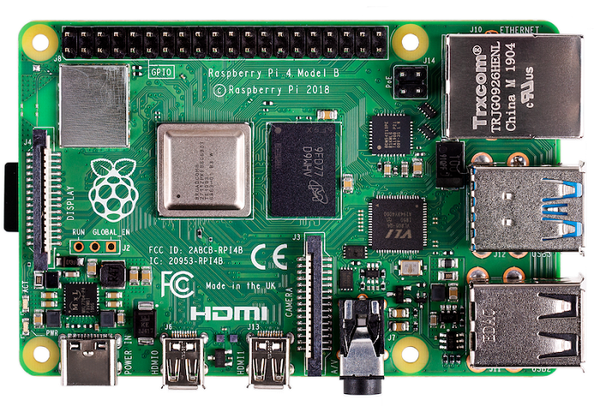
**Thiết kế hệ thống kiểm soát ra vào bãi gửi xe ô tô thông minh**

**I. Phần cứng**

Hệ thống được xây dựng dựa trên nền tảng **Raspberry Pi 4**, một máy tính nhúng nhỏ gọn nhưng mạnh mẽ, phù hợp cho các ứng dụng yêu cầu khả năng xử lý cao và linh hoạt trong triển khai. Dưới đây là các thông số kỹ thuật chính của Raspberry Pi 4 được sử dụng trong hệ thống:



* **Bộ xử lý (CPU):** Quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz, cung cấp hiệu suất xử lý mạnh mẽ, giúp thực thi các tác vụ nhanh chóng và ổn định.
* **Dung lượng RAM:** 8GB LPDDR4, đảm bảo khả năng xử lý dữ liệu hiệu quả, hỗ trợ đa nhiệm và chạy các ứng dụng phức tạp một cách mượt mà.
* **Dung lượng lưu trữ:** 64GB (sử dụng thẻ nhớ microSD), giúp hệ thống có đủ không gian để lưu trữ dữ liệu, cài đặt hệ điều hành và các ứng dụng cần thiết.
* **Cổng kết nối:**

**- USB:** 2 cổng USB 3.0 và 2 cổng USB 2.0, hỗ trợ kết nối các thiết bị ngoại vi như bàn phím, chuột, ổ cứng ngoài.

**- HDMI:** 2 cổng micro HDMI hỗ trợ đầu ra video độ phân giải cao, cho phép hiển thị hình ảnh trên màn hình với chất lượng tốt.

**- Ethernet:** Cổng Gigabit Ethernet, cung cấp kết nối mạng nhanh và ổn định.

**- Wi-Fi & Bluetooth:** Hỗ trợ Wi-Fi băng tần kép (2.4GHz và 5GHz) và Bluetooth 5.0, giúp hệ thống giao tiếp không dây linh hoạt.

### ****2. Cảm biến****

Hệ thống sử dụng các cảm biến để thu thập dữ liệu, hỗ trợ nhận diện xe và kiểm soát quá trình xe ra/vào. Cụ thể, hệ thống bao gồm các loại cảm biến chính sau:

#### **2.1. Camera nhận diện xe – biển số xe**

* **Loại camera:** Camera có tích hợp đèn hồng ngoại.





 **Chức năng:**

* Chụp ảnh phương tiện và biển số xe để phục vụ quá trình xử lý nhận diện biển số.
* Đèn hồng ngoại giúp cải thiện chất lượng hình ảnh trong điều kiện ánh sáng yếu hoặc ban đêm, đảm bảo dữ liệu đầu vào có độ chính xác cao.

 **Ưu điểm:**

* Giúp hệ thống hoạt động ổn định cả ngày lẫn đêm.
* Hình ảnh rõ nét, hỗ trợ tốt cho thuật toán nhận diện biển số xe.

#### **2.2. Cảm biến khoảng cách (hoặc cảm biến hồng ngoại)**

* **Loại cảm biến:** Có thể sử dụng **cảm biến khoảng cách siêu âm** hoặc **cảm biến hồng ngoại** để phát hiện sự có mặt của xe.

* **Chức năng:**
  + Xác định chính xác thời điểm xe ra/vào vị trí kiểm soát.
  + Kết hợp với camera để đảm bảo dữ liệu thu thập chính xác và tránh sai sót khi xe di chuyển.
* **Nguyên lý hoạt động:**
  + Cảm biến siêu âm hoạt động dựa trên sóng siêu âm, đo khoảng cách giữa cảm biến và vật thể (xe).
  + Cảm biến hồng ngoại sử dụng ánh sáng hồng ngoại để phát hiện vật thể trong vùng quét.
* **Ưu điểm:**
  + Giúp tự động phát hiện xe mà không cần can thiệp thủ công.
  + Nâng cao độ chính xác của hệ thống kiểm soát ra/vào.

### ****3. Các thiết bị ngoại vi****

Hệ thống được trang bị các thiết bị ngoại vi nhằm hỗ trợ hoạt động ổn định, nâng cao hiệu quả quản lý bãi đỗ xe.

#### **3.1. Hệ thống Internet**

* **Yêu cầu:** Đảm bảo tốc độ mạng **ổn định trên 300Mbps**, giúp hệ thống truyền tải dữ liệu nhanh chóng và liên tục.
* **Kết nối:**
  + Ưu tiên sử dụng **mạng Ethernet có dây** để đảm bảo độ ổn định và giảm thiểu độ trễ so với Wi-Fi.
  + Nếu sử dụng Wi-Fi, cần có bộ phát sóng công suất cao, tránh tình trạng mất kết nối.
* **Tác dụng:**
  + Truyền dữ liệu hình ảnh từ camera đến hệ thống xử lý.
  + Đồng bộ dữ liệu giữa hệ thống nhúng và máy chủ điều khiển.

#### **3.2. Hệ thống biển báo LED**



- **Chức năng:**

* Hiển thị trạng thái vị trí đỗ xe (còn trống, đã có xe).
* Hướng dẫn phương tiện di chuyển trong bãi đỗ, giảm ùn tắc và tăng hiệu quả vận hành.

- **Loại LED:**

* LED ma trận hoặc LED 7 đoạn, có thể điều khiển hiển thị bằng vi điều khiển.

- **Ưu điểm:**

* Giúp tài xế dễ dàng tìm kiếm vị trí đỗ.
* Nâng cao trải nghiệm người dùng, giảm thời gian tìm chỗ đỗ xe.

#### **3.3. Hệ thống điện**

* **Nguồn điện chính:** Hệ thống hoạt động chủ yếu dựa vào **điện lưới**.
* **Nguồn điện dự phòng:**
  + Trang bị **bộ lưu điện UPS** hoặc **máy phát điện** để đảm bảo hệ thống vẫn hoạt động trong trường hợp mất điện.
  + Giúp duy trì hoạt động của camera, cảm biến, biển báo LED và bộ xử lý.

### ****4. Giám sát hệ thống****

* **Yêu cầu:** Ít nhất **một giám sát viên** chịu trách nhiệm theo dõi và kiểm soát hoạt động của hệ thống.
* **Chức năng:**
  + Kiểm tra tình trạng hoạt động của camera, cảm biến và các thiết bị liên quan.
  + Xử lý các tình huống sự cố (mất kết nối, lỗi nhận diện, mất điện).
  + Hỗ trợ người dùng khi cần thiết.

### ****II. Phần mềm****

#### **2.1. Độ tin cậy**

Phần mềm của hệ thống được thiết kế để đảm bảo hoạt động ổn định trong thời gian dài, không bị gián đoạn hoặc gặp lỗi nghiêm trọng. Để đạt được điều này, hệ thống phần mềm cần đáp ứng các tiêu chí sau:

* **Khả năng hoạt động liên tục:**
  + Phần mềm được tối ưu hóa để giảm thiểu tình trạng **treo (crash)** hoặc **đơ (freeze)** khi chạy trong thời gian dài.
  + Sử dụng cơ chế **quản lý bộ nhớ hiệu quả** để tránh rò rỉ bộ nhớ (memory leak).
* **Xử lý lỗi tự động:**
  + Hệ thống có các cơ chế **phát hiện và khôi phục lỗi**, giúp phần mềm tiếp tục hoạt động ngay cả khi xảy ra sự cố.
  + Các lỗi phổ biến như **lỗi kết nối mạng, lỗi cảm biến, lỗi xử lý hình ảnh** đều có phương án xử lý dự phòng.
  + Khi phát hiện lỗi nghiêm trọng, phần mềm có thể **tự động khởi động lại tiến trình** mà không ảnh hưởng đến toàn bộ hệ thống.
* **Ghi log và giám sát hệ thống:**
  + Phần mềm liên tục **ghi lại nhật ký hoạt động (log file)** để dễ dàng phát hiện và khắc phục lỗi.
  + Hỗ trợ **cảnh báo lỗi** qua giao diện hoặc gửi thông báo đến giám sát viên để kịp thời xử lý.
* **Kiểm thử và tối ưu hóa:**
  + Trước khi triển khai, phần mềm được kiểm thử với các kịch bản khác nhau để đảm bảo khả năng **chịu tải và vận hành ổn định**.
  + Mã nguồn được tối ưu hóa để đảm bảo tốc độ xử lý nhanh, tiết kiệm tài nguyên hệ thống.

### ****2.2. Hiệu suất****

Phần mềm của hệ thống được thiết kế để đảm bảo **hiệu suất cao**, vận hành trơn tru và phản hồi nhanh trong mọi điều kiện hoạt động. Các yếu tố chính được tối ưu bao gồm:

#### **1. Tối ưu hóa mã nguồn (Code Optimization)**

* **Viết mã nguồn sạch, gọn gàng:** Tránh dư thừa, sử dụng cấu trúc dữ liệu và thuật toán phù hợp để tăng tốc độ xử lý.
* **Tận dụng đa luồng (Multithreading):**
  + Chia nhỏ các tác vụ như xử lý hình ảnh, giao tiếp cảm biến và ghi dữ liệu vào các luồng riêng biệt để tránh tắc nghẽn.
  + Giúp hệ thống phản hồi nhanh hơn và xử lý đồng thời nhiều tác vụ.
* **Giải phóng bộ nhớ:** Tránh rò rỉ bộ nhớ bằng cách giải phóng tài nguyên không cần thiết sau khi sử dụng.

#### **2. Tối ưu hóa quy trình xử lý dữ liệu**

* **Giảm độ trễ khi xử lý hình ảnh:**
  + Sử dụng các thư viện tối ưu như **OpenCV** hoặc **TensorFlow Lite** để tăng tốc nhận diện biển số.
  + Nếu cần, có thể áp dụng mô hình AI nhẹ hơn hoặc chạy trên GPU để giảm tải cho CPU.
* **Lưu trữ và truy xuất dữ liệu nhanh chóng:**
  + Sử dụng **cơ sở dữ liệu NoSQL hoặc SQLite** để tăng tốc truy vấn dữ liệu xe ra/vào.
  + Kết hợp bộ nhớ đệm (caching) để giảm truy xuất đĩa và tăng tốc độ xử lý.

#### **3. Tối ưu hóa thao tác sử dụng**

* **Thiết kế giao diện đơn giản, trực quan:** Giúp người dùng thao tác nhanh, hạn chế thao tác thừa.
* **Cải thiện tốc độ phản hồi:**
  + Các lệnh xử lý được tối ưu để đảm bảo thời gian phản hồi dưới **100ms**.
  + Tránh sử dụng các tác vụ đồng bộ (blocking) gây chậm hệ thống.
* **Tự động hóa quy trình:**
  + Hệ thống tự động xác định xe ra/vào mà không cần can thiệp thủ công.
  + Giảm thiểu số lần nhập liệu thủ công để tránh sai sót và tăng tốc độ xử lý.

### ****2.3. Khả năng mở rộng****

Hệ thống phần mềm được thiết kế với kiến trúc **mở** và **linh hoạt**, cho phép nâng cấp và mở rộng mà không cần xây dựng lại toàn bộ hệ thống từ đầu. Điều này giúp tiết kiệm thời gian, chi phí và công sức khi cần bổ sung tính năng mới hoặc mở rộng quy mô hệ thống.

#### **1. Kiến trúc phần mềm linh hoạt**

* Phần mềm được thiết kế theo mô hình **module hóa**, trong đó các thành phần chính như xử lý hình ảnh, quản lý cơ sở dữ liệu, giao diện người dùng và giao tiếp cảm biến hoạt động độc lập.
* Nhờ mô hình này, có thể **thay thế hoặc nâng cấp từng thành phần** mà không ảnh hưởng đến các phần khác.

#### **2. Hỗ trợ mở rộng phần cứng**

* Nếu cần nâng cấp **cảm biến, camera, bộ xử lý**, hệ thống vẫn có thể hoạt động với thiết bị mới bằng cách **cập nhật phần mềm** mà không cần thay đổi toàn bộ hệ thống.
* Ví dụ: Có thể nâng cấp từ Raspberry Pi 4 lên Raspberry Pi 5 mà chỉ cần tinh chỉnh phần mềm để tương thích với phần cứng mới.

#### **3. Hỗ trợ mở rộng quy mô hệ thống**

* Hệ thống có thể dễ dàng **tích hợp thêm camera, cảm biến** mà không cần thay đổi cấu trúc lõi của phần mềm.
* Hỗ trợ kết nối **nhiều thiết bị cùng lúc**, cho phép giám sát nhiều khu vực khác nhau mà không làm ảnh hưởng đến hiệu suất chung.
* Nếu cần quản lý **nhiều bãi đỗ xe**, chỉ cần bổ sung các node xử lý mới và đồng bộ dữ liệu qua mạng.

#### **4. Cập nhật phần mềm dễ dàng**

* Phần mềm hỗ trợ cập nhật từ xa (OTA - Over The Air), cho phép cập nhật tính năng mới mà không cần can thiệp trực tiếp vào hệ thống.
* Dữ liệu và cấu hình được lưu trữ theo chuẩn, đảm bảo tương thích với phiên bản mới mà không gây mất dữ liệu.

#### **5. Khả năng tích hợp với các hệ thống khác**

* Hệ thống có thể mở rộng để **tích hợp với hệ thống thanh toán tự động**, **hệ thống điều khiển bãi đỗ thông minh** hoặc **ứng dụng di động** mà không cần viết lại toàn bộ phần mềm.
* Hỗ trợ các giao thức tiêu chuẩn như **REST API, MQTT** giúp dễ dàng kết nối với các hệ thống bên ngoài.

### ****2.4. Bảo mật****

Hệ thống phần mềm được thiết kế với các cơ chế bảo mật nhằm **đảm bảo an toàn dữ liệu, ngăn chặn truy cập trái phép** và bảo vệ quyền lợi của người quản lý cũng như người sử dụng. Các cơ chế bảo mật chính bao gồm:

#### **1. Quản lý tài khoản và mật khẩu**

* **Hệ thống tài khoản riêng biệt:**
  + Mỗi người dùng (quản lý, nhân viên, khách hàng) có một tài khoản riêng để đăng nhập và sử dụng hệ thống.
* **Mật khẩu bảo mật:**
  + Mật khẩu được mã hóa bằng thuật toán **bcrypt hoặc SHA-256** trước khi lưu vào hệ thống, tránh bị lộ ngay cả khi dữ liệu bị đánh cắp.
  + Hỗ trợ **xác thực hai yếu tố (2FA)** để tăng cường bảo mật khi đăng nhập.
* **Cơ chế đặt lại mật khẩu an toàn:**
  + Nếu quên mật khẩu, người dùng cần xác minh danh tính qua **email hoặc mã OTP** trước khi được cấp lại mật khẩu mới.

#### **2. Xác minh người quản lý khi sử dụng hệ thống**

* **Phân quyền tài khoản:**
  + **Người quản lý** có quyền truy cập đầy đủ hệ thống, bao gồm cài đặt, giám sát và quản lý dữ liệu.
  + **Nhân viên** chỉ có quyền xem thông tin cần thiết và không thể thay đổi các thiết lập quan trọng.
* **Ghi lại lịch sử hoạt động (Audit Logs):**
  + Mọi thao tác như đăng nhập, sửa đổi dữ liệu, thêm/xóa tài khoản đều được ghi lại để dễ dàng kiểm tra khi cần thiết.
  + Nếu phát hiện hoạt động bất thường, hệ thống có thể gửi cảnh báo đến người quản lý.

#### **3. Bảo vệ dữ liệu và giao tiếp hệ thống**

* **Mã hóa dữ liệu:**
  + Dữ liệu quan trọng như thông tin tài khoản, biển số xe được **mã hóa trước khi lưu trữ** để tránh bị đánh cắp.
* **Giao tiếp an toàn:**
  + Sử dụng giao thức **HTTPS** để bảo vệ dữ liệu khi truyền qua mạng.
  + Nếu hệ thống có API, cần sử dụng **cơ chế xác thực API Key hoặc OAuth 2.0** để ngăn chặn truy cập trái phép.

#### **4. Phát hiện và ngăn chặn truy cập trái phép**

* **Giới hạn số lần đăng nhập sai:**
  + Nếu nhập sai mật khẩu nhiều lần, tài khoản sẽ bị tạm khóa để ngăn chặn tấn công dò mật khẩu (Brute-force Attack).
* **Cảnh báo đăng nhập bất thường:**
  + Nếu có đăng nhập từ vị trí hoặc thiết bị lạ, hệ thống sẽ yêu cầu xác minh bổ sung để đảm bảo tài khoản không bị đánh cắp.