

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

KHOA CÔNG NGHỆ THÔNG TIN I

HỌC PHẦN IOT VÀ ỨNG DỤNG



BÁO CÁO BÀI TẬP LỚN

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

GIẢNG VIÊN HƯỚNG DẪN: KIM NGỌC BÁCH

DANH SÁCH THÀNH VIÊN NHÓM 05 – LỚP D22CNPM02

HỌ VÀ TÊN	MÃ SINH VIÊN
NGUYỄN VĂN HỌC	B22DCCN352
NGUYỄN HỮU LỘC	B22DCCN508
NGUYỄN THANH PHONG	B22DCCN616

Hà Nội – 2025

Mục lục

PHẦN 1. GIỚI THIỆU CHUNG	5
1.1. Lý do chọn đề tài	5
1.2. Tổng quan đề tài	5
1.3. Mục đích nghiên cứu	6
1.4. Đối tượng và phạm vi nghiên cứu	6
1.4.1. Đối tượng nghiên cứu	6
1.4.2. Phạm vi nghiên cứu	7
1.5. Phương pháp nghiên cứu	7
1.5.1. Nghiên cứu lý thuyết	7
1.5.2. Thực nghiệm phần cứng	7
1.5.3. Thực nghiệm phần mềm	8
1.5.4. Kiểm thử	8
1.6. Công nghệ và thiết bị sử dụng (giải thích chi tiết)	8
PHẦN 2. NỀN TẢNG LÝ THUYẾT	9
2.1. ESP32-CAM	9
2.2. Module RFID RC522	11
2.3. Cảm biến siêu âm HC-SR04	13
2.4. Động cơ Servo SG90	15
2.5. Giao thức HTTP	16
2.6. RESTful API	17
2.7. Kết nối WiFi	17
2.8. WebSocket (truyền dữ liệu real-time)	18
2.9. Giao thức giữa cảm biến và ESP32-CAM	18
Phần 3: Phân tích hệ thống	19
3.1.1. Biểu đồ usecase tổng quan	19
3.1.2. Biểu đồ usecase chi tiết	21
3.1.2.1. Biểu đồ usecase chức năng xác thực vào/ra	21
3.1.2.2. Biểu đồ usecase chức năng đăng ký thẻ tháng	21
3.1.2.3. Biểu đồ usecase chức năng thống kê	22

3.1.2.4. Biểu đồ usecase chức năng xem lịch sử vào/ra.....	22
3.2. Kịch bản các usecase	22
3.2.1. Chức năng giám sát luồng vào/ra.....	22
3.2.2. Chức năng đăng ký thẻ tháng.....	23
3.2.3. Chức năng thống kê.....	25
3.2.4. Chức năng xem lịch sử vào/ra	25
Phần 4: THIẾT KẾ HỆ THỐNG	27
4.1. Kiến trúc tổng quan hệ thống.....	27
4.2. Thiết kế phần cứng (Hardware Design)	29
4.3. Thiết kế Cơ sở dữ liệu	32
4.4. Biểu đồ tuần tự.....	35
4.4.1. Chức năng xác thực luồng vào/ra	35
4.4.2. Chức năng đăng ký thẻ tháng.....	36
4.4.3. Chức năng thống kê.....	36
4.4.4. Chức năng xem lịch sử vào/ra	37
4.5. Giao diện ứng dụng	37
4.5.1. Giao diện giám sát lối vào (Check-in Dashboard).....	37
4.5.2. Giao diện giám sát lối ra (Check-out Dashboard).....	38
4.5.3. Giao diện đăng ký vé tháng (Monthly Ticket Registration)	39
4.5.4. Giao diện xem thống kê.....	40
4.5.5. Giao diện xem lịch sử ra vào.....	41
CHƯƠNG 5: KẾT LUẬN & HƯỚNG PHÁT TRIỂN	42
5.1. Kết quả đạt được	42
5.1.1. Về mặt lý thuyết.....	42
5.1.2. Về mặt thực tiễn.....	42
5.2. Hạn chế.....	43
5.3. Hướng phát triển tương lai.....	43
5.3.1. Về hệ thống.....	43
5.3.2. Về tính năng thông minh.....	43

Danh mục hình ảnh

Hình 2. 1: ESP32-CAM.....	10
Hình 2. 2: Thẻ từ RFID và đầu đọc RC522	12
Hình 2. 3: Cảm biến siêu âm	13
Hình 2. 4: Động cơ servo.....	15
Hình 3. 1:Usecase tổng quan	20
Hình 3. 2: Usecase xác thực phương tiện vào/ra	21
Hình 3. 3: Usecase đăng ký thẻ tháng	21
Hình 3. 4: Usecase thống kê	22
Hình 3. 5: Usecase xem lịch sử vào/ra	22
Hình 4. 1: Kiến trúc tổng quan hệ thống	27
Hình 4. 2: Sơ đồ mạch	29
Hình 4. 3: Cơ sở dữ liệu	32
Hình 4. 4: Sequence diagram xác thực phương tiện vào/ra.....	35
Hình 4. 5: Sequence diagram đăng ký thẻ tháng	36
Hình 4. 6: Sequence diagram chức năng thống kê	36
Hình 4. 7: Sequence diagram chức năng xem lịch sử vào/ra.....	37
Hình 4. 8: Giao diện luồng vào.....	37
Hình 4. 9: Giao diện luồng ra	38
Hình 4. 10: Giao diện đăng ký thẻ tháng.....	39
Hình 4. 11: Giao diện thống kê.....	40
Hình 4. 12: Giao diện xem lịch sử vào/ra.....	41

PHẦN 1. GIỚI THIỆU CHUNG

1.1. Lý do chọn đề tài

Trong bối cảnh xã hội hiện đại, quá trình đô thị hóa diễn ra nhanh chóng khiến mật độ phương tiện tại chung cư, trường học, khu công nghiệp, bãi đỗ xe công cộng, bệnh viện, trung tâm thương mại... ngày càng tăng cao. Điều này kéo theo nhu cầu quản lý phương tiện ra/vào một cách chính xác, minh bạch và hiệu quả hơn. Tuy nhiên, đa số mô hình quản lý bãi xe hiện nay vẫn vận hành theo cách thủ công, gây ra nhiều hạn chế:

- Thời gian xử lý lâu, đặc biệt vào giờ cao điểm.
- Dễ sai sót khi ghi chép, dẫn đến thất lạc dữ liệu hoặc nhầm lẫn biển số.
- Khó thống kê và không có dữ liệu thời gian thực.
- Dễ gian lận, ví dụ như dùng chung thẻ hoặc thay đổi biển số xe.
- Tốn nhiều nhân lực, chi phí vận hành cao.
- Tính bảo mật thấp, khó đối chiếu dữ liệu khi có sự cố.

Sự phát triển mạnh mẽ của công nghệ IoT, Machine Learning và AI thị giác máy tính (Computer Vision) đã tạo ra cơ hội tự động hóa hoàn toàn quy trình quản lý ra/vào bãi xe. Việc ứng dụng ESP32-CAM, RFID, cảm biến và AI không chỉ giúp xử lý nhanh hơn mà còn nâng mức độ an toàn, giảm chi phí và tối ưu vận hành.

Chính vì vậy, nhóm chọn đề tài “Hệ thống ra vào bãi đỗ xe thông minh sử dụng IoT và AI” với mong muốn giải quyết vấn đề thực tiễn, áp dụng kiến thức đã học để tạo ra một mô hình tự động hoá hoàn chỉnh, có thể triển khai thực tế với chi phí thấp.

1.2. Tổng quan đề tài

Đề tài hướng tới việc xây dựng một hệ thống bãi giữ xe tự động dựa trên sự kết hợp giữa IoT và AI, với các thành phần chính như sau:

- Nhận dạng biển số xe tự động bằng camera ESP32-CAM kết hợp mô hình AI (YOLOv8 và OCR).
- Xác thực chủ xe thông qua thẻ từ RFID RC522 để đảm bảo xe và thẻ khớp nhau.
- Phát hiện vật cản bằng cảm biến siêu âm HC-SR04 để nhận biết xe đến cổng.

- Đóng/mở barrier tự động bằng động cơ Servo SG90.
- Kết nối IoT qua WiFi, truyền dữ liệu tức thời về server qua HTTP và WebSocket.
- Server Backend xử lý thông minh (Spring Boot), gồm nhận diện biển số, xác thực người dùng, điều khiển barrier.
- Giao diện Web (ReactJS) giúp quản lý người dùng, biển số xe, lịch sử ra/vào theo thời gian thực.

Hệ thống mô phỏng một mô hình bãi đỗ xe thực tế, có tính ứng dụng cao và có thể triển khai ở quy mô nhỏ – phù hợp trường học, nhà xe, khu dân cư, văn phòng...

1.3 Mục đích nghiên cứu

Mục tiêu chính của đề tài:

- Xây dựng mô hình bãi giữ xe tự động có thể hoạt động thật với các tác vụ cơ bản.
- Kết hợp các thiết bị phần cứng IoT (camera, RFID, cảm biến, động cơ) để tạo ra hệ thống hoàn chỉnh.
- Tăng độ chính xác nhận diện biển số thông qua mô hình AI YOLOv8 kết hợp OCR.
- Tự động hóa hoạt động barrier mà không cần người vận hành.
- Tối ưu thời gian xử lý, đảm bảo xe đi qua nhanh và an toàn.
- Tạo giao diện Web trực quan, hỗ trợ xem dữ liệu real-time, biểu đồ, lịch sử.
- Giảm thiểu rủi ro, sai sót bằng quy trình xác thực hai yếu tố (RFID + biển số).

1.4. Đối tượng và phạm vi nghiên cứu

1.4.1. Đối tượng nghiên cứu

- Thiết bị IoT:
 - ESP32-CAM (chụp ảnh + điều khiển).
 - RFID RC522 (đọc UID thẻ).
 - Cảm biến siêu âm HC-SR04 (đo khoảng cách, phát hiện xe).

- Servo SG90 (đóng/mở barrier).
- Giao thức kết nối: WiFi, SPI, PWM, HTTP, WebSocket.
- Backend: Spring Boot xử lý AI, API, điều khiển thiết bị, lưu CSDL.
- Frontend: ReactJS quản lý giao diện.

1.4.2. Phạm vi nghiên cứu

- Xây dựng mô hình ở mức độ demo, chạy trong phòng lab.
- Không đi sâu vào:
 - Quản lý chỗ trống nâng cao.
 - Thu phí tự động qua QR/Ví điện tử.
 - Xử lý môi trường phức tạp (ánh sáng mạnh, mưa, đêm).
 - Hệ thống phân quyền bảo mật cấp doanh nghiệp.

1.5. Phương pháp nghiên cứu

Đề tài sử dụng các phương pháp nghiên cứu sau:

1.5.1. Nghiên cứu lý thuyết

- Tìm hiểu kiến trúc IoT 3 lớp: Nhận cảm – Kết nối – Ứng dụng.
- Học cách giao tiếp cảm biến:
 - SPI, I2C, UART, PWM.
- Nghiên cứu thuật toán nhận dạng ảnh:
 - YOLOv8 (phát hiện đối tượng).
 - OCR (nhận dạng ký tự từ biển số).
- Tìm hiểu chuẩn thiết kế API với phương pháp RESTful.

1.5.2. Thực nghiệm phần cứng

- Kết nối ESP32-CAM với:
 - RC522 qua giao tiếp SPI.
 - Servo SG90 qua PWM.
 - HC-SR04 qua chân Trigger / Echo.

- Test hoạt động của từng mô-đun, sau đó tích hợp hệ thống.

1.5.3. Thực nghiệm phần mềm

- Huấn luyện / Fine-tune mô hình YOLOv8 cho bài toán biển số Việt Nam.
- Tách biển số từ ảnh bằng OpenCV (crop, threshold...).
- OCR ký tự biển số (EasyOCR / Tesseract).
- Xây dựng server bằng Spring Boot:
 - Controller → Service → Repository.
- Xây dựng giao diện bằng ReactJS:
 - Dashboard.
 - Quản lý thẻ RFID.
 - Lịch sử xe ra vào.

1.5.4. Kiểm thử

- Kiểm thử thiết bị và AI:
 - Test ra/vào liên tục.
 - Test điều kiện ánh sáng khác nhau.
 - Test thẻ RFID sai/hỏng.
- Kiểm thử end-to-end:
 - IoT ↔ AI ↔ Server ↔ Web.

1.6. Công nghệ và thiết bị sử dụng (giải thích chi tiết)

Thành phần	Vai trò
ESP32-CAM	Thiết bị chính điều khiển, chụp ảnh, đọc RFID, kết nối WiFi
RC522	Đọc UID thẻ từ để xác thực người dùng
HC-SR04	Phát hiện xe đến gần cổng, kích hoạt camera chụp ảnh

Servo SG90	Điều khiển đóng/mở barrier
YOLOv8 + OCR	Nhận dạng biển số xe
WebSocket	Truyền dữ liệu real-time đến web
MySQL	Lưu lịch sử, thông tin thẻ, biển số
ReactJS	Giao diện quản lý web
Spring Boot	Backend API + xử lý logic AI

PHẦN 2. NỀN TẢNG LÝ THUYẾT

2.1. ESP32-CAM

2.1.1. Giới thiệu

ESP32-CAM là một board IoT nhỏ gọn nhưng được tích hợp nhiều chức năng mạnh mẽ, thường sử dụng trong các ứng dụng thị giác máy tính, truyền ảnh, giám sát và điều khiển từ xa. Board được trang bị module camera OV2640 cùng khả năng kết nối WiFi/Bluetooth, thích hợp cho các dự án IoT yêu cầu xử lý ảnh và truyền dữ liệu thời gian thực.

Theo Espressif (2023), ESP32-CAM sử dụng SoC ESP32 gồm CPU two-core 32-bit Tensilica Xtensa LX6 với xung nhịp lên đến 240 MHz, đi kèm bộ nhớ PSRAM 4MB hỗ trợ xử lý ảnh độ phân giải cao. Bảng mạch có kích thước nhỏ, giá thành rẻ, cộng đồng hỗ trợ lớn nên rất phổ biến trong các dự án như nhận dạng biển số, giám sát an ninh, điều khiển qua WiFi...

Trong hệ thống quản lý bãi đỗ xe thông minh, ESP32-CAM là thiết bị trung tâm đảm nhận các nhiệm vụ:

- Chụp ảnh biển số xe khi có xe đi vào vùng quét.
- Đọc UID từ module RFID qua giao tiếp SPI.
- Điều khiển Servo thông qua tín hiệu PWM để đóng/mở barrier.

- Gửi dữ liệu (ảnh biên số, UID RFID) lên server qua HTTP POST.
- Nhận lệnh điều khiển từ Backend (OPEN/CLOSE).

ESP32-CAM đóng vai trò như bộ điều khiển IoT tích hợp Camera, kết nối tất cả cảm biến và thực thi theo logic điều khiển do server đưa ra.



Hình 2. 1: ESP32-CAM

2.1.2. Thông số kỹ thuật

Theo Espressif, 2023:

- Chip vi xử lý: ESP32 Dual-core Xtensa LX6, 240 MHz
- Camera: OV2640 (2MP)
- PSRAM: 4MB
- Flash: 4MB
- WiFi: 2.4GHz 802.11 b/g/n
- Bluetooth: v4.2 BR/EDR và BLE
- Điện áp hoạt động: 5V
- GPIO: 9 chân khả dụng tùy chế độ
- Giao tiếp hỗ trợ: UART, SPI, PWM, I2C
- Hỗ trợ xử lý ảnh JPEG, RGB, YUV
- Kích thước: 40 x 27 mm

2.1.3. Các chân kết nối

- GPIO phục vụ dự án:
 - PWM → kết nối Servo
 - SPI → kết nối RFID RC522
 - GPIO Input → Trigger/Echo cảm biến siêu âm
- Các chân nguồn:
 - 5V: cấp nguồn
 - GND: mass

2.1.4. Nguyên lý hoạt động trong hệ thống

1. Khi cảm biến siêu âm phát hiện xe → ESP32-CAM được kích hoạt.
2. ESP32-CAM đọc UID thẻ RFID.
3. ESP32-CAM chụp ảnh biển số xe.
4. Gửi HTTP POST (ảnh + UID) lên server.
5. Server xử lý bằng AI và trả về lệnh OPEN/CLOSE.
6. ESP32-CAM xuất tín hiệu PWM → Servo mở barrier.
7. Xe đi qua → Servo quay lại 0° để đóng barrier.

2.2. Module RFID RC522

2.2.1. Giới thiệu

RFID RC522 là module đọc thẻ từ tiêu chuẩn MIFARE 13.56MHz, sử dụng rộng rãi trong các ứng dụng kiểm soát ra/vào như bãi gửi xe, chấm công, truy cập cửa từ. Module hoạt động theo chuẩn ISO/IEC 14443A, có thể đọc UID (mã định danh duy nhất) với khoảng cách 3–5 cm.

Theo NXP Semiconductors (2022), RC522 sử dụng giao tiếp SPI tốc độ cao, tiêu thụ năng lượng thấp, hỗ trợ tốc độ truyền dữ liệu lên đến 10 Mbit/s. Giá thành rẻ, dễ lập trình và độ ổn định cao khiến RC522 trở thành lựa chọn phổ biến trong các hệ thống xác thực IoT.

Trong mô hình bãi đỗ xe thông minh:

- Mỗi người dùng được gán một thẻ MIFARE chứa UID cố định.

- UID này được đọc tại cổng ra/vào.
- UID được gửi kèm ảnh biển số lên server để kiểm tra khớp lệnh
- Điều này giúp tránh gian lận, như dùng biển số giả hoặc xe không thuộc chủ sở hữu.



Hình 2. 2: Thẻ từ RFID và đầu đọc RC522

2.2.2. Thông số kỹ thuật

- Tần số hoạt động: 13.56 MHz
- Giao tiếp: SPI (mặc định), I2C, UART
- Điện áp hoạt động: 3.3V
- Khoảng cách đọc: 3–5 cm
- Loại thẻ hỗ trợ: MIFARE 1K, 4K, Ultralight...
- Dòng tiêu thụ: 13–26 mA
- Tốc độ truyền dữ liệu: 10 Mbit/s
- Kích thước: 40 x 60 mm

2.2.3. Cấu tạo và chân tín hiệu

- SDA (SS) – chọn thiết bị SPI
- SCK – chân clock
- MOSI – Master Out Slave In
- MISO – Master In Slave Out

- RST – reset
- 3.3V – cấp nguồn
- GND – mass

2.2.4. Nguyên lý hoạt động

1. Thẻ MIFARE đi vào vùng từ trường của RC522.
2. Module gửi yêu cầu → thẻ phản hồi bằng UID.
3. ESP32-CAM đọc UID qua SPI.
4. UID được gửi lên server cùng ảnh biển số.
5. Server kiểm tra khớp UID – biển số → ra quyết định mở barrier.

2.3. Cảm biến siêu âm HC-SR04

2.3.1. Giới thiệu

HC-SR04 là cảm biến siêu âm đo khoảng cách phổ biến trong robot, tự động hóa, tránh vật cản. Cảm biến sử dụng cặp phát – thu sóng âm ở tần số 40 kHz để đo thời gian phản hồi (echo), từ đó tính ra khoảng cách đến vật thể phía trước.

Trong hệ thống bãi đỗ xe, HC-SR04 đảm nhận các nhiệm vụ quan trọng:

- Phát hiện khi xe tiếp cận cổng → kích hoạt camera chụp ảnh.
- Giám sát xe đã đi qua để servo đóng barrier.
- Ngăn tình huống barrier đóng khi xe còn bên dưới.



Hình 2. 3: Cảm biến siêu âm

2.3.2. Thông số kỹ thuật

Theo Waveshare (2023):

- Điện áp hoạt động: 5V
- Dòng tiêu thụ: 15 mA
- Tần số siêu âm: 40 kHz
- Khoảng đo: 2 – 400 cm
- Độ chính xác: ± 3 mm
- Góc đo: 15°
- Kích thước: 45 x 20 mm

2.3.3. Cấu tạo và chân tín hiệu

- VCC: 5V
- GND: mass
- Trig: phát xung kích hoạt
- Echo: nhận xung phản hồi

2.3.4. Nguyên lý hoạt động

Theo How2Electronics (2023), cảm biến hoạt động theo 5 bước:

1. ESP32 gửi một xung Trigger rộng 10 μ s.
2. HC-SR04 phát ra sóng siêu âm tần số 40 kHz.
3. Sóng âm gặp vật \rightarrow phản xạ lại \rightarrow quay về cảm biến.
4. Cảm biến đưa mức cao lên chân Echo với thời gian tương ứng với thời gian sóng âm di chuyển.
5. Khoảng cách được tính bằng công thức:

$$\text{Khoảng cách} = \frac{\text{Thời gian Echo} \times 340 \text{ m/s}}{2}$$

Phần chia 2 vì sóng đi 2 chiều (đi – về).

HC-SR04 cho phép phát hiện xe chính xác và là tín hiệu mở đầu cho toàn bộ quy trình nhận dạng.

2.4. Động cơ Servo SG90

2.4.1. Giới thiệu

Servo SG90 là động cơ servo mini 9g, được sử dụng phổ biến trong robot, mô hình điều khiển, cơ cấu đóng/mở. Servo có khả năng quay từ 0° đến 180° với độ chính xác cao nhờ bộ điều khiển bên trong, hoạt động bằng tín hiệu PWM do vi điều khiển cung cấp.

Trong hệ thống bãi đỗ xe thông minh:

- Servo đóng vai trò barrier tự động.
- Góc quay được điều chỉnh theo lệnh từ server:
 - $0^\circ \rightarrow$ đóng cổng
 - $90^\circ \rightarrow$ mở cổng



Hình 2. 4: Động cơ servo

2.4.2. Thông số kỹ thuật

Theo TowerPro, 2024:

- Điện áp hoạt động: 4.8–6.0V
- Góc quay: $0^\circ - 180^\circ$
- Tốc độ quay: 0.1 s/ 60° (4.8V)

- Lực kéo: 1.8 kg·cm
- Tín hiệu điều khiển: PWM 50Hz
- Dải xung:
 - 0.5 ms \rightarrow 0°
 - 1.5 ms \rightarrow 90°
 - 2.5 ms \rightarrow 180°
- Kích thước: 22.8 x 12.6 x 22.5 mm

2.4.3. Cấu tạo và chân kết nối

- VCC: 5V
- GND: mass
- Signal (PWM): nhận xung điều khiển từ ESP32-CAM

2.4.4. Nguyên lý hoạt động

1. ESP32 gửi xung PWM 50Hz tới Servo.
2. Dựa trên độ rộng xung, servo quay tới góc tương ứng.
3. Khi server gửi lệnh “OPEN”, ESP32 xuất xung 1.5 ms \rightarrow Servo mở barrier.
4. Khi xe đi qua, cảm biến siêu âm báo \rightarrow ESP32 xuất xung 0.5 ms \rightarrow Servo đóng barrier.

Servo giúp quá trình đóng/mở diễn ra tự động, nhanh và chính xác.

2.5. Giao thức HTTP

HTTP (HyperText Transfer Protocol) là giao thức truyền tải dữ liệu theo mô hình Request – Response, được sử dụng phổ biến trong các hệ thống IoT – Web Server.

Cách hoạt động trong hệ thống

- ESP32-CAM gửi yêu cầu HTTP POST lên server bao gồm:
 - Ảnh biển số xe chụp từ camera.
 - Mã UID của thẻ RFID (nếu có).
 - Thời gian ghi nhận sự kiện.

- Server phản hồi bằng JSON, ví dụ:

```
{
  "status": "OPEN"
}
```

Hoặc “CLOSE” tùy theo kết quả xử lý.

Ưu điểm

- Cấu trúc đơn giản, dễ triển khai.
- Hoạt động ổn định với REST API của Spring Boot.
- Dễ kiểm tra bằng Postman, curl.

2.6. RESTful API

Hệ thống backend được xây dựng theo chuẩn REST (Representational State Transfer) để tổ chức API rõ ràng, dễ bảo trì.

Các phương thức REST dùng trong hệ thống

- GET – Lấy dữ liệu (biển số, lịch sử vào/ra, trạng thái barrier).
- POST – Nhận dữ liệu từ thiết bị IoT (ảnh, UID, thời gian).
- PUT – Cập nhật thông tin xe, chủ xe hoặc cấu hình hệ thống.
- DELETE – Xóa bản ghi hoặc dữ liệu không cần thiết.

Nhiệm vụ của Backend

- Nhận và xử lý ảnh biển số bằng mô hình AI.
- Đối chiếu biển số/UID thẻ RFID với cơ sở dữ liệu.
- Ghi lại log vào MySQL (thời gian vào – ra).
- Gửi lại lệnh OPEN/CLOSE cho ESP32-CAM để điều khiển barrier.

2.7. Kết nối WiFi

ESP32-CAM sử dụng WiFi để giao tiếp với server.

Dữ liệu truyền qua WiFi

- Ảnh biển số từ camera.

- UID thẻ RFID đọc được.
- Trạng thái barrier và phản hồi từ server.

Ưu điểm

- Không cần dây truyền dữ liệu → linh hoạt trong lắp đặt.
- Tốc độ truyền cao, phù hợp gửi ảnh và dữ liệu thời gian thực.
- Cấu hình đơn giản, dễ sử dụng cho bài demo và ứng dụng thực tế.

2.8. WebSocket (truyền dữ liệu real-time)

WebSocket là giao thức hỗ trợ truyền dữ liệu hai chiều, liên tục, theo thời gian thực (real-time).

Ứng dụng WebSocket trong hệ thống

- Gửi log vào/ra ngay lập tức lên giao diện web mà không cần reload trang.
- Truyền trạng thái barrier (OPEN/CLOSE/Đang xử lý).
- Có thể dùng để phát video stream real-time từ ESP32-CAM (nếu cần).

Lợi ích

- Giao diện quản lý xe hoạt động tức thời.
- Quản trị viên xem trực tiếp camera và trạng thái bãi xe.
- Hiệu quả khi cập nhật danh sách xe vào – ra theo từng giây.

2.9. Giao thức giữa cảm biến và ESP32-CAM

Chi tiết giao tiếp:

Thiết bị	Giao thức	Vai trò
RC522	SPI	Đọc UID
Servo	PWM	Điều khiển góc quay
HC-SR04	Trigger/Echo	Đo khoảng cách

ESP32 ↔ Server	HTTP	Gửi dữ liệu
ESP32 ↔ Web	WebSocket	Real-time

Luồng dữ liệu tổng quát:

1. HC-SR04 phát hiện xe → báo ESP32.
2. ESP32 đọc UID → chụp ảnh biển số.
3. Gửi HTTP POST (Ảnh + UID) lên Server.
4. Server chạy YOLOv8 + OCR → lấy biển số.
5. Kiểm tra biển số + UID trùng hợp → cấp quyền.
6. Server trả lệnh → ESP32 điều khiển servo mở barrier.
7. Xe đi qua → cảm biến báo → ESP32 đóng barrier.
8. WebSocket gửi log ra/vào real-time lên ReactJS.

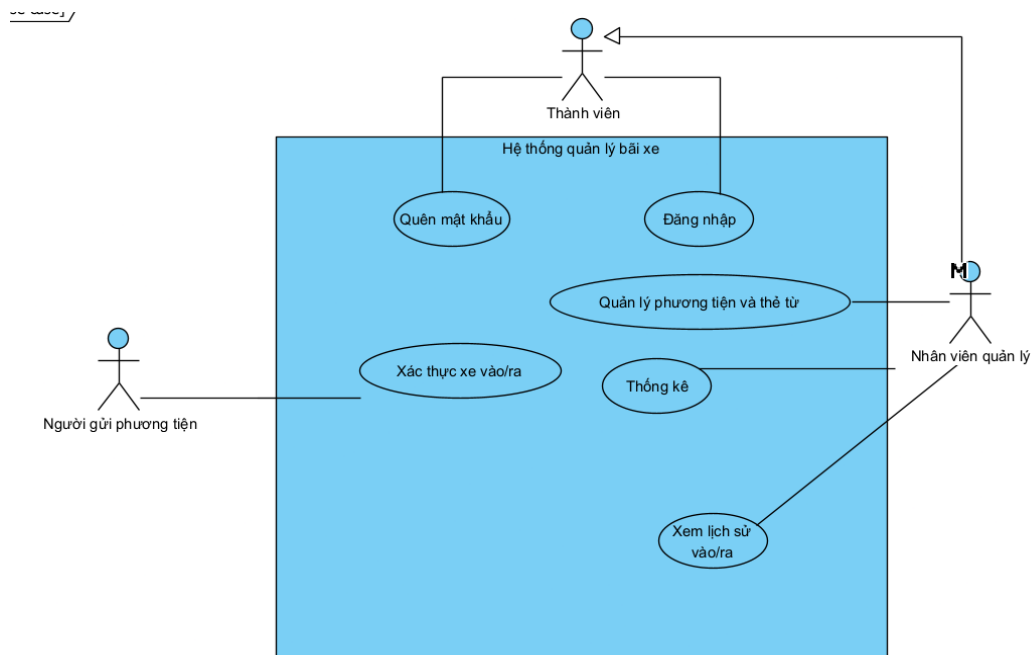
Phần 3: Phân tích hệ thống

3.1. Biểu đồ usecase hệ thống

3.1.1. Biểu đồ usecase tổng quan

- Xác định actor
 - Thành viên
 - Usecase đăng ký : cho phép thành viên đăng ký tài khoản
 - Usecase đăng nhập: cho phép thành viên đăng nhập vào website bằng tài khoản và mật khẩu
 - Nhân viên quản lý :
 - Kế thừa từ thành viên
 - Usecase quản lý phương tiện và thẻ từ : cho phép nhân viên quản lý thực hiện các thao tác thêm, sửa, xóa phương tiện và thẻ từ

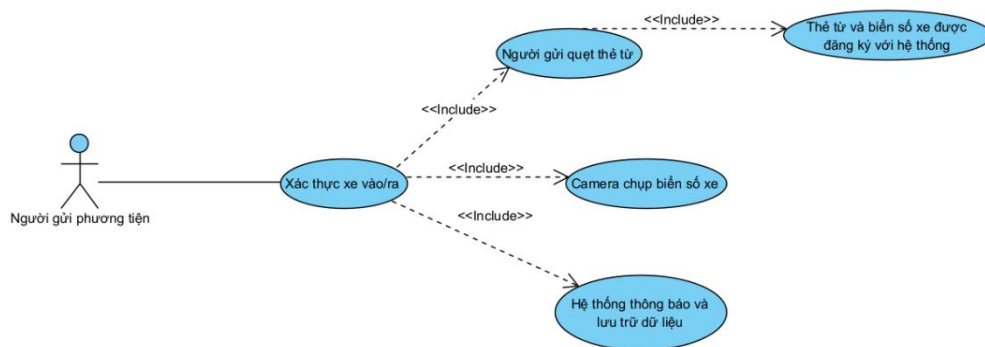
- Usecase thống kê: cho phép nhân viên quản lý xem các thống kê về lượt vào, lượt ra, số lượng xe, doanh thu
- Usecase xem lịch sử vào/ra: cho phép nhân viên quản lý xem lịch sử vào/ra của một phương tiện
- Người gửi phương tiện
 - Usecase xác thực xe vào/ra: cho phép người gửi phương tiện sử dụng thẻ từ để xác thực phương tiện



Hình 3. 1: Usecase tổng quan

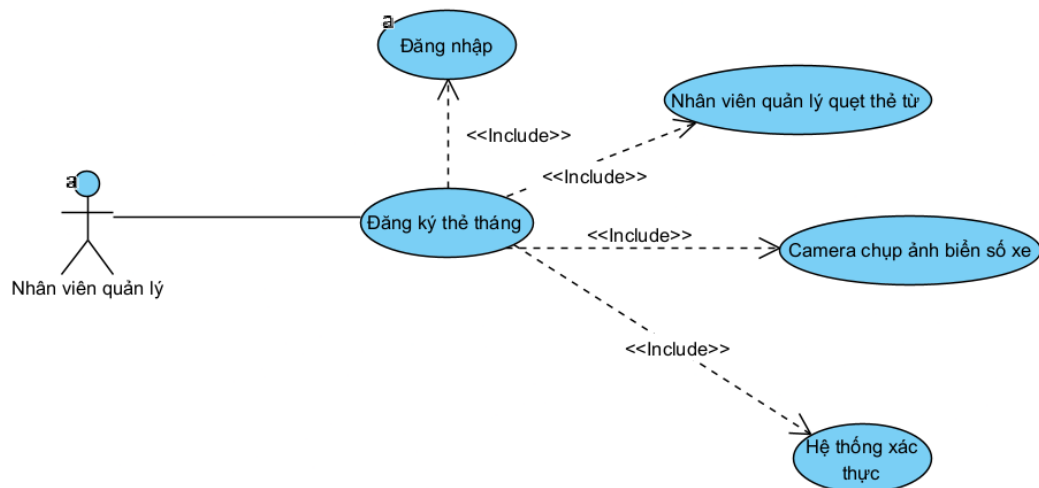
3.1.2. Biểu đồ usecase chi tiết

3.1.2.1. Biểu đồ usecase chức năng xác thực vào/ra



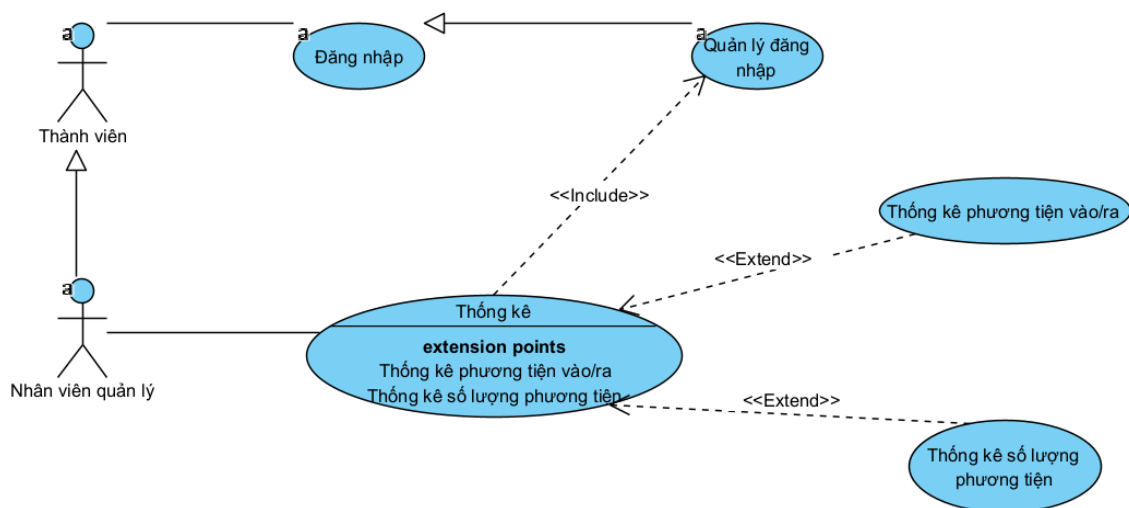
Hình 3. 2: Usecase xác thực phương tiện vào/ra

3.1.2.2. Biểu đồ usecase chức năng đăng ký thẻ tháng



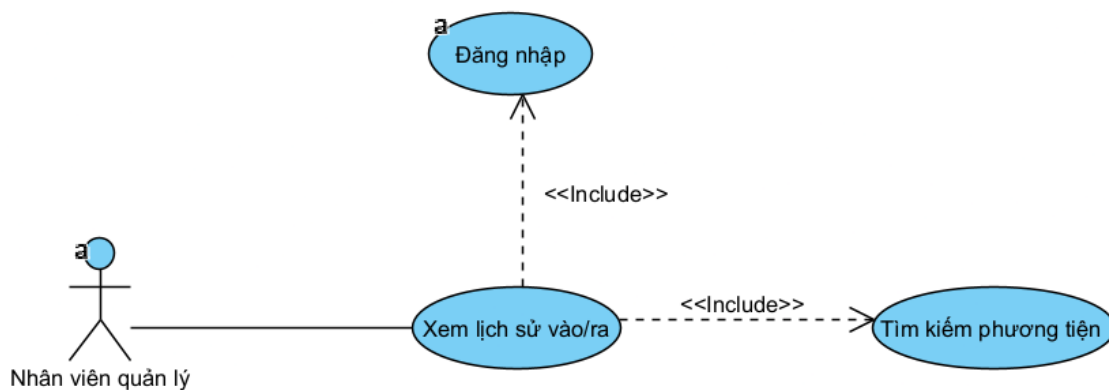
Hình 3. 3: Usecase đăng ký thẻ tháng

3.1.2.3. Biểu đồ usecase chức năng thống kê



Hình 3. 4: Usecase thống kê

3.1.2.4. Biểu đồ usecase chức năng xem lịch sử vào/ra



Hình 3. 5: Usecase xem lịch sử vào/ra

3.2. Kịch bản các usecase

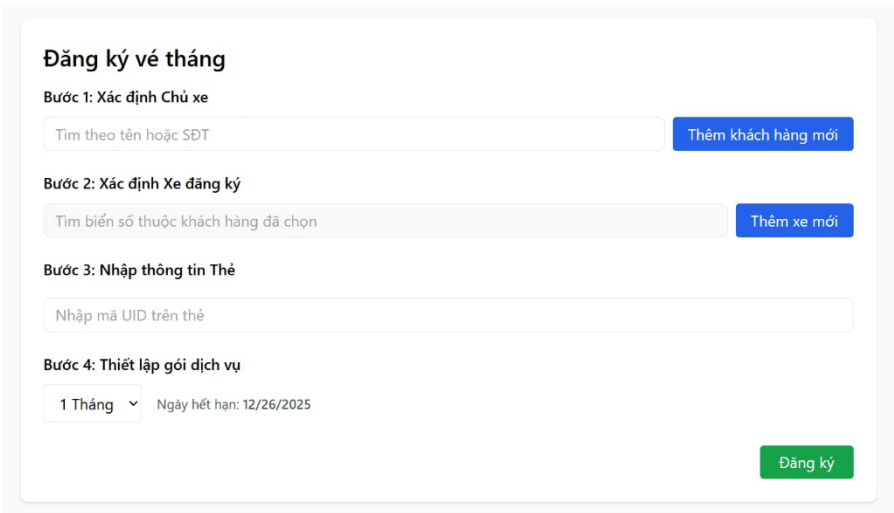
3.2.1. Chức năng giám sát luồng vào/ra

Scenario	Use case Xác thực xe vào/ra
Actor	Người gửi phương tiện

Tiền điều kiện	<ul style="list-style-type: none"> - Vé tháng: thẻ và phương tiện đã được đăng ký trên hệ thống - Vé ngày: thẻ đã được đăng ký trên hệ thống
Hậu điều kiện	Rào chắn mở cho người gửi phương tiện
Kịch bản chính	<ol style="list-style-type: none"> 1. Người dùng di chuyển phương tiện vào khu vực có camera và đầu đọc thẻ, thực hiện quét thẻ 2. RC522 đọc dữ liệu từ thẻ RFID , gửi dữ liệu đến ESP32-CAM, ESP32-CAM thực hiện chụp biển số xe và gửi dữ liệu gồm ảnh, mã thẻ đến hệ thống 3. Hệ thống nhận yêu cầu từ ESP32-CAM chuyển tiếp ảnh đến AI server 4. AI server xử lý ảnh và trả kết quả về cho hệ thống 5. Hệ thống xử lý logic và trả kết quả về cho ESP32-CAM, lưu trữ dữ liệu vào database và hiển thị thông tin ra màn hình 6. ESP32-CAM nhận kết quả, gửi tín hiệu đến động cơ servo 7. Động cơ servo nâng thanh barrier cho người dùng
Ngoại lệ	<ol style="list-style-type: none"> 4.1. AI không xử lý được ảnh biển số và trả về lỗi 5.1. Thẻ chưa được đăng ký với hệ thống 5.2. Biển số chưa được đăng ký với hệ thống 5.3. Biển số và thẻ không khớp

3.2.2. Chức năng đăng ký thẻ tháng

Scenario	Use case Đăng ký thẻ tháng
Actor	Nhân viên quản lý

Tiền điều kiện	<ul style="list-style-type: none"> - Thẻ chưa đăng ký với hệ thống - Xe chưa đăng ký với hệ thống - Nhân viên quản lý đã đăng nhập
Hậu điều kiện	Nhân viên quản lý đăng ký thẻ thành công
Kịch bản chính	<ol style="list-style-type: none"> 1. Nhân viên quản lý click chọn đăng ký thẻ tháng trên màn hình 2. Hệ thống hiển thị giao diện đăng ký thẻ tháng  <ol style="list-style-type: none"> 3. Nhân viên quản lý quét thẻ 4. RC522 đọc thẻ từ và gửi mã thẻ đến ESP32-CAM 5. ESP32-CAM gửi mã thẻ đến hệ thống 6. Hệ thống hiển thị mã thẻ 7. Nhân viên quản lý nhập biển số xe, chủ xe, chọn ngày đăng ký, ngày hết hạn và click lưu 8. Hệ thống hiển thị thông báo thành công
Ngoại lệ	<ol style="list-style-type: none"> 7.1. Thẻ không hợp lệ 7.2. Nhân viên không nhập biển số xe 7.3. Nhân viên không chọn ngày đăng ký, ngày hết hạn

3.2.3. Chức năng thống kê

Scenario	Use case Thống kê
Actor	Nhân viên quản lý
Tiền điều kiện	- Nhân viên quản lý đã đăng nhập
Hậu điều kiện	Nhân viên quản lý xem thống kê thành công
Kịch bản chính	<ol style="list-style-type: none">1. Nhân viên quản lý click chọn chức năng thống kê2. Hệ thống hiển thị giao diện thống kê gồm các nút<ul style="list-style-type: none">+ Thống kê số lượng xe+ Thống kê lượt vào/ra+ Thống kê doanh thu3. Nhân viên quản lý click chọn thống kê số lượng xe4. Hệ thống hiển thị giao diện thống kê số lượng xe
Ngoại lệ	

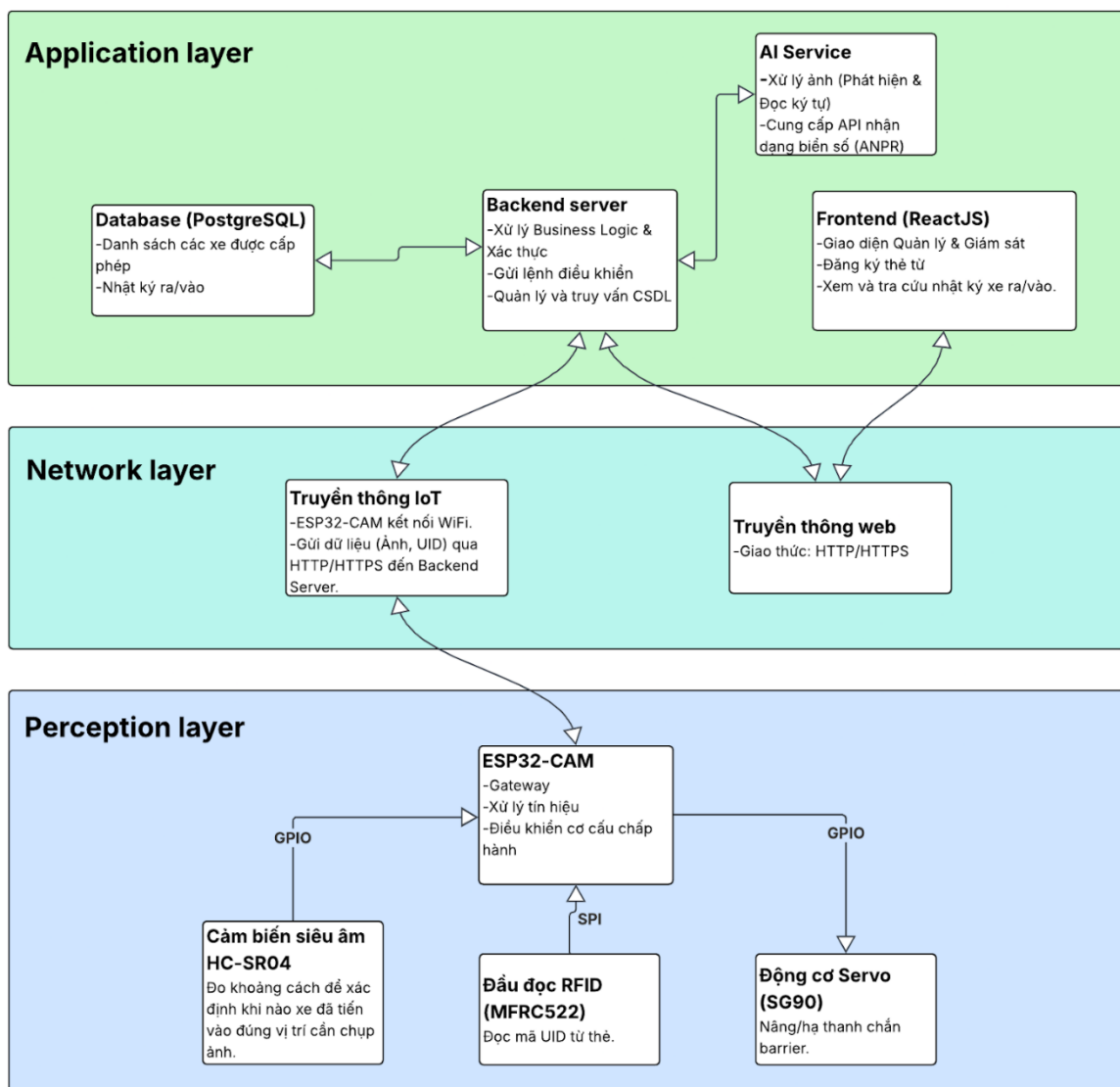
3.2.4. Chức năng xem lịch sử vào/ra

Scenario	Use case Xem lịch sử vào/ra
Actor	Nhân viên quản lý
Tiền điều kiện	- Nhân viên quản lý đã đăng nhập
Hậu điều kiện	Nhân viên quản lý xem lịch sử vào ra của một phương tiện

Kịch bản chính	<ol style="list-style-type: none"> 1. Nhân viên quản lý click chọn chức năng xem lịch sử vào/ra 2. Hệ thống hiển thị giao diện gồm thanh tìm kiếm và nút tìm kiếm 3. Nhân viên quản lý nhập biển số xe và click tìm kiếm 4. Hệ thống hiển thị danh sách các lần vào/ra của phương tiện 5. Nhân viên click vào một phần tử của danh sách 6. Hệ thống hiển thị lượt vào/ra , ảnh chụp, thời gian vào/ra của lần đó
Ngoại lệ	3.1. Xe không có trong hệ thống

Phần 4: THIẾT KẾ HỆ THỐNG

4.1. Kiến trúc tổng quan hệ thống



Hình 4. 1: Kiến trúc tổng quan hệ thống

Hệ thống được thiết kế dựa trên mô hình kiến trúc IoT 3 lớp tiêu chuẩn, đảm bảo sự tách biệt rõ ràng giữa các thành phần phần cứng, truyền thông và xử lý nghiệp vụ. Cụ thể các lớp bao gồm:

4.1.1. Lớp nhận thức (Perception Layer) Đây là lớp vật lý tiếp xúc trực tiếp với môi trường thực tế, chịu trách nhiệm thu thập dữ liệu và điều khiển các cơ cấu chấp hành.

- ESP32-CAM: Đóng vai trò là vi xử lý trung tâm (Gateway) của thiết bị biên. Nó chịu trách nhiệm thu thập tín hiệu từ các cảm biến, xử lý hình ảnh sơ bộ và gửi dữ liệu lên Server.
- Đầu đọc RFID (MFRC522): Giao tiếp với ESP32 qua chuẩn SPI, dùng để đọc mã UID từ thẻ từ của người dùng nhằm xác thực quyền ra vào.
- Cảm biến siêu âm HC-SR04: Kết nối qua chân GPIO, có nhiệm vụ đo khoảng cách để phát hiện xe đã tiến vào vị trí quy định, từ đó kích hoạt camera chụp ảnh.
- Động cơ Servo (SG90): Đóng vai trò cơ cấu chấp hành, nhận lệnh điều khiển (PWM) từ ESP32 qua chân GPIO để thực hiện đóng/mở thanh chắn barrier.

4.1.2. Lớp mạng (Network Layer) Lớp mạng đóng vai trò cầu nối trung gian, đảm bảo việc truyền tải dữ liệu tin cậy giữa các thiết bị ở lớp nhận thức và hệ thống máy chủ ở lớp ứng dụng.

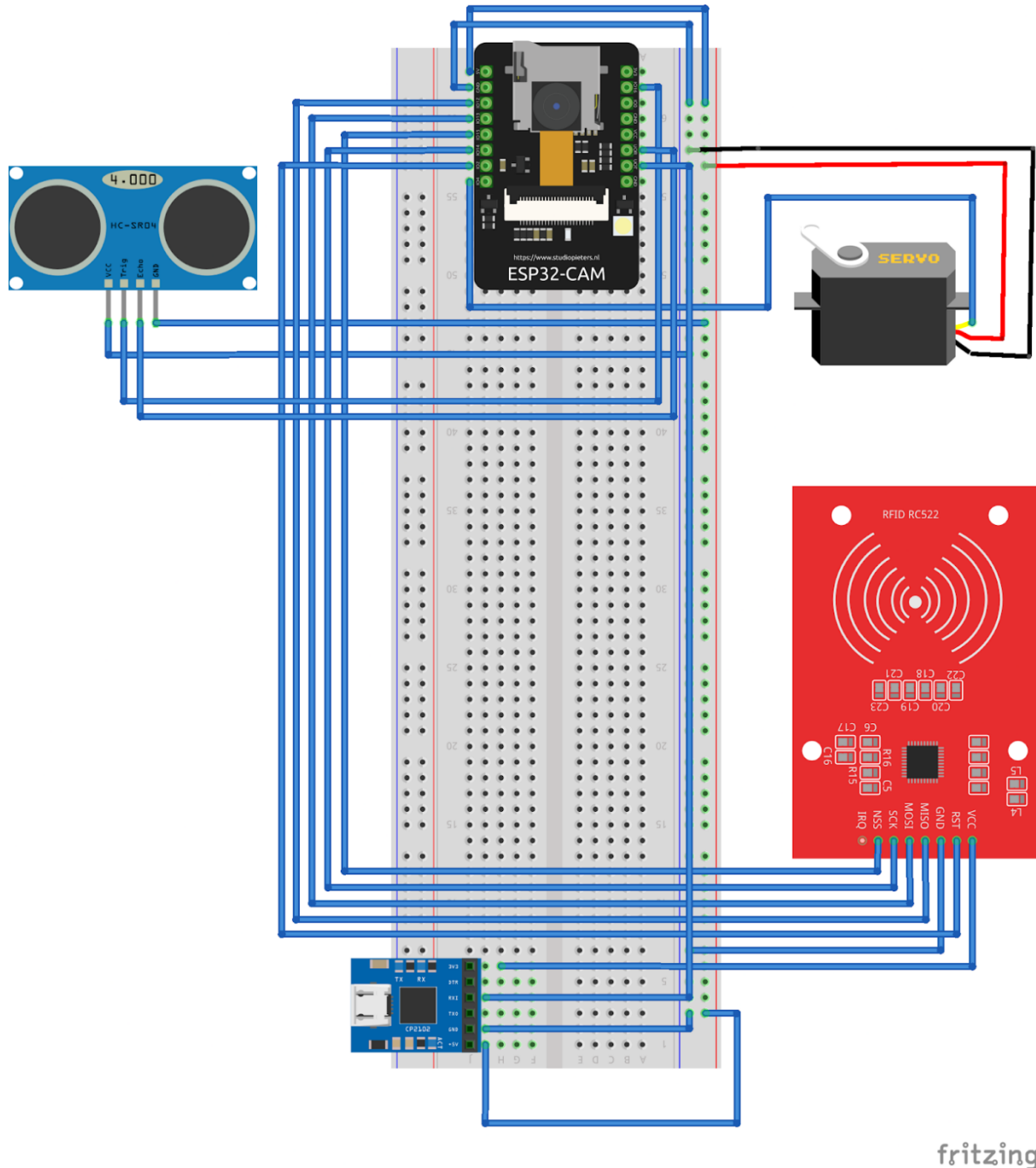
- Truyền thông IoT: ESP32-CAM kết nối Internet thông qua WiFi. Dữ liệu (ảnh chụp biển số, mã UID thẻ, trạng thái cảm biến) được đóng gói và gửi đi thông qua giao thức HTTP/HTTPS đến Backend Server.
- Truyền thông Web: Đảm bảo kết nối giữa giao diện người dùng (Frontend) và máy chủ (Backend), cho phép người quản trị giám sát hệ thống từ xa theo thời gian thực.

4.1.3. Lớp ứng dụng (Application Layer) Đây là trung tâm xử lý dữ liệu, lưu trữ và tương tác với người dùng.

- Backend Server: Là bộ não của hệ thống, chịu trách nhiệm xử lý các Business Logic (nghiệp vụ), xác thực thông tin thẻ từ lớp dưới gửi lên, và ra quyết định điều khiển (gửi lệnh mở barrier).
- Database (PostgreSQL): Hệ quản trị cơ sở dữ liệu quan hệ, dùng để lưu trữ danh sách thẻ được cấp phép, thông tin người dùng và nhật ký (logs) xe ra vào chi tiết.
- AI Service: Module tích hợp trí tuệ nhân tạo, cung cấp API để nhận diện biển số xe (ANPR) từ hình ảnh do ESP32-CAM gửi về, hỗ trợ việc đối chiếu và tăng cường bảo mật.

- Frontend (ReactJS): Cung cấp giao diện trực quan cho người quản trị. Tại đây, người dùng có thể đăng ký thẻ mới, xem camera giám sát thời gian thực và tra cứu lịch sử ra vào.

4.2. Thiết kế phần cứng (Hardware Design)



Hình 4. 2: Sơ đồ mạch

4.2.1. Tổng quan

Hệ thống sử dụng ESP32-CAM làm vi điều khiển trung tâm, chịu trách nhiệm thu thập dữ liệu từ các cảm biến, xử lý logic điều khiển và giao tiếp với máy tính. Toàn bộ mạch được lắp ráp trên Breadboard, sử dụng nguồn điện 5V DC được cung cấp thông qua module chuyển đổi USB-to-TTL.

4.2.2. Khối Nguồn (Power Supply)

- Hệ thống nhận nguồn 5V từ cổng USB máy tính thông qua module USB-to-TTL CP2102.
- Nguồn điện được phân phối qua hai đường ray nguồn (Power Rails) dọc hai bên Breadboard:
 - Đường + (VCC): Cấp 5V cho ESP32-CAM, Động cơ Servo và Cảm biến siêu âm.
 - Đường - (GND): Đóng vai trò nối đất chung (Common Ground) cho toàn bộ hệ thống.
- Lưu ý đặc biệt: Riêng module RFID RC522 sử dụng nguồn 3.3V được lấy trực tiếp từ chân 3V3 của module USB-to-TTL để đảm bảo tương thích điện áp và tránh hư hỏng linh kiện.

4.2.3. Khối Giao tiếp và Điều khiển (Chi tiết kết nối)

a. Module đọc thẻ từ (RFID RC522):

Sử dụng giao thức truyền thông SPI để giao tiếp với ESP32-CAM. Các chân tín hiệu được kết nối vào các GPIO bên trái của vi điều khiển, cụ thể:

- Chân SDA (SS) -> GPIO 15
- Chân SCK -> GPIO 14
- Chân MOSI -> GPIO 13
- Chân MISO -> GPIO 12
- Chân RST -> GPIO 2

b. Cảm biến siêu âm (HC-SR04):

Dùng để đo khoảng cách xe ra vào.

- Chân Trig (Phát xung) -> Kết nối với GPIO 16.

- Chân Echo (Thu xung) -> Kết nối với GPIO 3 (U0R).
 - *Cơ chế hoạt động:* GPIO 3 đồng thời là chân nhận dữ liệu nạp code (RX). Do đó, trong quá trình vận hành thực tế, tín hiệu Echo sẽ được ưu tiên. Khi cần nạp chương trình, kết nối này cần được ngắt tạm thời.

c. Động cơ Servo (SG90):

Đóng vai trò là thanh chắn (Barrier).

- Chân tín hiệu (Signal - Dây cam) -> Kết nối với GPIO 4.
- Vi điều khiển xuất xung PWM qua GPIO 4 để điều khiển góc quay của Servo (đóng/mở 90 độ).

d. Khối Nạp chương trình và Giao tiếp Serial (USB-to-TTL):

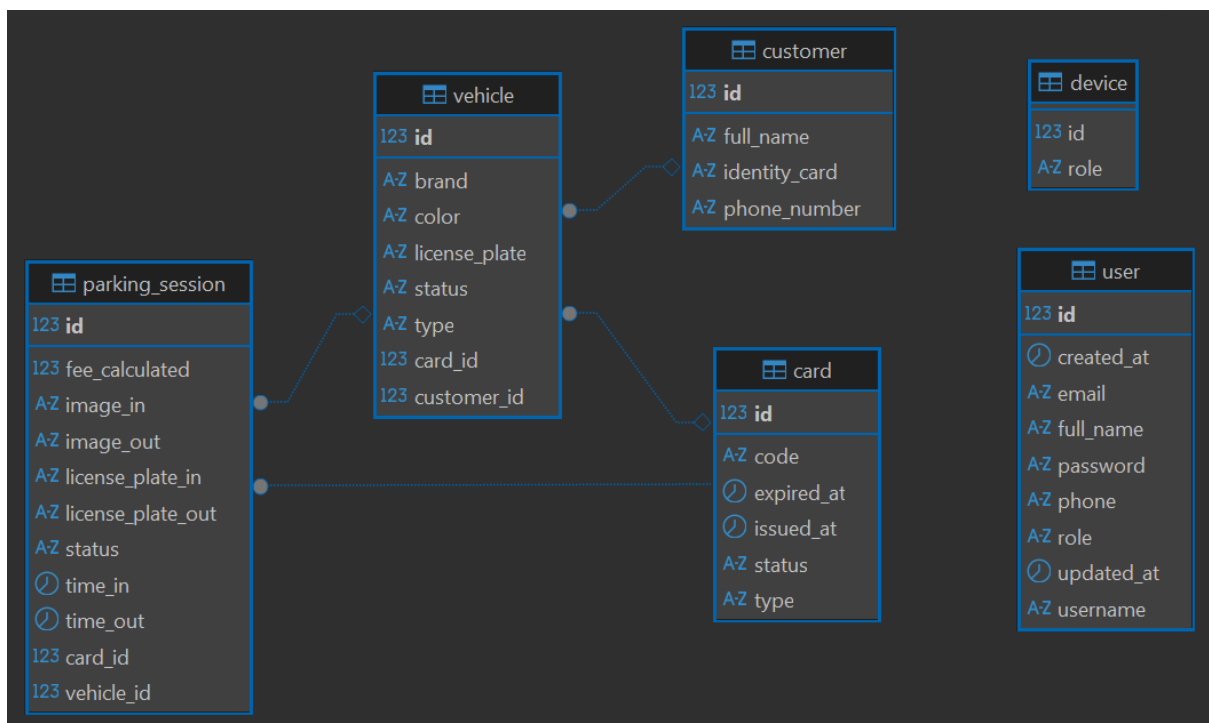
Module này đóng vai trò cầu nối UART giữa ESP32 và máy tính.

- TXD (USB) -> U0R / GPIO 3 (ESP32): Truyền dữ liệu nạp xuống chip.
- RXD (USB) -> U0T / GPIO 1 (ESP32): Nhận dữ liệu báo cáo từ chip lên Serial Monitor.
- Chế độ Bootloader: Chân GPIO 0 được thiết kế để nối xuống GND thông qua một dây jumper (hoặc nút nhấn) nhằm kích hoạt chế độ nạp (Flash Mode) khi khởi động.

GPIO	Chân thiết bị	Mô tả
GPIO 4	Signal (Servo)	Điều khiển xung PWM để xoay góc động cơ Servo.
GPIO 16	Trig (SRF05)	Kích hoạt cảm biến phát ra sóng siêu âm đo khoảng cách.
GPIO 3	Echo (SRF05) (*)	Nhận tín hiệu phản hồi siêu âm (Lưu ý: Dùng chung chân nạp code).
GPIO 15	SDA (RC522)	Chân chọn thiết bị (SS/CS) cho giao tiếp SPI của RFID.

GPIO 14	SCK (RC522)	Truyền tín hiệu xung nhịp đồng bộ dữ liệu (SPI Clock).
GPIO 13	MOSI (RC522)	Truyền dữ liệu từ ESP32 sang mạch RFID (Master Out Slave In).
GPIO 12	MISO (RC522)	Nhận dữ liệu từ mạch RFID về ESP32 (Master In Slave Out).
GPIO 2	RST (RC522)	Chân Reset để khởi động lại mạch đọc thẻ RFID.
GPIO 1	RXD (USB-TTL)	Truyền dữ liệu Serial từ ESP32 lên máy tính (Serial Monitor).
GPIO 0	GND (Switch)	Chân chọn chế độ: Nối đất (GND) để nạp code, bỏ trống để chạy.

4.3. Thiết kế Cơ sở dữ liệu



Hình 4. 3: Cơ sở dữ liệu

Hệ thống sử dụng cơ sở dữ liệu quan hệ (Relational Database) để quản lý thông tin toàn vẹn về người dùng, phương tiện, thẻ từ và lịch sử ra vào. Sơ đồ bao gồm 6 bảng dữ liệu chính với chức năng cụ thể như sau:

1. Bảng quản lý hệ thống và thiết bị

- Bảng user (Người dùng hệ thống):
 - Chức năng: Lưu trữ thông tin của các quản trị viên (Admin) hoặc nhân viên bảo vệ vận hành hệ thống.
 - Các trường chính: username, password (để đăng nhập), full_name, role (phân quyền quản trị hay nhân viên), created_at/updated_at (thời gian tạo/cập nhật).
- Bảng device (Thiết bị):
 - Chức năng: Quản lý danh sách các thiết bị phân cứng trong hệ thống (như ESP32-CAM tại cổng vào/ra, đầu đọc thẻ RC522 vào/ra).
 - Các trường chính: id, role (xác định thiết bị đặt tại cổng vào hay cổng ra).

2. Bảng quản lý khách hàng và phương tiện (Core Business)

- Bảng customer (Khách hàng):
 - Chức năng: Lưu trữ thông tin cá nhân của chủ phương tiện (thường dùng cho khách đăng ký vé tháng).
 - Các trường chính: full_name, identity_card (CMND/CCCD), phone_number.
- Bảng card (Thẻ từ RFID):
 - Chức năng: Quản lý các thẻ từ được phát hành trong hệ thống.
 - Các trường chính:
 - code: Mã UID duy nhất của thẻ RFID (dùng để đối chiếu khi quét thẻ).
 - type: Loại thẻ (Thẻ tháng/Thẻ lượt).

- expired_at: Ngày hết hạn (quan trọng cho logic kiểm tra thẻ tháng).
- status: Trạng thái thẻ (Hoạt động/Bị khóa).
- Bảng vehicle (Phương tiện):
 - Chức năng: Lưu trữ thông tin xe đăng ký cố định. Đây là bảng trung tâm liên kết giữa Khách hàng và Thẻ.
 - Các trường chính: license_plate (Biển số đăng ký), brand (Hãng xe), color, type (Xe máy/Ô tô).
 - Liên kết:
 - Có khóa ngoại customer_id trỏ đến bảng customer (Một khách có thể sở hữu xe).
 - Có khóa ngoại card_id trỏ đến bảng card (Xe được gán với một thẻ từ cụ thể).

3. Bảng nghiệp vụ (Transaction)

- Bảng parking_session (Phiên gửi xe/Nhập ký ra vào):
 - Chức năng: Đây là bảng quan trọng nhất, lưu trữ lịch sử mỗi lần xe vào và ra khỏi bãi. Mỗi dòng dữ liệu tương ứng với một lượt gửi xe hoàn chỉnh.
 - Các trường chính:
 - Thông tin vào: time_in (Giờ vào), image_in (Đường dẫn ảnh chụp lúc vào), license_plate_in (Biển số AI nhận diện lúc vào).
 - Thông tin ra: time_out (Giờ ra), image_out (Ảnh chụp lúc ra), license_plate_out (Biển số AI nhận diện lúc ra).
 - Thanh toán: fee_calculated (Phí gửi xe được tính toán dựa trên loại thẻ và thời gian gửi).
 - Trạng thái: status (Đang gửi/Đã trả xe).
 - Liên kết:
 - card_id: Xác định thẻ nào đã được sử dụng trong phiên này.

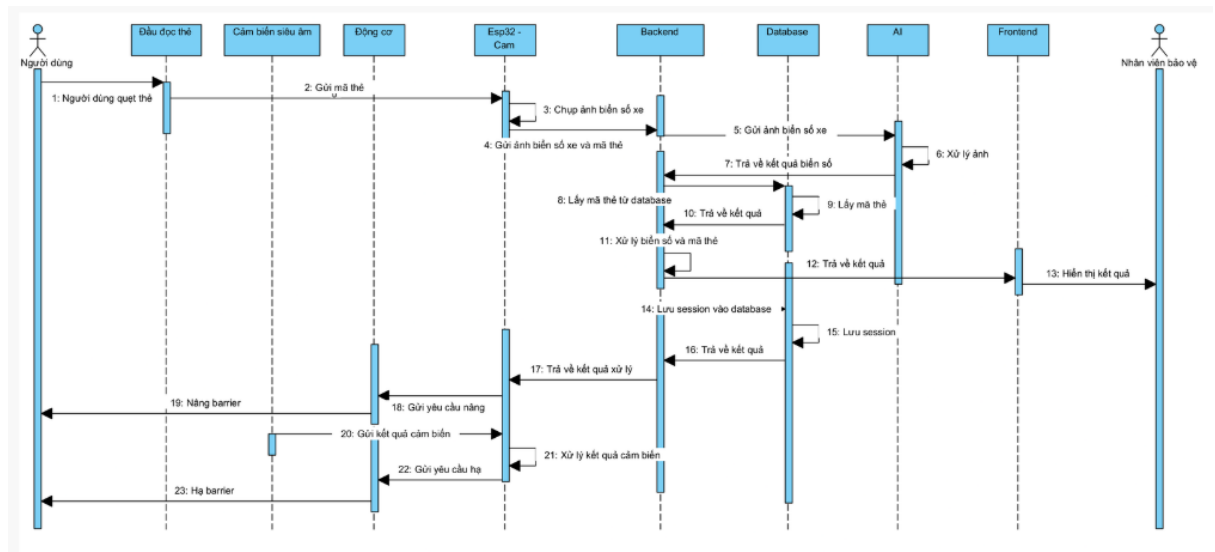
- `vehicle_id`: Xác định xe nào (nếu là xe đăng ký vé tháng).

Mô tả Mối quan hệ (Relationships) & Luồng dữ liệu:

1. Quan hệ 1-N (Một - Nhiều) giữa `customer` và `vehicle`: Một khách hàng có thể sở hữu nhiều phương tiện, nhưng một phương tiện chỉ thuộc về một khách hàng (trong hệ thống).
2. Quan hệ giữa `vehicle` và `card`: Mỗi phương tiện đăng ký vé tháng sẽ được cấp phát (map) với một thẻ RFID cụ thể thông qua `card_id`.
3. Quan hệ nghiệp vụ trong `parking_session`:
 - Khi xe vào: Hệ thống tạo một bản ghi mới trong `parking_session`, lưu `time_in`, `image_in` và liên kết với `card_id` vừa quét.
 - Khi xe ra: Hệ thống tìm bản ghi có `card_id` tương ứng mà chưa hoàn thành (status chưa kết thúc), sau đó cập nhật `time_out`, `image_out` và tính tiền `fee_calculated`.

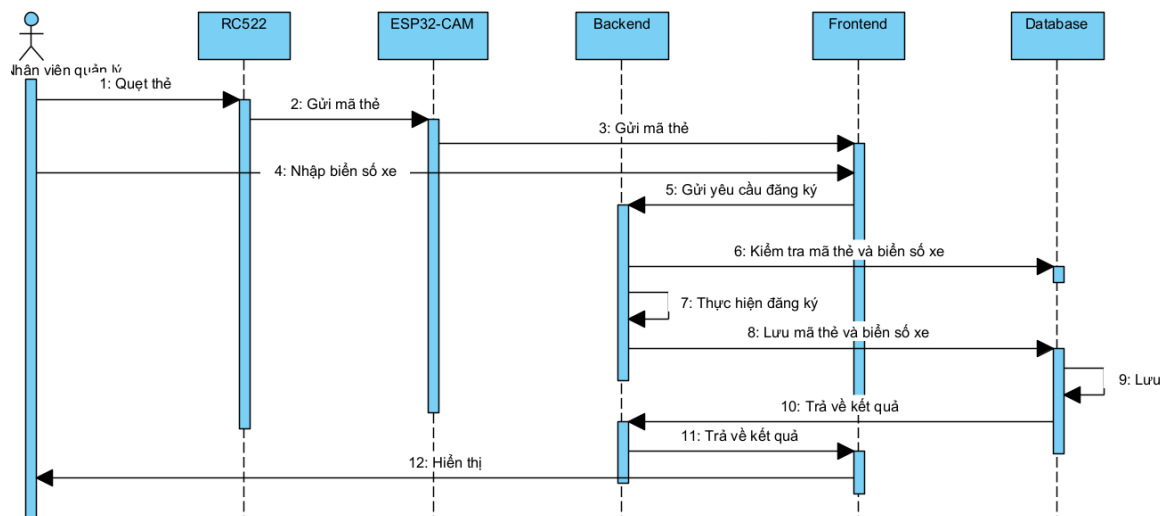
4.4. Biểu đồ tuần tự

4.4.1. Chức năng xác thực luồng vào/ra



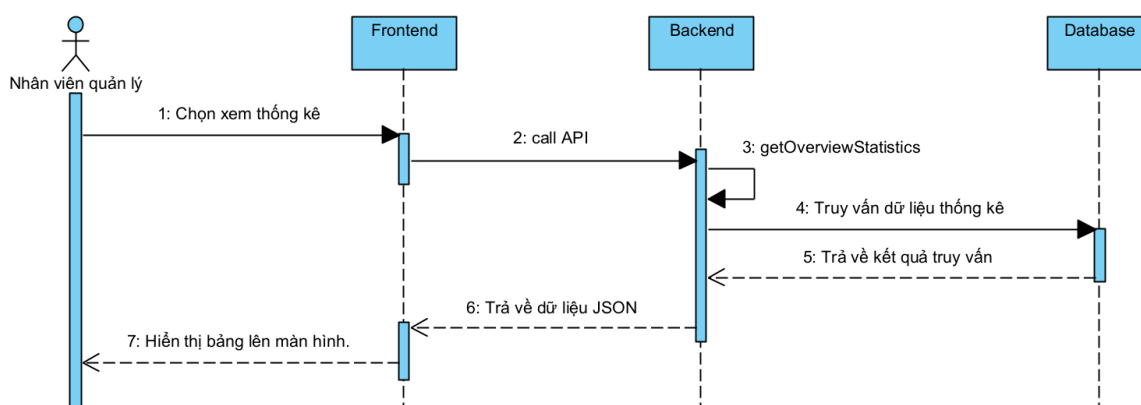
Hình 4. 4: Sequence diagram xác thực phương tiện vào/ra

4.2.2. Chức năng đăng ký thẻ tháng



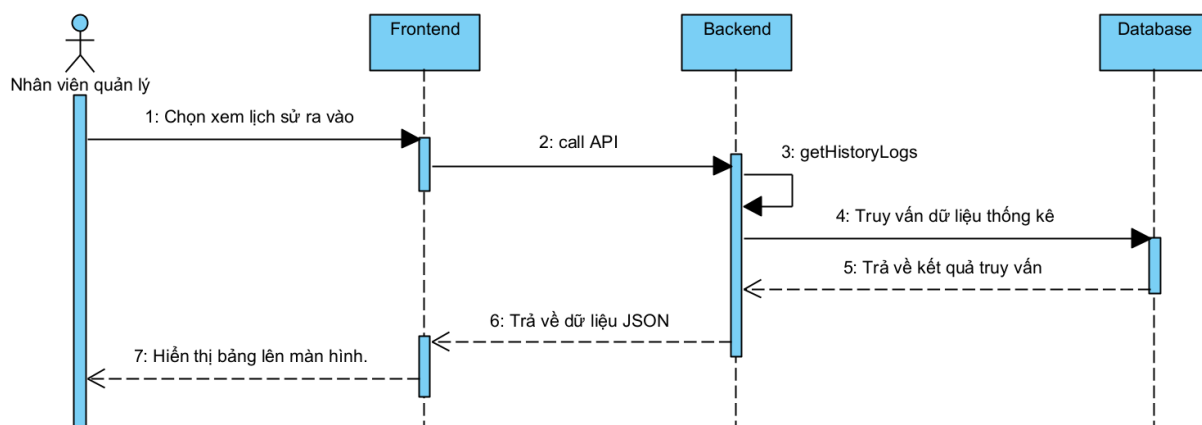
Hình 4. 5: Sequence diagram đăng ký thẻ tháng

4.4.3. Chức năng thống kê



Hình 4. 6: Sequence diagram chức năng thống kê

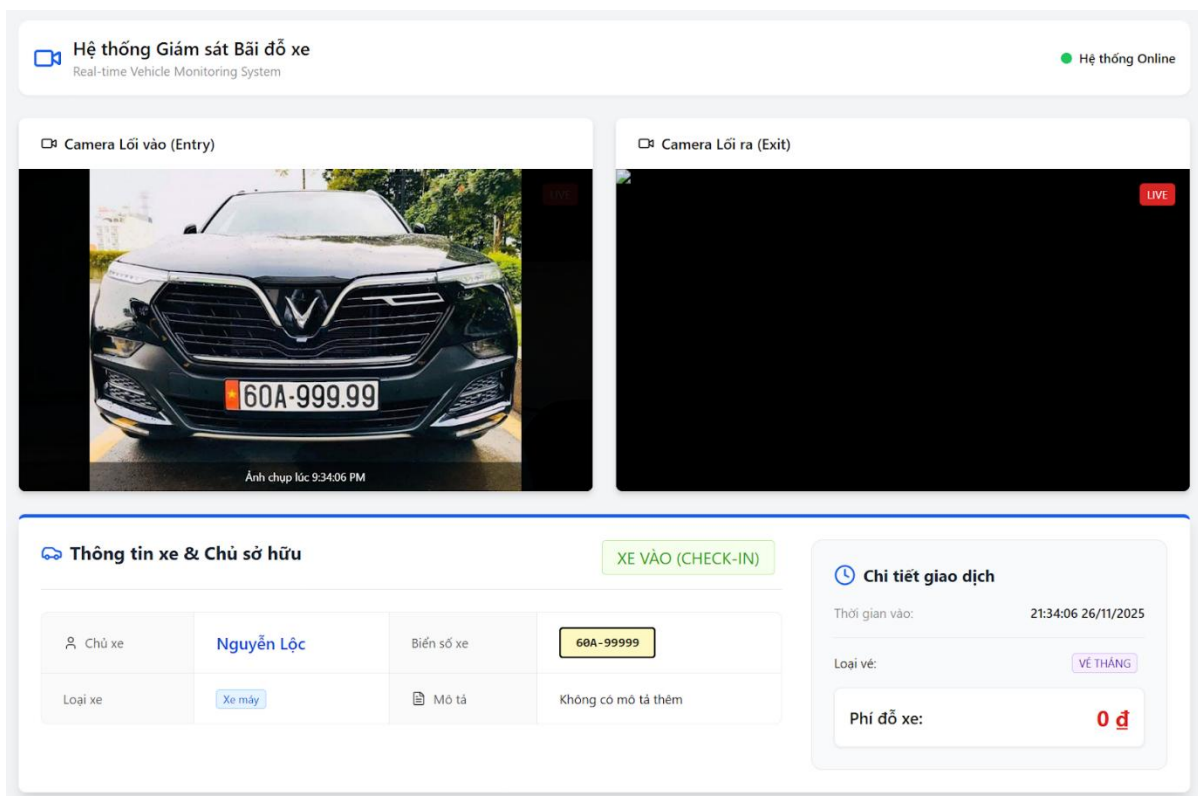
4.4.4. Chức năng xem lịch sử vào/ra



Hình 4. 7: Sequence diagram chức năng xem lịch sử vào/ra

4.5. Giao diện ứng dụng

4.5.1. Giao diện giám sát lối vào (Check-in Dashboard)



Hình 4. 8: Giao diện luồng vào

- Mô tả giao diện:

- Màn hình được chia làm hai phần chính: Khu vực hiển thị camera thời gian thực (Camera Lối vào/Lối ra) và Khu vực thông tin chi tiết.
- Khi cảm biến phát hiện xe tại lối vào, hệ thống tự động chụp ảnh và hiển thị tại khung "Camera Lối vào".
- Chức năng hoạt động:
 - Hệ thống nhận diện biển số xe từ hình ảnh camera và truy vấn cơ sở dữ liệu.
 - Nếu là xe đã đăng ký, hệ thống tự động hiển thị tên chủ xe, loại xe và biển số.
 - Trạng thái giao dịch được đánh dấu là "XE VÀO (CHECK-IN)" màu xanh.
 - Thời gian vào (time_in) được ghi nhận theo thời gian thực, phí đỗ xe khởi tạo là 0đ.

4.5.2. Giao diện giám sát lối ra (Check-out Dashboard)

The screenshot displays the 'Hệ thống Giám sát Bãi đỗ xe' (Real-time Vehicle Monitoring System) interface. It features two camera feeds at the top: 'Camera Lối vào (Entry)' and 'Camera Lối ra (Exit)'. Both show a black car with license plate '60A-999.99'. Below the feeds, there's a section for 'Thông tin xe & Chủ sở hữu' (Vehicle & Owner Information) and a 'XE RA (CHECK-OUT)' button. The vehicle information table lists the owner as 'Nguyễn Lộc', the license plate as '60A-999.99', and the vehicle type as 'Ô tô'. To the right, the 'Chi tiết giao dịch' (Transaction Details) section shows the entry time as 21:34:06 on 26/11/2025, the exit time as 22:37:11 on 26/11/2025, the ticket type as 'VÉ THƯỜNG', and the parking fee as 3000 đ.

Thông tin xe & Chủ sở hữu			
Chủ xe	Nguyễn Lộc	Biển số xe	60A-999.99
Loại xe	Ô tô	Mô tả	Không có mô tả thêm

Chi tiết giao dịch	
Thời gian vào:	21:34:06 26/11/2025
Thời gian ra:	22:37:11 26/11/2025
Loại vé:	VÉ THƯỜNG
Phí đỗ xe:	3000 đ

Hình 4. 9: Giao diện luồng ra

- Mô tả giao diện:

- Tương tự giao diện Check-in, nhưng thông tin tập trung vào sự kiện xe ra.
- Hệ thống hiển thị hình ảnh xe tại thời điểm ra để bảo vệ đối chiếu.
- Chức năng hoạt động:
 - Khi xe quét thẻ ra, hệ thống tìm kiếm phiên gửi xe tương ứng.
 - Trạng thái chuyển sang "XE RA (CHECK-OUT)" màu cam.
 - Hệ thống tính toán thời gian đỗ xe và tự động tính phí dựa trên loại vé.
 - Trong hình minh họa, hệ thống nhận diện đây là "VÉ THÁNG" và hiển thị mức phí cần thu (hoặc trừ vào tài khoản) là 3.000đ (hoặc hiển thị 0đ tùy theo cấu hình vé tháng).
 - Barie sẽ chỉ mở khi nhân viên xác nhận thông tin biển số và hình ảnh trùng khớp.

4.5.3. Giao diện đăng ký vé tháng (Monthly Ticket Registration)

Đăng ký vé tháng

Bước 1: Xác định Chủ xe

Nguyễn Lộc

012345

Bỏ chọn

Bước 2: Xác định Xe đăng ký

29A26469

*

Bỏ chọn

Bước 3: Nhập thông tin Thẻ

Bước 4: Thiết lập gói dịch vụ

1 Tháng

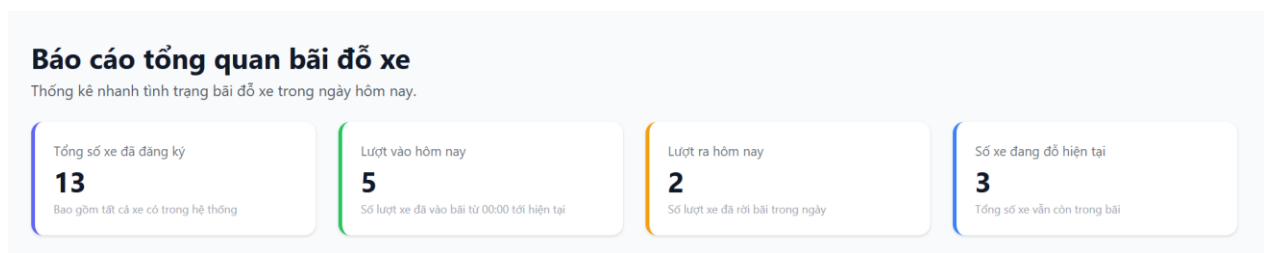
▼

Ngày hết hạn: 12/26/2025

Hình 4. 10: Giao diện đăng ký thẻ tháng

- **Mô tả giao diện:**
 - Đây là giao diện dành cho Quản trị viên (Admin) để cấp phát thẻ mới.
 - Quy trình được thiết kế theo dạng Wizard (từng bước) để tránh sai sót dữ liệu.
- **Chức năng hoạt động:**
 - Bước 1 - Xác định chủ xe: Tìm kiếm thông tin khách hàng trong CSDL hoặc thêm mới nhanh chóng.
 - Bước 2 - Xác định xe đăng ký: Chọn phương tiện thuộc sở hữu của khách hàng đó (Ví dụ: Biển số 29A26469). Điều này đảm bảo tính ràng buộc 1-N giữa Khách và Xe.
 - Bước 3 - Nhập thông tin thẻ: Quét mã UID từ thẻ RFID vào hệ thống để liên kết (map) thẻ với xe.
 - Bước 4 - Thiết lập dịch vụ: Chọn gói dịch vụ (1 tháng, 3 tháng...) và hệ thống tự động tính ngày hết hạn (Expired Date).
 - Sau khi nhấn "Đăng ký", dữ liệu sẽ được lưu vào các bảng card, customer, và vehicle như thiết kế cơ sở dữ liệu.

4.5.4. Giao diện xem thống kê



Hình 4. 11: Giao diện thống kê

- **Mô tả giao diện:**
 - Đây là màn hình chính (Dashboard) mà người quản trị nhìn thấy ngay khi đăng nhập vào hệ thống.
 - Giao diện được thiết kế tối giản dạng các thẻ thông tin (Info Cards), giúp người quản lý nắm bắt nhanh các chỉ số quan trọng mà không cần tra cứu dữ liệu chi tiết.
 - Các chỉ số được phân biệt bằng màu sắc cạnh viền (tím, xanh lá, cam, xanh dương) giúp tăng tính trực quan.

- **Chức năng hoạt động:**

- **Tổng số xe đã đăng ký:** Hệ thống đếm tổng số lượng bản ghi trong bảng vehicle (hoặc card loại vé tháng), cho biết quy mô khách hàng cố định hiện tại của bãi.
- **Lượt vào hôm nay:** Truy vấn từ bảng parking_session, đếm số lượng xe có thời gian vào (time_in) bắt đầu từ 00:00 của ngày hiện tại. Chỉ số này phản ánh lưu lượng xe đến.
- **Lượt ra hôm nay:** Đếm số lượng các phiên gửi xe đã hoàn tất (có time_out) trong ngày. Chỉ số này phản ánh lưu lượng xe đi.
- **Số xe đang đỗ hiện tại:** Đây là chỉ số quan trọng nhất để quản lý sức chứa. Hệ thống đếm số lượng các phiên gửi xe đang có trạng thái "đang gửi" (chưa check-out).

4.5.5. Giao diện xem lịch sử ra vào

Lịch sử ra vào bãi đỗ xe

Quay về Dashboard

Tra cứu chi tiết các lượt xe vào/ra theo thời gian, biển số và trạng thái.

Biển số xe: VD: 29A-123.45 Trạng thái: Tất cả Khoảng thời gian: Start date → End date

Biển số xe	Chủ xe	Trạng thái	Thời gian vào	Thời gian ra
60A-99999	Nguyễn Lộc	Đã rời bãi (OUT)	21:34:06 26/11/2025	21:37:11 26/11/2025

1-1 của 1 bản ghi < 1 > 10 / page

Hình 4. 12: Giao diện xem lịch sử vào/ra

- **Mô tả giao diện:**

- Đây là trang quản lý nhật ký hệ thống, đóng vai trò như một "sổ cái điện tử" ghi lại toàn bộ hoạt động ra vào của bãi xe.
- Giao diện được thiết kế dạng bảng (Table) truyền thống để hiển thị được nhiều thông tin nhất có thể, đi kèm với thanh công cụ tìm kiếm mạnh mẽ ở phía trên.

- **Chức năng hoạt động:**

- **Bộ lọc tìm kiếm đa tiêu chí (Advanced Search):** Người quản trị có thể tra cứu nhanh chóng thông qua các trường:
 - **Biển số xe:** Tìm kiếm chính xác hoặc gần đúng phương tiện cần kiểm tra.

- *Trạng thái*: Lọc theo các xe "Đang gửi" (để kiểm kê) hoặc "Đã rời bãi" (để đối soát).
- *Khoảng thời gian*: Cho phép chọn ngày bắt đầu và ngày kết thúc để xuất báo cáo theo ngày/tuần/tháng.
- **Hiện thị dữ liệu chi tiết**: Bảng dữ liệu liên kết thông tin từ bảng parking_session với customer để hiển thị đầy đủ: Biển số, Tên chủ xe, Trạng thái hiện tại, Thời gian vào và Thời gian ra chính xác đến từng giây.

CHƯƠNG 5: KẾT LUẬN & HƯỚNG PHÁT TRIỂN

5.1. Kết quả đạt được

5.1.1. Về mặt lý thuyết

- Nắm vững kiến thức về kiến trúc hệ thống IoT 3 lớp (Perception, Network, Application) và cách áp dụng vào thực tế.
- Hiểu sâu về cơ chế hoạt động và giao tiếp của các thiết bị phần cứng như ESP32-CAM, module RFID RC522 và động cơ Servo qua các chuẩn giao tiếp SPI, GPIO.
- Làm chủ được các công nghệ truyền thông và xử lý dữ liệu hiện đại: Giao thức HTTP/WebSocket để truyền tải hình ảnh/lệnh điều khiển, và kỹ thuật xử lý ảnh tích hợp AI để nhận diện biển số (ANPR).

5.1.2. Về mặt thực tiễn

- Xây dựng thành công mô hình phần cứng hệ thống bãi giữ xe thông minh hoạt động ổn định với đầy đủ các chức năng: Quẹt thẻ, chụp ảnh, rào chắn tự động và giám sát thời gian thực.
- Hiệu năng ấn tượng: Hệ thống đã tích hợp thành công AI nhận diện biển số xe, giúp tự động hóa quy trình kiểm soát. Thời gian xử lý trung bình từ lúc người dùng quẹt thẻ đến khi hệ thống hoàn tất xác thực và mở thanh chắn (Servo) chỉ mất khoảng 2 giây. Đây là mức độ trễ thấp, đảm bảo trải nghiệm người dùng mượt mà và giảm ùn tắc tại cổng ra vào.
- Giao diện phần mềm (Web App) được thiết kế trực quan, cho phép quản lý thẻ, xem lịch sử ra vào và đối chiếu hình ảnh/biển số xe một cách chính xác.

5.2. Hạn chế

- Phụ thuộc điều kiện môi trường: Khả năng nhận diện biển số của AI có độ chính xác cao vào ban ngày hoặc điều kiện ánh sáng tốt, nhưng bị giảm sút khi trời tối, ngược sáng hoặc khi biển số xe bị mờ, bẩn.
- Phần cứng: ESP32-CAM là thiết bị giá rẻ nên chất lượng cảm biến ảnh chưa cao, đôi khi xảy ra hiện tượng quá nhiệt nếu hoạt động liên tục trong thời gian dài (video streaming), gây ảnh hưởng đến tốc độ khung hình (FPS).
- Kết nối mạng: Hệ thống phụ thuộc hoàn toàn vào sóng WiFi. Nếu tín hiệu mạng không ổn định hoặc mất kết nối, việc gửi dữ liệu lên Server và nhận lệnh điều khiển sẽ bị gián đoạn.

5.3. Hướng phát triển tương lai

5.3.1. Về hệ thống

- Nâng cấp Camera: Thay thế ESP32-CAM bằng các loại IP Camera chuyên dụng để cải thiện chất lượng hình ảnh và khả năng nhận diện AI trong điều kiện thiếu sáng.
- Tích hợp thanh toán điện tử: Bổ sung tính năng thanh toán phí gửi xe tự động qua mã QR (Momo, VNPAY) hoặc ví điện tử ngay trên giao diện Web/App Mobile cho khách hàng.
- Chế độ Offline: Nghiên cứu cơ chế lưu trữ đệm (buffer) trên thiết bị biên (Edge Device) để hệ thống có thể tiếp tục hoạt động cơ bản (mở thẻ bằng mã offline) khi mất kết nối Internet.

5.3.2. Về tính năng thông minh

- Nhận diện không cần thẻ: Phát triển thuật toán AI mạnh hơn để cho phép xe đăng ký vé tháng có thể vào thẳng (barrier tự mở) chỉ dựa trên nhận diện biển số mà không cần thao tác quét thẻ.