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

**KHOA CÔNG NGHỆ THÔNG TIN**

**A logo with a red and blue design

Description automatically generated\*\*\*\*\*\*\*\*\*\*\***

**BÁO CÁO CUỐI KỲ**

**KIẾN TRÚC VÀ THIẾT KẾ PHẦN MỀM**

*Đề tài****: TÁI CHẾ LỐP XE CAO SU***

Giảng viên hướng dẫn: Võ Văn Hải

Lớp học phần: DHKTPM16ETT

Mã lớp học phần: 422000191403

Nhóm sinh viên thực hiện: Nhóm 5

TP. Hồ Chí Minh, ngày 20 tháng 05 năm 2024

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

**KHOA CÔNG NGHỆ THÔNG TIN**

**\*\*\*\*\*\*\*\*\*\*\***



**BÁO CÁO CUỐI KỲ**

**KIẾN TRÚC VÀ THIẾT KẾ PHẦN MỀM**

*Đề tài****: TÁI CHẾ LỐP XE CAO SU***

***DANH SÁCH THÀNH VIÊN***

***Nhóm 5***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **MSSV** | **HỌ VÀ TÊN** | **CHÚ THÍCH** | **CHỮ KÝ** | **ĐIỂM** |
| 1 | 20099441 | Nguyễn Tuấn Anh | Thành viên |  |  |
| 2 | 20070691 | Ngô Phi Long | Nhóm trưởng |  |  |

**LỜI MỞ ĐẦU**

Trong bối cảnh sự phát triển không ngừng của xã hội hiện nay, con người chúng ta hằng ngày đã và đang xả ra môi trường một lượng rác thải vô cùng lớn và nguy hại ra môi trường mà không thể nào tái chế hoặc tận dụng chúng, gây ảnh hưởng rất nghiêm trọng đến môi trường sống của chúng ta như rác thải, vải, chai lọ, độc hơn là cao su, pin, kim loại nặng,… Điều này thể hiện ngày càng rõ bằng các hiện tượng tự nhiên bất thường trong một thập kỷ trở lại đây: khói bụi, ô nhiễm, sự nóng lên toàn cầu, hạn hán, bang tan, mực nước biển dâng đang nhấn chìm các vùng trùng thấp, tỷ lệ người mắc các bệnh về đường hô hấp ngày càng cao,.. nếu không có biện pháp đúng đắn thì con người, các loài động vật khác trên trái đất này sẽ rơi vào tuyệt chủng là một tương lai không xa. Hiểu được thực trạng này, nhóm chúng em thấy rác thải cao su là một loại rác thải vô cùng độc hại và khó bị phân hủy nên chúng em sẽ đi sâu vào phân tích đề tài: “TÁI CHẾ LỐP XE CAO SU” nhằm tái chế lại các sản phẩm cao su mà con người vức đi nhằm xây dựng một ý thức tốt đẹp cho mọi người, góp phần bảo vệ sự trong lành của trái đất.

Em xin cảm ơn thầy Võ Văn Hải đã truyền đạt nhiều kiến thức, kinh nghiệm trong quá trình học tập, chia sẻ cho chúng em nhiều bài học hay, nhiều kiến thức bổ ích để chúng em hoàn thiện mình cũng như hoàn thành tốt bài báo cáo tốt hơn.

Với thời gian cho phép cũng như kinh nghiệm bản thân còn hạn chế, bài báo cáo này không thể tránh được những thiếu sót, em rất mong nhận được sự chia sẻ, góp ý của thầy để học hỏi, nâng cao kinh nghiệm của mình.

Cuối cùng, em xin kính chúc thầy dồi dào sức khỏe và thành công trong cuộc sống và sự nghiệp trồng người cao quý.

Chúng em xin chân thành cảm ơn!

**MỤC LỤC**

[**LỜI MỞ ĐẦU** 3](#_Toc167115455)

[**CHƯƠNG 1: GIỚI THIỆU CHUNG** 5](#_Toc167115456)

[**1.1. Tổng quan:** 5](#_Toc167115457)

[**1.2. Mục tiêu đề tài:** 5](#_Toc167115458)

[**1.3. Phạm vi đề tài:** 5](#_Toc167115459)

[**1.3.1. Phạm vi địa lý:** 5](#_Toc167115460)

[**1.3.2. Phạm vi công nghệ:** 5](#_Toc167115461)

[**1.4. Chức năng của hệ thống:** 6](#_Toc167115462)

[**1.5. Mô tả hệ thống:** 6](#_Toc167115463)

[**CHƯƠNG 2: CƠ SỞ LÝ THUYẾT** 8](#_Toc167115464)

[**2.1. ReactJS** 8](#_Toc167115465)

[**2.1.1. Tổng quan về ReactJS:** 8](#_Toc167115466)

[**2.1.2. Ưu, nhược điểm của ReactJS:** 8](#_Toc167115467)

[**2.2. NodeJS** 9](#_Toc167115468)

[**2.2.1. Tổng quan về NodeJS:** 9](#_Toc167115469)

[**2.2.2. Ưu, nhược điểm của NodeJS:** 9](#_Toc167115470)

[**2.3. MongoDB** 10](#_Toc167115471)

[**2.3.1. Tổng quát về MongoDB:** 10](#_Toc167115472)

[**2.3.2. Ưu, nhược điểm của MongoDB:** 10](#_Toc167115473)

[**2.4. Kiến trúc sử dụng trong dự án:** 11](#_Toc167115474)

[**2.4.1. Tổng quan về kiến trúc:** 11](#_Toc167115475)

[**2.4.2. Ưu, nhược điểm của kiến trúc:** 12](#_Toc167115476)

[**2.4.3. Môi trường phù hợp với kiến trúc:** 13](#_Toc167115477)

[**CHƯƠNG 3: THIẾT KẾ** 14](#_Toc167115478)

[**3.1. Usecase Tổng quát:** 14](#_Toc167115479)

[**3.3. Kiểm thử một vài usecase cơ bản:** 17](#_Toc167115480)

[**CHƯƠNG 4 : HIỆN THỰC** 18](#_Toc167115481)

[**4.1. Giao diện màn hình chính:** 18](#_Toc167115482)

[**4.2. Giao diện màn hình trang chủ** 18](#_Toc167115483)

[**4.3. Giao diện tạo đơn hàng thành công** 19](#_Toc167115484)

[**CHƯƠNG 5 : KẾT LUẬN** 20](#_Toc167115485)

[**5.1. Kết quả đạt được:** 20](#_Toc167115486)

[**5.2. Hạn chế của đề tài:** 20](#_Toc167115487)

[**5.3. Hướng phát triển:** 20](#_Toc167115488)

[**TÀI LIỆU THAM KHẢO** 21](#_Toc167115489)

# **CHƯƠNG 1: GIỚI THIỆU CHUNG**

## **1.1. Tổng quan:**

Trong thời đại công nghiệp hóa và phát triển của ngành vận tải, lốp xe cao su đã trở thành một phần quan trọng không thể thiếu của phương tiện giao thông. Tuy nhiên, việc xử lý rác thải lốp xe đã trở thành một thách thức lớn đối với môi trường và sức khỏe con người. Tại Việt Nam, vấn đề quản lý và xử lý rác thải cao su đặc biệt là lốp xe vẫn còn nhiều hạn chế, gây ra nhiều vấn đề về môi trường và an toàn giao thông.

Để giải quyết vấn đề này, việc tái chế lốp xe cao su trở thành một giải pháp quan trọng nhằm giảm thiểu tác động tiêu cực đến môi trường và tạo ra nguồn tài nguyên mới từ lốp xe tái chế. Đồng thời, việc nghiên cứu và áp dụng các công nghệ tái chế tiên tiến cũng đang là một xu hướng quan trọng trong ngành công nghiệp tái chế.

Kết hợp với môn học Kiến trúc và Thiết kế phần mềm, nhóm chúng em sẽ tạo ra một website sử dụng kiến trúc đã được học nhằm giúp cả bên mua và bên bán lốp xe một cách thuận tiện, nhanh chóng, quan trọng hơn hêt là giúp mọi người nâng cao ý thức tái chế rác thải độc hại, vừa giúp môi trường xanh – sạch – đẹp. Góp phần vào một tương lai tốt đẹp không chỉ cho chúng ta mà còn cho con cháu mai sau.

## **1.2. Mục tiêu đề tài:**

- Giúp người quản trị (người thu mua sản phẩm) có thể thu mua nhanh chóng, lên kế hoạch phân loại: tái chế lại hoặc tiêu hủy đi một cách dễ dàng và tối ưu. Rút ngắn thời gian, công sức so với việc thu mua truyền thống trước đây, giảm chi phí xây dựng các cửa hàng/điểm thu mua, giảm chi phí nhân viên, mặt bằng, vận hành,…

- Giúp người dùng (người bán sản phẩm) có thể dễ dàng thanh lí lốp xe cao su cũ của mình mà không cần phải di chuyển đến các cửa hàng/các điểm thu mua. Từ đó mang lại sự tiện lợi, nhanh chóng, thoải mái cho người dùng.

## **1.3. Phạm vi đề tài:**

**1.3.1. Phạm vi địa lý:**

Thành phố Hồ Chí Minh vì nơi đây là đầu tàu kinh tế của cả nước, là nơi đông đúc, có số lượng phương tiện giao thông lớn và lượng lốp xe thải ra cao, rất phù hợp cho đề tài.

### **1.3.2. Phạm vi công nghệ:**

* Về Sử dụng các công nghệ, máy móc có sẵn thực tế ở Việt Nam hiện nay: nghiền, nhiệt phân, tái sử dụng trực tiếp.
* Về trang web: sử dụng các kiến thức đã học ở những môn học ở những học kì trước: ReactJS, NodeJS, MongoDB.

## **1.4. Chức năng của hệ thống:**

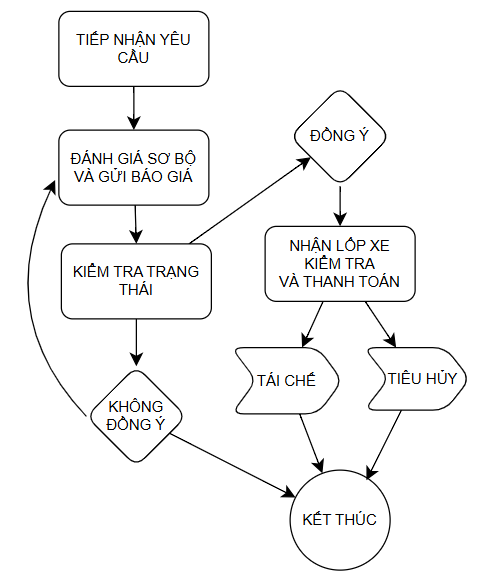
**1.4.1. Chức năng của người dùng:**

* Đăng ký
* Đăng nhập
* Tạo đơn hàng (thực hiện điền form để bên người mua liên hệ)
* Chấp nhận đơn hàng
* Đăng xuất

**1.4.2. Chức năng của người quản trị :**

* Đăng ký
* Đăng nhập
* Xem/Xóa/Cập nhật đơn hàng
* Báo giá sản phẩm trong đơn hàng
* Xem thống kê các phân loại: tái chế hoặc tiêu hủy
* Đăng xuất

## **1.5. Mô tả hệ thống:**



*Hình 1 – Hình mô tả hệ thống*

***1. Tiếp nhận yêu cầu:***

Hệ thống tiếp nhận yêu cầu của người dùng thông qua biểu mẫu trực tuyến mà người dùng đã điền. Thông tin bao gồm chi tiết về loại lốp xe họ cần bán và các yêu cầu cụ thể từ người dùng.

***2. Đánh giá sơ bộ và gửi báo giá:***

Đơn vị tiếp nhận sẽ tiến hành đối chiếu thông tin, đánh giá sơ bộ dưa trên những thông tin khách quan ban đầu mà người dùng cung cấp. Sau đó họ gửi báo giá và phản hồi cho người dùng hiểu.

***3. Kiểm tra trạng thái:***

Tới bước này, có 2 trường hợp xảy ra:

- Trường hợp 1: Người dùng Đồng ý: đi tiếp tới bước 4.

- Trường hợp 2: Người dùng Không đồng ý, quay lại bước 2: Đánh giá sơ bộ và gửi báo giá lại cho người dùng. Nếu người dùng và bên đơn vị tiếp nhận không thống nhất được kết quả cuối cùng thì quy trình một đơn hàng sẽ kết thúc tại đây.

***4. Nhận lốp xe, kiểm tra và thanh toán:***

Tới địa chỉ mà người dùng đã điền vào form ban đầu, tiến hành kiểm tra, đánh giá thực trạng và thực hiện thanh toán cho họ.

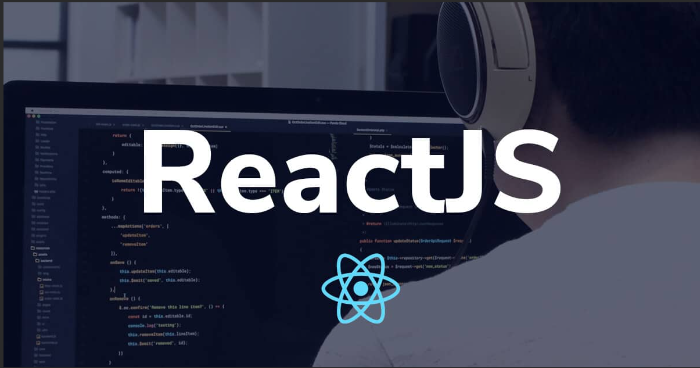
***5. Tái chế hoặc tiêu hủy:***

Sau khi tiến hành bước 4, đơn vị tiếp nhận sẽ tiến hành phân loại để thực hiện các mục đích tái sử dụng hoặc tiêu hủy theo quy trình nội bộ của công ty. Kết thúc quy trình một đơn hàng.

# **CHƯƠNG 2: CƠ SỞ LÝ THUYẾT**

## **2.1. ReactJS**

*Hình 2 – Hình background ReactJS*

****

### **2.1.1. Tổng quan về ReactJS:**

React (còn được gọi là Reactjs hay React.js) là một Thư viện javascript được tạo ra bởi sự cộng tác giữa Facebook và Instagram. Nó cho phép những nhà phát triển web tạo ra giao diện người dung nhanh chóng. Phần Views của Reactjs thường được hiển thị bằng việc chủ yếu dung các component mà chứa các component cụ thể hoặc các thẻ HTML. Một trong những đặc trưng duy nhất của Reactjs là việc render dữ liệu không những có thể thực hiện ở tầng server mà còn ở tầng client.

### **2.1.2. Ưu, nhược điểm của ReactJS:**

***- Ưu điểm:***

+ Reactjs giúp việc viết các đoạn code JS dễ dàng hơn: Nó dung cú pháp đặc biệt là JSX (Javascript mở rộng) cho phép ta trộn giữa code HTML và Javascript.

+ Làm việc với vấn đề test giao diện: Nó cực kì dễ để viết các test case giao diện vì virtual DOM được cài đặt hoàn toàn bằng JS.

+ Hiệu năng cao đối với các ứng dụng có dữ liệu thay đổi liên tục, dễ dàng cho bảo trì và sửa lỗi.

***- Nhược điểm :***

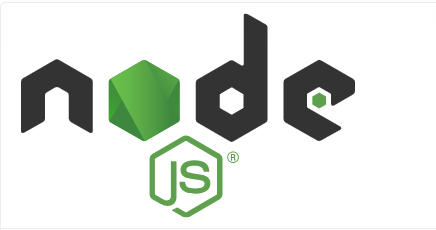
+ Reactjs chỉ phục vụ cho tầng View. React chỉ là View Library nó không phải là một MVC framework như những framework khác. Đây chỉ là thư viện của Facebook giúp render ra phần view. Vì thế React sẽ không có phần Model và Controller, mà phải kết hợp với các thư viện khác.

+ Tích hợp Reactjs vào các framework MVC truyền thống yêu cầu cần phải cấu hình lại.

+ React khá nặng nếu so với các framework khác React có kích thước tương tương với Angular (Khoảng 35kb so với 39kb của Angular). Trong khi đó Angular là một framework hoàn chỉnh. **[1]**

## **2.2. NodeJS**

*Hình 3 – Hình background NodeJS*

****

### **2.2.1. Tổng quan về NodeJS:**

Node.js được phát triển bởi Ryan Dahl vào năm 2009, là một môi trường runtime của JavaScript viết bằng C++, có mã nguồn mở và chạy trên đa nền tảng. Nó được dùng để chạy các ứng dụng web bên ngoài trình duyệt của client, mang lại nhiều ưu điểm so với các nền tảng phía máy chủ khác như: Java, PHP,… Hiện nay, Node.js đang được sử dụng rộng rãi bởi cả các ông lớn: Paypal, NASA, Walmart...

### **2.2.2. Ưu, nhược điểm của NodeJS:**

***- Ưu điểm:***

+ Đầu tiên ta phải kể đến là Node.js dễ học đối với mọi người.

+ Hiệu suất cao với các ứng dụng thời gian thực.

+ Khả năng mở rộng cao.

+ Giúp xây dựng các ứng dụng đa nền tảng, tiết kiệm chi phí.

+ Giảm thời gian chờ đợi phản hồi.

+ Có thể sử dụng như FullStack.

+ Có nguồn tài liệu và cộng đồng đủ lớn mạnh để hỗ trợ giải quyết các khó khăn.

***- Nhược điểm:***

+ Hiệu suất khi xử lý các tác vụ máy tính nặng bị giảm.

+ API đôi lúc không ổn định.

+ Việc tái sử dụng mã gặp khó khăn khi mô hình lập trình Node.js không đồng bộ.

+ Hỗ trợ thư viện không đầy đủ.**[2]**

## **2.3. MongoDB**

*Hình 4 – Hình background MongoDB*

****

**2.3.1. Tổng quát về MongoDB:**

MongoDB là một cơ sở dữ liệu NoSQL hướng đối tượng, đơn giản, linh động và có thể mở rộng. Nó dựa trên mô hình lưu trữ NoSQL document. Các đối tượng dữ liệu được lưu trữ dưới dạng các tài liệu riêng biệt bên trong một collection – thay vì lưu trữ dữ liệu vào các cột và hàng của cơ sở dữ liệu quan hệ truyền thống. Ngôn ngữ MongoDB triển khai một kho lưu trữ dữ liệu cung cấp hiệu suất cao, tính sẵn sàng cao và tự động mở rộng

### **2.3.2. Ưu, nhược điểm của MongoDB:**

***- Ưu điểm:***

+ Hiệu suất cao

+ Tính sẵn sàng cao

+ Khả năng mở rộng cao

+ Năng động – Không có lược đồ cứng nhắc.

+ Linh hoạt – thêm/xóa trường có ít hoặc không ảnh hưởng đến ứng dụng

+ Dữ liệu không đồng nhất

+ Biểu diễn dữ liệu trong JSON hoặc BSON

***- Nhược điểm:***

+ Giao dịch phức tạp

+ Không có chức năng hoặc thủ tục lưu trữ tồn tại nơi bạn có thể liên kết logic.**[3]**

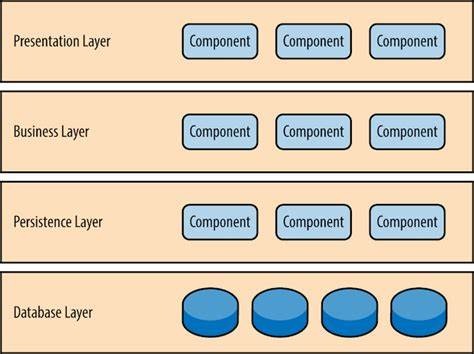
## **2.4. Kiến trúc sử dụng trong dự án:**

Nhóm Chúng em sử dụng **kiến trúc phân lớp (Layered Architecture)**.

### **2.4.1. Tổng quan về kiến trúc:**

Kiến trúc phân lớp là khung kiến trúc phổ biến và được sử dụng rộng rãi nhất trong phát triển phần mềm. Nó còn được gọi là kiến trúc n-tier và **mô tả một mô hình kiến trúc bao gồm một số lớp ngang riêng biệt hoạt động cùng nhau như một đơn vị phần mềm duy nhất**. Một lớp là một sự phân tách hợp lý của các thành phần hoặc mã.

*Hình 5 – Tổng quan kiến trúc phân lớp*



Được chia thành 4 lớp cơ bản:

- Presentation Layer (Lớp giao diện): là lớp nhận dữ liệu, thực hiện tương tác với người dùng.

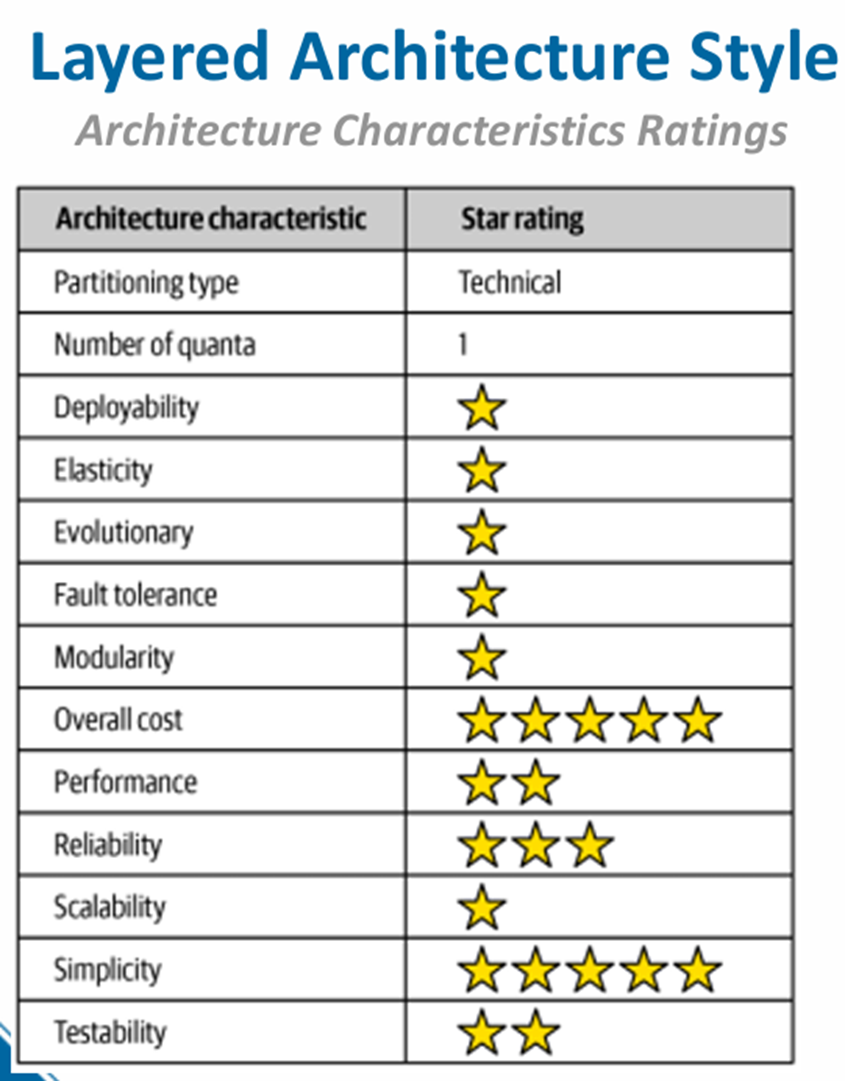
- Business Layer (Lớp xử lý): nhận dữ liệu và thực hiện xử lý logic.

- Persistence Layer (Lớp giao tiếp với dữ liệu): tương tác với dữ liệu

- Database Layer (Lớp dữ liệu): chứa dữ liệu cần xử lý

### **2.4.2. Ưu, nhược điểm của kiến trúc:**

*Hình 6 – Kiến trúc phân lớp*



* ***Ưu điểm:***

+ Framework này rất đơn giản và dễ học và thực hiện.

+ Có sự phụ thuộc giảm vì chức năng của mỗi lớp tách biệt với các lớp khác.

+ Kiểm thử dễ dàng hơn vì các thành phần riêng biệt, mỗi thành phần có thể được kiểm tra riêng lẻ.

+ Chi phí chung khá thấp.

* ***Nhược điểm:***

+ Khả năng mở rộng là khó khăn vì cấu trúc của khung không cho phép tăng trưởng.

+ Chúng có thể khó duy trì. Một sự thay đổi trong một lớp duy nhất có thể ảnh hưởng đến toàn bộ hệ thống vì nó hoạt động như một đơn vị duy nhất.

+ Có sự phụ thuộc lẫn nhau giữa các lớp vì một lớp phụ thuộc vào lớp phía trên nó để nhận dữ liệu.

+ Xử lý song song là không thể.

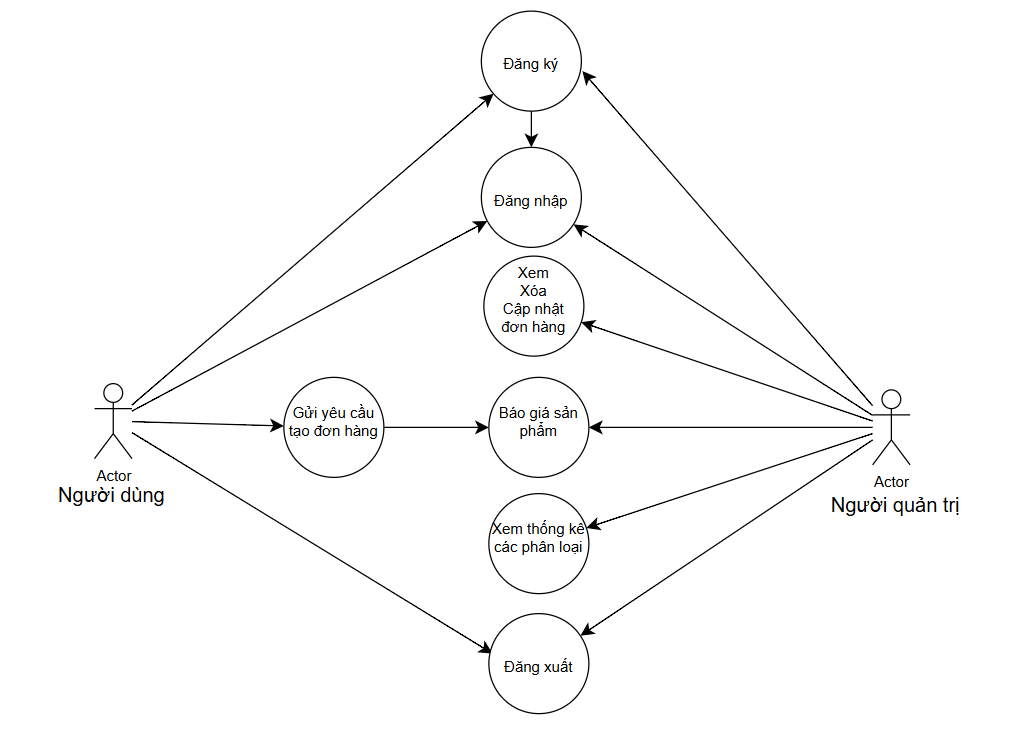
### **2.4.3. Môi trường phù hợp với kiến trúc:**

* Phù hợp với các trang web/ứng dụng nhỏ, đơn giản vì đây là framework đơn giản nhất.
* Các dự án hạn chế về thời gian và ngân sách rất chặt chẽ
* Hoặc các dự án mà nhà phát triền chưa có nhiều kiến thức phần mềm hoặc chưa biết quyết định dùng kiến trúc nào. **[4]**

# **CHƯƠNG 3: THIẾT KẾ**

## **3.1. Usecase Tổng quát:**

*Hình 7 – Usecase Tổng quát*

****

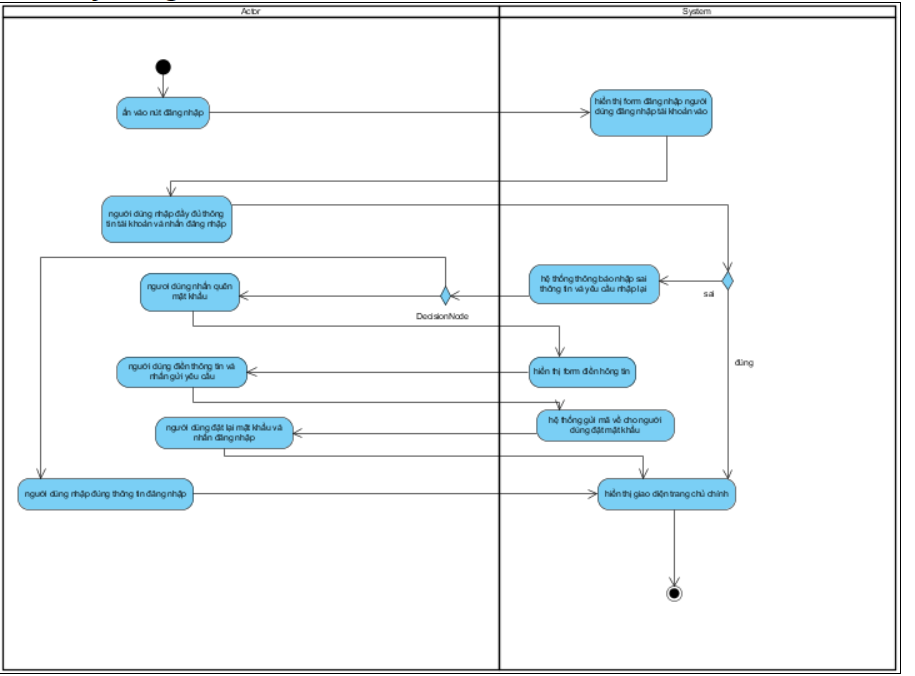
*Bảng 3.1. Danh sách tác nhân và mô tả hành động*

|  |  |
| --- | --- |
| **Tác nhân** | **Mô tả hành động có thể xảy ra của tác nhân** |
| **Người dùng** | * Đăng ký * Đăng nhập * Tạo đơn hàng (thực hiện điền form để bên người mua liên hệ) * Chấp nhận đơn hàng * Đăng xuất |
| **Người quản trị** | * Đăng ký * Đăng nhập * Xem/Xóa/Cập nhật đơn hàng * Báo giá sản phẩm trong đơn hàng * Xem thống kê các phân loại: tái chế hoặc tiêu hủy * Đăng xuất |

**3.2. Mô tả Usecase và vẽ các biểu đồ cơ bản:**

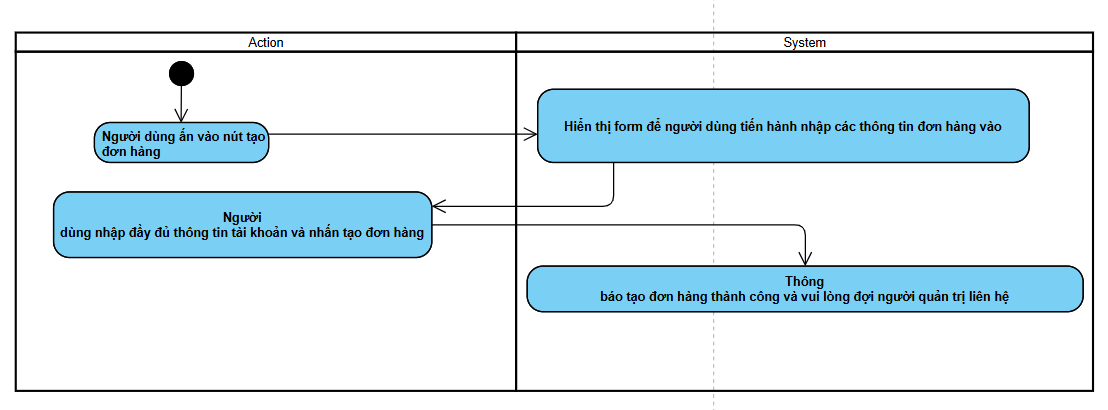
* **Usecase Đăng nhập**

|  |  |
| --- | --- |
| Tên use case: đăng nhập | |
| Mô tả sơ lược: giúp người dùng đăng nhập vào tài khoản | |
| Actor chính:người dùng | |
| Actor phụ:không | |
| Tiền điều kiên(pre-condition):đăng kí thành công | |
| Hậu điều kiện(post-condition:sau khi đăng nhập thành công vào giao diện của ứng dụng | |
| Actor | System |
| 1. ấn vào nút đăng nhập |  |
|  | 2.hiển thị form đăng nhập người dùng đăng nhập tài khoản vào |
| 3.người dùng nhập đầy đủ thông tin tài khoản và nhấn đăng nhập |  |
|  | 4.hiển thị giao diện trang chủ chính |
| Luồng sự kiện thay thế(alternate flow): | |
| Actor | System |
| 3.1 người dùng nhập sai thông tin |  |
|  | 3.1.2 hệ thống thông báo nhập sai thông tin và yêu cầu nhập lại |
| 3.1.3 người dùng nhập đúng thông tin đăng nhập |  |
|  | 5. quay lại bước 4 |
| Luồng sự kiện ngoại lệ(exception flow) | |
| Actor | System |
| 3.2 ngươi dùng nhấn quên mật khẩu |  |
|  | 3.3 hiển thị form điền hông tin |
| 3.4 người dùng điền thông tin và nhấn gửi yêu cầu |  |
|  | 3.5 hệ thống gửi mã về cho người dùng đặt mật khẩu |
| 3.6 người dùng đặt lại mật khẩu và nhấn đăng nhập |  |
|  | 3.7 quay lại bước 4 |



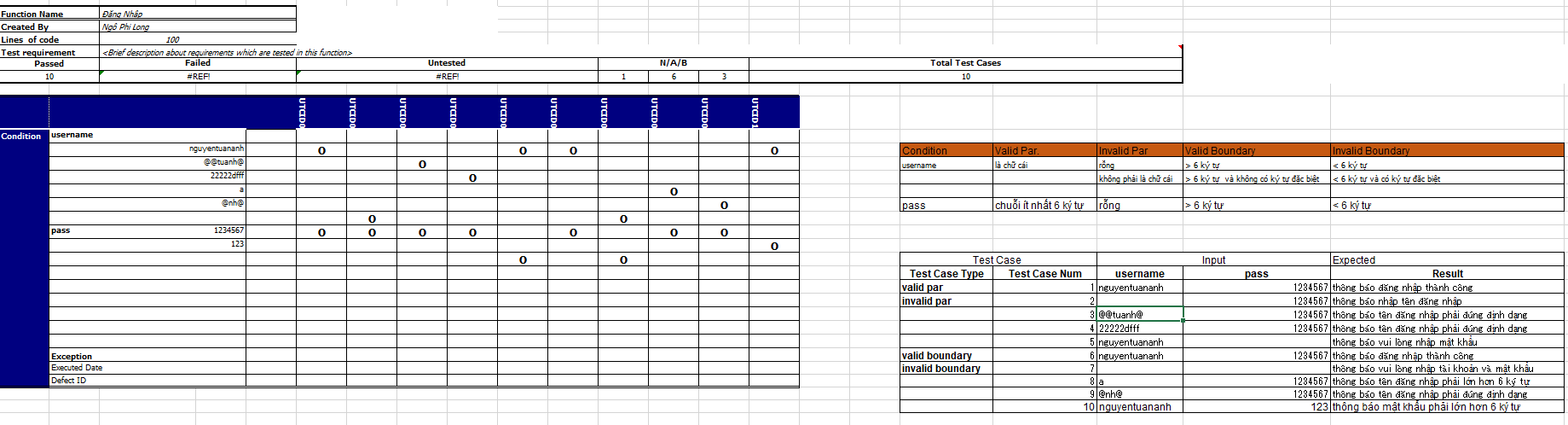
* **Usecase Tạo đơn hàng:**

|  |  |
| --- | --- |
| Tên use case: Tạo đơn hàng | |
| Mô tả sơ lược: Người dùng gửi thông tin hàng cần bán tới bên mua | |
| Actor chính: Người dùng | |
| Actor phụ: Không | |
| Tiền điều kiên(pre-condition): đăng nhập thành công | |
| Hậu điều kiện(post-condition: sau khi đăng nhập thành công vào giao diện của ứng dụng | |
| Actor | System |
| 1. Ấn vào nút tạo đơn hàng |  |
|  | 2. Hiển thị form đăng nhập người dùng đăng nhập tài khoản vào. Trong đó có: tên lốp xe, mã hiệu, tình trạng, hình ảnh, họ tên, địa chỉ, số điện thoại và các thông tin khác có liên quan |
| 3. Người dùng nhập đầy đủ thông tin tài khoản và nhấn tạo đơn hàng |  |
|  | 4. Thông báo tạo đơn hàng thành công và vui lòng đợi người quản trị liên hệ |
| Luồng sự kiện thay thế(alternate flow): | |
| Actor | System |
|  | 4.1. Hệ thống thông báo: người quản trị không tìm thấy thông tin chính xác của bạn đã đăng ký |
| 4.2. Người nhận nhấn OK |  |
|  | 4.3. Hệ thống quay lại bước 2 |

****

## **3.3. Kiểm thử một vài usecase cơ bản:**

* **Usecase Đăng nhập**

****

# **CHƯƠNG 4 : HIỆN THỰC**

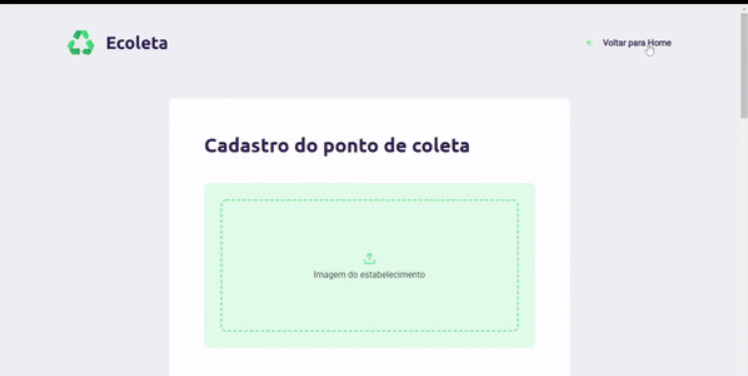
## **4.1. Giao diện màn hình chính:**

Tạo đơn hàng (Kéo nút xanh qua phải)

****

## **4.2. Giao diện màn hình trang chủ**

* Chụp hình ảnh của lốp xe cần bán

****

* Form điền các thông tin:

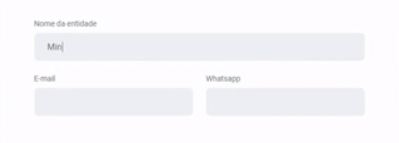
+ Tên, loại, trạng thái của lốp xe cao cu

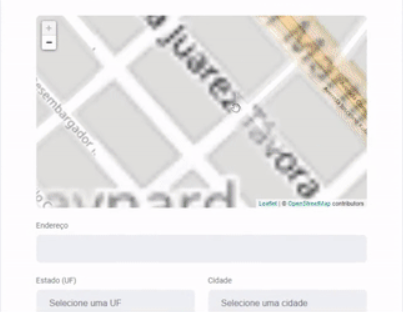
+ Giá cả muốn bán (có thể bỏ trống)

+ Họ và tên

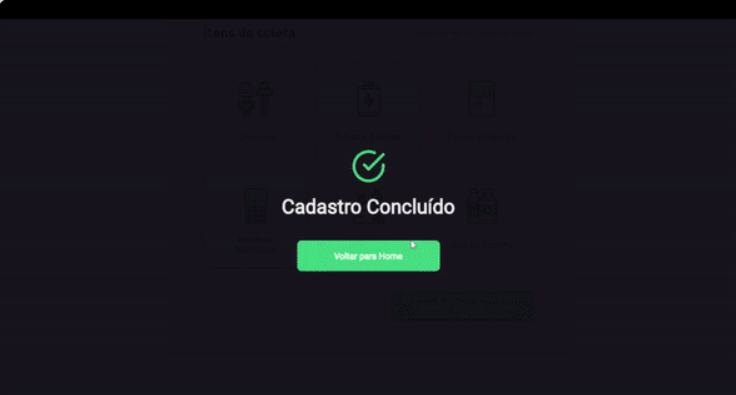
+ Số điện thoại

+ Vị trí đang ở

****

****

## **4.3. Giao diện tạo đơn hàng thành công**

****

# **CHƯƠNG 5 : KẾT LUẬN**

## **5.1. Kết quả đạt được:**

* Đã tạo ra được hệ thống cơ bản đáp ứng nhu cầu bán lốp xe cao su không sử dụng nữa cho người dùng. Trong hệ thống có các chức năng như: Trang chủ, Tạo đơn hàng, Thông báo tạo thành công.
* Màu sắc, giao diện đơn giản, phù hợp với mọi lứa tuổi mang lại sự thuận tiện đối với đại đa số người sử dụng.
* Hệ thống hoạt động ổn định

## **5.2. Hạn chế của đề tài:**

* Chưa làm được hệ thống bên phần người quản trị, mọi thứ diễn ra theo cơ chế truyền thống, nghe gọi
* Chưa làm đăng ký, đăng nhập cho khách hàng. Họ chỉ cần vào bán là xong  
  Chưa có đăng xuất
* Chỉ mới thanh toán bằng tiền mặt, chưa có thanh toán bằng chuyển khoản

## **5.3. Hướng phát triển:**

Nhóm sẽ cố gắng khắc phục và sửa đổi cho toàn diện hơn như :

* Bổ sung tính năng đăng ký, đăng nhập cho người dùng
* Bổ sung tính năng đăng xuất
* Bổ sung tính năng thanh toán để thuận tiện cho người dùng lẫn đơn vị tiếp nhận sản phẩm

**TÀI LIỆU THAM KHẢO**

**[1]:** [ReactJS - Ưu điểm và nhược điểm (viblo.asia)](https://viblo.asia/p/reactjs-uu-diem-va-nhuoc-diem-V3m5WzexlO7)

**[2]:** [Những ưu và nhược điểm của Node.js - Vietdev](https://vietdev.com/community/articles/nhung-uu-va-nhuoc-diem-cua-nodejs-sTTqpa/)

**[3]:** [Ưu và nhược điểm của MongoDB (hybrid-technologies.vn)](https://jobs.hybrid-technologies.vn/blog/uu-nhuoc-diem-mongodb/#:~:text=%C6%AFu%20%C4%91i%E1%BB%83m%20c%E1%BB%A7a%20MongoDB%20Document%20oriented%20Hi%E1%BB%87u%20su%E1%BA%A5t,%C3%ADt%20ho%E1%BA%B7c%20kh%C3%B4ng%20%E1%BA%A3nh%20h%C6%B0%E1%BB%9Fng%20%C4%91%E1%BA%BFn%20%E1%BB%A9ng%20d%E1%BB%A5ng)

**[4]:** [Kiến trúc phân lớp | Baeldung về Khoa học Máy tính](https://www.baeldung.com/cs/layered-architecture)