HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

KHOA CÔNG NGHỆ THÔNG TIN 1

BÁO CÁO BÀI TẬP LỚN IOT VÀ ỨNG DỤNG

ĐỀ TÀI: HỆ THỐNG RA VÀO BÃI ĐỖ XE THÔNG MINH

Giảng viên hướng dẫn: Kim Ngọc Bách

Lớp: D22CNPM02

Nhóm: 05

Thành viên:

Nguyễn Hữu Lộc – B22DCCN508

Nguyễn Văn Học – B22DCCN352

Nguyễn Thanh Phong – B22DCCN616

HÀ NỘI, 2025

Mục lục

I. Cơ sở lý thuyết và công nghệ áp dụng

1. Kiến trúc 3 lớp của hệ thống IoT

Hệ thống IoT thường được chia thành 3 lớp chính:

1.1. Lớp cảm nhận (Perception Layer)

- Là lớp thấp nhất, chịu trách nhiệm thu thập dữ liệu từ môi trường thực tế.
- Bao gồm các thiết bị cảm biến và bộ điều khiển như:
    - Cảm biến siêu âm (đo khoảng cách, phát hiện xe vào/ra bãi)
    - Module ESP32-CAM (chụp ảnh biển số)
    - Đầu đọc RFID RC522 (nhận dạng thẻ từ của người gửi xe)
- Dữ liệu được thu thập ở lớp này là dữ liệu thô, ví dụ: ảnh, mã thẻ, khoảng cách, trạng thái xe có mặt hay không.

1.2. Lớp mạng (Network Layer)

- Đảm nhận truyền dữ liệu từ thiết bị IoT lên máy chủ hoặc cloud.
- Các phương thức kết nối có thể là:
    - WiFi (ESP32)
    - Ethernet
    - 4G/5G hoặc Zigbee/Bluetooth (nếu mở rộng)
- Các giao thức truyền dữ liệu thường dùng: MQTT, HTTP, CoAP, WebSocket.
- Lớp này đảm bảo dữ liệu đi qua mạng an toàn, nhanh, ổn định.

1.3. Lớp ứng dụng (Application Layer)

- Là lớp cao nhất, nơi người dùng và hệ thống quản trị tương tác.
- Các chức năng chính:
    - Quản lý thông tin xe ra/vào, nhận dạng biển số bằng AI
    - Quản lý tài khoản, lịch sử gửi xe, thanh toán
    - Hiển thị dữ liệu theo thời gian thực trên web
- Được xây dựng bằng:
    - Frontend: ReactJS
    - Backend: Spring Boot (API, xử lý logic)
    - Database: MySQL

2. Mô hình hoạt động của các thiết bị IoT

Quy trình hoạt động tổng quát:

1. Thu thập dữ liệu
    - ESP32-CAM chụp ảnh biển số.
    - RFID RC522 đọc mã thẻ từ của xe hoặc người dùng.
    - Cảm biến siêu âm đo khoảng cách để phát hiện xe ra/vào.
2. Truyền dữ liệu
    - ESP32 gửi dữ liệu (ảnh hoặc ID thẻ) qua WiFi đến MQTT Broker hoặc HTTP Server.
    - Dữ liệu có thể đi qua một IoT Gateway để lọc, nén hoặc xử lý sơ bộ.
3. Xử lý dữ liệu

- Server (Spring Boot) nhận dữ liệu → lưu vào MySQL Database.
- Module AI (Python hoặc TensorFlow) xử lý ảnh để nhận diện biển số.
- Dữ liệu được trả về web dashboard để hiển thị và thống kê.

3. Các giao thức IoT phổ biến

Giao thức	Đặc điểm chính	Ứng dụng
MQTT (Message Queuing Telemetry Transport)	Giao thức nhẹ, nhanh, tiết kiệm băng thông, dùng mô hình Publish/Subscribe.	Phù hợp cho truyền dữ liệu cảm biến nhỏ, real-time.
HTTP (HyperText Transfer Protocol)	Giao thức phổ biến, dễ dùng, dựa trên mô hình Request/Response.	Phù hợp cho REST API giữa ESP32 và server.
CoAP (Constrained Application Protocol)	Giao thức nhẹ dựa trên UDP, dùng cho thiết bị có tài nguyên hạn chế.	Dùng trong môi trường công nghiệp hoặc cảm biến năng lượng thấp.

4. Giao tiếp giữa ESP32/Arduino với máy tính hoặc Cloud

4.1. Giao tiếp với máy tính

- Thông qua Serial (UART) để debug hoặc truyền dữ liệu.
- Dùng phần mềm như Arduino IDE hoặc Thonny để nạp code, giám sát log.

4.2. Giao tiếp với Cloud

- ESP32 kết nối WiFi → gửi dữ liệu qua Internet:
  - MQTT: gửi đến MQTT Broker như Mosquitto hoặc HiveMQ Cloud.
  - HTTP: gửi đến API của Spring Boot Server.

5. Mô hình Edge Computing và IoT Gateway

5.1.Edge Computing

- Là mô hình xử lý dữ liệu tại biên mạng, gần nơi dữ liệu được sinh ra.
- Thay vì gửi toàn bộ dữ liệu lên cloud, Edge Device (ESP32) có thể xử lý sơ bộ:
    - Lọc nhiễu, nhận diện biển số trước khi gửi.
    - Giảm tải cho server, tiết kiệm băng thông.

5.2.IoT Gateway

- Là thiết bị trung gian giữa mạng cảm biến và cloud.
- Chức năng:
    - Thu thập dữ liệu từ các cảm biến (ESP32, RFID, cảm biến siêu âm).
    - Tổng hợp, mã hóa và gửi dữ liệu đến server (Spring Boot hoặc MQTT Broker).

Trong đề tài bãi xe, ESP32-CAM có thể đóng vai trò Edge Device/Gateway.

6. Lý thuyết về xử lý ảnh (AI nhận diện biển số)

Trong hệ thống này, quá trình nhận diện biển số xe được thực hiện bằng mô hình học sâu (Deep Learning) – cụ thể là YOLOv8 (You Only Look Once version 8), kết hợp với OCR (Optical Character Recognition) để đọc ký tự.

Quy trình nhận diện gồm 3 giai đoạn chính:

1. Phát hiện vùng biển số (License Plate Detection)

- Sử dụng mô hình YOLOv8 để phát hiện vị trí biển số trong ảnh chụp từ ESP32-CAM.

- YOLOv8 là mô hình Object Detection hiện đại, có tốc độ nhanh và độ chính xác cao.
- Mô hình được huấn luyện trên tập dữ liệu biển số Việt Nam (hoặc quốc tế) với nhãn là vùng biển số.
- Kết quả là một bounding box bao quanh vùng biển số trong ảnh.

2. Cắt và tiền xử lý biển số (Preprocessing)

- Sau khi phát hiện vùng biển số, mô hình sẽ crop (cắt) vùng đó ra khỏi ảnh gốc.
- Ảnh biển số được chuyển sang ảnh xám (Grayscale), lọc nhiễu (Gaussian Blur), và chuẩn hoá kích thước để phục vụ cho bước nhận dạng ký tự.

3. Nhận dạng ký tự (Character Recognition)

- Sử dụng thư viện Tesseract OCR hoặc một mô hình CNN nhỏ để đọc ký tự trên vùng biển số đã tách.
- Kết quả là chuỗi ký tự biển số (VD: "79A-12345").
- Chuỗi này sẽ được gửi về Spring Boot Server, lưu vào MySQL, và hiển thị trên giao diện ReactJS.

4. Ưu điểm của YOLOv8 so với phương pháp truyền thống:

Tiêu chí	**Phương pháp truyền thống (Contour + OCR)**	**YOLOv8**
Độ chính xác	Phụ thuộc ánh sáng và góc chụp	Ổn định, chính xác cao
Tốc độ xử lý	Nhanh với ảnh nhỏ	Nhanh hơn với GPU hoặc Edge AI

Khả năng mở rộng	Khó nhận diện nhiều loại biển	Dễ huấn luyện thêm dữ liệu
Ứng dụng thực tế	Giới hạn trong môi trường cố định	Phù hợp cho bãi xe thực tế ngoài trời

7. Công nghệ sử dụng

1. Các thiết bị IoT

1.1. Bộ xử lý: ESP32-CAM

ESP32-CAM là một vi điều khiển tích hợp camera và WiFi, phù hợp cho các ứng dụng IoT có yêu cầu thu thập và truyền hình ảnh.
Thiết bị này có khả năng:

- Chụp ảnh hoặc quay video trực tiếp.
- Kết nối WiFi để gửi dữ liệu hình ảnh về máy chủ hoặc cloud.
- Hỗ trợ xử lý nhẹ tại thiết bị (edge processing) như nén ảnh, phát hiện chuyển động.

Trong hệ thống quản lý bãi xe thông minh, ESP32-CAM được sử dụng để:

- Chụp ảnh biển số xe khi xe ra/vào.
- Gửi hình ảnh đến server qua giao thức HTTP hoặc MQTT.
- Có thể thực hiện nhận diện sơ bộ biển số (nếu tích hợp AI nhẹ).

1.2. Đầu đọc thẻ từ: RFID RC522

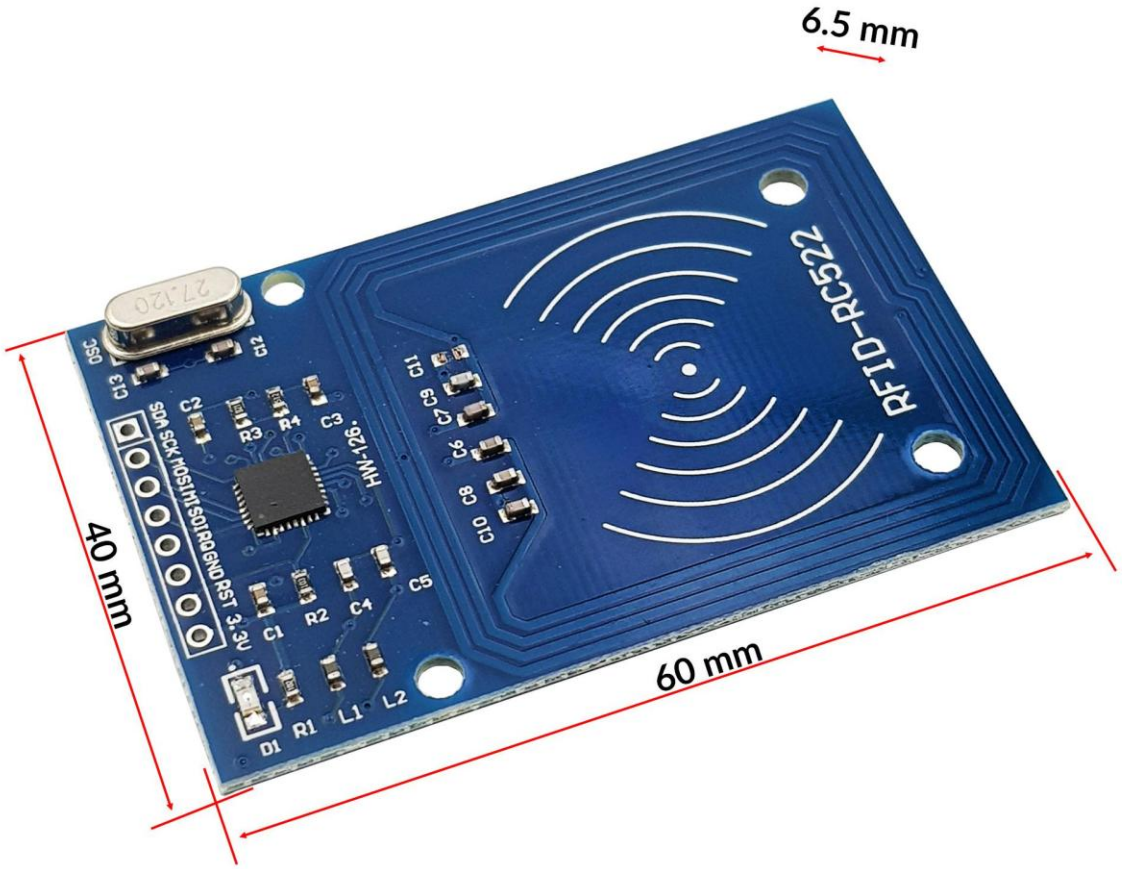
Module RFID RC522 là thiết bị đọc/ghi thẻ từ hoạt động ở tần số 13.56 MHz.
Thiết bị này giúp hệ thống nhận dạng người dùng thông qua thẻ từ gắn trên xe hoặc do người gửi xe mang theo.

Chức năng trong hệ thống:

- Mỗi người dùng được cấp một thẻ từ (RFID tag) có mã định danh duy nhất (UID).

- Khi xe đến cổng, đầu đọc RFID sẽ quét mã thẻ để xác định thông tin người gửi xe.

- Dữ liệu thẻ được gửi đến ESP32 và truyền về server để đối chiếu thông tin trong cơ sở dữ liệu.



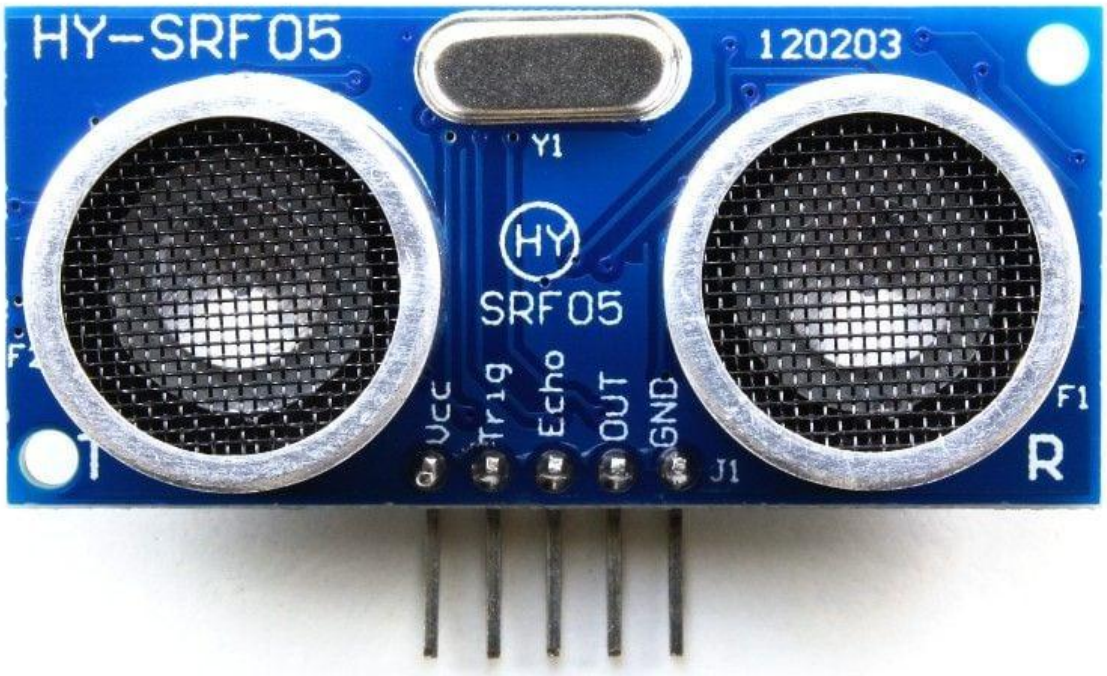
1.3. Cảm biến phụ: Cảm biến siêu âm (Ultrasonic Sensor) *(tùy chọn)*

Cảm biến siêu âm được dùng để phát hiện khoảng cách giữa cảm biến và vật thể phía trước.

Trong hệ thống bãi xe, cảm biến siêu âm có thể được gắn ở:

- Cổng ra/vào bãi để phát hiện xe đang tiến vào hoặc rời khỏi.
- Vị trí từng chỗ đỗ xe để xác định chỗ còn trống hay đã có xe.

Chức năng này giúp hệ thống hoạt động chính xác hơn, hỗ trợ thống kê số lượng xe còn chỗ trống theo thời gian thực.



1.4. Giao thức truyền thông: MQTT / HTTP / WebSocket

Các giao thức được lựa chọn nhằm đảm bảo dữ liệu giữa thiết bị IoT và server được truyền nhanh, ổn định, và bảo mật:

- MQTT (Message Queuing Telemetry Transport):
  Giao thức nhẹ, hoạt động theo mô hình Publish/Subscribe, rất phù hợp cho IoT.
  Dùng để truyền dữ liệu trạng thái xe, thẻ từ, và thông tin real-time (vd: xe vào/ra).

- HTTP (HyperText Transfer Protocol):
  Dùng khi gửi dữ liệu dạng JSON, đặc biệt là hình ảnh biển số từ ESP32-CAM về server.
  Tương thích tốt với REST API của Spring Boot.

- WebSocket:
  Được dùng trong phần web (ReactJS) để cập nhật trạng thái bãi xe theo thời gian thực, không cần tải lại trang.

1.5. Thẻ từ (RFID Tag)

Là thẻ có gắn chip lưu trữ mã định danh duy nhất.
Khi đưa gần đầu đọc RC522, tín hiệu được truyền không tiếp xúc qua sóng điện từ.
Mỗi người dùng hoặc phương tiện sẽ được cấp một thẻ riêng, giúp hệ thống xác định nhanh chóng người gửi xe mà không cần nhập tay.

1.6. Động cơ Servo SG90

Servo SG90 là loại động cơ servo mini, thường được dùng trong các dự án IoT và robot để điều khiển vị trí góc quay chính xác.
Servo có thể quay 0° – 180°, được điều khiển bằng xung PWM (Pulse Width Modulation) từ vi điều khiển (ESP32, Arduino...).

Chức năng trong hệ thống bãi xe thông minh:

- Servo SG90 được gắn với thanh chắn barie.
- Khi hệ thống xác định thẻ RFID hợp lệ hoặc biển số được nhận diện chính xác, ESP32 gửi tín hiệu PWM đến Servo để mở barie.

- Sau khi xe đi qua (cảm biến siêu âm phát hiện xe đã rời khỏi vùng), Servo tự động quay về vị trí đóng.

2. Trí tuệ nhân tạo (AI)

Hệ thống AI sử dụng mô hình YOLLO v8 để nhận diện biển số xe từ ảnh do ESP32-CAM gửi về.

Công nghệ sử dụng:

- Ngôn ngữ: Python
- Thư viện:
    - Ultralytics YOLOv8: Phát hiện vùng chứa biển số xe.

    - OpenCV: Tiền xử lý và cắt ảnh vùng biển số.
    - Tesseract OCR: Nhận dạng ký tự từ vùng biển số sau khi cắt.
- Kết quả xử lý:
    - Trả về chuỗi ký tự biển số.
    - Gửi dữ liệu qua API đến Spring Boot Server để lưu vào MySQL Database.
    - Hiển thị thông tin lên giao diện ReactJS theo thời gian thực.

Luồng xử lý dữ liệu AI:

1. ESP32-CAM chụp ảnh → gửi về Server (qua HTTP hoặc MQTT).
2. Server chuyển ảnh đến module AI (Python).
3. AI chạy YOLLO v8 để phát hiện biển số, OCR để đọc ký tự.
4. Kết quả được lưu vào MySQL và hiển thị trên ReactJS.

3. Hệ thống Web

Hệ thống Web đóng vai trò là giao diện cho người quản lý và nhân viên vận hành bãi xe.
Các chức năng chính:

- Theo dõi xe ra/vào, hiển thị hình ảnh và biển số nhận dạng.
- Quản lý thông tin người dùng, thẻ từ, lịch sử gửi xe.
- Thống kê, báo cáo tình trạng bãi xe.

Web gồm 3 thành phần chính:

Backend: Spring Boot

- Là framework Java mạnh mẽ, dùng để xây dựng RESTful API.
- Chịu trách nhiệm:
    - Nhận và xử lý dữ liệu từ thiết bị IoT.
    - Tương tác với cơ sở dữ liệu MySQL.
    - Cung cấp API cho frontend ReactJS.
    - Gọi đến module AI để xử lý ảnh khi cần.

Spring Boot giúp phát triển nhanh, dễ mở rộng và bảo mật tốt (Spring Security).

Database: MySQL

- Lưu trữ toàn bộ dữ liệu của hệ thống:
    - Thông tin người dùng, xe, thẻ từ.
    - Lịch sử ra/vào bãi xe.
    - Kết quả nhận diện biển số và ảnh chụp.
- MySQL được chọn vì dễ triển khai, hỗ trợ tốt cho hệ thống web có quy mô vừa và nhỏ.

Frontend: ReactJS

- Là thư viện JavaScript hiện đại, chuyên xây dựng giao diện người dùng tương tác cao.
- Trong hệ thống bãi xe, ReactJS được dùng để:
    - Hiển thị dữ liệu real-time (biển số, ảnh xe, tình trạng bãi).
    - Cung cấp giao diện quản lý trực quan, thân thiện.
    - Giao tiếp với API Spring Boot qua HTTP hoặc WebSocket.

ReactJS giúp tách biệt phần hiển thị với phần xử lý backend, đảm bảo hệ thống dễ mở rộng và bảo trì

II. Tài liệu đặc tả yêu cầu hệ thống

1. Giới thiệu

1.1. Mục tiêu hệ thống

Mục tiêu chính của hệ thống là tự động hóa hoàn toàn quy trình quản lý xe ra vào bãi, kết hợp công nghệ IoT trong việc thu thập và truyền dữ liệu từ thiết bị, cùng AI trong việc xử lý hình ảnh để nhận diện biển số xe.

Cụ thể, hệ thống hướng tới các mục tiêu sau:

- Xây dựng mô hình bãi xe thông minh có khả năng nhận dạng biển số xe tự động bằng camera kết hợp AI.
- Tích hợp thẻ từ RFID để xác thực người dùng và phương tiện ra/vào.
- Tự động điều khiển barrier mở hoặc đóng dựa trên kết quả xác thực.
- Lưu trữ thông tin phương tiện, thời gian gửi, hình ảnh và log vào/ra trong cơ sở dữ liệu tập trung.
- Cung cấp giao diện web trực quan cho phép quản lý, giám sát và thống kê hoạt động của bãi xe theo thời gian thực.
- Đảm bảo hệ thống hoạt động ổn định, chính xác, dễ mở rộng và dễ sử dụng, có khả năng tích hợp thêm các tính năng nâng cao trong tương lai như thanh toán tự động hoặc cảnh báo thông minh.

Mục tiêu tổng quát của đề tài là ứng dụng công nghệ IoT và AI để tạo ra một giải pháp quản lý bãi xe thông minh, hiệu quả và có tính thực tiễn cao, phù hợp với nhu cầu thực tế hiện nay tại các khu vực đông phương tiện.

1.2. Phạm vi triển khai

Hệ thống sẽ tập trung vào việc thực hiện các chức năng cốt lõi sau:

1. Xác thực kép: Quá trình xác thực xe hợp lệ được thực hiện dựa trên sự kết hợp của hai yếu tố: mã thẻ từ (RFID UID) và biển số xe (nhận dạng qua ảnh).
2. Thu thập và gửi dữ liệu: Trạm ESP32-CAM tại barrier chịu trách nhiệm chụp ảnh, đọc mã thẻ và gửi đồng thời cả hai dữ liệu này đến máy chủ trung tâm để xử lý.
3. Tự động điều khiển Barrier: Tự động nâng thanh chắn khi xác thực thành công và tự động hạ thanh chắn sau khi xe đã đi qua an toàn.
4. Ghi nhận nhật ký: Hệ thống sẽ tự động ghi lại thông tin của mỗi lượt xe ra/vào (biển số, mã thẻ, thời gian) vào cơ sở dữ liệu.
5. Quản lý cơ bản: Cung cấp một giao diện web đơn giản cho phép Người quản lý có thể thêm, xóa, và chỉnh sửa thông tin các phương tiện được cấp phép trong hệ thống.

Không bao gồm (Out-of-Scope)

Để đảm bảo dự án được hoàn thành đúng thời hạn và mục tiêu, các chức năng sau sẽ không được phát triển:

1. Tính phí và thanh toán: Chức năng tính toán chi phí đỗ xe và tích hợp các cổng thanh toán nằm ngoài phạm vi dự án.
2. Quản lý chỗ trống: Hệ thống không theo dõi hay đếm số lượng vị trí còn trống trong bãi đỗ xe.
3. Phân quyền người dùng phức tạp: Chỉ có một cấp độ quản lý duy nhất, không có các vai trò như nhân viên kế toán, nhân viên giám sát...

1.3. Tiêu chí thành công

Để đánh giá mức độ thành công và hiệu quả hoạt động của hệ thống, nhóm thực hiện đề ra các tiêu chí và chỉ số đo lường cụ thể như sau:

Tiêu chí	Mục tiêu đạt được	Ý nghĩa
Độ chính xác nhận diện biển số	>= 95 %	Đảm bảo AI nhận dạng đúng biển số trong đa số trường hợp
Thời gian xử lý / lượt xe	<= 2 giây	Thời gian xe đến cổng đến khi barrier mở
Tỷ lệ hoạt động ổn định	>= 98%	Hệ thống hoạt động liên tục, không gián đoạn

Khả năng mở rộng hệ thống	Mở rộng >= 3 cổng	Có thể dễ dàng thêm thiết bị và điểm kiểm soát mới
Chi phí triển khai	Thấp – sử dụng linh kiện phổ thông	Đảm bảo khả năng ứng dụng thực tế với chi phí hợp lý
Bảo mật dữ liệu	Có xác thực JWT, HTTPS	Đảm bảo an toàn dữ liệu người dùng và thiết bị IoT

1.4. Kết quả mong đợi

Khi hoàn thành, đề tài dự kiến đạt được các kết quả sau:

- Xây dựng thành công mô hình bãi xe thông minh tích hợp đầy đủ các thành phần IoT và AI hoạt động đồng bộ.
- Hệ thống có khả năng tự động nhận diện biển số xe và xác thực người dùng bằng thẻ RFID một cách chính xác và nhanh chóng.
- Barrier hoạt động tự động theo kết quả xác thực, giúp giảm tối đa sự can thiệp của con người.
- Giao diện quản lý web cho phép theo dõi, thống kê và xuất báo cáo tình hình hoạt động của bãi xe theo thời gian thực.
- Dữ liệu được lưu trữ tập trung, có thể truy xuất, phân tích và mở rộng dễ dàng trong các hệ thống lớn hơn.
- Hệ thống đạt được hiệu năng, độ chính xác và độ tin cậy cao, chứng minh tính khả thi và hiệu quả của mô hình IoT – AI trong việc tự động hóa quản lý bãi xe.

Ngoài ra, đề tài còn giúp sinh viên vận dụng tổng hợp các kiến thức về IoT, AI, lập trình web, cơ sở dữ liệu và bảo mật để xây dựng một sản phẩm hoàn chỉnh, có tính ứng dụng thực tế và khả năng phát triển thương mại trong tương lai.

2. Mô tả tổng quan

2.1. Bối cảnh hệ thống

Đây là một hệ thống tự động điều khiển ra vào tự động tại các bãi đỗ xe. Sản phẩm được xây dựng với mục đích học tập. Hệ thống mô phỏng một giải pháp quản lý bãi xe tự động, tích hợp các công nghệ hiện đại để thay thế cho việc kiểm soát thủ công.

2.2. Tính năng của hệ thống

Hệ thống cung cấp các tính năng chính sau:

- Nhận dạng và xác thực vào/ra:
    - Tự động chụp ảnh và nhận dạng biển số xe tại cổng vào và cổng ra bằng mô hình AI.
    - Cho phép xác thực bằng cách quét thẻ từ RFID đã được đăng ký.
- Điều khiển ra vào tự động:
    - Tự động mở thanh chắn khi biển số xe và thẻ từ được xác thực hợp lệ.
- Quản lý và giám sát:
    - Cung cấp một giao diện đơn giản bằng web để người quản lý đăng ký biển số xe, đăng ký thẻ từ mới.
    - Ghi lại và hiển thị lịch sử các lượt xe ra vào hệ thống.
    - Quản lý Thiết bị: Cho phép Người Quản lý (Admin) đăng ký các bộ điều khiển cổng mới (gồm ESP32-CAM, RFID reader,...) vào hệ thống. Mỗi thiết bị sẽ được gán một mã định danh và vị trí

2.3. Các lớp người dùng

Hệ thống có hai lớp người dùng chính:

- Người Quản lý (Admin):
    - Đặc điểm: Là người vận hành hệ thống. Có kiến thức cơ bản về cách hoạt động của hệ thống. Có thể là bảo vệ hoặc quản trị viên hệ thống.
    - Quyền hạn: Có toàn quyền truy cập giao diện quản lý để thêm/xóa/sửa thông tin xe, thẻ từ và xem lại lịch sử ra vào.
- Người dùng cuối (Driver):
    - Đặc điểm: Là người lái xe ô tô/xe máy.
    - Tương tác: Tương tác với hệ thống một cách gián tiếp bằng cách đưa xe đến vị trí camera hoặc quét thẻ từ lên đầu đọc.

2.4. Môi trường vận hành

- Phần cứng

    - Module ESP32-CAM, module đọc thẻ RFID RC522, động cơ Servo SG90, cảm biến siêu âm HC-SR04..
    - Máy tính trung tâm có kết nối internet
- Phần mềm
    - Trên Máy chủ: Chạy hệ điều hành Windows, ngôn ngữ lập trình python, thư viện xử lý ảnh OpenCV, Thư viện mô hình AI ultralytics
    - Cơ sở dữ liệu MySQL
- Mạng
    - Hệ thống yêu cầu một mạng Wifi nội bộ ổn định để ESP32-CAM có thể gửi hình ảnh về máy chủ xử lý.
- Môi trường vật lý
    - Hệ thống hoạt động trong môi trường phòng thí nghiệm
    - Nhiệt độ 0 - 50°C
    - Ánh sáng cường độ ổn định

2.5. Các ràng buộc về thiết kế và triển khai

- Phạm vi: Là sản phẩm demo cho môn học, do đó các tính năng phức tạp như thanh toán, quản lý chỗ trống sẽ được bỏ qua.
- Hiệu năng: Độ chính xác của AI và tốc độ xử lý phụ thuộc vào cấu hình máy chủ và chất lượng hình ảnh từ ESP32-CAM. Hệ thống được tối ưu cho điều kiện demo (ánh sáng tốt, góc chụp thẳng).
- Chi phí: Ưu tiên sử dụng các linh kiện điện tử giá rẻ, phổ biến.
- Bảo mật: Các yếu tố bảo mật mạng và dữ liệu chỉ ở mức cơ bản, không phù hợp cho môi trường triển khai thực tế.

2.6. Các giả định và phụ thuộc

Giả định:

- ESP32-CAM có thể chụp được ảnh đủ rõ nét để AI nhận dạng chính xác trong môi trường demo có kiểm soát.
- Mạng Wi-Fi nội bộ ổn định, độ trễ thấp.
- Biển số xe được sử dụng trong demo là biển số chuẩn, sạch sẽ, không bị che khuất.

Phụ thuộc:

- Hoạt động của hệ thống phụ thuộc vào các thư viện phần mềm mã nguồn mở (OpenCV, TensorFlow, Pytesseract, v.v.).
- Dự án phụ thuộc vào sự hoạt động ổn định của các linh kiện phần cứng (ESP32-CAM, đầu đọc RFID, máy chủ).

3. Yêu cầu chức năng

3.1. Các tác nhân

Trong hệ thống "Quản lý bãi xe thông minh ứng dụng IoT và AI nhận diện biển số xe", các tác nhân chính tham gia bao gồm:

- Người dùng (Chủ xe):
  Là người gửi xe tại bãi. Họ thực hiện thao tác quẹt thẻ RFID khi ra hoặc vào bãi xe. Hệ thống tự động nhận diện biển số xe và xác thực phương tiện của họ.

- Nhân viên bảo vệ:
  Là người vận hành và giám sát hoạt động của hệ thống. Nhân viên theo dõi xe ra/vào, xử lý các trường hợp bất thường, và hỗ trợ khách hàng khi cần thiết thông qua giao diện quản lý.

- Người quản lý bãi xe (Quản trị viên):
  Là người có quyền cao nhất trong hệ thống. Quản lý thông tin người dùng, phương tiện, thẻ RFID, và giám sát toàn bộ hoạt động của bãi xe. Ngoài ra, người quản lý còn chịu trách nhiệm thống kê, báo cáo và cấu hình hệ thống.

3.2. Các chức năng chính

3.2.1. Nhận diện và xác thực phương thiện vào/ra

- Mô tả:
  - Hệ thống có chức năng tự động nhận diện và xác thực phương tiện khi xe ra hoặc vào bãi. Khi xe đến gần cổng, camera sẽ tự động chụp ảnh biển số. Ảnh được gửi đến mô-đun AI xử lý ảnh (YOLO + OCR) để nhận dạng ký tự biển số. Đồng thời, người dùng quẹt thẻ RFID tại đầu đọc để xác minh quyền ra/vào.

- Chức năng:
  - Máy chủ trung tâm đối chiếu biển số nhận dạng được với thông tin xe và mã thẻ RFID trong cơ sở dữ liệu.
  - Nếu dữ liệu trùng khớp, hệ thống gửi tín hiệu đến ESP32 để mở barrier tự động.
  - Hệ thống lưu thông tin về biển số xe, mã thẻ, thời gian vào/ra, hình ảnh, trạng thái xe

- Nếu không hợp lệ, hệ thống từ chối truy cập và hiển thị cảnh báo trên giao diện giám sát cho nhân viên bảo vệ.

3.2.2. Quản lý thiết bị, phương tiện và thẻ từ

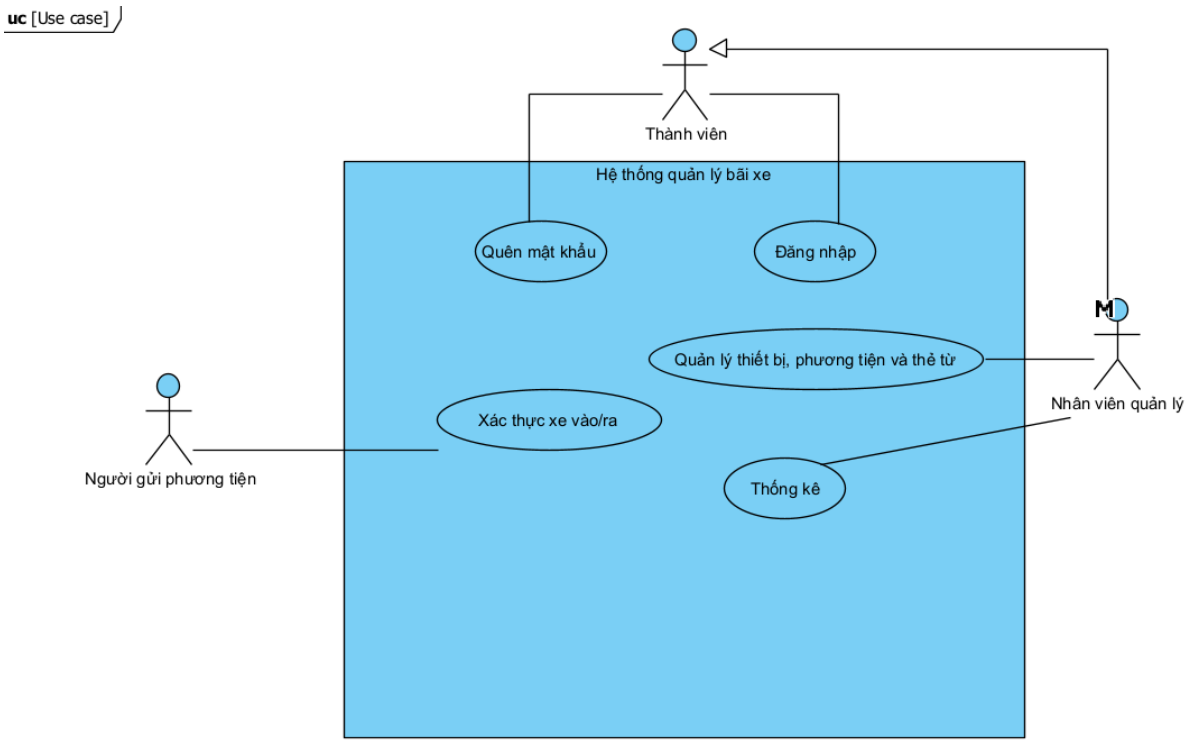
- Mô tả:
  - Hệ thống cho phép người quản lý thực hiện các thao tác quản trị dữ liệu người dùng, phương tiện và thẻ RFID trên giao diện quản lý.
- Chức năng:
  - Quản lý thiết bị: thêm, sửa, xóa thiết bị khi cần thay đổi
  - Quản lý phương tiện: đăng ký xe mới, cập nhật hoặc xóa thông tin phương tiện.
  - Quản lý thẻ RFID: đăng ký, kích hoạt, vô hiệu hóa hoặc thay đổi thẻ.
  - Đảm bảo tính đồng bộ giữa dữ liệu thiết bị – phương tiện – thẻ từ trong hệ thống.

3.2.4. Thống kê

- **Mô tả:**
  - Hệ thống có chức năng **thống kê, tổng hợp và hiển thị dữ liệu** để phục vụ việc quản lý, phân tích và đánh giá hoạt động của bãi xe.
- **Chức năng:**
  - Tổng hợp dữ liệu từ các bản ghi gửi xe để tạo báo cáo theo ngày, tuần, tháng.
  - Thống kê số lượng xe vào – ra, thời gian gửi xe trung bình, tỷ lệ sử dụng bãi.
  - Hiển thị biểu đồ doanh thu, lưu lượng xe, cảnh báo sự cố (nếu có).

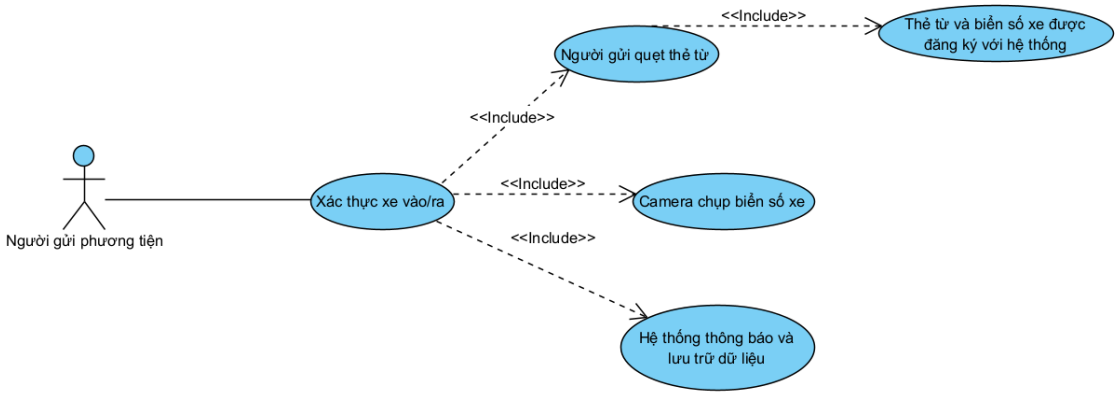
3.3. Biểu đồ use case

3.3.1. Biểu đồ use case tổng quan

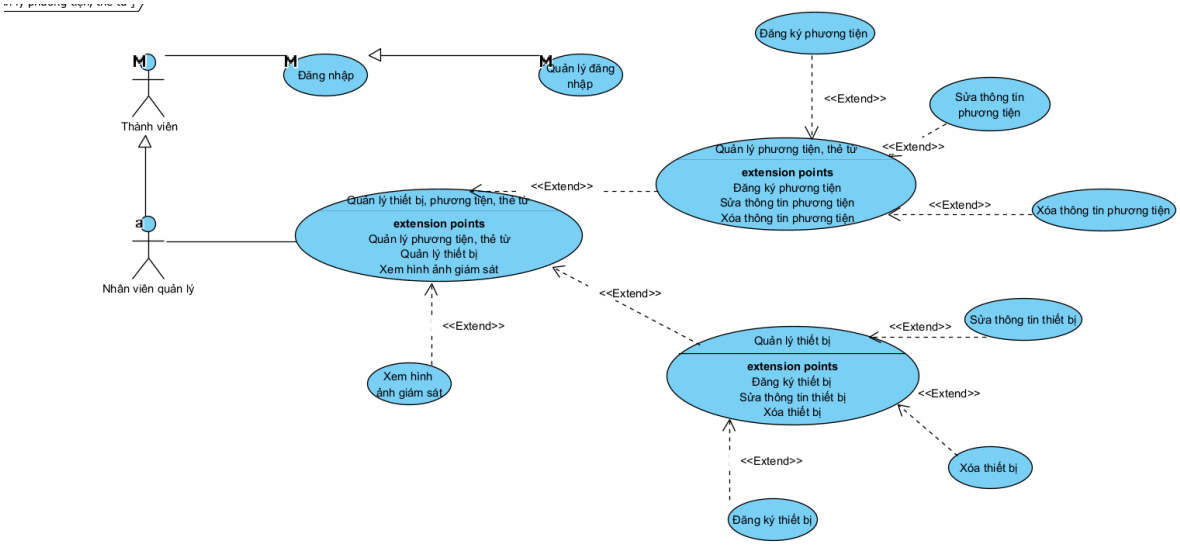


3.3.2. Biểu đồ use case chi tiết

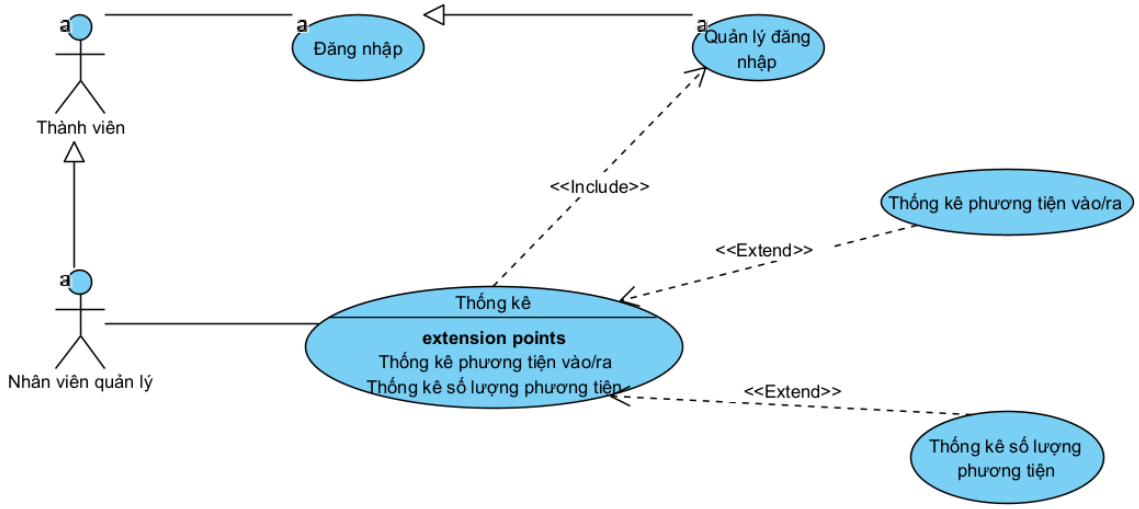
- Biểu đồ use case chi tiết xác thực xe vào ra



- Biểu đồ use case chi tiết quản lý người dùng, phương tiện và thẻ từ



- Biểu đồ use case chi tiết thống kê



3.4. Đặc tả luồng công việc

3.4.1. Đặc tả luồng công việc cho use case xác thực phương tiện vào/ra

Đặc tả luồng công việc	Use case: Xác thực phương tiện vào/ra
Actor	Người gửi phương tiện, nhân viên bảo vệ
Tiền điều kiện	Biển số xe của người gửi đã được đăng ký với hệ thống
Hậu điều kiện	Xe được xác nhận vào/ra thành công
Kịch bản	1. Người gửi phương tiện di chuyển xe đến vị trí chỉ định và quẹt

	thẻ từ vào đầu đọc. 2. Hệ thống đọc mã UID thẻ từ, đồng thời kích hoạt camera chụp lại ảnh biển số xe. 3. ESP32 đóng gói mã UID và dữ liệu ảnh, sau đó gửi đến máy chủ trung tâm qua một yêu cầu HTTP. 4. Máy chủ tiếp nhận dữ liệu và chạy AI để nhận dạng biển số từ ảnh. 5. Hệ thống tại máy chủ truy vấn cơ sở dữ liệu, so sánh đồng thời mã UID và biển số xe vừa nhận dạng được với dữ liệu đã đăng ký. 6. Máy chủ xác nhận thông tin hợp lệ và gửi lại kết quả "THÀNH CÔNG" cho ESP32. 7. ESP32 nhận kết quả, bật đèn xanh và điều khiển servo nâng thanh chắn barrier. 8. Đồng thời, máy chủ ghi nhận lượt xe vào CSDL và cập nhật trên giao diện quản lý. 9. Người gửi phương tiện điều khiển xe đi qua. 10. Cảm biến phát hiện xe đã đi qua và gửi tín hiệu cho ESP32. 11. Sau một khoảng trễ an toàn, ESP32 điều khiển servo hạ thanh chắn barrier, kết thúc quy trình.
Ngoại lệ	- Thẻ từ đã được sử dụng trước đó, nhân viên bảo vệ phải xác nhận thủ công - Dữ liệu không hợp lệ (Thẻ không tồn tại, biển số không khớp): Hệ thống gửi kết quả "THẤT BẠI". ESP32 giữ nguyên barrier đóng - Biển số không nhận dạng được (do mờ, bẩn, ngược sáng): Máy chủ gửi kết quả "LỖI NHẬN DẠNG"

3.4.2. Đặc tả luồng công việc cho use case quản lý phương tiện và thẻ từ

Đặc tả luồng công việc	Use case: Quản lý phương tiện và thẻ từ
Actor	Người quản lý
Tiền điều kiện	Người quản lý đăng nhập vào hệ thống với quyền “ADMIN”
Hậu điều kiện	Người quản lý xem , thay đổi các thông tin về phương tiện, thẻ từ, thiết bị
Kịch bản	1. Người quản lý đăng nhập vào hệ thống với username và password 2. Từ giao diện quản lý, có thể chọn các mục tương ứng: thẻ xe, phương tiện

	3. Người quản lý thực hiện các thao tác: thêm mới, chỉnh sửa, xóa hoặc tìm kiếm 4. Hệ thống kiểm tra và cập nhật dữ liệu, sau đó hiển thị ra trên màn hình
Ngoại lệ	- Dữ liệu nhập không hợp lệ hoặc sai định dạng, hệ thống hiển thị thông báo lỗi

3.4.3. Đặc tả luồng công việc cho use case thống kê

Đặc tả luồng công việc	Use case: Thống kê
Actor	Người quản lý đăng nhập vào hệ thống
Tiền điều kiện	Người quản lý đã đăng nhập vào hệ thống với quyền "ADMIN"
Hậu điều kiện	Người quản lý có thể xem được báo cáo các thống kê
Kịch bản	1. Người quản lý đăng nhập vào hệ thống 2. Hệ thống hiển thị giao diện quản lý và có chức năng thống kê 3. Người quản lý chọn chức năng thống kê 4. Hệ thống hiển thị giao diện thống kê với các chức năng: thống kê lượt vào/ra , thống kê số lượng phương tiện… 5. Người quản lý chọn chức năng thống kê lượt vào/ra 6. Hệ thống truy xuất dữ liệu trên cơ sở dữ liệu và hiển thị kết quả trên màn hình
Ngoại lệ	- Mục thống kê chưa có dữ liệu, hệ thống hiển thị thông báo

4. Yêu cầu phi chức năng

4.1. Hiệu năng

- Hệ thống phải đảm bảo thời gian xử lý trung bình ≤ 2 giây cho mỗi lượt xe (từ khi xe đến cổng đến khi barrier mở).

- Mô-đun AI xử lý ảnh phải có khả năng nhận dạng tối thiểu 15–20 khung hình/giây đối với ảnh đầu vào từ camera độ phân giải 720p.

- Hệ thống phải có khả năng xử lý đồng thời ít nhất 2 lượt xe (một xe vào và một xe ra) mà không gây nghẽn dữ liệu hoặc chậm trễ.

- Giao diện web phải phản hồi trong vòng ≤ 1 giây khi người dùng thao tác (chuyển trang, tra cứu, thống kê).

- Cơ sở dữ liệu phải hỗ trợ ghi và truy xuất tối thiểu 1000 bản ghi gửi xe/ngày mà không ảnh hưởng đến tốc độ truy vấn.

4.2. Bảo mật

- Hệ thống phải sử dụng cơ chế xác thực và phân quyền người dùng (Authentication & Authorization), trong đó các tài khoản được phân loại: *Admin*, *Nhân viên bảo vệ*, *Người dùng thường*.

- Việc đăng nhập và truyền dữ liệu giữa frontend – backend – thiết bị IoT phải được mã hóa bằng giao thức HTTPS hoặc MQTT bảo mật (SSL/TLS).

- Thông tin nhạy cảm như mật khẩu người dùng phải được băm bằng thuật toán BCrypt hoặc SHA-256, không lưu trữ ở dạng thuần văn bản.

- Các API quan trọng (gửi dữ liệu từ ESP32, AI module, truy cập dữ liệu người dùng) phải được bảo vệ bằng token JWT hợp lệ.

- Hệ thống phải có cơ chế ghi log hoạt động và phát hiện truy cập bất thường, giúp người quản lý dễ dàng kiểm soát và phục hồi khi xảy ra sự cố.

4.3. Độ tin cậy

- Hệ thống phải duy trì tỷ lệ hoạt động (uptime) ≥ 98% trong suốt quá trình vận hành.

- Dữ liệu gửi xe phải được ghi nhận và lưu trữ an toàn ngay cả khi mất kết nối tạm thời giữa thiết bị IoT và server; các bản ghi chưa gửi sẽ được tự động đồng bộ lại khi kết nối được phục hồi.

- Cơ sở dữ liệu cần có cơ chế sao lưu định kỳ (backup) để ngăn ngừa mất dữ liệu.

- Hệ thống phải xử lý lỗi một cách an toàn: nếu camera hoặc AI module gặp sự cố, nhân viên vẫn có thể nhập tay thông tin để đảm bảo hoạt động không gián đoạn.

- Phản hồi lỗi và cảnh báo được hiển thị rõ ràng trên giao diện để người vận hành dễ dàng khắc phục.

4.4. Khả năng mở rộng

- Hệ thống phải cho phép mở rộng quy mô dễ dàng khi tăng số lượng thiết bị IoT, camera hoặc cổng ra/vào.

- Kiến trúc phần mềm được xây dựng theo mô hình modular (chia mô-đun độc lập: IoT, AI, Backend, Frontend), giúp dễ dàng nâng cấp từng phần mà không ảnh hưởng đến toàn hệ thống.

- Cơ sở dữ liệu được thiết kế hỗ trợ nhiều điểm kiểm soát (multi-gate) và nhiều người dùng đồng thời (multi-user).

- Hệ thống có thể được triển khai trên máy chủ cục bộ (Local Server) hoặc mở rộng lên Cloud (AWS, Firebase, ThingsBoard) khi cần quản lý từ xa.

- Hỗ trợ tích hợp thêm các tính năng nâng cao như thanh toán điện tử, phân tích lưu lượng, cảnh báo thông minh trong tương lai mà không cần thay đổi cấu trúc lõi.

4.5. Chi phí và năng lượng

- Hệ thống phải sử dụng các linh kiện phổ thông, chi phí thấp và dễ tìm trên thị trường như: ESP32, đầu đọc RFID RC522, camera USB/IP, cảm biến siêu âm HC-SR04.

- Phần mềm được xây dựng hoàn toàn từ các công nghệ mã nguồn mở (Spring Boot, ReactJS, Flask, MySQL, MQTT) để giảm chi phí bản quyền.

- Các thiết bị IoT (ESP32, camera) phải tiêu thụ điện năng thấp, có thể vận hành liên tục trong 24 giờ mà không quá tải.

- Hệ thống ưu tiên tối ưu năng lượng xử lý của AI module bằng cách chỉ kích hoạt mô-đun nhận dạng khi phát hiện chuyển động, nhằm tiết kiệm tài nguyên và điện năng.

5. Ràng buộc kỹ thuật và môi trường

5.1. Ràng buộc phần cứng

- Module ESP32-CAM:
    - Do bộ nhớ trong và khả năng xử lý hạn chế (CPU lõi kép 240MHz, ~520KB SRAM), module này chỉ được sử dụng cho nhiệm vụ thu nhận và truyền hình ảnh, không thực hiện xử lý AI tại biên. * Chất lượng hình ảnh và tốc độ khung hình phụ thuộc vào cảm biến OV2640, đòi hỏi điều kiện ánh sáng tốt để có thể nhận dạng biển số.
- Máy chủ trung tâm (Central Server):
    - Hiệu năng của toàn bộ hệ thống (đặc biệt là tốc độ nhận dạng biển số) phụ thuộc hoàn toàn vào cấu hình của máy chủ (CPU, RAM, GPU nếu có).
    - Hệ thống yêu cầu một máy tính luôn hoạt động để xử lý các yêu cầu từ thiết bị tại cổng.
- Module đọc thẻ RFID RC522:
    - Hoạt động ở tần số 13.56MHz và có tầm đọc ngắn (khoảng 3-5 cm). Điều này ràng buộc thiết kế vật lý của trạm kiểm soát, yêu cầu người dùng phải đưa thẻ lại gần đầu đọc.
- Mạng Wi-Fi:
    - Hệ thống phụ thuộc nghiêm ngặt vào một mạng Wi-Fi ổn định, có độ trễ thấp để giao tiếp giữa ESP32-CAM và máy chủ. Mất kết nối mạng sẽ làm toàn bộ hệ thống ngưng hoạt động.

5.2. Ràng buộc phần mềm

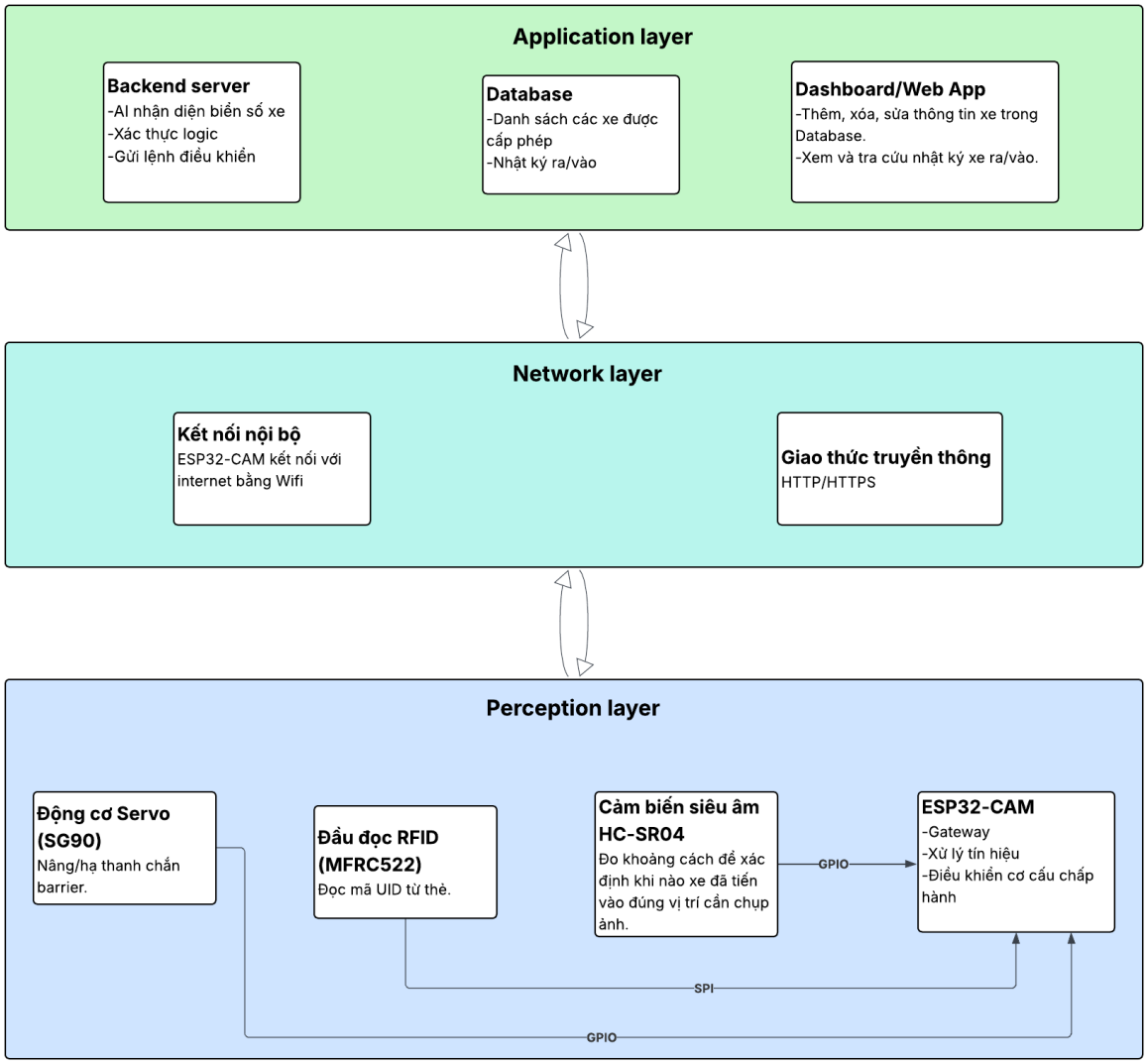
- Mô hình AI nhận dạng biển số:
    - Mô hình AI (ví dụ: YOLO, EAST) được triển khai trên máy chủ trung tâm, không phải trên thiết bị nhúng.
    - Yêu cầu máy chủ phải cài đặt đầy đủ môi trường và các thư viện cần thiết (Python, OpenCV, Ultralytics/TensorFlow).
- Firmware trên ESP32-CAM:
    - Được phát triển bằng ngôn ngữ C/C++ (trên nền tảng Arduino IDE hoặc ESP-IDF).
    - Chức năng của firmware bị giới hạn trong việc: chụp ảnh, kết nối Wi-Fi, gửi dữ liệu hình ảnh đến server, nhận lệnh điều khiển (mở/đóng barie) và điều khiển servo.
- Giao thức giao tiếp:
    - Dữ liệu hình ảnh và lệnh điều khiển được truyền tải giữa ESP32-CAM và máy chủ thông qua các giao thức mạng như HTTP POST hoặc WebSocket. Lựa chọn này ràng buộc kiến trúc phần mềm của cả hai phía.
- Cơ sở dữ liệu:

    - Hệ thống sử dụng MySQL, do đó máy chủ phải được cài đặt và cấu hình MySQL server. Cấu trúc bảng biểu được thiết kế từ đầu sẽ ràng buộc các chức năng quản lý và thống kê.

5.3. Ràng buộc môi trường hoạt động

- Điều kiện ánh sáng và góc chụp:
    - Hệ thống được thiết kế để hoạt động tối ưu trong môi trường phòng thí nghiệm với ánh sáng được kiểm soát và không đổi.
    - Camera phải được đặt ở một góc và khoảng cách cố định so với vị trí của biển số xe để đảm bảo độ chính xác nhận dạng cao nhất.
- Phụ thuộc vào kết nối mạng:
    - Không giống các hệ thống xử lý tại biên, hệ thống này không thể hoạt động ngoại tuyến (offline). Mọi chức năng cốt lõi đều yêu cầu kết nối liên tục đến máy chủ trung tâm.
- Đối tượng nhận dạng:
    - Hệ thống được huấn luyện và kiểm thử với biển số xe chuẩn của Việt Nam, trong điều kiện sạch sẽ, không bị móp méo, che khuất. Các loại biển số không chuẩn hoặc bị bẩn có thể làm giảm độ chính xác.

6. Mô hình yêu cầu



Lớp Perception

Đây là lớp thấp nhất, có nhiệm vụ tương tác trực tiếp với môi trường vật lý để thu thập dữ liệu và thực thi hành động. Trong hệ thống barrier tự động, lớp này bao gồm các cảm biến và cơ cấu chấp hành sau:

- ESP32-CAM: Đóng vai trò là Gateway (cổng trung gian) và bộ não trung tâm của lớp này. Nó có nhiệm vụ:
    - Chụp ảnh biển số xe khi có tín hiệu từ cảm biến.
    - Nhận dữ liệu mã UID từ đầu đọc RFID.
    - Kiểm tra khoảng cách từ cảm biến siêu âm.
    - Đóng gói dữ liệu và gửi lên Lớp Application.
    - Nhận lệnh điều khiển (mở/đóng) từ Lớp Application và ra lệnh cho Servo.

- Đầu đọc RFID (MFRC522): Đọc mã định danh duy nhất (UID) từ thẻ từ của người lái.
- Động cơ Servo (SG90/MG90S): Nhận lệnh từ ESP32-CAM để nâng hoặc hạ thanh chắn barrier.
- Cảm biến siêu âm: Phát hiện khi có xe tiến vào đúng vị trí, gửi tín hiệu để ESP32-CAM bắt đầu quá trình xác thực.

Lớp Network

Lớp này chịu trách nhiệm truyền tải dữ liệu một cách an toàn và hiệu quả giữa Lớp Perception và Lớp Application.

- Kết nối nội bộ: ESP32-CAM sử dụng Wi-Fi để kết nối vào mạng LAN thông qua router.
- Giao thức truyền thông: HTTP/HTTPS là giao thức chính được sử dụng.
    - Nhiệm vụ: Dùng để đóng gói và gửi đồng thời cả dữ liệu ảnh biển số và mã UID thẻ từ trong một yêu cầu POST duy nhất từ ESP32-CAM lên Backend Server. Sau đó, nó cũng nhận phản hồi (mở/đóng) từ server. Cách tiếp cận này đảm bảo tính toàn vẹn và đồng bộ cho mỗi lượt xác thực.

Lớp Application

Đây là lớp cao nhất, nơi dữ liệu được xử lý, lưu trữ và đưa ra các quyết định logic, đồng thời cung cấp giao diện tương tác cho người quản lý.

- Backend Server: Là "bộ não" của hệ thống, có các nhiệm vụ:
    - Nhận yêu cầu HTTP chứa ảnh và mã thẻ từ ESP32-CAM.
    - Sử dụng các thư viện (OpenCV, Tesseract, Yolo) để nhận dạng biển số xe từ dữ liệu ảnh.
    - Xác thực logic: Đối chiếu đồng thời cả biển số xe và mã thẻ từ với thông tin trong Database.
    - Gửi lệnh điều khiển ("OPEN" hoặc "CLOSE") ngược lại cho ESP32-CAM.
- Database: Lưu trữ toàn bộ thông tin của hệ thống:
    - Danh sách các xe được cấp phép (Biển số xe - Mã UID thẻ từ tương ứng).
    - Nhật ký ra/vào (lưu lại biển số, mã thẻ và thời gian của mỗi lượt xe).
- Dashboard/Web App (Giao diện quản lý): Cung cấp giao diện cho người quản lý để thực hiện các tác vụ:
    - Thêm, xóa, sửa thông tin xe trong Database.
    - Xem và tra cứu nhật ký xe ra/vào.