TRAFFIC MANAGEMENT SYSTEM

Team Members:

1. SOUNDARI V.
2. ABINAYA R.
3. AKSHAYA R.
4. SWETHA K.
5. VASUNDHARA V.

Project Title: Traffic Management System

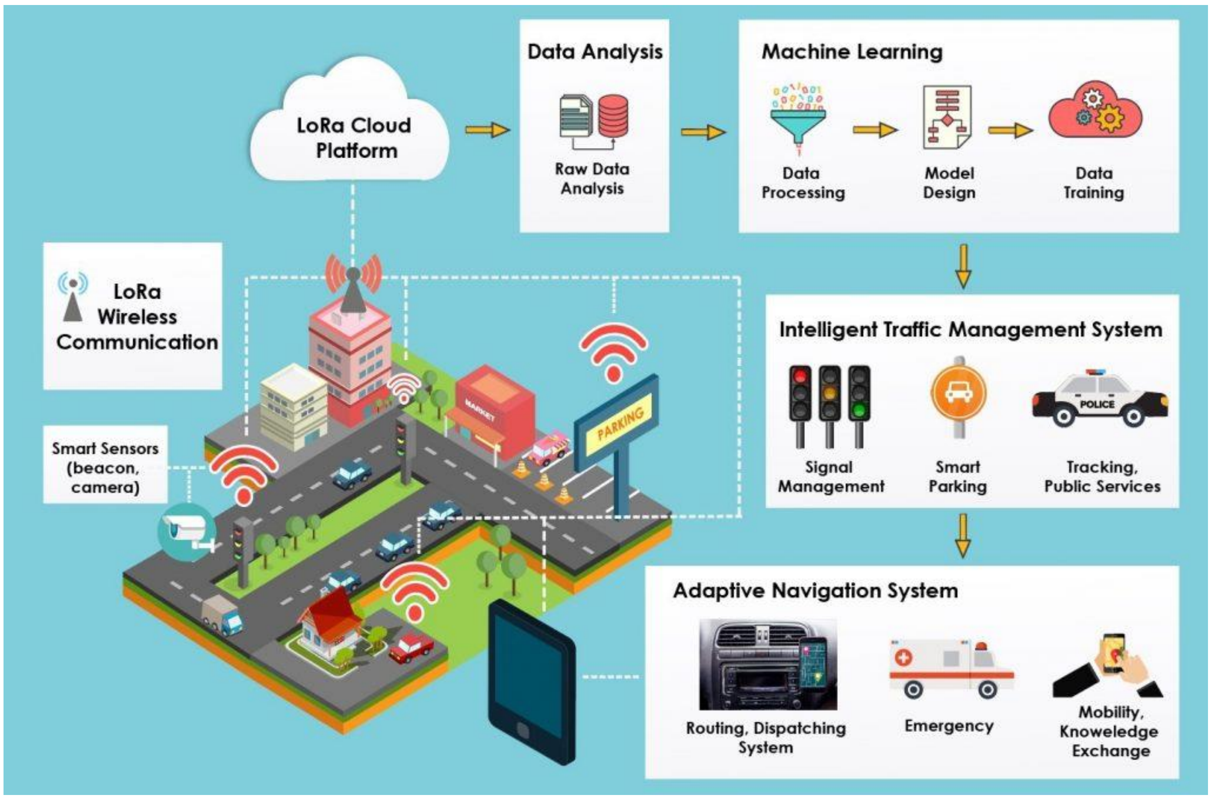
Project Steps

Phase 2: Innovation

Traffic management systems have seen significant innovations and advancements in recent years, driven by the need to improve traffic flow, reduce congestion, enhance safety, and address the challenges of urbanization and population growth. Here are some key innovations in traffic management systems:

1. **Connected Vehicles and V2X (Vehicle-to-Everything):** The integration of smart sensors, IoT devices, and communication technology has enabled vehicles to communicate with each other and with traffic infrastructure. This allows for real-time data sharing, such as traffic conditions, accidents, and road closures, leading to improved traffic management.

2. **Traffic Prediction and Analytics:** Advanced data analytics and machine learning techniques are used to predict traffic patterns and congestion. This information is invaluable for adjusting traffic signals and managing traffic flow to reduce congestion and improve traffic efficiency.


LoRa Cloud Platform
Data Analysis
Raw Data Analysis
Machine Learning
Data Processing
Model Design
Data Training
LoRa Wireless Communication
Smart Sensors (beacon, camera)
Intelligent Traffic Management System
Signal Management
Smart Parking
Tracking, Public Services
Adaptive Navigation System
Routing, Dispatching System
Emergency
Mobility, Knowledge Exchange
MARKET
PARKING
POLICE

3. **Dynamic Traffic Signal Control:** Traditional traffic signals are being replaced by dynamic systems that can adapt to changing traffic conditions. Adaptive traffic signal systems use real-time data to adjust signal timing, reducing wait times and improving traffic flow.

4. **Smart Traffic Management Apps:** Mobile apps that provide real-time traffic updates and navigation assistance have become commonplace. These apps can suggest alternate routes, helping drivers avoid congestion and save time.

5. **Autonomous Vehicles:** As autonomous vehicles become more prevalent, they can communicate with traffic management systems, improving traffic coordination and safety. Self-driving cars can also optimize traffic flow and reduce congestion.

6. **Integrated Public Transportation Systems:** Many cities are integrating their public transportation systems with traffic

management to provide a more seamless experience for commuters. This includes real-time tracking of buses and trains, as well as integrated ticketing systems.

7. **Dynamic Pricing and Congestion Charging:** Some cities have implemented congestion pricing to reduce traffic in congested areas during peak hours. Dynamic pricing adjusts tolls and fees based on traffic conditions to encourage off-peak travel.

8. **Pedestrian and Cyclist Integration:** Traffic management systems are becoming more inclusive, providing safer crossings for pedestrians and dedicated lanes for cyclists. This encourages alternative modes of transportation and reduces reliance on cars.

9. **AI-Powered Cameras and Sensors:** Artificial intelligence is used to analyze data from cameras and sensors placed throughout road networks. This helps in detecting traffic violations, monitoring traffic flow, and even identifying accidents in real-time.

10. **Electric Vehicle (EV) Infrastructure:** The growth of electric vehicles has led to innovations in traffic management to support EV charging infrastructure. Smart charging stations and EV-specific lanes are being integrated into traffic systems.

11. **Environmental Considerations:** Traffic management is increasingly focused on environmental concerns. Cities are implementing low-emission zones and policies that promote eco-friendly transportation options.

12. **Emergency Response Integration:** Traffic management systems are now closely integrated with emergency response services

to quickly clear accident scenes and manage traffic during emergencies.

These innovations in traffic management systems are designed to make transportation more efficient, safer, and environmentally sustainable while addressing the challenges posed by urbanization and increased road traffic. They often rely on a combination of data collection, communication technology, and advanced analytics to achieve these goals.

Coding:

```
import pyfirmata

import time

import pygame


COLOURS = {

    "black": "\u001b[30;1m",

    "red": "\u001b[31;1m",

    "green": "\u001b[32m",

    "yellow": "\u001b[33;1m",

    "blue": "\u001b[34;1m",

    "magenta": "\u001b[35m",

    "cyan": "\u001b[36m",

    "white": "\u001b[37m",

    "reset": "\u001b[0m",

    "yellow-background": "\u001b[43m",

    "black-background": "\u001b[40m",
    "cyan-background": "\u001b[46;1m",
}



def colortext(text):
    for colour in COLOURS:
        text = text.replace("[[" + colour + "]]", COLOURS[colour])
    return text



def ledgreenhorizontal(b):
    if b == 1:
        print(colortext('[[green]]horizontalgreenon[[reset]]'))
    else:
        print('horizontalgreenoff')



def ledyellowhorizontal(b):
    if b == 1:
        print(colortext('[[yellow]]horizontalyellowon[[reset]]'))
    else:
        print('horizontalyellowoff')
```

```
def ledredhorizontal(b):
    if b == 1:
        print(colortext('[[red]]horizontalredon[[reset]]'))
    else:
        print('horizontalredoff')



def ledgreenvertical(b):
    if b == 1:
        print(colortext('[[green]]verticalgreenon[[reset]]'))
    else:
        print('verticalgreenoff')



def ledyellowvertical(b):
    if b == 1:
        print(colortext('[[yellow]]verticalyellownon[[reset]]'))
    else:
        print('verticalyellowoff')



def ledredvertical(b):
    if b == 1:
        print(colortext('[[red]]verticalredon[[reset]]'))
    else:
        print('verticalredoff')

def RR():
    import pygame

    pygame.init()

    white = (255, 255, 255)

    display_surface = pygame.display.set_mode((1000, 800))

    image = pygame.image.load(r'D:\HACKATHON\RRlights.png')

    display_surface.fill(white)

    display_surface.blit(image, (0, 0))

    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            pygame.time.delay(5)
            pygame.quit()

            quit()
        pygame.display.update()



def RG():
    import pygame

    pygame.init()

    white = (255, 255, 255)

    display_surface = pygame.display.set_mode((1000, 800))

    image = pygame.image.load(r'D:\HACKATHON\RGlights.png')

    display_surface.fill(white)

    display_surface.blit(image, (0, 0))

    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            pygame.time.delay(5)
            pygame.quit()
            quit()

        pygame.display.update()


def RY():
    import pygame

    pygame.init()

    white = (255, 255, 255)

    display_surface = pygame.display.set_mode((1000, 800))

    image = pygame.image.load(r'D:\HACKATHON\RYlights.png')

    display_surface.fill(white)

    display_surface.blit(image, (0, 0))

    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            pygame.time.delay(5)
            pygame.quit()
            quit()

        pygame.display.update()



def GG():
    import pygame

    pygame.init()

    white = (255, 255, 255)

    display_surface = pygame.display.set_mode((1000, 800))

    image = pygame.image.load(r'D:\HACKATHON\GG lights.png')

    display_surface.fill(white)

    display_surface.blit(image, (0, 0))

    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            pygame.time.delay(5)
            pygame.quit()

            quit()

        pygame.display.update()


def GR():
    import pygame

    pygame.init()

    white = (255, 255, 255)

    display_surface = pygame.display.set_mode((1000, 800))

    image = pygame.image.load(r'D:\HACKATHON\GR lights.png')

    display_surface.fill(white)
    display_surface.blit(image, (0, 0))
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            pygame.time.delay(5)
            pygame.quit()
            quit()
        pygame.display.update()
```

output: