Benha National University
Faculty of Computer Science

CARWARE

Integrated Vehicle Service and Management System

This project was submitted in partial fulfillment of the requirements for the Bachelor's Degree in Computer Science.

Project Team

Mohamed Ayman Mohamed
Zyad Osama Aboelsoud
Abeer Reda Abdelaziz
Fatma Ahmed Abdelhamid
Abdo Mohamed Abdo
Habiba Mohamed Sayed
Menna Ahmed Abdel Mohsen
Marwan Mohamed Elattar

Supervised by: Dr. Ahmed Tawfik

2025–2026

Contents

List of Figures

ABSTRACT

The rapid advancement of automotive technology has resulted in vehicles becoming increasingly complex and data-driven. Despite the availability of onboard diagnostics and vehicle telemetry, most vehicle owners lack effective digital tools to monitor vehicle status, organize maintenance activities, and manage service records. At the same time, automotive service centers often rely on manual workflows that limit operational efficiency and transparency.

This project presents **CarWare**, an integrated vehicle service and management system designed to connect vehicle owners and service centers through a unified digital platform. The system consists of a mobile application for vehicle owners and a web-based management system for service centers, supported by a centralized backend server. The mobile application allows users to manage vehicle data, track maintenance history, and schedule service appointments, while the service center platform supports workflow management, diagnostics handling, inventory monitoring, and customer communication.

CarWare aims to enhance transparency, improve service efficiency, and provide a scalable and secure solution suitable for real-world deployment in the automotive service industry.

Chapter 1

INTRODUCTION

The automotive industry has undergone a significant transformation over the past decades, evolving from systems that were primarily mechanical in nature to highly sophisticated, digitally controlled, and data-driven platforms. Modern vehicles are now equipped with numerous electronic control units (ECUs), embedded sensors, and standardized diagnostic interfaces that continuously generate large volumes of operational and diagnostic data. This data includes information related to engine performance, fuel consumption, emissions levels, system faults, and overall vehicle health.

Despite the availability of this valuable data, its effective utilization in everyday vehicle maintenance and service operations remains limited. For many vehicle owners, diagnostic information is inaccessible or difficult to interpret without specialized tools or technical expertise. As a result, maintenance activities are often performed reactively, only after a fault occurs, rather than proactively based on real-time vehicle condition and historical data. This reactive approach can lead to increased repair costs, unexpected breakdowns, and reduced vehicle lifespan.

On the service center side, many automotive workshops still rely on traditional or semi-digital workflows for managing appointments, diagnostics, service records, and customer communication. These practices often result in fragmented data storage, inefficient processes, and limited transparency for vehicle owners. The lack of a unified platform that connects vehicle data with service operations creates a disconnect between vehicle owners and service providers, negatively impacting service quality and customer trust.

This chapter introduces the motivation, objectives, scope, and importance of the CarWare system, an integrated vehicle service and management solution designed to address these challenges. CarWare aims to bridge the gap between vehicle owners and service centers by leveraging modern software technologies to centralize vehicle data, streamline service workflows, and enhance communication. By providing a unified digital platform, the system seeks to improve maintenance efficiency, increase transparency, and support the transition toward smarter and more reliable vehicle service management. The automotive industry has transitioned from purely mechanical systems to highly digital, data-driven platforms. Modern vehicles generate extensive diagnostic and operational data through embedded sensors and electronic control units. However, the utilization of this data in daily vehicle maintenance and service operations remains limited. This chapter introduces the motivation, objectives, scope, and importance of the CarWare system.

1.1 Background

In addition to technological complexity, modern vehicle ownership has become more demanding due to increasing maintenance costs, stricter emission regulations, and the growing number of electronic components. Vehicle owners are expected to follow precise maintenance schedules to preserve vehicle reliability and warranty coverage, yet most lack centralized tools to support these responsibilities.

At the same time, service centers are under pressure to improve service quality, reduce turnaround time, and manage customer expectations. The absence of integrated digital systems makes it difficult to synchronize vehicle data, service history, and customer communication, highlighting the need for a comprehensive digital solution.

1.1.1 The Evolving Automotive Service Industry

The automotive sector has undergone rapid technological development with the extensive integration of electronic control units, onboard sensors, and standardized diagnostic interfaces. Modern vehicles continuously generate operational data related to engine performance, fuel efficiency, emissions, and system status. This wealth of data is crucial for maintaining vehicle reliability and extending its operational lifespan, provided it is properly collected and analyzed [1]. The project, **CarWare**, is designed to create an intelligent system to support this necessary vehicle monitoring and service management [3].

1.1.2 The IoT and OBD-II Telemetry

The increasing adoption of Internet of Things (IoT) solutions has created a need for integrated systems that support advanced vehicle monitoring and service management. Central to this is the **OBD-II (On-Board Diagnostics, second generation)** interface [2]. CarWare leverages this interface to collect and monitor real-time vehicle health data. This capability shifts maintenance from a reactive approach (fixing problems after they occur) to a proactive one (preventing issues through timely, data-driven service), which is vital for the advancement of smart vehicle management systems.

1.1.3 The Digital Disconnect

Despite the availability of comprehensive vehicle diagnostic data, a significant digital disconnect exists. Most vehicle owners lack effective digital tools to access, interpret, and manage this information in a practical manner. Furthermore, maintenance activities and records are frequently stored in fragmented formats, making it difficult for owners to track their vehicle's history accurately and manage service expenses. This lack of digital continuity forms the foundation of the problem CarWare seeks to solve.

1.2 Problem Statement

The challenges in the automotive service industry are dual-sided, affecting both vehicle owners and service providers, leading to operational friction and reduced trust.

1.2.1 Fragmented Vehicle Owner Experience

- **Lack of Proactive Maintenance:** Most owners rely on reactive maintenance, often leading to increased repair costs and unexpected vehicle breakdowns, as they lack tools to access and interpret diagnostic data.

- **Disorganized Record Keeping:** Service records and maintenance histories are often stored in fragmented formats (e.g., paper receipts, isolated files), making it difficult for owners to maintain comprehensive, accurate vehicle records.

- **Inefficient Communication:** Owners must often rely on phone calls or fragmented methods to coordinate schedules and service updates with repair shops.

1.2.2 Operational and Information Silos in Service Centers

- **Manual and Isolated Operations:** Many service centers rely on manual procedures or isolated software solutions for critical functions like appointment scheduling, spare part management, and customer communication [6].
- **Reduced Operational Efficiency:** These non-integrated approaches limit operational efficiency and reduce the capacity to provide consistent, transparent services.
- **Poor Information Flow:** The service center may not have ready access to accurate vehicle histories or real-time owner-reported issues, resulting in delays, miscommunication, and the inability to provide targeted maintenance recommendations.

1.2.3 Problem Summary

The lack of integration between vehicle owners and service centers results in a poor flow of information, reduced trust, and inefficient operations. The market needs a centralized, unified platform that leverages vehicle telemetry to automate maintenance needs and streamline the entire service workflow, from initial scheduling to final reporting.

1.3 Project Objectives

The primary objective is to design and develop **CarWare**, an integrated vehicle service and management system. This overarching goal is achieved through specific functional and non-functional objectives.

1.3.1 Functional Objectives

1. **Vehicle Health Monitoring:** Enable users of the client-facing Android application to monitor real-time vehicle health data collected through OBD-II telemetry.

2. **Maintenance Scheduling & Reminders:** Implement features for vehicle owners to schedule service appointments and receive automated maintenance reminders.

3. **Expense and Record Tracking:** Provide tools for vehicle owners to track maintenance expenses and maintain comprehensive, digital service records.

4. **Appointment and Workflow Management (Web):** Develop a web-based system for service centers to manage appointments, automate workflow, and handle diagnostic reports efficiently.

5. **Inventory Monitoring (Web):** Provide service centers with tools for basic inventory monitoring and spare part management.

6. **Secure Communication:** Implement secure and transparent communication channels between vehicle owners and service centers regarding service status.

1.3.2 Non-Functional Objectives

1. **Usability and Intuitive Design:** Design an intuitive client application and an efficient service center web platform to ensure a positive user experience and minimize training requirements.

2. **Scalability:** Design a secure and suitable architecture capable of handling future expansion, accommodating a growing number of vehicle owners and service centers.

3. **Security:** Implement secure user authentication and data protection protocols, ensuring the confidentiality and integrity of vehicle telemetry and customer information.

4. **Reliability:** Ensure the system demonstrates high reliability with robust error handling and continuous availability for core functions like scheduling and data retrieval.

1.4 Scope of the Project

This graduation project delivers a fully functional academic prototype, focused on demonstrating core workflows and the feasibility of the integrated system design.

1.4.1 First Semester: Mobile Application and Backend Foundation

During the first semester, the user interface of the mobile application was designed using Figma and implemented using Kotlin Multiplatform (KMP). In parallel, a backend server was developed to handle authentication and login functionality.

1.4.2 Second Semester: Service Center Platform

The second semester will focus on developing the service center web application, completing system integration, testing, and deployment.

1.5 Motivation and Importance

1.5.1 Motivation

The development of CarWare is driven by two main factors:

1. **Bridging the Technological Gap:** The current fragmented service model is outdated given modern vehicle technology. The project is motivated by the opportunity to leverage smartphone use and web systems to provide a modern, integrated service experience.

2. **Improving Trust and Efficiency:** By centralizing vehicle data and service operations, the system offers a pathway to reduce manual errors, optimize resource utilization in service centers, and enhance the overall transparency for the vehicle owner, thereby improving coordination and trust.

1.5.2 Importance

CarWare provides significant value by:

- **Enhancing Transparency:** It enhances transparency in the service process, providing vehicle owners with confidence in the work performed.

- **Improving Operational Efficiency:** It provides service centers with workflow automation and operational tools to streamline their daily tasks, contributing to a modern service ecosystem.

- **Advancing Smart Vehicle Management:** It contributes directly to the advancement of smart vehicle management systems by offering a scalable and secure architecture suitable for real-world deployment.

1.6 Methodology Overview

The development of the CarWare system follows a structured and iterative methodology to ensure reliability, scalability, and alignment with real-world automotive service requirements. The project adopts a modular development approach, where system components are designed, implemented, and validated independently before full integration.

The initial phase focused on requirements analysis through studying existing automotive service workflows, reviewing related literature, and identifying key pain points experienced by both vehicle owners and service centers. Based on this analysis, system requirements were defined and translated into functional modules covering vehicle management, service scheduling, expense tracking, and service center operations.

User interface design was carried out using Figma to create intuitive and consistent layouts for the mobile application. These designs were validated through iterative refinement to ensure usability and clarity. The mobile application was then implemented using Kotlin Multiplatform (KMP), enabling clean architecture separation and maintainability.

In parallel, the backend system was developed using ASP.NET Core, following a layered architecture pattern. This approach ensures separation of concerns between business logic, data access, and presentation layers. Secure authentication and authorization mechanisms were implemented to protect sensitive vehicle and user data.

Finally, system integration and testing phases validate data consistency, workflow correctness, and overall system performance. This methodology ensures that CarWare evolves as a robust academic prototype suitable for real-world deployment.

Chapter 2

Related Work

This chapter presents the overall system architecture and design models of the CarWare system. A well-defined architecture is essential for ensuring system scalability, maintainability, and effective communication between different system components. Given the dual-platform nature of CarWare—serving both vehicle owners and service centers—a clear architectural design is required to support seamless data exchange, secure operations, and efficient workflow management.

The chapter focuses on illustrating how the system components interact through a set of standardized modeling diagrams. These diagrams provide both high-level and detailed views of system behavior, data flow, and structural relationships. By visualizing system functionality using modeling techniques, the design becomes easier to analyze, validate, and extend in future development phases.

Several Unified Modeling Language (UML) and system design diagrams are presented to describe different perspectives of the CarWare system. Use-case diagrams define the interactions between users and the system, highlighting the core functionalities available to vehicle owners and service center staff. Entity-relationship diagrams illustrate the database structure and the relationships between key system entities, ensuring data consistency and integrity. Sequence diagrams demonstrate the dynamic behavior of the system during critical operations such as appointment scheduling, maintenance tracking, and payment processing. Additionally, data-flow and activity diagrams provide insight into how information moves through the system and how various processes are executed.

Through these architectural representations, this chapter establishes a solid foundation for system implementation. The presented models serve as a blueprint for the development process, ensuring that both the mobile application and the web-based service center platform are built in alignment with the system's functional and non-functional requirements. This structured approach supports the development of a robust, scalable, and efficient vehicle service management system.

2.1 IoT-Based Vehicle Monitoring Systems

Recent academic research has emphasized the role of IoT technologies in enhancing vehicle monitoring and predictive maintenance. IoT-based systems utilize onboard sensors, communication modules, and cloud platforms to collect and analyze vehicle data in real time. These systems aim to improve fault detection, reduce downtime, and optimize maintenance schedules.

2.2 Mobile Applications for Vehicle Management

Mobile applications have become an essential interface for vehicle-related services. Existing applications provide features such as fuel tracking, basic diagnostics, and trip logging. However, most applications operate independently of service center systems and lack comprehensive maintenance workflows.

2.3 Web-Based Automotive Service Platforms

Web-based platforms are widely used by automotive service centers to manage internal operations. These systems typically support appointment scheduling, customer records, billing, and inventory management. Although effective for internal organization, they rarely integrate real-time vehicle data or provide vehicle owners with direct access to service progress.

The lack of synchronization between service center systems and vehicle owner applications remains a key limitation identified in the literature.

Research highlights that user adoption significantly improves when applications provide intuitive interfaces, actionable insights, and integration with service providers. These findings support the design choice of CarWare to emphasize usability and cross-platform accessibility.

2.4 Existing Monitoring and Service Platforms

Existing digital solutions for vehicle monitoring and service management often focus on isolated functionalities rather than providing a fully integrated platform. We categorize and analyze the main types of existing systems below:

2.5 Vehicle History Record Platforms (e.g., Carfax)

Commercial applications such as Carfax primarily focus on maintaining historical records of a vehicle. These records typically include ownership details, accident reports, and service history performed by authorized dealers or major shops.

Strengths: Highly useful for transparency, establishing vehicle value, and tracking historical service milestones, particularly for resale purposes.

Limitations: These platforms are largely limited to historical data and do not support real-time vehicle data monitoring or direct, interactive communication with independent service centers. Their functionality is generally reactive rather than active vehicle management.

2.6 OBD-II Diagnostic Tools (e.g., Torque Pro)

Applications like Torque Pro and OBD-based mobile tools provide real-time access to vehicle diagnostic data through a OBD-II adapter. Strengths: They allow technical users to monitor engine parameters, read diagnostic trouble codes (DTCs), and view live sensor data, offering an immediate, deep dive into vehicle health. Limitations: These tools typically

lack structured maintenance management features, expense tracking, and service scheduling functionalities. Crucially, they do not offer integrated systems for service centers to manage customer interactions or workflow processes, thus catering only to the technical owner and not solving the service disconnect.
Several studies propose architectures where vehicle telemetry data is transmitted to cloud-based platforms for analysis and visualization. While these approaches demonstrate technical feasibility, many focus on experimental setups or fleet-level monitoring rather than individual vehicle owners and service center integration.

2.7 Workshop Management Systems (e.g., AutoCare)

These systems are designed to assist service centers with internal operations, including appointment scheduling, managing customer records, and job tracking.

Strengths: They significantly improve internal workflow efficiency and organization within the service center.

Limitations: They operate independently from vehicle owner applications, meaning vehicle owners have limited visibility into service operations, and data exchange between both parties remains minimal or manual. They solve the center's operational problem but not the communication/transparency problem.

2.8 Comparison and Analysis

A comparative analysis of existing commercial applications, academic research solutions, and the proposed CarWare system highlights several functional and architectural gaps that motivate the development of an integrated platform. Most existing solutions focus on addressing a single stakeholder or a limited set of functionalities, whereas CarWare is designed to unify vehicle owners and service centers within one coherent ecosystem.
Commercial vehicle history platforms such as Carfax primarily emphasize historical data aggregation, including ownership records, accident history, and previous service reports. While these systems enhance transparency during vehicle resale and ownership verification, they do not provide real-time vehicle monitoring or ongoing maintenance management. Furthermore, interaction between vehicle owners and service centers is not supported, limiting these platforms to passive data repositories rather than active service management tools.

OBD-II–based mobile applications, including Torque Pro and similar tools, provide real-time access to vehicle sensor data and diagnostic trouble codes. These applications are valuable for technically experienced users who seek direct insights into engine performance and vehicle status. However, such tools typically lack structured maintenance planning, service appointment scheduling, expense tracking, and long-term vehicle record management. In addition, they do not integrate service center operations, making them unsuitable for end-to-end service lifecycle management.

Workshop and service center management systems, such as AutoCare and similar platforms, focus on improving internal service center operations by offering appointment scheduling, job tracking, and customer record management. Although these systems enhance workflow efficiency within service centers, they often operate independently from vehicle owner applications. Consequently, vehicle owners have limited visibility into service progress, performed tasks, or future maintenance recommendations, leading to continued reliance on manual communication methods. In contrast, CarWare differentiates itself by offering a fully integrated dual-platform architecture. The system combines a client-facing mobile application, developed using Kotlin Multiplatform and designed through Figma-based user interfaces, with a web-based service center management system supported by a centralized backend server. This architecture enables seamless data exchange, unified authentication, and consistent vehicle record management across both stakeholders.

By addressing the limitations of existing commercial tools , CarWare provides a more comprehensive solution that supports real-time vehicle monitoring, structured maintenance management, transparent service workflows, and improved communication between vehicle owners and service centers. This integrated approach positions CarWare as a practical and scalable system suitable for real-world deployment in the automotive service industry.

2.9 Summary of Findings

The literature review reveals that existing solutions for vehicle monitoring and service management are largely fragmented. Commercial vehicle history platforms focus on static historical records, while OBD-based mobile applications provide real-time diagnostics without structured service coordination.

Workshop management systems enhance internal service center efficiency but lack integration with vehicle owner applications, resulting in limited transparency. Academic research demonstrates the technical potential of IoT-based vehicle monitoring but often overlooks usability and real-world service integration. These findings confirm the need for a unified system that integrates vehicle monitoring, maintenance management, and service center operations. CarWare directly addresses these gaps by combining mobile and web platforms within a single, cohesive architecture.

Chapter 3

SYSTEM ARCHITECTURE AND DIAGRAMS

This chapter focuses on the design and implementation aspects of the CarWare system, with particular emphasis on the backend architecture, core system components, and the technologies used to realize the proposed solution. After establishing the system requirements and architectural models in the previous chapters, this chapter translates those designs into practical implementation details that enable the system to operate reliably and securely.

The chapter explains how the backend of CarWare is structured using a layered architectural approach to separate concerns and improve maintainability. Each layer is designed to handle a specific responsibility, including business logic, data persistence, authentication, and application programming interfaces (APIs). This modular structure facilitates scalability, simplifies testing, and allows future enhancements to be introduced with minimal impact on existing components.

In addition, this chapter discusses the key technologies, frameworks, and development tools employed in building the CarWare system. These include backend frameworks for API development, database management solutions, authentication mechanisms, and communication protocols that support secure data exchange between the mobile application, the web-based service center platform, and the server. Particular attention is given to security considerations, such as user authentication and authorization, to ensure data privacy and system integrity.

By detailing the implementation strategy and technological choices, this chapter provides a comprehensive understanding of how CarWare's conceptual design is transformed into a functional system. The content presented serves as a foundation for evaluating system performance, reliability, and extensibility, and prepares the ground for future testing, deployment, and enhancement phases.

3.1 System Architecture

The proposed system follows a client–server architecture designed to support scalability, security, and ease of maintenance. It consists of two main platforms: a mobile application used by vehicle owners and a web-based management system used by service centers. Both platforms communicate with a centralized backend server that handles business logic, data processing, and database operations.

The system architecture is divided into three main layers: the presentation layer, the application layer, and the data layer. This layered approach ensures separation of concerns, allowing each layer to function independently while maintaining clear interaction interfaces. The architecture enables real-time data exchange between users and service centers, ensuring accurate service tracking and efficient management of vehicle-related information.

3.2 System Modules

The system is composed of several interconnected modules, each responsible for specific functionalities.

3.2.1 Mobile Application Module

This module allows vehicle owners to register and log in to the system, manage their vehicle information, book service appointments, and track service status. The mobile application provides a user-friendly interface designed to simplify interaction and enhance user experience.

3.2.2 Web Application Module

The web-based module is dedicated to service centers and administrators. It enables service centers to manage service requests, update service status, view customer and vehicle details, and generate service records. Administrators can oversee system operations and manage users.

3.2.3 Backend Server Module

The backend server handles all business logic, request processing, authentication, and communication between the frontend platforms and the database. It ensures data consistency, validation, and secure access to system resources.

3.2.4 Database Module

The database design focuses on maintaining data integrity and minimizing redundancy. Core entities include users, vehicles, service records, appointments, expenses, and service center profiles. Relationships between entities are carefully defined to ensure efficient querying and accurate data retrieval.
Normalization techniques are applied to reduce data duplication, while indexing strategies improve query performance. This design supports reliable data storage and enables advanced reporting and analytics in future system extensions.

3.3 Architectural Design

The system adopts a layered architectural design combined with a client–server model. The presentation layer includes the mobile and web user interfaces, which interact with users directly. The application layer contains the core business logic and handles data processing, authentication, and service workflows. The data layer is responsible for data storage and retrieval.

This architectural design enhances maintainability and allows future system expansion, such as adding new services or integrating external systems, without affecting existing functionalities.

3.4 Use Case Diagram

The Use-Case Diagram illustrates the interactions between the CarWare system and its external actors. The primary actors include the Vehicle Owner, Service Center, and OBD-II Device. This diagram defines the functional scope of the system and highlights the services available to each actor.

Vehicle owners can register and log in, add and manage vehicle information, reserve service appointments, receive maintenance reminders, monitor live OBD telemetry, view diagnostic error explanations, manage expenses, process payments, and provide feedback. The service center actor manages appointments, sends notifications, creates invoices, and views analytical dashboards. The OBD-II device interacts with the system to provide real-time vehicle diagnostic and telemetry data.

This diagram provides a comprehensive overview of system functionality and actor responsibilities.

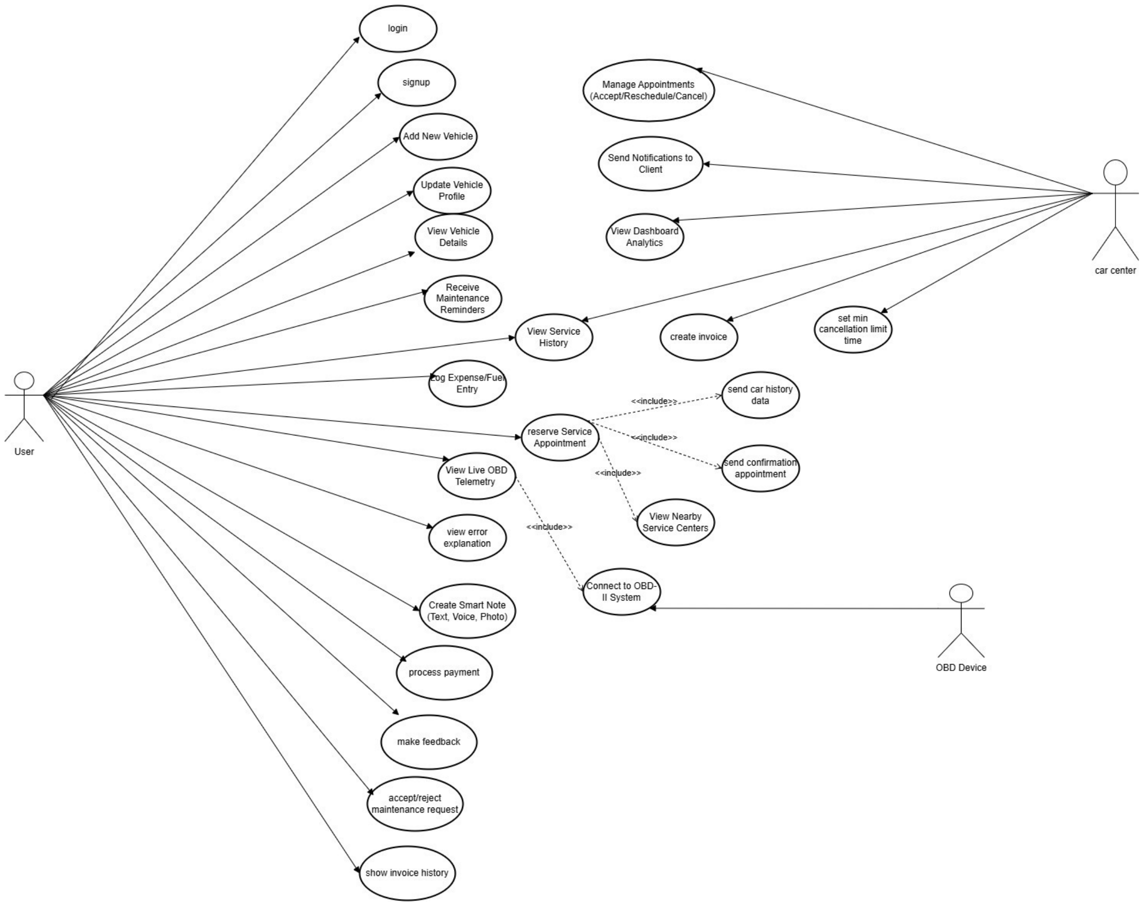

Figure 3.1: Use-Case Diagram

3.5 Database Design (ER Diagram)

The Entity Relationship Diagram (ERD) represents the logical database structure of the CarWare system and serves as the foundation for its data storage and management processes. It defines the core entities required to support both vehicle owners and service centers, including User, Vehicle, Appointment, Service Center, Maintenance Record, Expense, Payment, Reminder, Invoice, and Feedback. Each entity is designed to model real-world components of the vehicle service ecosystem and to capture all relevant attributes necessary for system functionality.

Relationships between entities are established using primary and foreign keys to ensure referential integrity and consistency across the database. For instance, a single user may own multiple vehicles, reflecting real-life ownership scenarios, while each vehicle can be associated with multiple appointments, maintenance records, expenses, and reminders over its operational lifetime. This structure allows the system to maintain a complete and chronological history of vehicle usage, servicing, and costs.

The ERD is designed following database normalization principles to minimize data redundancy and prevent update anomalies. By separating concerns into well-defined entities and relationships, the design supports efficient querying, scalability, and long-term maintainability. Overall, the ERD ensures that the CarWare database can reliably support integrated vehicle monitoring, maintenance management, and service center operations within a unified and scalable system architecture.

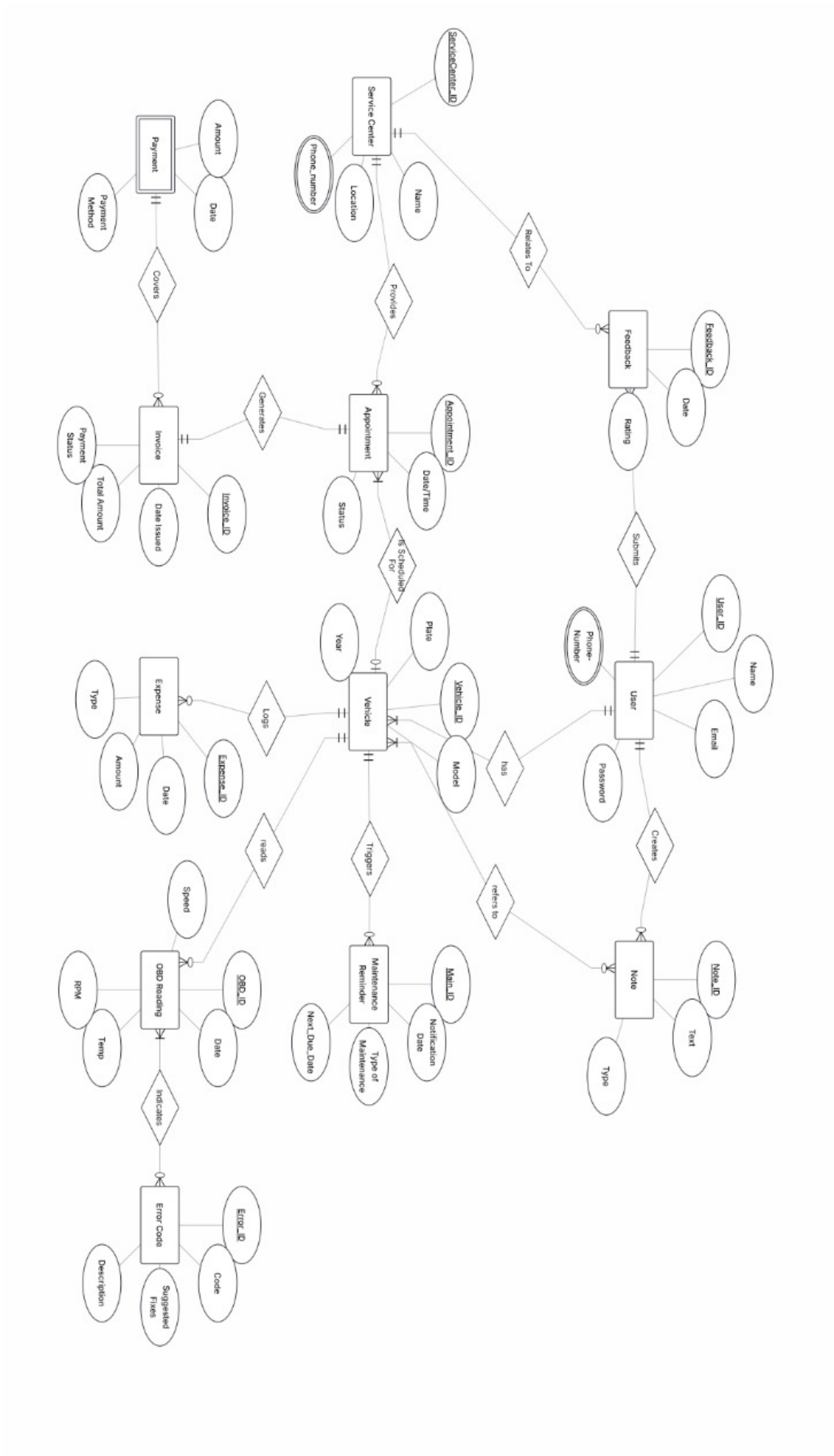

Figure 3.2: Entity Relationship Diagram

3.6 Sequence Diagrams

3.6.1 Appointment Sequence Diagram

The Appointment Sequence Diagram illustrates the interaction flow involved in reserving a service appointment. The process begins when the vehicle owner submits an appointment request through the mobile application.

The system backend validates the request, checks service center availability, and forwards the request to the selected service center. The service center can accept, reschedule, or reject the appointment. The decision is then communicated back to the user, and the appointment status is updated in the database. This diagram emphasizes time-ordered communication between the user, backend, and service center.

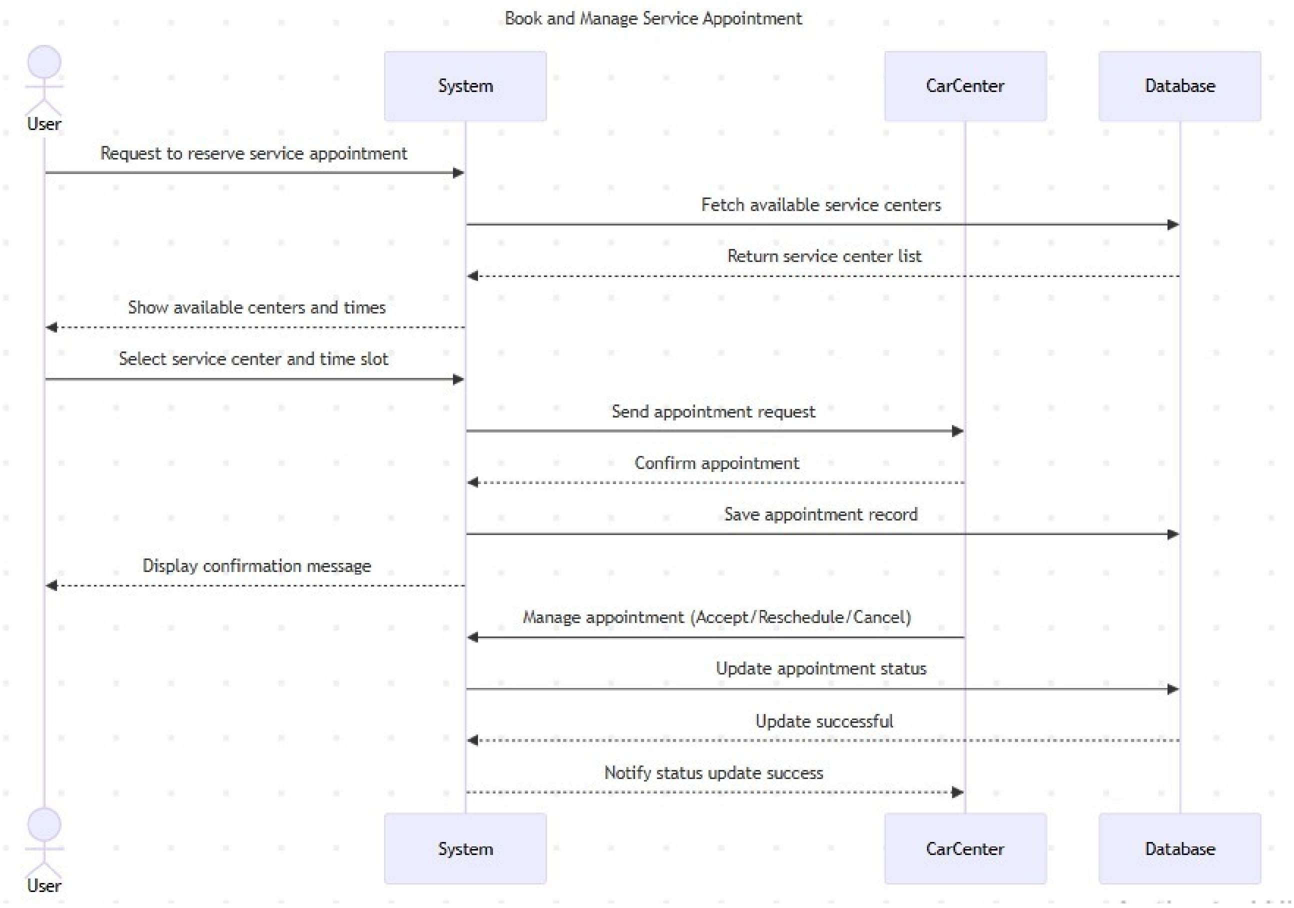

Figure 3.3: Appointment Sequence Diagram

3.6.2 Expense and Payment Sequence Diagram

This diagram describes the sequence of operations for managing expenses and processing payments. After service completion, the service center generates an invoice and submits payment details to the system.
The backend validates the transaction and securely records the payment information. Expense data is updated and stored in the database, allowing users to track service costs and payment history. This sequence ensures transparency and accuracy in financial operations.

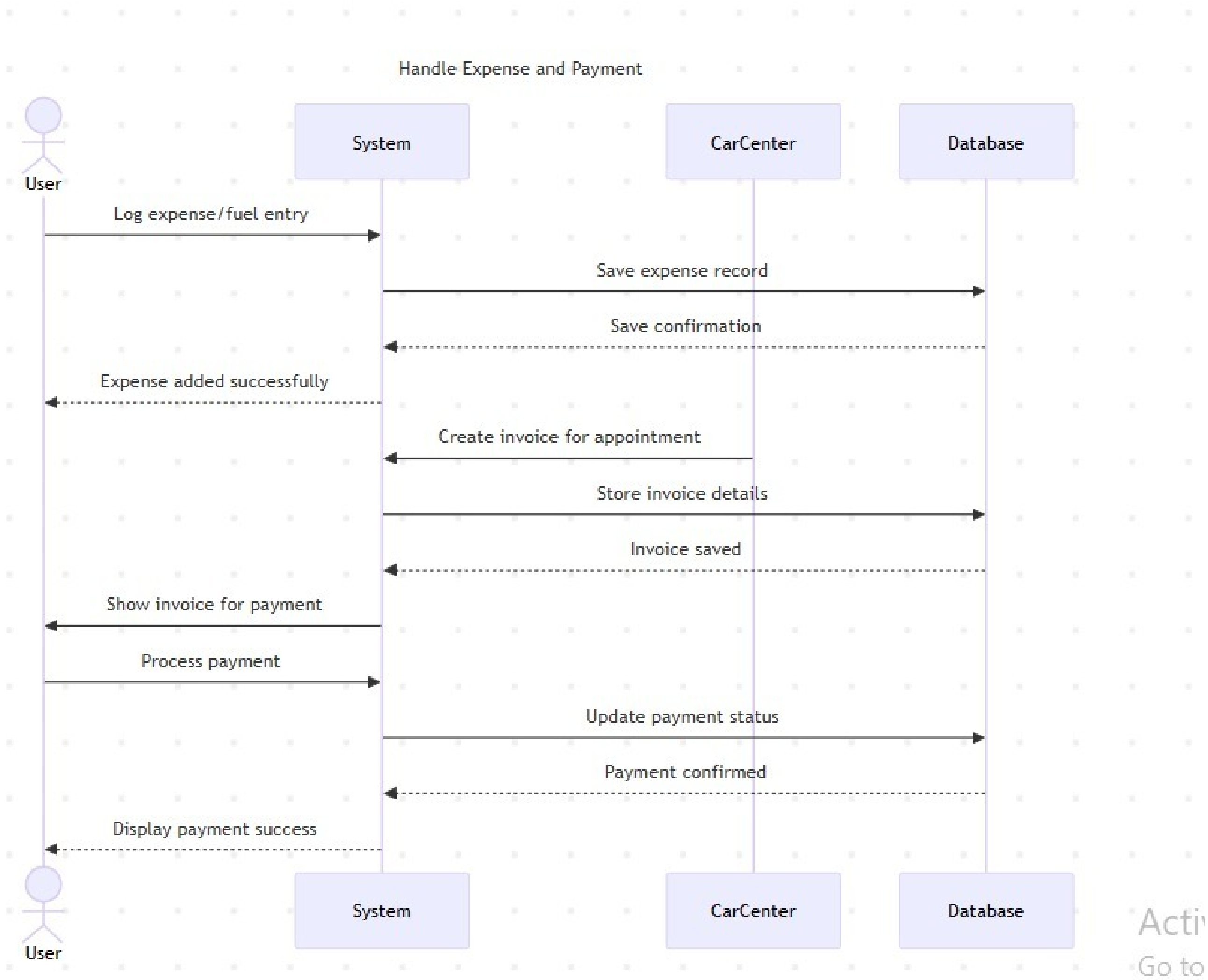

Figure 3.4: Expense and Payment Sequence Diagram

3.6.3 Maintenance Reminder Diagram

The Maintenance Reminder Sequence Diagram explains how the system handles scheduled maintenance reminders. The backend periodically checks vehicle service data and maintenance intervals.

When a maintenance milestone is reached, the system generates a reminder notification and sends it to the vehicle owner through the mobile application. The reminder is also logged in the database for reference. This process ensures timely vehicle maintenance and improves user awareness.

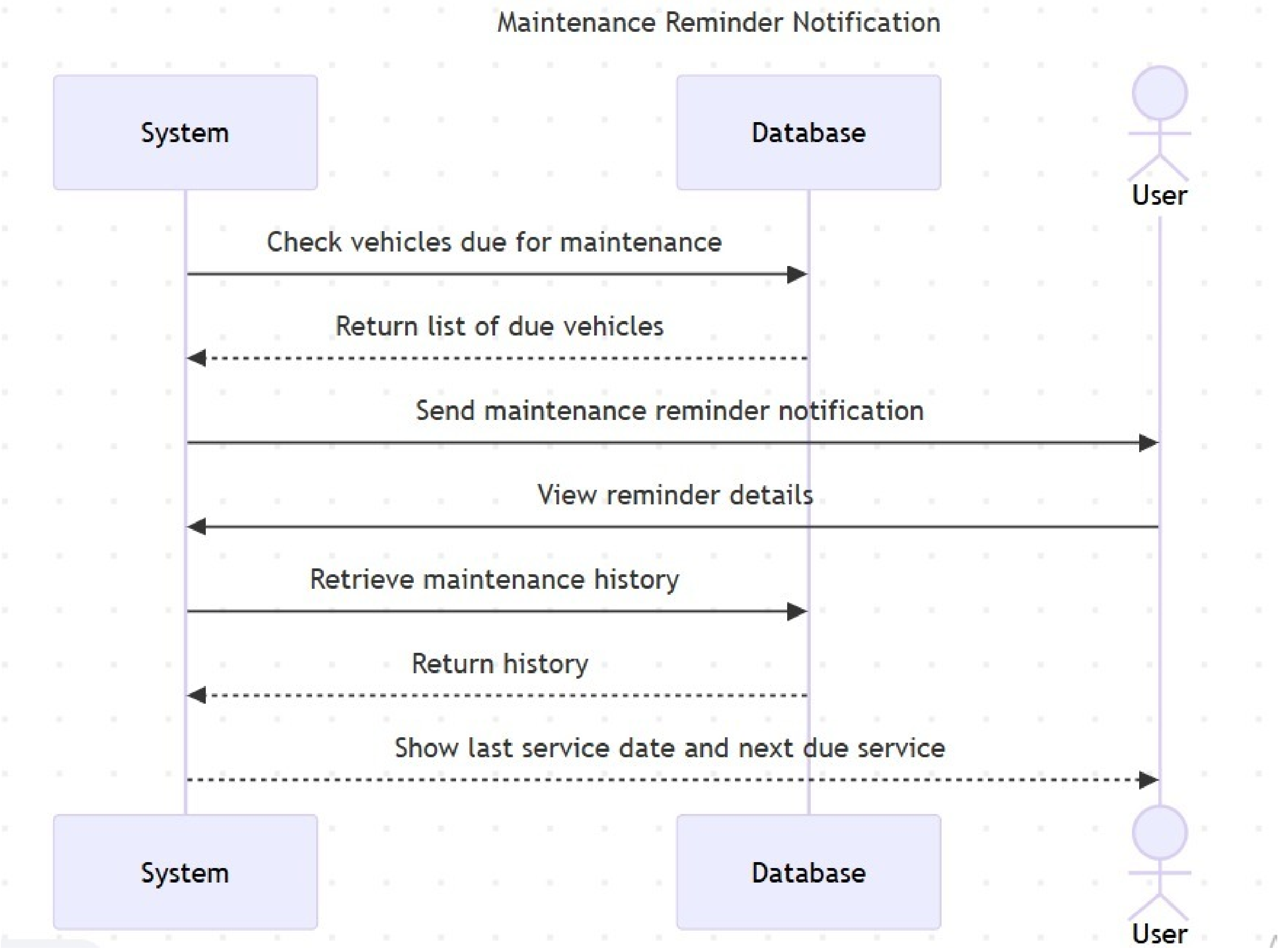

Figure 3.5: Maintenance Reminder Diagram

3.6.4 Notes and Feedback Diagram

This diagram illustrates the process of creating and managing notes and feedback. Vehicle owners can create smart notes using text, voice, or images and submit feedback after service completion.

The system stores the notes and feedback in the database and associates them with the relevant vehicle or service record. Service centers can review feedback to improve service quality. This interaction supports better communication and user engagement.

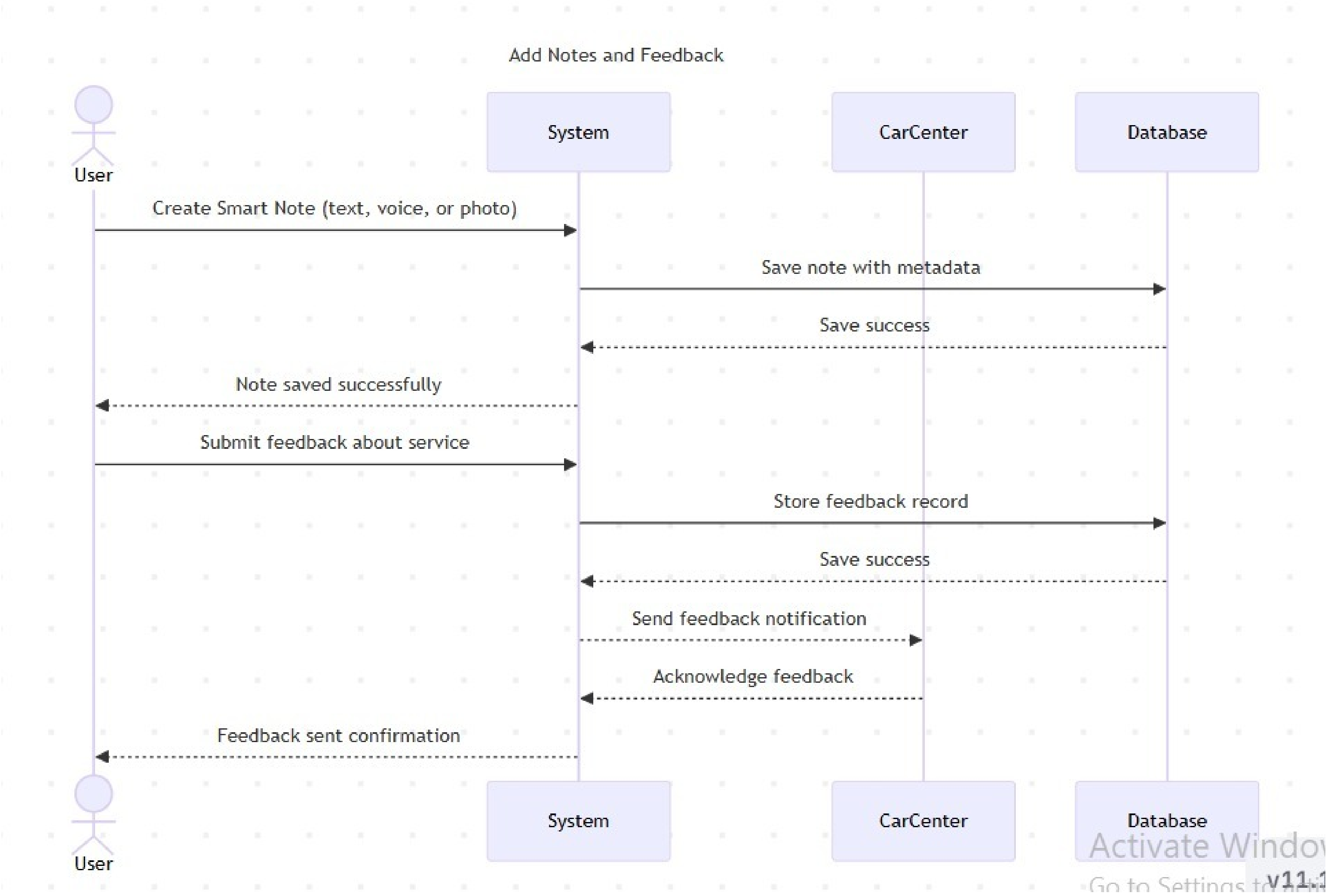

Figure 3.6: Notes and Feedback Diagram

3.6.5 OBD II Diagram

The OBD-II Sequence Diagram shows how the system interacts with the OBD-II device to retrieve live vehicle telemetry and diagnostic data. The user initiates a connection to the OBD-II system through the mobile application.

The system requests telemetry data such as speed, RPM, and temperature from the OBD device. The received data is stored in the database and displayed to the user in real time. If diagnostic error codes are detected, the system retrieves error descriptions and presents explanations and suggested fixes to the user.

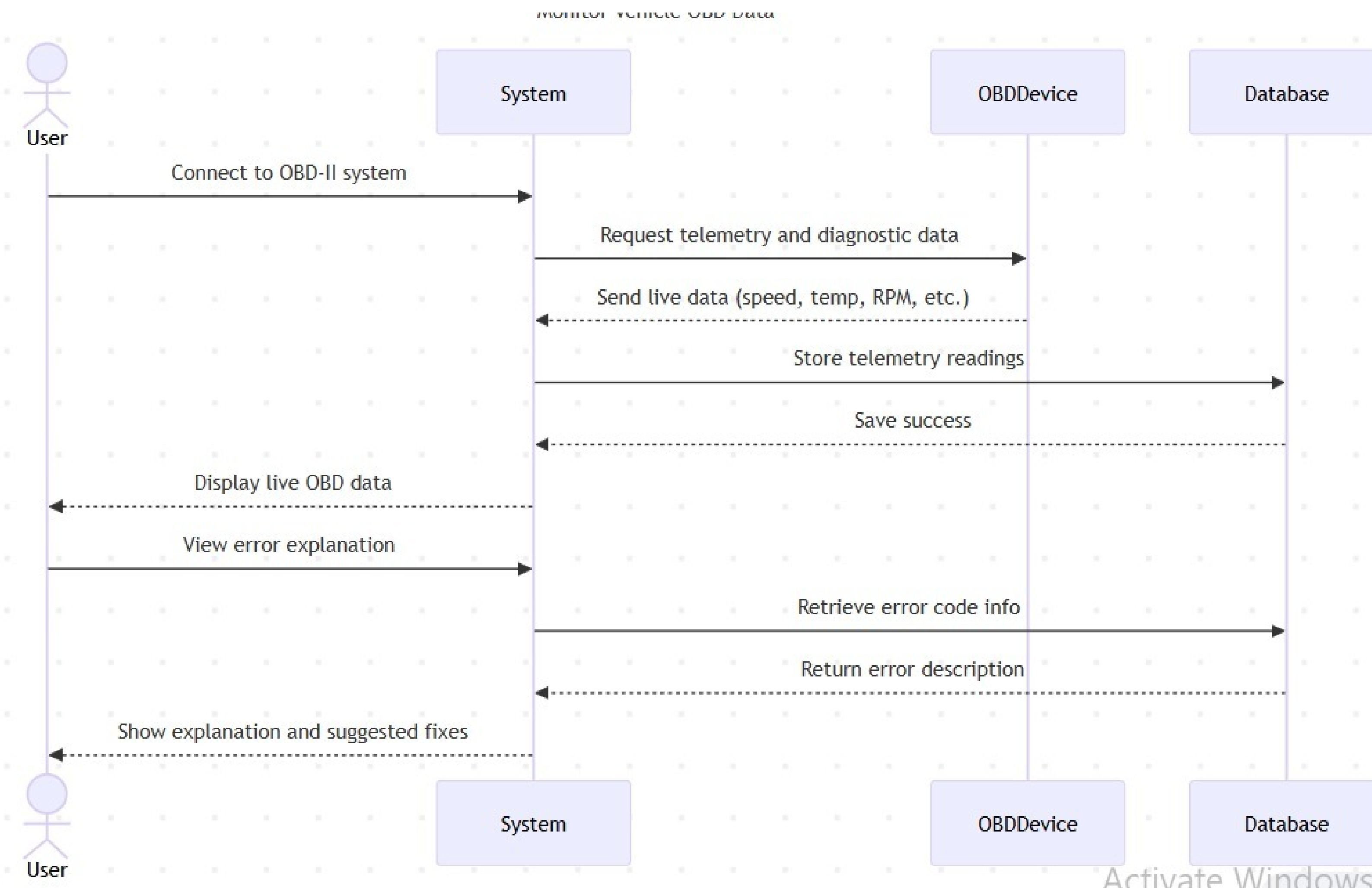

Figure 3.7: OBD II Diagram

3.6.6 Vehicle Management Diagram

This diagram represents the process of adding, updating, and viewing vehicle information. The vehicle owner submits vehicle details such as make, model, year, and plate number.

The system validates the data and stores or updates the vehicle record in the database. Upon request, vehicle details are retrieved and displayed to the user. This sequence ensures accurate vehicle data management and retrieval.

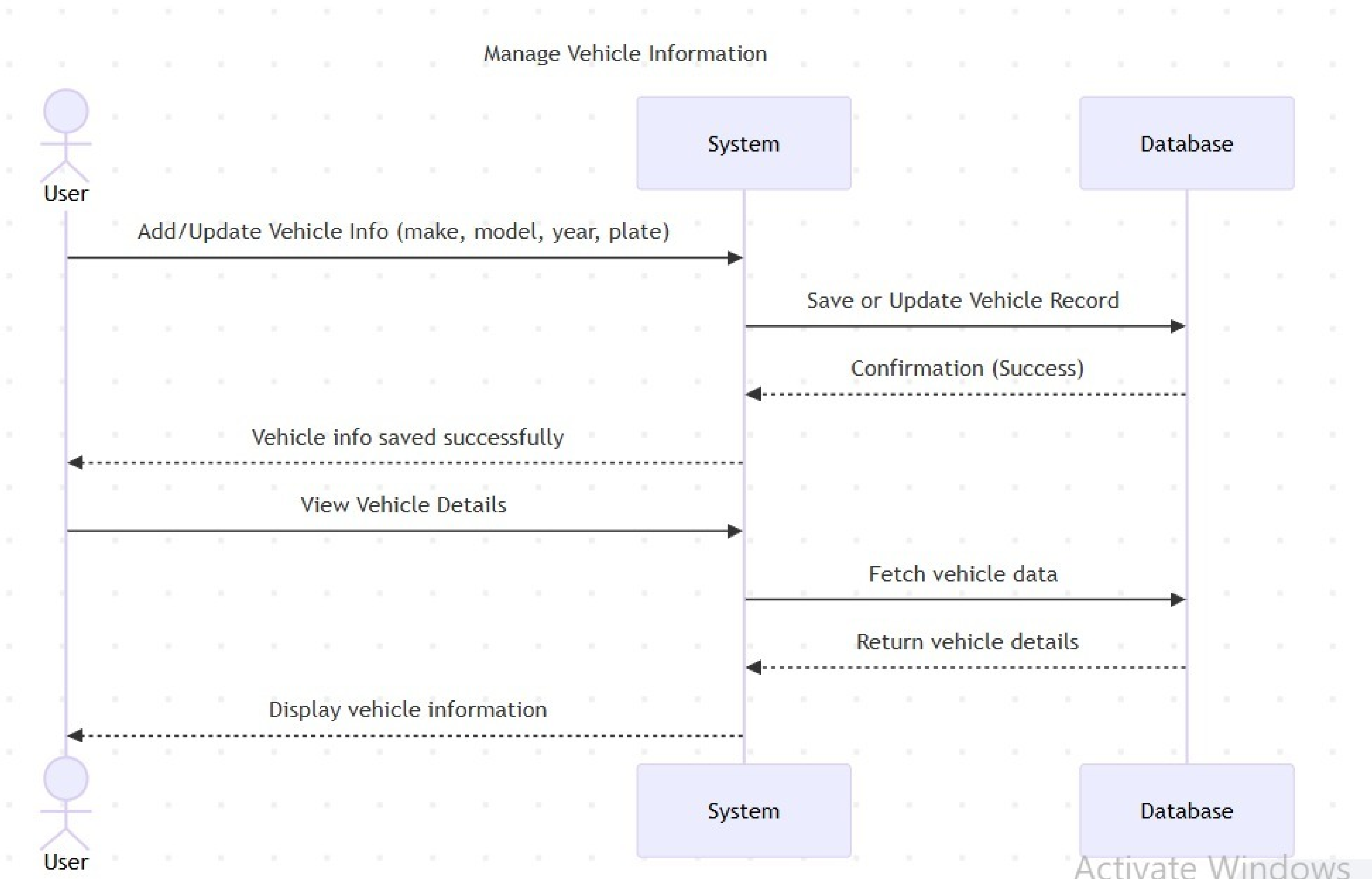

Figure 3.8: Vehicle Management Diagram

3.7 Data-flow Diagram

The Data Flow Diagram (DFD) illustrates how data moves through the CarWare system and how different components interact to support core system functions. External entities such as vehicle owners, service centers, and OBD devices exchange information with the system through defined processes, including vehicle management, appointment scheduling, maintenance tracking, and payment processing.

The diagram clearly identifies the main data inputs and outputs, such as vehicle diagnostic data, service requests, maintenance records, invoices, and payment confirmations. Data storage components, including user profiles, vehicle records, and transaction histories, are shown to demonstrate how information is stored and retrieved during system operation.

By visualizing the flow of data between entities, processes, and data stores, the DFD provides a clear understanding of system behavior and information dependencies. This representation helps ensure data consistency, supports efficient system design, and assists in validating that all functional requirements are properly addressed within the CarWare platform.

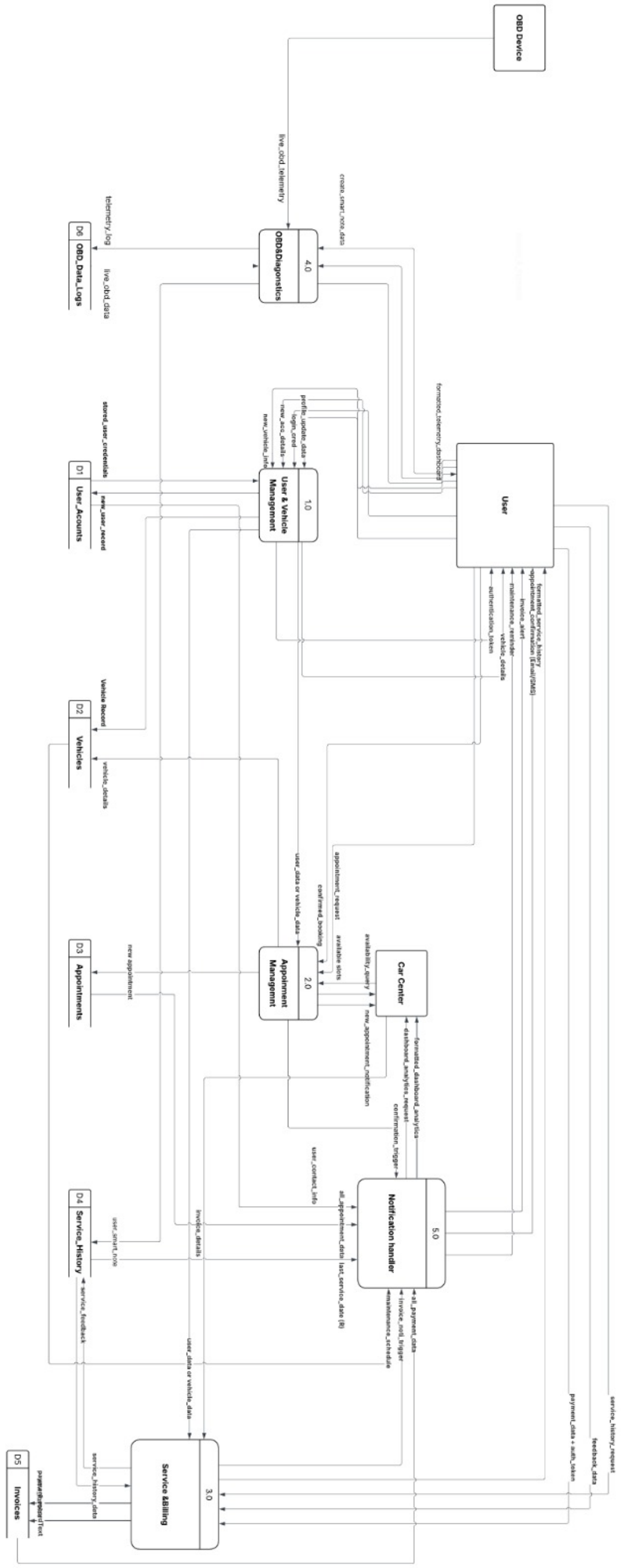

Figure 3.9: Data-Flow Diagram

3.8 Activity Diagram

The Activity Diagram represents the overall workflow of the CarWare system, starting from user authentication and extending through service completion. It illustrates both sequential and parallel activities, including vehicle management, OBD data monitoring, appointment scheduling, service confirmation, repair progress tracking, payment processing, and feedback submission.

Decision points within the diagram highlight alternative system paths, such as appointment acceptance or rejection by the service center and service availability based on workload or resources. By presenting a high-level view of user and system interactions, this diagram clarifies the operational flow of the system and supports a better understanding of how different components collaborate to deliver a complete vehicle service experience.

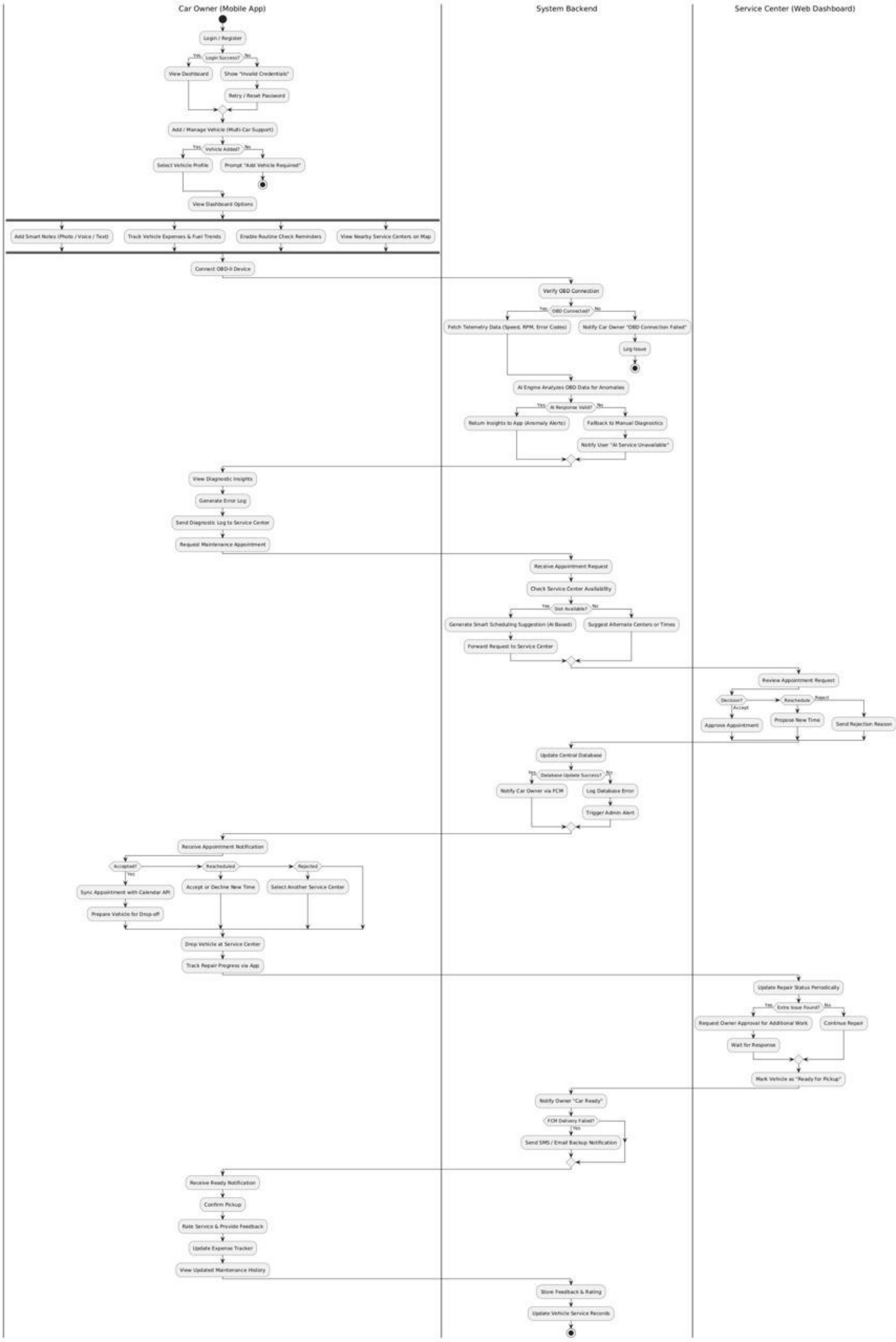

Figure 3.10: Activity Diagram

References

[1] R. Bosch GmbH, *Automotive Handbook*, 10th ed., Wiley, 2018.

[2] SAE International, "On-Board Diagnostics II (OBD-II) Standards," SAE J1979, 2016.

[3] S. Lee, J. Kim, and H. Park, "Design and implementation of a vehicle maintenance management system," *IEEE Access*, vol. 8, 2020.

[4] Google, "Android Developers Documentation," 2024. [Online]. Available: `https://developer.android.com`

[5] Microsoft, "ASP.NET Core Documentation," 2024. [Online]. Available: `https://learn.microsoft.com/aspnet/core`

[6] S. Kumar and R. Gupta, "Web-based service management systems for automotive applications," *IJERT*, 2020.