

HỆ THỐNG BÃI ĐỖ XE THÔNG MINH

Đề xuất dự án ứng dụng công nghệ AIoT

Ngày 22 tháng 5 năm 2025

Tóm tắt nội dung

Báo cáo này trình bày đề xuất phát triển hệ thống bãi đỗ xe thông minh ứng dụng công nghệ AIoT, nhằm tự động hóa và tối ưu hóa quy trình quản lý bãi đỗ xe. Giải pháp đề xuất kết hợp các công nghệ nhận diện hình ảnh (biển số xe, khuôn mặt người lái) và hệ thống cảm biến IoT để theo dõi tình trạng chỗ đỗ trong thời gian thực, loại bỏ thao tác thủ công và giảm thiểu sai sót trong quá trình vận hành. Các tính năng chính bao gồm: tự động nhận diện phương tiện ra vào, giám sát trạng thái chỗ đỗ, lưu trữ dữ liệu trên nền tảng cloud và cung cấp giao diện dashboard trực quan cho người quản lý. Hệ thống có thể triển khai linh hoạt tại nhiều môi trường khác nhau như trường học, khu dân cư, bãi xe công cộng, góp phần xây dựng hạ tầng đô thị thông minh và hiện đại.

Mục lục

1	Giới thiệu	3
1.1	Bối cảnh	3
1.2	Giải pháp	3
1.3	Phát Biểu Bài toán	4
1.3.1	Dữ liệu đầu vào (input)	4
1.3.2	Kết quả đầu ra (ouput)	4
1.3.3	Giới hạn hệ thống	4
1.4	Đóng góp chính của đề tài	5
2	Mô hình mô phỏng và luồng hoạt động	5
2.1	Mô hình hệ thống	5
2.2	Luồng hoạt động chính của hệ thống	6
2.2.1	Entry flow (luồng hoạt động ở cổng vào)	6
2.2.2	Exit flow (luồng hoạt động ở cổng ra):	7
3	Công nghệ	8
3.1	Công nghệ sử dụng	8
3.1.1	Hệ thống phần cứng	8
3.1.2	Ứng dụng AI phân tích hình ảnh	9
3.1.3	Giao diện người dùng và quản trị	10
3.1.4	Kết nối và nền tảng lưu trữ	10
3.2	Thiết bị	10
4	Kế hoạch thực hiện	12
4.1	Phân công nhân lực	12
4.2	Timeline	12
4.3	Chi phí thực hiện (chi phí thiết bị, chi phí làm)	13

1 Giới thiệu

1.1 Bối cảnh

Hiện nay, nhiều bãi đỗ xe tại các khu vực như chung cư, trường học, tòa nhà văn phòng đã bắt đầu áp dụng phần mềm và thẻ từ (hoặc thẻ RFID) để quản lý việc ra vào. Tuy nhiên, các hệ thống này chủ yếu chỉ dừng lại ở mức cơ bản: người dùng quét thẻ khi vào – ra, hệ thống ghi lại thời gian và biển số xe (nếu có camera phụ trợ), sau đó lưu dữ liệu cục bộ. Chúng không kết nối với các cảm biến để giám sát tình trạng từng vị trí đỗ, không cung cấp thông tin theo thời gian thực cho người dùng hoặc người quản lý, và thường thiếu khả năng đồng bộ dữ liệu lên nền tảng trực tuyến để truy xuất khi có sự cố.

Hệ quả là việc vận hành vẫn phụ thuộc nhiều vào thao tác thủ công, dễ xảy ra sai sót như mất thẻ, quên quét thẻ, hoặc ghi nhận sai thời gian. Đồng thời, người dùng cũng gặp bất tiện khi không thể biết trước bãi còn chỗ hay không, dẫn đến mất thời gian di chuyển và tìm kiếm chỗ đỗ. Trong bối cảnh đô thị hóa nhanh chóng, nhu cầu tự động hóa và tối ưu hóa quản lý bãi đỗ xe trở nên cấp thiết hơn bao giờ hết.

Công nghệ AIoT – kết hợp giữa trí tuệ nhân tạo (AI) và Internet vạn vật (IoT) – mang đến giải pháp khả thi để xây dựng các bãi đỗ xe thông minh: tự động nhận diện xe qua camera, cập nhật tình trạng chỗ đỗ theo thời gian thực, hỗ trợ giám sát tập trung và nâng cao trải nghiệm người dùng, hướng tới một hệ thống hạ tầng đô thị hiện đại và hiệu quả hơn.

1.2 Giải pháp

Nhóm đề xuất phát triển một hệ thống **Bãi đỗ xe thông minh**, nhằm tự động hóa và tối ưu hóa quy trình quản lý bãi xe. Hệ thống bao gồm các chức năng chính sau:

- **Tự động nhận diện biển số xe** hoặc **khuôn mặt tài xế** khi phương tiện ra vào, giúp loại bỏ thao tác thủ công như quét thẻ hay ghi tay.
- **Ghi nhận thời điểm vào – ra một cách chính xác**, đồng bộ dữ liệu theo thời gian thực.
- **Giám sát tình trạng chỗ đỗ (còn/trống)** thông qua cảm biến đặt tại từng vị trí và hiển thị bằng hệ thống đèn báo trực quan.

- **Lưu trữ toàn bộ dữ liệu** về phương tiện, người dùng và lịch sử ra vào trên nền tảng cloud (ví dụ như Firebase hoặc ThingsBoard), hỗ trợ truy xuất và phân tích khi cần.
- **Cung cấp giao diện dashboard trực quan**, giúp ban quản lý dễ dàng theo dõi tình trạng bãi xe, thống kê lưu lượng phương tiện và kiểm tra lịch sử hoạt động.

Hệ thống này không chỉ giúp **rút ngắn thời gian xử lý, giảm sai sót do con người**, mà còn có thể **triển khai linh hoạt** tại các khu vực như **trường học, khu dân cư, bãi xe công cộng hoặc cơ quan hành chính**, góp phần xây dựng hạ tầng đô thị thông minh và hiện đại.

1.3 Phát Biểu Bài toán

1.3.1 Dữ liệu đầu vào (input)

- Hình ảnh/video từ camera tại lối vào và lối ra bãi xe.

1.3.2 Kết quả đầu ra (output)

- Kết quả nhận diện biển số hoặc khuôn mặt tại thời điểm vào – ra.
- Ghi nhận và lưu trữ thời điểm vào – ra xe.
- Hiển thị trạng thái chỗ đỗ (còn trống hoặc đã chiếm chỗ) theo thời gian thực.
- Giao diện trực quan cho người quản lý và người dùng.
- Dữ liệu được lưu trữ và truy xuất thông qua nền tảng cloud.

1.3.3 Giới hạn hệ thống

- Môi trường lắp đặt (ánh sáng yếu, thời tiết xấu) ảnh hưởng đến độ chính xác nhận diện.
- Hệ thống cần camera có độ phân giải tối thiểu và lắp đặt đúng góc.
- Tài nguyên xử lý (RAM, bộ nhớ, mạng) cần đảm bảo để hệ thống hoạt động thời gian thực.
- Độ trễ thấp là yêu cầu bắt buộc để đảm bảo trải nghiệm người dùng.

1.4 Đóng góp chính của đề tài

- Đề xuất mô hình hệ thống bãi đỗ xe thông minh tích hợp công nghệ **AIoT**, ứng dụng được vào thực tế.
- Tích hợp **nhận diện hình ảnh (biển số/khuôn mặt)** và **cảm biến IoT** để tự động hóa hoàn toàn quy trình vận hành.
- Thiết kế giao diện dashboard giúp quản lý dễ dàng theo dõi, giám sát và truy xuất lịch sử.
- Định hướng mở rộng và triển khai tại các bãi giữ xe trường học, tòa nhà, trung tâm thương mại...
- Hướng đến việc **giảm sai sót con người, nâng cao hiệu quả quản lý và tối ưu trải nghiệm người dùng**.

2 Mô hình mô phỏng và luồng hoạt động

2.1 Mô hình hệ thống

Hệ thống bãi đỗ xe thông minh sử dụng nhận diện **khuôn mặt kết hợp biển số xe**, với các thiết bị phần cứng như sau:

1. Cổng vào:

- **ESP32-CAM_1**: quét và nhận diện **biển số xe**.
- **ESP32-CAM_2**: quét và nhận diện **khuôn mặt tài xế**.
- **Màn hình OLED**: hiển thị số lượng ô đỗ xe còn trống hoặc báo lỗi.
- **Servo motor**: điều khiển barrier tự động mở nếu xác thực thành công.
- **Buzzer**: phát âm báo xác nhận **thành công hoặc lỗi**.

2. Khu vực đỗ xe:

- Các ô giữ xe được xếp thành các hàng.

3. Cổng ra:

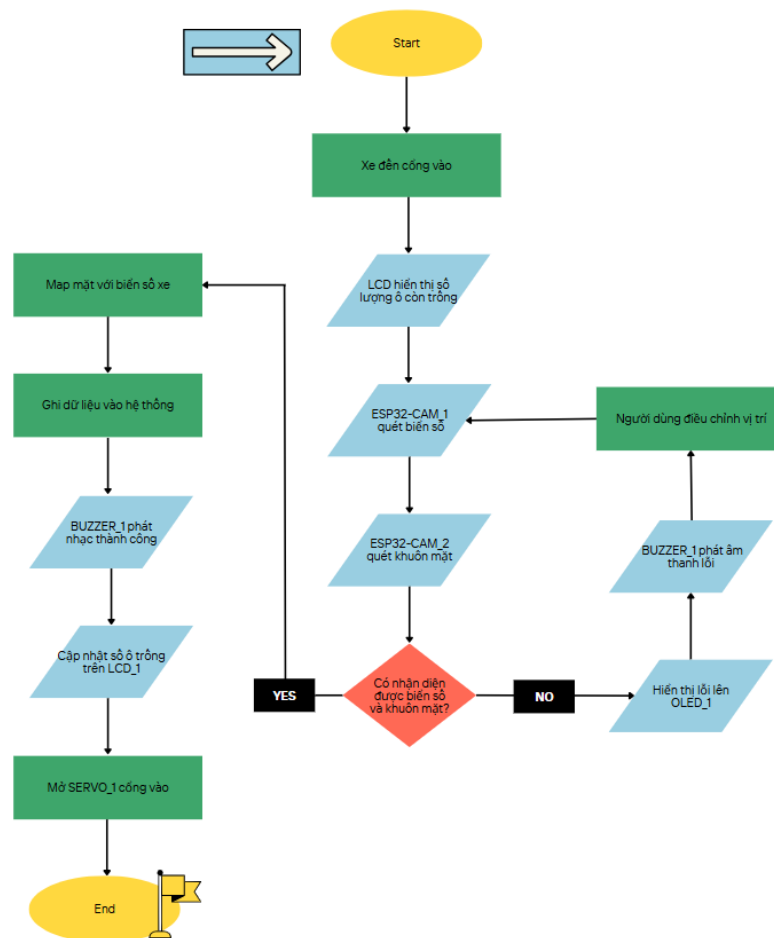
- **ESP32-CAM_3**: quét lại **biển số xe**.
- **ESP32-CAM_4**: quét lại **khuôn mặt tài xế**.
- **Màn hình OLED**: hiển thị số lượng ô đỗ xe còn trống hoặc báo lỗi.

- **Servo motor**: điều khiển barrier tự động mở nếu xác thực thành công.
- **Buzzer**: phát âm báo xác nhận **thành công hoặc lỗi**.

2.2 Luồng hoạt động chính của hệ thống

2.2.1 Entry flow (luồng hoạt động ở cổng vào)

- ESP32-CAM_1 quét **biển số xe**.
- ESP32-CAM_2 quét **khuôn mặt tài xế**.
- Hệ thống xử lý và kiểm tra:
 - Nếu **cả hai đều nhận diện thành công**:
 - * Map dữ liệu: `face_id` `license_plate`.
 - * Lưu vào cơ sở dữ liệu tạm thời.
 - * Servo mở barrier cho xe vào.
 - * Buzzer phát âm báo thành công.
 - * Màn hình OLED hiển thị số ô trống hiện tại.
 - Nếu **không nhận diện được**:
 - * Hiển thị thông báo lỗi trên màn hình OLED.
 - * Buzzer phát cảnh báo nhẹ.
 - * Chờ người dùng điều chỉnh vị trí xe để quét lại.

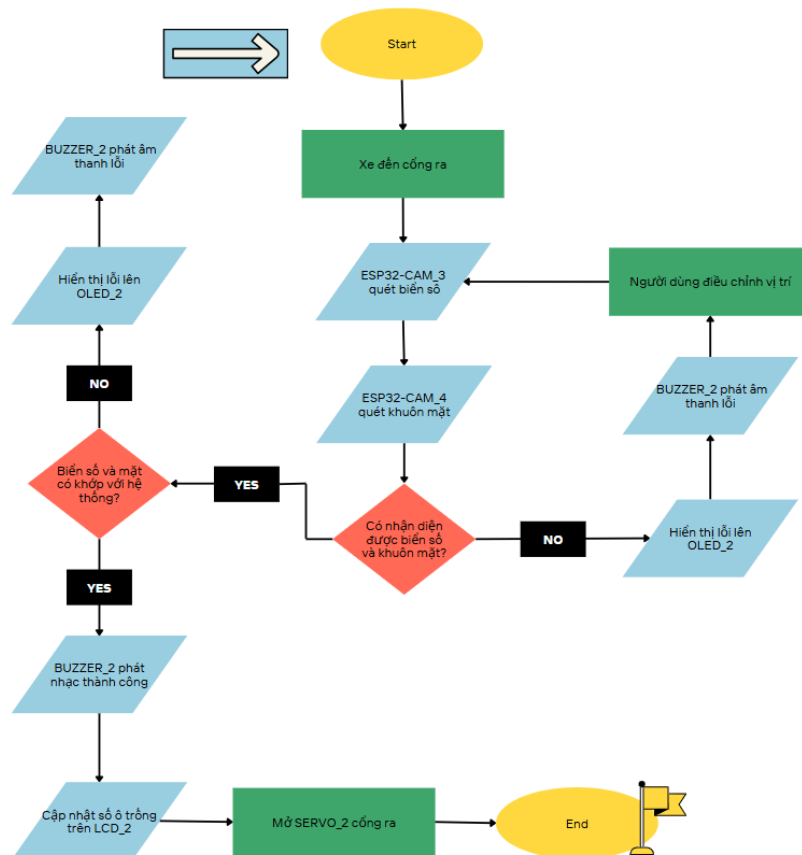


Hình 1: Flowchart cho entry flow

2.2.2 Exit flow (luồng hoạt động ở cổng ra):

- ESP32-CAM_3 quét **biển số xe**.
- ESP32-CAM_4 quét **khuôn mặt tài xế**.
- Hệ thống kiểm tra:
 - Biển số và khuôn mặt có được nhận diện không?
 - Có khớp với dữ liệu đã lưu khi vào không?
 - Nếu **có khớp**:
 - * Mở barrier cho xe ra.
 - * Buzzer phát âm báo thành công.
 - * Xóa dữ liệu mapping khỏi hệ thống.
 - * Cập nhật số lượng ô trống trên màn hình OLED ở cổng vào.
 - Nếu **không khớp**:
 - * Barrier không mở.

- * Buzzer phát âm báo nguy hiểm để cảnh báo vi phạm.
- * Màn hình OLED hiển thị thông tin lỗi.



Hình 2: Flowchart cho exit flow

3 Công nghệ

3.1 Công nghệ sử dụng

3.1.1 Hệ thống phần cứng

- Camera ESP32-CAM

- Sử dụng 2 module ESP32-CAM tại mỗi cổng (vào và ra), một camera chụp khuôn mặt tài xế, một camera chụp biển số xe.
- ESP32-CAM sẽ **đợi tín hiệu từ nút bấm** để chụp, sau đó gửi ảnh lên server qua kết nối Wi-Fi.

- Board điều khiển trung tâm ESP32

- Một board ESP32 dùng để điều khiển các thiết bị ngoại vi như màn hình OLED hiển thị thông tin hướng dẫn tài xế và nhận tín hiệu nút bấm từ người dùng.
- Board giao tiếp không dây qua Wi-Fi với các ESP32-CAM để điều phối hoạt động chung như ra lệnh chụp ảnh, nhận dữ liệu, điều khiển barrier.
- Ngoài ra, board điều khiển relay để đóng/mở thanh chắn barrier tự động khi nhận diện thành công.

- **Màn hình OLED**

- Màn hình OLED nhỏ gọn được kết nối với board ESP32 trung tâm để hiển thị thông tin trực quan như số ô đỗ được phân, trạng thái chỗ đỗ, hoặc các thông báo hướng dẫn cho tài xế.
- OLED phù hợp để hiển thị đồ họa đơn giản, tiết kiệm năng lượng, dễ dàng kết nối qua giao tiếp I2C hoặc SPI.

- **Nút bấm**

- Nút bấm được gắn với board ESP32 điều khiển trung tâm, cho phép người dùng nhấn khi đã đứng đúng vị trí chờ quét để kích hoạt camera.

- **Servo motor**

- Servo motor được điều khiển bởi board ESP32 trung tâm.
- Servo sẽ quay mở hoặc đóng khi hệ thống xác nhận thành công khuôn mặt hoặc biển số xe hợp lệ, đảm bảo an ninh và quản lý xe ra/vào chính xác.

- **Cảm biến siêu âm HC-SR04**

- Cảm biến siêu âm HC-SR04 được lắp tại các vị trí đỗ xe để xác định xem ô đó đang có xe đậu hay trống.
- Cảm biến hoạt động bằng cách phát sóng siêu âm từ đó tính khoảng cách đến vật thể phía trước. Nếu khoảng cách nhỏ hơn ngưỡng định sẵn, hệ thống sẽ xác định ô đã có vật thể chiếm.

3.1.2 Ứng dụng AI phân tích hình ảnh

- **Nhận diện khuôn mặt**

- **Phát hiện khuôn mặt:** Sử dụng mô hình **YOLOv6** để phát hiện vùng chứa khuôn mặt (bounding box) trong ảnh. Mô hình có tốc độ nhanh, độ chính xác cao, phù hợp xử lý ảnh thời gian thực.
- **Nhận diện khuôn mặt:** Dựa trên các vector đặc trưng được tạo bởi mô hình **FaceNet**, so sánh với database để xác định danh tính.
- **Nhận diện biển số xe**
 - **Phát hiện biển số xe:** Sử dụng mô hình **YOLOv8** để phát hiện vùng chứa biển số xe trong ảnh.
 - **Nhận dạng ký tự biển số:** Cắt vùng biển số từ ảnh dựa trên kết quả phát hiện, áp dụng thuật toán nhận dạng ký tự **Tesseract OCR** để trích xuất chuỗi ký tự.

3.1.3 Giao diện người dùng và quản trị


- **Frontend:** Sử dụng các công nghệ phổ biến HTML, CSS, JavaScript để xây dựng giao diện thân thiện, dễ sử dụng cho tài xế và quản trị viên.
- **Backend:** Sử dụng Django làm framework backend, xử lý các logic nghiệp vụ như xác thực người dùng, quản lý dữ liệu xe và vị trí bãi đỗ, cung cấp API cho frontend và thiết bị IoT kết nối.

3.1.4 Kết nối và nền tảng lưu trữ

- **MQTT (Message Queuing Telemetry Transport):** Giao thức truyền thông nhẹ, tối ưu cho các thiết bị IoT với băng thông hạn chế. MQTT hỗ trợ kết nối ổn định, truyền dữ liệu theo dạng publish-subscribe, rất phù hợp để truyền trạng thái cảm biến, điều khiển thiết bị trong hệ thống bãi đỗ xe thông minh.
- **MongoDB:** Cơ sở dữ liệu NoSQL dạng tài liệu, linh hoạt trong việc lưu trữ dữ liệu phi cấu trúc như log hệ thống, dữ liệu người dùng, thông tin xe, hình ảnh. Được sử dụng phổ biến trong các hệ thống backend nhờ khả năng mở rộng và thao tác linh hoạt với dữ liệu JSON.

3.2 Thiết bị

STT	Tên thiết bị	Số lượng	Giá tiền	Hình ảnh	Nguồn
1	Camera ESP32-CAM	4	269,000		Xem tại đây
2	Board điều khiển ESP32	2	170,000		Xem tại đây
3	Màn hình OLED 0.96 inch	2	105,000		Xem tại đây
4	Nút bấm	2	4,000		Xem tại đây
5	Servo motor MG90S	2	39,000		Xem tại đây
6	Cảm biến siêu âm HC-SR04	Theo số ô đồ	20,000		Xem tại đây

STT	Tên thiết bị	Số lượng	Giá tiền	Hình ảnh	Nguồn
7	Dây dẫn	Theo nhu cầu	8,000		Xem tại đây

4 Kế hoạch thực hiện

4.1 Phân công nhân lực

Thành viên	Vai trò chính	Nhiệm vụ cụ thể
Cao Uyên Nhi	Trưởng nhóm, quản lý tiến độ	
Phụ trách AI – Nhận diện biển số/khuôn mặt	Tổng hợp báo cáo, điều phối nhóm, hỗ trợ kỹ thuật	
Thu thập dữ liệu, huấn luyện mô hình, triển khai xử lý ảnh		
Trần Thị Cát Tường	Phụ trách IoT – Cảm biến và cloud	Kết nối cảm biến, lập trình ESP32, gửi dữ liệu lên Firebase/ThingsBoard
Võ Lê Việt Tú	Thiết kế phần cứng – Mạch, cảm biến	Lắp ráp hệ thống, bố trí camera và sensor thực tế
Lưu Thanh Thuý	Giao diện & báo cáo	Thiết kế dashboard, lập trình hiển thị và làm tài liệu

4.2 Timeline

Tuần	Công việc chính
1–2	Nghiên cứu tài liệu, xác định yêu cầu, phân công nhiệm vụ
3–4	Thu thập dữ liệu khuôn mặt/biển số, khảo sát thiết bị cần thiết
5–6	Triển khai mô hình AI nhận diện, kết nối cảm biến với ESP32
7–8	Thiết kế giao diện dashboard, xây dựng cơ sở dữ liệu trên cloud
9–10	Tích hợp toàn hệ thống: AI + IoT + giao diện
11	Chạy thử, hiệu chỉnh, test toàn bộ quy trình thực tế
12	Viết báo cáo, chuẩn bị slide thuyết trình, hoàn thiện demo

4.3 Chi phí thực hiện (chi phí thiết bị, chi phí làm)

Hạng mục	Số lượng	Đơn giá (VNĐ)	Thành tiền (VNĐ)
ESP32	2	120.000	240.000
Camera ESP32-CAM	2	180.000	360.000
Cảm biến siêu âm (HC-SR04)	4	30.000	120.000
Breadboard, dây nối, điện trở,...	-	100.000	100.000
Nguồn, adapter	2	50.000	100.000
Vật tư lắp đặt, hộp, khung mô hình	-	200.000	200.000
Tổng chi phí dự kiến			1.120.000

Chi phí có thể thay đổi tùy vào lựa chọn thiết bị và nguồn cung ứng.