

**HỌC VIỆN HÀNG KHÔNG VIỆT NAM**

**KHOA CÔNG NGHỆ THÔNG TIN**

---



## **BÁO CÁO**

### **Lập trình Python**

**Xây dựng hệ thống quản lý kho và vận chuyển Logistics sử dụng  
Django và Firebase**

**Giảng viên hướng dẫn: ThS Ngô Minh Nhựt**

**Sinh viên/ Nhóm sinh viên thực hiện: Nhóm 03**

**Mã số sinh viên:**

**Lớp: 010100087202**

**Thành Phố Hồ Chí Minh, tháng 11/2025**

**HỌC VIỆN HÀNG KHÔNG VIỆT NAM**

**KHOA CÔNG NGHỆ THÔNG TIN**

---



## **BÁO CÁO**

### **Lập trình Python**

**Xây dựng hệ thống quản lý kho và vận chuyển Logistics sử dụng  
Django và Firebase**

**Giảng viên hướng dẫn: ThS Ngô Minh Nhựt**

**Sinh viên/ Nhóm sinh viên thực hiện: Nhóm 03**

**Mã số sinh viên:**

**Lớp: 010100087202**

**Thành phố Hồ Chí Minh, tháng 11/2025**

**Danh sách Nhóm:**

STT	Họ và tên	MSSV	Lớp	Ghi chú
1	Ngô Thành Danh	2254810280	22ĐHTT06	Nhóm Trưởng
2	Trần Khang	2254810123	22ĐHTT05	Thành viên
3	Dương Đình Thuận	2331540247	23ĐHTT05	Thành viên
4	Đàm Tuấn Hưng	2331540260	23ĐHTT05	Thành viên
5	Nguyễn Thế Minh	2331540226	23ĐHTT05	Thành viên

<b>Cán bộ chấm thi 1</b> <i>(ký và ghi rõ họ tên)</i>	<b>Cán bộ chấm thi 2</b> <i>(ký và ghi rõ họ tên)</i>
<b>Cán bộ chấm thi phúc khảo 1</b> <i>(ký và ghi rõ họ tên)</i>	<b>Cán bộ chấm thi phúc khảo 2</b> <i>(ký và ghi rõ họ tên)</i>

## DANH MỤC CÁC KÝ HIỆU, CHỮ VIẾT TẮT

Ký hiệu, chữ viết tắt	Chữ viết đầy đủ
MVT	Model-View-Template
API	Application Programming Interface
GPS	Global Positioning System
URL	Uniform Resource Locator
HTTP	HyperText Transfer Protocol
MVC	Model-View-Controller
OSM	OpenStreetMap
OSRM	Open Source Routing Machine

## DANH MỤC CÁC BẢNG, HÌNH VẼ, ĐỒ THỊ

Hình 1 - Ngôn ngữ lập trình Python.....	13
Hình 2 - Framework Django.....	14
Hình 3 - Visual Studio Code.....	14
Hình 4 - Google Firebase.....	15
Hình 5 - Thư viện Leaflet.....	16
Hình 6 - OpenStreetMap.....	17
Hình 7 - Framework Bootstrap.....	17
Hình 8 - Sơ đồ khối của hệ thống.....	18
Hình 9 - Sơ đồ Use-Case.....	19
Hình 10 - Sơ đồ hoạt động đăng ký tài khoản.....	20
Hình 11 - Sơ đồ hoạt động đăng nhập hệ thống.....	21
Hình 12 - Sơ đồ hoạt động quản lý sản phẩm.....	22
Hình 13 - Sơ đồ hoạt động nhập kho số lượng sản phẩm.....	23
Hình 14 - Sơ đồ hoạt động quy trình xuất kho và giao hàng.....	25
Hình 15 - Sơ đồ tuần tự quy trình đăng nhập hệ thống.....	26
Hình 16 - Sơ đồ tuần tự quy trình Cập nhật vị trí GPS (real-time).....	27
Hình 17 - Sơ đồ tuần tự tạo phiếu xuất và trừ số lượng trong kho.....	29
Hình 18 - Sơ đồ tuần tự hoàn thành giao hàng.....	29
Hình 19 - Giao diện đăng nhập hệ thống.....	30
Hình 20 - Giao diện đăng ký tài khoản.....	31
Hình 21 - Giao diện Dashboard.....	32
Hình 22 - Giao diện tất cả sản phẩm trong kho.....	33
Hình 23 - Giao diện nhập kho các sản phẩm đã có.....	33
Hình 24 - Giao diện bước 1 của tạo phiếu xuất kho.....	34
Hình 25 - Giao diện bước 2 của tạo phiếu xuất kho.....	34
Hình 26 - Giao diện chi tiết đơn hàng xuất kho.....	35

Hình 27 - Giao diện danh sách tất cả phiếu xuất kho.....	35
Hình 28 - Giao diện cập nhật sản phẩm.....	36
Hình 29 - Giao diện nhận đơn hàng của tài xế.....	37
Hình 30 - Giao diện danh sách đơn hàng đã được nhận vận chuyển.....	37
Hình 31 - Giao diện theo dõi vị trí đơn hàng.....	38

# MỤC LỤC

*Trang*

DANH MỤC CÁC BẢNG, HÌNH VẼ, ĐỒ THỊ.....	5
MỞ ĐẦU.....	9
CHƯƠNG 1. GIỚI THIỆU.....	10
1.1. Lý do chọn đề tài.....	10
1.2. Mục tiêu đề tài.....	10
1.3. Phạm vi đề tài.....	10
1.4 Đối tượng nghiên cứu.....	11
1.5. Phương pháp nghiên cứu.....	11
1.6. Bố cục đề tài.....	12
CHƯƠNG 2. CƠ SỞ LÝ THUYẾT.....	13
2.1. Ngôn ngữ lập trình Python.....	13
2.2. Framework Django.....	13
2.3. Môi trường lập trình Visual Studio Code.....	14
2.4. Nền tảng Google Firebase.....	15
2.5. Leaflet.js và OpenStreetMap.....	16
2.6. Framework Bootstrap.....	17
CHƯƠNG 3. PHÂN TÍCH HỆ THỐNG VÀ XÂY DỰNG SẢN PHẨM.....	18
3.1. Phân tích hệ thống.....	18
3.1.1. Sơ đồ khối của hệ thống.....	18
3.1.2. Use-Case Diagram.....	19
3.1.3. Activity Diagram.....	20
3.1.3.1. Sơ đồ hoạt động đăng ký tài khoản.....	20
3.1.3.2. Sơ đồ hoạt động đăng nhập hệ thống.....	21
3.1.3.3. Sơ đồ hoạt động quản lý sản phẩm.....	22
3.1.3.4. Sơ đồ hoạt động nhập kho số lượng sản phẩm.....	23

3.1.3.5. Sơ đồ hoạt động xuất kho và giao hàng.....	24
3.1.4. Sequence Diagram.....	26
3.1.4.1. Sơ đồ tuần tự quy trình đăng nhập hệ thống.....	26
3.1.4.2. Sơ đồ tuần tự quy trình cập nhật vị trí GPS.....	27
3.1.4.3. Sơ đồ tuần tự quy trình tạo phiếu xuất và trừ số lượng trong kho.....	28
3.1.4.4. Sơ đồ tuần tự hoàn thành giao hàng.....	29
3.2. Xây dựng giao diện sản phẩm.....	30
3.2.1. Giao diện Đăng nhập hệ thống.....	30
3.2.2. Giao diện Đăng ký tài khoản.....	31
3.2.3. Giao diện Dashboard của hệ thống.....	31
3.2.4. Giao diện danh sách tất cả sản phẩm trong kho.....	32
3.2.5. Giao diện nhập kho các sản phẩm đã có.....	33
3.2.6. Giao diện tạo phiếu xuất kho.....	34
3.2.7. Giao diện chi tiết đơn hàng xuất kho.....	35
3.2.8. Giao diện cập nhật sản phẩm.....	36
3.2.9. Giao diện nhận đơn hàng của tài xế.....	36
3.2.10. Giao diện theo dõi danh sách các đơn hàng đã được nhận vận chuyển.....	37
3.2.11. Giao diện theo dõi vị trí đơn hàng.....	38
KẾT LUẬN.....	39
DANH MỤC TÀI LIỆU THAM KHẢO.....	40



## MỞ ĐẦU

Đề tài "Nghiên cứu và xây dựng hệ thống quản lý và vận chuyển kho hàng Logistics" được thực hiện nhằm giải quyết các thách thức trong việc vận hành kho vận của doanh nghiệp, đặc biệt là các vấn đề phát sinh từ phương pháp quản lý thủ công như sai sót dữ liệu và chậm trễ trong khâu vận chuyển. Dự án phát triển một hệ thống quản lý tập trung dưới dạng website, cho phép người dùng quản lý hàng hóa trong kho, xử lý quy trình xuất/nhập kho và giám sát lộ trình giao hàng theo thời gian thực.

Hệ thống sử dụng ngôn ngữ lập trình Python và framework Django làm công nghệ backend chính. Điểm nổi bật của dự án là việc tích hợp Firebase Firestore làm cơ sở dữ liệu, cho phép đồng bộ dữ liệu (như trạng thái đơn hàng, tồn kho) theo thời gian thực, kết hợp với Firebase Authentication để xác thực và phân quyền người dùng (như Quản trị viên, Nhân viên kho, Tài xế). Ngoài ra, đề tài còn tích hợp các API bản đồ (như OpenStreetMap/Google Maps) để hiện thực hóa tính năng theo dõi vị trí đơn hàng, giúp tối ưu hóa quy trình logistics và nâng cao hiệu quả vận hành.

# CHƯƠNG 1. GIỚI THIỆU

## 1.1. Lý do chọn đề tài

Trong bối cảnh cạnh tranh gay gắt, doanh nghiệp cần tối ưu hóa quản lý và vận chuyển để nâng cao hiệu suất. Logistics đóng vai trò then chốt, nhưng các phương pháp thủ công thường gây sai sót dữ liệu, tốn thời gian kiểm kê, tồn kho ảo và khó khăn trong vận chuyển. Vì vậy, nhóm đã chọn đề tài “Nghiên cứu và xây dựng hệ thống quản lý và vận chuyển kho hàng Logistics” để đưa ra giải pháp tốt hơn cho hoạt động quản lý của doanh nghiệp..

## 1.2. Mục tiêu đề tài

Mục tiêu của đề tài là xây dựng một website quản lý kho vận tập trung, giúp doanh nghiệp vừa và nhỏ số hóa quy trình logistics. Hệ thống cung cấp các chức năng cốt lõi như quản lý sản phẩm, xử lý xuất/nhập kho, chỉ định đơn hàng và giám sát lộ trình vận chuyển qua bản đồ theo thời gian thực, phục vụ ba nhóm người dùng chính: Quản trị viên, Nhân viên kho và Tài xế. Kết quả cần đạt được gồm hai phương diện: Về mặt hệ thống, sản phẩm là một website hoàn chỉnh giúp tối ưu hóa logistics, giảm sai sót thủ công, quản lý tồn kho chính xác và giám sát vận đơn theo thời gian thực. Về mặt công nghệ, đề tài áp dụng thành công Python và Django cho backend, đồng thời tích hợp Firebase Firestore (CSDL NoSQL thời gian thực), Firebase Authentication (xác thực, phân quyền), cùng các API bản đồ (Nominatim, OSRM) cho chức năng định vị.

## 1.3. Phạm vi đề tài

Để đảm bảo tính khả thi, đề tài được giới hạn trong phạm vi nghiệp vụ logistics cơ bản, tập trung xây dựng ứng dụng web cho ba đối tượng người dùng chính: Quản trị viên (Admin), Nhân viên kho (Staff), và Tài xế (Deliver). Hệ thống bao quát hai luồng nghiệp vụ chính là Quản lý Sản phẩm/Kho bãi và Điều phối/Theo dõi vận chuyển.

Về chức năng, hệ thống hỗ trợ toàn bộ vòng đời đơn hàng. Admin và Staff thực hiện quản lý sản phẩm (CRUD), nhập kho (tăng số lượng) và xuất kho (tạo phiếu, trừ số lượng, phân công tài xế). Tài xế sử dụng giao diện riêng để nhận đơn, kích hoạt theo dõi GPS thời gian thực, và xác nhận hoàn thành, sau đó hệ thống sẽ tự động lưu lại lịch sử.

Về kỹ thuật, đề tài sử dụng framework Django (Python) cho backend, kết hợp với các dịch vụ của Firebase (Firestore cho cơ sở dữ liệu NoSQL, Authentication để xác thực). Điểm nhấn công nghệ là khả năng theo dõi vị trí và vẽ lộ trình, được tích hợp qua thư viện Leaflet.js cùng các API công cộng như Nominatim và OSRM.

#### **1.4 Đối tượng nghiên cứu**

Đối tượng nghiên cứu chính của đề tài là các nghiệp vụ logistics cốt lõi và các công nghệ phần mềm được sử dụng để tự động hóa và theo dõi các nghiệp vụ đó.

Về nghiệp vụ, đề tài tập trung nghiên cứu bốn mảng: Quản lý Kho hàng, Quản lý Vận chuyển, Hệ thống Phân quyền (qua Firebase Authentication), và Theo dõi Vị trí (thu thập dữ liệu GeoPoint). Về công nghệ, đề tài áp dụng framework Django (MVT), nghiên cứu cơ sở dữ liệu Firestore (cấu trúc NoSQL và GeoPoint), đồng thời tích hợp thư viện Leaflet.js và các API mã nguồn mở để theo dõi thời gian thực..

#### **1.5. Phương pháp nghiên cứu**

Để hoàn thành đề tài, nhóm đã kết hợp hai phương pháp nghiên cứu chính. Đầu tiên, nhóm thu thập thông tin và tài liệu, nghiên cứu lý thuyết logistics và các công nghệ áp dụng như Firebase (Firestore, Authentication) cùng các API bản đồ (Leaflet, Nominatim, OSRM) để giải quyết bài toán định vị. Tiếp theo, nhóm dùng phương pháp phân tích hướng đối tượng để xác định các thực thể chính (Sản phẩm,

Đơn hàng, Người dùng) và áp dụng kiến trúc MVT của Django để phân tách luồng xử lý, đảm bảo mã nguồn dễ bảo trì và mở rộng.

Cuối cùng, phương pháp thực nghiệm được triển khai thông qua việc xây dựng sản phẩm theo mô hình Agile/Lặp (Iterative Development). Quy trình này cho phép nhóm phát triển và kiểm thử cục bộ các module chức năng (như Quản lý Sản phẩm, Theo dõi Vị trí) một cách liên tục, giúp phát hiện và khắc phục lỗi kịp thời, đảm bảo sản phẩm cuối cùng đáp ứng mục tiêu đề ra.

## **1.6. Bố cục đề tài**

Phần còn lại của báo cáo tiểu luận môn học này được tổ chức thành ba chương chính, theo sau là phần Kết luận, Tài liệu tham khảo.

Về chương 2, nhóm trình bày nền tảng công nghệ được lựa chọn, bao gồm việc nghiên cứu chuyên sâu về Framework Django của Python và cơ chế tích hợp với Firebase (Firestore và Authentication). Chương này cũng tập trung phân tích cách sử dụng thư viện Leaflet.js và các API bản đồ mở như Nominatim/OSRM để xây dựng tính năng theo dõi vị trí và vẽ lộ trình vận chuyển.

Trong chương 3, nhóm đưa ra phần trình bày chi tiết về quá trình xây dựng hệ thống, phân tích yêu cầu nghiệp vụ cho ba tác nhân chính (Admin, Staff, Deliver), mô tả cách Thiết kế Cơ sở dữ liệu trên Firestore và triển khai các module chức năng, triển khai thành công module Quản lý Sản phẩm, Quản lý Nhập/Xuất kho, và chức năng Theo dõi đơn hàng thời gian thực thông qua việc cập nhật tọa độ GPS liên tục từ thiết bị của tài xế.

Cuối cùng, phần Kết luận sẽ tổng hợp những đóng góp chính mà đề tài đã đạt được và đánh giá mức độ hoàn thành các mục tiêu ban đầu, đồng thời đề xuất Hướng phát triển sản phẩm trong tương lai, như việc tích hợp thuật toán tối ưu hóa lộ trình hoặc phát triển các công cụ báo cáo phân tích dữ liệu chuyên sâu hơn.

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

### 2.1. Ngôn ngữ lập trình Python

Ngôn ngữ lập trình Python là cơ sở lý thuyết và công cụ triển khai chính cho toàn bộ hệ thống backend của dự án. Python là ngôn ngữ lập trình bậc cao, thông dịch (interpreted), hướng đối tượng (object-oriented) và có kiểu dữ liệu động (dynamically typed).



**Hình 1 - Ngôn ngữ lập trình Python**

Các đặc điểm nổi bật của Python khiến nó trở thành lựa chọn phù hợp cho đề tài này bao gồm: Cú pháp Đơn giản, Rõ ràng: Cú pháp của Python gần gũi với ngôn ngữ tự nhiên, giúp giảm thời gian học và phát triển. Điều này được thể hiện rõ trong các tệp logic của dự án như `views.py`, nơi các nghiệp vụ phức tạp được viết một cách tường minh; Tính Thông dịch: Python là ngôn ngữ thông dịch, nghĩa là mã nguồn được thực thi từng dòng mà không cần biên dịch sang mã máy.

### 2.2. Framework Django

Django là một framework phát triển web Python bậc cao, tuân theo triết lý "Don't Repeat Yourself" (DRY). Nó được xây dựng để thúc đẩy việc phát triển nhanh chóng, an toàn và có thể mở rộng. Django sử dụng kiến trúc MVT (Model-View-Template), một biến thể của MVC (Model-View-Controller). Trong dự án này, Django đóng vai trò là bộ khung xương chính, xử lý các yêu cầu HTTP, định tuyến URL, và kết xuất giao diện người dùng.



**Hình 2 - Framework Django**

Django cung cấp nhiều công cụ "làm sẵn" rất hữu ích cho dự án. Ví dụ, nhóm đã sử dụng hệ thống Form của Django (như ProductForm và ExportsForm) để đảm bảo rằng mọi dữ liệu người dùng nhập vào đều an toàn và chính xác trước khi xử lý. Để giúp người dùng không phải đăng nhập lại ở mỗi trang, dự án dùng tính năng Session để lưu trữ thông tin của họ. Về mặt bảo mật, thẻ được thêm vào các biểu mẫu như một "con dấu" bảo vệ, giúp ngăn chặn các cuộc tấn công mạo danh phổ biến.

### **2.3. Môi trường lập trình Visual Studio Code**

Visual Studio Code (VS Code) là một trình soạn thảo mã nguồn (source-code editor) nhẹ, miễn phí và mạnh mẽ do Microsoft phát triển. Nó chạy được trên đa nền tảng (Windows, macOS, Linux) và đã trở thành công cụ rất phổ biến trong cộng đồng lập trình.



**Hình 3 - Visual Studio Code**

Sức mạnh của VS Code đến từ sự kết hợp của nhiều tính năng. Nổi bật nhất là tính năng IntelliSense, cung cấp khả năng hỗ trợ mã thông minh như tự động hoàn thành và tô màu cú pháp. Bên cạnh đó, hệ sinh thái mở rộng (Marketplace) không lồ cho phép lập trình viên tùy chỉnh và bổ sung chức năng cho hầu hết mọi ngôn ngữ, bao gồm cả Python và Django. VS Code cũng tích hợp một trình gỡ lỗi (debugger) trực quan, cho phép đặt điểm dừng và theo dõi biến. Ngoài ra, việc tích hợp sẵn Git để quản lý phiên bản và một cửa sổ Terminal để chạy lệnh trực tiếp trong trình soạn thảo giúp quy trình làm việc trở nên liền mạch.

## **2.4. Nền tảng Google Firebase**

Firebase là một nền tảng phát triển ứng dụng toàn diện (Backend-as-a-Service - BaaS) của Google, cung cấp một bộ công cụ mạnh mẽ để xây dựng các ứng dụng chất lượng cao mà không cần quản lý cơ sở hạ tầng máy chủ phức tạp. Trong dự án này, nhóm đã sử dụng hai dịch vụ chính: Cloud Firestore để lưu trữ dữ liệu và Firebase Authentication để quản lý người dùng.



**Hình 4 - Google Firebase**

Cloud Firestore là một cơ sở dữ liệu NoSQL, linh hoạt và có khả năng mở rộng, được lưu trữ trên đám mây. Nó không sử dụng mô hình bảng/hàng truyền thống mà tổ chức dữ liệu theo cấu trúc Collections và Documents

Firebase Authentication là dịch vụ quản lý người dùng, cung cấp các tính năng xác thực an toàn và dễ sử dụng. Dự án sử dụng dịch vụ này để quản lý toàn bộ quy trình từ đăng ký, đăng nhập đến duy trì phiên làm việc.

## 2.5. Leaflet.js và OpenStreetMap

Để triển khai các tính năng cốt lõi của hệ thống logistics như hiển thị lộ trình, theo dõi vị trí tài xế, và quản lý các điểm giao nhận, dự án đã kết hợp hai công nghệ bản đồ mã nguồn mở hàng đầu là Leaflet.js và OpenStreetMap.

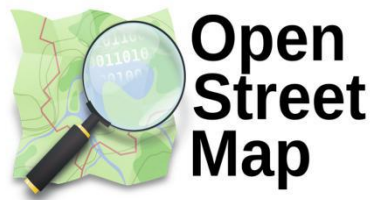


**Hình 5 - Thư viện Leaflet**

Nhiệm vụ chính của Leaflet trong hệ thống là: Hiển thị bản đồ: Khởi tạo và quản lý khung nhìn bản đồ, cho phép người dùng phóng to, thu nhỏ và di chuyển bản đồ một cách mượt mà; Điểm đánh dấu (Markers): Sử dụng các biểu tượng tùy chỉnh để đánh dấu vị trí của Kho hàng, điểm đến của khách hàng, và vị trí hiện tại của xe tải; Vẽ lộ trình: Tiếp nhận dữ liệu đường đi (GeoJSON) từ dịch vụ định tuyến và vẽ chúng lên bản đồ dưới dạng đường kẻ trực quan.

OpenStreetMap (OSM) là nguồn dữ liệu bản đồ toàn cầu miễn phí, được xây dựng và duy trì bởi cộng đồng. Dự án sử dụng OSM làm lớp bản đồ cơ sở (base map tiles), cung cấp nền tảng địa lý chi tiết cho giao diện theo dõi. Sự lựa chọn này đảm bảo độ chính xác, tính cập nhật cao của dữ liệu bản đồ mà không phải chịu chi phí bản quyền. OSM được tích hợp vào Leaflet thông qua việc gọi URL để tải các ô bản đồ.





**Hình 6 - OpenStreetMap**

Trong hệ thống theo dõi đơn hàng, OSM được tích hợp thông qua Leaflet.js để tải về các lớp bản đồ, tạo nên khung nền chi tiết và đáng tin cậy để hiển thị vị trí tĩnh (kho hàng, điểm đến) và vị trí động (xe giao hàng).

## **2.6. Framework Bootstrap**

Bootstrap là một framework HTML, CSS và JavaScript mã nguồn mở hàng đầu, được sử dụng để phát triển các giao diện web đáp ứng (responsive design) một cách nhanh chóng và hiệu quả. Nó cung cấp sẵn một thư viện lớn các thành phần giao diện và các tiện ích CSS được chuẩn hóa, giúp đồng bộ hóa thiết kế và đảm bảo trải nghiệm người dùng nhất quán trên mọi kích thước màn hình.



**Hình 7 - Framework Bootstrap**

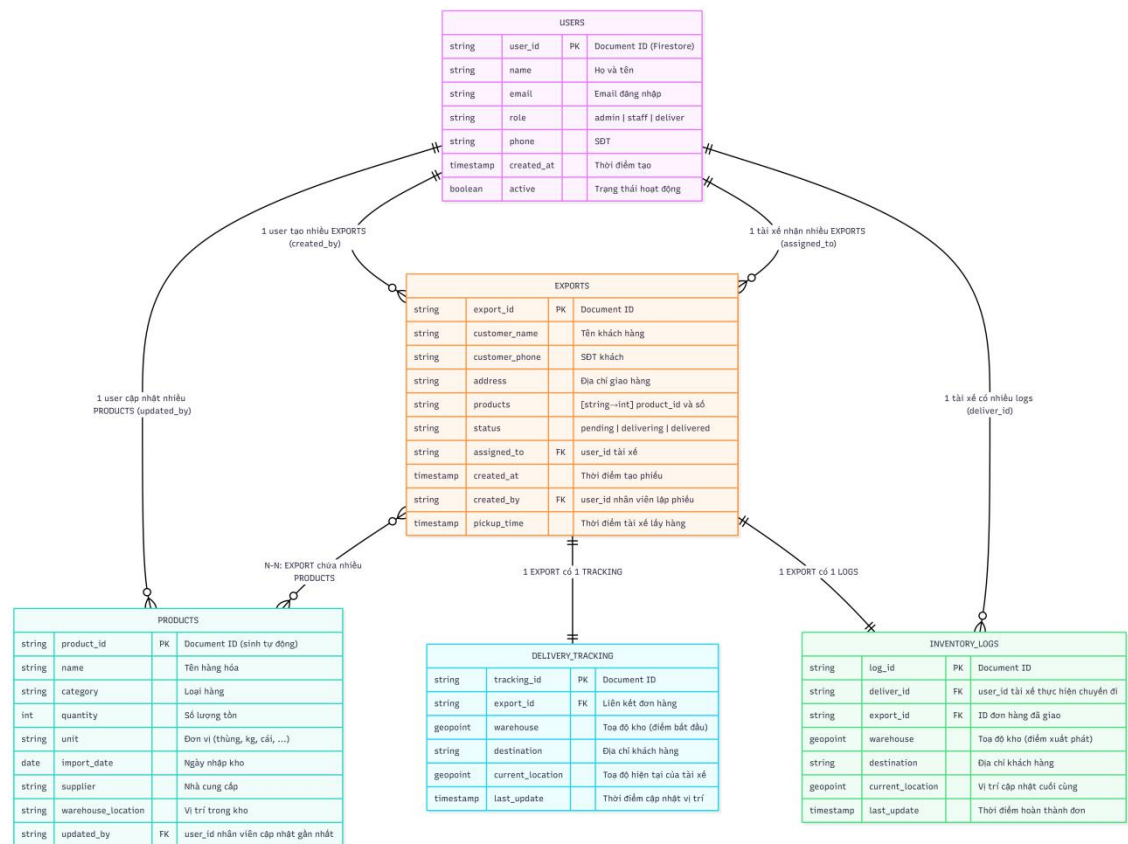
Việc áp dụng Bootstrap mang lại hai lợi ích lớn, đầu tiên là thiết kế Đáp ứng: Bootstrap sử dụng hệ thống lưới (Grid System) dựa trên Flexbox, cho phép các bố cục như trang đăng nhập, đăng ký hay dashboard tự động điều chỉnh linh hoạt, hiển thị tối ưu trên cả máy tính và điện thoại di động

# CHƯƠNG 3. PHÂN TÍCH HỆ THỐNG VÀ XÂY DỰNG SẢN PHẨM

## 3.1. Phân tích hệ thống

### 3.1.1. Sơ đồ khối của hệ thống

Sơ đồ này phác thảo Mô hình Dữ liệu (Data Model) NoSQL được triển khai trên Cloud Firestore, làm nền tảng cho hệ thống logistics. Mô hình bao gồm 5 bộ sưu tập (Collection) chính: USERS (quản lý người dùng), PRODUCTS (quản lý sản phẩm tồn kho), EXPORTS (quản lý đơn hàng xuất kho), DELIVERY TRACKINGS (theo dõi vị trí thời gian thực), và INVENTORY\_LOGS (lưu trữ lịch sử giao hàng). Cấu trúc này xác định cách thức dữ liệu được tổ chức và liên kết với nhau để phục vụ các chức năng nghiệp vụ.

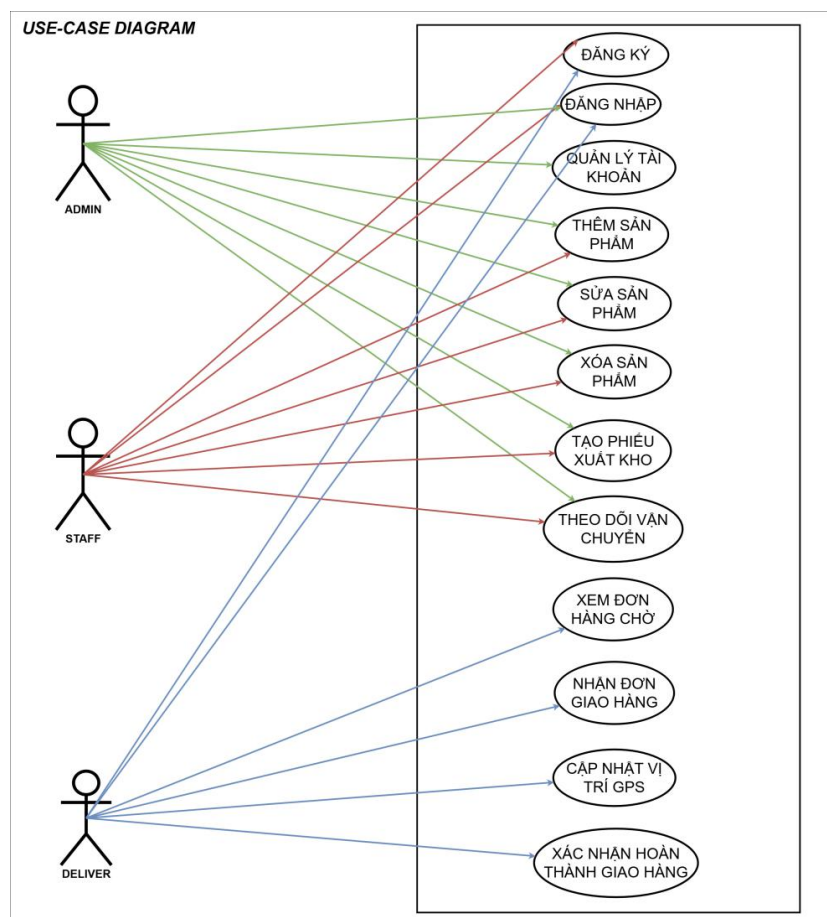


Hình 8 - Sơ đồ khối của hệ thống

Các mối quan hệ logic giữa các bộ sưu tập được thể hiện rõ ràng. Một USER có thể tạo nhiều đơn hàng EXPORTS (thông qua 'created\_by') và được gán để vận chuyển chúng (thông qua 'assigned\_to'). Đơn hàng EXPORTS liên kết trực tiếp với một bản ghi DELIVERY\_TRACKINGS để giám sát lộ trình. Điểm nhấn kỹ thuật của mô hình là việc sử dụng kiểu dữ liệu GeoPoint cho các trường warehouse, destination, và current\_location, tạo ra xương sống cho tính năng định vị và theo dõi bản đồ.

### 3.1.2. Use-Case Diagram

Sơ đồ Use-Case sẽ trình bày các chức năng chính của hệ thống Logistics và mối quan hệ giữa chúng với ba vai trò người dùng cốt lõi (Tác nhân): ADMIN, STAFF (Nhân viên kho), và DELIVER (Tài xế).



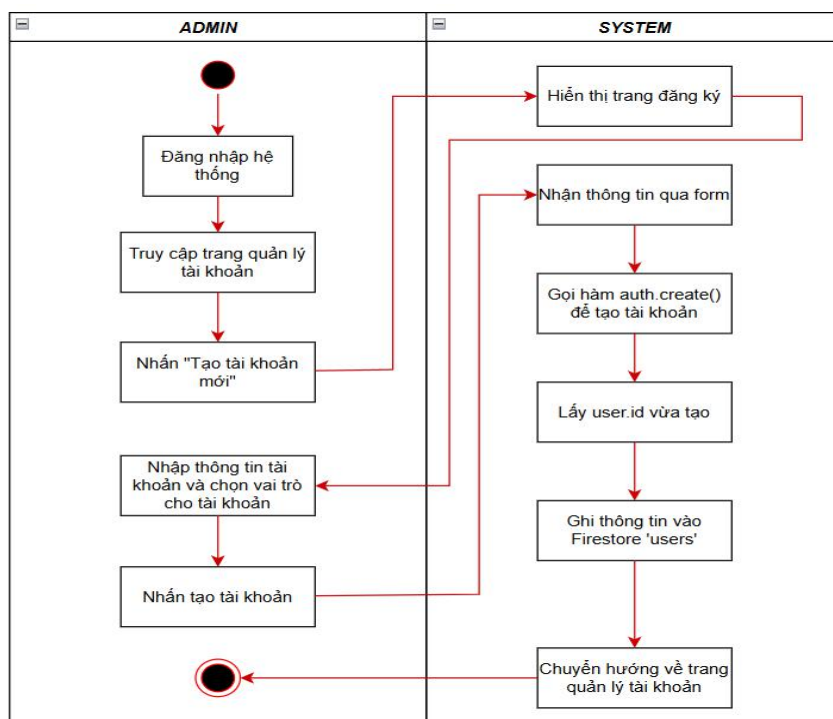
**Hình 9 - Sơ đồ Use-Case**

Sơ đồ này chia các chức năng thành ba nhóm nghiệp vụ chính. Nhóm Quản lý Kho (bao gồm Thêm, Sửa, Xóa Sản phẩm và Tạo Phiếu Xuất Kho) là trách nhiệm chung của cả ADMIN và STAFF. Mặc dù mọi tác nhân đều phải Đăng nhập, nhưng chỉ ADMIN mới có quyền Quản lý Tài khoản. Nhóm Vận chuyển và Theo dõi chủ yếu do DELIVER đảm nhiệm, bắt đầu bằng việc Xem Đơn Hàng Chờ và Nhận Đơn Giao Hàng, sau đó liên tục Cập nhật Vị trí GPS. Dữ liệu vị trí này cho phép STAFF và ADMIN thực hiện chức năng Theo dõi Vận chuyển, và quy trình kết thúc khi DELIVER Xác nhận Hoàn thành Giao hàng.

### 3.1.3. Activity Diagram

#### 3.1.3.1. Sơ đồ hoạt động đăng ký tài khoản

Sơ đồ Hoạt động này mô tả luồng công việc chi tiết cho ca sử dụng Tạo tài khoản (chức năng của Admin), phân tách rõ ràng trách nhiệm giữa Tác nhân (ADMIN) và Hệ thống (SYSTEM).

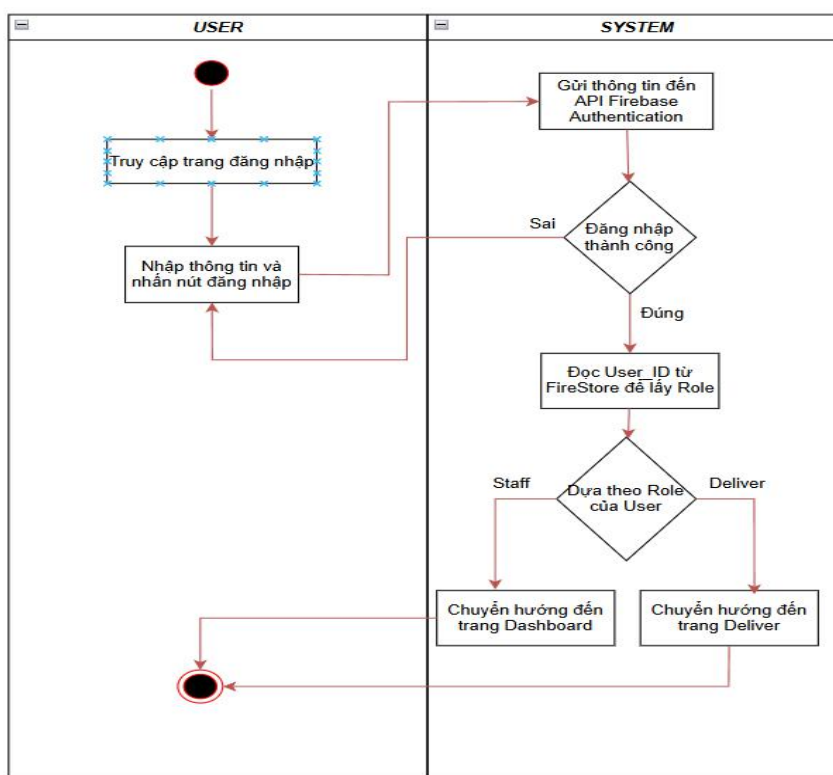


**Hình 10 - Sơ đồ hoạt động đăng ký tài khoản**

Quy trình bắt đầu khi ADMIN đã đăng nhập và truy cập vào trang "Quản lý tài khoản", sau đó nhấn nút "Tạo tài khoản mới". Hệ thống hiển thị trang đăng ký. ADMIN tiến hành nhập thông tin (tên, email, mật khẩu) và quan trọng nhất là chọn Vai trò (Role) cho tài khoản sắp tạo.

### 3.1.3.2. Sơ đồ hoạt động đăng nhập hệ thống

Sơ đồ Hoạt động này mô tả luồng kiểm soát chi tiết cho việc Đăng nhập, phân chia trách nhiệm giữa Tác nhân (USER) và Hệ thống (SYSTEM). Quy trình bắt đầu khi USER Truy cập trang đăng nhập và Nhập thông tin (email, mật khẩu) rồi nhấn nút đăng nhập. Hệ thống tiếp nhận thông tin và Gửi thông tin đến API Firebase Authentication để xác thực.



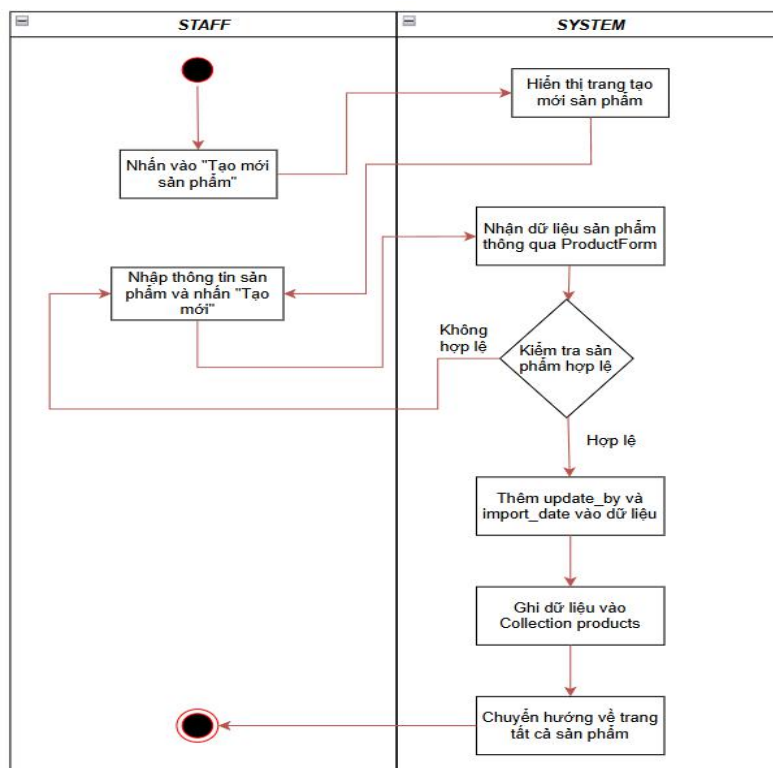
**Hình 11 - Sơ đồ hoạt động đăng nhập hệ thống**

Tùy vào kết quả xác thực, hệ thống sẽ rẽ nhánh: Nếu xác thực Sai, hệ thống hiển thị thông báo lỗi và đưa người dùng quay lại bước nhập thông tin. Nếu xác

thực Đúng, hệ thống tiếp tục luồng bằng cách Đọc Doc ID (User ID) từ Firestore để lấy Vai trò (Role) của người dùng. Dựa trên vai trò này, hệ thống sẽ thực hiện rẽ nhánh lần thứ hai để Chuyển hướng người dùng đến trang Dashboard nếu Role là Staff hoặc đến trang Deliver nếu Role là Deliver. Quy trình kết thúc sau khi chuyển hướng thành công.

### 3.1.3.3. Sơ đồ hoạt động quản lý sản phẩm

Sơ đồ Hoạt động này mô tả luồng công việc để tạo mới một sản phẩm trong hệ thống, tập trung vào sự phối hợp giữa Nhân viên Kho (STAFF) và Hệ thống (SYSTEM). Quy trình bắt đầu khi STAFF nhấn vào "Tạo mới sản phẩm", khiến hệ thống Hiển thị trang tạo mới sản phẩm. STAFF sau đó Nhập thông tin sản phẩm và nhấn "Tạo mới", khiến hệ thống Nhận dữ liệu sản phẩm thông qua ProductForm. Hệ thống tiếp theo Kiểm tra sản phẩm hợp lệ. Nếu Không hợp lệ, hệ thống chuyển hướng người dùng về trang nhập thông tin sản phẩm và nhấn "Tạo mới". Nếu Hợp lệ, hệ thống Thêm update\_by và import\_date vào dữ liệu, Ghi dữ liệu vào Collection products, và Chuyển hướng về trang tất cả sản phẩm.



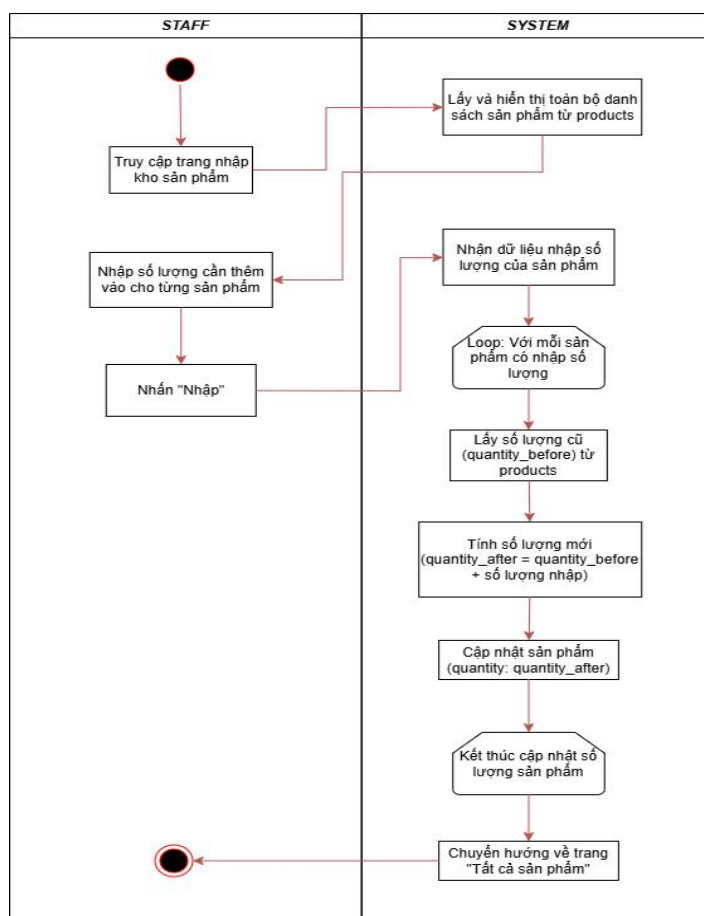
**Hình 12 - Sơ đồ hoạt động quản lý sản phẩm**

Phía SYSTEM tiếp nhận dữ liệu thông qua ProductForm của Django để chuẩn hóa và tiến hành Kiểm tra sản phẩm hợp lệ. Nếu dữ liệu Không hợp lệ, hệ

thống sẽ đưa STAFF quay lại bước nhập thông tin để sửa lỗi. Nếu dữ liệu Hợp lệ, hệ thống sẽ tự động thêm các trường quan trọng như `update_by` (người tạo/cập nhật) và `import_date` (ngày nhập) vào dữ liệu. Cuối cùng, dữ liệu được Ghi vào Collection products trên Firestore, và hệ thống Chuyển hướng về trang tất cả sản phẩm, hoàn tất quá trình tạo sản phẩm.

#### 3.1.3.4. Sơ đồ hoạt động nhập kho số lượng sản phẩm

Sơ đồ Hoạt động này mô tả luồng công việc để nhập hàng hóa vào kho, tập trung vào sự tương tác giữa Nhân viên Kho (STAFF) và Hệ thống (SYSTEM).



**Hình 13 - Sơ đồ hoạt động nhập kho số lượng sản phẩm**

Quy trình bắt đầu khi STAFF truy cập trang nhập kho. Hệ thống phản hồi bằng cách lấy và hiển thị toàn bộ danh sách sản phẩm từ collection products. STAFF nhập số lượng cần thêm cho từng sản phẩm và nhấn "Nhập". Phía

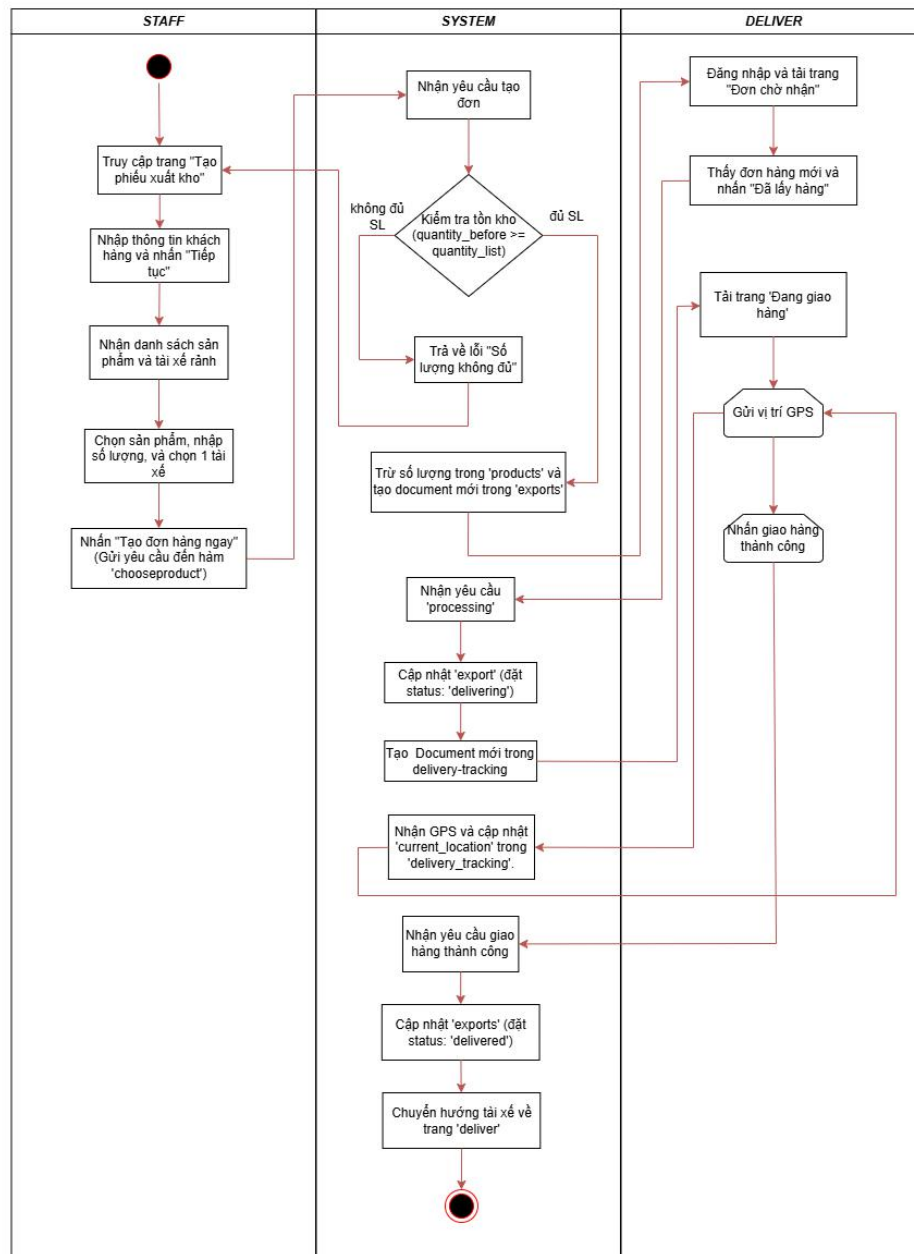
SYSTEM sẽ lặp (Loop) qua các sản phẩm được nhập, lấy số lượng cũ (quantity\_before), tính toán số lượng mới ( $\text{quantity\_after} = \text{quantity\_before} + \text{số lượng nhập}$ ), và cập nhật lại số lượng mới vào products. Sau khi hoàn tất cập nhật, hệ thống chuyển hướng về trang "Tất cả sản phẩm".

#### **3.1.3.5. Sơ đồ hoạt động xuất kho và giao hàng**

Sơ đồ này mô tả một luồng công việc phức tạp liên quan đến ba tác nhân chính: STAFF (Nhân viên kho), SYSTEM (Hệ thống), và DELIVER (Tài xế). Quy trình bắt đầu với STAFF truy cập trang tạo phiếu, Nhận yêu cầu tạo đơn từ hệ thống, và Nhập thông tin khách hàng. Sau đó, hệ thống hiển thị danh sách sản phẩm và tài xế rảnh, cho phép STAFF Chọn sản phẩm, nhập số lượng, và chọn 1 tài xế.

Khi STAFF "Tạo đơn hàng ngay", hệ thống tiến hành Kiểm tra tồn kho bằng cách so sánh số lượng cũ (quantity\_before) với số lượng yêu cầu. Nếu số lượng không đủ, hệ thống trả về lỗi; nếu đủ, hệ thống Trừ số lượng trong products và tạo document mới trong Collection exports để hoàn tất việc xuất kho.





**Hình 14 - Sơ đồ hoạt động quy trình xuất kho và giao hàng**

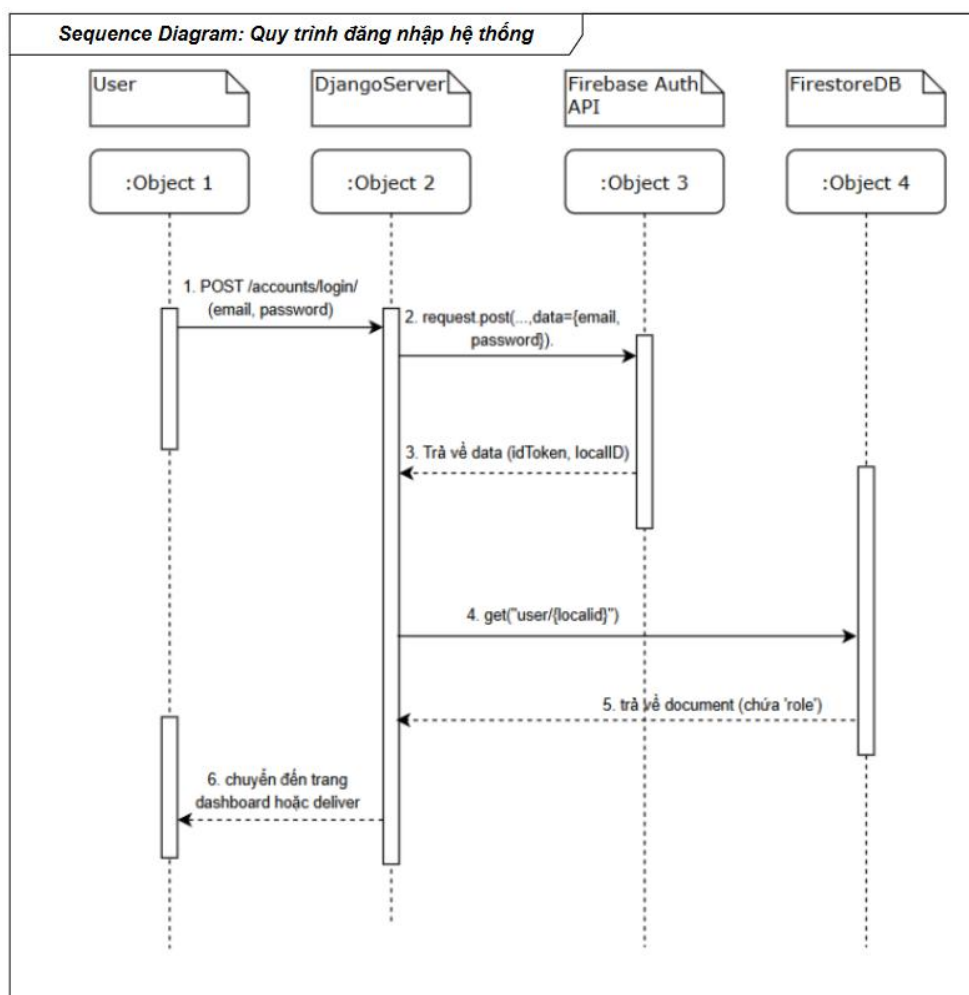
Giai đoạn tiếp theo là giao hàng, bắt đầu khi DELIVER truy cập trang "Đơn chờ nhận" và Thấy đơn hàng mới. Tài xế nhận đơn bằng cách nhấn "Đã lấy hàng", gửi Yêu cầu processing tới hệ thống. Hệ thống phản hồi bằng cách Cập nhật đơn hàng (exports) với status là 'delivering' và Tạo Document mới trong delivery\_tracking. Trong suốt quá trình vận chuyển, DELIVER liên tục Gửi vị trí GPS, và hệ thống Cập nhật current\_location trong delivery\_tracking. Khi giao hàng thành công, tài xế Nhận giao hàng thành công (Gửi yêu cầu giao hàng thành công).

Cuối cùng, hệ thống Cập nhật exports với status 'delivered' và Chuyển hướng tài xế về trang 'deliver' để kết thúc toàn bộ quy trình.

### 3.1.4. Sequence Diagram

#### 3.1.4.1. Sơ đồ tuần tự quy trình đăng nhập hệ thống

Sơ đồ tuần tự này mô tả luồng giao tiếp và thứ tự các bước giữa các thành phần chính của hệ thống trong quá trình đăng nhập. Các thành phần tham gia bao gồm User (người dùng cuối), DjangoServer (máy chủ backend), Firebase Auth API (dịch vụ xác thực), và FirestoreDB (cơ sở dữ liệu).



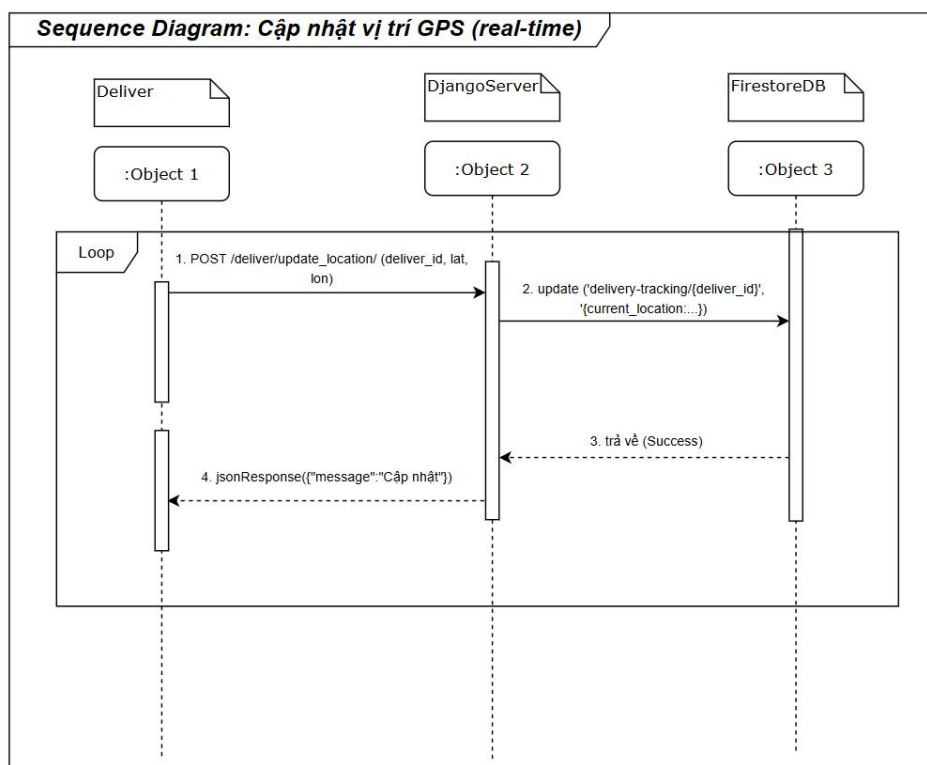
**Hình 15 - Sơ đồ tuần tự quy trình đăng nhập hệ thống**

Quy trình bắt đầu khi User gửi yêu cầu POST chứa email và mật khẩu đến endpoint /accounts/login/ của DjangoServer (Bước 1). DjangoServer sau đó chuyển

tiếp yêu cầu xác thực này đến Firebase Auth API (Bước 2). Firebase Auth xác thực thông tin và trả về các định danh cần thiết như idToken và localId (User ID) cho DjangoServer (Bước 3). DjangoServer tiếp tục truy vấn FirestoreDB bằng cách sử dụng localId để lấy document chứa thông tin vai trò (role) của người dùng (Bước 4 và 5). Cuối cùng DjangoServer sẽ chuyển hướng trình duyệt của User đến trang dashboard hoặc deliver, hoàn tất quá trình đăng nhập và phân quyền (Bước 6).

#### 3.1.4.2. Sơ đồ tuần tự quy trình cập nhật vị trí GPS

Sơ đồ tuần tự này mô tả luồng thông tin liên tục và tuần hoàn (Loop) khi Tài xế (Deliver) cập nhật vị trí thời gian thực về máy chủ. Các thành phần chính tham gia vào quy trình là Deliver (nguồn GPS), DjangoServer (backend xử lý), và FirestoreDB.



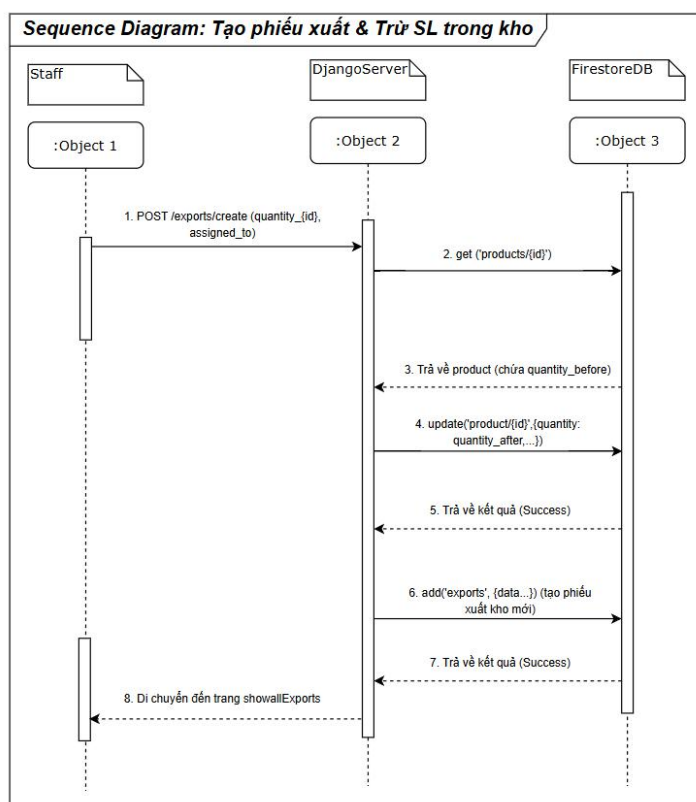
**Hình 16 - Sơ đồ tuần tự quy trình Cập nhật vị trí GPS (real-time)**

Trong vòng lặp cập nhật liên tục, Deliver gửi một yêu cầu POST tới endpoint /deliver/update\_location/, kèm theo ID tài xế (deliver\_id), vĩ độ (lat), và kinh độ

(lon) hiện tại (Bước 1). DjangoServer nhận yêu cầu và ngay lập tức thực hiện lệnh update lên FirestoreDB (Bước 2), cập nhật trường `current_location` trong document theo `deliver_id`. Sau khi cập nhật thành công (Bước 3), DjangoServer trả về phản hồi `JsonResponse` (Bước 4) để xác nhận việc cập nhật vị trí đã hoàn tất, chuẩn bị cho lần cập nhật tiếp theo trong vòng lặp.

### 3.1.4.3. Sơ đồ tuần tự quy trình tạo phiếu xuất và trừ số lượng trong kho

Sơ đồ Trình tự này trình bày luồng giao tiếp và thứ tự các bước giữa Staff (Nhân viên kho), DjangoServer (máy chủ backend), và FirestoreDB (cơ sở dữ liệu) để xử lý việc xuất kho. Quy trình bắt đầu khi Staff gửi yêu cầu POST tới endpoint `/exports/create/`, mang theo ID sản phẩm (`quantity_id`) và ID tài xế được chỉ định (`assigned_to`) (Bước 1). DjangoServer nhận yêu cầu và bắt đầu tương tác với cơ sở dữ liệu: đầu tiên, nó get thông tin sản phẩm từ Firestore (Bước 2, 3) để lấy `quantity_before` (số lượng tồn kho cũ).

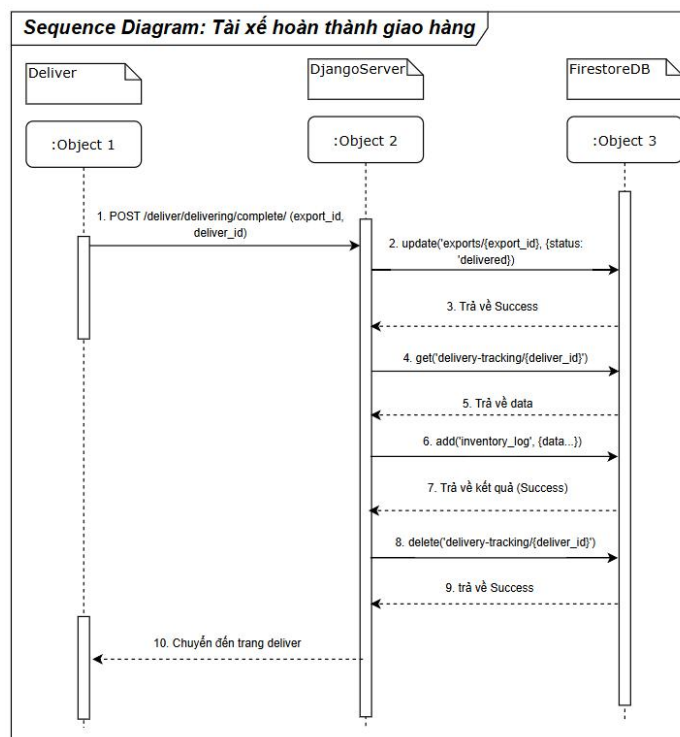


### Hình 17 - Sơ đồ tuần tự tạo phiếu xuất và trừ số lượng trong kho

Sau khi có số lượng cũ, DjangoServer tính toán số lượng mới (quantity\_after) và tiến hành update sản phẩm trong Firestore với số lượng mới (Bước 4, 5), hoàn tất việc trừ số lượng trong kho. Tiếp theo, để tạo phiếu xuất kho, DjangoServer thực hiện lệnh add một document mới vào Collection exports (Bước 6, 7). Khi tất cả các bước cập nhật và thêm dữ liệu vào Firestore đều thành công, DjangoServer sẽ Chuyển hướng trình duyệt của Staff đến trang showallExports, kết thúc quy trình.

#### 3.1.4.4. Sơ đồ tuần tự hoàn thành giao hàng

Sơ đồ này mô tả luồng giao tiếp giữa Tài xế (Deliver), DjangoServer, và FirestoreDB khi đơn hàng được xác nhận hoàn thành. Quy trình bắt đầu khi Deliver gửi yêu cầu POST chứa export\_id và deliver\_id tới DjangoServer (Bước 1). DjangoServer ngay lập tức thực hiện lệnh update lên Collection exports trong Firestore, đặt trạng thái (status) của đơn hàng là 'delivered' (Bước 2, 3). DjangoServer ngay lập tức thực hiện lệnh update lên Collection exports trong Firestore, đặt trạng thái (status) của đơn hàng là 'delivered' (Bước 2, 3).



Hình 18 - Sơ đồ tuần tự hoàn thành giao hàng

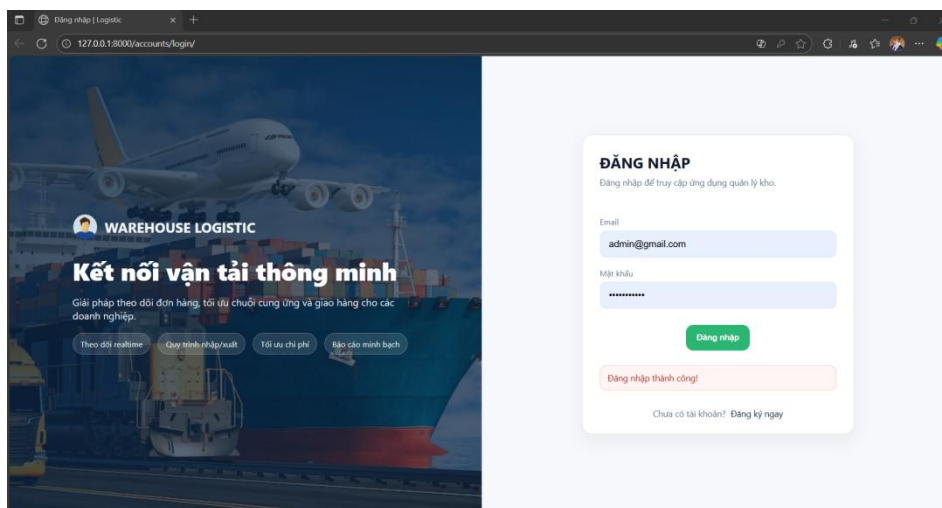
Sau khi cập nhật trạng thái đơn hàng, DjangoServer tiến hành các bước dọn dẹp và ghi lại lịch sử. Hệ thống get dữ liệu theo dõi cuối cùng từ Collection delivery-tracking (Bước 4, 5), sau đó sử dụng dữ liệu này để add một bản ghi vĩnh viễn vào Collection inventory\_logs (Bước 6, 7). Cuối cùng, hệ thống delete document của tài xế khỏi Collection delivery-tracking (Bước 8, 9) để loại bỏ trạng thái "đang giao" và Chuyển hướng tài xế đến trang deliver để kết thúc quy trình (Bước 10).

### 3.2. Xây dựng giao diện sản phẩm

Các giao diện được thiết kế theo nguyên tắc đơn giản, trực quan, và nhất quán màu sắc giúp người dùng thao tác nhanh chóng. Mỗi giao diện tương ứng với một chức năng nghiệp vụ chính của hệ thống.

#### 3.2.1. Giao diện Đăng nhập hệ thống

Cho phép người dùng (Admin, Staff, hoặc Deliver) đăng nhập vào hệ thống. Hệ thống kiểm tra thông tin xác thực thông qua Firebase Authentication, sau đó tự động điều hướng đến trang tương ứng với vai trò của người dùng.



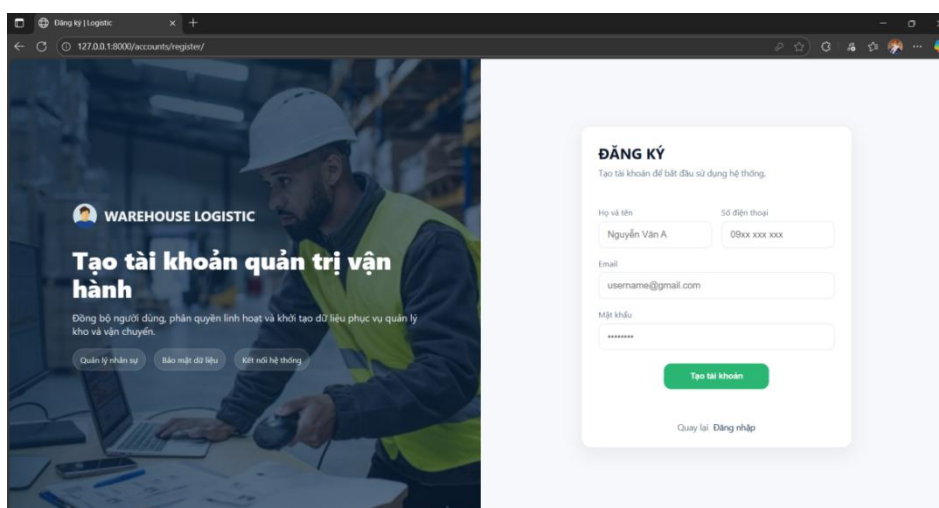
**Hình 19 - Giao diện đăng nhập hệ thống**

Giao diện bao gồm hai ô nhập Email và Mật khẩu, nút “Đăng nhập” và liên kết “Đăng ký tài khoản mới”. Giao diện được thiết kế bằng Bootstrap 5, tông màu chủ đạo trắng – xanh dương, đảm bảo tính thẩm mỹ và dễ đọc. Khi người dùng

nhập sai thông tin, hệ thống hiển thị thông báo lỗi; nếu đúng, người dùng được chuyển đến trang Dashboard hoặc Deliver tương ứng với vai trò.

### 3.2.2. Giao diện Đăng ký tài khoản

Đây là giao diện chỉ dành cho Quản trị viên (Admin) để tạo tài khoản mới cho nhân viên. Sau khi tạo thành công, thông tin người dùng (họ tên, số điện thoại, vai trò) được lưu trong Firestore collection "users". Giao diện gồm các trường Họ tên, Số điện thoại, Email, Mật khẩu và một trường "Vai trò" (dạng dropdown) cho phép Admin lựa chọn phân quyền (ví dụ: Staff, Deliver, Admin) cho tài khoản mới.



The screenshot displays a web browser window with the URL `127.0.0.1:8000/accounts/register/`. The page is split into two main sections. On the left, there is a dark-themed banner featuring a warehouse worker in a hard hat and safety vest, with the text 'WAREHOUSE LOGISTIC' and 'Tạo tài khoản quản trị vận hành'. Below this, there is a brief description and three buttons: 'Quản lý nhân sự', 'Bảo mật dữ liệu', and 'Kết nối hệ thống'. On the right, there is a white registration form titled 'ĐĂNG KÝ' with the subtitle 'Tạo tài khoản để bắt đầu sử dụng hệ thống.' The form contains four input fields: 'Họ và tên' (filled with 'Nguyễn Văn A'), 'Số điện thoại' (filled with '090x xxx xxx'), 'Email' (filled with 'username@gmail.com'), and 'Mật khẩu' (filled with '\*\*\*\*\*'). A green 'Tạo tài khoản' button is at the bottom of the form, and a link 'Quay lại: Đăng nhập' is below it.

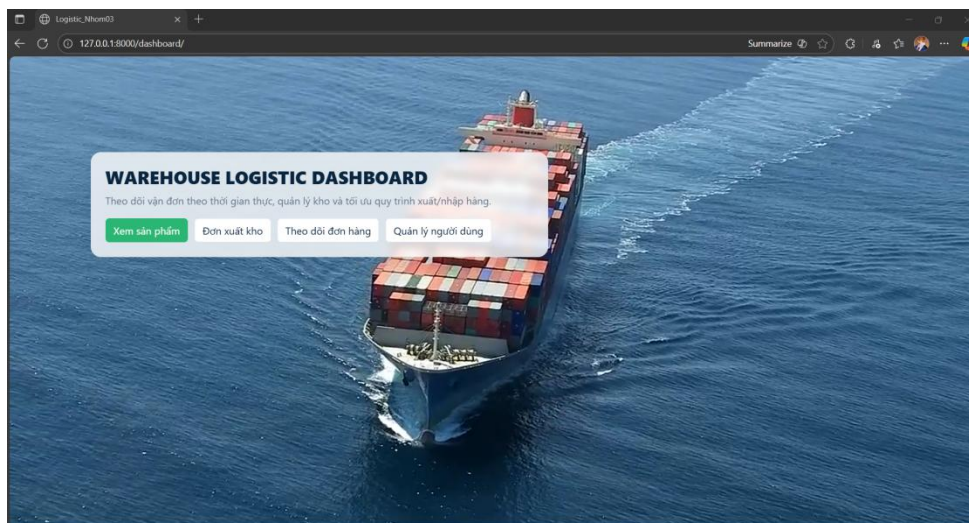
**Hình 20 - Giao diện đăng ký tài khoản**

Khi Admin nhấn "Tạo tài khoản", hệ thống sẽ tạo tài khoản trên Firebase Authentication, ghi thông tin người dùng (bao gồm vai trò được chỉ định) vào collection "users", và sau đó chuyển hướng Admin trở lại trang Quản lý tài khoản.

### 3.2.3. Giao diện Dashboard của hệ thống

Giao diện Dashboard là trang chủ và bảng điều khiển trung tâm, cho phép người dùng (Admin, Staff) truy cập nhanh chóng vào các chức năng nghiệp vụ của hệ thống.





**Hình 21 - Giao diện Dashboard**

Về mặt thiết kế, giao diện sử dụng hình nền video về hoạt động logistics để tăng tính trực quan. Bảng thông tin chính nằm ở trung tâm với tiêu đề WAREHOUSE LOGISTIC DASHBOARD và danh sách các nút điều hướng chức năng. Các nút này dẫn đến các mô-đun chính là Xem sản phẩm, Đơn xuất kho, và Theo dõi đơn hàng. Đặc biệt, nút Quản lý người dùng chỉ được hiển thị khi người dùng có vai trò Admin, đảm bảo phân quyền truy cập tính năng quản lý tài khoản hệ thống.

#### **3.2.4. Giao diện danh sách tất cả sản phẩm trong kho**

Giao diện "Tất cả sản phẩm" là trang quản lý trung tâm cho toàn bộ hàng hóa trong kho. Trang này cho phép người dùng (Admin và Staff) xem nhanh danh sách các sản phẩm đang được lưu trữ, đồng thời cung cấp thanh "Tìm kiếm" để lọc sản phẩm theo tên một cách hiệu quả. Nút "Tạo mới sản phẩm" được đặt ở vị trí nổi bật, dẫn người dùng trực tiếp đến biểu mẫu để nhập thông tin hàng hóa mới.



Logistic Nhóm 03   Sản phẩm   Theo dõi đơn hàng   Công cụ quản kho						
<b>Tất cả sản phẩm</b> Danh sách và tìm kiếm sản phẩm trong kho <div> <input type="text" value="Nhập tên sản phẩm"/> <input type="button" value="Tìm kiếm"/> </div> <div>Tạo mới sản phẩm</div>						
Ngày nhập	Tên sản phẩm	Số lượng	Đơn vị	Cập nhật bởi	Vị trí kho	Thao tác
05:22 Ngày 08 tháng 11 năm 2025	Cà phê Trung Nguyên	450	Hộp	Demo1	A3-B1	<input type="button" value="Sửa"/> <input type="button" value="Xóa"/>
08:09 Ngày 10 tháng 11 năm 2025	Nước ngọt C4 (440ml)	200	Thùng	Dương Đình Thuận	A2-B2	<input type="button" value="Sửa"/> <input type="button" value="Xóa"/>
07:44 Ngày 28 tháng 10 năm 2025	ChocolatePie	10	Thùng	Demo1	A1-B1	<input type="button" value="Sửa"/> <input type="button" value="Xóa"/>
07:48 Ngày 28 tháng 10 năm 2025	Nước ngọt Sipep (330ml)	160	Thùng	Ngô Thành Danh	A1-B1	<input type="button" value="Sửa"/> <input type="button" value="Xóa"/>

Hình 22 - Giao diện tất cả sản phẩm trong kho

Phần chính của giao diện là bảng danh sách, hiển thị các thông tin chi tiết và quan trọng bao gồm "Ngày nhập", "Tên sản phẩm", "Số lượng" tồn kho, "Đơn vị", "Cập nhật bởi" và "Vị trí kho". Tại cột "Thao tác", mỗi sản phẩm đều có các nút chức năng "Sửa" và "Xóa", cho phép người dùng thực hiện đầy đủ các nghiệp vụ CRUD (Tạo, Đọc, Cập nhật, Xóa) để quản lý vòng đời sản phẩm trong cơ sở dữ liệu Firestore.

3.2.5. Giao diện nhập kho các sản phẩm đã có

Giao diện "Nhập kho" cho phép nhân viên cập nhật số lượng tồn kho cho các sản phẩm đã có. Hệ thống hiển thị toàn bộ danh sách sản phẩm, đi kèm ô nhập "Số lượng" cho từng mục, giúp người dùng có thể nhập hàng loạt nhiều mặt hàng cùng lúc.

Logistic Nhóm 03   Sản phẩm   Theo dõi đơn hàng   Công cụ quản kho	
<b>Nhập kho</b> Chon sản phẩm và số lượng để nhập kho <div> <input type="text" value="Nhập tên sản phẩm"/> <input type="button" value="Tìm kiếm"/> </div> <div>Nhập</div>	
<b>Số lượng</b>	<b>Tên sản phẩm</b>
<input type="text"/>	Cà phê Trung Nguyên
<input type="text"/>	Nước ngọt C4 (440ml)
<input type="text"/>	ChocolatePie
<input type="text"/>	Nước ngọt Sipep (330ml)

Hình 23 - Giao diện nhập kho các sản phẩm đã có

Khi nhấn nút "Nhập", backend sẽ lặp qua các sản phẩm được gửi lên. Đối với mỗi sản phẩm, hệ thống đọc số lượng cũ từ Firestore, cộng thêm số lượng mới nhập, và cập nhật lại tổng số lượng (quantity\_after) vào tài liệu sản phẩm, hoàn tất quy trình nhập kho.

### 3.2.6. Giao diện tạo phiếu xuất kho

Giao diện "Tạo phiếu xuất kho" (Bước 1) là nơi nhân viên kho khởi tạo một đơn hàng mới bằng cách nhập các thông tin cơ bản của khách hàng. Các trường bắt buộc bao gồm "Tên khách hàng", "Số điện thoại khách hàng", "Địa chỉ giao hàng" và "Ngày lấy hàng". Khi nhấn "Tiếp tục", dữ liệu này sẽ được xác thực và lưu tạm vào session, sau đó hệ thống chuyển người dùng đến Bước 2.

Logistic Nhóm 03 Sản phẩm Theo dõi đơn hàng Công cụ quản kho

### Tạo phiếu xuất kho

Thông tin khách hàng, tài xế và hạn giao hàng

Tên khách hàng Số điện thoại khách hàng Địa chỉ giao hàng

Ngày lấy hàng  
mm/dd/yyyy --:-- --

Tiếp tục

Hình 24 - Giao diện bước 1 của tạo phiếu xuất kho

Logistic Nhóm 03 Sản phẩm Theo dõi đơn hàng Công cụ quản kho

### Chọn sản phẩm xuất kho

Thiết lập số lượng xuất cho từng sản phẩm

Tài xế khả dụng  
Lê Tuấn Khang

Tạo đơn hàng ngay

Số lượng	Tên sản phẩm
<input type="text"/>	Cà phê Trung Nguyên
<input type="text"/>	Nước ngọt C4 (440ml)
<input type="text"/>	ChocolatePie
<input type="text"/>	Nước ngọt Sipep (330ml)
<input type="text"/>	Mì SuperKay
<input type="text"/>	Sữa tươi SambaMilk (220ml)

Hình 25 - Giao diện bước 2 của tạo phiếu xuất kho

Tại Bước 2, "Chọn sản phẩm xuất kho", nhân viên sẽ hoàn tất đơn hàng. Giao diện hiển thị danh sách tất cả sản phẩm trong kho, cho phép nhân viên nhập "Số lượng" cần xuất cho từng mặt hàng. Đồng thời, hệ thống cũng hiển thị danh sách "Tài xế khả dụng" để phân công. Khi nhấn "Tạo đơn hàng ngay", hệ thống sẽ kiểm tra tồn kho, tự động trừ số lượng sản phẩm trong Firestore, và tạo một tài liệu đơn hàng mới trong bộ sưu tập exports.

### 3.2.7. Giao diện chi tiết đơn hàng xuất kho

Giao diện "Chi tiết đơn hàng" cung cấp một cái nhìn tổng quan, chỉ đọc về một phiếu xuất kho cụ thể đã được tạo. Trang này hiển thị tất cả thông tin cố định của đơn hàng, bao gồm thông tin khách hàng (tên, SĐT, địa chỉ), tài xế được phân công (dưới dạng ID), trạng thái hiện tại (ví dụ: 'delivering'), và thông tin về người tạo phiếu.

Logistic Nhóm 03 Sản phẩm Theo dõi đơn hàng Công cụ quản kho

#### Chi tiết đơn hàng

Thông tin khách hàng và vận chuyển

Tên khách hàng	Số điện thoại khách hàng	Địa chỉ giao hàng
NGÔ THÀNH DANH	0919308384	Xã Châu Đức, Thành Phố Hồ Chí Minh
Tài xế	Hạn giao hàng	Trạng thái
kjeuhdhM2XRCDooPHx9yIQhWjYY2		delivering
Được tạo bởi	Ngày tạo phiếu	
Demo1	10:25 Ngày 10 tháng 11 năm 2025	
Sản phẩm		
('Cà phê Trung Nguyên': '50')		

Xem tình trạng nhận hàng

Hình 26 - Giao diện chi tiết đơn hàng xuất kho

Logistic Nhóm 03 Sản phẩm Theo dõi đơn hàng Công cụ quản kho

#### Danh sách xuất kho

Theo dõi đơn xuất, hạn giao và trạng thái

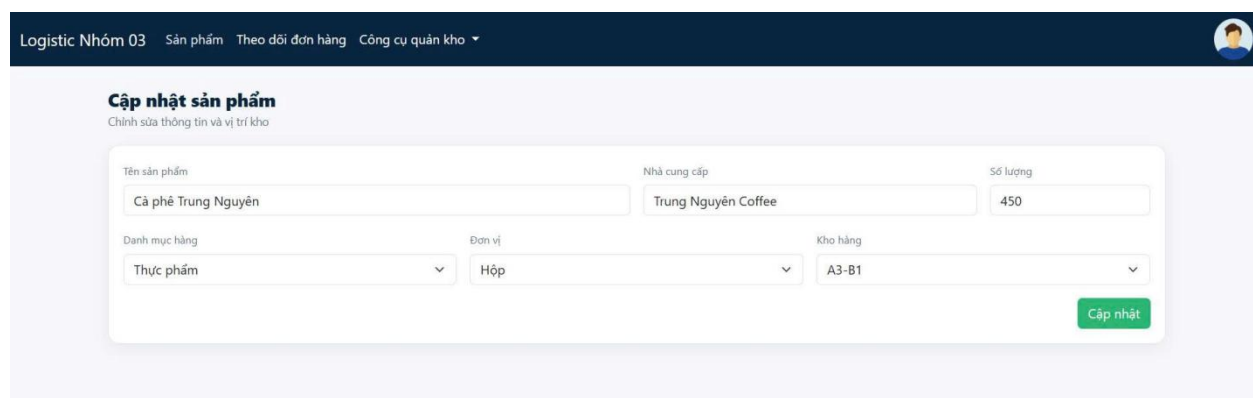
Mã đơn hàng	Tài xế	Tên khách hàng	Trạng thái	Quá hạn	Hạn giao
9Q2W57HfWmYDuAwrx2R	kjeuhdhM2XRCDooPHx9yIQhWjYY2	NGÔ THÀNH DANH	delivering	Còn hạn lấy	Chi tiết
nmpNwozeURj792A64gX	kjeuhdhM2XRCDooPHx9yIQhWjYY2	Test 2	pending	Còn hạn lấy	Chi tiết

Hình 27 - Giao diện danh sách tất cả phiếu xuất kho

Một điểm đáng chú ý là trường "Sản phẩm". Khi tải trang, backend sẽ xử lý dữ liệu từ đơn hàng, tự động truy vấn ID sản phẩm trong collection products để lấy tên sản phẩm tương ứng, sau đó hiển thị chi tiết số lượng xuất kho cho từng mặt hàng.

### 3.2.8. Giao diện cập nhật sản phẩm

Giao diện "Cập nhật sản phẩm" cho phép người dùng chỉnh sửa thông tin của một mặt hàng đã có trong kho. Khi người dùng truy cập, hệ thống sẽ lấy ID của sản phẩm, truy vấn dữ liệu hiện tại từ Firestore và tự động điền thông tin đó vào các trường như "Tên sản phẩm", "Nhà cung cấp", "Số lượng", "Danh mục hàng", "Đơn vị", và "Kho hàng".

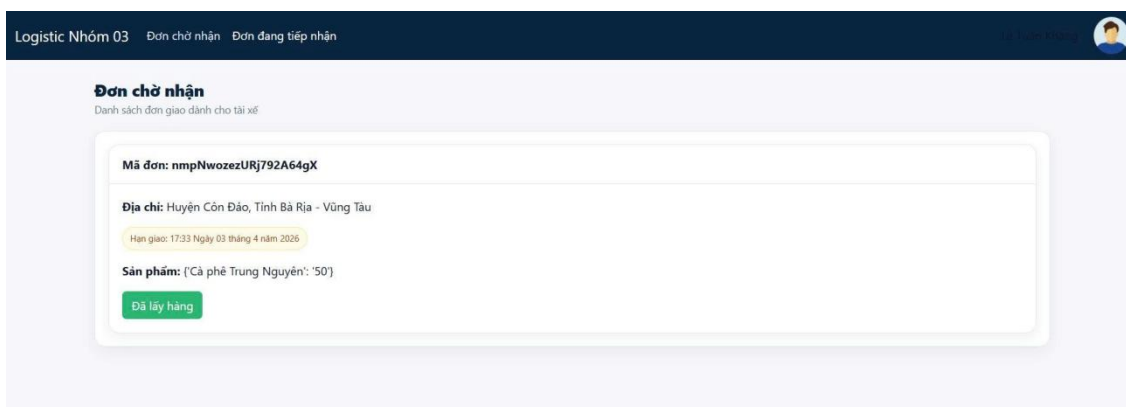


**Hình 28 - Giao diện cập nhật sản phẩm**

Sau khi người dùng chỉnh sửa và nhấn nút "Cập nhật", dữ liệu biểu mẫu sẽ được gửi đi. Phía backend sẽ xác thực dữ liệu này bằng ProductForm, tự động đính kèm tên người dùng đã thực hiện thay đổi (update\_by), và sau đó gửi lệnh update() đến tài liệu Firestore tương ứng để lưu các thay đổi.

### 3.2.9. Giao diện nhận đơn hàng của tài xế

Đây là giao diện "Đơn chờ nhận", trang chủ dành riêng cho tài xế. Hệ thống tự động lọc và chỉ hiển thị các đơn hàng (có trạng thái 'pending') đã được phân công cho tài xế đang đăng nhập. Mỗi thẻ đơn hàng hiển thị thông tin quan trọng như "Mã đơn", "Địa chỉ", "Hạn giao" và "Sản phẩm".



**Hình 29 - Giao diện nhận đơn hàng của tài xế**

Hành động chính tại đây là nút "Đã lấy hàng". Khi tài xế nhấn nút này, hệ thống sẽ cập nhật trạng thái đơn hàng sang 'delivering' và tự động tạo một bản ghi mới trong bộ sưu tập delivery-tracking, chính thức bắt đầu phiên theo dõi vị trí GPS thời gian thực.

### 3.2.10. Giao diện theo dõi danh sách các đơn hàng đã được nhận vận chuyển

Giao diện "Theo dõi vị trí đơn hàng" cung cấp cho Admin và Staff một cái nhìn tổng quan về tất cả các đơn hàng đang trong quá trình vận chuyển. Bảng điều khiển này đọc dữ liệu trực tiếp từ bộ sưu tập delivery-tracking trên Firestore. Nó hiển thị các thông tin quan trọng bao gồm "ID đơn hàng", "Người vận chuyển" và thời gian "Cập nhật cuối".



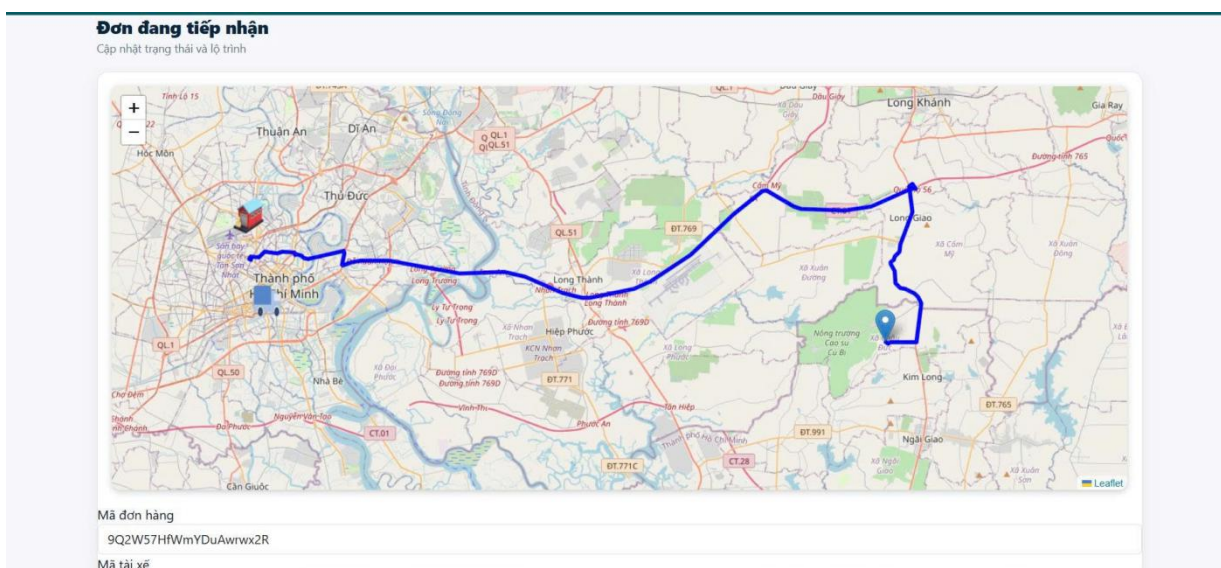
**Hình 30 - Giao diện danh sách đơn hàng đã được nhận vận chuyển**

Tính năng kỹ thuật nổi bật của trang này là cột "Vị trí xe hiện tại". Thay vì chỉ hiển thị tọa độ GPS, backend sẽ lấy vĩ độ và kinh độ (current\_location) từ Firestore và gửi một yêu cầu đến API Nominatim (OpenStreetMap) để thực hiện tra

cứu địa chỉ ngược (reverse geocoding). Kết quả là một địa chỉ văn bản dễ đọc (ví dụ: "Đường số 1, Xã Bình Hưng...") được hiển thị cho người quản lý, và nút "Chi tiết" sẽ dẫn đến bản đồ theo dõi trực quan cho đơn hàng đó.

### 3.2.11. Giao diện theo dõi vị trí đơn hàng

Giao diện "Đơn đang tiếp nhận" là màn hình làm việc chính của tài xế, tích hợp bản đồ Leaflet.js. Bản đồ này hiển thị trực quan ba vị trí then chốt: điểm xuất phát (kho hàng), điểm giao hàng (đích đến) và vị trí hiện tại của tài xế (xe tải).



**Hình 31 - Giao diện theo dõi vị trí đơn hàng**

Công nghệ GPS thời gian thực được kích hoạt bằng `navigator.geolocation.watchPosition` trên trình duyệt của tài xế. Các tọa độ này được liên tục gửi qua fetch (POST) đến backend, sau đó cập nhật vào trường `current_location` (GeoPoint) trên Firestore để người quản lý có thể giám sát trực tiếp.

## KẾT LUẬN

Đề tài “Xây dựng hệ thống quản lý kho và vận chuyển Logistics sử dụng Django và Firebase” đã hoàn thành mục tiêu đề ra, cung cấp một website quản lý tập trung, giúp số hóa quy trình logistics cho doanh nghiệp. Hệ thống đã triển khai thành công các nghiệp vụ cốt lõi cho ba vai trò (Admin, Staff, Deliver), bao gồm quản lý sản phẩm (CRUD), xử lý xuất/nhập kho, và phân công đơn hàng. Về mặt công nghệ, đề tài đã áp dụng thành công Python với framework Django để xử lý logic backend, đồng thời tích hợp sâu Google Firebase, sử dụng Authentication để xác thực, phân quyền và Firestore làm cơ sở dữ liệu NoSQL. Điểm nổi bật nhất là việc hiện thực hóa thành công tính năng theo dõi lộ trình GPS theo thời gian thực, thông qua việc kết hợp thư viện Leaflet.js, API OSRM, và kiểu dữ liệu GeoPoint của Firestore.

Bên cạnh các kết quả đạt được, trong phạm vi đồ án, hệ thống vẫn còn nhiều tiềm năng để phát triển trong tương lai. Hướng phát triển trước mắt là hoàn thiện hệ thống thông báo cho tài xế và xây dựng một module báo cáo, thống kê chuyên sâu dựa trên dữ liệu lịch sử đã được lưu trữ. Về lâu dài, hệ thống có thể được nâng cấp để tích hợp thuật toán tối ưu hóa lộ trình (thay vì chỉ vẽ đường đi), cho phép tài xế nhận và giao nhiều đơn hàng một cách hiệu quả nhất. Cuối cùng, việc phát triển một ứng dụng di động (native app) chuyên biệt cho tài xế sẽ giúp tăng cường tính ổn định của việc cập nhật GPS và mở rộng thêm nhiều tính năng nghiệp vụ như quét mã vạch hay xác nhận chữ ký điện tử.

## DANH MỤC TÀI LIỆU THAM KHẢO

- [1] Django Software Foundation, “Django documentation,” The Django Project, 2025. <https://www.djangoproject.com/> Truy cập: 15/09/2025.
- [2] Google Firebase, “Cloud Firestore — documentation,” Firebase, 2025. <https://firebase.google.com/> Truy cập: 22/09/2025.
- [3] LeafletJS Team, “Leaflet — the leading open-source JavaScript library for interactive maps,” LeafletJS, 2025. <https://leafletjs.com/> Truy cập: 05/10/2025.
- [4] OpenStreetMap Foundation, “Nominatim — search engine for OpenStreetMap data,” Nominatim, 2025. <https://nominatim.org/> Truy cập: 12/10/2025.
- [5] Project OSRM, “Open Source Routing Machine (OSRM),” Project OSRM, 2025. <https://project-osrm.org/> Truy cập: 18/10/2025.
- [6] Bootstrap Team, “Bootstrap — the world’s most popular front-end framework,” Bootstrap, 2025. <https://getbootstrap.com/> Truy cập: 25/10/2025.
- [7] Python Software Foundation, “Python — official website,” Python.org, 2025. <https://www.python.org/> Truy cập: 05/11/2025.
- [8] GitHub, “GitHub — code hosting platform for version control and collaboration,” GitHub, 2025.: <https://github.com/> Truy cập: 08/11/2025.
- [9] Stack Overflow, “Stack Overflow — community for developers,” Stack Overflow, 2025. <https://stackoverflow.com/> Truy cập: 09/11/2025.
- [10] W3Schools, “HTML, CSS, JavaScript, and Python tutorials,” W3Schools, 2025. <https://www.w3schools.com/> Truy cập: 10/11/2025.