ComTech: Computer, Mathematics and Engineering Applications, 14(2), December 2023, 129−141 P-ISSN: 2087-1244
DOI: 10.21512/comtech.v14i2.9138 E-ISSN: 2476-907X

Mobile-Based Car Diagnostic Application Using Onboard Diagnostic-II Scanner

Karto Iskandar[1]*, Alfred Tambayong[2], Muhammad Rafif Fawwaz Mulya[3], Steven Cendra Elfanlie[4], and Maria Grace Herlina[5]

[1-4]Computer Science Department, School of Computer Science, Bina Nusantara University
Jln. K. H. Syahdan No. 9, Jakarta Barat 11480, Indonesia
[5]Management Department, BINUS Business School Undergraduate Program, Bina Nusantara University
Jln. K. H. Syahdan No. 9, Jakarta Barat 11480, Indonesia
[1]Karto_i@binus.edu; [2]alfred.tambayong@binus.ac.id; [3]muhammad.mulya001@binus.ac.id;
[4]steven.elfanlie@binus.ac.id; [5]herlina01@binus.edu

Received: 27th October 2022/ **Revised:** 6th March 2023/ **Accepted:** 6th March 2023

How to Cite: Iskandar, K., Tambayong, A., Mulya, M. R. F., Elfanlie, S. C., & Herlina, M. G. (2023). Mobile-Based Car Diagnostic Application Using Onboard Diagnostic-II Scanner. *ComTech: Computer, Mathematics and Engineering Applications, 14*(2), 129−141. https://doi.org/10.21512/comtech.v14i2.9138

Abstract **-** Mobile applications today serve as versatile tools across diverse sectors, enhancing human productivity through specialized software on electronic devices. Implementation of the mobile application can also be applied to vehicles, with inspection and checking functions assisted by the Onboard Diagnostic-II (OBD-II) scanner. The research aimed to develop an integrated mobile application that utilized the OBD-II scanner and Data Acquisition System (DAS) to monitor vehicle health and provide timely service reminders. Vehicle information was taken by the DAS process into a Diagnostic Trouble Code (DTC) from the vehicle itself. The method applied the waterfall model, which consisted of communication, planning, modeling, construction, and evaluation. The problem analysis and requirements gathering for developing the application involves the interview method and Google Forms-generated questionnaires with 101 responses. Then, the research used OBD-II series ELM327 and ELM 327 IC devices for testing. The research results in an application developed for vehicle diagnostics using a recommendation system through notifications that provide vehicle health information and service time reminders to users. This application consists of eight modules, with the main module being able to provide recommendations for vehicle owners. These recommendations are helpful for users to maintain the health of their vehicles regularly. Further research is recommended to enhance the development of the application, aiming to create a more comprehensive user interface.

Keywords: mobile-based application, car diagnostic, Onboard Diagnostic-II Scanner

I. INTRODUCTION

The importance of vehicles in human life cannot be separated from the functions provided, especially motorized vehicles like cars, buses, commercial vehicles, transport vehicles, and others. Therefore, the vehicle is no less important in daily human life as it can help and facilitate the mobility of humans quickly. The presence of vehicles nowadays, primarily cars, varies from wheel drive type, engine type, fuel type, and type of car based on class and style. From the wheel drive type, various types are usually used in vehicles differently. Generally, there are four types of wheel drive: Front Wheel Drive (FWD), Rear Wheel Drive (RWD/4×2), All Wheel Drive (AWD), and Four-Wheel Drive (4×4) (Ju et al., 2020). There are also several types of engines used in the car. These are the petrol engine, diesel engine, and Plug-in Hybrid Electric Vehicle (PHEV). Moreover, full Battery Electric Vehicle (BEV) engines use fully electric motors (He, Shou et al., 2022).

Having a car for daily mobility can lighten human activities and occupations. However, the vehicle must be maintained and serviced periodically for better condition. Hence, it is always ready to be used. Car maintenance, like a requirement and obligation, must be carried out regularly and periodically (Arena et al., 2020). However, sometimes, maintenance and checking the car's condition can be difficult for several people. For some of those who own cars, it is still unclear, and they do not understand how to check their vehicle properly and correctly. Car owners sometimes only do a quick check and take general care of the car or keep the car's physical condition in top condition.

*Corresponding Author

Nowadays, Internet development is growing fast, and most aspects of human life cannot be separated from it. In Indonesia, the term Internet of Things (IoT) is no stronger anymore. Even today, by using additional scanner tools like Onboard Diagnostic-II (OBD-II) can display an interface from the Electric Control Unit (ECU) about car detail information (Nugroho et al., 2018; Nurcahya et al., 2022). Usually, the OBD-II scanner works on the fourth layer (transport layers) and third layer (control area network) in the car or other vehicles. The system applied to the car or other vehicles is monitored using parameters that occur in the vehicle system and are stored in memory. If there are troubles in the car or vehicle, the car owner will be warned via Malfunction Indicator Light (MIL). Using an OBD-II scanner for independent car diagnostics eliminates the need to take the vehicle to a workshop or seek specialized treatment.

OBD scanner was first developed by the California Air Resources Board (CARB), Environmental Protection Agency (EPA), and Society of Automotive Engineers (SAE) in 1996 in the United States of America. It became a standard system for United States countries (Ling et al., 2020). OBD scanner tools and media connectors are like smartphones or computers. Via the Data Acquisition System (DAS) process, data from the vehicle (vehicle information) will be taken and recorded. The mediator is required to connect appropriately and display car information with an OBD scanner tool and smartphone. The mediator in question is an application that runs on the smartphone with a process via media connectivity provided by a Bluetooth provider.

Previous research explains the development of generic OBD devices, their works with different vehicles, and OBD II standard-based. The vehicle test uses the Land Rover Defender as a tester device, and the development process uses OBD scanner tools for connecting with the vehicle's ECU. This development uses ELM327, MCP2251 CAN Transmitter hardware, and a laptop as components hardware for high-level design. This process shows the real-time vehicle system status, including vehicle speed, engine Revolutions Per Minute (RPM), throttle position, battery voltage, engine coolant temperature, and Diagnostic Trouble Code (DTC) (for different vehicles) (Niazi et al., 2013).

Another previous research uses algorithms, such as Random Forest (RF), Support Vector Machine (SVM), and Gradient Boosting Decision Tree (GBDT), to identify data from OBD and compare emission data from trucks. Its result is a data comparison testing of the three algorithms. These algorithms are used to calculate the accuracy level of truck emissions by comparing OBD-II data. The data are used as a reference to evaluate whether the vehicles have high Nitrogen Oxides (NOx) for exhaust emissions (He, Zheng, et al., 2022).

Previous research also shows how to detect bottlenecks using an OBD-II scanner based on a Cyber-Physical System (CPS). Arduino Mega is used as a sensor to see information and vehicle conditions to calculate the cost it has spent on impact bottlenecks. The parameters used are vehicle information such as speed, RPM, Mass Air Flow (MAF), Intake Air Temperature (IAT), Air-Fuel Ratio (AFR), Global Positioning System (GPS), and time. The information obtained is data processed on a cloud platform and connected over the Internet with validation and calibration of mathematical traffic models and traffic simulation software (Zeb et al., 2022).

Table 1 Comparison Similar Applications Features

No	Features & other Comparison	Torque (Lite & Pro Version)	Car Scanner ELM327 OBD2	inCarDoc (ELM327 Scanner)
1.	Show real-time machine data	✓	✓	✓
2.	Showing Error Code/DTC	✓	✓	✓
3.	Dashboard Customizable	x	✓	x
4.	Show time fuel consumption data	✓	✓	✓
5.	Support Head Up Display (HUD)	✓	x	x
6.	Reset Error Code	✓	✓	✓
7.	Version Apps	Varies with device	1.91.2	7.6.9a
8.	Required Android	Varies with device	5.0 or higher	4.0.3 and higher
9.	Release on	Mar 25, 2010	Jan 23, 2018	Jan 28, 2012
10.	Content rating	3+	3+	Everyone
11.	Downloads	10M+	10M+	5M+
12.	Reviews	47.7K	165K	38.8K
13.	Average Ratings	4.5★	4.7★	4.1★
14	Data Safety	Not Available	Data Shared	No Data Shared

Comparison with other similar apps from the Play Store is done based on features, number of downloads, rating reviews, and multiple comparisons. Comparison of similar applications can be seen in Table 1. No application gives notifications for reminder vehicle service and clear fault codes. Hence, the researchers want to develop the application with a recommendation system via notification.

Based on current conditions, it is found that modern people do not understand the situation of their vehicles. As a result, they are less concerned with the health of their vehicles. However, every vehicle needs regular maintenance. Therefore, the research objective is to develop an application that can remind vehicle owners of the condition of their vehicles is needed.

II. METHODS

The application development process uses the Waterfall model method. The process of the Waterfall model consists of several phases that include communication, planning, modeling, construction, and deployment (Pressman & Maxim, 2020). The Waterfall model is implemented with detailed stages, as depicted in Figure 1. Additionally, each phase comprises several stages.

At the communication stage, there are four stages: (creating project group), (defining project idea, specifications, and technical), (source and data collection), and (task and role distribution). Then, in the planning stage, there are also four stages:

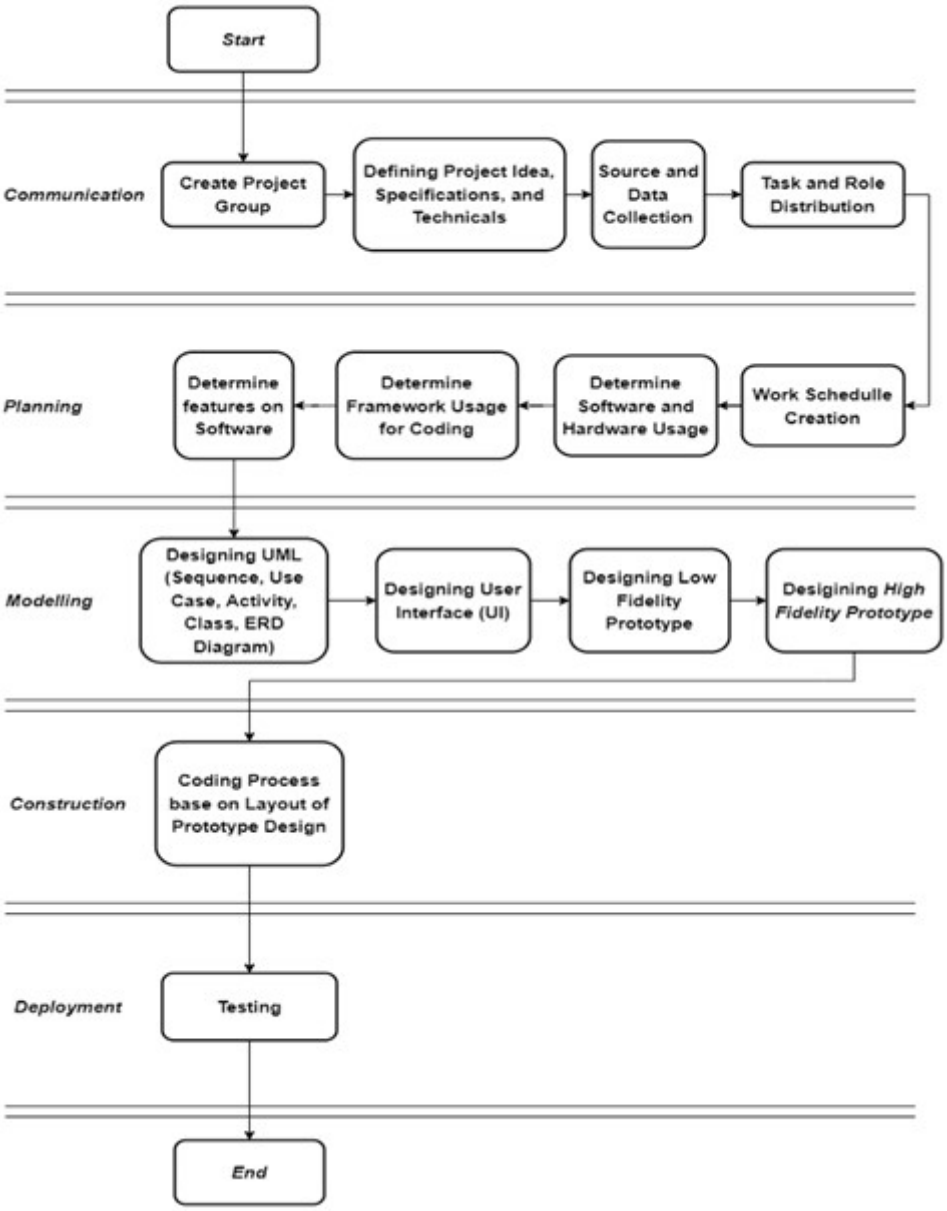

Figure 1 Waterfall Model Diagram

(determining feature on software), (determining framework usage for coding), (determining software and hardware usage), and (work schedule creation). Similarly, in the modeling stage, there are also four stages: (designing UML), (designing user interface), (designing low fidelity prototype), and (designing high fidelity prototype). The next stage is the construction and coding process based on the layout of the prototype design. Finally, the deployment process begins with the testing process (Pressman & Maxim, 2020).

The problem analysis and requirements gathering for the development application are conducted using the interview method and Google Forms-generated questionnaires. The questionnaires provide several questions, including name, age, profession, frequency of car use, vehicle years, diagnosed car type, application features, and interest in the application. For the best result from analyzing the questionnaires for developing applications, data questionnaires are used as parameter features in the apps and diagnose car type. Interviews were conducted with 3 workshop owners and questionnaires were distributed to 101 respondents.

The research begins by conducting a literature study. DAS is the process of sampling. The signals measure real-world physical conditions and convert the resulting samples into digital numeric values that a computer or another device can manipulate. DAS is abbreviated by the initializes as DAS, DAQ, or DAU. Software programs usually control DAS applications. They are developed using programming languages, such as Assembly, BASIC, C, C++, C#, Fortran, Java, LabView, Lisp, Pascal, and others (Jamal & Wenzel, 1995). Data acquisition begins with the physical phenomenon or physical property to be measured, including temperature, vibration, light intensity, gas pressure, fluid flow, and force. The task of performing such transformations falls on devices called sensors. The completed DAS consists of DAQ hardware, sensors and actuators, signal conditioning hardware, and a computer running DAQ software. This DAS component is shown in Figure 2.

In DAS, hardware is typically used as a media connector to connect a signal to a computer. It is a module that connects to computer ports, such as Serial and USB and other slots. The other slots are S-100, AppleBUS, ISA, MCA, PCI, and PCI-E that are integrated into PC motherboards, such as Computer-Aided Measurement and Control (CAMAC), including Nuclear Instrumentation Module (NIM) and Versa Module Eurocard (VME) (Dong & Gaofei, 2019; Riswal et al., 2020). In the DAS process, the sensor is important for obtaining a numeric value in digital form. The sensor is also known as a transducer, which refers to a changing phenomenon in the real world. Generally, sensors used in the DAS are thermocouples, thermistors, RTD Temperature, accelerometers, and strain gauges. When using the sensor, the part of the core in selecting the sensor is essential, such as the sensor's accuracy, signal conditioning, and strength (Malini et al., 2020).

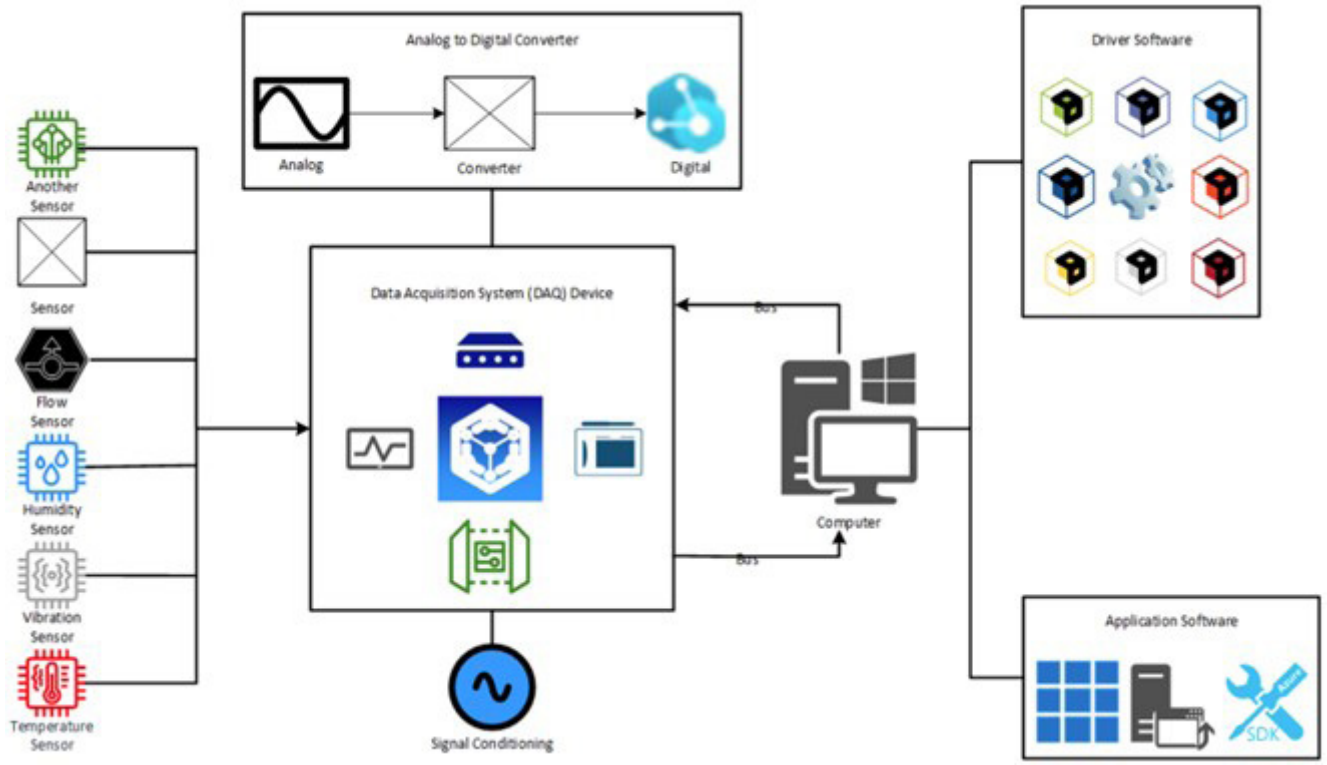

Figure 2 Data Acquisition System (DAS) Components

Android Studio is Integrated Development (IDE) for Google's Android operating system. Android Studio is built on JetBrains' IntelliJ IDEA software and is designed to develop Android applications. Android Studio is based on the IntelliJ IDEA, a Java-integrated development environment for software. It incorporates its code editing and developer tools (Gerber et al., 2015). The Application Programming Interface (API) is required to develop a car diagnosing application for vehicles in Android Studio. Meanwhile, in OBD scanner tools, it is required to connect and display the OBD library between the vehicle and the smartphone via the scanner tool.

OBD-II is a second version of the predecessor version of the first-generation standard OBD. Generally, in the automotive world, OBD refers to the diagnostic capability to create, diagnose, and report independently. The system contained in OBD provides access to sub-systems in vehicles. The amount of diagnostic information available via OBD has varied widely since its introduction in the early 1980s versions of Onboard Vehicle Computers (OVC) (Gallardo, 2018). OBD-II is an improvement over OBD-I in both capability and standardization.

The OBD-II standard specifies the type of diagnostic connector and its pinout, the electrical signaling protocols available, and the messaging format. Signal and electrical protocols are available in message formats, and vehicle parameters are to be monitored along with coding for each data. A pin in the connector provides power for the scan tool from the vehicle battery separately. In addition, the OBD-II standard provides an extensible list of DTCs. As a result of this standardization, single-device OBD tools can query the onboard computer(s) in any vehicle. Emissions requirements prompt OBD-II standardization. Though only emission-related codes and data are required to be transmitted through it, most manufacturers have made the OBD-II data link connector the only one in the vehicle through which all systems are diagnosed and programmed (Gallardo, 2018).

Parameter IDs (PIDs) are codes used to request vehicle data and used as a diagnostic tool. Standard Automotive of Engineering (SAE) standard J1979 defines many OBD-II PIDs. Generally, PIDs are coded as state-mandated emissions inspections. The status of the parameter condition in the vehicle is obtained by OBD via PID request. Request and parameter standards are supported by the Engine Control Unit (ECU), which is a standardization by SAE J1979 (Akhibi & Akingbade, 2021). Standard provides a parameter list based on standardization from formula to converting hexadecimal data into data of DTC in J1989.

In 1996, light-duty vehicles less than 8,500 lb. (3,900 kg) and vehicles less than between 8.500–14.000 lb. (3.900–6.400 kg) were required to follow SAE roles to be accessed through a standardized data link connector defined by SAE J1962. In 2010, Heavy-duty vehicles greater than 14.000 lb. (6.400 kg), especially in the US, were allowed to support OBD-II diagnostics through SAE standard J1939-13. Some heavy-duty trucks in North America used the SAE J1962 OBD-II diagnostic connector common with passenger cars, notably Mack and Volvo Trucks. However, they also used 29-bit CAN identifiers (Martinelli et al., 2021). DTC uses ten modes or service codes to support accurate diagnostic systems.

Table 2 PIDs on Service/Mode 01

PIDs	PID	Byte	Description	Min.	Max.	Units	Formula
0C	12	2	Engine Speed	0	5.388	rpm	256 + B/4
0D	13	1	Vehicle Speed	0	255	km/h	A
0A	10	1	Fuel Pressure	0	765	kPa	3A
5C	92	1	Engine Oil Temperature	-40	210	°C	A-40
67	103	3	Engine Coolant Temperature	-40	215	°C	[A0]== Sensor 1 Supported [A1]== Sensor 2 Supported Sensor 1:B-40 Sensor 2: C-40
2F	47	1	Fuel Tank Level Input	0	100	%	100/255×A
A6	166	4	Odometer	0	429,4	km	$(A(2^{24})+B(2^{16})+C(2^{8})+D)/10$
42	66	2	Control Module Voltage	0	15,5	v	256A+B/1000
1F	31	2	Run Time Since Engine Start	0	65.535	s	256A+B
04	4	1	Calculated Engine Load	0	100	%	100/255×A or A/255

For vehicle manufacturing, it is not required to support all modes or services from DTC. Each manufacturer can determine an additional mode/service for their vehicle. SAE J2190 defines service 22 for Ford/GM and service 21 for Toyota (Martinelli et al., 2021).

DTC is a code for showing trouble that happens in the vehicle system. For PIDs standard from SAE J1979, there are ten modes/services, each mode with each PID (hex). Service 01 and service 02 in basic are identical, except for the difference between service 01 and 02. Service 01 tells the real information, and service 02 just provides a snapshot from the same data taken when there is diagnostic trouble or the last DTC (Martinelli et al., 2021). There are 169 PIDs in service 01, and Table 2 shows 10 examples of PIDs code in service 01.

The ELM327 is a programmed microcontroller to translate the OBD interface. The original ELM327 is implemented on the PIC18F2480 microcontroller from Microchip Technology (BinMasoud & Cheng, 2019). ELM327 is one of a family of OBD translators from ELM Electronics Corporation. The ELM327 abstracts the low-level protocol and presents a simple interface that can be called via a Universal Asynchronous Receiver-Transmitter (UART). ELM 327 Integrated Circuit (IC) is responsible for handling the OBD-II communication and translating the data from the vehicle's computer into a format that can be understood and displayed by diagnostic tools, software, or applications. It refers to the specific implementation of the ELM327 chip within a device or circuit (BinMasoud & Cheng, 2019). This microcontroller is shown in Figure 3.

Utilizing the ELM327 programmed microcontroller extends to various vehicle diagnostic functions, including reading diagnostic trouble codes, deactivating MILs, and displaying real-time sensor data, such as engine RPM, load value, coolant temperature, fuel system status, vehicle speed, short/long-term fuel data, intake pressure, timing advance, air flow rate, oxygen levels, voltage, and fuel pressure (Pranjoto et al., 2017). During ELM327 operation and diagnosis, ELM327 must support certain protocols.

ELM327 is for sale in a dongle form with dimensions like a flash disk. When ELM Electronics sells version 1.0 of its ELM327, it does not enable the copy protection feature of the Peripheral Interface Controller (PIC) microcontroller. Consequently, anyone can buy a genuine ELM327 and read ELM's proprietary binary microcontroller software using a device programmer. As a result, ELM327 copies are widely sold in devices claiming to contain an ELM327 device, and problems have been reported with the copies (Macias-Bobadilla et al., 2020). This diagram is shown in Figure 4.

Figure 3 ELM327 (left) and ELM 327 IC (right)

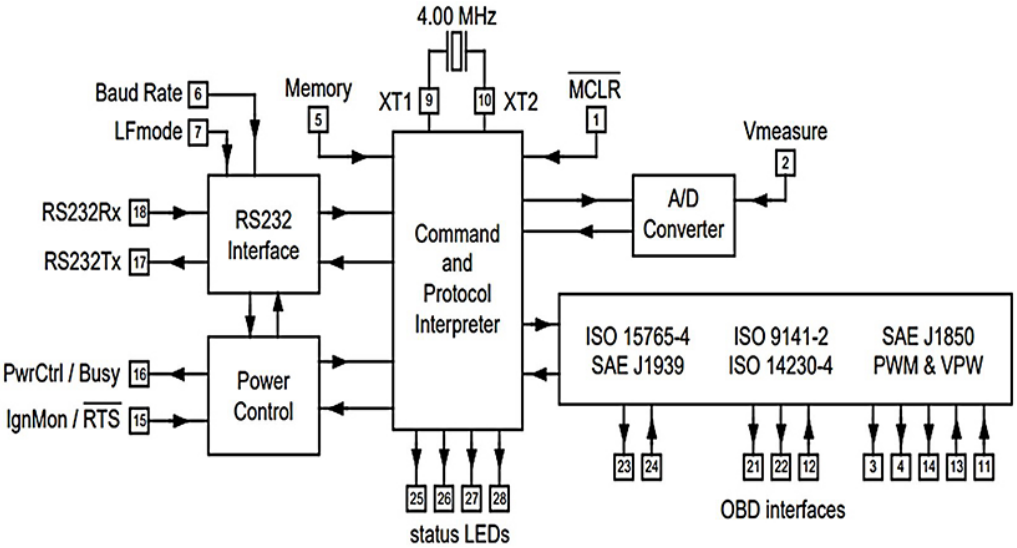

Figure 4 Block Diagram of OBD-II ELM327

During the system development process, a variety of software and equipment tools are employed to facilitate application development. The software tools utilized encompass Android Studio, Java Programming Language, OBD-Java, Bluetooth API, as well as Figma Tools and Draw.io. In parallel, the equipment and tools enlisted for this endeavor include the ELM327 OBD-II Scanner, the POCO F3 Pro Smartphone, the Mazda Biante 2.0L 2WD 6AT (151 HP) vehicle, and standard laptops. A detailed breakdown of the software and equipment/tools employed, along with their respective functions, is presented in Table 3.

III. RESULTS AND DISCUSSIONS

The research begins by creating a survey in the form of a questionnaire and distributing it to respondents to collect data. The data obtained from the questionnaire provides valuable insights into the habits of respondents regarding the routine diagnosis of their vehicles. Among the 101 responses, it is noteworthy that a substantial majority, approximately 56,4%, equivalent to 57 respondents, indicates that they often opt for diagnosing their vehicles through professional auto repair services. The result suggests that many respondents place importance on regular vehicle diagnostics, possibly as part of a proactive maintenance approach to ensure their vehicles remain in good working condition.

The data further reveal that there is a segment of respondents who opt for less frequent diagnosis at auto repair shops, with approximately 31,7% of the participants or 32 individuals falling into this category. This group may choose to diagnose their vehicles at auto repair shops only when specific issues or concerns arise, reflecting a less proactive approach to vehicle maintenance.

Then, a smaller portion of respondents, constituting approximately 11% or 12 individuals, indicate that they never opt for vehicle diagnosis at auto repair shops. This finding may suggest that these respondents prefer to handle vehicle diagnostics themselves or do not consider it a priority in their vehicle maintenance routine.

Upon analyzing the data, it is evident that a significant number of respondents prioritize regular vehicle diagnosis, underlining the importance of professional auto repair services in ensuring the overall health and performance of their vehicles. This insight highlights the various approaches people take when it comes to vehicle maintenance and suggests opportunities for further research into the factors that influence their choices. The result from the questionnaire is shown in Figure 5.

Table 3 Software and Equipment

No	Type	Software/Tools Equipment Name	Function
1	Software	Android Studio	Develop android mobile applications
2	Software	Java Programming Language	Code and develop software applications
3	Software	OBD-Java and Bluetooth API	Develop API code and Bluetooth connection
4	Software	Figma Tools & Draw.io	Design software interface
5	Tools	ELM327 OBD-II Scanner	Interpret data from the vehicle's onboard computer system
6	Tools	POCO F3 Pro Smartphone	Develop and test android mobile applications
7	Tools	Mazda Biante 2.0L 2WD 6AT (151 HP)	Test vehicle and connect to software applications
8	Tools	Laptop	Code, design, and develop software applications

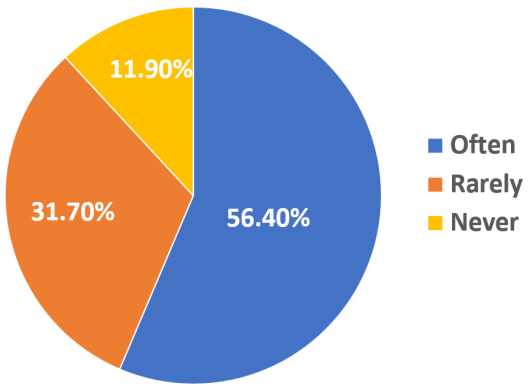

Figure 5 Responses of Routine Diagnose

The next question is about the features provided in the application development process. This question aims to know what features are most liked by respondents. The question is that if the MyCar application provides several features, what features that they think are most useful. With a total of 101 responses and an array of features to choose from, the data are carefully structured to provide insights into the order of preference, offering a comprehensive overview of which features garnered the most preference.

It is intriguing that among the respondents, the feature with the highest favor is "Vehicle Data Record," which captures the attention of approximately 61,8% of the participants, equivalent to 62 respondents. This finding underscores the significance of having the capability to record and monitor vehicle data. It is a feature that contributes to enhancing users' understanding of their vehicle's performance and condition.

Closely following as a highly favored feature is "Car Maintenance Notification," with a substantial 60,7% of respondents, translating to 61 individuals. They express their preference for this functionality. This result points to the importance of timely maintenance reminders and notifications in ensuring the longevity and reliability of their vehicles.

Furthermore, the "Workshop Location Search" feature secures a notable preference, with 57,3% of respondents, or 58 individuals, recognizing its utility. It suggests that users highly value the convenience of locating nearby workshops and service centers for their vehicle maintenance needs.

The "Interactive Theme Display" feature is selected by 49,4% of participants (50 respondents). It means a considerable interest in personalizing and customizing the application interface to suit individual preferences. Next, other features, such as "GPS Tracking", "Vehicle Configuration", and "Others" features, still represent notable components of the MyCar application's offerings while receiving slightly lower preference percentages. Therefore, from analyzing the data, the "Vehicle Data Record" is the most liked feature. This result is shown in Figure 6.

Based on the questionnaire results, the researchers start to develop the application. Next, the research provides a use case diagram shown in Figure 7. The use case diagram designed for the MyCar application only involves one actor in using the system, which is the user. The user has eight use cases as follows. First, it is a use case of connecting the OBD-II scanner to the application. The user must connect the application to the OBD-II to get information. Second, in the use case of viewing OBD data, the application displays OBD data from the user's vehicle. Third, in the use case of viewing vehicle data, the application shows the user's vehicle data in real-time. Fourth and fifth, there is a use case of clearing and viewing fault code. Users can view and clear the fault code from the error code in vehicle malfunction. Sixth, in the use case of viewing workshop location, the users can find and go to the nearest workshop from their location. Seventh, it is the use case of receiving notification. The user receives a notification about the fault code and service reminder. Last, in the use case of changing the control on the setting menu, the user can change the setting in the application.

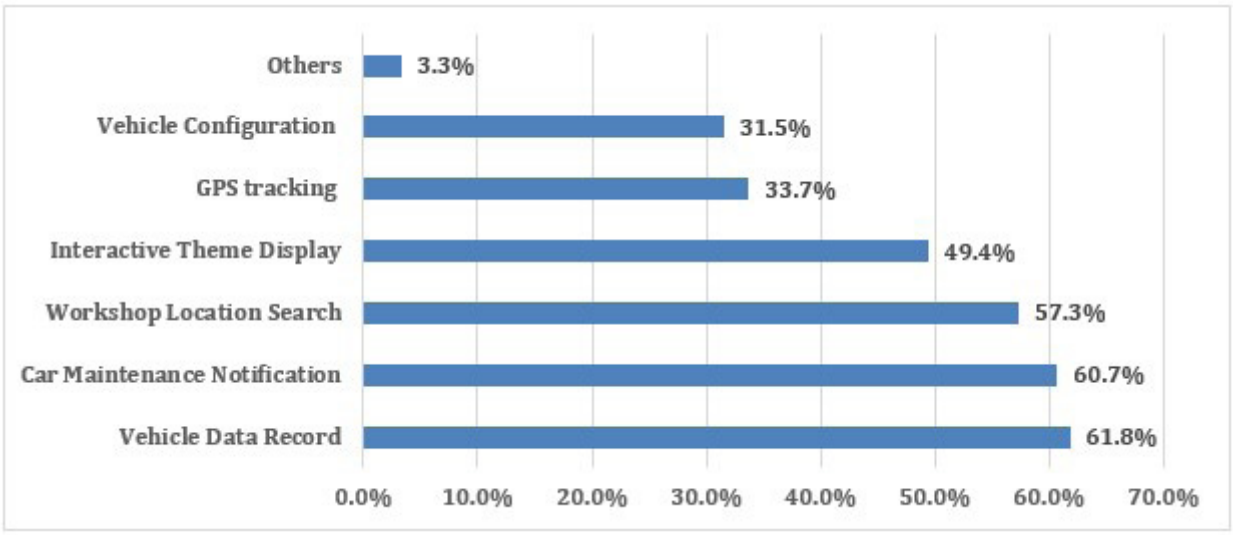

Figure 6 Responses of Feature Choices

The class diagram of the MyCar application that contains classes, relations, operations, attributes, and cardinalities between classes is shown in Figure 8. There are 13 classes in the class diagram of the MyCar application, and each class has its respective attributes. Furthermore, each class has a cardinality between one another, such as zero to one, zero too many, and one too many. Based on the class diagram, the application is developed and tested to connect to the vehicle and OBD-II device via Bluetooth.

For the discussion of developing the application process, the result from the development process is shown in Figure 9. It shows the screenshot of the notification and reminder. The MyCar application offers a user-friendly solution for car owners aiming to enhance their vehicle management and upkeep routines. Individuals can access a range of advantages by integrating the application with an OBD-II scanner. Among these benefits is the ability to monitor vehicle conditions in real-time, facilitated by swift and effective interpretation of fault codes. This feature provides users with immediate insights into their cars' health, enabling prompt identification and resolution of issues and ultimately contributing to a safer and more dependable driving experience.

Furthermore, the MyCar application extends its utility by providing users with convenient access to workshop services for vehicle repairs and maintenance. It establishes a meaningful connection between car owners and automotive service providers, streamlining the process of finding reliable repair services and minimizing downtime and inconvenience when issues arise. Hence, it not only preserves valuable time and resources but also promotes a comprehensive approach to vehicle maintenance.

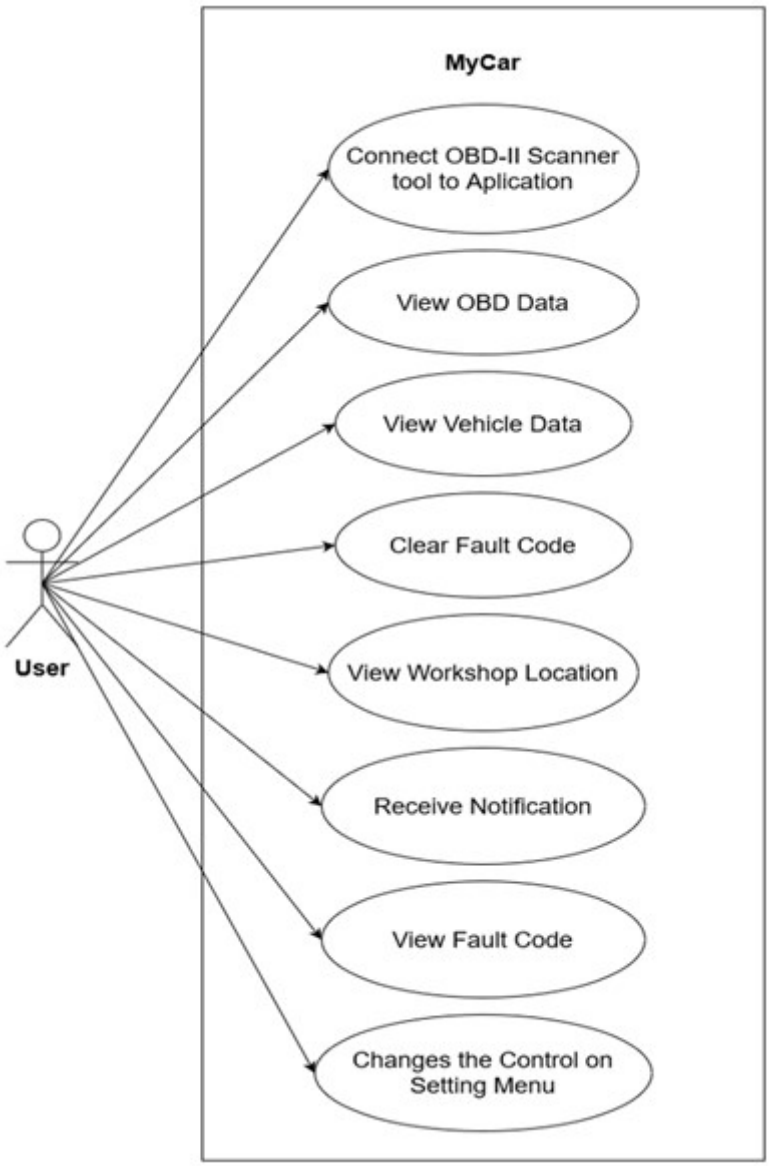

Figure 7 Use Case Diagram

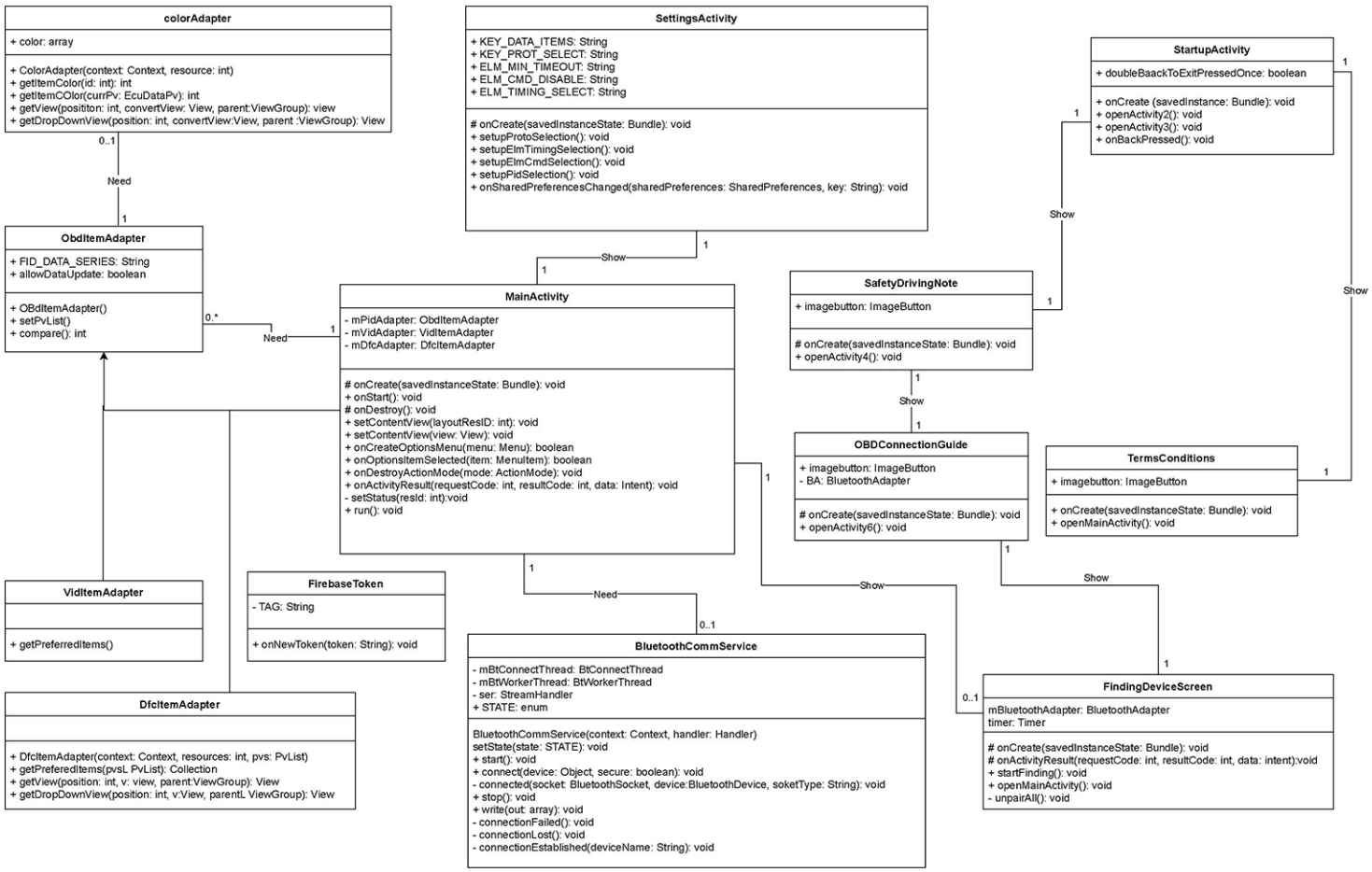

Figure 8 Class Diagram

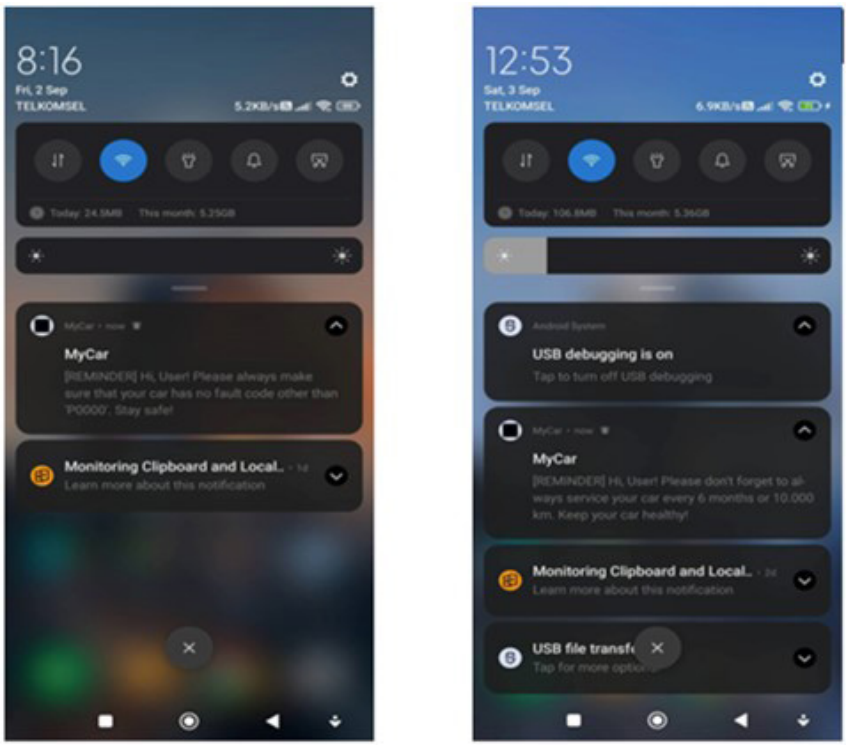

Figure 9 Notification for Clear Fault Code (Left) And Reminder Service Vehicle (Right)

Figure 10 shows the OBD data from vehicles. The data are coolant temperature, engine RPM, vehicle speed, and intake air temperature. However, the data are not permanent. The data are just real-time data and can be changed in vehicle data. Next, the data are obtained from its exact data like Vehicle Identification Number (VIN) and the number of VIN items from OBD-connected devices to explain vehicle information. The explained fault code is fault code, error code, or DTC. The DTC data have meaning in each part of the number and alphabet. For example, P0302 consists of P as a Powertrain, 0 as the standard code from International Organization for Standardization (ISO), and 3 as affected by the sub-system of 02 as a specific code (exact problem). Every fault code can be searched by an online engine with a long press of the fault code. The significance of the research is substantial. The application's capacity to identify and interpret car issues by reading fault codes stored in vehicle systems has the potential to transform how car owners approach and execute vehicle maintenance practices. The screenshot of OBD data and fault code information are shown in Figure 10.

IV. CONCLUSIONS

The conclusions are obtained through the application development process regarding the problem studied through surveys and questionnaires. The conclusions drawn from the application development process have significant implications for the automotive industry and vehicle owners. The research findings indicate that the application is feasible and highly beneficial for individuals who own cars. This groundbreaking application can be seamlessly implemented by car owners who seek to enhance their vehicle management and maintenance practices. By connecting the application with an OBD-II scanner, users can unlock many advantages. One such benefit is the real-time monitoring of their vehicle's condition through the swift and efficient interpretation of fault codes. This feature gives users immediate insights into their car's health, allowing them to identify and address issues promptly and ultimately contributing to safer and more reliable driving experiences.

Beyond its diagnostic capabilities, the application offers another compelling dimension of

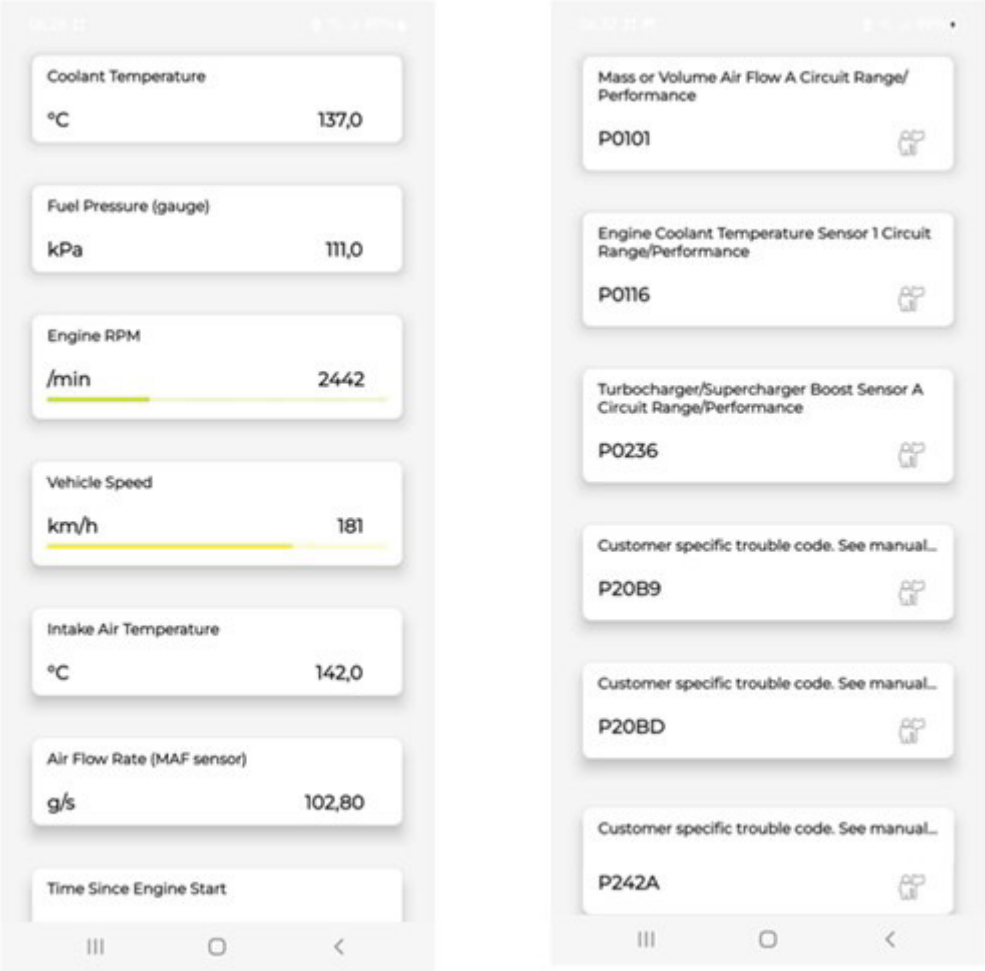

Figure 10 Data Onboard Diagnostic (OBD) (Left) and Fault Code (Right)

utility. It provides access to workshop services for vehicle repair and maintenance, forging a vital link between car owners and automotive service providers. This seamless connection streamlines finding reputable repair services, reducing downtime and inconvenience for vehicle owners when issues arise. It not only saves valuable time and resources but also fosters a more holistic approach to vehicle care.

The implications of this research are profound. The application's ability to detect and interpret car problems by reading error codes stored in the vehicle's system can revolutionize how car owners approach vehicle maintenance. This innovation promises to save users substantial time and money that will otherwise be spent on diagnosing and resolving car problems through conventional means. It embodies the essence of proactive vehicle maintenance, ensuring that users are well-informed about their cars' health and can take swift action when necessary.

Nevertheless, it is important to acknowledge the limitations of the research. One notable constraint is that the application's compatibility is currently restricted to Android devices due to the utilization of existing libraries. This limitation, while a practical consideration, may exclude a portion of potential users who prefer other operating systems. Therefore, future research and development efforts should focus on expanding the application's compatibility to cater to a broader spectrum of users and ensure the maximum reach and impact of this promising automotive tool.

Further research is suggested to improve the development of the User Interface (UI) of the application by adding a data export feature as a report of the vehicle condition and information and a data service record as a service of maintenance history. This improvement should involve the incorporation of a data export feature, which will enable users to generate detailed reports about their vehicle's condition and related information. Additionally, a data service record function should be integrated to maintain a comprehensive history of vehicle maintenance activities. These enhancements will provide a more user-friendly and informative experience for application users.

REFERENCES

Arena, F., Pau, G., & Severino, A. (2020). An overview on the current status and future perspectives of smart cars. *Infrastructures, 5*(7), 1–16.

Akhibi, S. D., & Akingbade, F. K. (2021). Development of an automobile on-board diagnostic reader. *International Journal of Electrical and Electronics Engineering Studies*, *7*(1), 28–34.

BinMasoud, A., & Cheng, Q. (2019). Design of an IoT-based vehicle state monitoring system using Raspberry Pi. In *2019 International Conference on Electrical Engineering Research & Practice (ICEERP)* (pp. 1–6). IEEE.

Dong, X., & Gaofei, Z. (2019). The signal-oriented test system model of automated test systems and its identification technology. Measurement and Control, 52(7-8), 869–878.

Gallardo, F. B. (2018). *Extraction and analysis of car driving data via OBD II* (Doctoral dissertation). Universidad Miguel Hernández de Elche.

Gerber, A., Craig, C., & Selvaraj, D. (2015). *Learn Android Studio: Build Android apps quickly and effectively*. Apress.

He, H., Shou, Y., & Wang, H. (2022). Fuel economy optimization of diesel engine for plug-in hybrid electric vehicle based on equivalent operating points. *Control Engineering Practice, 123*.

He, W., Zheng, X., Zhang, Y., & Han, Y. (2022). Study on determination of excessive emissions of heavy diesel trucks based on OBD data repaired. *Atmosphere, 13*(6), 1–16.

Jamal, R., & Wenzel, L. (1995). The applicability of the visual programming language LabVIEW to large real-world applications. In *Proceedings of Symposium on Visual Languages* (pp. 99–106). IEEE.

Ju, F., Zhuang, W., Wang, L., & Zhang, Z. (2020). Comparison of four-wheel-drive hybrid powertrain configurations. *Energy, 209*.

Ling, J., Li, Y., Li, J., & Yan, Y. (2020). Research on production vehicle evaluation method of China VI OBD for light-duty vehicles. In *IOP Conference Series: Materials Science and Engineering* (Vol. 774). IOP Publishing.

Macias-Bobadilla, G., Becerra-Ruiz, J. D., Estévez-Bén, A. A., & Rodríguez-Reséndiz, J. (2020). Fuzzy control-based system feed-back by OBD-II data acquisition for complementary injection of hydrogen into internal combustion engines. *International Journal of Hydrogen Energy, 45*(51), 26604–26612.

Malini, T., Sudha, R., Raj, P. A. C., & Stalin, B. (2020). The role of RTD and liquid sensors in electric arc furnace for melting of aluminium. *Materials Today: Proceedings, 33*, 4793–4796.

Martinelli, F., Mercaldo, F., Nardone, V., & Santone, A. (2021). driver identification through formal methods. *IEEE Transactions on Intelligent Transportation Systems, 23*(6), 5625–5637.

Niazi, M. A. K., Nayyar, A., Raza, A., Awan, A. U., Ali, M. H., Rashid, N., & Iqbal, J. (2013). Development of an On-Board Diagnostic (OBD) kit for troubleshooting of compliant vehicles. In *2013 IEEE 9th International Conference on Emerging Technologies (ICET)* (pp. 1–4). IEEE.

Nugroho, S. A., Ariyanto, E., & Rakhmatsyah, A. (2018). Utilization of On-Board Diagnostic II (OBD-II) on four wheel vehicles for car data recorder prototype. In *2018 6th International Conference on Information and Communication Technology (ICoICT)* (pp. 7–11). IEEE.

Nurcahya, S., Erfianto, B., & Setyorini, S. (2022). Forecasting fuel consumption based-on OBD II data. *Indonesia Journal on Computing (Indo-JC), 7*(2), 93–102.

Pranjoto, H., Agustine, L., & Mereditha, M. (2017). OBD-II-based vehicle management over GPRS wireless network for fleet monitoring and fleet maintenance

management. *Journal of Telecommunication, Electronic and Computer Engineering, 10*(2-3), 15–18.

Pressman, R., & Maxim, B. (2020). *Software engineering: A practitioner's approach* (9th ed.). McGraw Hill.

Riswal, N. S., Joko, S., & Hari, N. (2020). Multi diagnosis medical device engineering based on radioimmunoassay. In *Prosiding Seminar Nasional Inovasi dan Pendayagunaan Teknologi Nuklir 2020* (pp. 259–268)

Zeb, A., Khattak, K. S., Agha, A., Khan, Z. H., Sethi, M. A. J., & Khan, A. N. (2022). On-Board Diagnostic (OBD-II) based cyber physical system for road bottlenecks detection. *Journal of Engineering Science and Technology, 17*(2), 0906–0922.