

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**  
**KHOA CÔNG NGHỆ THÔNG TIN 1**

---



**BÁO CÁO BÀI TẬP LỚN**  
**ĐỀ TÀI: XÂY DỰNG HỆ THỐNG BÃI ĐỖ XE**  
**THÔNG MINH**

<b>Học phần:</b>	<b>IoT và ứng dụng</b>
<b>Giảng viên:</b>	<b>TS. Trần Tiến Công</b>
<b>Nhóm học phần:</b>	<b>02</b>
<b>Nhóm bài tập lớn:</b>	<b>03</b>
<b>Thực hiện:</b>	<b>1. Nguyễn Hữu Tuấn</b> <b>2. Bùi Hoàng Vinh</b> <b>3. Phạm Thị Phương Anh</b>

**HÀ NỘI, 2023**

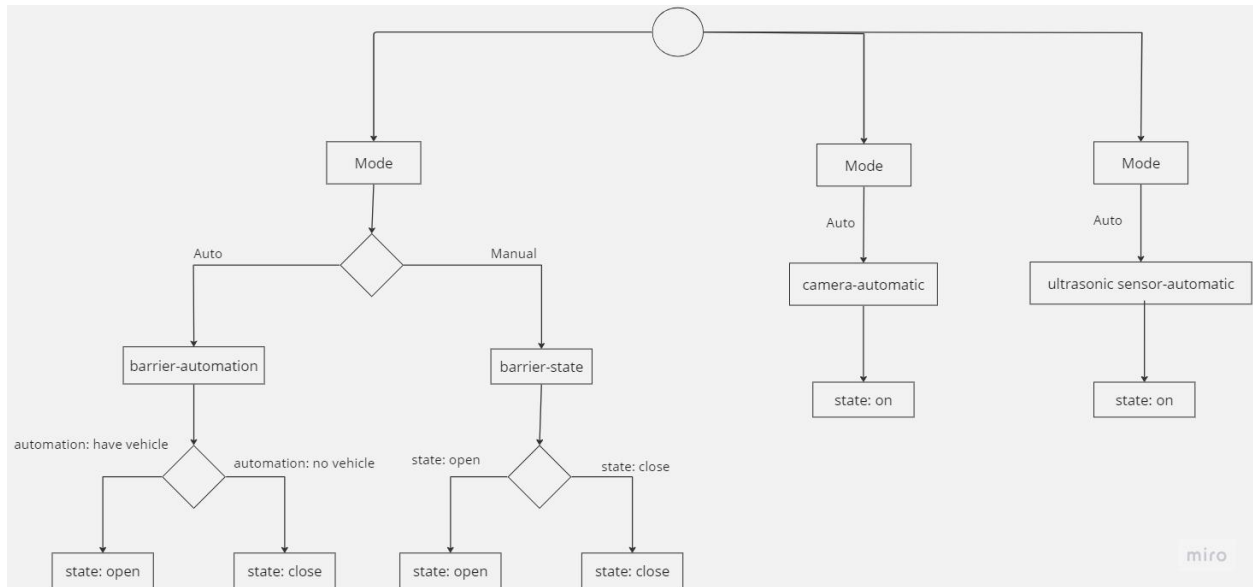
## MỤC LỤC

<b>MỤC LỤC .....</b>	<b>1</b>
<b>I. MÔ TẢ HỆ THỐNG .....</b>	<b>2</b>
<b>II. ĐẶC TẢ TIỀN TRÌNH.....</b>	<b>3</b>
<b>III. ĐẶC TẢ MÔ HÌNH MIỀN .....</b>	<b>3</b>
<b>IV. ĐẶC TẢ MÔ HÌNH THÔNG TIN .....</b>	<b>4</b>
<b>V. THÔNG SỐ DỊCH VỤ .....</b>	<b>4</b>
<b>VI. ĐẶC TẢ CẤP ĐỘ IOT .....</b>	<b>6</b>
<b>VII. ĐẶC TẢ THÀNH PHẦN CHỨC NĂNG.....</b>	<b>7</b>
<b>VIII. ĐẶC TẢ THÀNH PHẦN HOẠT ĐỘNG .....</b>	<b>7</b>
<b>IX. TÍCH HỢP THIẾT BỊ .....</b>	<b>8</b>
<b>X. PHÁT TRIỂN ỨNG DỤNG.....</b>	<b>9</b>

## I. MÔ TẢ HỆ THỐNG

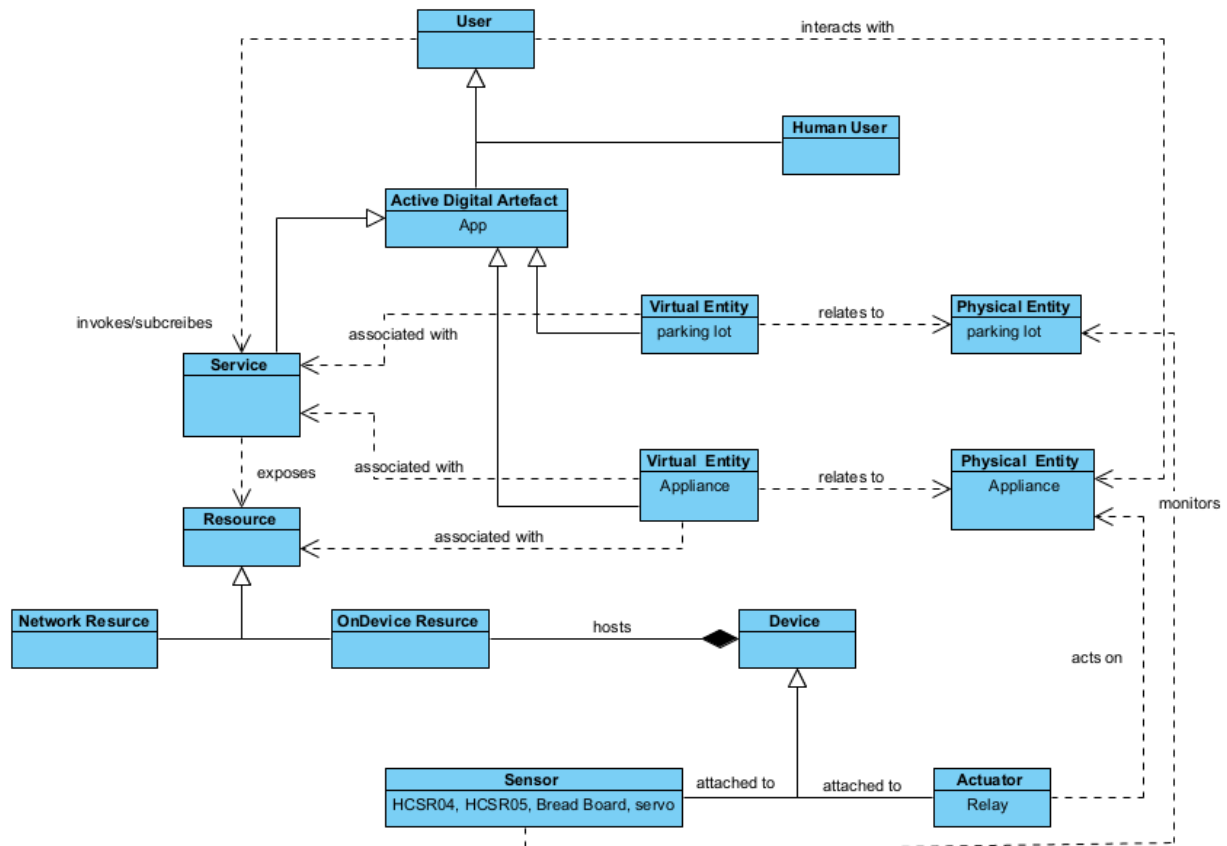
- Mục tiêu: tạo ra một hệ thống bãi đỗ xe thông minh, đáp ứng được các tiêu chí sau:
  - + Thực hiện phân loại phương tiện giao thông khi có phương tiện vào bãi đỗ xe
  - + Hiển thị vị trí đỗ xe phù hợp với loại phương tiện đi vào
  - + Kiểm tra phương tiện đỗ sai vị trí và cảnh báo
  - + Cập nhật số lượng slot đỗ xe còn trống theo từng loại xe
- Hành vi: Hệ thống tự động hóa cho bãi đỗ xe nên có chế độ tự động và thủ công.
  - + Ở chế độ tự động, hệ thống tự động phát hiện xe ra vào bãi đỗ để mở cổng, thực hiện phân loại xe, đọc biển số xe và đưa ra vị trí đỗ xe phù hợp; đồng thời phát cảnh báo khi có xe đỗ sai chỗ.
  - + Ở chế độ thủ công, hệ thống cung cấp tùy chọn mở cổng bằng tay. Người quản lý bãi đỗ xe có thể sử dụng giọng nói để thực hiện mở cổng
- Đặc tả chức năng hệ thống:
  - + Tính năng xác thực người dùng:
    - Hệ thống có giao diện đăng nhập cho phép người dùng đăng nhập để sử dụng các tính năng hệ thống
    - Hệ thống tạo các vai trò khác nhau cho từng người dùng sử dụng
  - + Tính năng phân loại phương tiện giao thông:
    - Sử dụng camera để phát hiện phương tiện
    - Sau khi phát hiện phương tiện, hệ thống sẽ đưa ra loại được phát hiện
  - + Tính năng tự động đóng mở cổng cho phương tiện
  - + Hiển thị vị trí đỗ xe phù hợp với loại xe: Sau khi thực hiện phân loại, hệ thống sẽ đưa ra vị trí đỗ phù hợp với từng loại xe
  - + Nhận diện biển số xe và lưu lại quản lý:
    - Camera ghi lại hình ảnh chứa biển số của phương tiện
    - Thực hiện nhận diện biển số của phương tiện
    - Lưu lại thông tin biển số xe, ngày/giờ vào để quản lý
  - + Kiểm tra xe đỗ sai và cảnh báo
    - Tại mỗi vị trí đỗ có lắp đặt 1 camera để ghi lại hình ảnh xe đỗ
    - Nếu phát hiện phương tiện đỗ tại vị trí không đúng sẽ đưa ra cảnh báo
  - + Cập nhật số lượng slot đỗ xe còn trống hiện tại theo từng loại xe: Tại mỗi vị trí đỗ có lắp đặt 1 cảm biến siêu âm, nếu phát hiện có xe đỗ sẽ cập nhật số lượng vị trí đỗ đã có phương tiện
  - + Tính khoảng thời gian đỗ xe
- Tính năng thông minh: phát hiện và phân loại xe đi vào bãi đỗ, phát hiện biển số xe của các phương tiện

## II. ĐẶC TẢ TIẾN TRÌNH



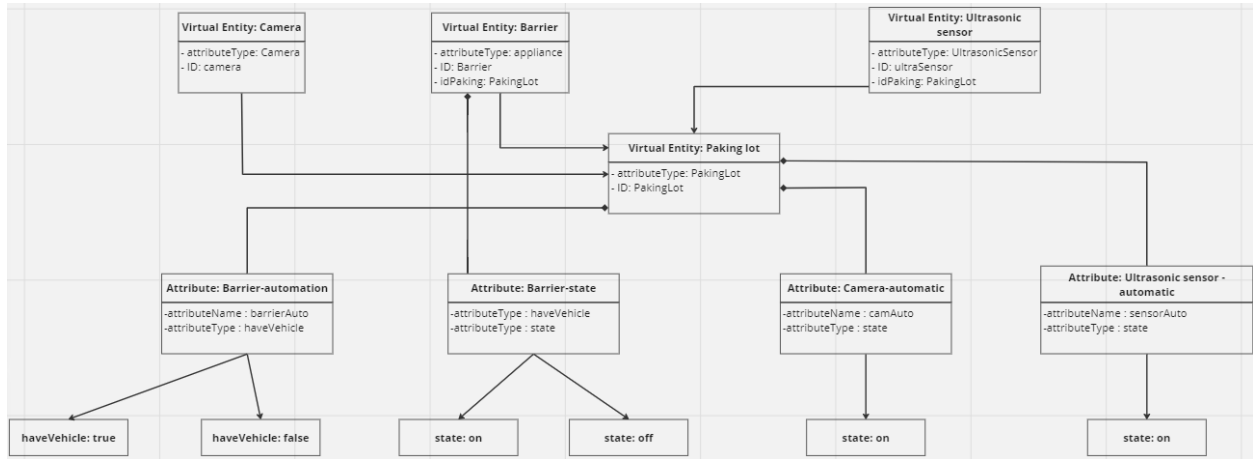
Hình 1. Sơ đồ đặc tả tiến trình

## III. ĐẶC TẢ MÔ HÌNH MIỀN



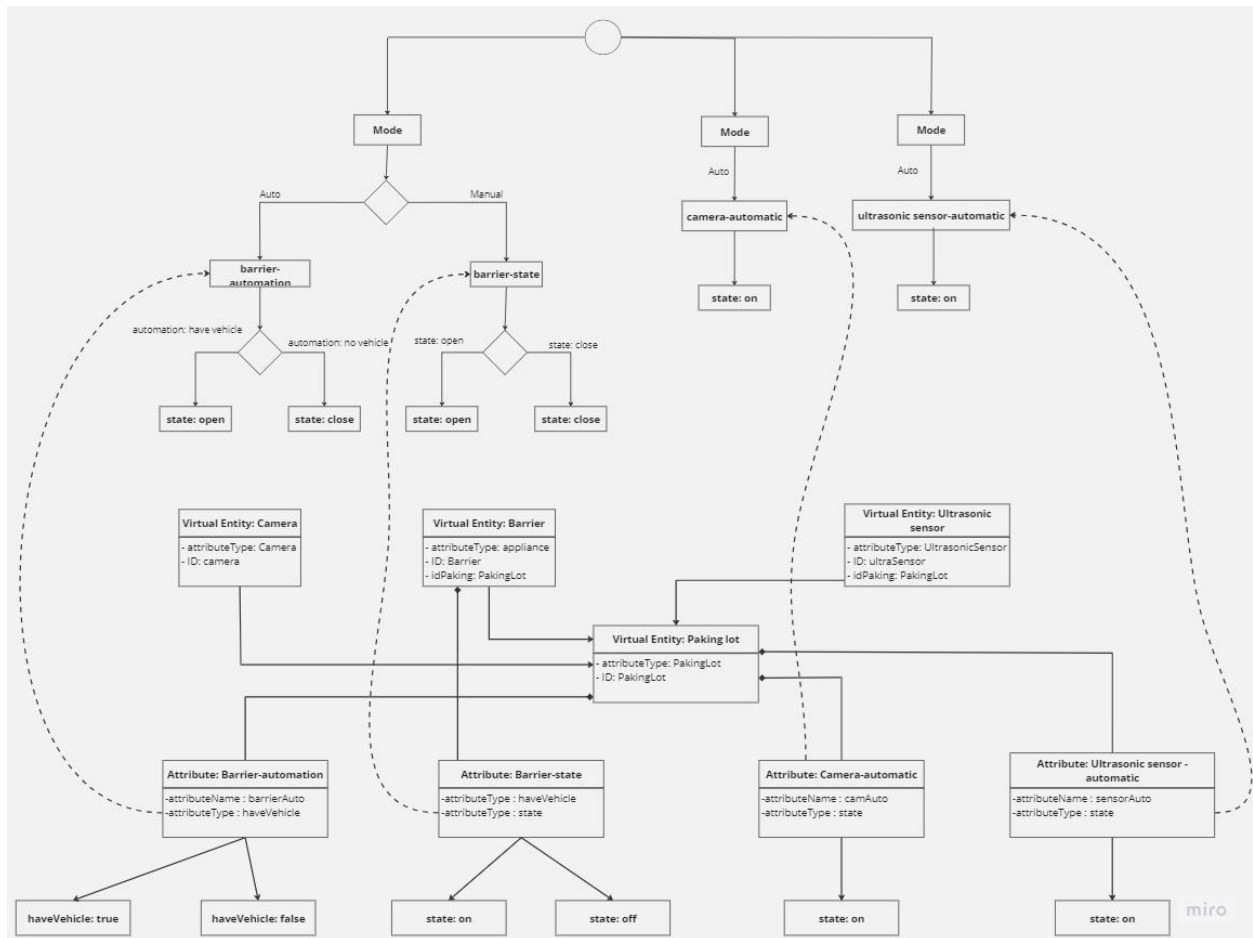
Hình 2. Sơ đồ đặc tả mô hình miền

## IV. ĐẶC TẢ MÔ HÌNH THÔNG TIN

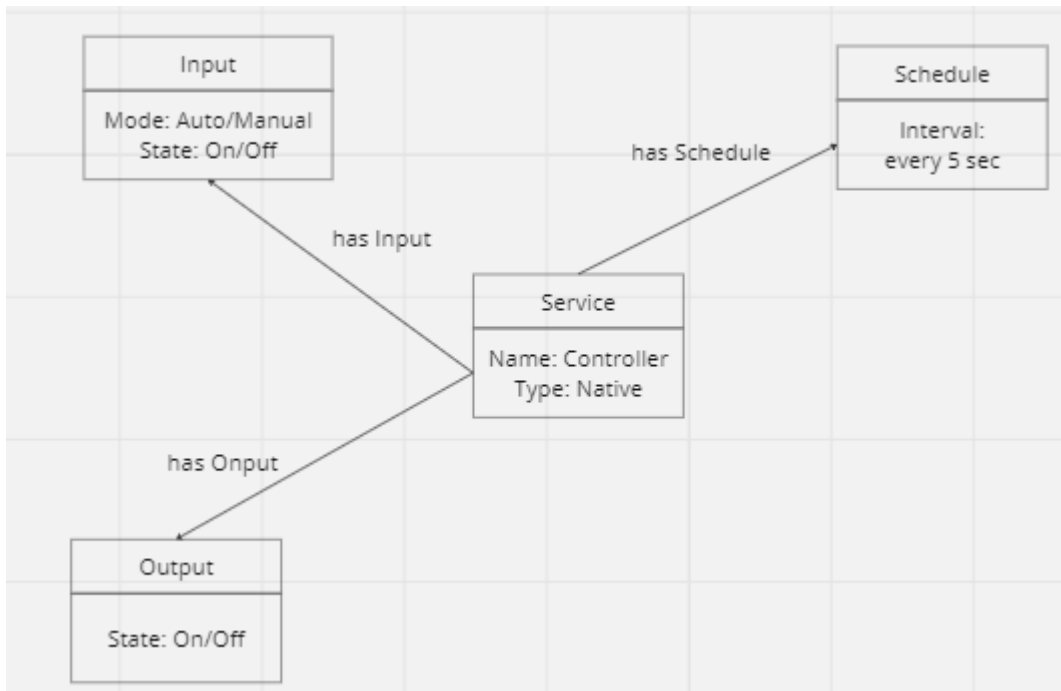


Hình 3. Sơ đồ đặc tả mô hình thông tin

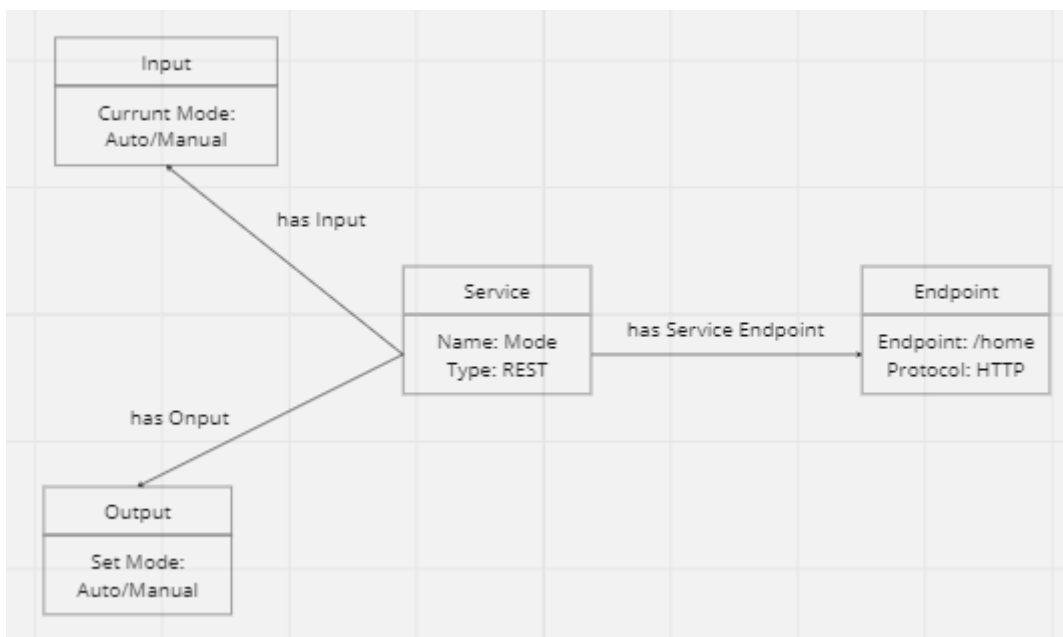
## V. THÔNG SỐ DỊCH VỤ



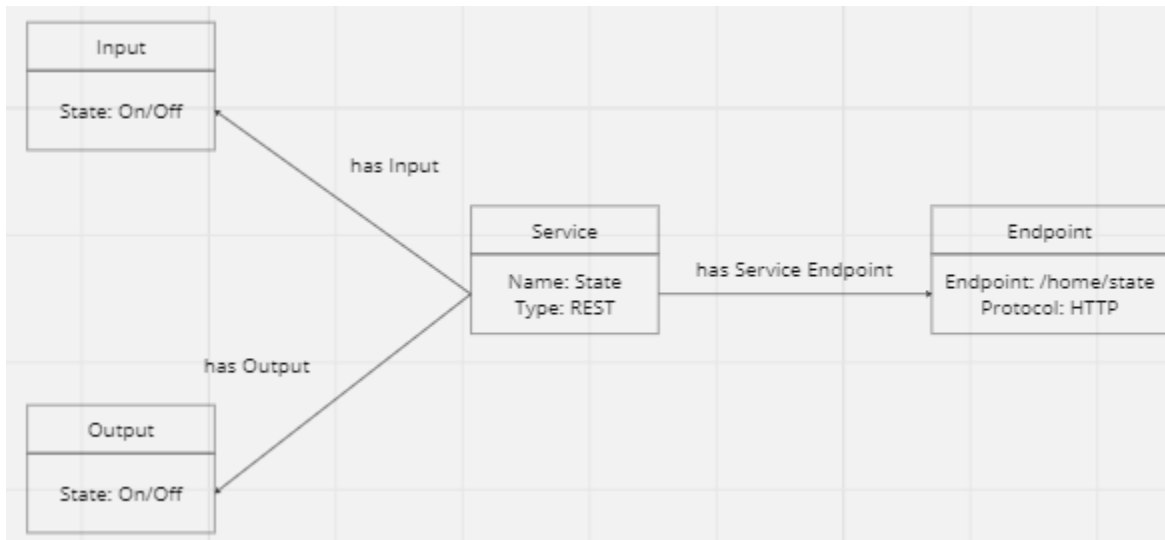
Hình 4. Sơ đồ thông số dịch vụ



**Hình 5.** Sơ đồ thông số dịch vụ - 2

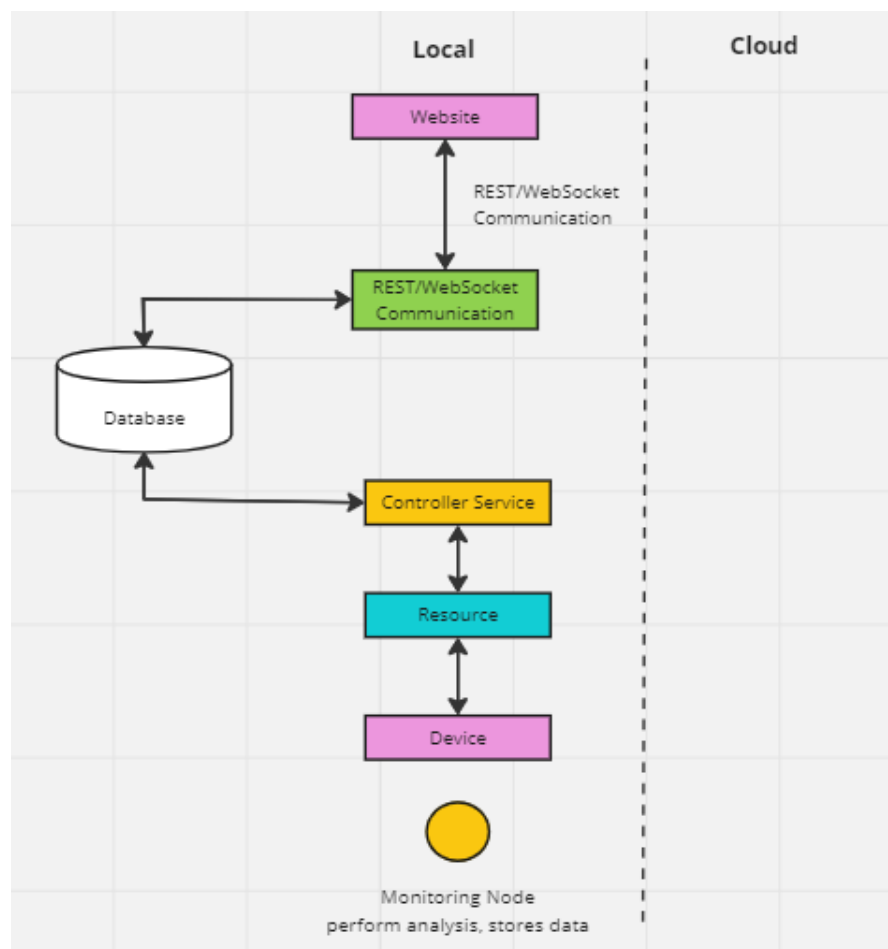


**Hình 6.** Sơ đồ thông số dịch vụ - 3



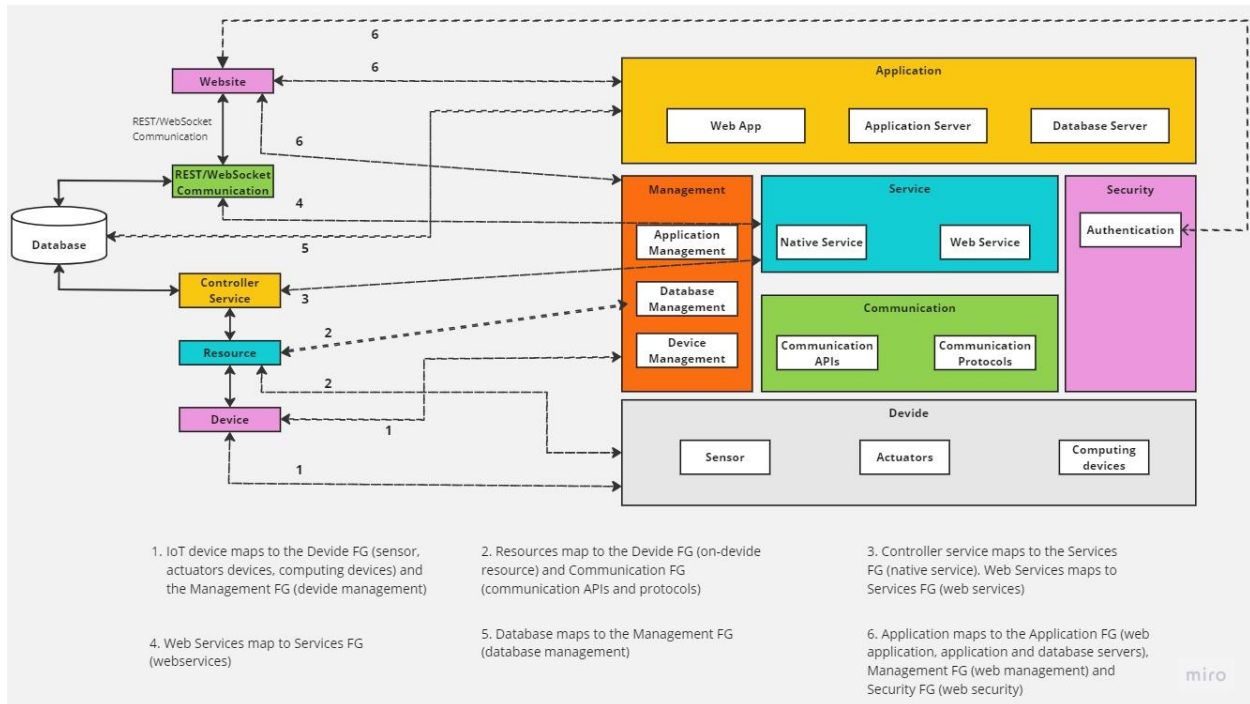
*Hình 7. Sơ đồ thông số dịch vụ - 4*

## VI. ĐẶC TẢ CẤP ĐỘ IOT



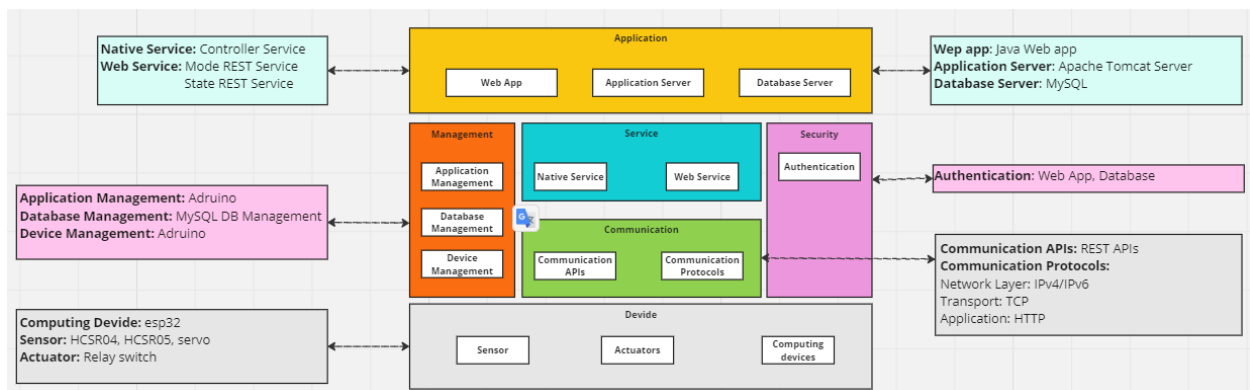
*Hình 8. Sơ đồ đặc tả cấp độ IoT – Cấp 3*

## VII. ĐẶC TẢ THÀNH PHẦN CHỨC NĂNG



Hình 9. Sơ đồ đặc tả thành phần chức năng

## VIII. ĐẶC TẢ THÀNH PHẦN HOẠT ĐỘNG

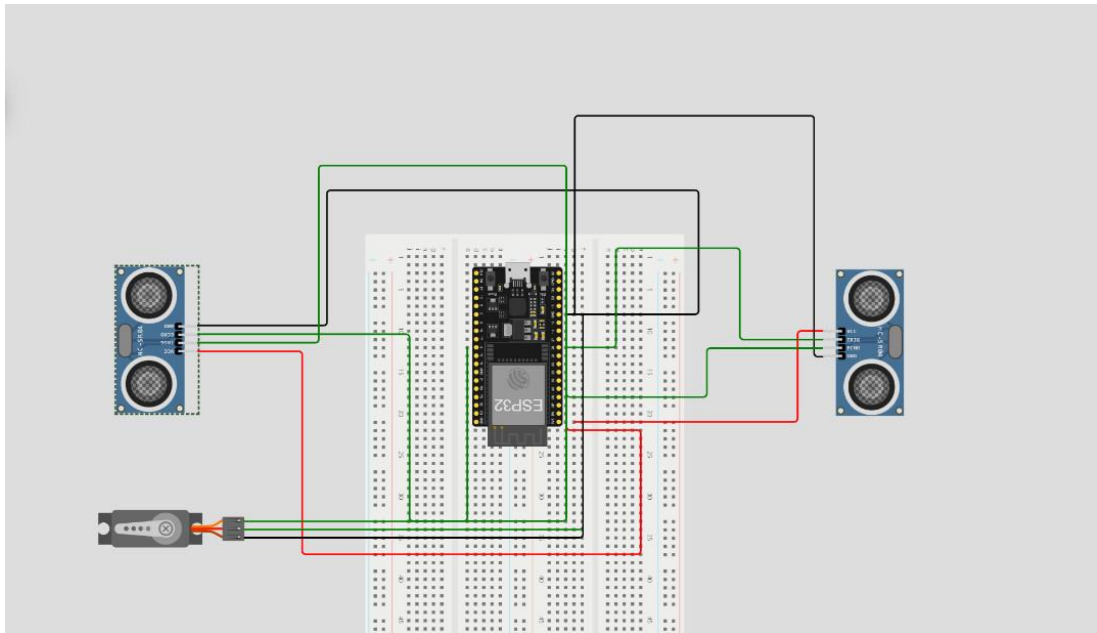


Hình 10. Sơ đồ đặc tả thành phần hoạt động



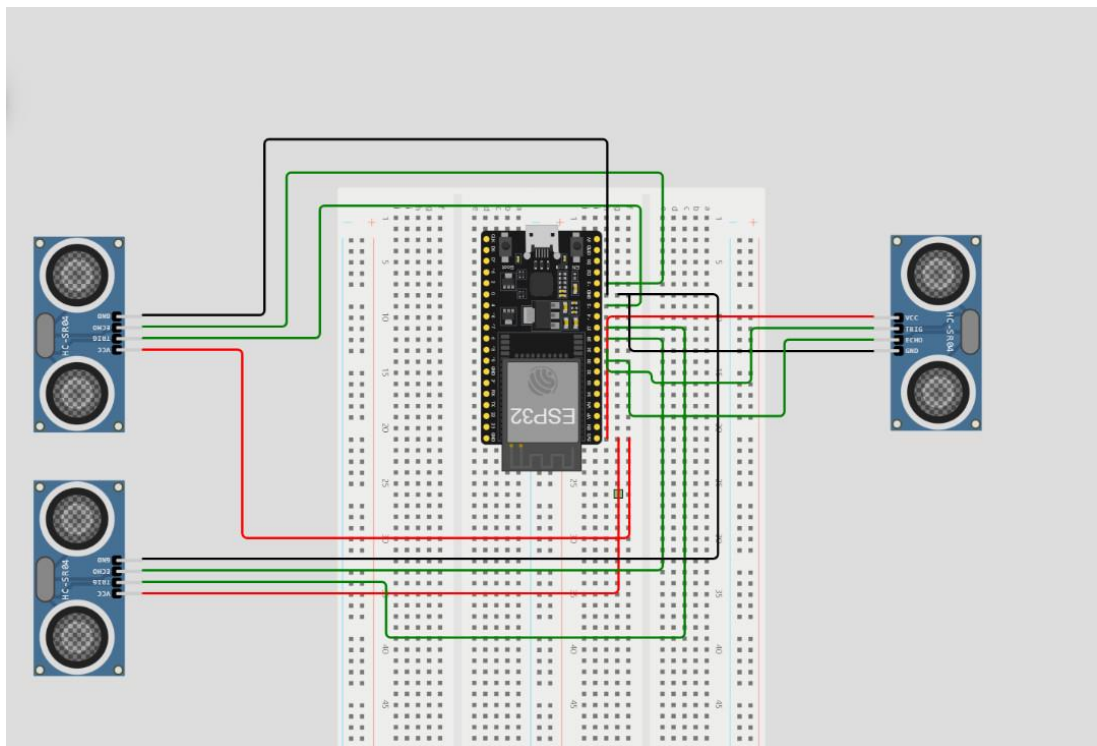
## IX. TÍCH HỢP THIẾT BỊ

- Tích hợp thiết bị phát hiện xe đi vào đóng mở cửa tự động



*Hình 11. Mô hình thiết bị cho phát hiện xe đi vào và đóng mở cửa tự động*

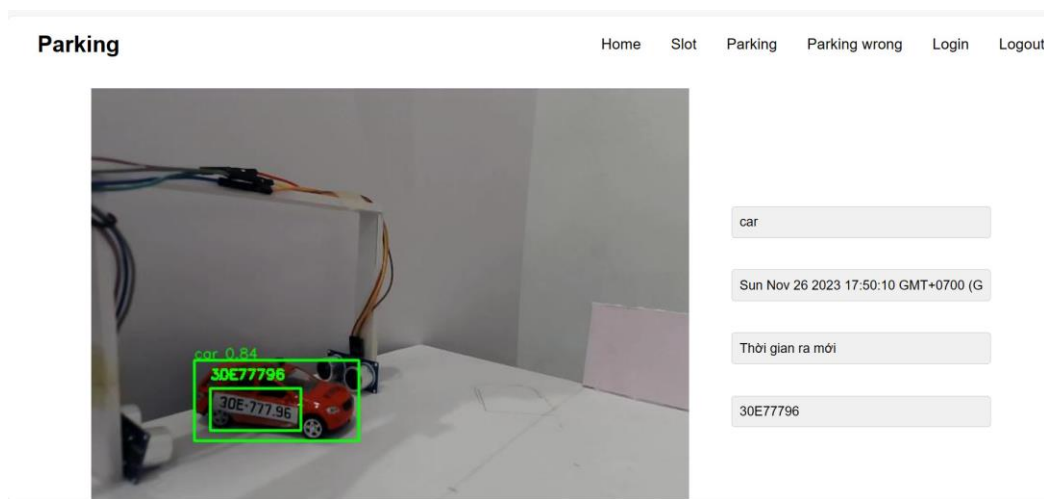
- Tích hợp thiết bị cảm biến tại các vị trí đỗ xe



*Hình 12. Mô hình thiết bị cho cảm biến tại các vị trí đỗ xe*

## X. PHÁT TRIỂN ỨNG DỤNG

- Khi có xe đi qua cổng, các cảm biến siêu âm sẽ phát hiện và gửi tín hiệu yêu cầu đến server thông minh nhận diện phương tiện và đọc biển số xe từ hình ảnh do camera thu được
- Sau khi thu được ảnh, server thông minh sẽ gửi hình ảnh thu nhận được và thông tin về loại xe, biển số xe cho phía Client hiện lên website cho người quản lý bãi đỗ xe theo dõi.
- Khi nhận được hình ảnh và thông tin về loại xe, biển số xe, hệ thống sẽ gửi tín hiệu cho servo mở cổng để xe tiến vào bãi đỗ, đồng thời backend của Client sẽ lưu lại thông tin của phương tiện, thời gian vào của phương tiện vào cơ sở dữ liệu



Hình 13. Giao diện phát hiện xe vào bãi đỗ, phân loại và đọc biển số xe

```
void loop()
{
    client.loop();
    if (!client.connected()) {
        connect_to_broker();
    }
    int distance1 = getDistance(echo1, trig1);
    int distance2 = getDistance(echo2, trig2);

    // Nhận diện xe đã đến khi 2 khoảng cách nhỏ hơn 8cm
    if(distance1 < 8 && distance2 < 8 && isCalculated == 0){
        client.publish("BTL_IOT_PTIT", "1;XE_DA_DEN");
        isCalculated = 1;
    } else if (distance1 > 8 && distance2 > 8 && isCalculated == 1){
        closeServo();
        client.publish("BTL_IOT_PTIT", "5;DA_DONG_CONG");
        isCalculated = 0;
    }
    /* In kết quả ra Serial Monitor */
    Serial.print("distance1 = ");
    Serial.print(distance1);
    Serial.print(" ; distance2 = ");
    Serial.println(distance2);
    delay(2000);
}
```

Hình 14. Hàm loop cho cảm biến siêu âm phát hiện xe vào bãi đỗ

```

void callback(char* topic, byte *payload, unsigned int length) {
    Serial.println("-----new message from broker-----");
    Serial.print("topic: ");
    Serial.println(topic);
    Serial.print("message: ");
    Serial.write(payload, length);
    Serial.println();
    // khi nhận được tín hiệu '2' sẽ mở cổng servo
    // khi nhận được tín hiệu '0' sẽ đóng cổng servo
    if (*payload == '2'){
        openServo();
        client.publish("BTL_IOT_PTIT", "4;DA_MO_CONG");
    }
    else if (*payload == '0'){
        closeServo();
    }
}

void openServo(){
    for(int i = 0; i <= 90 ; i += 10){
        myServo.write(i);
        delay(150);
    }
}

void closeServo(){
    for(int i = 90; i >= 0 ; i -= 10){
        myServo.write(i);
        delay(150);
    }
}

```

**Hình 15.** Hàm callback xử lý nhận message từ mqtt

```

11 from detect import detect
12
13 broker_address = "broker.hivemq.com"
14 port = 1883
15
16 # Tạo client
17 client = mqtt.Client("huutuan1705")
18 # Kết nối đến broker
19 client.connect(broker_address, port)
20 # Publish message
21 topic_publish = "BTL_IOT_PTIT"
22
23 def send_message(message_publish):
24     client.publish(topic_publish, message_publish)
25     print(f"Published message '{message_publish}' to topic '{topic_publish}'")
26 send_message("HELLO FROM PYTHON")
27
28 def send_image(image):
29     client.publish(topic_publish, image)
30     print(f"Send image success")
31

```

**Hình 16.** Thực hiện kết nối tới mqtt tại server thông minh và các hàm gửi message, gửi hình ảnh cho Client

```

32 # Callback khi message được publish thành công
33 def on_publish(client, userdata, mid):
34     print(f"Message published successfully (mid={mid})")
35
36 client.on_publish = on_publish
37 topic_subscribe = "BTL_IOT_PTIT"
38
39 xeDaDen = False
40 def on_message(client, userdata, msg):
41     message = msg.payload.decode()
42     print(f"Received message '{msg.payload.decode()}' on topic '{msg.topic}'")
43     print(str(message))
44     if message == '1;XE_DA_DEN':
45         global xeDaDen
46         xeDaDen = True
47
48 client.subscribe(topic_subscribe)
49 client.on_message = on_message
50
51 cam_ip = '172.20.10.2'
52
53 # URL của API luồng video trên ESP32-CAM
54 api_url = f'http://{cam_ip}/capture'
55
56 vid = cv2.VideoCapture(1)

```

**Hình 17.** Hàm Callback và nhận thông điệp từ cảm biến siêu âm

```

57 while(True):
58     ret, frame = vid.read()
59
60     if xeDaDen:
61         frame, vehicel_detect, plate_detect = detect(frame)
62         if vehicel_detect != '' and plate_detect != '':
63             send_message("2;" + vehicel_detect + ";" + plate_detect)
64             send_message("...NHAN_DIEN_XE")
65
66             xeDaDen = False
67
68             # _, buffer = cv2.imencode('.jpg', frame)
69             image = Image.fromarray(cv2.cvtColor(frame, cv2.COLOR_BGR2RGB))
70             img_buffer = BytesIO()
71             image.save(img_buffer, format="JPEG")
72             jpg_code = base64.b64encode(img_buffer.getvalue()).decode('utf-8')
73             send_image(jpg_code)
74             # decode_img(jpg_code)
75
76             cv2.imshow('frame', frame)
77             if cv2.waitKey(1) & 0xFF == ord('q'):
78                 break
79             client.loop_start()
80
81 client.loop_stop()
82 vid.release()
83 cv2.destroyAllWindows()
84 client.disconnect()

```

**Hình 18.** Mở camera, nếu nhận được thông điệp từ cảm biến siêu âm tại cổng sẽ thực hiện phân loại xe và đọc biển số

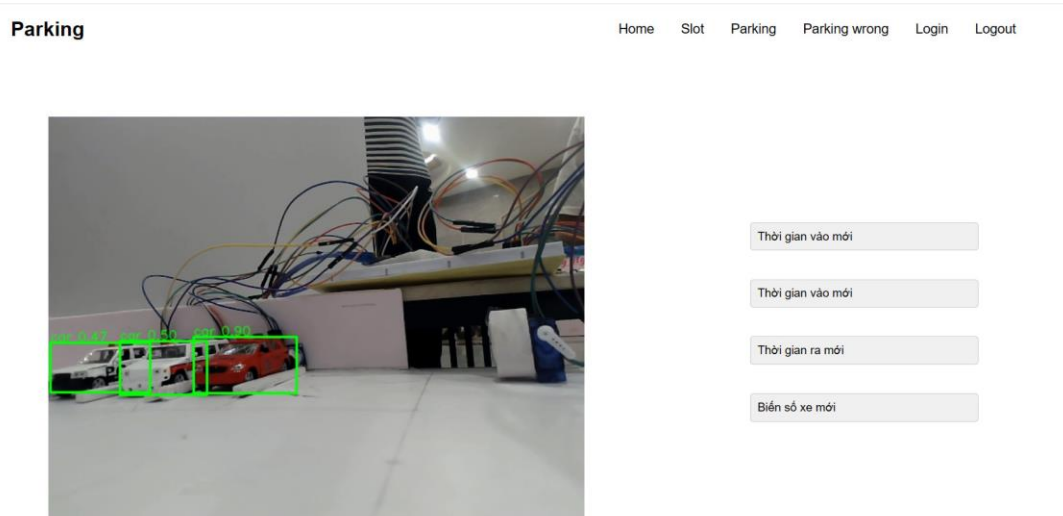
```

35 def detect(frame):
36     results_vehicle = predict_vehicle(model_vehicle, frame)
37     results_plate = predict_plate(model_plate, frame)
38
39     if results_vehicle is not None :
40         results_vehicle = non_max_suppression(results_vehicle, 0.45)
41         frame, vehicel_detect, plate_detect = draw_box(frame, results_vehicle, results_plate)
42
43     return frame, vehicel_detect, plate_detect

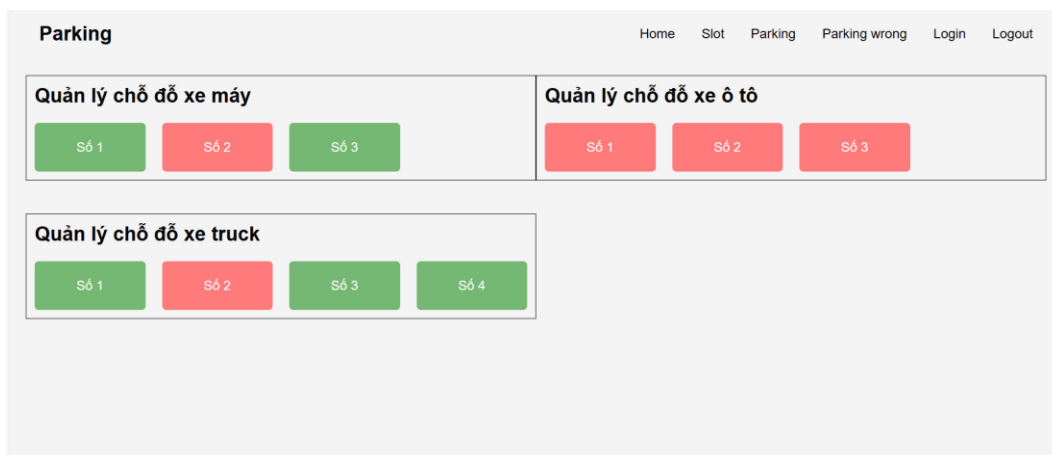
```

**Hình 19.** Hàm phân loại và đọc biển số của phương tiện đi vào bãi đỗ

- Khi có xe đi vào vị trí đỗ, các cảm biến siêu âm được lắp đặt trước sẽ phát hiện được có xe đi vào và cập nhật vị trí đỗ xe còn trống.
- Đồng thời, cảm biến cũng gửi tín hiệu đến server thông minh nhận diện phương tiện tại khu vực đỗ xe, nếu có phương tiện đỗ sai khu vực sẽ hiện thông báo lên website cho quản lý bãi đỗ xe có thể phát hiện.



**Hình 20.** Giao diện phát hiện xe vào vị trí đỗ.



**Hình 20.** Giao diện cập nhật số lượng vị trí còn trống trong bãi đỗ tại mỗi khu vực (màu xanh thể hiện vị trí còn trống, màu đỏ thể hiện vị trí đã có xe)

```

void loop()
{
    client.loop();
    if (!client.connected()) {
        connect_to_broker();
    }
    int distance1 = getDistance(echo1, trig1);
    int distance2 = getDistance(echo2, trig2);
    int distance3 = getDistance(echo3, trig3);

    handleParkingState(distance1, distance2, distance3);

    /* In kết quả ra Serial Monitor */
    Serial.print("distance1 = ");
    Serial.print(distance1);
    Serial.print(" ; distance2 = ");
    Serial.print(distance2);
    Serial.print(" ; distance3 = ");
    Serial.println(distance3);
    delay(2000);
}

```

**Hình 21.** Hàm loop cho cảm biến siêu âm phát hiện xe vào vị trí đỗ

```

void handleParkingState(int distance1, int distance2, int distance3){
    char newState[200] = "";
    strcpy(newState, parking_state);
    if(distance1 <= 10){
        newState[1] = 'Y';
    }
    else{
        newState[1] = 'N';
    }

    if(distance2 <= 10){
        newState[2] = 'Y';
    }
    else{
        newState[2] = 'N';
    }

    if(distance3 <= 10){
        newState[3] = 'Y';
    }
    else{
        newState[3] = 'N';
    }
    Serial.print("newState = ");
    Serial.println(newState);
    if(strcmp(newState, parking_state)){
        Serial.print("SENT NEW STATE");
        strcpy(parking_state, newState);
        char sendStateMsg[200] = "6;";
        strcat(sendStateMsg, newState);
        client.publish("BTL_IOT_PTIT", "5;XE_VAO_CHO");
        client.publish("BTL_IOT_PTIT", sendStateMsg);
    }
}

```

**Hình 22.** Hàm xử lý trạng thái đỗ xe



```

62 vid = cv2.VideoCapture(1)
63 while(True):
64     ret, frame = vid.read()
65
66     if xeDaDen:
67         frame = detect_slot_car(frame)
68         send_message("")
69         xeDaDen = False
70
71         _, buffer = cv2.imencode('.jpg', frame)
72         jpg_code = base64.b64encode(buffer)
73         send_image(jpg_code)
74
75     cv2.imshow('frame', frame)
76     if cv2.waitKey(1) & 0xFF == ord('q'):
77         break
78     client.loop_start()
79
80 client.loop_start()
81 vid.release()
82 cv2.destroyAllWindows()
83 client.disconnect()

```

**Hình 23.** Mở camera, nếu nhận được thông điệp từ cảm biến siêu âm tại vị trí đỗ sẽ thực hiện phát hiện các xe tại khu vực đỗ xe

```

45 def detect_slot_car(frame):
46     results_vehicle = predict_vehicle(model_vehicle, frame)
47     if results_vehicle is not None :
48         results_vehicle = non_max_suppression(results_vehicle, 0.45)
49         frame = draw_box_slot_car(frame, results_vehicle)
50     return frame

```

**Hình 24.** Hàm thực hiện phát hiện phương tiện tại khu vực ô tô

```

38 def draw_box_slot_car(image, boxes_vehicle):
39     wrong_vehicle = []
40     for i, (left, top, right, bottom, score, label) in enumerate(boxes_vehicle):
41         c1, c2 = (left, top), (right, bottom)
42
43         if label == "car":
44             cv2.rectangle(image, c1, c2, color = (0,255,0), thickness= 2) # vẽ hcn
45             cv2.putText(image, "{}".format(label) + " {:.2f}".format(score), (c1[0], c1[1] - 2), \
46                 0, fontScale= 0.5, color = (0,255,0), thickness= 1, lineType= cv2.LINE_AA)
47         else:
48             cv2.rectangle(image, c1, c2, color = (0,0,255), thickness= 2) # vẽ hcn
49             cv2.putText(image, "{}".format(label) + " {:.2f}".format(score), (c1[0], c1[1] - 2), \
50                 0, fontScale= 0.5, color = (0,0,255), thickness= 1, lineType= cv2.LINE_AA)
51         wrong_vehicle.append([left, top, right, bottom])
52
53     return image

```

**Hình 25.** Hàm thực hiện phát hiện phương tiện tại khu vực ô tô, nếu phương tiện không phải ô tô sẽ bị khoanh đỏ

- Khi xe ra khỏi bãi đỗ, các cảm biến siêu âm sẽ phát hiện và gửi tín hiệu yêu cầu server thông minh nhận diện phương tiện và đọc biển số xe từ hình ảnh do camera thu được.
- Sau khi thu được ảnh, server thông minh sẽ gửi hình ảnh thu nhận được và thông tin về loại xe, biển số xe cho phía Client, backend của Client sẽ cập nhật thời gian ra của phương tiện và tính toán khoảng thời gian đỗ xe, từ đó giúp người quản lý tính tiền gửi của phương tiện.