

**BỘ GIAO THÔNG VẬN TẢI**  
**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI TP HỒ CHÍ MINH**  
**VIỆN ĐÀO TẠO CHẤT LƯỢNG CAO**



**NỘI DUNG THUYẾT TRÌNH**  
**HỌC PHẦN: HỆ THỐNG GIAO THÔNG THÔNG MINH**  
**TÊN ĐỀ TÀI: BÃI ĐẬU XE THÔNG MINH**  
**GVHD: Thầy VŨ ĐÌNH LONG**  
**Mã học phần: HT22CLC**

*Thực hiện:*

Họ và tên	Mã số sinh viên	Mã Học Phần
Lê Văn Khởi(NT)	054205002859	HT22CLC
Nguyễn Đức Trí	068205006293	HT22CLC
Phạm Tiến Đạt	22h1120037	HT22CLC
Đoàn Minh Đăng	091205011073	HT22CLC

## LỜI CẢM ƠN

Lời cảm ơn đầu tiên, chúng em xin cảm ơn chân thành thầy Vũ Đình Long đã trực tiếp giảng dạy chúng em học phần “Chuyên đề Hệ thống giao thông thông minh”. Trong quá trình tìm hiểu và làm bài tiểu luận, Thầy đã tận tình chỉ dẫn, giải đáp mọi thắc mắc của chúng em. Chính vì những kinh nghiệm và kiến thức vô cùng bổ ích của thầy mà chúng em tích lũy được nhiều kiến thức và có cái nhìn sâu sắc về môn học này.

Cuối cùng, xin cảm ơn đến với các bạn thành viên trong nhóm, các bạn đã nỗ lực hết sức mình đóng góp ý kiến cùng nhau để hoàn thành tốt bài thuyết trình. Các bạn đều hoàn thành rất tốt những nhiệm vụ được giao. Vì kiến thức và kinh nghiệm thực tế còn nhiều hạn chế nên bài thuyết trình chúng em không thể tránh khỏi những thiếu sót và những chỗ chưa chính xác. Chúng em rất mong nhận được sự nhận xét và ý kiến đóng góp từ thầy để giúp chúng em nhận ra những điểm còn chưa làm tốt và có cơ hội sửa đổi cho những bài tiểu luận sau.

Chúng em xin chân thành cảm ơn!

ĐÁNH GIÁ KẾT QUẢ HOẠT ĐỘNG NHÓM

Họ và tên	Đánh giá mức độ hoàn thiện	Mức độ nhất trí của nhóm
Lê Văn Khởi(NT)	5	10/10
Nguyễn Đức Trí	5	10/10
Phạm Tiến Đạt	5	10/10
Đoàn Minh Đăng	5	10/10

# MỤC LỤC

<b>CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI</b>	1
1.1. Lý do chọn đề tài	1
1.2. Mục tiêu thực hiện	1
1.3. Phạm vi thực hiện & công cụ sử dụng	1
<b>CHƯƠNG 2: MÔ PHỎNG HỆ THỐNG BÃI GIỮ XE THÔNG MINH BẰNG ARDUINO</b>	2
2.1. Sơ đồ hệ thống tổng quát Hệ thống mô phỏng bao gồm hai phần chính:	2
2.2. Quy trình hoạt động khi xe vào	2
2.3. Quy trình hoạt động khi xe ra	4
2.4. Phân khu vực bãi xe (khu A – xe máy, khu B – ô tô)	6
2.5. Mô phỏng phần cứng	7
<b>CHƯƠNG 3: PHÁT TRIỂN GIAO DIỆN PYTHO</b>	8
3.1. Môi trường và thư viện sử dụng	8
3.2. Nhận diện biển số xe qua camera	9
3.3. Tạo và gán thẻ ID cho xe	10
3.4. Quản lý thông tin xe và tính phí	10
3.5. Giao diện người dùng bằng Tkinter	11
<b>CHƯƠNG 4: KẾT NỐI ARDUINO VỚI PHẦN MỀM VÀ CƠ SỞ DỮ LIỆU</b>	11
4.1. Giao tiếp giữa Arduino và Python	11
4.2. Đồng bộ trạng thái mô phỏng và phần mềm	12
4.3. Cơ sở dữ liệu và xử lý dữ liệu	13
<b>CHƯƠNG 5: ĐÁNH GIÁ HỆ THỐNG</b>	14
5.1. Ưu điểm	14
5.2. Hạn chế	14
<b>CHƯƠNG 6: HƯỚNG PHÁT TRIỂN HỆ THỐNG</b>	15
<b>CHƯƠNG 7: KẾT LUẬN</b>	15
<b>CHƯƠNG 8: TÀI LIỆU THAM KHẢO</b>	15

## CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI

### 1.1. Lý do chọn đề tài

Trong thời đại công nghệ phát triển mạnh mẽ như hiện nay, việc tự động hoá các hệ thống quản lý đang trở nên phổ biến, đặc biệt trong lĩnh vực giao thông và đô thị thông minh. Một trong những vấn đề rất thực tế mà hầu hết các thành phố đều gặp phải, đó là tình trạng thiếu chỗ đỗ xe và việc quản lý xe ra vào còn mang tính thủ công, gây tốn nhân lực và dễ xảy ra sai sót.

Vậy nên, bọn mình chọn mô phỏng một hệ thống bãi đỗ xe thông minh – không cần người điều hành trực tiếp, có khả năng nhận diện biển số xe, tự động phát thẻ ID, phân làn xe máy và ô tô theo khu vực A/B, đồng thời hỗ trợ thanh toán phí giữ xe bằng phần mềm.

Hệ thống này không chỉ giúp tiết kiệm thời gian, công sức mà còn là cơ hội để bọn mình áp dụng những gì đã học về lập trình nhúng (Arduino), thị giác máy tính (OpenCV + Tesseract), cũng như giao diện phần mềm với Python và cơ sở dữ liệu MySQL.

### 1.2. Mục tiêu thực hiện

- Mô phỏng hệ thống bãi đỗ xe thông minh, tự động nhận diện xe ra/vào.
- Sử dụng Arduino và cảm biến để điều khiển barrier và phát hiện xe.
- Giao diện bằng Python hiển thị thông tin, nhận diện biển số xe bằng OCR.
- Tự động phân loại xe máy và ô tô vào khu A và B dựa trên biển số.
- Gán thẻ ID cho từng xe, quản lý thông tin xe hiện tại và lưu lịch sử ra vào.
- Tạo môi trường mô phỏng sát với thực tế, có thể triển khai nâng cấp về sau.

### 1.3. Phạm vi thực hiện & công cụ sử dụng

Hệ thống được chia 2 thành phần chính:

Phần cứng: Sử dụng Arduino UNO kết nối với cảm biến siêu âm, servo motor để mô phỏng barrier, LCD hiển thị và các đèn tín hiệu.

Phần mềm:

Ngôn ngữ: Python

Thư viện sử dụng: OpenCV (nhận diện biên số), Tesseract OCR (chuyển hình ảnh sang text), Tkinter (tạo giao diện người dùng), serial (giao tiếp Arduino)

Cơ sở dữ liệu: MySQL

Trong các chương tiếp theo, tui mình sẽ trình bày chi tiết từng thành phần, hoạt động, giao diện và mã nguồn cụ thể cho từng chức năng.

## **CHƯƠNG 2: MÔ PHỎNG HỆ THỐNG BÃI GIỮ XE THÔNG MINH BẰNG ARDUINO**

### **2.1. Sơ đồ hệ thống tổng quát**

Hệ thống mô phỏng bao gồm hai phần chính:

Phần phần cứng: Arduino, LCD, nút nhấn, servo motor (mô phỏng barrier), đèn LED, thẻ ID cố định.

Phần phần mềm: Python sử dụng OpenCV + Tesseract OCR để nhận dạng biên số, quản lý thẻ ID, tính phí và lưu trữ dữ liệu.

Cấu trúc hệ thống:

Người dùng bấm nút yêu cầu vào bãi.

Hệ thống nhả một thẻ ID cố định (ví dụ ID001, ID002...), đồng thời chụp ảnh biển số xe bằng camera.

Python xử lý ảnh, trích xuất biển số bằng Tesseract OCR.

Biển số được gán với thẻ ID, lưu vào database.

Từ biển số, hệ thống xác định là xe máy hay ô tô → phân làn vào khu A hoặc khu B.

Khi xe ra, người dùng trả lại thẻ ID.

Hệ thống kiểm tra ID trong cơ sở dữ liệu → nếu khớp biển số đã lưu → hiển thị thông tin và tính phí.

Sau khi thanh toán, barrier mở và dữ liệu được chuyển qua bảng lịch sử.

### **2.2. Quy trình hoạt động khi xe vào**

Quá trình hoạt động cụ thể như sau:

1. Người dùng nhấn nút để yêu cầu vào bãi.

2. Arduino gửi tín hiệu đến Python thông qua Serial → Python hiểu đây là lệnh “XE\_VAO”.

3. Python xử lý:

Sinh mã ID từ danh sách thẻ còn trống (ví dụ lấy ID003).

Kích hoạt camera, chụp ảnh biển số xe.

Dùng OpenCV để xử lý ảnh, Tesseract OCR để trích xuất biển số.

Gán biển số và mã ID → xác định xe là xe máy hay ô tô dựa trên mã (H1, B1...) để phân khu A/B.

Lưu vào bảng current\_vehicles.

4. Sau khi lưu thành công → gửi lệnh mở barrier (qua Serial) để Arduino mở servo → barrier nâng lên cho xe vào.

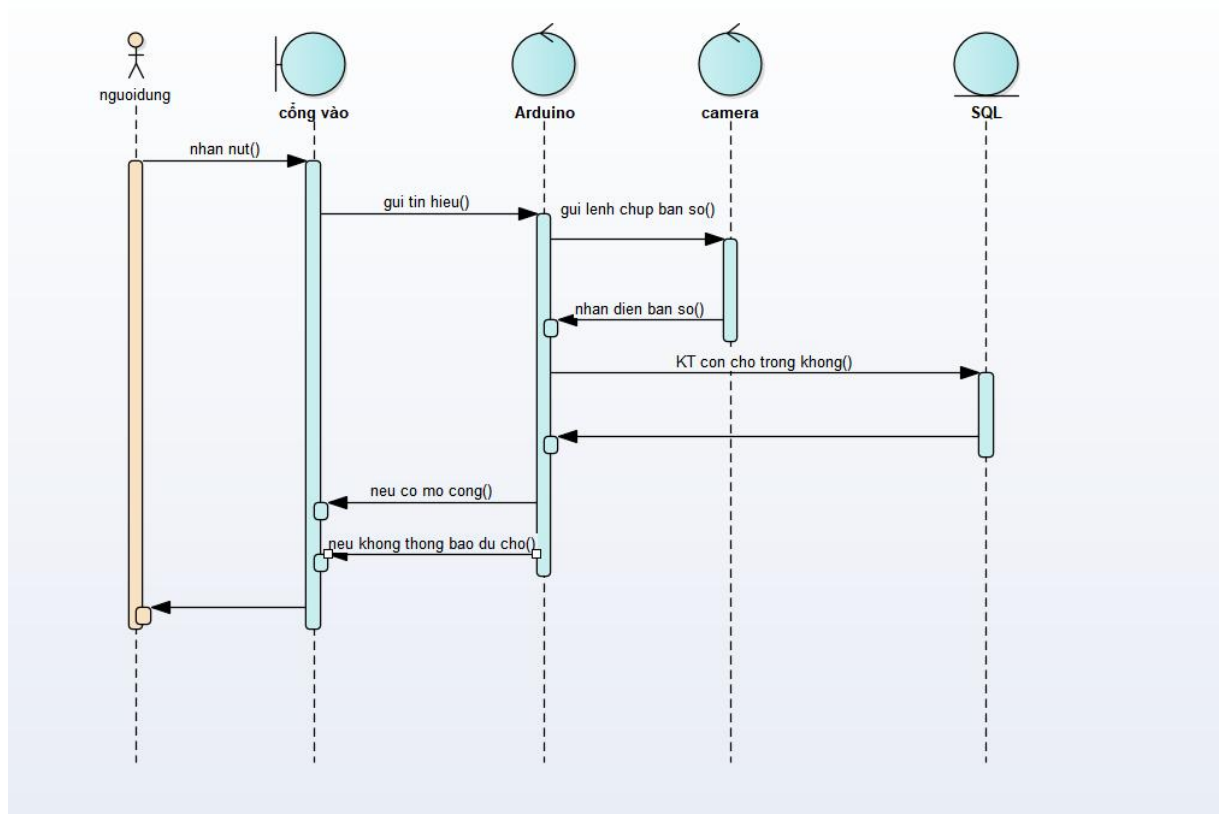
5 LCD trên Arduino hiển thị thông điệp “Chào mừng” hoặc ID xe được phát.

```
# Xử lý xe vào bãi
def process_parking(tree, image_canvas):
    image_path = select_image()
    if image_path:
        # Hiển thị ảnh trong khung nhỏ
        img = Image.open(image_path)
        img = img.resize((200, 150), Image.Resampling.LANCZOS)
        img_tk = ImageTk.PhotoImage(img)
        image_canvas.image = img_tk
        image_canvas.create_image(0, 0, anchor=tk.NW, image=img_tk)

        # Nhận diện biển số
        license_plate_text = recognize_plate(image_path)
        if license_plate_text and license_plate_text != "UNKNOWN":
            vehicle_type = check_vehicle_type(license_plate_text)
            park_vehicle(tree, license_plate_text, vehicle_type)
        else:
            messagebox.showwarning("Không nhận diện được", "Không thể nhận diện biển số từ ảnh này.")
    else:
        messagebox.showwarning("Chưa chọn ảnh", "Bạn chưa chọn ảnh biển số.")

# Xử lý xe ra bãi
```

Sơ đồ tuần tự (sequence diagram):



### 2.3. Quy trình hoạt động khi xe ra

1. Người dùng trả lại thẻ ID.
2. Python nhận lệnh “XE\_RA” từ Arduino hoặc nút nhấn.
3. Kích hoạt camera chụp biển số → OCR để trích xuất biển số xe.
4. Kiểm tra trong bảng current\_vehicles xem biển số và ID có khớp nhau không.
5. Nếu hợp lệ:



- Tính phí gửi xe dựa trên thời gian vào/ra.
- Hiện thị thông tin và xác nhận thanh toán.
- Gửi lệnh mở barrier cho Arduino → barrier mở bằng servo.
- Xóa thông tin khỏi bảng `current_vehicles` → chuyển vào bảng `vehicle_history`.

6. Nếu không khớp: hệ thống thông báo lỗi “Sai thẻ hoặc sai biển số”.

```
# Xử lý xe ra bãi
def process_exit(tree):
    selected_item = tree.selection()
    if not selected_item:
        messagebox.showwarning("Chưa chọn xe", "Vui lòng chọn xe muốn ra.")
        return

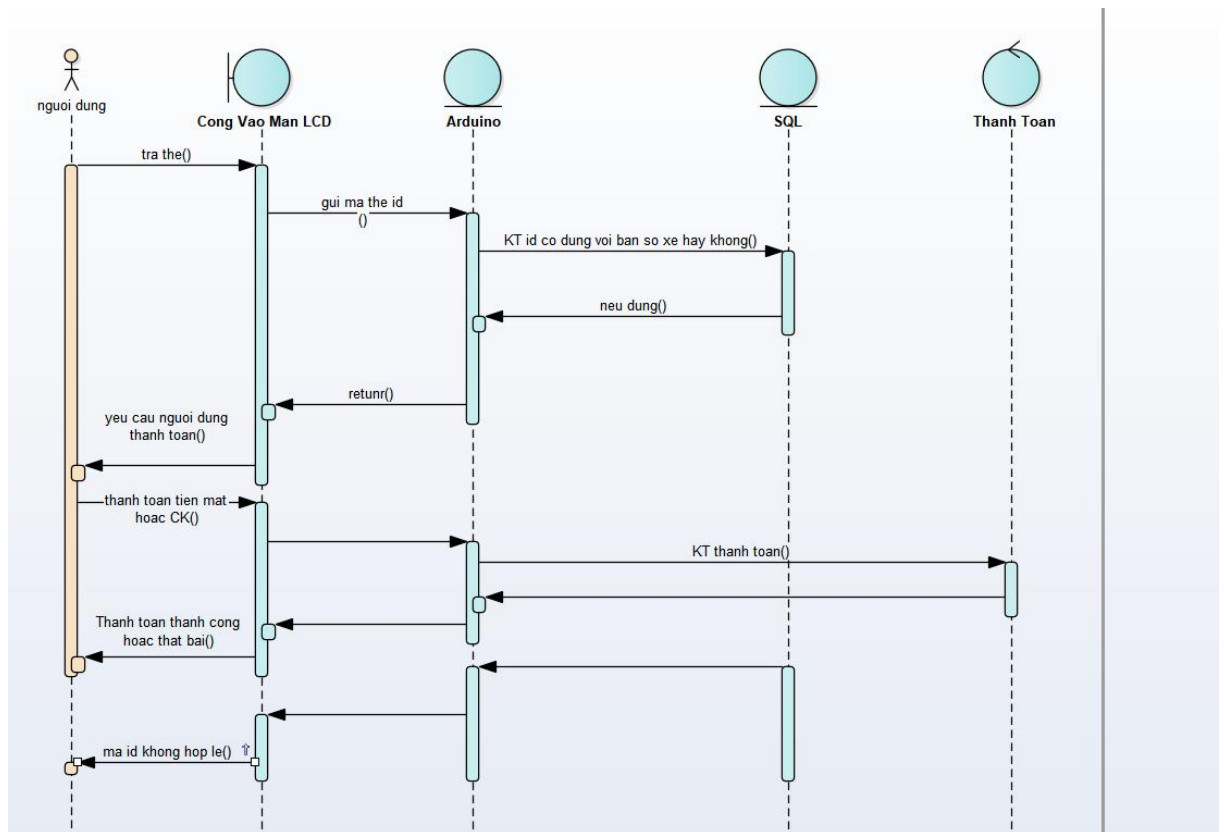
    item = tree.item(selected_item)
    license_plate = item["values"][0]
    time_in_str = item["values"][1]
    vehicle_type = item["values"][4]

    # Tính phí và cập nhật thời gian ra
    time_in = time.mktime(time.strptime(time_in_str, "%Y-%m-%d %H:%M:%S"))
    fee = calculate_parking_fee(vehicle_type, time_in)
    time_out = time.strftime("%Y-%m-%d %H:%M:%S", time.localtime())

    # Cập nhật bảng và xóa khỏi danh sách
    tree.item(selected_item, values=(license_plate, time_in_str, time_out, f"{fee} VND", vehicle_type, item["values"][5]))
    if vehicle_type == "Xe máy":
        motorbike_spots[:] = [spot for spot in motorbike_spots if spot[0] != license_plate]
    else:
        car_spots[:] = [spot for spot in car_spots if spot[0] != license_plate]

    messagebox.showinfo("Thanh toán", f"Phí đậu xe cho {license_plate} là {fee} VND.")
```

Sơ đồ tuần tự (diagram):



## 2.4. Phân khu vực bãi xe (khu A – xe máy, khu B – ô tô)

Hệ thống xác định khu vực đỗ dựa vào mã phân loại trên biển số xe.

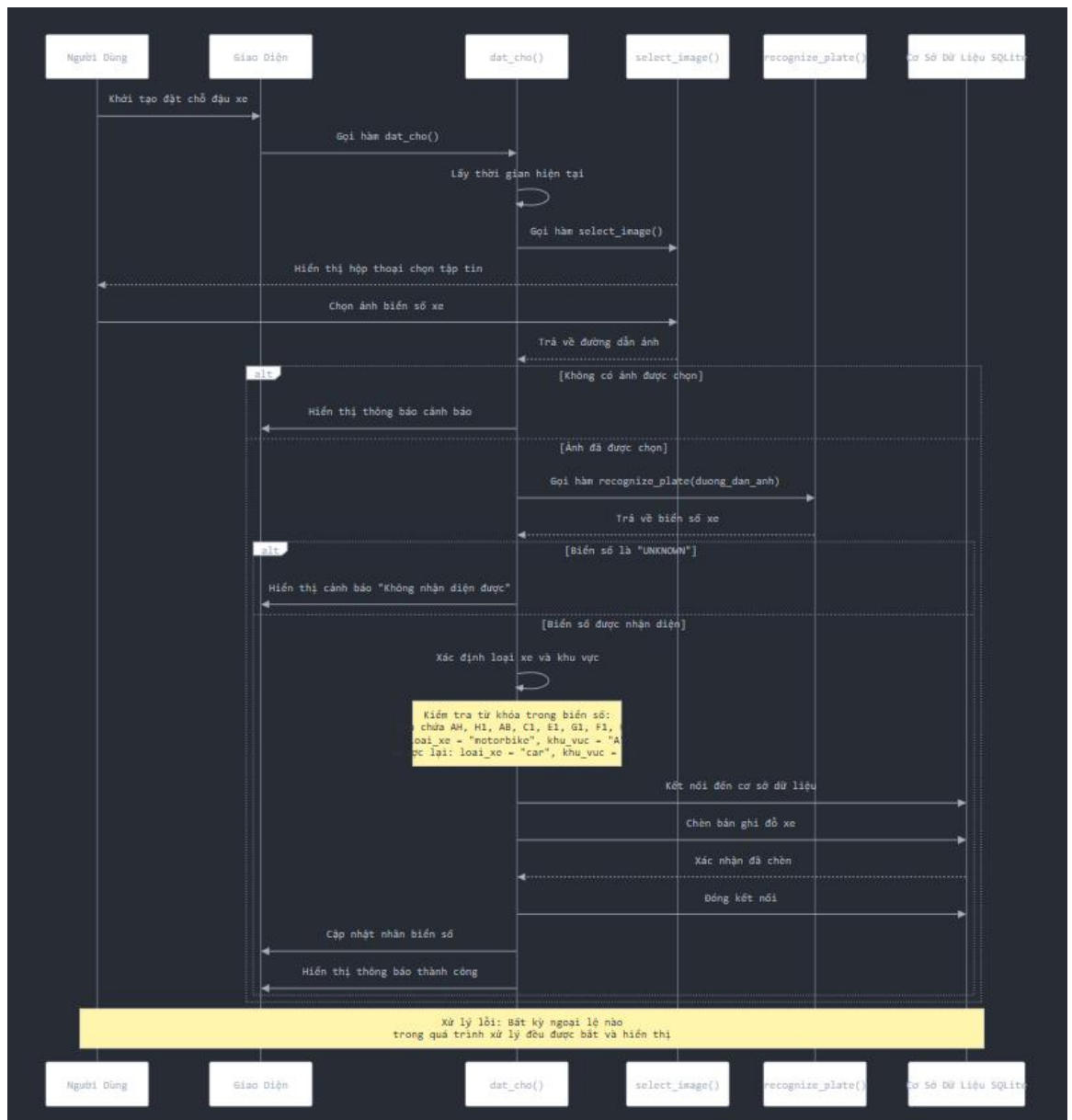
- Nếu biển số có dạng “59-H1”, “66-B1”, “65-C1”, v.v... → được xác định là xe máy → đưa vào khu A.
- Nếu biển số không thuộc nhóm đó → mặc định là ô tô → đưa vào khu B.

Phân khu sẽ được lưu kèm trong cơ sở dữ liệu để quản lý theo từng loại phương tiện.

```

# Phân biệt loại xe dựa trên biển số
def check_vehicle_type(license_plate):
    motorbike_keywords = ["AH", "H1", "AB", "C1", "E1", "G1", "F1", "K1"]
    for keyword in motorbike_keywords:
        if keyword in license_plate:
            return 'motorbike'
    return 'car'
  
```

Sơ đồ sequence:



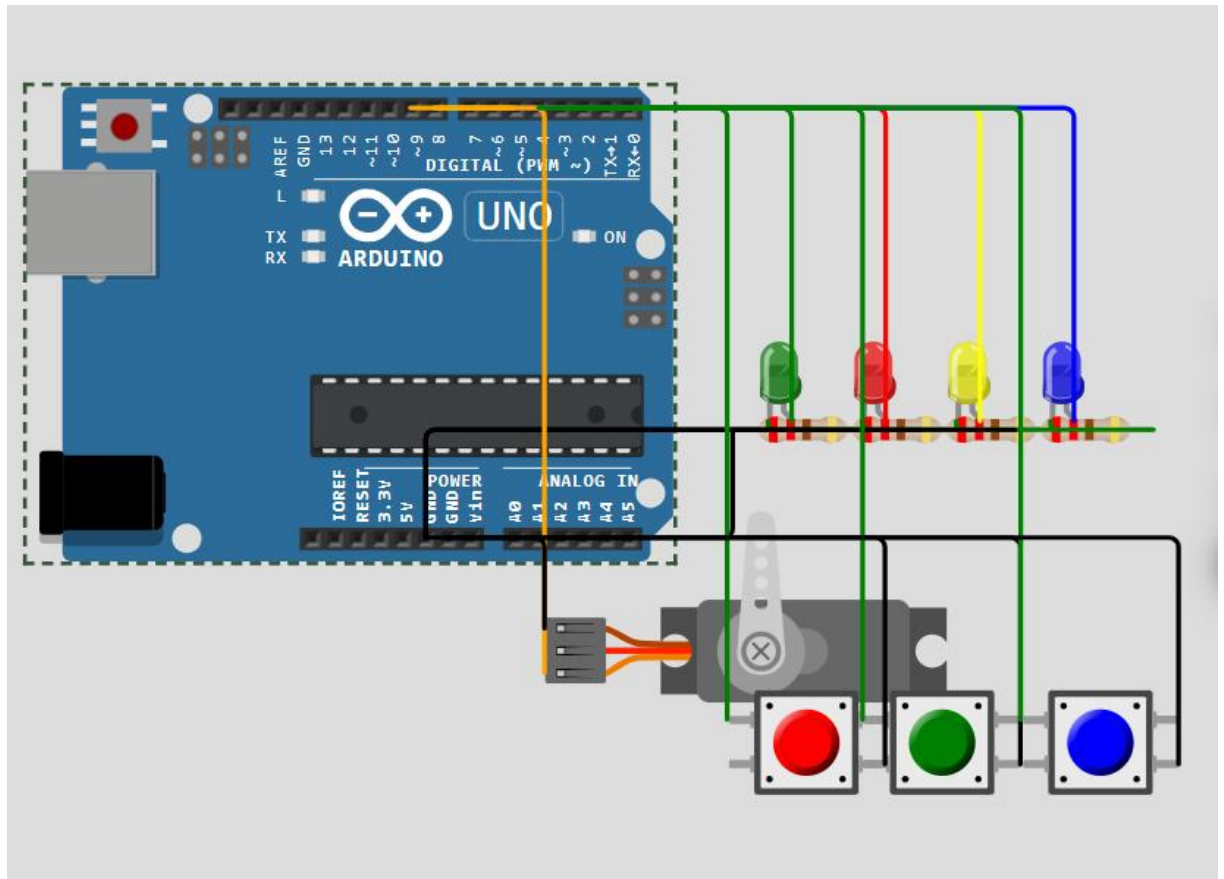
## 2.5. Mô phỏng phần cứng

Hệ thống phần cứng được mô phỏng bằng:

- Arduino Uno làm trung tâm điều khiển.
- LCD 16x2 hiển thị thông báo (còn chỗ, ID cấp phát, lỗi...).
- Servo motor mô phỏng barrier nâng lên/hạ xuống.
- Đèn LED: báo hiệu trạng thái bãi còn chỗ hoặc đầy.
- Nút nhấn:
  - ◦ Nút 1: xin vào bãi (gửi lệnh “XE\_VAO”).
  - ◦ Nút 2: xin ra bãi (gửi lệnh “XE\_RA”).
- Thẻ ID là các giá trị cố định được lưu sẵn trong hệ thống (ví dụ: ID001 → ID050).

Hiện tại hệ thống không sử dụng cảm biến siêu âm để phát hiện xe thật sự — điều này sẽ được đưa vào phần “Hướng phát triển” như một tính năng mở rộng để tăng độ tự động hoá sau này.

Sơ đồ tổng quát:



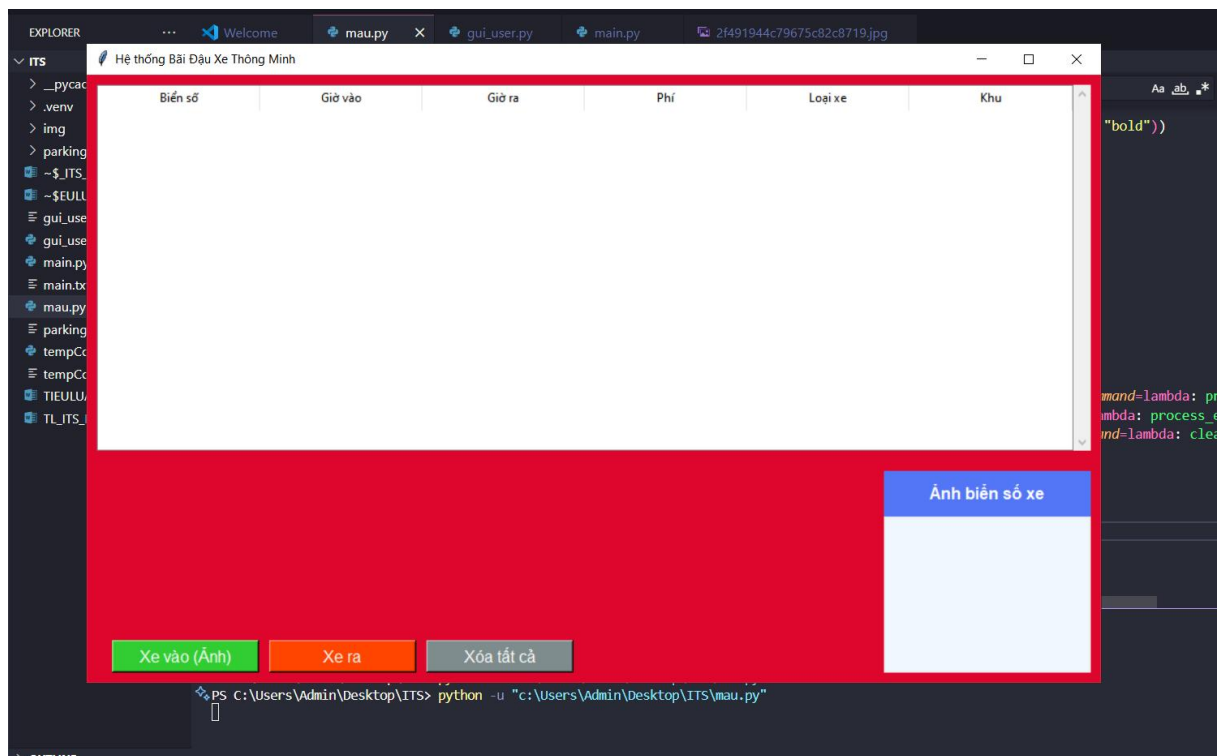
## CHƯƠNG 3: PHÁT TRIỂN GIAO DIỆN PYTHON

### 3.1. Môi trường và thư viện sử dụng

Giao diện phần mềm của hệ thống được xây dựng bằng ngôn ngữ lập trình Python, do Python có khả năng xử lý ảnh tốt, thư viện đa dạng và dễ kết nối với cơ sở dữ liệu. Các thư viện sử dụng gồm:

- Tkinter: tạo giao diện người dùng.
- OpenCV: xử lý hình ảnh, cắt biển số xe từ khung hình camera.
- pytesseract (Tesseract OCR): trích xuất văn bản (biển số xe) từ hình ảnh.
- mysql.connector: kết nối và tương tác với cơ sở dữ liệu MySQL.
- Serial (pyserial): giao tiếp với Arduino qua cổng COM.

Phần mềm này giúp mô phỏng toàn bộ quy trình vận hành hệ thống bãi xe một cách trực quan và có thể mở rộng về sau.



Giao diện của hệ thống

### 3.2. Nhận diện biển số xe qua camera

Khi người dùng bấm nút xin vào bãi, Python sẽ tự động kích hoạt camera và chụp lại hình ảnh xe. Hình ảnh sau đó được xử lý bằng OpenCV:

- Chuyển ảnh sang grayscale.
- Làm mờ ảnh để giảm nhiễu.
- Dùng các thuật toán phát hiện cạnh (Canny).
- Dùng contour để xác định vùng chứa biển số.
- Cắt vùng biển số ra và gửi vào pytesseract để nhận diện ký tự.

```

# Nhận diện biển số xe
def recognize_plate(image_path):
    try:
        img = cv2.imread(image_path)
        gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
        thresh = cv2.threshold(gray, 150, 255, cv2.THRESH_BINARY)[1]
        text = pytesseract.image_to_string(thresh, lang='eng')
        text = text.strip().upper().replace(" ", "")
        if len(text) >= 6:
            return text
        return "UNKNOWN"
    except Exception as e:
        print(f"OCR Error: {str(e)}")
        return "ERROR"

# Phân biệt loại xe dựa trên biển số

```

Sau khi nhận diện thành công, biển số sẽ được gán vào ID thẻ đã cấp và lưu xuống database.

### 3.3. Tạo và gán thẻ ID cho xe

Các thẻ ID trong hệ thống là cố định và được quản lý trong một danh sách có sẵn. Khi xe vào, hệ thống sẽ:

- Lấy một ID chưa sử dụng (ví dụ ID004).
- Gán ID đó với biển số xe vừa nhận diện được.
- Lưu thông tin vào bảng `current_vehicles` gồm các trường: `id`, `plate`, `zone`, `time_in`.

Mỗi ID chỉ được dùng cho 1 xe đang đỗ, khi xe ra thì ID đó sẽ được giải phóng để tái sử dụng.

### 3.4. Quản lý thông tin xe và tính phí

Giao diện phần mềm hiển thị danh sách các xe đang gửi trong bãi (kèm biển số, ID, thời gian vào, khu vực). Khi xe trả thẻ và rời đi, hệ thống:

- Kiểm tra thẻ ID có hợp lệ không.
- So khớp biển số và ID.
- Tính phí giữ xe dựa trên thời gian gửi (có thể dùng giá cố định hoặc tính theo giờ).
- Hiển thị thông tin và xác nhận thanh toán.
- Chuyển dữ liệu xe qua bảng lịch sử.

```
# Tính phí và cập nhật thời gian ra
time_in = time.mktime(time.strptime(time_in_str, "%Y-%m-%d %H:%M:%S"))
fee = calculate_parking_fee(vehicle_type, time_in)
time_out = time.strptime("%Y-%m-%d %H:%M:%S", time.localtime())

# Cập nhật bảng và xóa khỏi danh sách
tree.item(selected_item, values=(license_plate, time_in_str, time_out, f"{fee} VND", vehicle_type, item["values"][5]))
if vehicle_type == "Xe máy":
    motorbike_spots[:] = [spot for spot in motorbike_spots if spot[0] != license_plate]
else:
    car_spots[:] = [spot for spot in car_spots if spot[0] != license_plate]

messagebox.showinfo("Thanh toán", f"Phí đậu xe cho {license_plate} là {fee} VND.")
```

### 3.5. Giao diện người dùng bằng Tkinter

Phần mềm có giao diện đồ họa đơn giản, gồm các khu vực chính:

- Danh sách xe đang gửi (TreeView hoặc Listbox).
- Nút “Xem lịch sử” → mở cửa sổ hiển thị các xe đã rời bãi.
- Nút “Xóa lịch sử” → xóa dữ liệu trong bảng history.
- Nút “Thoát” → đóng phần mềm.
- Các thông báo như: “Phát thẻ thành công”, “Biển số đã nhận diện”, “Thanh toán hoàn tất”, v.v.

Giao diện tuy cơ bản nhưng dễ sử dụng và có thể mở rộng sau này bằng các công cụ hiện đại hơn (PyQt5, Kivy...).

## CHƯƠNG 4: KẾT NỐI ARDUINO VỚI PHẦN MỀM VÀ CƠ SỞ DỮ LIỆU

### 4.1. Giao tiếp giữa Arduino và Python

Hệ thống sử dụng cổng Serial (UART) để trao đổi dữ liệu giữa Arduino và Python. Khi người dùng nhấn nút “xe vào” hoặc “xe ra” trên mạch Arduino, Arduino sẽ gửi tín hiệu cụ thể (dưới dạng chuỗi) đến phần mềm Python thông qua cổng COM. Phần mềm Python sẽ lắng nghe cổng Serial để xác định hành động tương ứng.

Giao tiếp đơn giản này đảm bảo phản hồi thời gian thực giữa hai hệ thống.



```

Ví dụ trong Arduino.cpp
void loop() {
    if (digitalRead(nutXeVao) == LOW) {
        Serial.println("XE_VAO");
        delay(1000);
    }
    if (digitalRead(nutXeRa) == LOW) {
        Serial.println("XE_RA");
        delay(1000);
    }
}

python
import serial

ser = serial.Serial('COM3', 9600)

while True:
    if ser.in_waiting > 0:
        command = ser.readline().decode().strip()
        if command == 'XE_VAO':
            handle_entry()
        elif command == 'XE_RA':
            handle_exit()

```

#### 4.2. Đồng bộ trạng thái mô phỏng và phần mềm

Phần mềm Python sau khi xử lý thành công quá trình cấp thẻ và nhận dạng biển số xe (hoặc xác nhận thanh toán khi xe ra) sẽ gửi ngược lại tín hiệu cho Arduino để mở barrier. Mã lệnh có thể là:

- OPEN\_BARRIER: yêu cầu mở barrier
- ERROR: có lỗi, không mở

Việc đồng bộ này đảm bảo cả phần mô phỏng và phần mềm luôn hoạt động nhất quán.



### 4.3. Cơ sở dữ liệu và xử lý dữ liệu

Hệ thống sử dụng cơ sở dữ liệu MySQL để lưu trữ thông tin xe. Cấu trúc bao gồm hai bảng chính:

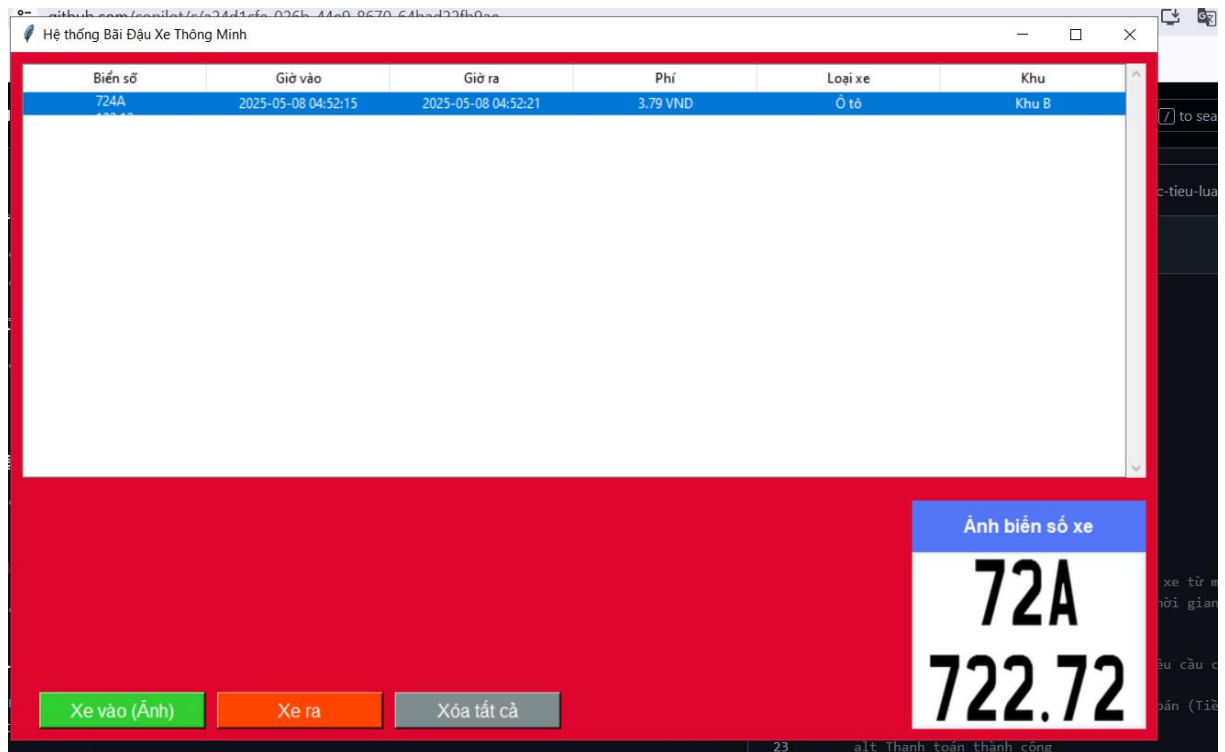
Lưu Xe Trong Bãi:

id	Bản số	khu	Thời gian vào
Id 0001	72A72272	B	2025-05-08 12:55:23

Lưu lịch sử xe rời bãi:

id	Bản số	khu	Thời gian ra
Id 0001	72A72272	B	2025-05-08 14:55:23

```
# Lưu xe vào bãi đậu
def park_vehicle(tree, license_plate, vehicle_type):
    time_in = time.strftime("%Y-%m-%d %H:%M:%S", time.localtime())
    if vehicle_type == 'motorbike':
        if len(motorbike_spots) < total_motorbike_spots:
            motorbike_spots.append((license_plate, time.time()))
            tree.insert("", "end", values=(license_plate, time_in, "None", "None", "Xe máy", "Khu A"))
            messagebox.showinfo("Thông báo", f"Xe máy {license_plate} đã được đậu vào khu A.")
        else:
            messagebox.showwarning("Bãi đậu đầy", "Khu A (xe máy) đã đầy.")
    else:
        if len(car_spots) < total_car_spots:
            car_spots.append((license_plate, time.time()))
            tree.insert("", "end", values=(license_plate, time_in, "None", "None", "Ô tô", "Khu B"))
            messagebox.showinfo("Thông báo", f"Ô tô {license_plate} đã được đậu vào khu B.")
        else:
            messagebox.showwarning("Bãi đậu đầy", "Khu B (ô tô) đã đầy.")
```



## CHƯƠNG 5: ĐÁNH GIÁ HỆ THỐNG

### 5.1. Ưu điểm

Hệ thống hoạt động mượt với giao diện thân thiện, đơn giản và trực quan.

Tự động nhận diện biển số xe nhờ kết hợp OpenCV và Tesseract OCR.

Gán thẻ ID cố định cho từng xe giúp quản lý đơn giản mà hiệu quả.

Chia khu xe máy và ô tô tự động theo biển số.

Dữ liệu được lưu trữ rõ ràng, hỗ trợ truy xuất lịch sử.

Mô phỏng sát thực tế, có khả năng triển khai mở rộng.

### 5.2. Hạn chế

Phụ thuộc vào chất lượng ảnh và góc chụp của camera, OCR có thể sai nếu biển số bị mờ/bẩn.

Phân khu xe máy/ô tô bằng biển số có thể sai trong một số trường hợp đặc biệt.

Việc sử dụng nút nhấn và không có cảm biến làm giảm tính tự động hóa.

Giao diện còn cơ bản, chưa thân thiện về mặt trải nghiệm người dùng hiện đại.

## CHƯƠNG 6: HƯỚNG PHÁT TRIỂN HỆ THỐNG

Trong tương lai, hệ thống có thể được nâng cấp với các tính năng sau:

- Tích hợp cảm biến siêu âm để phát hiện xe đến/lùi vào một cách tự động, không cần nhấn nút.
- Bổ sung camera phân làn riêng cho xe máy và ô tô, tăng độ chính xác phân loại.
- Cải thiện thuật toán xử lý ảnh để nhận diện biển số tốt hơn trong điều kiện ánh sáng yếu.
- Giao diện chuyển sang framework mạnh hơn như PyQt5 hoặc phát triển web-app để quản lý từ xa.
- Kết hợp thanh toán tự động qua mã QR, ví điện tử (Momo, ZaloPay).
- Mở rộng lên quy mô thực tế cho trường học, công ty hoặc bãi xe dân cư.

## CHƯƠNG 7: KẾT LUẬN

Đề tài “Mô phỏng hệ thống bãi giữ xe thông minh sử dụng Arduino và Python” đã được hoàn thành với các chức năng cơ bản: nhận diện biển số, phát thẻ ID cố định, phân khu xe máy – ô tô, thanh toán phí và lưu trữ lịch sử. Hệ thống sử dụng các công nghệ phổ biến và dễ triển khai như Arduino, Python, MySQL, OpenCV, Tesseract OCR.

Dù chỉ là mô phỏng, nhưng hệ thống đã thể hiện khả năng ứng dụng thực tế cao và có thể nâng cấp dễ dàng. Việc chia tách rõ ràng giữa phần cứng (Arduino) và phần mềm (Python) giúp dễ dàng mở rộng và phát triển tiếp tục trong tương lai.

## CHƯƠNG 8: TÀI LIỆU THAM KHẢO

Arduino Documentation <https://www.arduino.cc/reference/en/>

OpenCV Python Documentation <https://docs.opencv.org/>

Tesseract OCR GitHub <https://github.com/tesseract-ocr/tesseract>

Tkinter      GUI      Programming      by      Python      Docs  
<https://docs.python.org/3/library/tkinter.html>

MySQL Reference Manual <https://dev.mysql.com/doc/>