Code :4212 Reg :421221121016

PUBLIC TRANSPORT OPTIMIZATION

Phase 1: Project Definition and Design Thinking

Project Definition:

Real-Time Information:

• Provide real-time information such as traffic conditions, vehicle locations, estimated time of arrival and service status.

• Real-time updates on traffic flow, incidents, and delays.

• Live tracking of vehicles to inform passengers about their current location.

Arrival Time Prediction:

• Develop an accurate system that predicts the arrival times of public transportation vehicles at stops or stations.

• Integration with mobile apps and displays at stations to showcase predicted arrival times for different transportation modes

Ridership Monitoring:

• Monitor and analyze ridership (number passengers in travel) to optimize transportation services and resources based on demand.

• Notify all the information about ridership to nearby stations.

Enhanced Public Transportation Services:

• Integration of user feedback mechanisms to collect suggestions and complaints.

• Implementation of features like seat availability indicators, priority seating for specific groups, and improved accessibilit

IoT Sensor Design:

• Deploying IoT sensors in public transportation vehicles can help improve safety, efficiency, and passenger compatibility.

Sensor Placement:

• Doorways (Entrance / Exit):

To count and analyze passengers' ridership and notify to the nearby station using IR Sensor, Ultrasonic Sensor, Motion Sensor.

• Seating Areas:

To check how many empty spaces are available in the public vehicles using Compression Load Cells, Vibration Sensor place under the seats.

GPS & RFID:

Provide real-time information about various aspects of the transportation system, such as traffic conditions, vehicle live locations, predicts the arrival times using GPS Devices and RFID Tags placed on roof of the Bus or front of the Bus.

Real-Time Transit Information Platform:

• Integration with Mapping Services:

Incorporate maps and geospatial data into your platform to display transit routes, vehicle positions, and stop locations.

Consider using mapping APIs such as Google Maps or Map box and display in each station.

• Accessibility and Mobile Responsiveness:

Ensure that your platform is accessible to users with disabilities by following web accessibility guidelines (e.g., WCAG).

Optimize the platform for mobile devices to reach a broader passenger

• Connectivity:

Public Vehicles connect to nearby wireless stations (i.e., Base stations, WLAN) and displays details about ridership, seat availability, Bus Routings and arrival Timings to every passenger.

• Integration Approach:

•Data Collecting using IoT sensor:

•It determines how IoT Sensors send data to the real time transit

- Data Aggregation & Processing:

In some cases, IoT sensors may have limited processing capabilities

to aggregate and preprocess data locally.

- Data Transmission:

•Cellular Networks:

Sensors can transmit data over 3G,4G,5G networks. This is a

common choice for sensors on moving vehicles.

- Protocols used in Data Transmission:

•MQTT (Message Queuing Telemetry Transport): MQTT is a

standards-based messaging protocol, or set of rules, used for

machine-to-machine communication. Smart sensors, wearables, and

other Internet of Things (IoT) devices typically must transmit and

receive data over a resource-constrained network with limited

bandwidth.

•CoAP (The Constrained Application Protocol): It is also a client server protocol designed for the Internet of Things. It too targets

constrained devices but is modeled on the World Wide Web of

resources (URIs) and it uses a binary header format, a smaller message size, and a simpler request-response mode.

•Advanced Message Queuing Protocol (AMQP): It is an open source published standard for asynchronous messaging by wire. AMQP enables encrypted and interoperable messaging between organizations and applications. The protocol is used in client/server messaging and in IoT device management.

INTRODUCTION

► Provide real-time information such as traffic conditions , vehicle locations, estimated time of arrival and service status. Live tracking of vehicles to inform passengers about their current location. Develop an accurate systems that predicts that arrival times of public transportation vehicles at stops or stations. notify all the information about ridership to nearby stations. Implementation of features like seat availability indicators, priority seating for specific groups and improved in the accessibility.

IDEALOGY

- The public transportation optimization ,in particular, plays a

vital role in people movement.

- Intelligent public transportation is a fundamental development sector with great potentials for fast digitization as cities across.
- By bringing the best of the IoT, RFID, Cloud Computing, various solutions can be introduced to address many issues faced by the public transportation especially buses.

Components Required

► Arduino Uno microcontroller

► SIM800/900 GPRS Module

► NEO-6M GPS Module

► RFID Reader

► Standard 125KHz RFID card (with unique 32-bit ID)

In the circuit, if you want AC Voltage they you can used to standard 125KHz.

Otherwise you can directly give 5 volt DC supply to the circuit.

The Arduino Uno microcontroller there are connected to SIM800/900 GPRS Module and NEO-6M GPS Module.

They have in RFID Reader will be connected to the

ardunio uno microcontroller.

Project Description:

This project aims to optimize public transportation services through the deployment of IoT devices equipped with various sensors. The collected data will be processed using Python scripts to improve operational efficiency, provide real-time information to passengers, and enhance the overall public transportation experience.

Sensors for Deployment:

To achieve effective public transport optimization, several types of sensors can be deployed in vehicles and at transport infrastructure locations. Here are the key sensors and their applications:

GPS Sensor:

Application: Used for real-time vehicle tracking and route optimization.

Benefits: Provides accurate location data, helping to monitor vehicle movement, calculate ETA, and optimize routes based on traffic conditions.

Passenger Counting Sensors:

Application: Used to monitor passenger loads on vehicles.

Benefits: Allows for optimization of vehicle capacity, leading to better resource allocation and service planning.

Temperature and Climate Sensors:

Application: Monitoring and maintaining comfortable climate conditions inside vehicles.

Benefits: Ensures passenger comfort and safety by regulating heating, ventilation, and air conditioning systems.

Proximity Sensors (Ultrasonic or Infrared):

Application: Detecting the proximity of vehicles to obstacles, objects, or pedestrians.

Benefits: Enhances safety by providing alerts to drivers and helping avoid collisions.

Camera Sensors (CCTV):

Application: Surveillance and monitoring of passengers, driver behavior, and security.

Benefits: Improves safety and security by recording video footage for analysis and incident resolution.

Project Steps:

The project steps remain consistent with the previous outline. Here is how the sensors are integrated into the process:

Project Planning:

Define the scope, objectives, budget, and timeline for the project.

Select IoT Devices with Sensors:

Research and choose IoT devices that include the necessary sensors.

Deployment of IoT Devices:

Install IoT devices with sensors in public transport vehicles and at key infrastructure locations.

Data Collection:

IoT devices with sensors will collect data, including GPS, passenger counts, temperature, proximity, and camera footage.

Data Processing and Storage:

Develop a data processing pipeline to clean and store the collected sensor data.

Python Script Development:

Create Python scripts to analyze and process sensor data, implementing algorithms for optimization, real-time passenger information, and more.

Real-time Passenger Information:

Develop a user interface for passengers to access real-time information based on sensor data.

Optimization Algorithms:

Utilize sensor data to improve route efficiency, minimize delays, and optimize resource allocation.

Visualization and Reporting:

Create data visualization tools and generate reports for operational decision-making, leveraging the sensor data.

Testing and Validation:

Thoroughly test the system, including sensor accuracy and script functionality.

Python Program:

```
import serial
import requests
# Configure the serial port (update the port name accordingly)
ser = serial.Serial('COMX', 9600)
# ThingSpeak settings
thingspeak_api_key = '6EKT0ALDBXGG60Q1'
thingspeak_url = f'https://thingspeak.com/channels/2303456/private_show'
try:
    while True:
        # Read passenger count data from Arduino
        passenger_count = ser.readline().strip().decode('utf-8')
        # Send data to ThingSpeak
        response = requests.get(f'{thingspeak_url}&field1={passenger_count}')
        if response.status_code == 200:
            print(f"Passenger count sent to ThingSpeak: {passenger_count}")
        else:
            print("Failed to send data to ThingSpeak")
except KeyboardInterrupt:
    ser.close()
    print("Connection closed")
```

Project Description:

This project aims to solve the problem of tracking and accountability of vehicles by providing a software platform. This project would serve as an important step to help in Vehicle tracking, component monitoring, vehicle analysis and fleet management. An efficient vehicle tracking system is implemented for monitoring of any equipped vehicle from any location at any time with the help of Global Positioning System (GPS) and Arduino Board which will enable users to locate their vehicles with ease and in a convenient manner.

GPS Sensor:

Application: Used for real-time vehicle tracking and route optimization.

Benefits: Provides accurate location data, helping to monitor vehicle movement, calculate ETA, and optimize routes based on traffic conditions.

Application: This project can be deployed in various fields like elderly and disabled care services, logistic division, emergency services and rescue operation, school bus tracking, and accounting...

1. Real-Time Tracking
   Here, the GPS receiver receives the location data like latitude and longitude of a vehicle and sends them by using an HTTP request to web server. The browser is used to load the PHP web page which contains Google maps to show the location of the vehicle in real time. The web page containing the map directly marks the coordinates, as it arrives, without reloading the page. That means, in real time, we get to see the location of the vehicle.
2. History
   In history, we ask the user to select the date of journey, the names of vehicles which were on a journey on that day are displayed by selecting the vehicle the journey of that vehicle on that date is fetch. The google map with a marker is displayed on the screen. Along with the map, the Time-Speed Graph of journey and details of the journey is displayed.
   In history, the user must select the date of journey and correspondingly the names of vehicles active on that day are displayed. After the vehicle has been selected, corresponding journey details are displayed. Journey details include google maps representing journey, graph showing speed of vehicle at every instance of time and driver details consisting of driver details.
3. Report Generation
   This part includes generation of specific reports for drivers, trucks, and journeys. In this pdf report is generated, which are used by owners for analysis. For the driver's report, driver must be selected which then generates a report showing basic details and journeys done by him. For truck report, truck must be selected which then generates a report showing basic details and journeys done by him. For journey report, date and vehicle need to be selected which then generates a report showing basic journey details, google maps representing complete journey and graph showing speed of vehicle at every instance of time.

4. **Geofencing**

   Geofencing is one of the important features of our project. As the problem statement stated that driver may take a longer route to increase fuel consumption and ask for more money for his long journey or the driver unintentionally took the wrong path, such scenario affects the throughput of the journey. It also affects the performance of vehicles and drivers. Thus, geofencing is applied.

   Geofencing is the virtual boundary to the optimized path from source to destination which is defined at the time of journey creation. It is a radius which the user has to input in meters. Whenever vehicle deviates from its optimized path and the deviation is more than the defined radius then it means it's a geofencing break, which results in a notification popup generated on the screen. This helps the user to know that geofencing is broken by this vehicle.

5. **SMS Module**

   SMS module is an important feature for the user. As the project is for vendors and they are very busy they can't stay in front of the computer and check the current location of the vehicle. In this scenario, the SMS module is useful to know the current location of vehicle. To know the current location of the vehicle user, you must send an SMS to several of the sims attached to the hardware. In response, the SMS a link is generated when user clicks on that link on the browser it shows a google map with the marker. The marker is the current location of the vehicle at that time. Thus, the SMS module helps the user to check the location of vehicle anytime he wants.

6. **Agile Hardware**

   Hardware is reliable, low cost and compact. As we stated that this project has various applications in various areas so for those applications' hardware will be fixed only, we must change the web application and the method to use the functionality. The functionality and construct of this hardware is fixed and working for this.

Languages: C, PHP, JavaScript, Ajax, Bootstrap, HTML, CSS

Hardware: Arduino UNO, SIM808, GPS antenna

DEVELOPING THE REAL-TIME TRANSIT INFORMATION PLATFORM BY USING WEB TECHNOLOGY

```
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
    <title>Real-Time Transit Information</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>
    <header>
        <h1>Real-Time Transit Information</h1>
    </header>
    <main>
        <div id="map"></div>
        <div id="schedule">
            <h2>Upcoming Buses</h2>
            <ul id="bus-schedule">
                <!-- Real-time bus schedule data will be displayed here -->
            </ul>
        </div>
    </main>
    <script src="script.js"></script>
</body>
</html>
```

CSS

```
body {
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 0;
}
Header
 {
    background-color: #333;
    color: #fff;
```

```
    text-align: center;
    padding: 20px;
}
Main
 {
    display: flex;
    justify-content: space-between;
    padding: 20px;
}
{
    flex: 1;
    height: 400px;
}
{
    flex: 1;
    padding: 10px;
}

#bus-schedule {
    list-style-type: none;
    padding: 0;
}
{
    margin: 5px 0;
    border: 1px solid #ccc;
    padding: 10px;
}
```

Java script

```
document.addEventListener("DOMContentLoaded", () => {
    const busSchedule = [
        { route: "Bus 101", destination: "Downtown", nextArrival: "10:15 AM" },
        { route: "Bus 202", destination: "Suburbia", nextArrival: "10:30 AM" },
        { route: "Bus 303", destination: "Airport", nextArrival: "10:45 AM" },
    ];
    const busScheduleList = document.getElementById("bus-schedule");
    busSchedule.forEach((bus) => {
        const listItem = document.createElement("li");
                listItem.innerHTML = `<strong>${bus.route}</strong> to ${bus.destination}<br>Next arrival:
${bus.nextArrival}`;
        busScheduleList.appendChild(listItem);
    });
});
```

DESIGN THE PLATFORM TO RECEIVE AND DISPLAY REAL-TIME DATA FROM IOT SENSORS: