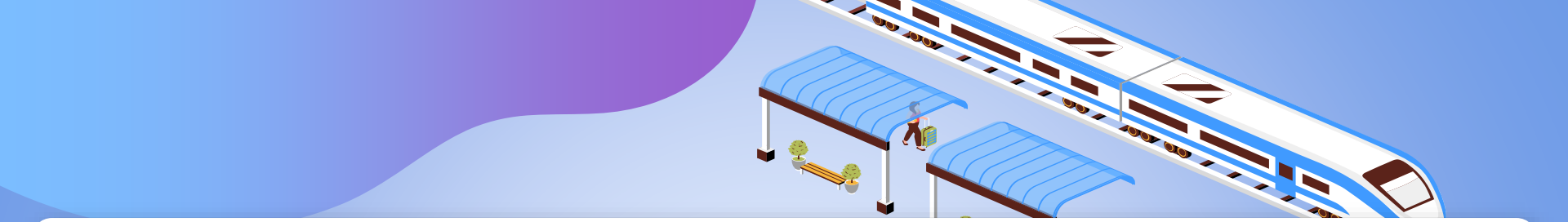


TRAIN DETECTION & ALERT SYSTEM

TMP-23-302

TEAM MEMBERS

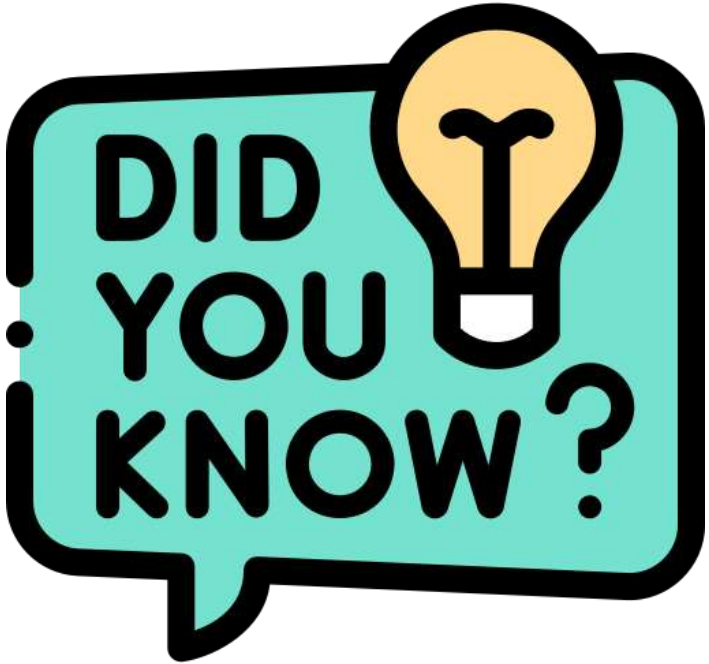
Biyanwila B.D.V.J
IT20212490
Data Science

Jayanga B.M.C
IT20188672
Data Science

Amarasinghe C.D
IT20187064
Cyber Security

Wijewardene L.L
IT20101824
Information Technology

INTRODUCTION



- In Sri Lanka ,

- Every 3 hours, A person dies in Road Accident.
- Every 3 days, a Child dies in Road Accident. [3]

- Population : Mobile Phone ratio is 1.4 per person [4]

INTRODUCTION

WHY WE CHOOSE THIS TOPIC ?

- So many railway-crossing collisions happen in Sri Lanka annually.
- These railway-crossing collisions cost many loss of lives and property damages annually.

[1]

Polgahawela level crossing accident was a collision between a bus travelling from Galkiriyagama to Colombo and a train at a level crossing in Yangalmodara, near Polgahawela in Kurunegala district on 27 April 2005 at 8.30 local time, which resulted in the death of 41 people

[2]

INTRODUCTION

- Citizens have to face collisions at the railway-crossings due to many reasons, unfortunately.

1. **Unsafety Railway Crossings** – Many unsafety railway crossings all around Sri Lanka.

2. **Poor Visibility** – This can happen due to heavy rain, and fog conditions.

3. **Human Errors** - Citizens may misjudge the speed or distance of an approaching train, leading to accidents or near-misses.

4. **Lack of Awareness** - Citizens may not be aware of the dangers posed by railway crossings or the proper safety procedures to follow when crossing tracks.

INTRODUCTION

A technology-based solution called the Train Tracking and Detection System for Citizens via the Sim or a Tracker enables people to follow the movement and location of trains in real time. Both a separate tracker device and a mobile phone with a sim card can be used to access this system. For the most recent information on the whereabouts and status of trains, the system makes use of GPS and other real-time data sources. Passengers can make better travel plans with this information, decreasing the likelihood of missing trains or having a long wait. In order to increase the general effectiveness and safety of railway operations, train operators can also utilize this technology to monitor the performance of their trains.

OBJECTIVE

MAIN OBJECTIVE

PROVIDE AN IT-BASED SOLUTION TO ADDRESS THE SAFETY OF THE CITIZEN IN SRI LANKA FROM THE RAILWAY – CROSSINGS COLLISIONS

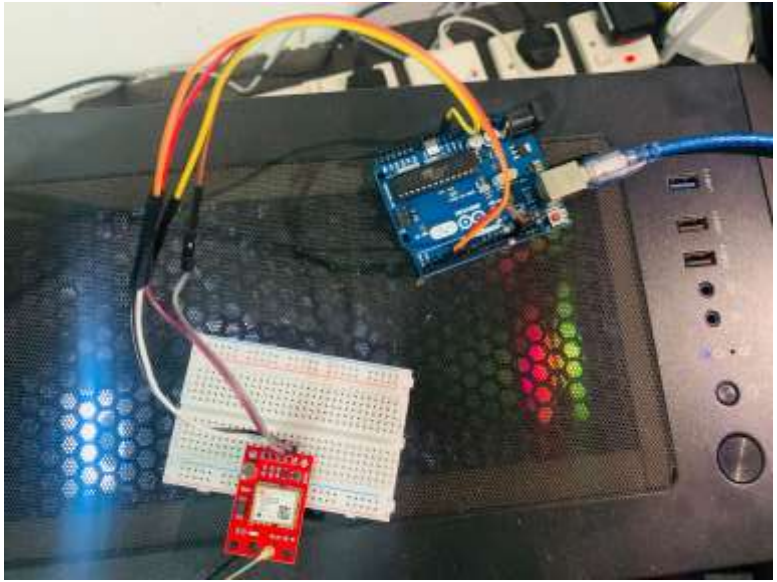
SUB OBJECTIVE

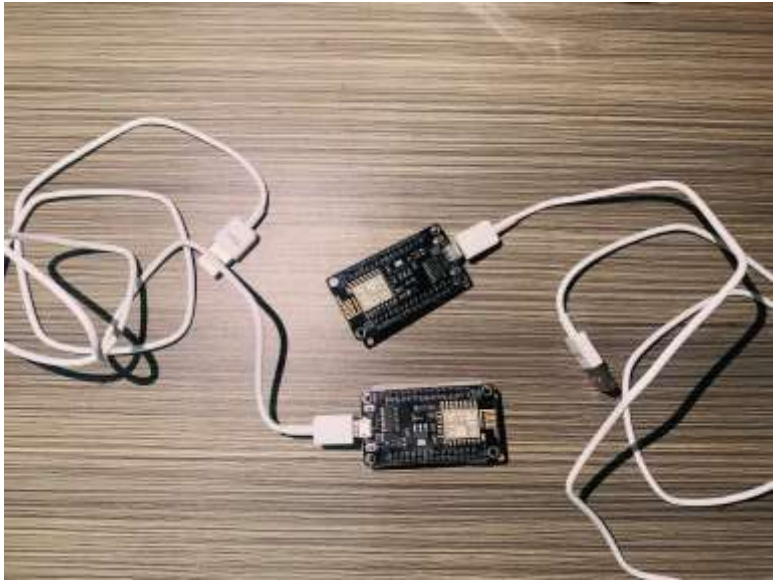
- Develop an User–friendly application that provide real-time data of trains and send alerts to the user properly.
- Gather and analyze data to train models.
- Provide the security and the privacy to the application as appropriate.
- To evaluate the effectiveness and reliability of the developed system through testing and validation.

METHODOLOGY

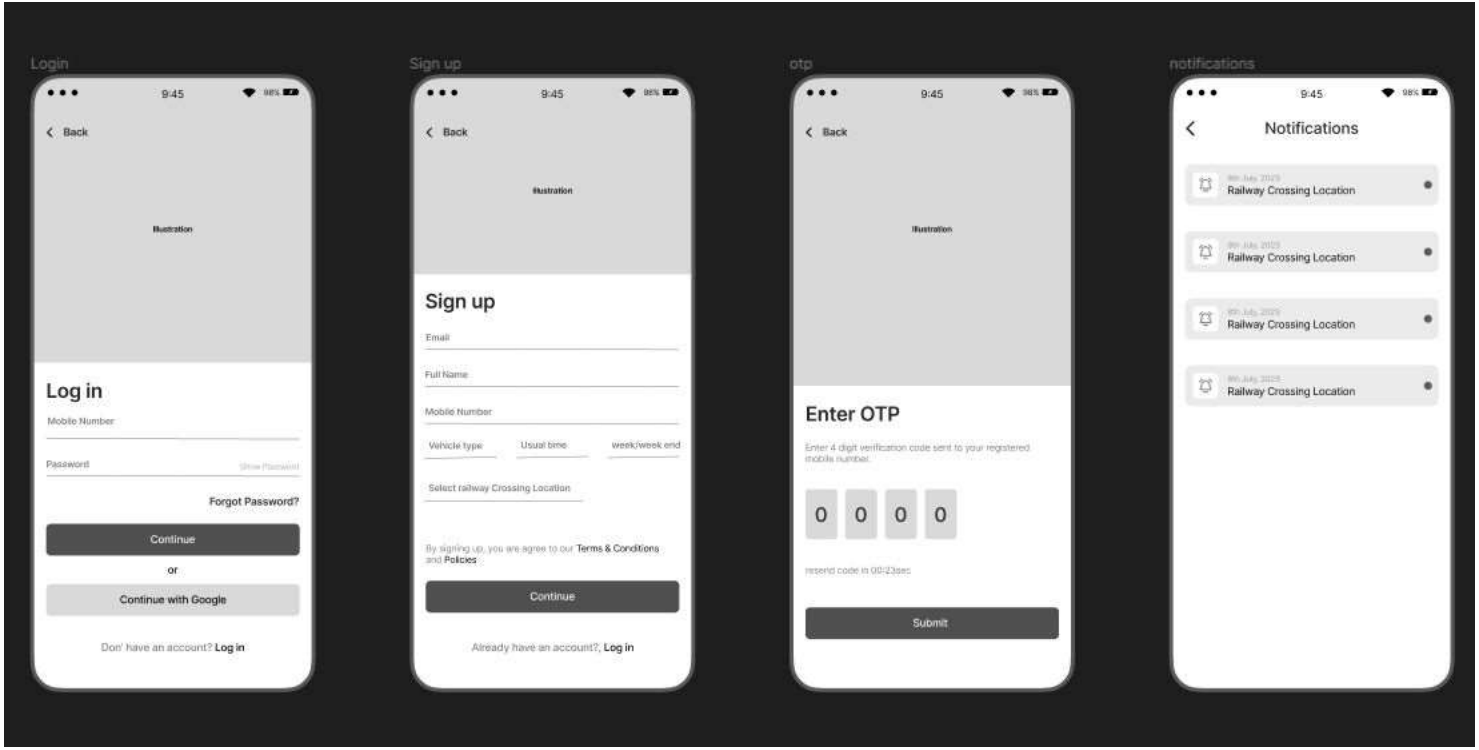
1. Data Collecting
02. Model Building
03. Model Testing
04. UI Designing
05. Security Designing
06. Security Testing
07. Integrating
08. Testing
09. Deployment of the Prototype

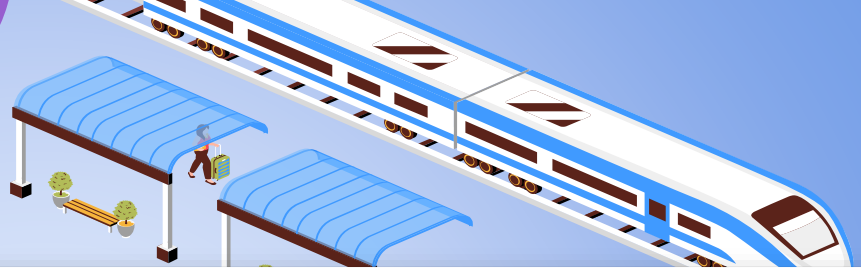
USED DEVICES





WIREFRAMES for MOBILE APPLICATION





To develop a system that utilizes GSM trackers on trains and IoT devices at railway crossings to predict and alert potential blind spots on the train.

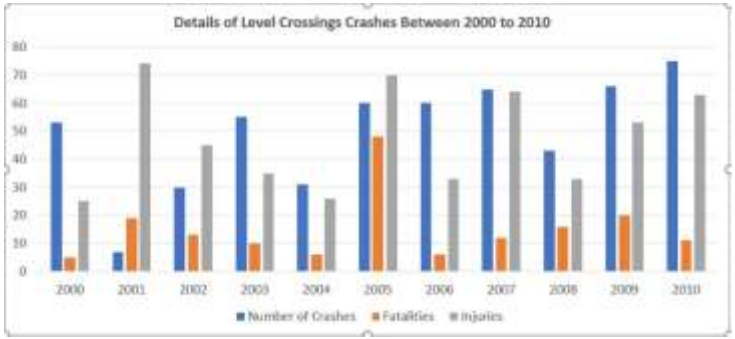
Biyanwila B.D.V.J
IT20212490
Data Science

RESEARCH PROBLEM

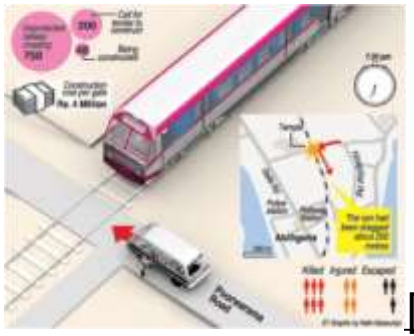
Predict when to send the alert to the IOT device

- So many railway-crossing collisions annually happen in Sri Lanka.
- Lack of accurate and efficient to determine the impact of real-time alerts on passenger safety and satisfaction

[1]



[2]



[3]

OBJECTIVE

MAIN OBJECTIVE

- To develop a system that utilizes GSM trackers on trains and IoT devices at railway crossings to predict and alert potential blind spots on the train.

SUB OBJECTIVE

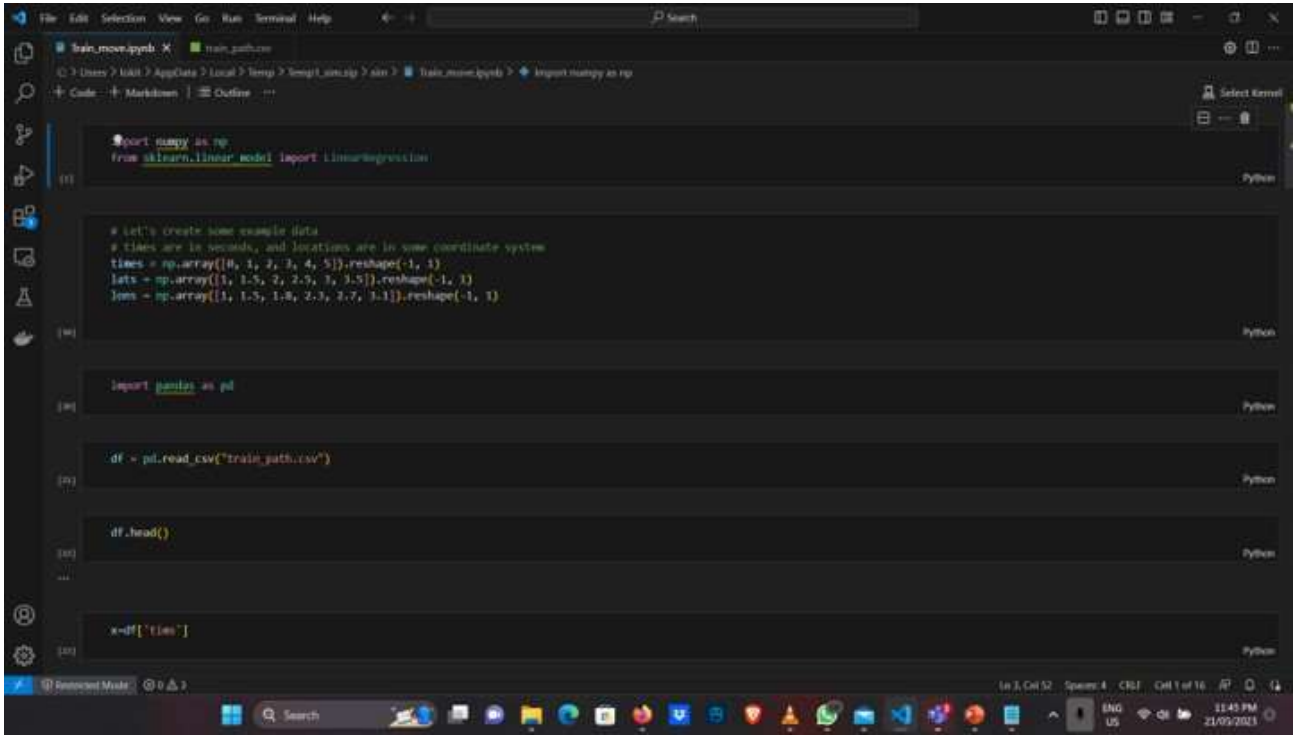
- To develop a GSM tracker that can transmit its location to the IoT device on the railway crossing.
- To enhance the safety measures at railway crossings by developing an IoT device that can detect approaching trains and send an alert to the nearby devices.
- To integrate the IoT device and GSM tracker to establish a communication link to send an alert when a train approaches the crossing. To investigate the feasibility and effectiveness of using manual training of datasets to predict the location of a lost GSM tracker signal in the railway industry.
- To evaluate the performance of the integrated system and its impact on improving the safety measures at railway crossings.

CURRENT PROCESS

Gather data from the tracking device

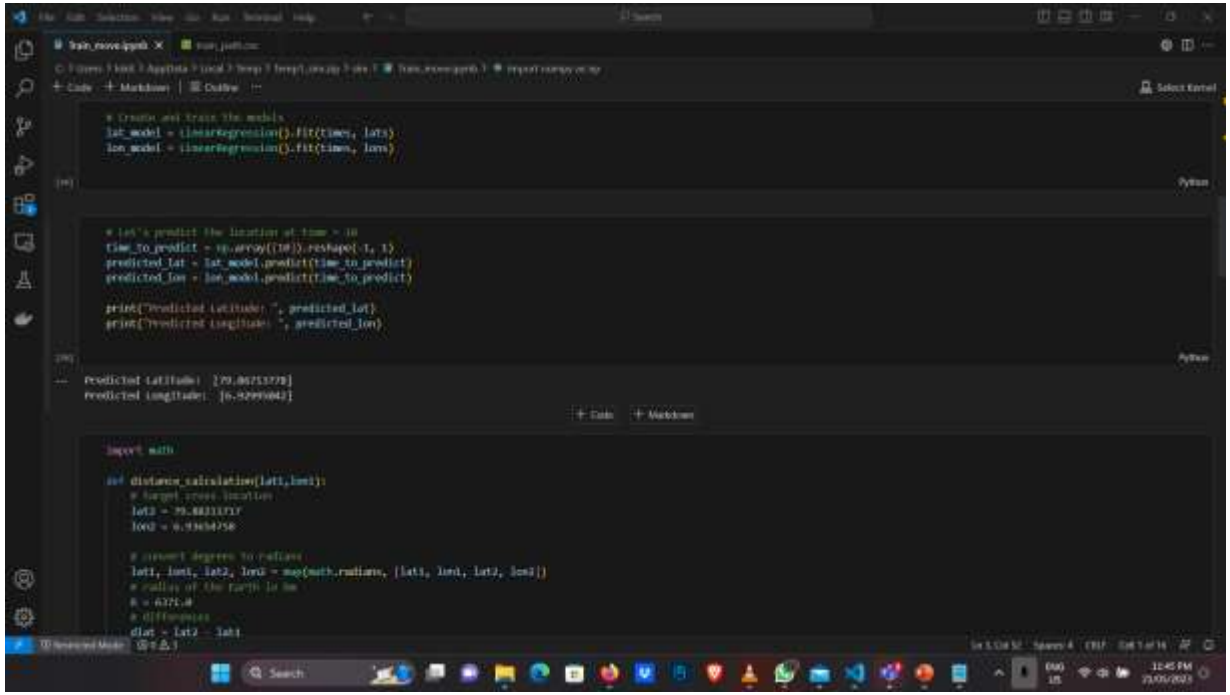
Send Distance for the IOT device on the crossing

EVIDENCE



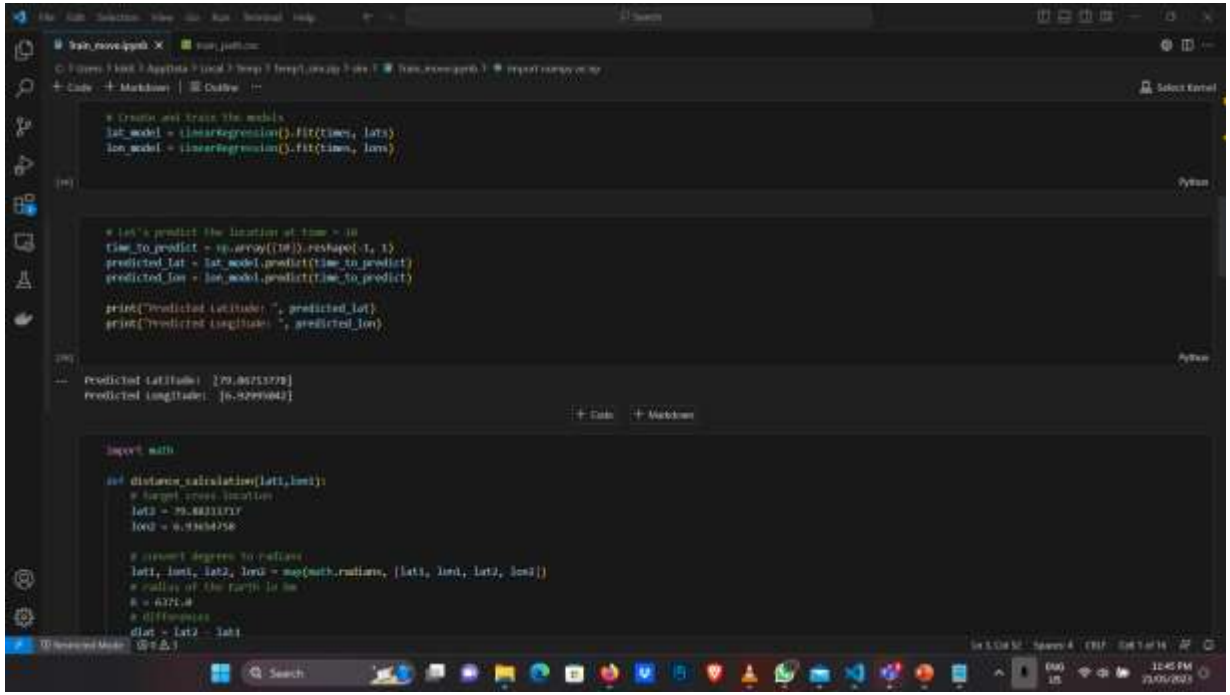
In this screenshot, I imported the Linear Regression library and trained the data.

EVIDENCES



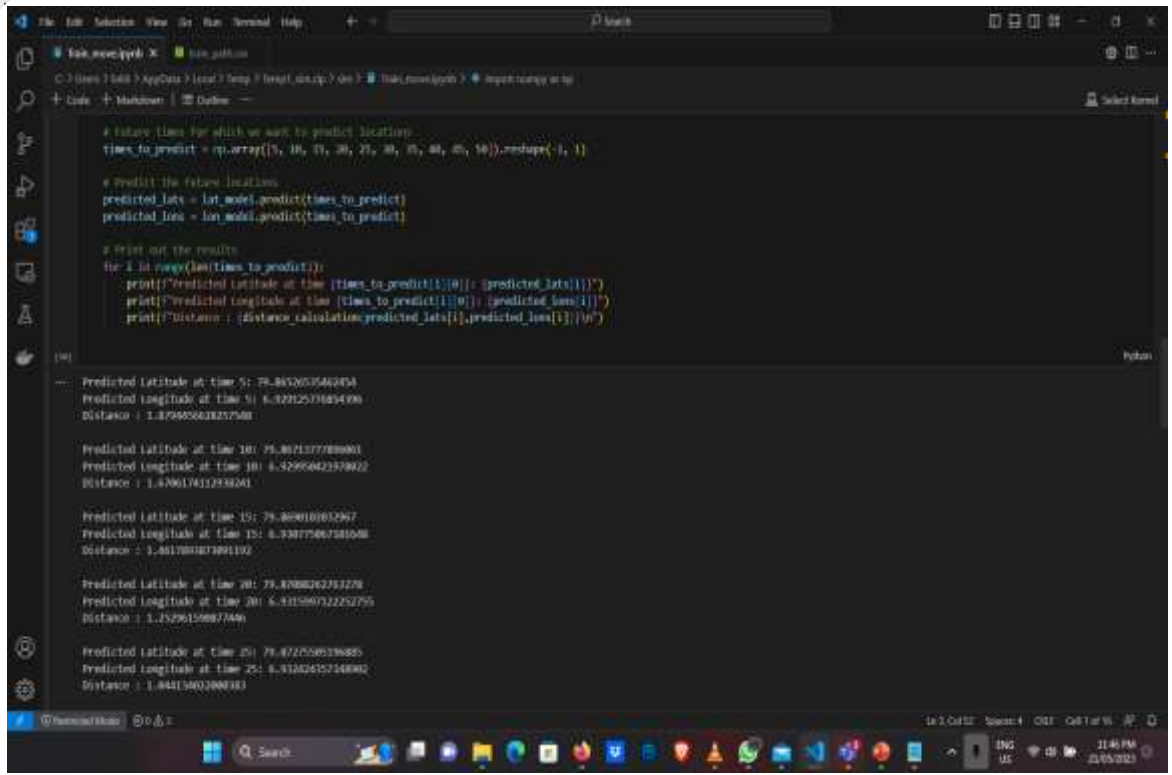
In this step, I retrieve the longitude and latitude coordinates of the train and use a linear regression model to predict its location in the next 10 seconds

EVIDENCE



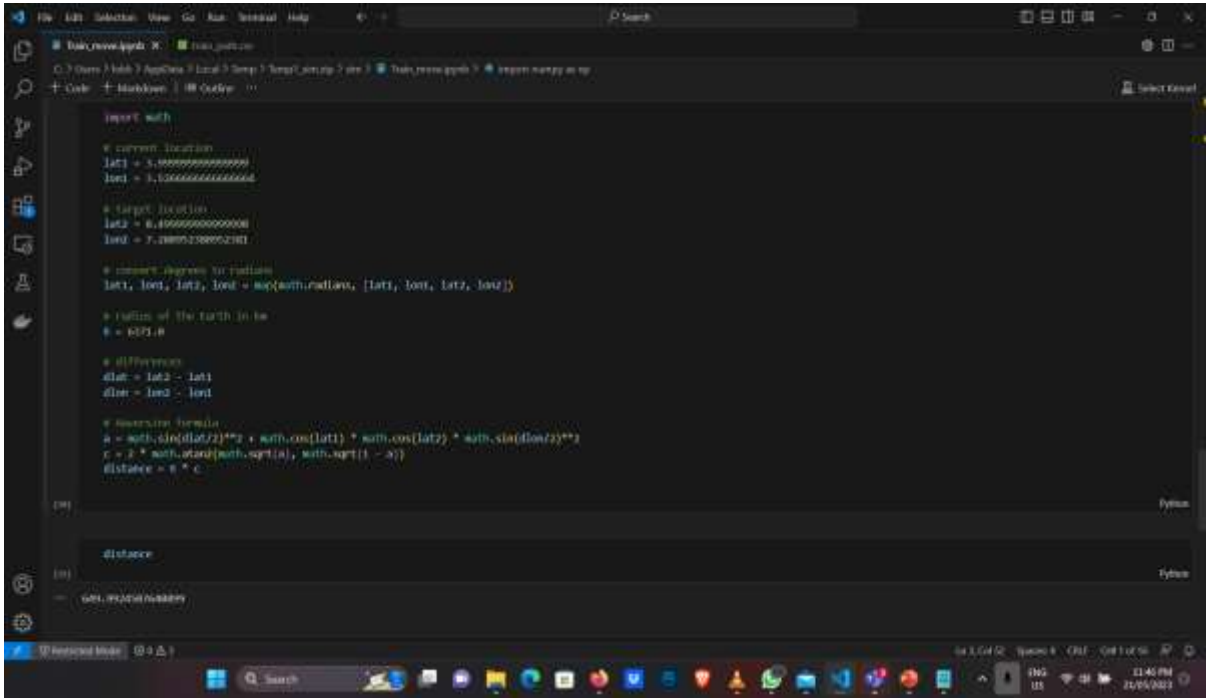
The distance calculations are displayed in this screenshot.

EVIDENCE



This screenshot displays the predicted location of the train at future time stamps.

EVIDENCE



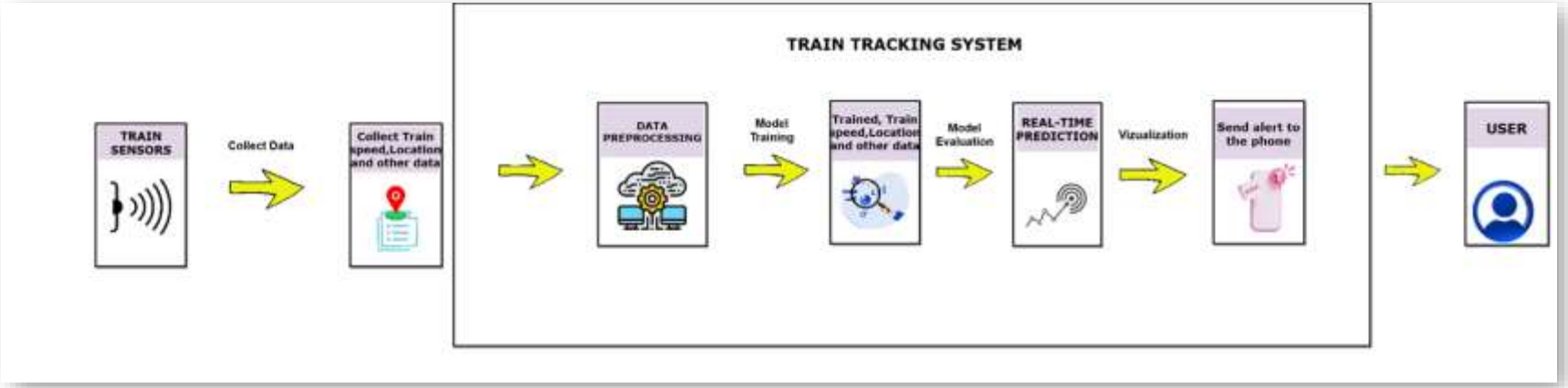
In this step, the difference between the crossing and real-time train location is calculated and the distance difference between these locations is predicted.

TASKS TO BE COMPLETED

When tracking signals lost on the blind spots Predict and send the necessary data to the IOT device on crossing.

To Develop the tracking device and Create Alerting system on the time stamps.

SYSTEM OVERVIEW



TOOLS AND TECHNOLOGIES

Microcontrollers	Arduino uno, NodeMCU
DATABASE	FIREBASE
Trcaking Device	Tracker module (TCRT 5000IR), Sim800 GPS Tracker

Libraries :

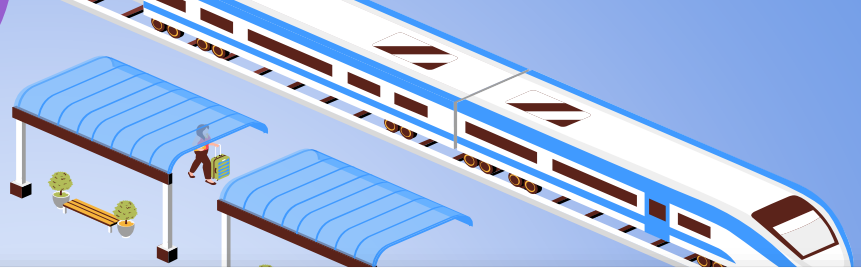
IDE s :

TCRT5000 IR Infrared Reflective Switch Sensor

SIM800L GSM/GPRS Module

REFERENCES

- [1] S. TImes, "sunday times," [Online]. Available: https://www.sundaytimes.lk/110612/News/nws_23.html.
- [2] N. A. Kulasingham Ragulan, "SLSTL," 2021. [Online]. Available: https://slstl.lk/wp-content/uploads/2021/10/A1.3.docx.
- [3] S. Times, "Sunday Times," 2013. [Online]. Available: https://www.sundaytimes.lk/130630/news/railways-750-blood-gates-48-crossings-to-get-gates-by-years-end-50946.html.



Sending alerts to the users via the app & Predicting the likelihood of crossing the railway-crossing

Jayanga B.M.C
IT20188672
Data Science

BACKGROUND

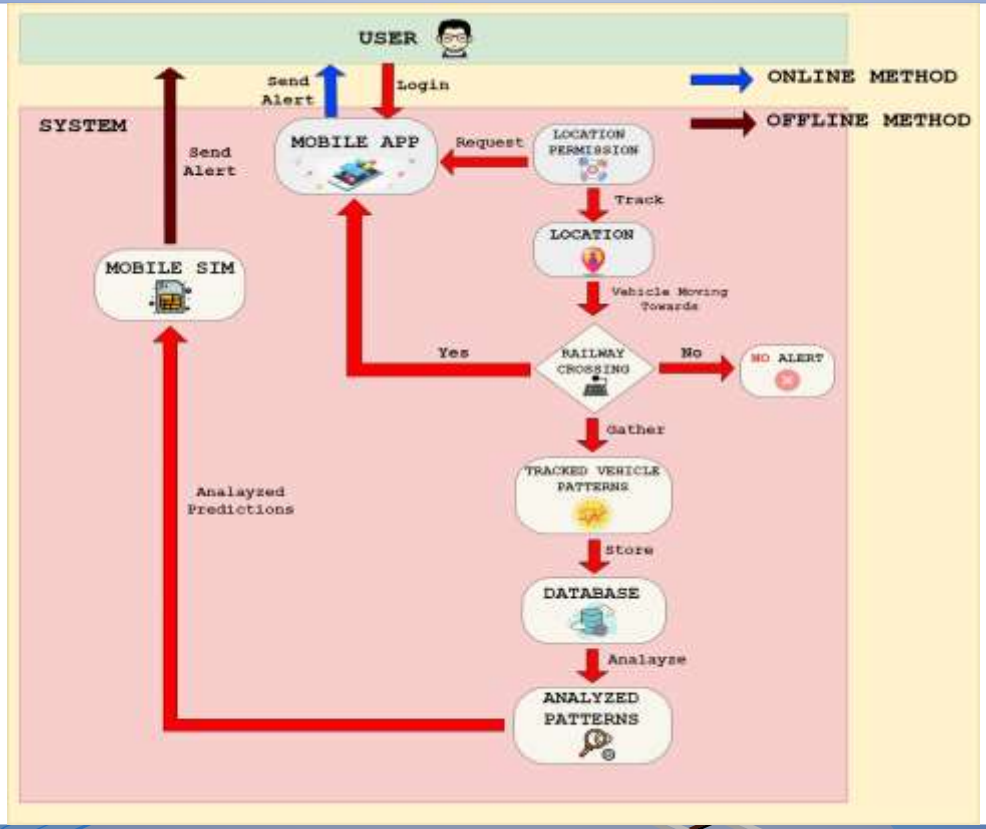
ANALYZE THE PAST PATTERNS OF VEHICLES AND PREDICT IF THEY ARE LIKELY TO CROSS THE RAILWAY CROSSING ON A GIVEN DAY

- Main Objective.
- Below objective should be completed first.
- If the app user crosses the railway crossing, the data will be collected to do the predictions.
- Data will be sent to the cloud-based database.

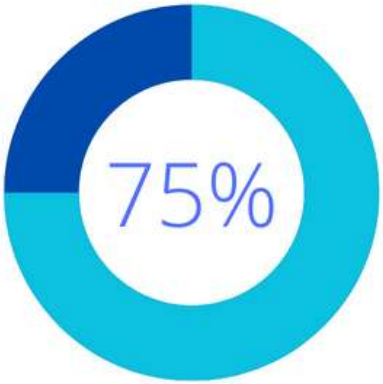
IDENTIFY THE VEHICLES THAT ARE MOVING TOWARDS THE RAILWAY CROSSING, AND SEND ALERTS ONLY TO THEM THROUGH THE APP.

- Secondary Objective.
- Ask the location permission from the user through the app.
- If only the user is moving towards the railway crossing, alert messages are sent.
- Otherwise, alert messages are not sent.

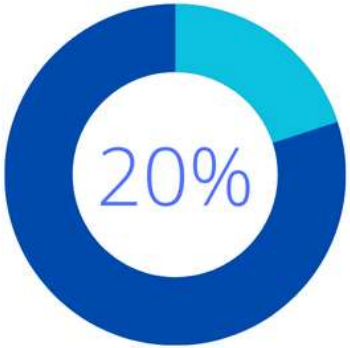
INDIVIDUAL COMPONENT DIAGRAM



CURRENT PROGRESS



Implementation of Identify the vehicles that are moving towards the railway crossing



Implementation of Analyze the Past Patterns of Vehicles and Predict if they are likely to cross the railway crossing on a given day

EVIDENCE

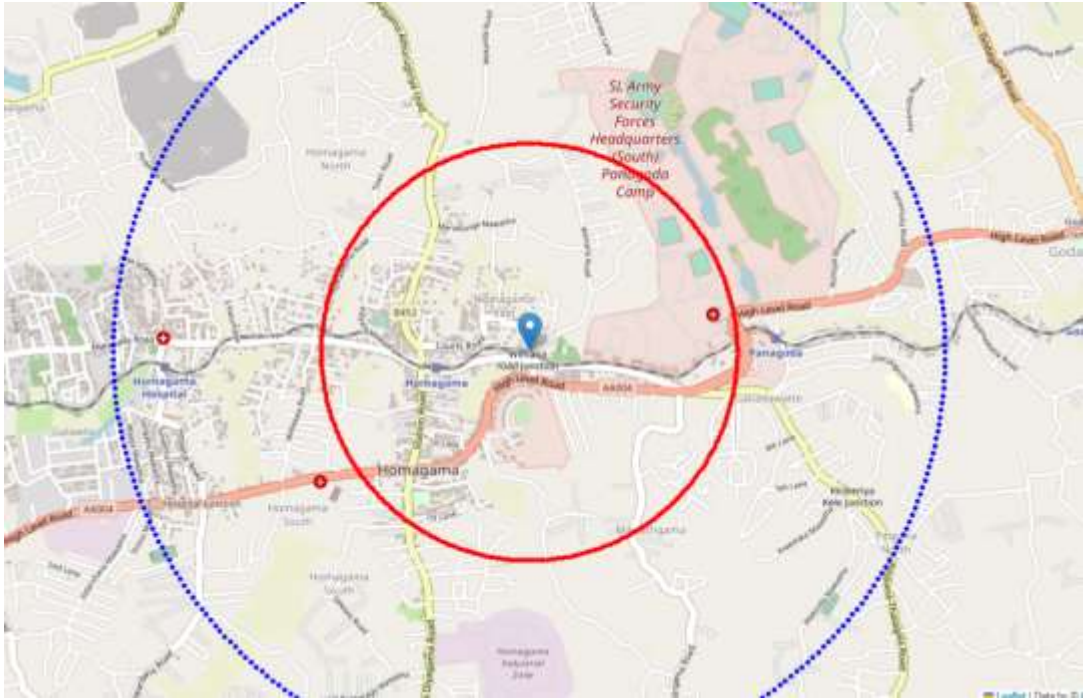
```python
from geopy import Point
from geopy.distance import geodesic
import numpy as np

# This is your starting point
start = Point(79.86713778, 6.92995042)

# This will store your circle of points
circle_points = []

# For each degree (0-360)
for degree in np.arange(0, 360, 1):
    # Get the point 2km away in this direction
    d = geodesic(kilometers=2)
    point = d.destination(point=start, bearing=degree)
    # Append to list
    circle_points.append((point.latitude, point.longitude))

# Print circle of points
for point in circle_points:
    print(point)
```

```
(79.88504944136207, 6.92995042)
(79.88504670855178, 6.931730000560174)
(79.88503851095774, 6.93350903619147)
(79.88502485109014, 6.935286982133204)
(79.88500573131178, 6.937063293965546)
(79.88498116293675, 6.9388374277756055)
(79.88495114882863, 6.940608840328197)
(79.88491569759815, 6.942376989234447)
(79.88487482250027, 6.944141333120438)
(79.8848285352509, 6.945901331795523)
(79.88477685002294, 6.947656446420242)
(79.88471978264191, 6.9494061396738O3)
(79.88465735058098, 6.951149875921062)
(79.8845895729556, 6.952887121378938S)
(79.88451647051754, 6.954617344282213)
(79.88443806564838, 6.956340015048696)
(79.88435438235265, 6.958054606443634)
(79.88426544625027, 6.959760593743371)
```

In this screen shot, the coordinates of the IoT device near the railway crossing is given.

And also, range circle points are also printed below in the screenshot.

EVIDENCE



In this screenshot, 2 range circles are shown centered on the IoT device at the railroad crossing.

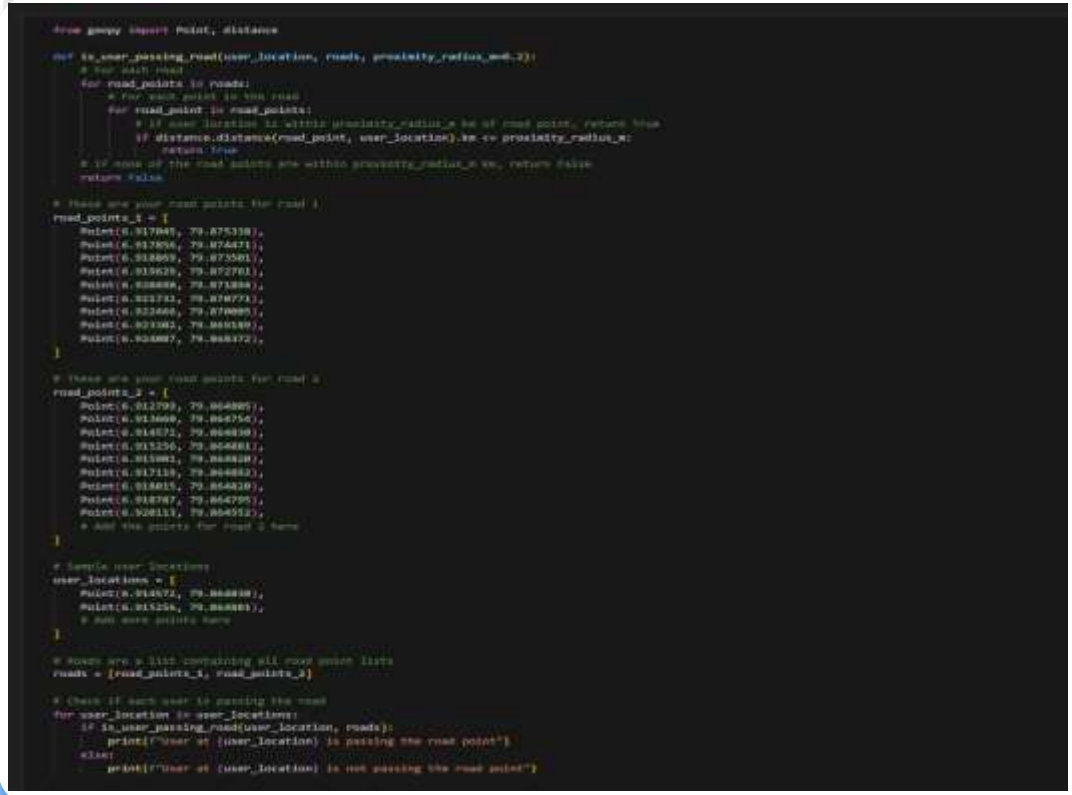
EVIDENCE

```
from geopy import Point, distance

def is_user_within_range(user_location, center, radius_km=2):
    return distance.distance(center, user_location).km <= radius_km

# This is your starting point
start = Point(6.92995042,79.86713778)

# Sample user locations
user_locations = [
    Point(6.914029, 79.878967),
    Point(6.914586, 79.878014),
    Point(6.915372, 79.876910),
    Point(6.915372, 79.876910),
    Point(6.916285, 79.876078),
    # Add more points here
]

# Check if each user is within the 2km range
for user_location in user_locations:
    if is_user_within_range(user_location, start):
        print(f"User at {user_location} is now in 2km range")
    else:
        print(f"User at {user_location} is outside the 2km range")
```

```
User at 6 54m 50.5044s N, 79 52m 44.2812s E is outside the 2km range
User at 6 54m 52.5096s N, 79 52m 40.8504s E is outside the 2km range
User at 6 54m 55.3392s N, 79 52m 36.876s E is now in 2km range
User at 6 54m 55.3392s N, 79 52m 36.876s E is now in 2km range
User at 6 54m 58.626s N, 79 52m 33.8808s E is now in 2km range
```

In this screenshot, shows whether the app user is in the range or outside the range.

EVIDENCE



```
from geopy import Point, distance

def is_user_passing_road(user_location, roads, proximity_radius_km=0.2):
    # For each road
    for road_points in roads:
        # For each point in the road
        for road_point in road_points:
            # If user location is within proximity_radius_m km of road point, return True
            if distance.distance(road_point, user_location).km <= proximity_radius_m:
                return True
    # If none of the road points are within proximity_radius_m km, return False
    return False

# These are your road points for road 1
road_points_1 = [
    Point(6.917845, 79.875338),
    Point(6.917856, 79.874471),
    Point(6.918059, 79.873501),
    Point(6.919629, 79.872701),
    Point(6.920888, 79.871884),
    Point(6.921731, 79.870771),
    Point(6.922460, 79.870005),
    Point(6.923381, 79.869589),
    Point(6.924887, 79.868372),
]

# These are your road points for road 2
road_points_2 = [
    Point(6.912799, 79.864805),
    Point(6.913660, 79.864756),
    Point(6.914572, 79.864830),
    Point(6.915256, 79.864881),
    Point(6.915981, 79.864820),
    Point(6.917119, 79.864811),
    Point(6.918015, 79.864820),
    Point(6.918787, 79.864795),
    Point(6.920111, 79.864512),
    # Add the points for road 2 here
]

# Sample user locations
user_locations = [
    Point(6.914572, 79.864830),
    Point(6.915256, 79.864881),
    # Add more points here
]

# Roads are a list containing all road point lists
roads = [road_points_1, road_points_2]

# Check if each user is passing the road
for user_location in user_locations:
    if is_user_passing_road(user_location, roads):
        print(f"User at {user_location} is passing the road point")
    else:
        print(f"User at {user_location} is not passing the road point")
```

In this screenshot, shows multiple road coordinates are added near the railway crossing.

TOOLS AND TECHNOLOGIES

Flutter python™ Firebase

FRONTEND	FLUTTER
BACKEND	PYTHON
DATABASE	FIREBASE

Libraries :

Flask NumPy Folium matplotlib

- Flask
- Numpy
- Folium
- Matplotlib

IDE s :

Android Studio

TASKS TO BE COMPLETED

DEVELOPING THE APPLICATION FRONT-END WITH USER FRIENDLY USER INTERFACES.

COLLECT DATA TO ANALYZE THE PAST PATTERNS OF VEHICLES AND PREDICT IF THEY ARE LIKELY TO CROSS THE RAILWAY CROSSING ON A GIVEN DAY.

COMPLETE THE APPLICATION BACK-END WITH ALL NECESSARY IMPLEMENTATIONS.

Security analysis for Train Tracking System.

Amarasinghe C.D
IT20187064
Cyber Security

RESEARCH GAP

Security analysis for Train Tracking System

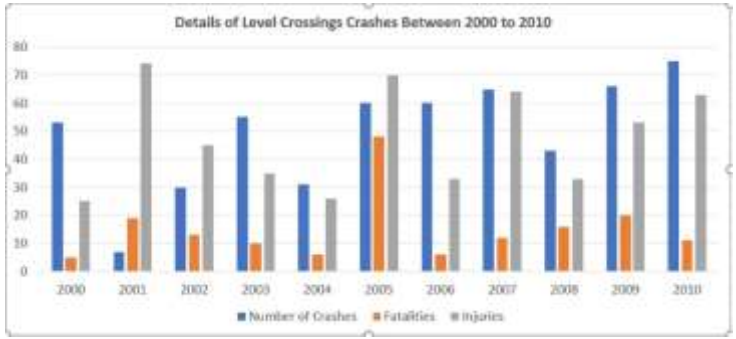
- Wireless Communication Security: Train tracking systems rely heavily on wireless communication to transmit data between trains and control centers. However, wireless communication is vulnerable to interception and interference. Research could focus on identifying the security risks associated with wireless communication in train tracking systems and developing effective security measures to address these risks.

- Privacy and security: Mobile applications for railway detection and tracking may gather private data regarding train routes, stops, and freight. To create safe and private applications that safeguard user information and guarantee the security and safety of railway operations, research is required.

RESEARCH PROBLEM

Security analysis for Train Tracking System

- There are many problems which occur in railways as there are no proper system implemented for the railways in Sri Lanka.
- Above statistics show the no of crashes are increasing year by year in railway crossings.
- There's no proper solution for these crashes, so implementation of a mobile application which have a connection with the railway crossing is the proposed solution for this problem.

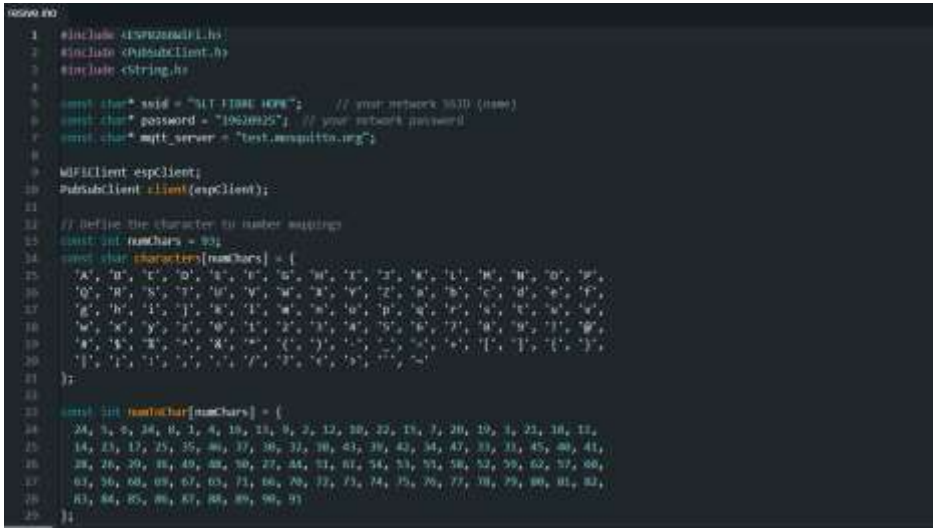


CURRENT PROCESS

Implementation of encryption system

Implementation of Security

EVIDENCE



```
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include <string.h>

const char* ssid = "SLT FIBRE HOME";     // your network SSID (name)
const char* password = "19620825";  // your network password
const char* mqtt_server = "test.mosquitto.org";

WiFiClient espClient;
PubSubClient client(espClient);

// Define the character to number mappings
const int numChars = 93;
const char characters[numChars] = {
  'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P',
  'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z', 'a', 'b', 'c', 'd', 'e', 'f',
  'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v',
  'w', 'x', 'y', 'z', '0', '1', '2', '3', '4', '5', '6', '7', '8', '9', '!', '@',
  '#', '$', '%', '^', '&', '*', '(', ')', '-', '_', '=', '+', '[', ']', '{', '}',
  '|', '\', ';', ':', '"', '/', '?', '<', '>', ',', '~'
};

const int numToChar[numChars] = {
  24, 5, 9, 34, 8, 1, 4, 10, 11, 9, 3, 12, 10, 22, 19, 7, 20, 19, 1, 23, 18, 12,
  14, 13, 17, 25, 35, 36, 37, 38, 32, 30, 43, 39, 42, 34, 47, 33, 31, 45, 40, 41,
  28, 26, 29, 46, 49, 48, 50, 27, 44, 51, 67, 54, 53, 55, 58, 52, 59, 62, 57, 60,
  63, 56, 68, 69, 67, 65, 71, 66, 70, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82,
  83, 84, 85, 86, 87, 88, 89, 90, 91
};
```

So this custom encrypt works as, I selected alphabet and make alphabet letters to numbers randomly.

The reason for putting it like that is because it is easy to find if you put it like 1,2,3,4.

EVIDENCE

Catch the encrypted data and decrypt on this device.

This output one converts latitude and longitude numbers. It is a frequent one.

It is set to transfer when a message is sent.

```
19:00:07.961 -> Assigned numbers: 33,40,45,43,42,36,49,30,32,83,60,52,85,52,52,73,33,35,36,42,36,49,30,32,83,58,61,85,53,52
19:00:07.961 -> Reversed string: longitude:95.55-latitude:40.25
19:00:08.981 -> Assigned numbers: 33,40,45,43,42,36,49,30,32,83,60,52,85,52,52,73,33,35,36,42,36,49,30,32,83,58,61,85,53,52
19:00:08.981 -> Reversed string: longitude:95.55-latitude:40.25
19:00:09.974 -> Assigned numbers: 33,40,45,43,42,36,49,30,32,83,60,52,85,52,52,73,33,35,36,42,36,49,30,32,83,58,61,85,53,52
19:00:09.974 -> Reversed string: longitude:95.55-latitude:40.25
```

OBJECTIVE

MAIN OBJECTIVE

- To develop a system to provide the security for the mobile application.

SUB OBJECTIVE

- Gathering data required for the implementation of the mobile security. This component need dataset such as general mobile threats and problems occur due to lack of security in mobile applications. .
- Using of tools for ensure the safety of the mobile application such as penetration testing tools.
- The mobile application is using personal information as the user needs to be a registered user. An encryption is used to secure the password and the safety of the user data safety should be ensured within the mobile application.

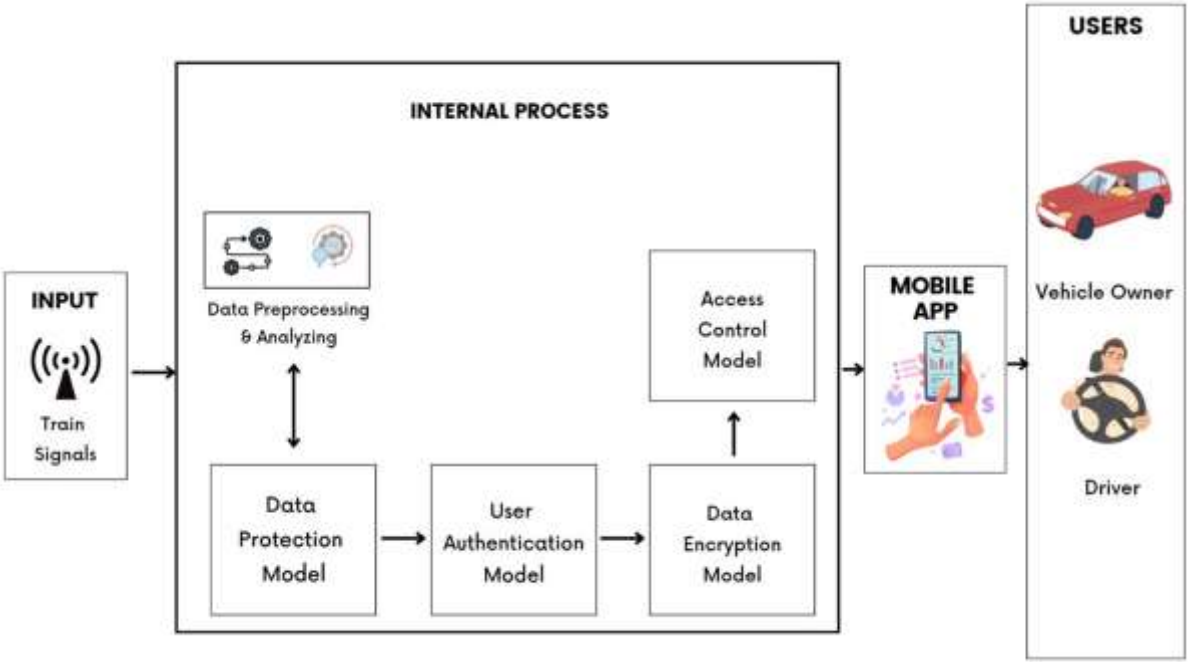
TASK TO BE COMPLETED

secure the otp come from mobile application otp.

A physical security can be placed on the physical device.

physical secure the device and also when using internet using https.

SYSTEM OVERVIEW



TOOLS AND TECHNOLOGIES

FRONTEND	FLUTTER
BACKEND	PYTHON
DATABASE	FIREBASE

REFERENCES

- https://www.synopsys.com/glossary/what-is-penetration-testing.html#:~:text=A%20p
- https://www.getastra.com/blog/app-security/mobile-application-security-testing/
- https://cyberlegion.io/mobile-application-penetration-testing/
- https://techbeacon.com/security/5-essential-steps-securing-enterprise-mobile-apps

Alerting system from IOT Device to SIM

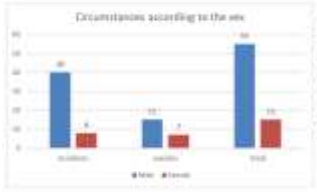
Wijewardene L.L
IT20101824
Information Technology

RESEARCH PROBLEM

Sending the flooded messages from the IOT device for the SIM users who are within a 1.5km radius

- So many collisions because of lack of real-time flooding alerting systems near railway crossings.

The following figure illustrates the different circumstances of death.



Sixty nine percent (n= 48) were accidental while the rest were suicidal. Homicidal deaths or post mortem disposal was not observed. Accident was the foremost apparent manner of death among males 73% (n=48) (Figure 1).

Figure 1: Circumstances of death

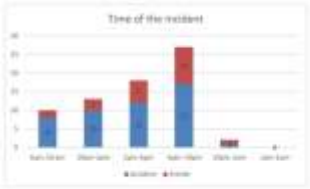
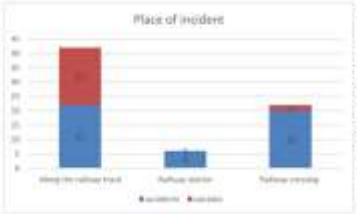


Figure 2: Time of day vs circumstances

The highest number of fatalities 64% (n=45) were observed between 2 pm to 10 pm. Sixty three percent (n =30) of accidents occurred during day light (6 am to 6 pm) whilst suicides showed a higher frequency 77% (n=17) in the evening and night. This was statistically significant (Figure 2).



The place of incident included railway station (and surrounding), rail track away from station, protected/ unprotected level crossings or close to the level crossing. (Figure 3) Rail track, railway station or the rail crossings are not safe for the users showing 22 (45.8%), 6 (12.5%) and 20 (41.7%) accidental fatalities respectively in these locations.

Of the 230 deaths and 477 cases of injury from 1,456 train accidents in 2018, 212 fatalities were classified as suicides or due to trespass on tracks. The vast majority of victims (167) were male. In 2018, level crossing accidents have caused 13 deaths and 69 cases of injury. Five people lost their lives falling off trains. In 2019, there were 215 deaths and 369 injuries coming from 1,385 train accidents. All but 15 of the deaths were classified as suicides/trespass on tracks. A further 10 deaths came from level crossing accidents, and five more from 76 cases of people falling off trains.

OBJECTIVE

MAIN OBJECTIVE

- Sending the flooded messages from the IOT device for the SIM users who are within a 1.5km radius

SUB OBJECTIVE

- Gathering the data from the IOT Device
- Provide the accurate real-time alert for the user within the specific radius.
- Flooding the alert among the users through the SIM.
- Make the flooding alert fast as possible among all the user's within the radius.

EVIDENCE

```
FirebaseData firebaseData;
FirebaseESP8266 firebase(FIREBASE_ESP8266HTTP_CLIENT);

RTC_DS3231 rtc;
SoftwareSerial sim800l(7, 8); // RX, TX

void setup() {
  Wire.begin();
  sim800l.begin(9600);
  delay(3000); // Let the module self-initialize

  if (!rtc.begin()) {
    sim800l.println("Couldn't find RTC");
    while (1);
  }

  if (rtc.lostPower()) {
    sim800l.println("RTC lost power, let's set the time!");
    rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
  }

  firebase.begin(FIREBASE_HOST, FIREBASE_AUTH); // connect to Firebase
}
```

In this screen shot, establish a connection with Firebase, retrieve data from the RTC module,

EVIDENCE

```
void loop() {
  DateTime now = rtc.now();

  if (Firebase.getJSON(firebaseData, "/")) { // get all the users from Firebase
    for (JsonPair keyValue : firebaseData.jsonObject()) { // loop over all users
      const char* key = keyValue.key().c_str();
      JsonObject& value = keyValue.value().as<JsonObject>();

      const char* number = value["number"];
      const char* time = value["time"];

      // Here you need to make sure the format of your time string from Firebase matches the one from your RTC
      // If they don't match, you'll need to parse/convert them to a common format for comparison
      String now_time = String(now.hour()) + ":" + String(now.minute());

      if (now_time == time) { // if the current time matches the time from Firebase
        sendSms(number, "Careful"); // send the SMS
      }
    }
  }
}
```

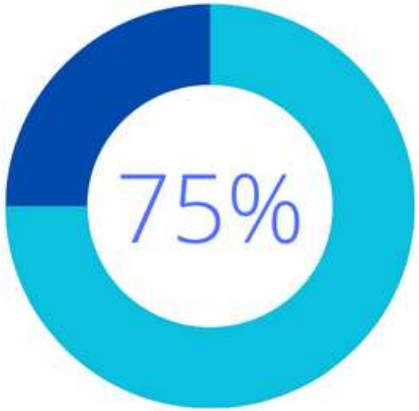
In this screen shot, get all the gathered users from the database firebase and also setting up the variables number and time.

EVIDENCE

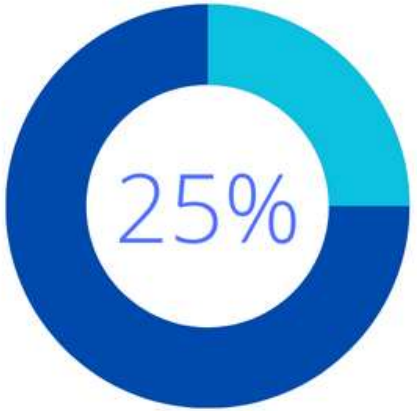
```
55
56  void sendSms(const char* number, const char* message) {
57    sim800l.println("AT+CMGF=1"); // Set the SMS mode to text
58    delay(1000);
59    sim800l.print("AT+CMGS=\"");
60    sim800l.print(number);
61    sim800l.println("\"");
62    delay(1000);
63    sim800l.print(message);
64    sim800l.println((char)26);
65    delay(1000);
```

In this screen shot, setting up the initial SMS sending mode

CURRENT PROCESS



Gathering the data from the IOT Device



Flooding the alert among the users through the SIM.

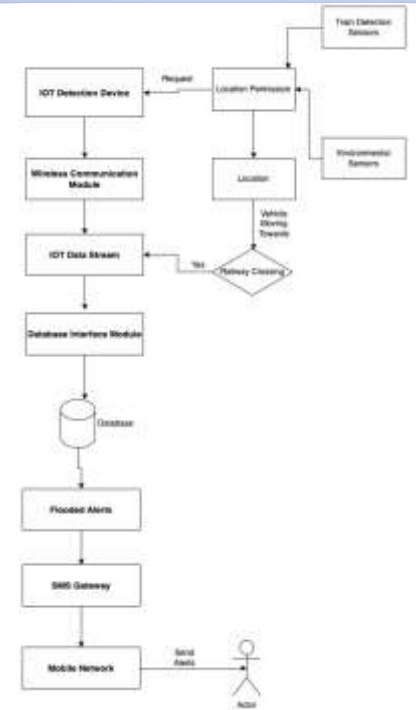
TASK TO BE COMPLETED

Integrated to the SIM 800I GSM model

Implementation of flooded message device

Implementation of the frontend for the mobile application

SYSTEM OVERVIEW



TOOLS AND TECHNOLOGIES

FRONTEND	ADRUINO IDE
SERIAL DEVICE	RTC_DS3231
DATABASE	FIREBASE

GANTT CHART

No	Task List	December	January	February	March	April	May	June	July	August	September	October	November
1	Initial Stage												
	Research Topic Selection												
	Requirement Gathering												
	Study on Research Area												
	Topic Evaluation form submission												
	Topic Evaluation (Project pre-assessments) resubmission												
	Topic Approved												
	Project Charter												
2	Proposal Stage												
	Proposal Draft Submission												
	proposal Presentation												
3	Implementation Stage 1												
	System Design and Planning												
	Implementation of functions												
	Integration and testing Level 1												
	Progress presentation -50%												
	Prepare Research Paper												
4	Implementation Stage 2												
	Implementation of functions												
	Integration and testing Level 2												
	Progress presenation -100%												
5	Final Stage												
	Final Thesis												
	Final Presentation												

COMMERCIALIZE

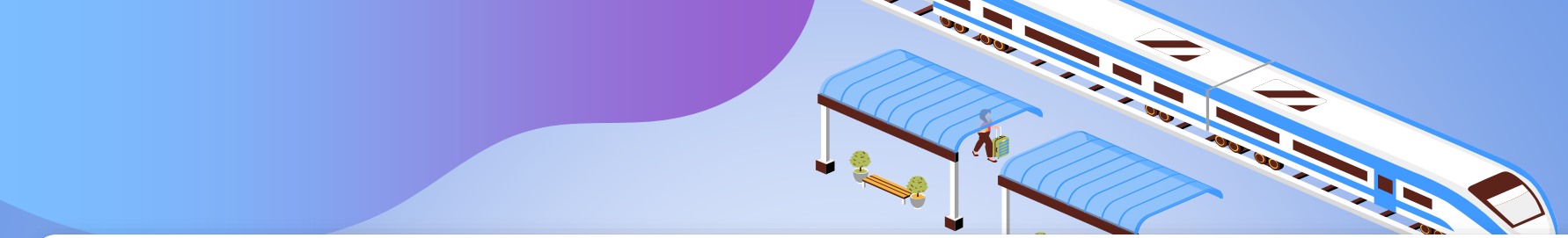
- Social Media Marketing
- Identifying the Target Audience
- Partnership with a reputed company
- Attending Award Competitions

BUDGET (Up to Now)

ITEM	BUDGET
Node MCU Board (x2)	4000.00 LKR
Micro USB Cable(x2)	700.00 LKR
GSM Tracker	1000.00 LKR
Jumper Wires	150.00 LKR
Bread Board	200.00 LKR
Travelling Costs	1000.00 LKR
Electricity & Internet Costs	1000.00 LKR
Total	8050.00 LKR

REFERENCES

- [1] N. News, "NBC News," 27 April 2005. [Online]. Available: https://www.nbcnews.com/id/wbna7647780.
- [2] Wikipedia, "Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Polgahawela_level_crossing_accident#:~:text=Polgahawela%20level%20crossing%20accident%20was,the%20death%20of%2041%20people..
- [3] S. L. Tweet, "Twitter," 25 May 2019. [Online]. Available: https://twitter.com/SriLankaTweet/status/1132093011997863936/photo/1.
- [4] W. D. Info, "WorldDataInfo," 2020. [Online]. Available: https://www.worlddata.info/asia/sri-lanka/telecommunication.php#:~:text=Under%20the%20country%20code%20%2B94,the%20world's%20average%20by%20population..



THANK YOU!

TMP-23-302