TRAIN TRACKING AND DETECTION SYSTEM

23-302

Jayanga B.M.C

B.Sc. (Hons) Degree in Information Technology
(Specializing in Data Science)

Department of Information Technology

Sri Lanka Institute of Information Technology
Sri Lanka

2023

TRAIN TRACKING AND DETECTION SYSTEM

23-302

B.Sc. (Hons) Degree in Information Technology
(Specializing in Data Science)

Department of Information Technology

Sri Lanka Institute of Information Technology
Sri Lanka

2023

DECLARATION

I Jayanga B.M.C, declare that the research project report is my original work, and it has not been submitted in whole or in part for any other degree or qualification. Any ideas, data, or information obtained from other sources have been fully acknowledged by means of a citation or reference. I have followed the guidelines for academic writing and referencing provided by my university, and the project conforms to the required standard.

Name	Student ID	Signature
Jayanga B.M.C	IT20188672	

The above candidates are carrying out research for the undergraduate Dissertation under my supervision.

Signature of the supervisor: Date:

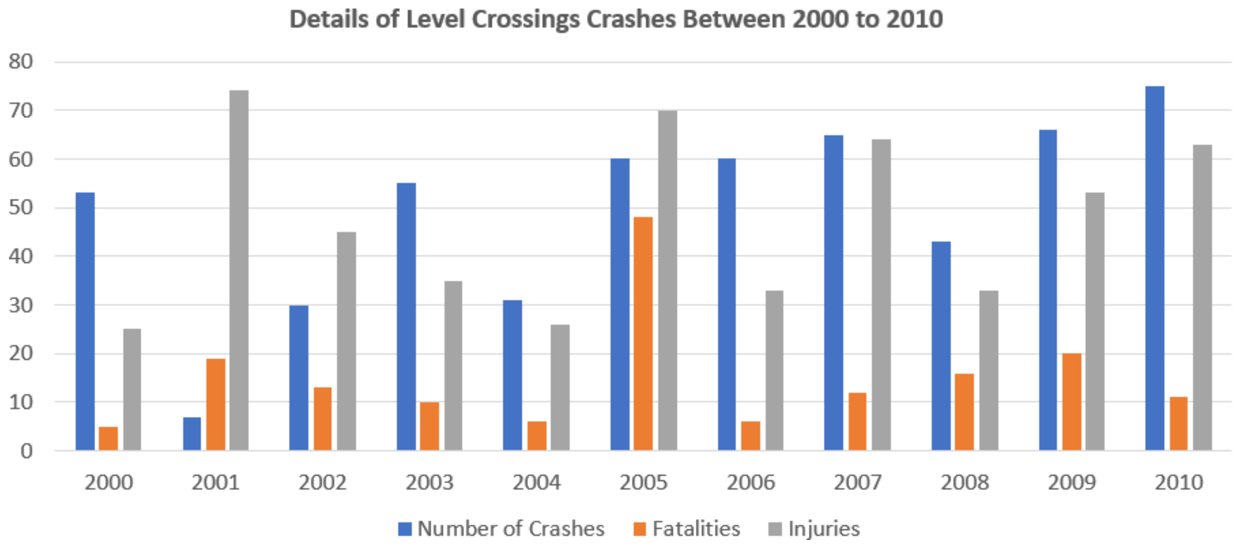
ABSTRACT

Railway Crossing accidents have been a significant safety concern not only worldwide, but also in Sri Lanka, resulting in significant loss of life and property annually. The biggest railway crossing collision that happen in Sri Lanka was on 27th April 2005, near Polgahawela, resulted the death of 41 citizens.

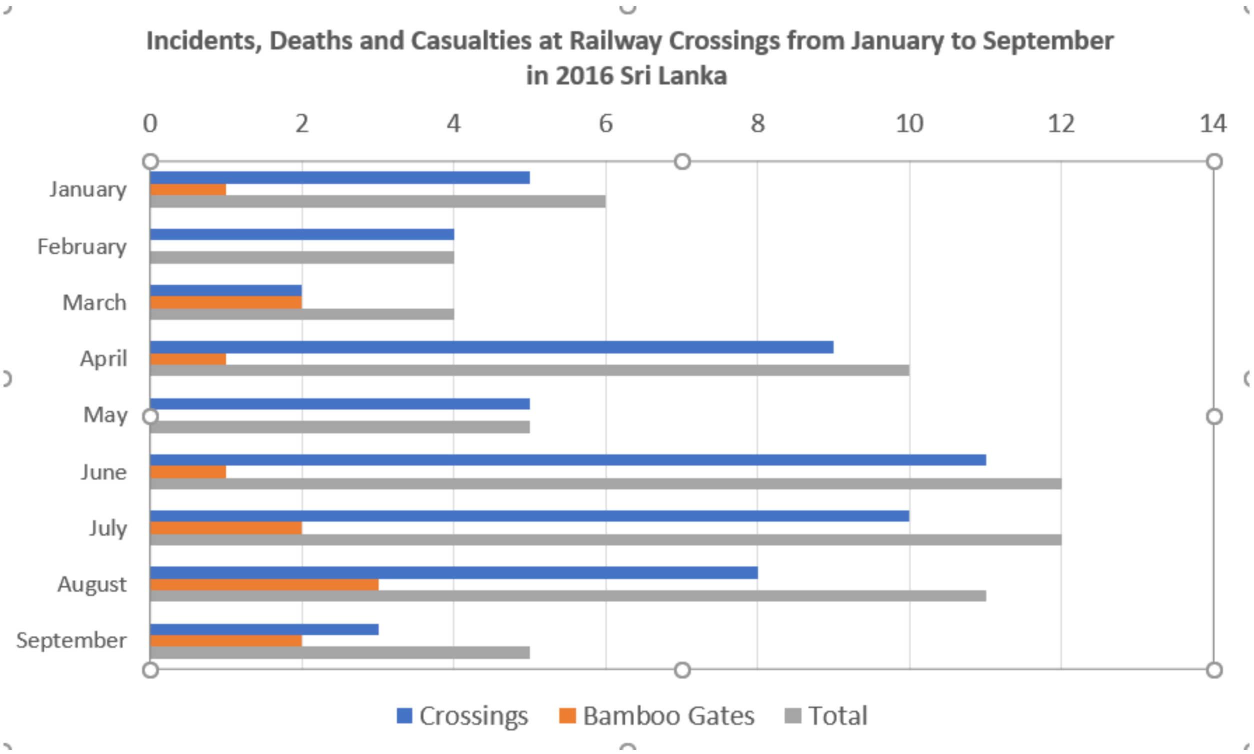
[1]

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 [15]

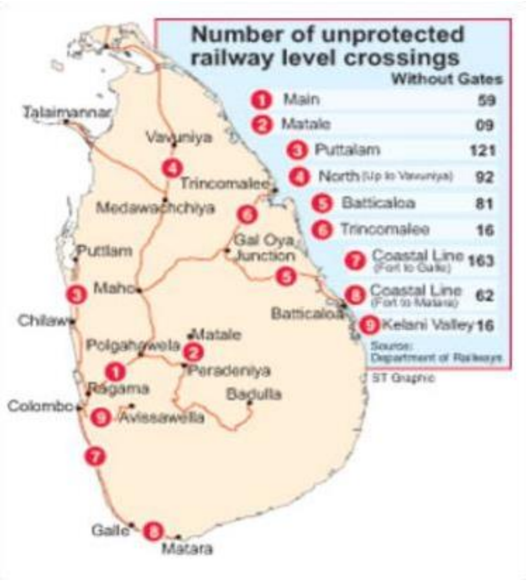
These are some detailed facts that proves how much damage is done annually from railway collision in Sri Lanka.



[2]


Incidents, Deaths and Casualties at Railway Crossings from January to September in 2016 Sri Lanka

[2]


Number of unprotected railway level crossings

		Without Gates
1	Main	59
2	Matale	09
3	Puttalam	121
4	North (Up to Vavuniya)	92
5	Batticaloa	81
6	Trincomalee	16
7	Coastal Line (Fort to Galle)	163
8	Coastal Line (Fort to Matara)	62
9	Kelani Valley	16

Source: Department of Railways

ST Graphic

[3]

And also statistics proves that, 1 person dies due to a road accident in every 3 hours averagely in Sri Lanka. [4] and Every 3 days, a Child dies in Road Accident. [4]

We identified some major reasons for these collisions. Major reason for so much collisions in Sri Lanka is there are so much unsafety unprotected railway crossings all around the Island. Above image [3] shows how many unprotected railway crossings in all around the Island. Due to some heavy rains or fog conditions, people cannot see the upcoming railway crossings when they are driving the vehicle. Human Errors is also another fact for railway collisions in Sri Lanka. That means citizens may misjudge the speed or distance of an approaching train, leading to accidents or near-misses. And also, Citizens may not be aware of the dangers posed by railway crossings or the proper safety procedures to follow when crossing tracks. Therefore, providing solutions to reduce these collisions happening in railway crossings became a major need specially in Sri Lanka because many human and animal lives and properties lost to the country annually. But from the government side, we cannot see they are considering this as a big problem. They didn't even take necessary steps to reduce these railway crossing collisions like providing gates near railway crossings, signal and alarms likewise. As citizens who feel this as a major problem in Sri Lanka, we need to find some solution to reduce these annual railway crossings collisions since the government is not involving on this matter much.

As undergraduate students doing the degree on IT field, we felt that we need to address this problem from IT based solution.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviation	Description
IT	Information Technology
IOT	Internet Of Things
App	Application
GPS	Global Positioning System
SMS	Short Message Service
UI	User Interface

1. INTRODUCTION

Currently, there is no proper solution to address the railway crossing collisions in Sri Lanka. So as undergraduate students who are specializing in IT field, we propose a IT-based solution to address these railway crossing collisions in Sri Lanka. Even though there are apps to alert train schedules and train routes, time tables, there is no app to address the safety of citizens who are crossing the railway crossings in Sri Lanka. The "Train Detection & Alert System" for citizens ins Sri Lanka project aims to address the challenges faced by citizens who cross the railway crossings through the use of advanced technology and infrastructure improvements. To address this problem, our research project proposes a system that utilizes various technologies, including IoT devices, GPS tracking, machine learning algorithms, security technologies and real-time databases.

Our Train Detection & Alert System includes multiple components, such as real-time tracking of trains and vehicles near the railway crossing, alert system to vehicles which are moving towards to the railway crossing, alert system to all the users who are near to the railway crossing, sending real-time train location details to relevant IOT devices, security system for the system and predictive maintenance system. These all components work together to detect potential safety hazards and prevent accidents from occurring. The project aims to create a user-friendly interface for the system, making it accessible to a wide range of users and enhancing the overall passenger experience. Additionally, the system will be used by train authorities to monitor the railway crossings collisions, helping to improve the overall efficiency and safety of citizens and taking relevant actions to reduce the collisions happening in railway crossings.

My component of the research project aims to predict whether a vehicle is likely to cross a railway crossing on a particular time and particular day when a train is also approaching, and if he/she is likely to cross the railway crossing, send alert warning messages. And then Analyse the past patterns of vehicles and predict if they are likely to cross the railway crossing on a given day.

Since we are developing an app for our users, we ask the location permission from the user when they download, install and login to the app. From the location permission we asked, we can track the app and the user. So we are creating a range circle about 1.5km rounded by the IoT device near the railway crossing. And if we detect a vehicle user who use our app, entered that range circle from tracking, we start continuing tracking that app user. So if only that app user is approaching towards the way of railway crossing, warning message alerts are sent to that user that railway crossing is nearby and train is approaching on that way via the app. Otherwise, warning message alerts are not sent to the app user. That means if we detect there is no movement of the app, or the app is going away from the railway crossing, or the app is not in a road near a railway crossing, that alert messages are not sent to that particular user of the app. Then we store the vehicle patterns of that particular app user, vehicle patterns mean the roads which he or she used to drive near that particular railway crossing. And also, we store the data of date, time as well. And then store these data in some real-time database. We are planning to "Firebase-Real time database" to store these data at the moment. So after collecting these kind of vehicle patterns of the same app user more times, we pre-process that data to analyse. Then analyse that particular data to make predictions. Then we do predictions that whether this app user is likely to cross the railway crossing on a particular day and particular time when a train is approaching towards the railway crossing. If it is, we send alert messages to that user not only through the app, but also through the SIM card as well. The reason for that is, for some reason that user's mobile data might be off for some reason, or that particular user might closed the app. So even though on such scenarios, the user will be notified via SMS alert notifications.

The proposed Train detection and Alert System is expected to improve safety of the citizen who cross the railway crossings significantly. It will help to reduce the number of accidents and fatalities near railway crossings. Overall, this research project has the potential to make a significant contribution to the citizens of Sri Lanka who are crossing the railway crossings, safety and development of them. And ultimately, improving the quality of life for citizens in Sri Lanka.

2. LITERATURE REVIEW

2.1 BACKGROUND

The concept of providing solutions for the safety of people who are crossing railway crossings has been around for a long time, but the development of modern technology has made it possible to implement more effective solutions. The first research on providing solutions for the safety of people who are crossing railway crossings may date back to the late 19th century. At that time, many accidents occurred due to the lack of proper signalling systems, and there was a growing need to improve safety measures. The first documented effort to address this issue dates back to the 1870s when the first automatic signal system was installed in England. Since then, many countries have made significant progress in developing and implementing safety measures for railway crossings. However, with the increasing number of vehicles and railway crossings, new and innovative solutions are still being researched and developed to improve safety for pedestrians and drivers alike.

I mainly focused on some points when I am studying about the literature review background for my relevant component.

1. **App Based safety solutions** – I explored literature on app–based safety solutions for the users who are crossing the railway crossings regularly.

2. **Vehicle Tracking** – I explored literature on vehicle tracking near a railway crossing.

3. **Alerts Sending -** I explored literature on alerts sending to the users when they are approaching towards a railway crossing and a train is coming on that way as well.

4. **Predictive Analysis -** I explored literature on previous researches that do some predictive analysis and predicting whether some vehicle is likely to cross the railway crossing.

5. **Real-time Databases** - Research literature on real-time databases and their use in applications. This could include information on different types of real-time databases,

The use of real-time data analysis and prediction has been increasingly popular in the transportation industry, with the aim of improving the overall efficiency and safety of the transport systems. In the context of train tracking, various methods such as GPS, RFID, and GSM have been used to monitor train movement and location. However, the use of these technologies has its limitations, and real-time prediction of train location remains a challenge.

In recent years, machine learning and artificial intelligence techniques have been used to improve train tracking and prediction but not to analyse vehicle near the railway crossings. These techniques have the potential to analyse and process large amounts of real-time data, leading to more accurate predictions of train location and arrival times. One promising approach is the use of recurrent neural networks (RNNs), which have shown success in predicting the future trajectory of trains.

2.2 LITERATURE SURVEY

The study by A. A. T. P. De Silva University of Colombo School of Computing [5] has introduced a Real-time location based crowdsourcing train tracking android application to enhance the effectiveness and efficiency of public train transportation. The proposed application is developed by combining Global Positioning System (GPS), mobile computing, and crowdsourcing technologies to gain information from the passengers and provide visual positioning using Google map in real-time. Additionally, it predicts the estimated time to arrival (ETA) of a train to any given railway station for better user experience and for better admin management.

The study [6] introduces a method to detect the train using a vibration sensor on the railway track. The Arduino microcontroller was chosen to detect the train and send the signal to the level crossing subsystem (LCS) as it has enough input lines to connect the relevant sensors and actuators. An alarm system with lights is used with an IoT system that consists of Radio Frequency (RF) receiver, Raspberry Pi Model 3 and cameras.

“Evaluation of Safety Degree at Railway Crossings in Order to Achieve Sustainable Traffic Management: A Novel Integrated Fuzzy MCDM Model” is a research done in 2021 by Aleksandar Blagojević, Sandra Kasalica, Željko Stević, Goran Tričković and Vesna Pavelkić. [7] They have done a research to address railway crossings collisions by measuring the analysis of safety degree on some railway crossings. They have used some mathematical formulas to count this analysis of safety degree.

“Traffic Signal Operations Near Highway-Rail Grade Crossings” is a another research done in USA by Transportation Research Board and National Research Council in 1999. [8] They mainly focused on the alarm and alert systems in the railway crossings. It’s an automatic railway crossing safety system that means, when a train is approaching, automatically safety options near the railway crossings like alarms, red lights are activated.

“An Automatic Railway Level Crossing System with Crack Detection” is another research done to minimize the collisions in the railway crossings. [9] This researched have used IoT devices like microcontrollers, Ultra Sonic Sensors, GSM Modules to detect cracks and barrier control when a train is approaching. GSM module is used to send alert messages to relevant authorities when a crack is detected. It is said that this system have a high accuracy of 93% reliability of crack detection. And also they send alerts of railway crossings gates and alarm status like whether they are on or off when a train is approaching to the relevant authorities.

The studies [10] and [11] have used IoT-based systems to automate railway crossings safety system controls. In the [10] system, Train current location information is sent to users with a link to see the current location of the train via SMS in this system. Arduino Uno, server motors, and sonar sensors have been utilized to implement the automated railway crossings bars in this system. In the [11] system, manually operated railway crossing gates are transformed into a fully automatic system operated through wirelessly by the stationmaster using Arduino Uno and ultrasonic sensors. And, this particular system detects the number of persons and vehicles that remain on the tracks after closing of gates by the help of Open-Source Computer Vision Library (OpenCV) with a buzzer and LED bulb connected to it to give alerts during opening and closing of gates.

“RDMNS.LK” is a Sri Lankan-developed mobile application to alert railway alerts in Sri Lanka. [12] This application shows the users the live locations of the trains when a train is approaching. And also, this app alerts the user whether trains are delayed or not and if the train is delayed, how much time the train is delayed. This app also shows us train station contact numbers, ticket prices from Colombo Fort and so many more details related to trains in Sri Lanka. This application is available for both Android and IOS currently.

“Smart Railway Crossing Surveillance System” is another research project done to address railway crossings collisions. [13] This system addressed railway collisions by displaying whether the railway gate is opened or closed based on the train passing, the most recent gate closure timestamp, and each time when the gate is closed, a centralized platform for knowing the condition of the gate, traffic density, and intrusion detection near the railway gate. Based on the experimental results, this automated railway crossing assistance outperformed human-assisted railway crossings, resulting in an efficient and cost-effective solution for traffic regulation and avoidance of accidents at railway gates. They have used an Arduino IoT system for this system as well.

The study [14] aimed to aware the train driver about what is blocking the railway path and automatically suggest him to follow safety procedures. And also, the other thing focused on is preventing accidents in wildlife areas in railway areas in Sri Lanka. Also, to create an inter-connected smart grid to provide real-time train movement tracking across the country and make way to insert an AI system to automate entire train operations. An IoT system has been utilized to identify train movements and some formulas have been used to calculate some calculations.

The literature survey highlights several research studies related to the safety of railway crossings. Researchers have proposed various solutions, including mathematical models to evaluate the safety degree of railway crossings, automatic safety systems activated by approaching trains, IoT- based systems for crack detection and barrier control, mobile applications to alert users about train locations and delays, and smart surveillance systems to monitor railway gates and traffic density. These studies provide valuable insights into addressing railway crossing collisions and can help inform the development of effective safety measures.

3. RESEARCH GAP

Research Features	Proposed Component	Automatic Railway Crossing System with Crack Detection[8]	RDMNS.LK: LIVE Train Alerts[9]	Smart Railway Crossing Surveillance System[10]
Mobile App	✓	✗	✓	✓
Vehicle Tracking	✓	✓	✗	✗
Alerts Sending to the Users	✓	✗	✓	✓
Analyzing the Past Vehicle Patterns	✓	✗	✗	✗
Predict the likely to Cross the Railway Crossing on particular Day	✓	✗	✗	✗
Send Alerts to Relevant Authorities	✗	✓	✗	✗
Train Tracking	✗	✓	✓	✓

Table 1: Table of Research Gap

This research project aims to bridge this gap by developing a Train Detection System for citizens via the sim or the app, which can be accessed by a wider range of users. The proposed system utilizes GSM trackers, prediction patterns, flooded message systems and other real-time data sources to provide up-to-date information of trains and provide alerts about trains, which can be used by the app users and people near the railway crossings to avoid accidents with trains in the railway crossings via the app, sim or IoT devices near the railway crossing.

My component is “Sending alert messages only to the app users who are approaching a railway crossing”. The main objective of this is to analyse the past patterns of vehicles and predict if they are likely to cross the railway crossing on a given day. And sub objectives are to identify the vehicles that are moving towards the railway crossing, and send alerts only to them through the app, To track the location of the user's vehicle when they enter the range circle near the railway crossing, Store the tracked patterns of the user's vehicle and send a notification to the user's SIM card if they are predicted to cross the railway crossing on a particular day, even if they are not using the app or have mobile data turned off.

According to the literature survey above conducted, it is evident that Train detection system for citizens who cross the railway crossing is a much-needed implementation since it is really rare to find researches that targets the safety of citizen who cross the railway crossings. Up to now in the researches that were conducted under this scope, we can see that they have covered the following areas.

- Train Detection and live location showing
- Railway crossings crack detection
- Railway crossings Alarms and alert systems Improvements
- Measuring the analysis of safety degree on railway crossings
- Railway Crossing Surveillance System
- Train alerts and delay time showing
- Train detection using IoT devices

4. RESEARCH PROBLEM

According to Sri Lanka Tweet (2019), Every 3 hours, a person dies from a road accident in Sri Lanka. [5] And also, Every 3 days, a child dies from a road accident in Sri Lanka. According to a study on railway-roadway level crossing safety, which is done by Kulasingham Ragulan and Niranga Amarasingha, there are so many railway crossing collisions happen annually in Sri Lanka. [3]. And also, WorldData.info website shows that Population : Mobile Phone ratio is 1.4 per person in Sri Lanka. [11] That means the total number of phones are greater than the population of Sri Lanka. Even though the count of phones are higher than the population of Sri Lanka, there is no proper ways to use these phones to address the safety of people in Sri Lanka.

These railway crossing collisions has taken so many human lives as well as lot's of property damage. Today, there are several web applications and mobile applications about trains in Sri Lanka. But there are no proper IT based solutions to address the safety of people who cross the railway crossings, specially in Sri Lanka.

- How to enhance the safety of passengers and drivers near railway crossings?

One of the major research problems addressed by my component is the need to enhance the safety of passengers and drivers near railway crossings. Railway crossings can be dangerous places, and accidents involving vehicles and trains can result in severe injury or even death. My component seeks to address this issue by providing real-time message alerts about the approaching trains when a user is coming towards a railway crossing via the app and SMS messages.

- How to provide the predictions with highest accuracy rate?

Another research problem to be addressed in my component is the accuracy of vehicle likelihood of crossing the railway crossing when a train is approaching prediction. Inaccurate predictions could lead to mis timely alerts, which could cause passengers and drivers to become desensitized to the warnings provided by the app, resulting in a decreased level of safety. Additionally, the accuracy of vehicle prediction of crossing the railway is critical to the overall effectiveness of the this research project, and any improvements to the accuracy of the system could have a significant impact on the overall safety of railway crossings. Predictions should be made in higher accuracy for better safety of people.

- How to sort the vehicles which are approaching toward the railway crossing?

Since we are developing an app to alert the users, it is important to sort the users who are moving toward the railway crossing only. Otherwise, if the alerts are sent every time to the user when a train is approaching, it can be like a headache for the user. That will be a major User experience issue.

- How to develop the mobile application with a better user experience?

The app should be a optimized and should be in a simple and attractive UI for a better User Experience. Since we are targeting to commercialize this, the app will play a major rule since it will directly be connected with the user.

- How to integrate with real-time databases?

Since the most parts of the whole system connected with a database, it is important to connect them with a real-time database because outputs should be delivered so faster since this address the safety of people. So connecting the component parts with a better real-time database is so much important.

- How to send alert messages not only via the app but also to the SIM card as well?

The app user might not use the mobile app due to various reasons. Like whether he/ she may turned off the mobile data or his/her 3G or 4G connection's signal strength is not enough to connect to the internet at the moment. When this kind of scenario happens, it's important to send alerts to the user somehow.

5. OBJECTIVES

5.1 MAIN OBJECTIVE

- Analyse the past patterns of vehicles and predict if they are likely to cross the railway crossing on a given day.

The ultimate main objective of my component is to analyse the past patterns of drive patterns near a particular railway crossing of the vehicles of the app users and predict if that particular user is likely to cross the railway crossing when a train is approaching. To analyse the vehicle moving patterns, we need to track them. For that, the mobile application come to play. We ask the permission of location when the user is registering to the app. Then we store the vehicle movement patterns in the particular database. After storing vehicle movement patterns of same user more times, we can analyse that patterns and make a prediction of whether this user is likely to cross the railway crossing on the train approaching time or not. More and more vehicle patterns, increases the accuracy of the prediction as well.

5.2 SPECIFIC OBJECTIVES

Sub Objective 1: Identify the vehicles that are moving towards the railway crossing, and send alerts only to them through the app –

Vehicles which are moving toward the railway crossings are identified by the location permission of the particular user’s app. When it is identified that particular user is moving towards the railway crossing way only from the tracked location, alert messages are sent. Otherwise, that user is not moving, the user is going away from the railway crossing, like that kind of scenarios, alert messages are not sent to that particular user. This makes the User Experience of the user about app better.

Sub Objective 2: Send notification alerts to the user's SIM card if they are predicted to cross the railway crossing on a particular day, even if they are not using the app or have mobile data turned off –

This is a sub-objective should be accomplished after accomplishing the main objective. The app user might not use the mobile app due to various reasons. Like whether he/ she may turned off the mobile data or his/her 3G or 4G connection’s signal strength is not enough to connect to the internet at the moment. When this kind of scenario happens, it’s important to send alerts to the user to the SIM card as well.

In accomplishing the main objective, below sub-objectives need to be satisfied.

Sub Objective 3: Track the location of the user's vehicle when they enter the range circle near the railway crossing.

Sub Objective 4: Store the tracked patterns of the user's vehicle – Tracked patterns of vehicles should be stored in a real-time database.

Sub Objective 5: Analyse the vehicle movement patterns and do predictions with high accuracy – To make better predictions, Machine Learning algorithm should be chosen with the highest accuracy rate and it should be trained well.

---

6. METHODOLOGY

After having some discussions with our supervisor, co-supervisor and research panel members, requirements were identified, some were changed and finally finalized them. Background and literature survey was done for my component in the area of alerting the people who cross the railway crossings. And then found some implementations has been done before which are looks like bit similar world-wide. As it is evident in the literature survey, that there has not been done a specific IT based system to address the safety of people on the road, when they are moving towards a railway crossing and when a train is approaching as well.

The methodology for creating the proposed train detection and alert system includes the following steps:

1. Literature Survey
2. Data Collecting
3. Model Building
4. Model Testing
5. UI Designing
6. Security Designing
7. Security Testing
8. Integrating
9. Testing
10. Deployment of the Prototype

6.1 SYSTEM ARCHITECTURE DIAGRAM

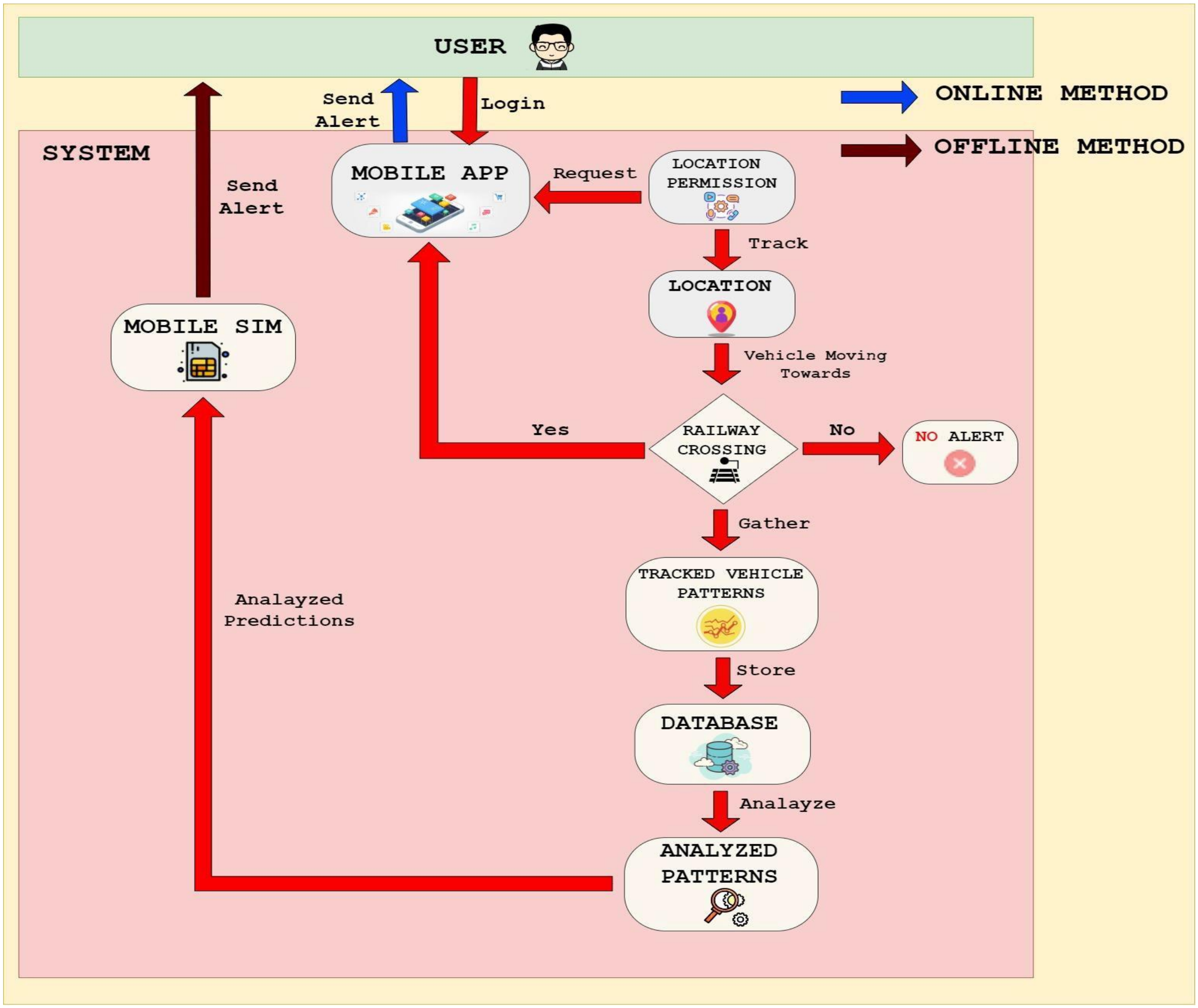


Figure 6.1: Overall System Architecture Diagram

The location permission is asked from the user when he/she logs to the app to track him/her. If that particular user is moving towards the railway crossing when a train is approaching, alerts messages are sent to him/her. Otherwise alert messages are not sent to that particular user. That is measured using the location tracking. Then the vehicle movement patterns are collected of that particular user and they are stored in the real-time database. Then the patterns which are in the database are analysed and give alerts to the particular user form the analysed patterns via the app (Online Method) and the SIM card (Offline Method) as well.

6.2 TOOLS AND TECHNOLOGIES

- **Model Building** – Python will be used for model building as the back-end of the system.
- **Mobile App Development** - Flutter is used to create the front-end thus facilitating a cross-platform mobile application.
- **Data Storage** – Firebase real-time database
- **Version Control System** – GitLab
- **IDE** - Visual Studio Code 1 (VS Code) and Android Studio
- **Machine Learning Techniques –** Linear Regression, Decision Tree, SVM

7. RESULTS & DISCUSSION

The Train Detection and Alerting System was implemented and evaluated to assess its performance in detecting trains and providing real-time alerts to both vehicle and pedestrian users to ensure the security when crossing the railway crossing. The system was deployed in a test environment with a simulated railway crossing scenario.

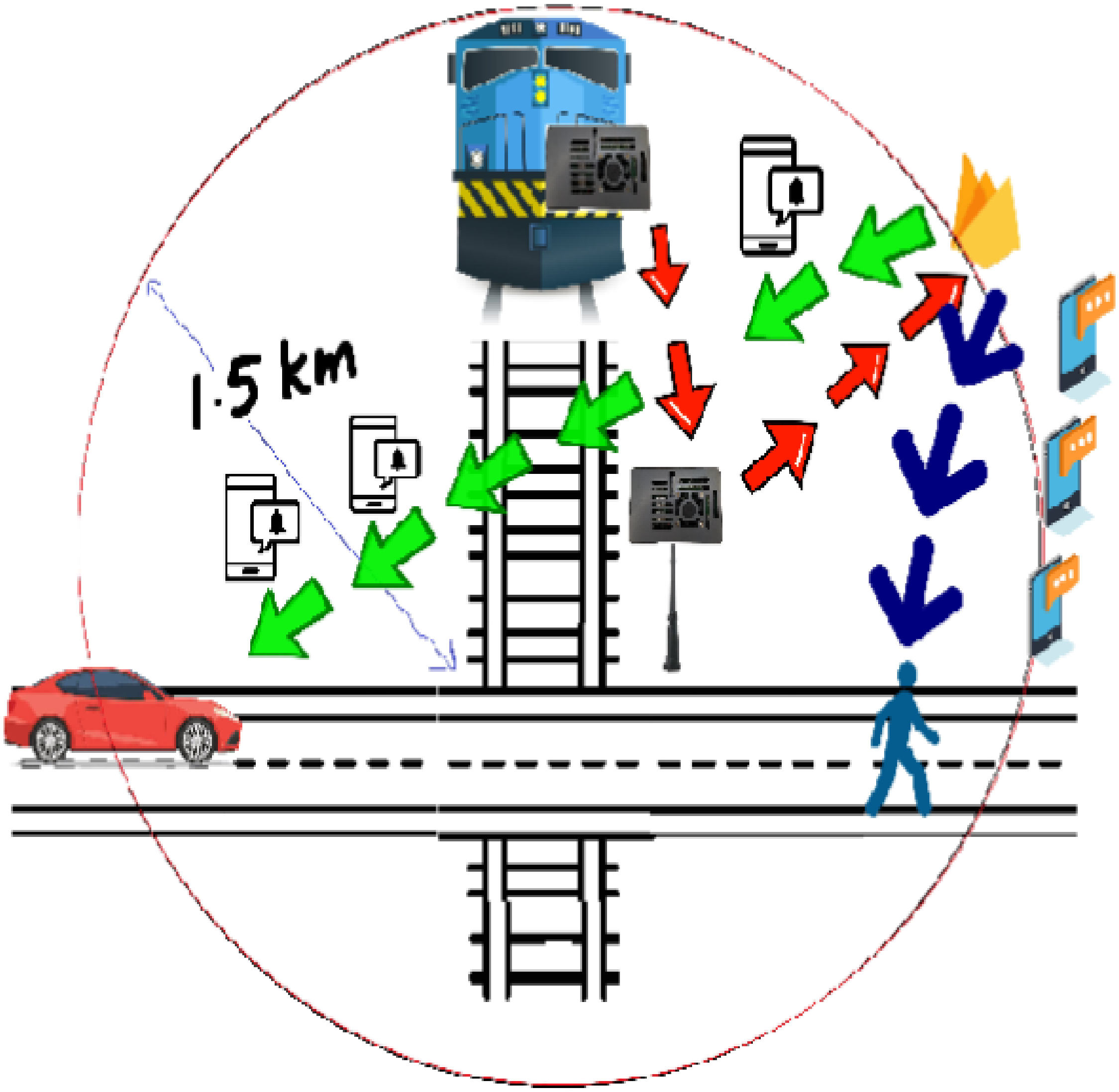


Figure 1: Overview of the System

Figure 1 shows the basic overview of the system. The system incorporates a IoT device with tracking device installed on the train, which transmits its precise location to the IoT device near the railway crossing. Furthermore, the IoT device near the railway crossing is triggered when the train approaches a designated radius near a railway crossing. All the data are transmitted to the cloud database system (Firebase real-time database) from the IoT device near the railway crossing. When a vehicle user who installed our app, is driving towards a railway crossing side and a train is approaching on that particular time as well, alert messages are sent to their mobile app indicating that a railway crossing is approaching and a train is approaching nearby as well, if they are inside the range circle. Alert messages are sent as SMS messages to the pedestrians who are near the railway crossing if they are inside the range circle.

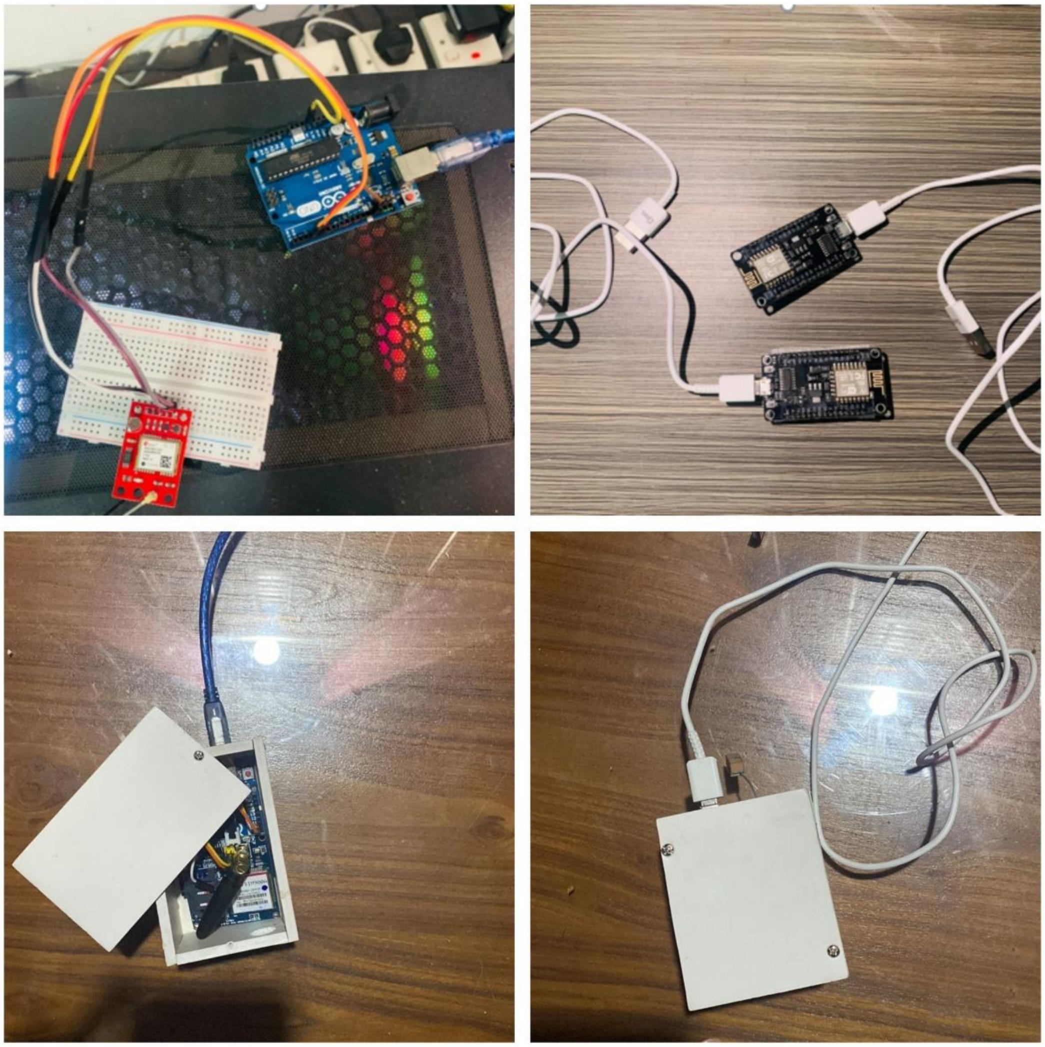


Figure 2: IoT Devices

Figure 2 shows the IoT devices which are used in our system. One IoT device is placed in the train and other IoT device is placed near the railway crossing. The IoT device on the train includes a Arduino Uno board and SIM 900 GSM Module which is connected to the Arduino Uno board.

SIM 900 GSM Module is used to track the location of the train. The IoT device the railway crossing includes a Node MCU board and a speaker. Speaker is used to notify the people who are nearby the railway crossing when a train is approaching.

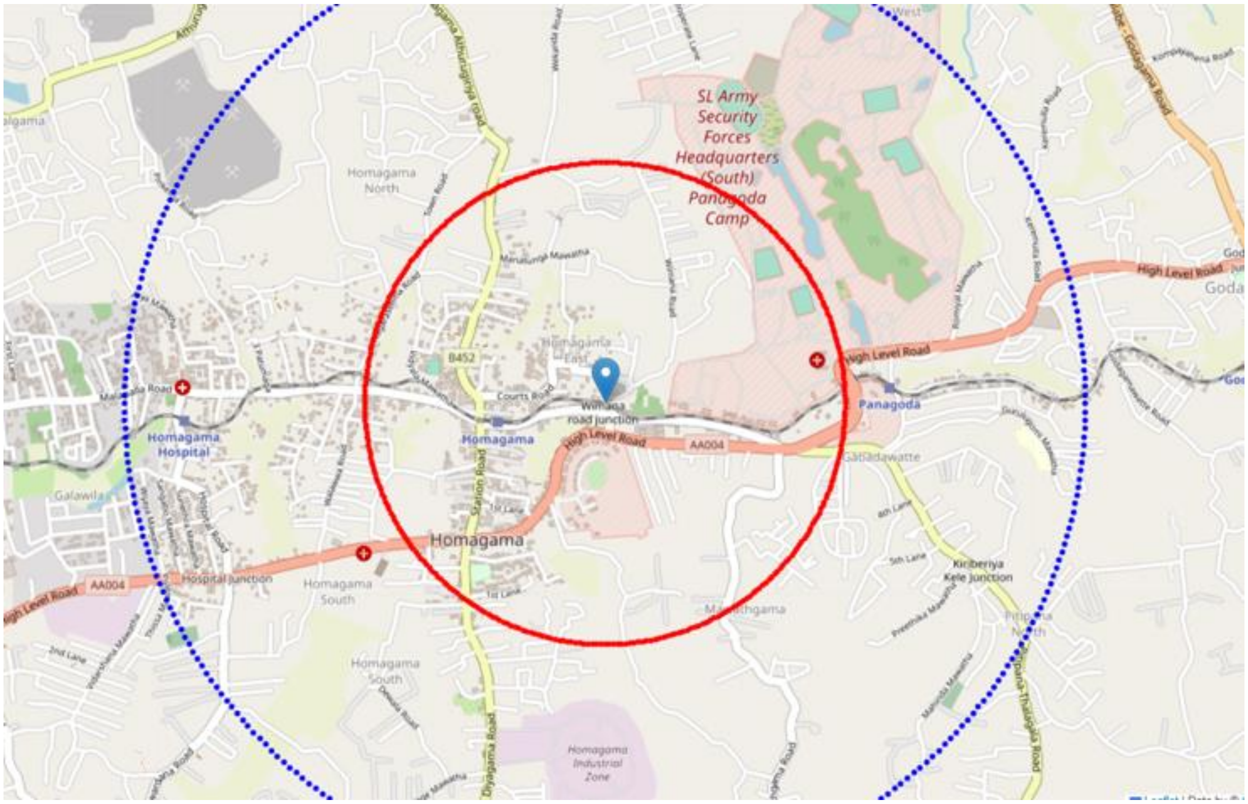


Figure 3: Range Circle on the Map

Figure 3 shows the range circles in the map. In this figure, the co-ordinates of the ‘Homagama’ railway crossing is used as the central point since it is a crowded railway crossing. There are 2 range circles are drawn around the ‘Homagama’ railway crossing. The red circle is 1km range circle and the blue circle is 2km range circle.

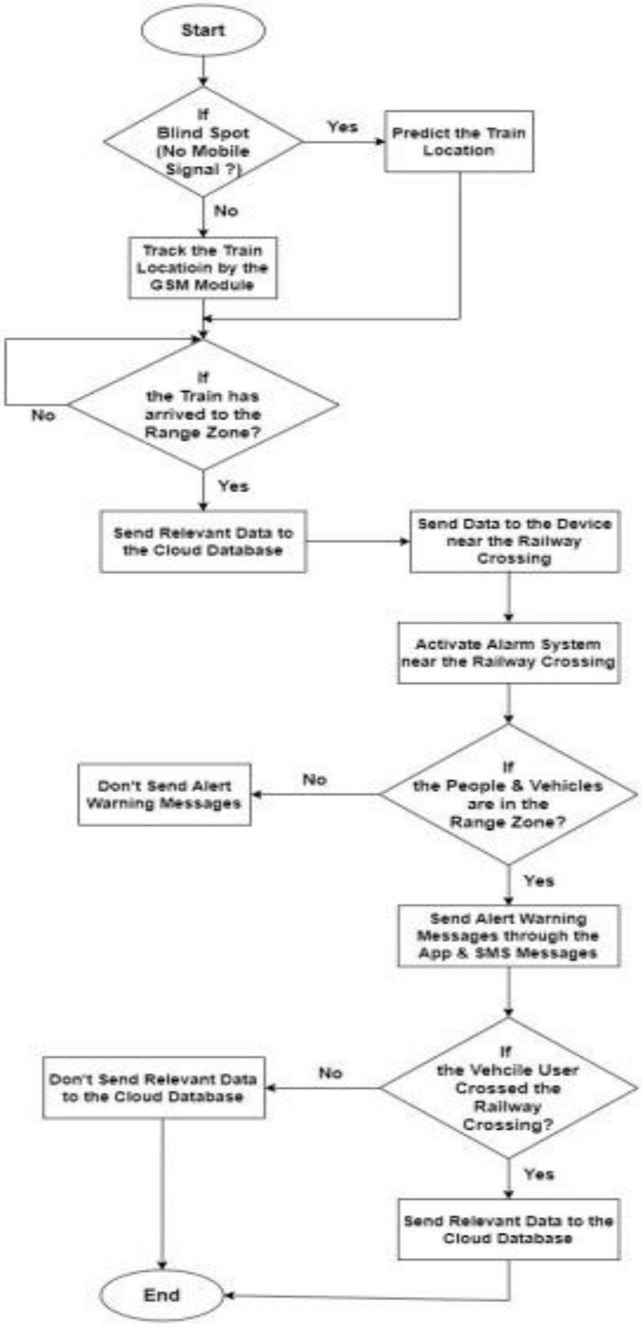
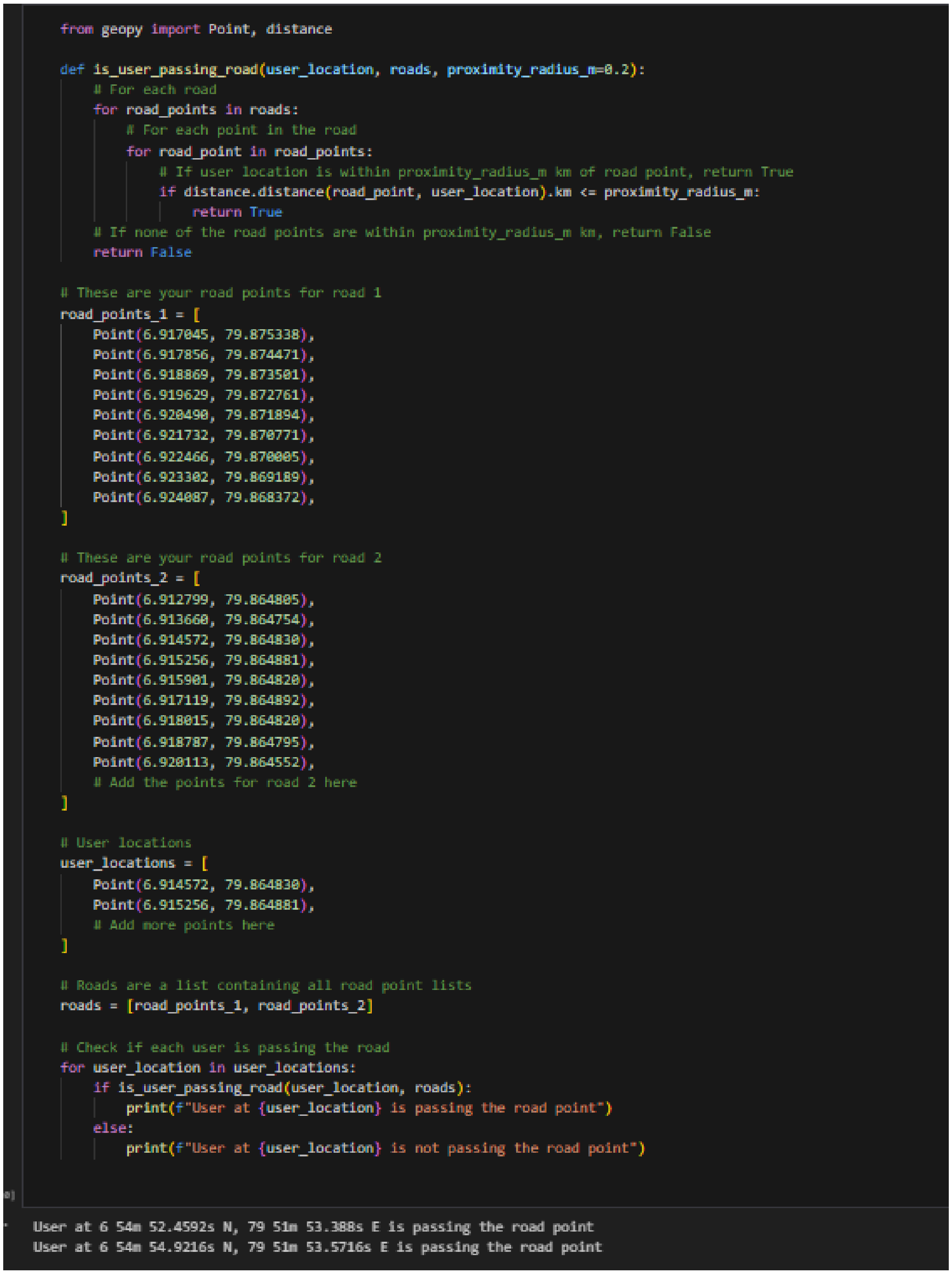


Figure 4: Flowchart of the System

Figure 4 shows the basic flowchart of the system. The flowchart is a useful tool for understanding the process of a railway crossing safety system.

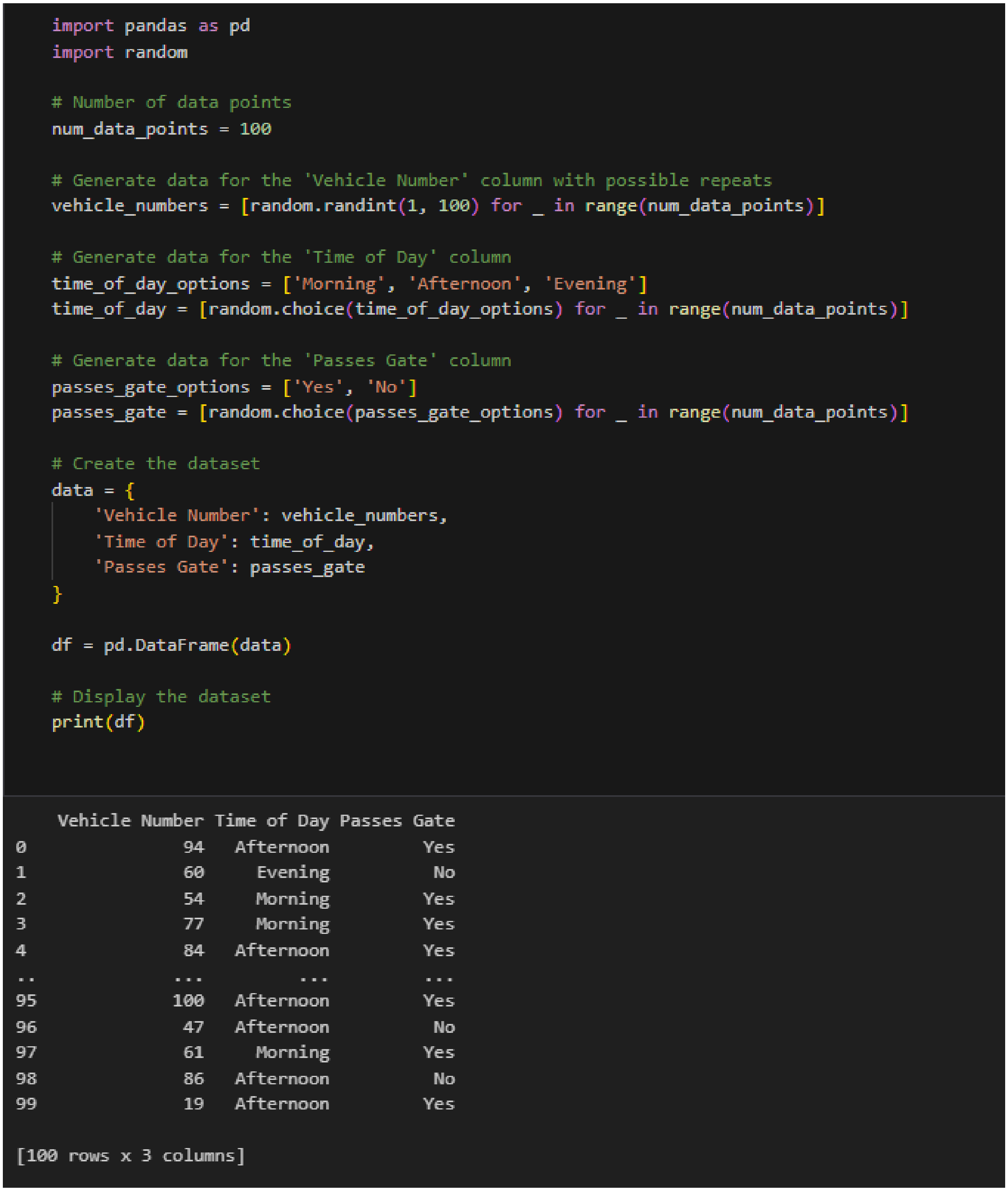


```python
from geopy import Point, distance

def is_user_passing_road(user_location, roads, proximity_radius_m=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(6.917045, 79.875338),
    Point(6.917856, 79.874471),
    Point(6.918869, 79.873501),
    Point(6.919629, 79.872761),
    Point(6.920490, 79.871894),
    Point(6.921732, 79.870771),
    Point(6.922466, 79.870005),
    Point(6.923302, 79.869189),
    Point(6.924087, 79.868372),
]

# These are your road points for road 2
road_points_2 = [
    Point(6.912799, 79.864805),
    Point(6.913660, 79.864754),
    Point(6.914572, 79.864830),
    Point(6.915256, 79.864881),
    Point(6.915901, 79.864820),
    Point(6.917119, 79.864892),
    Point(6.918015, 79.864820),
    Point(6.918787, 79.864795),
    Point(6.920113, 79.864552),
    # Add the points for road 2 here
]

# 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")
```

```
User at 6 54m 52.4592s N, 79 51m 53.388s E is passing the road point
User at 6 54m 54.9216s N, 79 51m 53.5716s E is passing the road point
```

Figure 5: Code of defining the Roads

Figure 5 shows the Python code of defining the roads near the railway crossing. In here, just 2 roads are defined. The 1st road is named as ‘road_points_1’ and the 2nd road is named as ‘road_points_2’. The co-ordinates of the particular roads are defined as points. The vehicle user locations are received from the vehicle user who is using the mobile app. The ladder part of the code checks whether the vehicle is moving in a defined road consistently from the current location co-ordinates from the vehicle. The python output shows that the vehicle is moving in a defined road.

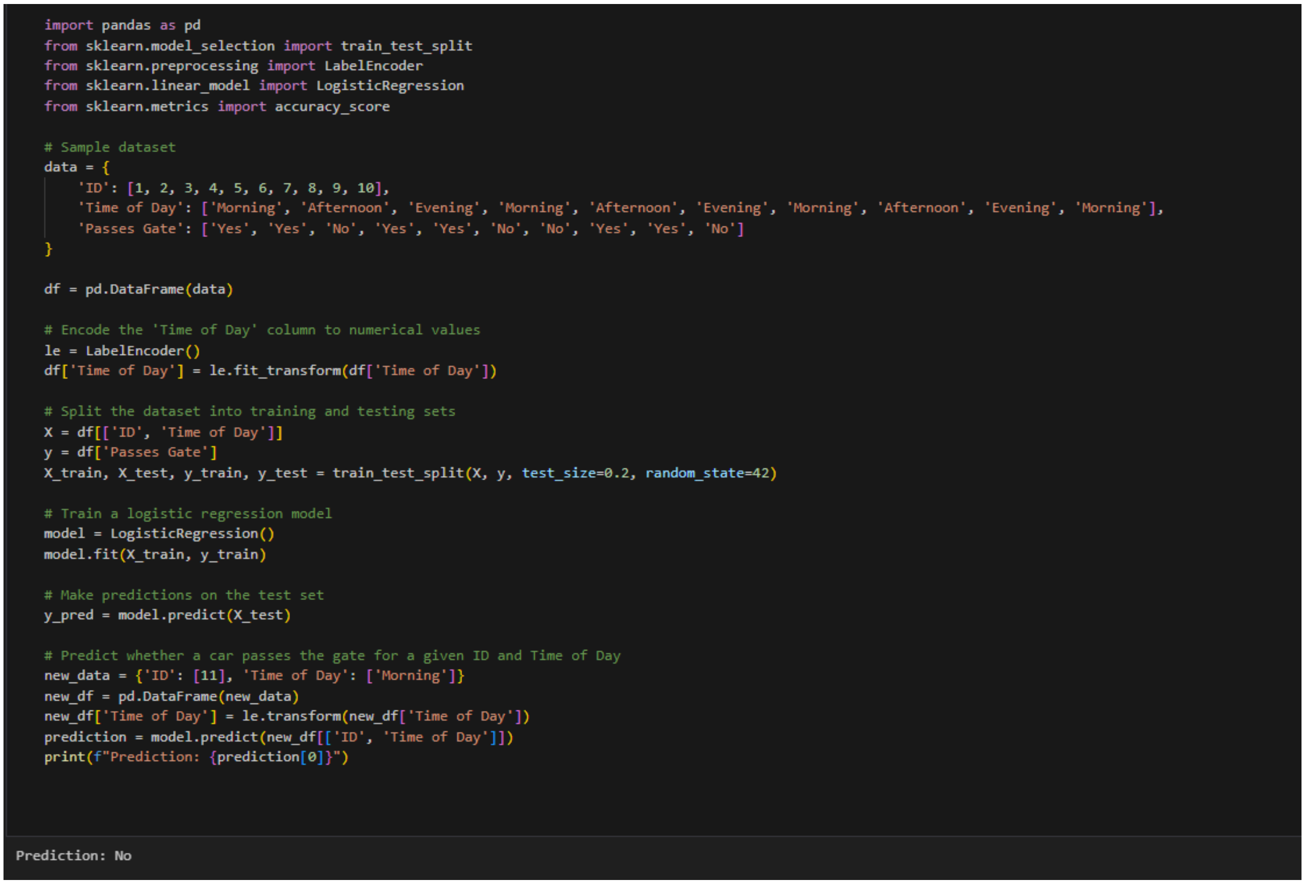


```python
import pandas as pd
import random

# Number of data points
num_data_points = 100

# Generate data for the 'Vehicle Number' column with possible repeats
vehicle_numbers = [random.randint(1, 100) for _ in range(num_data_points)]

# Generate data for the 'Time of Day' column
time_of_day_options = ['Morning', 'Afternoon', 'Evening']
time_of_day = [random.choice(time_of_day_options) for _ in range(num_data_points)]

# Generate data for the 'Passes Gate' column
passes_gate_options = ['Yes', 'No']
passes_gate = [random.choice(passes_gate_options) for _ in range(num_data_points)]

# Create the dataset
data = {
    'Vehicle Number': vehicle_numbers,
    'Time of Day': time_of_day,
    'Passes Gate': passes_gate
}

df = pd.DataFrame(data)

# Display the dataset
print(df)
```

```
    Vehicle Number Time of Day Passes Gate
0               94   Afternoon         Yes
1               60     Evening          No
2               54     Morning         Yes
3               77     Morning         Yes
4               84   Afternoon         Yes
..             ...         ...         ...
95             100   Afternoon         Yes
96              47   Afternoon          No
97              61     Morning         Yes
98              86   Afternoon          No
99              19   Afternoon         Yes

[100 rows x 3 columns]
```

Figure 6: Checking whether the vehicles passes the railway crossing

Figure 6 shows the Python code of checking whether the vehicle passes the railway crossing or not in a particular time. Vehicle number is generated randomly from number 1 to 100. The time of the day has 3 options, ‘Morning’, ‘Afternoon’ and ‘Evening’. Then a dataset is created from the vehicle number, time of the day and the passes gate. The python output shows whether vehicles are passing the railway crossing or not on a defined time of the day.



```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score

# Sample dataset
data = {
    'ID': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
    'Time of Day': ['Morning', 'Afternoon', 'Evening', 'Morning', 'Afternoon', 'Evening', 'Morning', 'Afternoon', 'Evening', 'Morning'],
    'Passes Gate': ['Yes', 'Yes', 'No', 'Yes', 'Yes', 'No', 'No', 'Yes', 'Yes', 'No']
}

df = pd.DataFrame(data)

# Encode the 'Time of Day' column to numerical values
le = LabelEncoder()
df['Time of Day'] = le.fit_transform(df['Time of Day'])

# Split the dataset into training and testing sets
X = df[['ID', 'Time of Day']]
y = df['Passes Gate']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train a logistic regression model
model = LogisticRegression()
model.fit(X_train, y_train)

# Make predictions on the test set
y_pred = model.predict(X_test)

# Predict whether a car passes the gate for a given ID and Time of Day
new_data = {'ID': [11], 'Time of Day': ['Morning']}
new_df = pd.DataFrame(new_data)
new_df['Time of Day'] = le.transform(new_df['Time of Day'])
prediction = model.predict(new_df[['ID', 'Time of Day']])
print(f"Prediction: {prediction[0]}")
```

```
Prediction: No
```

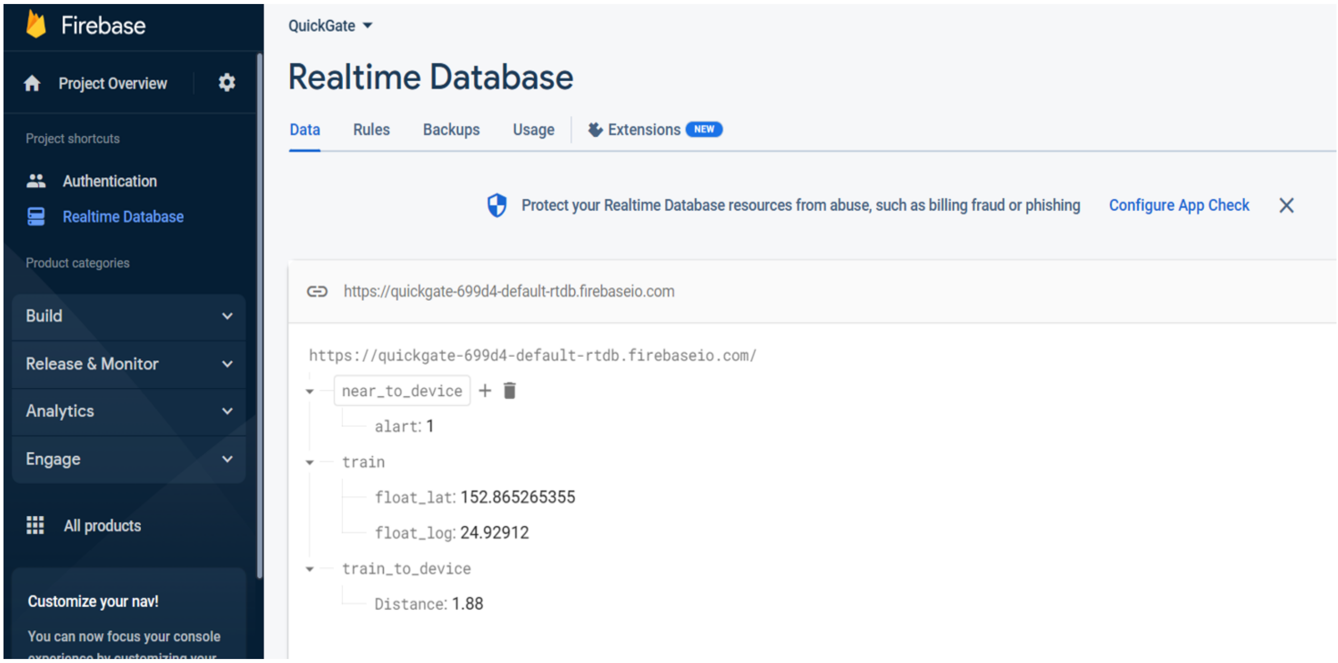
Figure 7: Predicting whether a vehicle crosses the railway crossing or not in a particular time

Figure 7 shows the python code of prediction of whether a vehicle crosses the railway crossing or not in a particular time. This is the main objective of the component. ‘Linear Regression’ model is used to do the prediction since it gave the highest prediction accuracy rate. The ‘Linear Regression’ model gave a prediction accuracy about 75% - 80% while other models like ‘Decision Tree’, ‘Random Forest’ gave the prediction percentage less than 75%. The percentage of the accuracy is low than expected from the ‘Linear Regression’ model as well since the data is always updating. The code splits the dataset into training and testing sets. The train_test_split() function from the sklearn.model_selection module is used to do this. The test_size parameter is set to 0.2, which means that 20% of the data will be used for testing and the remaining 80% will be used for training.

The LogisticRegression() class from the sklearn.linear_model module is used to do this. The model is trained on the training set. The code then makes predictions on the testing set. The predict() method of the LogisticRegression() class is used to do this.

Finally, the code predicts whether a car passes the gate for a given ID and Time of Day. A new dataset is created with the given ID and Time of Day. The Time of Day column is encoded to numerical values. The predict() method of the LogisticRegression() class is then used to make a prediction.

The output of the code is the prediction, which is either Yes or No.



Customize your nav!

You can now focus your console

Realtime Database

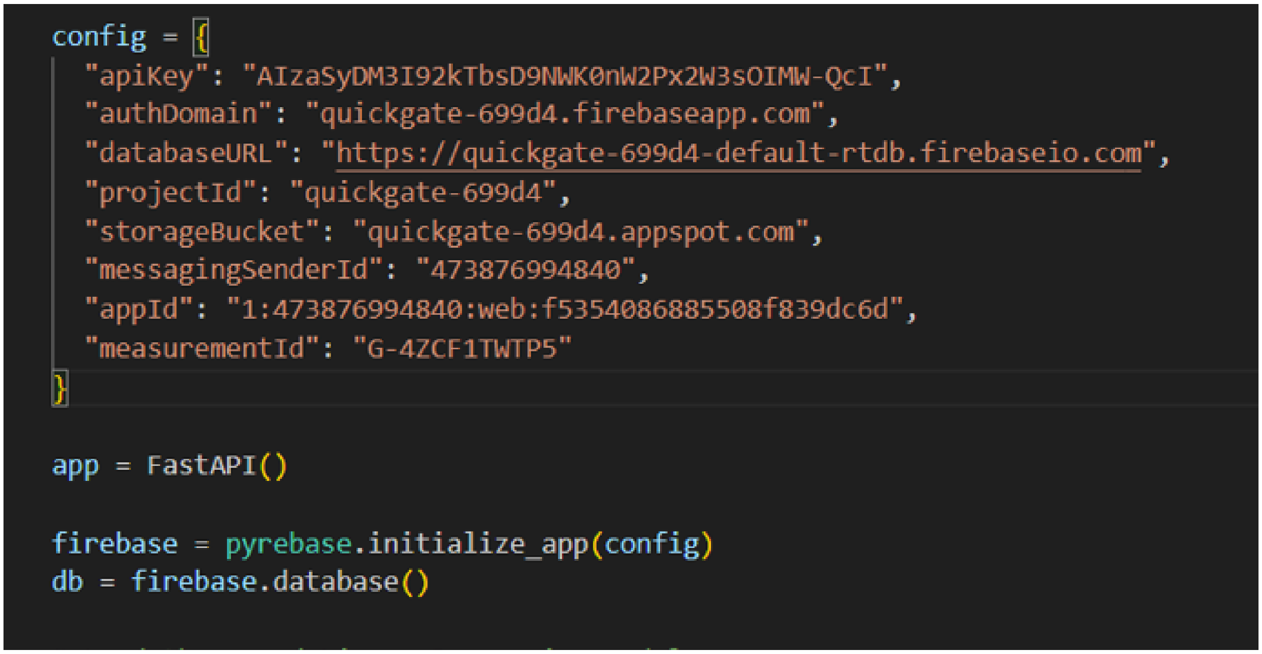
Protect your Realtime Database resources from abuse, such as billing fraud or phishing

https://quickgate-699d4-default-rtdb.firebaseio.com

```
https://quickgate-699d4-default-rtdb.firebaseio.com/
▾ near_to_device
    alart: 1
▾ train
    float_lat: 152.865265355
    float_log: 24.92912
▾ train_to_device
    Distance: 1.88
```

Figure 8: Data retrieved to Firebase Database

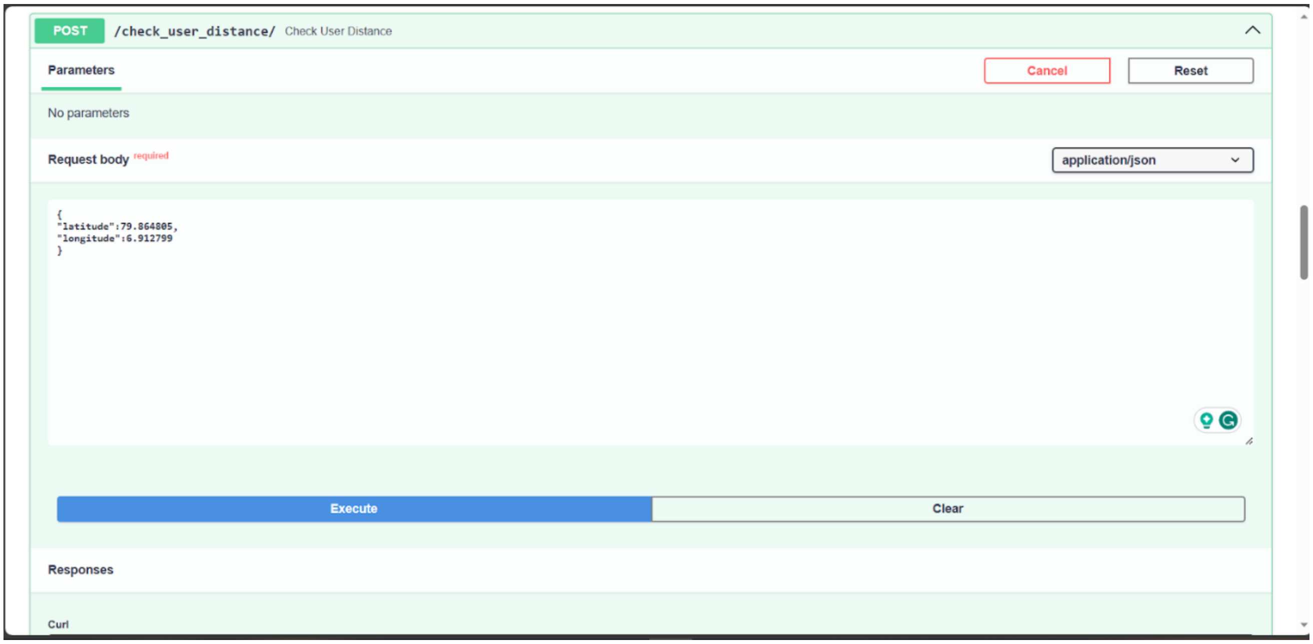
Figure 8 shows that the data is retrieved to the Firebase real-time database. In here, ‘near_to_device’, ‘alert’: 1 means the train is approaching towards the railway crossing. The ‘alert’: 0, when a train is not approaching towards the railway crossing. ‘train’, displays the current location co-ordinates of the train as longitude and latitude. The ‘train_to_device’, ‘Distance’ shows the distance between the railway crossing and the train.



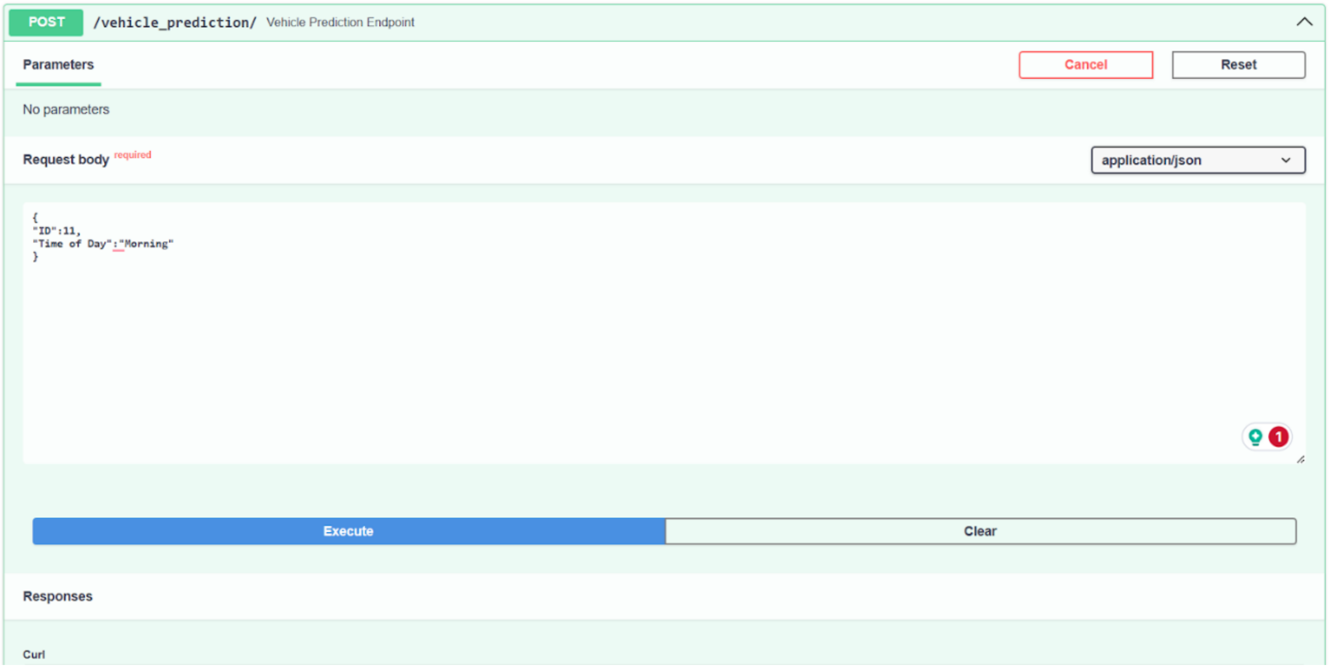
```
config = {
    "apiKey": "AIzaSyDM3I92kTbsD9NWK0nW2Px2W3sOIMW-QcI",
    "authDomain": "quickgate-699d4.firebaseapp.com",
    "databaseURL": "https://quickgate-699d4-default-rtdb.firebaseio.com",
    "projectId": "quickgate-699d4",
    "storageBucket": "quickgate-699d4.appspot.com",
    "messagingSenderId": "473876994840",
    "appId": "1:473876994840:web:f5354086885508f839dc6d",
    "measurementId": "G-4ZCF1TWTP5"
}

app = FastAPI()

firebase = pyrebase.initialize_app(config)
db = firebase.database()
```

Figure 9: Connecting Firebase with the Backend

Figure 9 shows the code to connect the backend with the Firebase cloud database.



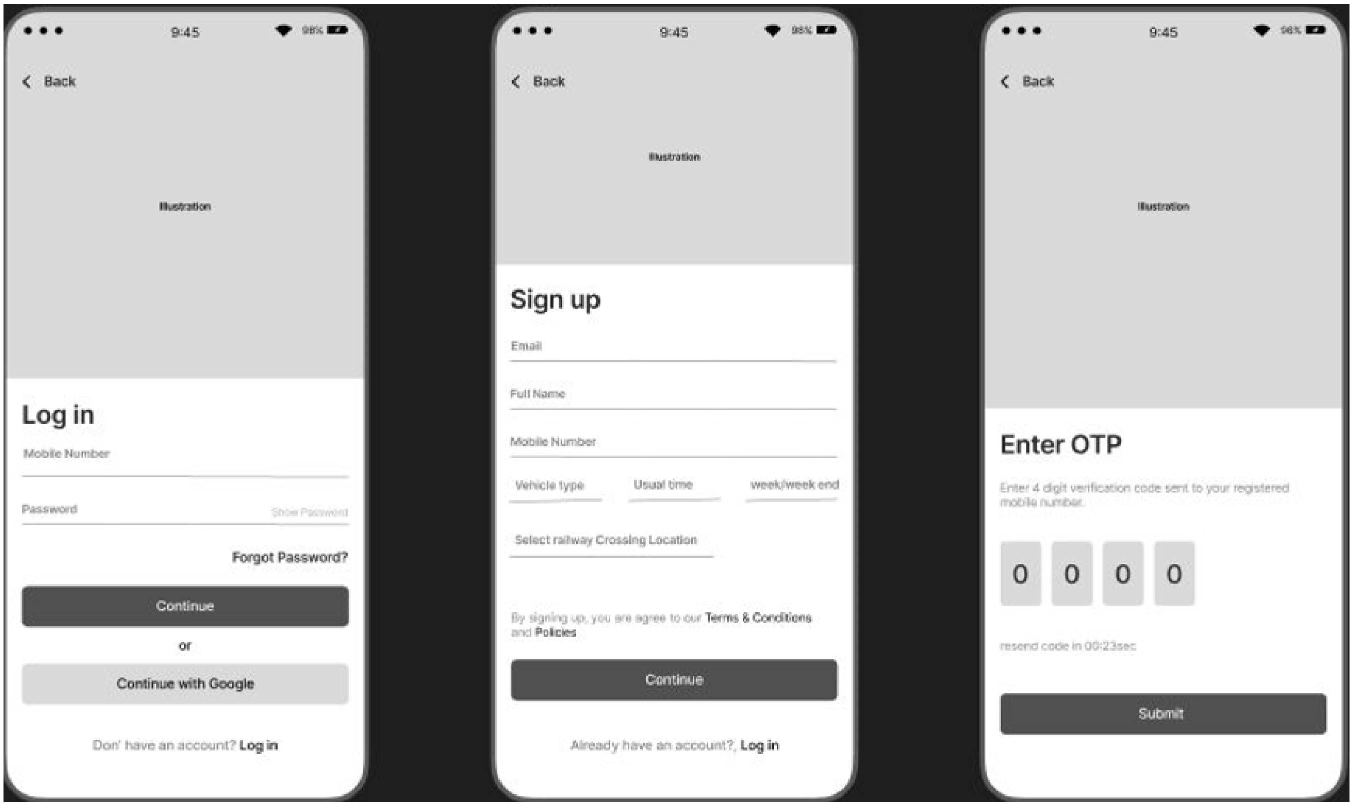
```
{
"latitude":79.864805,
"longitude":6.912799
}
```



```
{
"ID":11,
"Time of Day":"Morning"
}
```

Figure 10: Output display using API

In Figure 9, the prediction output of likelihood of a vehicle crossing the railway can be seen using an API called ‘FastAPI’. The vehicle ID and the time of the day can be typed and the API shows the predicted output of whether that vehicle with the ID is crossing the railway or not on the particular time of the day.



Don' have an account? Log in

By signing up, you are agree to our Terms & Conditions and Policies

Already have an account?, Log in

Enter 4 digit verification code sent to your registered mobile number.

resend code in 00:23sec

Figure 11: User Interfaces of the App

Figure 11 shows some user interfaces of the mobile application. The user should allow location permission when using the app since the location of the vehicle user is tracked by the system. The user can log into the application if he/she is already registered to the app. If the user is not registered to the app, he/she should register to the app by providing necessary details. Since the location permission is granted form user when using the app, the vehicle user locations are always tracked by the system to take decisions and alert the user.

7.1. Implementation Process

The implementation of the Train Detection and Alerting System involved several steps, including:

System Design:

The system design phase focused on defining the architecture, components, and functionalities of the system. Various technologies, such as IoT devices, SIM cards, and Integration of GPS tracking facilitated real-time train detection and alerting. The vehicle ID and the time of the day can be typed and the API shows the predicted output of whether that vehicle with the ID is crossing the railway or not on the particular time of the day.

Development Phase: During the development phase, the system components were coded and programmed. The mobile application interface was constructed using the Flutter framework, whereas backend development employed Python and Java. The system was devised to interface with IoT devices for the collection of train location data and the dissemination of user alerts.

Integration and Testing Phase: Following the completion of individual component development, integration and testing procedures were carried out a unified system. Integration testing involved verifying the proper functioning of different modules and ensuring seamless communication between the mobile application, IoT devices, and backend servers. Extensive testing was conducted to ensure the accuracy and reliability of train detection and alerting.

7.2. Strengths and Weaknesses

In the course of the implementation phase, diverse strengths and weaknesses of the Train

Detection and Alerting System have been identified. Noteworthy strengths encompass the system's proficient achievement of real-time train detection, facilitating the swift issuance of user alerts. Additionally, the mobile application interface is distinguished by its user-friendly attributes, thereby enhancing the overall user experience. Moreover, the system demonstrates adept

integration with Internet of Things (IoT) devices, effectively harnessing them for the acquisition of precise train location data and subsequently ensuring accurate dissemination of alerts.

In contrast, several weaknesses have been discerned. The system exhibits limited coverage, as its operational scope is currently confined to a radius of 1.5 km. This constraint may impede its efficacy within expansive railway networks. Furthermore, a notable weakness lies in the system's dependency on the functionality and availability of IoT infrastructure, thereby potentially introducing vulnerabilities to its operation.

7.3. Future Improvements

In light of insights derived from the implementation process and the assessment of strengths and weaknesses, several potential avenues for enhancing the system's capabilities come to the fore. One such avenue involves the expansion of the coverage region. The system's current coverage, limited to a radius of 1.5 km, prompts consideration of a broader geographical span to better serve extensive railway networks.

Another prospect pertains to scalability improvement. As user numbers grow and railway intersection points become more intricate, strategies should be devised to fortify the system's scalability, ensuring optimal performance in the face of increasing demands.

Furthermore, enhancing fault tolerance emerges as a crucial goal. By incorporating redundancy mechanisms and backup systems, the system can be made resilient against potential failures of IoT devices or disruptions in connectivity, thereby maintaining continuous and reliable operation.

8. SYSTEM REQUIREMENTS

8.1. User Requirements:

- User should have a smart phone.
- Mobile Application should be installed to the smart phone.
- Mobile phone should be switched on.
- Capability of understanding simple English.
- Guidance if a user is unable to use the application alone.
- Message alerts should be viewed regularly.
- App notifications should be viewed regularly.
- User registration details should be provided correctly.

8.2. Functional Requirements:

- **Providing accurate predictions** – The component should be able to provide accurate prediction about the vehicles likelihood of crossing the railway crossing with higher accuracy.

- **Vehicles (Users) identification -** The component should be able to identify the vehicles that are moving towards the railway crossing.

- **User Location Tracking** - The component should be able to track the location of the user's vehicle when they enter the range circle near the railway crossing.

- **Integration with the App** - The component should be integrated with the existing mobile application for the railway crossing safety system.

- **Real-time updates** - The component should be able to provide real-time updates about the location of incoming train and the railway crossings ahead in the road to the user.

- **Vehicle movement pattern storage** - The component should be able to store the tracked patterns of the user's vehicle.

- **Alert Delivery** - The component should be able to send alerts only to the app users who are approaching the railway crossing.

- **Notification Delivery -** The component should be able to send a notification to the user's SIM card if they are predicted to cross the railway crossing on a particular day, even if they are not using the app or have mobile data turned off.

8.3. Non- Functional Requirements:

- **Ease of Usability:** A system that is easy to use will be more likely to be used by users, which can lead to increased productivity and satisfaction.
- **High performance:** A system that performs well will be more responsive and efficient, which can lead to a better user experience.
- **Availability**: A system that is available when users need it can help to avoid downtime and lost productivity.
- **Reliability:** A reliable system is less likely to fail, which can help to protect data and avoid financial losses.
- **Security:** A secure system can help to protect sensitive data from unauthorized access, which can help to avoid legal problems and financial losses.
- **Performance:** The system should meet certain performance requirements, such as response time, throughput, and scalability.
- **Reliability:** The system should be reliable and should not fail frequently.
- **Security:** The system should be secure and should protect sensitive data from unauthorized access.
- **Usability:** The system should be easy to use and should have a good user interface.
- **Maintainability:** The system should be easy to maintain and update.
- **Portability:** The system should be portable and should be able to run on multiple platforms.
- **Scalability:** The system should be scalable and should be able to handle increasing loads.
- **Interoperability:** The system should be interoperable and should be able to communicate with other systems.
- **Testability:** The system should be testable and should have good testability features.
- **Documentation:** The system should be well-documented and should have good documentation features.

- **Simple UI / UX designs**: Simple UI/UX designs can make the system easier to learn and use, which can also lead to increased productivity and satisfaction.

9. COMMERCIALIZATION

Identifying the target Audience: Since our ultimate goal of this project is to reduce the number of collisions happen in the railway crossings in Sri Lanka, we are targeting this system to whole people in Sri Lanka, there is no limitation.

Social media marketing: Social media platforms such as Facebook, Twitter, Instagram, YouTube and LinkedIn offer a cost-effective way to promote our product. By creating and sharing engaging content on social media, researchers can reach a wider audience and increase the visibility of their research. Social media can also be used to engage with potential collaborators and industry partners.

Partnership with a reputed company: Partnerships and collaborations with industry partners and other research institutions can help to commercialize research findings. By partnering with organizations that can provide resources, funding, or expertise, researchers can accelerate the commercialization process and increase the impact of the research. As the research project team, we expect to commercialize our final output product by building some partnership with well reputed company.

Attending Award Competitions: Attending conferences and events related to the research topic can provide opportunities to network with potential collaborators and industry partners. Researchers can also use these events to present their research findings and gain valuable feedback from peers and experts in the field.

10. CONCLUSION

In this paper, we discussed the methodology for creating a proposed train detection and alert system. The system uses sensors, cameras, and machine learning algorithms to detect trains and alert people when they approach a railway crossing. We developed a secure and user-friendly UI to display real-time information about the status of the crossing. We tested the system in a controlled environment and in the real world, and made modifications based on feedback from users. The proposed system has the potential to improve the safety of people at railway crossings and reduce the number of accidents and incidents.

11.REFERENCES

[1] N. News, "NBC News," 27 April 2005. [Online]. Available: https://www.nbcnews.com/id/wbna7647780.

[2] N. A. Kulasingham Ragulan, "A STUDY ON RAILWAY-ROADWAY LEVEL CROSSING SAFETY," R4TLI, 2021.

[3] N. Fazlulhaq, "Sunday TImes," 12 June 2011. [Online]. Available: https://www.sundaytimes.lk/110612/News/nws_23.html#:~:text=According%20to%20the%20Railways%20Department,from%20Colombo%20Fort%20to%20Galle..

[4] S. L. Tweet, "Twitter," 25 May 2019. [Online]. Available: https://twitter.com/SriLankaTweet/status/1132093011997863936.

[5] A. A. T. P. D. Silva, "Smart Train Tracker," 2020. [Online]. Available: https://www.bing.com/ck/a?!&&p=770890b59084ce11JmltdHM9MTY5NDIxNzYwMCZpZ3VpZD0yYWRkYmRhNy0wZTA3LTZiNjgtMDc0OS1hZjVkMGYwNjZhNGImaW5zaWQ9NTE3Mw&ptn=3&hsh=3&fclid=2addbda7-0e07-6b68-0749-af5d0f066a4b&psq=A.+A.+T.+P.+De+Silva%2c+%22Smart+Train+Tracker%2c%22+U.

[6] T. K. M. I. M. Amjath, "An Automated Railway Level Crossing System," 2020.

[7] S. K. Ž. S. G. T. V. P. Aleksandar Blagojević, "Evaluation of Safety Degree at Railway Crossings in Order to Achieve Sustainable Traffic Management," *MDPI,* no. 15 January 2021, 2021.

[8] T. R. B. -. N. R. Council, "Traffic Signal Operations Near Highway-Rail Grade Crossings".

[9] Y. K. A. Akinwumi Abimbola Amusan, "An Automatic Railway Level Crossing System with Crack Detection," *IEEE,* 2022.

[10] M. A. H. A. A. L. A. Khandakar Rabbi Ahmed, "A Secure Automated Level Crossing and Train Detection System for Bangladesh Railway," *IEEE,* 2022.

[11] G. Singh, R. K. M. S. S. Praveen Kumar and K. Singh, "Security System for Railway Crossings using Machine Learning," *IEEE,* 2020.

[12] RDMNS, "RDMNS," [Online]. Available: https://rdmns.lk/.

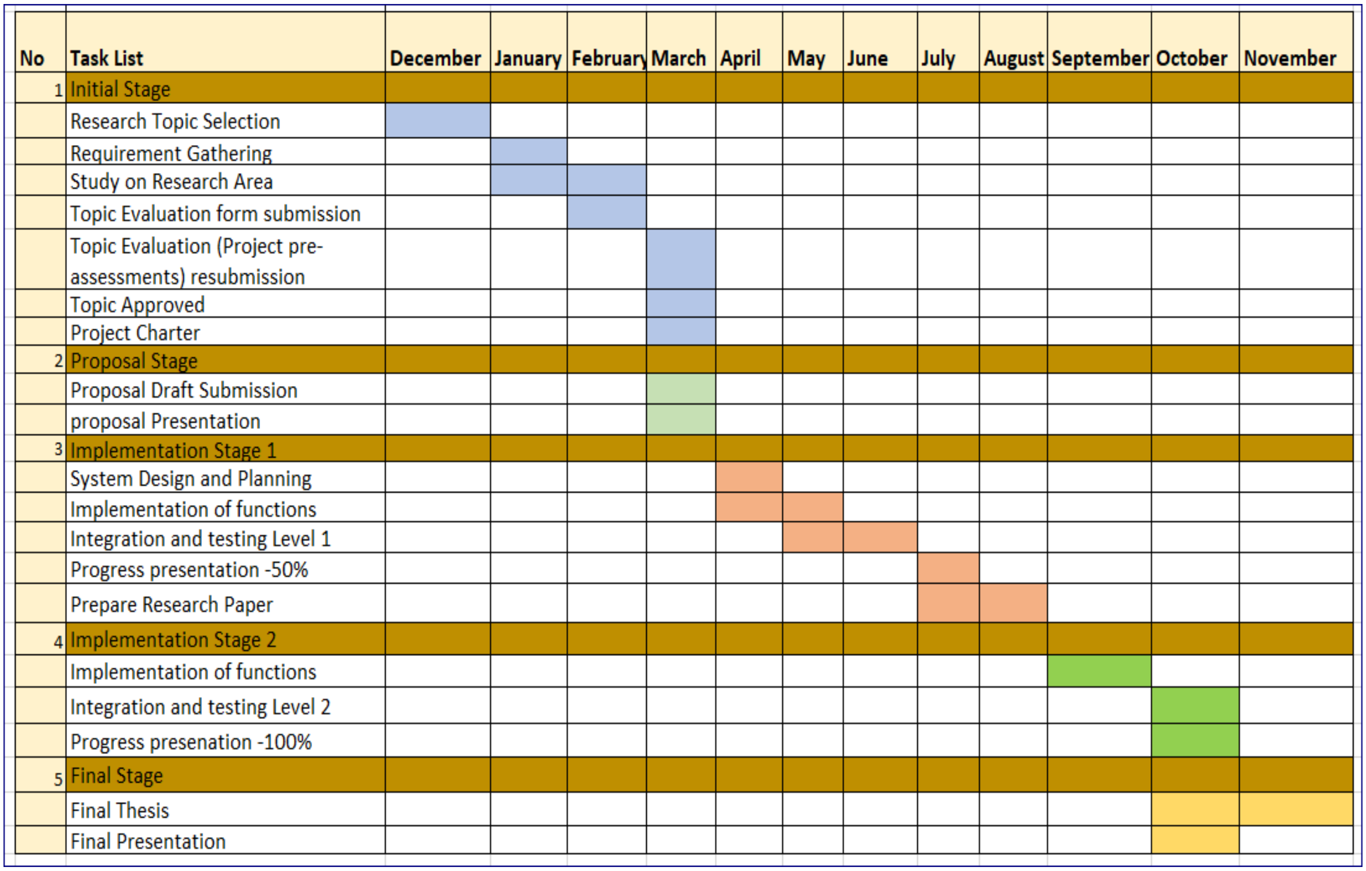
[13] N. C. Brintha and L. A. B. P. K. R. M. T. R. Chakka Venkata Pavan Tarun, "Smart Railway Crossing Surveillance System," *IEEE,* 2022.

[14] D. N. Suwaris, "Smart Railway Route Sensor Grid for wild life area and unsafe railway crossings in Sri Lanka," [Online]. Available: https://www.academia.edu/68922253/Smart_Railway_Route_Sensor_Grid_for_wild_life_area_and_unsafe_railway_crossings_in_Sri_Lanka.

[15] Wikipedia, "Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Polgahawela_level_crossing_accident#:~:text=Polgahawela%20lev.

APPENDICES

Appendix A: Gannt 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											🟨	

Figure 1 - Gannt Chart

Appendix B: Work Break Down Chart

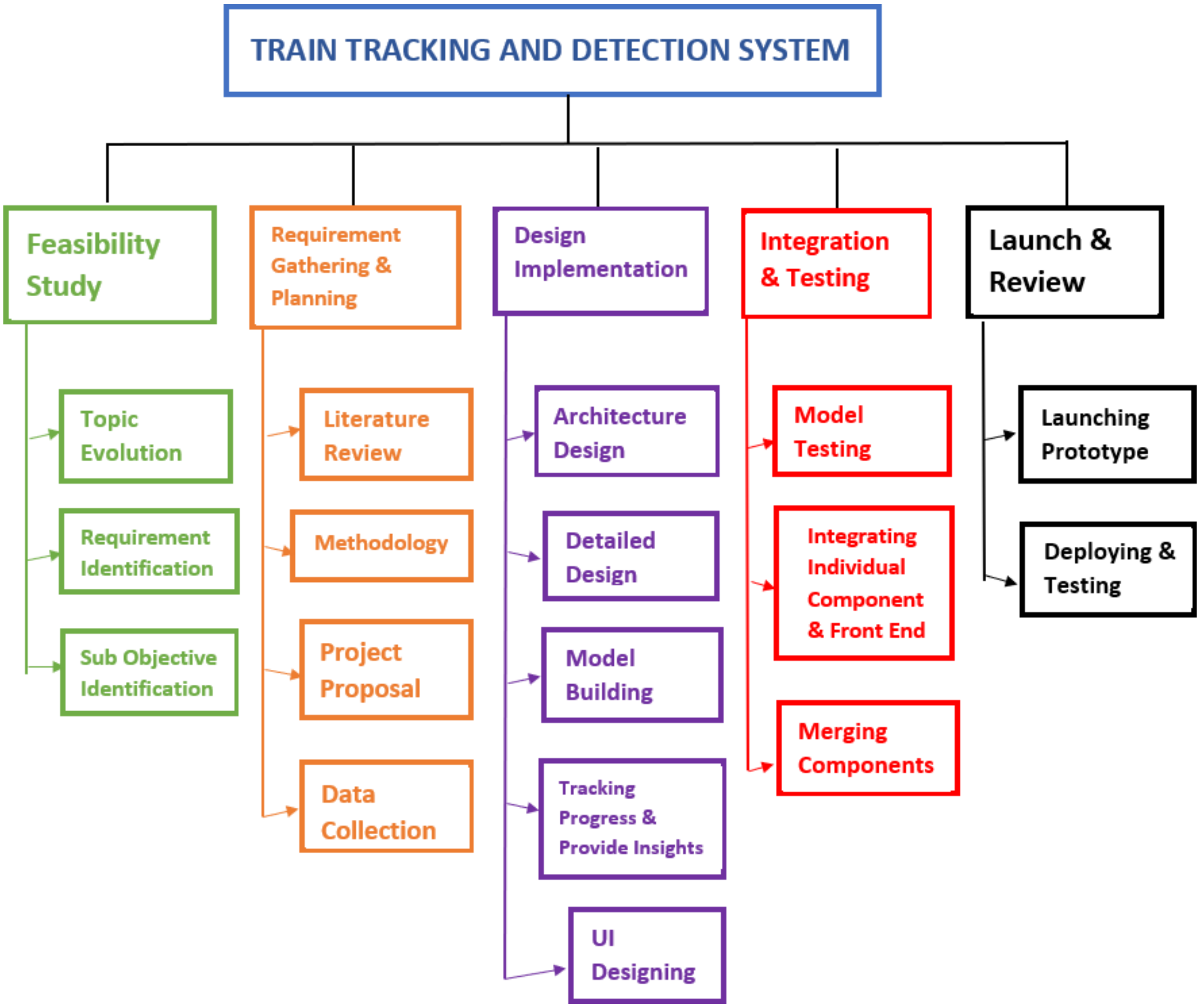
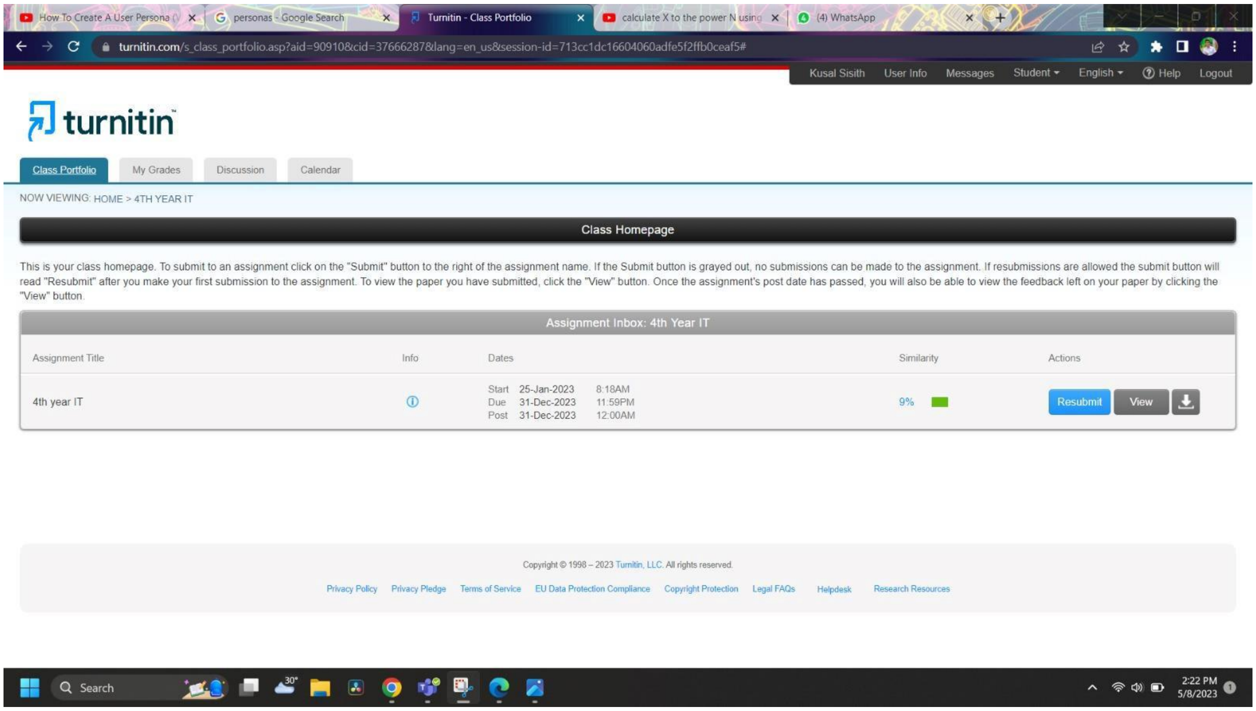


Figure 2: Work Break Down chart

Turnitin Report Screenshot



How To Create A User Persona (
personas - Google Search
Turnitin - Class Portfolio
calculate X to the power N using
(4) WhatsApp
turnitin.com/s_class_portfolio.asp?aid=90910&cid=37662287&lang=en_us&session-id=713cc1dc16604060adfe5f2ffb0ceaf5#
Kusal Sisith
User Info
Messages
Student
English
Help
Logout
turnitin
Class Portfolio
My Grades
Discussion
Calendar

NOW VIEWING: HOME > 4TH YEAR IT

Class Homepage

This is your class homepage. To submit to an assignment click on the "Submit" button to the right of the assignment name. If the Submit button is grayed out, no submissions can be made to the assignment. If resubmissions are allowed the submit button will read "Resubmit" after you make your first submission to the assignment. To view the paper you have submitted, click the "View" button. Once the assignment's post date has passed, you will also be able to view the feedback left on your paper by clicking the "View" button.

Assignment Inbox: 4th Year IT

Assignment Title	Info	Dates	Similarity	Actions
4th year IT	ⓘ	Start 25-Jan-2023 8:18AM Due 31-Dec-2023 11:59PM Post 31-Dec-2023 12:00AM	9% 🟩	Resubmit View

Copyright © 1998 – 2023 Turnitin, LLC. All rights reserved.