**IOT\_PHASE5:** **PUBLIC TRANSPORT OPTIMIZATION**

**DOCUMENTATION AND SUBMISSION**

**TEAM MEMBER: BALAMURUGAN S**(822721106008)

**Project Title:** Public Transport Optimization

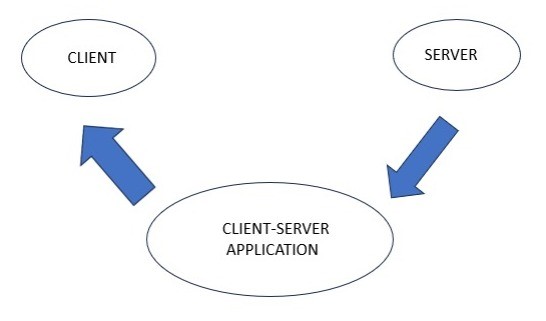
**Project Description:** This project involves integrating IOT Sensors into public transportation vehicles to monitor ridership, track locations, and predict perfect arrival times. The goal is to provide real -time transit information to the public through a public platform, enhancing the efficiency and quality of public transportation services. This project includes defining Objectives, designing the IOT sensor system, developing the real-time transit information platform, and integrating them using IOT technology and python.

**Project Objectives:** This project includes defining Objectives, designing the IOT sensor system, developing the real-time transit information platform, and integrating them using IOT technology and python.

**Design Thinking:**

* Designing the IOT sensor system
* Real-time transit information platform

**FLOW DIAGRAM:**



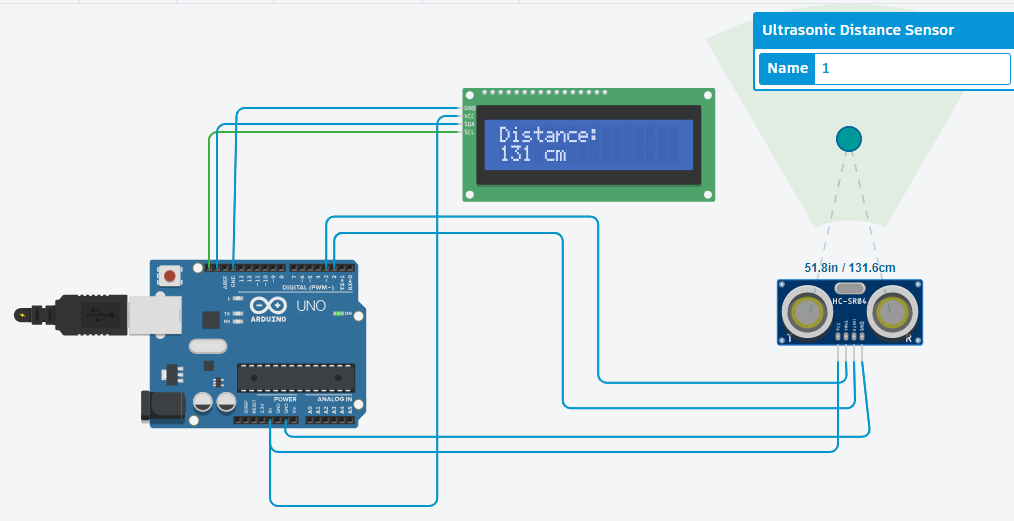
**Client**:

The client refers to the end-user or device, such as a smartphone or computer, that requests and receives information about the real-time location of buses. Clients typically use a mobile app or a web interface to view the bus's location, estimated arrival times, and other relevant data.

**Server**:

The server is a centralized computer or system responsible for collecting and storing the location data from the buses. It processes client requests, retrieves the bus location data, and delivers it to the clients. The server also manages the communication and data exchange between the buses and the clients, ensuring accurate and up-to-date information is provided to users.

**DESIGNING THE IOT SENSOR SYSTEM:**

**ARDUINO UNO:**

Arduino Uno's flexibility and ease of use make it a suitable choice for implementing various aspects of a location tracking system for buses, especially when cost-efficiency and customization are important considerations.

**GPS Data Receiver**: The Arduino Uno can be equipped with a GPS module to receive location data from the bus. It collects information such as latitude and longitude which can be sent to a central server for real-time tracking.

**Data Processing:** Arduino Uno can process the incoming GPS data and prepare it for transmission. It can format the data, calculate distances, or perform other relevant calculations to enhance the accuracy and efficiency of the tracking system.

**Data Transmission:** Arduino Uno can transmit the processed GPS data to a central server through various communication methods, such as GSM, Wi-Fi, or LoRa. This allows the server to update the bus's location in real-time on a map, which can be accessed by passengers via a mobile app or web interface.

**ULTRASONIC SENSOR:**

Ultrasonic sensors can be used in location tracking for buses to perform several important functions;

**Distance Measurement:** Ultrasonic sensors emit high-frequency sound waves, and they measure the time it takes for these waves to bounce off an object and return. By calculating the distance to objects in the bus's vicinity, these sensors provide real-time data on the bus's proximity to other vehicles, obstacles, or landmarks.

**Obstacle Avoidance:** Ultrasonic sensors can help detect obstacles in the bus's path, such as other vehicles or pedestrians. This information is critical for collision avoidance and ensuring passenger safety.

**LCD:**

Liquid Crystal Displays (LCDs) play a different role in location tracking for buses compared to sensors like GPS or ultrasonic sensors. LCDs are primarily used for displaying information to bus drivers and passengers rather than directly contributing to location tracking.

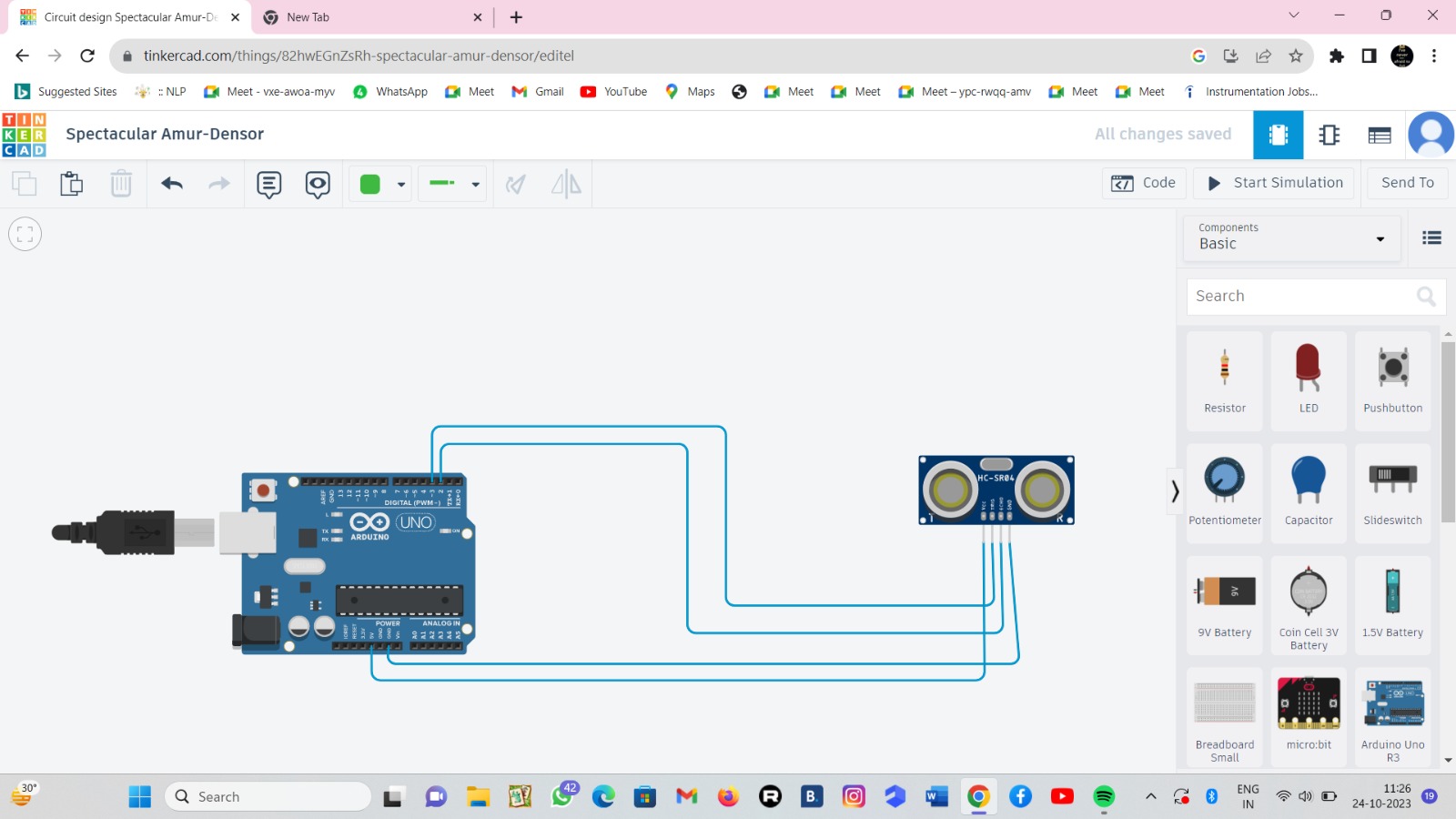
**SOFTWARE REQUIREMENT:**

* TinkerCad used for Simulation Process

**CONNECTION DETAILS:**

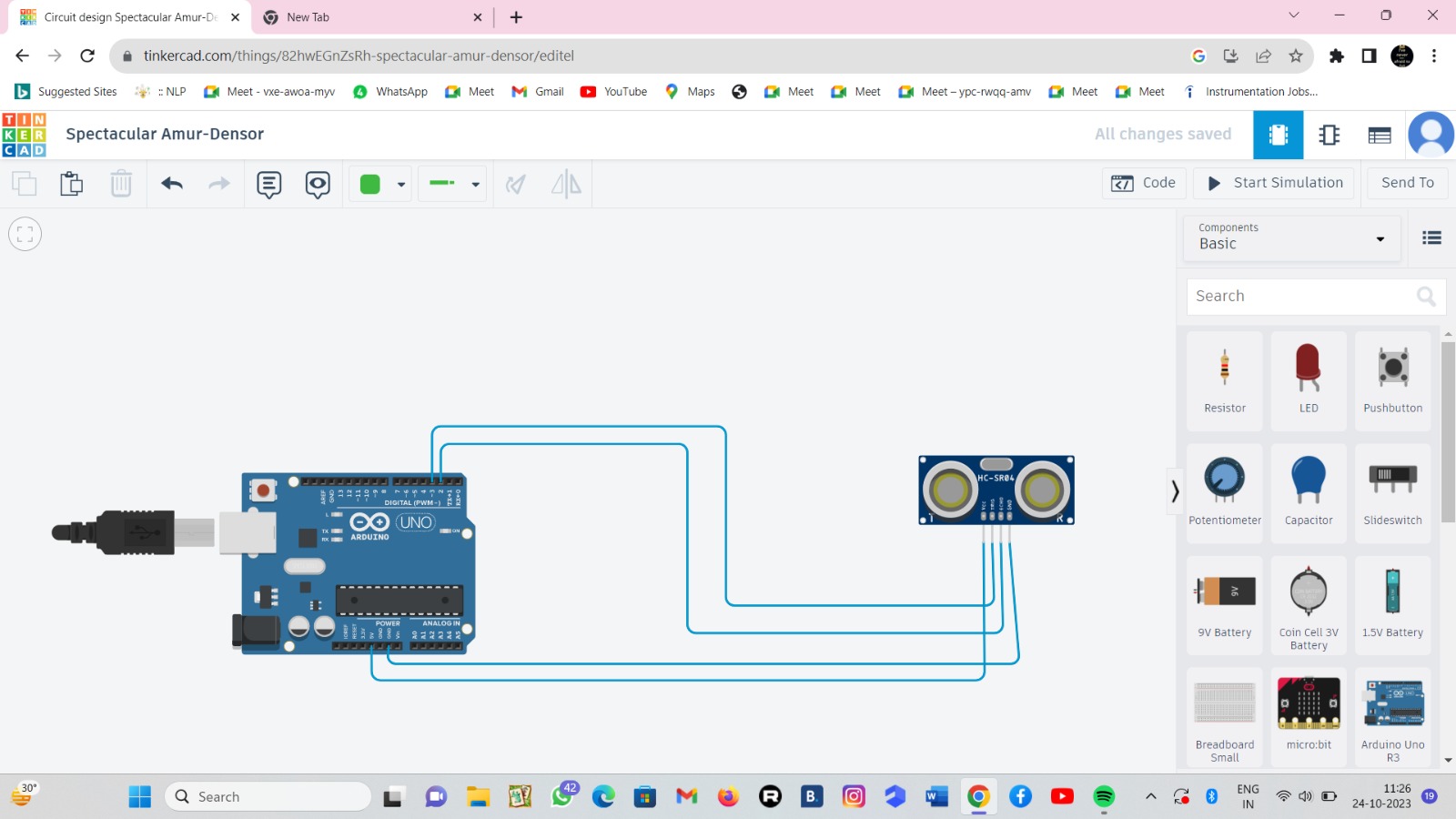
**BETWEEN ARDUINO UNO & ULTRASONIC SENSOR:**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **ARDUINO UNO** | **ULTRASONIC SENSOR** |
| 1 | 5V | VCC |
| 2 | DIGITAL PIN 3 | TRIGGER |
| 3 | DIGITAL PIN 2 | ECHO PIN |
| 4 | GND | GND |

****

**BETWEEN ARDUINO UNO & LED:**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **ARDUINO UNO** | **LCD** |
| 1 | 5V | VCC |
| 2 | FDA PIN | SBA PIN |
| 3 | SCL PIN | SCL PIN |
| 4 | GND | GND |

****

**C++ CODE FORSIMULATION PROCESS:**

#include <LiquidCrystal\_I2C.h>

LiquidCrystal\_I2C lcd(0x20, 16, 2); // Format - > (Address,Width,Height )

#define echoPin 2 // attach pin D2 Arduino to Echo pin of Sensor module

#define trigPin 3 // attach pin D3 Arduino to Trig pin of Sensor module

long duration; // Declare variable to store echo time duration

int distance; // Declare variable to store the result (distance)

void setup()

{

lcd.init(); // initialize the lcd

lcd.backlight(); // Turn on the Backlight

pinMode(trigPin,OUTPUT); // Sets the trigPin as an OUTPUT

pinMode(echoPin, INPUT); // Sets the echoPin as an INPUT

// Serial Communication is starting with 9600 of baudrate speed

Serial.begin(9600);

// The text to be printed in serial monitor Serial.println("Distance measurement using Arduino Uno");

delay(500);

}

void loop()

{

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

duration = pulseIn(echoPin, HIGH);

distance = duration \* 0.0344 / 2;

Serial.print("Distance: ");

Serial.print(distance);

Serial.println(" cm");

lcd.clear(); // Clear the display buffer

lcd.setCursor(0, 0); // Set cursor for "Distance:" (Column, Row)

lcd.print("Distance:"); // print "Distance:" at (0, 0)

lcd.setCursor(0,1); // Set cursor for output value (0, 1)

lcd.print(distance); // print Output in cm at (0, 1)

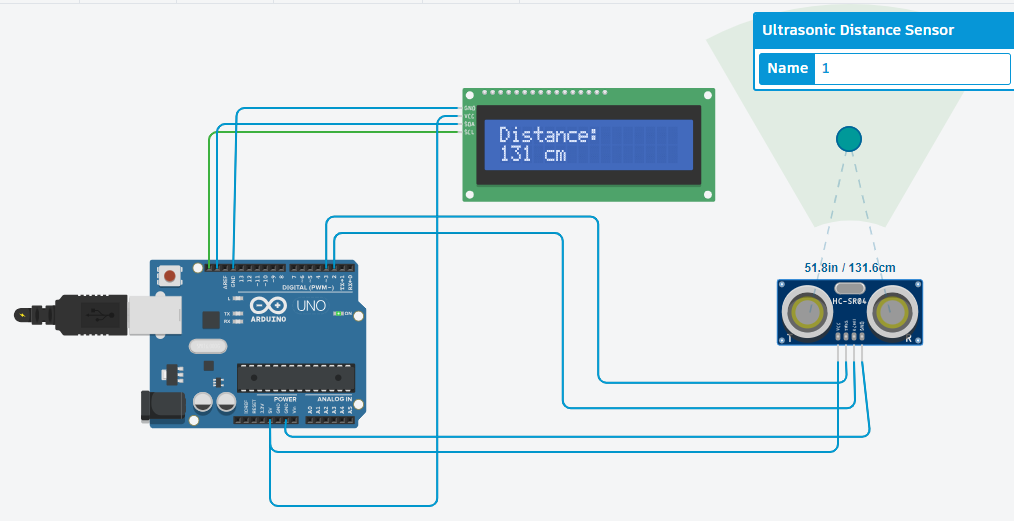
lcd.setCursor(4, 1); // move cursor to (4, 1)

lcd.print("cm"); // print "cm" at (4, 1)

delay(100);

}

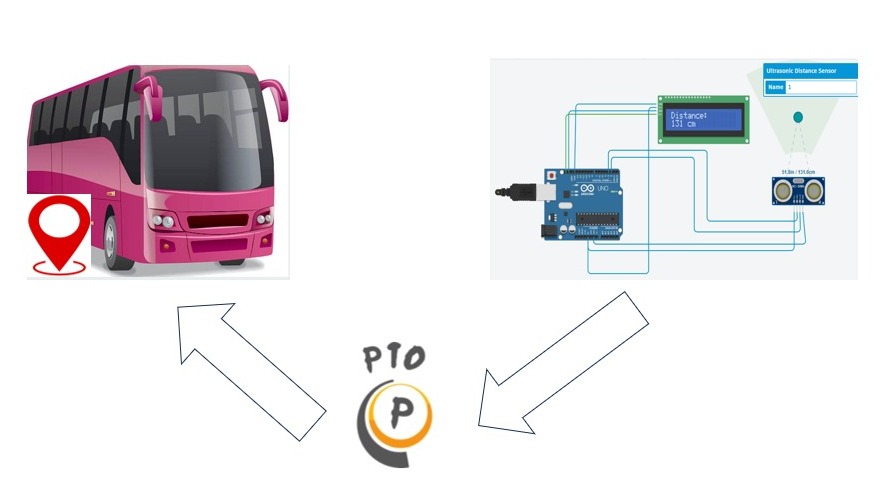
**AFTER SIMULATION:**



**REAL-TIME TRANSIT INFORMATION PLATFORM:**

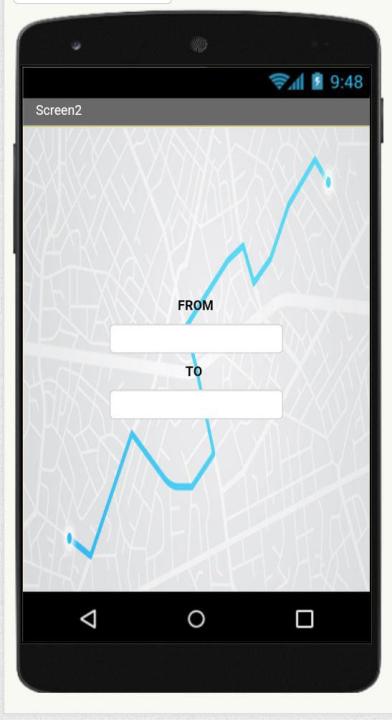
 Location tracking for a bus using an app refers to the capability of monitoring and displaying the real-time geographical position of a bus through a mobile application. This technology allows users to track a bus's current location, estimated arrival times, and other relevant information, enhancing public transportation efficiency and providing passengers with the ability to plan their journeys more effectively.

**WORKFLOW OF APPLICATION:**



**MOTIVE FOR AN PTO APPLICATION:**

The primary motive for a location tracking app for buses is to enhance the efficiency, safety, and convenience of public transportation.

While bus location tracking apps offer numerous advantages in terms of improving public transportation systems, addressing the potential disadvantages is crucial to ensure widespread adoption and passenger satisfaction.

**MERITS:**

**Better Route Planning:** The app provides information on bus schedules, estimated arrival times, and alternative routes, helping passengers plan their journeys more effectively.

**Improved Passenger Experience:** Passengers can easily track the real-time location of buses, reducing uncertainty and wait times, leading to a more convenient and efficient public transportation experience.

**Enhanced Safety:** Real-time tracking enables quicker response to emergencies or accidents, ensuring the safety of passengers and bus drivers.

**DEMERITS:**

**Accessibility:** Not all passengers may have access to smartphones or the ability to use such apps, potentially creating a digital divide.

**Implementation Costs:** Developing and maintaining a bus location tracking app and associated infrastructure can be costly for transit agencies.

**Technical Issues:** Connectivity problems or software glitches can disrupt the accuracy of location tracking, leading to potential passenger frustration.

**HOW IS GOING TO BE WORK?**



