Towards an IoT based Vehicle Management System for Vehicle Tracking & Vehicle Diagnostics with OBD2 telematics

WNS Dabarera[1], NT Jayatilake[1], RHNS Jayathissa[1] and TL Weerawardane[1]

[1]*Department of Computer Engineering, General Sir John Kotelawela Defence University, Sri Lanka*

# 36-ce-0005@kdu.ac.lk

***Abstract**: In a fleet of vehicles, concern about each vehicle is mandatory. A Vehicle Management System (VMS) is primarily used to manage the vehicles' details and track the details of the vehicles. The importance of vehicle tracking and diagnostics must be stressed as other fleet management features also depend on them. Most fleet management systems depend only on vehicle location tracking using Global Positioning System (GPS) technology to manage the fleet's data. Through this proposed system, it aims to combine the vehicles' tracking details and diagnostic details for doing fleet management remotely by minimizing human resources. According to the reviewed systems, On-Board Diagnostics (OBD) has been identified as a reliable automotive technology that needs to track the performance inside the vehicle and regulate the performance. The proposed system has both a tracking device and an ELM327 Bluetooth OBD scanner in order to receive coordinates of the vehicle's location and vehicle diagnostics, respectively. This paper signifies the use of the Internet of Things (IoT) to accomplish the remote access of vehicles' data. For vehicle data to be sent to the cloud, GSM technology is required to send the vehicle's data to the cloud server for remote monitoring. As a cloud server, it uses a Message Queuing Telemetry Transport (MQTT) broker. The Arduino sensor data is lightweight and therefore uses the messaging protocol for the IoT for data transmission by connecting the devices to the internet. The proposed system incorporates the most advantageous technologies and devices for fleet management.*

***Keywords**: Fleet Management, IoT, OBD, GPS, MQTT, ECU, Remote monitoring, Telematics*

1. Introduction

Remote fleet management is addressed in the transportation sector unless there is no way to reliably monitor the required details of vehicles in real time. Determining the vehicle's conditions, calculating the instant fuel consumption, predicting the maintenance periods, etc. have to be more concerned because most fleet owners assume that doing predictions and determining the vehicles' conditions may be difficult in real time monitoring. As a result, most users still use traditional methods in vehicle management, such as manually recording data on each vehicle, manually calculating expenses, and relying on a mechanic to always have an idea about a vehicle, among other things. So that led to delaying the upliftment of the fleet management area. When considering existing systems, most of the vehicle management systems depend only on GPS technology to track the vehicle's position. According to the reviewed systems, it can be identified that for vehicle tracking it uses both mobile phones as well as telematics devices by embedding GPS. But installing a mobile phone inside a vehicle or using the driver's mobile phone's location to track the vehicle is not always the best way to track the vehicles in a fleet. There are telematics devices available as off-the-shelf devices, but most of the devices are highly expensive. Due to the high cost of telematics devices and the payable amount for the web application, it has also reduced the usage of the telematics devices in fleet management.

Major part of the transportation depends on fleets and buses, therefore the money allocated for that should be managed with minimal human resources (Mahaveer Penna, 2017). By considering the importance of fleet management and issues in the existing systems, it is easy to introduce a new system with a telematics device at low cost and dedicated software that can be used to keep track of vehicles and give the ability of remote accessing of vehicles' information with the use of the Internet of Things (IoT). Using this proposed system, fleet owners can always stay in touch with their fleets by knowing real-time information, including the history of each vehicle in their fleet. To collect the vehicles' data, it is using GPS for vehicle's location tracking and on-board diagnostics for vehicle diagnostics. Each vehicle uses Electronic Control Units (ECUs) to collect data, and from that data it can determine the vehicle and driver behaviour, environment, and be able to control the functionalities of the vehicle (Lotfi ben Othmane, 2018). ECU sensor data is collected from the ELM327 scanner by connecting it to the OBD II port, which is available in vehicles manufactured after 1996 as the on-board diagnostic system is a standard which was developed in the United States of America (USA) in 1996 and the OBD II port has been utilized in vehicles for diagnostic purposes (Reza Malekian, 2017). In the next sections of this paper, it discusses the technologies and features in reviewed systems, the architecture of the proposed system, including the specificity of each technology, and elucidates some of the results by discussing them appropriately. In this paper, we summarize the findings and new concepts to resolve the issues in existing systems and present future work that may be essential for building a good vehicle management system with cost-effective components.

2. Related Work

Vehicle-to-vehicle communication is done for the clearance of lanes in an emergency (Mallikarjuna Gowda, C.P., 2017). The vehicle that has an emergency sends a notification to the other nearest vehicles to clear the path. The cooling system of the vehicle is activated at high temperatures sensed by the attached temperature sensor in the vehicle. The GPS module calculates the speed of the vehicle, distance travelled, and the location of the vehicle. If the vehicle's speed is greater than the threshold value, then an alert is given to the driver. Proximity sensors, which are attached to the rear end of the vehicle, are used to measure the distance to the obstacle when reversing the vehicle, and if the obstacle is very close to the rear side, then a buzzer beeps to alert the vehicle driver. The real-time data is also displayed on a liquid crystal display to be viewed by the driver.

A solution to economic loss in fleet management is resolved by introducing a secure, reliable system for fleet organizations to gain remote control over their fleet using IoT (Mahaveer Penna, 2017). There is a fuel level sensor to monitor the fuel level and, along with the GPS based odometer connected to the system, to determine the travelled distance by the vehicle at an instance. The authors have stated that a GPS-based odometer which is connected to the Arduino board can generate a pulse for each covered meter, and then, through that pulse reading, it can easily find the distance travelled by the vehicle at a particular instant. It has been mentioned that in the future they'll add RFID sensors for the driver's identification and, through that, willing to link the vehicle data with the driver, and an alcohol sensor will also be added for the safety of the driver and the vehicle.

A real-time fleet management and security system are built on a Linux-based embedded microprocessor (Channakeshava Gowda V.R., 2015). The in-vehicle system gathers GPS data and has a biometric fingerprint sensor for taking the authentication of the driver, a panic button to show a panic situation, a camera for grasping pictures in the vehicle, and a remote-controllable voice alarm for alerting. The GPS-634R GPS receiver module is fixed in the vehicle to take the location details along with the date, time, and ground level speed. A SIM 900 GSM or General Packet Radio Service (GPRS) interface is included to enable communication between the in-vehicle system and the base station server. GPRS Hypertext Transfer Protocol (HTTP) is used for the sending of vehicle data like driver id, longitude and latitude of the location, date, time, and speed transmitted to the remote server. If the GPRS connection gets lost, then the program will automatically recognize it and re-enable the network connection. As well as in an instance of fading GSM channels, the program has been written to store the data in an internal database and then later send the data like camera images to the server by File Transfer Protocol (FTP).

A prototype model has been designed for low-cost fleet monitoring (Lotfi ben Othmane, 2018). Each vehicle in the fleet is attached with sensors and ECUs to collect fleet data like speed, engine temperature, revolutions per minute (RPM), and location details. The Raspberry PI 3 uses PICAN2 for communicating between the vehicles' CAN-bus, OBD-II cables for having the connection of PICAN 2 to a vehicle's OBD-II, Adafruit Ultimate GPS to take the location details, and Hologram Nova is used for communication between the PI and the server over the cellular network. Through a Web-application, it can visualize the fleet management data. Python is used for the fleet data collection and the Raspbian Operating System (OS) is the OS used here. Node.js has been selected to run the fleet management service, and MongoDB is the database used.

A concept of an intelligent and sensor network system has been designed for vehicle maintenance and safety(Christo Ananth, 2014). The PIC16F8777A is the microcontroller used here, which is low in cost, gives power-saving operating modes and has code protection. The power supply, Radio Frequency (RF) receiver, speed control section in zones, engine cooling fan, buzzer unit, GSM module for SMS, vibration detection sensor, engine and temperature sensor are connected to the microcontroller. The engine temperature monitoring system, accident detection and intimation system, speed and control section, and RF transmission are the major modules which are hosted by the PIC 16F877A. Through this proposed system, it can automatically send the various parameters like temperature of the engine, speed, etc. and send the records in case of any emergency or accident.

A system proposed for the military environment, to do predictive maintenance based on information given by on-board diagnostics and proactive maintenance based on both diagnostic signals from CAN-bus and statistical methods (Jan Mazal, 2020). European standard in OBDII supports for determine the controlling in exhaust such as lambda probes, fuel and air intake system's conditions like fuel injection pressure, ignition advance, intake air temperature, intake air quality etc., and standard input operational data like vehicle speed, engine speed, coolant temperature and oils etc. are also taken. Monitoring braking systems, monitoring safety systems like garbage, etc., monitoring transmissions, state of the brake pads, state of the brake fluid, and condition of the spark plugs, monitoring active chassis and monitoring the quality of motor oil are some characteristics parameters of the proposed system.

This study was done to identify the suitable technologies and to add features to the proposed system by comparing the best of them in referred systems. Table 1 summarizes the communication technologies, data processor types, and data storage methods identified in the referred systems. According to the summarized data in Table 1, Bluetooth has been included in existing systems to fulfill certain tasks. Vehicles that are equipped with GPS are beneficial in locating the vehicle's position (Kai, 2020). OBD is responsible for monitoring the vehicle's engine parameters, transmission, and emissions control components (Srinivasan, 2018).

Table 1. Qualitative analysis of the reviewed VMSs

R0eference work	Communication Techniques	Data Processor	Data Storage
(Mallikarjuna Gowda C P, 2017)	ZigBee protocols, GPS	Arduino Mega	-
(Mahaveer Penna, 2017)	GPS, IoT	Arduino Uno(ATMEGA328/P)	Cloud server
(Channakeshava Gowda V R, 2015)	GPS/GSM.,GPRS HTTP, FTP	Cubietruck	GUI server
(Lotfi ben Othmane, 2018)	GPS, IoT, ECUs, PICAN2 Hologram Nova-2G/3G	Raspberry Pi 3	Remote service
(Christo Ananth, 2014)	GSM	PIC 16F877A	-
(Jan Mazal, 2020)	Bluetooth, GPS, ECUs CAN bus protocol SAE J1939	OBD(have microcontroller based processing system)	CANcase log device
(Srinivasan, 2018)	BLE,4G WIFI dongle, GPS	Raspberry Pi 3, OBD-II	Cloud server
(Dhanalakshmi, 2017).	GSM, Ethernet ,GPS/GPRS	Arduino UNO ATMEGA 328p	Web server

Table 2. Analysis of frequently used techniques for VMSs based on specificity

Technique	Use for	(Mallikarjuna Gowda C P, 2017)	(Mahaveer Penna, 2017)	(Channakeshava Gowda V R, 2015)	(Lotfi ben Othmane, 2018)	(Christo Ananth, 2014)	(Jan Mazal, 2020)	(Srinivasan, 2018)	(Dhanalakshmi, 2017)	(Garba Suleiman, 2018)
GPS	Location Tracking	✓	□	✓	✓	□	□	✓	✓	✓
	Calculate the distance	✓	✓	□	□	□	□	□	□	□
	Vehicle’s speed calculation	✓	□	✓	□	□	□	✓	□	□
OBD	Collect ECU’s data	□	□	□	✓	□	✓	✓	□	□
	Gain vehicle’s speed	□	□	□	✓	□	✓	□	□	□
GPRS/GSM	Connect to the Internet	□	□	✓	□	□	□	□	□	□
	Send SMS	□	□	□	□	✓	□	□	□	✓
External sensors	Sense the vehicle parameters	✓	✓	✓	□	✓	□	✓	□	□

The Arduino or Raspberry Pi boards are generally used to read and process data and send it to a server or else use another service as a data storage method. As shown in Table 2, the specificity of GPS and OBD depends on purpose of what those techniques are used for. If it is taken, a system proposed by Lotfi ben Othmane, (2018), both GPS and OBD data are combined, and the vehicle’s speed is taken from the OBD data by manifesting the accuracy of OBD data. The calculated distance travelled throughout a trip is calculated by GPS data and is also applicable in existing systems. In most of the reviewed systems, they use the external sensors to measure some of the vehicles’ parameters like fuel consumption, coolant temperature, etc. In the next section of this paper, it suggests a more reliable way to use GPS and OBD by reducing the overall cost of having a good VMS.

A. The Proposed System

The proposed system consists mainly of a telematics device, an ELM327 Bluetooth OBD scanner, and a web application. Live vehicles’ data and history are visible through a web application for vehicle owners. Variations of OBD parameters can be seen on live charts. Live location displays on a Google map. The trip details like distance travelled, time taken, fuel consumption and abnormal OBD readings etc. are stored in the database and alert the malfunctions and maintenance through the web application based on the unusual OBD readings , Diagnostic Trouble Codes (DTC)s and mileage.

B. Architecture

The architecture of the IoT-based vehicle management system is shown in Figure 1. The system consists of a web application developed using MEAN (MongoDB, Express.js, Angular, Node.js) stack technology which provides real-time and past details of the vehicles for the fleet owners; a telematics device to receive the vehicle data and send those data to the server using 2G or 3G technology with the help of a GSM SIM 900 module; and an OBD device, which is also a sub part of the system which needs to gain the diagnostic details. The GSM SIM900 module, GPS module, and HC-05 Bluetooth module are connected to the Arduino mega in the telematics device. 12 V is supplied to the GPRS module, and a 5V DC to DC voltage regulator is used to take the power to the Arduino mega board. Altogether, it forms a telematic device which can be powered by the cigarette port in the vehicles.

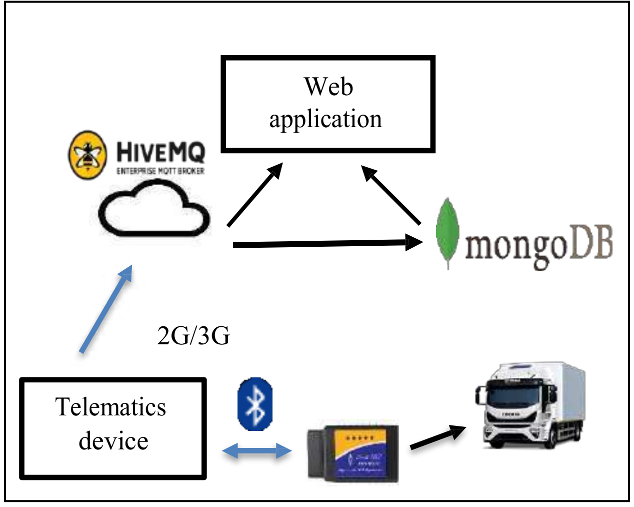


Figure 1. Architecture of the proposed system

C. OBD Scanner and On-Board Diagnostics

OBD is a standard tool that allows a person to check a car's engine status by plugging the OBD scanner into the port that has a 16-pin in it, and the port can be found under the steering wheel (J V Moniaga, 2017). The status of the vehicles can be obtained from diagnostic parameters like emission control, driving speed, battery voltage, engine coolant temperature, MAF rate, and more information from ECU. OBD-I has been replaced by the current OBD system, OBD-II. (Hussein Ali Ameen, 2021). OBD II uses Diagnostic Trouble Codes (DTCs) and Parameter Identifiers (PIDs) to diagnose malfunctions in a vehicle's subsystems. PIDs, which are hexadecimal values, are used to measure real-time parameters. Bluetooth and Wi-Fi adapters are available for low cost in the market. For this proposed system, it has chosen the ELM 327 Bluetooth scanner by considering the ease of retrieving data.

Figure 2. ELM 327 Bluetooth Scanner

A scanning tool sends a message having a hexadecimal code to request information from the ECU associated with a specific parameter, which is defined by the SAE J1979 standard, and there are five OBD-II signalling protocols called SAE J1850 (VPW and PWM), ISO 15765, ISO 1941-2, and IS0 142300-4 used to interpret the message (Reza Malekian, 2017). The ECU finally sends back a hexadecimal code in response. The actual measurement is obtained by converting the hexadecimal value to a decimal value or by doing calculations.

D. Technologies

The technologies used in this project are selected by considering the optimum advantage of each one individually to fulfill the requirements of the system. GPS technology is usually needed for location tracking. By using a GPS module, the latitude, longitude, and time of the current location of the vehicle will be grasped and sent to the server. A Map API is required to visualize the GPS coordinates on a map. The Google Map API and Mapbox API are well-known map APIs which offer the service for tracking using maps and related features. In this project, it is using Google Map API as it is easy to include maps to the web application and provide benefits to the users reliably.

In the automotive field, on-board diagnostics is essential for vehicle emission diagnostics and malfunction diagnostics. There are dedicated sensors to send electrical signals as feedback to the vehicle's main ECU (Tamer Abukhalil, 2019). OBD-II is the latest technology in OBD systems and the Controller Area Network (CAN) bus connects the ECUs inside the vehicle to consecutively monitor the running condition of the vehicle (Jheng-Syu Jhou, 2013). The ELM327 Bluetooth OBD-II is used for sending the OBD data to the microcontroller board in a telematics device by connecting a HC-05 Bluetooth module to the microcontroller board. OBD data and GPS data are very small in size. Therefore, there is no need to use a high-microprocessing-power microcontroller to read and process the data (Chetan S. Patel(M.Tech), 2020). In this proposed system, it is using an Arduino Mega, which has sufficient software serial pins and hardware serial pins to connect other modules. For communication purposes, it has chosen the GSM SIM 900 module to get a 2G or 3G internet connection to send the vehicle data to the cloud server. GSM technology is more reliable and low-cost for use by the number of vehicles in a fleet. IoT devices mostly use the MQTT protocol to send the sensor data to a server or in an instance of device-to-device communication. This system makes use of the HIVEMQ cloud public broker, which can be tested for free and can be used in the future as a private cloud MQTT broker to send encrypted messages in a more secure manner. The MQTT broker acts as the cloud server, and the vehicle data is sent to the cloud server for real-time monitoring purposes and for future usage of data. Data is sent to the MongoDB Atlas, which is a cloud-based database. As MongoDB is used, as it is a NoSQL database, which gives the flexibility of handling the data of a fleet, and there is no restriction on the variety of data that is stored in the database.

E. Web application.

The MEAN stack is used for web application design, providing powerful application by NodeJS backend, angular framework handling by JavaScript or Typescript as a frontend language, MongoDB as a no SQL database, and Express.js, which manages routes and servers. Figure 3 shows the home page, which displays the vehicle asset number on the left side, and on the right side it displays the vehicle's details, which is searched by asset number.

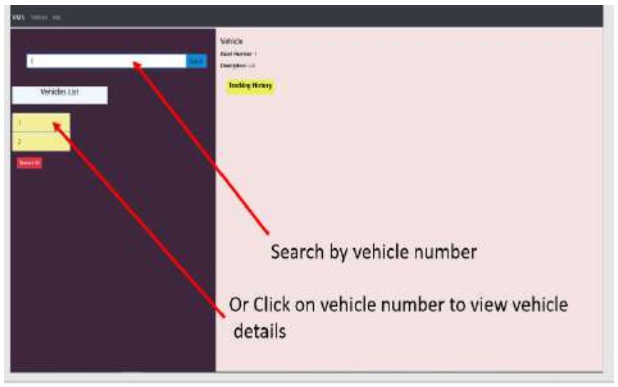
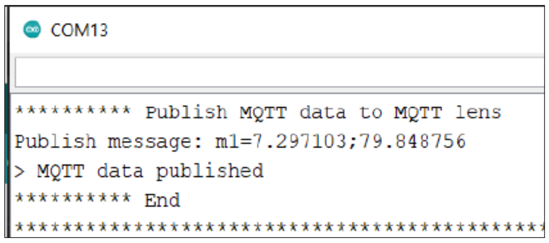


Figure 3. Home page of Web Application

If the user wants to track the tracking history of the vehicle, they can click the yellow-colored tracking history button on the right side of the home page. In addition to these, add a new vehicle, update the vehicle's details, real-time monitoring of variations of OBD parameters, real-time vehicle location viewing, and past diagnostics and tracking data can be viewed through the web application.

F. Database and MQTT Broker

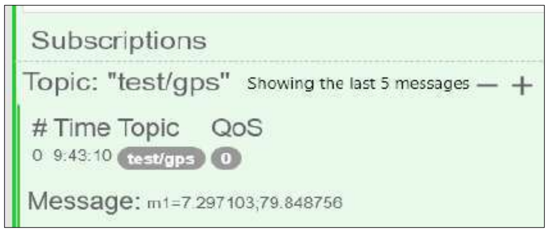
As the name suggests, the Message Queuing Telemetry Transport (MQTT) protocol can be used for telemetry applications, which is a lightweight, open messaging protocol used for communications among IoT related devices. In the MQTT protocol, there is a broker(server) that contains "topics. Topics are the way to determine who receives the data generated by a sender (Jay Lohokare, 2017). The vehicle data is published in real time by Arduino devices via 2G/3G connection to the subscribed topic of the HIVEMQ broker, which is a public MQTT broker used to test the demonstration process. As in Figure 5 the published data to the broker can be viewed through the MQTT lens, which can be installed as an extension in Google Chrome.



```
COM13

********** Publish MQTT data to MQTT lens
Publish message: m1=7.297103;79.848756
> MQTT data published
********** End
***********************************************
```

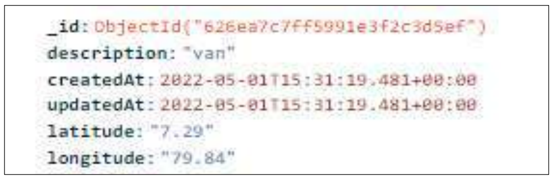
Figure 4. Serial monitor reading of publishing the GPS coordinates to MQTT broker



```
Subscriptions
Topic: "test/gps"   Showing the last 5 messages — +
# Time Topic   QoS
0  9:43:10  test/gps  0
Message: m1=7.297103;79.848756
```

Figure 5. Test the publication of data through MQTT Lens

MongoDB is a document-oriented database system, classified as a NoSQL database system, and the data is stored in a document format with a structure like JSON documents (Choopan Rattanapoka, 2019). As shown in Figure 6, The vehicles' data published to the broker/server is sent to the MongoDB Atlas for future usage and to do analysis about vehicles in the fleet.

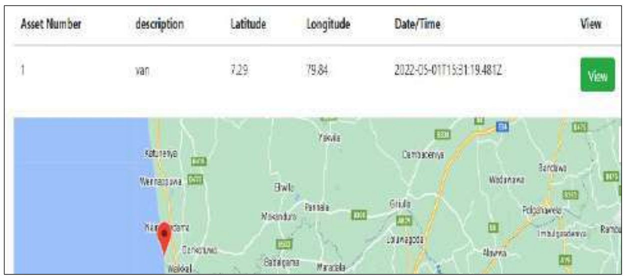


```
_id: ObjectId("626ea7c7ff5991e3f2c3d5ef")
description: "van"
createdAt: 2022-05-01T15:31:19.481+00:00
updatedAt: 2022-05-01T15:31:19.481+00:00
latitude: "7.29"
longitude: "79.84"
```

Figure 6. Recorded GPS data in MongoDB Atlas

3. Results and Discussion

The Google Maps API provides clear and accurate maps to view a certain asset's location. Figure 7 shows the location of a vehicle at a certain position. The stored coordinates are taken from the MongoDB and displayed using Google Maps in the web application. Live location can be monitored through the web application by retrieving the live GPS data directly from the server. At the same time, the trip details like distance travelled , average speed, and overall time for the trip are stored in the database.



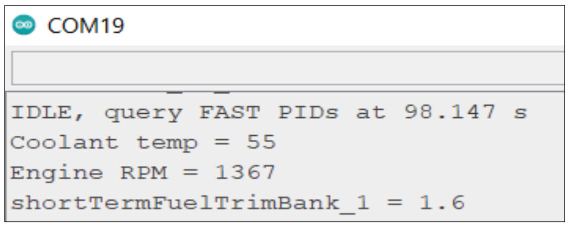
Asset Number	description	Latitude	Longitude	Date/Time	View
1	van	7.29	79.84	2022-05-01T15:31:19.481Z	View

Figure 7. GPS coordinates displayed using google maps

To monitor the vehicle's diagnostics details, it should connect the OBD device to the OBD port in the vehicle. If it connects well and receives power from the vehicle, then the red light in the scanner is lit up and remains constant. Then connect the OBD scanner with the HC-05 Bluetooth module by pairing it using the Bluetooth Media Access Control (MAC) address of the ELM 327 OBD device. The MAC address of the OBD device used in this project is 00:1D:A5:01:0D:1C. These addresses can be found using AT commands, and AT commands are needed for pairing the Bluetooth devices. After connecting with the Bluetooth, the OBD readings are received only after the ignition key of the vehicle. When the OBD scanner starts to connect with the vehicle CAN Bus network by connecting to the ECUs, the green lights of the ELM327 scanner blink automatically and remain blinking when requesting OBD data by sending the PIDs. The below code part is written for the purpose of reading vehicle rpm, and Figure 8 shows some OBD parameters displayed in the serial monitor after sending the PID commands. There are different PIDs to read the OBD parameters, and in this project, it is using the ARDuino library, which already has predefined OBD PIDs as understandable functions. The OBD data can be easily accessed by calling functions such as ex, rpm ().

```
float tempRPM = myELM327.rpm();
```

```
 if (myELM327.nb_rx_state == ELM_SUCCESS)
 {
   rpm = (uint32_t)tempRPM;
   Serial.print("RPM: "); Serial.println(rpm);
 }
 else if (myELM327.nb_rx_state !=
ELM_GETTING_MSG)
   myELM327.printError();
```



```
IDLE, query FAST PIDs at 98.147 s
Coolant temp = 55
Engine RPM = 1367
shortTermFuelTrimBank_1 = 1.6
```

Figure 8. Sample OBD parameter readings viewed in serial monitor.

Table 3. OBD readings of vehicles

OBD parameter	Suzuki van	KDH van	Wagon R car	Toyota Allion car	Vezel car
Throttle (%)	16.9	-	17.25	-	16.86
MafRate (g/s)	-	-	3.39	-	4.32
Intake AirTemp (^{0}C)	-30.80	-40	-40	-40	35
Mph	0	-	3.73	-	-
Kph	1.8	-	6	-	-
Engine speed (rpm)	0	-	1117.25	-	1000
LongTermFuel TrimBank_2 (%)	-100	-100	-	-100	-
ShortTermFuel TrimBank_2 (%)	-	-100	-	-100	-
LongTermFue lTrimBank_1 (%)	-100	-100	3.91	-100	-
ShortTermFuel TrimBank_1 (%)	-	-100	-5.47	-100	-
EngineCoolant Temp (^{0}C)	-40.35	-40	51	-40	33
EngineLoad %	-	-	40.39	-	68
FuelSystem Status	-	-	512	-	-
FreezeDTC	-	-	-	-	-
Battery voltage (V)	11.9	12.24	13.70	13.80	11.94

Table 3 shows the readings taken from several models of vehicles at a certain instant. According to the data in Table 3, it shows that not all PIDs support all types of vehicles. But each sensor reading is responsible for giving the current status of the vehicle, which is essential in determining the malfunctions of the subsystems in the vehicle. In further research on this project, the necessity of estimating the vehicle's performance remotely will be addressed using live OBD readings displayed via the web application and the OBD data recorded in the database to know the abnormal readings. Then the alerts are displayed in the web application if the OBD parameter reading increases or decreases more than the normal range of values for a certain OBD parameter. Some models of vehicles that have been manufactured lately support many PIDs, so the calculation of the fuel consumption can also be applied by using some OBD parameters like Mass Air Flow (MAF) reading, intake manifold pressure, rpm, etc. Using equation 1, one can find the fuel consumption of a certain vehicle. Fuel flow and vehicle speed are measured in litres per hour (l/h) and in kilometres per hour (km/h) respectively to obtain the fuel consumption in litres per kilometre (l/km) (Reza Malekian, 2017).

$$F□e□(C□□□□□□□i□□)\left[\frac{1}{100□□}\right] = \frac{Fu□□F□□w\left[\frac{□}{h}\right]}{□□□□□\left[\frac{□□}{h}\right]} \times 100 \quad \text{(eq 1)}$$

From current fuel flow and the current vehicle speed, instantaneous fuel consumption is calculated when the vehicle is moving and the engine is operating (Hu Jie, 2010).

$$I□□□a□.\ F□e□\ C□□□□.\left[\frac{1}{□□}\right] \frac{Fu□□F□□w\left[\frac{□}{h}\right]}{□□□□□\left[\frac{□□}{h}\right]} \qquad \text{(eq 2)}$$

If MAF PID is available, then fuel flow can be calculated as below (Samuel Shaw, 2019), (Reza Malekian, 2017).

$$F□e□\ F□□w\left[\frac{□}{h}\right] = \frac{MAF}{AF□ \times FD} \times 3600 \qquad \text{(eq 3)}$$

Where MAF is the Mass Air Flow in g/s, AFR is the Air to Fuel Ratio, and Fuel Density is denoted in FD. If the MAF sensor is not available in some vehicles, then there are different methods to calculate the fuel flow. It can directly calculate fuel consumption using engine fuel rate and speed or as a function of absolute load, RPM, and engine displacement, or it can be used to calculate fuel flow with intake manifold pressure, intake air temperature, RPM, and engine displacement (Hu Jie, 2010).

4. Conclusion

This proposed system is targeted to automate vehicle management to a certain extent with cost-effective and reliable methodologies utilizing IoT. The existing systems use GPS for vehicle tracking and OBD for vehicle diagnostics purposes. But it seems that there are no proper implementations of a VMS with the integration of GPS and OBD technologies. Therefore, the proposed system can resolve many issues in existing systems by allowing real-time monitoring of tracking and diagnostics data with the combination of GPS and OBD technologies. It reduces the additional cost of fixing external senses to determine fuel consumption, coolant temperature, vehicle speed, etc. A complete system for fleet management can be built using GPS and OBD integration by fulfilling the features like vehicle tracking, vehicle health monitoring, and predicting maintenance with the use of data like the distance covered calibration through GPS data and diagnostics data taken from OBD. Bluetooth technology has been chosen for data transmission to the microcontroller by the ELM327 Bluetooth scanner because of its easy accessibility and cost effectiveness compared to using wired OBD devices. GSM is used in most systems to send SMS in an immediate instant, but in this proposed system, GSM is used for connecting the devices to the internet via 2G or 3G to send the data to the cloud server using the MQTT protocol. Using Wi-Fi or Ethernet devices inside a vehicle is an unnecessary cost as GSM technology is sufficient to send the sensor data to a cloud server. According to the referred technologies for building the web application and backend side, the Angular framework based on MEAN stack technology was selected as the appropriate one for analysis of telematics data. Therefore, it can be concluded that the issues recognized in existing systems will be resolved through the proposed system, which consists of the most suitable technologies, components, and methodologies for data retrieval and transmission.

5. Future Works

There is some other future work to be done and evaluated, such as keeping records on trips like distance travelled, travelled path, and started and ended times of a trip, which are important in determining a vehicle's travelled history. The fuel consumption needs to be calculated from OBD readings and displayed in a web application to have a clear idea about the fuel consumption according to the distance travelled and predict the vehicle maintenance according to the diagnostics details. Other things discussed in this paper are also implemented, and the system will be distributed to a fleet in the future to evaluate its working procedure.

References

Ahmad Aljaafreh, M. K. I. A.-F. K. A. R. A.-S. S. A.-E. a. W. S., 2011. *Vehicular Data Acquisition System for Fleet for Fleet Management Automation.* s.l., s.n.

Channakeshava Gowda V R, G. K., 2015. *Real Time Vehicle Fleet Management and Security.* s.l., s.n.

Chetan S. Patel(M.Tech), P. J. A. G., 2020. IoT based Augmented Reality Application for Diagnostic vehicle's condition using OBD-II Scanner. *International Journal of Engineering Research & Technology (IJERT).*

Choopan Rattanapoka, S. C. A. C. A. S., 2019. *An MQTT-based IoT Cloud Platform with Flow Design by Node-RED.* s.l., s.n.Christo Ananth, N. J. T. A., 2014. Intelligent Sensor Network for Vehicle Maintenance System. *International Journal of Emerging Trends in Engineering and Development.*

Cui Dong, W. G. H. K., 2020. *Research On The Method Of Intelligent Vehicle Management Analysis And System Design.* s.l., s.n.

Dhanalakshmi, A. S. L., 2017. INSTANCE VEHICLE MONITORING AND TRACKING WITH INTERNET OF THINGS USING ARDUINO. *INTERNATIONAL JOURNAL ON SMART SENSING AND INTELLIGENT SYSTEMS SPECIAL ISSUE, SEPTEMBER.*

Garba Suleiman, A. O. U. a. S. E. A., 2018. *Development and Implementation of Vehicle Movement Notification and Location using GSM, GPS and* (Anon., n.d.) (Kai, 2020) (Kai, 2020)*Web based Information System.* s.l., s.n.

Hu Jie, Y. F. T. J. W. P. C. K., 2010. *Developing PC-Based Automobile Diagnostic System Based on OBD System.* s.l., s.n.

Hussein Ali Ameen, S. S. A. K. M., 2021. Driving Behaviour Identification based on OBD Speed and GPS Data Analysis. *Article in Advances in Science Technology and Engineering Systems Journal.*

J V Moniaga, S. R. M. D. A. H. a. F. S., 2017. *Diagnostics vehicle's condition using obd-ii and raspberry pi.* s.l., s.n.

Jan Mazal, A. F. V. T., 2020. *Modelling and Simulation for Autonomous Systems.* s.l., s.n.

Jay Lohokare, R. D. S. A. P. R. A., 2017. *Scalable tracking system for public buses using IoT technologies.* s.l., s.n.

Jheng-Syu Jhou, S.-H. C., 2013. The Implementation of OBD-II Vehicle Diagnosis System Integrated with Cloud. *2013 Second International Conference on Robot, Vision and Signal Processing.*

Jyong Lin, S.-C. C. Y.-T. S. a. S.-H. C., 2009. *A Study on Remote On-Line Diagnostic System.* s.l., s.n.

Kai, W. G. a. H., 2020. *Research on the Method of Intelligent Vehicle Management Analysis and System Design.* s.l., s.n.

Lotfi ben Othmane, V. A. K. B. M. F. W. F. G. T. H., 2018. *Demo: A Low-Cost Fleet Monitoring System.* s.l., s.n.

Mahaveer Penna, D. A. B. G. K. R. L. N. M. S. K. G., 2017. *Smart Fleet Monitoring System using Internet of.* s.l., s.n.

Mallikarjuna Gowda C P, R. H. C. S. M. R. K. R. A. R. N. P. P., 2017. *Design and Implementation of Real Time Wireless System for Vehicle Safety and Vehicle to Vehicle.* s.l., s.n.

Reza Malekian, N. R. M. L. N. B. U. A. K. C.-O., 2017. Design and Implementation of a Wireless OBD II Fleet Management System. *IEEE Sensors Journal.*

Samuel Shaw, Y. H. W. Z., 2019. *Instantaneous Fuel Consumption Estimation Using Smartphones.* s.l., s.n.

Srinivasan, A., 2018. *IoT Cloud Based Real Time Automobile Monitoring System.* s.l., s.n.

Tamer Abukhalil, H. A. M. A. a. B. A. Y. A., 2019. Fuel Consumption Using OBD-II and Support Vector Machine Model. *Journal of Robotics.*

Acknowledgment

For this research, I have received immense support and guidance from lecturers at the Department of Computing of General Sir John Kotelawala Defence University. I greatly appreciated the assistance provided by them to successfully complete this research.

Author Biography

WNS Dabarera is a final year undergraduate in BSc. (Hons.) in Computer Engineering at the General Sir John Kotelawala Defence University. She is currently pursuing an internship in software engineering in the information technology industry. She is also interested in research areas such as the Internet of Things, machine learning, wireless technologies, and embedded systems.

Naduni Thamalka Jayathilake, currently employed in the Department of Computer Engineering, Faculty of Computing, General Sir John Kotelawala Defence University, is currently reading for her M.SSc. in Applied Electronics at the University of Colombo. She holds a B.Sc. (Hons.) in Computer Engineering from General Sir John Kotelawala Defence University, and a B.Sc. Degree in Electronics and Automation from the University of Colombo. Her fields of interest include multi- agent systems, sensor networks, IoT, and wireless networking.

Nilani Jayathissa received the B.Sc.Eng. degree in Electrical and Information Engineering from the University of Ruhuna, Galle, Sri Lanka, in 2015, and the M.Sc. degree in Electronics and Automation from the University of Moratuwa, Moratuwa, Sri Lanka, in 2021. After graduation, she joined the National Institute of Business Management as a consultant/lecturer and served for more than five years. Currently, she is working as an instructor at the Department of Computer Engineering, Faculty of Computing, General Sir John Kotelawala Defence University, Ratmalana, Sri Lanka. Her current research interests include wireless communication, localization, and digital signal processing.

Prof. TL Weerawardane was graduated in Electrical Engineering from the University of Moratuwa in 1998 and consequently he received MSc Degree "Communication and Information Technology" in 2004 and received Ph.D. from the University of Bremen, Germany in 2010. Prof. Thushara Lanka Weerawardane worked in Kotelawala Defence University as senior lecturer Gr.1 from 2012 to 2016 during this period he held several academic and administrative positions such as Head of the Department and Dean, Faculty of Engineering. Currently, he is working in Kotelawala Defence University as Professor.