**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG THƯƠNG TP.HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

A blue and white logo with a book and a square cap

Description automatically generated

**BÀI TẬP NHÓM**

**CƠ SỞ DỮ LIỆU NOSQL**

**ĐỀ TÀI: ỨNG DỤNG QUẢN LÝ BẢO HÀNH SẢN PHẨM**

Giảng viên hướng dẫn : Nguyễn Thị Thu Tâm

Sinh viên thực hiện :

1. Nguyễn Thành Luân - 2001215939
2. Nguyễn Anh Kiệt - 2001215897
3. Đoàn Văn Hòa – 2001215783
4. Nguyễn Quốc Thịnh - 2001210621

TP. HỒ CHÍ MINH – 2024

BẢNG PHÂN CÔNG CÔNG VIỆC

|  |  |  |  |
| --- | --- | --- | --- |
| Tên | Công Việc | Phần trăm hoàn thành | Điểm |
| Nguyễn Thành Luân | Thiết kế cơ sở dữ liệu  Thiết kế form và chức năng Đăng nhập, Trang chủ, quản lý sản phẩm Giao nhiệm vụ, tổng hợp các theo giai đoạn  Word, ppt | 100% | 10 |
| Nguyễn Anh Kiệt | Đóng góp ý kiến chính sửa cơ sở dữ liệu  Thiết kế form và chức năng quản lý chính sách bảo hành, danh sách bảo hành  Word, ppt | 100% | 10 |
| Đoàn Văn Hòa | Đóng góp ý kiến chính sửa cơ sở dữ liệu  Thiết kế form và chức năng quản lý khách hàng, sản phẩm và khách hàng Giao nhiệm vụ, tổng hợp các theo giai đoạn  Word, ppt | 100% | 10 |
| Nguyễn Quốc Thịnh | Đóng góp ý kiến chính sửa cơ sở dữ liệu  Thiết kế form và chức năng lịch sử sửa chửa, gia hạn bảo hành  Word, ppt | 100% | 10 |

**Mục Lục Hình Ảnh**

[Hình 1:Kiến trúc tổng thể 5](#_Toc180276723)

[Hình 2: Đăng nhập 2](#_Toc180276724)

[Hình 3: Giao diện đăng nhập 3](#_Toc180276725)

[Hình 4: Giao diện tranh chủ 4](#_Toc180276726)

[Hình 5: Giao diện sản phẩm 5](#_Toc180276727)

[Hình 6: Giao diện danh sách bảo hành 6](#_Toc180276728)

[Hình 7: Giao diện chính sách bảo hành 7](#_Toc180276729)

[Hình 8: Giao diện khách hàng 8](#_Toc180276730)

[Hình 9: . Giao diện sản phẩm và khách hàng 9](#_Toc180276731)

[Hình 10: Giao diện gia hạn bảo hành 10](#_Toc180276732)

[Hình 11: Giao diện lịch sửa chữa 11](#_Toc180276733)

[Hình 12:Chức năng đăng nhập 22](#_Toc180276734)

[Hình 13: Giao diện form trang chủ 23](#_Toc180276735)

[Hình 14: Giao diện chức năng form sản phẩm 25](#_Toc180276736)

[Hình 15: .Giao diện chức năng form danh sách bảo hành 27](#_Toc180276737)

[Hình 16:Giao diện chức năng form chính sách bảo hành 30](#_Toc180276738)

[Hình 17:Giao diện chức năng form khách hàng 32](#_Toc180276739)

[Hình 18: Giao diện chức năng form sản phẩm và khách hàng 34](#_Toc180276740)

[Hình 19:Giao diện chức năng form gia hạn bảo hành 35](#_Toc180276741)

[Hình 20:Giao diện chức năng form lịch sửa chữa 37](#_Toc180276742)

**Mục lục**

[Chương 1. Giới thiệu 1](#_Toc180278136)

[1.1. Bối cảnh 1](#_Toc180278137)

[1.2. Vấn đề và giải pháp 1](#_Toc180278138)

[1.3. Công nghệ sử dụng 1](#_Toc180278139)

[Chương 2. Cơ sở lý thuyết 2](#_Toc180278140)

[2.1. Giới thiệu Java Swing 2](#_Toc180278141)

[2.1.1 Ưu điểm 2](#_Toc180278142)

[2.1.2 Nhược điểm 2](#_Toc180278143)

[2.2. Giới thiệu Cassandra 3](#_Toc180278144)

[2.2.1 Ưu điểm 3](#_Toc180278145)

[2.2.2 Nhược điểm 3](#_Toc180278146)

[Chương 3. Thiết kế hệ thống 5](#_Toc180278147)

[3.1. Kiến trúc tổng thể: 5](#_Toc180278148)

[3.2. Thiết kế giao diện (UI): 6](#_Toc180278149)

[3.2.1 Giao diện login 6](#_Toc180278150)

[3.2.2 Giao diện menu 7](#_Toc180278151)

[3.2.3 Giạo diện sản phẩm 8](#_Toc180278152)

[3.2.4 Giao diện Chính sách bảo hành 11](#_Toc180278153)

[3.2.5 Giao diện khách hàng 12](#_Toc180278154)

[3.2.6 Giao diện sản phẩm và khách hàng 13](#_Toc180278155)

[3.2.7 Giao diện gia hạn bảo hành 13](#_Toc180278156)

[3.2.8 Giao diện lịch sửa chữa 14](#_Toc180278157)

[3.3. Thiết kế cơ sở dữ liệu Cassandra: 16](#_Toc180278158)

[3.3.1 Các câu hỏi thiết kế database 16](#_Toc180278159)

[3.3.2 Cơ sở dử liệu 16](#_Toc180278160)

[3.3.3 Dữ liệu 19](#_Toc180278161)

[3.3.4 Các câu truy vấn 24](#_Toc180278162)

[3.3.5 Sao lưu phục hồi dữ liệu 25](#_Toc180278163)

[Chương 4. Chức năng của hệ thống 26](#_Toc180278164)

[4.1. Form trang chủ 26](#_Toc180278165)

[4.2. Form trang chủ 27](#_Toc180278166)

[4.2.1 Tổng quan số liệu chính 28](#_Toc180278167)

[4.2.2 Biểu đồ chính sách sản phẩm 28](#_Toc180278168)

[4.2.3 Biểu đồ trạng thái sửa chữa 28](#_Toc180278169)

[4.3. Form sản phẩm 29](#_Toc180278170)

[4.3.1 Danh sách loại sản phẩm 29](#_Toc180278171)

[4.3.2 Chức năng tìm kiếm theo số serial 29](#_Toc180278172)

[4.3.3 Xuất dữ liệu dưới dạng CSV 30](#_Toc180278173)

[4.3.4 Chức sắp xếp dữ liệu 30](#_Toc180278174)

[4.4. Form danh sách bảo hành 31](#_Toc180278175)

[4.4.1 Thêm yêu cầu bảo hành 31](#_Toc180278176)

[4.4.2 Cập nhật yêu cầu bảo hành 32](#_Toc180278177)

[4.4.3 Xóa yêu cầu bảo hành 32](#_Toc180278178)

[4.4.4 Hủy lọc 32](#_Toc180278179)

[4.4.5 Xuất file PDF 32](#_Toc180278180)

[4.4.6 Lọc bằng cách chọn khách hàng và lọc theo trạng thái 33](#_Toc180278181)

[4.4.7 Quản lý danh sách yêu cầu bảo hành 33](#_Toc180278182)

[4.5. Form chính sách bảo hành 34](#_Toc180278183)

[4.5.1 Thêm chính sách bảo hành 34](#_Toc180278184)

[4.5.2 Cập nhật chính sách bảo hành 35](#_Toc180278185)

[4.5.3 Xóa chính sách bảo hành 35](#_Toc180278186)

[4.5.4 Xuất file PDF 35](#_Toc180278187)

[4.5.5 Xem danh sách chính sách bảo hành 35](#_Toc180278188)

[4.5.6 Lọc chính sách theo loại sản phẩm 36](#_Toc180278189)

[4.6. Form khách hàng 36](#_Toc180278190)

[4.6.1 Chức năng Thêm mới khách hàng: 36](#_Toc180278191)

[4.6.2 Chức năng Cập nhật thông tin khách hàng: 37](#_Toc180278192)

[4.6.3 Chức năng Xóa khách hàng: 37](#_Toc180278193)

[4.6.4 Chức năng Xuất dữ liệu khách hàng sang định dạng JSON: 37](#_Toc180278194)

[4.7. Form Sản phẩm và khách hàng 38](#_Toc180278195)

[4.7.1 Hiển thị thông tin: 38](#_Toc180278196)

[4.7.2 Cập nhật thông tin 38](#_Toc180278197)

[4.7.3 Tìm kiếm 38](#_Toc180278198)

[4.8. Form gia hạn bảo hành 39](#_Toc180278199)

[4.8.1 Thêm gia hạn bảo hành 39](#_Toc180278200)

[4.8.2 Load gia hạn bảo hành 39](#_Toc180278201)

[4.8.3 Update gia hạn bảo hành 40](#_Toc180278202)

[4.8.4 Xóa gia hạn bảo hành 40](#_Toc180278203)

[4.8.5 Xuất file gia hạn bảo hành 40](#_Toc180278204)

[4.8.6 Tìm kiếm gia hạn bảo hành 40](#_Toc180278205)

[4.9. Form lịch sử sửa chữa 41](#_Toc180278206)

[4.9.1 Thêm lịch sửa chữa 41](#_Toc180278207)

[4.9.2 Sửa lịch sửa chữa 41](#_Toc180278208)

[4.9.3 Xóa lịch sửa chữa 42](#_Toc180278209)

[4.9.4 Load lịch sửa chữa 42](#_Toc180278210)

[4.9.5 Xuất lịch sửa chữa 42](#_Toc180278211)

[4.9.6 Tìm kiếm lịch sửa chữa 42](#_Toc180278212)

[4.9.7 Bảng tìm kiếm thông tin sửa chữa của 1 sản phẩm(Bảng nhỏ) 42](#_Toc180278213)

[Chương 5. Kết quả và đánh giá 43](#_Toc180278214)

[5.1. Kết quả đạt được 43](#_Toc180278215)

[5.2. Đánh giá 43](#_Toc180278216)

[5.2.1 Ưu điểm 43](#_Toc180278217)

[5.2.2 Nhược điểm 44](#_Toc180278218)

[Chương 6. Kết luận 45](#_Toc180278219)

[Tài liệu tham khảo 46](#_Toc180278220)

**LỜI CẢM ƠN**

Lời đầu tiên, nhóm chúng em xin chân thành cảm ơn cô Nguyễn Thị Thu Tâm đã tận tình giảng dạy và hướng dẫn chúng em trong suốt thời gian học môn “NoSql”. Sự nhiệt huyết, tận tâm và chu đáo của cô trong việc truyền đạt kiến thức đã tạo ra một môi trường học tập đầy cảm hứng và sáng tạo cho chúng em. Những bài giảng phong phú và sinh động của cô không chỉ cung cấp cho chúng em những kiến thức lý thuyết cần thiết mà còn giúp chúng em áp dụng những kiến thức đó vào thực tiễn thông qua các bài tập và đồ án thực tế.

Trong quá trình học, nhóm chúng em đã được tiếp xúc với nhiều khái niệm mới, từ cách tổ chức dữ liệu, xử lý dữ liệu đến các phương pháp tối ưu hóa truy vấn. Những kiến thức này không chỉ hữu ích cho bài tiểu luận này mà còn là nền tảng quan trọng cho sự nghiệp sau này của chúng em trong lĩnh vực công nghệ thông tin.

Nhóm em xin chúc cô luôn mạnh khỏe, hạnh phúc và thành công trên con đường giảng dạy của mình. Hy vọng rằng trong tương lai, chúng em sẽ có cơ hội được học hỏi thêm từ cô cũng như tiếp tục phát triển những kiến thức mà cô đã dạy cho chúng em.

Cuối cùng, chúng em xin chân thành cảm ơn cô một lần nữa vì tất cả những gì cô đã dành cho chúng em trong suốt thời gian qua!

# Giới thiệu

## Bối cảnh

Trong bối cảnh cạnh tranh ngày càng gay gắt của thị trường tiêu dùng hiện đại, việc cung cấp dịch vụ sau bán hàng chất lượng cao trở thành yếu tố sống còn quyết định sự thành công của doanh nghiệp. Quản lý bảo hành sản phẩm không chỉ giúp doanh nghiệp tăng cường sự hài lòng của khách hàng mà còn bảo vệ thương hiệu và nâng cao giá trị sản phẩm. Một hệ thống quản lý bảo hành hiệu quả sẽ cho phép doanh nghiệp theo dõi, ghi nhận và xử lý các yêu cầu bảo hành một cách nhanh chóng và chính xác, từ đó cải thiện trải nghiệm của khách hàng và tối ưu hóa quy trình làm việc nội bộ.

## Vấn đề và giải pháp

Mặc dù nhu cầu quản lý bảo hành sản phẩm ngày càng cao, nhiều doanh nghiệp vẫn gặp phải những khó khăn trong việc theo dõi các yêu cầu bảo hành, lưu trữ và quản lý dữ liệu lịch sử bảo hành. Việc thiếu một hệ thống tích hợp dẫn đến sự chậm trễ trong xử lý yêu cầu và gây ra sự không hài lòng cho khách hàng. Hệ thống quản lý bảo hành mà chúng tôi phát triển sẽ giải quyết những vấn đề này thông qua việc cung cấp một giao diện trực quan, cho phép người dùng dễ dàng nhập liệu và truy xuất thông tin. Hệ thống cũng sẽ tự động ghi nhận các hoạt động bảo hành, giúp doanh nghiệp theo dõi được lịch sử và tình trạng bảo hành của từng sản phẩm.

## Công nghệ sử dụng

Để phát triển ứng dụng quản lý bảo hành, chúng tôi đã lựa chọn sử dụng Java Swing để xây dựng giao diện người dùng. Java Swing nổi bật với khả năng tạo ra các ứng dụng desktop đẹp mắt và dễ sử dụng, đồng thời tích hợp tốt với ngôn ngữ lập trình Java, giúp tăng cường tính linh hoạt và khả năng bảo trì của hệ thống. Về phần lưu trữ dữ liệu, chúng tôi chọn Cassandra, một hệ quản trị cơ sở dữ liệu NoSQL, vì khả năng xử lý lượng lớn dữ liệu và mở rộng mạnh mẽ. Cassandra cho phép chúng tôi lưu trữ và truy xuất thông tin bảo hành một cách nhanh chóng và hiệu quả, đồng thời hỗ trợ tính sẵn sàng cao, giảm thiểu thời gian ngừng hoạt động của hệ thống.

# Cơ sở lý thuyết

## Giới thiệu Java Swing

- Là một phần của **Java Foundation Classes (JFC)**, cung cấp một bộ công cụ mạnh mẽ để phát triển giao diện đồ họa người dùng (GUI) cho các ứng dụng Java. Swing được phát triển như một giải pháp thay thế cho AWT (Abstract Window Toolkit) với nhiều tính năng và khả năng mở rộng hơn, cho phép các nhà phát triển tạo ra các ứng dụng desktop phong phú và tương tác.

- Nó hoạt động dựa trên mô hình **event-driven** (dựa trên sự kiện), nơi các sự kiện như nhấn nút, di chuyển chuột hay nhập dữ liệu được xử lý thông qua các bộ lắng nghe sự kiện (event listeners).

### Ưu điểm

* Một trong những điểm mạnh lớn nhất của Java Swing là tính đa nền tảng. Ứng dụng Swing có thể chạy trên nhiều hệ điều hành khác nhau mà không cần chỉnh sửa mã nguồn nhờ vào môi trường Java Virtual Machine (JVM).
* Swing cung cấp các thành phần giao diện người dùng phong phú như JButton, JLabel, JTextField, JTable, JTree, JPanel, v.v. Điều này cho phép nhà phát triển xây dựng các giao diện phức tạp và dễ sử dụng.
* Với Swing, bạn có thể dễ dàng tùy chỉnh các thành phần giao diện người dùng. Nó hỗ trợ **Look and Feel** (giao diện và cảm nhận) cho phép thay đổi toàn bộ giao diện của ứng dụng để phù hợp với hệ điều hành hoặc yêu cầu của ứng dụng.
* Swing là một thư viện GUI nhẹ vì nó không dựa hoàn toàn vào thành phần giao diện của hệ điều hành, mà thay vào đó vẽ lại các thành phần GUI riêng của nó. Điều này giúp Swing độc lập với các hạn chế của hệ điều hành.

### Nhược điểm

* Mặc dù Swing có thể xây dựng các giao diện phong phú và mạnh mẽ, nhưng khi làm việc với các ứng dụng có giao diện lớn và phức tạp, nó có thể trở nên cồng kềnh và khó bảo trì. Nhà phát triển cần hiểu rõ cấu trúc của Swing và biết cách tổ chức mã nguồn một cách hợp lý.
* Mặc dù Swing cho phép tùy chỉnh giao diện người dùng, nhưng theo mặc định, nó không cung cấp các thành phần giao diện hiện đại như các framework mới hơn như JavaFX. Việc tạo ra giao diện hiện đại và hấp dẫn có thể đòi hỏi nhiều công sức và kiến thức về thiết kế.
* So với JavaFX, Swing có thể hoạt động chậm hơn khi xử lý các thành phần đồ họa phức tạp hoặc yêu cầu hiệu năng cao, đặc biệt là khi cần xử lý các thao tác đồ họa động.

## Giới thiệu Cassandra

* Là một cơ sở dữ liệu NoSQL phân tán, mã nguồn mở, được thiết kế để xử lý một lượng lớn dữ liệu phi cấu trúc trên nhiều máy chủ mà không gặp phải các vấn đề về điểm tắc nghẽn hoặc sự phụ thuộc vào một điểm duy nhất. Nó được phát triển ban đầu bởi Facebook và sau đó được trao cho Apache Foundation.
* Tính năng chính của Cassandra là khả năng chịu lỗi và mở rộng ngang (horizontal scaling), làm cho nó trở thành lựa chọn tốt cho các ứng dụng yêu cầu dữ liệu lớn và khả năng đáp ứng thời gian thực.
* Cassandra được thiết kế để hoạt động trên nhiều nút trong một cụm (cluster), dữ liệu được phân tán trên tất cả các nút để đảm bảo khả năng chịu lỗi, bảo mật và khả năng truy cập dữ liệu.

### Ưu điểm

* Cassandra có thể mở rộng dễ dàng bằng cách thêm các nút mới vào cụm mà không làm gián đoạn hoạt động. Nó sử dụng kiến trúc phân tán với khả năng cân bằng tải tốt, giúp tăng cường hiệu suất khi quy mô dữ liệu lớn lên.
* Các nút trong Cassandra đều bình đẳng, không có nút nào đóng vai trò trung tâm, giúp loại bỏ nguy cơ lỗi toàn bộ hệ thống khi một nút gặp sự cố.
* Cassandra sử dụng phương pháp partitioning để chia dữ liệu thành các phân vùng, giúp truy vấn và lưu trữ dữ liệu được phân tán đều trên các nút trong hệ thống. Điều này giúp tăng tốc độ xử lý truy vấn khi dữ liệu được phân phối rộng rãi.
* Cassandra được thiết kế cho các hệ thống yêu cầu thời gian đáp ứng nhanh, đặc biệt là các ứng dụng có khối lượng truy cập đọc và ghi lớn.
* Với cơ chế sao chép dữ liệu (replication), Cassandra có thể đảm bảo rằng nếu một hoặc nhiều nút bị lỗi, dữ liệu vẫn được bảo toàn và có thể truy cập từ các bản sao ở các nút khác.

### Nhược điểm

* Cassandra sử dụng một mô hình lưu trữ khác với các cơ sở dữ liệu quan hệ truyền thống, đòi hỏi nhà phát triển phải hiểu rõ về cách tổ chức dữ liệu phù hợp với kiểu dữ liệu phân tán. Điều này dẫn đến sự phức tạp khi thiết kế bảng và cấu trúc dữ liệu.
* Cassandra được tối ưu cho các hoạt động đọc/ghi tốc độ cao, nhưng không phù hợp với các ứng dụng yêu cầu nhiều mối quan hệ phức tạp giữa các bảng dữ liệu (join). Nó không hỗ trợ câu lệnh JOIN như trong các hệ quản trị cơ sở dữ liệu quan hệ (RDBMS), dẫn đến việc cần phải tái cấu trúc dữ liệu để xử lý các yêu cầu phức tạp.

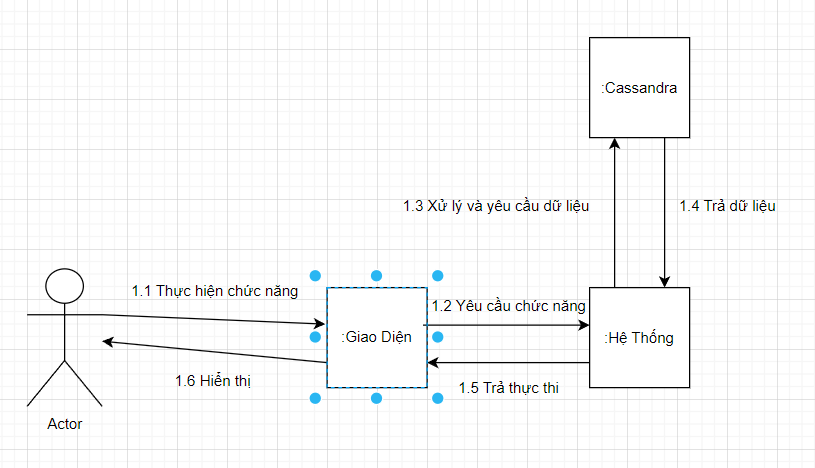
Cassandra không hỗ trợ đầy đủ các tính năng giao dịch như các cơ sở dữ liệu quan hệ, ví dụ như đảm bảo toàn vẹn dữ liệu hoặc khả năng rollback (hoàn tác trong trường hợp lỗi xảy ra trong quá trình thực hiện giao dịch.

# Thiết kế hệ thống

## Kiến trúc tổng thể:

Hệ thống bao gồm các thành phần chính: Giao diện người dùng (UI) bằng Java Swing, Business Logic và Cơ sở dữ liệu Cassandra.

• Sơ đồ kiến trúc: Sơ đồ mô tả sự tương tác của người dùng đến với hệ thống



Hình 1:Kiến trúc tổng thể

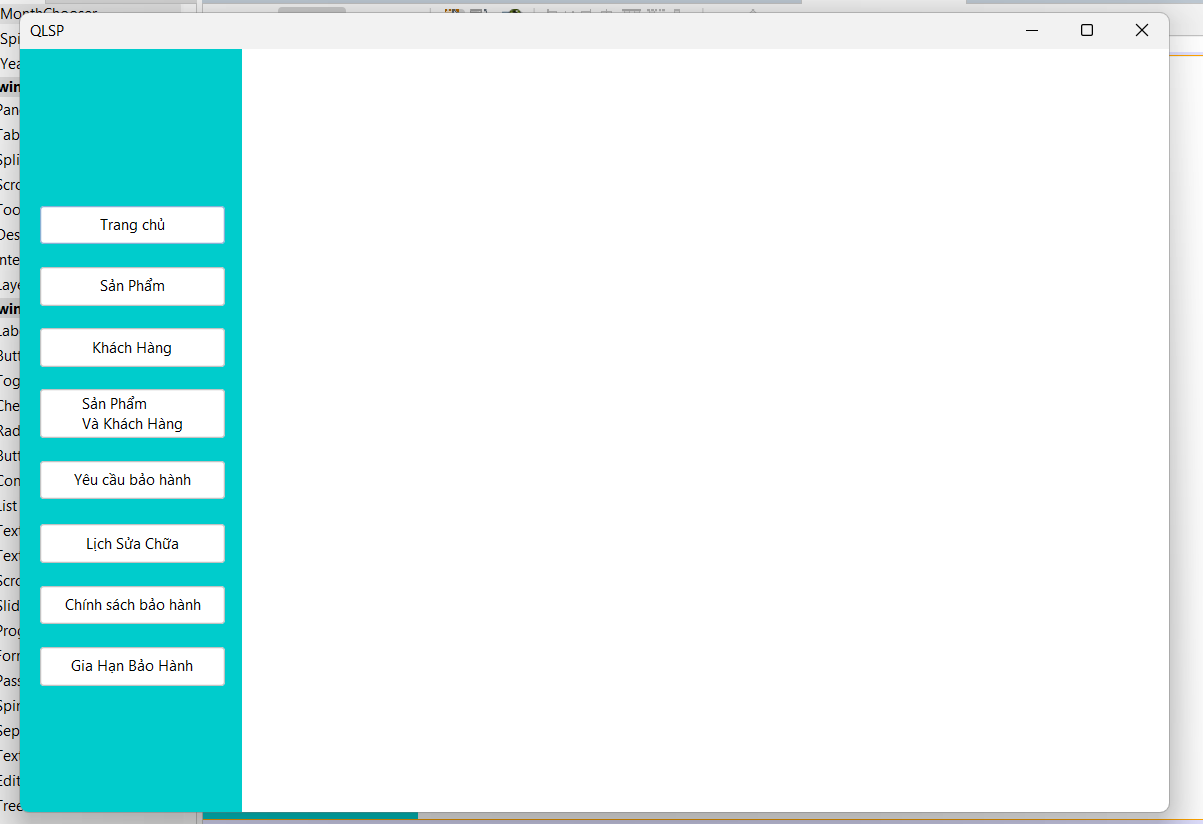
## Thiết kế giao diện (UI):

### A screenshot of a computer Description automatically generatedGiao diện login

Hình 2: Đăng nhập

* Giao diện giúp admin đăng nhập vào hệ thống

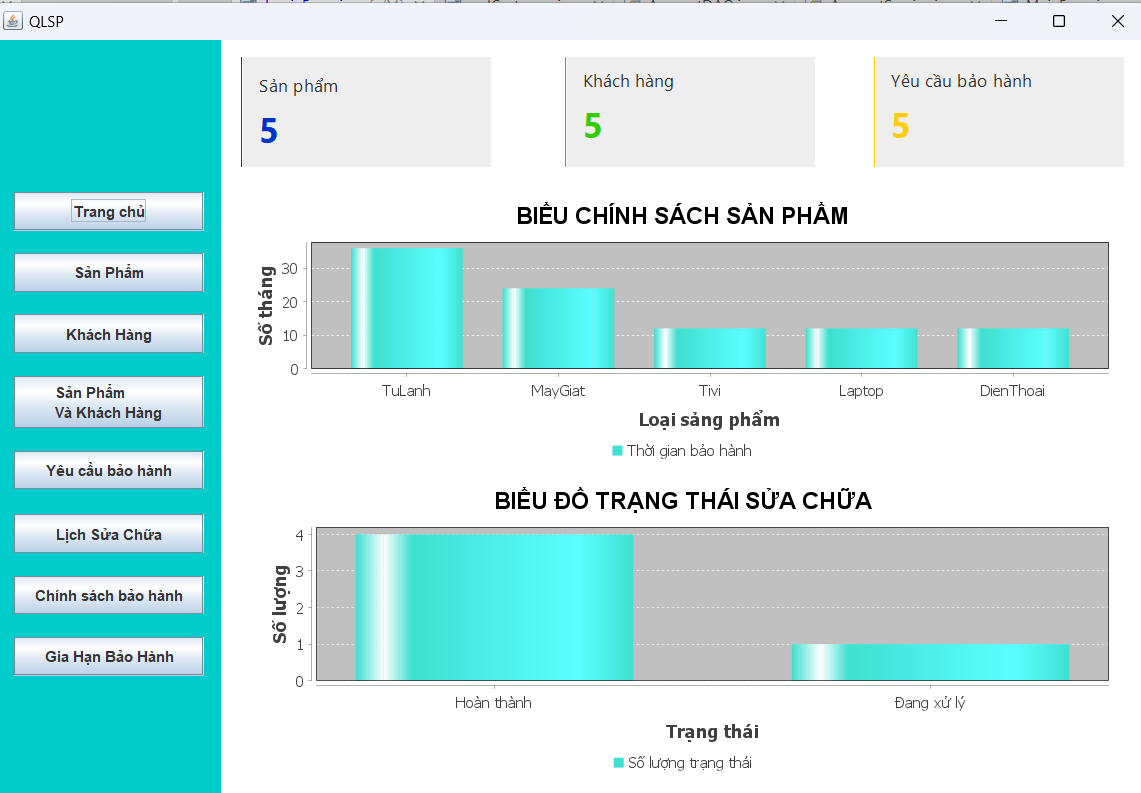
### Giao diện menu

* Giao diện giúp điều hướng đến các phần cần thực hiện qua các button ở menu, khoảng trắng sẽ hiện thị các giao diện cần thao tác****

Hình 3: Giao diện đăng nhập

Giao diện trang chủ

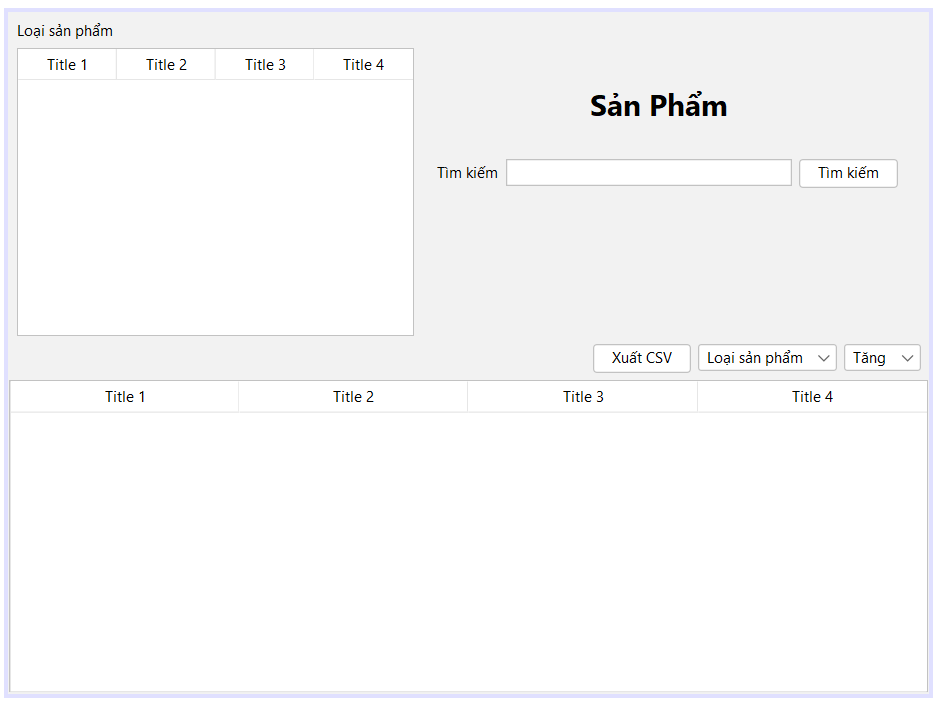
* Thống kê và biểu đồ liên quan tới các hệ thống

****

Hình 4: Giao diện tranh chủ

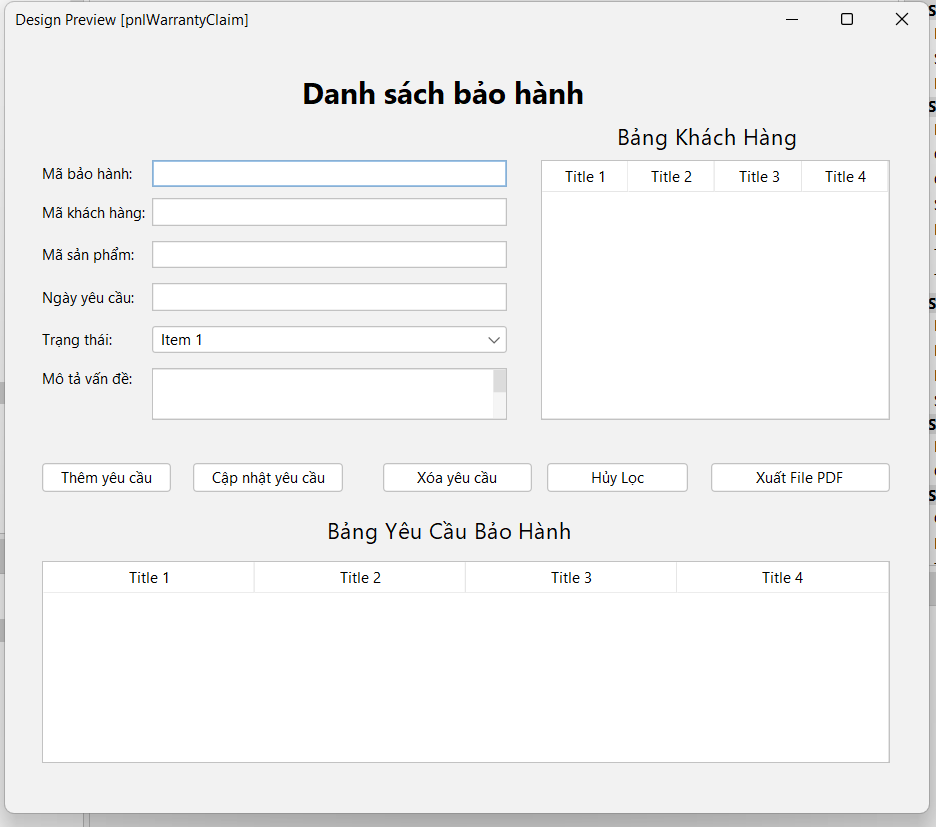
### Giạo diện sản phẩm

* Xem chi tiết sản phẩm và loại sản phẩm, tìm kiếm sản phẩm và xuất dữ liệu

****

Hình 5: Giao diện sản phẩm

Giao diện danh sách bảo hành

****

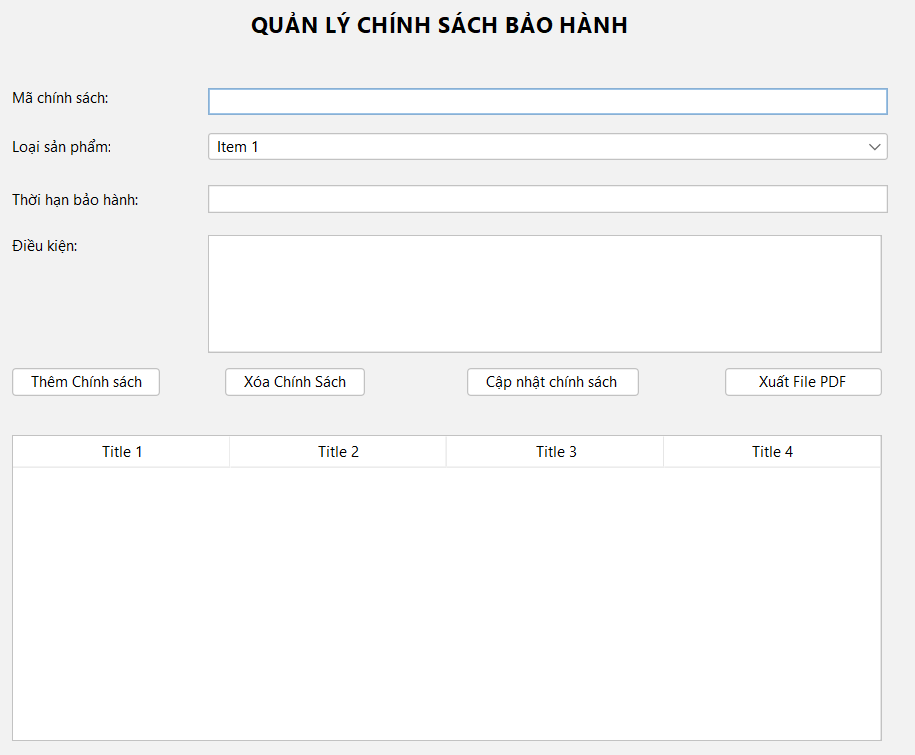
Hình 6: Giao diện danh sách bảo hành

- Form thêm yêu cầu bảo hành: Người dùng có thể nhập thông tin liên quan đến sản phẩm, lỗi sản phẩm, và thời gian yêu cầu bảo hành để tạo yêu cầu mới.

- Bảng yêu cầu bảo hành: Hiển thị danh sách các yêu cầu bảo hành đã được tạo, người dùng có thể xem thông tin chi tiết hoặc chọn để chỉnh sửa, cập nhật hoặc xuất dữ liệu thành file PDF.

- Form cập nhật trạng thái bảo hành: Sau khi sản phẩm đã được sửa chữa, người dùng có thể cập nhật trạng thái bảo hành của yêu cầu để phản ánh tiến độ xử lý.

### Giao diện Chính sách bảo hành



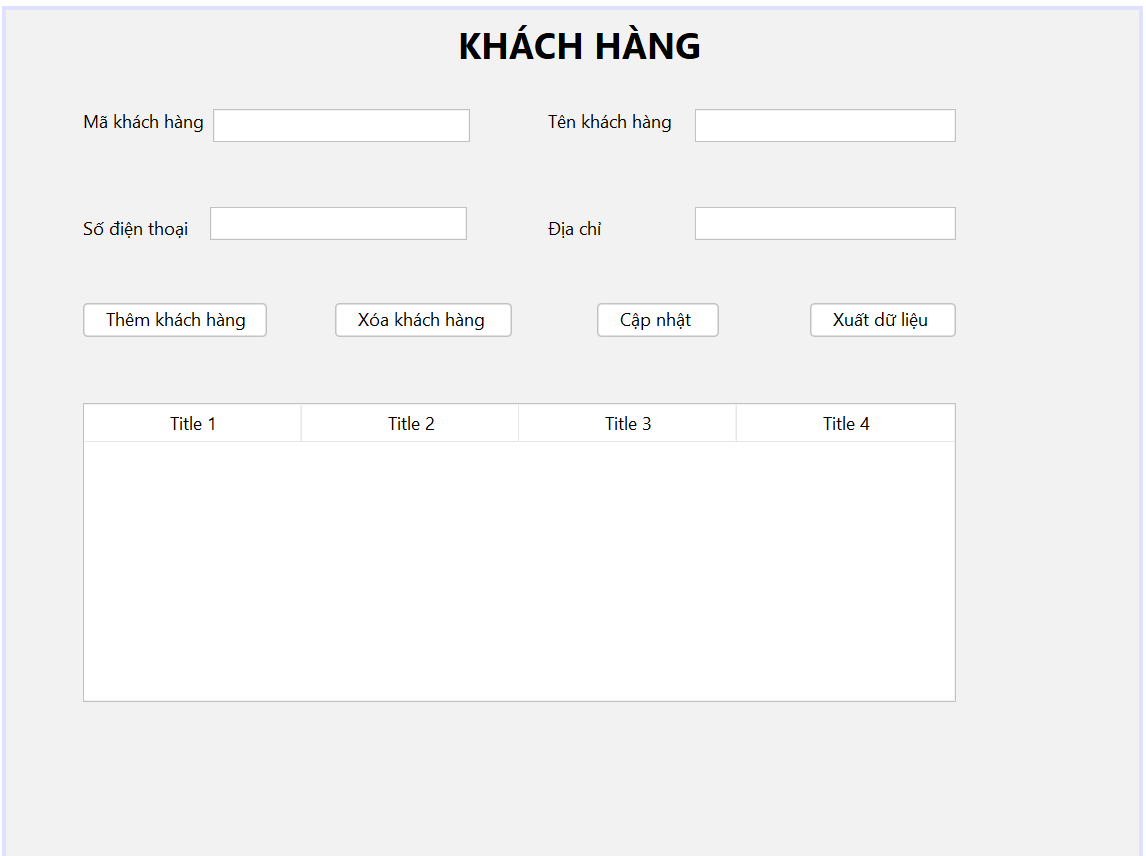
Hình 7: Giao diện chính sách bảo hành

- Form thêm chính sách bảo hành: Người dùng nhập mã chính sách, loại sản phẩm, thời hạn bảo hành, và các điều kiện đi kèm để tạo mới một chính sách bảo hành cho sản phẩm.

- Bảng yêu cầu bảo hành: Hiển thị danh sách các yêu cầu bảo hành đã được tạo, người dùng có thể xem thông tin chi tiết hoặc chọn để chỉnh sửa, cập nhật hoặc xuất dữ liệu thành file PDF.

- Form xuất thông tin chính sách: Người dùng có thể xuất dữ liệu của các chính sách bảo hành thành file PDF để lưu trữ hoặc chia sẻ.

### Giao diện khách hàng



Hình 8: Giao diện khách hàng

- Form Khách hàng: Người dùng có thể tùy chỉnh nhập dữ liệu vào các trường như Mã khách hàng, Tên khách hàng, Số điện thoại, Địa chỉ.

- Bảng Khách hàng sẽ hiển thị thông tin của toàn bộ khách hàng có trong hệ thống, khi click vào hàng bất kì thì thông tin của khách hàng đó sẽ được đổ vào các trường tương ứng.

- Các nút chức năng: Người dùng có thể tùy chỉnh thêm xóa sửa thông tin của 1 khách hàng và có thể xuất những dữ liệu của người dùng về dưới dạng JSON

### Giao diện sản phẩm và khách hàng

A screenshot of a computer

Description automatically generated

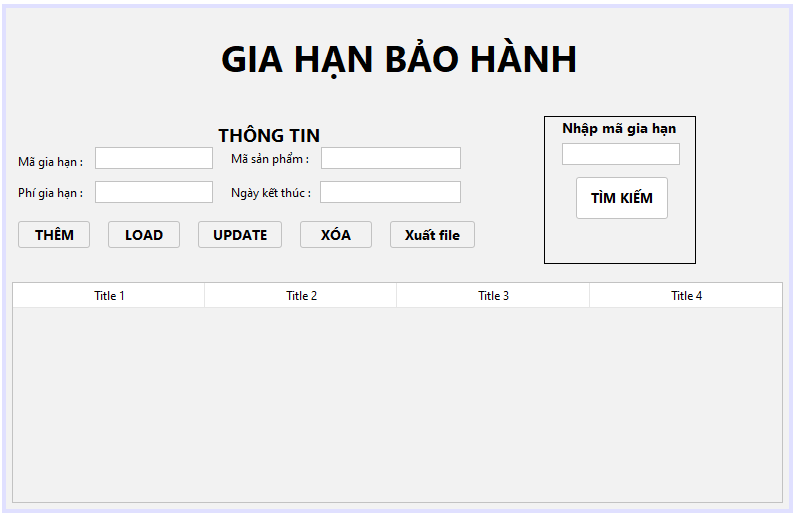
Hình 9: . Giao diện sản phẩm và khách hàng

- Form Sản phẩm và khách hàng: Sẽ lấy thông tin khách hàng và sản phẩm thông qua 1 bảng Customer\_Products.

- Bảng thông tin khách hàng và sản phẩm: Chứa thông tin của bảng Customer\_Product, dữ liệu sẽ được hiển thị lên và khi người dùng click vào dòng nào thì thông tin của khách hàng và sản phẩm tương ứng sẽ được xuất hiện vào các ô text và 2 bảng bên dưới.  
- Bảng sản phẩm: Hiển thị thông tin sản phẩm ứng với mã khách hàng.

- Bảng khách hàng: Hiển thị thông tin khách hàng khi chúng click vào dòng bất kì của bảng Customer\_Product.

### Giao diện gia hạn bảo hành



Hình 10: Giao diện gia hạn bảo hành

- Form giao diện gia hạn bảo hành cho phép người dùng có thể nhập liệu mã gia hạn, mã sản phẩm, phí gia hạn và ngày kết thúc để lưu thông tin.

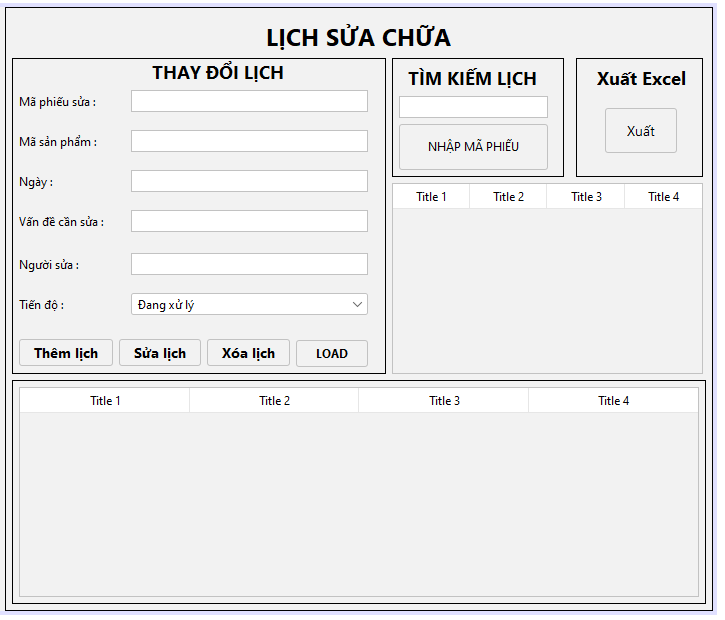
- Có các chức năng thêm, xóa, sửa,tương ứng, phù hợp với nhu cầu của người dùng sử dụng khi cần thiết.

- Chức năng Load, dùng để load lại dữ liệu ban đầu khi thao tác xong và cũng có thể sắp xếp lại theo thứ tự từ nhỏ đến lớn của mã để người dùng có thể quan sát dễ hơn

- Khi người dùng muốn thống kê, hoặc sao kê, có thể sử dụng chức năng Xuất ra file excel(.xlsx) để có thể dễ dàng quản lí hơn.

- Ngoài ra còn có thể tìm kiếm thông tin bằng cách nhập mã gia hạn, khi đó thông tin sẽ đổ vào bảng giúp tối ưu hóa việc tìm kiếm dữ liệu.

### Giao diện lịch sửa chữa



Hình 11: Giao diện lịch sửa chữa

- Form giao diện lịch sữa chữa cho phép người dùng có thể nhập mã phiếu sữa, mã sản phẩm, ngày sửa, vấn đề cần sửa, người phụ trách và tiến độ.

- Có các chứng năng, thêm xáo, sửa, tương ứng phù hợp với nhu cầu của người dùng sử dụng khi cần thiết.

- Chức năng load lại dữ liệu ban đầu khi thao tác xong và cũng có thể sắp xếp lại thoe thứ tự từ nhỏ đến lớn của mã phiếu sửa để người dùng dễ quan sát hơn.

- Người dùng cũng có thể xuất ra file excel(.xlsx) để thống kê để dễ dàng quản lý hơn.

- Có chức năng tìm kiếm thông tin bằng cách nhập mã phiếu sửa., khi đó thông tin sẽ đổ vào bảng giúp tối ưu hóa việc tìm kiếm dữ liệu.

- Theo như form, bảng nhỏ ở trên là bảng danh sách sản phẩm, khi mình muốn kiểm tra sản phẩm đang trong tiến độ nào, mình click vào sản phẩm trong bảng danh sách sản phẩm, lập tức bảng lớn phía dưới sẽ hiện lên thông tin ngày sửa, vấn đề cần sửa, tiến độ và người phụ trách sửa

## Thiết kế cơ sở dữ liệu Cassandra:

### Các câu hỏi thiết kế database

* Loại sản phẩm: Có các loại sản phẩm nào cần được quản lý bảo hành? Một loại sản phẩm có thể chứa nhiều sản phẩm không?
* Thông tin sản phẩm: Bạn muốn lưu những thông tin gì về sản phẩm (ví dụ: số serial, mã sản phẩm, ngày mua, nhà sản xuất)?
* Trạng thái bảo hành: Bạn có cần quản lý trạng thái bảo hành của sản phẩm không (ví dụ: đang hoạt động, hết hạn, đã sửa chữa)?
* Thời hạn bảo hành: Thời hạn bảo hành được lưu như thế nào? Bạn có cần theo dõi các mốc thời gian quan trọng như ngày bắt đầu và ngày kết thúc bảo hành?
* Thông tin khách hàng: Bạn có cần lưu thông tin về khách hàng mua sản phẩm không? Nếu có, cần lưu những thông tin gì (ví dụ: tên, địa chỉ, số điện thoại)?
* Yêu cầu bảo hành: Bạn có cần lưu các yêu cầu bảo hành của khách hàng không? Thông tin này bao gồm gì (ví dụ: ngày yêu cầu, vấn đề gặp phải, trạng thái xử lý)?
* Lịch sử sửa chữa: Bạn có cần lưu lịch sử sửa chữa hoặc thay thế linh kiện không? Nếu có, cần theo dõi những thông tin nào (ví dụ: ngày sửa chữa, bộ phận thay thế, kỹ thuật viên)?
* Bảo hành mở rộng: Bạn có cung cấp dịch vụ bảo hành mở rộng không? Nếu có, thông tin gì cần được lưu liên quan đến bảo hành mở rộng (ví dụ: phí mở rộng, thời hạn mở rộng)?
* Chính sách bảo hành: Bạn có cần lưu thông tin về các chính sách bảo hành theo từng loại sản phẩm không? Nếu có, chính sách bảo hành cần lưu trữ những gì (ví dụ: điều kiện áp dụng, giới hạn thời gian)?

### Cơ sở dử liệu

CREATE KEYSPACE bhsp WITH replication = {'class': 'SimpleStrategy', 'replication\_factor': 1};

* **CREATE KEYSPACE bhsp**: Câu lệnh này tạo ra một "keyspace" có tên là bhsp. Trong cơ sở dữ liệu Cassandra, keyspace là một container ở mức cao nhất cho dữ liệu. Nó tương tự như một cơ sở dữ liệu trong các hệ quản trị cơ sở dữ liệu khác. Mọi bảng, chỉ mục, và các thành phần liên quan đến cơ sở dữ liệu đều được đặt trong một keyspace.
* **WITH replication = {'class': 'SimpleStrategy', 'replication\_factor': 1}**: Đây là phần cấu hình phương thức sao chép dữ liệu trong Cassandra.
  + class: 'SimpleStrategy': Đây là chiến lược sao chép đơn giản nhất trong Cassandra. Nó phù hợp để sử dụng trong môi trường đơn cluster hoặc các môi trường phát triển. Chiến lược này sao chép dữ liệu theo một cách trực tiếp, nghĩa là dữ liệu sẽ được sao chép đến một số lượng node (nút) nhất định dựa trên giá trị của **replication\_factor.**
  + **replication\_factor: 1:** Điều này chỉ ra rằng dữ liệu chỉ được lưu trữ trên 1 node duy nhất. Với replication\_factor bằng 1, không có sự sao chép dữ liệu giữa các node khác, nên nếu node này gặp sự cố, dữ liệu có thể bị mất. Đây thường là cấu hình dùng trong môi trường phát triển hoặc thử nghiệm.
* Lưu ý: Trong môi trường sản xuất thực tế, người ta thường sử dụng các chiến lược phức tạp hơn (như **NetworkTopologyStrategy**) và thường có **replication\_factor** lớn hơn 1 để đảm bảo tính dự phòng và sẵn sàng cao.

a. Bảng type\_product

* **Chức năng**: Lưu trữ thông tin về loại sản phẩm.
* **Cột**:
  + **product\_type** (**TEXT, PRIMARY KEY**): Loại sản phẩm, ví dụ như "LapTop", "DienThoai". Đây là khóa chính của bảng.
  + **description** (TEXT): Mô tả chi tiết về loại sản phẩm.
* **Ý nghĩa**: Bảng này giúp phân loại các sản phẩm, cho phép mô tả và quản lý các loại sản phẩm khác nhau.

b. Bảng products

* **Chức năng**: Lưu trữ thông tin về các sản phẩm được bán ra và bảo hành.
* **Cột**:
  + **product\_id** (TEXT): Mã sản phẩm duy nhất.
  + **serial\_number** (TEXT): Số seri của sản phẩm.
  + **product\_type** (TEXT): Loại sản phẩm, liên kết đến bảng **type\_product**.
  + **product\_name** (TEXT): Tên sản phẩm.
  + **manufacturer** (TEXT): Nhà sản xuất.
  + **purchase\_date** (DATE): Ngày mua sản phẩm.
  + **warranty\_start\_date** (DATE): Ngày bắt đầu bảo hành.
  + **warranty\_end\_date** (DATE): Ngày kết thúc bảo hành.
* **Khóa chính**: (**product\_id**, **product\_type**). **product\_id** được sử dụng để phân biệt các sản phẩm, còn **product\_type** đảm bảo phân biệt sản phẩm dựa trên loại.
* **Ý nghĩa**: Bảng này quản lý thông tin cơ bản về sản phẩm, bao gồm thông tin bảo hành và nhà sản xuất.

c. Bảng customers

* **Chức năng**: Lưu trữ thông tin về khách hàng.
* **Cột**:
  + **customer\_id** (**TEXT**, **PRIMARY KEY**): Mã khách hàng duy nhất.
  + **name** (TEXT): Tên khách hàng.
  + **address** (TEXT): Địa chỉ của khách hàng.
  + **phone\_number** (TEXT): Số điện thoại của khách hàng.
* **Ý nghĩa**: Bảng này quản lý thông tin của khách hàng đã mua sản phẩm.

d. Bảng customer\_products

* **Chức năng**: Lưu trữ thông tin về sản phẩm mà mỗi khách hàng đã mua.
* **Cột**:
  + **customer\_id** (TEXT): Mã khách hàng, liên kết với bảng **customers**.
  + **product\_id** (TEXT): Mã sản phẩm, liên kết với bảng products.
* **Khóa chính**: (**customer\_id**, **product\_id**). Điều này giúp phân biệt các sản phẩm mà mỗi khách hàng sở hữu.
* **Ý nghĩa**: Bảng này tạo liên kết giữa khách hàng và sản phẩm họ đã mua.

e. Bảng warranty\_claims

* **Chức năng**: Lưu trữ các yêu cầu bảo hành từ khách hàng.
* **Cột**:
  + **claim\_id** (TEXT, **PRIMARY KEY**): Mã yêu cầu bảo hành duy nhất.
  + **customer\_id** (TEXT): Mã khách hàng, liên kết với bảng **customers**.
  + **product\_id** (TEXT): Mã sản phẩm, liên kết với bảng **products**.
  + **claim\_date** (DATE): Ngày yêu cầu bảo hành.
  + **issue\_description** (TEXT): Mô tả chi tiết về vấn đề sản phẩm gặp phải.
  + **claim\_status** (TEXT): Trạng thái yêu cầu bảo hành (ví dụ: "Chưa giải quyết", "Đã giải quyết").
* **Ý nghĩa**: Bảng này giúp theo dõi các yêu cầu bảo hành từ khách hàng và tình trạng giải quyết của chúng.

f. Bảng repair\_history

* **Chức năng**: Lưu trữ lịch sử sửa chữa của sản phẩm.
* **Cột**:
  + **repair\_id** (TEXT, **PRIMARY** KEY): Mã sửa chữa duy nhất.
  + **product\_id** (TEXT): Mã sản phẩm, liên kết với bảng **products**.
  + **repair\_date** (DATE): Ngày sửa chữa.
  + **repair\_description** (TEXT): Mô tả chi tiết về sửa chữa.
  + **status** (TEXT): Trạng thái sửa chữa (ví dụ: "Đang sửa", "Đã sửa xong").
  + **technician** (TEXT): Tên kỹ thuật viên thực hiện sửa chữa.
* **Ý nghĩa**: Bảng này ghi lại các thông tin chi tiết về các lần sửa chữa của sản phẩm, giúp theo dõi lịch sử bảo hành và sửa chữa.

g. Bảng warranty\_policies

* **Chức năng**: Lưu trữ thông tin về chính sách bảo hành cho từng loại sản phẩm.
* **Cột**:
  + **policy\_id** (TEXT, **PRIMARY KEY**): Mã chính sách bảo hành duy nhất.
  + **product\_type** (TEXT): Loại sản phẩm, liên kết với bảng **type\_product**.
  + **policy\_description** (TEXT): Mô tả chi tiết về chính sách bảo hành.
  + **warranty\_duration** (INT): Thời gian bảo hành (tính theo tháng).
  + **conditions** (TEXT): Các điều kiện áp dụng chính sách bảo hành.
* **Ý nghĩa**: Bảng này quy định chính sách bảo hành cho các loại sản phẩm khác nhau, bao gồm cả thời gian và điều kiện bảo hành.

h. Bảng extended\_warranty

* **Chức năng**: Lưu trữ thông tin về các lần gia hạn bảo hành cho sản phẩm.
* **Cột**:
  + **extended\_warranty\_id** (TEXT, **PRIMARY KEY**): Mã gia hạn bảo hành duy nhất.
  + **product\_id** (TEXT): Mã sản phẩm, liên kết với bảng **products**.
  + **extension\_fee** (DOUBLE): Phí gia hạn bảo hành.
  + **new\_warranty\_end\_date** (DATE): Ngày kết thúc bảo hành mới sau khi gia hạn.
* **Ý nghĩa**: Bảng này giúp theo dõi các gói bảo hành mở rộng mà khách hàng đã mua cho sản phẩm của họ.

i. Bảng account

* **Chức năng**: Lưu thông tin tài khoản của admin
* **Cột**:
  + **username**(TEXT, **PRIMARY KEY**): Tên đăng nhập là duy nhấtt.
  + **password**(TEXT):Mật khẩu.
* **Ý nghĩa**: Bảng này giúp theo admin đăng nhập vào hệ thống

### Dữ liệu

a. Thêm dữ liệu cho type\_product

INSERT INTO type\_product (product\_type, description)

VALUES ('Laptop', 'Máy tính xách tay');

INSERT INTO type\_product (product\_type, description)

VALUES ('DienThoai', 'Thiết bị di động');

INSERT INTO type\_product (product\_type, description)

VALUES ('MayGiat', 'Máy giặt quần áo');

INSERT INTO type\_product (product\_type, description)

VALUES ('Tivi', 'Máy truyền hình');

INSERT INTO type\_product (product\_type, description)

VALUES ('TuLanh', 'Tủ lạnh');

b. Thêm dữ liệu vào bảng products

INSERT INTO products (product\_id, serial\_number, product\_type, product\_name, manufacturer, purchase\_date, warranty\_start\_date, warranty\_end\_date)

VALUES

('P001', 'SN123456', 'Laptop', 'Laptop Dell XPS 13', 'Dell', '2023-01-01', '2023-01-15', '2024-01-15');

INSERT INTO products (product\_id, serial\_number, product\_type, product\_name, manufacturer, purchase\_date, warranty\_start\_date, warranty\_end\_date)

VALUES ('P002', 'SN789012', 'DienThoai', 'iPhone 14 Pro', 'Apple', '2023-02-20', '2023-03-01', '2024-03-01');

INSERT INTO products (product\_id, serial\_number, product\_type, product\_name, manufacturer, purchase\_date, warranty\_start\_date, warranty\_end\_date)

VALUES ('P003', 'SN345678', 'MayGiat', 'Máy giặt LG Inverter', 'LG', '2023-03-15', '2023-04-01', '2025-04-01');

INSERT INTO products (product\_id, serial\_number, product\_type, product\_name, manufacturer, purchase\_date, warranty\_start\_date, warranty\_end\_date)

VALUES ('P004', 'SN901234', 'Tivi', 'Tivi Samsung 55 inch', 'Samsung', '2023-04-10', '2023-04-20', '2024-04-20');

INSERT INTO products (product\_id, serial\_number, product\_type, product\_name, manufacturer, purchase\_date, warranty\_start\_date, warranty\_end\_date)

VALUES ('P005', 'SN567890', 'TuLanh', 'Tủ lạnh Panasonic 450L', 'Panasonic', '2023-05-05', '2023-05-10', '2026-05-10');

c. Thêm dữ liệu vào bảng customers

INSERT INTO customers (customer\_id, name, address, phone\_number)

VALUES

('C001', 'Nguyễn Văn A', '123 Đường 1, Quận 1, TP.HCM', '0901234567');

INSERT INTO customers (customer\_id, name, address, phone\_number)

VALUES ('C002', 'Trần Thị B', '456 Đường 2, Quận 2, TP.HCM', '0912345678');

INSERT INTO customers (customer\_id, name, address, phone\_number)

VALUES ('C003', 'Lê Văn C', '789 Đường 3, Quận 3, TP.HCM', '0923456789');

INSERT INTO customers (customer\_id, name, address, phone\_number)

VALUES ('C004', 'Phạm Thị D', '321 Đường 4, Quận 4, TP.HCM', '0934567890');

INSERT INTO customers (customer\_id, name, address, phone\_number)

VALUES ('C005', 'Đỗ Văn E', '654 Đường 5, Quận 5, TP.HCM', '0945678901');

d. Thêm dữ liệu vào bảng customer\_products

INSERT INTO customer\_products (customer\_id, product\_id)

VALUES

('C001', 'P001');

INSERT INTO customer\_products (customer\_id, product\_id)

VALUES ('C002', 'P002');

INSERT INTO customer\_products (customer\_id, product\_id)

VALUES ('C003', 'P003');

INSERT INTO customer\_products (customer\_id, product\_id)

VALUES ('C001', 'P004');

INSERT INTO customer\_products (customer\_id, product\_id)

VALUES ('C004', 'P005');

e. Thêm dữ liệu vào bảng warranty\_claims

INSERT INTO warranty\_claims (claim\_id, customer\_id, product\_id, claim\_date, issue\_description, claim\_status)

VALUES

('CL001', 'C001', 'P001', '2023-02-01', 'Màn hình bị nứt', 'Đang xử lý');

INSERT INTO warranty\_claims (claim\_id, customer\_id, product\_id, claim\_date, issue\_description, claim\_status)

VALUES ('CL002', 'C002', 'P002', '2023-03-10', 'Pin yếu', 'Đã hoàn thành');

INSERT INTO warranty\_claims (claim\_id, customer\_id, product\_id, claim\_date, issue\_description, claim\_status)

VALUES ('CL003', 'C003', 'P003', '2023-04-15', 'Nước rò rỉ', 'Đang xử lý');

INSERT INTO warranty\_claims (claim\_id, customer\_id, product\_id, claim\_date, issue\_description, claim\_status)

VALUES ('CL004', 'C004', 'P004', '2023-05-20', 'Âm thanh không hoạt động', 'Chờ phản hồi');

INSERT INTO warranty\_claims (claim\_id, customer\_id, product\_id, claim\_date, issue\_description, claim\_status)

VALUES ('CL005', 'C005', 'P005', '2023-06-25', 'Tủ lạnh không lạnh', 'Đang xử lý');

f. Thêm dữ liệu vào bảng repair\_history

INSERT INTO repair\_history (repair\_id, product\_id, repair\_date, repair\_description, status, technician)

VALUES

('R001', 'P001', '2023-02-10', 'Thay màn hình', 'Hoàn thành', 'Nguyễn Văn T');

INSERT INTO repair\_history (repair\_id, product\_id, repair\_date, repair\_description, status, technician)

VALUES ('R002', 'P002', '2023-03-15', 'Thay pin', 'Hoàn thành', 'Trần Văn S');

INSERT INTO repair\_history (repair\_id, product\_id, repair\_date, repair\_description, status, technician)

VALUES ('R003', 'P003', '2023-04-20', 'Sửa nước rò rỉ', 'Hoàn thành', 'Lê Văn M');

INSERT INTO repair\_history (repair\_id, product\_id, repair\_date, repair\_description, status, technician)

VALUES ('R004', 'P004', '2023-05-30', 'Sửa âm thanh', 'Đang xử lý', 'Phạm Văn Q');

INSERT INTO repair\_history (repair\_id, product\_id, repair\_date, repair\_description, status, technician)

VALUES ('R005', 'P005', '2023-07-10', 'Thay block lạnh', 'Hoàn thành', 'Đỗ Văn R');

g. Thêm dữ liệu vào bảng warranty\_policies

INSERT INTO warranty\_policies (policy\_id, product\_type, policy\_description, warranty\_duration, conditions)

VALUES

('P001', 'Laptop', 'Bảo hành 1 năm', 12, 'Bảo hành cho lỗi kỹ thuật');

INSERT INTO warranty\_policies (policy\_id, product\_type, policy\_description, warranty\_duration, conditions)

VALUES ('P002', 'DienThoai', 'Bảo hành 1 năm', 12, 'Bảo hành cho lỗi phần cứng');

INSERT INTO warranty\_policies (policy\_id, product\_type, policy\_description, warranty\_duration, conditions)

VALUES ('P003', 'MayGiat', 'Bảo hành 2 năm', 24, 'Bảo hành cho lỗi kỹ thuật');

INSERT INTO warranty\_policies (policy\_id, product\_type, policy\_description, warranty\_duration, conditions)

VALUES ('P004', 'Tivi', 'Bảo hành 1 năm', 12, 'Bảo hành cho lỗi phần cứng');

INSERT INTO warranty\_policies (policy\_id, product\_type, policy\_description, warranty\_duration, conditions)

VALUES ('P005', 'TuLanh', 'Bảo hành 3 năm', 36, 'Bảo hành cho lỗi kỹ thuật');

h. Thêm dữ liệu vào bảng extended\_warranty

INSERT INTO extended\_warranty (extended\_warranty\_id, product\_id, extension\_fee, new\_warranty\_end\_date)

VALUES

('EW001', 'P001', 500000, '2024-02-01');

INSERT INTO extended\_warranty (extended\_warranty\_id, product\_id, extension\_fee, new\_warranty\_end\_date)

VALUES ('EW002', 'P002', 300000, '2024-03-10');

INSERT INTO extended\_warranty (extended\_warranty\_id, product\_id, extension\_fee, new\_warranty\_end\_date)

VALUES ('EW003', 'P003', 600000, '2025-04-15');

INSERT INTO extended\_warranty (extended\_warranty\_id, product\_id, extension\_fee, new\_warranty\_end\_date)

VALUES ('EW004', 'P004', 400000, '2024-05-20');

INSERT INTO extended\_warranty (extended\_warranty\_id, product\_id, extension\_fee, new\_warranty\_end\_date)

VALUES ('EW005', 'P005', 700000, '2026-06-25');

### Các câu truy vấn

a. Lấy thông tin ngày sửa, thông tin sửa, tiến độ và người phụ trách của mã sản phẩm R003

SELECT repair\_date, repair\_description, status, technician FROM repair\_history WHERE repair\_id = 'R003'

b. Truy vấn 2(Cập nhật lại tiến độ 'Đang xử lý' cho mã sữa chữa R004)

--Đầu tiên lấy dữ liệu của bảng repair\_history

Select \* from repair\_history;

--Bắt đầu câu truy vấn như yêu cầu.

UPDATE repair\_history

Set status = 'Đang xử lý'

WHERE repair\_id = 'R004'

c. Lấy tất cả thông tin lịch sử sửa chữa của product\_type = 'Dien thoai'

--Đầu tiên lấy thông tin product\_id và product\_name của product\_type = ‘Dien thoai’

SELECT product\_id, product\_name

FROM products

WHERE product\_type = 'DienThoai';

--Tiếp theo xuất thông tin của lịch sửa chữa theo mã product\_id tương ứng với Dien thoai.

SELECT \*

FROM repair\_history

WHERE product\_id = 'P002'

ALLOW FILTERING;

d. Lấy ra thông tin sản phẩm và ngày mua của bà Phạm Thị D

--Đầu tiên lấy được mã khách hàng của bà Phạm Thị D

SELECT customer\_id

FROM customers

WHERE name = 'Phạm Thị D'

ALLOW FILTERING;

--Sau đó dựa vào mã khách hàng của bà Phạm Thị D lấy ra được mã sản phẩm

SELECT \*

FROM customer\_products

WHERE customer\_id = 'C004'

ALLOW FILTERING;

--Từ mã sản phẩm có được ta lấy ra sản phẩm, và ngày mua của Phạm Thị D.

SELECT product\_type, product\_name, purchase\_date

FROM products

WHERE product\_id = 'P005'

ALLOW FILTERING;

### Sao lưu phục hồi dữ liệu

Để sao lưu và phục hồi dữ liệu trong Cassandra, bạn có thể sử dụng snapshot cho quá trình sao lưu và nạp lại snapshot để phục hồi dữ liệu. Quá trình này không chỉ lưu trữ dữ liệu, mà còn giữ nguyên trạng thái nhất quán của cơ sở dữ liệu tại thời điểm snapshot.

\*Sao lưu:

* Truy cập vào node Cassandra và sử dụng lệnh sau để tạo snapshot:
  + nodetool snapshot <tên\_keyspace> -t <tên\_snapshot>
  + Ví dụ: nodetool snapshot my\_keyspace -t snapshot\_2024\_10\_19
* Có thể xem danh sách các snapshot đã tạo, sử dụng lệnh:
  + nodetool listsnapshots
* Snapshot sẽ được lưu dưới dạng file trong thư mục snapshots.

\*Phục hồi:

* Nếu bạn muốn phục hồi dữ liệu từ snapshot và thay thế dữ liệu hiện tại, trước hết bạn có thể xóa các file SSTable hiện tại trong thư mục dữ liệu dùng lệnh sau:

rm -rf /var/lib/cassandra/data/<tên\_keyspace>/<tên\_bảng>/\*.

* Sau khi xóa dữ liệu hiện tại, bạn có thể sao chép các file từ thư mục snapshot về lại thư mục dữ liệu chính của Cassandra dùng lệnh sau:

cp-R /var/lib/cassandra/data/<tên\_keyspace>/<tên\_bảng>/snapshots/<tên\_snapshot>/\* /var/lib/cassandra/data/<tên\_keyspace>/<tên\_bảng>/

* Sau khi sao chép xong, bạn cần reload lại node Cassandra để áp dụng dữ liệu từ snapshot dùng lệnh như sau:

nodetool refresh <tên\_keyspace> <tên\_bảng>

# Chức năng của hệ thống

## A screenshot of a computer Description automatically generatedForm trang chủ

Hình 12:Chức năng đăng nhập

Form đăng nhập này là một phần của hệ thống xác thực dành riêng cho quản trị viên trong việc quản lý bảo hành sản phẩm. Chỉ có tài khoản có quyền quản trị viên mới được phép đăng nhập để đảm bảo truy cập an toàn vào các chức năng quan trọng của hệ thống. Dưới đây là mô tả chi tiết về chức năng của form đăng nhập:

Trường Tài khoản:

* + Đây là trường nhập liệu, nơi người dùng, cụ thể là quản trị viên, nhập tên tài khoản của mình.
  + Hệ thống sẽ kiểm tra tên tài khoản đã nhập trong cơ sở dữ liệu để xác minh xem người dùng có quyền quản trị hay không.

Trường Mật khẩu:

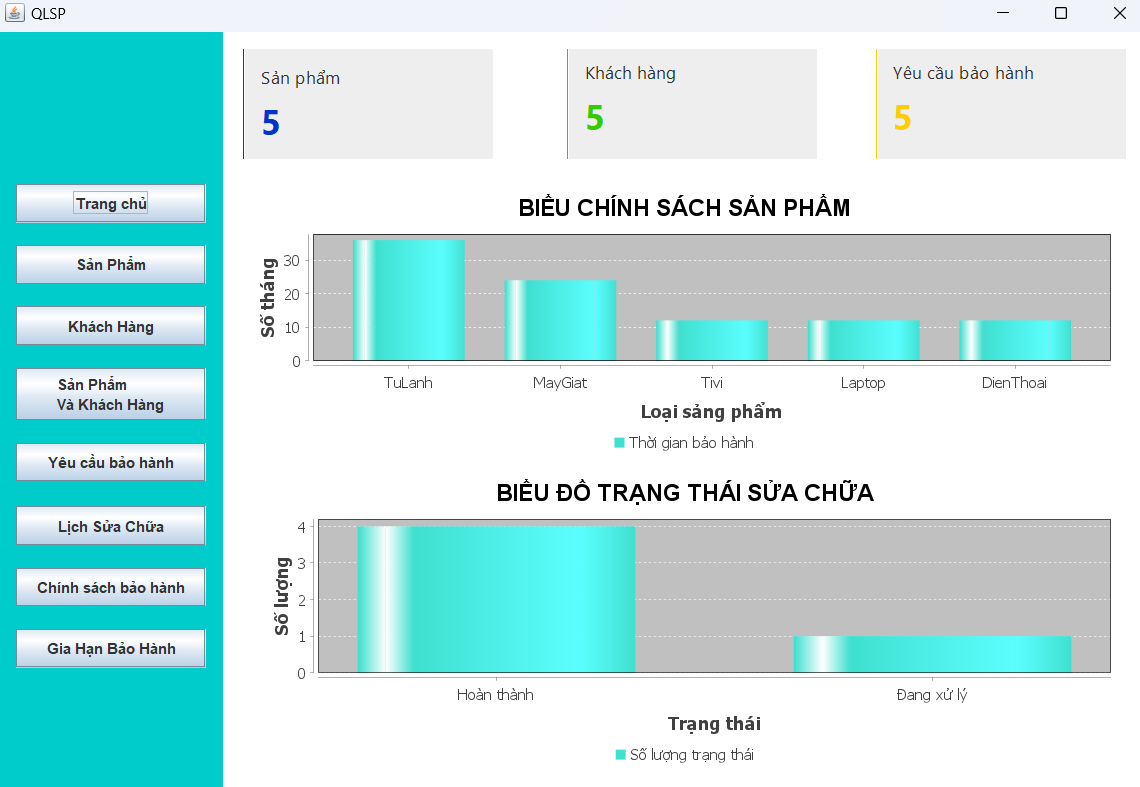
* + Trường này dùng để nhập mật khẩu tương ứng với tài khoản quản trị viên.
  + Mật khẩu sẽ được ẩn (hiển thị dưới dạng dấu chấm hoặc dấu sao) để đảm bảo bảo mật.
  + Hệ thống sẽ xác thực mật khẩu bằng cách so sánh mật khẩu đã nhập với mật khẩu đã mã hóa được lưu trong cơ sở dữ liệu.

Nút Đăng nhập:

* + Khi người dùng nhấn nút này, thông tin tài khoản và mật khẩu sẽ được gửi tới máy chủ.
  + Hệ thống sẽ kiểm tra tính hợp lệ của tài khoản và mật khẩu, sau đó cho phép truy cập vào khu vực quản trị của hệ thống bảo hành sản phẩm nếu thông tin đúng.
  + Nếu thông tin không chính xác hoặc tài khoản không có quyền quản trị, hệ thống sẽ hiển thị thông báo lỗi.

Form này đảm bảo chỉ có những người dùng được ủy quyền (quản trị viên) mới có thể truy cập vào các chức năng quản lý bảo hành sản phẩm, góp phần bảo mật và đảm bảo tính toàn vẹn cho hệ thống.

## Form trang chủ



Hình 13: Giao diện form trang chủ

Form Trang chủ của hệ thống quản lý bảo hành sản phẩm này là giao diện tổng quan, cung cấp cái nhìn toàn diện về tình trạng sản phẩm, khách hàng và các yêu cầu bảo hành hiện tại. Nó tích hợp các biểu đồ minh họa rõ ràng về chính sách bảo hành và tình trạng sửa chữa, hỗ trợ người dùng theo dõi và quản lý hệ thống một cách hiệu quả. Dưới đây là mô tả chi tiết từng phần:

### Tổng quan số liệu chính

Trên cùng của giao diện Trang chủ, các thông số quan trọng được hiển thị rõ ràng:

* **Sản phẩm**: Hiển thị tổng số lượng sản phẩm hiện có trong hệ thống.
* **Khách hàng**: Thể hiện tổng số khách hàng đã được quản lý.
* **Yêu cầu bảo hành**: Thể hiện số lượng yêu cầu bảo hành hiện tại.

Phần này giúp người quản trị dễ dàng theo dõi số liệu tổng quan về sản phẩm, khách hàng và các yêu cầu bảo hành một cách nhanh chóng mà không cần phải tra cứu chi tiết từng mục.

### Biểu đồ chính sách sản phẩm

Biểu đồ đầu tiên mang tên **Biểu chỉnh sách sản phẩm**. Nó thể hiện thời gian bảo hành của từng loại sản phẩm dưới dạng biểu đồ cột:

* Các loại sản phẩm như Tủ lạnh, Máy giặt, Tivi, Laptop, và Điện thoại được liệt kê trên trục ngang.
* Thời gian bảo hành (số tháng) được hiển thị trên trục dọc.

Thông qua biểu đồ này, người dùng có thể dễ dàng nắm bắt thời gian bảo hành trung bình của từng loại sản phẩm, từ đó giúp hỗ trợ việc theo dõi và điều chỉnh chính sách bảo hành phù hợp với từng sản phẩm cụ thể.

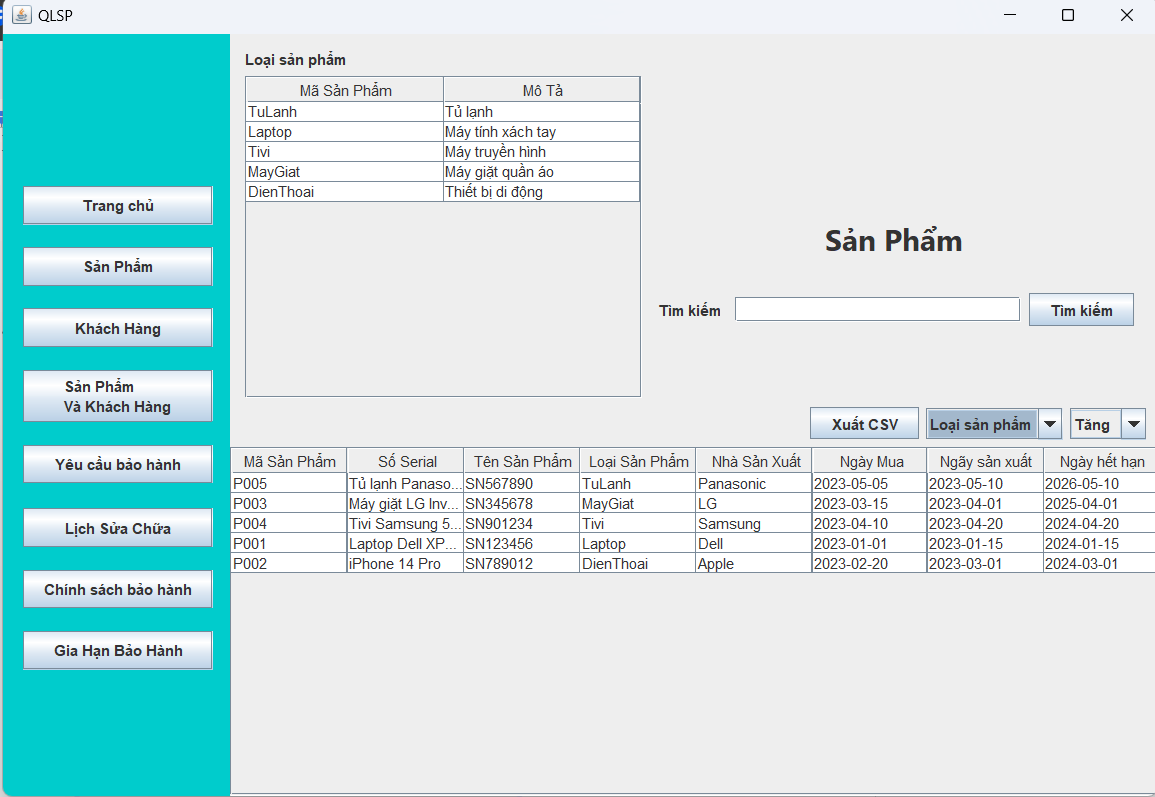
### Biểu đồ trạng thái sửa chữa

Biểu đồ thứ hai là **Biểu đồ trạng thái sửa chữa**, thể hiện số lượng yêu cầu bảo hành và trạng thái sửa chữa hiện tại của sản phẩm:

* Các trạng thái như "Hoàn thành" và "Đang xử lý" được minh họa dưới dạng cột.
* Trục ngang thể hiện trạng thái, trục dọc thể hiện số lượng yêu cầu tương ứng.

Biểu đồ này giúp người quản lý theo dõi tình trạng các yêu cầu bảo hành một cách trực quan, từ đó đưa ra các quyết định kịp thời trong việc xử lý yêu cầu và đảm bảo chất lượng dịch vụ bảo hành.

## Form sản phẩm



Hình 14: Giao diện chức năng form sản phẩm

Form quản lý sản phẩm này là một phần không thể thiếu trong hệ thống quản lý bảo hành sản phẩm, giúp người dùng dễ dàng quản lý và tra cứu các thông tin sản phẩm. Với giao diện trực quan, form này tích hợp nhiều tính năng thông minh và tiện lợi, hỗ trợ người quản trị trong việc theo dõi trạng thái bảo hành cũng như thông tin chi tiết của từng sản phẩm.

### Danh sách loại sản phẩm

Ở góc trên bên trái của form, phần **Loại sản phẩm** cung cấp một danh sách các loại hàng hóa tiêu biểu như "Tủ lạnh", "Máy tính xách tay", "Máy giặt", "Tivi" và "Thiết bị di động". Khi người dùng chọn một loại sản phẩm từ danh sách này, bảng dữ liệu bên dưới sẽ tự động lọc và hiển thị những sản phẩm thuộc loại đã chọn. Chức năng này giúp người dùng nhanh chóng tập trung vào các sản phẩm cụ thể mà họ cần tìm kiếm hoặc quản lý, đặc biệt khi hệ thống chứa đựng số lượng lớn các sản phẩm khác nhau.

### Chức năng tìm kiếm theo số serial

Ngay bên cạnh danh sách loại sản phẩm là phần **Tìm kiếm**. Đây là nơi người dùng có thể nhập vào **số serial** của một sản phẩm cụ thể để nhanh chóng tra cứu thông tin chi tiết liên quan đến sản phẩm đó. Thông qua số serial duy nhất, người quản lý có thể xác định rõ sản phẩm bao gồm tên sản phẩm, loại sản phẩm, ngày mua, ngày sản xuất, nhà sản xuất, và đặc biệt là **ngày hết hạn bảo hành**. Tính năng này rất quan trọng trong việc hỗ trợ quá trình kiểm tra và quản lý bảo hành, đảm bảo rằng không có trường hợp sản phẩm hết hạn bảo hành mà chưa được xử lý kịp thời.

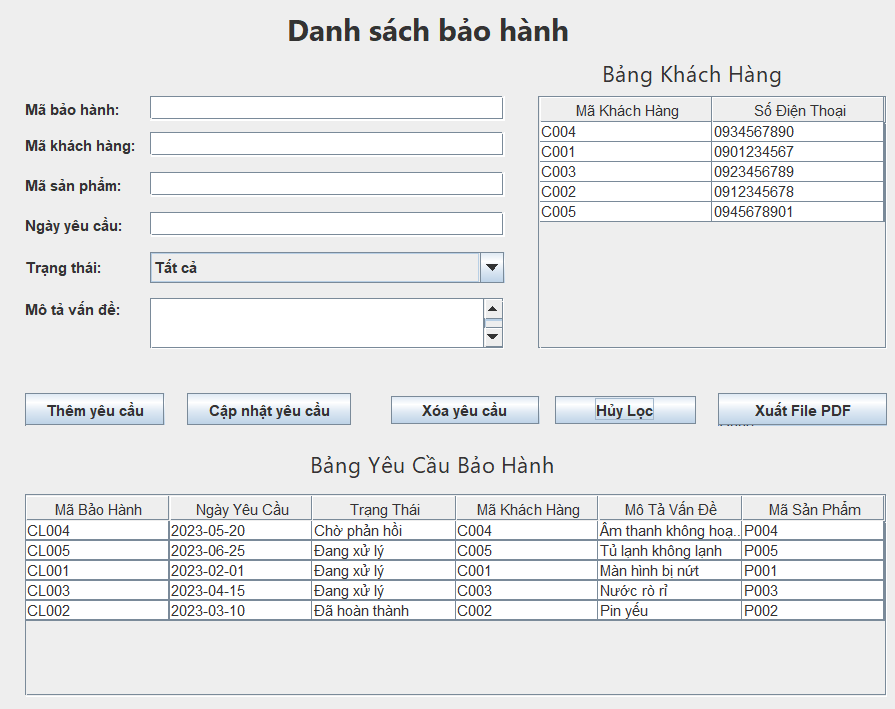
### Xuất dữ liệu dưới dạng CSV

Một tính năng hữu ích khác của form này là nút **Xuất CSV**, cho phép người dùng xuất toàn bộ dữ liệu về sản phẩm dưới dạng một tệp CSV (Comma-Separated Values). Tệp CSV là định dạng chuẩn để lưu trữ và chia sẻ dữ liệu, giúp người quản lý dễ dàng truyền tải và sử dụng thông tin sản phẩm trên các hệ thống khác nhau hoặc để làm tài liệu báo cáo. Các thông tin bao gồm trong tệp CSV thường bao gồm: mã sản phẩm, số serial, tên sản phẩm, loại sản phẩm, nhà sản xuất, ngày mua, ngày sản xuất, và ngày hết hạn bảo hành. Chức năng này tối ưu hóa quy trình quản lý và cung cấp sự linh hoạt trong việc xử lý dữ liệu.

### Chức sắp xếp dữ liệu

Để tối ưu hóa việc quản lý và tra cứu dữ liệu, form còn cung cấp chức năng **lọc và sắp xếp** các sản phẩm theo nhiều tiêu chí khác nhau. Người dùng có thể sắp xếp danh sách sản phẩm dựa trên các trường như **Loại sản phẩm**, **Nhà sản xuất**, **Ngày mua**, **Ngày sản xuất**, **Ngày hết hạn bảo hành**, và **Tên sản phẩm**. Chức năng này không chỉ giúp tìm kiếm dữ liệu nhanh chóng mà còn cho phép sắp xếp thông tin theo thứ tự **tăng dần** hoặc **giảm dần**. Điều này đặc biệt hữu ích trong việc theo dõi tình trạng bảo hành, khi người quản lý có thể dễ dàng ưu tiên những sản phẩm sắp hết hạn bảo hành để kịp thời xử lý, hoặc lọc những sản phẩm mới nhất để theo dõi doanh số và thời gian sử dụng.

## Form danh sách bảo hành



Hình 15: .Giao diện chức năng form danh sách bảo hành

Form quản lý yêu cầu bảo hành là một phần quan trọng trong hệ thống quản lý bảo hành sản phẩm, giúp người dùng dễ dàng theo dõi, quản lý và xử lý các yêu cầu bảo hành của khách hàng.

### Thêm yêu cầu bảo hành

Người dùng có thể dễ dàng tạo mới một yêu cầu bảo hành bằng cách nhập đầy đủ các thông tin cần thiết. Đầu tiên, mã bảo hành được nhập để định danh duy nhất cho yêu cầu, giúp việc tra cứu và theo dõi yêu cầu trở nên thuận tiện hơn trong hệ thống. Tiếp theo là mã khách hàng, cho phép liên kết yêu cầu bảo hành với một khách hàng cụ thể, đảm bảo tính chính xác khi truy xuất thông tin. Mã sản phẩm cũng được nhập để xác định rõ ràng sản phẩm cần được bảo hành, từ đó hệ thống có thể quản lý tốt hơn từng sản phẩm riêng lẻ. Ngày yêu cầu bảo hành là một trường quan trọng, cung cấp thông tin về thời điểm khách hàng gửi yêu cầu, từ đó giúp người quản lý theo dõi quá trình xử lý theo thời gian. Ngoài ra, trạng thái xử lý cho phép người dùng chọn tình trạng hiện tại của yêu cầu, chẳng hạn như "Đang xử lý," "Hoàn thành," hoặc "Đã hủy." Cuối cùng, mô tả chi tiết về vấn đề gặp phải được yêu cầu để người quản lý có thể nắm bắt rõ ràng vấn đề của sản phẩm, từ đó đưa ra giải pháp phù hợp và nhanh chóng.

### Cập nhật yêu cầu bảo hành

Sau khi yêu cầu bảo hành đã được tạo, chức năng cập nhật cho phép người dùng chỉnh sửa các thông tin liên quan khi có thay đổi hoặc tiến độ mới trong quá trình xử lý. Người dùng có thể thay đổi trạng thái xử lý, chẳng hạn như từ "Đang xử lý" sang "Hoàn thành" khi sản phẩm đã được bảo hành xong. Điều này giúp hệ thống luôn có thông tin chính xác và kịp thời về trạng thái của từng yêu cầu. Không chỉ trạng thái, người dùng còn có thể cập nhật mô tả chi tiết nếu phát hiện thêm các vấn đề khác trong quá trình kiểm tra sản phẩm hoặc khi có thêm thông tin từ khách hàng. Nhờ vào tính năng này, các yêu cầu bảo hành sẽ luôn được theo dõi và quản lý một cách chính xác và toàn diện, giúp đảm bảo quá trình bảo hành được diễn ra suôn sẻ và hiệu quả.

### Xóa yêu cầu bảo hành

Chức năng xóa yêu cầu bảo hành cho phép người dùng loại bỏ các yêu cầu bảo hành không còn cần thiết hoặc đã bị hủy. Người dùng có thể chọn một hoặc nhiều yêu cầu từ danh sách để xóa chúng khỏi hệ thống. Điều này giúp duy trì danh sách yêu cầu bảo hành gọn gàng và chỉ chứa những yêu cầu thực sự cần theo dõi. Tính năng này đặc biệt hữu ích trong các trường hợp khi yêu cầu bảo hành đã được xử lý từ lâu hoặc khi có yêu cầu bảo hành bị nhập nhầm hoặc không hợp lệ. Việc loại bỏ các yêu cầu không còn giá trị sẽ giúp hệ thống hoạt động hiệu quả hơn và giúp người quản lý dễ dàng tập trung vào các yêu cầu cần thiết hiện tại.

### Hủy lọc

Chức năng hủy lọc cung cấp một cách tiện lợi cho người dùng khi họ muốn xem lại toàn bộ danh sách yêu cầu bảo hành mà không áp dụng bất kỳ tiêu chí lọc nào. Khi người dùng đã sử dụng bộ lọc để tìm kiếm các yêu cầu theo mã bảo hành, mã khách hàng, trạng thái hoặc các tiêu chí khác, họ có thể dùng chức năng này để xóa bỏ tất cả các bộ lọc đó và quay lại danh sách đầy đủ. Điều này rất hữu ích khi người dùng cần xem tổng quan các yêu cầu bảo hành mà không cần phải đi qua các tiêu chí lọc cụ thể, đặc biệt khi họ đang tìm kiếm dữ liệu tổng hợp hoặc muốn xem xét lại các yêu cầu từ đầu.

### Xuất file PDF

Chức năng xuất file PDF cho phép người dùng nhanh chóng lưu trữ và chia sẻ thông tin yêu cầu bảo hành dưới dạng tệp PDF. Sau khi người dùng đã chọn các yêu cầu bảo hành cần xuất hoặc lọc dữ liệu theo các tiêu chí cụ thể, họ có thể xuất toàn bộ thông tin đó thành một tệp PDF. Điều này tạo ra một bản ghi dễ dàng lưu trữ hoặc gửi qua email cho khách hàng hoặc các bên liên quan khác. Chức năng này rất hữu ích trong việc quản lý và lưu trữ dữ liệu dưới dạng tài liệu chính thức, giúp đảm bảo thông tin luôn có sẵn và có thể truy xuất khi cần thiết.

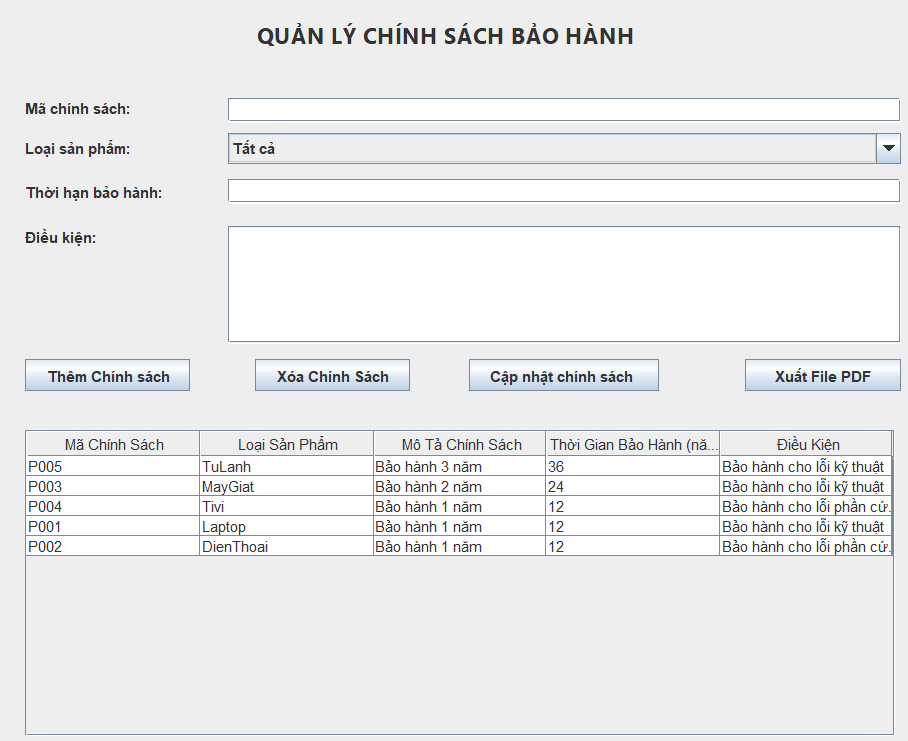
### Lọc bằng cách chọn khách hàng và lọc theo trạng thái

Giao diện bao gồm một bảng danh sách khách hàng, giúp người dùng dễ dàng chọn lựa và xem thông tin của từng khách hàng cụ thể. Điều này rất quan trọng khi người dùng cần lọc các yêu cầu bảo hành liên quan đến một khách hàng cụ thể, giúp quá trình tìm kiếm và xử lý yêu cầu trở nên chính xác hơn. Người dùng có thể chọn một khách hàng từ bảng, và ngay lập tức, hệ thống sẽ hiển thị các yêu cầu bảo hành liên quan đến khách hàng đó. Ngoài ra, còn có thể lọc theo trạng tháng mỗi khi chọn một trạng thái khác thì bảng danh sách bảo hành sẽ hiển thị ra các dòng có trạng thái giống với dữ liệu trên thanh combobox. Chức năng này giúp tăng cường khả năng quản lý dữ liệu khách hàng và đảm bảo việc theo dõi yêu cầu bảo hành được thực hiện theo từng khách hàng một cách rõ ràng và dễ dàng.

### Quản lý danh sách yêu cầu bảo hành

Ở phần cuối giao diện, bảng hiển thị danh sách yêu cầu bảo hành cung cấp một cái nhìn tổng quan về tất cả các yêu cầu đã được tạo. Bảng này chứa các thông tin quan trọng như mã bảo hành, ngày yêu cầu, trạng thái hiện tại, mã khách hàng, mô tả vấn đề, và mã sản phẩm. Người dùng có thể dễ dàng xem qua danh sách này và chọn các yêu cầu cần thiết để chỉnh sửa hoặc xóa bỏ. Bảng danh sách giúp người quản lý nhanh chóng tìm thấy các yêu cầu bảo hành cần được xử lý, theo dõi các trạng thái yêu cầu, và đảm bảo rằng không có yêu cầu nào bị bỏ sót. Người dùng có thể nhấp vào các dòng của bảng thì dữ liệu của dòng đó sẽ được hiển thị trên các thành phần giao diện ở trên. Việc quản lý danh sách này là một phần quan trọng trong quá trình bảo hành, giúp nâng cao hiệu suất công việc và đảm bảo rằng mọi yêu cầu đều được xử lý một cách hiệu quả.

## Form chính sách bảo hành



Hình 16:Giao diện chức năng form chính sách bảo hành

### Thêm chính sách bảo hành

Thêm chính sách bảo hành là một trong những chức năng quan trọng của giao diện này. Người dùng có thể nhập đầy đủ các thông tin liên quan đến chính sách bảo hành như mã chính sách, loại sản phẩm, thời hạn bảo hành, và điều kiện áp dụng. Mã chính sách là một trường bắt buộc, giúp xác định rõ ràng từng chính sách bảo hành. Người dùng cũng sẽ chọn loại sản phẩm cụ thể để liên kết với chính sách này, điều này giúp hệ thống quản lý hiệu quả hơn. Thời hạn bảo hành, với định dạng ngày tháng, được sử dụng để xác định khoảng thời gian mà chính sách này có hiệu lực. Điều kiện áp dụng cho chính sách là phần mô tả chi tiết về những trường hợp nào sản phẩm sẽ được bảo hành, giúp tránh các tranh cãi sau này khi chính sách được thực thi. Việc nhập đầy đủ các thông tin này không chỉ giúp hệ thống lưu trữ mà còn giúp người dùng và khách hàng dễ dàng theo dõi các chính sách liên quan.

### Cập nhật chính sách bảo hành

Cập nhật chính sách bảo hành là chức năng cho phép người dùng chỉnh sửa các chính sách bảo hành sau khi chúng đã được tạo. Đây là một công cụ quan trọng để đảm bảo rằng thông tin luôn được duy trì và cập nhật kịp thời. Người dùng có thể điều chỉnh lại loại sản phẩm áp dụng nếu cần, cập nhật thời hạn bảo hành khi có những thay đổi về thời gian bảo hành, hoặc thay đổi điều kiện bảo hành nếu chính sách cần được điều chỉnh để phù hợp với các điều khoản mới. Điều này đặc biệt hữu ích khi doanh nghiệp thay đổi quy trình bảo hành hoặc khi có thêm sản phẩm mới được áp dụng chính sách bảo hành này. Việc cập nhật chính sách dễ dàng giúp doanh nghiệp duy trì tính chính xác và kịp thời của hệ thống bảo hành.

### Xóa chính sách bảo hành

Xóa chính sách bảo hành là một tính năng giúp quản lý hệ thống bảo hành hiệu quả hơn. Khi một chính sách không còn hiệu lực hoặc không cần thiết nữa, người dùng có thể dễ dàng xóa nó khỏi danh sách. Chức năng này đặc biệt hữu ích khi hệ thống bắt đầu có quá nhiều chính sách và cần phải dọn dẹp để giữ cho danh sách gọn gàng và dễ quản lý. Khi chọn xóa một chính sách, người dùng chỉ cần chọn mã chính sách tương ứng từ bảng danh sách và thực hiện thao tác xóa nhanh chóng.

### Xuất file PDF

Xuất file PDF là chức năng tiện lợi giúp người dùng xuất thông tin chính sách bảo hành thành file PDF để lưu trữ hoặc chia sẻ. Sau khi nhập hoặc cập nhật thông tin về các chính sách bảo hành, người dùng có thể dễ dàng xuất toàn bộ hoặc các chính sách cụ thể dưới dạng file PDF. Chức năng này đặc biệt hữu ích khi người dùng cần tạo các báo cáo bảo hành để chia sẻ với khách hàng hoặc lưu trữ dưới dạng tài liệu chính thức. Xuất file PDF không chỉ đảm bảo sự chuyên nghiệp trong việc lưu trữ thông tin mà còn giúp thông tin được bảo mật tốt hơn dưới dạng tệp tài liệu không thể chỉnh sửa.

### Xem danh sách chính sách bảo hành

Xem và chọn khách hàng là một tính năng tích hợp bảng danh sách khách hàng ở phía bên phải của giao diện. Bảng này hiển thị danh sách các mã khách hàng và thông tin liên hệ như số điện thoại, giúp người dùng dễ dàng chọn lựa khách hàng để liên kết với yêu cầu bảo hành hoặc chính sách bảo hành. Điều này giúp tăng tốc độ trong quá trình tìm kiếm và xử lý các yêu cầu bảo hành, đặc biệt khi hệ thống có một số lượng lớn khách hàng.

### Lọc chính sách theo loại sản phẩm

Lọc chính sách bảo hành theo loại sản phẩm giúp người dùng dễ dàng chọn lựa

và xem thông tin của từng sản phẩm một cách dễ dàng hơn. Điều này giúp tăng tốc độ trong quá trình tìm kiếm.

## Form khách hàng

A screenshot of a computer

Description automatically generated

Hình 17:Giao diện chức năng form khách hàng

### Chức năng Thêm mới khách hàng:

Khi người dùng có nhu cầu bổ sung thông tin của một khách hàng mới vào hệ thống, họ sẽ tiến hành nhập đầy đủ các thông tin cá nhân cần thiết vào các trường dữ liệu tương ứng trên giao diện phần mềm. Sau khi hoàn tất việc nhập liệu, hệ thống sẽ tiến hành kiểm tra tính hợp lệ của dữ liệu. Nếu dữ liệu hợp lệ, hệ thống sẽ tạo một bản ghi mới đại diện cho khách hàng đó và lưu trữ vào cơ sở dữ liệu.

### Chức năng Cập nhật thông tin khách hàng:

Để chỉnh sửa hoặc bổ sung thông tin cho một khách hàng đã tồn tại trong hệ thống, người dùng chỉ cần chọn khách hàng cần cập nhật và tiến hành sửa đổi các thông tin trên giao diện. Sau khi hoàn tất quá trình chỉnh sửa, hệ thống sẽ cập nhật lại thông tin của khách hàng vào cơ sở dữ liệu.

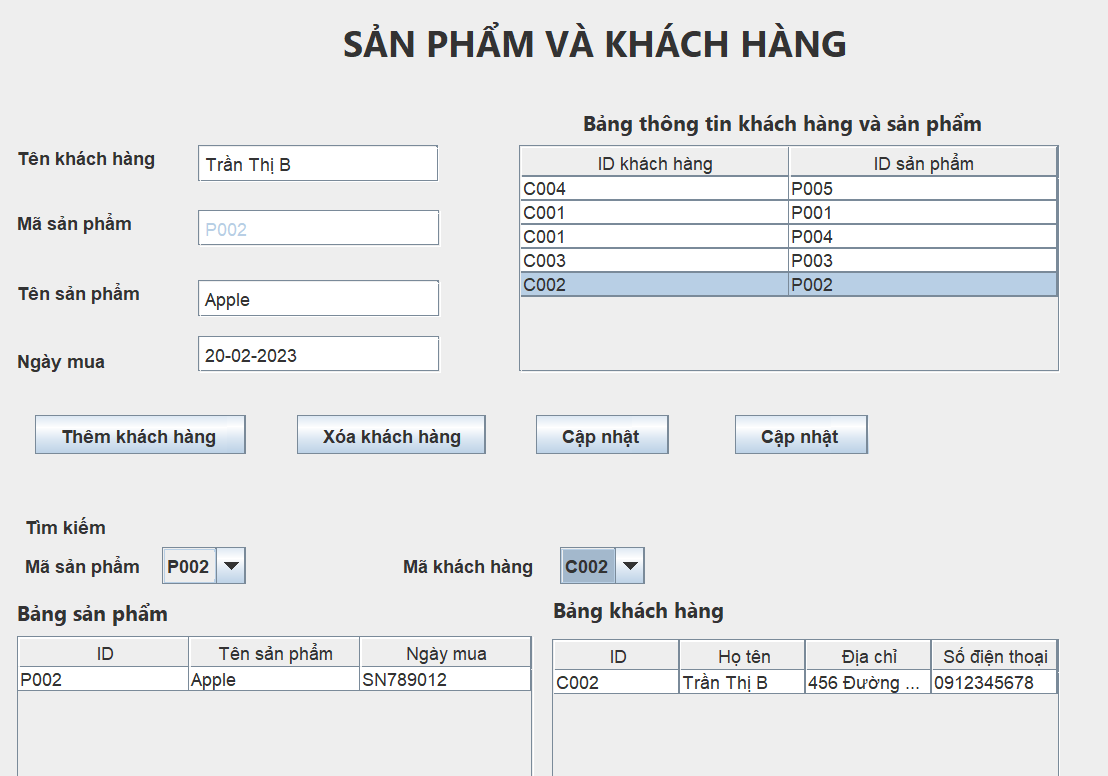
### Chức năng Xóa khách hàng:

Trong trường hợp cần loại bỏ thông tin của một khách hàng khỏi hệ thống, người dùng có thể chọn khách hàng đó và xác nhận yêu cầu xóa. Hệ thống sẽ tiến hành xóa bản ghi tương ứng của khách hàng khỏi cơ sở dữ liệu. Tuy nhiên, để tránh việc xóa nhầm, hệ thống thường yêu cầu người dùng xác nhận lại một lần nữa trước khi thực hiện thao tác xóa.

### Chức năng Xuất dữ liệu khách hàng sang định dạng JSON:

Để tạo báo cáo hoặc chia sẻ dữ liệu khách hàng với các hệ thống khác, người dùng có thể yêu cầu hệ thống xuất toàn bộ danh sách khách hàng hoặc một nhóm khách hàng cụ thể sang định dạng JSON. JSON (JavaScript Object Notation) là một định dạng trao đổi dữ liệu phổ biến, có cấu trúc dễ đọc và dễ xử lý. Khi xuất dữ liệu sang định dạng này, hệ thống sẽ chuyển đổi toàn bộ thông tin của các khách hàng được chọn thành một chuỗi dữ liệu tuân theo chuẩn JSON, sau đó lưu vào một file có phần mở rộng .json.

## Form Sản phẩm và khách hàng



Hình 18: Giao diện chức năng form sản phẩm và khách hàng

### Hiển thị thông tin:

* Khi form được load lần đầu, hệ thống sẽ hiển thị danh sách các sản phẩm và khách hàng hiện có trong cơ sở dữ liệu.
* Khi click vào một dòng trong bảng "customer\_product", hệ thống sẽ:
  + Đưa thông tin của sản phẩm và khách hàng tương ứng vào các trường nhập liệu ở phần trên.
  + Cập nhật dữ liệu trong các bảng sản phẩm và khách hàng.

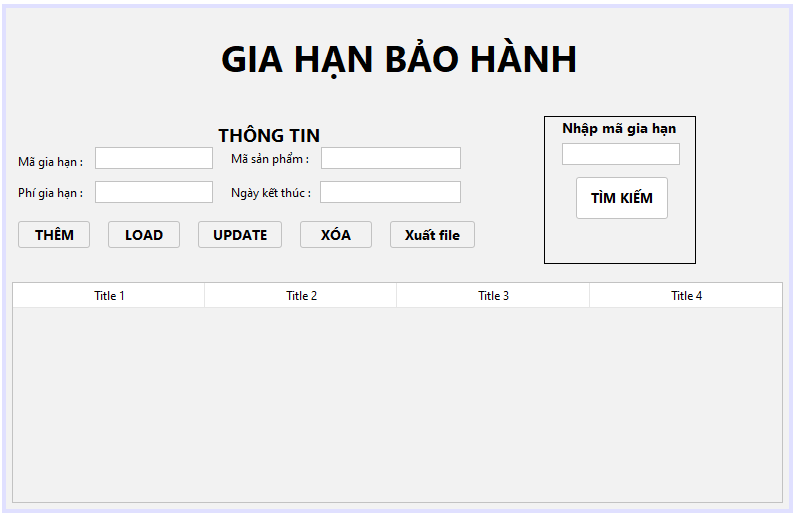
### Cập nhật thông tin

* Để chỉnh sửa hoặc bổ sung thông tin cho một khách hàng đã tồn tại trong hệ thống, khi người dùng chọn 1 hàng bất kỳ ở bảng sản phẩm và khách hàng thông tin của khách hàng đó sẽ được thêm vào các ô text, sau đó chúng ta sẽ tiến hành sửa đổi các thông tin trên giao diện. Sau khi hoàn tất quá trình chỉnh sửa, hệ thống sẽ cập nhật lại thông tin của khách hàng vào cơ sở dữ liệu tương ứng.

### Tìm kiếm

* Khi chọn một giá trị trong combobox "Mã sản phẩm", hệ thống sẽ lọc danh sách khách hàng và chỉ hiển thị những khách hàng đã mua sản phẩm đó trong bảng "Bảng khách hàng".
* Tương tự, khi chọn một giá trị trong combobox "Mã khách hàng", hệ thống sẽ lọc danh sách sản phẩm và chỉ hiển thị những sản phẩm mà khách hàng đó đã mua trong bảng "Bảng sản phẩm".

## Form gia hạn bảo hành



Hình 19:Giao diện chức năng form gia hạn bảo hành

### Thêm gia hạn bảo hành

Thêm gia hạn bảo hành giúp người dùng có thể nhập thông tin khi khách hàng muốn gia hạn thêm thời gian bảo hành của 1 sản phẩm. Người dùng có thể thêm (Mã gia hạn, Mã sản phẩm, Phí gia hạn, Ngày kết thúc gia hạn). Hệ thống sẽ tạo ra 1 vùng dữ liệu để lưu lại thông tin mà người dùng vừa nhập. Sau khi nhập xong và nhấn hoàn thành. Thì dưới dữ liệu sẽ thêm mới dòng có đầy đủ thông tin mà người dùng đã nhập.

### Load gia hạn bảo hành

Chức năng load gia hạn bảo hành cho phép người dùng sắp xếp lại dữ liệu theo mã gia hạn(từ nhỏ đến lớn) giúp việc quản lí dữ liệu dễ dàng hơn. Hoặc khi người dùng thao tác xong 1 trong những chức năng ví dụ như update, xóa và tìm 1 thông tin mã gia hạn nào đó, thì khi load lại dữ liệu, thì dữ liệu sẽ trở về dữ liệu ban đầu 1 cách nhanh chóng.

### Update gia hạn bảo hành

Chức năng update gia hạn bảo hành dùng để chỉnh sửa hoặc bổ sung thông tin một khách hàng khi muốn kéo dài thêm ngày hết hạn, hoặc sửa chữa chi phí gia hạn đã tồn tại trong hệ thống. Sau khi hoàn tất nhập liệu chỉ cần sử dụng chức năng update thì mọi dữ liệu sẽ được cập nhật lại mới hết đúng với yêu cầu của khách hàng mong muốn.

### Xóa gia hạn bảo hành

Chức năng xóa gia hạn bảo hành dùng để loại bỏ thông tin của 1 phiếu gia hạn khi đã hết hạn, hoặc không còn muốn gia hạn nữa. Hệ thống sẽ xóa dữ liệu tương đương khỏi cơ sở dữ liệu khi người dùng sử dụng chức năng xóa.

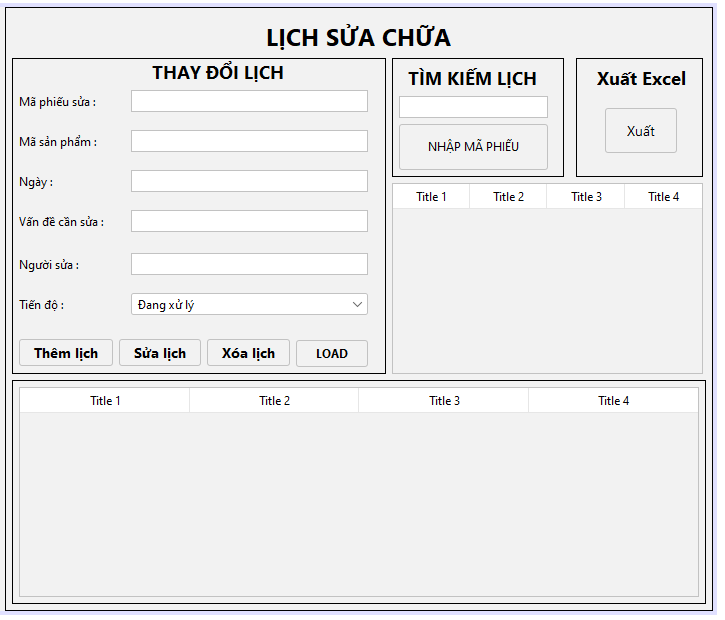
### Xuất file gia hạn bảo hành

Chức năng xuất file gia hạn bảo hành dùng để lưu trữ dữ liệu, chia sẻ thông tin, hoặc báo cáo lại với mọi người, in ấn sao kê giúp cho người dùng dễ dàng quản lý, chia sẻ và lưu trữ thông tin một cách dễ dàng và hiệu quả.

### Tìm kiếm gia hạn bảo hành

Chức năng tìm kiếm gia hạn bảo hành theo mã giúp cho người dùng tra cứu thông tin cần thiết của khách hàng khi được yêu cầu.

## Form lịch sử sửa chữa



Hình 20:Giao diện chức năng form lịch sửa chữa

### Thêm lịch sửa chữa

Thêm lịch sửa chữa giúp người dùng có thể nhập thông tin khi khách hàng muốn sửa chữa một sản phẩm đã hư. Người dùng có thể thêm (Mã phiếu sửa, Mã sản phẩm, Ngày sửa, Vấn đề cần sửa, Người phụ trách sửa chữa và Tiến độ sửa chữa). Hệ thống sẽ tạo ra 1 vùng dữ liệu để lưu lại thông tin mà người dùng vừa nhập. Sau khi nhập xong và nhấn hoàn thành. Thì dưới dữ liệu sẽ thêm mới dòng có đầy đủ thông tin mà người dùng đã nhập.

### Sửa lịch sửa chữa

Chức năng sửa lịch sửa chữa dùng để chỉnh sửa hoặc bổ sung thông tin sửa chữa ví dụ khi sửa xong, nhưng trong quá trình sửa, có sự thay thế của người sửa hoặc thiết bị bị hỏng khi đã sửa xong. Sau khi hoàn tất nhập liệu chỉ cần sử dụng chức năng update thì mọi dữ liệu sẽ được cập nhật lại mới hết đúng với yêu cầu của khách hàng mong muốn.

### Xóa lịch sửa chữa

Chức năng xóa lịch sửa chữa dùng để loại bỏ thông tin của 1 lịch sửa chữa, hoặc sản phẩm đã được sửa xong trong 1 thời gian dài. Hệ thống sẽ xóa dữ liệu tương đương khỏi cơ sở dữ liệu khi người dùng sử dụng chức năng xóa.

### Load lịch sửa chữa

Chức năng load lịch sửa chữa cho phép người dùng sắp xếp lại dữ liệu theo mã sửa chữa(từ nhỏ đến lớn) giúp việc quản lí dữ liệu dễ dàng hơn. Hoặc khi người dùng thao tác xong 1 trong những chức năng ví dụ như update, xóa và tìm 1 thông tin mã sửa chữa nào đó, thì khi load lại dữ liệu, thì dữ liệu sẽ trở về dữ liệu ban đầu 1 cách nhanh chóng.

### Xuất lịch sửa chữa

Chức năng xuất file lịch sửa chữa dùng để lưu trữ dữ liệu, chia sẻ thông tin, hoặc báo cáo lại với mọi người, in ấn sao kê giúp cho người dùng dễ dàng quản lý, chia sẻ và lưu trữ thông tin một cách dễ dàng và hiệu quả.

### Tìm kiếm lịch sửa chữa

Chức năng tìm kiếm lịch sửa chữa theo mã giúp cho người dùng tra cứu thông tin cần thiết của khách hàng khi được yêu cầu.

### Bảng tìm kiếm thông tin sửa chữa của 1 sản phẩm(Bảng nhỏ)

Chức năng của bảng này là hiển thị thông tin sản phẩm bao gồm Mã sản phẩm, Tên của sản phẩm. Khi người dùng muốn kiểm tra thông tin sửa chữa của 1 sản phẩm nào đó. Người dùng click vào dòng danh sách tương ứng được hiển thị trong bản, thì thông tin sửa chữa của sản phẩm đó sẽ được load lên bảng(bảng to) bao gồm ngày sửa chữa, thông tin sửa chữa, tiến độ sửa chữa. Giúp cho người dùng có thể tra cứu 1 cách nhanh chóng, dễ dàng, hiệu quả.

# Kết quả và đánh giá

## Kết quả đạt được

Hệ thống quản lý bảo hành sản phẩm đã được triển khai thành công và hoạt động ổn định, mang lại những kết quả đáng kể như sau:

* **Hoạt động ổn định**: Hệ thống cho phép quản lý toàn bộ các yêu cầu bảo hành từ khách hàng một cách dễ dàng và hiệu quả. Người dùng có thể tra cứu thông tin sản phẩm, theo dõi tình trạng bảo hành, cũng như cập nhật các trạng thái sửa chữa và bảo hành một cách nhanh chóng. Điều này giúp giảm thiểu sai sót và tăng cường sự minh bạch trong quy trình bảo hành.
* **Hiệu suất cao với Cassandra**: Với sự tích hợp của hệ cơ sở dữ liệu Cassandra, hệ thống có khả năng xử lý khối lượng dữ liệu lớn mà vẫn đảm bảo tốc độ truy xuất nhanh. Nhờ vào kiến trúc phân tán và khả năng mở rộng của Cassandra, hệ thống có thể đáp ứng tốt nhu cầu quản lý dữ liệu sản phẩm, khách hàng, và yêu cầu bảo hành mà không gặp vấn đề về hiệu năng ngay cả khi số lượng dữ liệu tăng lên đáng kể.

## Đánh giá

### Ưu điểm

**Dễ sử dụng**: Hệ thống được thiết kế với giao diện đơn giản, thân thiện với người dùng, giúp việc tra cứu và quản lý bảo hành sản phẩm trở nên dễ dàng. Các chức năng chính đều được bố trí hợp lý, trực quan, giúp người quản trị dễ dàng nắm bắt thông tin quan trọng mà không cần phải thao tác nhiều.

**Hiệu năng cao**: Nhờ vào sự hỗ trợ của Cassandra, hệ thống có khả năng xử lý và truy xuất dữ liệu nhanh chóng, ngay cả khi số lượng sản phẩm và yêu cầu bảo hành ngày càng tăng. Điều này đặc biệt hữu ích cho các doanh nghiệp có quy mô lớn và có nhu cầu xử lý lượng dữ liệu khổng lồ.

**Dễ mở rộng**: Hệ thống được thiết kế với khả năng mở rộng linh hoạt, cho phép dễ dàng tích hợp thêm các sản phẩm mới hoặc yêu cầu bảo hành mới mà không làm giảm hiệu suất. Việc mở rộng hệ thống không đòi hỏi sự can thiệp sâu vào kiến trúc, đảm bảo tính linh hoạt trong tương lai khi doanh nghiệp phát triển.

### Nhược điểm

* **Giao diện Java Swing**: Mặc dù hệ thống được xây dựng trên nền tảng Java Swing, giao diện này có phần lỗi thời so với các tiêu chuẩn thiết kế hiện đại. Người dùng có thể cảm thấy rằng trải nghiệm không được mượt mà và trực quan như các giao diện người dùng hiện đại (UI/UX) phổ biến hiện nay. Đặc biệt là khi so sánh với các công nghệ web hiện đại, giao diện Java Swing có phần thiếu hấp dẫn và không tối ưu cho trải nghiệm người dùng trên các thiết bị khác nhau.
* **Yêu cầu kiến thức chuyên sâu về Cassandra**: Một trong những thách thức của hệ thống là việc sử dụng Cassandra yêu cầu phải có kiến thức sâu rộng để có thể thiết kế **schema** (mô hình dữ liệu) hợp lý. Cassandra có một mô hình dữ liệu khác biệt so với các hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) truyền thống, và điều này đòi hỏi người phát triển phải nắm vững các nguyên tắc cơ bản của Cassandra để tối ưu hóa hệ thống. Việc quản lý một cơ sở dữ liệu phân tán như Cassandra cũng có thể phức tạp hơn trong các tình huống triển khai thực tế.

# Kết luận

Trong quá trình nghiên cứu và phát triển hệ thống quản lý bảo hành, chúng ta đã đạt được một giải pháp hoàn chỉnh và toàn diện. Hệ thống không chỉ đáp ứng đầy đủ các yêu cầu về quản lý thông tin bảo hành mà còn cung cấp các chức năng quan trọng như theo dõi yêu cầu bảo hành, cập nhật trạng thái xử lý, và quản lý danh sách sản phẩm. Qua việc triển khai và thực hiện, hệ thống đã chứng minh tính hiệu quả và khả năng áp dụng vào thực tế, giúp cho doanh nghiệp dễ dàng kiểm soát toàn bộ quá trình bảo hành sản phẩm một cách chính xác và kịp thời. Từ việc lưu trữ thông tin bảo hành chi tiết cho đến hỗ trợ tra cứu thông tin nhanh chóng, hệ thống đã tạo ra một nền tảng quản lý bảo hành tiện lợi và đáng tin cậy.

Hơn nữa, tầm quan trọng của việc quản lý bảo hành hiệu quả trong môi trường kinh doanh hiện nay không thể bị coi nhẹ. Trong thời đại công nghệ số, khi mà sự cạnh tranh ngày càng khốc liệt, khả năng duy trì mối quan hệ tốt đẹp với khách hàng thông qua việc cung cấp các dịch vụ hậu mãi chuyên nghiệp và nhanh chóng là yếu tố quyết định sự thành công của doanh nghiệp. Hệ thống quản lý bảo hành chính là cầu nối giúp doanh nghiệp giữ được sự tin tưởng từ phía khách hàng, khi họ có thể yên tâm về sản phẩm của mình sẽ được bảo vệ và xử lý kịp thời nếu có sự cố xảy ra. Đồng thời, việc số hóa và quản lý bảo hành hiệu quả cũng giúp doanh nghiệp tiết kiệm được nhiều chi phí vận hành, giảm thiểu sai sót trong việc xử lý thông tin và nâng cao chất lượng dịch vụ.

Tuy nhiên, như bất kỳ hệ thống nào khác, vẫn còn nhiều tiềm năng cải tiến và phát triển cho tương lai. Một trong những hướng cải tiến đáng xem xét là việc nâng cấp giao diện người dùng (UI/UX) để tăng cường trải nghiệm người dùng, giúp giao diện trở nên thân thiện và trực quan hơn. Điều này không chỉ giúp cho quá trình thao tác trở nên dễ dàng mà còn thu hút người dùng nhiều hơn, đặc biệt là những người không quen thuộc với các công nghệ mới. Ngoài ra, một hướng phát triển khác là mở rộng hệ thống để có thể kết nối với các nền tảng khác như hệ thống quản lý quan hệ khách hàng (CRM) hoặc các nền tảng thương mại điện tử. Sự kết nối này sẽ giúp doanh nghiệp có cái nhìn tổng thể về quá trình từ bán hàng đến hậu mãi, giúp cải thiện không chỉ quản lý bảo hành mà còn tối ưu hóa quy trình kinh doanh và dịch vụ khách hàng.

Tóm lại, hệ thống quản lý bảo hành mà chúng ta đã xây dựng không chỉ đáp ứng được các yêu cầu ban đầu mà còn tạo ra nền tảng cho các cơ hội phát triển và cải tiến trong tương lai. Với những đề xuất cải tiến trên, hệ thống có thể tiếp tục được nâng cao về cả tính năng lẫn khả năng tích hợp, góp phần tạo ra những bước tiến lớn trong việc quản lý bảo hành và dịch vụ khách hàng trong thời đại công nghệ số.

# Tài liệu tham khảo

[1]: <https://cassandra.apache.org/_/index.html>

[2]: <https://www.javatpoint.com/java-swing>

[3]: <https://www.w3schools.com/git/>

[4]: https://teamvietdev.com/tich-hop-bieu-do-jfreechart-trong-java-swing/