JOURNAL OF DIGITAL SYSTEM DEVELOPMENT

https://e-journal.uum.edu.my/index.php/jdsd

How to cite this article:
Puteh, N., Choong, Q.Y., Husin, M.Z. & Ahmad, C.W. S.B.C. (2024). A Mobile Application for Personalised Car Information. *Journal of Digital System Developments*, *2*(2), 108-125. https://doi.org/10.32890/jdsd2024.2.2.8

A MOBILE APPLICATION FOR PERSONALISED CAR INFORMATION

[1]Nurnasran B Puteh, [2]Choong Qian Yu, [3]Mohd Zabidin B Husin & [4]Che Wan Shamsul Bahri B C.W.Ahmad

[1,2,3]School of Computing,
University Utara Malaysia, 06010 UUM Sintok, Kedah, Malaysia

[4]Faculty of Creative Multimedia & Computing,
Selangor Islamic University, 43000 Kajang, Selangor, Malaysia

[1]Corresponding author: nasran@uum.edu.my

Received: 30/07/2024 Revised: 04/09/2024 Accepted: 09/09/2024 Published: 20/10/2024

ABSTRACT

The high rate of car ownership in Malaysia may lead to difficulties for owners in remembering all the details of their vehicles. Therefore, an application called CarBuddy, which provides personalised information about users' cars, would help them manage and maintain their vehicles more efficiently. The project adopted a waterfall methodology during the application development to produce a prototype based on the requirements gathered during the analysis phase. Field-testing for the application involved 30 participants who installed the application and provided feedback through a questionnaire distributed via Google Forms after using the application. The respondents' feedback is based on four dimensions of the prototype: user-friendliness, usefulness, ease of learning, and satisfaction. The test's overall result indicates that most respondents are satisfied with this application. Usability tests revealed that the application significantly improved the user experience by providing reminders and detailed insights into vehicle performance. This application can benefit car owners in Malaysia by enhancing vehicle maintenance and the overall user experience.

Keywords: Personalised car information, Mobile application, Vehicle performance optimisation.

INTRODUCTION

According to traffic safety specialist Professor Dr Kulanthayan K.C. Mani of University Putra Malaysia, the country had 33.3 million verified vehicles in 2021 compared to the country's population size of 32.6 million people (Chan, 2022). Of those 33.3 million, as many as 47.3 per cent were cars, 46.6 per cent were motorbikes, and 4.7 per cent were commercial trucks. The remainder included public transportation, taxis, self-drive rentals, and other types of automobiles. This data suggests that many citizens own more than one vehicle, potentially leading to challenges managing vehicle information such as service dates and insurance renewals. Therefore, an application providing personalised information about their vehicles could help citizens optimise their vehicle performance.

Mobile app personalisation is the process of customising content, products, or services within a mobile application to meet individual users' specific needs and preferences based on their past behaviours and preferences (Mao et al., 2023). This strategy has been extensively researched in various domains, including hearing aid technology (Ni & Kehtamavaz, 2023) and food recommendation systems (Jalali et al., 2022). App personalisation is essential for creating an outstanding user experience, which is vital for an app's sustainability. By tailoring application performance to the preferences of specific individuals, groups, or communities, developers can enhance user engagement and ensure the app remains distinctive and frequently used.

In this project, the application was developed as a mobile application. A mobile application, or app, is a software program designed to run on portable devices such as tablets or smartphones (Hanna & Wigmore, 2023). Many computing languages and platforms are used to create mobile applications, which may be acquired and installed from applications like Google Play or the Apple App Store. The use of mobile applications has several benefits, including availability. Users can use mobile applications' features and products anywhere by downloading and installing them on their smartphones. Moreover, the mobile application also has the benefit of accessing offline. Many mobile applications are designed to operate offline, allowing users to access essential data and functionalities without an internet connection (Juillet, 2022). Additionally, mobile applications can utilise push notifications, enabling users to receive real-time updates and critical information even when they are not actively using the application.

The objectives of this project are to determine the requirements for a personalised car information mobile application and to design and develop the application accordingly. The intended users are individuals who own at least one car. Additionally, the project also aims to evaluate the usability of this application. The application is anticipated to significantly impact car maintenance practices by improving their overall satisfaction, guaranteeing protection, maximising servicing, encouraging community involvement, exhibiting technical innovation, and supporting environmental initiatives.

In summary, this personalised car information application aims to provide users with a comprehensive experience for overseeing, managing, and tracking their vehicle details. Besides, the intuitive user interface enhances user experience, allowing users to navigate and utilise the application's functionalities efficiently.

RELATED WORK

According to Ravimalar (2022), Malaysia's three most used car maintenance applications are Serv, MyCar Management and Drivvo. Firstly, Serv is a car maintenance application developed by Serv Technology

Sdn Bhd (Serv. My, 2017). Users of Serv may utilise the application to monitor the condition of their cars and set up recurring maintenance schedule notifications. Setting up schedules with the closest workshop and emergency services is one of the app's main features. However, the workshops that are collaborating with Serv are not available in every area of the user. Certain areas of users might not experience this feature provided by Serv as the collaborating workshops are too far away or it is not available from the user's area. With this application, users may also renew their road tax and insurance. Since battery problems are the most frequent reason for breakdowns, Serv also offers a function that lets customers sell their current car batteries and seek breakdown help for a new battery.

The second most used car maintenance application is MyCar (MyCar, n.d.). This software is intended for users who want to keep track of the mileage, gas consumption, and the number of fill-ups for their car. Users can append invoices and other records to automobile events. MyCar is also available as a web-based application, in addition to being available for iOS and Android users. Nevertheless, this application requires the user to have an in-app payment to unlock some premium functions. Users of the premium edition may log private or corporate travels, publish reports in a PDF file with graphics and data, and set up maintenance notifications. The app can obtain information from any device that has the app installed as a result of its cloud sync function.

The third application is "Drivvo" (Drivvo et al.). Unlike other automobile management apps, Drivvo is versatile and can manage cars, motorcycles, buses, and trucks. It helps users track fuel consumption, refuelling, maintenance, and other expenses such as fines. This data lets users understand their monthly expenditures, average usage, cost per kilometre, fuel efficiency, and financing payments. Drivvo allows users to collect fuel efficiency data from various fuel stations, manage multiple vehicles, and calculate fuel usage. The app offers up to 60 language options, automatically matching the language to the user's device settings. However, users must subscribe to the premium version to access additional features. The basic version of Drivvo permits users to record only two vehicles and one income entry per month and includes advertisements within the app. In contrast, the premium version encompasses all basic version features but without advertisements and allows users to register up to ten vehicles. Drivvo manages vehicles for corporate use with various premium plans that fit the company's operation.

Advanced technologies like the Internet of Things (IoT) and machine learning are increasingly being utilised in vehicle maintenance systems. Srikanth, Kumar, and Sharma (2021) describe developing a vehicle system that integrates IoT, cloud computing, and machine learning to automate the monitoring and tracking of vehicle conditions. The system collects data on various vehicle parameters, including distance travelled, lubricant levels, tyre conditions, and smoke emissions, using IR, MQ-6 gas, and ultrasonic sensors. This data is subsequently stored in the cloud, where machine learning algorithms analyse it to predict the vehicle's next service date. The authors highlight the system's capacity to reduce human intervention in vehicle maintenance by providing timely service reminders via an Android application, enhancing efficiency and user convenience. This approach significantly addresses the limitations of traditional vehicle service monitoring by offering a more proactive and data-driven solution. However, the system's effectiveness depends on the reliability and proper calibration of the sensors employed; inaccuracies in sensor data could result in incorrect maintenance predictions, potentially leading to unnecessary services or the oversight of critical maintenance, thereby compromising vehicle safety.

Similarly, Al-Jaafari et al. (2020) explore the development of an intelligent mobile system to improve vehicle safety and maintenance. This system utilises IoT and cloud computing to deliver real-time monitoring and maintenance alerts. By integrating sensors with an embedded system, the vehicle can

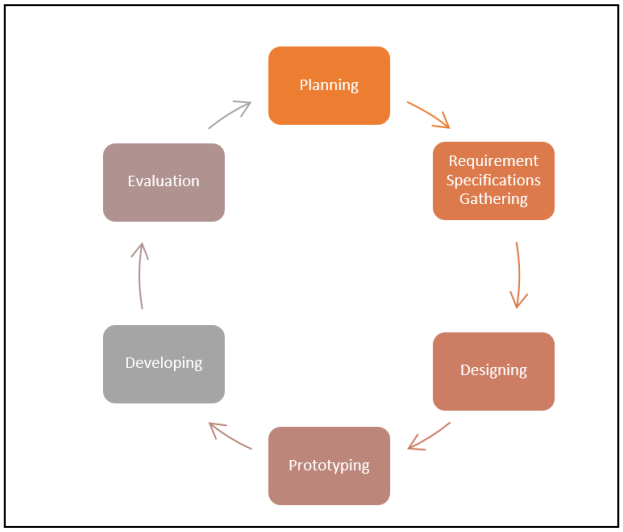
directly communicate with a mobile application, alerting the owner to any maintenance requirements or safety concerns. The system supports proactive maintenance by issuing timely alerts and enabling users to request services directly through the app, including locating nearby workshops using GPS. However, the system's reliance on continuous internet connectivity for real-time data transmission poses a significant challenge. In regions with poor or inconsistent internet access, the system may fail to function optimally, resulting in delayed maintenance alerts or the potential loss of critical data.

METHODOLOGY

The development methodology of the application was adopted from the Agile methodology (Alsaqqa et al., 2020). This methodology involves six main stages, as illustrated in Figure 1.

Figure 1

Project Methodology with Six Main Phases



The first phase is the planning phase, during which research on personalised car systems was carried out. This involved an examination of related prior work, problem statement formulation, establishing research objectives, project scope and significance. Next is the requirement specification gathering phase, during which the functional and non-functional requirements are gathered. Additionally, the platforms to be used, the timeline, limitations and constraints, and other issues related to the application were discussed. Next, the user interface and workflow were developed during the design phase. The user interface and workflow design were based on the requirements and specifications outlined during the previous phases. After completing the design phase, the application moved to the prototyping phase, where a low-fidelity prototype was developed to experiment with different design solutions and functionalities without the risk of wasting time or resources. After several rounds of refinement, the prototype was finalised, and the project moved on to the development phase. During this phase, the application was coded to create a fully functional product by the established requirements and specifications. Detailed information regarding the requirements specification phase up to the development phase is provided in the design and development section. Upon the completion of the application implementation, it was necessary to subject it to testing and evaluation to confirm that it met user requirements, thus marking the final stage of the project: the evaluation phase. This phase was crucial to confirm that the application was error-free and provided a

satisfactory user experience. If any errors were found or improvements were needed, the application had to return to the implementation phase for code modifications. After completing the evaluation stage, the application was deemed ready for launch. The analysis and results section provides a detailed elaboration of the evaluation process and its results.

DESIGN AND DEVELOPMENT

This section specifies the design and development of CarBuddy, outlining the functional and non-functional requirements for the application that would meet the end user's expectations and detailing the prototype development. Table 1 lists the functional requirements, while Table 2 illustrates the non-functional requirements that enhance the user experience with the application. The functional requirements include new account registration, logging into the application, managing user accounts, registering new car information, managing registered car maintenance information and inserting fuel records to track fuel consumption. These functionalities help the user to centralise their care management processes. The non-functional requirements emphasise the security, usability and reliability issues that enhance the performance of the application.

Table 1

Functional Requirements for Carbuddy

No	Requirement ID	Requirements Description	Priority
	PCI_01	Register account	
1.	PCI_01_01	Users can click the register button to create an account on the login page.	M
2.	PCI_01_02	User must enter their username, email and phone number to register their account.	M
3.	PCI_01_03	User can click the sign-up button to create and save their personal information.	M
4.	PCI_01_04	An error message will be displayed if the input is invalid.	D
	PCI_02	Log in account	
5.	PCI_02_01	The user must enter the username and password on the register page.	M
6.	PCI_02_02	The user clicks on the login button to enter the application's main page.	M
7.	PCI_02_03	An error message will be displayed if the user enters the wrong username or password.	D
8.	PCI_02_04	The user clicks forget password button to reset the password	M
9.	PCI_02_05	The user clicks the save button to save the reset password.	M
	PCI_03	Manage account	
10.	PCI_03_01	User can update their personal information.	M
11.	PCI_03_02	User can scan their driver’s license.	M
12.	PCI_03_03	User can update their driver's license.	M
13.	PCI_03_04	After updating, the save button to save the edited details.	M
14.	PCI_03_05	Users can log out by clicking the logout button.	M

	PCI_04	Register car	
15.	PCI_04_01	User can add their car information, including insurance policy and road tax.	M
16.	PCI_04_02	The save button allows the system to store the information.	M
17.	PCI_04_03	The system will display the car information, insurance details, and road tax information on the main page.	M
18.	PCI_04_04	User can update their car information and insurance policy.	M
19.	PCI_04_05	User can delete their car information.	M
	PCI_05	Manage maintenance information	
20.	PCI_05_01	Users can add the details of maintenance.	M
21.	PCI_05_02	Users can modify the maintenance details	M
22.	PCI_05_03	Users can delete the maintenance details	M
23.	PCI_05_04	The system will notify the user when the maintenance date is near.	M
24.	PCI_05_05	Users can review the history of maintenance	D
	PCI_06	Add fuel consumption information	
25.	PCI_06_01	Users can enter the cost of fuel and the date they refuel.	M
26.	PCI_06_02	The system can analyse data on the vehicle's fuel consumption.	M
27.	PCI_06_03	The system can generate graphs or pie charts according to the analysis data.	M
28.	PCI_06_04	The system can display the report in the application	M

Table 2

Non-Functional Requirements for Carbuddy

Num.	Requirement ID	Requirements Description	Priority
	PCI_07	Security issues	
1.	PCI_07_01	The username and password must be the same as in the system.	M
2.	PCI_07_02	Automatically log out and show “Login Session Expired”.	D
	PCI_08	Usability issues	
3.	PCI_08_01	The system should be able to add information within 1 minute.	D
4.	PCI_08_02	The user should be able to log in/log out within 20 seconds.	D
5.	PCI_08_03	The interface design should be user-friendly.	D
	PCI_09	Reliability issues	
6.	PCI_09_01	For a single user, the system should crash no more than once per 2 hours.	D
7.	PCI_09_02	If the system crashes, it should behave perfectly normal when reloaded again.	D

The requirements in Table 1 were visualised and modelled into the Unified Modelling Language (Bell, 2023). The use case diagram and the class diagram are the two behavioural diagrams that best describe the requirement structure of CarBuddy. The use case diagram helps understand and communicate the functional requirements and interactions between the system and the user, as shown in Figure 2. The class diagram provides a detailed illustration of the system's static structure, illustrated in Figure 3. The actor in the use case diagram is CarBuddy's end user. The primary use cases the user can perform are registering an account, logging in, managing a profile, registering a car, managing maintenance information, and adding fuel consumption information. The user can carry out more detailed use cases from the primary ones. The user can view and update their profile by managing their profile. Next, the user can enter their car information, register their insurance policy, add additional car details, and update or delete existing car information. For the maintenance use case, the user can perform add, update, and delete maintenance, and the system can generate the history of the past maintenance. Lastly, the system can generate consumption reports from the fuel record.

Figure 2

Use Case Diagram for CarBuddy

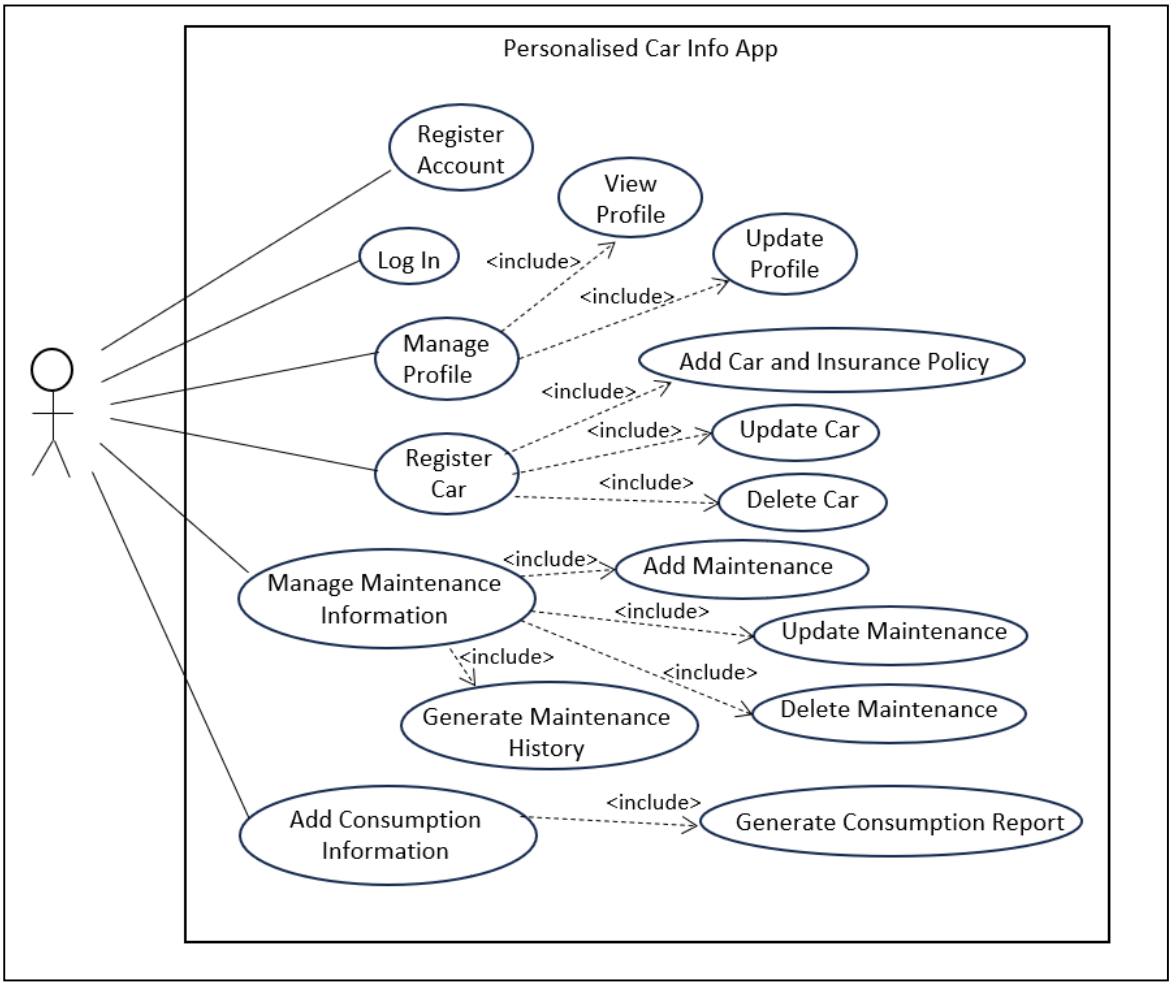
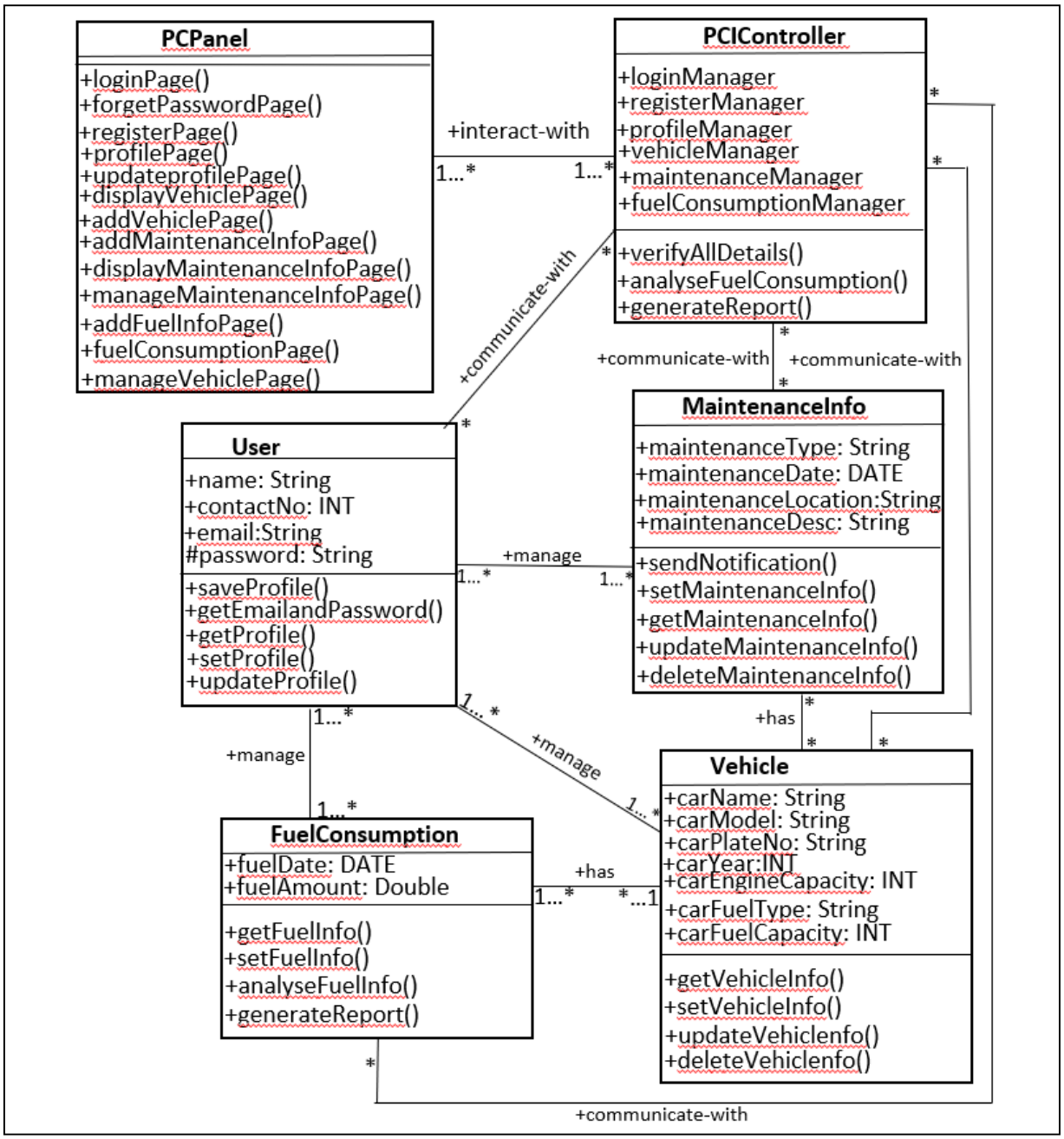


Figure 3

Class Diagram for CarBuddy



The prototype of CarBuddy was designed and developed according to the requirements gathered. The tools involved include Visual Studio Code as the leading integrated development environment (IDE) tool, Android Studio as the IDE that provides an Android emulator, and cPanel, a web hosting control panel software. The programming framework and languages involved are Flutter (Gonzalez, 2024), which uses the Dart language as the front-end service, the PHP scripting language (Toal, 2023) as the back-end service as well as MySQL and phpMyAdmin (Hughes, 2022) for managing the database. The figures below are some of CarBuddy's interfaces that adopted the functional requirements mentioned in the above section. Figures 4 to 6 show the splash screen, registration page, and login page of CarBuddy.

Figure 4

Splash Screen

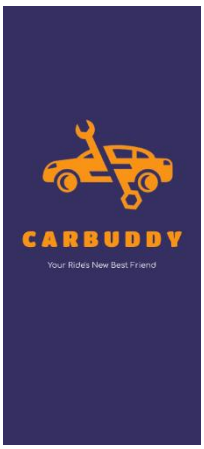


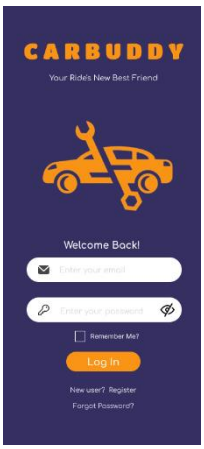
Figure 5

Register Page



Figure 6

Login Page



The user needs to add their car information to explore more functionalities. Figure 7 shows the interface where the user can add the car information, and Figure 8 shows the Home Page after the car information is added. Figure 9 shows the interface where the user can edit or delete the car information.

Figure 7

Add Car Details Page

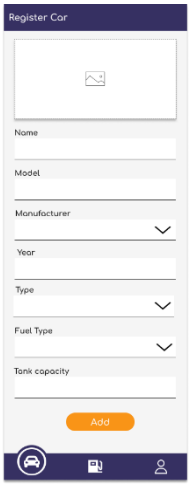


Figure 8

Home Page

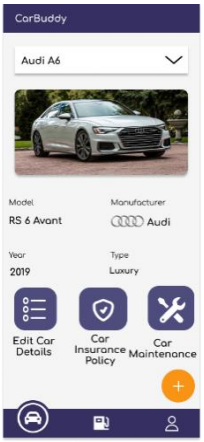
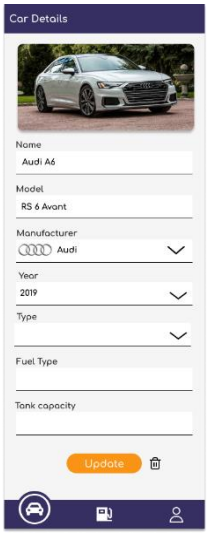


Figure 9

Edit Car Details Page



Users can add their car insurance policy details, allowing them to refer to the details while using the application. Reminders will be sent when the expiration date is near. Figure 10 illustrates the Add Insurance Policy page, where users can upload their documents and enter all the details on the Insurance Policy page, as shown in Figure 11. The user can update the insurance policy details after the renewal, as displayed in Figure 12.

Figure 10

Add Insurance Policy Page

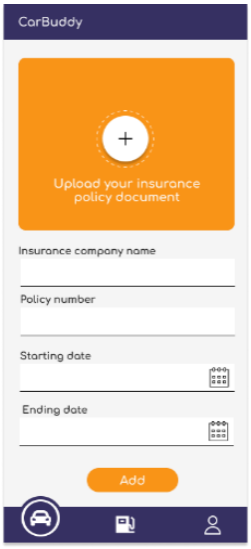


Figure 11

Insurance Policy Page

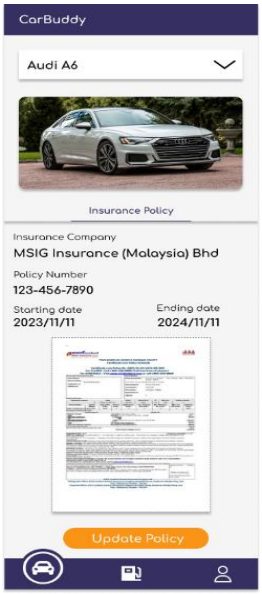
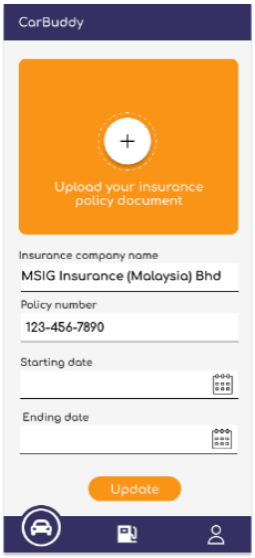


Figure 12

Update the Insurance Policy Page



Managing maintenance information is one of CarBuddy's functional requirements. Maintenance information can be added as in Figure 13. The record is displayed on the Manage Maintenance page, as in Figure 14. Users can access the Edit Maintenance Information page, as in Figure 15, to modify a record.

Figure 13

Add Maintenance Information Page

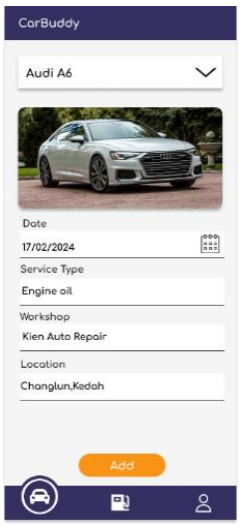


Figure 14

Manage Maintenance Page

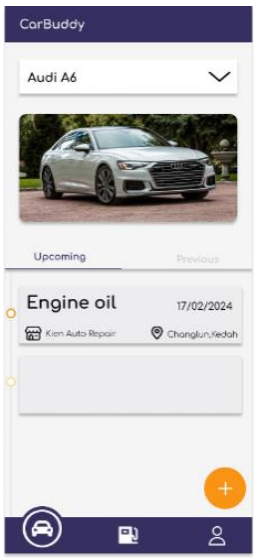


Figure 15

Edit Maintenance Information Page

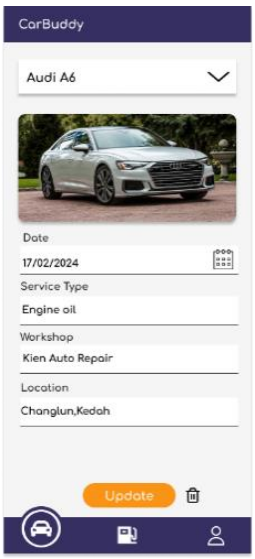


Figure 16-19 depicts the fuelling record management interfaces. The user can enter refuel details on the Add Refuel Information page (Figure 16), which will then appear in the calendar (Figure 17). The application analyses these details and visualises fuel consumption with a bar graph (Figure 18).

Figure 16

Add Refuel Information Page

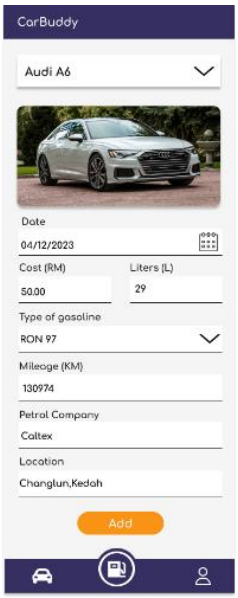


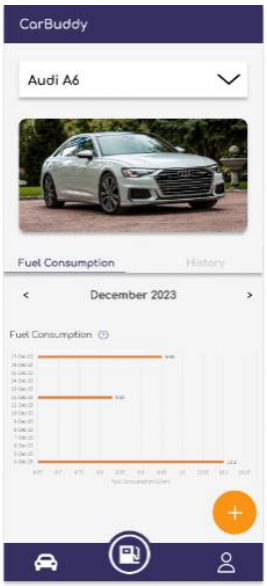
Figure 17

Refuel History Page



Figure 18

Fuel Consumption Analysis Page



The user can manage their profile by adding their driver's license and editing their personal information on the Profile page by clicking the floating button, as in Figure 19-21.

Figure 19

Profile Page



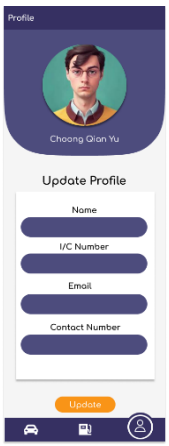
Figure 20

Profile Page



Figure 21

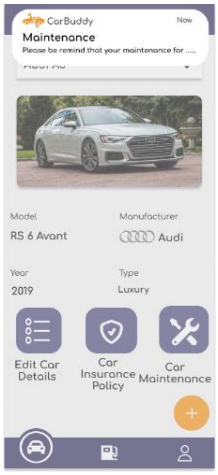
Update Profile Page



A push notification will be sent to the user when the insurance policy expiration date and the scheduled maintenance date are near, as shown in Figure 22.

Figure 22

Push Notification



ANALYSIS AND RESULTS

CarBuddy conducted a usability test with 30 end users to gather feedback and suggestions. The respondents were selected based on their experience in car ownership and maintenance. It was required that they had owned at least one car for a minimum period of one year. The purpose of this feedback was to assess the users' satisfaction and identify areas for improvement. Each participant installed and used the application via Google Forms before completing a questionnaire. The questionnaire, utilised to assess the application's usability, is adapted from the USE Questionnaire (Lund, 2001). This survey-based instrument evaluates user experience by concentrating on three critical dimensions: 'Usefulness' (the extent to which the application fulfils users' needs), 'Satisfaction' (users' overall contentment with the system), and 'Ease of Use' (the degree of simplicity with which users interact with the system). The questionnaire was divided into five sections: Demographics, User-Friendliness, Usefulness, Ease of Learning, and Satisfaction. Users provided their feedback and suggestions at the end of the survey. Each question is rated on a five-point scale: Strongly Agree, Agree, Neutral, Disagree, and Strongly Disagree.

The distribution demographic of the respondents

Among the 30 respondents, 17 were female (56.7%) and 13 were male (43.3%). Most respondents were aged 20–39 (17 individuals, 56.7%), followed by those aged 40–59 (7 individuals, 23.3%). The least represented age group was 60 and above, with only two respondents (6.7%), while four (13.3%) were under 20. Most respondents had a monthly income between RM1,000 and RM2,999 (20 out of 30, 66.7%). Fewer respondents had monthly incomes between RM3,000 and RM4,999 (3 individuals, 10%) or less than RM999 (7 individuals, 23.3%). The survey also revealed that 24 respondents (80%) had no experience with car-related applications.

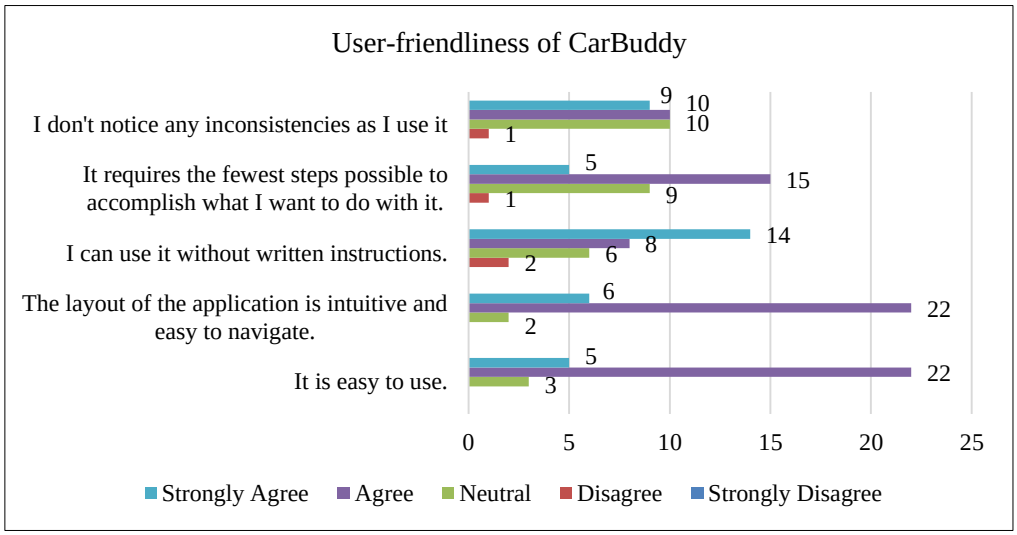
User-friendliness of CarBuddy

Figure 23 shows that CarBuddy is generally viewed as user-friendly. Most respondents (90%) found the app easy to use, and an even higher percentage (93.3%) considered the layout intuitive and easy to navigate.

Most users (73.4%) felt they could use the app without written instructions, though some disagreed or were neutral. Additionally, 66.7% agreed that the app requires minimal steps to complete tasks, with 30% neutral. Consistency was noted by 63.3% of users, while a third remained neutral. Overall, CarBuddy excels in ease of use and intuitive design but could benefit from more explicit instructions and addressing inconsistencies. Developers should use this feedback to enhance the app's usability.

Figure 23

User-friendliness of CarBuddy

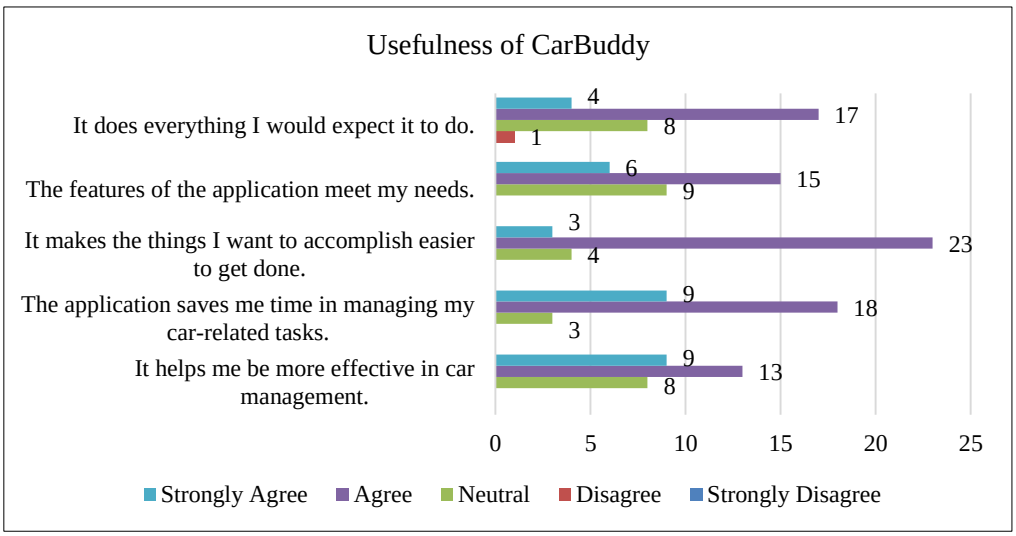


Usefulness of CarBuddy

Figure 24 indicates that users generally find the CarBuddy helpful application. Most respondents (73.3%) believe it improves their car management efficiency, and 90% agree that it saves time. Additionally, 86.7% feel that CarBuddy simplifies their tasks. About 70% of users believe the app meets their needs and expectations, although there were notable neutral responses. These findings suggest CarBuddy effectively delivers valuable functionality and streamlines tasks but also highlights areas for improvement, such as demonstrating the app's full potential and addressing gaps between user expectations and features. Overall, CarBuddy offers significant benefits with opportunities for further refinement to boost user satisfaction.

Figure 24

Usefulness of CarBuddy

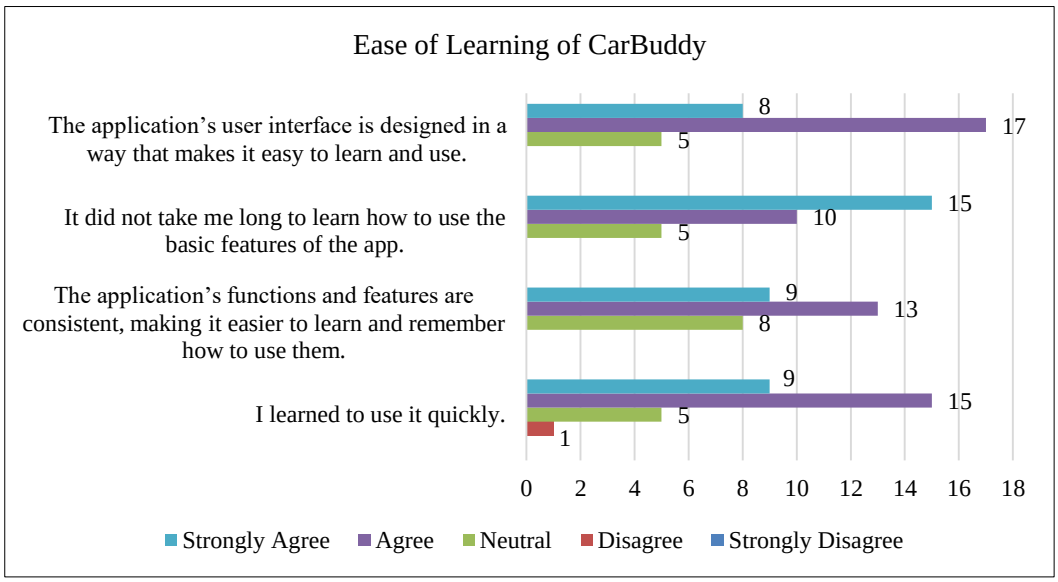


Ease of Learning of CarBuddy

Figure 25 shows that users generally find CarBuddy easy to learn and use. 80% of respondents quickly learned to use the app, and a similar percentage found its basic features straightforward. The app's consistency aids in memorability, with 73.3% agreeing that it helps them remember how to use CarBuddy. Additionally, 83.3% felt the user interface design supported easy learning and use. Despite the positive feedback, there were consistent neutral responses (16.7% to 26.7%), suggesting that some users, particularly those less familiar with technology or older individuals, may encounter difficulties. Overall, CarBuddy’s design facilitates quick learning for most users but could benefit from improvements to assist those struggling with new technology.

Figure 25

Ease of Learning of CarBuddy

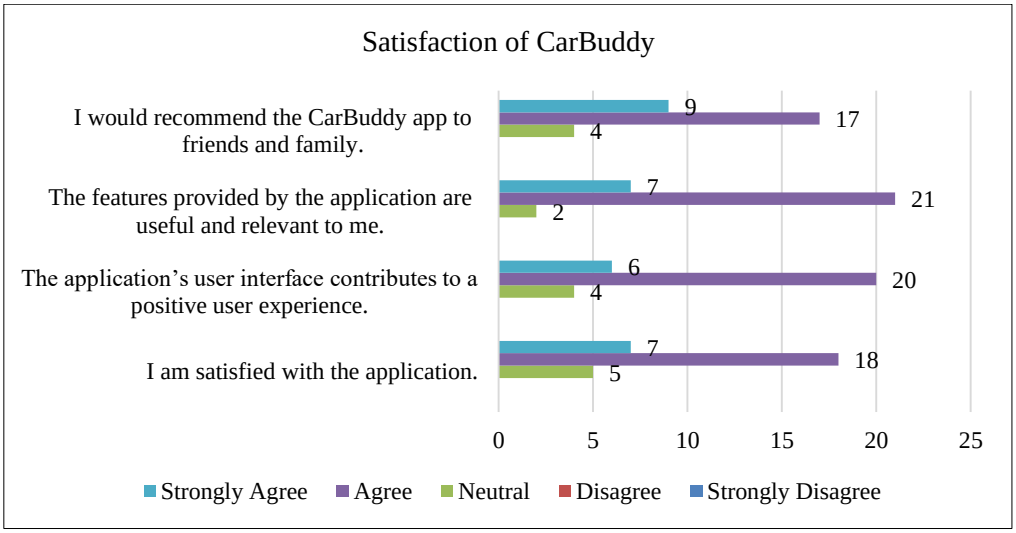


Satisfaction of CarBuddy

Figure 26 shows high user satisfaction with the CarBuddy application. A significant majority (83.3%) are overall satisfied, with no users expressing dissatisfaction. Additionally, 86.7% find the user interface enhances their experience, and 93.3% consider the features useful and relevant. Furthermore, 86.7% would recommend CarBuddy to others, reflecting strong confidence in its value. Although minor neutral responses (6.7% to 16.7%) indicate some areas for improvement, the absence of negative feedback highlights the app's success in meeting user needs. Overall, CarBuddy is well-received, with high satisfaction regarding its interface, features, and functionality, though there is still room for enhancement.

Figure 26

Satisfaction of CarBuddy



Feedback and Suggestions

The respondents identified both the strengths and weaknesses of CarBuddy and offered suggestions for enhancing the application. Table 3 highlights several favourable aspects of CarBuddy, demonstrating its values in car management.

Table 3

The Favourable Aspects Highlighted by Respondents

Auto Login	User-friendly Interface. I love the clean and intuitive design of the CarBuddy app. It's easy to navigate.
The fuel consumption part which makes me easy to manage my budget	Maintenance reminder
Add license function	Can you check the history
Graph for interpreting fuel consumption	Notification reminder
Fuel record tracker	Maintenance and insurance expired reminder

Table 4 outlines several issues with the CarBuddy application. Users reported problems such as hidden passwords during login, slow loading times (notably for fuel data), and the need to manually enable notifications. Other concerns included limited language options, no location input suggestions, inability to change passwords in-app, and confusion over specific fields. Additional drawbacks were the lack of dark mode, automatic fuel consumption display, and alignment issues affecting readability and design. These insights highlight areas for improvement to enhance functionality and usability.

Table 4

The Unfavourable Aspects Highlighted by Respondents

Password invisible when login.	Slow Loading Times. Sometimes, the app takes too long to load, especially when accessing fuel consumption.
Need to manually allow the notification	Only one language option is available
Needing us to type out the locations instead of giving options when we type out the first few alphabets	Cannot change password
Confusion in some required information, not sure what to enter	Unavailable in dark mode
The number of fuel consumption is not showing automatically	Alignment

Table 5 summarises user recommendations for enhancing the CarBuddy application. Suggested improvements include adding interface customisation options, enabling auto-allowed notifications, and incorporating automatic fuel calculation based on cost. Users also recommend expanding language options, making passwords visible during login, and adding a "forgot password" feature. Additionally, providing examples for information fields and improving alignment for readability are suggested. These recommendations target current shortcomings to enhance user experience and accessibility.

Table 5

The Recommendations Highlighted by Respondents

Customisation Options: More options for personalising the app interface, such as theme colour and profile picture, are added.	Auto allowed notification
Auto calculate the fuel litter based on the cost	Provide more language options
Make the password visible when login	Provide an example of the required information
Enhance the alignment	Make the password visible when login
Add the forget password feature	

CONCLUSION

CarBuddy is a mobile application designed for car owners in Malaysia. It features car registration, maintenance tracking, fuel consumption analysis, and insurance management. The app aims to enhance user satisfaction and maintenance practices. Evaluations of CarBuddy reveal high user satisfaction, with most users finding the app user-friendly, helpful, and easy to learn. While the app is well-received for its interface and features, feedback suggests areas for improvement, such as enhancing customisation options, adding auto-calculation features, and addressing alignment and readability issues. Future enhancements for CarBuddy include integrating with vehicle manufacturers' systems and smart devices for seamless data syncing and user access. Additional improvements include advanced analytics for fuel efficiency and maintenance trends, expanding language options, and incorporating security features like multi-factor authentication.

REFERENCES

Al-Jaafari, N. G., Ahmed, I., Bhatt, A. Z., & Khan, M. S. (2020). Intelligent vehicular maintenance systems using IoT, mobile applications, and internet-enabled vehicles could be used. *Journal of Student Research*.

Alsaqqa, S., Sawalha, S., & Abdel-Nabi, H. (2020). Agile Software Development: Methodologies and Trends. *International Journal of Interactive Mobile Technologies (iJIM)*, *14*(11), pp. 246–270.

Bell, D. (2023). *An introduction to the Unified Modelling Language*. IBM Developer. https://developer.ibm.com/articles/an-introduction-to-uml/

Chan, D. (2022, June 9). Vehicles outnumber people in Malaysia. *New Straits Times*. https://www.nst.com.my/news/nation/2022/06/803654/vehicles-outnumber-people-malaysia

Drivvo App. (n.d.). *Driv*er. https://www.drivvo.com/en

Gonzalez, C. (2024). *An introduction to Flutter's world*. Custom Software Development & IT Staffing | FullStack. https://www.fullstack.com/labs/resources/blog/an-introduction-to-flutters-world

Hanna, K. T., & Wigmore, I. (2023, February 13). *What is a mobile app (mobile application)? – TechTarget definition*. TechTarget. https://www.techtarget.com/whatis/definition/mobile-app

Hughes, J. (2022, September 16). *A quick guide to phpMyAdmin (And how you can use it)*. Elegant Themes Blog. https://www.elegantthemes.com/blog/resources/a-quick-guide-to-phpmyadmin-and-how-you-can-use-it

Jalali, A., Arjun Manoj, K. P., Amulya, N., Siddiqua, A., Singh, A. K., & Hari Krishna, S. M. (2022, April 07-09). Mobile Application for Personalized Food Recommendation. *2022 IEEE 7th International Conference for Convergence in Technology (I2CT)*, 1–6.

Juillet, R. (2022, November 8). *Why you should develop a mobile application that works offline?* Bocasay. https://www.bocasay.com/why-develop-mobile-application-offline/

Lund, A. (2001). Measuring Usability with the USE Questionnaire. *Usability Interface*, *8*(2), 3–6. https://www.researchgate.net/publication/230786746_Measuring_Usability_with_the_USE_Questionnaire

Mao, S., Dewan, S., & Ho, Y.-J. (2023). Personalised Ranking at a Mobile App Distribution Platform. *Information Systems Research*, *34*(3), 811–827.

MyCar. (n.d.). *Home | My Car - Vehicle Management, Vehicle Maintenance Log & Mileage Tracking. MyCar App*. https://www.mycar-app.com

Ni, A., & Kehtamavaz, N. (2023). A Real-Time Smartphone App for Field Personalization of Hearing Enhancement by Adaptive Dynamic Range Optimization. *2023 5th International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, 1–6.

Ravimalar, R. (2022, April 25). *Listicle: Maintaining your vehicle digitally*. The Edge Malaysia. https://theedgemalaysia.com/article/listicle-maintaining-your-vehicle-digitally

Serv. my - App that sends mechanics directly to you. (2017, November 5). *CarKaki.My*. https://www.carkaki.my/2017/11/05/serv-app-sends-mechanics-directly/

Srikanth, M. S., Kumar, T. G. K., & Sharma, V. (2021). Automatic vehicle service monitoring and tracking system using IoT and machine learning. In A. P. Pandian, X. Fernando, & S. M. S. Islam (Eds.), *Computer networks, big data and IoT (Vol. 66)*, pp. 953–967. Springer Singapore.

Toal, R. (2023, October 23). *What is PHP? Uses & introduction*. *Code Institute Global*. https://codeinstitute.net/global/blog/what-is-php-programming/