

Print Marksheet

Back

▪ Show Details

Display Students

Rohit



Search

Back

SID	Name	Course	Address	Contact	Semester
1	Rohit	BCA	Lahan, Mithila	98134567890	1

■ Generate Result

☆

1 / 1 | - 100% + |  

Enter Student ID (SID):

1

Enter Semester:

1

Subject	Marks
Computer Fundamentals and Applications	67
Society and Technology	89
English I	90
Mathematics I	89
Digital Logic	90

Name: Rohan, Course: BCA

Current Semester: 1

Percentage: 85.00%, Grade: A, GPA: 4.00

Search

Save PDF

Print Marksheet

Back