**Trường Đại Học Công Nghiệp TP.HCM**

**Khoa Công Nghệ Thông Tin**

**------**

Icon

Description automatically generated with medium confidence

|  |  |  |
| --- | --- | --- |
| **STT** | **MSSV** | **HỌ TÊN** |
| 1 | 20054321 | Nguyễn Thành Hiệp |
| 2 | 20055021 | Dương Văn Quy |
| 3 | 20054351 | Trần Văn Tú |

NHÓM 4

GVHD: Võ Văn Hải

Lớp: DHKTPM16FTT

**Đề tài**: Thiết kế kiến trúc và phần mềm cho hệ thống tái chế thiết bị điện tử

**Báo cáo tiểu luận**

**Môn: Kiến trúc và thiết kế phần mềm**

MỤC LỤC

[Chương I. Giới thiệu tổng quan 3](#_Toc167700193)

[1.Tổng quan 3](#_Toc167700194)

[2. Mục tiêu chính của đề tài: 4](#_Toc167700195)

[Chương II. Các công nghệ liên quan 5](#_Toc167700196)

[1. Ngôn ngữ phát triển ứng dụng 5](#_Toc167700197)

[2. Những framework hỗ trợ phát triển ứng dụng 5](#_Toc167700198)

[2.1 React.js 5](#_Toc167700199)

[2.2 Node.js 6](#_Toc167700200)

[2.3 MongoDB 7](#_Toc167700201)

[Chương III. Tổng quan về kiến trúc Microservices 8](#_Toc167700202)

[1. Khái niệm về kiến trúc Microservices 8](#_Toc167700203)

[2. Ưu điểm và Nhược điểm của kiến trúc Microservices 10](#_Toc167700204)

[2.1 Ưu điểm của Kiến Trúc Microservices 10](#_Toc167700205)

[2.2 Nhược điểm của Kiến Trúc Microservices 11](#_Toc167700206)

[3. So Sánh Kiến Trúc Microservices với các kiến trúc khác 12](#_Toc167700207)

[3.1 Kiến Trúc Monolithic 12](#_Toc167700208)

[3.2 Kiến trúc Service-Oriented (SOA) 13](#_Toc167700209)

[3.3 Kiến Trúc Serverless 13](#_Toc167700210)

[4. Lý do chọn kiến trúc microservices cho hệ thống tái chế thiết bị điện tử 15](#_Toc167700211)

[4.1 Đặc điểm của hệ thống tái chế 15](#_Toc167700212)

[4.2 Tại sao microservices phù hợp với hệ thống này? 15](#_Toc167700213)

[4.3 Các lợi ích cụ thể mà microservices mang lại cho hệ thống tái chế 15](#_Toc167700214)

[Chương IV. Phân tích và thiết kế phần mềm 16](#_Toc167700215)

[4.1 Usecase tổng quát 16](#_Toc167700216)

[4.1.1 Danh sách các tình huống hoạt động (Use cases) 16](#_Toc167700217)

[4.2 Tình huống hoạt động 17](#_Toc167700218)

[4.3 Giao diện đồ họa 41](#_Toc167700219)

[Chương V. Kết luận và hướng phát triển 42](#_Toc167700220)

[5.1 Kết quả đạt được 42](#_Toc167700221)

[5.2 Hạn chế 42](#_Toc167700222)

[5.3 Hướng phát triển trong tương lai 42](#_Toc167700223)

[Tài liệu tham khảo 43](#_Toc167700224)

# **Chương I. Giới thiệu tổng quan**

# **1.Tổng quan**

Trong giai đoạn phát triển hiện nay, thiết bị điện tử nhiều và hầu hết có mặt trên toàn thế giới. Bên cạnh đó việc tái chế các thiết bị điện tử sẽ trở thành vấn đề lớn đối với chúng ta.Vì thế cần tạo ra sự kết nối khách hàng với công ty để thuận tiện trao đổi thiết bị điện tử cũ nếu khách hàng thấy mức giá công ty đưa ra là hợp lý.

Dựa vào những tiêu chí trên cùng với yêu cầu đề bài đặt ra, nhóm đã chọn kiến trúc hợp lý và thiết kế phần mềm dựa trên kiến trúc cho website tái chế, giúp đễ thao tác sử dụng từ khách hàng đến công ty tái chế điện tử.

Hệ thống được sử dụng để tái chế các thiết bị điện tử cũ (chẳng hạn như điện thoại di động iPhone Hoặc Galaxy). Các quy trình xử lý tái chế các thiết bị điện tử cũ hoạt động như sau: đầu tiên, khách hàng yêu cầu công ty (thông qua một trang web hoặc kiosk) bao nhiêu tiền họ có thể nhận được cho các thiết bị điện tử cũ (gọi là trích dẫn). Nếu hài lòng, khách hàng sẽ gửi thiết bị điện tử đến tái chế công ty, mà lần lượt sẽ nhận được thiết bị (được gọi là nhận). Sau khi nhận được, công ty tái chế sau đó sẽ đánh giá thiết bị xác định xem thiết bị có hoạt động tốt hay không (được gọi là đánh giá). Nếu thiết bị đang trong tình trạng làm việc tốt, công ty gửi cho khách hàng số tiền đã hứa cho thiết bị (được gọi là kế toán). Thông qua quá trình này khách hàng có thể truy cập trang web bất cứ lúc nào kiểm tra trạng thái của mục (được gọi là trạng thái). Dựa trên đánh giá, thiết bị là sau đó tái chế bằng cách phá hủy nó một cách an toàn bán lại nó (được gọi là tái chế). Cuối cùng, công ty chạy báo cáo tài chính theo lịch trình và báo cáo hoạt động dựa trên tái chế hoạt động (được gọi là báo cáo).

# **2. Mục tiêu chính của đề tài:**

- Xây dựng và phát triển hệ thống tái chế điện tử:

Thiết kế và triển khai hệ thống hiệu quả: Tạo ra một hệ thống tái chế thiết bị điện tử hiện đại và hiệu quả, nhằm giải quyết một cách toàn diện các vấn đề liên quan đến việc tái chế thiết bị điện tử cũ. Hệ thống sẽ được xây dựng với các dịch vụ quan trọng như báo giá, nhận hàng, đánh giá, kế toán, tái chế và báo cáo, tất cả đều được tích hợp chặt chẽ và liên kết qua một cơ sở dữ liệu chung.

- Đáp ứng nhu cầu người dùng:

Tạo ra hệ thống thân thiện và tiện lợi: Phát triển một hệ thống dễ sử dụng, thuận tiện và nhanh chóng, giúp người dùng dễ dàng thực hiện các bước tái chế thiết bị điện tử. Đảm bảo rằng hệ thống cung cấp giao diện web thân thiện, cho phép người dùng truy cập và sử dụng một cách hiệu quả, mang lại trải nghiệm tốt nhất cho họ.

# **Chương II. Các công nghệ liên quan**

# **1. Ngôn ngữ phát triển ứng dụng**

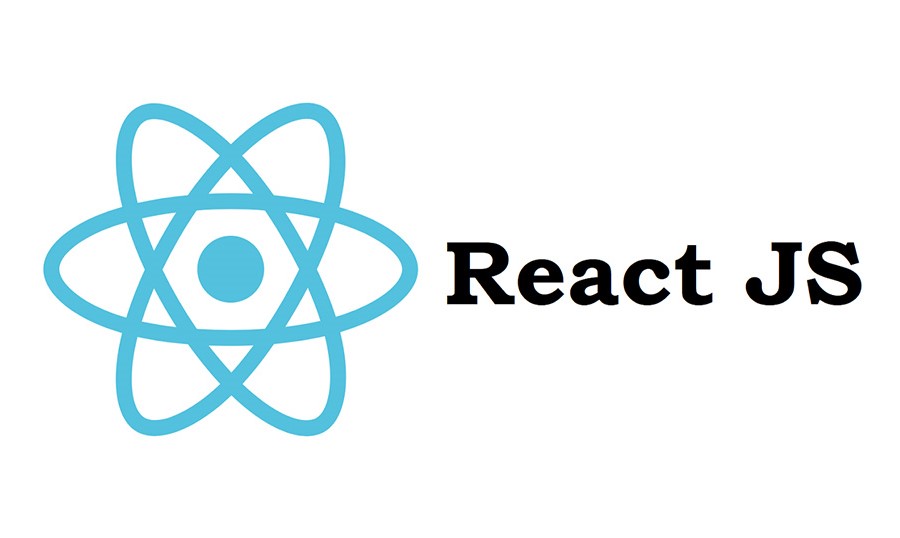
**Ngôn ngữ lập trình java**



Java là ngôn ngữ lập trình hướng đối tượng đang được duy trì và phát triển bởi Oracle. Java được sử dụng trong phát triển phần mềm ứng dụng desktop, website, game hay ứng dụng trên các thiết bị di động. Java cũng là một ngôn ngữ cho phép phát triển các ứng dụng chạy trên mọi nền tảng. Hiện nay, đội ngũ phát triển ứng ụng dùng ngôn ngữ Java ngày càng hùng hậu. Đội ngũ này cũng cung cấp một tập hợp các chương trình giúp lập trình viên phát triển và chạy các ứng dụng một cách hiệu quả. Java được tổ chức bao gồm một công cụ thực thi, một trình biên dịch và một bộ thư viện. James Gosling đã phát triển nền tảng Java tại Sun Microsystems và tập đoàn Oracle sau đó đã mua lại nó và phát triển đến ngày hôm nay

# **2. Những framework hỗ trợ phát triển ứng dụng**

## **2.1 React.js**



React JS, hay đơn giản là React, là một thư viện JavaScript mã nguồn mở được phát triển bởi Facebook để xây dựng giao diện người dùng, đặc biệt là cho các ứng dụng web đơn trang (single-page applications). React cho phép các nhà phát triển tạo ra các thành phần UI có thể tái sử dụng, dễ dàng quản lý và tối ưu hóa hiệu suất.

## **2.2 Node.js**



Node JS là một môi trường máy chủ thực thi Javascript mã nguồn mở, hoạt động trên nhiều nền tảng như Windows, MacOS, Linux...NodeJs là 1 công nghệ có cộng đồng hỗ trợ và phát triển lớn mạnh. Năm 2019 Node JS bổ sung thêm được hơn 50% số lượng lập trình viên sử dụng trên thế giới. Đến nay vẫn chiếm vị thế là công nghệ được ưa chuộng và sử dụng nhiều nhất [11]. NodeJS không cần phải cấu hình phức tạp, xây dựng dựa trên Javascript, cho nên NodeJ S được nhiều lập trình viên ưa chuộng bởi tính đơn giản và tiện dụng của nó. Node JS được sử dụng làm webserver, làm môi trường phát triển web (ReactJS), ứng dụng di động (React Native),…Webserver được phát triển bằng Node JS có hiệu năng, tốc độ xử lý cao nhưng vẫn có khả năng bảo trì nâng cấp tốt.

## **2.3 MongoDB**



MongoDB hay Mongo Database là phần mềm cơ sở dữ liệu opensource (mã nguồn mở) dạng NoSQL hỗ trợ nhiều nền tảng lập trình và được thiết kế theo kiểu hướng đối tượng. Những bảng dữ liệu trong Mongo Database (các Collection) được thiết kế với cấu trúc linh hoạt cho phép dữ liệu được lưu trữ không cần phải tuân theo định dạng cấu trúc nào.

# **Chương III. Tổng quan về kiến trúc Microservices**

# **Khái niệm về kiến trúc Microservices**

Kiến trúc Microservices là một phong cách thiết kế phần mềm trong đó một ứng dụng lớn được chia thành các dịch vụ nhỏ, độc lập với nhau. Mỗi dịch vụ (microservice) chịu trách nhiệm cho một chức năng cụ thể của ứng dụng và có thể được triển khai, nâng cấp và mở rộng một cách độc lập.

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, thiết kế

Mô tả được tạo tự động

*(Kiến trúc Microsevice dành cho hệ thống tái chế điện tử)*

*Các thành phần chính của kiến trúc Microservices:*

Các Microservices:

- Đặc điểm: Mỗi microservice thực hiện một chức năng cụ thể và độc lập. Ví dụ, trong hệ thống tái chế thiết bị điện tử, các dịch vụ có thể bao gồm Quoting, Receiving, Assessment, Accounting, và Recycling.

- Ưu điểm: Dễ dàng quản lý và bảo trì, khả năng phát triển và triển khai độc lập.

Giao tiếp giữa các Microservices:

- Công nghệ: Thường sử dụng giao tiếp qua HTTP/REST, gRPC hoặc hệ thống quản lý thông điệp như Kafka, RabbitMQ.

- Đặc điểm: Cần thiết kế giao tiếp hiệu quả để đảm bảo hiệu suất và tính nhất quán.

API Gateway:

- Đặc điểm: Cung cấp một điểm truy cập duy nhất cho tất cả các microservice. Đảm nhận việc điều phối, xác thực, và bảo mật.

- Ưu điểm: Tăng cường bảo mật, giảm tải cho các microservice, dễ dàng quản lý và giám sát.

Quản Lý Dữ Liệu:

- Đặc điểm: Mỗi microservice có thể có cơ sở dữ liệu riêng hoặc chia sẻ cơ sở dữ liệu, tùy thuộc vào yêu cầu cụ thể.

- Ưu điểm: Tăng cường tính độc lập, giảm độ phức tạp trong quản lý dữ liệu.

Triển Khai và Quản Lý:

- Công cụ: Sử dụng các công cụ như Docker, Kubernetes để quản lý container và điều phối.

- Ưu điểm: Dễ dàng triển khai, mở rộng và quản lý các microservice trong môi trường production.

Giám sát và Logging:

- Công cụ: Sử dụng Prometheus, Grafana, ELK Stack (Elasticsearch, Logstash, Kibana) để giám sát và ghi log.

- Ưu điểm: Dễ dàng theo dõi, phát hiện và xử lý sự cố.

# **2. Ưu điểm và Nhược điểm của kiến trúc Microservices**

## **2.1 Ưu điểm của Kiến Trúc Microservices**

- Tính linh hoạt cao: Các microservice có thể được phát triển, triển khai và nâng cấp độc lập.

Ưu điểm: Dễ dàng phản ứng với các thay đổi trong yêu cầu kinh doanh và công nghệ.

- Khả năng mở rộng: Có thể mở rộng từng microservice dựa trên nhu cầu thực tế mà không cần phải mở rộng toàn bộ hệ thống.

Ưu điểm: Tối ưu hóa tài nguyên và chi phí.

- Độ tin cậy cao: Một dịch vụ gặp sự cố không ảnh hưởng đến các dịch vụ khác.

Ưu điểm: Hệ thống vẫn hoạt động bình thường ngay cả khi một phần của nó bị lỗi.

- Khả năng tái sử dụng và phát triển nhanh chóng: Các dịch vụ nhỏ, dễ hiểu và dễ bảo trì.

Ưu điểm: Giảm thời gian phát triển và đưa sản phẩm ra thị trường.

## **2.2 Nhược điểm của Kiến Trúc Microservices**

- Phức tạp hơn trong quản lý: Cần quản lý nhiều dịch vụ nhỏ thay vì một hệ thống lớn.

Nhược điểm: Đòi hỏi các công cụ và kỹ năng quản lý dịch vụ phức tạp.

- Giao tiếp và tích hợp: Phải đảm bảo giao tiếp giữa các dịch vụ diễn ra mượt mà.

Nhược điểm: Có thể gặp vấn đề về độ trễ và tính nhất quán.

- Theo dõi và giám sát: Cần giám sát từng dịch vụ nhỏ.

Nhược điểm: Yêu cầu hệ thống giám sát và logging phức tạp.

- Bảo mật: Mỗi microservice cần được bảo mật độc lập.

Nhược điểm: Tạo ra nhiều điểm tấn công hơn, yêu cầu quản lý bảo mật nghiêm ngặt.

=> Kết Luận

Kiến trúc Microservices mang lại nhiều lợi ích vượt trội về tính linh hoạt, khả năng mở rộng và quản lý hiệu quả, đặc biệt phù hợp với các hệ thống phức tạp như hệ thống tái chế thiết bị điện tử. Tuy nhiên, để áp dụng thành công kiến trúc này, cần có sự chuẩn bị kỹ lưỡng về mặt công nghệ, quản lý và bảo mật nhằm đảm bảo hệ thống hoạt động trơn tru và an toàn.

# **3. So Sánh Kiến Trúc Microservices với các kiến trúc khác**

Để hiểu rõ hơn về ưu điểm của kiến trúc microservices đối với hệ thống tái chế thiết bị điện tử, chúng ta sẽ so sánh nó với ba kiến trúc khác: Monolithic, Service-Oriented Architecture (SOA), và Serverless.

## **3.1 Kiến Trúc Monolithic**

Đặc điểm:

- Tất cả các chức năng của ứng dụng được gộp lại trong một khối duy nhất.

- Các thành phần của ứng dụng chia sẻ cùng một codebase và cơ sở dữ liệu.

Ưu điểm:

- Đơn giản để phát triển ban đầu.

- Dễ dàng triển khai và quản lý trong giai đoạn đầu của dự án.

Nhược điểm:

- Khả năng mở rộng kém: Khó mở rộng từng thành phần riêng lẻ. Nếu hệ thống cần nhiều tài nguyên hơn, phải mở rộng toàn bộ ứng dụng.

- Bảo trì phức tạp: Mỗi thay đổi nhỏ đều cần xây dựng và triển khai lại toàn bộ ứng dụng.

- Tính linh hoạt thấp: Khó áp dụng các công nghệ mới cho một phần nhỏ của hệ thống mà không ảnh hưởng đến toàn bộ ứng dụng.

*So sánh với Microservices:*

- Microservices: Cho phép mở rộng và bảo trì từng dịch vụ độc lập, tăng tính linh hoạt và khả năng mở rộng.

- Monolithic: Phù hợp với các ứng dụng nhỏ hoặc ở giai đoạn đầu, nhưng gặp khó khăn khi hệ thống phát triển và yêu cầu tăng cao.

## **3.2 Kiến trúc Service-Oriented (SOA)**

Đặc điểm:

- Tương tự như microservices, SOA chia ứng dụng thành các dịch vụ.

- Các dịch vụ trong SOA thường lớn hơn và phức tạp hơn so với microservices.

- Sử dụng ESB (Enterprise Service Bus) để quản lý giao tiếp giữa các dịch vụ.

Ưu điểm:

- Khả năng tái sử dụng: Các dịch vụ trong SOA có thể được tái sử dụng cho nhiều ứng dụng khác nhau.

- Tính linh hoạt: Các dịch vụ có thể được phát triển và triển khai độc lập

Nhược điểm:

- ESB phức tạp: Quản lý và duy trì ESB có thể phức tạp và tốn kém.

- Hiệu suất: ESB có thể trở thành nút cổ chai, làm giảm hiệu suất tổng thể của hệ thống.

*So Với Microservices:*

- Microservices: Không cần ESB, sử dụng các giao tiếp nhẹ nhàng hơn như REST hoặc gRPC, dễ dàng triển khai và mở rộng.

- SOA: Tập trung vào tích hợp và tái sử dụng dịch vụ, nhưng có thể trở nên phức tạp và tốn kém.

## **3.3 Kiến Trúc Serverless**

Đặc điểm:

- Ứng dụng được xây dựng dưới dạng các function nhỏ, tự động quản lý bởi cloud provider.

- Chỉ trả phí cho thời gian thực thi của function, không cần quản lý server.

Ưu điểm:

- Chi phí hiệu quả: Chỉ trả tiền cho thời gian chạy của function.

- Tính linh hoạt cao: Dễ dàng mở rộng và triển khai các function nhỏ độc lập.

- Quản lý đơn giản: Không cần quản lý cơ sở hạ tầng.

Nhược điểm:

- Hiệu suất: Có thể gặp độ trễ khi khởi tạo function (cold start).

- Giới hạn: Không phù hợp với các ứng dụng yêu cầu xử lý liên tục hoặc có thời gian chạy dài.

- Khó khăn trong Debugging: Quá trình kiểm tra và gỡ lỗi phức tạp hơn do tính phi trạng thái của function.

*So Với Microservices:*

- Microservices: Phù hợp với các ứng dụng phức tạp, yêu cầu xử lý liên tục và cần kiểm soát chi tiết hạ tầng.

- Serverless: Tối ưu cho các tác vụ ngắn hạn, sự kiện nhỏ và không yêu cầu quản lý hạ tầng phức tạp.

=> Kết Luận

Kiến trúc Microservices mang lại nhiều ưu điểm vượt trội cho hệ thống tái chế thiết bị điện tử, bao gồm tính linh hoạt, khả năng mở rộng, triển khai và bảo trì độc lập. So với các kiến trúc khác như Monolithic, SOA, và Serverless, microservices cung cấp một giải pháp cân bằng giữa quản lý hạ tầng, hiệu suất và khả năng mở rộng, đặc biệt phù hợp với các hệ thống lớn và phức tạp như hệ thống tái chế thiết bị điện tử.

# **4. Lý do chọn kiến trúc microservices cho hệ thống tái chế thiết bị điện tử**

## **4.1 Đặc điểm của hệ thống tái chế**

* Thu thập thiết bị: Nhận và ghi nhận thông tin về các thiết bị điện tử cần tái chế từ người dùng.
* Phân loại và đánh giá: Đánh giá tình trạng và giá trị của các thiết bị, phân loại theo các tiêu chí tái chế.
* Quản lý và xử lý: Thực hiện các bước xử lý tái chế, bao gồm tháo rời, tái sử dụng các bộ phận và xử lý chất thải.
* Báo cáo và theo dõi: Cung cấp các báo cáo chi tiết về quá trình và kết quả tái chế, theo dõi tiến trình và hiệu quả của hệ thống.

## **4.2 Tại sao microservices phù hợp với hệ thống này?**

* Phân chia chức năng rõ ràng: Microservices cho phép tách biệt các chức năng của hệ thống thành các dịch vụ độc lập, giúp dễ dàng quản lý và phát triển.
* Tính mở rộng: Khi cần mở rộng hệ thống, chỉ cần nâng cấp hoặc thêm mới các dịch vụ mà không ảnh hưởng đến toàn bộ hệ thống.
* Tính linh hoạt: Cho phép mỗi dịch vụ sử dụng các công nghệ và ngôn ngữ lập trình khác nhau, tối ưu hóa hiệu suất cho từng chức năng cụ thể.
* Tính bền vững: Hệ thống có khả năng tự phục hồi khi một dịch vụ gặp sự cố, đảm bảo các phần còn lại vẫn hoạt động bình thường.

## **4.3 Các lợi ích cụ thể mà microservices mang lại cho hệ thống tái chế**

* **Scalability (Khả năng mở rộng):** Dễ dàng mở rộng từng phần của hệ thống khi nhu cầu tăng lên mà không cần phải tái cấu trúc toàn bộ hệ thống.
* **Flexibility (Tính linh hoạt):** Mỗi dịch vụ có thể được phát triển, triển khai và bảo trì một cách độc lập, giảm thiểu thời gian và chi phí phát triển.
* **Resilience (Tính bền vững):** Hệ thống sẽ ít bị ảnh hưởng nếu một dịch vụ gặp sự cố, vì các dịch vụ khác vẫn có thể hoạt động bình thường.
* **Technological Diversity (Đa dạng công nghệ):** Cho phép sử dụng các công nghệ khác nhau cho từng dịch vụ, tận dụng tối đa các lợi thế của mỗi công nghệ.
* **Improved Deployment and Updates (Cải thiện triển khai và cập nhật):** Các dịch vụ có thể được triển khai và cập nhật độc lập, giảm thiểu thời gian gián đoạn và rủi ro khi nâng cấp hệ thống.

# **Chương IV. Phân tích và thiết kế phần mềm**

### **Ảnh có chứa biểu đồ, văn bản, hàng Mô tả được tạo tự động4.1 Usecase tổng quát**

### **4.1.1 Danh sách các tình huống hoạt động (Use cases)**

|  |  |
| --- | --- |
| **ID use case** | **Tên Use case** |
| **UC01** | Đăng nhập |
| **UC02** | Đăng ký |
| **UC03** | Quên mật khẩu |
| **UC04** | Cập Nhập Thông Tin Người Dùng |
| **UC05** | Gửi thông tin thiết bị tái chế |
| **UC06** | Tiếp nhận yêu cầu tái chế |
| **UC07** | Phân loại thiết bị và báo giá |
| **UC08** | Danh sách đơn hàng |
| **UC09** | Xem chi tiết đơn hàng |
| **UC10** | Theo dõi trạng thái thiết bị |
| **UC11** | Xem danh sách nhân viên |
| **UC12** | Xem chi tiết nhân viên |
| **UC13** | Thêm nhân viên |
| **UC14** | Xóa nhân viên |
| **UC15** | Sửa thông tin nhân viên |
| **UC16** | Xem danh sách khách hàng |
| **UC17** | Kiểm tra tình trạng thiết bị |
| **UC18** | Báo giá thu lại |
| **UC19** | Đánh giá thiết bị |

Bảng 3‑2 Danh sách các tình huống trong hệ thống

### **Tình huống hoạt động**

#### 4.2.1 UC01\_Đăng nhập

|  |  |
| --- | --- |
| * **Tên use case**: UC01\_Đăng nhập | |
| * **Mô tả sơ lược**: Người dùng muốn đăng nhập vào hệ thống để sử dụng các dịch vụ | |
| * **Actor chính**: Người dùng | |
| * **Actor phụ**: | |
| * **Tiền điều kiện (Pre-condition):** Người dùng đã có tài khoản | |
| * **Hậu điều kiện (Post-condition):** | |
| * **Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Người dùng chọn chức năng đăng nhập |  |
|  | 2. Hệ thống hiển thị giao diện nhập thông tin tài khoản |
| 1. Người dùng nhập tài khoản gồm email và mật khẩu |  |
| 1. Chọn nút đăng nhập |  |
|  | 1. Hệ thống kiểm tra thông tin nhập vào |
|  | 1. Hệ thống chuyển hướng người dùng vào giao diện trang chủ |
|  |  |
| * **Luồng sự kiện thay thế (alternate flow):** | |
|  | 6.1. Hệ thống thông báo thông tin tài khoản đăng nhập không đúng |
|  | 6.2. Hệ thống quay lại bước 3 |

#### Sơ đồ Activity:

Ảnh có chứa ảnh chụp màn hình, vòng tròn, thiết kế

Mô tả được tạo tự động

#### 4.2.2 UC02\_Đăng ký

|  |  |
| --- | --- |
| * **Tên use case**: UC02\_Đăng ký | |
| * **Mô tả sơ lược**: Người dùng muốn đăng ký vào hệ thống để sử dụng các dịch vụ | |
| * **Actor chính**: Người dùng | |
| * **Actor phụ**: Không. | |
| * **Tiền điều kiện (Pre-condition):** Người dùng chưa có tài khoản | |
| * **Hậu điều kiện (Post-condition):** Người dùng tạo tài khoản thành công | |
| * **Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Người dùng chọn chức năng đăng ký trên trang chủ |  |
|  | 1. Hiển thị giao diện đăng ký |
| 1. Người dùng nhập thông tin gồm tên, email, mật khẩu |  |
| 1. Người dùng nhấn đăng ký |  |
|  | 1. Hệ thống kiểm tra có tồn tại tài khoản chưa |
|  | 1. Thông báo đang ký thành công và chuyển hướng tới giao diện trang chủ |
| * **Luồng sự kiện thay thế (alternate flow):** | |
|  | 5.1. Hệ thống thông báo đã tồn tại |
|  | 5.2 Hệ thống quay lại bước 3 |

#### Sơ đồ Activity

Ảnh có chứa ảnh chụp màn hình, biểu đồ, vòng tròn, thiết kế

Mô tả được tạo tự động

#### 4.2.3 UC03\_Quên mật khẩu

|  |  |
| --- | --- |
| * **Tên use case**: UC03\_Quên mật khẩu | |
| * **Mô tả sơ lược**: Người dùng lấy lại mật khẩu đã mất | |
| * **Actor chính**: Người dùng | |
| * **Actor phụ**: Không. | |
| * **Tiền điều kiện (Pre-condition):** Người dùng đã có tài khoản | |
| * **Hậu điều kiện (Post-condition):** Người dùng lấy lại được mật khẩu mới | |
| * **Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Người dùng chọn chức năng quên mật khẩu. |  |
|  | 1. Hiển thị giao diện lấy lại mật khẩu |
| 1. Người dùng nhập email và bấm xác nhận |  |
|  | 1. Hệ thống kiểm tra tài khoản |
|  | 1. Hệ thống hiển thị giao diện nhập mật khẩu |
|  | 1. Mật khẩu mới được gửi về gmail |
| 1. Người dùng nhập mật khẩu mới từ gmail |  |
|  | 1. Hệ thống kiểm tra mật khẩu |
|  | 1. Hệ thống xác nhận trang đăng nhập thành công. |
| * **Luồng sự kiện thay thế (alternate flow):** | |
|  | 4.1. Hệ thống thông báo không tồn tại |
|  | 4.2. Hệ thống quay lại bước 3 |
|  | * 1. Hệ thống thông báo mật khẩu không hợp lệ |
|  | 7.2 Hệ thống quay lại bước 2 |

#### Sơ đồ Activity

Ảnh có chứa ảnh chụp màn hình, biểu đồ, văn bản, hàng

Mô tả được tạo tự động

#### UC04\_Cập nhật thông tin người dùng

|  |  |
| --- | --- |
| * **Tên use case**: UC04\_Cập Nhập Thông Tin Người Dùng | |
| * **Mô tả sơ lược**: Giúp người cập nhập thông tin của người dùng trong hệ thông | |
| * **Actor chính**: người dùng | |
| * **Actor phụ**: Không. | |
| * **Tiền điều kiện (Pre-condition):** Người dùng đăng nhập thành công vào hệ thống | |
| * **Hậu điều kiện (Post-condition):** Thông tin người dùng được cập nhập | |
| * **Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Người dùng chọn chức năng cập nhật thông tin |  |
|  | 1. Hệ thống hiển thị giao diện quản lí thông tin người dùng |
| 1. Người dùng nhập thông tin tài khoản cần thay đổi |  |
| 1. Người dùng nhấn cập nhật |  |
|  | 1. Hệ thống kiểm tra thông tin nhập vào |
|  | 1. Thông báo cập nhập thành công |
| * **Luồng sự kiện thay thế (alternate flow):** | |
|  | 5.1. Hệ thống thông báo thông tin không hợp lệ |
|  | 5.2. Hệ thống quay lại bước 3 |

#### Sơ đồ Activity

Ảnh có chứa ảnh chụp màn hình, biểu đồ, hàng, vòng tròn

Mô tả được tạo tự động

#### 4.2.5 UC05\_Gửi thông tin thiết bị tái chế

|  |  |
| --- | --- |
| * **Tên use case**: UC05\_Gửi thông tin thiết bị tái chế | |
| * **Mô tả sơ lược**: Giúp người dùng gửi thông tin, tình trạng của thiết bị cần tái chế | |
| * **Actor chính**: Người dùng | |
| * **Actor phụ**: Không. | |
| * **Tiền điều kiện (Pre-condition):** Người dùng đăng nhập thành công | |
| * **Hậu điều kiện (Post-condition):** Thông tin, tình trạng thiết bị đã được gửi đi | |
| * **Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Người dùng chọn chức năng gửi thông tin thiết bị tái chế |  |
|  | 2. Hệ thống hiển thị giao diện gửi thông tin thiết bị tái chế |
| 1. Người dùng nhập các thông tin liên quan về thiết bị tái chế |  |
| 1. Người dùng nhấn gửi thông tin |  |
|  | 1. Hệ thống kiểm tra các thông tin nhập |
|  | 1. Thông báo gửi thông tin thành công |
| * **Luồng sự kiện thay thế (alternate flow):** | |
|  | 5.1. Hệ thống thông báo nhập thiếu thông tin hoặc không hợp lệ |
|  | 5.2. Hệ thống quay lại bước 3 |

#### Sơ đồ Activity

Ảnh có chứa ảnh chụp màn hình, biểu đồ, hàng, thiết kế

Mô tả được tạo tự động

#### UC06\_Tiếp nhận yêu cầu tái chế

|  |  |
| --- | --- |
| * **Tên use case**: UC06\_ Tiếp nhận yêu cầu tái chế | |
| * **Mô tả sơ lược**: Giúp nhân viên tiếp nhận thông tin thiết bị tái chế từ khách hàng | |
| * **Actor chính**: Nhân viên | |
| * **Actor phụ**: Không. | |
| * **Tiền điều kiện (Pre-condition):** Đăng nhập thành công | |
| * **Hậu điều kiện (Post-condition):** Hệ thống thông báo đã tiếp nhận thông tin | |
| * **Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Nhân viên chọn chức năng tiếp nhận thông tin |  |
|  | 1. Hệ thống hiển thị giao diện các danh sách tiếp nhận từ người dùng |
| 1. Nhân viên nhấn đã tiếp nhận thông tin |  |
|  | 1. Hệ thống thông báo đã tiếp nhận thông tin cho khách hàng |

#### 4.2.6.1 Sơ đồ Activity

Ảnh có chứa ảnh chụp màn hình, biểu đồ, hàng, vòng tròn

Mô tả được tạo tự động

#### 4.2.7 UC07\_Phân loại thiết bị và báo giá

|  |  |
| --- | --- |
| * **Tên use case**: UC07\_ Phân loại thiết bị và báo giá | |
| * **Mô tả sơ lược**: Giúp người dùng biết giá thiết bị tái chế | |
| * **Actor chính**: Nhân viên | |
| * **Actor phụ**: Không. | |
| * **Tiền điều kiện (Pre-condition):** Đăng nhập thành công | |
| * **Hậu điều kiện (Post-condition):** Hệ thống thông báo giá thiết bị cho khách hàng | |
| * **Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Nhân viên chọn chức năng xem thông tin thiết bị và phân loại |  |
|  | 1. Hệ thống hiển thị giao diện thông tin thiết bị |
| 1. Nhân viên xem thông tin thiết bị, phân loại sản phẩm |  |
| 1. Nhân viên nhập giá và gửi đi |  |
|  | 1. Hệ thống tiếp nhận giá, và thông báo giá đến khách hàng |

#### 4..2.7.1 Sơ đồ Activity

Ảnh có chứa ảnh chụp màn hình, văn bản, biểu đồ, hàng

Mô tả được tạo tự động

#### UC08\_Danh sách đơn hàng

|  |  |
| --- | --- |
| * **Tên use case**: UC08\_ Danh sách đơn hàng | |
| * **Mô tả sơ lược**: Giúp nhân viên xem danh sách các đơn hàng | |
| * **Actor chính**: Nhân viên | |
| * **Actor phụ**: Không. | |
| * **Tiền điều kiện (Pre-condition):** Nhân viên đăng nhập thành công | |
| * **Hậu điều kiện (Post-condition):** Hệ thống hiển thị danh sách đơn hàng | |
| * **Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Nhân viên chọn chức năng danh sách đơn hàng |  |
|  | 2. Hệ thống hiển thị giao diện danh sách đơn hàng |
| 1. Nhân viên xem các đơn hàng |  |
|  | 1. Hệ thống hiển thị danh sách đơn hàng |

* + - 1. Sơ đồ Activity

Ảnh có chứa ảnh chụp màn hình, văn bản, biểu đồ, hàng

Mô tả được tạo tự động

#### 4.2.9 UC09\_Xem chi tiết đơn hàng

|  |  |
| --- | --- |
| * **Tên use case**: UC09\_Xem chi tiết đơn hàng | |
| * **Mô tả sơ lược**: Giúp nhân viên xem chi tiết các đơn hàng | |
| * **Actor chính**: Nhân viên | |
| * **Actor phụ**: Không. | |
| * **Tiền điều kiện (Pre-condition):** Nhân viên đăng nhập thành công | |
| * **Hậu điều kiện (Post-condition):** Hệ thống hiển thị chi tiết thông tin đơn hàng | |
| * **Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Nhân viên chọn chức năng xem chi tiết đơn hàng |  |
|  | 2. Hệ thống hiển thị giao diện thông tin chi tiết của từng đơn hàng |
| 1. Nhân viên xem các thông tin |  |
|  | 1. Hệ thống hiển thị chi tiết các thông tin đơn hàng |

4.2.9.1 Sơ đồ Activity

Ảnh có chứa ảnh chụp màn hình, biểu đồ, hàng, vòng tròn

Mô tả được tạo tự động

#### 4.2.10 UC10\_ Theo dõi trạng thái thiết bị

|  |  |
| --- | --- |
| * **Tên use case**: UC10\_Theo dõi trạng thái thiết bị | |
| * **Mô tả sơ lược**: Giúp người dùng xem được tình trạng thiết bị | |
| * **Actor chính**: Người dùng | |
| * **Actor phụ**: Không. | |
| * **Tiền điều kiện (Pre-condition):** Người dùng đăng nhập thành công | |
| * **Hậu điều kiện (Post-condition):** Hệ thống hiển thị tiến trình theo dõi | |
| * **Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Người dùng chọn chức năng theo dõi thiết bị tái chế |  |
|  | 2. Hệ thống hiển thị giao diện tình trạng của thiết bị được làm đến đâu |
| 1. Người dùng chọn xem chi tiết đơn hàng |  |
|  | 1. Hệ thống hiển thị chi tiết các thông tin đơn hàng |

4.2.10.1 Sơ đồ Activity

Ảnh có chứa ảnh chụp màn hình, biểu đồ, hàng, vòng tròn

Mô tả được tạo tự động

#### 4.2.11 UC11\_ Xem danh sách nhân viên

|  |  |
| --- | --- |
| * **Tên use case**: UC11\_ Xem danh sách nhân viên | |
| * **Mô tả sơ lược**: Giúp quản lý xem danh sách nhân viên | |
| * **Actor chính**: Quản lý | |
| * **Actor phụ**: Không. | |
| * **Tiền điều kiện (Pre-condition):** Quản lý đăng nhập thành công | |
| * **Hậu điều kiện (Post-condition):** Hệ thống hiển thị danh sách nhân viên | |
| * **Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Quản lý chọn chức năng xem danh sách nhân viên |  |
|  | 2. Hệ thống hiển thị giao diện danh sách nhân viên |

4.2.11.1 Sơ đồ Activity

Ảnh có chứa ảnh chụp màn hình, biểu đồ, hàng, Sơ đồ

Mô tả được tạo tự động

#### 4.2.12 UC12\_ Xem chi tiết nhân viên

|  |  |
| --- | --- |
| * **Tên use case**: UC12\_Xem chi tiết nhân viên | |
| * **Mô tả sơ lược**: Giúp quản lý xem chi tiết thông tin nhân viên | |
| * **Actor chính**: Quản lý | |
| * **Actor phụ**: Không. | |
| * **Tiền điều kiện (Pre-condition):** Quản lý đăng nhập thành công | |
| * **Hậu điều kiện (Post-condition):** Hệ thống hiển thị chi tiết thông tin nhân viên | |
| * **Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Quản lý chọn chức năng xem danh sách nhân viên |  |
|  | 2. Hệ thống hiển thị giao diện danh sách nhân viên |
| 3. Quản lý chọn vào nhân viên cần xem thông tin |  |
|  | 4. Hệ thống hiển thị thông tin nhân viên |

4.2.12.1 Sơ đồ Activity

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, hàng

Mô tả được tạo tự động

#### 4.2.13 UC13\_ Thêm nhân viên

|  |  |
| --- | --- |
| * **Tên use case**: UC13\_Thêm nhân viên | |
| * **Mô tả sơ lược**: Giúp quản lý thêm nhân viên mới | |
| * **Actor chính**: Quản lý | |
| * **Actor phụ**: Không. | |
| * **Tiền điều kiện (Pre-condition):** Quản lý đăng nhập thành công | |
| * **Hậu điều kiện (Post-condition):** Hệ thống thông báo thêm nhân viên thành công | |
| * **Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Quản lý chọn chức năng danh sách nhân viên |  |
|  | 2. Hệ thống hiển thị giao diện danh sách nhân viên |
| 1. Quản lý chọn thêm nhân viên |  |
|  | 1. Hệ thống hiển thị giao diện nhập thông tin nhân viên |
| 1. Quản lý nhập các thông tin nhân viên, cấp tài khoản và mật khẩu cho nhân viên |  |
|  | 1. Hệ thống lưu thông tin, và thông báo thêm thành công |
| * **Luồng sự kiện thay thế (alternate flow):** | |
| 5.1 Quản lý nhập thiếu thông tin nhân viên |  |
|  | 5.2. Hệ thống thông báo nhập đẩy đủ thông tin |

4.2.13.1 Sơ đồ Activity

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, hàng

Mô tả được tạo tự động

#### 4.2.14 UC14\_ Xóa nhân viên

|  |  |
| --- | --- |
| * **Tên use case**: UC14\_Xóa nhân viên | |
| * **Mô tả sơ lược**: Giúp quản lý xóa nhân viên | |
| * **Actor chính**: Quản lý | |
| * **Actor phụ**: Không. | |
| * **Tiền điều kiện (Pre-condition):** Quản lý đăng nhập thành công | |
| * **Hậu điều kiện (Post-condition):** Hệ thống thông báo xóa nhân viên thành công | |
| * **Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Quản lý chọn chức năng danh sách nhân viên |  |
|  | 2. Hệ thống hiển thị giao diện danh sách nhân viên |
| 1. Quản lý chọn nhân viên muốn xóa |  |
|  | 1. Hệ thống hiển thị giao diện thông tin nhân viên |
| 1. Quản lý chọn xóa nhân viên |  |
|  | 1. Hệ thống gửi xác nhận “Bạn mốn xóa nhân viên này!” |
| 1. Quản lý chọn “Đồng ý” |  |
|  | 1. Hệ thống xóa thông tin nhân viên và thông báo “Xóa thành công” |
| * **Luồng sự kiện thay thế (alternate flow):** | |
| 7.1 Quản lý chọn “Hủy” |  |
|  | 7.2. Hệ thống quay lại giao diện danh sách nhân viên |

4.2.14.1 Sơ đồ Activity

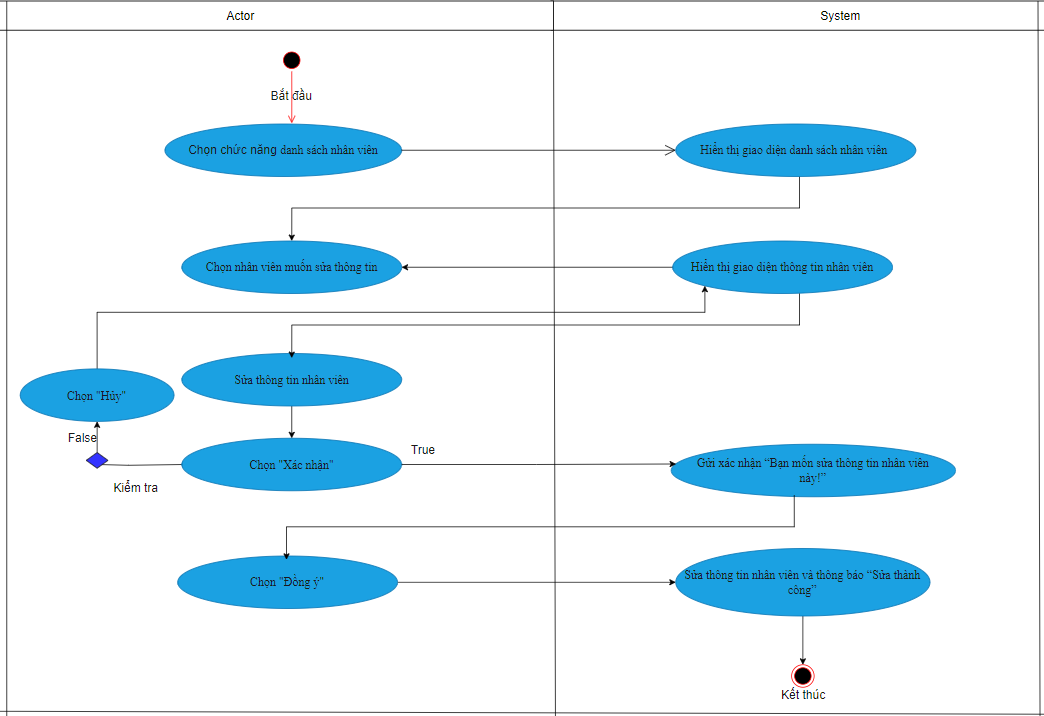
Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, hàng

Mô tả được tạo tự động

#### 4.2.15 UC15\_ Sửa thông tin nhân viên

|  |  |
| --- | --- |
| * **Tên use case**: UC15\_ Sửa thông tin nhân viên | |
| * **Mô tả sơ lược**: Giúp quản lý sửa thông tin nhân viên | |
| * **Actor chính**: Quản lý | |
| * **Actor phụ**: Không. | |
| * **Tiền điều kiện (Pre-condition):** Quản lý đăng nhập thành công | |
| * **Hậu điều kiện (Post-condition):** Hệ thống thông báo sửa thông tin nhân viên thành công | |
| * **Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Quản lý chọn chức năng danh sách nhân viên |  |
|  | 2. Hệ thống hiển thị giao diện danh sách nhân viên |
| 1. Quản lý chọn nhân viên muốn sửa thông tin |  |
|  | 1. Hệ thống hiển thị giao diện thông tin nhân viên |
| 1. Quản lý sửa thông tin nhân viên |  |
| 1. Quản lý chọn “Xác nhận” |  |
|  | 1. Hệ thống gửi xác nhận “Bạn mốn sửa thông tin nhân viên này!” |
| 1. Quản lý chọn “Đồng ý” |  |
|  | 1. Hệ thống sửa thông tin nhân viên và thông báo “Sửa thành công” |
| * **Luồng sự kiện thay thế (alternate flow):** | |
| 6.1 Quản lý chọn “Hủy” |  |
|  | 4.1. Hệ thống quay lại giao diện thông tin nhân viên |

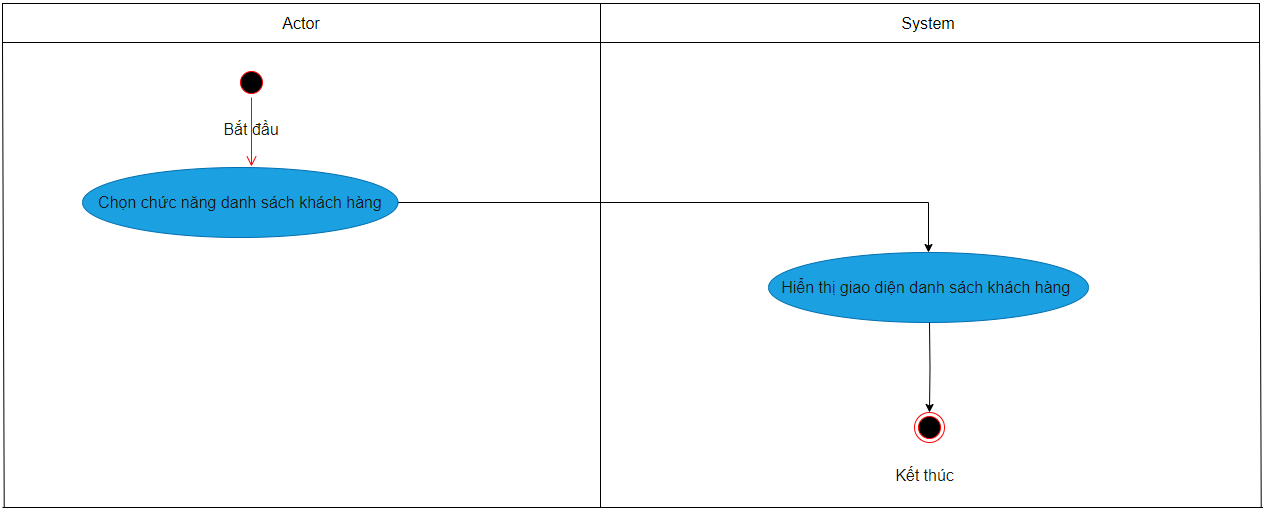
4.2.15.1 Sơ đồ Activity



#### 4.2.16 UC16\_Xem danh sách khách hàng

|  |  |
| --- | --- |
| * **Tên use case**: UC16\_ Xem danh sách khách hàng | |
| * **Mô tả sơ lược**: Giúp quản lý xem danh sách khách hàng | |
| * **Actor chính**: Quản lý | |
| * **Actor phụ**: Không. | |
| * **Tiền điều kiện (Pre-condition):** Quản lý đăng nhập thành công | |
| * **Hậu điều kiện (Post-condition):** Hệ thống hiển thị danh sách khách hàng | |
| * **Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Quản lý chọn chức năng xem danh sách khách hàng |  |
|  | 2. Hệ thống hiển thị giao diện danh sách khách hàng |

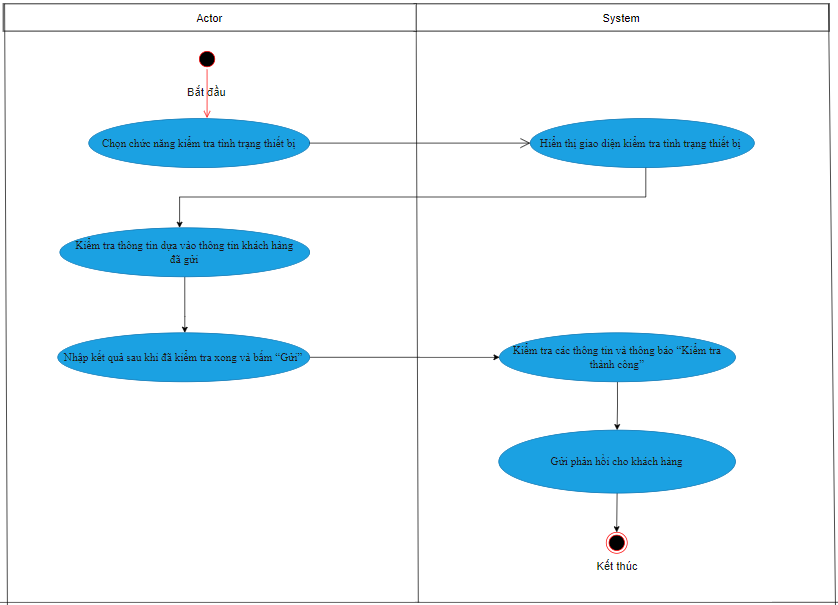
4.2.16.1 Sơ đồ Activity



#### 4.2.17 UC17\_Kiểm tra tình trạng thiết bị

|  |  |
| --- | --- |
| * **Tên use case**: UC17\_ Kiểm tra tình trạng thiết bị | |
| * **Mô tả sơ lược**: Giúp nhân viên kiểm tra tình trạng thiết bị | |
| * **Actor chính**: Nhân viên | |
| * **Actor phụ**: Không. | |
| * **Tiền điều kiện (Pre-condition):** Nhân viên đăng nhập thành công | |
| * **Hậu điều kiện (Post-condition):** Hệ thống thông báo kiểm tra thành công | |
| * **Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Nhân viên chọn chức năng kiểm tra tình trạng thiết bị |  |
|  | 2. Hệ thống hiển thị giao diện kiểm tra tình trạng thiết bị |
| 1. Nhân viên kiểm tra thông tin dựa vào thông tin khách hàng đã gửi |  |
| 1. Nhân viên nhập kết quả sau khi đã kiểm tra xong và bấm “Gửi” |  |
|  | 1. Hệ thống kiểm tra các thông tin và thông báo “Kiểm tra thành công” |
|  | 1. Hệ thống gửi phản hồi cho khách hàng |
| * **Luồng sự kiện thay thế (alternate flow):** | |
| 4.1 Nhân viên bấm “Hủy” |  |
|  | 4.2. Hệ thống quay lại bước 3 |

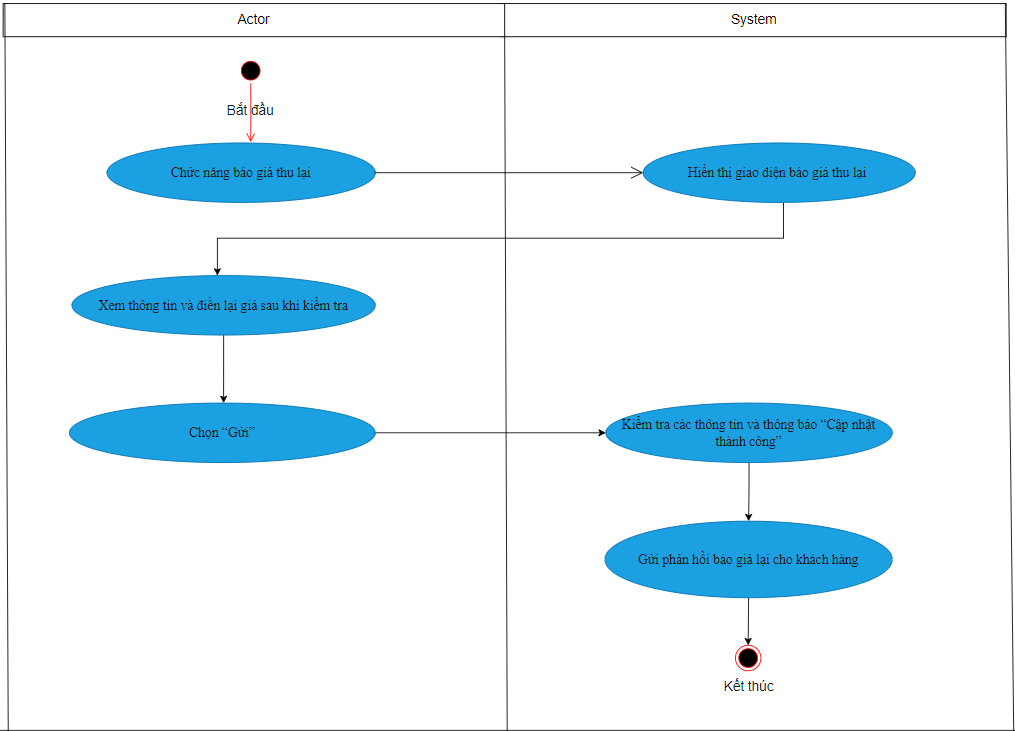
4.2.17.1 Sơ đồ Activity



#### 4.2.18 UC18\_Báo giá thu lại

|  |  |
| --- | --- |
| * **Tên use case**: UC18\_Báo giá thu lại | |
| * **Mô tả sơ lược**: Giúp nhân viên cập nhật lại giá lên hệ thống | |
| * **Actor chính**: Nhân viên | |
| * **Actor phụ**: Không. | |
| * **Tiền điều kiện (Pre-condition):** Nhân viên đăng nhập thành công | |
| * **Hậu điều kiện (Post-condition):** Hệ thống thông báo cập nhật thành công | |
| * **Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Nhân viên chọn chức năng báo giá thu lại |  |
|  | 2. Hệ thống hiển thị giao diện báo giá thu lại |
| 1. Nhân viên xem thông tin và điền lại giá sau khi kiểm tra |  |
| 1. Nhân viên chọn “Gửi” |  |
|  | 1. Hệ thống kiểm tra các thông tin và thông báo “Cập nhật thành công” |
|  | 1. Hệ thống gửi phản hồi báo giá lại cho khách hàng |
| * **Luồng sự kiện thay thế (alternate flow):** | |
| 4.1 Nhân viên bấm “Hủy” |  |
|  | 4.2. Hệ thống quay lại bước 3 |

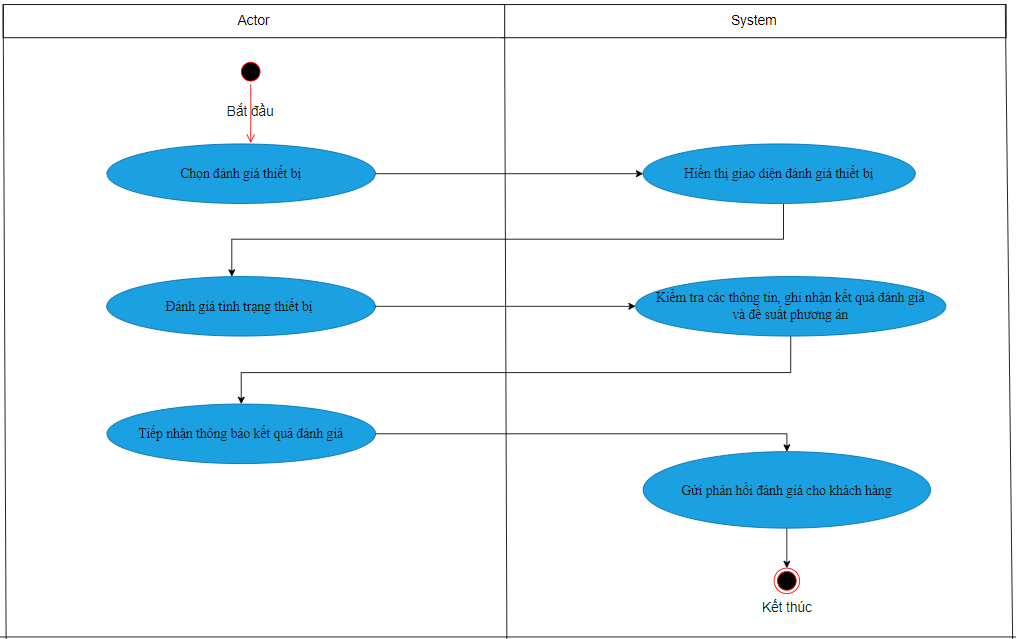
4.2.18.1 Sơ đồ Activity



#### 4.2.19 UC19\_Đánh giá thiết bị

|  |  |
| --- | --- |
| * **Tên use case**: UC19\_ Đánh giá thiết bị | |
| * **Mô tả sơ lược**: Giúp nhân viên đánh giá thiết bị | |
| * **Actor chính**: Nhân viên | |
| * **Actor phụ**: Không. | |
| * **Tiền điều kiện (Pre-condition):** Nhân viên đăng nhập thành công | |
| * **Hậu điều kiện (Post-condition):** Hệ thống thông báo đánh giá thành công | |
| * **Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Nhân viên chọn chức đánh giá thiết bị |  |
|  | 2. Hệ thống hiển thị giao diện đánh giá thiết bị |
| 1. Nhân viên đánh giá tình trạng thiết bị |  |
|  | 1. Hệ thống kiểm tra các thông tin, ghi nhận kết quả đánh giá và đề suất phương án |
| 1. Nhân viên tiếp nhận thông báo kết quả đánh giá |  |
|  | 1. Hệ thống gửi phản hồi đánh giá cho khách hàng |

4.2.19.1 Sơ đồ Activity



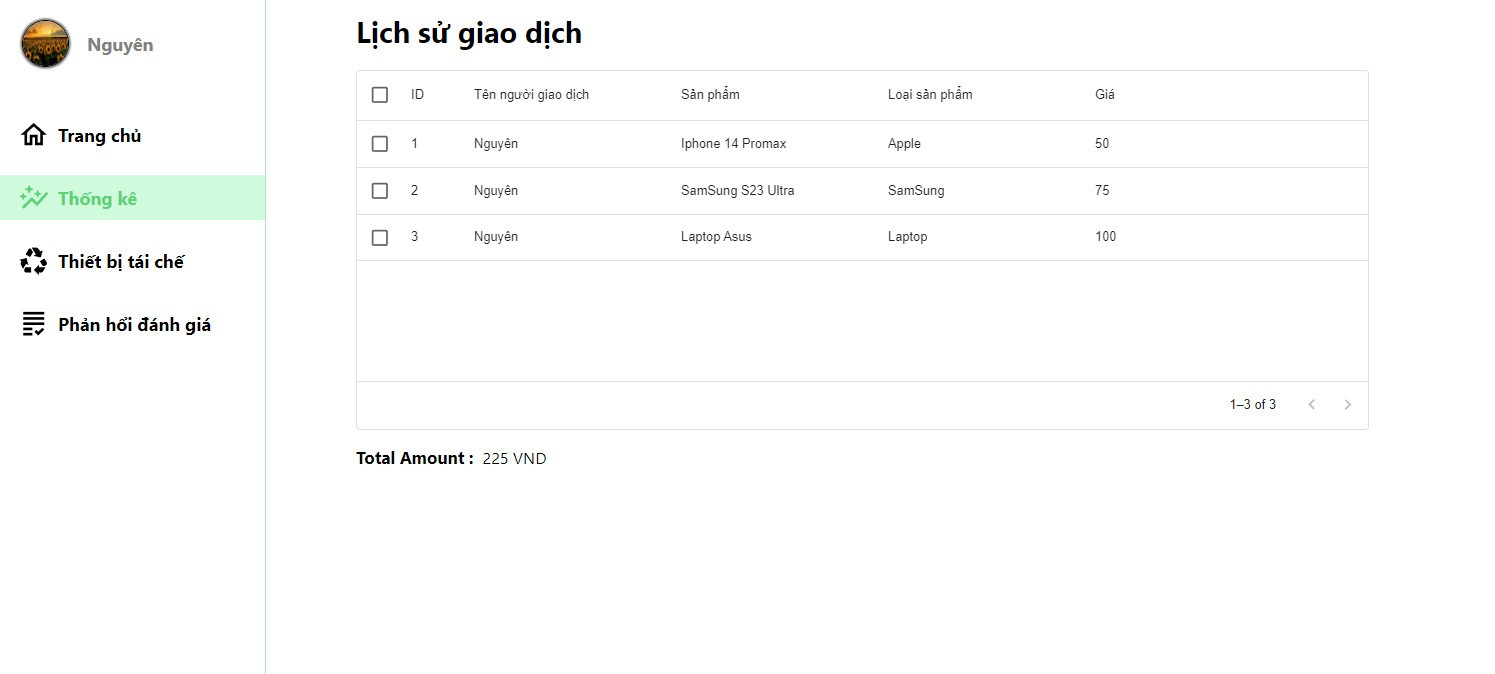
### **4.3 Giao diện đồ họa**

Giao diện Trang chủ



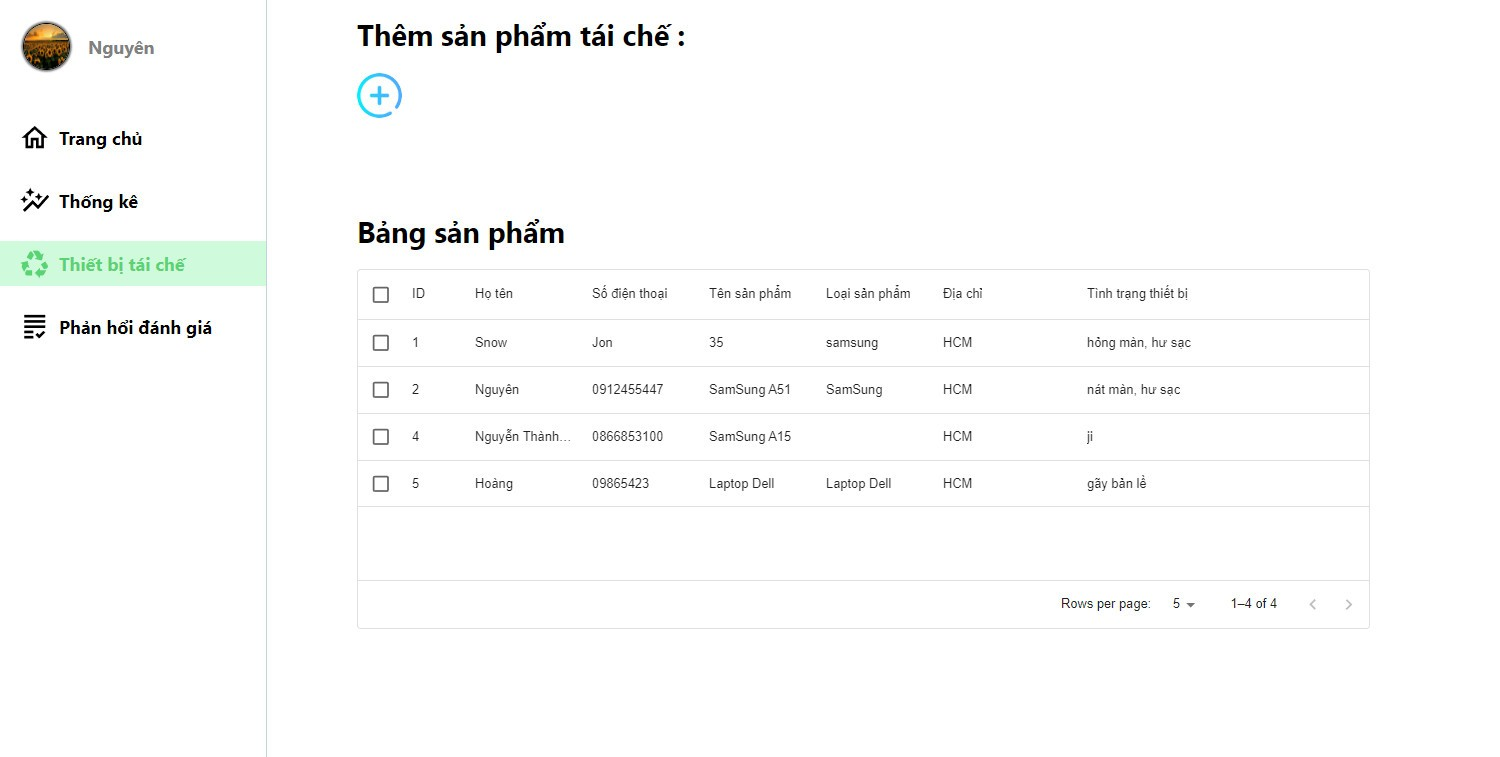
Hình 4.3.1: Giao diện Trang chủ

Giao diện Thống kê



Hình 4.3.2: Giao diện Thống kê

Giao diện Thiết bị tái chế

****

Hình 4.4.3: Giao diện Thiết bị tái chế

# **Chương V. Kết luận và hướng phát triển**

## **5.1 Kết quả đạt được**

- Hiệu quả cao: Hệ thống tái chế thiết bị điện tử dựa trên kiến trúc microservices đã hoạt động hiệu quả, đảm bảo xử lý nhanh chóng và chính xác các quy trình tái chế.

- Tính linh hoạt và mở rộng: Các microservices độc lập giúp hệ thống dễ dàng mở rộng và nâng cấp, tăng cường khả năng đáp ứng nhu cầu sử dụng.

- Trải nghiệm người dùng: Hệ thống cung cấp giao diện thân thiện, tiện lợi, giúp người dùng dễ dàng thực hiện các thao tác.

## **5.2 Hạn chế**

- Độ phức tạp: Quản lý và điều phối nhiều dịch vụ độc lập đòi hỏi kỹ thuật và công cụ phức tạp, làm tăng chi phí phát triển và vận hành.

- Giao tiếp giữa các dịch vụ: Sử dụng nhiều giao thức giao tiếp có thể gây ra độ trễ và yêu cầu cơ sở hạ tầng mạnh mẽ để đảm bảo hiệu suất.

- Quản lý dữ liệu: Đảm bảo tính nhất quán và toàn vẹn dữ liệu giữa các microservices là một thách thức lớn.

## **5.3 Hướng phát triển trong tương lai**

- Tối ưu hóa hiệu suất: Tiếp tục tối ưu hóa các giao thức giao tiếp và quản lý dịch vụ để giảm thiểu độ trễ và tăng cường hiệu suất hệ thống.

- Tích hợp AI: Sử dụng trí tuệ nhân tạo để tự động phân loại và đánh giá thiết bị điện tử, cải thiện độ chính xác và hiệu quả.

- Mở rộng chức năng: Phát triển thêm các tính năng mới như hệ thống theo dõi và dự báo, cung cấp báo cáo chi tiết hơn về quá trình tái chế và tác động môi trường.

- Cải thiện bảo mật: Nâng cao các biện pháp bảo mật để bảo vệ dữ liệu người dùng và đảm bảo an toàn cho toàn bộ hệ thống.

# Tài liệu tham khảo

[1] <https://vsi-international.com/microservice-api-ung-dung-trong-thuc-te/#:~:text=Hiểu%20một%20cách%20đơn%20giản,và%20kiểm%20tra%20riêng%20biệt.>

[2] <https://topdev.vn/blog/phat-trien-phan-mem-theo-kien-truc-microservice/>

[3] <https://appmaster.io/vi/blog/cach-chon-kien-truc-phan-mem>

[4] <https://github.com/cleophasmashiri/Patient-Management-UI-Microsevices>

[5] https://spring.io/blog/2015/07/14/microservices-with-spring

[6] https://github.com/paulc4/microservices-demo