UNIVERSIDAD DE GRANADA

BACHELOR'S THESIS

BACHELOR'S DEGREE IN COMPUTER SCIENCE AND ENGINEERING

AlDiaCAR

Design and development of a sustainability-focused app for optimizing personal vehicle usage

Author
Juan Manuel Segura Duarte

Thesis supervisor
Rosana Montes Soldado

ETSIIT
Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación

Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación

—

Granada, September 2025

Design and development of a sustainability-focused app for optimizing personal vehicle usage

Juan Manuel Segura Duarte

Keywords: Sustainable Mobility, Green IT, Full-Stack Development, React Native, Vehicle Management, Gamification.

Abstract:

This project presents the design and development of *AlDiaCAR*[1], a cross-platform mobile and web application aimed at promoting sustainable and conscious use of a personal fleet of vehicles. The application is designed to support an individual user managing one or more vehicles, offering tools to centralize maintenance tracking and encourage environmentally-aware decisions.

AlDiaCAR provides three core functionalities: (1) unified vehicle maintenance tracking with smart reminders based on upcoming dates and distance-based intervals, (2) a trip logging system that updates vehicle metrics and user statistics, and (3) a recommendation system that suggests the most sustainable vehicle for a given route based on estimated emissions. To further engage the user, the application incorporates gamification elements and statistical feedback, fostering eco-responsible habits.

The system architecture is built with Expo (a React Native framework) for the frontend and Express.js (a Node.js framework) for the backend, supported by a MongoDB database for flexible data management. The project was developed following modular design principles and includes a comprehensive testing suite to ensure quality and reliability.

This report documents the design rationale, implementation details, and validation of the application, which aligns with the broader goals of sustainable software and "Green through IT" initiatives.

[1]The complete source code for this project is available at: https://github.com/jseg380/Trabajo-Fin-Grado

Diseño y desarrollo de una aplicación para la optimización del uso del vehículo privado con enfoque en sostenibilidad

Juan Manuel Segura Duarte

Palabras clave: Movilidad Sostenible, TI Verde, Desarrollo Full-Stack, React Native, Gestión de Vehículos, Gamificación.

Resumen:

Este proyecto presenta el diseño y desarrollo de *AIDiaCAR*[2], una aplicación multiplataforma (móvil y web) orientada a fomentar un uso sostenible y consciente de una flota de vehículos personal. La aplicación está diseñada para dar soporte a un usuario individual que gestiona uno o más vehículos, ofreciendo herramientas para centralizar el seguimiento del mantenimiento y fomentar decisiones respetuosas con el medio ambiente.

AIDiaCAR ofrece tres funcionalidades principales: (1) seguimiento unificado del mantenimiento del vehículo con recordatorios inteligentes basados en fechas próximas e intervalos de distancia, (2) un sistema de registro de viajes que actualiza las métricas del vehículo y las estadísticas del usuario, y (3) un sistema de recomendaciones que sugiere el vehículo más sostenible para una ruta determinada en función de las emisiones estimadas. Para implicar al usuario, la aplicación incorpora elementos de gamificación y retroalimentación estadística que fomentan hábitos eco-responsables.

La arquitectura del sistema se ha desarrollado con Expo (un framework de React Native) para el frontend y Express.js (un framework de Node.js) para el backend, utilizando una base de datos MongoDB para una gestión de datos flexible. El proyecto se ha desarrollado siguiendo principios de diseño modular e incluye una completa suite de tests para garantizar su calidad y fiabilidad.

Este informe documenta la justificación del diseño, los detalles de implementación y la validación de la aplicación, que se alinea con los objetivos más amplios del software sostenible y las iniciativas de "TI Verde" (Green through IT).

[2]El código fuente completo de este proyecto está disponible en: https://github.com/jseg380/Trabajo-Fin-Grado

Acknowledgements

I would like to take this opportunity to express my heartfelt gratitude to all those who have supported me throughout my journey in completing this thesis.

First and foremost, I would like to thank my family for their unwavering love and encouragement. Your belief in my abilities has been a constant source of motivation, and I am forever grateful for your sacrifices and support.

I am also grateful to my friends and peers who have stood by me during this challenging process. Your camaraderie, late-night study sessions, and shared laughter have made this journey not only bearable but also enjoyable. Thank you for being my sounding board and for always believing in me.

Additionally, I would like to acknowledge the faculty and staff of the University of Granada for providing a nurturing academic environment. Your dedication to teaching and mentorship has profoundly impacted my educational experience.

This thesis is not just a reflection of my efforts but a testament to the collective support of all those around me. Thank you for being part of this journey.

Contents

List of Figures

List of Tables

Glossary

A

Autónomo An autónomo is the Spanish legal and fiscal term for a self-employed individual or sole trader, roughly equivalent to a "freelancer" or "independent contractor" in anglophone contexts. This is a highly prevalent form of employment in Spain, encompassing a vast range of professions from skilled tradespeople like plumbers and electricians to professional service providers such as consultants, designers, and lawyers. Unlike employees with fixed 9-to-5 schedules, autónomos typically manage their own working hours and client engagements, resulting in a highly variable and often unpredictable daily schedule. This lack of a fixed routine is a critical factor in the user persona for this thesis, as it complicates household resource planning—particularly the availability and use of shared vehicles—necessitating a dynamic, real-time coordination tool rather than a static, schedule-based system.. 3

D

Distintivo Ambiental The Distintivo Ambiental is an official vehicle classification system implemented by Spain's Dirección General de Tráfico (DGT). It categorizes vehicles based on their pollutant emission levels, assigning a corresponding colored sticker that must be displayed on the vehicle. This system is the primary mechanism used by municipalities to regulate traffic in Low Emission Zones (ZBEs). The badges are not a measure of CO_2 emissions but of local air pollutants like NOx and particulate matter. The main classifications for passenger cars are:

• **0 Emisiones (Blue Badge)**: Reserved for the cleanest vehicles, primarily Battery Electric Vehicles (BEVs), extended-range electric vehicles (REEVs), plug-in hybrids (PHEVs) with a range of over 40km, and fuel cell vehicles. They enjoy the most privileges, including unrestricted access to all ZBEs.

• **ECO (Green and Blue Badge)**: This category includes standard hybrid vehicles (HEVs), plug-in hybrids with a range under 40km, and vehicles powered by compressed natural gas (CNG) or liquefied petroleum gas (LPG). They

have significant privileges, though slightly fewer than '0 Emisiones'.
• **C (Green Badge)**: For gasoline vehicles registered from 2006 onwards (Euro 4, 5, 6 standards) and diesel vehicles registered from 2014 onwards (Euro 6). This is a very common category for modern internal combustion engine vehicles.
• **B (Yellow Badge)**: For gasoline vehicles registered from 2001 to 2005 (Euro 3) and diesel vehicles from 2006 to 2013 (Euro 5). These vehicles face the most significant access restrictions in ZBEs.
• **Sin Distintivo (No Badge)**: The oldest and most polluting vehicles, which are typically barred from entering any ZBE. . 3, 15

Z

Zona de Bajas Emisiones (ZBE) A *Zona de Bajas Emisiones (ZBE)* is a geographically defined urban area where access for certain polluting vehicles is restricted to improve air quality. Mandated by Spain's Climate Change and Energy Transition Law, all municipalities with over 50,000 inhabitants are required to establish ZBEs. The implementation and severity of these restrictions are at the discretion of the local city council, leading to a fragmented and often confusing regulatory landscape for drivers.
Access restrictions are enforced based on the vehicle's Distintivo Ambiental. A ZBE may, for example:
• **0 Emisiones (Blue Badge)**: Reserved for the cleanest vehicles, primarily Battery Electric Vehicles (BEVs), extended-range electric vehicles (REEVs), plug-in hybrids (PHEVs) with a range of over 40km, and fuel cell vehicles. They enjoy the most privileges, including unrestricted access to all ZBEs.
• **ECO (Green and Blue Badge)**: This category includes standard hybrid vehicles (HEVs), plug-in hybrids with a range under 40km, and vehicles powered by compressed natural gas (CNG) or liquefied petroleum gas (LPG). They have significant privileges, though slightly fewer than '0 Emisiones'.
• **C (Green Badge)**: For gasoline vehicles registered from 2006 onwards (Euro 4, 5, 6 standards) and diesel vehicles registered from 2014 onwards (Euro 6). This is a very common category for modern internal combustion engine vehicles.
• **B (Yellow Badge)**: For gasoline vehicles registered from 2001 to 2005 (Euro 3) and diesel vehicles from 2006 to 2013 (Euro 5). These vehicles face the most significant access restrictions in ZBEs.
• **Sin Distintivo (No Badge)**: The oldest and most polluting vehicles, which are typically barred from entering any ZBE. . 3

1 Introduction

1.1 Context and relevance: global emissions and personal vehicle management

The escalating climate crisis, a defining global challenge of the 21st century, necessitates profound and immediate transformations across all sectors of society. Among these, the transportation sector emerges as a particularly critical domain for intervention due to its substantial contribution to anthropogenic greenhouse gas emissions. According to a comprehensive analysis of global climate data, the transport sector was directly responsible for approximately 16.2% of all greenhouse gas (GHG) emissions recorded in 2016.

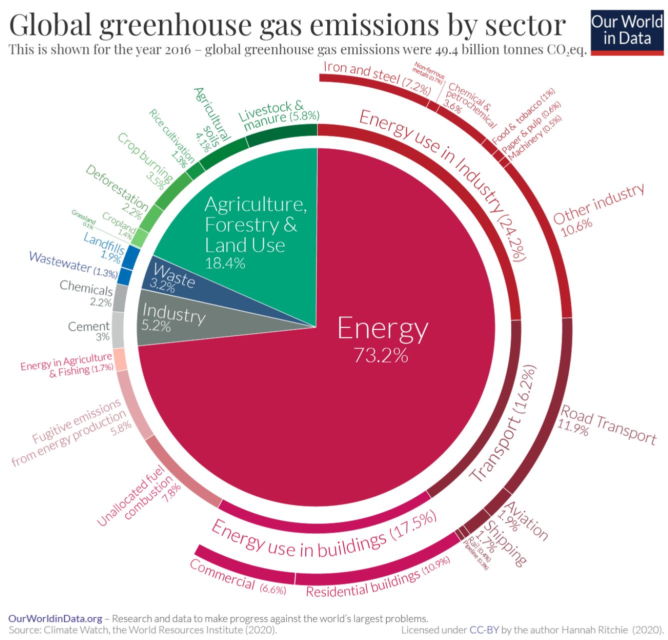


Figure 1.1: Breakdown of global greenhouse gas emissions by sector.

A more granular examination of this figure reveals that road transport—encompassing a wide array of vehicles such as cars, motorcycles, buses, and trucks—constituted the single largest contributor, accounting for a significant 11.9% of total global emissions. Most notably, a striking 60% of these road transport emissions can be attributed directly to passenger travel, a figure dominated by the widespread use of private vehicles [1]. This disaggregation underscores the disproportionate environmental impact of individual mobility choices.

This pronounced reliance on private automobiles is a hallmark of developed economies. In the European Union, for example, statistical trends indicate a persistent pattern of increasing private vehicle ownership, with the number of passenger cars per thousand inhabitants reaching a new peak of 560 in the year 2022 [2]. This saturation of private vehicles not only places immense strain on public infrastructure but also complicates efforts to meet ambitious regional and national emissions reduction targets.

This thesis posits that the millions of multi-car households globally constitute a vast, decentralized, and inefficiently managed network of small-scale vehicle fleets. It is within this context of the fragmented, privately-managed household "micro-fleet" that this research identifies a unique and critically underserved technological niche. The cumulative impact of suboptimal vehicle selection, uncoordinated maintenance, and inefficient usage patterns within these micro-fleets represents a substantial, yet largely unaddressed, opportunity for environmental and economic improvement.

Consequently, this thesis directly confronts these widespread inefficiencies by championing a framework of holistic sustainability for private vehicle utilization. This comprehensive concept necessarily extends beyond the singular act of purchasing an electric vehicle. True sustainability in this context must encompass a multi-faceted approach, including the disciplined and diligent maintenance of all vehicles to ensure they operate at their peak design efficiency. It is well-documented that poorly maintained vehicles, regardless of their powertrain, can experience a significant degradation in performance, leading to increased fuel consumption and a corresponding rise in harmful pollutant emissions [3]. Furthermore, this holistic view involves fostering more conscious and data-informed vehicle selection for each specific journey, as well as promoting the consistent adoption of eco-driving habits among all users within a household.

Furthermore, the theoretical underpinnings of this project are firmly grounded in the principles of Green Computing, with a specific focus on the paradigm known as "ICT for Sustainability." This particular pillar of Green IT is oriented towards the strategic application of information and communication technologies to monitor, model, and ultimately improve the efficiency of real-world physical processes. By architecting intelligent software solutions, it becomes possible to empower end-users to make more informed and environmentally conscious decisions in their daily lives. This project hypothesizes that a well-designed mobile application can function as a potent form of persuasive technology, a term defined by B.J. Fogg as technology that is intentionally designed to change a person's attitudes or behaviors without resorting to coercion or deception [4]. By providing timely feedback, relevant information, and facilitating collaborative planning, such a system can effectively nudge users towards more sustainable mobility patterns, thereby transforming abstract environmental goals into concrete, actionable behaviors.

1.2 The problem statement: the management gap for the modern vehicle owner

1.2.1 User persona: The Martínez household

To precisely articulate and contextualize the real-world challenges this thesis aims to resolve, we employ the methodological tool of a user persona. We introduce the Martínez family—a carefully constructed archetype of a modern, middle-class household comprising four individuals. This family unit consists of two parents and their two adult children, all residing in the same home. Their professional and academic lives reflect contemporary societal structures: one parent is a self-employed professional (autónomo) whose work demands a highly variable and often unpredictable schedule. The other parent maintains a traditional 9-to-5 office job with a regular commute. One of the adult children works primarily from home in a remote capacity, while the other is a full-time university student with their own distinct travel requirements.

Driven by both economic prudence and a genuine concern for their environmental footprint, the family has made a conscious decision to manage a shared pool of three vehicles, correctly identifying the acquisition of a fourth car as an unnecessary financial and ecological burden. This shared 'household fleet' is deliberately diverse, designed to mirror common vehicle ownership patterns observed in contemporary Spain and other parts of Europe:

- **Vehicle 1: The City runner.** This vehicle is a 2014 diesel-powered hatchback. Its primary attributes are its compact size, superior fuel efficiency in urban environments, and ease of parking. However, its age and engine type mean it has been assigned a Distintivo Ambiental B, an environmental classification that legally restricts its access to designated Low Emission Zones (Zonas de Bajas Emisiones - ZBE) now prevalent in major city centers. This regulatory constraint creates a significant operational limitation.

- **Vehicle 2: The All-rounder.** A 2017 gasoline sedan, this car holds a more favorable Distintivo 'C'. It represents a compromise, offering a satisfactory balance of passenger comfort, luggage capacity, and acceptable fuel efficiency. Crucially, its environmental classification grants it full access to many of the ZBEs that restrict the City Runner, making it a more versatile asset for a wider range of journeys.
- **Vehicle 3: The Workhorse.** The newest vehicle in the fleet is a 2021 diesel SUV, which also possesses a Distintivo 'C'. This vehicle is reserved for specific use cases such as long-distance family trips, transporting bulky items, or when maximum passenger comfort is a priority. Despite its utility and modern features, it inherently has the highest fuel consumption, greatest carbon emissions, and most expensive running costs of the three.

The confluence of the family members' unpredictable, often overlapping schedules with the distinct operational and regulatory characteristics of their diverse vehicle set generates a complex, daily logistical puzzle. The management of this micro-fleet thus transcends simple maintenance scheduling and becomes a significant source of recurring inefficiency and household friction.

1.2.2 A day in the Martínez's household: The cascading inefficiencies

To illustrate this dynamic, let us consider a typical Wednesday morning scenario. The *autónomo* parent must attend a critical client meeting located deep within the city center's primary ZBE, a destination that immediately renders the 'B' rated City Runner unusable for this specific trip. Concurrently, the university student needs to travel to campus for a lecture and, given the choice, would naturally prefer the small, fuel-efficient 'B' car for its low running cost and ease of parking near the university. Meanwhile, the parent with the 9-to-5 job has already departed for their office, having taken the 'C' rated sedan, the 'All-Rounder,' as is their daily routine.

This seemingly mundane situation triggers a cascade of logistical queries that, in the absence of a centralized management system, must be resolved through a flurry of ad-hoc, inefficient communication and physical verification. Questions arise instantly: Is the SUV's key readily available, or did someone misplace it? Is the vehicle's mandatory technical inspection (ITV) due in the coming week, making a long journey inadvisable and potentially illegal? Does the SUV have sufficient fuel for the required trip, or will it necessitate an unplanned stop?

Lacking a unified, digital source of truth for the status of their shared assets, the parent is compelled to make a decision based on incomplete information. In this instance, they take the high-emission, high-cost SUV for what is essentially a short urban errand. This represents a demonstrably suboptimal choice, a direct consequence of a systemic coordination failure. This single, commonplace event serves to crystallize the core pain points that this thesis seeks to address.

1.2.3 Core pain points

The daily challenges faced by the Martínez household can be deconstructed into four distinct, yet interconnected, core pain points. These issues represent significant barriers to the efficient, economical, and environmentally responsible management of a shared household vehicle fleet.

The first and most immediate issue is **coordination friction**. The current management method relies entirely on synchronous, manual communication channels such as text messages, phone calls, and face-to-face conversations. This constant need to query the status and location of vehicles and their keys introduces a significant cognitive overhead into the daily lives of the family members. It consumes valuable time, generates unnecessary stress, and frequently leads to minor conflicts and misunderstandings, creating a persistent source of domestic friction.

Secondly, the family suffers from **shared maintenance blindness**. In a multi-driver, multi-vehicle environment, responsibility for vehicle upkeep becomes dangerously diffused. No single individual possesses a complete, accurate, or real-time overview of the comprehensive maintenance status—including mandatory inspections (ITV), scheduled oil changes, tire wear and pressure, and other critical service intervals—across all three vehicles. This lack of centralized oversight inevitably leads to critical and potentially hazardous oversights, increasing the risk of mechanical failures, legal infractions, and avoidable repair costs.

The third pain point is **inefficient, constraint-unaware selection**. In the heat of the moment, vehicle choices are predominantly dictated by immediate physical availability rather than a rational, holistic assessment of all relevant constraints. An optimal decision would weigh multiple factors simultaneously: ZBE access restrictions, real-time fuel costs, the specific emissions profile of each car, upcoming maintenance deadlines, and the nature of the journey itself. The absence of a tool to facilitate this multi-criteria analysis ensures that the family consistently makes suboptimal choices, leading to higher expenses and a greater environmental impact than necessary.

Finally, the household operates with a complete **lack of shared visibility** into their collective mobility behavior. There exists no mechanism for the family to aggregately track and visualize their total transportation expenditures, fuel consumption, or cumulative environmental impact over time. This information vacuum prevents the formation of effective feedback loops, which are essential for behavioral change. Without access to clear, quantifiable data on their collective habits, the family is unable to make informed group decisions, set meaningful goals for improvement, or objectively measure their progress towards becoming more sustainable and economically efficient.

2 | Background and related work

To adequately contextualize the novel contribution of this thesis, this chapter undertakes a comprehensive review of the existing landscape of software solutions, theoretical frameworks, and academic literature pertinent to vehicle management and sustainable mobility. The primary objective of this chapter is to meticulously map the state of the art, thereby establishing a clear and defensible rationale for the development of the *AIDiaCAR* system. We will systematically examine the current technological offerings across three distinct and highly segmented categories of applications: enterprise-grade commercial fleet management systems, consumer-focused personal car maintenance trackers, and specialized eco-routing navigation tools. This analysis will not only detail the functionalities of existing systems but also critically evaluate their inherent limitations when applied to the specific problem domain of a multi-vehicle, multi-driver household.

Furthermore, this review will delve into the theoretical foundations that provide the intellectual scaffolding for this project's design philosophy. Specifically, we will explore the principles of gamification as a mechanism for behavior change and the broader paradigm of Green Information Technology (Green IT), particularly the concept of "Green through IT" or Sustainable Human-Computer Interaction (HCI). By grounding our practical design choices in established academic theory, we aim to enhance the project's robustness and potential for real-world impact. The chapter will culminate in a detailed gap analysis, synthesizing the findings from our market and literature review to precisely identify the unoccupied technological and conceptual niche that the *AIDiaCAR* project is designed to fill. This final section will serve as the definitive justification for the thesis, articulating its unique value proposition and contribution to the field.

2.1 State of the art in vehicle management applications

The contemporary market for vehicle-related software is both mature and extensive, yet it remains conspicuously fragmented. Existing solutions are typically characterized by a high degree of specialization, with software platforms meticulously engineered to target either large-scale commercial enterprises or individual users. This segmentation has resulted in a landscape where tools are powerful within their intended vertical but fundamentally ill-equipped to address the hybrid challenges faced by a modern household managing a small, private fleet of vehicles. The following subsections will dissect each major category, highlighting their capabilities and, more importantly, their shortcomings in relation to the problem statement of this thesis.

2.1.1 Commercial fleet management systems

The Business-to-Business (B2B) sector of fleet management is dominated by powerful, data-intensive software-as-a-service (SaaS) platforms such as *Fleetio* and *Samsara*. These systems are engineered to provide corporations with granular control and comprehensive oversight over large, complex, and geographically dispersed vehicle fleets. Their core value proposition lies in optimizing operational efficiency, ensuring regulatory compliance, and minimizing costs at scale.

The feature set of these platforms is consequently vast and sophisticated. Key functionalities typically include:

- **Real-time GPS tracking and geofencing:** Constant monitoring of vehicle location, speed, and heading, coupled with the ability to define virtual perimeters (geofences) that trigger alerts when crossed.
- **Advanced telematics:** Integration with onboard vehicle hardware (often via the OBD-II port) to capture a rich stream of diagnostic data, including engine RPM, fuel consumption, fault codes (DTCs), and harsh braking or acceleration events.
- **Comprehensive maintenance scheduling:** Automated workflows for preventative maintenance based on mileage, engine hours, or calendar intervals, including work order generation and service history logging.
- **Fuel management:** Detailed tracking of fuel purchases, consumption analysis, and integration with fuel card providers to detect fraud or inefficiency.

- **Driver behavior monitoring:** Scoring and reporting on driver performance to identify unsafe or inefficient habits, often used for training and safety programs.
- **Regulatory compliance:** Features designed to automate compliance with regulations such as the Electronic Logging Device (ELD) mandate for hours of service and International Fuel Tax Agreement (IFTA) reporting.

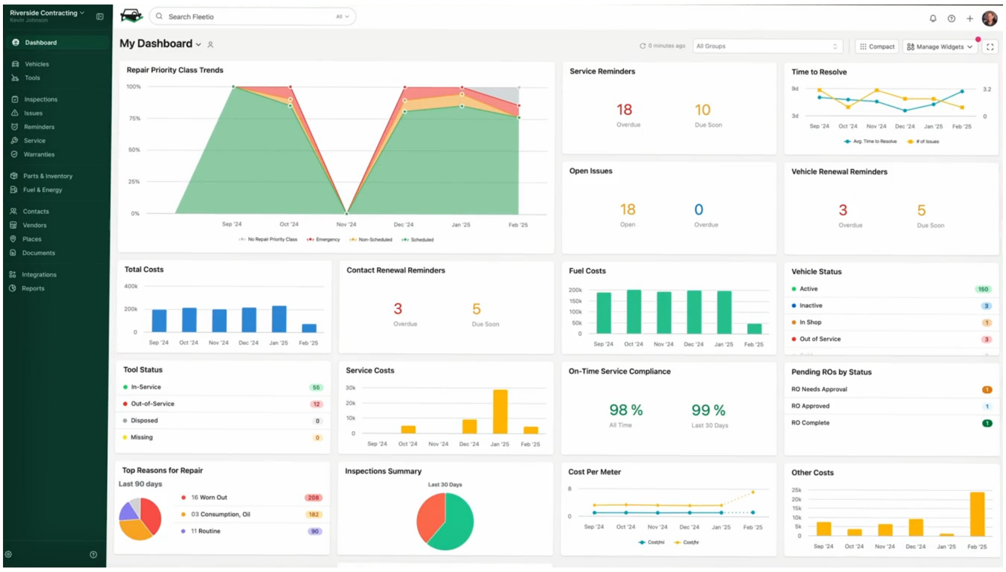


Figure 2.1: Illustrative example of a typical commercial fleet management dashboard (Fleetio), showcasing the complexity and data-density designed for professional logistics managers.

While these systems represent the pinnacle of vehicle management technology, their direct application to the context of a private household, such as the Martínez family persona, is fundamentally unviable for a confluence of reasons:

- **Prohibitive complexity and cost:** These are enterprise-grade platforms. Their pricing models are typically structured on a per-vehicle, per-month basis, often with substantial setup fees. For a family managing three cars, the cost would be unjustifiable. Moreover, the feature set is overwhelmingly complex, presenting a steep learning curve and a significant cognitive burden for non-technical users.

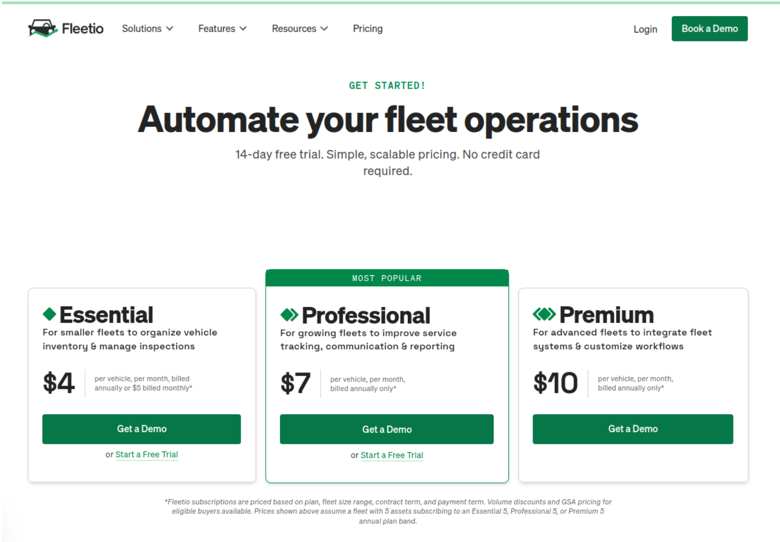


Figure 2.2: Cost of Fleetio starts at 5$ per vehicle per month if billed monthly, 4$ if billed annually. Totalling at about 96$ for a 2 vehicle fleet in the best of the cases.

- **Misaligned user experience (UX):** The user interface and overall experience are meticulously crafted for a professional fleet manager or dispatcher whose primary job function is to analyze data and manage logistics. The UX prioritizes data density and administrative control over the simplicity, accessibility, and collaborative ease-of-use required for a family context.

- **Fundamentally different focus:** The teleological purpose of a commercial fleet system is the optimization of business assets to maximize profit and minimize risk. Their design philosophy is rooted in surveillance and control. In contrast, the goal of a personal fleet management tool should be to facilitate collaboration, reduce domestic friction, and promote positive, sustainable habits among trusted users. The underlying motivations are fundamentally different.

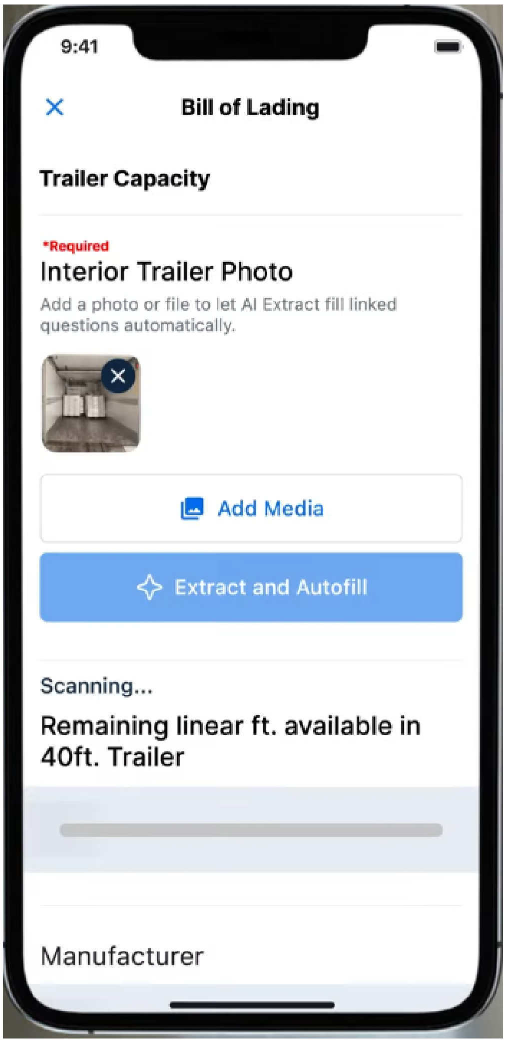


Figure 2.3: Example of a mobile interface for a commercial fleet application (Samsara). The focus remains on asset tracking and driver logs, which is misaligned with the needs of a family.

- **Data privacy and social dynamics:** The level of granular tracking (e.g., real-time location, driving habits) that is standard in a corporate setting is often socially unacceptable and raises significant privacy concerns within a family unit. Implementing such a system could introduce an uncomfortable dynamic of surveillance rather than cooperation.

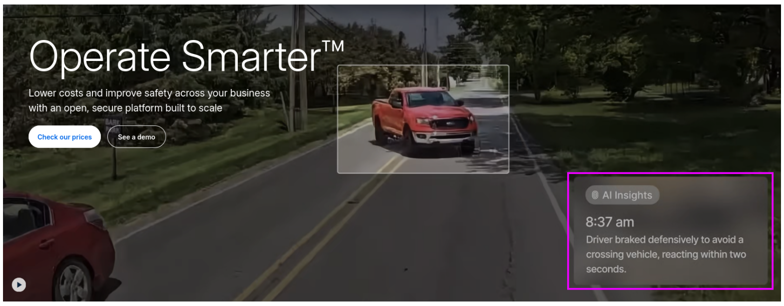


Figure 2.4: Every action of the driver is monitored and analysed. In this demo video from Samsara we can see that the driver reaction was stored and analysed

2.1.2 Personal car maintenance applications

On the other end of the spectrum are applications developed for the individual car owner. This category includes popular apps like *Drivvo*, *Fuelly*, and *aCar*. These tools are designed to serve as digital logbooks, empowering a single user to meticulously track the health, expenses, and history of their personal vehicle.

Their core competency lies in the detailed logging of specific data points. Users can manually record every fuel-up, calculate fuel economy (MPG or L/100km), log all maintenance and repair expenses, and set personalized reminders for crucial service intervals, such as oil changes, tire rotations, or mandatory technical inspections like the Spanish ITV. Many of these applications also offer basic reporting features, allowing users to visualize their spending and fuel consumption over time.

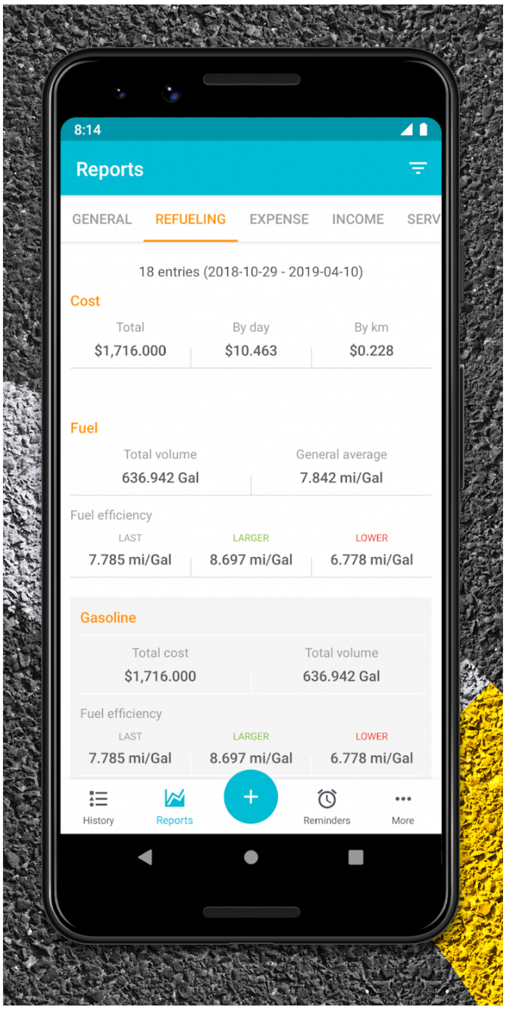


Figure 2.5: Typical user interface for a personal car maintenance app like Drivvo, centered on logging fuel, services, and expenses for a single vehicle.

However, despite their utility for the dedicated single-car enthusiast, their effectiveness rapidly diminishes when applied to the multi-vehicle, multi-driver scenario that this project specifically addresses. Their primary drawbacks include:

- **Inherent single-user, single-vehicle focus:** These applications are architected around the mental model of one person managing one primary vehicle. While some may allow the user to add multiple vehicle profiles, the interface is rarely optimized for comparing them or managing them as a collective "fleet." They lack a centralized dashboard that provides an at-a-glance overview of the entire household's automotive assets.

- **Absence of collaborative features:** Crucially, these apps are not designed for shared access or collaborative management. They function as data silos on a single user's device. There is no mechanism for multiple family members to view and update the status of a shared vehicle in real-time. This completely fails to address the core problem of "shared maintenance blindness" identified in Chapter 1.

- **Superficial approach to sustainability:** While fuel economy calculation is a standard feature, its purpose is almost exclusively financial—to help the user track their fuel expenses. These applications do not leverage this data to provide intelligent, forward-looking recommendations. They cannot, for example, advise a user on which of their three cars would be the most ecologically and economically sound choice for an upcoming journey based on its specific parameters.

- **Passive and utilitarian engagement model:** The vast majority of these apps are functional utilities that require disciplined, manual data entry. They are passive tools that depend entirely on the user's intrinsic motivation. They do not incorporate modern engagement techniques, such as gamification, to actively encourage positive behaviors like performing timely maintenance or adopting more efficient driving habits.

2.1.3 Eco-routing and navigation tools

A third category of relevant software has emerged within mainstream navigation applications. In response to growing environmental awareness, technology giants have begun to integrate sustainability-focused features into their platforms. The most prominent example is *Google Maps*, which introduced an "eco-friendly routing" option. This feature's algorithm analyzes various factors, including road incline, traffic congestion, and the number of stops, to suggest a route that is optimized for the lowest possible fuel or energy consumption, often presenting it alongside the fastest route.

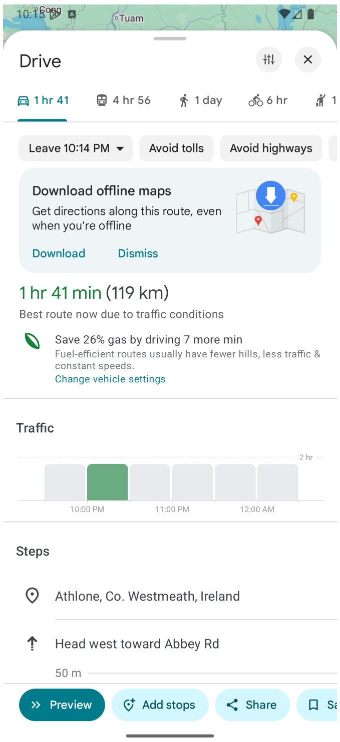


Figure 2.6: An illustration of the eco-friendly routing feature in Google Maps.

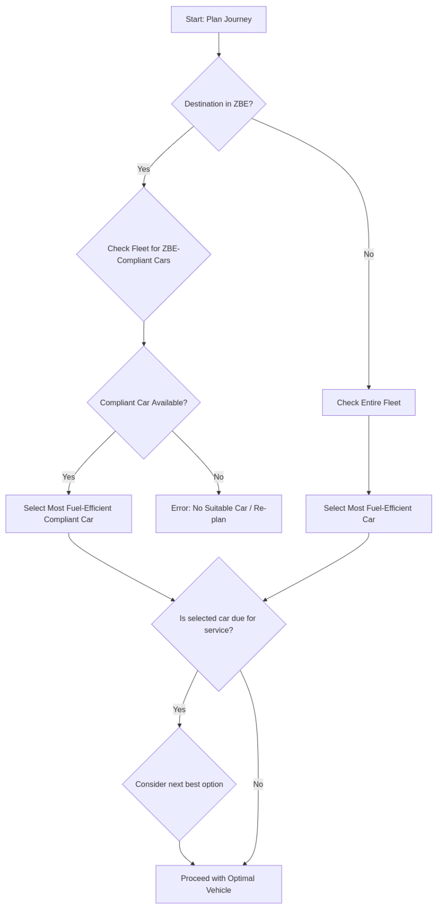
While this represents a positive step towards promoting sustainable mobility on a massive scale, the limitation of these tools lies in their profound lack of integration with the user's specific context and available resources. They operate as isolated, single-purpose functions and are consequently incapable of answering the more complex, multi-faceted questions that are central to the problem this thesis addresses. The core deficiencies are:

- **Vehicle agnosticism:** The eco-routing algorithm operates on a generic vehicle model or a very basic user selection (e.g., gasoline, diesel, electric). It is entirely unaware of the specific set of vehicles a user actually owns, their individual efficiencies, their real-time maintenance status, or their regulatory constraints (like a Distintivo Ambiental).

- **Inability to recommend a vehicle:** The system can optimize the route for a given, abstract car, but it fundamentally cannot help the user choose the optimal car for that route from their personal fleet. It cannot advise the Martínez family on whether to take the diesel hatchback or the gasoline sedan for a specific trip into the city.

- **Disconnection from broader management goals:** The feature is transactional and ephemeral. It provides a recommendation for a single journey and is completely disconnected from the user's long-term vehicle management goals, such as managing upcoming maintenance, or tracking a cumulative household carbon footprint.

The decision-making process for a multi-vehicle household is a complex tree of interdependent variables, which current eco-routing tools are not equipped to navigate. The following diagram illustrates a simplified version of this logic.



2.2 Theoretical foundations

The design and implementation of the *AIDiaCAR* system are not based merely on functional requirements but are deeply informed by established theoretical principles from the fields of human-computer interaction (HCI) and sustainable computing. By leveraging these academic frameworks, the project aims to create a solution that is not only useful but also engaging and genuinely effective at influencing user behavior.

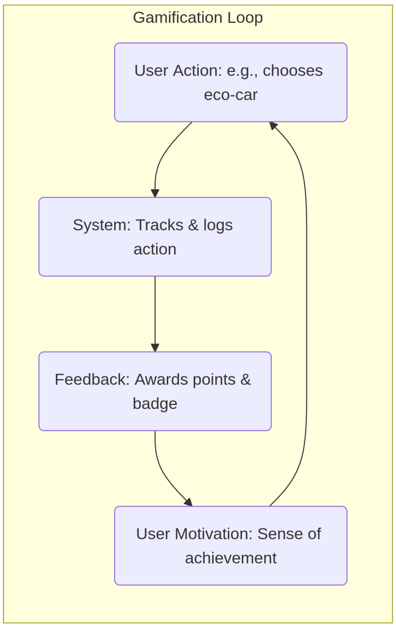
2.2.1 Gamification for behavior change

Gamification is a concept that has gained significant traction in HCI research and practice over the past decade. It is formally defined as "the use of game design elements in non-game contexts" [5]. The core premise is that the motivational techniques that make games so compelling—such as points, badges, leaderboards, and progress bars—can be extracted and applied to real-world applications to increase user engagement and encourage specific, desirable behaviors.

In the context of *AIDiaCAR*, gamification is not a superficial addition but a core mechanic designed to foster long-term habit formation. The application aims to provide positive reinforcement for actions that align with the goals of sustainability and responsible vehicle ownership. For example:

- **Achievements and badges:** Users can earn badges for achievements like "Eco-Warrior" (consistently choosing the lowest-emission vehicle), "Maintenance Master" (logging all service on time), or "Planner Pro" (coordinating vehicle usage ahead of time).
- **Progress statistics:** The application will provide clear, visual feedback on the family's collective progress, showing trends in fuel savings, CO_2 reduction, and money saved. This makes the positive impact of their choices tangible and rewarding.

This approach is supported by a large body of empirical research. A comprehensive meta-analysis by Hamari et al. concluded that gamification is effective in a variety of domains, with significant positive outcomes for user engagement and behavior change, provided it is thoughtfully implemented and aligned with user motivations [6]. The following diagram illustrates the intended motivational loop within *AIDiaCAR*.



2.2.2 Green IT and sustainable HCI

This project is fundamentally an exercise in Green Information Technology (Green IT), a field that explores the intersection of computing and environmental sustainability. Green IT is often bifurcated into two distinct sub-domains:

1. **Green in IT:** This focuses on reducing the direct environmental impact of the computing industry itself, through measures like creating energy-efficient data centers, designing recyclable hardware, and reducing e-waste.

2. **Green through IT (or ICT for Sustainability):** This focuses on leveraging information and communication technologies (ICT) as a tool to effect positive environmental change in other domains of society. This project is a clear exemplar of this second category.

AIDiaCAR uses software as a persuasive technology to influence user behavior in the physical world—specifically, their transportation choices—leading to tangible environmental benefits such as reduced fuel consumption and lower greenhouse gas emissions. This approach directly aligns with research by scholars like Berkhout and Hertin, who analyzed the complex role of digital technologies in mediating environmental impacts. They argue that while technology can help to "de-materialise" economic activity (e.g., by replacing physical travel with teleconferencing), it can also lead to "re-materialise" effects or rebounds, where efficiency gains are offset by increased consumption [7]. A well-designed system like *AIDiaCAR* must be conscious of these potential rebounds and focus on promoting a net reduction in environmental impact, not merely more efficient consumption. This project, therefore, contributes to the field of Sustainable HCI by providing a practical case study in designing digital interventions for real-world ecological benefit.

2.3 Gap analysis and niche identification

The comprehensive review of the state of the art in both commercial and personal vehicle management software, as well as the underlying theoretical frameworks, reveals a distinct and underserved gap in the existing technological landscape. While highly specialized tools exist for corporate fleets, individual car enthusiasts, and generic navigation, no current solution holistically integrates their respective functionalities into a single, cohesive platform designed specifically for the collaborative context of a private, multi-vehicle household. This synthesis of capabilities represents the core contribution and unique value proposition of the *AIDiaCAR* project, as illustrated conceptually in Figure 2.7.

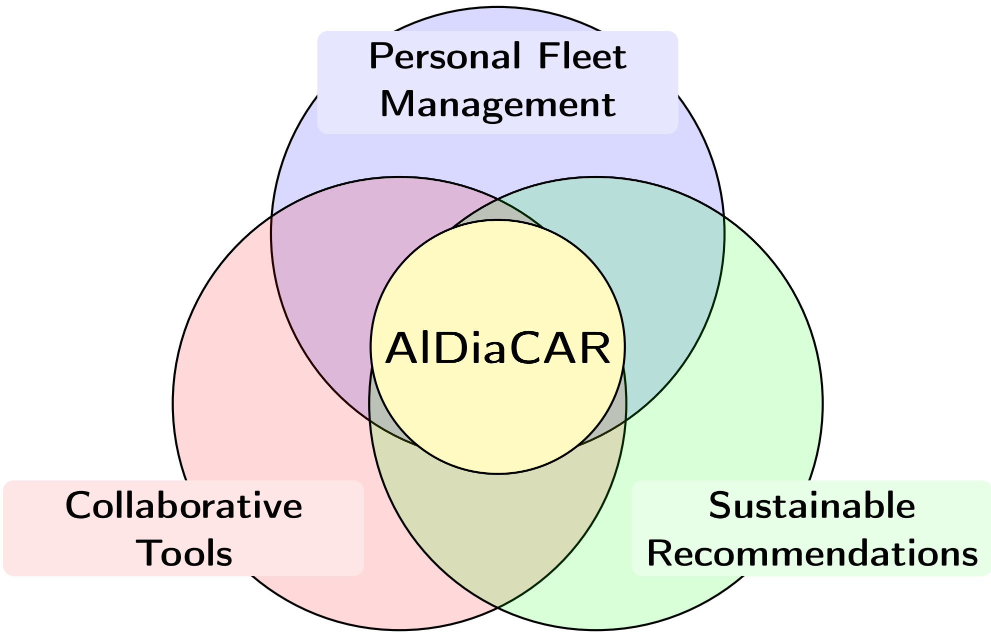


Figure 2.7: Visual representation of the market gap. AIDiaCAR is positioned at the intersection of Personal Fleet Management, Sustainable Recommendations, and Collaborative Tools, a niche not currently occupied by existing solutions.

To further delineate this unoccupied niche, Table 2.1 provides a detailed comparative analysis, summarizing the limitations of existing application categories against the key requirements derived from our problem statement.

Table 2.1: Detailed gap analysis of existing application categories

Application category	**Target user**	**Core functionality**	**Sustainability focus**	**Identified gaps for AIDiaCAR Project**
Commercial Fleet Management Systems	Professional Fleet Manager (B2B)	Logistics, cost control, driver surveillance, large-scale analytics.	Primarily financial (fuel cost reduction); operational efficiency.	Prohibitively expensive and complex for personal use; lacks collaborative, non-surveillance model; misaligned UX and social context.
Personal Maintenance Apps	Individual Car Enthusiast (B2C)	Manual logging of fuel, expenses, and service for a single vehicle.	Minimal; limited to calculating fuel economy for financial tracking.	No collaborative features for shared management; poor multi-vehicle comparison; passive, utilitarian engagement model.
Eco-Routing & Navigation Tools	General Driver / Commuter (B2C)	Calculating a fuel-efficient route for a single, generic journey.	High; focused on optimizing a single trip's route for a generic vehicle model.	Disconnected from user's actual vehicle fleet; cannot recommend the optimal vehicle, only the optimal route; lacks maintenance context.

Therefore, this thesis project, *AIDiaCAR*, is strategically positioned to fill this clearly defined niche. It is conceived as the first mobile application to holistically address the complex needs of an individual or family managing multiple vehicles by uniquely unifying three critical pillars of functionality into a single, user-friendly experience.

First, it provides robust **Personal fleet management** capabilities. This goes beyond the single-vehicle logbook model by offering a centralized, collaborative dashboard where all family members can view the real-time status, location, and maintenance needs of every vehicle in the household fleet. This directly addresses the critical pain point of "shared maintenance blindness" and reduces "coordination friction."

Second, the system incorporates **Integrated sustainability recommendations**. Unlike generic eco-routers, *AIDiaCAR* leverages the detailed data of the user's specific fleet. Its recommendation engine will consider not only the destination but also the unique emissions profile, fuel efficiency, maintenance status, and regulatory constraints of each available vehicle to suggest the holistically optimal car for every journey. This transforms the application from a passive data logger into an active decision-support tool.

Third, it implements a thoughtful **Behavioral reinforcement** layer through gamification. By embedding game-like elements such as badges, progress tracking, and positive feedback, the application aims to make sustainable choices and diligent vehicle care intrinsically rewarding. This actively encourages long-term habit formation, moving beyond mere utility to foster genuine user engagement and a collective household commitment to more sustainable mobility.

By seamlessly combining these three pillars, the *AIDiaCAR* project provides a novel and significant contribution to the field of Sustainable HCI. It offers a practical, user-centric tool designed to empower individuals and families to systematically reduce their personal transportation footprint, mitigate household friction, and manage their shared assets more efficiently and responsibly. This chapter has established the clear need and opportunity for such a solution, setting the stage for the subsequent chapters which will detail its design, implementation, and evaluation.

3 AIDiaCAR: System architecture and phased development

Having established the theoretical foundations and identified a distinct gap in the state-of-the-art of vehicle management applications in the preceding chapter, this chapter provides a comprehensive definition of the proposed solution: *AIDiaCAR*. This chapter will serve as the architectural blueprint for the project, articulating its core mission, foundational principles, and detailed feature set. The objective is to translate the conceptual framework into a tangible product specification that directly addresses the coordination frictions, maintenance blindness, and suboptimal decision-making prevalent in the modern multi-vehicle household, as exemplified by the Martínez family persona.

The project's vision is ambitious, encompassing a wide range of functionalities from real-time coordination to intelligent, data-driven recommendations and behavioral reinforcement. To manage this complexity and ensure a methodologically sound development process, this chapter also introduces a phased implementation roadmap. This iterative approach deconstructs the complete vision into three distinct, sequential stages: the Alpha version, representing the core viable product; the Beta version, which enhances the system with intelligent features and user engagement mechanics; and finally, the Zenith release, which embodies the fully realized, feature-complete expression of the AIDiaCAR concept. This structured methodology allows for focused development, iterative testing, and the progressive validation of the project's hypotheses at each stage.

3.1 The AIDiaCAR vision: A holistic framework

The core mission of AIDiaCAR is to serve as the intelligent, collaborative "Digital Garage" for the modern household. It is designed to seamlessly integrate vehicle management, user coordination, and sustainable decision-making into a single, cohesive, and effortless user experience. The system is architected upon four foundational pillars, each conceived to address a critical point of friction and inefficiency in the management of a shared, personal vehicle fleet. These pillars function interdependently to create a holistic solution that is more than the sum of its parts.

3.1.1 Visual identity and design principles

Before detailing the functional pillars, it is important to define the visual and interaction design philosophy that underpins the user experience. A consistent and thoughtful visual identity is critical for establishing user trust, ensuring usability, and creating an intuitive interface. The design of AIDiaCAR is guided by three core principles: clarity, accessibility, and encouragement. All visual elements, from the color palette to the typography, are selected to support these principles, creating a clean, modern, and non-intrusive user environment.

Figure 3.1 presents the project's style guide, which specifies the primary and secondary color palettes, the chosen fonts for headings and body text, and the application logo. The color scheme utilizes a base of calming blues and greens to evoke a sense of reliability and sustainability, with vibrant accent colors used purposefully to draw attention to key actions and notifications.

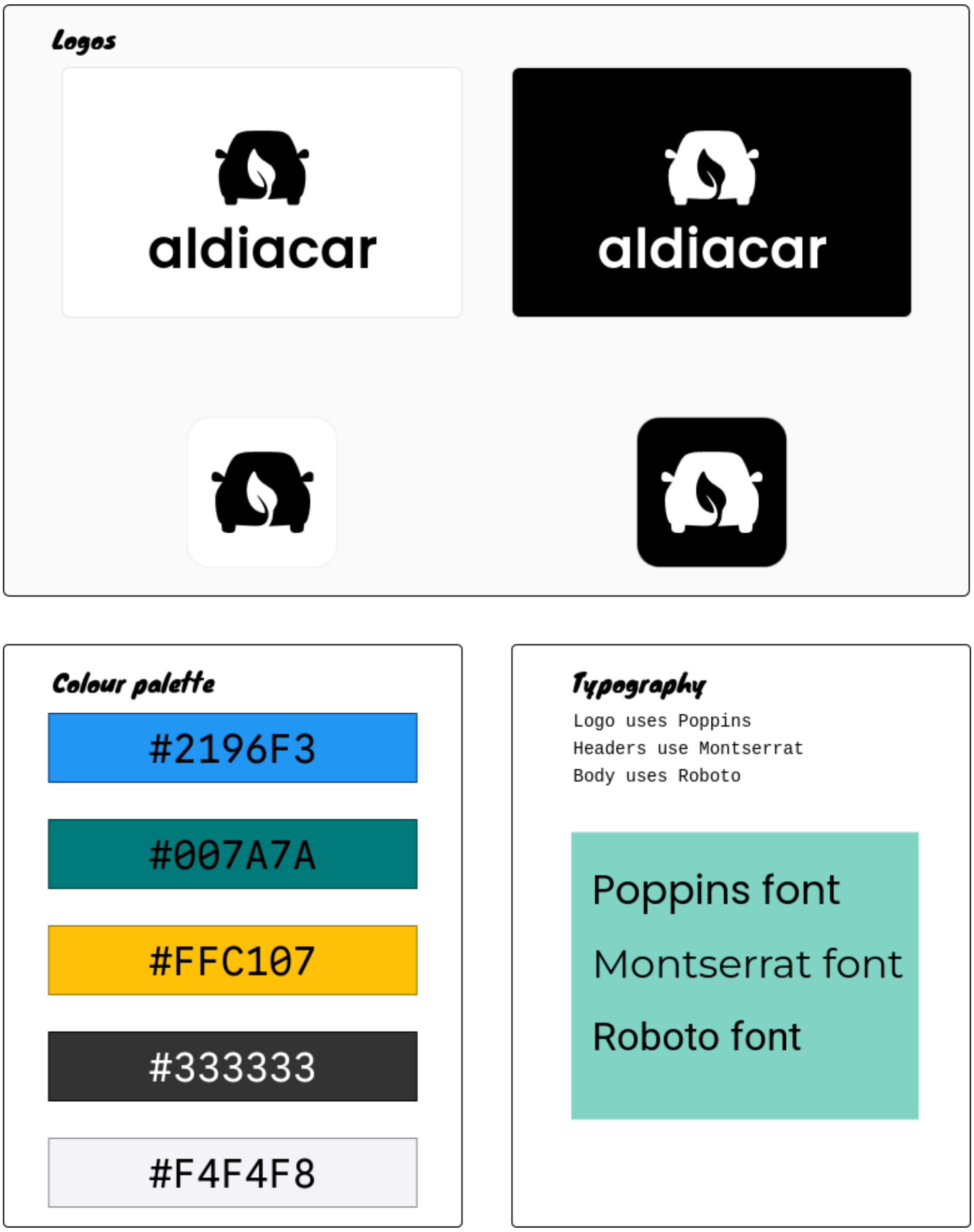


Figure 3.1: AlDiaCAR visual identity guide, defining the color palette, typography, and logo.

3.1.2 Pillar 1: The shared digital key rack (Coordination)

This foundational pillar is designed to definitively eradicate the persistent and friction-inducing "who has what car?" problem. It transforms the physical, opaque state of the household's vehicles into a transparent, digital, and universally accessible source of truth. The primary goal is to eliminate ambiguity and the need for constant, ad-hoc communication.

Key functionalities of this pillar include:

- **Real-time vehicle status:** At the heart of the application is a centralized dashboard where each vehicle in the household fleet is represented with a clear, instantly discernible status. These states are designed to be unambiguous: 'At Home' (available for use), 'In Use (by [Family Member Name])', 'Reserved' (for a future time), or 'Unavailable (Maintenance)'. This immediate visibility allows any family member to understand the state of their shared assets with a single glance.

- **Seamless check-out / check-in protocol:** To maintain the accuracy of the system, a lightweight protocol for tracking vehicle usage is implemented. When a user takes a vehicle, they perform a simple, one-tap "Check-Out" action within the application. This immediately updates the vehicle's status to 'In Use' and associates it with their user profile. Upon returning, a corresponding "Check-In" action reverts the status to 'At Home', making it available for others. This process is designed to be as frictionless as possible to encourage consistent adoption.

- **Proactive reservation system:** To facilitate forward planning and prevent scheduling conflicts, users are empowered to reserve a specific vehicle for a future date and time slot (e.g., "Reserve the SUV on Saturday from 14:00 to 17:30"). The system automatically updates the vehicle's timeline, displaying it as 'Reserved' during that period to all other users. This feature is critical for coordinating important events and ensuring that a required vehicle is available when needed, thus shifting the family's management paradigm from reactive to proactive.

3.1.3 Pillar 2: The proactive co-pilot (Intelligent recommendation)

This pillar represents the core intelligence of the AIDiaCAR system. It is designed to answer the complex, multi-faceted question: "What is the smartest vehicle choice for *this specific trip*?" It moves beyond simple availability to provide data-driven, context-aware decision support, making the optimal choice the easiest choice.

The engine's process involves a multi-factor trip analysis:

- **Multi-vector analysis engine:** When a user inputs their destination, the system's recommendation engine initiates a real-time analysis across several critical vectors. This process is designed to be instantaneous from the user's perspective.

  1. **Regulatory constraints analysis:** The engine first cross-references the destination's geographic coordinates with a constantly updated national database of Low Emission Zones (Zonas de Bajas Emisiones - ZBEs) and their specific access rules. It compares these rules against the stored 'distintivo ambiental' of each vehicle in the user's fleet, instantly filtering out any non-compliant options.

  2. **Economic optimization:** For all compliant vehicles, the engine calculates the estimated fuel or energy cost for the proposed journey. This calculation is based on the vehicle's pre-configured efficiency profile (L/100km or kWh/100km), the route distance, and real-time local fuel price data fetched from an external API.

  3. **Environmental impact assessment:** Concurrently, the system calculates the estimated total CO_2 emissions for the journey for each eligible vehicle. This provides users with a clear, quantifiable measure of the environmental consequence of their choice.

  4. **Maintenance status awareness:** The engine performs a critical check against the maintenance schedules defined in Pillar 3. If a vehicle is approaching a critical service interval (e.g., its mandatory ITV inspection is due next week, or its tires are within 50km of their replacement threshold), the application will flag this vehicle as "Not Recommended" for a long-distance trip, thereby preventing potential safety issues or legal infractions.

- **Holistic, user-centric recommendations:** After completing the analysis, the application does not present the raw data. Instead, it synthesizes the findings into a simple, ranked list of recommendations, such as: "Best Choice" (the optimal balance of all factors), "Eco Choice" (the lowest CO_2 emissions), and "Cheapest Choice." Each recommendation is accompanied by a clear, concise summary of its pros and cons, empowering the user to make a truly informed, one-glance decision that aligns with their immediate priorities.

3.1.4 Pillar 3: The omniscient mechanic (Automated maintenance)

This pillar is architected to be the household's collective, digital memory for vehicle health, ensuring that all automotive assets are consistently maintained to be safe, reliable, and operating at peak efficiency. It combats the "shared maintenance blindness" that often plagues multi-driver environments.

Its core functionalities are:

- **Smart maintenance profiling and scheduling:** The system maintains a comprehensive and distinct maintenance profile for each vehicle. This includes standard items like mandatory technical inspections (ITV), oil changes, and tire rotations, as well as user-definable custom reminders (e.g., "Check brake fluid"). For each item, the system tracks due dates based on both fixed time intervals (e.g., annually) and dynamic mileage data, which is updated with each logged trip. This allows for accurate, predictive scheduling of required services.

- **Shared alerts and collaborative task management:** When a maintenance event is approaching its due date, the system generates a notification that is sent not to a single individual, but to a shared "Household Tasks" list visible to all family members. An alert such as "ITV due for Sedan in 15 days" becomes a collective responsibility. Any member of the household can take ownership of the task, and once completed, mark it as such. This closes the communication loop and ensures everyone is aware that the maintenance has been handled. The system can also be configured to suggest nearby approved service centers or ITV stations via integration with mapping services.

3.1.5 Pillar 4: The household impact dashboard (Gamification & analytics)

This final pillar addresses the motivational and behavioral aspects of sustainable mobility. It transforms the abstract data generated by the family's daily activities into tangible insights, shared goals, and a system of positive reinforcement, leveraging the principles of gamification and persuasive technology.

The key components include:

- **Unified impact dashboard:** A central screen in the application serves as the household's command center for analytics. It displays key performance metrics in a clear, graphical format. These include the total monthly transportation expenditure, the household's aggregate CO_2 emissions for the month, and a composite "Household Eco-Score" that provides an at-a-glance measure of their overall sustainability.

- **Personal and team-based achievements:** To foster engagement, the system incorporates a dual-layered achievement system. It includes individual achievements designed to reward personal positive actions (e.g., "Eco-Warrior: Your first 100% eco-friendly week!"). Crucially, it also features collaborative household goals (e.g., "Team Effort: Reduce our collective emissions by 10% this month compared to last"). This encourages cooperation and transforms sustainability into a shared family objective.

- **Actionable and insightful analytics:** The dashboard goes beyond simply displaying raw data. It is designed to provide clear, actionable insights that connect choices to consequences. For example, the system might generate a card stating, "Insight of the Month: Choosing the hatchback over the SUV for short urban trips saved the family €45 and prevented 50kg of CO_2 emissions." This direct feedback loop makes the economic and environmental benefits of sustainable choices tangible, visible, and rewarding, reinforcing positive behavior over the long term.

3.2 Phased implementation roadmap

To systematically construct the comprehensive system described above, a phased development roadmap is proposed. This iterative methodology deconstructs the full feature set into three manageable and logically sequential versions. Each phase builds upon the last, allowing for focused development, targeted testing, and the potential for user feedback to inform subsequent stages. The three phases are designated as the Alpha version, the Beta version, and the Zenith release.

3.2.1 The Alpha version: Establishing the core viable product

The primary objective of the Alpha phase is to build and validate the foundational functionality of AIDiaCAR. This version is conceived as the Minimum Viable Product (MVP), focusing exclusively on solving the most acute and immediate pain points: coordination friction and shared maintenance blindness. The feature set is deliberately constrained to what is essential for the system to be useful and to prove the core concept.

Features of the Alpha version:

- **Pillar 1 (Coordination):**
  - User and multiple vehicle registration.
  - A simple, clear dashboard showing the status of each vehicle.
  - Manual, one-tap "Check-Out" and "Check-In" functionality to update a vehicle's status between 'At Home' and 'In Use'. The user who checked out the vehicle is displayed.
- **Pillar 2 (Recommendation):**
  - This pillar is **not implemented** in the Alpha version. The focus is on coordination, not decision support, at this initial stage.
- **Pillar 3 (Maintenance):**
  - A dedicated section for each vehicle to manually log key maintenance dates (e.g., next ITV, next oil change).
  - Simple, time-based push notifications to all users when a logged date is approaching (e.g., "ITV for Sedan is due in 30 days").
- **Pillar 4 (Gamification):**
  - This pillar is **not implemented** in the Alpha version. The focus is on pure utility before engagement mechanics are introduced.

The Alpha version, in essence, is a digital replacement for the family's whiteboard or group chat, but with the added structure and reliability of a dedicated application. Its successful implementation would validate the core user need for a centralized coordination and information hub.

3.2.2 The Beta version: Enhancing intelligence and engagement

Building upon the stable and validated foundation of the Alpha version, the Beta phase introduces the "smart" features that define AlDiaCAR's unique value proposition. The focus shifts from merely logging status to actively assisting in decision-making and beginning to foster user engagement.

Features of the Beta version (Includes all Alpha features, plus):

- **Pillar 1 (Coordination):**
  - Introduction of the vehicle **Reservation System**, allowing users to book cars for future time slots.
- **Pillar 2 (Recommendation):**
  - Implementation of the **initial Recommendation Engine**.
  - Users can input a destination, and the app will perform a basic check.
  - The engine's initial capability will be limited to **ZBE constraint analysis**. It will identify which vehicles are legally permitted to travel to the destination and present a simple "Compliant" or "Not Compliant" status.
- **Pillar 3 (Maintenance):**
  - Enhanced maintenance tracking that includes **mileage-based intervals**. Users can log their trip mileage, which the system uses to provide more accurate service reminders.
- **Pillar 4 (Gamification):**
  - Introduction of the **first layer of gamification**.
  - A basic dashboard showing total trips and distance per vehicle.
  - A system of **individual achievements and badges** for positive actions, such as consistently logging trips or completing maintenance on time.

The Beta version transforms AIDiaCAR from a simple utility into an intelligent assistant. It begins to deliver on the promise of smarter, more informed choices and introduces the motivational mechanics intended to drive long-term adoption.

3.2.3 The Zenith release: The complete vision

The Zenith release represents the culmination of the development roadmap, a feature-complete version of the application that fully realizes the initial four-pillar vision. It integrates all planned functionalities, creating a deeply interconnected and powerful tool for holistic household fleet management.

Features of the Zenith release (Includes all Beta features, plus):

- **Pillar 1 (Coordination):**
  - Polished UI/UX with potential integration into third-party calendar applications for seamless reservation management.

- **Pillar 2 (Recommendation):**
  - The **full, multi-factor Recommendation Engine** is implemented.
  - The engine analyzes trips based on ZBE constraints, **estimated fuel cost** (with real-time price data), **CO_2 emissions**, and **upcoming maintenance status**.
  - The app presents the final, synthesized recommendations ("Best Choice," "Eco Choice," etc.) with clear justifications.
- **Pillar 3 (Maintenance):**
  - Implementation of the shared, collaborative **Household Task List** for maintenance items.
  - Integration with mapping services to **suggest nearby service centers** and ITV stations.
- **Pillar 4 (Gamification):**
  - The complete **Household Impact Dashboard** is launched.
  - It displays aggregate metrics for household cost, emissions, and an Eco-Score.
  - It features **collaborative household goals** in addition to individual achievements.
  - The **insightful analytics engine** is activated, providing users with tangible feedback on the positive impact of their choices.

The Zenith release is the ultimate expression of AIDiaCAR: an intelligent layer that removes friction, saves money, enhances safety, and systematically guides the entire family toward more sustainable mobility patterns by making the most responsible choice the easiest and most rewarding one.

4 System design and architecture

This chapter details the architectural design of the application, named *AlDiaCAR*[1]. The design is guided by the project's core objectives: to provide a cross-platform, scalable, and maintainable solution for optimizing an individual's personal vehicle use with a focus on sustainability. We will discuss the high-level architecture, break down the system into its primary components, define the data flow, justify the technology stack, and outline the logic for the sustainability and gamification features.

4.1 High-level architecture

The system is designed following a classic client-server architecture, which decouples the user interface from the core business logic and data storage. This model enhances security, scalability, and allows for multiple types of clients to be developed in the future without altering the backend.

The main components are:

- **Frontend client:** A cross-platform mobile application built with React Native and the Expo framework. It is responsible for all user interactions, data presentation, and communication with the backend.
- **Backend server:** A Node.js application using the Express.js framework. It serves as the central hub, handling business logic, user authentication, data processing, and communication with the database.
- **Database:** A MongoDB NoSQL database that stores all persistent data, including user profiles, vehicle details, and trip logs.
- **Mock API:** A secondary Node.js server that simulates a third-party API for retrieving vehicle specifications, ensuring development and testing can occur offline and with deterministic data.

[1] A placeholder name for the application, derived from "Al Día" (up-to-date) and "Car".

Figure 4.1 provides a visual representation of this architecture and the data flow between components within the development environment.

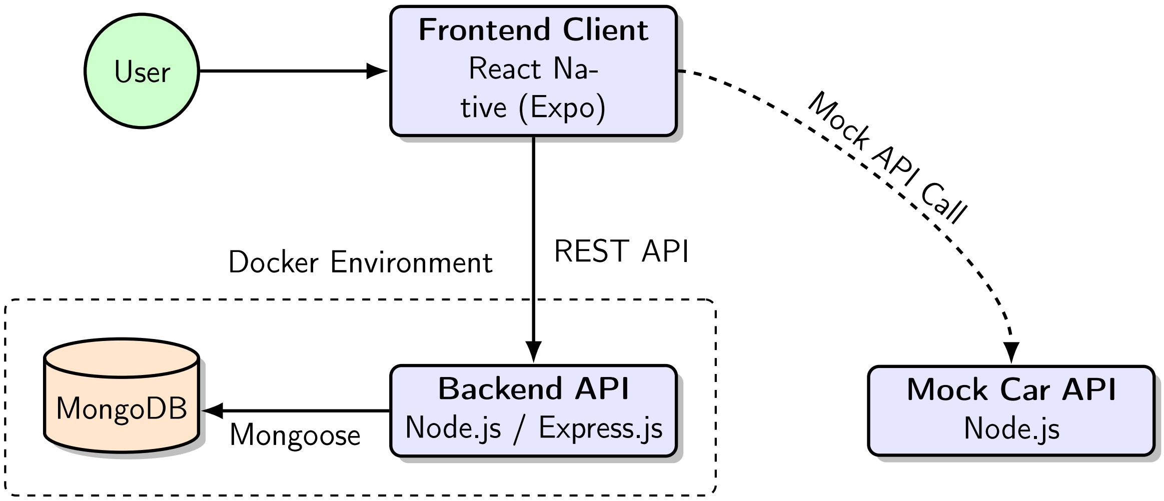


Figure 4.1: High-level system architecture, including the development environment.

4.2 Component overview

The system is logically divided into frontend and backend components, each with distinct responsibilities that reflect the project's source code organization.

4.2.1 Frontend components (React Native)

The frontend is structured as a mobile application using Expo, with a file-based routing system. The key components, located in `src/frontend/app/`, are:

- **Authentication (‘/app/(auth)‘):** Screens for user registration and login.
- **Main application tabs (‘/app/(tabs)‘):** The core user-facing part of the app:
  - **Home (‘/home‘):** A dashboard showing upcoming maintenance alerts.
  - **Vehicles (‘/vehicles‘):** A module for a user to manage their personal fleet, including adding, viewing, and editing vehicles.
  - **Routes (‘/routes‘):** The interface for the recommendation system.
  - **Statistics (‘/stats‘):** Implements gamification and data visualization, showing user stats and earned achievements.

- **State management (‘/context/AuthContext.tsx‘):** A React Context that manages the global authentication state, making user session data available throughout the component tree.
- **Localization (‘/localization‘):** Supports internationalization with dedicated folders for English and Spanish, allowing for easy translation of all UI text.

4.2.2 Backend components (Node.js)

The backend, located in `src/backend/`, is a RESTful API built with Node.js and Express.js. Its structure promotes separation of concerns:

- **Routes (‘/routes‘):** Defines the API endpoints. The project uses dedicated files for each resource (e.g., `authRoutes.js`, `vehicleRoutes.js`, `maintenanceRoutes.js`).
- **Controllers (‘/controllers‘):** Contains the business logic for each route. For example, `authController.js` handles registration and login, while `maintenanceController.js` contains the logic to calculate and sort vehicle alerts.
- **Models (‘/models‘):** Defines the data schemas for MongoDB using Mongoose. The core models are `User.js`, `Vehicle.js`, and `Trip.js`.
- **Middleware (‘/middleware‘):** Contains functions that process requests before they reach the controller. `authMiddleware.js` is crucial for protecting routes by verifying the user's JWT.

4.2.3 Data model

The NoSQL data model, implemented in MongoDB, was chosen for its flexibility. The primary collections are `Users`, `Vehicles`, and `Trips`. Relationships are managed through object references (e.g., a `Trip` document contains a reference to the `_id` of the `User` and `Vehicle`). This structure is well-suited for the application's needs, as vehicle attributes and maintenance requirements can vary significantly. A diagram of the data model is presented in Figure 4.2.

4.3 Data flow and API design

Communication between the frontend and backend occurs over HTTPS via a RESTful API using standard HTTP methods and JSON. A typical data flow for a core feature, such as getting a vehicle recommendation, is as follows:

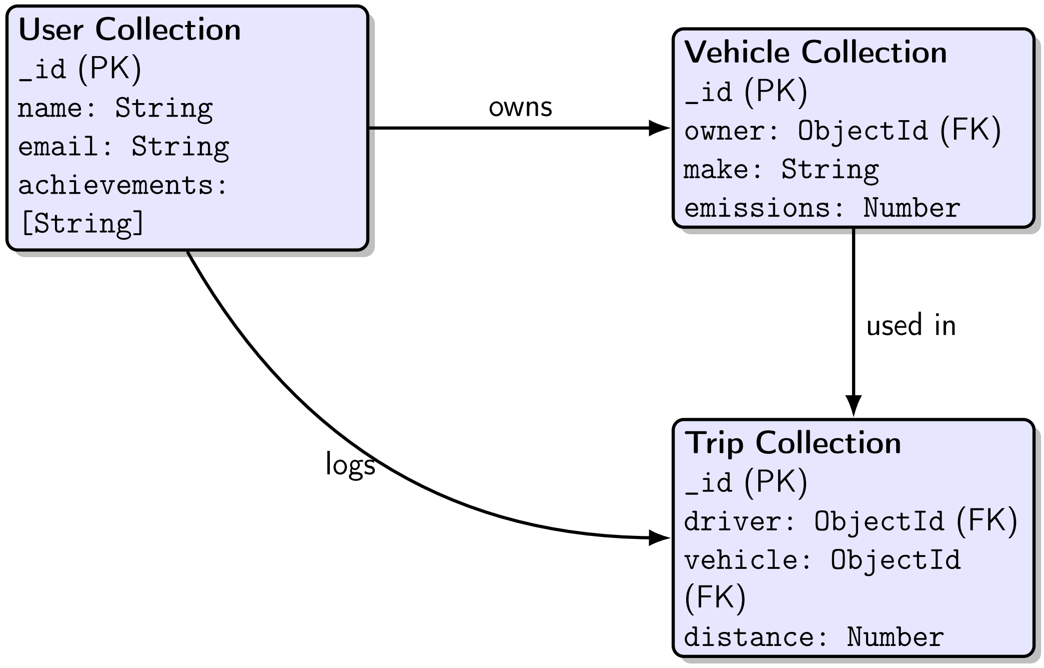


Figure 4.2: Simplified Data Model illustrating the relationships between the primary collections.

1. The user enters a trip distance on the "Routes" screen and requests a recommendation.
2. The React Native client sends a POST request to the backend endpoint `/api/recommendations`. The browser automatically includes the JWT cookie with the request.
3. The backend's authentication middleware intercepts the request and verifies the JWT.
4. The recommendation controller fetches all vehicles associated with that user from the database.
5. The controller's algorithm calculates the estimated emissions for each vehicle and responds with a sorted JSON array.
6. The frontend parses the response and displays the recommendations to the user.

A complete specification of all API endpoints is available in Appendix A.

4.4 Non-functional requirements

Beyond the core features, the system was designed with several non-functional requirements in mind:

- **Cross-platform compatibility:** React Native and Expo ensure a single codebase can be deployed to both iOS and Android, as well as the web.
- **Scalability:** The backend is stateless, allowing for horizontal scaling. MongoDB also offers robust scaling capabilities.
- **Security:** User authentication is handled using JWTs stored in secure, HTTP-only cookies. Passwords are never stored in plain text, only as bcrypt hashes.
- **Maintainability:** The project is structured into logical modules with a clear separation of concerns. This modularity, combined with TypeScript, improves code quality and makes future maintenance easier.
- **Usability:** The application provides internationalization support and aims for a clean, intuitive user interface.

4.5 Technology choices justification

- **React Native (with Expo):** Chosen for its ability to build native-quality applications from a single JavaScript/TypeScript codebase. Expo further accelerates development by simplifying the build process and managing native project configuration.
- **Node.js with Express.js:** Selected for the backend due to its high performance with asynchronous I/O, making it ideal for an API-driven application. Its use of JavaScript creates language synergy with the frontend.
- **MongoDB:** A NoSQL database was preferred over a relational one due to its flexible schema, which is advantageous for storing vehicle data where attributes and maintenance items can vary.
- **Docker:** The inclusion of a `docker-compose.yaml` file facilitates a consistent and isolated development environment, eliminating "it works on my machine" issues.
- **Jest & Playwright:** Selected for testing. Jest is a standard for unit testing Node.js applications, while Playwright is a powerful tool for E2E testing that simulates real user interactions.

4.6 Gamification and sustainability logic

The logic for these features is implemented on the backend to ensure consistency and security.

- **Sustainability recommendation:** The algorithm calculates total CO_2 emissions for a trip using the formula:

$$\text{Emissions } (\text{gCO}_2) = \text{Distance (km)} \times \text{Emission Factor } (\text{gCO}_2/\text{km}) \tag{4.1}$$

  The system recommends the vehicle with the lowest calculated emissions.

- **Gamification:** The backend tracks user actions to award achievements. For example, upon creating the first vehicle, the 'FIRST_VEHICLE' key is added to the user's `achievements` array in the database. When a trip is logged that surpasses a distance milestone (e.g., 1000 km), the corresponding achievement key is granted.

5 Implementation details

This chapter transitions from the abstract design of Chapter 3 to the concrete technical implementation of the *AIDiaCAR* system. It provides a detailed walkthrough of the project's source code, explaining the structure, key modules, and programming patterns used in both the backend and frontend.

5.1 Project structure overview

The project is organized as a monorepo, a single repository containing all the code for the system. This approach simplifies dependency management and streamlines development across the different parts of the application. The high-level directory structure is shown in Figure 5.1.

The `src` directory cleanly separates the `backend` and `frontend` concerns. The `docker` directory ensures a reproducible development environment, while the `tests` directory houses the quality assurance framework.

5.2 Backend implementation (Node.js)

The backend is a RESTful API built with Node.js and the Express.js framework. It is responsible for all business logic, data persistence, and security.

5.2.1 Data models (Mongoose)

Data schemas are defined in `src/backend/models/` using Mongoose. The principal models are `User.js`, `Vehicle.js`, and `Trip.js`. A key feature is the use of Mongoose's `timestamps` option in the User schema to automatically manage `createdAt` and `updatedAt` fields. The password field uses `select: false` to prevent it from ever being sent in an API response. The full schema definitions are detailed in Appendix B.

```
├── config/ (Docker service configurations)
│   └── mongo-express-config.js
├── docker/ (Local environment definition)
│   └── docker-compose.yaml
├── src/ (Main source code)
│   ├── backend/ (Node.js REST API)
│   │   ├── controllers/
│   │   ├── middleware/
│   │   ├── models/
│   │   ├── routes/
│   │   └── server.js
│   └── frontend/ (React Native mobile app)
│       ├── app/ (Expo file-based routes)
│       ├── components/
│       ├── context/
│       └── localization/
├── tests/ (Automated tests)
│   ├── backend/ (Jest tests)
│   └── frontend/ (Playwright E2E tests)
└── package.json (Root monorepo scripts)
```

Figure 5.1: High-level project directory structure.

5.2.2 Controllers and business logic

Controllers, located in `src/backend/controllers/`, contain the core logic for each API endpoint. While many controllers perform standard CRUD operations, some, like `tripController.js`, encapsulate more complex business logic. The `logSimulatedTrip` function demonstrates how a single API call can trigger multiple state changes across the system.

Listing 5.1: Core logic from `tripController.js`

```
// 1. UPDATE VEHICLE MAINTENANCE METRICS
if (vehicle.upcomingMaintenance) {
  if (vehicle.upcomingMaintenance.brakes?.distance) {
    vehicle.upcomingMaintenance.brakes.distance -= distance;
  }
  // ... other distance-based counters are reduced
}
```

```
// 2. CREATE THE TRIP RECORD
const calculatedEmissions = distance * (vehicle.emissions || 150);
await Trip.create({ driver: userId, vehicle: vehicleId, distance, ... })

// 3. UPDATE USER STATS & GRANT ACHIEVEMENTS
user.stats.distanceTraveled += distance;
const achievementsToGrant = [];
if (!user.achievements.includes('FIRST_TRIP')) {
  achievementsToGrant.push('FIRST_TRIP');
}
if (oldDistance < 1000 && user.stats.distanceTraveled >= 1000) {
  achievementsToGrant.push('DIST_1000');
}
if (achievementsToGrant.length > 0) {
  user.achievements.push(...achievementsToGrant);
}

// 4. SAVE ALL CHANGES ATOMICALLY
await Promise.all([vehicle.save(), user.save()]);
```

This function shows how logging a trip correctly decrements maintenance counters on the vehicle, creates a new trip record, and updates the user's statistics and achievements, all within a single transaction. Figure 5.2 illustrates this interaction.

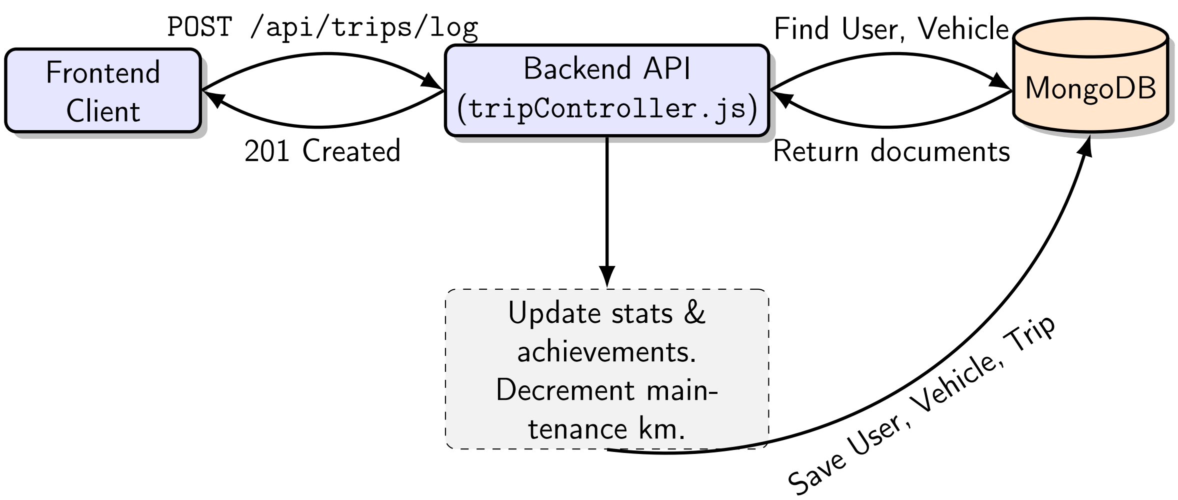


Figure 5.2: Sequence diagram for the trip logging data flow.

5.2.3 API routes and middleware

Routes defined in `src/backend/routes/` map the API endpoints to controller functions. The most critical middleware, `authMiddleware.js`, protects these routes. It extracts the JWT from an `HttpOnly` cookie, verifies it, and attaches the user's ID to the request object. This makes the user's identity available to any protected controller.

5.3 Frontend implementation (React Native)

The frontend is a cross-platform mobile application developed using React Native with the Expo framework and TypeScript for static typing.

5.3.1 Navigation and screen structure

The application uses Expo's file-based router. Directories and files within `src/frontend/app/` automatically become routes.

- **Layouts ('_layout.tsx'):** These files define the shell UI, such as the main tab bar defined in `app/(tabs)/_layout.tsx`.
- **Authentication flow ('app/(auth)'):** A route group for the login and register screens, active when a user is not authenticated.
- **Main screens ('app/(tabs)'):** Represent the core features. A key pattern used is `useFocusEffect` from Expo Router to re-fetch data whenever a screen comes into view, ensuring data is always fresh after an update (e.g., after adding a new vehicle).

5.3.2 State management and API communication

Global state for user authentication is managed via React's Context API in `context/AuthContext.tsx`. This provider stores the user's authentication status and profile information. Communication with the backend is handled using the `axios` library, which is configured with `withCredentials: true` to automatically handle the sending and receiving of authentication cookies.

5.4 Integration and deployment

5.4.1 Local development with Docker

To ensure a consistent development environment, the project utilizes Docker and Docker Compose. The `docker/docker-compose.yaml` file defines all services required to run the application stack locally, including the backend, frontend, database, and the Mongo Express GUI. A developer can start the entire stack with a single command: `docker-compose up`.

5.4.2 Deployment strategy

While the current focus is a robust prototype, a potential production deployment strategy would involve:

- **Backend:** Containerizing the Node.js application and deploying it to a Platform-as-a-Service (PaaS) like Heroku or a container orchestrator.
- **Database:** Using a managed database service like MongoDB Atlas.
- **Frontend:** Building the mobile application for production using Expo Application Services (EAS) and submitting the binaries to the app stores.

5.5 Testing strategy and implementation

The `tests/` directory houses a multi-layered testing strategy. For full details on the specific test cases, see Appendix D.

- **Backend unit & Integration tests:** The `tests/backend/` directory uses Jest to test modules in isolation and Supertest to test API endpoint integration.
- **Frontend E2E tests:** The `tests/frontend/` directory is configured for Playwright, which runs automated tests that simulate key user journeys in a real browser environment.

5.6 Security and privacy considerations

Security is implemented through several mechanisms:

- **Password hashing:** The `bcryptjs` library is used to securely hash and salt user passwords.
- **Authentication:** JWTs are stored in secure, `HttpOnly` cookies, which helps mitigate cross-site scripting (XSS) attacks.
- **Route protection:** The `authMiddleware` ensures that only authenticated users can access protected endpoints.

- **Environment variables:** Sensitive information like database connection strings and JWT secrets are managed via `.env` files and are not committed to version control.

6 Results and validation

This chapter presents the results of the implementation, validating the functionality of the AIDiaCAR prototype against the objectives defined in Chapter 1. The validation process confirms that the system's key features are operational and effectively address the problem statement outlined for our user persona, Alejandro Martínez. This chapter provides visual evidence of the working frontend, complemented by a summary of the automated tests that verify the backend logic.

6.1 Frontend validation via core user workflows

The most direct way to validate the application is to walk through the essential user journeys. The following sections use screenshots from the running application to demonstrate that the primary features have been successfully implemented.

6.1.1 User registration and authentication

The first step for any user is to create an account and log in. The system's authentication flow, managed by the components in `/app/(auth)`, was tested manually and via Playwright's E2E tests. Figure 6.1 shows the application's entry point, and Figure 6.2 shows the user's profile after a successful login, confirming the authentication and data retrieval process is working.

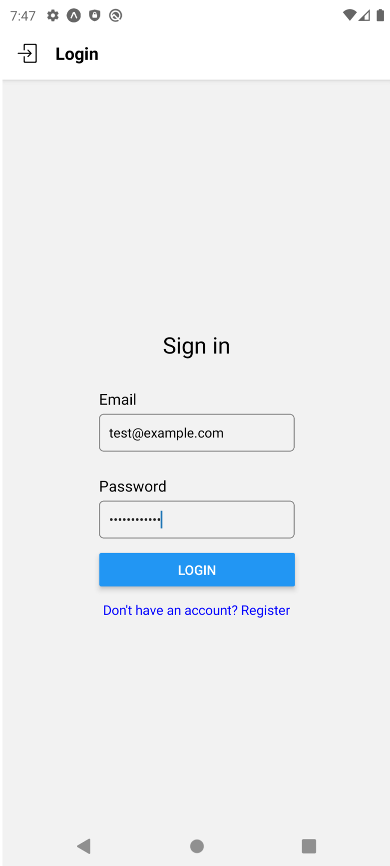


Figure 6.1: The user login screen.

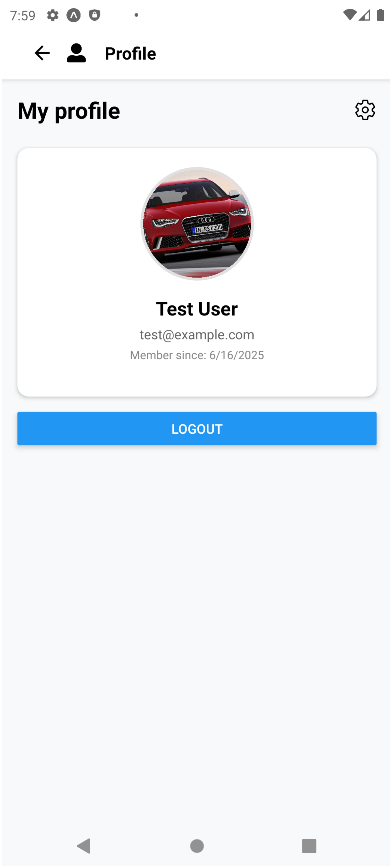


Figure 6.2: The user profile screen after successful authentication.

6.1.2 Vehicle fleet management

A core requirement for Alejandro is to manage his personal fleet. The application provides a seamless interface to add and view his cars. Figure 6.3 shows the form used to add a vehicle, and Figure 6.4 shows the central list of all his registered vehicles, providing a complete overview of his fleet. This validates the vehicle CRUD (Create, Read, Update, Delete) functionality.

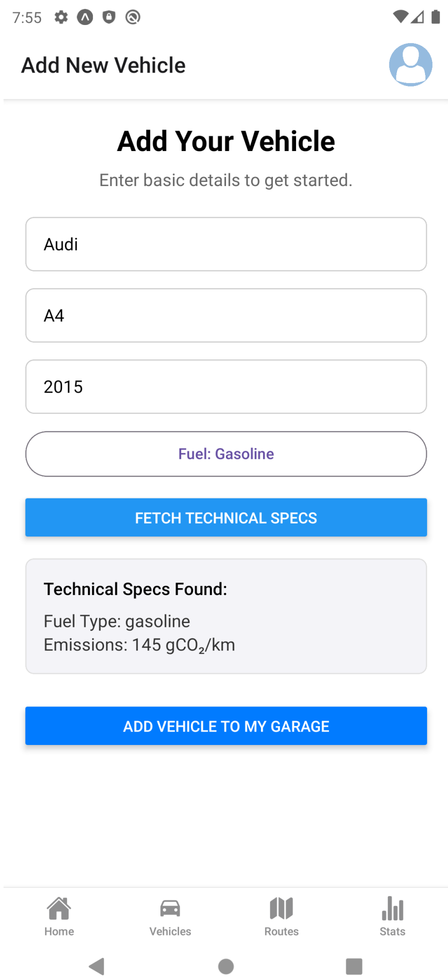


Figure 6.3: Adding a new vehicle to the user's fleet.

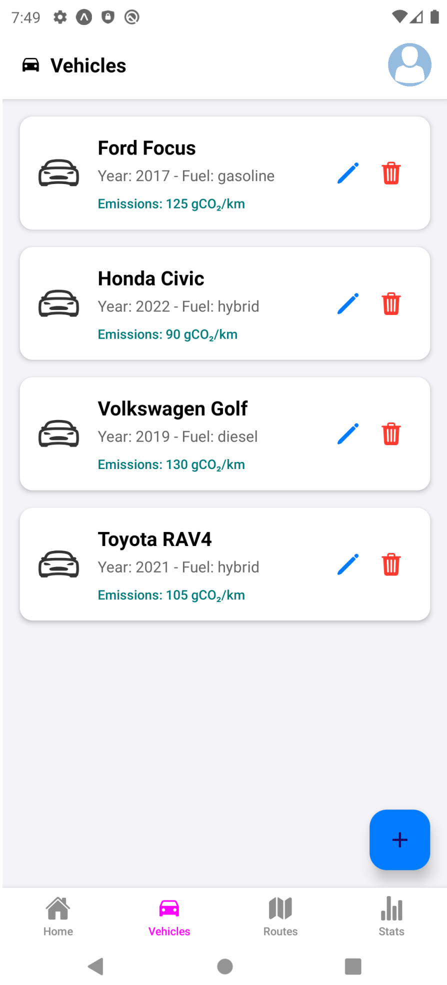


Figure 6.4: The user's list of registered vehicles.

6.1.3 Sustainability and maintenance recommendations

The application's primary goal is to provide actionable, data-driven advice. Figure 6.5 demonstrates the recommendation system in action, suggesting the most eco-friendly vehicle for a trip. Figure 6.6 shows the main dashboard, which displays sorted maintenance alerts, directly addressing Alejandro's pain point of overlooking important tasks.

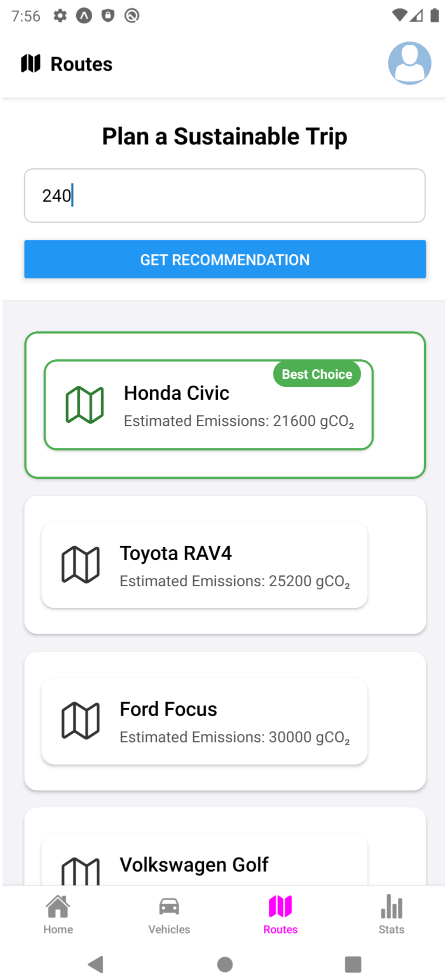


Figure 6.5: The vehicle recommendation interface.

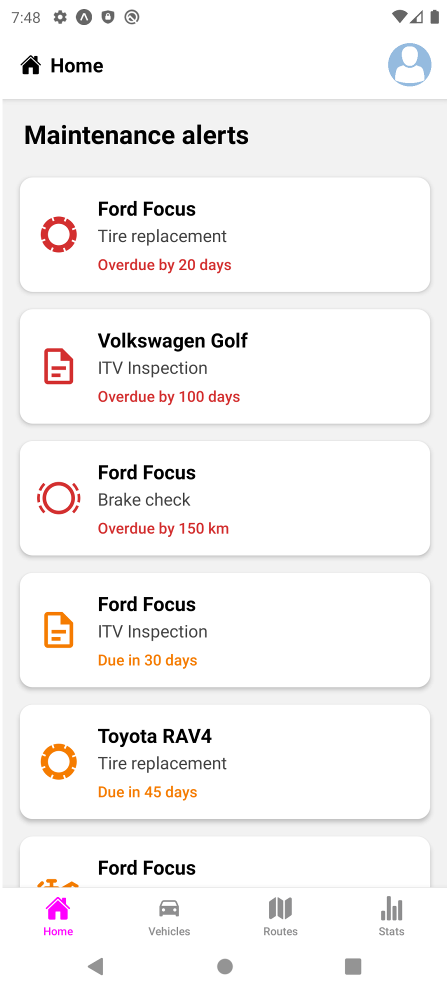


Figure 6.6: The main dashboard displaying maintenance alerts.

6.2 Backend logic and API validation

To validate the backend logic independently of the frontend, a comprehensive suite of automated integration tests was developed using Jest and Supertest. These tests verify the correctness of the API endpoints, business logic, and database interactions in a controlled environment.

A key example from this test suite is the validation of the sustainability recommendation endpoint. The test case, shown below, verifies that the API correctly processes an authenticated request and returns a properly calculated and sorted list of vehicle recommendations, which is a core piece of the application's business logic.

Listing 6.1: Integration test for the recommendation API from `tests/backend/features.test.js`

```
it('should return a sorted list of vehicle recommendations', async () =>
  const res = await request(app)
    .post('/api/recommendations')
    .set('Cookie', `jwt=${token}`) // Manually set auth cookie
    .send({ distance: 100 });

  expect(res.statusCode).toEqual(200);
  expect(Array.isArray(res.body)).toBe(true);

  // Verify the business logic: is the list sorted by emissions?
  expect(res.body[0].totalEmissions).toBeLessThanOrEqual(res.body[1].tot
});
```

This test confirms not only that the endpoint is reachable and secure, but also that its complex logic—calculating emissions for multiple vehicles and sorting them—is functioning as designed. A full summary of the test strategy is available in Appendix D.

6.3 Validation summary

The combination of visual confirmation from the frontend and successful execution of the automated test suites confirms that the prototype successfully implements its core objectives. Table 6.1 maps the project objectives to their implemented and validated status.

Table 6.1: Summary of Validated Project Objectives

Objective	Validation Method	Status
Cross-platform accessibility	Manual testing & Playwright E2E tests on web browser	Implemented
Vehicle management (CRUD)	Frontend Screenshots, Backend tests (Jest), E2E test (Playwright)	Implemented
Authentication system	Frontend Screenshots, Backend tests (Jest), E2E test (Playwright)	Implemented
Maintenance automation	Frontend Screenshots, Backend integration test (Jest)	Implemented
Sustainability recommendation	Frontend Screenshots, Backend integration test (Jest)	Implemented
Gamification system	Backend integration test for achievements (Jest)	Implemented

Bibliography

[1] Hannah Ritchie. Sector by sector: where do global greenhouse gas emissions come from? *Our World in Data*, 2020. https://ourworldindata.org/ghg-emissions-by-sector.

[2] Eurostat. Passenger cars per 1 000 inhabitants reached 560 in 2022. 2024. https://ec.europa.eu/eurostat/web/products-eurostat-news/w/ddn-20240117-1.

[3] International Energy Agency. Fuel economy in major car markets: Technology and policy drivers 2005-2019. Technical report, IEA, 2021. https://www.iea.org/reports/fuel-economy-in-major-car-markets.

[4] B.J. Fogg. *Persuasive Technology: Using Computers to Change What We Think and Do*. Morgan Kaufmann, 2002.

[5] Sebastian Deterding, Dan Dixon, Rilla Khaled, and Lennart Nacke. Gamification: Toward a definition. *CHI 2011 Workshop*, 2011.

[6] Juho Hamari, Jonna Koivisto, and Harri Sarsa. Does gamification work?–a literature review of empirical studies on gamification. *HICSS*, 2014.

[7] Frans Berkhout and Julia Hertin. De-materialising and re-materialising: digital technologies and the environment. *Futures*, 36(8):903–920, 2004.