School Management System

Enterprise Application Development - 1

Project Report

Diploma in Software Engineering

23.2F

Submitted by

CODSE232F-062 - T M T N KUMARI

School of Computing

National Institute of Business Management

Colombo 7

Table of Contents

Introduction

The School Management System is an enterprise-level software application designed to automate and streamline key administrative tasks within educational institutions. This enterprise application system aims to simplify the management of student information, providing a comprehensive solution for handling student records, calculating Grade Point Averages (GPA), and generating student's exam reports accurately.

Key Features

1. **Login Form** - Provides secure access to the system.
2. **Main Menu Form** - Acts as a central navigation hub to access different functionalities.
3. **Student Registration Form** - Allows for the addition of new student records.
4. **GPA Calculation Form** - Facilitates the calculation of GPAs based on student marks.

Tools Used

- **Java** - For developing applications.
- **NetBeans v18** - The integrated development environment (IDE).
- **MySQL** - For database management.
- **JasperSoft studio** - For generating reports.

The system is designed to be user-friendly and efficient, aiming to support administrative tasks and improve overall management within educational institutions.

Requirement analysis

Functional requirements

Student Record Management

- The system must allow users to add, update, and delete student records, including personal details and contact information.
- Furthermore, this system helps to search for students by their Student ID.

GPA Calculation

- Users should be able to input marks for multiple subjects, with the system calculating the GPA based on these marks and predefined grading scales.

Report Generation

- The system must generate detailed reports on student performance, including GPA reports and overall academic performance, which can be used for decision-making.

Non-functional requirements

Usability

- The application must provide a user-friendly interface that is easy to navigate and interact with, ensuring a smooth user experience.

Security

- The application must ensure data security, including secure login procedures and protection of sensitive student information.

Major scenario

Administrator Login and Student Registration

- An administrator logs into the system using their credentials. Upon successful login, they gain access to the Student Registration Form. The administrator enters the new student's personal details, such as name, student ID, contact information, and other relevant data. The system allows the administrator to update or delete existing student records as needed through the same form.

Data Management

- The administrator can search for any student's details by entering their student ID into the search feature of the Student Registration Form. This functionality helps quickly locate and manage student records.

GPA Calculation

- The administrator enters marks for the student across multiple subjects (10 subjects) into the GPA Calculation Form. The system calculates the GPA based on these marks and predefined grading criteria. Additionally, it displays the total marks and average marks for the student.

Report Generation

- The calculated GPA and academic performance data are used to generate detailed performance reports. These reports provide insights into the student's academic progress and are useful for evaluating and making informed decisions regarding the student's academic standing.

System design

UI design

Layout and Navigation

The UI features a clean, organized layout with a main menu for easy access to all forms and functions.

Form Design

Forms are designed with clear labels and input fields, ensuring a straightforward data entry process.

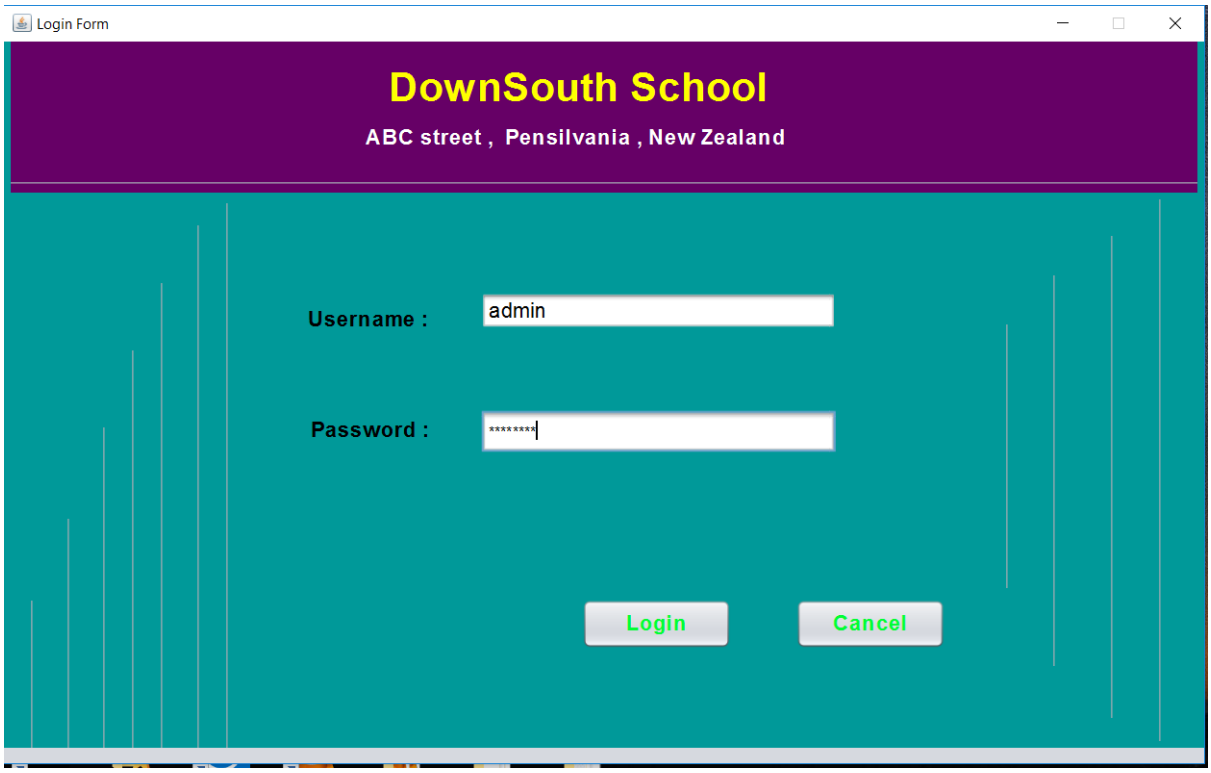
Visual Style

Consistent colors and fonts are used throughout to enhance readability and create a unified look.

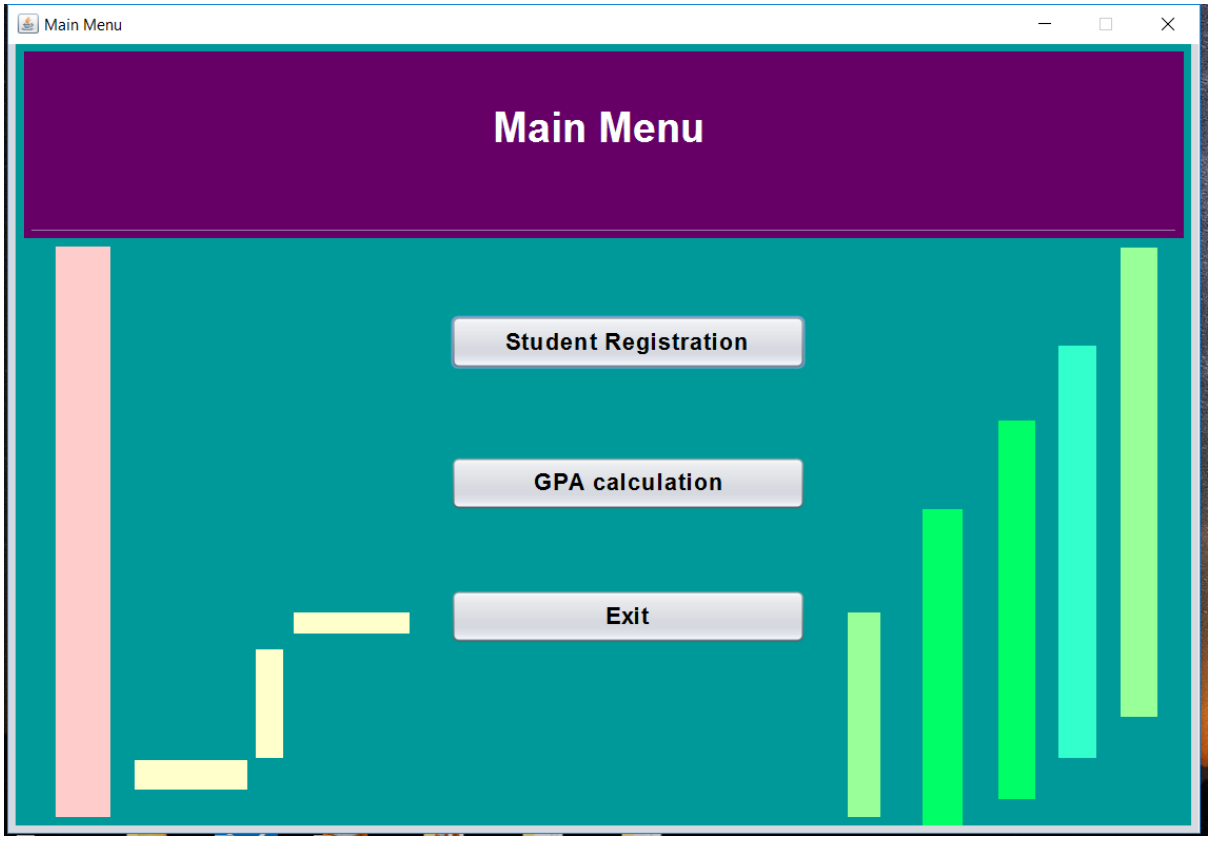
Feedback and Interaction

Interactive elements like buttons provide immediate feedback, helping users understand and correct their actions quickly

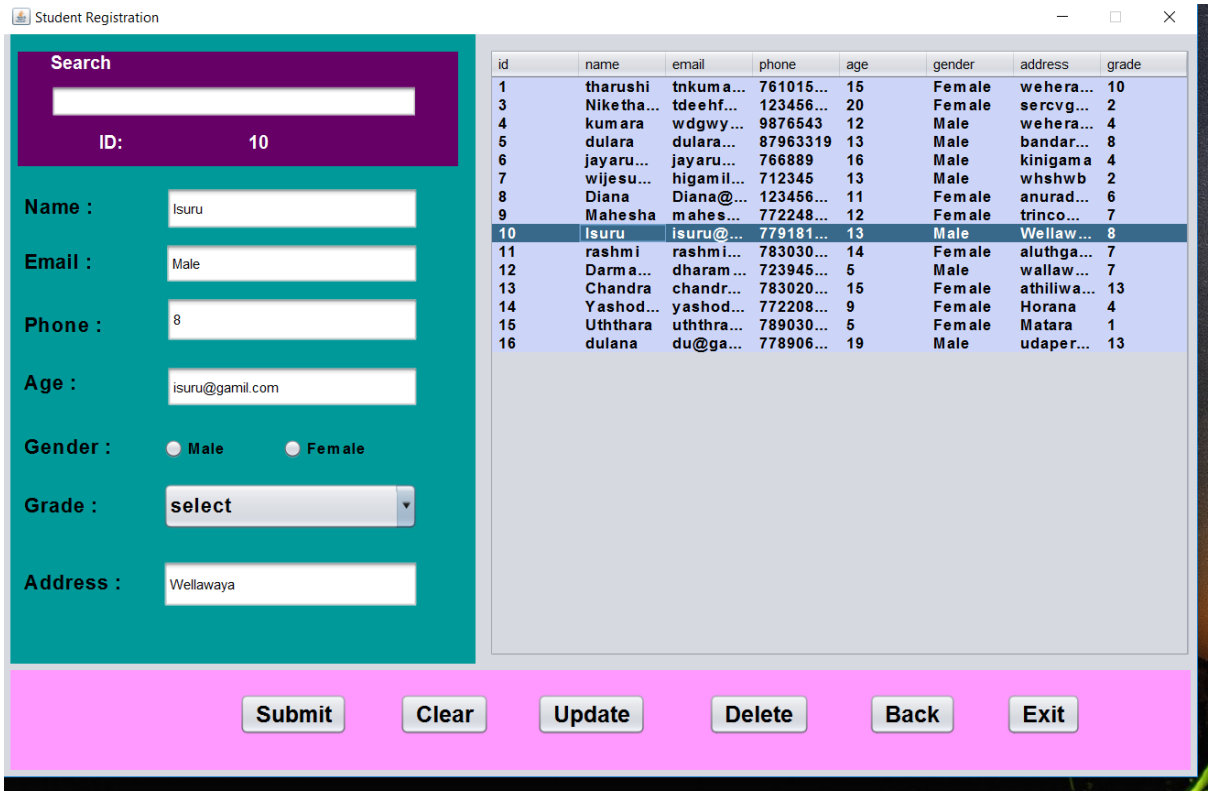
Login Form



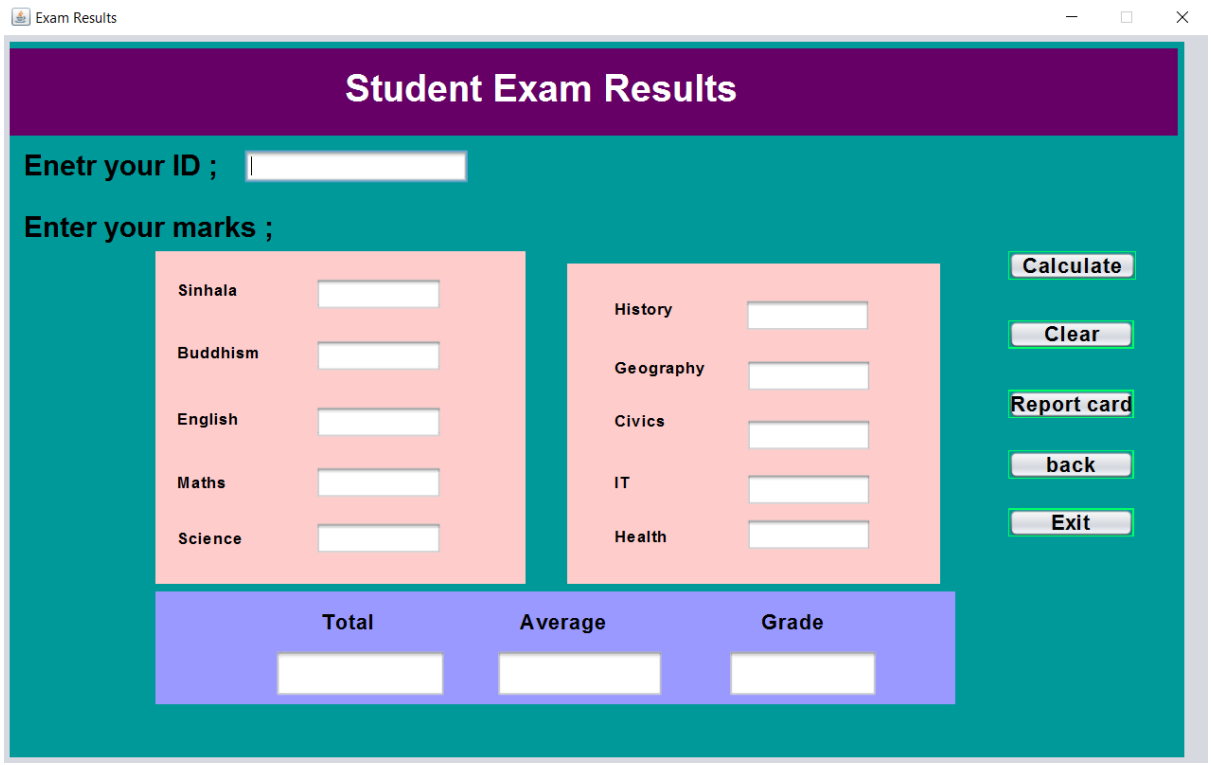
Menu Form



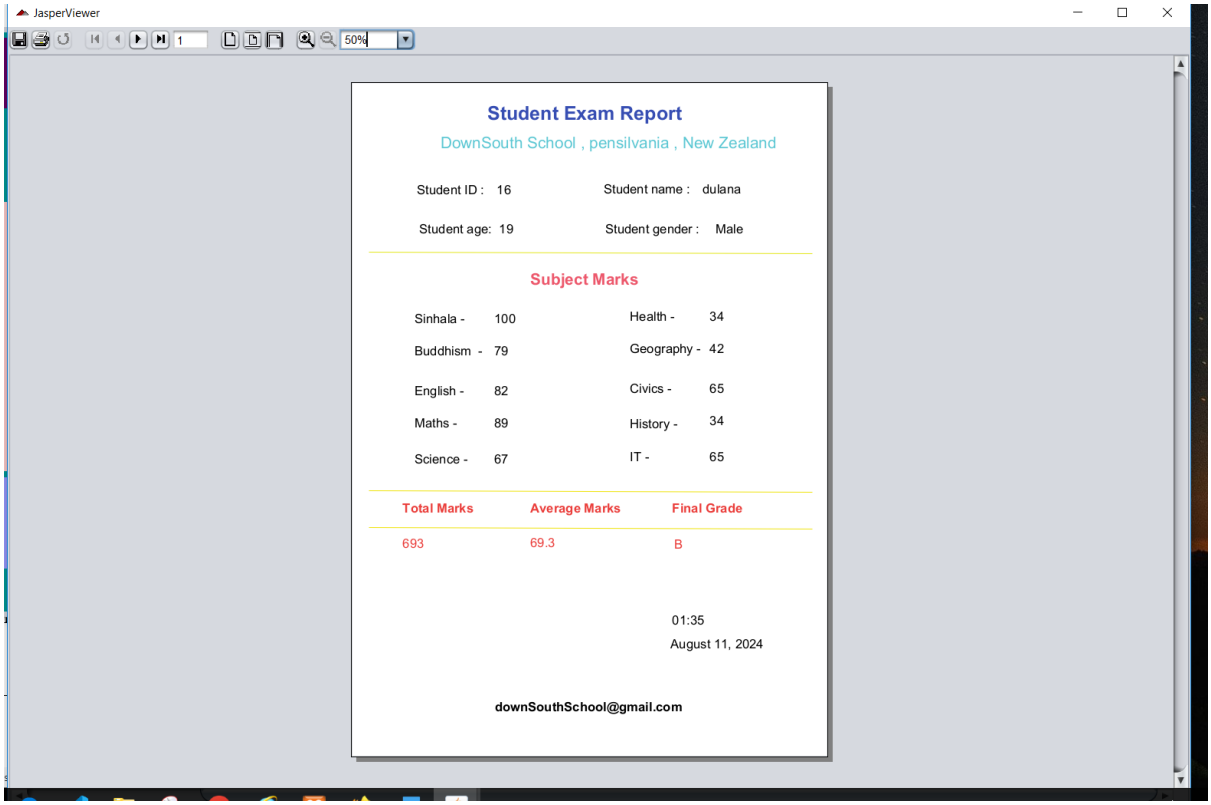
Student Registration Form



GPA calculation Form



Jasper Report



Implementation

Development Environment

- **Tools and Languages**

  - ✓ Java is used for application development
  - ✓ Apache NetBeans IDE 22 serves as the development environment.
  - ✓ Reporting is handled by Jaspersoft Studio version 6.8.0.

- **Database**

  - ✓ MySQL is managed using phpMyAdmin.
  - ✓ XAMPP server is used to run the database.
  - ✓ Database connectivity is established using MySQL Connector/J.

- **Libraries**

JasperReports and related JAR files include,

- ✓ common-logging-1.2.jar
- ✓ commons-digester-2.1.jar
- ✓ commons-beanutils-1.9.4.jar
- ✓ jasperreports-6.8.0.jar
- ✓ jasperreports-javaflow-6.8.0.jar
- ✓ rz2xml.jar.

Code Structure

- **Packages**

  - **Code Package**

    - ✓ Contains classes responsible for core functionalities, including database operations and business logic.
    - ✓ This package includes the class for handling database connections and other related operations.

  - **Interface Package**

- ✓ Contains classes for user interface components, such as forms and dialogs.
- ✓ This package includes the main class LoginForm, which serves as the entry point for the application.

- **Key Classes**
  - **LoginForm -** The main class of the application, responsible for user authentication and navigation to other forms.
  - **Database Connection Class** - Manages the connection between the application and the MySQL database. This class handles connection setup and operations.
  - **Other Classes** - Include those that handle student registration, GPA calculation, and report generation.

Database Design

- **Schema** - The database schema includes two tables
  - **student Table** - Contains 8 columns for student details including auto incrementing id.
  - **marks Table** - Contains 15 columns for subject marks, grade, total, and average, with an auto-incrementing id and student's table id acting as a foreign key (st_id) in the marks table.

Validations and exception handling

Input Validations

- **Login Form**
    - **Username and Password** - Must match the credentials (username: admin, password: admin123). If fields are empty, an error message is shown. Incorrect credentials result in a specific error message.
- **Student Registration Form**
    - **Required Fields** - All inputs must be filled except for the address, which can be null. Tooltip messages are provided for each text field.
    - **Age** - Must be between 5 and 22 years.
    - **Email** - Must include the '@' character.
    - **Phone Number** - Must be exactly 10 digits and cannot include any characters.
    - **Student ID** - Cannot be edited by the user.
    - **Gender Selection** - Only one radio button can be selected at a time.
- **GPA Calculation Form**
    - **Marks Input** - Marks must be numeric and within the range of 0 to 100. Fields for total, average, and grade are non-editable. Marks input is optional for absent exams.
- **Record Deletion**
    - A confirmation message is displayed before deleting a record from the database.
- **Form Navigation**
    - When navigating to a new form, the existing form is disposed of to free up resources.

Exception Handling

- **Form Validation**
    - Exceptions are managed by displaying appropriate error messages for invalid inputs or missing data.
- **Database Operations**
    - Errors related to database connections or operations are handled gracefully, with user-friendly error messages provided.
- **Report Generation**
    - Reports can only be generated for students existing in the database, preventing errors related to invalid student IDs.

Assumptions

- The system assumes that users have a working internet connection for accessing the database and generating reports.
- The application is designed to run on systems compatible with Java, including common operating systems like Windows.
- The user interface is designed to be intuitive and assumes users are familiar with basic form operations and input requirements.
- The database and reporting tools (e.g., MySQL, Jaspersoft Studio) are correctly configured and operational.
- Operating system have XAMPP server and Apache and MYSQL is started

Features and Functionality

Transaction UI

- The main transaction interface is the **Student Registration Form**. It represents a major functionality by allowing administrators to manage student records. This form supports adding, updating, and deleting student information. It also provides search functionality to locate student details by ID and ensures data validation for accurate entry.

Input UIs

- **Student Registration Form** - Used to input student details such as name, age, gender, grade, email contact information, and address. It includes validation rules and tooltips to guide users.
- **GPA Calculation Form** - Used to input and manage student marks across multiple subjects. It calculates GPA and displays total marks and average, with non-editable fields for results.
- **Login Form** - Allows administrators to log into the system using predefined credentials. It includes validation for correct username and password entries and error messages for invalid inputs.

Dashboard

- The dashboard provides a central view of key functionalities, including access to student registration and GPA calculation and exit. It helps administrators navigate the application efficiently and view summaries of student data and performance metrics.

Report Generation

- The system generates performance reports based on student marks and GPA calculations. The reports pull data from the student and marks tables, providing detailed insights into individual student performance. These reports are used to evaluate academic progress and support decision-making. Only records with existing student IDs are included in the reports to ensure data accuracy.

Testing

Test Case ID	Test Case Scenario	Test Case Meta Data	Expected Result	Actual Result	Pass/Fail
TC01	Enter email without '@'	tnkgamil.com	Display error message about invalid email format	“Please enter a valid email with ‘@””	Pass
TC02	Log in without entering any credentials	Username – null Password - null	Display error message for empty fields	“Username and password cannot be empty”	Pass
TC03	Log in with incorrect credentials	Username – hi Password - 74847	Display error message for incorrect username/password	“Invalid username or password”	Pass
TC04	Enter mobile number with 11 digits	Phone = 12345678900	Display error message for invalid mobile number length	“Please enter a valid number”	Pass
TC05	Enter mobile number with characters	Phone = 123456he78	Display error message for invalid characters in mobile number	“Please enter a valid number”	Pass
TC06	Enter age with characters	Age = 2k	Display error message for invalid characters in mobile number	“Please enter a valid age”	Pass
TC07	Enter age greater than 22 or less than 5	Age = 23	Display error message for age out of range	“Age must between 5 and 22”	Pass
TC08	Leave required fields empty and click submit	Name = null	Display error message for missing required fields	“Name field is required”	Pass
TC09	Generate a report for a student not in the database	Student id = 100	Display error message for student not found	“Database error : cannot update a child row : foreign key constraint fails”	Pass

TC10	Input characters for marks	Sinhala – 5d	Display error message for invalid marks input	"Marks must be a valid number between 0 and 100"	Pass
TC11	Input marks out of range (0 to 100)	English - 101	Display error message for marks out of range	"Mark must be between 0 and 100"	Pass
TC12	Calculate GPA with partial marks	Maths – 78, It - 92	Correct GPA calculation based on provided marks	Total = 170 Average = 1.7 Grade = "F"	Pass

Challenges Faced

JAR Files Compatibility

- Managing various JAR files for JasperReports and ensuring compatibility with your project. Different versions of JAR files were causing issues, particularly with JasperReports when compiling JRXML files to XML and integrating them into the NetBeans Java Ant project.
- Resolved this by researching JasperReports tutorials to understand compatibility requirements. Although the tutorials used older versions, they provided useful insights. Additionally, consulted resources like Maven repositories and SourceForge to find compatible versions of the required JAR files.

JasperReports Integration

- Integrating JasperReports into the Java Ant project was problematic due to version conflicts and difficulties in compiling JRXML files.
- Searched for tutorials and guidance on JasperReports integration. Utilized ChatGPT and youtube to identify compatible versions and solutions for integration issues. Leveraged Maven repositories and SourceForge to obtain the necessary libraries and dependencies.

Conclusion

The School Management System project aimed to streamline the management of student information within educational institutions. It successfully implemented core functionalities including student registration, GPA calculation, and report generation. By creating a user-friendly interface and integrating robust validation and exception handling, the system ensures accurate and efficient management of student records.

- **Effective Management** - Enabled administrators to efficiently register, update, and manage student details.
- **GPA Calculation** - Automated GPA calculations and provided clear performance metrics.
- **Report Generation** - Facilitated the creation of detailed performance reports to assist in evaluating academic progress.

The project improves administrative efficiency, enhances data accuracy, and supports informed decision-making through comprehensive reports. The use of advanced UI components, careful validation, and seamless integration with reporting tools contributes to a reliable and user-friendly system.

Appendices

Maven Repository

- **JasperReports Dependencies** - Maven Central Repository - JasperReports
- **Apache Commons Dependencies** - Maven Central Repository - Apache Commons

JasperReports Tutorials

- **Official JasperReports Documentation** - JasperReports Documentation
- **JasperReports Tutorial (Version 6.8.0)** - JasperReports Tutorial by Jaspersoft
- **JasperReports Tutorial on TutorialsPoint** - TutorialsPoint JasperReports

SourceForge

- **JasperReports Project on SourceForge**: JasperReports on SourceForge
- **Apache Commons Project on SourceForge**: Apache Commons on SourceForge