Car Care 360

CRM System for vehicle Maintenance Centre
Software Architecture Document

Version 1.0

Project ID: 14
Group No: 04
Mentor:

- Dr. Buddhika Karunarathne

Group Members:

- CHANDRASEKARA C.A.H.M.P.P 200084L
- CHATHUNKA W.A.P. 200089G
- CHINTHANA S.K.G 200093M

Contributions

- **CHANDRASEKARA C.A.H.M.P.P 200084L**
  - Introduction
  - User activity diagrams
  - Maintenance Manager Activity Diagrams
  - Marketing team activity diagrams
  - Admin activity diagrams
  - System Size
  - System performance

- **CHATHUNKA W.A.P. 200089G**
  - Architectural Representation
  - Architectural Goals and Constraints
  - Sequence Diagrams
  - Package Diagram
  - Quality
- **CHINTHANA S.K.G 200093M**
  - Use-Case View
  - Logical View
  - Deployment View
  - Implementation View

Revision History

Date	Version	Description	Author
27/08/2023	1.0	Initial version of SAD	• CHANDRASEKARA C.A.H.M.P.P 200084L • CHINTHANA S.K.G 200093M • CHATHUNKA W.A.P. 200089G

Table of Contents

1. Introduction 5

1.1 Purpose 5
1.2 Scope 5
1.3 Definitions, Acronyms, and Abbreviations 5
1.4 References 5
1.5 Overview 6

2. Architectural Representation 6

3. Architectural Goals and Constraints 6

4. Use-Case View 7
4.1 Use-Case Realizations 14

5. Logical View 14
5.1 Overview 20
5.2 Architecturally Significant Design Packages 20

6. Process View 22
6.1 Activity Diagrams 22
6.2 Sequence diagrams 33

7. Deployment View 44

8. Implementation View 45
8.1 Overview 45
8.2 Layers 46

9. Data View (optional) 47

10. Size and Performance 47

11. Quality 47

Software Architecture Document

1. Introduction

1.1 Purpose

The purpose of this Software Architecture Document is to provide a comprehensive overview of the architectural design and structure of the proposed CRM system for a vehicle maintenance center. This document outlines the system's complete architectural overview of the system. The document consists of number of different diagrams including class diagrams, state diagrams and activity diagrams will helps developers to understand the system when they implementing in real world. Apart from developers, this document serves as a reference for all stakeholders involved in the software development lifecycle.

1.2 Scope

This document covers the architectural aspects of the CRM for Vehicle Maintenance System. It serves a high-level view of architecture, components, data flow, and deployment strategy. But this document does not address detailed implementation instructions or code-level details. To make it clear, this Software Architecture Document uses 5 different views also known as 4+1 view model to outline the architecture of the system. That 5 views are Logical view, Process view, Use Case view, Implementation view, Deployment view. Multiple different diagrams have been used including use case diagrams, class diagrams, sequence diagrams, ER diagrams for database and package diagrams.

1.3 Definitions, Acronyms, and Abbreviations

Term, Acronym, Abbreviation	**Definition**
SAD	Software Architecture Document
UI	User Interface
IEEE	Institute of Electrical and Electronics Engineers
Stakeholder	Any party who are interested, affected, or involved in the project
CDN	Content Delivery Network

1.4 References

All diagrams were drawn using Lucid Chart diagramming software.

Lucidchart, “Online diagram software & visual solution | Lucidchart,” *Lucidchart*, 2023. https://www.lucidchart.com/pages/

1.5 Overview

The following sections of this document provide a comprehensive description of the software architecture for the CRM for Vehicle Maintenance System. Architectural Representation section describes what software architecture is for the current system using 4+1 view model. It is represented well using different views including Use-Case, Logical, Process, Deployment, and Implementation View. Architectural goals and constraints section describes the software requirements and objectives that have some significant impact on the architecture including, safety, security and privacy. It also captures the special constraints that may apply. Apart from above architectural views, the size and performance and software quality concerns of the software architecture are also addressed in latter part of this document.

2. Architectural Representation

The Architectural Representation section provides an overview of the software architecture for the current system, utilizing the 4+1 view model. It is comprehensively depicted through various viewpoints, encompassing Use-Case, Logical, Process, Deployment, and Implementation Views.

Use-Case View

This perspective illuminates interactions between users and the system. It elaborates on diverse usage scenarios, illustrating how various actors engage with the software for specific functions.

Logical View

Focusing on structural and functional elements, this view portrays the system's core components. It employs diagrams to showcase relationships, providing clarity on data flow and process organization.

Process View

This dynamic view emphasizes sequences of actions and interactions. It employs diagrams to display how roles interact and how processes unfold in response to user actions.

Deployment View

Highlighting system distribution in a physical environment, this view outlines server configurations, network specifics, and component allocation to hardware resources.

Implementation View

This view delves into the software's internal structure, revealing the organization of classes and components in the codebase. It offers insights into software design, coding patterns, and third-party library integration.

3. Architectural Goals and Constraints

The architectural foundation of the CRM for Vehicle Maintenance system is rooted in key goals and constraints that drive its design and implementation.

Scalability and Extensibility:

Given the dynamic nature of the vehicle maintenance industry, the system should be designed to accommodate growth and evolving requirements. Scalability and extensibility are paramount, enabling seamless integration of new functionalities and efficient expansion to meet changing business needs.

Portability and Accessibility:

To cater to diverse user preferences and device choices, the system should be designed for cross-device compatibility. It ensures consistent performance across various platforms.

Loose Coupling and Reusability:

Components within the system are designed with loose coupling, allowing independent functionality and enhancing reusability. Reusable modules enable efficient development and integration of new features.

Time-Conscious Development:

The system is designed to meet objectives within a defined timeline, balancing feature-rich implementation with project completion.

4. Use-Case View

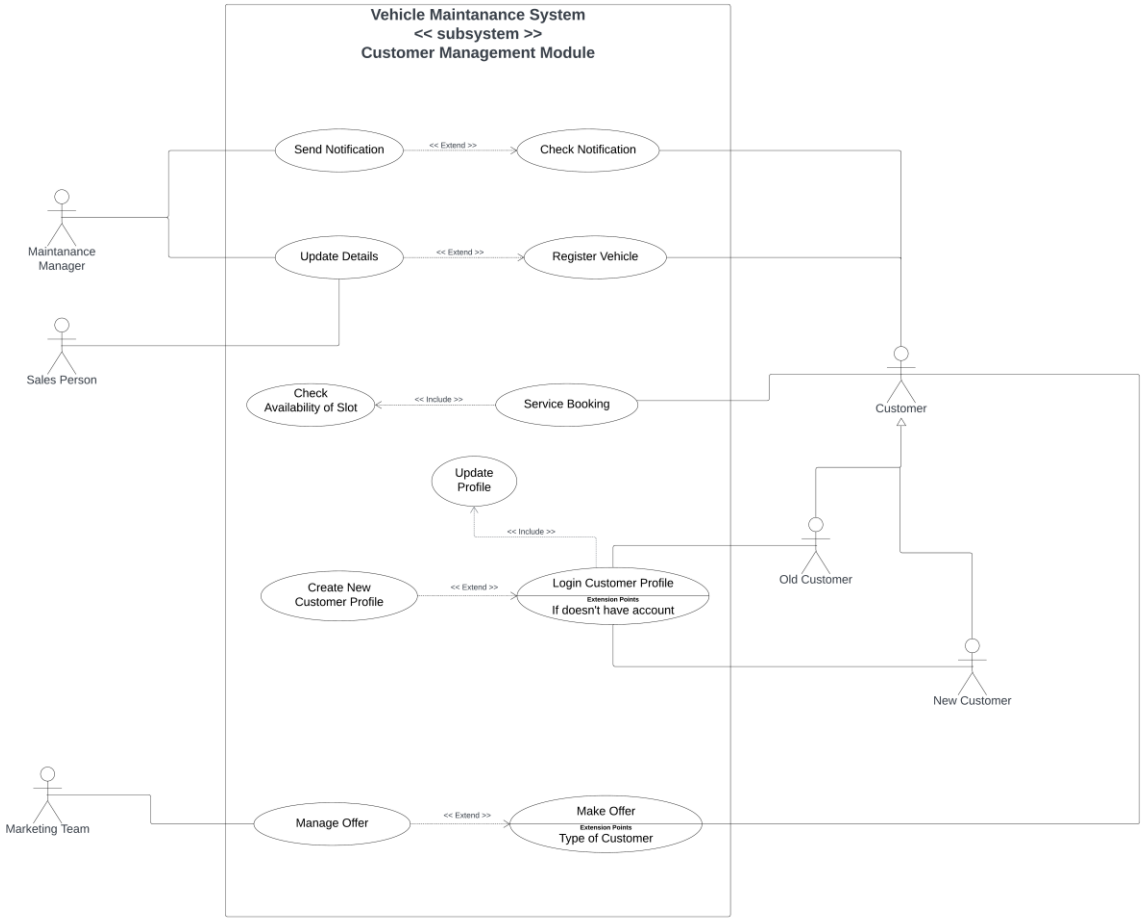

Figure 4.1 illustrating the Customer Management subsystem of vehicle maintenance system. It shows customer login, vehicle register, service booking, and customer offer tasks performed by the customers.

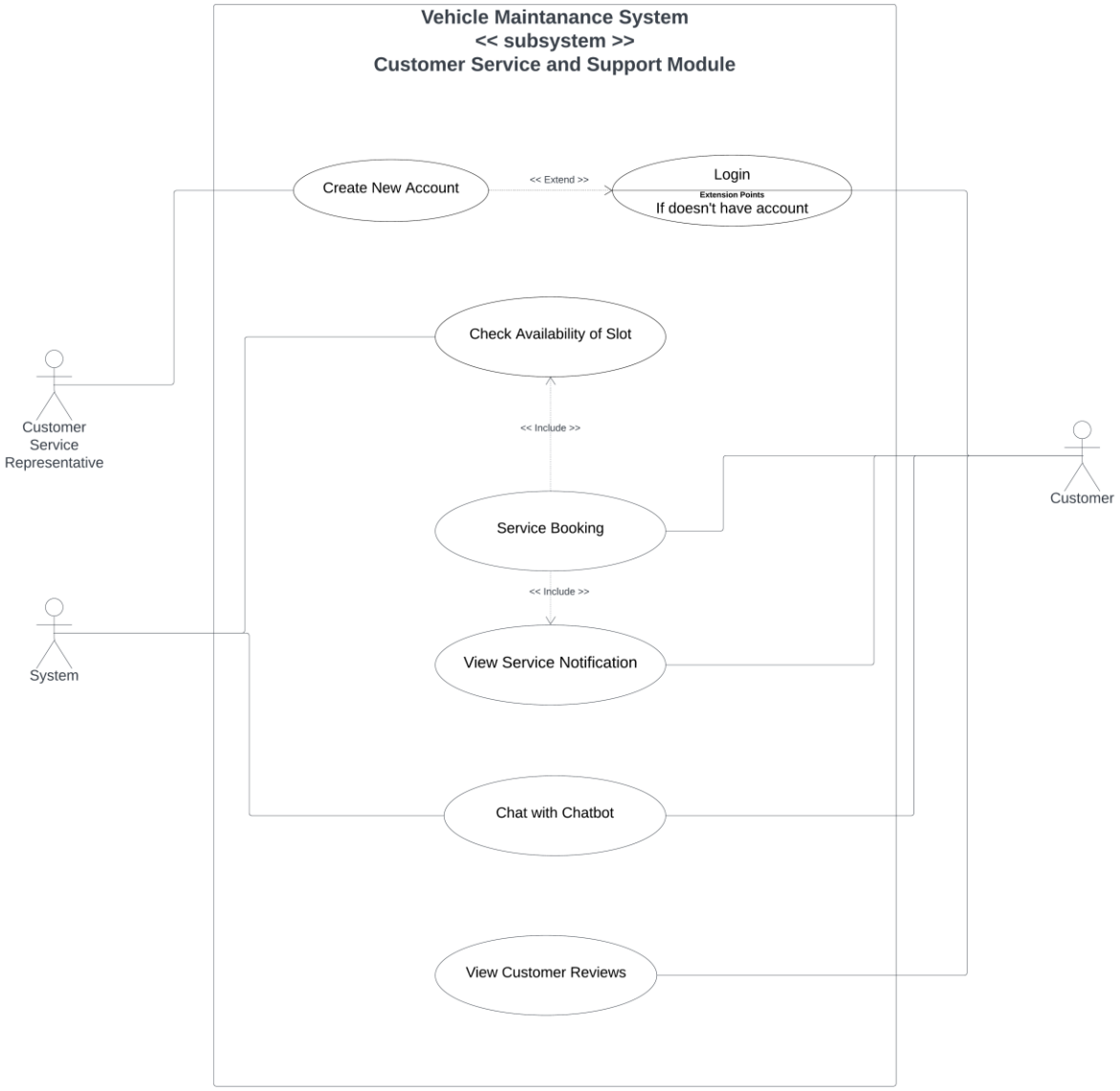

Figure 4.2 illustrating the Customer Service and Support subsystem of vehicle maintenance system. It shows facilities provided by the system to customers.

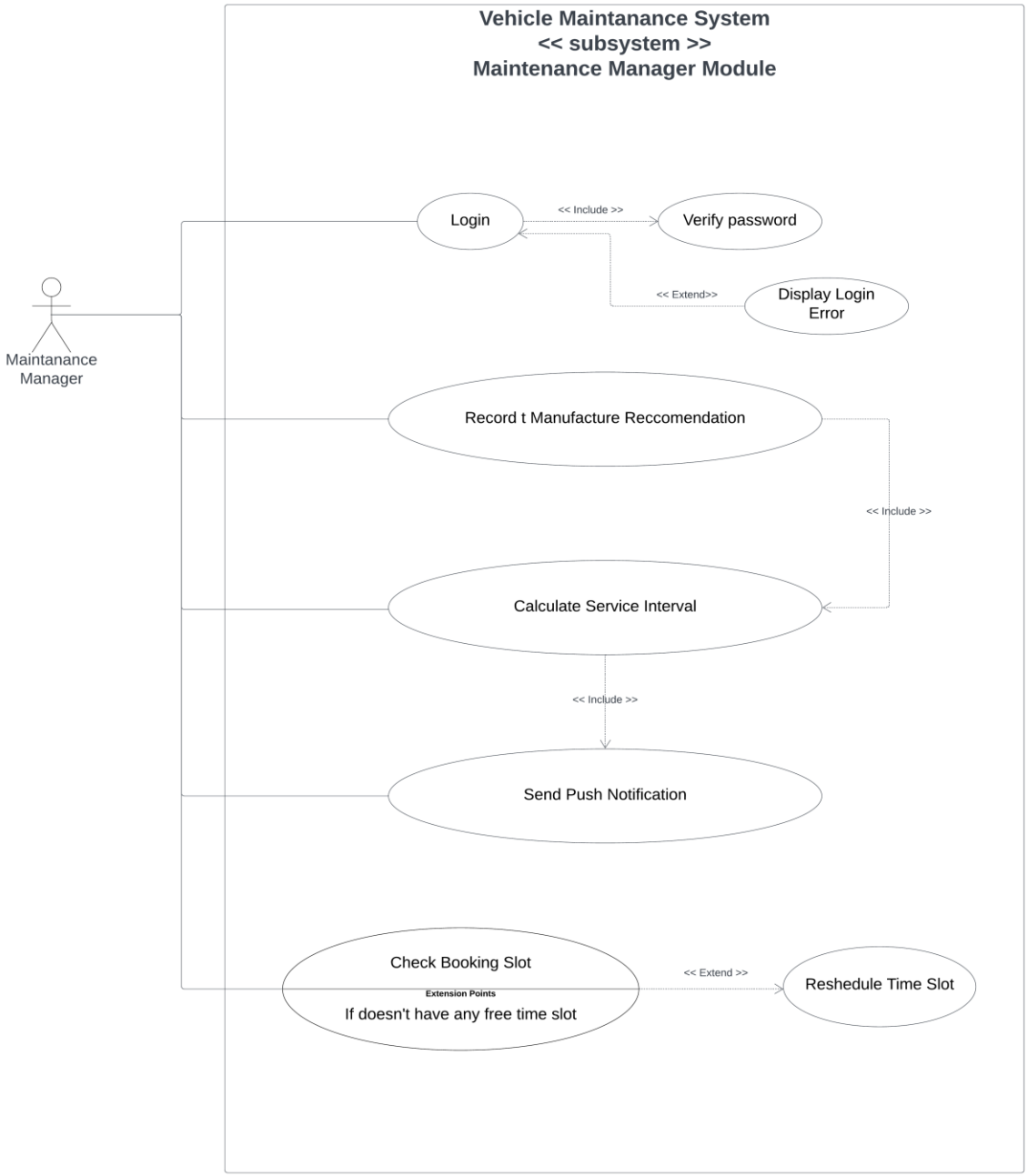

Figure 4.3 illustrating the Maintenance Manager subsystem of vehicle maintenance system. It shows the duty of the maintenance manager.

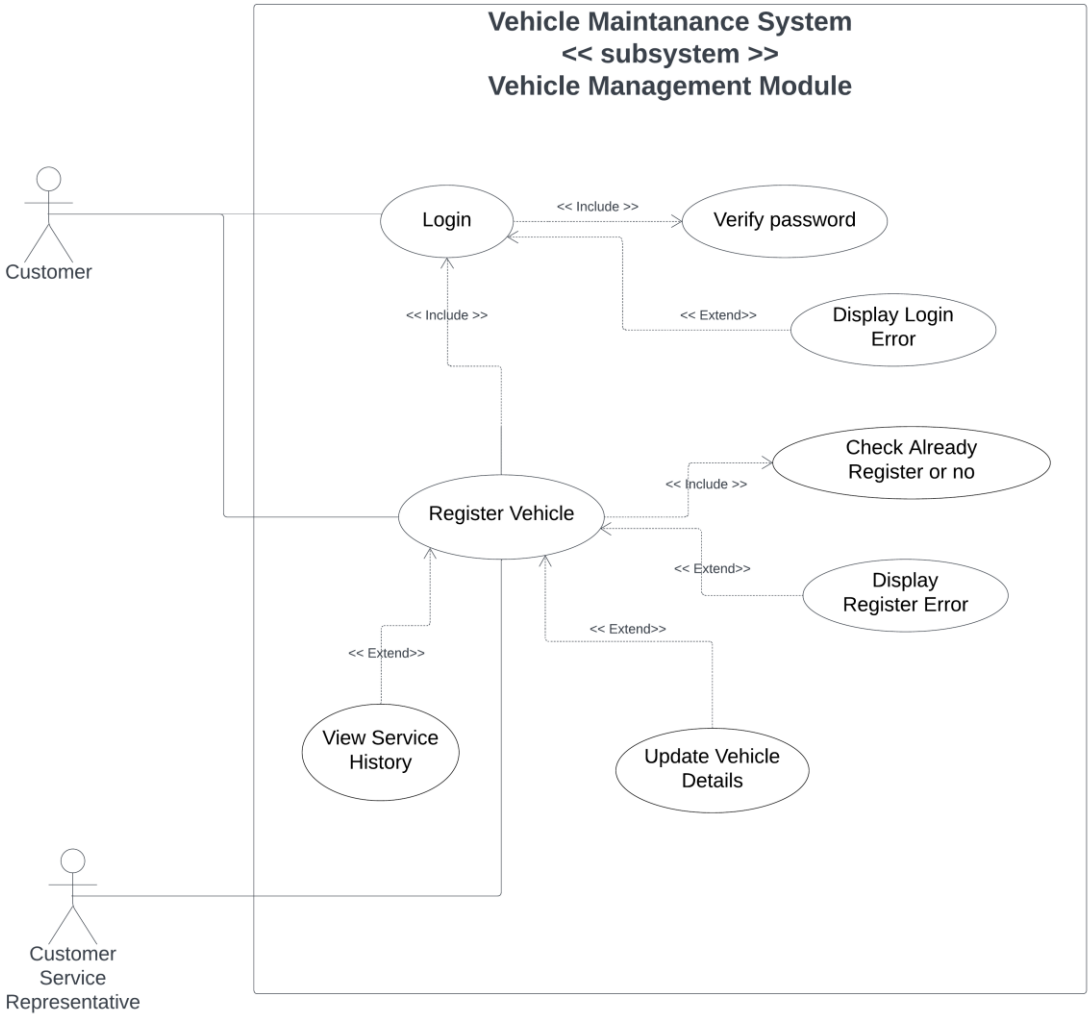

Figure 4.4 illustrating the Vehicle Management subsystem of vehicle maintenance system. It shows the procedure of vehicle registration of the CRM vehicle maintenance system.

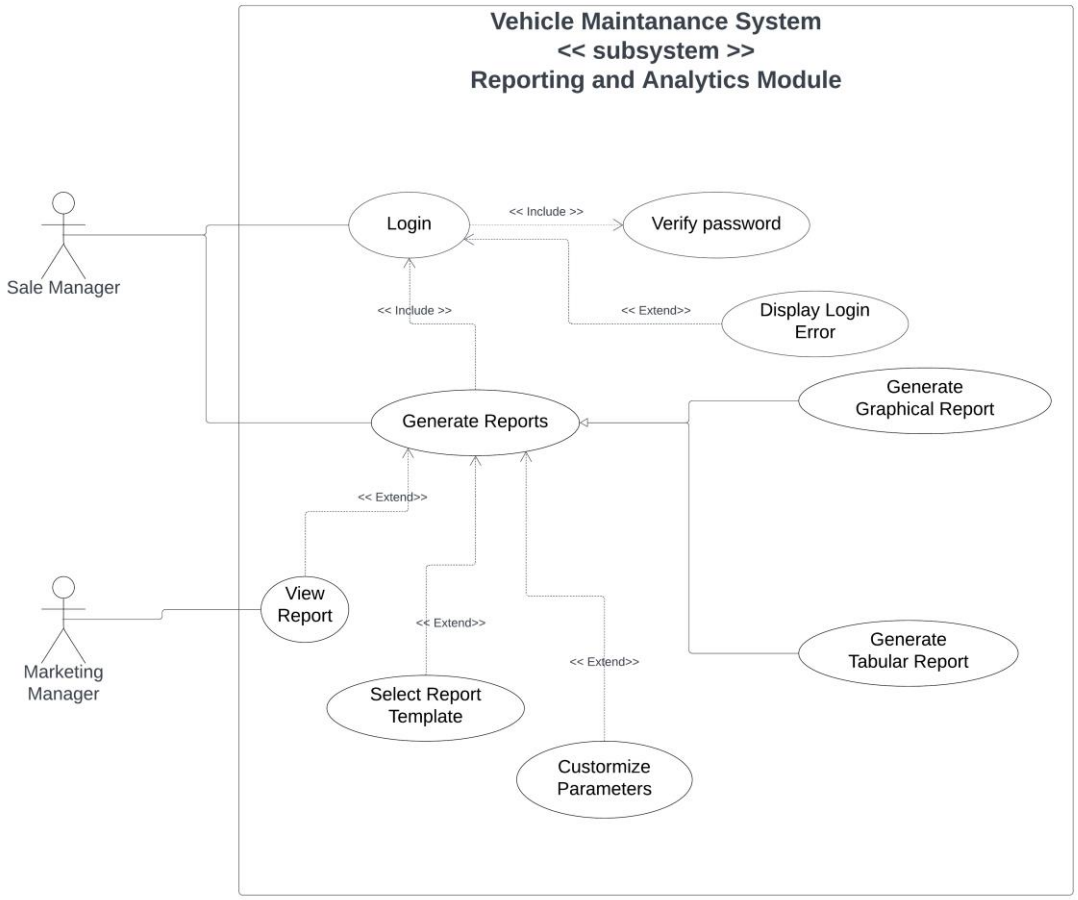

Figure 4.5 illustrating the Reporting and Analytics subsystem of vehicle maintenance system. It demonstrates how to report the sales and analyze the generated report by marketing and sales managers.

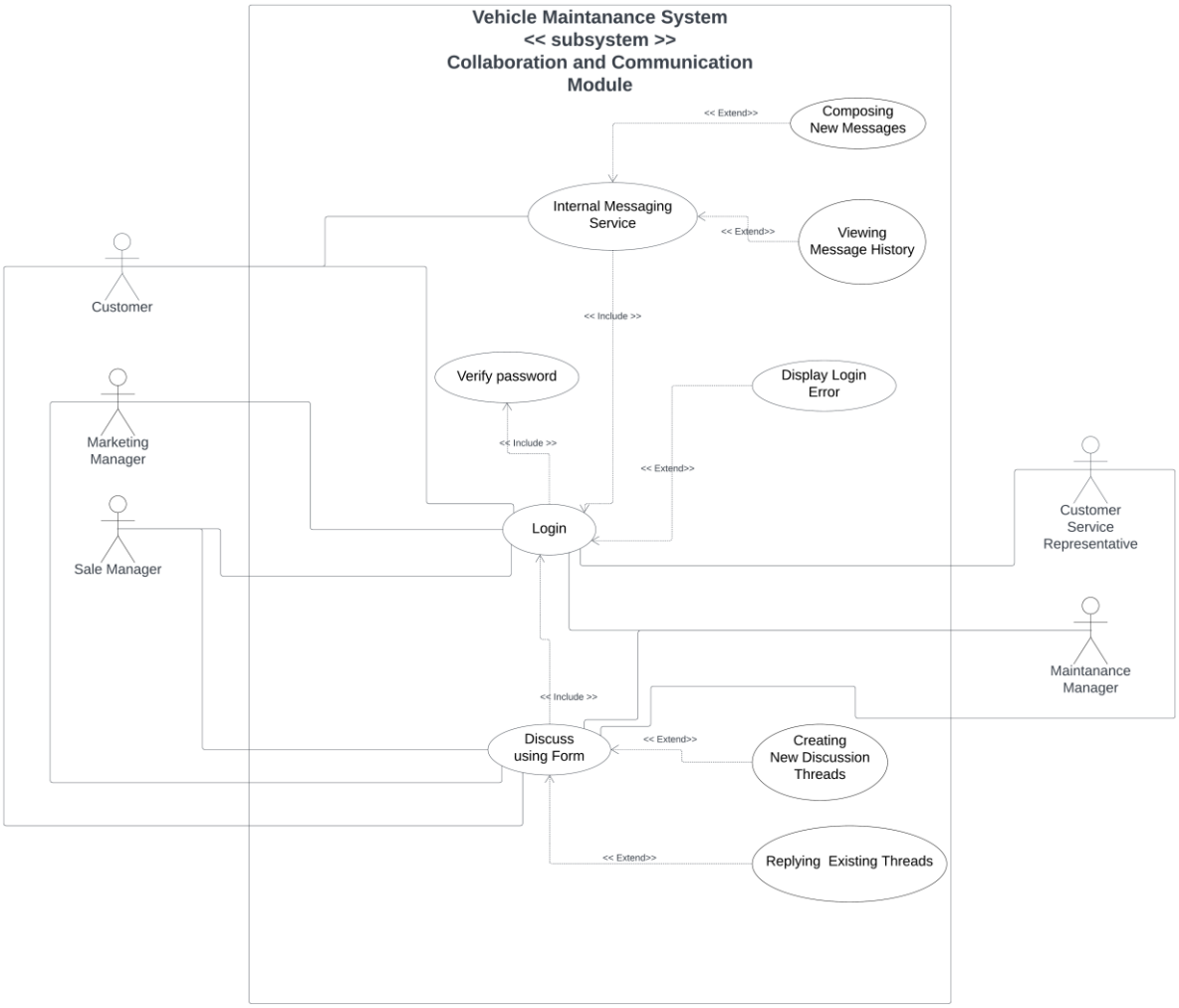

Figure 4.6 illustrating the Collaboration and Communication subsystem of vehicle maintenance system. It describes how to communicate among customers and the board panel of Car Care 360 through the system.

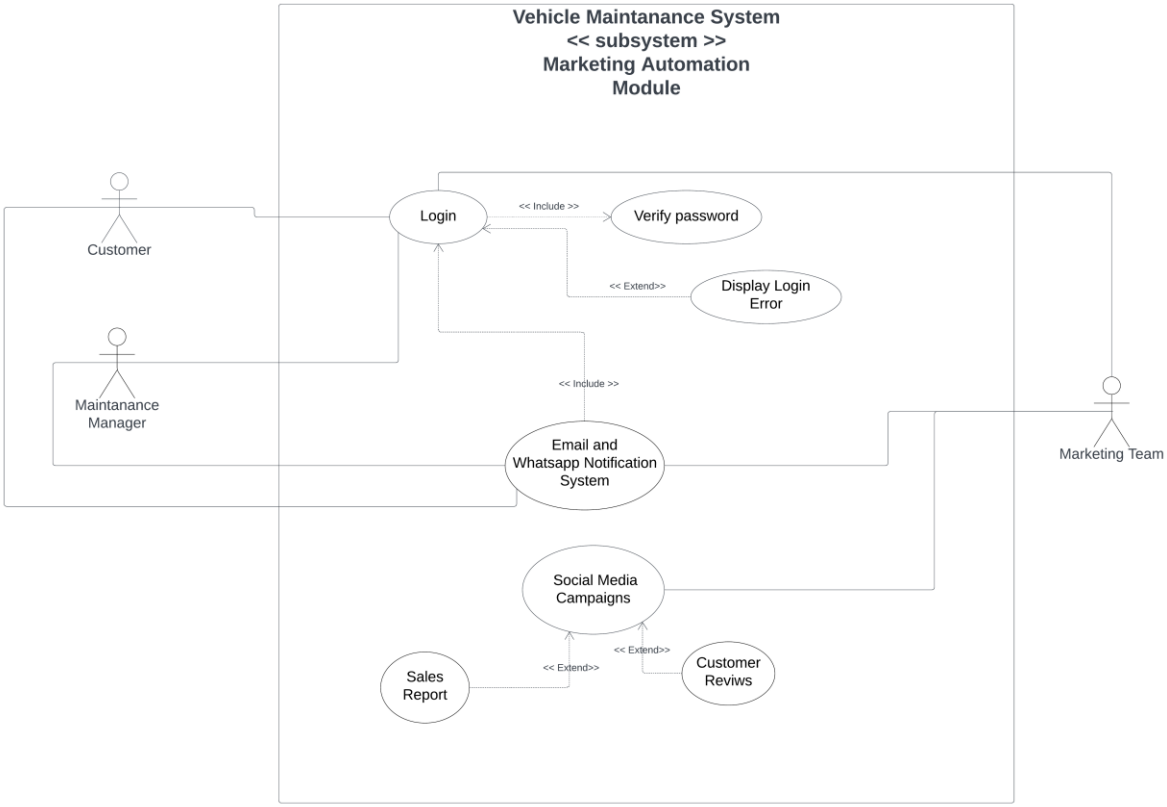

Figure 4.7 illustrating the Marketing Automation subsystem of vehicle maintenance system. It shows how to work maintenance manager and marketing team to improve productivity of a company and how to advertise their brand name among customers.

4.1 Use-Case Realizations

4.1.1 Login Customer Profile

Use case name	Login Customer Profile
Actor	Customer

Description	This is a fundamental functionality of the crm vehicle maintenance system. Customers should register in the system by giving relevant details to register menu or by using their google account. If a customer register using a google account, they have to fill in some extra details.
preconditions	Customer should reach the Car Care 360 landing page
Main flow	1. Customers should choose what register option is perfect for them. 2. Customers should fill in all required details. 3. Customers touch the 'Signup' button and submit the registration form. 4. If customers choose the 'Sign up using Google' option google API provides a popup menu and customers can choose a relevant google account and submit it. After that the process system provides another form to collect extra details. 5. After doing 3 or 4 step customers can login their dashboard
Successful end/post condition	A Massage will be displayed saying "Successfully Create Account"
Fail end/post condition	When customers provide invalid email or telephone number security alert generate by system
Extensions	Customer service representative can provide support ticket interface for register customer

4.1.2 Service Booking

Use case name	Service Booking
Actor	Customer
Description	This is a major functionality of the crm vehicle maintenance system. Customers can apply for the appointment using the web interface to book the time slot for their vehicle maintenance.
preconditions	The already booking time slot should be displayed to the customer
Main flow	1. Customers should register their vehicles. 2. The system displays a booking details page. 3. Customers touch the schedule calendar and select a free slot of remaining slots.

	4. Customers should note their requirements. 5. Customers touch 'Submit' button and confirm their appointment. 6. System displays the schedule time slot and author important details to customer should be known
Successful end/post condition	Booking time slots are displayed in schedule calendar
Fail end/post condition	Some service requirements are not display in display
Extensions	N/A

4.1.3 Register Vehicle

Use case name	Register Vehicle
Actor	Customer
Description	This is a major functionality of the crm vehicle maintenance system. The registered customers can register their vehicle by filling separate form each vehicle.
preconditions	The customer should be registered in the system.
Main flow	1. Customers should create their own profile. 2. Customer should fill all required details. 3. Customers torch the "Register" button and submit form.
Successful end/post condition	A message will be displaying "The vehicle is successfully registered"
Fail end/post condition	A message will be displaying "The vehicle registration is not successful"
Extensions	N/A

4.1.4 Record Manufacture Recommendations

Use case name	Record Manufacture Recommendations
Actor	Maintenance Manager
Description	This is a major functionality of the crm vehicle maintenance system. Maintenance manager should update manufacture recommendations for each model of brands.
preconditions	N/A

Main flow	1. Maintenance manager should login to his/her dashboard. 2. Update information using dashboard.
Successful end/post condition	Displays all recommendation of band of vehicle.
Fail end/post condition	Some recommendations of model are not display in collection
Extensions	N/A

4.1.5 Calculate Service Interval

Use case name	Calculate Service Interval
Actor	Maintenance Manager
Description	Maintenance manager should calculate service interval of each vehicle by using current millage and last service millage
preconditions	The customers should already provide their vehicle current millages.
Main flow	1. Maintenance manager should login to his/her dashboard. 2. Calculate service interval using relevant information.
Successful end/post condition	N/A
Fail end/post condition	N/A
Extensions	Send push message to customer include service information.

4.1.6 Send Push Notification

Use case name	Send Push Notification
Actor	Maintenance Manager
Description	Maintenance manager sends SMS/ WhatsApp message to customer include service details.
preconditions	N/A
Main flow	1. Maintenance manager should login to his/her dashboard. 2. Manager sends push message to each customer separately.
Successful end/post condition	A message will be saying “The message is successfully sent”

Fail end/post condition	A popup error message will be displaying "The massage is not sent"
Extensions	N/A

4.1.7 Generate Reports

Use case name	Generate Reports
Actor	Sales Manager (sales teams)
Description	The sales team is generated sales report using relevant parameter
preconditions	N/A
Main flow	1. The sales manager should login to his/her dashboard. 2. Generate report using templates
Successful end/post condition	Get good idea about sales of company
Fail end/post condition	N/A
Extensions	Display obtained information graphically and tabular chart

4.1.8 Discuss using Forum.

Use case name	Discuss using Forum
Actor	General users
Description	The all users of Car Care 360 system can discuss with each other.
preconditions	The user should have user profile.
Main flow	1. The user should login to his/her dashboard. 2. Create new discussion thread and discuss their problems.
Successful end/post condition	N/A
Fail end/post condition	N/A
Extensions	N/A

4.1.9 Social Media Campaigns

Use case name	Social Media Campaigns
Actor	Marketing team

Description	The marketing team held promotion campaign though social media platform
preconditions	The sales teams should generate sale reports.
Main flow	1. The marketing manager should login to his/her dashboard. 2. Send All advertisement through the social medial platform.
Successful end/post condition	A message will be saying “The add is successfully published”
Fail end/post condition	A message will be saying “The add is discarded”
Extensions	N/A

4.1.10 Manage Offer

Use case name	Manage Offer
Actor	Marketing Team
Description	This is part of the promotional campaign. Marketing team separate the customers three group and provide offer separately.
preconditions	N/A
Main flow	1. The marketing manager should login to his/her dashboard. 2. Marketing team send push message to notify offer
Successful end/post condition	Customers respond to push message.
Fail end/post condition	A message will be saying “The message is discarded”
Extensions	N/A

4.1.11 Chat with Chatbot

Use case name	Chat with Chatbot
Actor	System
Description	This is an optional functionality of the crm vehicle maintenance system. Customers can chat with chatbot and get common information about vehicle and parts of vehicle.
preconditions	Customer should reach the Car Care 360 landing page

Main flow	3. Customer reaches landing page of Car Care 360 web page and can chat with him and get details
Successful end/post condition	N/A
Fail end/post condition	N/A
Extensions	N/A

5. Logical View

5.1 Overview

The logical view of the system presented by the class diagram decomposes the whole system into the relevant classes that relate to one another by various relationships. Here the application has been decomposed into classes as user class and its extending child classes, service history, notification, vehicle, service history, booking, maintenance task and crmvehicle maintenance system with their aggregation and composition related classes.

5.2 Architecturally Significant Design Packages

User class - This is the parent class for the typical user object, and all lower-level user classes, such as Marketing Team, Maintenance Manager, Salespeople, and Customer, will extend it. Viewing notifications and logging in or signing up are the key functions of this class.

Vehicle class - This class shows important information about the system should know about each customer vehicle.

Notification class - This class shows the notifications are produced by the maintenance manager, marketing team, sales team and system.

Chatbot class - This class shows a computer program designed to simulate conversation with human users, especially over the internet.

Booking class - This class is used to handle the vehicle maintenance appointments that are applied by the user.

Service History class - This class is used to record vehicle service information.

Forum and Internal Messaging Service class - Those classes are used to keep up communication among the people in the system.

Maintenance Task class - Maintenance Task class is used to manage the service task by helping of maintenance manager.

Figure 5.1 Class Diagram

6. Process View

6.1 Activity Diagrams

6.1.1 *User activity diagrams*

6.1.1.1 User registration

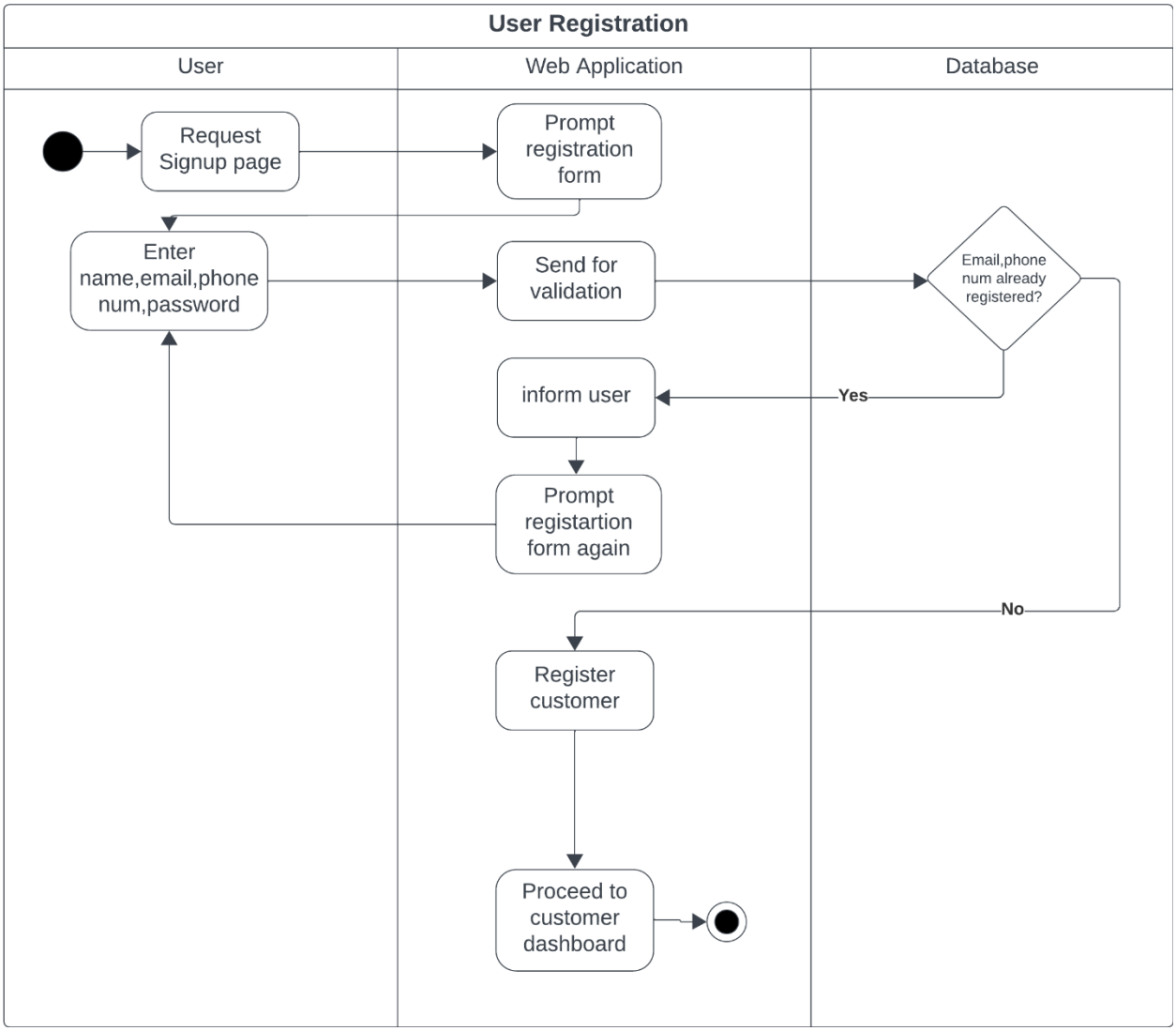

Figure 6.1 Activity Diagram for User registration

6.1.1.2 User Login

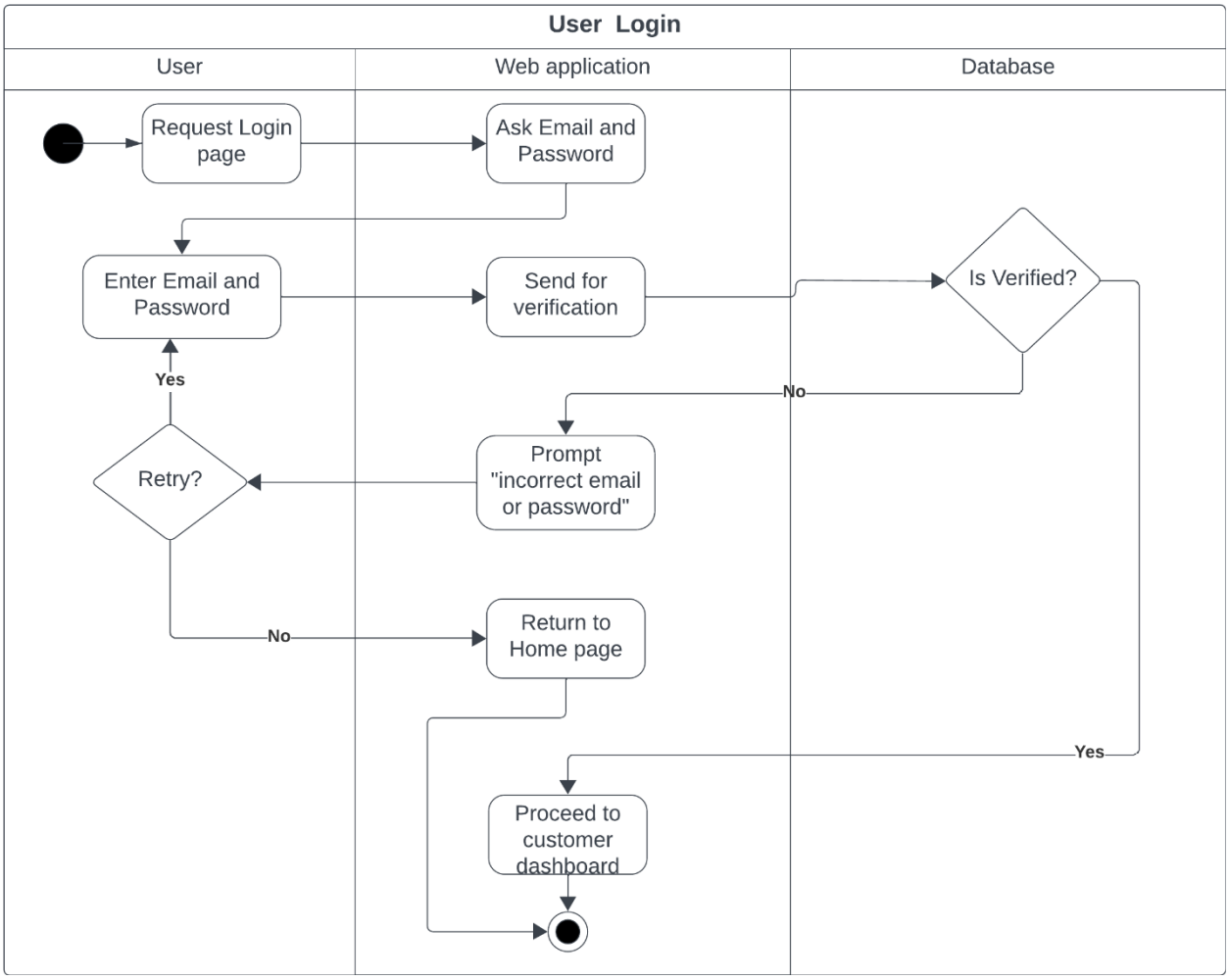

Figure 6.2 Activity Diagram for User login

6.1.1.3 Registering a vehicle on the system

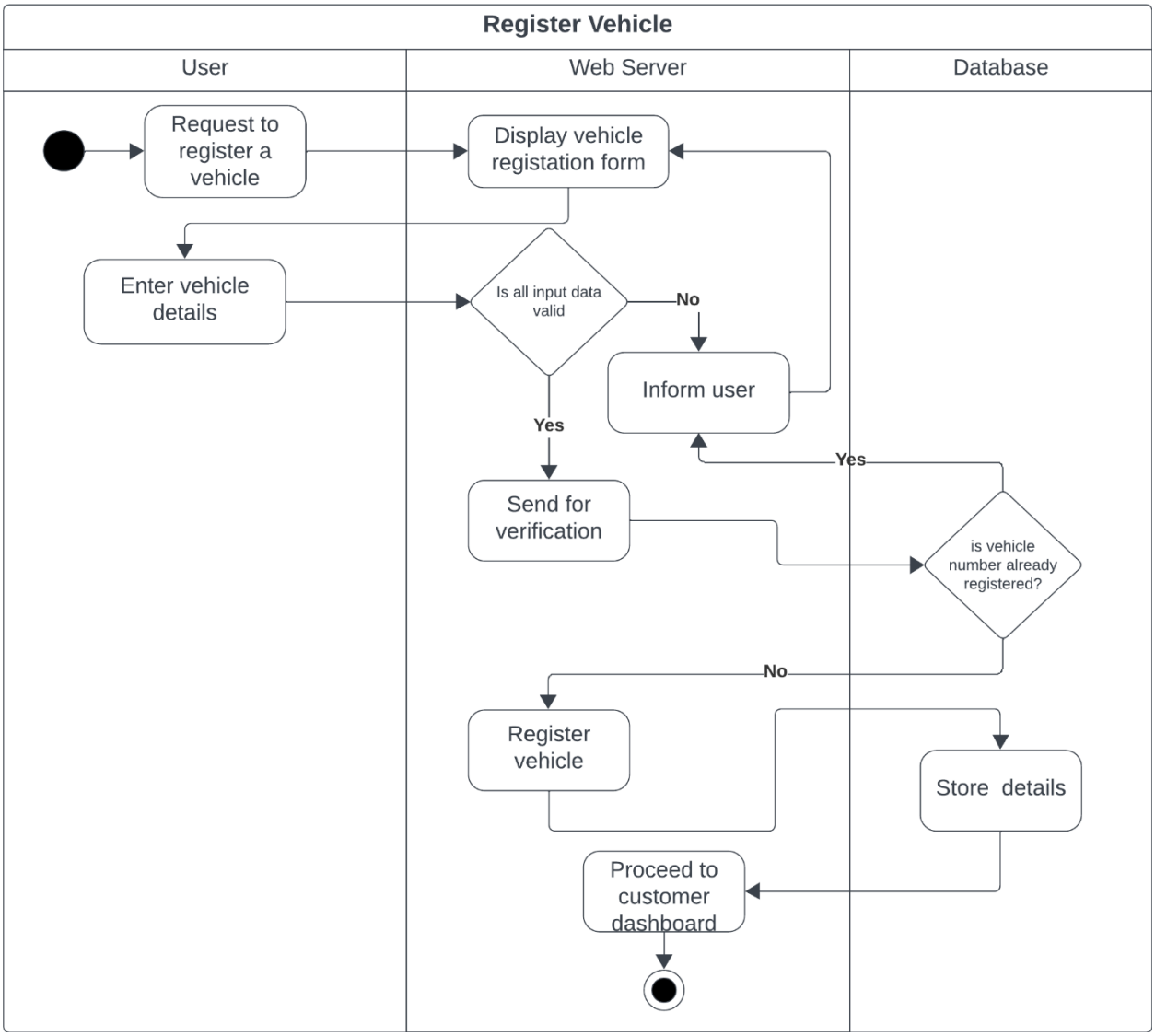

Figure 6.3 Activity Diagram for Registering a vehicle.

6.1.1.4 Booking a service slot

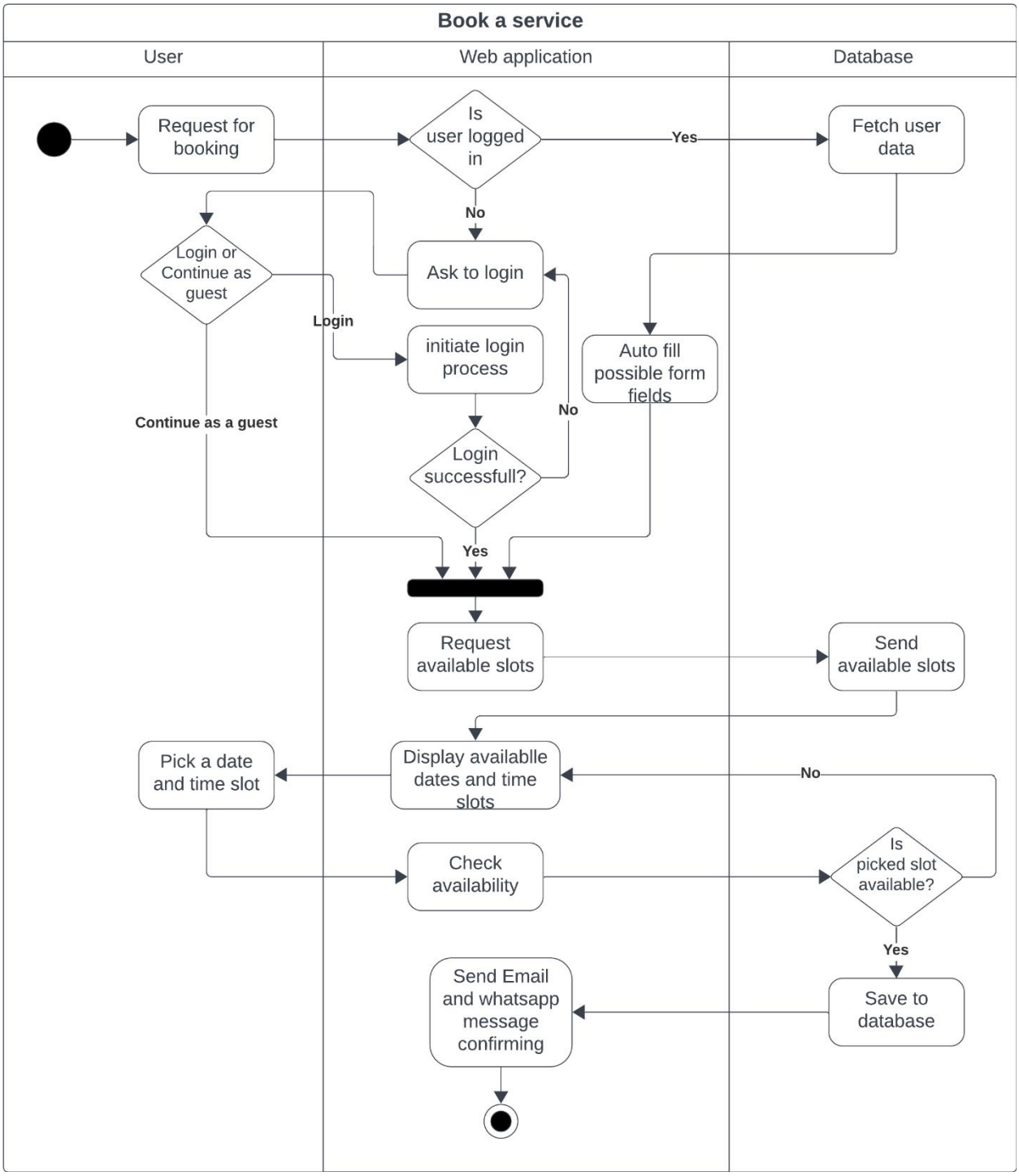

Figure 6.4 Activity Diagram for booking a service slot.

6.1.1.5 Send current mileage of the vehicle through WhatsApp on each month end.

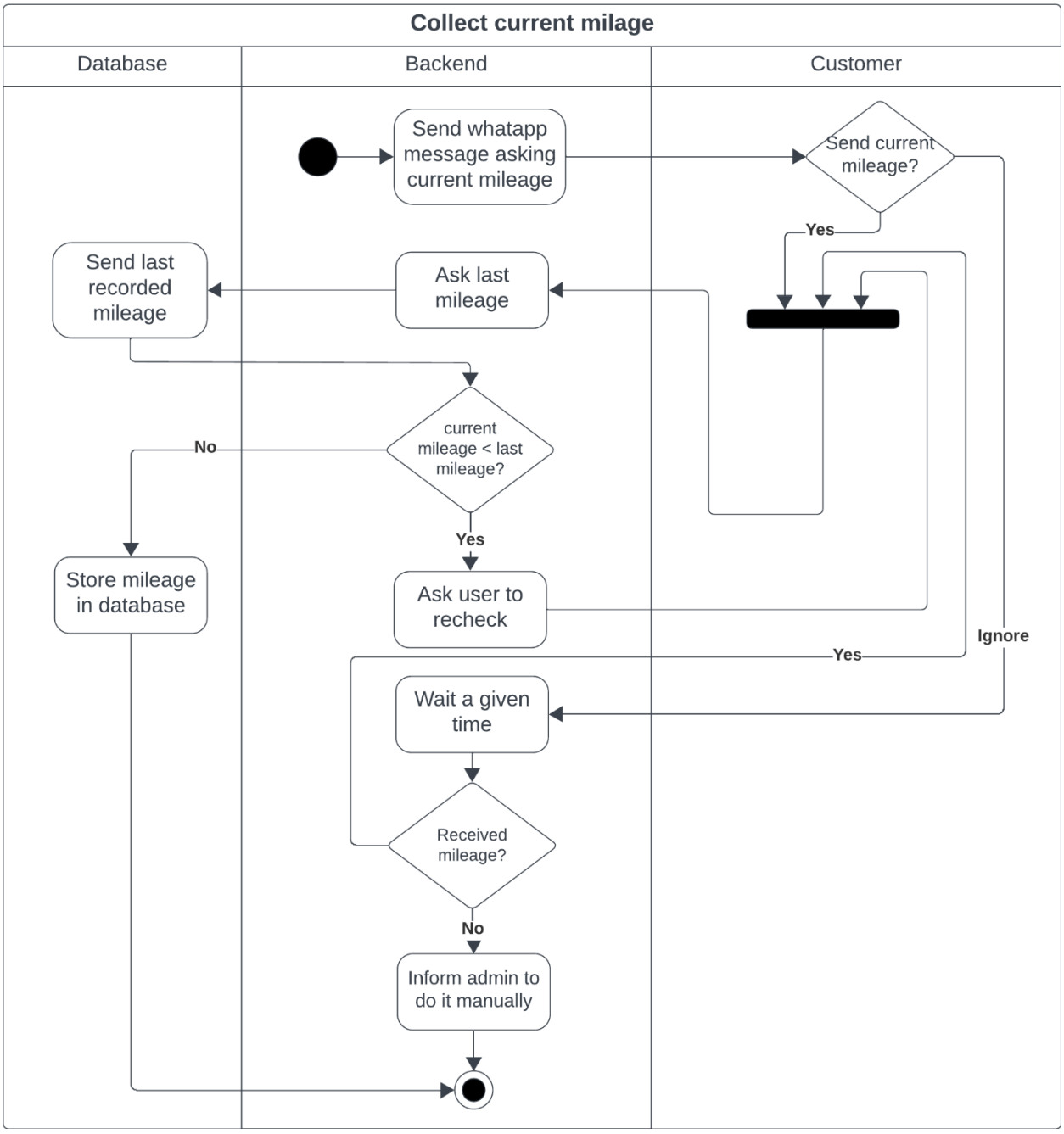

Figure 6.5 Activity Diagram for Sending current mileage.

6.1.2 *Maintenance Manager Activity Diagrams*

6.1.2.1 Enter manufacturer recommendations of each vehicle model.

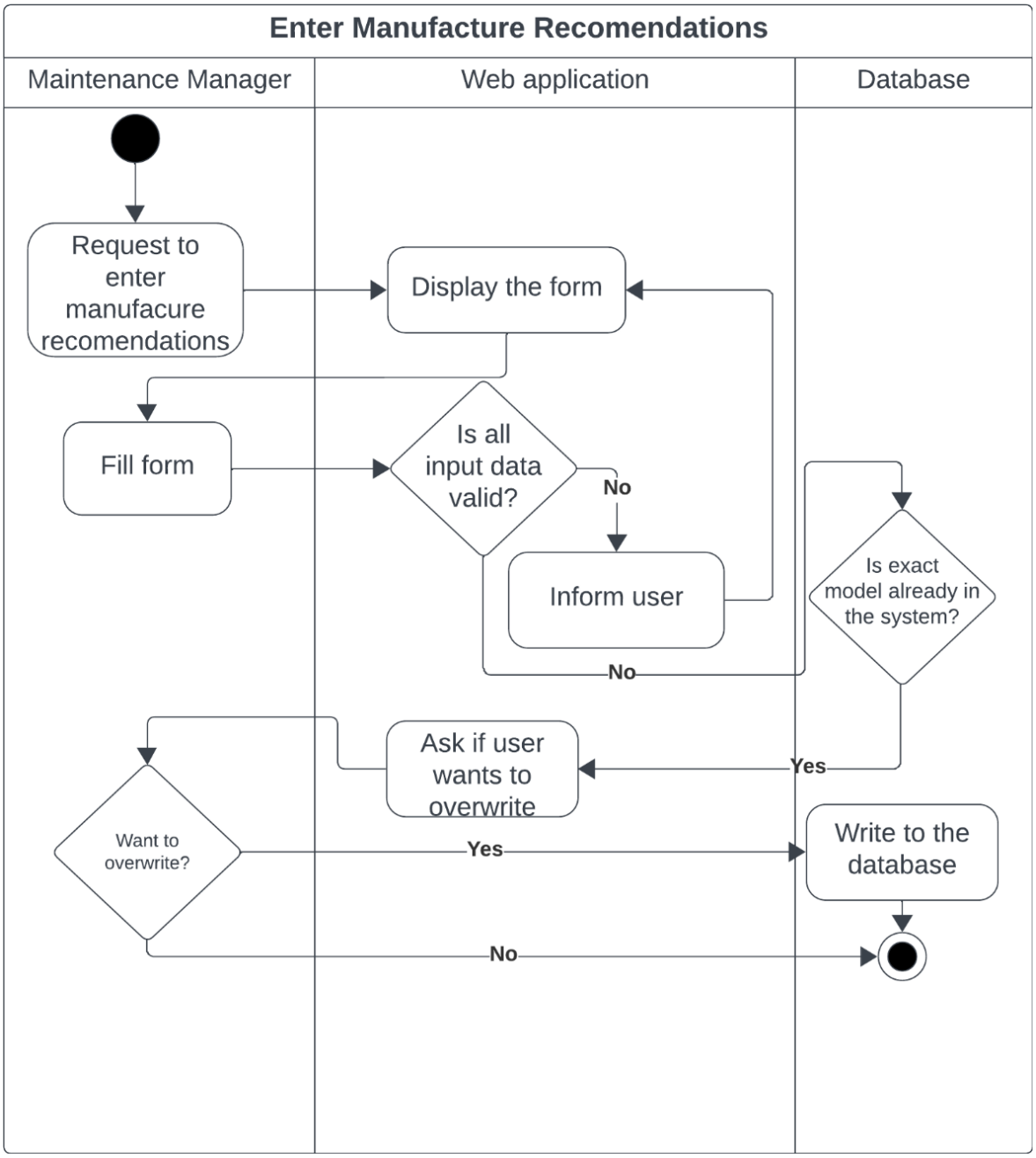

Figure 6.6 Activity Diagram for Entering manufacture recommendations of vehicle models.

6.1.2.2 See service booking details.

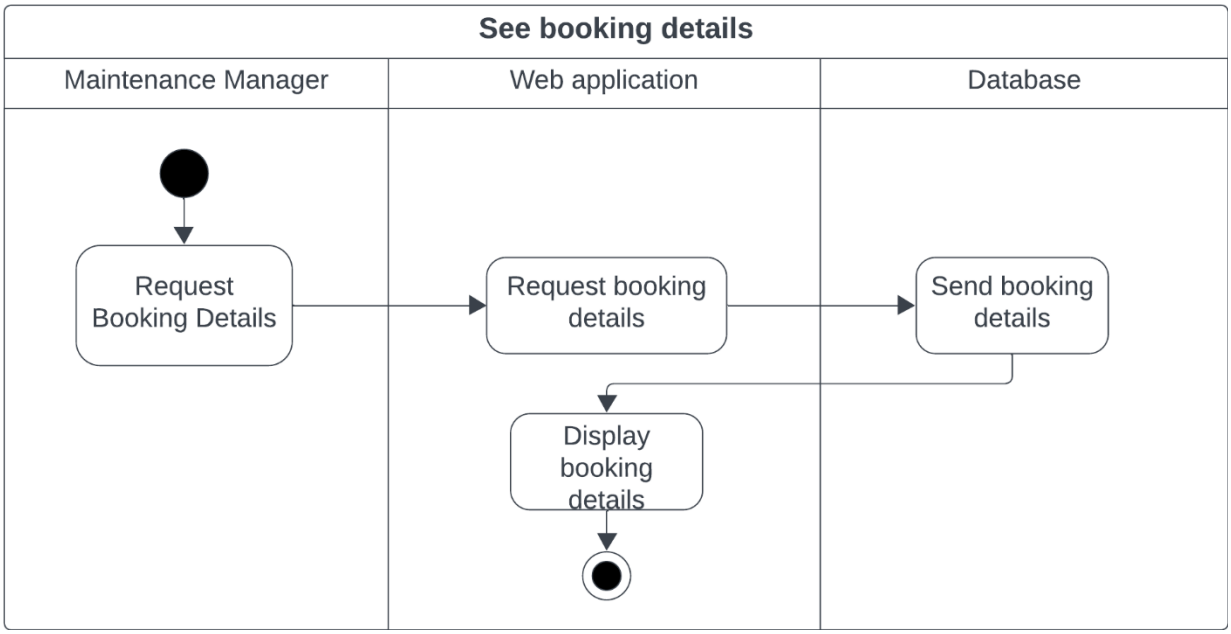

Figure 6.7 Activity Diagram for Requesting service booking details

6.1.2.3 Update the progress of booking.

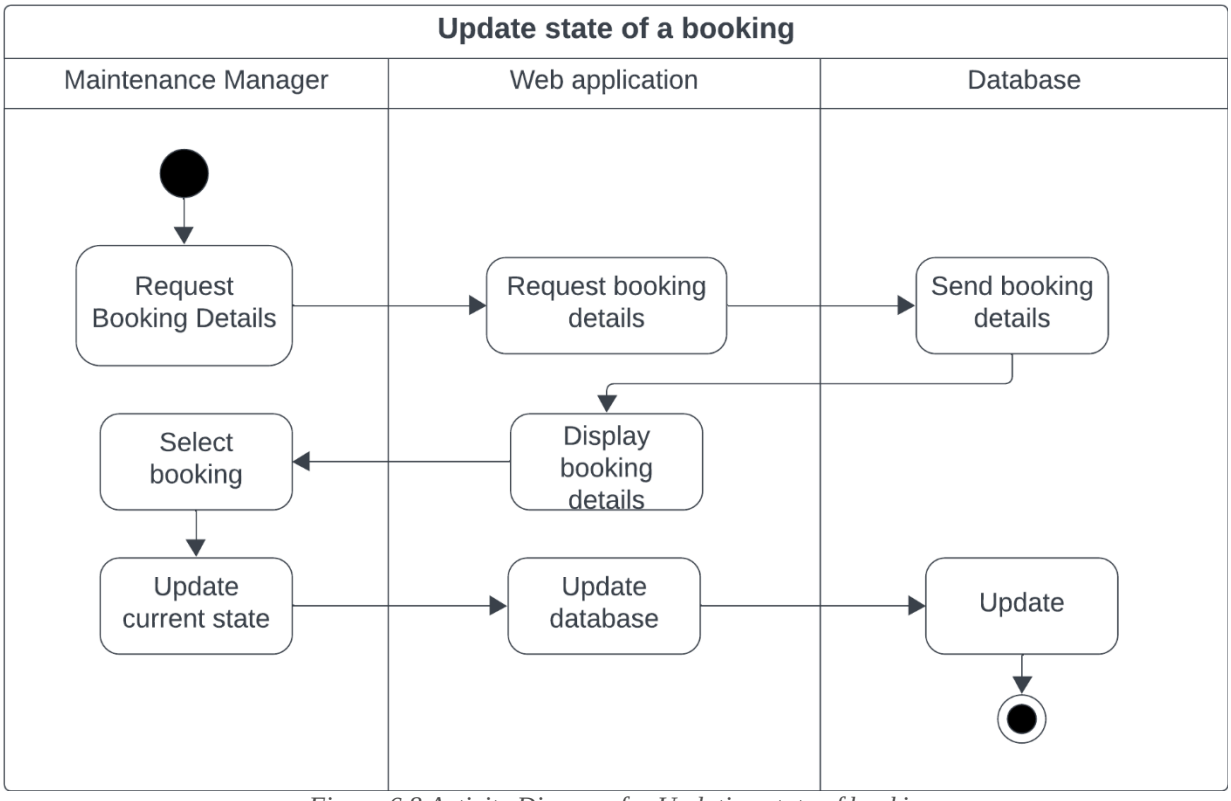

Figure 6.8 Activity Diagram for Updating state of bookings

6.1.2.4 Register customer on behalf of customer.

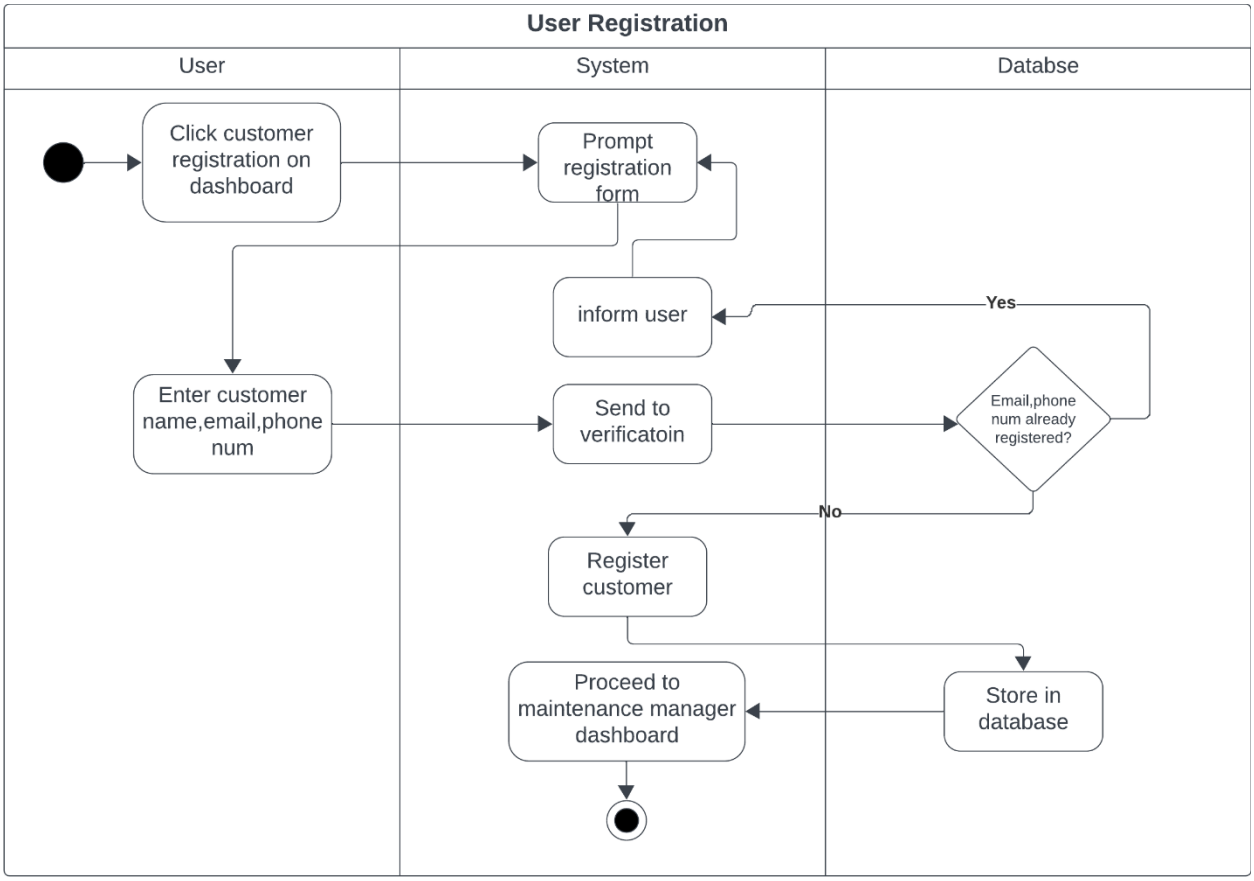

Figure 6.9 Activity Diagram for Registering a customer on behalf of customer

6.1.3 *Marketing team activity diagrams*

6.1.3.1 Send personalized offers to customers.

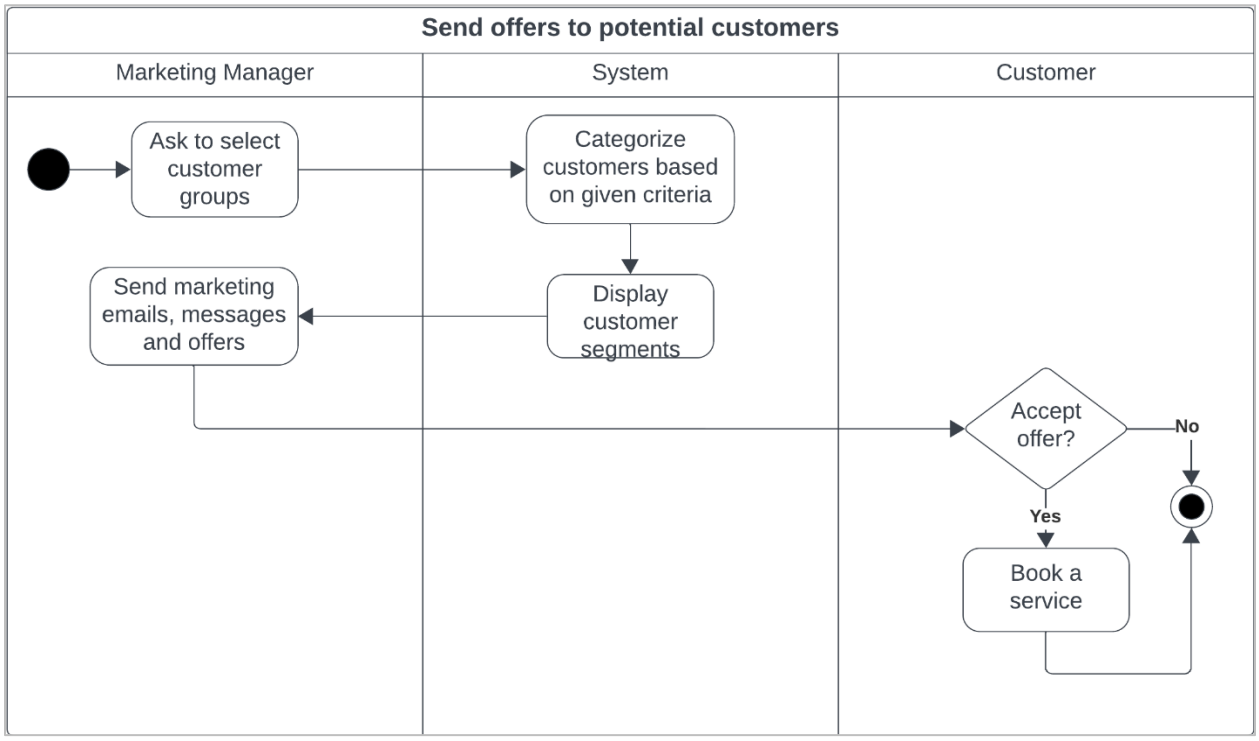

Figure 6.10 Activity Diagram for Sending offers.

6.1.4 *Admin activity diagrams*

6.1.4.1 Generate sales report.

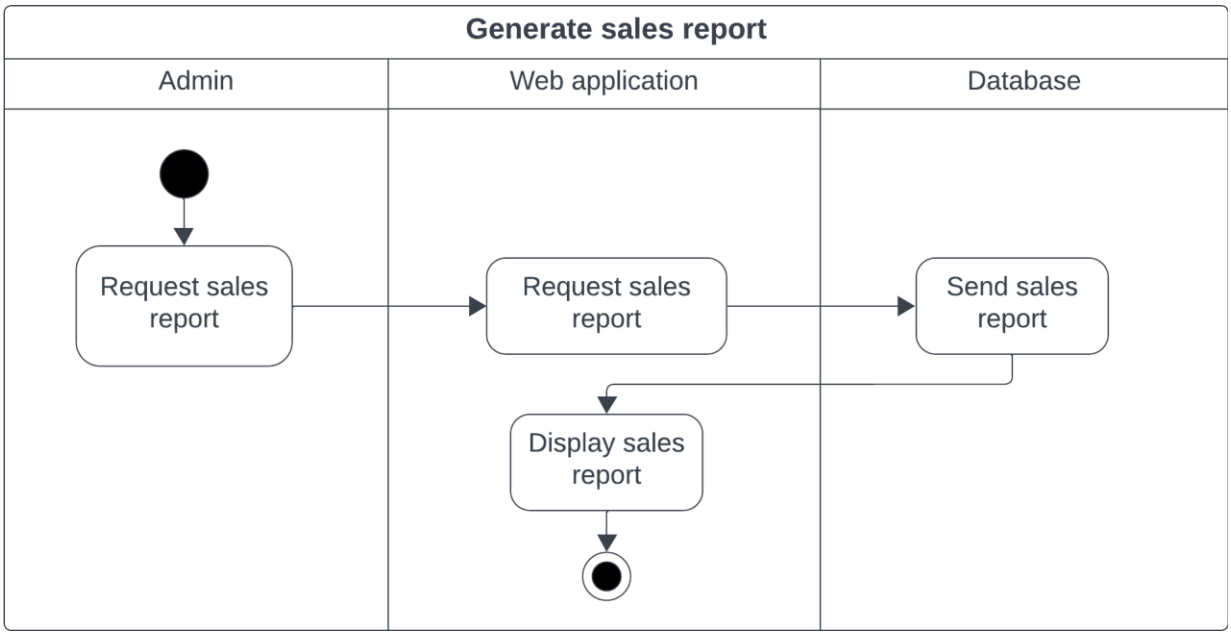

Figure 6.11 Activity Diagram for Generating sales reports

6.1.4.2 Generate statistics report.

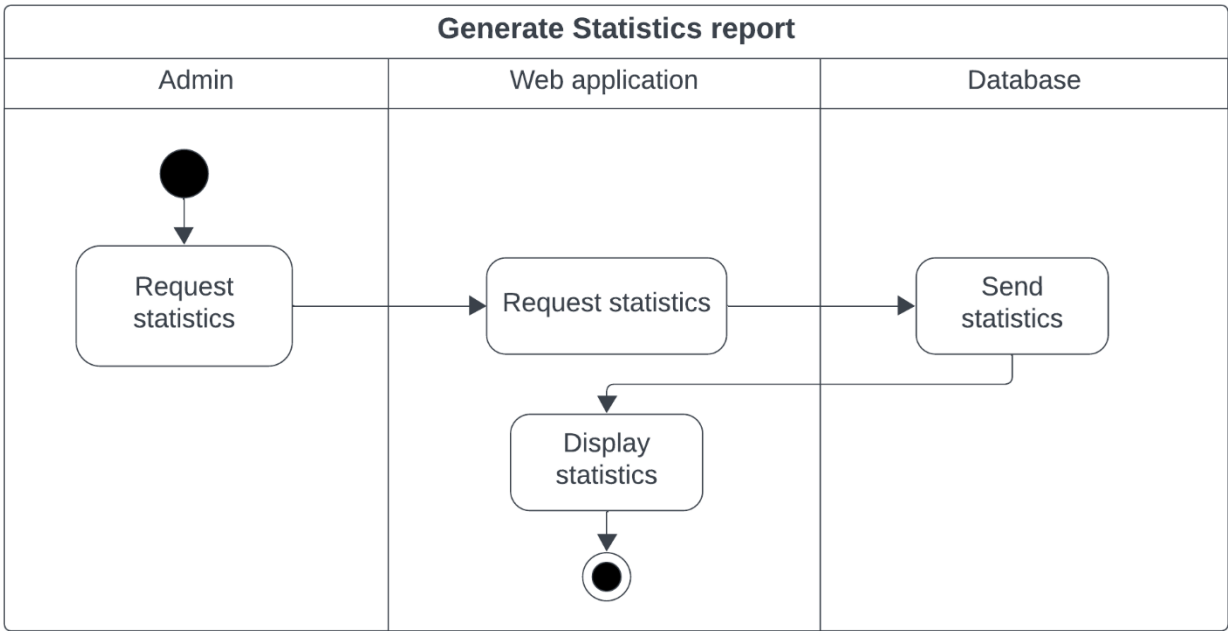

Figure 6.12 Activity Diagram for Generating statistics reports.

6.1.4.3 Register an employee.

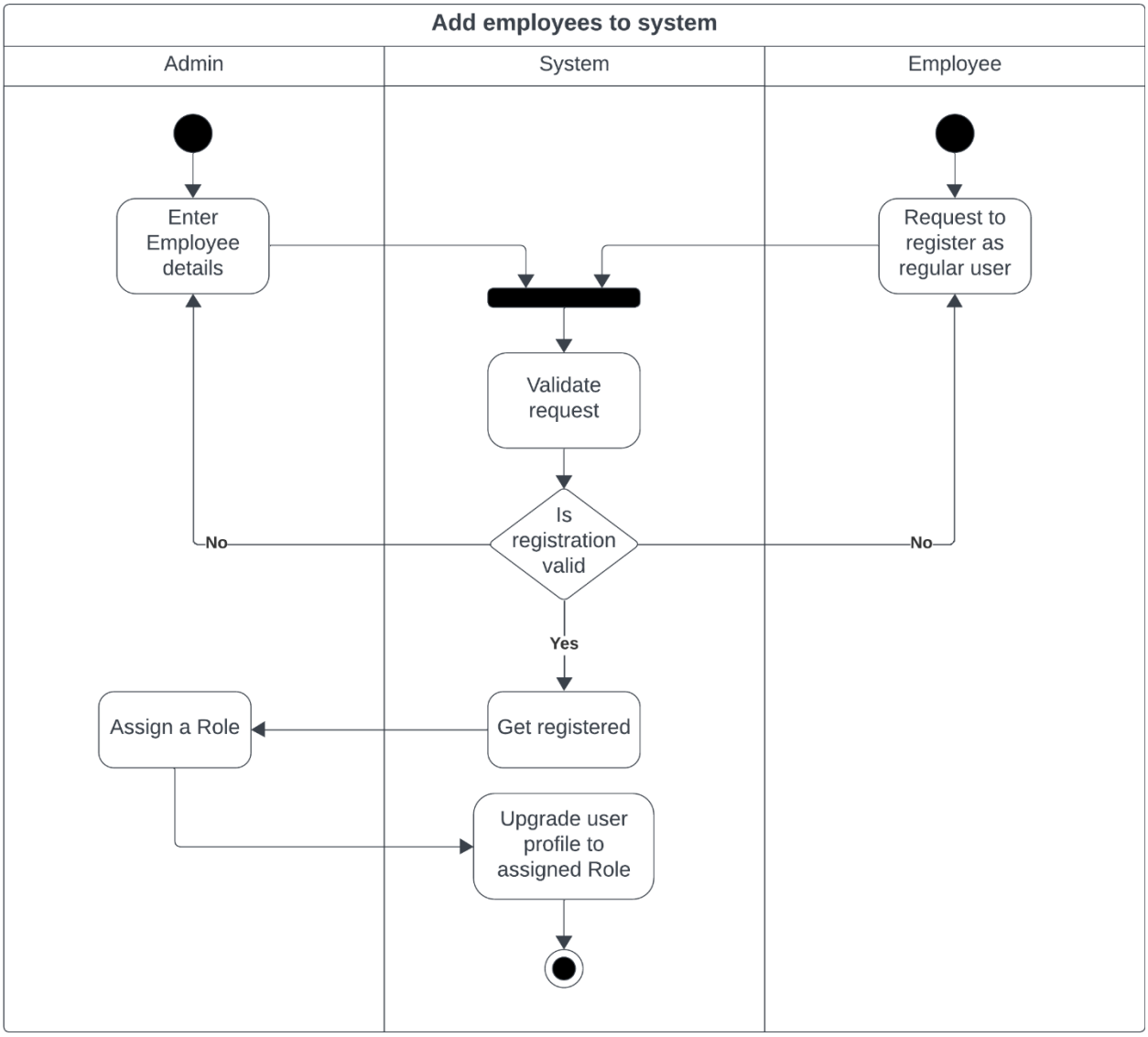

Figure 6.13 Activity Diagram for Adding employee to the system.

6.1.4.4 Access customer or user profiles to see their activity.

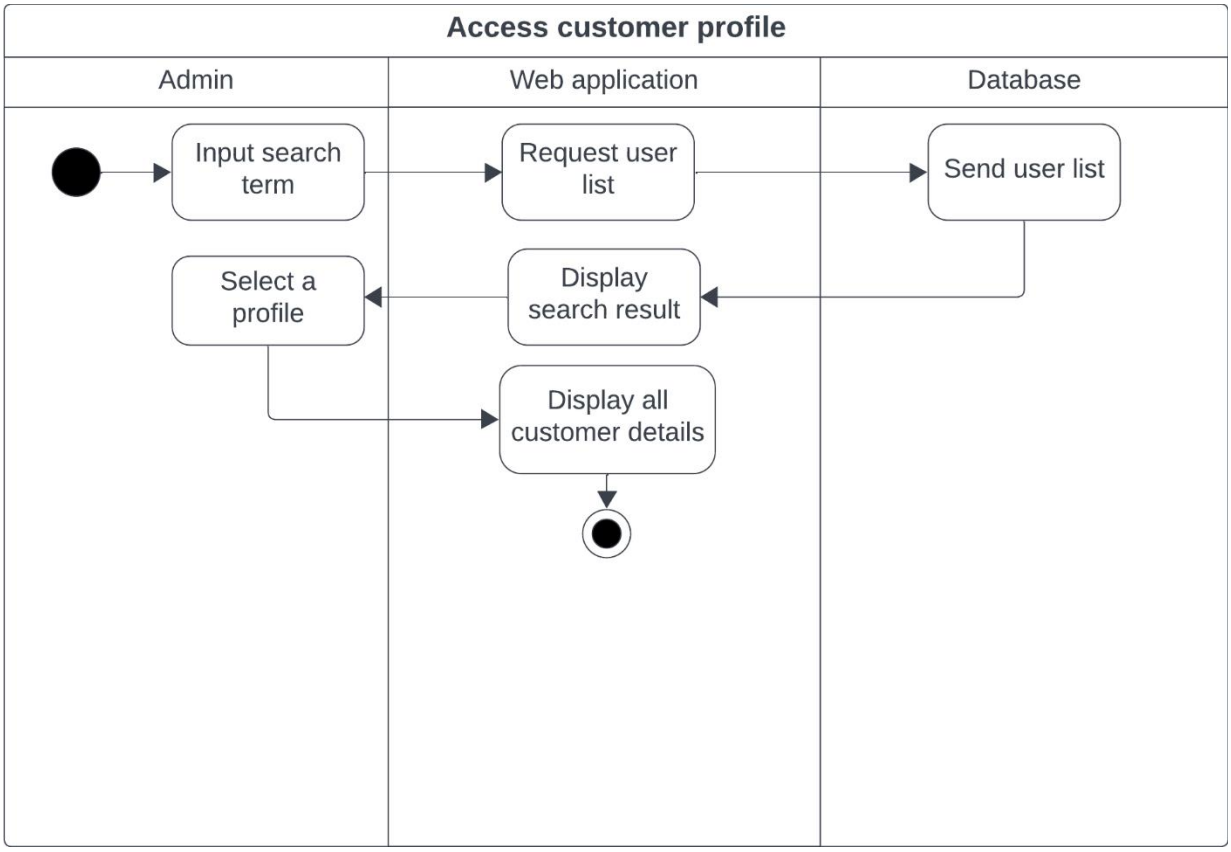

Figure 6.14 Activity Diagram for Accessing customer or user profiles.

6.2 Sequence diagrams

6.2.1 Register to the system

This sequence diagram illustrates how customers initiate the registration process either individually or through a service center. The User Interface collects personal and vehicle information, and the Backend validates and processes the data. The Database stores the registered details for future use. This interaction ensures a smooth and efficient registration experience within the CRM system.

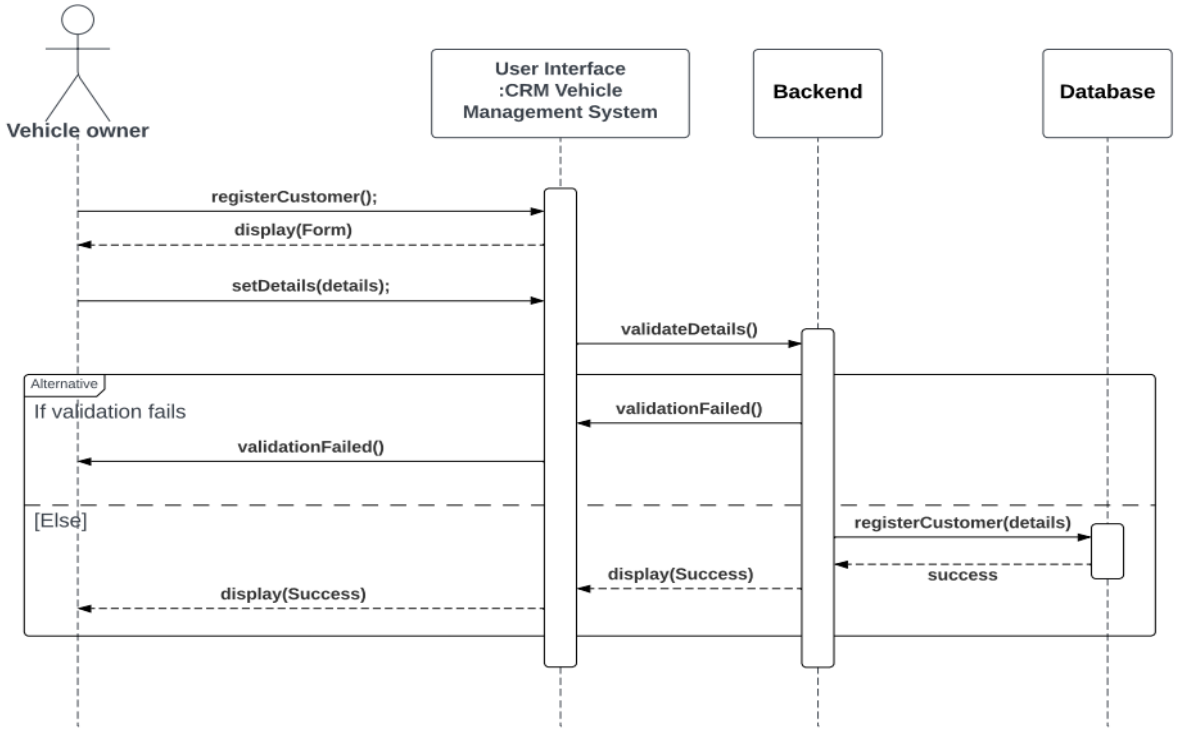

Figure 6.15 Sequence diagram for registering to the system

6.2.2 Register new vehicle

This sequence diagram showcases the process of registering a new vehicle within the CRM system, involving both the Vehicle Owner and the Maintenance Manager. The Vehicle Owner interacts with the User Interface to provide vehicle details, while the Maintenance Manager validates and approves the registration. The Backend processes the information, updates the Database, and generates a confirmation.

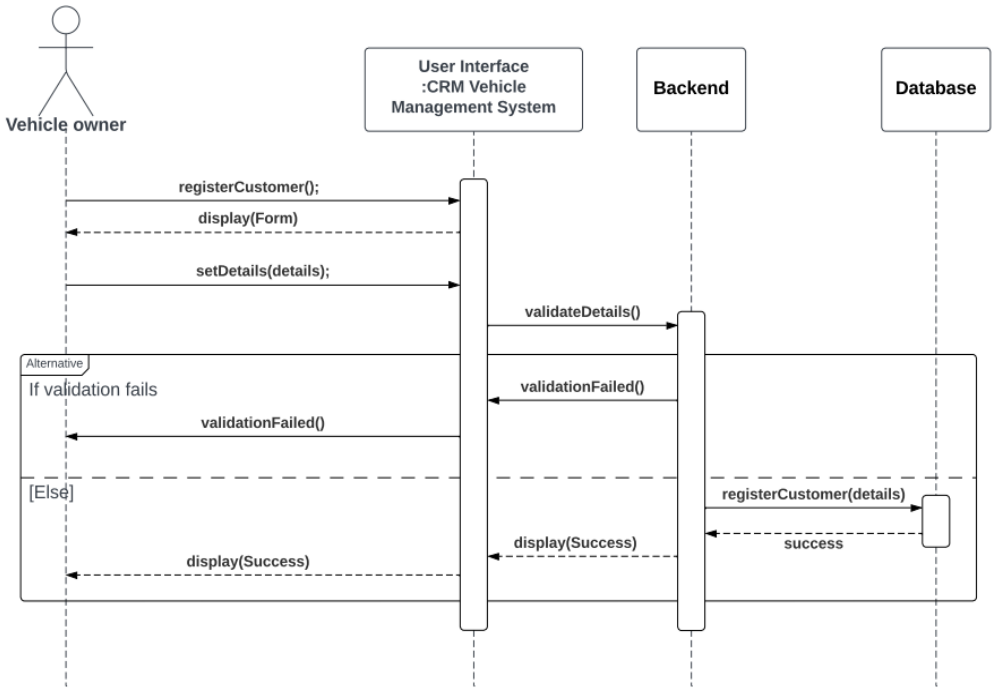

Figure 6.16.1 Sequence diagram for registering a new vehicle by the Vehicle Owner

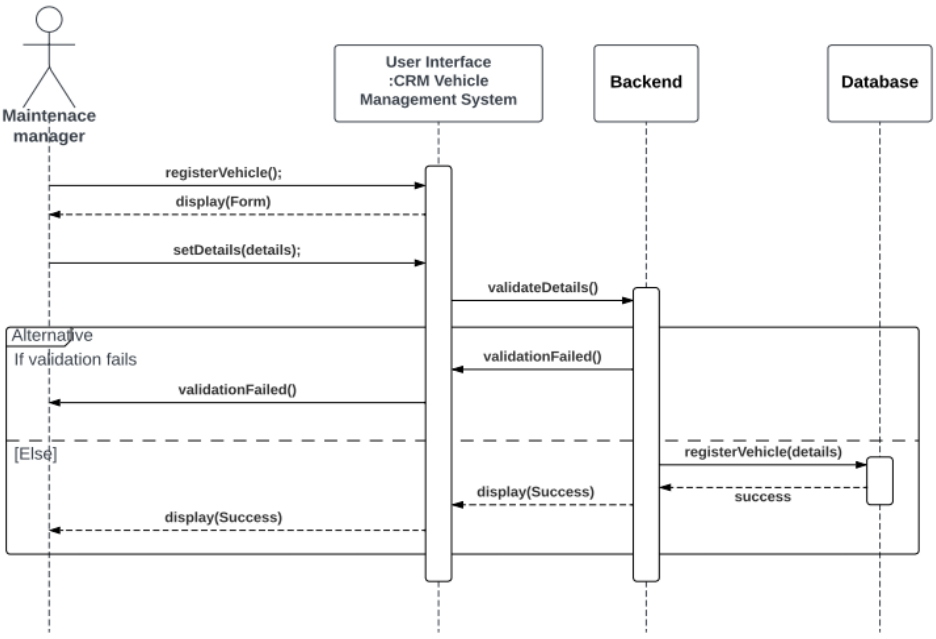

Figure 6.2.2.2 Sequence diagram for registering a new vehicle by the Maintenance Manager

6.2.3 *Register manufacturer recommendations*

This sequence diagram illustrates the procedure for the Maintenance Manager to record manufacturer recommendations for vehicle maintenance. The Maintenance Manager accesses the User Interface, inputs the relevant details, and submits the recommendations. The Backend validates and processes the information, updating the Database with the new manufacturer recommendations.

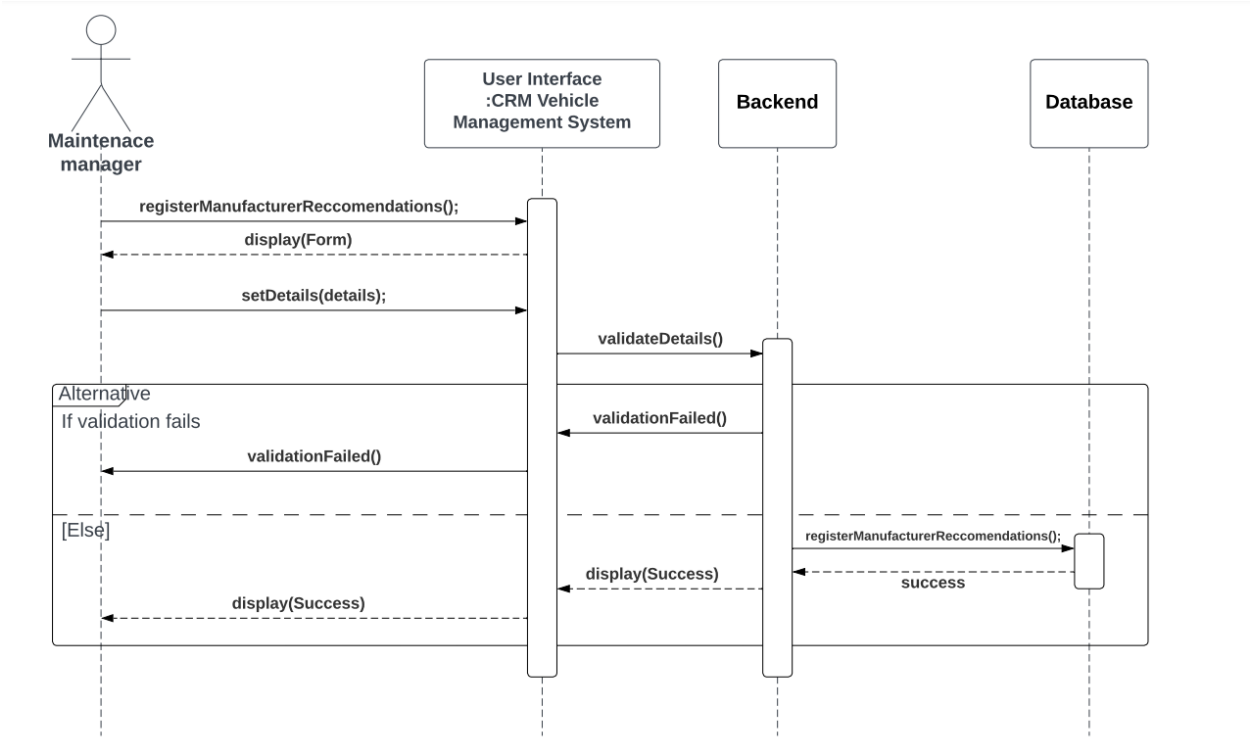

Figure 6.2.3 Sequence diagram for registering manufacturer recommendation details

6.2.4 *Book a service by the vehicle user*

This sequence diagram outlines the steps involved in a vehicle user booking a service. The user selects the desired service, triggering the system to fetch available timeslots from the database. The Backend checks the available timeslots and communicates this information to the User Interface, which then displays the options to the user. After the user selects a timeslot, the Backend schedules the booking, updates the Database with the booking details, and sends a success message back to the User Interface for display.

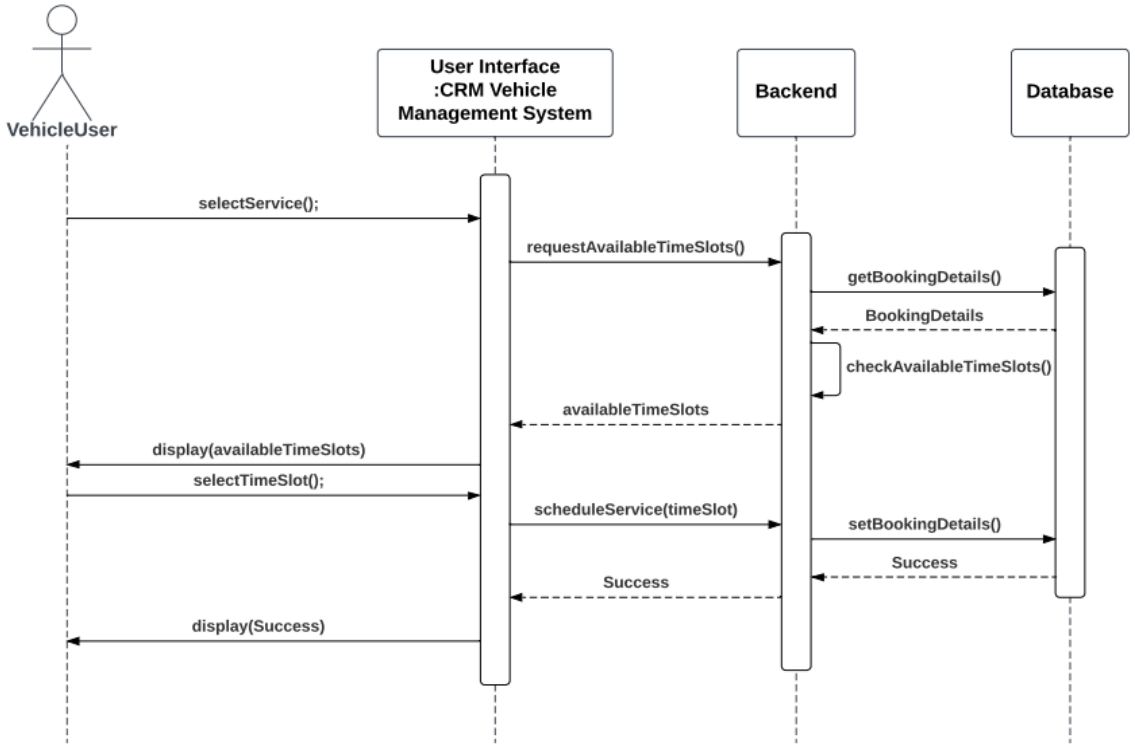

Figure 6.2.4 Sequence diagram for book a service by the Vehicle User

6.2.5 *Service booking by the maintenance manager.*

This sequence diagram illustrates the process of a maintenance manager booking a service for a customer. The maintenance manager initiates the process by checking the booking slots. The Backend retrieves the booking details from the Database and sends them to the User Interface for display. If the desired booking slot is not available, the maintenance manager can choose to reschedule the timeslot. In that case, the Backend updates the timeslot, and if the slot is available, the maintenance manager schedules the booking. The updated booking details are then stored in the Database

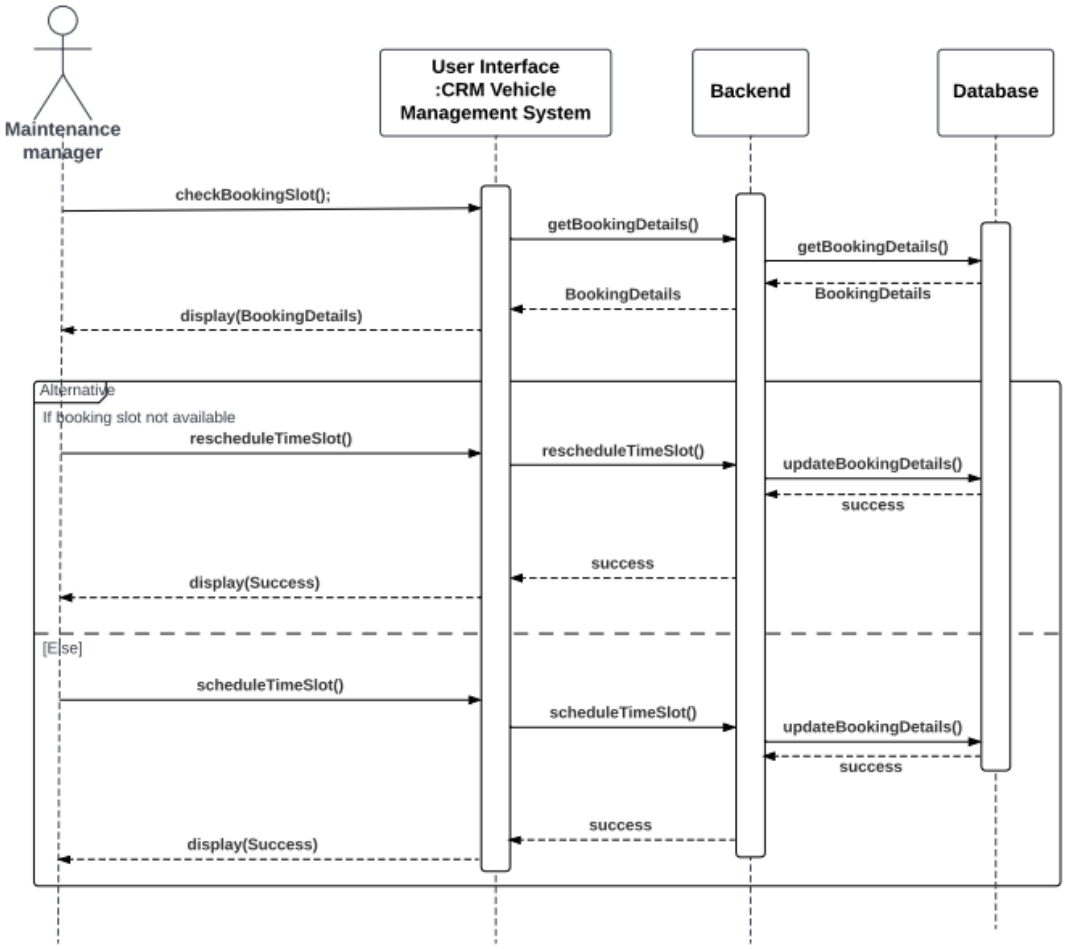

Figure 6.2.5 Sequence diagram for registering a new vehicle by the Maintenance manager.

6.2.6 *Send notifications to the vehicle users.*

This sequence diagram outlines the process of sending notifications to a vehicle user. The process begins with the submission of notification details, which are then validated by the system. The User Interface gathers customer details, and the Backend retrieves manufacturer recommendation details from the Database. Using this information, the system calculates the service interval. Subsequently, a notification is sent to the user. The details of the notification are saved in the Database, and a success message is displayed, confirming the successful notification process.

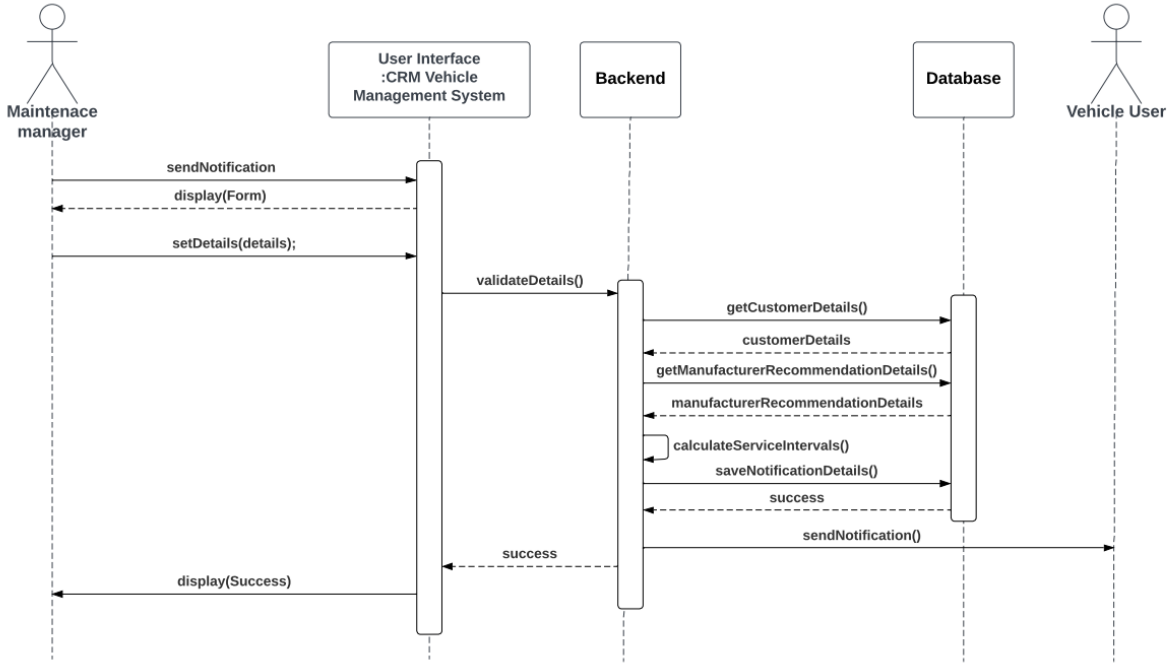

Figure 6.2.6 Sequence diagram for send notifications to the vehicle users

6.2.7 *Generate sales report.*

These sequence diagrams illustrate the procedure of generating a sales report by a sales manager and a marketing manager. The process commences with the sales manager's request to generate a report. The system fetches the relevant report data, followed by a request for the report type template and necessary parameters. Using this information, the system generates the report, and upon completion, displays the generated report to the sales manager.

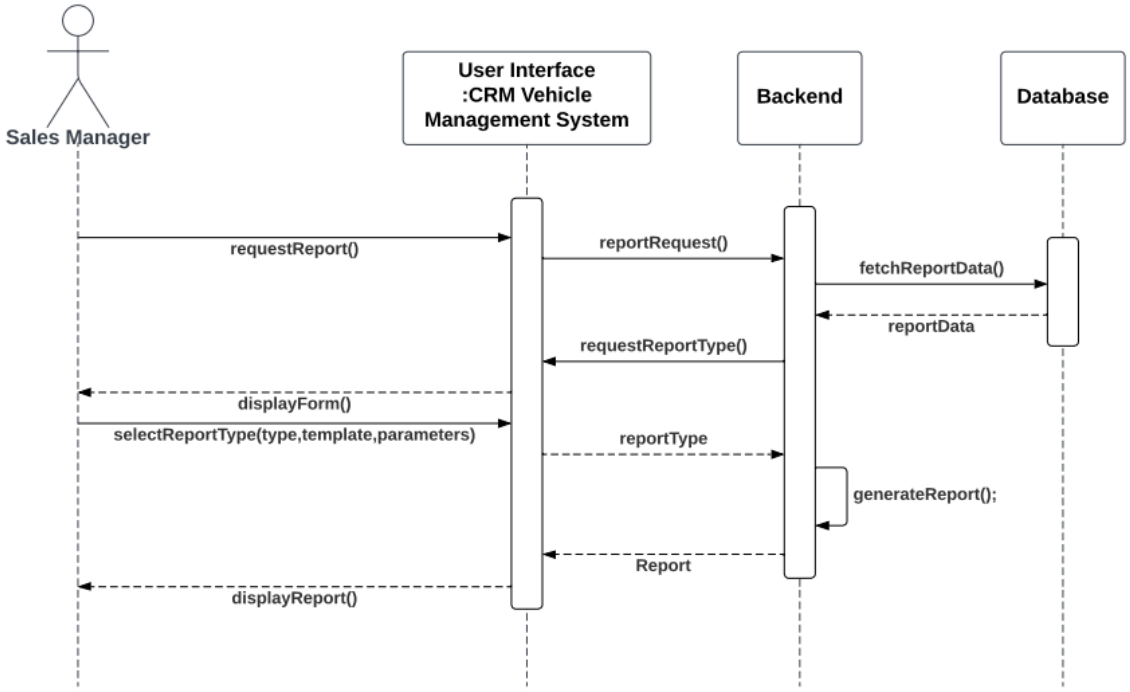

Figure 6.2.7.1 Sequence diagram for generating report for the sales manager

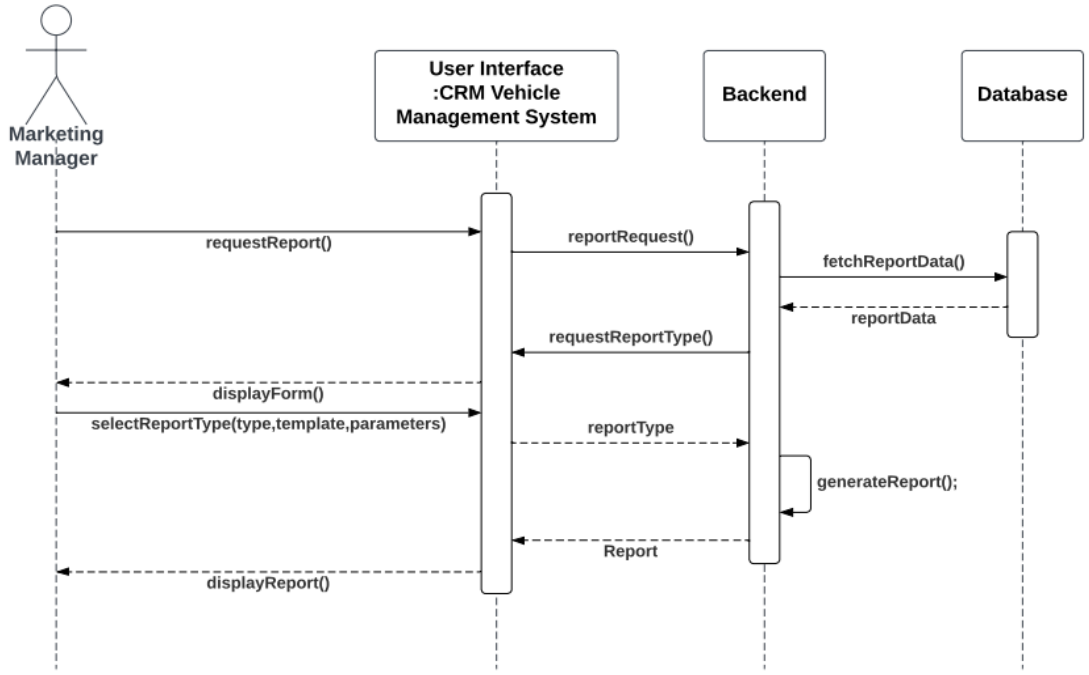

Figure 6.2.7.2 Sequence diagram for generating report for Marketing manager

6.2.8 *Send a message using internal message system*

This sequence diagram outlines the process of sending a message through the internal messaging system. The user initiates the action by composing a message and sending it. The system then displays the interface for composing the message. After composing, the user sends the message. The system, in turn, stores the message data in the database and confirms the successful message transmission.

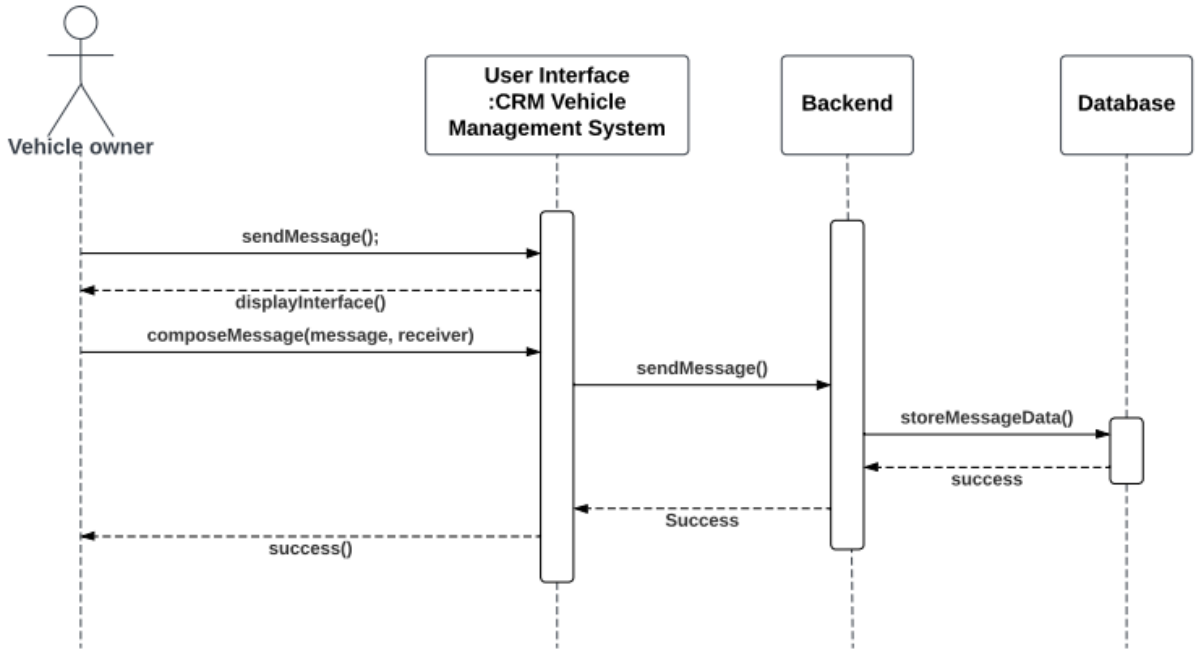

Figure 6.2.8 Sequence diagram for sending a message using the internal message system

6.2.9 *View messages on the internal messaging system*

This sequence diagram illustrates the process of viewing message history within the internal messaging system. The user initiates the action by requesting to view their message history. The system then retrieves message data from the database and displays it to the user.

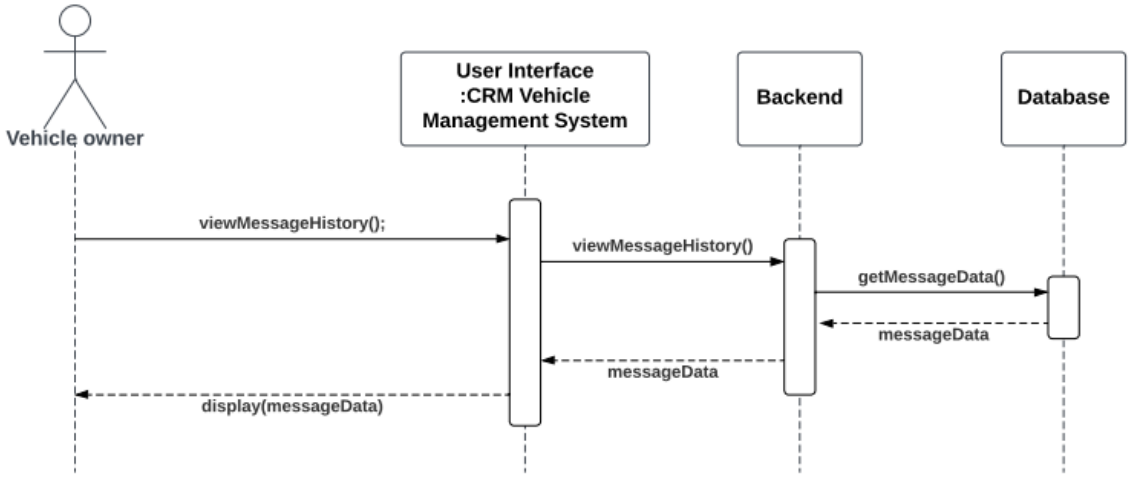

Figure 6.2.9 Sequence diagram for viewing messages.

6.2.10 *Create new forum discussion thread.*

This sequence diagram outlines the process of creating a new forum discussion thread within the CRM system. The user initiates the action by providing the necessary details for the new discussion thread. The system then creates the discussion thread, storing it in the database. Upon successful creation, a confirmation message is displayed to the user.

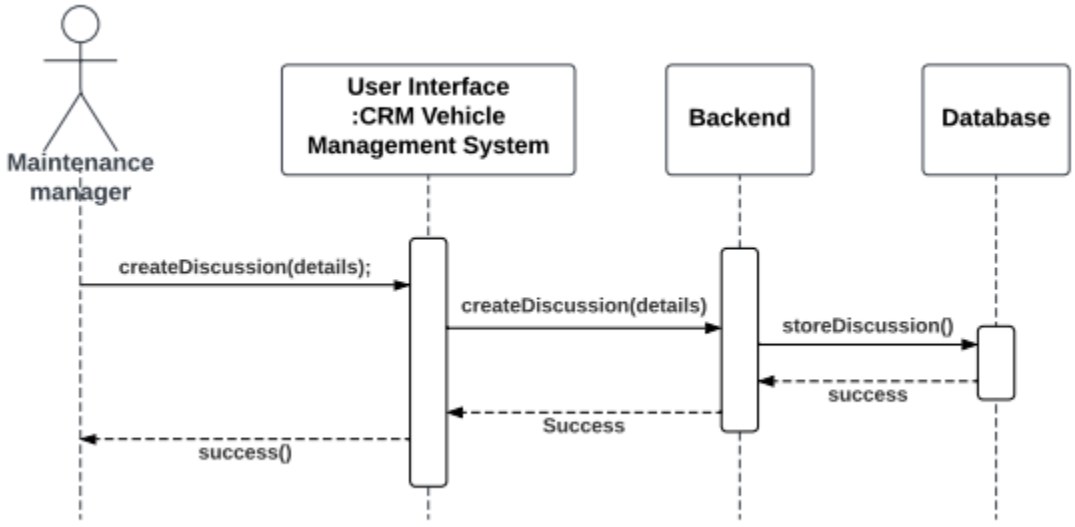

Figure 6.2.10 Sequence diagram for creating a new forum discussion

6.2.11 *Reply to a currently existing discussion thread*

This sequence diagram illustrates the process of replying to an existing forum discussion thread in the CRM system. The user selects the relevant thread, composes the reply, and submits it. The system validates the reply and stores it in the database, associating it with the corresponding discussion thread. After successful storage, a confirmation message is displayed to the user.

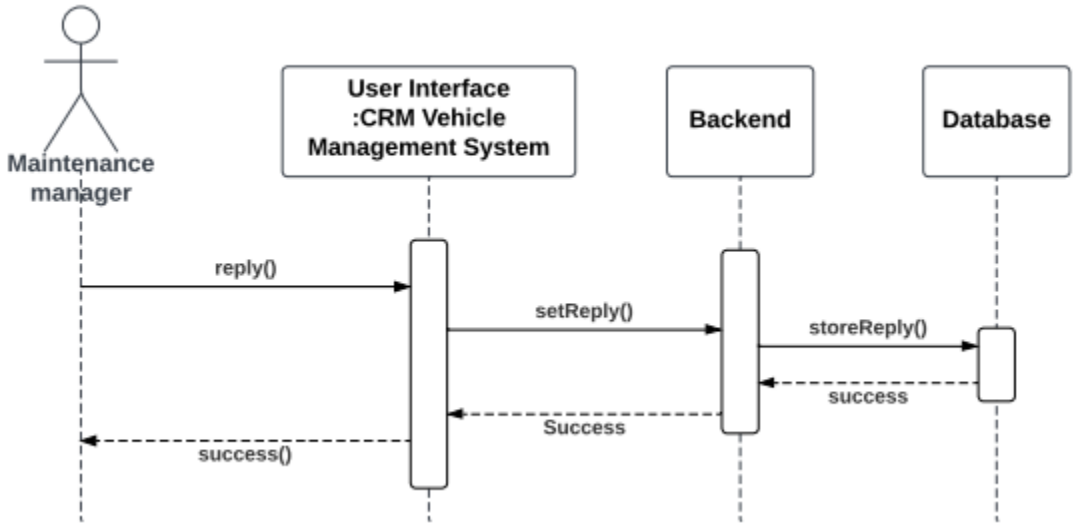

Figure 6.2.11 Sequence diagram for reply to currently existing discussion thread

7. Deployment View

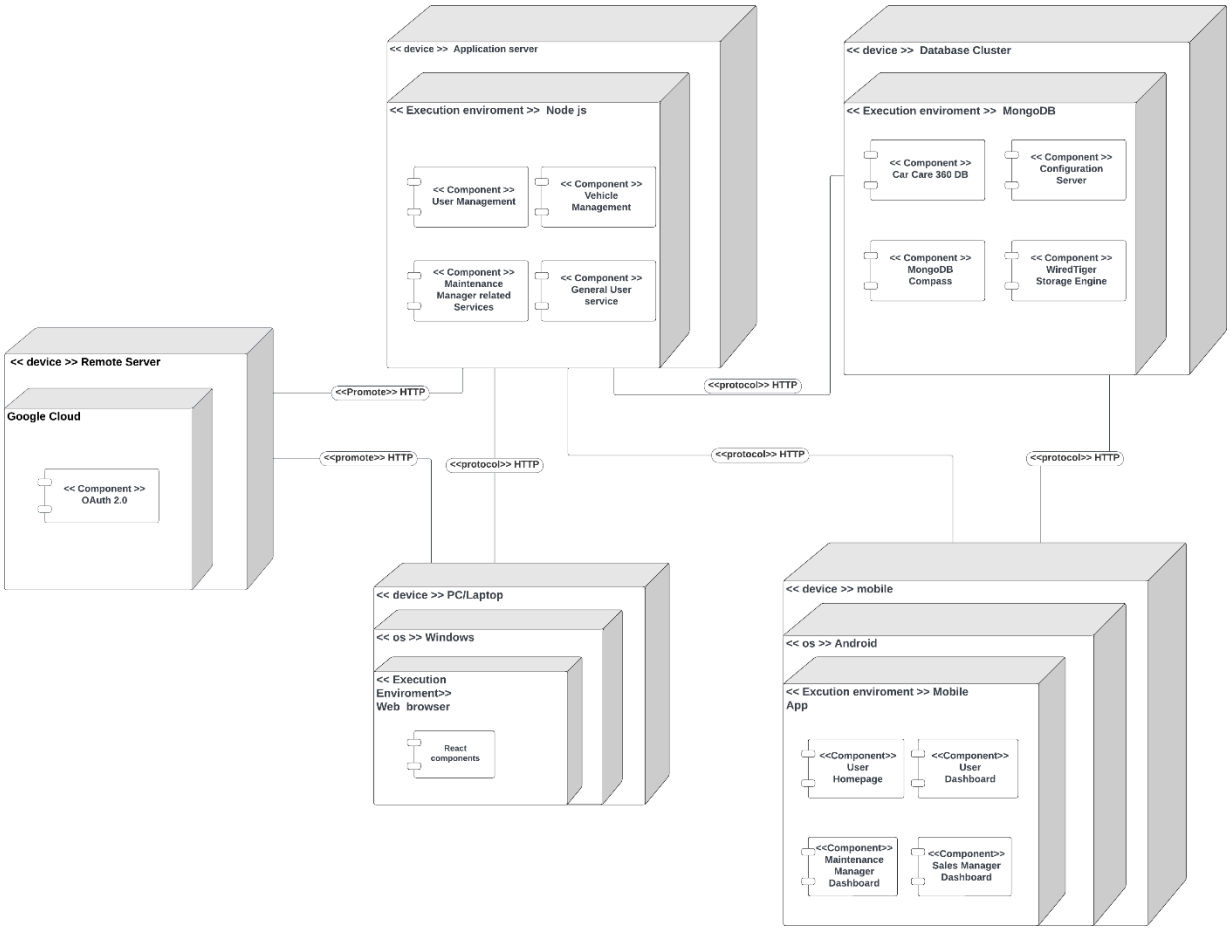

Figure 7.1 Deployment View

The deployment diagram establishes a connection between the software architecture formulated during design and the physical system architecture responsible for its execution. when it comes to the distributed systems, it patterns the distribution of the software across the physical nodes. The software systems are embodied using varies of artifacts, thereinafter they are mapped to the execution environment where the software such as nodes are executed. There are various nodes are included in the deployment diagram. Accordingly, the proximity between them is characterized using communication.

The deployment diagram for the proposed system which has been indicated in the above figure, The fundamental elements of this system encompass the web client, mobile client, application server, remote server, and database server. The web browser where the React components are running will be on the web client's PC. MongoDB databases will be performed in the database server. Moreover, these subsystems will communicate through different protocols such as HTTP and TCP/IP.

8. Implementation View

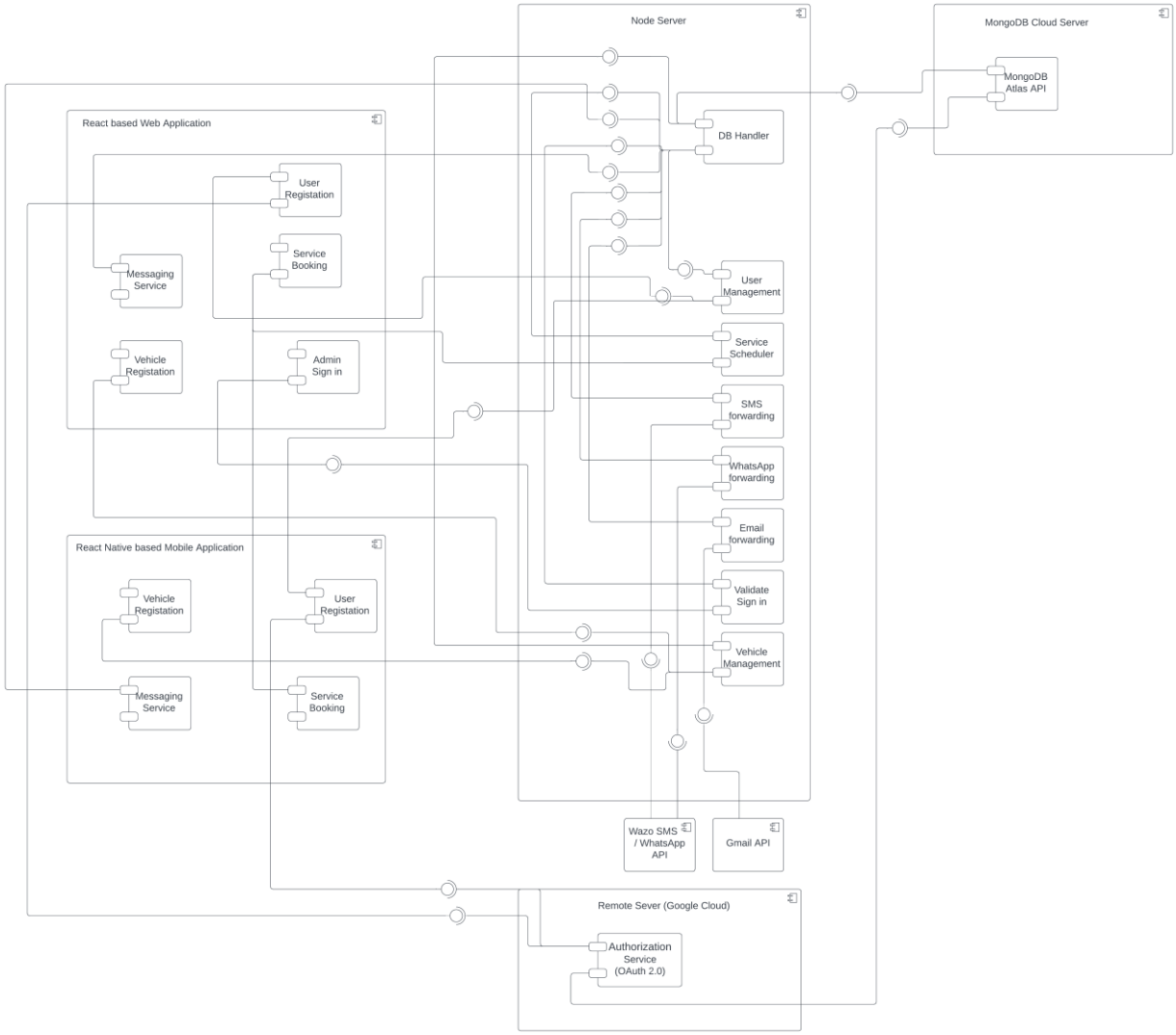

Figure 8.1 Implementation View

8.1 Overview

The architecture consists of a React-based web application responsible for features like messaging service, vehicle registration, user registration, service booking, and admin sign-in. A React Native mobile application mirrors these functionalities, including vehicle registration, messaging, user registration, and service booking. A Node server handles database interactions, user management, service scheduling, and communication through SMS, WhatsApp, and email. Wazo SMS/WhatsApp API, GmailAPI, and remote authorization service support these functions. Data is stored in a MongoDB cloud server via its Atlas API.

This modular structure enables seamless interaction and functionality across both web and mobile platforms.

8.2 Layers

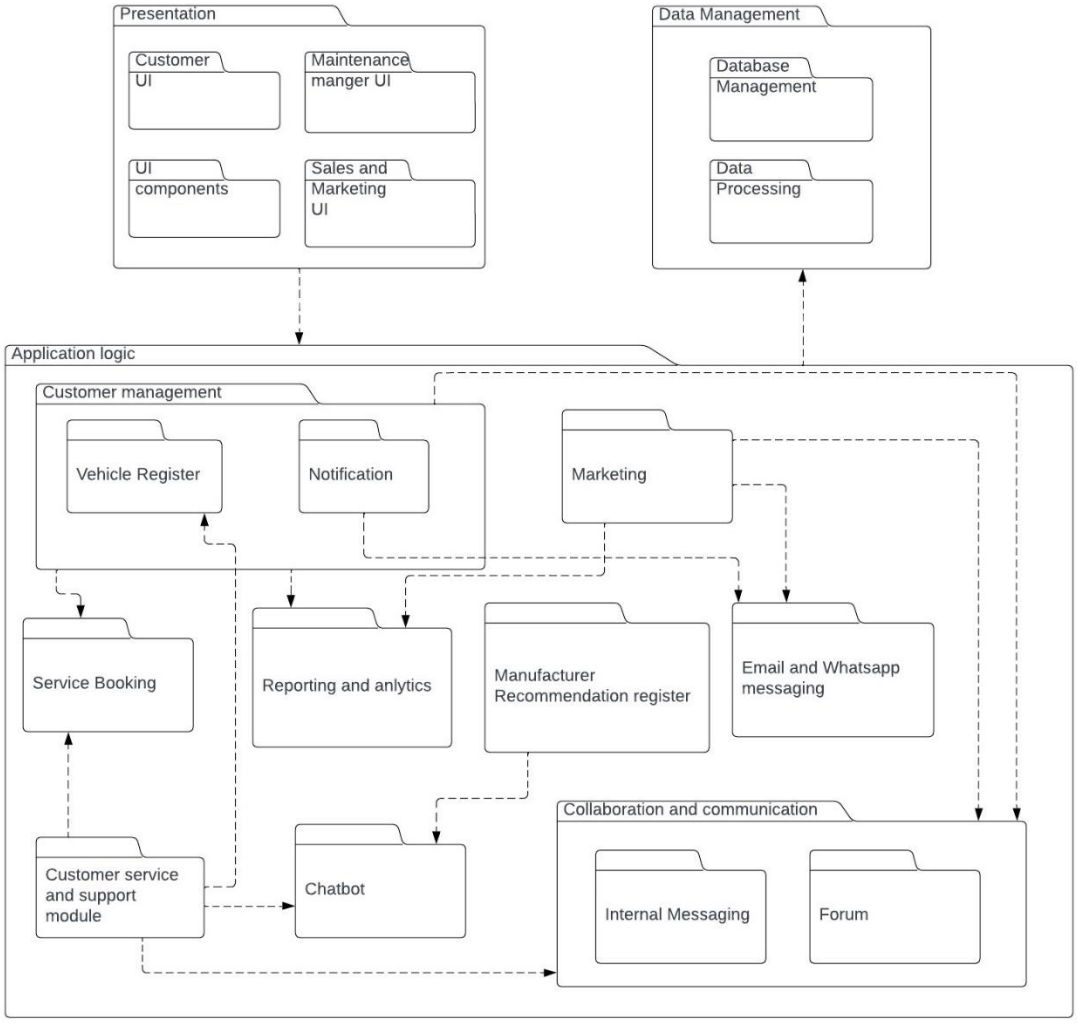

Figure 8.2. Package Diagram

The package diagram provides a compact representation of the CRM for Vehicle Maintenance system's architecture. It highlights three essential layers: Presentation, Data Management, and Application Logic. The Presentation layer encompasses user interfaces and UI components, while the Data Management layer encapsulates database management and data processing. The Application Logic layer hosts multiple modules and sub-modules such as Customer Management, Manufacturer Recommendation Register, Notification, Marketing, Customer Service and Support, Chatbot, Email and WhatsApp messaging, Service booking. Reporting and analytics and Collaboration and Communication. These modules interact through arrows, fostering seamless communication and functionality.

9. Data View (optional)

10. Size and Performance

10.1 System Size

The CRM System is designed to accommodate various types of users including regular customers, guest customers, maintenance managers, marketing people, sales managers, business owners and system administrators. The size of the system is heavily dependent on the expected number of users, vehicles, service bookings, and interactions within the system. The system is expected to serve a considerable user base for a vehicle maintenance Centre, with thousands of customers and several employees across different roles. The system is expected to be developed with the capability of scaling as business grows. The system will store information about registered vehicles, service history, manufacturer recommendations, booking details and many more. The volume of vehicle and service data will impact the size of the database and performance. Non-relational MongoDB database used because of powerful querying capabilities. Since MongoDB free plan is proposed to be used, database size is limited to 512 MB. But it is possible to get more space and performance by upgrading to paid plan.

10.2 System performance

The architecture of the CRM System is designed to maximize performance of the application. Several architectural decisions were taken to address the performance requirements of the system. The system implements client-side caching of frequently accessed data to reduce repetitive server requests while increasing responsiveness.

Database performance is crucial for system performance. The architecture integrates a non-relational database system, MongoDB, because of its scalability and powerful querying capabilities. Data is structured and indexed to support quick retrieval.

Content delivery networks (CDNs) are utilized to deliver frequently accessed assets closer to users, minimizing data transmission times. This helps application to perform fast with increased reliability.

In summary, the performance considerations within the architecture of the CRM System address various aspects of performance characteristics. The system architecture ensures that the system delivers a highly responsive user experience for all users at the end.

11. Quality

The software architecture of the CRM for Vehicle Maintenance System is designed with a focus on various quality attributes beyond mere functionality. The architecture embodies principles that contribute to key aspects such as extensibility, reliability, and portability. Additionally, considerations related to safety, security, and privacy have been integrated into the architectural decisions.

- **Extensibility**:

  The modular and layered architecture of the CRM system facilitates extensibility. New features, modules, or functionalities can be added with minimal impact on existing components. This architectural design enables the system to evolve and adapt to changing requirements over time.

- **Reliability**

  The division of the system into distinct layers helps in isolating concerns and reducing interdependencies. This separation promotes system stability and fault tolerance.

- **Usability**

The user-centric design of the architecture ensures a high degree of usability. User interfaces are designed with intuitiveness and user-friendliness in mind. Clear navigation paths, consistent layouts contribute to an enhanced user experience. The system minimizes the learning curve for users of varying technical expertise through its straightforward interfaces and well-organized workflows.

- **Security and Privacy measures:**

  Security and privacy considerations are embedded in the architecture. User authentication and authorization mechanisms are implemented using trusted APIs. With a role-based architecture, access to different system modules is granted based on user roles, preventing unauthorized access, and maintaining the confidentiality of customer and business data.