**TRƯỜNG ĐẠI HỌC ĐÀ LẠT**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO CUỐI KỲ**

**MẪU THIẾT KẾ**

**ĐỀ TÀI: QUẢN LÝ ĐIỂM SINH VIÊN**

**Giáo viên hướng dẫn:** Đoàn Minh Khuê

**Sinh viên thực hiện:** 2212436 Phan Thành Phát

***Đà Lạt, tháng 10 năm 2025­­***

**NHẬN XÉT CỦA GIÁO VIÊN**

Đà Lạt, ngày … tháng … năm ...

Giảng viên

[Kí tên và ghi rõ họ tên]

# **MỤC LỤC**

[**MỤC LỤC** 3](#_Toc212455060)

[**Lời mở đầu** 4](#_Toc212455061)

[**1. Giới thiệu dự án** 5](#_Toc212455062)

[**2. Mục tiêu và phạm vi** 5](#_Toc212455063)

[**3. Công nghệ và thư viện sử dụng** 6](#_Toc212455064)

[**4. Kiến trúc hệ thống** 6](#_Toc212455065)

[**5. Thiết kế chi tiết** 7](#_Toc212455066)

[**6. Chức năng chính** 12](#_Toc212455067)

[**7. Hướng dẫn cài đặt và chạy** 13](#_Toc212455068)

[**8. Hướng dẫn sử dụng (tóm tắt)** 13](#_Toc212455069)

[**9. Kiểm thử và kết quả** 14](#_Toc212455070)

[**10. Hạn chế và đề xuất nâng cấp** 15](#_Toc212455071)

[**11. Kết luận** 15](#_Toc212455072)

[**12. Phụ lục** 16](#_Toc212455073)

**­**

# **Lời mở đầu**

Trong bối cảnh giáo dục hiện đại, việc quản lý thông tin sinh viên và điểm số một cách chính xác, minh bạch và thuận tiện là yêu cầu thiết yếu cho công tác đào tạo. Báo cáo này trình bày chi tiết dự án "Hệ Thống Quản Lý Điểm Sinh Viên" được phát triển bằng Java, sử dụng cơ sở dữ liệu SQLite và áp dụng Strategy Pattern cho các thuật toán tính điểm. Mục tiêu của dự án là cung cấp một công cụ thân thiện, dễ dùng để quản lý sinh viên, quản lý điểm, tính toán điểm trung bình có trọng số, chuyển đổi điểm chữ và xếp loại học lực.

Tài liệu bao gồm mô tả yêu cầu, kiến trúc hệ thống, thiết kế chi tiết, hướng dẫn cài đặt và sử dụng, kết quả kiểm thử, hạn chế và các đề xuất phát triển tiếp theo.

# **1. Giới thiệu dự án**

Tên dự án: Hệ Thống Quản Lý Điểm Sinh Viên

Ứng dụng được phát triển nhằm hỗ trợ quản lý thông tin sinh viên và điểm số học tập. Giao diện hiện tại sử dụng Java Swing; dữ liệu được lưu trữ bằng SQLite để đảm bảo tính bền vững và dễ di chuyển. Dự án áp dụng các mẫu thiết kế phần mềm như MVC (Model-View-Controller), DAO (Data Access Object) và Strategy để tách biệt trách nhiệm, dễ bảo trì và mở rộng.

# **2. Mục tiêu và phạm vi**

**Mục tiêu chính:**

Cung cấp giao diện để thêm/sửa/xóa sinh viên và điểm.

Lưu trữ dữ liệu bền vững bằng SQLite.

Hỗ trợ nhiều phương pháp tính điểm (trung bình có trọng số, điểm chữ, xếp loại, GPA) thông qua Strategy Pattern.

Đóng gói ứng dụng thành file JAR có thể chạy độc lập trên Windows (kèm script .bat).

**Phạm vi:**

Quản lý sinh viên (mã SV, họ tên, email, số điện thoại, ngành học).

Quản lý điểm (môn học, điểm, hệ số, học kỳ, năm).

Tính toán và hiển thị kết quả tính điểm theo nhiều chiến lược.

Báo cáo tổng quan và bảng điểm chi tiết cho từng sinh viên.

# **3. Công nghệ và thư viện sử dụng**

**Ngôn ngữ:** Java SE 8+

**Giao diện:** Java Swing

**Cơ sở dữ liệu:** SQLite (qua JDBC driver sqlite-jdbc)

**Mẫu thiết kế:** MVC, DAO, Strategy

**Công cụ build/run:** javac/java hoặc IDE (ví dụ: VS Code)

**Script:** RunStudentManager.bat (Windows)

# **4. Kiến trúc hệ thống**

Hệ thống được tổ chức theo mô hình MVC:

**Model:** Student, Grade (gói com.studentmanager.model).

**DAO:** StudentDAO, GradeDAO, DatabaseConnection (gói com.studentmanager.dao) chịu trách nhiệm thao tác cơ sở dữ liệu.

**Controller:** StudentController xử lý nghiệp vụ, kết nối giữa GUI và DAO.

**View/GUI:** StudentManagementGUI (Swing) gồm các tab: quản lý sinh viên, quản lý điểm, tính điểm (Strategy), báo cáo.

**Strategy:** interface GradeCalculationStrategy và các lớp triển khai AverageGradeStrategy, LetterGradeStrategy, ClassificationStrategy, GPACalculationStrategy; GradeCalculator làm context.

Thiết kế phân lớp rõ ràng giúp tách biệt trách nhiệm: DAO chỉ tương tác DB, Controller quản lý logic, GUI chỉ lo hiển thị và nhận dữ liệu từ người dùng.

# **5. Thiết kế chi tiết**

## 5.1 Yêu cầu chức năng và phi chức năng

Yêu cầu chức năng (Functional Requirements)

Đăng ký/Quản lý thông tin sinh viên: thêm, sửa, xóa, tra cứu.

Quản lý điểm: thêm, sửa, xóa điểm cho từng sinh viên (môn, điểm, hệ số, học kỳ, năm).

Tính toán điểm: tính điểm trung bình có trọng số, quy đổi điểm chữ, xếp loại học lực, chuyển sang GPA.

Báo cáo: xuất bảng điểm chi tiết, báo cáo tổng quan, thống kê theo xếp loại.

Giao diện GUI thân thiện, có thể chạy độc lập (JAR) trên Windows.

Yêu cầu phi chức năng (Non-Functional Requirements)

**Hiệu năng:** xử lý tốt với cơ sở dữ liệu hàng nghìn bản ghi.

**Bảo mật:** tránh SQL injection (sử dụng PreparedStatement trong DAO).

**Tính di động:** chạy trên máy có Java 8+; cơ sở dữ liệu SQLite dễ di chuyển.

**Tính mở rộng:** dễ thêm chiến lược tính điểm mới (mở rộng Strategy).

## 5.2 Mô hình dữ liệu (Database schema)

Bảng students:

id INTEGER PRIMARY KEY AUTOINCREMENT

student\_id VARCHAR(20) UNIQUE NOT NULL

full\_name VARCHAR(100) NOT NULL

email VARCHAR(100)

phone\_number VARCHAR(15)

major VARCHAR(100)

Bảng grades:

id INTEGER PRIMARY KEY AUTOINCREMENT

student\_id INTEGER NOT NULL (FK → students.id)

subject VARCHAR(100) NOT NULL

score REAL NOT NULL (0..10)

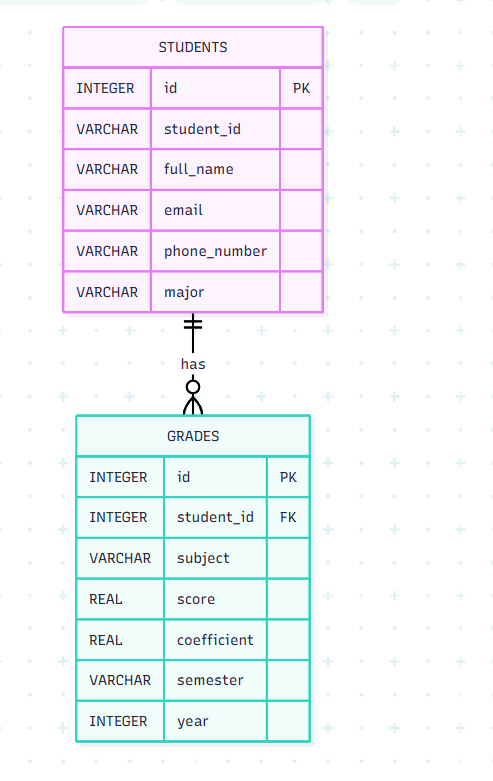
coefficient REAL NOT NULL DEFAULT 1.0

semester VARCHAR(20) NOT NULL

year INTEGER NOT NULL

Ràng buộc: quan hệ 1-n giữa students và grades (một sinh viên có nhiều bản ghi điểm).

## 5.3 Sơ đồ thực thể liên kết (ER Diagram)

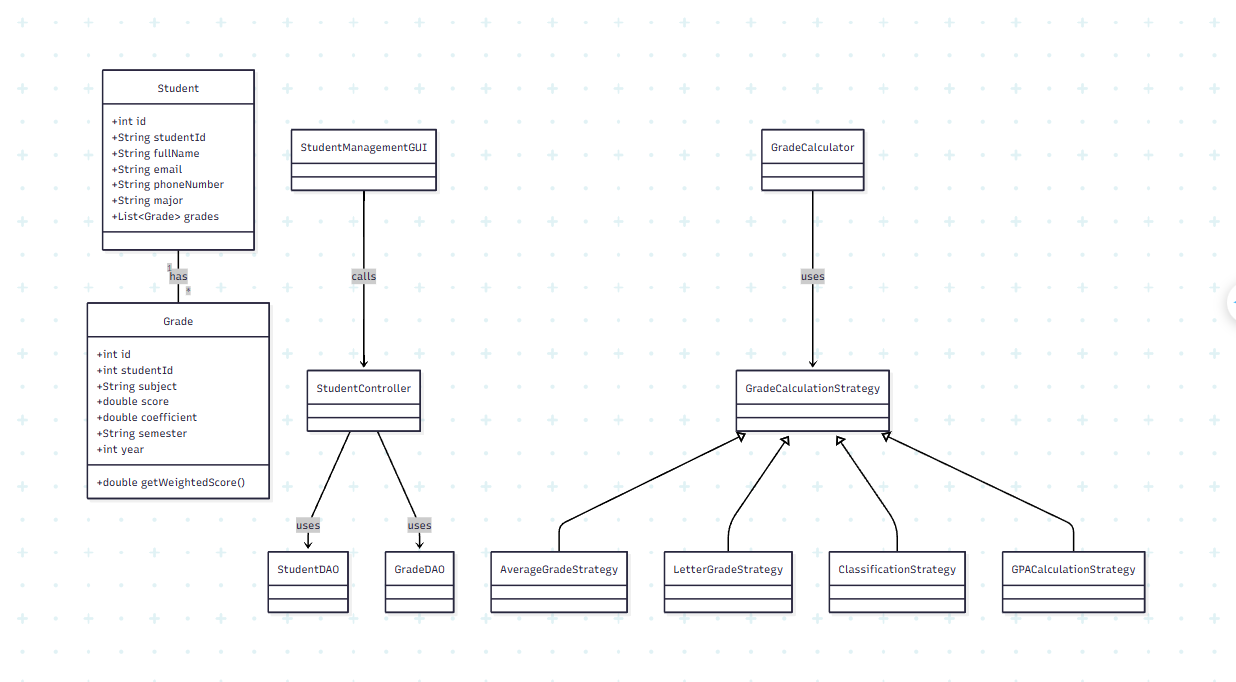


## 5.4 Các trường hợp sử dụng chính (Use Cases)

1. **Quản trị viên thêm sinh viên:** Admin nhập thông tin và lưu sinh viên mới.
2. **Giáo viên thêm điểm cho sinh viên:** Giáo viên chọn sinh viên, nhập thông tin điểm và lưu.
3. **Người dùng xem bảng điểm:** Người dùng chọn sinh viên và xem toàn bộ điểm.
4. **Người dùng tính toán điểm:** Người dùng chọn sinh viên, chọn một chiến lược (Strategy) và xem kết quả (TB, GPA, Xếp loại).
5. **Người dùng xuất báo cáo:** Người dùng chọn sinh viên hoặc toàn khóa và xem báo cáo.

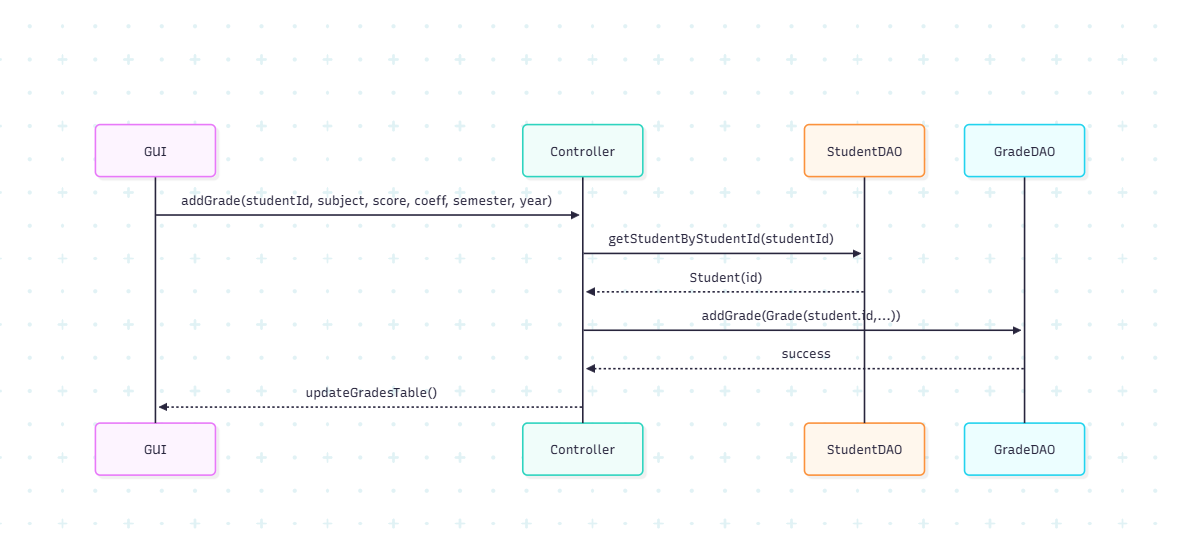
## 5.5 Sơ đồ lớp (Class Diagram)

Sơ đồ lớp minh họa mối quan hệ các lớp chính trong hệ thống:



**5.6 Sơ đồ tuần tự (Sequence Diagram)**

Mô tả luồng nghiệp vụ "Thêm điểm cho sinh viên":



## 5.7 Thiết kế Strategy Pattern cho tính toán điểm

Mục tiêu: tách rời các thuật toán tính toán điểm (trung bình có trọng số, chuyển sang điểm chữ, xếp loại, GPA) vào các lớp độc lập để dễ dàng mở rộng và kiểm thử.

Các strategy chính:

AverageGradeStrategy: tính điểm trung bình có trọng số: $\Sigma(score \times coefficient) / \Sigma(coefficient)$.

LetterGradeStrategy: chuyển đổi điểm số thành điểm chữ (A, B, C, D, F) theo ngưỡng.

ClassificationStrategy: xếp loại học lực (Xuất sắc, Giỏi, Khá, Trung bình, Yếu).

GPACalculationStrategy: chuyển sang thang GPA 4.0.

GradeCalculator (Context) làm nhiệm vụ gọi strategy tương ứng; strategy có thể thay đổi tại runtime.

## 5.8 Các lớp chính (Model / DAO / Controller / GUI)

Student (model): lưu thông tin sinh viên và danh sách Grade.

Grade (model): lưu thông tin môn học, điểm, hệ số, học kỳ, năm.

DatabaseConnection (DAO): khởi tạo kết nối SQLite và tạo bảng khi cần.

StudentDAO, GradeDAO (DAO): thực hiện CRUD cho bảng students và grades.

StudentController (controller): xử lý nghiệp vụ, tổng hợp dữ liệu và gọi các DAO/strategy phù hợp.

StudentManagementGUI (GUI): giao diện Swing, gọi controller để thực hiện thao tác.

## 5.9 Chi tiết kỹ thuật (Xử lý lỗi, Transaction)

**Xử lý lỗi và xác thực:** Controller và GUI thực hiện kiểm tra đầu vào (điểm 0..10, năm hợp lệ, mã SV không trùng).

**Bảo mật:** DAO sử dụng PreparedStatement để tránh SQL injection.

**Transaction:** Khi cần các thao tác phức tạp (ví dụ xóa sinh viên và toàn bộ điểm của sinh viên đó), cần sử dụng transaction (tắt auto-commit, thực thi, commit/rollback) để đảm bảo nhất quán dữ liệu.

# **6. Chức năng chính**

**Quản lý sinh viên:** Thêm, xem, tìm kiếm, cập nhật, xóa.

**Quản lý điểm:** Thêm, xem, cập nhật, xóa điểm (môn học, điểm, hệ số, học kỳ, năm).

**Tính điểm theo Strategy:** tính điểm trung bình, điểm chữ, xếp loại, GPA cho sinh viên.

**Báo cáo:** bảng điểm chi tiết theo sinh viên, báo cáo tổng quan, thống kê phân phối xếp loại.

**Giao diện:** Swing-based, thân thiện, có các bảng và ô nhập liệu rõ ràng.

**Đóng gói:** xuất file JAR và kèm script .bat để chạy nhanh trên Windows.

# **7. Hướng dẫn cài đặt và chạy**

Yêu cầu môi trường

Java JDK/JRE 8 hoặc cao hơn

SQLite JDBC driver (có trong thư mục dist hoặc lib)

Chạy nhanh (Windows)

Mở thư mục dự án và nhấp đúp RunStudentManager.bat.

Hoặc mở PowerShell/CMD và chạy:

cd dist

java -cp "StudentManagement.jar;sqlite-jdbc-3.43.0.0.jar" com.studentmanager.gui.StudentManagementGUI

# **8. Hướng dẫn sử dụng (tóm tắt)**

**Tab "Quản Lý Sinh Viên":** nhập Mã SV, Họ tên (bắt buộc), Email, SĐT, Ngành → Thêm / Cập nhật / Xóa.

**Tab "Quản Lý Điểm":** chọn sinh viên → chọn môn học → nhập Điểm (0–10), Hệ số, Học kỳ, Năm → Thêm / Cập nhật / Xóa.

**Tab "Tính Điểm (Strategy)":** chọn sinh viên và strategy → bấm "Tính Điểm" hoặc "Tính Tất Cả" để so sánh kết quả.

**Tab "Báo Cáo":** chọn sinh viên → bấm "Bảng Điểm" / "Tổng Quan" / "Thống Kê" để hiện báo cáo.

# **9. Kiểm thử và kết quả**

## 9.1 Kết quả kiểm thử thủ công

Các hạng mục kiểm thử đã thực hiện:

**Build:** biên dịch thành công và đóng gói JAR.

**Chức năng CRUD:** kiểm thử thủ công cho thêm / sửa / xóa sinh viên và điểm.

**Tính toán điểm:** so sánh kết quả giữa các strategy, kết quả hợp lý với dữ liệu mẫu.

**Ràng buộc dữ liệu:** điểm giới hạn 0..10; hệ số có thể chọn từ combo box; mã SV không trùng.

**Kiểm tra DB:** dữ liệu lưu trong student\_management.db, có thể mở bằng DB Browser for SQLite.

Kết quả kiểm thử thủ công: **PASS** cho các trường hợp chính; ứng dụng hoạt động ổn định trên Windows (Java 8+).

## 9.2 Kịch bản kiểm thử mẫu (Test Cases)

1. **Thêm sinh viên hợp lệ:**
   * Thao tác: Nhập đầy đủ thông tin SV (mã SV, tên), bấm Thêm.
   * Mong đợi: trả về success, sinh viên xuất hiện trong bảng.
2. **Thêm điểm với điểm không hợp lệ:**
   * Thao tác: Chọn SV, nhập điểm = 11.
   * Mong đợi: Hiển thị thông báo lỗi validate (điểm phải từ 0-10), không cho lưu.
3. **Cập nhật điểm:**
   * Thao tác: Chọn 1 điểm, thay đổi điểm số, bấm Cập nhật.
   * Mong đợi: Dữ liệu DB thay đổi, giao diện cập nhật điểm mới.
4. **Xóa sinh viên:**
   * Thao tác: Chọn 1 sinh viên, bấm Xóa.
   * Mong đợi: Sinh viên và tất cả điểm liên quan bị xóa khỏi DB.
5. **Tính toán bằng tất cả strategy:**
   * Thao tác: Chọn SV có điểm, bấm "Tính Tất Cả".
   * Mong đợi: Kết quả (TB, GPA, Xếp loại, Chữ) hiển thị hợp lý (so sánh tay với tính toán mẫu).

# **10. Hạn chế và đề xuất nâng cấp**

**Hạn chế:**

Chưa có cơ chế phân quyền người dùng (authentication/authorization).

Thiếu test tự động.

Giao diện Swing đơn giản, chưa tối ưu trải nghiệm trên nhiều kích thước màn hình.

Chưa hỗ trợ xuất báo cáo sang PDF/Excel.

Chưa có chức năng import/export hay backup/restore dữ liệu.

**Đề xuất nâng cấp:**

Thêm hệ thống xác thực và phân quyền (admin/teacher/student).

Viết unit test (JUnit) cho Controller, DAO và các Strategy.

Đóng gói fat-jar (gồm driver JDBC) để đơn giản hoá triển khai.

Xem xét chuyển sang web app (Spring Boot + React/Angular) hoặc JavaFX cho GUI hiện đại hơn.

Bổ sung tính năng xuất báo cáo PDF/Excel và import dữ liệu từ CSV/Excel.

Thêm tính năng gửi thông báo (email) cho sinh viên.

# **11. Kết luận**

Dự án "Hệ Thống Quản Lý Điểm Sinh Viên" đã hoàn thành các yêu cầu cốt lõi: quản lý sinh viên, quản lý điểm, tính toán điểm theo nhiều chiến lược, báo cáo tổng quan và đóng gói triển khai. Kiến trúc rõ ràng theo MVC và việc áp dụng Strategy Pattern giúp hệ thống dễ bảo trì và mở rộng. Ứng dụng phù hợp làm đồ án mẫu và có thể tiếp tục phát triển những tính năng nâng cao.

# **12. Phụ lục**

## 12.1 Cấu trúc thư mục chính

## 2212436\_PTPhat\_QLĐSV\_MTK/

## ├── 2212436\_PTPhat\_QLĐSV\_MTK.docx # 📄 Báo cáo chính (Word)

## ├── README.md # 📋 Tài liệu tổng quan

## ├── RunStudentManager.bat # 🚀 Script chạy ứng dụng

## ├── student\_management.db # 💾 Database SQLite

## ├── dist/

## │ ├── StudentManagement.jar # ☕ JAR chính

## │ └── sqlite-jdbc-3.43.0.0.jar # 📚 SQLite driver

## └── src/main/java/com/studentmanager/

## ├── controller/ # 🎛️ Logic nghiệp vụ

## ├── dao/ # 💽 Truy cập dữ liệu

## ├── gui/ # 🖥️ Giao diện Swing

## ├── model/ # 📊 Mô hình dữ liệu

## ├── strategy/ # 🎯 Strategy Pattern

## └── util/ # 🔧 Tiện ích

## 12.2 Các file quan trọng và mô tả ngắn

**StudentManagementGUI.java:** Giao diện chính (Swing), tầng View.

**StudentController.java:** Xử lý nghiệp vụ, tầng Controller.

**StudentDAO.java / GradeDAO.java:** Thao tác cơ sở dữ liệu (CRUD), tầng DAO.

**GradeCalculationStrategy.java (và các implement):** Các chiến lược tính điểm (Strategy Pattern).

**GradeCalculator.java:** Context cho Strategy, dùng để gọi chiến lược.

**Student.java / Grade.java:** Các lớp thực thể, tầng Model.

**DatabaseConnection.java:** Quản lý kết nối đến file SQLite.