**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐÔNG Á**

**KHOA: CÔNG NGHỆ THÔNG TIN**



**BÀI TẬP LỚN**

**HỌC PHẦN : LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG VỚI JAVA**

**ĐỀ SỐ:TÊN ĐỀ TÀI: XÂY DỰNG ỨNG DỤNG QUẢN LÝ CỦA HÀNG ĐIỆN THOẠI HÙNG MOBLE**

|  |  |  |
| --- | --- | --- |
| **Sinh viên thực hiện** | **Lớp** | **Khóa** |
| **Lê Viết Hùng** | **DCCNTT12.10.11** | **12** |
| **Phạm Văn Tuyến** | **DCCNTT12.10.11** | **12** |
| **Trần Quốc Duy** | **DCCNTT12.10.11** | **12** |
| **Nguyễn Lê Điệp** | **DCCNTT12.10.11** | **12** |
| **Bùi Thị Yến** | **DCCNTT12.10.11** | **12** |

**Bắc Ninh, năm 2023**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐÔNG Á**

**KHOA: CÔNG NGHỆ THÔNG TIN**

**BÀI TẬP LỚN**

**HỌC PHẦN : LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG VỚI JAVA**

**Nhóm 3**

**ĐỀ SỐ:TÊN ĐỀ TÀI: XÂY DỰNG ỨNG DỤNG QUẢN LÝ CỦA HÀNG ĐIỆN THOẠI HÙNG MOBLE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Sinh viên thực hiện** | **Mã sinh viên** | **Điểm bằng số** | **Điểm bằng chữ** |
| **1** | **Lê Viết Hùng** | **20213140** |  |  |
| **2** | **Phạm Văn Tuyến** | **20213085** |  |  |
| **3** | **Trần Quốc Duy** | **20213279** |  |  |
| **4** | **Nguyễn Lê Điệp** | **20213177** |  |  |
| **5** | **Bùi Thị Yến** | **20213093** |  |  |

|  |  |
| --- | --- |
| **CÁN BỘ CHẤM 1**  *(Ký và ghi rõ họ tên)* | **CÁN BỘ CHẤM 2**  *(Ký và ghi rõ họ tên)* |

# **LỜI NÓI ĐẦU**

Ngày nay, công nghệ thông tin ngày càng phát triển thì nhu cầu cuộc sống cũng như việc ứng dụng Công nghệ thông tin vào công việc của con người được phát triển mạnh mẽ và ngày càng đa dạng hơn, đặc biệt vấn để quản lý đang được mọi người quan tâm và phát triển. Nhiều công nghệ sử dụng mới được áp dụng hàng ngày, cũng như đưa phần mềm vào quản lý ngày càng trực quan và sinh động hơn vời người sử dụng, nhằm phục vụ nhu cầu truyền thông của con người, hoạt động kinh doanh điện thoại di động ngày càng phát triển mạnh mẽ trong xu hướng bán lẻ của các doanh nghiệp.

Hiện nay, đa số các cửa hàng điện thoại di động đều chưa có phần mềm quản lý, các phương pháp quản lý đều thực hiện một cách thủ công chủ yếu bằng ghi chép sổ sách... Nhằm giảm thao tác thủ công, mang lại tính chính xác và hiệu quả cao trong công tác quản lý hoạt động kinh doanh. Với mục tiêu trên thì trọng tâm của để tài này sẽ đi sâu nghiên cứu và phát triển phần mềm "Quản lý cửa hàng diện thoại di động" Là một dể tài mang tính thực tế và đáp ứng được nhu cầu trong công tác quản lý. Để đáp ứng được nhu cầu cho việc quản lý cửa hàng dễ dàng và thuân tiện ..., Vì vậy mà em đã chọn để tài này để phát triển phần mềm của mình

# **MỤC LỤC**

[**LỜI NÓI ĐẦU** 3](#_Toc148303765)

[**MỤC LỤC** 4](#_Toc148303766)

[DANH MỤC TỪ VIẾT TẮT 6](#_Toc148303767)

[**PHẦN 1 : CÁC KIẾN THỨC NỀN TẢNG** 7](#_Toc148303768)

[**I. TỔNG QUAN VỀ NGÔN NGỮ JAVA** 7](#_Toc148303769)

[**I.1 Khái niệm Java** 7](#_Toc148303770)

[**I.2 Các tính năng Java** 7](#_Toc148303771)

[**I.3 Ưu nhược điểm của Java** 8](#_Toc148303772)

[**I.4 Ứng dụng của Java** 8](#_Toc148303773)

[**II. TỔNG QUAN VỀ LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG** 9](#_Toc148303774)

[**II.1. Khái niệm lập trình hướng đối tượng** 9](#_Toc148303775)

[**II.2. Các nguyên lý cơ bản của lập trình hướng đối tượng** 10](#_Toc148303776)

[**II.3. Các ưu nhược điểm của lập trình hướng đối tượng** 11](#_Toc148303777)

[**PHẦN 2 : TỔNG QUAN VÀ PHÂN TÍCH BÀI TOÁN** 12](#_Toc148303778)

[**I. GIỚI THIỆU BÀI TOÁN** 12](#_Toc148303779)

[**I.1. Phân tích hiện trạng** 12](#_Toc148303780)

[**II. CÁC TÁC NHÂN VÀ CHỨC NĂNG** 12](#_Toc148303781)

[**II.1. Các tác nhân của hệ thống** 12](#_Toc148303782)

[**II.2. Các chức năng của hệ thống** 20](#_Toc148303783)

[**PHẦN 3 : THIẾT KẾ HỆ THỐNG** 25](#_Toc148303784)

[**I. THIẾT KẾ MÔ HÌNH LỚP** 25](#_Toc148303785)

[**I.1 Biểu đồ lớp** 25](#_Toc148303786)

[**II. THIẾT KẾ CƠ SỞ DỮ LIỆU** 28](#_Toc148303787)

[**II.1 Sơ đồ cơ sở dữ liệu** 28](#_Toc148303788)

[**II.2 Nhập dữ liệu** 28](#_Toc148303789)

[**PHẦN 4 : CÀI ĐẶT CHƯƠNG TRÌNH** 32](#_Toc148303790)

[**I. MỘT SỐ CÀI ĐẶT CHÍNH** 32](#_Toc148303791)

[**I.1. Các công cụ** 32](#_Toc148303792)

[**II. GIAO DIỆN CÁC CHỨC NĂNG CHÍNH** 34](#_Toc148303793)

[**II.1. Các giao diện của hệ thống** 34](#_Toc148303794)

[**PHẦN 5 : KẾT LUẬN** 43](#_Toc148303795)

[Kết quả đạt được 43](#_Toc148303796)

[Hướng phát triển 43](#_Toc148303797)

[**TÀI LIỆU THAM KHẢO** 44](#_Toc148303798)

DANH MỤC TỪ VIẾT TẮT

|  |  |
| --- | --- |
| **Từ viết tắt** | **Diễn giải** |
| API | Giao diện lập trình ứng dụng (Application Programming Interface) |
| CNTT | Công nghệ thông tin |
| CSDL | Cơ sở dữ liệu |
| IDE |  |

# **PHẦN 1 : CÁC KIẾN THỨC NỀN TẢNG**

## **I. TỔNG QUAN VỀ NGÔN NGỮ JAVA**

### **I.1 Khái niệm Java**

Java là một ngôn ngữ lập trình được sử dụng rộng rãi để viết mã cho các ứng dụng web. Ngôn ngữ này là lựa chọn phổ biến của các nhà phát triển trong hơn 2 thập niên. Hiện nay có hàng triệu ứng dụng Java đang được sử dụng. Java là một ngôn ngữ đa nền tảng, hướng đến đối tượng, lấy mạng làm trung tâm và có thể được sử dụng như một nền tảng. Đây là một ngôn ngữ lập trình nhanh, bảo mật, đáng tin cậy dùng để viết mã cho mọi thứ từ ứng dụng di động, phần mềm doanh nghiệp cho đến các ứng dụng dữ liệu lớn và công nghệ phía máy chủ.

Java sử dụng cấu trúc lập trình hướng đối tượng (object-oriented programming - OOP) và được xây dựng trên cơ sở của ngôn ngữ lập trình C++. Nó cung cấp một môi trường chạy ảo (virtual machine) gọi là Java Virtual Machine (JVM), giúp các chương trình Java có thể chạy trên nhiều nền tảng khác nhau mà không cần phải biên dịch lại.

Hiện tại, Java là một trong những ngôn ngữ lập trình phổ biến nhất hiện nay, được sử dụng rộng rãi trong các ứng dụng di động, phát triển web, lập trình trên các thiết bị nhúng, máy tính cá nhân, máy chủ, game và nhiều lĩnh vực khác.

### **I.2 Các tính năng Java**

* Đa nền tảng :

Java được thiết kế để có thể chạy trên nhiều nền tảng khác nhau, vì vậy nó rất phù hợp cho việc phát triển các ứng dụng đa nền tảng. Java sử dụng một máy ảo (JVM - Java Virtual Machine) để chạy mã nguồn, vì vậy mã nguồn được viết một lần và có thể chạy trên nhiều hệ điều hành khác nhau mà không cần thay đổi.

* **Quản lý bộ nhớ tự động :**

Java có tính năng tự động quản lý bộ nhớ, tức là nó tự động thu dọn các vùng nhớ không sử dụng nữa để giảm thiểu các lỗi bộ nhớ. Điều này giúp cho các ứng dụng được viết bằng Java có thể chạy ổn định và tránh các lỗi liên quan đến bộ nhớ.

* **Hỗ trợ đa luồng :**

Java có thể xử lý đa luồng, cho phép chương trình thực hiện nhiều tác vụ cùng một lúc. Điều này giúp cho các ứng dụng có thể chạy nhanh và hiệu quả hơn, đặc biệt là khi phải xử lý nhiều tác vụ cùng một lúc.

* **Tính bảo mật cao :**

Java có các tính năng bảo mật như kiểm tra kiểu tĩnh và kiểm tra lỗi trên đường dẫn. Java được thiết kế để giảm thiểu các lỗ hổng bảo mật và các vấn đề liên quan đến an ninh.

* **Hỗ trợ các thư viện và công cụ phát triển mạnh mẽ :**

Java có các thư viện tiêu chuẩn và các công cụ phát triển như IDE (Integrated Development Environment) để giúp các nhà phát triển tạo ra các ứng dụng dễ dàng hơn. Các thư viện và công cụ này cung cấp các tính năng như: tạo giao diện người dùng, xử lý dữ liệu, kết nối cơ sở dữ liệu và nhiều hơn nữa,...

* **Tính di động :**

Java được sử dụng rộng rãi trong lĩnh vực di động bao gồm: các thư viện hỗ trợ việc phát triển ứng dụng di động, đóng gói ứng dụng thành các file .jar hoặc .apk, cung cấp các tính năng như xử lý đa nhiệm, kết nối mạng và tích hợp với các thiết bị phần cứng như máy ảnh.

* **Tính độc lập với nền tảng :**

Java có thể chạy trên nhiều nền tảng khác nhau và không phụ thuộc vào bất kỳ nền tảng cụ thể nào. Điều này giúp cho các ứng dụng Java có thể được triển khai trên nhiều hệ thống khác nhau mà không cần sửa đổi mã nguồn.

* **Tính kế thừa và đa hình :**

Java là ngôn ngữ lập trình hướng đối tượng (OOP), vì vậy nó có các tính năng như kế thừa, đa hình và đóng gói. Các tính năng này giúp cho mã nguồn được tái sử dụng và giảm thiểu sự trùng lặp trong mã nguồn.

* **Tính mở rộng :**

Java có tính năng mở rộng, cho phép các nhà phát triển thêm các tính năng mới vào ngôn ngữ bằng cách tạo các thư viện và API riêng. Điều này giúp cho Java có thể được sử dụng trong nhiều lĩnh vực khác nhau và được phát triển theo các hướng khác nhau.

### **I.3 Ưu nhược điểm của Java**

* **Ưu điểm của Java**
* Độ tin cậy cao
* Tính đa nền tảng
* Quản lý bộ nhớ tự động
* Công cụ phát triển phong phú
* Hỗ trợ đa luồng
* **Nhược điểm của Java**
* Tốc độ chậm hơn so với các ngôn ngữ lập trình gần sát với phần cứng, chẳng hạn như C hoặc C++.
* Java có thể chạy trên nhiều nền tảng khác nhau, nhưng ứng dụng này có thể cần đến một trình biên dịch hoặc máy ảo Java riêng biệt để có thể chạy trên các thiết bị di động.
* Sử dụng bộ nhớ lớn hơn so với một số ngôn ngữ lập trình khác.
* Cú pháp phức tạp hơn so với một số ngôn ngữ lập trình khác.

### **I.4 Ứng dụng của Java**

* **Phát triển ứng dụng máy tính :**

Java là một trong những ngôn ngữ lập trình phổ biến nhất để phát triển các ứng dụng máy tính đa nền tảng. Java có thể được sử dụng để phát triển các ứng dụng trên , máy chủ và các ứng dụng trên nền web. Nó cung cấp các thư viện mạnh mẽ để xử lý các nhu cầu của các ứng dụng máy tính như xử lý dữ liệu, đồ họa, đầu vào và đầu ra.

* **Phát triển ứng dụng di động :**

Java là một trong những ngôn ngữ lập trình phổ biến nhất để phát triển các ứng dụng di động cho [hệ điều hành Android](https://fptshop.com.vn/tin-tuc/danh-gia/he-dieu-hanh-android-la-gi-lich-su-va-nhung-cau-chuyen-thu-vi-ve-android-152248). Nó được sử dụng để phát triển các ứng dụng cho điện thoại di động và [máy tính bảng](https://fptshop.com.vn/may-tinh-bang). Java cung cấp một số khung như Android SDK để phát triển ứng dụng di động.

* **Phát triển game :**

Java cũng được sử dụng để phát triển các game trên nhiều nền tảng khác nhau, bao gồm các game trên máy tính, [điện thoại](https://fptshop.com.vn/dien-thoai) di động và máy tính bảng. Các tính năng của Java như: tính đa luồng, xử lý đồ họa âm thanh đã được sử dụng để phát triển các game phức tạp.

* **Phát triển web :**

Java là một ngôn ngữ lập trình phía máy chủ rất nổi tiếng trong việc xây dựng các ứng dụng web, với các framework phổ biến nhất là Spring, Struts, Hibernate, Apache Hadoop và JSF. Các trang web lớn như LinkedIn, AliExpress, Amazon… đều sử dụng Java để lập trình.

* Phát triển các ứng dụng IOT

Các nhà phát triển thường xem Java là ngôn ngữ tốt nhất để phát triển thiết bị IoT. Về cơ bản, Java là một ngôn ngữ lập trình không có nền tảng. Vì vậy, người ta chỉ cần tạo mã một lần và nó sẽ hoạt động trên nhiều thiết bị khác nhau, từ đó giúp giảm bớt khối lượng công việc tổng thể.

Lý do khác khiến Java trở thành sự lựa chọn tốt nhất cho IoT là vì khả năng thích ứng và tính di động của nó. Như đã đề cập ở trên, Java được sử dụng để phát triển web, phát triển Android, phát triển ứng dụng desktop… So với các ngôn ngữ lập trình khác, Java có khả năng tích hợp nhiều công nghệ dễ dàng hơn. Chính vì vậy, Java được sử dụng trong các hệ thống định vị ô tô, các thiết bị y tế khác nhau và thiết bị điện tử nhúng.

* Phát triển Big Data

Ngoài Python thì còn có nhiều ngôn ngữ lập trình phổ biến khác cũng được sử dụng trong Big Data. Mặc dù vậy, khi nói đến Công nghệ Big Data, đa phần các lập trình viên đều ưa chuộng Java hơn. Điều này là do hầu hết các công cụ Big Data nổi tiếng (như Hadoop, Deeplearning4j) đều sử dụng Java.

## **II. TỔNG QUAN VỀ LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG**

### **II.1. Khái niệm lập trình hướng đối tượng**

**Lập trình hướng đối tượng (Object Oriented Programming – OOP)** là một trong những kỹ thuật lập trình rất quan trọng và sử dụng nhiều hiện nay.

1. Đối tượng

Một đối tượng bao gồm 2 thông tin là thuộc tính và phương thức :

* Thuộc tính chính là những thông tin, đặc điểm của đối tượng. Ví dụ: con người có các đặc tính như mắt, mũi, tay, chân…
* Phương thức là những thao tác, hành động mà đối tượng đó có thể thực hiện. Ví dụ: một người sẽ có thể thực hiện hành động nói, đi, ăn, uống, . . .

1. Lớp

* Một lớp là một kiểu dữ liệu bao gồm các thuộc tính và các phương thức được định nghĩa từ trước. Đây là sự trừu tượng hóa của đối tượng. Khác với kiểu dữ liệu thông thường, một lớp là một đơn vị (trừu tượng) bao gồm sự kết hợp giữa các phương thức và các thuộc tính. Hiểu nôm na hơn là các đối tượng có các đặc tính tương tự nhau được gom lại thành một lớp đối tượng.

1. Sự khác nhau giữa đối tượng và lớp

Lớp bạn có thể hiểu nó như là khuôn mẫu, đối tượng là một thực thể thể hiện dựa trên khuôn mẫu đó. Ví dụ: Ta nói về loài chó, bạn có thể hiểu nó là class (lớp) chó có:

* Các thông tin, đặc điểm: 4 chân, 2 mắt, có đuôi, có chiều cao, có cân nặng, màu lông…
* Các hành động như: sủa, đi, ăn, ngủ…

Đối tượng thì chính là con chó Phú Quốc ta đang nuôi trong nhà cũng mang đặc tính của lớp chó.

### **II.2. Các nguyên lý cơ bản của lập trình hướng đối tượng**

1. Tính đóng gói (Encapsulation)

* Các dữ liệu và phương thức có liên quan với nhau được đóng gói thành các lớp để tiện cho việc quản lý và sử dụng. Tức là mỗi lớp được xây dựng để thực hiện một nhóm chức năng đặc trưng của riêng lớp đó.
* Ngoài ra, đóng gói còn để che giấu một số thông tin và chi tiết cài đặt nội bộ để bên ngoài không thể nhìn thấy.

1. Tính kế thừa (Inheritance)

* Nó cho phép xây dựng một lớp mới dựa trên các định nghĩa của lớp đã có. Có nghĩa là lớp cha có thể chia sẽ dữ liệu và phương thức cho các lớp con. Các lớp con khỏi phải định nghĩa lại, ngoài ra có thể mở rộng các thành phần kế thừa và bổ sung thêm các thành phần mới. Tái sử dụng mã nguồn 1 cách tối ưu, tận dụng được mã nguồn. Một số loại kế loại kế thừa thường gặp: đơn kế thừa, đa kế thừa, kế thừa đa cấp, kế thừa thứ bậc.
* Khi bắt đầu xây dựng ứng dụng chúng ta sẽ bắt đầu thiết kế định nghĩa các lớp trước. Thông thường một số lớp có quan hệ với những lớp khác, chúng có những đặc tính giống nhau.
* Mỗi lớp đều đại diện cho một loại smartphone khác nhau nhưng lại có những thuộc tính giống nhau như gọi điện, nhắn tin, chụp hình. Thay vì sao chép những thuộc tính này, ta nên đặt chúng vào một lớp chung gọi là lớp cha. Chúng ta có thể định nghĩa lớp cha – trong trường hợp này là Smartphone và có những lớp con kế thừa từ nó, tạo ra một mối quan hệ cha/con.

1. Tính đa hình (Polymorphism)

* Tính đa hình là một hành động có thể được thực hiện bằng nhiều cách khác nhau. Đây lại là một tính chất có thể nói là chứa đựng hầu hết sức mạnh của lập trình hướng đối tượng.
* Hiểu một cách đơn giản hơn: Đa hình là khái niệm mà hai hoặc nhiều lớp có những phương thức giống nhau nhưng có thể thực thi theo những cách thức khác nhau.

1. Tính trừu tượng (Abstraction)

* Trừu tượng có nghĩ là tổng quát hóa một cái gì đó lên, không cần chú ý chi tiết bên trong. Nó không màng đến chi tiết bên trong là gì và người ta vẫn hiểu nó mỗi khi nghe về nó.
* Ở đây trong lập trình OOP, tính trừu tượng nghĩa là chọn ra các thuộc tính, phương thức của đối tượng cần cho việc giải quyết bài toán đang lập trình. Vì một đối tượng có rất nhiều thuộc tính phương thức, nhưng với bài toán cụ thể không nhất thiết phải chọn tất cả.

### **II.3. Các ưu nhược điểm của lập trình hướng đối tượng**

* Dựa trên nguyên lý kế thừa, trong quá trình mô tả các lớp có thể loại bỏ những chương trình bị lặp, dư. Và có thể mở rộng khả năng sử dụng các lớp mà không cần thực hiện lại. Tối ưu và tái sử dụng code hiệu quả.
* Đảm bảo rút ngắn thời gian xây dựng hệ thống và tăng năng suất thực hiện.
* Sự xuất hiện của 2 khái niệm mới là lớp và đối tượng chính là đặc trưng của phương pháp lập trình hướng đối tượng. Nó đã giải quyết được các khuyết điểm của phương pháp lập trình hướng cấu trúc để lại. Ngoài ra 2 khái niệm này đã giúp biểu diễn tốt hơn thế giới thực trên máy tính.

# **PHẦN 2 : TỔNG QUAN VÀ PHÂN TÍCH BÀI TOÁN**

## **I. GIỚI THIỆU BÀI TOÁN**

### **I.1. Phân tích hiện trạng**

**\* Đánh giá hiện trạng**

Điện thoại là một thiết bị không thể thiếu trong cuộc sống hiện đại. Nó không chỉ giúp chúng ta liên lạc, giải trí, học tập, làm việc mà nó còn là một phương tiện để thể hiện cá tính và phong cách của mỗi người. Chính vì vậy, nhu cầu mua điện thoại luôn cao và sôi động trên thị trường.

Khách hàng phụ thuộc 100% vào lời tư vấn của người bán hàng. Đây là tình trạng rất hay gặp phải hiện nay. Nhiều bạn trẻ, bạn nữ mua điện thoại nhưng không am hiểu về công nghệ nên dễ bị lừa hoặc mua phải hàng kém chất lượng.

Khách hàng không kiểm tra kỹ thông tin về điện thoại trước khi mua. Đây là sai lầm rất nguy hiểm khi mua bán điện thoại online. Nhiều người chỉ nhìn vào hình ảnh hoặc giá cả mà quyết định mua hàng, không biết rằng có thể hình ảnh được chỉnh sửa hoặc giá cả được đẩy lên cao để thu hút khách hàng.

Khách hàng không biết cách phân biệt hàng chính hãng và hàng nhái. Đây là vấn đề nan giải khi mua bán điện thoại hiện nay. Hàng nhái ngày càng được làm giống hàng chính hãng đến từng chi tiết, khó phát hiện bằng mắt thường. Nếu không có kiến thức hoặc công cụ kiểm tra, khách hàng dễ bị lừa mua phải hàng nhái với giá cao.

Khách hàng không biết cách bảo quản và sử dụng điện thoại đúng cách. Đây là nguyên nhân khiến cho điện thoại mau hỏng hoặc giảm tuổi thọ. Nhiều người không chú ý đến việc sạc pin, lau chùi máy, tránh va đập hay nhiệt độ cao khi sử dụng điện thoại.

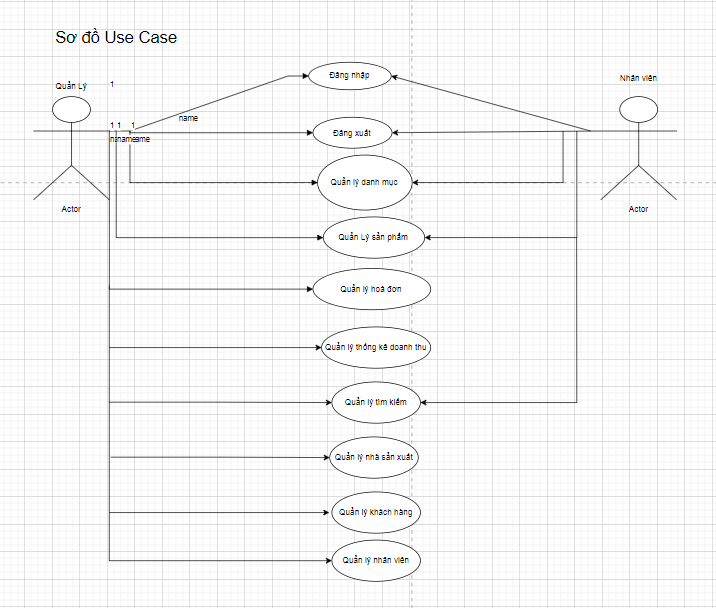
Cửa hàng điện thoại di động Hùng Mobile xây dựng ra để mang đến cho khách hàng những trải nghiệm và sản phẩm tốt nhất và mang lại sự thoải mái đến cho khách hàng.

Hệ thống cửa hàng điện thoại Hùng Mobile là một vấn đề quan trọng đối với việc quản lý vận hành một cửa hàng bán lẻ trong điện thoại di động. Hệ thống tích hợp nhiều khía cạnh, từ quản lý hàng kho hàng tới khách hàng, từ quản lý nhân viên đến phân tích dữ liệu.

## **II. CÁC TÁC NHÂN VÀ CHỨC NĂNG**

### **II.1. Các tác nhân của hệ thống**

**II.1.1. Sơ đồ User case**



*Hình 2.2.1 Sơ đồ User-Case*

\* Danh sách Actor của hệ thống

|  |  |  |
| --- | --- | --- |
| STT | Tên Actor | Ý nghĩa |
| 1 | Admin(Quản lý) | Có toàn quyền tương tác với hệ thống, có quyền điều khiển cũng như kiểm soát mọi hoạt động của hệ thống. |
| 2 | Nhân viên | Thực hiện các chức năng: bán điện thoại, tìm điện thoại, quản lý sản phẩm … |

*Bảng 2.2.1 : Danh sách các Actor*

\* Danh sách các Usecase của hệ thống

|  |  |  |
| --- | --- | --- |
| STT | Tên Use case | Ý nghĩa |
| 1 | Đăng nhập | Cho phép người dùng đăng nhập vào hệ thống. |
| 2 | Đăng xuất | Cho phép người dùng đăng xuất khỏi hệ thống. |
| 3 | Quản lý danh mục | Người dùng sử dụng chức năng thêm, sửa, xoá , cập nhật và lưu danh mục. |
| 4 | Quản lý sản phẩm | Người dùng sử dụng chức năng thêm , sửa , xoá , lưu và cập nhật sản phẩm trong hệ thống. |
| 5 | Quản lý hoá đơn | Người dùng sử dụng chức năng thêm, sửa, xoá , cập nhật và lưu thông tin hoá đơn, thêm hoá đơn mới , thanh toán hoá đơn và xuất hoá đơn cho khách hàng. |
| 6 | Quản lý thống kê doanh thu | Người dùng cập nhật và thống kê doanh thu từ ngày bao nhiêu đến ngày bao nhiêu. |
| 7 | Quản lý tìm kiếm | Người dùng sử dụng chức năng tìm kiếm sản phẩm và tìm kiếm nhà sản xuất |
| 8 | Quản lý nhà sản xuất | Người dùng sử dụng chức năng thêm, sửa xoá, lưu, và cập nhật thông tin về nhà sản xuất. |
| 9 | Quản lý khách hàng | Người dùng sử dụng chức năng thêm, sửa xoá, lưu, và cập nhật thông tin về khách hàng. |
| 10 | Quản lý nhân viên | Người dùng sử dụng chức năng thêm, sửa xoá, lưu, và cập nhật thông tin về nhân viên. |

*Bảng 2.2.2 : Danh sách các Usecase*

**II.1.2. Đặc tả các Use case**

**1. Đặc tả Use case “Đăng nhập”**

\* Mô tả : Admin, nhân viên sử dụng Use case “Đăng nhập” thực hiện đăng nhập vào hệ thống.

\* Sự kiện chính :

* Hệ thống hiển thị giao diện đăng nhập.
* Người dùng nhập tài khoản bao gồm tên đăng nhập và mật khẩu.
* Hệ thống kiểm tra và xác nhận thông tin đăng nhập.
* Hiển thị giao diện chính của chương trình.
* Usecase kết thúc.

- Sự kiện phụ:

- Sự kiện phụ 1:

* + Người dùng hủy bỏ yêu cầu đăng nhập.
  + Hệ thống đóng lại.
  + Usecase kết thúc.

- Sự kiện phụ 2:

* Thông tin người dùng nhập vào không hợp lệ.
* Hệ thống hiển thị thông báo lỗi.
* Usecase kết thúc.

*-* Trạng thái hệ thống trước khi Usecase “Đăng nhập” được thực hiện:

* không đòi hỏi yêu cầu gì trước đó

- Trạng thái hệ thống sau khi Usecase “Đăng nhập” được thực hiện:

* Trường hợp thành công: hệ thống sẽ hiển thị giao diện chính và người dùng có thể thực hiện các chức theo đúng quyền hạn của mình.
* Trường hợp thất bại: hệ thống thông báo lỗi .

**2. Đặc tả Use case “Đăng xuất”**

\* Mô tả : Admin, nhân viên sử dụng Use case “Đăng xuất” thực hiện đăng xuất khỏi hệ thống.

\* Sự kiện chính :

* Hệ thống hiển thị giao diện trang chủ sau khi đăng nhập.
* Usecase kết thúc.

- Sự kiện phụ:

- Sự kiện phụ 1:

* + Người dùng chấp nhận yêu cầu đăng xuất.
  + Hệ thống đóng lại.
  + Usecase kết thúc.

- Sự kiện phụ 2:

* Người dùng không chấp nhận yêu cầu đăng xuất.
* Người dùng tiếp tục sử dụng hệ thống.
* Usecase kết thúc.

*-* Trạng thái hệ thống trước khi Usecase “Đăng xuất” được thực hiện:

* không đòi hỏi yêu cầu gì trước đó.

- Trạng thái hệ thống sau khi Usecase “Đăng xuất” được thực hiện:

* Trường hợp thành công: hệ thống sẽ bị đăng xuất.

**3. Đặc tả Use case “Quản lý bán hàng”**

\* Mô tả: Quản lý hoá đơn của khách hàng, bao gồm thông tin về sản phẩm, loại sản phẩm được đặt mua, mã sản phẩm, số lượng, giá.

\* Thành phần chính: Nhân viên bán hàng.

\* Tương tác:

* Tạo hoá đơn mới.
* Sửa thông tin hoá đơn.
* Xoá hoá đơn tạo cho khách hàng.
* Lưu hoá đơn vào kho dữ liệu.
* Cập nhật thông tin hoá đơn.
* Thanh toán tiền cho khách hàng.
* Xuất hoá đơn cho khách hàng.

**4. Đặc tả Use case “Quản lý sản phẩm”**

\* Mô tả: Quản lý thông tin về sản phẩm điện thoại, bao gồm tên, mô tả, nhà sản xuất, thời hạn bảo hành, giá, số lượng tồn kho, hình ảnh, và các thông tin khác.

\* Thành phần chính: Nhân viên quản lý sản phẩm.

\* Tương tác:

* Thêm sản phẩm mới.
* Sửa thông tin sản phẩm mới.
* Cập nhật thông tin sản phẩm.
* Xóa sản phẩm khỏi hệ thống.
* Lưu thông tin sản phẩm vào kho.

**5. Đặc tả Use case “Quản lý danh mục”**

\* Mô tả: Quản lý danh mục bao gồm mã danh mục và tên danh mục.

\* Thành phần chính: Nhân viên quản lý sản phẩm.

\* Tương tác:

* Thêm loại sản phẩm mới.
* Sửa thông tin loại sản phẩm mới.
* Cập nhật thông tin loại sản phẩm.
* Xóa loại sản phẩm khỏi hệ thống.
* Lưu thông tin loại sản phẩm vào kho.

**6. Đặc tả Use case “Quản lý thống kê”**

\* Mô tả: Use case này cho phép quản lý hệ thống thực hiện các thống kê và báo cáo liên quan đến doanh thu của cửa hàng bán hàng điện thoại.

\* Thành phần chính: Nhân viên quản lý doanh thu.

\* Tương tác:

* Cập nhật bảng thống kê.
* Xuất thống kê được từ ngày bao nhiêu đến ngày bao nhiêu.

**7. Đặc tả Use case “Quản lý tìm kiếm”.**

\* Mô tả: Use case này cho phép người dùng tìm kiếm sản phẩm điện thoại và nhà sản xuất dựa trên các tiêu chí tìm kiếm dựa trên thông tin có trong bảng mà người dùng muốn tìm kiếm.

\* Thành phần chính: Người dùng.

\* Tiến hành:

Người dùng nhập thông tin tìm kiếm:

* Mô tả: Người dùng mở trang tìm kiếm và nhập thông tin tìm kiếm, bao gồm tên sản phẩm, hãng sản xuất, giá, và thông số kỹ thuật.
* Tiền hành thụ động:
* Hệ thống hiển thị gợi ý tìm kiếm dựa trên các tiêu chí phù hợp.

Hệ thống thực hiện tìm kiếm:

* Mô tả: Hệ thống tiến hành tìm kiếm dựa trên thông tin nhập từ người dùng.
* Tiền hành:
* Hệ thống tìm kiếm trong cơ sở dữ liệu sản phẩm điện thoại dựa trên thông tin tìm kiếm.
* Hệ thống trả về danh sách các sản phẩm phù hợp với tìm kiếm.

Hệ thống hiển thị kết quả tìm kiếm:

* Mô tả: Hệ thống hiển thị danh sách các sản phẩm điện thoại phù hợp với tìm kiếm của người dùng.
* Kết quả:
* Hệ thống hiển thị tên sản phẩm, hãng sản xuất, giá, hình ảnh sản phẩm và một số thông số kỹ thuật cơ bản.
* Người dùng có thể xem danh sách các sản phẩm tìm kiếm được.

Người dùng xem chi tiết sản phẩm:

* Mô tả: Người dùng có thể chọn sản phẩm từ danh sách kết quả để xem thông tin chi tiết về sản phẩm.
* Tiền hành:
* Hệ thống hiển thị trang chi tiết sản phẩm với thông tin chi tiết, mô tả, hình ảnh, giá và thông số kỹ thuật đầy đủ của sản phẩm.

**8. Đặc tả Use case “Quản lý nhà sản xuất”.**

\* Mô tả: Use case này cho phép người quản lý hoặc quản trị viên hệ thống thực hiện các hoạt động liên quan đến quản lý thông tin về nhà sản xuất của sản phẩm.

\* Thành phần chính: Người trị viên hệ thống.

\* Tiền hành:

Xem danh sách nhà sản xuất:

* Mô tả: Quản trị viên truy cập danh sách các nhà sản xuất đã được thêm vào hệ thống.
* Kết quả:
* Hệ thống hiển thị danh sách nhà sản xuất với thông tin cơ bản như tên, địa chỉ, số điện thoại và ngày tạo.

Thêm nhà sản xuất mới:

* Mô tả: Quản trị viên thêm một nhà sản xuất mới vào hệ thống.
* Tiền hành:
* Quản trị viên nhập thông tin cơ bản về nhà sản xuất, bao gồm tên, địa chỉ, số điện thoại, và các thông tin khác.
* Hệ thống ghi nhận thông tin mới và tạo một bản ghi mới trong cơ sở dữ liệu.

Sửa thông tin nhà sản xuất:

* Mô tả: Quản trị viên cập nhật thông tin của một nhà sản xuất đã tồn tại trong hệ thống.
* Tiền hành:
* Quản trị viên chọn một nhà sản xuất từ danh sách và chỉnh sửa thông tin liên quan.
* Hệ thống cập nhật thông tin của nhà sản xuất trong cơ sở dữ liệu.

Xóa nhà sản xuất:

* Mô tả: Quản trị viên xóa một nhà sản xuất khỏi hệ thống.
* Tiền hành:
* Quản trị viên chọn một nhà sản xuất từ danh sách và xác nhận xóa.
* Hệ thống xóa thông tin nhà sản xuất và ghi nhật ký xóa.

Xem chi tiết nhà sản xuất:

* Mô tả: Quản trị viên có thể chọn một nhà sản xuất từ danh sách để xem thông tin chi tiết về nhà sản xuất.
* Tiền hành:
* Hệ thống hiển thị trang chi tiết nhà sản xuất với thông tin đầy đủ, bao gồm tên, địa chỉ, số điện thoại và các thông tin khác.

**9. Đặc tả Use case “Quản lý khách hàng”.**

\* Mô tả: Use case này cho phép nhân viên hoặc quản trị viên hệ thống thực hiện các hoạt động quản lý thông tin về khách hàng, bao gồm thêm, sửa, xóa, và xem thông tin khách hàng.

\* Thành phần chính: Nhân viên hoặc quản trị viên hệ thống.

\* Tiền hành:

Xem danh sách khách hàng:

* Mô tả: Nhân viên hoặc quản trị viên truy cập danh sách các khách hàng đã được thêm vào hệ thống.
* Kết quả:
* Hệ thống hiển thị danh sách khách hàng với thông tin cơ bản như tên, địa chỉ, số điện thoại, địa chỉ email, và ngày tạo.

Thêm khách hàng mới:

* Mô tả: Nhân viên hoặc quản trị viên thêm một khách hàng mới vào hệ thống.
* Tiền hành:
* Nhân viên hoặc quản trị viên nhập thông tin cơ bản về khách hàng, bao gồm tên, địa chỉ, số điện thoại, địa chỉ email, và các thông tin khác.
* Hệ thống ghi nhận thông tin mới và tạo một bản ghi mới trong cơ sở dữ liệu.

Sửa thông tin khách hàng:

* Mô tả: Nhân viên hoặc quản trị viên cập nhật thông tin của một khách hàng đã tồn tại trong hệ thống.
* Tiền hành:
* Nhân viên hoặc quản trị viên chọn một khách hàng từ danh sách và chỉnh sửa thông tin liên quan.
* Hệ thống cập nhật thông tin của khách hàng trong cơ sở dữ liệu.

Xóa khách hàng:

* Mô tả: Nhân viên hoặc quản trị viên xóa một khách hàng khỏi hệ thống.
* Tiền hành:
* Nhân viên hoặc quản trị viên chọn một khách hàng từ danh sách và xác nhận xóa.
* Hệ thống xóa thông tin của khách hàng và ghi nhật ký xóa.

Xem chi tiết khách hàng:

* Mô tả: Nhân viên hoặc quản trị viên có thể chọn một khách hàng từ danh sách để xem thông tin chi tiết về khách hàng.
* Tiền hành:
* Hệ thống hiển thị trang chi tiết khách hàng với thông tin đầy đủ, bao gồm tên, địa chỉ, số điện thoại, địa chỉ email và các thông tin khác.

**10. Đặc tả Use case “Quản lý nhân viên”.**

\* Mô tả: Use case này cho phép quản lý hoặc quản trị viên hệ thống thực hiện các hoạt động quản lý thông tin về nhân viên, bao gồm thêm, sửa, xóa, và xem thông tin nhân viên.

\* Thành phần chính: Quản trị viên hệ thống.

\* Tiền hành:

Xem danh sách nhân viên:

* Mô tả: Quản trị viên truy cập danh sách các nhân viên đã được thêm vào hệ thống.
* Kết quả:
* Hệ thống hiển thị danh sách nhân viên với thông tin cơ bản như tên, vị trí công việc, số điện thoại, địa chỉ email và ngày tạo.

Thêm nhân viên mới:

* Mô tả: Quản trị viên thêm một nhân viên mới vào hệ thống.
* Tiền hành:
* Quản trị viên nhập thông tin cơ bản về nhân viên, bao gồm tên, vị trí công việc, số điện thoại, địa chỉ email, và các thông tin khác.
* Hệ thống ghi nhận thông tin mới và tạo một bản ghi mới trong cơ sở dữ liệu.

Sửa thông tin nhân viên:

* Mô tả: Quản trị viên cập nhật thông tin của một nhân viên đã tồn tại trong hệ thống.
* Tiền hành:
* Quản trị viên chọn một nhân viên từ danh sách và chỉnh sửa thông tin liên quan.
* Hệ thống cập nhật thông tin của nhân viên trong cơ sở dữ liệu.

Xóa nhân viên:

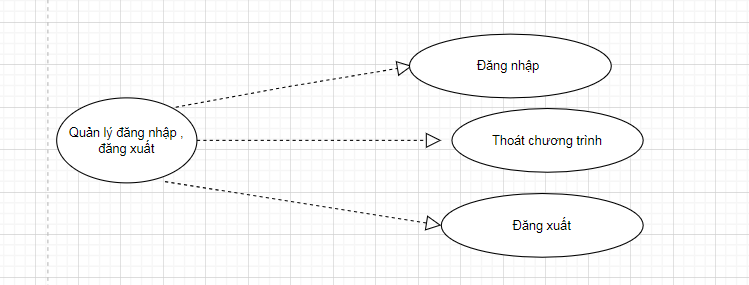
* Mô tả: Quản trị viên xóa một nhân viên khỏi hệ thống.
* Tiền hành:
* Quản trị viên chọn một nhân viên từ danh sách và xác nhận xóa.
* Hệ thống xóa thông tin của nhân viên và ghi nhật ký xóa.

Xem chi tiết nhân viên:

* Mô tả: Quản trị viên có thể chọn một nhân viên từ danh sách để xem thông tin chi tiết về nhân viên.
* Tiền hành:
* Hệ thống hiển thị trang chi tiết nhân viên với thông tin đầy đủ, bao gồm tên, vị trí công việc, số điện thoại, địa chỉ email và các thông tin khác.

### **II.2. Các chức năng của hệ thống**

* **Quản lý đăng nhập , đăng xuất :**



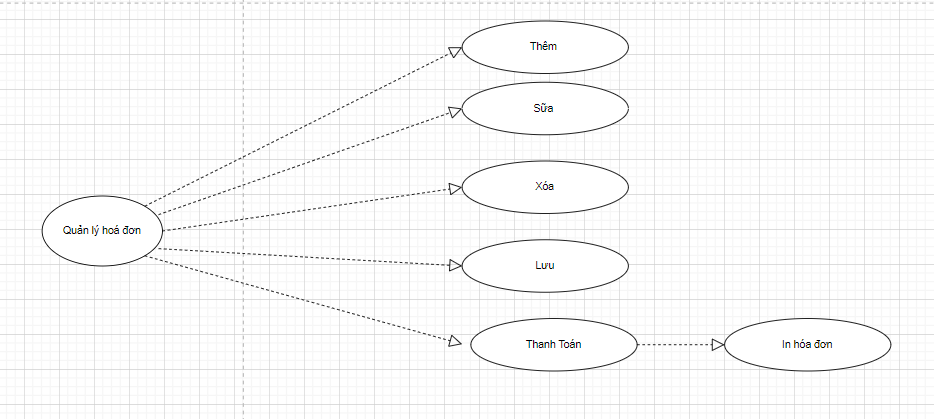
*Hình 2.2.2 Sơ đồ chức năng quản lý đăng nhập, đăng xuất*

+ Chức năng đăng nhập của hệ thống cho phép người dùng đăng nhập vào trang chủ của hệ thống.

+ Chức năng thoát chương trình cho phép người dùng thoát toàn bộ khỏi hệ thống trong giao diện đăng nhập.

+ Chức năng đăng xuất cho phép người dùng đăng xuất khỏi trang chủ hệ thống trong giao diện trang chủ của hệ thống.

* **Quản lý hoá đơn :**



*Hình 2.2.3 Sơ đồ chức năng quản lý bán hàng*

+ Chức năng thêm cho phép người dùng thêm thông tin hoá đơn bán hàng.

+ Chức năng sửa cho phép người dùng sửa thông tin trong hoá đơn bán hàng.

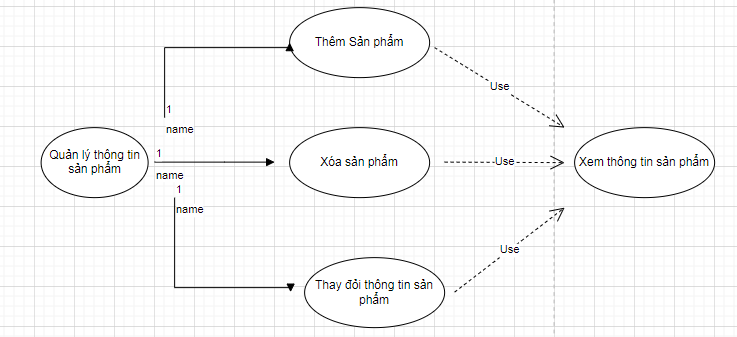
+ Chức năng xoá cho phép người dùng xoá hoá đơn bán hàng.

+ Chức năng lưu cho phép người dùng lưu thông tin hoá đơn.

+ Chức năng thanh toán cho phép người dùng thanh toán hoá đơn của khách hàng mua sản phẩm.

+ Chức năng in hoá đơn cho phép người dùng in ra hoá đơn cho khách hàng.

* **Quản lý sản phẩm :**



*Hình 2.2.4 Sơ đồ chức năng quản lý sản phẩm*

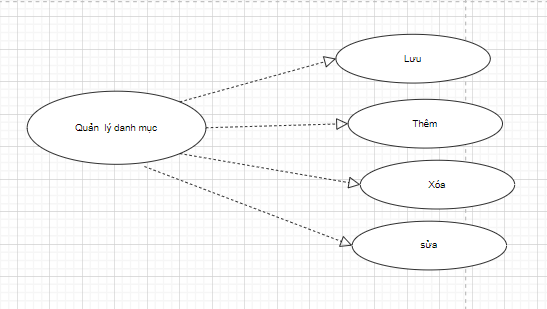
+ Chức năng thêm sản phẩm cho phép người dùng thêm sản phẩm vào kho hàng.

+ Chức năng xoá sản phẩm cho phép người dùng xoá sản phẩm trong kho.

+ Chức năng thay đổi thông tin sản phẩm (sửa) cho phép người dùng thay đổi thông tin của sản phẩm.

+ Chức năng cập nhật ảnh của sản phẩm cho phép người dùng cập nhật ảnh của sản phẩm.

* **Quản lý danh mục :**



*Hình 2.2.5 Sơ đồ chức năng quản lý loại sản phẩm*

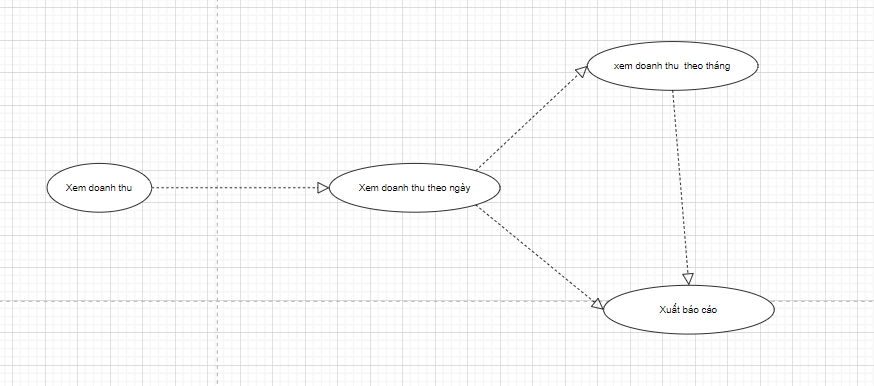
+ Chức năng thêm cho phép người dùng thêm loại sản phẩm hàng hoá nhập có trong kho.

+ Chức năng xoá cho phép người dùng xoá loại sản phẩm trong kho.

+ Chức năng sửa dùng để sửa thông tin loại sản phẩm

+ Chức năng lưu cho phép người dùng lưu thông tin người dùng đã nhập thông tin.

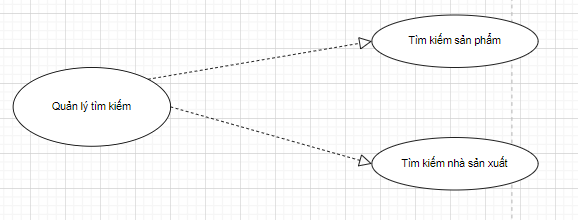
* **Quản lý thống kê doanh thu :**



*Hình 2.2.6 Sơ đồ chức năng quản lý thống kê doanh thu*

+ Chức năng thống kê cho phép người dùng thống kê doanh thu theo ngày hoặc theo tháng và xuất báo cáo.

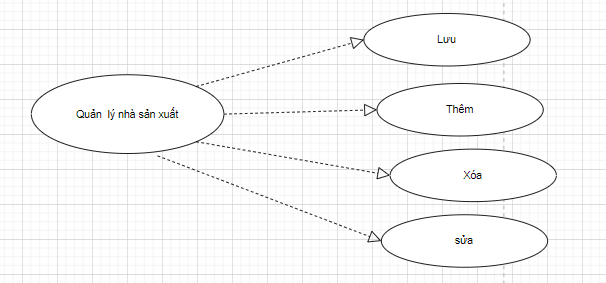
* **Quản lý tìm kiếm**



+ Chức năng tìm kiếm sản phẩm cho phép người dùng tìm kiếm các sản phẩm có trong kho hàng sản phẩm.

+ Chức năng tìm kiếm nhà sản xuất cho phép người dùng tìm kiếm các thông tin liên quan đến nhà sản xuất của sản phẩm.

* **Quản lý nhà sản xuất**



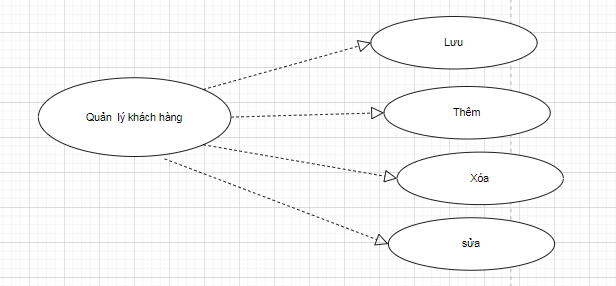
+ Chức năng thêm cho phép người dùng thêm thông tin nhà sản xuất.

+ Chức năng xoá cho phép người dùng xoá thông tin nhà sản xuất.

+ Chức năng sửa dùng để sửa thông tin nhà sản xuất.

+ Chức năng lưu cho phép người dùng lưu thông tin người dùng đã nhập thông tin của nhà sản xuất.

* **Quản lý khách hàng**



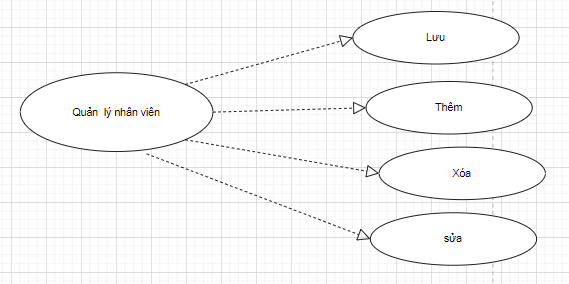
+ Chức năng thêm cho phép người dùng thêm thông tin khách hàng vào danh sách các khách hàng mua sản phẩm.

+ Chức năng xoá cho phép người dùng xoá thông tin mua hàng của khách hàng.

+ Chức năng sửa dùng để sửa thông tin mua hàng của khách hàng.

+ Chức năng lưu cho phép người dùng lưu thông tin người dùng đã nhập thông tin mua hàng của hàng.

* **Quản lý nhân viên**



+ Chức năng thêm cho phép người dùng thêm nhân viên của cửa hàng.

+ Chức năng xoá cho phép người dùng xoá nhân viên của cửa hàng khi hết hợp đồng.

+ Chức năng sửa dùng để sửa thông tin của nhân viên cửa hàng.

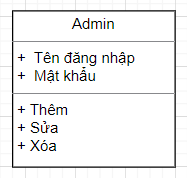
+ Chức năng lưu cho phép người dùng lưu thông tin người dùng đã nhập thông tin của nhân viên cửa hàng.

# **PHẦN 3 : THIẾT KẾ HỆ THỐNG**

## **I. THIẾT KẾ MÔ HÌNH LỚP**

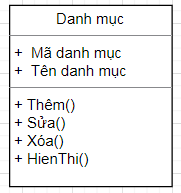
### **I.1 Mô hình lớp**

a. Lớp “**Admin**”.



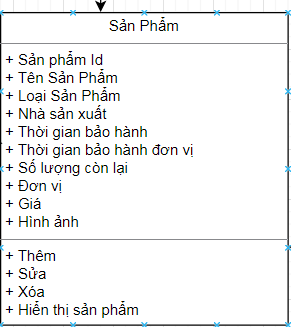
*Hình 3.1.1 Mô hình lớp Admin*

b. Lớp “**Loại danh mục**”



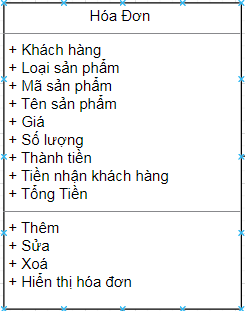
*Hình 3.1.2 Mô hình lớp Danh mục*

c. Lớp “**Sản phẩm**”



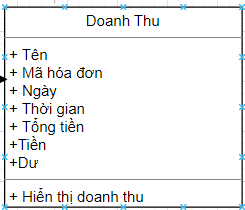
*Hình 3.1.3 Mô hình lớp Sản phẩm*

d. Lớp “**Hóa đơn**”



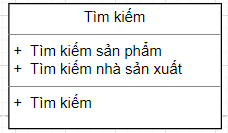
*Hình 3.1.4 Mô hình lớp Hoá đơn*

e. Lớp “**Doanh thu**”



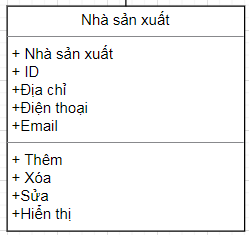
*Hình 3.1.5 Mô hình lớp thống kê doanh thu*

f. Lớp **“Tìm kiếm”**

****

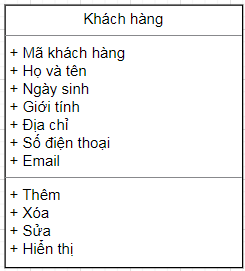
*Hình 3.1.6 Mô hình lớp Tìm kiếm*

g. Lớp **“Nhà sản xuất”**

****

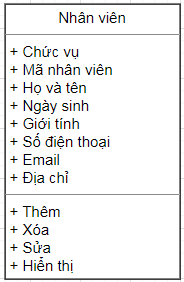
*Hình 3.1.7 Mô hình lớp Nhà sản xuất*

h. Lớp **“Khách hàng”**

****

*Hình 3.1.8 Mô hình lớp khách hàng*

i. Lớp **“Nhân viên”**



*Hình 3.1.9 Mô hình lớp Nhân viên*

**I.2 Mối quan hệ hợp thành kết tập**

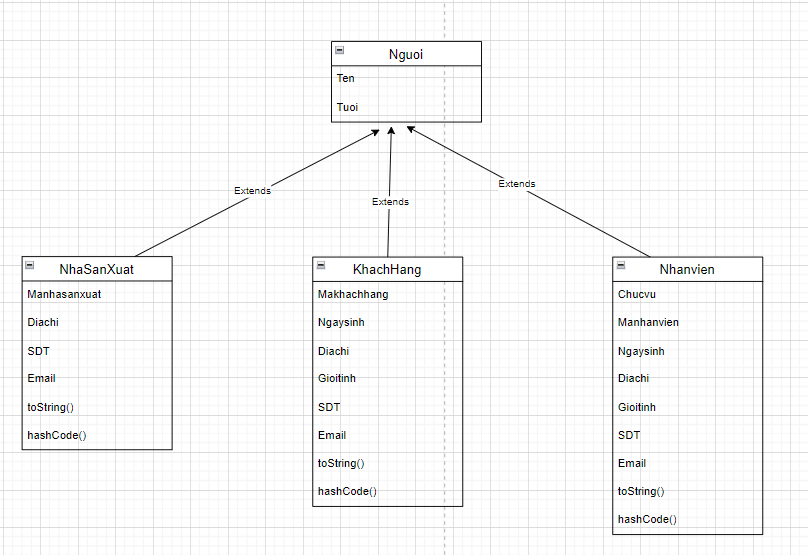
- “Loại sản phẩm” và “Sản phẩm”: Mỗi loại sản phẩm có thể chứa một hoặc nhiều sản phẩm

- “Nhà sản xuất” và “Sản phẩm” : Mỗi nhà sản xuất chứa nhiều sản phẩm

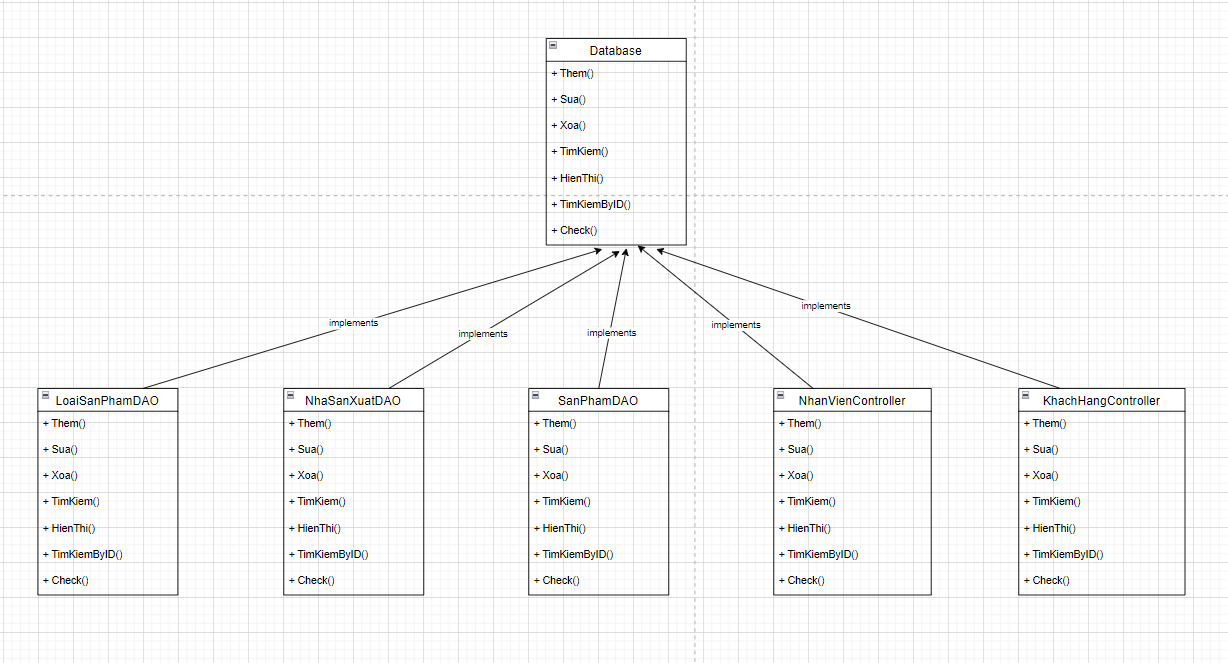
-“Sản phẩm” và “Hóa đơn” : Mỗi hóa đơn có thể chứa nhiều sản phẩm

-“Nhân viên” và “Sản phẩm” : Mỗi nhân viên có thể quản lý nhiều sản phẩm

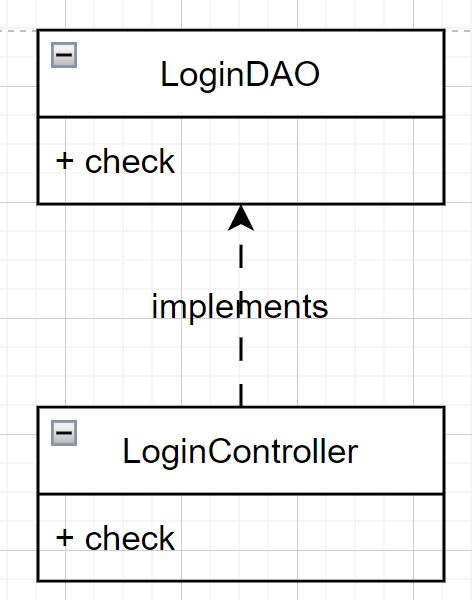
Mối quan hệ interface:

****

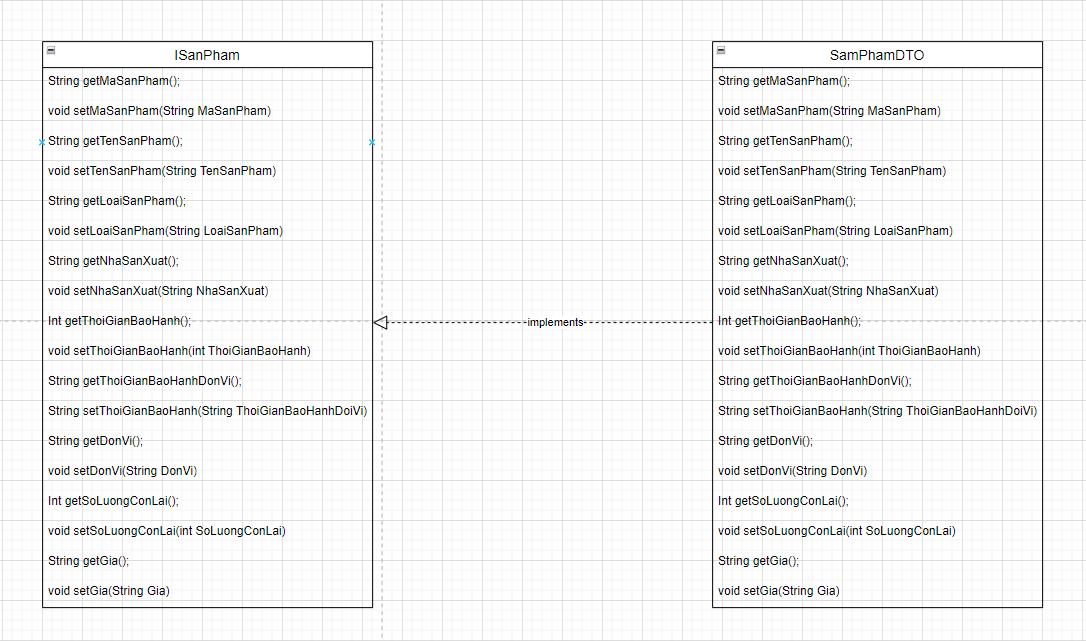
*Hình 3.1.10 Quan hệ kế thừa các lớp*

**

*Hình 3.1.11 Quan hệ interface và các lớp DAO*

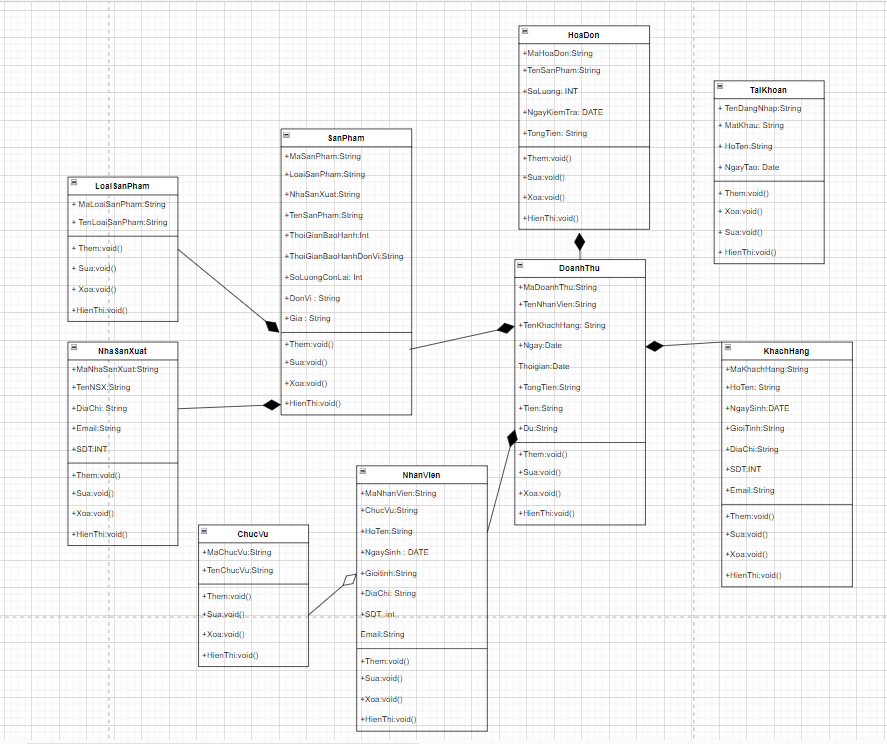
****

*Hình 3.1.12 Interface LoginDAO và lớp LoginController*

**

*Hình 3.1.13. Quan hệ giữa interface va lớp DTO*

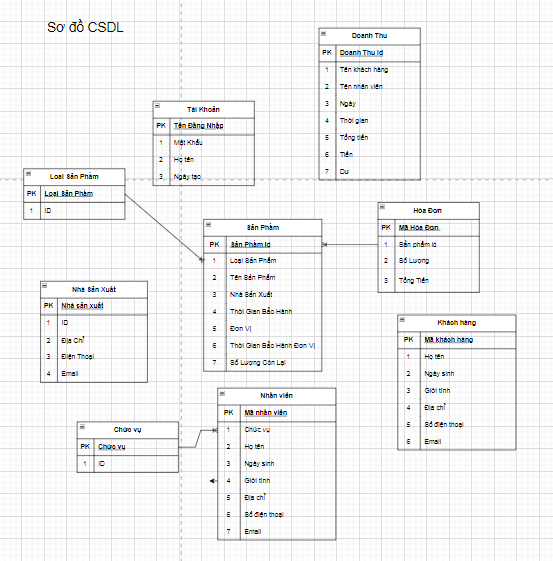
**\*Sơ đồ lớp tổng quát:**



*Hình 3.1.14. Sơ đồ lớp tổng quát*

## **II. THIẾT KẾ CƠ SỞ DỮ LIỆU**

### **II.1 Sơ đồ cơ sở dữ liệu**

****

*Hình 3.2.1 Sơ đồ cơ sở dữ liệu*

### **II.2 Nhập dữ liệu**

\* Bảng **Tài Khoản**

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Ý nghĩa |
| 1 | TenDangNhap | Nvarchar | Tên đăng nhập |
| 2 | MatKhau | Nvarchar | Mật khẩu |
| 3 | HoTen | Nvarchar | Họ tên |
| 4 | NgayTao | Date | Ngày tạo |

*Bảng 3.2.1 : Nhập dữ liệu bảng sản phẩm*

**\*** Bảng **Sản phẩm**

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Ý nghĩa |
| 1 | ID | Nvarchar | Mã sản phẩm |
| 2 | LoaiSanPham | Nvarchar | Loại sản phẩm , khoá phụ |
| 3 | TenSanPham | Nvarchar | Tên sản phẩm, khoá chính |
| 4 | NhaSanXuat | Nvarchar | Nhà sản xuất |
| 5 | ThoiGianBaoHanh | Int | Thời gian bảo hành |
| 6 | ThoiGianBaoHanhDonVi | Nvarchar | Đơn vị thời gian bảo hành |
| 7 | SoLuongConLai | Int | Số lượng còn lại |
| 8 | DonVi | Nvarchar | Đơn vị |
| 9 | Gia | Nvarchar | Giá |

*Bảng 3.2.2 : Nhập dữ liệu bảng sản phẩm*

\* Bảng **Loại Sản Phẩm**

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Ý nghĩa |
| 1 | LoaiSanPham | Nvarchar | Loại Sản Phẩm , Khoá chính |
| 2 | ID | Nvarchar | Mã ID của loại sản phẩm |

*Bảng 3.2.3 : Nhập dữ liệu bảng loại sản phẩm*

\* Bảng **Nhà Sản Xuất**

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Ý nghĩa |
| 1 | ID | Nvarchar | Mã ID của nhà sản xuất |
| 2 | TenNSX | Nvarchar | Tên nhà sản xuất , Khoá chính |
| 3 | DiaChi | Nvarchar | Địa chỉ |
| 4 | DienThoai | Nvarchar | Số điện thoại nhà sản xuất |
| 5 | Email | Nvarchar | Email của nhà sản xuất |

*Bảng 3.2.4 : Nhập dữ liệu bảng nhà sản xuất*

\* Bảng **Hoá Đơn**

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Ý nghĩa |
| 1 | TenSanPham | Nvarchar | Tên sản phẩm |
| 2 | MaHoaDon | Nvarchar | Mã hoá đơn , khoá chính |
| 3 | SoLuong | Int | Số lượng |
| 4 | TongTien | Nvarchar | Tổng tiền |

*Bảng 3.2.5 : Nhập dữ liệu bảng hoá đơn*

\* Bảng **Chức Vụ**

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Ý nghĩa |
| 1 | ID | Nvarchar | Mã chức vụ |
| 2 | ChucVu | Nvarchar | Chức vụ, khoá chính |

*Bảng 3.2.6 : Nhập dữ liệu bảng chức vụ*

\* Bảng **Doanh Thu**

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Ý nghĩa |
| 1 | ID | Int indentify | Mã ID doanh thu , khoá chính |
| 2 | TenNV | Nvarchar | Tên nhân viên bán hàng |
| 3 | TenKH | Nvarchar | Tên khách hàng |
| 4 | Ngay | Nvarchar | Ngày tạo hoá đơn |
| 5 | ThoiGian | Nvarchar | Thời gian |
| 6 | TongTien | Nvarchar | Tổng tiền |
| 7 | Tien | Nvarchar | Tiền |
| 8 | Du | Nvarchar | Dư |

*Bảng 3.2.7 : Nhập dữ liệu bảng doanh thu*

\* Bảng **Nhân Viên**

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Ý nghĩa |
| 1 | ChucVu | Nvarchar | Chức vụ |
| 2 | MaNV | Nvarchar | Mã nhân viên, khoá chính |
| 3 | HoTen | Nvarchar | Họ tên nhân viên |
| 4 | NgaySinh | Date | Ngày sinh nhân viên |
| 5 | GioiTinh | Nvarchar | Giới tính nhân viên |
| 6 | DiaChi | Nvarchar | Địa chỉ nhân viên |
| 7 | SDT | Nvarchar | Số điện thoại nhân viên |
| 8 | Email | Nvarchar | Email nhân viên |

*Bảng 3.2.8 : Nhập dữ liệu bảng nhân viên*

\* Bảng **Khách Hàng**

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Ý nghĩa |
| 1 | MaKH | Nvarchar | Mã khách hàng |
| 2 | HoTen | Nvarchar | Họ tên khách hàng |
| 3 | NgaySinh | Date | Ngày sinh khách hàng |
| 4 | GioiTinh | Nvarchar | Giới tính nhân viên |
| 5 | DiaChi | Nvarchar | Địa chỉ nhân viên |
| 6 | SDT | Nvarchar | Số điện thoại nhân viên |
| 7 | Email | Nvarchar | Email nhân viên |

*Bảng 3.2.9 : Nhập dữ liệu bảng khách hàng*

# **PHẦN 4 : CÀI ĐẶT CHƯƠNG TRÌNH**

## **I. MỘT SỐ CÀI ĐẶT CHÍNH**

### **I.1. Các công cụ**

**I.1.1. Netbean IDE**

**\* Khái niệm**

NetBeans IDE là một công cụ hỗ trợ lập trình viết mã code miễn phí được cho là tốt nhất hiện nay, được sử dụng chủ yếu cho các lập trình viên phát triển Java tuy nhiên phần mềm có dung lượng khá là nặng dành cho các máy cấu hình có RAM, CPU tương đối cao để vận hành.

NetBeans IDE là môi trường phát triển tích hợp và cực kỳ cần thiết cho các lập trình viên, công cụ này có thể hoạt động tốt với rất nhiều nền tảng hệ điều hành khác nhau như Linux, Windows, MacOS,... là một mã nguồn mở cung cấp các tính năng cần thiết nhất nhăm tại ra các ứng dụng web, thiết bị di động, desktop.

Để có thể cài đặt được **NetBeans IDE** các bạn phải cài đặt và cấu hình sẵn JDK trong máy thì mới cài đặt được. NetBean IDE được chia thành nhiều gói khác nhau để người dùng có thể tải về từng phần, tiết kiệm dung lượng, tùy theo nhu cầu của mình.

**\* Tính năng**

* Hỗ trợ nhiều loại ngôn ngữ lập trình.
* Chỉnh sửa mã nguồn thông minh.
* Giao diện trực quan, dễ thao tác, sử dụng.
* Là một công cụ lập trình phần mềm máy tính hoặc phần mềm trên các thiết bị di động.
* Gỡ lỗi mạng nội bộ và từ xa.
* Thử nghiệm tính năng xây dựng giao diện đồ họa.
* Tính năng QuickSearch (Tìm kiếm nhanh), tự động biên dịch, hỗ trợ các Framework cho website, trình ứng dụng máy chủ GlassFish và cơ sở dữ liệu.

**I.1.2. SQLServer Manager Studio**

SSMS là một ứng dụng phần mềm thiết kế bởi Microsoft, ra mắt lần đầu năm 2005. Ứng dụng này cho phép lập trình viên cấu hình, quản lý và quản trị bộ máy cơ sở dữ liệu (database engine) SQL Server. SSMS phổ biến và được sử dụng rộng rãi trong cộng đồng lập trình viên và quản trị viên cơ sở dữ liệu bởi những ưu điểm:

* Miễn phí
* Trải nghiệm người dùng tốt
* Nhiều lựa chọn add-in
* Dễ cài đặt

**I.1.3. JDK**

[Java Development Kit](https://bizflycloud.vn/tin-tuc/jdk-la-gi-2018121318321223.htm) (JDK) là một trong ba gói công nghệ cốt lõi được sử dụng trong lập trình Java, cùng với JVM (Máy ảo Java - Java Virtual Machine) và JRE (Java Runtime Environment - Môi trường Java Runtime). Việc phân biệt giữa ba công nghệ này, cũng như hiểu được cách chúng kết nối với nhau là rất quan trọng.

* JVM là thành phần Java nền tảng để chạy các chương trình.
* JRE là phần on-disk của Java, phần tạo ra JVM.
* JDK cho phép các developer tạo các chương trình Java, trong đó các chương trình có thể được JVM và JRE xử lý và chạy.

Các developer khi mới sử dụng Java thường dễ nhầm lẫn Java Development Kit với Java Runtime Environment. Để phân biệt, ta cần nhớ rằng JDK là một gói các công cụ phát triển phần mềm dựa trên Java, trong khi JRE là gói công cụ sử dụng để chạy Java code.

JRE có thể được sử dụng như một thành phần độc lập để chạy các chương trình Java, dù vậy, JRE vẫn là một phần của JDK. JDK đòi hỏi bộ công cụ JRE vì chạy các chương trình Java là một phần của việc phát triển chúng.

**I.1.4. JDBC**

**\* Khái niệm**

JDBC, là viết tắt của **J**ava **D**ata**b**ase **C**onnectivity, là một Java API chuẩn để kết nối giữa ngôn ngữ lập trình Java và các cơ sở dữ liệu đa dạng. Sử dụng JDBC, bạn có thể thực hiện nhiều tác vụ đa dạng khi làm việc với cơ sở dữ liệu như tạo, xóa cơ sở dữ liệu; tạo và thực thi các lệnh SQL hoặc MySQL; tạo, xóa các bản ghi; …

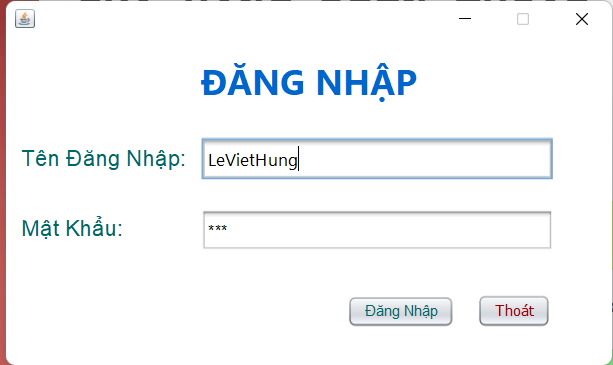
**\* Các thành phần**

* DriverManager: Lớp này quản lý các Database Driver. Ánh xạ các yêu cầu kết nối từ ứng dụng Java với Data driver thích hợp bởi sử dụng giao thức kết nối phụ.
* Driver: Interface này xử lý các kết nối với Database Server. Hiếm khi, bạn tương tác trực tiếp với các đối tượng Driver này. Thay vào đó, bạn sử dụng các đối tượng DriverManager để quản lý các đối tượng kiểu này.
* Connection: Đối tượng Connection biểu diễn ngữ cảnh giao tiếp. Interface này chứa nhiều phương thức đa dạng để tạo kết nối với một Database.
* Statement: Bạn sử dụng các đối tượng được tạo từ Interface này để đệ trình các lệnh SQL tới Database. Ngoài ra, một số Interface kết thừa từ nó cung chấp nhận thêm các tham số để thực thi các thủ tục đã được lưu trữ.
* ResultSet: Các đối tượng này giữ dữ liệu được thu nhận từ một Database sau khi bạn thực thi một truy vấn SQL. Nó nóng vai trò như một Iterator để cho phép bạn vọc qua dữ liệu của nó.
* SQLException: Lớp này xử lý bất cứ lỗi nào xuất hiện trong khi làm việc với Database.

## **II. GIAO DIỆN CÁC CHỨC NĂNG CHÍNH**

### **II.1. Các giao diện của hệ thống**

* **Giao diện đăng nhập**

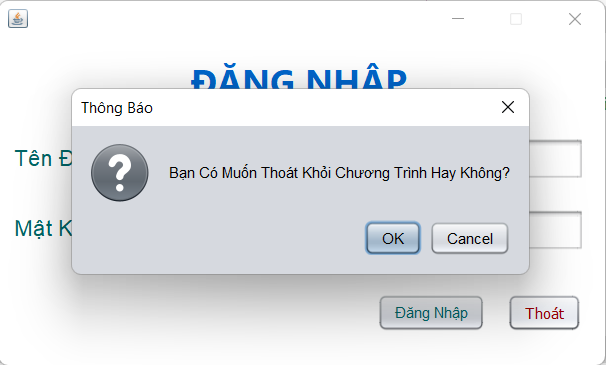


- Giao diện gồm phần nhập “Tên Đăng Nhập”, “Mật Khẩu” và 2 nút “đăng nhập”, “thoát”.

+ Người dùng nhập tên đăng nhập : LeVietHung và mật khẩu : 123

+ Sau đó bấm nút Đăng nhập hệ thống sẽ tự động đăng nhập vào hệ thống với giao diện chính

+ Người dùng không muốn sử dụng hệ thống có thể nhấp nút Thoát. Chương trình hỏi bạn có muốn thoát khỏi chương trình không?



+ Nếu người dùng muốn rời hệ thống nhấn nút OK , nếu người dùng muốn tiếp tục đăng nhập nhấn nút Cancel.

* **Giao diện Chính**



- Giao diện gồm các chức năng :

+ Danh mục : người dùng khi bấm vào sẽ được chuyển đến giao diện Danh mục.

+ Sản phẩm : người dùng khi bấm vào sẽ được chuyển đến giao diện Sản phẩm.

+ Hoá đơn : người dùng khi bấm vào sẽ được chuyển đến giao diện Hoá đơn.

+ Thống kê : người dùng khi bấm vào sẽ được chuyển đến giao diện Thống kê.

+ Tìm kiếm : người dùng khi bấm vào sẽ được chuyển đến giao diện Tìm kiếm.

+ Nhà sản xuất : người dùng khi bấm vào sẽ được chuyển đến giao diện Nhà sản xuất.

+ Khách hàng : người dùng khi bấm vào sẽ được chuyển đến giao diện Khách hàng.

+ Nhân viên : người dùng khi bấm vào sẽ được chuyển đến giao diện Nhân viên.



+ Đăng xuất : nếu người dùng muốn thoát khỏi chương trình nhấn nút OK chương trình sẽ được đăng xuất

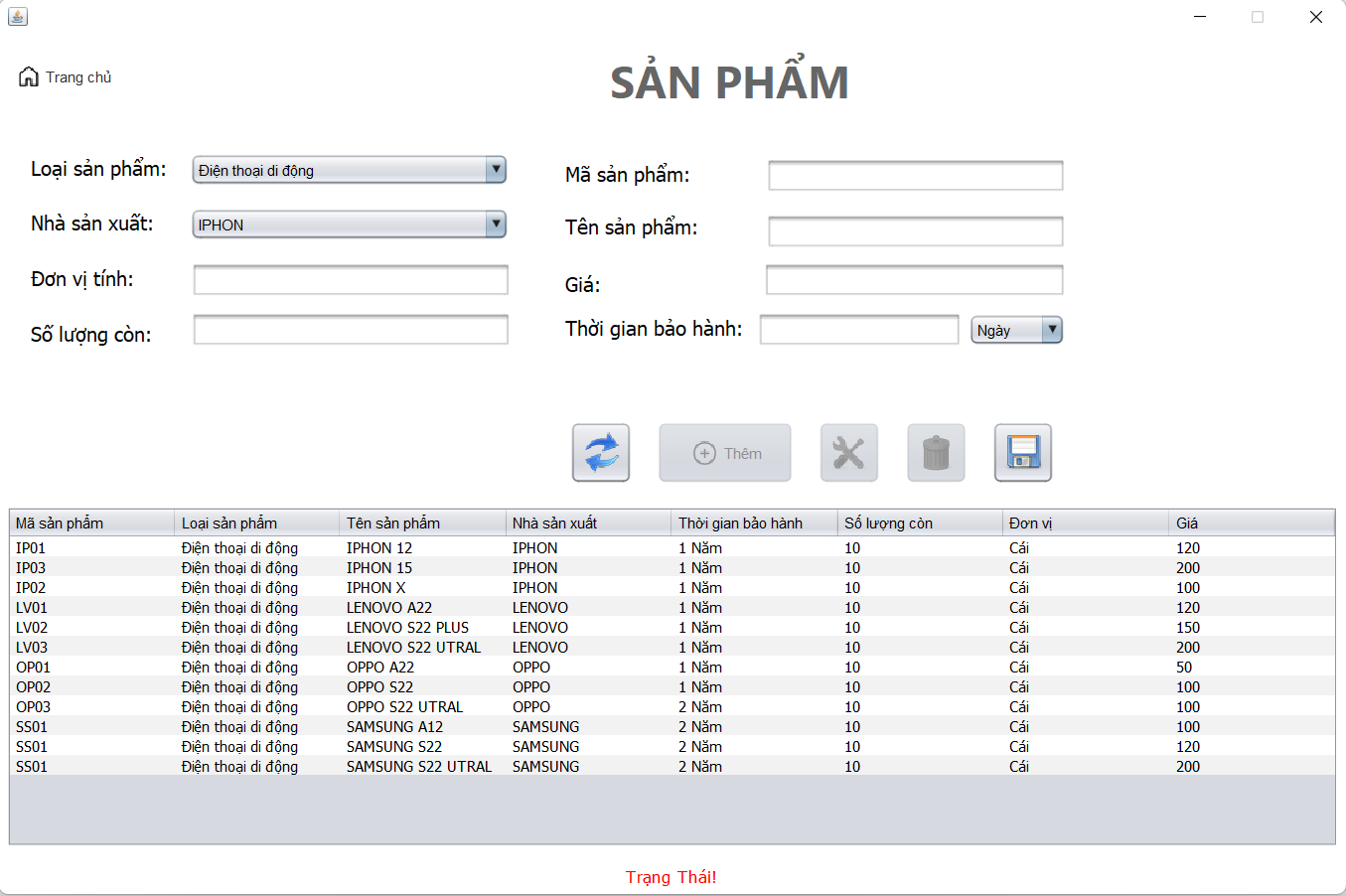
* **Giao diện Danh mục**



- Gồm các chức năng thêm, sửa, xoá, lưu và hiển thị Mã danh mục, Tên danh mục.

- Nếu người dùng muốn thoát khỏi giao diện danh mục bấm vào góc trái màn hình có icon Trang chủ , nhấp vào đó và trở về trang chủ.

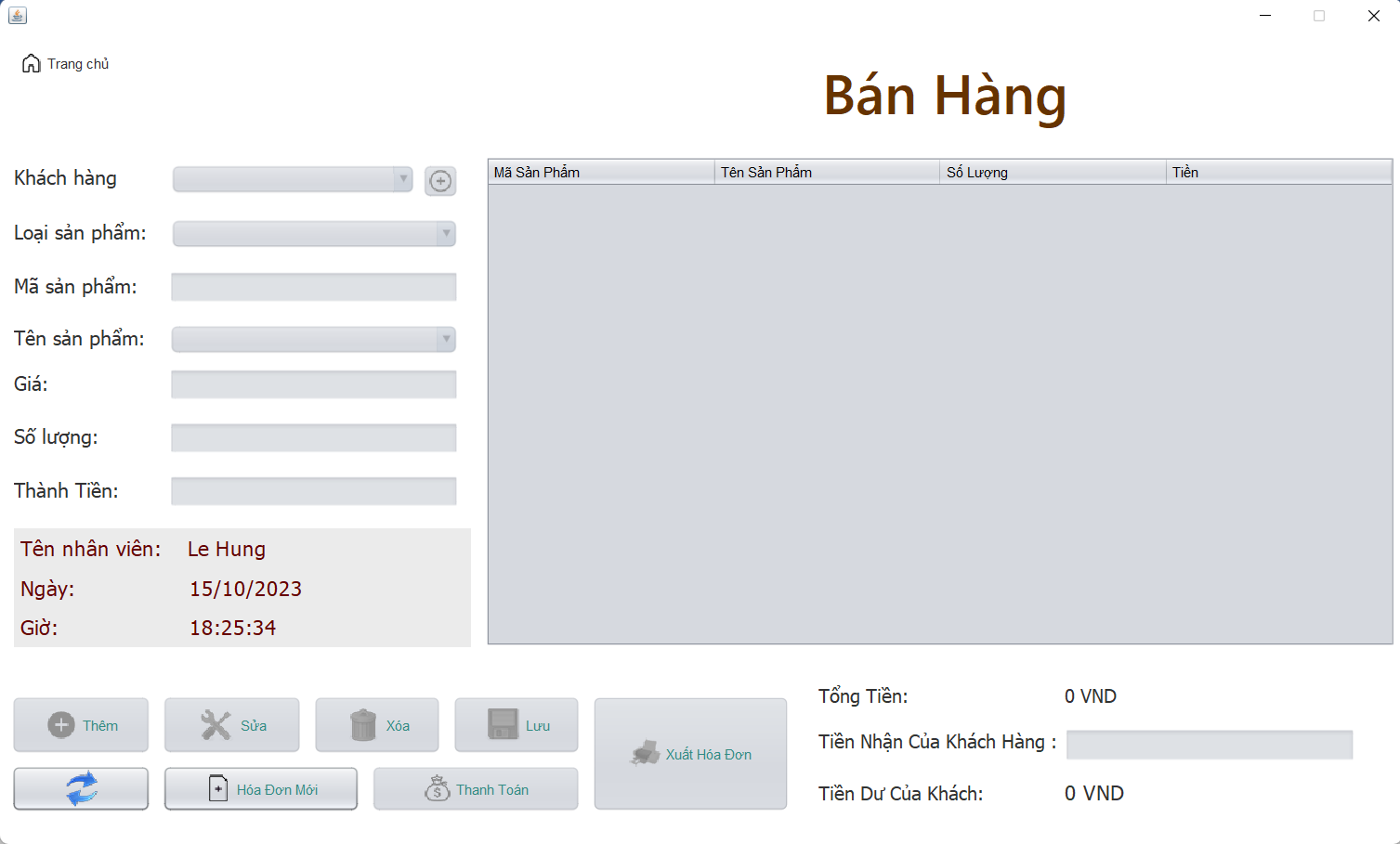
* **Giao diện Sản phẩm**



+ Gồm các nút chức năng thêm, sửa, xóa, lưu các thông tin liên quan tới sản phẩm.

+ Nếu người dùng muốn thoát khỏi giao diện sản phẩm bấm vào góc trái màn hình có icon Trang chủ , nhấp vào đó và trở về trang chủ.

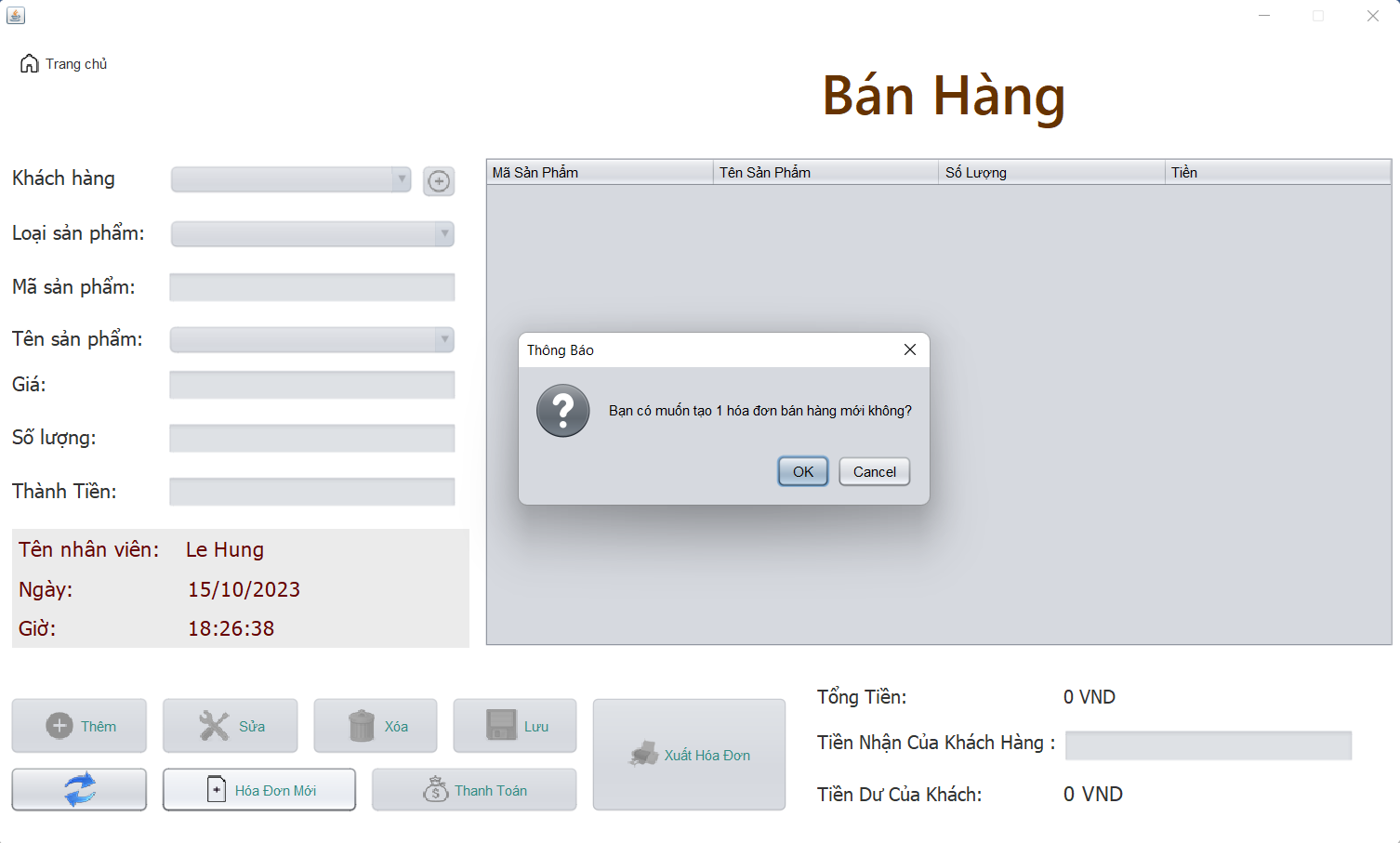
* **Giao diện Hóa đơn**



+ Giao diện gồm các nút chức năng thêm, sửa , xóa , lưu , tạo hóa đơn mới, thanh toán và xuất hóa đơn.

+ Giao diện có 1 bảng hiển thị dữ liệu về hóa đơn.

+ Nếu người dùng muốn thoát khỏi giao diện hóa đơn bấm vào góc trái màn hình có icon Trang chủ , nhấp vào đó và trở về trang chủ.



+ Người dùng muốn tạo hóa đơn mới nhấp vào nút Hóa Đơn Mới , chương trình hiện thông báo Bạn có muốn tạo 1 hóa đơn bán hàng mới không? . Nếu người dùng muốn tạo nhấp OK để tiếp tục tạo hóa đơn mới , nếu không thì nhấp Cancel.

* **Giao diện Thống kê doanh thu**

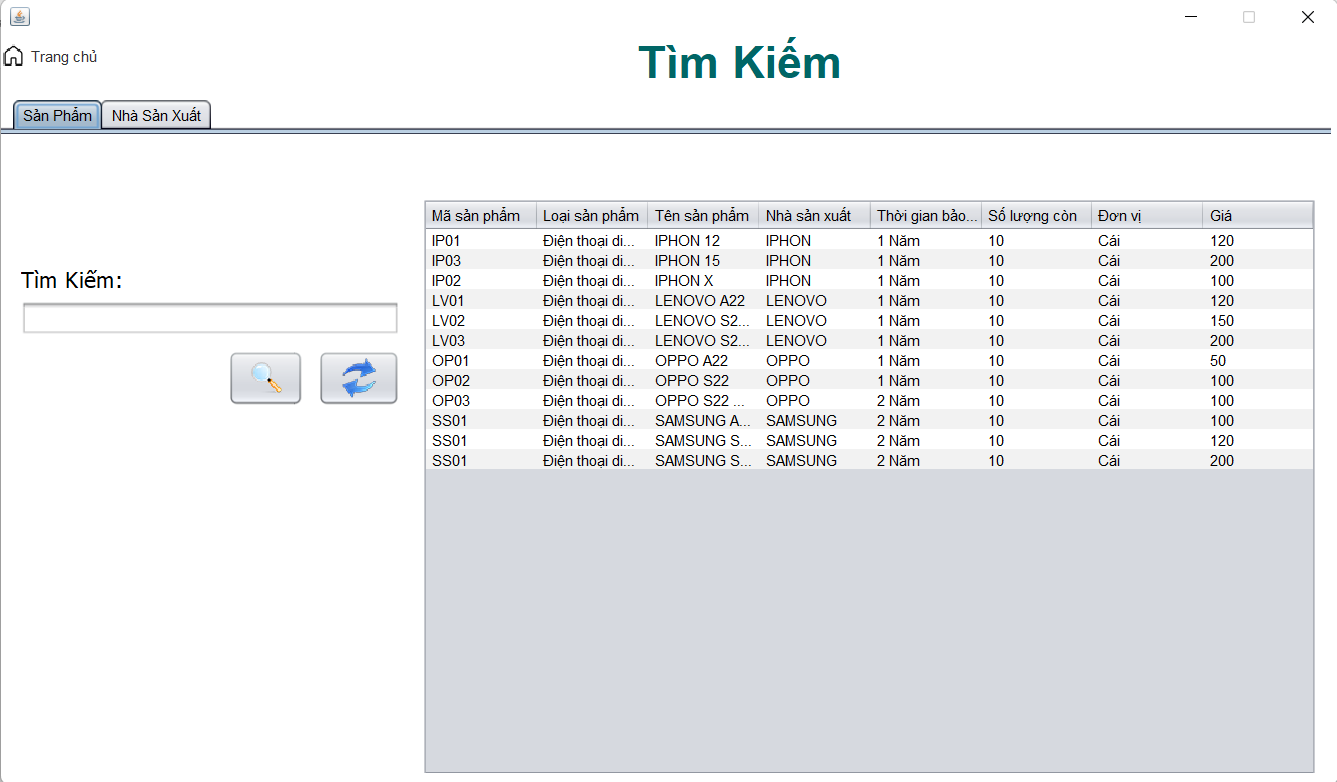


+ Giao diện có chức năng thống kê cho người dùng thống kê doanh thu từ ngày bao nhiêu đến ngày bao nhiêu.

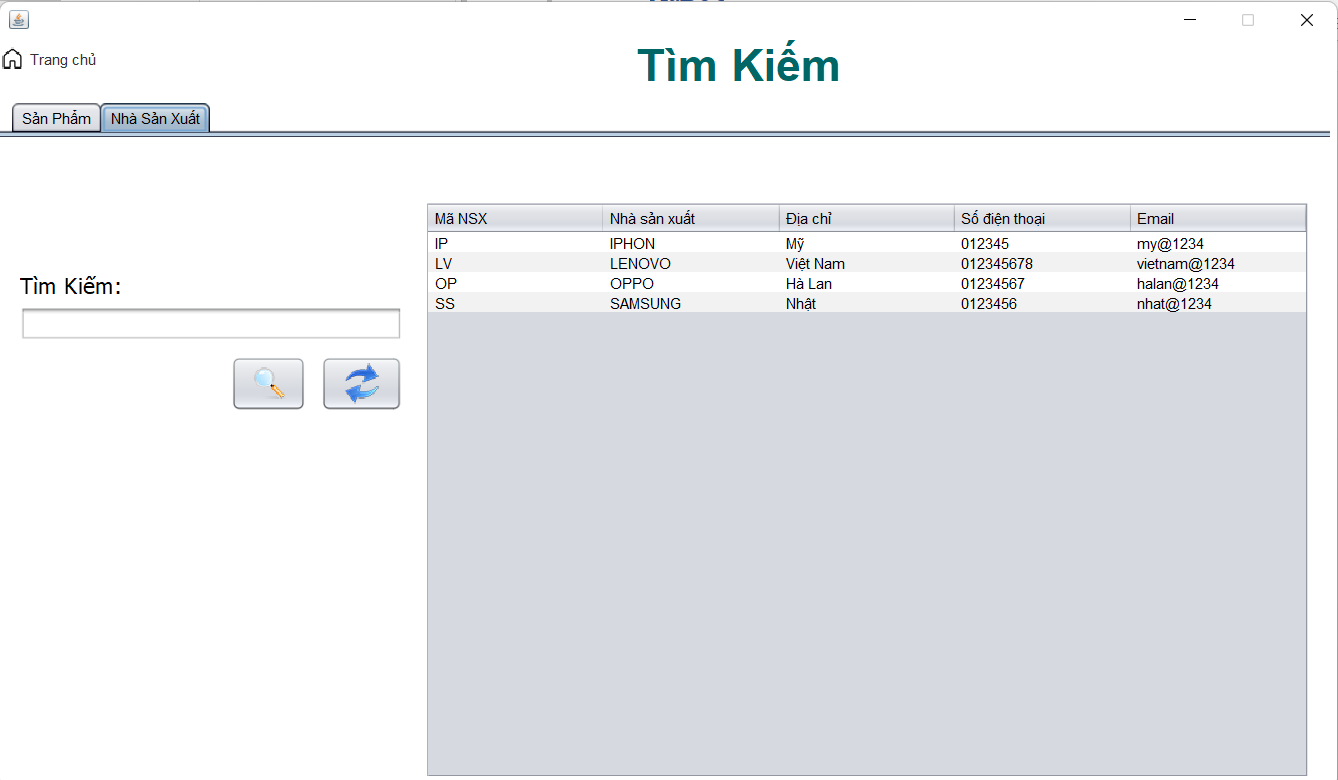
+ Giao diện có bảng hiển thị các hóa đơn do các nhân viên bán hàng bán được.

+ Nếu người dùng muốn thoát khỏi giao diện thống kê doanh thu bấm vào góc trái màn hình có icon Trang chủ , nhấp vào đó và trở về trang chủ.

* **Giao diện Tìm kiếm**



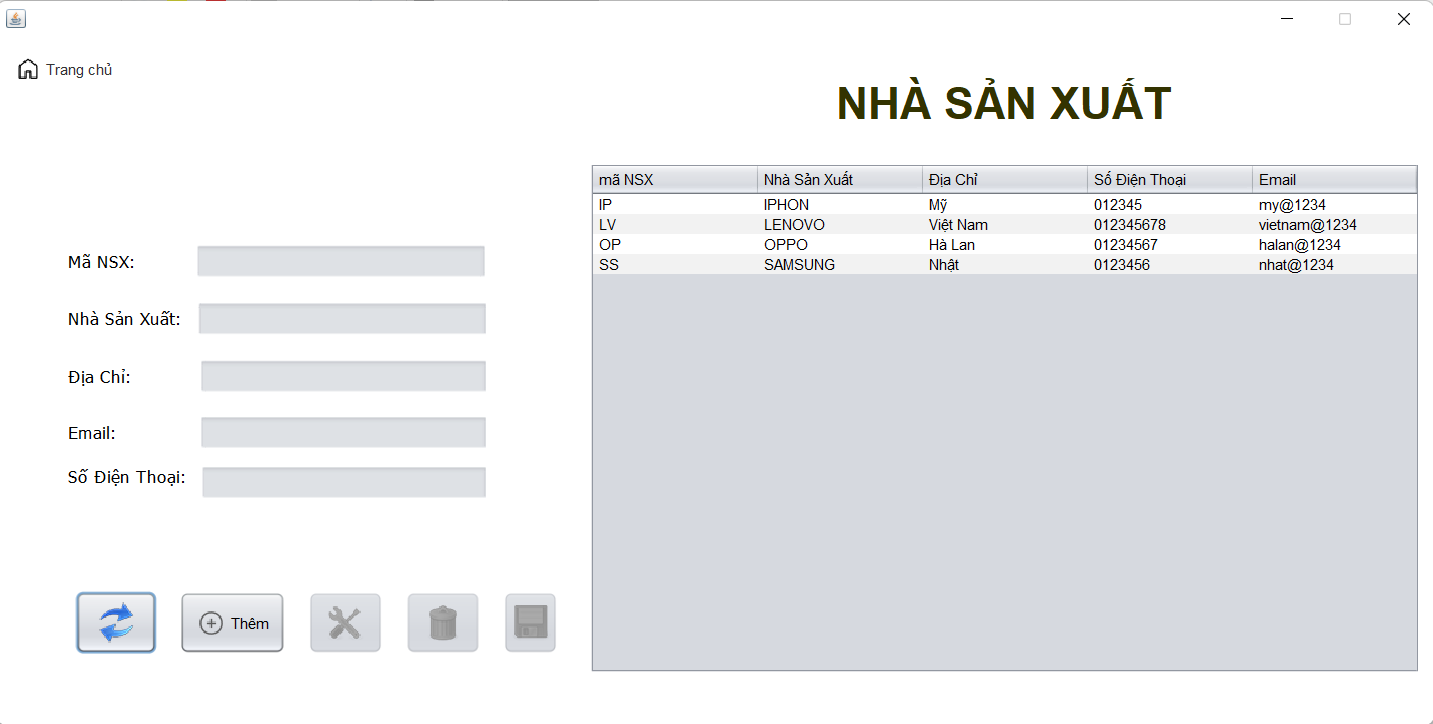
+ Giao diện tìm kiếm sản phẩm có chức năng tìm kiếm thông tin sản phẩm và hiển thị các sản phẩm muốn tìm kiếm.



+ Giao diện tìm kiếm thông tin nhà sản xuất có chức năng tìm kiếm thông tin nhà sản xuất và hiển thị các nhà sản xuất muốn tìm kiếm.

+ Nếu người dùng muốn thoát khỏi giao diện tìm kiếm bấm vào góc trái màn hình có icon Trang chủ , nhấp vào đó và trở về trang chủ.

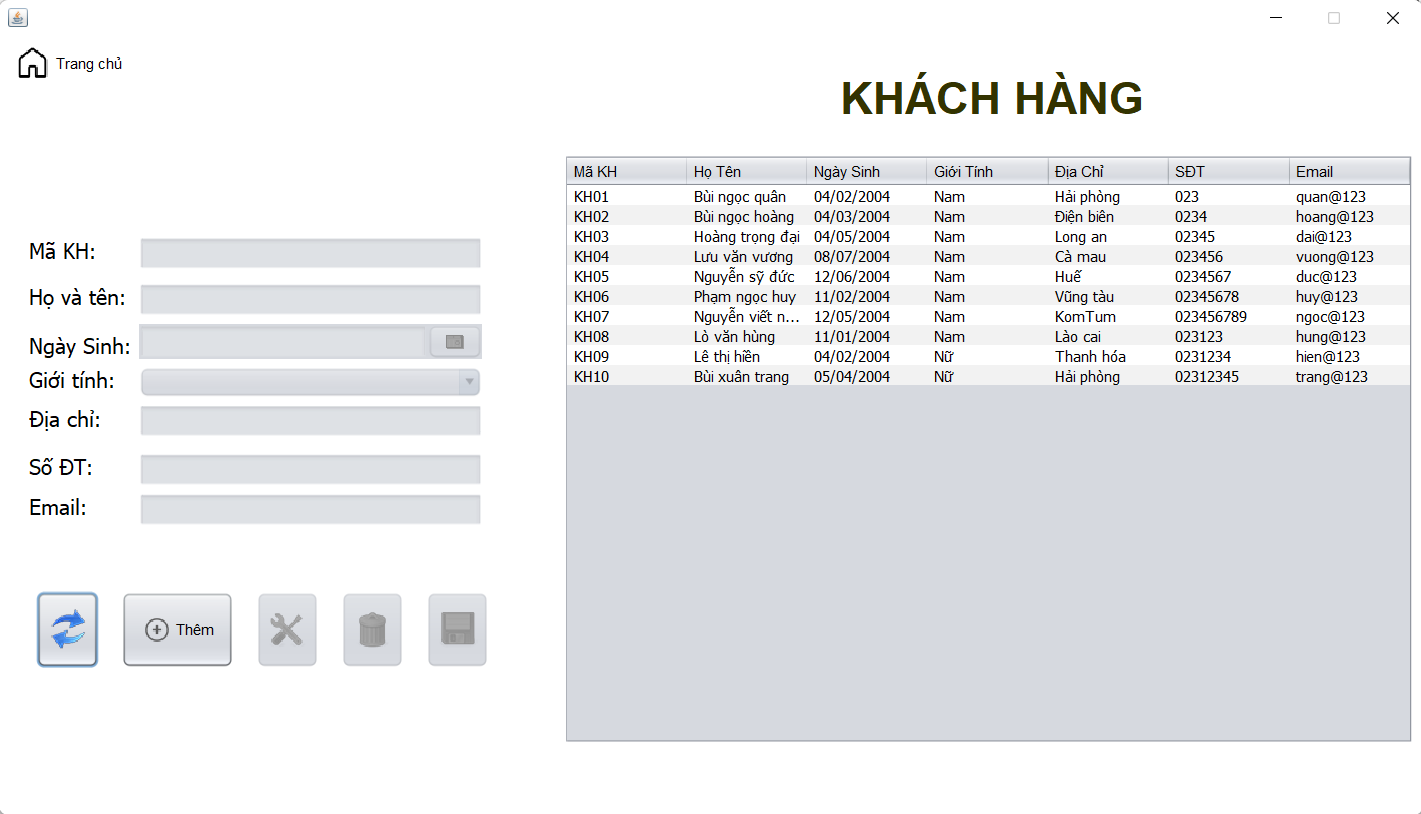
* **Giao diện Nhà sản xuất**



+ Giao diện nhà sản xuất có các chức năng thêm sửa xóa lưu để người dùng có thể thao thác nhập thông tin nhà sản xuất vào bảng danh sách hiển thị các nhà sản xuất.

+ Nếu người dùng muốn thoát khỏi giao diện nhà sản xuất bấm vào góc trái màn hình có icon Trang chủ , nhấp vào đó và trở về trang chủ.

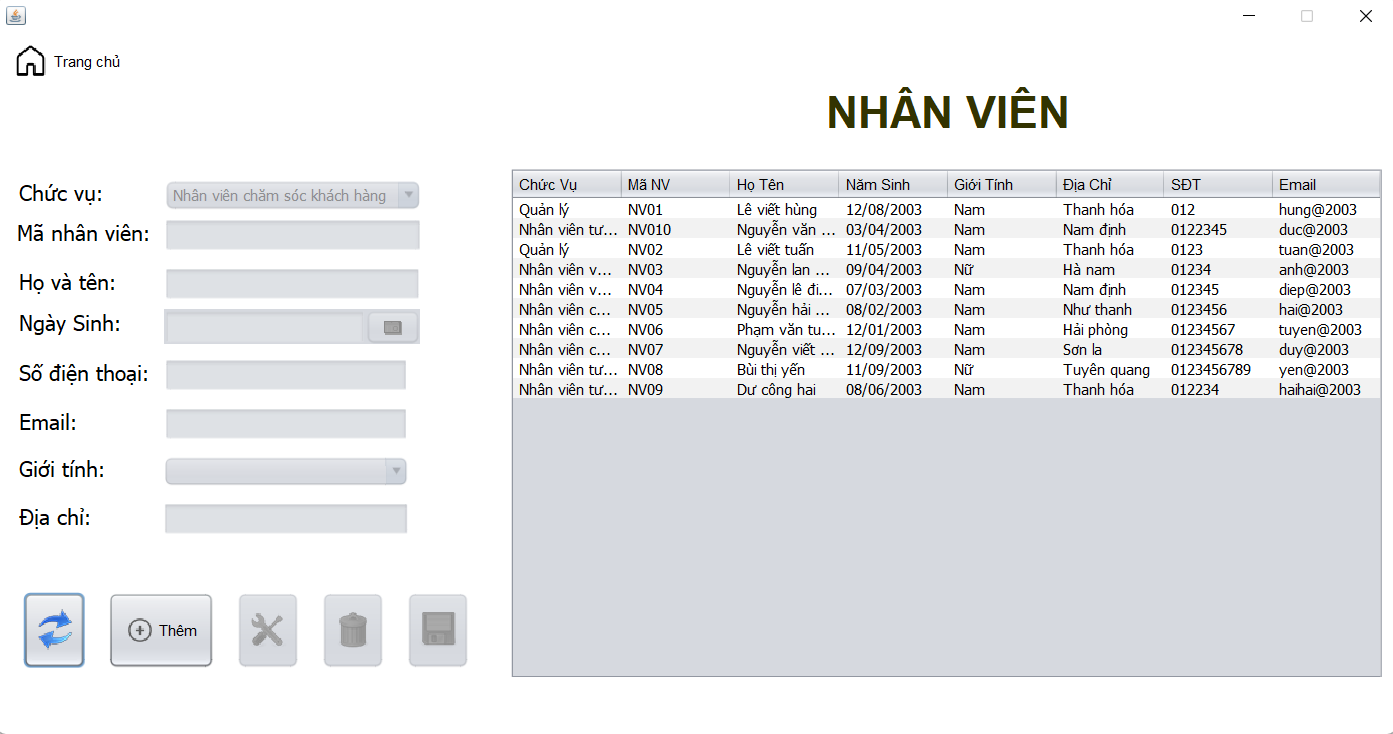
* **Giao diện Khách hàng**



+ Giao diện nhà sản xuất có các chức năng thêm sửa xóa lưu để người dùng có thể thao thác nhập thông tin khách hàng vào bảng danh sách hiển thị các khách hàng.

+ Nếu người dùng muốn thoát khỏi giao diện khách hàng bấm vào góc trái màn hình có icon Trang chủ , nhấp vào đó và trở về trang chủ.

* **Giao diện Nhân Viên**



+ Giao diện nhà sản xuất có các chức năng thêm sửa xóa lưu để người dùng có thể thao thác nhập thông tin nhân viên vào bảng danh sách hiển thị các nhân viên.

+ Nếu người dùng muốn thoát khỏi giao diện nhân viên bấm vào góc trái màn hình có icon Trang chủ , nhấp vào đó và trở về trang chủ.

# **PHẦN 5 : KẾT LUẬN**

## Kết quả đạt được

Trong quá trình xây dựng Ứng dụng quản lý Cửa Hàng Điện Thoại, chúng em đã đạt được những kết quả như: đã học được cách lập trình với ngôn ngữ java, lập trình hướng đối tượng và cũng học được cách dùng những phương thức và thuộc tính của lập trình hướng đối tượng, biết cách kết nối giữa ngôn ngữ java và cơ sở dữ liệu thông qua JDBC và ứng dụng Netbean.

Tuy nhiên, ứng dụng còn một số hạn chế cần được cải thiện. Điều này bao gồm việc chưa tối ưu hóa hoàn toàn về hiệu suất và giao diện người dùng cần được thiết kế tốt hơn để mang lại trải nghiệm người dùng tốt hơn.

Với nỗ lực của bản thân, nhóm đã cố gắng hoàn thành yêu cầu đề tài. Do thời gian và năng lực có hạn nên ứng dụng mà nhóm xây dựng mới chỉ đi sâu vào các chức năng chính như thêm, xóa, sửa, tìm kiếm, thống kê. Nhóm đề tài hướng phát triển ứng dụng trở thành một ứng dụng có thể demo được trên nhiều hệ điều hành và được ứng dụng trên những hệ thống vừa và lớn, cung cấp đầy đủ những tính năng cần thiết và có giao diện thân thiện dễ sử dụng.

## Hướng phát triển

Khảo sát thêm thực tế để nâng cao chất lượng hệ thống quản lý bán điện thoại di động. Bổ sung thêm phần đánh giá dịch vụ cho khách hàng nhằm nắm bắt được nhu cầu thị hiếu và đồng thời nâng cao chất lượng dịch vụ để mang đến trải nghiệm tốt nhất cho khách hàng. Đa dạng hóa các mẫu mã điện thoại với nhiều kiểu dáng, chức năng khác nhau để với các phân khúc giá theo nhóm sản phẩm từ thấp đến cao nhằm giúp khách hàng có nhiều lựa chọn hơn và phù hợp với thu nhập cá nhân của mỗi khách hàng. Mở rộng thêm nhiều chi nhánh hơn, ví dụ như 1 huyện ta sẽ có 1 chi nhánh, nhờ vào đó khách hàng có thể thuận tiện trong quá trình đi lại, cũng như là dễ dàng trong việc sửa chữa, bảo trì điện thoại.

# **TÀI LIỆU THAM KHẢO**

[1] Đoàn Văn Ban, Lập trình hướng tượng với Java, NXB Khoa học Kỹ thuật.

[2] C. Thomas Wu. McGraw-Hill Inc, An Introduction to Object-Oriented

Programming with Java.

[3] J.N. Patterson Hume and Christine Stephenson, Introduction to Programming in

Java, first edition.

[4] Elliot B. Koffman and Paul A.T. Wolfgang. John Wiley &amp; Sons Inc, Objects,

Abstractions, Data Structures and Designe using Java, 2011.

[5] Cay S. Horstmann, Big Java, 7th edition, Wiley.

[6] Bruce Eckel, Thinking in Java, 4th edition, Prentice Hall.

[7] Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides, Addision-Wesley

Professional, Design Patterns: Elements of Reusable Object-Oriented Software.

[8] The Java Tutorials, https://docs.oracle.com/javase/tutorial/

[9] Introduction to Programming in Java, https://introcs.cs.princeton.edu/java/home/