Student Name	**Ronit Rao**
Registration Number	**230968182**
Assignment number	**IA-3**
Subject code	**DSE-2123**
Subject	**OOP with Java**
Marks	**10**

Title Page

Student Grade Management System Documentation
Developed By: Ronit Rao
Date: 30/10/2024

Problem Statement

Problem Statement 8: Develop a GUI-Based Student Grade Management System.

Requirements:

1. **Login Page:**
   Create a login interface for students and administrators with role-based access.
2. **Grade Entry and Management:**
   Administrators should be able to add, edit, and delete student grades. Each student's grade record should include subject, marks, and grade.
3. **Student View:**
   Students can log in to view their grades for each subject, including the overall GPA.
4. **Search and Filter:**
   Implement a search feature for administrators to filter students by name, grade, or subject.

Methodology

The Student Grade Management System is built using Java Swing for the GUI components and follows an object-oriented programming approach. The system consists of multiple classes, each serving a specific function, from managing user data to handling grades and providing the user interface.

Class Descriptions

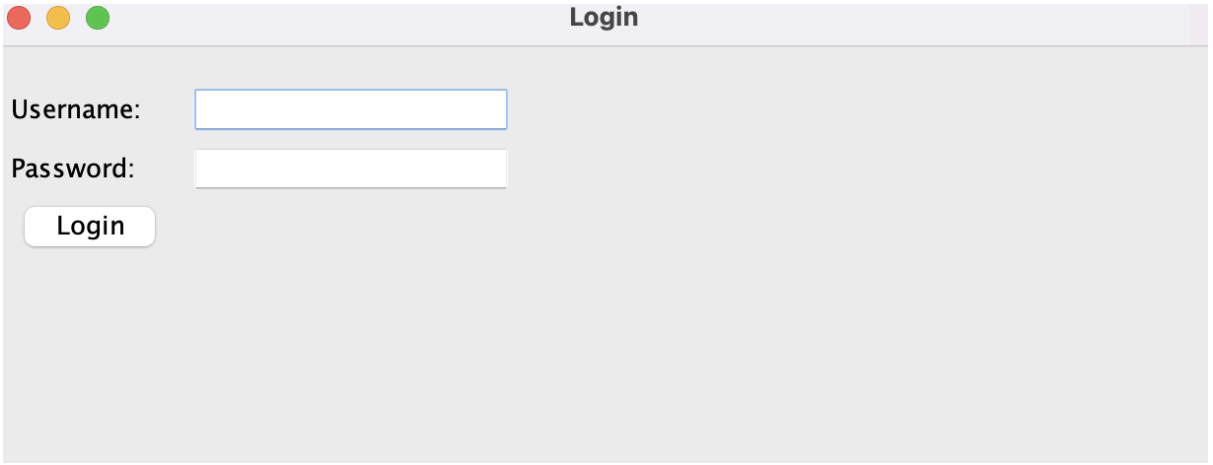
1. `Grade` **Class**
   Represents a student's grade in a specific subject, containing fields for the subject name, marks obtained, and the corresponding grade.
2. `GradeManager` **Class**
   Manages the collection of grades for all students. It allows adding, retrieving, modifying, and deleting grades. It also handles loading and saving grades to a CSV file for persistence.
3. `User` **Class**
   Represents a user with fields for username, password, and role (admin or student). This facilitates role-based access control during login.
4. `LoginPage` **Class**
   Displays the login interface, where users can enter their credentials. It validates the credentials and redirects users to either the admin or student view.
5. `AdminView` **Class**
   Provides the interface for administrators to manage student grades, including adding, modifying, and deleting grades. It also includes filtering capabilities for viewing specific records.
6. `AddGradeDialog` **Class**
   A dialog for entering new grades for a selected student. It validates input and updates the grade manager upon submission.
7. `ModifyGradeDialog` **Class**
   A dialog that allows administrators to modify existing grades. It pre-fills the dialog fields with the current grade data and updates the manager after changes.
8. `StudentView` **Class**
   Allows students to view their grades and GPA after logging in. It displays a table of grades and calculates the GPA based on the grades.
9. `StudentGradeManagementSystem` **Class**
   The main entry point of the application. It initializes the grade manager and launches the login page.

How to Use the Code

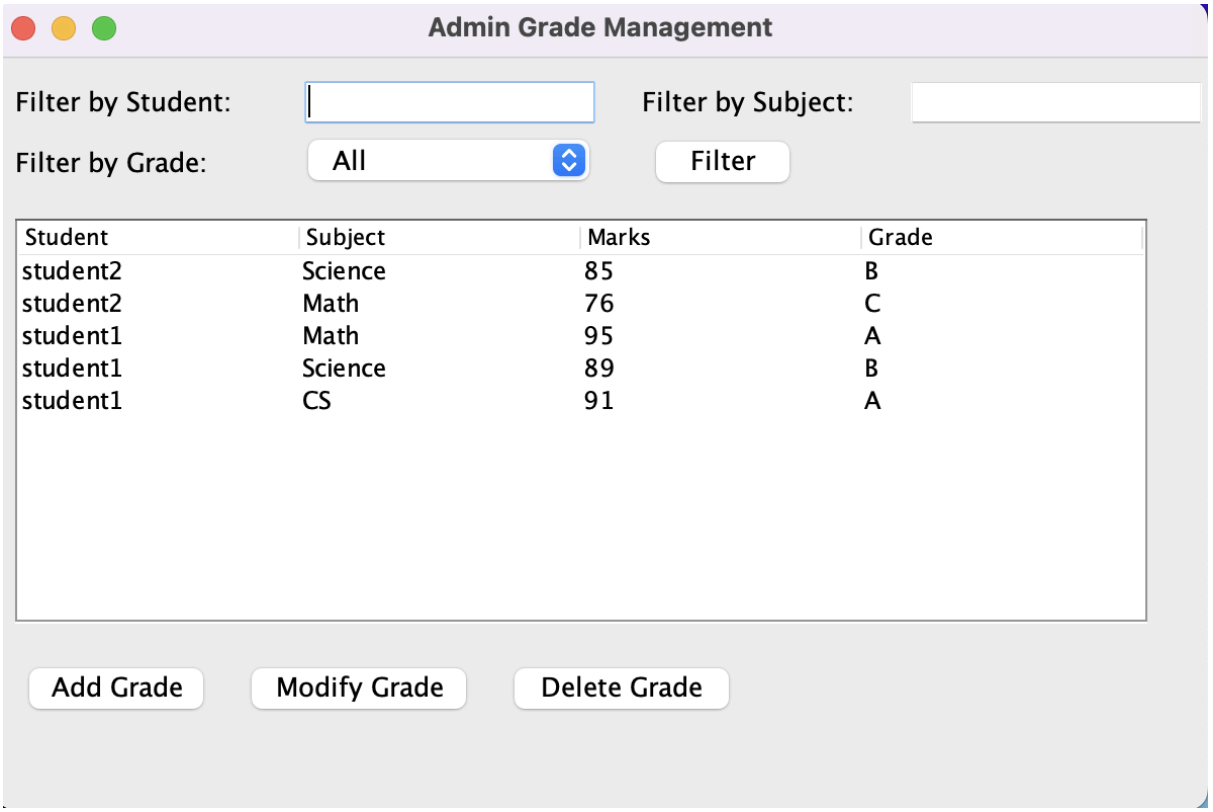
1. **Compile the Code:**
   Ensure you have Java Development Kit (JDK) installed. Compile the code files using your IDE or command line.
2. **Run the Application:**
   Execute the `StudentGradeManagementSystem` class to start the application.
3. **Login:**
   - Use the following credentials for admin access:
     Username: `admin`
     Password: `adminpass`
   - For students, use usernames that exist in the grade manager (e.g., `student1`, `student2`) and password for both the student logins are `studentpass`
4. **Admin Functions:**
   - Once logged in as an admin, you can add, modify, or delete grades.
   - Use the filter fields to search for specific students or subjects.
5. **Student Functions:**
   - After logging in as a student, view your grades and overall GPA.

Results and Screenshots

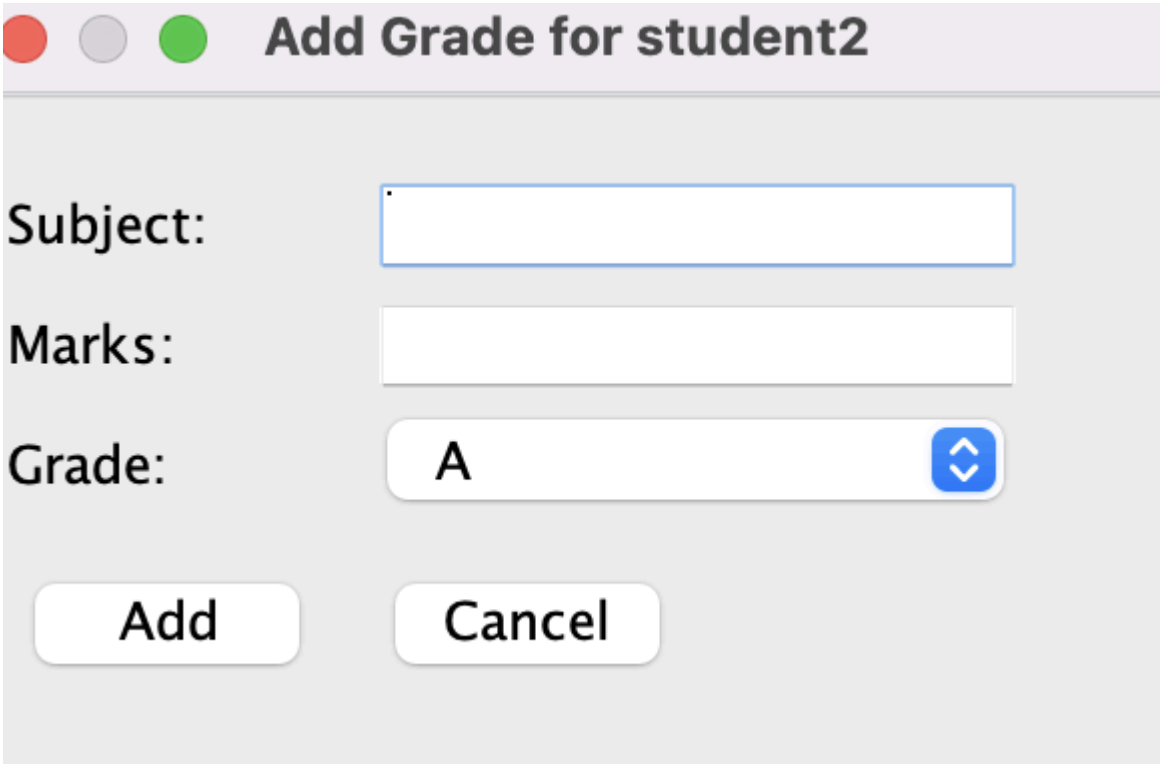
Login Window



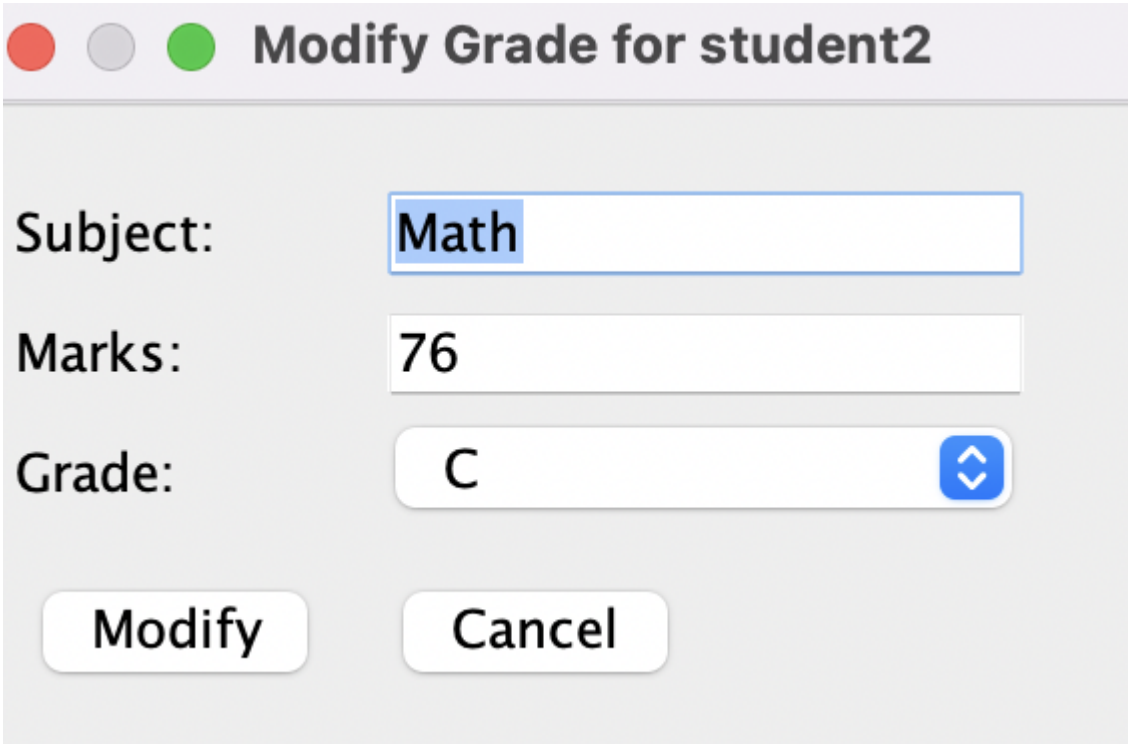
Admin Window View



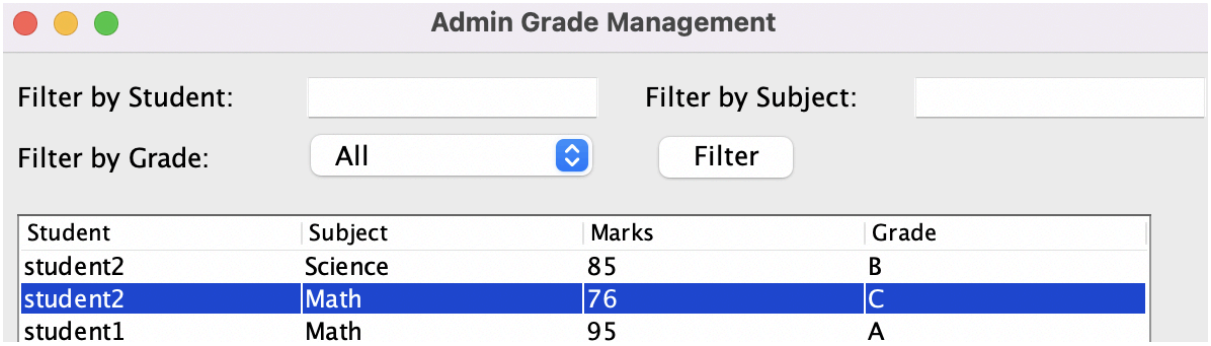
Add Grade Window



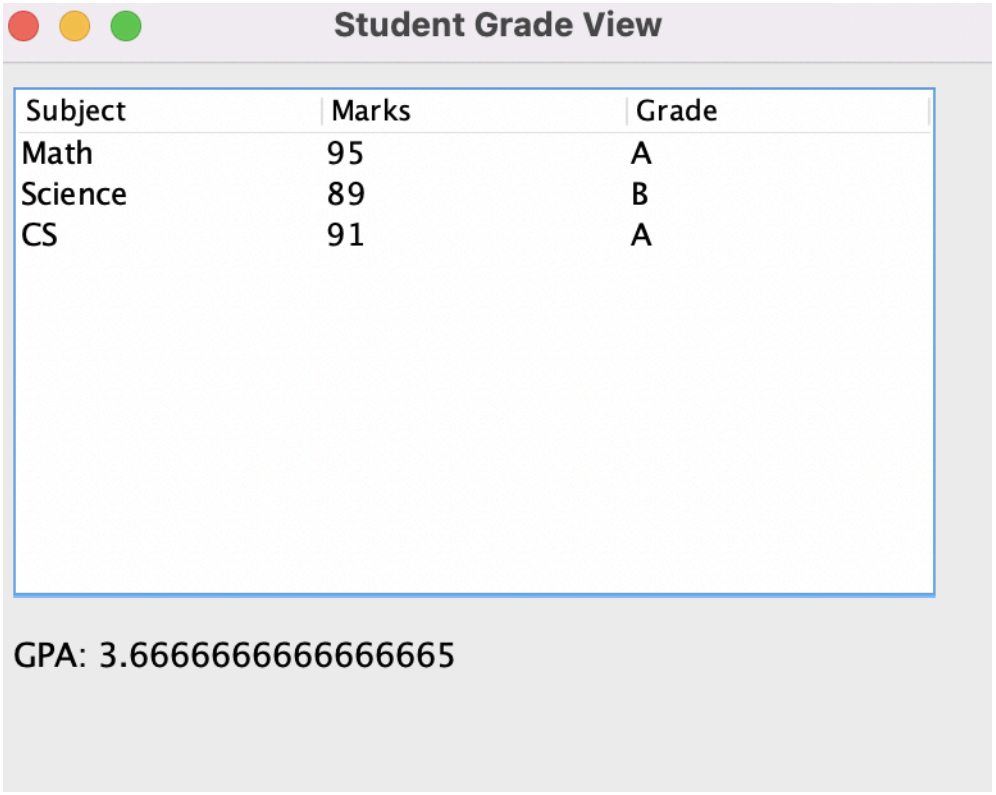
Modify Grade Window



Filter window(by student/grade/Subject)



Student view window



Program Code with Comments

Code in Zip File.

1. ***AddGradeDialog.java***
2. ***AdminView.java***
3. ***Grade.java***
4. ***GradeManager.java***
5. ***grades.csv***
6. ***LoginPage.java***
7. ***ModifyGradeDialog.java***
8. ***StudentGradeManagementSystem.java***
9. ***StudentView.java***
10. ***User.java***

References

- Java Documentation: https://docs.oracle.com/en/java/javase/
- Java Swing Tutorial: https://www.tutorialspoint.com/swing/index.html