

DESIGN AND IMPLEMENTATION OF VEHICLE MAINTENANCE AND SERVICES SYSTEM FOR AUTOMOBILE SERVICES COMPANIES IN NIGERIA

¹Patrick B. Asu-Nandi & ²Wahab Habib Babatunde

¹Department of Technical Education, Cross River State College of Education, Akamkpa +2348054131125; Patrick.Nandi@Crs-Coeakamkpa.Edu.Ng

²Department of Technical Education, Kwara State College of Education, Ilorin, Khhart6@Gmail.Com; 08052275955

Abstract

The vehicle maintenance and service system are an automation of the various vehicle services needed by a vehicle user in a mobile application. This application provides vehicle service reminders while providing vehicle-related solutions. The vehicle maintenance and service system provide repair cost estimates to help mitigate the rigged system vehicle users pass through for vehicle repairs and maintenance. This paper contains a review of existing systems related to the proposed system and the prototype development methodology used to develop the system. It also provides the design of the system using the Unified Modelling Language as well as testing of the system. The examination of the literature uncovered several empirical evidence areas, and the resulting discussion provides the basis for the design and implementation of vehicle maintenance and services systems.

Keywords: Vehicle, Vehicle maintenance system, Automobile services companies

Introduction

Over the years the use of vehicles has played a vital role in the lives of humans in their entirety, cutting down days and weeks of journey to hours. Since the introduction of vehicles, there has been a need for regular maintenance for every manufactured vehicle, such maintenance is necessary to guarantee safety and a longer lifetime for these vehicles. In the course of this regular maintenance, the engine oil and filter are replaced. The schedule of this maintenance differs depending on the vehicle, the manufacturer of each vehicle recommends the schedule of the maintenance based on factors like distance travelled, time intervals, and weather conditions (Collins, 2018). Other than regular maintenance, the need for repair arises due to breakdowns, faulty parts, and auto crashes. In Nigeria today with a vehicle population of 11.6 million there has been a non-stop need for such repairs. 2018 According to (Adesoji, 2019), 18,729 people were involved in road accidents in Q4 2018 as opposed to 15,696 people in Q4 2017 noticing a 19% increase. These accidents are majorly linked to bad and faulty vehicles.

Since the dawn of time, one of man's main goals is to make moving from one place to another easier and more efficient. As such, man has come up with a wide range of ideas ranging from riding horses, carts, and canoes to present-day cars, airplanes, boats, etc. The first internal combustion four-cycle, gasoline-fueled engine was manufactured in 1876 in Germany,

Carl Benz commenced the first commercial production of vehicles with combustion engines. Vehicles reached the modern stage of development by the 1890s. The models in the 1980s were so successful which has led to no significant transformation in the principles of the automobile engine used then (Melosi, n.d.). Cars today are used throughout the world as the most popular mode of transportation, making it highly important to people. Although the car was to have its greatest impact on man, making the journey easy, like any other mode of transportation it has its flaws, the constant need for maintenance and the non-ending need for repairs. Though there are many types, models, brands, and makes of cars, they all require these services.

The use of vehicles was introduced to Nigeria when oil was flourishing in the 70's by the Western world. The Nigerian government in partnership with some advanced economies like Germany, Britain, America, and France, issued a license of operation and control policy, built six Vehicle assembly plants in various states of the country, Peugeot Automobile Nigeria Limited (PAN) established in the year 1975 in Kaduna, Volkswagen of Nigeria Limited (VWON) Lagos 1978, Anambra Motor Manufacturing Limited (ANAMMCO) Emene – Enugu (1980); Steyr Nigeria Limited Bauchi in Bauchi, National Truck

manufacturers (NTM) Kano Fiat Production in Kano, LeyLand Nigeria Limited (LNL) Ibadan, between 1970 and 1980 (Akaigwe, 2010). The launch of the six-vehicle plant in Nigeria served as an economic boost to the country (Aganga, 2013). The former minister of trade and investment (Aganga, 2013) said that the "automobile industry is an engine of growth and stimulus to other economic activities like creating of employment opportunities growth of other satellite industries, enhancement of technology transfer of skill acquisition" since the establishment of the vehicle assembly plants in Nigeria, the country's economy has recognized an enormous improvement as a result of the ease in transporting finished products of crude oil from the factories to fuel filling stations. Farm products are transported with ease between the six geopolitical zones. Humans were able to move from one point to another with ease. Compared to when man was using other means of transportation such as horses, camels, etc. Over the years there has been a sharp decline in vehicle manufacture in the country, some of the assembly plants manufactured in the 70s and 80s are not operational, due to two challenges that have befallen them (Okuhit, 2015). The majority of automobile companies in Nigeria in present times deal only with car sales, about 80% of vehicles on the roads today are fairly used vehicles popularly known as Tokumbo, as a result of the decline in vehicles manufactured in Nigeria (Chamberlain, 2013). According to (Izogo & Ogba, 2016), the need for vehicle repair services has been on the rise due to the change from manual to automatic transmissions having more complex systems due to technological variations.

Statement of the Problem

The purpose of a maintenance and services system is to track and acquire relevant information for vehicle-related services and maintenance, while also tackling arising problems. Problems such as breakdown and faulty parts such as wear and tear of tyres, brake pads, tie rods, sharp, etc might come up while on the move leaving the vehicle user stranded. The proposed system will provide a go-to for stranded individuals to request immediate help. In tackling these problems both mechanics and vehicle users face tremendous challenges with the vehicle users taking a fair share of these challenges. In our environment today vehicle users end up paying exorbitant amounts of money on spare parts hiked by both the mechanics and those that sell these spare parts. The vehicle maintenance and services will provide an interface to purchase these

parts at the exact retail price while also providing repair estimates.

Objectives of the Study

The project is aimed at implementing a vehicle maintenance and services system. The specific objectives of the research paper are:

- i. To review related materials on vehicle servicing and maintenance systems.
- ii. To identify the requirements for vehicle maintenance and services system.
- iii. To design vehicle maintenance and service system models using the unified modelling language.
- iv. To implement the design models for vehicle maintenance and services systems.

Methodology

The prototype development methodology was adopted for the development of the vehicle maintenance and services system. The methodology process allows developers to create only the prototype of the solution to assess its functionality and make necessary modifications before developing the actual application. The best feature of this software development methodology is that it solves many issues that often occur in a traditional waterfall model. It is an example of a plan-driven process in principle, one must plan and schedule all of the process activities before starting work on them. This method drastically decreases the risk of failure, as probable risks can be detected in the early stage and moderation steps can be taken quickly. Other than the prototype model there are other models such as the Waterfall, Agile, Spiral model, etc. The waterfall model is based on a sequential flow requiring the previous phase to be completed before moving on to the next phase. Unlike the prototype model, the waterfall model can only be adopted in the development of projects whose requirements are clearly stated. Agile software development methodology is a blend of both incremental and iterative process models that allows recurrent modification in development. One drawback of the agile development methodology is the possibility of drifting from the path to the required goal as a result of ambiguity in the customer's understanding of the outcome of the project. The prototype development methodology helps in requirement gathering and requirement analysis when requirement documents are absent. When a prototype is shown to the clients, they get a vibrant understanding and whole sense of the functionality of the software. Thus, the prototype methodology has been chosen to be the methodology

to be used in the development of the Vehicle Maintenance and Services System.

Approach to Chosen Methodology

The approach in developing the project was an iterative, trial-and-error method with the users. Requirements were gathered and analyzed by interviewing users and knowing their expectations of the system. The second phase paves the way for a quick design of the system. It gives an idea to the user of the system and a guide for building the prototype. A prototype of the proposed system is built and presented to the user for evaluation. In this stage, comments and suggestions from the user are noted and any alteration to the system requirement is put into consideration when refining the prototype. A refined prototype is built with all requirements as specified by the user and sent for final evaluation. The system was then developed based on the final prototype

Tools and Techniques

The tools used during this project are

- A computer of 2.4GHz, 8GB of RAM, and 500GB SSD
- Visual Studio Code
- Microsoft Office
- Firebase Cloud Firestore
- Visual Paradigm

Technologies include

- React

- React Native
- Node JS
- JavaScript

Ethical Consideration

The ethical consideration is the accumulation of values and principles that address questions of what is good or bad during the process of conducting this research. Firstly, handling, storing, and sharing of user data only under the circumstances and for the research. Energy consumption of the system artifact was regulated, caring about energy throughout the development process and in the documentation. Also, the decisions on which bugs to fix and how quickly, were done to ensure the quality of the code before release. Should the system cause any harm, the administrators will be held responsible.

Requirement Analysis

To analyze the system requirements, responses were collected from the users of the system. More specifically they were consulted to establish the goals, requirements, and services that are expected from the system. This involved a proper definition of the nature scope and peculiarity of the problem. The problem with which this project is based is the design and implementation of a vehicle maintenance and services system that does not suffer the setbacks of the initial prototype. An interview was conducted to collect feedback from the users, which served as an additional requirement of the proposed system.

Requirements Specifications**Table 1: Functional Requirements Specification**

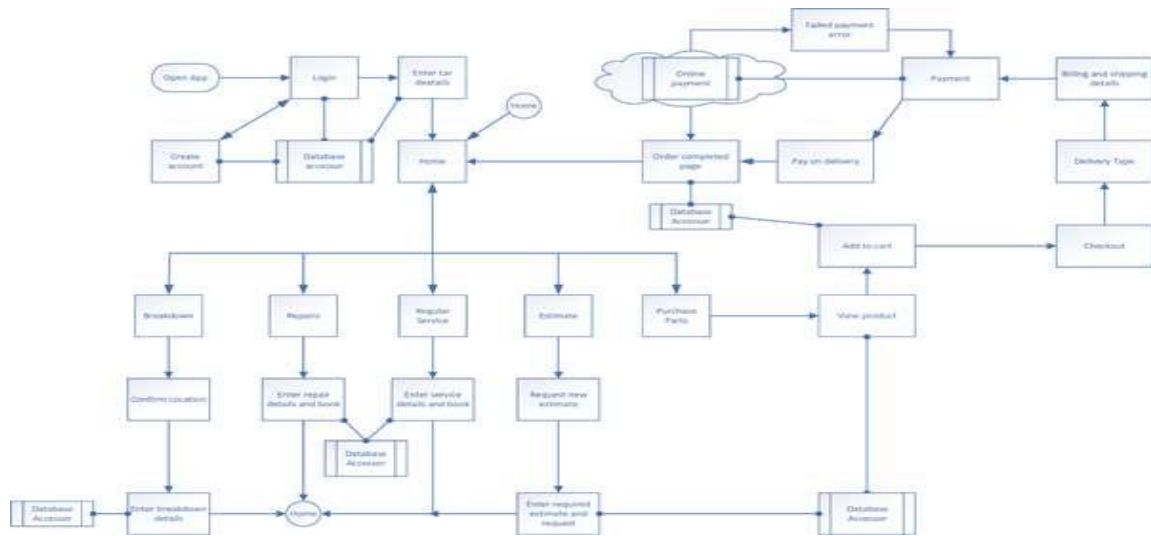
Req. No.	Description	Type
R-101	The system shall allow a new User to create an account.	Configuration
R-102	The system shall allow a User to log in to the system.	Functional
R-103	The system shall allow a user to enter his/her vehicle details.	Functional
R-104	The System shall allow a User to request a breakdown.	Functional
R-105	The system should be able to collect User location.	Configuration
R-106	The system shall allow a User to book a repair.	Functional
R-107	The system shall allow a User to book a service.	Functional
R-108	The system shall allow a User to request an estimate.	Functional
R-109	The system should allow a User to purchase parts.	Functional
R-110	The system should be able to get service dating using mileage details	Configuration
R-111	The system should send notifications when a User's vehicle is set for service.	Configuration
R-112	The system should allow a User to view order details.	Functional
R-113	The system should allow a User to view breakdown details.	Functional
R-114	The system should allow a User to view booking details.	Functional
R-115	The system should allow Users to make payments online.	Configuration
R-116	The system should allow the Admin to view estimate request details.	Functional
R-117	The system should allow the Admin to view booked repair details.	Functional
R-118	The system should allow the Admin to view booked service details.	Functional
R-119	The system should allow the Admin to view breakdown details.	Functional
R-120	The system should allow the Admin to view order details.	Functional
R_121	The system should allow the Admin to send an estimated price to the user.	Functional
R_122	The system should allow the Admin to add a product	Functional
R_123	The system shall run on iOS 10.0 and Android 4.1 or later.	Configuration

Non-Functional Requirement Specifications**Table 2: Non-Functional Requirements Specification**

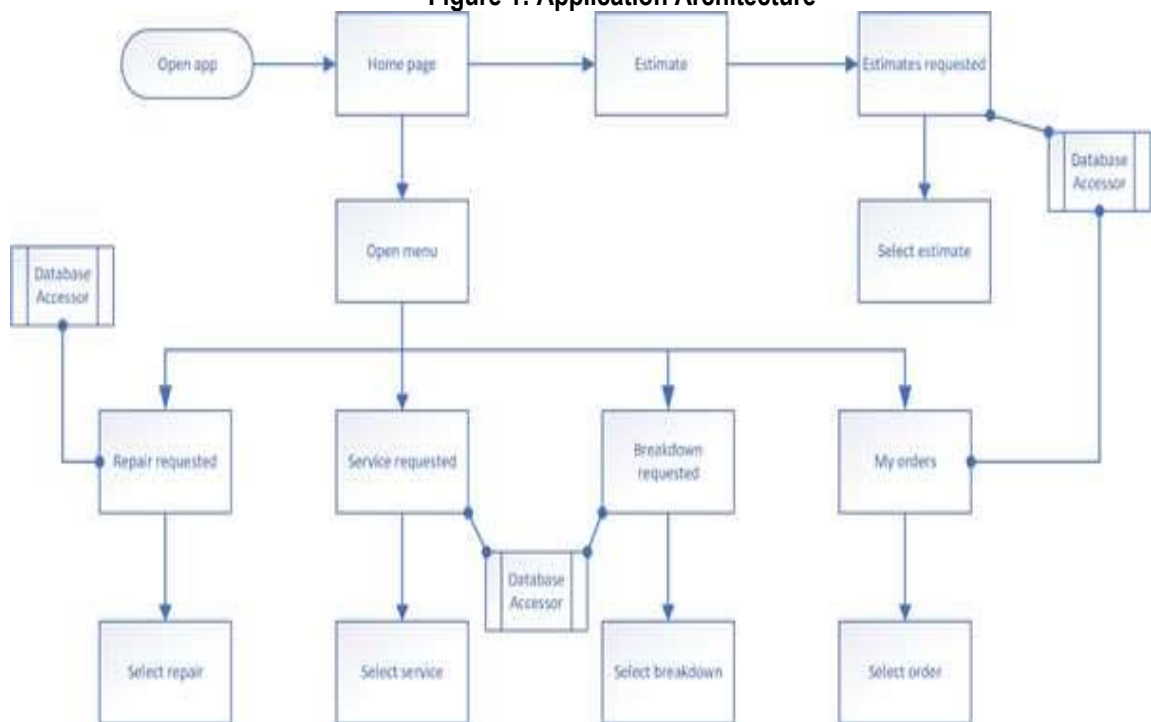
Req. No.	Description	Type
R-101	When launched, the application shall stay running unless there is an intentional shutdown of the application or the platform.	Performance
R-102	The system screens should load in not more than 5 seconds.	Performance
R-103	Online payments should be made secure.	Security
R-104	The system shall ensure a user can view only his/her details.	Security
R-105	The system shall have a user-friendly interface.	Usability
R-106	The system should not be restricted to a single OS.	Portability

System Design

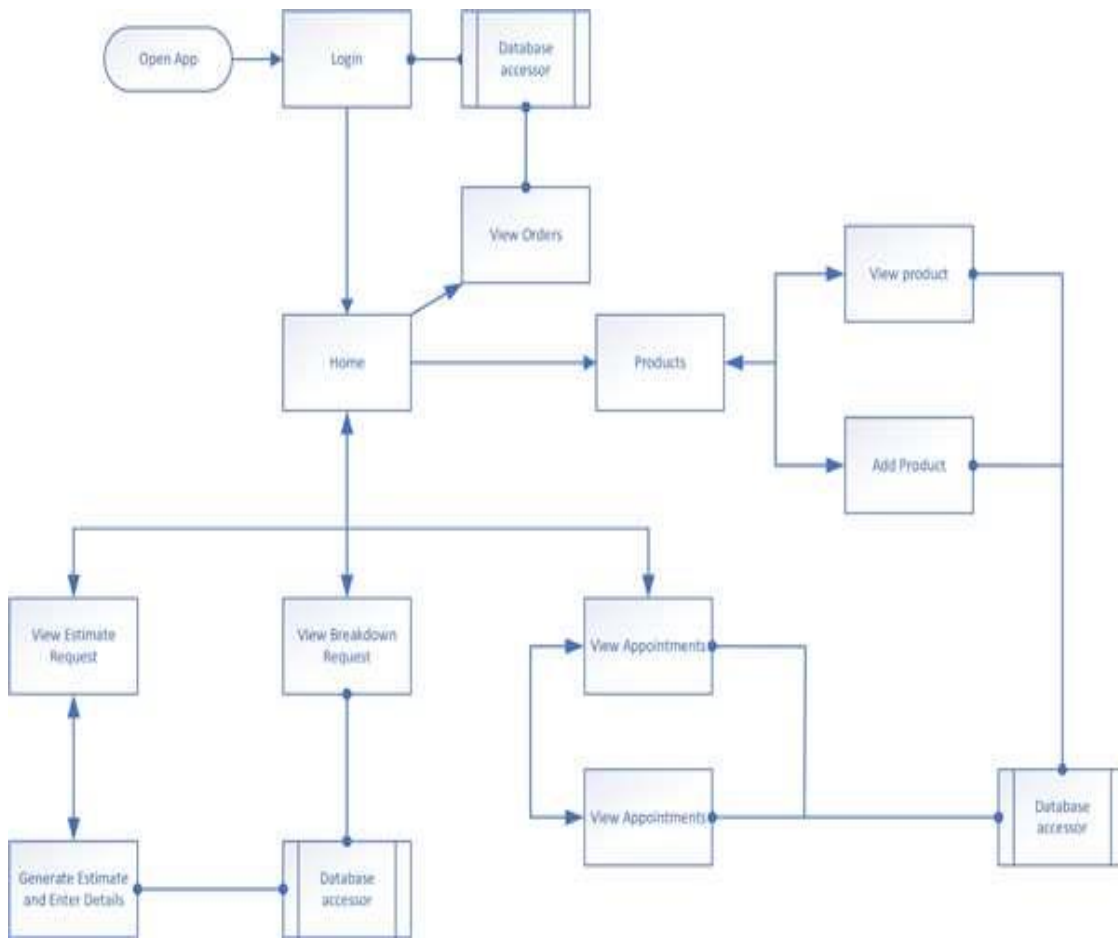
Application Architecture



Source: Researchers design
Figure 1: Application Architecture



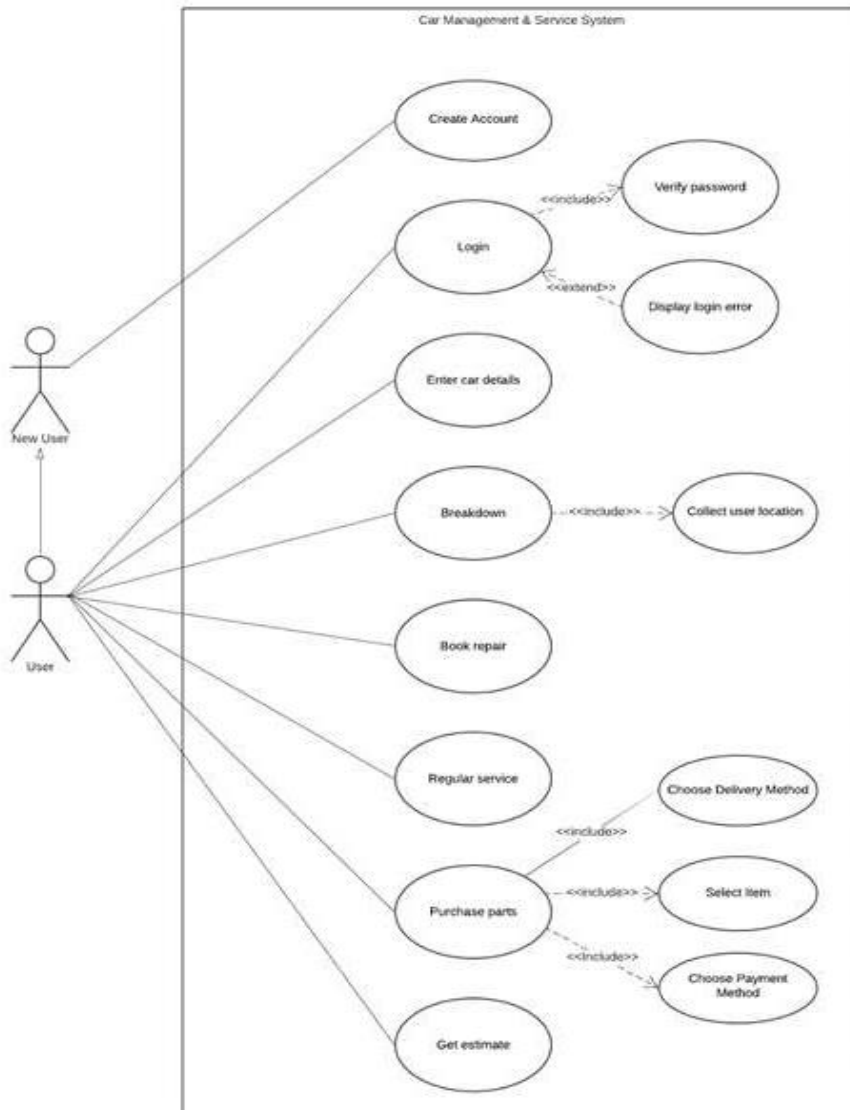
Source: Researcher design
Figure 2: Services Details Architecture



Source: Researchers design
Figure 3: Admin System Architecture

Figures 1–3 are the application architecture. These architectures give a high-level representation of the vehicle maintenance system for client and admin applications.

Use Case



Source: newprojecttopics.com

Figure 4: Use Case

Figure 4 shows the interactions between the system and the user of the system. The stick figure at the top left side of the rectangle represents a new user of the system who initially creates an account to gain access to the system while the stick figure underneath the new

user represents an existing user. The rectangle defines the scope of the system. The oval shape (use case) within the rectangle represents actions that accomplish some tasks within the system.



Source: newprojecttopics.com
 Figure 5: Admin System Use Case

The figure 5 shows the interactions between the system and the admin of the system. The stick figure at the left side of the rectangle represents the admin and the rectangle defining the scope of the system.

The oval shape (use case) within the rectangle represents actions that accomplish some tasks within the system.

Sequence Diagram

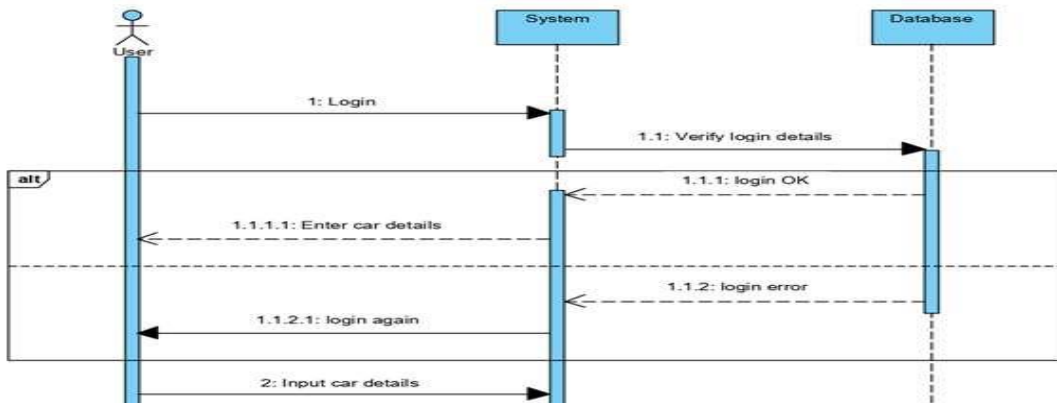


Figure 6: Login Sequence Diagram

The figure 6 shows the login process and interactions in the order they take place.

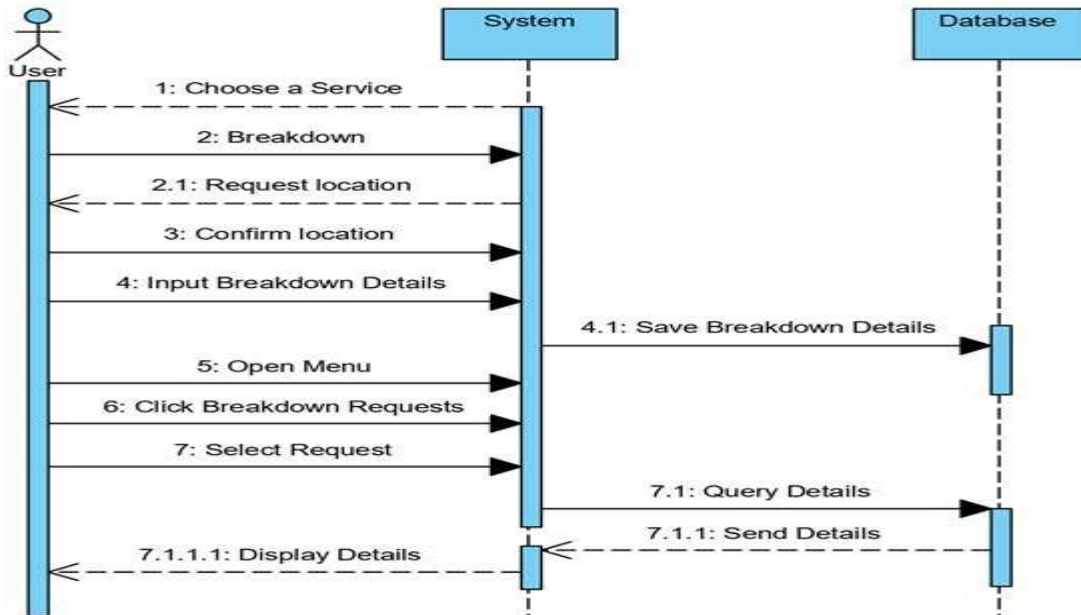


Figure 7: Requesting Breakdown Service Sequence diagram

The figure 7 shows the process of requesting and viewing a breakdown service and interactions in the order they take place.

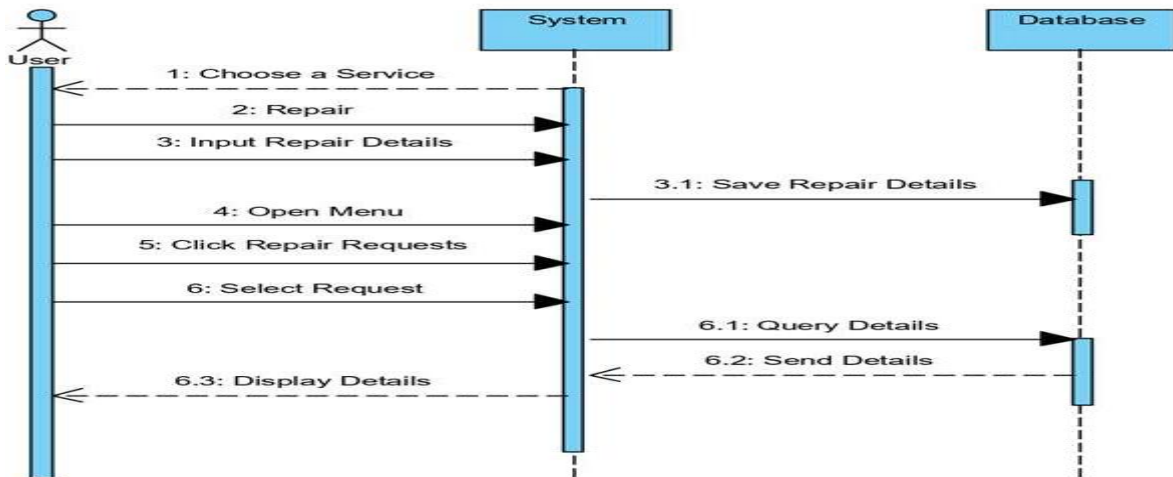


Figure 8: Repair Sequence diagram

The figure 8 shows the process of booking and viewing details of repair and interactions in the order they take place.

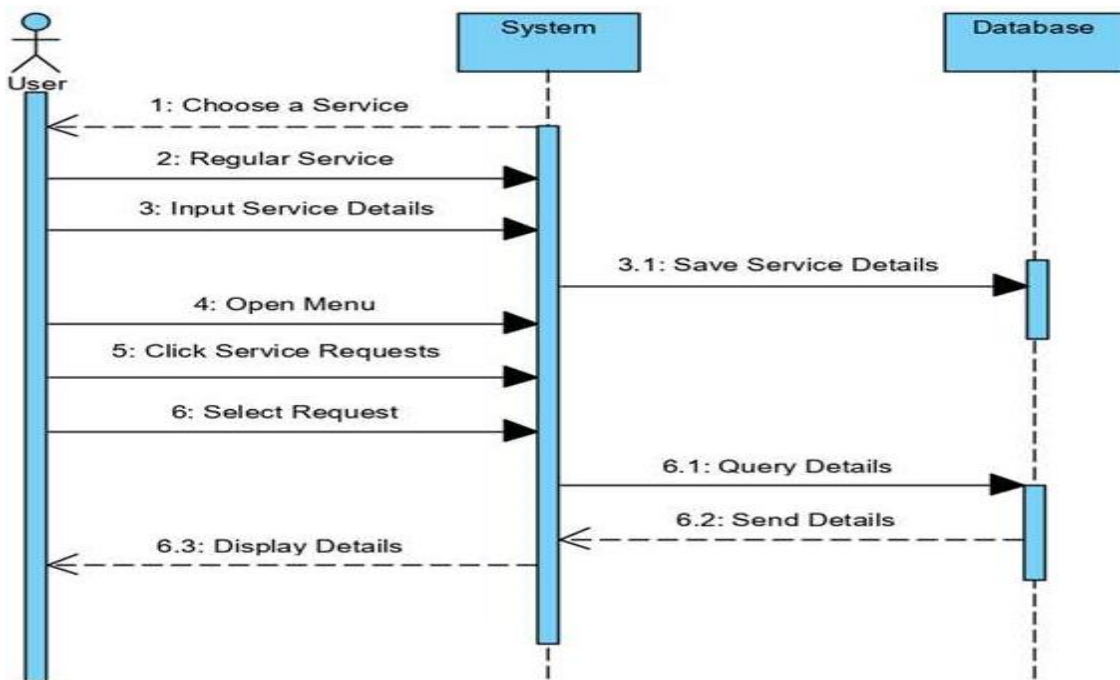


Figure 9: Service Sequence diagram

The figure 9 shows the process of booking and viewing details of a service and interactions in the order they take place.

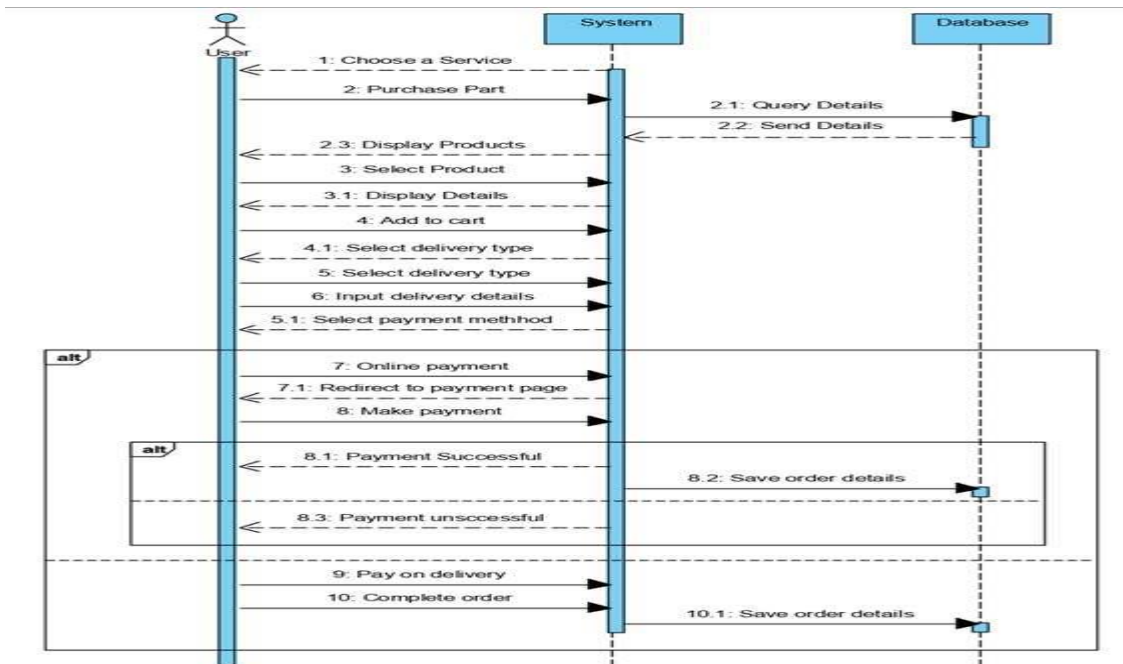


Figure 10: Purchase Parts Sequence diagram

The figure 10 shows the process of placing an order for spare parts with different delivery and payment options.

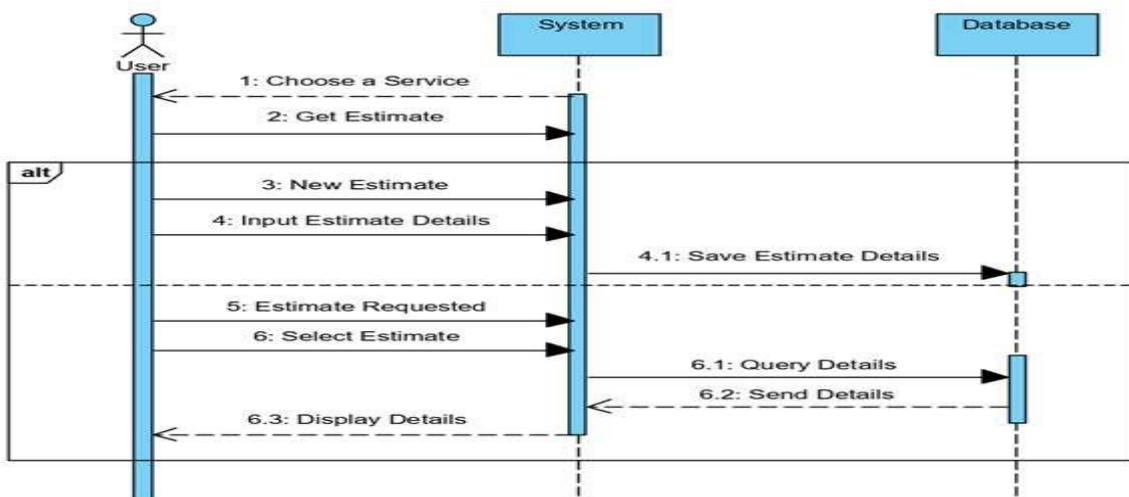


Figure 11: Estimate Request Sequence Diagram

The figure 11 shows the process of requesting an estimate and viewing estimate details and interactions in the order they take place.

Context Diagram

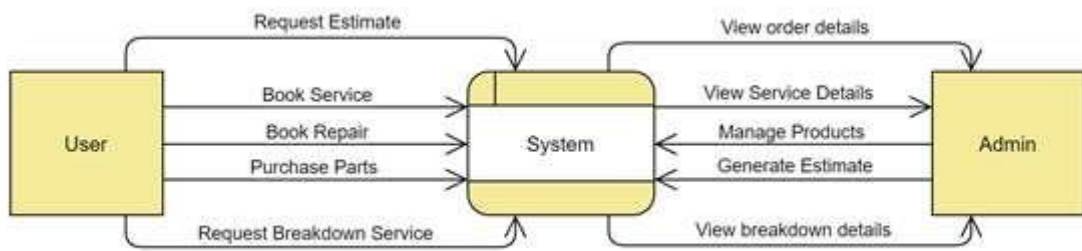


Figure 12: Context diagram

The figure 12 is a context diagram of the Vehicle Maintenance and Service System. The diagram defines the boundaries of the Vehicle Maintenance and Service System. It depicts the flow of information in the system.

Entity-Relationship Diagram (ERD)

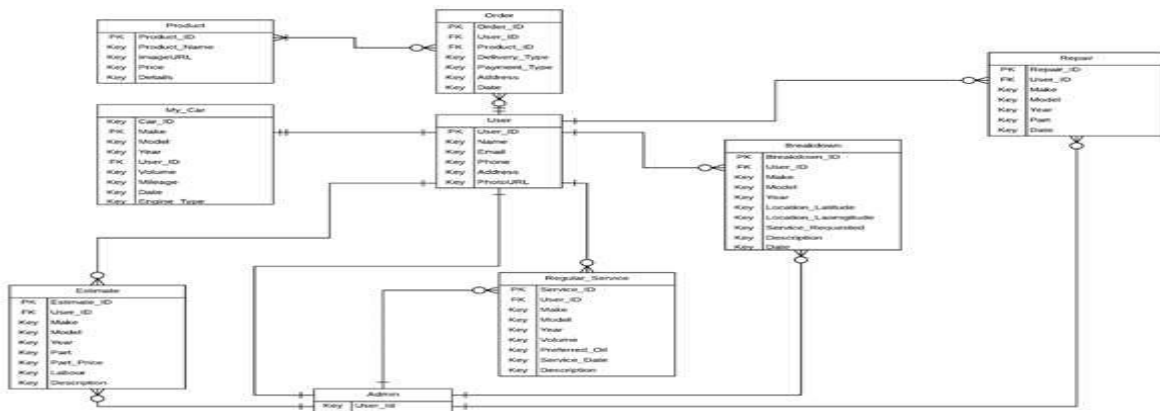
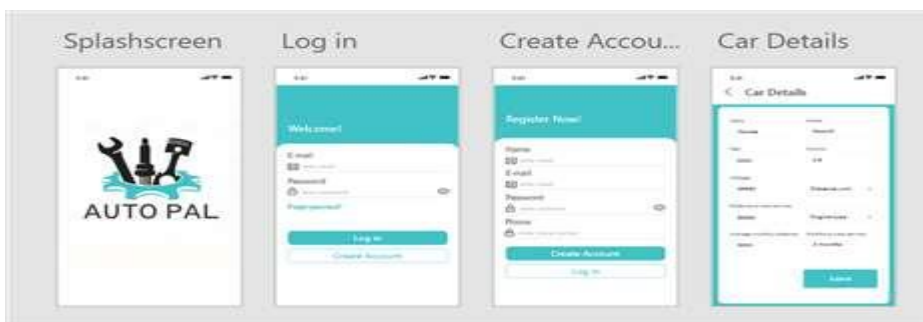
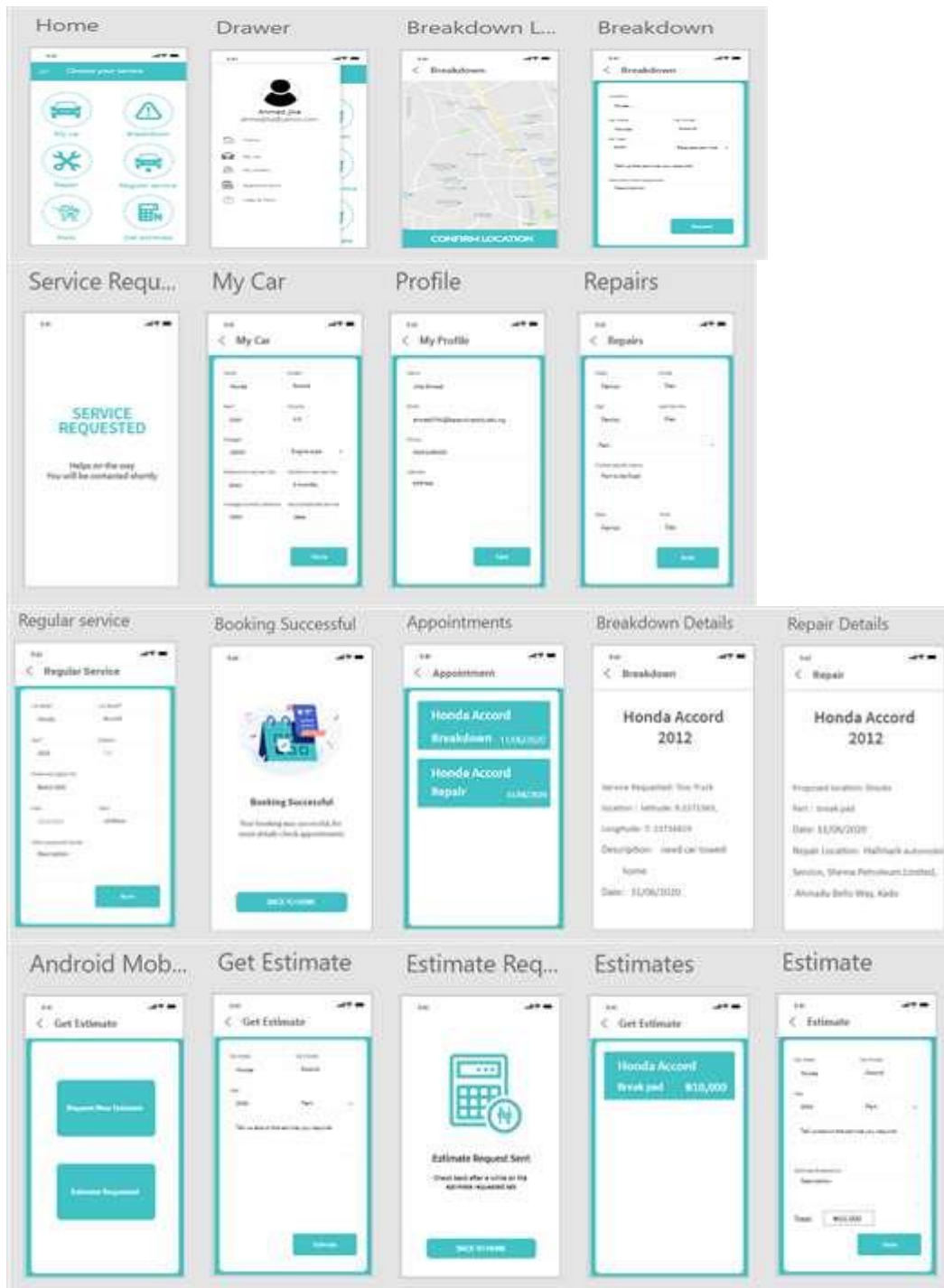


Figure 13: Entity Relationship Diagram (ERD)

The figure 13 shows The Vehicle Maintenance and Service System ERD. The ERD shows the entities, it's attributes and relationship between them

User Interface Design





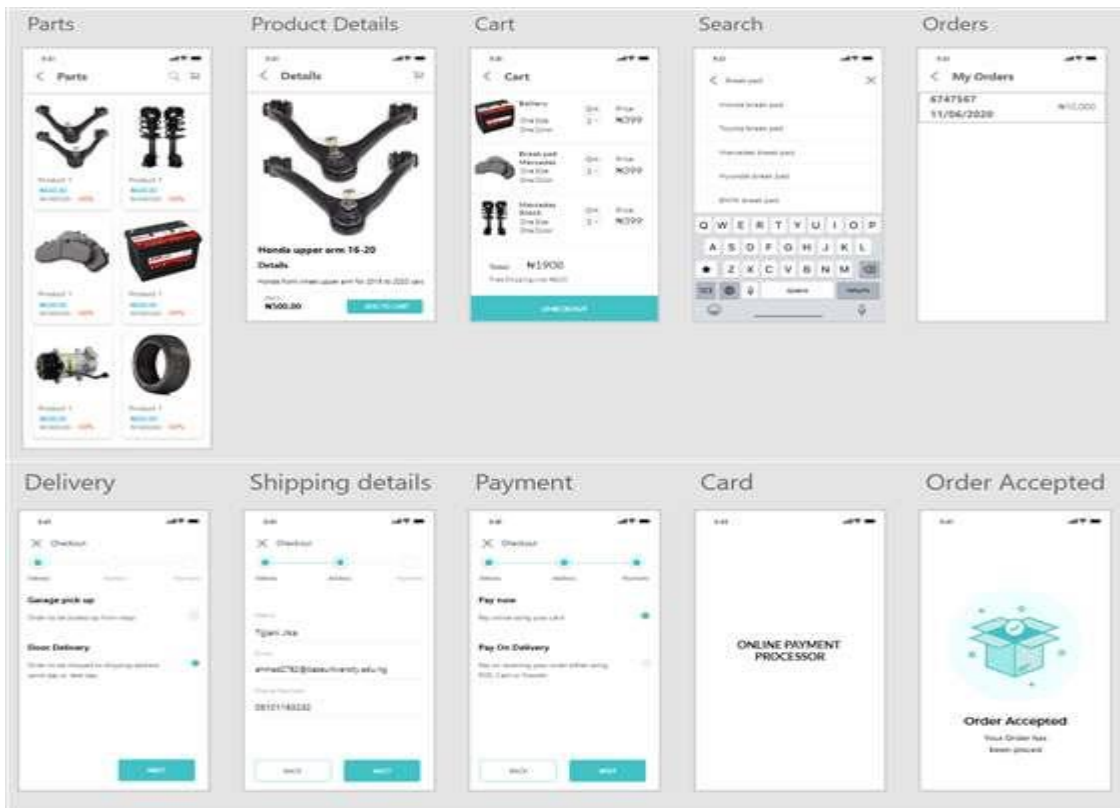


Figure 14: User Interface Design

Summary

This section of the paper talks about the methodology used in the development of the system, the requirements of the system both functional and non-functional, and UML diagrams (Activity Diagram, use case diagram, sequence diagram, etc.) used to model the requirements of a system.

IMPLEMENTATION AND TESTING

Overview

This section emphasizes the implementation and testing of the system, illustrating the tools and technologies used for the development and testing of the system.

Test suite, test cases, test reports, and error reports related to the system (Zhi & Garousi 2013). This section also shows the problems encountered during the implementation, and how the problems were overcome.

Main Features

Breakdown: In times of an unforeseen breakdown a user can request a breakdown service which will provide professional help (Gongolidis, Kommata, Papantoniou & Loumos, 2012). A user's location and other relevant details will be collected using Google Maps API for an accurate pinpoint.

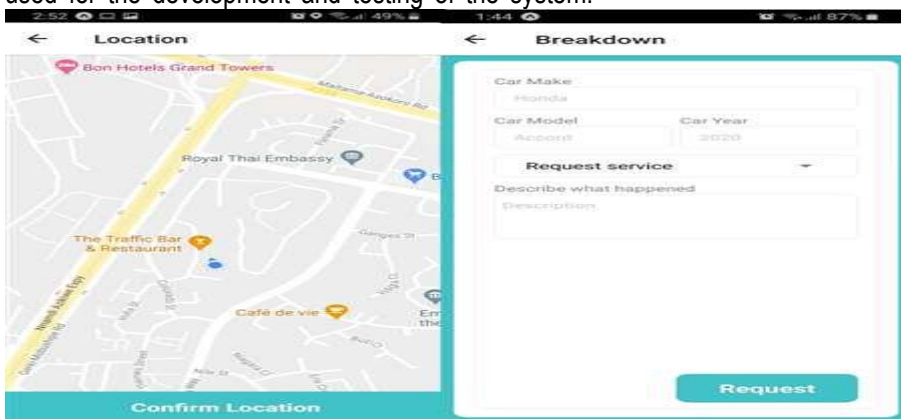


Figure 15: Requesting Breakdown Service

Repair: the repair service collects information such as vehicle make, model, year, part to be fixed, date, and time for the repair to take place. This service is in the form of an appointment.

Figure 16: Requesting Repair

Regular service: The regular service collects information such as vehicle make, model, year, part to be fixed, date, and time for the repair to take place. This service is in the form of an appointment.

Figure 17: Requesting Regular Service

Purchase parts: Just like a conventional ecommerce application the purchase part service serves as an online shop for buying vehicle spare parts with the

option of either home delivery or garage pick up and also payment options of either online payment or pay on delivery.

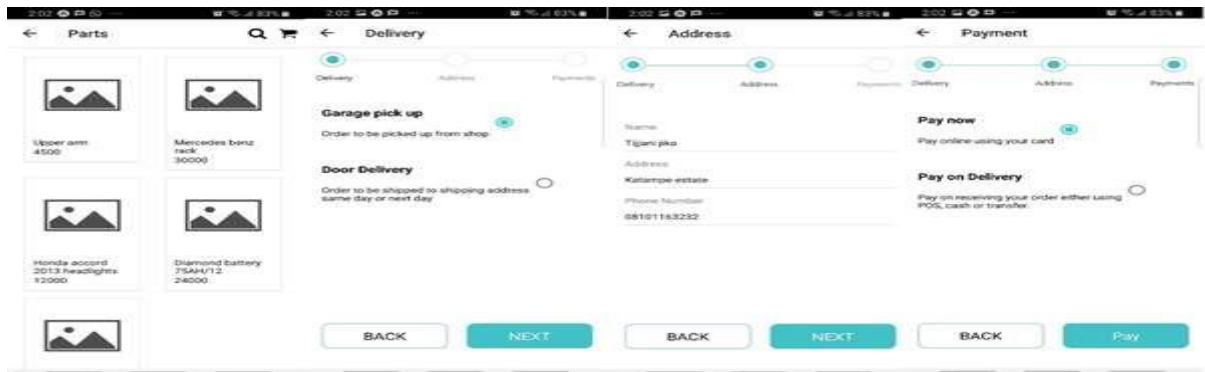


Figure 18: Ordering Parts

Service reminder: the application provides email notification to a user's email on the date of the service entered by the user in the My Vehicle section.

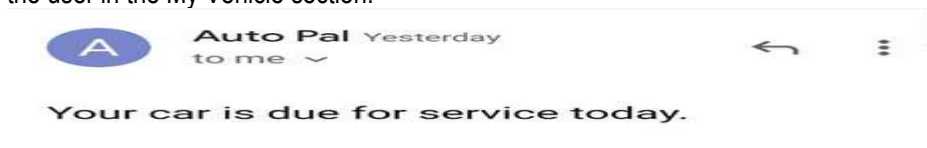


Figure 19: Service Reminder

Get an estimate: An estimate for repairs will be generated by collecting user input of vehicle make, model, and year, the part to be fixed, and a detailed description. This estimate will be generated by a

professional (system admin) who has access to the details of the requested estimate. The admin will provide a detailed breakdown (the parts price, labor, etc.) of the cost of the repair which will be forwarded to the user.

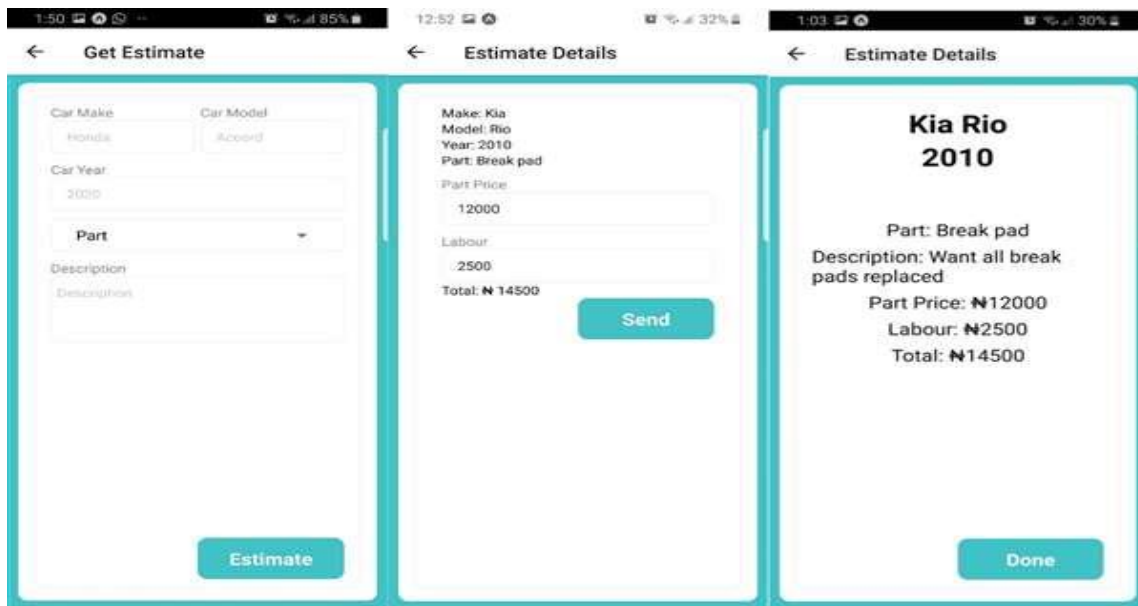


Figure 20: Estimate Requests

Implementation Problems

The first problem encountered is a result of nesting a navigation drawer into the Home Stack Navigator. The problem associated with the navigation drawer was it

not opening on click of the drawer icon; this is associated with the Home Stack Navigator nested in the Auth Stack Navigator.

```

const Drawer = createDrawerNavigator();

const drawerRoutes = () => {
  return (
    <Drawer.Navigator initialRouteName="Home">
      <Drawer.Screen name="Home" component={HomeStackNavigator} />
      <Drawer.Screen name="Profile" component={Profile} />
      <Drawer.Screen name="My Orders" component={MyOrders} />
    </Drawer.Navigator>
  );
};

const HomeStackNavigator = ({ navigation }) => {
  return (
    <HomeStackNavigator initialRouteName="Home">
      <HomeStack.Screen name="Home" component={drawerRoutes}
        options={{
          headerStyle: {
            backgroundColor: "#40C2C7",
          },
          title: "Choose your service",
          headerTintColor: "fff",
          headerLeft: () => {
            <Icon.Button name="ios-menu" size={25}
              backgroundColor="#40C2C7" onPress={() => navigation.openDrawer()} />
          }
        }}
      />
    </HomeStackNavigator>
  );
};

```

Figure 21: Drawer Implementation Problem

The second implementation problem is with the notification. The application was developed using Expo which is a platform for making native Android, IOS, and web applications with JavaScript, react, and react native. At the time of developing this system, React Native Push Notifications on the Expo application are triggered only when the application is running.

The third implementation problem is located in the RegularServiceRequest.js. The problem is located in useEffect which is supposed to get user-requested service details, PUT was passed into the method because service details are not requested from Firestore.

```

useEffect(() => {
  setLoading(true);
  var requestOptions = {
    method: 'PUT',
    redirect: 'follow'
  };

  fetch("https://us-central1-auto-pal.cloudfunctions.net/getServiceRequests", requestOptions)
    .then(response => response.json())
    .then(result => {
      servicesData = result.data;
      setServices(servicesData);
    })
    .catch(error => console.log('error', error))
    .finally(() => {
      setLoading(false)
    });
});

return () => {

```

Figure 22: Service Requests Implementation Problem

Overcoming Implementation Problems

The problem with the drawer navigation was solved by creating a menu screen and adding navigation to the menu screen with the click of the menu icon.

```

const HomeStackNavigator = ({ navigation }) => {
  return (
    <HomeStack.Navigator initialRouteName="Home">
      <HomeStack.Screen name="Home" component={Home}
        options={({ navigation }) => ({
          headerStyle: {
            backgroundColor: "#48C2C7",
          },
          title: "Choose your service",
          headerTintColor: "#fff",
          headerLeft: () => (
            <Icon.Button name="ios-menu" size={25}
              backgroundColor="#48C2C7" onPress={() => navigation.navigate("Menu")}></Icon.Button>
          )
        })
      />
    </HomeStack.Navigator>
  );
};

```

Figure 23: Drawer Implementation Solution

The second implementation problem was overcome by integrating Firebase cloud functions with email notifications as opposed to push notifications.

```

await admin.firestore()
  .collection("cars").where("date", "==", dateFormatted)
  .get()
  .then(async (querySnapshot) => {
    const promises = [];

    querySnapshot.forEach(async doc => {
      const user = await admin.auth().getUser(doc.data().userId);
      const email = user.email;
      const mailOptions = {
        from: 'Auto Pal <carautopal@gmail.com>',
        to: email,
        subject: 'SERVICE REMINDER',
        html: '<p style="font-size: 16px;">Your car is due for service today.</p>'
      };

      promises
        .push(transporter.sendMail(mailOptions));
    });
  });

```

Figure 24: Notification Implementation Solution

The third implementation problem was navigated through bypassing the GET method.

```

useEffect(() => {
  setLoading(true);
  var requestOptions = {
    method: 'GET',
    redirect: 'follow'
  };

  fetch("https://us-central1-auto-pal.cloudfunctions.net/getServiceRequests", requestOptions)
    .then(response => response.json())
    .then(result => {
      servicesData = result.data;
      setServices(servicesData);
    })
    .catch(error => console.log('error', error))
    .finally(() => {
      setLoading(false)
    });
});
return () => {

```

Figure 25: Service Requests Implementation Solution

Testing

Tests are to be conducted on the system with the motive of tracking the availability of errors.

Tests Plans**Table 1: Test Plans**

Req. No.	Description	Type Case Id
R-101	The system shall allow a new User to create an account.	T_01
R-102	The system shall allow a User to log in to the system.	T_02
R-103	The system shall allow a user to enter his/her vehicle details.	T_03
R-104	The System shall allow a User to request a breakdown.	T_04
R-105	The system should be able to collect User location.	T_05
R-106	The system shall allow a User to book a repair.	T_06
R-107	The system shall allow a User to book a service.	T_07
R-108	The system shall allow a User to request an estimate.	T_08
R-109	The system should allow a User to purchase parts.	T_09
R-110	The system should be able to get service dating using mileage details	T_10
R-111	The system should send notifications when a User's vehicle is set for service.	T_11
R-112	The system should allow a User to view order details.	T_12
R-113	The system should allow a User to view breakdown details.	T_13
R-114	The system should allow a User to view booking details.	T_14
R-115	The system should allow users to make payments online.	T_15
R-116	The system should allow the admin to view estimate request details.	T_16
R-117	The system should allow the admin to view booked repair details.	T_17
R-118	The system should allow the admin to view booked service details.	T_18
R-119	The system should allow the admin to view breakdown details.	T_19
R-120	The system should allow the admin to view order details.	T_20
R_121	The system should allow the Admin to send an estimated price to the user.	T_21
R_122	The system should allow the Admin to add a product	T_22
R_123	The system shall run on iOS 10.0 and Android 4.1 or later.	T_23

Test Traceability Matrix**Table 2: Traceability Matrix**

Type Case Id	Description	Priority	Test Date	Test Result
T_01	The system shall allow a new User to create an account.	H	13/09/2020	PASS
T_02	The system shall allow a User to log in to the system.	H	13/09/2020	PASS
T_03	The system shall allow a user to enter his/her vehicle details.	H	13/09/2020	PASS
T_04	The System shall allow a User to request a breakdown.	H	13/09/2020	PASS
T_05	The system should be able to collect User location.	H	13/09/2020	PASS
T_06	The system shall allow a User to book a repair.	H	13/09/2020	PASS

T_07	The system shall allow a User to book a service.	H	13/09/2020	PASS
T_08	The system shall allow a User to request an estimate.	H	13/09/2020	PASS
T_09	The system should allow a User to purchase parts.	H	13/09/2020	PASS
T-10	The system should be able to get service dates using mileage details	L	13/09/2020	FAIL
T_11	The system should send notifications when a User's vehicle is set for service.	H	13/09/2020	PASS
T_12	The system should allow a User to view order details.	H	13/09/2020	PASS
T_13	The system should allow a User to view breakdown details.	H	13/09/2020	PASS
T_14	The system should allow a User to view booking details.	H	13/09/2020	PASS
T_15	The system should allow users to make payments online.	H	13/09/2020	PASS
T_16	The system should allow the admin to view estimate request details.	H	13/09/2020	PASS
T_17	The system should allow the admin to view booked repair details.	H	13/09/2020	PASS
T_18	The system should allow the admin to view booked service details.	L	13/09/2020	PASS
T_19	The system should allow the admin to view breakdown details.	L	13/09/2020	PASS
T_20	The system should allow the admin to view order details.	H	13/09/2020	PASS
T_21	The system should allow the Admin to send an estimated price to the user.	H	13/09/2020	PASS
T_22	The system should allow the Admin to add a product	H	13/09/2020	PASS
T_23	The system shall run on iOS 10.0 and Android 4.1 or later.	H	13/09/2020	PASS

Test Report Summary

Table 3: Test Report

Summary of Test Vehicle Ride Out	Results
Number of tests executed:	23
Number of tests not executed:	0
Number of tests passed:	22
Several tests failed:	1
Percentage of tests passed:	96%
Percentage of tests failed:	4%

Use Guide

The user guide contains instructions on how the intended users of the software can use it with ease to achieve their objectives. The user guide also serves as a tool for referencing if users encounter difficulties while using the software.

Summary

The section provides details on the implementation of the system, technologies used in the implementation of the system, and problems encountered in the course of implementing the system. The section also covers the various tests executed on the system to track the availability of errors.

Discussion**Overview**

This system was developed to automate vehicle-related services to enhance transparency and efficiency. This paper elaborates on the evaluation of the system from an objective viewpoint through adequate testing and a subjective viewpoint through an appraisal. It also talks about the limitations of the research as well as the challenges encountered. Future enhancements are to be made to the research. Finally, recommendations were made followed by a conclusion.

Objective Assessment

The application has achieved some of its main goals, the application;

i. **USER APPLICATION**

- The application allows a user to register with the system.
- The application allows a user to log in to the system.
- The application allows users to add vehicle details and send notifications when service is due.
- The application allows a user to request a breakdown service.
- The application allows a user to book for service and repairs.
- The application allows a user to request a repair estimate.
- The application allows a user to purchase parts.
- The application allows a user to make online payments for parts purchased.
- The application allows a user to view breakdown request details.
- The application allows a user to view booked service and repair details.

The application allows a user to view estimate requested details.

The application allows a user to view order details.

The application sends service notifications to the user.

ii. **ADMIN APPLICATION**

The application allows an admin to add products.

The application allows an admin view breakdown request.

The application allows an admin to view service and repairs booked.

The application allows an admin view breakdown estimate requested.

The application allows an admin to view and send repair estimates.

The application allows an admin to view orders made.

Limitations and Challenges

During the course of the study, the researcher was faced with some limitations and constraints which has impeded the progress of the project. Such limitations are:

- i. Financial constraint: Insufficient fund tends to impede the efficiency of the researcher in sourcing the relevant materials, hardware, software, literature, or information and in the process of data collection through the Internet and interviews.
- ii. Time constraint: The researcher simultaneously engaged in his studies with other academic-related work which consequently, cut down on the time devoted to the research work.
- iii. Global pandemic (coronavirus): With the movement restrictions in place, data collection was not properly and sufficiently obtained.

Enhancements

The vehicle maintenance and services system are vast application with room for improvement, such improvements are;

- User should be able to include more than one vehicle.
- The system should book a regular service when a vehicle's service is due.
- The system should track arriving help on the map when a user requests a breakdown service.
- The system should be able to track order delivery.

Recommendations

This paper titled Design and Implementation of Vehicle Maintenance and Services System recommends:

1. The full adoption and implementation of this system as an integrated enterprise software in the country for evaluation purposes and success quantifiable matrix visibility.
2. Collecting the statistics about vehicles (e.g., frequency of maintenance, period of usage, parts prone to malfunction, etc.) enables organizations and researchers to do some machine learning research to improve the method of design and development of the auto parts.

Conclusion

The research undertaken was aimed at implementing a simple vehicle maintenance and services system. The study sought to design a solution to vehicle management by observing and interviewing the users of the system. An analysis of the manual system was carried out to discover the prerequisite conditions for the proposed system. The proposed system provides a user-friendly platform for service reminders, booking for repairs and servicing, requesting estimates, etc. The proposed system was developed using React Native. The Firebase database was used for the database design. The actual coding was done using NodeJS. The system has reached a steady state where all bugs have been eliminated. The system would be operated at a high level of efficiency and all the users associated with the system understand its advantage.

References

- Adesoji, B. D. (2019, March 11). *Number of vehicles on Nigerian roads hit 11.8m in a year*. Retrieved March 10, 2020, from Nairametrics: <https://nairametrics.com/2019/03/11/data-on-the-total-number-of-on-nigerian-roads/>
- Aganga, O. (2013). *The New Auto Policy is Ready*. Retrieved May 7, 2020, from The Punch Nigeria: <http://www.punchng.com/business/industry>
- Akaigwe. (2010). *Auto industry @ 50. Nigerian Journal of Engineering Science*.
- Chamberlain, O. (2013). *History of Automobile Past and Present Challenges Facing. IOSR Journal of Research & Method in Education (IOSRJRME)*, 11-16.
- Collins, D. (2018, January 16). *Vehicle Bibles*. Retrieved April 23, 2020, from Vehicle Bibles: <https://www.vehiclebibles.com/vehicle-maintenance-tips/>
- Gongolidis, V., Kommata, M., Papantoniou, A., & Loumos, V. (2012). *Semantic Driven Location Services in the Web of Data. 2012 16th Panhellenic Conference on Informatics*, 7-12. <https://doi.org/10.1109/PCi.2012.17>.
- Izogo, E. E., & Ogba, I.-E. (2016). *Service quality, customer satisfaction, and loyalty in the automobile services sector. International Journal of Quality & Reliability Management*.
- Okuhit, J. (2015). *Sweet New Vehicles. Huge Maintenance Cost*.
- Prototyping Model in Software Engineering: Methodology, Process, Approach*. (2017). Retrieved April 16, 2020, from Guru99.com: <https://www.guru99.com/software-engineering-prototyping-model.html#6>
- Top 12 Software Development Methodologies & its Advantages / Disadvantages*. (2015, April 15). Retrieved May 5, 2020, from Custom Software Development & Enterprise Mobile Apps: <https://www.tatvasoft.com/blog/top-12-software-development-methodologies-and-its-advantages-disadvantages/>
- Zhi, J., & Garousi, V. (2013). *On Adequacy of Assertions in Automated Test Suites: An Empirical Investigation. 2013 IEEE Sixth International Conference on Software Testing, Verification and Validation Workshops*, 382-391. <https://doi.org/10.1109/ICSTW.2013.49>.