

**Faculty of Science Department of Computer Science**

****

**Senior Project - I (CMPS 443) [CarFinder Hub]**

**Student Name: Student ID:**

[Abdal Rahman Marquise] [202303792]

[Ahmad Harmoush] [202306048]

[Maher Mezher] [202306017]

[Mohammad Ali Absi] [202403256]

**Supervisor:**

[Dr May Itani] Principal Supervisor

**March 2025**

**Acknowledgements**

**Abstract**

**Table of Contents**

[Chapter 1. Introduction 7](#_bookmark0)

1. [Main Project Description 7](#_bookmark1)
2. [Problem Statement 7](#_bookmark2)
3. [Project Goal 8](#_bookmark3)
4. [Project Stakeholders 8](#_bookmark4)
5. System and Domain Review

[Chapter 2. Feasibility Study 8](#_bookmark5)

1. [Operational Feasibility 9](#_bookmark6)

[Market Feasibility 9](#_bookmark7)

[Legal Feasibility 12](#_bookmark8)

[Social Feasibility 12](#_bookmark9)

[Environmental Feasibility 12](#_bookmark10)

1. [Economic and Financial Feasibility 13](#_bookmark11)
2. [Technical Feasibility 14](#_bookmark12)

[Chapter 3. Project Plan 16](#_bookmark13)

1. [SDLC Model 16](#_bookmark14)
2. [Project Schedule 16](#_bookmark15)
3. [Team Roles 17](#_bookmark16)
4. [Budget and Cost Estimation 17](#_bookmark17)
5. [Project Monitoring and Control 19](#_bookmark18)

[Risk Management 19](#_bookmark19)

[Quality Assurance 19](#_bookmark20)

[People and Time 20](#_bookmark21)

[Chapter 4. Software Requirement Specification 21](#_bookmark22)

1. [Product Functions 21](#_bookmark23)
2. [User Permissions and Types 21](#_bookmark24)
3. [Functional Requirements 22](#_bookmark25)
4. [Non-functional Requirements 23](#_bookmark26)
5. [Domain Requirements 24](#_bookmark27)

[Chapter 5. Project Design 26](#_bookmark28)

1. [User Interface Prototype 27](#_bookmark29)
2. [Use Case Diagram 38](#_bookmark30)
3. [Data Flow Diagram 43](#_bookmark31)
4. [Database Diagram 44](#_bookmark32)
5. [Sequence Diagrams 49](#_bookmark34)

[Chapter 6. Implementation Plan 50](#_bookmark36)

1. [Technical Resources 50](#_bookmark37)
2. [Testing Plan 50](#_bookmark38)
3. [Deployment Plan 50](#_bookmark39)
4. [Maintenance Plan 50](#_bookmark40)

[Chapter 7. Conclusion and Future Work 51](#_bookmark41)

[Chapter 8. References 52](#_bookmark42)

# Chapter 1. Introduction

## Main Project Description

What is the project about?

CarFinder Hub allows users to compare cars based on their preferences (budget, type, mode, etc.). It integrates multiple car company and multiple car apps databases to provide the users with the best possible options. Users can also sell cars, view offers, and review sellers/cars with a star rating system. A unique feature is the ability to view the interior/exterior of the car and watch videos of the car when it was published.

- Why is it important?

Many people struggle to find the right car that fits their budget and preferences. Our app simplifies this process by providing a centralized platform for car comparison, buying, and selling

## Problem Statement

What problem are you solving?

* 1. Difficulty in finding car within a budget.
  2. Lack of centralized platform for comparing cars from multiple brands.
  3. Limited access to detailed car information (interior/exterior, videos, offers).
  4. Inability to review sellers and cars before making a purchase.
  5. Fragmented car market data.
  6. Inability to visualize cars effectively.
  7. Lack of reliable seller and car reviews.
  8. Limited support for Eco-Friendly options.

## Project Goal

Our goal is to develop an intuitive platform where users can easily compare cars from different brands and dealerships based on their preferences, including price, model, and location. Seamlessly connect with various car databases to provide real-time data and a wide selection of options. Enhance decision making by offering detailed specifications, images, videos, and customer reviews.

Enable both buying and selling, allowing users to list their vehicles while maintaining a reliable marketplace. Implement a transparent rating system for cars and sellers to build trust. Provide advanced filtering options, ensuring users can refine their search based on specific criteria like fuel type and transmission.

Ensure all listings are verified for accuracy, reducing misinformation. Introduce unique features such as virtual tours, AI-driven suggestions, and live price comparisons to enhance user experience. Build a system capable of handling high traffic efficiently while maintaining smooth performance. Design with scalability in mind to accommodate future growth, expanding to new regions and integrating advanced technologies like AI-powered analytics.

## Project Stakeholders

#### Who will use the app?

#### Customers: People looking to buy or sell cars.

#### Car Dealers/Sellers: Businesses or individuals selling cars.

#### Car Manufactures: Companies Whose cars are listed on the platform.

#### Society: People who benefit from a more transparent and efficient car-buying process.

**E. System and Domain Review**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Platform | Real-Time Price Updates | Multi-Dealer  Integration | Virtual Car Tours | User Reviews | Selling Feature |
| CarDekho | Yes | Limited | No | Yes | No |
| Edmunds | Yes | Yes | No | Yes | No |
| Dubizzle | Manual Updates | No | No | No | Yes |
| Facebook MarketPlace | Manual Updates | No | No | No | Yes |
| Carwale | Yes | Limited | No | Yes | No |
| CarFinder Hub | Yes | Yes | Yes | Yes | Yes |

# Chapter 2. Feasibility Study

## Operational Feasibility

***System and Domain Review:***

#### There are many car platforms available today. Here are some well-known ones:

#### Car Dekho :

#### \_Pros: provides detailed car comparisons, user reviews, and financing options.

#### \_Cons: Weak seller verification, which can lead to fake listings. The website also has too many ads.

* + Edmunds :

\_Pros: Offers expert reviews, price comparisons, and market trends.

\_Cons: Doesn’t integrate used car listings well and lacks virtual car tours.

* + Car Wale:

\_Pros: Focuses on the Indian market with local price comparisons and EMI calculators

\_Cons: Limited global support and no virtual car tour feature.

* + Kelley Blue Book (KBB)

\_Pros: Trusted car price valuations for new and used cars.

\_Cons: Doesn’t allow direct buying/selling and lacks detailed video reviews.

How Our System is Better: Our CarFinder Hub platform improves on these existing platforms by offering:

1. Verified Listings: Every car listing goes through verification to prevent fraud.
2. Virtual Car Tours: Users can see 360 views of car interiors and exteriors.
3. Real Time Data: Integrates with multiple car databases for accurate prices and availability
4. Detailed Reviews: Users can rate and review both sellers and cars.
5. Buy & Sell Marketplace: People can list their own cars for sale, not just dealerships.
6. Eco-Friendly Options: Highlights electric and hybrid cars for green buyers.

### Market Feasibility:

We spoke with 46 people to understand if our CarFinder app would really fit into their lives and the response was very encouraging. Most of them said they often struggle with switching between different websites just to compare cars, and they would love a simple app that brings everything together. People told us they want easy search options, clear side-by-side comparisons, and updates when prices change. Overall, it became clear that not only do users want this app, but it would genuinely make the car- buying process easier and less stressful for them.

#### Here are the most Important results that we had during our study about the needs of people who are looking for cars.

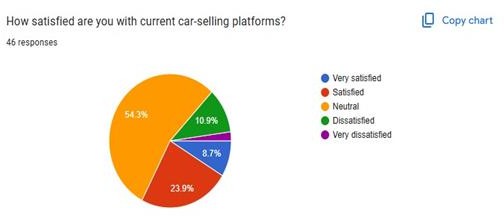


Figure Satisfied People

The survey results from 46 participants show that 54.3% feel neutral about current car-selling platforms, suggesting there’s room for improvement. 23.9% reported being satisfied, while 10.9% are dissatisfied, and 1 person (2.2%) is very dissatisfied. Only 8.7% said they are very satisfied. Overall, these results highlight that many users are not fully happy with the existing platforms, reinforcing the need for a more user-friendly and efficient car comparison solution

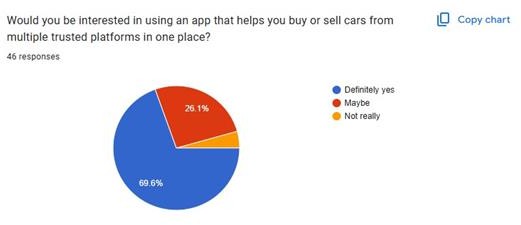


Figure Interested People

When we asked people if they’d be interested in using an app that lets them buy or sell cars from multiple trusted platforms in one place, the response was super positive. Nearly 70% said "definitely yes" —

they’re clearly looking for an easier, more streamlined way to shop for cars. Another 26% said "maybe", showing curiosity and potential interest. Only a small 4% weren’t really into the idea. Overall, it’s clear that most people would welcome a simple, all-in-one car comparison tool that saves them time and effort

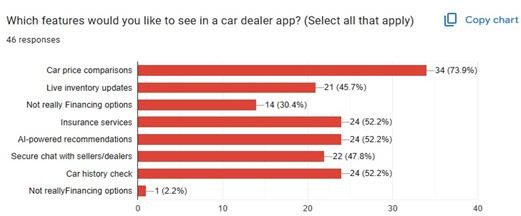


Figure 3 What Features Do They Like

We asked people what features they’d love to see in a car dealer app — and their answers gave us great insight. The most wanted feature by far was car price comparisons, with nearly 74% of respondents selecting it. People also really care about having AI-powered recommendations, insurance services, and a car history check — each chosen by about 52% of participants. Features like secure chat with dealers (47.8%) and live inventory updates (45.7%) were also popular, showing that trust and up-to-date info really matter. Financing options were less of a priority, with only 30% finding them important. This feedback gives us a clear picture of what users need most in the app — simple comparisons, trustworthy info, and smart suggestions.

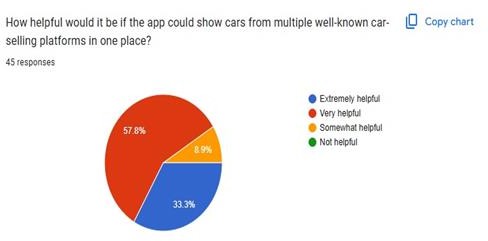


Figure How Helpful Would it be

The survey results show that most users are really excited about the idea of a CarFinder Hub app. Over half of the respondents (57.8%) think it would be extremely helpful to see cars from multiple popular car- selling platforms all in one place. Another 33.3% said it would be very helpful. Only a small group (8.9%) felt it would be somewhat helpful, and another 33.3% didn’t find it helpful at all. This feedback tells us that a lot of people would love a single app to compare cars easily, making the car-buying process simpler and more convenient. It’s clear that there’s a strong demand for this kind of tool, and it could really make a difference for users looking to find their perfect car.

### Legal Feasibility

* 1. Data privacy & security: Protect user’s personal information (e.g., names, contact

details, payment info) from misuse, unauthorized access, and data breaches.

* 1. Intellectual property rights: Avoiding copyright or trademark violations when using third-party content, APIs, or software.
  2. Consumer protection laws: Ensuring fair practices in selling, advertising, and providing accurate information about cars.
  3. Contacts & agreements: Complying with legal agreements when integrating third-party APIS or databases from car manufactures.
  4. Fraud prevention & liability: Implementing legal measures to prevent scams, misleading listings, and unauthorized transactions.

### Social Feasibility

1. No Discrimination: The platform will be accessible to everyone, and search results will be unbiased
2. Fraud Prevention: AI-based fraud detection will help remove fake listings, and users will get tips on safe transactions.
3. Honest Reviews: Reviews will be moderated to prevent fake ratings and misleading information.
4. Eco-Friendly Focus: We will highlight electric and hybrid cars to encourage sustainable choices.

### Environmental Feasibility

1. Reducing Carbon Footprint: Virtual car tours and online comparisons help minimize unnecessary travel
2. Less Paper Waste: Digitizing documents instead of relying on printed materials.
3. Promoting Green Awareness: Educating users on fuel-efficient cars and sustainable driving habits.

## Economic and Financial Feasibility

1. Estimated Budget Breakdown:
   1. Development Costs: $10,000-$25,000
      * Web & mobile app development
      * API integration with car databases
      * UI/UX design
   2. Hosting & Infrastructure: $5,000-$10,000 per year
      * Cloud hosting (AWS, Google Cloud, or Azure)
      * Data storage & security
   3. Marketing & Advertising: $3,000-$8,000 per year
      * Social media & search engine ads
      * Influencer & partnership marketing
   4. Operational Costs: $2,000-$5,000
      * Customer support
      * Maintenance & updates
   5. Legal & Compliance: $2,000-$5,000
      * Data protection compliance
      * Business registration & legal fees
2. Will it make economic sense?

Since it’s a digital platform, expenses are lower than physical dealerships. The platform can

expand by adding more brands, dealers, and regions. We can generate income through:

* Premium Listings: charging sellers for featured ads.
* Advertisements: Earning from car brands and dealers.
* Transaction fees: Charging a small commission on car sales.
* Subscription plans: Offering premium features for advanced car comparisons.

1. Will it generate profits?

Yes! With the right marketing and user growth, the platform can be profitable within 1-2 years. The key factors include:

* Attracting dealerships and private sellers to list their cars.
* Increasing user traffic to drive ad revenue.
* Expanding to multiple markets over time.

## Technical Feasibility

Our Car Finder Hub Platform requires a combination of modern technologies to ensure performance, scalability, and security

1. Frontend (User Interface):
   * Flutter / React Native (For mobile app)
   * Tailwind CSS (For styling and responsive design)
2. Backend (Server & APIs):
   * Node.js with Express.js or Django (Python) for API handling
   * GraphQL / REST API for seamless data communication
3. Database Management:
   * PostgreSQL / MySQL (for structured car listings & user data)
   * MongoDB (for flexible data storage, like reviews & preferences)
4. Cloud Hosting & Infrastructure
   * AWS / Google Cloud / Firebase for hosting & scalability
   * Cloudinary (for storing and serving car images/videos)
5. Third-Party API Integrations:
   * Car Manufacture APIs (for real-time car data)
   * Google Maps API (for location-based searches)
   * Payment Gateway (Stripe / PayPal) (for premium listings & services)
6. Security & Authentication
   * OAuth 2.0 / Firebase Authentication (for user logins)
   * SSL Encryption (to protect user data & transactions)
   * AI-powered fraud detection (to prevent scams & fake listings)

# Chapter 3. Project Plan

## SDLC Model

#### After careful consideration and planning and assessing the project's complexity and risks, we decided that the Scrum format will be the most suitable for me and my team. Our decision came down to multiple factors which are:

#### Simplicity: scrum has a simple and straightforward framework which defines everyone's role and responsibilities clearly. This makes it helpful for me and my team since we are newcomers to this space.

#### Iterative Approach: Scrum's iterative approach allows teams to break down work into smaller, manageable increments. Which means we as a team can add or change any feature that comes to our mind if we find it suitable.

#### Continuous Improvement: Scrum emphasizes continuous improvement through regular retrospectives. This encourages teams to reflect on their work, identify areas for improvement, and make adjustments in subsequent sprints.

#### Time Management and Planning: Scrum's time-boxed iterations (sprints) provide a structured framework for managing time and setting achievable goals within specific times. This can help with project planning and ensure steady progress throughout the project duration.

## Project Schedule

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Phase | Task | Assigned To | Duration | Milestone |
| Week 1-2 | Project Planning & Research | Abed Markiz (PM) | 2 weeks | Project kickoff, tech stack  finalized |
| Week 3-4 | UI/UX  Wireframing & Prototyping | Maher & Mohmad | 2 weeks | Low & High-  fidelity designs completed |
| Week 5-6 | Database Design & API Planning | Ahmad (Backend) Abed Markiz | 2 weeks | Database schema  & API structure finalized |
| Week 7-8 | User Authentication (Signup/Login) | Ahmad (Backend), Maher  & Mohmad (Frontend) | 2 weeks | User authentication completed |
| Week 9-10 | Car Listing & Search Features | Ahmad (Backend), Maher  & Mohmad (Frontend) | 2 weeks | Listings & search functionality |
| Week 11-12 | AI Price  Comparison & Suggestions | Ahmad (Backend) Abed Markiz | 2 weeks | AI model for price  comparison implemented |
| Week 13-14 | Payment & Secure Transactions | Ahmad (Backend), Maher  & Mohmad (Frontend) | 2 weeks | Secure payment integration completed |
| Week 15 | Testing & Bug  Fixing | All Members | 1 week | Beta Version  Ready |
| Week 16 | Deployment & Launch | All Members | 1 week | Final release & marketing |

## Team Roles

|  |  |
| --- | --- |
| Name | Role |
| Abdal Rahman Marquise | * Oversees project timeline and team coordination. * Ensures Agile principles are followed. * Manages meetings, sprint planning, and backlog prioritization. * Facilitates communication between team members. |
| Maher Mezher Mohamad Halabi | **FRONT-END**   * Develops the UI/UX of the mobile application using Flutter/Dart * Implements user interactions, search filters, and listing features. * Integrates APIs for real-time car price comparison and dealer listings. * Works on responsiveness and accessibility. |
| Ahmad Harmoush Abdal Rahman Marquise | **BACK-END**   * Develops and manages the server-side logic using Firebase * Designs and manages the database (Firebase). * Implements authentication, data processing, and AI-based price suggestions. * Ensures data security and performance optimization. |
| Maher Mezher Mohamad Halabi Ahmad Harmoush | **QA Tester & UI/UX Designer**   * Creates wireframes and ensures an intuitive app design. * Conducts usability testing and gathers user feedback. * Performs manual and automated testing for bug fixes. * Ensures app performance and compatibility across different devices. |

## Budget and Cost Estimation

The budget estimation for the development and operation of our app is critical in determining whether the project will be economically viable.

So based on the economic and financial feasibility

* + The total estimated budget:
    - Low estimate: $22,000
    - High estimate: $53,000

This budget reflects the full development, launch, and operational costs for the first year including hosting, marketing, and legal compliance.

## Project Monitoring and Control

### Risk Management

Risk management is essential for the success of our Car Comparison app. A major risk is the integration of multiple car APIs from various manufacturers and dealerships. To mitigate this. We will begin the integration early and thoroughly test it to ensure smooth data synchronization.

User trust is another concern, especially with buying and selling cars. To address this, we will implement a robust verification process for both buyers and sellers, along with a user review system. Engaging industry professionals will also help build credibility.

Legal and privacy concerns are significant. We will consult legal experts early to ensure compliance with data protection regulations, and we will implement strong encryption and secure payment methods.

A potential risk is a skill gap in some areas, like advanced API integration or UI/UX design. We will address this by seeking external help or hiring freelancers where necessary.

Scalability and performance are also important, especially with high traffic. To manage this, we will use cloud hosting solutions and conduct load testing to ensure the platform can handle growth.

Lastly, timeline and budget delays could pose risks. We will create a detailed project plan with milestones, buffer time for unforeseen issues, and regular progress reviews to stay on track.

By proactively managing these risks, we aim to ensure the successful development of the Car Finder Hub app.

### Quality Assurance

Our approach will include thorough testing throughout the development process, including unit, integration, and system testing to catch issues early. We will use a bug tracking system like Jira to log and prioritize any bugs that arise.

Performance testing will be conducted to ensure the app can handle high traffic, while security testing will focus on protecting user data and ensuring compliance with privacy regulations. Before launch, we will perform user Acceptance Testing (UAT) with real users to gather feedback and make necessary adjustments.

After the app goes live, we will continue to monitor user feedback and performance, addressing any issues quickly through regular updates. This approach ensures the app remains reliable, secure, and user-friendly.

### People and Time

Effective people and time management are crucial for the success of our CarFinder Hub app. Our team will ensure clear communication, timely execution of tasks, and efficient collaboration through well- defined roles and responsibilities.

To organize and manage tsks, we will use tools like Trello and Jira. These tools will help us track progress, assign task, and ensure that we meet deadlines. Trello will be used for visual task management, where we can create boards for different stages of development, set deadlines, and prioritize tasks. Jira will help in tracking bugs, progress, and managing sprints for better workflow in the development process.

For communication, we will rely on WhatsApp for real-time messaging, quick updates, and resolving queries. Zoom will be used for weekly team meetings to discuss progress, challenges, and any changes to the project. Additionally, we use Google drive for document sharing and collaborative work, ensuring that everyone has access to the latest project files and documentation.

By using these tools, we can streamline communication, track progress effectively, and ensure that the project stays on schedule. Regular check-ins and updates will help us stay aligned and quickly address any issues that arise.

# Chapter 4. Software Requirement Specification

## Product Functions

The CarFinder Hub app will offer several major features to fulfill its purpose effectively. These include:

* 1. Car Comparison Tool:

The system will allow users to compare multiple cars based on key factors such as price, specifications, user ratings, and other personalized preferences. This feature will help users make informed decisions about which car to purchase based on their unique needs.

* 1. Car Listings and Marketplace:

The app will integrate a marketplace where users can view car listings from various dealerships and individual sellers. Additionally, users will be able to list their own cars for sale, facilitating a two-sided marketplace for both buyers and sellers.

* 1. Advanced Search and Filtering:

The system will include advanced search and filtering options, allowing users to narrow down car options based on specific criteria such as price range, model, fuel type, a more. This will help users find the car that best meet their preferences and needs.

* 1. User Reviews and Ratings:

Users will be able to leave reviews and rate both cars and sellers on the platform. This feature will help build trust within the marketplace, providing valuable feedback from other users to guide future buyers.

## User Permissions and Types

The app is designed to accommodate different types of users, each with specific roles and access levels. Buyers will have access to car listings, comparison tools, search filters, and user reviews to help them make informed decisions. They can save favorite cars, contact sellers, and leave reviews after purchasing a vehicle. Sellers, including both individuals and dealerships, will be able to create and manage car listings, set prices, track listing performance, and respond to buyer inquiries.

Third-party dealers will have additional features such as bulk listing management, promotional tools, and API integration for automatic inventory updates. To maintain trust and quality, these dealers will undergo an authentication process before gaining full access. Administrators will

oversee platform operations, verify sellers accounts, monitor user activity, and address any reports of fraudulent listings or misconduct. They will ensure the platform remains secure, functional, and user-friendly.

The system is designed with simplicity and ease of use in mind, providing clear navigation, guidance, and support features to help all users effectively utilize the platform.

## Functional Requirements

* 1. User Management:
     + Our application will offer a simple yet detailed user management that will allow users register, log in and edit their profiles, uploading documents, profile pictures and names. However, there will be a highly developed for guests who want to use our application features without registration.
  2. CarFinder Hub Feature:
     + Requirement Description: The system should allow users to compare multiple cars based on specifications, pricing, and reviews.
     + Use Case Description: A buyer selects multiple cars from different brands ad views a side- by-side comparison of key features, such as engine type, fuel efficiency, price, and customer ratings. The system provides a summary of differences and highlights the best option on the user’s preferences.
  3. Car Listing Management
     + Requirement Description: Sellers should be able to list their cars with necessary details and media.
     + Use Case Description: A seller logs into the system, uploads car details such as model, price, mileage, and images, and submits the listings for verification. The admin verifies the listing before making it publicly visible.
  4. Advanced Search and Filtering:
     + Requirement Description: The system should provide an advanced search feature allowing users to filter cars based on various criteria.
     + Use Case Description: A buyer uses filters like budget, car type, fuel type, and location to find suitable car listings. The system dynamically updates results based on selected filters
  5. User Review and Rating System:
     + Requirement Description: Users should be able to rate and review cars and sellers for better decision-making.
     + Use Case Description: After purchasing a car, a buyer can leave a star rating and a written review about the car’s condition and the seller’s reliability. Others users can see these reviews when browsing listings.
  6. Secure Authentication and Verification:
     + Requirement Description: The system should authenticate users and verify seller legitimacy to prevent fraud.
     + Use Case Description: A new user signs up using an email or phone number. If they register as a seller, they must provide additional verification documents, which the system checks before approving the account for listing cars.

## Non-functional Requirements

#### Performance & Scalability:

#### The system should handle multiple users performing searches, comparisons, and transactions simultaneously without lag.

#### It should support future expansion, allowing more car brands, dealers, and regions to be integrated.

#### Security & Data Protection:

#### All user data, including personal details and payment information, should be securely stored and encrypted.

#### The app should implement multi-factor authentication (MFA) for seller accounts to prevent fraud.

#### Usability & User Experience:

#### The interface should e intuitive and easy to navigate for both buyers and sellers.

#### The app should provide responsive design that works across desktop, tablet, and mobile devices.

#### Reliability & Availability:

#### The system should have 100% uptime, ensuring users can access it at any time without disruptions.

#### Regular backups should be conducted to prevent data loss.

#### Compliance & Legal Standards:

#### The app should comply with data protections regulations such as GDPR to ensure privacy.

#### Seller verification should follow legal procedures to prevent fraudulent listings.

#### Real time Bidding Performance: The system should handle live bid updates with minimal latency

## Domain Requirements

#### Car Database Integration:

#### The system must connect with APIs from various car manufacturers and dealerships to fetch real-time car details, prices, and availability.

#### It should support integration with at least 4 car databases for accurate comparisons

#### Vehicle Verification & Authenticity:

#### The app must verify car details such as model, mileage, accident history, and ownership status before listing.

#### It should automated checks and manual review processes to reduce fraudulent listings.

#### Regulatory compliance:

#### The system must comply with automotive sales regulations and data protection laws (e.g., GDPR f user data privacy)

#### It should ensure that all sellers meet the necessary legal requirements before listing vehicles.

#### User Authentication & Authorization:

#### The system must differentiate between buyers, sellers, and administrators, granting access to relevant features accordingly.

#### Secure login mechanisms should b in place, with optional multi-factor authentication for high-risk actions.

#### Marketplace & Payment processing (Future Enhancement):

#### The app should allow users to make secure payments through integrated payment gateways.

#### It must provide secure escrow services to prevent fraud transactions between buyers and sellers

# Chapter 5. Project Design

## User Interface Prototype

## 

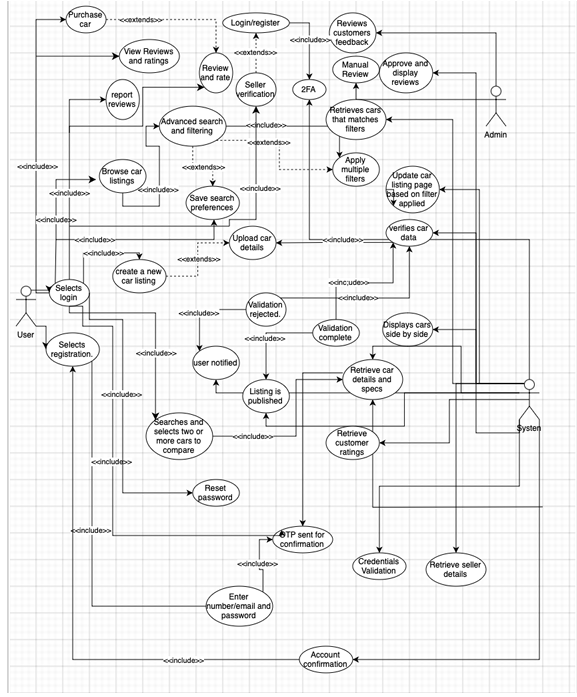
## 

## 

## 

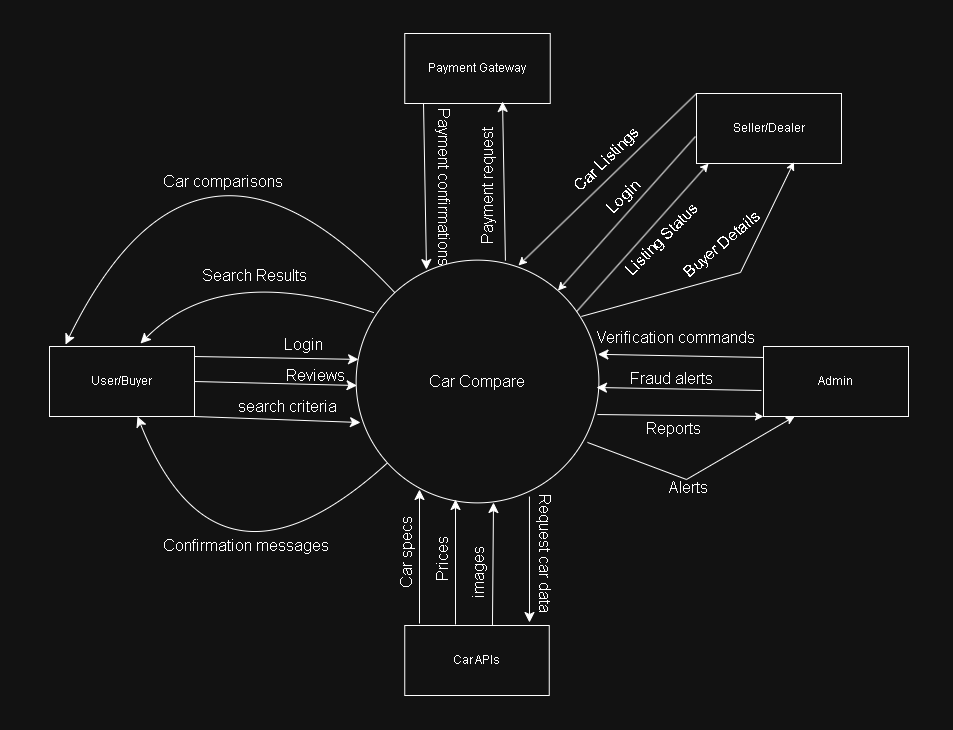
## 

## Use Case Diagram

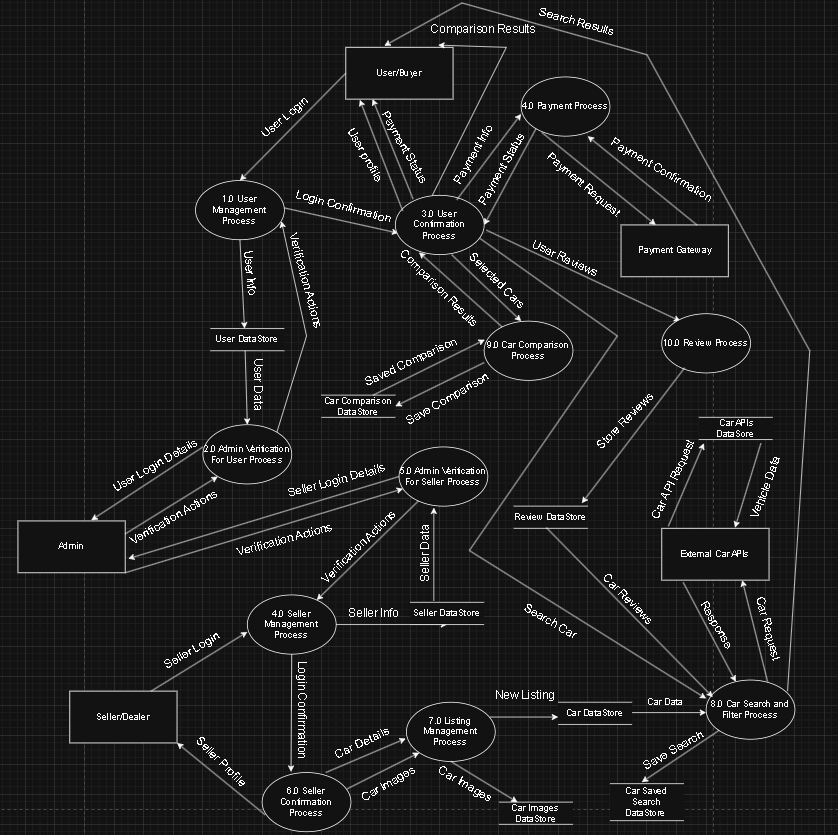


## Data Flow Diagram

Level 0:



Level 1:



## Database Diagram

#### Our system is designed to aggregate car listings from multiple dealership platforms (like

#### Edmunds, OLX, Facebook Marketplace , etc. ) into a single searchable database. Here's a

#### comprehensive breakdown of what data we store and how it's organized:

#### **User Data:**

#### Basic Information: We store user account details including email, name, contact information,

#### and securely hashed passwords.

#### Preferences: User role information (regular user vs. dealer), activity status, and login history.

#### Interactions: We track saved searches, favorite cars, and comparison lists created by users.

#### **Dealer Data:**

#### Business Information: Complete dealer profiles with name, location details (address, city, state,

#### country), and contact information.

#### Integration Data: API endpoints and keys for connecting to dealer inventory systems.

#### Performance Metrics: Activity status and timestamps for account creation and updates.

#### Vehicle Inventory

#### **Car Listings**: Comprehensive vehicle details including make, model, year, trim, price,

#### mileage, VIN, and condition.

#### Specifications: Technical details like fuel type, transmission, drivetrain, body type, engine

#### specifications, and number of doors.

#### Media: Multiple images per vehicle with indicators for primary images and display order.

#### Features: Individual vehicle features categorized for better searchability.

#### Source Information: Original dealer ID, external listing ID, and the original listing URL.

#### Availability: Current status of the listing (available or sold).

#### **Platform Integration**

#### **External Platforms:** We track which third-party platform each listing comes from (Edmunds,

#### OLX, Facebook Marketplace, etc.).

#### API Communication: We log all API interactions with dealer systems, including request

#### types, response codes, and error messages.

#### **User Engagement:**

#### Search Activity: We store search criteria, result counts, and session information to improve

#### search algorithms.

#### **Saved Searches:** Users can save search criteria with optional notification settings for new

#### matching vehicles.

#### **User Reviews**: Users can bookmark vehicles of interest with optional personal notes.

#### **Comparisons**: Users can create named comparison sets with multiple vehicles for side-by-side

#### evaluation.

#### **- Relational Structure**

#### ***Dealers and Cars***

#### \* One-to-Many:

#### o A Dealer can have many Cars.

#### o A Car belongs to one Dealer.

#### Cars & Related Entities

#### \* One-to-Many:

#### o A Car can have multiple Car Media Features. (type can be image or feature)

#### \* Many-to-Many via comparison\_cars:

#### o A Car can be part of multiple Car Comparisons.

#### o A Comparison can include multiple Cars.

#### ***Users & Interactions***

#### \* One-to-Many:

#### o A User can write multiple User Reviews.

#### o A User can save multiple Saved Searches.

#### o A User can create multiple Car Comparisons.

#### o A User can have multiple Search Logs.

#### 

#### \* One-to-Many:

#### o A User Review is written by one User and refers to one Car.

#### o A Saved Search belongs to one User.

#### o A Car Comparison belongs to one User.

#### o A Search Log belongs to one User.

#### Comparisons

#### \* Many-to-Many (via comparison\_cars):

#### o A Car Comparison can include multiple Cars.

#### o A Car can appear in multiple Comparisons.

#### API & Search Logs

#### 

#### \* One-to-Many:

#### o A Dealer can have multiple API Logs.

#### \* One-to-Many:

#### o A User can have multiple Search Logs.

#### *This architecture allows us to efficiently*

#### Aggregate inventory from multiple sources

#### Provide powerful search and filtering capabilities

#### Enable personalized user experiences

#### Maintain data integrity across the platform

#### Track and analyze system performance and user behavior

#### **System Overview**

#### This database schema supports a car dealer aggregation platform that allows multiple car dealers

#### to list their inventory in a centralized marketplace. The system facilitates car searching,

#### comparing, and favoriting while maintaining detailed information about vehicles and dealers.

#### **Key Entities**

#### \* Users: End-users who search for and compare cars.

#### \* Dealers: Car dealerships that provide inventory data.

#### \* Cars: The core entity representing vehicle listings from all dealers.

#### \* Car Features & Images: Additional information and media for each car.

#### \* Saved Searches: User-defined search criteria for notifications.

#### \* Comparisons: User-created comparisons between multiple vehicles.

#### \* Logging: API and search activity tracking for analytics and troubleshooting.

#### **Data Integration**

#### The system integrates with external dealer systems through:

#### 

#### \* API connections managed through the dealers table

#### \* Regular synchronization of inventory data

#### \* Detailed logging of all API interactions

#### Each dealer can provide data through a custom API endpoint or standardized feed, with all

#### communication tracked in api\_logs for troubleshooting and auditing.

#### **Search & User Experience**

#### The schema supports advanced search functionality through:

#### \* Detailed car specifications for filtering (make, model, price, features, etc.)

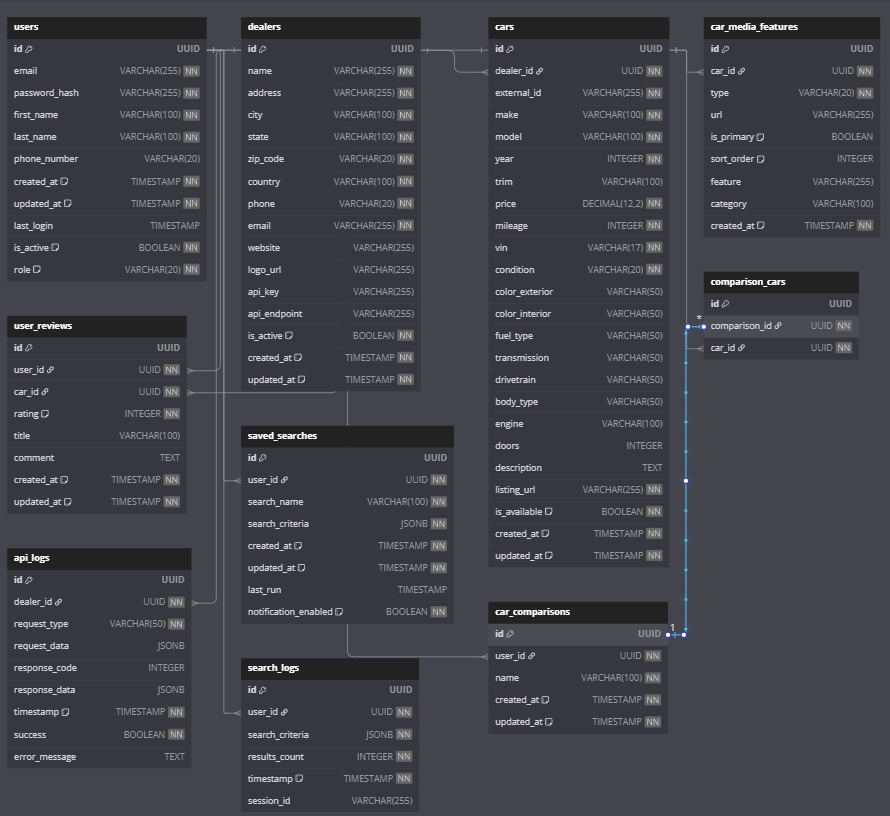
#### \* Saved searches with optional notifications for new matches

#### \* Comparison tools for evaluating multiple vehicles

#### \* Favoriting mechanism to bookmark interesting listings

#### User search behavior is tracked in search logs to improve search algorithms and provide analytics

#### on popular search criteria.



#### 

## Sequence Diagrams

## - Car Listing and Management Sequence Diagram:

## This diagram shows the seller side of our app, demonstrating how car listings are created and managed:

## Seller uploads car details (VIN, photos, price).

## Verification Service checks seller’s trust score.

## - If high score: Auto-approves → listing goes live.

## - If low score: Flags for admin review.

## Car Database stores listing as "Live" or "Pending". Key Points:

## 

## - Car Comparison Sequence Diagram:

## This diagram shows the complete journey of a user searching for cars, comparing options, and eventually making a purchase:

## User selects filters (budget, type, fuel).

## System checks login validity.

## Search Controller queries Car Database for matching cars.

## Comparison Engine generates side-by-side specs (engine, mileage, ratings).

## UI displays results with 360° view.

## 

## s

# Chapter 6. Implementation Plan

## Technical Resources

#### **Tools we are using for our project:**

#### - Flutter SDK: the backbone of our project

#### - Dart: the programming language flutter runs on

#### - Android Studio: IDE

#### Libraries:

#### - flutter\_bloc: for managing the app state

#### - flutter\_hooks: for keeping the widget logic clean and readable

#### - flutter\_svg: for the use of crisp SVG graphics (great for logos)

#### - cached\_network\_image: loads and cache images efficiently for a smooth UX

#### Networking and API integration:

#### - http : For Handling the network requests

#### - json\_serailizable: retrieve records from databse

#### Authentication:

#### -firebase\_auth: For easily setting up email, google, or social login

#### - flutter\_secure\_storage: Safely stores sensitive info like tokens.

#### Filtering and Search:

#### - Custom search logic – Adding debounce for smoother UX.

#### - flutter\_typeahead – Suggest cars while typing, autocomplete-style.

#### Optional Add-ons (maybe for the future maintenance)

#### - google\_maps\_flutter – Show car locations or dealer spots on a map.

#### - firebase\_analytics – Track what users do inside the app.

#### - firebase\_messaging – Send push notifications for updates or promotions.

#### hive / shared\_preferences – For quick key-value storage.

#### - sqflite – if our project needed a full relational database pn-device

#### **Software Component:**

#### For the software component we chose: MVC (Model-View-Controller) which is a software architectural pattern that separates n application into three main components :

#### Model, View, Controller.

#### **Software Toolset:**

#### - Dart: This is the main language our app will be written in. It’s built specifically for Flutter, which is what we’re using to build a single app that runs on both iOS and Android.

#### - JavaScript (for the backend or admin panel, if needed in the future maintanance)

#### To build a custom backend to connect with all the car dealer APIs or create an admin dashboard, this is where JavaScript comes in.

#### - SQL (for structured data):

#### If we go with a relational database (like PostgreSQL or SQLite), SQL will be used to handle queries — things like filtering cars by price, brand, or location.

#### - Data Sources:

#### External Car Dealer APIs

#### **Why we chose these tools and languages:**

#### We chose Flutter with Dart for building a fast, cross-platform mobile app with a native feel. For backend or admin tools, JavaScript offers flexibility, strong API support, and maintainable code. SQL is ideal for managing structured car data, enabling powerful search and filtering. The app will pull real-time listings from external car dealer APIs, with optional use of Firebase Firestore for caching and user data. For offline access and quick local storage, tools like Hive, SharedPreferences, or SQLite ensure performance and responsiveness. Together, this stack delivers a modern, scalable, and user-friendly car listing app

## Testing Plan

## Unit Testing

## Unit testing will focus on testing individual components in isolation:

## API Integration Tests: Write tests to verify your app correctly processes data from car manufacturer APIs

## - Test data parsing from different API formats

## - Test error handling when APIs return unexpected data

## Search Filter Tests: Unit test the logic that filters cars based on user criteria

## - Test price range filtering works correctly

## - Test multiple filter combinations (make + model + year)

## User Authentication Tests: Verify user registration and login functions

## - Test email verification process

## - Test password strength validation

## Car Comparison Logic Tests: Test the core comparison functionality

## - Verify correct calculation of price differences

## - Test feature comparison highlighting

## System (Integration) Testing

## Integration testing will ensure different components work together properly:

## End-to-End Search Flow: Test the entire search process from filter input to results display

## - Verify search results match the applied filters

## - Test pagination of search results

## Database Integration Tests: Test how your app interacts with Firebase

## - Verify car data is stored and retrieved correctly

## - Test synchronization between app and database

## Cross-Platform Testing: Ensure consistent functionality across different devices

## - Test on iOS and Android platforms

## - Test on various screen sizes

## API Integration Flow: Test complete flows that involve external APIs

## - Test full data synchronization process with car dealer platforms

## - Verify real-time data updates

## Usability Testing

## Usability testing will ensure the app meets user needs and expectations:

## User Acceptance Testing (UAT):

## - Recruit 8-12 participants representing different user personas

## - Create specific tasks (search for cars, compare models, etc.)

## - Gather feedback through observation and post-test interviews

## A/B Testing:

## - Test different UI layouts for key screens (homepage, comparison view)

## - Measure which design leads to better engagement metrics

## Performance Testing:

## - Test app response time under different network conditions

## - Verify app performance with large result sets (100+ cars)

## Accessibility Testing:

## - Test app with screen readers for visually impaired users

## - Verify color contrast meets accessibility standards

## Test Suites Mapped to Requirements

## Connect your testing approach directly to your functional requirements:

## Car Comparison Test Suite:

## - Tests the ability to select and compare multiple cars

## - Verifies correct display of differences between vehicles

## - Confirms data accuracy in comparison views

## Car Listing Management Test Suite:

## - Tests the complete flow of creating, editing, and deleting listings

## - Verifies the verification process for new listings

## - Tests image upload and management features

## Search and Filter Test Suite:

## - Tests all filter combinations work as expected

## - Verifies search performance with large datasets

## - Tests saved search functionality

## User Review Test Suite:

## - Tests the process of leaving ratings and reviews

## - Validates review moderation functionality

## - Tests display of average ratings

## Deployment Plan

## Our App will be hosted on mobile application:

## - Google Play Store for Android users

## - Apple App Store for iOS users

## Maintenance Plan

## Version Control

## Tool: Git (with GitHub or GitLab)

## Every change in the code will be tracked through Git.

## We'll use branches for features, bug fixes, and releases.

## Pull requests and code reviews will help catch issues early and ensure code quality.

## Bug

## Tool: GitHub Issues / Jira

## Bugs, enhancements, and tasks will be tracked through issue boards.

## Each issue can be linked to specific features or milestones.

## We can assign priorities, labels example UI bug, API error, and team members.

## Error Monitoring & Crash Reporting

## Tool: Firebase Crashlytics

## Real-time crash reporting with detailed logs.

## Alerts us instantly when a crash occurs and tells us exactly where in the code the problem is.

# Chapter 7. Conclusion and Future Work

# Our CarFinder Hub project successfully addresses key challenges in the car-buying process by providing a centralized platform for comparing, buying, and selling vehicles. With features like verified listings, virtual tours, and AI-driven recommendations, our app enhances user trust and decision-making. Market research confirmed strong demand for such a solution, with 70% of respondents expressing definite interest. Our project follows a structured SDLC Scrum approach, leveraging modern technologies like Flutter, Firebase, and secure APIs to ensure scalability and performance. Rigorous testing and risk management strategies were implemented to guarantee reliability. Future work includes expanding integrations with more dealers, adding advanced analytics, and enhancing eco-friendly options. CarFinder Hub has the potential to revolutionize the car marketplace by simplifying comparisons and fostering transparency.

# Chapter 8. References

##### Bansal, C. L., Shruti B. Gupta, and Shalini Aggarwal. "Case Study of Car Dekho. Com: Racing to Emerge as a Unicorn in Auto Portal." *Journal of IMS Group* 13.2 (2016).

1. **Chandak, Ashish, et al. "Car price prediction using machine learning." *International Journal of Computer Sciences and Engineering* 7.5 (2019): 444-450.**

##### Book, Kelley Blue. "Kelley blue book." 2015,

##### 4) Almeida, F., & Monteiro, J. (2020). *The Role of APIs in Digital Transformation: A Case Study in Automotive Platforms.* Journal of Information Systems and Technology, 15(2), 45-60.

##### 5) Nielsen, J. (2020). *Usability Engineering.* Morgan Kaufmann.

##### 6) Statista. (2024). *Global Online Car Sales Market Size, 2020–2025.*

##### 7) World Economic Forum. (2023). *The Future of Mobility: Sustainable Transportation Trends*

# 