Software Requirements Specification

for

Turbo Torque

**Version 1.0 approved  
Deliverable 2**

**Prepared by   
  
Syed Arshaq Hussain Kirmani | 22i-0834  
Ali Asif | 22i-1019  
Muhammad Taha | 22i-0870**

**15/03/25**

**Table of Contents**

**Table of Contents ii**

**Revision History ii**

**1.** **Introduction 1**

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Product Scope 1

1.5 References 1

**2.** **Overall Description 2**

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Classes and Characteristics 2

2.4 Operating Environment 2

2.5 Design and Implementation Constraints 2

2.6 User Documentation 2

2.7 Assumptions and Dependencies 3

**3.** **External Interface Requirements 3**

3.1 User Interfaces 3

3.2 Hardware Interfaces 3

3.3 Software Interfaces 3

3.4 Communications Interfaces 3

**4.** **System Features 4**

4.1 System Feature 1 4

4.2 System Feature 2 (and so on) 4

**5.** **Other Nonfunctional Requirements 4**

5.1 Performance Requirements 4

5.2 Safety Requirements 5

5.3 Security Requirements 5

5.4 Software Quality Attributes 5

5.5 Business Rules 5

**6.** **Diagrams 5**

6.1 Use Case Diagram 5

6.2 Activity Diagram 5

6.3 Sequence Diagram 6

**Appendix A: Glossary 6**

**Appendix B: Analysis Models 6**

**Appendix C: To Be Determined List 6**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Ali Asif | 5/3/25 | Initial draft of SRS | 1.0 |
| Arshaq Kirmani | 7/3/25 | Updated functional requirements | 1.1 |
| Muhammad Taha | 10/3//25 | Updated non-functional requirements | 1.2 |
| Ali Asif | 15/3/25 | Finalized SRS document | 2.0 |

# Introduction

## Purpose

This document specifies the software requirements for **Turbo Torque**, a comprehensive online car marketplace. The current revision of the system is **Version 2.0**.

Turbo Torque enables users to **buy, sell, and rent** cars from individuals, dealerships, and companies. The platform also features newly launched vehicles for user reference. The system is managed by an **admin** who oversees user activities and advertisements. This SRS covers the **core functionalities of the entire platform**, including user management, car listings, transactions, and administrative controls.

## Document Conventions

This Software Requirements Specification (SRS) document follows standard conventions to ensure clarity and consistency. The conventions used are as follows:

**1.2.1 Font Style & Size:**

* + Headings: **Bold, Size 14 (e.g., Section Titles)**
  + Subheadings: **Bold, Size 12**
  + Body Text: **Regular, Size 11**
  + Code Snippets: **Ubuntu Sans Mono Font**

**1.2.2 Requirement Prioritization:**

* + High-priority requirements are marked as **[High]**
  + Medium-priority requirements are marked as **[Medium]**
  + Low-priority requirements are marked as **[Low]**

**1.2.3 Numbering & Referencing:**

* + Sections follow a hierarchical numbering format (e.g., 1.1, 1.1.1).

**1.2.4 Terminology & Notations:**

* + Key technical terms are **italicized** when first introduced.
  + User roles such as **Admin**, **Buyer**, **Seller**, **Company**, **Dealer** are capitalized for distinction.
  + Mandatory requirements use **"shall"**, while optional features use **"may"** or **"should"**.

## Intended Audience and Reading Suggestions

### 1.3.1 Intended Audience

This SRS is intended for:

* **Developers & Testers** – For implementation and validation of system requirements.
* **Project Managers & Business Analysts** – To track scope and ensure business alignment.
* **UI/UX Designers** – To design interfaces based on system features.
* **Marketing Team & End Users** – To understand platform capabilities.

### 1.3.2 Document Structure & Reading Guide

1. **Introduction** – Overview of the system, including purpose, scope, conventions, audience, and references.  
   * **Recommended for:** All readers for initial context.
2. **Overall Description** – High-level system perspective, key functions, user roles, and constraints.  
   * **Recommended for:** Project managers, business analysts, and developers.
3. **External Interface Requirements** – Details on user, hardware, software, and communication interfaces.  
   * **Recommended for:** Developers, testers, and UI/UX designers.
4. **System Features** – Breakdown of core functionalities with feature descriptions.  
   * **Recommended for:** Developers, testers, and business analysts.
5. **Other Nonfunctional Requirements** – Performance, security, safety, and business rules.  
   * **Recommended for:** Developers, testers, and project managers.
6. **Diagrams** – Visual representations including use case, activity, and sequence diagrams.  
   * **Recommended for:** Developers, designers, and analysts.

**1.3.3 Reading Order**

* **All Readers:** Start with **Introduction**.
* **Developers & Testers:** Focus on **Requirements & Features**.
* **Project Managers & Analysts:** Review **Overall Description**.
* **UI/UX Designers & Marketing:** Refer to **System Features**.

## Product Scope

**Turbo Torque** is an online car marketplace that enables users to **buy, sell, and rent** vehicles from individuals, dealerships, and companies. The platform also showcases **newly launched cars**, providing a comprehensive solution for car enthusiasts and buyers.

### 1.4.1 Purpose & Objectives

* **Streamline Vehicle Transactions** – Facilitate easy and secure buying, selling, and renting of cars.
* **Enhance User Experience** – Provide a seamless and intuitive interface for all user roles.
* **Expand Market Reach** – Connect individual buyers, sellers, and dealers on a single platform.
* **Increase Revenue Opportunities** – Support advertisements and premium listings for businesses.
* **Improve Trust & Security** – Implement verification processes and secure payment options.

### 1.4.2 Business Alignment

Turbo Torque aligns with corporate goals by leveraging **digital transformation** in the automotive market, driving growth through **customer engagement, monetization of listings, and partnerships with dealerships**. If a separate vision and scope document exists, please refer to it for further details.

## References

This SRS refers to the following external resources:

1. **GitHub Repository** – Contains source code, version control, and development updates.  
   * **URL:** <https://github.com/aliasif78/pak-wheels/tree/main>
2. **Trello Board** – Tracks project tasks, milestones, and development progress.  
   * **URL:** <https://trello.com/b/cpwqFTm3/turbo-torque>

# Overall Description

## Product Perspective

### 2.1.1 Product Perspective

**Turbo Torque** is a **new, self-contained** online car marketplace designed to facilitate **buying, selling, and renting** of vehicles among individuals, dealers, and companies. It is not a follow-on product or a replacement for an existing system but is developed from the ground up to streamline vehicle transactions.

### 2.1.2 System Context

Turbo Torque operates as an independent platform with the following key components and interfaces:

* **User Interface (Web)** – Allows buyers, sellers, and dealers to interact with the platform.
* **Admin Panel** – Enables the management of users, ads, and platform activities.
* **Database** – Stores user profiles, vehicle listings, transaction history, and advertisements.
* **External APIs** – Integrates authentication gateways, vehicle verification services, and third-party data sources.

## Product Functions

Turbo Torque provides the following major functionalities:

* **User Registration & Management** – Users can sign up, log in, and manage profiles.
* **Car Listings** – Sellers and dealers can list vehicles with details, images, and pricing.
* **Buying & Selling** – Users can browse, negotiate, and purchase vehicles.
* **Car Rentals** – Users can rent vehicles for specified durations from other users or companies.
* **New Car Showcase** – Displays newly launched cars for user reference.
* **Admin Controls** – Admin manages users, listings, and advertisements.
* **Search & Filtering** – Users can find vehicles using various filters like price, brand, location, etc.
* **User Reviews & Ratings** – Buyers and renters can leave feedback for sellers and rental providers.

## User Classes and Characteristics

Turbo Torque is designed for multiple user classes, each with distinct roles and privileges:

1. **Buyers**
   * **Usage Frequency:** Moderate to high
   * **Functions Used:** Browse listings, search/filter cars, negotiate, purchase, and leave reviews.
   * **Technical Expertise:** Basic to moderate; general consumers.
   * **Importance:** High – Ensuring a smooth buying experience is crucial for platform success.
2. **Sellers (Individuals & Dealers)**
   * **Usage Frequency:** Moderate to high
   * **Functions Used:** Create and manage listings, set prices, respond to inquiries, and promote listings.
   * **Technical Expertise:** Basic to moderate; some familiarity with online selling.
   * **Importance:** High – The platform's inventory and success depend on active sellers.
3. **Renters & Rental Providers** (Users or Companies)  
   * **Usage Frequency:** Moderate
   * **Functions Used:** List rental cars, set pricing and availability, manage bookings, and process payments.
   * **Technical Expertise:** Basic to moderate.
   * **Importance:** Medium – Expands the marketplace but secondary to buying/selling.
4. **Admin**
   * **Usage Frequency:** High
   * **Functions Used:** Manage user accounts, review and approve listings, monitor transactions, handle disputes, and oversee advertisements.
   * **Technical Expertise:** Advanced; requires platform management knowledge.
   * **Importance:** Critical – Ensures platform security, compliance, and smooth operation.
5. **Guest Users**
   * **Usage Frequency:** Low to moderate
   * **Functions Used:** Browse listings, view new car launches, and register for an account.
   * **Technical Expertise:** Basic; general web users.
   * **Importance:** Medium – Encouraging conversion to registered users is a priority.

## Operating Environment

### *Operating Environment*

Turbo Torque operates in the following technical environment:

* **Frontend Framework:** Next.js (built with TypeScript)
* **Backend:** Server-side functions within Next.js that is built upon node js
* **Database:** MongoDB
* **Authentication & Security:** OAuth for user authentication, reCAPTCHA for bot protection
* **Hosting & Deployment:** Vercel
* **Supported Platforms:** Web browsers (Chrome, Firefox, Safari, Edge)
* **Third-Party Integrations:** Various APIs for payments, vehicle data, and user verification

## Design and Implementation Constraints

The development of **Turbo Torque** is subject to the following constraints:

* **Technology Stack:** Must use **Next.js (TypeScript)** for frontend and backend, **MongoDB** for database, and be deployed on **Vercel**.
* **Authentication & Security:** OAuth must be used for user authentication, and reCAPTCHA must be implemented for bot protection.
* **Hosting Limitations:** Since the platform is deployed on **Vercel**, backend processes must align with **serverless functions**, avoiding long-running background tasks.
* **Database Constraints:** MongoDB should be optimized for scalability and performance, ensuring efficient query execution.
* **Third-Party Integrations:** Payment processing, vehicle verification, and other external APIs must be integrated seamlessly.
* **Performance Considerations:** Must ensure fast load times, responsive UI, and optimized serverless API calls.
* **Scalability:** The system should support high concurrent traffic and dynamic content without affecting performance.

## User Documentation

The following documentation will be provided with **Turbo Torque**:

* **User Manual** – Guides for buyers, sellers, and renters on platform usage.
* **Admin Guide** – Instructions for managing users, listings, and advertisements.
* **API Documentation** – For integrating third-party services (if applicable).
* **Online Help & FAQs** – Built-in support with common troubleshooting steps.
* **Video Tutorials** – Short guides for key platform features.

Documentation will be available in **PDF, web-based help, and video formats** for easy access.

## Assumptions and Dependencies

#### Assumptions

* Users will have a **stable internet connection** for optimal platform performance.
* The platform will be accessed primarily via **modern web browsers** (Chrome, Firefox, Edge, Safari).
* Third-party APIs for **payments, vehicle verification, and authentication (OAuth)** will remain available and functional.
* Vercel’s serverless environment will support **scalability and performance** needs without major limitations.
* MongoDB will provide **efficient data storage and retrieval** for large-scale listings and user transactions.

#### Dependencies

* **Third-Party Services:** Payment gateways, OAuth authentication, reCAPTCHA, and external car data sources.
* **Hosting Provider:** Vercel’s infrastructure for deployment and backend API execution.
* **Database Management:** MongoDB for handling all user and listing data.
* **Security Measures:** Compliance with **GDPR** and other data privacy regulations.

# External Interface Requirements

## User Interfaces

1. The platform will have a **modern, responsive web UI** built using **Next.js and TypeScript**.

**Standard UI Components**:

* **Navigation Bar** – Provides quick access to key sections (Buy, Sell, Rent, New Cars, Profile).
* **Search & Filters** – Allows users to refine car listings based on price, brand, location, etc.
* **Listing Pages** – Displays vehicle details with images, pricing, and seller/rental information.
* **User Dashboard** – Personalized interface for managing listings, transactions, and account settings.
* **Admin Panel** – Controls for user management, ad approvals, and site moderation.

1. **Error Handling** – Clear, user-friendly error messages with tooltips and notifications.
2. **Standard Elements** – Common buttons (Submit, Cancel, Edit, Delete), forms, and interactive UI elements.

## Hardware Interfaces

Turbo Torque is a **web-based platform**, requiring no direct hardware integration.

Supports **desktops, laptops, tablets, and mobile devices** with a compatible web browser.

Works with standard **input devices** (keyboard, mouse, touchscreen).

## Software Interfaces

* **Database:** MongoDB for storing user data, car listings, transactions, and reviews.
* **Authentication:** OAuth for secure login (Google, Facebook, etc.).
* **Security:** reCAPTCHA for bot protection and fraud prevention.
* **Hosting:** Vercel for deployment, using **serverless functions** for backend logic.
* **External APIs:  
    
  1. Payment Gateway API** – Secure transactions for purchases and rentals.  
  **2. Vehicle Data API** – Fetch car details, specifications, and pricing insights.

## Communications Interfaces

* The platform will use **HTTPS** for secure communication.
* **Email Notifications** – Order confirmations, account updates, and promotional messages.
* **WebSockets or Polling** – For real-time updates (e.g., new messages, price drops).
* **Standard Protocols** – RESTful APIs for data exchange, WebRTC (if live chat is implemented).
* **Encryption & Security** – User data transmission will be encrypted using **TLS/SSL**.

# System Features

## Car Listing and Selling

4.1.1 Description and Priority

This feature allows users (individuals, dealers, and companies) to list cars for sale or rent. Users can provide details such as price, model, year, and images.

**Priority:** High

4.1.2 Stimulus/Response Sequences

* User selects "Sell a Car" or "Rent a Car."
* System prompts for car details, pricing, and images.
* User submits the listing.
* System verifies details and publishes the listing.

4.1.3 Functional Requirements

**REQ-1:** The system shall allow users to list cars with required details (make, model, year, price, condition, etc.).

**REQ-2:** Users shall be able to upload multiple images of the car.

**REQ-3:** The admin shall have the ability to review and approve listings before publishing.

**REQ-4:** The system shall support editing and deleting listings.

## Car Buying and Renting

4.1.1 Description and Priority

Users can browse, filter, and buy/rent cars listed on the platform.

**Priority:** High

4.1.2 Stimulus/Response Sequences

* Buyer searches for a car using filters.
* System displays matching results.
* Buyer selects a car and contacts the seller or initiates the purchase.
* System processes payment and finalizes the transaction.

4.1.3 Functional Requirements***REQ-5:*** *Users shall be able to search and filter listings by make, model, price, and location.****REQ-6:*** *The system shall allow users to contact sellers through a built-in messaging feature.****REQ-7:*** *Payment processing shall be integrated for online transactions.*

**4.3 User Management**

4.1.1 Description and Priority

User registration, authentication, and profile management.

**Priority:** High

4.1.2 Stimulus/Response Sequences

* User registers or logs in via email/password or OAuth (Google, Facebook, etc.).
* System authenticates and redirects to the dashboard.
* Users can update their profile, view transactions, and manage listings.

4.1.3 Functional Requirements***REQ-8:*** *Users shall be able to register and log in via email or OAuth.****REQ-9:*** *The system shall provide password reset functionality.****REQ-10:*** *Admins shall be able to manage user roles and permissions.*

**4.4 Advertisement and Promotion**

4.1.1 Description and Priority

Users can promote their car listings through paid ads to increase visibility.

**Priority:** Medium

4.1.2 Stimulus/Response Sequences

* Seller selects a listing to promote.
* System provides promotion options (featured listing, highlighted ad, etc.).
* Seller makes payment, and the listing gets boosted.

4.1.3 Functional Requirements***REQ-11:*** *The system shall allow users to purchase advertisement packages.****REQ-12:*** *Admin shall manage advertisement pricing and placements.****REQ-13:*** *Promoted listings shall be displayed with higher visibility.*

# Other Nonfunctional Requirements

## Performance Requirements

* The platform should load within **2 seconds** on high-speed internet and within **5 seconds** on slower connections.
* Search and filter queries should return results in **under 1 second**.
* The system should support **at least 100,000 concurrent users** without degradation.
* Image uploads and processing should complete within **3 seconds**.

## Safety Requirements

* Prevent fraudulent listings and transactions through **identity verification** and fraud detection.
* Secure user data to avoid data leaks and comply with **GDPR** and other privacy regulations.
* Implement **automated monitoring** to detect suspicious activities, such as scams or misleading listings.

## Security Requirements

* **OAuth authentication** required for user login to prevent unauthorized access.
* All data should be transmitted over **HTTPS (TLS/SSL encryption)** to protect user privacy.
* **Role-based access control (RBAC):** Admins manage users and ads, while regular users only modify their own listings.
* **Two-factor authentication (2FA)** for high-risk operations like transactions and account recovery.
* Secure payment gateway integration with **PCI-DSS compliance**.

## Software Quality Attributes

* **Availability:** 99.9% uptime to ensure reliable access.
* **Scalability:** Auto-scaling on Vercel to handle high traffic.
* **Maintainability:** Modular codebase with clean architecture for easy updates.
* **Usability:** Responsive UI that works smoothly on both desktop and mobile.
* **Interoperability:** Support for third-party APIs (e.g., payment, car data).

## Business Rules

* Only **verified users** can list cars for sale or rent.
* Admins have authority to **approve, reject, or remove** listings and ads.
* **Refund and dispute policies** apply for rental transactions.
* Listings should include **valid vehicle details**; spam or misleading ads will be removed.

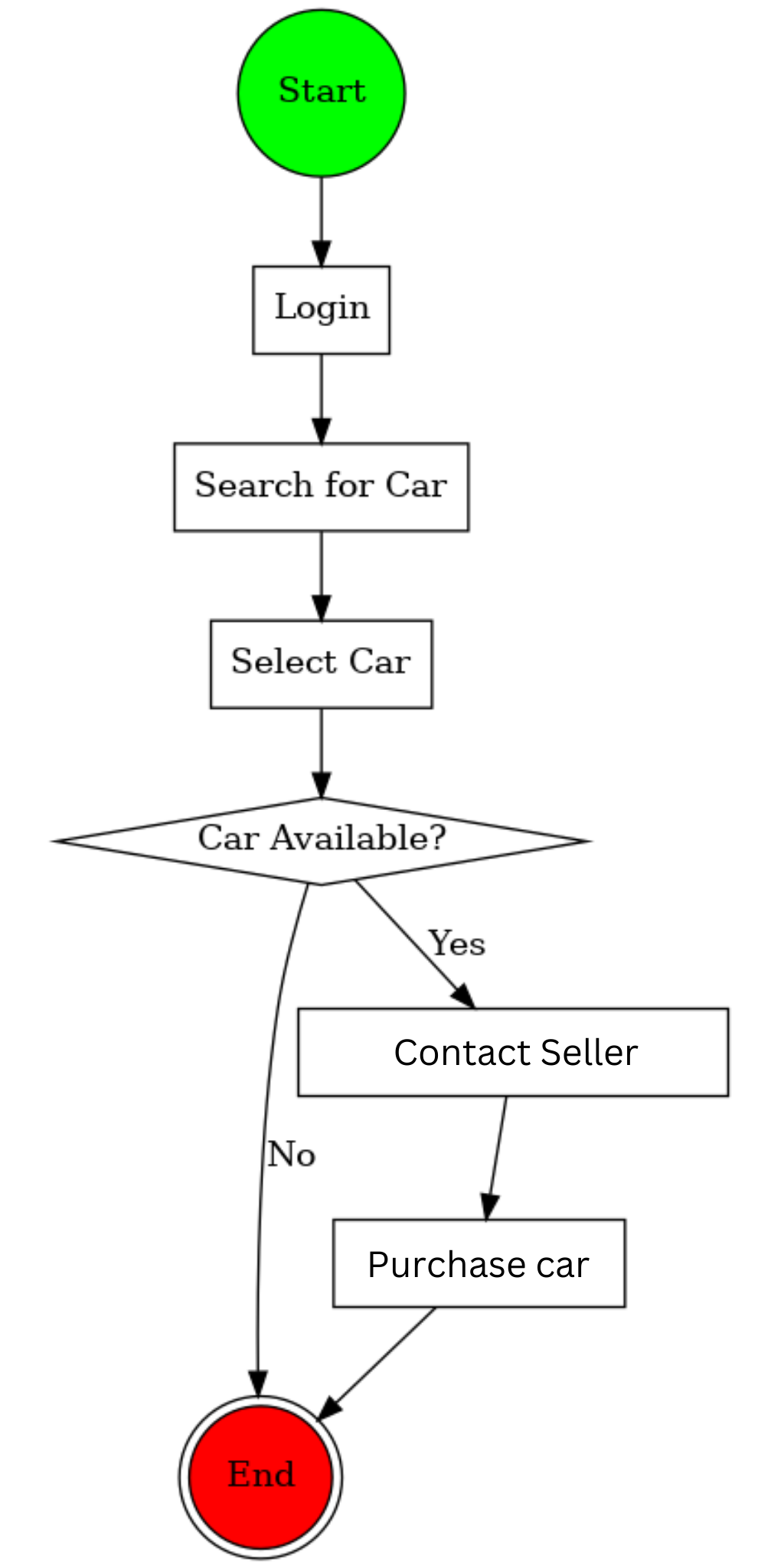
# Diagrams

*<Include visual representations that help convey the structure, behavior, and interactions within the software system.*

## Use Case Diagram



## Activity Diagram



## Sequence Diagram 6.3.1 Rent Car



## 6.3.1 Purchase Car 6.3.1 Add newly launched car

**Appendix A: Glossary**

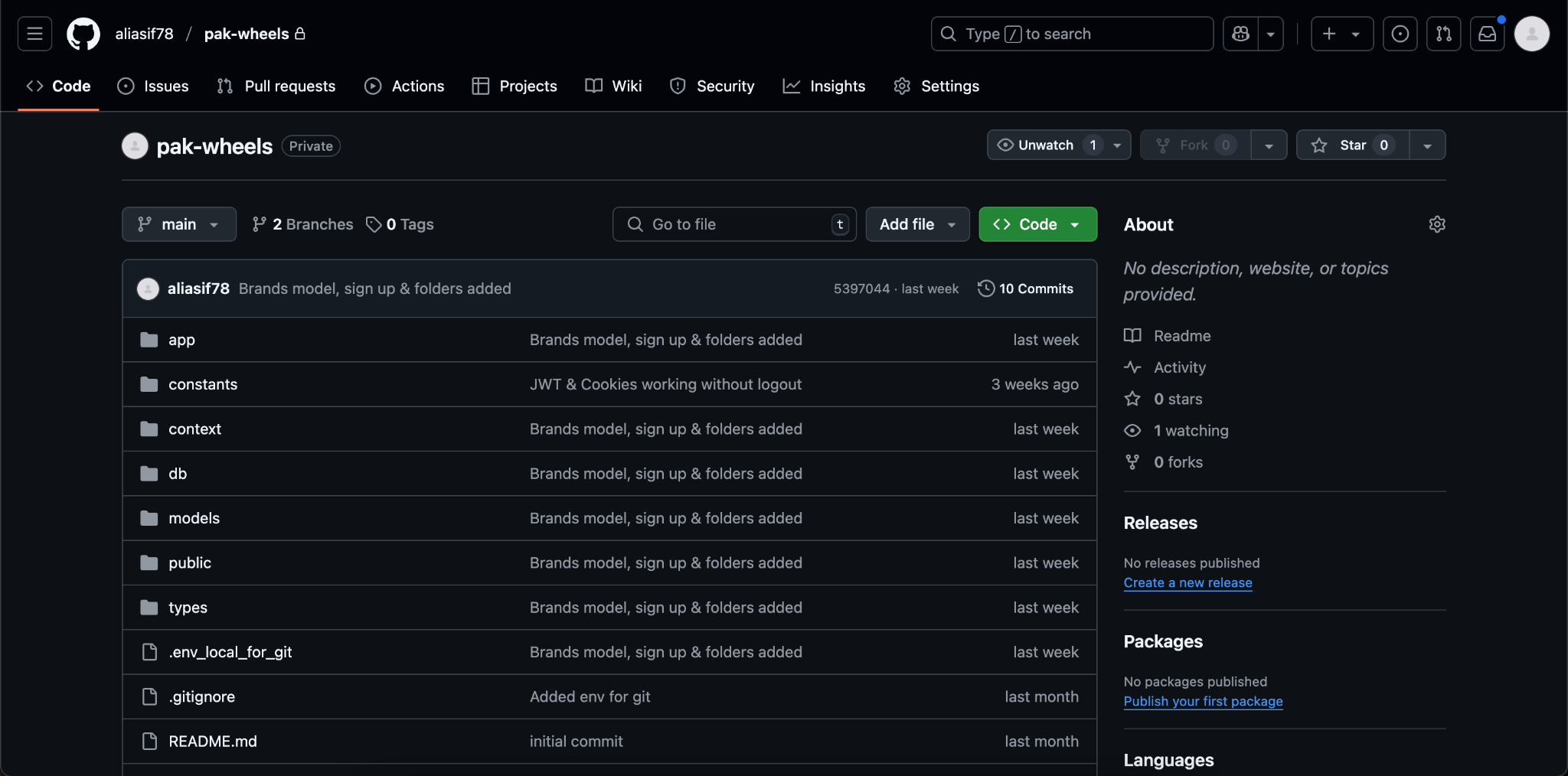
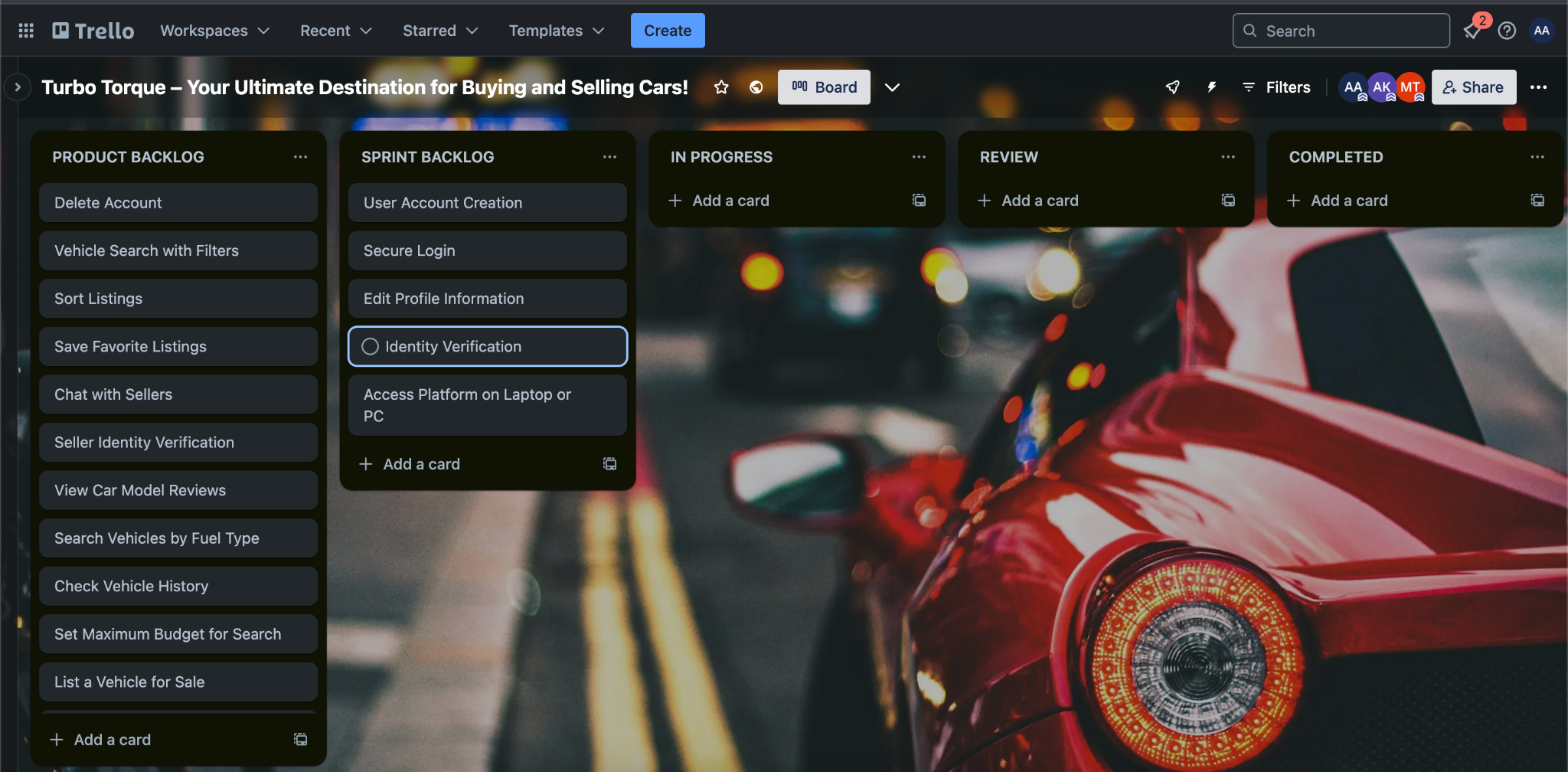
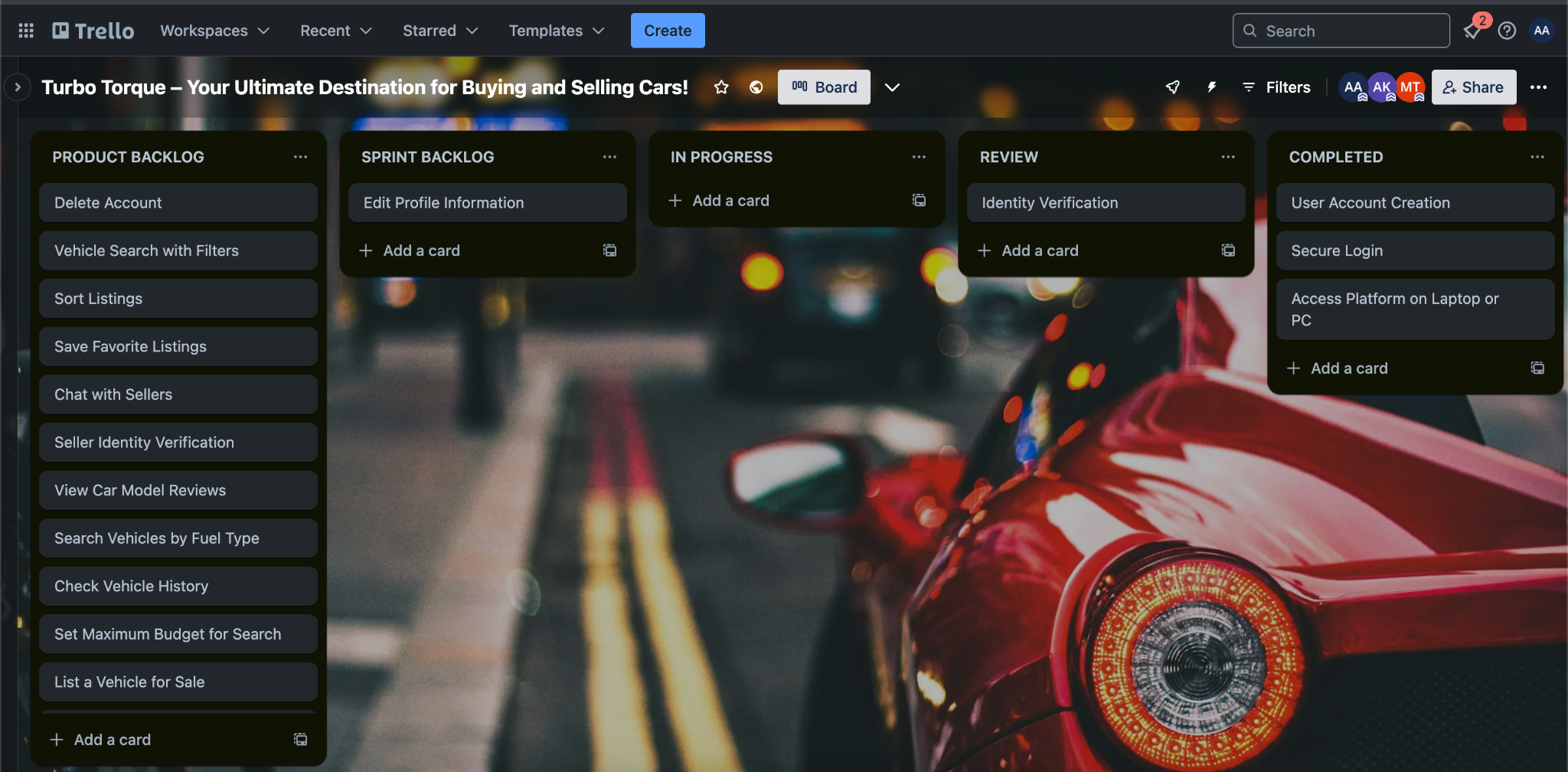
1. **Turbo Torque** – A car marketplace for buying, selling, and renting vehicles.
2. **Admin** – The user responsible for managing platform operations, user accounts, and advertisements.
3. **Seller** – A user or dealer who lists a car for sale or rent.
4. **Buyer** – A user who purchases or rents a car through the platform.
5. **OAuth** – A protocol used for secure authentication and authorization.
6. **MongoDB** – The NoSQL database used to store platform data.
7. **Vercel** – The deployment platform for Turbo Torque.
8. **ReCAPTCHA** – A security measure used to prevent bots from misusing the platform.

**Appendix B: Analysis Models**

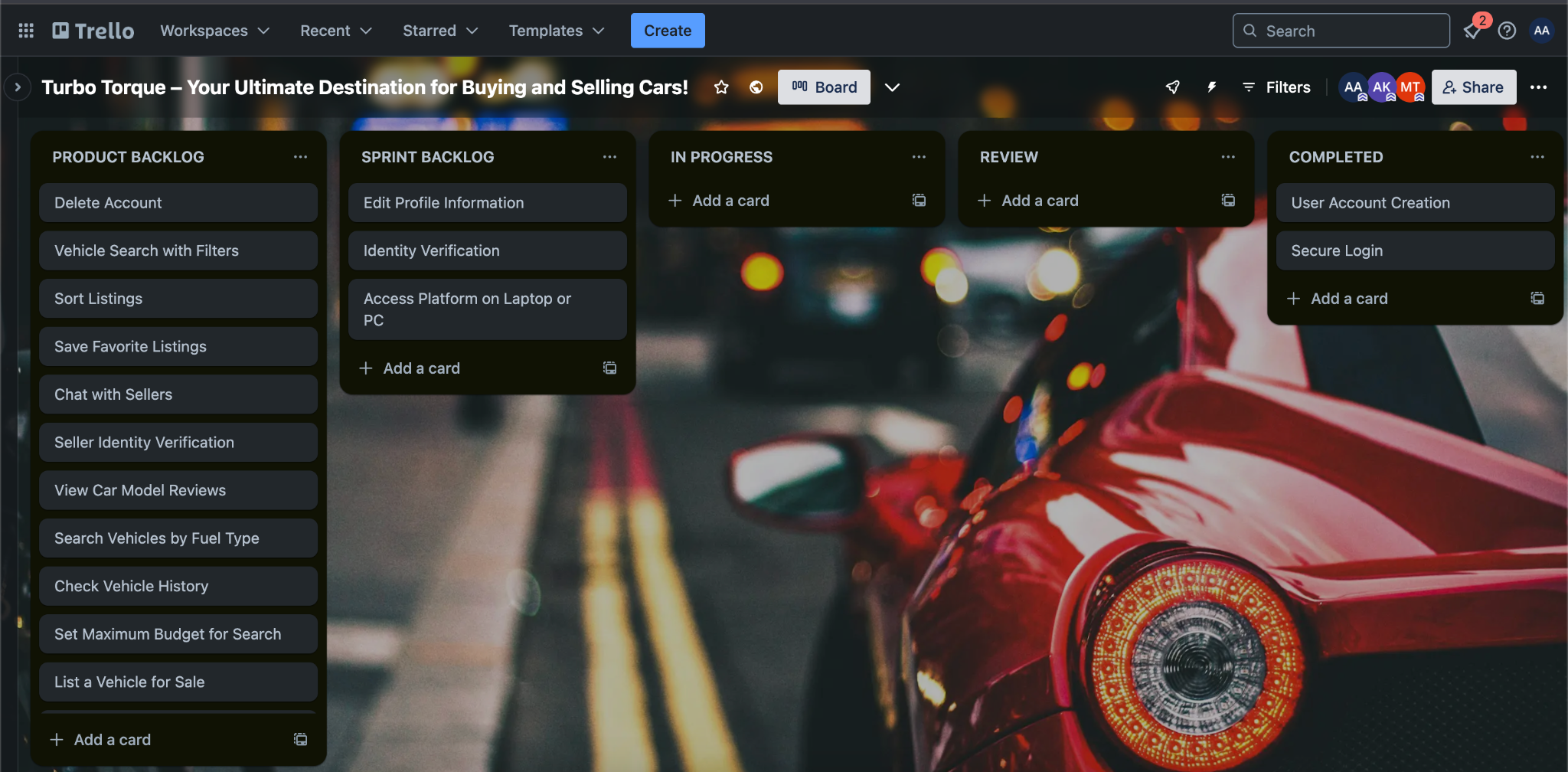
1. **Use Case Diagram** – Illustrates interactions between users and the system.
2. **Activity Diagram** – Describes the workflow of key processes like buying, selling, and renting cars.
3. **Sequence Diagram** – Shows step-by-step interactions in transactions such as car purchases.
4. **Entity-Relationship Diagram (ERD)** – Defines the data relationships between users, vehicles, transactions, and other entities.

**Appendix C: To Be Determined List**

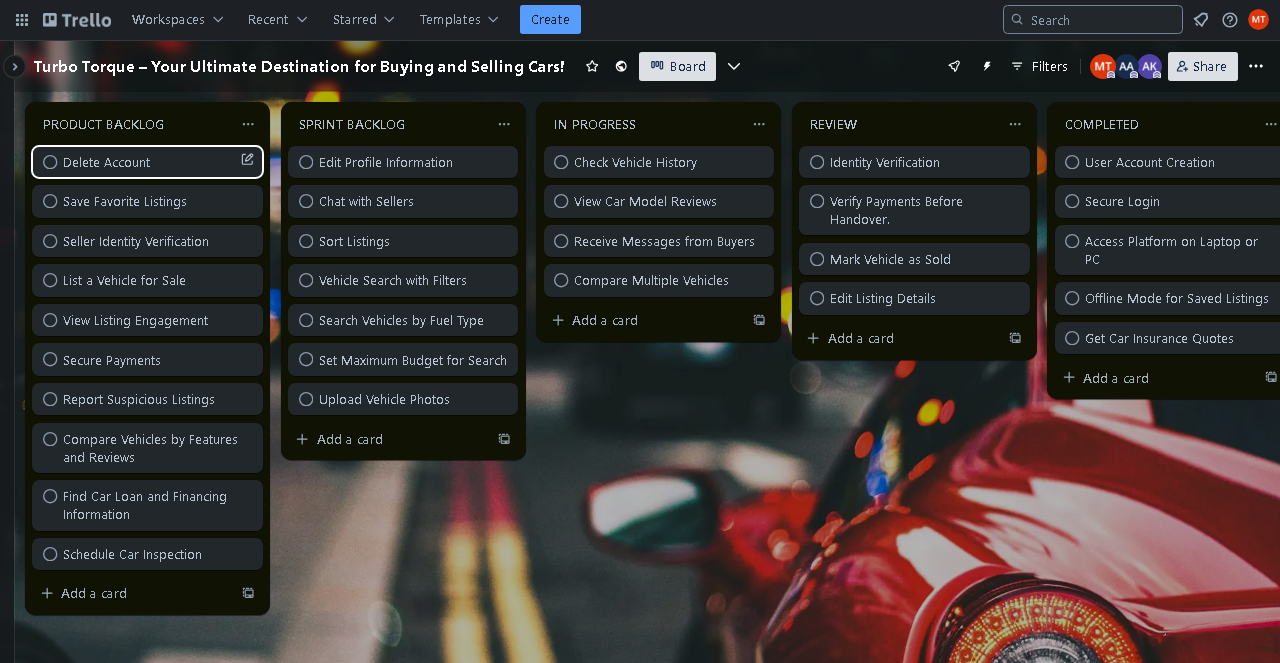
1. **Exact commission structure for transactions** – Needs finalization.
2. **Third-party API integrations** – Selection of additional services like payment gateways or verification tools.
3. **Legal compliance** – Regulatory requirements specific to different regions for car sales and rentals.
4. **Scalability plans** – Server and database scaling strategies for increased traffic.

***Updated GitHub*** **[*https://github.com/aliasif78/pak-wheels/tree/main*](https://github.com/aliasif78/pak-wheels/tree/main)***Snapshot-1********Snapshot-2***

***Snapshot-3***

******

***Snapshot-4***



**TurboTorque Testing:**

**Black-Box Testing:**

**Equivalence Class Partitioning:**

| **Field** | **Equivalence Classes (EC)** | **Test Case ID** |  |
| --- | --- | --- | --- |
| Full Name | EC1: Valid name (letters + spaces) | TC1 |  |
|  | EC2: Contains digits or symbols (invalid) | TC2 |  |
|  |  |  |  |
| Email | EC3: Correct format (user@domain.com) | TC3 |  |
|  | EC4: Missing @ or domain part | TC4 |  |
|  |  |  |  |
| Phone Number | EC5: Valid 10‑12 digits | TC5 |  |
|  | EC6: Includes letters or < 10 digits | TC6 |  |
|  |  |  |  |
| Password | EC7: Meets complexity (≥8 chars, letters + numbers) | TC7 |  |
|  | EC8: Too short (<8 chars) | TC8 |  |
|  |  |  |  |
| Re-enter Password | EC9: Matches Password | TC9 |  |
|  | EC10: Does not match | TC10 |  |
|  |  |  |  |
| Car Year | EC11: Within valid range (1886–2025) | TC11 |  |
|  | EC12: Below 1886 or above 2025 | TC12 |  |
|  |  |  |  |
| Price | EC13: Positive number | TC13 |  |
|  | EC14: Zero or negative | TC14 |  |
|  |  |  |  |
| Brand Name | EC15: Non-empty string | TC15 |  |
|  | EC16: Empty string | TC16 |  |
|  |  |  |  |

* Validates "Email" field for correct format (user@domain.com) and missing parts (e.g., missing @ or domain).
* Ensures "Password" meets complexity requirements (≥8 characters with letters and numbers).

**Boundary Value Analysis**

| **Field** | **Boundary Values** | **Test Inputs** | **Test Case ID** |
| --- | --- | --- | --- |
| Phone length | 10 (min), 12 (max), 9, 13 | 10, 12, 9, 13, 11 | BVA1–BVA4 |
| Password length | 8 (min), 20 (max), 7, 21 | 8, 20, 7, 21, 13 | BVA5–BVA8 |
| Car Year | 1886 (min), 2025 (max), 1885, 2026 | 1886, 2025, 1885, 2026, 1950 | BVA9–BVA12 |
| Price | 1 (min valid), 0, -1 | 1, 0, -1, 50000 | BVA13–BVA15 |
|  |  |  |  |

* Phone Length: Validates edge cases with minimum (10), maximum (12), and out-of-range values (9, 13).
* Password Length: Ensures system handles minimum (8), maximum (20), and invalid lengths (7, 21).
* Car Year Range: Confirms input is within valid bounds (1886–2025) and detects outliers (1885, 2026).
* Price Validation: Checks for minimum valid price (1), invalid zero, and negative values.

**Validation Against User Stories**

Map each test case to functional requirements/user stories:

| **Test Case** | **Feature** | **User Story** |
| --- | --- | --- |
| TC1–TC10 | User Registration Form | As a user, I want to create an account using my email or phone number so that  I can access all features. |
| TC11–TC14 | Car Addition/Delete | As a seller, I want to list my vehicle by providing details like make, model,  price, and condition so that I can find a buyer. |
| TC15–TC16 | Brand Management | As admin, I can manage (add/delete) brands. |
| BVA# | Edge behavior | Shift from basic to boundary validation. |

### Detailed Test Cases

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Precondition** | **Test Steps** | **Expected Result** |
| TC1 | Register with valid full name and email | User on registration page | 1. Enter "Muhammad Taha" in Full Name  2. Enter [muhammadtaha@gmail.com](mailto:muhammadtaha@gmail.com)  3. Fill other valid fields  4. Click Continue | Account is created  Redirect to dashboard  Success message shown |
| TC2 | Register with invalid characters in full name | User on registration page | 1. Enter "Name123" in Full Name  2. Fill other valid fields  3. Click Continue | Error under Full Name: "Only letters and spaces allowed" |
| TC4 | Register with malformed email | User on registration page | 1. Enter valid name  2. Enter "email.com" in Email  3. Complete other valid fields  4. Click Continue | Error under Email: "Invalid email format" |
| TC5 | Register with short phone number | User on registration page | 1. Enter valid name and email  2. Enter "123456789" in Phone Number  3. Complete other valid fields  4. Click Continue | Error under Phone: "Phone must be 10–12 digits" |
| TC7 | Register with weak password (<8 chars) | User on registration page | 1. Enter valid name, email, phone  2. Enter "Pass1" in Password  3. Re-enter "Pass1"  4. Click Continue | Error under Password: "Password must be at least 8 characters" |
| TC9 | Register with non-matching re-entered password | User on registration page | 1. Enter valid name, email, phone  2. Enter "Password123" in Password  3. Enter "Password321" in Re-enter  4. Click Continue | Error under Re-enter Password: "Passwords do not match" |
| TC11 | Add car with valid year and price | Logged in as seller | 1. Navigate to Add Car  2. Enter valid name  3. Enter 2020 for Year and 15000 for Price  4. Submit | Car appears in listings; success message shown |
| TC12 | Add car with out-of-range year | Logged in as seller | 1. Navigate to Add Car  2. Enter valid name  3. Enter 1880 for Year  4. Submit | Error under Year: "Year must be between 1886 and current year" |
| TC13 | Add car with negative price | Logged in as seller | 1. Navigate to Add Car  2. Enter valid details, Price = -50003. Submit | Error under Price: "Price must be positive" |
| TC15 | Add brand with empty name | Logged in as admin | 1. Navigate to Brand Management  2. Leave Name blank  3. Click Add Brand | Error under Brand Name: "Name is required" |

**White-Box Testing**

