# BÀI TẬP THỰC TẬP CƠ SỞ

Viết tài liệu báo cáo thực tập cơ sở:

Mã số sinh viên: K23DTCN463 Họ và tên: Nguyễn Phúc Thanh

Tên Đề tài: Xây dựng ứng dụng quản lý sinh viên bằng Java

Các chức năng (thuật toán) chính:

1. Thêm, sửa, xóa thông tin sinh viên

2. Tìm kiếm và thống kê sinh viên theo khoa, lớp

3. Xuất báo cáo danh sách sinh viên ra file Excel/PDF

## Chương I: Giới thiệu đề tài và công nghệ áp dụng.

**1.1. Giới thiệu đề tài**

**1.1.1. Lý do chọn đề tài**

Trong bối cảnh chuyển đổi số của giáo dục đại học hiện nay, việc quản lý thông tin sinh viên một cách hiệu quả, chính xác và khoa học là một yêu cầu cấp thiết đối với các cơ sở đào tạo. Phương thức quản lý truyền thống bằng văn bản, bảng tính hoặc các hệ thống lạc hậu đã bộc lộ nhiều hạn chế như: tốn kém thời gian, dễ xảy ra sai sót, khó khăn trong việc tra cứu, tổng hợp dữ liệu, và thiếu tính bảo mật.

Vì vậy, việc xây dựng một hệ thống phần mềm quản lý sinh viên hiện đại, tự động hóa các quy trình nghiệp vụ là giải pháp tối ưu. Hệ thống này không chỉ giúp giảm tải công việc cho cán bộ quản lý mà còn nâng cao chất lượng dữ liệu, tạo nền tảng vững chắc cho việc ra quyết định và báo cáo thống kê chính xác. Xuất phát từ nhu cầu thực tiễn đó, em đã lựa chọn đề tài **"Xây dựng ứng dụng quản lý sinh viên bằng Java"** với mong muốn áp dụng các kiến thức đã học vào thực tế, tạo ra một công cụ hữu ích, đóng góp vào công cuộc hiện đại hóa quản lý giáo dục.

**1.1.2. Mục đích và nội dung của ứng dụng**

* **Mục đích**: Xây dựng một phần mềm hoàn chỉnh, giúp đơn giản hóa và tối ưu hóa quy trình quản lý thông tin sinh viên tại trường học.
* **Nội dung**: Phần mềm sẽ tập trung vào các chức năng quản lý thông tin cốt lõi, bao gồm:
  + **Thao tác CRUD**: Cho phép thêm mới, sửa đổi, xóa và xem thông tin chi tiết của sinh viên một cách dễ dàng.
  + **Tìm kiếm và thống kê**: Cung cấp các công cụ tìm kiếm nâng cao theo khoa, lớp, hoặc các tiêu chí khác, cùng với chức năng thống kê số lượng sinh viên theo từng nhóm cụ thể.
  + **Xuất báo cáo**: Hỗ trợ xuất dữ liệu danh sách sinh viên ra các định dạng phổ biến như Excel và PDF để thuận tiện cho việc báo cáo và lưu trữ.

**1.1.3. Kết quả dự kiến đạt được**

Hệ thống sau khi hoàn thành dự kiến sẽ đạt được các kết quả sau:

* Hệ thống có khả năng chạy ổn định trên môi trường **Linux (Ubuntu)**.
* Giao diện được thiết kế đơn giản, trực quan, dễ sử dụng, bao gồm giao diện **console** hoặc **web** tùy theo yêu cầu.
* Toàn bộ dữ liệu sinh viên được lưu trữ một cách an toàn và có cấu trúc trong hệ quản trị cơ sở dữ liệu quan hệ **PostgreSQL**.

**1.2. Lựa chọn công nghệ áp dụng**

Để xây dựng hệ thống quản lý sinh viên, các công nghệ sau đây đã được lựa chọn và áp dụng:

**1.2.1. Ngôn ngữ lập trình: Java**

**Java** là một ngôn ngữ lập trình hướng đối tượng, mạnh mẽ và ổn định. Với nguyên lý "Write Once, Run Anywhere", Java cho phép ứng dụng chạy trên nhiều nền tảng khác nhau mà không cần thay đổi mã nguồn. Hệ sinh thái của Java rất phong phú với nhiều thư viện và framework hỗ trợ, giúp việc phát triển ứng dụng trở nên nhanh chóng và hiệu quả.

**1.2.2. Các Framework và Thư viện**

* **Spring Boot**: Được sử dụng để xây dựng ứng dụng độc lập, cấu hình tự động và giảm thiểu thời gian thiết lập ban đầu. Spring Boot giúp đơn giản hóa việc phát triển ứng dụng Java, cung cấp các tính năng cần thiết như quản lý phụ thuộc và cấu hình máy chủ nhúng.
* **Apache POI**: Một thư viện mã nguồn mở mạnh mẽ, được dùng để thao tác và xử lý các tệp định dạng của Microsoft Office như Word, Excel. Trong đồ án này, Apache POI được sử dụng để xuất dữ liệu sinh viên ra file Excel.
* **JasperReports**: Một công cụ tạo báo cáo linh hoạt, cho phép tạo các báo cáo từ các nguồn dữ liệu khác nhau. JasperReports được tích hợp để xuất báo cáo danh sách sinh viên dưới dạng PDF, phục vụ cho việc in ấn và lưu trữ.

**1.2.3. Hệ quản trị cơ sở dữ liệu: PostgreSQL**

**PostgreSQL** là một hệ quản trị cơ sở dữ liệu quan hệ mã nguồn mở mạnh mẽ, nổi tiếng với sự ổn định, tính toàn vẹn dữ liệu và khả năng mở rộng. Thay vì sử dụng MySQL, PostgreSQL được lựa chọn vì những ưu điểm sau:

* **Tính toàn vẹn dữ liệu**: Hỗ trợ các ràng buộc phức tạp (constraints), transaction và ACID (Atomicity, Consistency, Isolation, Durability) đảm bảo dữ liệu luôn chính xác và nhất quán.
* **Khả năng xử lý dữ liệu phức tạp**: PostgreSQL hỗ trợ nhiều kiểu dữ liệu nâng cao (JSON, XML,...) và các tính năng mạnh mẽ như các hàm cửa sổ (window functions) và CTE (Common Table Expressions), rất phù hợp cho các truy vấn thống kê và báo cáo phức tạp của hệ thống quản lý.
* **Cộng đồng lớn và bảo mật cao**: PostgreSQL có một cộng đồng phát triển và hỗ trợ rộng lớn, cùng với các tính năng bảo mật tiên tiến, đảm bảo sự an toàn cho dữ liệu nhạy cảm của sinh viên.

**1.2.4. Môi trường phát triển**

* **Hệ điều hành**: **Linux (Ubuntu)** được chọn làm môi trường triển khai nhờ vào tính ổn định, bảo mật và chi phí thấp.
* **Phần cứng**: Máy tính cấu hình tối thiểu **Core i3, RAM 4GB** đủ để phát triển và chạy ứng dụng.
* **Mạng**: Cần có kết nối **LAN/WiFi** ổn định để truy cập cơ sở dữ liệu và các tài nguyên mạng khác.

## Chương II: Phân tích hệ thống

**2.1. Khảo sát yêu cầu**

Trong quá trình khảo sát thực tế, hệ thống quản lý sinh viên được yêu cầu phải giải quyết các vấn đề cơ bản sau:

* **Thực trạng hiện tại**:  
  Quá trình quản lý sinh viên chủ yếu dựa vào sổ sách hoặc các file Excel riêng lẻ, gây khó khăn khi cần thống kê hoặc tìm kiếm. Dữ liệu dễ trùng lặp, phân tán và mất nhiều thời gian để tổng hợp.
* **Yêu cầu chức năng**:
  1. Quản lý thông tin sinh viên: thêm mới, chỉnh sửa, xóa dữ liệu.
  2. Quản lý khoa và lớp học: phân loại, thống kê số lượng sinh viên.
  3. Tìm kiếm sinh viên theo tên, mã số, khoa, lớp.
  4. Xuất báo cáo thống kê ra các định dạng phổ biến như Excel, PDF.
  5. Hỗ trợ phân quyền: người dùng thông thường chỉ được xem và tra cứu, admin được quản lý toàn bộ hệ thống.
* **Yêu cầu phi chức năng**:
  1. Hệ thống phải ổn định, phản hồi dưới 2 giây cho các thao tác CRUD.
  2. Dữ liệu được lưu trữ tập trung, bảo mật cao.
  3. Giao diện trực quan, dễ sử dụng cho người không chuyên CNTT.
  4. Có khả năng mở rộng để tích hợp với hệ thống đào tạo, điểm danh, quản lý môn học.

**2.2. Biểu đồ Usecase**

* **Tác nhân**:
  + **Admin**: quản lý toàn bộ hệ thống (CRUD sinh viên, quản lý khoa/lớp, xuất báo cáo).
  + **Người dùng (phòng đào tạo/giảng viên)**: sử dụng chức năng tìm kiếm, thống kê, xuất báo cáo.
* **Các Usecase chính**:
  + Đăng nhập/Đăng xuất hệ thống.
  + Quản lý thông tin sinh viên.
  + Quản lý khoa/lớp.
  + Tìm kiếm và thống kê sinh viên.
  + Xuất báo cáo (Excel/PDF).

**2.3. Kịch bản**

**Kịch bản 1: Thêm sinh viên mới**

* **Tác nhân**: Admin.
* **Điều kiện trước**: Đăng nhập thành công.
* **Luồng sự kiện chính**:
  1. Admin chọn chức năng “Thêm sinh viên”.
  2. Nhập thông tin sinh viên (mã số, họ tên, ngày sinh, khoa, lớp).
  3. Hệ thống kiểm tra dữ liệu nhập.
  4. Nếu hợp lệ → lưu vào CSDL và thông báo thành công.
  5. Nếu không hợp lệ → hiển thị thông báo lỗi.

**Kịch bản 2: Tìm kiếm sinh viên**

* **Tác nhân**: Người dùng.
* **Điều kiện trước**: Truy cập hệ thống.
* **Luồng sự kiện chính**:
  1. Nhập từ khóa (mã số, họ tên, khoa, lớp).
  2. Hệ thống truy vấn dữ liệu.
  3. Hiển thị danh sách kết quả.

**Kịch bản 3: Xuất báo cáo**

* **Tác nhân**: Admin hoặc người dùng.
* **Điều kiện trước**: Có dữ liệu sinh viên trong hệ thống.
* **Luồng sự kiện chính**:
  1. Người dùng chọn chức năng “Xuất báo cáo”.
  2. Hệ thống lấy dữ liệu từ CSDL.
  3. Xuất ra file Excel hoặc PDF.
  4. Thông báo “Xuất thành công”.

**2.4. Biểu đồ tuần tự**

2.4.1. Chức năng **Thêm sinh viên**:

A screenshot of a computer

AI-generated content may be incorrect.

*Chú thích:* Biểu đồ mô tả toàn bộ luồng thêm sinh viên, từ nhập liệu → kiểm tra → lưu cơ sở dữ liệu.

2.4.2. Chức năng **Tìm kiếm sinh viên**:

A diagram of a company

AI-generated content may be incorrect.

*Chú thích:* Biểu đồ tuần tự minh họa quá trình người dùng tìm kiếm thông tin sinh viên, Service và Repository phối hợp truy vấn dữ liệu.

2.4.3. Chức năng **Xuất báo cáo (Excel)**:

A diagram of a computer

AI-generated content may be incorrect.

**2.5. Biểu đồ hành động**

2.5.1. Thêm mới sinh viên:

A diagram of a flowchart

AI-generated content may be incorrect.

*Chú thích:* Biểu đồ hành động cho luồng thêm sinh viên với kiểm tra hợp lệ và xử lý lỗi.

2.5.2 Tìm kiếm sinh viên:

A diagram of a flowchart

AI-generated content may be incorrect.

*Chú thích:* Biểu đồ hành động mô tả các bước tìm kiếm và hiển thị kết quả.

2.5.3 Xuất báo cáo:

A screenshot of a computer

AI-generated content may be incorrect.

**2.6. Lưu đồ (Thuật toán)**

2.6.1. Thuật toán thêm sinh viên:

A diagram of a flowchart

AI-generated content may be incorrect.

*Chú thích:* Lưu đồ thuật toán cho thao tác thêm mới, tập trung vào rẽ nhánh hợp lệ/không hợp lệ.

2.6.2. Thuật toán tìm kiếm sinh viên:

A diagram of a diagram

AI-generated content may be incorrect.

*Chú thích:* Lưu đồ thuật toán thể hiện kiểm tra rỗng và hiển thị kết quả.

2.6.3. Thuật toán xuất báo cáo:

A diagram of a flowchart

AI-generated content may be incorrect.

*Chú thích:* Lưu đồ thuật toán cho xuất báo cáo theo hai định dạng, có xử lý lỗi sinh tệp.

# Chương III: Thiết kế hệ thống

## 3.1. Thiết kế giao diện và kịch bản cho các chức năng

### 1. Thiết kế giao diện

Giao diện được xây dựng theo hướng tối giản, tập trung vào khả năng thao tác nhanh của người dùng. Các màn hình chính gồm:

- Màn hình chính: danh sách sinh viên, thanh tìm kiếm, nút Thêm/Sửa/Xóa/Xuất báo cáo.

- Màn hình thêm/sửa: form nhập Mã SV, Họ tên, Ngày sinh, Giới tính, Lớp, Khoa, kèm kiểm tra hợp lệ.

- Màn hình tìm kiếm: lọc theo từ khóa, khoa, lớp; hỗ trợ phân trang.

- Màn hình báo cáo: chọn phạm vi (khoa/lớp), định dạng (Excel/PDF), xác nhận và tải xuống.

### 2. Kịch bản

Kịch bản tiêu biểu:

• Thêm sinh viên: nhập dữ liệu → kiểm tra hợp lệ → lưu → hiển thị thông báo thành công.

• Sửa thông tin: chọn một sinh viên → cập nhật trường thay đổi → lưu → làm mới danh sách.

• Xóa: chọn một/một nhóm sinh viên → xác nhận → xóa mềm (soft delete) → cập nhật thống kê.

• Tìm kiếm: nhập từ khóa (mã/tên) hoặc lọc theo khoa, lớp → xem kết quả → xuất báo cáo theo kết quả lọc.

• Xuất báo cáo: chọn định dạng → hệ thống dựng dữ liệu → sinh file Excel (Apache POI)/PDF (JasperReports).

## 3.2. Thiết kế các lớp thực thể cho hệ thống

### 1. Thiết kế chi tiết các lớp

 **SinhVien**(maSV, hoTen, ngaySinh, gioiTinh, maLop, maKhoa)

 **Lop**(maLop, tenLop, maKhoa)

 **Khoa**(maKhoa, tenKhoa)

 **TaiKhoan**(username, passwordHash, vaiTro, maSV)

 **BaoCao**(maBC, loai, ngayLap, dinhDang, maSV)

### 2. Biểu đồ lớp (Class Diagram)

A diagram of a computer

AI-generated content may be incorrect.

Hình 3.1: Biểu đồ lớp thực thể của hệ thống quản lý sinh viên.

## 3.3. Thiết kế cơ sở dữ liệu cho hệ thống

A diagram of a computer

AI-generated content may be incorrect.

Hình 3.2: Lược đồ thực thể–quan hệ (ERD) hệ thống quản lý sinh viên.

## 3.4. Biểu đồ tuần tự cho các chức năng

Ví dụ: Biểu đồ tuần tự cho chức năng ThA screenshot of a computer

AI-generated content may be incorrect.êm sinh viên.

Hình 3.3: Biểu đồ tuần tự chức năng thêm sinh viên.

## 3.5. Lưu đồ (Thuật toán)

Thuật toán xuất báo cáo Excel.

A diagram of a flowchart

AI-generated content may be incorrect.

Hình 3.4: Lưu đồ thuật toán xuất báo cáo Excel.

## 3.6. Biểu đồ thành phần cho hệ thống

Các thành phần chính: UI, Service, Repository, Database.

A diagram of a company

AI-generated content may be incorrect.

Hình 3.5: Biểu đồ thành phần của hệ thống.

## 3.7. Biểu đồ triển khai cho hệ thống

Mô hình triển khai theo kiến trúc Client–Server, ứng dụng Spring Boot chạy trên máy chủ ứng dụng, dữ liệu lưu tại PostgreSQL.

A diagram of a server

AI-generated content may be incorrect.

Hình 3.6: Biểu đồ triển khai hệ thống.

# CHƯƠNG IV: CÀI ĐẶT ỨNG DỤNG

## 4.1. Yêu cầu hệ thống

### 4.1.1 Phần cứng:

 CPU: Intel Core i3 hoặc tương đương trở lên

 RAM: 4 GB tối thiểu (khuyến nghị 8 GB)

 Ổ đĩa: Dung lượng trống ≥ 2 GB

 Màn hình: Độ phân giải 1366×768 hoặc cao hơn

### 4.1.2 Phần mềm:

 Hệ điều hành: Ubuntu 22.04 LTS hoặc Windows 10+

 Java Development Kit: **JDK 17**

 Maven: **3.9.x**

 Cơ sở dữ liệu: **PostgreSQL 15**

 IDE phát triển: **IntelliJ IDEA** hoặc **VS Code**

 Trình duyệt: **Chrome / Firefox**

 Tùy chọn triển khai: **Docker Desktop** (nếu chạy container hóa)

### 4.1.3. Hướng dẫn cấu hình cơ sở dữ liệu và chạy ứng dụng

1. Cấu hình cơ sở dữ liệu PostgreSQL:

- Cài đặt PostgreSQL 15 và pgAdmin.

- Tạo cơ sở dữ liệu mới: CREATE DATABASE qlsv\_db;

- Tạo người dùng và phân quyền:

CREATE USER postgres WITH PASSWORD 'admin';

GRANT ALL PRIVILEGES ON DATABASE qlsv\_db TO postgres;

- Mở pgAdmin hoặc DBeaver để xác nhận kết nối hoạt động.

2. Cấu hình kết nối trong Spring Boot (file application.yml):

spring:

datasource:

url: jdbc:postgresql://localhost:5432/qlsv\_db

username: postgres

password: admin

jpa:

hibernate:

ddl-auto: update

show-sql: true

application:

name: student-management

server:

port: 8080

3. Chạy ứng dụng:

cd student-management

mvn spring-boot:run

Sau khi chạy thành công, truy cập địa chỉ: <http://localhost:8080>

**Có thể build thành file jar để chạy độc lập:**  
mvn clean package  
java -jar target/student-management-0.0.1-SNAPSHOT.jar

4. Cấu trúc project sau khi khởi chạy:

student-management/

├── src/

│ ├── main/

│ │ ├── java/com/example/qlsv/

│ │ │ ├── controller/ → Xử lý REST API

│ │ │ ├── service/ → Nghiệp vụ ứng dụng

│ │ │ ├── repository/ → Tầng truy cập dữ liệu (JPA)

│ │ │ └── model/ → Lớp thực thể (Entity)

│ │ └── resources/

│ │ ├── templates/ → Giao diện Thymeleaf (nếu có)

│ │ ├── static/ → CSS, JS, hình ảnh

│ │ └── application.yml → Cấu hình ứng dụng

└── pom.xml → Quản lý dependency Maven

## 4.2. Kết quả cài đặt

### 4.2.1 Trang chủ

Giao diện chính hiển thị danh sách sinh viên, thanh tìm kiếm và thống kê.  
Thanh menu gồm các nút: **Thêm – Sửa – Xóa – Xuất báo cáo**.

### 4.2.2. Chức năng CRUD (Thêm, Sửa, Xóa)

 **Thêm:**  
Người dùng nhập Mã SV, Họ tên, Ngày sinh, Lớp, Khoa → hệ thống kiểm tra hợp lệ → lưu vào DB.

 **Sửa:**  
Chọn sinh viên cần sửa → cập nhật thông tin → hệ thống ghi đè bản ghi cũ.

 **Xóa:**  
Chọn một/một nhóm sinh viên → xác nhận thao tác → hệ thống thực hiện **xóa mềm** (cập nhật cờ trạng thái).

 **Thông báo:**  
Sau mỗi thao tác, hiển thị popup “Thành công” hoặc “Thất bại”.

Ví dụ Service xử lý:

@Service

@RequiredArgsConstructor

public class SinhVienService {

private final SinhVienRepository sinhVienRepository;

public SinhVien save(SinhVien sv) {

return sinhVienRepository.save(sv);

}

public void delete(Long id) {

sinhVienRepository.deleteById(id);

}

public List<SinhVien> findAll() {

return sinhVienRepository.findAll();

}

### }4.2.3. Chức năng xuất báo cáo Excel và PDF

**Xuất Excel:**  
Ứng dụng sử dụng **Apache POI** để ghi dữ liệu ra .xlsx.  
Các cột định dạng rõ: Mã SV, Họ tên, Giới tính, Ngày sinh, Khoa, Lớp.  
Khi nhấn nút Xuất Excel, người dùng chọn nơi lưu, hệ thống tự tạo tệp:

Workbook workbook = new XSSFWorkbook();

Sheet sheet = workbook.createSheet("DanhSachSinhVien");

Row header = sheet.createRow(0);

header.createCell(0).setCellValue("Mã SV");

header.createCell(1).setCellValue("Họ tên");

header.createCell(2).setCellValue("Khoa");

header.createCell(3).setCellValue("Lớp");

 **Xuất PDF:**  
Tích hợp **JasperReports**, sinh tệp PDF từ mẫu report\_template.jrxml.  
Dữ liệu lấy qua truy vấn SQL và truyền vào JRBeanCollectionDataSource.

 **Kết quả thực tế:**  
Các file Excel/PDF xuất ra hiển thị đầy đủ thông tin, định dạng đẹp, dễ đọc, thuận tiện cho việc in ấn và thống kê.

### 4.2.4. Triển khai bằng Docker

Dockerfile:

FROM openjdk:17-jdk-slim

WORKDIR /app

COPY target/student-management.jar app.jar

EXPOSE 8080

ENTRYPOINT ["java","-jar","app.jar"]

**docker-compose.yml:**

version: '3.8'

services:

app:

build: .

ports:

- "8080:8080"

depends\_on:

- db

db:

image: postgres:15

environment:

POSTGRES\_DB: qlsv\_db

POSTGRES\_USER: postgres

POSTGRES\_PASSWORD: admin

ports:

- "5432:5432"