A logo with red and blue letters

Description automatically generated

**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP. HỒ CHÍ MINH**

**BÁO CÁO**

**KIẾN TRÚC VÀ THIẾT KẾ PHẦN MỀM**

**Đề tài : CÔNG CỤ TÁI CHẾ THIẾT BỊ ĐIỆN THOẠI DI ĐỘNG**

**Giáo viên hướng dẫn:** Võ Văn Hải

**Lớp học phần:** ĐHKTPM16ETT

**Mã học phần:** 422000191403

**Danh sách thành viên:**

|  |  |  |
| --- | --- | --- |
| **STT** | **MSSV** | **HỌ TÊN** |
| 1 | 20065261 | NGUYỄN VIẾT NGỌC THỊNH |
| 2 | 20053561 | HỒ QUỐC HUY |
| 3 | 20064261 | HOÀNG TRỌNG NHÂN |

*Khoa Công nghệ thông tin, Trường Đại học Công nghiệp thành phố Hồ Chí Minh*

**ReSIUH: CÔNG CỤ TÁI CHẾ THIẾT BỊ ĐIỆN THOẠI DI ĐỘNG**

HỒ QUỐC HUY, HOÀNG TRỌNG NHÂN, NGUYỄN VIẾT NGỌC THỊNH

*Khoa Công nghệ thông tin, Trường Đại học Công nghiệp thành phố Hồ Chí Minh*

**Tóm tắt.**

Sự gia tăng phế thải điện tử, đặc biệt là điện thoại di động, đang gây ra những tác động tiêu cực đến môi trường và xã hội. Việc tái chế điện thoại di động không chỉ giúp giảm thiểu ô nhiễm mà còn tận dụng được nguồn tài nguyên quý giá. Điều này đặt ra một nhu cầu cấp thiết cho việc tái chế hiệu quả. Chúng tôi đề xuất một công cụ có tên là ReSIUH nhằm giúp người dùng tái chế thiết bị di động một cách dễ dàng và thuận tiện. Nó giáo dục người dùng, cung cấp các dịch vụ cần thiết cho việc thu gom điện thoại di động cũ. Công cụ này chúng tôi tập trung vào 2 phân hệ: Ứng dụng web và ứng dụng cho thiết bị di động. Hai phân hệ này chúng tôi triển khai đồng bộ với nhau nhằm tăng cường khả năng xử lý và truyền tải thông tin kịp thời người dùng. Cả hai phân hệ này chúng tôi tập trung giải quyết các vấn đề như: Định giá thiết bị, Tìm kiếm điểm thu gom, Quản lý đơn hàng và theo dõi quy trình tái chế. Chúng tôi sử dụng các công nghệ như Java, Javascript, Spring Boot, ReactJS và React Native để phát triển ứng dụng này. Ngoài ra, chúng tôi cũng sử dụng điện toán đám mây (cloud computing) và cơ sở hạ tầng dưới dạng mã (infrastructure as code) để triển khai và quản lý ứng dụng một cách hiệu quả. Ứng dụng web này sẽ được triển khai và đánh giá trên một nhóm người dùng mẫu để kiểm tra tính hiệu quả và khả năng đáp ứng nhu cầu của người dùng.

**Từ khóa:** Ứng dụng trên di động, phế thải điện tử, tái chế điện thoại di động, công cụ ReSIUH

**GIỚI THIỆU TỔNG QUAN**

Theo nhu cầu thi trường hiện nay, một lượng lớn về thiết bị điện tử bị loại bỏ mỗi năm. Rác thải điện tử không chỉ gây ô nhiễm môi trường mà còn lãng phí các nguồn tài nguyên quý qiá như vàng, bạc, đồng. Việc tái chế các thiết bị điện tử cũ, không chỉ giúp giảm thiểu số lượng rác thải điện tử mà còn có thể tái chế các thiết bị này.

Trong thời đại công nghệ phát triển nhanh chóng hiện nay, việc tái chế và xử lý các thiết bị điện tử cũ đang trở thành một vấn đề không chỉ về môi trường mà còn về tài nguyên. Để đáp ứng nhu cầu này, chúng tôi giới thiệu hệ thống web tái chế điện thoại cũ, một giải pháp hiệu quả và linh hoạt cho các doanh nghiệp và tổ chức trong ngành công nghiệp thu mua và tái chế.

Hệ thống web tái chế điện thoại cũ được phát triển với mục tiêu cung cấp một nền tảng toàn diện để quản lý và tự động hóa quy trình tái chế các thiết bị điện tử cũ. Về kiến trúc dựa trên dịch vụ, ứng dụng này cung cấp và cho phép ta sử dụng một loạt tính năng để hỗ trợ việc tái chế, xử lý và thu mua các linh kiện đồ dùng điện tử cũ, dựa trên kiến trúc Microservices, nhằm đảm bảo tính linh hoạt, khả năng mở rộng và hiệu quả hoạt động cao. Chẳng hạn như:

Trong hệ thống web tái chế đồ điện tử được sử dụng để tái chế các thiết bị điện tử cũ (**chẳng hạn như Iphone hoặc SamSung).** Quy trình xử lý tái chế các thiết bị điện tử cũ hoạt động như sau: Đầu tiên, khách hàng hỏi công ty (**thông qua trang web hoặc ki-ốt**) xem họ có thể nhận được bao nhiêu tiền cho thiết bị điện thoại cũ (**gọi là báo giá**). Nếu hài lòng, khách hàng sẽ gửi thiết bị điện tử cũ cho công ty tái chế, công ty sẽ nhận thiết bị từ khách hàng (**gọi là nhận hàng**). Sau khi nhận, công ty tái chế sẽ nhận thiết bị sẽ đánh giá thiết bị để xác định xem thiết bị có hoạt động tốt hay không (**gọi là đánh giá**). Nếu thiết bị tốt thì sẽ gửi tiền đã hứa cho khách hàng (gọi là kế toán). Trong quá trình này, khách hàng có thể vào trang web bất cứ lúc nào để xem xét hoặc kiểm tra trên hệ thống về trạng thái của món hàng (**Trạng thái của món hàng**), dựa trên đánh giá đó thiết bị sẽ được tiêu hủy an toàn hay bán lại . Cuối cùng, công ty sẽ định kỳ và chạy các báo cáo tài chính và hoạt động theo yêu cầu, lịch trình dựa trên hoạt động tái chế (**gọi là báo cáo**).

Để xử lý các vấn đề đó, chúng tôi đã sử dụng csdl là MySql để có thể lưu trữ tên khách hàng và linh kiện điện tử để bên công ty có thể dựa vào đó để biết được thông tin của linh kiện điện tử mà khách hàng đã gửi đẻ bên công ty có thể thẩm định và báo giá lại cho khách hàng.

Ứng dụng phân hủy an toàn và tái chế thiết bị điện tử được xây dựng trên kiến trúc microservices , cho phép các thành phần của hệ thống hoạt động độc lập và phối hợp với nhau một cách hiệu quả. Mỗi Microservices đảm nhận một chức năng cụ thể và có thể phát triển, triển khai và mở rộng .

**CÁC CÔNG NGHỆ LIÊN QUAN**

**2.1 Ngôn ngữ phát triển ứng dụng**

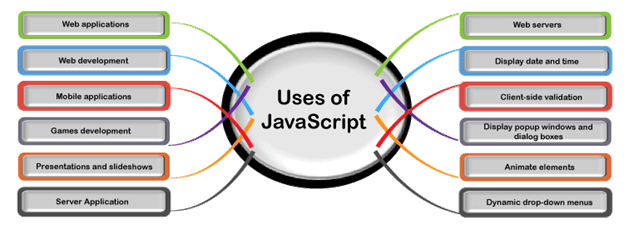
***2.1.1 Ngôn ngữ lập trình Java***

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 [5].

***2.1.2 Ngôn ngữ lập trình JavaScript***

Javascript ra đời năm 1995, ban đầu Javascript không được xem là một ngôn ngữ lập trình mà chỉ được xem là một ngôn ngữ kịch bản (scripting language). Nguyên nhân là thời bấy giờ Javascript chỉ được sử dụng để xây dựng các tương tác với trang Web. Bản thân nó phải thực thi trong một chương trình không phải hệ điều hành (cụ thể là trình duyệt Web) [6]. Nhưng sau này, khi vai trò và sự phổ biến của websites ngày càng lớn, cộng với đó là sự ra đời và lớn mạnh thần tốc của NodeJS (2009) giúp Javascript không còn là một ngôn ngữ phải thực thi trong trình duyệt. Hiện Javascript chính thức được công nhận là một ngôn ngữ lập trình chính thống. Javascript hoạt động được trên nhiều nền tảng (Web, Server) và là ngôn ngữ đơn giản, dễ học cho người mới sử dụng. Hiện nay, nhiều công nghệ xây dựng dựa trên Javascript như Nodejs, React, VueJS...

Đồng thời Javasript cũng được dùng cho việc xây dựng nhiều ứng dụng với vai trò khác nhau từ Web, ứng dụng Mobile và một số ứng dụng khác (Hình 2.1.1).

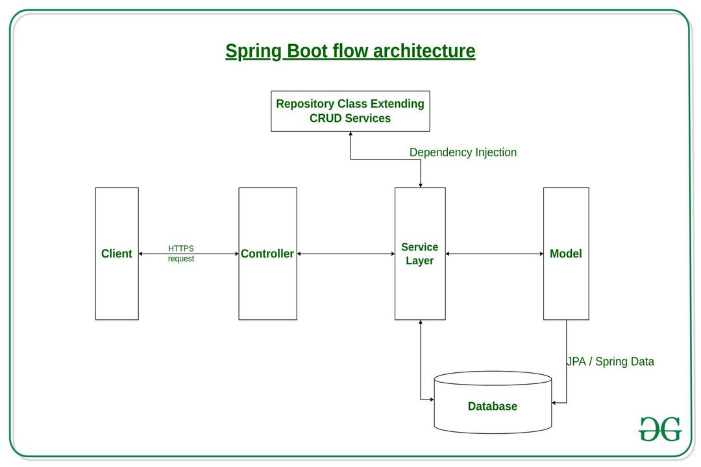


*Hình 2.21 Những ứng dụng được phát triển bởi JavaScript [6]*

**2.2 Những framework hỗ trợ phát triển ứng dụng**

***2.2.1 Framework Spring Boot***

Spring Boot là khuôn mẫu (framework) phát triển dựa trên ngôn ngữ Java trong hệ sinh thái Spring framework [7]. Nó giúp cho các lập trình viên đơn giản hóa quá trình lập trình một ứng dụng với Spring. Trong Spring Boot, người lập trình chỉ tập trung vào việc phát triển các nghiệp vụ cho ứng dụng mà không quan tâm nhiều về kiến trúc và mã nguồn của ứng dụng. Một kiến trúc webserver xây dựng dựa trên spring boot gồm các thành phần cơ bản như sau: Repository Class Extending, Service Layer, Model, Controller (Hình 2.2.2)

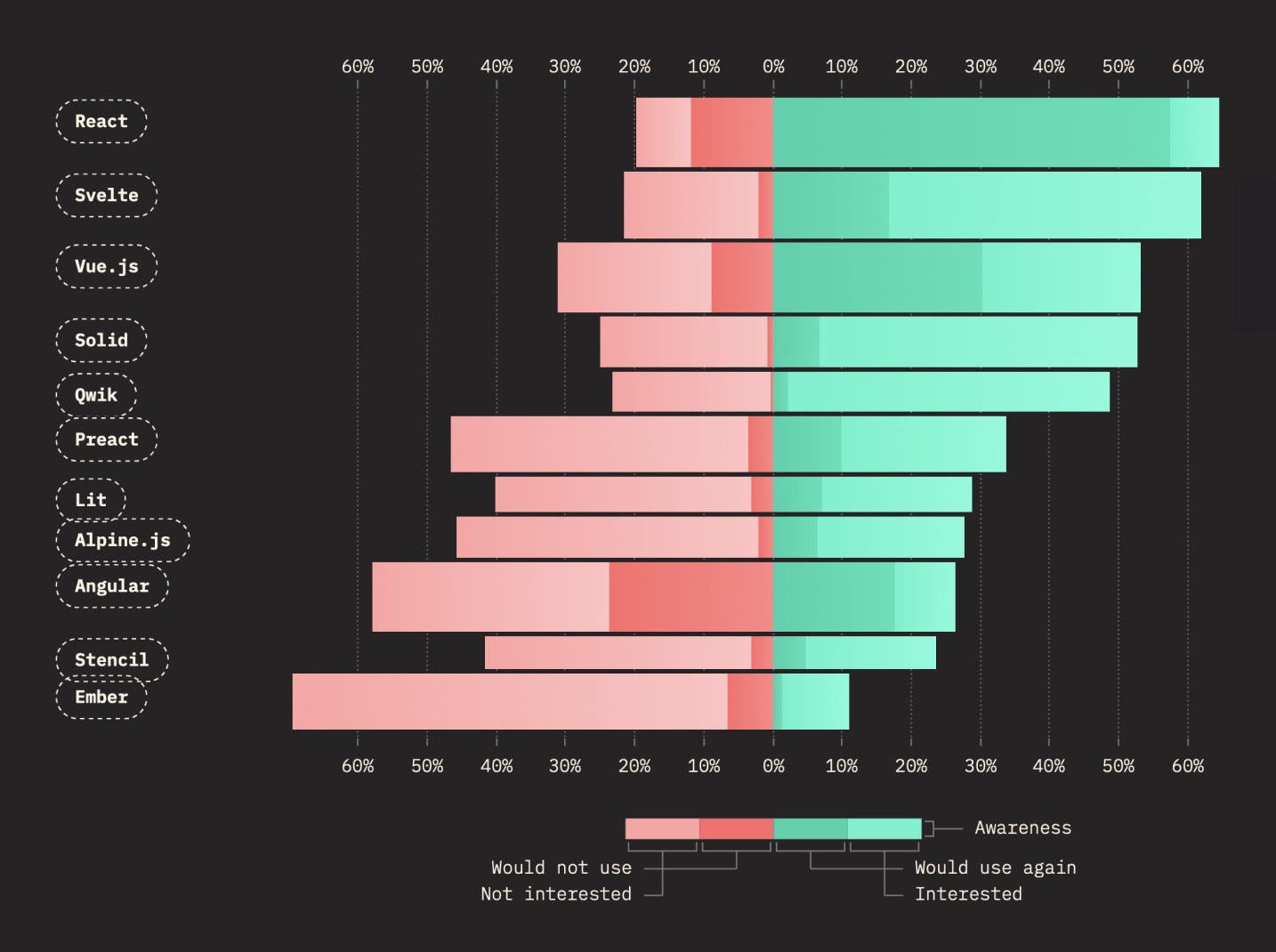


*Hình 2.2.2 Kiến trúc của luồng hoạt động của Spring Boot [7]*

Repository Class Extending là các thư viện hoặc framework được phát triển và duy trì bởi cộng đồng các lập trình viên Java để hỗ trợ quá trình xây dựng hệ thống một cách nhanh chóng hoặc cũng có thể là các lớp chứa các phương thức xử lý, truy xuất dữ liệu (CRUD) do lập trình viên viết ra; Service Layer là lớp dịch vụ chứa các phương thức (hàm) xử lý các yêu cầu như chỉnh sửa, lớp Service thường sử dụng các phương thức của các lớp trong Repository Class Extending (bằng cách gọi tới một hay kết hợp nhiều phương thức với nhau) để truy xuất, thêm, xóa dữ liệu trong cơ sở dữ liệu. Nó là nơi tập trung các xử lý nghiệp vụ của hệ thống Back-end; Model là thành phần được đính kèm theo các truy vấn của Services Layer xuống cơ sở dữ liệu. Model đóng nhiều vai trò quan trọng như phát sinh hệ cơ sở dữ liệu theo liên kết các lớp của lập trình viên, chuyển đổi kết quả truy xuất từ cơ sở dữ liệu thành các thực thể lớp; Controller là thành phần giao tiếp trực tiếp với phía client, tồn tại dưới dạng các API. Model có trách nhiệm xử lý yêu cầu của client, xác minh các yêu cầu đó. Model truy cập đến phương thức phù hợp trong Service Layer để đạt được kết quả mong muốn, trả kết quả về cho phía client.

***2.2.2 Framework React Native, ReactJs***

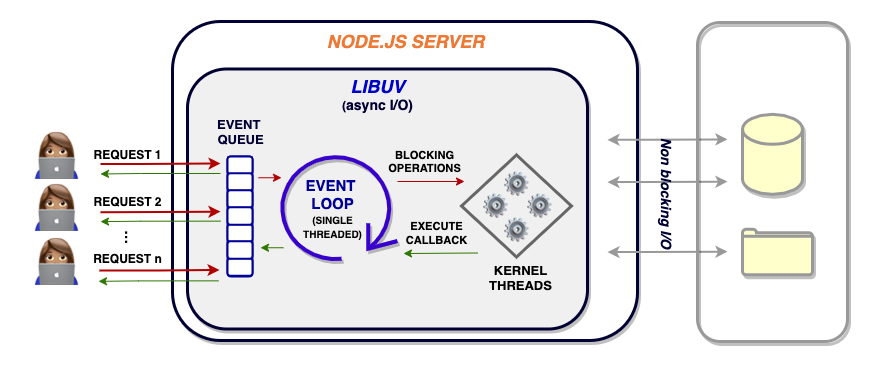
React Native là một thư viện mở dùng xây dựng ứng dụng di động dựa trên JavaScript. React Native được phát triển và duy trì bởi Facebook [8]. React Native giúp phát triển ứng dụng di động một cách nhanh chóng. Nó hỗ trợ cho các ứng dụng chạy được trên hai nền tảng iOS và Android. React Native giúp lập trình viên chỉ cần phát triển ứng dụng trên một mã nguồn duy nhất mà vẫn có thể tương thích với cả hai hệ điều hành Android và iOS (thậm chí là cả các hệ điều hành cho Ipad và và các dòng tivi). Ngoài ra, React.js nói chung và React Native nói riêng là công nghệ được yêu thích hiện nay. Phong cách xây dựng mã nguồn trong React Native là phân chia các khối chức năng giúp dễ dàng trong việc quản lý, bảo trì, sửa lỗi, nâng cấp ứng dụng. Tương tự React Native, ReactJs là công nghệ sử dụng việc xây dựng ứng dụng web dựa trên JavaScript được duy trì và phát triển bởi Facebook. ReactJs được nhận xét là dễ học, dễ hiểu, dễ làm quen nhưng hiệu quả lại cao, phù hợp với nhiều ứng dụng web, kể cả những ứng dụng có quy mô cực lớn (như Facebook). Tương tự React Native, ReactJs cũng có cộng đồng lớn mạnh, nhiều thư viện và framework hỗ trợ [9]. Đội ngũ phát triển React cũng đang không ngừng làm việc hằng ngày để phát triển công nghệ này ngày một tốt hơn. Theo khảo sát của Stack Overflow, ReactJS là công nghệ Front end được nhiều người quan tâm nhất trong vòng những năm từ 2019 trở lại đây (Hình 2.2.3). Cụ thể trong năm 2020, React.js dẫn đầu bảng khảo sát ở mức độ phổ biến và yêu thích trong danh sách các framework Front-end [10].



*Hình 2.2.3 Khảo sát sự phổ biến của các framework Frontend [10]*

***2.2.3 NodeJs***

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. Hệ thống web xây dựng từ Node JS có khả năng hoạt động ổn định, khắc phục nhiều điểm yếu của các webserver xây dựng dựa trên các công nghệ đa luồng (các bài toán chống ách tắc luồng, xử lý đồng bộ và bất đồng bộ khi cùng truy cập đến một hoặc nhiều tài nguyên,… là những bài toán lớn đối với các lập trình viên). Bên cạnh đó, NodeJS có số lượng rất lớn các thư viện được phát triển bởi cộng đồng lập trình viên. Hiện nay số lượng thư viện đã vượt qua con số 1,3 triệu là minh chứng cho thấy sức ảnh hưởng của công nghệ này.



*Hình 2.2.4 Kiến trúc của luồng hoạt động của Node JS [11]*

**2.3 Cloud Computing**

Điện toán đám mây (Cloud Computing) là việc ảo hóa các tài nguyên tính toán và các ứng dụng. Thay vì việc sử dụng một hoặc nhiều máy chủ thật (ngay trước mắt, có thể sờ được, có thể tự bạn ấn nút bật tắt được) thì nay sẽ sử dụng các tài nguyên được ảo hóa (virtualized) thông qua môi trường internet [12]. Việc phân phối các tài nguyên máy tính theo nhu cầu qua internet với chính sách thanh toán theo mức sử dụng. Thay vì mua, sở hữu và bảo trì các trung tâm dữ liệu và máy chủ vật lý, có thể tiếp cận các dịch vụ công nghệ, như năng lượng điện toán, lưu trữ và cơ sở dữ liệu, khi cần thiết [13]. Hiện có nhiều nhà cung cấp dịch vụ đám mây như Amazon Web Services (AWS), Google Cloud, Microsoft có Azure. Lợi ích của Cloud Computing mang lại cho người dùng rất nhiều như tiết kiệm chi phí, khả năng mở rộng linh hoạt về quy mô, hiệu năng cao, bảo mật, tốc độ truy xuất dữ liệu, năng suất của một hệ thống [14]. Hiện nay, Cloud cung cấp dịch vụ theo ba mô hình cơ bản: Cơ sở hạ tầng như một dịch vụ (IaaS – Infrastructure as a Service), nền tảng như một dịch vụ (PaaS – Platform as a Service), phần mềm như một dịch vụ (SaaS – Software as a Service) [15]. Việc cung cấp nhiều mô hình sẽ đáp ứng được nhu cầu đa dạng của các doanh nghiệp hiện nay. Ngoài ra, điện toán đám mây còn có mô hình triển khai như public/private cloud.

**CÔNG CỤ TÁI CHẾ THIẾT BỊ ĐIỆN THOẠI DI ĐỘNG**

**3.1 Giới thiệu**

Để triển khai các giải pháp mà chúng tôi đề ra trong phần giới thiệu, đề xuất xây dựng hệ thống web tái chế điện thoại cũ. Hệ thống của chúng tôi dựa trên các công nghệ Spring Boot, html, js. Đồng thời triển khai Hệ thống web tái chế điện thoại cũ. Hệ thống web tái chế điện thoại cũ trên website (Hình 3.1.1)

A diagram of a cloud

Description automatically generated

*Hình 3.1.1 Mô hình Micro-services của Hệ thống tái chế điện thoại cũ*

Spring boot chịu trách nhiệm tạo ra các API, xử lý các request, MySQL dùng lưu trữ dữ liệu của toàn hệ thống như dữ liệu điện thoại cũ, dữ liệu khách hàng.

**3.2 Kiến trúc phần mềm**

Micro-service là một kỹ thuật phát triển phần mềm, một biến thể của kiến trúc hướng dịch vụ (SOA), trong đó cấu trúc một ứng dụng như một tập hợp các dịch vụ được ghép với nhau [3]. Trong kiến trúc micro-service, các dịch vụ được xử lý tốt và các giao thức rất nhẹ. Đó cũng là lý do mà chúng tôi đã đề xuất kiến trúc cho hệ thống tái chế thiết bị điện thoại cũ.

*Vì sao lại chọn kiến trúc:*

Việc nhu cầu tái chế là thiết yếu để môi trường không bị ô nhiễm rác thải điện tử gây ảnh hưởng tới sức khỏe con người, thì yêu cầu 1 hệ thống mạnh mẽ, chất lượng cao, vận hành ổn ngay từ ban đầu mà không cần quan tâm chi phí , thời gian phát triển ko quá gấp gáp thì kiến trúc microservices là một sự lựa chọn hoàn hảo.

Lượng traffic biến động lớn nhưng có thể đoán trước được lại là đặc điểm cần cân nhắc sử dụng kiến trúc này. Vì microservice đặc biệt tối ưu khi áp dụng vào hệ thống cần tăng nhanh server nhưng lại có thể giảm chúng đi một cách đơn giản mà không mất nhiều chi phí. Nếu các kiến trúc monolithic khác khi cần mở rộng phải deploy toàn bộ hệ thống, rất lãng phí tài nguyên. Nhưng đối với kiến trúc microservice thì chỉ cần deploy nhỏ lẻ cho từng service nào cần tăng thôi.

Ngoài ra, nếu công ty sẽ mở thêm các chi nhánh trong tương lại thì micservice cũng xử lý rất tốt điều đó. Có thể triển khai hệ thống tại từng khu vực tăng hiệu suất truy vấn, trong đó dữ liệu sẽ được đồng bộ về máy chủ trung tâm để đảm bảo sự nhất quán dữ liệu. Việc phân thành khu vực cũng có thể hạn chế downtime cho toàn bộ hệ thống.

Downtime luôn là yếu tố then chốt nhất khi đánh giá độ tin cậy của một hệ thống. Đặc điểm của microservice là độc lập vậy nên khi hệ thống cần cập nhật thì chỉ cần ‘hạ’ các service có liên quan xuống thay vì cả hệ thống, các service còn lại vẫn khả dụng để người dùng sử dụng. Cũng với tính độc lập đó giúp hệ thống chịu lỗi tốt hơn, bảo mật hơn. Khi có sự cố ở một service nào đó thì chỉ cần kiểm tra service đó, dữ liệu có bị đánh cắp thì cũng nhỏ lẻ từng service. Vì vậy có thể tối ưu bảo mật tại các service, nghĩa là service nào chứa dữ liệu quan trọng liên quan đến sống còn của công ty thì cần siết chặt bảo mật như thông tin khách hàng, báo cáo tài chính ,… những service còn lại thì có thể giảm bớt bảo mật để tăng hiệu suất và độ trải nghiệm người dùng.

Việc tương lai có thể có thêm các chứng năng khác điều đó là rất dể dàng nếu ta chọn kiến trúc microservice, vì đây là tính chất được đánh giá 5 sao của kiến trúc này. Với cơ chế độ lập, ta có thể phát triển các chức năng mới thành các service và deploy chúng lên mà không ảnh hưởng đến hệ thống đang chạy.

Vì các chức năng được phân chia thành các service độc lập để phát triển. Vậy nên mỗi service sẽ có thể viết bằng các ngôn ngữ khác nhau, có thể áp dụng nhiều công nghệ khác nhau.

**Điểm mạnh:**

* Kiếm trúc micro-services cho phép hệ thống dễ dàng mở rộng và scale để đáp ứng lượng truy cập đồng thời vào những thời điểm cao điểm trong quá trình gửi sản phẩm để tái chế, phân hủy an toàn mỗi năm mà không gặp phải các vấn đề liên quan đến hiệu suất.
* Phân tán các dịch vụ nhỏ lẻ trong kiến trúc Microservices giúp giảm thiểu thời gian downtime để cập nhật và kiểm soát tính nhất quán của dữ liệu. Điều này đảm bảo rằng hệ thoongs liên tục mà không gây ra sự gián đoạn đáng kể.
* Microservices giúp tách biệt các dịch vụ thành các thành phần nhỏ, dễ dàng tích hợp, thuận lợi cho việc sử dụng công nghệ và ngon ngữ lập trình khác nhau cho từng dịch vụ
* Xử lý lỗi, hạn chế tin tặc tấn công hệ thống

**Điểm yếu:**

* Có thể dẫn đến sự phức tạp trong việc quản lý toàn bộ hệ thống. Mỗi Microservices cần được quản lý, theo dõi và bảo trì một cách độc lập, tạo ra một lượng công việc quản lý tăng lên đáng kể.
* Đòi hỏi cơ sở hạ tầng và tài nguyên vận hành phức tạp hơn, do cần triển khai và vận hành nhiều dịch vụ khác nhau. Điều này có thể dẫn đến tăng chi phí vận hành và quản lý
* Vì Kiến trúc này tách biệt và chạy 1 cách độc lập, việc kiểm thử và gỡ lỗi có thể trỏe nên phức tạp hơn so vưới monolithic. Điều này đặc biệt đúng khi cần phải kiểm thử tích hợp giữa các dịch vụ.

**3.3 Phân tích hệ thống công cụ tái chế thiết bị điện thoại di động**

* Để phân tích hệ thống thu mua và tái chế thiết bị điện thoại cũ với các đối tượng người dùng là khách hàng và quản trị viên (admin), chúng tôi sẽ mô tả các chức năng chính của từng đối tượng. Trong đó, chúng tôi tập trung vào các chức năng chính của hệ thống mà hiện nay công ty đang gặp phải vấn đề khó khăn như quá trình thu mua, đánh giá và tái chế thiết bị. Ngoài ra, các chức năng của quản trị viên cũng được đưa vào nhằm hỗ trợ cho hệ thống đồng bộ các nghiệp vụ và các thao tác của khách hàng.
* **Đối tượng Người dùng**

**1. Khách hàng (Customer)**

Khách hàng là những người có thiết bị điện tử cũ (như điện thoại di động iPhone hoặc Galaxy) muốn bán hoặc tái chế. Các chức năng chính của khách hàng bao gồm:

**Yêu cầu báo giá (Quoting):** Khách hàng có thể yêu cầu báo giá cho thiết bị cũ của họ thông qua trang web hoặc ki-ốt.

**Gửi thiết bị (Sending Device):** Nếu hài lòng với báo giá, khách hàng sẽ gửi thiết bị đến công ty tái chế.

**Kiểm tra trạng thái (Checking Item Status):** Khách hàng có thể kiểm tra trạng thái của thiết bị đã gửi bất cứ lúc nào thông qua trang web.

**Thanh toán (Payment):** Nhận tiền thanh toán sau khi thiết bị được đánh giá và xác nhận tình trạng.

**2. Quản trị viên (Admin)**

Quản trị viên là nhân viên của công ty tái chế, chịu trách nhiệm quản lý và điều hành hệ thống. Các chức năng chính của quản trị viên bao gồm:

**Nhận thiết bị (Receiving):** Quản lý việc nhận thiết bị từ khách hàng.

Đánh giá thiết bị (Assessment): Đánh giá tình trạng hoạt động của thiết bị để xác định giá trị thực tế của nó.

**Tái chế thiết bị (Recycling):** Xử lý việc tái chế thiết bị, bao gồm tiêu hủy an toàn hoặc bán lại thiết bị.

**Quản lý thanh toán (Accounting):** Xác nhận và thực hiện thanh toán cho khách hàng dựa trên kết quả đánh giá.

**Báo cáo (Reporting):** Tạo báo cáo tài chính và hoạt động định kỳ và không định kỳ dựa trên hoạt động tái chế.

Danh sách tác nhân và mô tả.

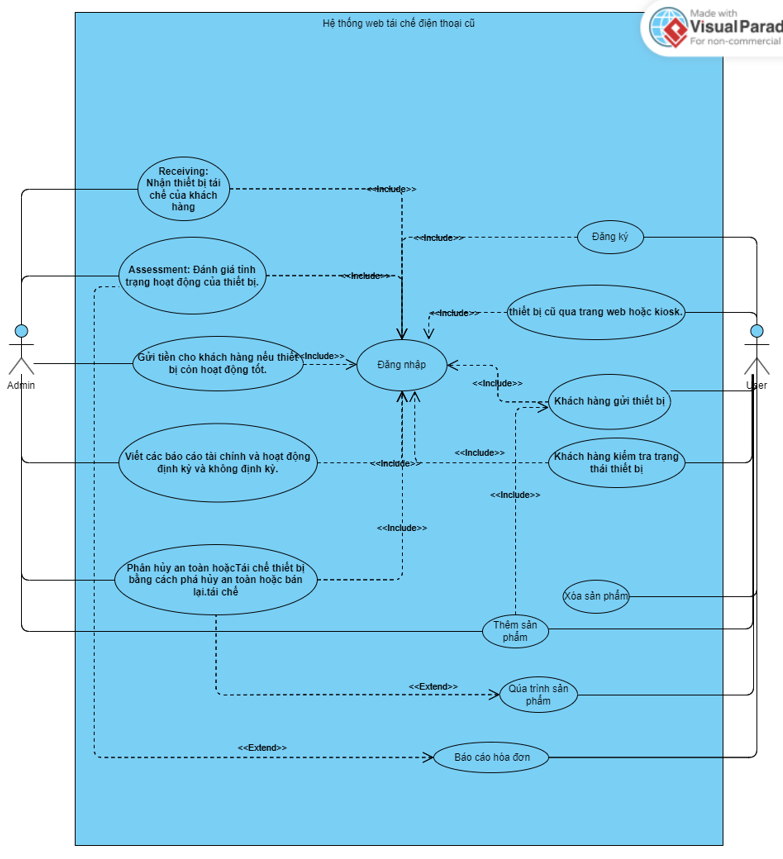
|  |  |
| --- | --- |
| **Tác nhân** | **Mô tả tác nhân** |
| Admin | Người quản lý hệ thống và cung cấp hỗ trợ kỹ thuật. |
| Khách hàng | Người sử dụng hệ thống để bán và giao dịch thu mua tái chế điện thoại |

Bảng 3-3 Danh sách tác nhân và mô tả

|  |  |
| --- | --- |
| **ID** | **Tên Use case** |
| UC01 | Đăng ký |
| UC02 | Đăng nhập |
| UC03 | Xem thông tin chi tiết |
| UC04 | Theo dõi trạng thái và tiến độ tái chế |
| UC05 | Tiếp nhận yêu cầu tái chế |
| UC06 | Xem danh sách KH |
| UC07 | Tạo báo cáo |

Bảng 3-4 Danh sách các tình huống trong hệ thống

**Use case tổng quát:**

****

**3.1. UC001\_Đăng ký:**

**3.1.1. Mô tả UC001**

|  |  |  |
| --- | --- | --- |
| **Use case: Đăng ký** | | |
| Mục đích: | Giúp Khách hàng có thể tạo tài khoản | |
| Mô tả: | Use case mô tả các bước đăng ký tài khoản | |
| Tác nhân: | Khách hàng | |
| Điềukiện trước: | Không có | |
| Điều kiện sau: | Nếu đăng ký thành công thì thông tin tài khoản sẽ được lưu trữ và chuyển sang giao diện đăng nhập. | |
| Luồng sự kiện chính (Basic flows): | | |
| Actor | | System |
| 1.Khách hàng vô trang web | |  |
|  | | 2.Hệ thống hiện giao diện đăng ký. |
| 3.Khách hàng nhập thông tin cá nhân | |  |
|  | | 4.Hệ thống kiểm tra thông tin khách hàng nhập |
|  | | 5.Hệ thống thông báo đăng ký thành công |
|  | | 6.Hệ thống chuyển qua giao diện đăng nhập |
| Luồng sự kiện phụ (Alternative Flows): | | |
|  | | 4.1 Hệ thống thông báo nhập thiếu hoặc sai thông tin |
|  | | 4.2.Hệ thống quay lại bước 2 |

**3.1.2. Sơ đồ Activity:**

**A diagram of a flowchart

Description automatically generated**

**3.2. UC002\_Đăng nhập:**

**3.2.1. Mô tả UC002:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use case: Đăng nhập** | | | |
| Mục đích: | Giúp khách hàng, nhân viên đăng nhập vào web | | |
| Mô tả: | Use case mô tả các bước đăng nhập | | |
| Tác nhân: | Khách hàng, Nhân viên | | |
| Điều kiện trước: | Có tài khoản đăng ký | | |
| Điều kiện sau: | Đăng nhập thành công và hệ thống web tái chế điện thoại | | |
| Luồng sự kiện chính (Basic flows): | | | |
| Actor | | System | |
| 1. Khách hàng, nhân viên vào hệ thống web | |  | |
|  | | 2.Hệ thống hiện thị giao diện đăng nhập | |
| 3.Khách hàng, nhân viên đăng nhập bao gồm:số điện thoại, mật khẩu | |  | |
|  | | 4.Hệ thống kiểm tra thông tin đăng nhập | |
|  | | 5.Hệ thống thông báo đăng nhập thành công | |
|  | | 6.Hệ thống hiển thị giao diện chính | |
| Luồng sự kiện phụ (Alternative Flows): | | | |
|  | | | 4.1 Hệ thống thông báo “Thông tin đăng nhập không hợp lệ” |
|  | | | 4.2 Hệ thống quay lại bước 2 |

**3.2.2 Sơ đồ Activity:**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a diagram

Description automatically generated**

**3.4.2 Sơ đồ Activity:**

**A diagram of a diagram

Description automatically generated**

**3.5 UC005\_Xem chi tiết:**

**3.5.3. Mô tả UC005:**

|  |  |  |
| --- | --- | --- |
| **Use case: Xem chi tiết** | | |
| Mục đích: | Giúp khách hàng xem được thông tin và giá tiền chi tiết của điện thoại | |
| Mô tả: | Use case mô tả các bước xem chi tiết | |
| Tác nhân: | Khách hàng | |
| Điềukiện trước: | Phải đăng ký tái chế thành công và gửi điện thoại. | |
| Điều kiện sau: | Hệ thống hiển thi chi tiết đơn hàng và giá tiền | |
| Luồng sự kiện chính (Basic flows): | | |
| Actor | | System |
| 1. Khách hàng chọn chức năng thông tin định giá | |  |
|  | | 2.Hệ thống hiện thị giao diện thông tin định giá |
| 3.Khách hàng chọn xem chi tiết | |  |
|  | | 4.Hệ thống hiển thị thông tin chi tiết |

**3.5.2 Sơ đồ Activity:**

**A diagram of a system

Description automatically generated**

**3.6 UC005\_Theo dõi trạng thái và tiến độ tái chế**

**3.6.1 Mô tả UC006**

|  |  |  |
| --- | --- | --- |
| **Use case: Theo dõi trạng thái và tiến độ tái chế** | | |
| Mục đích: | Giúp khách hàng theo dõi tình trạng của điện thoại và tiến độ tái chế | |
| Mô tả: | Use case mô tả các bước theo dõi tình trạng của điện thoại trong quá trình tái chế | |
| Tác nhân: | Khách hàng | |
| Điềukiện trước: | Phải gửi điện thoại thành công. | |
| Điều kiện sau: | Hệ thống báo “Thành công” | |
| Luồng sự kiện chính (Basic flows): | | |
| Actor | | System |
| 1. Khách hàng chọn chức năng theo dõi trạng thái và tiến độ tái chế | |  |
|  | | 2.Hệ thống hiện thị giao diện theo dõi trạng thái và tiến độ tái chế |

**3.6.2 Sơ đồ Activity:**

**A diagram of a diagram

Description automatically generated**

**3.7 UC006\_Tiếp nhận yêu cầu tái chế**

**3.7.1 Mô tả UC006**

|  |  |  |
| --- | --- | --- |
| **Use case: Tiếp nhận yêu cầu tái chế** | | |
| Mục đích: | Giúp nhân viên Tiếp nhận yêu cầu tái chế | |
| Mô tả: | Use case mô tả các bước Tiếp nhận yêu cầu tái chế | |
| Tác nhân: | Nhân viên | |
| Điềukiện trước: | Phải gửi điện thoại thành công. | |
| Điều kiện sau: | Hệ thống báo “Thành công” | |
| Luồng sự kiện chính (Basic flows): | | |
| Actor | | System |
| 1.Nhân viên chọn chức năng Tiếp nhận yêu cầu tái chế | |  |
|  | | 2.Hệ thống hiện thị giao diện Tiếp nhận yêu cầu tái chế |
| 3.Nhân viên xác nhận yêu cầu | |  |
|  | | 4.Hệ thống thông báo “Xác nhận thành công” |

**3.7.2 Sơ đồ activity:**

**A screenshot of a computer

Description automatically generated**

**3.8 UC008\_Xem danh sách báo cáo tái chế**

**3.8.1 Mô tả UC008**

|  |  |  |
| --- | --- | --- |
| **Use case: Xem danh sách báo cáo tái chế** | | |
| Mục đích: | Giúp quản lý xem được danh sách báo cáo về quá trình tái chế điện thoại | |
| Mô tả: | Use case mô tả các bước xem danh sách | |
| Tác nhân: | Quản lý | |
| Điềukiện trước: | Phải đăng nhập thành công. | |
| Điều kiện sau: | Hệ thống hiển thị danh sách báo cáo | |
| Luồng sự kiện chính (Basic flows): | | |
| Actor | | System |
| 1. Quản lý chọn chức năng xem danh sách báo cáo tái chế | |  |
|  | | 2.Hệ thống hiện thị giao diện danh sách báo cáo |

**3.8.2 Sơ đồ Activity**

**A diagram of a diagram

Description automatically generated with medium confidence**

**3.9 UC009\_Xem chi tiết báo cáo tái chế**

**3.9.1 Mô tả UC009**

|  |  |  |
| --- | --- | --- |
| **Use case: Xem chi tiết báo cáo tái chế** | | |
| Mục đích: | Giúp quản lý xem được chi tiết báo cáo về quá trình tái chế điện thoại | |
| Mô tả: | Use case mô tả các bước xem chi tiết báo cáo | |
| Tác nhân: | Quản lý | |
| Điềukiện trước: | Phải đăng nhập thành công. | |
| Điều kiện sau: | Hệ thống hiển thị chi tiết báo cáo | |
| Luồng sự kiện chính (Basic flows): | | |
| Actor | | System |
| 1. Quản lý chọn chức năng xem danh sách báo cáo tái chế | |  |
|  | | 2.Hệ thống hiện thị giao diện danh sách báo cáo |
| 3.Quản lý chọn báo cáo cần xem chi tiết | |  |
|  | | 4.Hệ thống hiển thị chi tiết báo cáo |

**3.9.2 Sơ đồ Activity**

**A diagram with blue text and black dots

Description automatically generated with medium confidence**

**A diagram of a system

Description automatically generated**

**3.12 UC0012\_Xóa báo cáo tái chế**

**3.12.1 Mô tả UC0012**

|  |  |  |
| --- | --- | --- |
| **Use case: Xóa báo cáo về quá trình tái chế** | | |
| Mục đích: | Giúp quản lý xóa được báo cáo về quá trình tái chế | |
| Mô tả: | Use case mô tả các bước xóa báo về quá trình tái chế | |
| Tác nhân: | Quản lý | |
| Điềukiện trước: | Phải đăng nhập thành công. | |
| Điều kiện sau: | Hệ thống báo “Xóa thành công” | |
| Luồng sự kiện chính (Basic flows): | | |
| Actor | | System |
| 1. Quản lý chọn chức năng xem danh sách báo cáo tái chế | |  |
|  | | 2.Hệ thống hiện thị giao diện danh sách về quá trình tái chế |
| 3. Quản lý chọn báo cáo cần xóa | |  |
|  | | 4.Hệ thống hiển thị chi tiết báo cáo |
| 5.Quản lý bấm nút xóa | |  |
|  | | 6.Hệ thống gửi xác nhận “Bạn chắc chắn muốn xóa” |
| 7. Quản lý bấm nút “đồng ý” | |  |
|  | | 8. Hệ thống xóa báo cáo và lưu lại ở cơ sở dữ liệu |
|  | | 9. Hệ thống thông báo “ Xóa thanh công!” |
| Luồng sự kiện phụ (Alternative Flows): | | |
| 7.1 Quản lý bấm nút “Hủy” | |  |
|  | | 7.2 Hệ thống quay về giao diện danh sách báo cáo |

**3.12.2 Sơ đồ Activity**

**A diagram of a diagram

Description automatically generated with medium confidence**

**3.19 UC0019\_Xem danh sách khách hàng**

**3.19.1 Mô tả UC0019**

|  |  |  |
| --- | --- | --- |
| **Use case: Xem danh sách khách hàng** | | |
| Mục đích: | Giúp quản lý xem được danh sách các khách hàng | |
| Mô tả: | Use case mô tả các bước xem danh sách khách hàng | |
| Tác nhân: | Quản lý | |
| Điềukiện trước: | Phải đăng nhập thành công. | |
| Điều kiện sau: | Hệ thống hiển thị danh sách khách hàng | |
| Luồng sự kiện chính (Basic flows): | | |
| Actor | | System |
| 1. Quản lý chọn danh sách khách hàng | |  |
|  | | 2.Hệ thống hiện thị giao diện danh sách khách hàng |

**3.19.2 Sơ đồ Activity**

**A diagram of a diagram

Description automatically generated**

**3.20 UC0020\_Xem chi tiết khách hàng**

**3.20.1 Mô tả UC0020**

|  |  |  |
| --- | --- | --- |
| **Use case: Xem chi tiết khách hàng** | | |
| Mục đích: | Giúp quản lý xem được chi tiết thông tin khách hàng | |
| Mô tả: | Use case mô tả các bước xem chi tiết thông tin khách hàng | |
| Tác nhân: | Quản lý | |
| Điềukiện trước: | Phải đăng nhập thành công. | |
| Điều kiện sau: | Hệ thống hiển thị thông tin khách hàng | |
| Luồng sự kiện chính (Basic flows): | | |
| Actor | | System |
| 1. Quản lý chọn danh sách khách hàng | |  |
|  | | 2.Hệ thống hiện thị giao diện danh sách khách hàng |
| 3.Quản lý chọn khách hàng muốn xem | |  |
|  | | 4.Hệ thống hiện thị giao diện thông tin khách hàng |

**3.20.2 Sơ đồ Activity**

**A screenshot of a computer

Description automatically generated**

**3.21 UC0021\_Kiểm tra tình trạng điện thoại**

**3.21.1 Mô tả UC0021**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use case: Kiểm tra tình trạng điện thoại** | | | |
| Mục đích: | Giúp nhân viên cập nhật tình trạng, thông tin điện thoại lên hệ thông | | |
| Mô tả: | Use case mô tả các bước kiểm tra tình trạng điện thoại | | |
| Tác nhân: | Nhân viên | | |
| Điềukiện trước: | Phải đăng nhập thành công | | |
| Điều kiện sau: | Hệ thống thông báo kiểm tra thành công | | |
| Luồng sự kiện chính (Basic flows): | | | |
| Actor | | System | |
| 1. Nhân viên chọn chức năng kiểm tra tình trạng điện thoại | |  | |
|  | | 2.Hệ thống hiện thị giao diện kết quả kiểm tra | |
| 3. Nhân viên nhập kết quả kiểm tra, thông tin điện thoại và số tiền | |  | |
| 4. Nhân viên bấm nút lưu | |  | |
|  | | 5.Hệ thống lưu lại thông tin vào cơ sở dữ liệu | |
|  | | 6.Hệ thống thông báo “Lưu thành công” | |
| Luồng sự kiện phụ (Alternative Flows): | | | |
| 4.1 Nhân viên bấm nút “ Hủy” | | |  |
|  | | | 4.2 Hệ thống quay về bước 2 |

**3.21.2 Sơ đồ Activity**

**A screenshot of a computer

Description automatically generated**

**3.22 UC0022\_Xử lý thanh toán**

**3.24 UC0024\_Báo cáo trạng thái của điện thoại và tiến độ tái chế**

**3.24.1 Mô tả UC0024**

|  |  |  |
| --- | --- | --- |
| **Use case: Báo cáo trạng thái của điện thoại và tiến độ tái chế** | | |
| Mục đích: | Giúp nhân viên xem các báo cáo trạng thái của điện thoại và tiến độ tái chế mà nhân viên đã tạo | |
| Mô tả: | Use case mô tả các bước xem báo cáo về trạng thái của điện thoại và tiến độ tái chế mà nhân viên đã tạo | |
| Tác nhân: | Nhân viên | |
| Điềukiện trước: | Phải đăng nhập thành công | |
| Điều kiện sau: | Hệ thống hiển thị báo cáo | |
| Luồng sự kiện chính (Basic flows): | | |
| Actor | | System |
| 1. Nhân viên chọn chức năng báo cáo trạng thái của điện thoại và tiến độ tái chế | |  |
|  | | 2.Hệ thống hiển thị danh sách báo cáo đã tạo |
| 3.Nhân viên chọn báo cáo muốn xem | |  |
|  | | 4.Hệ thống hiện thị chi tiết báo cáo |

**3.24.2 Sơ đồ Activity**

**A diagram with blue text

Description automatically generated**

**3.25 UC0025\_Xóa báo cáo tái chế**

**3.25.1 Mô tả UC0025**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use case: Xóa báo cáo** | | | |
| Mục đích: | Giúp nhân viên xóa báo cáo | | |
| Mô tả: | Use case mô tả các xóa báo cáo | | |
| Tác nhân: | Nhân viên | | |
| Điềukiện trước: | Phải đăng nhập thành công | | |
| Điều kiện sau: | Hệ thống thông báo “Xóa thành công” | | |
| Luồng sự kiện chính (Basic flows): | | | |
| Actor | | System | |
| 1. Nhân viên chọn chức năng báo cáo trạng thái của điện thoại và tiến độ tái chế | |  | |
|  | | 2. Hệ thống hiển thị danh sách báo cáo đã tạo | |
| 3. Nhân viên chọn báo cáo muốn xóa | |  | |
|  | | 4. Hệ thống hiện thị giao diện báo cáo | |
| 5. Nhân viên chọn xóa | |  | |
|  | | 6.Hệ thống thông báo”Chắc chắn xóa” | |
| 7.Nhân viên bấm”Đồng ý” | |  | |
|  | | 8.Hệ thống xóa và cập nhật lại ở cở sở dữ liệu | |
|  | | 9.Hệ thống thông báo”Xóa thành công” | |
| Luồng sự kiện phụ (Alternative Flows): | | | |
| 7.1 Nhân viên bấm nút “ Hủy” | | |  |
|  | | | 7.2 Hệ thống quay về bước 4 |

**3.25.2 Sơ đồ Activity**

**A screenshot of a computer

Description automatically generated**

**3.26 UC0026\_Tạo báo cáo**

**3.26.1 Mô tả UC0026**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use case: Tạo báo cáo** | | | |
| Mục đích: | Giúp nhân viên tạo báo cáo trạng thái của điện thoại và tiến độ tái chế | | |
| Mô tả: | Use case mô tả các tạo báo cáo | | |
| Tác nhân: | Nhân viên | | |
| Điềukiện trước: | Phải đăng nhập thành công | | |
| Điều kiện sau: | Hệ thống thông báo “Tạo thành công” | | |
| Luồng sự kiện chính (Basic flows): | | | |
| Actor | | System | |
| 1.Nhân viên chọn chức năng báo cáo trạng thái của điện thoại và tiến độ tái chế | |  | |
|  | | 2.Hệ thống hiện thị giao diện danh sách báo cáo | |
| 3.chọn chức năng tạo báo cáo | |  | |
|  | | 4.Hệ thống hiển thị giao diện tạo báo cáo | |
| 5.Nhân viên nhập trạng thái, tiến độ tái chế | |  | |
| 6. chọn nút lưu | |  | |
|  | | 7.Hệ thống lưu thông tin vào cơ sở dữ liệu | |
|  | | 8.Hệ thống thông báo “Tạo báo cáo thành công” | |
| Luồng sự kiện phụ (Alternative Flows): | | | |
| 6.1 Nhân viên bấm nút “ Hủy” | | |  |
|  | | | 6.2 Hệ thống quay về bước 4 |

**3.26.2 Sơ đồ Activity**

**A screenshot of a computer

Description automatically generated**

**3.27 UC0027\_Sửa báo cáo tái chế**

**3.27.1 Mô tả UC0027**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use case: Xóa báo cáo** | | | |
| Mục đích: | Giúp nhân viên sửa báo cáo | | |
| Mô tả: | Use case mô tả các bước sửa báo cáo | | |
| Tác nhân: | Nhân viên | | |
| Điềukiện trước: | Phải đăng nhập thành công | | |
| Điều kiện sau: | Hệ thống thông báo “Sửa thành công” | | |
| Luồng sự kiện chính (Basic flows): | | | |
| Actor | | | System |
| 1. Nhân viên chọn chọn chức năng báo cáo trạng thái của điện thoại và tiến độ tái chế | | |  |
|  | | | 2. Hệ thống hiện thị giao diện danh sách báo cáo |
| 3.Nhân viên chọn báo cáo cần sửa | | |  |
|  | | | 4.Hệ thống hiển thị giao diện thông tin chi tiết báo cáo |
| 5.chọn nút sửa | | |  |
| 6. Nhân vân sửa thông tin báo cáo cần sửa và bấm lưu | | |  |
|  | | | 7.Hệ thống lưu thông tin mới vào cơ sở dữ liệu |
|  | | | 8 s.Hệ thống thông báo “Sửa thành công” |
| Luồng sự kiện phụ (Alternative Flows): | | | |
| 5.1 Nhân viên bấm “Hủy” | |  | |
|  | | 5.2. Hệ thống quay lại nước 4 | |

**3.27.2 Sơ đồ Activity**

**A screenshot of a diagram

Description automatically generated**

**3.4 Class diagram:**

Tại sao chúng tôi chọn lưu trữ dữ liệu bằng MySQL

Quy mô Dự án: Dự án sửa chữa điện thoại là một ứng dụng web nhỏ đến trung bình về quy mô và có khối lượng dữ liệu không quá lớn, và có thể xử lý một lượng dữ liệu cơ bản một cách hiệu quả.

Yêu cầu Dữ liệu: MySQL phù hợp cho các ứng dụng có cấu trúc dữ liệu rõ ràng và liên quan đến các hoạt động giao dịch cơ bản, như lưu trữ thông tin về khách hàng, sản phẩm, đơn hàng và các giao dịch sửa chữa.

Tính ổn định và Hiệu suất: MySQL thường được biết đến với tính ổn định và hiệu suất tốt. Điều này làm cho nó trở thành một lựa chọn phù hợp cho các ứng dụng yêu cầu độ tin cậy cao và thời gian hoạt động lâu dài, như các hệ thống quản lý sửa chữa điện thoại có thể hoạt động 24/7.

Phổ biến và Hỗ trợ Cộng đồng tốt: MySQL là một hệ thống quản lý cơ sở dữ liệu mở mã nguồn phổ biến, có một cộng đồng lớn và tích cực hỗ trợ từ cộng đồng người dùng. Điều này có nghĩa là bạn có thể tìm thấy nhiều tài liệu, tài nguyên và hỗ trợ trực tuyến để giải quyết vấn đề khi gặp phải.

Chi phí: MySQL là một hệ thống quản lý cơ sở dữ liệu mã nguồn mở, do đó thường miễn phí hoặc có chi phí thấp hơn so với các hệ thống thương mại. Điều này có thể giúp giảm thiểu chi phí cho dự án, đặc biệt là đối với các dự án có ngân sách hạn chế.

**3.5 Giao diện đồ họa**

Giao diện đăng nhập:

**A screenshot of a computer

Description automatically generated**

Giao diện đăng ký:

**A screenshot of a computer

Description automatically generated**

Giao diện thêm sản phẩm để tái chế an toàn hoặc phân hủy, tiến trình, mô tả sản phẩm:

**A screenshot of a computer

Description automatically generated**

Giao diện cập nhật sản phẩm:

**A screenshot of a computer

Description automatically generated**

Giao diện xem chi tiết sản phẩm:

**A screenshot of a computer

Description automatically generated**

**Giao diện thanh toán**

**A screenshot of a computer

Description automatically generated**

**Giao diện thanh toán thành công**

**A group of people in a white background

Description automatically generated with medium confidence**

**Giao diện thanh toán bị hủy**

**A white background with blue text

Description automatically generated**

**Chương 4 Kết luận**

**4.1 Kết quả đạt được**

**4.2 Hạn chế**

* Hạn chế về thời gian, không đủ để thực hiện nhiều chức năng
* Dự án của chúng tôi hiện vẫn còn nhiều khía cạnh cần được cải thiện để đạt được hiệu quả cao nhất.
* Kiến trúc microservice chưa được triển khai một cách tối ưu. Điều này chủ yếu bắt nguồn từ nhóm em vẫn đang là sinh viên, chưa từng có kinh nghiệm nào liên quan đến kiến trúc này trước đây. Dẫn đến việc chúng tôi chưa thể khai thác hết tiềm năng và lợi ích mà mô hình microservice mang lại.
* Hiện tại, chúng tôi vẫn chưa hoàn thành được service dành cho cựu sinh viên, vẫn còn thiếu một vài các api nhỏ lẻ có thể sẽ cần khi frontend triển khai.

*Tiếp theo là phần front-end:*

* Các tính năng quan trọng như: đăng nhập, đăng ký, thêm sản phẩm, sửa , xóa, thanh toán.
* Hiện tại, giao diện người dùng chưa thực sự thân thiện và chưa đáp ứng đầy đủ các yêu cầu chức năng của dự án. Các tính năng hiện có vẫn đang trong quá trình phát triển và chưa đạt được mức độ hoàn chỉnh như mong muốn, điều này có thể gây ra khó khăn cho người dùng khi thao tác.

4**.3 Hướng phát triển**

- Để hệ thống thu mua và tái chế thiết bị điện thoại cũ hoạt động hiệu quả hơn và đáp ứng nhu cầu ngày càng tăng của khách hàng, chúng em cần xác định các hướng phát triển chính. Những hướng phát triển này không chỉ giúp cải thiện hiệu suất hệ thống mà còn nâng cao trải nghiệm người dùng và tối ưu hóa quy trình nghiệp vụ.

1. Tối Ưu Hóa Quy Trình Đăng Ký và Nhận Thiết Bị

* Tự động hóa quy trình báo giá: Sử dụng công nghệ trí tuệ nhân tạo (AI) để tự động đánh giá tình trạng thiết bị dựa trên hình ảnh và thông tin khách hàng cung cấp. Điều này sẽ giúp tăng tốc độ và độ chính xác của báo giá.
* Cải thiện giao diện người dùng: Thiết kế lại giao diện trang web và ki-ốt để người dùng dễ dàng thao tác khi yêu cầu báo giá và kiểm tra trạng thái thiết bị.
* Ứng dụng di động: Phát triển ứng dụng di động để khách hàng có thể yêu cầu báo giá và theo dõi trạng thái thiết bị một cách thuận tiện hơn.

2. Nâng Cao Khả Năng Đánh Giá và Tái Chế

* Công nghệ IoT (Internet of Things): Sử dụng các cảm biến IoT để kiểm tra tình trạng thiết bị một cách chính xác hơn. Cảm biến có thể phát hiện các lỗi phần cứng và phần mềm một cách nhanh chóng.
* Phân loại thông minh: Áp dụng công nghệ machine learning để phân loại thiết bị dựa trên tình trạng và khả năng tái sử dụng hoặc tái chế, giúp tối ưu hóa quá trình xử lý.

3. Cải Thiện Hệ Thống Thanh Toán và Kế Toán

* Tích hợp các cổng thanh toán: Kết nối với nhiều cổng thanh toán khác nhau (như ví điện tử) để khách hàng có nhiều lựa chọn khi nhận tiền thanh toán.
* Hệ thống kế toán tự động: Sử dụng phần mềm kế toán tự động để quản lý các giao dịch tài chính, giảm thiểu sai sót và tăng cường tính minh bạch.

4. Nâng Cao Khả Năng Báo Cáo và Phân Tích

* Phân tích dữ liệu lớn (Big Data): Áp dụng công nghệ big data để phân tích dữ liệu từ các giao dịch, giúp công ty hiểu rõ hơn về xu hướng thị trường và nhu cầu của khách hàng.
* Báo cáo động (Dynamic Reporting): Phát triển hệ thống báo cáo động cho phép quản trị viên tạo các báo cáo tùy chỉnh theo nhu cầu cụ thể.

5. Phát Triển Tính Năng Thân Thiện Với Môi Trường

* Giải pháp tái chế xanh: Nghiên cứu và áp dụng các công nghệ tái chế thân thiện với môi trường, giảm thiểu tác động tiêu cực đến môi trường.
* Chương trình khuyến khích tái chế: Tạo ra các chương trình khuyến khích khách hàng tái chế thiết bị cũ bằng cách cung cấp ưu đãi hoặc phần thưởng.

6. Tăng Cường Bảo Mật và Bảo Vệ Dữ Liệu

* Bảo mật thông tin khách hàng: Áp dụng các biện pháp bảo mật tiên tiến để bảo vệ thông tin cá nhân của khách hàng, đảm bảo tuân thủ các quy định về bảo vệ dữ liệu (như GDPR).
* Kiểm soát truy cập: Phát triển hệ thống kiểm soát truy cập để đảm bảo chỉ những nhân viên có thẩm quyền mới có thể truy cập vào các thông tin nhạy cảm.

7. Mở Rộng Quy Mô và Tương Tác Quốc Tế

* Mở rộng thị trường: Xem xét mở rộng hoạt động ra các thị trường quốc tế, tạo điều kiện cho khách hàng ở nhiều quốc gia có thể sử dụng dịch vụ.
* Hợp tác với các đối tác quốc tế: Thiết lập quan hệ hợp tác với các công ty tái chế và nhà sản xuất thiết bị điện tử trên toàn cầu để tối ưu hóa quy trình thu mua và tái chế.

# TÀI LIỆU THAM KHẢO

**Các tài liệu từ Internet**

1. Spring. Spring Document Guide [Online]. Từ <https://spring.io/quickstart>
2. Mongodb. Mongodb Document Guide [Online]. Từ [https://www.mongodb.com/docs/manual/core/document](https://www.mongodb.com/docs/manual/core/document/)
3. W3Schools. W3Schools Online Web Tutorials. Từ <https://www.w3schools.com/>