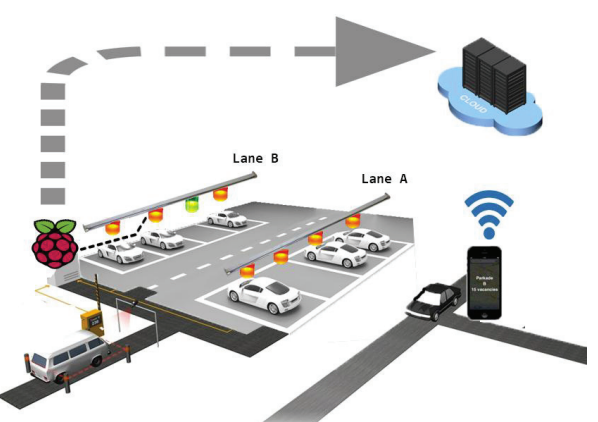
**SMART PARKING**

**Phase 3: DEVELOPMENT PART 2**

* **To Continue Building the Project by Developing the Mobile App using Python.**
* **Also Use a Mobile App Development Framework (e.g., Flutter) to create an App that Displays RealTime Parking Availability.**
* **To Design App Functions to Receive and Display Parking Availability Data Received From the Raspberry Pi.**

**REQUIREMENTS:**

* **Raspberry Pi 4**
* **Pi Camera Module**
* **GPS Sensor**
* **Ultrasonic Sensors**
* **Firebase**
* **MIT App Inventor**
* **MQTTServer**

**Developing Mobile App Using Python:**

* **The Mobile App Can Be Created Using Python Kivy Framework.**
* **That Retrieve The Stored Data From The MQTT server, shows the available slot for parking.**

**Python Script:**

**import kivy**

**from kivy.app import App**

**from kivy.uix.label import Label**

**from kivy.uix.gridlayout import GridLayout**

**from kivy.uix.textinput import TextInput**

**from kivy.uix.button import Button**

**from kivy.uix.widget import Widget**

**from kivy.properties import ObjectProperty**

**import paho.mqtt.client as mqtt**

**class SmartParkingApp(App):**

**def build(self):**

**layout = GridLayout(cols=2)**

**self.lot1 = Label(text="Lot 1: Available")**

**layout.add\_widget(self.lot1)**

**self.lot2 = Label(text="Lot 2: Full")**

**layout.add\_widget(self.lot2)**

**return layout**

**def on\_start(self):**

**self.mqtt\_client = mqtt.Client()**

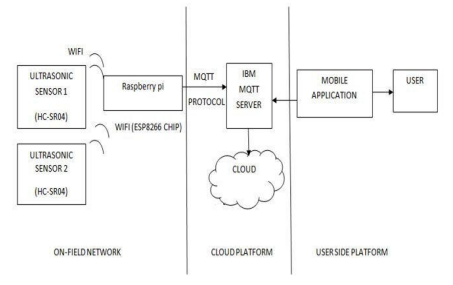
**self.mqtt\_client.connect("mqtt\_broker\_ip", 1883)**

**self.mqtt\_client.subscribe("parking/lot1")**

**self.mqtt\_client.subscribe("parking/lot2")**

**self.mqtt\_client.message\_callback\_add("parking/lot1", self.lot1\_message)**

**self.mqtt\_client.message\_callback\_add("parking/lot2", self.lot2\_message)**

**self.mqtt\_client.loop\_start()**

**def lot1\_message(self, client, userdata, msg):**

**data = msg.payload.decode()**

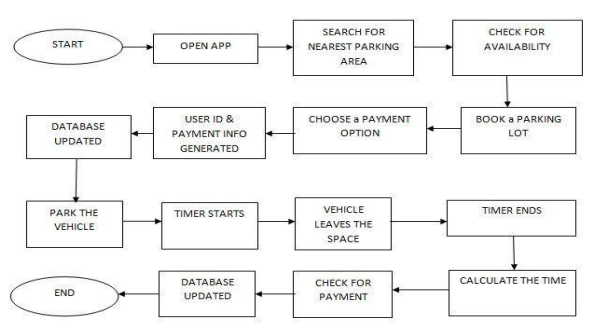
**self.lot1.text = "Lot 1: " + data**

**def lot2\_message(self, client, userdata, msg):**

**data = msg.payload.decode()**

**self.lot2.text = "Lot 2: " + data**

**SmartParkingApp().run()**



**Mobile App Development Framework:**

* **The Mobile App Can Be Developed Using Varies Frameworks.**
* **Here, I am Using The Flutter Framework Which is Developed by Google.**
* **It Will Show The Real Time Parking Slot Availability.**
* **The Programming Language Used In Flutter Is Dart.**

**Program:**

**import 'package:flutter/material.dart';**

**import 'package:mqtt\_client/mqtt\_client.dart';**

**import 'package:mqtt\_client/mqtt\_server\_client.dart';**

**void main() => runApp(MyApp());**

**class MyApp extends StatelessWidget {**

**@override**

**Widget build(BuildContext context) {**

**return MaterialApp(**

**title: 'Parking Availability',**

**theme: ThemeData(**

**primarySwatch: Colors.blue,**

**),**

**home: MyHomePage(),**

**);**

**}**

**}**

**class MyHomePage extends StatefulWidget {**

**@override**

**\_MyHomePageState createState() => \_MyHomePageState();**

**}**

**class \_MyHomePageState extends State<MyHomePage> {**

**final client = MqttServerClient('localhost', '');**

**int availableSpaces = 0;**

**@override**

**void initState() {**

**super.initState();**

**client.connect();**

**client.subscribe('parking/availability', MqttQos.atLeastOnce);**

**client.updates.listen((List<MqttReceivedMessage<MqttMessage>> c) {**

**final message = c[0].payload as MqttPublishMessage;**

**setState(() {**

**availableSpaces = int.parse(message.payload.toString());**

**});**

**});**

**}**

**@override**

**Widget build(BuildContext context) {**

**return Scaffold(**

**appBar: AppBar(**

**title: Text('Parking Availability'),**

**),**

**body: Center(**

**child: Text(**

**'Available Parking Spaces: $availableSpaces',**

**style: Theme.of(context).textTheme.headline4,**

**),**

**),**

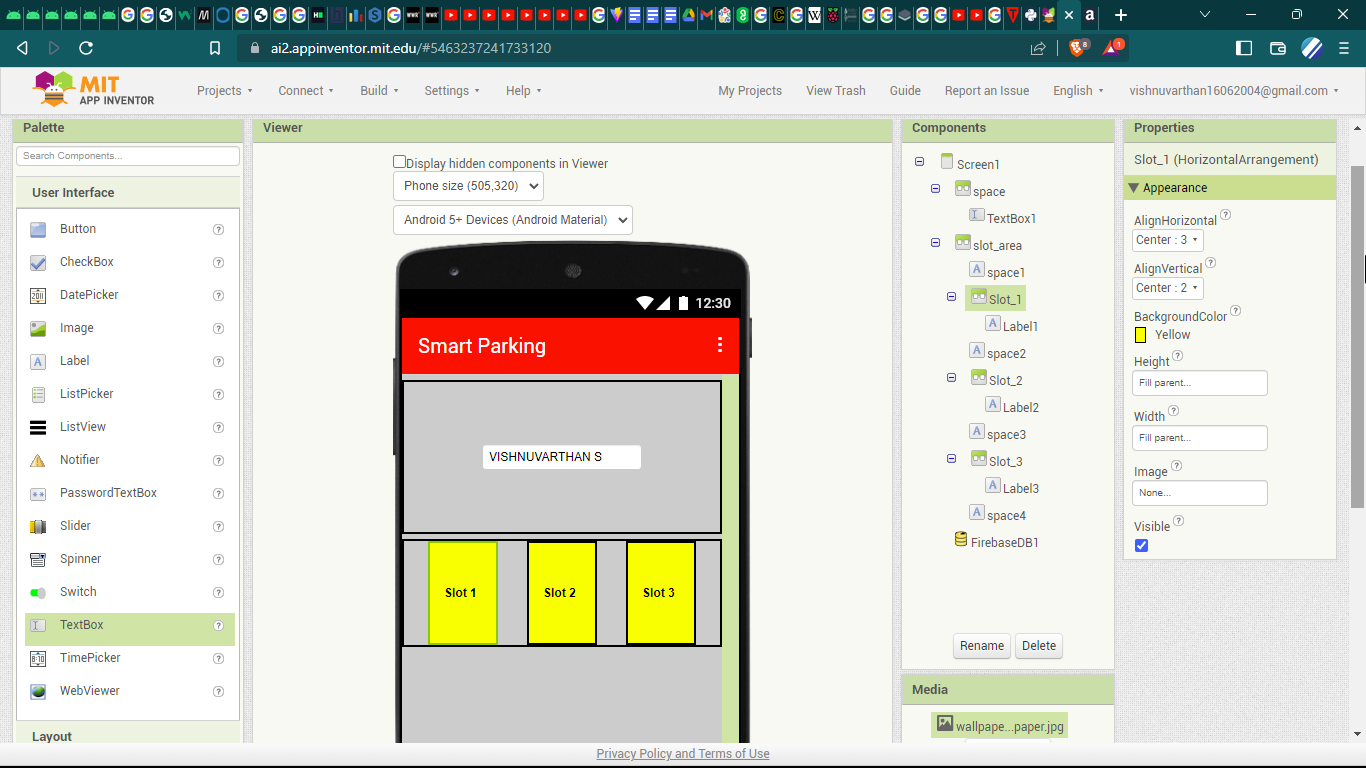
**);**

**}**

**}**

**Designing App to Display Parking Availability:**

* **In This Section I am Going To Design The App to Receive and Display Parking Space Availability From The Data Received Through the Raspberry Pi.**
* **Here I am using Google's Firebase and MIT App Inventor as Mentioned During The Session.**
* **By using the Firebase Token and Firebase URL the Data Can Be Fetched.**



**Prepared By,**

**VISHNUVARTHAN S**