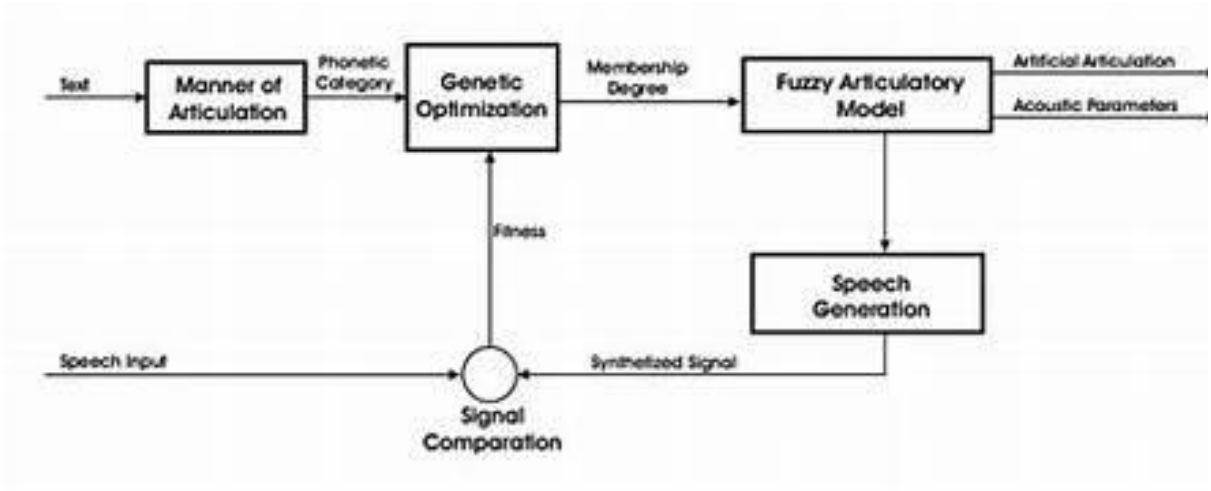
PUBLIC TRANSPORT OPTIMIZATION

Name.sandhuru Chaitanya
Regester no. 723921106018
Mail.chaithuchaitanya786@gmail.com

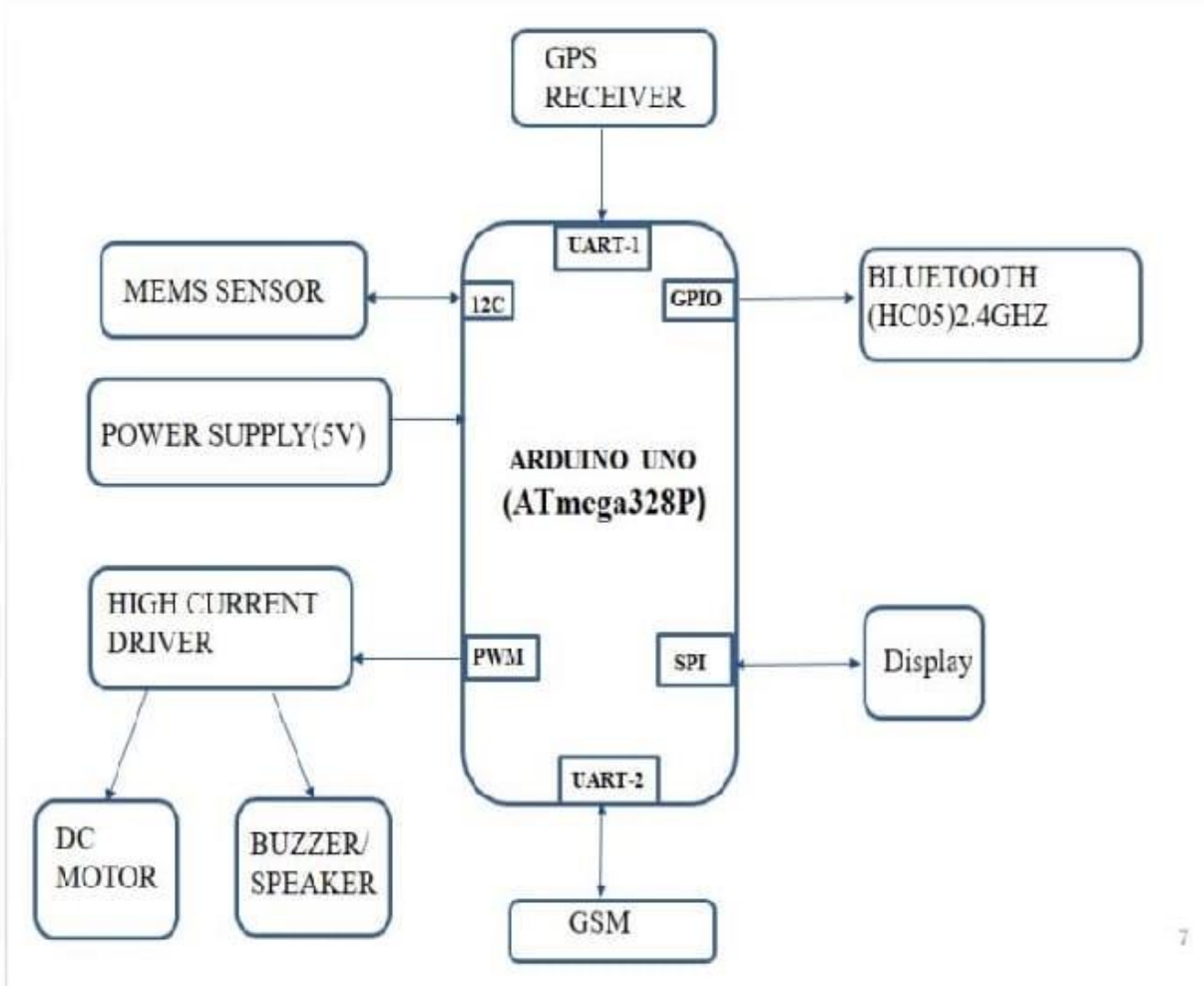
INTRODUCTION:

The Public Transport Optimization Project aims to improve the efficiency, accessibility, and sustainability of the existing public transport network in [City Name]. By leveraging data analytics, technology, and innovative strategies, the project will identify key areas of improvement, develop solutions, and implement changes that benefit both commuters and the environment. This project will take a holistic approach, considering the needs of diverse stakeholders, from the citizens who rely on public transport to the city's policymakers and urban planners.

BLOCK DIAGRAM:



FLOW CHART DESIGN:



CONNECTING MOBILE APPS WITH PUBLIC TRANSPORT OPTIMIZATION:

Connecting a mobile app to a Public Transport Optimization IoT project involves setting up a communication pathway between the mobile app and the IoT devices or backend server. Here's a high-level overview of the steps to achieve this connection:

1.Define App Requirements:

Determine the specific functionalities and features you want to offer in the mobile app. These could include real-time tracking, route information, alerts, and notifications.

2.Choose Development Platforms:

Decide whether you want to develop native apps for specific platforms (e.g., iOS and Android) or use cross-platform frameworks like React Native, Flutter, or Xamarin to build the app for multiple platforms simultaneously.

3.Select Development Tools:

Choose the development tools and integrated development environments (IDEs) suitable for the selected platform and framework.

4.Develop Mobile App:

Create the mobile app using the chosen platform and development tools. Integrate user interfaces, real-time tracking, and any other relevant features.

5.Implement Communication:

To connect the app with IoT devices or the backend server:
1.APIs: Develop RESTful or WebSocket APIs on the backend server to expose data and functionality to the app.
2.Mobile App Client: Implement communication within the app using libraries like fetch (for HTTP requests), WebSockets, or specialized IoT communication protocols (e.g., MQTT).

6.Authentication and Security:

Implement user authentication mechanisms to ensure secure access to the app.
Ensure data security by using encryption and authentication methods, especially when dealing with sensitive data.

7.Real-Time Data Retrieval:

Enable the app to request and display real-time data from the IoT devices, such as vehicle location, passenger count, and alerts.

8.User-Friendly Interfaces:

Create user-friendly interfaces within the app to display real-time information and allow users to interact with the Public Transport Optimization system.

9.Push Notifications:

Implement push notification services to send real-time alerts and updates to the mobile app users. This could be for service delays, route changes, or other relevant information.

10.Testing:

Thoroughly test the app's functionality, performance, and user experience to ensure it works seamlessly with the IoT system.

11.Deployment:

Deploy the mobile app to app stores (e.g., Apple App Store, Google Play Store) for public or limited access.

12.Maintenance and Updates:

Continuously monitor the app's performance and user feedback. Address issues, release updates, and add new features as needed.

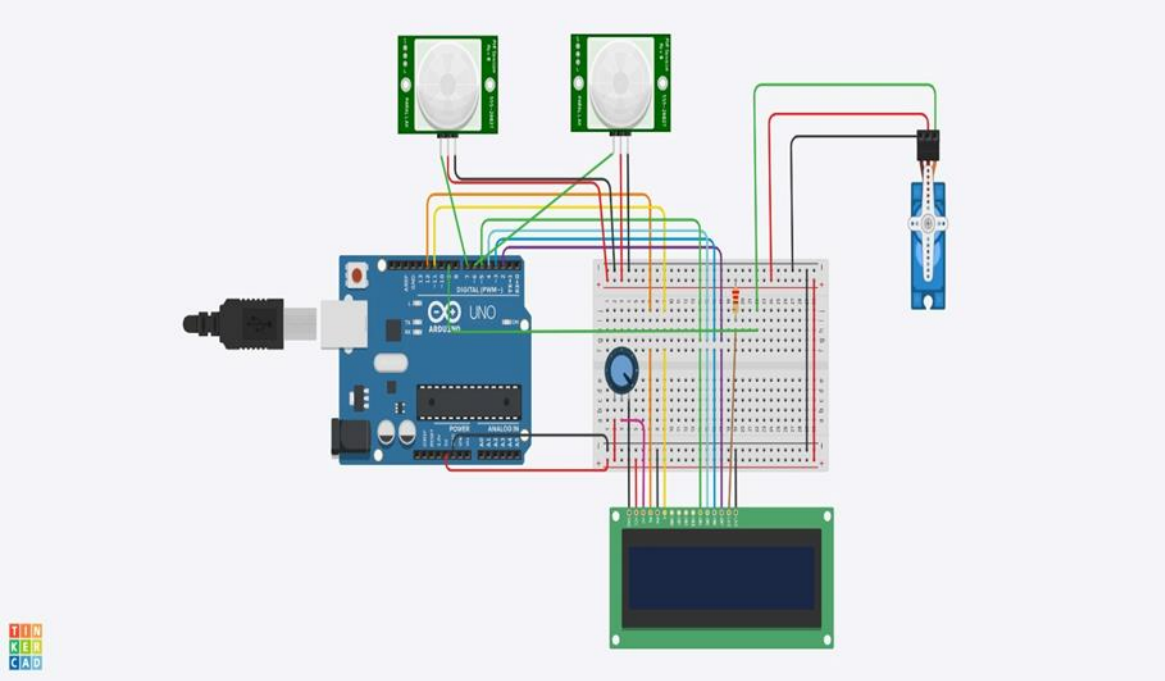
PROGRAM:

```
 import 'package:flutter/material.dart';
import 'package:http/http.dart' as http;
import 'dart:convert';
void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  @override
Widget build(BuildContext context) {
 return MaterialApp(
     home: VehicleLocations(),
   );
  }
}
class VehicleLocations extends StatefulWidget {
  @override
  _VehicleLocationsState createState() =>
_VehicleLocationsState();
}
class _VehicleLocationsState extends State<VehicleLocations>
{
  String locationData = "";
  Future<void> fetchVehicleLocations() async {
   final response = await http.get('http://your-python-server-
url/get_vehicle_location?vehicle_id=bus1');
    if (response.statusCode == 200) {
      setState(() {
        locationData = json.decode(response.body).toString();
      });
    }
  }
  @override
  Widget build(BuildContext context) {
   return Scaffold(
```

```
    appBar: AppBar(
      title: Text('Public Transport Optimization App'),
    ),
    body: Center(
      child: Column(
        children: <Widget>[
          ElevatedButton(
            onPressed: fetchVehicleLocations,
            child: Text('Get Vehicle Location'),
          ),
          Text(locationData),
        ],
      ),
    ),
  );
 }
}
```

SIMULATION:



CONCLUSION:

Public transport optimization is essential for reducing traffic congestion, emissions, and improving urban mobility. Effective strategies include route planning, technology integration, and sustainable infrastructure development, ultimately enhancing accessibility and the quality of life in cities.