**TRƯỜNG ĐẠI HỌC PHENIKAA**

**KHOA CÔNG NGHỆ THÔNG TIN**

⸎⸎⸎⸎⸎

Ảnh có chứa Phông chữ, biểu tượng, văn bản, Đồ họa

Mô tả được tạo tự động

**ĐỀ BÀI: QUẢN LÝ ĐĂNG KÝ MÔN HỌC CỦA SINH VIÊN**

Nhóm :6

Lớp                  : Lập trình hướng đối tượng-1-1-24(N01)

 Giảng viên hướng dẫn : Nguyễn Lệ Thu

**HÀ NỘI, THÁNG 10/2024**

**TRƯỜNG ĐẠI HỌC PHENIKAA**

**KHOA CÔNG NGHỆ THÔNG TIN**

⸎⸎⸎⸎⸎

Ảnh có chứa Phông chữ, biểu tượng, văn bản, Đồ họa

Mô tả được tạo tự động

**ĐỀ BÀI: QUẢN LÝ ĐĂNG KÝ MÔN HỌC CỦA SINH VIÊN**

|  |  |  |  |
| --- | --- | --- | --- |
| Họ và tên sinh viên | MSSV | Đóng góp | Công việc |
| Trần Thị Lan | 22010135 | 1/3 | Làm phần Student, StudentXML, lưu trữ dữ liệu bằng file, viết báo cáo |
| Bạch Quang Anh | 22010434 | 1/3 | Làm phần Course, CourseXML, User |
| Phạm Thế Minh | 22010075 | 1/3 | Làm phần SectionClass, SectionClassXML, View |

#### ****Link Github Repo:https://github.com/tranlan149/Nhom6****

#### Link video trình bày (Youtube):https://youtu.be/vki3u3GjdJQ?si=mst6LUPwlogENvCb

**MỤC LỤC**

[1. Giới thiệu về bài toán 5](#_Toc17232)

[2. Phân tích bài toán 5](#_Toc20646)

*[2.1. Mô tả hệ thống](#_Toc4868)* [5](#_Toc4868)

*[2.2. Phân tích chức năng](#_Toc24640)* [6](#_Toc24640)

*[2.3. Phân tích cơ sở dữ liệu](#_Toc314)* [8](#_Toc314)

[2.4. Lưu trữ cơ cở dữ liệu(Lưu trữ bằng file XML) 9](#_Toc29485)

[2.4.1.Phương thức writeXMLtoFile(String fileName, Object object): 9](#_Toc4573)

[2.4.2.Phương thức readXMLFile(String fileName, Class<?> clazz): 10](#_Toc16041)

[2.4.3. Phương thức isExistFile(String fileName): 10](#_Toc10850)

*[2.5. Cài đặt và sử dụng](#_Toc10956)* [10](#_Toc10956)

[3. Kết luận: 14](#_Toc20729)

**1. Giới thiệu về bài toán**

Đề tài xây dựng hệ thống “Quản lý đăng ký môn học của sinh viên” bao gồm các nghiệp vụ chính liên quan đến đăng ký môn học với các đối tượng như khóa học, sinh viên, người dùng, lớp học. Hệ thống cho phép sinh viên đăng ký vào các lớp học của các khóa học khác nhau.

Mục tiêu của hệ thống là đảm bảo việc đăng ký môn học được thực hiện một cách hiệu quả và chính xác, đồng thời cung cấp các chức năng quản lý cần thiết cho cả sinh viên và người dùng quản lý.

Cùng các đối tượng và chức năng cơ bản (Quản lý tài khoản, đăng ký sinh viên vào lớp học, thêm sửa xóa các đối tượng) hệ thống quản lý đăng ký môn học sẽ có thể theo dõi và quản lý hiệu quả thông tin về các khóa học, lớp học và sinh viên, hỗ trợ việc đăng ký và quản lý lớp học một cách tiện lợi và khoa học.

**2. Phân tích bài toán**

***2.1. Mô tả hệ thống***

* Các sinh viên đều được gán mã sinh viên để phân biệt với nhau và quản lý các thông tin liên quan.
* Các thông tin lưu trữ liên quan của sinh viên gồm: mã sinh viên (id) để phân biệt giữa các sinh viên với nhau và phân biệt khóa học (id), họ tên sinh viên (name), giới tính (age), địa chỉ thường trú (address), điểm trung bình (gpa).
* Các người dùng khi sử dụng hệ thống đều đăng ký tài khoản gồm các thuộc tính: tên đăng nhập (userName;), mật khẩu (password;).
* Các thông tin lưu trữ liên quan tới khóa học gồm: mã khóa học(id), tên khóa học (courseName), loại khóa học (courseType).
* Các thông tin lưu trữ liên quan tới lớp học của một khóa học gồm: mã lớp học (id), loại lớp học (classType), số lượng sinh viên (maxStudents), thuộc khóa học (course), sinh viên đăng kí (students).

A diagram of a computer program

Description automatically generated

**Hình 2.1. Sơ đồ lớp**

***2.2. Phân tích chức năng***

* Hệ thống “Quản lý đăng ký môn học của sinh viên” được phân rã như sau:
* Quản lý sinh viên
* Lưu môn học đã đăng ký: Sau khi hệ thống kiểm tra các điều kiện thỏa mãn sẽ đăng ký thành công và lưu lại môn học đã đăng ký của sinh viên.
* Danh sách sinh viên đã đăng ký: Thống kê ra thành list danh sách sinh viên đã đăng ký của từng lớp học.
* Quản lý khóa học
* Danh sách khóa học: Thống kê ra thành list danh sách khóa học có trong hệ thống
* Tạo khóa học: Dựa vào thông tin liên quan đến khóa học, tạo loại khóa học cụ thể.
* Cập nhật khóa học: Khi có sự thay đổi cần chỉnh sửa liên quan đến các thuộc tính trong khóa học.
* Quản lý lớp học
* Danh sách lớp học: Thống kê ra thành list danh sách lớp học có trong hệ thống
* Tạo lịch học: Dựa vào thông tin khóa học, tạo lịch học cụ thể để sinh viên có thể sắp xếp thời khóa biểu.
* Cập nhật lớp học: Khi có sự thay đổi cần chỉnh sửa liên quan đến các thuộc tính trong lớp học.
* Xử lý đăng ký
* Kiểm tra: khi sinh viên đăng ký lớp học, hệ thống sẽ kiểm tra các điều kiện đăng ký (số lượng, lịch học)
* Tạo thông báo thất bại/thành công: nếu sinh viên không đủ điều kiện hoặc môn học/lớp học đã hết chỗ thì sinh viên không thể đăng ký học phần.
* Lập phiếu đăng ký: sau khi sinh viên lưu danh sách môn học đã đăng ký, hệ thống sẽ lập phiếu thông tin.

A diagram of a company

Description automatically generated

**Hình 2.2. Sơ đồ phân rã chức năng của hệ thống**

***2.3. Phân tích cơ sở dữ liệu***

* Các bảng thực thể và thuộc tính
* Bảng Course (Khóa học)

|  |  |  |
| --- | --- | --- |
| Tên thực thể | Kiểu dữ liệu | Diễn giải |
| id | int | Mã khóa học |
| courseName | String | Tên khóa học |
| courseType | String | Loại khóa học |

* Bảng User (Người dùng)

|  |  |  |
| --- | --- | --- |
| Tên thực thể | Kiểu dữ liệu | Diễn giải |
| userName | String | Tên người dùng |
| password | String | Mật khẩu |

* Bảng SectionClass (Lớp học)

|  |  |  |
| --- | --- | --- |
| Tên thực thể | Kiểu dữ liệu | Diễn giải |
| id | int | Mã lớp học |
| classType | String | Loai lớp học |
| maxStudents | int | Số lượng sinh viên |
| course | Course | Lớp thuộc khóa học |
| students | List<Student> | Danh sách sinh viên trong lớp |

* Bảng Student (Sinh viên)

|  |  |  |
| --- | --- | --- |
| Tên thực thể | Kiểu dữ liệu | Diễn giải |
| id | int | Mã sinh viên |
| name | String | Tên sinh viên |
| age | byte | Tuổi sinh viên |
| address | String | Địa chỉ |
| gpa | float | Điểm trung bình |

* Mô hình quan hệ dữ liệu
* Lược đồ quan hệ
* Student: id; name; age; address; gpa.
* Course: id; courseName; courseType.
* User: userName; password.
* SectionClass: id; classType; maxStudents; course; students.
* Xây dựng cơ sở dữ liệu
* Bảng khóa học:

public class Course {

private int id;

private String courseName;

private String courseType;

}

* Bảng sinh viên:

public class Student {

private int id;

private String name;

private byte age;

private String address;

private float gpa;

}

* Bảng người dùng:

public class User {

private String userName;

private String password;

}

* Bảng lớp học:

public class SectionClass {

private int id;

private String classType;

private int maxStudents;

private Course course;

private List<Student> students;

}

**2.4. Lưu trữ cơ cở dữ liệu(Lưu trữ bằng file XML)**

**2.4.1.Phương thức writeXMLtoFile(String fileName, Object object):**

* Chuyển đổi một đối tượng Java thành định dạng XML và ghi nó vào một file.
* **Quá trình hoạt động**:
  + Tạo một JAXBContext dựa trên lớp của đối tượng cần chuyển đổi.
  + Tạo đối tượng Marshaller để chuyển đổi đối tượng Java thành XML.
  + Sử dụng Marshaller để ghi XML vào file được chỉ định (fileName).
  + **Định dạng XML**: Được đặt để hiển thị có thụt lề (JAXB\_FORMATTED\_OUTPUT).

**2.4.2.Phương thức readXMLFile(String fileName, Class<?> clazz):**

Đọc dữ liệu từ một file XML và chuyển đổi nội dung đó thành một đối tượng Java.

· **Quá trình hoạt động**:

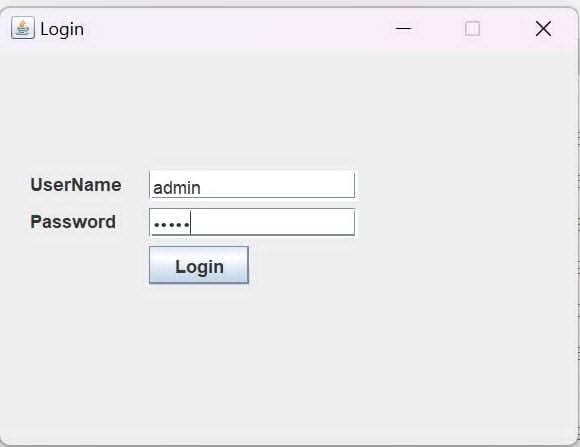
* Mở file XML bằng đối tượng File.
* Tạo JAXBContext dựa trên lớp của đối tượng cần chuyển đổi.
* Tạo đối tượng Unmarshaller để chuyển đổi dữ liệu từ XML thành đối tượng Java.
* Trả về đối tượng đã được chuyển đổi từ file XML.

**2.4.3. Phương thức isExistFile(String fileName):**

* · Kiểm tra xem file có tồn tại hay không.
* **Quá trình hoạt động**:
  + Sử dụng lớp File để kiểm tra sự tồn tại của file thông qua phương thức exists().
  + Trả về true nếu file tồn tại, ngược lại là false.

***2.5. Cài đặt và sử dụng***

* Ngôn ngữ sử dụng: Javaswing
* Giao diện hệ thống
* Giao diện đăng nhập

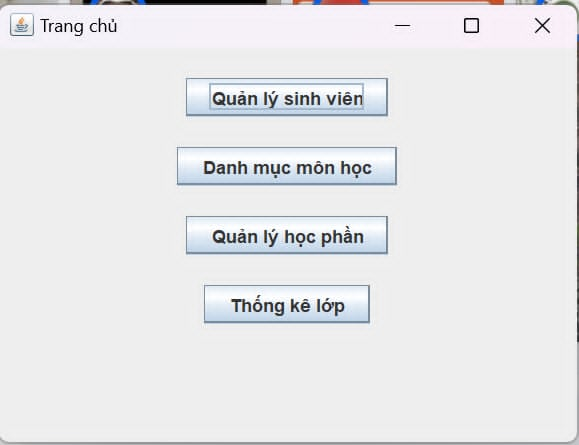


Người dùng khi bắt đầu vào hệ thống cần có tài khoản đăng nhập gồm các thuộc tính (tên đăng nhập , mật khẩu) người dùng click vào ô text để nhập thông tin. Nhập xong người dùng click “Login”, hệ thống kiểm tra thông tin

+ Nếu thông tin chính xác thông báo đăng nhập thành công -> chuyển giao diện “Trang chủ”.

+ Nếu thông tin chưa chính xác hiện thông báo lỗi cho người dùng biết.

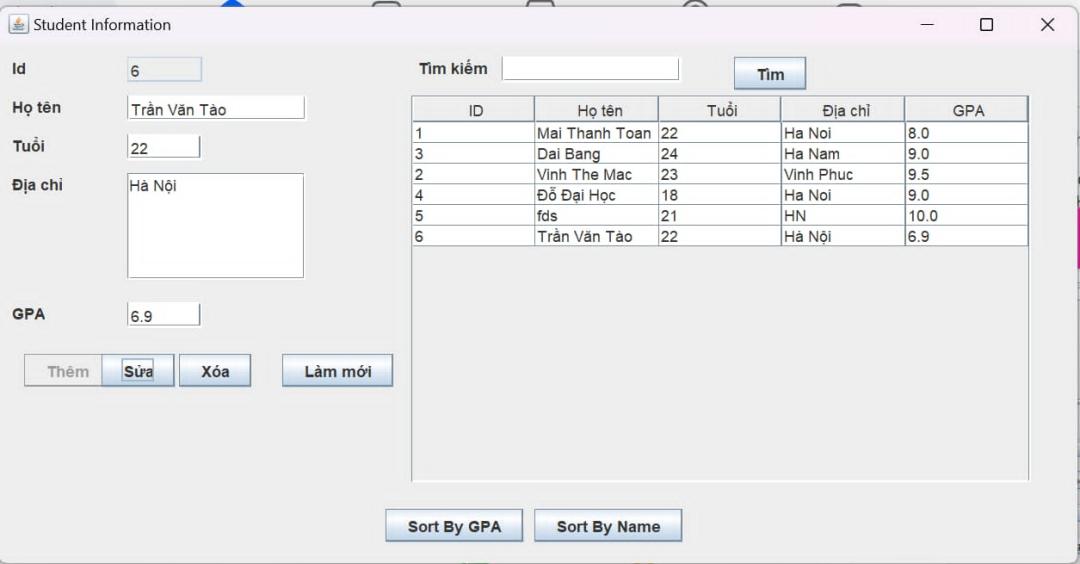
* Giao diện trang chủ



Gồm các chức năng quản lý sinh viên, danh muc môn học, quản lý học phần, thống kê lớp. Khi người dùng click vào một chức năng bất kỳ sẽ chuyển giao diện sang giao diện thuộc chức năng đó.

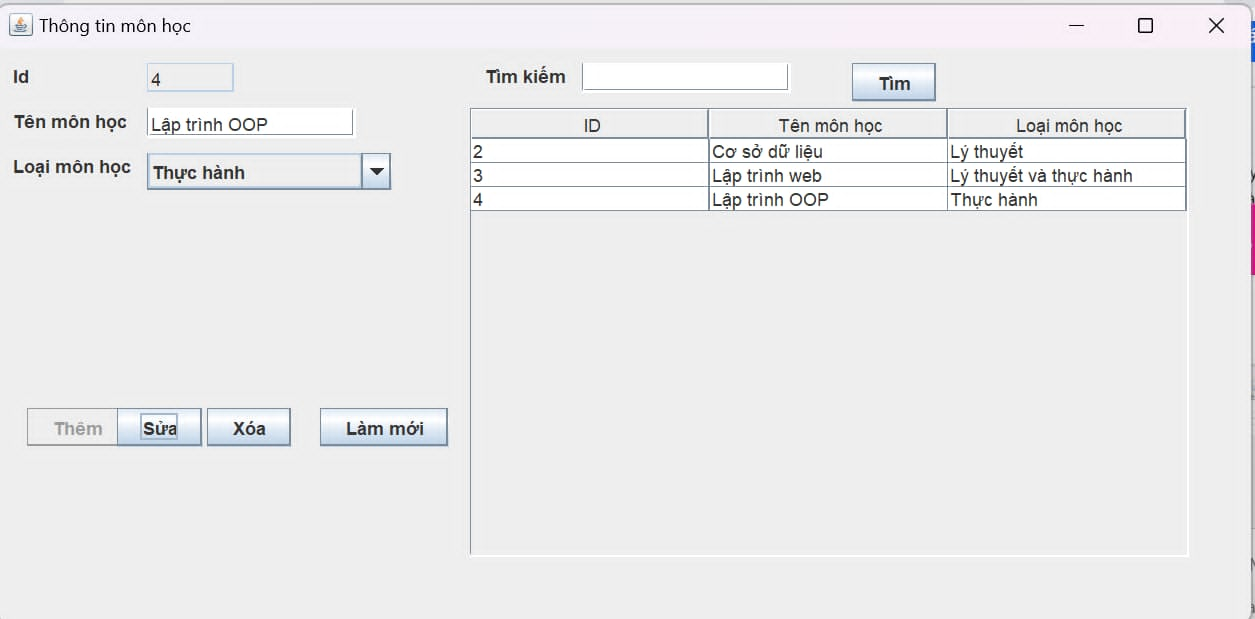
+ Click “Quản lý sinh viên” -> chuyển giao diện sang trang quản lý sinh viên của hệ thống. Chức năng khác tương tự.

* Giao diện quản lý sinh viên



Giao diện gồm các thuộc tính của sinh viên như họ tên, tuổi, địa chỉ,… với các chức năng thêm sinh viên mới, sửa thông tin sinh viên trên hệ thống, xóa thông tin về sinh viên khỏi hệ thống, làm mới các ô text nhập dữ liệu thông tin, hiện danh sách thông tin sinh viên có trong hệ thống và cho phép tìm kiếm sinh viên theo tên sinh viên hoặc id sinh viên.

* Giao diện quản lý môn học



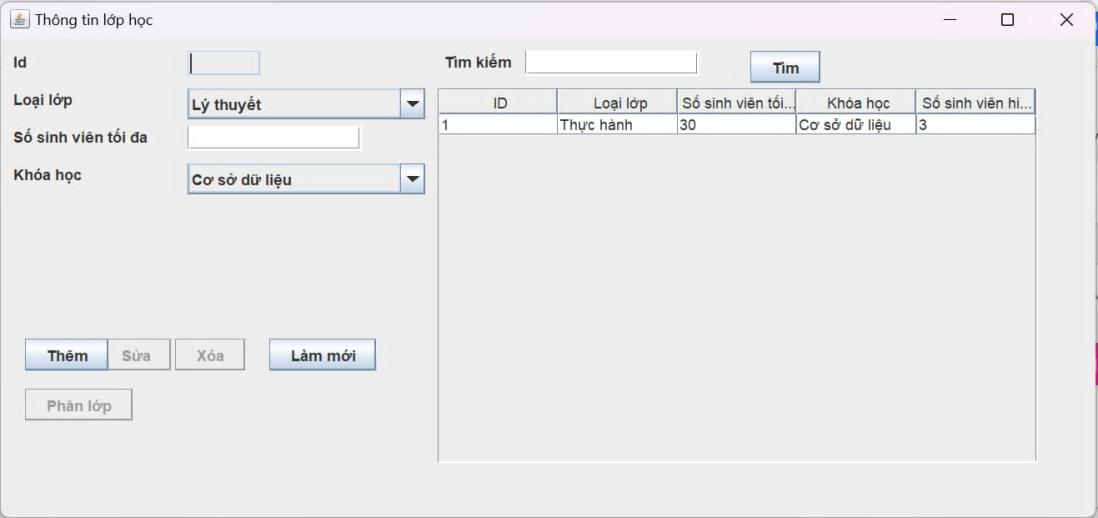
Giao diện gồm các thuộc tính của môn học như tên môn, loại môn học với các chức năng thêm môn học mới, sửa thông tin môn học trên hệ thống, xóa thông tin về môn học khỏi hệ thống, làm mới các ô text nhập dữ liệu thông tin, hiện danh sách thông tin môn học có trong hệ thống và cho phép tìm kiếm môn học theo tên môn học hoặc id môn học.

* Giao diện phân lớp học



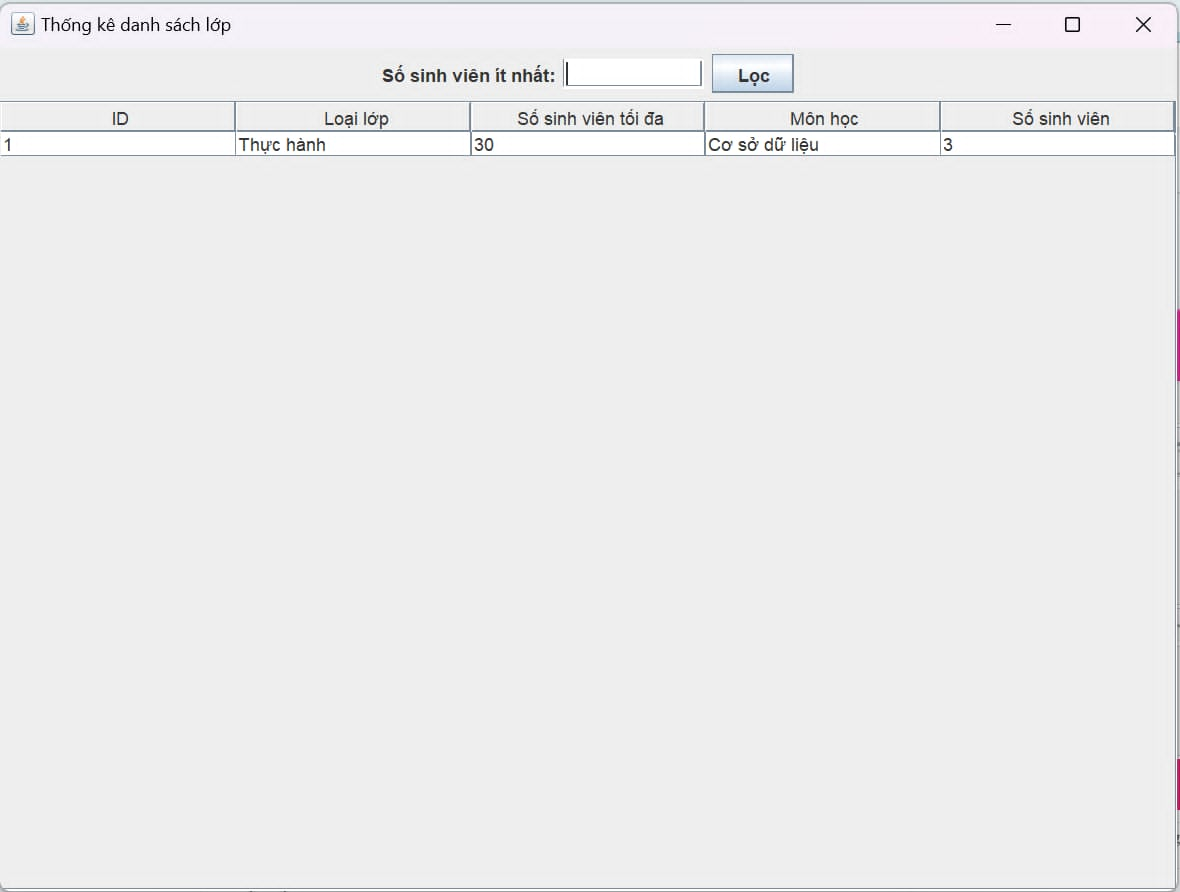
Giao diện gồm hai bảng danh sách sinh viên, người dùng thực hiện phân lớp cho sinh vien bằng cách click vào 1 sinh viên cụ thể -> chọn “<<” hoặc “>>” để đưa thông tin sang bảng danh sách phân lớp. Khi đã thực hiện xong người dùng nhấn “Lưu ” để hệ thống lưu lại danh sách.

* Giao diện thông tin lớp học



Giao diện gồm các thuộc tính của lớp học như loại lớp, số lượng sinh viên, khóa học với các chức năng thêm lớp học mới, sửa thông tin lớp học trên hệ thống, xóa thông tin về lớp học khỏi hệ thống, làm mới các ô text nhập dữ liệu thông tin, phân lớp cho sinh viên, hiện danh sách thông tin lớp học có trong hệ thống và cho phép tìm kiếm lớp học theo loại lớp học hoặc id lớp học.

* Thống kê



Giao diện gồm danh sách thông tin các lớp học và cho phép người dùng lọc lớp học theo số sinh viên ít nhất hiện có trong hệ thống.

1. **Kết luận:**

Trong quá trình phát triển ứng dụng quản lý sinh viên, chúng ta đã đạt được nhiều kết quả tích cực và hoàn thành mục tiêu đề ra ban đầu. Ứng dụng đã giúp quản lý thông tin sinh viên một cách hiệu quả thông qua các chức năng cơ bản như thêm, sửa, xóa và tìm kiếm sinh viên. Các thông tin về sinh viên như họ tên, ID, điểm số đã được tổ chức và lưu trữ hợp lý, cho phép dễ dàng tra cứu và quản lý dữ liệu.

#### Tóm tắt kết quả đạt được:

* **Giao diện thân thiện**: Ứng dụng cung cấp giao diện đơn giản, dễ sử dụng giúp người dùng có thể thao tác một cách nhanh chóng và chính xác.
* **Chức năng quản lý hiệu quả**: Các chức năng chính như thêm sinh viên, cập nhật thông tin, xóa và tìm kiếm đều hoạt động ổn định. Người dùng có thể quản lý thông tin của nhiều sinh viên một cách hiệu quả.
* **Lưu trữ dữ liệu bằng XML**: Chức năng lưu trữ dữ liệu sinh viên dưới dạng file XML giúp việc quản lý dữ liệu được an toàn, linh hoạt và dễ dàng chuyển đổi, trao đổi dữ liệu.
* **Kiểm tra tính hợp lệ của file**: Ứng dụng đã bổ sung chức năng kiểm tra xem file XML có tồn tại hay không trước khi thực hiện các thao tác, giúp giảm thiểu rủi ro xảy ra lỗi khi xử lý dữ liệu.

#### Tổng kết về mục tiêu ban đầu và cách hoàn thành:

Mục tiêu ban đầu của dự án là xây dựng một ứng dụng quản lý sinh viên hỗ trợ các thao tác cơ bản như lưu trữ, cập nhật và tìm kiếm thông tin sinh viên một cách dễ dàng và tiện lợi. Đã hoàn thành mục tiêu này thông qua:

* **Thiết kế hệ thống**: Ứng dụng được xây dựng dựa trên mô hình hướng đối tượng, giúp việc mở rộng và bảo trì dễ dàng hơn.
* **Sử dụng JAXB cho xử lý XML**: Đã thành công trong việc triển khai việc lưu trữ và đọc dữ liệu từ file XML, đảm bảo tính ổn định và khả năng tương thích cao.
* **Thử nghiệm và kiểm tra**: Ứng dụng đã được thử nghiệm với nhiều trường hợp sử dụng khác nhau và hoạt động ổn định, đáp ứng được yêu cầu đặt ra.