**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP.HCM**

A logo with a red and blue letter and a drop of water

Description automatically generated

**quản lí webite tái chế thiết bị điện tử**

***Nhóm 07 - Sinh viên thực hiện***

1. Nguyễn Thanh Hậu– 20043511
2. Trần Văn Bảo-20050241
3. Ngô Thiên Bâng-19523171

# MỤC LỤC

[MỤC LỤC 1](#_Toc167116156)

[1: GIỚI THIỆU 3](#_Toc167116157)

[1.1 Tổng quan 3](#_Toc167116158)

[1.2 Mục tiêu đề tài 3](#_Toc167116159)

[1.3 Phạm vi đề tài 4](#_Toc167116160)

[***Chức năng dành cho người quản trị*** 4](#_Toc167116161)

[1.4 Mô tả yêu cầu chức năng 4](#_Toc167116162)

[*Mô tả hoạt động của hệ thống* 4](#_Toc167116163)

[2: CƠ SỞ LÝ THUYẾT 6](#_Toc167116164)

[3: PHÂN TÍCH VÀ THIẾT KẾ 16](#_Toc167116165)

[3.1 Phân tích yêu cầu bằng UML 16](#_Toc167116166)

[Usecase tổng quát 16](#_Toc167116167)

[Danh sách tác nhân và mô tả 17](#_Toc167116168)

[Danh sách các tình huống hoạt động (Use cases) 17](#_Toc167116169)

[3.2 Tình huống hoạt động 18](#_Toc167116170)

[3.3 Kiến trúc 37](#_Toc167116171)

[**Chương 4: KẾT LUẬN** 39](#_Toc167116172)

[**4.1 Giao diện** 39](#_Toc167116173)

[**4.2 Kết quả đạt được** 41](#_Toc167116174)

[**4.3 Hạn chế** 42](#_Toc167116177)

[**4.4 Hướng phát triển** 42](#_Toc167116181)

[**4.5 Tài liệu tham khảo** 42](#_Toc167116181)

# 1: GIỚI THIỆU

## 1.1 Tổng quan

Mô tả về hệ thống quản lí website tái chế thiết bị điện tử của một shop thu mua điện thoại cũ. Hệ thống này được thiết kế để cung cấp cho shop thu mua một cách tiếp cận dễ dàng và linh hoạt, đồng thời hỗ trợ họ trong việc quản lý khách hàng và theo dõi danh sách của khác hàng và danh sách bán hàng. Ngoài ra, hệ thống còn cung cấp các tính năng hỗ trợ cho việc quản lý khách hàng cho phép người dùng đăng ký tài khoản.

Và quản lý đơn hàng hỗ trợ quản lý đơn hàng, bao gồm xác nhận, xử lý và theo dõi trang thái giáo hàng.

## 1.2 Mục tiêu đề tài

Mục tiêu của hệ thống quản lý website tái chế thiết bị điện tử, với mục tiêu chính là cung cấp cho người dùng một trải nghiệm bán sản phẩm linh hoạt và hiệu quả. Hệ thống sẽ tập trung vào việc giúp người dùng dễ dàng sử dụng và bán các sản phẩm tái chế, đồng thời hỗ trợ họ theo dõi và quản lý danh sách đơn hàng một cách thuận tiện. Mục tiêu của hệ thống không chỉ dừng lại ở việc hỗ trợ quá trình bán hàng mà còn là cung cấp các tiện ích như quản lý thông tin cá nhân, sản phẩm, và đơn hàng. Đồng thời, hệ thống cũng nhằm tạo điều kiện thuận lợi cho người dùng sau khi hoàn thành quá trình bán hàng, bằng cách cung cấp các tính năng như thống kê hoạt động bán hàng và báo cáo, giúp họ đánh giá và cải thiện hiệu suất kinh doanh. Mở rộng hệ thống, chúng tôi cũng nhấn mạnh vào việc tích hợp các tính năng phân tích dữ liệu để cung cấp cá nhân hóa hơn cho người dùng, từ việc đề xuất sản phẩm phù hợp đến kết nối với các cơ hội thực tập và việc làm sau này. Với nền tảng website người dùng, họ có thể dễ dàng quản lý thông tin cá nhân, sản phẩm, và đơn hàng, tạo ra một môi trường mua bán trực tuyến thuận tiện và linh hoạt. Với nền tảng website người dùng có thể :

* Xem và chỉnh sửa sản phẩm
* Xem danh sách khách hàng
* Xem danh sách sản phẩm khách hàng đã thêm
* Xác nhận, hủy, duyệt đơn của khách hàng
* Thêm sản phẩm

## 1.3 Phạm vi đề tài

#### Chức năng dành cho người dùng:

* Khách hàng thêm được đơn hàng, Xem danh sách đơn hàng đang được thêm và chở duyệt, đăng nhập và đăng kí cho khách hàng, tìm kiểm sản phầm bằng mã.
* Có giao diện trực quan và thân thiện, dễ tương tác và sử dụng thuận tiện.

#### **Chức năng dành cho người quản trị**:

Ngoài các yêu cầu giống như của khách hàng, thì hệ thống phải đảm bảo những yêu cầu sau:

* Thêm sản phẩm ,sửa giá, và xác nhận đơn hàng.

## 1.4 Mô tả yêu cầu chức năng

#### Phân tích yêu cầu của hệ thống

- Người dùng có thể thêm sản phẩm họ mún bán…

- Người dùng thực hiện đăng ký tài khoản trên web, hệ thống sẽ tiếp nhận và phản hồi cho đối tượng đăng kí.

- Từ mô tả trên, có thể đưa ra yêu cầu của hệ thống với hai đối tượng chính tương tác với hệ thống như sau:

• Đối với hệ thống:

- Thêm sản phẩm, sửa giá sản phẩm, xóa các đơn hàng

- Tìm kiếm thông tin sản phẩm

- Quản lý danh sách sản phẩm

• Đối với người dùng:

- Khách hàng có thể đăng ký, đăng nhập vào tài khoản,có thể xem thêm được sản phẩm cần mún bán và xem được danh sách chờ xác nhận của shop.

#### Mô tả hoạt động của hệ thống

1. Giao diện người dùng (UI):Giao diện người dùng cung cấp trải nghiệm dễ sử dụng cho người dùng khi họ truy cập trang web. Đây là nơi người dùng tìm kiếm thông tin về việc tái chế thiết bị điện tử, đăng ký tham gia hoạt động tái chế, và tìm hiểu về các sản phẩm tái chế có sẵn.
2. Chức năng tìm kiếm:Hệ thống cung cấp một công cụ tìm kiếm cho phép người dùng tìm kiếm sản phẩm khi thêm vào danh sách bán hàng.
3. Đăng ký và quản lý tài khoản:Người dùng có thể đăng ký tài khoản để đăng nhập vào website thêm sản phẩm cần bán hoặc theo dõi trạng thái của đơn hàng và giao dịch.
4. Thông tin sản phẩm và dịch vụ:Hệ thống cung cấp thông tin chi tiết về các sản phẩm tái chế có sẵn, mô tả sản phẩm.
5. Quản lý kho hàng:Hệ thống quản lý kho hàng để theo dõi số lượng và trạng thái của các sản phẩm tái chế khi người dùng nhập vào .

Top of Form

# 2: CƠ SỞ LÝ THUYẾTTop of Form

**2.1 Khái niệm** [Visual Studio Code](https://vinasupport.com/thu-thuat/visual-studio-code/)

 (VS Code) là một trình soạn thảo được phát triển bởi Microsoft, với ưu điểm nhẹ nhàng, dễ sử dụng, dễ dàng mở rộng với nhiều extension được cộng đồng lập trình viên hỗ trợ. VS Code được viết bằng Javascript (Electron) – 1 ngôn ngữ lập trình đã được mình giới thiệu trong bài viết [5 ngôn ngữ lập trình để viết ứng dụng Desktop cho Linux tốt nhất](https://vinasupport.com/5-ngon-ngu-lap-trinh-de-viet-ung-dung-desktop-cho-linux/). Vì vậy VS Code có thể chạy trên nhiều hệ điều hành khác nhau như Windows, Linux, MacOS.

- Visual Studio Code là một trình soạn thảo tuyệt vời cho việc [lập trình PHP](https://vinasupport.com/tag/lap-trinh-php/). Với các tính năng như syntax highlighting, bracket matching, IntelliSense (gợi nhớ code) và bạn có thể thêm nhiều chức năng hơn thông qua các extension được tạo bởi cộng đồng lập trình viên. Các bạn có thể tìm hiểu thêm các extension thông qua bài viết: [Tổng hợp các extension nên dùng cho Visual Studio Code](https://vinasupport.com/tong-hop-extension-nen-dung-cho-visual-studio-code/).

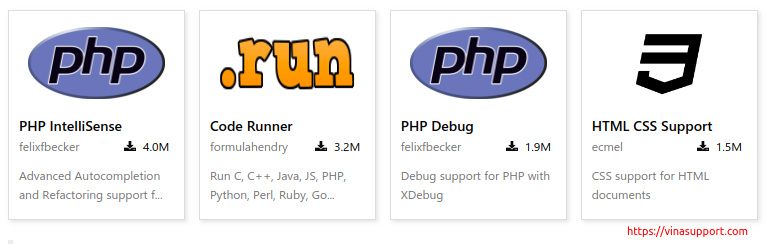
A screenshot of a computer

Description automatically generated

Hình ảnh về Visual Studio Code

**2.2 Khái niệm ngôn ngữ lập trình PHP**

+) Để bắt đầu lập trình PHP sử dụng Visual Code đầu tiên các bạn cần cài đặt 1 số extension hỗ trợ. Các extension cho PHP thì có rất nhiều trên [VS Code Marketplace](https://vinasupport.com/tag/vs-code-marketplace/) nhưng dưới đây là những [VS Code Extension](https://vinasupport.com/tag/visual-studio-code-extension/) được khuyến khích sử dụng.



+) [PHP IntelliSense](https://marketplace.visualstudio.com/items?itemName=felixfbecker.php-intellisense): Gợi nhớ code (Suggest), Tự động hoàn thành code (Auto completion).

+) [Code Runner](https://marketplace.visualstudio.com/items?itemName=formulahendry.code-runner): Chạy PHP Script.

+) [PHP Debug](https://marketplace.visualstudio.com/items?itemName=felixfbecker.php-debug): Dành cho việc debug ứng dụng PHP, script PHP.

+) [HTML CSS Support](https://marketplace.visualstudio.com/items?itemName=ecmel.vscode-html-css): Hỗ trợ viết mã HTML/CSS.

- **PHP: Hypertext Preprocessor**, thường được viết tắt thành **PHP** là một [ngôn ngữ lập trình](https://vi.wikipedia.org/wiki/Ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh" \o "Ngôn ngữ lập trình) kịch bản hay một loại [mã lệnh](https://vi.wikipedia.org/wiki/M%C3%A3_l%E1%BB%87nh" \o "Mã lệnh) chủ yếu được dùng để phát triển các [ứng dụng](https://vi.wikipedia.org/wiki/%E1%BB%A8ng_d%E1%BB%A5ng_web" \o "Ứng dụng web) viết cho máy chủ, [mã nguồn mở](https://vi.wikipedia.org/wiki/Ph%E1%BA%A7n_m%E1%BB%81m_ngu%E1%BB%93n_m%E1%BB%9F" \o "Phần mềm nguồn mở), dùng cho mục đích tổng quát. Nó rất thích hợp với [web](https://vi.wikipedia.org/wiki/Internet) và có thể dễ dàng nhúng vào trang [HTML](https://vi.wikipedia.org/wiki/HTML). Do được tối ưu hóa cho các [ứng dụng web](https://vi.wikipedia.org/wiki/%E1%BB%A8ng_d%E1%BB%A5ng_web" \o "Ứng dụng web), tốc độ nhanh, nhỏ gọn, cú pháp giống [C](https://vi.wikipedia.org/wiki/C_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)) và [Java](https://vi.wikipedia.org/wiki/Java_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)), dễ học và thời gian xây dựng sản phẩm tương đối ngắn hơn so với các ngôn ngữ khác nên PHP đã nhanh chóng trở thành một [ngôn ngữ lập trình](https://vi.wikipedia.org/wiki/Ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh" \o "Ngôn ngữ lập trình) web phổ biến nhất thế giới.

- Ngôn ngữ, các thư viện, tài liệu gốc của PHP được xây dựng bởi cộng đồng và có sự đóng góp rất lớn của [Zend Inc.](https://vi.wikipedia.org/w/index.php?title=Zend_Inc.&action=edit&redlink=1), công ty do các nhà phát triển cốt lõi của PHP lập nên nhằm tạo ra một môi trường chuyên nghiệp để đưa PHP phát triển ở quy mô [doanh nghiệp](https://vi.wikipedia.org/wiki/Doanh_nghi%E1%BB%87p" \o "Doanh nghiệp).



Hình 1.1: Giới thiệu về ngôn ngữ lập trình PHP

**2.3 Khái niệm PhpMyAdmin**

- Là một công cụ [nguồn mở miễn phí](https://vi.wikipedia.org/wiki/Ph%E1%BA%A7n_m%E1%BB%81m_ngu%E1%BB%93n_m%E1%BB%9F" \o "Phần mềm nguồn mở) được viết bằng ngôn ngữ lập trình [PHP](https://vi.wikipedia.org/wiki/PHP) để xử lý các tác vụ quản trị của [MySQL](https://vi.wikipedia.org/wiki/MySQL) thông qua một [trình duyệt web](https://vi.wikipedia.org/wiki/Tr%C3%ACnh_duy%E1%BB%87t_web" \o "Trình duyệt web). Nó có thể thực hiện nhiều tác vụ như tạo, sửa đổi hoặc xóa bỏ [cơ sở dữ liệu](https://vi.wikipedia.org/wiki/C%C6%A1_s%E1%BB%9F_d%E1%BB%AF_li%E1%BB%87u), [bảng](https://vi.wikipedia.org/w/index.php?title=B%E1%BA%A3ng_(c%C6%A1_s%E1%BB%9F_d%E1%BB%AF_li%E1%BB%87u)&action=edit&redlink=1" \o "Bảng (cơ sở dữ liệu) (trang không tồn tại)), các [trường](https://vi.wikipedia.org/w/index.php?title=Tr%C6%B0%E1%BB%9Dng_(c%C6%A1_s%E1%BB%9F_d%E1%BB%AF_li%E1%BB%87u_quan_h%E1%BB%87)&action=edit&redlink=1" \o "Trường (cơ sở dữ liệu quan hệ) (trang không tồn tại)) hoặc [bản ghi](https://vi.wikipedia.org/wiki/B%E1%BA%A3n_ghi" \o "Bản ghi); thực hiện theo báo cáo [SQL](https://vi.wikipedia.org/wiki/SQL); hoặc quản lý người dùng và cấp phép.

- Là một công cụ quản trị dựa trên web để quản lý cơ sở dữ liệu MySQL và MariaDB. Nó cung cấp giao diện đồ họa người dùng (GUI) để tương tác với cơ sở dữ liệu, giúp người dùng thực hiện các tác vụ quản lý cơ sở dữ liệu khác nhau dễ dàng hơn mà không cần kiến ​​thức sâu rộng về các lệnh SQL.

A screenshot of a computer

Description automatically generated

Hình 1.2: Giới thiệu về PHP my admin.

**2.4 Khái niệm Xampp**

- Là một công cụ tích hợp đầy đủ các thành bao gồm Apache, Mysql, PHP, Perl. Giúp chúng ta tạo ra môi trường web server trên máy tính của mình, để có thể chạy được kịch bản PHP.

* XAMPP là gói giải pháp máy chủ web đa nền tảng mã nguồn mở miễn phí được phát triển bởi Apache Friends. Từ viết tắt XAMPP là viết tắt của Đa nền tảng, Apache, MySQL, PHP và Perl. Nó được thiết kế để giúp dễ dàng thiết lập môi trường máy chủ web cục bộ cho mục đích thử nghiệm và phát triển. XAMPP bao gồm các thành phần thiết yếu cần thiết cho việc phát triển web, giúp đơn giản hóa quá trình định cấu hình và quản lý các thành phần này.

A screenshot of a computer program

Description automatically generated

Hình 1.3: Giới thiệu về Xampp

Apache: Là một chương trình máy chủ, dùng để giao tiếp với các giao thức HTTP. Apache chạy tốt trên mọi hệ điều hành.

Mysql: Là hệ quản trị cơ sở dữ liệu, được dùng để lưu trữ thông tin của một website. Mỗi website có thể sử dụng một hoặc nhiều cơ sở dữ liệu.

PHP: Là ngôn ngữ kịch bản trên phía Server, dùng để xử lý các thao tác của người dùng và làm việc trực tiếp với cơ sở dữ liệu (Database).

Perl: Là một tầng cao hơn, một ngôn ngữ lập trình năng động hơn. Sử dụng rộng rãi trong lập trình mạng và quản trị hệ thống. Ít phổ biến cho mục đích phát triển web, Perl thích hợp với nhiều ứng dụng.

Xampp là một công cụ tuyệt vời cho bất kỳ ai muốn có một development server đầy đủ và nhanh chóng. Nhưng bên cạnh đó Xampp rất dễ cài đặt, nên nó không có các tính năng bảo mật như production server.

**2.5 Kiến trúc 3 layered architecture**

**2.5.1.Kiến trúc 3 *layered architecture* là gì?**

- Mô hình 3 lớp hay còn được gọi là mô hình Three Layer(3-Layer), mô hình này ra đời nhằm phân chia các thành phần trong hệ thống, các thành phần cùng chức năng sẽ được nhóm lại với nhau và phân chia công việc cho từng nhóm để dữ liệu không bị chồng chéo và chạy lộn xộn.

- Mô hình này phát huy hiệu quả nhất khi bạn xây dựng một hệ thống lớn, việc quản lý code và xử lý dữ liệu lỗi dễ dàng hơn.

Ưu điểm:

- Phân loại rõ ràng các lớp có các nhiệm vụ khác nhau. Từ đó ta có thể quản lý và maintain project tốt hơn.

- Dễ dàng phân loại các hành động tại Business.

- Dễ dàng phân loại các hàm truy xuất tại Database, phân loại hàm theo table,…

- Ứng dụng được cho các project lớn ở bên ngoài.

A diagram of a software development

Description automatically generated

Mô hình 3-layer gồm có 3 phần chính:

Presentation Layer (GUI):

* Lớp này có nhiệm vụ chính là giao tiếp với người dùng. Nó gồm các thành phần giao diện ( winform, webform, …) và thực hiện các công việc như nhập liệu, hiển thị dữ liệu, kiểm tra tính đúng đắn dữ liệu trước khi gọi lớp Business Logic Layer (BLL).

Business Logic Layer (BLL) :

* Layer này phân ra 2 thành nhiệm vụ: Đây là nơi đáp ứng các yêu cầu thao tác dữ liệu của GUI layer, xử lý chính nguồn dữ liệu từ Presentation Layer trước khi truyền xuống Data Access Layer và lưu xuống hệ quản trị CSDL.
* Đây còn là nơi kiểm tra các ràng buộc, tính toàn vẹn và hợp lệ dữ liệu, thực hiện tính toán và xử lý các yêu cầu nghiệp vụ, trước khi trả kết quả về Presentation Layer.

Data Access Layer (DAL) :

* Lớp này có chức năng giao tiếp với hệ quản trị CSDL như thực hiện các công việc liên quan đến lưu trữ và truy vấn dữ liệu ( tìm kiếm, thêm, xóa, sửa,…).

**2.5.2. Áp dụng kiến trúc 3 Layer Architecture vào project PHP**

**2.5.2.1. Các vấn đề thường gặp khi sử dụng nền tảng PHP**

- Về quản lý mã nguồn, PHP là một ngôn ngữ linh hoạt và không yêu cầu một cấu trúc cụ thể, dẫn đến việc có thể viết mã nguồn một cách không có tổ chức. Điều này khiến cho việc quản lý mã nguồn trở nên khó khăn, đặc biệt là trong các dự án lớn. Thứ hai là tính mở rộng và bảo trì, khi không có sự phân chia rõ ràng về các lớp logic và dữ liệu có thể gây ra vấn đề khi sửa đổi hoặc thêm mới tính năng. Vấn đề thứ ba là về tính kiểm thử, khi việc kiểm thử trở nên phức tạp hơn do mã nguồn không được phân chia thành các thành phần logic riêng biệt. Thứ tư là về hiệu suất, khi một kiến trúc không tổ chức có thể dẫn đến hiệu suất kém do việc tìm kiếm và thực thi mã nguồn không được tối ưu. Vấn đề tiếp theo là về bảo mật, khi thiếu sự phân tách giữa các lớp có thể gây ra lỗ hổng bảo mật, đặc biệt là khi xử lý dữ liệu người dùng. Cuối cùng là vấn đề về quản lý dự án, khi việc phối hợp giữa các thành viên trong dự án trở nên khó khăn hơn khi mã nguồn không được tổ chức một cách logic và có cấu trúc.

**2.5.2.2. Áp dụng kiến trúc 3 Layer**

Cấu trúc này gồm ba lớp chính: Presentation Layer, Business Logic Layer, và Data Access Layer .

- Presentation Layer: Được sử dụng để hiển thị giao diện người dùng và tương tác với người dùng. Ở đây, các controller hoặc các file tương tự được đặt trong thư mục controller. Chúng xử lý yêu cầu từ người dùng và gửi các yêu cầu xử lý đến tầng Business Logic Layer.

- Business Logic Layer: Là nơi thực hiện các xử lý nghiệp vụ của ứng dụng, bao gồm kiểm tra tính hợp lệ của dữ liệu, tính toán, và các quy tắc kinh doanh khác. Trong dự án PHP, các class hoặc các tệp tin tương tự được đặt trong thư mục service hoặc business. Chúng không trực tiếp tương tác với người dùng hoặc dữ liệu, mà thay vào đó, chúng nhận các yêu cầu từ tầng Presentation và gửi các yêu cầu tương tác dữ liệu đến tầng Data Access Layer.

- Data Access Layer: Được sử dụng để tương tác với cơ sở dữ liệu, bao gồm các thao tác như truy vấn, cập nhật, xóa và thêm dữ liệu. Trong dự án PHP, các class hoặc các tệp tin tương tự được đặt trong thư mục dao hoặc repository. Chúng thực hiện các thao tác truy vấn và cập nhật dữ liệu dựa trên yêu cầu từ tầng Business Logic.

* Cấu trúc thư mục (Folder structure)

A screen shot of a computer

Description automatically generated

Config: Thường thì trong thư mục này sẽ chứa các tệp tin cấu hình như kết nối đến cơ sở dữ liệu, cài đặt hệ thống, v.v.

Controller: Đây những tệp này có nhiệm vụ điều khiển luồng dữ liệu và xử lý các yêu cầu từ người dùng.

Data Access Object: Chịu trách nhiệm giao tiếp với cơ sở dữ liệu và thực hiện các thao tác truy vấn.

Model: Các tệp trong thư mục này sẽ chứa định nghĩa của các đối tượng dữ liệu (model) được sử dụng trong dự án.

View: Thư mục này là nơi chứa các tệp giao diện người dùng, được sử dụng để hiển thị dữ liệu cho người dùng.

**2.5.2.3 Ưu điểm**

- Phân chia rõ ràng và tổ chức dự án một cách logic: Kiến trúc ba lớp giúp phân tách ứng dụng thành ba phần chính: giao diện người dùng, logic kinh doanh và lớp dữ liệu. Điều này giúp tổ chức code một cách logic và dễ dàng hiểu.

- Quản lý dễ dàng: Dễ dàng quản lý và bảo trì mã nguồn vì mã được phân chia thành các lớp có chức năng riêng biệt. Mỗi lớp có nhiệm vụ cụ thể, giúp việc debug và fix lỗi trở nên dễ dàng hơn.

- Tính tái sử dụng cao: Các lớp có thể được sử dụng lại trong các phần khác nhau của dự án hoặc trong các dự án khác, giúp tiết kiệm thời gian và công sức phát triển.

- Dễ dàng mở rộng: Cấu trúc linh hoạt của kiến trúc ba lớp giúp dễ dàng mở rộng ứng dụng khi cần thiết, bằng cách thêm mới các lớp hoặc thay đổi cấu trúc hiện có mà không ảnh hưởng đến các phần khác của hệ thống.

Nhược điểm:

- Độ phức tạp tăng lên: Dự án sử dụng kiến trúc ba lớp có thể trở nên phức tạp khi số lượng lớp và mối quan hệ giữa chúng tăng lên. Điều này có thể làm tăng độ phức tạp của mã nguồn và làm cho việc hiểu và bảo trì dự án trở nên khó khăn hơn.

- Yêu cầu kỹ năng cao: Phát triển và bảo trì một dự án sử dụng kiến trúc ba lớp đòi hỏi kiến thức chuyên sâu về cả phần mềm lẫn các công nghệ sử dụng, đặc biệt là trong việc thiết kế và triển khai các lớp kinh doanh và lớp dữ liệu.

- Hiệu suất có thể giảm: Việc chia nhỏ ứng dụng thành nhiều lớp có thể dẫn đến việc truy cập dữ liệu hoặc thực hiện các tác vụ kinh doanh qua nhiều lớp, điều này có thể ảnh hưởng đến hiệu suất của ứng dụng, đặc biệt là khi cần xử lý lượng dữ liệu lớn.

# 3: PHÂN TÍCH VÀ THIẾT KẾTop of Form

## 3.1 Phân tích yêu cầu bằng UML

### Usecase tổng quát

A diagram of a company

Description automatically generated

Hình 3.1 Usecase tổng quát

### Danh sách tác nhân và mô tả

Danh sách tác nhân và mô tả xem ở Bảng 3-1.

Bảng 3‑1 Danh sách tác nhân và mô tả

|  |  |
| --- | --- |
| Tác nhân | Mô tả tác nhân |
| User | * Đăng nhập * Đăng ký * Gửi thông tin thiết bị * Tìm thông tin đơn hàng * Bán thiết bị |
| Admin | * Báo giá * Cập nhật tình trạng đơn hàng * Quản lý nhãn hàng * Đăng nhập |

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

|  |  |
| --- | --- |
| **ID use case** | **Tên Use case** |
| **U1** | Đăng nhập |
| U2 | Đăng xuất |
| U3 | Đăng ký |
| U4 | Gửi thông tin thiết bị |
| U5 | Tìm thông tin đơn hàng |
| U6 | Bán thiết bị |
| U7 | Báo giá |
| U8 | Cập nhật tình trạng đơn hàng |
| U9 | Thêm nhãn hàng |

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

### 3.2 Tình huống hoạt động

#### UC01\_Đăng nhập

|  |  |
| --- | --- |
| **Tên use case**: UC01\_Đăng nhập | |
| **Mô tả sơ lược**: Truy cập vào hệ thống | |
| **Actor chính**: người dùng và nhân viên quản lý | |
| **Tiền điều kiện (Pre-condition):** Đã có tài khoản đăng nhập | |
| **Hậu điều kiện (Post-condition):** | |
| * **Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Người dùng truy cập trang đăng nhập | 1. Hệ thống hiển thị giao diện trang chủ |
| 1. Người dùng nhập thông tin tài khoản |  |
| 1. Chọn nút đăng nhập | 1. Hệ thống kiểm tra thông tin đăng nhập |
|  | 1. Hệ thống hiển trang chủ |
| * **Luồng sự kiện thay thế (alternate flow):** | |
|  | 6.1. Thông báo thông tin tài khoản đăng nhập không đúng |
| 6.2. Xác nhận | 6.2. Hệ thống quay lại bước 2 |

Activity diagram:

A diagram of a flowchart

Description automatically generated

Sequence diagram:

A diagram of a project

Description automatically generated

#### UC02\_Đăng xuất

|  |  |
| --- | --- |
| **Tên use case**: UC02\_ Đăng xuất | |
| **Mô tả sơ lược**: Ngừng truy cập vào hệ thống | |
| **Actor chính**: Người dùng và nhân viên quản lý | |
| **Tiền điều kiện (Pre-condition):** Đăng nhập thành công | |
| **Hậu điều kiện (Post-condition):** | |
| * **Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Bấm vào đăng xuất | 2. Ngừng truy cập tài khoản |
|  | 3. Hiện giao diện đăng nhập |

Activity diagram:

A diagram of a system

Description automatically generated

Sequence diagram:

A diagram of a project

Description automatically generated

#### UC03\_ Đăng ký

|  |  |
| --- | --- |
| **Tên use case**: UC03\_ Đăng ký | |
| **Mô tả sơ lược**: Đăng ký tài khoản mới dùng để đăng nhập | |
| **Actor chính**: người dùng | |
| **Tiền điều kiện (Pre-condition):** Username chưa có trong hệ thống | |
| **Hậu điều kiện (Post-condition):** Thêm Username mới vào hệ thống | |
| * **Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Tại trang đăng nhập, chọn đăng ký | 1. Hệ thống hiển thị giao diện đăng ký |
| 1. Người dùng nhập thông tin đăng ký |  |
| 1. Chọn nút Đăng ký | 1. Hệ thống kiểm tra thông tin |
|  | 1. Thông báo đăng ký thành công |
| **Luồng sự kiện thay thế (alternate flow):** | |
|  | 5.1. Thông báo thông tin đăng ký thất bại |
| 5.2. Xác nhận | 5.2. Hệ thống quay lại bước 2 |

Activity diagram:

A diagram of a flowchart

Description automatically generated

Sequence diagram:

A diagram of a diagram

Description automatically generated

#### UC04\_Gửi thông tin thiết bị

|  |  |
| --- | --- |
| **Tên use case**: UC04\_ Gửi thông tin thiết bị | |
| **Mô tả sơ lược**: Gửi thông tin thiết bị tái chế | |
| **Actor chính**: Người dùng | |
| **Tiền điều kiện (Pre-condition):** Đăng nhập thành công | |
| **Hậu điều kiện (Post-condition):** Thông tin được lưu vào cơ sở dữ liệu | |
| **Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Chọn bán thiết bị | 2. Hiển thị giao diện thêm thiết bị |
| 3. Nhập thông tin thiết bị | 4. Kiểm tra thông tin |
|  | 5. Thông báo gửi thành công |
| **Luồng sự kiện thay thế (alternate flow):** | |
|  | 4.1. Thông báo “Vui lòng nhập đầy đủ thông tin” |
|  | 4.2. Quay lại bước 2 |

Activity diagram:

A diagram of a chat

Description automatically generated with medium confidence

Sequence diagram:

A diagram of a diagram

Description automatically generated

#### UC05\_ Tìm thông tin đơn hàng

|  |  |
| --- | --- |
| **Tên use case**: UC05\_ Tìm thông tin đơn hàng | |
| **Mô tả sơ lược**: Cho phép người dùng xem thông tin đơn hàng đã gửi | |
| **Actor chính**: Người dùng | |
| **Tiền điều kiện (Pre-condition):** Đăng nhập thành công | |
| **Hậu điều kiện (Post-condition):** Hiện danh sách các đơn hàng | |
| **Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Chọn “Danh sách thiết bị” | 2. Hiển thị giao diện danh sách thiết bị |
| 3. Nhập mã đơn hàng | 4. Tìm mã đơn hàng |
|  | 5. Hiển thị đơn hàng cần tìm |
| **Luồng sự kiện thay thế (alternate flow):** | |
|  | 4.1. Không hiển thị sản phẩm nào |

Activity diagram:

A diagram of a diagram

Description automatically generated with medium confidence

Sequence diagram:

A diagram of a diagram

Description automatically generated

#### UC06\_ Bán thiết bị

|  |  |
| --- | --- |
| **Tên use case**: UC06\_ Bán thiết bị | |
| **Mô tả sơ lược**: Cho phép người dùng lựa chọn bán sản phẩm hoặc không | |
| **Actor chính**: Người dùng | |
| **Tiền điều kiện (Pre-condition):** Đăng nhập thành công. | |
| **Hậu điều kiện (Post-condition):** Cập nhật tình trạng đơn hàng | |
| **Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Chọn “Danh sách thiết bị” | 2. Hiển thị giao diện danh sách thiết bị |
| 3. Nhập mã đơn hàng | 4. Tìm mã đơn hàng |
|  | 5. Hiển thị đơn hàng cần tìm |
| 6. Chọn đồng ý | 7. Cập nhật trạng thái sản phẩm |
| **Luồng sự kiện thay thế (alternate flow):** | |
| 6.1. Chọn từ chối | 6.2. Cập nhật tình trạng đơn hàng là “Hủy” |

Activity diagram:

A diagram of a flowchart

Description automatically generated

Sequence diagram:

A diagram of a project

Description automatically generated

#### UC07\_ Báo giá

|  |  |
| --- | --- |
| **Tên use case**: UC07\_ Báo giá | |
| **Mô tả sơ lược**: Admin nhập giá thiết bị | |
| **Actor chính**: Admin | |
| **Tiền điều kiện (Pre-condition):** Đăng nhập thành công. | |
| **Hậu điều kiện (Post-condition):** Hệ thống cập nhật giá mới | |
| **Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Chọn “Danh sách sản phẩm” | 2. Hiển thị giao diện “Danh sách sản phẩm” |
| 3. Chọn đơn hàng |  |
| 4. Chọn “Đã nhận” | 5. Hiển thị nút nhập giá |
| 6. Chọn nhập giá | 7. Hiển thị giao diện nhập giá |
| 8. Nhập giá |  |
| 9. Chọn lưu | 10. Hệ thống lưu giá mới cho thiết bị |

Activity diagram:

A diagram of a flowchart

Description automatically generated

Sequence diagram:

A diagram of a diagram of a diagram

Description automatically generated

#### UC08\_ Cập nhật tình trạng đơn hàng

|  |  |
| --- | --- |
| **Tên use case**: UC08\_ Cập nhật tình trạng đơn hàng | |
| **Mô tả sơ lược**: cho phép admin cập nhật trang thái thiết bị | |
| **Actor chính**: Admin | |
| **Tiền điều kiện (Pre-condition):** Đăng nhập thành công. User đồng ý bán thiết bị | |
| **Hậu điều kiện (Post-condition):** Cập nhật tình trạng đơn hàng | |
| **Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Chọn “Danh sách sản phẩm” | 2. Hiển thị giao diện “Danh sách sản phẩm” |
| 3. Chọn đơn hàng |  |
| 4. Chọn “Tái chế” | 5. Cập nhật tình trạng thiết bị là “Tái chế” |
| **Luồng sự kiện thay thế (alternate flow):** | |
| 4. Chọn “Bán lại” | 5. Cập nhật tình trạng thiết bị là “Bán lại” |

Activity diagram:

A diagram of a diagram

Description automatically generated

Sequence diagram:

A diagram of a project

Description automatically generated

#### UC09\_ Thêm loại hàng

|  |  |
| --- | --- |
| **Tên use case**: UC09\_ Thêm hãng hàng | |
| **Mô tả sơ lược**: Cho phép admin thêm loại sản phẩm mới | |
| **Actor chính**: Người dùng | |
| **Tiền điều kiện (Pre-condition):** Đăng nhập thành công. Tìm thành công đơn hàng | |
| **Hậu điều kiện (Post-condition):** Cập nhật tình trạng đơn hàng | |
| **Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Chọn “Loại sản phẩm” | 2. Hiển thị giao diện “Loại thiết bị” |
| 3. Nhập tên loại |  |
| 4. Chọn thêm | 5. Thông báo thêm thành công |

Activity diagram:

A diagram of a system

Description automatically generated

Sequence diagram:

A diagram of a diagram

Description automatically generated

## 3.3 Kiến trúc

Kiến trúc sẽ có 2 phần là Client và Sever, ở trong mỗi phần sẽ có sự phân tầng khác nhau

A screenshot of a computer

Description automatically generated

**Client-Side Presentation Layer**

* HTML/CSS/JS (view):
  + HTML: Định nghĩa cấu trúc và nội dung của trang web.
  + CSS: Định nghĩa kiểu dáng và bố cục của trang web.
  + JavaScript: Xử lý tương tác động và sự kiện, quản lý giao tiếp bất đồng bộ với server (AJAX).

Server-Side Presentation Layer

* view:
* Hiển thị giao diện người dùng cho các chức năng cụ thể.
* Quản lý các thành phần giao diện như biểu mẫu đăng nhập, thông tin sản phẩm, hóa đơn giao hàng, và các phần tử chung như header và footer.
* model:
* Chứa các lớp mô hình đại diện cho dữ liệu của ứng dụng.
* Định nghĩa các thuộc tính và phương thức để quản lý dữ liệu của đối tượng.

**Business Logic Layer**

* controller:
* Xử lý các yêu cầu từ người dùng, định tuyến và điều phối các hành động cần thiết.
* Tương tác với các lớp dịch vụ (service layer) để thực hiện logic nghiệp vụ.
* Quản lý luồng dữ liệu giữa presentation layer và data access layer.

**Data Access Layer**

* dao:
* Thực hiện các thao tác truy cập và tương tác với cơ sở dữ liệu.
* Cung cấp các phương thức để thực hiện các thao tác CRUD (Create, Read, Update, Delete) trên dữ liệu.
* Đóng vai trò trung gian giữa business logic layer và database server, trừu tượng hóa các chi tiết của việc truy cập dữ liệu.

**Data Server (SQL Server)**

* config:
* Chứa các thông tin cấu hình cần thiết để kết nối đến cơ sở dữ liệu.
* Quản lý kết nối cơ sở dữ liệu, thiết lập các thông số kết nối, và bảo mật thông tin kết nối.
* Thực hiện việc khởi tạo và quản lý phiên làm việc với cơ sở dữ liệu.
* Mỗi layer trong kiến trúc này đóng một vai trò cụ thể và tương tác với các layer khác để đảm bảo hoạt động mượt mà của ứng dụng, từ việc hiển thị giao diện người dùng đến quản lý và lưu trữ dữ liệu.

# **Chương 4: KẾT LUẬN**

**4.1 Giao diện**

Một số hình ảnh về giao diện

A green globe with green arrows around it

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A white sheet with black text

Description automatically generated with medium confidence

**4.2 Kết quả đạt được**

* Hoàn thành các nghiệp vụ cơ bản cho người dùng và người quản lý: Hệ thống đã có thể thực hiện các chức năng chính để gửi yêu cầu, xem trạng thái và xử lý yêu cầu của người dùng, cũng như xác nhận, báo giá, huỷ hoặc tái chế từ phía người quản lý.
* Khách hàng phải đăng nhập mới thêm sản phẩm cần bán.

**4.3 Hạn chế**

* Chưa báo cáo được tài chính và hoạt động dựa trên hoạt động tái chế
* Xử lý thanh toán: xử lý thanh toán còn hơi sơ sài và không an toàn.
* Giao diện người dùng: còn hơi thô sơ chưa bắt mắt.

**4.4 Hướng phát triển**

* Khắc phục hạn chế và sửa đổi hệ thống: Tập trung vào việc giải quyết các hạn chế đã xác định và cải thiện hệ thống để phù hợp hơn với nhu cầu kinh doanh và quản lý.
* Phát triển website tái chế: Tạo ra một website tái chế thiết bị điện tử chuyên nghiệp và thân thiện với người dùng, có các tính năng mở rộng và tương tác để thu hút và giữ chân người dùng.
* Tối ưu hóa quy trình: Nâng cao hiệu suất hoạt động và giảm thiểu thời gian xử lý bằng cách tối ưu hóa quy trình và công nghệ sử dụng trong hệ thống.
* Nâng cấp tính năng và giao diện người dùng: Thêm các tính năng mới và cải thiện giao diện người dùng để tăng tính thân thiện và thu hút người dùng

**4.5 Tài liệu tham khảo**

<https://viblo.asia/p/tim-hieu-ve-mo-hinh-3-lop-3-layer-Qbq5Qmyz5D8>

[PHP: Hypertext Preprocessor](https://www.php.net/)

<https://200lab.io/blog/kien-truc-phan-lop/>

[Software Architecture Patterns](file:///C:\Users\Admin\Documents\Zalo%20Received%20Files\Software%20Architecture%20Patterns)

[Download XAMPP (apachefriends.org)](https://www.apachefriends.org/download.html)