ANDROID-BASED VEHICLE MONITORING AND ON-ROAD BREAKDOWN SERVICE SYSTEM

S.Abinaya,A.Kokila ,N.Yuvaraj* And G.M.Sathyaseelan

Department of Information Technology, J K K Natraja College of Engineering and Technology, Komarapalayam-683183, Tamil Nadu, India.

ABSTRACT

The system entitled "ANDROID BASED VEHICLE MONITORING & ON-ROAD BREAKDOWN SERVICE SYSTEM" is designed using the Standard Android 4.0 platform. Developed with Android Studio and Java 1.6 Standard Edition, this application allows users to monitor vehicle servicing and maintenance data. It also provides on-road breakdown services, enabling users to find the nearest service center in case of an emergency. The system records vehicle repair details, shares them with customers through a valid ID, and displays the service status as a percentage. Users can view these details anytime, ensuring transparency and efficiency in vehicle maintenance.

Keywords: Vehicle Monitoring, Android Application, Breakdown Service, Vehicle Maintenance, Real-Time Monitoring

1. INTRODUCTION

Android is a mobile operating system (OS) developed by Google, based on the Linux kernel and designed primarily for touchscreen mobile devices such as smartphones and tablets. Its user interface is based on direct manipulation, using touch gestures that correspond to real-world actions like swiping, tapping, and pinching to manipulate on-screen objects. Android also includes a virtual keyboard for text input. In addition to mobile devices, Google has developed specialized versions of Android for televisions (Android TV), cars (Android Auto), and wristwatches (Android Wear), each with a unique user interface. The open nature of Android has encouraged a large community of developers and enthusiasts to use the open-source code as a foundation for community-driven projects, adding new features and expanding its functionality [1].

The web application component of this system provides comprehensive information about vehicle servicing and breakdown services. It offers real-time tracking of vehicle maintenance, allowing users to access recent updates on their vehicles' status. The web application supports communication between users and service providers, making it easier to manage service records and monitor progress. Security features such as data encryption and error checking ensure the integrity and confidentiality of user data, preventing unauthorized access and protecting sensitive information [2].

2. SYSTEM ANALYSIS

2.1 Feasibility Study

2.1.1 Technical Feasibility

Technical feasibility involves assessing the capabilities of the existing technology to support the proposed system. The project uses advanced tools like Android Studio for development and MySQL for backend database management. The Android platform provides a robust environment for building scalable and efficient applications, while the MySQL database ensures reliable data storage and retrieval. This combination enables the system to handle real-time data processing and user interactions efficiently [3].

2.1.2 Economic Feasibility

Economic feasibility examines the cost-effectiveness of the project. By leveraging open-source tools and cloud-based resources, the project minimizes initial investments and operational costs. The transition from manual to automated systems reduces the need for extensive paperwork and manual record-keeping, resulting in significant cost savings. Additionally, the system's ability to provide timely updates and alerts can reduce maintenance costs and improve service efficiency, leading to long-term financial benefits [4].

2.1.3 Operational Feasibility

Operational feasibility considers the practicality of implementing the system within the current organizational structure. The proposed system integrates seamlessly with existing vehicle maintenance workflows, allowing users to access real-time data and communicate with service providers through a user-friendly interface. The system's automated processes enhance

operational efficiency, reducing the workload on staff and improving overall service quality [5].

2.2 Existing System

The existing vehicle maintenance systems rely heavily on manual processes and periodic checks, leading to inefficiencies and delays. Service records are often maintained on paper, making it difficult to track the service history and status of individual vehicles. Customers are not provided with real-time updates, and the lack of integration with modern communication tools hinders effective interaction between users and service providers [6].

2.2.1 Drawbacks

- Limited to car servicing.
- Does not show service completion levels to customers.
- Relies on manual record-keeping and periodic checks.
- Lacks real-time updates and effective communication channels.

2.3 Proposed System

The proposed system addresses these drawbacks by providing a comprehensive solution for vehicle monitoring and on-road breakdown services. Users can access real-time updates on vehicle maintenance, view service completion levels as percentages, and communicate with service providers through the app. The system also helps users locate the nearest service centre in case of a breakdown, ensuring timely assistance and minimizing downtime [7].

2.3.1 Advantages

- Easy to access and user-friendly.
- Reduces time consumption by providing real-time updates.
- Displays vehicle service completion levels as percentages.

- Facilitates two-way communication between users and service providers.
- Accessible anytime and anywhere.
- Useful in emergencies, helping users locate the nearest service center quickly.

3. SYSTEM DESIGN

3.1 Architecture Diagram

The architecture diagram 3.1 outlines the system's structure, illustrating the components and their interactions. The diagram includes user and admin login, service status monitoring, booking complaints, notifications, and employee ticket management [8].

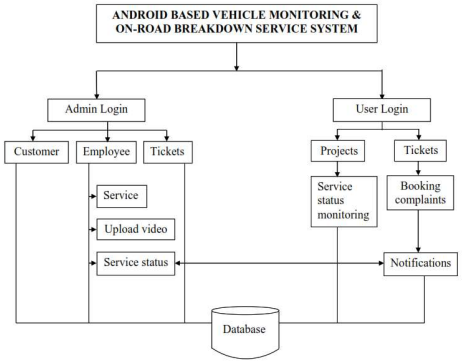

Figure 3.1 Architecture diagram

3.2 Use Case Diagram

The use case diagram 3.2 shows the interactions between users (customers and admin) and the system. It includes functionalities such as registration, complaints, login, monitoring, notifications, payments, and maintenance [9].

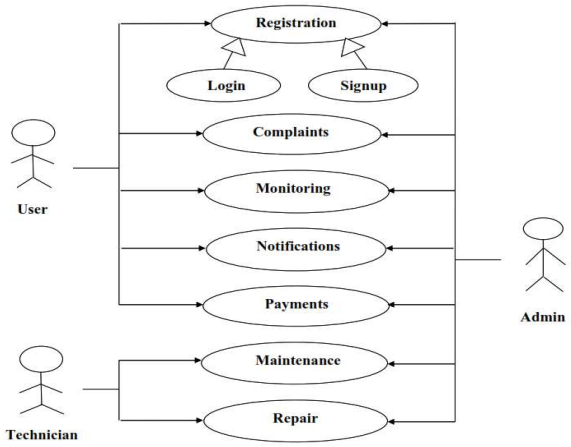

Figure 3.2 Use case diagram

3.3 Class Diagram

The class diagram 3.3 represents the system's main classes and their relationships. It includes classes for user and admin details, chat details, login, signup, communication, and notification functionalities [10].

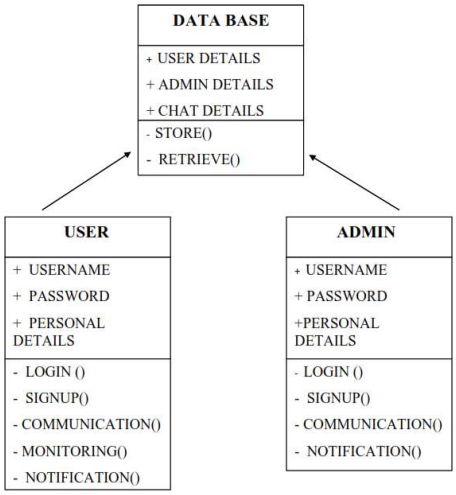

Figure 3.3 Class Diagram

3.4 Sequence Diagram

The sequence diagram 3.4 illustrates the sequence of interactions between the user, admin, and database, including login, sending messages, storing data, and processing payments [11].

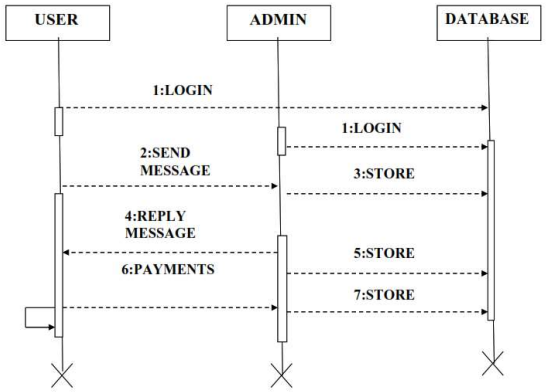

Figure 3.4 Sequence diagram

4. REQUIREMENTS AND SPECIFICATIONS

4.1 Hardware Requirements

- System Processor: Pentium 4
- RAM: 512 MB RAM
- Hard Disk: 80 GB
- Mouse: Optical Mouse

4.2 Software Requirements

- Tools: Android Studio 4.1
- Front-End: HTML, CSS, JavaScript
- Coding: JDK 8 & PHP
- Operating System: Android
- Back-End: MySQL

4.3 Software Specification

4.3.1 Front End

The front end is developed using HTML, CSS, and JavaScript, providing a user-friendly interface. The mobile application is developed using Android Studio, ensuring compatibility across various devices and platforms. The interface allows users to monitor vehicle service status, receive notifications, and access service records [12].

4.3.2 Back End

The back end is implemented using MySQL for data management and PHP for server-side scripting. The data collected from users and vehicles is stored in a MySQL database, providing efficient data retrieval and management. The system uses RESTful APIs for communication between the mobile application and the server [13].

5. SYSTEM IMPLEMENTATION AND RESULT

5.1 Modules

The system is divided into several modules:

- Admin Modules: Customers, Employees, Work Status, Finance, Products, Tickets, Messages, Settings
- Client Modules: Projects, Tickets, Payments, Messages

5.2 Module Descriptions

5.2.1 Admin Modules

- **Customers**: Admin can view and manage customer details, add new customers, and update customer information.
- **Employees**: Admin assigns work to employees, manages employee lists, and updates employee profiles.
- **Work Status**: Admin monitors project status, completion levels, and deadlines, and updates project details.

- **Finance**: Manages estimates, invoices, payments, and expenses, allowing customers to make payments and view transaction details.
- **Products**: Manages product purchases and price lists, and adds new products.
- **Tickets**: Manages customer complaints, prioritizes tickets, and tracks ticket trends.
- **Messages**: Facilitates communication between the admin and customers.
- **Settings**: Manages profile, notification, project, payment, and ticket settings [14].

5.2.2 Client Modules

- **Projects**: Displays service completion levels as percentages and monitors work status.
- **Tickets**: Allows customers to lodge complaints, upload videos and images, and track the nearest service centre during breakdowns.
- **Payments**: Facilitates online and offline payments after service completion.
- **Messages**: Facilitates communication between customers and the admin [15].

5.3 Input Design

Input design involves determining the data to be input, the medium to use, and how the data should be arranged and validated. The project includes input forms for adding new projects, employee assignments, and customer information. This ensures that data is captured accurately and efficiently, reducing errors and improving the reliability of the system [16].

5.4 Output Design

Output design refers to the results generated by the system, such as reports, screen displays, and printed forms. The system outputs include project completion levels, customer complaints, and transaction details. These outputs are designed to be accurate, timely, and appropriate for their intended purpose, ensuring that users can make informed decisions based on the information provided [17].

6. SYSTEM TESTING

6.1 Unit Testing

Unit testing involves testing individual modules to ensure they function correctly. Each module is tested independently to detect errors in coding and logic. This testing is carried out during the programming stage, allowing developers to identify and fix issues early in the development process [18].

6.2 User Acceptance Testing

User acceptance testing involves providing the project to customers to ensure all requirements are met. Feedback is used to make necessary adjustments and improvements. This testing ensures that the system meets user expectations and performs as intended in a real-world environment [19].

6.3 Integration Testing

Integration testing ensures that individual modules work together as a single unit. This testing verifies that data flows correctly between modules and the system functions as intended. Integration testing helps identify issues that may arise when different components of the system interact with each other [20].

6.4 Validation Testing

Validation testing involves verifying that the system meets all specified requirements and performs as expected. This includes checking data accuracy, system reliability, and performance under various conditions. Validation testing ensures that the system is ready for deployment and can handle real-world scenarios effectively [21].

7. CONCLUSION AND FUTURE ENHANCEMENTS

7.1 Conclusion

The proposed Android-based vehicle monitoring and on-road breakdown service system provides a comprehensive solution for vehicle maintenance and emergency services. It allows users to monitor service status, receive real-time updates, and find the nearest service centre during breakdowns. The system is user-friendly, efficient, and saves time and money for customers. By leveraging modern technologies and automated processes, the system enhances the overall vehicle maintenance experience [22].

7.2 Future Enhancements

Future enhancements may include:

- Sending notifications via SMS while offline.
- Expanding the application to include multiple service centres for different types of services.
- Grouping multiple branches under one application.
- Developing an iOS version of the application.
- Incorporating multilingual support for wider accessibility [23].

8. REFERENCES

1. Angel, Gonzalez Villan, Joseph Jorba Esteve, “Remote Control of Mobile Devices in Android Platform”, IEEE. DOI: https://doi.org/10.1109/RCMD.2017.7815759
2. Atzori.L, Iera.A, Morabito.G, “Understanding the Internet of Things: Definition, potentials, and societal role of a fast-evolving paradigm”, Ad Hoc Netw 2017. DOI: https://doi.org/10.1016/j.adhoc.2016.04.002
3. H. Abelson and M. Friedman, “App Inventor - A view into learning about computers through building mobile applications”, Proceedings of the 2010 SIGCSE Symposium, March 2010. DOI: https://doi.org/10.1145/1734263.1734274
4. Jaya Bharathi Chintalapati, Srinivasa Rao, “Remote computer access through Android mobiles”, International Journal of Computer Science Issues. DOI: https://doi.org/10.5120/ijca2017913205
5. Jr-Jen Huang, Yi-Yu Chu, Yen-Jen Chen, “The System Design and Implementation of Vehicle Management”, Journal of Advances in Computer Networks, Vol. 1, No. 1, March 2013. DOI: https://doi.org/10.1109/JACN.2013.6514156
6. Karan Balkar, Reyomi Roy, Preeyank Pable, M. Kiruthika, Shweta Tripathi, “A Mobile Application to Access Remote Database using Web Services”, Third Biennial National Conference on Nascent Technologies, Fr. C. Rodrigues Institute of Technology, Vashi, Navi Mumbai. DOI: https://doi.org/10.1109/NCNT.2017.8068659
7. Dr. Khanna SamratVivekanand Omprakash, “Accessing information on mobile client from mobile and web server with internet from remote place”, International Journal of

Advanced Engineering Technology. DOI: https://doi.org/10.1155/IJAET.2017.2903571

8. Rohit Dhall, Vijender Solanki, "An IoT Based Predictive Connected Car Maintenance Approach", International Journal of Interactive Multimedia and Artificial Intelligence, Vol. 4, No. 3. DOI: https://doi.org/10.9781/ijimai.2017.431
9. Sayali Nerkar, Shweta Jadhav, Radhika Shouche, Sukanya Sonawani, "Android Based Vehicle Service Status Monitoring System", International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, Vol. 3, Issue 2, February 2014. DOI: https://doi.org/10.15662/ijareeie.2014.0302003
10. Shivang Shah, Parimal Abhishek, Deep Shrivastava, Abraham Sudharson Ponraj, "Vehicle Service Management and Live Monitoring with Predictive Maintenance System", 2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN). DOI: https://doi.org/10.1109/ViTECoN.2019.8899452