VEHICULAR ON-BOARD DIAGNOSTICS TOOL APPLICATION

A THESIS SUBMITTING TO KWAME NKRUMAH UNIVERSITY OF SCIENCE AND TECHNOLOGY IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
BACHELOR OF SCIENCE IN COMPUTER ENGINEERING

COLLEGE OF ENGINEERING
KWAME NKRUMAH UNIVERSITY OF SCIENCE AND TECHNOLOGY

AUGUST 2023

Declaration

We the undersigned solemnly declare that the project report dubbed 'VEHICULAR ON-BOARD DIAGNOSTICS TOOL APPLICATION' is based on our work conducted during the study. We assert the statements made and conclusions drawn are an outcome of our research work. We further certify that:

1. The work contained in the report is original and has been done by us.
2. The work has not been submitted to any other Institution for any other degree/certificate. in this university or any other university.
3. We have followed the guidelines provided by the university in authoring the report.
4. Whenever we have used materials (data, theoretical analysis, and text) from other sources, we have given due credit to them in the text of the report and given their details in the references.

Team Members:

.... Undergraduate Student
Koduah Hansen Kwadwo Junior

.... Undergraduate Student
Machu Rexford Agyabeng

.... Undergraduate Student
Frimpong Osei Kofi

.... Supervisor
Dr. Tutu Tchao

Abstract

This thesis presents the design and implementation of an On-Board Diagnostic II (OBD II) application that connects wirelessly to an OBD II adapter to present the user with an intuitive interface about the performance of the vehicle. The project focuses on addressing the problem of limited accessibility of real-time vehicle data and information by providing a user-friendly interface for users to interact with their vehicles. The application is designed to support Bluetooth wireless communication protocols for seamless connectivity with the OBD II adapter. The application features an intuitive user interface, including real-time data monitoring, fault code scanning, and the diagnostic trouble code interpretation.

It aims to provide access to a wide range of vehicle data, including engine RPM, fuel efficiency, and fault codes. The primary goal of the project is to create a user-friendly app interface that displays instant data in an easy-to-understand format, allowing vehicle owners and mechanics to quickly diagnose and address vehicle issues. The project also focuses on the design of a data storage and retrieval system that allows users to save and export data for future analysis.

Dedication

We dedicate this project to God Almighty, our creator, strong pillar, and source of inspiration, wisdom, knowledge, and understanding. He has been the source of our strength throughout this study and on His wings only, have we soared.

Acknowledgements

We would like to extend a special thank you to Dr. Tutu Tchao, our supervisor, and the entire Computer Engineering department for providing us with the chance to work on this project and for assisting us with the extensive research.

Additionally, we would like to thank our classmates and fellow students for always helping us get back up when we collapsed. We had a great experience, and we will always cherish that. We have come to gain experience and acquire more knowledge in this field. We are thankful to them.

Contents

List of Figures

Acronyms

API Application Programming Interface

ASCII American Standard Code for Information Interchange

AWS Amazon Web Services

BEV Battery Electric Vehicle

BLE Bluetooth Low Energy

DB Database

DTC Diagnostic Trouble Code

ECU Electronic Control Unit

ELM A programmed micro-controller produced for translating the on-board diagnostics (OBD) interface found in most modern cars

GNSS Global Navigation Satellite System

GPS Global Positioning System

OBD On-Board Diagnostics

PID Parameter Identification

RPM Revolutions Per Minute

SQL Structured Query Language

VIN Vehicle Identification Number

Chapter 1

Introduction

1.1 Background of the Study

On-Board Diagnostics (OBD) is a standardized system implemented in modern vehicles to monitor and report the performance of various systems and components. The OBD system is responsible for identifying and diagnosing issues with the engine, transmission, exhaust, and other vital components of the vehicle.

The second generation of the OBD system, commonly referred to as OBD II, was introduced in 1996 and has since become the standard for all vehicles sold in the United States. OBD II systems are designed to provide real-time information about the performance of a vehicle and to alert drivers to potential issues or malfunctions.

With the increasing availability of wireless technology and mobile devices, there has been a growing demand for OBD II applications that can wirelessly connect to OBD II adapters and present vehicle data in a user-friendly way. These applications allow users to monitor the performance of their vehicles and diagnose issues on the go, without the need for specialized equipment or technical knowledge.

Numerous commercial OBD II applications are available in the market, but most of them have limited compatibility with different OBD II adapters and vehicle models. Moreover, many of these applications lack a tool-selective user interface and do not provide accurate or reliable data.

This undergraduate study aims to design and implement an OBD II application that wirelessly connects to an OBD II adapter and provides an intuitive interface for users to interact with their vehicles. The study seeks to address the limitations of existing OBD II applications by providing a user-friendly and reliable solution that can support various OBD II adapters and vehicle models. The study will also evaluate the performance of the application in terms of accuracy, reliability, and usability.

1.2 Problem Statement

Despite the availability of wireless OBD II adapters that provide real-time access to vehicle data, there is still a lack of user-friendly applications that can easily connect to these adapters and present the data intuitively and comprehensively. Most existing applications are limited in their functionality and do not provide users with a comprehensive diagnostic tool set. This lack of user-friendly and comprehensive applications hinders the ability of vehicle owners to easily access and interpret vehicle data and diagnostic trouble codes. Therefore, there is a need for an OBD II application that can wirelessly connect to an OBD II adapter and present the user with an intuitive interface for accessing real-time vehicle data and diagnostic trouble codes, as well as a comprehensive diagnostic toolset for fault code scanning and interpretation. Such an application would provide vehicle owners with an easy and convenient way to monitor and diagnose their vehicles, improving vehicle maintenance and reducing vehicle-related expenses.

1.3 Objectives

The main objective of this study is to design and develop an OBD II application that wirelessly connects to an OBD II adapter and presents the user with an intuitive interface for accessing real-time vehicle data and diagnostic trouble codes. To achieve this objective, the following specific objectives will be considered:

1. To conduct a comprehensive literature review of the OBD II standard, wireless communication technologies, and existing OBD II applications to identify best practices and design considerations.

2. To design and develop an OBD II application that leverages the latest wireless communication technologies to wirelessly connect to an OBD II adapter and provide real-time vehicle data monitoring and the diagnostic trouble code interpretation.

3. To test and evaluate the performance and user-friendliness of the application and compare it with existing OBD II applications and diagnostic tools.

4. To provide recommendations for future improvements to the application, such as adding more features and expanding compatibility with different vehicle models.

5. To contribute to the body of knowledge on OBD II applications and their use in improving vehicle maintenance and reducing vehicle-related expenses.

By achieving these objectives, this study will contribute to the development of user-friendly and comprehensive OBD II applications that can provide vehicle owners with an easy and convenient way to monitor and diagnose their vehicles, improving vehicle maintenance and reducing vehicle-related expenses.

1.4 Significance of the Study

In recent years, the automotive industry has seen a significant increase in the use of technology to enhance vehicle performance and provide drivers with real-time data. One such technology is the On-Board Diagnostics (OBD) system, which has become a crucial part of modern vehicles. OBD is an electronic system that monitors various components of a vehicle's engine, transmission, and other systems, and alerts the driver if any issues are detected.

With the rise of connected vehicles and the increasing demand for real-time vehicle data, vehicle drivers have long wanted an interface that provides them with the necessary information without having to know the technical jargon and details involved. With an OBD II app, drivers can monitor their vehicle's performance in real time, receive alerts for potential issues, and even diagnose and fix some problems themselves. This can save vehicle owners time and money by reducing the need for expensive mechanic visits.

The development of OBD II apps has been slow for several reasons. One reason is the complexity of the technology and the data it provides. OBD II systems generate a vast amount of data, and interpreting and analyzing this data can be challenging. Developing an app that can effectively communicate with an OBD II adapter and accurately interpret the data it receives requires specialized knowledge and expertise.

Another reason for the slow development of OBD II apps is the lack of standardization in the OBD II system. Although there are industry standards for OBD II, different vehicle manufacturers may use different protocols, making it difficult to develop a single app that works with all OBD II adapters. This means that developers must create different versions of their apps for different vehicle manufacturers, which can be time-consuming and expensive.

Finally, there are regulatory and privacy concerns that developers must consider when building an OBD II app. OBD II systems are subject to regulations such as the Clean Air Act, and developers must ensure that their app complies with these regulations. There are also privacy concerns related to the collection and storage of vehicle data, which must be addressed to ensure that the app does not infringe on the user's privacy rights.

The technology behind OBD II is evolving rapidly, with new advancements in-vehicle connectivity and data analysis. Keeping up with these advancements can be a challenge.

Despite the unattractive challenges relating to OBD II systems, there are still several reasons including, providing drivers with real-time vehicle data, cost savings, environmental benefits, and improved road safety to continue building such systems. As OBD II technology continues to evolve, the development of new and innovative apps can help to unlock even more potential benefits for drivers and the automotive industry as a whole.

Collocating the benefits offered by OBD II systems, an intuitive and user-friendly interface will offer a groundbreaking, user-friendly system in Ghana for use by vehicle owners, enthusiasts and drivers. Vehicle drivers and owners can easily take a glance at an app and see the general performance of the vehicle in the environment and the effect of the environment on the vehicle.

1.5 Organization of the Study

The rest of this study is organized as follows:

Chapter two describes the motivation for this work. Specifically, it discusses the current technologies used for OBD II systems and interfaces in Ghana and the reason why new ways must be sought to better them. Also, existing literature on the design and integration of OBD II systems focusing on related studies in the automotive industry; identifying the progress and drawbacks, how some solutions and integrations were made will be analyzed, identifying the downsides of these implementations.

Chapter three typically focuses on the methodology used in the study, including an overview of the various subsystems used in the proposed design. This chapter provides insights into the approach used to develop the OBD II system and subsystems.

Chapter four focuses on the outcomes of the design and how they are used to evaluate the quality of the suggested implementation. It is essential as it provides insights into the effectiveness of the design and helps to identify any areas for improvement.

Chapter five of the research study is dedicated to discussing the noteworthy difficulties encountered during the study and making suggestions for a more successful future study. It provides an opportunity to reflect on the challenges faced during the research and offers insights into how to improve future studies

Chapter 2

Literature Review

2.1 Introduction

Over time, there has been a major transformation in the automobile industry. Modern vehicles are becoming more complicated, which has prompted the creation of sophisticated diagnostic tools to help with fault finding. The second generation (OBD II) of the On-Board Diagnostics (OBD) system, in particular, has received wide acceptance as a crucial diagnostic tool in the automotive business. OBD II technology saves time and money by enabling mechanics and car owners to rapidly identify issues with a vehicle's systems. OBD II technology has been used more and more frequently in recent years by mechanics and car owners to diagnose and fix their cars.

With an emphasis on its use in car diagnostics, this part of the report analyzes the various uses of the OBD II diagnostic instrument in the automotive industry. The chapter examines the literature that has already been written on the subject, including studies that have examined the OBD II tool's capabilities, constraints, and potential for advancements in the future. The chapter ends by emphasizing the most important research results and pointing out knowledge gaps that require further study.

2.2 Analysis Of Existing Systems

In [1], Engineers are focused on collecting battery pack voltage, battery pack current, motor RPM, and state of charge whose trends can be used to estimate the life of the battery. To power the device, the ignition must be turned on. Also, Images (background, buttons, etc) were designed specifically for a portrait-oriented Google Nexus 7 tablet and the design was specific for Android Version 4.2. The total cost of production was $310

In [2], Engineers sought to reduce the security risk associated with attacks on Bluetooth OBD-II dongles by controlling what access users have via the OBD-II port. Though it was designed for Android as in [1] [3] [4] [5] [6] more parameters were made to be retrieved such as data related to Air Intake, Temperature Ambient air temperature, Engine coolant temperature, Barometric pressure, Fuel pressure, Intake manifold pressure, Timing advance, DTC number, OBD standard, PID check, Equiv ratio, vin. It also utilizes WiFi or Ethernet to enable its web client to run via the application to send and store data unto its server.

[3] takes advantage of the ELM327 and Linkmart 2.0 Bluetooth Transceiver as well as JAVA [7] programming language thus running on virtual machines in order support multiple android instances.

George Roșca [7] takes advantage of the ELM to integrate controller is a chip that helps you connect to the ECU of your mobile equipment and communicate with it through a standard serial connection. It interprets sent ASCII commands and passes them to the OBD. The mobile equipment interrogates the OBD and stores the retrieved information in a mini-database so it can be later forwarded and processed by the remote host.

[8] gears a different function by monitoring using a PC with a Bluetooth board connected to the serial port of the PC. The diagnostic system is developed using Visual C++ MFC and protocol stack of Bluetooth for Windows environment as well as the S3C3410X used for communicating ECU signals.
Malintha Amarasinghe [4] utilizes a Complex Event Processor (CEP) at both the smartphone and the backend to detect and notify unsafe and anomalous events in real-time.

[9] includes a GNSS positioning system. As such, with the use of existing navigation systems or with the use of newly installed equipment, the exact geographic location where a given risk occurs will be identified.

Amith Khandakar [5] reverse engineers data such as the car speed from the OBD-II port and uses the information to control the driver's phone such as preventing them from making or receiving calls at specific speed limits.

[10] focuses on OBD-II Data-Logging Systems for Battery Electric Vehicles. An interactive web interface was developed so that the processed BEV data can be viewed and trended. This tool will allow drivers to perform their diagnostics and gain insights from their vehicles.

[11] aims to propose a platform capable of estimating the amount of carbon dioxide based on sensor readings in vehicles, indirectly contributing to a more proactive city planning based on the monitoring of vehicular pollution. A mobile application installed on a smartphone is used to retrieve the required data. Due to the limitations of the OBD-II scanners, only Versions 2 and 3 of the Bluetooth protocol are supported. A distance between 10 and 15 m to avoid any communication failure.

Jheng-Syu Jhou & Shi-Huang Chen [12] The vehicle information will be transmitted to the cloud computing server via a 3.5G wireless network for fault analysis. Once the cloud computing server detects fault conditions, the proposed system could classify the fault conditions depending on vehicle type and model year.

2.3 Conclusion

Throughout the literature review, it became apparent that previous works in the field of Bluetooth OBD tool application have focused mainly on the hardware components of the technology, with little emphasis on software development. Those that did, limited their work to only android. While the hardware components of

Bluetooth OBD tools are critical, the software aspect plays an equally important role in determining the overall performance and effectiveness of the tool.

Therefore, the lack of attention to software development in previous works is a significant limitation in the field. This gap in knowledge presents a significant opportunity for future research, particularly in the development of efficient and effective software applications that can complement the hardware components of Bluetooth OBD tools. Such software can potentially improve the accuracy and reliability of diagnostic tests and provide valuable insights into vehicle performance and maintenance requirements.

Furthermore, with the rapid advancements in technology, the integration of Bluetooth OBD tools with other emerging technologies such as machine learning and artificial intelligence could unlock even greater potential in this area. Therefore, future research should focus on exploring the software development possibilities of Bluetooth OBD tools and harnessing the full potential of this technology.

Chapter 3

Methodology

This chapter seeks to elaborate on the components and the general design of the entire system, henceforth known as ScanMate. The chapter also focuses on the decision making processes involved in choices of the technologies, components and tools to be used among the various alternatives in its category.
The scope of the project involves the use of an adapter to collect data from the OBD II port of the vehicular system wirelessly via Bluetooth, integrated with ScanMate, the mobile application system which allows for performance monitoring and analysis. Furthermore, the project seeks to enable the vehicle owner to see the performance and data from the vehicle on a smart device.

3.1 Design Objective

We seek to complete the following objectives for the design aspect of this project:

- To design an application interface which displays information fetched by the adapter from the OBD II port
- To design an application which enables users monitor performance and gain data insights
- To design a multi-platform application to facilitate the ease of access to vehicular information.

3.2 Requirements

3.2.1 User requirements

These requirements beset the features and functionalities that are important and expected by the user in the final version of the completed project making up the user interface and experience. The majority stakeholders of this project include mechanics, fleet managers , vehicle owners and car enthusiasts. These are groups of people who are either directly or indirectly going to use the solution for various purposes.
We used various requirement gathering techniques to collect user requirements such as short oral interviews and form surveys.A summary of the user requirements from

the interviews include the following:

1. The application should be compatible with both Android and iOS devices.
2. The app should support multiple OBD II adapters and be able to connect to them seamlessly.
3. The application should be able to read and display real-time data from the OBD II system
4. The app should be able to display diagnostic trouble codes (DTCs) and provide information on how to fix them.
5. The app should be able to display fuel economy and provide suggestions on how to improve it.
6. The app should provide the ability to log and save data for future reference.
7. The app should be user-friendly and easy to navigate, with clear instructions and explanations.
8. The app should be able to provide personalized alerts and notifications based on the user's driving habits and preferences.
9. The app should be able to offer additional features such as trip tracking, performance analysis, and vehicle maintenance reminders.
10. The app should be regularly updated with new features and improvements based on user feedback and industry advancements.

3.3 System Overview

This section is dedicated to the discussion of functional and non-functional requirements of the system

3.3.1 Functional Requirements

These requirements outline the specific functionalities and features that the application must have to satisfy and meet the needs and expectations of its users. The system should have the following functional requirements:

1. The application should connect via wireless bluetooth with the OBD II adapter: This requirement would enable the application to communicate with the vehicle's onboard diagnostic system wirelessly, without the need for a physical connection. This can be particularly useful for users who want to monitor their vehicle's performance in real-time while driving or for mechanics who need to troubleshoot problems on the go.
2. Real-time monitoring of vehicle performance data: The application should be able to collect and display real-time data on various aspects of vehicle performance, such as engine RPM, speed, coolant temperature, and fuel consumption.

3. Diagnostic trouble code (DTC) analysis: The application should be able to read and interpret DTCs generated by the vehicle's onboard diagnostic system and provide clear explanations of what each code means.

4. Fuel efficiency tracking: The application should be able to calculate and display fuel efficiency metrics, such as miles per gallon or liters per 100 kilometers, based on data collected from the vehicle's sensors.

5. Maintenance reminders: The application should be able to track the vehicle's service history and provide reminders for routine maintenance tasks, such as oil changes or tire rotations.

6. Trip logging: The application should be able to log data from individual trips, including distance traveled, average speed, and fuel consumption, to help users track their driving habits and fuel costs.

7. Performance analysis: The application should be able to analyze the collected data to identify areas for improvement in vehicle performance, such as optimizing fuel consumption or reducing engine wear.

8. User customization: The application should allow users to customize the data displayed and the metrics used to track vehicle performance according to their specific needs and preferences

3.3.2 Non-functional Requirements

Non-functional requirements are those that define the characteristics or qualities of the application. These requirements are essential because they ensure that the application is designed and built to meet the broader expectations of its users beyond just its functionalities. Some non-functional requirements may include:

- The application should alert the user if the OBD II adapter is connected or not.
- The application should have fast response times
- It should include secure data transmission of vehicular and user data.
- The data in the application should be accurately and properly displayed
- The application interface should be easy to use and information should be easily accessible
- The application system should have high availability for users to be able to use comfortably at various times

3.3.3 Bluetooth Low Energy

Bluetooth Low Energy (BLE) is useful in building this system by enabling wireless communication between the OBD II adapter device and a mobile device. BLE is a low-power wireless communication technology that can be used to transmit data over short distances. Users can connect to the device using their mobile devices and access real-time data on vehicular performance. BLE can also be used to enable location-based services, such as tracking the location of a vehicle or providing turn-by-turn directions. Additionally, BLE can be used to enable remote diagnostics, allowing mechanics to diagnose and fix problems with a vehicle without having to physically connect to the OBD II device. Overall, BLE is a useful technology for building OBD II applications that can provide users with real-time data about their vehicle's performance and enable remote diagnostics and location-based services.

3.3.4 System Design Architecture

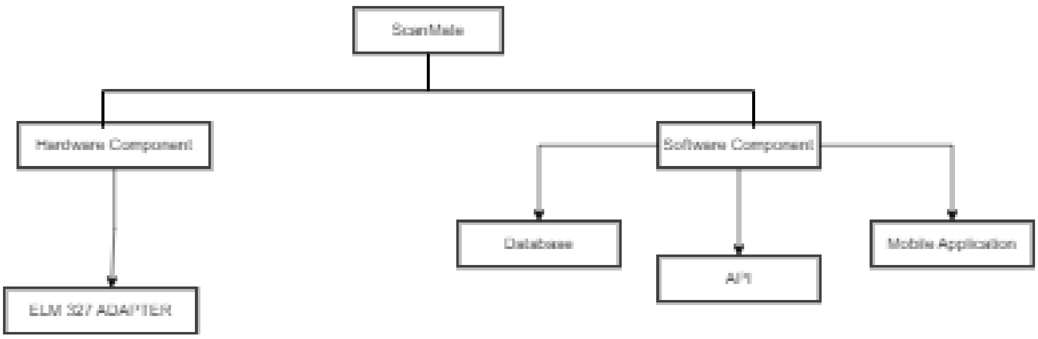

Figure 3.1: System Design Architecture

3.3.5 Component Specification

- Database: The system uses both a MySQL database to better define relationships between users and their entities and MongoDB database for flexibility and ease of integration with applications.
- API: The API is implemented using the Node Javascript Framework. All APIs built with this platform handle all forms of communication with the database. There is also the Google Maps API and Apple Maps API which handles map-related features.
- Mobile Application: This is the platform the user will interact with. Flutter will be used to build the final product, and this is because it is cross-platform which means it will be able to build applications for different environments with one codebase.
- Microcontroller: The ELM 327 is going to be used as it is popular with a lot of protocols and has good documentation for easier and better development and low power consumption.

3.3.6 Software System Design

This section is devoted to the emphasized breakdown of the software section of the project. For storage of the vehicular and personal details and the performance history of the vehicle and the received data from the OBD II port, various databases were used to contain the data. The system's software component will provide an interface for users to access vehicular information and performance history, provide feedback and perform diagnostics. This component will be made up of three main parts: The mobile application , the application programming interface (API) and the database. These three will work seamlessly with the OBD II adapter to ensure effective ,correct and efficient communication between the components

Mobile Application

This system will include a cross-platform application that users can interact with to access information. The mobile phone application will be developed using Flutter with Dart. It is used to build both the user interface and develop the user interaction. A fast, responsive and user-friendly user interface will be the a major characteristic of the application. It will be used to control the flow of information between the mobile application and the API which holds all data for future access and reference from different locations, in turn making the application with dynamic data.

Application Programming Interface

The API is going to be used in the mobile application for communicating between other services and components of the system and other outside services that may be integrated making it easier to build a robust and comprehensive system. An API enables communication between the various components of the software component, i.e the database and the mobile application. It also enables the user to communicate to the OBD II adapter to communicate commands to the vehicle. NodeJS is ideal for building scalable, high performance applications. Its speed and rich ecosystem make it a good choice to build APIs for the system.

Database

The database is the primary storage unit of the software component. It is cloud based to ensure that users and stakeholders can access information from different locations. Both the SQL and NoSQL databases will be used in the data store for the system. This is because of the difference in structured and unstructured data and capitalizing on the flexibility and ease of use in NoSQL systems and the robust nature and structured format of SQL systems.
The database will hold information such as :

- Vehicular information from OBD II adapter
- User information and account credentials
- Data analysis and user customizations or preferences

Chapter 4

Testing and Evaluation

Chapter four focuses on the outcomes of the design and how they are used to evaluate the quality of the suggested implementation. It is essential as it provides insights into the effectiveness of the design and helps to identify any areas for improvement.

Chapter 5

Conclusion and Recommendations

Chapter five of the research study is dedicated to discussing the noteworthy difficulties encountered during the study and making suggestions for a more successful future study. It provides an opportunity to reflect on the challenges faced during the research and offers insights into how to improve future studies.

Appendix

5.1 Interview Survey Questionnaire

1. What kind of vehicle do you currently use?
2. Have you ever used an OBD scanner or app before? If yes, what for?
3. What are some specific features or functionalities you would like to see in an OBD II app?
4. What types of vehicle data do you find most useful or important to monitor (e.g. fuel efficiency, engine performance, emissions, etc.)?
5. How often do you perform vehicle maintenance or diagnostics?
6. What type of data visualization would you prefer (e.g., graphs, charts, tables, etc.)?
7. How important is it to you that the app is user-friendly and easy to navigate?
8. Are there any particular challenges or pain points you experience with existing OBD II apps that you would like to see addressed in a new app?

Bibliography

[1] J. Allande, P. Tyler, and E. Woodruff, "Nissan leaf on-board diagnostic bluetooth utility," 2013.

[2] A. Tahat, A. Said, F. Jaouni, and W. Qadamani, "Android-based universal vehicle diagnostic and tracking system," in *2012 IEEE 16th International Symposium on Consumer Electronics*, 2012, pp. 137–143. DOI: 10.1109/ISCE.2012.6305105.

[3] M. Kalmeshwar and K. Nandini Prasad, "Development of on-board diagnostics for car and it's integration with android mobile," in *2017 2nd International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS)*, 2017, pp. 1–6. DOI: 10.1109/CSITSS.2017.8447540.

[4] M. Amarasinghe, S. Kottegoda, A. L. Arachchi, S. Muramudalige, H. M. N. Dilum Bandara, and A. Azeez, "Cloud-based driver monitoring and vehicle diagnostic with obd2 telematics," in *2015 Fifteenth International Conference on Advances in ICT for Emerging Regions (ICTer)*, 2015, pp. 243–249. DOI: 10.1109/ICTER.2015.7377695.

[5] A. Khandakar *et al.*, "Portable system for monitoring and controlling driver behavior and the use of a mobile phone while driving," *Sensors*, vol. 19, no. 7, p. 1563, 2019.

[6] C. S. Patel and J. A. Gaikwad, *Iot based augmented reality application for diagnostic vehicle's condition using obd-ii scammer*, 2020.

[7] G. Roşca *et al.*, "Remote equipment diagnosis using bluetooth communication," in *2011 RoEduNet International Conference 10th Edition: Networking in Education and Research*, 2011, pp. 1–5. DOI: 10.1109/RoEduNet.2011.5993681.

[8] K.-H. Choi, H.-H. Lee, Y.-C. Lee, T.-K. Kwon, and S.-C. Lee, "Self-diagnostic signal monitoring system of kwp2000 vehicle ecu using bluetooth," *:*, pp. 132–137, 2004.

[9] J. Wideberg, P. Luque, and D. Mantaras, "A smartphone application to extract safety and environmental related information from the obd-ii interface of a car," *International Journal of Vehicle Systems Modelling and Testing*, vol. 7, no. 1, pp. 1–11, 2012.

[10] C. Ramai, V. Ramnarine, S. Ramharack, S. Bahadoorsingh, and C. Sharma, "Framework for building low-cost obd-ii data-logging systems for battery electric vehicles," *Vehicles*, vol. 4, no. 4, pp. 1209–1222, 2022.

[11] M. Silva, G. Signoretti, J. Oliveira, I. Silva, and D. G. Costa, “A crowdsensing platform for monitoring of vehicular emissions: A smart city perspective,” *Future Internet*, vol. 11, no. 1, p. 13, 2019.

[12] J.-S. Jhou and S.-H. Chen, “The implementation of obd-ii vehicle diagnosis system integrated with cloud computation technology,” in *Intelligent Data analysis and its Applications, Volume I: Proceeding of the First Euro-China Conference on Intelligent Data Analysis and Applications, June 13-15, 2014, Shenzhen, China*, Springer, 2014, pp. 413–420.