

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



BÀI TẬP LỚN IOT

**Đề tài : Hệ Thống Giám Sát Bãi Đỗ
Xe PTIT**

Nhóm : 8-1

Sinh viên thực hiện:

Hoàng Đình Nam: B21DCCN544

Trần Quang Huy: B21DCCN441

Hà Nội – 2024

Lời cảm ơn

Đầu tiên, chúng em xin gửi lời cảm ơn sâu sắc đến Học viện nghệ Bưu chính Viễn thông và khoa CNTT1 đã đưa môn học IOT vào trong chương trình giảng dạy. Đặc biệt, chúng em xin gửi lời cảm ơn sâu sắc đến giảng viên bộ môn thầy Nguyễn Tài Tuyên đã dạy dỗ, rèn luyện và truyền đạt những kiến thức quý báu cho chúng em trong suốt thời gian học tập vừa qua.

Trong thời gian được tham dự lớp học của thầy, chúng em đã được tiếp thu thêm nhiều kiến thức bổ ích, học tập được tinh thần làm việc hiệu quả, nghiêm túc. Đây thực là những điều rất cần thiết cho quá trình học tập và công tác sau này của chúng em. Thêm vào đó, nhờ sự dẫn dắt và chỉ bảo của thầy, chúng em đã thực hiện được một đề tài bài tập lớn hoàn chỉnh cho môn học này, chúng em rất biết ơn điều đó. Trong quá trình làm không tránh được sai sót mong được thầy góp ý để chúng em có thể hoàn thành tốt vào các bài báo cáo sau.

Chúng em chúc thầy luôn khỏe mạnh và tiếp tục đạt được nhiều thành công trong công việc và cuộc sống. Chúng em xin chân thành cảm ơn!

MỤC LỤC

I. Lý do lựa chọn đề tài	4
1. Tính cấp thiết và thực tiễn	4
2. Khả năng sáng tạo và phát triển.....	4
3. Giá trị học thuật	4
4. Tiềm năng nghiên cứu sâu hơn.....	4
II. Nguyên lí hoạt động của hệ thống	5
Ý tưởng chính	5
Các thành phần chính và cách thức hoạt động:	5
Quy trình hoạt động	6
Các yếu tố cần cân nhắc.....	6
Mã Nguồn Arduino	6
Mở rộng:	12
III. Các ưu điểm của hệ thống	13
IV. Demo app	14
1. Trạng thái vị trí đỗ xe	14
2. Thống kê sử dụng.....	16
V. Tài Liệu Tham Khảo	18

I. Lý do lựa chọn đề tài

Bạn đã bao giờ loay hoay tìm chỗ đỗ xe trong một bãi đỗ đông đúc?

1. Tính cấp thiết và thực tiễn

- **Vấn đề đô thị:** Ở các thành phố lớn, việc tìm chỗ đỗ xe luôn là một thách thức lớn, gây ra ùn tắc giao thông và lãng phí thời gian.
- **Cơ hội ứng dụng công nghệ:** IoT cung cấp các giải pháp hiệu quả để quản lý bãi đỗ xe, từ đó giải quyết vấn đề trên.
- **Tiềm năng thương mại:** Hệ thống bãi đỗ xe thông minh có thể tạo ra nhiều mô hình kinh doanh mới, mang lại lợi nhuận.

2. Khả năng sáng tạo và phát triển

- **Đa dạng công nghệ:** Bạn có thể kết hợp nhiều công nghệ IoT khác nhau như cảm biến, RFID, camera, ứng dụng di động để xây dựng hệ thống.
- **Cải tiến liên tục:** Công nghệ IoT luôn phát triển, bạn có thể liên tục cập nhật và nâng cấp hệ thống của mình.
- **Khả năng mở rộng:** Hệ thống bãi đỗ xe thông minh có thể được mở rộng để quản lý nhiều bãi đỗ xe khác nhau, tạo thành một mạng lưới lớn.

3. Giá trị học thuật

- **Thực hành kiến thức:** Đề tài này giúp bạn áp dụng kiến thức về IoT vào một vấn đề thực tế, rèn luyện kỹ năng thiết kế, lập trình và triển khai hệ thống.
- **Đóng góp cho cộng đồng:** Giải pháp của bạn có thể cải thiện chất lượng cuộc sống của mọi người, giảm thiểu ùn tắc giao thông và ô nhiễm môi trường.
- **Mở rộng kiến thức:** Trong quá trình nghiên cứu, bạn sẽ tiếp cận với nhiều lĩnh vực khác nhau như quản lý dữ liệu, trí tuệ nhân tạo, giao diện người dùng.

4. Tiềm năng nghiên cứu sâu hơn

- **Tối ưu hóa thuật toán:** Bạn có thể nghiên cứu các thuật toán để tối ưu hóa việc quản lý chỗ đỗ, giảm thiểu thời gian tìm kiếm.

- **An ninh mạng:** Xây dựng hệ thống bảo mật để bảo vệ dữ liệu của người dùng và hệ thống khỏi các cuộc tấn công.
- **Tích hợp với các hệ thống khác:** Kết hợp hệ thống bãi đỗ xe thông minh với các hệ thống giao thông thông minh khác để tạo ra một thành phố thông minh.

II. Nguyên lí hoạt động của hệ thống

Ý tưởng chính

Hệ thống bãi đỗ xe thông minh này tận dụng các công nghệ IoT để tự động hóa việc quản lý chỗ đỗ xe, cung cấp thông tin chính xác và tiện lợi cho người dùng thông qua ứng dụng di động.

Các thành phần chính và cách thức hoạt động:

1. **Cảm biến hồng ngoại:**
 - Được lắp đặt ở mỗi vị trí đỗ xe.
 - Khi có xe đỗ vào, chùm tia hồng ngoại bị chắn, cảm biến sẽ phát hiện và gửi tín hiệu về ESP32.
 - Khi xe rời đi, chùm tia hồng ngoại không bị chắn, cảm biến gửi tín hiệu về ESP32 cho biết vị trí đó đã trống.
2. **ESP32:**
 - Là một vi điều khiển có khả năng kết nối Wi-Fi và Bluetooth.
 - Thu thập dữ liệu từ các cảm biến hồng ngoại.
 - Xử lý dữ liệu và cập nhật trạng thái của từng vị trí đỗ xe.
 - Truyền dữ liệu về trạng thái các vị trí đỗ xe lên một nền tảng đám mây hoặc một máy chủ.
3. **Camera ESP32:**
 - Hình ảnh từ camera có thể được xử lý để nhận diện biển số xe, giúp quản lý và tìm kiếm xe dễ dàng hơn.
4. **Ứng dụng di động:**
 - Người dùng tải ứng dụng về điện thoại.
 - Ứng dụng kết nối với nền tảng đám mây hoặc máy chủ để lấy thông tin về các vị trí đỗ xe trống.

- Hiển thị thông tin trên bản đồ hoặc danh sách, giúp người dùng dễ dàng tìm được vị trí đỗ xe phù hợp.

Quy trình hoạt động

1. **Thu thập dữ liệu:** Các cảm biến hồng ngoại liên tục gửi dữ liệu về trạng thái của từng vị trí đỗ xe về ESP32. ESP32 xử lý dữ liệu và truyền lên nền tảng đám mây.
2. **Xử lý dữ liệu:** Trên nền tảng đám mây, dữ liệu từ các cảm biến được xử lý và cập nhật vào cơ sở dữ liệu trên firebase.
3. **Hiển thị thông tin:** Ứng dụng di động lấy dữ liệu từ nền tảng đám mây và hiển thị thông tin về các vị trí đỗ xe trống cho người dùng.
4. **Tương tác người dùng:** Người dùng chọn vị trí đỗ xe

Các yếu tố cần cân nhắc

- **Độ phủ sóng Wi-Fi:** Đảm bảo kết nối Wi-Fi ổn định để truyền dữ liệu từ ESP32 lên nền tảng đám mây.
- **Độ chính xác của cảm biến:** Chọn loại cảm biến hồng ngoại phù hợp để đảm bảo độ chính xác cao.
- **Khả năng xử lý của ESP32:** ESP32 cần có đủ khả năng để xử lý dữ liệu từ nhiều cảm biến và camera.
- **Bảo mật:** Đảm bảo an toàn cho dữ liệu của người dùng và hệ thống.

Mã Nguồn Arduino

1. Khai báo thư viện và thông tin cảm biến

```
#include <WiFi.h>
```

```
#include <FirebaseESP32.h>
```

```
#define SENSOR1 32
```

```
#define SENSOR2 33
```

```
#define SENSOR3 25
```

```
#define SENSOR4 26
```

- **Thư viện:**
 - WiFi.h: Dùng để kết nối ESP32 với Wi-Fi.
 - FirebaseESP32.h: Thư viện giúp kết nối và tương tác với Firebase Realtime Database.
- **Khai báo cảm biến:**
 - Các chân GPIO trên ESP32 được gán với cảm biến hồng ngoại (IR Sensors).

2. Thông tin Wi-Fi và Firebase

```
const char *ssid = "Galaxy A047787";
```

```
const char *pass = "dung12345";
```

```
#define FIREBASE_HOST "test-7dfa4-default-rtdb.asia-southeast1.firebaseio.com"
```

```
#define FIREBASE_AUTH  
"ChE5mR3x5FcY4Eab98BEddrimTgILxtPkWDJo2SQ"
```

```
FirebaseData firebaseData;
```

```
FirebaseConfig config;
```

```
FirebaseAuth auth;
```

- **Wi-Fi:**
 - ssid: Tên mạng Wi-Fi.
 - pass: Mật khẩu Wi-Fi.
- **Firebase:**
 - FIREBASE_HOST: URL của Firebase Realtime Database.
 - FIREBASE_AUTH: Token xác thực để kết nối Firebase.
- **FirebaseData:** Biến lưu trữ dữ liệu giao tiếp với Firebase.
- **FirebaseConfig và FirebaseAuth:** Cấu hình và xác thực cho Firebase.

3. Hàm setup() - Cấu hình ban đầu

```
void setup()
```

```
{
```

```
Serial.begin(115200);

// Kết nối Wi-Fi

WiFi.begin(ssid, pass);

while (WiFi.status() != WL_CONNECTED)

{

    delay(500);

    Serial.print(".");

}

Serial.println("\nConnected to Wi-Fi");

Serial.println(WiFi.localIP());

config.host = FIREBASE_HOST;

config.database_url = "https://" FIREBASE_HOST;

config.signer.tokens.legacy_token = FIREBASE_AUTH;

Firebase.begin(&config, &auth);

Firebase.reconnectWiFi(true);

// Khởi tạo các cảm biến

pinMode(SENSOR1, INPUT_PULLUP);

pinMode(SENSOR2, INPUT_PULLUP);

pinMode(SENSOR3, INPUT_PULLUP);

pinMode(SENSOR4, INPUT_PULLUP);

}
```


- **Wi-Fi:**
 - ESP32 kết nối Wi-Fi, trạng thái kết nối được kiểm tra liên tục bằng `WiFi.status()`.
- **Firebase:**
 - Cấu hình và khởi tạo kết nối Firebase.
 - `Firebase.reconnectWiFi(true)`: Tự động kết nối lại Wi-Fi khi bị mất kết nối.
- **Cảm biến:**
 - `INPUT_PULLUP`: Cấu hình chân GPIO đọc tín hiệu từ cảm biến hồng ngoại.

4. Cập nhật trạng thái cảm biến vào Firebase

```
void updateSensorStatus(int sensorNumber, int status)
```

```
{
    String path = "/parking/sensor" + String(sensorNumber);
    if (Firebase.setInt(firebaseData, path, status))
    {
        Serial.println("Firebase update successful: " + path + " = " +
String(status));
    }
    else
    {
        Serial.println("Firebase update failed: " + firebaseData.errorReason());
    }
}
```

- Hàm này cập nhật trạng thái (có xe hoặc không) vào Firebase:
 - `path`: Đường dẫn đến cảm biến trong Firebase, ví dụ: `/parking/sensor1`.
 - `Firebase.setInt()`: Ghi giá trị status (0 hoặc 1) vào Firebase.

- Trạng thái ghi được hiển thị trên Serial Monitor.

5. Cập nhật số lần sử dụng của cảm biến

```
void updateSensorSoLanSuDung(int sensorNumber)
{
    String path = "/soLanSuDung/sensor" + String(sensorNumber);

    if (Firebase.getInt(firebaseData, path))
    {
        int currentCount = firebaseData.intData();

        Serial.println("Current usage count for sensor " + String(sensorNumber) + ": "
+ String(currentCount));

        if (Firebase.setInt(firebaseData, path, currentCount + 1))
        {
            Serial.println("Updated usage count for sensor " + String(sensorNumber) +
": " + String(currentCount + 1));
        }
        else
        {
            Serial.println("Failed to update usage count for sensor " +
String(sensorNumber) + ": " + firebaseData.errorReason());
        }
    }
}
```

```

else

{

    Serial.println("Failed to get usage count for sensor " + String(sensorNumber)
+ ": " + firebaseData.errorReason());

}

}

```

- Hàm này:
 - Lấy số lần sử dụng hiện tại từ Firebase bằng Firebase.getInt().
 - Tăng giá trị số lần sử dụng lên 1 và cập nhật lại Firebase bằng Firebase.setInt().

6. Đọc trạng thái cảm biến và xử lý

```

void sensor()

{

    int L1 = digitalRead(SENSOR1);

    int L2 = digitalRead(SENSOR2);

    int L3 = digitalRead(SENSOR3);

    int L4 = digitalRead(SENSOR4);


    if (L1 == 1)

    {

        Serial.println("IR SENSOR 1 detected");

        updateSensorStatus(1, 1);

        updateSensorSoLanSuDung(1);

```

```

    }

    else

    {

        Serial.println("1 === All clear");

        updateSensorStatus(1, 0);

    }

    // Tương tự cho SENSOR2, SENSOR3, SENSOR4

}

```

- Hàm đọc tín hiệu từ các cảm biến:
 - `digitalRead()`: Đọc trạng thái (1: có xe, 0: không có xe).
 - Nếu phát hiện xe (1), cập nhật trạng thái và tăng số lần sử dụng trong Firebase.
 - Nếu không có xe (0), cập nhật trạng thái là 0.

7. Hàm `loop()` - Vòng lặp chính

```

void loop()

{

    sensor();

    delay(1000);

}

```

Mở rộng:

Hệ thống có thể được mở rộng thêm các tính năng như:

- **Nhận diện biển số xe:** Giúp quản lý xe và tìm kiếm xe dễ dàng hơn dựa vào định danh biển số xe..

- **Phân tích dữ liệu:** Thu thập và phân tích dữ liệu để tối ưu hóa việc quản lý bãi đỗ xe.

III. Các ưu điểm của hệ thống

1.Quản lý hiệu quả

Theo dõi số lần mỗi vị trí được sử dụng, từ đó đánh giá mức độ phổ biến và lên kế hoạch bố trí phù hợp.

2.Tối ưu doanh thu

Xác định các khung giờ cao điểm, các vị trí đỗ "hot" để điều chỉnh giá cả hoặc các dịch vụ đi kèm.

3.Bảo trì hệ thống

Dựa vào số lần sử dụng và thời gian hoạt động của các thiết bị, lên kế hoạch bảo dưỡng định kỳ.

4.Cải thiện trải nghiệm khách hàng

Giúp khách hàng dễ dàng tìm được vị trí đỗ trống và rút ngắn thời gian chờ đợi.

5.Tối ưu nguồn lực

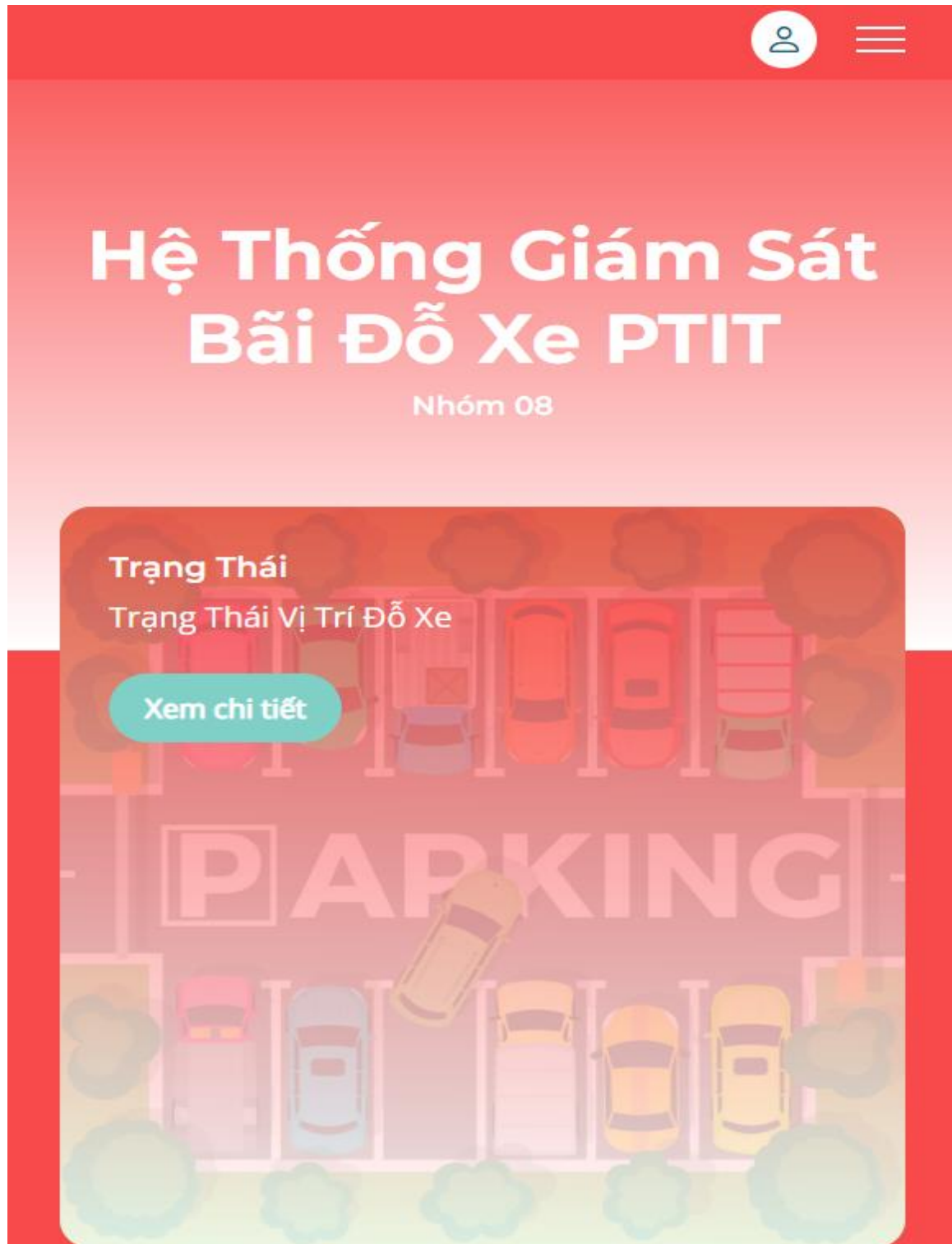
Giúp chủ sở hữu bãi đỗ xe sử dụng nguồn lực hiệu quả hơn dựa vào dữ liệu thu thập được từ hệ thống cơ sở dữ liệu.

6.Nâng cao độ an toàn

Giúp theo dõi và quản lý giám sát tại bãi đỗ xe tốt hơn, hiệu quả an toàn cao hơn.

IV. Demo app

1. Trạng thái vị trí đỗ xe





Trạng Thái Vị Trí

Vị trí 1



Vị trí 2



Vị trí 3

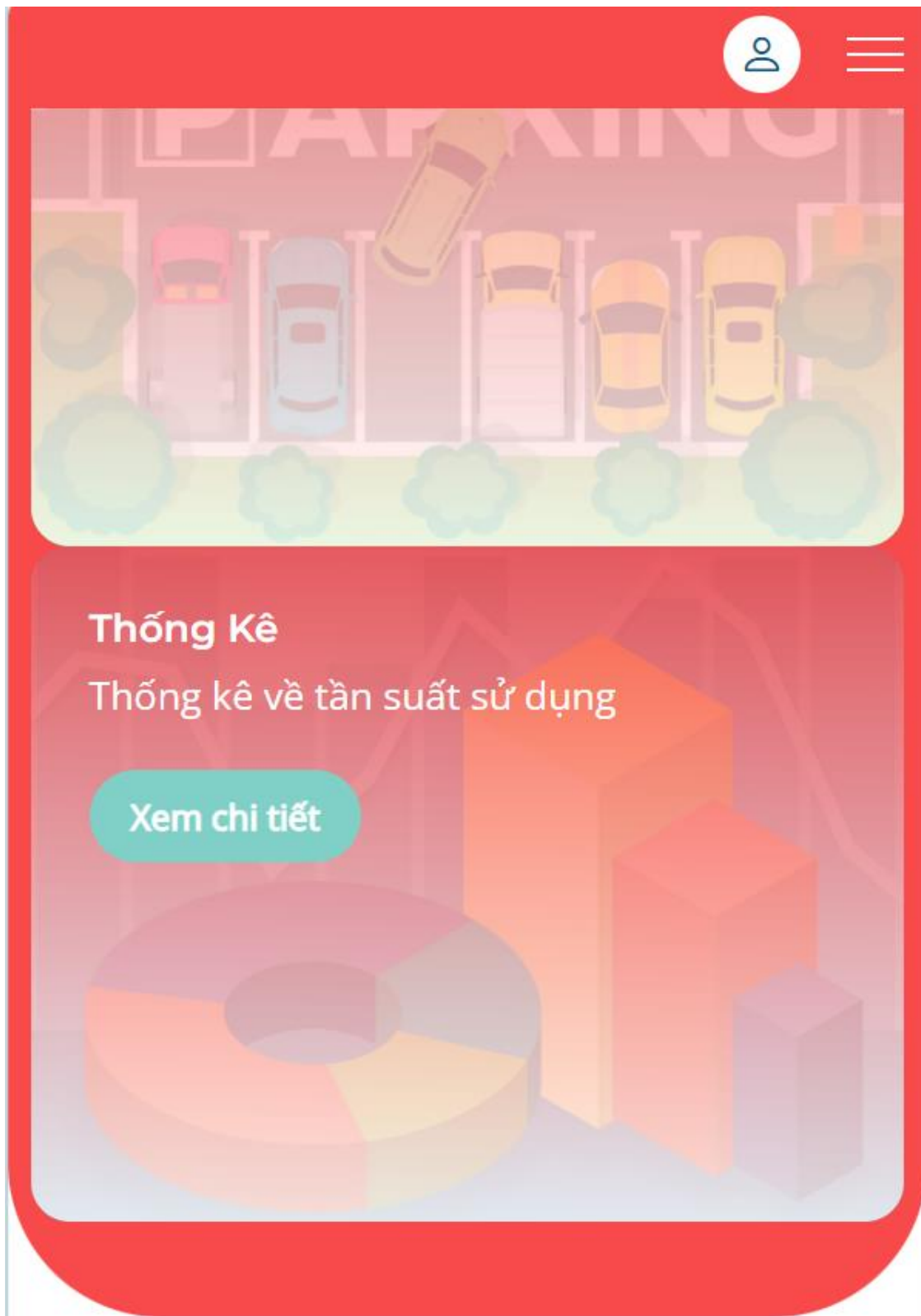


Vị trí 4



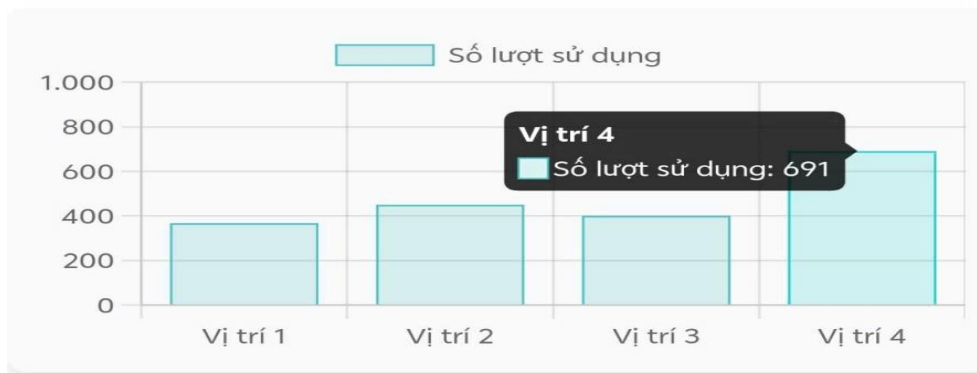
Trạng thái các vị trí đỗ xe. Màu đỏ tương ứng với vị trí đã đầy và màu xanh tương ứng với vị trí trống.

2. Thống kê sử dụng





Thống Kê Số Lượt Sử Dụng



Thống Kê Thời Gian Sử Dụng



V. Tài Liệu Tham Khảo

- [1] S. Nandyal, S. Sultana, and S. Anjum, “Smart Car Parking System using Arduino UNO,” *International Journal of Computer Applications*, vol. 169, no. 1, pp. 13–18, 2017.**
- [2] H. Kaur, and J. Malhotra, “A Review of Smart Parking System based on Internet of Things,” *International Journal of Intelligent Systems and Applications in Engineering (IJISAE)*, vol. 6, no. 4, pp. 248–250, 2018.**
- [3] A. Khanna, and R. Anand, “IoT based Smart Parking System,” *International Conference on Internet of Things and Applications (IOTA)*, pp. 266–270, 2016.**
- [4] S. S. Thorat et al., “IoT Based Smart Parking System Using RFID,” *International Journal of Computer Engineering in Research Trends*, vol. 4, no. 1, pp. 9-12, 2017**
- [5] S. Dey, A. Roy, and S. Das, “Smart Car Parking System using Internet of Thing,” *7th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, pp. 1–6, 2016.**
- [6] N. V. Chokshi, “A Survey: Smart Parking System using Internet of Things (IoT),” *International Journal of Advance Engineering and Research Development*, vol. 4, no. 4, pp. 847–849, 2017.**
- [7] Abhay Deshmukh, and Radhika D. Joshi, “Understanding the Architecture of Internet of Things using a Case Study of Smart Parking,” *Asian Journal of Convergence in Technology*, vol. 5, no. 1, 2019.**
- [8] Mohd Mustari Syafiq Ismai et al., “IoT Based Smart Parking System,” *Journal of Physics; Conference series*, 2nd International Conference on Advance & Scientific Innovation, vol. 1424, 2019.**