

ĐẠI HỌC QUỐC GIA TP.HCM  
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN  
KHOA CÔNG NGHỆ THÔNG TIN



TÀI LIỆU ĐỒ ÁN

---

## Chỉ Dẫn Kỹ Thuật (Specification)

---

Thành viên

| Họ Tên             | ID       | Email                       |
|--------------------|----------|-----------------------------|
| Cao Uyển Nhi       | 22127310 | cunhi22@clc.fitus.edu.vn    |
| Lưu Thanh Thuý     | 22127410 | ltthuy22@clc.fitus.edu.vn   |
| Võ Lê Việt Tú      | 22127435 | vlvtu22@clc.fitus.edu.vn    |
| Trần Thị Cát Tường | 22127444 | ttctuong22@clc.fitus.edu.vn |

Giáo viên hướng dẫn

---

Họ Tên

---

Giáo viên: Võ Hoài Việt

Giáo viên: Đỗ Thị Thanh Hà

---

25 tháng 7, 2025

# Mục lục

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Giới thiệu</b>   | <b>4</b>  |
| 1.1      | Bối cảnh . . . . .  | 4         |
| 1.2      | Mục đích . . . . .  | 4         |
| 1.3      | Mục tiêu . . . . .  | 4         |
| 1.4      | Phát biểu bài toán . . . . .                                | 5         |
| 1.4.1    | Dữ liệu đầu vào . . . . .                                   | 5         |
| 1.4.2    | Kết quả đầu ra . . . . .                                    | 5         |
| 1.5      | Phạm vi và Giới hạn của Hệ thống . . . . .                  | 5         |
| 1.5.1    | Phạm vi triển khai . . . . .                                | 5         |
| 1.5.2    | Giới hạn của Hệ thống . . . . .                             | 5         |
| 1.6      | Đóng góp chính của Đề tài . . . . .                         | 6         |
| <b>2</b> | <b>Thiết kế hệ thống</b>                                    | <b>7</b>  |
| 2.1      | Kiến trúc phần mềm . . . . .                                | 7         |
| 2.1.1    | Thiết bị đầu vào (Input Layer) . . . . .                    | 7         |
| 2.1.2    | Tầng xử lý trung tâm (Processing Layer) . . . . .           | 8         |
| 2.1.3    | Thiết bị điều khiển - phản hồi (Output Layer) . . . . .     | 8         |
| 2.2      | Hệ thống phần cứng . . . . .                                | 8         |
| 2.2.1    | Chi tiết thiết bị . . . . .                                 | 8         |
| 2.2.2    | Tổng hợp . . . . .  | 12        |
| 2.3      | Luồng thực thi hệ thống . . . . .                           | 12        |
| <b>3</b> | <b>Triển khai và Công Nghệ Sử dụng</b>                      | <b>14</b> |
| 3.1      | Quy trình triển khai . . . . .                              | 14        |
| 3.2      | Công nghệ sử dụng . . . . .                                 | 15        |
| <b>4</b> | <b>Kế hoạch thực hiện</b>                                   | <b>16</b> |
| 4.1      | Phân công nhân lực . . . . .                                | 16        |
| 4.2      | Timeline . . . . .  | 16        |
| 4.3      | Chi phí thực hiện (chi phí thiết bị, chi phí làm) . . . . . | 17        |
| <b>5</b> | <b>So sánh</b>  | <b>18</b> |
| 5.1      | Bảng tổng hợp khác biệt . . . . .                           | 18        |
| 5.2      | Tóm tắt định lượng . . . . .                                | 18        |
| <b>6</b> | <b>Kiểm thử và Đánh giá</b>                                 | <b>19</b> |
| 6.1      | Mục tiêu kiểm thử . . . . .                                 | 19        |
| 6.2      | Phương pháp kiểm thử . . . . .                              | 19        |
| 6.3      | Kết quả đánh giá . . . . .                                  | 19        |
| 6.3.1    | Các mô hình AI . . . . .                                    | 19        |
| 6.3.2    | Thời gian xử lý . . . . .                                   | 19        |
| 6.4      | Đánh giá chung . . . . .                                    | 20        |

|          |  |           |
|----------|--|-----------|
| <b>7</b> | <b>Kết luận, Hạn chế và Hướng phát triển</b> | <b>21</b> |
| 7.1      | Kết luận . . . . .                           | 21        |
| 7.2      | Hạn chế . . . . .                            | 21        |
| 7.3      | Hướng phát triển . . . . .                   | 21        |

# Chương 1

## Giới thiệu

### 1.1 Bối cảnh

Trong bối cảnh đô thị hóa nhanh chóng, việc quản lý bãi đỗ xe tại các khu vực đông dân cư như chung cư, trường học, tòa nhà văn phòng đang gặp nhiều thách thức. Các hệ thống hiện tại sử dụng thẻ từ (RFID) và phần mềm quản lý cơ bản vẫn còn nhiều hạn chế:

- **Thiếu tự động hóa:** Phụ thuộc vào thao tác thủ công, dễ sai sót và tốn thời gian.
- **Thiếu thông tin thời gian thực:** Không giám sát được tình trạng từng vị trí đỗ, gây bất tiện cho người dùng.
- **Lưu trữ dữ liệu cục bộ:** Khó khăn trong truy xuất và phân tích dữ liệu khi cần thiết.

Công nghệ AIoT (Artificial Intelligence of Things) với sự kết hợp giữa AI và IoT mở ra tiềm năng lớn để giải quyết những thách thức này, hướng tới xây dựng hệ thống quản lý bãi đỗ xe thông minh, tự động và hiệu quả.

### 1.2 Mục đích

Đề tài này nhằm mục đích nghiên cứu, thiết kế và xây dựng một mô hình thử nghiệm (prototype) **Hệ thống Bãi đỗ xe Thông minh** ứng dụng công nghệ AIoT. Hệ thống hướng đến việc tự động hóa toàn diện quy trình quản lý phương tiện ra vào, giảm thiểu sự can thiệp của con người, đồng thời nâng cao trải nghiệm người dùng và hiệu quả vận hành cho đơn vị quản lý.

### 1.3 Mục tiêu

Để đạt được mục đích đã đề ra, hệ thống tập trung vào ba nhóm mục tiêu chính. Mỗi mục tiêu được cụ thể hóa bằng các tiêu chí định lượng nhằm đảm bảo khả năng kiểm chứng trong quá trình phát triển và đánh giá hệ thống.

| ID   | Mục tiêu   | Chỉ số đo   | Điều kiện đạt   |
|------|--|---|---|
| 0-01 | Xây dựng chức năng nhận diện gương mặt và biển số xe hoạt động ổn định trong điều kiện ban ngày và ảnh nhòe/không rõ nét | Accuracy $\geq 90\%$ (ban ngày);<br>Accuracy $\geq 70\%$ (ảnh nhòe)           | Test trên $\geq 100$ lượt xe vào/ra với dữ liệu đa dạng, gồm ảnh rõ và ảnh nhòe |
| 0-02 | Đảm bảo hệ thống phản hồi nhanh và cập nhật thời gian thực   | Thời gian xử lý 1 lượt $\leq 15$ giây; Cập nhật trạng thái slot $\leq 5$ giây | Môi trường test chuẩn, kết nối mạng ổn định                                     |

| ID   | Mục tiêu  | Chỉ số đo   | Điều kiện đạt                              |
|------|---|---|--|
| 0-03 | Hoàn thiện đầy đủ tính năng quản lý bãi xe quy mô nhỏ (5–10 vị trí) | 100% module hoạt động; tự động ghi nhận biển số & thời gian; lưu trữ cloud; dashboard trực quan | Đáp ứng đầy đủ yêu cầu vận hành thử nghiệm |

## 1.4 Phát biểu bài toán

### 1.4.1 Dữ liệu đầu vào

Hình ảnh chất lượng cao từ camera giám sát tại các lối vào và lối ra của bãi xe.

### 1.4.2 Kết quả đầu ra

- Thông tin nhận diện với độ chính xác  $\geq 90\%$  (điều kiện ban ngày) và  $\geq 70\%$  (ảnh nhòe/không rõ nét) cho biển số xe và khuôn mặt tại thời điểm phương tiện vào – ra.
- Bản ghi chi tiết về thời gian vào – ra của từng phương tiện với thời gian xử lý  $\leq 15$  giây/lượt.
- Trạng thái cập nhật liên tục của từng vị trí đỗ (còn trống/đã chiếm chỗ) với thời gian cập nhật  $\leq 5$  giây.
- Giao diện dashboard trực quan, dễ sử dụng cho người quản lý và thông tin cho người dùng.
- Cơ sở dữ liệu được lưu trữ an toàn và có khả năng truy xuất hiệu quả trên nền tảng cloud.

## 1.5 Phạm vi và Giới hạn của Hệ thống

### 1.5.1 Phạm vi triển khai

Hệ thống được thiết kế với định hướng triển khai tại các tổ chức và doanh nghiệp có nhu cầu kiểm soát an ninh và quản lý phương tiện ra vào một cách tự động và hiệu quả, ví dụ như:

- Công ty, tòa nhà văn phòng.
- Nhà máy, khu công nghiệp.
- Trường học, bệnh viện.

Mô hình hoạt động phù hợp với các đơn vị có sẵn hạ tầng mạng cục bộ ổn định và có khả năng đầu tư cơ bản vào thiết bị như ESP32-CAM, ESP32 điều khiển trung tâm và màn hình hiển thị. Ban đầu, hệ thống sẽ được phát triển dưới dạng một mô hình thử nghiệm (prototype) cho một bãi đỗ xe quy mô nhỏ (5-10 chỗ) để kiểm chứng tính khả thi và hiệu quả của các giải pháp công nghệ được lựa chọn.

*Lưu ý: Trong giai đoạn hiện tại, hệ thống chưa tích hợp chức năng thu phí tự động. Tuy nhiên, đây là một hướng phát triển tiềm năng trong tương lai, đặc biệt khi ứng dụng tại các bãi xe công cộng hoặc dịch vụ.*

### 1.5.2 Giới hạn của Hệ thống

Hệ thống nhận diện và quản lý tự động có thể gặp một số giới hạn sau:

- Điều kiện môi trường:** Độ chính xác của module nhận diện hình ảnh (biển số, khuôn mặt) có thể bị ảnh hưởng bởi các yếu tố như ánh sáng yếu (ban đêm, khu vực thiếu sáng), điều kiện thời tiết bất lợi (mưa lớn, sương mù dày đặc), hoặc góc quay camera không tối ưu.

- **Chất lượng camera:** Yêu cầu camera có độ phân giải đủ tốt và được lắp đặt ở vị trí phù hợp để đảm bảo chất lượng hình ảnh đầu vào cho quá trình nhận diện.
- **Tài nguyên phần cứng:** Hệ thống cần được cung cấp đủ tài nguyên xử lý (CPU, RAM), dung lượng lưu trữ và băng thông mạng để đảm bảo hoạt động ổn định và đáp ứng yêu cầu thời gian thực.
- **Độ trễ hệ thống:** Mặc dù mục tiêu là giảm thiểu độ trễ, một khoảng trễ nhất định trong quá trình xử lý dữ liệu và phản hồi là khó tránh khỏi, cần được tối ưu để không ảnh hưởng đến trải nghiệm người dùng.

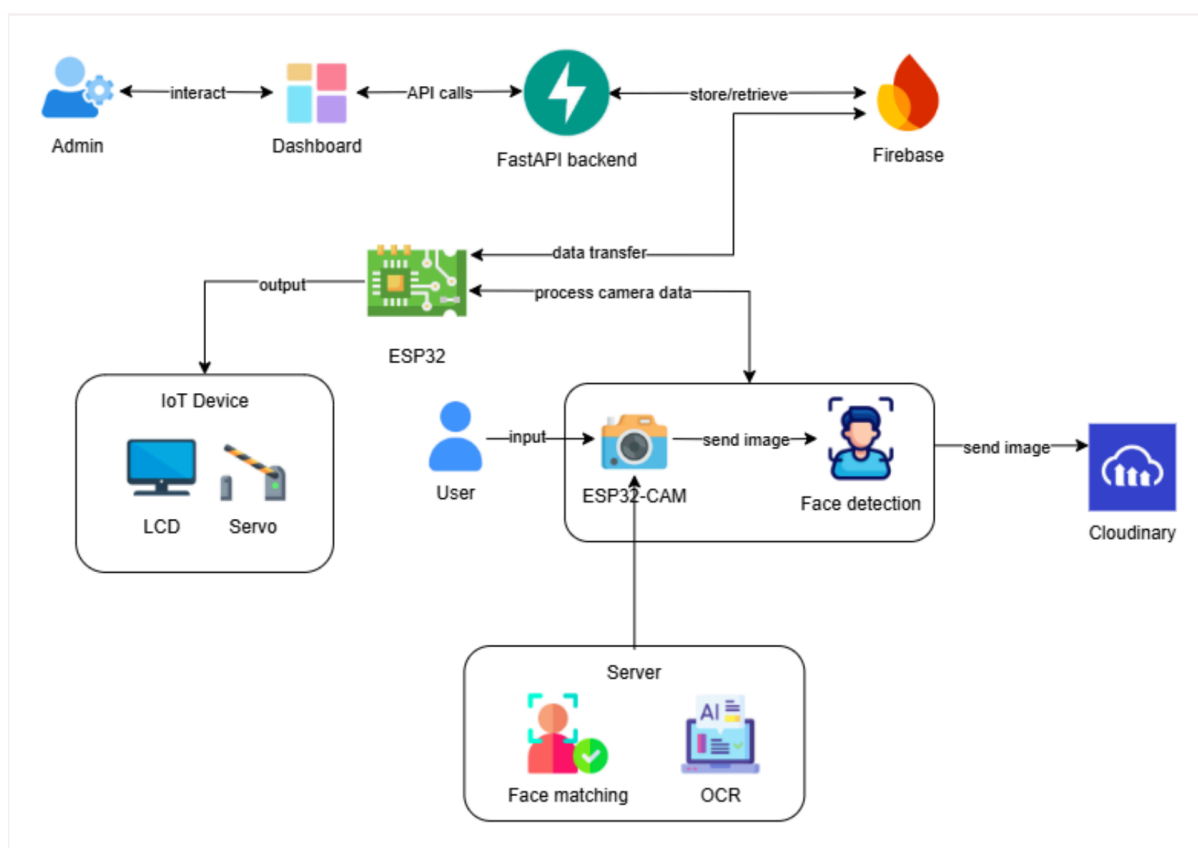
## 1.6 Đóng góp chính của Đề tài

Đề tài dự kiến mang lại những đóng góp khoa học và thực tiễn sau:

- **Đề xuất mô hình hệ thống bãi đỗ xe thông minh toàn diện:** Tích hợp công nghệ AIoT một cách hiệu quả, có khả năng ứng dụng vào thực tế tại Việt Nam.
- **Tự động hóa quy trình vận hành:** Kết hợp nhận diện hình ảnh (biển số/khuôn mặt) để giảm thiểu tối đa sự can thiệp thủ công, nâng cao độ chính xác.
- **Nâng cao hiệu quả quản lý và trải nghiệm người dùng:** Cung cấp công cụ giám sát trực quan cho người quản lý và thông tin hữu ích cho người dùng (ví dụ: tình trạng chỗ trống).
- **Định hướng phát triển và mở rộng:** Tạo tiền đề cho việc triển khai hệ thống tại các loại hình bãi đỗ xe khác nhau (trường học, tòa nhà, trung tâm thương mại) và tích hợp thêm các tính năng nâng cao trong tương lai.
- **Giảm thiểu sai sót và tối ưu hóa nguồn lực:** Góp phần giảm thiểu các lỗi thường gặp trong quản lý thủ công và sử dụng hiệu quả hơn nguồn nhân lực.

## Chương 2

# Thiết kế hệ thống



Hình 2.1. Sơ đồ kiến trúc hệ thống

### 2.1 Kiến trúc phần mềm

Hệ thống được thiết kế dựa trên mô hình phân tầng, bao gồm các thành phần chính: tầng thiết bị đầu vào, tầng xử lý trung tâm, và tầng điều khiển - phản hồi.

#### 2.1.1 Thiết bị đầu vào (Input Layer)

- Người dùng tương tác trực tiếp với hệ thống thông qua camera.
- Camera OV2640 tích hợp trên ESP32-CAM
  - Chụp ảnh biển số và khuôn mặt tại lối vào/ra.
  - Chạy mô hình MTMN (ESP-FACE) để phát hiện khuôn mặt và trả về bounding box.

- Gửi ảnh hợp lệ lên Cloudinary và URL ảnh lên Firebase qua REST API.

### 2.1.2 Tầng xử lý trung tâm (Processing Layer)

Tầng này chịu trách nhiệm xử lý dữ liệu từ thiết bị đầu vào, thực hiện các tác vụ trí tuệ nhân tạo và điều phối hoạt động của hệ thống. Server Backend (FastAPI) chịu trách nhiệm:

- Lấy URL ảnh từ Firebase.
- Nhận diện khuôn mặt: mô hình FaceNet trích xuất embedding và so khớp với cơ sở dữ liệu.
- Nhận dạng biển số: Plate Recognizer API sẽ nhận request là hình ảnh chứa biển số và trả về chuỗi ký tự.
- Ghi kết quả nhận diện (biển số, trạng thái hợp lệ, thời gian) vào Firebase.

### 2.1.3 Thiết bị điều khiển - phản hồi (Output Layer)

Thiết bị IoT gồm:

- **Servo (Barrier):** mở/đóng rào chắn tự động.
- **LCD Display:** hiển thị thông tin trực quan (số ô trống, lỗi, trạng thái...).

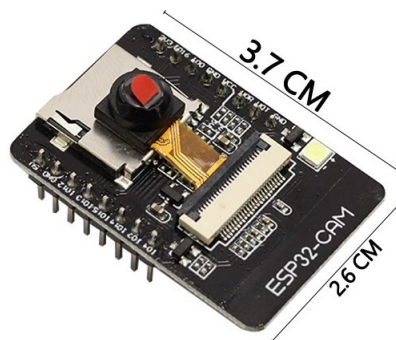
Dashboard (giao diện quản trị cho Admin):

- Kết nối Firebase để lấy dữ liệu thời gian thực.
- Hiển thị danh sách lượt vào/ra, trạng thái slot, thống kê lịch sử.
- Cho phép truy xuất dữ liệu và giám sát hệ thống từ xa.

## 2.2 Hệ thống phần cứng

### 2.2.1 Chi tiết thiết bị

#### 1. ESP32-CAM (AI Thinker ESP32-CAM)



Hình 2.2. ESP32-CAM

ESP32-CAM là một module phát triển nhỏ gọn của hãng **AI. Thinker**, tích hợp sẵn: **Wi-Fi, Bluetooth, camera OV2640, khe cắm thẻ microSD, GPIO** đa năng.

Module này được thiết kế chuyên dụng cho các ứng dụng thị giác máy tính, IoT, nhận diện khuôn mặt, giám sát an ninh, ...

ESP32-CAM là lựa chọn phổ biến nhờ giá rẻ, tính năng mạnh, và khả năng chạy AI trực tiếp trên thiết bị.



| Thành phần              | Thông tin chi tiết   | Ghi chú  |
|-------------------------|--|--|
| <b>Vi xử lý chính</b>   | ESP32-WROOM-32   | Xử lý chương trình nạp vào   |
| <b>Tốc độ xung nhịp</b> | Lên tới 240 MHz  | Xử lý được <b>240 triệu lệnh mỗi giây</b>  |
| <b>RAM</b>              | 520 KB SRAM nội  | Lưu dữ liệu tạm thời khi chương trình đang chạy, tự động mất khi tắt nguồn   |
| <b>Flash</b>            | 4 MB (SPI flash)   | Lưu trữ chương trình được nạp vào chip, không mất khi tắt nguồn  |
| <b>Camera</b>           | OV2640, độ phân giải tối đa 1600x1200 (UXGA), hỗ trợ nhiều chế độ JPEG | Có LED flash trắng tích hợp (nối chân GPIO nếu dùng), không tự động lấy nét, hình bị mờ nếu đặt sai khoảng cách.             |
| <b>GPIO</b>             | ~9 chân GPIO khả dụng  | Các chân đa năng trên chip, cho phép nhận tín hiệu vào và gửi tín hiệu ra  |
| <b>Nguồn hoạt động</b>  | 3.3V   | Khi dùng nguồn 5V, cấp nguồn vào chân VCC/5V, bo mạch ESP32-CAM có sẵn mạch ổn áp AMS1117, chuyển 5V thành 3.3V cấp cho chip |
| <b>Dòng tiêu thụ</b>    | 160–250mA khi hoạt động  | Nên dùng nguồn ổn định từ MB102 hoặc pin sạc 5V - 1A trở lên   |
| <b>Cổng USB</b>         | Không có   | Cần dùng <b>USB to UART</b> để nạp code  |
| <b>Kích thước</b>       | 27 x 40.5 x 4.5 mm   |  |

Hệ thống sử dụng **2 module ESP32-CAM** tại cổng:

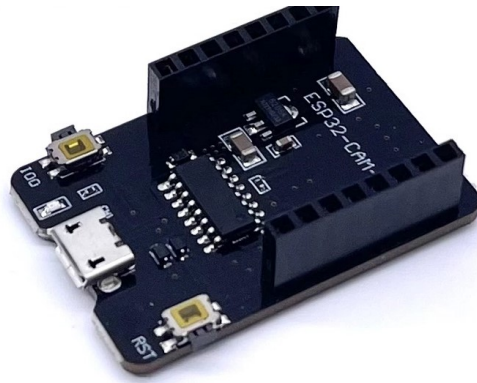
- Một module quét và nhận diện khuôn mặt tài xế.
- Một module quét và nhận diện biển số xe.

Mỗi ESP32-CAM hoạt động độc lập, được nạp sẵn chương trình, luôn bật camera và model nhận diện, có khả năng:

- Nhận diện sự xuất hiện của xe
- Tự động chụp ảnh
- Thực hiện nhận diện khuôn mặt hoặc biển số trực tiếp trên thiết bị
- Truyền tải thông tin:
  - Gửi hình chụp khuôn mặt và biển số xe tới **Cloudinary** qua giao thức **HTTP POST**.
  - Nhận **url** hình ảnh từ **Cloudinary**, thực hiện nhận diện khuôn mặt và gửi gói tin gồm **url** và kết quả nhận diện vừa xử lý về **Firebase**.

ESP32 trung tâm sẽ kết nối chung mạng Wi-Fi với 2 ESP32-CAM, đóng vai trò nhận thông tin từ các module này. Sau khi nhận được kết quả nhận diện thông qua Wi-Fi, ESP32 trung tâm xử lý và điều khiển các thiết bị output liên quan như màn hình OLED và servo.

Nạp code:

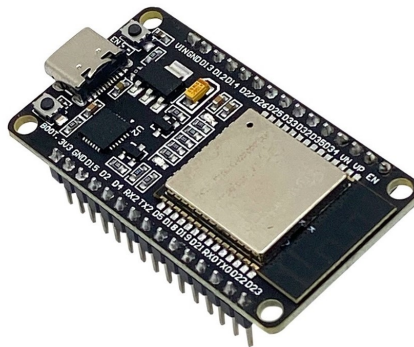


Hình 2.3. Đế nạp ESP32-CAM

- Vì ESP32-CAM không có cổng USB, nên ta cần thiết bị trung gian để nạp code là **Đế Nạp ESP32-CAM**, sau khi nạp ESP32-CAM sẽ chạy độc lập và không mất code khi tắt nguồn. Đây là module chuyển đổi tín hiệu USB và UART, cho phép kết nối giữa máy tính và các thiết bị không có cổng USB trực tiếp như ESP32-CAM.

Lưu ý khi vừa nạp code, ESP32-CAM đang ở chế độ bootloader, cần ấn nút *reset* để chip khởi động lại chế độ chạy bình thường.

## 2. ESP32 trung tâm (ESP32 DevKit V1)



Hình 2.4. ESP32 DevKit V1

ESP32 trung tâm là **bộ điều khiển chính** của toàn hệ thống, xử lý và điều phối giữa các thiết bị. Board sử dụng **chip ESP32-WROOM-32**, có thể xử lý nhiều tác vụ song song.

| Thành phần    | Thông tin chi tiết | Ghi chú  |
|---------------|--------------------|--|
| Chip chính    | ESP32-WROOM-32     |  |
| RAM           | 520 KB             |  |
| Flash         | 4MB SPI Flash      |  |
| Tốc độ xử lý  | Lên đến 240MHz     |  |
| Điện áp logic | 3.3V               | Thiết bị gửi tín hiệu 5V vào ESP32 GPIO có thể gây cháy GPIO, cần sử dụng mạch chia áp |

| Thành phần        | Thông tin chi tiết                                       | Ghi chú  |
|-------------------|--|--|
| USB-to-Serial     | Tích hợp sẵn CP2102                                      | Có thể <b> nạp code trực tiếp qua cổng USB-C / microUSB</b>                            |
| GPIO khả dụng     | ~25 chân, đủ dùng cho nhiều ngoại vi đồng thời           |  |
| Wi-Fi / Bluetooth | Tích hợp sẵn, dùng được ở chế độ Station và Access Point | Với ESP32 và 4 ESP32-CAM, ta sẽ dùng chế độ Station, nghĩa là cùng kết nối wifi có sẵn |

### 3. Breadboard

Là bảng mạch kết nối không cần hàn, được sử dụng để phân phối nguồn điện và kết nối các thiết bị với nhau.

Cho phép kết nối linh kiện một cách linh hoạt thông qua dây jumper, giúp xây dựng và tổ chức hệ thống điện tử một cách gọn gàng, dễ bảo trì và dễ mở rộng.

Trong hệ thống này, breadboard sẽ được cấp nguồn trực tiếp từ adapter 5V – 3A thông qua cáp chia nguồn, sử dụng dây jumper để kết nối các thành phần trong mạch breadboard.

| Loại dây      | Đầu cắm                 | Ứng dụng  |
|---------------|-------------------------|---|
| Male-Male     | Đầu cắm kim 2 bên       | Dùng để cắm từ chân này sang chân khác <b>trên breadboard</b> hoặc từ <b>breadboard đến board mạch</b> (ESP32, MB102...)            |
| Male-Female   | Một đầu kim, một đầu lỗ | Dùng để nối <b>module/cảm biến (có chân đực)</b> với <b>breadboard hoặc board mạch</b>  |
| Female-Female | Hai đầu lỗ              | Dùng để <b>nối giữa 2 thiết bị đều có chân đực</b> , ví dụ <b>ESP32-CAM và FTDI</b> , hoặc giữa <b>module logic với module khác</b> |

### 4. Màn hình LCD

Là màn hình **LCD 16x2** sử dụng giao tiếp **I2C (SDA, SCL)** với ESP32 trung tâm (thường là GPIO21 và GPIO22). Màn hình có module chuyển đổi I2C tích hợp giúp đơn giản hóa kết nối và tiết kiệm chân GPIO.

Trong hệ thống này, LCD có chức năng hiển thị:

- **Khi xe vào:** Hiển thị số chỗ còn trống hoặc thông báo “Đã hết chỗ” nếu không còn slot đỗ.
- **Khi xe ra:** Hiển thị lời chào (VD: “Tạm biệt”) hoặc hiển thị lỗi nếu quá trình nhận diện không hợp lệ.

Điện áp hoạt động là **5V** (tương thích tốt với ESP32 thông qua module I2C).









### 5. Servo motor SG90

**SG90** là một loại **servo mini phổ biến** với cấu trúc bánh răng nhựa, trọng lượng nhẹ, phù hợp với các cơ cấu điều khiển như **thanh chắn xe** trong mô hình nhỏ.

Servo hoạt động ở **điện áp 5V**, tiêu thụ dòng khoảng **100–250mA khi tải nhẹ**, do đó vẫn cần cấp nguồn ổn định để đảm bảo hoạt động liên tục và không ảnh hưởng đến ESP32.

**ESP32 trung tâm điều khiển servo qua tín hiệu PWM** từ một chân GPIO bất kỳ (thường dùng GPIO13 hoặc GPIO14). Tín hiệu PWM xác định góc quay của servo trong khoảng từ **0° đến 180°**, dùng để mở hoặc đóng thanh chắn.

### 2.2.2 Tổng hợp

| STT | Tên thiết bị                | Số lượng | Giá/cái | Hình ảnh   | Nguồn liên kết              |
|-----|-----------------------------|----------|---------|--|-----------------------------|
| 1   | <b>AI Thinker ESP32-CAM</b> | 2        | 165,000 |    | <a href="#">Xem tại đây</a> |
| 2   | <b>ESP32 DevKit V1</b>      | 1        | 154,000 |    | <a href="#">Xem tại đây</a> |
| 3   | <b>Màn hình LCD</b>         | 1        | 81,000  |    | <a href="#">Xem tại đây</a> |
| 4   | <b>Servo motor SG90</b>     | 1        | 69,000  |    | <a href="#">Xem tại đây</a> |
| 5   | <b>Adapter 5V - 3A</b>      | 1        | 52,000  |  | <a href="#">Xem tại đây</a> |
| 6   | <b>Breadboard</b>           | 1        | 20,000  |  | <a href="#">Xem tại đây</a> |
| 7   | <b>Dây jumper</b>           | 40 sợi   | 66,000  |  | <a href="#">Xem tại đây</a> |
| 8   | <b>Đế Nạp ESP32-CAM</b>     | 1        | 30,000  |  | <a href="#">Xem tại đây</a> |

## 2.3 Luồng thực thi hệ thống

### 1. Phát hiện & chụp ảnh

- ESP32-CAM liên tục quét khung hình tại lối vào/ra.
- Khi phát hiện có khuôn mặt (MTMN – ESP-FACE), thiết bị sẽ chụp ảnh.

### 2. Upload ảnh & gửi URL

- ESP32-CAM upload ảnh lên Cloudinary qua REST API (server).

- Cloudinary trả về URL ảnh.
- ESP32-CAM gửi URL ảnh + metadata (`id cam` và `isFace`) cho ESP32 trung tâm qua giao thức (TCP/IP socket).

### 3. Gọi AI model

- ESP32 trung tâm nhận URL ảnh từ ESP32-CAM.
- Gọi API AI (server) với URL để thực hiện:
  - Nhận dạng biển số (luồng vào/ra) bằng Plate Recognizer API → trích xuất ký tự biển số.
  - Nhận diện khuôn mặt (luồng ra) bằng FaceNet → xác định danh tính.

### 4. Nhận & xử lý kết quả

- API AI model trả về dữ liệu nhận diện (tên, biển số, trạng thái hợp lệ/không hợp lệ).
- ESP32 trung tâm ghi kết quả vào Firebase để lưu trữ và đồng bộ với Dashboard.

### 5. Điều khiển thiết bị

- Dựa trên kết quả nhận diện, ESP32 trung tâm điều khiển Servo (Barrier), LCD.

### 6. Hiện thị & giám sát

- Dashboard (ReactJS) kết nối Firebase để hiển thị lịch sử vào/ra, ảnh chụp, biển số, trạng thái slot, và thống kê theo thời gian thực.

## Chương 3

# Triển khai và Công Nghệ Sử dụng



Hình 3.1. Prototype mô hình toàn bộ hệ thống

### 3.1 Quy trình triển khai

Hệ thống được triển khai theo quy trình có hệ thống gồm bốn giai đoạn chính:

| Giai đoạn | Tên giai đoạn                  | Mô tả   |
|-----------|--------------------------------|---|
| 1         | Lắp đặt phần cứng              | Camera ESP32-CAM được bố trí tại các điểm lối vào và lối ra. ESP32 trung tâm điều khiển barrier được kết nối với màn hình LCD và servo.   |
| 2         | Cài đặt firmware cho ESP32-CAM | Firmware được tích hợp mô hình MTMN (ESP-FACE) để thực hiện phát hiện khuôn mặt trực tiếp trên thiết bị. Hệ thống được cấu hình để upload ảnh lên Cloudinary thông qua REST API và gửi URL ảnh cùng metadata đến ESP32 trung tâm. |
| 3         | Cấu hình ESP32 trung tâm       | ESP32 trung tâm nhận URL ảnh từ camera và thực hiện gọi API đến các mô hình AI đã được deploy trên server để nhận diện khuôn mặt và biển số xe. Kết quả xử lý được lưu trữ vào Firebase cho việc truy xuất và phân tích.          |

| Giai đoạn | Tên giai đoạn        | Mô tả   |
|-----------|----------------------|---|
| 4         | Triển khai Dashboard | Giao diện Dashboard được phát triển bằng ReactJS, kết nối trực tiếp với Firebase để hiển thị dữ liệu thời gian thực, thống kê hoạt động và lịch sử ra vào của hệ thống. |

## 3.2 Công nghệ sử dụng

| Thành phần                     | Công nghệ / Công cụ        | Mục đích   |
|--------------------------------|----------------------------|--|
| <b>Thiết bị nhúng</b>          | ESP32-CAM, ESP32 DevKit V1 | Chụp ảnh, điều khiển barrier, giao tiếp với server |
| <b>Phát hiện khuôn mặt</b>     | MTMN – ESP-FACE            | Xác định vị trí khuôn mặt trên ảnh                 |
| <b>Nhận diện khuôn mặt</b>     | FaceNet                    | So khớp với cơ sở dữ liệu người dùng               |
| <b>Nhận dạng biển số (OCR)</b> | Plate Recognizer API       | Trích xuất ký tự từ ảnh biển số                    |
| <b>Backend</b>                 | FastAPI (Python)           | Xử lý API, gọi AI model, quản lý logic hệ thống    |
| <b>Cloud Storage</b>           | Cloudinary                 | Lưu trữ ảnh, trả về URL                            |
| <b>Database</b>                | Firestore DB               | Lưu trữ kết quả nhận diện, đồng bộ dữ liệu         |
| <b>Dashboard</b>               | ReactJS                    | Hiển thị dữ liệu và giám sát                       |
| <b>Giao tiếp</b>               | HTTP REST, TCP/IP          | Truyền dữ liệu và điều khiển thời gian thực        |

## Chương 4

# Kế hoạch thực hiện

### 4.1 Phân công nhân lực

| Thành viên                | Vai trò chính                        | Nhiệm vụ cụ thể  |
|---------------------------|--------------------------------------|--|
| <b>Cao Uyển Nhi</b>       | Trưởng nhóm, phụ trách AI            | Quản lý tiến độ, phát triển và tích hợp mô hình AI (OCR, face), đánh giá hiệu năng |
| <b>Trần Thị Cát Tường</b> | Thiết kế phần cứng – ESP32 trung tâm | Phát triển mạch điều khiển, giao tiếp servo và truyền dữ liệu lên cloud            |
| <b>Lưu Thanh Thuý</b>     | Phụ trách ESP32-CAM                  | Thiết lập, triển khai face detection, giao tiếp giữa các ESP32-CAM với nhau        |
| <b>Võ Lê Việt Tú</b>      | Phụ trách backend & dashboard        | Xây dựng dashboard web, kết nối backend với phần cứng và lưu trữ dữ liệu           |

### 4.2 Timeline

| Thời gian   | Công việc chính  |
|-------------|--|
| 17/5 - 23/5 | Nghiên cứu tài liệu, xác định yêu cầu, phân công nhiệm vụ  |
| 23/6 - 28/6 | Thiết kế kiến trúc hệ thống và lựa chọn công nghệ          |
| 10/6 – 5/7  | Xây dựng prototype và thử nghiệm                           |
| 10/6 – 15/7 | Bảo mật và hệ thống dữ liệu                                |
| 10/6 - 20/7 | Lắp ghép và triển khai phần cứng                           |
| 25/7 - 30/7 | Kiểm thử và đánh giá hiệu năng                             |
| 30/7 - 2/8  | Viết báo cáo, chuẩn bị slide thuyết trình, hoàn thiện demo |

Cụ thể, xem timeline dạng Gantt Chart tại [đây](#)



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

| Tên thiết bị                     | Số lượng    | Đơn giá | Thành tiền       |
|----------------------------------|-------------|---------|------------------|
| AI Thinker ESP32-CAM             | 2           | 165,000 | 330,000          |
| ESP32 DevKit V1 (CP2102, Type-C) | 1           | 154,000 | 154,000          |
| Màn hình LCD                     | 1           | 81,000  | 81,000           |
| Servo motor SG90                 | 1           | 69,000  | 69,000           |
| Adapter 5V - 3A                  | 1           | 52,000  | 52,000           |
| Breadboard                       | 1           | 20,000  | 20,000           |
| Dây jumper                       | 40 sợi/loại | 66,000  | 66,000           |
| Đế Nạp ESP32-CAM                 | 1           | 30,000  | 30,000           |
| Chi phí lắp đặt, khung mô hình   | -           | 200,000 | 200,000          |
| <b>Tổng chi phí</b>              |             |         | <b>1,002,000</b> |

## Chương 5

# So sánh

### 5.1 Bảng tổng hợp khác biệt

| Nội dung                | Trước (Spec 1)                                 | Sau (Spec mới)   | Mức thay đổi        |
|-------------------------|--|--|---------------------|
| Mục tiêu cốt lõi        | Giữ nguyên                                     | Giữ nguyên   | –                   |
| Độ chính xác ban ngày   | 80%  | 90%  | ↑ ~12%              |
| Số lượt test            | 6 lượt   | 100 lượt đa dạng   | ↑ nhiều             |
| Thời gian xử lý         | 5s   | 15s  | Nới nhẹ             |
| Thời gian cập nhật slot | 3s   | 5s   | Nới nhẹ             |
| Mô hình AI              | ESP-WHO, MobileFaceNet                         | ESP-FACE MTMN, FaceNet                                   | Đổi công nghệ       |
| Màn hình                | OLED 0.96”                                     | LCD 16×2   | Đổi linh kiện       |
| Servo                   | MG90S  | SG90   | Đổi linh kiện       |
| Luồng xử lý             | ESP32-CAM → Cloudinary → Firebase → Backend AI | ESP32-CAM → Cloudinary → ESP32 trung tâm → AI → Firebase | Thêm bước điều phối |

### 5.2 Tóm tắt định lượng

- **Điểm giữ nguyên:** mục tiêu cốt lõi, mô hình 3 tầng, chức năng chính.
- **Điều chỉnh chính:** tăng nhẹ yêu cầu độ chính xác, mở rộng quy mô test, nới thời gian xử lý để phù hợp thực tế.
- **Thay đổi công nghệ:** cập nhật mô hình AI, thay một số linh kiện, tinh chỉnh luồng xử lý để tăng khả năng điều phối.
- **Tác động:** thay đổi mang tính tinh chỉnh và tối ưu, không ảnh hưởng đến định hướng ban đầu.

## Chương 6

# Kiểm thử và Đánh giá

### 6.1 Mục tiêu kiểm thử

Quá trình kiểm thử được thực hiện nhằm:

- 0-01: Đánh giá độ chính xác nhận diện biển số xe và khuôn mặt.
- 0-02: Đo thời gian xử lý và khả năng cập nhật trạng thái chỗ đỗ.
- 0-03: Xác nhận độ hoàn thiện chức năng quản lý bãi xe.

### 6.2 Phương pháp kiểm thử

- **Đánh giá mô-đun (Module-level Evaluation):** Sử dụng tập dữ liệu test để tính độ chính xác, precision, recall cho từng mô hình AI.
- **Đo thời gian xử lý thực tế:** Ghi nhận thời gian trung bình từ khi xe vào vùng nhận diện đến khi barrier mở (session).

### 6.3 Kết quả đánh giá

#### 6.3.1 Các mô hình AI

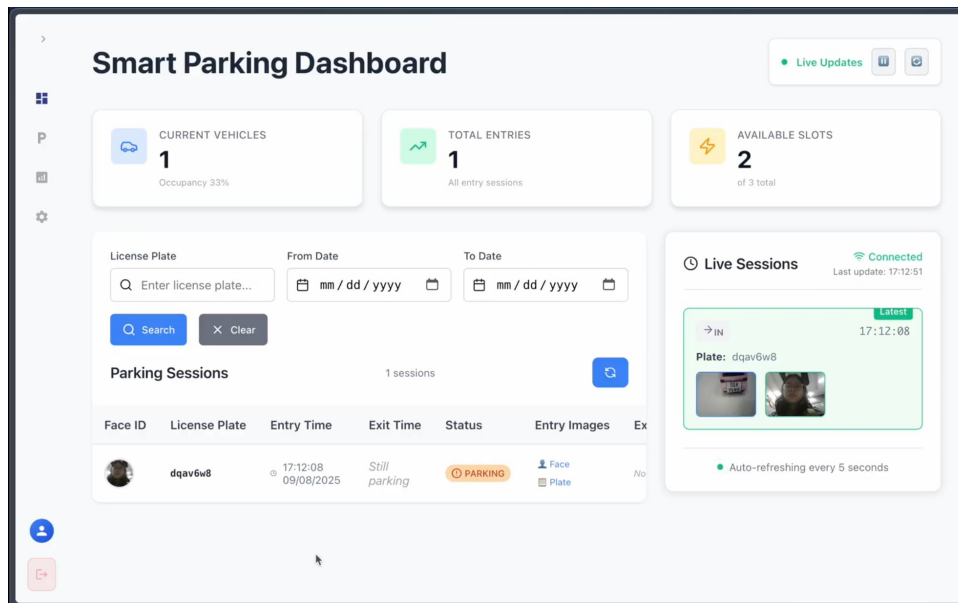
| Mô-đun AI                      | Nguồn Dataset  | Kích thước test                          | Precision | Recall | Accuracy | F1-score | Ghi chú                    |
|--------------------------------|--|--|-----------|--------|----------|----------|----------------------------|
| <b>Face Detection (MTMN)</b>   | 285 ảnh từ ESP32-CAM (thu thực tế)                                       | 285 ảnh                                  | 1.0000    | 0.9439 | —        | —        | TP=269, FP=0, FN=16        |
| <b>Face Matching (FaceNet)</b> | <a href="#">LFW Dataset</a>  | 500 cặp (300 cùng người, 200 khác người) | 0.970     | 0.967  | 0.962    | 0.968    | Ngưỡng tối ưu 0.576        |
| <b>OCR Biển số xe</b>          | <a href="#">Vietnam License Plate Dataset</a> – <a href="#">Roboflow</a> | 101 ảnh                                  | —         | —      | 1.000    | —        | Nhận diện đúng 101/101 ảnh |

#### 6.3.2 Thời gian xử lý

- Thời gian xử lý trung bình:  $\approx 12$  giây cho một session xe vào hoặc ra (từ lúc phát hiện đến khi barrier mở).

## 6.4 Đánh giá chung

- 0-01: Các mô-đun AI đạt độ chính xác cao:
  - Face Detection: Precision 1.0, Recall  $\approx 94\%$
  - Face Matching (LFW dataset): Accuracy  $\approx 96.2\%$
  - OCR Biển số (Vietnam License Plate Dataset): Accuracy 100%
- 0-02: Thời gian xử lý trung bình 12s.
- 0-03: Hệ thống có đầy đủ chức năng chính (nhận diện, ghi log, lưu cloud, dashboard, điều khiển barrier).



Hình 6.1. Màn hình Dashboard quản lý

## Chương 7

# Kết luận, Hạn chế và Hướng phát triển

### 7.1 Kết luận

Nghiên cứu đã triển khai thành công hệ thống bãi đỗ xe thông minh sử dụng công nghệ AIoT, bao gồm các module ESP32/ESP32-CAM, dịch vụ xử lý trung tâm, lưu trữ đám mây và giao diện quản lý.

**Kết quả thực nghiệm:**

- Nhận diện khuôn mặt: Precision = 1.0, Recall = 0.94
- Khớp khuôn mặt: Accuracy = 96.2%, F1-score = 96.8%
- Nhận diện biển số: Accuracy = 100%
- Thời gian xử lý trung bình: 12 giây
- Hệ thống hoạt động ổn định với các chức năng cơ bản

### 7.2 Hạn chế

- Hiệu suất giảm đáng kể với ảnh chất lượng thấp hoặc điều kiện ánh sáng kém
- Ngưỡng nhận diện cần được điều chỉnh dựa trên dữ liệu thực tế
- Quy mô dữ liệu huấn luyện còn hạn chế
- Thiếu các tính năng thu phí tự động và quản lý đa cổng vào

### 7.3 Hướng phát triển

- **Nâng cao độ chính xác:** Mở rộng bộ dữ liệu với mẫu Việt Nam, cải thiện thuật toán xử lý ảnh
- **Tối ưu hiệu năng:** Giảm thời gian phản hồi xuống dưới 10 giây
- **Tăng độ tin cậy:** Bổ sung cơ chế backup và monitoring hệ thống
- **Mở rộng chức năng:** Tích hợp thanh toán tự động và hỗ trợ nhiều camera đồng thời