

COMSATS University Islamabad (CUI)

**Software Design Description
(SDS DOCUMENT)**

for

Motors Bay

Version 1.0

By

Muhammad Sarim CIIT/FA21-BSE-059/ISB

Mujeeb Munsab CIIT/FA21-BSE-070/ISB

Supervisor

Dr. Nusrat Shaheen

Bachelor of Science in Software Engineering (2021-2025)

Table of Contents

1. Introduction.....	1
1.1 Scope.....	1
1.2 Modules.....	1
1.2.1 Module 1: User Management	1
1.2.2 Module 2: Search and Buy.....	1
1.2.3 Module 3: Post Your Ride or Auto Part.....	2
1.2.4 Module 4: 3D Model Visualization and Comparison	2
1.2.5 Module 5: Chat and Notification	2
1.2.6 Module 6: Review and Ratings	2
1.2.7 Module 7: Vehicle Verification.....	3
1.2.8 Module 8: Events and News	3
1.2.9 Module 9: Forums and Community.....	3
1.2.10 Module 10: Recommendation System	3
1.2.11 Module 11: Maintenance and Tracking	3
1.2.12 Module 12: AI Chatbot	4
1.2.13 Module 13: Auto Services Marketplace.....	4
2. Design Methodology and Software Process Model	4
2.1 Design Methodology:	4
2.1.1 Rationale behind this approach:	4
2.2 Process Methodology	5
2.2.1 Rationale behind this approach:	5
3. System Overview	6
3.1 Architectural Design.....	7
3.1.1 Box and Line Diagram.....	7
3.1.2 Architectural Design.....	8
4. Design Models.....	9
4.1 Activity Diagrams.....	9
4.1.1 Register User	9
4.1.2 Login.....	10
4.1.3 User Profile Management	11
4.1.4 Admin Dashboard Management.....	12
4.1.5 Dealership Registration.....	13
4.1.6 Dealership Profile Management	14

4.1.7	Search	15
4.1.8	Filter and Sort Search Results	16
4.1.9	Add to Watchlist.....	16
4.1.10	Post Ride or Auto part	17
4.1.11	Manage Dealership Inventory	18
4.1.12	Upload Media.....	18
4.1.13	Advert Status.....	19
4.1.14	3D Model of a Vehicle.....	20
4.1.15	Customize Vehicle in 3D	21
4.1.16	Compare Vehicles	22
4.1.17	Real-Time Private Messaging	23
4.1.18	Receive Push Notifications	24
4.1.19	Customize Notification Preferences	25
4.1.20	Submit Review	25
4.1.21	Filter reviews.....	26
4.1.22	Report Reviews	27
4.1.23	Vehicle Verification	28
4.1.24	Request Vehicle Inspection.....	29
4.1.25	Verify Driver's License	30
4.1.26	Read Car News and Articles	31
4.1.27	View Fuel Prices	32
4.1.28	Publish Blogs	33
4.1.29	Post Questions and Answers in Forums.....	34
4.1.30	Create Groups.....	35
4.1.31	Create Club	36
4.1.32	Join Clubs or Groups	37
4.1.33	Share Content	38
4.1.34	View Personalized Recommendations.....	38
4.1.35	Track Vehicle HealthVia OBD-II	39
4.1.36	Expense Tracking	40
4.1.37	Generate Reports	41
4.1.38	Service Reminders.....	42
4.1.39	AI Chatbot Interaction	43
4.1.40	Live Support.....	44
4.1.41	Auto Services	45
4.1.42	Book Auto Services	46

4.2	Class Diagram	47
4.3	Sequence Diagrams	48
4.3.1	User Registration Process	48
4.3.2	User Login Authentication.....	49
4.3.3	Password Recovery Flow.....	50
4.3.4	Search Vehicle Listings	51
4.3.5	Filter and Sort Results	51
4.3.6	View Vehicle Details.....	52
4.3.7	Create Vehicle Listing	52
4.3.8	Upload Images and Videos	53
4.3.9	Save Listing Details.....	54
4.3.10	Load 3D Model Viewer	54
4.3.11	Apply Customization Options	55
4.3.12	User Initiates Chat with Seller	56
4.3.13	Send and Receive Notifications	57
4.3.14	Escalate to Live Support	58
4.3.15	Submit a Review	59
4.3.16	View Ratings for a Vehicle/Seller	59
4.3.17	Submit Verification Request	60
4.3.18	Process Verification by Admin	61
4.3.19	View List of Users.....	62
4.3.20	Edit User Account Details.....	62
4.3.21	Delete User Account.....	63
4.3.22	Set Service Reminders	63
4.3.23	View Maintenance History	64
4.3.24	Receive Notifications for Service Due	64
4.3.25	Fetch Personalized Recommendations	65
4.3.26	List Available Auto Services	65
4.3.27	Manage Bookings.....	66
4.3.28	Generate Monthly/Yearly Reports.....	67
4.3.29	Export Reports in Desired Format.....	67
4.3.30	Post Questions or Answers	68
4.3.31	Interact with Other Users in Discussions	69
4.3.32	Create Clubs	70
4.3.33	Create Groups.....	70
4.3.34	Add Expense	71

4.3.35	View Expense Details	71
4.3.36	Edit or Delete Expense	72
4.3.37	Connect OBD-II Device	73
4.3.38	Retrieve Vehicle Health Data.....	73
4.3.39	Display and Log Health Metrics	74
4.4	State Transition Diagrams	75
4.4.1	User Account.....	75
4.4.2	Vehicle Listing	76
4.4.3	Chat Session.....	77
4.4.4	Service Reminder	78
4.4.5	Vehicle Health.....	79
4.4.6	Content Management.....	80
4.4.7	Forum Post	81
4.4.8	Recommendation System.....	82
4.4.9	Expense Tracking	82
4.4.10	Report	83
4.4.11	Notification	84
4.4.12	Verification	85
5.	Data Design.....	86
5.1	Data Dictionary	86
5.1.1	User	86
5.1.2	Vehicle.....	86
5.1.3	Booking.....	87
5.1.4	Post	87
5.1.5	Notification	87
5.1.6	Verification	87
5.1.7	Expense.....	88
5.1.8	Review	88
5.1.9	Report	88
5.1.10	Model Visualization	88
5.1.11	Event.....	89
5.1.12	News	89
5.1.13	Forum.....	89
5.1.14	Group/Club	89
5.1.15	Recommendation.....	90
5.1.16	Service	90

5.1.17 ServiceProvider	90
5.1.18 Maintenance	90
5.1.19 Chat	91
6. Implementation	92
7.1 Algorithm.....	92
7.2 External APIs/SDKs	95
7.3 User Interface	96
7. Testing and Evaluation	100
7.1 Unit Testing.....	100
7.2 Functional Testing	102
7.3 Business Rules Testing	104
7.3.1 Business Rules Testing	104
7.4 Integration Testing.....	107

Revision History

Name	Date	Reason for changes	Version

Application Evaluation History

Comments (by committee)	Action Taken
*include the ones given at scope time both in doc and presentation	
Scope limited with respect to 3D Modelling	360 view for published ads introduced
Filter keywords is not enough for Recommendation Module	Recommendation based on advanced machine learning models introduced that filter based on user activity, wishlist and community interactions

Supervised by
Supervisor's Name

Signature _____

1. Introduction

1.1 Scope

The scope of our project is a comprehensive mobile application designed for car enthusiasts and everyday drivers. The core functionality revolves around a user-friendly car search engine. This platform provides users with an engine to look at new and used car listings from dealerships and private parties with many filters to narrow the search and find the right car. To improve car exploration, user will be able to explore 3D models and tour car exteriors virtually. Additionally, a recommendation system will analyze the user profiles and preferences, suggesting appropriate vehicles, upcoming car shows, and personal maintenance tips providing personalized approach. Building a vital online car society, the app will comprise of forum and chat features, which will encourage the users to have the freedom to connect, share their experiences, and ask questions. Providing convenient management of car owning, the app will have streamlined tools of expense tracking, service reminders and a detailed maintenance history. Finally, to ensure platform trust, a vehicle verification system will authenticate listings and dealerships, promoting transparency and security. This project deliberately excludes functionalities beyond car ownership, aiming to deliver a focused and user-centric experience that caters to all aspects of a user's car life cycle.

1.2 Modules

The overall modules of our system are as follows:

1.2.1 Module 1: User Management

FE-1: User registration and login with various options (email, social media, phone number) with two-factor authentication for added security.

FE-2: Allow users to create a profile including personal information.

FE-3: User profile management (name, location, contact information, bio, preferred communication methods, vehicle collection/Wishlist).

FE-4: Account settings management (password change, notification preferences, privacy settings).

FE-5: Users can customize their profiles and can also delete accounts according to their needs.

FE-6: User verification process (E-mail/phone verification, social media verification).

FE-7: Dealership registration and profile management (company details, contact information, brands offered, selling licenses, verification badges).

FE-8: Admin dashboard for managing user accounts, app permissions, verification requests, content moderation, and dealership applications.

1.2.2 Module 2: Search and Buy

FE-1: Search for vehicles, parts, dealerships using basic and advanced filters (brand, make, model, year, price range, mileage, location, body style, variant, generation, features, seller type, popularity).

FE-2: Listings with detailed descriptions, high-quality photos/videos, and VIN verification.

FE-3: Sorting options based on price, year, distance, user/dealership rating.

FE-4: Watchlist for saving favorite listings.

1.2.3 Module 3: Post Your Ride or Auto Part

FE-1: Listing creation for vehicles (cars), parts (new/used), and dealership.

FE-2: Ability to upload multiple high-resolution photos and videos from various angles.

FE-3: Detailed description fields for specifications, features, condition, modifications, and history.

FE-4: Inventory management for dealerships with stock level updates.

FE-5: Ability to mark listings as sold or deactivate them.

1.2.4 Module 4: 3D Model Visualization and Comparison

FE-1: Interactive 3D models of vehicles showcasing exterior and interior details.

FE-2: Virtual customization options for colors, rims, wheels, body kits, and accessories.

FE-3: Option to compare customized versions of vehicles side-by-side.

FE-4: Compare up to 3 vehicles side-by-side based on key specifications, features, user ratings, and pricing history.

FE-5: Visual comparison tool with 3D model rotations (exterior and interior) for a more immersive experience.

FE-6: Ability to overlay key differences between compared vehicles.

FE-7: Option to save and share car comparisons with others.

1.2.5 Module 5: Chat and Notification

FE-1: Real-time chat functionality with private messaging between buyers and sellers, community members, and dealership representatives.

FE-2: Group chat functionality for creating communities or clubs around specific car interests.

FE-3: Push notifications for messages, replies, inquiries, offers, purchase confirmations, and reminders

FE-4: Saved searches with automatic notifications for new listings matching criteria.

FE-5: Option to customize notification preferences for different categories.

1.2.6 Module 6: Review and Ratings

FE-1: User reviews and ratings for vehicles, parts, dealerships, and individual sellers.

FE-2: Filtering of reviews based on criteria.

FE-3: Option to report inappropriate reviews.

FE-4: Ability for sellers and dealerships to respond to reviews publicly.

1.2.7 Module 7: Vehicle Verification

- FE-1: Integration with third-party government databases for vehicle history reports (accidents, ownership history, taxes).*
- FE-2: Offer a vehicle inspection service to verify car condition*
- FE-3: Driver's license verification for added security and make bookings.*

1.2.8 Module 8: Events and News

- FE-1: Display upcoming car shows, test drives, swap meets, and industry events in the local area with user reviews and RSVP options.*
- FE-2: Latest news and articles related to specific car brands, car awards, technology, industry trends.*
- FE-3: Read and post new blogs related to car industry.*
- FE-4: Ability for dealerships to sponsor events and publish targeted news articles.*
- FE-5: Provide real-time updates on fuel prices.*

1.2.9 Module 9: Forums and Community

- FE-1: Dedicated forum sections for different car types (muscle cars, SUVs, electric vehicles), repair topics, and general car discussions.*
- FE-2: Allow users to post questions, share experiences, and interact with each other.*
- FE-3: Question and answer platform for users to seek advice and help from other car enthusiasts and professionals.*
- FE-4: User-generated mini-vlogs, content sharing (photos, videos, stories) and short videos showcasing car projects, reviews, and maintenance tips.*
- FE-5: User-created clubs and groups based on specific car interests, car brands, or geographic locations*

1.2.10 Module 10: Recommendation System

- FE-1: Integration with machine learning algorithms to identify user trends and suggest relevant content dynamically.*
- FE-2: Content-based recommendations for relevant car listings, events, forums, and clubs based on user activity and browsing history.*
- FE-3: Personalized recommendations for car purchases based on user budgets, preferences, and previous searches.*

1.2.11 Module 11: Maintenance and Tracking

- FE-1: OBD-II scanner integration for automated data collection on vehicle health and performance metrics.*

FE-2: Manual expense tracking for fuel, repairs, maintenance, insurance, and other car-related costs.

FE-3: Automatic generation of monthly and annual expense reports with breakdowns and visualizations.

FE-4: Customizable service reminders based on user-defined intervals for routine maintenance tasks (oil changes, tire rotations, etc.).

FE-5: Integrated maintenance history log for tracking repairs, parts replacements, and service appointments.

1.2.12 Module 12: AI Chatbot

FE-1: Use an AI chatbot for basic customer support and frequently asked questions

FE-2: Assistance with car search, listings, and general information

FE-3: Ability to escalate complex user queries to live customer support representatives.

1.2.13 Module 13: Auto Services Marketplace

FE-1: List or book car-related services like car washes, detailing, or mobile repairs

FE-2: Connect with skilled individuals for car maintenance tasks or car-related assistance

FE-3: Rating and review system for service providers

2. Design Methodology and Software Process Model

2.1 Design Methodology:

In developing Motors Bay, we've adopted a **Object Oriented Programming (OOP) approach** as our chosen design methodology.

2.1.1 Rationale behind this approach:

Object-oriented programming would be best suited to Motors Bay for reasons of modularity, scalability, and most important of all, the consideration of Firebase as a database.

- **Entity Representation:** Real-world entities, such as User, Vehicle, and Booking, will be directly mapped into classes and Firebase collections to make data organization better.
- **Modularity:** Each module, such as User Management and Listings, is encapsulated into different classes for efficient development, testing, and maintenance.
- **Reusability:** If there is an inheritance relationship of a shared functionality, a user can be promoted with very little redundancy: user → buyer, seller.
- **Encapsulation:** Protect sensitive information like passwords while Firebase gives its role-based access reinforcement.

- **Polymorphism:** Common methods, such as `createListing()`, will behave contextually depending on the user type using them; this increases the flexibility of the system.
- **Scalability:** OOP goes hand in glove with multi-tier architecture at Motors Bay, and Firebase supports real-time updates, which help to handle data efficiently.

By applying OOP, Motors Bay will have a neat, maintainable, scalable design that works best with Firebase.

2.2 Process Methodology

In developing Motors Bay, we've adopted **Agile Scrum** for process methodology.

2.2.1 Rationale behind this approach:

Agile Scrum divides the **Motors Bay** project into smaller, manageable parts called sprints, enabling focused progress and efficient resource use. Team members collaborate on specific tasks, which integrate to form the complete software.

- **Transparency:** Clear and up-to-date information ensures seamless collaboration and smarter decision-making.
- **Adaptability:** Incremental delivery allows easy changes, keeping the software relevant and aligned with priorities.
- **User-Friendly Design:** Regular feedback helps create intuitive and user-friendly software.
- **Team Ownership:** Team members focus deeply on tasks, fostering expertise and responsibility.

Agile Scrum ensures flexibility, continuous improvement, and an efficient development process for **Motors Bay**.

3. System Overview

Motors Bay is a new and innovative smartphone application designed to simplify the automotive marketplace by integrating key features like 3D vehicle visualization, personalized recommendations, and maintenance tracking. Unlike existing platforms that focus on specific aspects of buying or selling cars, Motors Bay offers a comprehensive solution that brings together car buyers, sellers, and dealerships in one seamless platform. The system also emphasizes trust and transparency with features like vehicle verification and community engagement through forums.

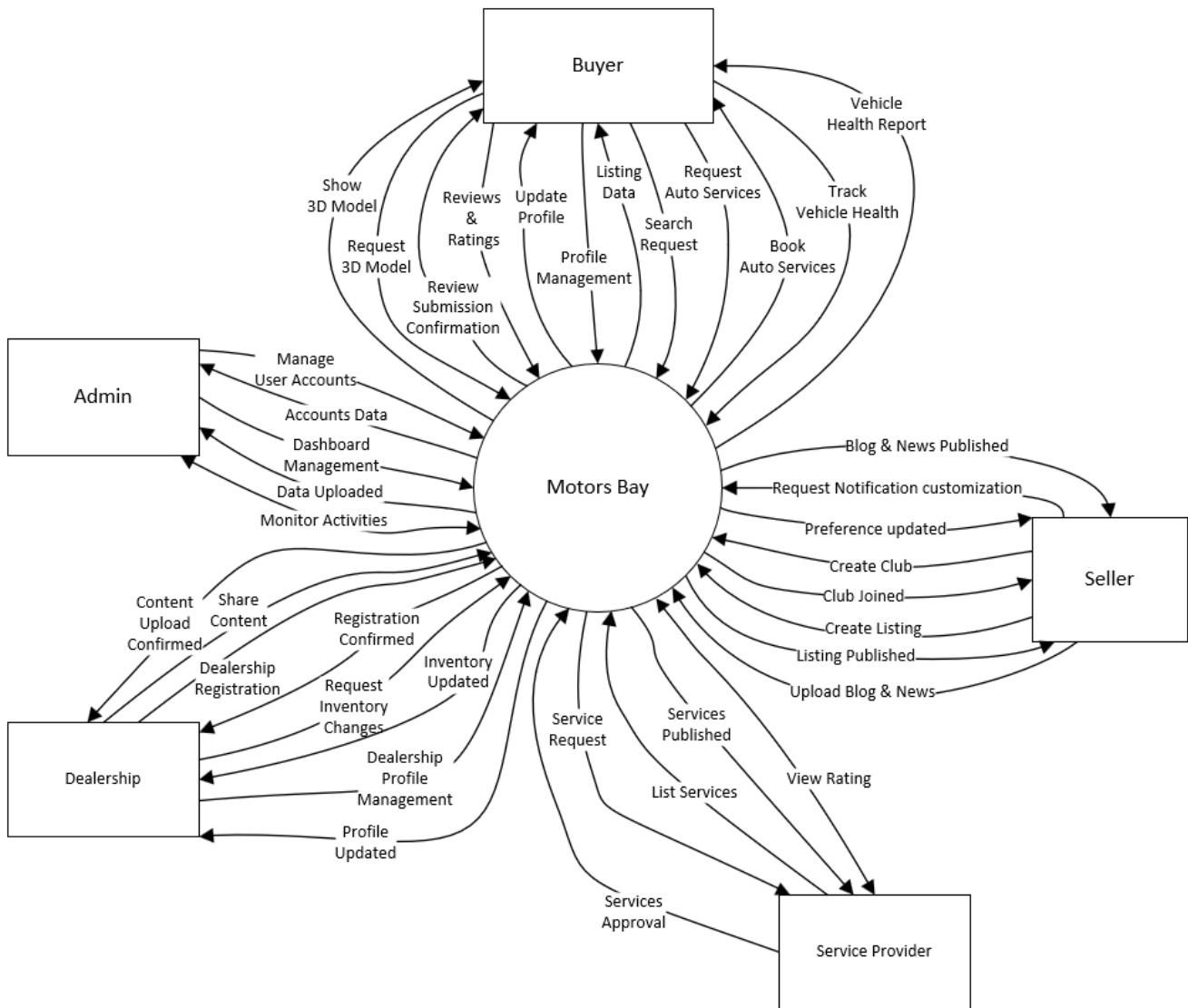


Figure 1: Context diagram of Motors Bay.

3.1 Architectural Design

3.1.1 Box and Line Diagram

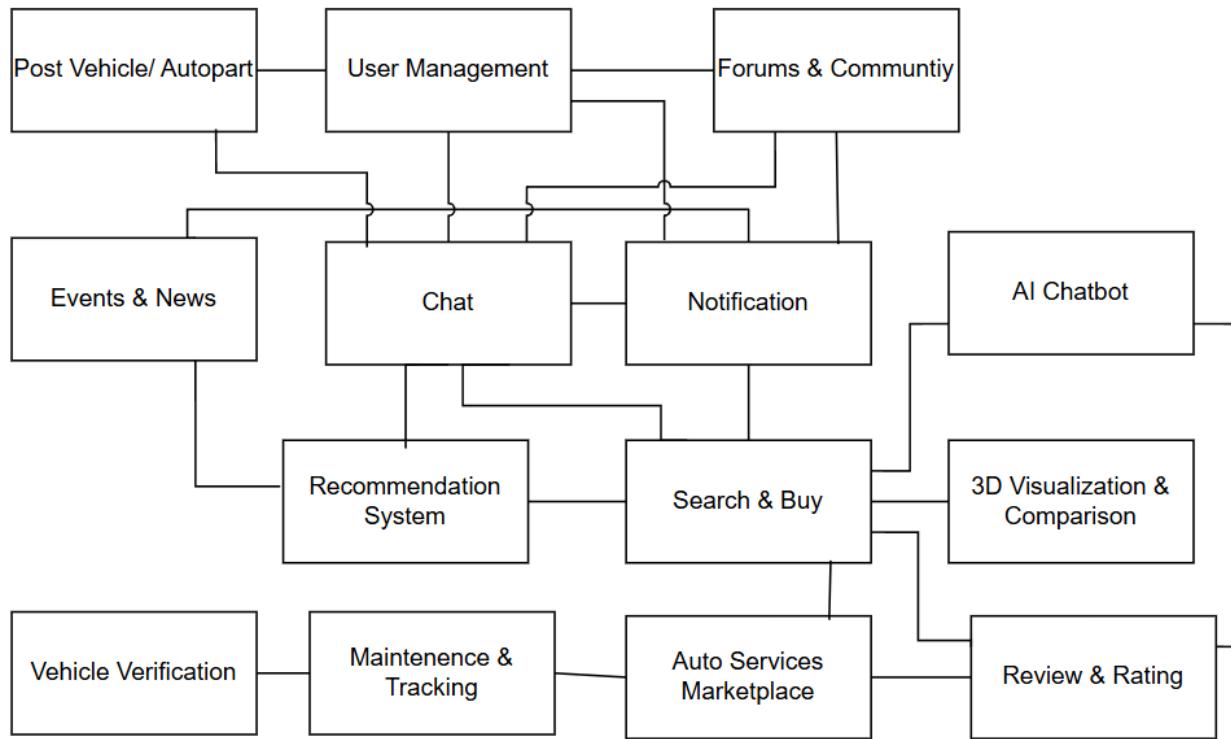


Figure 2: Box and Line diagram of Motors Bay.

3.1.2 Architectural Design

Motors Bay follows a **multi-tier architecture** to ensure modularity, scalability, and maintainability. The **Presentation Layer** manages user interactions through intuitive UI components for modules like User Management, Listings, and Chat. The **Application Layer** handles business logic, supporting workflows such as booking, verification, and recommendations. The **Data Layer** securely stores user profiles, vehicle listings, and maintenance records in Firebase. The **External Integration Layer** connects to APIs for vehicle verification, OBD-II, and notifications. This design ensures efficient layer communication, adaptability to evolving needs, and seamless module integration.

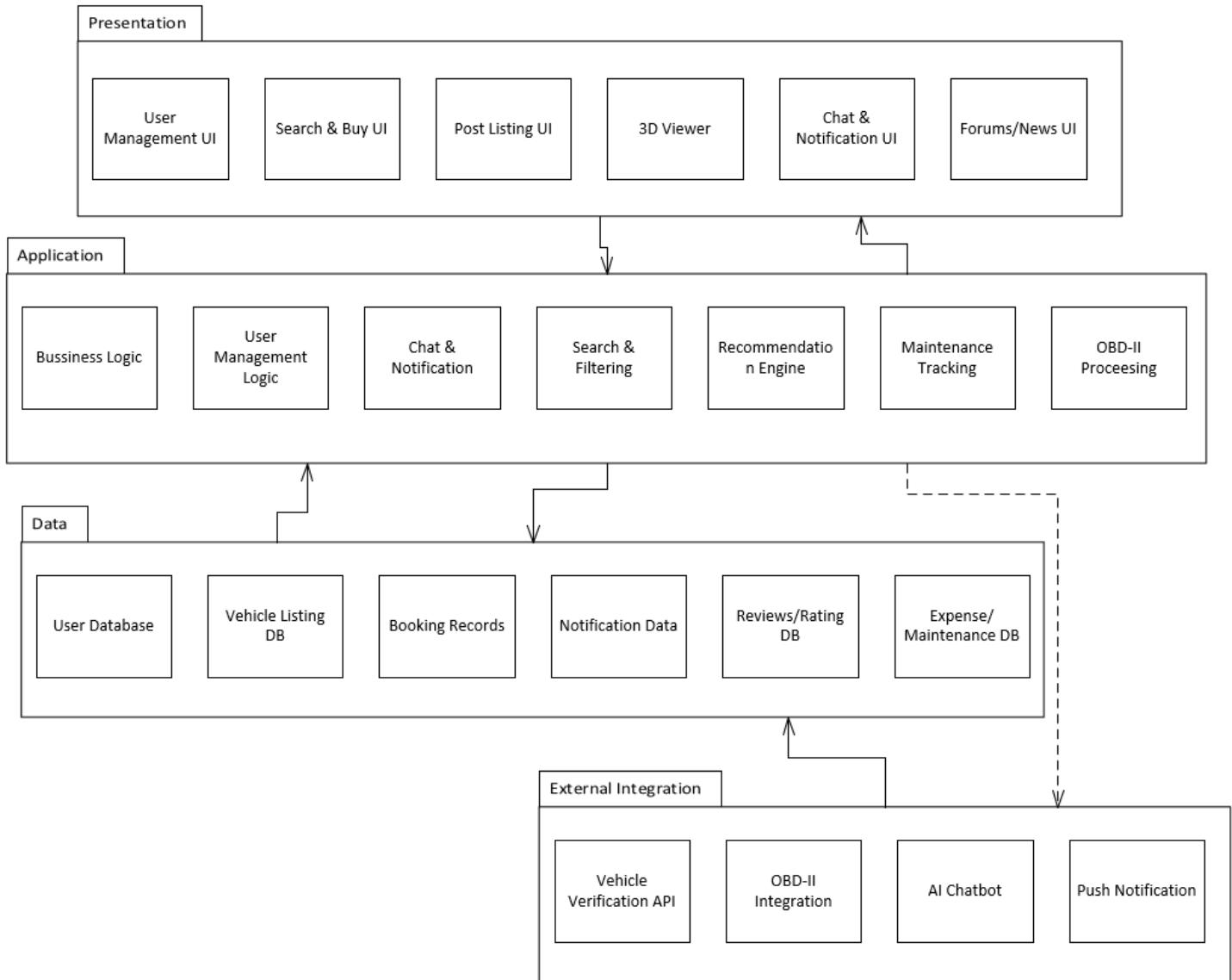


Figure 3: Multi-tier Architecture diagram of Motors Bay.

4. Design Models

4.1 Activity Diagrams

4.1.1 Register User

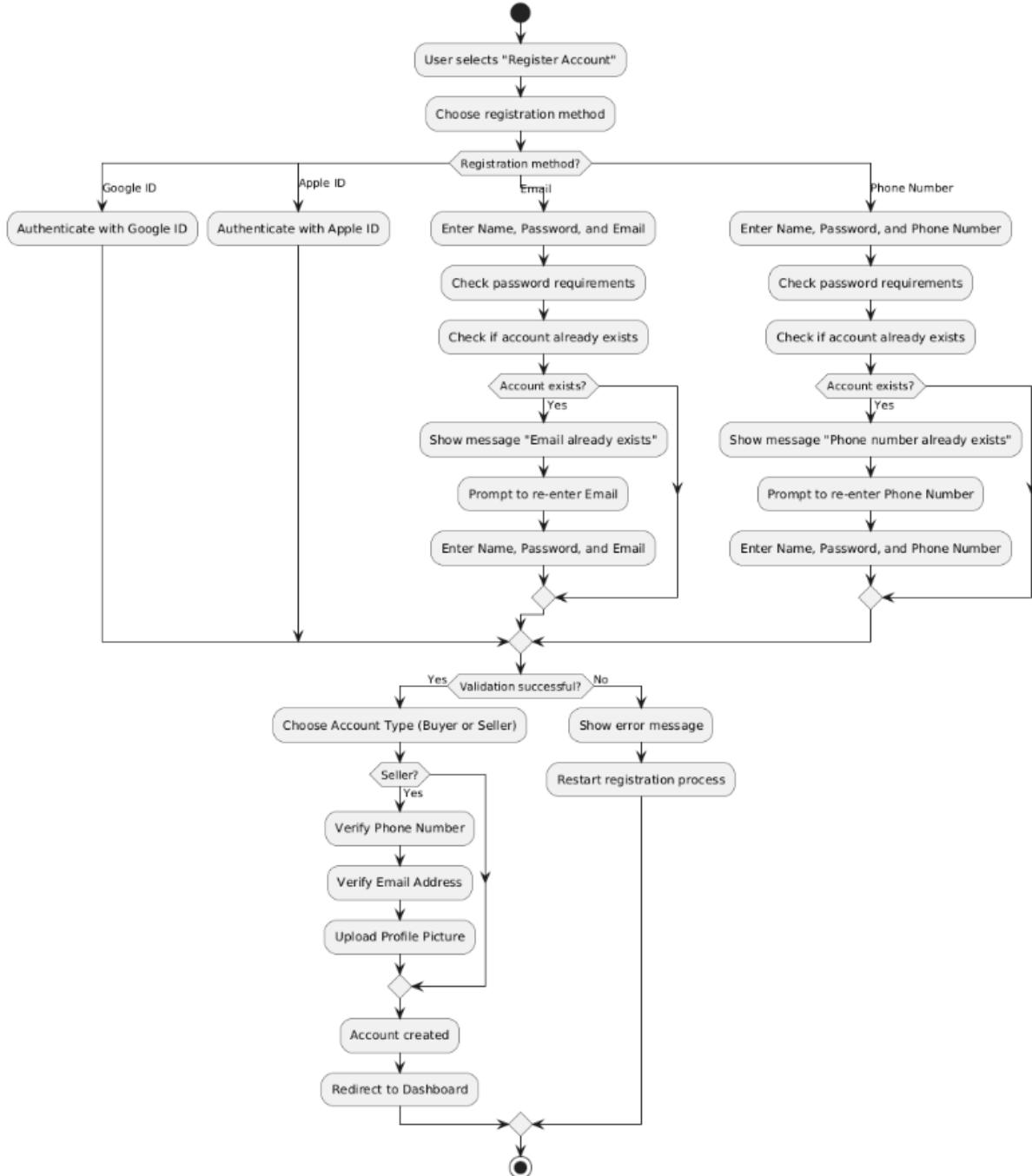


Figure 4: Activity diagram for Register User.

4.1.2 Login

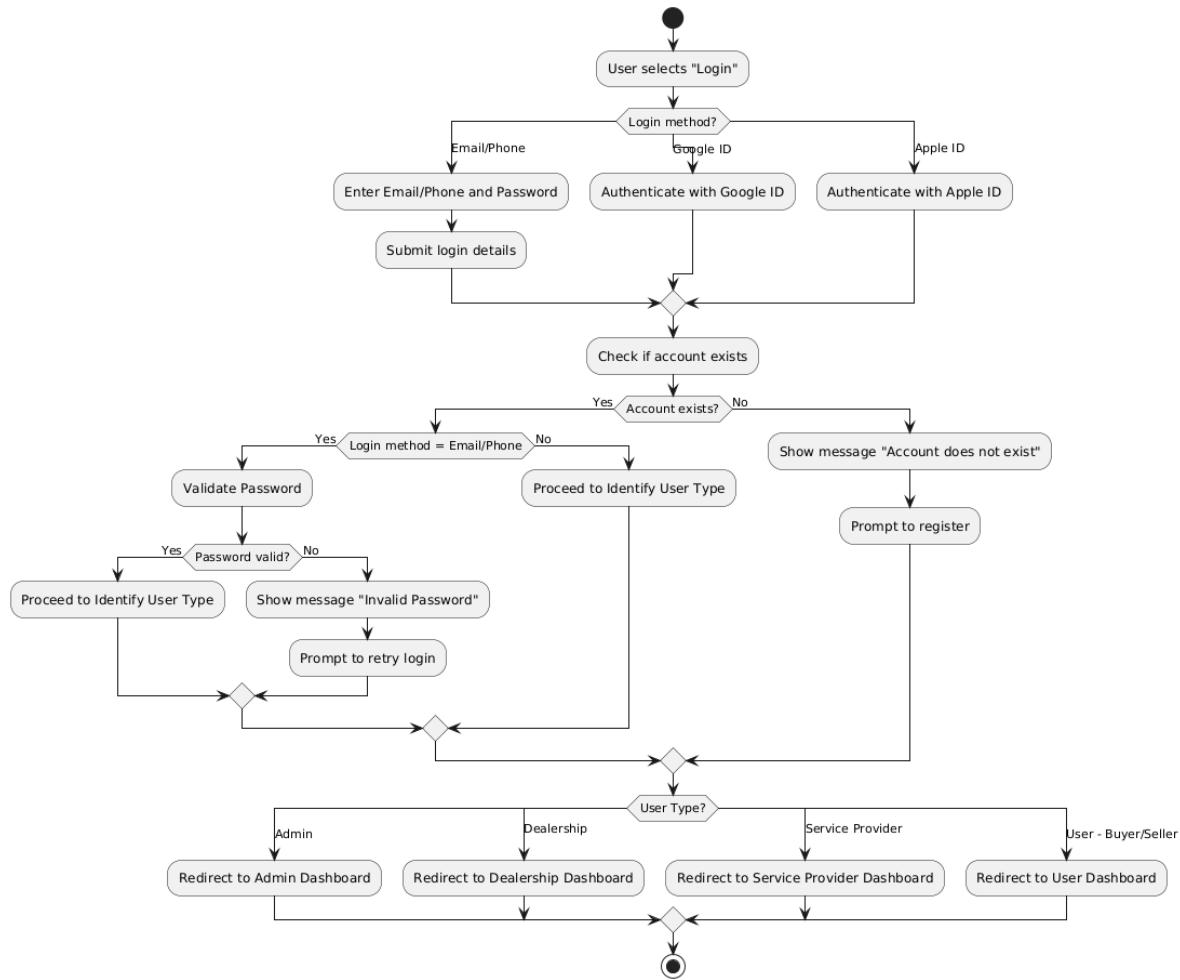


Figure 5: Activity diagram for Login.

4.1.3 User Profile Management

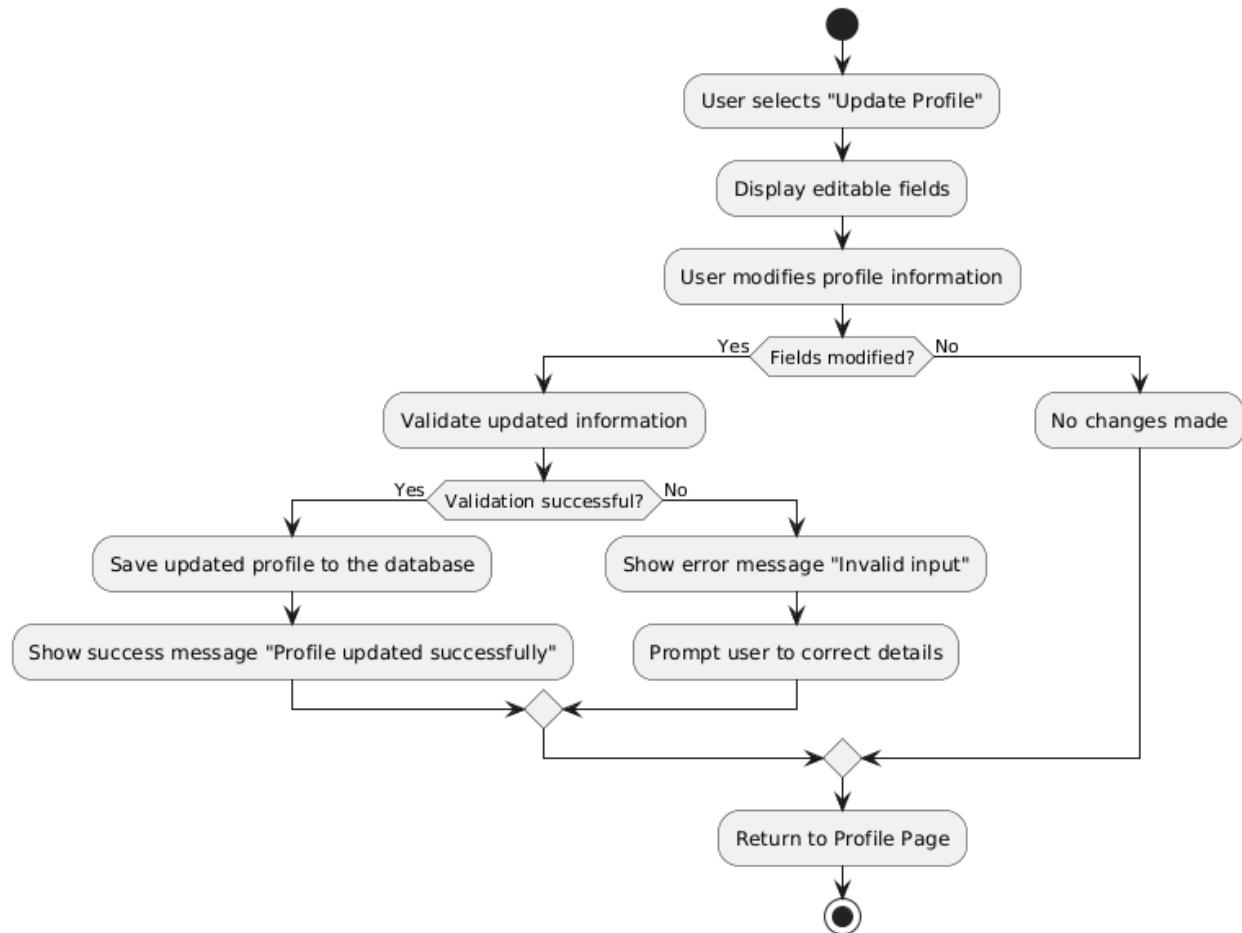


Figure 6: Activity diagram for User Profile Management.

4.1.4 Admin Dashboard Management

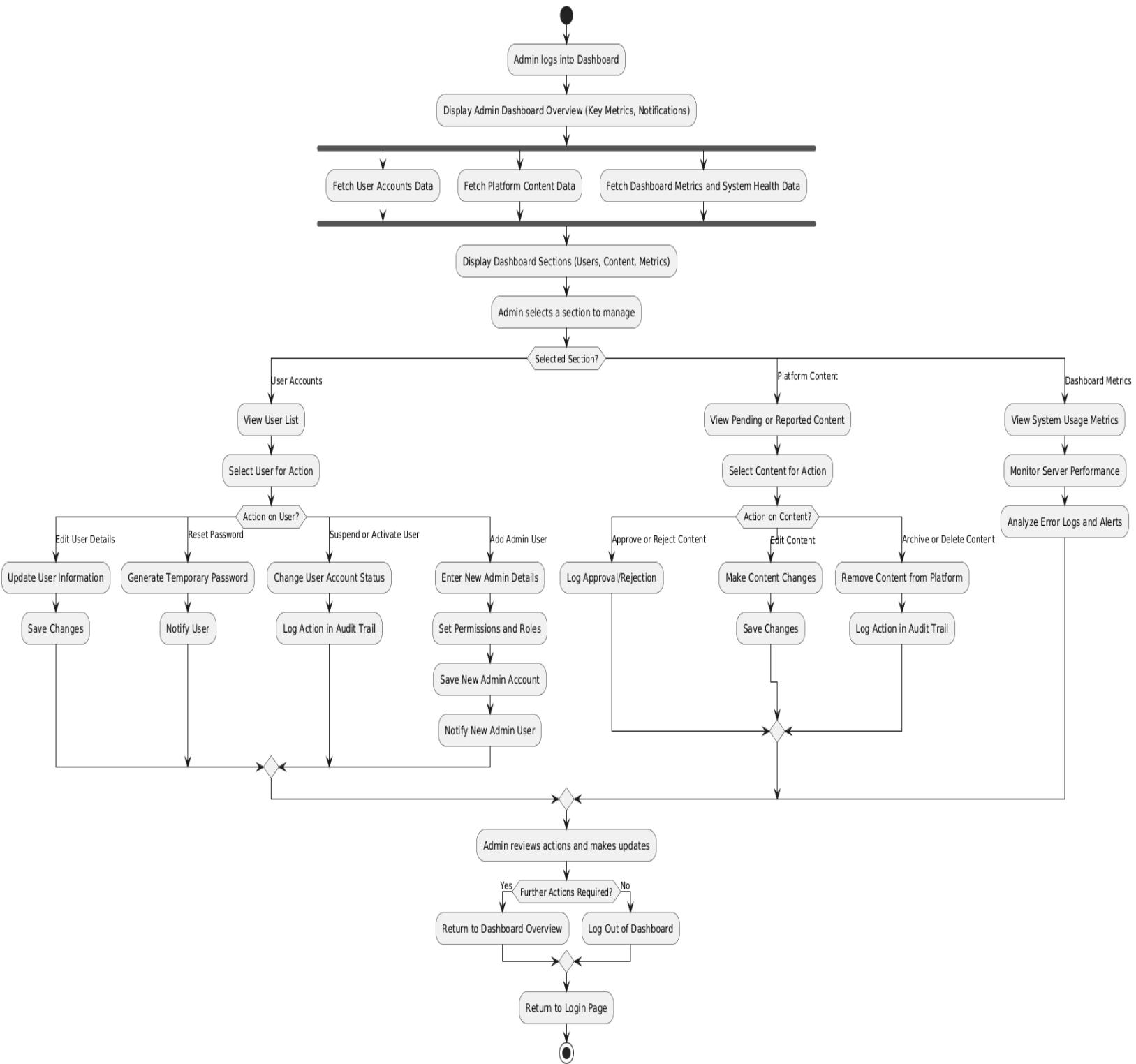


Figure 7: Activity diagram for Admin Dashboard Management.

4.1.5 Dealership Registration

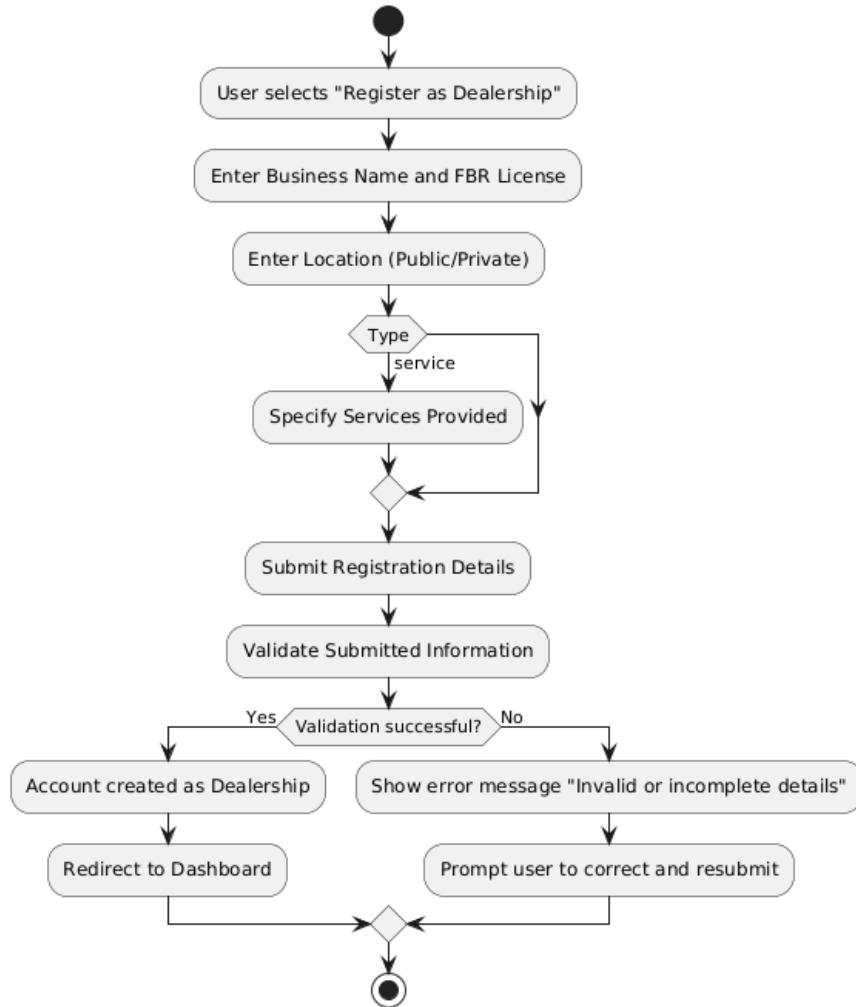


Figure 8: Activity diagram for Dealership Registration.

4.1.6 Dealership Profile Management

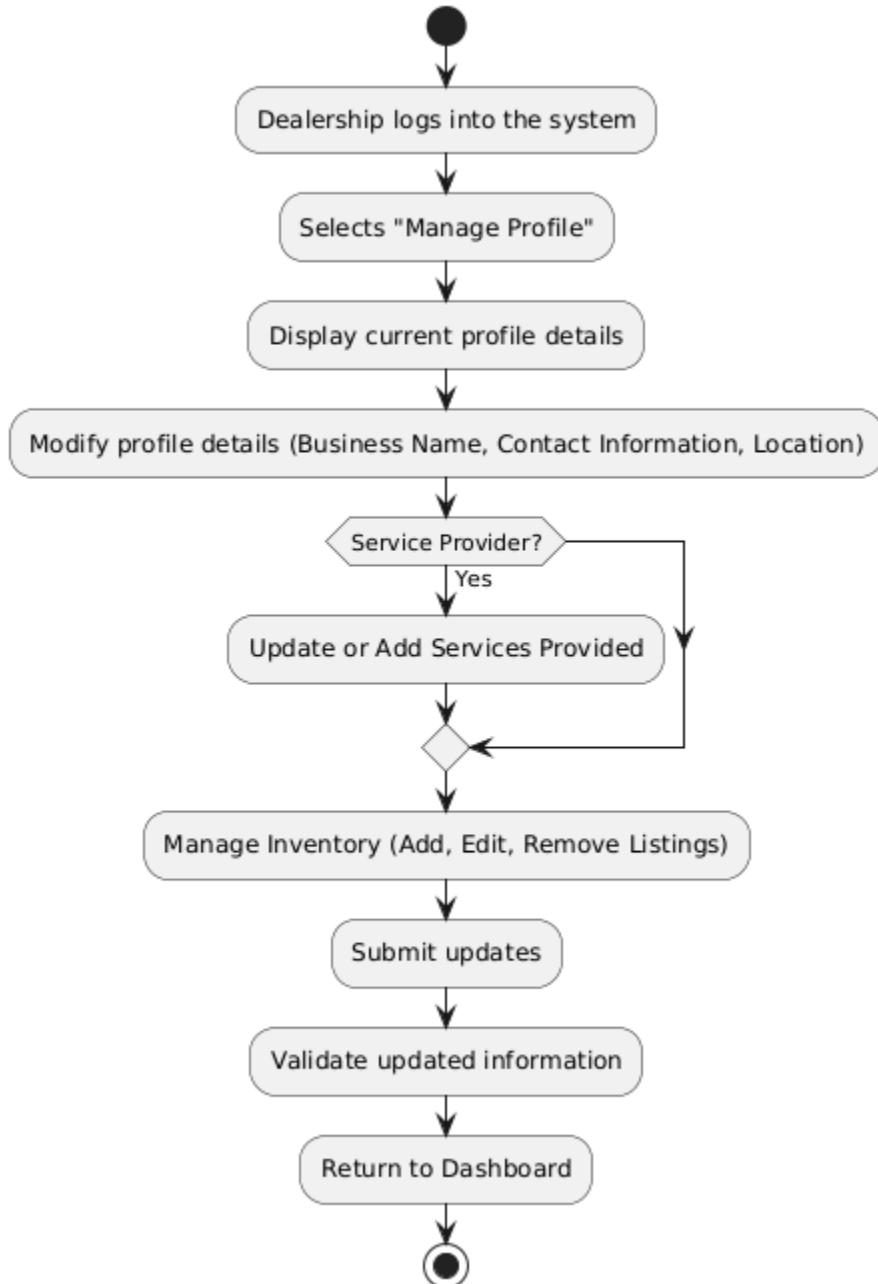


Figure 9: Activity diagram for Dealership Profile Management.

4.1.7 Search

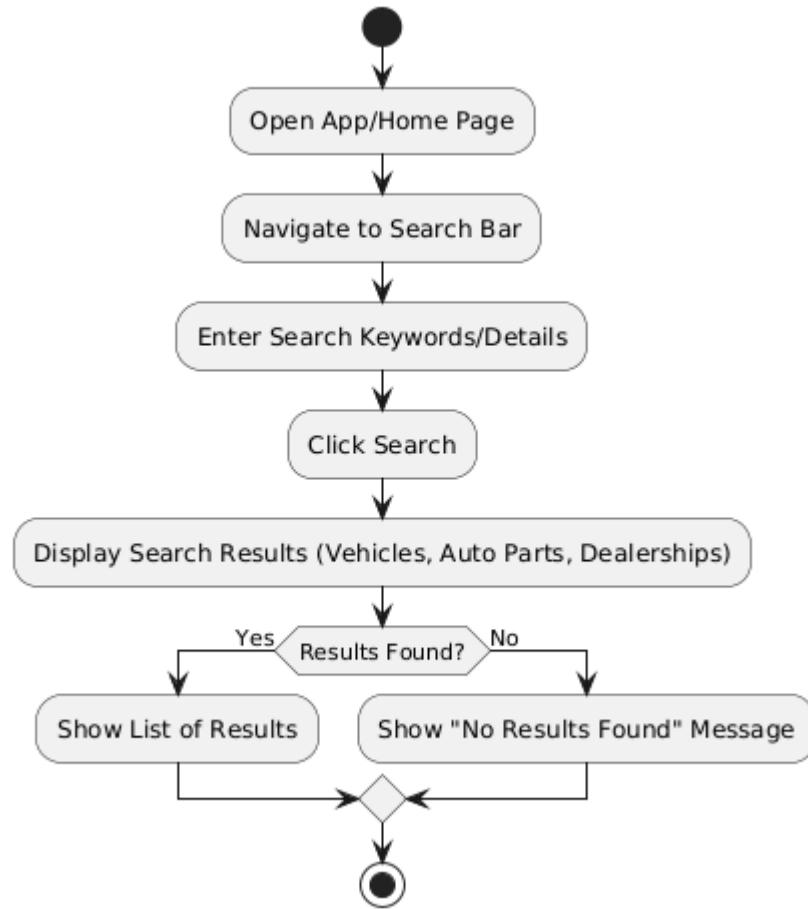


Figure 10: Activity diagram for Search.

4.1.8 Filter and Sort Search Results

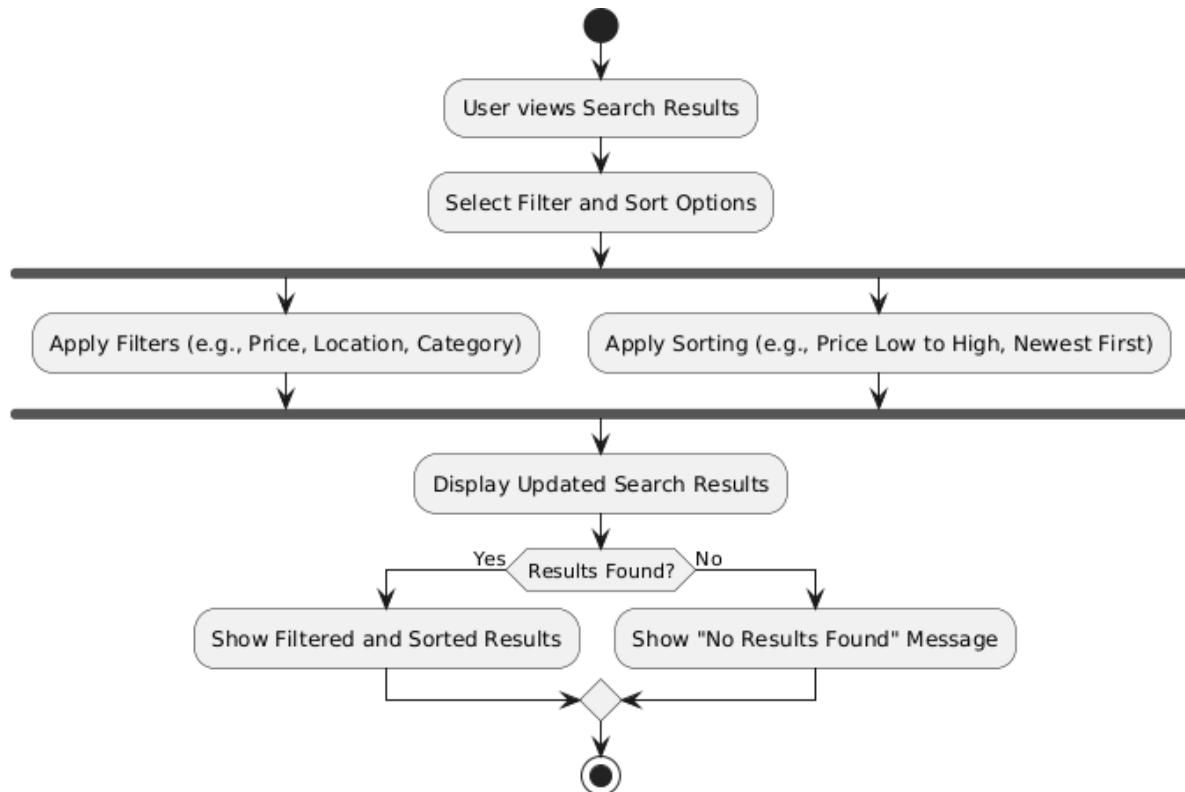


Figure 11: Activity diagram for Filter & sort Search Results.

4.1.9 Add to Watchlist

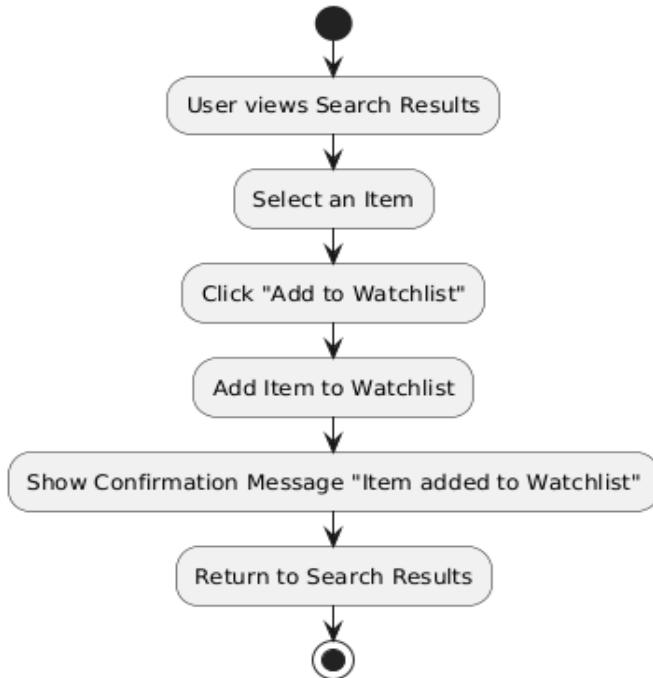


Figure 12: Activity diagram for Watchlist.

4.1.10 Post Ride or Auto part

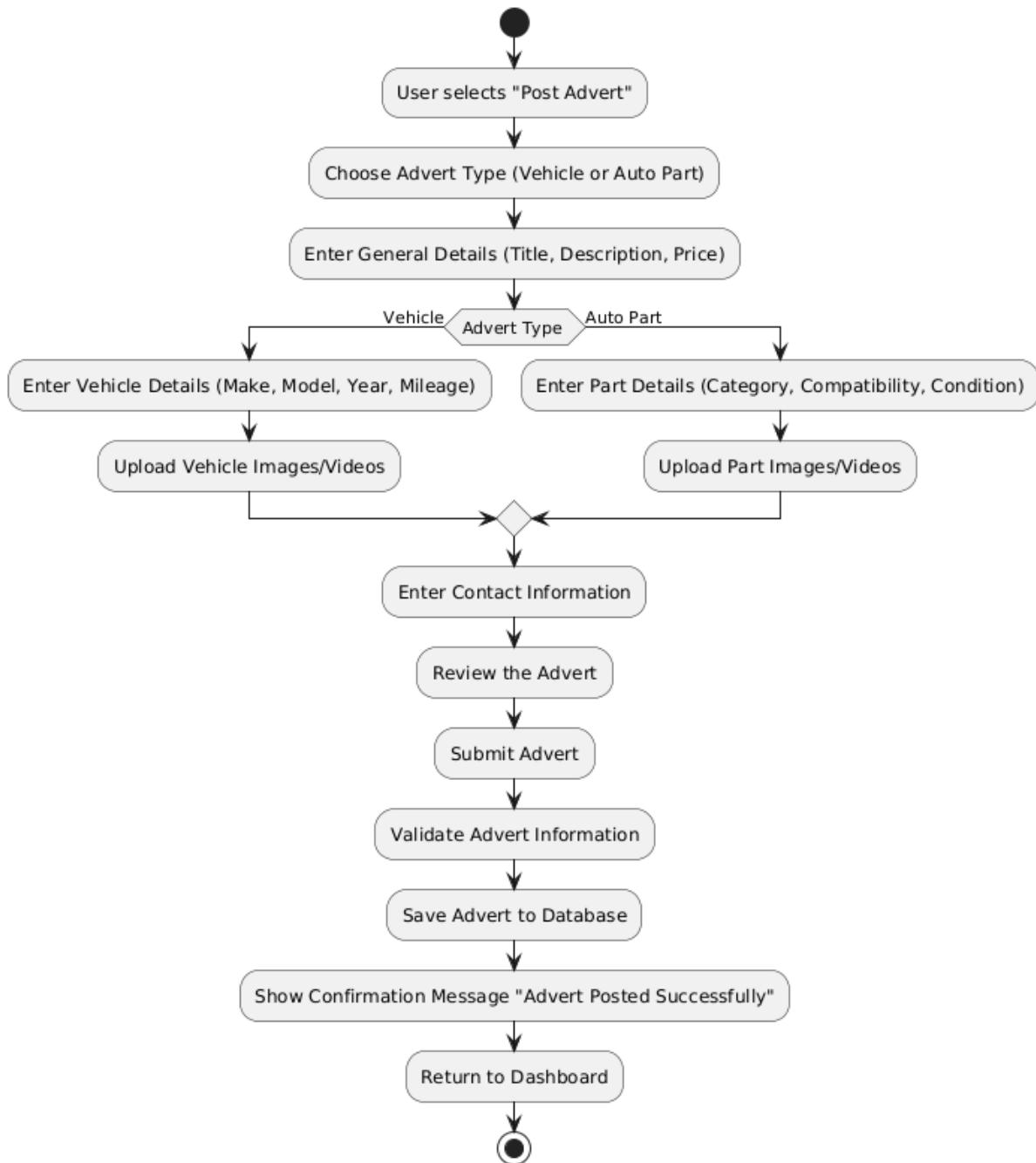


Figure 13: Activity diagram for Post Ride Or Auto Part.

4.1.11 Manage Dealership Inventory

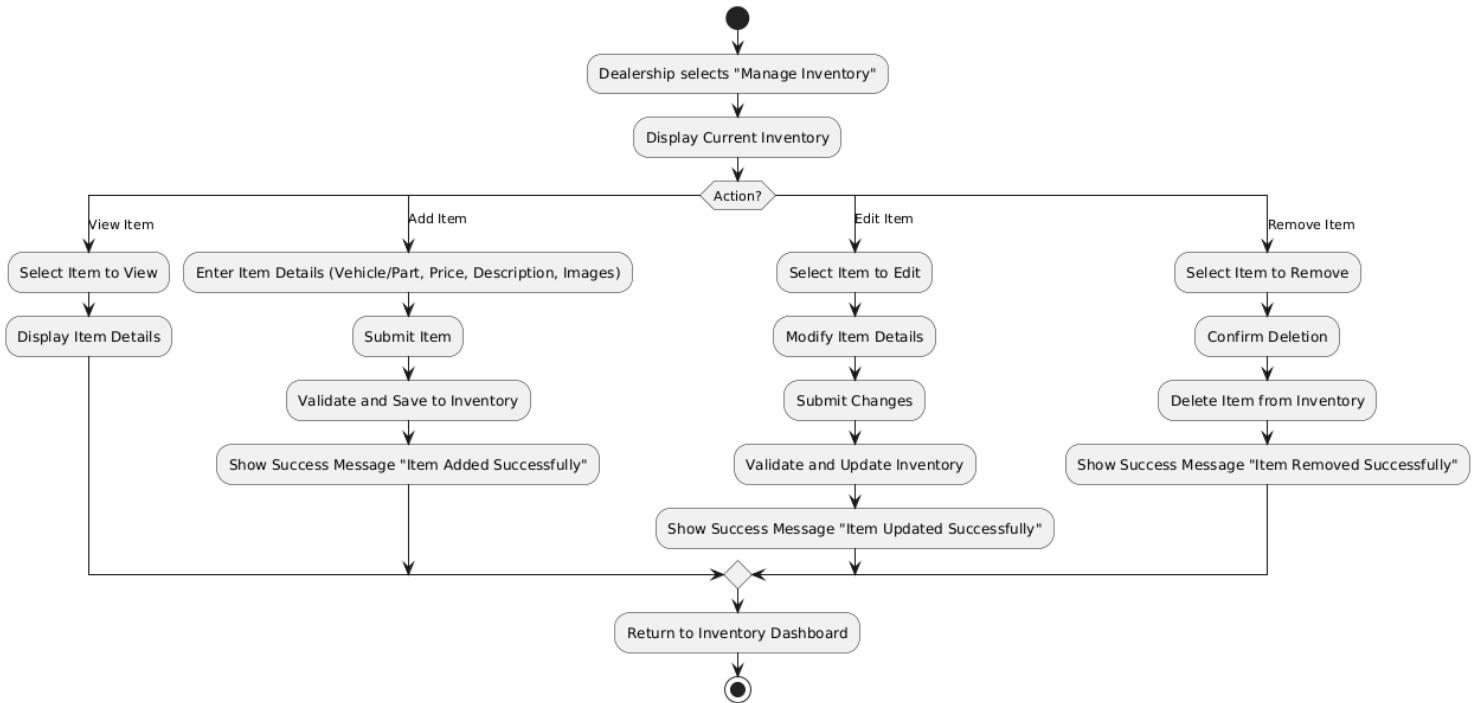


Figure 14: Activity diagram for Manage Dealership Inventory.

4.1.12 Upload Media

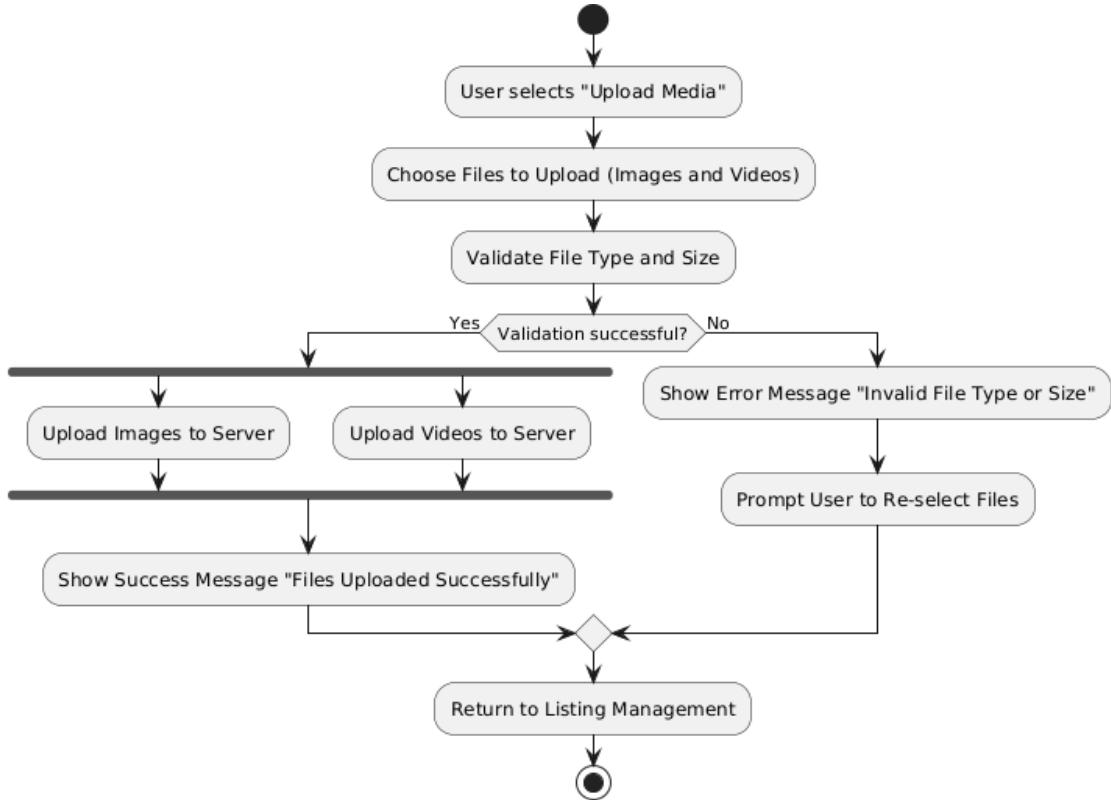


Figure 15: Activity diagram for Upload Media.

4.1.13 Advert Status

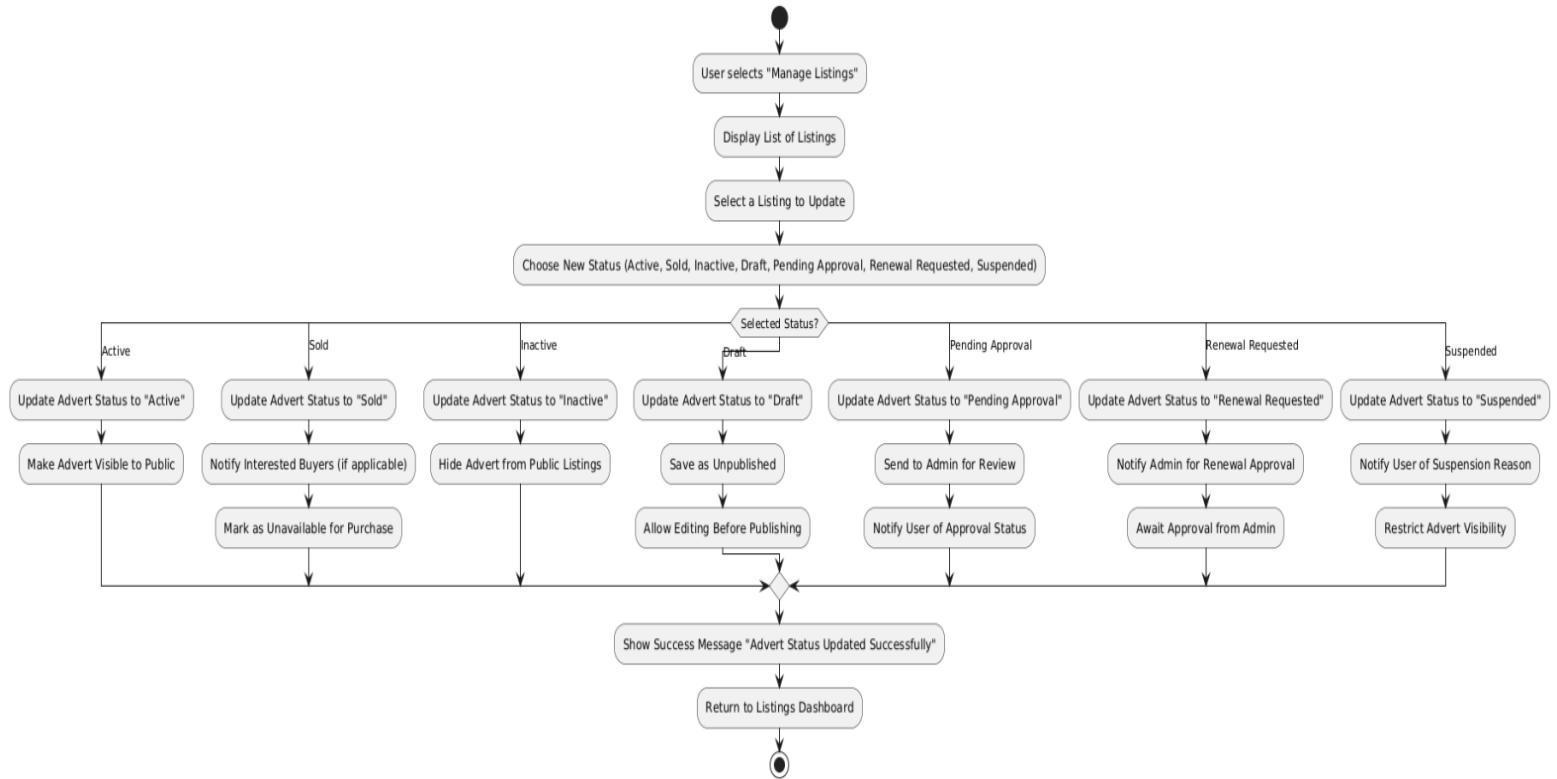


Figure 16: Activity diagram for Advert Status.

4.1.14 3D Model of a Vehicle

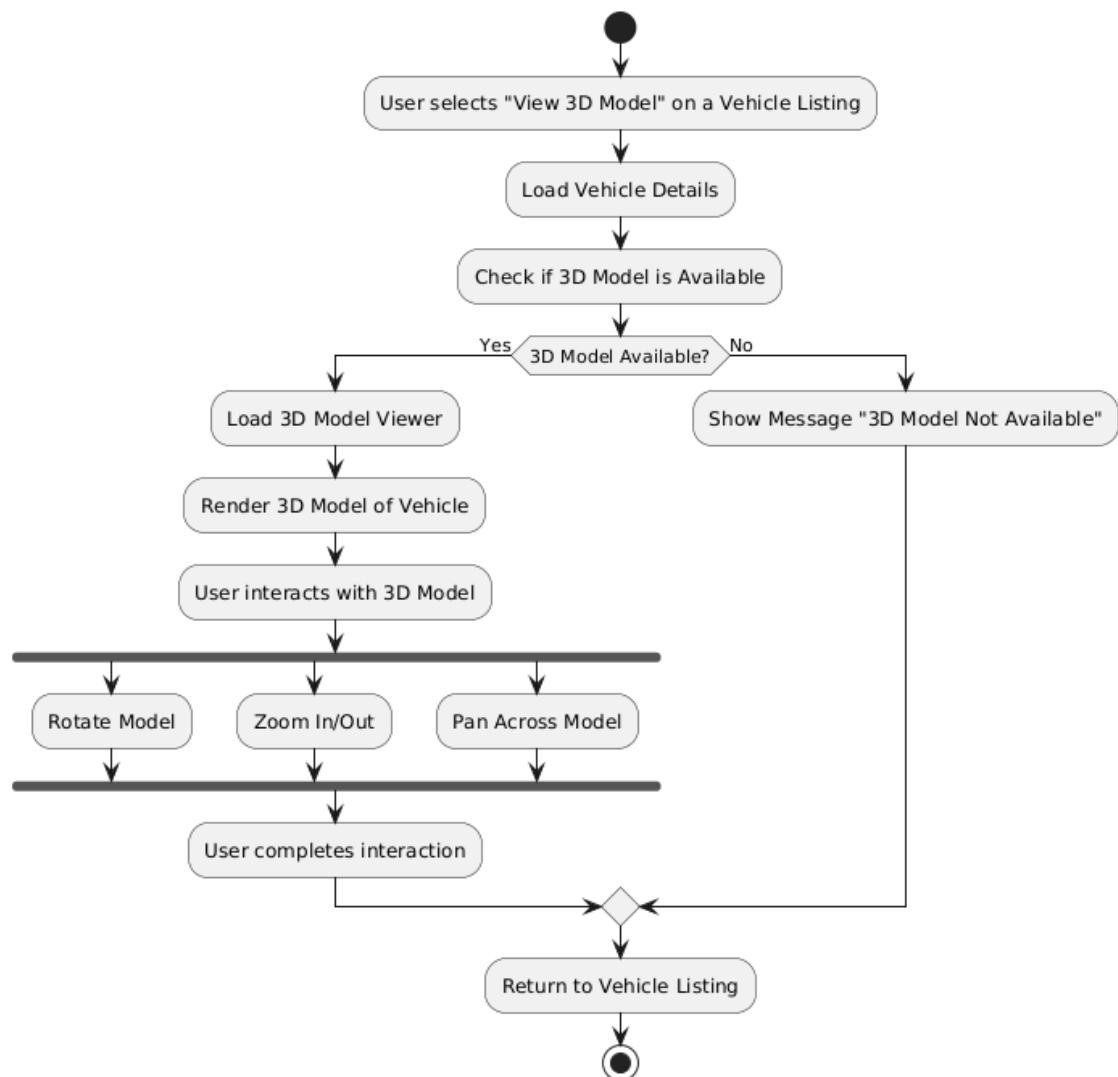


Figure 17: Activity diagram for 3D Model of Vehicle.

4.1.15 Customize Vehicle in 3D

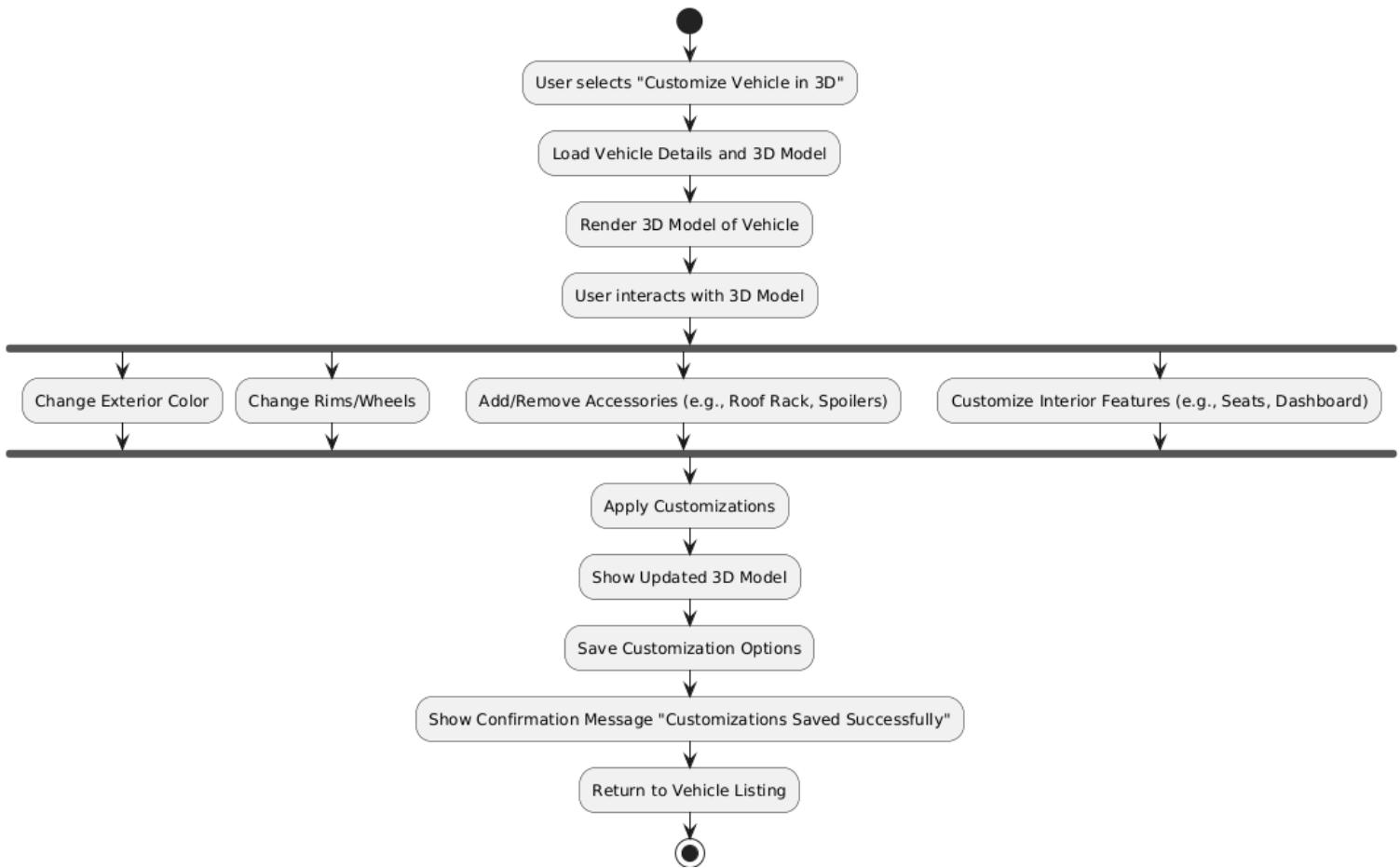


Figure 18: Activity diagram for Customize Vehicle in 3D.

4.1.16 Compare Vehicles

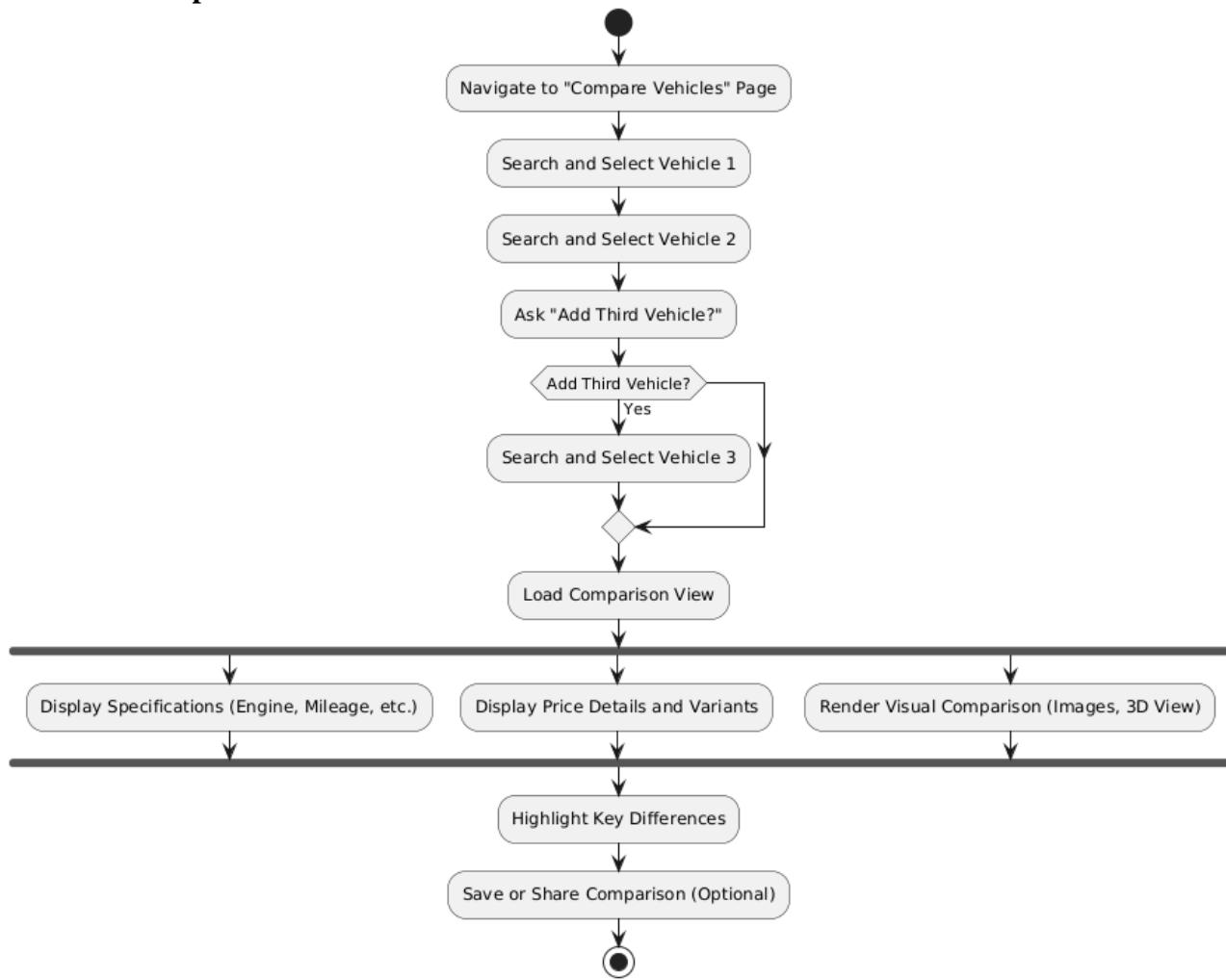


Figure 19: Activity diagram for Compare Vehicles.

4.1.17 Real-Time Private Messaging

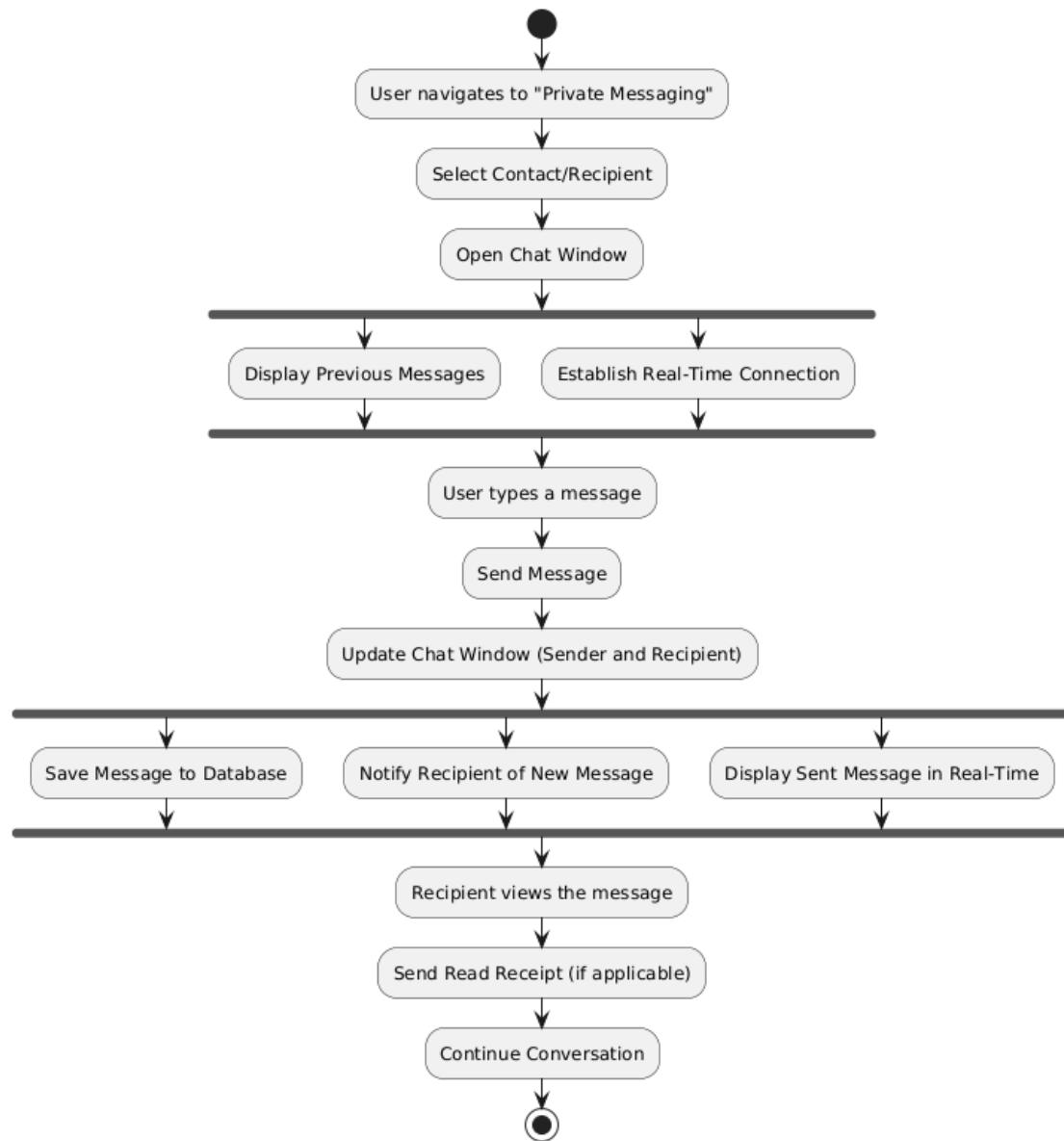


Figure 20: Activity diagram for Real-Time Private Messaging.

4.1.18 Receive Push Notifications

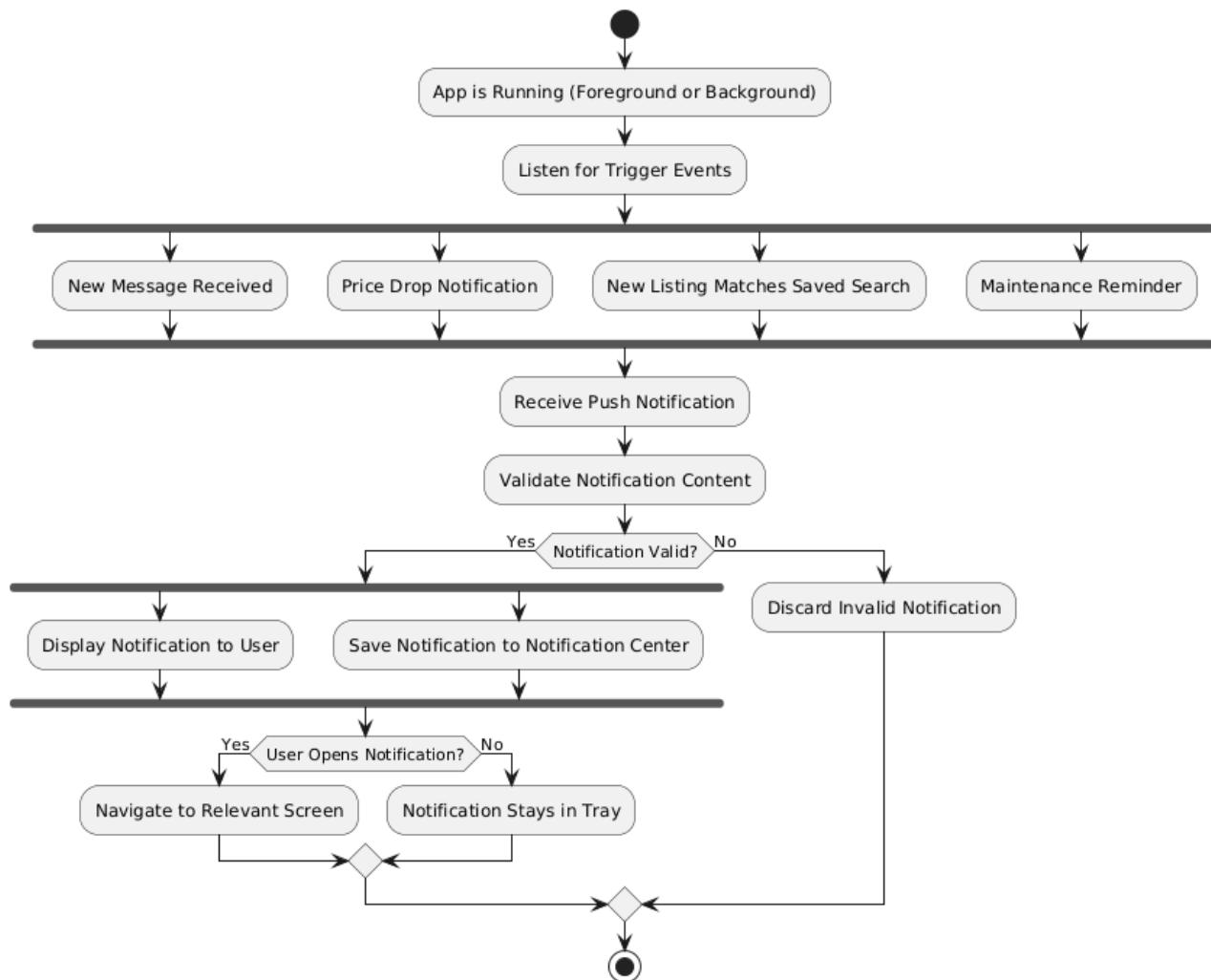


Figure 21: Activity diagram for Receive Push Notifications.

4.1.19 Customize Notification Preferences

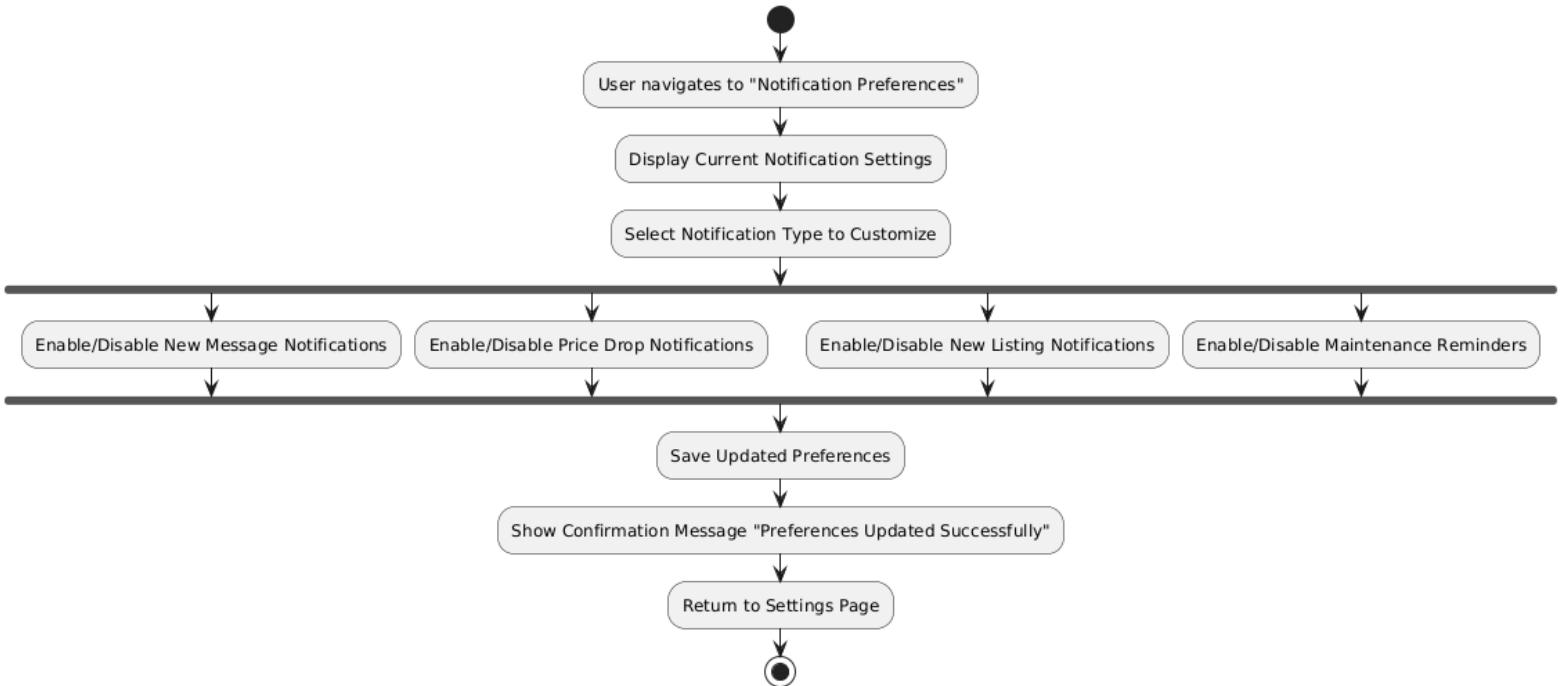


Figure 22: Activity diagram for Customize Notification Preferences.

4.1.20 Submit Review

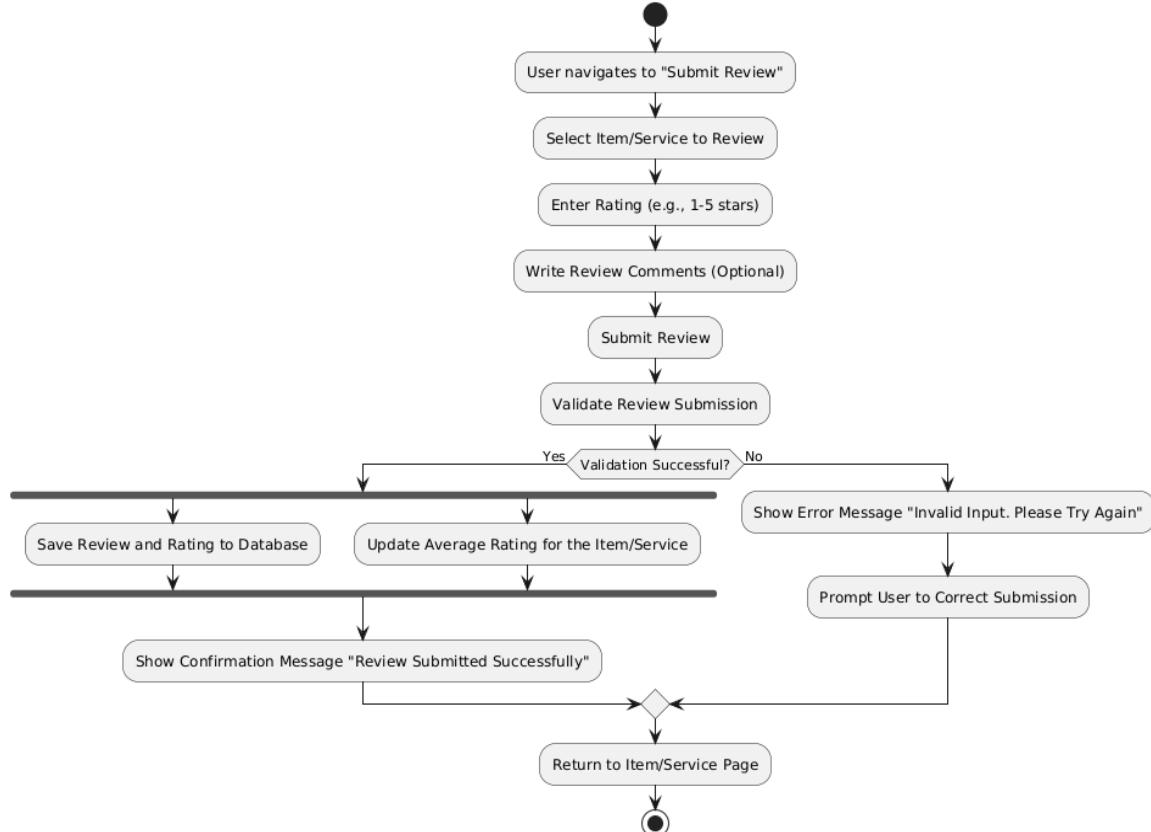


Figure 23: Activity diagram for Submit Review.

4.1.21 Filter reviews

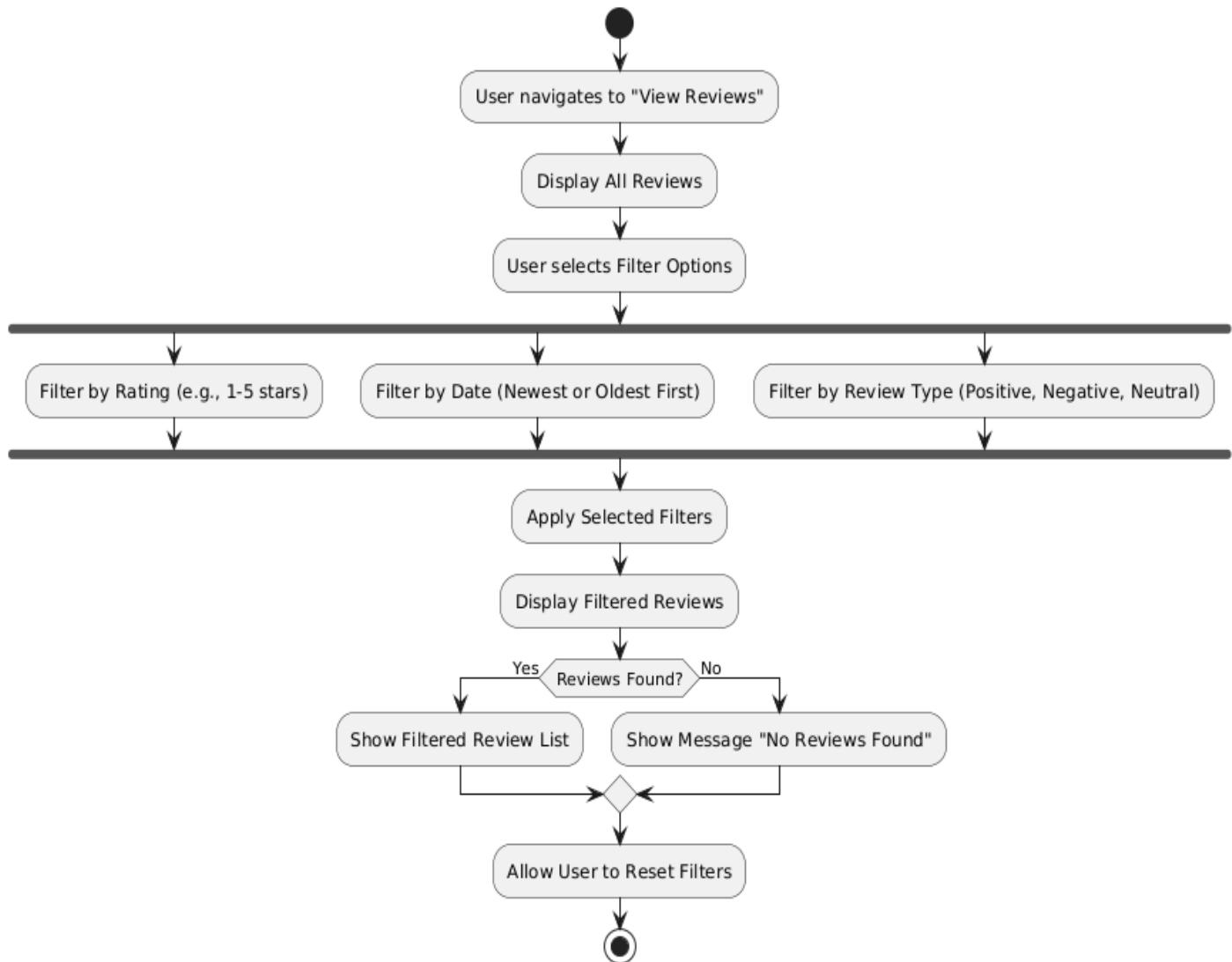


Figure 24: Activity diagram for Filter Review.

4.1.22 Report Reviews

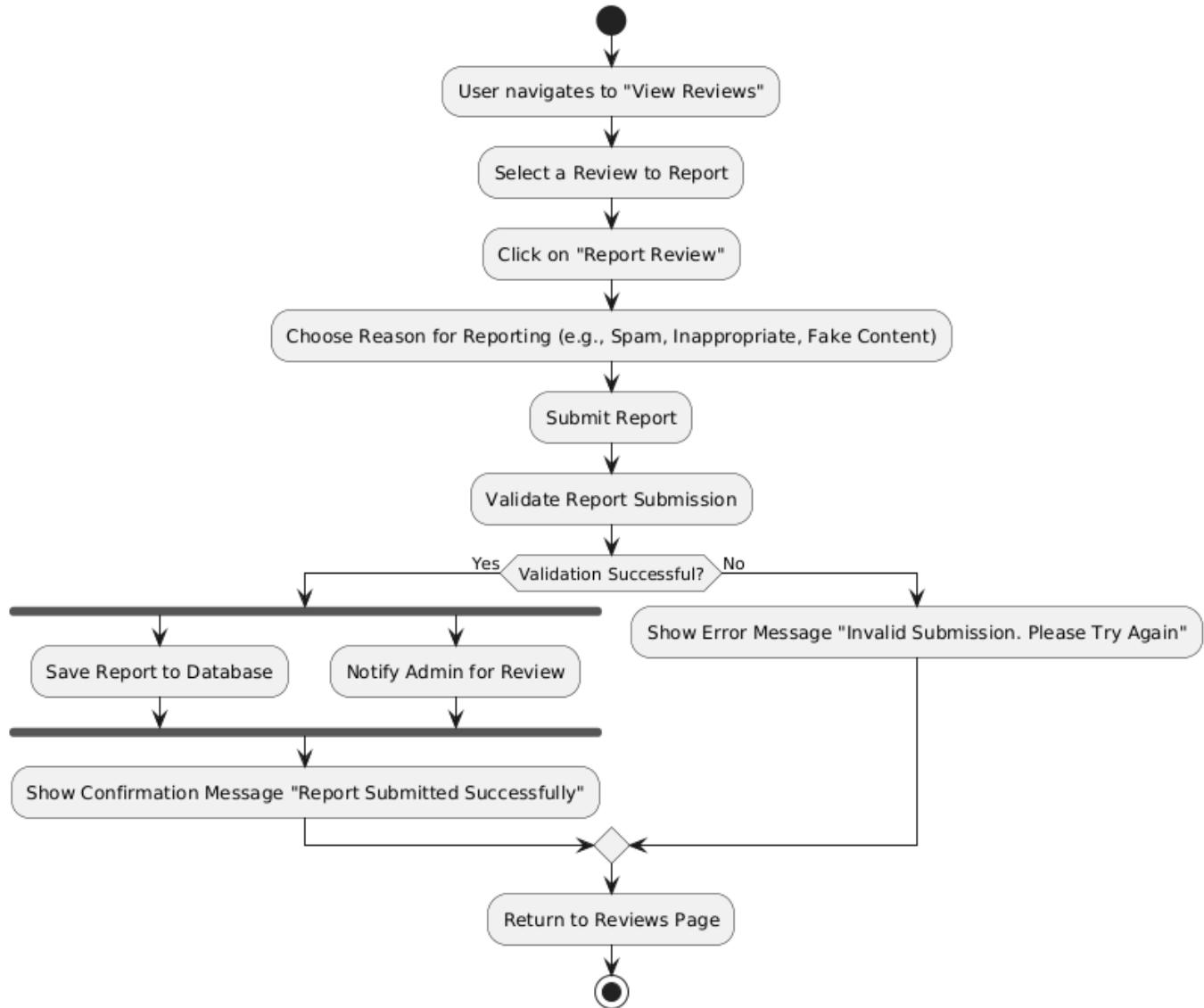


Figure 25: Activity diagram for Report Review.

4.1.23 Vehicle Verification



Figure 26: Activity diagram for Vehicle Verification.

4.1.24 Request Vehicle Inspection



Figure 27: Activity diagram for Request Vehicle Inspection.

4.1.25 Verify Driver's License

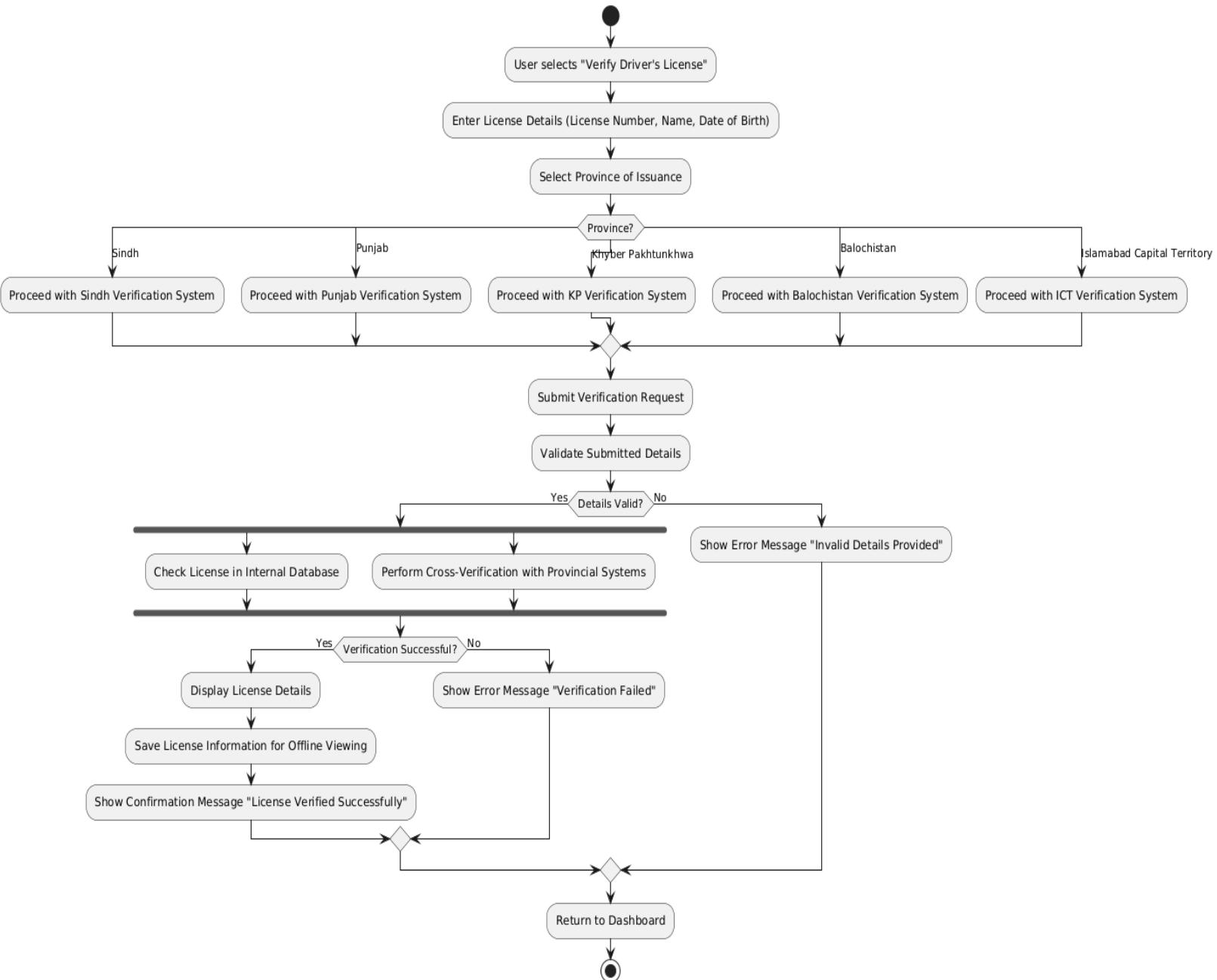


Figure 28: Activity diagram for Verify Driver's License.

4.1.26 Read Car News and Articles

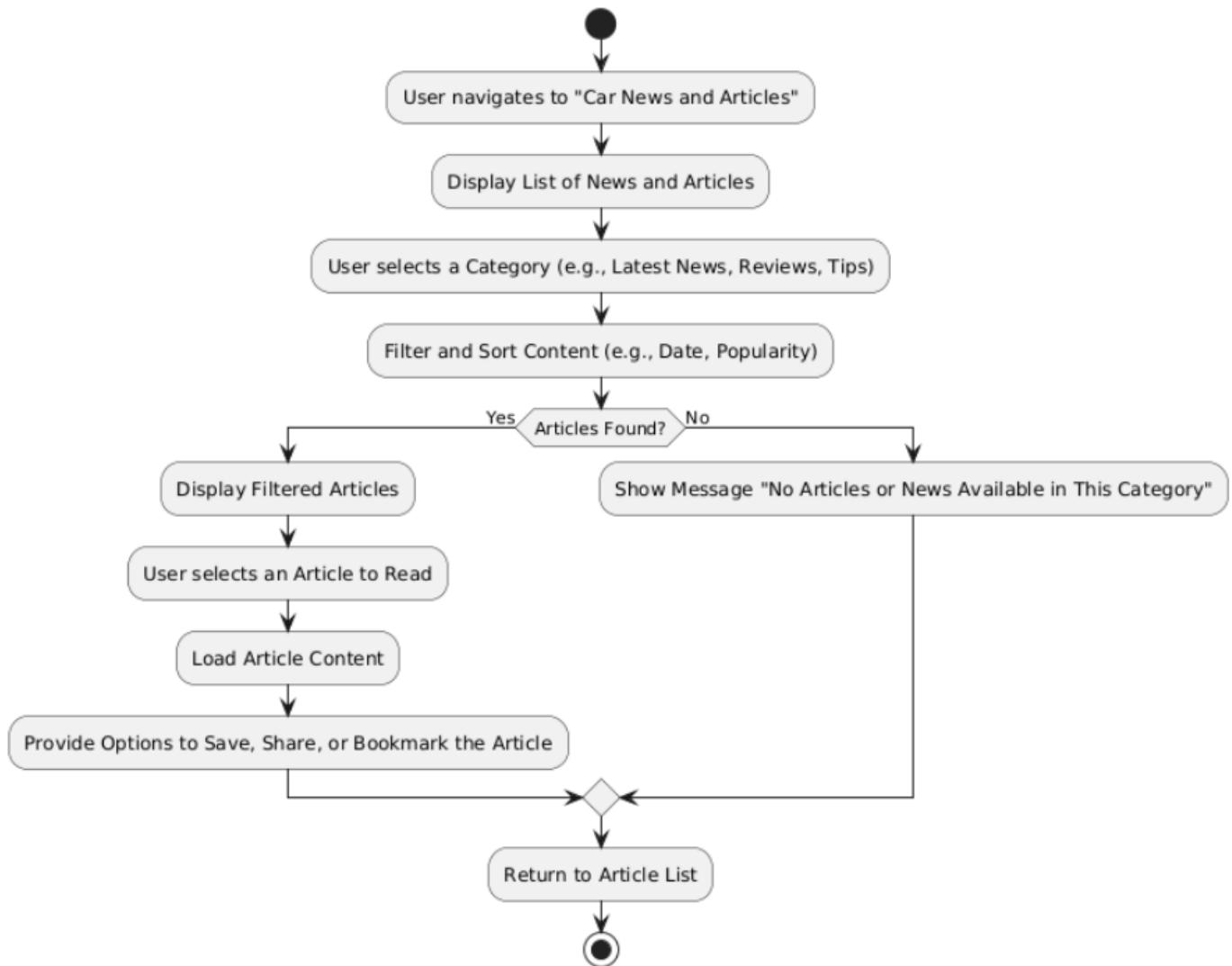


Figure 29: Activity diagram for Read Car News & Articles.

4.1.27 View Fuel Prices

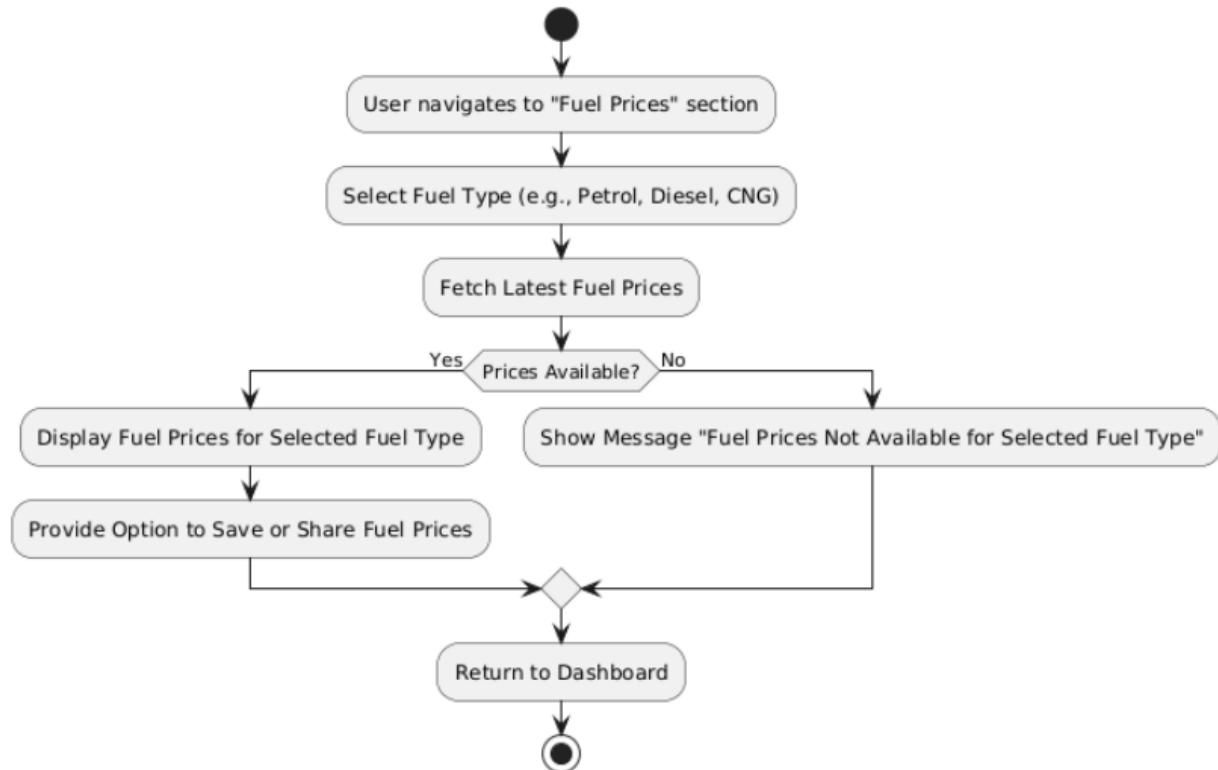


Figure 30: Activity diagram for View Fuel Prices.

4.1.28 Publish Blogs

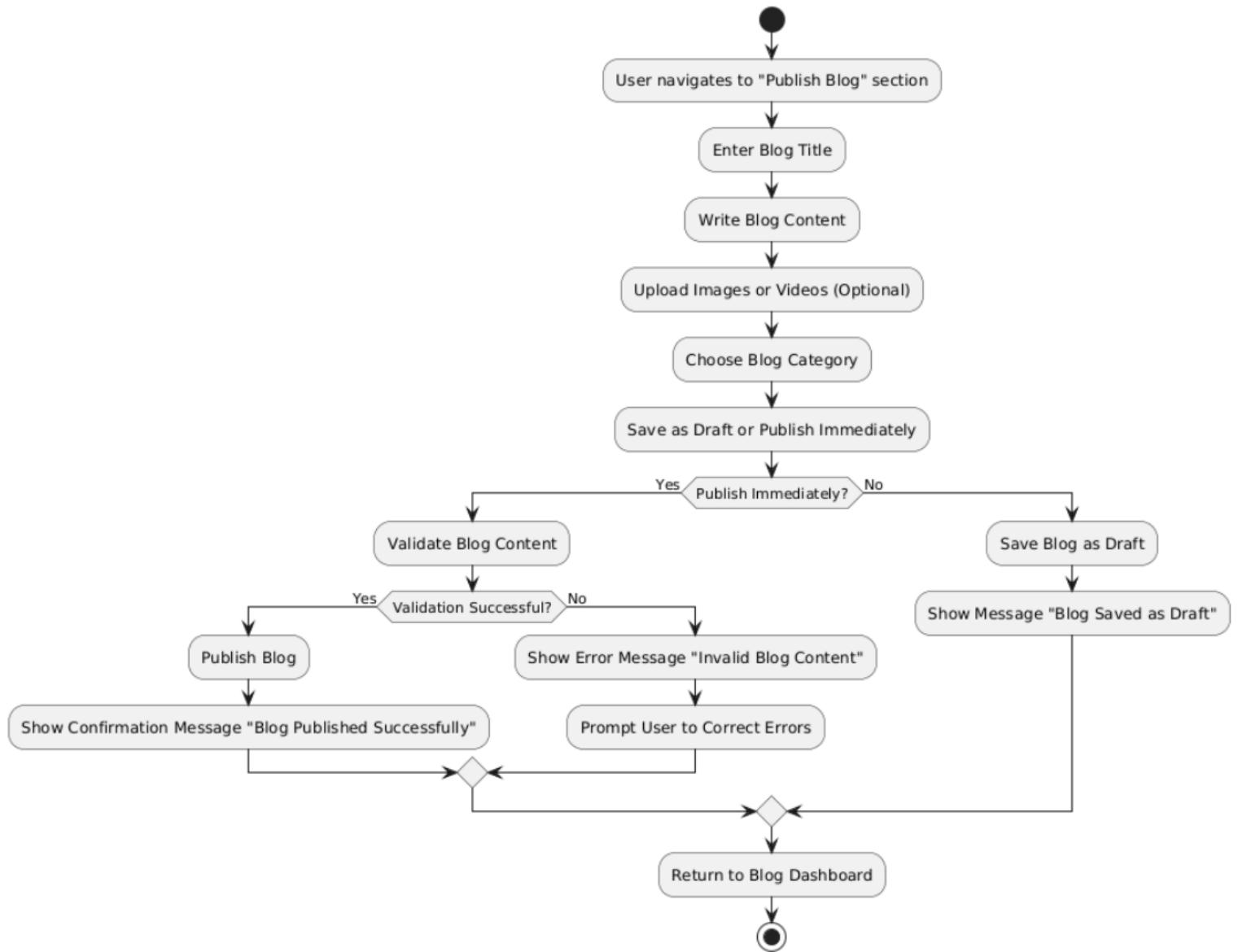


Figure 31: Activity diagram for Public Blogs.

4.1.29 Post Questions and Answers in Forums



Figure 32: Activity diagram for Post Questions & Answers.

4.1.30 Create Groups

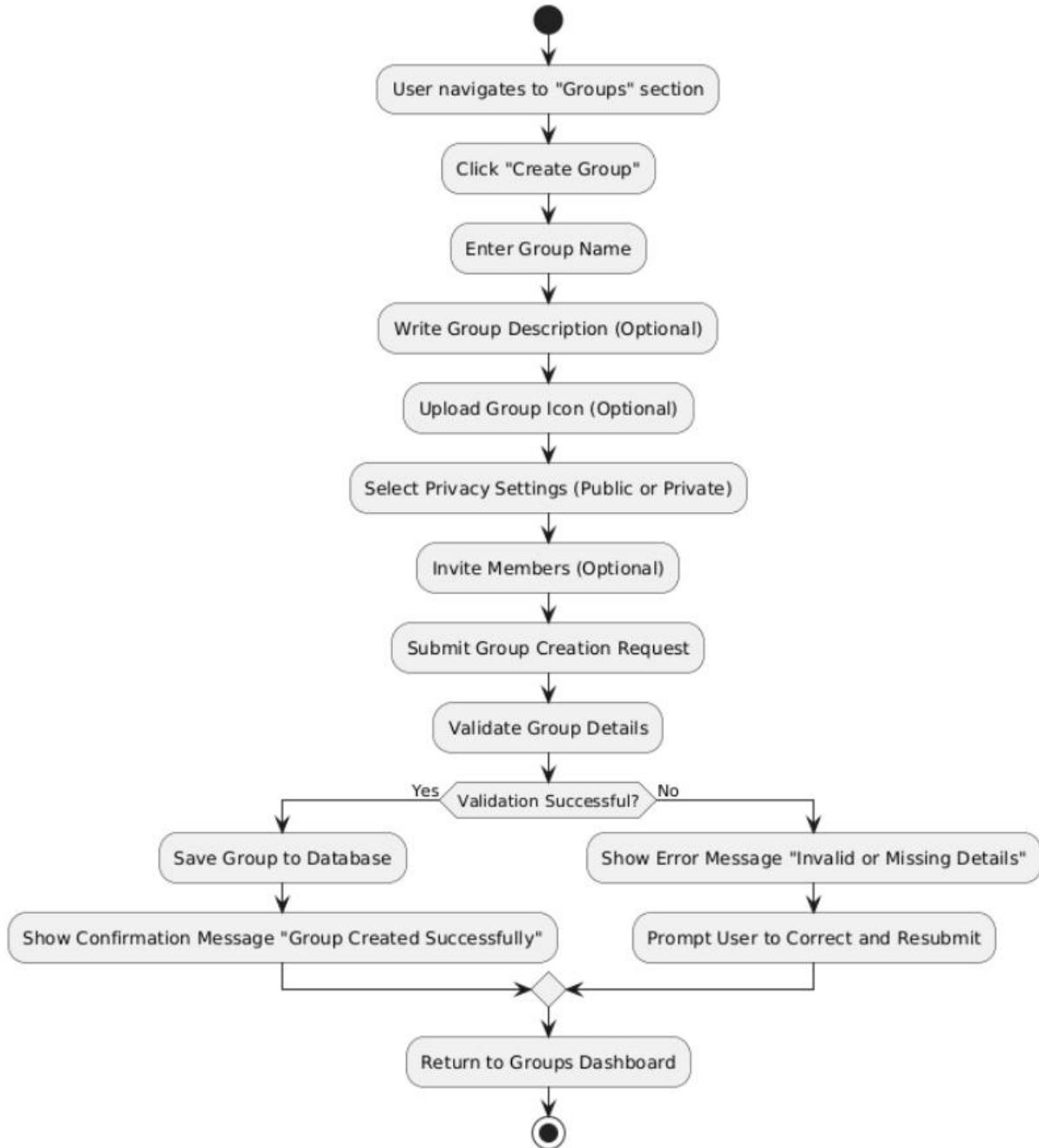


Figure 33: Activity diagram for Create Groups.

4.1.31 Create Club

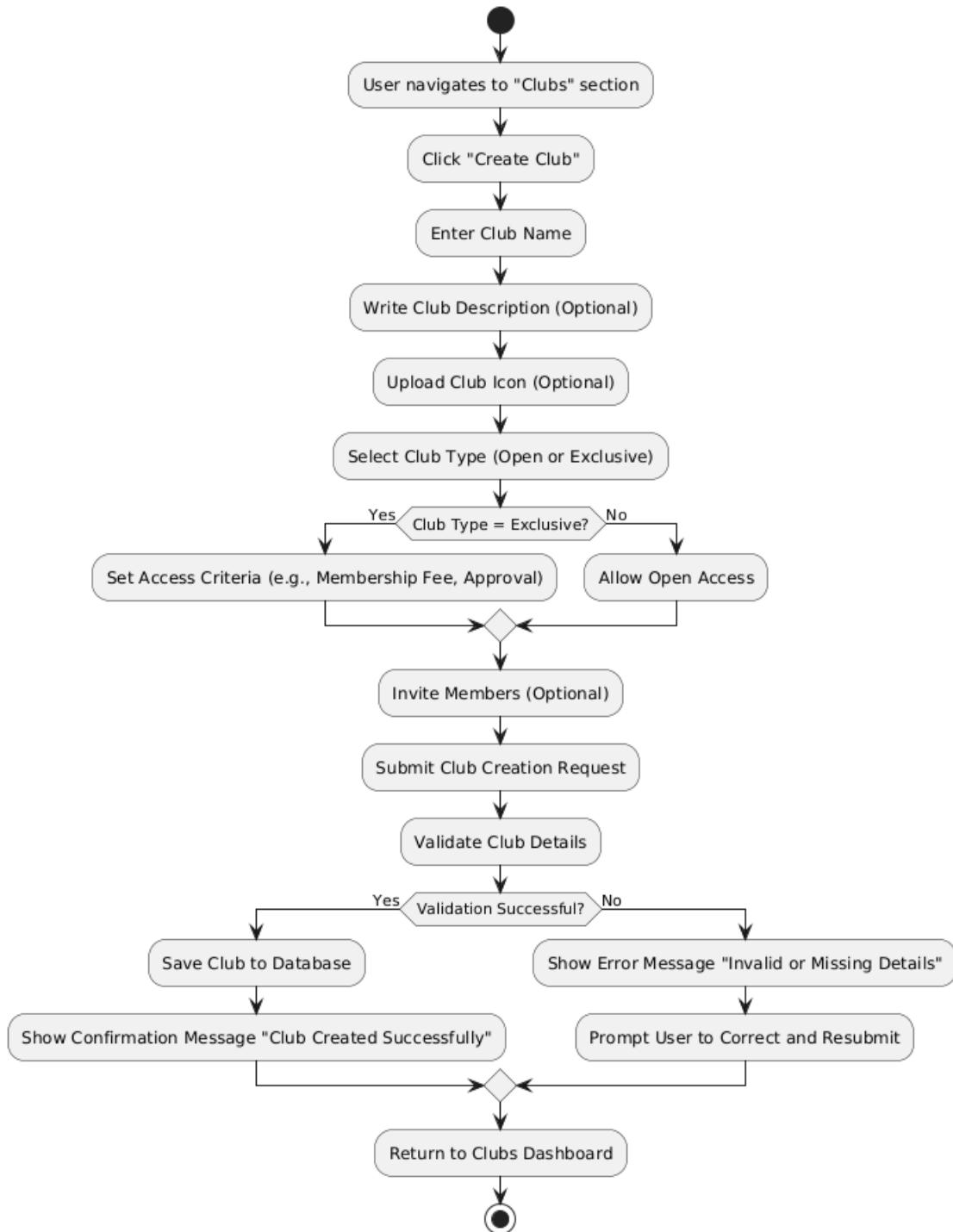


Figure 34: Activity diagram for Create Clubs.

4.1.32 Join Clubs or Groups

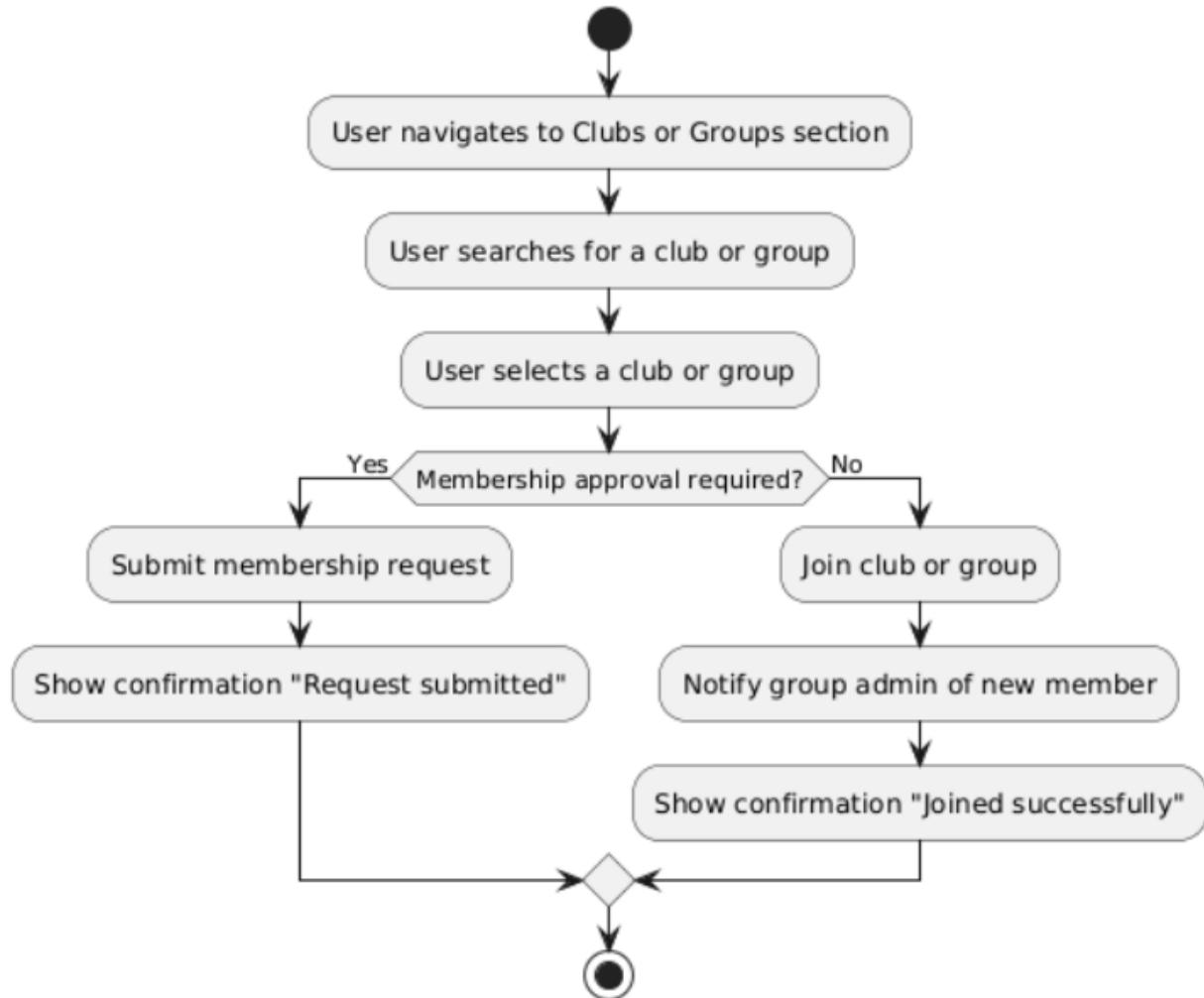


Figure 35: Activity diagram for Join Clubs/Groups.

4.1.33 Share Content

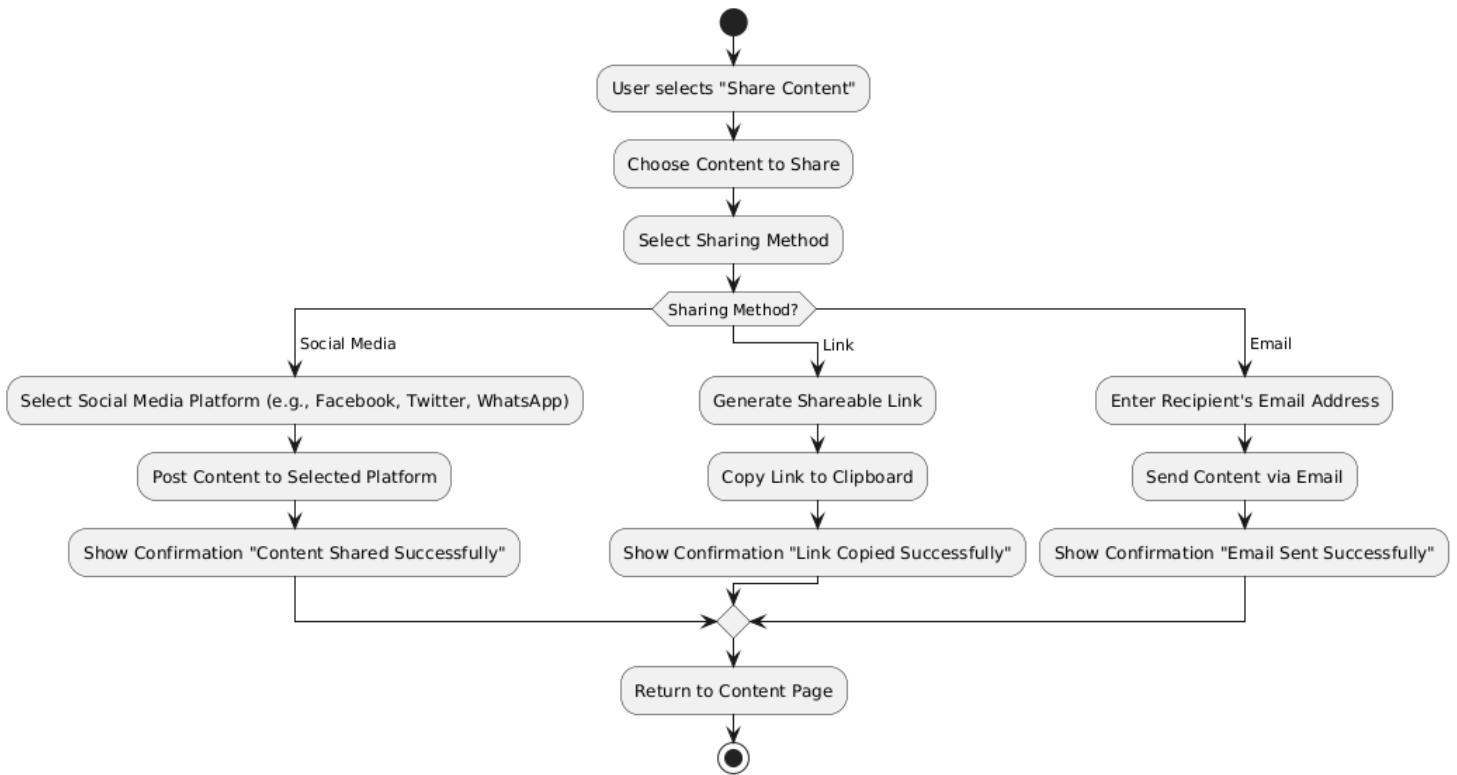


Figure 36: Activity diagram for Share Content.

4.1.34 View Personalized Recommendations

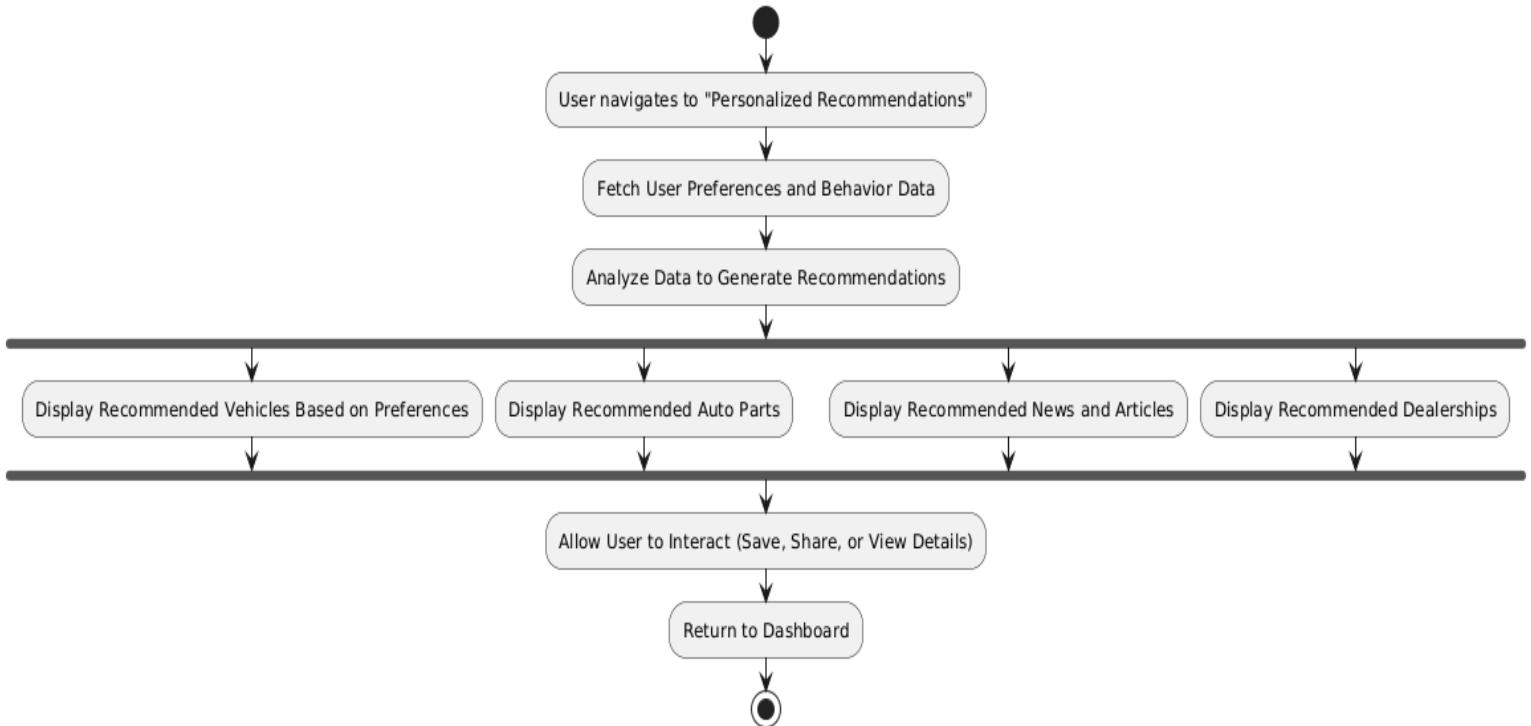


Figure 37: Activity diagram for Personalized Recommendations.

4.1.35 Track Vehicle Health Via OBD-II

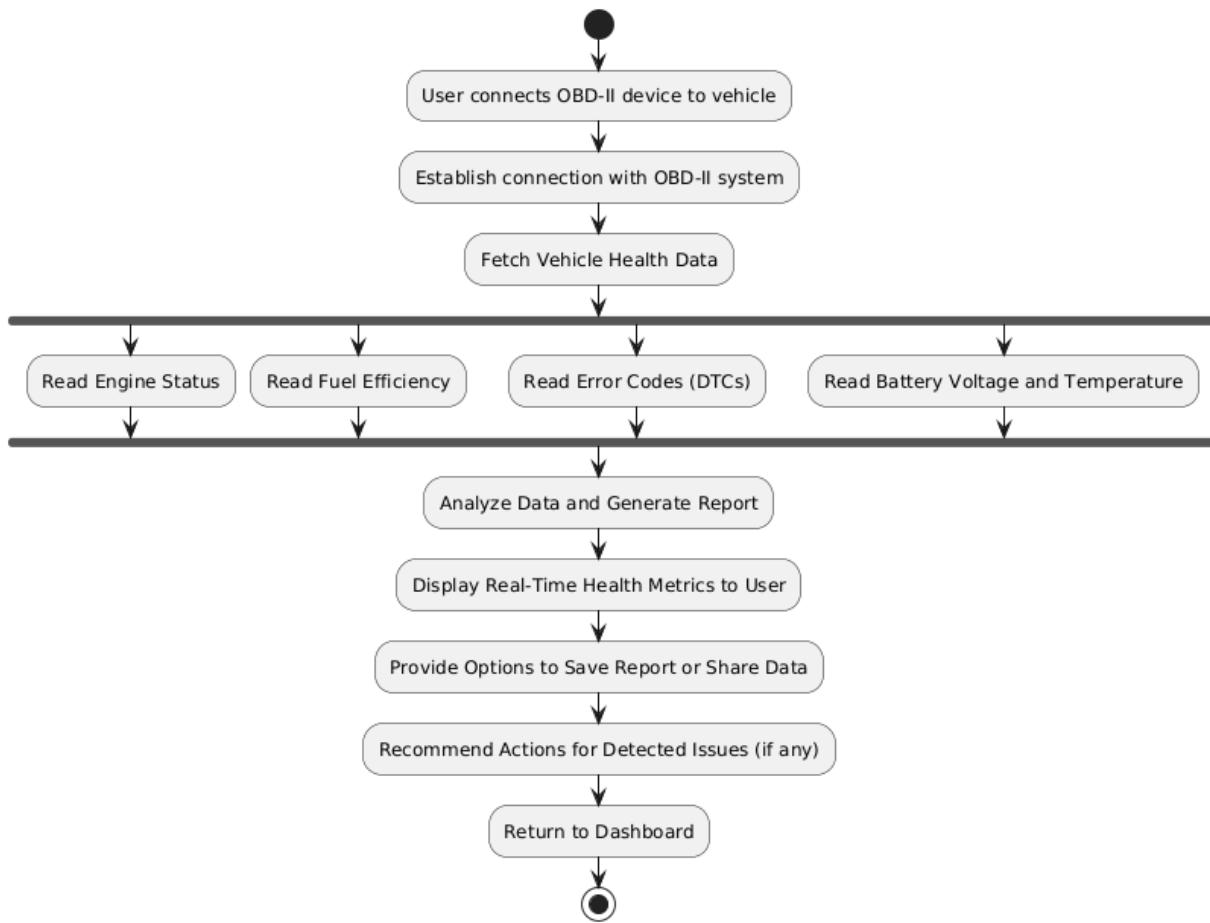


Figure 38: Activity diagram for Track Health Via OBD-II.

4.1.36 Expense Tracking

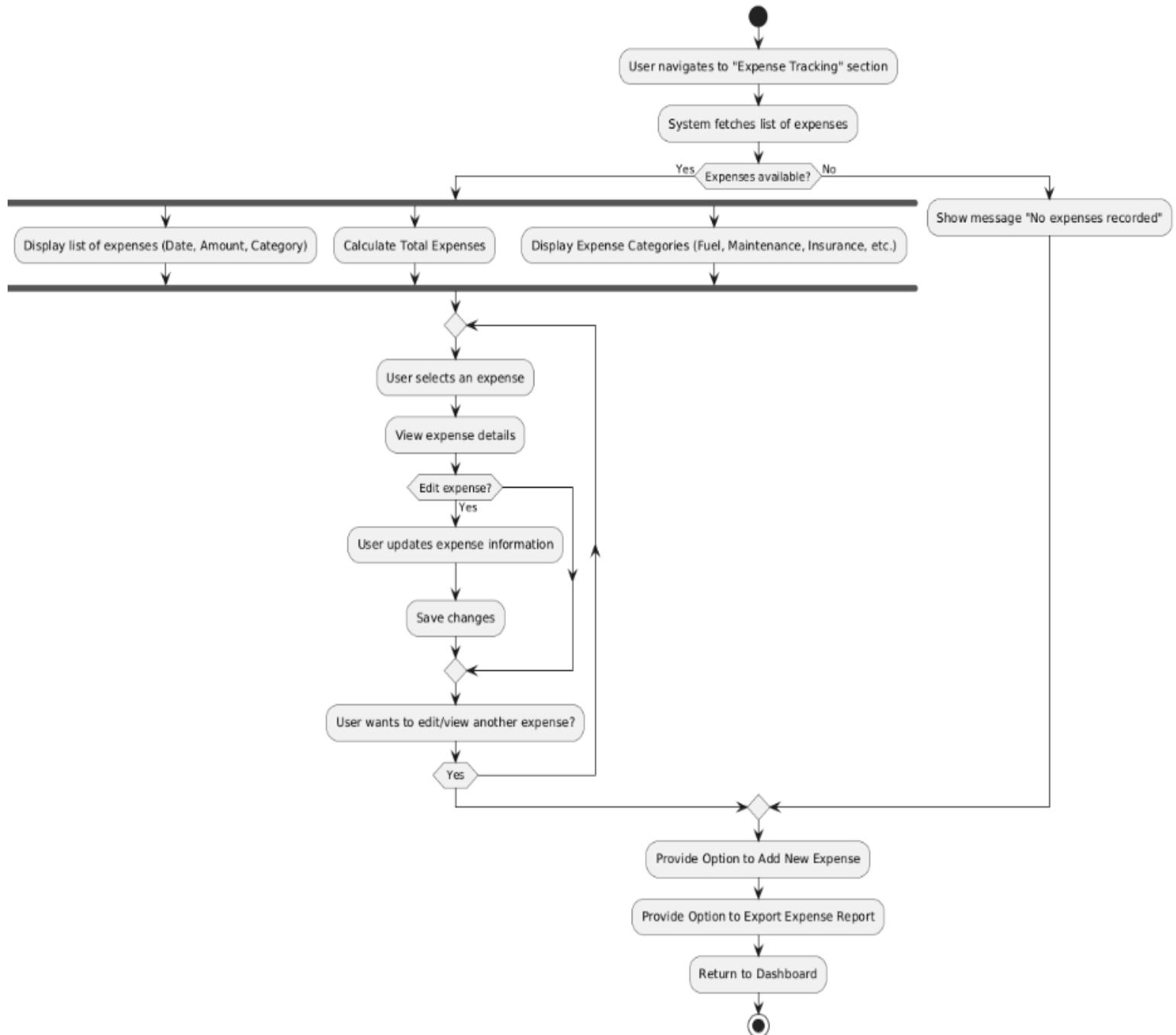


Figure 39: Activity diagram for Expense Tracking.

4.1.37 Generate Reports

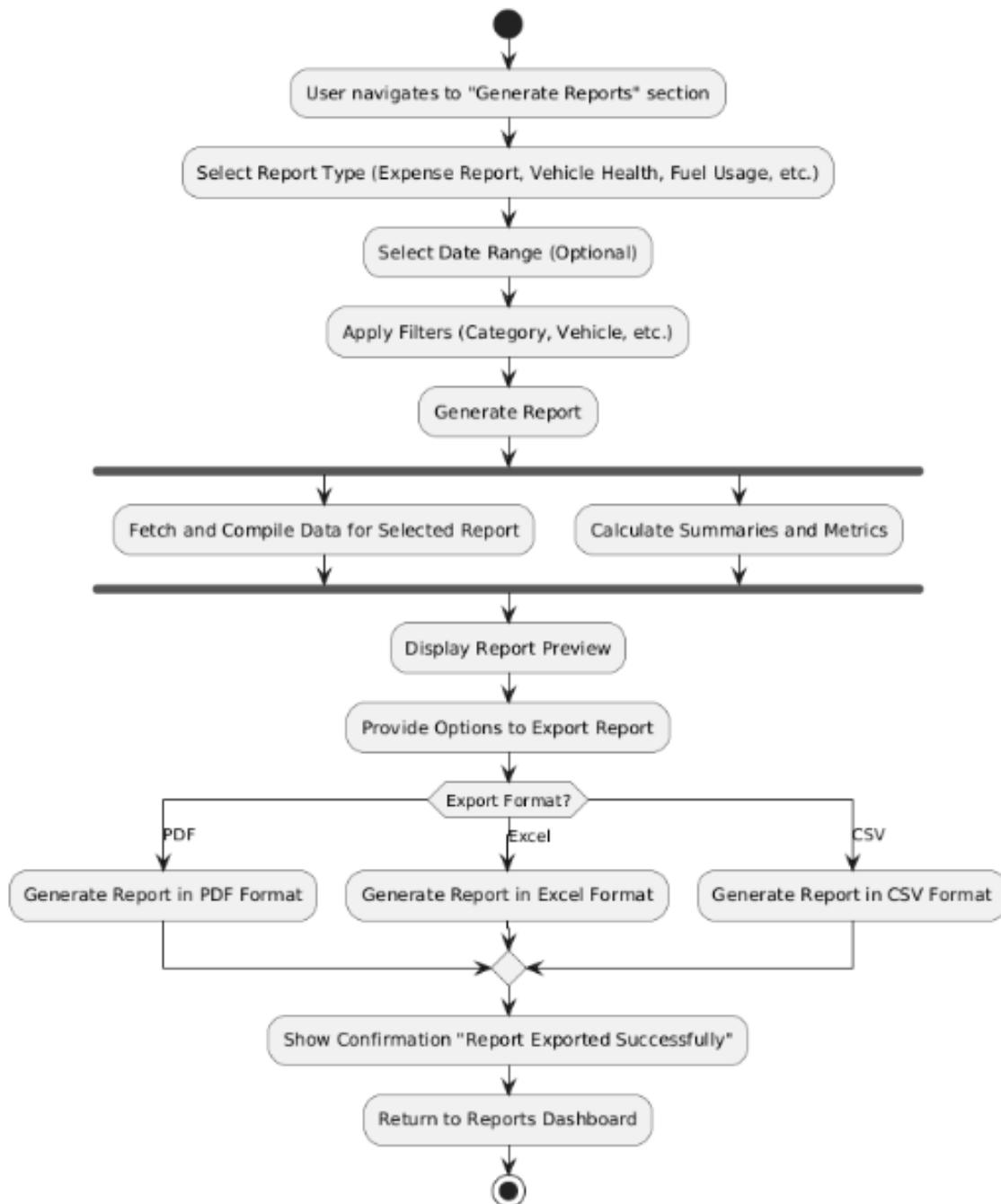


Figure 40: Activity diagram for Generate Reports.

4.1.38 Service Reminders

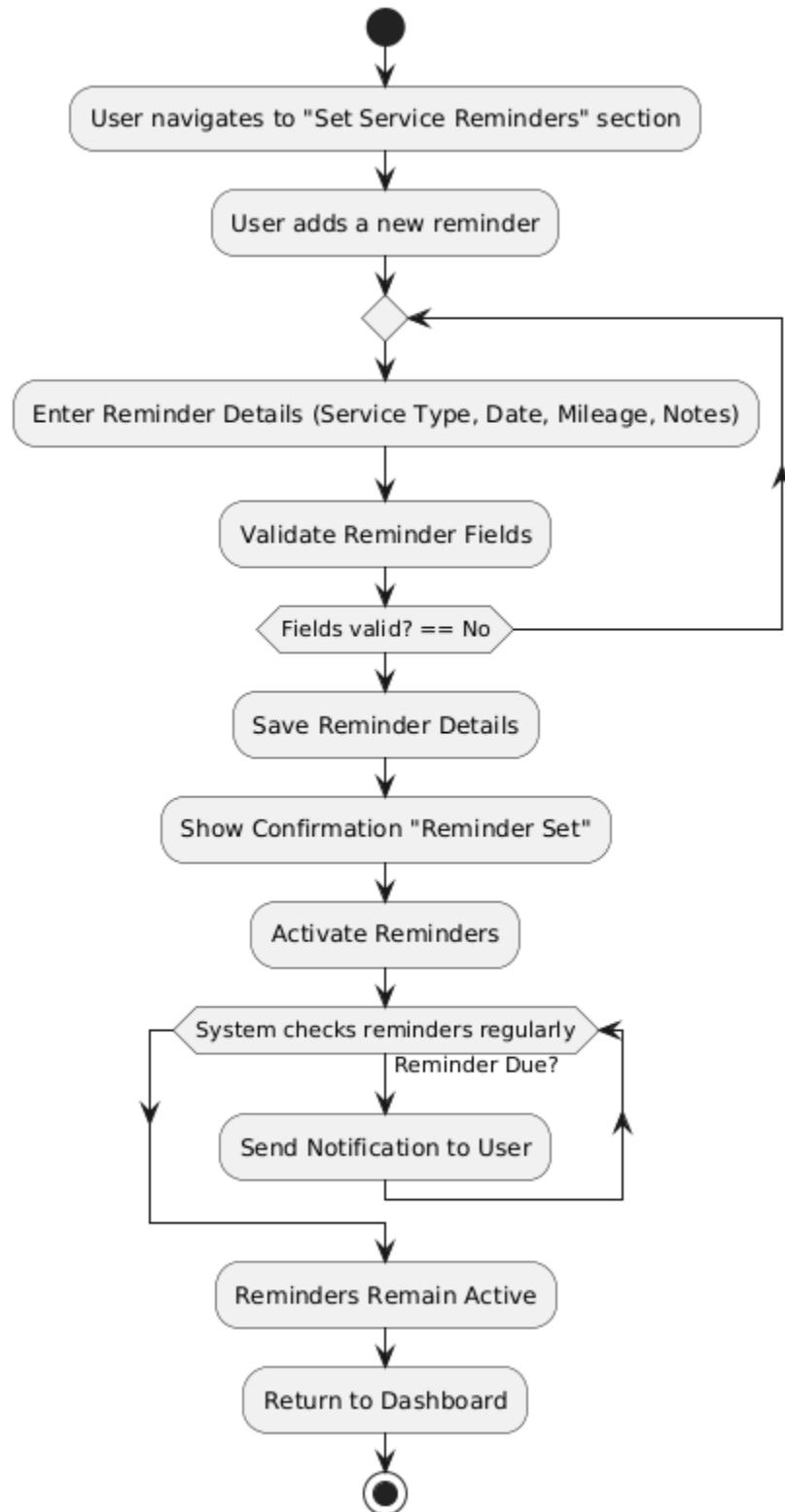


Figure 41: Activity diagram for Set Service Reminders.

4.1.39 AI Chatbot Interaction

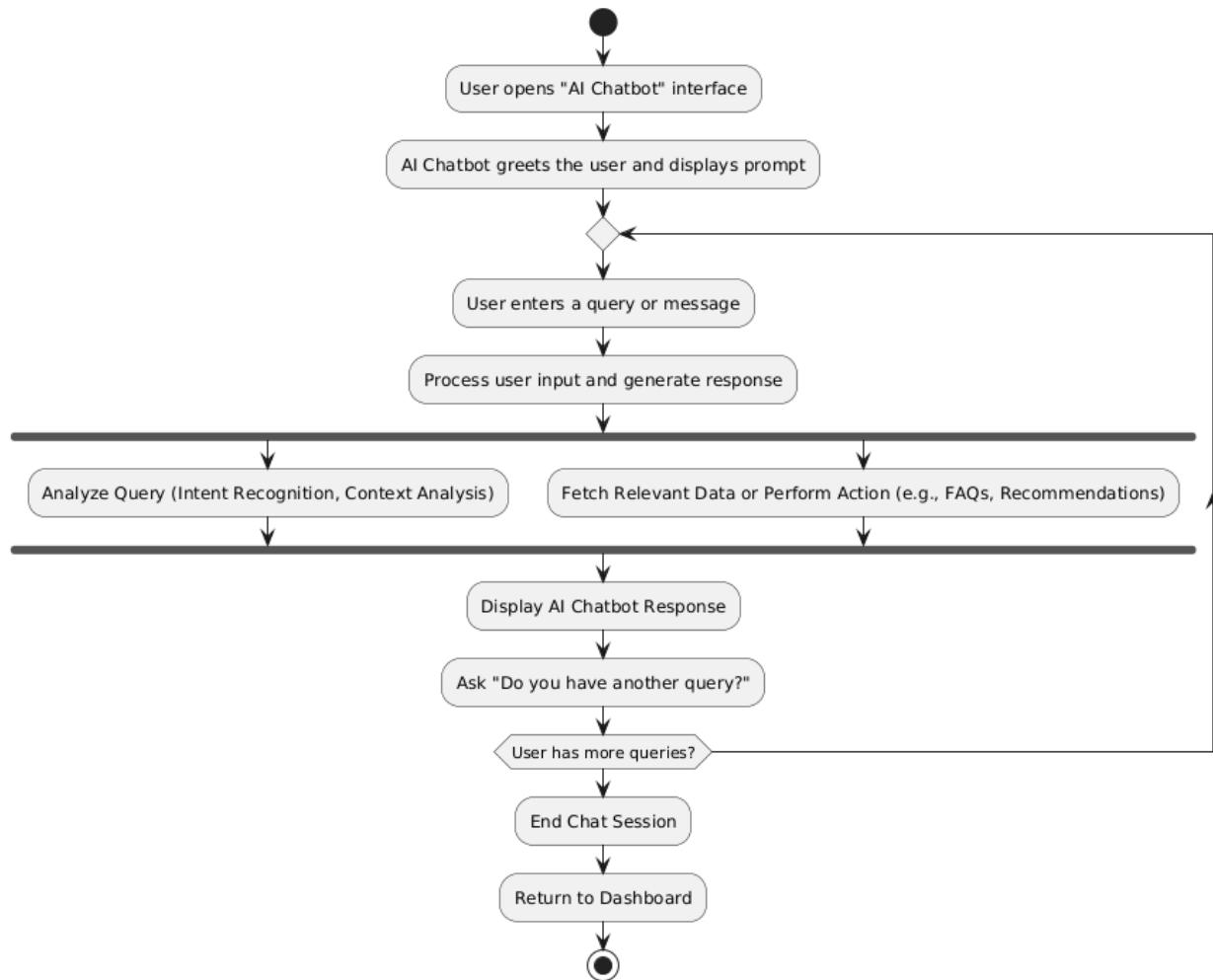


Figure 42: Activity diagram for AI Chatbot Interaction.

4.1.40 Live Support

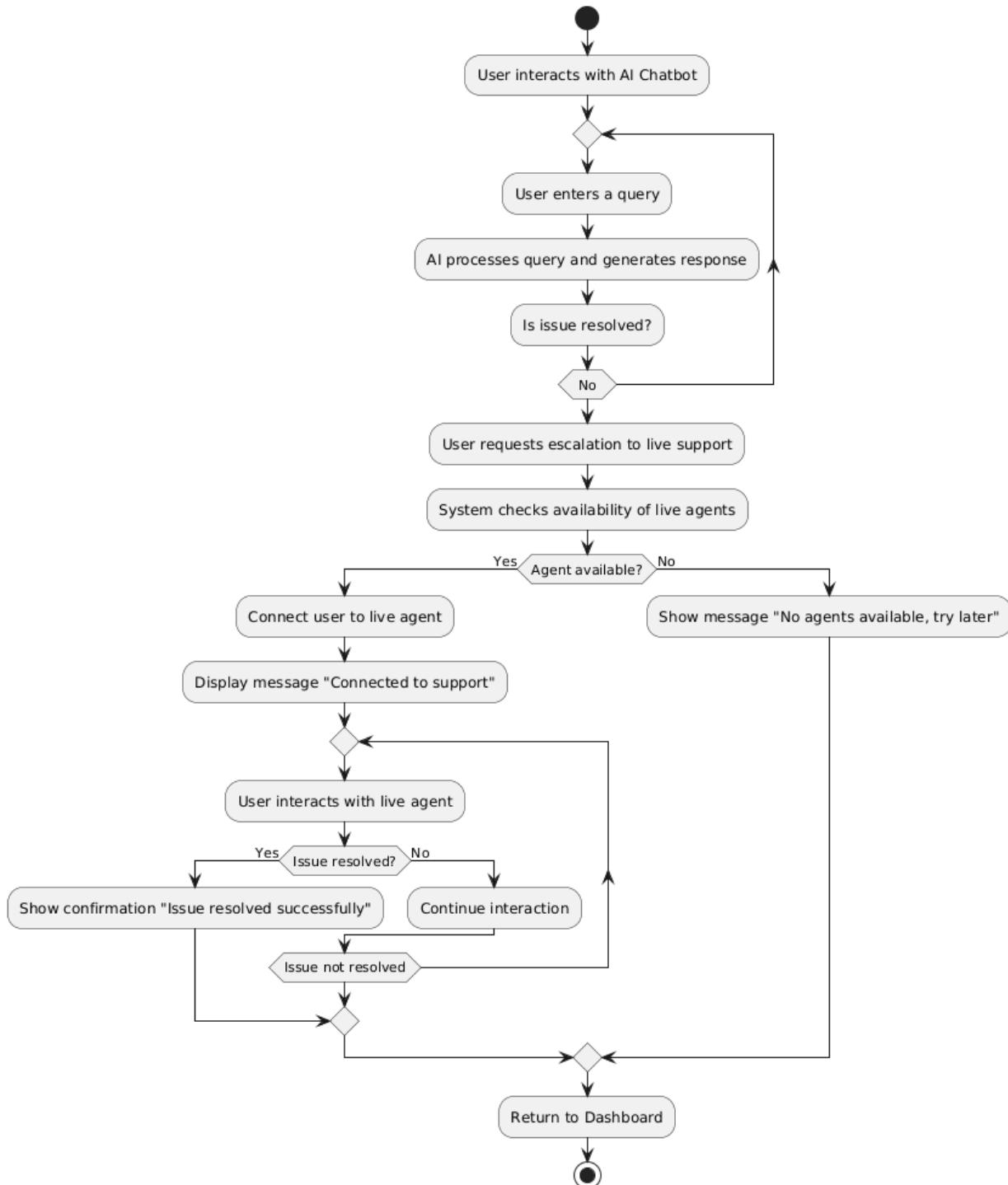


Figure 43: Activity diagram for Live Support.

4.1.41 Auto Services

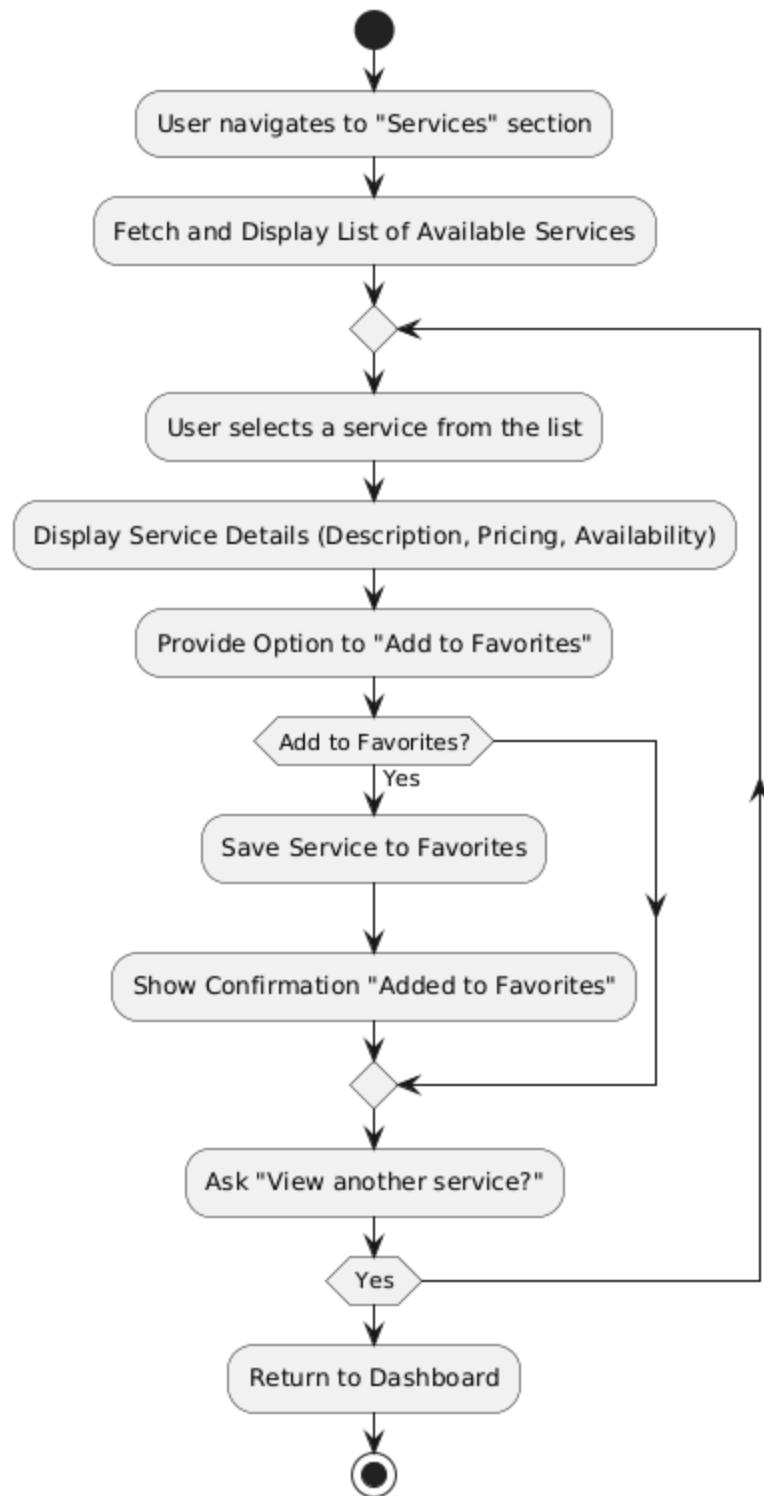


Figure 44: Activity diagram for Auto Services.

4.1.42 Book Auto Services

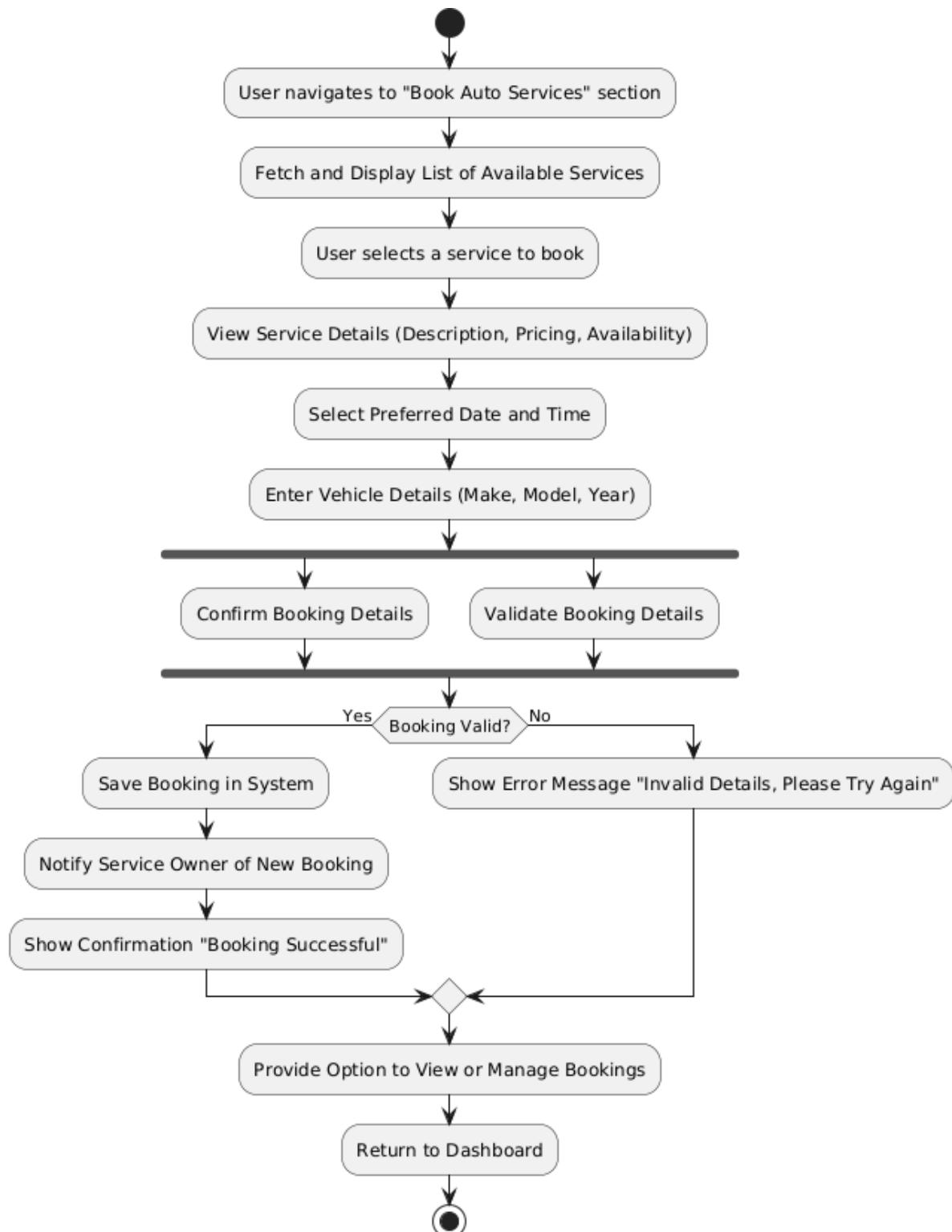


Figure 45: Activity diagram for Book Auto Services.

4.2 Class Diagram

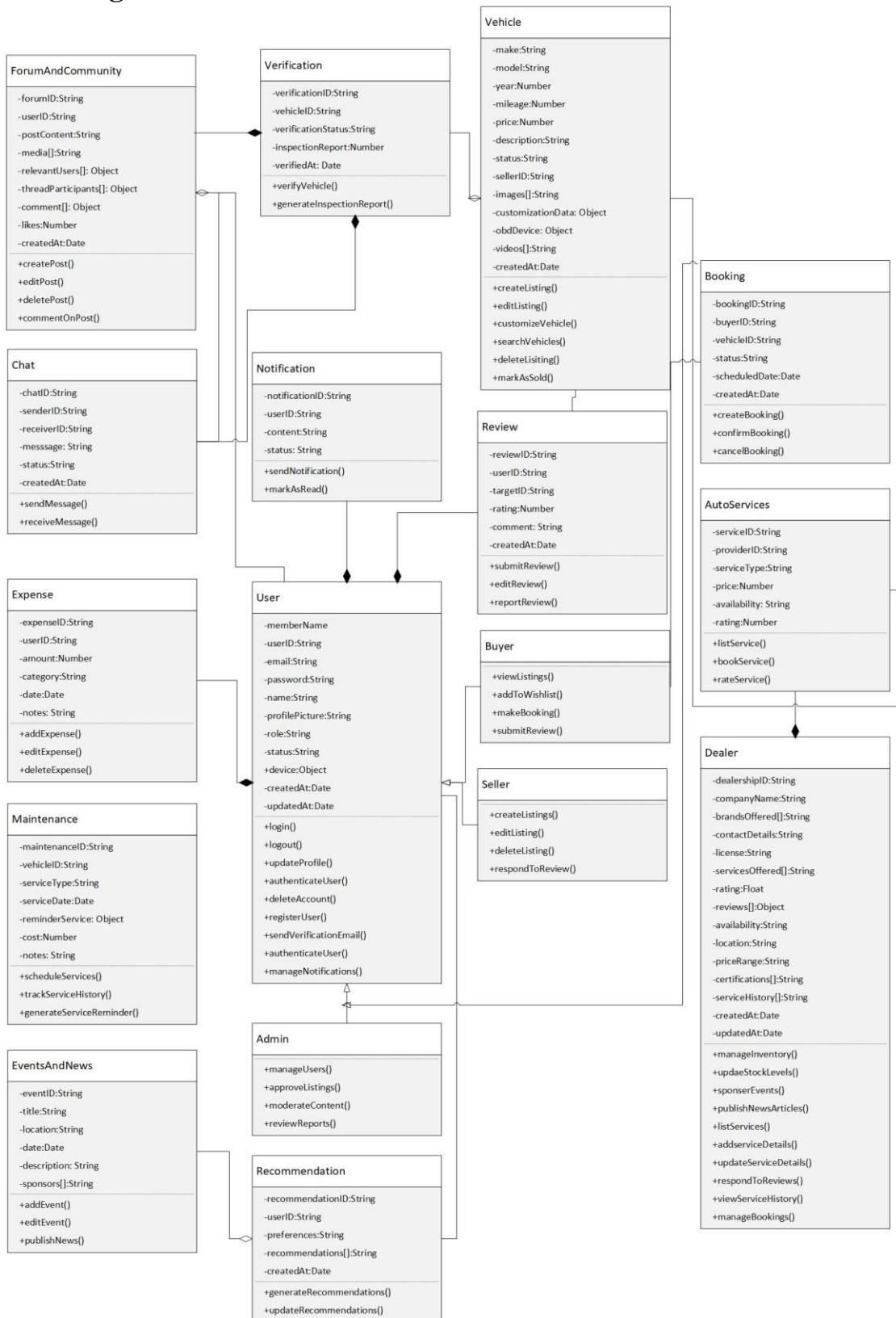


Figure 46: Class diagram for Motors Bay.

4.3 Sequence Diagrams

4.3.1 User Registration Process

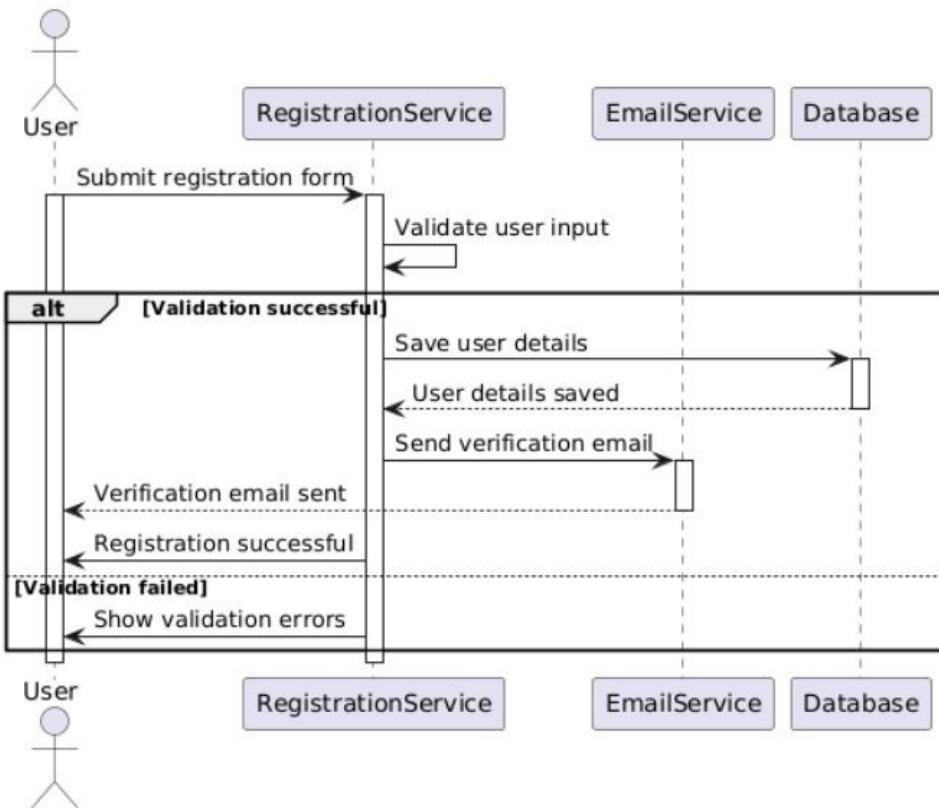


Figure 47: Sequence diagram of User Registration Process.

4.3.2 User Login Authentication

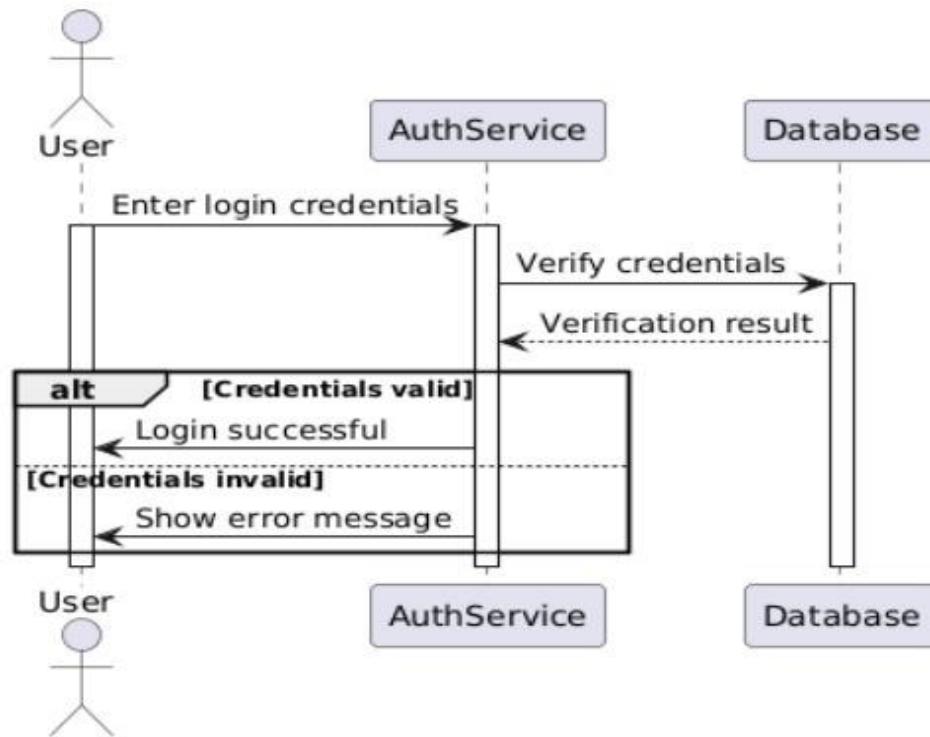


Figure 48: Sequence diagram of User Login Authentication.

4.3.3 Password Recovery Flow

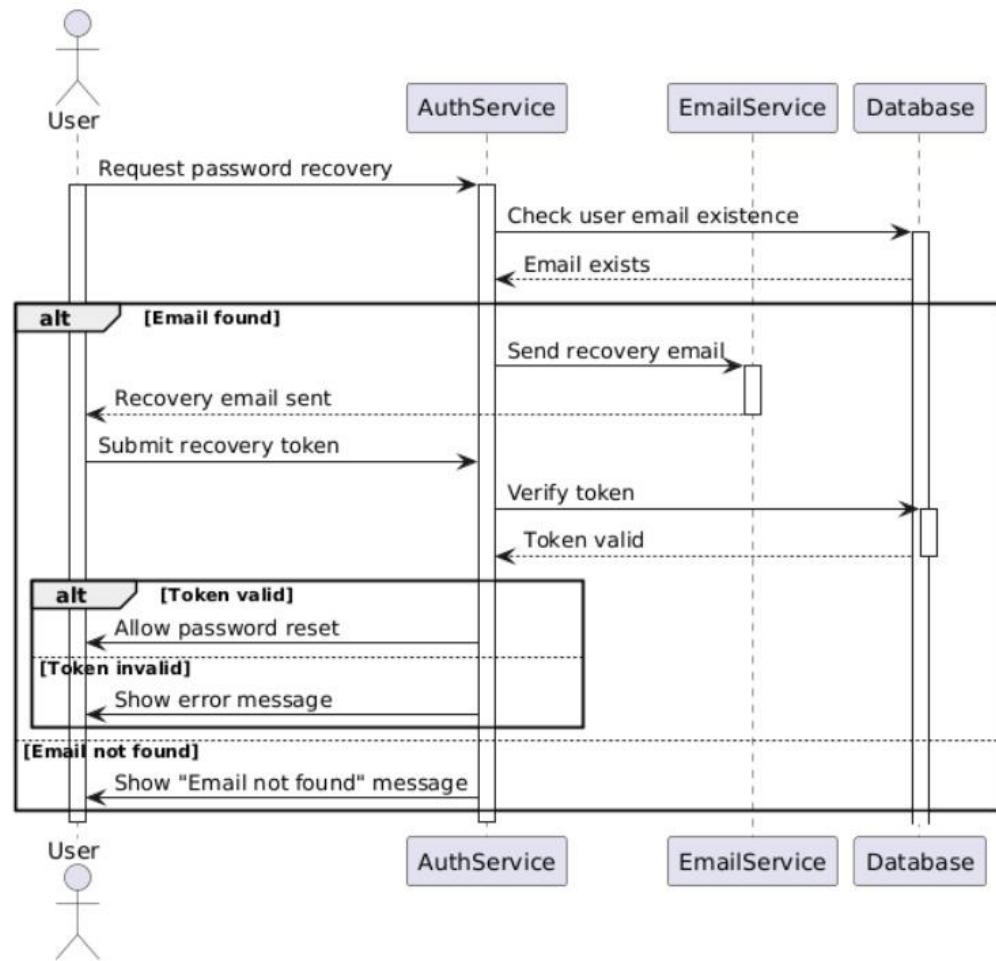


Figure 49: Sequence diagram of Password Recovery Flow.

4.3.4 Search Vehicle Listings

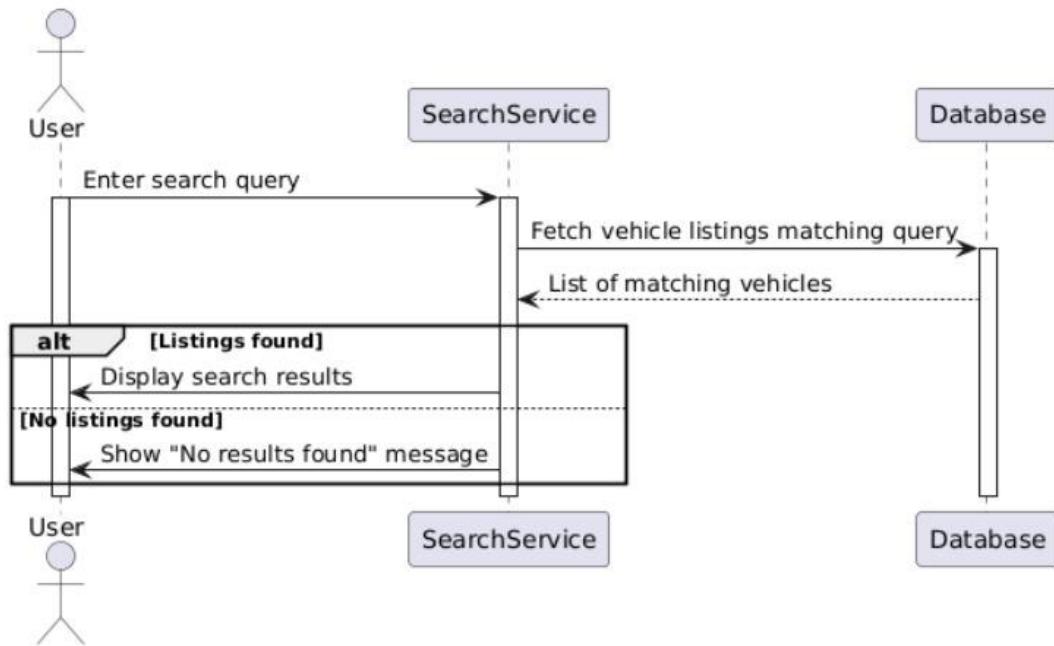


Figure 50: Sequence diagram of Vehicle Listings.

4.3.5 Filter and Sort Results

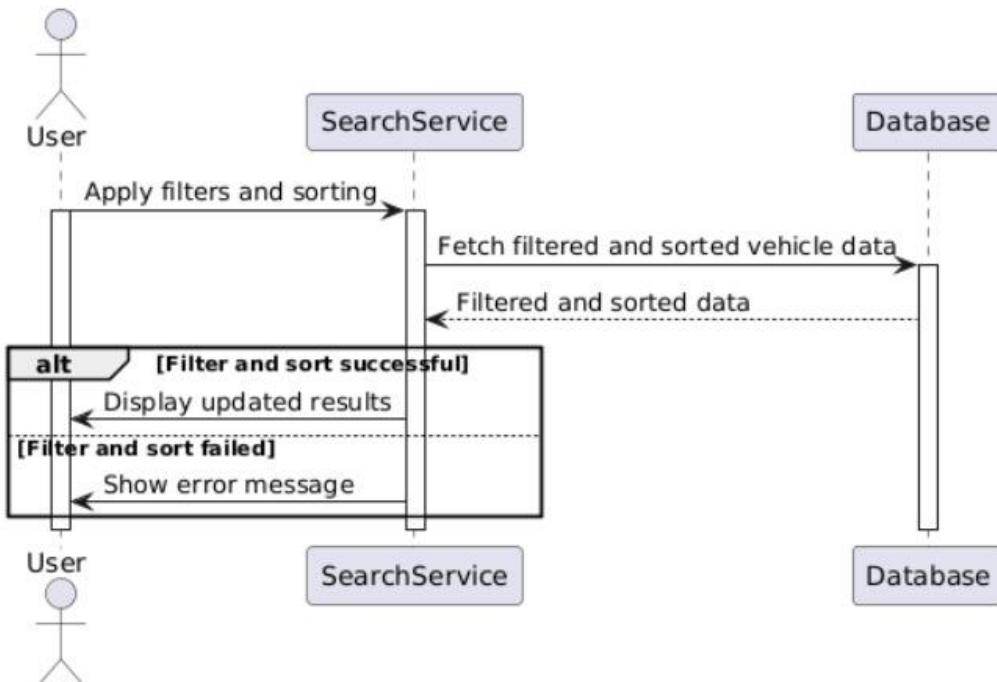


Figure 51: Sequence diagram of Filter & Sort Results.

4.3.6 View Vehicle Details

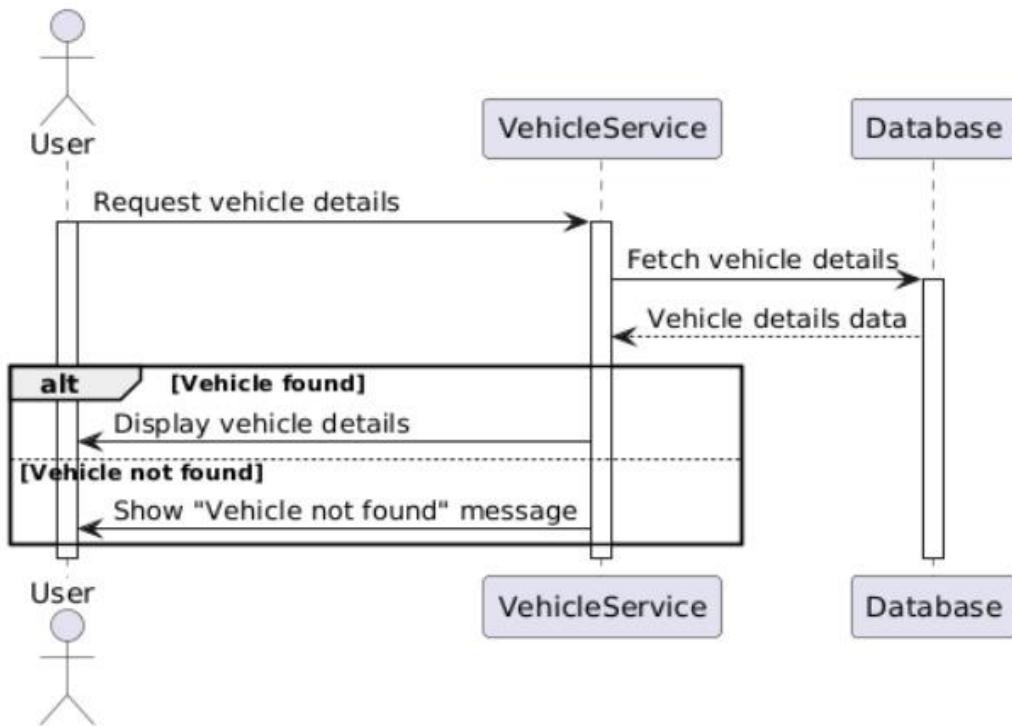


Figure 52: Sequence diagram of View Vehicle Details.

4.3.7 Create Vehicle Listing

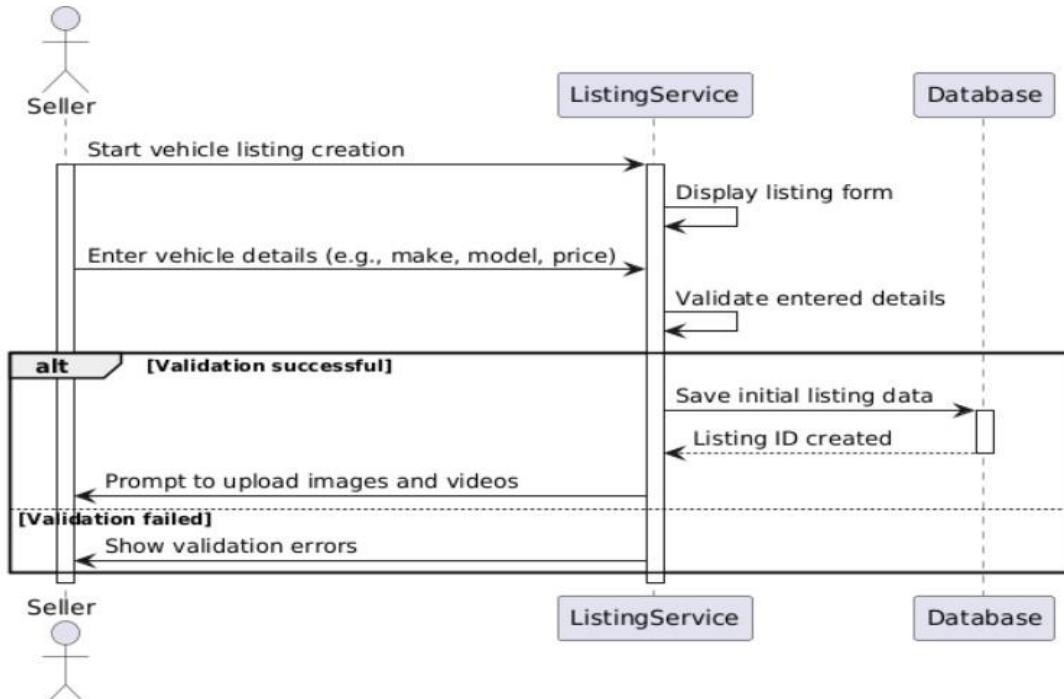


Figure 53: Sequence diagram of Create Vehicle Listing.

4.3.8 Upload Images and Videos

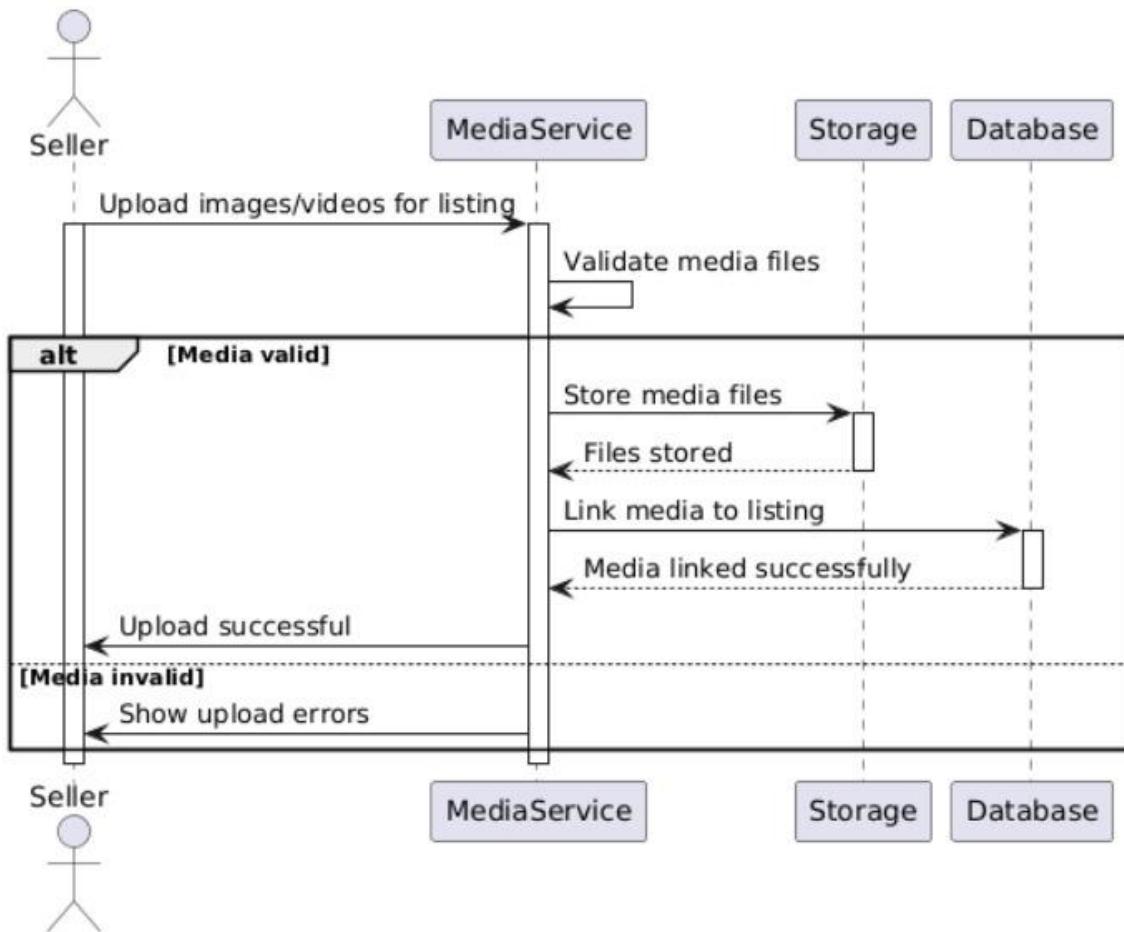


Figure 54: Sequence diagram of Upload Images & Videos.

4.3.9 Save Listing Details

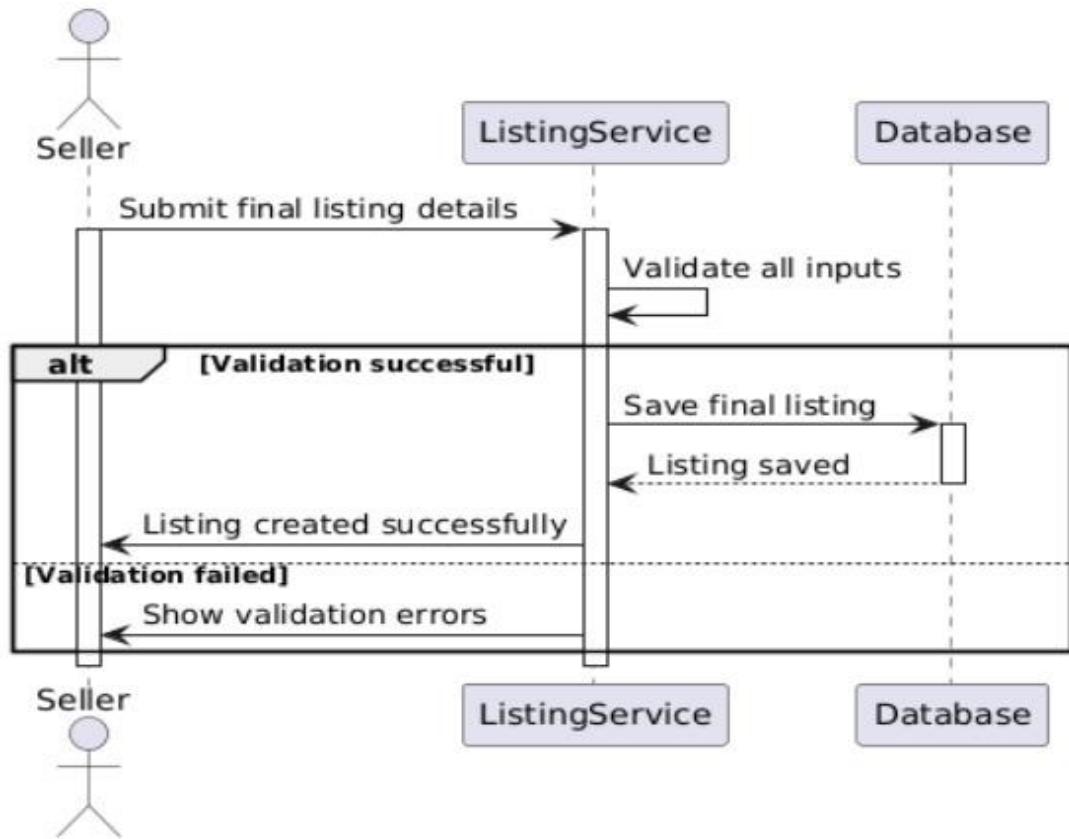


Figure 55: Sequence diagram of Save Listing Details.

4.3.10 Load 3D Model Viewer

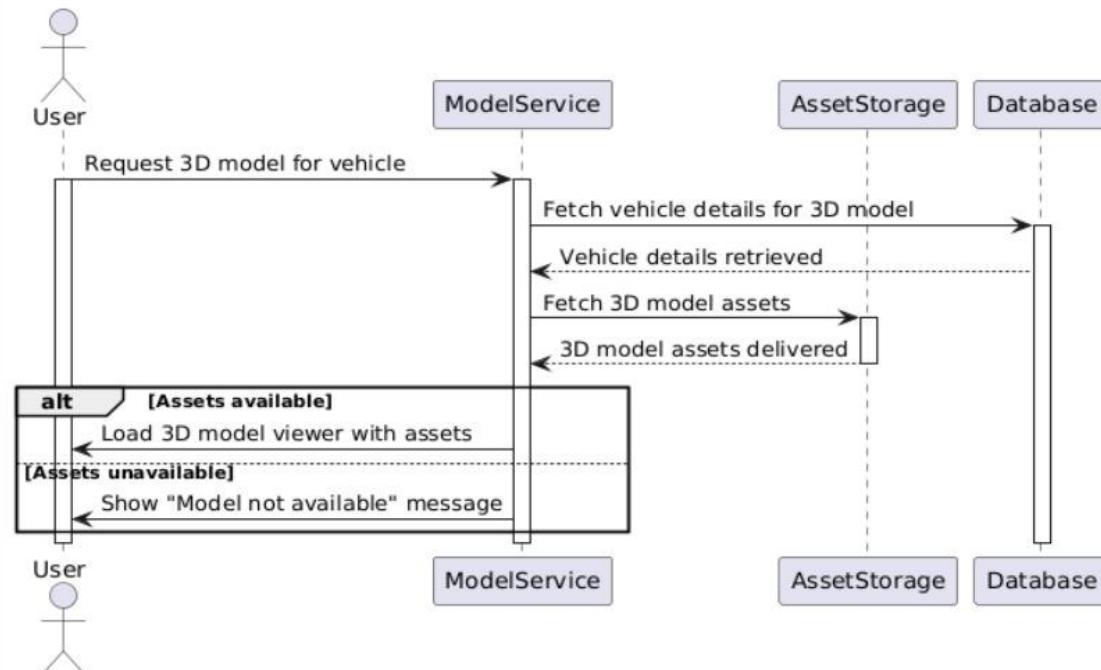


Figure 56: Sequence diagram of Load 3D Model Viewer.

4.3.11 Apply Customization Options

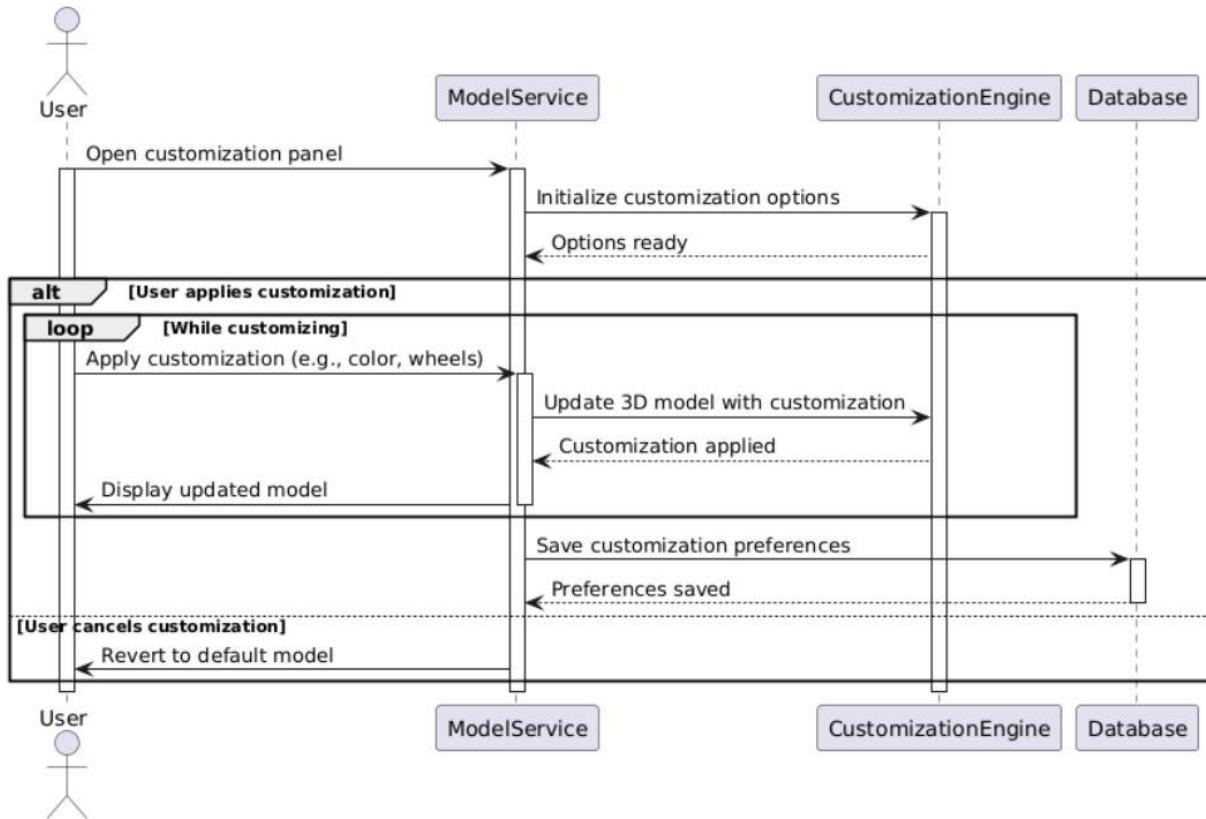


Figure 57: Sequence diagram of Apply Customization Options.

4.3.12 User Initiates Chat with Seller

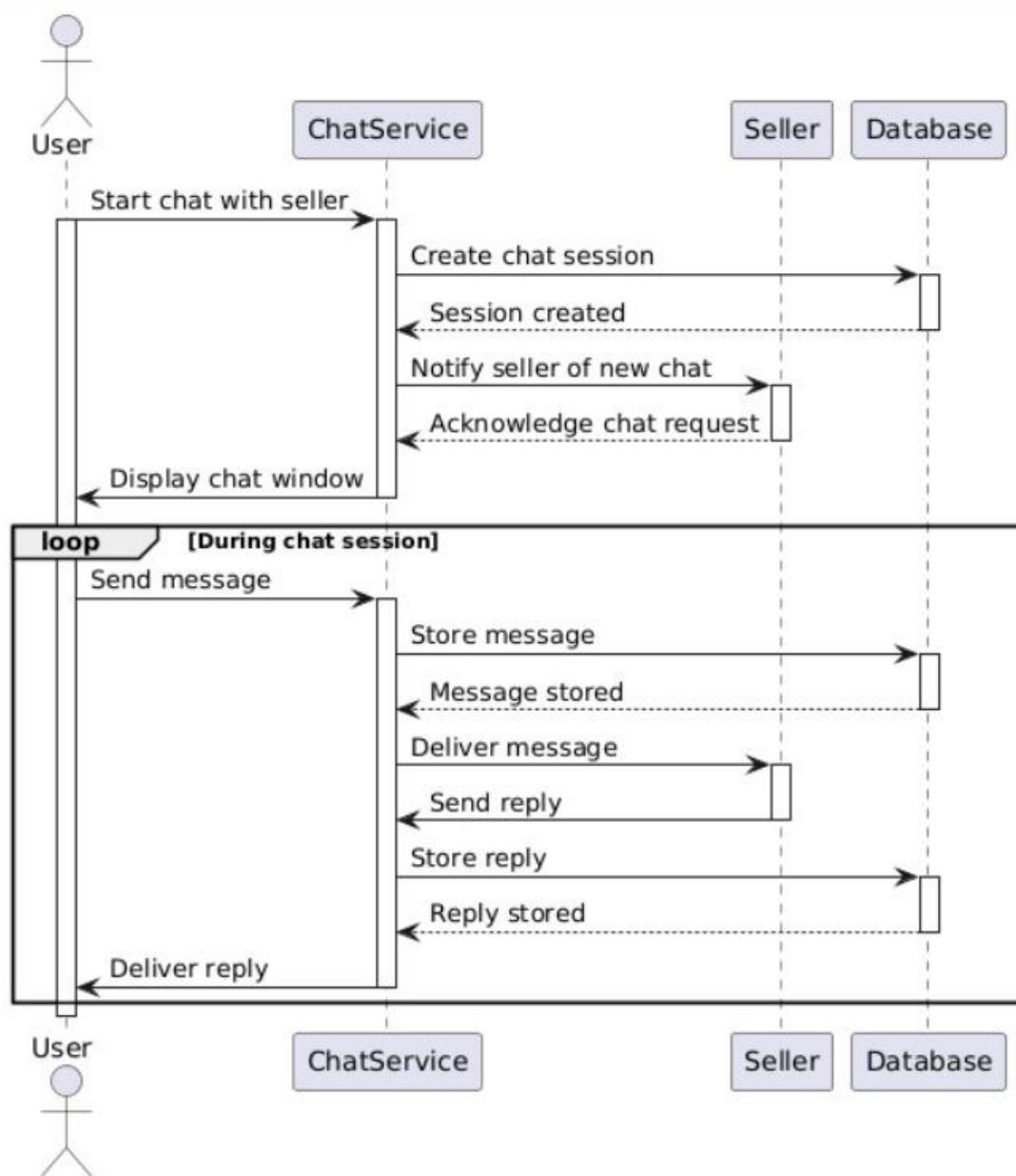


Figure 58: Sequence diagram of User Initiates Chat with Seller.

4.3.13 Send and Receive Notifications

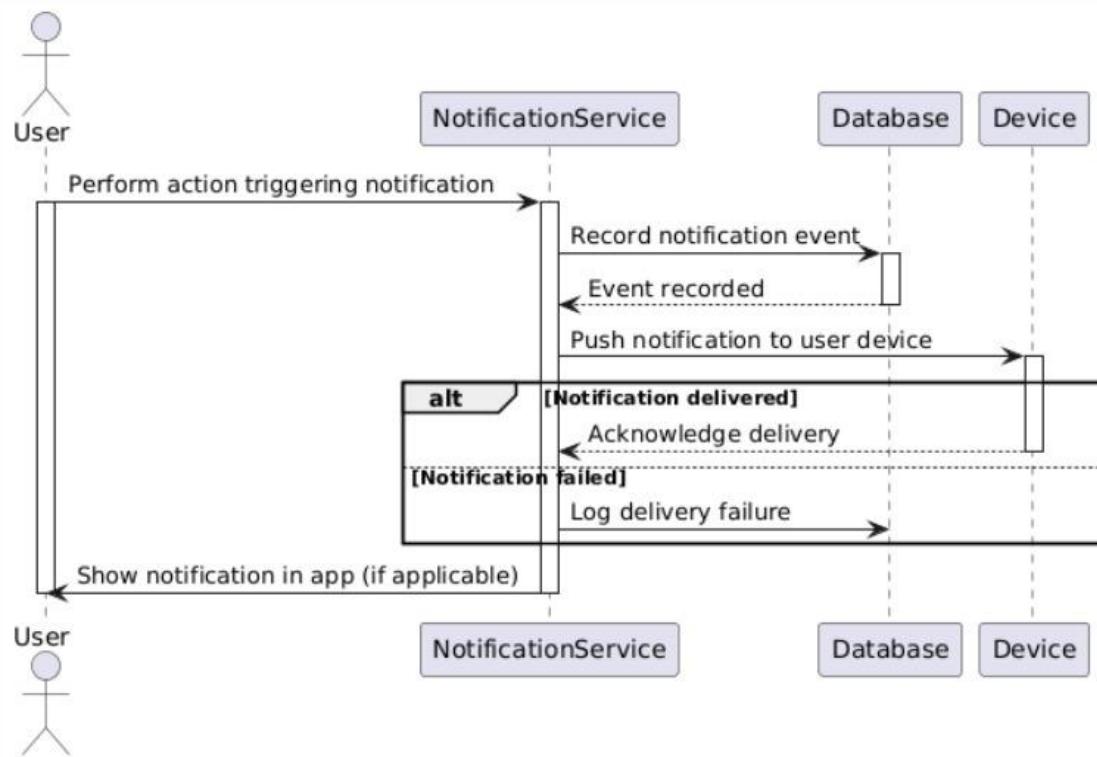


Figure 59: Sequence diagram of Send & Receive Notifications.

4.3.14 Escalate to Live Support

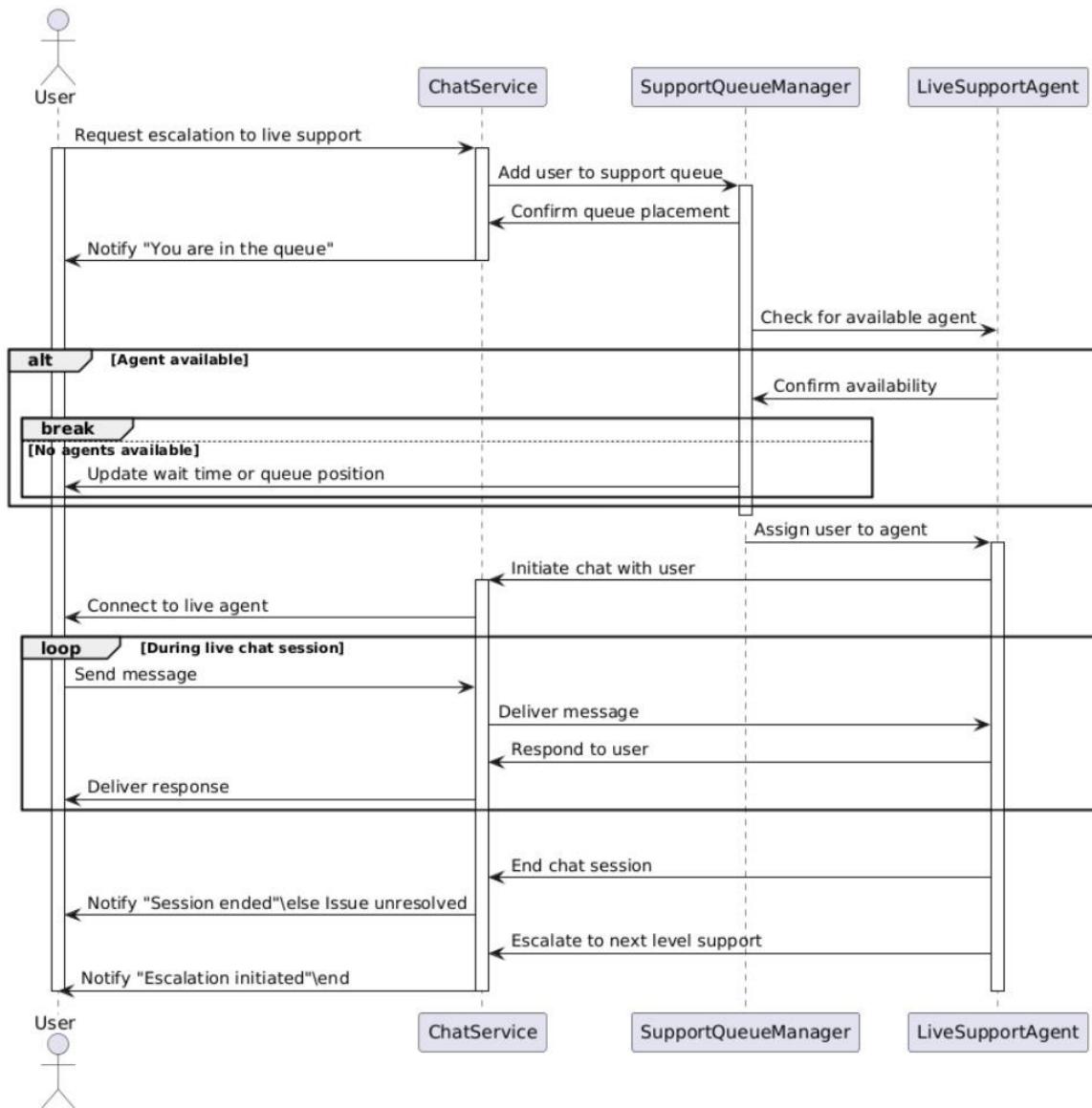


Figure 60: Sequence diagram of Escalate To Live Support.

4.3.15 Submit a Review

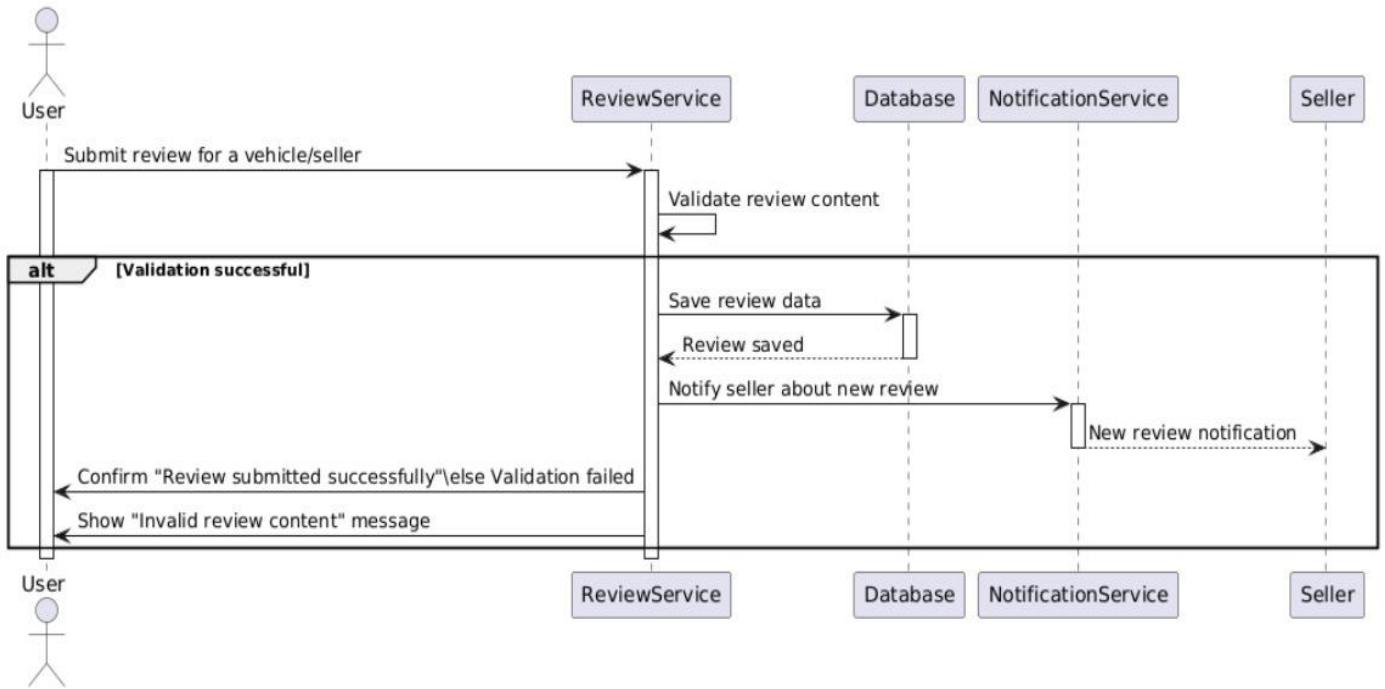


Figure 61: Sequence diagram of Submit A Review.

4.3.16 View Ratings for a Vehicle/Seller

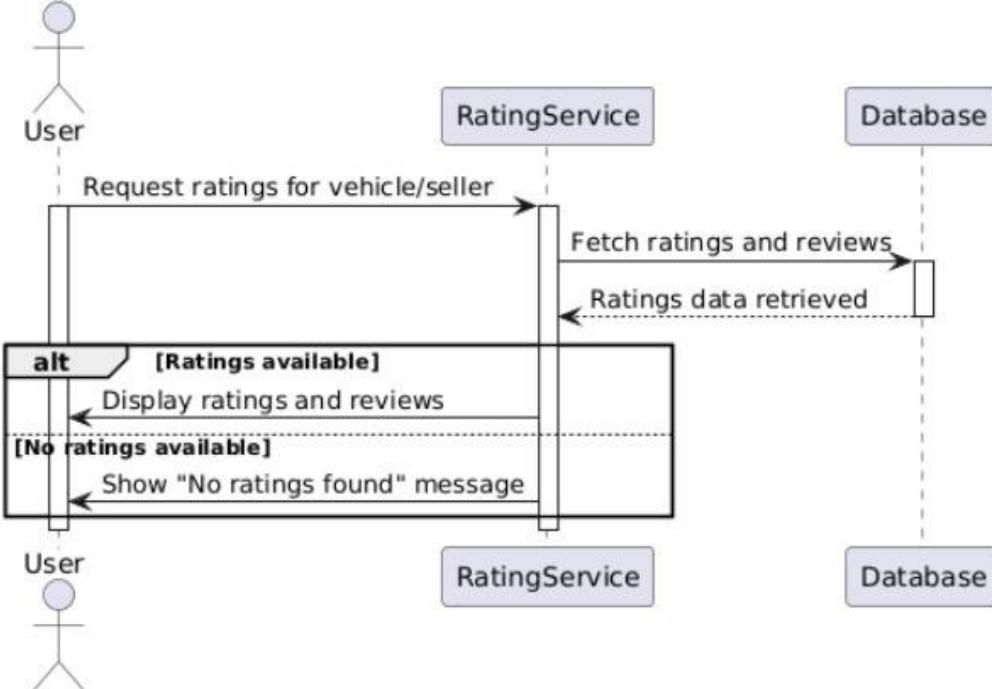


Figure 62: Sequence diagram of View Ratings for A Seller/Vehicle.

4.3.17 Submit Verification Request

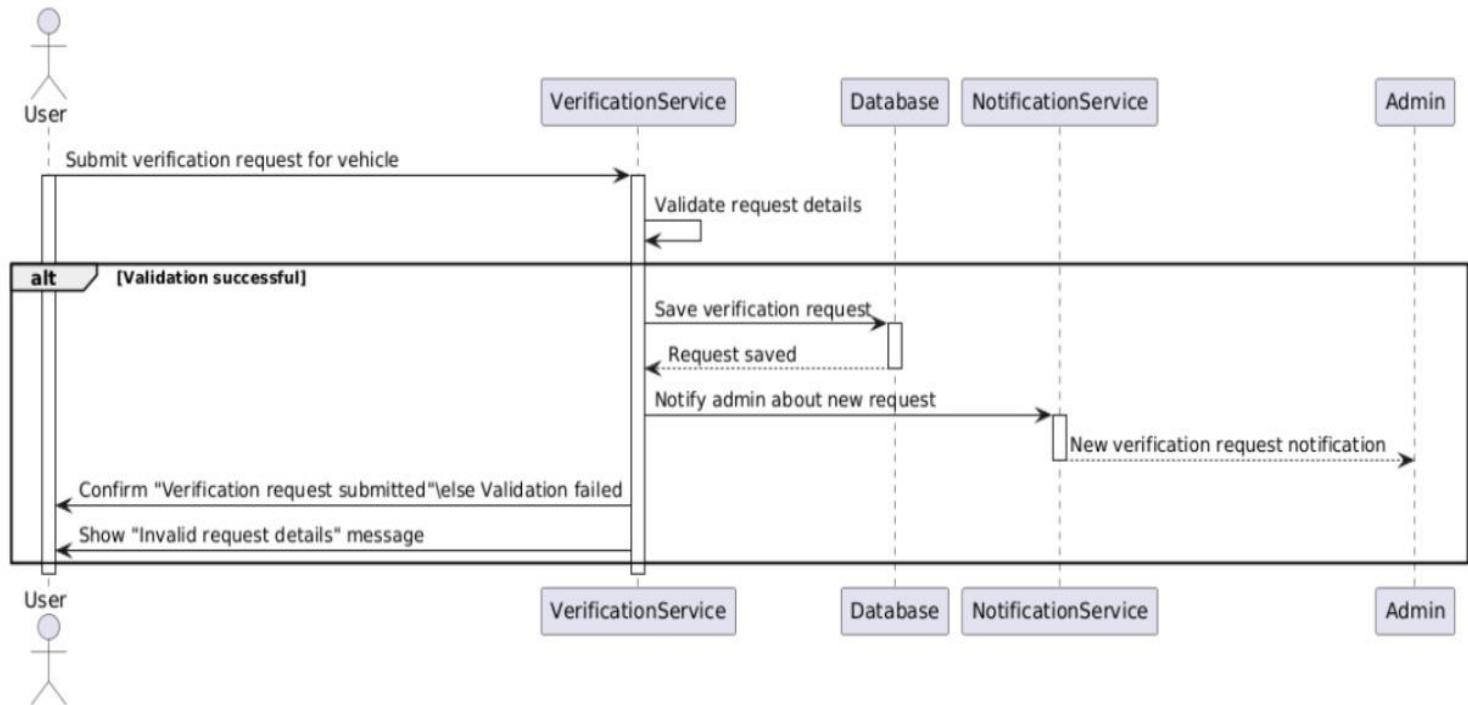


Figure 63: Sequence diagram of Submit Verification Request.

4.3.18 Process Verification by Admin

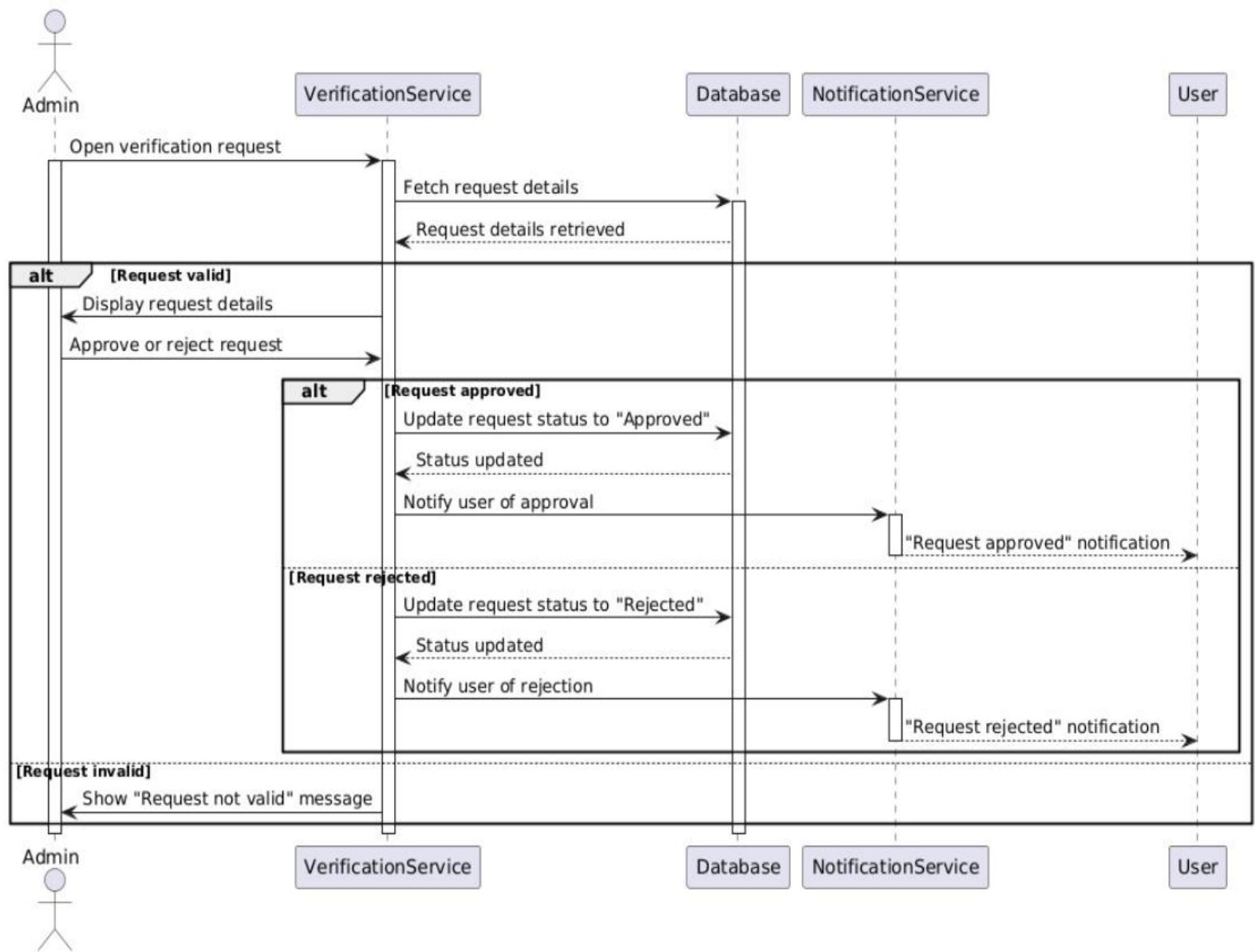


Figure 64: Sequence diagram of Process Verification by Admin.

4.3.19 View List of Users

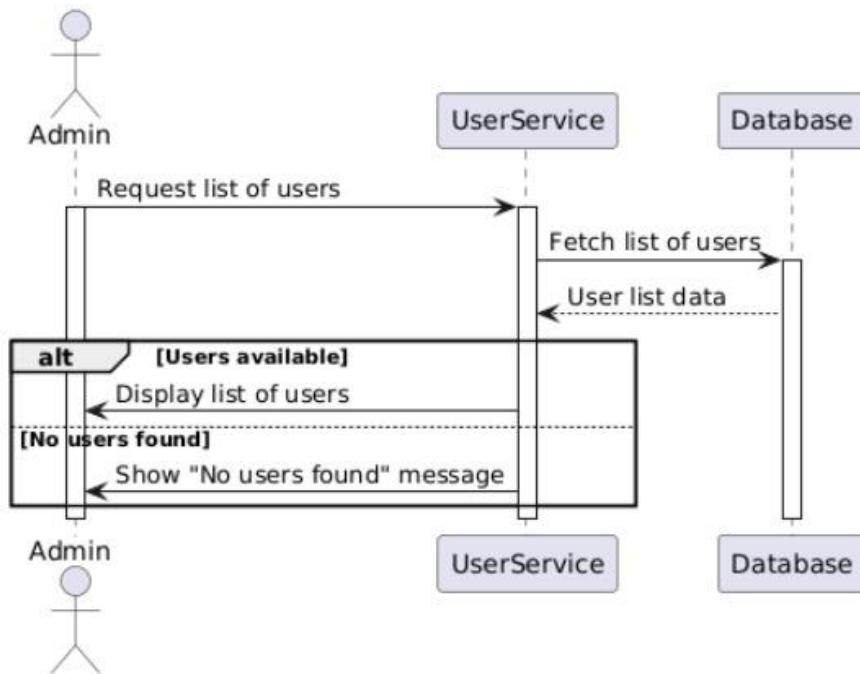


Figure 65: Sequence diagram of View List of Users.

4.3.20 Edit User Account Details

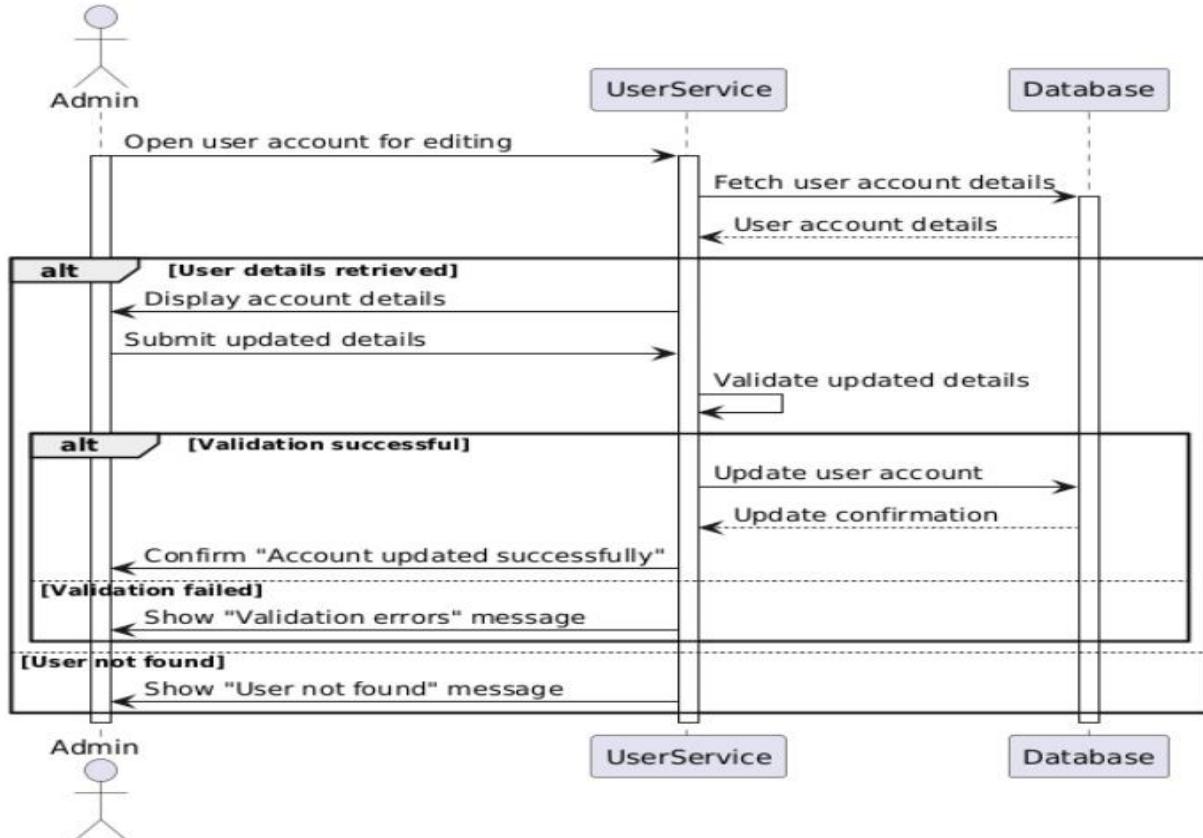


Figure 66: Sequence diagram of Edit User Account Details.

4.3.21 Delete User Account

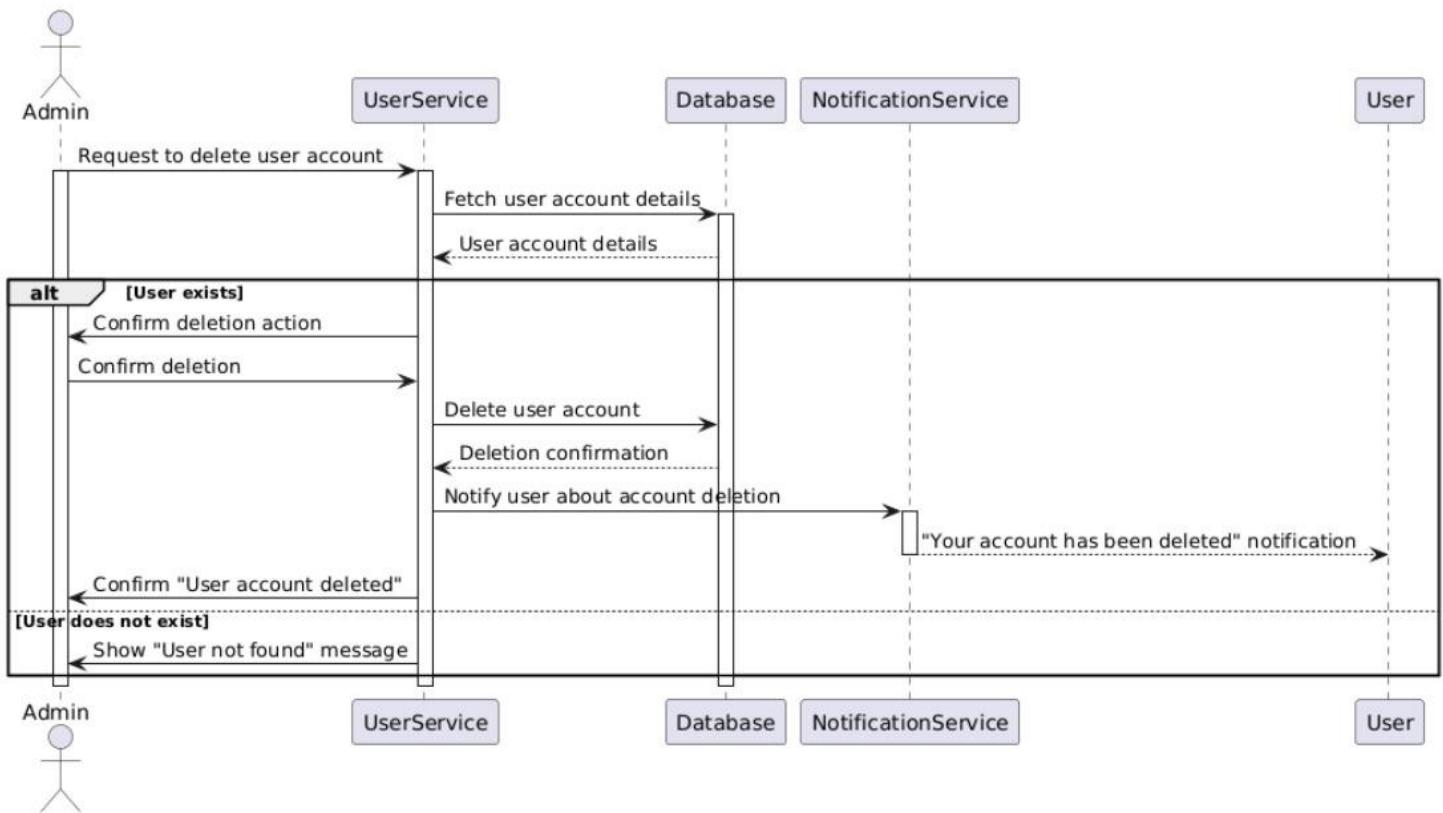


Figure 67: Sequence diagram of Delete User Account.

4.3.22 Set Service Reminders

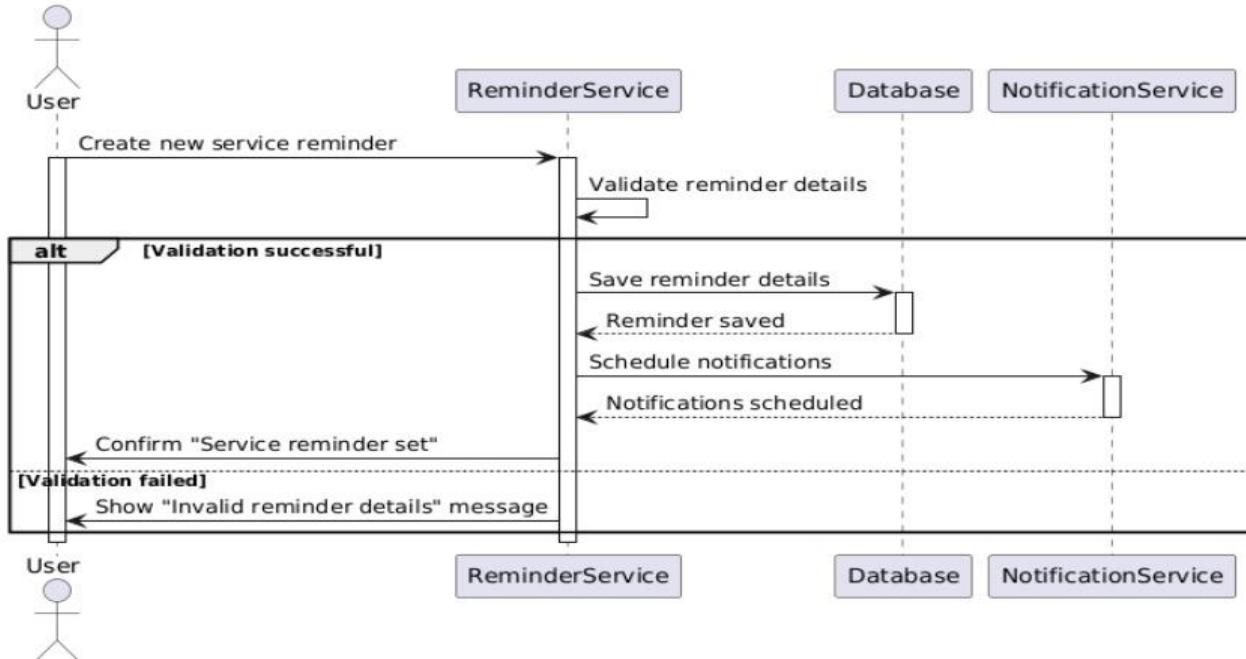


Figure 68: Sequence diagram of Set Service Reminders.

4.3.23 View Maintenance History

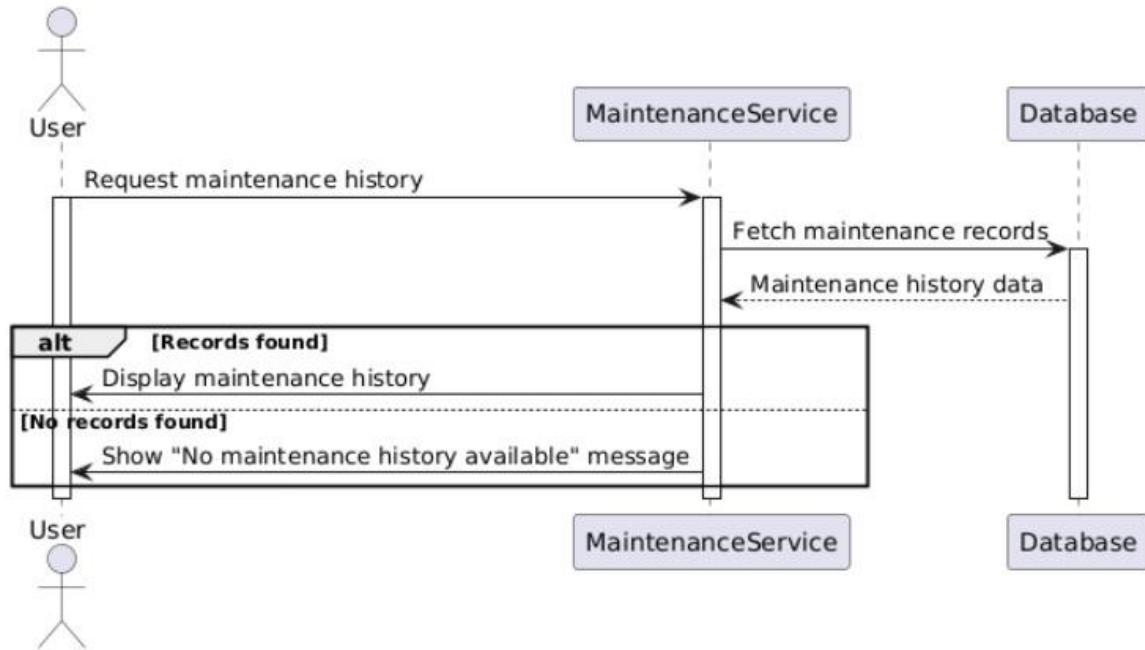


Figure 69: Sequence diagram of view Maintenance History.

4.3.24 Receive Notifications for Service Due

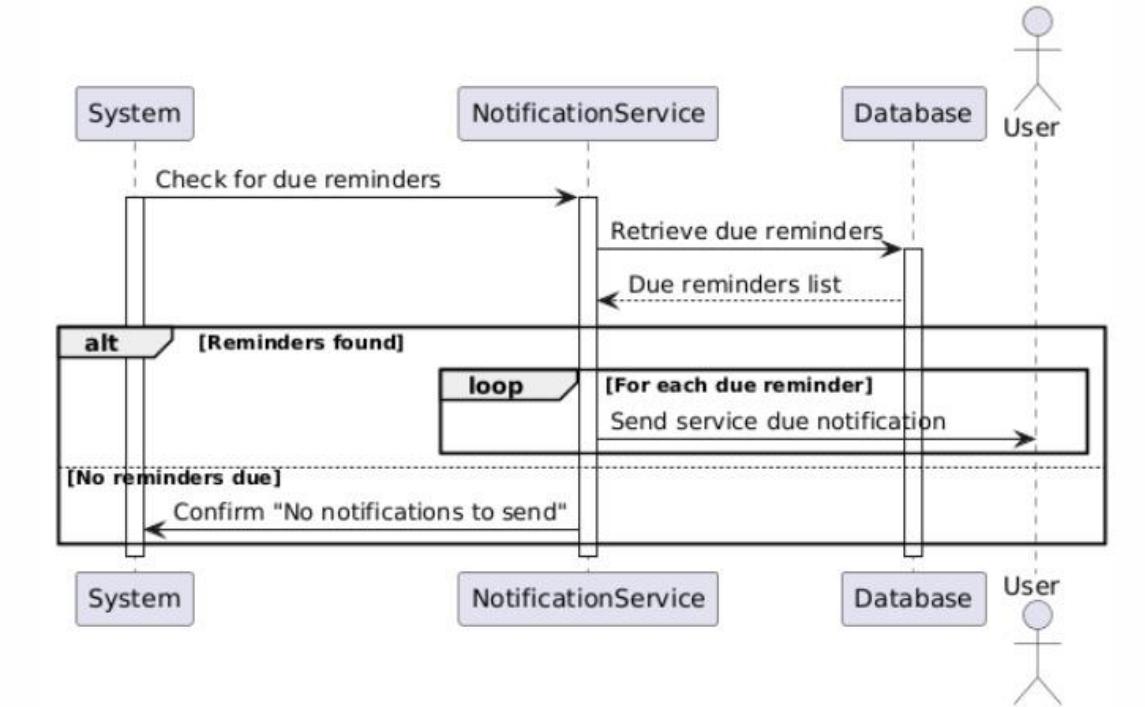


Figure 70: Sequence diagram of Receive Notifications for Service Due.

4.3.25 Fetch Personalized Recommendations

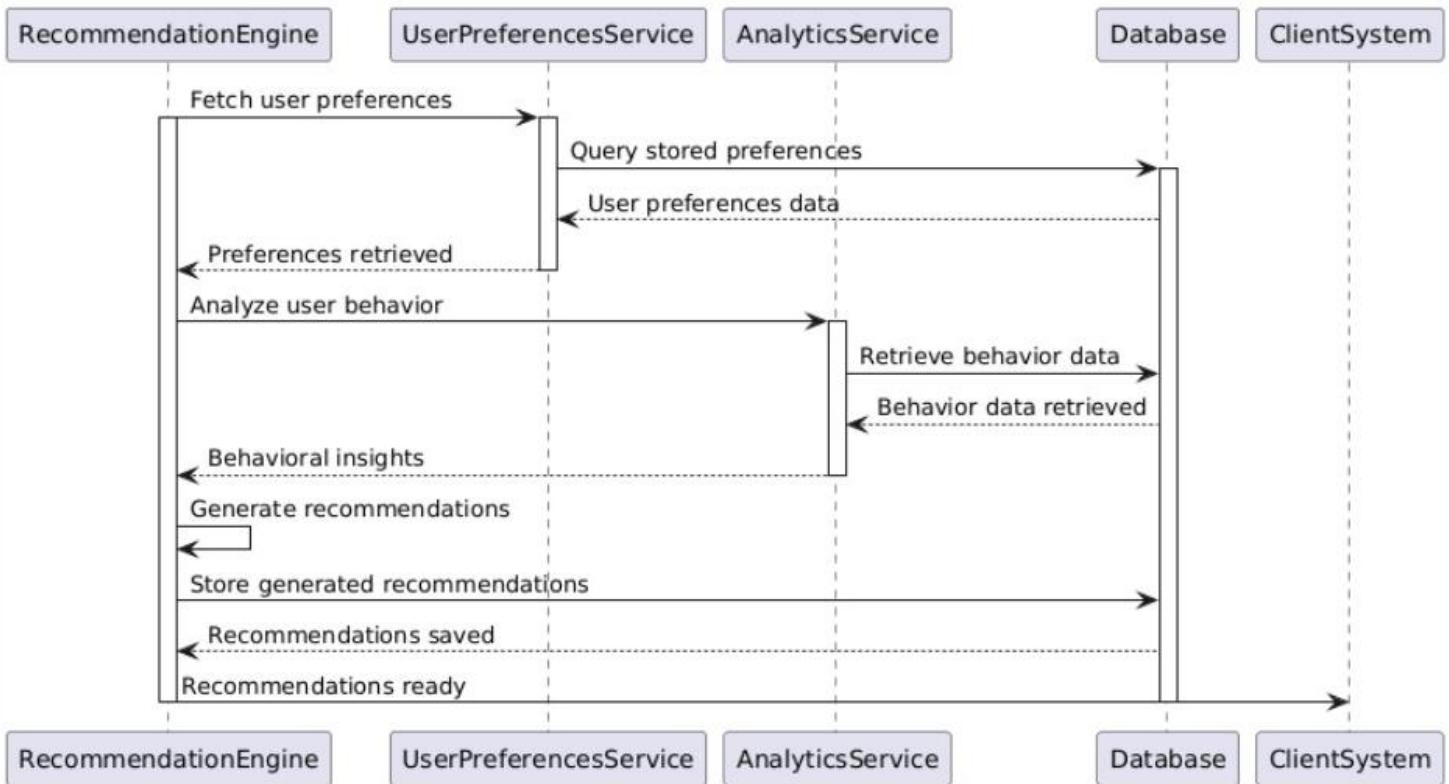


Figure 71: Sequence diagram of Fetch Personalized Recommendations.

4.3.26 List Available Auto Services

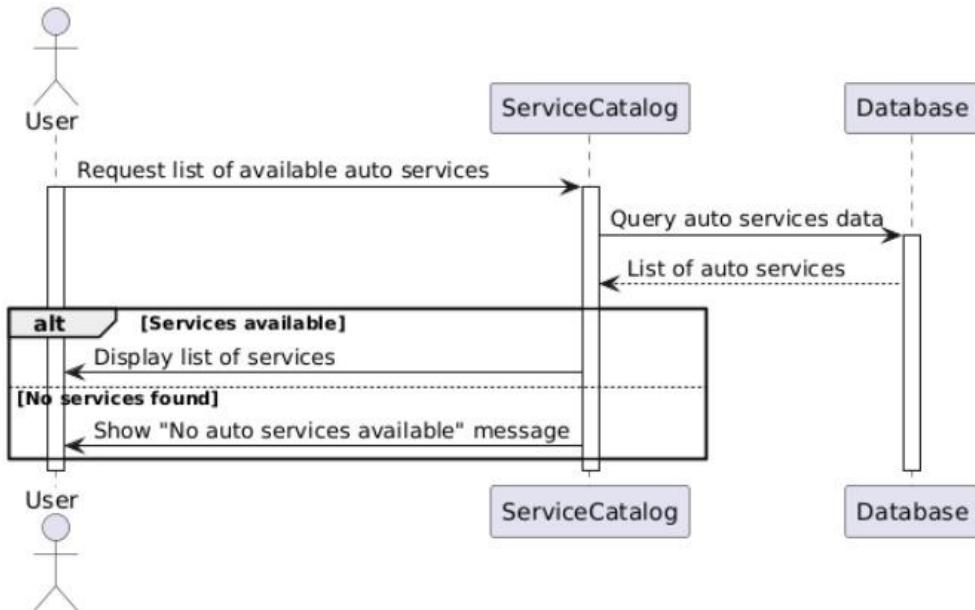


Figure 72: Sequence diagram of List Available Auto Services.

4.3.27 Manage Bookings

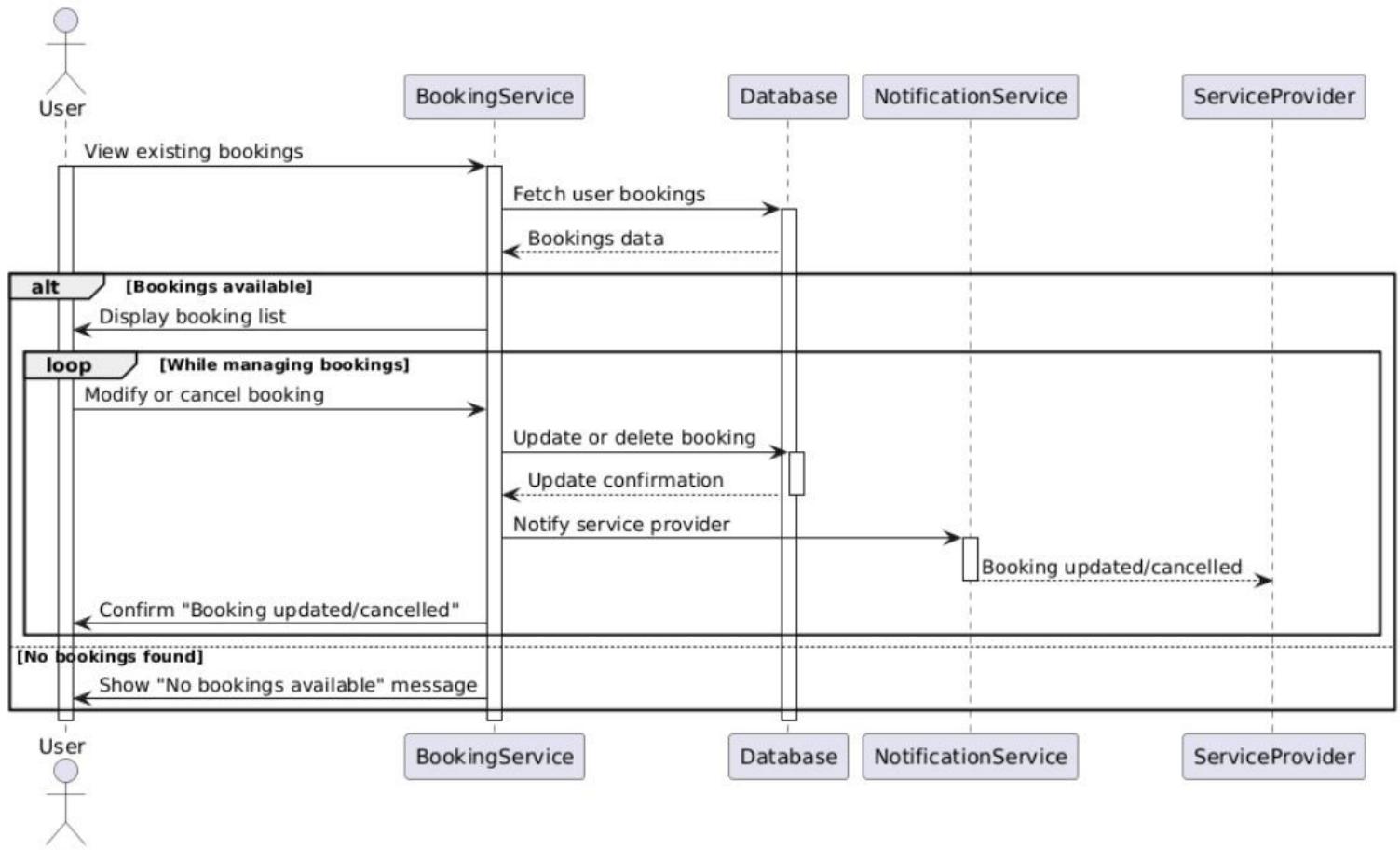


Figure 73: Sequence diagram of Manage bookings.

4.3.28 Generate Monthly/Yearly Reports

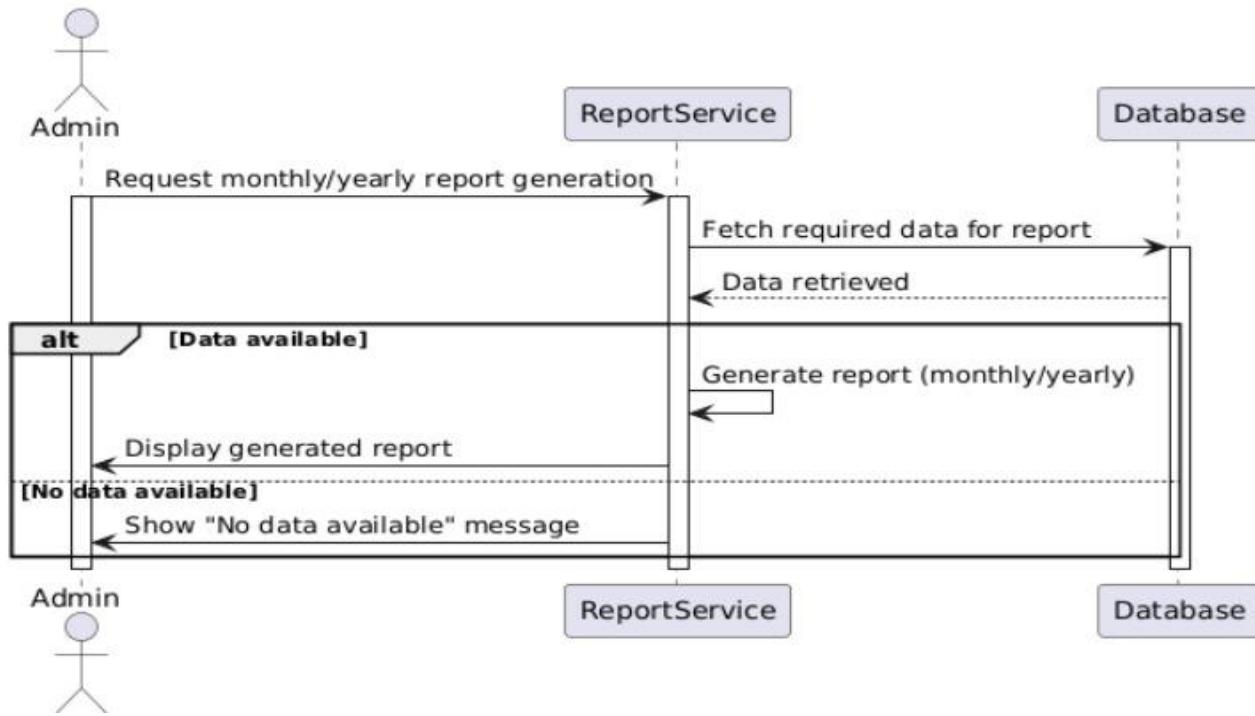


Figure 74: Sequence diagram of Generate Monthly/Yearly Reports.

4.3.29 Export Reports in Desired Format

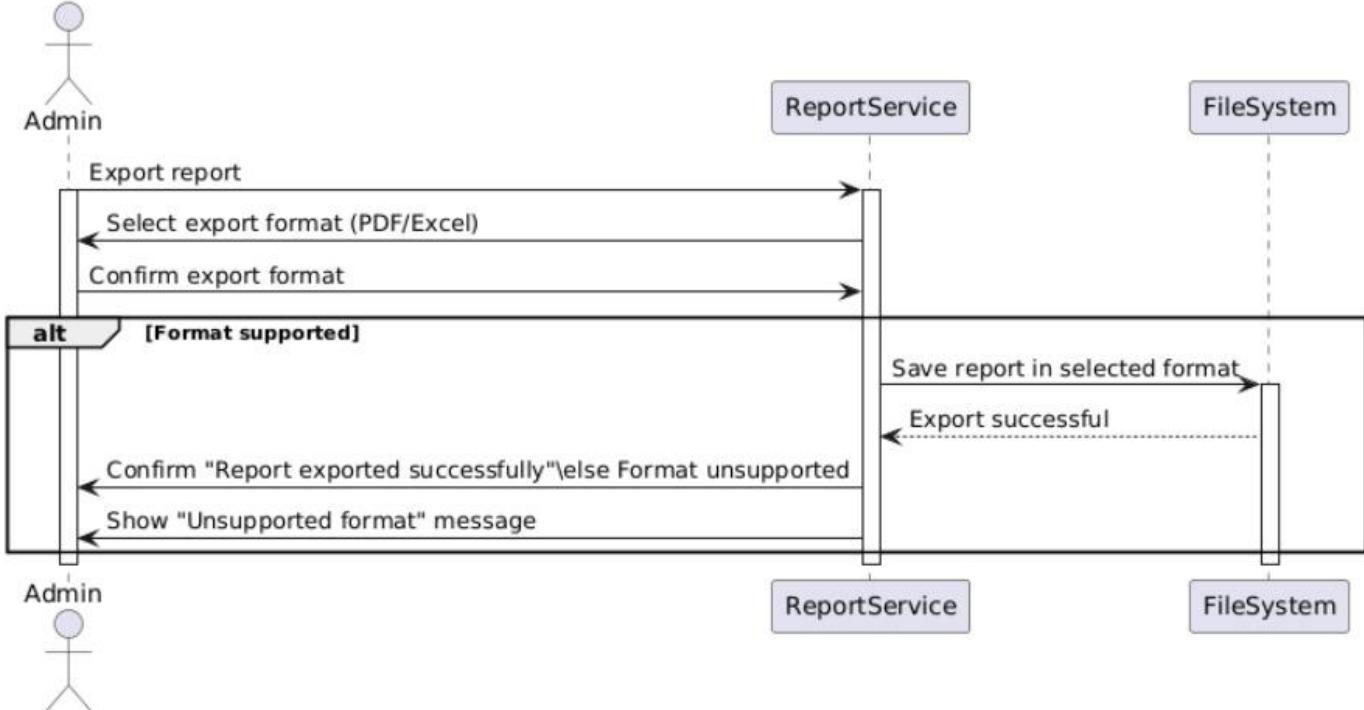


Figure 75: Sequence diagram of Export Reports in Desired Format.

4.3.30 Post Questions or Answers

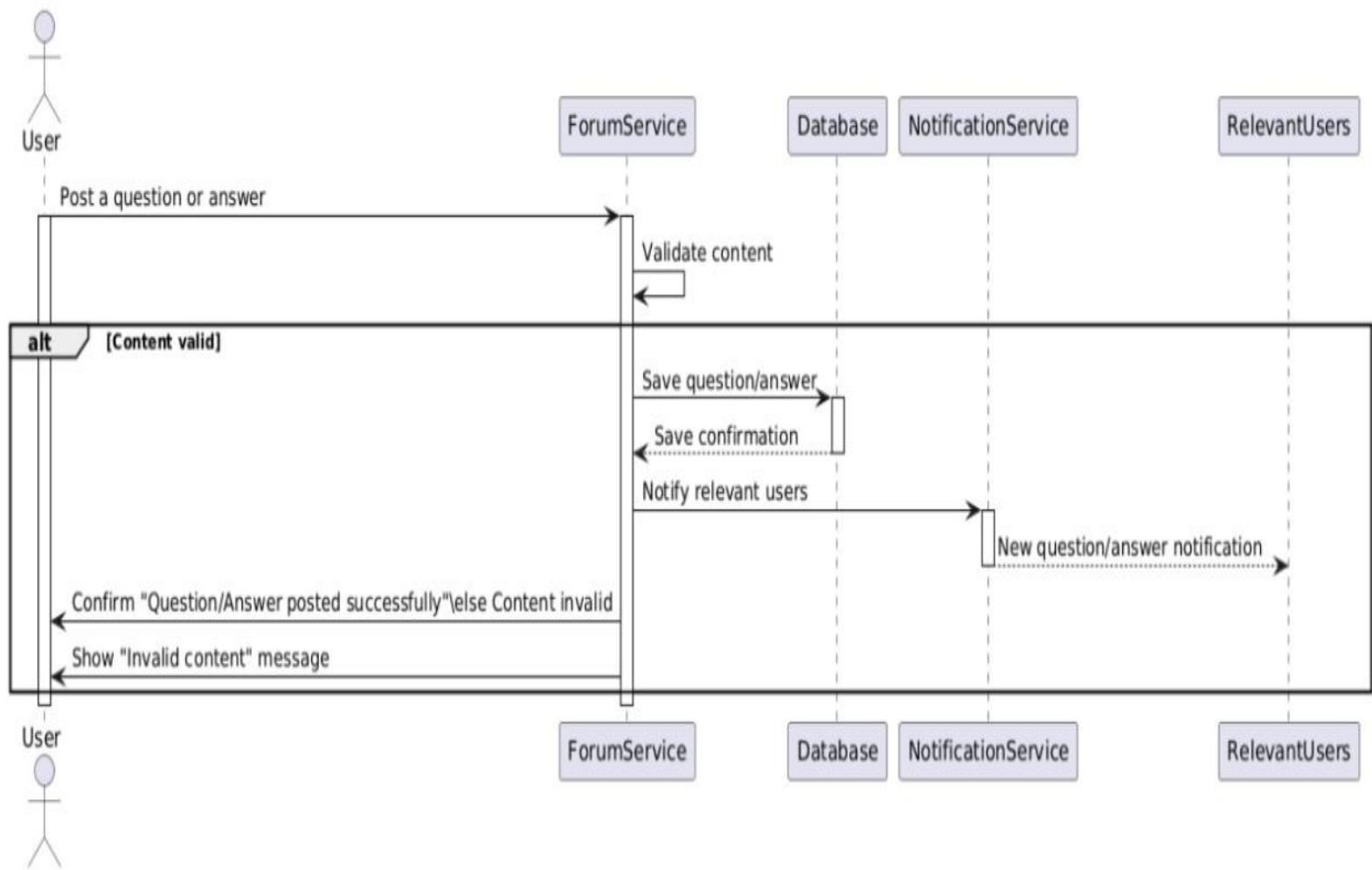


Figure 76: Sequence diagram of Post Question/Answers.

4.3.31 Interact with Other Users in Discussions

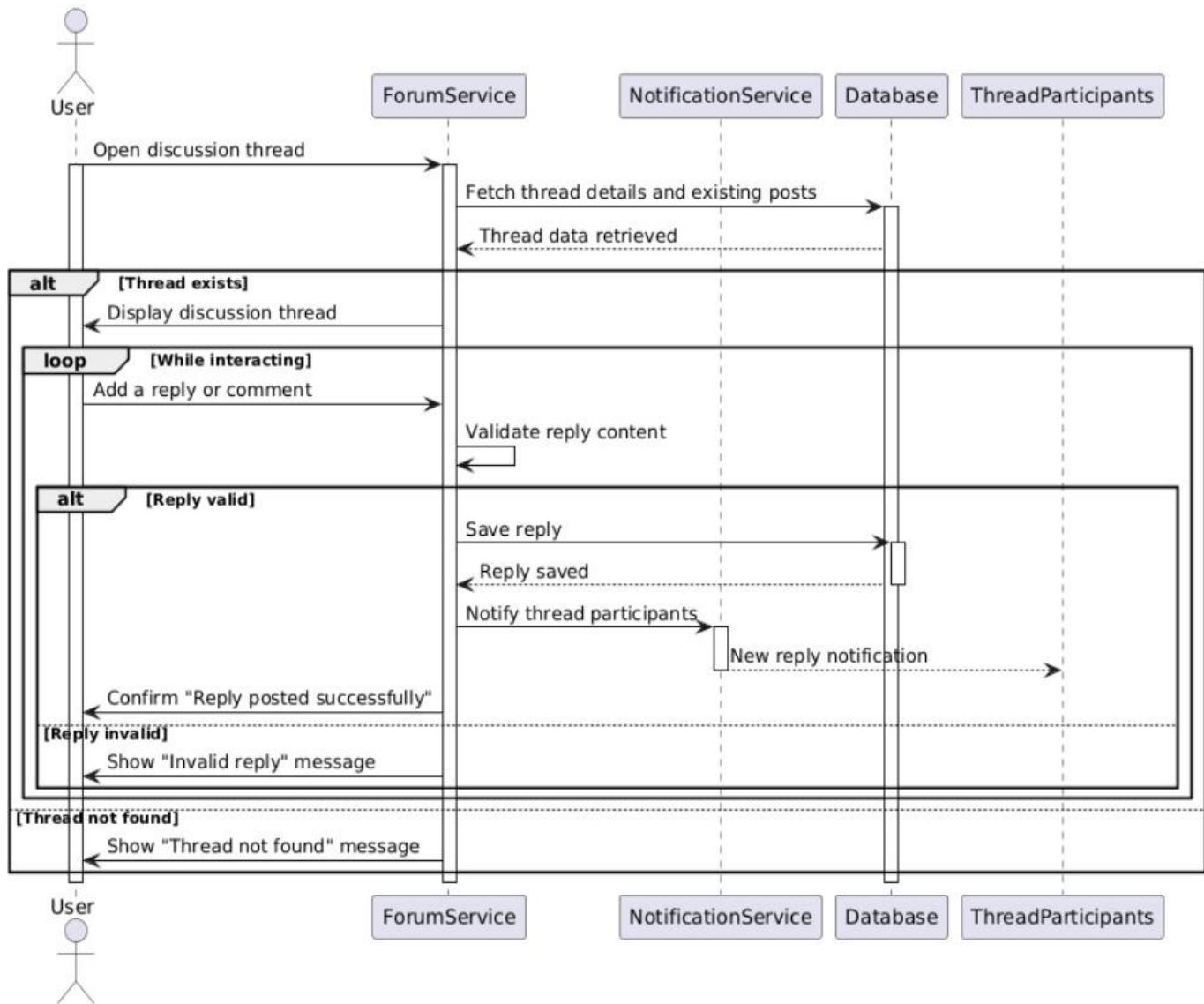


Figure 77: Sequence diagram of Interact with Users in Discussions.

4.3.32 Create Clubs

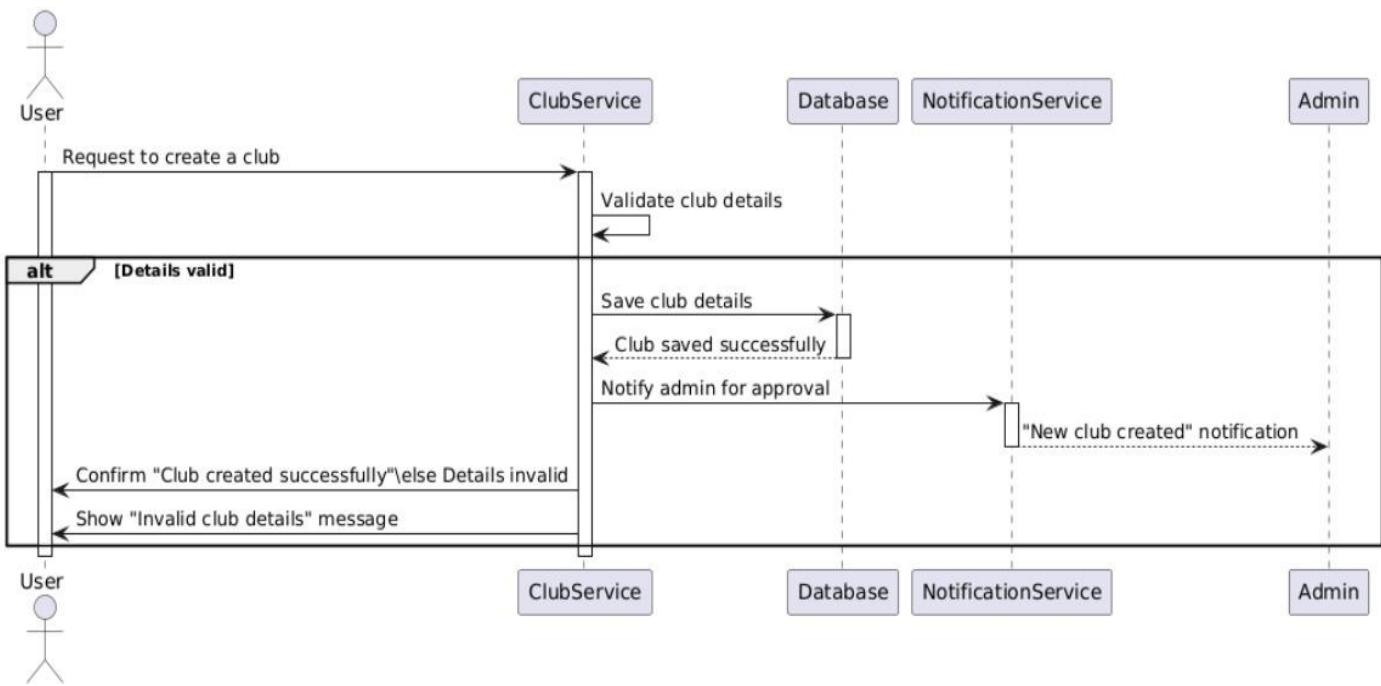


Figure 78: Sequence diagram of Create Clubs.

4.3.33 Create Groups

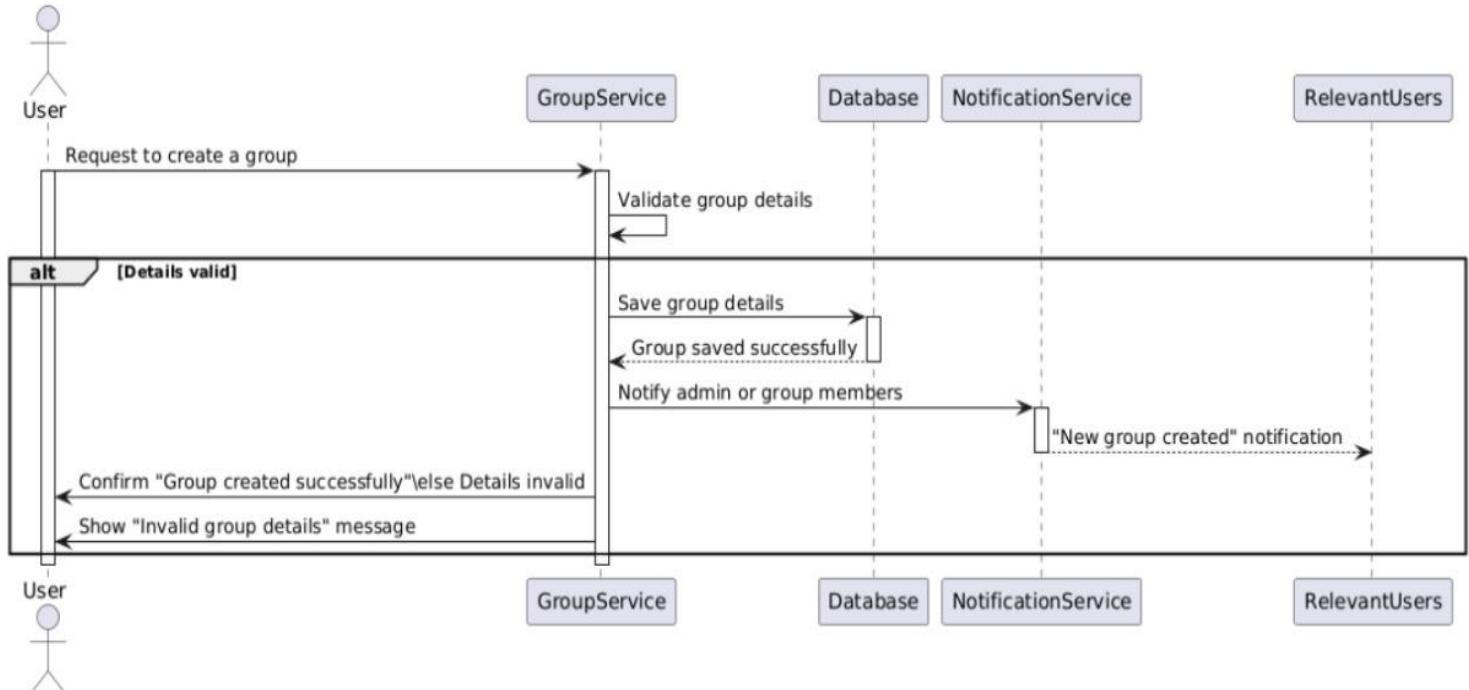


Figure 79: Sequence diagram of Create Groups.

4.3.34 Add Expense

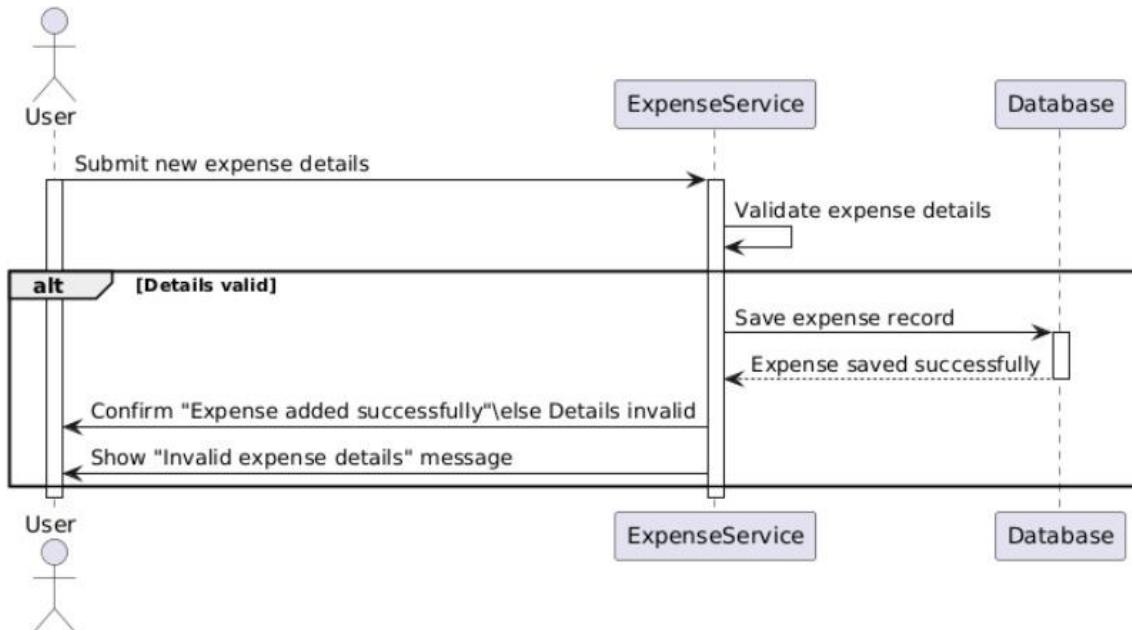


Figure 80: Sequence diagram of Add Expense.

4.3.35 View Expense Details

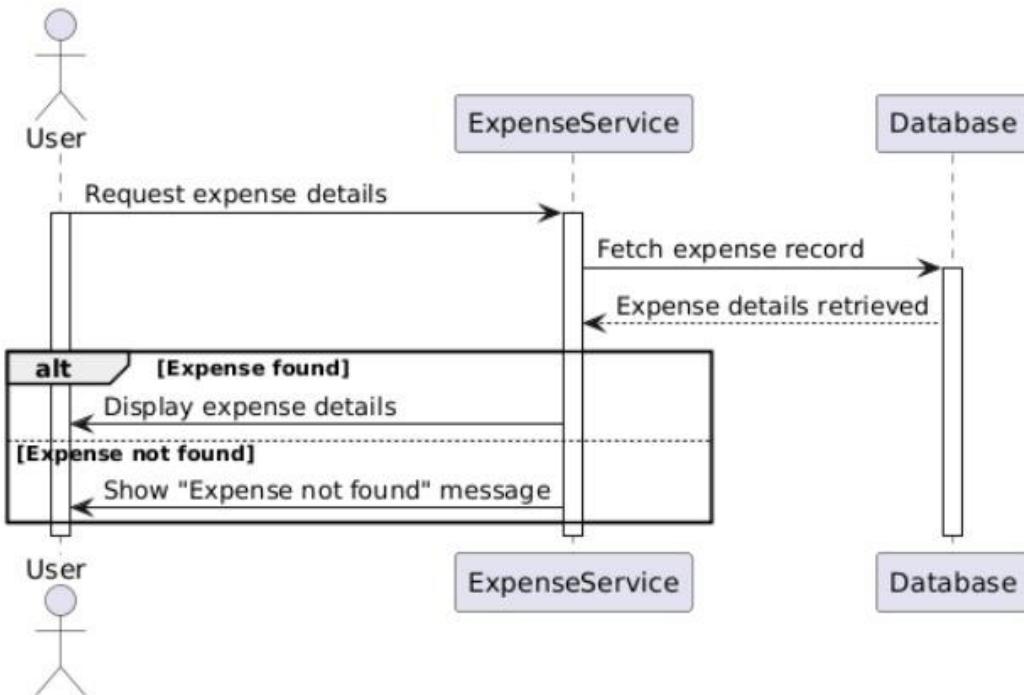


Figure 81: Sequence diagram of View Expense Details.

4.3.36 Edit or Delete Expense

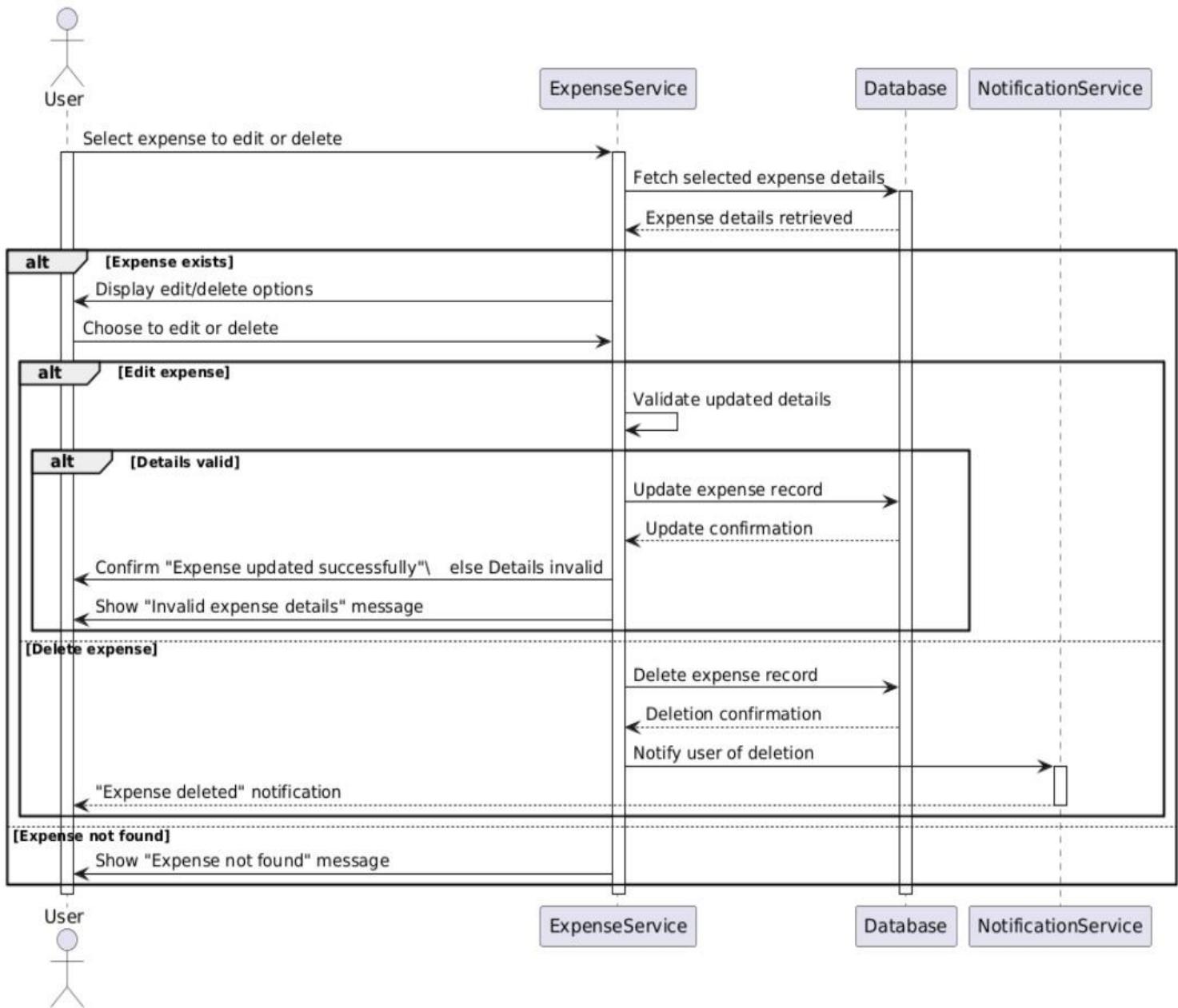


Figure 82: Sequence diagram of Edit/Delete Expense.

4.3.37 Connect OBD-II Device

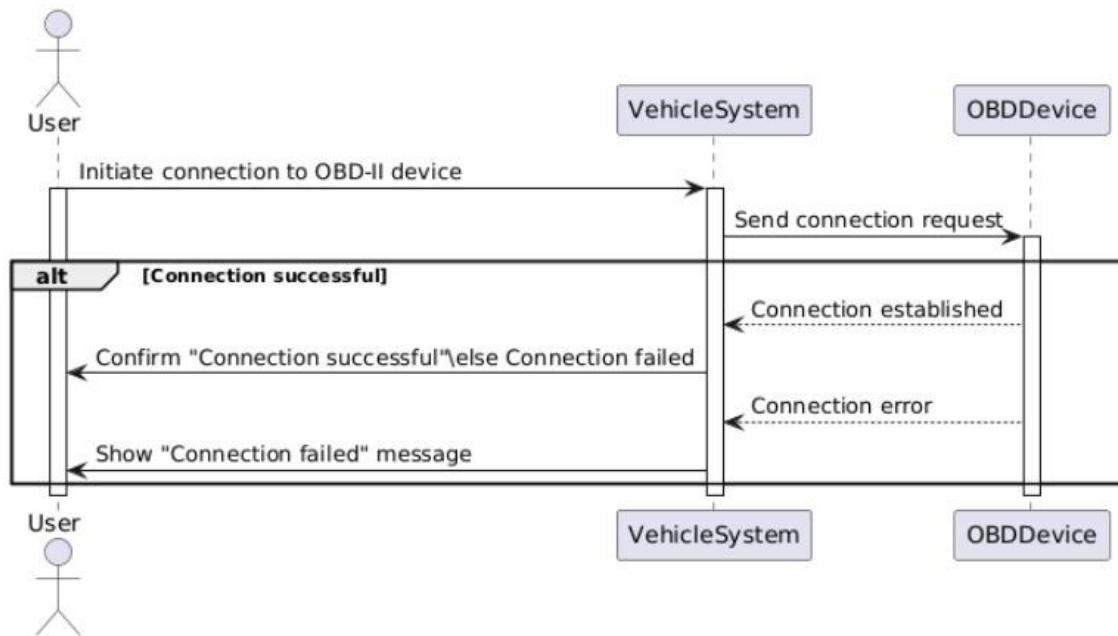


Figure 83: Sequence diagram of Connect OBD-II Device.

4.3.38 Retrieve Vehicle Health Data

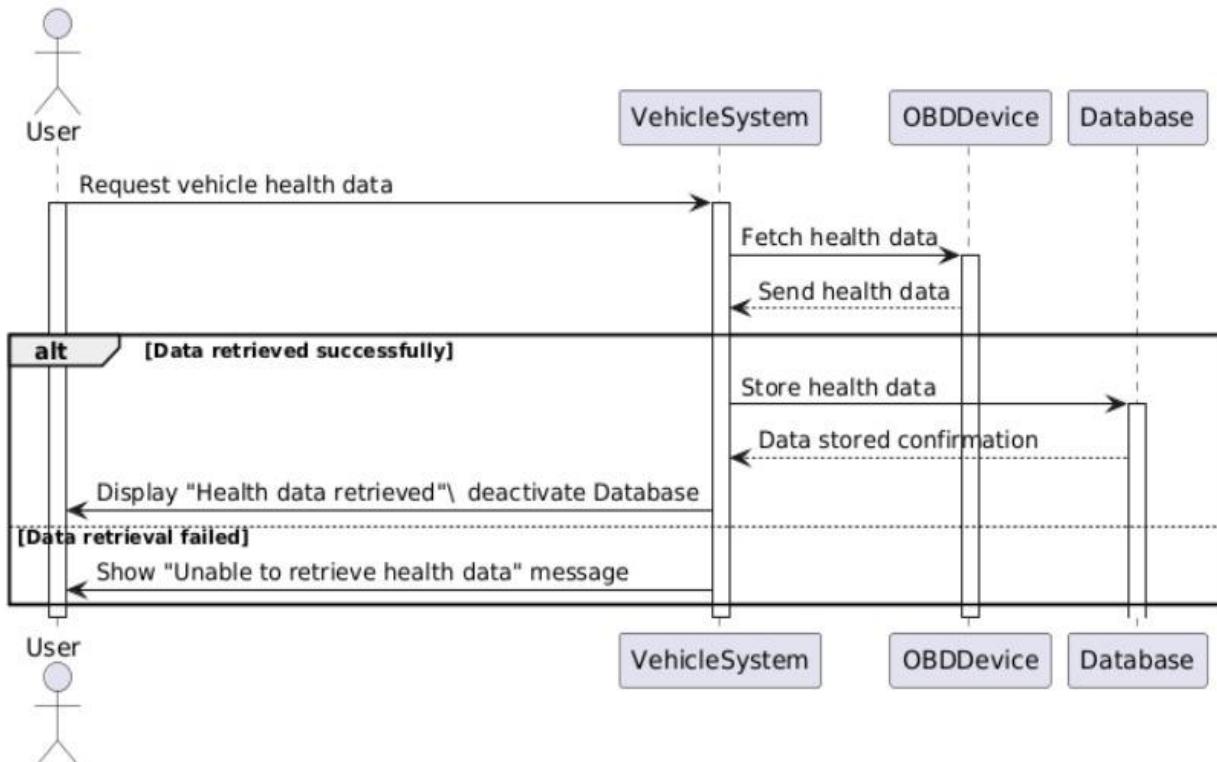


Figure 84: Sequence diagram of Retrieve Vehicle Health Data.

4.3.39 Display and Log Health Metrics

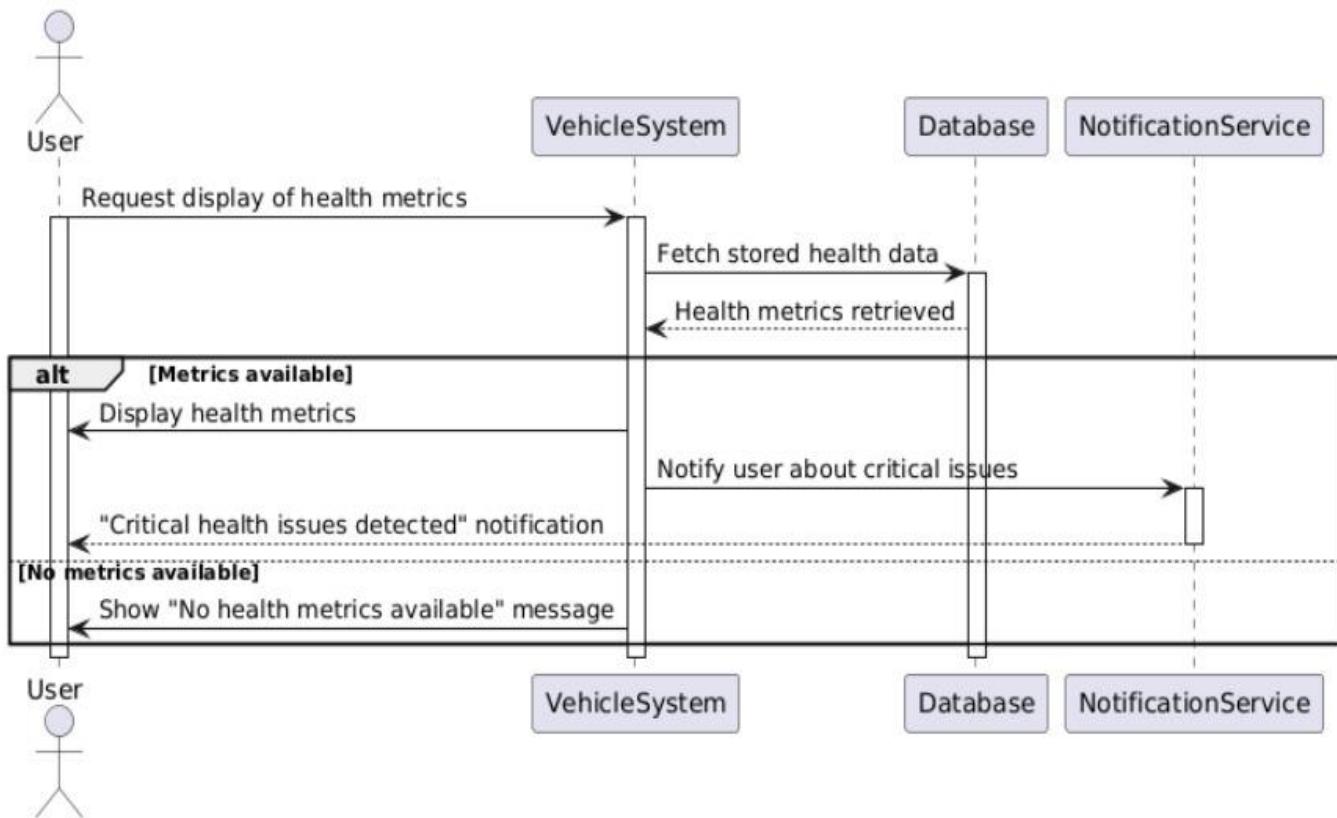


Figure 85: Sequence diagram of Display & Log Health Metrics.

4.4 State Transition Diagrams

4.4.1 User Account

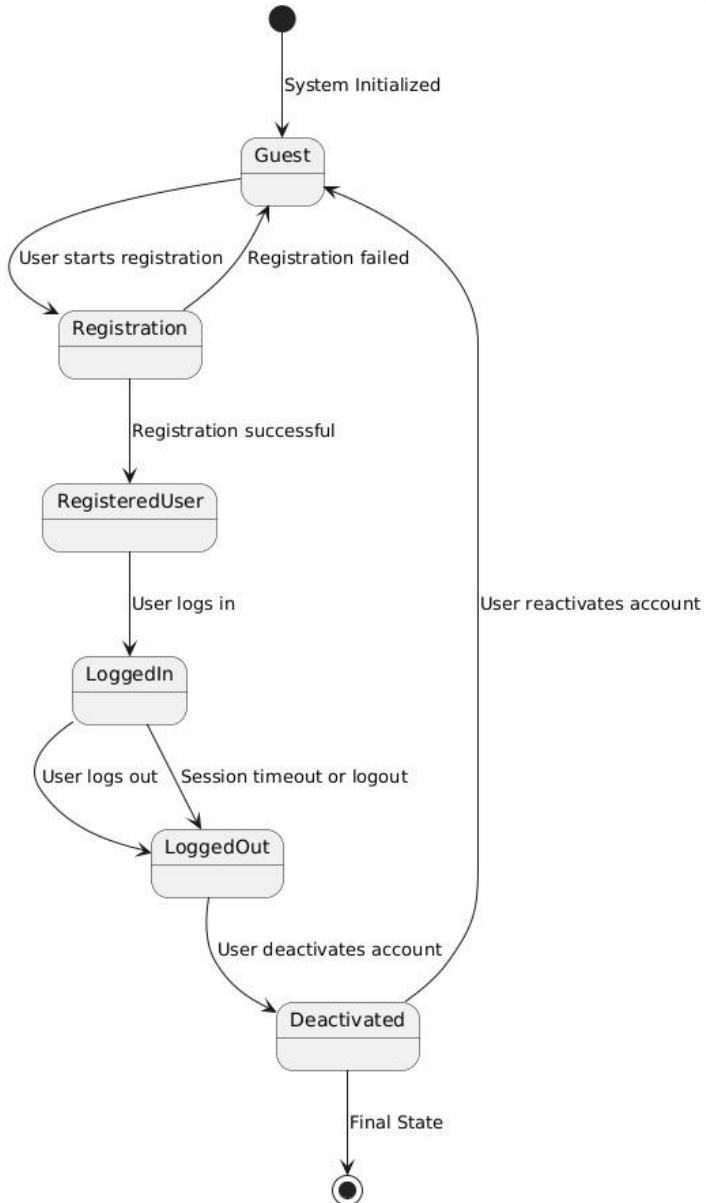


Figure 86: State Transition Diagram for User Account.

4.4.2 Vehicle Listing

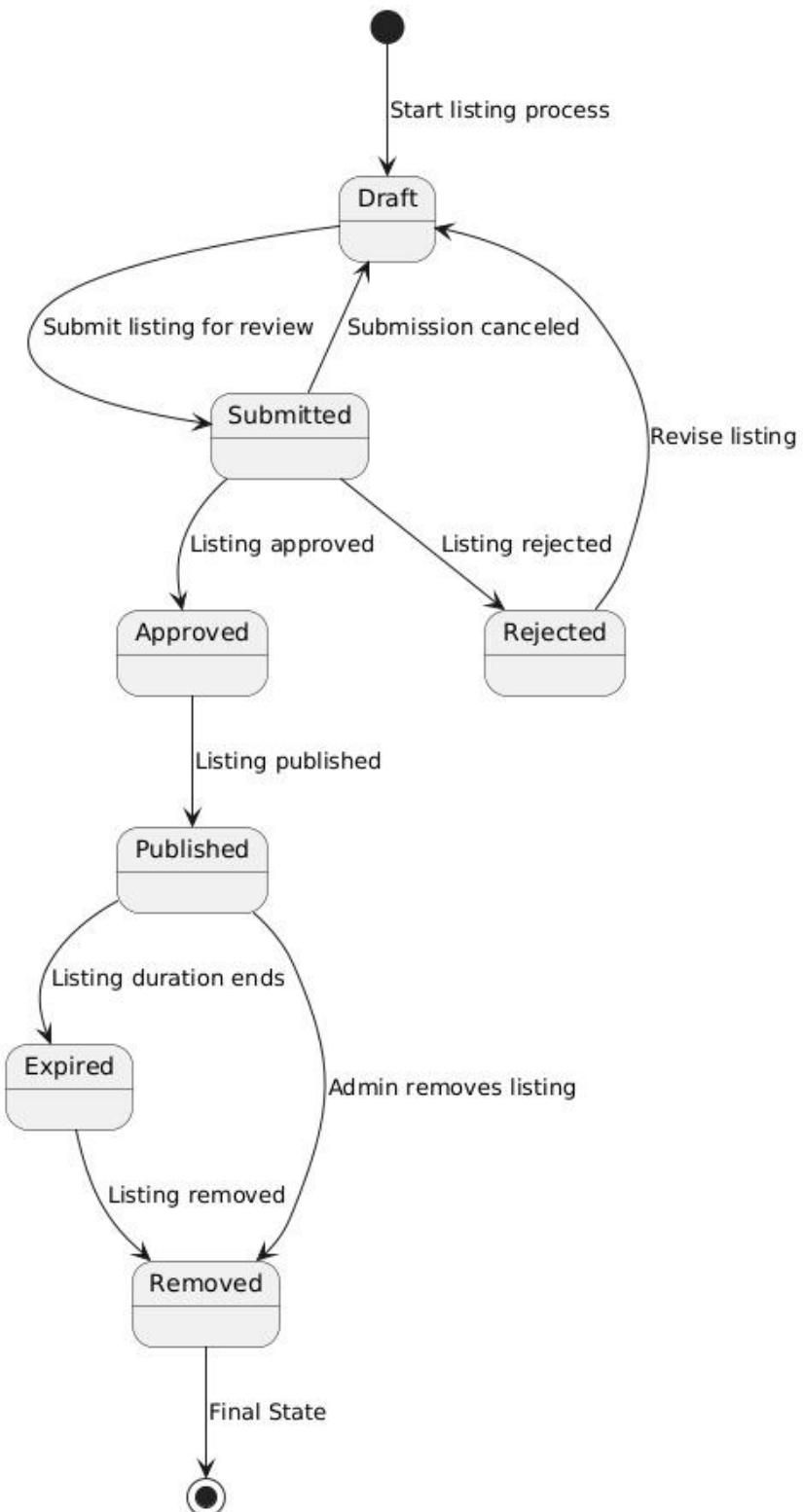


Figure 87: State Transition Diagram for Vehicle Listing.

4.4.3 Chat Session

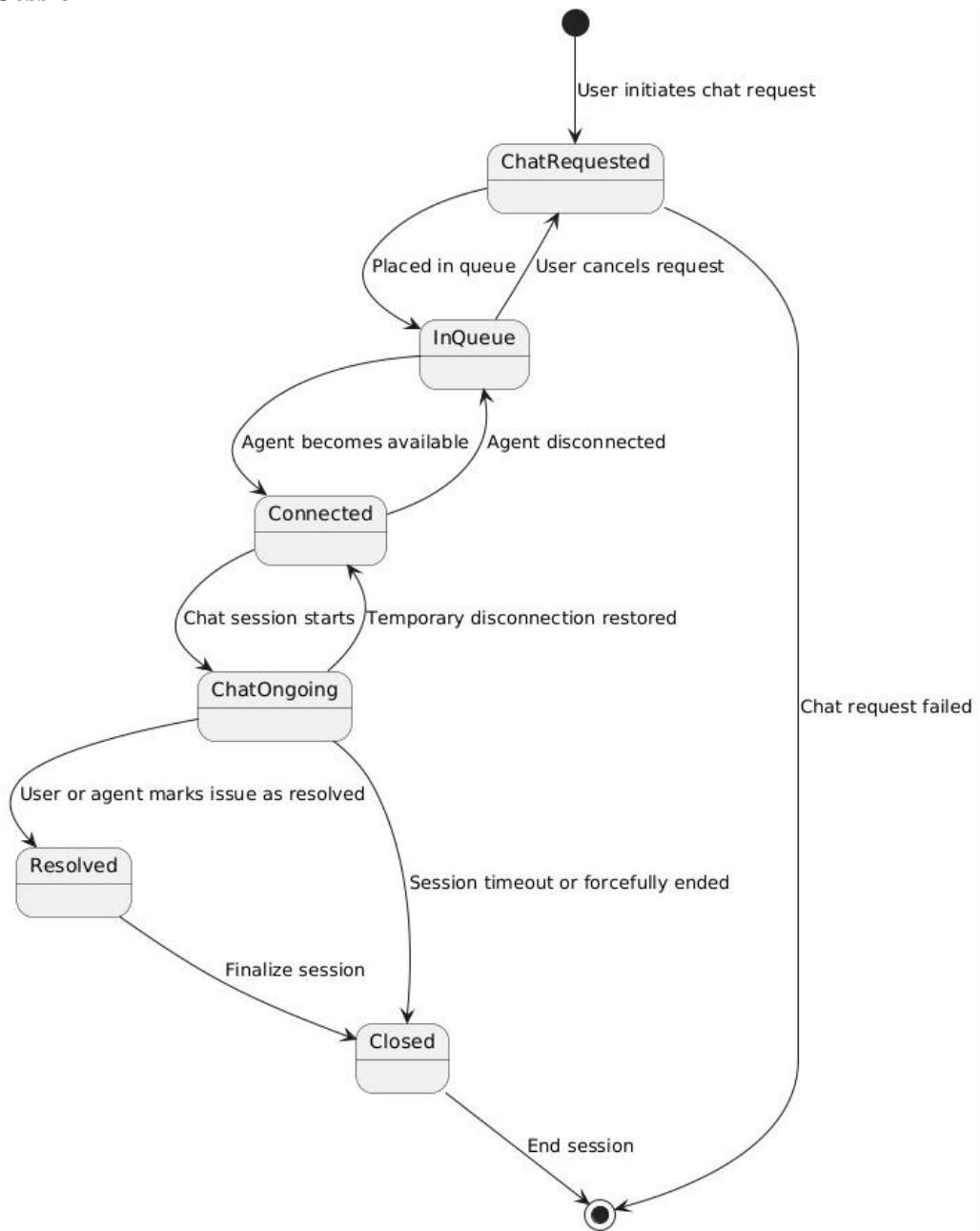


Figure 88: State Transition Diagram for Chat Session.

4.4.4 Service Reminder

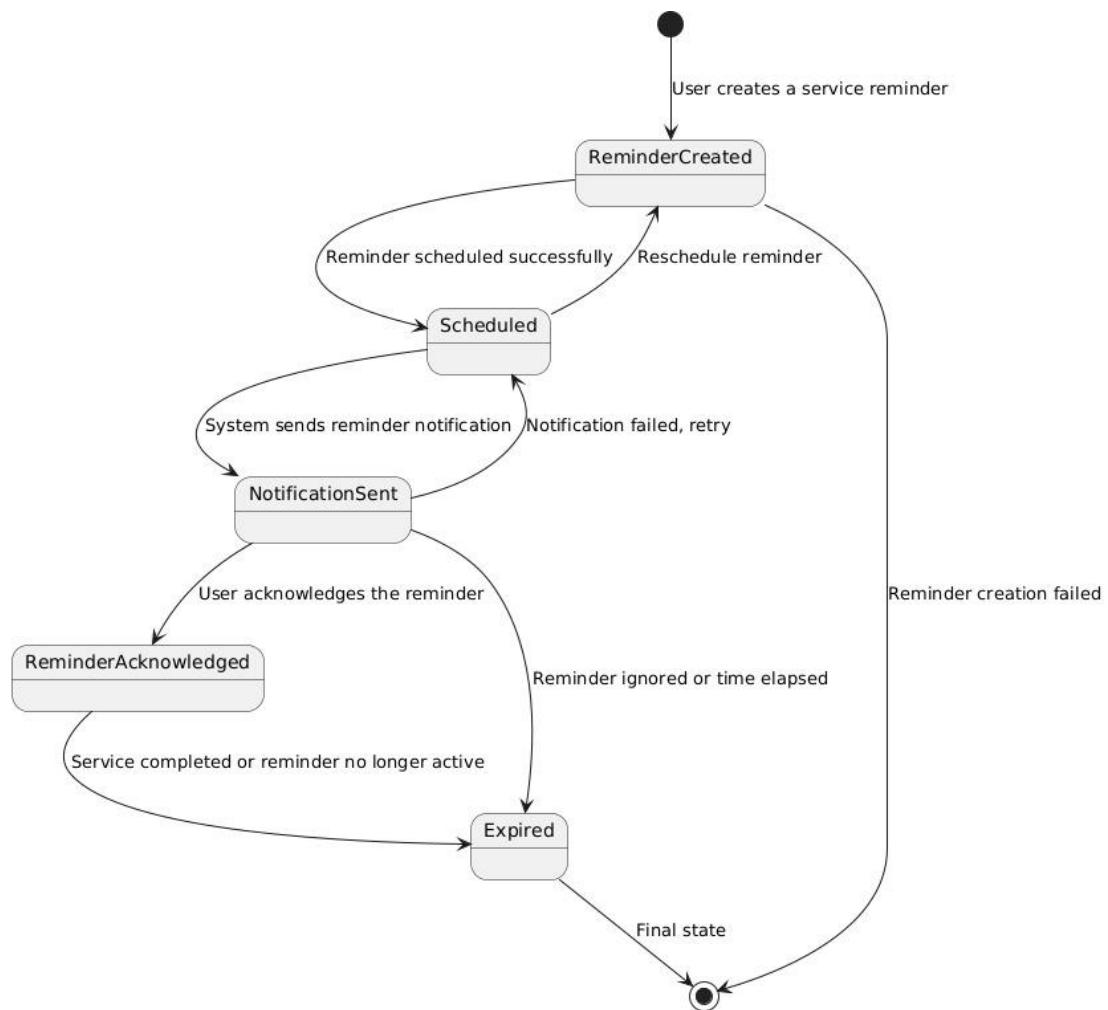


Figure 89: State Transition Diagram for Service Reminder.

4.4.5 Vehicle Health

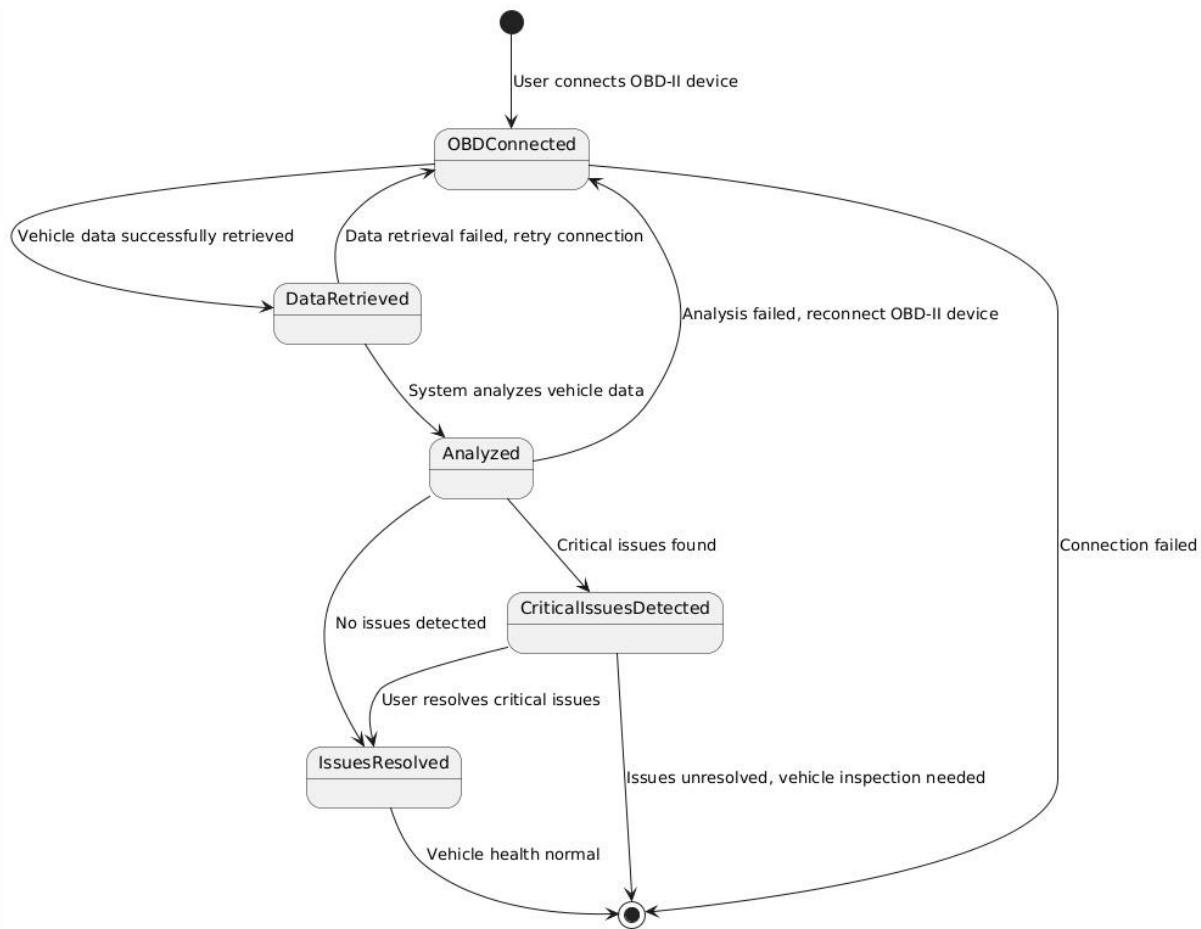


Figure 90: State Transition Diagram for Vehicle Health.

4.4.6 Content Management

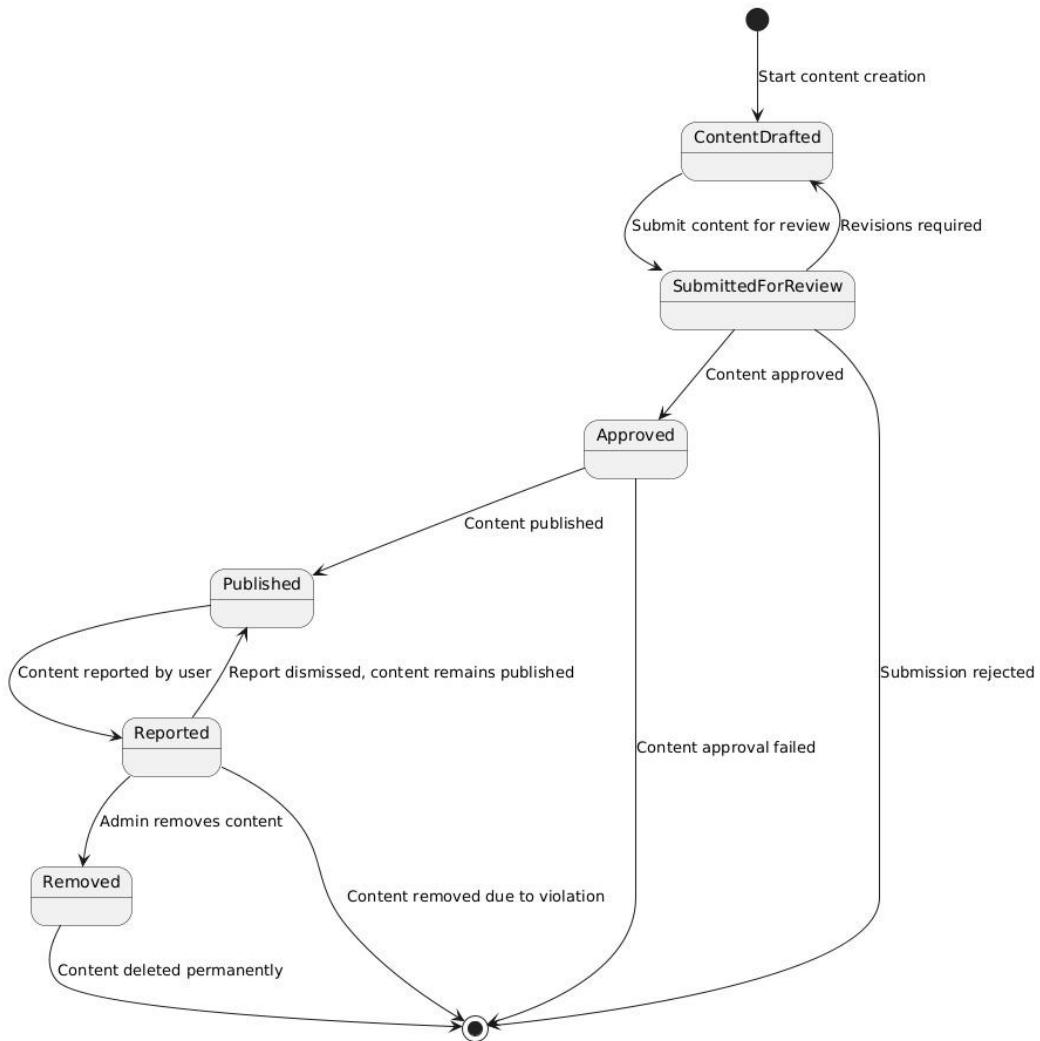


Figure 91: State Transition Diagram for Content Management.

4.4.7 Forum Post

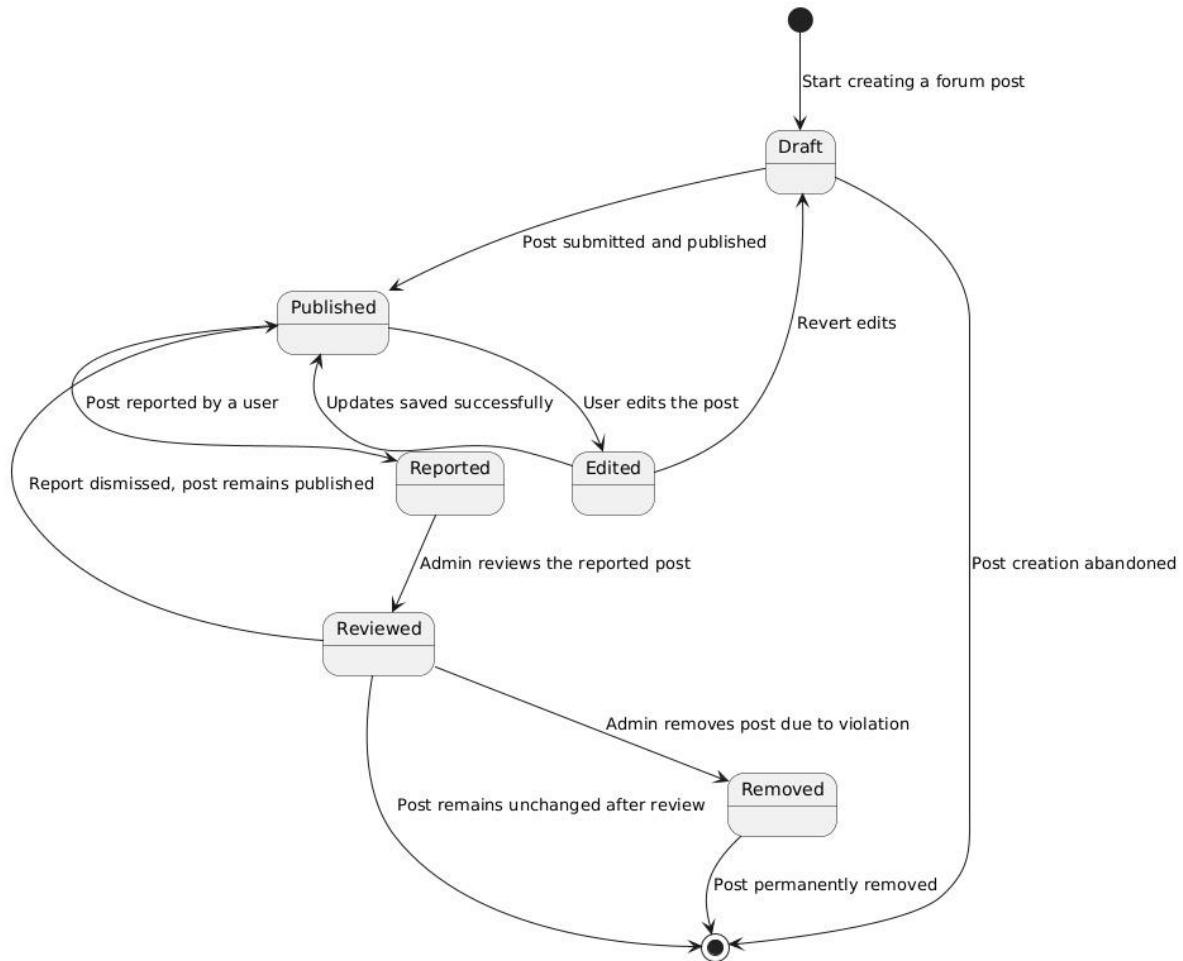


Figure 92: State Transition Diagram for Forum Post.

4.4.8 Recommendation System

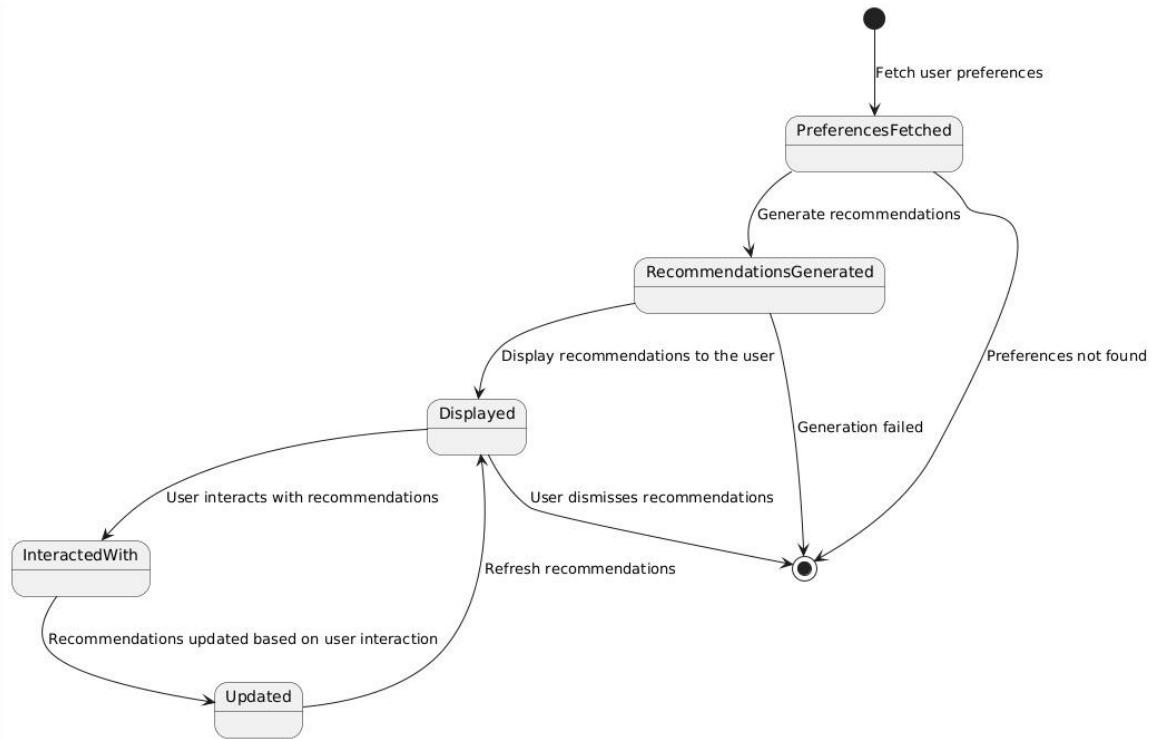


Figure 93: State Transition Diagram for Recommendation System.

4.4.9 Expense Tracking

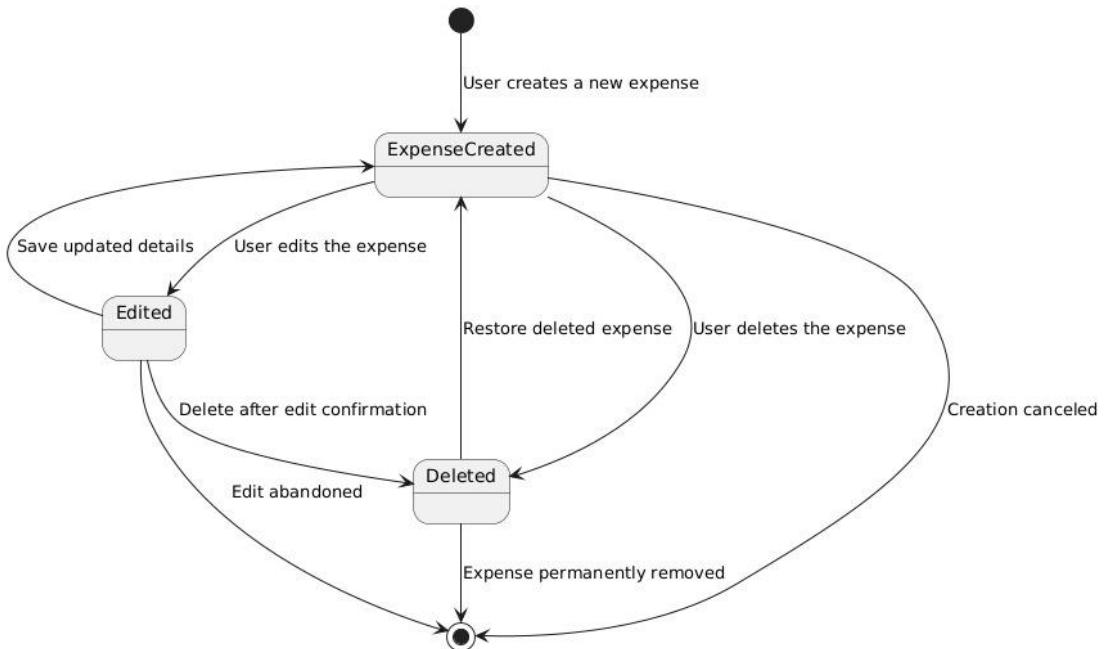


Figure 94: State Transition Diagram for Expense Tracking.

4.4.10 Report

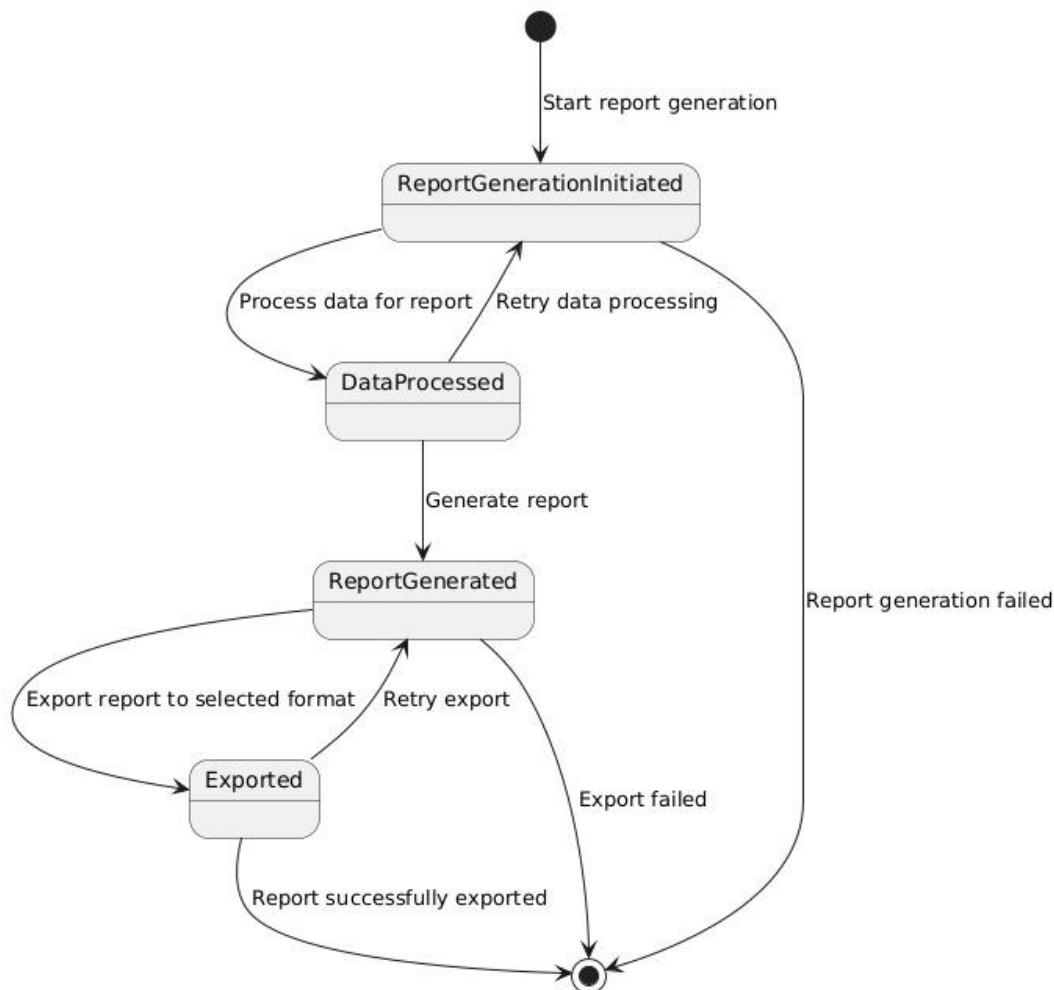


Figure 95: State Transition Diagram for Report.

4.4.11 Notification

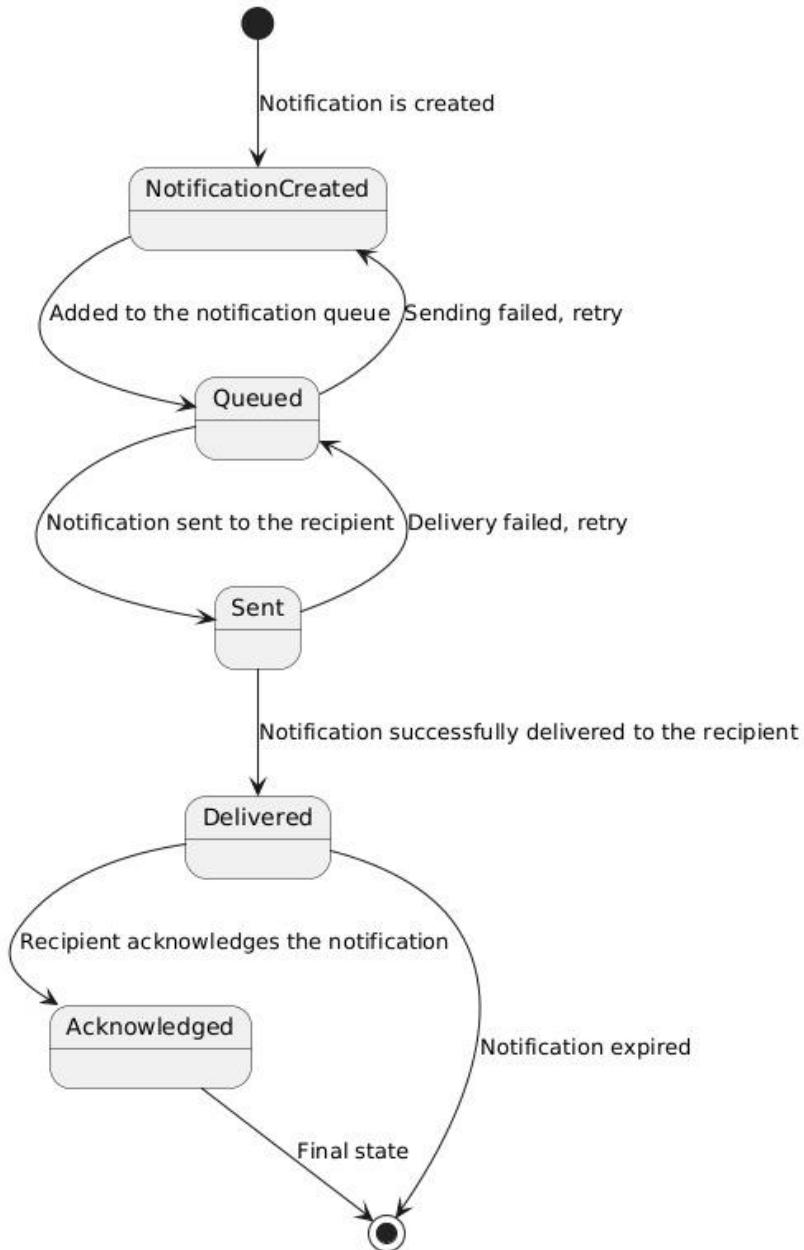


Figure 96: State Transition Diagram for Notification.

4.4.12 Verification

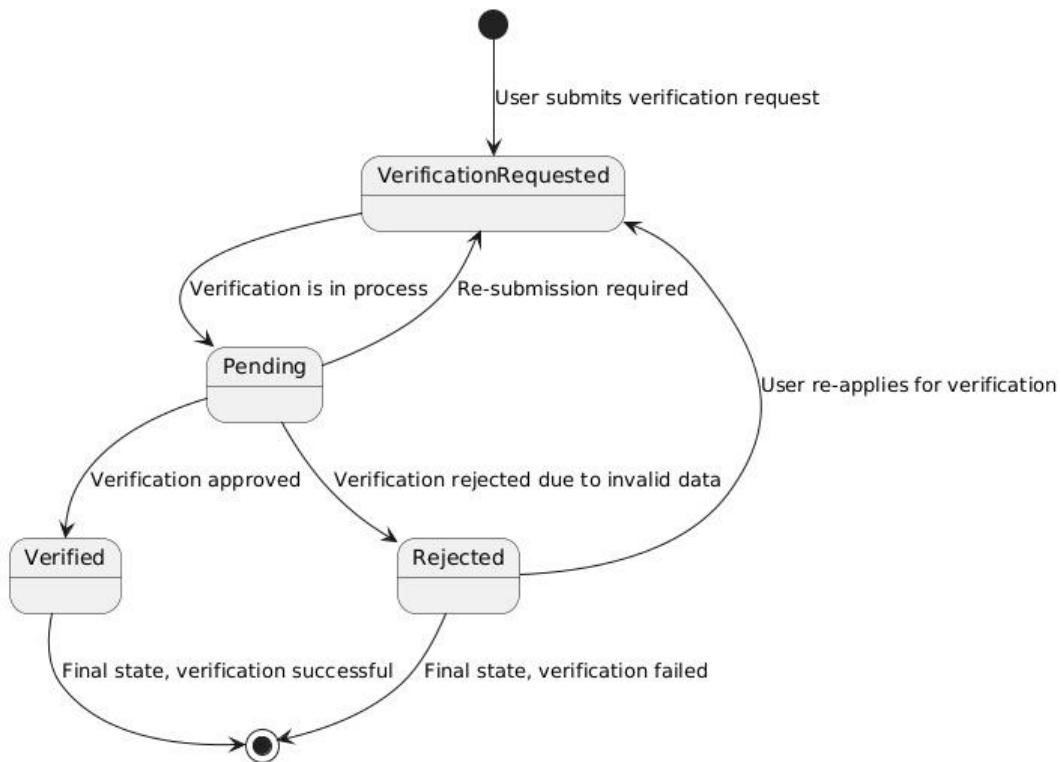


Figure 97: State Transition Diagram for Verification.

5. Data Design

The Motors Bay app uses Firebase as its cloud-based NoSQL database for efficient data storage and real-time synchronization. Data is structured in JSON format using Firestore, ensuring seamless management of user profiles, listings, and chats. Firebase Authentication securely handles user login and access. Its scalability and real-time updates make Firebase ideal for features like User Management, Chat and Notifications.

5.1 Data Dictionary

5.1.1 User

Table 1: Users Collection

Name	Data Type	Constraints	Description
_id	String	Primary Key	Auto-generated unique identifier for each user.
email	String	Required, Unique	Email address of the user.
password	String	Required	Password for account authentication.
name	String	Required	Full name of the user.
role	String	Default: Buyer	Role of the user (Buyer, Seller, Admin, Dealer).
profilePicture	String	Optional	URL for the user's profile picture.
status	String	Default: Active	Status of the account (Active, Inactive).
timestamps	Date	Auto-generated	Account creation and update timestamps.

5.1.2 Vehicle

Table 2: Vehicle Collection

Name	Data Type	Constraints	Description
_id	String	Primary Key	Auto-generated unique identifier for each vehicle.
sellerID	ObjectId	Required	Reference to the seller user.
title	String	Required	Title or name of the vehicle listing.
make	String	Required	Vehicle manufacturer (e.g., Toyota, Ford).
model	String	Required	Model of the vehicle.
year	Integer	Required	Manufacturing year of the vehicle.
description	String	Optional	Detailed description of the vehicle.
status	String	Default: Draft	Status of the listing (Draft, Approved, Removed).
media	Array<Object>	Optional	Photos/videos associated with the vehicle.
timestamps	Date	Auto-generated	Creation and update timestamps.

5.1.3 Booking

Table 3: Booking Collection

Name	Data Type	Constraints	Description
_id	String	Primary Key	Auto-generated unique identifier for each booking.
vehicleID	ObjectId	Required	Reference to the booked vehicle.
buyerID	ObjectId	Required	Reference to the buyer user.
status	String	Default: Pending	Status of the booking (Pending, Confirmed).
scheduledDate	Date	Required	Date of the booking.
timestamps	Date	Auto-generated	Booking creation and update timestamps.

5.1.4 Post

Table 4: Post Collection

Name	Data Type	Constraints	Description
_id	String	Primary Key	Auto-generated unique identifier for each post.
user	ObjectId	Required	Reference to the post creator.
content	String	Required	Text content of the post.
media	Object	Optional	Associated images/videos with the post.
likes	Array<ObjectId>	Optional	Users who liked the post.
comments	Array<ObjectId>	Optional	Comments made on the post.
status	String	Default: Active	Status of the post (Active, Reported).
timestamps	Date	Auto-generated	Post creation and update timestamps.

5.1.5 Notification

Table 5: Notification Collection

Name	Data Type	Constraints	Description
_id	String	Primary Key	Auto-generated unique notification ID.
userID	ObjectId	Required	Reference to the target user.
content	String	Required	Notification message content.
status	String	Default: Created	Notification state (Created, Sent, Acknowledged).
timestamps	Date	Auto-generated	Notification creation timestamp.

5.1.6 Verification

Table 6: Verification Collection

Name	Data Type	Constraints	Description
_id	String	Primary Key	Auto-generated verification ID.
userID	ObjectId	Required	Reference to the user being verified.
status	String	Default: Pending	Verification state (Pending, Verified, Rejected).
timestamps	Date	Auto-generated	Creation and completion timestamps.

5.1.7 Expense

Table 7: Expense Collection

Name	Data Type	Constraints	Description
_id	String	Primary Key	Unique identifier for the expense entry.
userID	ObjectId	Required	Reference to the user recording the expense.
amount	Float	Required	Amount spent on expense.
category	String	Required	Category (Fuel, Repair, Maintenance).
date	Date	Required	Date of the expense.
notes	String	Optional	Additional notes for the expense.

5.1.8 Review

Table 8: Review Collection

Name	Data Type	Constraints	Description
_id	String	Primary Key	Unique identifier for each review.
vehicleID	ObjectId	Required	Reference to the vehicle reviewed.
buyerID	ObjectId	Required	Reference to the buyer submitting the review.
rating	Integer	Required	Rating score (1-5).
comment	String	Optional	Review comment provided by the buyer.
timestamps	Date	Auto-generated	Creation timestamp for the review.

5.1.9 Report

Table 9: Report Collection

Name	Data Type	Constraints	Description
_id	String	Primary Key	Auto-generated unique report ID.
reporterID	ObjectId	Required	Reference to the user reporting content.
contentID	ObjectId	Required	Reference to the reported content.
contentType	String	Required	Type of content (Post, Comment, Listing).
status	String	Default: Pending	Report status (Pending, Reviewed, Resolved).
timestamps	Date	Auto-generated	Report creation and resolution timestamps.

5.1.10 Model Visualization

Table 10: Visualization Collection

Name	Data Type	Constraints	Description
_id	String	Primary Key	Unique identifier for 3D model entry.
userID	ObjectId	Required	Reference to the user who saved the model.
modelPath	String	Required	Path to the 3D model file.
customOptions	Object	Optional	Customization details (color, rims, etc.).
comparisonID	Array<ObjectId>	Optional	References to compared models.
timestamps	Date	Auto-generated	Creation and update timestamps.

5.1.11 Event

Table 11: Event Collection

Name	Data Type	Constraints	Description
_id	String	Primary Key	Unique identifier for the event.
title	String	Required	Title of the event.
description	String	Optional	Description of the event.
date	Date	Required	Event date.
location	String	Optional	Event location.
organizerID	ObjectId	Required	User ID or dealership organizing the event.
RSVPs	Array<ObjectId>	Optional	References to users attending the event.
status	String	Default: Active	Status (Active/Cancelled).
timestamps	Date	Auto-generated	Creation and update timestamps.

5.1.12 News

Table 12: News Collection

Name	Data Type	Constraints	Description
_id	String	Primary Key	Unique identifier for the news article.
title	String	Required	Title of the news article.
content	String	Required	Content of the news article.
author	String	Required	Author of the article.
media	Object	Optional	Associated media (images/videos).
sponsorID	ObjectId	Optional	Reference to the sponsoring dealership.
timestamps	Date	Auto-generated	Creation and update timestamps.

5.1.13 Forum

Table 13: Forum Collection

Name	Data Type	Constraints	Description
_id	String	Primary Key	Unique identifier for the post.
topic	String	Required	Topic or title of the post.
userID	ObjectId	Required	Reference to the user creating the post.
content	String	Required	Content of the post.
likes	Array<ObjectId>	Optional	List of users who liked the post.
comments	Array<ObjectId>	Optional	List of comments on the post.
media	Object	Optional	Associated media (images/videos).
status	String	Default: Active	Status (Active/Reported/Removed).
timestamps	Date	Auto-generated	Creation and update timestamps.

5.1.14 Group/Club

Table 14: Group/Club Collection

Name	Data Type	Constraints	Description
_id	String	Primary Key	Unique identifier for the group/club.
name	String	Required	Name of the group or club.
adminID	ObjectId	Required	User ID of the group admin.
members	Array<ObjectId>	Optional	List of user IDs in the group.
description	String	Optional	Description of the group.
createdAt	Date	Auto-generated	Date when the group was created.

5.1.15 Recommendation

Table 15: Recommendation Collection

Name	Data Type	Constraints	Description
_id	String	Primary Key	Unique identifier for the recommendation.
userID	ObjectId	Required	User ID receiving the recommendation.
preferences	Object	Optional	User's preferences (budget, search history).
contentIDs	Array<ObjectId>	Required	Recommended content references.
generatedAt	Date	Auto-generated	Date and time of recommendation generation.
status	String	Default: Generated	Status (Generated/Displayed/Updated).

5.1.16 Service

Table 16: Service Collection

Name	Data Type	Constraints	Description
_id	String	Primary Key	Unique identifier for the service.
title	String	Required	Name or title of the service.
description	String	Optional	Description of the service.
price	Float	Required	Cost of the service.
providerID	ObjectId	Required	Reference to the service provider.
duration	String	Optional	Duration of the service.
status	String	Default: Active	Status (Active/Inactive).
ratings	Float	Optional	Average ratings for the service.
timestamps	Date	Auto-generated	Creation and update timestamps.

5.1.17 ServiceProvider

Table 17: ServiceProvider Collection

Name	Data Type	Constraints	Description
_id	String	Primary Key	Unique identifier for the service provider.
name	String	Required	Name of the provider or business.
contactInfo	String	Required	Contact information.
servicesOffered	Array<String>	Required	List of services offered.
ratings	Float	Optional	Overall rating of the provider.
location	String	Optional	Service provider's location.
timestamps	Date	Auto-generated	Creation and update timestamps.

5.1.18 Maintenance

Table 18: Maintenance Collection

Name	Data Type	Constraints	Description
_id	String	Primary Key	Unique identifier for OBD-II data.
vehicleID	ObjectId	Required	Reference to the vehicle.
healthMetrics	Object	Required	Vehicle health metrics (e.g., engine stats).
issuesDetected	Array<String>	Optional	List of detected issues.
loggedAt	Date	Auto-generated	Timestamp when the data was logged.

5.1.19 Chat

Table 19: Chat Collection

Name	Data Type	Constraints	Description
_id	String	Primary Key	Unique identifier for the message.
sessionID	ObjectId	Required	Chat session reference.
senderID	ObjectId	Required	Reference to the message sender.
receiverID	ObjectId	Required	Reference to the message receiver.
content	String	Required	Content of the message.
media	Object	Optional	Media associated with the message.
status	String	Default: Unread	Status (Sent/Delivered/Read).
timestamps	Date	Auto-generated	Message creation and update timestamps.

6. Implementation

Implementation is a critical part of the business process that requires careful planning, preparation, and execution. It involves taking the necessary steps to ensure that the plan is executed correctly and that the desired results are achieved. Effective implementation requires effective communication, collaboration, and coordination among all stakeholders involved in the process. Following are some of the algorithms used while implementing **Motors Bay**.

7.1 Algorithm

Table 20: Algorithm for User Registration and Login

Algorithm 1: User Registration and Login	
Input:	email, password, profileInfo
Output:	User Account Creation or Login Confirmation
	<pre> 1. FUNCTION RegisterUser(email, password, profileInfo): 2. IF NOT ValidateEmail(email): 3. RETURN "Invalid Email Format" 4. IF NOT ValidatePassword(password): 5. RETURN "Password does not meet security requirements" 6. IF EmailExists(email): 7. RETURN "Email Already Registered" 8. 9. hashedPassword = HashPassword(password) 10. userID = GenerateUniqueUserID() 11. newUser = { 12. "_id": userID, 13. "email": email, 14. "password": hashedPassword, 15. "profileInfo": profileInfo, 16. "isVerified": false, 17. "createdAt": CURRENT_TIMESTAMP 18. } 19. STORE newUser IN DATABASE 20. SENDVerificationEmail(email) 21. RETURN "Registration Successful, Verify Your Email" 22. 23. FUNCTION LoginUser(email, password): 24. userRecord = FetchUserFromDatabase(email) 25. IF NOT userRecord: 26. RETURN "User Not Found" 27. IF VerifyPassword(password, userRecord.password): 28. sessionToken = GenerateSessionToken(userRecord._id) 29. UpdateLastLogin(userRecord._id) 30. RETURN "Login Successful", sessionToken 31. ELSE: 32. RETURN "Incorrect Password"</pre>

Table 21: Algorithm for Vehicle Listing Management

Algorithm 2: Vehicle Search and Filter

Input:	searchCriteria (filters: brand, model, year, etc.)
Output:	Matching Listings
	<pre> 1. FUNCTION SearchListings(searchCriteria): 2. matchingListings = [] 3. FOR listing IN ListingsDatabase: 4. IF MatchesFilter(listing, searchCriteria): 5. matchingListings.ADD(listing) 6. SORT matchingListings BY searchCriteria.sortOption</pre>

7. RETURN matchingListings
8.
9. FUNCTION MatchesFilter(listing, searchCriteria):
10. FOR key, value IN searchCriteria:
11. IF key EXISTS IN listing AND listing[key] != value:
12. RETURN false
13. RETURN true

Table 22: Algorithm for Post Ride/Auto Part

Algorithm 3: Post Your Ride or Auto Part

Input: userID, listingDetails, mediaFiles
Output: Listing Confirmation
1. FUNCTION CreateListing(userID, listingDetails, mediaFiles):
2. IF NOT VerifyUser(userID):
3. RETURN "User Not Verified"
4. newListingID = GenerateUniqueID()
5. listing = {
6. "_id": newListingID,
7. "userID": userID,
8. "details": listingDetails,
9. "media": UploadMedia(mediaFiles),
10. "status": "Active",
11. "createdAt": CURRENT_TIMESTAMP
12. }
13. STORE listing IN ListingsDatabase
14. RETURN "Listing Created Successfully"
15.
16. FUNCTION MarkListingAsSold(listingID):
17. listing = FetchListingFromDatabase(listingID)
18. IF listing EXISTS:
19. listing.status = "Sold"
20. UPDATE listing IN DATABASE
21. RETURN "Listing Marked as Sold"
22. ELSE:
23. RETURN "Listing Not Found"

Table 23: Algorithm for 3D Model Visualization

Algorithm 4: 3D Model Visualization

Input: vehicleID, customizationOptions
Output: Customized 3D Model
1. FUNCTION Render3DModel(vehicleID, customizationOptions):
2. modelPath = Fetch3DModelFromDatabase(vehicleID)
3. customizedModel = ApplyCustomizations(modelPath, customizationOptions)
4. RETURN customizedModel
5.
6. FUNCTION CompareModels(model1ID, model2ID):
7. model1 = Fetch3DModelFromDatabase(model1ID)
8. model2 = Fetch3DModelFromDatabase(model2ID)
9. comparison = GenerateComparison(model1, model2)
10. RETURN comparison

Table 24: Algorithm for Chat and Notification

Algorithm 5: Chat and Notification

Input: senderID, receiverID, messageContent
Output: Message Sent
1. FUNCTION SendMessage(senderID, receiverID, messageContent):
2. chatSession = FetchOrCreateSession(senderID, receiverID)
3. message = {
4. "sessionID": chatSession._id,
5. "senderID": senderID,
6. "receiverID": receiverID,
7. "content": messageContent,

```

8.     "status": "Sent",
9.     "timestamps": CURRENT_TIMESTAMP
10.    }
11.    STORE message IN ChatDatabase
12.    SENDNotification(receiverID, "New Message Received")
13.    RETURN "Message Sent"
14.
15. FUNCTION FetchNotifications(userID):
16.     notifications = GET Notifications WHERE userID = userID
17.     RETURN notifications

```

Table 25: Algorithm for Vehicle Verification**Algorithm 6: Vehicle Verification****Input:** vehicleVIN**Output:** Vehicle Verification Report

```

1. FUNCTION VerifyVehicle(vehicleVIN):
2.     vehicleReport = FetchVehicleHistoryFromAPI(vehicleVIN)
3.     IF vehicleReport == null:
4.         RETURN "Vehicle Verification Failed"
5.     ELSE:
6.         STORE vehicleReport IN VerificationDatabase
7.         RETURN vehicleReport

```

Table 26: Algorithm for Recommendation System**Algorithm 7: Recommendation System****Input:** userID**Output:** Recommended Listings

```

1. FUNCTION GenerateRecommendations(userID):
2.     userPreferences = FetchUserPreferences(userID)
3.     userActivity = FetchUserActivity(userID)
4.     recommendations = ApplyMachineLearningAlgorithm(userPreferences, userActivity)
5.     RETURN recommendations

```

Table 27: Algorithm for Maintenance & Tracking**Algorithm 8: Maintenance and Tracking****Input:** vehicleID, OBD-II Data, expenseDetails**Output:** Health Metrics or Expense Reports

```

1. FUNCTION LogOBDData(vehicleID, obdData):
2.     healthMetrics = AnalyzeOBDData(obdData)
3.     STORE healthMetrics IN MaintenanceDatabase
4.     RETURN "Data Logged Successfully"
5.
6. FUNCTION AddExpense(vehicleID, expenseDetails):
7.     expenseID = GenerateUniqueID()
8.     expense = {
9.         "_id": expenseID,
10.        "vehicleID": vehicleID,
11.        "details": expenseDetails,
12.        "createdAt": CURRENT_TIMESTAMP
13.    }
14.    STORE expense IN ExpensesDatabase
15.    RETURN "Expense Added"

```

7.2 External APIs/SDKs

Table 28: Details of APIs used in the project

Name of API and Version	Description of API	Purpose of Usage	List down the API endpoint/function/class in which it is used
Firebase Authentication	User authentication and authorization	Secure login, registration, and user verification	firebase.auth().createUserWithEmailAndPassword , firebase.auth().signInWithEmailAndPassword
Cloudinary	Image and video management solution	Uploading product images and videos	https://api.cloudinary.com/v1_1/demo/image/upload
Google Maps API	Location and mapping integration	Displaying dealership locations and nearby searches	maps.googleapis.com/maps/api/geocode/json
Vehicle VIN API	Vehicle verification service	Fetching vehicle history reports for verification	/api/vin/verify?vin=<VIN>
Twilio API	SMS and notifications service	Sending SMS notifications to users	twilio.messages.create
SendGrid API	Email delivery service	Sending verification and notification emails	sendgrid.mail.send
OBD-II API	Vehicle diagnostics and performance data	Fetching vehicle performance and maintenance logs	/api/obd/data

7.3 User Interface

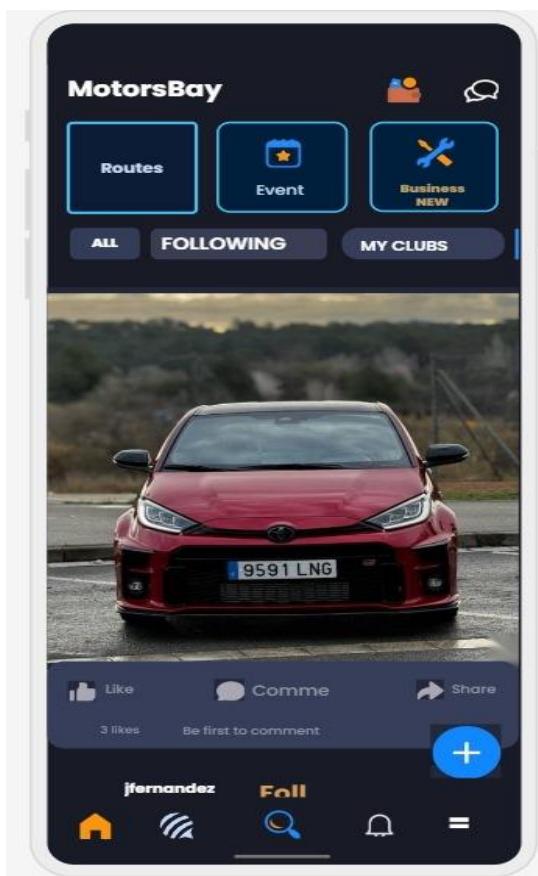


Figure 98:

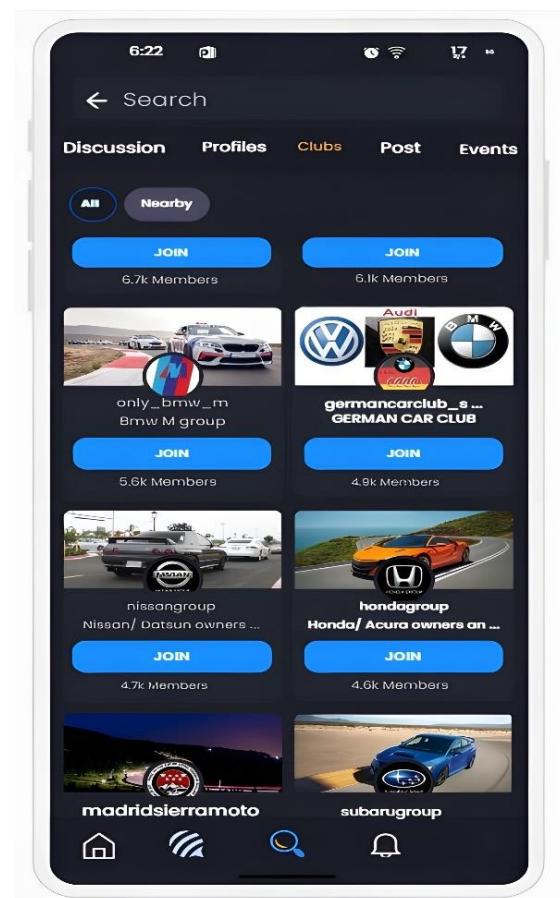


Figure 98

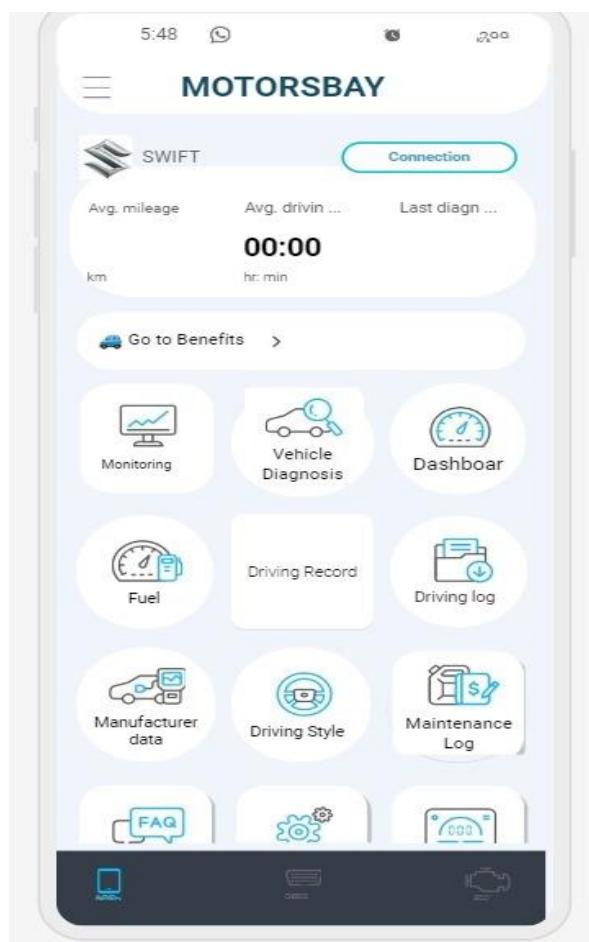


Figure 14.3

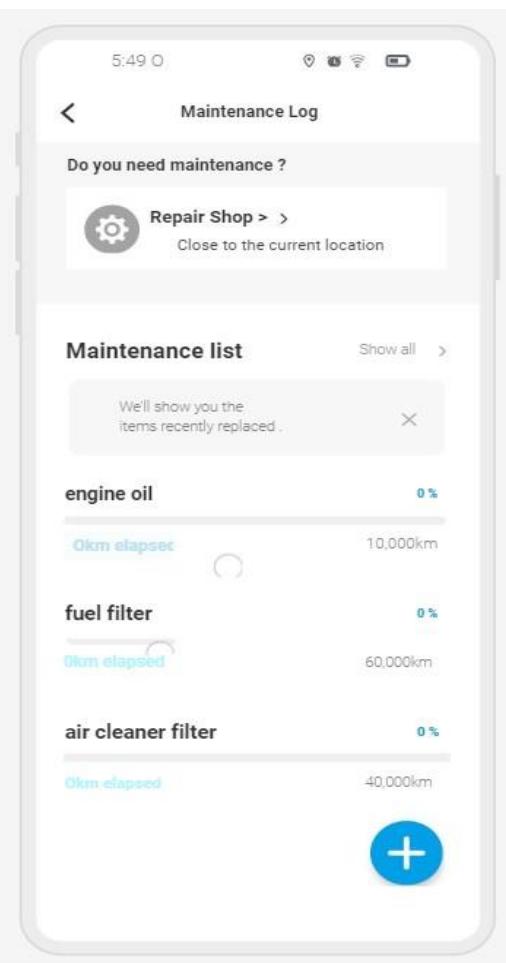


Figure 14.4

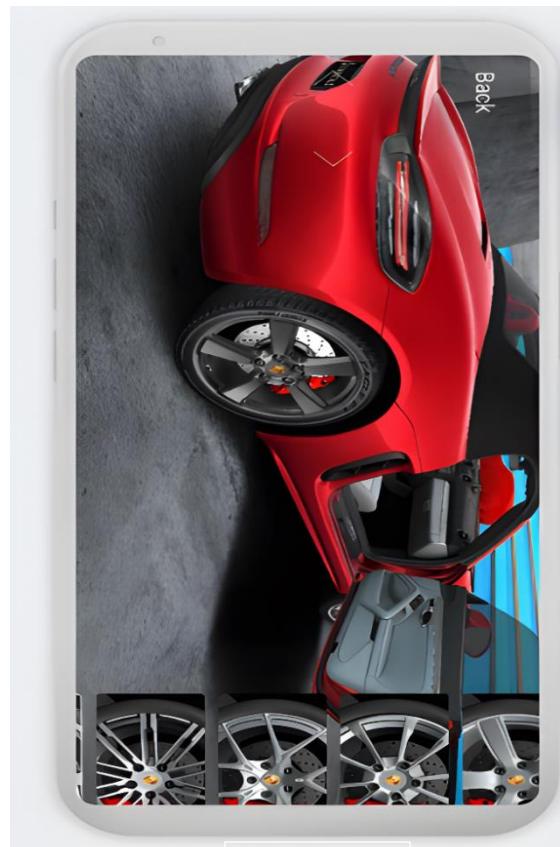


Figure 14.5



Figure 14.6



Figure 14.7



Figure 14.8

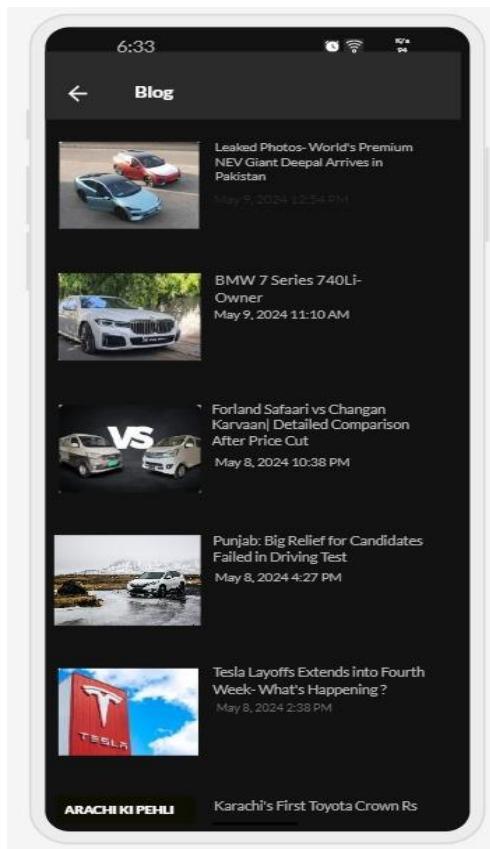


Figure 14.9

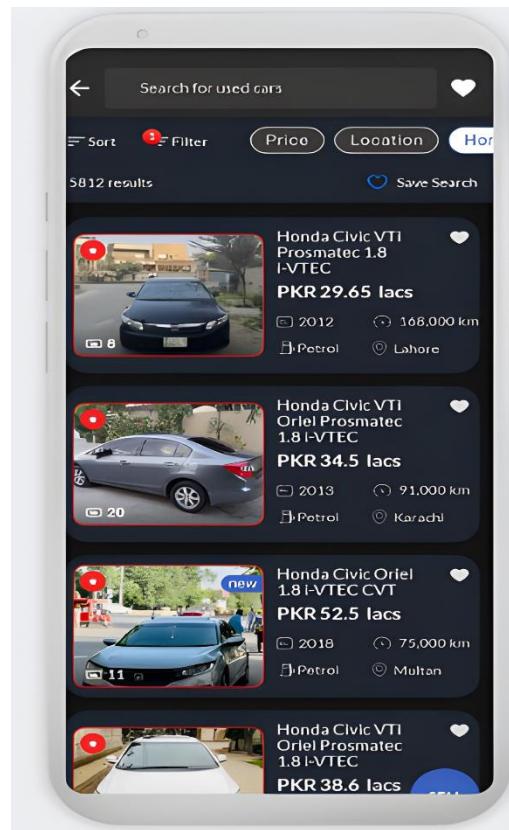


Figure 14.10

7. Testing and Evaluation

This section contains the test cases for the system developed so far. It includes unit tests, functional tests, business rules testing & integration testing.

7.1 Unit Testing

Unit Testing 1: User Management

Testing Objective: Ensure all user authentication, registration, and profile management features work accurately.

Table 29: Unit Test Cases for User Management

No.	Test case/Test script	Attribute and value	Expected result	Result
1	Register with valid details	Email: user@example.com, Password: ValidPass123	User account is created, and user is redirected to dashboard	Pass
2	Register with existing email	Email: existing@example.com, Password: TestPass456	Error message "Email already in use" is displayed	Pass
3	Login with correct credentials	Email: user@example.com, Password: ValidPass123	User logs in successfully and lands on home page	Pass
4	Login with incorrect password	Email: user@example.com, Password: WrongPass321	Error message "Incorrect password" is displayed	Pass
5	Update user profile	Name: John Doe, Contact: 1234567890	User profile is successfully updated	Pass
6	Attempt to update profile while logged out	Action: Update Profile	Error message "Please log in to update profile" is displayed	Pass
7	Delete user account	Status: Logged In	User account is successfully deleted	Pass
8	Admin approves dealership registration request	Status: Pending Request	Dealership profile is approved and visible to other users	Pass

Unit Testing 2: Search and Buy

Testing Objective: Verify search filters, sorting, and watchlist functionalities.

Table 30: Unit Test Cases for Search and Buy

No.	Test case/Test script	Attribute and value	Expected result	Result
1	Search vehicles using valid filters	Filters: Make: Toyota, Year: 2022, Price: < 30k	Vehicles matching the filters are displayed	Pass
2	Search with no matching filters	Filters: Make: Unknown, Year: 2020, Price: < 20k	Message "No results found" is displayed	Pass

3	Add vehicle to watchlist	Vehicle ID: 12345	Vehicle is added to the user's watchlist	Pass
4	Remove vehicle from watchlist	Vehicle ID: 12345	Vehicle is removed from the user's watchlist	Pass
5	Sort search results by price	Sort By: Price Low to High	Vehicles are displayed sorted by increasing price	Pass
6	View vehicle listing details	Listing ID: 54321	Detailed description and images of the vehicle are displayed	Pass

Unit Testing 3: Post Your Ride/Auto Part

Testing Objective: Ensure users can create, update, and manage vehicle/part listings.

Table 31: Unit Test Cases for Post Your Ride or Auto Part

No.	Test case/Test script	Attribute and value	Expected result	Result
1	Create a new vehicle listing	Make: Toyota, Model: Corolla, Year: 2023, Price: 30k	Vehicle listing is created and visible to other users	Pass
2	Upload invalid image format	Image: file.txt	Error message "Invalid file format" is displayed	Pass
3	Edit vehicle listing details	Update: Price = 25k	Vehicle details are updated successfully	Pass
4	Mark a listing as sold	Listing ID: 78901	Vehicle is marked as "Sold"	Pass
5	Delete a vehicle listing	Listing ID: 78901	Vehicle listing is successfully removed	Pass
6	Attempt to create a listing without images	Missing: Images	Error message "Please upload at least one image" is displayed	Pass

Unit Testing 4: Chat and Notification

Testing Objective: Verify the real-time chat system and notification functionalities.

Table 32: Unit Test Cases for Chat and Notification

No.	Test case/Test script	Attribute and value	Expected result	Result
1	Send a message in a private chat	Recipient ID: 12345, Message: "Hello!"	Message is successfully sent and displayed in chat	Pass
2	Attempt to send a blank message	Message: ""	Error message "Message cannot be empty" is displayed	Pass
3	Receive a new message notification	Sender ID: 54321, Message: "Hi!"	User receives a push notification	Pass
4	Save search filters for automatic notifications	Filters: Toyota, Price < 20k	Notifications for matching listings are enabled	Pass
5	Group chat message posting	Group ID: 45678, Message: "Check out this car!"	Message is displayed to all group participants	Pass
6	View notification preferences	Action: View Settings	Notification preferences are displayed and editable	Pass

7	Disable notifications for chat messages	Preference: Turn off Chat Notifications	Chat notifications are disabled successfully	Pass
---	---	---	--	------

Unit Testing 5: Vehicle Verification

Testing Objective: To ensure vehicle verification and inspection features work correctly.

Table 33: Unit Test Cases for Vehicle Verification

No.	Test case/Test script	Attribute and value	Expected result	Result
1	Verify integration with third-party databases to retrieve vehicle history reports.	CNIC: 42101-5678921-9	Vehicle history retrieved successfully.	Pass
2	Ensure vehicle inspection requests can be made for a listing.	Inspection Request: Requested	Inspection requested successfully.	Pass
3	Test driver's license verification for booking or additional security.	Driver's License: Verified	License verification completed.	Pass

7.2 Functional Testing

Functional Testing 1: User Management

Objective: Test the complete process of user account management, including profile creation and updates.

Table 34: Functional Test Cases for User Management

No.	Test case/Test script	Attribute and value	Expected result	Actual result	Result
1	Create a new user account with valid details	Email: user@example.com, Password: ValidPass123	Account is created successfully; user is redirected to login	Account created	Pass
2	Attempt account creation with invalid email	Email: invalidemail, Password: TestPass123	Error message "Invalid email format" is displayed	Error displayed	Pass
3	Update user profile details	Name: John Doe, Contact: 1234567890	Profile details are updated successfully	Changes reflected	Pass
4	Attempt to update profile without login	Action: Update Profile	Error message "Please log in to update profile" is displayed	Error displayed	Pass
5	Delete user account	Status: Logged In	Account is successfully deleted	Account deleted	Pass
6	Admin verifies a dealership account	Dealership ID: 67890, Status: Pending	Dealership account is approved	Approved	Pass

Functional Testing 2: Search and Buy

Objective: Test the search, filter, and sorting functionalities for vehicle and parts listings.

Table 35: Functional Test Cases for Search and Buy

No.	Test case/Test script	Attribute and value	Expected result	Actual result	Result
1	Search for vehicles using valid filters	Filters: Brand: Toyota, Year: 2023, Price < 30k	Vehicles matching the filters are displayed	Vehicles displayed	Pass
2	Search with no matching results	Filters: Brand: Unknown, Price: > 100k	Message "No results found" is displayed	Message displayed	Pass
3	Sort search results by price (low to high)	Sort Option: Price Ascending	Vehicles are sorted by increasing price	Sorted results	Pass
4	View detailed listing information	Listing ID: 12345	Vehicle details, images, and descriptions are displayed	Details displayed	Pass
5	Add a vehicle listing to the watchlist	Listing ID: 56789	Vehicle is added to the user's watchlist	Watchlist updated	Pass
6	Remove a vehicle listing from watchlist	Listing ID: 56789	Vehicle is successfully removed from the watchlist	Watchlist updated	Pass

Functional Testing 3: Post Your Ride/Auto Part

Objective: Verify users can create, edit, and manage vehicle and part listings.

Table 36: Functional Test Cases for Post Your Ride or Auto Part

No.	Test case/Test script	Attribute and value	Expected result	Actual result	Result
1	Create a new vehicle listing	Make: Toyota, Model: Corolla, Year: 2023, Price: 30k	Listing is created and visible to other users	Listing created	Pass
2	Attempt to create a listing without images	Missing: Images	Error message "Please upload at least one image" is displayed	Error displayed	Pass
3	Edit an existing vehicle listing	Update: Price = 25k	Vehicle details are successfully updated	Changes reflected	Pass
4	Mark a vehicle listing as sold	Listing ID: 34567, Status: Sold	Vehicle status is updated to "Sold"	Status updated	Pass
5	Delete a vehicle listing	Listing ID: 34567	Vehicle listing is successfully removed	Listing deleted	Pass
6	Attempt to view a deleted vehicle listing	Listing ID: 34567	Error message "Listing not found" is displayed	Error displayed	Pass

Functional Testing 4: Chat and Notification

Objective: Ensure the chat and notification system works accurately for private and group messaging.

Table 37: Functional Test Cases for Chat and Notification

No.	Test case/Test script	Attribute and value	Expected result	Actual result	Result
1	Send a message in private chat	Recipient ID: 54321, Message: "Hello!"	Message is sent and displayed in the chat	Message sent	Pass
2	Attempt to send an empty message	Message: ""	Error message "Message cannot be empty" is displayed	Error displayed	Pass
3	Receive a new message notification	Sender ID: 12345, Message: "Hi!"	User receives a push notification	Notification sent	Pass
4	Post a message in group chat	Group ID: 999, Message: "Check out this car!"	Message is displayed to all group members	Message posted	Pass
5	Customize notification preferences	Preferences: Disable Chat Notifications	Chat notifications are turned off successfully	Notifications off	Pass
6	Enable saved search notifications	Search Criteria: Toyota, Price < 30k	Notifications for matching listings are enabled	Notifications on	Pass

7.3 Business Rules Testing

7.3.1 Business Rules Testing

BR-1: The email address and phone number provided by the user must be unique in the system.

BR-2: The password must be at least eight characters long and include both alphabets and digits, and the new password must meet these criteria.

BR-3: Only registered users can log in, request password recovery, and change passwords by entering the current password correctly.

BR-4: Only one account can be registered for a given username, and deleted accounts cannot be restored.

BR-5: Dealership accounts must be verified by the system before activation, and only verified dealerships can manage inventory.

BR-6: All profile updates and listings (vehicles or parts) must pass validation checks before being saved, and listings must include essential specifications and clear images.

BR-7: Only Admins with required permissions can access the dashboard and manage user accounts.

BR-8: Services can only be booked during the available working hours of the Service Provider.

BR-9: Sorting and filter options must be relevant to the criteria provided by the system and the type of search conducted (e.g., vehicle filters for vehicle searches).

BR-10: Each user can have multiple listings in their watchlist.

BR-11: Search results must reflect the exact criteria entered by the Buyer.

BR-12: Only supported file formats (e.g., JPG, PNG, MP4) can be uploaded.

BR-13: Sellers and Dealerships can mark their listings as sold.

BR-14: Only supported listings with a 3D model will offer customization and comparison features, which must be based on valid vehicle specifications and remain private unless shared.

BR-15: Messaging and communication must comply with platform policies.

BR-16: Only Buyers who have used the service can submit ratings and reviews, and reviews must comply with platform guidelines. Only valid, non-fraudulent reviews can be posted, and violations should be reported.

BR-17: Notifications must be sent in real-time, unless disabled by the user, and users must be able to enable or disable notifications at their discretion.

BR-18: Only verified vehicles can display history or inspection details, and only certified inspection services can be used.

BR-19: Only valid licenses are accepted for verification.

BR-20: Content, including blogs, posts, and shared materials, must comply with platform guidelines and be relevant to cars and vehicles.

BR-21: Only users with verified accounts can post questions, answers, or participate in clubs and groups, which must follow platform rules.

BR-22: Recommendations, including vehicle recommendations, must be accurate and tailored to user preferences.

BR-23: Vehicle health data and expense tracking must be accurate, with reports categorized for better management.

BR-24: Service reminders should include timely notifications

BR-25: The chatbot must provide accurate and helpful responses, and live support must be available during designated hours.

BR-26: Only verified Service Providers can list car-related services

BR-27: The filter options must be relevant to the review data

BR-28: Fuel prices must be accurate and frequently updated

Module 1: User Management

Table 38: Business Rules Test Cases for User Management

Module 2: Search and Buy

Table 39: Business Rules Test Cases for Search & Buy

Module 3: Post Your Ride or Auto Part

Table 40: Business Rules Test Cases for Post Ride/Auto Part

ion checks																								
BR-13: Mark listings as sold	T	T	T	T	T	F	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
BR-14: Listings with valid 3D model	T	T	T	T	T	F	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Module 4: Chat and Notification

Table 41: Business Rules Test Cases for Chat & Notification

Condition/Rule	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
BR-15: Messaging compliance	T	T	F	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
BR-17: Notifications in real-time	T	T	T	F	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

7.4 Integration Testing

Integration Testing 1: User Management & Search and Buy

Testing Objective: To ensure that user management is integrated correctly with search and buy features.

Table 42: Business Rules Test Cases for User Management & Search and Buy

No.	Test case/Test script	Attribute and value	Expected result	Actual result	Result
1	Verify that registered users can log in and access search and buy features without issues.	User credentials: valid_email@gmail.com, valid_password	User logged in successfully, access to features granted.	User logged in successfully.	Pass
2	Ensure that when a user adds a vehicle to their watchlist, it's reflected in their	Vehicle ID: #12345	Vehicle added to watchlist.	Vehicle added to watchlist.	Pass

	profile under "Saved Listings".				
3	Test that user-specific search filters are saved and applied correctly after logging in and returning to the search page.	Filters: Make: Toyota, Model: Camry	Filters applied successfully upon returning.	Filters applied successfully.	Pass
4	Ensure that when a user updates their profile (e.g., location), the search results update to reflect relevant nearby listings.	Updated Location: New York, NY	Nearby listings updated based on location.	Nearby listings updated successfully.	Pass

Integration Testing 2: User Management & Post Your Ride

Testing Objective: To ensure that user management is integrated correctly with the posting of rides.

Table 43: Business Rules Test Cases for User Management & Post Ride

No.	Test case/Test script	Attribute and value	Expected result	Actual result	Result
1	Verify that users can successfully post listings after registering or logging in.	Listing: 2019 Honda Civic	Listing posted successfully.	Listing posted successfully.	Pass
2	Ensure that a seller's posted vehicle or part is correctly linked to their user profile.	Seller ID: User123	Vehicle linked to user profile.	Vehicle linked to user profile.	Pass
3	Test the interaction between user profile updates (e.g., name, dealership info) and their active listings.	Updated Name: New Dealership Name	Name reflected on active listings.	Name updated successfully.	Pass
4	Verify that when a dealership user adds a new vehicle, the listing is visible to other users in the search module.	New Vehicle: 2020 Ford F-150	Vehicle visible in search results.	Vehicle visible in search results.	Pass

Integration Testing 3: Search and Buy & Chat and Notification

Testing Objective: To ensure the search and buy features are integrated with chat and notification functionalities.

Table 44: Business Rules Test Cases for Search and Buy & Chat and Notification

No.	Test case/Test script	Attribute and value	Expected result	Actual result	Result
1	Ensure that once a user sends a message to a seller regarding a listing, a notification is triggered on the seller's side.	Message: Is this car still available?	Notification sent to seller.	Notification sent successfully.	Pass
2	Verify that saved search notifications (for price drops, new listings) are sent to users in real-time.	Notification Type: Price Drop	Notification received by user.	Notification received successfully.	Pass
3	Test that when a buyer and seller initiate a chat about a specific listing, the vehicle details are reflected in the chat window.	Chat initiated: Ford Mustang inquiry	Vehicle details shown in chat.	Vehicle details displayed correctly.	Pass
4	Ensure that push notifications for offers or messages are sent to both buyer and seller in the app and work correctly in real-time.	Push Notification: New offer received	Notifications sent successfully.	Notifications sent successfully.	Pass

Integration Testing 4: Vehicle Verification & Search and Buy

Testing Objective: To ensure the vehicle verification process is integrated with search results.

Table 45: Business Rules Test Cases for Search and Buy & Vehicle Verification

No.	Test case/Test script	Attribute and value	Expected result	Actual result	Result
1	Ensure that the vehicle verification process (VIN check) is integrated correctly with the search results and is visible on the listing page for verified vehicles.	VIN: 1HGCM82633A123456	Verification status visible on listing.	Status displayed correctly.	Pass
2	Test that users can request a vehicle inspection from the search module, and the inspection status is updated on the listing page.	Inspection Request: Requested	Inspection status updated.	Status updated successfully.	Pass

3	Verify that when a vehicle passes verification, the "Verified" badge is displayed on the listing page and can be filtered in searches.	Vehicle ID: #12345	"Verified" badge displayed on listing.	Badge displayed successfully.	Pass
---	--	--------------------	--	-------------------------------	------

8. Plagiarism

Report.docx

ORIGINALITY REPORT

4%
SIMILARITY INDEX

3%
INTERNET SOURCES

0%
PUBLICATIONS

4%
STUDENT PAPERS

PRIMARY SOURCES

1 Submitted to Lindenwood University **2%**
Student Paper

2 academic-master.com **2%**
Internet Source

3 Submitted to Higher Education Commission
Pakistan **1%**
Student Paper

Exclude quotes On
Exclude bibliography On

Exclude matches Off