**Trần Quốc Đại TÊN ĐỀ TÀI: FreshMarket 2020**

**ĐẠI HỌC ĐÀ NẴNG**

**TRƯỜNG ĐẠI HỌC BÁCH KHOA**

**KHOA CÔNG NGHỆ THÔNG TIN**

**ĐỒ ÁN TỐT NGHIỆP**

**NGÀNH: CÔNG NGHỆ THÔNG TIN**

**CHUYÊN NGÀNH: CÔNG NGHỆ PHẦN MỀM**

**ĐỀ TÀI:**

**Xây dựng website buôn bán nông sản**

**FreshMarket**

Người hướng dẫn: **ThS PHAN CHÍ TÙNG**

Sinh viên thực hiện: **TRẦN QUỐC ĐẠI**

Số thẻ sinh viên: **102150089**

Lớp**: 15T2**

**Đà Nẵng, 06/2020**

**ĐẠI HỌC ĐÀ NẴNG**

**TRƯỜNG ĐẠI HỌC BÁCH KHOA**

**KHOA CÔNG NGHỆ THÔNG TIN**

**ĐỒ ÁN TỐT NGHIỆP**

**NGÀNH: CÔNG NGHỆ THÔNG TIN**

**CHUYÊN NGÀNH: CÔNG NGHỆ PHẦN MỀM**

**ĐỀ TÀI:**

**Xây dựng website buôn bán nông sản**

**FreshMarket**

Người hướng dẫn: **ThS PHAN CHÍ TÙNG**

Sinh viên thực hiện: **TRẦN QUỐC ĐẠI**

Số thẻ sinh viên: **102150089**

Lớp**: 15T2**

**Đà Nẵng, 06/2020**

**NHẬN XÉT CỦA NGƯỜI HƯỚNG DẪN**

…………………………………………………………………………………………...

…………………………………………………………………………………………...

…………………………………………………………………………………………...

…………………………………………………………………………………………...

…………………………………………………………………………………………...

…………………………………………………………………………………………...

…………………………………………………………………………………………...

…………………………………………………………………………………………...

…………………………………………………………………………………………...

…………………………………………………………………………………………...

…………………………………………………………………………………………...

…………………………………………………………………………………………...

…………………………………………………………………………………………...

…………………………………………………………………………………………...

…………………………………………………………………………………………...

…………………………………………………………………………………………...

…………………………………………………………………………………………...

…………………………………………………………………………………………...

…………………………………………………………………………………………...

…………………………………………………………………………………………...

…………………………………………………………………………………………...

…………………………………………………………………………………………...

…………………………………………………………………………………………...

…………………………………………………………………………………………...

…………………………………………………………………………………………...

…………………………………………………………………………………………...

…………………………………………………………………………………………...

…………………………………………………………………………………………...

**NHẬN XÉT CỦA NGƯỜI PHẢN BIỆN**

…………………………………………………………………………………………...

…………………………………………………………………………………………...

…………………………………………………………………………………………...

…………………………………………………………………………………………...

…………………………………………………………………………………………...

…………………………………………………………………………………………...

…………………………………………………………………………………………...

…………………………………………………………………………………………...

…………………………………………………………………………………………...

…………………………………………………………………………………………...

…………………………………………………………………………………………...

…………………………………………………………………………………………...

…………………………………………………………………………………………...

…………………………………………………………………………………………...

…………………………………………………………………………………………...

…………………………………………………………………………………………...

…………………………………………………………………………………………...

…………………………………………………………………………………………...

…………………………………………………………………………………………...

…………………………………………………………………………………………...

…………………………………………………………………………………………...

…………………………………………………………………………………………...

…………………………………………………………………………………………...

…………………………………………………………………………………………...

…………………………………………………………………………………………...

…………………………………………………………………………………………...

…………………………………………………………………………………………...

…………………………………………………………………………………………...

…………………………………………………………………………………………...

# **TÓM TẮT**

Tên đề tài: Xây dựng hệ thống website buôn bán nông sản FreshMarket

Sinh viên thực hiện: TRẦN QUỐC ĐẠI

Số thẻ SV: 1021500089

Lớp: 15T2

Đề tài: Buôn bán nông sản FreshMarket

FreshMarket là một hệ thống chuyên buôm bán,làm cầu nối cho việc buôn bán các mặt hàng nông sản giữa người cung cấp và người tiêu thụ sản phẩm.

Như ta đã biết, nông nghiệp luôn là ngành kinh tế có vị trí quan trọng với khoảng 70% dân số (tính theo năm 2019) làm nông nghiệp của nước ta. Từ một nước nông nghiệp nghèo nàn lạc hậu, trong những năm qua nhờ có chủ trương, đường lối đúng đắn của Nhà nước, nông nghiệp Việt Nam đã có nhiều chuyển biến tích cực.Nhưng cũng đặt ra thách thức làm sao sẽ tạo ra một kênh phân phối uy tín,nhanh chống và đáng tin cậy.Để giải quyết bài toán này thương mại điện tử ra đời,nó giải quyết được hoàn toàn vấn đề hậu cần và phân phối.

FreshMarket là một trang thương mai như vậy,nó là nơi để các nhà cung cấp các nguồn nông sản có thể trưng bày các mặt hàng của mình và có thể xem có các mặt hàng của các nhà cung cấp khác. FreshMarket cũng là nơi để người tiêu dùng có thể tìm kiếm,so sánh,mua các mặt hàng nông sản mà mình mong muốn.

|  |  |
| --- | --- |
| ĐẠI HỌC ĐÀ NẴNG  **TRƯỜNG ĐẠI HỌC BÁCH KHOA**  KHOA CÔNG NGHỆ THÔNG TIN | **CỘNG HÒA XÃ HÔI CHỦ NGHĨA VIỆT NAM**  Độc lập - Tự do - Hạnh phúc |

# **NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP**

Họ tên sinh viên: : **TRẦN QUỐC ĐẠI** Số thẻ sinh viên: 102150089

Lớp: 15T2 Khoa: Công nghệ thông tin Ngành: Công nghệ phần mềm

1. *Tên đề tài đồ án:*

Xây dựng hệ thống website buôn bán nông sản FreshMarket.

1. *Đề tài thuộc diện:*  *Có ký kết thỏa thuận sở hữu trí tuệ đối với kết quả thực hiện*
2. *Các số liệu và dữ liệu ban đầu:*

* Thông tin cá nhân của user trong hệ thống
* Thông tin các nguồn hàng nông sản

1. *Nội dung các phần thuyết minh và tính toán:*

* Cấu hình hệ thống: thuật toán so sánh, blocking
* Tạo mới, quản lý tài khoản cho các user
* Tạo mới, quản lý các bài đăng về sản phẩm nông sản
* Giải quyết bài toán đặt hàng,quản lí kho hàng,Newfeed,….

1. *Họ tên người hướng dẫn:* **Th.S PHAN CHÍ TÙNG**
2. *Ngày giao nhiệm vụ đồ án:*  *……../……./2020*
3. *Ngày hoàn thành đồ án: ……../……./2020*

|  |  |
| --- | --- |
|  | *Đà Nẵng, ngày tháng năm 2020* |
| **Trưởng Bộ môn:**TS.Nguyễn Văn Hiệu | **Người hướng dẫn** |

Th.S Phan Chí Tùng

# **LỜI NÓI ĐẦU**

Đầu tiên, em xin gửi lời cảm ơn chân thành tới Th.S Phan Chí Tùng đã giúp đỡ em hoàn thành tốt đề tài đồ án tốt nghiệp này. Trong quá trình hướng dẫn em làm đồ án, thầy đã chỉ bảo và hướng dẫn tận tình cho em những lý thuyết, cũng như các kỹ năng trong lập trình, cách giải quyết các vấn đề trong đề tài. Thầy cũng là người truyền động lực, đề xuất cho em các ý tưởng mới mẽ để giúp em hoàn thành tối đề tài đồ án.

Và nhân đây,Em cũng muốn gửi lời cảm ơn và ghi nhớ sự dạy dỗ tận tình của các thầy cô khoa Công nghệ thông tin – Trường Đại học Bách khoa – Đại học Đà Nẵng.

Trong quá trình hoàn thành đồ án, em đã rất cố gắng thực hiện hết khả năng trong thời gian cho phép nhưng chắc chắn sẽ không tránh khỏi những thiếu sót. Em rất mong nhận sự góp ý, chỉ bảo của quý thầy cô.

# **CAM ĐOAN**

Tôi xin cam đoan:

1. Nội dung trong đồ án này là do tôi thực hiện dưới sự hướng dẫn trực tiếp của Th.S Phan Chí Tùng
2. Các tham khảo dùng trong đồ án đều được trích dẫn rõ ràng tên tác giả, tên công trình, thời gian, địa điểm công bố
3. Nếu có những sao chép không hợp lệ, vi phạm, tôi xin chịu hoàn toàn trách nhiệm.

Sinh viên thực hiện

Trần Quốc Đại

**MỤC LỤC**

[TÓM TẮT](#_Toc43327454)

[NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP](#_Toc43327455)

[LỜI NÓI ĐẦU i](#_Toc43327456)

[CAM ĐOAN ii](#_Toc43327457)

[DANH SÁCH CÁC BẢNG, HÌNH VẼ v](#_Toc43327458)

[DANH SÁCH CÁC KÝ HIỆU, CHỮ VIẾT TẮT vii](#_Toc43327459)

[MỞ ĐẦU viii](#_Toc43327460)

[Chương 1: GIỚI THIỆU CHUNG 1](#_Toc43327461)

[**1.1.** **Giới thiệu chung về đề tài:** 1](#_Toc43327462)

[***1.1.1.*** ***Đề tài:*** 1](#_Toc43327463)

[***1.1.2.*** ***Lợi ích mang lại của đề tài:*** 2](#_Toc43327464)

[**1.2. Các công nghệ đã được sử dụng:** 4](#_Toc43327465)

[***1.2.1. Back end***: 4](#_Toc43327466)

[***1.2.2. Front end:*** 4](#_Toc43327467)

[***1.2.3 UI/UX:*** 5](#_Toc43327468)

[***1.2.4 Hệ quản trị cơ sở dữ liệu:*** 6](#_Toc43327469)

[***1.2.4 Hệ quản ảnh Cloudinary:*** 8](#_Toc43327470)

[Chương 2: PHÂN TÍCH HỆ THỐNG 10](#_Toc43327471)

[**2.1. Giới thiệu chung** 10](#_Toc43327472)

[***2.1.1. Mô tả:*** 10](#_Toc43327473)

[***2.1.2. Chức năng :*** 10](#_Toc43327474)

[***2.1.3. Đối tượng:*** 10](#_Toc43327475)

[***2.1.4. Môi trường hoạt động:*** 10](#_Toc43327476)

[**2.2. Actors:** 11](#_Toc43327477)

[**2.2. Sơ đồ Use cases:** 11](#_Toc43327478)

[**2.3. Sơ đồ hoạt động:** 18](#_Toc43327479)

[**2.4. Sơ đồ tuần tự:** 19](#_Toc43327480)

[**2.5. Từ điển dữ liệu:** 22](#_Toc43327481)

[Chương 3: PHÁT TRIỂN VÀ TRIỂN KHAI HỆ THỐNG 27](#_Toc43327482)

[**3.1. Phát triển hệ thống:** 27](#_Toc43327483)

[**3.1.1. Triển khai hệ thống REST-API:** 27](#_Toc43327484)

[**3.1.2. Cơ sở dữ liệu:** 29](#_Toc43327485)

[**3.2. Các công nghệ sử dụng:** 31](#_Toc43327486)

[**3.3. Triển khai hệ thống:** 31](#_Toc43327487)

[***3.3.1. Triển khai back end:*** 31](#_Toc43327488)

[***3.2.2. Triển khai frontend:*** 34](#_Toc43327489)

[**3.4. Kết quả đã thực hiện:** 35](#_Toc43327490)

[***3.4.1. Màn hình đăng nhập:*** 36](#_Toc43327491)

[***3.4.2. Màn hình quản lí của admin:*** 36](#_Toc43327492)

[***3.4.3. Màn hình xem màn danh sách bài đăng(sản phẩm):*** 37](#_Toc43327493)

[***3.4.4. Màn hình viết bài đăng:*** 38](#_Toc43327494)

[***3.4.5. Màn hình xem thông tin chi tiết về sản phẩm:*** 38](#_Toc43327495)

[***3.4.6. Màn hình quản lí bài đăng của người dùng:*** 39](#_Toc43327496)

[***3.4.7. Màn hình quản lí đơn hàng của người dùng:*** 39](#_Toc43327497)

[KẾT LUẬN 40](#_Toc43327498)

[TÀI LIỆU THAM KHẢO 41](#_Toc43327499)

[PHỤ LỤC 1 1](#_Toc43327500)

# **DANH SÁCH CÁC BẢNG, HÌNH VẼ**

* **Danh sách bảng:**

[Bảng 2.1 Actors 11](#_Toc43326222)

[Bảng 2.4.1 Avatar 22](#_Toc43326223)

[Bảng 2.4.2 CalculationUnit 22](#_Toc43326224)

[Bảng 2.4.3 CommentPost 22](#_Toc43326225)

[Bảng 2.4.4 ImagePost 23](#_Toc43326226)

[Bảng 2.4.5 Post 23](#_Toc43326227)

[Bảng 2.4.6 RoleUser 24](#_Toc43326228)

[Bảng 2.4.7 Province 24](#_Toc43326229)

[Bảng 2.4.8 Purchase 25](#_Toc43326230)

[Bảng 2.4.9 RatePost 25](#_Toc43326231)

[Bảng 2.4.10 StatusPurchase 25](#_Toc43326232)

[Bảng 2.4.11 User 26](#_Toc43326233)

[Bảng 2.4.12 TransportType 26](#_Toc43326234)

* **Danh sẽ hình ảnh:**

[Hình 1.1 Logo FreshMarket 2](#_Toc43326235)

[Hình 1.2 Kiến trúc của Spring Framework 4](#_Toc43326236)

[Hình 1.3 Mô hình của AngularJS 5](#_Toc43326237)

[Hình 1.4 Hình ảnh Boostrap 4 5](#_Toc43326238)

[Hình 1.5 Hình ảnh Angular Materal 6](#_Toc43326239)

[Hình 1.6 Hình ảnh MySQL 7](#_Toc43326240)

[Hình 1.7 Kiến trúc của MySQL 8](#_Toc43326241)

[Hình 1.8 Kiến trúc của Cloudinary 9](#_Toc43326242)

[Hình 2.2.1 Sơ đồ Use-case cho người quản lý 12](#_Toc43326243)

[Hình 2.2.2 Sơ đồ Use-case cho admin quản lý user 12](#_Toc43326244)

[Hình 2.2.3 Sơ đồ Use-case cho admin quản lý bài đăng 13](#_Toc43326245)

[Hình 2.2.4 Sơ đồ Use-case cho admin quản lý các đơn hàng 13](#_Toc43326246)

[Hình 2.2.5 Sơ đồ Use-case cho admin quản lý comments 14](#_Toc43326247)

[Hình 2.2.6 Sơ đồ Use-case cho user 15](#_Toc43326248)

[Hình 2.2.7 Sơ đồ Use-case cho user quản lý thông tin cá nhân 16](#_Toc43326249)

[Hình 2.2.8 Sơ đồ Use-case cho user quản lý bài đăng 16](#_Toc43326250)

[Hình 2.2.9 Sơ đồ Use-case cho user quản lý đơn hàng 17](#_Toc43326251)

[Hình 2.3 Sơ đồ hoạt động 18](#_Toc43326252)

[Hình 3.1.1 Mô hình hoạt động của Rest API 27](#_Toc43326259)

[Hình 3.1.2 Mô hình client call API 28](#_Toc43326260)

[Hình 3.1.3 ER Diagram của FreshMarket 30](#_Toc43326261)

[Hình 3.3.1 Màn hình đăng ký Heroku 32](#_Toc43326262)

[Hình 3.3.2 Đăng nhập Heroku CLI thành công 33](#_Toc43326263)

[Hình 3.3.3 Màn hình thêm mới app 33](#_Toc43326264)

[Hình 3.3.4 Thêm as-on heroku-postgresql 34](#_Toc43326265)

[Hình 3.3.5 Cấu hình datasource 34](#_Toc43326266)

[Hình 3.3.6 Toàn bộ file server.js 35](#_Toc43326267)

[Hình 3.4.1 Màn hình Login 36](#_Toc43326268)

[Hình 3.4.2 Màn hình quản lý của admin 37](#_Toc43326269)

[Hình 3.4.3 Màn hình danh sách bài đăng 37](#_Toc43326270)

[Hình 3.4.4 Màn hình viết bài đăng 38](#_Toc43326271)

[Hình 3.4.5 Màn hình chi tiết về sản phẩm 39](#_Toc43326272)

[Hình 3.4.6 Màn hình quản lý bài đăng của người đăng 39](#_Toc43326273)

[Hình 3.4.7 Màn hình quản lý đơn hàng của người dùng 39](#_Toc43326274)

# **DANH SÁCH CÁC KÝ HIỆU, CHỮ VIẾT TẮT**

KÝ HIỆU:

…….……..........................................................................................................................

…….……..........................................................................................................................

…….……..........................................................................................................................

…….……..........................................................................................................................

…….……..........................................................................................................................

…….……..........................................................................................................................

CHỮ VIẾT TẮT:

* FreshMarket: hệ thống website buôn bán nông sản FreshMarket
* Java EE: Java Enterprise Edition
* CSDL: Cơ sở dữ liệu
* SQL: Structure Query Language
* CLI: Command Line Interface
* UI:User Interface

# **MỞ ĐẦU**

Mục đích thực hiện đề tài:

* Củng cố, hoàn thiện các kiến thức và kỹ năng đã trau dồi suốt thời gian học tập
* Nắm bắt những công nghệ mới, đang là xu thế trong ngành công nghệ thông tin
* Xây dựng một hệ thống có thể mang những lợi ích thiết thực trong cuộc sống

Mục tiêu đề tài:

* FreshMarket – nhằm mang lại một người buôm bán nông sản cho các nhà cung cấp,nhà tiêu dùng Việt Nam
* Hệ thông có đầy đủ chức năng và đáp ứng tốt các yêu cầu phi chức năng

Phạm vi và đối tượng nghiên cứu:

* Tham khảo các hệ thống tương tự đã có trên các trong và ngoài nước, các phần mềm mã nguồn mở
* Các bài toán liên quan đến thương mại điện tử.

Phương pháp nghiên cứu:

* Tìm hiểu, nghiên cứu các tài liệu, hệ thống mã nguồn mở có sẵn trên internet
* Học cách sử dụng các công nghệ như Spring, Angular

Cấu trúc của đồ án tốt nghiệp:

* Chương 1 Giới thiệu chung
* Chương 2 Phân tích hệ thống
* Chương 3 Phát triển hệ thống và triển khai

# 

# **Chương 1: GIỚI THIỆU CHUNG**

## **Giới thiệu chung về đề tài:**

* + 1. ***Đề tài:***

FreshMarket là một hệ thống chuyên buôm bán,làm cầu nối cho việc buôn bán các mặt hàng nông sản giữa người cung cấp và người tiêu thụ sản phẩm.

FreshMarket nhằm mục đích cung cấp một nơi thuận tiện,có nhiều sự lựa chọn cho việc phân phối các mặt hàng nông sản Việt.

Khi một sản phẩm được user tạo ra trên hệ thống,nó được xem là một mặt hàng để người mua(người tiêu dùng) có thể đặt mua khi có thể tạo ra một đơn hàng với người cung cấp sản phẩm đó.Sau đó,người quản lí đơn hàng có thể liên lạc với khách hàng bằng thông tin của khách hang để lại để chọn phương thức nhận hàng.Kết thúc là sự xác nhận đơn hàng đã được giao của người quản lí đơn hang.

Để tăng độ tin cậy của bài đăng,hệ thống cho người dùng có thể đánh giá,viết nhận xét của người dùng thông qua các lần mua.

Hệ thống có thể sắp xếp (news feed) các bài đăng dựa vào số điểm đánh giá của các user có trong hệ thống và số sản phẩm bán được của bài đăng so với thời gian trôi qua của bài đăng.



Hình 1.1 Logo FreshMarket

* + 1. ***Lợi ích mang lại của đề tài:***

Với sự phát triển của thương mại điện tử đang phát triển mạnh mẽ và là một phần không thể thiếu trong hoạt động của doanh nghiệp trên toàn thế giới. Quy mô thị trường thương mại điện tử của Việt Nam với xuất phát điểm khá khiêm tốn, khoảng 4 tỷ USD vào năm 2015, nhưng tăng trưởng khá nhanh và ổn định trong những năm 2016, 2017, 2018, nếu tiếp tục giữ mức tăng trưởng 30%/năm thì thị trường ước tính đạt 13 tỷ USD vào năm 2020. Với tốc độ tăng trưởng nhanh và ổn định như vậy, kết hợp với xu hướng chuyển dịch thói quen mua sắm của người tiêu dùng, có thể khẳng định thương mại điện tử sẽ ngày càng chiếm lĩnh những vị thế quan trọng hơn đối với các doanh nghiệp. Vì thế,những lợi ích mang lại doanh nghiệp,nông dân,các hợp tác xã,…khi đưa sản phẩm nông nghiệp của mình khi tham gia thị trường thương mại điện tử là một điều tất yếu để tăng trưởng năng suất tiêu thụ và đó là một cách hiệu quả để phát triển nền nông nghiệp Việt Nam.

FreshMarket là một sản phẩm dựa trên ý tưởng đó,nó cho phép tạo ra một cầu nối giữa cung-cầu hàng hóa trong nền nông sản đang tiến lên hiện đại hóa. Nhờ đó,các user có trong hệ thống có thể giao dịch các sản phẩm của mình mà bỏ đi các khâu trung gian,qua đó giúp giảm chi phí vận chuyển và giảm giá thành của sản phẩm,…với mục đích cuối cùng sẽ giảm giá thành nông sản,tăng sức cận tranh với các sản phẩm nông sản khác trên thị trường.

FreshMarket là một nơi để tìm sản phẩm nông sản phù hợp với nhau cầu cá nhân,tập thể,doanh nghiệp,…thông qua việc so sánh các bài đăng,các thông tin của người đăng sản phẩm

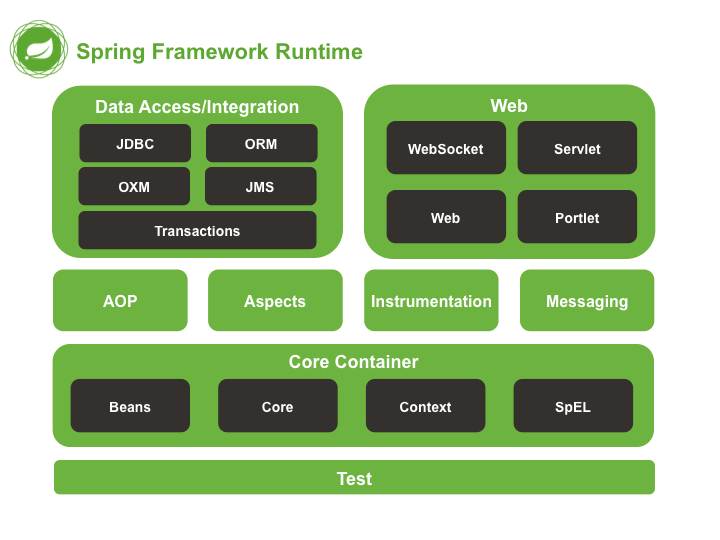
Bên cạnh đó FreshMarket còn là một kênh giúp người dùng có thể quảng bá sản phẩm của mình thông qua các miêu tả các đánh giá của các người dùng khác.

## **1.2. Các công nghệ đã được sử dụng:**

### ***1.2.1. Back end***:

Sử dụng ngôn ngữ Java với Framework Spring Boot

Spring Framework là một bộ khung ứng dụng và bộ chứa đảo ngược điều khiển cho nền tảng Java được phát hành vào 1 tháng 10 năm 2002(với bản ổn định nhất là bản 4.3.4 được phát hành 7 tháng 11, 2016). Chức năng tính của bộ khung này có thể áp dụng cho bất kỳ ứng dụng Java nào, tuy vậy, nhiều bản mở rộng dành cho việc xây dựng ứng dụng nền web cũng được phát triển trên nền tảng Java EE. Tuy bộ khung không ấn định lên một cụ thể nào, Spring Framework đã trở nên phổ biến trong cộng đồng Java như một phương án song hành, thay thế, hoặc thậm chí bổ sung cho mô hình Enterprise JavaBeans (EJB). Đây là bộ khung mã nguồn mở.



Hình 1.2 Kiến trúc của Spring Framework

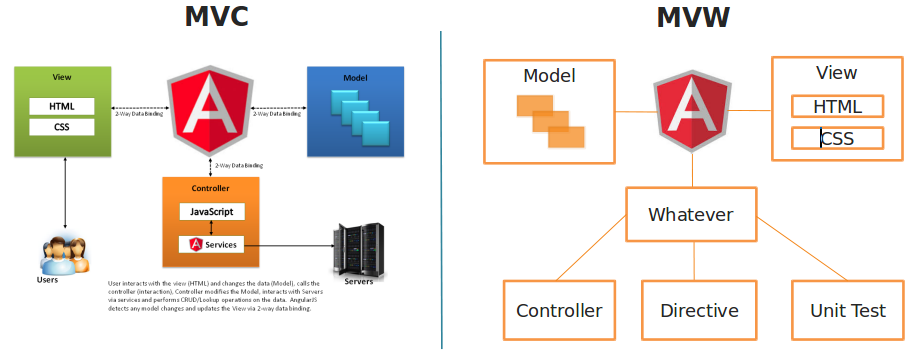
### ***1.2.2. Front end:***

Sử dụng ngôn ngữ JavaScript với Framework AngularJS

AngularJS là một thư viện JavaScript nhằm đơn giản hóa việc phát triển giao diện người dùng.

Được phát triển bởi Google và bản release đầu tiên trên thế giới vào 20 tháng 10 năm 2010(phiên bản mới nhất là bản 1.7.9 được phát hành vào 19 tháng 11 năm 2019) và nhằm mục đích để đơn giản hóa cả phát triển và [thử nghiệm](https://vi.wikipedia.org/wiki/Ki%E1%BB%83m_th%E1%BB%AD_ph%E1%BA%A7n_m%E1%BB%81m" \o "Kiểm thử phần mềm) của các ứng dụng bằng cách cung cấp một framework với kiến trúc cho client side đó là MVC và MVVM.

Mục tiêu của Angular chính là đơn giản để phát triển. Tất cả trạng thái đều được tập trung tại một thời điểm, bằng cách chia giao diện người dùng thành tập hợp các thành phần (components)



Hình 1.3 Mô hình của AngularJS

### ***1.2.3 UI/UX:***

- Sử dụng Boostrap 4:



Hình 1.4 Hình ảnh Boostrap 4

Bootstrap là một framework bao gồm các HTML, CSS và JavaScript template dùng để phát triển website chuẩn responsive. Bootstrap cho phép quá trình thiết kế website diễn ra nhanh chóng và dễ dàng hơn dựa trên những thành tố cơ bản sẵn có như typography, forms, buttons, tables, grids, navigation, image carousels…với mã nguồn mở.

- Sử dụng Angular material design:

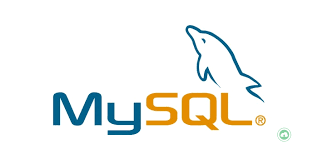


Hình 1.5 Hình ảnh Angular Materal

Angular material design cung cấp sẵn rất nhiều built-in module cho project của bạn. Một số tính năng như autocomplete, datepicker, slider, menu, grid,các toolbar, select, checkbox, radio button, card, grid, list,... được sử dụng cùng với material trong Angular.

### ***1.2.4 Hệ quản trị cơ sở dữ liệu:***

Sử dụng hệ quản trị cơ sở dữ liệu MySQL.

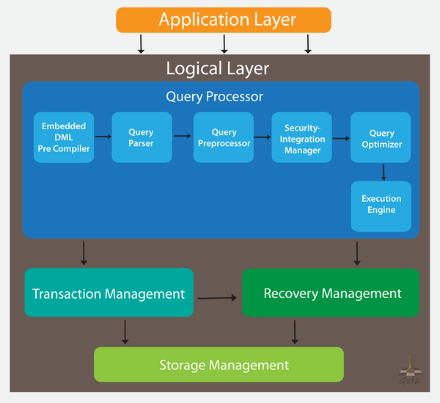


Hình 1.6 Hình ảnh MySQL

MySQL là hệ quản trị cơ sở dữ liệu tự do nguồn mở phổ biến nhất thế giới và được các nhà phát triển rất ưa chuộng trong quá trình phát triển ứng dụng. Vì MySQL là cơ sở dữ liệu tốc độ cao, ổn định và dễ sử dụng, có tính khả chuyển, hoạt động trên nhiều hệ điều hành cung cấp một hệ thống lớn các hàm tiện ích rất mạnh. Với tốc độ và tính bảo mật cao, MySQL rất thích hợp cho các ứng dụng có truy cập CSDL trên internet. Người dùng có thể tải về MySQL miễn phí từ trang chủ. MySQL có nhiều phiên bản cho các hệ điều hành khác nhau: phiên bản Win32 cho các hệ điều hành dòng Windows, Linux, Mac OS X, Unix, FreeBSD, NetBSD, Novell NetWare, SGI Irix, Solaris, SunOS,..

MySQL là một trong những ví dụ rất cơ bản về Hệ Quản trị Cơ sở dữ liệu quan hệ sử dụng Ngôn ngữ truy vấn có cấu trúc (SQL).

MySQL được sử dụng cho việc bổ trợ NodeJs, PHP, Perl, và nhiều ngôn ngữ khác, làm nơi lưu trữ những thông tin trên các trang web viết bằng NodeJs, PHP hay Perl,...



Hình 1.7 Kiến trúc của MySQL

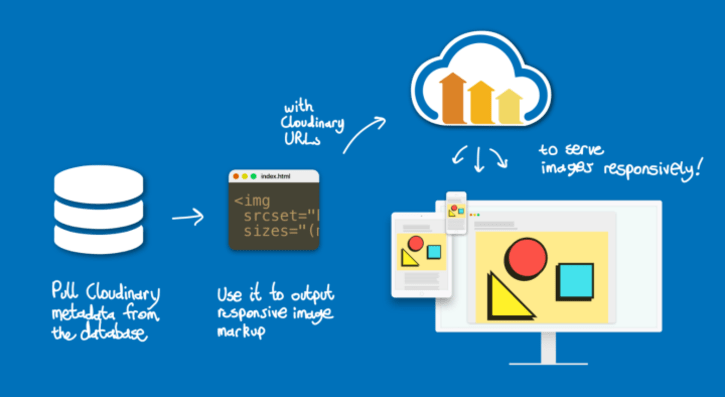
### ***1.2.4 Hệ quản ảnh Cloudinary:***

Cloudinary là một cloud-based service, nó cung cấp một giải pháp quản lý hình ảnh bao gồm upload, lưu trữ, thao tác, tối ưu hóa và delivery.

Với cloudinary bạn có thể dễ dàng upload ảnh lên cloud, tự động thực thi các thao tác với ảnh một cách thông minh mà không cần phải cài đặt bất kì một phần mềm phức tạp nào khác. Cloudinary cung cấp các APIs toàn diện và màn hình quản lý giúp chúng ta dễ dàng tích hợp vào các trang web và ứng dụng di động.

Cloudinary cung cấp một Ruby Gem để dễ dàng cho việc tương tác với các app viết bởi ngôn ngữ Ruby với một số framwork như là Rails hay Sinatra với các tương tác chính sau:

* Xây dựng các URL để chuyển đổi và thao tác với hình ảnh
* Rails view helper cho việc nhúng và thay đổi hình ảnh
* API wrappers: upload image, quản lý và nhiều thứ khác
* Upload image trực tiếp từ trình duyệt sử dụng một jQuery plugin.
* Tích hợp với Active Record
* CarrierWave plugin
* Hình ảnh tĩnh đồng bộ với CDN delivery
* Migration tool



Hình 1.8 Kiến trúc của Cloudinary

# **Chương 2: PHÂN TÍCH HỆ THỐNG**

## **2.1. Giới thiệu chung**

### ***2.1.1. Mô tả:***

Hệ thống chạy trên nền web, giúp cho người dùng có thể dễ dàng xem và lựa chọn các sản phẩm yêu thích của mình, bình luận nhận xét về địa điểm mình đã trải nghiệm và đồng thời cho phép người dùng đăng sản phẩm lên hệ thống để có thể nhận được các đơn hàng.

### ***2.1.2. Chức năng :***

- Giúp cho người dùng có thể dễ dàng xem và lựa chọn các sản phẩm yêu thích của mình thông qua việc lưu vào giỏ hàng, mỗi sản phẩm sẽ được phân loại theo danh mục để dễ dàng tìm kiếm.

- Bình luận, nhận xét về địa điểm mình đã trải nghiệm để cung cấp cái nhìn trực quan nhất cho các bài đăng có trong hệ thống.

- Người dùng cũng có thể xem các bài đăng cụ thể của 1 user có trong hệ thóng.

- Quản lý các các bài đăng,đơn hàng trong cơ sở dữ liệu.

### ***2.1.3. Đối tượng:***

Hệ thống phục vụ cho 2 đối tượng người dùng chính:

- Người quản trị

- Người dùng cuối

### ***2.1.4. Môi trường hoạt động:***

Hệ thống hoạt dộng trên mỗi trường web, có thể chạy trên nhiều trình duyệt web. ***2.1.5. Giao diện :***

Giao diện người dùng dễ nhìn, trực quan và hiển thị được đầy đủ thông tin nhất. Giao diện chia thành nhiều khu vực, nhưng phải đảm bảo sự thống nhất và không gây rối mắt khi thao tác Giao diện quản lý của thành viên phải có đầy đủ các phím chức năng, trực quan và dễ dàng sử dụng nhất. Hiển thị đầy đủ thông tin nhưng không phức tạp. Giao diện quản trị viên phải cung cấp cái nhìn tổng quan nhất của hệ thống như danh sách các user, danh sách các bài đăng, danh sách các đơn hàng,danh sách comments,… và đầy đủ.

***2.1.6. Tính cấp thiết của đề tài Xây dựng mạng xã hội We-travel :***

Hỗ trợ kịp thời cho những người nông dân,doanh nghiệp chưa tìm được một kênh phân phối cho sản phẩm của mình - Là một kênh thông tin bổ ích cho những ai muốn tìm hiểu về các sản phẩm nông sản Viêt Nam,luôm cập nhập và ở đó họ có thể quảng bá,bán sản phẩm của mình và nhận được những đóng góp ý kiếm để có thể phát triển sản phẩm của mình.

## **2.2. Actors:**

Bảng dưới đây sẽ mô tả tất cả các tác nhân tham gia vào hệ thông FreshMarket. Mỗi tác nhân sẽ có một vai trò khác nhau trong hệ thống. Hiện tại chúng ta có các tác nhân : Admin,User.

|  |  |
| --- | --- |
| **Actor** | **Description** |
| Admin | Có quyền chỉnh sửa thông tin cá nhân, mật khẩu  Quản lí,thêm,chỉnh sửa thông tin các user có trong hệ thông  Quản lý,chỉnh sửa các bài đăng của các user.  Quản lý cac đơn hàng có trong hệ thống.  Quản lý các comments của các bài đăng. |
| User | Có quyền chỉnh sửa thông tin cá nhân, mật khẩu  Tạo mới, cập nhật,quản lí bài đăng các mặt hàng của mình  Tạo mới,cập nhật,quản lí của đơn hàng mình đặt mua.  Cập nhập,quản lí các đơn hàng các user khác đặt mua qua các bài đăng của mình.  Tạo mới,chỉnh sửa các comments về các bài đăng.  Tạo mới,chỉnh sửa các đánh giá về các bài đăng. |

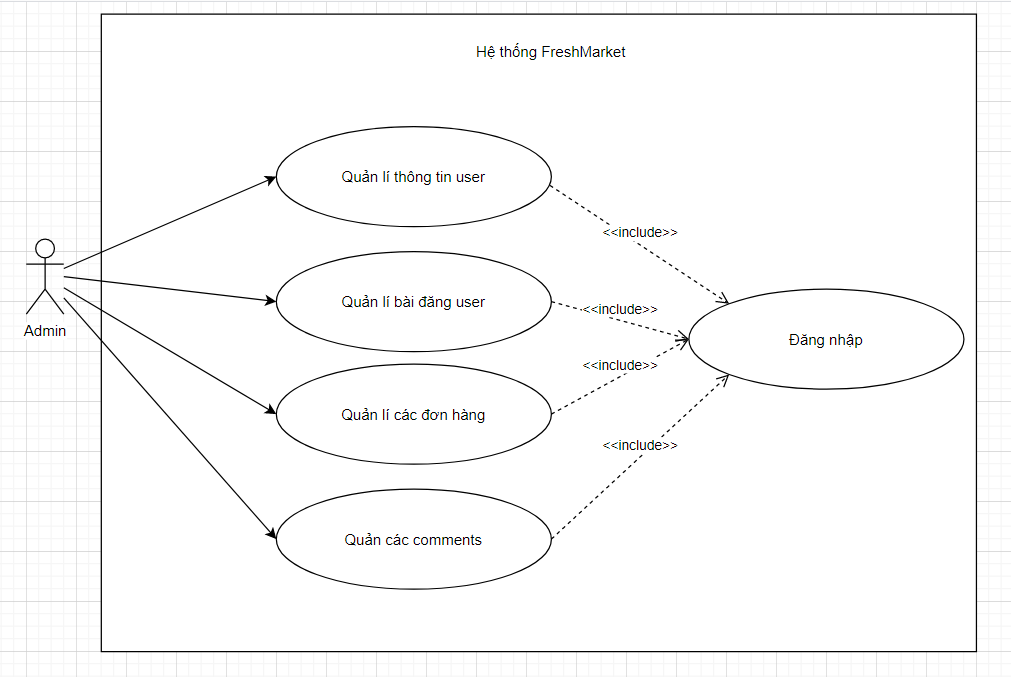
Bảng 2.1 Actors

## **2.2. Sơ đồ Use cases:**

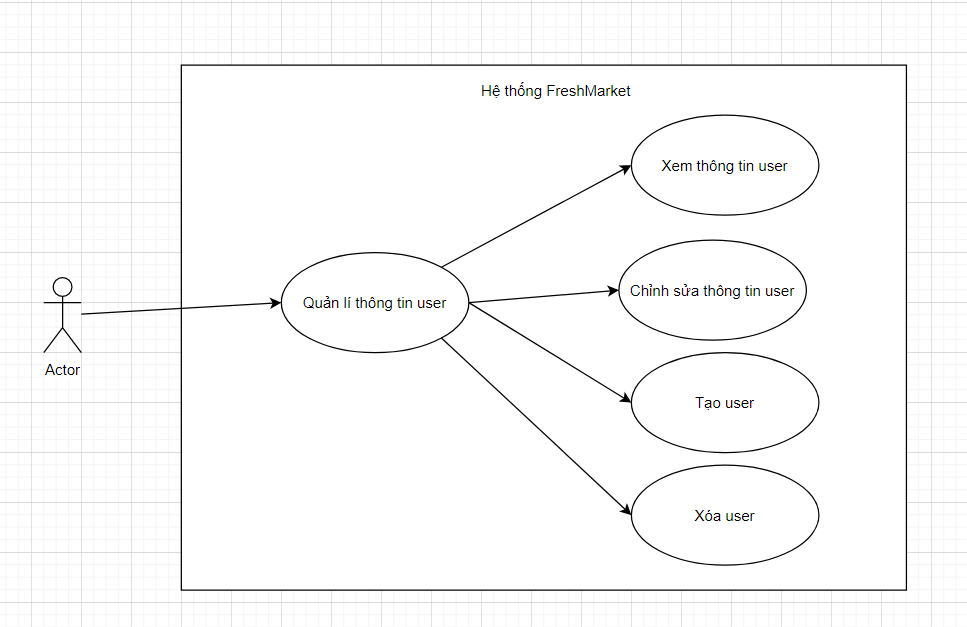
Chúng ta có 2 sơ đồ use case tương ứng với 2 actors có trong hệ thông:

* Admin: là người quản trị hệ thống
* User: là người dùng của hệ thống.

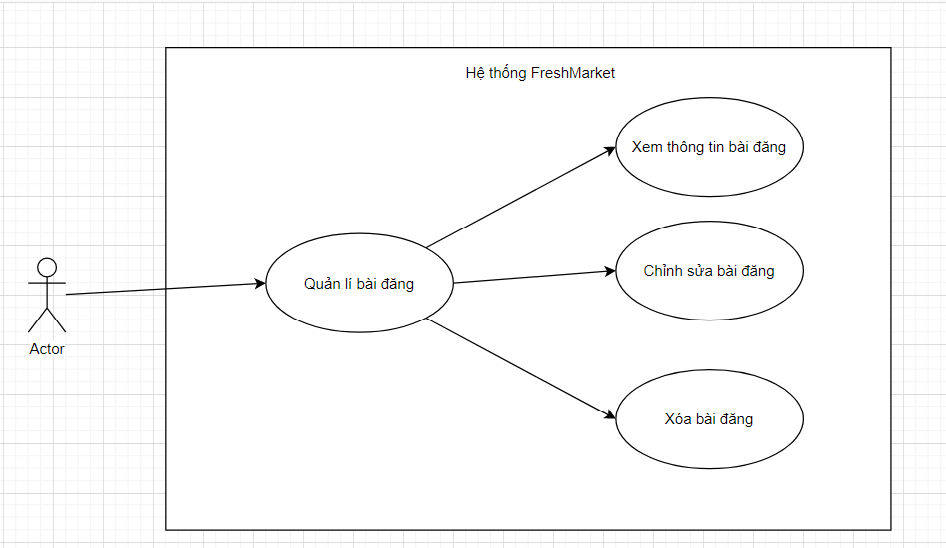
1. Admin



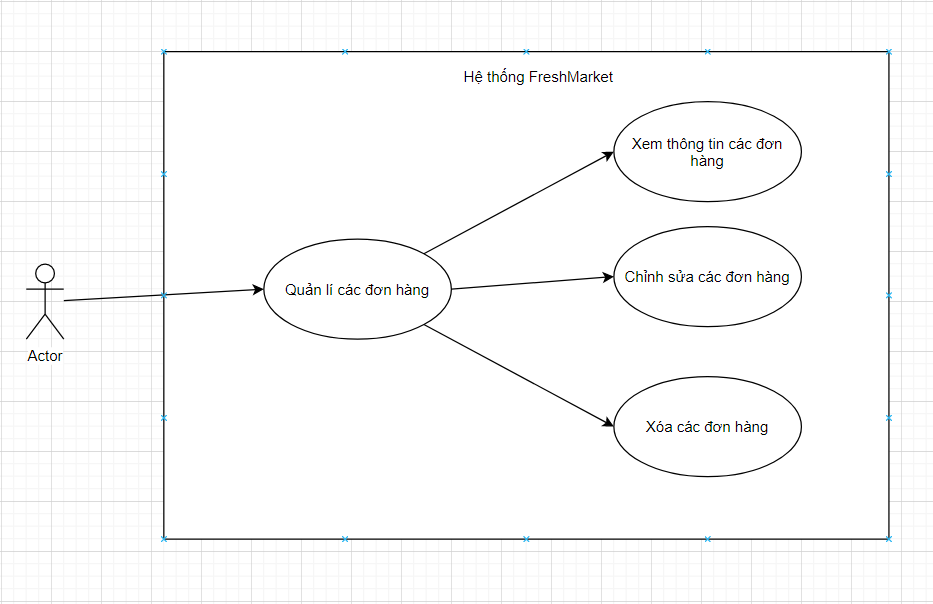
Hình 2.2.1 Sơ đồ Use-case cho người quản lý



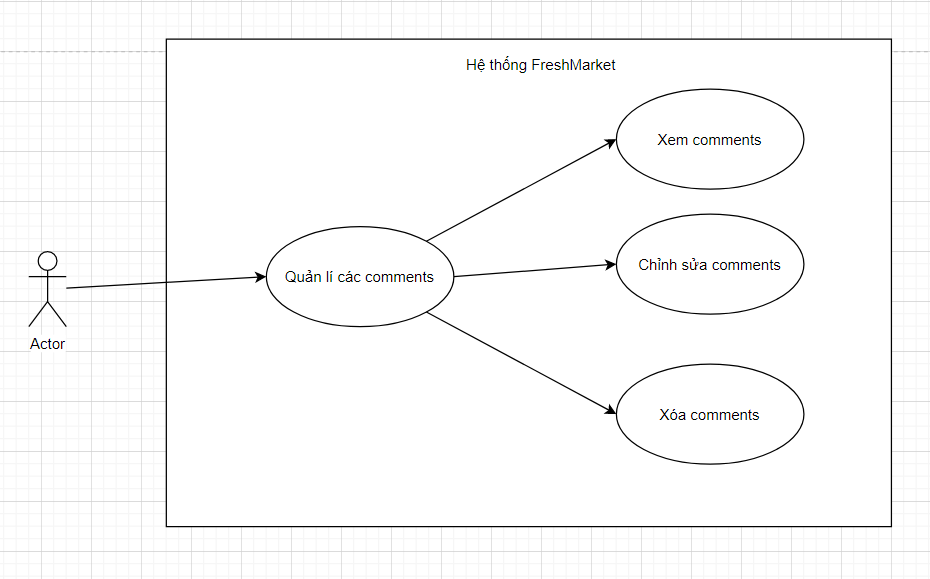
Hình 2.2.2 Sơ đồ Use-case cho admin quản lý user



Hình 2.2.3 Sơ đồ Use-case cho admin quản lý bài đăng

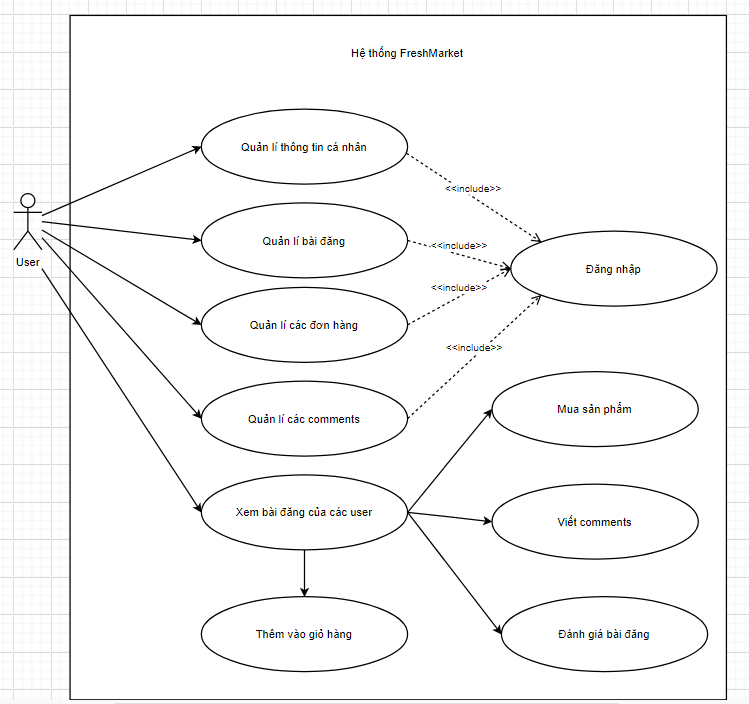


Hình 2.2.4 Sơ đồ Use-case cho admin quản lý các đơn hàng

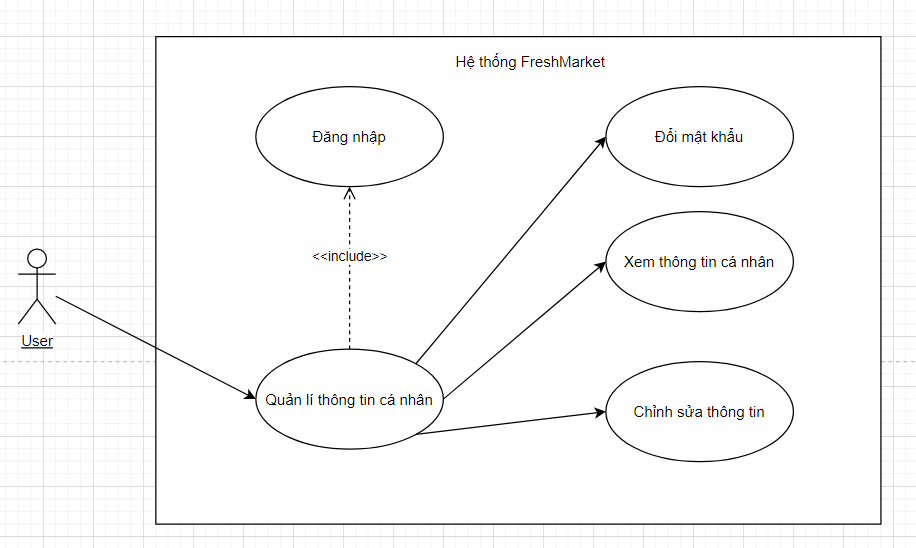


Hình 2.2.5 Sơ đồ Use-case cho admin quản lý comments

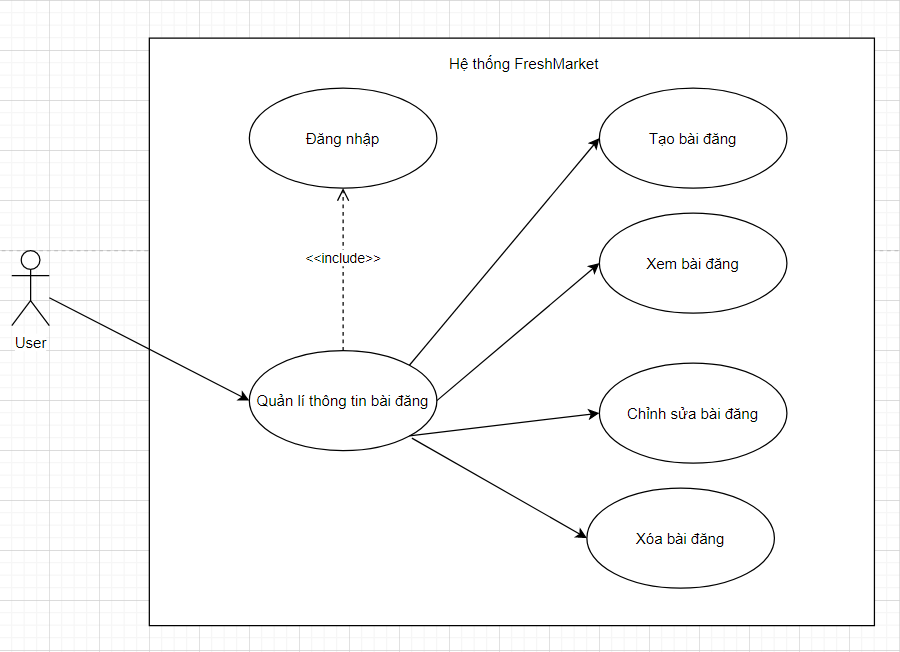
1. User



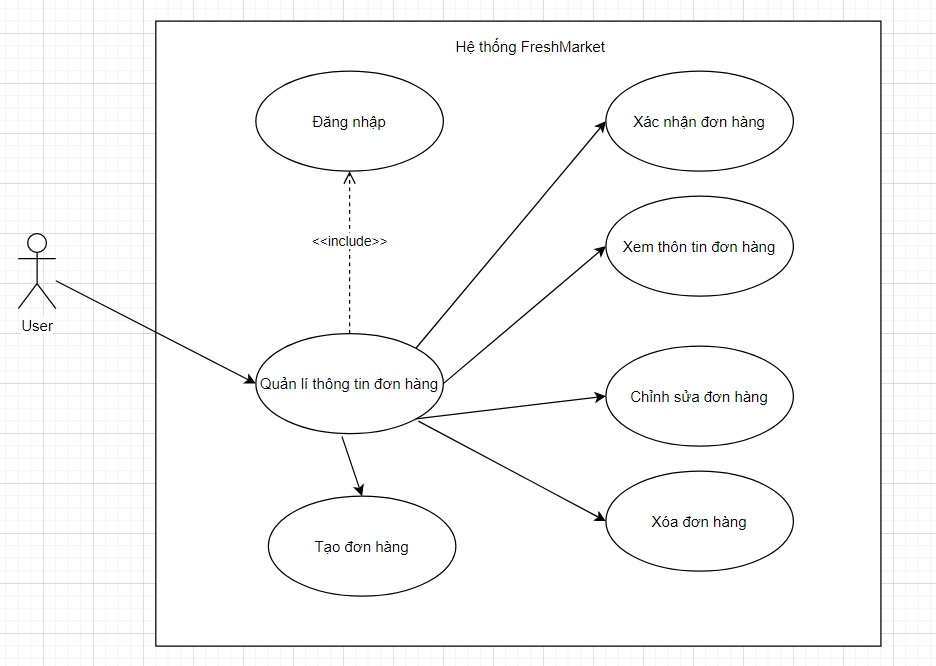
Hình 2.2.6 Sơ đồ Use-case cho user



Hình 2.2.7 Sơ đồ Use-case cho user quản lý thông tin cá nhân

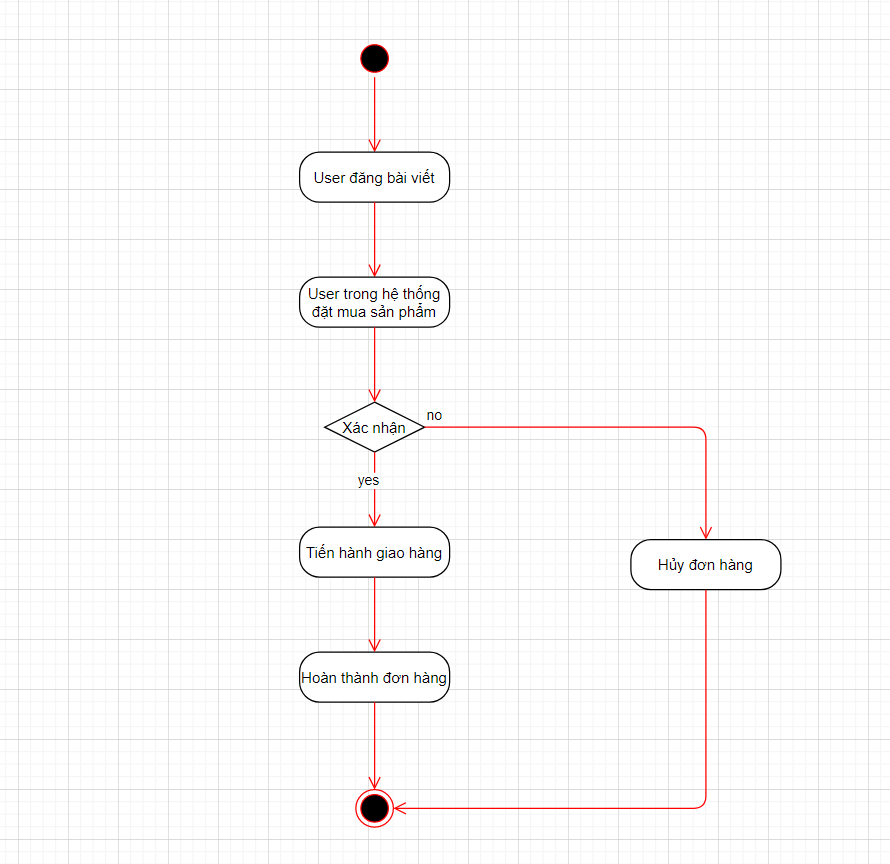


Hình 2.2.8 Sơ đồ Use-case cho user quản lý bài đăng



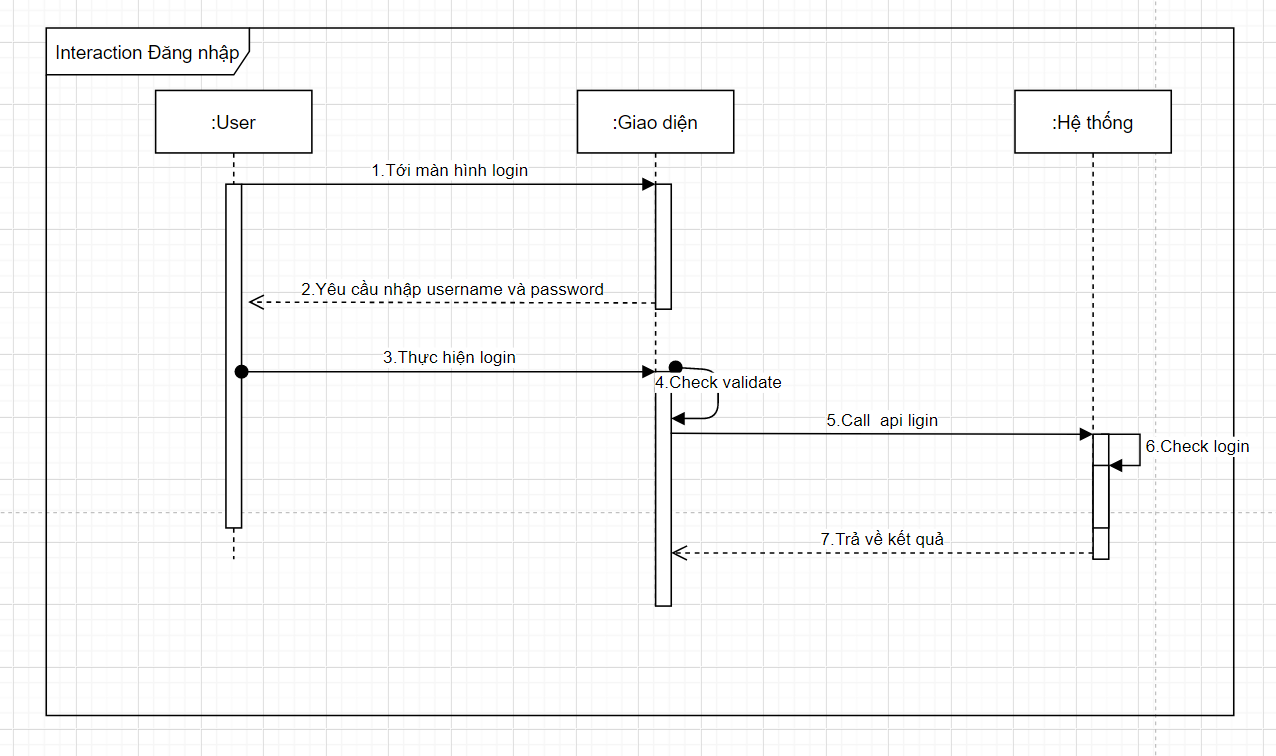
Hình 2.2.9 Sơ đồ Use-case cho user quản lý đơn hàng

## **2.3. Sơ đồ hoạt động:**

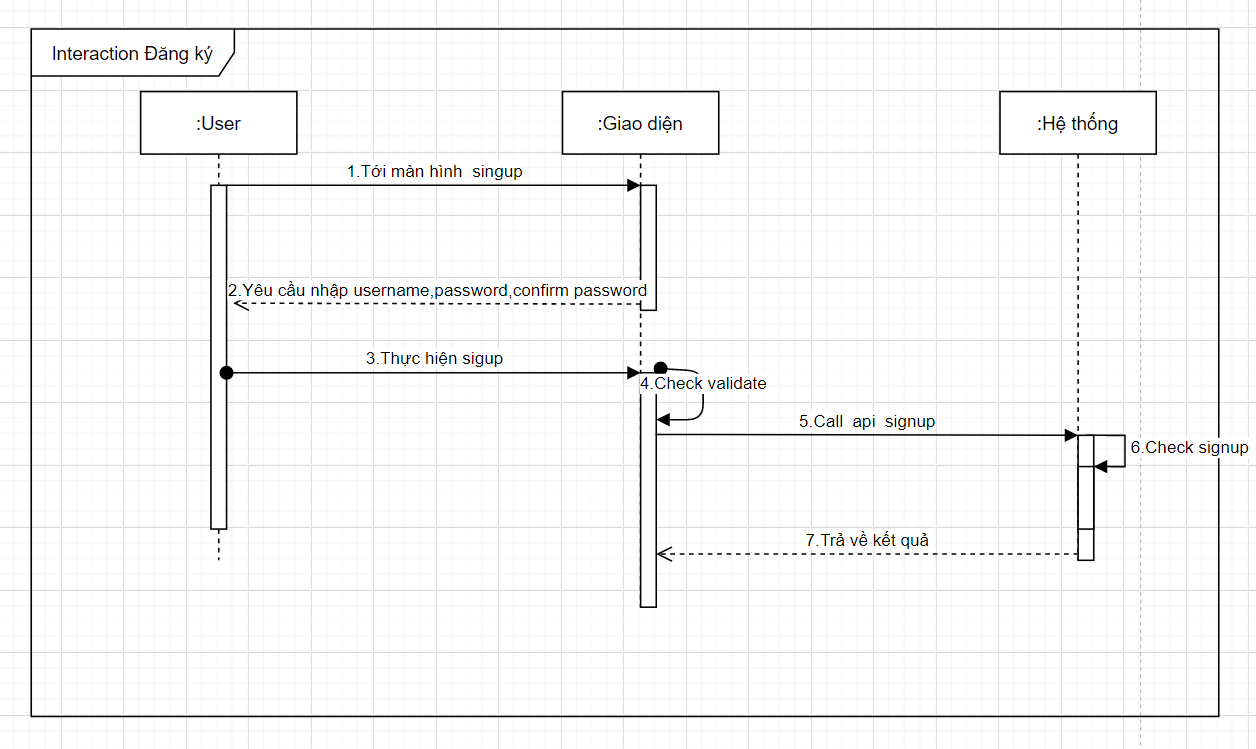


Hình 2.3 Sơ đồ hoạt động

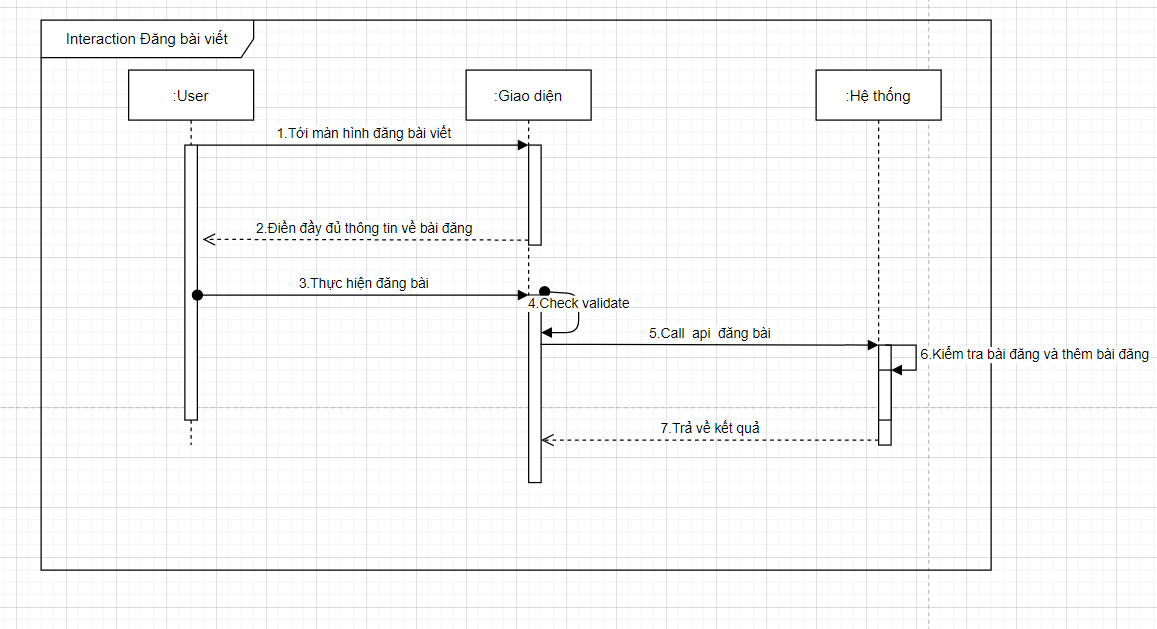
## **2.4. Sơ đồ tuần tự:**



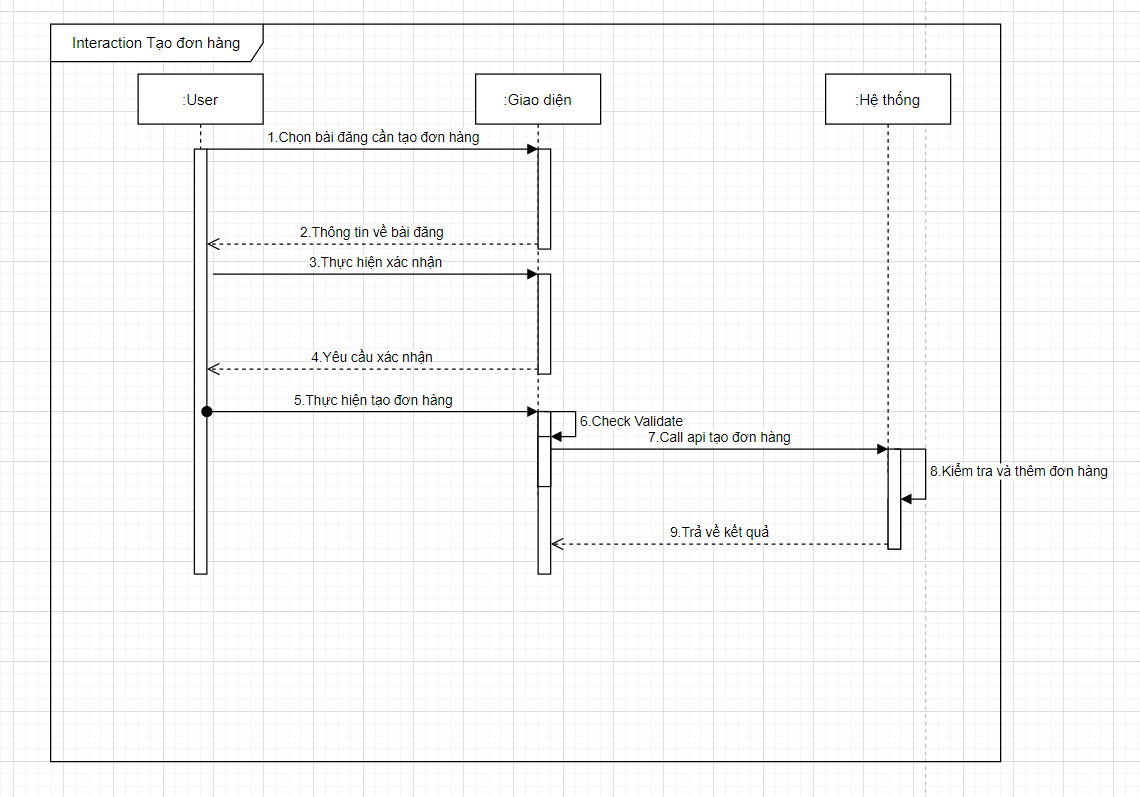
Hình 2.4.1 Sơ đồ tuần tự đăng nhập



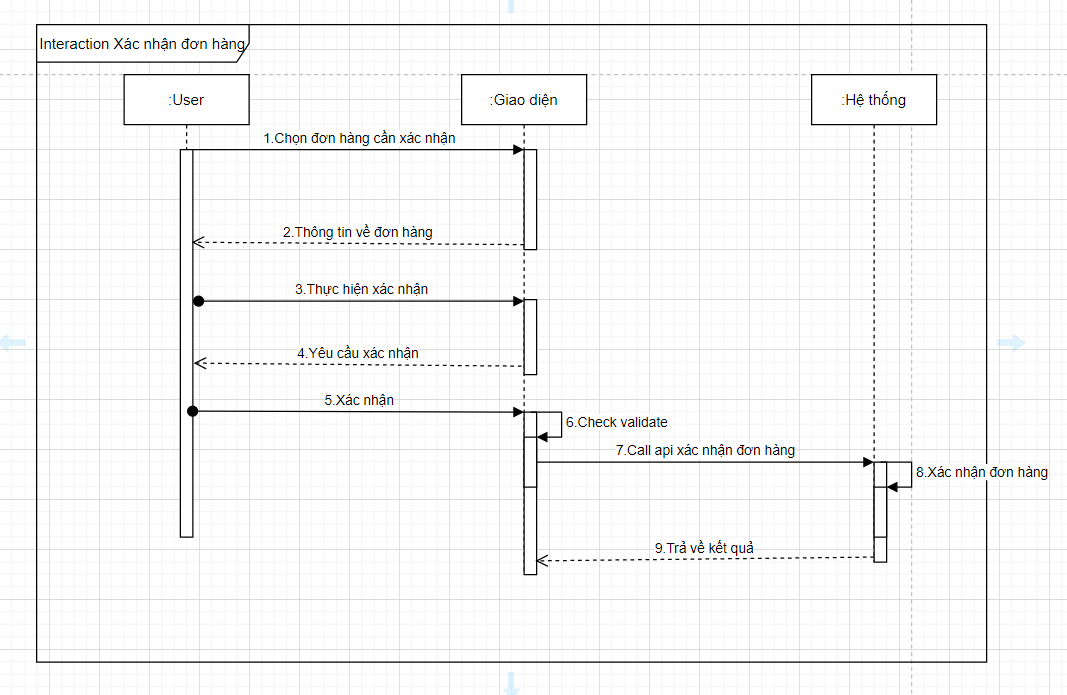
Hình 2.4.2 Sơ đồ tuần tự đăng ký tài khoản



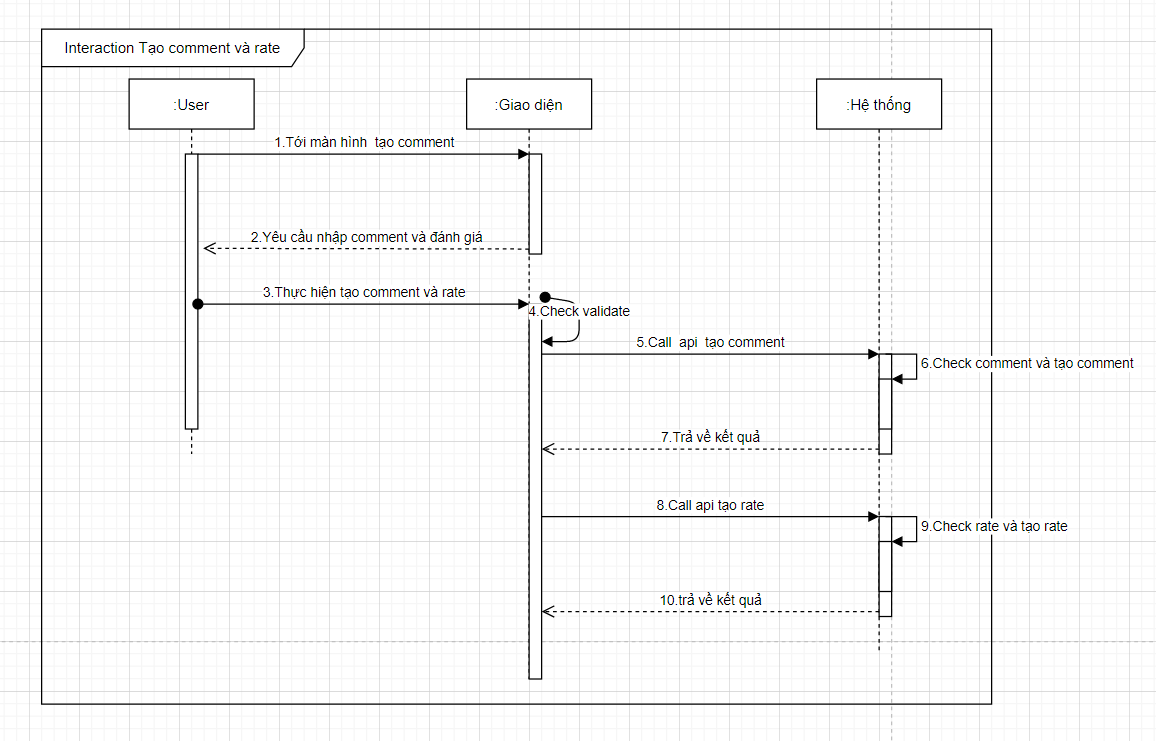
Hình 2.4.3 Sơ đồ tuần tự đăng bài viết



Hình 2.4.3 Sơ đồ tuần tự tạo đơn hàng



Hình 2.4.4 Sơ đồ tuần tự xác nhận đơn hàng



Hình 2.4.5 Sơ đồ tuần tự tạo comments và rate

## **2.5. Từ điển dữ liệu:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Avatar | | | | |
| STT | Trường | Kiểu dữ liệu | Độ dài | Mô tả |
| 1 | id | long | 10 | Id tự động tăng |
| 2 | url | varchar | 255 | url của image |

Bảng 2.4.1 Avatar

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CalculationUnit | | | | |
| STT | Trường | Kiểu dữ liệu | Độ dài | Mô tả |
| 1 | id | long | 10 | Id tự động tăng |
| 2 | UnitName | varchar | 255 | Tên của đơn vị tính |

Bảng 2.4.2 CalculationUnit

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CommentPost | | | | |
| STT | Trường | Kiểu dữ liệu | Độ dài | Mô tả |
| 1 | id | long | 10 | Id tự động tăng |
| 2 | Content | varchar | 255 | Nội dung của comment |
| 3 | DateOfComment | DateTime |  | Ngày tạo comment |
| 4 | Name | varchar | 255 | Tên của blocking round |

Bảng 2.4.3 CommentPost

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ImagePost | | | | |
| STT | Trường | Kiểu dữ liệu | Độ dài | Mô tả |
| 1 | id | long | 10 | Id tự động tăng |
| 2 | url | varchar | 255 | url của image |
| 3 | PostId | long | 10 | Khóa ngoại nối với Post tương ứng |

Bảng 2.4.4 ImagePost

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Post | | | | |
| STT | Trường | Kiểu dữ liệu | Độ dài | Mô tả |
| 1 | id | long | 10 | Id tự động tăng |
| 2 | Address | varchar | 255 | Địa chỉ của mặt hàng của bài đăng |
| 3 | DateOfPost | DateTime |  | Ngày tạo bài post |
| 4 | Description | varchar | 255 | Mô tả bài đăng |
| 5 | PostName | varchar | 100 | Tên của bài đăng |
| 6 | UnitPrince | Double |  | Giá của mỗi đơn hàng |
| 7 | CalculationUnitId | long | 10 | Khóa ngoại nối với CalculationUnit tương ứng |
| 8 | CategoryId | long | 10 | Khóa ngoại nối với Category tương ứng |
| 9 | ProvinceId | long | 10 | Khóa ngoại nối với Province tương ứng |
| 10 | UserId | long | 10 | Khóa ngoại nối với User tương ứng |

Bảng 2.4.5 Post

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| RoleUser | | | | |
| STT | Trường | Kiểu dữ liệu | Độ dài | Mô tả |
| 1 | id | long | 10 | Id tự động tăng |
| 2 | role\_name | varchar | 50 | Tên của role |

Bảng 2.4.6 RoleUser

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Province | | | | |
| STT | Trường | Kiểu dữ liệu | Độ dài | Mô tả |
| 1 | id | long | 10 | Id tự động tăng |
| 2 | ProvinceName | varchar | 50 | Tên của tỉnh thành |

Bảng 2.4.7 Province

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purchase | | | | |
| STT | Trường | Kiểu dữ liệu | Độ dài | Mô tả |
| 1 | id | long | 10 | Id tự động tăng |
| 2 | address | varchar | 255 | Địa chỉ người mua hàng |
| 3 | DateOfOder | DateTime |  | Ngày đặt hàng |
| 4 | Fullname | varchar | 50 | Họ và tên người mua |
| 5 | PhoneNumber | varchar | 11 | Số điện thoại người mua |
| 6 | PurchaseNumber | int | 9 | Số lượng hàng mua |
| 7 | BuyerId | long | 10 | Khóa ngoại nối với User tương ứng |
| 8 | PostId | long | 10 | Khóa ngoại nối với Post tương ứng |
| 9 | StatusPurchaseId | long | 10 | Khóa ngoại nối với StatusPurchase tương ứng |
| 10 | TransportTypeId | long | 10 | Khóa ngoại nối với TransportType tương ứng |
| 11 | TransportCost | Double |  | Giá của quá trình vận chuyển |

Bảng 2.4.8 Purchase

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| RatePost | | | | |
| STT | Trường | Kiểu dữ liệu | Độ dài | Mô tả |
| 1 | id | long | 10 | Id tự động tăng |
| 2 | RateNumber | int | 1 | Số điểm đánh giá |
| 3 | PostId | long | 10 | Khóa ngoại nối với Post tương ứng |
| 4 | UserId | long | 10 | Khóa ngoại nối với User tương ứng |

Bảng 2.4.9 RatePost

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| StatusPurchase | | | | |
| STT | Trường | Kiểu dữ liệu | Độ dài | Mô tả |
| 1 | id | long | 10 | Id tự động tăng |
| 2 | status | varchar | 10 | Tên của status |

Bảng 2.4.10 StatusPurchase

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| User | | | | |
| STT | Trường | Kiểu dữ liệu | Độ dài | Mô tả |
| 1 | id | long | 10 | Id tự động tăng |
| 2 | Email | varchar | 255 | Địa chỉ email của user |
| 3 | Fullname | varchar | 50 | Tên của user |
| 4 | Password | varchar | 255 | Mật khẩu của tài khoản user |
| 5 | PhoneNumber | varchar | 11 | Số điện thoại của user |
| 6 | UserName | varchar | 50 | Tên tài khoản của user |
| 7 | AvatarId | long | 10 | Khóa ngoại nối với Avatar tương ứng |
| 8 | RoleUserId | long | 10 | Khóa ngoại nối với RoleUser tương ứng |

Bảng 2.4.11 User

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TransportType | | | | |
| STT | Trường | Kiểu dữ liệu | Độ dài | Mô tả |
| 1 | id | long | 10 | Id tự động tăng |
| 2 | TransportName | varchar | 50 | Tên của transport |

Bảng 2.4.12 TransportType

# **Chương 3: PHÁT TRIỂN VÀ TRIỂN KHAI HỆ THỐNG**

## **3.1. Phát triển hệ thống:**

### **3.1.1. Triển khai hệ thống REST-API:**

**RESTful API** là một tiêu chuẩn dùng trong việc thiết kế API cho các ứng dụng web (thiết kế Web services) để tiện cho việc quản lý các resource. Nó chú trọng vào tài nguyên hệ thống (tệp văn bản, ảnh, âm thanh, video, hoặc dữ liệu động…), bao gồm các trạng thái tài nguyên được định dạng và được truyền tải qua HTTP.



Hình 3.1.1 Mô hình hoạt động của Rest API

**API** (**A**pplication **P**rogramming **I**nterface) là một tập các quy tắc và cơ chế mà theo đó, một ứng dụng hay một thành phần sẽ tương tác với một ứng dụng hay thành phần khác. API có thể trả về dữ liệu mà bạn cần cho ứng dụng của mình ở những kiểu dữ liệu phổ biến như [JSON](https://topdev.vn/blog/json-la-gi/) hay XML.

**REST** (**RE**presentational **S**tate **T**ransfer) là một dạng chuyển đổi cấu trúc dữ liệu, một kiểu kiến trúc để viết API. Nó sử dụng phương thức HTTP đơn giản để tạo cho giao tiếp giữa các máy. Vì vậy, thay vì sử dụng một URL cho việc xử lý một số thông tin người dùng, REST gửi một yêu cầu HTTP như GET, POST, DELETE, vv đến một URL để xử lý dữ liệu.

**RESTful API** là một tiêu chuẩn dùng trong việc thiết kế các API cho các ứng dụng web để quản lý các resource. RESTful là một trong những kiểu thiết kế API được sử dụng phổ biến ngày nay để cho các ứng dụng (web, mobile…) khác nhau giao tiếp với nhau.

Chức năng quan trọng nhất của **REST** là quy định cách sử dụng các HTTP method (như GET, POST, PUT, DELETE…) và cách định dạng các URL cho ứng dụng web để quản các resource. RESTful không quy định logic code ứng dụng và không giới hạn bởi ngôn ngữ lập trình ứng dụng, bất kỳ ngôn ngữ hoặc framework nào cũng có thể sử dụng để thiết kế một **RESTful API**.

REST hoạt động chủ yếu dựa vào giao thức HTTP. Các hoạt động cơ bản nêu trên sẽ sử dụng những phương thức HTTP riêng.Cách thức hoạt động của hệ thống **RESTful API như sau:**



Hình 3.1.2 Mô hình client call API

* GET (SELECT): Trả về một Resource hoặc một danh sách Resource.
* POST (CREATE): Tạo mới một Resource.
* PUT (UPDATE): Cập nhật thông tin cho Resource.
* DELETE (DELETE): Xoá một Resource.

Những phương thức hay hoạt động này thường được gọi là **CRUD.**

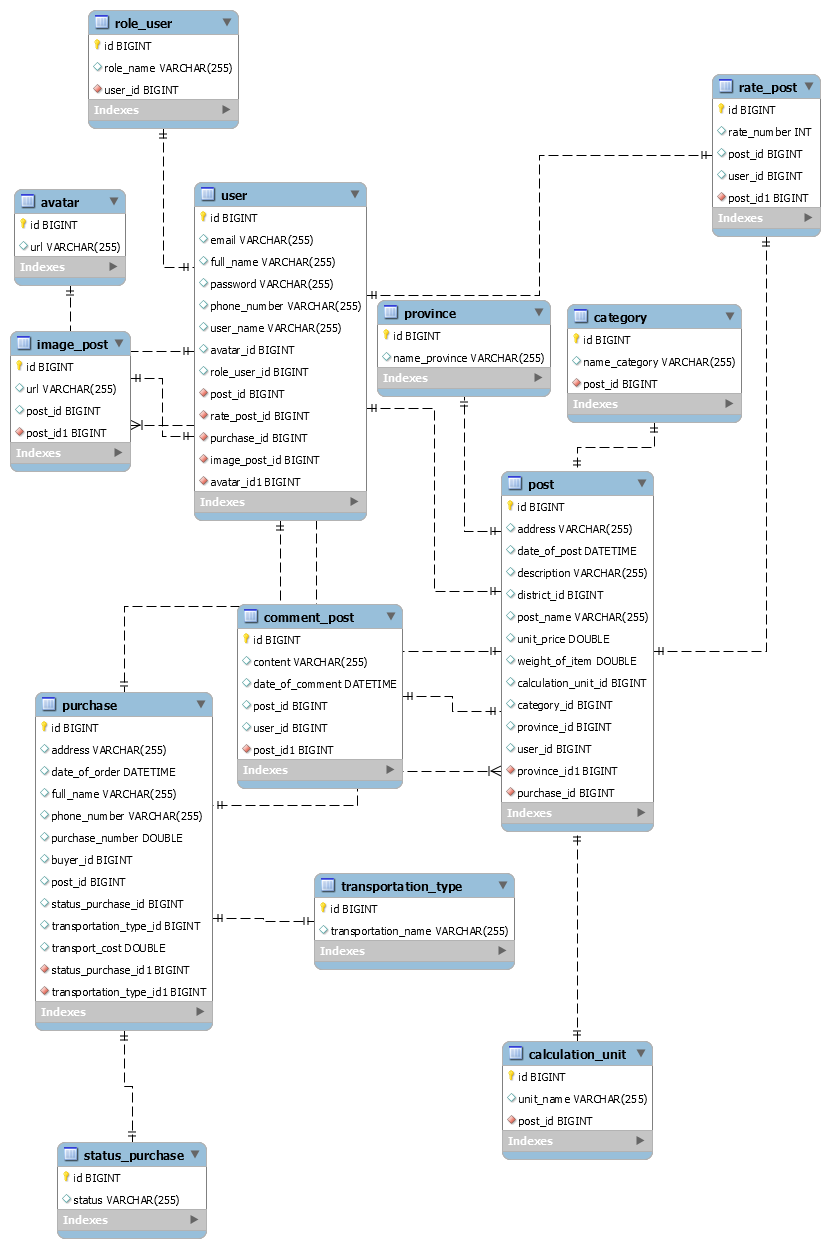
Authentication và data trả về: **RESTful API** không sử dụng [session](https://topdev.vn/blog/session-la-gi/) và cookie, nó sử dụng một access\_token với mỗi request,data trả về là một Json.

Khi chúng ta request một API nào đó thường thì sẽ có vài status thường gặp như sau:

* 200 OK – Trả về thành công cho những phương thức GET, PUT, PATCH hoặc DELETE.
* 201 Created – Trả về khi một Resouce vừa được tạo thành công.
* 204 No Content – Trả về khi Resource xoá thành công.
* 304 Not Modified – Client có thể sử dụng dữ liệu cache.
* 400 Bad Request – Request không hợp lệ
* 401 Unauthorized – Request cần có auth.
* 403 Forbidden – bị từ chối không cho phép.
* 404 Not Found – Không tìm thấy resource từ URI
* 405 Method Not Allowed – Phương thức không cho phép với user hiện tại.
* 410 Gone – Resource không còn tồn tại, Version cũ đã không còn hỗ trợ.
* 415 Unsupported Media Type – Không hỗ trợ kiểu Resource này.
* 422 Unprocessable Entity – Dữ liệu không được xác thực.
* 429 Too Many Requests – Request bị từ chối do bị giới hạn.

### **3.1.2. Cơ sở dữ liệu:**

Từ những thiết kế ở từ điển dữ liệu, chúng ta triển khai cài đặt trên hệ quản trị cơ sở dữ liệu MySQL. Kết quả thu được như hình dưới:

Hình 3.1.3 ER Diagram của FreshMarket

## **3.2. Các công nghệ sử dụng:**

- Back end: sử dụng nền tảng Java với framework Sptring API.

- Front end: sử dụng ngôn ngữ JavaScript với Framework Angular CLI.

- Sử dụng bộ UI: Angular material design UI,Boostrap 4.

- Database: MySQL.

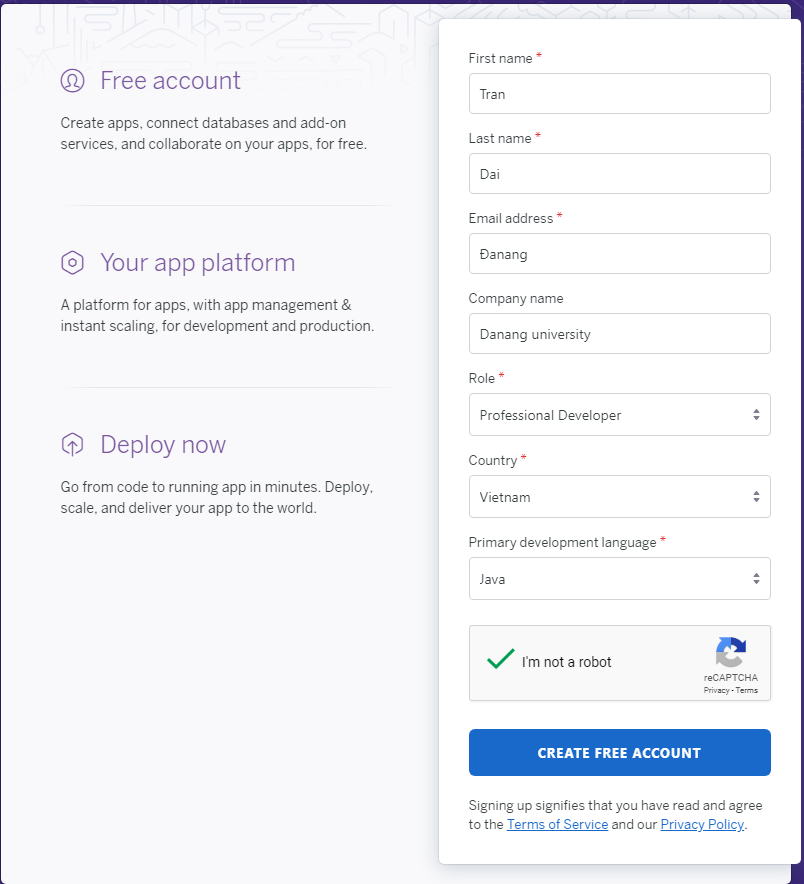
- Cloudinary.

## **3.3. Triển khai hệ thống:**

### ***3.3.1. Triển khai back end:***

**Bước 1:** Đăng ký tài khoản tại:

<https://signup.heroku.com/>

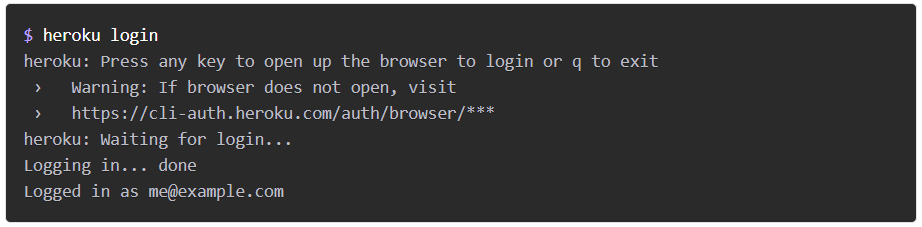


Hình 3.3.1 Màn hình đăng ký Heroku

**Bước 2:** Tải và cài đặt Heroku CLI tại:

<https://devcenter.heroku.com/articles/heroku-cli#download-and-install>

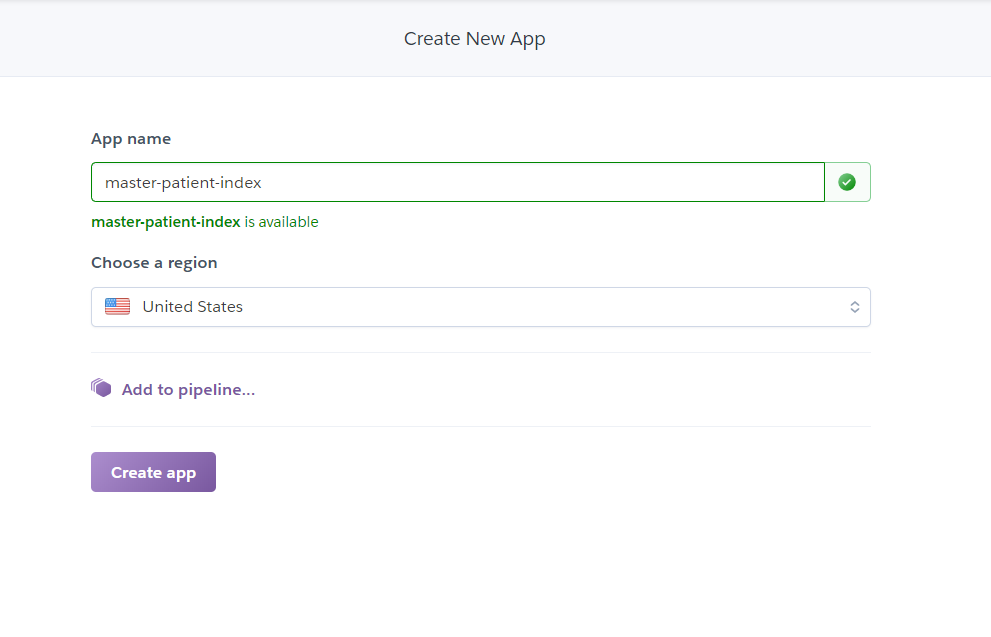
**Bước 3:** Sau khi cài đặt thành công Heroku CLI, tiến hành đăng nhập Heroku CLI với lệnh: heroku login



Hình 3.3.2 Đăng nhập Heroku CLI thành công

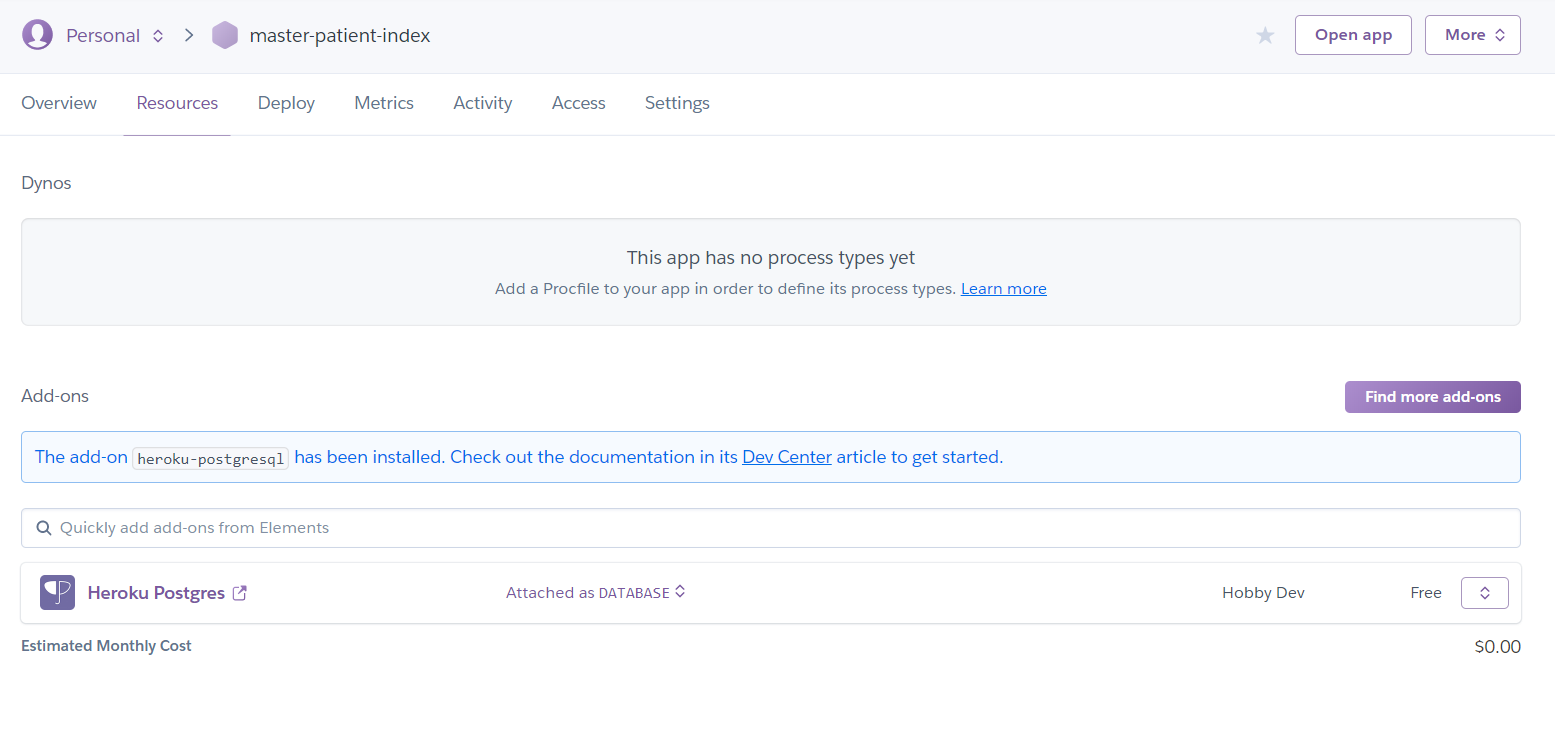
**Bước 4:** Thêm mới app cần deploy

<https://dashboard.heroku.com/new-app>



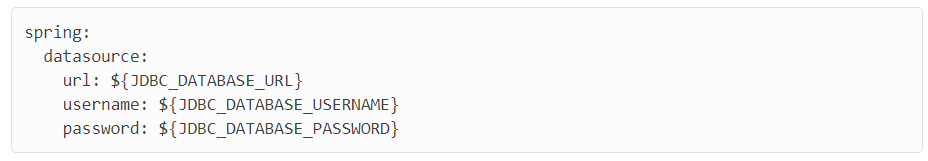
Hình 3.3.3 Màn hình thêm mới app

**Bước 5:** Thêm cơ sở dữ liệu postgreSQL bằng add-on heroku-postgresql



Hình 3.3.4 Thêm as-on heroku-postgresql

**Bước 6:** Thêm thông tin về datasource vào file application-prod.properties



Hình 3.3.5 Cấu hình datasource

**Bước 7:** Sau khi hoàn thành việc coding, ta tiến hành commit code:

git add .

git commit -am "make it better"

**Bước 8:** Deploy lên heroku:

heroku git:remote -a master-patient-index

git push heroku master

### ***3.2.2. Triển khai frontend:***

**Bước 1:** Commit source lên git:

git add .  
git commit -m "initial commit"  
git push -u origin master

**Bước 2:** Create postinstall script in package.json

"heroku-postbuild": "ng build --prod"

**Bước 3:** Từ thư mục root của project, chúng ta tạo file “server.js”. Sau đó, import Express, và tạo instance mới của Express

"start": "node server.js"

**Bước 4:** Đảm bảo rằng: port hoạt động của trang web sẽ là port 4200 hoặc port do heroku cung cấp. Đồng thời, đảm bảo người dùng có thể truy cập thư mục public, index.html.



Hình 3.3.6 Toàn bộ file server.js

**Bước 5:** Sửa dòng start trong file package.json thành:

"start": "node server/server.js"

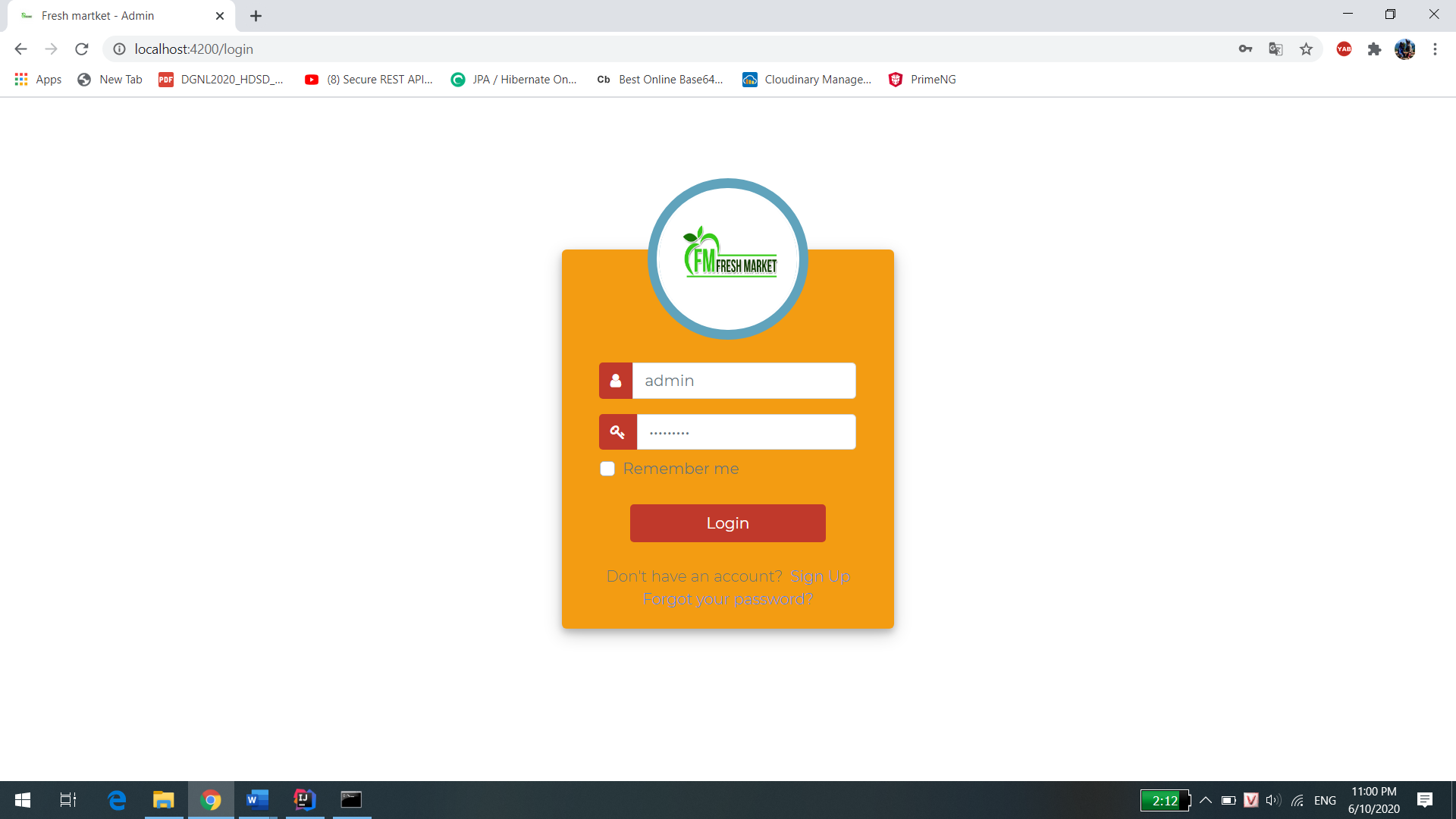
**Bước 6:** Tiến hành tạo app trên Heroku rồi deploy tương tự như phần backend.

## **3.4. Kết quả đã thực hiện:**

Sau khi đã triển khai trên môi trường internet, ứng dụng đáp ứng tốt các chức năng nghiệp vụ quan trọng trong 1 hệ thống FreshMarket, như: thêm mới user, đặt hàng,tạo giỏ hàng,xem thông tin đơn hàng,... Đồng thời, ứng dụng phải có giao diện thân thiện, dễ sử dụng, mang lại trải nghiệm tốt cho người dùng.

### ***3.4.1. Màn hình đăng nhập:***

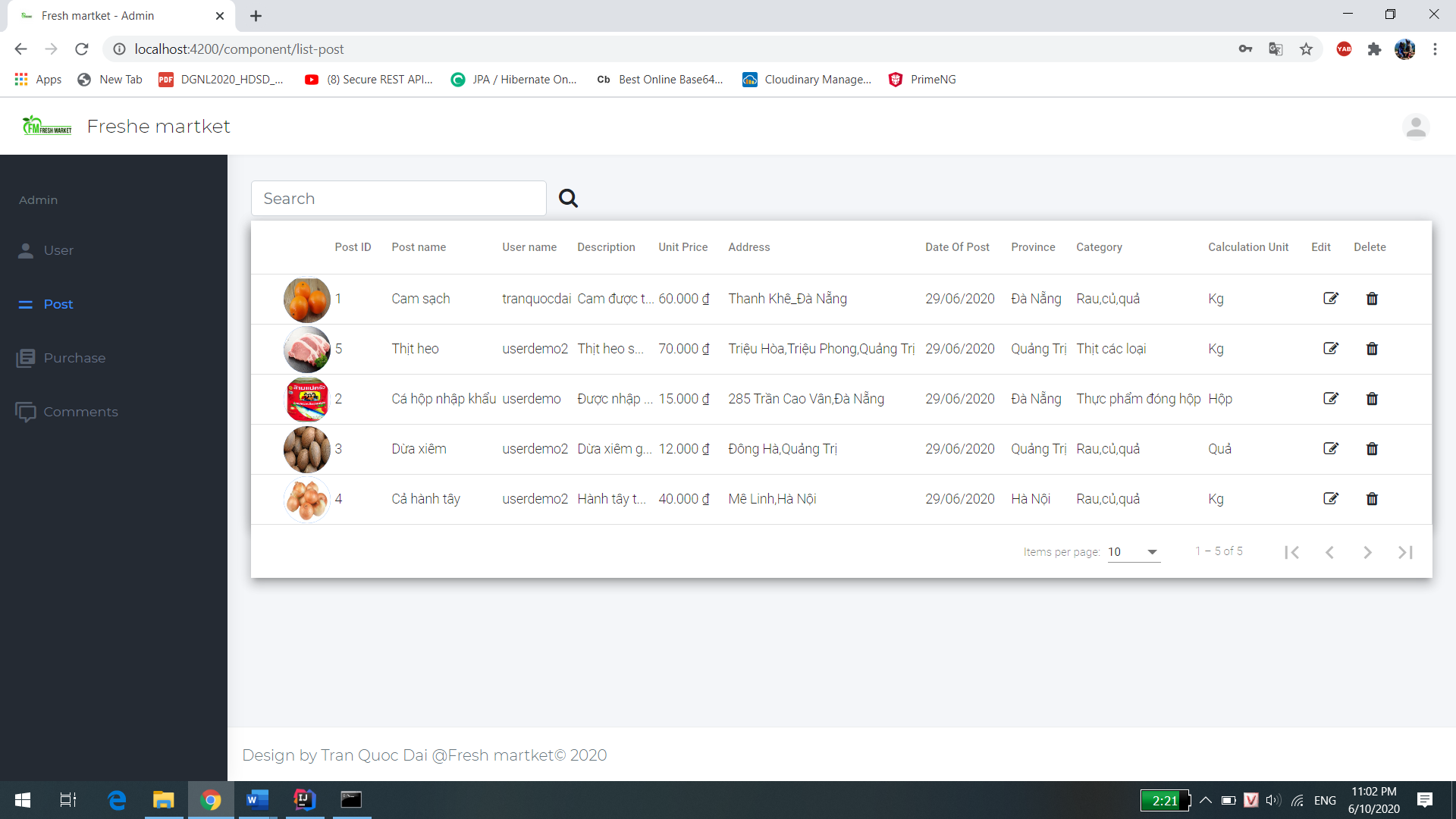
Tại màn hình này, người dung có thể đăng nhập với tài khoản, mật khẩu để vào sử dụng những chức năng của hệ thống.



Hình 3.4.1 Màn hình Login

### ***3.4.2. Màn hình quản lí của admin:***

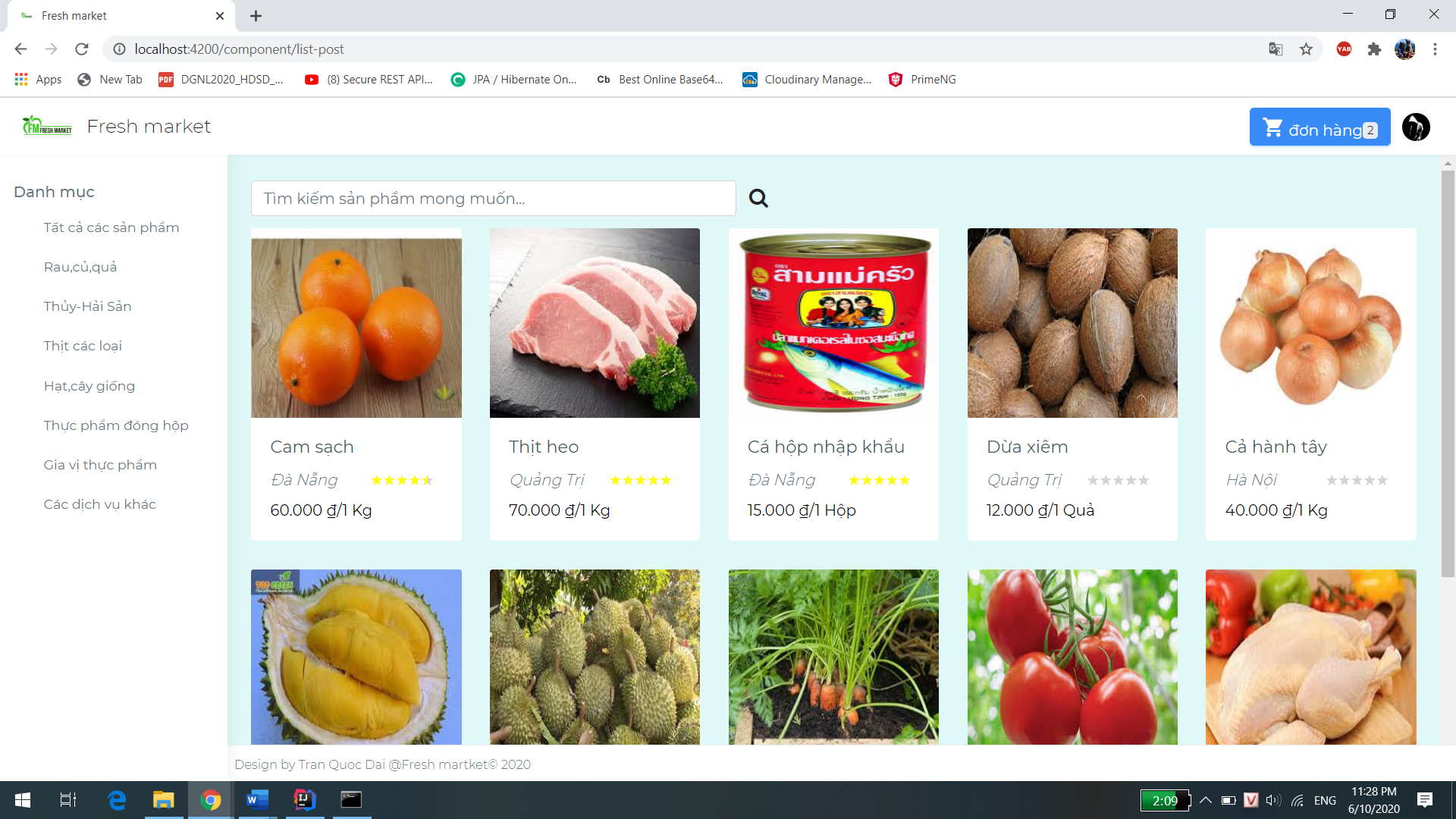
Tại màn hình này, admin có thể quản lí thông tin của các user,các bài đăng,có đơn hang và những comments về bài đăng có trong hệ thống.



Hình 3.4.2 Màn hình quản lý của admin

### ***3.4.3. Màn hình xem màn danh sách bài đăng(sản phẩm):***

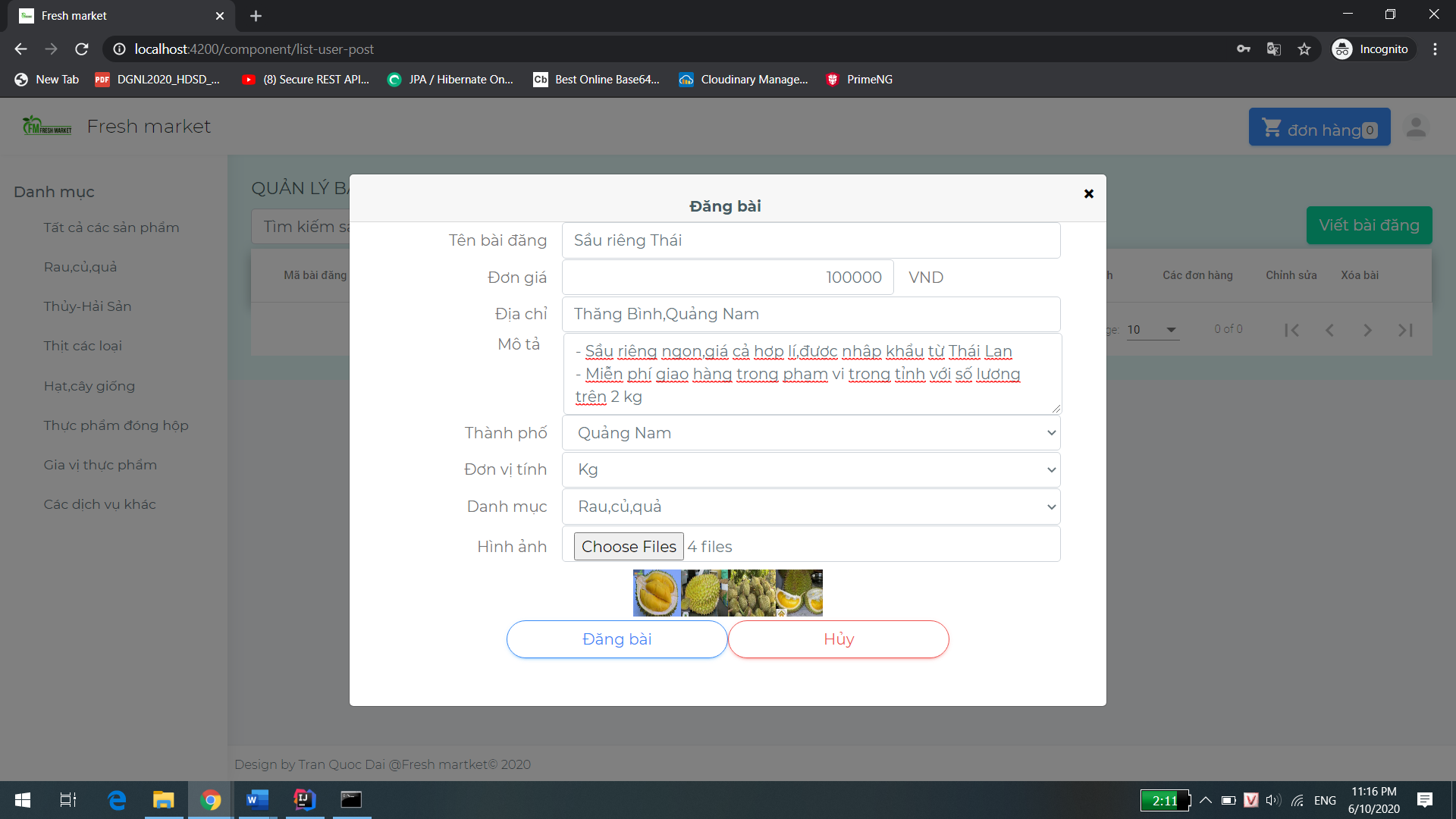
Người dùng có thể xem, tìm kiếm các sản phẩm có trong hệ thống.



Hình 3.4.3 Màn hình danh sách bài đăng

### ***3.4.4. Màn hình viết bài đăng:***

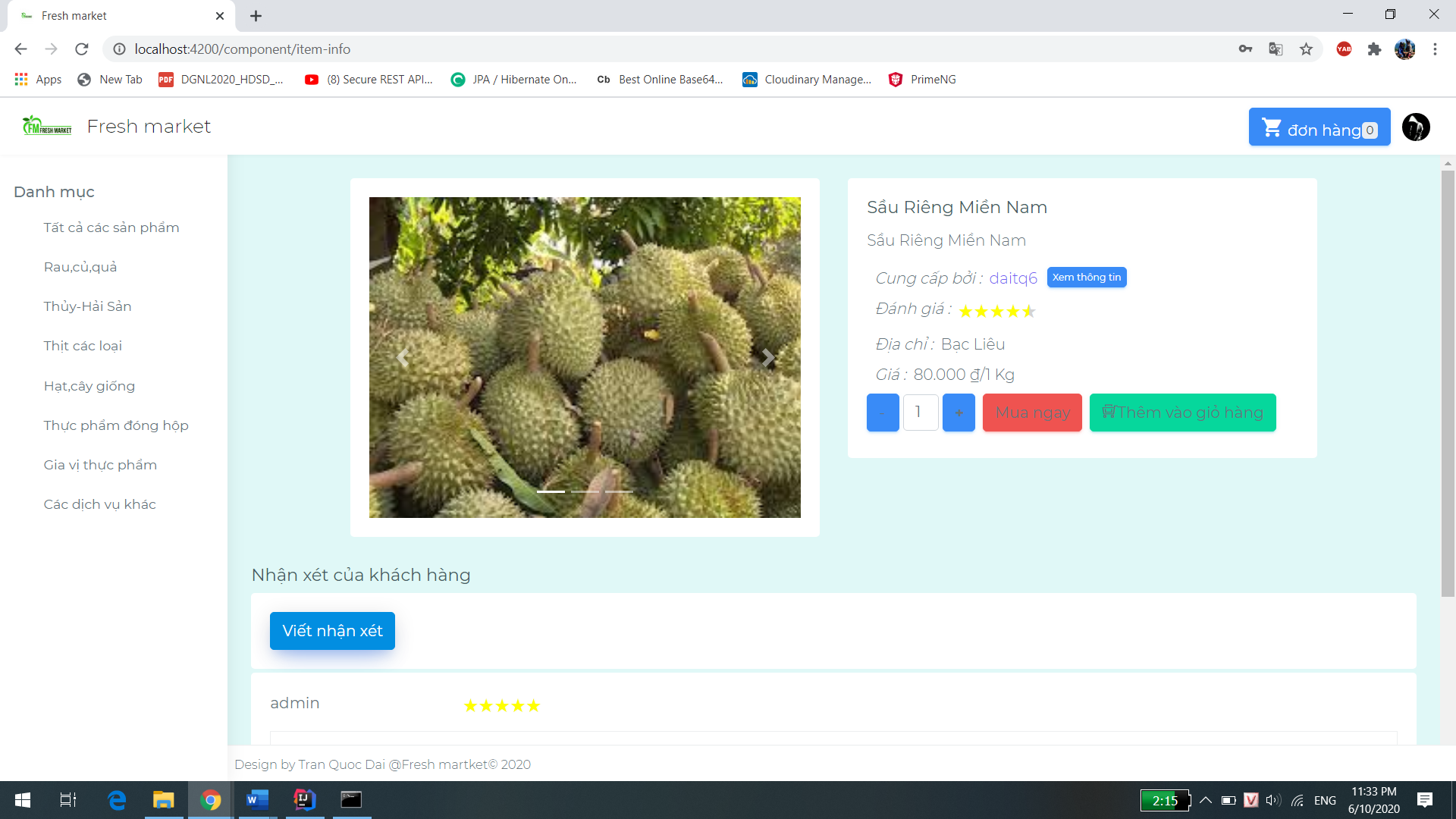
Màn hình này cho phép người dùng có thể đăng kí bài viết về sản phẩm của mình.



Hình 3.4.4 Màn hình viết bài đăng

### ***3.4.5. Màn hình xem thông tin chi tiết về sản phẩm:***

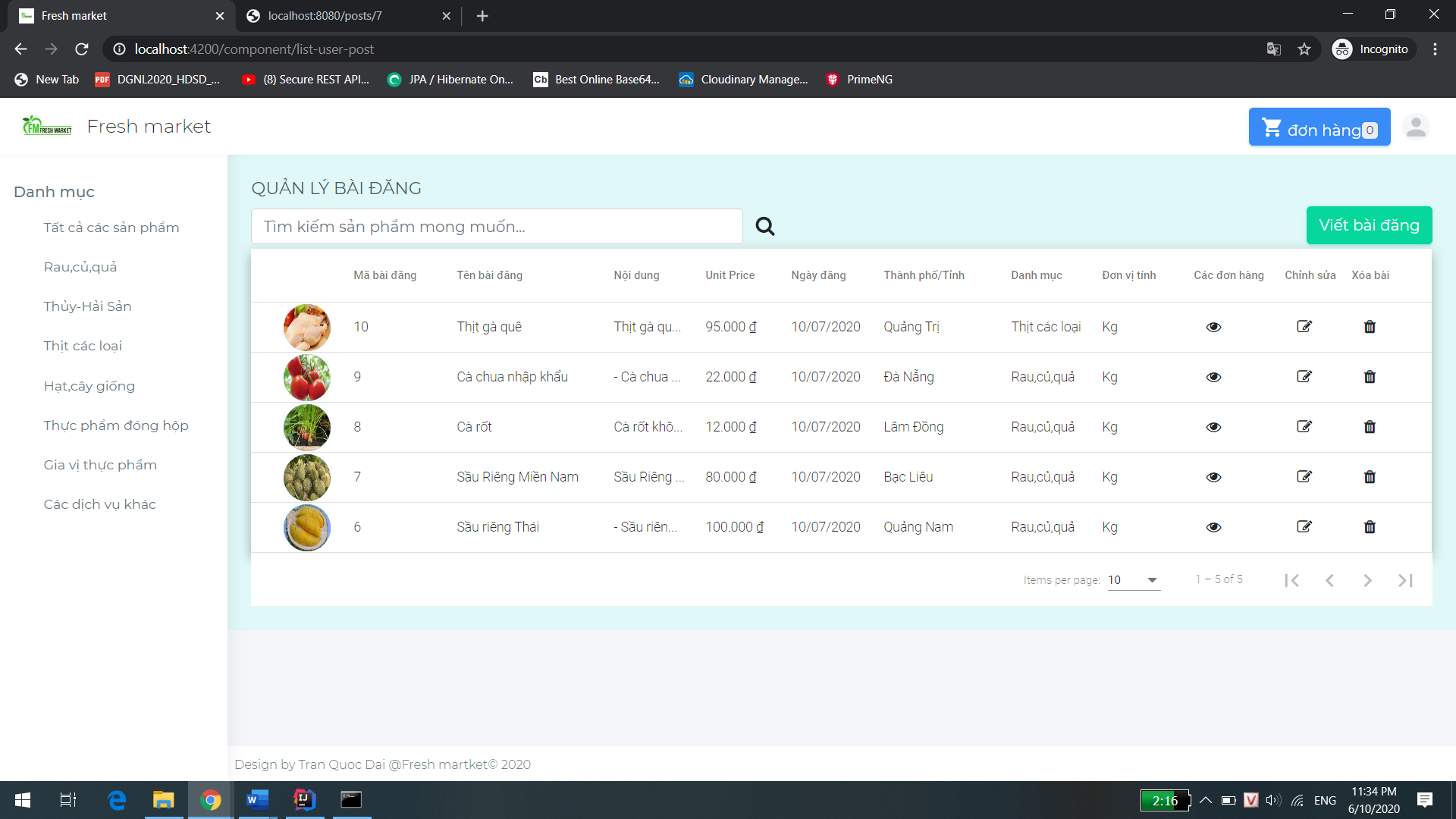
Màn hình này cho phép người dùng có thể đăng kí bài viết về sản phẩm của mình.



Hình 3.4.5 Màn hình chi tiết về sản phẩm

### ***3.4.6. Màn hình quản lí bài đăng của người dùng:***

Màn hình cho phép người dùng có thể quản lí bài đăng của mình.



Hình 3.4.6 Màn hình quản lý bài đăng của người đăng

### ***3.4.7. Màn hình quản lí đơn hàng của người dùng:***

Màn hình cho phép người dùng có thể quản lí đơn hang mua hoặc bán của mình.



Hình 3.4.7 Màn hình quản lý đơn hàng của người dùng

# **KẾT LUẬN**

**Kết quả đạt được**

Trong thời gian tìm hiểu, nghiên cứu cơ sở lý thuyết và triển khai ứng dụng công nghệ, đề tài đã đạt được những kết quả sau:

Về mặt lý thuyết, đề tài đã đạt được việc củng cố kiến thức về cách viết các tài liệu như sơ đồ use case, sơ đồ hoạt động,sơ đồ tuần tự và cách xây dựng một hệ thống theo hướng dịch vụ với RESTful API.

Về mặt thực tiễn ứng dụng, đề tài đã đạt được việc hoàn thành một website hoàn chỉnh có thể vận hành tốt trên internet, đáp ứng tốt các yêu cầu chức năng cũng như phi chức năng đối với hệ thống FreshMarket.

Tuy nhiên, đề tài còn tồn tại các vấn đề như sau:

- Do sử dụng cloud free nên tốc độ truy cập không ổn định

- Giao diện cần chỉnh chu, đồng bộ hơn.

**Hướng phát triển**

Một số số hướng nghiên cứu và phát triển của đề tài như sau:

* Cải thiện UI, UX của ứng dụng
* Thực hiện cân bằng tải(load balancing) để tối ưu hiệu năng hệ thống

# **TÀI LIỆU THAM KHẢO**

<https://angular.io/cli>

<https://stackoverflow.com/>

<https://material.angular.io/>

<https://dev.mysql.com/doc/>

<https://www.jetbrains.com/idea/documentation/>

<https://viblo.asia/p/thuat-toan-xep-hang-bai-viet-dang-hot-thinh-hanh-nhu-reddit-va-hacker-news-hoat-dong-the-nao-gAm5y8xXldb?fbclid=IwAR0D9irs43ANyYz8eZOq4KgbSVe3uz4M9veRmUnaDe0von0pybh6pYju_6Y>

<https://docs.oracle.com/javase/8/docs/api/org/w3c/dom/Document.html>

<https://getbootstrap.com/docs/4.0/getting-started/introduction/>

<https://www.w3schools.com/>

# **PHỤ LỤC 1**

***Function calculateScore():***

private float calculateScore(Post post) {

List<RatePost> ratePostList = ratePostRepository.findByPost(post);

int s = 0;

if (ratePostList.size() > 0) {

for (RatePost element : ratePostList) {

s += (element.getRateNumber().intValue() - 3);

}

}

List<Purchase> purchasesCancel = purchaseRepository.findPurchaseCancel(post.getId());

List<Purchase> purchases = purchaseRepository.findByPost(post);

s += purchases.size() - 2 \* purchasesCancel.size();

int sign = 0;

if (s > 0) {

sign = 1;

} else if (s < 0) {

sign = -1;

}

long seconds = Constants.LOCAL\_DATE\_TIME\_START.until(post.getDateOfPost(), ChronoUnit.MINUTES);

int n = 1;

if (s >= 1) {

n = Math.abs(s);

}

return (float) Math.log10(n) + (float) (sign \* seconds) / 45000;

}