**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

=========\*\*\*=========

****

**BÁO CÁO THÍ NGHIỆM/THỰC NGHIỆM LẬP TRÌNH JAVA**

**XÂY DỰNG PHẦN MỀM QUẢN LÝ SINH VIÊN**

**BẰNG JAVA**

| GVHD: | TS. Hà Mạnh Đào |
| --- | --- |
| Sinh viên: | Nguyễn Phi Hùng |
| Nhóm: | 1 |
| Lớp: | IT6001 |
| Khóa: | K18 |
|  |  |

Hà Nội, Năm 2024

MỤC LỤC

[**CHƯƠNG 1: MỞ ĐẦU** 2](#_Toc185464082)

[**1.1. Giới thiệu về Java** 2](#_Toc185464083)

[**1.2. Giới thiệu về Java Swing** 2](#_Toc185464084)

[**1.3. Tổng quan về mô hình MVC** 3](#_Toc185464085)

[**1.4. Tổng quan về bài toán quản lý sinh viên** 4](#_Toc185464086)

[**CHƯƠNG 2: KẾT QUẢ NGHIÊN CỨU** 7](#_Toc185464087)

[**2.1. Quá trình xây dựng** 7](#_Toc185464088)

[**2.2. Khảo sát hệ thống** 8](#_Toc185464089)

[**2.2.1. Khảo sát sơ bộ** 8](#_Toc185464090)

[**2.2.2. Khảo sát chi tiết** 9](#_Toc185464091)

[**2.3. Phân tích hệ thống** 10](#_Toc185464092)

[**2.3.1. Mô hình hóa chức năng của hệ thống** 10](#_Toc185464093)

[**2.4. Thiết kế hệ thống** 17](#_Toc185464094)

[**2.4.1. Thiết kế dữ liệu** 17](#_Toc185464095)

[**2.4.2. Thiết kế giao diện** 18](#_Toc185464096)

[**2.4.3. Kiểm tra thử nghiệm** 25](#_Toc185464097)

[**CHƯƠNG 3: TỔNG KẾT** 26](#_Toc185464098)

[**3.1. Nội dung đã thực hiện** 26](#_Toc185464099)

[**3.2. Kết quả đạt được** 27](#_Toc185464100)

[**3.4. Tính khả thi** 28](#_Toc185464101)

[**3.5.Tính Mở Rộng** 29](#_Toc185464102)

[**3.6.Kết Luận** 30](#_Toc185464103)

[**TÀI LIỆU THAM KHẢO** 31](#_Toc185464104)

# **CHƯƠNG 1: MỞ ĐẦU**

## **1.1. Giới thiệu về Java**

Java là một ngôn ngữ lập trình hướng đối tượng, được phát triển bởi Sun Microsystems vào năm 1995 và hiện nay thuộc sở hữu của Oracle Corporation. Java nổi bật với khả năng chạy trên nhiều nền tảng nhờ vào cơ chế biên dịch trung gian gọi là bytecode, cho phép chương trình Java có thể chạy trên bất kỳ hệ điều hành nào có cài đặt Java Virtual Machine (JVM).

Java được biết đến với nhiều ưu điểm vượt trội như tính đa nền tảng, bảo mật cao, và hỗ trợ lập trình hướng đối tượng. Ngoài ra, Java còn cung cấp một hệ thống thư viện phong phú hỗ trợ nhiều lĩnh vực từ phát triển ứng dụng web, di động đến các ứng dụng doanh nghiệp và trí tuệ nhân tạo.

## **1.2. Giới thiệu về Java Swing**

Java Swing là một phần của Java Foundation Classes (JFC) - tập hợp các thư viện Java dùng để tạo giao diện đồ họa người dùng (GUI). Swing được xây dựng dựa trên AWT (Abstract Window Toolkit), nhưng cung cấp nhiều tính năng mạnh mẽ hơn và khả năng tùy biến cao hơn. Swing cho phép lập trình viên tạo ra các ứng dụng với giao diện phong phú và đồng nhất trên mọi nền tảng nhờ các thành phần GUI độc lập với hệ điều hành.

Swing cung cấp nhiều thành phần giao diện như nút, hộp thoại, bảng, và cây, giúp lập trình viên dễ dàng tạo ra các ứng dụng GUI đa dạng và phức tạp. Ngoài ra, Swing còn cho phép tùy chỉnh giao diện của các thành phần thông qua các lớp vẽ tùy chỉnh và các trình renderer, tạo ra giao diện người dùng độc đáo và thân thiện. Swing cũng tuân thủ mô hình Model-View-Controller (MVC), giúp tách biệt logic xử lý, dữ liệu và giao diện, tăng tính bảo trì và mở rộng của ứng dụng.

## **1.3. Tổng quan về mô hình MVC**

Mô hình MVC là viết tắt của Model-View-Controller, là một kiến trúc phần mềm phổ biến được sử dụng để tổ chức mã nguồn trong các ứng dụng phần mềm. Mô hình này giúp tách biệt các thành phần của hệ thống để dễ quản lý và bảo trì.

1. Model

- Đại diện cho dữ liệu và logic xử lý dữ liệu.

- Cập nhật trạng thái của ứng dụng khi dữ liệu thay đổi.

- Không có thông tin về cách dữ liệu được hiển thị hoặc tương tác với người dùng.

b. View

- Hiển thị thông tin cho người dùng.

- Được cập nhật khi trạng thái của Model thay đổi.

- Không chứa logic xử lý dữ liệu.

c. Controller

- Nhận đầu vào từ người dùng thông qua giao diện.

- Chịu trách nhiệm xử lý sự kiện và tương tác với Model.

- Cập nhật trạng thái của Model và thông báo cho View về các thay đổi.

A diagram of a system

Description automatically generated

Hình 1.1 Mô hình tổng quan MVC

Ưu điểm của mô hình MVC:

- Tách biệt giữa các thành phần: Việc tách biệt giữa Model, View và Controller giúp dễ dàng quản lý và bảo trì mã nguồn. Mỗi thành phần có nhiệm vụ riêng biệt, giảm sự phụ thuộc lẫn nhau.

- Tái sử dụng mã nguồn: Với mô hình MVC, bạn có thể tái sử dụng các thành phần của ứng dụng một cách dễ dàng. Ví dụ, bạn có thể thay đổi giao diện người dùng (View) mà không cần phải thay đổi logic xử lý dữ liệu (Model).

- Tăng cường tính linh hoạt và mở rộng: Mô hình MVC giúp ứng dụng dễ dàng mở rộng và phát triển. Bạn có thể thêm các tính năng mới mà không làm ảnh hưởng đến toàn bộ hệ thống.

- Kiểm thử dễ dàng: Từng thành phần trong mô hình MVC có thể được kiểm thử một cách độc lập, giúp việc phát hiện và sửa lỗi trở nên dễ dàng hơn.

Mô hình MVC là một kiến trúc phần mềm mạnh mẽ và linh hoạt, giúp tổ chức mã nguồn một cách rõ ràng và hiệu quả. Bằng cách tách biệt giữa Model, View và Controller, MVC không chỉ giúp ứng dụng dễ dàng bảo trì và mở rộng mà còn tăng cường khả năng tái sử dụng mã nguồn và hỗ trợ quá trình kiểm thử. Việc áp dụng mô hình MVC trong phát triển phần mềm là một lựa chọn thông minh để đảm bảo chất lượng và hiệu suất của ứng dụng.

## **1.4. Tổng quan về bài toán quản lý sinh viên**

a) Bối cảnh

Ngành giáo dục của Việt Nam là ngành đi đầu trong việc trồng người. Với dân số ngày càng tăng trưởng cao thì việc học sinh, sinh viên ngày càng nhiều là điều dễ hiểu. Nhưng để quản lý học sinh, sinh viên một cách hiệu quả là rất khó. Việc làm việc thủ công là rất lâu và mất thời gian.

b) Đối tượng khách hàng

Đối tượng khách hàng của phần mềm quản lý sinh viên bao gồm các nhà trường vừa và nhỏ. Đây là những trường chưa áp dụng nhiều công nghệ vào việc quản lý cũng như có thể chưa tiếp cận nhiều về công nghệ thông tin.

c) Các vấn đề và thách thức

▪ Quản lý sinh viên thủ công: Hiện nay, nhiều nhà trường vẫn còn quản lý các sinh viên của mình một cách thủ công, dẫn đến mất thười gian và có thể gây ra nhiều sai sót trong quá trình kiểm tra và cập nhật thông tin sinh viên.

▪ Khó khăn trong việc theo dõi và kiểm soát sinh viên về việc sinh viên còn đi học hay không và học tập như nào.

d) Xác định và nêu lên vấn đề cần giải quyết

Vấn đề cần giải quyết trong đề tài "Thiết kế phần mềm quản lý sinh viên" là cải thiện quá trình quản lý sinh viên tại các trường học nhỏ và vừa tại Việt Nam. Hiện nay, các trường vừa và nhỏ gặp nhiều khó khan và hạn chế trong việc quản lý sinh viên gây ảnh hưởng tới hiệu quả và sự phát triển của nhà trường.

e) Yêu cầu chung đối với vấn đề cần giải quyết

- Các mục lớn cần xử lý trong vấn đề đã xác định :

▪ Quản lý thông tin sinh viên: phần mềm cần hỗ trợ quá trình thu thập thông tin của sinh viên và lưu trữ lại thông tin của từng sinh viên để tiện sau có thể tra khảo lại một cách dễ dàng.

▪ Quản lý điểm: phần mềm cần hỗ trợ trong việc quản lý điểm của từng sinh viên để đánh giá và xếp loại sinh viên.

- Giới hạn phạm vi :

Phần mềm được thiết kế dành cho các nhà trường nhỏ và vừa tại Việt Nam.

- Đối tượng sử dụng :

Giáo viên, giảng viên đang theo dạy ở các trường, trường đại học nhỏ và vừa ở Việt Nam.

f) Yêu cầu cụ thể

Qua kết quả khảo sát yêu cầu khách hàng của một số các nhà quản lý đã cho chúngem cái nhìn rõ ràng về những vấn đề mà khách hàng cần giải quyết trong quá trình quản lý sinh viên. Dưới đây là các yêu cầu cụ thể mà phần mềm quản lýsinh viên cần đáp ứng:

- Quản lý thông tin sinh viên

▪ Theo dõi số lượng sinh viên còn theo học tại nhà trường.

▪ Cập nhật sơ yếu lý lịch của sinh viên.

▪ Cung cấp chức năng tìm kiếm nhanh để tìm kiếm thông tin về học ính, sinh viên đang học tập tại nhà trường.

- Quản lý điểm

Quản lý điểm của từng sinh viên đang theo học tại nhà trường.

- Đối tượng dữ liệu quản lý

▪ Danh sách học sinh, sinh viên học tập tại nhà trường.

▪ Thông tin về học sinh, sinh viên về điểm, môn học, sơ yếu lý lịch

# **CHƯƠNG 2: KẾT QUẢ NGHIÊN CỨU**

## **2.1. Quá trình xây dựng**

Đề tài xây dựng ứng dụng Swing quản lý sinh viên nhằm giải quyết các vấn đề liên quan đến việc Quản lý thông tin sinh viên một cách hiệu quả. Ứng dụng có thể giúp quản lý thông tin cơ bản của sinh viên như tên, mã sinh viên, mã lớp, điểm và trạng thái môn học.

Quá trình xây dựng sản phẩm bao gồm các bước sau:

1. Phát hiện và phân tích yêu cầu :

* Phân tích và mô hình hóa các yêu cầu, lập danh sách các chức năng chính cần có.

1. Thiết kế kiến trúc hệ thống :

* Lựa chọn kiến trúc và công nghệ phù hợp.
* Thiết kế các thành phần chính của hệ thống.
* Xây dựng sơ đồ luồng dữ liệu và các use case chính.

1. Xây dựng và tích hợp các module :

Sắp xếp và chia nhỏ công việc.

* Lập trình và phát triển từng module/chức năng.
* Tích hợp các module lại với nhau, kiểm thử từng chức năng.

1. Kiểm thử toàn diện :

* Kiểm thử chức năng, hiệu năng, tính ổn định của hệ thống.
* Sửa lỗi và hoàn thiện sản phẩm.

1. Triển khai và bàn giao :

* Cài đặt sản phẩm trên môi trường thực tế.
* Tiếp nhận phản hồi và cải tiến sản phẩm.

Hiểu được vấn đề trên, nhóm đã tiến hành xây dựng hệ thống quản lý sinh viên, hỗ trợ việc quản lý một cách nhanh chóng, dễ dàng và chính xác.

Mục tiêu: Thiết kế giao diện thân thiện với người dùng, dễ dàng sử dụng, xây dựng đầy đủ chức năng cần thiết trong phạm vi cho phép, ít xảy ra lỗi nhất.

## **2.2. Khảo sát hệ thống**

### **2.2.1. Khảo sát sơ bộ**

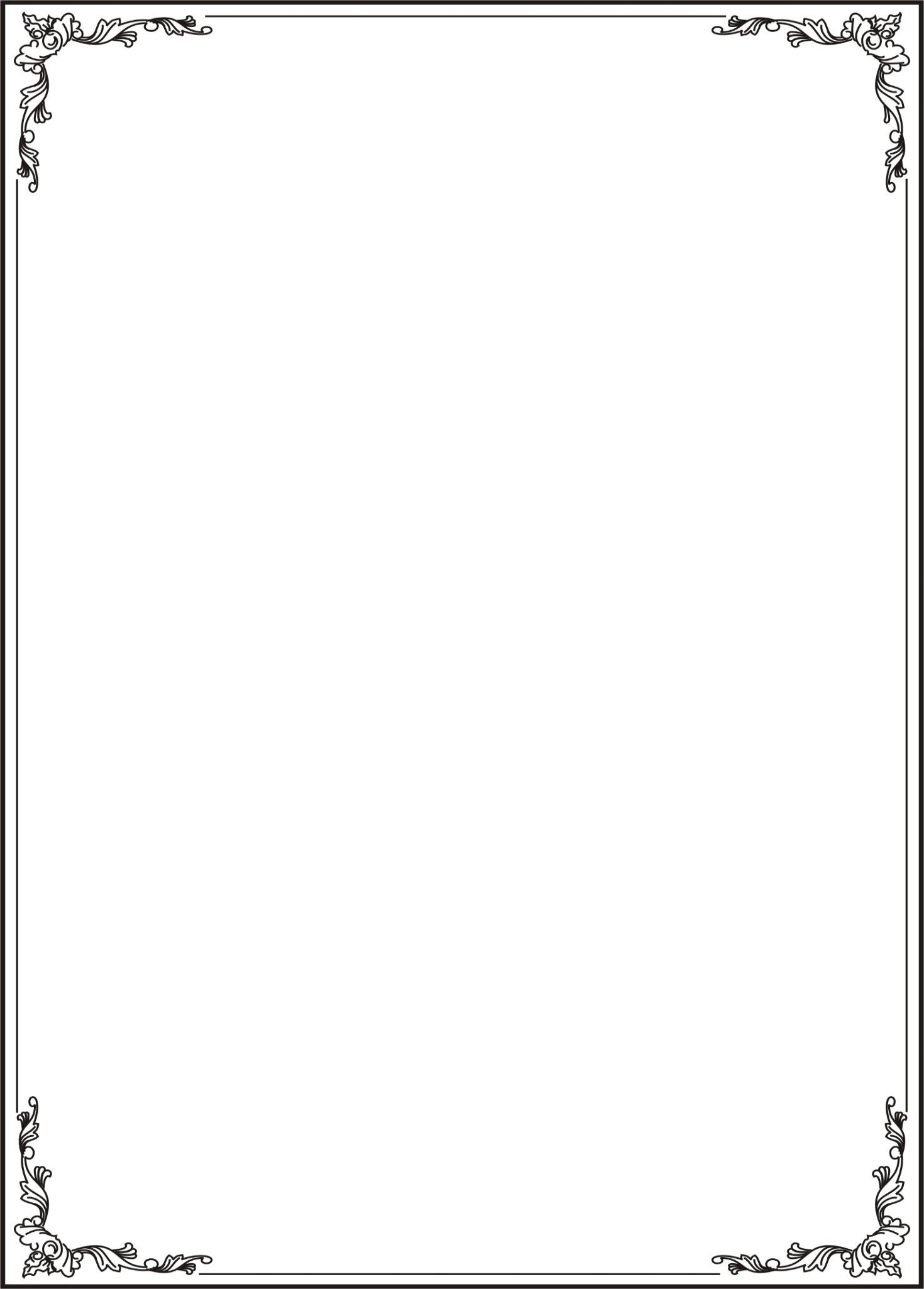
1. *Mục tiêu*

Điều tra, khảo sát về hệ thống quản lý sinh viên. Tìm hiểu cách quản lý thông tin sinh viên, trạng thái điểm môn học…

*b) Phương pháp*

**-** Phỏng vấn.

|  |  |
| --- | --- |
| Kế hoạch phỏng vấn | |
| Người phỏng vấn: Phạm Văn Bách | Người được hỏi: Trịnh Thị Trinh |
| Thời gian hẹn: 8h40  Thời gian bắt đầu: 9h05  Thời gian kết thúc: 10h45 | Địa chỉ: Tòa A1 Trường đại học Công Nghiệp Hà Nội, Hà Nội. |
| Yêu cầu:  Vị trí: Người quản lý  Trình độ: Hiểu rõ về quản lý sinh viên, thông tin sinh viên. | Đối tượng:  Vị trí: Quản lý sinh viên khoa CNTT  Cần thu thập dữ liệu: Phương thức quản lý sinh viên và cung cấp thông tin cho sinh viên. |



*c) Kết quả sơ bộ.*

* Điểm mạnh và điểm yếu của hệ thống đang vận hành
  + Điểm mạnh: Hệ thống hoạt động ổn định.
  + Điểm yếu: Hệ thống chưa tối ưu các chức năng chính ,phương thức hoạt động rườm rà, đôi lúc bị sai sót, tính bảo mật chưa cao.

### **2.2.2. Khảo sát chi tiết**

Để thực hiện khảo sát chi tiết cho việc thiết kế hệ thống quản lý sinh viên bằng Java, nhóm thực hiện các bước sau:

1. Xác định đối tượng khảo sát:

* Người dùng chính của hệ thống sẽ là phòng quản lý sinh viên, các bên liên quan khác như giáo viên, các phòng ban liên quan..
* Cần xác định số lượng đại diện cần được phỏng vấn/khảo sát từ mỗi nhóm người dùng này.

1. Thu thập thông tin:

Xây dựng bộ câu hỏi khảo sát chi tiết bao gồm:

* Các chức năng hiện tại và mong muốn của hệ thống quản lý sinh viên (quản lý thông tin sinh viên, điểm, trạng thái môn học)
* Các vấn đề, thách thức trong quản lý sinh viên hiện tại
* Lựa chọn phương pháp thu thập thông tin như phỏng vấn trực tiếp, khảo sát online, quan sát quy trình, etc.

1. Phân tích và tổng hợp thông tin:

* Tổng hợp, phân loại và phân tích dữ liệu thu thập được.
* Xác định các chức năng chính, quy trình quan trọng và các yêu cầu ưu tiên của người dùng.
* Xác định các yêu cầu chức năng và phi chức năng chính cho hệ thống mới, bao gồm:
* Quản lý thông tin sinh viên
* Cập nhật thông tin sinh viên
* Tính toán điểm và trạng thái môn học
* Quản lý tài khoản

1. Báo cáo và trình bày kết quả:

* Đảm bảo các yêu cầu được ghi lại đầy đủ và rõ ràng để làm cơ sở cho giai đoạn thiết kế và phát triển.

## **2.3. Phân tích hệ thống**

### **2.3.1. Mô hình hóa chức năng của hệ thống**

#### **2.3.1.1. Biểu đồ use case**

|  |  |
| --- | --- |
| Tên tác nhân | Mô tả ngắn gọn tác nhân |
| Quản trị viên | Người quản trị hệ thống , sinh viên , tài khoản cấp. |

|  |  |  |
| --- | --- | --- |
| ID | Tên use case | Mô tả ngắn gọn use case |
| UC\_01 | Đăng ký | Cho phép quản trị viên đăng ký tài khoản |
| UC\_02 | Đăng nhập | Cho phép quản trị viên đăng nhập khi đã có tài khoản |
| UC\_03 | Quản lý điểm sinh viên | Quản lí số lượng, thông tin sinh viên, thực hiện các hành động thêm, sửa, xóa, tìm kiếm |

A diagram of a diagram

Description automatically generated

*Hình 2.1 Biểu đồ use case chính*

**2.3.1.2. Mô tả chi tiết use case**

**2.3.1.2.1. Mô tả use case “Đăng nhập”**

1. Tên Use case: Đăng nhập

2. Mô tả vắn tắt: Use case này cho phép người quản trị đăng nhập vào hệ thống .

3. Luồng các sự kiện

3.1. Luồng cơ bản

1. Use case này bắt đầu khi quản trị viên nhập tài khoản vào các trường dữ liệu và kích vào nút “Login” trên giao diện đăng nhập.
2. Hệ thống sẽ kiểm tra tài khoản nhập vào với danh sách tài khoản trong file dữ liệu “Account.txt” và đối chiếu.
3. Hệ thống kiểm tra quyền đăng nhập của tài khoản. Hệ thống sẽ hiển thị giao diện quản trị.

Use case kết thúc.

3.2. Các luồng rẽ nhánh

1. Tại bước 1 trong luồng cơ bản, nếu người dùng không nhập thông tin hoặc nhập thông tin tài khoản, mật khẩu không hợp lệ, hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.
2. Tại bước 2 nếu không có dữ liệu trong file dữ liệu hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.
3. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với file dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và Use case kết thúc.

4. Các yêu cầu đặc biệt: Không có.

5. Tiền điều kiện: Quản trị viên phải có tài khoản đã được đăng ký.

6. Hậu điều kiện: Khi use case kết thúc thành công, hiện thông báo đăng nhập thành công và hiển thị giao diện cho quản trị viên

7. Điểm mở rộng: Không có.

**2.3.1.2.2. Mô tả use case “Đăng ký”**

1. Tên Use case: Đăng ký

2. Mô tả vắn tắt: Use case này cho phép quản trị viên đăng ký tài khoản hệ thống.

3. Luồng các sự kiện

3.1. Luồng cơ bản

1. Use case này bắt đầu khi quản trị viên nhập các thông tin và kích nút “Create” trên giao diện Đăng ký.
2. Hệ thống sẽ lưu thông tin tài khoản vào file dữ liệu “Account.txt”. Use case kết thúc

3.2. Các luồng rẽ nhánh

1. Tại bước 1 luồng cơ bản khi người dùng không nhập đầy đủ các trường hệ thống thông báo lỗi và use case kết thúc.
2. Tại bước 2 luồng cơ bản, hệ thống sẽ kiểm tra tên tài khoản đã tồn tại trong file dữ liệu “Account.txt” chưa, nếu đã tồn tại thông báo lỗi sẽ xuất hiện, nếu chưa thông tin tài khoản đăng ký sẽ lưu vào file “Account.txt” hệ thống sẽ chuyển hướng tới giao diện đăng nhập.
3. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với file dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

4. Các yêu cầu đặc biệt: Không có.

5. Tiền điều kiện: Tài khoản muốn đăng ký chưa được sử dụng trước đó.

6. Hậu điều kiện: Sau khi đăng ký thành công tài khoản được lưu vào file dữ liệu và giao diện chuyển hướng sang giao diện Đăng nhập

7. Điểm mở rộng: Không có

**2.3.1.2.3 Mô tả use case “Quản lý** **điểm sinh viên”**

1. Tên Use case: Quản lý điểm sinh viên.

2. Mô tả vắn tắt: Use case này cho phép người quản trị thực hiện các chức năng thêm, sửa, xóa, tìm kiếm thông tin các sinh viên và xuất danh sách thông tin các sinh viên trong hệ thống.

3. Luồng các sự kiện

3.1. Luồng cơ bản

1. Use case này bắt đầu khi người quản trị đăng nhập thành công vào hệ thống .Hệ thống hiển thị cửa sổ quản lý sinh viên đồng thời lấy dữ liệu từ file dữ liệu “data\_Students.txt” và hiển thị danh sách toàn bộ sinh viên lên bảng danh sách sinh viên trên màn hình.
2. Thêm sinh viên mới:
   1. Người quản trị kích vào nút “Thêm” trên giao diện.
   2. Người quản trị nhập các thông tin sinh viên mới bao gồm: Tên sinh viên, mã sinh viên, mã lớp, điểm lý thuyết, điểm thực hành.
   3. Người quản trị click vào nút “Lưu”. Hệ thống sẽ tạo một bản ghi mới gồm các thông tin Tên sinh viên, mã sinh viên, mã lớp, điểm lý thuyết, điểm thực hành và ghi vào file “data\_Students.txt”
3. Sửa thông tin sinh viên:
   1. Người quản trị kích chọn nút “Sửa” trên giao diện
   2. Người quản trị nhập Tên sinh viên,mã lớp, điểm lý thuyết, điểm thực hành mới.
   3. Sau đó kích nút “Đồng ý”. Hệ thống sẽ tự động sửa thông tin của sinh viên trên bảng danh sách sinh viên trên màn hình. Hệ thống sẽ cập nhật dữ liệu vào file dữ liệu “data\_Students.txt”.
4. Xóa thông tin sinh viên:
   1. Người quản trị kích chọn nút “Xóa” trên giao diện.
   2. Người quản nhập mã sinh viên của sinh viên cần xóa.
   3. Sau đó kích chọn nút “OK”.Hệ thống sẽ xóa bản ghi đó khỏi danh sách tài khoản trên giao diện, hệ thống sẽ cập nhật lại dữ liệu trong file dữ liệu “data\_Students.txt”.
5. Tìm kiếm thông tin sinh viên:

a.Người quản trị kích chọn nút “Tìm” trên giao diện.

b.Người dùng nhập mã sinh viên của sinh viên cần tìm kiếm vào ô tìm kiếm trong giao diện

c.Sau đó kích chọn nút “Tìm”.Hệ thống sẽ lấy thông tin tài khoản từ file dữ liệu “data\_Students.txt” bao gồm: Mã sinh viên, tên sinh viên, mã lớp, điểm lý thuyết, điểm thực hành, điểm trung bình, trạng thái môn và hiển thị lên bảng thông tin sinh viên

Use case kết thúc.

6.Tính toán điểm và trạng thái môn:

a. Người quản trị nhập các thông tin sinh viên bao gồm: Tên sinh viên, mã sinh viên, mã lớp, điểm lý thuyết, điểm thực hành.

b.Người quản trị kích chọn nút “Kết quả” trên giao diện.Hệ thống sẽ tính toán và hiển thị điểm trung bình và trạng thái môn của sinh viên trong bảng thông tin sinh viên.

7.Sắp xếp danh sách:

a.Người quản trị kích chọn nút “Sắp xếp” trên bảng menu.

b.Sau đó kích chọn nút “Tăng dần” hoặc “Giảm dần”.Hệ thống sẽ sắp xếp lại thông tin sinh viên trên bảng danh sách sinh viên dựa theo mã sinh viên.

8.Thêm mã lớp:

a.Người quản trị kích chọn nút “Create a new class” trên bảng menu.

b.Người quản trị nhập mã lớp muốn thêm vào ô mã lớp.

c.Sau đó kích chọn nút “OK”. Hệ thống sẽ thêm mã lớp vừa nhập vào bảng mã lớp mặc định trong bảng thông tin sinh viên.

3.2. Các luồng rẽ nhánh

1. Tại bước 2b, 3b, 4b, 5b, 6a, 8b luồng cơ bản khi người quản trị không nhập đầy đủ thông tin thì hệ thống thông báo lỗi “Hãy nhập đủ các trường thông tin” và use case kết thúc.
2. Tại bước 2b, 3b, 4b, 5b, 6a, 8b luồng cơ bản khi người quản trị nhập sai định dạng các trường thông tin thì hệ thống thông báo lỗi và yêu cầu nhập lại.
3. Tại bước 3b luồng cơ bản khi người dùng nhập mã sinh viên mới, hệ thống sẽ hiển thị thông báo lỗi “Không được thay đổi mã sinh viên” và use case kết thúc.
4. Tại bước 4c trong luồng cơ bản nếu người quản trị nhập mã sinh viên không tồn tại trong file dữ liệu “data\_Students.txt” ,thông báo lỗi “Sinh viên không tồn tại trong danh sách” xuất hiện.
5. Tại bước 5c trong luồng cơ bản nếu người quản trị nhập mã sinh viên không tồn tại trong file dữ liệu “data\_Students.txt” ,thông báo lỗi “Không tìm thấy” xuất hiện.
6. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với file dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

4. Các yêu cầu đặc biệt: Không có.

5. Tiền điều kiện: Người quản trị có tài khoản đã đăng kí và đăng nhập thành công.

6. Hậu điều kiện: Khi use case kết thúc thành công, mọi dữ diệu được lưu vào file dữ liệu “data\_Students.txt” thành công.

7. Điểm mở rộng: Không có.

**2.3.2. Mô hình hóa dữ liệu của hệ thống**

#### **2.3.2.1. Xác định các lớp , thuộc tính và phương thức chính**

#### Lớp Student (Sinh viên):

* msv (String)
* hoTen (String)
* maLop (String)
* diemLyThuyet(double)
* diemThucHanh(double)
* diemTrungBinh(double)
* ketQua(String)

#### Lớp DoAllObject (Các phương thức):

* findMsv() :int
* delete() :int
* search() :String
* edit() :int
* copyData()
* sortTangDan()
* sortGiamDan()

**2.3.2.2. Biểu đồ lớp**

A screenshot of a computer

Description automatically generated

## **2.4. Thiết kế hệ thống**

### **2.4.1. Thiết kế dữ liệu**

* Bảng:
* **Student (**msv, hoTen, maLop, diemLyThuyet, diemThucHanh, diemTrungBinh, ketQua)

### **2.4.2. Thiết kế giao diện**

A screenshot of a computer

Description automatically generated

*Hình2.4.2.1 . Giao diện đăng nhập*

A screenshot of a computer

Description automatically generated

*Hình2.4.2.2 . Giao diện đăng ký*

A screenshot of a computer

Description automatically generated

*Hinh2.4.2.3a. Giao diện thông tin sinh viên*

A screenshot of a computer

Description automatically generated

*Hinh2.4.2.3b. Giao diện thông tin sinh viên*

A screenshot of a computer

Description automatically generated

*Hình2.4.2.4a . Giao diện thêm mới sinh viên*

A screenshot of a computer

Description automatically generated

*Hình2.4.2.4b . Giao diện thêm mới sinh viên*

A screenshot of a computer

Description automatically generated

*Hình2.4.2.5 . Giao diện sửa thông tin sinh viên*

A screenshot of a computer

Description automatically generated

*Hình2.4.2.6a . Giao diện xóa thông tin sinh viên*

A screenshot of a computer

Description automatically generated

*Hình2.4.2.6b . Giao diện xóa thông tin sinh viên*

A screenshot of a computer

Description automatically generated

*Hình2.4.2.7a . Giao diện tìm kiếm thông tin sinh viên*

A screenshot of a computer

Description automatically generated

*Hình2.4.2.7b . Giao diện tìm kiếm thông tin sinh viên*

A screenshot of a computer

Description automatically generated

*Hình2.4.2.8 . Giao diện tính kết quả*

A screenshot of a computer

Description automatically generated

*Hình2.4.2.9 . Giao diện sắp xếp danh sách sinh viên*

A screenshot of a computer

Description automatically generated

*Hình2.4.2.10 . Giao diện thêm mới mã lớp*

### **2.4.3. Kiểm tra thử nghiệm**

1.Kế hoạch kiểm thử

Mục tiêu kiểm thử: Đảm bảo ứng dụng hoạt động đúng chức năng, giao diện thân thiện, hiệu suất tốt và an toàn bảo mật.

Các loại kiểm thử được thực hiện:

* Kiểm thử chức năng (Functional Testing)
* Kiểm thử giao diện người dùng (UI Testing)
* Kiểm thử tích hợp (Integration Testing)
* Kiểm thử hiệu năng (Performance Testing)
* Kiểm thử an toàn bảo mật (Security Testing)

2.Kết quả kiểm thử

2.1. Kiểm thử chức năng

* Kiểm tra các chức năng chính như quản lý thông tin sinh viên, quản lý điểm, trạng thái môn học, tài khoản v.v.
* Các chức năng đều hoạt động đúng theo yêu cầu và mô tả.
* Xử lý các trường hợp đầu vào hợp lệ và không hợp lệ.
* Các thao tác C-R-U-D (Tạo, Đọc, Cập nhật, Xóa) dữ liệu hoạt động chính xác.

2.2. Kiểm thử giao diện người dùng

* Giao diện được thiết kế rõ ràng, dễ hiểu và thân thiện với người dùng.
* Các thành phần giao diện như nút nhấn, văn bản, biểu tượng hiển thị đúng.
* Tương tác người dùng như click, nhập liệu, điều hướng diễn ra trơn tru.

2.3. Kiểm thử tích hợp

* Các module/thành phần của ứng dụng tích hợp và hoạt động đúng khi kết hợp với nhau.
* Luồng dữ liệu và trao đổi thông tin giữa các thành phần diễn ra chính xác.

2.4. Kiểm thử hiệu năng

* Ứng dụng đáp ứng tốc độ phản hồi và thời gian xử lý tương đối nhanh.
* Sử dụng tài nguyên hệ thống (bộ nhớ, CPU) trong giới hạn chấp nhận được.
* Ứng dụng vẫn hoạt động ổn định khi có nhiều người dùng truy cập cùng lúc.

2.5. Kiểm thử an toàn bảo mật

* Phân quyền truy cập người dùng được thiết lập đúng.
* Dữ liệu sinh viên được bảo vệ an toàn.

1. Đánh giá và kết luận

* Dự án hệ thống quản lý sinh viên đáp ứng các yêu cầu chức năng và phi chức năng đã đề ra.
* Giao diện thân thiện, dễ sử dụng, các chức năng hoạt động ổn định.
* Hiệu năng và tính bảo mật của ứng dụng đạt yêu cầu.
* Dự án có thể được triển khai và đưa vào sử dụng.

# **CHƯƠNG 3: TỔNG KẾT**

## **3.1. Nội dung đã thực hiện**

Phân tích nội dung cần thực hiện, các chức năng, dữ liệu liên quan để xây dựng hệ thống. Phân tích và đặc tả qua đó xác định những tác nhân, những use case của hệ thống. Áp dụng các kiến thức các thuật toán cùng ngôn ngữ lập trình java để áp dụng hoàn thành các chức năng như yêu cầu của hệ thống. Áp dụng giao diện đồ họa GUI vào thiết kế giao diện, form của hệ thống. Thành thạo Collections, bắt lỗi xử lý ngoại lệ, đọc lưu dữ liệu vào cơ sở dữ liệu, thiết kế giao diện hệ thống. Phân chia công việc theo từng phần để đảm bảo kết quả được thống nhất, hiệu quả.

Sau bài tiểu luận nhóm rút ra được những bài học:

● Tổ chức công việc sao cho phù hợp để đảm bảo tính hiệu quả và thống nhất cho bài tiểu luận.

● Cách thức để hoàn thành hệ thống một cách chỉnh chu.

● Phân tích vấn đề rõ ràng để tìm cách giải quyết khi gặp vấn đề trong việc xây dựng hệ thống.

## **3.2. Kết quả đạt được**

1. Giao Diện Người Dùng (UI/UX):
   * Thiết kế giao diện người dùng trực quan và thân thiện: Ứng dụng có giao diện dễ sử dụng, hỗ trợ tốt cho người dùng cuối.
   * Phát triển các màn hình chính: Bao gồm màn hình đăng nhập, đăng kí tài khoản,quản lí sinh viên, v.v.
2. Chức Năng Quản Lý Sinh viên:
   * Quản lý thông tin nhân viên: Quản lý thông tin cơ bản của sinh viên, thực hiện được các chức năng thêm, sửa, xóa và tìm kiếm sinh viên.
   * Quản lý điểm: Lưu trữ được điểm số của sinh viên theo từng môn và tính toán được điểm trung bình.
   * Quản lý trạng thái môn: Kiểm tra được trạng thái môn học của sinh viên theo điểm.
3. Phản Hồi và Điều Chỉnh:
   * Thu thập phản hồi từ người dùng: Dựa trên phản hồi của người dùng cuối để điều chỉnh và cải tiến ứng dụng.
   * Cải tiến và cập nhật: Định kỳ cập nhật các tính năng mới và sửa lỗi để nâng cao trải nghiệm người dùng.

**3.3. Hướng phát triển**

Qua việc xây dựng và phát triển hệ thống lần này nhóm em nhận thấy việc phân tích cũng như định hướng cho phần mềm theo yêu cầu đòi hỏi thành viên phải tìm hiểu kĩ cũng như đóng góp, xây dựng theo đúng hướng chức năng của người dùng, về cách thức xây dựng hệ thống bám sát theo những kiến thức đã học được ở bộ môn lập trình Java. Ban đầu sau khi nhận đề tài nhóm em gặp khó khăn trong việc xác định những công việc cần làm và phân chia công việc sao cho phù hợp nhất nhưng sau một thời gian đóng góp ý kiến cũng như đưa ra một số hướng đi cụ thể nhóm đã xác định được hướng đi cần thiết để xây dựng hệ thống sao cho hiệu quả nhất. Sau khi hoàn thiện hệ thống nhóm em đã hiểu hơn về phần thiết kế giao diện đồ họa cũng như biết các bước cần thiết từ khâu lên ý tưởng đến xây dựng hệ thống hoàn chỉnh. Do khả năng lập trình và kiến thức còn nhiều hạn chế nên không tránh khỏi những sai sót trong hệ thống. Kính mong thầy cô và các bạn có thể đóng góp ý kiến để nhóm em có thể khắc phục những tồn tại và phát triển hệ thống được bảo đảm hơn.

## **3.4. Tính khả thi**

1. Java Swing:
   * Java Swing là một phần của Java Foundation Classes (JFC) và được sử dụng rộng rãi để phát triển các ứng dụng desktop với giao diện người dùng đồ họa (GUI).
   * Swing cung cấp một bộ công cụ phong phú cho việc tạo giao diện người dùng, bao gồm các thành phần như button, text field, table, và tree, giúp dễ dàng xây dựng các ứng dụng quản lý nhân sự.
2. Phát Triển Nhanh và Dễ Dàng:
   * Java Swing có tài liệu phong phú và nhiều ví dụ minh họa, giúp lập trình viên dễ dàng bắt đầu và phát triển ứng dụng.
   * Swing hỗ trợ thiết kế giao diện theo mô hình MVC (Model-View-Controller), giúp tách biệt logic xử lý và giao diện người dùng, tạo điều kiện thuận lợi cho việc bảo trì và mở rộng.
3. Khả Năng Tương Thích Cao:
   * Ứng dụng Swing chạy trên MVC(Model-View-Controller) nên có tính đa nền tảng cao, chạy được trên nhiều hệ điều hành như Windows, macOS, và Linux.
4. Hiệu Suất và Bảo Mật:
   * Java là một ngôn ngữ hiệu suất cao, và Swing tận dụng được sức mạnh của Java để cung cấp trải nghiệm người dùng mượt mà.
   * Hệ thống bảo mật của Java giúp bảo vệ dữ liệu người dùng và ngăn chặn các mối đe dọa từ bên ngoài.

### **3.5.Tính Mở Rộng**

1. Kiến Trúc Phân Lớp:
   * Thiết kế ứng dụng theo kiến trúc phân lớp giúp dễ dàng mở rộng và bảo trì ứng dụng.
   * Sử dụng các mô hình thiết kế như MVC, DAO (Data Access Object) để quản lý logic nghiệp vụ và tương tác với cơ sở dữ liệu.
2. Tích Hợp Công Nghệ Mới:
   * Dễ dàng tích hợp các thư viện Java khác, chẳng hạn như Hibernate cho ORM (Object-Relational Mapping) và Spring Framework cho quản lý các nghiệp vụ phức tạp.
   * Khả năng tích hợp với các dịch vụ web (RESTful APIs) để mở rộng chức năng và kết nối với các hệ thống khác.
3. Mở Rộng Chức Năng:
   * Thêm các module hoặc plugin mới để mở rộng chức năng của ứng dụng quản lý sinh viên, ví dụ như quản lý công nợ, đăng kí tín chỉ ,..
   * Thiết kế giao diện người dùng linh hoạt, cho phép tùy chỉnh và thêm mới các màn hình và chức năng theo yêu cầu của doanh nghiệp.
4. Quản Lý Dữ Liệu Lớn:
   * Sử dụng các công nghệ như Apache Kafka và Apache Hadoop để xử lý và phân tích dữ liệu lớn, giúp cung cấp các báo cáo và dự đoán thông minh.
   * Tối ưu hóa cơ sở dữ liệu và sử dụng caching (bộ nhớ đệm) để cải thiện hiệu suất ứng dụng khi xử lý lượng dữ liệu lớn.
5. Hỗ Trợ Đa Ngôn Ngữ và Đa Văn Hóa:
   * Ứng dụng có thể được quốc tế hóa (i18n) để hỗ trợ nhiều ngôn ngữ và văn hóa khác nhau, giúp dễ dàng triển khai ở nhiều quốc gia.
6. Phân Quyền và Bảo Mật:
   * Tích hợp các hệ thống quản lý quyền truy cập và xác thực người dùng để đảm bảo chỉ những người dùng có quyền mới có thể truy cập vào các chức năng quan trọng.
   * Sử dụng mã hóa để bảo vệ dữ liệu nhạy cảm và tuân thủ các quy định bảo mật dữ liệu như GDPR.

### **3.6.Kết Luận**

Java Swing là một lựa chọn khả thi và mạnh mẽ để xây dựng ứng dụng quản lý nhân sự với giao diện người dùng đồ họa. Với tính khả thi cao và khả năng mở rộng linh hoạt, ứng dụng có thể phát triển và đáp ứng các nhu cầu đa dạng của doanh nghiệp. Việc sử dụng Java Swing cùng với các công nghệ và kiến trúc hiện đại giúp đảm bảo rằng ứng dụng có thể duy trì và nâng cấp một cách dễ dàng trong tương lai.

# **TÀI LIỆU THAM KHẢO**

[1] Đoàn Văn Ban; *Lập trình hướng đối tượng với Java*; NXB khoa học và kỹ thuật, Hà Nội 2005

[2] Đoàn Văn Ban; *Lập trình Java nâng cao*; NXB khoa học và kỹ thuật, Hà Nội 2006

[3] Nguyễn Bá Nghiễn; *Lập trình java*; NXB thông tin và truyền thông 2018.