



SIFFAT AHMED & MUHAMMAD AREEB ALI
01-134181-059 & 01-134181-082

CarRolla: Self Automotive Diagnostic

Bachelor of Science in Computer Science

Supervisor: Syed Hassan Tanvir

Department of Computer Science
Bahria University, Islamabad

January 25, 2022

Certificate

We accept the work contained in the report titled "**CarRolla: Self Automotive Diagnostic**", written by Siffat Ahmed & Muhammad Areeb Ali as a confirmation to the required standard for the partial fulfilment of the degree of Bachelor of Science in Computer Science.

Approved by . . . :

Supervisor: Syed Hassan Tanvir

Internal Examiner: Dr. Moazam Ali

External Examiner: Dr. Salabat Khan

Project Coordinator: Dr. Moazam Ali

Head of the Department: Dr. Arif Ur Rahman

January 2022

Acknowledgments

We owe all of our educational efforts and sacrifices to our loving parents and respected instructors; without them, we would be nothing. We are fortunate to have received such a learning environment in the shape of Bahria University, which has assisted us in polishing and building confidence, as well as developing into excellent human beings. Bahria University provided us with a variety of opportunities to learn about and use new technology. We would like to express our gratitude to our supervisor, Sir Syed Hassan Tanvir, for his assistance of utmost importance in completing our project successfully. Our supervisor contributed a great deal of encouragement, technical help, and other resources to ensure that our project, including the documentation and execution, was a success.

SIFFAT AHMED & MUHAMMAD AREEB ALI

ISLAMABAD, PAKISTAN

JANUARY, 2022

“As the Internet of things advances, the very notion of a clear dividing line between reality and virtual reality becomes blurred, sometimes in creative ways.”

Geoff Mulgan

Abstract

Car maintenance and diagnostics have become an integral part of the automotive industry in urban societies. Although the industry is quite old, the challenges faced in this flourishing industry require smart solutions on how to serve both the producer and the consumer. This can be done by bridging the gap between the service provider and the customer and by adding transparency to the equation. Also, the fraud in buying and selling cars has been on the rise and under the radar at the same time.

This IoT (Internet Of Things) based project presents a Three-Tier application that takes multiple services and combines them into one cross-platform application. It provides smart logs of car maintenance and reminds the user regarding the maintenance requirements of the car beforehand to ensure on-time maintenance of the vehicle which results in smoothness in the running of the car and increases car longevity at the same time. By connecting the application to the car using an OBD-II scanner, the car's problems can be pointed out before taking it to a workshop to get it fixed, resulting in the ease of diagnosis and increase in awareness of the user regarding the problem in their car. At the time of buying and selling of that car, the buyers can use our cloud-based verification system to ensure if the seller is selling the car with accurate information, hence, preventing odometer forgery fraud.

Contents

Acknowledgments	i
Abstract	iii
Abbreviations	xi
1 Introduction	1
1.1 Overview	1
1.2 Problem Description	2
1.3 Existing Solutions	4
1.4 Problem Statement	5
1.5 Objectives	5
1.6 Scope	6
2 Literature Review	7
2.1 Evolution of Car Engines	7
2.2 Conducted Survey	9
2.2.1 Questionnaire	10
2.2.2 Responses of the Questionnaire	15
2.3 Comparison of OBD-II Scanner Tools	20
2.4 Comparison of Existing Applications	22
2.5 Getting User Location to Show Workshops	25
2.5.1 Longitudes and Latitudes	26
2.5.2 How Longitudes and Latitudes are measured?	27
2.6 Research on Car Maintenance: Do All Cars Have Different Maintenance Schedules?	27
3 Requirement Specifications	32
3.1 Proposed System	32
3.2 Proposed Solution	32
3.2.1 The Application Will Get the Following Information	32
3.2.2 Using the Extracted Information	33
3.2.3 Driver State (in rest or driving)	33
3.2.4 ODOmeter Rollback Fraud Prevention	34
3.2.5 Workshop and Product Recommendation	34
3.2.6 Payment Integration	34
3.3 External Interface Requirements	35

3.4	System Requirements	52
3.5	Software Development Life Cycle (SDLC) Selection	53
3.5.1	Why Iterative Incremental Model?	53
3.6	Use Cases	54
4	Design	63
4.1	System Architecture	63
4.2	High-Level Design	66
4.3	Sequence Diagram for Registered User	70
4.4	Activity Diagram	73
4.5	Design and Implementation Constraints	74
5	System Implementation	75
5.1	Working	75
5.2	Tools and Technologies	76
5.2.1	Tools	76
5.2.2	Packages Used	77
5.2.3	API used	78
5.3	Google Firebase	79
5.3.1	Firebase Authentication	79
5.3.2	Firebase Database	79
5.3.3	Firebase Cloud Backup	79
6	System Testing and Evaluation	80
6.1	Testing Techniques	80
6.1.1	Functional Testing	80
6.1.2	Graphical User Interface Testing	82
6.1.3	Compatibility Testing	82
6.2	Test Cases	82
6.2.1	Register User	82
6.2.2	Sign In User	83
6.2.3	Guest User	83
6.2.4	Get Car Data	83
6.2.5	View Car Data	84
6.2.6	View workshops	84
6.2.7	The User can View Products	85
6.2.8	The User can Buy Products	85
6.2.9	Database is Integrated	85
6.2.10	Cloud Backup is integrated	86
6.2.11	Driver State Testing is Working	86
6.2.12	Application Installation	87
6.2.13	Application Launch	87
7	Conclusions	88
7.1	Conclusions	88
7.2	Future Enhancements	88

8	User Manual	90
8.1	Installation	90
8.2	Registering Yourself	91
8.3	Signing In	93
8.4	Navigating Through The Tabs	95
8.4.1	Home	95
8.4.2	Car Repair	97
8.4.3	Maps	102
8.4.4	Notifications	104
8.4.5	Settings	105

List of Figures

1.1	Pie chart of people who got their ODOmeter reading changed	2
1.2	If decreasing the Odometer reading offers better value, will the seller decrease it?	3
1.3	If yes, what percentage of actual reading will the seller reduce?	3
2.1	Data Trouble Code format [1]	7
2.2	Data Trouble Code format. [2]	8
2.3	DLC port in a Mitsubishi Car	9
2.4	Customer ODOmeter forgery survey	10
2.5	Education Level participants who were the part of the questionnaire	11
2.6	Region of the participants who were the part of the questionnaire	11
2.7	Brand of car owned by the participants who were the part of the questionnaire	12
2.8	Model year of the car owned by participants who were the part of the questionnaire	12
2.9	Information about the car owned by the Participants	12
2.10	Information about the car owned by the Participants	13
2.11	Intent of Participants about selling their car	13
2.12	Preference of Participants about getting there car serviced before selling	13
2.13	Did the Participants ever changed their vehicle's ODOmeter reading?	14
2.14	Asking the participants if they will forge the ODOmeter reading to get better value?	14
2.15	Percentage of value reduction participants will change	14
2.16	Results of Age group of participants	15
2.17	Result of current level of Education of the Participants.	15
2.18	Results of users in different regions of Pakistan	16
2.19	Result of Car Brands Most of the Users own (Part 1)	16
2.20	Result of Car Brands Most of the Users own (Part 2)	17
2.21	Result of Paticipant's intent to sell the car	17
2.22	Result of participant's accepting the act of Odometer forgery	18
2.23	Results about people who have changed Odometer readings	18
2.24	Results about people how will change value for better price	19
2.25	Result of percentage of actual reading that will be reduced	19
2.26	Ancel OBD-II Scanner (Front)	21
2.27	Ancel OBD-II Scanner (Back)	21
2.28	Torque Pro Mobile Application logo.	22
2.29	OBD Auto Doctor Application logo.	22
2.30	EOBD Facile Application logo.	23

2.31	Carly Application logo.	23
2.32	CarRolla: Smart Car Diagnostic Application.	24
2.33	Have you ever changed the reading of the Odometer? (by yourself or through a mechanic)	25
2.34	Graphical representation of Longitudes or Meridians	26
2.35	Graphical representation of Latitudes	27
2.36	Toyota's maintenance schedule chart.	28
2.37	Toyota's maintenance schedule chart.	28
2.38	Honda maintenance schedule	29
2.39	Suzuki's maintenance schedule chart.[3]	30
2.40	General maintenance schedule chart.	31
3.1	Vehicle Identification Number Format [4]	33
3.2	Sign up page	36
3.3	Sign up page	37
3.4	Dashboard	38
3.5	Car Repair list	39
3.6	Find Workshop	40
3.7	Notification List	41
3.8	Settings	42
3.9	Driver driving prompt	43
3.10	Driver location based distance	44
3.11	Products view	45
3.12	Products Details	46
3.13	Check out view	47
3.14	Pay total bill	48
3.15	Workshops on map	49
3.16	Pin layout of OBD-II scanner.	50
3.17	Ancel OBD-II Scanner (Front)	51
3.18	Ancel OBD-II Scanner (Back)	51
3.19	Iterative Incremental Model SDLC diagram [5]	53
3.20	Use case diagram	54
3.21	UC1: Register User	55
3.22	UC2: Sign In Case	56
3.23	UC3: Continue as Guest User	57
3.24	UC4: Get Data from Car	58
3.25	UC5: View Analytical Data	59
3.26	UC6: View Workshops in the vicinity	60
3.27	UC7: View car care/maintenance products	61
3.28	UC8: The user adds products to the cart	62
4.1	System Architecture diagram	64
4.2	Application Architecture Diagram	65
4.3	High Level Design	66
4.4	Database Design Diagram	67
4.5	Bykea Logo	68
4.6	Workshops on map	69
4.7	Sequence Diagram for Registered User part 1	70

4.8	Sequence Diagram for Registered User part 2	71
4.9	Sequence Diagram for Guest User	72
4.10	System behaviour Activity Diagram	73
5.1	Python Logo	76
5.2	Google Maps Logo	78
5.3	Google Firebase Logo	79
6.1	Manual Functional Testing	81
8.1	User Registration	91
8.2	User Sign in	93
8.3	Home Page	95
8.4	Car Repair Page	97
8.5	List of Products Page	98
8.6	Product Details Page	99
8.7	Check out Page	100
8.8	Bill payment Page	101
8.9	Towards Google Maps Page	102
8.10	View Workshops on Map Page	103
8.11	Notifications Page	104
8.12	Product Details Page	105

List of Tables

2.1	OBD-II Comparison Table	20
2.2	OBD-II Application Comparison Table	22
3.1	OBD-II Scanner Pin Configuration	50
3.2	UC 1: Register User	55
3.3	UC 2: Sign in Use case	56
3.4	UC 3: Continuing as a guest user	57
3.5	UC 4: Getting Data from Car	58
3.6	UC 5: Viewing analytical data	59
3.7	UC 6: Viewing workshops in the vicinity.	60
3.8	UC 7: Viewing car maintenance products	61
3.9	UC 8: Adding products to the cart	62
6.1	TC1: Sign Up	82
6.2	TC 2: Sign In	83
6.3	TC 3: Guest User	83
6.4	TC 4: Get Car Data	83
6.5	TC 5: Viewing analytical data	84
6.6	TC 6: Testing if workshops appear on Maps	84
6.7	TC 7: Testing if a user can successfully buy products from the application	85
6.8	TC 8: Testing if a user can successfully buy products from the application	85
6.9	TC 9: Testing database integration.	85
6.10	TC 10: Testing if cloud back up is integrated successfully	86
6.11	TC 11: To test if driver state is correctly recognised	86
6.12	Test Case 12: Application Installation	87
6.13	Test Case 13: Application Launch	87

Acronyms and Abbreviations

API	Application Programming Interface
APK	Android application PacKage
CAN	Controller Area Network
CU	Control Unit
DLC	Diagnostic Link Connector
DTC	Data Trouble Codes
ECU	Electronic Control Unit
EFI	Electronic Fuel Injection
GUI	Graphical User Interface
ISO	International Organization for Standardization
KM	Kilo Meter
MIL	Malfunction Indicator Lamp
OBD-II	On Board Diagnostic II
OS	Operating System
SDK	Standard Development Kit
SDLC	Software Development Life Cycle
TC	Test Case
UC	Use Case
UI	User Interface
URL	Uniform Resource Locator
UUID	Universally Unique IDentifier
VIN	Vehicle Identification Number

Chapter 1

Introduction

1.1 Overview

People often have bad experiences with maintaining their cars due to multiple reasons such as office routines, time constraints, forgetting the last time they went to the workshop for maintenance. Would not it be nice to have an application on your phone that will take care of the needs of your car on your behalf? As we know, during the past year due to the global pandemic, the lives of billions have been affected. Many people lost their jobs, many people had to start working from their homes, and in many cities of many countries' lock downs were imposed multiple times. During these times, where people have been thrown off their routines, many people faced issues with their cars as their daily commuting was affected, it had a big toll on the general health of their cars as cars require periodic maintenance and care. Many people forgot to take care of their cars and people having existing problems with their cars couldn't take them to a certified mechanic to get them electronically scanned. We understand that this pandemic has had an impact on a global level which gives rise to a strong possibility that a lot of people over the world would be facing related issues. We checked the market for self-diagnostic tools, we found a few systems/devices designed to scan and diagnose cars. Although what we thought was missing from the market was an application that would be connected to your car and serve as guardian of your car. An application that would let the owners connect with their cars in a better way. This is where our proposed system comes into play. A system that will serve as a caretaker of your car by keeping a check on many factors such as oil serviceability, oil viscosity, fuel efficiency, apparent tyre health, mileage, etc. Along with the diagnostics, our proposed system will also set reminders based on your car's needs and will remind you in advance whenever your car would require routine maintenance. It will also suggest

different local dealerships on a map provided in the application as well as helping the owner of the car to know the prices of the essentials and the accessories.

1.2 Problem Description

During the pandemic, most of the people lost their jobs, had to work from home, and in some countries, there was a strict lockdown situation therefore many people were using their cars without maintenance or they did not use them at all. Therefore, maintenance could not be done and people got their vehicles in a worse condition where they ended up paying a lot more than regular costs. Some people are too busy in their daily life routines due to which they cannot frequently visit the workshop to know the condition of their vehicle as a result either they get the maintenance done too early or too late.

ODometer rollback fraud is a legal offense carried out by many people around the world. This unethical activity is carried out to increase the value of a vehicle to deceive the buyer into thinking that the car has low mileage which means that the engine is relatively new. Odometer fraud is more common than most people think [3]. Due to this phenomenon, the car sellers trick the buyers into paying more money. To solve this problem, we are going to use OBD2 Code Reader Scanner and an android application. To test the state of the market for ODOMeter rollback frauds, we conducted a survey (complete survey is given in section 2.2) that gave us the following results:

Have you ever changed the reading of the Odometer? (by yourself or through a mechanic)
93 responses

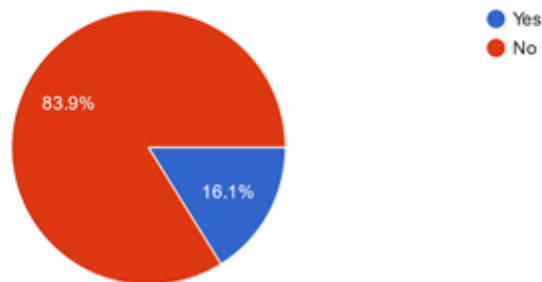


Figure 1.1: Pie chart of people who got their ODOMeter reading changed

If decreasing the Odometer reading offers you better value, will you decrease it?
93 responses

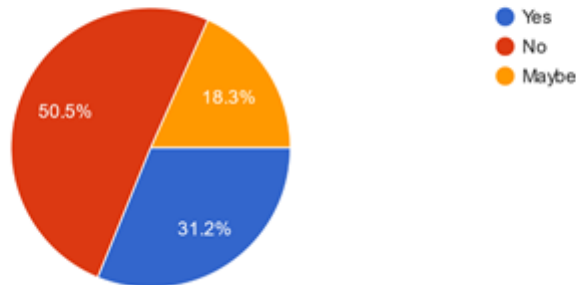


Figure 1.2: If decreasing the Odometer reading offers better value, will the seller decrease it?

If yes, what percentage of actual reading will you reduce?
64 responses

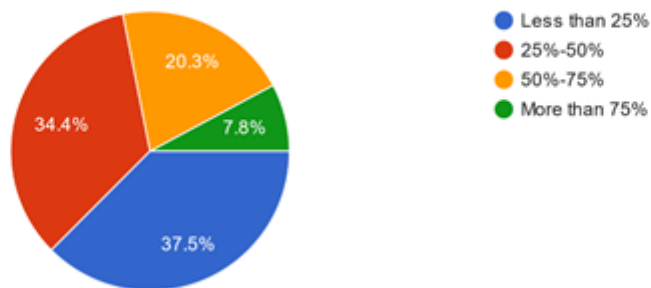


Figure 1.3: If yes, what percentage of actual reading will the seller reduce?

The above survey we conducted gives the following results which clearly show that:

- 16.1% of people have already changed their Odometer readings. From figure 1.1.
- 31.2% of people are prepared to change Odometer readings if they get better value by doing so. Also, 18.3% may change the readings. From figure 1.2.
- The amount of reading people are ready to change is visible from Figure 1.3 where 34.4% of people are ready to change readings by 25-50

This large amount of people having the mentality of changing the readings clearly shows that for higher value people who are not currently willing to do this illegal activity may be tempted to do it as a result a new buyer having no prior experience with buying cars will be heavily affected by ending up paying more than the actual value of the car. Therefore,

controlling these illegal activities can lead us to a better market for the consumer. Moreover, it can help in consumer protection from fraud and market quality assurance.

1.3 Existing Solutions

There are plenty of solutions that are available in the market that can solve the problems that we have mentioned individually, but none of them provide a one place solution to all the problems. As it has been stated that the proposed system revolves around 3 major problems.

1. Keeping car maintenance in check by connecting to the car using the OBD-II scanner, while identifying the problem in the paired vehicle by using the manufacturer-provided error codes.
2. Preventing ODOmeter forgery by creating online records of car maintenance logs that can be used on a large scale by creating a cloud-based database that can be publically available and also to law enforcement agencies to keep a check on the mileage of the car.
3. Reducing the hassle of buying car maintenance products and finding good workshops especially while relocating or during a trip to an unfamiliar city.

There are a lot of online tools available to track the maintenance of your car, for the examples, these are few popular Android/iOS apps that make use of the OBD-II scanners in 2021 [6] are:

- Torque Pro
- OBD Auto doctor
- InCarDoc Pro
- Car Scanner ELM OBD2 App, etc.

All of these apps available for Android and iOS on the Play Store and App Store respectively have similar motives for their target customer i.e, they identify the issue with the car that they are paired with, they help their customer view engine sensory data in numeric and graphical form. But there is no cloud-based system that keeps a record of the car maintenance, mileage logs for knowing future maintenance and no mechanism to prevent odometer forgery. Although there are systems and apps like “Carly app and scanner” which also takes help from an OBD-II scanner to analyze all the CU (Control Unit) of the car. Where it takes the mileage of the car and then compares the values of the gathered data, in case of inconsistency it informs that there might have been some forgery problem [7].

This solution seems very promising in terms of catching forgery but on the landing website of this app, they have themselves stated that forgery is easier to spot only if the manipulator is amateur because professional manipulators know their way around the ECUs so they can manipulate data at a much deeper level [8]. Thirdly, to find workshops in cities is easy if we search for workshops, but there is no way to know what prices would be offered and how much differences there will be in car maintenance products among the workshops. In light of above discussion on existing applications they all are uni dimensional solutions. One application solves error code scanning and another one comes with the Forgery prevention. Whereas we are trying to come up with a solution where we can provide a comprehensive solution for the problems car owners face.

1.4 Problem Statement

As it has been briefly explained in the introduction that there are many factors on why it's becoming harder and harder for a normal person to remember and keep a note of little things such as car maintenance. Personal vehicles make most of the market share of modes of transportation in urban setups. As cars require regular maintenance, it is hard for busy individuals to make time out of their busy schedules to find reliable workshops with reliable products. On the other hand, it's also hard for workshop owners to get adequate exposure to the consumer market. As every business sector is innovating bits and pieces of their existing system by shifting relevant functionalities online, there is a need for the car maintenance business, especially the less popular ones, to be visible to the consumer market. Secondly, after conducting a survey, we have highlighted an existing problem in the automotive industry known as "Odometer forgery", there are a couple of conventional ways to avoid this problem but the professional fraudsters have taken this into a deeper level where they change the car mileage readings at an electronic level to sell the vehicles for a far greater value than what a car actually has.

1.5 Objectives

- **Car Maintenance Logs** Maintain the car's maintenance logs including all details such as when was the last maintenance carried out, what things were included such as oil, oil filter, air filter, spark plugs, etc. And to inform on when to get the next periodic maintenance service.
- **Expected Maintenance** Know the maintenance situation before your vehicle is in an un-ideal technical condition. So, you can plan ahead and timely go to the workshop and get the maintenance done. The detection of the maintenance situation will be based on the number of kilometers covered since the last maintenance. To perform

maintenance at 4000 km, the application will send the first notification at 3500 km and then every 200 km which will be at 3500, 3700, and 3900 km respectively. We will be notifying users about Engine Oil, Oil filter, Air Filter, Battery, Tires, Wiper Blades.

- **Time Saving**

Providing all the details on the fingertips of vehicle owners will save their time by ensuring a good vehicle maintenance situation and timely alerts.

- **Avoid Fatal Problems**

Avoid major problems such as Frequent Overheating, Engine Damage, Break Failure, etc caused due to basic maintenance issues.

- **Transferring Vehicle Across User**

When a used car is bought, the new user has no idea about the previous maintenance history of that vehicle. So, we allow the seller to Transfer the car's previous maintenance records to a new user to prevent ODOmeter forgery.

- **Gauge Serviceability of an Engine**

Every vehicle requires service and car inspection after some specific number of kilometers (km). Most of the cars require these services and inspection at 5000 km.

1.6 Scope

As we are living in an era where everyone has smartphones, as of 2021, Android OS has acquired more than 72.84% of the smartphone market [6]. Using this opportunity, our proposed system makes use of smartphones to build a one-platform system for vehicle owners to keep a log of their car maintenance, get suggestions and reminders on when to get the next car service done. Secondly, providing a cloud-based system that helps in preventing and catching odometer forgery. Finally, the scope of our system extends to the car workshop market providing opportunities for both workshop owners and their customers to get more exposure and transparency using our application. The workshop owners will be able to sell their products through our application. This will help workshop owners to get more customer exposure and it will serve the customers by providing them with the verified prices of maintenance services and products.

Chapter 2

Literature Review

2.1 Evolution of Car Engines

Car engine technology has evolved over the years, it initially started with carburetors. Carburetors were used for decades, with great success, but when it comes to clean-air and precise air-to-fuel ratios, they were often unpredictable at best [9]. After carburetors, in the 1940s, automobile companies started to experiment with Mechanical Fuel Injection, inheriting the concept from commercial aeroplane engines. Many different technologies came into consideration but shortly after aeroplanes started using EFI (Electronic fuel injection) technology, it was also introduced in cars. As it was electronically controlled, cars needed an onboard computer to electronically control the fuel injection. The inception of the On-Board Diagnostic II (OBD-II) System came in 1995. Less than half of all vehicles manufactured in the U.S. were fitted with the new system in 1995, but the Federal mandate ensured that all 1996 models were OBD-II certified. OBD-II offered precise fuel delivery, tedious engine monitoring, and universal diagnostic connectors [9]. This trend was shortly followed by countries and manufacturers all over the world.

What has been explained in the paragraph above is the reason why every modern car made after 1996 has a diagnostic link connector (DLC) port (ELM327) which can help connect to the On-Board Diagnostic II (OBD-II) System of the car which can provide diagnostic data from the car's internal computer which we are going to use extensively in our proposed system.



Figure 2.1: Data Trouble Code format [1]

The error code derived from the DLC is called DTC (Data Error Code) and it is represented in the format displayed in the figure above.

The first letter indicates the category of the error which is usually in B,U,C,P

- **B** indicates a problem with the body of the vehicle
- **U** indicates a problem with the User Network.
- **C** indicates a problem with the Chassis of the vehicle
- **P** indicates a problem with the Power train, i.e. gearbox or engine, etc. of the vehicle.

The first digit of the DTC indicates if the fault is generic fault or Manufacturer Fault.

- **0** being a Generic Fault.
- **1** being a Manufacturer Fault.

While the the first digit of the last three indicate the subfamily of family codes that can either be P,C,B,U. And the last two digits are reserved for specific fault index.

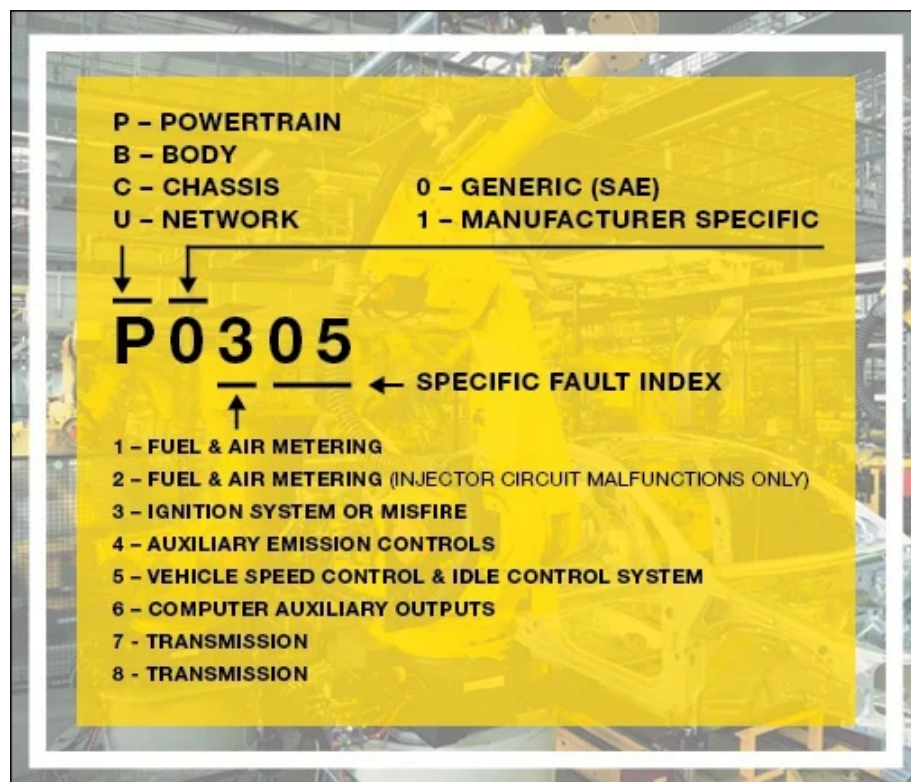


Figure 2.2: Data Trouble Code format. [2]

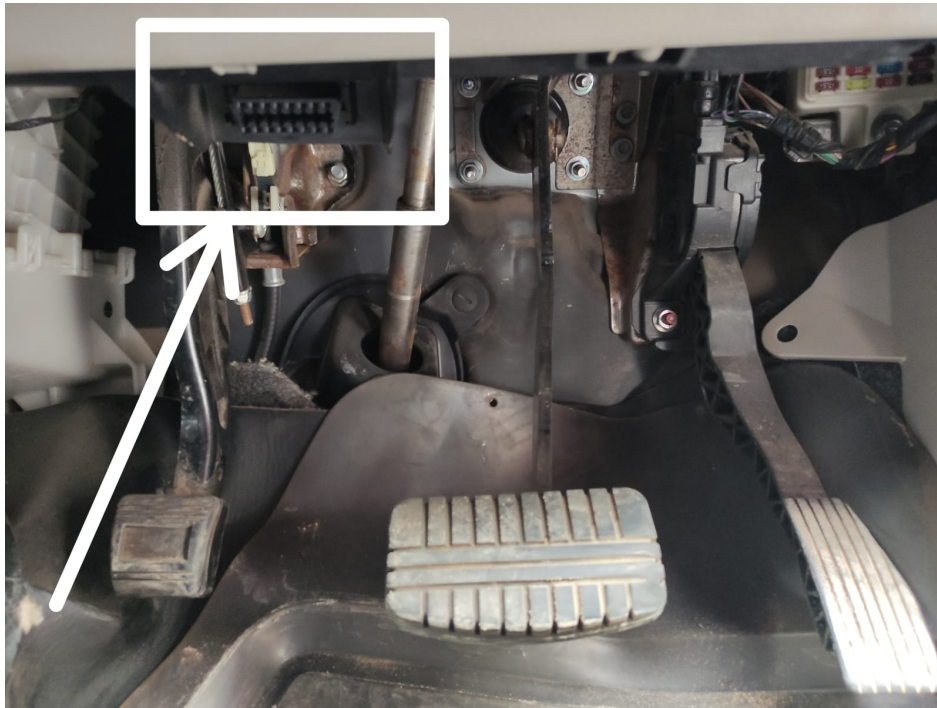


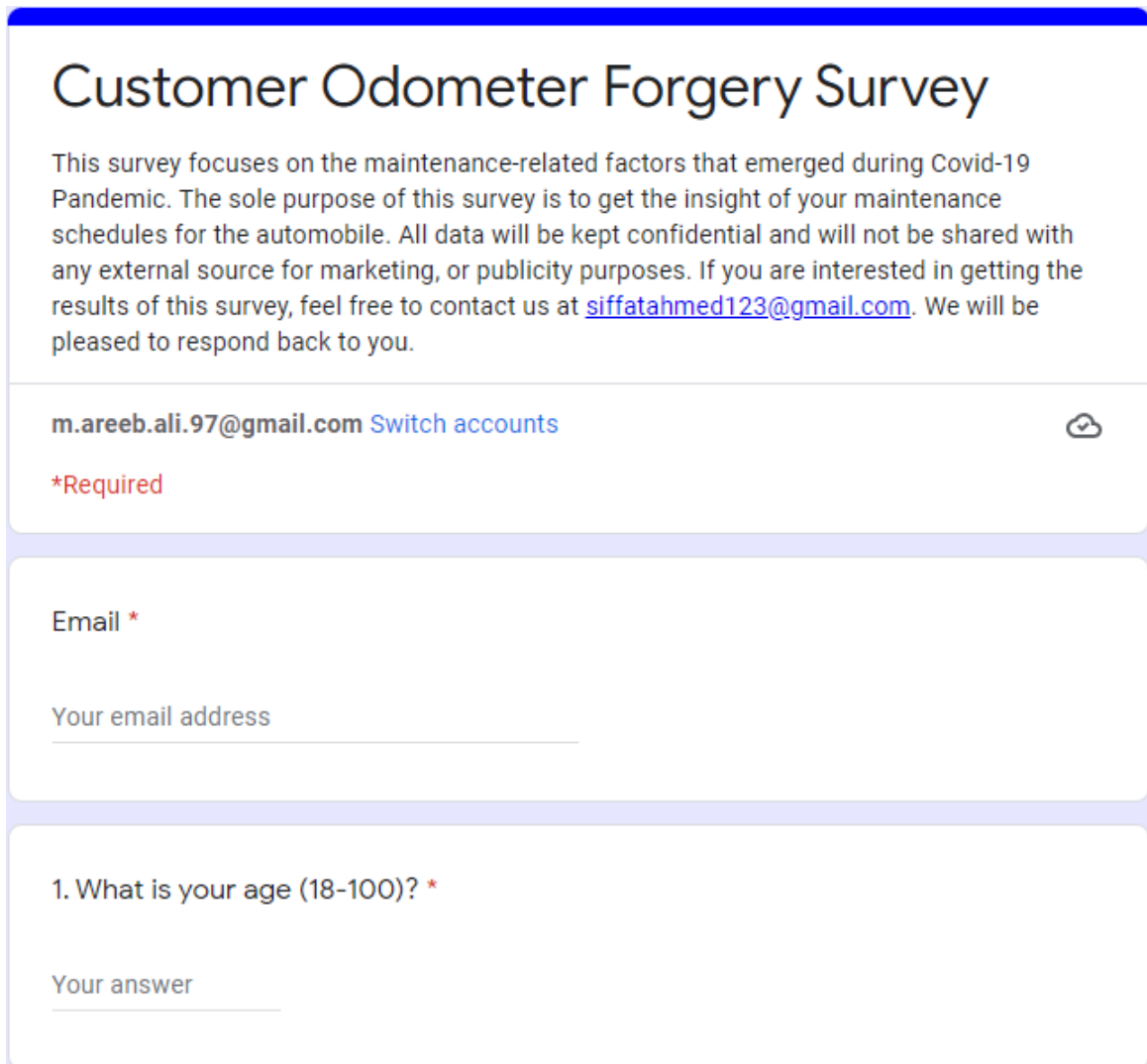
Figure 2.3: DLC port in a Mitsubishi Car

2.2 Conducted Survey

It is established that ODOmeter forgery is a serious problem in the car consumer market. To get a further understanding of the public opinion, we conducted a survey to understand deeply the public preference and opinion regarding this problem and whether or not people are willing to make use of this fraud to get a better value off selling their vehicles.


2.2.1 Questionnaire

Following is the questionnaire that was included in our survey.



Customer Odometer Forgery Survey

This survey focuses on the maintenance-related factors that emerged during Covid-19 Pandemic. The sole purpose of this survey is to get the insight of your maintenance schedules for the automobile. All data will be kept confidential and will not be shared with any external source for marketing, or publicity purposes. If you are interested in getting the results of this survey, feel free to contact us at siffatahmed123@gmail.com. We will be pleased to respond back to you.

m.areeb.ali.97@gmail.com [Switch accounts](#) 

*Required

Email *

Your email address

1. What is your age (18-100)? *

Your answer

Figure 2.4: Customer ODOMeter forgery survey

2. What is your current Education? *

- Below Secondary
- Secondary (Matric/O-levels)
- Higher Secondary (FSc/A-levels)
- Graduate
- Master
- Doctorate

Figure 2.5: Education Level participants who were the part of the questionnaire

3. Which region of Pakistan are you in? *

- Sindh
- Punjab
- Balochistan
- Khyber Pakhtunkhwa
- FATA
- Federal Capital

Figure 2.6: Region of the participants who were the part of the questionnaire

4. Which car brand do you own? *

Choose ▼

Figure 2.7: Brand of car owned by the participants who were the part of the questionnaire

5. What is your car model (year 1990-2021)? *

Your answer _____

Figure 2.8: Model year of the car owned by participants who were the part of the questionnaire

6. What car type you own? *

Sedan

Coupe

SUV

Other: _____

Figure 2.9: Information about the car owned by the Participants

7. What is the color of your car?

- Black
- White
- Red
- Blue
- Green
- Silver
- Other: _____

Figure 2.10: Information about the car owned by the Participants

8. Are you planning to sell your car? *

- Yes
- No
- Maybe

Figure 2.11: Intent of Participants about selling their car

9. Do you prefer to perform maintenance before selling your car to next client? *

- Yes
- No

Figure 2.12: Preference of Participants about getting there car serviced before selling

10. Have you ever changed the reading of the Odometer? (by yourself or through a mechanic) *

Yes

No

Figure 2.13: Did the Participants ever changed their vehicle's ODOmeter reading?

11. If decreasing the Odometer reading offers you better value, will you decrease it? *

Yes

No

Maybe

Figure 2.14: Asking the participants if they will forge the ODOmeter reading to get better value?

12. If yes, what percentage of actual reading will you reduce?

Less than 25%

25%-50%

50%-75%

More than 75%

Figure 2.15: Percentage of value reduction participants will change

2.2.2 Responses of the Questionnaire

Following are the responses we got in our survey based on a questionnaire. Question 1:

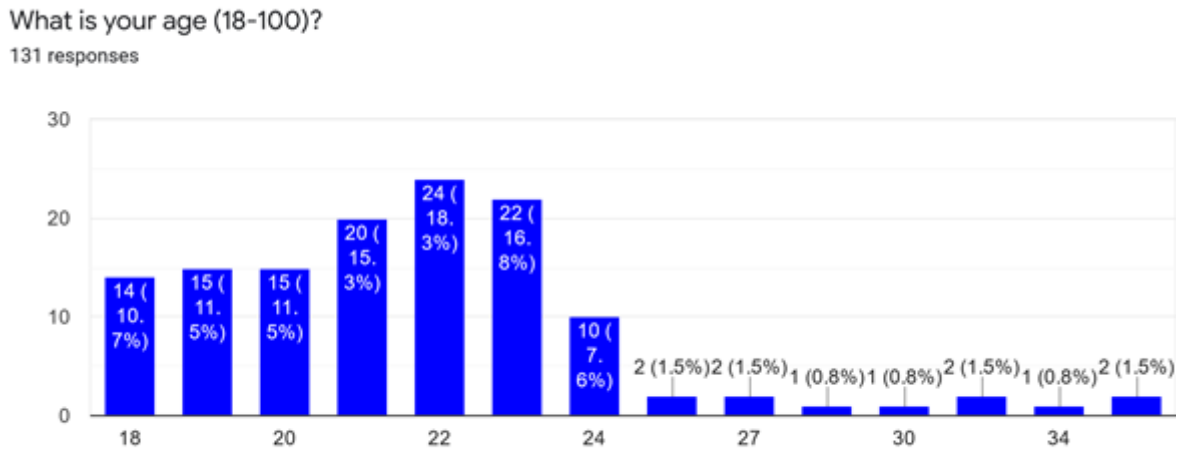


Figure 2.16: Results of Age group of participants

To get to know the age group of our research participants we performed asked participants to tell us age. We restricted it to be over 18 so we don't get any garbage data. Most of our participants range between age of 18-36 as they are most active in online platforms.

Question 2:

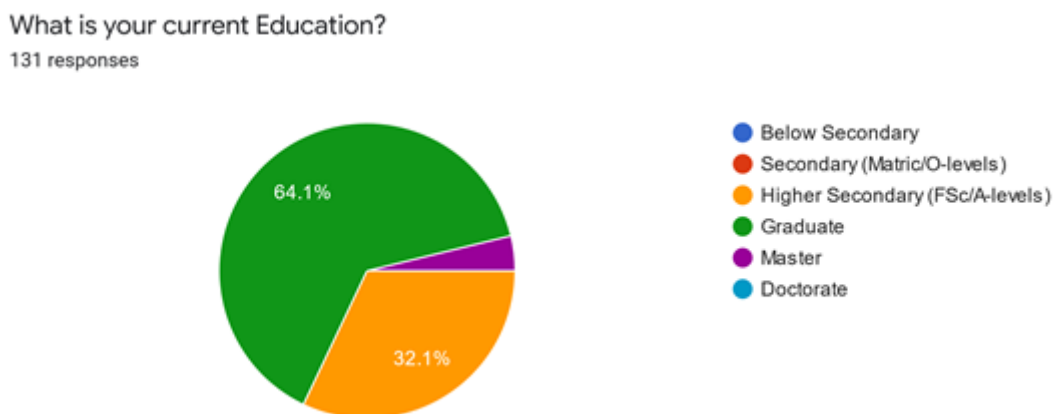


Figure 2.17: Result of current level of Education of the Participants.

We took the information related to the education of the participants as well which shows all the participants have done Higher Secondary level of education. 32.1% having done Higher Secondary and 64.1% done with the Bachelor's.

Question 3:

Which region of Pakistan are you in?
131 responses

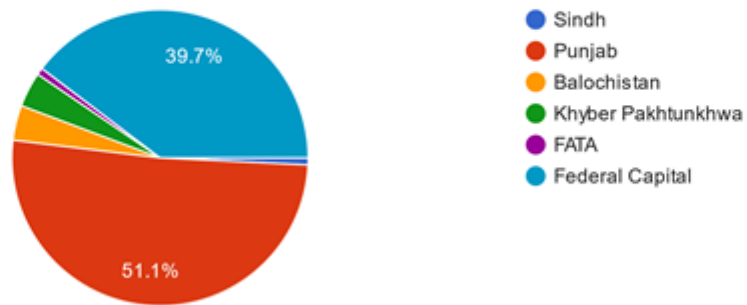


Figure 2.18: Results of users in different regions of Pakistan

We received responses from across the country with most of the responses coming from Punjab 51.1% and Federal Capital 39.7%. Sindh, Balochistan, Khyber Pakhtunkhwa and FATA gave response 0.8%, 3.8% 3.8% and 0.8% respectively.

Question 4:

Which car brand do you own?
131 responses

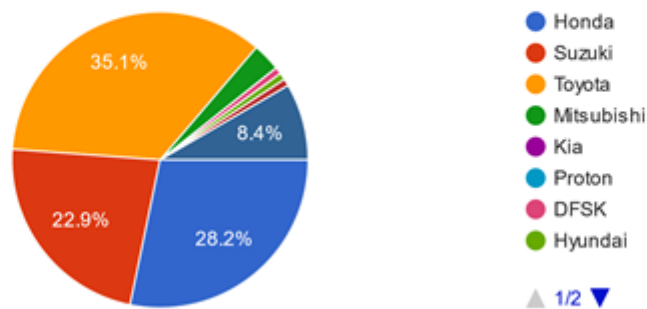


Figure 2.19: Result of Car Brands Most of the Users own (Part 1)

Which car brand do you own?
131 responses

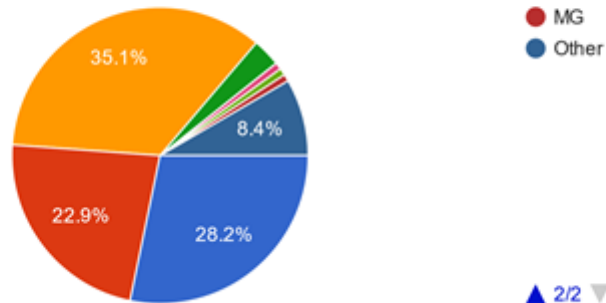


Figure 2.20: Result of Car Brands Most of the Users own (Part 2)

Above both Images show responses to the question regarding the vehicle different people own. Top 4 include Toyota, Suzuki, Honda, and Mitsubishi with votes of 35.1%, 28.2%, 22.9% and 3.1% respectively. There are 8.4% participants who are users of some other brand that was not mentioned in question.

Question 5:

Are you planning to sell your car?
131 responses

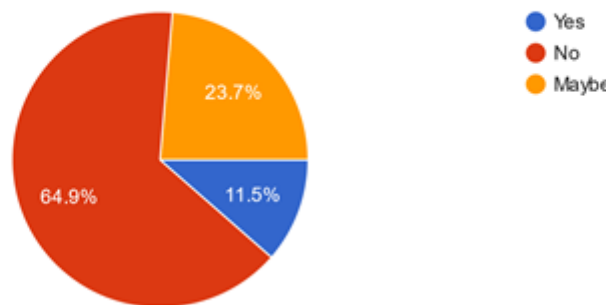


Figure 2.21: Result of Participant’s intent to sell the car

Above chart shows the number of people who are currently planning to sell car right now or maybe selling in near future. The responses show 11.5% people are willing to sell and 23.7% may sell where as 64.9% don’t want to sell car.

Question 6:

Do you prefer to perform maintenance before selling your car to next client?
131 responses

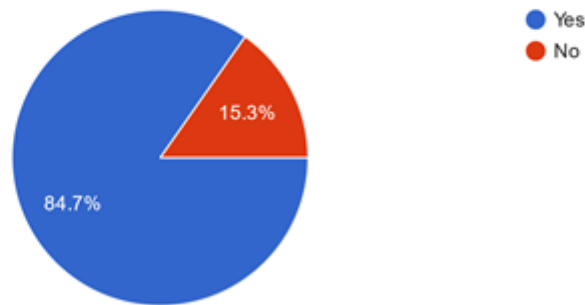


Figure 2.22: Result of participant's accepting the act of Odometer forgery

To know user mentality regarding whether they perform maintenance before selling or not. It shows 15.3% of people don't do it. That's why we are allowing transfer of vehicle logs so users can bargain price as per maintenance situation.

Question 7

Have you ever changed the reading of the Odometer? (by yourself or through a mechanic)
131 responses

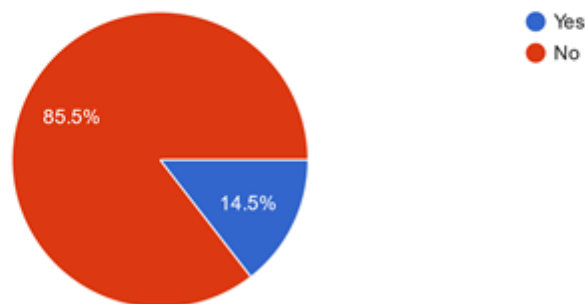


Figure 2.23: Results about people who have changed Odometer readings

Now this shows there are 14.5% of respondents who have changed Odometer readings which means they have done Odometer forgery.

Question 8:

If decreasing the Odometer reading offers you better value, will you decrease it?
131 responses

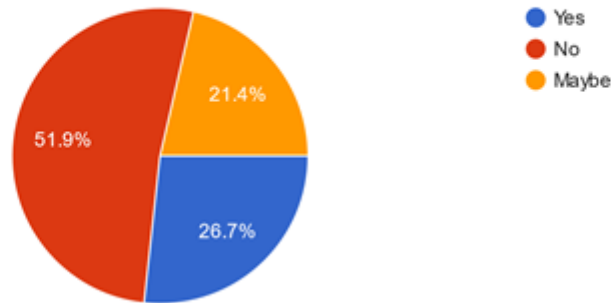


Figure 2.24: Results about people how will change value for better price

Now if changing Odometer reading offers better value 26.7% sellers are ready to forge and 21.4% will think about the possibility. Question 9:

If yes, what percentage of actual reading will you reduce?
91 responses

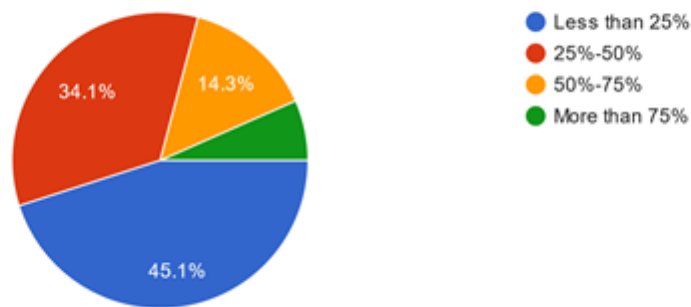


Figure 2.25: Result of percentage of actual reading that will be reduced

Above chart shows how much values will be changed by people who were ready to change Odometer reading and those who were gonna think about it. There are 45.1% who will be looking to reduce it by less than 25% and 34.1% sellers are ready to reduce values by 25%-50% value. Whereas 14.3% people are ready to change reading by 50%-75% and 6.6% are gonna change it by 75%.

2.3 Comparison of OBD-II Scanner Tools

There are many OBD-II tools available on the market but the ones that serve our idea are the ones that have Bluetooth connectivity. Bluetooth OBD-II scanners are much more convenient as compared to the traditional wired ones. This is a very optimal way your phone can connect to the scanner via Bluetooth and access all the features by the pre-installed app which communicates with the car using the scanner and then translates the information to the user. Following are the common OBD-II Bluetooth scanning tools which, in our case, are wireless plugs with Bluetooth connectivity:

Table 2.1: OBD-II Comparison Table

OBD-II Model	Features	Price	Drawbacks
Kitbest Bluetooth OBD2 Scanner and Adapter for Android	A very easy-to-use device. Known for its wide range of compatibility.	900-1700 PKR	Doesn't work with diesel cars and trucks. It is only compatible with android devices
Foseal WiFi OBD2 Scan Tool	Affordability, easy and fast connection, supports all platforms.	4000-6000 PKR	No battery saving mode, large in size, heavy, no confirmed list of car models
Ancel iCar2 icar 2 OBD2 Scan Tool	Auto-turn off to save car battery, works with a wide range of encodings, support all platforms	3000-3600 PKR	Compatibility problem with some cars
BlueDriver Professional OBD2 Scan Tool	Complete compatibility, perfect for Apple iOS and Google Android phones.	24000-28000 PKR	Very expensive, Wifi or Bluetooth connections can occasionally drop or be slow.

We are using Ancel iCar2 icar 2 OBD2 Scan Tool due to 4 main reasons:

- Affordability
- Fast Connections
- Availability



Figure 2.26: Ancel OBD-II Scanner (Front)



Figure 2.27: Ancel OBD-II Scanner (Back)

We have added support for automobiles made by Toyota, Honda, Suzuki, and Mitsubishi. The application has its error code information via Engine Control Unit (ECU) and makes sure there are no connectivity issues or incorrect reading issues in it. Already there are similar applications that are available on the Google Playstore and Apple app store but there are problems due to which users are unable to take full advantage of the available technology. Some of the available applications and their shortcomings are:

2.4 Comparison of Existing Applications

Table 2.2: OBD-II Application Comparison Table

Application Name	Major Issues	Platform
Torque Pro (OBD2 & Car)	Bluetooth connectivity issues, Doesn't show the ODOmeter reading	Cross Platform
OBD Auto Doctor	Connectivity issues and has problems of inaccuracy.	Cross Platform
EOBD facile	The interface is too cluttered, therefore prior experience with an OBD2 app is required. USB adapter support is missing.	Cross Platform
Carly app	A premium app with all the basic functionality but is expensive to buy and then ship to Pakistan	Cross Platform



Figure 2.28: Torque Pro Mobile Application logo.

Torque is a vehicle / car performance / diagnostics tool and scanner that uses an OBD II Bluetooth adapter to connect to your OBD2 engine management / ECU. [10]



Figure 2.29: OBD Auto Doctor Application logo.

OBD Auto Doctor is the leading OBD2 car diagnostics tool in your pocket! Ever wonder why the check engine light or MIL keeps going on? Check and reset the trouble codes with OBD Auto Doctor. With this OBD 2 diagnostic app, you can communicate with your car's OBD-II system quickly and easily, and turn your mobile into a highly capable automotive car scanner. [11]



Figure 2.30: EOBD Facile Application logo.

Run a car diagnostic with your smartphone and discover the reason why your engine light (or OBD light) is on ... No need to go to the car dealer for a diagnosis. [12]



Figure 2.31: Carly Application logo.

The Carly solution accesses your car's data through the OBD port and translates this information into powerful insights about your car's health, maintenance, and customization possibilities. [13] [14]

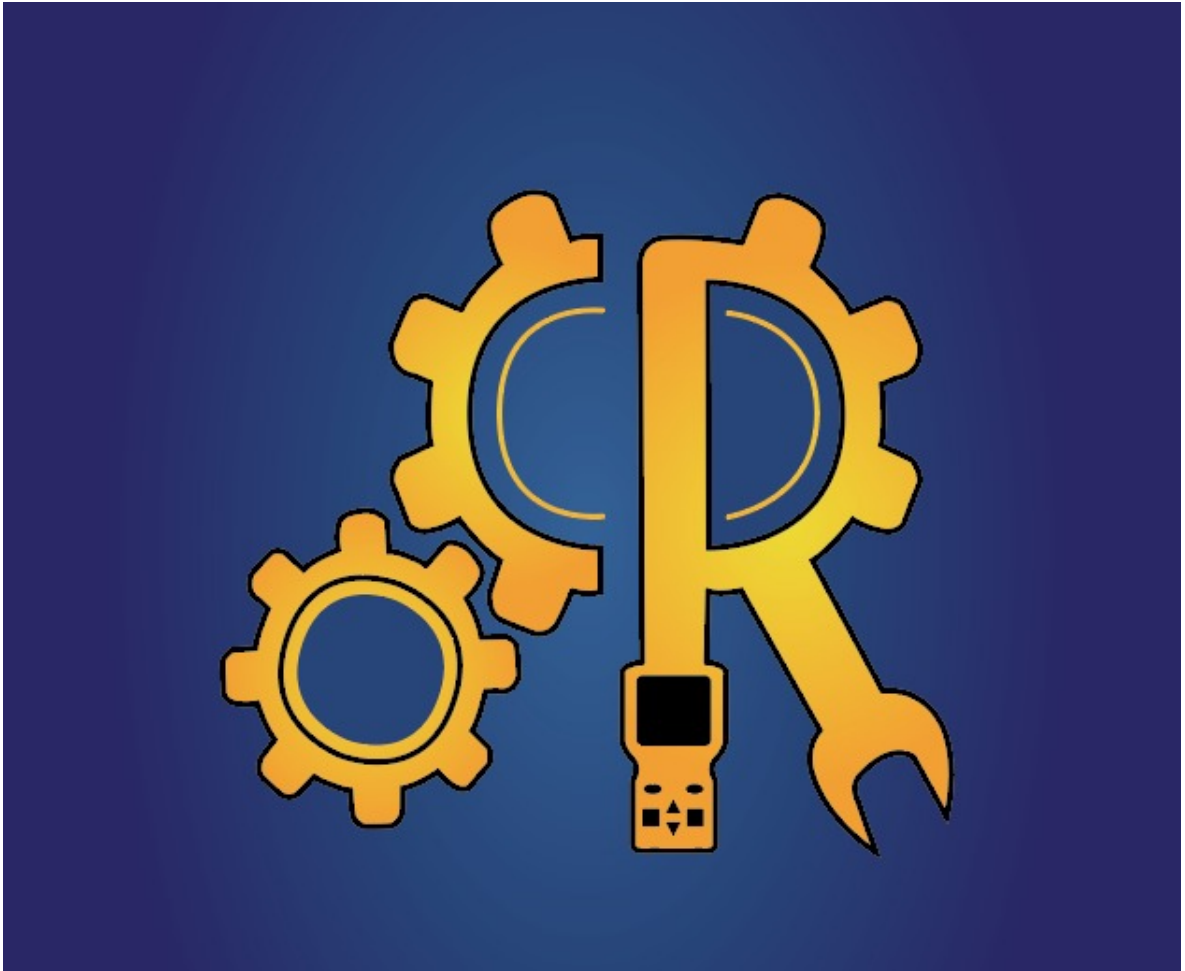
CarRolla: Smart Diagnostic Application

Figure 2.32: CarRolla: Smart Car Diagnostic Application.

CarRolla is a smart car diagnostic application which accesses your car's ECU to obtain useful data. CarRolla then uses the obtained information and combines it with plethora of features and options to give you more control over your car. No matter if it's about just keeping on a check on your car's maintenance needs or to generate a "Car worthiness certificate" to insure that your car is free of ODOmeter forgery, CarRolla has you covered.

Name	Carly	TorquePro	OBD-2 Auto doctor	EOBD facile	CarRolla
Smart Reminders	✓	✗	✗	✗	✓
Diagnostic Data	✓	✓	✓	✓	✓
ODometer forgery protection	✗	✗	✗	✗	✓
Back up of car data	✗	✗	✗	✗	✓
Error Identification	✓	✓	✓	✓	✓
Workshops in vicinity	✗	✗	✗	✗	✓
Product inventory	✗	✗	✗	✗	✓
Payment Options	✗	✗	✗	✗	✓

Figure 2.33: Have you ever changed the reading of the Odometer? (by yourself or through a mechanic)

2.5 Getting User Location to Show Workshops

A vehicle can cause any problem on any occasion despite doing all the maintenance. Such as radiator leakage, any type of oil leakage, damage to tires, etc. Under such situations, a driver wants to find a workshop quickly to get it fixed before further damage. But what if the driver is new to the area? He'll ask the native people for the workshop and may not be directed to a good shop. That's where our application comes into play. To solve the situations like the one discussed in the last paragraph we are integrating Google Maps into the application that will be helping with navigating the user to the workshops in the user's vicinity. To do this we take the user's current location in form of Longitudes and Latitudes. We show Google Map in the application containing markers for multiple workshops in the area and users can see the location of any workshop. As mentioned that our mode to locate on Google Maps is Longitudes and Latitudes we must mention two important things:

2.5.1 Longitudes and Latitudes

Longitudes and Latitudes in geographic systems are used for locating the exact location of anything on the earth.

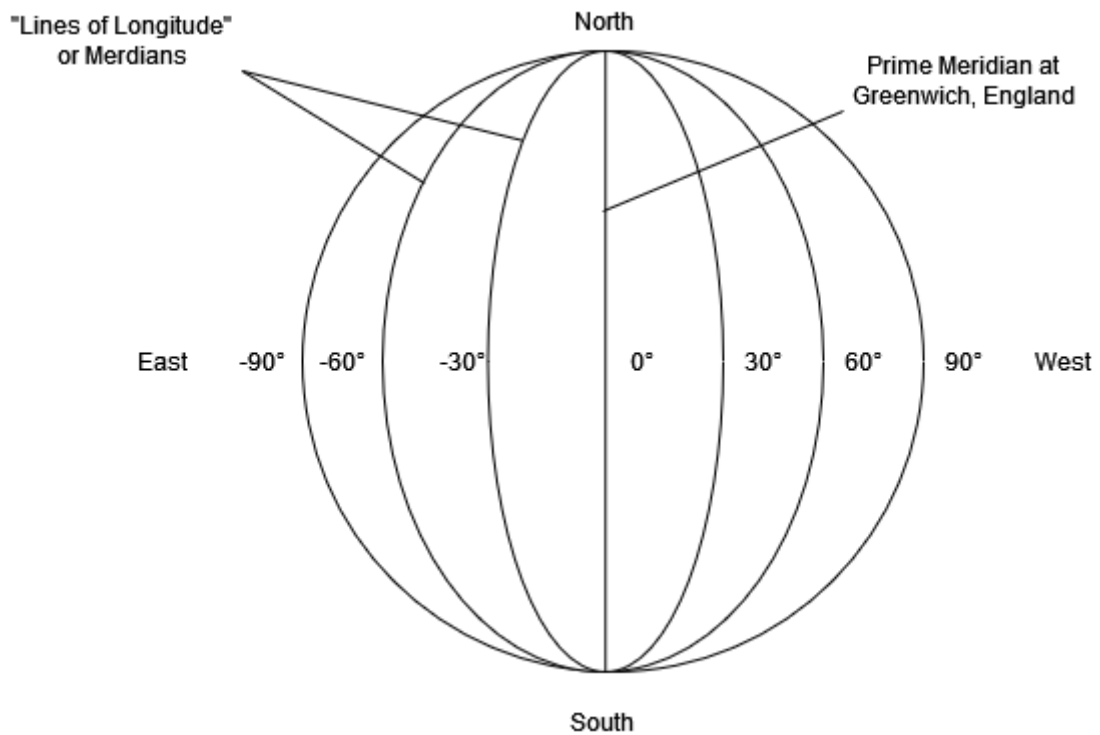


Figure 2.34: Graphical representation of Longitudes or Meridians

Longitudes:

The imaginary lines running from North to South are known as “lines of Longitudes” or “meridians”. The zero degrees line that passes through Greenwich, London is known as the “prime meridian”. The angle lies between -180° to 180° with lines separated by 30 degrees each as shown in the figure . We also call them 180° West to 180° East. [15]

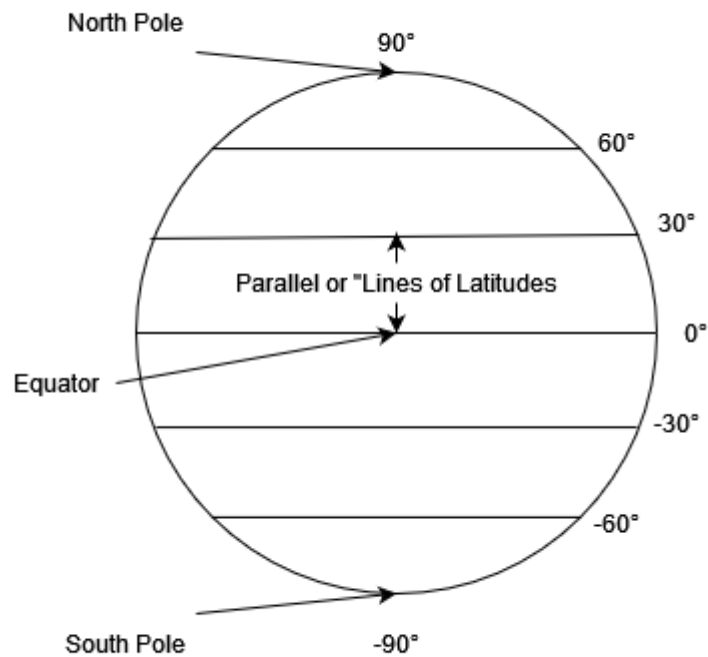


Figure 2.35: Graphical representation of Latitudes

Latitudes:

The imaginary lines running east to west are called “lines of Latitudes” or “Parallels”. They are parallel to the Equator of Earth. The angle lies between -90° to 90° with lines separated by 30 degrees each as shown in the figure. We also call them 90° North to 90° South. [15]

2.5.2 How Longitudes and Latitudes are measured?

Longitudes and Latitudes are typically measured in “degrees”. For accuracy of the location, they tend to have decimal places to locate the exact points. For example, Pakistan lies at 30.3753° North, 69.3451° East [16].

2.6 Research on Car Maintenance: Do All Cars Have Different Maintenance Schedules?


CarRolla: Smart Car Diagnostic Application features maintenance logs and smart reminders, for that it is important that we know what maintenance requirements do cars

require and do the requirements of cars belonging from different manufacturers differ from each other? To get an accurate approximation on when ordinary cars require maintenance, we checked maintenance schedules provided by different car manufacturers and different automotive servicing companies, the results were quite expected.

TOYOTA FORTUNER MAINTENANCE SCHEDULE

Maintenance Operations:

I = Inspect and correct or replace as necessary
R = Replace, change or lubricate
L = Lubricate



TOYOTA
GIBRALTAR STOCKHOLDINGS LTD
Agents for
Toyota Motor Europe NVSA

SERVICE INTERVAL: (Odometer reading or months, whichever comes first)	Odometer Reading									Months	
	x 1000 km x 1000 miles	10	20	30	40	50	60	70	80		
BASIC ENGINE COMPONENTS											
1	Valve clearance (1GR-FE engine)									I	96
2	Drive belts		I		I		I		I		24
3	Engine oil: 1GR-FE engine <See note 1>	Replace every 5,000 km (3,000 miles)									6
	Case A	R	R	R	R	R	R	R	R	R	12
	Case B	R	R	R	R	R	R	R	R	R	12
	2TR-FE engine	R	R	R	R	R	R	R	R	R	12
4	Engine oil filter	R	R	R	R	R	R	R	R	R	12
5	Cooling and heater system <See note 2>				I					I	24
6	Engine coolant <See note 3>				I					I	-
7	Engine oil cooler hoses and connections (1GR-FE engine) <See note 2>				I					I	24
8	Exhaust pipes and mountings		I		I		I		I		12
IGNITION SYSTEM											
9	Spark Plugs										
	Vehicles without three-way catalytic converters	I	R	I	R	I	R	I	R	I	I:12/R:24
	Vehicles with three-way catalytic converters				R					R	48
10	Battery	I	I	I	I	I	I	I	I	I	12
FUEL AND EMISSION CONTROL SYSTEMS											
11	Fuel filter <See note 4>									R	96
12	Air cleaner filter <See note 5>		I		R		I			R	I:12/R:24
13	Fuel tank cap, fuel lines, connections and fuel vapor control valve <See note 2>				I					I	24
14	Positive crankcase ventilation valve, ventilation hoses and connections (vehicles without three-		I		I		I		I		12
15	Charcoal canister (vehicles with three-way catalytic converters)				I					I	24

Figure 2.36: Toyota’s maintenance schedule chart.

CHASSIS AND BODY											
16	Brake pedal and parking brake	I	I	I	I	I	I	I	I	I	6
17	Brake linings and drums (include paking brake linings and drums)		I		I		I		I		12
18	Brake pads and discs	I	I	I	I	I	I	I	I	I	6
19	Brake fluid	I	I	I	R	I	I	I	I	R	I:6/R:24
20	Clutch fluid	I	I	I	I	I	I	I	I	I	6
21	Brake pipes and hoses		I		I		I		I		12
22	Power steering fluid	I	I	I	I	I	I	I	I	I	6
23	Steering wheel, linkage and gear box		I		I		I		I		12
24	Propeller shafts grease (including tightening of bolts) <See note 6>	L	L	L	L	L	L	L	L	L	6
25	Drive shaft boots (4WD models)		I		I		I		I		24
26	Suspension ball joints and dust cover	I	I	I	I	I	I	I	I	I	6
27	Automatic transmission oil				I					I	24
28	Manual transmission oil				I					I	48
29	Transfer oil (4WD models)				I					I	48
30	Differential gear oil		I		R		I			R	I:12/R:48
31	Front and rear suspension		I		I		I		I		12
32	Tyres and inflation pressure	I	I	I	I	I	I	I	I	I	6
33	All lights, horns, wipers and washer	I	I	I	I	I	I	I	I	I	6
Air conditioning filter											
34	For Thailand, Indonesia, the Philippines, Malaysia and Vietnam			R			R				-
	Others		R		R		R			R	-
	Refrigerant amount for air conditioner		I		I		I		I		12

- Notes**
- Case A:
For Colombia, when using fuel with an ethanol concentration greater than 10% or more than 50% of total engine running time.
 - Case B:
For all other countries, and only for Colombia in cases excluding case A.
 - After 80,000 km (48,000 miles) or 48 months, inspect every 20,000 km (12000 miles) or 12 months
 - First replace at 160,000 km (96,000 miles) then every 80,000 km (48,000 miles).
 - Including the filter in fuel tank.
 - Including checking pre-air cleaner. (1GR-FE engine only).
 - After submerged, be sure to apply grease within 2 hours regardless of the maintenance intervals.

Source: Toyota Fortuner Owners Manual, Part No: 01999-0K181

Figure 2.37: Toyota’s maintenance schedule chart.

HENDRICK HONDA HICKORY

5,000 MILE SERVICE	30,000 MILE SERVICE	105,000 MILE SERVICE
<p><i>Due at 5,000 / 10,000 / 20,000 / 25,000 / 35,000 / 40,000 / 50,000 / 55,000 / 65,000 / 70,000 / 80,000 / 85,000 / 95,000 / 100,000 / 105,000 & 110,000 Miles</i></p> <ul style="list-style-type: none"> • Replace Engine Oil, Oil Filter & Drain Bolt Washer • Tire Rotation • Inspect All Fluid Levels & Conditions (replenish as necessary) • Inspect the following: <ul style="list-style-type: none"> - Front & Rear Brakes - Brake Hoses & Lines (including ABS) - Constant Velocity Joint Boot - Drive Belt - Driveshaft Boots - Exhaust System - Fuel Lines Connections - Lighting & Horns - Parking Brake - Suspension Components - Tie Rod Ends - Steering Gear Box & Boots - Tires <p style="text-align: right; margin-top: 10px;">..... \$49.95*</p>	<p><i>Due at 30,000 / 60,000 & 90,000 Miles</i></p> <ul style="list-style-type: none"> • Replace Engine Oil, Oil Filter & Drain Bolt Washer • Replace Air Cleaner Element & Cabin Air Filter • Replace Transmission Fluid • Tire Rotation • Lube Door Hinges, Locks & Window Run Channels • Inspect All Fluid Levels & Conditions (replenish as necessary) • Install Wiper Blades • Install Valvoline's Fuel System Cleaner • Inspect the following: <ul style="list-style-type: none"> - Front & Rear Brakes - Brake Hoses & Lines (including ABS) - Constant Velocity Joint Boot - Drive Belt - Driveshaft Boots - Exhaust System - Fuel Lines Connections - Lighting & Horns - Parking Brake - Suspension Components - Tie Rod Ends - Steering Gear Box & Boots - Tires <p style="text-align: right; margin-top: 10px;">..... 2WD \$369.95* 4WD \$429.95*</p>	<p><i>Due at 105,000 Miles</i></p> <ul style="list-style-type: none"> • Replace Engine Oil & Oil Filter • Replace Air Cleaner Element & Cabin Air Filter • Replace Spark Plugs* • Tire Rotation • Inspect All Fluid Levels & Conditions (replenish as necessary) • 4WD Only: Replace Differential Fluid & Transfer Case Oil • Inspect the following: <ul style="list-style-type: none"> - Front & Rear Brakes - Brake Hoses & Lines (including ABS) - Constant Velocity Joint Boot - Drive Belt - Driveshaft Boots - Exhaust System - Fuel Lines Connections - Lighting & Horns - Parking Brake - Suspension Components - Tie Rod Ends - Steering Gear Box & Boots - Tires <p style="text-align: right; margin-top: 10px;">..... 2WD \$359.95* 4WD \$459.95*</p> <p style="text-align: right; font-size: 0.8em;">*Platinum plugs \$60 additional</p>

Figure 2.38: Honda maintenance schedule



Routine Service Rapid Data Sheet
 Maintenance Schedule, Flat Rate Times & Basic Service Data

Model: AN650A K7/8


The service interval should be decided by mileage or time, whichever occurs first.

Item	Interval	km	1 000	6 000	12 000	18 000	24 000
		miles	600	4 000	7 500	11 000	14 500
		months	2	12	24	36	48
Air cleaner element			-	I	I	R	I
Exhaust pipe bolts and muffler bolts			T	-	T	-	T
Valve clearance			-	-	-	-	I
Spark plugs			-	I	R	I	R
Fuel line			-	I	I	I	I
			Replace every 4 years.				
Engine oil			R	R	R	R	R
Engine oil filter			R	-	-	R	-
Transmission oil			R	-	R	-	R
Final gear oil			R	-	R	-	R
CVT filter			-	-	I	-	I
Throttle cable play			I	I	I	I	I
Throttle valve synchronization			I	-	I	-	I
			CA, only				
Evaporative emission control system CA (California) model only			-	-	I	-	I
			Replace vapor hose every 4 years.				
PAIR (air supply) system			-	-	I	-	I
Engine coolant			Replace every 2 years.				
Radiator hose			-	I	I	I	I
			Replace every 4 years.				
Brakes			I	I	I	I	I
Brake hose			-	I	I	I	I
			Replace every 4 years.				
Brake fluid			-	I	I	I	I
			Replace every 2 years.				
Tires			-	I	I	I	I
Steering			I	-	I	-	I
Front forks			-	-	I	-	I
Rear suspension			-	-	I	-	I
Chassis bolts and nuts			T	T	T	T	T
Flat rate time (CA)			4.0 H (4.1 H)	4.2 H	4.9 H (4.9 H)	4.2 H	7.1 H (7.1 H)

I: Inspect T: Tighten R: Replace C: Clean
NOTE: I = Inspect and clean, adjust or lubricate as necessary.

Inspect times (I) in the above chart do not include additional times for clean, adjust or lubricate

Figure 2.39: Suzuki's maintenance schedule chart.[3]



SERVICE MAINTENANCE SCHEDULE

SERVICE	MILEAGE INTERVAL	MONTHLY INTERVAL	SERVICE	MILEAGE INTERVAL	MONTHLY INTERVAL
REPLACE BRAKE FLUID	NONE	36	INSPECT ALL FLUID LEVELS	20,000	12
OIL CHANGE	5,000	6	INSPECT BRAKE HOSES AND LINES	20,000	12
CHECK BRAKES	10,000	12	INSPECT EXHAUST SYSTEM	20,000	12
INSPECT SUSPENSION COMPONENTS	10,000	12	CHECK BELTS	30,000	24
INSPECT TIE RODS, GEAR BOX AND BOOTS	10,000	12	REPLACE DUST & POLLEN FILTER	30,000	24
LUBE HINGES, LOCKS AND LATCHES	10,000	12	REPLACE REAR DIFFERENTIAL FLUID	60,000	48
ROTATE TIRES	10,000	NONE	INSPECT VALVE CLEARANCES	110,000	NONE
REPLACE AIR CLEANER ELEMENT	15,000	NONE	REPLACE SPARK PLUGS	110,000	NONE
CHECK PARKING BRAKE ADJUSTMENT	20,000	12	REPLACE ENGINE COOLANT	120,000	120

Figure 2.40: General maintenance schedule chart.

Conclusion:

After checking out oil and general maintenance schedules given out by different motorcar companies and major workshop companies, it's evident that most road legal vehicles shared the the same maintenance schedules like:

- Oil change after a mileage interval of 5000 Kilometers or after a monthly interval of 6 months.
- Inspection of different fluids (brake fluid, transmission fluid, engine coolant, clutch fluid, etc) and car parts after a mileage interval of 10,000 Kilometers or after a monthly interval of 12 months.
- Inspection of all fluids, car parts and exhaust system after a mileage interval of 20,000 Kilometers or after a monthly interval of 12 months.

The data that we found translates into the result that all cars have similar maintenance requirements. Although, due the advancement in technology, cars with hybrid technologies sometimes differ in such metrics from ordinary petrol engine and diesel engine cars. Modified schedules for uncommon cars will be added to our application as a part of future enhancements.

Chapter 3

Requirement Specifications

3.1 Proposed System

Just that there will be an application with the following features Car Rolla: Car diagnostic Application is a multi-service application that will provide services to a user by providing functionality to create and update car maintenance log, to diagnose issues in the car using car's manufacturer-provided error codes and it updates a cloud-based maintenance log which is regularly updated and can be later verified at the time of selling or buying of that car.

3.2 Proposed Solution

The proposed system uses a Bluetooth transmitter that will be plugged into the car's DLC (Data Link Connector) and will transmit the car's information received through the plug to the mobile device in which our application will be installed. DLC is a multi-pin diagnostic connection port for vehicles used for interfacing an OBD-II scanner with the control modules of the vehicle and accessing the vehicle's on board diagnostics and live data streams.

3.2.1 The Application Will Get the Following Information

The first thing that will be extracted by the application will be the Vehicle Identification Number (VIN) [17] which provides the information about the model and make of the car, along with its serial number. We also have to consider the fact that different companies have different diagnostic codes for their cars, although majority of the diagnostic Diagnostic Trouble Codes-DTC are shared by all car manufacturers as a standard.

There is a very rare exception for some cars; they don't have their VIN stored in their ECU. Such rare cases will be registered by asking the user to enter the VIN and government registration number manually.

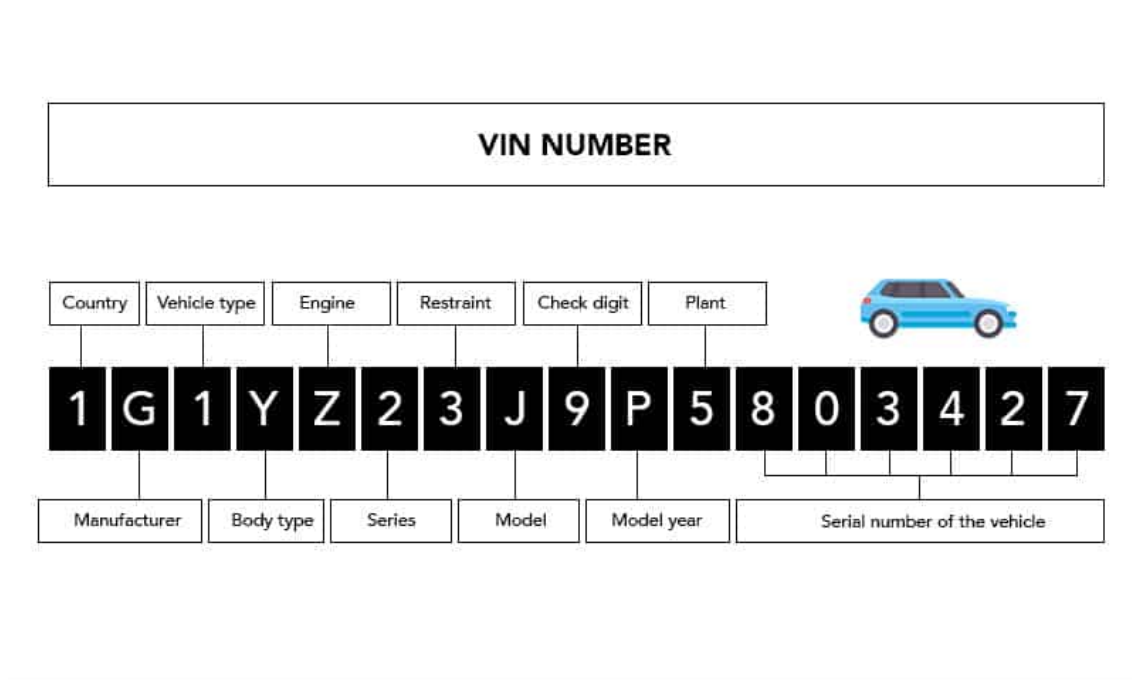


Figure 3.1: Vehicle Identification Number Format [4]

3.2.2 Using the Extracted Information

The information that is gathered using the DLC, will be used in many other features that are provided in the application. Our developed system takes care of the following tasks:

- The application keeps track of the car mileage.
- The application keeps a check on car maintenance based on the car movement.
- The application periodically scans the car and informs the car owner in case of any abnormality in the system.
- The application uses error codes for problem identification.

3.2.3 Driver State (in rest or driving)

Whenever Mobile's accelerometer sensors detect motion similar to driving, the application will ask the user whether the user is driving his car or not. If the user is driving his car, the application will start incrementing the ODOMeter in the application and start keeping track of the updated data. For detection purposes, we will use Activity Recognition Package of

flutter. [18] Flutter Activity Recognition Package supports many user movements based on multiple sensor data such as Running, Walking, Cycling, Tilting, Idle (still), Driving, etc. This Package automatically detects activities by periodically reading short bursts of sensory data and processing them using machine learning models. Now comes the question that it will heavily reduce device battery performance but it doesn't. In order to optimize battery consumption, the package stops activity reporting given that the device has been idle (or still) for some time, and uses low power sensors to resume reporting when movement is detected. Furthermore, it detects when a user stops performing a specific activity. In our case, for confirmation that the driving activity has stopped the application will ask the user to mark (or confirm) the end of the activity.

3.2.4 ODOmeter Rollback Fraud Prevention

Our application will periodically upload the car maintenance log and ODOmeter reading to the protected cloud storage, this will ensure that the ODOmeter reading against that vehicle is accurate and unchanged. Users buying the car can verify the actual ODOmeter reading with the one displayed on the cloud. This will ensure transparency. Also, the car log maintained on the cloud will be transferred to the new owner if it is sold. Secondly, along with the maintenance logs the seller can also share the online "car worthiness certificate" to prove that the car has not been forged.

3.2.5 Workshop and Product Recommendation

The Android application of our system will have a local workshop locator which helps recommend nearby workshops. To show locations of nearby workshops we are using Google Maps API. It will help users find the required workshop(s). It also has a list of car maintenance products compatible with the user's car such as paint-specific polish and wax, car shampoo, micro fibre cloth, etc, and their price range which helps the car owner have an insight into the expenditures beforehand. To do so all data regarding prices of car maintenance products are collected from multiple shops and manufacturers to provide economic and original products.

3.2.6 Payment Integration

Once a user selects to purchase products or services given in the application he can pay via in-app payment integrations. The integrated payment system is Bykea [19]. The integration is done using the provided unique URL for payment. Here is the provided link that we have used <https://bykea.cash/carrolla>

3.3 External Interface Requirements

- **User Interface**

The user will interact with the application designed for smartphones running on Android OS. The user will have different interfaces for every main division of the application with respect to the service that is being accessed i.e., Maintenance log, diagnostics, nearby workshops, etc.



Figure 3.2: Sign up page

This is account registration page that allows users to sign up using Email and password as well as options to sign up with Google, continue as a guest user, and go to the sign-in page.



Figure 3.3: Sign up page

This is a sign-in page that allows users to sign in using Email and password as well as options to sign in with Google, continue as a guest user, and go to the register account page.

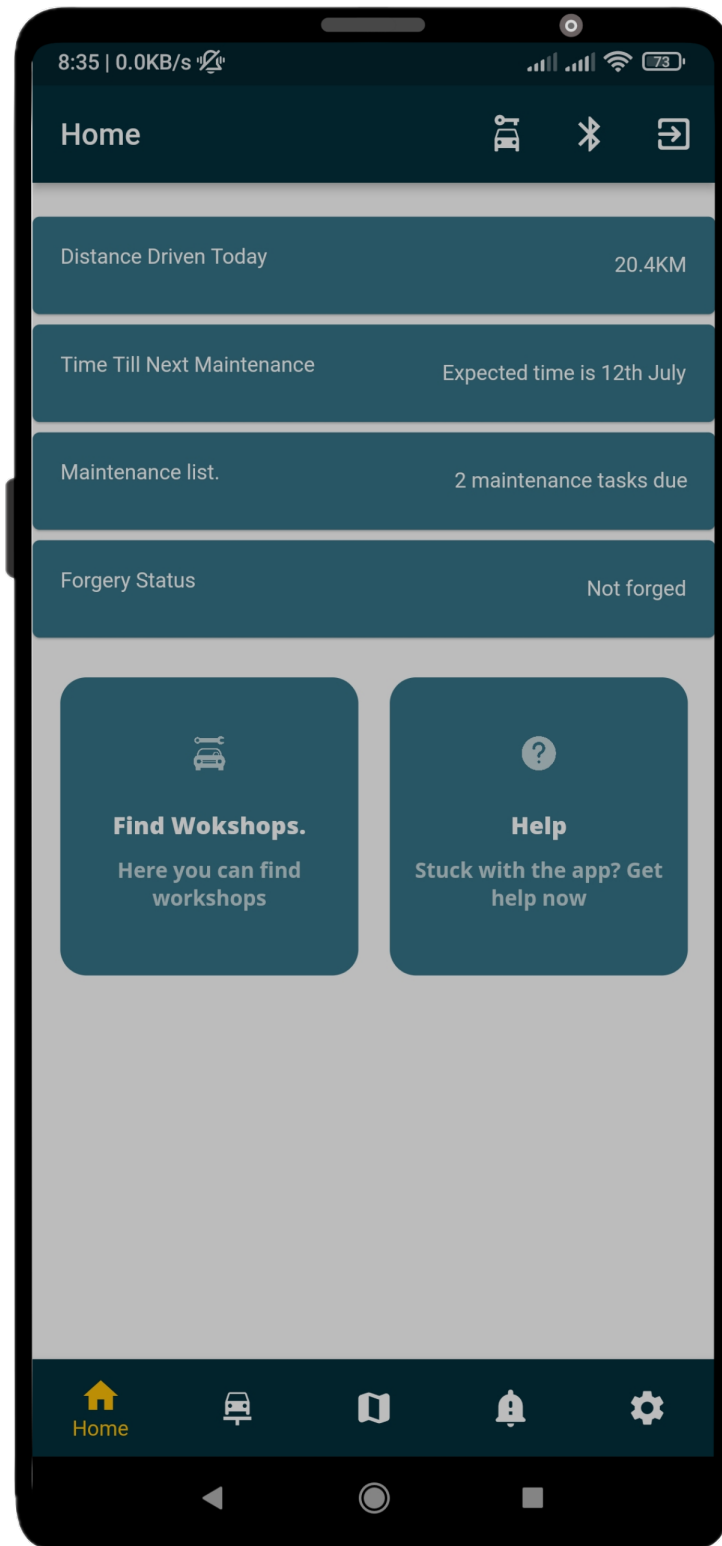


Figure 3.4: Dashboard

The homepage gives detailed information regarding vehicle data received via scanner and expected date for maintenance along with the forgery status.

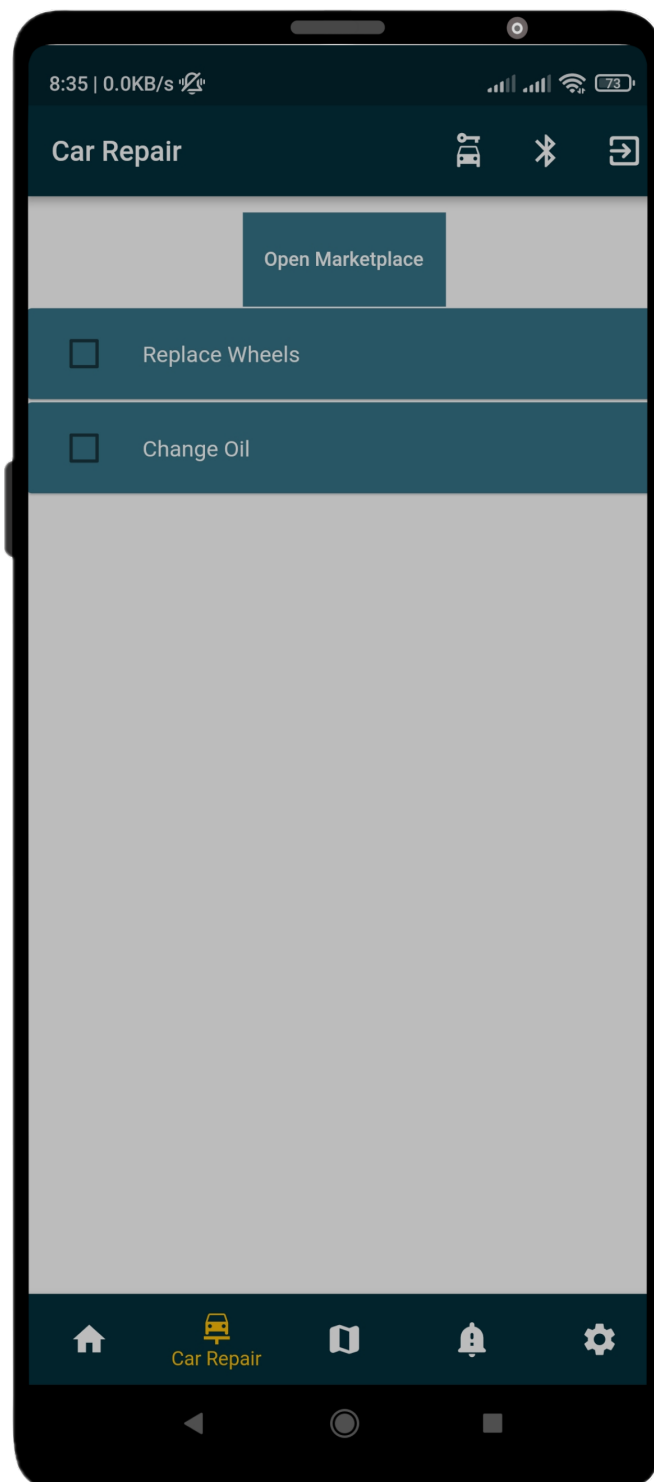


Figure 3.5: Car Repair list

This screen view shows the pending maintenance tasks to be done and lets the user go to the marketplace to purchase different vehicle products.



Figure 3.6: Find Workshop

In the above screen, we provide control to the user to find workshops in the area.

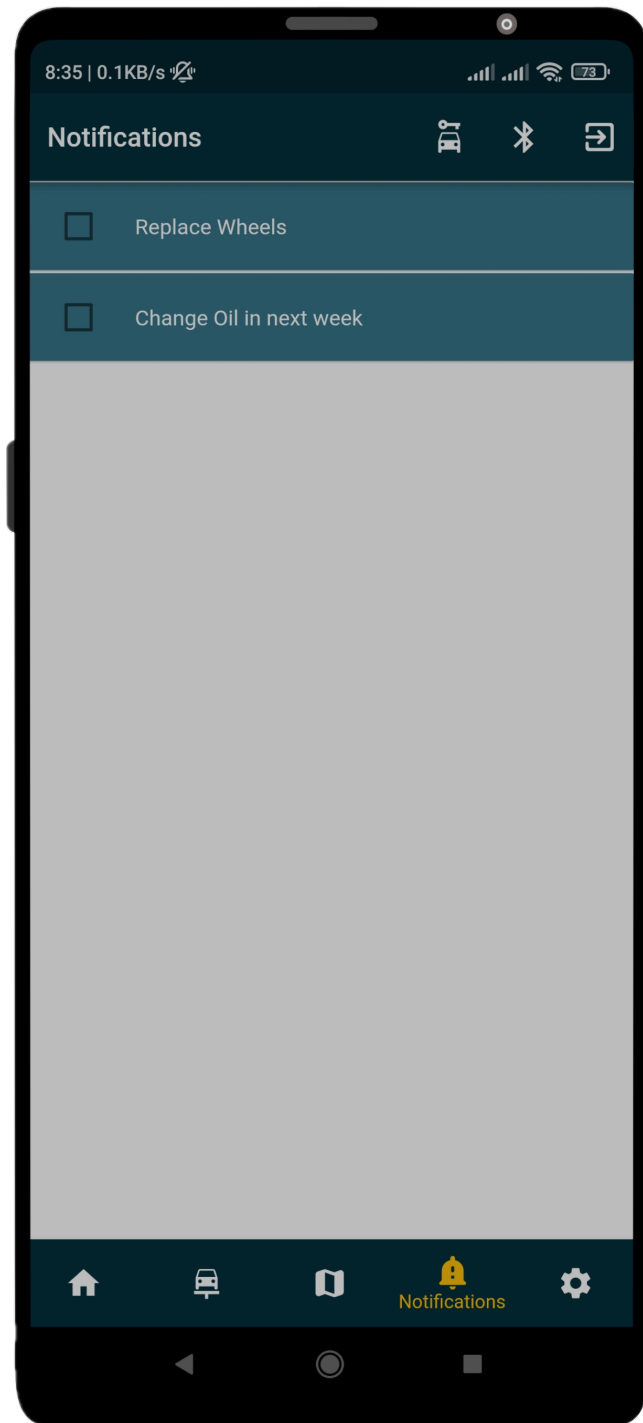


Figure 3.7: Notification List

Notifications screen allows the user to check the maintenance list and mark it as done. This is the list of notifications the user has received.

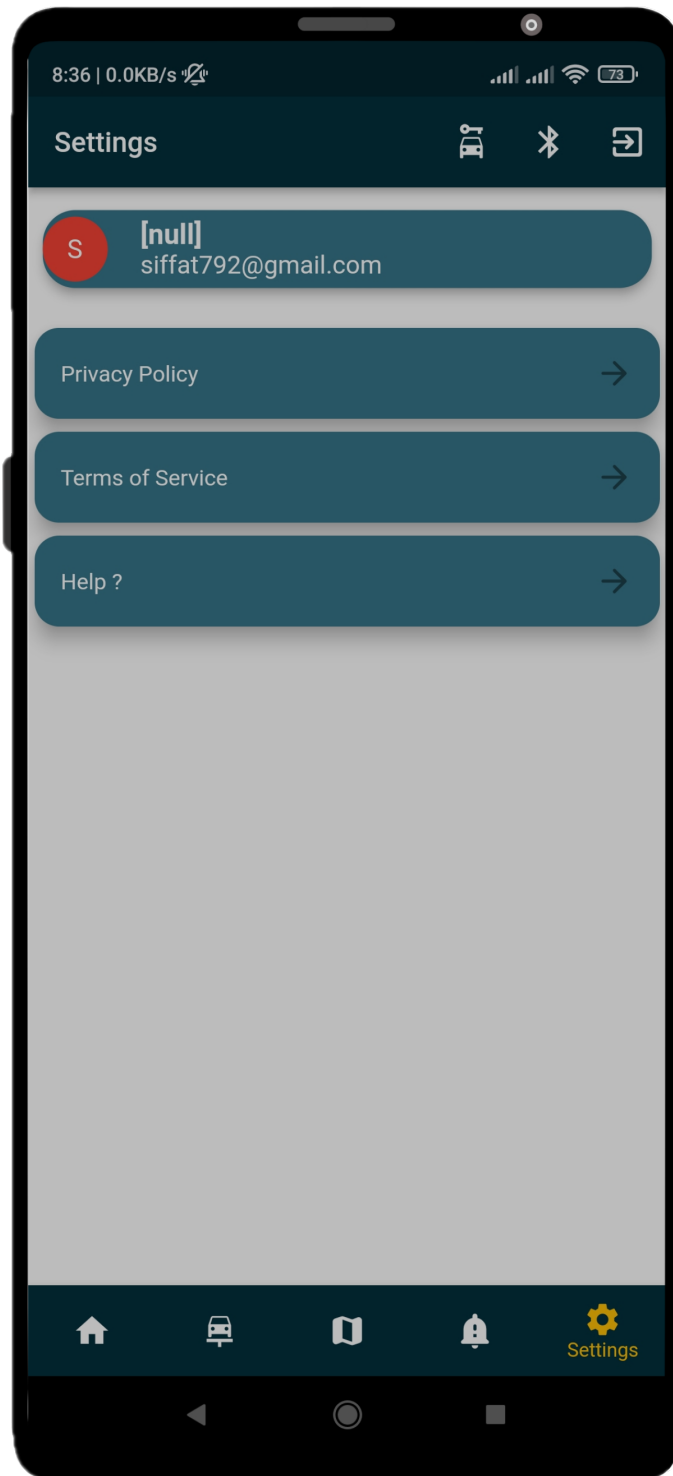


Figure 3.8: Settings

The Settings section provides information related to Privacy Policy, Terms of Service, Help as well as User's sign-in status. By displaying the user name and email address.

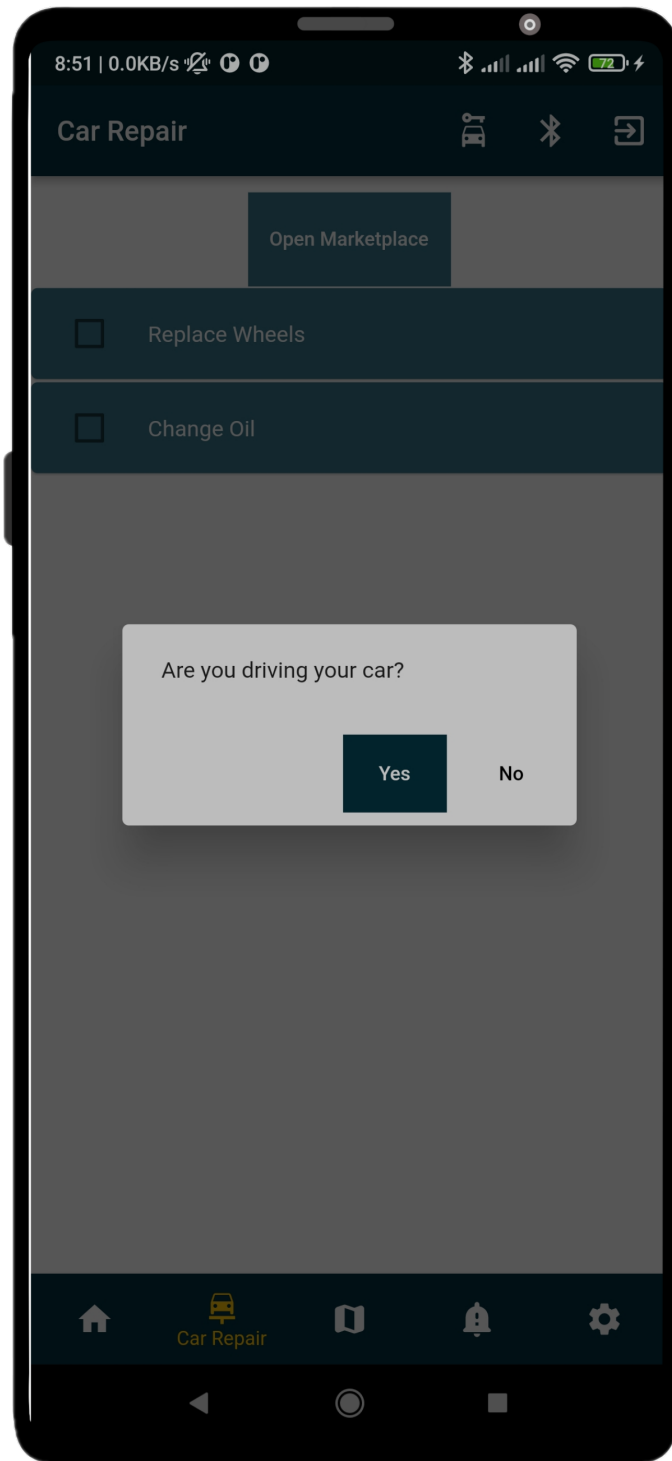


Figure 3.9: Driver driving prompt

This screen shows prompt of asking whether user is driving or not based on user's driving state.

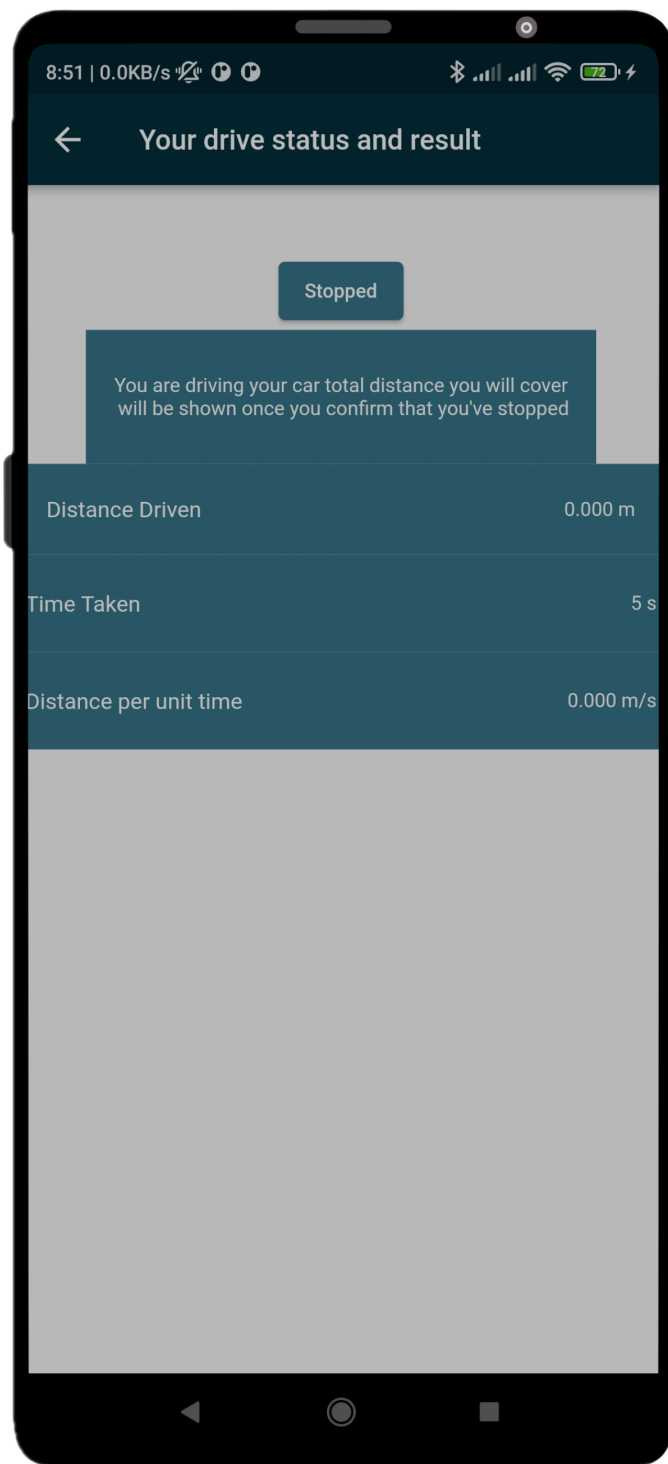


Figure 3.10: Driver location based distance

User's location based distance covered is calculated user see time it took to cover that distance and average speed.

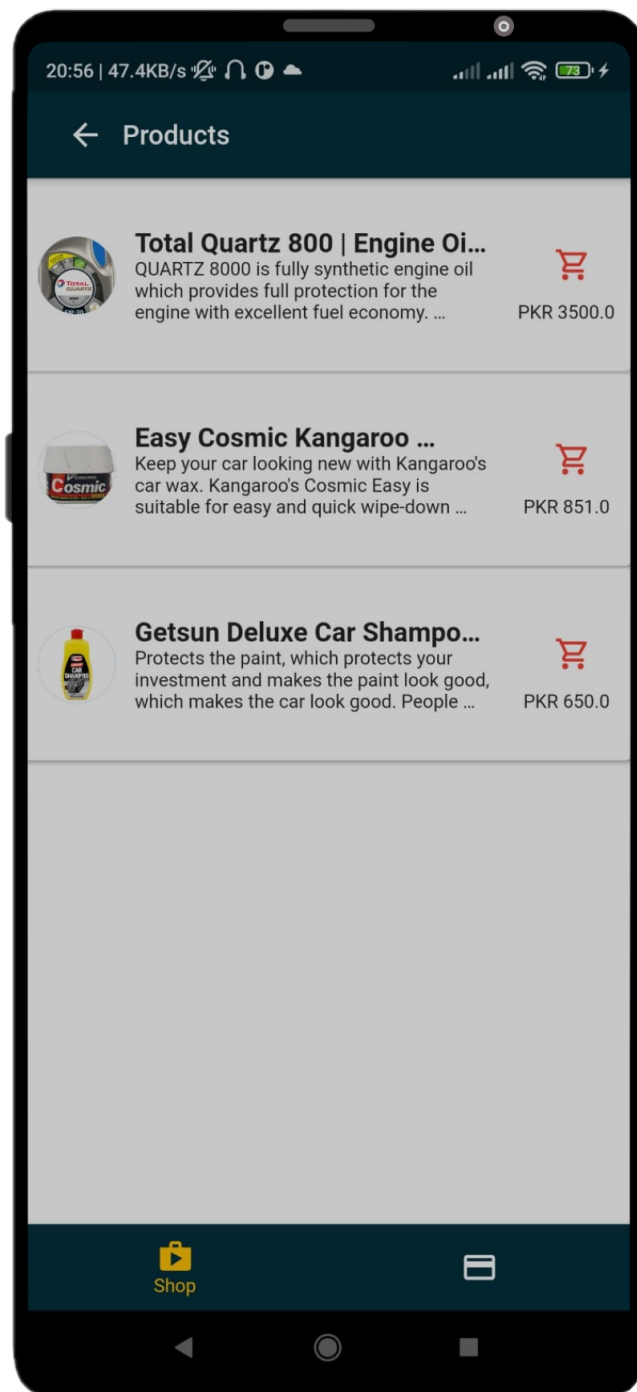


Figure 3.11: Products view

When a user accesses the marketplace he sees a list of products in the application and details by clicking the product. Along with the price and option to add products to the cart.

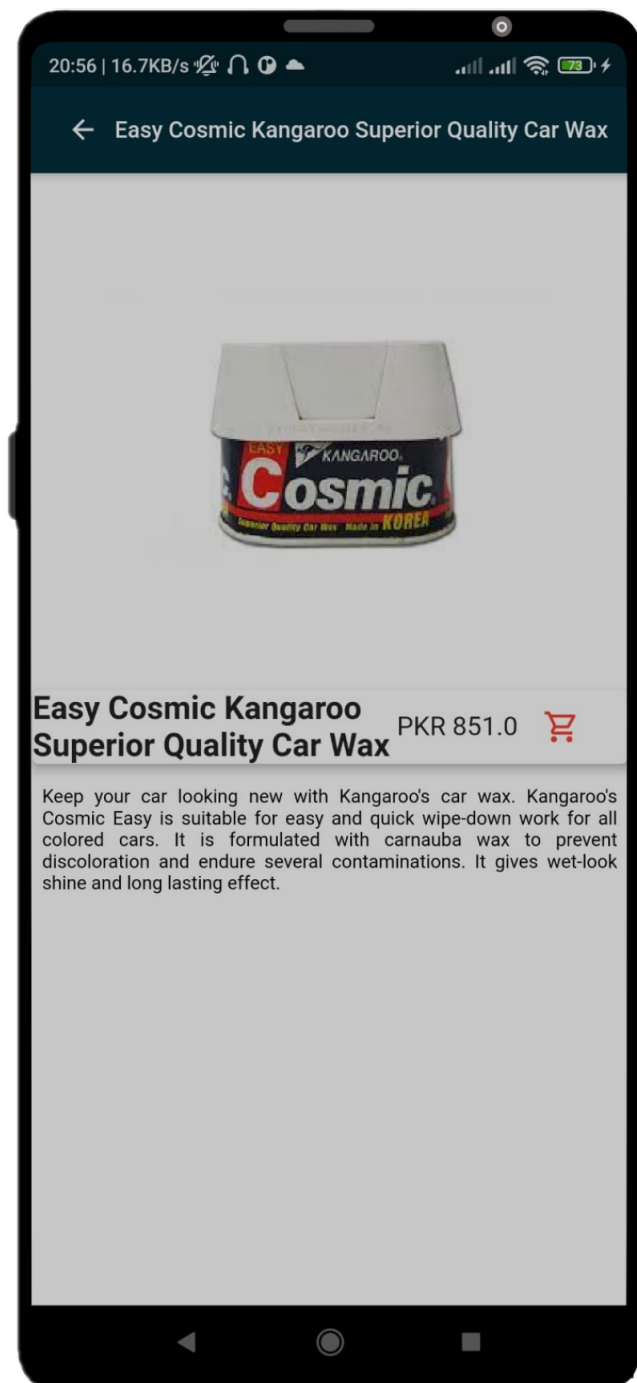


Figure 3.12: Products Details

The product details page shows all the product detailed description inserted by the seller.

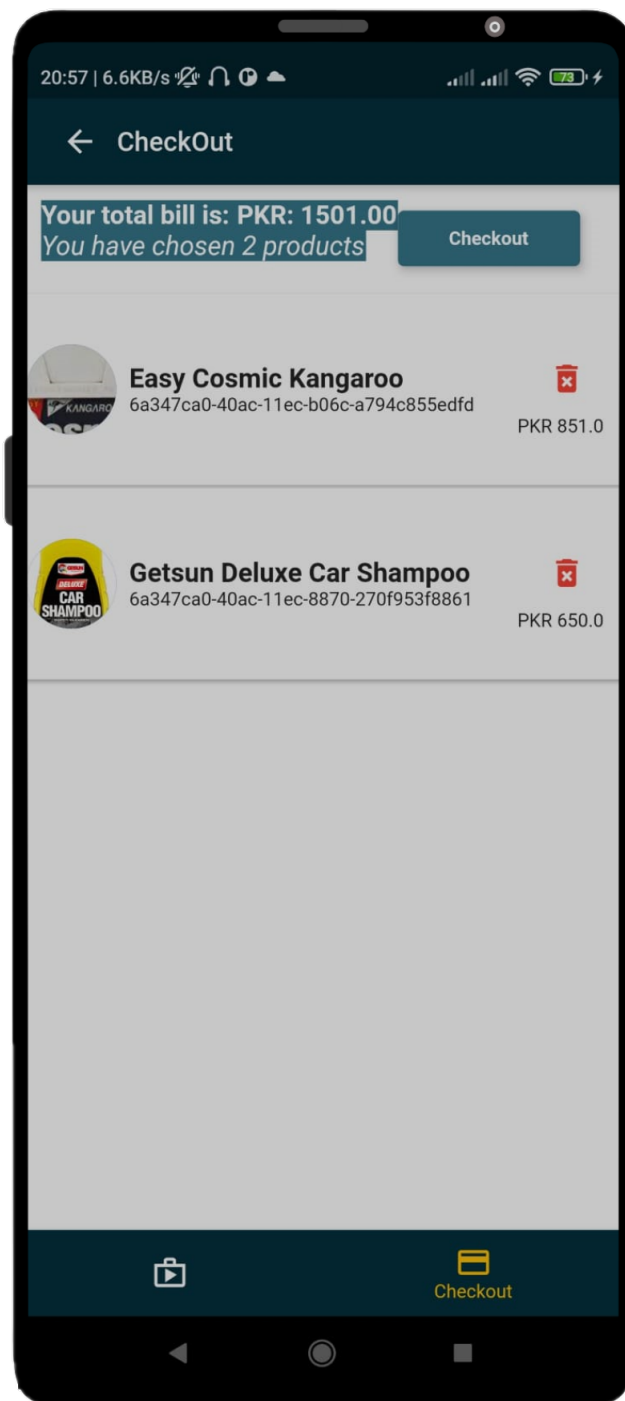


Figure 3.13: Check out view

On the Checkout page, the buyer can see the list of products in his/her cart along with the total bill. From where the user can go to the payment page.

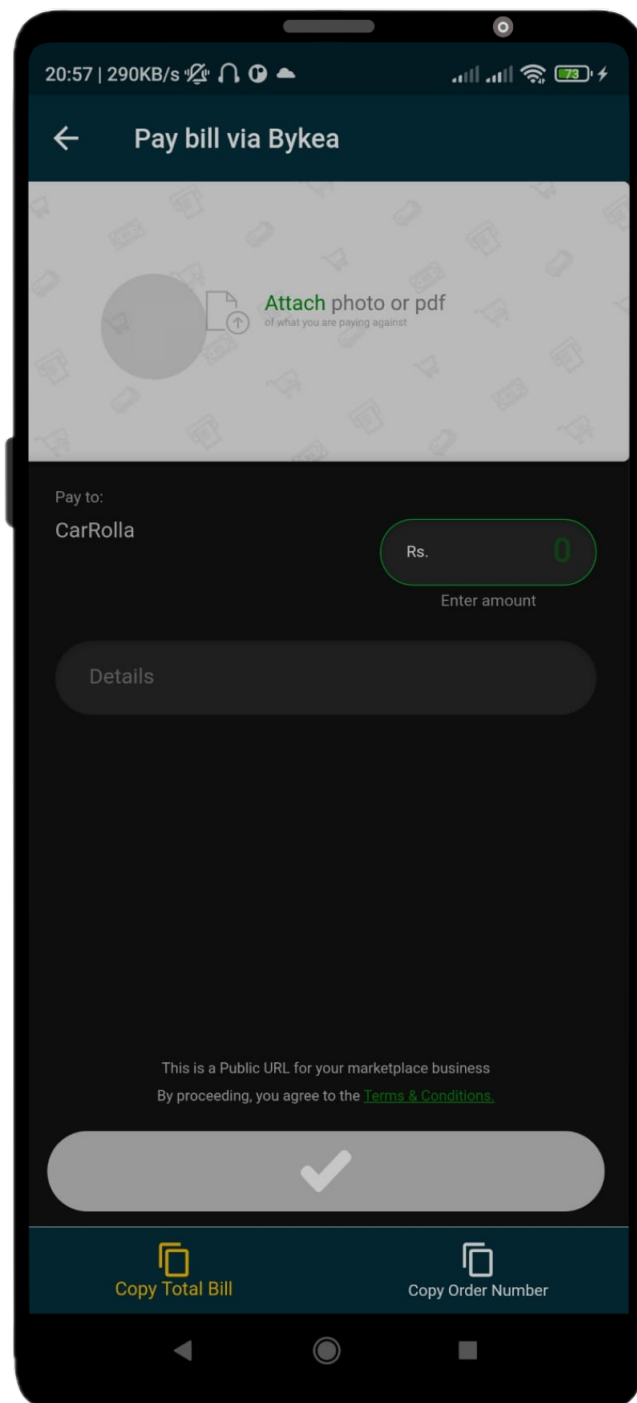


Figure 3.14: Pay total bill

This is a webpage shown inside the page that is the Bykea Payment Link <https://bykea.cash/carrolla>. Since data from previous controls can't be passed into a third-party web link so we ask the buyer to copy the total bill and paste in the amount text field and copy the order number and paste it in the details section. After clicking the tick sign user only has to enter their card details to pay.

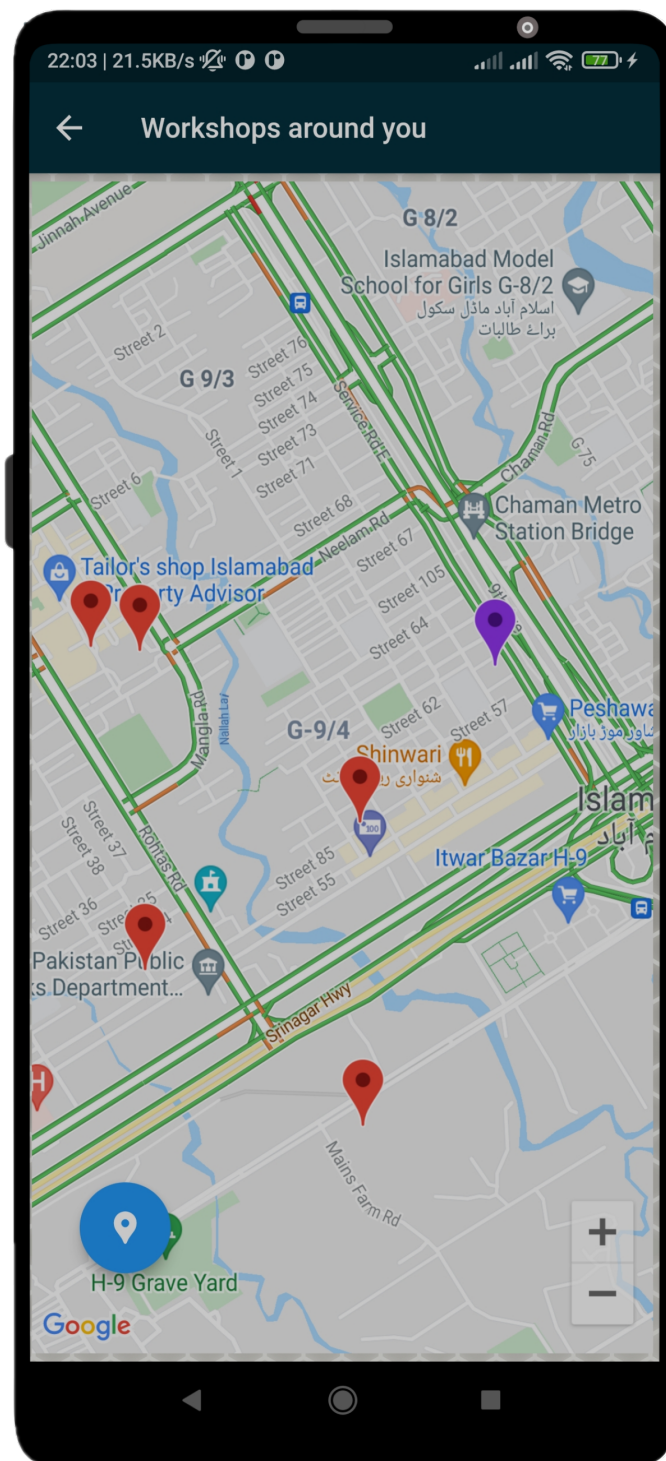


Figure 3.15: Workshops on map

The image shows the workshops close to the application user. Based on the user's current coordinates. For user location and workshop location, we are using Longitudes and Latitudes. In case the user's directions are not correct, tapping the left top corner of the map will fix it using Map's inbuilt compass functionality.

• **Hardware Interface**

The system mainly consists of a smartphone device running on either Android OS or iOS and a wireless OBD-II scanner with Bluetooth capability that will be physically plugged into the car and will send information from the car to the smartphone app. The OBD-II scanner has the following pin configuration:

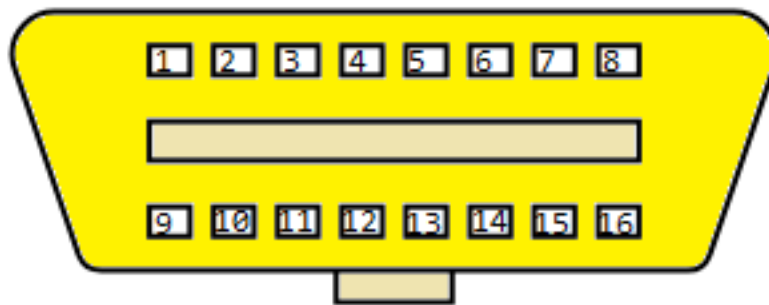


Figure 3.16: Pin layout of OBD-II scanner.

Table 3.1: OBD-II Scanner Pin Configuration

Pin	Description	Pin	Description
1	Vendor Option	9	Vendor Option
2	J1850 Bus+	10	J1850 Bus-
3	Vendor Option	11	Vendor Option
4	Chassis Ground	12	Vendor Option
5	Signal Ground	13	Vendor Option
6	CAN (J-2234) High	14	CAN (J-2234) Low
7	ISO 9141-2 K-Line	15	ISO 9141-2 K-Line
8	Vendor Option	16	Battery



Figure 3.17: Ancel OBD-II Scanner (Front)



Figure 3.18: Ancel OBD-II Scanner (Back)

- **Software Interface**

The application is designed on Flutter, which is an open-source Software development kit, designed by Google. Some functionality requires external plugins for maps and for Bluetooth functionality Java is used.

- **Communication Interface**

The application running on a smartphone will communicate with the OBD-II scanner using Bluetooth as a medium of communication. Secondly, we have a cloud backup for keeping track of vehicle activities and odometer forgery prevention.

3.4 System Requirements

Functional Requirements:

- Application is usable on the Android Mobile Devices.
- User can have registered account or use as guest user.
- Vehicle ECU data can be scanned by using application and an OBD scanner.
- Display analysed vehicle data and track it for maintenance alerts.
- Send notifications for pending maintenance.
- Vehicle maintenance logs are maintained.
- Serviceability alerts are sent.
- User logs are transferable at time of sale.
- User is able to find workshops in vicinity using embedded user Google Map.
- Mechanism to prevent ODOmeter forgery.
- Purchase vehicle care products.
- Pay for purchased products via Bykea cash.

Non-functional Requirements:

- **Cloud backup** to prevent loss of data.
- **User-friendly UI** design of the application.
- **Usability:** The system is usable for the claimed functional requirements and is efficient as well as effective.

3.5 Software Development Life Cycle (SDLC) Selection

For this project we are using **Iterative incremental model**.

In the **Iterative incremental model** a system's partial implementation is completed such that it is in deliverable state. Further functionality is added in next iterations unless software products is completed.

In this model we repeatedly perform requirements, analysis, design, code, and get software therefore each run of process is called as '**iteration**' and the with each iteration new increment is ready.

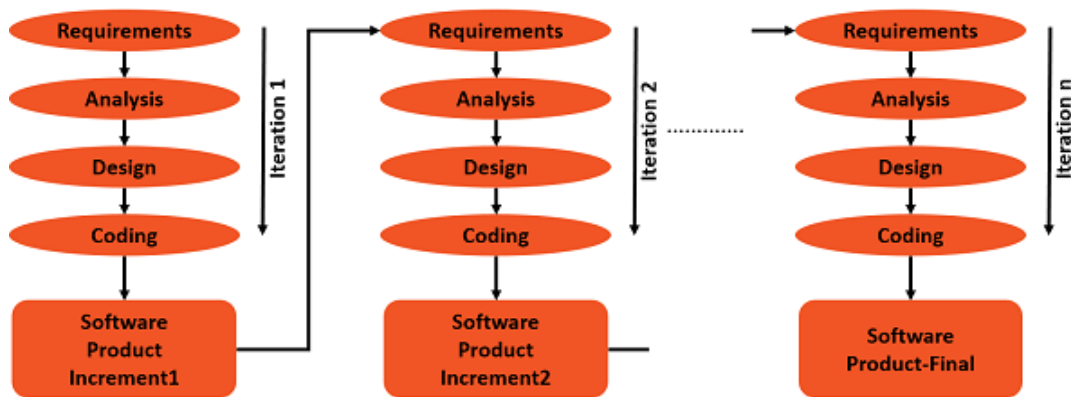


Figure 3.19: Iterative Incremental Model SDLC diagram [5]

3.5.1 Why Iterative Incremental Model?

Following are the reasons we are using Iterative Incremental Model are:

- Some requirements were prioritized.
- Project technology was new for the team.
- Project domain is new to the team.

3.6 Use Cases

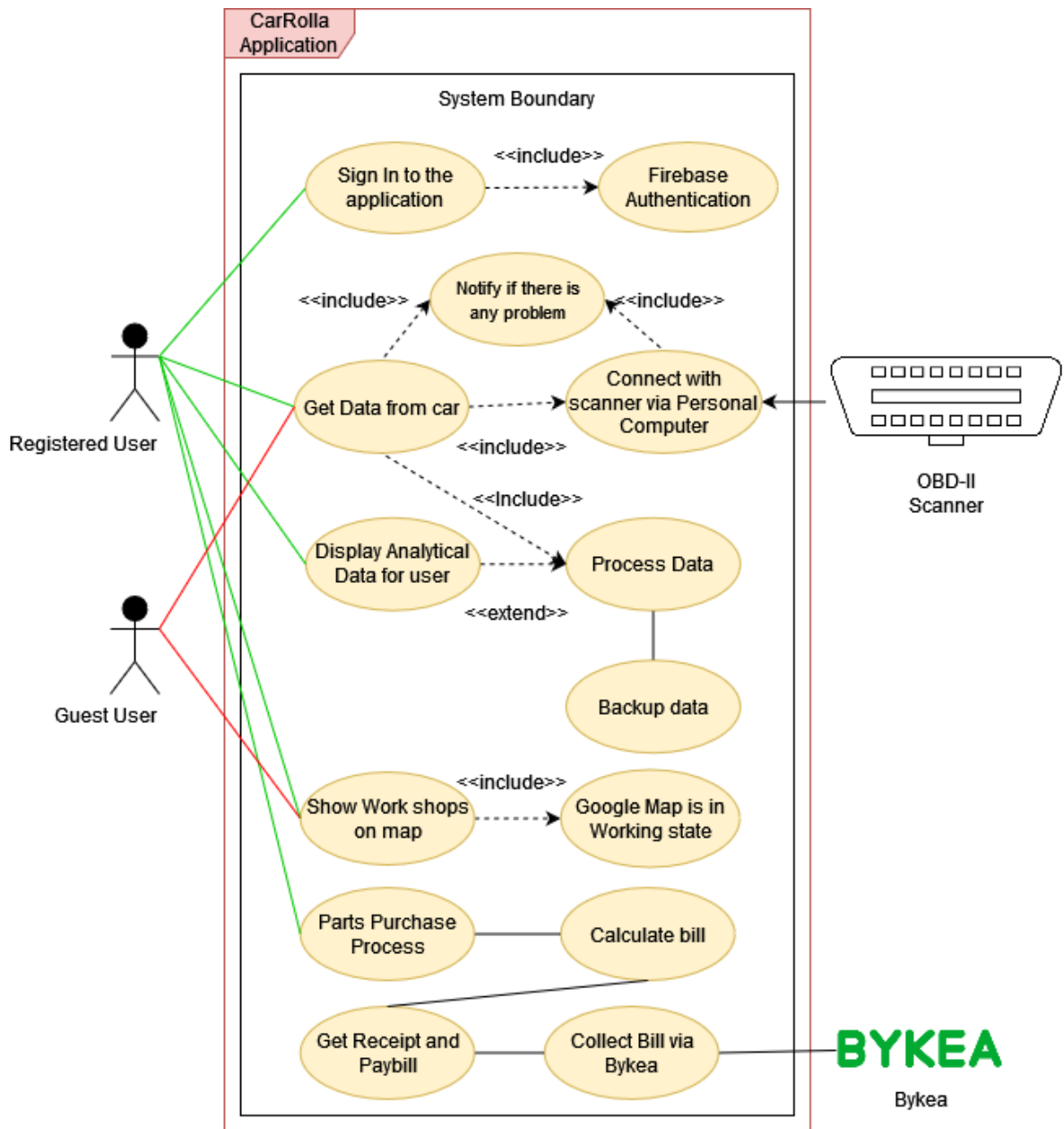


Figure 3.20: Use case diagram

Above Use case diagram shows all the major Use cases of the application. The detailed use cases of each scenario are described below.

Use Case 1

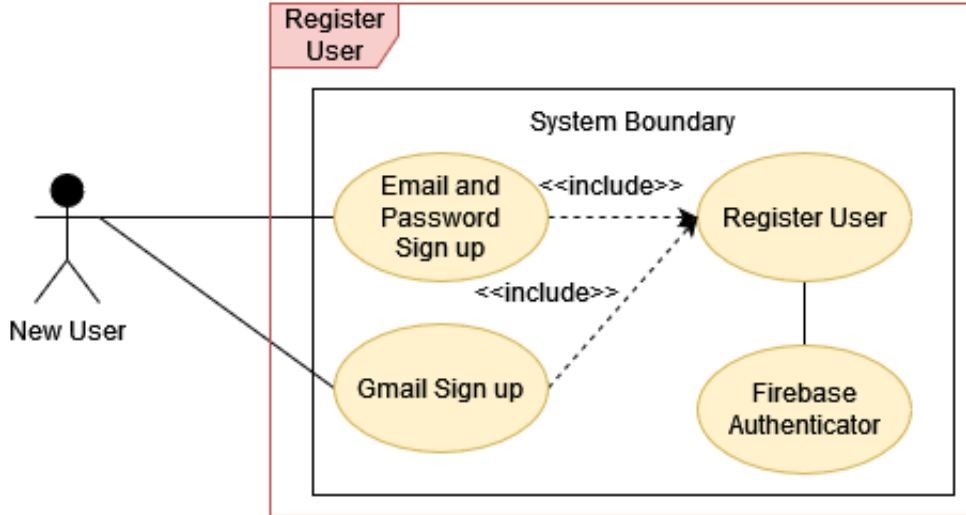


Figure 3.21: UC1: Register User

Table 3.2: UC 1: Register User

Use Case ID	UC1
Title	Register User
Trigger	Application is opened for the first time or user has no previously active two session
Description	This use case depicts the user sign up process for the application.
Primary Actor	New user.
Pre-Condition	The user has an internet connection.
Post Condition	The user is registered for an account.
Basic Flow	The user is registered.
Alternate Flow	<ul style="list-style-type: none"> • If user enter invalid inputs user is asked to enter valid credentials such as email format is not in format as user-name@mail.com or password is less than 6 characters. • If user exists, then user is requested to sign in. • If there is no internet connection user is asked to connect and retry again.

Above use case describes the process of user registration via Google sign in and via email and password.

Use Case 2

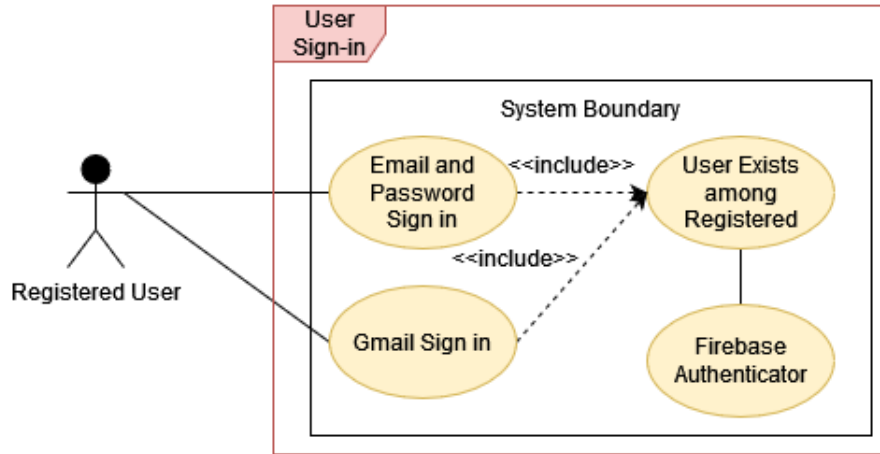


Figure 3.22: UC2: Sign In Case

Table 3.3: UC 2: Sign in Use case

Use Case ID	UC2
Title	Sign In
Trigger	Application is opened.
Description	The user Sign-in to the application.
Primary Actor	Registered user.
Pre-Condition	The user has an internet connection and registered.
Post Condition	The user is signed in to the account.
Basic Flow	The user is authorized.
Alternate Flow	<ul style="list-style-type: none"> • The user is not authorized to continue and is asked to re-enter credentials. • If user enters invalid credential user is asked to register or enter valid credentials. • If there is problem with internet connectivity user is asked to check internet connectivity.

Above use case describes the process of user sign in via Google sign in and via email and password.

Use Case 3

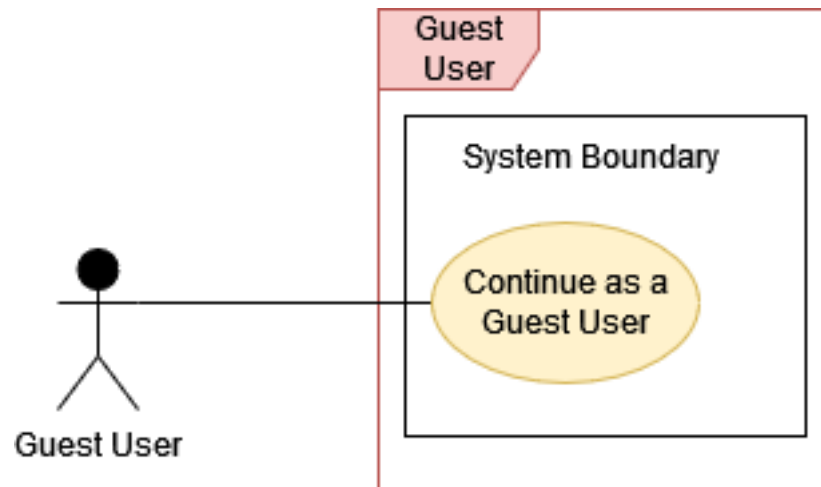


Figure 3.23: UC3: Continue as Guest User

Table 3.4: UC 3: Continuing as a guest user

Use Case ID	UC3
Title	Continue as a guest user.
Trigger	Application is opened and there is no user.
Description	The user will continue without logging in to the application.
Primary Actor	Guest User.
Pre-Condition	The application is installed.
Post Condition	The user can connect to the OBD-II scanner.
Basic Flow	The user can connect to the OBD-II scanner.
Alternate Flow	If application crashes restart application from same state.

Above use case describes that how a user will be able to continuing as a guest user.

Use Case 4

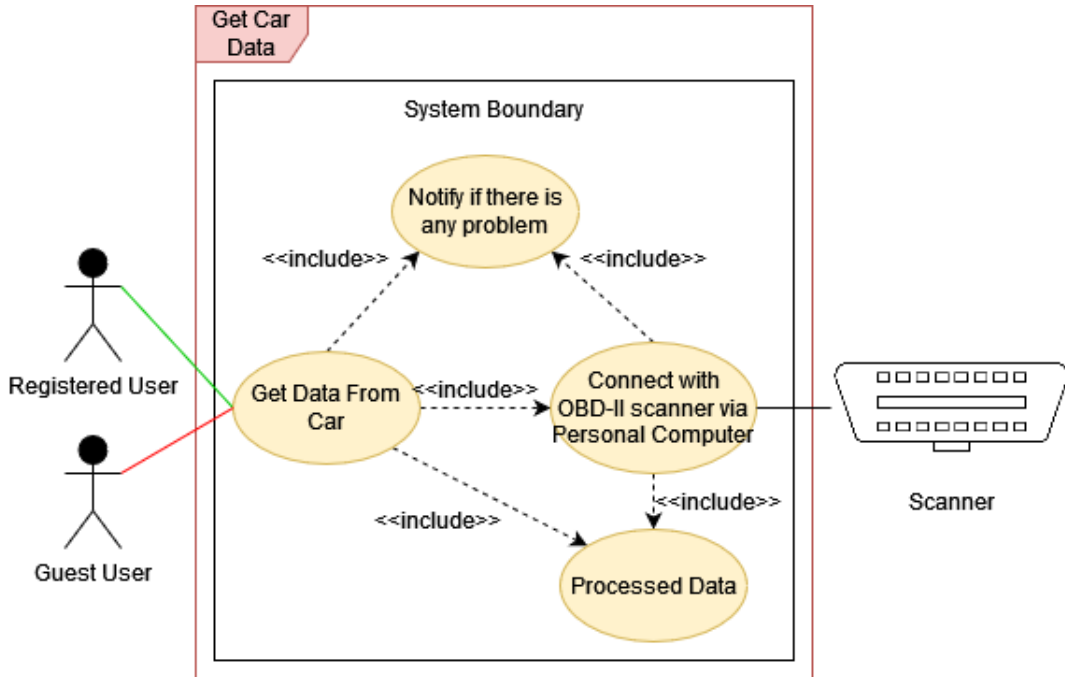


Figure 3.24: UC4: Get Data from Car

Table 3.5: UC 4: Getting Data from Car

Use Case ID	UC4
Title	Get Data from Car.
Trigger	User connects with OBD-II scanner.
Description	The user will pair the OBD-II scanner with the application.
Primary Actor	Guest User and Registered User.
Pre-Condition	OBD-II scanner is plugged into the car’s DLC.
Post Condition	OBD-II scanner is paired with the application and receives vehicle data.
Basic Flow	OBD-II scanner connects with the application and vehicle data is received.
Alternate Flow	<ul style="list-style-type: none"> • Ask the user to re-connect to the scanner if the connection fails. • Data is not loaded and previously saved data is loaded.

Above use case describes that how a data will be taken from the vehicle/car for reading via OBD-II scanner along with all the processing that will be applied.

Use Case 5

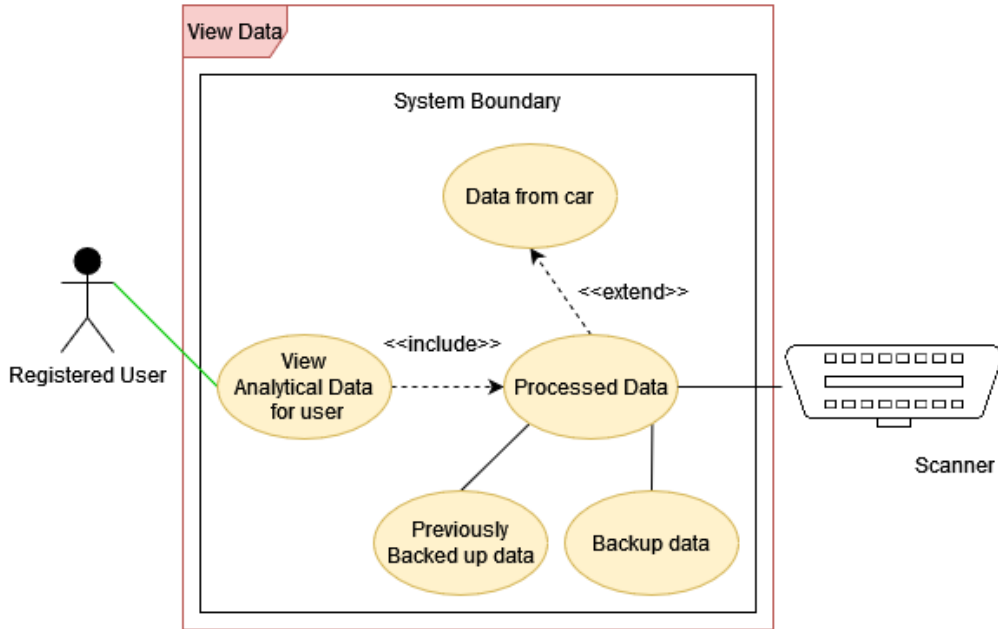


Figure 3.25: UC5: View Analytical Data

Table 3.6: UC 5: Viewing analytical data

Use Case ID	UC5
Title	View Analytical Data.
Trigger	Device connects to the OBD-II scanner or requests data explicitly.
Description	The data received from the OBD-II scanner will be shown to the user in a presentable analytical format.
Primary Actor	Guest User and Registered User.
Pre-Condition	OBD-II is connected to the car and paired with the application.
Post Condition	The user will have the ability to view the analytical data received from the ECU in real-time.
Basic Flow	The user can view the analytical data of the car.
Alternate Flow	<ul style="list-style-type: none"> • The user is notified about the Data Load error and previously stored data is used • The user is requested to connect with the OBD-II scanner.

Above use case describes that the user will be able to view the new data in human friendly modified data.

Use Case 6

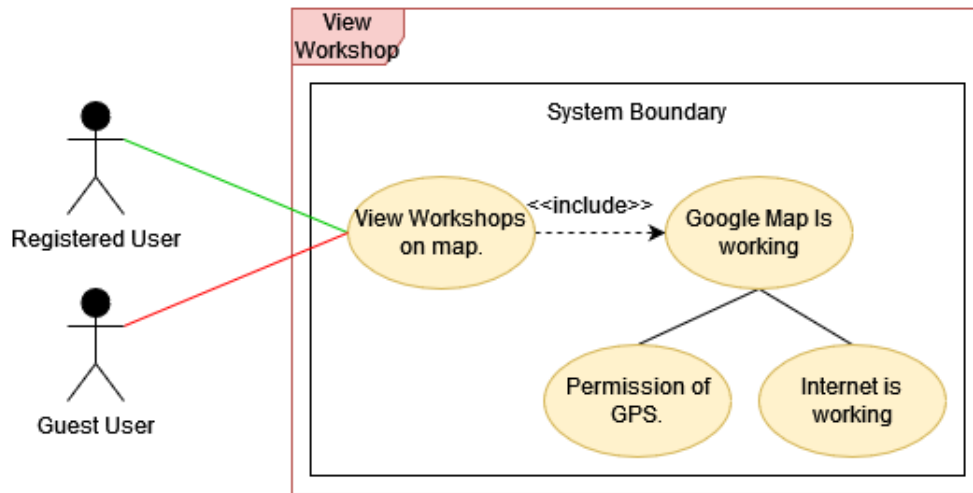


Figure 3.26: UC6: View Workshops in the vicinity

Table 3.7: UC 6: Viewing workshops in the vicinity.

Use Case ID	UC6
Title	View Workshops in the vicinity.
Trigger	User opens workshops availability page.
Description	The user views workshops in the vicinity.
Primary Actor	Guest User and Registered User.
Pre-Condition	The user has internet connectivity.
Post Condition	Users will be able to find the location of the workshops in the area.
Basic Flow	The user successfully views maps with workshop markers.
Alternate Flow	<ul style="list-style-type: none"> • Shows an error in loading map. • If user is not connected to the internet notify them. • If user has not permitted GPS permission is requested.

Above use case describes user’s ability to view workshops in the vicinity.

Use Case 7

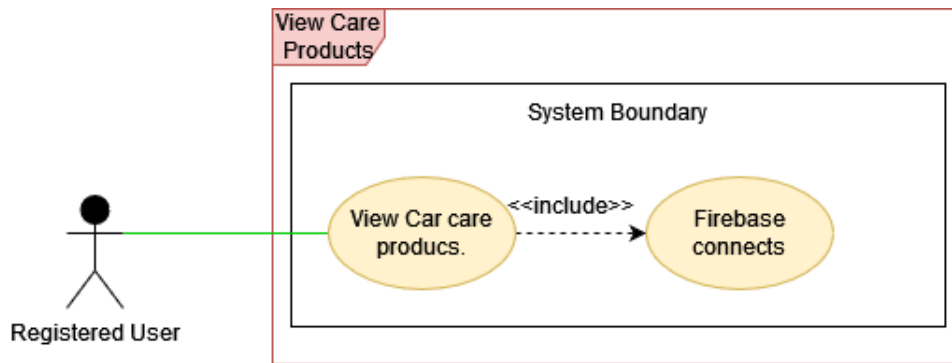


Figure 3.27: UC7: View car care/maintenance products

Table 3.8: UC 7: Viewing car maintenance products

Use Case ID	UC7
Title	View car care/maintenance products.
Trigger	User opens car care/maintenance products.
Description	The User searches for required maintenance or car care products among the ones that are available for the vehicle.
Primary Actor	Registered User.
Pre-Condition	The user is signed in to the application.
Post Condition	Users will see a variety of car care products.
Basic Flow	Variety of products appear in application.
Alternate Flow	<ul style="list-style-type: none"> Shows an error that the server connection failed.

Above use case describes the ability of a user to view car care products.

Use Case 8

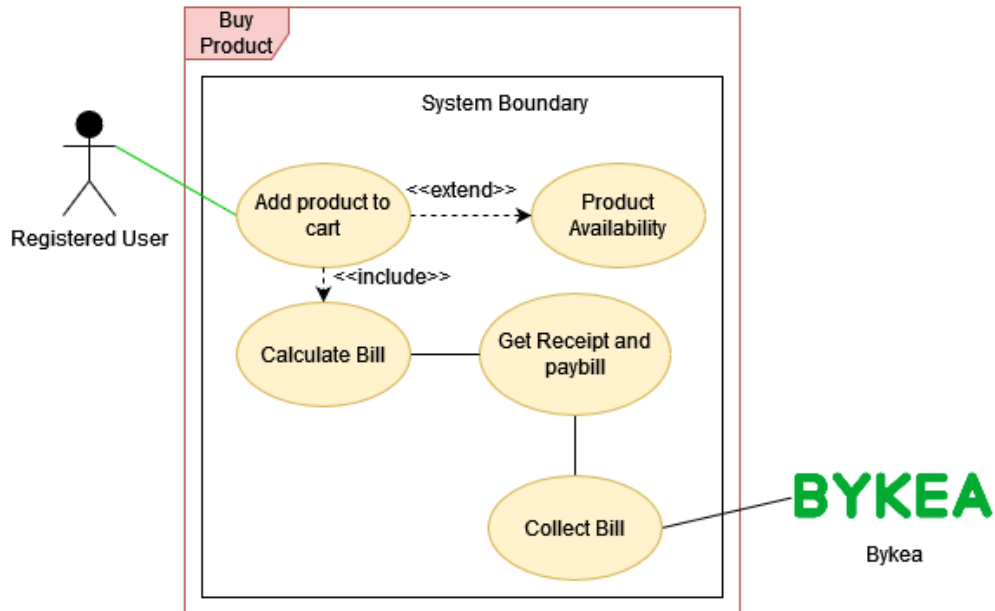


Figure 3.28: UC8: The user adds products to the cart

Table 3.9: UC 8: Adding products to the cart

Use Case ID	UC8
Title	The user adds products to the cart.
Description	The User adds vehicle care products to the cart.
Primary Actor	Registered User.
Pre-Condition	The user has opened the marketplace section.
Post Condition	Users’ selected items will be reflected in the cart.
Basic Flow	Users selected items will be added successfully and reflected in the cart.
Alternate Flow	<ul style="list-style-type: none"> • If the product is out of stock, the user will be informed with an error prompt that the selected product is out of stock. • If connection to database fails user is asked to reconnect.

Above use case describes the product purchasing ability and the process along with billing details.

Chapter 4

Design

4.1 System Architecture

The architecture of our system is described in this chapter. In this chapter, we will discuss in detail, the overall design of the application, High-level design, implementation constraints, database design, GUI design, and our assumptions regarding the system.

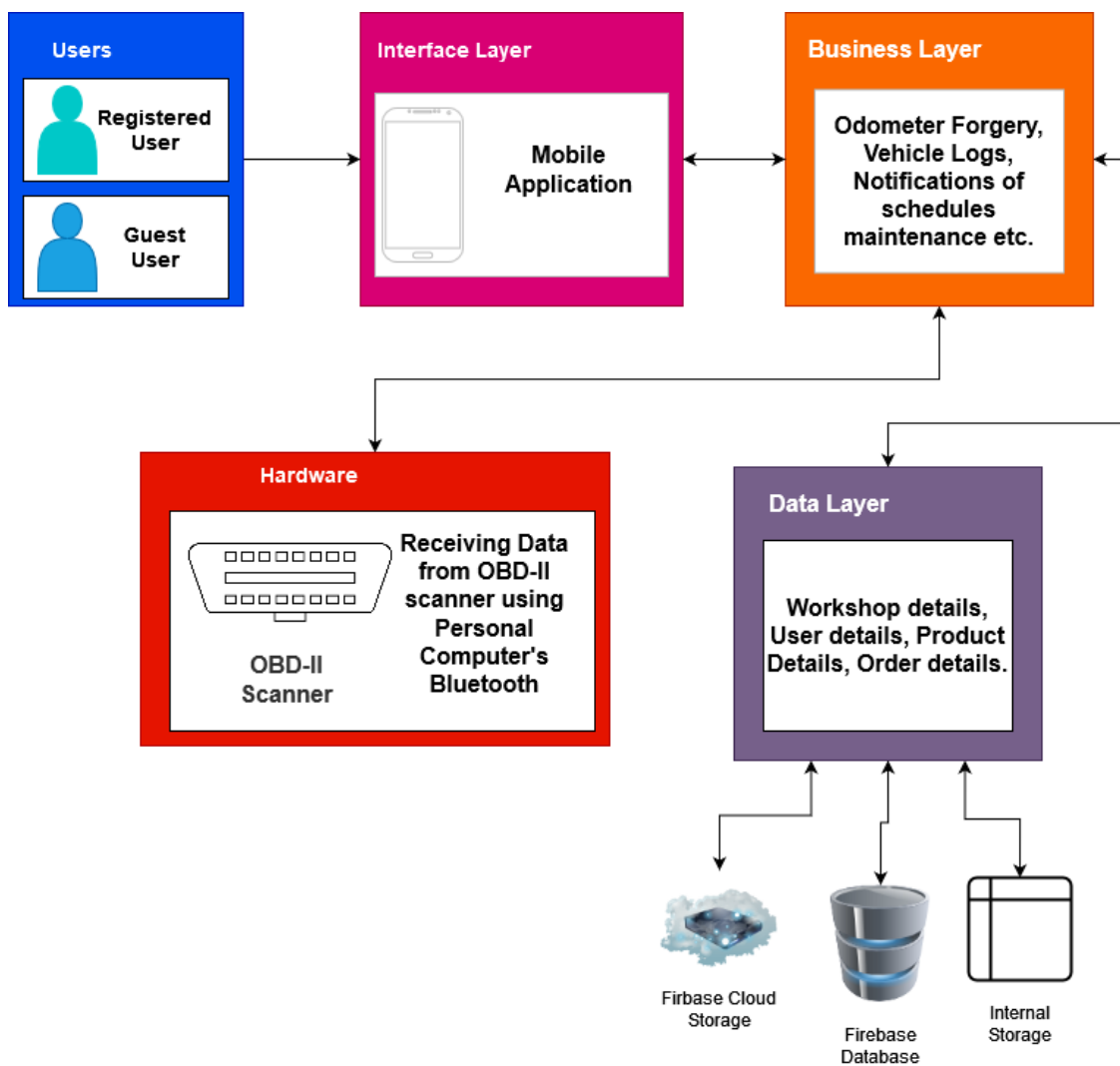


Figure 4.1: System Architecture diagram

The above diagram is the System Architecture of our project. It shows that we are using the three-tier architecture of the application where the interface is a Mobile application having all the business/logic layer that handles everything and interact with the OBD-II Scanner and the third tier is the data layer that stores data and accesses it in multiple ways such as local/internal storage, firebase cloud, and Firebase database.

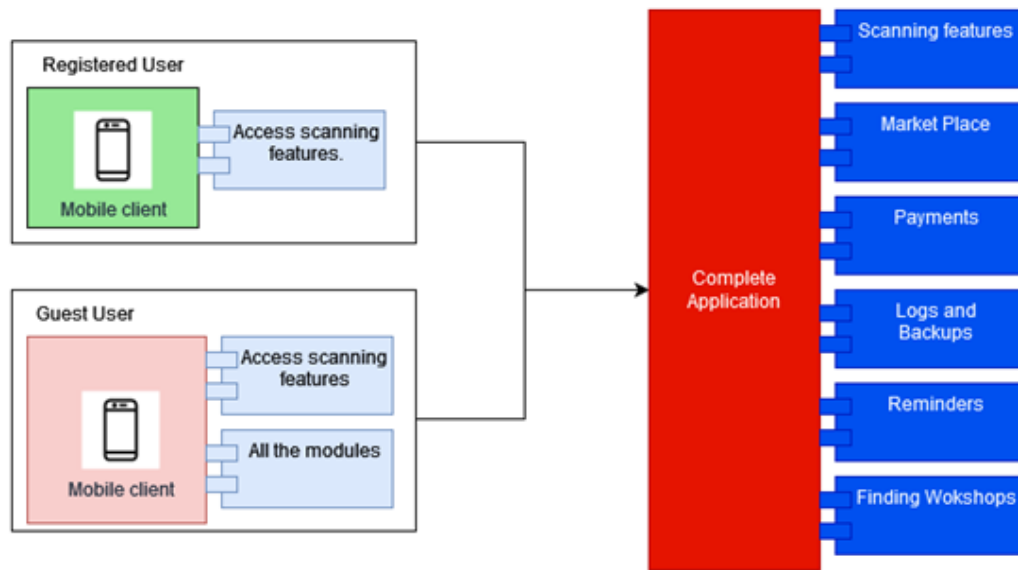


Figure 4.2: Application Architecture Diagram

The above diagram shows the architecture of our application where we allow the user to select have two modes of using our application: Registered user Registered users will have access to all the available features discussed earlier. Guest user Guest users have access to use features that are linked to scanning only such as scanning fault codes, seeing description of the found fault codes, and finding nearby workshops.

The application has plenty of features Scanning vehicle information, Purchasing vehicle care products from Marketplace and Paying for the products via Bykea cash. Furthermore, vehicle logs are backed up and timely reminders regarding the maintenance are sent. Also user can find workshops in his vicinity using Google Map.

4.2 High-Level Design

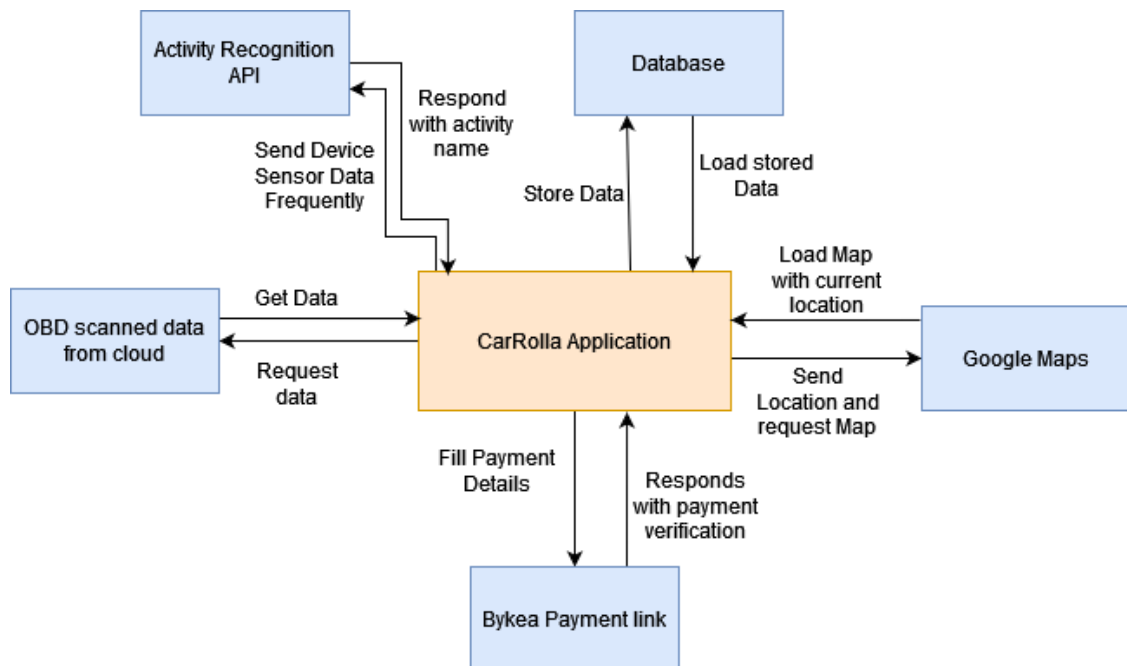


Figure 4.3: High Level Design

High-Level design shows the details of the application and how it will interact with the different dependencies. Moving step by step for each of these dependencies.

- **OBD Scanner**

When we connect with the OBD scanner, the application requests vehicle data. In response to the request we get data of the vehicle. The data that comes in response is as follows.

- **Activity Recognition Package**

Activity Recognition Package of Flutter Application checks the state of the user by taking data from device sensors and checking for user activity such as Cycling, Driving, Walking, Running. In our case we only recognize 'Driving' activity to ask whether the user is driving or not. [18]

- **Database**

Database stores different data regarding vehicle, vehicle logs, market place and orders. We can store, update or request the required data.

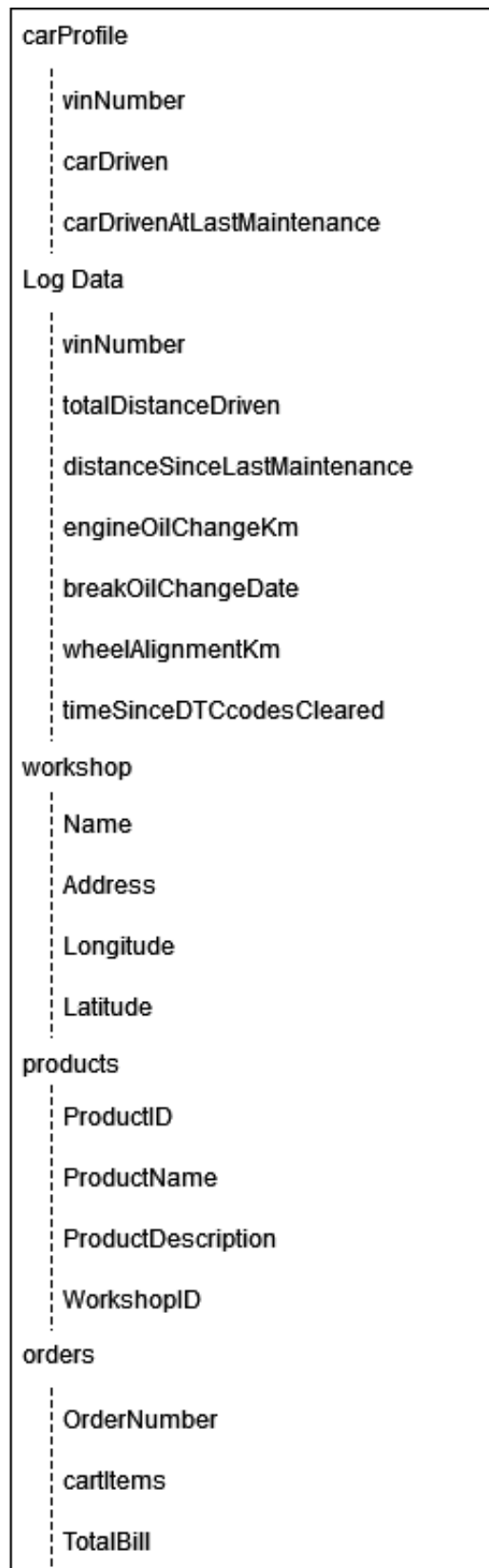


Figure 4.4: Database Design Diagram

The above database design shows a Firebase tree for the database. It shows what objects will be stored in the database and what will be its attributes that are stored.

- **Bykea Payment**

Bykea Payment works via a link. We only redirect users to the specified link (<https://bykea.cash/carrolla>) from our application where the user can perform the payments.



Figure 4.5: Bykea Logo

- **Google Maps**

In this project Google Maps API is used for finding workshops in the vicinity. As any vehicle may need some repairs any time while driving. To show workshops and user location their Longitudes and Latitudes are used. Below figure shows the workshops location in our application.

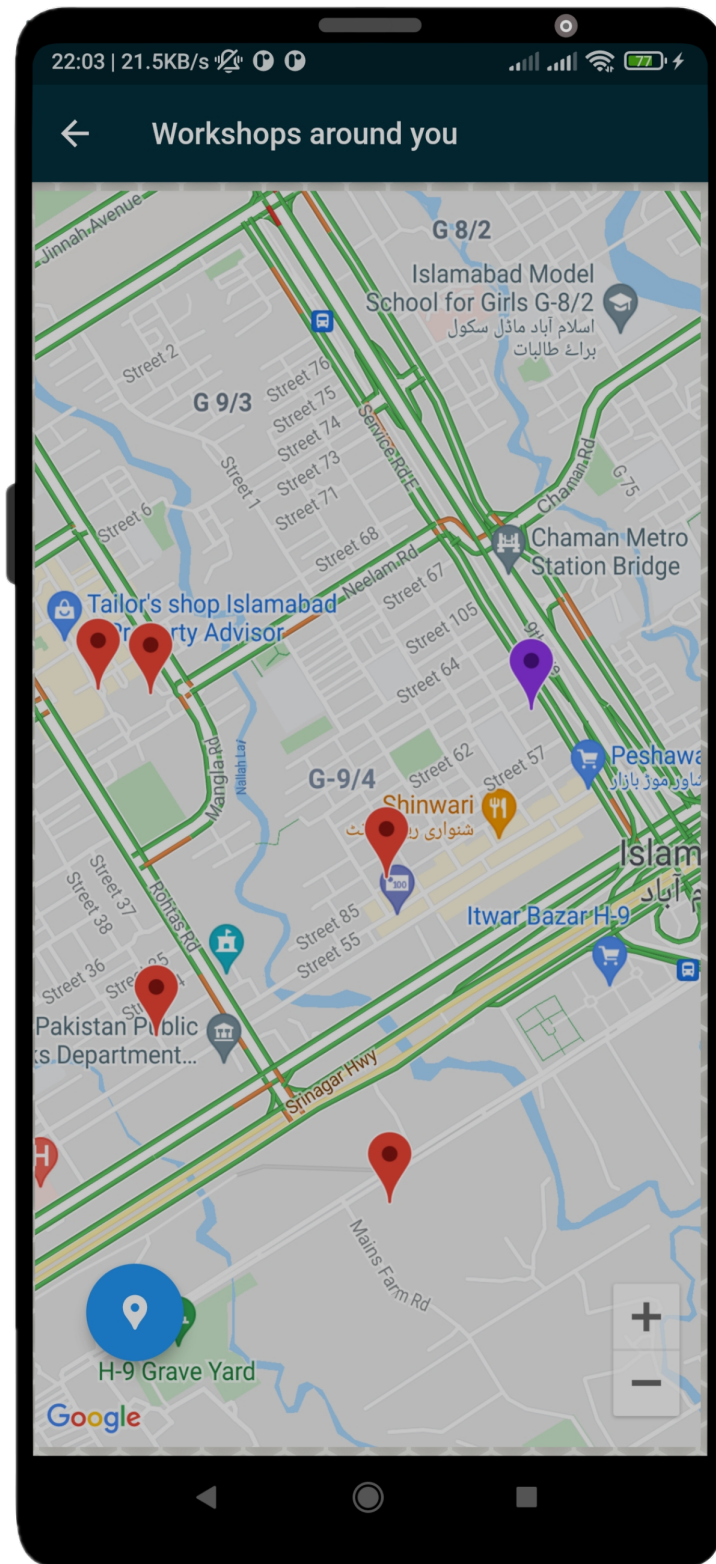


Figure 4.6: Workshops on map

4.3 Sequence Diagram for Registered User

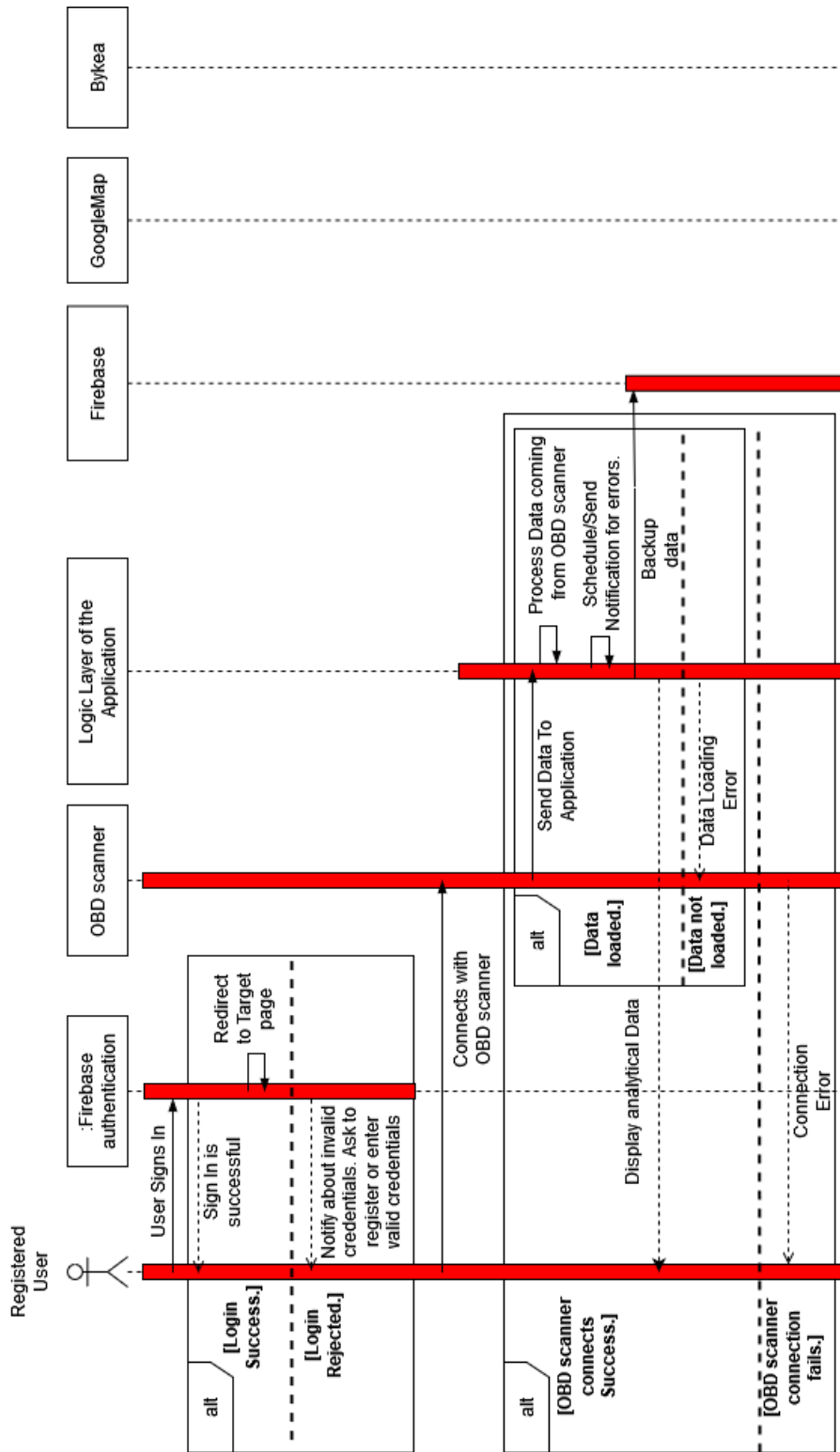


Figure 4.7: Sequence Diagram for Registered User part 1

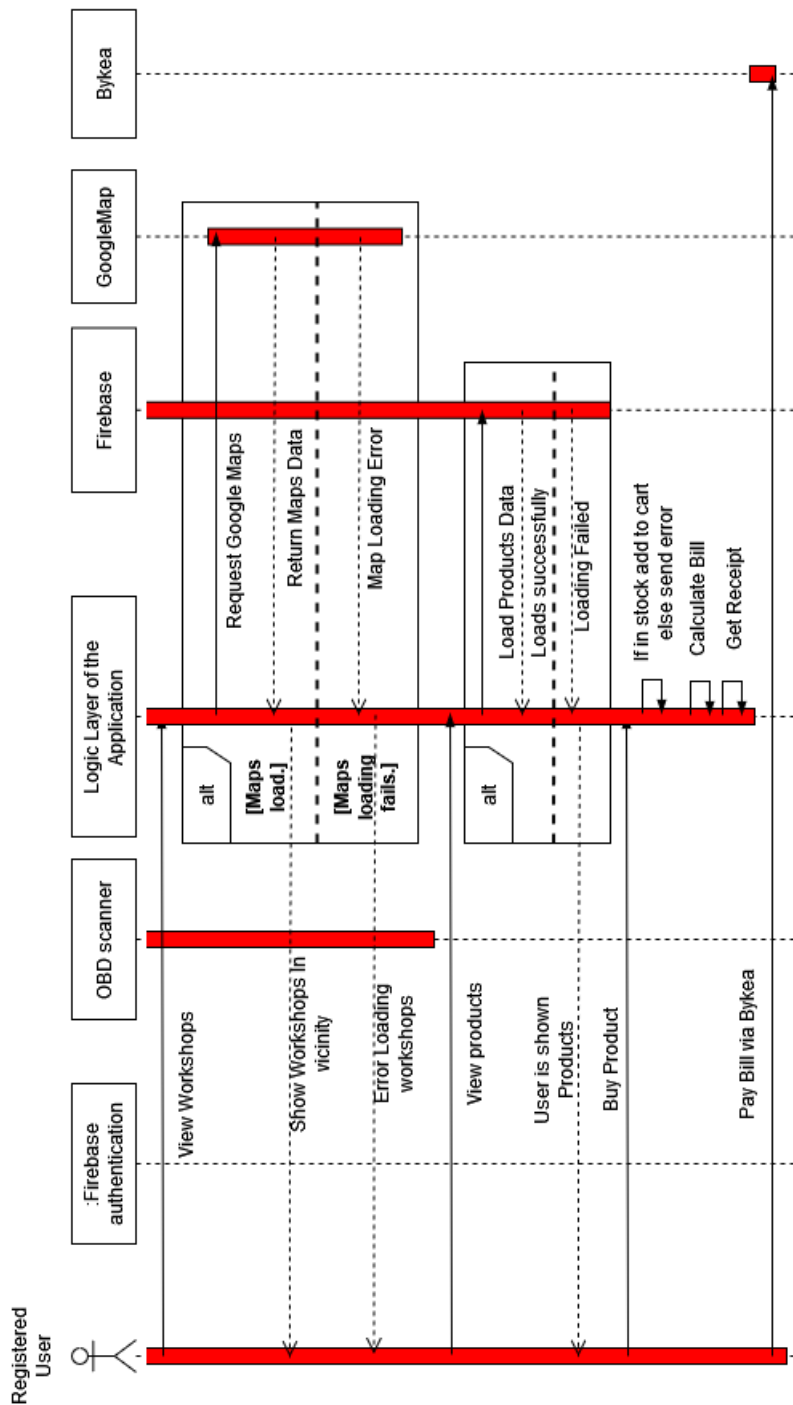


Figure 4.8: Sequence Diagram for Registered User part 2

Above Sequence diagram for registered user shows the flow of events in our project from the given condition that user is a registered user using the application. It shows all the processes for different tasks from start to end for the registered user.

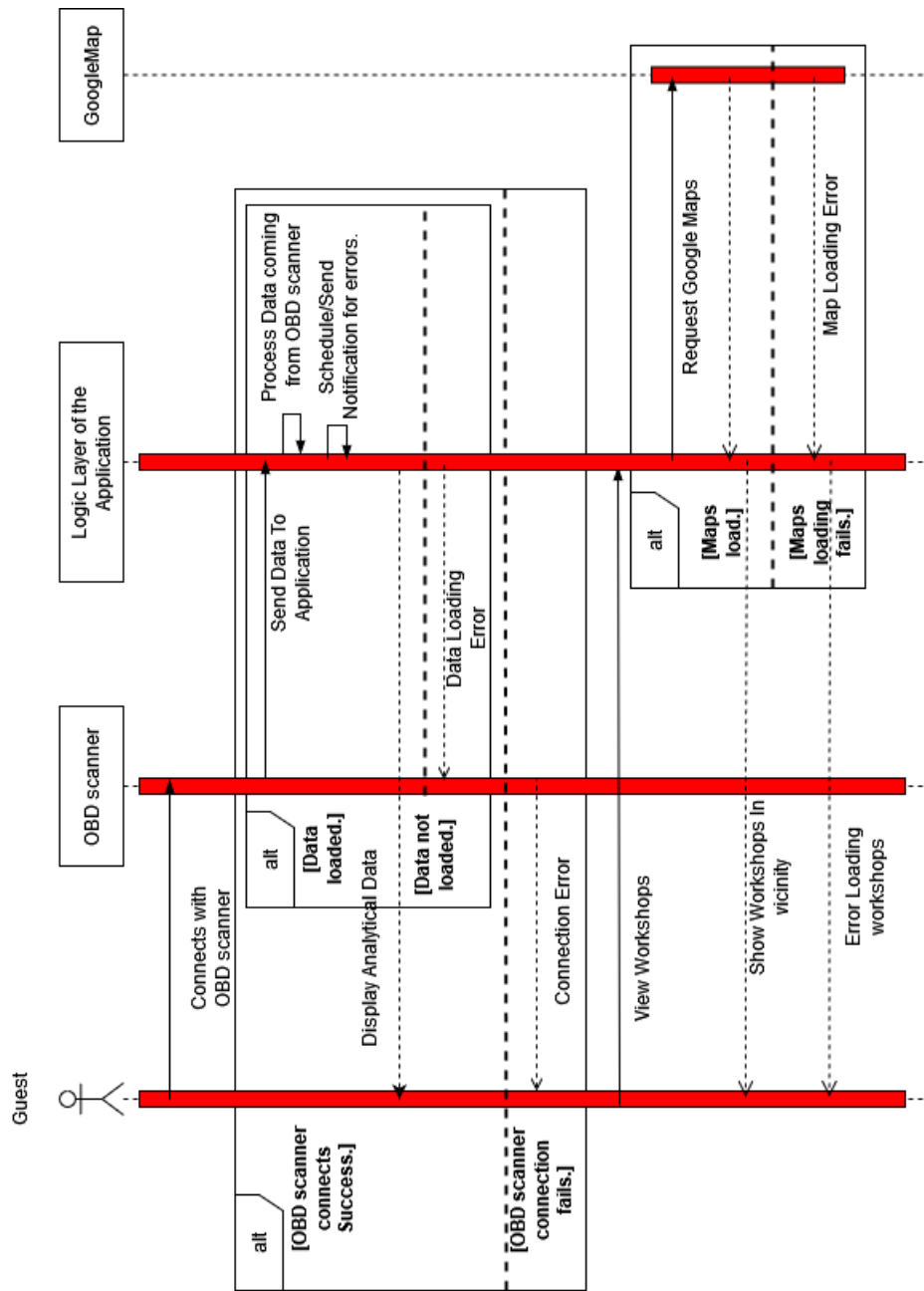


Figure 4.9: Sequence Diagram for Guest User

This Sequence diagram for guest user shows the flow of events in our project from the given condition the user is a guest user. It show all different tasks from start to end for the guest user.

4.4 Activity Diagram

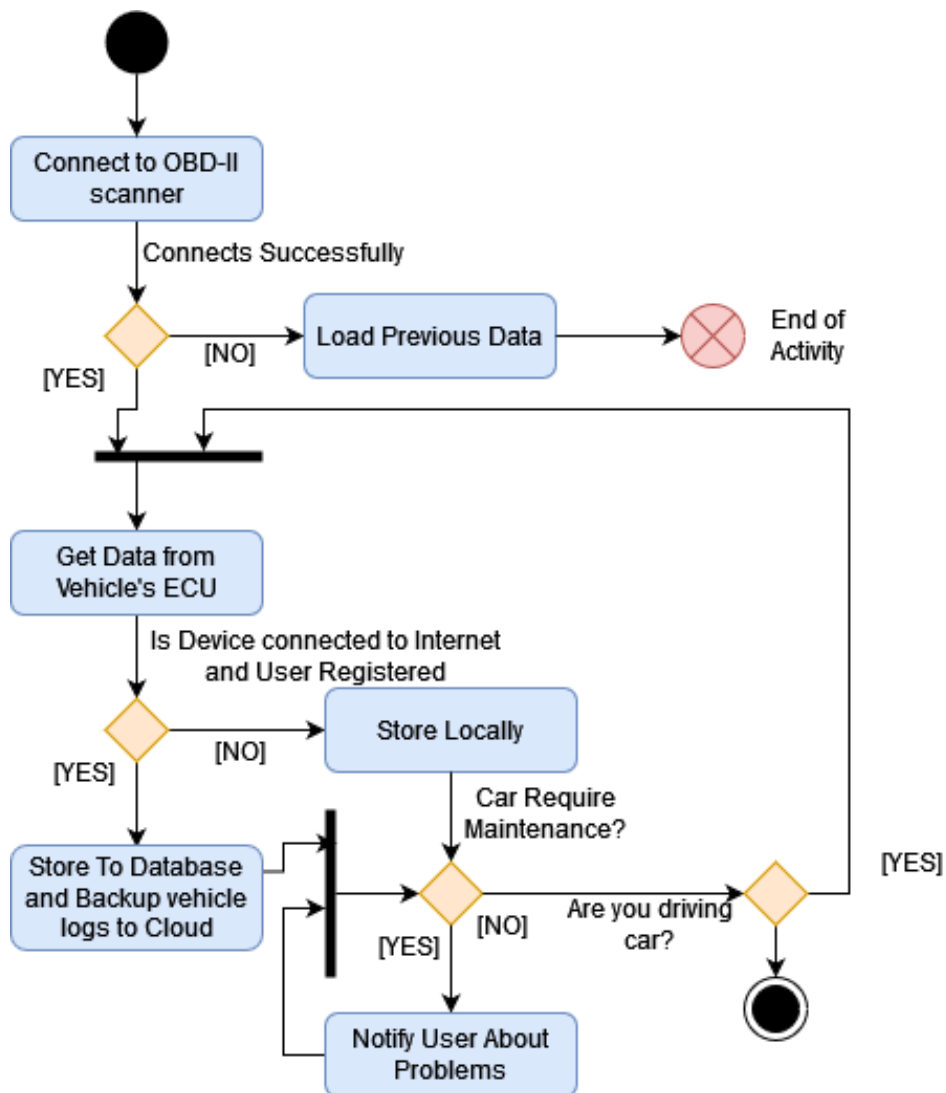


Figure 4.10: System behaviour Activity Diagram

The activity diagram shown in Figure 4.3 depicts the behavior of our system. It portrays the control flow from start to end showing all the decisions and steps.

4.5 Design and Implementation Constraints

Our system is designed to work keeping in consideration the following constraints and assumptions.

- **Time Constraint**

The time taken for the application to launch depends mostly on the processing speed of the device the application is installed on. So it can vary from device to device but the application is not very heavy duty so it will perform well on most devices.

- **Hardware Constraint**

The designed application may have inadequate performance if the device it is being used on is very low-end, although most modern low-end and high-end devices have more than enough capabilities to run this application, still it can be a rare issue. Secondly, the OBD-II scanner connected to the car can be faulty or can get short-circuited if a good quality product is not purchased.

- **Deployment Constraint**

The application can easily be installed and paired with the plugged-in OBD-II scanner. It is designed for easy installation and expected not to cause any problems.

- **Platform Constraint**

The application is designed to be running on mobile devices having Android OS. The Android operating systems comprise more than 72.84% of the smartphone market as it is mentioned in the introduction. The unavailability of of MacOS cross platform application cannot be developed in Flutter. So, the application can not run on iOS and Flutter does not support other mobile operating systems such as Blackberry OS, Windows OS, Harmony OS, etc.

- **Cloud Backup Constraint**

Cloud backup of data is only possible if the mobile phone on which the application is installed has internet connectivity most of the time. In case of no internet connectivity, the backup will be delayed.

- **Assumptions**

The application will perform better with a high-speed Bluetooth version such as Bluetooth v4.0. The OBD-II scanner used in out features Bluetooth v3.0, which is not a constraint as the requirement of the application does not require long range connectivity. The backup feature will be done frequently and the internet connectivity will be available at least once every day. The driver will not be visually impaired.

Chapter 5

System Implementation

In this section, we discuss the processes and tools we have used in order to bring our idea to life ensuring all application quality checks.

5.1 Working

The basic working of our application is as follows:

- This is an Android application working with an OBD-II scanner to achieve its main objective.
- We ask the user to sign in or continue without signing in.
- Guest and Registered users can scan the data of their car to get fault codes from the vehicle via cloud where data is uploaded by the laptop by connecting to the OBD-II scanner.
- Registered User's maintenance logs are stored.
- Registered Users can purchase products for their vehicles.
- Registered Users can see odometer forgery status.
- Guest as well as registered user can view workshops in vicinity.

5.2 Tools and Technologies

5.2.1 Tools

In this section, we discuss Components we used for development. To develop our application we have used 3 components.

- **Flutter SDK**

The system uses the android mobile application and to develop it, Flutter framework of Dart is used. The Flutter SDK provides the setup for the development of android mobile applications. It also supports cross-platform application development. Moreover, it already contains Dart SDK. Flutter SDK version 2.8.0 is used.

- **Android Studio for Android SDK and Emulator To Test Application**

For the development process here Android Studio is used just for getting Android SDK that is required for Android app development. It is also for Android Emulator but in our development process, we used physical devices for testing purposes. Android SDK version 31.1.0 and Android Studio Version 4.1.2 is used

- **Visual Studio Code as a Code Editor**

Visual studio code is used as code editor for flutter development as well as writing Python script. Visual Studio comes with plenty of code editing features such as intellisense, code snippets, and code refactoring tools. The main reason of using Visual Studio code is being light and efficient.

- **Ancel iCar 2 Bluetooth OBD-II Scanner**

For getting data from the vehicle we use Ancel iCar 2 Bluetooth OBD-II scanner which is responsible for sending data to the mobile application via Bluetooth.

- **Python**



Figure 5.1: Python Logo

The system uses Python for reading data through the OBD-II scanner. Here Python is used as back-end of our application's functionality. The use of Python was done because currently Flutter framework is very new and do not support most of the back-end tasks. Whereas Python has lot of support for performing such extensive tasks. Therefore, we used Python as a back-end of our application. Python version 3 is used.

5.2.2 Packages Used

In this section, we will be describing all the libraries used to produce multiple processes in application development.

1. Flutter Packages

- **Email_Validator**

Package `email_validator` version 2.0.1 is used for validating user email format in case of sign-in or sign-up via Email and Password. [20]

- **Flutter_Webview_Plugin**

Package `flutter_webview_plugin` version 0.4.0 is used for displaying the webpage for Bykea payments. The payments are used for billing of Purchased parts. [21]

- **UUID**

Package `UUID` (Universally Unique Identifier) version 3.0.5 is used for managing unique numbers in situations such as order numbers, and product IDs used in the section of purchasing car products. [22]

- **Google_Sign_In and Firebase_Auth**

We are using Firebase authentication service for Google Sign In and Sign in via Email and Password. To do so, we use two flutter packages `google_sign_in` and `firebase_auth` having versions 5.2.0 and 3.1.0 respectively. [23] [24]

- **Google_Maps_Flutter**

We are using Google Maps to display workshops. To use maps api in flutter we use flutter package authentication service for Google Sign In and Sign in via Email and Password. To do so, we use two flutter packages `google_map_flutter` having versions 2.1.0. [25]

- **Geolocator**

We are using Google Maps to display workshops. For displaying workshops in vicinity we need user's current location to do so, we use `geolocator` package versions 7.7.1. [26]

2. Python Packages

- **OBD**

We are using OBD library to access the data coming from the OBD-II scanner via Bluetooth. It helps us with accessing data from Engine Control Unit. We use **OBD** version 0.7.1. [27]

- **Time**

We are using **time** library for different time related functionalities. Mainly getting repeated data from Car every now and then. [28]

- **MSVCRT**

This library is used for Windows special services. It provides access to useful capabilities of the Windows platform. [29]

5.2.3 API used



Figure 5.2: Google Maps Logo

To navigate users to the automobile workshop in the vicinity we used Google Maps API. [30] To do so we take the current user location and display the authorized workshops in the nearby area.

5.3 Google Firebase



Figure 5.3: Google Firebase Logo

Firebase is used for three purposes in this project.

5.3.1 Firebase Authentication

Firebase authentication is a service provided by Google for providing automated user account management service. In our project, we use two authentication methods: Google authentication and Email and Password-based authentication.

5.3.2 Firebase Database

Firebase's database is used for storing collections of Data. In this application, we used it for storing records of the User's vehicles, details of vehicles, and vehicle maintenance logs.

5.3.3 Firebase Cloud Backup

Firebase cloud backup allows an application to backup Files such as Media files, document files. In our project, we use Firebase Cloud Backup for storing maintenance logs of the vehicle.

Chapter 6

System Testing and Evaluation

To check the capabilities and working of the designed application system testing is a very important part of software development. The testing feedback was taken from 10 users. So, different methods are used to test or evaluate the system that is discussed in detail in this chapter.

6.1 Testing Techniques

To perform testing of our system for different tasks and objectives we have applied the following testing techniques.

- Functional testing
- Graphical User Interface (GUI) testing
- Compatibility testing

6.1.1 Functional Testing

In the Functional testing, we test our system's working with respect to the Functional requirements that are discussed in Section 3.4.1.

- **Unit Testing**

In unit testing, we individually test each and every module of the system. This testing technique helps us figure out bugs and solve them easily. As a result, every module is tested and brought into working order. In our application, different modules we tested are:

1. User Registration
2. User sign-in
3. User can continue as Guest User
4. Verify OBD scanner Connection
5. Verify data retrieval via OBD scanner
6. Workshops appear on Google map
7. Verify working of the product marketplace
8. Verify Database integration and functionality
9. Verify cloud backup integrated
10. Verify driver test state is working

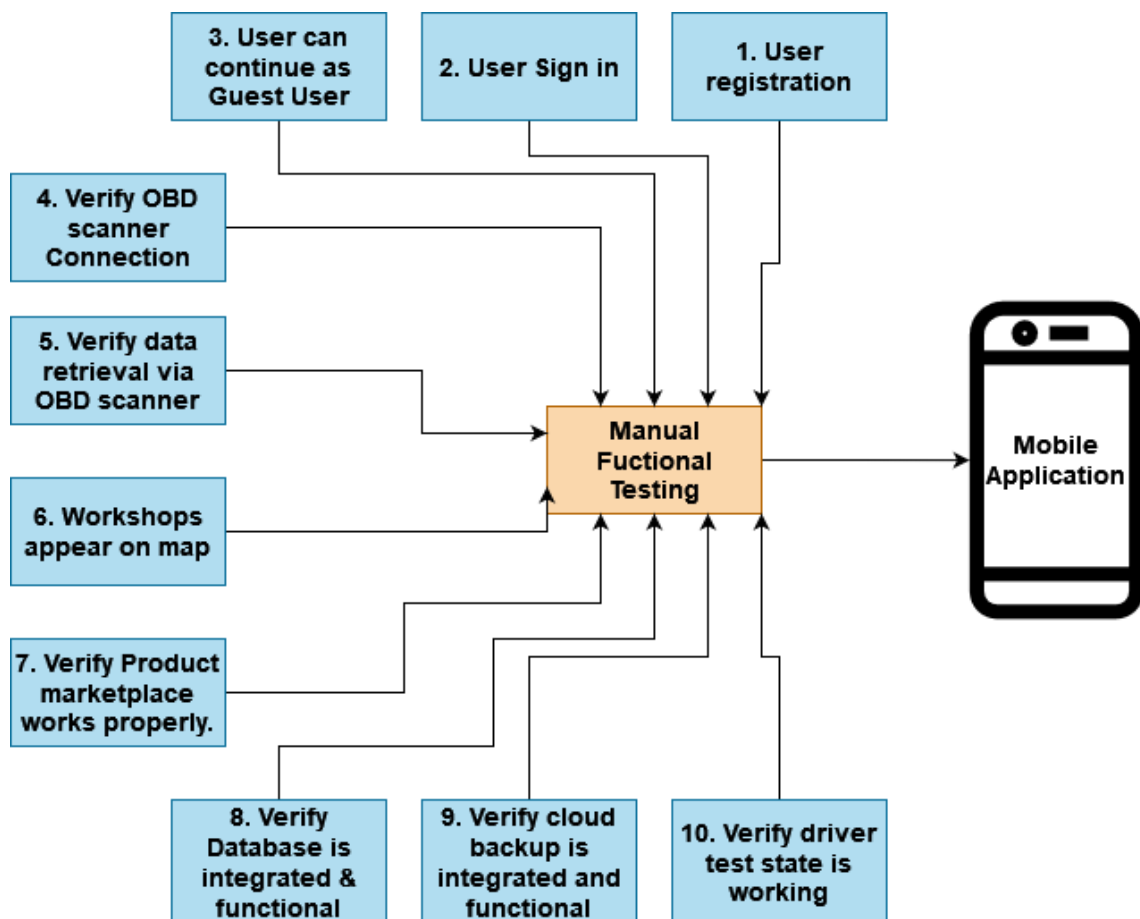


Figure 6.1: Manual Functional Testing

- **Integration Testing**

Integration testing is performed once various modules are integrated. This testing helps in figuring out whether our application works correctly once different modules

are integrated. Initially, we performed unit testing on the user sign-in/sign-up module and Google maps module and then tested them after integration to check their working after integration and in the same way other modules were tested.

- **Sytem Testing**

System testing is done once the whole system is ready. So, after the completion of our whole system, we performed system testing to check the working of our system after the integration of all the modules.

6.1.2 Graphical User Interface Testing

In Graphical User Interface (GUI) testing we checked the overall interface of the application keeping the target user in mind. We checked screen designs, controls like labels, buttons, icons, and different data displaying widgets used. We also checked the size and color of different widgets used in UI design. The design is easy to remember and easy to use.

6.1.3 Compatibility Testing

Compatibility testing application is a test performed across different versions of supported platforms. As our application is Android-based, we tested it on different versions of Android on specific available devices. As per our testing, the application is compatible with the Android versions: 11,10 and 9.

6.2 Test Cases

6.2.1 Register User

Table 6.1: TC1: Sign Up

Test Case ID	TC1
Test Unit Title	To verify the user can sign up.
Description	Users can sign up via email password as well as google.
Test Setup	Internet availability.
Expected Result	The user is successfully registered.
Actual Result	Pass

This test case is to verify user registration success using account registration via Google or any email address or password. For registration we are using Google Firebase authentication. The success of this test depend upon the availability of the internet.

6.2.2 Sign In User

Table 6.2: TC 2: Sign In

Test Case ID	TC2
Test Unit Title	To verify the user can sign in.
Description	Users can sign in via email password as well as google.
Test Setup	Internet availability and the user is registered.
Expected Result	The user is successfully signed in.
Actual Result	Pass

This test case is to verify user sign-in success using account sign-in via Google or any email address or password. For sign-in we are using Google Firebase authentication. The success of this test depend upon the availability of the internet.

6.2.3 Guest User

Table 6.3: TC 3: Guest User

Test Case ID	TC3
Test Unit Title	To verify the user can proceed without an account
Description	Users can continue without being registered or signed in
Test Setup	Application is installed and internet is available
Expected Result	User lands on the main page
Actual Result	Pass

This test case is to verify user can continue as a Guest User. In this test case we verified if all the functionality for guest user is in correct working. The success of this test depend upon the availability of the internet.

6.2.4 Get Car Data

Table 6.4: TC 4: Get Car Data

Test Case ID	TC4
Test Unit Title	To verify Application connects with the OBD-II scanner and Data is retrieved.
Description	Application connects with the OBD-II scanner via Bluetooth and data is received
Test Setup	App installation and Bluetooth OBD-II scanner being connected to the diagnostic link connector (DLC).
Expected Result	Application connects with the OBD-II scanner and data is retrieved.
Actual Result	Pass

This test case is to verify that the data from Car's Electronic Circuit Unit (ECU) is retrieved successfully. The success of this test depend upon the availability of the internet and correctly working Bluetooth of laptop and mobile device's internet.

6.2.5 View Car Data

Table 6.5: TC 5: Viewing analytical data

Test Case ID	TC5
Test Unit Title	Viewing analytical data.
Description	View retrieved vehicle data and fault codes in processed state.
Test Setup	The OBD-II is connected and the device has internet connectivity.
Expected Result	Vehicle Data received through the OBD-II scanner is analysed and displayed.
Actual Result	Pass

This test case is to verify that the car data in processed form is displayed and is understandable for the user. In this test case we verified if all the functionality for processing of the car data as well as its representation is readable and understandable. The success of this test depend upon the availability of the internet as we load the new data from cloud.

6.2.6 View workshops

Table 6.6: TC 6: Testing if workshops appear on Maps

Test Case ID	TC6
Test Unit Title	View Workshops in the vicinity.
Description	Workshops in the vicinity appear on the Google Maps.
Test Setup	Internet availability.
Expected Result	Workshop markers appear for Workshops in the vicinity.
Actual Result	Pass

This test case is to verify that workshops are visible on the google map. In this test case we verified if all the markers, workshops locations and users current location are visible. To do so we use Google Maps and Geo location application. The success of this test depend upon the availability of the internet and location access.

6.2.7 The User can View Products

Table 6.7: TC 7: Testing if a user can successfully buy products from the application

Test Case ID	TC7
Test Unit Title	To verify that a user can view products.
Description	The user can reach the market place and view all the products.
Test Setup	Internet availability
Expected Result	The products loads and images, product name, description and price is visible.
Actual Result	Pass

This test case is to verify user can view the products in marketplace and all its attributes such as image, price, product name and description are visible. The success of this test is dependent on the availability of the internet as all the data of product is in database.

6.2.8 The User can Buy Products

Table 6.8: TC 8: Testing if a user can successfully buy products from the application

Test Case ID	TC8
Test Unit Title	To verify that a user can buy products.
Description	The user can reach the checkout page and the cart works properly.
Test Setup	Internet availability
Expected Result	The user proceed to checkout and pay.
Actual Result	Pass

This test case is to verify user can purchase products and proceed to checkout and pay for the total calculated bill. This test case primarily validates all the checks and logical steps for the calculations and product purchase. To complete the process internet is required as data is to be uploaded to the database.

6.2.9 Database is Integrated

Table 6.9: TC 9: Testing database integration.

Test Case ID	TC9
Test Unit Title	To verify the database is integrated.
Description	Check that data flow in to and from the database is working.
Test Setup	Internet availability.
Expected Result	Data is stored and loaded from Database.
Actual Result	Pass

This test case is to verify whether database is properly integrated and works. For completion of this test internet availability is necessary.

6.2.10 Cloud Backup is integrated

Table 6.10: TC 10: Testing if cloud back up is integrated successfully

Test Case ID	TC10
Test Unit Title	To verify the cloud backup is integrated.
Description	User-specific vehicle updated files are backed up to the cloud.
Test Setup	Internet availability.
Expected Result	User files upload to the cloud.
Actual Result	Pass

This test case is to verify whether Cloud Backup is properly integrated and works. For completion of this test internet availability is necessary.

6.2.11 Driver State Testing is Working

Table 6.11: TC 11: To test if driver state is correctly recognised

Test Case ID	TC11
Test Unit Title	To verify driver state testing is working.
Description	The user app shows an alert dialog once device sensors detect driving activity.
Test Setup	Application is open
Expected Result	The user sees an alert dialog.
Actual Result	Pass

This test case is to check the working of Motion Detection package whenever it detects driving activity. On the detection we ask driver to verify if the driver is driving their car. For completion of this test internet availability is necessary.

6.2.12 Application Installation

Table 6.12: Test Case 12: Application Installation

Test Case ID	TC12
Test Unit Title	To verify the app is installed successfully.
Description	The application installs from the APK file. The user downloads and installs the APK file from the provided link of Google Drive.
Test Setup	The Mobile device must be running Android OS and have internet access.
Expected Result	The application is successfully installed.
Actual Result	Pass

This test case is to verify that the mobile application is installed. For completion of this test the mobile device must be running on Android Operating System, and internet availability is necessary.

6.2.13 Application Launch

Table 6.13: Test Case 13: Application Launch

Test Case ID	TC13
Test Unit Title	To verify the app launches successfully.
Description	The app does not crash before opening.
Test Setup	The device must be running Android and the app is installed.
Expected Result	The application is launched successfully.
Actual Result	Pass

This test case is to verify whether application launches or not on launching application after installation. For completion of this test the mobile device must be running on Android Operating System, and internet availability is necessary.

Chapter 7

Conclusions

7.1 Conclusions

In the scope of our project, we aimed to develop an app that helps Vehicle owners keep track of their vehicle's maintenance situation, avoid odometer forgery while buying used vehicles and buy the maintenance products online without wasting time in the market. All the objectives described in the project proposal have been achieved successfully. Currently, we support the vehicle manufacturers Toyota, Honda, Suzuki, and Mitsubishi. The only hardware we have used is the "Ancel iCar2 iCar 2 OBD2 Scan Tool" rest of the things are managed by the mobile application. Our system will benefit users in the following ways:

- In keeping track of vehicle maintenance situations.
- Help in finding problems in vehicles via Fault Codes.
- In finding nearest workshops.
- Finding maintenance products of the vehicle.
- Avoid Odometer Forgery frauds for used vehicle buyers.
- Pay online through the app.

7.2 Future Enhancements

For now, our system is limited to using different vehicle manufacturing brands i.e Toyota, Honda, Suzuki, and Mitsubishi. Also for now we are product sellers in the app. Furthermore, there is no cross-platform support. In the future, there can be many possible enhancements but the most needed ones are:

- **Extend Support for More Vehicles**

As described earlier we are currently supporting vehicles manufactured by only four companies Toyota, Honda, Suzuki, and Mitsubishi in a future addition of vehicles manufactured by Kia, MG, Proton, Hyundai, that are growing rapidly these days will be needed in near future.

- **Cross-platform Support**

Due to the current limitation of resources and Flutter's Bluetooth support, the addition of Java native code restricts it to the Android platform only. Later it can be converted to cross-platform either by adding iOS native code or once Flutter Bluetooth support improves.

- **Integration of Better Payment Systems**

Currently, we are using Bykea's payment method that can later be improved via adding a Bank payment API.

- **Allow third party sellers**

Currently, we are dealing product sales on ourselves. Later, we can allow third party sellers to list their products.

Chapter 8

User Manual

8.1 Installation

This chapter is intended to give a new user an in depth explanation on how to navigate through the application and access the main, as well as the secondary features of the application.

This chapter will be mostly focusing on the fresh users and will take the reader from the primary steps of registering into the application to the steps where a user is required to connect the application to the OBD-II scanner.

8.2 Registering Yourself

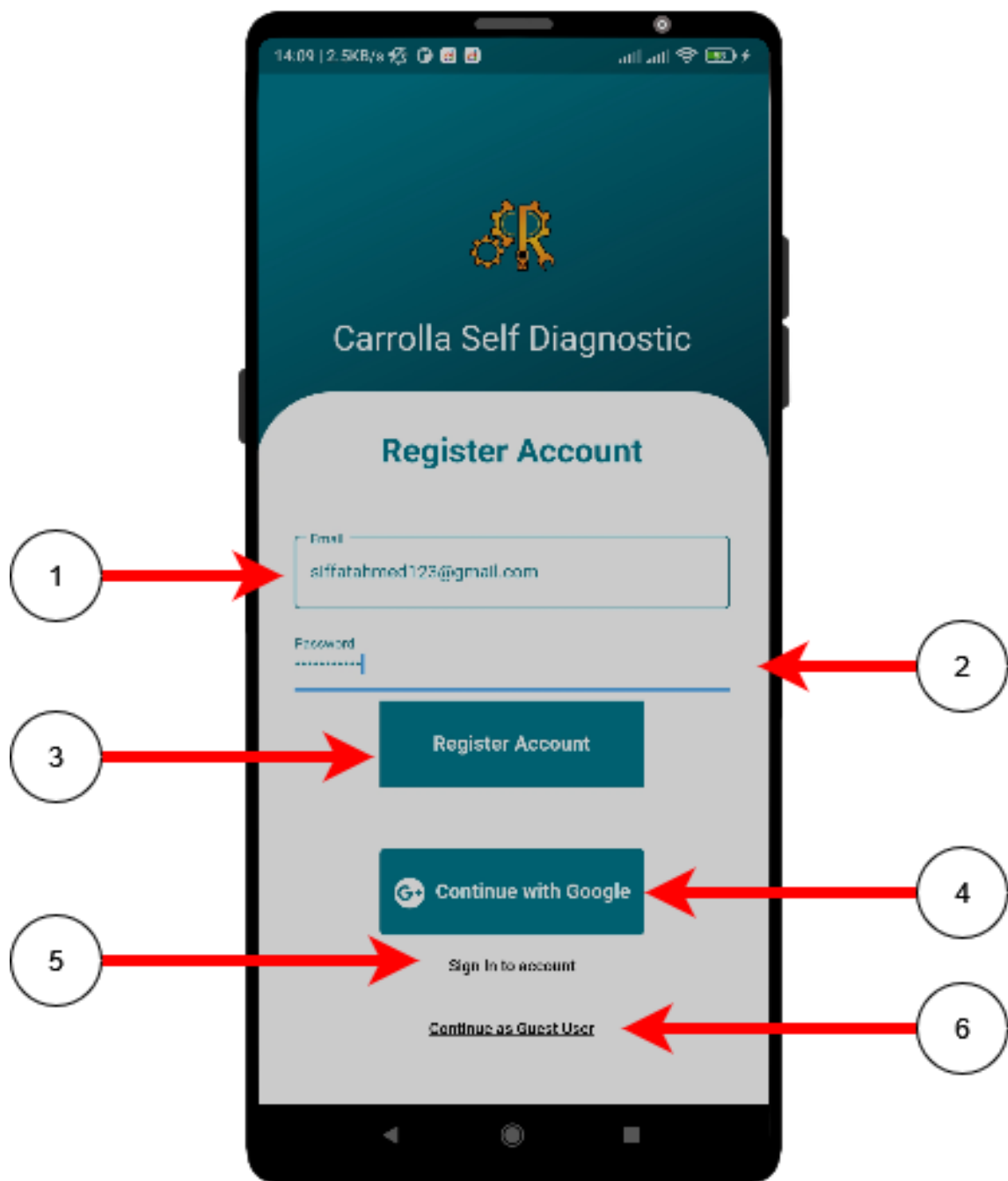


Figure 8.1: User Registration

1. Enter your Email for the account registration.
2. Enter a secure Password.
3. Click 'Register Account' Button and now your account will be created or if there is any problem such as user already exists or email format is invalid user will be informed.

OR

4. Use your Google account to register with the application.
5. If you are an existing user click 'Sign in to account' button.

OR

6. Continue as a Guest User by clicking 'Continue as Guest User' button.

8.3 Signing In

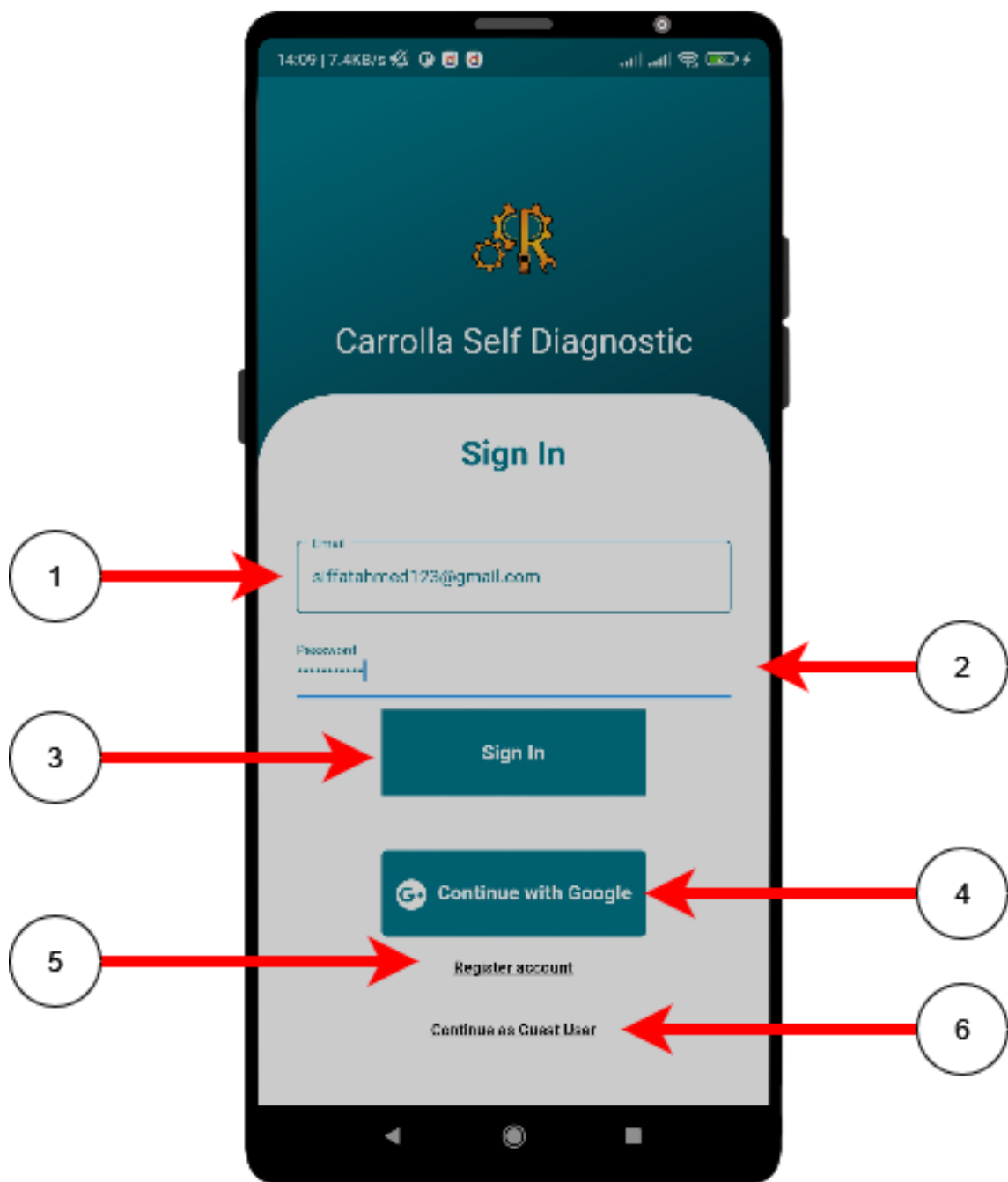


Figure 8.2: User Sign in

1. Enter your registered Email address to sign in.
2. Enter your password.
3. Click 'Sign in' Button and now your account will be signed in and if account doesn't exist or password is invalid error message will be shown.

OR

4. Use your Google account to sign in to the application.
5. If you are a new user click 'Register account' button.

OR

6. Continue as a Guest User by clicking 'Continue as Guest User' button.

8.4 Navigating Through The Tabs

8.4.1 Home



Figure 8.3: Home Page

1. Exit Application, Read data loaded based on Bluetooth data on cloud and start tracking journey buttons in the application bar.
2. Shows a short summary of vehicle maintenance situation.
3. Clicking 'Find Workshops' button leads to locating workshops in vicinity by using Google Maps.
4. Help Section Leads user to FAQs and Approach developer team.

Following are the controls inside the Bottom Navigation Bar.

5. Home Button leads to Main Dashboard that is the page currently open.
6. Car Repair shows pending maintenance list.
7. Leads to Workshop showing page that directs to Google Maps.
8. Notifications menu shows all the important notifications.
9. Setting menu let's user view different Privacy Policy, Terms of Service, and Account details.

8.4.2 Car Repair

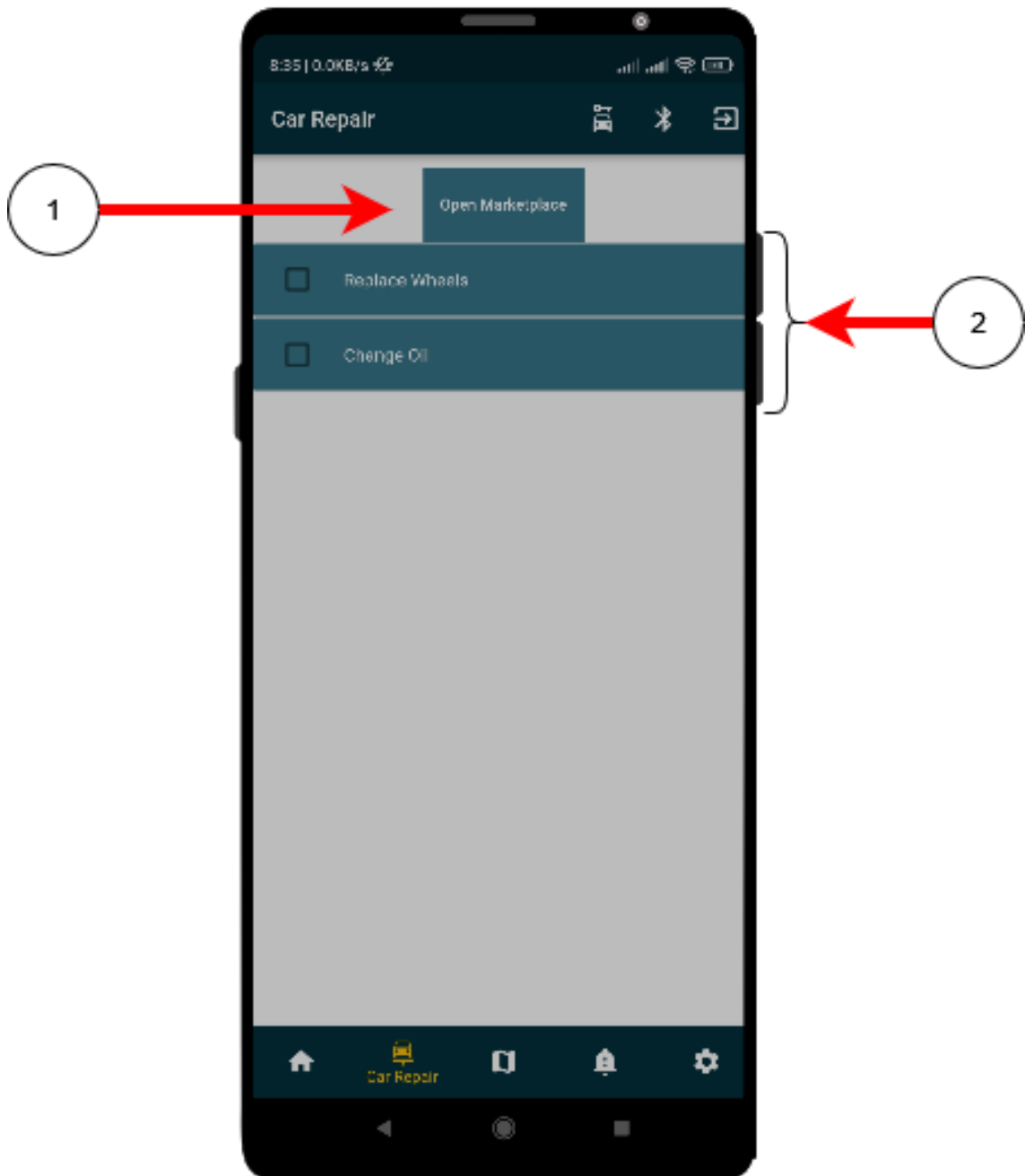


Figure 8.4: Car Repair Page

1. By clicking 'Open Market Place' button user reaches the Market Place where user can see list of all the Car care products and purchase them.
2. View list of pending maintenance.

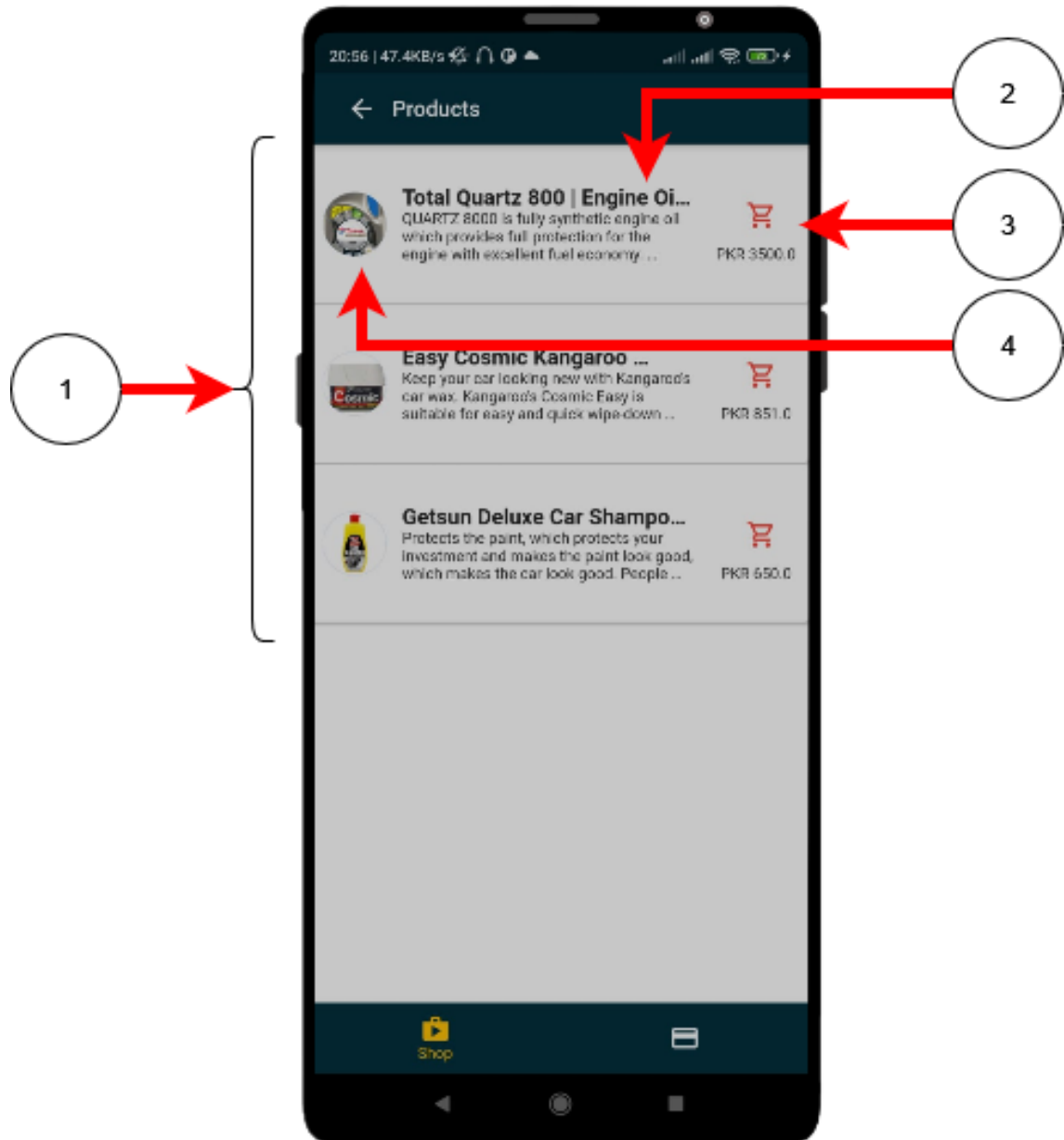
a. View List of Products

Figure 8.5: List of Products Page

1. This is a list of all the products.
Each entry consists of:
 2. Product name.
 3. Product Description.
 4. Product Price. Trolley icon lets you add product to the cart.

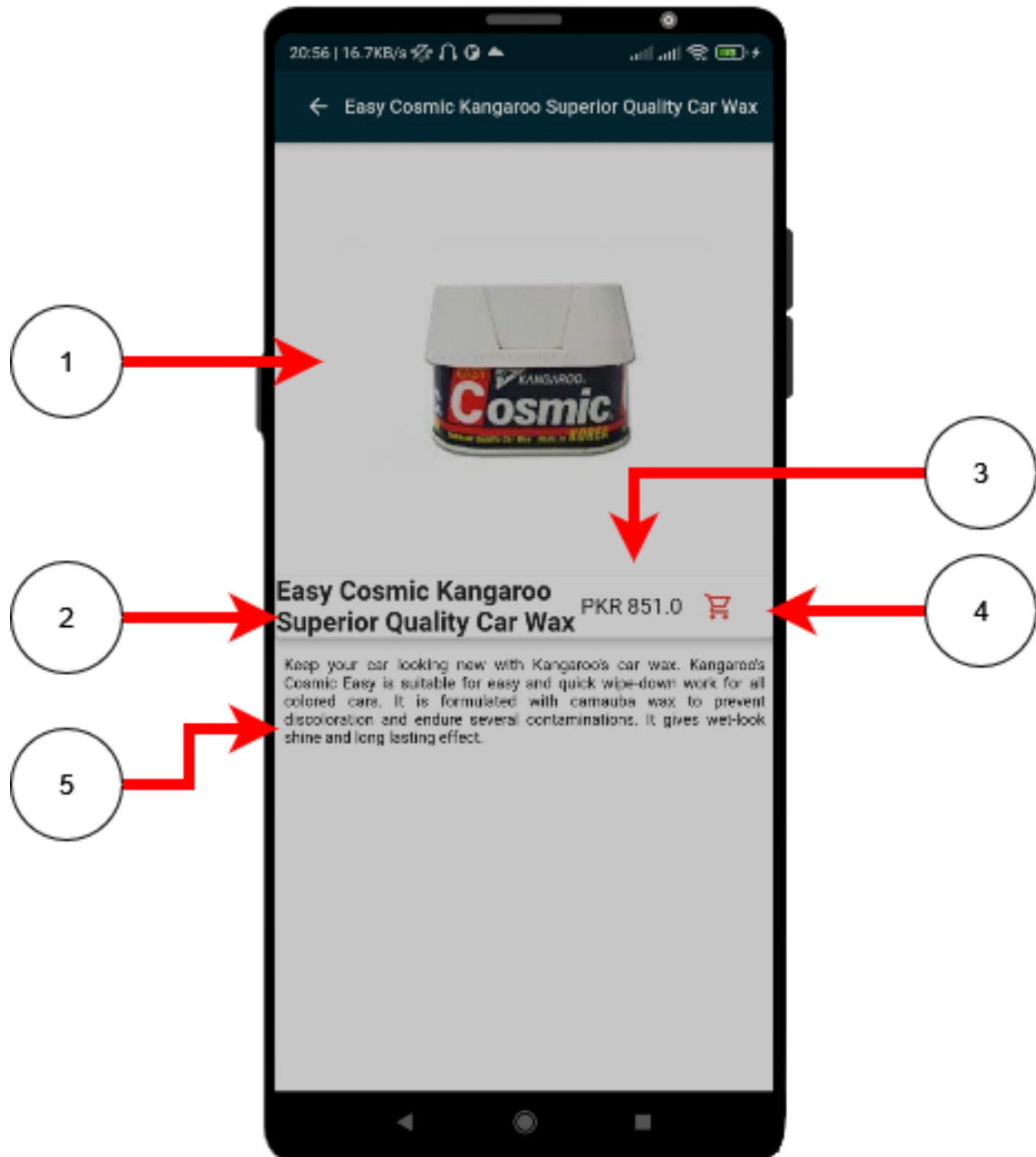
b. View Product Details

Figure 8.6: Product Details Page

1. By clicking 'Open Market Place' button user reaches the Market Place where user can see list of all the Car care products and purchase them.
2. View list of pending maintenance.

c. Checking Out



Figure 8.7: Check out Page

1. Shows total bill in Pakistani Rupee and number of products in the cart.
2. Clicking 'Checkout' button allows user to pay the total bill.
3. It shows list of Products in the cart.
4. Button with Recycle bin icon allows user to remove product from the cart.

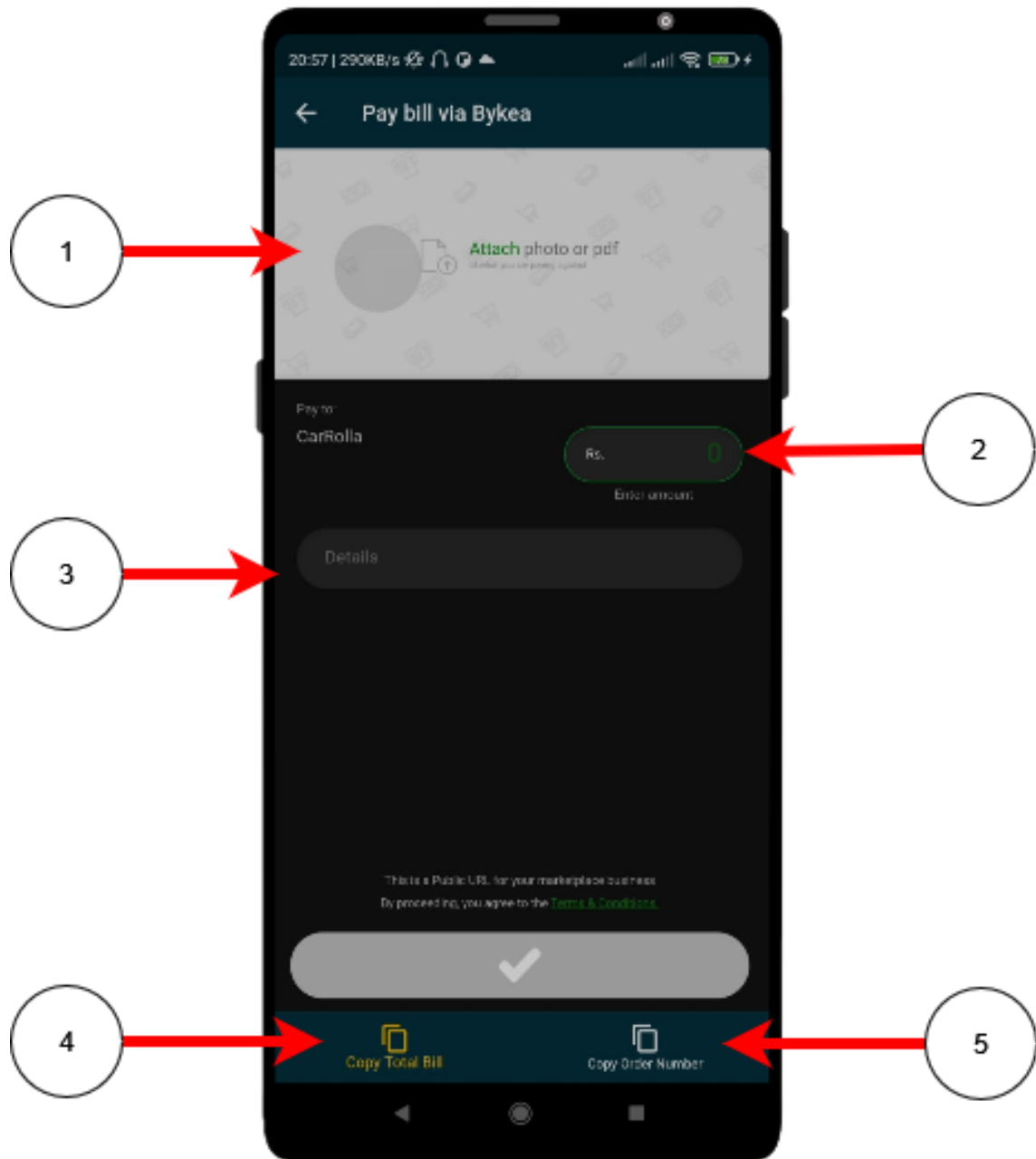
d. Pay the Bill

Figure 8.8: Bill payment Page

1. Attach the receipt generated or cart image in our case.
2. Enter the total amount. You can copy exact from navigation bar labeled as (4).
3. Enter Order Number in details field. You can the order number from navigation bar labeled as (5).
4. Copy your total Bill.
5. Copy your order number.

8.4.3 Maps

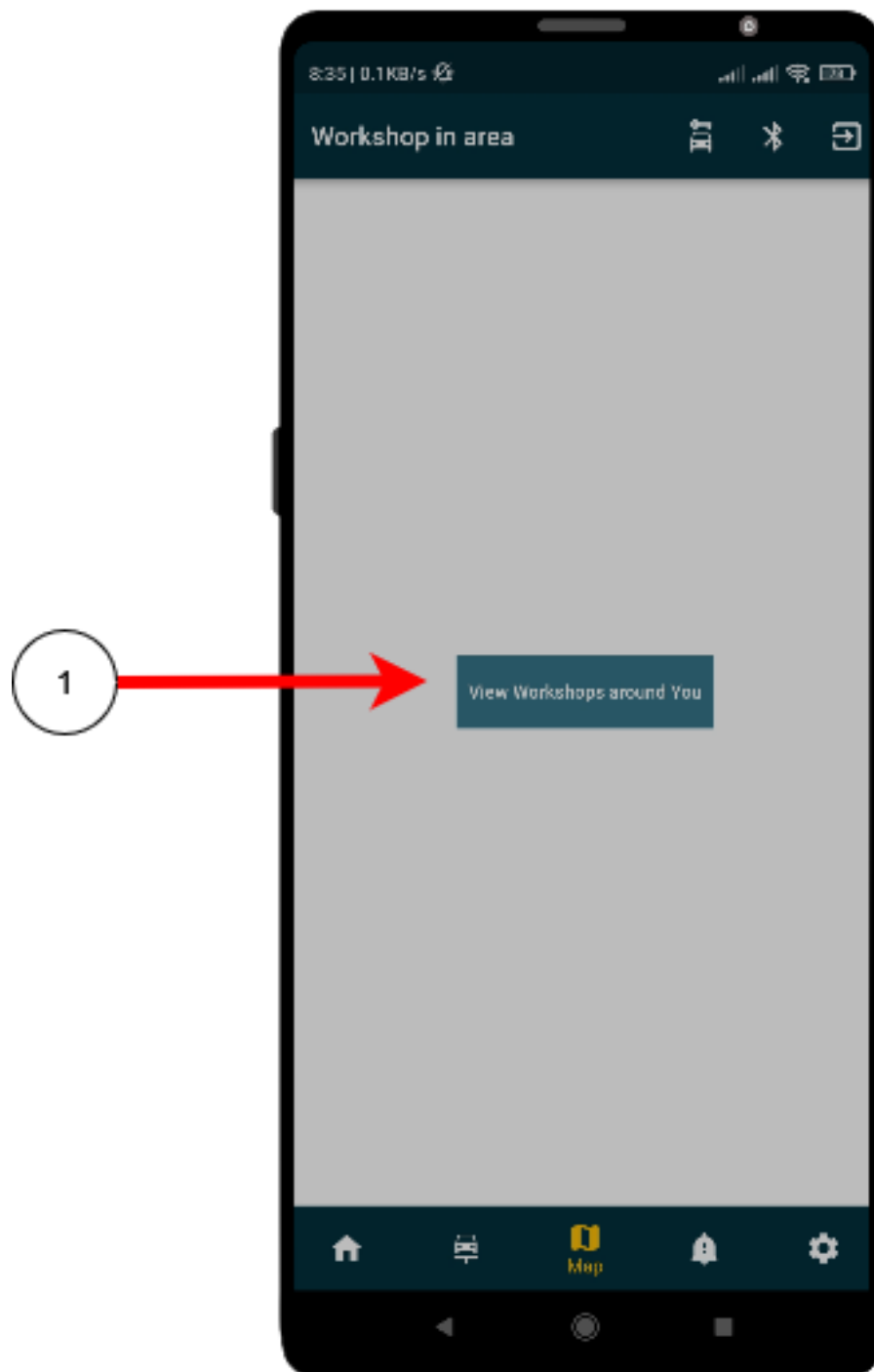


Figure 8.9: Towards Google Maps Page

1. By clicking user can find workshops in vicinity based on user location.

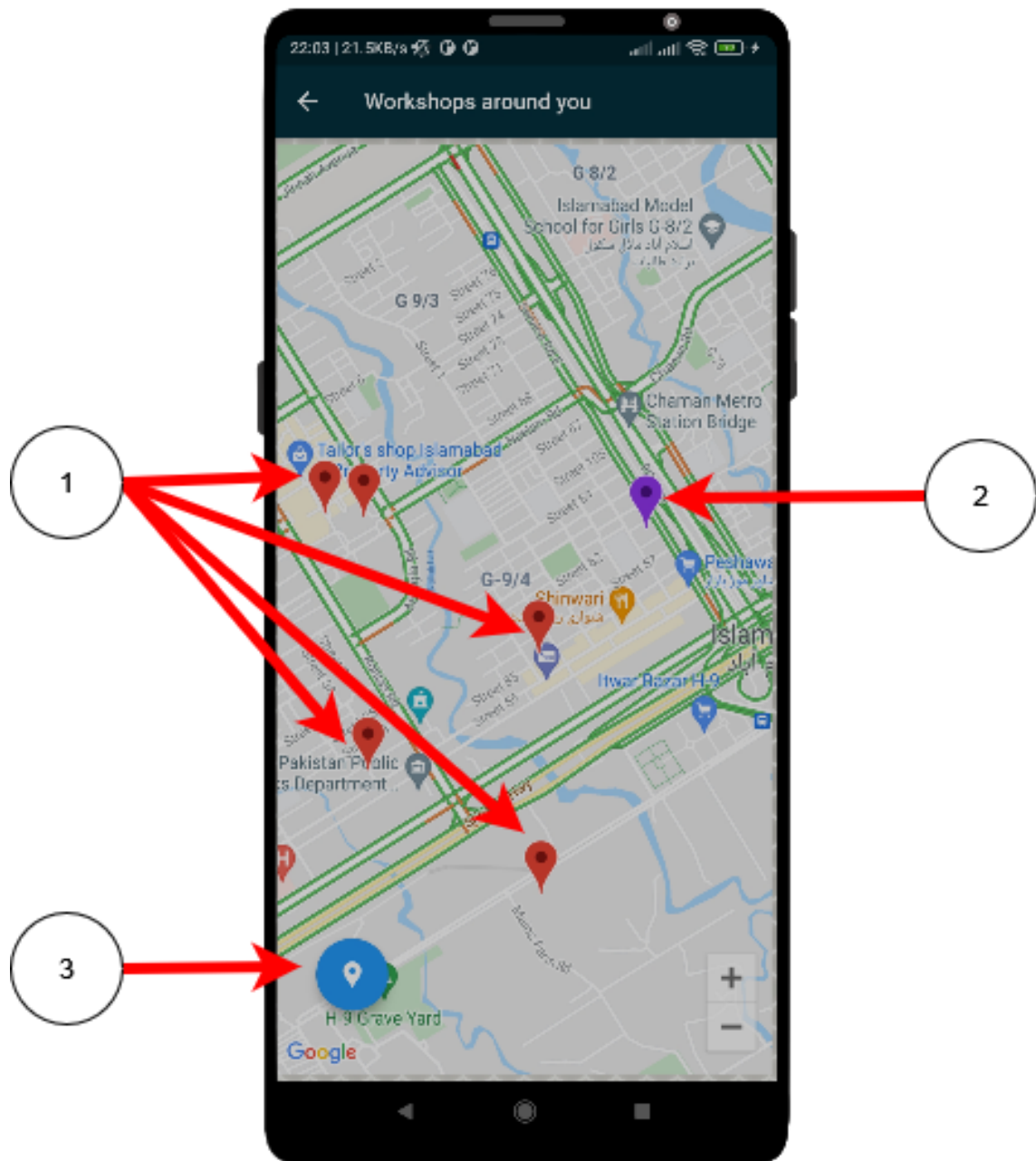
a. Viewing Workshops In The Vicinity

Figure 8.10: View Workshops on Map Page

1. Shows multiple workshops in the vicinity.
2. User's current location.
3. Change camera position on map to user's current position.

8.4.4 Notifications

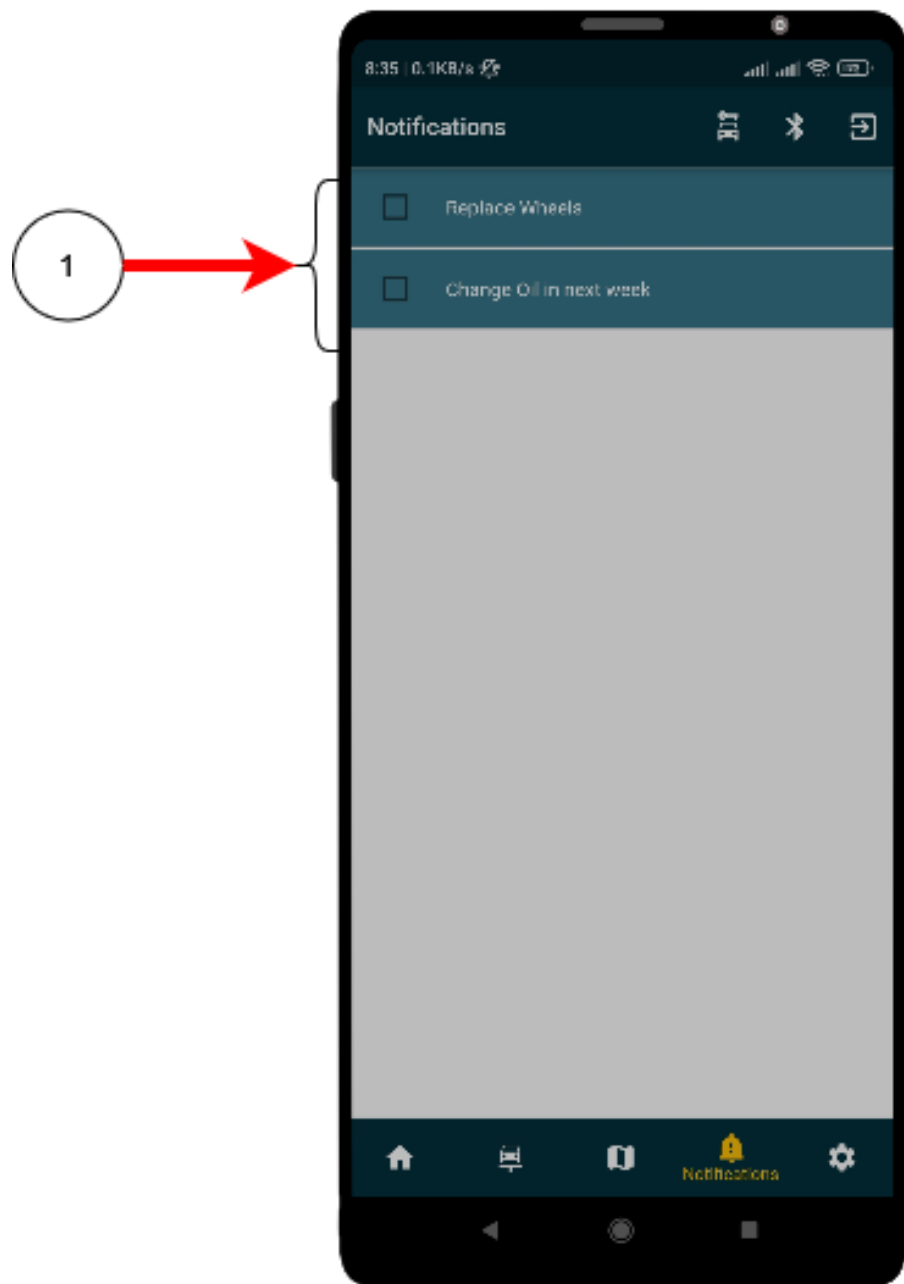


Figure 8.11: Notifications Page

1. User can view the notifications in the application and clear them by selecting the check box.

8.4.5 Settings



Figure 8.12: Product Details Page

1. View current user name and email.
2. Read Privacy Policy.
3. Read Terms of Service.
4. If user is stuck in anything, help is available under 'Help' section.

Bibliography

- [1] , “Full list standard fault obd codes.” [Online]. Available: <https://www.outilsobdfacile.com/data-trouble-code-obd2.php> Cited on pp. vii and 7.
- [2] T. Team, “Obd trouble codes for automotive technicians.” [Online]. Available: <https://www.tpcwire.com/blog/on-board-diagnostic-trouble-codes-for-automotive-technicians> Cited on pp. vii and 8.
- [3] bornagainbiker, “Suzuki service schedule.” [Online]. Available: <https://www.maxi-muppets.co.uk/forum/viewtopic.php?t=11506> Cited on pp. viii and 30.
- [4] D. McClave, “What is vin number.” [Online]. Available: <https://obdstation.com/vin-number/> Cited on pp. viii and 33.
- [5] Tutorialspoint, “Sdlc - iterative incremental model.” [Online]. Available: https://www.tutorialspoint.com/adaptive_software_development/sdlc_iterative_incremental_model.htm Cited on pp. viii and 53.
- [6] S. Odea, “Market share of mobile operating systems worldwide 2012-2021.” [Online]. Available: <https://www.statista.com/statistics/272698/global-market-share-held-by-mobile-operating-systems-since-2009/> Cited on pp. 4 and 6.
- [7] M. Carly, “How to detect odometer fraud with an app?” [Online]. Available: <https://www.mycarly.com/blog/car-check/how-to-detect-odometer-fraud/> Cited on p. 4.
- [8] C. Blog, “How to detect odometer fraud with an app?” [Online]. Available: https://www.mycarly.com/blog/car-check/how-to-detect-odometer-fraud/#How_to_detect_odometer_fraud_with_an_app Cited on p. 5.
- [9] S. Darby, “Top automotive innovations: History of fuel injection.” [Online]. Available: <https://bestride.com/news/top-automotive-innovations-history-of-fuel-injection> Cited on p. 7.
- [10] I. Hawkins, “Torque pro (obd 2 car).” [Online]. Available: <https://play.google.com/store/apps/details?id=org.prowl.torque&hl=en&gl=US> Cited on p. 22.

- [11] C. Ltd, “Obd auto doctor - elm327 obd2 car scanner tool.” [Online]. Available: <https://play.google.com/store/apps/details?id=com.obdautodoctor&hl=en&gl=US> Cited on p. 22.
- [12] O. O. Facile, “Eobd facile - obd 2 elm 327 car scanner torque pro.” [Online]. Available: <https://play.google.com/store/apps/details?id=org.eobdfacile.android&hl=en&gl=US> Cited on p. 23.
- [13] C. S. G. . C. KG, “Carly — obd2 car scanner.” [Online]. Available: <https://play.google.com/store/apps/details?id=com.iViNi.bmwhatLite&hl=en&gl=US> Cited on p. 23.
- [14] C. Solution, “My carly.” [Online]. Available: <https://www.mycarly.com/> Cited on p. 23.
- [15] H. S. G. Online, “Latitude and longitude.” [Online]. Available: https://gsp.humboldt.edu/olm/Lessons/GIS/01%20SphericalCoordinates/Latitude_and_Longitude.html Cited on pp. 26 and 27.
- [16] W. P. Review, “Where is pakistan in the world?” [Online]. Available: <https://worldpopulationreview.com/country-locations/where-is-pakistan\T1\textquotedblright> Cited on p. 27.
- [17] V. Barreto, “What is obdii? history of on-board diagnostics.” [Online]. Available: <https://www.geotab.com/blog/obd-ii/> Cited on p. 32.
- [18] F. P. Dev, “activity_recognition_flutter.” [Online]. Available: https://pub.dev/packages/activity_recognition_flutter Cited on pp. 34 and 66.
- [19] Bykea, “Bykea cash.” [Online]. Available: <https://bykea.cash/> Cited on p. 34.
- [20] F. Eilertsen, “email_validator.” [Online]. Available: https://pub.dev/packages/email_validator Cited on p. 77.
- [21] fluttercommunity.dev, “flutter_webview_plugin.” [Online]. Available: https://pub.dev/packages/flutter_webview_plugin Cited on p. 77.
- [22] yuli.dev, “uuid.” [Online]. Available: <https://pub.dev/packages/uuid> Cited on p. 77.
- [23] flutterdev, “google_sign_in.” [Online]. Available: https://pub.dev/packages/google_sign_in Cited on p. 77.
- [24] firebase.google.com, “firebase_auth.” [Online]. Available: https://pub.dev/packages/firebase_auth Cited on p. 77.
- [25] baseflow.com, “geolocator.” [Online]. Available: https://pub.dev/packages/google_maps_flutter Cited on p. 77.
- [26] baseflow.com, “geolocator.” [Online]. Available: <https://pub.dev/packages/geolocator> Cited on p. 77.
- [27] B. Whitfield, “Obd.” [Online]. Available: <https://pypi.org/project/obd/> Cited on p. 78.

- [28] Python, “Time.” [Online]. Available: <https://docs.python.org/3/library/time.html>
Cited on p. 78.
- [29] —, “Msvcrt.” [Online]. Available: [https://docs.python.org/3/library/msvcrt.html#
module-msvcrt](https://docs.python.org/3/library/msvcrt.html#module-msvcrt) Cited on p. 78.
- [30] Google, “Google maps platform.” [Online]. Available: [https://developers.google.
com/maps](https://developers.google.com/maps) Cited on p. 78.