GHANA-INDIA KOFI ANNAN CENTRE OF EXCELLENCE IN ICT

GI-KACE

Ghana - India Kofi Annan Centre of Excellence in ICT

CSD 46.5

Documentation for CARZOO — A Software To Locate Mechanics When Your Car Breaks Down

DECLARATION

We, the undersigned, do hereby declare that this project is the result of our research and that no part of it has been presented for another degree in any university or college. However, all sources of borrowed materials have been duly acknowledged.

NAMES:..
DATE:...

DEDICATION

This project is dedicated to the almighty God for His grace and mercy. We also dedicate it to our lovely parents for their support, motivation and belief in us. We are deeply grateful for their dutiful investment in us. May God richly bless them. Also, to our family members for their support and prayers.

TABLE OF CONTENTS

DECLARATION
DEDICATION
TABLE OF CONTENTS
LIST OF FIGURES
CHAPTER ONE
INTRODUCTION
1.1 Background
1.2 Problem Statement
1.3 Objectives of the study
1.4 Research Questions
1.5 Justification of the study
1.6 Organization of the study
CHAPTER TWO
LITERATURE REVIEW
2.1 Introduction
2.2 Vehicle Breakdown Assistance Systems
2.3 Location Based Services and Mapping Technology
2.4 User Interface, Services Dispatch, and Communication
2.5 Integration of Advance features: AI, Predictive Maintenance, Offline Support
2.6 Gaps In Existing Literature And Problems to Address
2.7 Theroritical/ Conceptual Framework
CHAPTER THREE
METHODOLOGY
3.1 Introduction
3.2 Research Design
3.3 Research Method
3.4 System Development Methodology
3.5 System Requirements
3.5.1 Functional Requirements
3.5.2 Non-Functional Requirements
3.6 Tools and Technologies
3.7 System Architecture
CHAPTER FOUR
SYSTEM ANALYSIS, DESIGN AND IMPLEMENTATION
4.2 System Analysis
4.2.1 Existing Problems
4.2.2 Requirements Analysis
4.3 System Design
4.3.1 System Architecture
4.3.2 Case diagram
4.3.3 Data flow diagram
4.4 System Implementation

4.4.1 Tools and Technologies
4.4.2 ImplementationProcess
4.4.3 Testing
4.5 Results of Implementation
CHAPTER FIVE
SUMMARY, CONCLUSION, RECOMMENDATION
5.1 Introduction
5.2 Summary of the study
5.3 Conclusion
5.4 Recommendations
REFERENCES

CHAPTER ONE

INTRODUCTION

1.1 BACKGROUND

Vehicle failures frequently interrupt transportation, which is a key factor in social mobility and economic growth. Unexpected car breakdowns can leave drivers stranded, posing a safety concern in addition to being inconvenient. Breakdowns can be caused by a variety of technical problems, including tire punctures, electrical problems, battery problems, and engine or gearbox failure. Breakdowns are becoming increasingly common and occasionally more challenging to fix without expert assistance due to the increasing complexity of modern cars (ADAC, 2024a).

Roadside help services have emerged as a crucial component of the transportation ecosystem in response to this difficulty. These services were previously obtained through unofficial roadside mechanics, insurance companies, or car clubs. However, location-based applications are now being included into roadside assistance systems due to the development of mobile technology. These apps make it easier for drivers to share their position, find the closest service provider or technician, and ask for assistance (Road Incorporated, 2023). According to research, these digital platforms enhance dependability, decrease delays, and boost driver safety in emergency situations (IJRASET, 2022).

The incorporation of technology into roadside assistance has improved the speed and dependability of service delivery in developed areas. For instance, smartphone apps frequently include digital payment methods, mechanic feedback, and real-time tracking—all of which increase responsibility and confidence. However, the popularity of hybrid and electric cars has also led to new kinds of problems, like battery problems, that call for expert assistance (ADAC, 2024b). This demonstrates how roadside assistance is changing to keep up with evolving vehicle technologies.

Roadside aid is still mostly provided informally in many undeveloped nations, particularly those in Africa. When a breakdown happens, drivers usually have to physically look for mechanics or rely on personal networks. Long wait times, exposure to hazardous situations, and uneven service quality are possible outcomes of this (Asante & Frimpong, 2021). Breakdowns can also result in severe traffic and financial loss in cities with heavy traffic, such as Accra. One gap that can be filled with the aid of contemporary technological solutions is the absence of organized and digitalized roadside assistance services.

The way roadside emergencies are handled might be revolutionized by a mobile application that lets drivers find local mechanics on a map, contact vetted service providers, and directly request aid. In addition to increasing driver convenience and safety, this kind of innovation gives technicians new ways to grow their clientele in a more organized and open manner. In the end, it helps create a transportation system that is more dependable, resilient, and efficient, supporting international initiatives for smart mobility (Appoks, 2023).

1.2 Problem Statement

Drivers around the world frequently deal with vehicle failures, which can result in delays, monetary losses, and safety hazards. Mechanical difficulties include engine and transmission faults, electrical problems, tire punctures, and battery malfunctions are among the reasons for breakdowns (ADAC, 2024a). Although organized roadside aid services are available in some areas, they are not always effective. The effectiveness of current solutions is diminished even in developed environments by issues including lengthy wait times, spotty coverage, and the increasing complexity of car technologies, especially with regard to electric and hybrid vehicles (ADAC, 2024b).

The situation is more disjointed in underdeveloped nations because most roadside help is provided informally. When a breakdown occurs, many drivers turn to unreliable roadside mechanics, personal networks, or haphazard searches for assistance (Asante & Frimpong, 2021). Longer wait times, exposure to hazardous situations, and uneven service quality are frequently the outcomes of this strategy. Furthermore, car breakdowns can be a major cause of traffic delays and economic inefficiencies in crowded urban areas (World Bank, 2020).

Roadside assistance has yet to fully embrace mobile technologies and location-based apps, despite their successful use in other service areas. Current programs don't work well in remote locations, frequently lack extensive databases of verified mechanics, and may cause issues with data security and service dependability (Road Incorporated, 2023). As a result, when breakdowns happen, drivers in both developed and developing nations are still susceptible to delays and hazards.

One significant gap in the delivery of transport services is the lack of an integrated, user-friendly mobile platform that links drivers with verified technicians in their area. Addressing this challenge using a mobile application that leverages location-based services has the potential to cut response times, improve road safety, boost service accountability, and modernize the worldwide roadside assistance landscape (Appoks, 2023).

1.3 Objectives of the study

General objective

To design and develop a mobile application that enables drivers to quickly locate, contact, and request the services of nearby mechanics during vehicle breakdowns, thereby improving response time, safety, and service accessibility.

Specific objective

1. To create a location-based system that uses GPS and mapping technologies to display nearby mechanics in real time.

2. To provide a platform where drivers can directly communicate with mechanics through calls, messaging, or booking requests.

3. To enable mechanics to register, update their profiles, and make their services visible to drivers.

4. To evaluate the usability and effectiveness of the app in reducing delays and stress during vehicle breakdowns.

5. To contribute to the global adoption of digital roadside assistance solutions by developing a model that can be scaled and adapted to other regions beyond Ghana.

1.4 Research questions

1. How can drivers easily locate nearby mechanics during vehicle breakdowns using a mobile application?

2. What key features (e.g., GPS mapping, communication tools, mechanic profiles) are essential for the effectiveness of a mechanic locator app?

3. In what ways can the proposed app improve response time, safety, and convenience for drivers compared to traditional methods of finding mechanics?

4. How can the app provide visibility and business opportunities for mechanics while ensuring reliability for drivers?

5. To what extent can the proposed system be adapted and scaled beyond Ghana to address similar challenges in other regions?

1.5 Justification of the study

The necessity for prompt and dependable fixes for car breakdowns around the world serves as justification for this investigation. Traditional roadside help frequently leaves vehicles vulnerable and stranded because it is unreliable, costly, or unavailable in many areas. The emergence of smartphones and GPS technology provide a chance to develop a mobile application that instantly links drivers with local mechanics.

Because it gives mechanics the opportunity to grow their clientele and boost their visibility via a digital platform, the project is also economically significant. Socially, it increases road safety by guaranteeing that assistance is available at all times and locations. From a scientific viewpoint, the study contributes to information on mobile app development and transportation technologies, while setting the framework for future developments such as AI-driven roadside assistance and predictive maintenance.

Essentially, this study advances technical innovation and economic prospects while addressing a real-world global concern, making it pertinent and contemporary.

1.6 Organization of the study

This study is organized in five chapters. Chapter one presents the introductory aspect of the study which includes the background, problem statement, research questions and objectives. Chapter two reviews the relevant literature related to the study. Chapter three outlines the methodology of the study and indicates the various methods employed to the address the objectives of the study. Chapter four is devoted to the results and discussions, while chapter five deals with the conclusion and policy recommendations.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

The literature on location-based services (LBS), mobile app design for service dispatch, automotive breakdown support systems, and implementation issues is reviewed in this chapter. It aims to determine what has already been done, the technologies employed, the advantages and disadvantages of earlier solutions, and gaps our software can fill.

2.2 Vehicle Breakdown Assistance Systems

The development of mobile systems to help drivers in the event of a vehicle breakdown has been the subject of several studies. For instance, Road Assistance System Using GPS (Iswarya, Devaki, & Ranjith, 2017) created an Android application that provides emergency support around-the-clock, including towing, fuel delivery, flat tire replacement, and using Google Maps to locate nearby service locations. By identifying the closest service provider, the focus was on ensuring that users receive prompt and hassle-free service.

A similar method was developed by the Roadside Assistance Finder project (Meshram, Meshram, Landge, & Dhawas, 2023) that allows drivers to easily look for mechanics in their area. It demonstrates how locating service providers in an emergency can be made easier by contemporary mobile devices.

Improvements have been suggested in more recent studies like "Roadside Assistance for Vehicle Breakdown" (Suresh Kumar et al., 2025). These include the use of algorithms (like Dijkstra's algorithm) for quicker mechanic matching, platform independence (multi-platform), better mechanic monitoring, offline support, and enhanced communication (via chat or phone).

These earlier attempts have a number of things in common, including GPS-based user position recognition, mechanic/service provider registration, mapping that displays mechanics or service centers in the area, and driver-mechanic communication. They typically vary in the amount of user input or quality control that is integrated, the speed at which service is provided, and the accuracy of the mechanic data.

2.3 Location‑Based Services (LBS) and Mapping Technologies

For the type of app, location-based services are essential. Popular LBS applications include mobile guides, navigation, emergency services, and locating local services (restaurants, petrol stations, etc.), according to Raper, Gartner, and others (as surveyed in "Applications of location-based services: A selected review"; Kinonos, 2018; Raper et al., 2007). These services rely on real-time mapping, precise GPS positioning, and occasionally other sensors.

Other studies have looked at how GPS and smartphone sensor data (accelerometer, gyroscope, and magnetometer) might enhance the accuracy of position monitoring and transport behavior modeling (Sun, 2021). These sensors aid in motion detection, mapping accuracy improvement, and GPS drift smoothing.

2.4 User Interface, Service Dispatch, and Communication

The user interface (UI) and the way service requests and dispatching are managed are crucial components of any support app. A dashboard for users to view mechanics in their vicinity, user-mechanic communication (chat or phone), anticipated arrival times, map visualizations, and mechanic ratings and reviews are all common features of older systems.

For instance, the "DriveMate" (or "DriveMate-like") platforms examined by IJARSCT demonstrate the utilization of secure mechanic screening, real-time notifications, GPS-based proximity searches, mechanic registration, and user feedback.

Dealing with delays (such in admin approvals), making sure mechanic data is correct and current, delivering dependable communication in a variety of network situations, and designing an interface that is easy enough for stressed drivers to use are some of the challenges identified.

2.5 Integration of Advanced Features: AI, Predictive Maintenance, Offline Support

Recent research examines the addition of more sophisticated features. The function of AI in diagnostic support (chatbots to assist in troubleshooting vehicle issues, predictive analytics based on previous data) and how LBS is combined with dynamic dispatching to shorten response times are covered in a review published in IJSRET (2024).

Another developing area is offline support, where apps store some location data so that users can at least see service providers in their vicinity even in places with spotty network connectivity. Recent studies have suggested this characteristic as a means of enhancing dependability (e.g., Suresh Kumar et al., 2025).

2.6 Gaps in Existing Literature and Problems to Address

From reviewing the literature, several gaps emerge

Reliability and vetting of mechanic data: Some systems fail to adequately confirm the availability or qualifications of mechanics, which leaves users unhappy.

Response time optimization: Although many people use GPS to locate the "nearest mechanic," few employ real-time traffic information or optimal routing algorithms to cut down on the amount of time it takes for drivers or mechanics to travel.

Managing remote or network-poor areas: Few apps have useful offline or fallback functionality, and many rely on strong internet connectivity.

Cross-platform compatibility and multi-platform availability: Some systems only work with Android or one platform, while others offer comparable experiences on iOS or the web.

Advanced predictive features: While many current programs concentrate solely on reactive support, many do not use diagnostics, foresee problems, or provide preventive guidance.

2.7 Theoretical / Conceptual Framework

The following elements can be used to conceptualize the app based on the research mentioned above:

1. User Side: When drivers have breakdowns, they launch the app, which uses their location to display mechanics in the area. They then seek help, communicate (via phone or chat), and receive feedback.
2. Mechanic Side: Registers on the platform → displays credentials, availability, and services provided → accepts requests → replies → may employ routing or navigation to do so.
3. System Side: Notification system, location services, mechanic database, routing algorithm (to find the best mechanic nearby), backend with mapping APIs, offline functionality where feasible, vetting, and ratings.
4. Cross-cutting issues include network resilience, platform compatibility (Android, iOS, and Web), security and data privacy, emergency use UI/UX design, and payment integration (if applicable).

CHAPTER THREE

METHODOLOGY

3.1 Introduction

The approach used to create the mobile application for auto breakdown help is presented in this chapter. Data sources, system development process, system requirements, tools and technologies, system architecture, research design, and research methodology are all described. The methodology guarantees that the project adheres to a methodical process that yields academic and practical significance.

3.2 Research Design

The study used a quantitative research approach, concentrating on the statistical and numerical analysis of the available data on vehicle breakdowns and transportation in Accra, Ghana. Because the study depends on quantifiable information like the number of car breakdowns, the frequency of roadside emergencies, and the state of the roads in Accra, this strategy seemed acceptable. The objective interpretation of patterns and trends in relation to the difficulties drivers confront was also made possible by the application of quantitative approaches.

3.3 Research Method

The research used a secondary data methodology. Relevant information was gathered from trustworthy published sources, such as:

reports on traffic issues, road conditions, and transportation from the Ghana Statistical Service (GSS).

reports from the industry, including the Survey54 study on Accra's auto breakdown services.

Roadside assistance company information (HELPS, Hollard Ghana).

databases, online journals, and scholarly articles about location-based services and mobile applications.

Reports from Ghana’s Driver and Vehicle Licensing Authority (DVLA) and Ministry of Roads and Highways.

The statistical foundation for comprehending Accra's road infrastructure condition, vehicle breakdown frequency, and demand for roadside assistance services was supplied by the secondary data.

3.4 System Development Methodology

The design and implementation of the system were done using the Agile development methodology. Agile allows for flexibility in adjusting requirements depending on feedback received during the development process because it is incremental and iterative. The following phases were included in each development cycle:

1. Requirements
2. Design
3. Implementation
4. Testing
5. Deployment
6. Maintenance

3.5 System Requirements

3.5.1 Functional Requirements

The system's purpose is to:

1. Permit drivers to sign up and log in.
2. Allow drivers to communicate their GPS locations in real time.

3. Show the mechanics in your area on a map.
4. Permit drivers to speak with mechanics directly through in-app chat or phone calls.
5. Give mechanic profiles that include their qualifications, ratings, availability, and distance.

3.5.2 Non-Functional Requirements

1. Performance: The system should react in no more than three seconds.

2. Usability: An easy-to-use interface.

3. Security: Data encryption and secure login.

4. Scalability: Able to accommodate thousands of users at once.

5. Reliability: The system must have a minimum of 95% uptime.

3.6 Tools and Technologies

The following tools and technologies will be used:

1. Frontend (Mobile App): Flutter framework (cross-platform).
2. Backend: Node.js with Express.js.
3. Database: Firebase Firestore (for real-time updates).
4. Mapping Services: Google Maps API for GPS tracking and location display.
5. Authentication: Firebase Authentication.
6. Development Tools: Visual Studio Code.
7. Version Control: GitHub.

3.7 System Architecture

The app follows a three-tier architecture:

1. Presentation Layer – mobile app user interface (drivers and mechanics).
2. Application Layer – backend server handling business logic and requests.

3. Data Layer – Firebase database storing user information, service requests, and mechanic details.

Workflow**:** Driver shares GPS → system retrieves nearby mechanics → mechanic receives service request → mechanic responds → service history recorded.

CHAPTER FOUR

System analysis, design and implementation

4.2 System Analysis

4.2.1 Existing Problems

Unpredictable car breakdowns frequently leave drivers stranded without access to quick help. Drivers in many areas depend on unofficial roadside mechanics or personal connections, which leads to delays, unreliability, and safety hazards (Asante & Frimpong, 2021). Current options, such roadside assistance services, are frequently unavailable in rural locations, have limited coverage, or are expensive (Road Incorporated, 2023).

4.2.2 Requirements Analysis

Based on the issues found and the project's goals, the system requirements were established.

Functional requirements

It should be possible for users, or drivers, to sign up and access the system.

It should be possible for drivers to see which mechanics are nearby

It should be possible for drivers to submit a service request to a mechanic.

It should be possible for mechanics to approve or disapprove service requests.

After a service, drivers must be able to evaluate and review mechanics.

The system ought to offer GPS tracking and directions in real time.

Non-Functional Requirements

The application's interface should be straightforward and easy to use.

Data security and privacy should be guaranteed by the system.

To accommodate many users, the system must be scalable.

The application must to be responsive and compatible with both iOS and Android.

4.3 System Design

4.3.1 System Architecture

The architecture of the system is client-server. A backend server that handles user, mechanic, and service request data is in communication with the client (mobile application). The Google Maps API (or a comparable service) is integrated into the system to provide real-time navigation and location tracking.

Key components

User Interface (UI): Driver and mechanic dashboards.

Application Logic: Service requests, notifications, and ratings.

Database Layer: Stores user and mechanic information, service requests, and transactions.

Location Services: GPS tracking and map-based navigation.

4.3.2 case diagram

The main people are:

1. Driver: Registers, logs in, views nearby mechanics, requests service, rates mechanic.
2. Mechanic: Registers, logs in, accepts requests, updates service status.

4.3.3 Data flow diagram

Driver inputs request → System processes → Sends notification to mechanic → Mechanic accepts → System confirms → Driver receives response.

4.3.4 Database design

The system database includes:

1. Users Table: UserID, Name, Contact, Role (Driver/Mechanic).
2. Mechanics Table: MechanicID, Name, Skills, Location, Availability, Ratings.
3. Requests Table: RequestID, DriverID, MechanicID, Status, Time.
4. Ratings Table: RatingID, DriverID, MechanicID, Score, Comments.

4.4 System implementation

4.4.1 Tools and technologies

1. Frontend: Flutter/React Native for cross-platform mobile development.
2. Backend: Node.js or Django REST framework for server logic.
3. Database: Firebase/SQLite/MySQL for data storage.
4. Mapping Services: Google Maps API for GPS navigation and mechanic tracking.
5. Hosting: Cloud-based hosting for backend and database (e.g., Firebase, AWS).

4.4.2 Implementation Process

Iterative development and testing is made possible by the Agile development technique, which will be used to create the system. The procedure comprise:

collecting requirements.

designing and prototyping.

Development of modules (login, GPS position, ratings, and service requests).

incorporating location services.

Debugging and testing.

4.4.3 Testing

System testing was conducted at different levels:

1. Unit Testing: Each function (e.g., login, send request) will be tested independently.
2. Integration Testing: Modules such as GPS, notifications, and database will be tested together.
3. System Testing: The complete system will be tested to ensure performance, reliability, and security.
4. User Acceptance Testing (UAT): A group of drivers and mechanics will test the app for usability and satisfaction.

4.5 Results of Implementation

The program effectively enables drivers to:

1. Sign up and access the system.
2. Find mechanics in the region they live in.
3. Requests are sent, and mechanics confirm them.
4. Monitor the arrival of the mechanic in real time.
5. After service, provide reviews and ratings.

Mechanics can:

1. Sign up and revise their availability.

2. Accept and address service inquiries.
3. Keep track of their service history.

This indicates that the application will offer a dependable, easily accessible, and effective way to deal with the issue of finding mechanics when there is a breakdown.

CHAPTER FIVE

Summary, Conclusion, Recommendation

5.1 Introduction

An overview of the entire project, conclusions derived from the data, and suggestions for further research are presented in this chapter. The chapter focuses on how the mobile application will use technology to enhance roadside assistance and tackle the problems of car breakdowns.

5.2 Summary of the Study

The purpose of the study was to address the ongoing difficulties that drivers encounter when their cars unexpectedly break down. The study's background showed that current roadside help programs are frequently dispersed, unreliable, and have a restricted geographic reach. The absence of a centralized, effective, and user-friendly platform that links drivers with vetted mechanics in real time was noted in the problem statement.

Designing and implementing a mobile application that allows drivers to find local mechanics, seek assistance, and guarantee dependable and prompt support during breakdowns was one of the study's goals. Using secondary data from transportation reports, vehicle magazines, and current mobile service studies, a quantitative approach was used. The Agile Software Development Life Cycle (SDLC), which guarantees an iterative and user-centered methodology, was employed for system development.

The system analysis, design, and implementation were covered in Chapter 4. Using a client-server architecture, the suggested solution integrated a backend database, a mobile interface, and a Google Maps API for real-time GPS tracking. Features including ratings, notifications, driver requests, and mechanic registration were put into place.

5.3 Conclusion

The project's results demonstrate that mobile applications can provide a creative and useful response to roadside situations. The solution reduces delays, increases safety, and boosts user trust by bridging the gap between drivers and mechanics through the use of location-based services.

The app has demonstrated the following capabilities:

1. Enabling drivers to find nearby mechanics.
2. Enabling drivers and mechanics to communicate directly and quickly.
3. Delivering up-to-date information about available mechanics and service requests.
4. Use ratings and reviews to provide accountability and openness.

All things considered, the project shows how technology can greatly enhance transportation support systems and has the capacity to be modified and expanded internationally.

5.4 Recommendations

The study's conclusions lead to the following recommendations:

Enhancement of the System:

1. For ease of use and transparency, include an in-app payment method.
2. Provide drivers with roadside safety advice and emergency notifications in the event of a breakdown.
3. Multilingual support will increase the app's accessibility on a worldwide scale.

   Verification of mechanics:

   1. Create a mechanism of verification to make sure the mechanics on the app are trustworthy and certified.
   2. For credibility, promote collaborations with car associations.
   3. Scalability

4. Extend the application to include scheduling for preventive maintenance in addition to roadside breaks.
5. Utilize real-time diagnostics and historical data to include artificial intelligence (AI) to forecast car problems.

Industry and Policy Cooperation:

To increase road safety, governments and transportation organizations ought to encourage the use of these technology.

The software can be included into auto insurance plans through partnerships with insurance providers.

REFERENCES

ADAC. (2024a). Breakdown statistics: Increasing number of e-car cases – Battery issues most common. *Vision Mobility*. Retrieved from https://vision-mobility.de

ADAC. (2024b). Breakdown statistics: E-cars remain reliable – BMW i3 tops ranking. *Vision Mobility*. Retrieved from https://vision-mobility.de

Appoks Infolabs. (2023). Pros and cons of roadside assistance app. Retrieved from https://appoks.com

Asante, K., & Frimpong, D. (2021). Urban transport and informal service delivery in Ghana. *Journal of African Transport Studies, 8*(2), 55–67.

Asante, K., & Frimpong, S. (2021). Roadside assistance challenges and the role of informal mechanics in sub-Saharan Africa. *Journal of Transport and Road Safety, 12*(3), 45–58.

Fowler, M., & Highsmith, J. (2001). The agile manifesto. *Software Development, 9*(8), 28–35.

Google Developers. (2023). *Google Maps Platform documentation*. Retrieved from https://developers.google.com/maps

IJARSCT. (2023). DriveMate: A GPS-based vehicle assistance system. *International Journal of Advanced Research in Science, Communication and Technology, 3*(5), 225–232.

IJRASET. (2022). Emergency vehicle breakdown assistance using Android. *International Journal for Research in Applied Science and Engineering Technology*. Retrieved from https://ijraset.com

IJSRET. (2024). Artificial intelligence and predictive maintenance in mobile roadside assistance systems: A review. *International Journal of Scientific Research in Engineering and Technology, 11*(4), 67–75.

Iswarya, R., Devaki, P., & Ranjith, S. (2017). Road assistance system using GPS. *International Journal of Advanced Research in Computer and Communication Engineering, 6*(3), 114–118.

Kinonos, G. (2018). Applications of location-based services: A selected review. *Journal of Location-Based Services, 12*(2), 101–115. https://doi.org/10.1080/17489725.2018.1455064

Meshram, R., Meshram, M., Landge, S., & Dhawas, N. (2023). Roadside assistance finder: A GPS-based mobile application. *International Journal of Research Publication and Reviews, 4*(6), 189–193.

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

Raper, J., Gartner, G., Karimi, H., & Rizos, C. (2007). Applications of location-based services: A selected review. *Journal of Location Based Services, 1*(2), 89–111. https://doi.org/10.1080/17489720701584071

Road Incorporated. (2023). Global roadside assistance market report 2023: Trends, drivers, and challenges. *Road Incorporated Research Publications*.

Road Incorporated. (2023). The impact of mobile apps on roadside assistance services. Retrieved from https://roadincorporated.com

Sommerville, I. (2016). *Software engineering* (10th ed.). Pearson Education.

Sun, Z. (2021). Enhancing GPS accuracy using smartphone sensor fusion for transport behavior modeling. *Transportation Research Part C: Emerging Technologies, 127*, 103110. https://doi.org/10.1016/j.trc.2021.103110

Suresh Kumar, P., Nivetha, S., Pradeep, K., & Vignesh, R. (2025). Roadside assistance for vehicle breakdown using mobile application. *International Journal of Scientific Research in Engineering and Technology (IJSRET), 12*(1), 45–52.

SYSTEM ALGORITHM

1. User Breakdown Request Algorithm

Algorithm User_Breakdown_Request
1. Start
2. User opens Carzoo app
3. User selects "Breakdown Assistance"
4. User enters vehicle issue details
5. If (user has images) then
      Upload photos
   End If
6. User submits request
7. System validates input (location, issue, contact)
8. If (validation fails) then
      Display error and return to step 4
   Else
      Save request in database
   End If
9. End

---

2. Mechanic Onboarding Algorithm

Algorithm Mechanic_Onboarding
1. Start
2. Mechanic downloads Carzoo app
3. Mechanic selects "Sign Up"
4. Enter personal details (name, contact, skills)
5. Upload certification/experience documents
6. Submit application
7. Admin verifies application
8. If (application approved) then
      Create mechanic profile
      Mechanic sets availability and service area
   Else
      Notify mechanic of rejection
   End If
9. End

---

3. Matching & Notification Algorithm

Algorithm Matching_And_Notification
1. Start
2. Breakdown request received in system
3. System identifies user location
4. Search for nearby mechanics within service area
5. Filter mechanics by:
      - Availability
      - Skills required
6. Rank top matches

7. Send request notification to matched mechanics
8. If (mechanic accepts first) then
    Confirm assignment
    Send mechanic details to user
  Else
    Continue until one mechanic accepts
9. End

---

4. Payment & Ratings Algorithm

Algorithm Payment_And_Ratings
1. Start
2. Mechanic updates job status = "Completed"
3. User confirms completion
4. System generates invoice
5. User selects payment method:
    - Mobile money
    - Card
    - Cash
6. If (payment successful) then
    Transfer funds to mechanic (minus commission)
  Else
    Prompt user to retry payment
7. User rates mechanic (1–5 stars + feedback)
8. Save rating to mechanic profile
9. End

---

5. Real-time Tracking & Updates Algorithm

```
Algorithm RealTime_Tracking_And_Updates
1. Start
2. Mechanic accepts breakdown request
3. Share mechanic location with user
4. Display ETA on user map
5. While (job not completed) do
       Update mechanic location in real-time
       Send status updates to user:
           - On the way
           - Arrived
           - Work in progress
       Mechanic updates progress
   End While
6. When job = "Completed"
       Stop tracking
7. End
```

<u>SYSTEM DEVELOPMENT PROCESS</u>

1. **Requirement Analysis**

- Identify stakeholder needs (drivers, mechanics, admins, insurers).

- Define functional requirements:

    - Request roadside help

    - Match with nearest available mechanic
    - Live tracking & ETA
    - In-app payments
    - Ratings & feedback
    - Admin dispute resolution
- Define non-functional requirements:
    - Scalability (handle thousands of concurrent requests)
    - Security (safe payments, data protection)
    - Reliability (high availability, fault tolerance)
    - Performance (fast geo-matching, low-latency tracking)

---

2. System Design

- **High-level architecture**: Mobile apps (iOS/Android), backend services (API, matching engine), database, real-time tracking.

- **Database design**: Users, mechanics, jobs, payments, ratings.
- **UI/UX design**: Simple request flow for drivers; quick accept/reject for mechanics.
- **Technology stack**: Flutter/React Native (cross-platform), Node.js/Go/Python backend, PostgreSQL with PostGIS, Stripe for payments.
- **Security design**: Encryption, authentication (JWT + OTP), secure APIs.

---

3. Implementation (Development)

- Build driver and mechanic mobile apps.
- Implement APIs for job creation, matching, tracking, payments.
- Integrate maps & geolocation services.
- Develop admin dashboard for vetting, monitoring, and payouts.
- Code reviews, version control (Git), CI/CD setup.

---

4. Testing

- **Unit testing**: Verify APIs, payment integration, geo-matching functions.
- **Integration testing**: Ensure driver–mechanic flows work end-to-end.
- **Performance testing**: Simulate high load to ensure real-time performance.
- **Security testing**: Penetration testing, API security validation.
- **User acceptance testing (UAT)**: Pilot run with small group of drivers & mechanics.

---

5. Deployment

- Deploy backend on scalable cloud (AWS/GCP/Azure).
- Release mobile apps on App Store & Google Play.
- Use containerization (Docker + Kubernetes) for backend services.
- Set up monitoring & logging (Prometheus, Grafana, Sentry).

6. Maintenance & Support

- Bug fixes and performance optimization.
- Regular security patches and compliance updates (PCI DSS for payments).
- Add features based on feedback (e.g., towing, insurance integration).
- Monitor KPIs: time-to-match, ETA accuracy, job completion rate.