**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP.HCM**



ỨNG DỤNG TÁI CHẾ ĐIỆN THOẠI CŨ

***Nhóm 03 - Sinh viên thực hiện***

1. Lê Trọng Nghĩa – 20095041
2. Mai Thị Thu Thúy – 20107041
3. Lê Văn Ngọc – 20022421

# MỤC LỤC

[MỤC LỤC 1](#_Toc14693502)

[DANH MỤC CÁC HÌNH VẼ 1](#_Toc14693503)

[DANH MỤC CÁC BẢNG BIỂU 1](#_Toc14693504)

[CHƯƠNG 1 : GIỚI THIỆU 1](#_Toc14693505)

[1.1 Tổng quan 1](#_Toc14693506)

[1.2 Mục tiêu đề tài 1](#_Toc14693507)

[1.3 Phạm vi đề tài 1](#_Toc14693508)

[1.4 Mô tả yêu cầu chức năng 1](#_Toc14693509)

[CHƯƠNG 2 : CƠ SỞ LÝ THUYẾT 1](#_Toc14693510)

[2.1 Cloud Computing với AWS 1](#_Toc14693511)

[2.2 Node.js 1](#_Toc14693512)

[2.3 MongoDB / DynamoDB 1](#_Toc14693513)

[CHƯƠNG 3 : PHÂN TÍCH VÀ THIẾT KẾ 1](#_Toc14693514)

[3.1 Phân tích yêu cầu bằng UML 1](#_Toc14693515)

[3.1.1 Usecase tổng quát 1](#_Toc14693516)

[3.1.2 Danh sách tác nhân và mô tả 1](#_Toc14693517)

[3.1.3 Danh sách các tình huống hoạt động (Use cases) 1](#_Toc14693518)

[3.1.4 Tình huống hoạt động 1](#_Toc14693519)

[3.2 Class diagram 1](#_Toc14693520)

[3.3 Deployment diagram 1](#_Toc14693521)

[CHƯƠNG 4 : HIỆN THỰC 1](#_Toc14693522)

[4.1 Cấu hình phần cứng, phần mềm 1](#_Toc14693523)

[4.2 Giao diện của hệ thống 1](#_Toc14693524)

[CHƯƠNG 5 : KẾT LUẬN 1](#_Toc14693525)

[5.1 Kết quả đạt được 1](#_Toc14693526)

[5.2 Hạn chế của đồ án 1](#_Toc14693527)

[5.3 Hướng phát triển 1](#_Toc14693528)

[TÀI LIỆU THAM KHẢO 1](#_Toc14693529)

1. **Giới thiệu**

Hệ thống tái chế thiết bị điện tử “Gen Z” là một hệ thống dùng để tái chế các thiết bị di động cũ của hãng Apple của khách hàng. Các quy trình xử lý tái chế các thiết bị điện tử cũ hoạt động như sau:

* Khách hàng lên website tham khảo thông tin thiết bị mà họ muốn gửi để tái chế. Hệ thống sẽ trả về số tiền tương ứng với các thông số mà khách hàng đã cung cấp (**trích dẫn**) (chọn điện thoại và thông số tương ứng)
  + Các tiêu chí đánh giá thiết bị mà hệ thống sử dụng:
    - Tên điện thoại
    - Màu điện thoại
    - Phiên bản điện thoại (thường, Pro, Pro Max)
    - RAM (GB)
      * 64GB
      * 128GB
      * 256GB
      * 512GB
      * 1TB
    - Năm sử dụng:
      * Dưới 3 tháng: trừ 15% giá mua thiết bị
      * Từ 3 đến 6 tháng: trừ 25% giá mua thiết bị
      * Từ 6 tháng đến 12 tháng: trừ 35% giá mua thiết bị
      * Từ 12 đến 18 tháng: trừ 40% giá mua thiết bị
      * Trên 18 tháng: trừ 45% giá mua thiết bị
    - Màn hình
      * Chưa thay màn hình:
        + Sọc màn hình:

Sọc ít: trừ **15**% giá sau khi đã xét năm sử dụng

Sọc nhiều: trừ **25**% giá sau khi đã xét năm sử dụng

* + - * + Độ cảm ứng:

Cảm ứng tốt:

Cảm ứng không tốt: trừ **15**% giá sau khi đã xét năm sử dụng

* + - * + Vỡ màn hình

Vỡ ít: trừ **30**% giá sau khi đã xét năm sử dụng

Vỡ nhiều: trừ **40**% giá sau khi đã xét năm sử dụng

* + - * + Trầy xước:

Trầy xước ít: trừ **10**% giá sau khi đã xét năm sử dụng

Trầy xước nhiều: trừ **15**% giá sau khi đã xét năm sử dụng

* + - * Đã thay màn hình: trừ **20%** giá
        + Sọc màn hình:

Sọc ít: trừ **10**% giá sau khi đã xét năm sử dụng

Sọc nhiều: trừ **15**% giá sau khi đã xét năm sử dụng

* + - * + Độ cảm ứng:

Cảm ứng tốt:

Cảm ứng không tốt: trừ **12**% giá sau khi đã xét năm sử dụng

* + - * + Vỡ màn hình

Vỡ ít: trừ **10**% giá sau khi đã xét năm sử dụng

Vỡ nhiều: trừ **15**% giá sau khi đã xét năm sử dụng

* + - * + Trầy xước:

Trầy xước ít: trừ **3**% giá sau khi đã xét năm sử dụng

Trầy xước nhiều: trừ **5**% giá sau khi đã xét năm sử dụng

* + - Vỏ điện thoại:
      * Trầy:
        + Trầy ít: trừ **10**%
        + Trầy nhiều: trừ **20**%
      * Nứt vỡ: trừ **35**%
    - Lưng điện thoại:
      * Trầy:
        + Trầy ít: trừ **5**%
        + Trầy nhiều: trừ **10**%
      * Nứt vỡ: trừ **15**%
      * Cảm ứng: (tùy chọn)
    - Pin (cũ hay mới, % mA, đã thay hay chưa)
      * Pin chưa thay:
        + Sức chứa tối đa (Maximum Capacity):

Từ 80% trở lên: trừ **5%**

Dưới 80%: trừ **20%**

* + - * Pin đã thay: trừ **50%** giá mua thiết bị sau khi đánh giá
    - Nút bấm (có mất hay hư ko)
      * Nút không bấm được: trừ **5%** mỗi nút
    - Loa (có nghe rõ ko, có hư ko) (tùy chọn)
      * Nghe rõ:
      * Nghe không rõ: trừ 10%
    - Camera (có nét ko, có hư ko)
      * Camera trước:
        + Trầy xước

Trầy ít: trừ **3%**

Trầy nhiều: trừ **5%**

* + - * + Độ sắc nét (tùy chọn)

Nét ít

* + - * Camera sau:
        + Vỡ: trừ **10%** mỗi camera
        + Trầy xước

Trầy ít: trừ **5%**

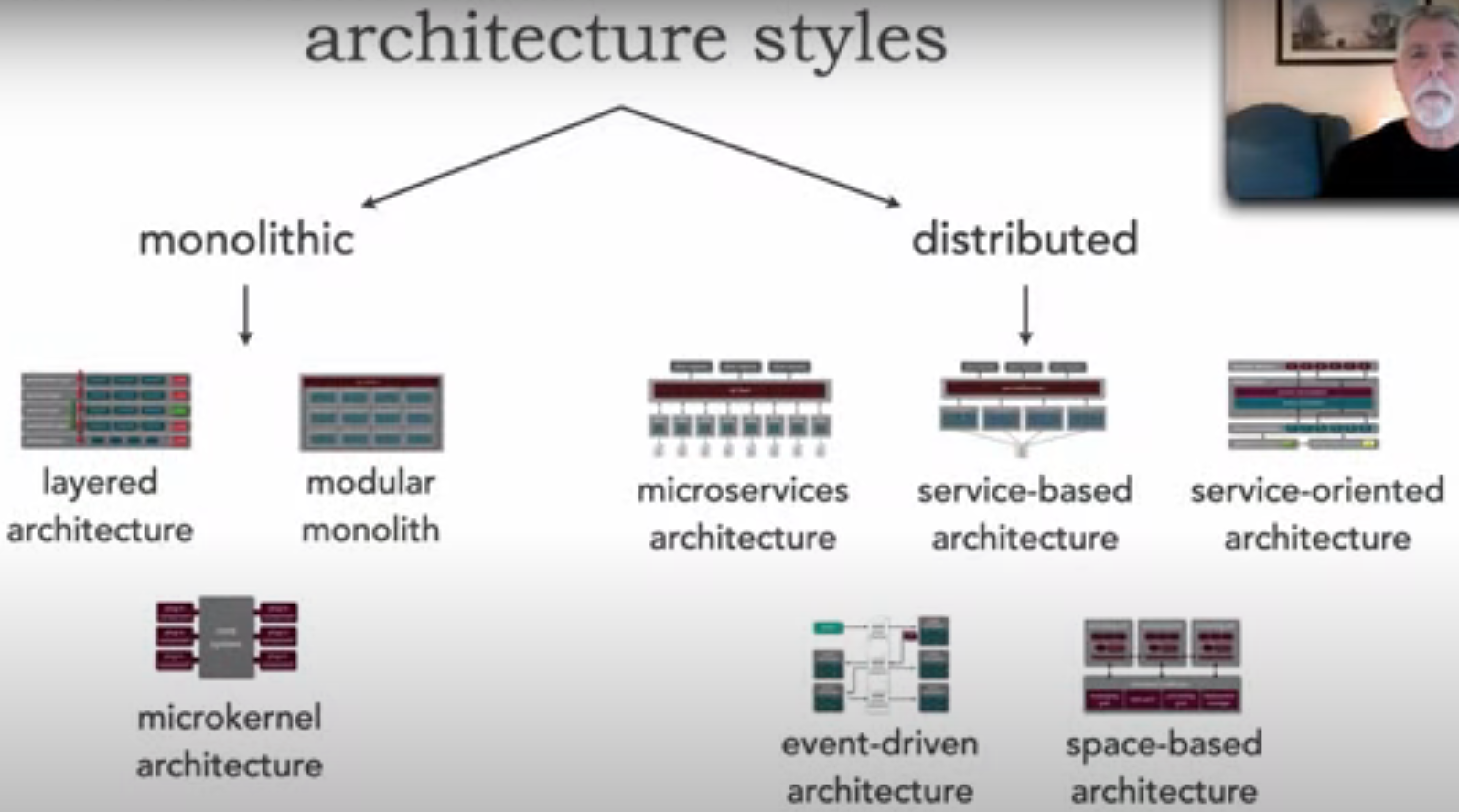
Trầy nhiều: trừ **10%**

* + - * + Độ sắc nét (tùy chọn)

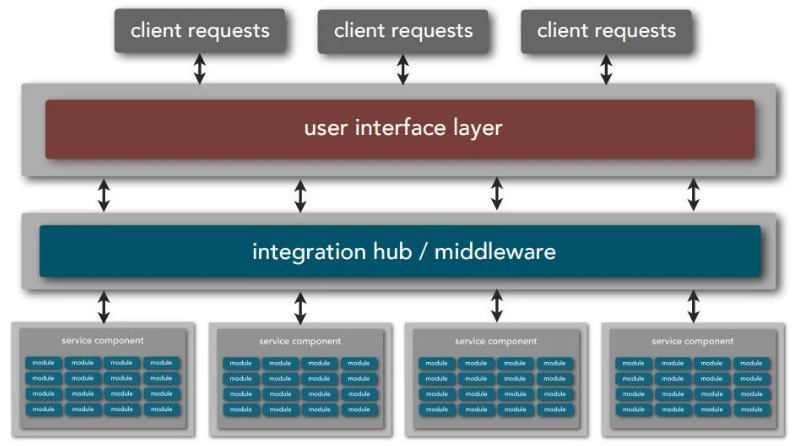
Nét ít

* + - Đèn pin (flash):
      * Bị cháy (bật nhưng không sáng đèn): trừ **10%**
    - Viền điện thoại (có trầy xước) (tùy chọn)
    - Chân sạc (có bị hư/gãy ko)
      * Không vào điện: trừ **10%**
    - Khay SIM (có hư ko)
      * Không nhận được sim: trừ **10%** mỗi khay sim
    - Rung:
      * Không rung: trừ **5%**
  + Ví dụ: Bạn có Iphone 13, màu đen, 128GB, với giá tiền 13.790.000 ngàn đồng, đã xài được 7 tháng, màn hình chưa thay bị trầy nhiều, lưng vỡ, pin chưa thay còn 80%, camera trước bị trầy nhẹ, camera sau hỏng 1 cái, đèn pin không hoạt động, cổng sạc bị hư, không rung:
    - Giá gốc: 13.790.000
    - Sau khi tính 7 tháng sử dụng: 13.790.00 – 13.790.00 \* 35% = 8963500
    - Sau khi tính màn hình bị trầy nhiều: 8963500 – 8963500 \* 15% = 7618975
    - Sau khi tính lưng vỡ: 7618975 \* (1 – 15%) = 6476128.75
    - Sau khi tính pin còn 80%: 6476128.75– 6476128.75\* 5% = 6152322.3125
    - Sau khi tính camera trước bị trầy nhẹ: 6152322.3125\* (1 – 3%) = 5967752.64312
    - Sau khi tính camera sau hỏng 1 cái: 5967752.64312\* (1 – 10%) = 5370977.37881
    - Sau khi tính đèn pin hư: 5370977.37881\* (1 – 10%) = 4833879.64093
    - Sau khi tính cổng sạc bị hư: 4833879.64093\* (1 – 10%) = 4350491.67684
    - Sau khi tinh rung bị hư: 4350491.67684 \* (1 – 10%) = 3915442.50916
* Nếu khách hàng chấp nhận về mức giá này, khách hàng có thể đăng kí tài khoản trên hệ thống với tên khách hàng, số điện thoại, địa chỉ, email và thông tin thanh toán. Nếu khách hàng đã có tài khoản trước đó thì không cần đăng kí.
  + Sau đó khách hàng có thể đăng nhập và gửi yêu cầu tái chế thiết bị, bao gồm số điện thoại tên thiết bị, các thông số mà khách hàng đã nhập trước đó và video khách hàng đóng gói thiết bị và hệ thống sẽ tạo một hóa đơn với trạng thái “**Hàng chưa được gửi**”
* Khách hàng có thể lên website với thông tin của mình để xem trạng thái của thiết bị của mình (đang gửi, đã nhận, đang tái chế)
* Khách hàng có thời hạn 1 tuần để gửi thiết bị và video lên cho công ty kể tử ngày lập đơn:
  + Khách hàng có thể gửi video đóng gói qua hóa đơn mà hệ thống đã tạo
  + Nếu khách hàng đã gửi thiết bị vật lí thì hóa đơn sẽ đổi trạng thái thành “**hàng đã được gửi**” và gửi mail xác nhận cho khách hàng
  + Nếu sau 1 tuần mà ko có thiết bị vật lí thì công ty sẽ gửi mail thông báo và hủy hóa đơn của khách hàng, cập nhật trạng thái hóa đơn thành “**đã hủy**”
* Công ty sau khi nhận được thiết bị và video minh chứng sản phẩm thì công ty phải trả về video mở sản phẩm của khách hàng lên trên hóa đơn tương ứng của tài khoản khách hàng
* Công ty kiểm tra xem thiết bị khách hàng gửi có đúng như mô tả của khách hàng hay không (thiết bị sẽ có trạng thái là “**đang đánh giá**”)(công ty phải quay video đánh giá):
  + Sau khi đánh giá xong thì công ty phải gửi email với thông tin là số tiền tương ứng cho thiết bị của khách hàng
    - Nếu khách hàng chấp nhận thì công ty thanh toán qua thông tin phương thức thanh toán của khách hàng, gửi mail xác nhận đã thanh toán và cập nhật trạng thái hóa đơn thành “**đã thanh toán**”
    - Nếu khách hàng không chấp nhận thì công ty sẽ đánh giá thiết bị lại một lần nữa, gửi mail đánh giá thiết bị cho khách hàng với số tiền mới.
      * Nếu khách hàng chấp nhận thì gửi tiền như trên
      * Nếu khách hàng không chấp nhận thì công ty gửi thiết bị về lại cho khách hàng với mail thông báo đã gửi thiết bị, và cập nhật trạng thái hóa đơn thành “**đã trả về**”

1. **Cơ sở lý thuyết**

****

* Service-based:
  + Khái niệm 2: Kiến trúc service-based là một kiến trúc hỗn hợp của microservice, đại diện cho một ngữ cảnh ứng dụng duy nhất, là một kiến trúc phân tán, 1 trong 2 nhánh của kiến trúc styles, là một phương pháp thiết kế ứng dụng phần mềm dưới dạng tập hợp các dịch vụ lỏng lẻo. Các dịch vụ này là các đơn vị tự chứa thực hiện các nhiệm vụ cụ thể và giao tiếp với nhau thông qua các giao diện được xác định rõ ràng, có thể hoặc không xài chung một cơ sở dữ liệu.

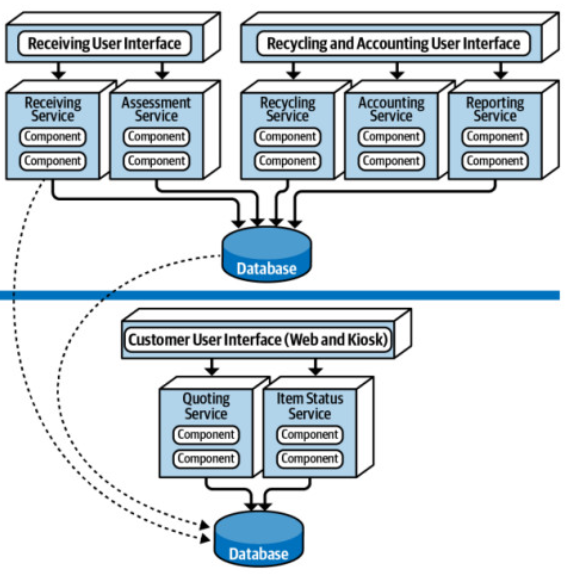


* + Lý do chọn service-based:
    - Hệ thống được miêu tả ở trên có 7 dịch vụ (quoting, receiving, assessment, accounting, item status, recyling, reporting), phù hợp để chia thành 7 domain cho kiến trúc này, tận dụng tối đa ưu điểm của kiến trúc
    - Hệ thống áp dụng kiến trúc này cho khả năng chịu lỗi tốt: một domain trong 7 domain bị hư thì sẽ không ảnh hướng đến các domain còn lại do chúng không giao tiếp trực tiếp với nhau
    - Chi phí hiện thực thấp, phù hợp cho các đội ngũ phát triển có quỹ chi phí vừa đủ
  + Ưu/nhược:

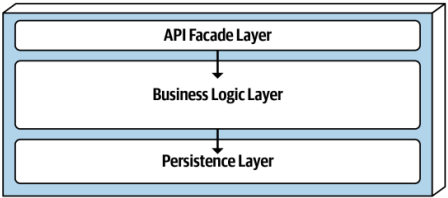
|  |  |
| --- | --- |
| Ưu điểm | Nhược điểm |
| Khả năng chịu lỗi tốt: Một service khi gặp lỗi thì sẽ không ảnh hưởng đến các service còn lại | Khó dùng cho hệ thống mà không có khả năng giãn nỡ do bản chất của các domain: các domain phản ứng quá chậm với việc giãn nỡ |
| Ít tốn chi phí:   * Không cần phải tách rời database ra để hiện thực kiến trúc này * Không cần devops để tự động hóa vận hành và kubernetes để dùng kiến trúc này | Nếu các domain trong hệ thống có coupling chặt chẽ với nhau thì không nên dùng kiến trúc này: sẽ có quá nhiều giao tiếp giữa các domain, trở thành một quả bóng bùn |
| Khả năng deploy cao: Các thành phần domain của kiến trúc này có thể được deploy từng đợt, không phải deploy như một khối thống nhất | Không phù hợp cho hệ thống có quá nhiều domain: phải chia nhỏ cơ sở dữ liệu để phù hợp với nhiều domain |
| Sử dụng tốt nếu như hệ thống có từ 4 đến 12 domain, tốt nhất là 7 |  |
| Là điểm khởi đầu tốt trước khi hiện thực microservice bởi vì kiến trúc này cho phép nhận ra các thành phần domain và không cần phải xử lí việc chia nhỏ dữ liệu trong cơ sở dữ liệu sớm. Các thành phần domain này có thể được phân tích sâu hơn để tìm ra cái nào có thể trở thành microservice |  |

* + Đánh giá tổng quan

1. **Phân tích và thiết kế**



Hệ thống sẽ được hiện thực thông qua kiến trúc service-based với các lí do đã được miêu tả ở trên. 7 dịch vụ trong hệ thống sẽ được triển khai dưới dạng domain theo kiến trúc này. Mỗi domain này sẽ được triển khai theo kiến trúc layer-by-layer như hình dưới



Chi tiết 7 domain trong kiến trúc service-based:

* Quoting Service:

Business Layer

* + - Các luồng xử lí thông tin số điện thoại mà khách hàng đã nhập vào trên trang web
  + Persistence Layer
    - Phone DAO
    - User DAO
  + Database Layer
    - Order
* Receiving Service:
  + Business Layer
    - Nghiệp vụ đăng nhập
    - Nghiệp vụ đăng ký
      * Đăng ký thông tin cá nhân, địa chỉ và thông tin thanh toán trực tuyến
    - Nghiệp vụ xử lí hóa đơn
  + Persistence Layer
    - User DAO
    - Order DAO
  + Database Layer
    - User
    - Order
* Assessment Service:
  + Business Layer
    - Nghiệp vụ kiểm tra thiết bị xem có hoạt động tốt hay không
    - Nghiệp vụ gửi email thông báo thiết bị có đủ tiêu chí không
  + Persistence Layer
    - Criteria DAO
    - Order DAO
    - Device DAO
  + Database Layer
    - Criteria
* Acounting Service:
  + Business Layer
    - Nghiệp vụ gửi email kết quả đánh giá
    - Nghiệp vụ thanh toán hóa đơn cho người dùng nếu người dùng chấp nhận
    - Nghiệp vụ gửi thiết bị vật lí về cho người dùng nếu người dùng không chấp nhận
  + Persistence Layer
    - User DAO
    - Payment DAO
    - Address DAO
    - Order DAO
  + Database Layer
    - Order
    - Payment
* Item Status Service:
  + Business Layer
    - Nghiệp vụ hiển thị tất cả hóa đơn mà người dùng đã tạo
    - Nghiệp vụ chỉnh sửa
  + Persistence Layer
    - Order DAO
    - User DAO
  + Database Layer
    - Order
* Recycling Service:
  + Business Layer
    - Nghiệp vụ thay đổi trạng thái hóa đơn
  + Persistence Layer
    - Order DAO
    - Device DAO
  + Database Layer
    - Order
    - Device
* Reporting Service:
  + Business Layer
    - Nghiệp vụ báo cáo tổng số hóa đơn theo ngày/tháng/quý/năm
  + Persistence Layer
    - Order DAO
  + Database Layer
    - Order DAO

1. **Công nghệ**

Front-end: React JS, Typescript

Back-end: Nodejs

Database: MongoDB

Kiến trúc: Service-based

1. **Định hướng phát triển**

**Phone**

**Order**

**User**

**Address**

**Payment**

**Order**

**Status:**

**Chưa gửi thiết bị**

**Đã gửi thiết bị**

**Đã kiểm tra thiết bị**

**Đã trả về thiết bị**

**Đã thanh toán**

**Đã tái chế**

**Đã bán lại**

**Đã phá hủy**