



IoT102 PROJECT REPORT

Auto-parking System (APS)

Class Name	SE1953
Group Members	Nguyễn Đăng Khoa Nông Tuấn Anh Trần Minh Anh
Supervisor	Đặng Minh Đức
Ext Supervisor	None

Hanoi, ngày 29 tháng 7 năm 2025

Đề cương Project Report IoT102:

Project: Auto-parking System (APS) - Hệ thống đăng ký đỗ xe thông minh.

Thành viên:

1. Nguyễn Đăng Khoa
2. Nông Tuấn Anh
3. Trần Minh Anh

Mô tả

Auto-parking System (APS) là một hệ thống hỗ trợ quản lý bãi đỗ xe thông minh, được thiết kế nhằm tối ưu hóa quá trình đăng ký, check-in và check-out xe thông qua giao diện web online/captive portal và phần cứng ESP32. Mục tiêu của dự án là tự động hóa việc điều khiển công bãi xe, giúp người dùng dễ dàng đặt chỗ và kiểm soát thời gian vào/ra bằng mã OTP. Hệ thống bao gồm ba thành phần chính: frontend (React), backend (Node.js + MySQL), và module điều khiển phần cứng (ESP32 với cảm biến và servo). Khi người dùng đăng ký slot, hệ thống sẽ cấp một mã OTP; quá trình check-in/check-out sau đó được thực hiện tự động thông qua API và tín hiệu từ cảm biến. Kết quả triển khai cho thấy hệ thống hoạt động ổn định trong môi trường mạng cục bộ, hỗ trợ hiển thị trạng thái slot theo thời gian thực và điều khiển công chính xác theo thao tác người dùng. Dự án này chứng minh khả năng tích hợp giữa phần mềm và phần cứng để giải quyết các vấn đề thực tiễn trong quản lý bãi đỗ xe, giúp giảm bớt nguồn nhân lực thực tế và hỗ trợ bởi hệ thống tự động.

I. Tổng quan dự án

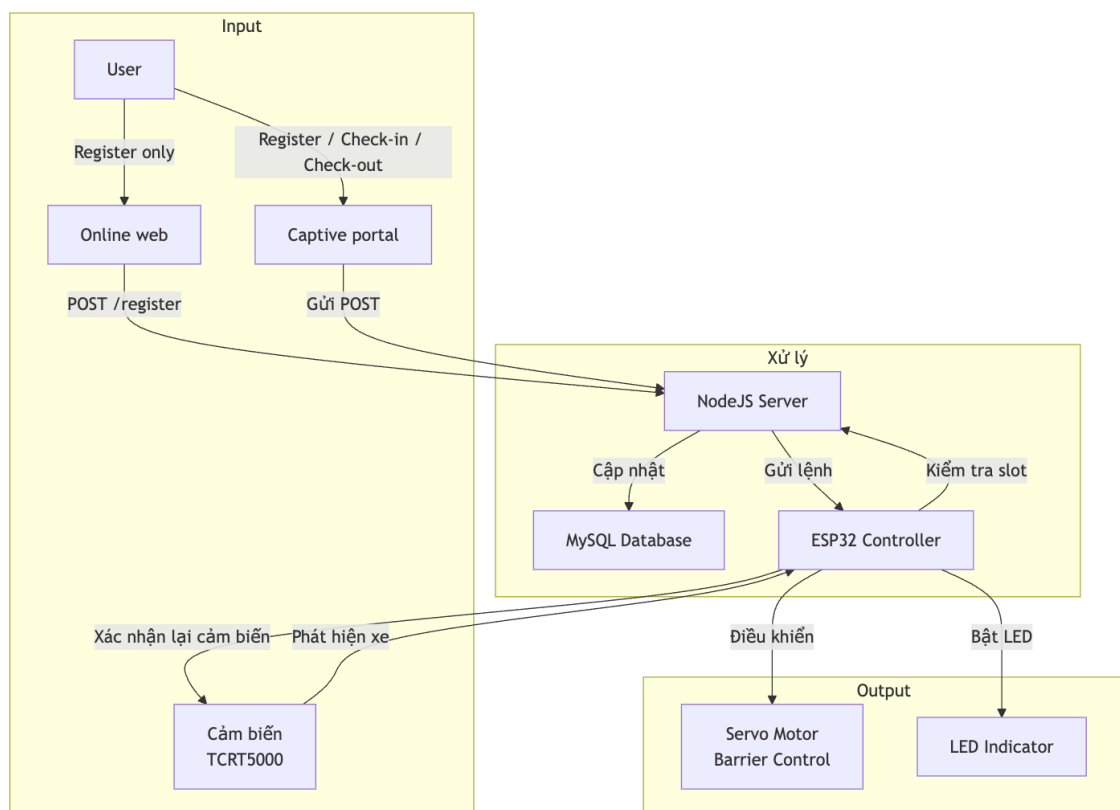
I.1. Lí do lựa chọn đề tài

Hiện nay, nhu cầu đỗ xe tại các khu dân cư, trường học, trung tâm thương mại và khu làm việc ngày càng tăng, trong khi việc quản lý bãi xe vẫn chủ yếu dựa vào con người và thao tác thủ công. Điều này dẫn đến nhiều vấn đề như:

- Mất thời gian trong việc đăng ký và kiểm soát xe ra/vào.
- Thiếu minh bạch trong việc ghi nhận thời gian đỗ xe.
- Không có hệ thống theo dõi trạng thái chỗ đỗ theo thời gian thực.

Trong bối cảnh phát triển của các mô hình “Smart City” và nhu cầu số hóa hạ tầng quản lý, việc xây dựng một hệ thống đỗ xe thông minh, tự động là hoàn toàn phù hợp và cần thiết. Dự án Auto-parking System (APS) được triển khai nhằm giải quyết bài toán trên bằng cách kết hợp giữa phần mềm (web + API) và phần cứng (ESP32 + cảm biến) để tự động hóa quy trình đăng ký, check-in và check-out chỗ đỗ xe, giảm thiểu can thiệp của con người, tăng tính chính xác và tiện lợi cho người sử dụng. Hơn thế nữa, độ phức tạp để triển khai dự án không quá là khó khăn, dễ bảo trì và nâng cấp phần mềm, sửa phần cứng

1.2. Khái quát chung về project.



Sơ đồ khối mô tả hệ thống hệ thống

Hệ thống bao gồm ba phần chính: người dùng, server và phần cứng điều khiển (ESP32). Người dùng có thể đăng ký chỗ đỗ xe từ xa qua website (kết nối Internet)


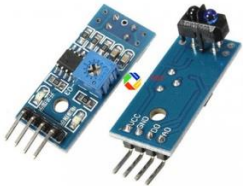

hoặc tại chỗ thông qua captive portal của ESP32 (kết nối nội bộ). Sau khi đăng ký, thông tin được gửi về server Node.js và lưu vào cơ sở dữ liệu MySQL.


Khi người dùng đến bãi xe, họ thực hiện check-in bằng biển số xe và mã OTP. Server gửi yêu cầu đến ESP32 để xử lý. ESP32 sẽ kiểm tra cảm biến tại vị trí xe, nếu xe đang ở đúng vị trí, nó sẽ điều khiển servo mở barrier trong thời gian xác định, đồng thời điều khiển LED hiển thị trạng thái slot. Tương tự, khi người dùng muốn check-out, hệ thống sẽ kiểm tra xem xe đã rời khỏi vị trí chưa rồi mới đóng barrier và cập nhật database.

Sự kết hợp giữa phần mềm và phần cứng giúp hệ thống hoạt động tự động, chính xác và giảm thiểu can thiệp thủ công.

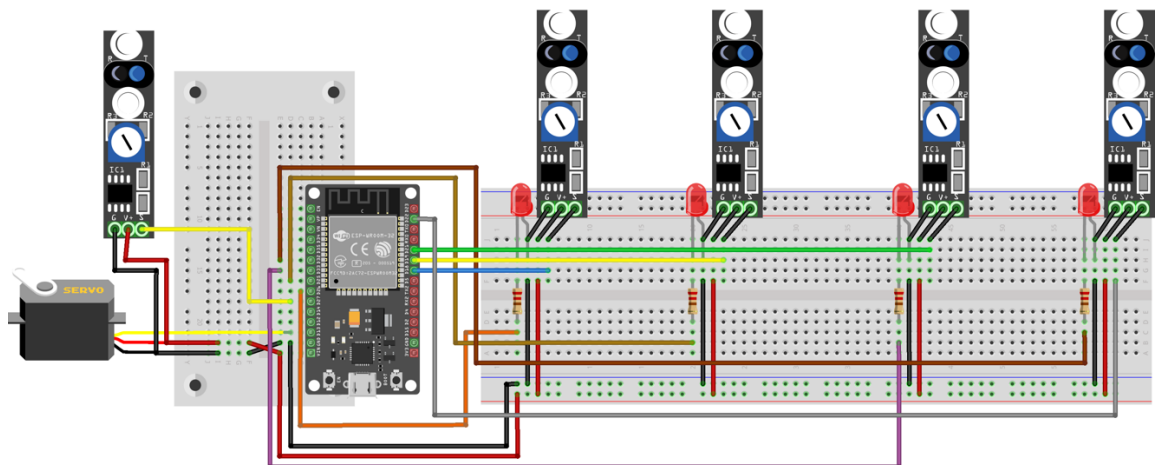
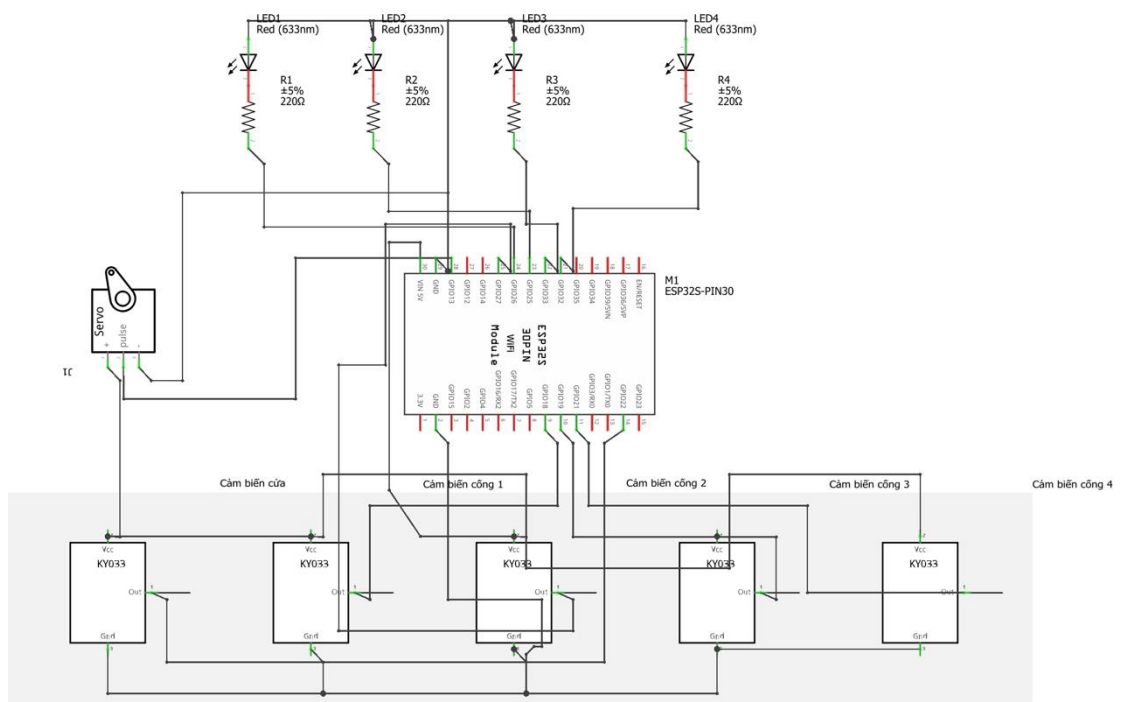
II. Phân tích, thiết kế hệ thống

II.1. Thiết kế phần cứng

STT	Tên thiết bị	Mã	Số lượng	Hình ảnh minh họa
1	Esp32	ESP32 DevKit v1	1	
2	Module cảm biến dò đường	TCRT500	5	
3	Động cơ Servo	SG90	1	

4	Module đèn led	LED	4	
---	----------------	-----	---	---

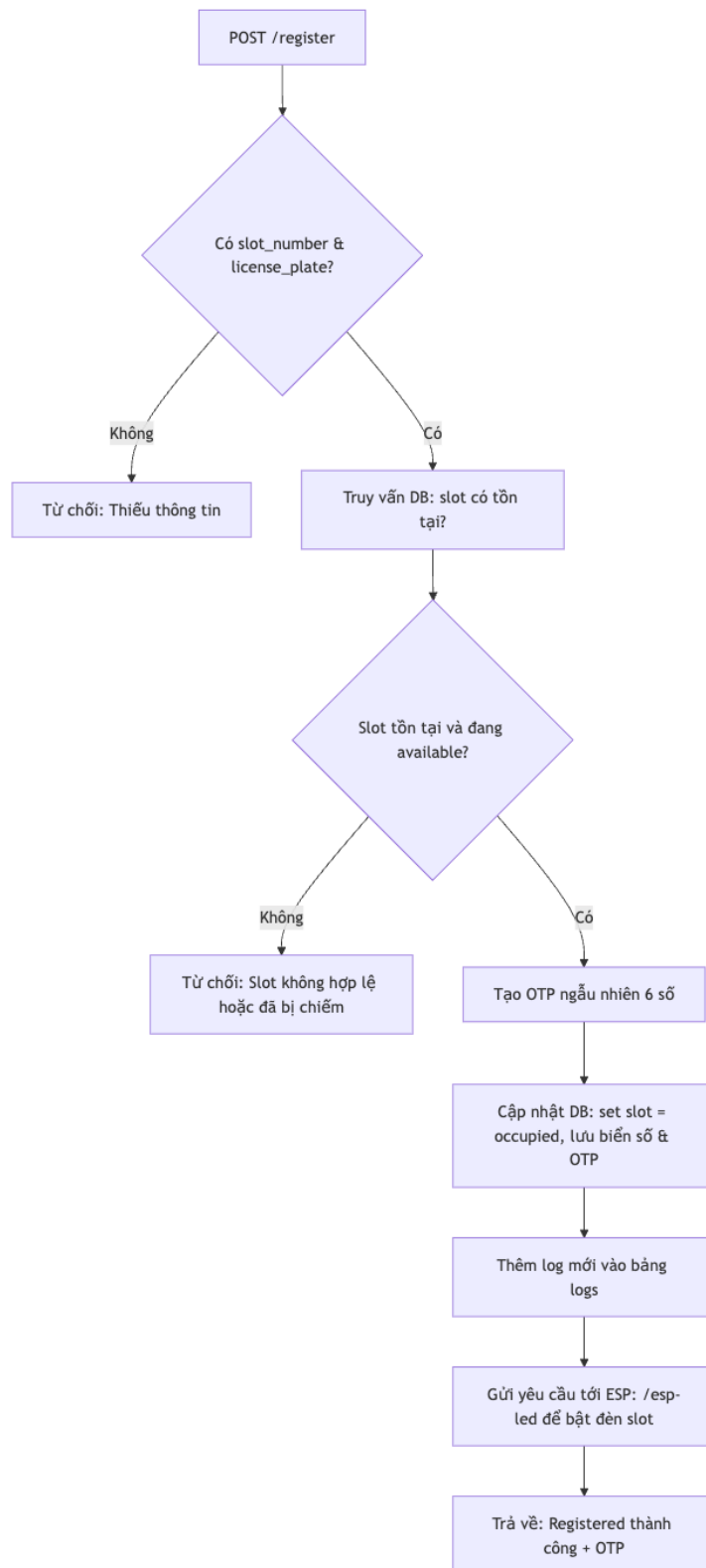
Bảng danh mục linh kiện, thiết bị của hệ thống



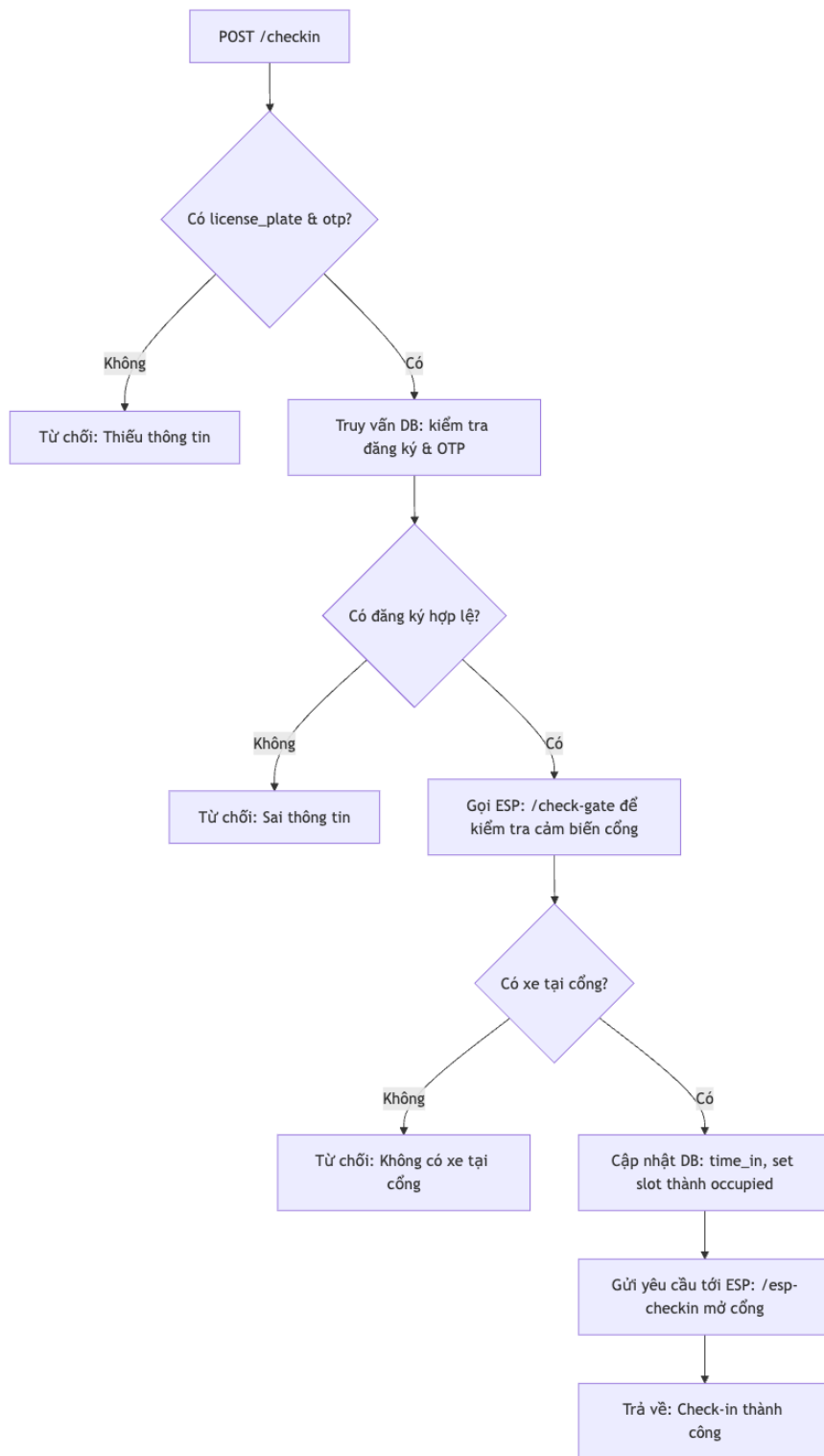
Sơ đồ nguyên lý và lắp đặt mạch điện hệ thống

II.2. Thiết kế phần mềm

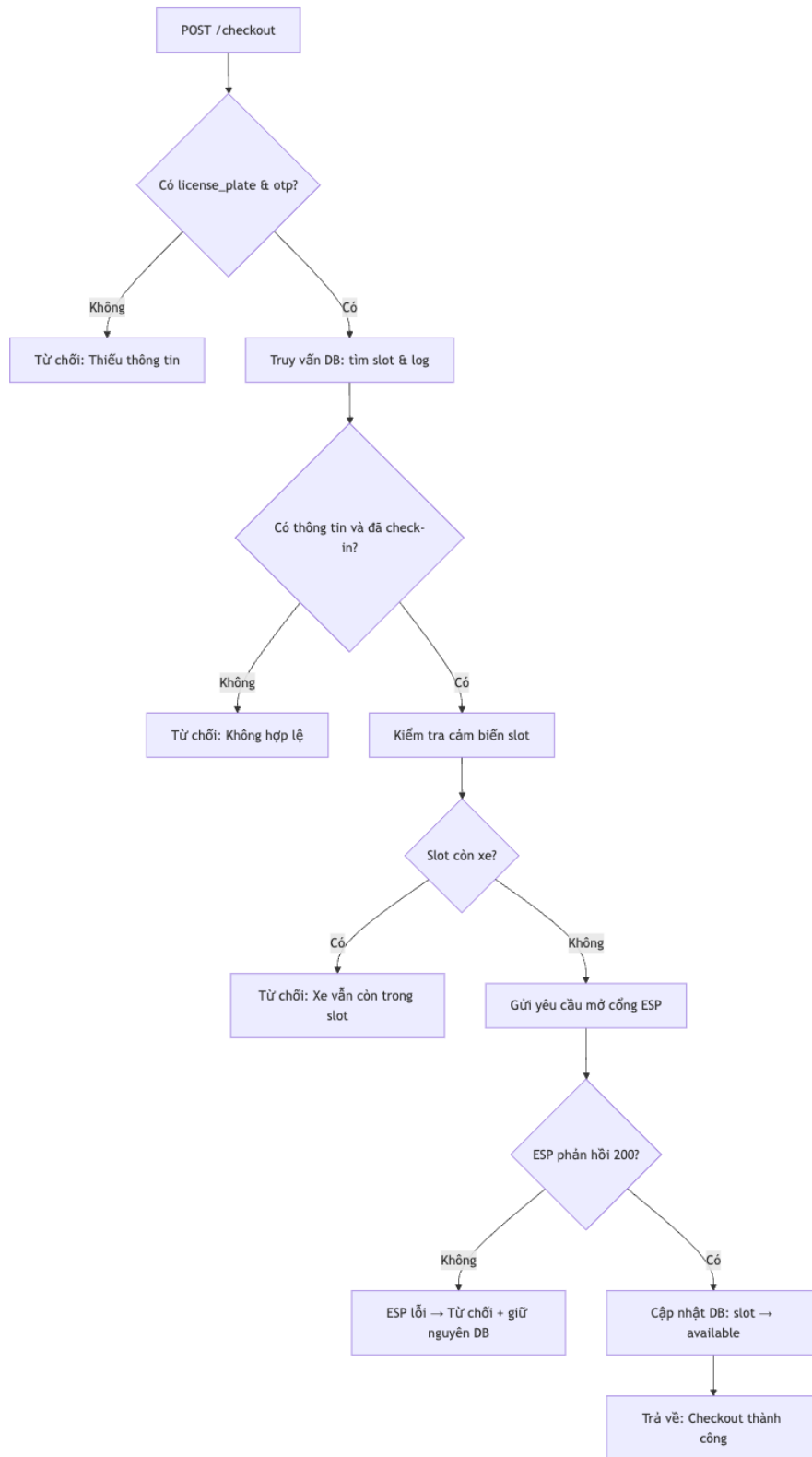
2.1. Lưu đồ thuật toán



Lưu đồ thuật toán chức năng đăng kí slot xe



Lưu đồ thuật toán chức năng check-in



Lưu đồ thuật toán chức năng check-out

2.2. Một số hàm quan trọng trong chương trình hệ thống


```

50 }
51   ^192KDeFeFe(INNG):
52   2fofoccnbjeq[i] = false;
53   89fe26e1a0*mlife(CFO2E^VMEFE):
54   qef9λ(200):
55   qif8if9fmlife(2fo1^fed[i], gom):
56
57   }
58   qef9λ(10):
59   }
60   f92fbjtuK[i] = uom:
61   qif8if9fmlife(2fo1^fed[i], f9q2f9fe[i] 3 HIGH : gom):
62   f9q2f9fe[i] = if9q2f9fe[i]:
63   if (uom - f92fbjtuK[i] > 100) {
64     nu2if9ueq foug uom = wifjiz():
65     mlife (qif8if9f9eq(2fo1^2E12OK[i]) ≠ gom) {
66
67     89fe26e1a0*mlife(OBE1^VMEFE):
68     2f91fbjtuKtu8(i):
69
70     if i = 2fof - 1:
71     qef9fe (if*)b919wefei:
72     if 2fof = *((if*)b919wefei):
73     101q f9uqf9eCpeckif192K(101q* b919wefei) {
74
75
76   // ----- CHECK-IN -----

```

```

1 server.on("/esp-checkin", HTTP_POST, [(AsyncWebServerRequest* req){}, NULL,
2   [(AsyncWebServerRequest* req, uint8_t* data, size_t len, size_t index, size_t total) {
3     StaticJsonDocument<200> doc;
4     if (deserializeJson(doc, data)) {
5       req->send(400, "application/json", "{\"error\":\"Invalid JSON\"}");
6       return;
7     }
8
9     int slot = doc["slot"];
10    if (!isVehicleAtGate()) {
11      req->send(403, "application/json", "{\"error\":\"No vehicle at gate\"}");
12      return;
13    }
14
15    slotOccupied[slot - 1] = true;
16    updateLEDs();
17    req->send(200, "application/json", "{\"status\":\"Check-in ok\"}");
18    int* arg = new int(slot);
19    xTaskCreate(handleCheckInTask, "CheckInTask", 2048, arg, 1, NULL);
20  });

```

Hàm thực hiện chức năng check-in trên ESP32

```

1 // ----- CHECK-OUT -----
2
3 void handleCheckOut(int slot, AsyncWebServerRequest* req) {
4     int i = slot - 1;
5
6     if (digitalRead(SLOT_SENSOR[i]) == LOW) {
7         req->send(403, "application/json",
8             "{\"error\":\"Xe vẫn ở trong slot\"}");
9         return;
10    }
11
12    gateServo.write(OPEN_ANGLE);
13
14    unsigned long timeout = millis() + 10000;
15    while (digitalRead(GATE_SENSOR_PIN) == HIGH && millis() < timeout) {
16        delay(50);
17    }
18
19    timeout = millis() + 10000;
20    while (digitalRead(GATE_SENSOR_PIN) == LOW && millis() < timeout) {
21        delay(50);
22    }
23
24    gateServo.write(CLOSE_ANGLE);
25    slotOccupied[i] = false;
26    updateLEDs();
27    req->send(200, "application/json", "{\"status\":\"Check-out ok\"}");
28 }

```

```

1 server.on("/esp-checkout", HTTP_POST, [](AsyncWebServerRequest* req){}, NULL,
2     [](AsyncWebServerRequest* req, uint8_t* data, size_t len, size_t index, size_t total) {
3         StaticJsonDocument<200> doc;
4         if (deserializeJson(doc, data)) {
5             req->send(400, "application/json", "{\"error\":\"Invalid JSON\"}");
6             return;
7         }
8         int slot = doc["slot"];
9         handleCheckOut(slot, req);
10    });

```

Hàm thực hiện chức năng check-out trên ESP32

III. Thực nghiệm, đánh giá

Hệ thống được lắp đặt và kiểm thử thực tế trong môi trường mạng nội bộ do ESP32 phát và dùng Captive Portal để hiện giao diện web tự động.

1. Quy trình lắp đặt phần cứng:

- Kết nối cảm biến hồng ngoại TCRT5000 tại mỗi slot để phát hiện xe.
- Kết nối mô-đun LED để hiển thị trạng thái chỗ đỗ.
- Kết nối servo motor để điều khiển đóng/mở cổng.
- ESP32 được cấp nguồn và lập trình để kết nối với các thiết bị trên.

2. Cài đặt và triển khai phần mềm:

- Phần mềm backend được triển khai trên máy chủ nội bộ.
- Giao diện client đơn giản được chạy trên trình duyệt web máy tính hoặc điện thoại.
- ESP32 chạy web server riêng, cung cấp API xử lý đóng/mở cổng, kiểm tra cảm biến, đồng bộ trạng thái slot.

3. Tiến trình kiểm thử:

Bước 1: Đăng ký slot qua giao diện web → kiểm tra đèn LED hiển thị đúng.

Bước 2: Đưa xe tới cổng → cảm biến phát hiện → gửi lệnh *check-in*.

Bước 3: Kiểm tra phản hồi ESP32, trạng thái LED, đóng/mở cổng.

Bước 4: Đưa xe ra khỏi slot → thực hiện *check-out* và kiểm tra quá trình.

Các thao tác được thử nghiệm trên cả máy tính, điện thoại, nhiều lượt liên tục.

4. Đánh giá hiệu năng và ổn định:

Trên điện thoại: trung bình mất 7 giây để mở cổng khi check-in, và 14 giây khi check-out (bao gồm xác thực và chờ phản hồi).

Trên máy chủ nội bộ: phản hồi gần như tức thì (0 giây).

Cảm biến hồng ngoại: hoạt động ổn định nếu đã căn chỉnh hợp lí, xác định đúng trạng thái slot và xe tại cổng, ghi nhận một số sai lệch không đáng kể do môi trường hoặc yếu tố con người trong thử nghiệm.

Server: Việc xử lý OTP, kiểm tra biển số và slot đều thực hiện bằng truy vấn SQL trực tiếp, đảm bảo độ chính xác cao.

5. Một số hạn chế và đề xuất:

- Có một số tình huống ESP32 mất kết nối Wi-Fi khi mạng nội bộ không ổn định (nhất là khi không có internet).
- Nếu triển khai thực tế, nên sử dụng Wi-Fi ổn định hoặc server public, tránh mạng tách biệt nội bộ để đảm bảo khả năng kết nối và phản hồi nhanh.

IV. Hướng phát triển của dự án

Để nâng cao hiệu quả và tính thực tiễn khi triển khai, hệ thống có thể được cải tiến theo các hướng sau:

1. Tự động hóa quá trình check-in/check-out bằng công nghệ nhận diện biển số (ALPR) kết hợp camera AI, giảm phụ thuộc vào thao tác thủ công và OTP.
2. Tích hợp quét mã vạch hoặc QR code để xác thực phương tiện nhanh và chính xác hơn.
3. Cải thiện kết nối mạng, sử dụng ESP32 trong hệ thống Wi-Fi có internet để đảm bảo không bị gián đoạn khi truyền dữ liệu.
4. Tối ưu tốc độ phản hồi từ server đến ESP32, đặc biệt trên các thiết bị di động.
5. Phát triển ứng dụng di động cho người dùng cuối để quản lý chỗ đỗ, nhận thông báo, theo dõi lịch sử vào/ra.
6. Nâng cấp cơ chế bảo mật, chống trùng biển số và lạm dụng OTP bằng cách áp dụng xác thực đa lớp.
7. Mở rộng số lượng slot và hỗ trợ nhiều ESP32, cho phép hệ thống quản lý nhiều khu vực hoặc tầng trong bãi đỗ xe, xây dựng các module riêng lắp đặt dễ dàng để có thể mở rộng một cách dễ dàng.