**Limitation of Smart Parking System**

1. **This System Effectively detects 4 Wheelers only.**

Sensors used in this system can detect vehicles only when sensors reach maximum threshold set as 2 wheelers cannot reach threshold it becomes difficult to identify such vehicles.

1. **Stable Internet connection is always required for effective data transfer.**

As the system uses wifi modules for establishing communication between cloud and components, stable internet connection is needed for effective communication between modules to coordinate with each other.

1. **Centralized Broker can limit scale.**

The Broker can affect scalability as there is additional overhead for each device connected to it. The network can only grow as large as the local broker hub can support.

1. **Time Consumption**

As the system reduces manual intervention, the time consumed for processing the bill and payment might be slight more time than manual systems.

**Test Cases for Smart Parking System**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test\_ID** | **Testcase** | **Expected input** | **Expected Output** | **Actual Output** |
| TH\_01 | QR code scanning | User Scans QR code generated in application after successful slot booking | User details gets stored in database and sends signal to microcontroller | Barricade opens on successful scanning |
| TH\_02 | Barricade opening | On successful QR code validation | Microcontroller triggers DC motor in barricade to lift from 0 to 90 degrees | Barricade opens and allows vehicles to pass in. |
| TH\_03 | Vehicle passing through barricade | When vehicle passes sensors in barricade will detect | Sensors triggers microcontroller to close barricade immediately after vehicle passes. | Barricade closes immediately after vehicle passing through it. |
| TH\_04 | Parking slot sensor | Vehicle is parked in parking slot | Sensors in parking slot sends data of slot occupancy to the cloud | Sensors in parking slot sends data of slot occupancy to the cloud |
| TH\_05 | QR code scanning | User Scans QR code generated in application after successful slot booking | User details gets stored in database and sends signal to microcontroller | Barricade opens on successful scanning |
| TA\_06 | Registration | User enters valid phone number and vehicle number | Registration Successful | Registration Successful and redirects to menu page |
| TA\_07 | Registration | User enters invalid phone number or vehicle number | Registration Unsuccessful | Notifies the user to re-enter a valid phone number or vehicle number |
| TA\_08 | Slot Booking | User Chooses time for the slot | Slot Booked and QR code displayed | Slot Booked, QR code and slot details are displayed. |
| TA\_09 | Slot Booking | User Chooses Invalid time for the slot (more than 30 Minutes) | Invalid time | Notifies the user to book slot only for next 30 minutes |
| TA\_10 | Check out | User scans valid QR code provided while booking the slot | Proceed to billing | Bill displayed with payment options |
| TA\_11 | Check out | User scans invalid QR code | Invalid QR code | Invalid QR code |
| TA\_12 | Billing | User enters valid credentials for payment | Proceed with Payment | Payment Successful |
| TA\_13 | Billing | User enters invalid credentials for payment | Invalid credentials | Invalid UPI/Card details |
| TC\_14 | Client establishing connection with broker | Mqttclient.connect() valid client ID, User name and Password | Connection established between mqtt broker and client | Successful connection |
| TC\_15 | The server must validate connect packet confirmation | Incorrect credentials, validation fails | Closes the network connection without sending acknowledgement confirmation | Closes the network connection |
| TC\_16 | Publish packet must not have both QOS bits set to 1 | Invalid value | Closes the network connection because of QOS out of range | Closes the network connection |

TH – Test Cases for Hardware

TA – Test Cases for Android Application

TC – Test Cases for Cloud