Public Transport Optimization

Phase 3: Development Part 1

Here, we start building the IoT-enabled public transportation optimization system.

Tasks:

- *Deploy IoT sensors (e.g., GPS, passenger counters) in public transportation vehicles to gather data.*
- *Develop a Python script on the IoT sensors to send real-time location and ridership data to the transit information platform.*

We are using ESP32-based Arduino for our project. Arduino programming uses C/C++-like syntax, and thus, our code is written in cpp. We have used java for our mobile application to communicate with the server. The code includes GPS data handling, Wi-Fi connectivity, and passenger counting logic.

microcontroller.cpp

```
#include <TinyGPS++.h>
#include <WiFi.h>
#include <HTTPClient.h>
#include <Wire.h>

#define TX_PIN 2   // TX pin of NEO-6 GPS module
#define RX_PIN 15  // RX pin of NEO-6 GPS module
#define PIR_SENSOR_PIN 12  // Pin for the PIR sensor

const char* ssid = "wifi_ssid";
const char* password = "wifi_password";
const char* serverUrl = "http://server_under_construction";

TinyGPSPlus gps;
int passengerCount = 0;

void setup() {
  Serial.begin(115200);
  Serial1.begin(9600, SERIAL_8N1, RX_PIN, TX_PIN); // Initialize the GPS
module
```

```
  pinMode(PIR_SENSOR_PIN, INPUT);

  // Connect to Wi-Fi
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi...");
  }
  Serial.println("Connected to WiFi");
}

void loop() {
  // Check PIR sensor for passenger detection
  if (digitalRead(PIR_SENSOR_PIN) == HIGH) {
    // Passenger boarded
    passengerCount++;
    Serial.println("Passenger boarded. Count: " + String(passengerCount));
    delay(1000); // Debounce
  }

  while (Serial1.available() > 0) {
    if (gps.encode(Serial1.read())) {
      if (gps.location.isUpdated()) {
        float latitude = gps.location.lat();
        float longitude = gps.location.lng();

        // Send GPS data and passenger count to a server
        sendDataToServer(latitude, longitude, passengerCount);
      }
    }
  }
  delay(10000);  // Update GPS data every 10 seconds
}

void sendDataToServer(float latitude, float longitude, int passengers) {
  HTTPClient http;
  String data = "lat=" + String(latitude, 6) + "&lng=" + String(longitude,
6) + "&passengers=" + String(passengers);
  http.begin(serverUrl);
  http.addHeader("Content-Type", "application/x-www-form-urlencoded");

  int httpResponseCode = http.POST(data);

  if (httpResponseCode > 0) {
    String response = http.getString();
    Serial.println("HTTP Response Code: " + String(httpResponseCode));
    Serial.println(response);
  } else {
    Serial.println("HTTP Error");
  }

  http.end();
}
```

app.java

```
import android.os.AsyncTask;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.TextView;

import org.json.JSONException;
import org.json.JSONObject;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;

public class MainActivity extends AppCompatActivity {
    private TextView gpsTextView;
    private TextView passengerCountTextView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        gpsTextView = findViewById(R.id.gpsTextView);
        passengerCountTextView =
findViewById(R.id.passengerCountTextView);

        // Perform an HTTP request to your server to get GPS data and
passenger count
        new RetrieveGPSDataTask().execute();
    }

    private class RetrieveGPSDataTask extends AsyncTask<Void, Void,
JSONObject> {
        @Override
        protected JSONObject doInBackground(Void... voids) {
            try {
                URL url = new URL("http://server_under_construction");
                HttpURLConnection connection = (HttpURLConnection)
url.openConnection();
                connection.setRequestMethod("GET");

                InputStream inputStream = connection.getInputStream();
                BufferedReader reader = new BufferedReader(new
InputStreamReader(inputStream));
                StringBuilder result = new StringBuilder();
                String line;

                while ((line = reader.readLine()) != null) {
                    result.append(line);
                }

                return new JSONObject(result.toString());
            } catch (IOException | JSONException e) {
                e.printStackTrace();
                return null;
            }
        }

        @Override
        protected void onPostExecute(JSONObject result) {
            if (result != null) {
                try {
                    double latitude = result.getDouble("lat");
                    double longitude = result.getDouble("lng");
                    int passengers = result.getInt("passengers");

                    // Display the GPS coordinates and passenger count in
the app
                    gpsTextView.setText("Latitude: " + latitude +
"\nLongitude: " + longitude);
                    passengerCountTextView.setText("Passengers: " +
passengers);
                } catch (JSONException e) {
                    e.printStackTrace();
                }
            } else {
                gpsTextView.setText("Error retrieving GPS data.");
                passengerCountTextView.setText("Passenger count not
available.");
            }
        }
    }
}
```