College Admission Management System

Project Report

1. Introduction

The **College Admission Management System** is a Java-based console application designed to streamline and automate the student admission process for educational institutions. The system handles student registration, merit calculation, admission assignments, and displays final admission results. This project demonstrates proficiency in fundamental Java programming, object-oriented design, JDBC, and database management with SQLite.

2. Objectives

- Provide a user-friendly, menu-driven application for college admissions.
- Ensure reliable storage and retrieval of student and admission information.
- Implement merit-based admission assignment logic.
- Prevent duplicate admission entries.
- Demonstrate database integration and robust exception handling.

3. Functional Requirements

- **Student Registration:**
    - Add, view, and manage student profiles (name, date of birth, score, contact, address).
- **Merit Calculation:**
    - Assign merit to students based on their academic score.
- **Admission Assignment:**
    - Assign qualified students to a selected program based on merit.

    - Prevent duplicate admissions for the same student and program.

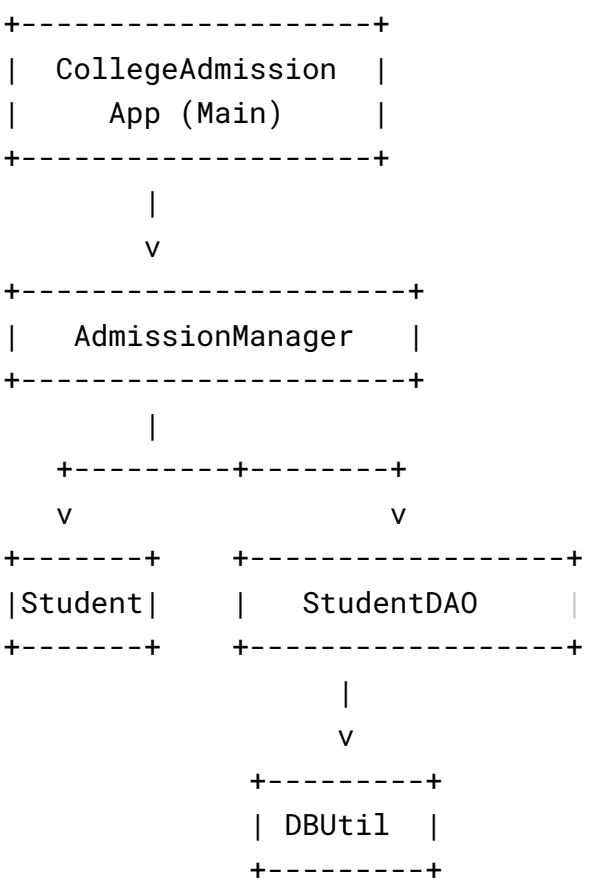
- **Result Display:**
    - Show admission list sorted by merit.

4. System Design

4.1 Architecture

- **Layered Approach:**
  The system is structured into entity/model classes, data access layer (DAO), utility classes for the database, and the main application interface.

text



4.2 Database Schema

Tables:

Table	Columns
students	id, name, dob, score, contact, address
admissions	id, student_id, program, merit

5. Technologies Used

- **Programming Language:** Java (JDK 8+)
- **Database:** SQLite (embedded, file-based)
- **Database Connector:** JDBC

6. Implementation Overview

6.1 Main Components

- `Student.java`: Model class for student details.
- `DBUtil.java`: Database connection utility class.
- `StudentDAO.java`: Handles CRUD operations for students.
- `AdmissionManager.java`: Business logic for assigning admissions and displaying results.
- `CollegeAdmissionApp.java`: Main application providing the console-based menu.

6.2 Key Operations

- **Add Student:** Collects data from user, validates, and stores in the database.
- **List Students:** Fetches and prints all registered students from the database.
- **Assign Admissions:** Assigns students to a program based on merit (score), avoiding duplicate assignments.
- **Show Admissions:** Displays the list of admitted students for the selected program, ordered by merit.

7. Exception Handling & Best Practices

- All JDBC resources are managed using try-with-resources for safety.
- Queries use prepared statements to prevent SQL injection.
- Duplicate admissions are checked for and handled gracefully.
- User input is validated before processing.
- Null and empty result handling provides informative feedback.

8. Sample Console Output

```
--- College Admission Management System ---
1. Add Student
2. List All Students
3. Assign Admissions
4. Show Admission Results
0. Exit
Enter choice: 1
Enter student name: Priya
...
Student added successfully!

--- College Admission Management System ---
...
Admission List:
ID | Student Name | Program | Merit
----------------------------------
1  | Priya       | BSc     | 87.00
```

. . .

9. Advantages

- **Educational Value:** Covers critical technical concepts often required in interviews.
- **Extensibility:** Structure allows easy additions (GUI, additional features).
- **Database Independence:** Swappable SQLite/MySQL backend.

10. Possible Enhancements

- Add support for program eligibility criteria.
- Export admission results to CSV/PDF.
- Integrate a graphical user interface (Swing/JavaFX).
- Implement user roles (admin, operator).

11. Conclusion

The College Admission Management System fulfills all stated objectives, providing a robust, scalable, and interview-ready Java application. By following best coding and database management practices, the project is both reliable and easy to extend for future requirements.

Prepared by: [Janani U S]
Date: July 25, 2025