

**BỘ GIAO THÔNG VẬN TẢI**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC HÀNG HẢI VIỆT NAM**



**BÁO CÁO THỰC TẬP TỐT NGHIỆP**

**XÂY DỰNG HỆ THỐNG KIỂM SOÁT TỰ ĐỘNG PHƯƠNG TIỆN  
RA/VÀO TRONG BÃI XE THÔNG MINH**

**NGÀNH: CÔNG NGHỆ THÔNG TIN**

**GVHD:**

**ThS. Phạm Trung Minh**

**Sinh viên thực hiện:**

**Trần Tuấn Phong – 89940**

**Đinh Thị Mỹ – 85718**

**Nguyễn Trịnh Tấn Phát – 86940**

Hải Phòng, tháng 4 năm 2024

**BỘ GIAO THÔNG VẬN TẢI**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC HÀNG HẢI VIỆT NAM**



**BÁO CÁO THỰC TẬP TỐT NGHIỆP**

**XÂY DỰNG HỆ THỐNG KIỂM SOÁT TỰ ĐỘNG PHƯƠNG TIỆN  
RA/VÀO TRONG BÃI XE THÔNG MINH**

**NGÀNH: CÔNG NGHỆ THÔNG TIN**

**GVHD:**

**ThS. Phạm Trung Minh**

**Sinh viên thực hiện:**

**Trần Tuấn Phong – 89940**

**Đinh Thị Mây – 85718**

**Nguyễn Trịnh Tấn Phát – 86940**

Hải Phòng, tháng 4 năm 2024

## LỜI MỞ ĐẦU

Trong thời đại công nghệ 4.0, việc ứng dụng trí tuệ nhân tạo (AI) và Internet of Things (IoT) đã đóng góp cho sự phát triển của nhiều lĩnh vực, đặc biệt là trong việc quản lý và giám sát giao thông. Trong bối cảnh xã hội đang hướng tới tự động hóa, việc xây dựng hệ thống giám sát tự động các phương tiện ra vào trong bãi xe thông minh không chỉ giúp tối ưu hóa quy trình quản lý bãi đỗ xe mà còn đảm bảo tính an toàn và bảo mật cho người dùng.

Bài báo cáo này tập trung vào việc phát triển một hệ thống hoàn chỉnh gồm ba thành phần chính: hệ thống nhúng kết hợp IoT sử dụng bo mạch Arduino, mô hình trí tuệ nhân tạo (AI) để đọc biển số phương tiện, và một máy chủ (server) được triển khai trên nền tảng Nodejs. Bằng cách kết hợp sức mạnh của các công nghệ này, nhóm chúng em đặt mục tiêu tạo ra một hệ thống thông minh, linh hoạt và hiệu quả trong việc giám sát và quản lý luồng phương tiện ra vào trong bãi xe.

Nội dung báo cáo thực tập gồm 4 chương:

- **CHƯƠNG 1: GIỚI THIỆU BÀI TOÁN**
- **CHƯƠNG 2: CƠ SỞ LÝ THUYẾT**
- **CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ HỆ THỐNG**
- **CHƯƠNG 4: CÁC KẾT QUẢ ĐẠT ĐƯỢC**

## MỘT SỐ TỪ VIẾT TẮT

<b>Từ viết tắt</b>	<b>Ý nghĩa</b>
ACK	Acknowledged
AI	Artificial Intelligence
API	Application Programming Interface
CDN	Content Delivery Network
CoAP	Constrained Application Protocol
CSDL	Cơ sở dữ liệu
CSS	Cascading Style Sheets
EJS	Embedded JavaScript
HTTP	HyperText Transfer Protocol
I2C	Inter-Integrate Circuit
IOT	Internet Of Things
JSON	JavaScript Object Notation
MQTT	Message Queuing Telemetry Transport
NACK	Not Acknowledged
NPM	Node Package Manager
SCL	Serial Clock
SDA	Serial Data
SPI	Serial Peripheral Interface
SVM	Server – Model – Controller
UART	Universal Asynchronous Receiver / Transmitter
UID	User ID
YOLO	You Only Look Once

## DANH MỤC CÁC HÌNH VẼ

Hình 1: Sơ đồ hệ thống (System model) .....	16
Hình 2: Tổng quan mô hình AI .....	18
Hình 3: Sơ đồ xử lý frame .....	19
Hình 4: Mô tả YOLO .....	20
Hình 5: Kiến trúc mạng YOLO .....	21
Hình 6: Ảnh huấn luyện YOLO V8 .....	22
Hình 7: Gán nhãn biên số .....	23
Hình 8: Thông tin của đối tượng trong ảnh .....	23
Hình 9: Kết quả huấn luyện YOLO V8 .....	23
Hình 10: Kết quả huấn luyện YOLO V8 .....	24
Hình 11: Ảnh gốc chưa xử lý .....	25
Hình 12: Kết quả của bước xử lý ảnh .....	25
Hình 13: Thư mục cha chứa ảnh huấn luyện CNN .....	26
Hình 14: Ảnh huấn luyện CNN .....	26
Hình 15: Kiến trúc model CNN .....	27
Hình 16: Kết quả huấn luyện model CNN .....	27
Hình 17: Sơ đồ chân của Arduino Uno R3 .....	29
Hình 18: Sơ đồ chân của ESP8266 NodeMCU .....	30
Hình 19: Sơ đồ chân của module RFID .....	31
Hình 20: Cảm biến vật cản hồng ngoại .....	32
Hình 21: Đèn led giao thông .....	33
Hình 22: Button .....	34
Hình 23: Servo .....	35
Hình 24: Sơ đồ hệ thống phần cứng .....	36
Hình 25: Sơ đồ luồng dữ liệu mô hình .....	37

Hình 26: Sơ đồ hoạt động ở chế độ thủ công.....	37
Hình 27: Sơ đồ hoạt động ở chế độ tự động .....	38
Hình 28: Sơ đồ hoạt động của WebSocket .....	40
Hình 29: Sơ đồ kết nối I2C .....	42
Hình 30: Cấu trúc thư mục của Express.....	47
Hình 31: Các tính năng của Express .....	47
Hình 32: Sơ đồ Use – Case .....	57
Hình 33: Sơ đồ lớp .....	58
Hình 34:Sơ đồ phân rã chức năng .....	59
Hình 35: Sơ đồ mức ngữ cảnh.....	60
Hình 36: Sơ đồ mức đỉnh .....	61
Hình 37: Sơ đồ mức dưới đỉnh - Quản lý hệ thống.....	63
Hình 38: sơ đồ mức dưới đỉnh - Quản lý danh mục .....	64
Hình 39: Sơ đồ mức dưới đỉnh - Quản lý lịch sử ra/vào.....	65
Hình 40: Sơ đồ mức dưới đỉnh - Báo cáo thống kê .....	66
Hình 41: Sơ đồ liên kết thực thể .....	68
Hình 42: Cơ sở dữ liệu .....	68
Hình 43: Kết quả nhận dạng biển số xe trong video.....	69
Hình 44: Kết quả demo hệ thống nhúng .....	70
Hình 45: Demo giao tiếp giữa Arduino-Esp-Server .....	70
Hình 46: Demo giao diện trang chủ .....	71
Hình 47: Demo giao diện thêm khách hàng.....	71
Hình 48: Demo giao diện đăng ký .....	72
Hình 49: Demo giao diện đăng nhập.....	72

## **DANH MỤC CÁC BẢNG**

Bảng 1: Thông số cơ bản của Arduino Uno R3 .....	29
Bảng 2: Thực thể và thuộc tính .....	67
Bảng 3: Mối liên kết giữa các thực thể .....	67

## MỤC LỤC

LỜI MỞ ĐẦU .....	3
MỘT SỐ TỪ VIẾT TẮT .....	4
DANH MỤC CÁC HÌNH VẼ.....	5
DANH MỤC CÁC BẢNG.....	7
MỤC LỤC .....	8
Chương 1: giới thiệu bài toán.....	11
1.1.Mục đích, yêu cầu .....	11
1.2.Mô tả hoạt động của bài toán .....	11
1.2.1. Các đối tượng chính.....	11
1.2.2. Các chức năng chính.....	12
1.2.3. Hoạt động hệ thống.....	13
1.3.Sơ đồ hệ thống (System model).....	16
1.4.Các module chính của hệ thống .....	16
1.4.1. Mô hình AI.....	16
1.4.2. Hệ thống nhúng kết hợp IOT .....	17
1.4.3. Hệ thống Server .....	17
Chương 2: cơ sở lý thuyết .....	18
2.1.Mô hình AI.....	18
2.1.1. Tổng quan .....	18
2.1.2. Phát hiện biên số với YOLO V8.....	20
2.1.3. Xử lý ảnh với OpenCV .....	24



2.1.4. Nhận dạng ký tự với model CNN .....	25
2.2.Hệ thống nhúng kết hợp IOT .....	28
2.2.1. Giới thiệu về các thiết bị .....	28
2.2.2. Sơ đồ hoạt động .....	36
2.2.3. Giao thức WebSocket .....	38
2.2.4. Cách thức truyền dữ liệu theo chuẩn giao tiếp I2C .....	42
2.3.Hệ thống Server.....	43
2.3.1. Front End.....	43
2.3.2. Back End.....	46
Chương 3: phân tích thiết kế hệ thống .....	50
3.1.Sơ đồ Use – Case .....	50
3.1.1. Các tác nhân và ca sử dụng.....	50
3.1.2. Mô tả kịch bản các ca sử dụng.....	51
3.1.3. Sơ đồ Use – Case .....	57
3.1.4. Sơ đồ lớp .....	58
3.2.CSDL – Liên kết .....	59
3.2.1. Sơ đồ phân rã chức năng.....	59
3.2.2. Sơ đồ mức ngữ cảnh .....	60
3.2.3. Sơ đồ mức đỉnh .....	61
3.2.4. Sơ đồ mức dưới đỉnh.....	63
3.2.5. Sơ đồ liên kết thực thể .....	67
3.2.6. Cơ sở dữ liệu .....	68

Chương 4: các kết quả đạt được.....	69
4.1.Mô hình AI.....	69
4.2.Hệ thống nhúng.....	70
4.3.Hệ thống Server.....	71
KẾT LUẬN.....	73
TÀI LIỆU THAM KHẢO.....	74
NHẬN XÉT TÓM TẮT CỦA GIẢNG VIÊN HƯỚNG DẪN .....	75

# CHƯƠNG 1: GIỚI THIỆU BÀI TOÁN

## 1.1.Mục đích, yêu cầu

- Sử dụng công nghệ IoT, RFID, AI trong việc nhận diện, điều khiển tự động barrier của bãi xe tự động thông minh của một tòa nhà chung cư.
- Xây dựng hệ thống phân cứng sử dụng công nghệ RFID để điều khiển đóng/mở cổng trong bãi xe.
- Áp dụng kiến thức trí tuệ nhân tạo, nghiên cứu công nghệ nhận dạng biển số xe nhằm xác định phương tiện đi qua cổng.
- Xây dựng hệ thống server: cho phép người dùng đăng ký/quản lý thẻ ra vào; người quản trị giám sát, đảm bảo an ninh, thống kê số lượt phương tiện ra vào... tòa nhà.
- Kết nối các phân hệ thành một hệ thống hoàn chỉnh.

## 1.2.Mô tả hoạt động của bài toán

### 1.2.1. Các đối tượng chính

- Người dùng: Là dân cư trong chung cư. Sử dụng thẻ dành cho dân cư.
- Khách: Người sử dụng xe đến và đi. Dùng thẻ ngày (dành cho khách vắng lai).
- Nhân viên an ninh: Người trực tại booth. Thực hiện các công việc như hướng dẫn người dùng, xử lý thẻ không hợp lệ hoặc vấn đề kỹ thuật.
- Nhân viên kỹ thuật: Người thực hiện các công việc phân quyền, quản lý thẻ, quản lý người dùng.
- Quản trị: Người quản trị có quyền truy cập và kiểm soát toàn bộ hệ thống. Có quyền xem báo cáo thống kê, nhật ký hoạt động...

- Hệ thống nhúng và IoT: gồm các module chính như RFID, Barrier (động cơ servo), cảm biến (nhận biết phương tiện đang đi ra hoặc vào).
- Mô hình AI: kết hợp với camera để đọc biển số xe. Mô hình gồm 2 camera để chụp ảnh toàn cảnh và chụp ảnh biển số.
- Server và cơ sở dữ liệu: Server hệ thống xử lý dữ liệu từ các thiết bị, tính toán doanh số. Cơ sở dữ liệu lưu trữ các thông tin về người dùng, nhân viên, thẻ, nhật ký hoạt động.

### ***1.2.2. Các chức năng chính***

#### ***a) Quản lý thẻ ra vào***

Có 2 loại thẻ: thẻ dành cho dân cư (người dùng đăng ký trước trên hệ thống, có thẻ thanh toán theo tháng/năm) và thẻ dành cho khách (không cần đăng ký, được coi như là thẻ ngày).

Đối với thẻ dành cho dân cư, chủ hộ đăng ký thẻ cho 1 hộ gia đình (số lượng thẻ). Các thẻ liên kết với số căn hộ và gắn với phương tiện (loại phương tiện, số xe). Người dùng có thể sử dụng tài khoản đã đăng ký để quản lý thẻ ra vào của mình thông qua trang web (xem lịch sử ra vào, xem số tiền thanh toán, hạn sử dụng...).

Đối với khách vắng lai, có thể sử dụng thẻ trắng (được đặt tại nơi quét thẻ) để ra vào bãi xe. Khách sẽ thực hiện thanh toán tiền gửi xe theo ngày.

Thông tin về tất cả thẻ của người dùng được lưu trữ trong cơ sở dữ liệu để nhân viên kỹ thuật có thể quản lý.

#### ***b) Nhận diện biển số***

Mô hình AI tích hợp với camera để nhận diện biển số xe. Thông tin nhận diện được gửi đến server để xác thực và ghi vào nhật ký hoạt động.

### ***c) Cổng barrier tự động và thủ công***

Cổng barrier sử dụng động cơ servo, việc tự động đóng/mở được điều khiển bởi server. Nhân viên có thể thực hiện mở cổng bằng cách thủ công thông qua nút nhấn vật lý bên ngoài.

### ***d) Thống kê và báo cáo***

Hệ thống tự động thống kê và báo cáo về số lượt ra vào, doanh thu, số lượng người đăng ký, số lượng xe đăng ký và các thông tin khác liên quan. Người quản trị có thể xem và tải xuống các báo cáo từ trang web của quản trị.

## ***1.2.3. Hoạt động hệ thống***

### ***a) Đăng ký dịch vụ***

Người dùng đến văn phòng quản lý để thực hiện đăng ký dịch vụ. Tại đây, người dùng khai báo các thông tin để nhân viên tạo tài khoản cho người dùng. Các thông tin cần thiết bao gồm thông tin cá nhân, số căn hộ, loại phương tiện, số lượng phương tiện, biển số xe...

Người dùng sau khi đã đăng ký thông tin sử dụng dịch vụ sẽ được phát một thẻ dân cư, lưu các thông tin đã đăng ký.

### ***b) Thanh toán dịch vụ***

Sau khi đăng ký dịch vụ, người dùng thực hiện việc thanh toán phí dịch vụ định kỳ hàng tháng, cùng với các khoản phí khác của chung cư.

Khi thẻ sắp đến hạn, hệ thống sẽ thông báo cho người dùng, sau 1 khoảng thời gian không gia hạn thì vô hiệu hệ thống đối với thẻ đó.

### ***c) Ra/vào bãi xe***

Khu vực quét thẻ được chia thành hai làn: một làn dành cho oto và một làn dành cho các phương tiện 2 bánh khác.

Người dùng sử dụng thẻ vật lý được cấp để ra/vào bãi xe. Khi người dùng quét thẻ, module RFID đọc UID của thẻ, camera chụp lại 2 hình ảnh: ảnh toàn cảnh và ảnh biển số để đọc biển số xe. Sau cùng, server sẽ nhận được dữ liệu từ hệ thống nhúng (UID của thẻ) và mô hình AI (biển số xe).

Đối với chiều vào, khi người dùng quét thẻ, biển số xe và UID của thẻ được gán trong cơ sở dữ liệu, trạng thái của xe sẽ chuyển thành “Đang trong bãi”, server gửi lệnh mở cổng để thực hiện mở barrier.

Đối với chiều ra, khi người dùng quét thẻ, server sẽ thực hiện so sánh UID của thẻ và biển số xe nhận được với dữ liệu trong cơ sở dữ liệu. Sau đó kiểm tra trạng thái của xe là “Đang trong bãi” hay “Ngoài bãi”. Khi mọi điều kiện được đáp ứng, thì thẻ hợp lệ, server gửi lệnh mở cổng để thực hiện mở barrier, ngược lại thì thông báo thẻ không hợp lệ, server sẽ thông báo điều kiện chưa được đáp ứng để nhân viên trực booth xử lý.

Đối với loại thẻ dành cho dân cư, cần phải kiểm tra thêm các thông tin: trạng thái đăng ký dịch vụ còn hay đã hủy, nếu còn sử dụng dịch vụ thì trạng thái thanh toán là đã thanh toán rồi hay chưa,... Ngoài ra, mỗi khi quét thẻ, hệ thống còn phải kiểm tra xem thẻ đó có phải là thẻ của chung cư hay không.

Phía sau mỗi barrier đều có 1 cảm biến để đảm bảo rằng khi phương tiện đã đi qua cổng, hệ thống sẽ thực hiện đóng cổng.

#### ***d) Dữ liệu hoạt động hệ thống***

Khi quét thẻ xe hợp lệ, trạng thái hoạt động được ghi là “Thẻ hợp lệ”, ngược lại, trạng thái hoạt động được ghi là “Thẻ không hợp lệ”. Khi Barrier được mở bằng cách thủ công, trạng thái hoạt động được ghi là “Thủ công”.

Các dữ liệu được ghi vào CSDL bao gồm UID của thẻ tại thời điểm quét, thời gian quét thẻ, biển số xe đọc được, ảnh biển số xe và ảnh toàn cảnh.

#### ***e) Quản lý thẻ và người dùng***

Người dùng có thể sử dụng tài khoản đã tạo để đăng nhập hệ thống, quản lý thông tin cá nhân của mình, xem lịch sử ra/vào, trạng thái thanh toán, yêu cầu hủy thẻ, yêu cầu tạo lại thẻ, thêm thẻ, cập nhật thông tin...

#### ***f) Báo cáo thống kê***

Nhân viên thực hiện báo cáo thống kê mỗi tháng bằng cách đăng nhập vào hệ thống với tài khoản của nhân viên.

Nhân viên nhấn chọn báo cáo doanh thu.

Hệ thống hiển thị các thông tin:

- Số lượng người đăng ký thẻ trong tháng.
- Loại phương tiện được đăng kí trong tháng.
- Số tiền thu được dựa trên từng loại phương tiện.

Hệ thống lưu bản báo cáo vào cơ sở dữ liệu hoặc xuất thành văn bản tùy theo yêu cầu của người sử dụng.

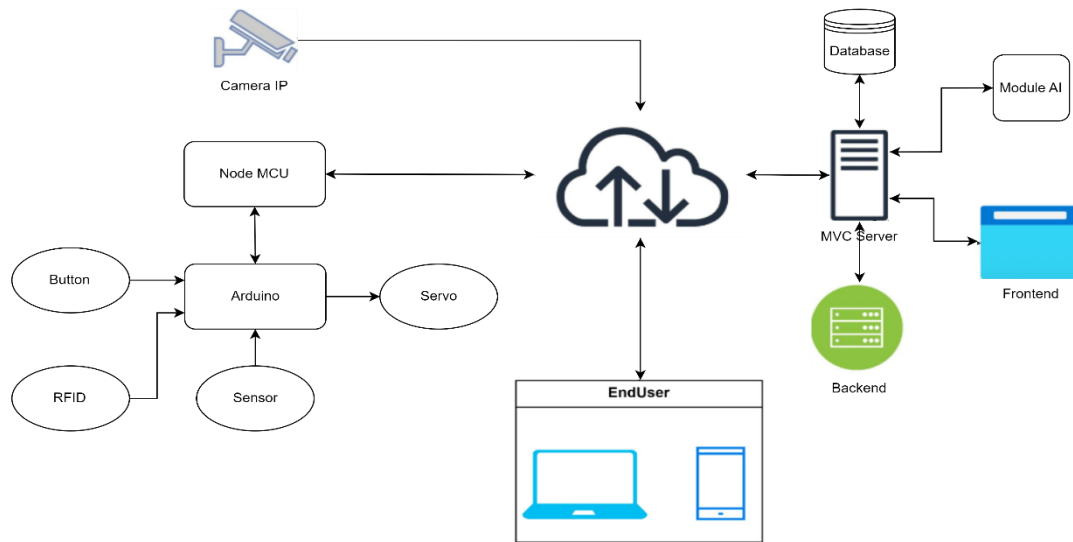
#### ***g) Website***

Giao diện người dùng: Đăng nhập, quản lý thẻ xe, lịch sử ra/ vào, trạng thái thẻ (đã thanh toán/ chưa thanh toán).

Giao diện nhân viên kỹ thuật: Có các chức năng như phân quyền, quản lý thẻ, quản lý người dùng, thống kê và báo cáo.

Giao diện điều khiển bãi xe: Hiển thị thời gian, ảnh chụp biển số xe và ảnh toàn cảnh, biển số xe đọc được, cho phép xem lịch sử phương tiện ra vào.

### 1.3.Sơ đồ hệ thống (System model)



Hình 1: Sơ đồ hệ thống (System model)

### 1.4.Các module chính của hệ thống

Hệ thống bao gồm 3 module chính:

- Mô hình AI: Hệ thống áp dụng mô hình, thuật toán trí tuệ nhân tạo để phát hiện vị trí của biển số xe, từ đó nhận dạng các ký tự trên biển số để gửi dữ liệu cho máy chủ.
- Hệ thống nhúng kết hợp IOT: Hệ thống vật lý mô phỏng cơ chế hoạt động của cổng barrier tại các bãi gửi xe.
- Hệ thống phần mềm máy chủ (Server): Hệ thống đảm nhiệm việc xử lý các luồng tín hiệu (tín hiệu phản hồi của hệ thống phần cứng thông báo tới giao diện người dùng), xử lý dữ liệu (Lưu lại lịch sử hoạt động vào database).

#### 1.4.1. Mô hình AI

Công cụ triển khai

- Thư viện chính: OpenCV, ultralytics, keras.



- Phần mềm: Visual Studio Code.
- Ngôn ngữ: Python.

Mục tiêu của mô hình này là phát triển một hệ thống nhận diện được các ký tự trên biển số xe trong video với độ chính xác cao nhất để gửi đến server xử lý. Đầu vào của mô hình là một video có chứa hình ảnh của phương tiện, đầu ra là một chuỗi ký tự đọc được từ biển số xe của phương tiện đó.

#### ***1.4.2. Hệ thống nhúng kết hợp IOT***

- Các mạch vi điều khiển: Arduino Uno R3, NodeMCU ESP8266.
- Động cơ Servo.
- Mạch RFID RC522 và thẻ từ.
- Cảm biến: Cảm Biến hồng ngoại LM393.
- Các linh kiện phụ trợ khác: Nút nhấn, đèn giao thông, điện trở, dây dẫn,...

Mục tiêu của hệ thống là có thể trao đổi thông tin giữa mô hình và máy chủ Web thông qua việc sử dụng giao thức WebSocket. Mô hình có thể gửi dữ liệu về UID của thẻ được quét tới máy chủ và nhận tín hiệu đóng/mở servo từ máy chủ.

#### ***1.4.3. Hệ thống Server***

Bao gồm: Frontend, Backend, Database.

Xây dựng theo mô hình MVC.

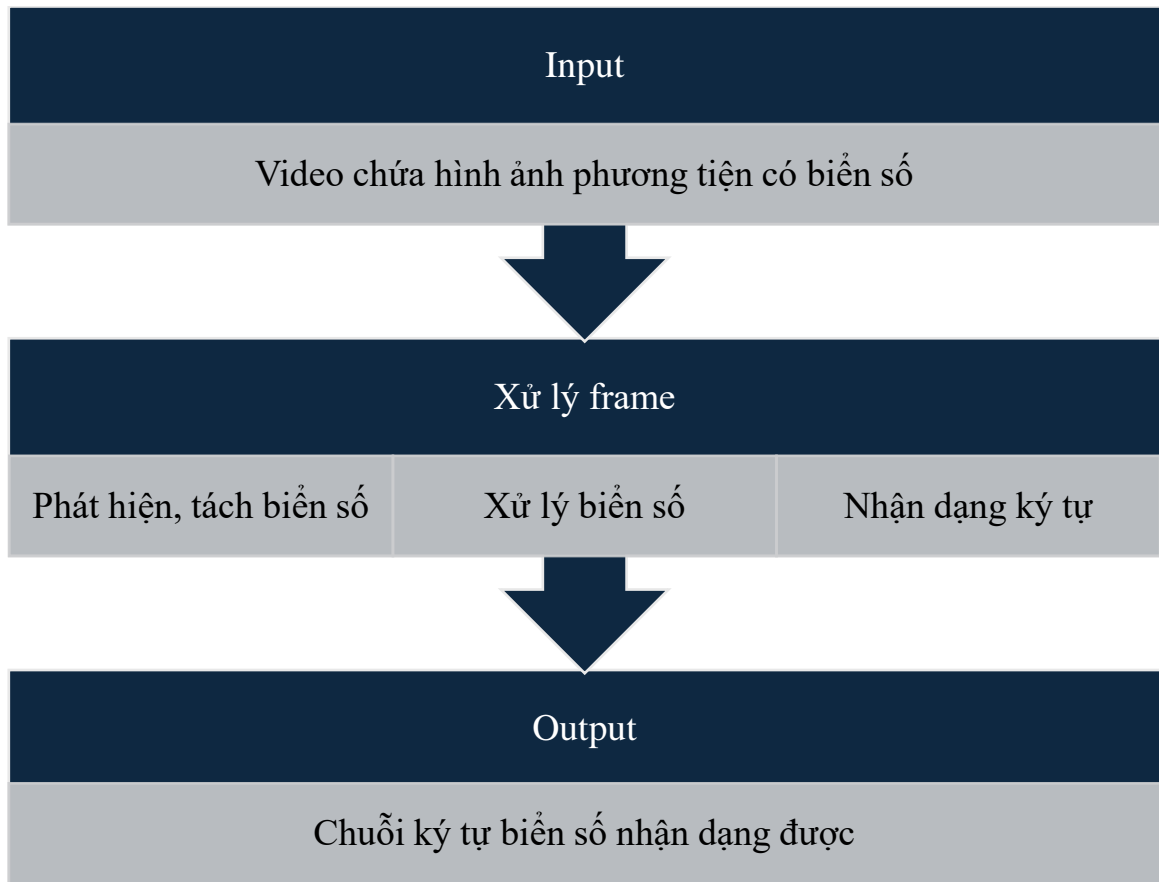
Mục tiêu của hệ thống là có thể tương tác với người dùng thông qua giao diện frontend, có thể truy vấn dữ liệu tùy theo yêu cầu của người dùng, giao tiếp với phần cứng thông qua giao thức Websocket, kết nối với mô hình AI để thu được dữ liệu về biển số xe.

## CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

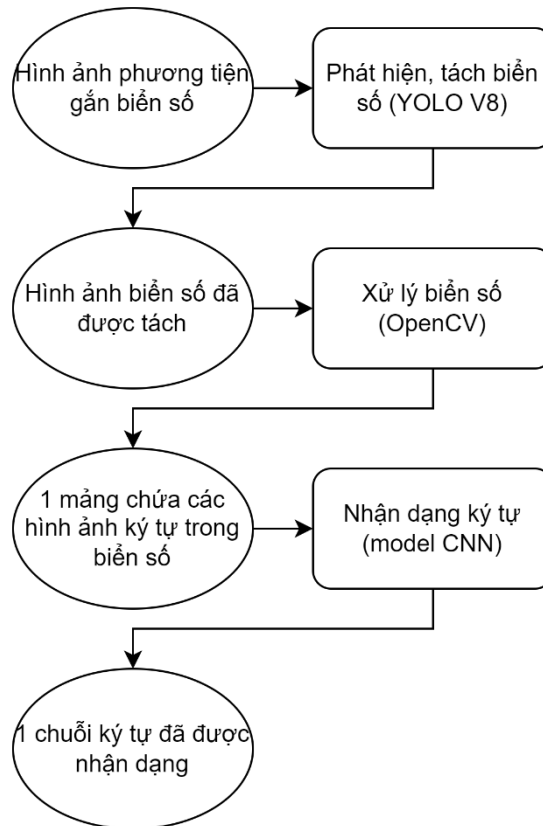
### 2.1.Mô hình AI

#### 2.1.1. Tổng quan

##### a) Mô hình AI



Hình 2: Tổng quan mô hình AI



Hình 3: Sơ đồ xử lý frame

### ***b) Biển số xe Việt Nam***

Biển số xe hay còn gọi là biển kiểm soát xe cơ giới được gắn ở đuôi xe đối với mô tô hay cả đầu xe và đuôi xe đối với xe ô tô. Biển số xe ô tô, xe máy ở mỗi địa phương sẽ được quy định với những ký hiệu khác nhau.

Biển số xe có 2 hình dạng chính là hình chữ nhật (đối với biển có 1 dòng ký tự) và hình vuông (đối với biển có 2 dòng ký tự). Đây là đặc điểm quan trọng, có thể làm tăng thêm sự phức tạp của thuật toán nếu muốn nhận diện được cả 2 loại biển số.

Màu sắc của biển số xe Việt Nam được chia thành các màu xanh, đỏ, vàng, trắng, phân chia theo từng nhóm cơ quan và cá nhân. Đặc điểm này không ảnh hưởng tới thuật toán của mô hình AI.

Các chữ cái sử dụng trên biển số xe Việt Nam: A, B, C, D, E, F, G, H, K, L, M, N, P, S, T, U, V, X, Y, Z. Cùng với các chữ số từ 0 đến 9, đây sẽ là số lượng đầu vào dùng để huấn luyện mô hình nhận dạng ký tự.

### 2.1.2. Phát hiện biển số với YOLO V8

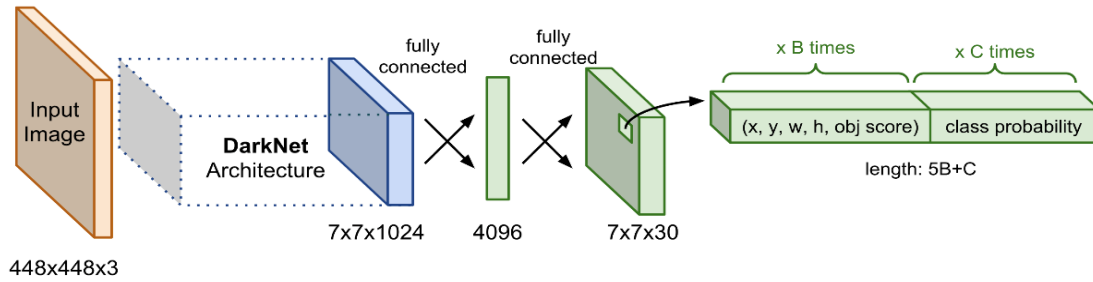
#### a) Giới thiệu YOLO

Mô hình YOLO (You Only Look Once) được phát triển bởi Joseph Redmon và Santosh Divvala. Joseph Redmon là hai nhà nghiên cứu trong lĩnh vực computer vision và deep learning. Bài báo khoa học về YOLO được công bố vào ngày 8/8/2016, sau đó mô hình này đã trải qua nhiều phiên bản cải tiến như YOLOv2, YOLOv3,...YOLOv8. Mục tiêu của YOLO không chỉ là dự báo nhãn cho đối tượng mà còn có thể xác định vị trí của đối tượng trên bức ảnh. Do đó, YOLO có thể phát hiện được nhiều đối tượng có nhãn khác nhau trên cùng một bức ảnh. Đây không phải thuật toán tốt nhất, nhưng là thuật toán nhanh nhất (đạt tốc độ gần như real time).



Hình 4: Mô tả YOLO

## Kiến trúc mạng YOLO:



Hình 5: Kiến trúc mạng YOLO

YOLO là một mô hình mạng CNN, được xây dựng cho việc phát hiện, nhận dạng, phân loại đối tượng. YOLO được tạo ra từ việc kết hợp giữa các lớp chập (convolutional layers) và lớp kết nối (connected layers). Trong đó các lớp chập sẽ trích xuất ra các đặc trưng (feature) của ảnh, còn lớp kết nối sẽ dự đoán ra xác suất đó và tọa độ của đối tượng.

Trong sơ đồ trên, thành phần Darknet Architecture được gọi là base network, bao gồm các các lớp chập (convolutional layers), có tác dụng trích xuất đặc trưng (feature) của ảnh. Output của base network là một feature map có kích thước 7x7x1024 sẽ được sử dụng làm input cho các Extra layers có tác dụng dự đoán nhãn và tọa độ bounding box của vật thể.

Đầu vào của mô hình là một bức ảnh, mô hình sẽ nhận dạng ảnh đó có đối tượng nào hay không, sau đó sẽ xác định tọa độ của đối tượng trong bức ảnh. Kết quả cuối cùng sẽ là một danh sách các bounding box (hộp bao quanh) của đối tượng đã được nhận diện, với thông tin về lớp, độ chính xác và vị trí. Đối với ảnh, kết quả có thể được trực tiếp vẽ lên ảnh đó, trong khi đối với video mô hình sẽ chia ra thành các khung hình và coi nó như một bức ảnh.

Trong bài toán này, em sử dụng mô hình YOLO V8 cho việc phát hiện biển số xe trong ảnh.

### ***b) Tập dữ liệu huấn luyện***

Tập dữ liệu huấn luyện là 1 file gồm 800 ảnh ô tô và 800 ảnh xe máy, tất cả phương tiện đều có gắn biển số.

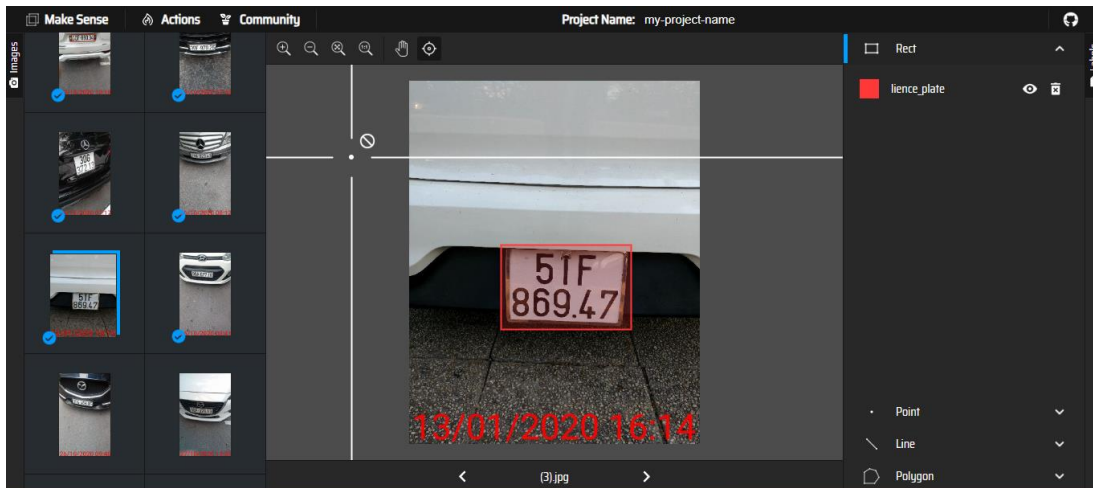


Hình 6: Ảnh huấn luyện YOLO V8

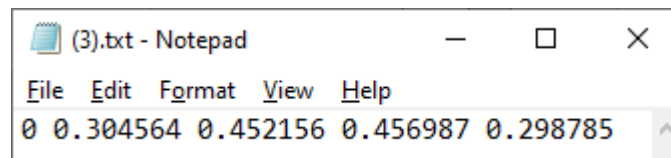
### ***c) Huấn luyện mô hình YOLO V8 với tập dữ liệu tùy chỉnh***

#### **Xử lý dữ liệu thu thập**

- Bước 1: Đưa kích thước toàn bộ ảnh thu thập được về 640 x 640. Vì mô hình Yolo hoạt động tốt nhất trên tập dữ liệu có kích thước này.
- Bước 2: Thực hiện gán nhãn cho đối tượng biển số trong ảnh.
- Bước 3: Sau khi gán nhãn cho toàn bộ ảnh, ta thu được một thư mục “train” gồm ảnh và text tương ứng chứa thông tin của các đối tượng trong ảnh đó (nhãn, tọa độ của đối tượng).



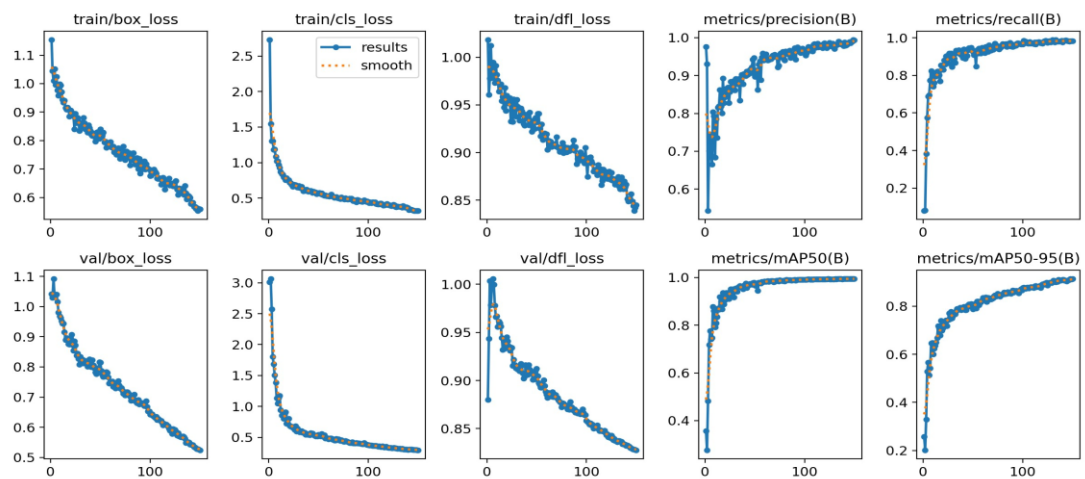
Hình 7: Gán nhãn biển số



Hình 8: Thông tin của đối tượng trong ảnh

### Huấn luyện

Sử dụng file train đã chuẩn bị, thực hiện train mô hình YOLOv8 bằng Google Colab với số epochs = 150. Sau 150 epochs, ta thu được kết quả sau:



Hình 9: Kết quả huấn luyện YOLO V8

Epoch 144/150	GPU_mem 2.41G Class all	box_loss 0.5681 Images 374	cls_loss 0.3351 Instances 1129	dfl_loss 0.8503 Box(P 0.986	Instances 12 R 0.985	Size 640: 100% 24/24 [00:08<00:00, 2.77it/s] mAP50 mAP50-95): 100% 12/12 [00:04<00:00, 2.72it/s] 0.994 0.904
Epoch 145/150	GPU_mem 2.42G Class all	box_loss 0.5682 Images 374	cls_loss 0.3257 Instances 1129	dfl_loss 0.8565 Box(P 0.989	Instances 21 R 0.984	Size 640: 100% 24/24 [00:07<00:00, 3.15it/s] mAP50 mAP50-95): 100% 12/12 [00:05<00:00, 2.10it/s] 0.994 0.906
Epoch 146/150	GPU_mem 2.39G Class all	box_loss 0.5602 Images 374	cls_loss 0.3231 Instances 1129	dfl_loss 0.849 Box(P 0.985	Instances 9 R 0.984	Size 640: 100% 24/24 [00:06<00:00, 3.69it/s] mAP50 mAP50-95): 100% 12/12 [00:06<00:00, 1.80it/s] 0.994 0.908
Epoch 147/150	GPU_mem 2.41G Class all	box_loss 0.5631 Images 374	cls_loss 0.3243 Instances 1129	dfl_loss 0.8441 Box(P 0.99	Instances 19 R 0.982	Size 640: 100% 24/24 [00:06<00:00, 3.72it/s] mAP50 mAP50-95): 100% 12/12 [00:05<00:00, 2.05it/s] 0.994 0.91
Epoch 148/150	GPU_mem 2.4G Class all	box_loss 0.5532 Images 374	cls_loss 0.3217 Instances 1129	dfl_loss 0.8389 Box(P 0.99	Instances 15 R 0.983	Size 640: 100% 24/24 [00:06<00:00, 3.63it/s] mAP50 mAP50-95): 100% 12/12 [00:04<00:00, 2.53it/s] 0.994 0.913
Epoch 149/150	GPU_mem 2.41G Class all	box_loss 0.56 Images 374	cls_loss 0.3221 Instances 1129	dfl_loss 0.8424 Box(P 0.995	Instances 14 R 0.983	Size 640: 100% 24/24 [00:07<00:00, 3.18it/s] mAP50 mAP50-95): 100% 12/12 [00:04<00:00, 2.81it/s] 0.994 0.912
Epoch 150/150	GPU_mem 2.43G Class all	box_loss 0.5589 Images 374	cls_loss 0.3213 Instances 1129	dfl_loss 0.8449 Box(P 0.993	Instances 31 R 0.984	Size 640: 100% 24/24 [00:08<00:00, 2.80it/s] mAP50 mAP50-95): 100% 12/12 [00:04<00:00, 2.69it/s] 0.994 0.913

Hình 10: Kết quả huấn luyện YOLO V8

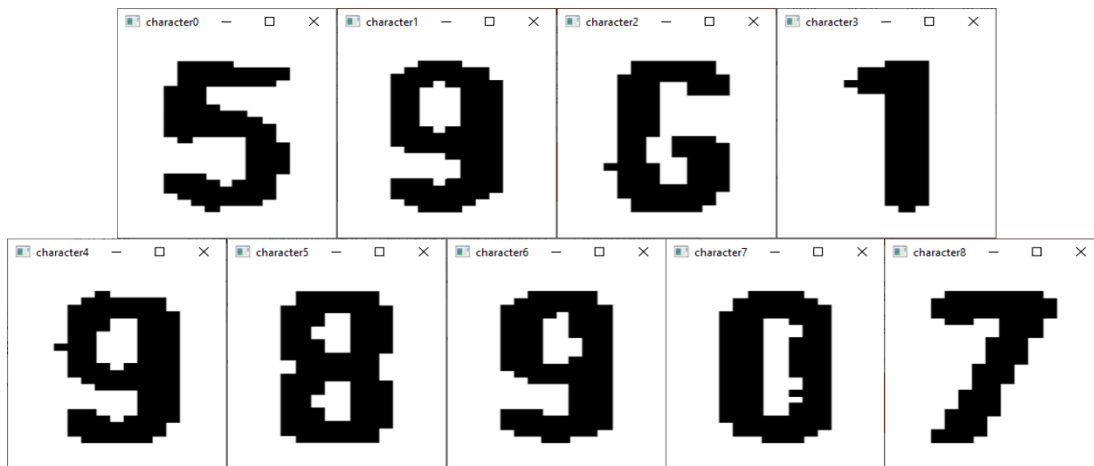
### 2.1.3. Xử lý ảnh với OpenCV

- Bước 1: Sau khi có được vị trí của biển số xe từ mô hình YOLO V8, thực hiện cắt, tách vùng biển số xe bằng hàm perspective.
- Bước 2: Thực hiện phân ngưỡng trên ảnh biển số xe vừa tách.
- Bước 3: Áp dụng hàm **findContours** để tạo các đường viền bao quanh các ký tự (bao gồm cả các nhiễu như dấu “.”, dấu “-“).
- Bước 4: Áp dụng hàm **boundingRect** để tạo ra các đường viền hình chữ nhật giới hạn (bouding box), bao quanh các contours vừa tìm được ở bước 3.
- Bước 5: Tính toán giá trị chiều cao và chiều rộng trung bình của các bounding box để loại bỏ nhiễu, chỉ giữ lại các ký tự có kích thước bounding box phù hợp.
- Bước 6: Sắp xếp các bouding box thành 2 dòng, từ trái qua phải dựa trên giá trị tọa độ x và y của các bouding box.





Hình 11: Ảnh gốc chưa xử lý



Hình 12: Kết quả của bước xử lý ảnh

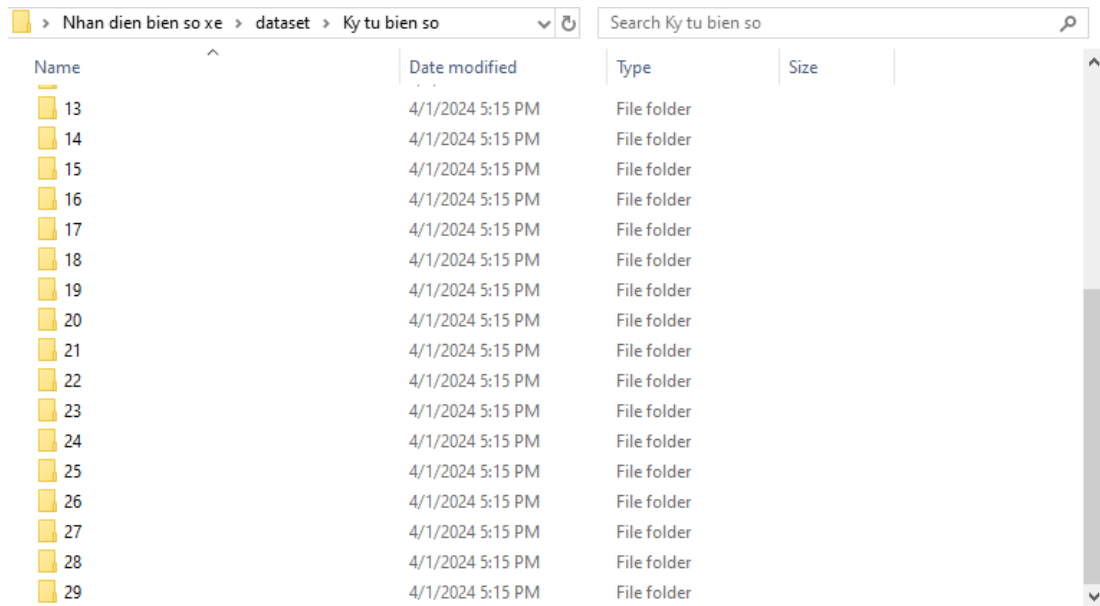
#### 2.1.4. Nhận dạng ký tự với model CNN

##### a) Giới thiệu mô hình CNN

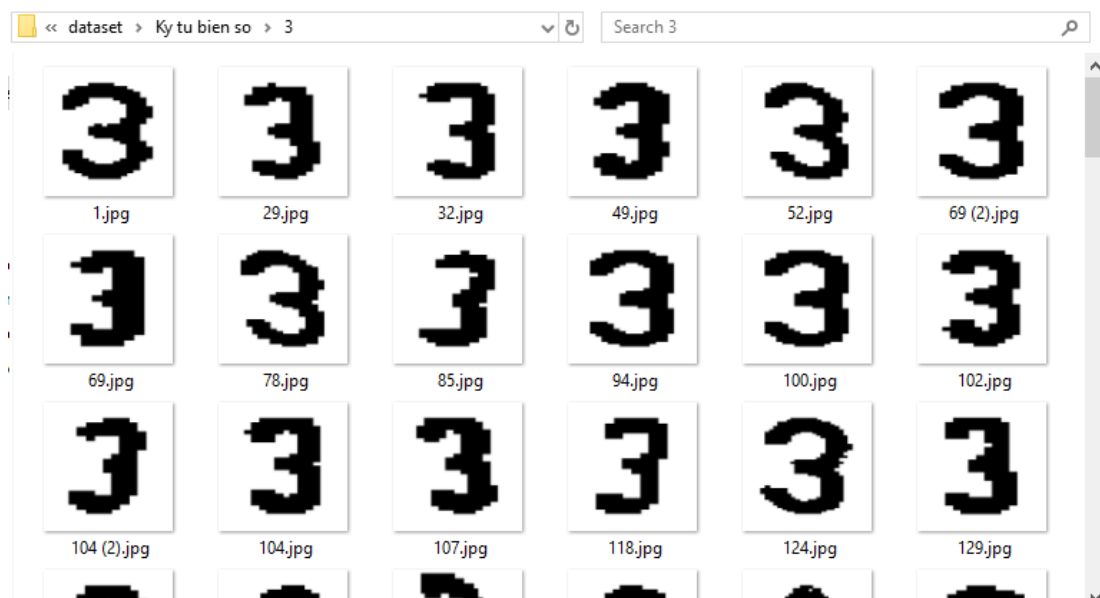
CNN (Convolutional Neural Networks) – mô hình mạng neural tích chập là một trong những mô hình Deep Learning tiên tiến, thường được sử dụng cho các bài toán liên quan đến thị giác máy tính, như phân loại ảnh, phát hiện đối tượng, và nhận dạng văn bản trong ảnh. CNN đã đạt được những thành tựu ấn tượng trong nhiều ứng dụng thực tế nhờ vào khả năng học và trích xuất các đặc trưng từ dữ liệu hình ảnh một cách tự động.

### ***b) Tập dữ liệu huấn luyện***

Tập dữ liệu huấn luyện là 1 thư mục gồm 30 thư mục con, mỗi thư mục con chứa ảnh của 1 ký tự cần nhận diện.



Hình 13: Thư mục cha chứa ảnh huấn luyện CNN



Hình 14: Ảnh huấn luyện CNN

### c) Huấn luyện model CNN

#### Xây dựng model CNN:

```
def create_alexnet():
    model = models.Sequential()

    model.add(layers.Conv2D(96, (11, 11), strides=(4, 4), activation='relu', input_shape=(227, 227, 3)))
    model.add(layers.MaxPooling2D((3, 3), strides=(2, 2)))
    model.add(layers.Conv2D(256, (5, 5), activation='relu'))
    model.add(layers.MaxPooling2D((3, 3), strides=(2, 2)))
    model.add(layers.Conv2D(384, (3, 3), activation='relu'))
    model.add(layers.Conv2D(384, (3, 3), activation='relu'))
    model.add(layers.Conv2D(256, (3, 3), activation='relu'))
    model.add(layers.MaxPooling2D((3, 3), strides=(2, 2)))

    model.add(layers.Flatten())

    model.add(layers.Dense(4096, activation='relu'))
    model.add(layers.Dense(4096, activation='relu'))
    model.add(layers.Dense(number_class, activation='softmax'))

    return model
```

Hình 15: Kiến trúc model CNN

#### Xử lý dữ liệu:

- Bước 1: Tải thư mục chứa ảnh huấn luyện.
- Bước 2: Thay đổi kích thước toàn bộ ảnh thành (227,227).
- Bước 3: Chia hình ảnh thành các tập khác nhau với nhãn tương ứng.

#### Huấn luyện:

Thực hiện train mô hình với 40 epochs. Sau 40 epochs, ta thu được kết quả sau:

```
Epoch 35/40
64/64 [=====] - 173s 3s/step - loss: 1.2256e-05 - accuracy: 1.0000 - val_loss: 0.3277 - val_accuracy: 0.9665
Epoch 36/40
64/64 [=====] - 160s 3s/step - loss: 1.0145e-05 - accuracy: 1.0000 - val_loss: 0.3301 - val_accuracy: 0.9665
Epoch 37/40
64/64 [=====] - 168s 3s/step - loss: 8.6245e-06 - accuracy: 1.0000 - val_loss: 0.3326 - val_accuracy: 0.9665
Epoch 38/40
64/64 [=====] - 155s 2s/step - loss: 7.3397e-06 - accuracy: 1.0000 - val_loss: 0.3352 - val_accuracy: 0.9665
Epoch 39/40
64/64 [=====] - 170s 3s/step - loss: 6.2844e-06 - accuracy: 1.0000 - val_loss: 0.3377 - val_accuracy: 0.9685
Epoch 40/40
64/64 [=====] - 160s 2s/step - loss: 5.4713e-06 - accuracy: 1.0000 - val_loss: 0.3399 - val_accuracy: 0.9665
20/20 [=====] - 12s 584ms/step - loss: 0.1824 - accuracy: 0.9780
Test accuracy: 0.977952778339386
```

Hình 16: Kết quả huấn luyện model CNN

## **2.2.Hệ thống nhúng kết hợp IOT**

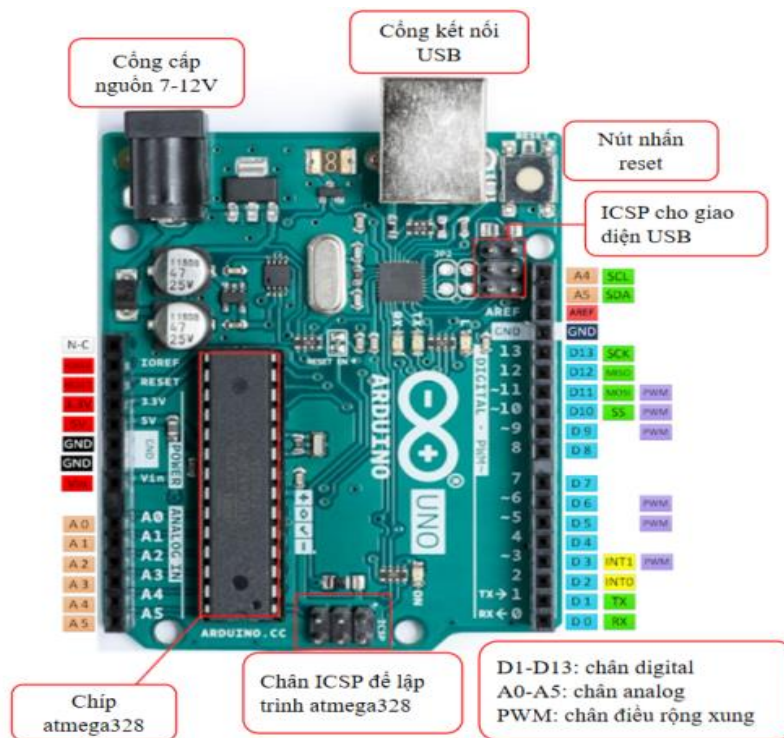
### **2.2.1. Giới thiệu về các thiết bị**

- Arduino Uno, ESP8266 Node MCU: sử dụng để lập trình điều khiển thiết bị.
- Thẻ từ: chứa các thông tin về phương tiện, chủ phương tiện.
- Barrier: sau khi xác thực được phương tiện sẽ được servo điều khiển để tự động đóng mở.
- Camera biển số đọc và ghi nhận thông tin về biển số xe.
- Camera toàn cảnh kiểm soát thông tin loại xe, người điều khiển, tổng thể khu vực ra vào trạm kiểm soát.
- Servo: điều khiển rào chắn.
- Module RFID để đọc thẻ.
- Nguồn cấp cho hệ thống.
- Cảm biến hồng ngoại phát hiện xe đã đi qua barrier và phát tín hiệu để đóng barrier.
- Gờ giảm tốc.
- Đèn led giao thông thông báo tình trạng đóng/mở của barrier.
- Button được sử dụng để đóng/mở barrier thủ công.

#### ***a) Arduino Uno R3***

Arduino là một nền tảng mã nguồn mở được sử dụng để xây dựng các dự án điện tử. Arduino bao gồm cả bảng mạch lập trình( thường được gọi là vi điều khiển) và một phần mềm( IDE) được sử dụng để lập trình viết và tải mã máy tính lên bo mạch.

### Sơ đồ chân của Arduino Uno R3:



Hình 17: Sơ đồ chân của Arduino Uno R3

### Thông số cơ bản:

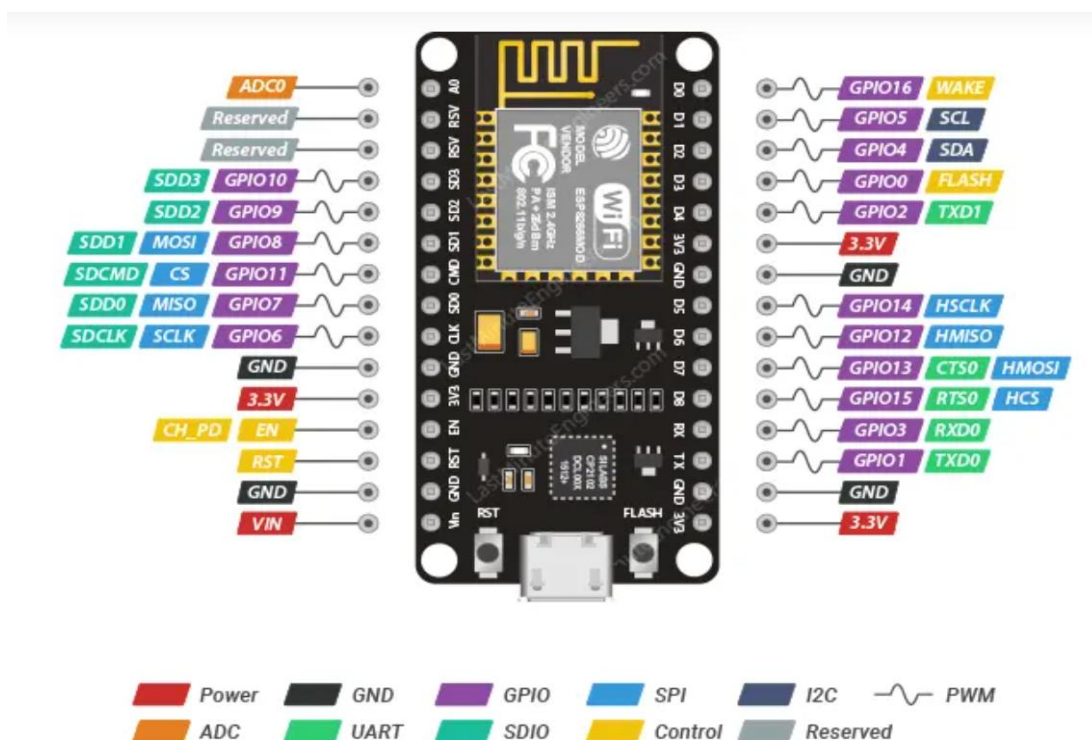
Vi điều khiển	Atmega 328 (họ 8 bit)
Điện áp hoạt động	5V – DC (cấp qua cổng USB)
Tần số hoạt động	16 MHz
Dòng tiêu thụ	30mA
Điện áp vào khuyến dùng	7 – 12V – DC
Điện áp vào giới hạn	6 – 20V – DC
Số chân Digital I/O	14 (6 chân PWM)
Số chân Analog	6 (độ phân giải 10 bit)
Dòng tối đa trên mỗi chân I/O	30 mA
Dòng ra tối đa (5V)	500 mA
Dòng ra tối đa (3.3V)	50 mA
Bộ nhớ flash	32 KB (Atmega328)
SRAM	2KB (Atmega328)
EEPROM	1KB (Atmega328)

Bảng 1: Thông số cơ bản của Arduino Uno R3

### ***b) ESP8266 NodeMCU***

ESP8266 NodeMCU là một nền tảng IoT mã nguồn mở, được phát triển bởi một nhóm kỹ sư tại Trung Quốc. Nền tảng này được xây dựng trên ESP8266, một vi điều khiển Wi-Fi SoC (System on a Chip) được sản xuất bởi Espressif Systems. NodeMCU cung cấp một bộ SDK để lập trình cho ESP8266 bằng ngôn ngữ Lua hoặc C++. Với các tính năng như Wi-Fi, GPIO, ADC, I2C, SPI, PWM và một số tính năng khác, NodeMCU ESP 8266 được sử dụng rộng rãi trong các ứng dụng IoT như kiểm soát thiết bị, thu thập dữ liệu và giao tiếp với các thiết bị khác.

#### Sơ đồ chân của ESP8266 NodeMCU:



Hình 18: Sơ đồ chân của ESP8266 NodeMCU

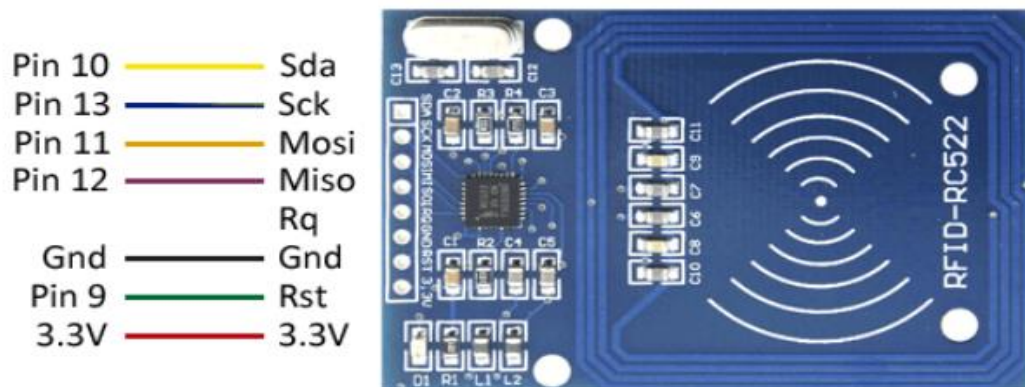
#### Thông số kỹ thuật:

- Microcontroller: ESP8266EX
- Điện áp hoạt động: 3.3V DC

- Số chân I/O: 17 chân GPIO
- Kết nối mạng: WiFi 802.11 b/g/n
- Giao diện mạng: TCP/IP
- Đồng hồ thời gian thực (RTC): không tích hợp
- Bộ nhớ trong: 4MB
- RAM: 80KB
- Cổng nạp: Micro-USB
- Hỗ trợ các giao thức: MQTT, CoAP, HTTP/HTTPS
- Kích thước: 49 x 24.5 x 13mm

### c) *Module RFID*

Sơ đồ chân của module RFID:



Hình 19: Sơ đồ chân của module RFID

Nguyên lý hoạt động:

RFID là viết tắt của “**Radio Frequency Identification**”, có nghĩa là nhận dạng



bằng tần số radio. Đây là một công nghệ sử dụng sóng radio để truyền và nhận dạng thông tin từ một thẻ hoặc một thiết bị RFID. Hệ thống RFID bao gồm ít nhất hai thành phần chính: một thẻ RFID (hay còn gọi là tag RFID) và một đầu đọc RFID (hay còn gọi là reader RFID).

Công nghệ RFID hoạt động theo nguyên lý khá đơn giản, đó là: đầu đọc RFID được đặt cố định ở một vị trí. Chúng sẽ phát ra sóng vô tuyến điện ở một tần số nhất định để phát hiện thiết bị phát xung quanh đó. Khi thẻ RFID đi vào vùng sóng vô tuyến điện mà đầu đọc RFID phát ra, hai bên sẽ cảm nhận được nhau.

#### ***d) Cảm biến vật cản hồng ngoại***

Cảm biến hồng ngoại được hiểu là thiết bị tự động được hoạt động dựa trên nguyên tắc điện dung, có chức năng đo và phát hiện những bức xạ hồng ngoại. Bức xạ hồng ngoại là những nguồn ánh sáng mà mắt của con người không thể nhìn thấy được.



Hình 20: Cảm biến vật cản hồng ngoại

Nguyên lý hoạt động: Vật thể nào cũng có thể phát ra được bức xạ hồng ngoại với mức độ nhiều hay ít. Vậy nên khi một người hoặc là vật thể lạ đi ngang qua thiết bị cảm biến thì sẽ xuất hiện ngay một tín hiệu. Và tín hiệu này sẽ được cảm biến thu vào và để cho mạch xử lý tạo tác dụng để điều khiển hoặc sẽ báo động khi cần thiết.



### Cấu tạo:

- Đèn led hồng ngoại: là thiết bị được phát ra từ nguồn sáng hồng ngoại.
- Máy dò hồng ngoại: là thiết bị nhận tín hiệu và phát hiện ra bức xạ hồng ngoại phản xạ trở lại.
- Điện trở: là thiết bị có tác dụng đi cường độ dòng điện quá lớn chạy qua đèn led làm cho hệ thống chập cháy.
- Dây điện: tác dụng chính là kết nối các chi tiết để tạo nên cảm biến hoạt động ổn định.

### ***e) Đèn led giao thông***

Mạch Led giao thông là một trong những module Led được sử dụng làm các mô hình điện tử, với 3 bóng đèn Led để mô phỏng mô hình giao thông chính xác.



Hình 21: Đèn led giao thông

### Thông số kỹ thuật:

- Kích thước: 56\*21\*11mm.
- Màu sắc: đỏ, vàng, xanh.
- Led: 3 led đục- đường kính bóng led 8mm.
- Điện áp: 5V.

- Trọng lượng: 25gram.

- 4 chân nối:

- GND.
- Red.
- Yellow.
- Green.

#### ***f) Button***

Button được sử dụng để bật tắt đèn hoặc phát ra âm thanh trong Arduino là một trong những hiệu ứng cơ bản, với việc kết nối 2 điểm trong mạch khi bạn nhấn chúng sẽ tạo thành mạch kín, giúp truyền tải dòng điện theo nhu cầu.



Hình 22: Button

#### ***g) Servo***

Servo là hệ thống hồi tiếp vòng kín và truyền động. Khi mà các động cơ Servo nhận được tín hiệu báo về từ lệnh PLC và từ đó thực hiện chính xác, nhanh chóng các thao tác.

Bộ Servo được xem là hoàn chỉnh sẽ gồm có: 1 motor Servo, 1 encoder và 1 Servo driver. Chức năng của động cơ Servo chính là thay đổi tốc độ chính xác, điều

khiểm vị trí, điều chỉnh momen sao cho phù hợp với những ứng dụng trong công việc. Thiết bị sẽ được làm việc dựa trên các cơ chế phản hồi âm về. Điều đó đồng nghĩa với việc nó sẽ chuyển đổi tín hiệu điện năng biến thành những chuyển động có thể kiểm soát.



Hình 23: Servo

Cấu tạo: gồm 2 phần chính: rotor, stator.

Nguyên lí điều khiển Servo:

Động cơ motor Servo sở hữu một rotor là một nam châm vĩnh cửu. Từ trường tạo ra mạnh. Stator của động cơ có thể quấn vào các cuộn dây một cách riêng biệt. Sau đó, chúng sẽ được cung cấp nguồn điện và sẽ hoạt động theo đúng một trình tự thích hợp nhất để làm quay động cơ rotor.

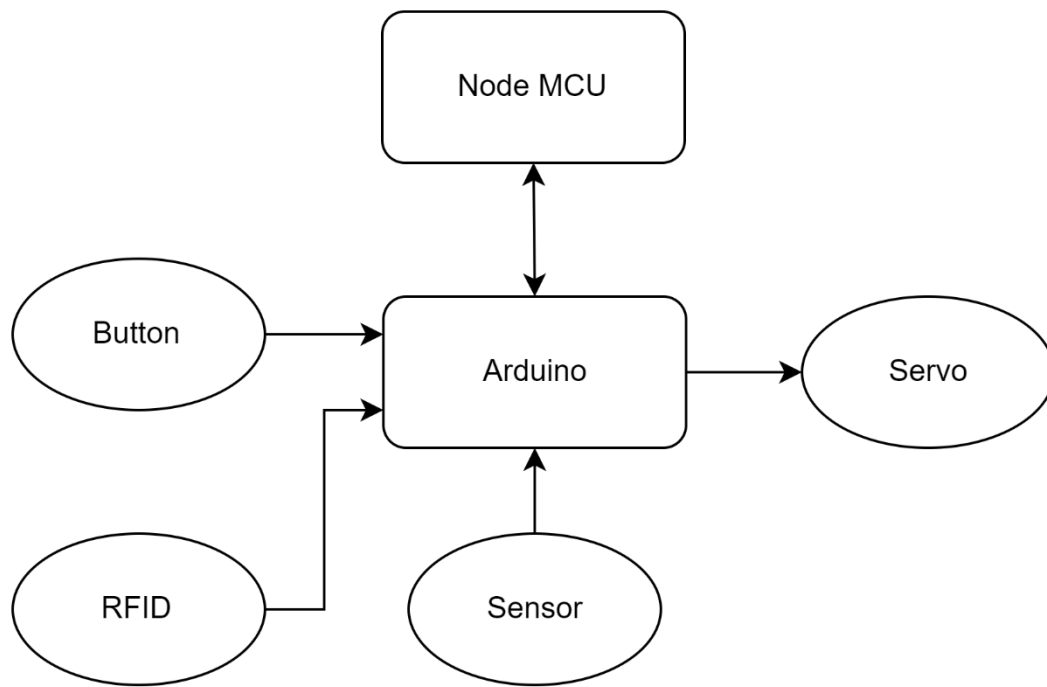
Khi truyền điện vào cuộn dây đúng thời điểm thì lúc đó chuyển động quay của động cơ rotor sẽ bị phụ thuộc vào pha của dòng điện và tần số của dòng điện. Do đó, dòng điện sẽ dần tới bị phân cực và chạy trong cuộn dây stator.

Động cơ Servo được tạo ra từ hệ thống vòng khép kín. Mạch điều khiển được nối với tín hiệu đầu ra của Servo. Khi mà động cơ quay liên tục kéo theo vận tốc và vị trí sẽ hồi tiếp rồi truyền đến mạch này.

Nếu có những tác nhân ảnh hưởng tới chuyển động quay và ngăn cản động cơ quay thì khi đó cơ cấu hồi tiếp sẽ nhận được tín hiệu chưa đạt vị trí như mong muốn. Sau đó mạch sẽ tiếp tục điều chỉnh, sửa chữa những sai lệch này cho đến khi mà động cơ đã đạt được vị trí và tốc độ chính xác nhất.

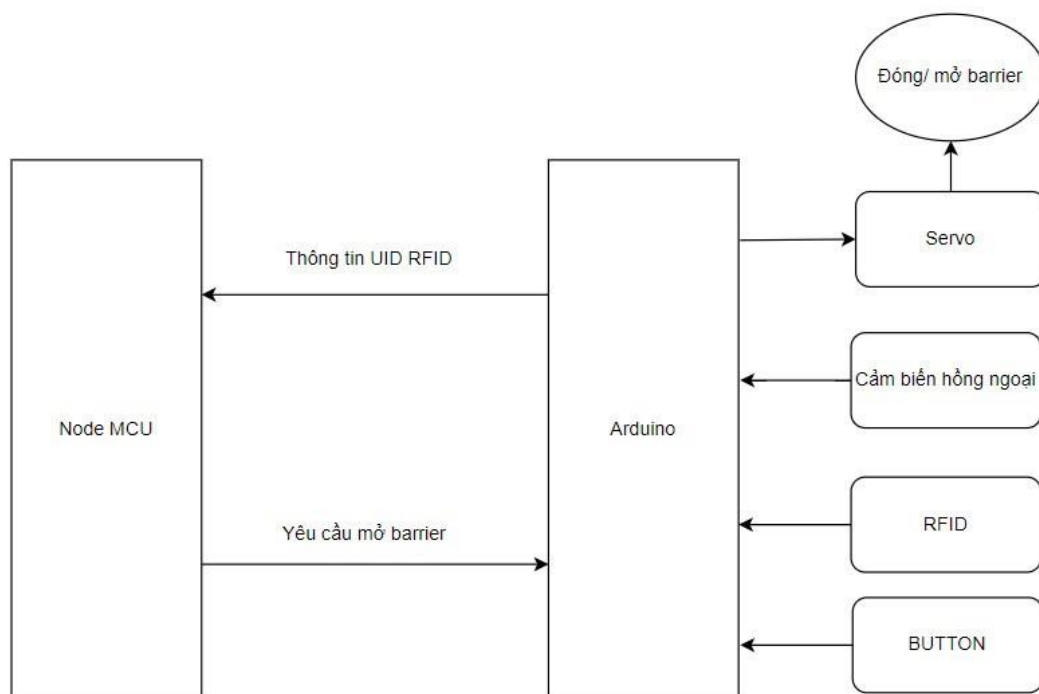
### ***2.2.2. Sơ đồ hoạt động***

#### ***a) Sơ đồ hệ thống phản cứng***



Hình 24: Sơ đồ hệ thống phản cứng

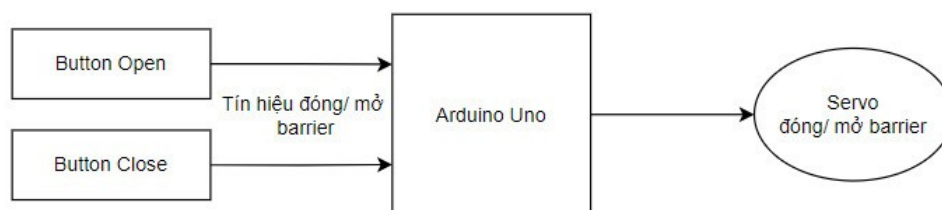
***b) Sơ đồ luồng dữ liệu mô hình***



Hình 25: Sơ đồ luồng dữ liệu mô hình

***c) Chế độ thủ công***

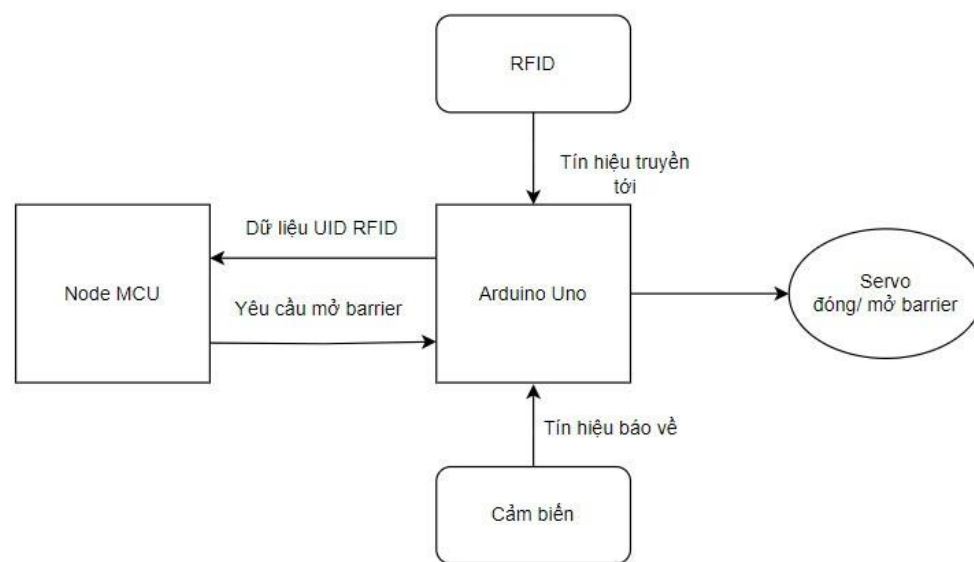
Đây là chế độ người dùng sẽ sử dụng các nút vật lý để điều khiển đóng/ mở barrier. Sau khi Button Open hoặc Button Close được nhấn sẽ gửi tín hiệu đóng/ mở barrier tới arduino và điều khiển servo hoạt động.



Hình 26: Sơ đồ hoạt động ở chế độ thủ công

#### ***d) Chế độ tự động***

Sau khi người dùng quét thẻ, thông tin về UID của thẻ được truyền tới Arduino Uno, sau đó sẽ được gửi lên Node MCU và được kiểm tra với thông tin dữ liệu UID trên server. Sau các bước xử lý dữ liệu, server sẽ gửi tín hiệu điều khiển tới Node MCU, từ đây, tín hiệu tiếp tục được gửi tới Arduino Uno. Cảm biến hồng ngoại được sử dụng để nhận biết đối tượng đã đi qua barrier hay chưa. Nếu nhận thấy đối tượng, lúc này Arduino Uno sẽ điều khiển servo thực hiện đóng barrier.



Hình 27: Sơ đồ hoạt động ở chế độ tự động

### ***2.2.3. Giao thức WebSocket***

#### ***a) WebSocket là gì?***

WebSoket là công nghệ hỗ trợ giao tiếp hai chiều giữa client và server bằng cách sử dụng một TCP socket để tạo một kết nối hiệu quả và ít tốn kém. Mặc dù được thiết kế để chuyên sử dụng cho các ứng dụng web, lập trình viên vẫn có thể đưa chúng vào bất kì loại ứng dụng nào.

WebSockets mới xuất hiện trong HTML5, là một kỹ thuật Reverse Ajax.

WebSockets cho phép các kênh giao tiếp song song hai chiều và hiện đã được hỗ trợ trong nhiều trình duyệt (Firefox, Google Chrome và Safari). Kết nối được mở thông qua một HTTP request (yêu cầu HTTP), được gọi là liên kết WebSockets với những header đặc biệt. Kết nối được duy trì để bạn có thể viết và nhận dữ liệu bằng JavaScript như khi bạn đang sử dụng một TCP socket đơn thuần.

Dữ liệu truyền tải thông qua giao thức HTTP (thường dùng với kỹ thuật Ajax) chứa nhiều dữ liệu không cần thiết trong phần header. Một header request/response của HTTP có kích thước khoảng 871 byte, trong khi với WebSocket, kích thước này chỉ là 2 byte (sau khi đã kết nối).

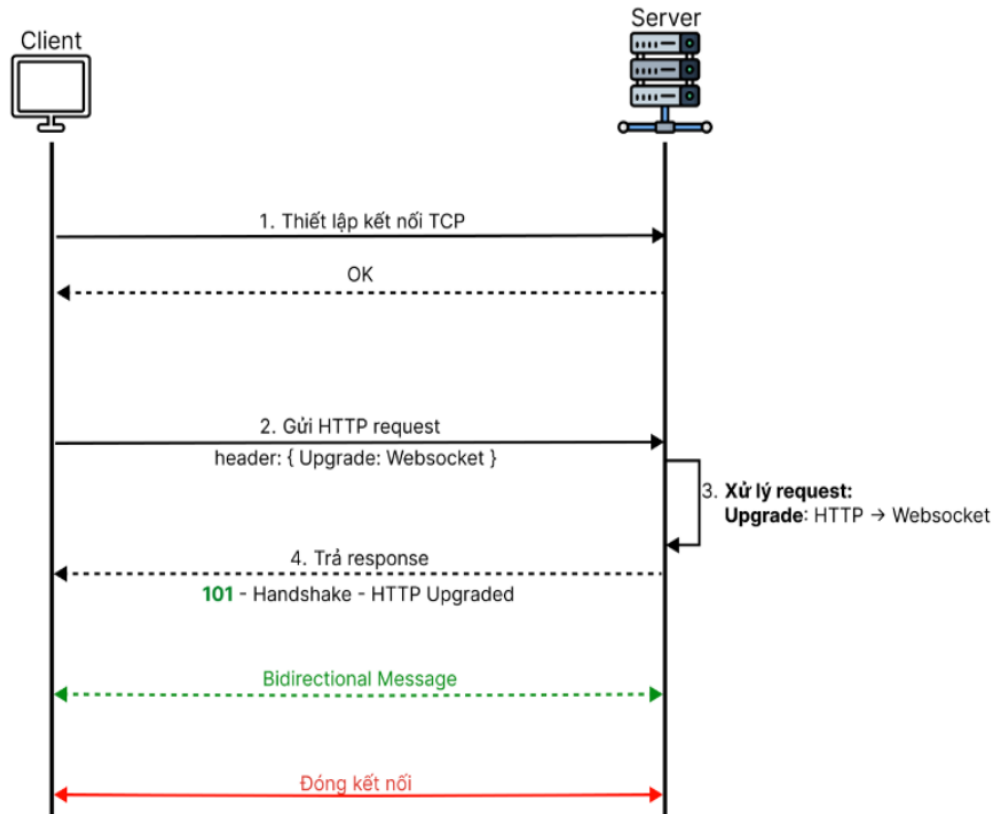
#### ***b) Các sự kiện chính trong WebSocket***

WebSocket cho phép các ứng dụng web tạo kết nối thời gian thực, có khả năng truyền dữ liệu hai chiều liên tục giữa máy chủ và trình duyệt. Các sự kiện chính của WebSocket bao gồm:

- Mở kết nối (Open): Khi một kết nối WebSocket được thiết lập giữa trình duyệt và máy chủ, sự kiện "open" được kích hoạt.
- Nhận dữ liệu (Message): Khi trình duyệt hoặc máy chủ nhận được dữ liệu mới, sự kiện "message" được kích hoạt.
- Gửi dữ liệu (Send): Khi trình duyệt hoặc máy chủ gửi dữ liệu, sự kiện "send" được kích hoạt.
- Đóng kết nối (Close): Khi kết nối WebSocket bị đóng, sự kiện "close" được kích hoạt.
- Lỗi (Error): Nếu có lỗi trong quá trình thiết lập hoặc sử dụng kết nối WebSocket, sự kiện "error" được kích hoạt.

- Các sự kiện này đều cung cấp các thông tin cần thiết cho ứng dụng để xử lý các tình huống khác nhau khi sử dụng WebSocket.

### c) Sơ đồ hoạt động của WebSocket



Hình 28: Sơ đồ hoạt động của WebSocket

Quy trình hoạt động của Websockets là một chuỗi các bước mà máy khách (client) và máy chủ (server) thực hiện để thiết lập và duy trì kết nối liên tục, cho phép truyền thông dữ liệu hai chiều thời gian thực. Dưới đây là quy trình hoạt động cơ bản của Websockets:

1. Client sẽ gửi 1 Request yêu cầu kết nối TCP lên server và nhận được Response cho phép từ server
2. Máy khách gửi một yêu cầu HTTP đặc biệt gọi là “WebSocket handshake



request” đến máy chủ thông qua một URL cụ thể, chẳng hạn như `ws://example.com/socket`. Yêu cầu này chứa một số thông tin, bao gồm phiên bản Websockets được sử dụng và một mã ngẫu nhiên gọi là “Sec-WebSocket-Key”.

3. Máy chủ nhận yêu cầu handshake, xác minh các thông tin và tạo ra một “WebSocket handshake response”.
4. Máy chủ gửi lại phản hồi HTTP với mã trạng thái 101 (“Switching Protocols”) và các tiêu đề bổ sung như “Upgrade” và “Connection” để thông báo rằng kết nối đã được nâng cấp thành Websockets.

#### ***Truyền Tin nhắn:***

- Cả máy khách và máy chủ có thể gửi và nhận tin nhắn qua kết nối Websockets.
- Tin nhắn Websockets thường được đóng gói trong các khung (frames) đặc biệt. Mỗi khung chứa dữ liệu cùng với các thông tin điều khiển như mã thao tác và mã xác định kiểu dữ liệu.

#### ***Đóng Kết nối:***

- Khi một bên muốn đóng kết nối, nó gửi một khung “close frame” để thông báo cho bên còn lại về việc đóng kết nối.
- Bên nhận cũng sẽ gửi một khung close frame như phản hồi để xác nhận việc đóng kết nối.
- Sau khi cả hai bên đã nhận được và xử lý khung close frame, kết nối Websockets sẽ được đóng.

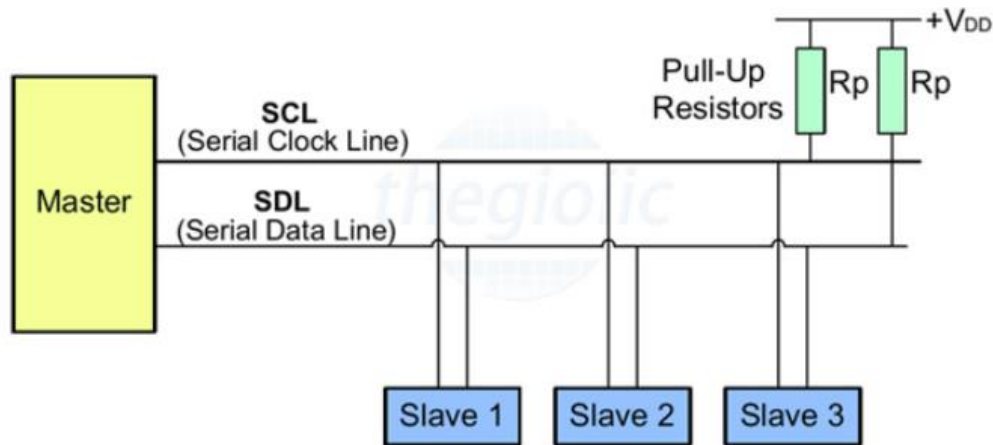
Quy trình trên giúp thiết lập và duy trì một kết nối liên tục giữa máy khách và máy chủ, cho phép truyền thông dữ liệu hai chiều thời gian thực. Kết nối này có thể tồn tại trong thời gian dài và cho phép các ứng dụng web tương tác và cập nhật dữ liệu một cách nhanh chóng và linh hoạt.

#### 2.2.4. Cách thức truyền dữ liệu theo chuẩn giao tiếp I2C

##### a) Chuẩn giao tiếp I2C là gì?

I2C hay IIC (Inter – Integrated Circuit) là 1 giao thức giao tiếp nối tiếp đồng bộ được phát triển bởi Philips Semiconductors, sử dụng để truyền nhận dữ liệu giữa các IC với nhau chỉ sử dụng hai đường truyền tín hiệu.

I2C kết hợp các tính năng tốt nhất của SPI và UART. I2C có thể kết nối nhiều slave với một master duy nhất (như SPI) và có thể có nhiều master điều khiển một hoặc nhiều slave. Điều này thực sự cần thiết khi muốn có nhiều hơn một vi điều khiển ghi dữ liệu vào một thẻ nhớ duy nhất hoặc hiển thị văn bản trên một màn hình LCD.



Hình 29: Sơ đồ kết nối I2C

I2C chỉ sử dụng hai dây để truyền dữ liệu giữa các thiết bị:

- SDA (Serial Data): đường truyền cho master và slave để gửi và nhận dữ liệu.

- SCL (Serial Clock): đường truyền mang tín hiệu xung nhịp.

Các bit dữ liệu sẽ được truyền từng bit một dọc theo một đường duy nhất (SDA) theo các khoảng thời gian đều đặn được thiết lập bởi 1 tín hiệu đồng hồ (SCL).

### ***b) Quá trình truyền nhận***

Khi bắt đầu Master sẽ gửi đi 1 xung Start bằng cách kéo lần lượt các đường SDA, SCL từ mức 1 xuống 0.

Tiếp theo đó, Master gửi đi 7 bit địa chỉ tới các Slave cùng với bit Read/Write.

Slave sẽ so sánh địa chỉ vừa được gửi tới. Nếu trùng khớp, Slave sẽ xác nhận bằng cách kéo đường SDA xuống 0 và set bit ACK/NACK bằng '0'. Nếu không trùng khớp thì SDA và bit ACK/NACK đều mặc định bằng '1'.

Thiết bị Master sẽ gửi hoặc nhận khung bit dữ liệu. Nếu Master gửi đến Slave thì bit Read/Write ở mức 0. Ngược lại nếu nhận thì bit này ở mức 1.

Nếu như khung dữ liệu đã được truyền đi thành công, bit ACK/NACK được set thành mức 0 để báo hiệu cho Master tiếp tục.

Sau khi tất cả dữ liệu đã được gửi đến Slave thành công, Master sẽ phát 1 tín hiệu Stop để báo cho các Slave biết quá trình truyền đã kết thúc bằng cách chuyển lần lượt SCL, SDA từ mức 0 lên mức 1.

## **2.3.Hệ thống Server**

### ***2.3.1. Front End***

#### ***a) Bootstrap***

**Bootstrap** là 1 framework HTML, CSS, và JavaScript cho phép người dùng dễ dàng thiết kế website theo 1 chuẩn nhất định, tạo các website thân thiện.

**Bootstrap** bao gồm những cái cơ bản có sẵn như: typography, forms, buttons,

tables, navigation, modals, image carousels và nhiều thứ khác. Trong bootstrap có thêm nhiều Component, Javascript hỗ trợ cho việc thiết kế reponsive của bạn dễ dàng, thuận tiện và nhanh chóng hơn.

**Bootstrap** là một trong những framework được sử dụng nhiều nhất trên thế giới để xây dựng nên một website. **Bootstrap** đã xây dựng nên 1 chuẩn riêng và rất được người dùng ưa chuộng. Chính vì thế, chúng ta hay nghe tới một cụm từ rất thông dụng "Thiết kế theo chuẩn **Bootstrap**".

Từ cái "chuẩn mực" này, chúng ta có thể thấy rõ được những điểm thuận lợi khi sử dụng Bootstrap:

- *Rất dễ để sử dụng*: Nó đơn giản vì được base trên HTML, CSS và Javascript chỉ cần có kiến thức cơ bản về 3 cái đó là có thể sử dụng bootstrap tốt.
- *Responsive*: **Bootstrap** xây dựng sẵn reponsive css trên các thiết bị Iphones, tablets, và desktops. Tính năng này khiến cho người dùng tiết kiệm được rất nhiều thời gian trong việc tạo ra một website thân thiện với các thiết bị điện tử, thiết bị cầm tay.
- *Tương thích với trình duyệt*: Nó tương thích với tất cả các trình duyệt (Chrome, Firefox, Internet Explorer, Safari, and Opera). Tuy nhiên, với IE browser, **Bootstrap** chỉ hỗ trợ từ IE9 trở lên. Điều này vô cùng dễ hiểu vì IE8 không support HTML5 và CSS3.

Có hai cách để sử dụng **Bootstrap**:

- Download Bootstrap packet từ [getbootstrap.com](http://getbootstrap.com).
- Thêm Bootstrap từ CDN.

## ***b) EJS***

**EJS**, viết tắt của “**Embedded JavaScript**” là một ngôn ngữ mẫu (template language) được sử dụng trong phát triển ứng dụng web. Nó giúp phát triển các trang web động bằng cách cho phép tạo ra các mẫu HTML được kết hợp với mã JavaScript. **EJS** thường được sử dụng cùng với Node.js và các framework web như Express để tạo giao diện người dùng động dựa trên dữ liệu từ máy chủ.

Một số đặc điểm quan trọng của EJS:

- *Tích hợp dễ dàng*: **EJS** tích hợp dễ dàng vào các dự án **Node.js** thông qua **npm** (Node Package Manager).
- *Cú pháp gần gũi với HTML*: Mã **EJS** được nhúng trực tiếp vào các tệp HTML, làm cho cú pháp rất gần gũi và dễ đọc. Điều này giúp người phát triển dễ dàng thao tác với mã HTML cùng với mã JavaScript.
- *Hiển thị dữ liệu động*: Với **EJS**, bạn có thể hiển thị dữ liệu động từ máy chủ trên trang web. Bằng cách sử dụng các biến và mã JavaScript, bạn có thể hiển thị danh sách, dữ liệu cơ sở dữ liệu, và thông tin động khác trên trang web.
- *Cấu trúc kiến thức*: **EJS** hỗ trợ cấu trúc kiến thức như vòng lặp, câu điều kiện, và các hàm tạo mẫu. Điều này cho phép bạn tạo các trang web phức tạp và linh hoạt.
- *Thay thế biến môi trường*: Bạn có thể sử dụng biến môi trường trong **EJS** để truyền các giá trị môi trường như đường dẫn tới tệp tin, ngôn ngữ, và cài đặt khác vào các mẫu.
- *Phát triển web đa ngôn ngữ*: **EJS** hỗ trợ phát triển các ứng dụng web đa ngôn ngữ bằng cách cho phép bạn thay đổi các văn bản trong mẫu dựa trên ngôn ngữ.

ngữ được chọn.

- *Tương thích với nhiều trình duyệt:* Mã **EJS** được tạo ra dưới dạng mã HTML chuẩn, do đó nó tương thích với hầu hết các trình duyệt web phổ biến.

### 2.3.2. *Back End*

#### *a) Node.js*

Định nghĩa trên trang chủ của **Node.js**: **Node.js** là một JavaScript runtime được build dựa trên Chrome's V8 JavaScript engine. **Node.js** sử dụng mô hình event-driven, non-blocking I/O khiến nó trở nên nhẹ và hiệu quả.

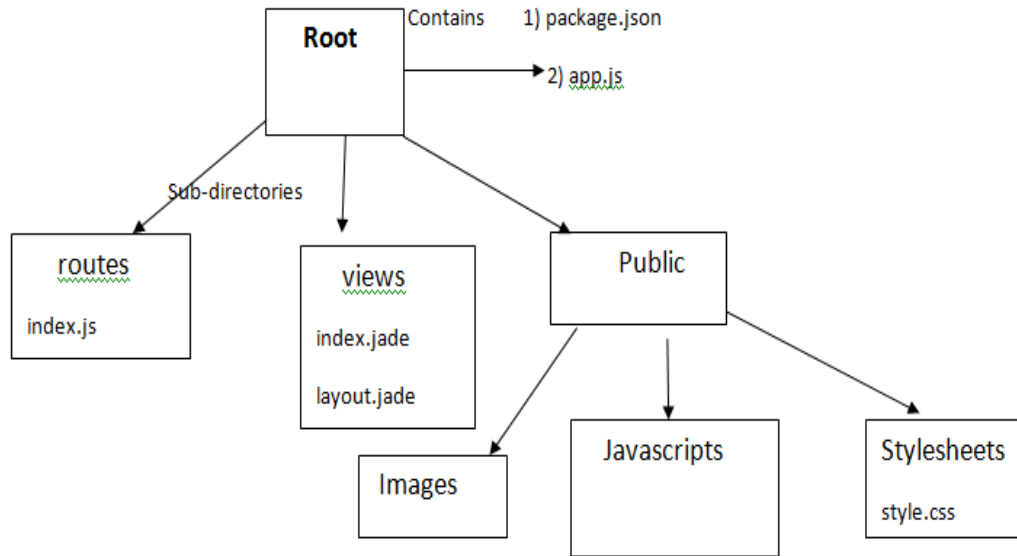
**Node** đặc biệt thích hợp để xây dựng những app cần đến tương tác hay hợp tác real-time, ví dụ như chat site, hoặc app kiểu CodeShare, nơi mà bạn có thể xem document được chỉnh sửa trực tiếp bởi người khác. **Node** cũng phù hợp để tạo các API nơi bạn cần xử lý một lượng lớn request liên quan đến I/O hoặc với những site liên quan đến data streaming, **Node** mang lại khả năng xử lý file trong khi chúng đang trong quá trình upload. Nếu bạn quan tâm đến vấn đề real-time.

Bên cạnh tốc độ và khả năng mở rộng, việc dùng JavaScript trên server và trình duyệt giúp cho bạn đỡ phải chuyển đổi giữa các ngôn ngữ. Bạn có thể làm bất cứ thứ gì với duy nhất một ngôn ngữ. Một điểm mạnh nữa của **Node** đó là nó cực thích hợp với JSON. JSON được coi là lý tưởng khi được sử dụng bởi một chương trình JavaScript. Khi làm việc với **Node**, dữ liệu có thể chuyển qua các tầng layer mà không cần phải reformat. Cuối cùng, JavaScript thực sự quá phổ biến, điều này giúp cho việc chuyển sang phát triển **Node** app dễ dàng hơn so với những ngôn ngữ phía server khác.

#### *b) Express*

**Expressjs** là một framework được xây dựng trên nền tảng của **Nodejs**. Nó cung

cấp các tính năng mạnh mẽ để phát triển web hoặc mobile. **Expressjs** hỗ trợ các method HTTP và middleware tạo ra API vô cùng mạnh mẽ và dễ sử dụng.



Hình 30: Cấu trúc thư mục của Express

Tính năng của **Express**:



Hình 31: Các tính năng của Express

- *Phát triển máy chủ nhanh hơn:* **ExpressJS** tối ưu hóa cú pháp và cung cấp các phương thức và hàm tiện ích để xử lý các tác vụ phổ biến trong lập trình web. Nhờ đó, bạn có thể viết code ngắn gọn và tối giản hóa quy trình phát triển.
- *Định tuyến (Routing):* **ExpressJS** cung cấp một hệ thống định tuyến mạnh mẽ, cho phép bạn xác định các tuyến đường (routes) để xử lý yêu cầu từ người dùng và phản hồi tương ứng. Điều này giúp tổ chức và quản lý các thành phần của ứng dụng một cách dễ dàng.
- *Middleware:* **ExpressJS** hỗ trợ middleware, cho phép bạn thêm các chức năng trung gian vào quy trình xử lý yêu cầu và phản hồi. Middleware giúp xác thực người dùng, ghi log, xử lý lỗi, nén dữ liệu và thực hiện nhiều tác vụ khác một cách linh hoạt.
- *Cấu hình môi trường:* **ExpressJS** cung cấp một cách để cấu hình môi trường phát triển và môi trường sản xuất. Bạn có thể thiết lập các biến môi trường, cấu hình định dạng và quy tắc, tùy chỉnh ứng dụng của mình theo các môi trường khác nhau.
- *Xử lý lỗi:* **ExpressJS** cung cấp cơ chế xử lý lỗi cho phép bạn kiểm soát và xử lý các lỗi xảy ra trong quá trình xử lý yêu cầu. Bạn có thể tạo ra các middleware để xử lý lỗi và phản hồi với các thông báo lỗi tùy chỉnh.

Ưu điểm của **Express**:

- *Đơn giản và dễ sử dụng:* **ExpressJS** có cú pháp đơn giản và dễ hiểu, giúp lập trình viên dễ dàng nắm bắt và triển khai các tính năng.
- *Linh hoạt:* **ExpressJS** không áp đặt một cấu trúc cụ thể, cho phép lập trình



viên tự do tùy chỉnh và xây dựng ứng dụng theo ý muốn.

- *Hỗ trợ middleware*: **ExpressJS** cung cấp hệ thống middleware mạnh mẽ, cho phép thực hiện các chức năng như xác thực, ghi log, nén dữ liệu và xử lý lỗi một cách linh hoạt và dễ dàng.
- *Hiệu suất cao*: **ExpressJS** được xây dựng trên Node.js, nền tảng có hiệu suất cao, cho phép xử lý nhanh chóng các yêu cầu web đồng thời và có khả năng mở rộng tốt.

Nhược điểm của **Express**:

- *Thiếu cấu trúc*: Do **ExpressJS** không áp đặt một cấu trúc nghiêm ngặt, việc tổ chức dự án và quản lý mã nguồn có thể trở nên khó khăn, đặc biệt khi ứng dụng phát triển lớn và phức tạp.
- *Khả năng mở rộng*: Khi ứng dụng phát triển lớn và phức tạp, việc quản lý mã nguồn và mở rộng có thể trở nên khó khăn với **ExpressJS**. Cần có sự kiểm soát cẩn thận để tránh sự phức tạp và rối rắm trong việc quản lý các module và tương tác giữa chúng.
- *Cộng đồng hỗ trợ*: Mặc dù **ExpressJS** có một cộng đồng lớn và đầy đủ tài liệu, tuy nhiên, không đạt được mức độ hỗ trợ như các framework web khác như Angular hoặc React.

## CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

### 3.1.Sơ đồ Use – Case

#### 3.1.1. Các tác nhân và ca sử dụng

Sơ đồ use-case cho biết cách người dùng (actor) sẽ tương tác với hệ thống. Trong hệ thống quản lý này, có 3 tác nhân tham gia tương tác là: quản trị viên hệ thống, nhân viên an ninh và người dùng.

##### *a) Quản trị viên hệ thống (Admin):*

- Thêm thẻ.
- Xóa thẻ.
- Cập nhật thông tin thẻ.
- Xem thông tin thẻ.
- Thêm nhân viên,
- Xóa nhân viên.
- Phân quyền.
- Quản lý nhân viên (xem).
- Xem lịch sử ra vào.
- Thống kê doanh thu khách vắng lai.

##### *b) Nhân viên an ninh:*

- Cập nhật lịch sử ra vào.

- Đóng barrier.
- Mở barrier.

***c) Người dùng (cư dân):***

- Đăng kí tài khoản.
- Đăng nhập.
- Đăng xuất.
- Xem thông tin thẻ.
- Xem lịch sử ra vào của xe.
- Yêu cầu thêm, sửa, xóa thẻ.

***3.1.2. Mô tả kịch bản các ca sử dụng***

***a) Kịch bản sử dụng ca đăng nhập***

- Người dùng yêu cầu đăng nhập.
- Hệ thống hiển thị giao diện đăng nhập.
- Người dùng điền thông tin tên đăng nhập, mật khẩu, xác nhận đăng nhập.
- Hệ thống kiểm tra thông tin đăng nhập.
  - Hợp lệ, đăng nhập thành công, chuyển thao tác hoạt động của hệ thống.
  - Không hợp lệ, thông báo người dùng sai thông tin hoặc tài khoản không tồn tại, quay lại bước đăng nhập (có thể đăng ký).

***b) Kịch bản sử dụng ca đăng ký tài khoản***

- Người dùng yêu cầu đăng ký tài khoản.

- Hệ thống hiện giao diện đăng ký tài khoản.
- Người dùng điền đầy đủ thông tin hệ thống yêu cầu gồm mã thẻ, mật khẩu, thông tin cá nhân, sau đó yêu cầu đăng ký.
- Hệ thống kiểm tra thông tin đăng ký. Hợp lệ, thông báo “Tạo tài khoản thành công” đồng thời lưu thông tin tài khoản của người dùng trên web và hệ thống.

***c) Kịch bản sử dụng ca tạo thẻ***

- Quản lý (người dùng) yêu cầu tạo thẻ.
- Hệ thống hiện giao diện đăng ký tạo thẻ.
- Quản lý điền đầy đủ thông tin vào các mục trong giao diện mà hệ thống yêu cầu, bao gồm các thông tin: Tên chủ hộ, ID căn hộ, tên chủ thẻ, ID thẻ, loại thẻ (vé tháng, ...), ngày tạo thẻ.
- Hệ thống kiểm tra thông tin đăng ký. Không có thông tin gì sai thì yêu cầu tạo thẻ.
- Hệ thống thông báo tạo thẻ thành công và lưu thông tin thẻ trên hệ thống.

***d) Kịch bản sử dụng ca yêu cầu thêm, sửa, xóa thẻ***

- Người dùng gửi yêu cầu về việc thêm, sửa hoặc xóa thẻ.
- Hệ thống hiện giao diện form điền thông tin yêu cầu.
- Người dùng thực hiện điền thông tin vào form:
  - Nếu người dùng muốn thêm thẻ cần cung cấp: tên chủ hộ, số căn hộ, tên chủ thẻ.
  - Nếu người dùng muốn sửa thẻ cần cung cấp thông tin muốn sửa và thông tin sau khi đã sửa lại.

- Nếu người dùng muốn xóa thẻ cần cung cấp tên chủ thẻ, tên chủ hộ, số căn hộ và lý do muốn xóa thẻ.
- Sau khi thực hiện gửi yêu cầu thêm, sửa hoặc xóa thành công, hệ thống sẽ hiện thông báo đã tiếp nhận yêu cầu thành công, đồng thời hệ thống cũng lưu trữ lại tất cả các yêu cầu đã tiếp nhận từ người dùng.

***e) Kịch bản sử dụng ca xem thông tin thẻ***

- Người dùng truy cập website của hệ thống.
- Hệ thống sẽ hiển thị giao diện có bảng gồm các mục, trong đó có mục quản lý thẻ.
- Người dùng yêu cầu xem thông tin thẻ.
- Hệ thống sẽ hiển thị giao diện danh sách các thông tin chi tiết được lưu trữ trong thẻ bao gồm: tên chủ thẻ, tên chủ hộ, số căn hộ, ID thẻ, loại thẻ (thẻ tháng, ...), ngày tạo thẻ.

***f) Kịch bản sử dụng ca đóng/mở barrier (tự động)***

- Khi người dùng vào bãi xe:
  - Người dùng (sử dụng phương tiện cá nhân) di chuyển đến trước khu vực barrier để tiến hành vào bãi xe.
  - Người dùng đứng trước vạch kẻ quy định, tiến hành sử dụng thẻ ra vào cá nhân quét thẻ.
  - Hệ thống tại cabin kiểm soát sẽ đọc thẻ của người dùng để xác nhận thông tin. Hệ thống sẽ ghi nhận biển kiểm soát của phương tiện, thời gian vào bãi, ảnh chụp toàn cảnh phương tiện và người điều khiển phương tiện, đồng thời sẽ lưu các thông tin đó vào thẻ người dùng khi vào bãi xe.

- Hệ thống sau đó ghi nhận phương tiện đã có mặt trong bãi xe.
- Ngay sau khi các thông tin được hệ thống ghi nhận một cách nhanh chóng thì barrier sẽ được tự động mở lên, các thông tin đồng thời được lưu vào hệ thống.
- Khi người dùng ra khỏi bãi xe:
  - Người dùng sử dụng phương tiện di chuyển đến vạch dừng trước barrier.
  - Người dùng sử dụng thẻ ra vào cá nhân để thực hiện quét thẻ.
  - Hệ thống sẽ kiểm tra các thông tin trong thẻ của người dùng.
  - Hệ thống sẽ kiểm tra trạng thái tồn tại của xe trong bãi, nếu đã xuất hiện trong bãi xe thì có nghĩa là xe có thể đi ra, ngược lại là người dùng có thể đã sử dụng sai phương tiện hoặc thẻ để đi ra.
  - Hệ thống đồng thời sẽ đối chiếu biển kiểm soát đọc được từ camera với biển kiểm soát đã lưu trong thẻ lúc xe vào, đối chiếu ID thẻ, số căn hộ, tên chủ hộ:
    - Nếu thông tin trùng hết thì xe có thể ra.
    - Nếu trùng biển kiểm soát mà khác ID thẻ thì đối chiếu số căn hộ và tên chủ hộ, nếu trùng thì xe có thể đi ra, nếu không trùng thì khả năng là xe bị cướp.
  - Sau khi đối chiếu các thông tin xong barrier sẽ tự động mở, các thông tin được đối chiếu được lưu vào hệ thống.
- Đối với khách vắng lai ra vào bãi xe:
  - Lúc vào:

- Người dùng tiếp nhận thẻ từ nhân viên an ninh, người dùng quẹt thẻ.
  - Hệ thống sẽ đọc biển số, chụp ảnh toàn cảnh phương tiện và người điều khiển phương tiện, ghi nhận thời gian vào bãi xe vào thẻ, trạng thái (có trong bãi).
  - Server gửi lệnh điều khiển barrier mở lên.
- Lúc ra:
- Người dùng quẹt thẻ.
  - Hệ thống đối chiếu biển kiểm soát, trạng thái.
  - Server gửi lệnh điều khiển barrier mở.

***g) Kịch bản sử dụng ca đóng/mở barrier (thủ công)***

- Người dùng (nhân viên an ninh) sử dụng chức năng nhập thủ công.
- Người dùng nhập thông tin: biển số xe, tên chủ thẻ, tên chủ hộ, mã căn hộ.
- Người dùng nhấn nút vật lý để đóng/mở barrier.

***h) Kịch bản sử dụng ca xem lịch sử ra vào***

- Người dùng truy cập website hệ thống.
- Người dùng đăng nhập.
- Người dùng chọn chức năng xem lịch sử ra vào.
- Hệ thống hiển thị lịch sử ra vào mặc định theo dòng thời gian, lần ra vào gần nhất sẽ được hiển thị trên cùng.
- Hệ thống hiển thị các thông tin về lịch sử ra vào gồm: tên chủ thẻ, tên chủ hộ, mã căn hộ, id thẻ, loại phương tiện, biển kiểm soát, thời gian vào bãi, thời gian ra bãi.

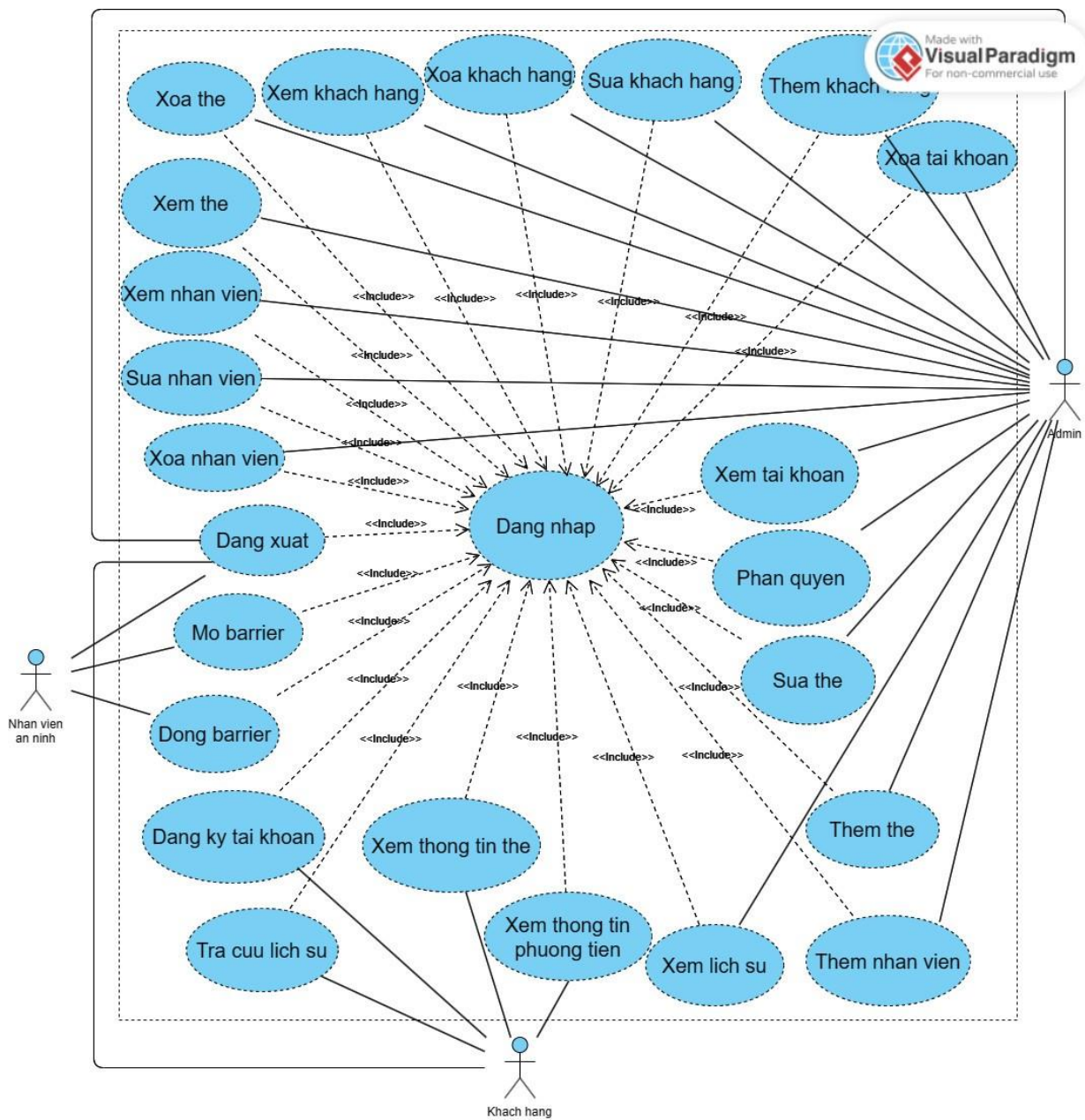
- Hệ thống hiển thị chức năng bộ lọc, tăng giảm dần theo thời gian, lọc khoảng thời gian, lọc theo loại phương tiện, ...
- Hệ thống hiển thị chức năng tìm kiếm theo tên chủ thẻ, id thẻ, ...

***i) Kịch bản sử dụng ca đăng xuất***

- Người dùng ấn nút đăng xuất.
- Hệ thống kết thúc phiên, giao diện hiện thị trang đăng nhập.

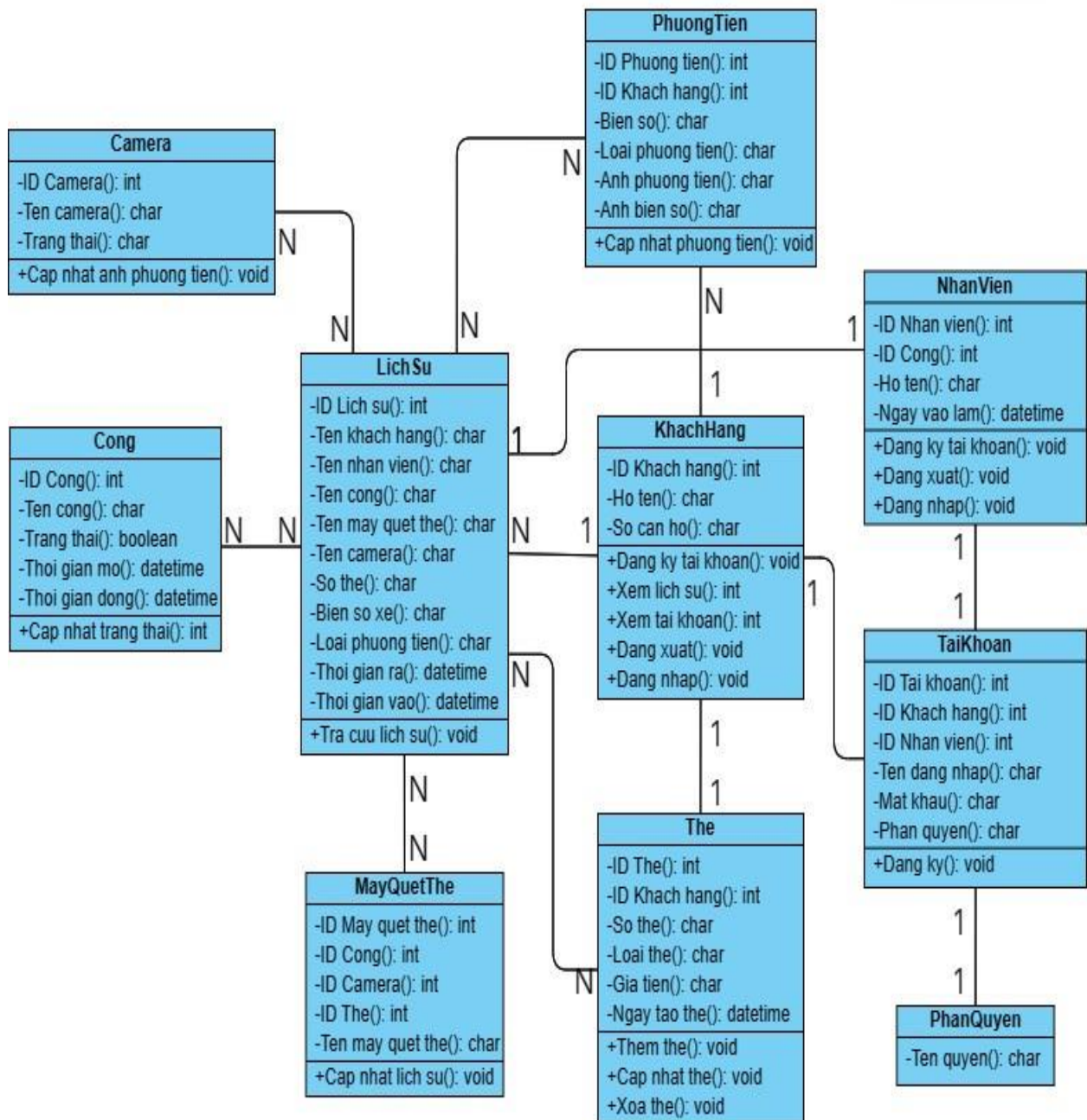


### 3.1.3. Sơ đồ Use – Case



Hình 32: Sơ đồ Use – Case

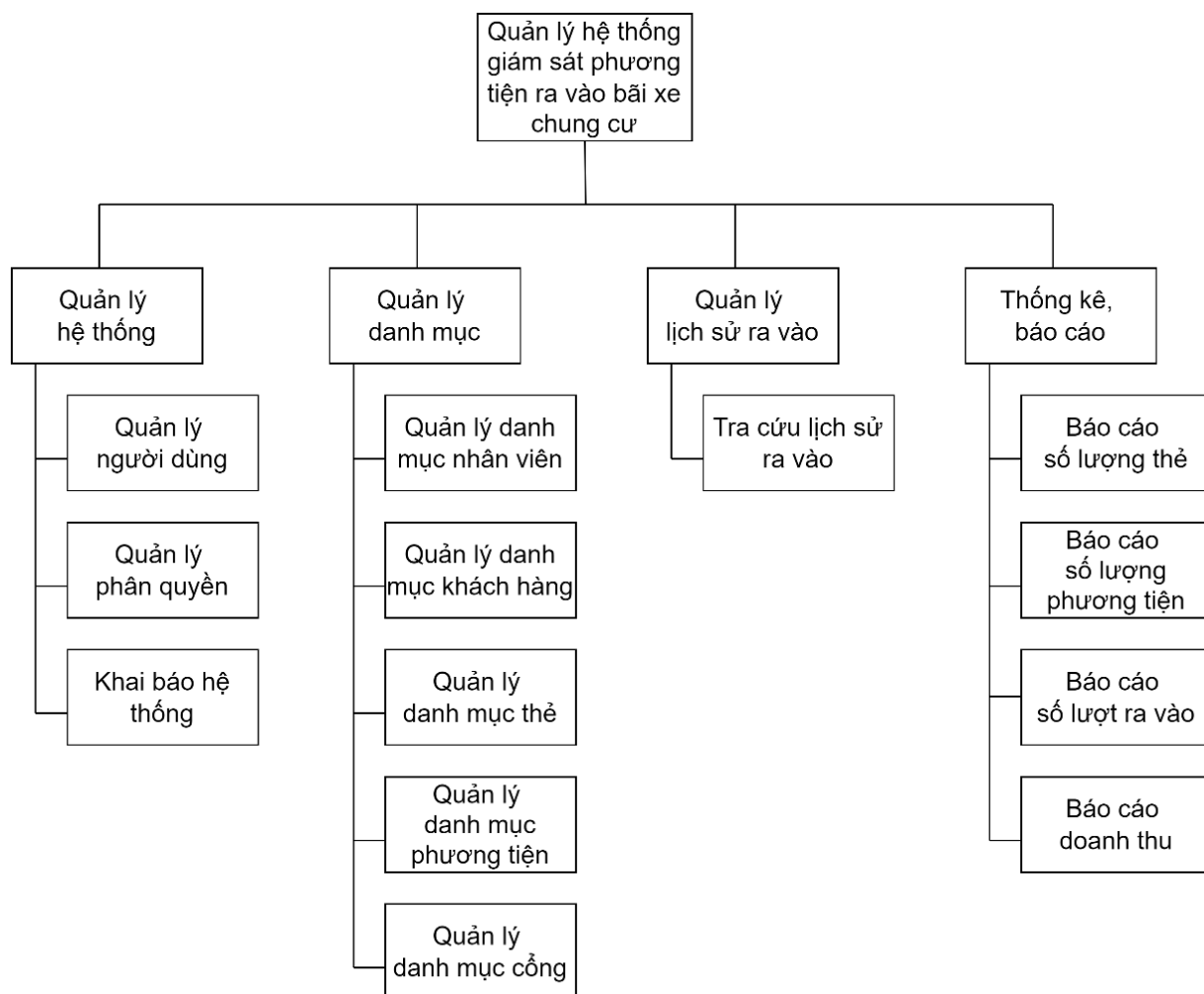
### 3.1.4. Sơ đồ lớp



Hình 33: Sơ đồ lớp

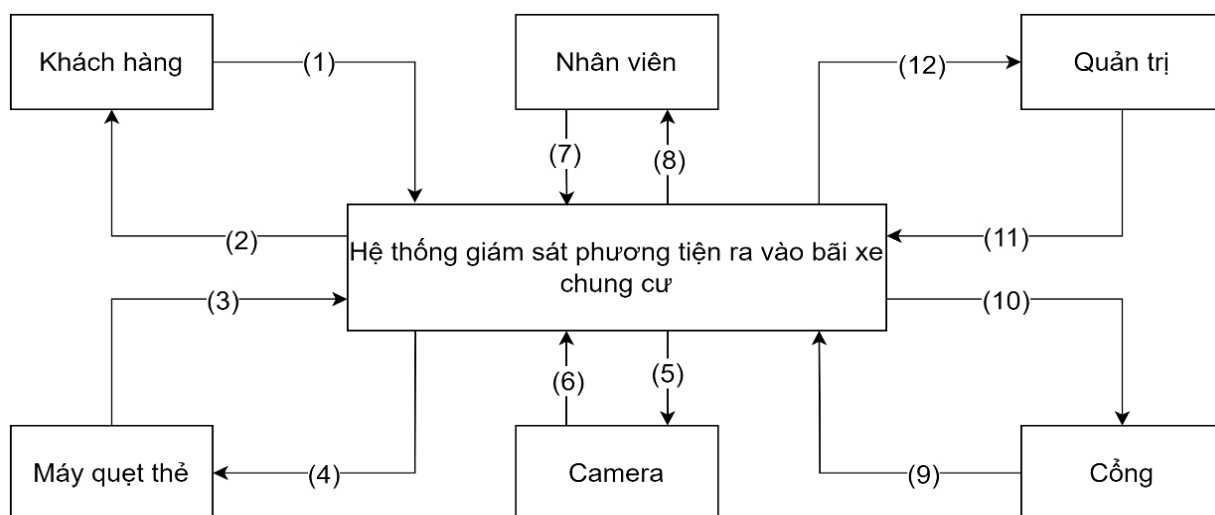
## 3.2.CSDL – Liên kết

### 3.2.1. Sơ đồ phân rã chức năng



Hình 34:Sơ đồ phân rã chức năng

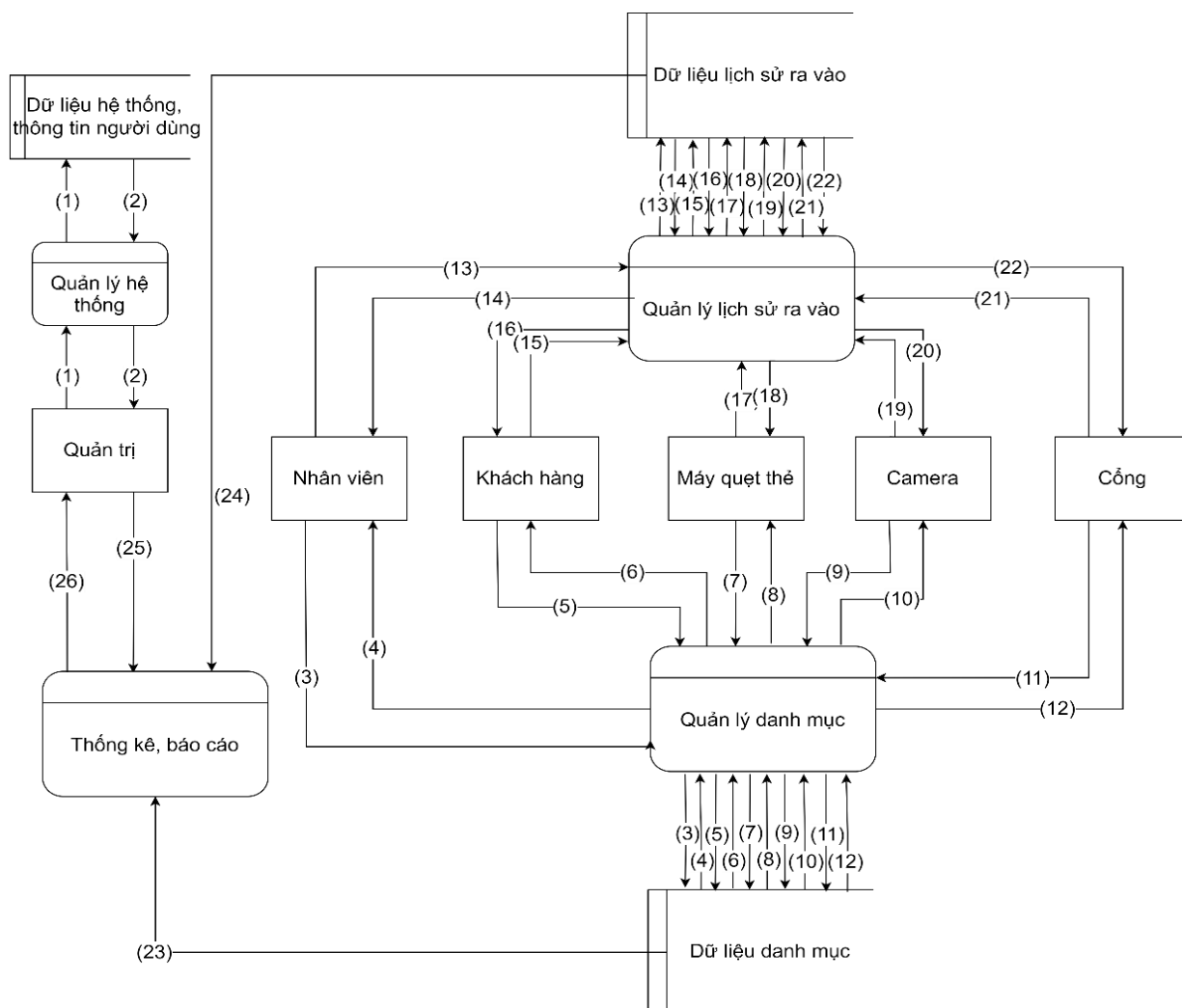
### 3.2.2. Sơ đồ mức ngữ cảnh



Hình 35: Sơ đồ mức ngữ cảnh

- (1) Khách hàng yêu cầu ra vào bãi xe.
- (2) Tín hiệu đèn báo dừng lại trước barrier.
- (3) Thông tin thẻ của khách hàng.
- (4) Tiếng báo quét thẻ thành công.
- (5) Yêu cầu nhận diện, chụp ảnh.
- (6) Thông tin biển số, ảnh biển số, ảnh phương tiện và khách hàng.
- (7) Yêu cầu đóng mở cổng thủ công.
- (8) Thông tin, trạng thái cổng barrier.
- (9) Thông tin, trạng thái cổng barrier.
- (10) Dữ liệu cập nhật thông tin, trạng thái cổng barrier.
- (11) Yêu cầu báo cáo thống kê.
- (12) Thông tin báo cáo thống kê khách hàng, thẻ, phương tiện, doanh thu.

### 3.2.3. Sơ đồ mức đỉnh



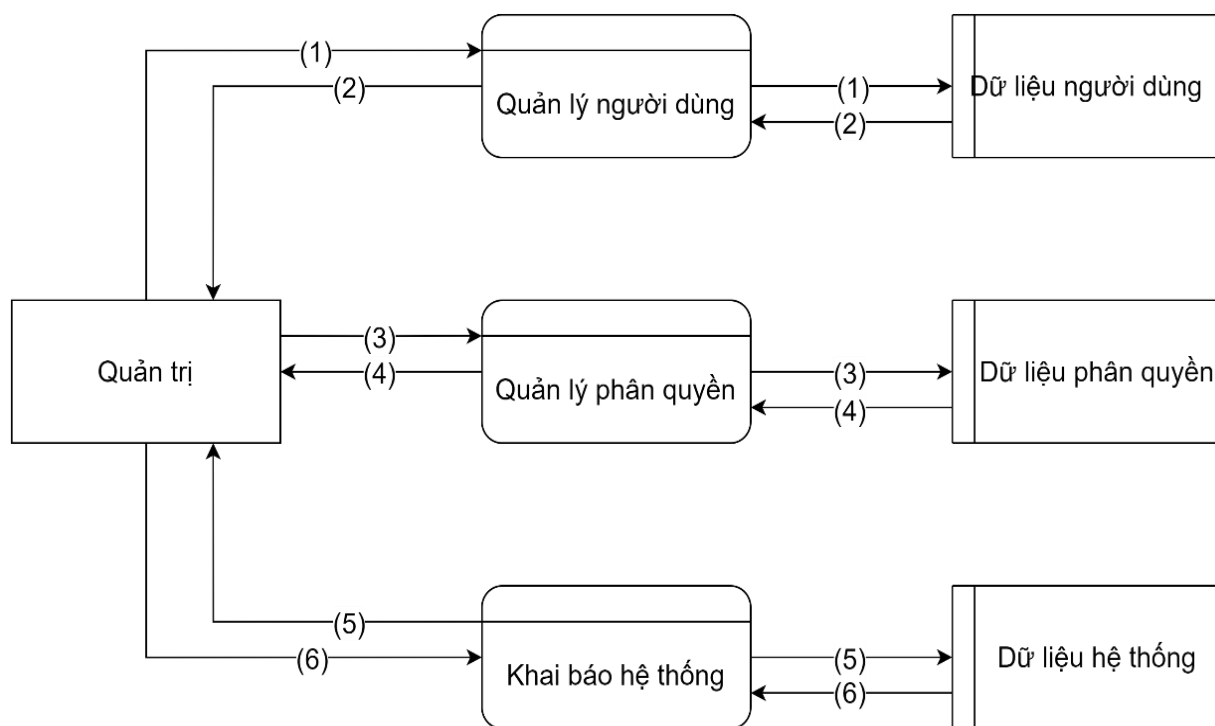
Hình 36: Sơ đồ mức đỉnh

- (1) Thông tin cập nhật người dùng và khai báo hệ thống.
- (2) Thông tin người dùng và khai báo hệ thống.
- (3) Thông tin cập nhật về nhân viên.
- (4) Thông tin về nhân viên.
- (5) Thông tin cập nhật về khách hàng
- (6) Thông tin về khách hàng.

- (7) Thông tin cập nhật về máy quét thẻ, thẻ.
- (8) Thông tin về máy quét thẻ, thẻ.
- (9) Thông tin cập nhật về camera, phương tiện.
- (10) Thông tin về camera, phương tiện.
- (11) Thông tin cập nhật về cổng, barrier.
- (12) Thông tin về cổng, barrier.
- (13) Thông tin cập nhật về nhân viên trong lịch sử ra vào.
- (14) Thông tin về nhân viên trong lịch sử ra vào.
- (15) Thông tin cập nhật về khách hàng trong lịch sử ra vào.
- (16) Thông tin về khách hàng trong lịch sử ra vào.
- (17) Thông tin cập nhật về máy quét thẻ, thẻ trong lịch sử ra vào.
- (18) Thông tin về máy quét thẻ, thẻ trong lịch sử ra vào.
- (19) Thông tin cập nhật về camera, phương tiện trong lịch sử ra vào.
- (20) Thông tin về camera, phương tiện trong lịch sử ra vào.
- (21) Thông tin cập nhật về cổng, barrier trong lịch sử ra vào.
- (22) Thông tin về cổng, barrier trong lịch sử ra vào.
- (23) Thông tin kho danh mục.
- (24) Thông tin lịch sử ra vào của khách hàng, phương tiện.
- (25) Thông tin yêu cầu các báo cáo, thông kê.
- (26) Thông tin về báo cáo, thống kê.

### 3.2.4. Sơ đồ mức dưới đỉnh

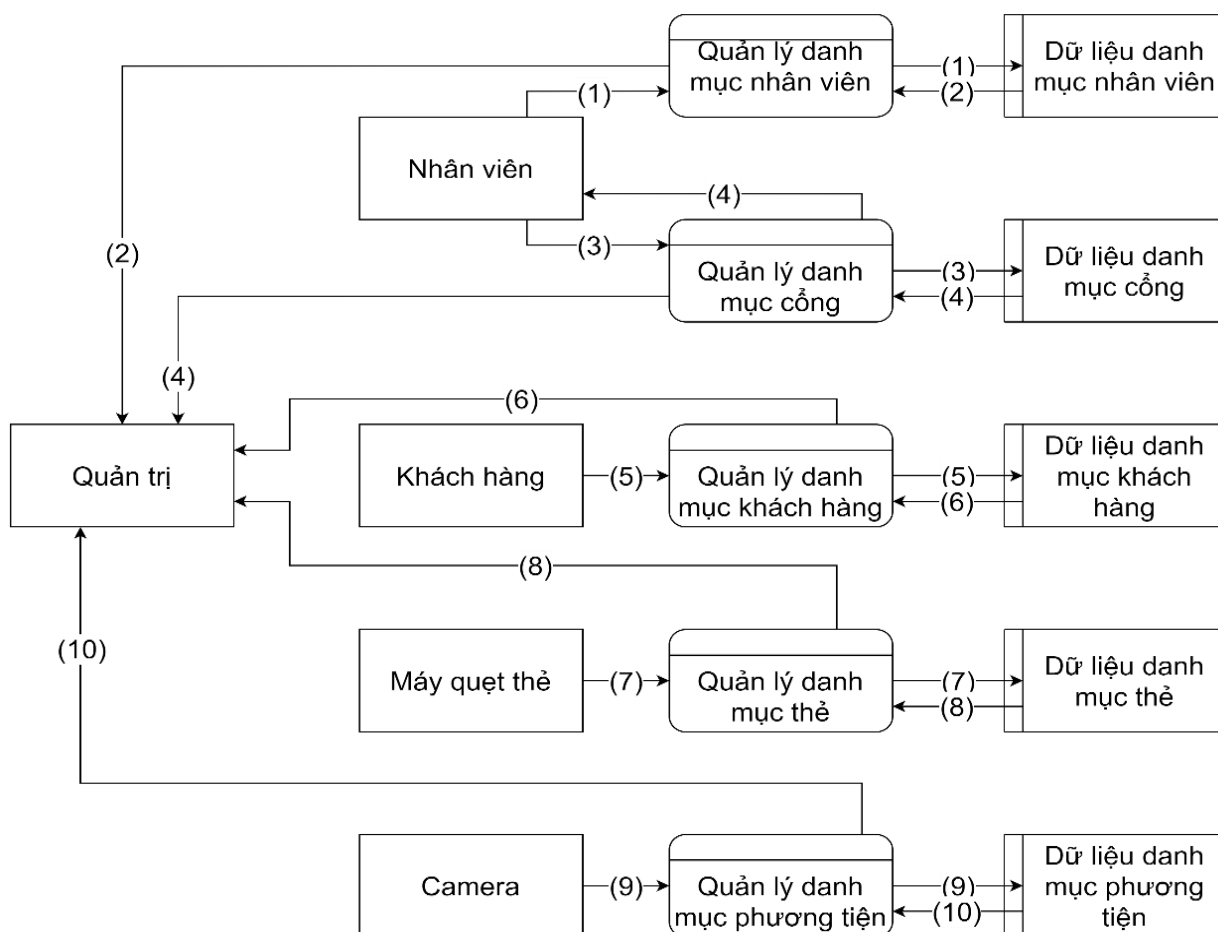
#### a) Quản lý hệ thống



Hình 37: Sơ đồ mức dưới đỉnh - Quản lý hệ thống

- (1) Thông tin cập nhật thông tin người dùng.
- (2) Thông tin người dùng.
- (3) Thông tin cập nhật phân quyền người dùng.
- (4) Thông tin phân quyền người dùng.
- (5) Thông tin yêu cầu khai báo hệ thống.
- (6) Thông tin dữ liệu hệ thống.

***b) Quản lý danh mục***



Hình 38: sơ đồ mức dưới đỉnh - Quản lý danh mục

- (1) Thông tin cập nhật nhân viên.
- (2) Thông tin danh mục nhân viên.
- (3) Thông tin cập nhật cổng.
- (4) Thông tin danh mục cổng.
- (5) Thông tin cập nhật khách hàng.
- (6) Thông tin danh mục khách hàng.
- (7) Thông tin cập nhật thẻ.

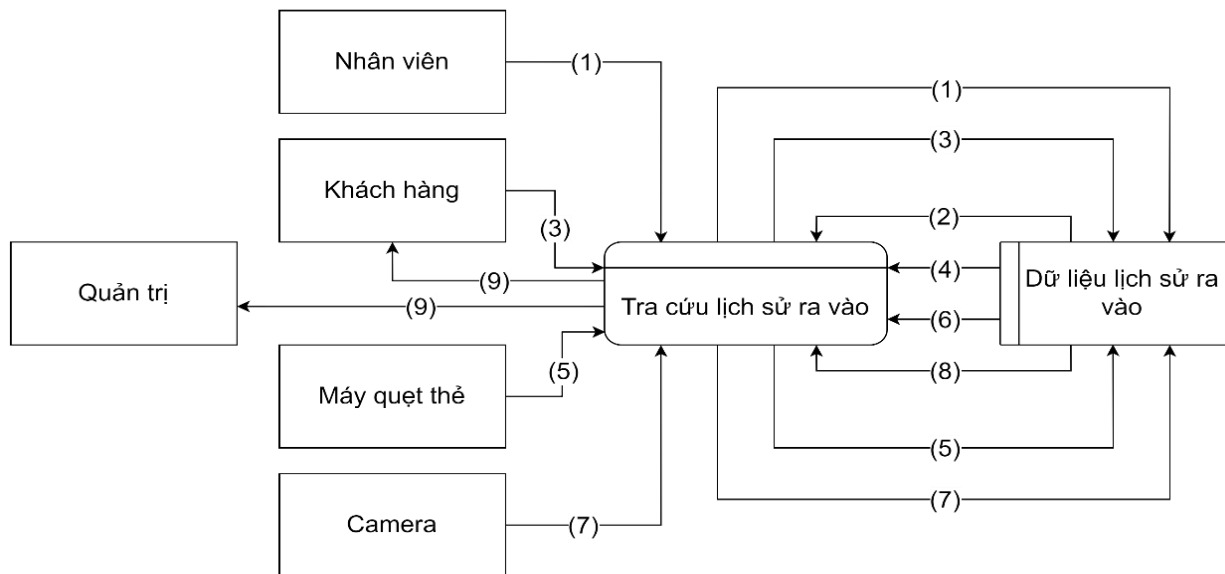


(8) Thông tin danh mục thẻ.

(9) Thông tin cập nhật phương tiện.

(10) Thông tin danh mục phương tiện.

**c) Quản lý lịch sử ra/vào**



Hình 39: Sơ đồ mức dưới đỉnh - Quản lý lịch sử ra/vào

(1) Thông tin cập nhật danh mục nhân viên.

(2) Lịch sử ra vào theo danh mục nhân viên.

(3) Thông tin cập nhật danh mục khách hàng.

(4) Lịch sử ra vào theo danh mục khách hàng.

(5) Thông tin cập nhật danh mục thẻ.

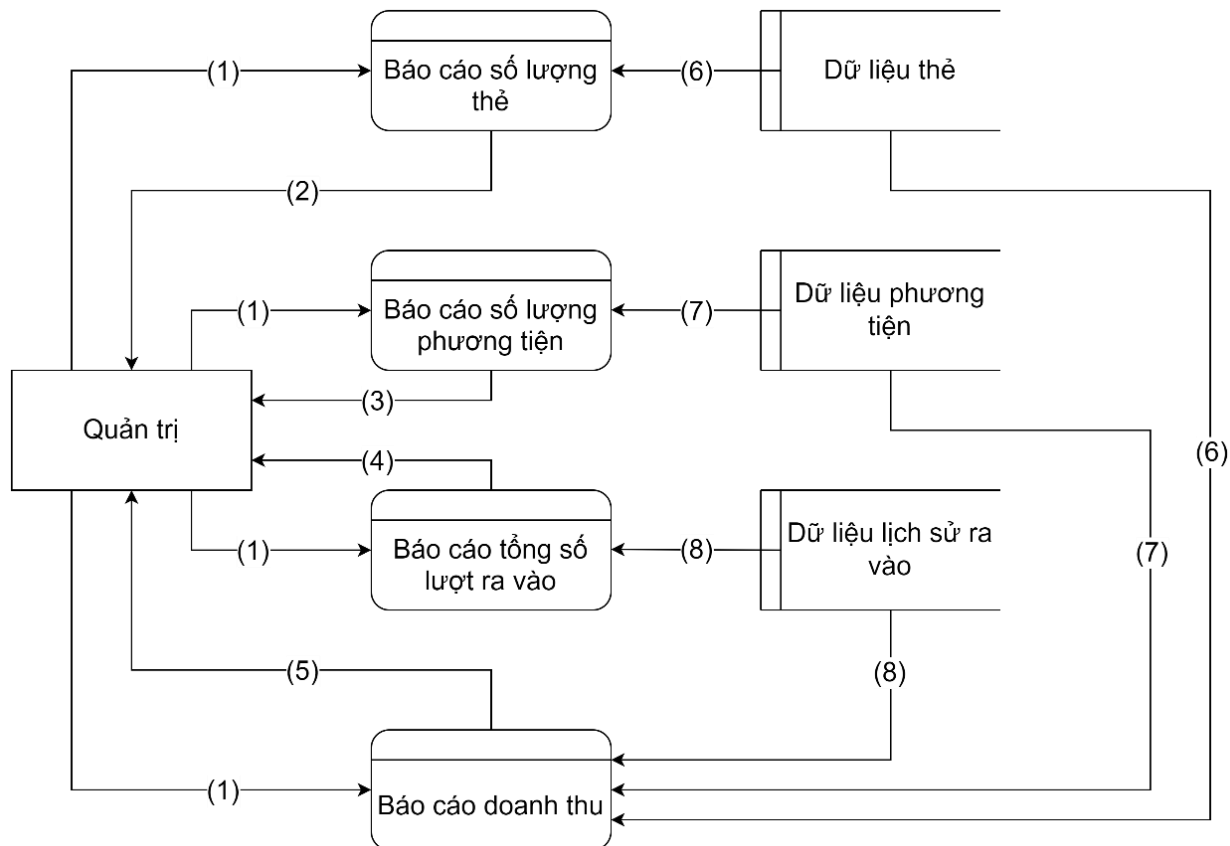
(6) Lịch sử ra vào theo danh mục thông tin thẻ.

(7) Thông tin cập nhật danh mục phương tiện.

(8) Lịch sử ra vào theo danh mục phương tiện.

(9) Lịch sử ra vào chi tiết.

***d) Báo cáo thống kê***



**Hình 40: Sơ đồ mức dưới đỉnh - Báo cáo thống kê**

- (1) Thông tin yêu cầu báo cáo thống kê.
- (2) Thông tin dữ liệu thẻ ra vào.
- (3) Thông tin dữ liệu số lượng phương tiện.
- (4) Thông tin dữ liệu tổng số lượt ra vào.
- (5) Thông tin dữ liệu doanh thu.
- (6) Dữ liệu thẻ.
- (7) Dữ liệu phương tiện.
- (8) Dữ liệu lịch sử ra vào.

### 3.2.5. Sơ đồ liên kết thực thể

#### a) Các thực thể và thuộc tính

Thực thể	Thuộc tính
Nhân viên	<u>Mã nhân viên</u> , Mã công, họ tên
Khách hàng	<u>Mã khách hàng</u> , Mã thẻ, Mã phương tiện, họ tên, số căn hộ
Cổng	<u>Mã thẻ</u> , số thẻ, loại thẻ, ngày tạo thẻ, giá tiền
Máy quét thẻ	<u>Mã máy quét thẻ</u> , tên máy, làn
Camera	<u>Mã camera</u> , tên camera, làn
Phương tiện	<u>Mã phương tiện</u> , loại phương tiện, biển số

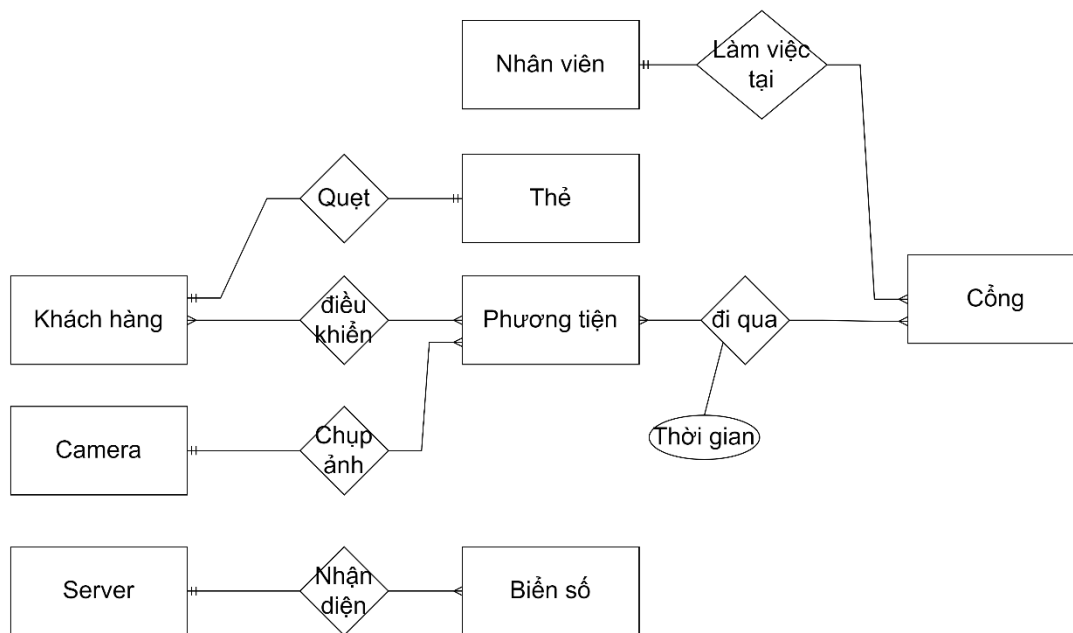
Bảng 2: Thực thể và thuộc tính

#### b) Môi liên kết giữa các thực thể

Mối liên kết	Loại liên kết	Thuộc tính
Nhân viên <b>làm việc</b> tại cổng	1:N	
Phương tiện <b>đi qua</b> cổng	N:N	Thời gian vào/ra
Khách hàng <b>điều khiển</b> phương tiện	N:N	
Khách hàng <b>quẹt</b> thẻ	1:1	
Camera <b>chụp ảnh</b> phương tiện	1:N	
Server <b>nhận diện</b> biển số	1:N	

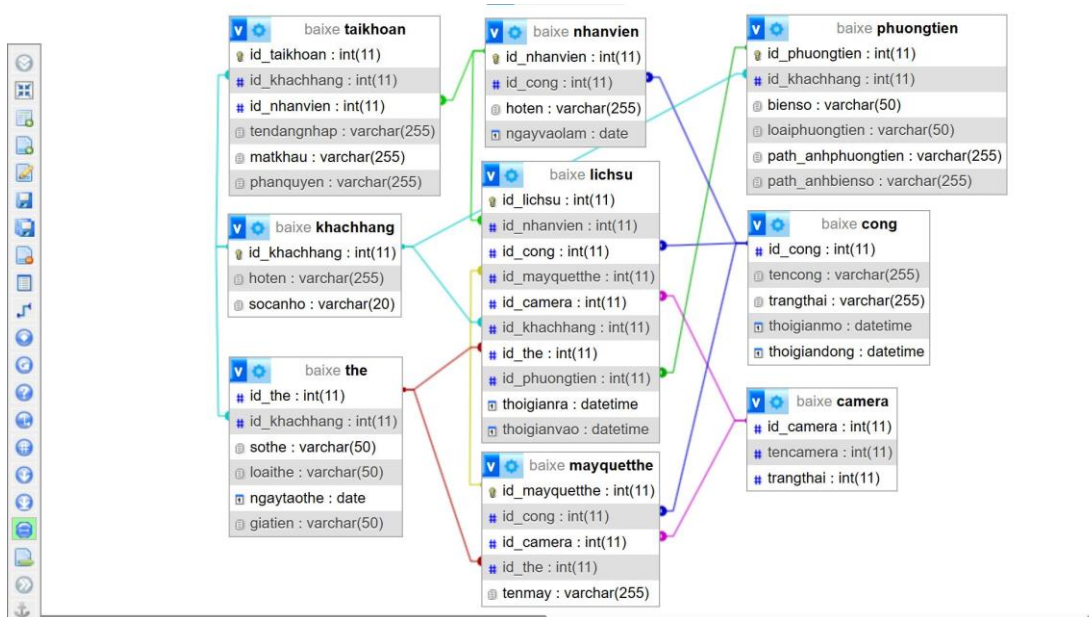
Bảng 3: Môi liên kết giữa các thực thể

### c) Sơ đồ liên kết thực thể



Hình 41: Sơ đồ liên kết thực thể

### 3.2.6. Cơ sở dữ liệu



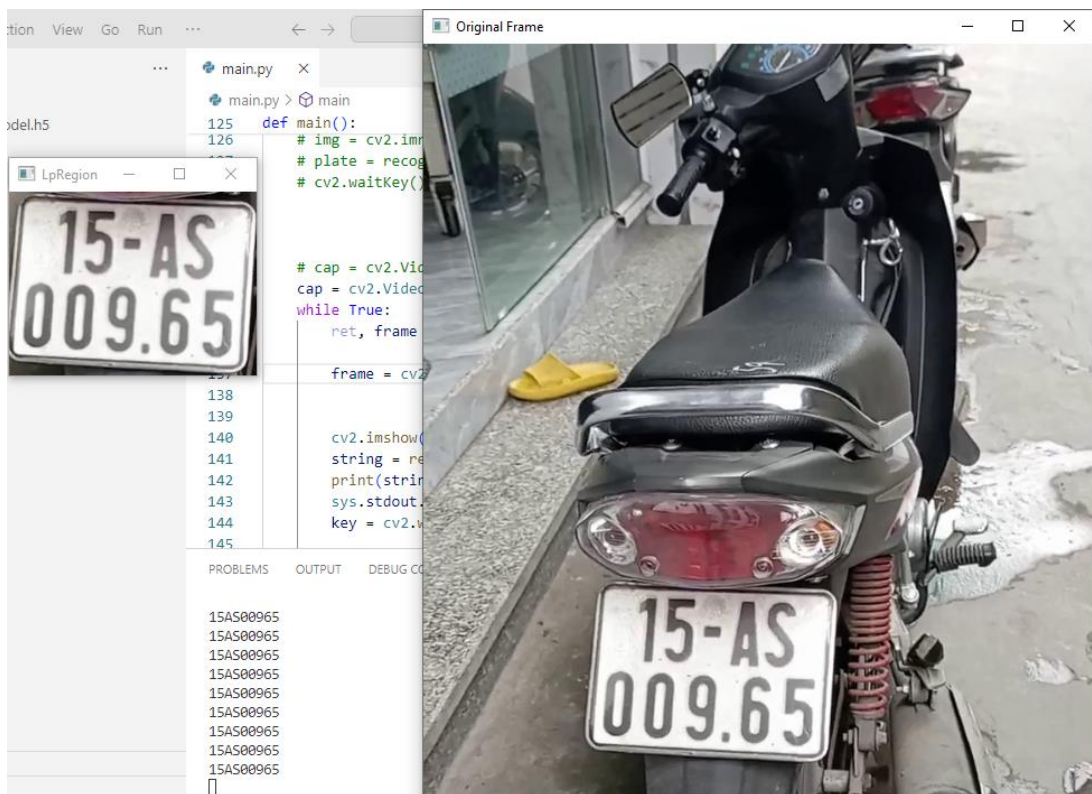
Hình 42: Cơ sở dữ liệu

## CHƯƠNG 4: CÁC KẾT QUẢ ĐẠT ĐƯỢC

### 4.1.Mô hình AI

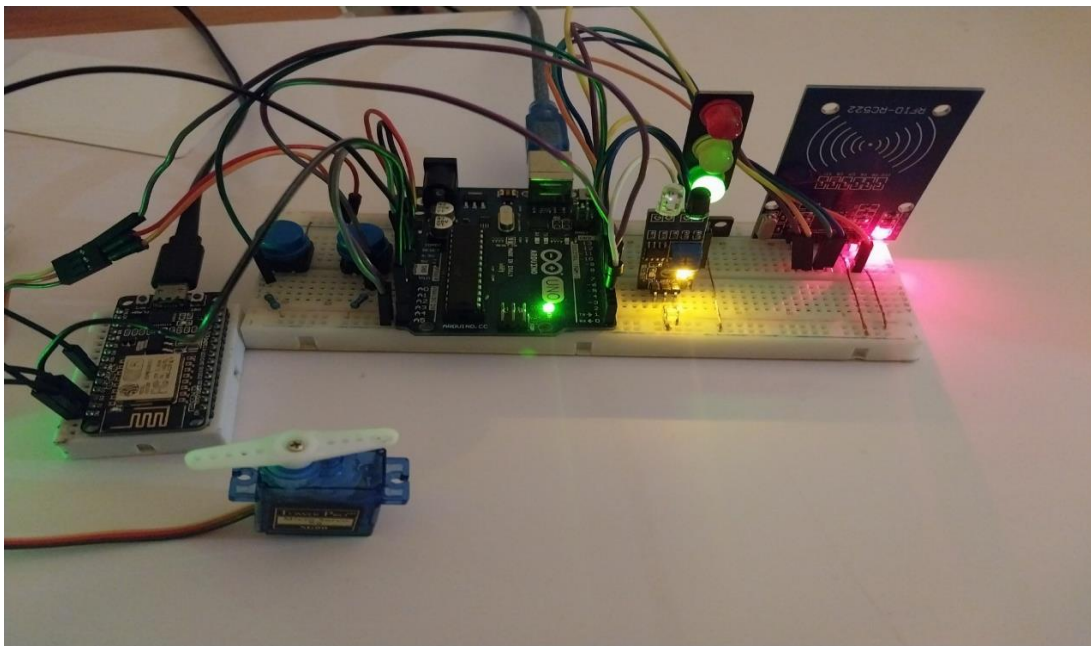
Sau khi triển khai mô hình trên 100 ảnh xe máy và 100 ảnh ô tô, em thu được kết quả sau:

Loại biển số	Số ảnh	Nhận diện đúng	Tỷ lệ	Ghi chú
Dài (1 dòng)	78	68	87.18%	Các ký tự không xếp theo thứ tự khi biển số bị nghiêng. (do các bounding box lệch hàng.
Vuông (2 dòng)	122	118	96.72%	Nhầm lẫn giữa số “2” và chữ cái “Z”
Tổng	200	186	91.95%	



Hình 43: Kết quả nhận dạng biển số xe trong video

## 4.2.Hệ thống nhúng



Hình 44: Kết quả demo hệ thống nhúng

```
--
54 case WStype_TEXT:
55   String message = String((char*)payload);
56
57   // Kiểm tra loại dữ liệu và gọi hàm tương ứng
58   if (message.indexOf("I") != -1) {
59     // Gọi hàm sendArduino_In và truyền vào dữ liệu từ server
60     sendArduino_In(message);
61   } else if (message.indexOf("O") != -1) {
62     // Gọi hàm sendArduino_Out và truyền vào dữ liệu từ server
63     sendArduino_Out(message);
64   }
65   Serial.println("Dữ liệu nhận được từ Server: " + message);
66   break;
67 }
68 }
```

Output Serial Monitor x

Message (Enter to send message to "NodeMCU 1.0 (ESP-12E Module)" on "COM4")

----- CUT HERE FOR EXCEPTION DECODER -----

\*\*\*\*\*Connecting to WiFi...

Connecting to WiFi...

Connecting to WiFi...

Connecting to WiFi...

Connecting to WiFi...

Connecting to WiFi...

Connecting to WiFi...

Connecting to WiFi...

Connected to WiFi

Connected to server

Dữ liệu nhận được từ Arduino: IN: 11 9C A9 23

Dữ liệu nhận được từ Server: OpenIn

Dữ liệu nhận được từ Server: CloseIn

### GIÁM SÁT PHƯƠNG TIỆN RA VÀO KHU CHUNG CƯ

Đây là cổng vào- INPUT

CloseIn

Dữ liệu UID:

IN: 11 9C A9 23

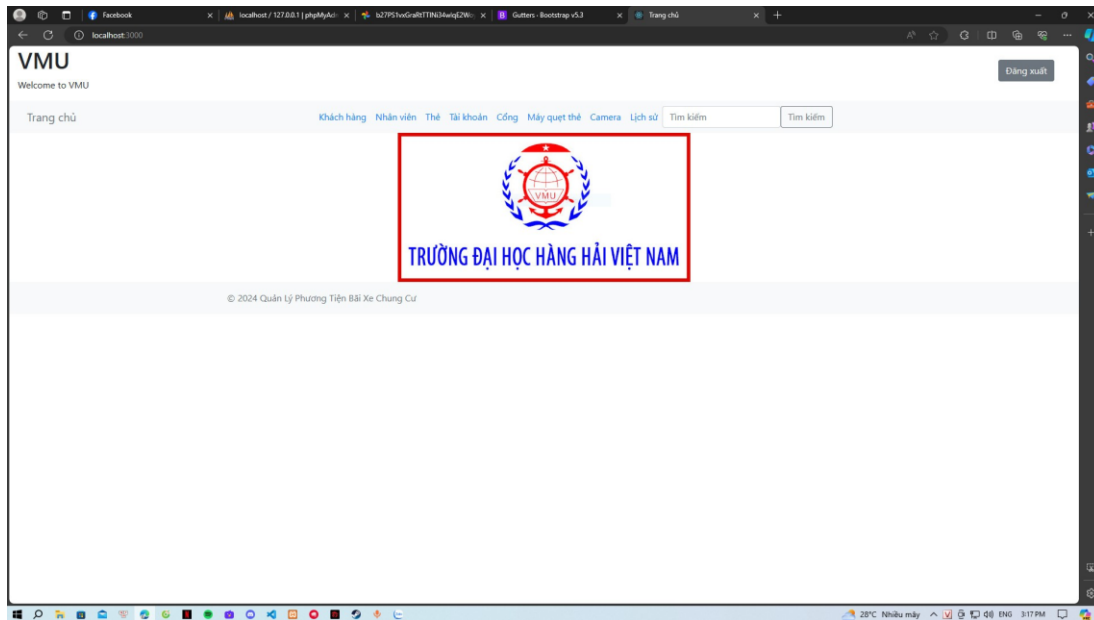
Đây là cổng ra- OUTPUT

Dữ liệu UID:

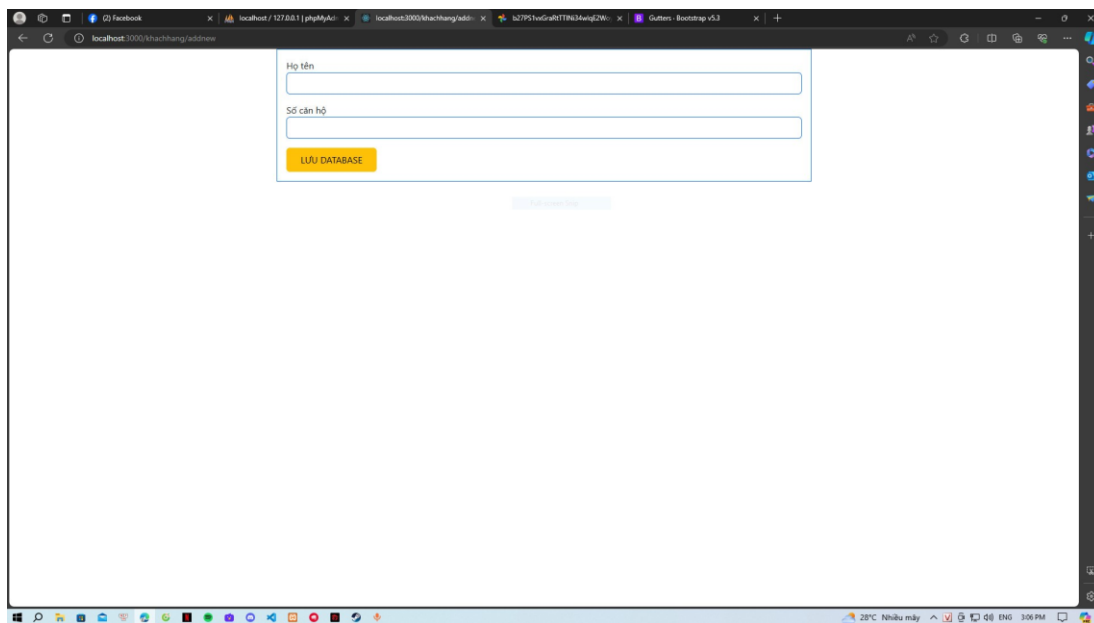
Dữ liệu UID sẽ hiển thị tại đây

Hình 45: Demo giao tiếp giữa Arduino-Esp-Server

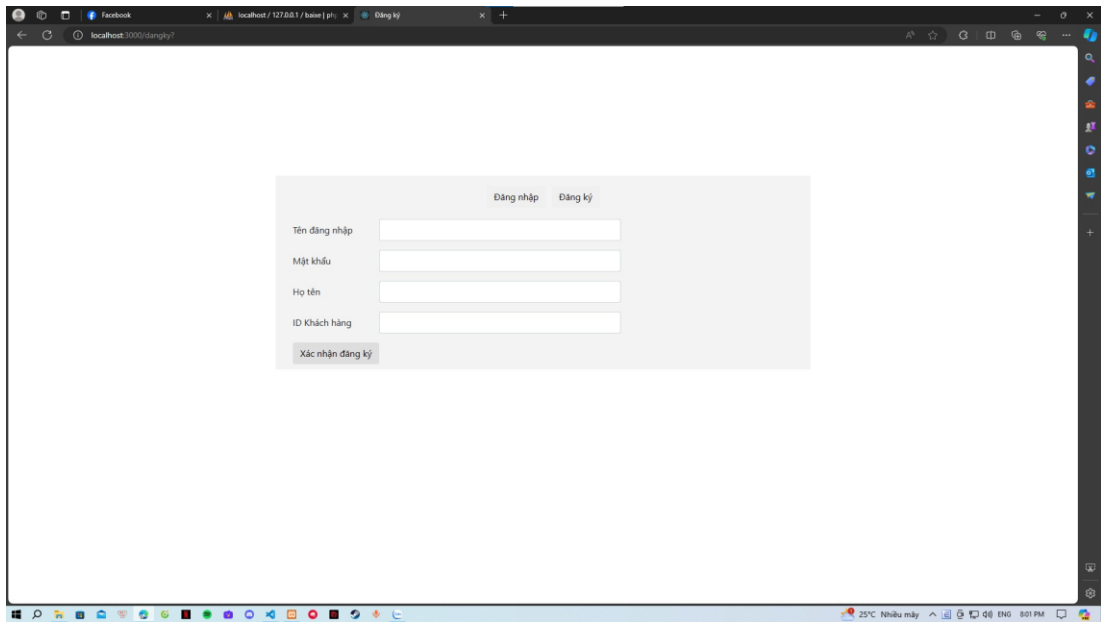
### 4.3.Hệ thống Server



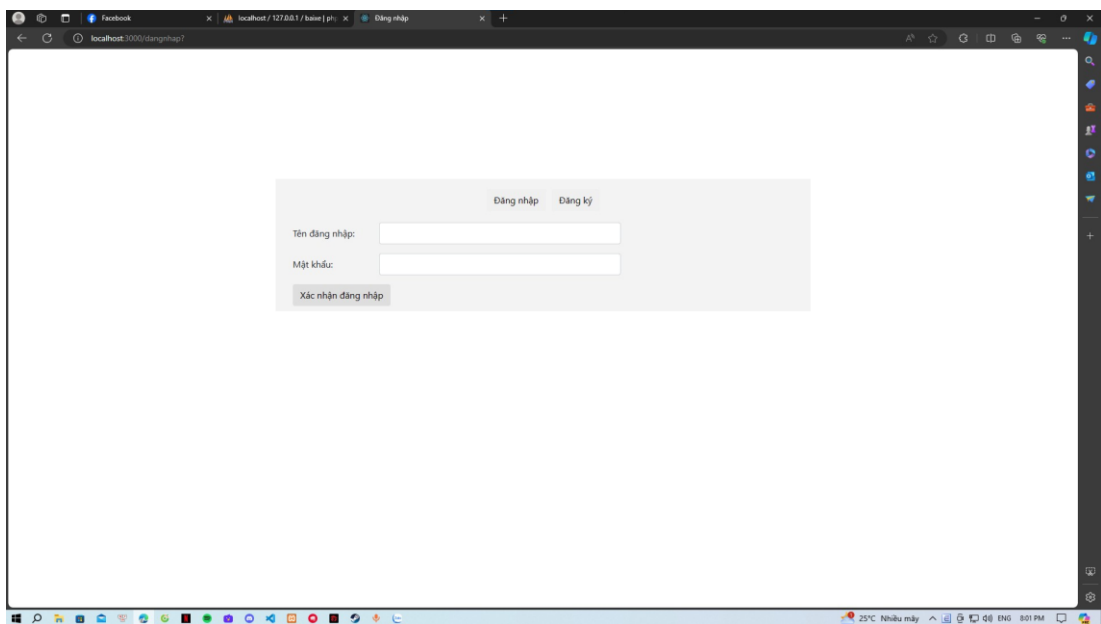
Hình 46: Demo giao diện trang chủ



Hình 47: Demo giao diện thêm khách hàng



Hình 48: Demo giao diện đăng ký



Hình 49: Demo giao diện đăng nhập



## KẾT LUẬN

Trong quá trình học tập và nghiên cứu triển khai hệ thống giám sát phương tiện ra vào trong bãi xe thông minh, nhóm chúng em đã đạt được nhiều kết quả tốt. Các thành viên nhóm đã có cái nhìn tổng quan và sự hiểu biết sâu hơn về cấu trúc của hệ thống cần xây dựng, được củng cố vững chắc thêm nhiều kiến thức liên quan đến đề tài như: lập trình arduino, lập trình web, trí tuệ nhân tạo.

Kết thúc thời gian thực tập, nhóm chúng em đã làm được các công việc sau:

- Xây dựng thành công mô hình AI để nhận dạng biển số xe.
- Xây dựng cơ bản cấu trúc của phần cứng.
- Hoàn thành phân tích thiết kế hệ thống, xây dựng các giao diện frontend.

Hướng phát triển: Trong tương lai, nhóm chúng em sẽ tiếp tục phát triển đề tài này, hướng tới bảo vệ đồ án. Để làm được điều đó, chúng em cần tìm hiểu và học tập để làm được các công việc sau:

- Xây dựng API để kết nối mô hình AI và hệ thống server.
- Lắp ráp hoàn thiện mô hình mô phỏng bãi đỗ xe, tích hợp màn hình oled hiển thị thông báo.
- Hoàn thiện hệ thống server.
- Kết nối các phân hệ thành một hệ thống nhất.

## TÀI LIỆU THAM KHẢO

Cộng đồng Arduino Việt Nam (<http://arduino.vn/>).

Longnv - Lập trình MVC trong NodeJS (<https://longnv.name.vn/lap-trinh-nodejs/lap-trinh-mvc-trong-nodejs#1-tao-project-voi-express>).

Văn Long - Nhận diện ký tự trên biển số xe Việt Nam ([viblo.asia/p/nhan-dien-ki-tu-tren-bien-so-xe-viet-nam-2oKLnXWZ4QO](https://viblo.asia/p/nhan-dien-ki-tu-tren-bien-so-xe-viet-nam-2oKLnXWZ4QO)).

Mi AI - Phát hiện biển số xe bằng pretrain model YOLO (<https://www.miai.vn/2019/11/12/nhan-dien-bien-so-xe-chuong-1-phat-hien-bien-so-xe-bang-pretrain-model-yolo/>).

## NHẬN XÉT TÓM TẮT CỦA GIẢNG VIÊN HƯỚNG DẪN

1. Tinh thần thái độ, sự cố gắng của sinh viên trong quá trình thực tập

.....

.....

.....

.....

.....

2. Đánh giá chất lượng Báo cáo thực tập (so với nội dung yêu cầu đã đề ra)

.....

.....

.....

.....

.....

.....

Đánh giá của GVHD

Hải phòng, ngày tháng 4 năm 2024

Giảng viên hướng dẫn

ThS. Phạm Trung Minh