**Trường Đại Học Công Nghiệp TP.HCM**

**Khoa Công Nghệ Thông Tin**

**------**

Icon

Description automatically generated with medium confidence

**Nhóm 10**:

1. Ngô Minh Thuận
2. Hồ Thị Mỹ Quỳnh
3. Huỳnh Văn Duy

**Đề Tài : Mô Hình Tái Sinh Thiết Bị Điện Tử với Kiến Trúc Microservice**

**Báo cáo tiểu luận**

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

Mục lục

[1. Giới thiệu (Introduction) 3](#_Toc167047411)

[a. Tổng quan. 3](#_Toc167047412)

[b. Mục đích chính của đề tài: 4](#_Toc167047413)

[c. Phạm vi đề tài 4](#_Toc167047414)

[2. Kiến trúc phần mềm (Software Architecture) 6](#_Toc167047415)

[a. Giới thiệu về kiến trúc hướng dịch vụ (Service oriented architecture) 6](#_Toc167047416)

[b. Các thành phần của kiến trúc microservice 7](#_Toc167047417)

[d. Áp dụng vào đề tài Mô hình tái chế thiết bị điện tử 9](#_Toc167047418)

[e. Mô tả chi tiết từng thành phần. 11](#_Toc167047419)

[2.1. User Interface (Giao diện người dùng) 11](#_Toc167047420)

[2.2. API Gateway 11](#_Toc167047421)

[2.3. UserProfile Service 11](#_Toc167047422)

[2.4. Inquiry Service 11](#_Toc167047423)

[2.5. Assessment Service 11](#_Toc167047424)

[2.6. Reporting Service 11](#_Toc167047425)

[2.7. Database 12](#_Toc167047426)

[3. Thiết kế phần mềm (Software Design) 12](#_Toc167047427)

[4. Mối quan hệ giữa kiến trúc và thiết kế phần mềm (Relationship between Software Architecture and Design) 13](#_Toc167047428)

[5. Ví dụ thực tế (Case Studies) 14](#_Toc167047429)

[6. Thách thức và xu hướng hiện tại (Challenges and Current Trends) 14](#_Toc167047430)

[7. Kết luận (Conclusion) 15](#_Toc167047431)

[8. Tài liệu tham khảo (References) 15](#_Toc167047432)

[10. Phụ lục (Appendices) 15](#_Toc167047433)

# 1. Giới thiệu (Introduction)

## a. Tổng quan.

Trong giai đoạn phát triển hiện nay, các thiết bị điện tử ngày càng trở nên phổ biến và hiện diện rộng rãi trên toàn thế giới. Tuy nhiên, cùng với sự phát triển đó, vấn đề tái chế các thiết bị điện tử cũ đang trở thành một thách thức lớn. Do đó, cần thiết lập một hệ thống kết nối hiệu quả giữa khách hàng và công ty để thuận tiện trong việc trao đổi và tái chế thiết bị điện tử cũ, đặc biệt khi khách hàng nhận thấy mức giá mà công ty đề xuất là hợp lý.

Dựa trên những tiêu chí và yêu cầu đề bài đặt ra, nhóm chúng tôi đã chọn lựa kiến trúc hợp lý và thiết kế phần mềm dựa trên kiến trúc đó cho trang web tái chế. Mục tiêu là tạo ra một hệ thống thân thiện và dễ sử dụng cho cả khách hàng và công ty. Hệ thống này sẽ hỗ trợ quy trình tái chế các thiết bị điện tử cũ, chẳng hạn như điện thoại di động iPhone hoặc Galaxy, với các bước cụ thể như sau:

Yêu cầu báo giá: Đầu tiên, khách hàng sẽ truy cập trang web hoặc kiosk để yêu cầu báo giá cho thiết bị điện tử cũ của họ. Hệ thống sẽ cung cấp một mức giá dựa trên tình trạng hiện tại của thiết bị.

Gửi thiết bị: Nếu hài lòng với mức giá, khách hàng sẽ gửi thiết bị điện tử đến công ty tái chế. Công ty sẽ nhận thiết bị này và chuẩn bị cho bước tiếp theo.

Đánh giá thiết bị: Sau khi nhận được thiết bị, công ty sẽ tiến hành đánh giá tình trạng của nó để xác định xem thiết bị có hoạt động tốt hay không. Quá trình đánh giá này giúp xác định giá trị thực của thiết bị.

Thanh toán: Nếu thiết bị đang trong tình trạng hoạt động tốt, công ty sẽ gửi cho khách hàng số tiền đã hứa ban đầu. Quá trình này đảm bảo sự minh bạch và tin cậy giữa khách hàng và công ty.

Theo dõi trạng thái: Khách hàng có thể truy cập trang web bất cứ lúc nào để kiểm tra trạng thái của thiết bị của họ trong quá trình tái chế. Điều này tạo sự thuận tiện và minh bạch trong toàn bộ quy trình.

Tái chế: Dựa trên kết quả đánh giá, thiết bị sẽ được tái chế một cách an toàn hoặc được bán lại. Quy trình tái chế đảm bảo giảm thiểu tác động tiêu cực đến môi trường.

Báo cáo: Công ty sẽ thực hiện các báo cáo tài chính và hoạt động tái chế định kỳ. Những báo cáo này giúp đánh giá hiệu quả của hệ thống và điều chỉnh các chiến lược cần thiết để cải thiện quy trình tái chế.

Thông qua hệ thống này, chúng tôi hy vọng sẽ tạo ra một giải pháp hiệu quả và bền vững cho vấn đề tái chế thiết bị điện tử, góp phần bảo vệ môi trường và mang lại lợi ích kinh tế cho cả khách hàng và công ty.

## b. Mục đích chính của đề tài:

Mục đích chính của đề tài này là thiết kế và triển khai một hệ thống trực tuyến kết nối khách hàng và công ty tái chế để quản lý quy trình tái chế các thiết bị điện tử cũ một cách hiệu quả và bền vững. Hệ thống nhằm mục tiêu:

Tạo thuận lợi cho khách hàng: Cung cấp một nền tảng trực tuyến giúp khách hàng dễ dàng yêu cầu báo giá, gửi thiết bị và theo dõi trạng thái của thiết bị trong quá trình tái chế. Điều này giúp khách hàng tiết kiệm thời gian và công sức, đồng thời đảm bảo tính minh bạch trong giao dịch.

Tối ưu hóa quy trình tái chế: Tự động hóa các bước từ nhận thiết bị, đánh giá tình trạng, xử lý tái chế đến thanh toán và báo cáo. Hệ thống giúp giảm thiểu sai sót, tăng cường hiệu quả và đảm bảo các thiết bị được xử lý một cách an toàn và thân thiện với môi trường.

Thúc đẩy bảo vệ môi trường: Khuyến khích và hỗ trợ việc tái chế thiết bị điện tử cũ, góp phần giảm lượng rác thải điện tử và bảo vệ tài nguyên thiên nhiên. Hệ thống giúp nâng cao nhận thức về tầm quan trọng của tái chế và tạo động lực cho khách hàng tham gia vào quá trình này

Cải thiện mối quan hệ khách hàng - công ty: Xây dựng một hệ thống đáng tin cậy và minh bạch, tăng cường sự tin tưởng và hài lòng của khách hàng đối với dịch vụ tái chế. Điều này không chỉ giúp giữ chân khách hàng hiện tại mà còn thu hút thêm nhiều khách hàng mới.

Đưa ra các báo cáo và phân tích hiệu quả: Cung cấp các báo cáo định kỳ và phân tích dữ liệu để đánh giá hiệu quả của quá trình tái chế, từ đó giúp công ty có thể điều chỉnh chiến lược và nâng cao chất lượng dịch vụ.

Tóm lại, đề tài này không chỉ tập trung vào việc giải quyết vấn đề tái chế thiết bị điện tử cũ mà còn hướng tới việc xây dựng một hệ thống bền vững, minh bạch và thân thiện với người dùng, góp phần bảo vệ môi trường và phát triển kinh tế.

## c. Phạm vi đề tài

Phân tích và Đánh giá Thị trường:

* Nghiên cứu nhu cầu và xu hướng tái chế thiết bị điện tử trên thị trường.
* Xác định các loại thiết bị điện tử phổ biến cần tái chế như điện thoại di động, máy tính bảng, máy tính xách tay, và các thiết bị điện tử khác.
* Đánh giá các tiêu chí về tình trạng và giá trị của thiết bị điện tử cũ.

Thiết kế Kiến trúc Hệ thống:

* Lựa chọn mô hình kiến trúc hệ thống phù hợp để đáp ứng yêu cầu của đề tài.
* Thiết kế giao diện người dùng (UI) thân thiện và dễ sử dụng cho cả khách hàng và công ty.
* Xác định các thành phần chính của hệ thống bao gồm các module yêu cầu báo giá, gửi thiết bị, theo dõi trạng thái, đánh giá, thanh toán, tái chế và báo cáo.

Phát triển Phần mềm:

* Lập trình các chức năng chính của hệ thống theo mô hình đã thiết kế.
* Tích hợp các công nghệ và công cụ cần thiết để đảm bảo hệ thống hoạt động hiệu quả.
* Xây dựng cơ sở dữ liệu để quản lý thông tin khách hàng, thiết bị, và các giao dịch tái chế.

Quy trình Tái chế Thiết bị Điện tử:

* Mô tả chi tiết các bước trong quy trình tái chế từ lúc khách hàng yêu cầu báo giá đến khi thiết bị được tái chế hoặc bán lại.
* Đảm bảo quy trình tái chế tuân thủ các tiêu chuẩn và quy định về bảo vệ môi trường.
* Kiểm thử và Triển khai Hệ thống:
* Thực hiện các bước kiểm thử để đảm bảo hệ thống hoạt động đúng như mong đợi và không gặp lỗi.
* Đánh giá hiệu quả của hệ thống qua các thử nghiệm thực tế.
* Triển khai hệ thống trên môi trường thực tế và đảm bảo hỗ trợ kỹ thuật cho người dùng.

Báo cáo và Phân tích Kết quả:

* Thu thập và phân tích dữ liệu từ hệ thống để đánh giá hiệu quả của quy trình tái chế.
* Cung cấp các báo cáo định kỳ và phân tích kết quả để điều chỉnh và cải thiện hệ thống.

Phạm vi đề tài tập trung vào việc xây dựng một hệ thống tái chế thiết bị điện tử cũ, từ giai đoạn nghiên cứu và thiết kế đến phát triển, triển khai và đánh giá. Đề tài không chỉ nhằm giải quyết vấn đề tái chế mà còn tạo ra một quy trình hiệu quả, minh bạch và thân thiện với người dùng, góp phần bảo vệ môi trường và phát triển kinh tế bền vững.

# 2. Kiến trúc phần mềm (Software Architecture)

## a. Giới thiệu về kiến trúc hướng dịch vụ (Service oriented architecture)



Microservice là một thuật ngữ xuất hiện vào đầu những năm 2010, dùng để chỉ một phong cách phát triển phần mềm tiên tiến và hiện đại. Trong mô hình này, các ứng dụng được xây dựng từ nhiều dịch vụ nhỏ và độc lập, được gọi là "microservices". Mỗi microservice được thiết kế để thực hiện một chức năng duy nhất hoặc một nhóm nhỏ các chức năng liên quan, và chúng có thể được quản lý một cách độc lập. Điều này mang lại sự linh hoạt và khả năng mở rộng cao cho hệ thống phần mềm, vì mỗi microservice có thể được phát triển, triển khai, và mở rộng một cách riêng biệt mà không ảnh hưởng đến các phần khác của ứng dụng.

Microservice cho phép các nhóm phát triển phần mềm làm việc đồng thời trên các phần khác nhau của hệ thống, tăng cường khả năng triển khai liên tục và cải thiện thời gian phản hồi đối với yêu cầu của khách hàng. Điều này cũng đồng nghĩa với việc các tổ chức có thể nhanh chóng áp dụng các công nghệ mới và cập nhật các phần cụ thể của ứng dụng mà không cần phải thay đổi toàn bộ hệ thống. Hơn nữa, vì mỗi microservice có thể được viết bằng một ngôn ngữ lập trình khác nhau hoặc sử dụng các công nghệ khác nhau, các nhóm phát triển có thể lựa chọn công cụ phù hợp nhất cho từng chức năng cụ thể.

Microservice cũng giúp cải thiện khả năng chịu lỗi của hệ thống, vì sự cố trong một microservice không ảnh hưởng đến các microservice khác. Nhờ vào sự phân chia rõ ràng này, việc duy trì và nâng cấp hệ thống trở nên dễ dàng hơn, giúp giảm thiểu thời gian ngừng hoạt động và tăng cường trải nghiệm người dùng. Ngoài ra, mô hình microservice còn hỗ trợ việc triển khai trên các nền tảng đám mây và môi trường phân tán, tạo điều kiện thuận lợi cho việc quản lý tài nguyên hiệu quả và giảm chi phí vận hành.

Tóm lại, kiến trúc microservice đã mang lại một cuộc cách mạng trong cách xây dựng và quản lý ứng dụng phần mềm, mang lại nhiều lợi ích về mặt linh hoạt, mở rộng, và hiệu quả cho các tổ chức phát triển phần mềm hiện đại.

## b. Các thành phần của kiến trúc microservice

Kiến trúc microservice là một phong cách thiết kế phần mềm trong đó ứng dụng được xây dựng từ nhiều dịch vụ nhỏ, độc lập, mỗi dịch vụ thực hiện một chức năng cụ thể và có thể triển khai riêng biệt. Các thành phần chính của kiến trúc microservice bao gồm:

**Dịch vụ (Services):**

Microservices: Các dịch vụ nhỏ, độc lập, mỗi dịch vụ thực hiện một chức năng cụ thể của ứng dụng. Các microservice này có thể được phát triển, triển khai và mở rộng riêng biệt.

**API Gateway:**

Là điểm vào duy nhất cho tất cả các client để truy cập vào các microservice. API Gateway xử lý các yêu cầu từ client, định tuyến chúng đến các microservice thích hợp, và có thể thực hiện các chức năng như xác thực, cân bằng tải, và tổng hợp phản hồi từ nhiều microservice.

**Service Registry and Discovery:**

Service Registry: Lưu trữ thông tin về các microservice đang chạy, bao gồm vị trí và trạng thái của chúng.

Service Discovery: Cơ chế cho phép các microservice và API Gateway tìm kiếm và kết nối với các microservice khác thông qua Service Registry.

**Load Balancer:**

Cân bằng tải giữa các instance của microservice để đảm bảo tính sẵn sàng và hiệu suất. Nó có thể được tích hợp vào API Gateway hoặc hoạt động độc lập.

Quản lý cấu hình của các microservice một cách tập trung và động. Điều này giúp các microservice lấy cấu hình từ một nguồn duy nhất, đảm bảo tính nhất quán và dễ dàng quản lý.

**Monitoring and Logging:**

Monitoring: Theo dõi hiệu suất và sức khỏe của các microservice, bao gồm việc thu thập số liệu, phân tích và báo cáo.

Logging: Ghi lại các sự kiện và hoạt động của các microservice để hỗ trợ trong việc phát hiện và xử lý lỗi.

**Security:**

Bảo vệ các microservice khỏi các mối đe dọa bảo mật. Điều này bao gồm xác thực, ủy quyền, mã hóa dữ liệu và bảo mật mạng.

**Data Management:**

Database per Service: Mỗi microservice quản lý dữ liệu của riêng nó và sử dụng cơ sở dữ liệu riêng biệt. Điều này giúp giảm thiểu sự phụ thuộc lẫn nhau và tăng khả năng mở rộng.

Event Sourcing and CQRS: Các mẫu thiết kế để quản lý trạng thái và xử lý sự kiện trong hệ thống phân tán.

**Inter-Service Communication:**

Synchronous Communication: Sử dụng HTTP/REST hoặc gRPC để các microservice gọi trực tiếp lẫn nhau.

Asynchronous Communication: Sử dụng các hệ thống nhắn tin như RabbitMQ, Kafka, hoặc hệ thống hàng đợi để giao tiếp không đồng bộ giữa các microservice.

**DevOps and CI/CD Pipeline:**

Continuous Integration/Continuous Deployment (CI/CD): Các pipeline tự động để xây dựng, kiểm thử, và triển khai các microservice.

Infrastructure as Code (IaC): Sử dụng mã để quản lý và cung cấp tài nguyên cơ sở hạ tầng, giúp tự động hóa và dễ dàng quản lý môi trường triển khai.

**Containerization and Orchestration:**

Containers: Đóng gói các microservice vào các container để đảm bảo tính nhất quán môi trường giữa phát triển, kiểm thử và sản xuất.

Orchestration: Sử dụng các công cụ như Kubernetes, Docker Swarm để quản lý việc triển khai, mở rộng và quản lý vòng đời của các container.

**Resilience and Fault Tolerance:**

Các cơ chế đảm bảo hệ thống vẫn hoạt động ngay cả khi một số microservice gặp sự cố. Bao gồm các mẫu thiết kế như Circuit Breaker, Retry, và Bulkhead.

Những thành phần này cùng nhau tạo nên một hệ thống microservice linh hoạt, dễ dàng quản lý và có khả năng mở rộng cao, đáp ứng tốt nhu cầu phát triển và triển khai ứng dụng hiện đại.

## d. Áp dụng vào đề tài Mô hình tái chế thiết bị điện tử

Mô hình tái chế thiết bị điện tử là một giải pháp sáng tạo nhằm giúp khách hàng dễ dàng trao đổi các thiết bị điện tử cũ, không còn nhu cầu sử dụng, với các công ty để nhận lại một khoản tiền. Mô hình này không chỉ mang lại lợi ích tài chính cho khách hàng mà còn giúp các công ty tận dụng các sản phẩm thiết bị cũ để tái chế, lấy các linh kiện còn sử dụng được hoặc bán lại dưới dạng đồ cũ cho những người có nhu cầu.

**Lợi ích đối với người dùng:**

Người dùng có thể tạo yêu cầu thu thiết bị điện tử cũ một cách đơn giản qua hệ thống. Họ cũng có thể theo dõi trạng thái xử lý của thiết bị, từ khi yêu cầu được gửi đi cho đến khi nhận được báo giá và thanh toán từ công ty.

**Lợi ích đối với doanh nghiệp:**

Nhân viên công ty có thể nhận và xử lý các yêu cầu thu mua thiết bị từ người dùng. Họ sẽ tiến hành đánh giá tình trạng của thiết bị và thông báo giá tiền cho khách hàng. Quá trình này không chỉ giúp công ty thu thập được các linh kiện còn giá trị mà còn cung cấp các thiết bị tái sử dụng cho thị trường, góp phần giảm thiểu rác thải điện tử.

**Quản lý và báo cáo:**

Công ty cần lập các báo cáo tài chính chi tiết về tình hình thu chi hàng tháng, hàng năm để theo dõi hiệu quả hoạt động. Điều này giúp doanh nghiệp quản lý tài chính tốt hơn và đưa ra các chiến lược kinh doanh phù hợp.

**Kiến trúc hệ thống đề xuất:**

Sau khi phân tích nghiệp vụ của dự án, hệ thống được đề xuất chia thành hai phần riêng biệt: nhóm dịch vụ cho người dùng và nhóm dịch vụ cho nhân viên công ty. Để đảm bảo tính phụ thuộc lỏng lẻo (loose coupling) và tối ưu hóa hiệu suất, cơ sở dữ liệu cũng sẽ được tách riêng thành hai cụm.

**Cụm cơ sở dữ liệu cho người dùng:**

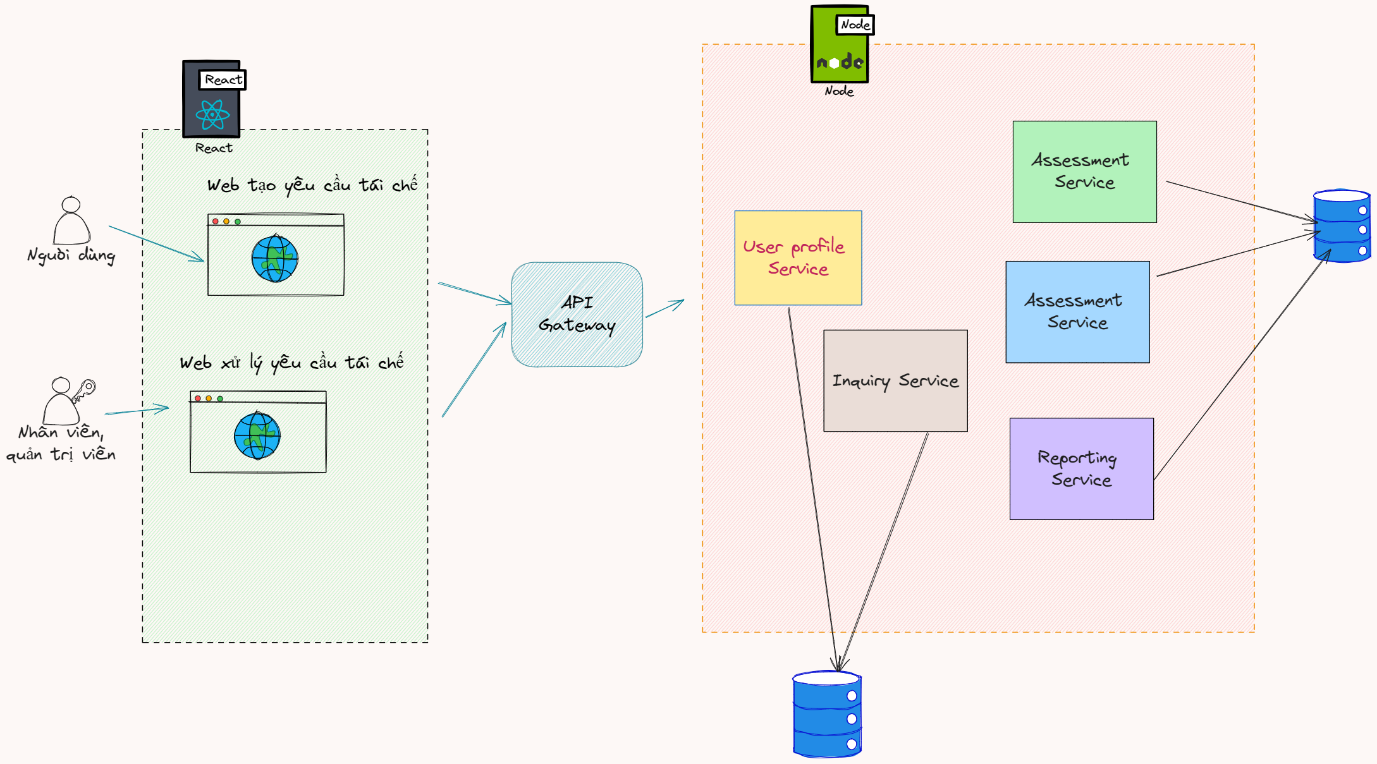
Chứa các bảng liên quan đến thông tin người dùng, yêu cầu thu thiết bị điện tử, và trạng thái xử lý yêu cầu.

**Cụm cơ sở dữ liệu cho quản trị viên và nhân viên:**

Bao gồm các bảng lưu lại lịch sử đánh giá thiết bị, các thiết bị điện tử đã thu mua, và các thông tin liên quan đến quy trình xử lý và tái chế.

**Thiết kế tổng quan của hệ thống**:

Hệ thống sẽ được thiết kế để tối ưu hóa trải nghiệm người dùng và hiệu quả quản lý của doanh nghiệp. Người dùng sẽ có một giao diện trực quan để tạo và theo dõi yêu cầu của họ. Nhân viên công ty sẽ có công cụ mạnh mẽ để xử lý các yêu cầu một cách hiệu quả, từ đánh giá tình trạng thiết bị đến quản lý quá trình tái chế và bán lại.

Với mô hình này, không chỉ tạo ra một quy trình tái chế thiết bị điện tử hiệu quả và thân thiện với người dùng mà còn góp phần vào bảo vệ môi trường và tối ưu hóa nguồn lực trong nền kinh tế tuần hoàn.

Kiến trúc cho hệ thống tái chế thiết bị điện tử

## e. Mô tả chi tiết từng thành phần.

### 2.1. User Interface (Giao diện người dùng)

Giao diện người dùng được phân chia thành hai phần: một là dành cho khách hàng và hai là dành cho quản trị viên cũng như nhân viên. Trên nền tảng web, cả khách hàng và nhân viên đều có thể dễ dàng tương tác với các biểu mẫu yêu cầu báo giá, kiểm tra trạng thái của thiết bị hoặc xem các báo cáo thống kê.

### 2.2. API Gateway

API Gateway đóng vai trò là điểm nhập duy nhất cho mọi yêu cầu từ giao diện người dùng. Nó thực hiện nhiệm vụ cân bằng tải, xác thực và định tuyến các yêu cầu tới các dịch vụ thích hợp.

### 2.3. UserProfile Service

Userprofile Service đảm nhận vai trò quản lý xác thực và phân quyền người dùng. Tại đây, người dùng có thể đăng ký tài khoản và cũng có thể truy cập để xem hoặc chỉnh sửa thông tin cá nhân của mình.

### 2.4. Inquiry Service

Inquiry Service là dịch vụ cho phép khách hàng tạo yêu cầu thu thiết bị di động. Khách hàng có thể xem lịch sử các yêu cầu, chỉnh sửa thông tin về sản phẩm họ muốn gửi cho công ty. Ngoài ra, dịch vụ cung cấp thông tin về trạng thái hiện tại của thiết bị trong quá trình tái chế, giúp khách hàng theo dõi quy trình một cách dễ dàng.

### 2.5. Assessment Service

Dịch vụ cung cấp các tính năng cho nhân viên. Họ có thể tạo các phiếu để xử lý quy trình đánh giá sản phẩm. Sau khi đánh giá xong, nhân viên có thể cập nhật trạng thái sản phẩm, quyết định liệu nó sẽ được bán lại hay tái chế, cũng như số tiền sẽ được trả lại cho khách hàng.

Thông tin về các sản phẩm sau khi được xem xét kỹ lưỡng cũng sẽ được lưu trong hệ thống. Dịch vụ cho phép nhân viên xem và chỉnh sửa thông tin của các thiết bị điện tử này.

### 2.6. Reporting Service

Tạo báo cáo về hoạt động tái chế và tình hình tài chính. Các thông tin về doanh thu, phiếu thu chi sẽ được xử lý tại service này

### 2.7. Database

Database đóng vai trò quan trọng trong việc lưu trữ dữ liệu liên quan đến thiết bị, báo giá, đánh giá, giao dịch và báo cáo. Để tối ưu hóa quản lý dữ liệu, chúng ta đã phân chia database thành hai cụm: một cụm chứa dữ liệu được xử lý bởi khách hàng và một cụm khác chứa dữ liệu được xử lý bởi nhân viên.

Tuy nhiên, một thách thức nảy sinh là làm thế nào để đảm bảo tính nhất quán và toàn vẹn của dữ liệu khi chia thành các cụm riêng lẻ. Để giải quyết vấn đề này, chúng ta đã thiết lập lịch trình chạy các dịch vụ batch để cập nhật dữ liệu và đồng bộ hóa giữa các cụm database. Các công việc batch này sẽ được kích hoạt vào những khoảng thời gian cụ thể, thường là trong các khoảng thời gian hệ thống không hoạt động chủ yếu, như những khoảng thời gian nghỉ của hệ thống, để tránh ảnh hưởng đến trải nghiệm của người dùng.

# 3. Thiết kế phần mềm (Software Design)

1. Ảnh có chứa biểu đồ, văn bản, hàng

   Mô tả được tạo tự độngUse case tổng quát
2. Đặc tả use case.
3. Activity.

# 4. Mối quan hệ giữa kiến trúc và thiết kế phần mềm (Relationship between Software Architecture and Design)

1. Sự liên quan và phân biệt: Làm rõ sự khác biệt và mối liên hệ giữa kiến trúc phần mềm và thiết kế phần mềm.

Kiến trúc phần mềm và thiết kế phần mềm là hai khái niệm quan trọng trong quá trình phát triển phần mềm, và chúng có mối liên hệ mật thiết nhưng cũng có những khác biệt đáng chú ý.

**Kiến trúc phần mềm:**

Khái niệm: Kiến trúc phần mềm là bản thiết kế cơ bản của hệ thống phần mềm, bao gồm cấu trúc tổng thể, các thành phần, các mối quan hệ giữa chúng và các nguyên tắc hướng dẫn cách chúng tương tác với nhau.

Mục tiêu: Mục tiêu của kiến trúc phần mềm là định hình cách thức tổ chức và triển khai các thành phần của hệ thống phần mềm để đảm bảo tính linh hoạt, hiệu quả và dễ bảo trì.

Phạm vi: Kiến trúc phần mềm tập trung vào việc xác định các yếu tố cấu thành hệ thống, như các thành phần, cấu trúc tổ chức và mối quan hệ giữa chúng, mà không đi sâu vào chi tiết về việc cài đặt từng phần.

**Thiết kế phần mềm:**

Khái niệm: Thiết kế phần mềm là quá trình tạo ra các kết cấu cụ thể cho mỗi thành phần của hệ thống phần mềm, bao gồm cách tổ chức mã nguồn, cấu trúc dữ liệu, giao diện người dùng và các chi tiết kỹ thuật khác.

Mục tiêu: Mục tiêu của thiết kế phần mềm là chuyển đổi kiến trúc phần mềm thành các thiết kế cụ thể, chi tiết hóa cách thức triển khai các tính năng và chức năng của hệ thống.

Phạm vi: Thiết kế phần mềm tập trung vào việc biến đổi các yếu tố kiến trúc thành các thiết kế cụ thể, tạo ra các kết cấu mã nguồn, cơ sở dữ liệu, giao diện và các thành phần khác của hệ thống.

Mối liên hệ:

Kiến trúc phần mềm cung cấp khung tổng thể cho quá trình thiết kế phần mềm. Thiết kế phần mềm tiếp tục từ kiến trúc, cung cấp các kết cấu cụ thể và chi tiết cho hệ thống dự án.

Một kiến trúc phần mềm tốt sẽ hướng dẫn thiết kế phần mềm sao cho hệ thống có thể dễ dàng mở rộng, bảo trì và mở rộng trong tương lai.

Trong tóm tắt, kiến trúc phần mềm quyết định cách tổ chức các thành phần cơ bản của hệ thống, trong khi thiết kế phần mềm tập trung vào việc tạo ra các kết cấu cụ thể và chi tiết hóa kiến trúc phần mềm thành các thiết kế hợp lý.

1. Quy trình tích hợp: Cách hai lĩnh vực này tích hợp và hỗ trợ lẫn nhau trong quá trình phát triển phần mềm.

# 5. Ví dụ thực tế (Case Studies)

1. Trình bày ví dụ cụ thể: Chọn một hoặc vài ví dụ thực tế về hệ thống phần mềm, phân tích kiến trúc và thiết kế của chúng.
2. Bài học rút ra: Những bài học quý báu từ các ví dụ này.

# 6. Thách thức và xu hướng hiện tại (Challenges and Current Trends)

1. Thách thức: Những khó khăn và thách thức chính trong việc thiết kế và triển khai kiến trúc phần mềm hiện đại.
2. Xu hướng mới: Các xu hướng và công nghệ mới trong kiến trúc và thiết kế phần mềm, như kiến trúc serverless, AI trong thiết kế phần mềm, v.v.

# 7. Kết luận (Conclusion)

1. Tóm tắt: Tóm tắt lại những điểm chính đã trình bày trong bài tiểu luận.
2. Kết luận: Những kết luận quan trọng và ý nghĩa của nghiên cứu.
3. Hướng phát triển tương lai: Gợi ý về những nghiên cứu hoặc cải tiến có thể trong tương lai.

# 8. Tài liệu tham khảo (References)

# 10. Phụ lục (Appendices)

Bao gồm các tài liệu bổ sung như sơ đồ, biểu đồ, hoặc mã nguồn mà bạn đã đề cập trong bài tiểu luận.