International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 06 Issue: 09 | Sep 2019 www.irjet.net p-ISSN: 2395-0072

Vehicle Monitoring System using Internet of Things

Binod Chandra Shrestha[1], Prakash Parajuli[2], Sachin Kafle[3], Prakash Bist[4], Ram Kumar Puri[5]

[1]Deputy Director, Nepal Telecommunication Authority, Kathmandu, Nepal
[2,3,4,5]Advanced College of Engineering and Management, Lalitpur, Nepal

---

Abstract - *The paper presents the use of RFID technology to monitor the vehicle information. The system is an IoT based android application for assistance on vehicle monitoring. The system is developed with the capabilities that allows the toll-booths to record the vehicle information when a vehicle is passed through the toll-booths. The toll-booths consists of the hardware integration of NodeMCU, GPS Module and RFID reader which reads the data from the tag installed on vehicles and send data on cloud, that is continuously monitored by government for various functionalities. Android application allows the government to view the logs of vehicles as they pass through the toll-booths. Government can report the driver if it is an illegal or unauthentic one. It will instantly notify the user about the report status. The government can track any reported vehicle. With the help of illustrative mathematical modeling, the government will have a detailed analytics on vehicle passed through different toll-booths. [14] Users will be able to view their vehicle profile. They will be notified about the tax of their vehicles and also their report status by government. Using the Haversine algorithm [4] users will be able get the nearest check-post distance from their location; so that culprit can be caught easily. Users can view their last checked location; which is implemented with the help of Linear Search Modeling algorithm. [13] Users will be able to get the location assistance on the nearby places. The system uses Firebase for our database services. The system is built as the android application(.apk) over hardware integration and can be used to provide the assistance of ongoing conventional transportation services.*

Key Words: **RFID**, **IoT**, **NodeMCU**, **Haversine algorithm**, **ESP8266**, **GPS Module**.

1. INTRODUCTION

The system emphasizes on using various things to establish a direct communication between the physical environment and the computer network system. Things in our system refers to the RFID (Radio Frequency Identification) sensors that takes the information from the unique RFID tags and transmit them to the cloud with the help of NodeMCU. The obtained data can be used accordingly to provide meaningful information. A GPS module is also used to locate the vehicles that passed through stations (toll-booths). This helps the government and people to locate the vehicles within the country. The system uses this information to provide the various functionalities to both government and people.

1.1 System Architecture

The paper discusses the system that consists of NodeMCU for the controlling process which has been interfaced with the RFID sensors, power supply, mobile application and a Wi-Fi module along with the GPS Module. RFID tag is used to determine the unique id which helps us to gain the vehicle information.

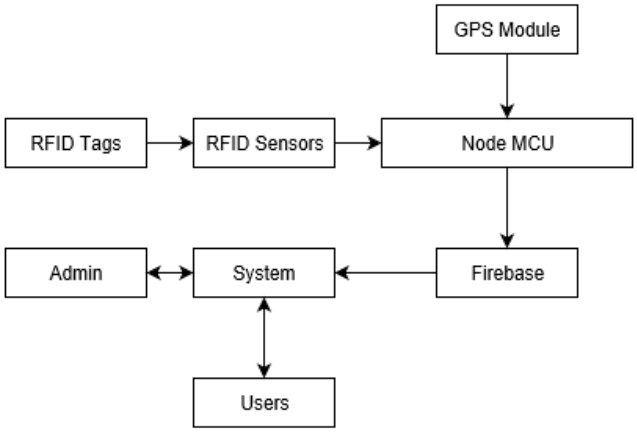

Fig -1: System Block Diagram

This RFID sensor is connected to the NodeMCU board. This sensor is connected to a 3.3V supply. This information is updated to the server using EPS 8266 Wi-Fi Module in NodeMCU. The mobile application acts as an interface between the system and the end user. The mobile application has the variety of purpose. For the admin side i.e. government it will alert them of the illegal vehicle or a stolen vehicle. The other users will be able to report theft as well as they can view their vehicle documents. It will further notify the users for additional fines for not renewing the vehicle documents. Then at the end the data which includes RFID tag and GPS location will be sent to the cloud with the help of Wi-Module through NodeMCU. In the cloud database all the information of the vehicle is recorded. Once the vehicle passes through the RFID sensors, it is checked for its authentication by the system and different actions are performed accordingly. By using the mobile application, the officials can able to track any unauthentic vehicles and it will also discourage criminals from vehicle theft. The merits of smart vehicle security system are shorter time to track the illegal vehicles, saves time and effort of traffic officials, discourages vehicles theft, easy assistance for vehicle owners and moreover it makes the overall city smart.

1.2 System Overview

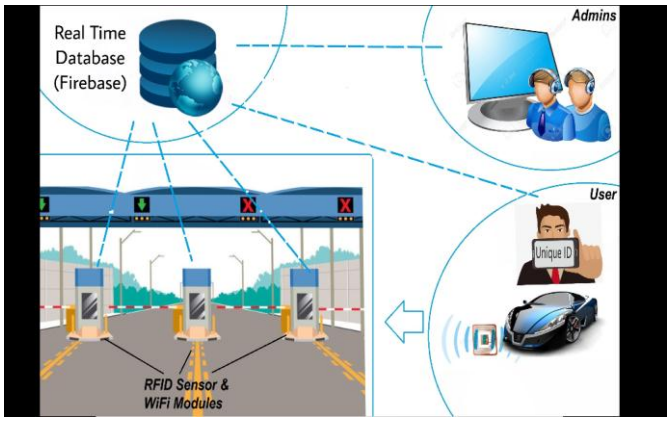

Fig -2: System overview

The figure above shows the overview of our system. It demonstrates the real world view when our system is implemented. It shows the vehicle with unique id i.e. RFID tag. When vehicle passes through the tollbooth containing RFID sensor, GPS module and NodeMCU, the data is read on the real time and sent to the real time database through the connected network. The data thus sent, is managed by the admins. The cases reported by the users are controlled by the government officials i.e. admin. Admin can view and delete the reported cases whenever needed. The users can also view their personal information and their status reports. The users can also view the registered locations of their vehicle. Whenever user passes from the check-post, real time data will be sent to the cloud which signifies the last location and using Haversine algorithm that deploys field which includes end points between user mobile GPS and last tracked vehicle location and notifies user with real time distance between the two entities (i.e. User and Vehicle). This mathematical illustrative modeling will help user report its vehicle based on the nearest check-post so that culprits will be traced easily.

2. SYSTEM ANALYSIS

The paper discusses the system which consists of two basic components: hardware components and an android application. The in-depth analysis of both of the components are given below.

2.1 Hardware component analysis.

The various hardware discussed above in the paper are assembled together and is presented in Fig-4. Various components are connected with each other with the help of NodeMCU and are presented as a circuit board.

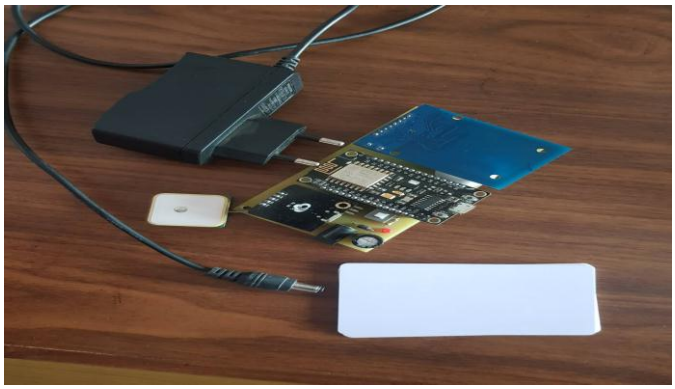

Fig -3: Hardware design.

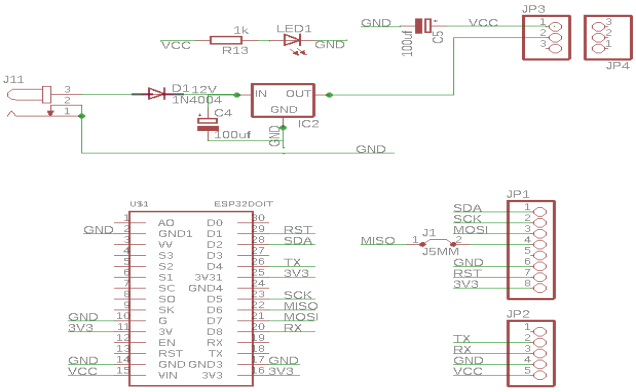

Fig -4: PCB schematic diagram

The above figure, Fig-4 represents the schematic representation of the printed circuit board(PCB) of our hardware components. Likewise, Fig-5 presents the board diagram of our system.

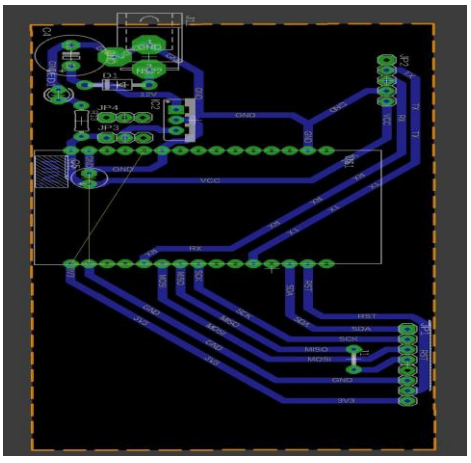

Fig -5: PCB board design

2.2 Android Application

Android application provides the interface for our system. It uses the data collected in Firebase from the tool-booths of various vehicles and serves various functionalities which are described in detail below:

2.2.1 Features of Admin

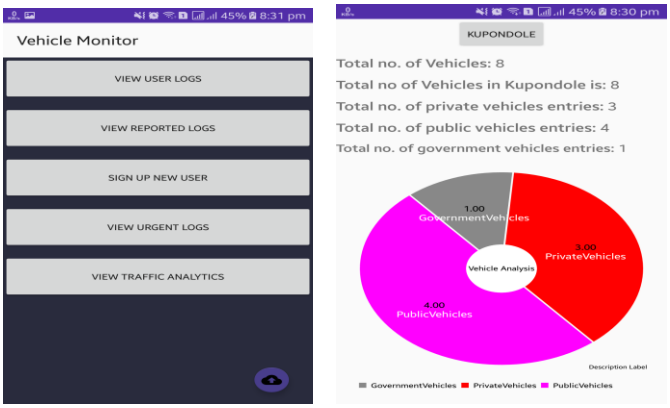

Fig -6: Admin Section

The android application has a separate sign-in features for admin and users. The android application will provide the information of vehicles passed through the various tool-booths along with the types of vehicles. It provides the statistics in the form of the pie-chart. Furthermore, it is the feature provided for admin only not for the users for security reasons.

2.2.2 Various additional features for users:

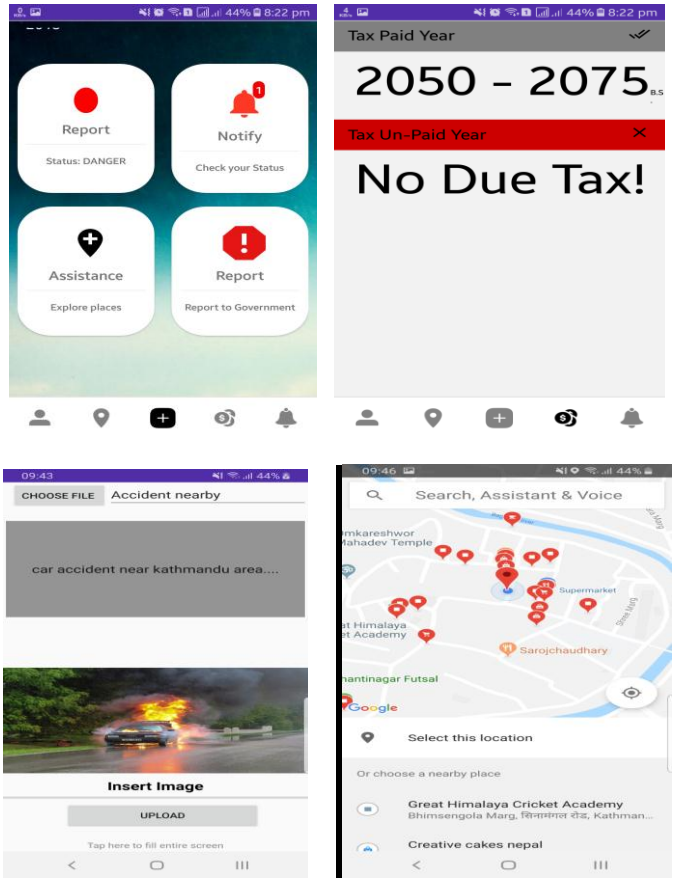

Fig -7: User functionalities.

The users of will have an interface that allows them to perform various functionalities. The users will be able to report any vehicle theft or other accidents to the government so that the government can take an immediate action. The user will be aware of the actions taken by government with the help of the icon and status of his vehicle that is provided to him or her. Furthermore, the user will receive the notification from government if the tax is due. The user can also view their profile and have a map assistance that allows them to view there nearby place.

2.2.3 Distance approximation using Haversine Algorithm.

Calculating distance or bearing between two end points or location has been one of the famous technology traditions; whether you try to provide recommendation to the user based on locations or give the nearest route to the user for their destination; you find this model everywhere. Haversine is the model which is there for very long time since 1900s. We attempted to use this model for the tracking the recent check post and tracking the movement of the vehicle from one check post to another. Whenever there is change in the distance; you are alerted so that you can directly locate your vehicle within the specified check post area. Data collected like latitude and longitude that is required for this algorithm is based on user mobile and last tracked location.

One can derive Haversine formula to calculate distance between two as:

$a=\sin^2(\Delta latDifference/2)+\cos(lat1).\cos(lat2).\sin^2(\Delta lonDifference/2)$

$c=2.atan2(\sqrt{a},\sqrt{(1-a)})$

$d = R.c$

where,

$\Delta latDifference = lat1 - lat2$ (difference of latitude)

$\Delta lonDifference = lon1 - lon2$ (difference of longitude)

R is radius of earth i.e. 6371 KM or 3961 miles

and d is the distance computed between two points.

We used this formula for the determination of distance between two end points where change in both endpoints is possible. So Haversine calculates difference in latitude of mobile GPS location and RFID tag last check post location and difference in longitude of them too and calculate distance with spherical law of cosine.

Example illustration:

Let's take one of latitude-longitude for calculation distance, GPS location: Shiphal, Gaushala, Kathmandu (Latitude 27.708496, longitude: 85.346573) and RFID tag last detected

check post location: Kupondole, Kathmandu (Latitude: 27.689922, Longitude: -85.312544)

Let's compute the distance with the above written formula. Your answer would be 3.32478Km(kilo-meter). The snapshot demonstrates the same result with calculation below:

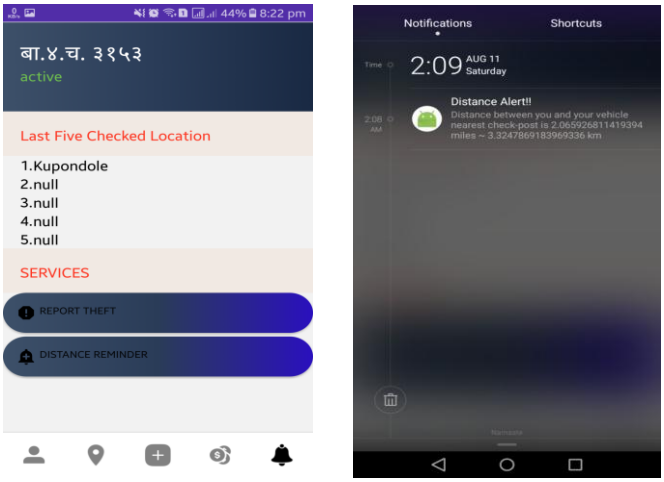

Fig -8: Distance approximation using Haversine algorithm

2.2.4 Linear Search through modeling for the latest visited check-post location

The paper depicts the project that started off with the integration of hardware components i.e. NodeMCU, ESP-8266 Wi-Fi module, MFRC522 RFID sensors and deploying it with the real time cloud so that the data of GPS and tag could be transferred to the cloud. Working on hardware integration; we simultaneously worked on the android application; as our hardware was able to send data to the cloud making node for each tag; now our task on hand was to get the user last location from that node which was created by our Hardware. So eventually we ended up using some searching algorithm. Since we were working with unstructured data sets; so using some advance algorithm like Interpolation and binary search would make our task more difficult. So we decided using very basic searching algorithm which is linear search. As per our study; we had certain realization that working with linear search would be more difficult while handling more data sets; As this is government based project so we nevertheless didn't expected data of user to be less than 1 lakh tag ID. So we started making model for our linear search. We started searching through bottom node of user on the cloud through the model called as model searching. We started searching node directly referred to the current tag so that searching would be faster.

Furthermore, we know that we don't have any capability to extend the performance of the hardware as the hardware has its limited specification; so we tried our best to extend the capacity our project to handle more user through software programming. To render layout faster for the user; we made use of the Fragment instead of Activity and instead of going through database for every time; we made use of the model to store the data temporarily and change it whenever needed.

A simple approach is to do linear search, i.e.

- Start from the leftmost element of arr[] and one by one compare x with each element of arr[].
- If x matches with an element, return the index.
- If x doesn't match with any of elements, return -1.

The time complexity of above algorithm is O(n).

3. CONCLUSIONS

Vehicle monitoring and tracking plays an effective and significant role in the area of transportation management system where efficient safety measures and preventing misuse of vehicles are the main concern. Our system discusses and helps in detecting vehicles, reporting theft and assistance to government officials. Although various researches have been done in this area and many methods have been implemented, still there is room for improvements. With a view to do improvements, our system provides an android application for vehicle data recognition and tracking using the RFID technology.

In future, the project can be used for large scale by increasing e capacity of hardware and obviously working with higher range of GPS.

The paper discusses the system which has broad domain. It is not only limited to the transportation security. It can be implemented in the various other sectors using the same concept. It can be used to for the apartments and companies to keep the track of the vehicles entering into their premises. It can be implemented in various service centers to give the benefits or facilities to their customers. It will make the society more managed. Likewise, it can also be implemented in any organization to keep the track of their vehicles where the problem of privacy is not affected. It can also be used more efficiently in shopping centers and for preventing various other crimes or theft control. It can be implemented in various other domains.

REFERENCES

[1] "AT Command Is Not Responding on Serial Monitor." NodeMCU Uno - AT Command Is Not Responding on Serial Monitor - NodeMCU Stack Exchange, NodeMCU.stackexchange.com/questions/38289/at-command-is-not-responding-on-serial-monitor.

[2] Lakshminarasimhan, Mahesh. IoT Based Traffic Management System. Mar. 2016, www.researchgate.net/publication/310036684_IoT_Based_Traffic_Management_System.

[3] Ding, Bin, et al. "The Research on Blind Navigation System Based on RFID - IEEE Conference Publication." Design and Implementation of Autonomous Vehicle Valet Parking System - IEEE Conference Publication, Wiley-IEEE Press, 8 Oct. 2007, ieeexplore.ieee.org/document/4340289/.

[4] "Haversine Formula." Haversine Formula - Rosetta Code, 29 Jan. 2016, rosettacode.org/wiki/Haversine_formula.

[5] Spk578. "GeoNet." Distance on a Sphere: The Haversine Formula,5Oct.2017, community.esri.com/groups/coordinate-reference-systems/blog/2017/10/05/haversine-formula.

[6] Miguelbalboa. "Miguelbalboa/Rfid." GitHub, 5 July 2018, github.com/miguelbalboa/rfid.

[7] Firebase. "Firebase/Firebase-Arduino." Firebase-Arduino-Master, 3 Aug. 2018, github.com/firebase/firebase-arduino.

[8] Mikalhart. "TinyGPS Library" TinyGPS Library for Getting Location, GitHub, 1 Sept. 2013, github.com/mikalhart/TinyGPS.

[9] PhilJay. "PhilJay/MPAndroidChart." MPAndroidChart for Android, GitHub, 6 June 2018, github.com/PhilJay/MPAndroidChart.

[10] "How to Use Google Maps on Your Android App." Google Maps SDK for Android, Google, developers.google.com/maps/documentation/android-sdk/utility/.

[11] Raahi. "Android Service Tutorial: How to Implement Service in Android." SwA, 24 July 2018, www.survivingwithandroid.com/2014/01/android-service-tutorial-2.html.

[12] *GPS Interfacing with NodeMCU ESP12: Getting Location Data*, circuitdigest.com/microcontroller-projects/interfacing-gps-with-nodemcu-esp12.

[13] Martins, Ruben, et al. "*Improving Linear Search Algorithms with Model-Based Approaches for MaxSAT Solving.*" Improving Linear Search Algorithms with Model Based Approaches, 11 Feb. 2015, www.tandfonline.com/doi/abs/10.1080/0952813X.2014.993508

[14] "How to Use Material Design in Your Android Apps." Material Design in Android, Android Authority, 30 Oct. 2017, www.androidauthority.com/how-to-use-material-design-in-your-android-apps-809937/