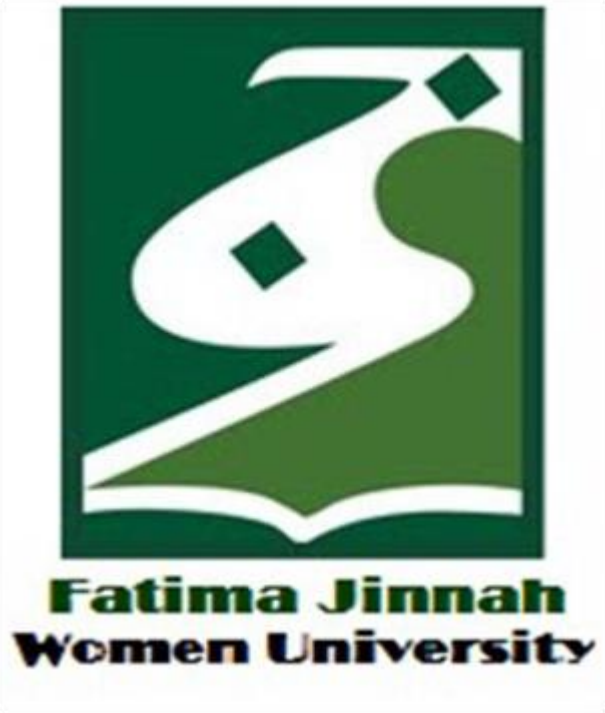
OBJECT ORIENTED PROGRAMMING PROJECT DOCUMENT

Student result management system



Submitted to :

MR ADEEL

Submitted by :

Maimoona bibi (055)

Malayka ibrar (058)

Natasha Maryam (068)

Introduction

A Student Result Management System (SRMS) is a software application designed to manage and maintain the results of students in an educational institution. Using Java Swing for the user interface, we can create a desktop application that is both user-friendly and efficient. This document outlines the key concepts and components involved in developing such a system using Java Swing.

Key Components

1. User Interface (UI)

The UI is built using Java Swing, a part of Java's Standard Library that provides a set of "lightweight" (all-Java language) components. Key components include:

- **JFrame**: The main window of the application.
- **JPanel**: A container that can hold a group of components.
- **JLabel**: Displays text or images.
- **JTextField**: A text input field.
- **JButton**: A button that triggers actions when clicked.
- **JTable**: Displays data in a tabular format.

2. Database Integration

To store and retrieve student results, we use a database. Java Database Connectivity (JDBC) is used to interact with the database.

3. Event Handling

Event handling in Swing is used to handle actions such as button clicks. Event listeners are added to components to define what happens when an event occurs.

4.Functional Features

- **Add Results**: Enables users to add new student result records by entering details into the form and saving them to the database.
- **View Results**: Displays student results in a tabular format (`JTable`) fetched from the database, allowing users to easily view all stored records.
- **Update and Delete**: Potential for implementing features to update existing records (e.g., marks) or delete records as necessary to maintain accuracy and relevance.

5. Data Management

- **Data Validation**: Includes validation checks to ensure the correctness of data entered (e.g., valid numeric marks, non-empty fields).
- **Error Handling**: Provides appropriate error messages or notifications in case of database connectivity issues or invalid data entry.

6. Usability and Accessibility

- **User Feedback**: Offers informative messages upon successful data insertion or updates and handles errors gracefully to enhance user experience.
- **Responsive Design**: Designed to be responsive and scalable, accommodating additional features such as search functionality or report generation.

Structure of this system:

The structure of the Student Result Management System (SRMS) using Java Swing and SQL database integration can be outlined as follows, focusing on its architectural components and organization:

1. **Main Frame (`JFrame`)**:
- Acts as the primary window of the application.

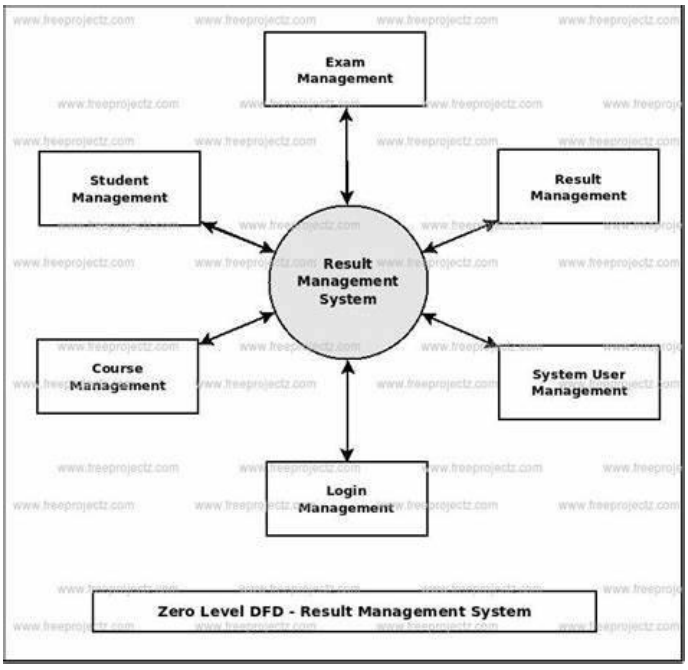
- Contains navigation elements such as buttons or menus for accessing different functionalities.
- Provides a central hub for user interaction and control.

2. **Forms (`JPanel`, `JTextField`, `JButton`)**:

- Used for data entry and interaction.
- Includes forms for adding new student results (`AddResultFrame`) and potentially for updating or deleting existing results.
- Each form includes appropriate input fields (`JTextField` for text input, `JButton` for submission) and labels (`JLabel`) for clarity.

3. **Result Viewing** :

- Displays student results in a tabular format (`JTable`).
- Uses `JScrollPane` to manage scrolling when displaying a large number of records.
- Allows users to view all stored results with columns for student ID, name, subject, and marks.



Database Connection:

Integrating a Java Swing-based Student Result Management System with an Oracle database involves configuring the JDBC driver, establishing a connection, and managing database operations effectively. By following these

steps and best practices, you can ensure a robust and secure database connection for your application. Adjust the connection details (URL, USER, PASSWORD) according to your specific Oracle database configuration for seamless integration and reliable data management.

```
import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

public class OracleDatabaseConnection {

   private static final String URL = "jdbc:oracle:thin:@localhost:1521:xe"; // Replace with your Oracle URL

   private static final String USER = "username";

   private static final String PASSWORD = "password";

  public static void main(String[] args) {

       Connection connection = null;

       try {

          // Register Oracle JDBC driver

          Class.forName("oracle.jdbc.OracleDriver");

//Establish connection

          connection = DriverManager.getConnection(URL, USER, PASSWORD);

          if (connection != null) {

             System.out.println("Connected to Oracle database!");

             // Perform database operations here    }

       } catch (ClassNotFoundException e) {

          System.out.println("Oracle JDBC driver not found.");
```

```
            e.printStackTrace();
        } catch (SQLException e) {
            System.out.println("Connection failed.");
            e.printStackTrace();
        } finally {
            // Close connection
            try {
                if (connection != null) {
                    connection.close();
                    System.out.println("Connection closed."); }
            } catch (SQLException e) {
                e.printStackTrace();  }  } }}
```

Conclusion

This document provides an overview of developing a Student Result Management System using Java Swing. The system includes a main frame for navigation, forms for adding results, and tables for viewing results. JDBC is used for database interaction, and event handling is implemented to manage user actions. This framework can be extended with additional features like updating and deleting records, advanced search, and report generation.