figs/Implementation/workshopmanager/report.pdf.pdffigs/Implementation/workshopmanager/report.pdf-
.pdf- width=!,height=!,pages=-

width=!,height=!,pages=-

Conteúdo

Lista de Figuras

Lista de Tabelas

Glossário

CAPÍTULO 1

Introduction

This chapter establishes the foundation for the dissertation by presenting the motivation behind the work, the main objectives to be achieved, and the problems that guided the development of the proposed solution. The rapid growth of vehicle-sharing services, combined with the increasing demand for sustainable transport, has intensified the need for efficient maintenance systems. However, many service providers still rely on outdated or manual approaches that can lead to inefficiencies, errors, and ultimately a loss of customer trust.
To address these challenges, this dissertation proposes the design and implementation of a web-based platform to support the daily operations of LightMobi's dealerships. The chapter begins by discussing the market context and the reasons why maintenance management is a critical concern. It then introduces the objectives of the system and highlights the problems that arise in current practices. Finally, it outlines the structure of the thesis, providing the reader with a roadmap of how the work is organized.

1.1 Motivation

The vehicle-sharing market has grown significantly since 2021 due to the global interest in sustainability and environmental issues. **cohesionOpenData bike_data_businessresearch** The bike-sharing market, in particular, has gained popularity from governments worldwide, which are investing in cycling infrastructure such as cycling lanes, secure parking facilities, bicycle production and repair industries **Clercq2023 Cerro2024 European_declararion_on_cycling bike_data_businessresearch cohesionOpenData**. This attention to this sector raises the need for vehicle maintenance infrastructure that may be outdated and inefficient. **MAS_MOTORS** This low quality of service can lead to a significant decrease in the loyalty of the customers, which is the main source of income. **Setting_the_after_sale_process** This dissertation proposes a solution to solve this problem.

The fast progress of technology can prove to be a valuable ally in this matter. From the use of IoT devices with machine learning algorithms to monitor the vehicle's health **Vasavi2021**,

to the use of task management software to manage the maintenance tasks **MAS_MOTORS**, there is a large range of solutions that can be implemented to improve the efficiency of the maintenance process.

This dissertation presents a software solution that facilitates and organizes the work at the dealerships of the company LightMobie. We will gather the solution requirements by communicating with the company workers and them validate the solution with orddinary users and the company EMEL, that is an entity that operates and manage the bike sharing services of the public system of Gira.

1.2 Objectives and problems

The work in a vehicle maintenance service provider may be organized by manual input with basic applications. **MAS_MOTORS** This may introduce accidental human errors, which in turn lead to break of promises or unsatisfaction from the customer. **MAS_MOTORS Setting_the_after_sale_process** This type of approach may not be very prejudiced in a small business, but the continuous growth of the company requires a more modern method to handle the huge amount of work and achieve continuous success. **MAS_MOTORS**

This dissertation will focus on the development of a web application that facilitates and increases the performance of the work at the dealership. To accomplish this, the application will allow, simultaneously, the sharing of information, and manage and control of a shared vehicle dealership network. It will also interact with a factory data warehouse to gather information and store data generated from the maintenance process. This dissertation presents a significant challenge related to the users who work at the dealerships and garages. These users may be already accustomed to their systems or their manual work's labor. The introduction of a new system may be a challenge to the users, so the system must be user-friendly, easy to use, and provide significant value in their work. Another requirement is the integration with the fleet management platform of LightMobie, a project that has been in development for a few years with a team in **IEETA!** (**IEETA!**) at the University of Aveiro. It is a platform to manage the vehicle sharing system of an entity like hotels, city hall, etc.

1.3 Structure of the Thesis

This dissertation is organized into six chapters, each addressing a specific stage of the research and development process:

- Chapter 1 – **Introduction**: Presents the motivation for the work, the main objectives, and the challenges that the proposed solution aims to address.
- Chapter 2 – **Literature Review**: Discusses the theoretical foundations of the research, surveys existing solutions, and examines real-world scenarios relevant to maintenance management.
- Chapter 3 – **Requirements Analysis**: Defines the functional and non-functional requirements of the system, supported by detailed use cases that guide the design and implementation.

- Chapter 4 – **System Design and Implementation**: Describes the developed solution, including the database structure, the application's layout, and the main features implemented for each user role.
- Chapter 5 – **Validation and Results**: Presents the evaluation process, including user testing with external users followed by an analysis of the results.
- Chapter 6 – **Conclusion and Future Work**: Summarizes the main findings and identified limitations, and outlines possible directions for future development.

This structure ensures a logical flow from the problem definition to the validation of the solution, concluding with reflections and prospects for further research.

CAPÍTULO 2

Literature Review

The purpose of this chapter is to establish the theoretical and practical foundation for the development of the proposed solution. It begins by examining the vehicle maintenance process, highlighting its relevance to service quality and customer satisfaction. To support this analysis, academic and industrial publications were consulted using research platforms such as Scopus and Google Scholar. From the initial set of results, only a small number of studies were retained, as many focused on unrelated dealership models or narrow applications of machine learning for predictive maintenance. The selected works provide a clearer understanding of the challenges in vehicle maintenance services, the role of software solutions in this sector, and the importance of ensuring high-quality service delivery.

In addition to academic sources, this chapter incorporates real-world insights gathered from EMEL, the company responsible for operating Lisbon's Gira bike-sharing system. Their operational practices illustrate the constraints of current maintenance workflows and provide valuable context for aligning the proposed system with practical needs.

The chapter also introduces the concept of Computerized Maintenance Management Systems (CMMS) as a technological approach to organizing and optimizing maintenance activities. Furthermore, the SERVQUAL model is presented as a method to evaluate service quality in the after-sales industry, emphasizing the gap between customer expectations and perceived service delivery.

Finally, the chapter reviews existing software solutions in the market, such as Fiix and the system developed for MAS Motors LLC, analyzing their strengths, limitations, and relevance to this dissertation. This analysis helps identify the opportunities and requirements that guide the design of the proposed application.

To gather relevant information for this topic, I conducted a literature search using the Scopus and Google Scholar platforms. In Scopus, I applied the query "*Vehicle AND Maintenance AND Dealerships*", which initially returned 49 papers. After a preliminary review of titles and abstracts, 24 were identified as potentially relevant. However, a closer analysis

revealed that only three were directly applicable to this dissertation. Many of the discarded papers treated dealerships primarily as sales units of larger companies, which lies outside the scope of this work. Others focused on narrow applications of machine learning, such as predictive maintenance. While valuable in their own right, these approaches did not align with the dissertation's emphasis on web applications and maintenance process management.

A complementary search on Google Scholar yielded three additional studies related to the electric vehicle market in China. Although these works provided useful insights into international contexts, they were excluded as the focus of this dissertation is on maintenance operations rather than market-specific trends.

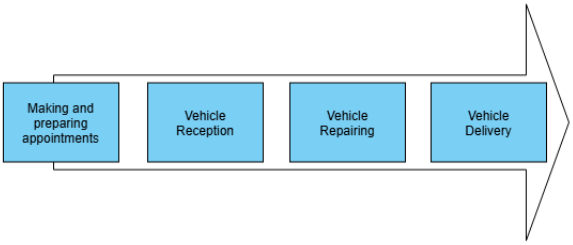
Ultimately, three papers were selected for detailed analysis. Two of them examined the vehicle maintenance process and highlighted the importance of service quality in ensuring customer satisfaction. The third presented a practical software solution with a use case comparable to the objectives of this dissertation. Together, these studies helped to define the research context and identify key problems and solutions to address during the development phase.

2.1 Theoretical Concepts

2.1.1 Vehicle Maintenance

Delivering vehicle maintenance services involves numerous activities, each crucial for maintaining service quality and ensuring customer satisfaction. Any deviation from these activities can negatively impact quality, leading to client dissatisfaction and loyalty loss. **Setting_the_after_sale_process** The loyalty of the client is the main source of income for the company, so it is important to maintain the quality of the service. **Setting_the_after_sale_process** To fulfill this requirement to the fullest, one must supervise every stage of the process. **Setting_the_after_sale_process**

Figura 2.1: Macro-level Flow of a vehicle maintenance or repair service. This figure was inspired by figure 6: After Sales process from **Setting_the_after_sale_process**.



The general flow of vehicle maintenance is illustrated in the figure **??**. The first step of the process starts with a client interacting with the garage to schedule a vehicle maintenance or repair. After that, the client goes to the garage, where the receptionist receives the client and the vehicle. Here, the mechanic will perform the maintenance of the vehicle and, when is done, the vehicle will be delivered to the client.

To ensure quality in the first step the receptionist must understand the fill capacity of the garage and the time to complete the job. With this information, the rececionist may accurately indicate the time to conclude the vehicle maintenance and get the client trust. **Setting_the_after_sale_process**

In the second step, the receptionist and service advisor, when receiving the vehicle, must do a visual confirmation of the vehicle's condition. **Setting_the_after_sale_process** In this step, the receptionist explains to the client the services of the garage and they both agree with the services to be performed until a determined date. **Setting_the_after_sale_process**

Following begins the third step and most important phase, the mechanic will perform the maintenance and repair of the vehicle. All of this process must be supervised by quality control to ensure that the job is done correctly. **Setting_the_after_sale_process** This includes the repair process, extra work, final tests, and service report. **Setting_the_after_sale_process** To accomplish that the use of a checklist is recommended to ensure precision and accuracy at each step. **Setting_the_after_sale_process**

Finally, the last step is the delivery of the vehicle to the client. Here, the workshop manager must review the work done and the final price of the service to avoid extra payments from the client and incomplete payments to the workers. **Setting_the_after_sale_process**

In this sense, the application of this dissertation will obey this flow. For the receptionist and mechanic, the application will focus on reducing their mistake by giving accurate and illustrated information and making the work more effective. And for the workshop manager to control the quality of the service with the assignment of tasks and authorization of purchases. Another user will be inserted into this flow, the warehouse operator, to manage the inventory of the dealership. The entire flow will be explained in Chapter III.

2.1.2 EMEL

In June, I visited the company **EMEL!** (**EMEL!**), an entity responsible for operating and managing the public bike-sharing system Gira in Lisbon. This system comprises approximately 2,000 bicycles and 188 stations, supporting between 12,000 and 14,000 trips per day. Given this high usage volume, equipment failures are frequent, making efficiency in the maintenance process a critical concern.

The maintenance workflow begins when a redistributor identifies a malfunction during routine operations and delivers the vehicle to the workshop. At this stage, the redistributor completes a paper form (Figures **??** and **??**), recording the detected faults and performing a simple diagnostic of electrical components such as the GPS unit and battery. If the vehicle fails these initial tests, it is set aside for a specialist to address before proceeding to standard maintenance. Otherwise, it enters a waiting list until a mechanic becomes available.

Once assigned, the mechanic repairs both the issues reported on the form and any additional problems identified during inspection. Upon completion, the mechanic records the parts used on the reverse side of the form. The vehicle is then forwarded to a quality control worker for validation. If defects are detected at this stage, the bicycle is returned to the same mechanic for correction, after which it waits to be redistributed back into circulation.

Due to the sheer volume of vehicles, EMEL prioritizes efficiency over thoroughness. As a result, inspections tend to be superficial, and hidden or secondary problems may go undetected.

Inventory management is handled through an Excel spreadsheet, where operators manually track stock quantities. Checks are performed periodically to update available parts, but this approach offers little visibility into consumption rates or predictive needs. EMEL works with multiple suppliers, each contracted to deliver parts in specific quantities by set deadlines. However, the absence of adequate tools to monitor contracts makes it difficult to determine when new tenders must be issued.

A digital system that streamlines maintenance records and provides real-time inventory management would significantly improve both efficiency and oversight, helping EMEL to better manage resources and maintain service quality.

2.1.3 Computerized Maintenance Management System

The software solution i will develop is a **CMMS!** (**CMMS!**). A **CMMS!** is able to centralize and automate the management of maintenance operations, helping the dealership to manage the work tasks and the inventory, providing metrics to optimize the maintenance process.

It maintains a database with the information about the equipment, maintenance schedules, work orders, inventory, and personnel. It also document and reports all maintenance actions to facilitate the adherence to the regulatory standards ensuring the quality of the service. And analyzes the maintenance costs, downtime, efficiency, and other metrics to improve performence, optimization and reduce cost to the dealership.

The key benefits of the **CMMS!** are the improvement of the Preventive Maintenance, since it's easier to analyse the vehicle life and schedule a maintenance before the equipment failures occur; reduce costs, as the life of the vehicle is prolonged and the critical maintenace being avoided, this sofware provides a long-term cost saving; efficiency enhancement, by allowing the workers to focus on completing task instead of paperwork; and Regulatory Compliance, by keeping track and documenting the maintenance activities to comply by the organization standards.

In conclusion, **CMMS!** provides a structured, data-driven approach to maintenance that enhances efficiency, reduces costs, ensures compliance, and ultimately supports the operational reliability of critical equipment and infrastructure.

2.1.4 Service Quality

Service quality is a critical factor in the success of vehicle maintenance services. One widely adopted framework for its evaluation is SERVQUAL, which provides a multidimensional approach for comparing consumers' perceptions of service quality against their expectations. The model emphasizes five dimensions **SERVQUAL_OLD**:

- Tangibles – Physical facilities, equipment, and appearance of personnel.
- Reliability – Ability to consistently deliver services as promised.
- Responsiveness – Willingness to assist the customer and proactivity.

- Assurance – Demonstrate courtesy and knowledge and inspire trust and confidence.
- Empathy – Caring and treating customers as individuals.

Figura 2.2: The 7-point Likert scale where the respondent may answer the question from strongly disagree to strongly agree. **master_servqual_model**

To assess these dimensions, SERVQUAL relies on a questionnaire consisting of 22 items, divided across the five categories. Each item requires two ratings: one for the expectation of the service and another for the perception of the service received. The responses are given on a 7-point Likert scale, ranging from strongly disagree to strongly agree, as illustrated in Figure **?? Measuring_After_sales_Service_Quality**. The quality gap is determined by the difference between the perception and expectation scores **servqual_blog_da_qualidade Measuring_After_sales_Service_Quality SERVQUAL_OLD**.

An illustrative application of the SERVQUAL model can be found in the study by **Measuring_After_sales_Service_Quality**, which assessed service quality at a CMV SA dealership in South Africa. The authors collected data through semi-structured questionnaires and interviews with customers, managers, and staff. The results, presented in Figure **??**, revealed negative scores across all five dimensions, with an overall average gap of -0.10. These findings highlight that the services provided by the dealership consistently fell short of customer expectations. Among the main suggestions from customers were the expansion of workshops to reduce travel inconvenience and improving the availability of parts, which were often delayed due to international supply chains. The study underscores the importance of continuous service quality monitoring and recommends that dealerships regularly apply the SERVQUAL framework to identify and address performance gaps.

Building on this insight, this dissertation proposes the development of a client-facing application that enables users to evaluate the quality of service received, provide feedback, and make recommendations. This approach follows the SERVQUAL model and aims to strengthen customer trust and satisfaction.

2.2 Existing solution

2.2.1 Fiix

Fiix is a cloud-based **CMMS!** that allow organizations to manage their maintenance operations. It's main strengths lies on its advanced preventive maintenance scheduling, Comprehensive Work Order and Asset Management and AI-Powered Analytics and Reporting.

Despite the advantages of this software, Fiix has a high learning curve due to it's complexity and not intuitive old-fashion design interface. Also Fiix repordly lacks a built-in interal

messaging system for crew communication. Most information sharing relies on notes and files attached to work orders, which may not be ideal for real-time collaboration.

With this effects in mind, i will aim in this software development to achieve a simple and user friendly solution where the users can intuitively interact with it by focusing on the key functional feature and easy navigation.

2.2.2 Service Management for MAS Motors LLC

MAS Motors LLC is a Toyota dealership in Libya that provides vehicle maintenance services. The company traditionally relied on manual processes, supported only by basic applications and paper-based documentation. While this approach was manageable at smaller scales, the expansion of the business exposed its inefficiencies and highlighted the need for a more modern solution **MAS_MOTORS**. To address this challenge, **MAS_MOTORS** developed a web application designed to streamline operations and enhance overall performance at the dealership.

The system was built to support multiple user groups—including service advisers, technicians, and customers—through a centralized platform that manages tasks such as job card handling, inventory updates, and customer service. The application was implemented using Laravel, a PHP web framework, together with MariaDB, an open-source relational database management system.

Evaluation results of the system were generally positive. A survey conducted with employees and customers assessed the solution using the **FURPS!** (**FURPS!**) model, yielding scores of 4.27 for functionality, 4.30 for usability, 4.27 for reliability, 4.46 for performance, and 3.36 for supportability. While supportability received the lowest score—mainly due to limited configuration options **MAS_MOTORS**—the system overall demonstrated clear improvements in service efficiency and customer satisfaction. Employees also provided valuable recommendations for future enhancements, including SMS-based service reminders, integration with social media platforms, and broader customer configuration options.

Building on these insights, my dissertation incorporates an email notification service, already used in the LightMobie bike-sharing system, to handle alerts, user validation, and service-related notifications. This addition addresses part of the communication gap identified by MAS Motors employees while leveraging existing infrastructure within the LightMobie ecosystem.

2.2.3 Architecture Comparison

The application will be developed using the ASP.NET Core MVC framework. This choice is motivated by several advantages over Laravel, the framework adopted in the solution proposed by **MAS_MOTORS**. A detailed comparison between the two frameworks is presented in Table **??**.

One of the key advantages of ASP.NET Core MVC is its seamless integration with Microsoft SQL databases through the Entity Framework package, which greatly simplifies database interactions. While Laravel also supports relational database integration, ASP.NET Core MVC is generally more suitable for Microsoft environments **asp_net_vs_laravel**.

Parameter	Laravel	ASP.NET Core MVC
Language	PHP	C#
Performance	Lower performance due to being an interpreted language	Higher performance due to being a compiled language
Security	Provides features such as hashing and input validation, but requires strong PHP knowledge and manual management	Offers advanced built-in security tools, including role-based authentication and Identity Framework support
Integration with SQL	Supports various relational databases	Recommended for Microsoft environments due to strong integration packages

Tabela 2.1: Comparison between Laravel and ASP.NET Core MVC frameworks for integration into the LightMobie platform.

In terms of performance, ASP.NET Core benefits from being based on C#, a compiled language, whereas Laravel is built on PHP, an interpreted language. This distinction gives ASP.NET Core a performance edge, especially in applications that require scalability and responsiveness.

Security is another critical factor. Although Laravel provides features such as hashing and secure input validation, it requires deeper PHP expertise and continuous monitoring for vulnerabilities. ASP.NET Core, on the other hand, integrates more advanced security mechanisms out of the box, including role-based authentication and login management through the Identity Framework. These abstractions reduce the burden on the developer, allowing greater focus on application functionality **asp_net_vs_laravel**.

Another strong motivation for adopting ASP.NET Core MVC is its compatibility with Lightmobie's Fleet Management System, which follows the same architectural approach. This ensures easier integration, particularly for functionalities such as authentication and role assignment, which are already implemented and can be reused. While this choice may introduce some additional complexity, my prior involvement in developing the Fleet Management System provides familiarity with its features, thereby reducing the learning curve and facilitating a smoother integration process.

2.3 Conclusion

In this chapter i explained the theoretical concepts of the vehicle maintenance process in the literature and in the real case example EMEL in lisbon, and The SERVQUAL method to evaluate the quality of the service of a organization. I also describe the existing solutions in the market, namely Fiix, and the results of MAS Motors LLC's application. This solutions lack a intuitive interface and a notification service, so in this dissertation i will focus on accomplish that.

CAPÍTULO 3

Requirements

This chapter defines the specification of the proposed system, combining insights from meetings with LightMobie, EMEL, and the literature. It begins with the requirements, organized by the FURPS model and extended with functional and system constraints. Then, the use cases and workflow are presented to illustrate how each role interacts with the system and how information flows during maintenance. Finally, a set of statuses is introduced to track the progress of each service request, ensuring transparency and consistency throughout the process.

3.1 Application Requirements

The requirements for the application were defined based on insights gathered from multiple sources. These include meetings with stakeholders from LightMobie and EMEL, as well as findings from the literature review. To structure and classify these requirements, the FURPS model was adopted, which organizes them into functional and non-functional categories.

3.1.1 Non-Functional Requirements

The non-functional requirements of the system were defined according to the FURPS model and are summarized in Table **??**. These requirements ensure that, beyond providing the necessary functionalities, the system also delivers high levels of quality, performance, and maintainability.

Category	Requirement
Scalability	The system must be capable of supporting users across Europe.
Reliability	The system must ensure at least 99% availability and be able to recover from failures within 4 hours.
Performance	The system's response time must not exceed 2 seconds.
Usability	Training a new user to effectively use the system should take no longer than 8 hours.
Supportability	The system must be compatible with the latest versions of Chrome, Firefox, Microsoft Edge, and Safari.

Tabela 3.1: Non-Functional Requirements (FURPS Model)

3.1.2 Functional Requirements

The functional requirements of the system define the operations and features necessary to support the daily activities of its users. The application is designed to accommodate multiple user roles, each with specific responsibilities: receptionist, mechanic, warehouse manager, and workshop manager. In addition, a dedicated client application is provided to support end users, while an administrator view ensures oversight and configuration of the system.

Administrator

- Create, edit, view, and remove task types
- Manage dealerships (create, edit, view, remove, add/remove vehicle types)
- Manage vehicle parts (create, edit, view, remove)
- Create and view dealership employees

Receptionist

- Switch interface language between English and Portuguese
- View daily working hours and user-specific hours
- Schedule vehicle reception (registration number, client email, owner name, reception date)
- Create maintenance (expected budget, tasks, expected conclusion date)
- View active maintenances (client name, registration number, entity, creation date, evaluation date, expected conclusion date, budget, planned work hours, tasks)
- Receive notifications on budget or deadline changes in maintenance
- Confirm or reject maintenance changes
- Cancel maintenance
- Conclude maintenance
- Notify when all maintenance tasks are completed

Mechanic

- Switch interface language between English and Portuguese
- View today's tasks
- Pause and continue tasks (e.g., for breaks)

- View maintenance task details (registration number, task type, parts, step descriptions, needed parts, client desired tasks)
- Finalize a task and comment before completion
- View evaluation tasks (maintenance tasks needed, client comment)
- If evaluations are not used, maintenance concludes when a mechanic finishes the evaluation task

Warehouse Manager

- Switch interface language between English and Portuguese
- View dealership inventory (name, code, quantity, warehouse location, description, category, price, quantity per group, min/max values for alarms/purchase requests)
- Edit part type details (location, min/max values, purchase request quantities)
- View quantity changes over time for each part
- See supplier details (name, phone, email, address, contract parts with start/end date)
- Create purchase requests (motive, parts, quantity per part)
- View all purchases (status, arrival date, total price, motive, creation date, parts, stock quantity, price, associated tasks)
- Register expected arrival date or delays for purchases
- Finalize delivery purchase by registering received parts

Workshop Manager

- Switch interface language between English and Portuguese
- View daily and user-specific working hours
- View unassigned tasks
- Assign tasks to mechanics
- View and manage active maintenances (full details including done/planned tasks)
- Add tasks to active maintenances
- View completed tasks with filters (client, vehicle, date), details (client name, vehicle registration, entity, creation date, evaluation date, conclusion date, budget, hours, price), and export as PDF
- View monthly maintenance revenue and total hours worked
- Assign purchases to operators
- View purchase requests (price, motive, creation date, parts info)
- Authorize or reject purchase requests
- Create supplier records (name, phone, email, address, contract parts/details)
- View suppliers and partnerships with bike sharing entities; accept/reject partnerships
- View employee list (email, name, DOB, phone, sex, role); create employee records (all listed data plus password)

Client

- View active maintenance information
- View maintenance history
- evaluate the perception of the service quality
- evaluate the expected service quality

3.1.3 System Requirements

The system also enforces several logical and structural requirements, detailed as follows:

- The system must generate a repair report for each maintenance completed.
- Each maintenance task must have a defined sequence to ensure correct order of execution.
- Each purchase record may include multiple parts, along with their respective details.
- The system must generate a purchase request when an available quantity of a part is under a determined threshold.
- Dealerships can only create maintenance tasks for vehicles belonging to their bike-sharing partners; otherwise, they may only accept maintenance from independent clients.
- The workflow must be flexible enough to support both general vehicle maintenance processes and the specific procedures used by EMEL.

3.2 Application Use Cases

To capture how different users interact with the system, a set of use cases was defined. The following subsections present the use cases for each user type, outlining their primary responsibilities and the actions they can perform within the application.

3.2.1 Receptionist Use Cases

The receptionist is responsible for direct communication with clients, handling vehicle check-in and check-out, as well as managing updates and changes to maintenance requests.

- **Use Case 1.1 – Maintenance Schedule**
  Scenario: Client arrives with a vehicle for repair.
  Objective: Create a new maintenance request.
  System: Receptionist enters vehicle and client details, evaluation date, entity, client notes, and requested tasks.
- **Use Case 1.2 – Maintenance Approval**
  Scenario: Vehicle evaluation concludes.
  Objective: Set budget, completion date, and tasks.
  System: Receptionist receives evaluation notification, consults customer, system calculates price, completion date set.
- **Use Case 1.3 – Collect Maintenance Information**
  Scenario: Client contacts dealership for maintenance status.
  Objective: View details of a maintenance request.
  System: Receptionist searches by vehicle or client and views relevant details.
- **Use Case 1.4 – Accept Maintenance Changes**
  Scenario: Issue arises and client accepts proposed changes.
  Objective: Accept changes to maintenance.
  System: Receptionist navigates to maintenance changes section on maintenance details and approves modifications.

- **Use Case 1.5 – Refuse Maintenance Changes**
  Scenario: Issue arises and client refuses the change.
  Objective: Refuse changes to maintenance.
  System: Receptionist navigates to maintenance changes section on maintenance details and rejects the modification.
- **Use Case 1.6 – Vehicle Delivery**
  Scenario: Vehicle is ready to be delivered to the client.
  Objective: Complete maintenance process.
  System: PDF report sent to client, maintenance marked as concluded and the delivery date is registered.
- **Use Case 1.7 – Cancel Maintenance**
  Scenario: Client rejects agreement and retrieves vehicle.
  Objective: Cancel the maintenance.
  System: Receptionist cancels maintenance within details section and invalidates all uncompleted tasks.
- **Use Case 1.8 – Cancel Task**
  Scenario: Issue leads to cancellation of a specific task.
  Objective: Cancel problematic task.
  System: Receptionist cancels task in maintenance changes section.

3.2.2 Mechanic Use Cases

Mechanics handle maintenance tasks such as vehicle repairs, part replacements, and diagnostics. The following use cases describe their core operations:

- **Use Case 2.1 – View To-Do List**
  Scenario: Mechanic begins shift.
  Objective: View tasks for completion.
  System: System presents assigned and available tasks with details.
- **Use Case 2.2 – Conduct Vehicle Analysis**
  Scenario: Vehicle needs analysis.
  Objective: Confirm initial assessment and detect additional issues.
  System: Mechanic reviews and selects analysis tasks.
- **Use Case 2.3 – Complete maintenance**
  Scenario: Maintenance tasks completed.
  Objective: Mark tasks as finished.
  System: Mechanic logs finished tasks for vehicle.
- **Use Case 2.4 – Execute Maintenance Task**
  Scenario: Vehicle scheduled for specific maintenance.
  Objective: Complete task following required steps.
  System: System presents a stepwise guidance.
- **Use Case 2.5 – Change Task**
  Scenario: Task has incorrect part associated.

Objective: Rectify the incorrect task.
System: Mechanic submits a task change request for client approval if the budget is impacted.

- **Use Case 2.6 – Continue Task**
  Scenario: Task was paused.
  Objective: Resume incomplete task.
  System: Mechanic selects task to continue from the list.

3.2.3 Warehouse Operator Use Cases

Warehouse operators manage inventory, handle part orders, and update records. Their use cases include:

- **Use Case 3.1 – View Stock**
  Scenario: Need to check part quantities.
  Objective: Show inventory levels.
  System: List all parts with current quantities.
- **Use Case 3.2 – Request Purchase**
  Scenario: Low stock identified.
  Objective: Request authorization to purchase.
  System: Place purchase order, notify administrator for approval.
- **Use Case 3.3 – Register Purchase**
  Scenario: Purchase request approved.
  Objective: Order new parts from the supplier.
  System: Operator enters the expected arrival date of the parts.
- **Use Case 3.4 – Register New Parts**
  Scenario: Parts arrive from supplier.
  Objective: Add new stock to inventory.
  System: Operator updates parts quantity and purchase completion date.
- **Use Case 3.5 – Edit Inventory**
  Scenario: Need to alter inventory details.
  Objective: Update inventory information.
  System: Changes made to part location, code or purchase generation.
- **Use Case 3.6 – Create Purchase Delay**
  Scenario: Purchase is delayed.
  Objective: Update expected arrival.
  System: Operator enters revised arrival date.

3.2.4 Workshop Manager Use Cases

Workshop managers oversee employee assignments, authorize purchases, track history, compile statistics, and manage partnerships.

- **Use Case 4.1 – Assign Tasks**
  Scenario: New maintenance requested.

Objective: Distribute tasks among staff.
System: Manager assigns tasks to employees.

- **Use Case 4.2 – Authorize Purchase**
  Scenario: Purchase request received.
  Objective: Accept/reject authorization.
  System: Manager approves or rejects the request.
- **Use Case 4.3 – View Maintenance History**
  Scenario: Access history of performed maintenance.
  Objective: Review details.
  System: Historical maintenance and detail review.
- **Use Case 4.4 – Compile Statistics**
  Scenario: Manager wants maintenance statistics.
  Objective: Present graphical summaries.
  System: System generates graphs of parts replaced, purchase volumes, costs, and ratings.
- **Use Case 4.5 – Assign Roles**
  Scenario: New employee hired.
  Objective: Set role and permissions.
  System: Assignment in employee record.
- **Use Case 4.6 – Add New Task**
  Scenario: Ongoing maintenance lacks a task.
  Objective: Add required task after client validation.
  System: Manager adds task to maintenance, client validates addition.
- **Use Case 4.7 – Create Employee**
  Scenario: Hiring new workshop staff.
  Objective: Register new employee.
  System: Manager completes employee registration in system.
- **Use Case 4.8 – Partnership Decision**
  Scenario: Entity proposes partnership.
  Objective: Accept or reject partnership.
  System: Status of partnership request updated by manager.

3.2.5 Client Use Cases

Clients interact through a dedicated application to monitor status, receive notifications, and provide ratings.

- **Use Case 5.1 – View Maintenance Status**
  Scenario: Client checks current status.
  Objective: Display maintenance progress.
  System: System displays completed and pending steps.
- **Use Case 5.2 – Receive End Notification**
  Scenario: Maintenance completed.
  Objective: Notify client.
  System: email sent.

- **Use Case 5.3 – Rate Completed Service**
  Scenario: Maintenance concludes and client retrieves vehicle.
  Objective: Obtain service feedback.
  System: Rating form offered in app.
- **Use Case 5.4 – Rate Expected Service**
  Scenario: Maintenance agreement finalized.
  Objective: Gather rating on expected service.
  System: Rating form presented to client.
- **Use Case 5.5 – View Maintenance History**
  Scenario: Client reviews previous maintenances.
  Objective: Show maintenance history.
  System: History shown and accessible via app interface.

3.3 Applications Workflow

To provide a clear view of how users interact both with the system and with each other, a workflow diagram was created based on the defined use cases. The overall application workflow is presented in Figure **??**.

The main workflow begins when a client arrives at the dealership to request a vehicle maintenance. The receptionist records the request and registers this information in the system (Use Case 1.1). Once the request is created, a mechanic evaluates the vehicle, identifies problems, and specifies the necessary replacement parts (Use Case 2.2).

After this evaluation, the receptionist contacts the client to present the results. At this stage, both parties agree on the final maintenance plan, including the tasks to be performed, the expected delivery date, and the associated costs (Use Case 1.2). The client also provides an initial evaluation of the expected service quality on the client app (Use Case 5.4). When the plan is confirmed, the workshop manager receives a notification of the tasks and assigns them to the appropriate mechanics (Use Case 4.1).

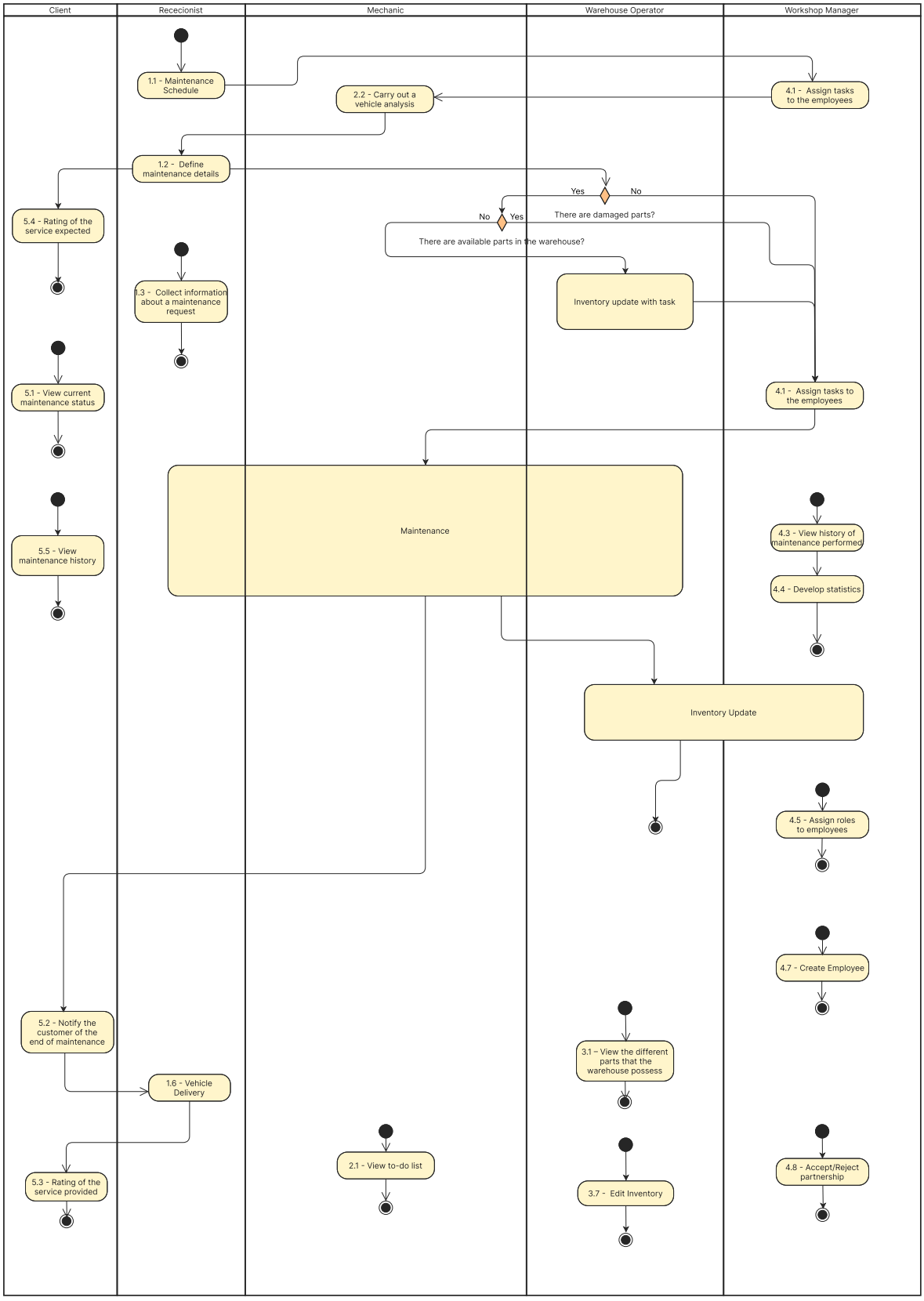
From this point onward, the execution of the tasks may or may not require replacement parts. If no parts are needed, the mechanic performs the assigned maintenance (Use Case 2.6). If parts are required, the warehouse operator verifies stock availability. When the parts are available, they are delivered to the mechanic (Use Case 2.4). If they are not available, the operator submits a purchase request that must be approved by the workshop manager. If approved, the operator proceeds to order the parts (Use Case 3.3), register them in the system (Use Case 3.4), and deliver them to the mechanic. If the purchase request is rejected, the operator must restart the process with a new request (Use Case 3.2).

During the course of the maintenance, unexpected events may occur. A delivery delay (Use Case 3.6) or a task modification identified by the mechanic (Use Case 2.5) may affect the budget or the expected completion time. In such cases, the receptionist informs the client and requests a decision. The client may authorize the changes and allow the work to continue (Use Case 1.4), reject the changes while maintaining the original agreement (Use Case 1.5), refuse the changes but authorize the completion of the remaining approved tasks (Use Case 1.8), or

cancel the maintenance entirely (Use Case 1.7). If the last option is chosen, all remaining tasks are invalidated, and the vehicle is prepared for check-out (Use Case 5.3).

When the maintenance tasks are completed, the system notifies the client that the vehicle is ready. The receptionist delivers the vehicle to the client (Use Case 1.6), after which the client is prompted by the application to rate the service (Use Cases 5.2 and 5.3).

Figura 3.1: Use case flow chart of the Client, Receptionist, Mechanic, Warehouse Operator, and Workshop Manager.



3.4 System Statuses

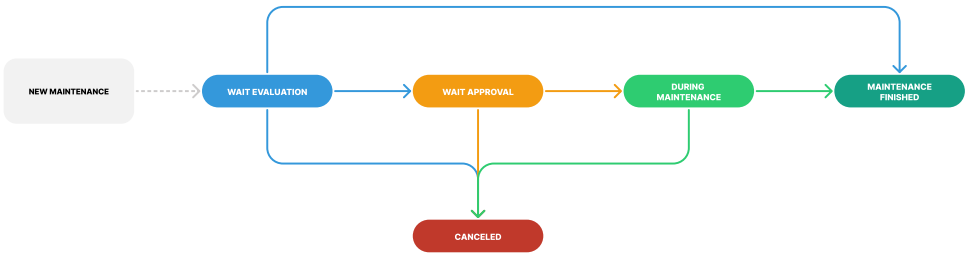
3.4.1 Maintenance status

In order to ensure transparency and consistency during the maintenance process, it is important to track the overall progress of each service request through a set of predefined statuses. These statuses provide a structured view of the workflow, from the initial vehicle evaluation to its final delivery, while also capturing critical decision points that may alter the course of the process.

These are the *maintenance status* key stages of this workflow:

- **Wait Evaluation**: Inicial status when the vehicle is waiting to be inspected by the mechanic.
- **Wait Approval**: The mechanic has proposed tasks that require the client's approval.
- **During Maintenance**: Approved tasks are actively being executed by the workshop.
- **Maintenance Finished**: All tasks have either been completed or invalidated.
- **Delivered**: The vehicle is returned to the client after maintenance conclusion.
- **Canceled**: The client cancels the maintenance process mid-workflow.

Figura 3.2: Status flow chart of a maintenance workflow



The status transitions represented in **??** outline how the workflow evolves over time:

- **Wait Evaluation** → **Wait Approval**: After the mechanic completes the initial evaluation.
- **Wait Approval** → **During Maintenance**: Once the client approves the proposed tasks and budget.
- **During Maintenance** → **Maintenance Finished**: When all approved tasks are completed.
- **Maintenance Finished** → **Delivered**: The vehicle is handed back to the client.
- Any status → **Canceled**: If the client decides to cancel the maintenance process.
- **Wait Evaluation** → **Maintenance Finished**: If the dealership skips the evaluation step and the mechanic completes all tasks directly.

In the figure **??** and **??** we can see how these status are divided in the use case flow of the application.

To further clarify how these statuses operate in practice, **??** and **??** illustrate their use within two different contexts: a dealership providing standard maintenance services and an entity managing maintenance without evaluation.

Figura 3.3: Use case flow chart of dealership maintenance, with statuses grouped by use cases

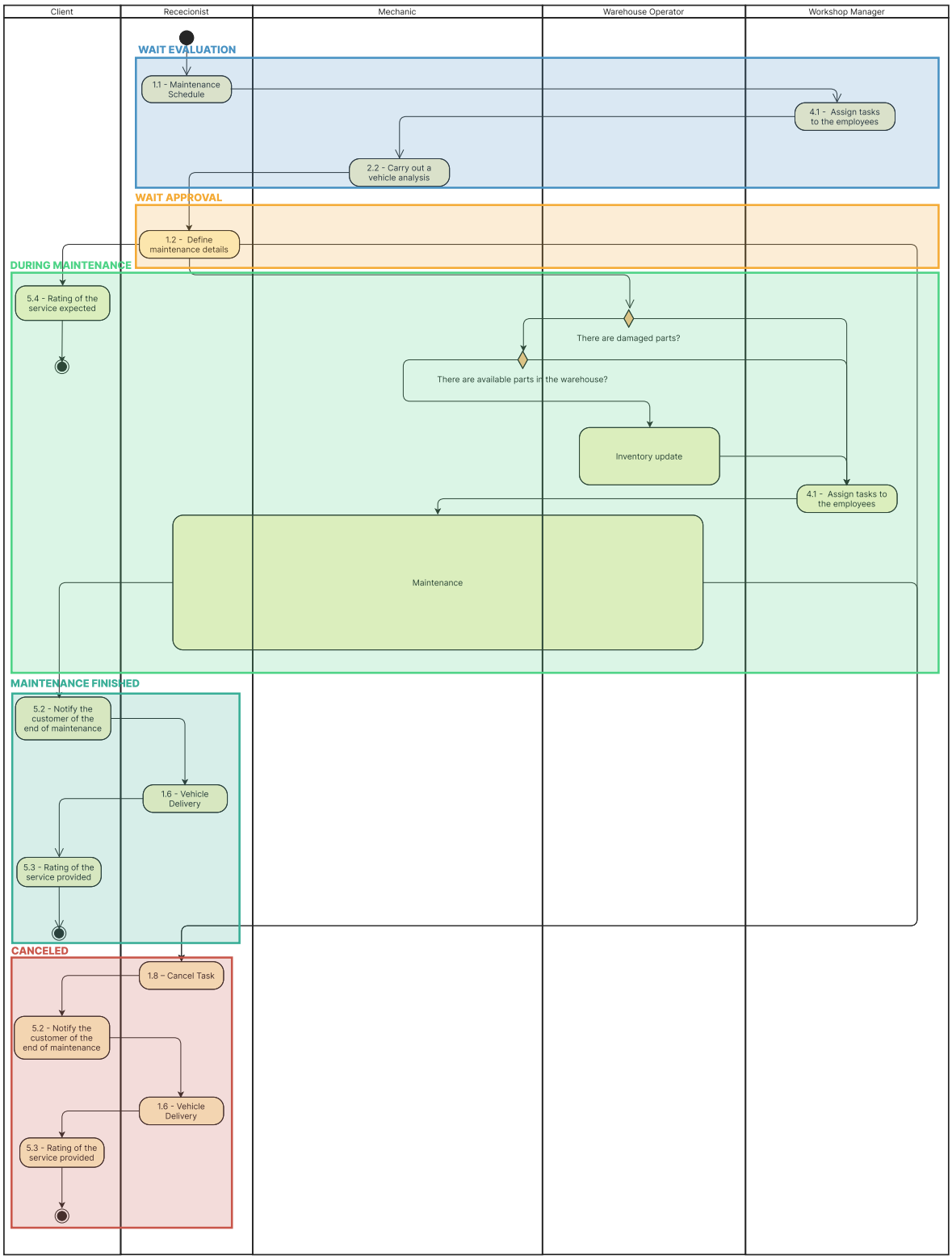
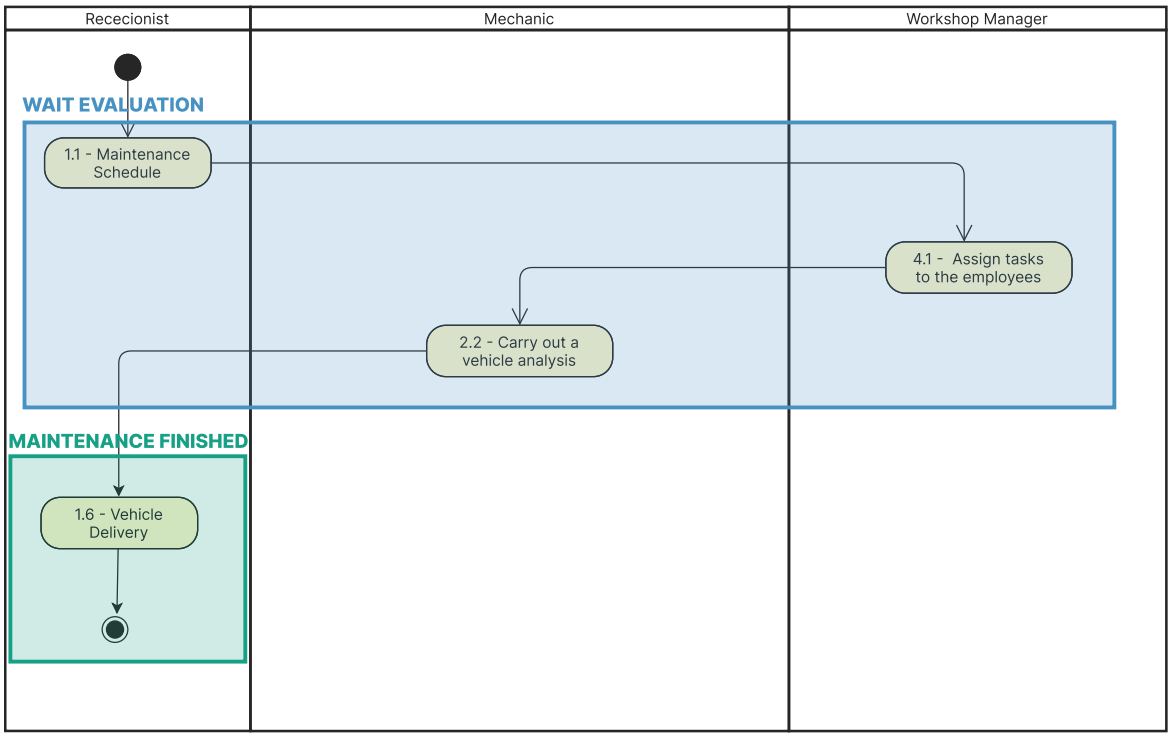


Figura 3.4: Use case flow chart of an entity managing maintenance without evaluation, with statuses grouped by use cases

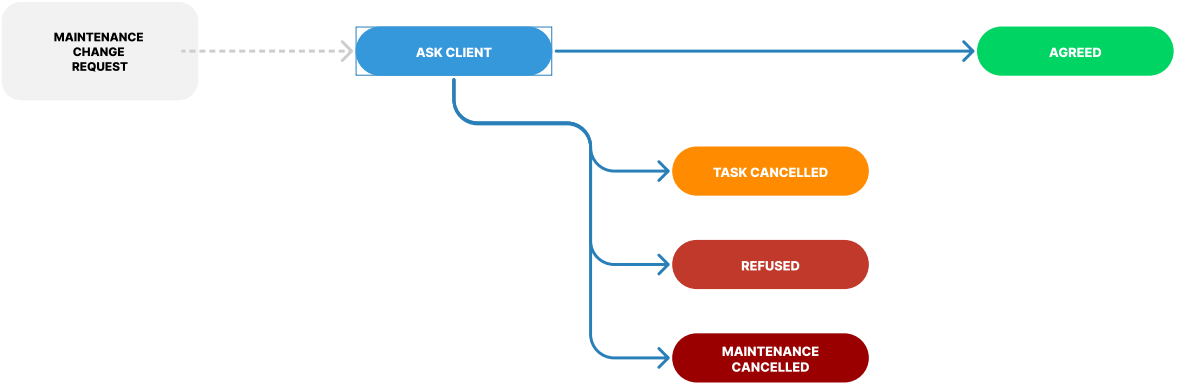


3.4.2 Maintenance Change Status

During the course of maintenance, situations may arise that require modifications to the originally planned tasks or agreements. To manage this process effectively, the workflow is modeled through a set of statuses that capture the client's decision and define the subsequent path of action.

- **Ask Client**: Initial status when a maintenance change is identified and requires confirmation from the client.
- **Agreed**: The client accepts the proposed changes.
- **Refused**: The client rejects the changes, and the maintenance continues according to the previous agreement.
- **Task Canceled**: The client cancels the specific task that triggered the maintenance change.
- **Maintenance Canceled**: The client terminates the entire maintenance process.

Figura 3.5: Status flow chart of a maintenance change



The transitions between these statuses, as shown in **??**, describe how the process evolves depending on the client's decision.

3.4.3 Maintenance Task Status

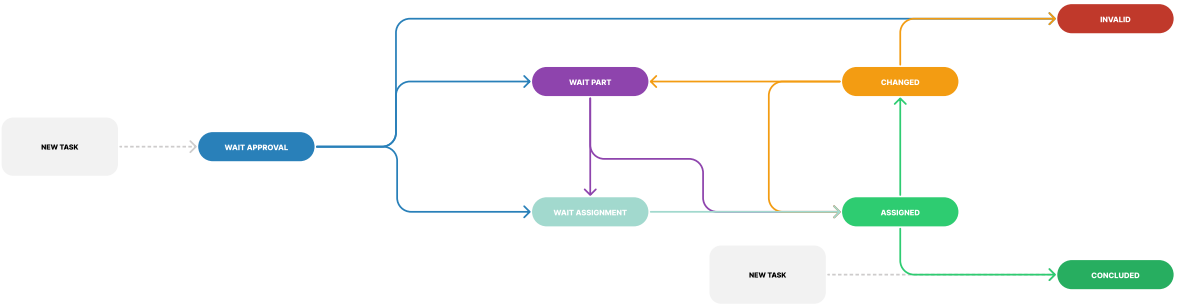
Each maintenance process is composed of individual tasks, which represent the specific operations that mechanics must perform. To ensure proper tracking and accountability, every task follows a defined workflow of statuses that describe its current stage, from creation and approval to execution and completion.

The *maintenance task status* includes the following stages:

- **Wait Approval**: Initial status when the task is created and requires client validation.
- **Wait Assignment**: The client has approved the task, no additional parts are needed (or they are available in stock), but the task has not yet been assigned to a mechanic.
- **Invalid**: The client rejects the task, and it is removed from the workflow.
- **Assigned**: The task is assigned to a mechanic and is ready to be executed.
- **Concluded**: The mechanic completes the task successfully.

- **Changed**: The mechanic identifies an issue during execution and suggests a modification that requires client approval.
- **Wait Part**: The task is approved by the client but cannot be completed until the necessary parts, not currently in inventory, become available.

Figura 3.6: Status flow chart of a maintenance task

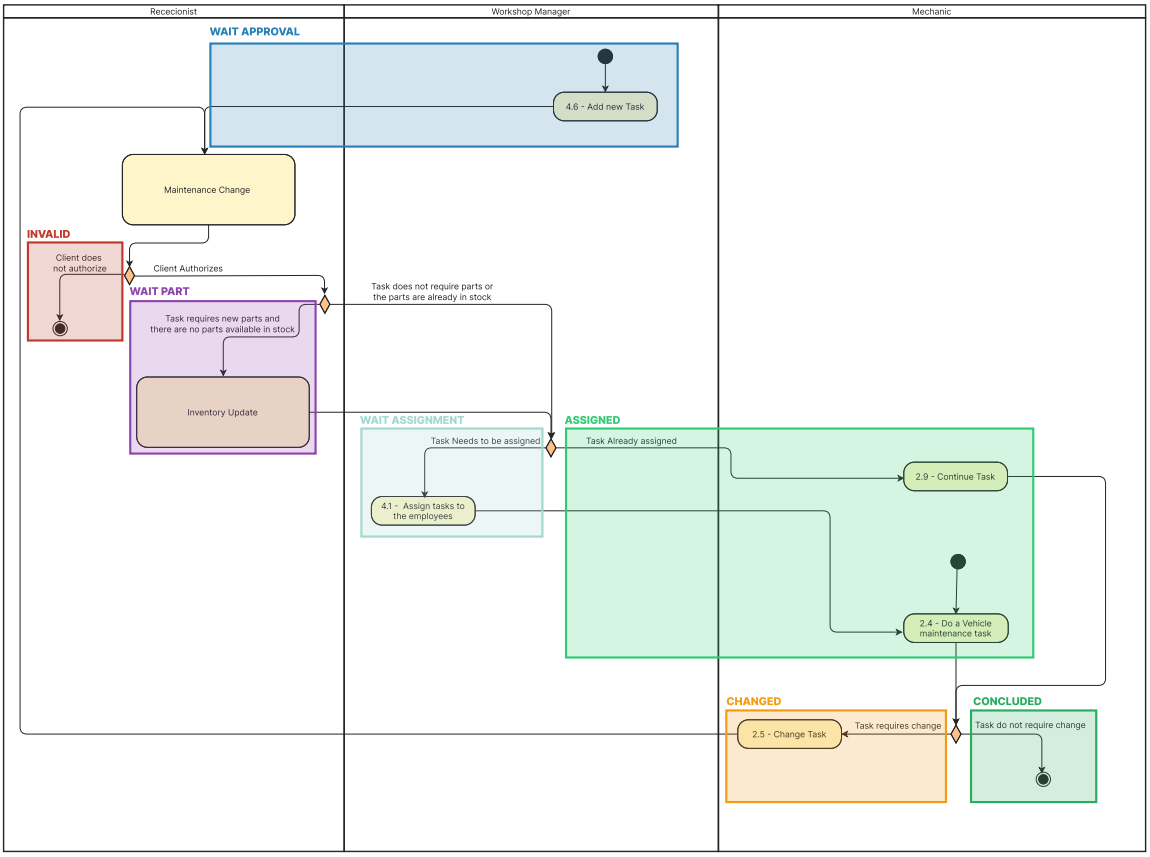


The task workflow shown in **??** highlights the possible transitions between these statuses:

- **Wait Approval** → **Invalid**: If the client rejects the task.
- **Wait Approval** → **Wait Assignment**: If the client approves and no external parts are required.
- **Wait Approval** → **Wait Part**: If the client approves but required parts are not available in stock.
- **Wait Assignment** → **Assigned**: Once a mechanic is assigned to the task.
- **Assigned** → **Concluded**: When the mechanic completes the task.
- **Assigned** → **Changed**: If the mechanic identifies an issue and suggests a change, requiring renewed client approval.
- **Changed** → **Assigned**: After the suggested change is approved by the client.
- **Changed** → **Wait Part**: After the suggested change is approved by the client, but the required part is not available in stock.
- **Wait Part** → **Wait Assignment**: After the part arrive at the warehouse and the task is not yet assigned.
- **Wait Part** → **Assigned**: After the part arrive at the warehouse and the task is already assigned.

In practice, the entry point of this workflow depends on the maintenance context. If the dealership skips the evaluation, tasks may begin directly in the **Concluded** state. In contrast, at a dealership, tasks typically originate from the workshop manager or during the initial vehicle evaluation, as illustrated in **??**. In this case, the workflow starts in **Wait Approval**, followed by the client's decision and subsequent progression through the task statuses until completion. This structured approach ensures flexibility while maintaining a standardized flow across different maintenance scenarios.

Figura 3.7: Use case flow chart of a maintenance task, showing status transitions within the workflow



3.4.4 Purchase Status

In order to keep maintenance workflows uninterrupted, parts and materials often need to be acquired through a structured purchasing process. This process ensures that requests are validated, assigned, tracked, and properly registered once the items arrive. To guarantee accountability, the purchase workflow is modeled with a set of statuses that represent each step, from the initial request to its completion.

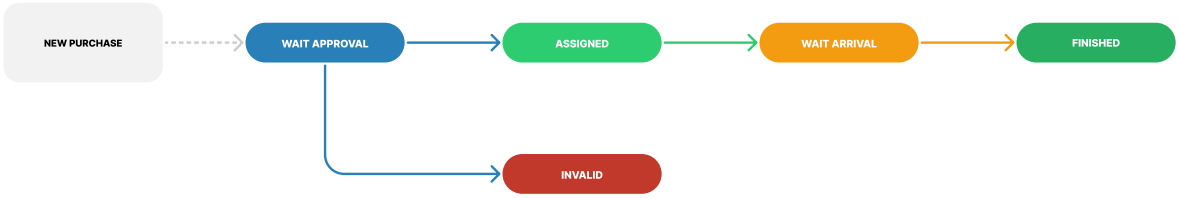
The *purchase status* is defined by the following stages:

- **Wait Approval**: Initial status when a purchase request is submitted and awaits validation by the workshop manager.
- **Invalid**: The request is rejected by the workshop manager and removed from the workflow.
- **Assigned**: The request is approved and assigned to a warehouse operator.
- **Wait Arrival**: The operator has contacted the supplier and is awaiting delivery of the requested parts.
- **Finished**: The parts have arrived at the dealership and are registered into the system.

The status transitions illustrated in **??** highlight the sequence of steps in a purchase workflow:

- **Wait Approval** → **Invalid**: If the workshop manager rejects the request.
- **Wait Approval** → **Assigned**: If the request is approved.

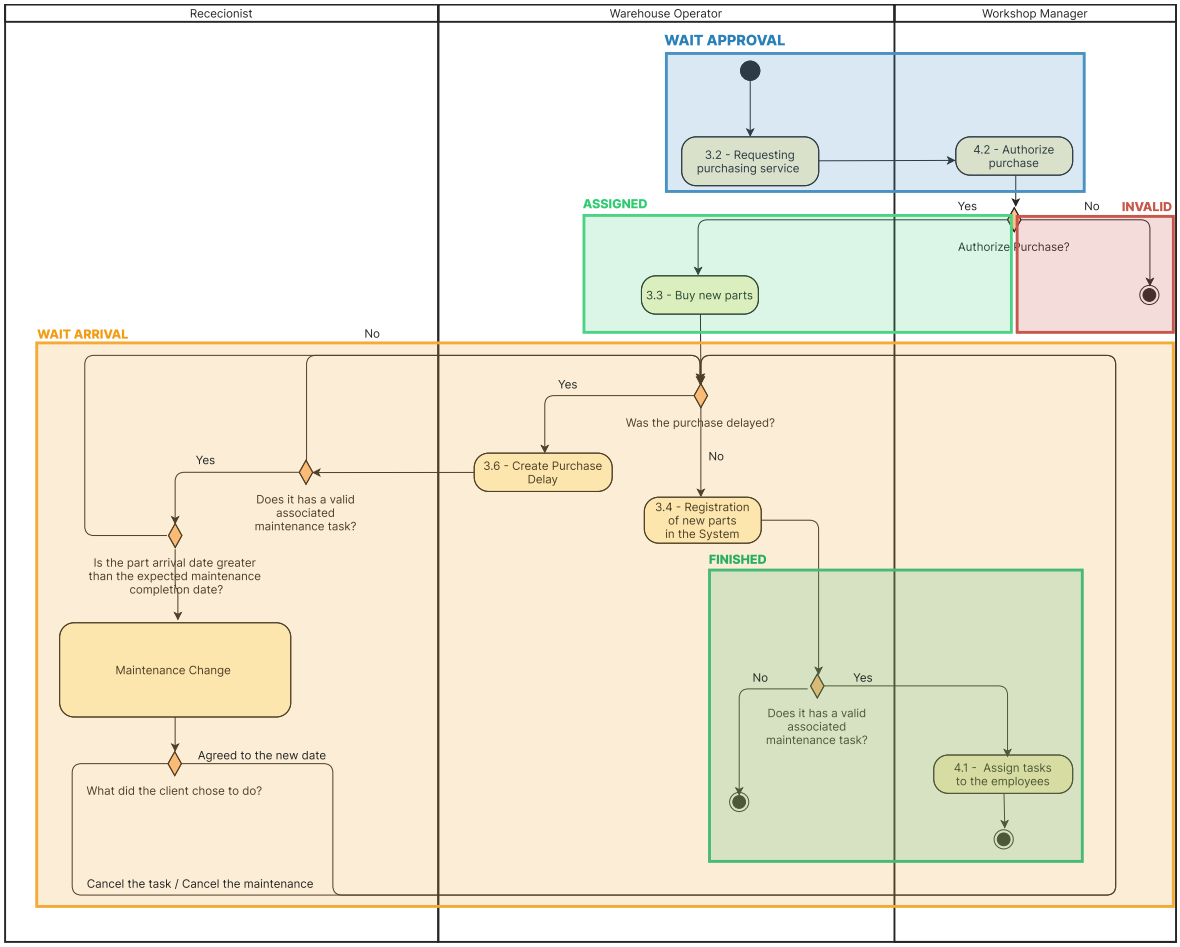
Figura 3.8: Status flow chart of a purchase



- **Assigned → Wait Arrival**: After the operator contacts the supplier and logs the expected delivery.
- **Wait Arrival → Finished**: Once the parts arrive and are recorded in the system.

As shown in **??**, the use case flow chart provides further detail by illustrating how purchase requests are initiated, validated, and completed. This structured process ensures that all acquisitions are traceable and properly integrated into the dealership's inventory.

Figura 3.9: Use case flow chart of a purchase, with statuses grouped by workflow steps



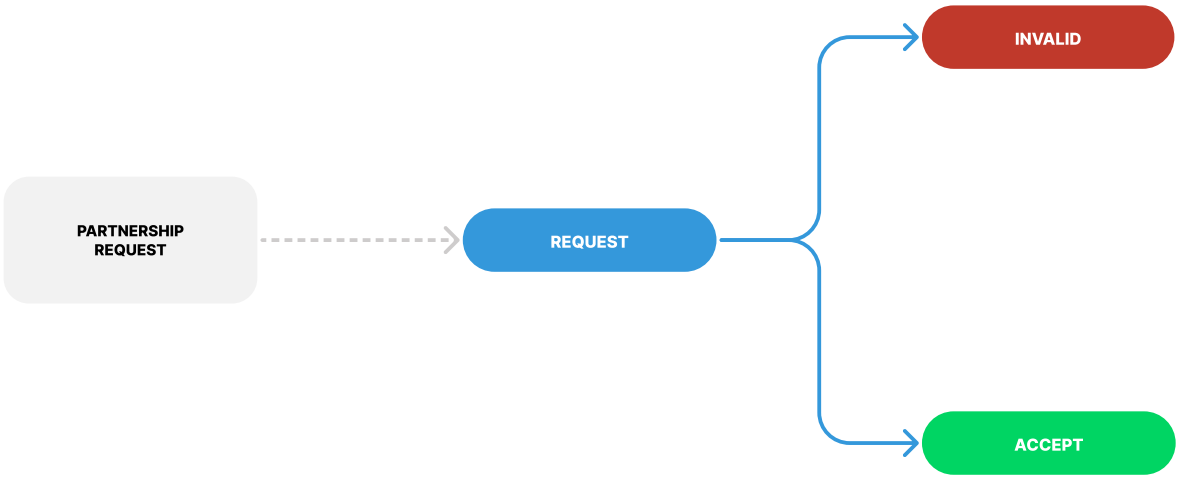
3.4.5 Owner Partnership Status

To extend maintenance services beyond individual clients, dealerships may establish partnerships with organizations that own and manage vehicle fleets. The process is modeled through a simple status flow that tracks partnership requests and decisions.

The *owner partnership status* consists of the following stages:

- **Request**: Initial state when a new entity submits a request to establish a partnership with the dealership.
- **Accept**: The dealership approves the request, gaining access to the entity's vehicle information and authorization to schedule maintenance.
- **Denied**: The dealership rejects the partnership request, ending the process.

Figura 3.10: Status flow chart of an owner partnership



This workflow, shown in **??**, ensures that all partnership requests are explicitly validated, preventing unauthorized access to sensitive vehicle data.

CAPÍTULO 4

Implementation

This chapter presents the design and implementation of the system. It begins with the database structure, organized into six core areas to support all workflows described in the previous chapter. Following this, the application views are detailed, illustrating how each user role — receptionist, mechanic, warehouse operator, workshop manager, client, and administrator — interacts with the system through tailored interfaces. The chapter concludes with a summary showing that the implemented views collectively fulfill the defined system requirements.

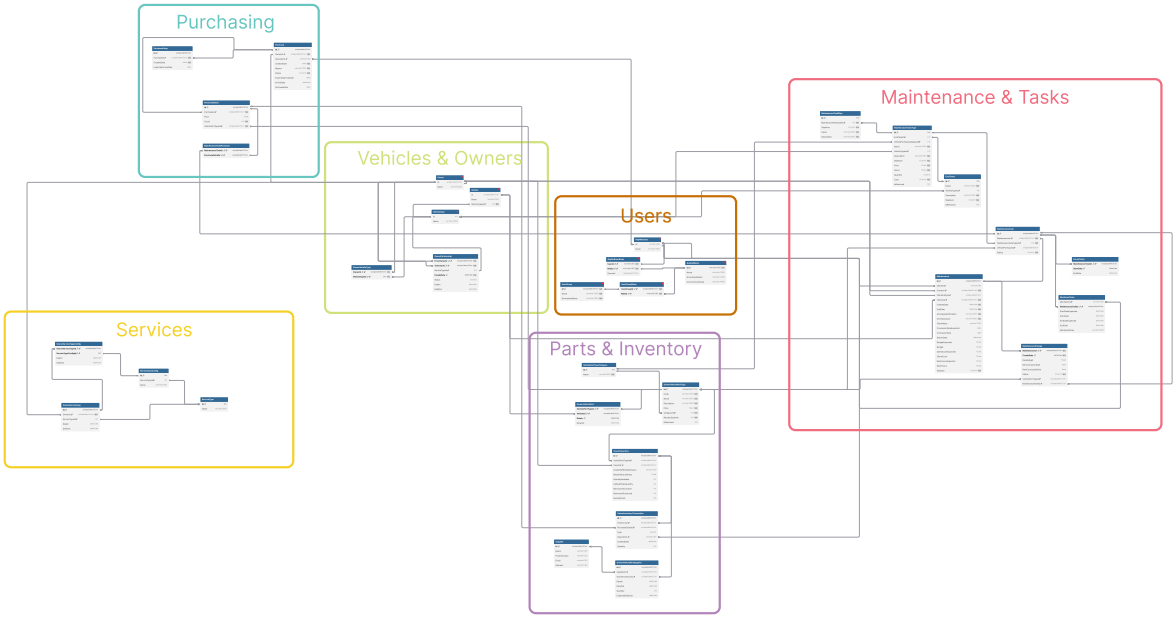
4.1 Database

The database was designed to support all the workflows described in the previous sections, ensuring that the system could track users, vehicles, maintenance tasks, parts, purchases, and dealership partnerships in a consistent way. The overall structure is shown in **??**, which organizes the database into six main areas:

- Users
- Maintenance and Tasks
- Parts and Inventory
- Purchasing
- Services
- Vehicles and Owners

Tables that already existed in the original Lightmobie database are marked with a red dot in the diagrams.

Figura 4.1: Database general diagram.

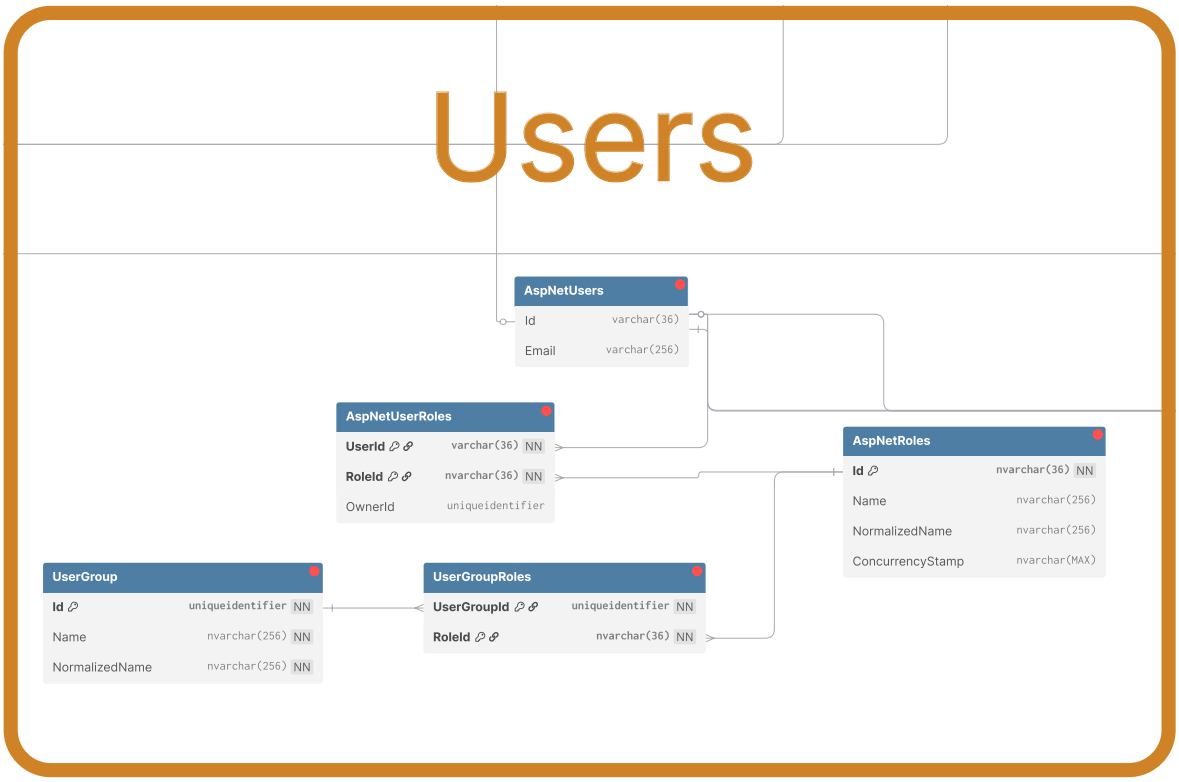


4.1.1 Users

The user section builds on the existing Lightmobie database. It manages user accounts, their roles (e.g., client, receptionist, mechanic, warehouse operator, administrator), and user groups. This structure defines the system's access control and ensures that each type of user can only perform their intended operations.

?? shows this structure.

Figura 4.2: Database diagram for Users.



4.1.2 Maintenance and Tasks

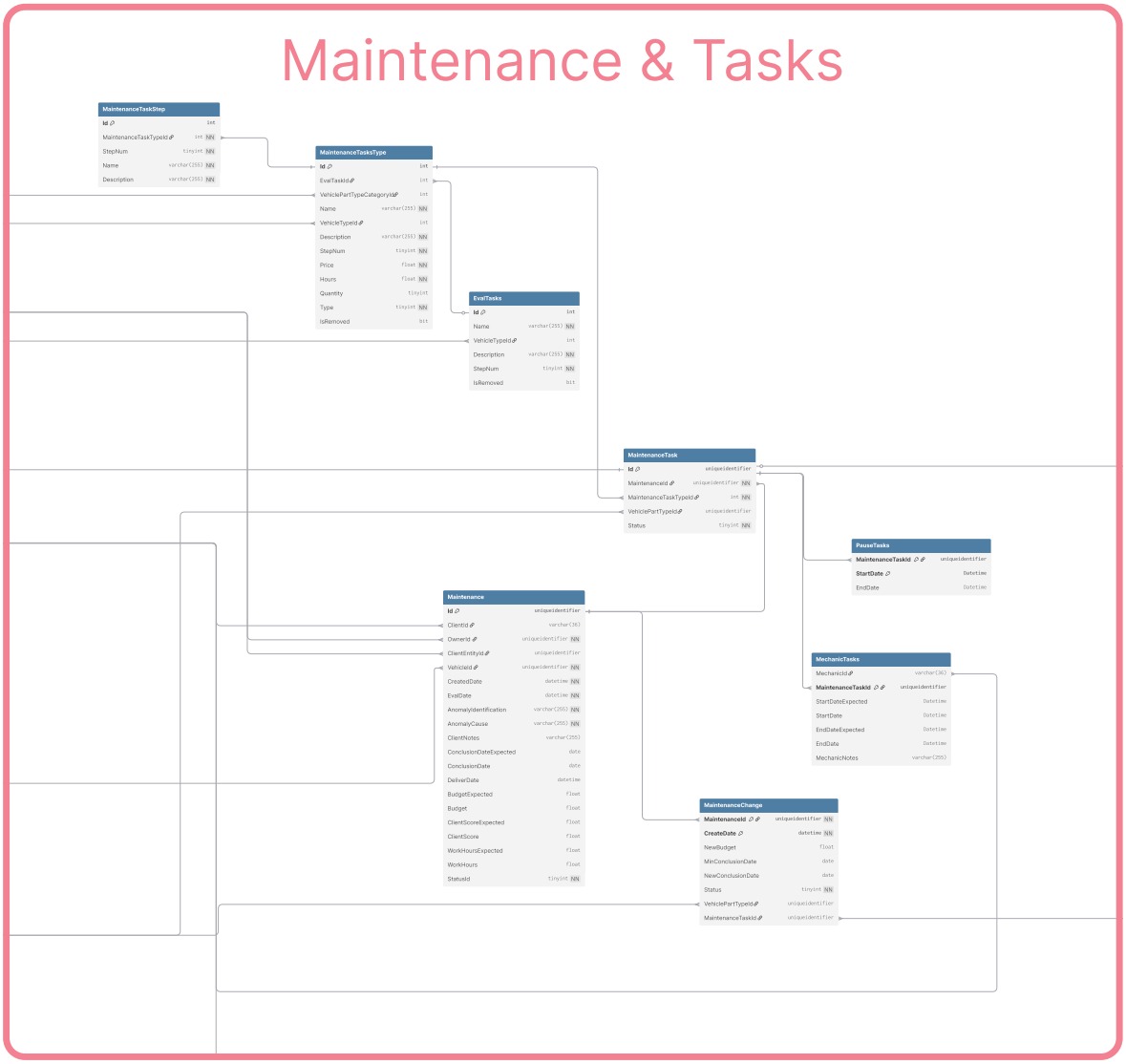
Maintenance is at the core of the system. Each maintenance record links a client, a dealership, and a vehicle, and stores important information such as evaluation results, expected and actual budgets, delivery dates, and quality scores.

Tasks are associated with a maintenance and describe the specific operations to be performed. Each task can have a status (e.g., waiting for client approval, assigned to a mechanic, waiting for parts, concluded), can be paused, and is eventually completed by a mechanic.

The system also supports maintenance changes (e.g., delays, budget adjustments, or task cancellations). These are stored separately so that all modifications can be tracked and require explicit validation from the client.

?? illustrates this structure.

Figura 4.3: Database diagram for Maintenance and Tasks.



4.1.3 Parts and Inventory

The inventory section keeps track of all parts available at a dealership. For each part, the system stores quantities, warehouse location, and stock thresholds to automate restocking when levels fall below a minimum.

Every change in stock (sales to mechanics, restocks from purchases) is logged as a transaction, which ensures traceability. The database also models suppliers and their contracts with dealerships, so the system knows which supplier provides which parts.

?? shows this structure.

Figura 4.4: Database diagram for Parts and Inventory.

4.1.4 Purchasing

When stock runs out, the system generates purchase requests. Each purchase goes through a workflow: waiting for approval by the workshop manager, assignment to a warehouse operator, ordering from a supplier, waiting for delivery, and finally being marked as finished once the parts arrive.

Purchase records store details of the items ordered, expected and actual arrival dates, and possible delays. Purchases can also be linked directly to maintenance tasks, ensuring that when parts arrive, the related tasks are updated automatically.

?? illustrates the purchasing tables.

Figura 4.5: Database diagram for Purchasing.

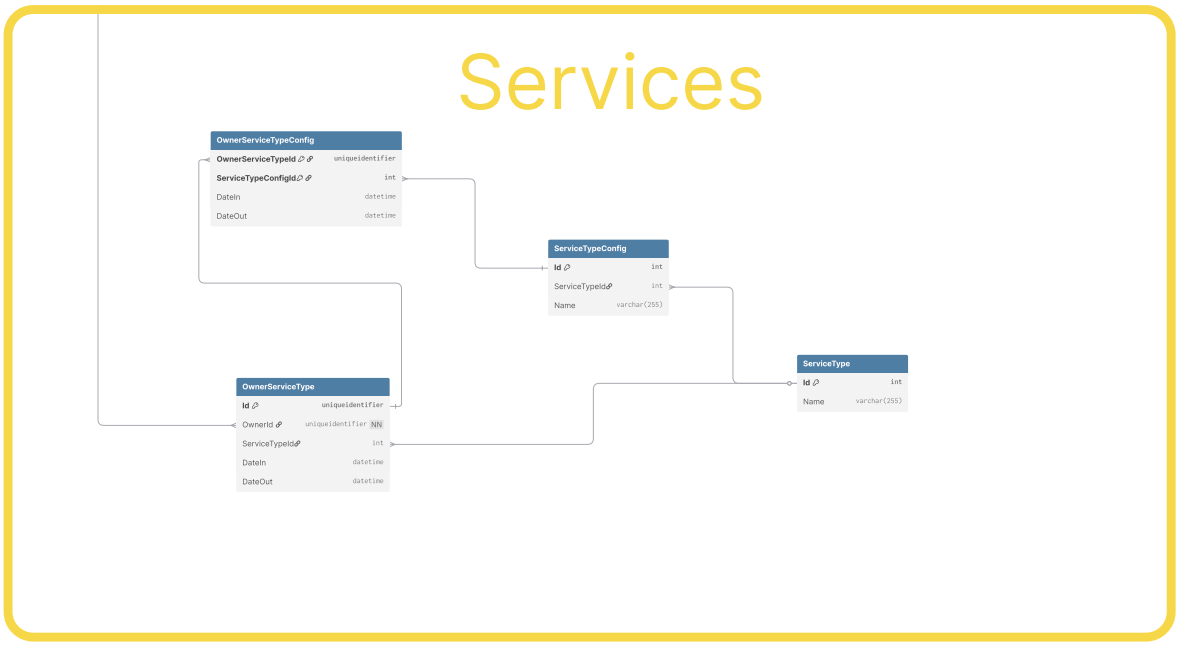
4.1.5 Services

The services section defines the types of services supported by the system, such as dealership maintenance or bike-sharing fleet management. Since not all dealerships operate in the same way, this part of the database allows them to configure how their workflow should behave.

By storing these configuration options, the system adapts to different service models while maintaining a consistent data structure.

?? shows the services structure.

Figura 4.6: Database diagram for Services.



4.1.6 Vehicles and Owners

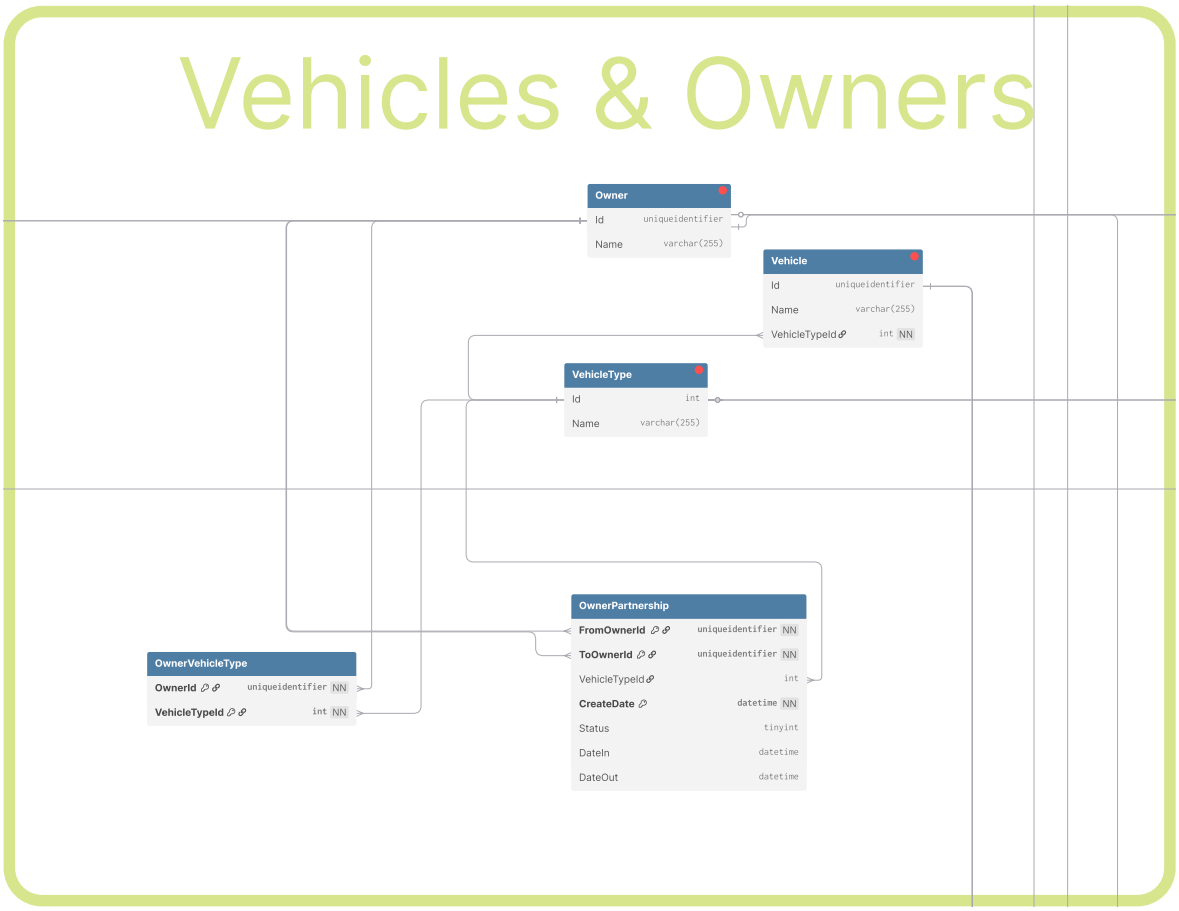
Finally, the system manages dealerships, bike-sharing companies, and vehicles. Vehicles are linked to their type (e.g., electric bike, conventional bike) and to the dealership or entity that owns them.

Partnerships between bike-sharing companies and dealerships are also modeled, allowing bike-sharing entities to grant specific dealerships access to their fleet information. Each partnership request can be accepted, denied, or remain pending.

?? shows this part of the database.

Together, these six areas form an integrated database that supports the entire maintenance workflow. User accounts and roles define who can perform which actions, while vehicles and owners provide the foundation for creating maintenance records. Each maintenance can then be broken into tasks, which may require parts from inventory or trigger purchase requests. Services configure how dealerships organize these operations, ensuring flexibility across different business models. By connecting all these elements, the database ensures

Figura 4.7: Database diagram for Vehicles and Owners



that every interaction—from a client request to the arrival of new parts—remains consistent, traceable, and adaptable to the needs of each dealership.

4.2.1 Rececionist view

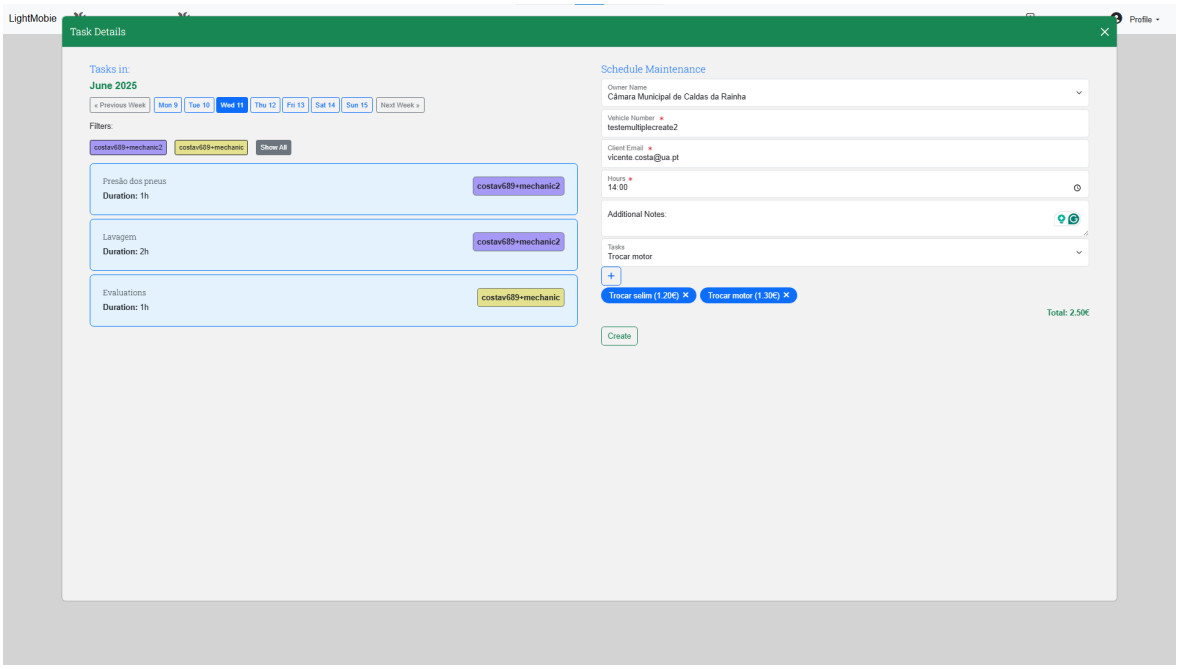
Figura 4.8: Rececionist home page.

The receptionist's interface is designed to simplify the scheduling and monitoring of maintenance operations. Its main screen (**??**) presents a bar chart that displays the expected working hours of each mechanic for the selected week. Navigation buttons allow the receptionist to move between weeks, making it easy to identify which mechanics are available to perform vehicle evaluations. This directly supports Use Case 1.1 – Maintenance Schedule.

To schedule a new maintenance, the receptionist can either click the "Adicionar Tarefa" button or select a specific date on the chart. Both options open a modal (**??**), where the date can be adjusted, existing tasks for that day are displayed, and a scheduling form is available. The form includes fields for the client's information, vehicle registration, arrival time, optional pre-selected tasks, and notes for the mechanic. Once completed, the receptionist confirms the schedule, creating a new maintenance record in the system.

The application also provides quick access to active maintenances through the "Manutenções Ativas" menu option (**??**). This section lists ongoing maintenances, including key details such as vehicle registration, associated entity, expected completion date, and progress percentage. Visual indicators highlight maintenances with pending changes requiring client confirmation. From here, the receptionist can open detailed views (**??**, **??** and **??**), which present information about the maintenance, associated tasks, and any proposed changes. These features enable the receptionist to respond accurately to client inquiries (Use Case 1.3 – Collect maintenance information) and to manage approvals or cancellations of changes (Use Cases 1.4, 1.5, and 1.7).

Figura 4.9: Rececionist schedule maintenance.



The final section of the receptionist view is the "Ações" menu (**??**, **??**), which lists maintenances that have completed evaluation and are awaiting client decisions. By accessing the details, the receptionist can review tasks proposed by the mechanic, their costs, required parts, and availability in the warehouse. The system dynamically updates the estimated budget and timeline as tasks are selected. Once the client's preferences are confirmed, the receptionist finalizes the maintenance plan (Use Case 1.2 – Define maintenance details).

Additional features include multilingual support (Portuguese and English) and account management options, allowing the receptionist to switch languages or log out, change email and change password via the profile menu.

Figura 4.10: List of active maintenances.

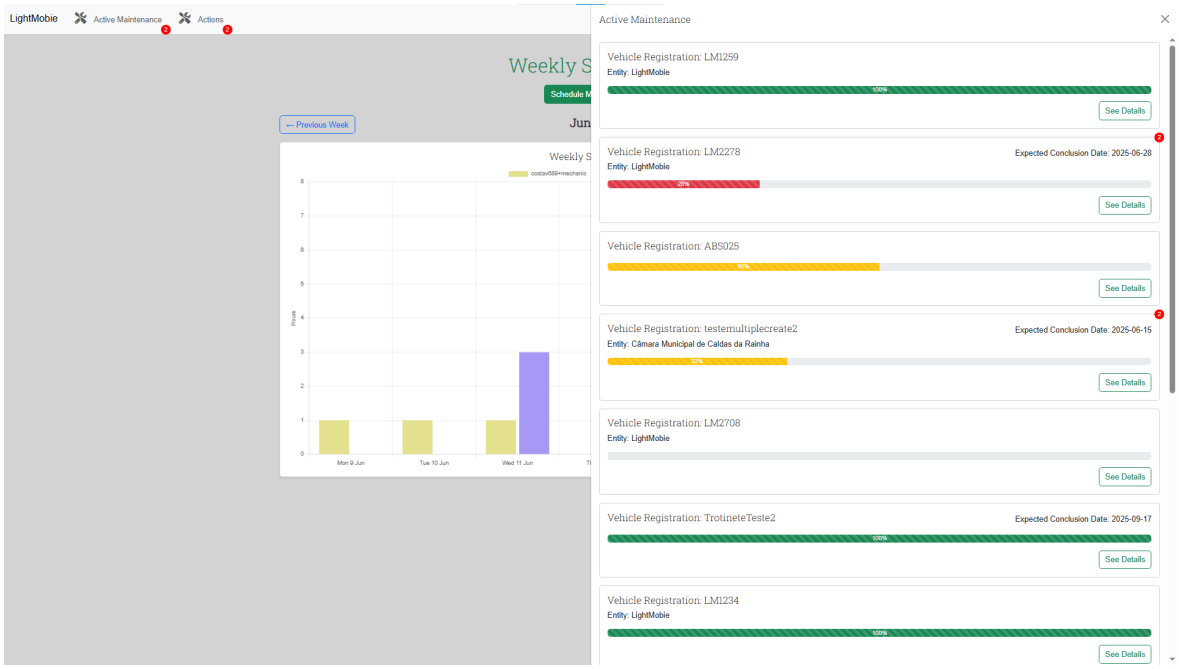
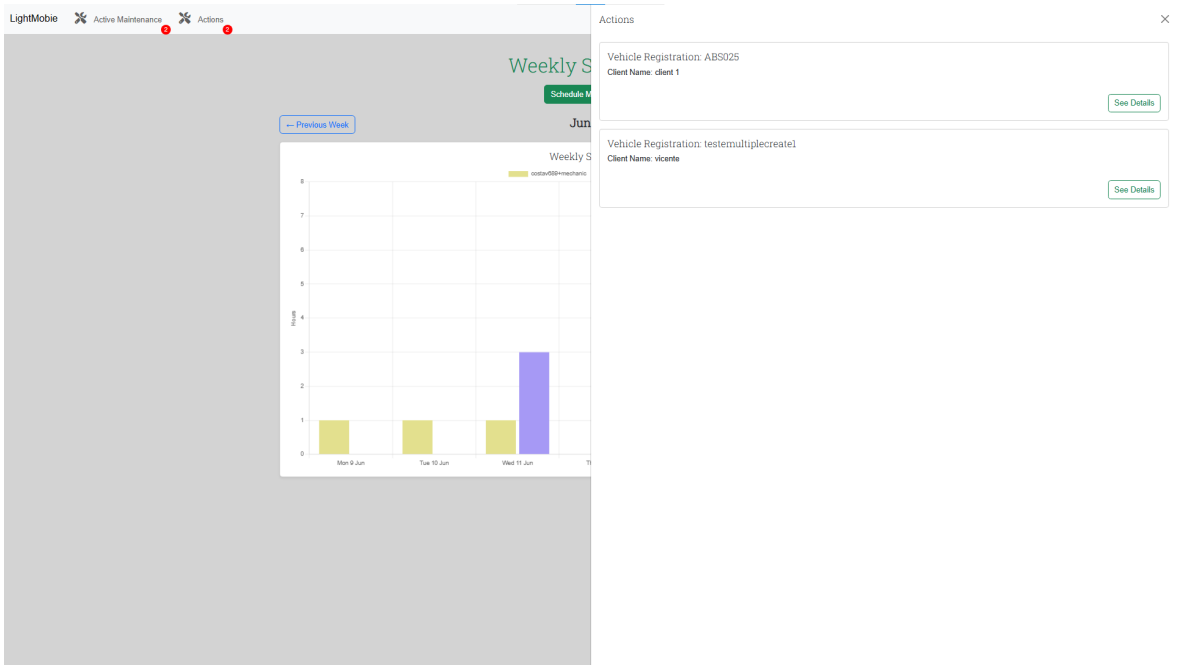
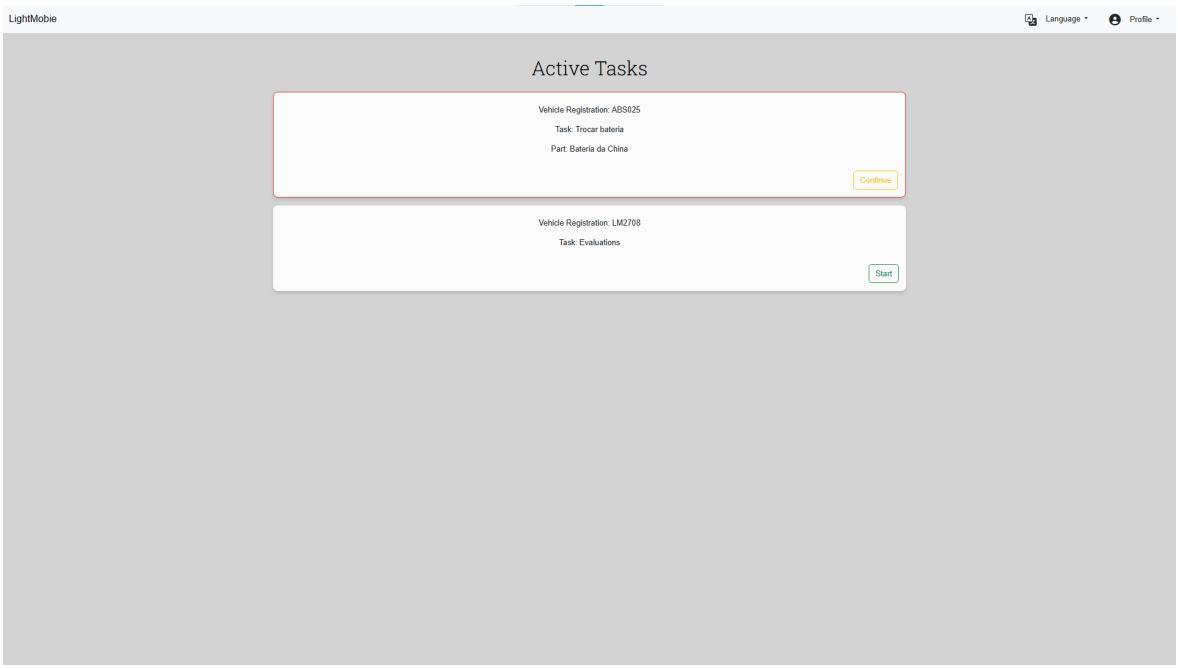


Figura 4.11: List of maintenances that the evaluation is finished and need to comunicate with the client to decide wich task to be done.



4.2.2 Mechanic view

Figura 4.12: Mechanic home page with simple, late and previous started task.

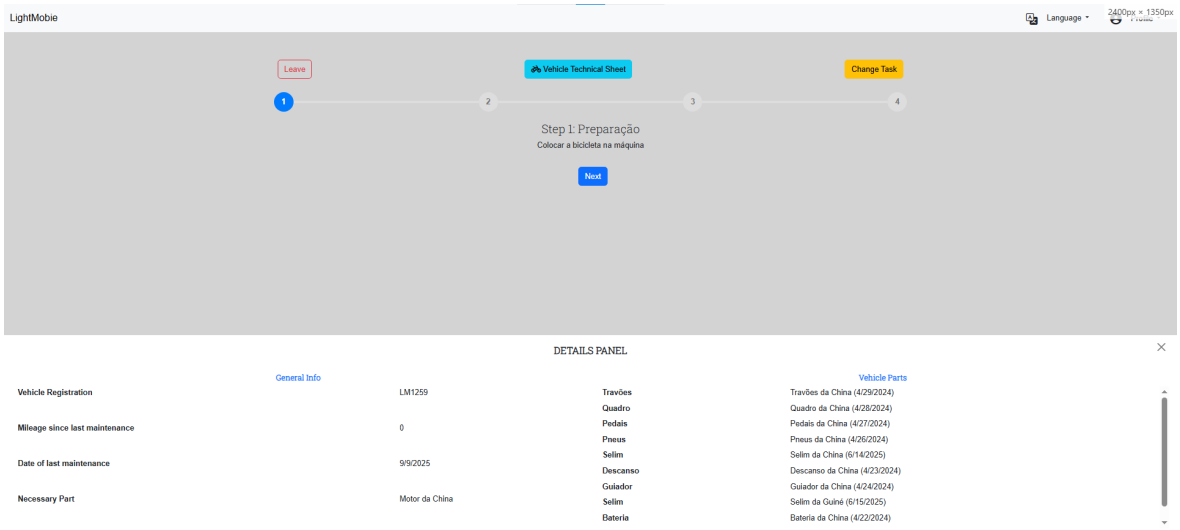


The mechanic interface mirrors the structure of the receptionist's view, but focuses on task execution and vehicle evaluation. On the home page (**??**), mechanics can see all tasks assigned to them for the day, including overdue tasks that were started previously but left incomplete. This directly supports Use Case 2.1 – View to-do list and Use Case 2.6 – Continue Task. Each task card provides essential information such as the vehicle registration, task name, and, where applicable, the part required to complete it.

When opening a task, the system guides the mechanic through a series of predefined steps (**??**). Each step includes instructions, navigation controls, and access to the vehicle's technical details, which can be viewed in an expandable panel (**??**). The technical sheet contains registration data, mileage since last maintenance, part history, and any client notes. If, during execution, the mechanic determines that a different part is needed, they may trigger a maintenance change request. This pauses the task, records the proposed modification, and initiates Use Case 2.5 – Change Task (Figure **??**). Once all steps are completed, the mechanic can leave notes and mark the task as finished, fulfilling Use Case 2.4 – Execute Maintenance Task (Figure **??**).

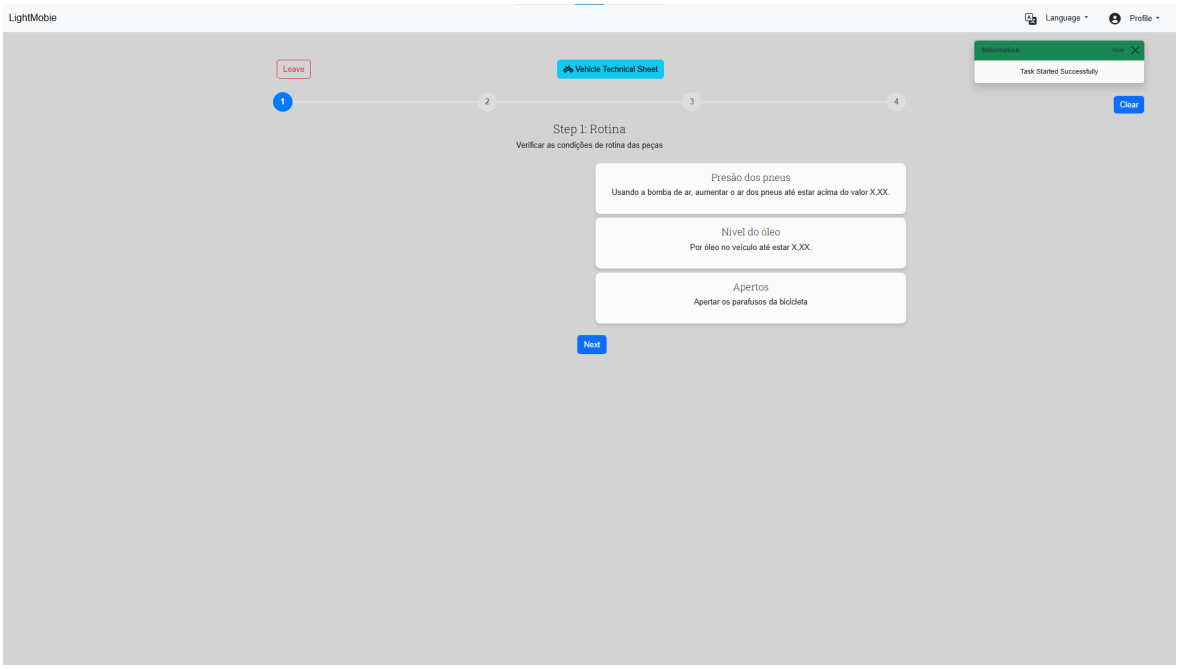
The interface also supports vehicle evaluations, where the mechanic identifies and registers new tasks (**??**-**??**). Starting an evaluation presents a list of potential tasks, each of which can be selected and, if necessary, linked to specific parts. The system checks warehouse stock levels in real time, alerting the mechanic if parts are unavailable so adjustments can be made. Selected tasks are visually highlighted, providing immediate feedback. In the final step of the evaluation (**??**), the mechanic reviews the chosen tasks and records the cause of each

Figura 4.13: Vehicle details during a task completion.



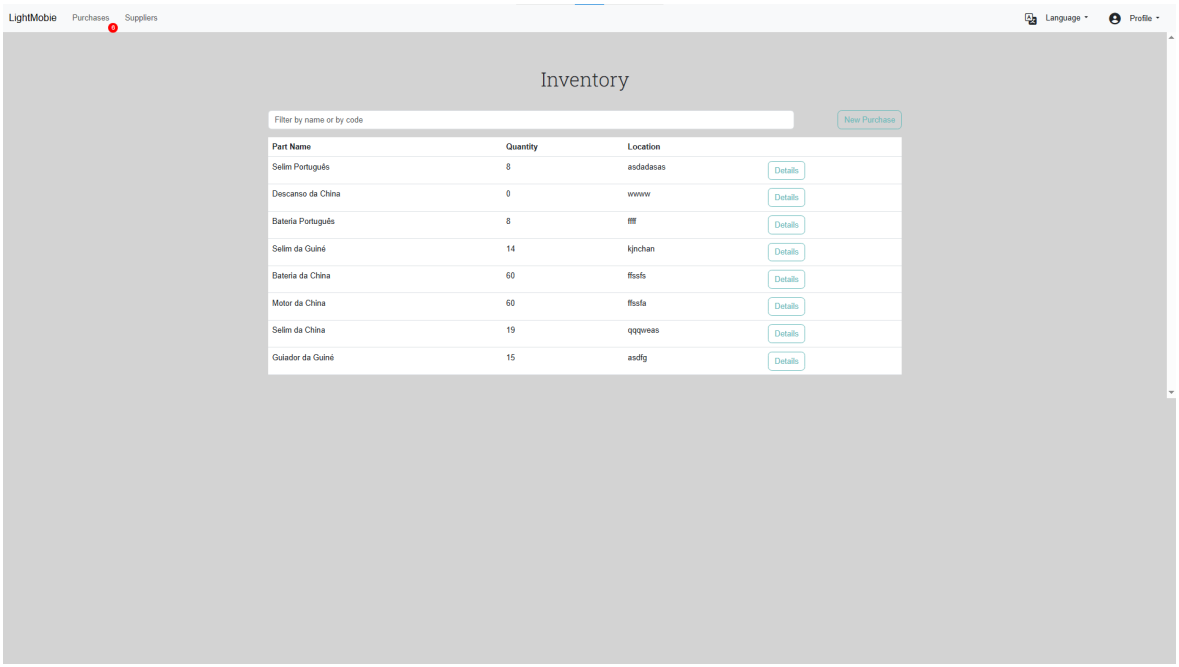
anomaly before submitting. Completing this workflow corresponds to Use Case 2.2 – Carry out a vehicle analysis or Use Case 2.3 – Register completed tasks, depending on whether the dealership configuration.

Figura 4.14: Mechanic Evaluation first step.



4.2.3 Warehouse Operator view

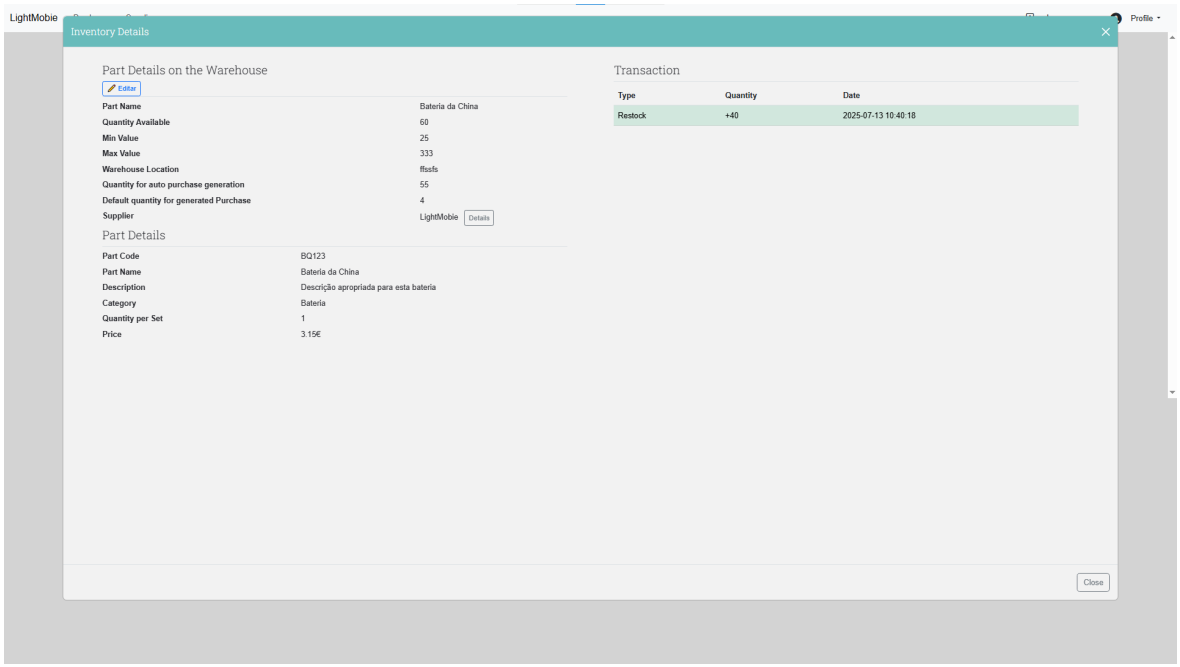
Figura 4.15: Warehouse home page.



The warehouse operator interface follows the same overall layout as the other views, with a top menu and a central workspace (**??**). Its primary purpose is to manage parts inventory, handle purchase requests, and maintain supplier information, covering all operations in Use

Case Group 3.

Figura 4.16: Inventory details.



The inventory section presents a searchable table of all parts stored in the warehouse, including their available quantity and location code (Use Case 3.1 – View warehouse parts). Each entry can be expanded to display detailed information such as stock thresholds, purchase configuration parameters, transaction history, and pricing (**??**). Operators can also edit inventory settings—such as location, minimum and maximum stock levels, and automatic purchase triggers—fulfilling Use Case 3.5 – Edit inventory (Figure **??**).

Purchases are managed directly from the home page. Operators may create new purchase requests by specifying a justification and selecting parts and quantities (**??**- **??**), which initiates Use Case 3.2 – Request purchasing service. All purchases are accessible from a dedicated view, where they are organized by status: assigned, awaiting delivery, or completed (**??**). Assigned purchases allow operators to review order details and record expected arrival dates (**??**). Waiting delivery purchases support both delaying arrivals (Use Case 3.6 – Create purchase delay) and registering the reception of parts (Use Case 3.4 – Register new parts in the system) (**??**). Delivered purchases remain available for reference (**??**).

Finally, the supplier management module enables operators to view and maintain supplier information. A dedicated view lists all suppliers with their contact details (**??**). Selecting a supplier provides extended information, including address, supplied parts, and contract duration (**??**). This functionality ensures that purchase negotiations and supplier relations remain transparent and up to date.

Figura 4.17: Create new purchase modal.

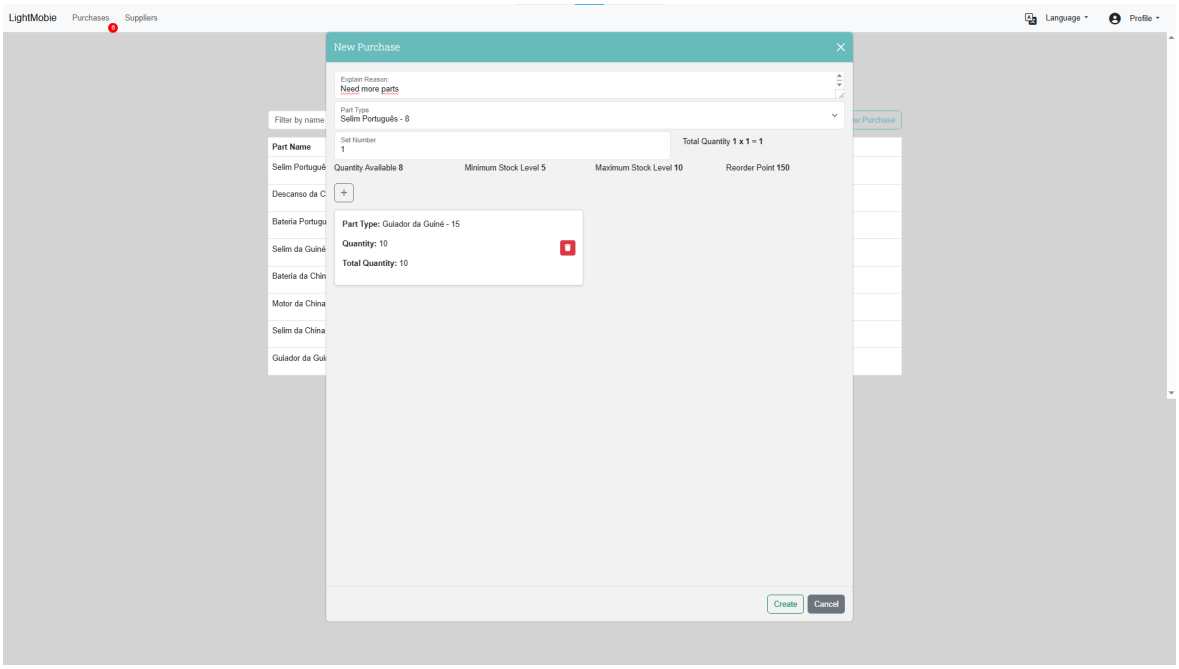


Figura 4.18: List of purchases.

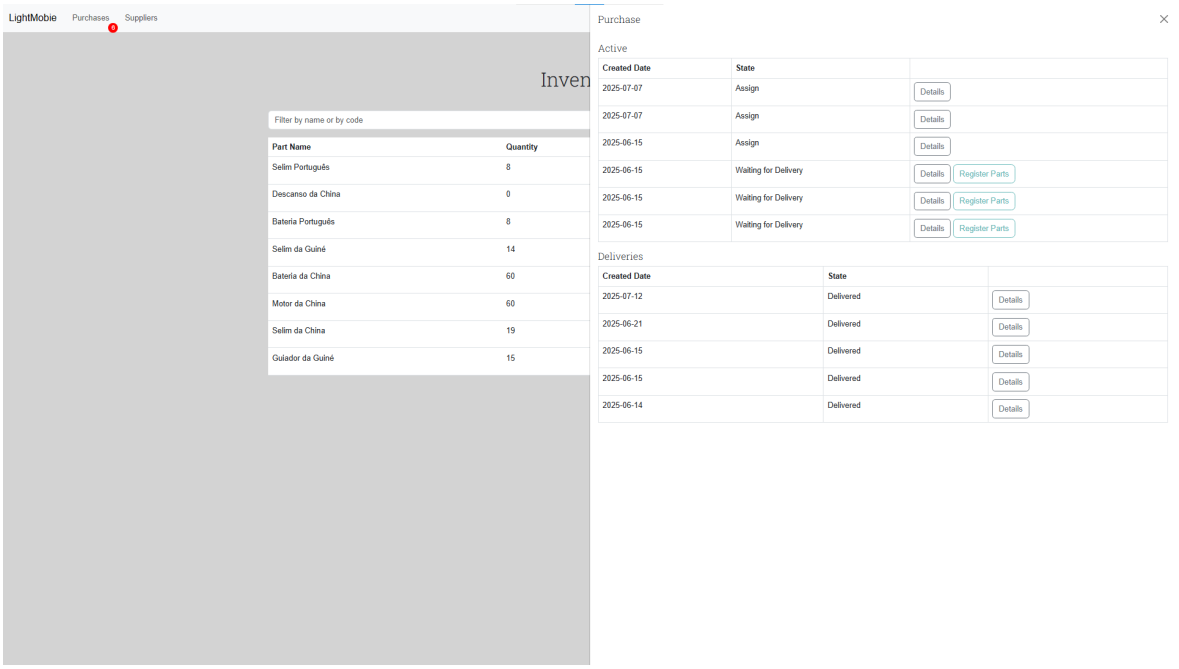
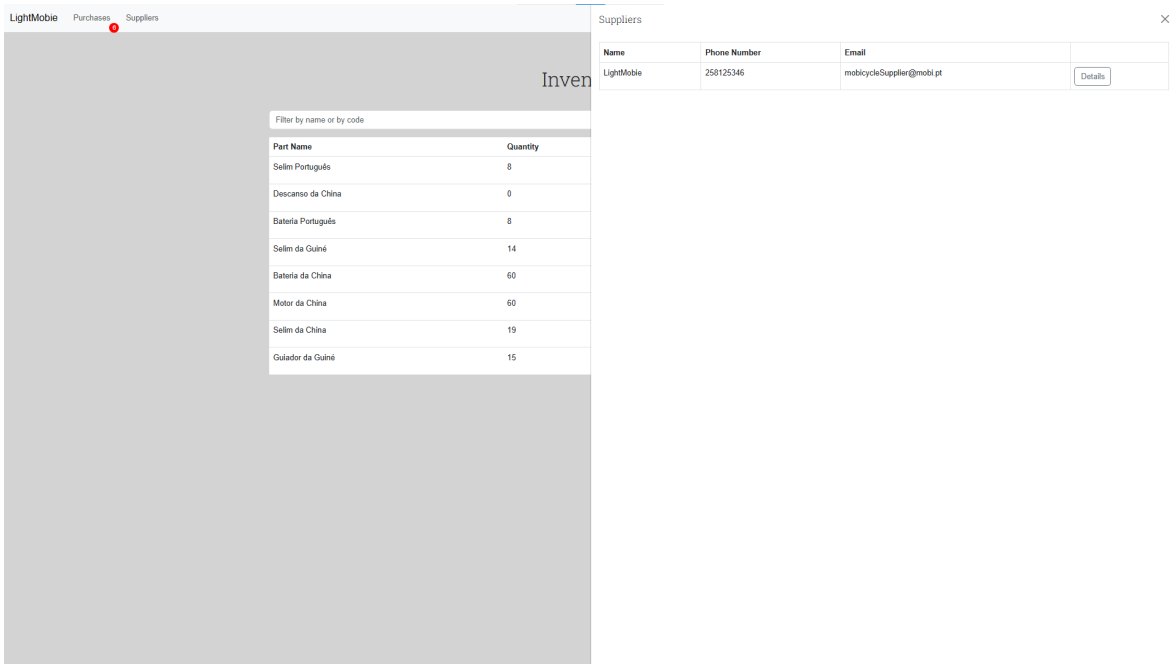


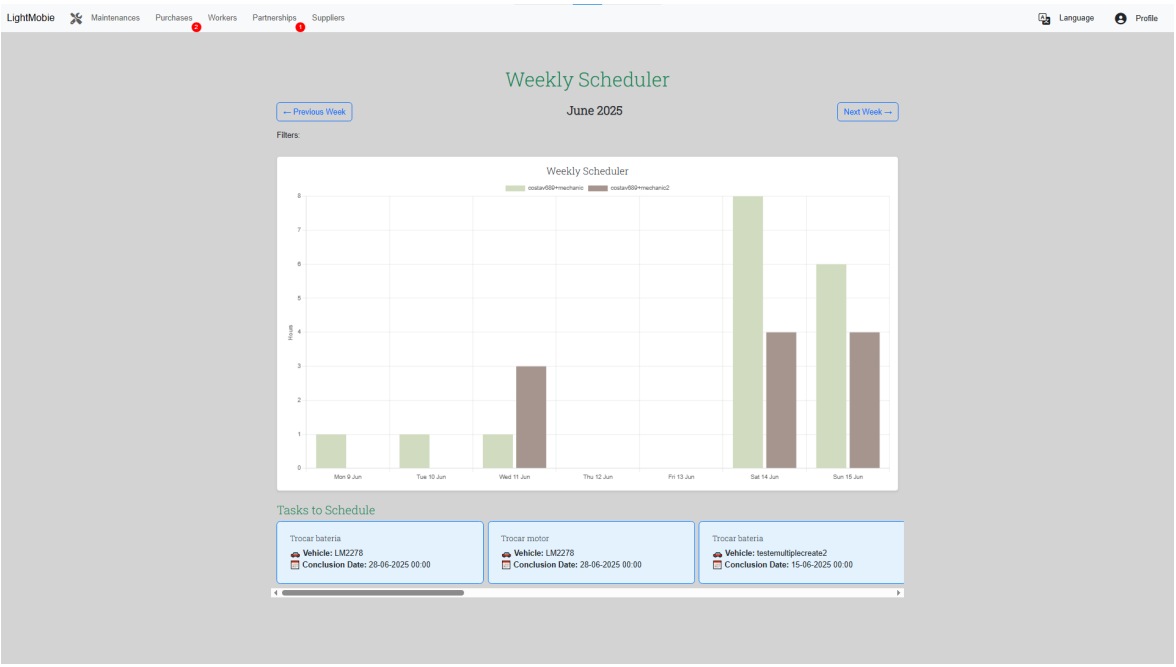
Figura 4.19: Supplier list.



4.2.4 Workshop Manager view

The workshop manager interface serves as the central hub for overseeing dealership operations. Its layout is consistent with the other views, but it integrates features from both the receptionist and warehouse operator roles, reflecting the manager's broader responsibilities.

Figura 4.20: Workshop manager home page.



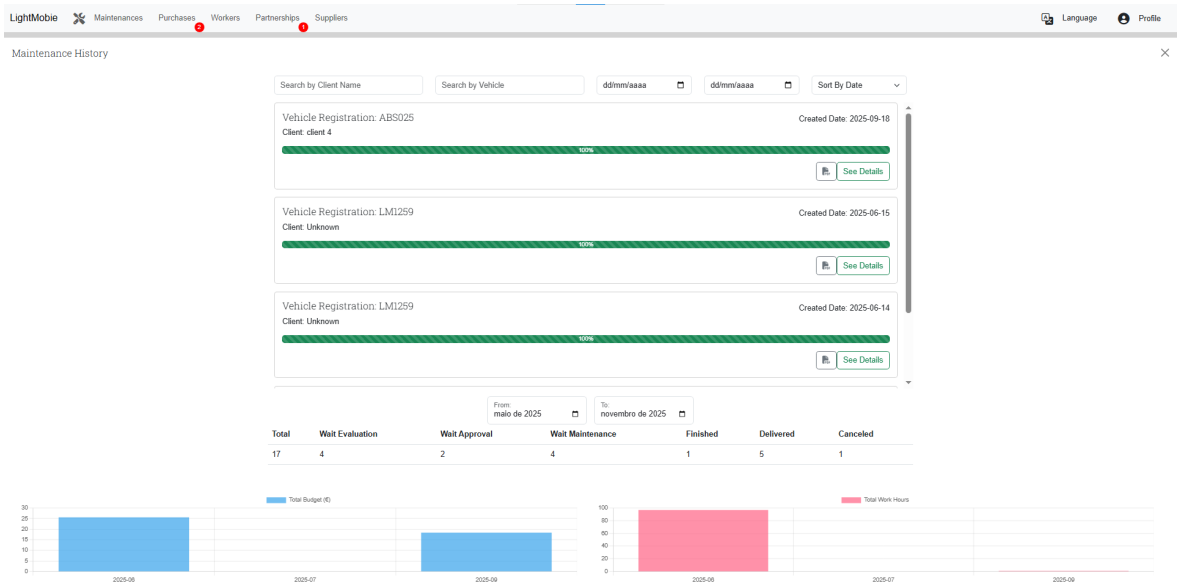
The home page presents a weekly overview of mechanic tasks (Figure **??**). Unlike the receptionist's schedule, it highlights unassigned tasks, which the manager can allocate to mechanics, fulfilling Use Case 4.1 – Assign tasks. This can be done either directly from the schedule or by opening a task assignment modal (Figure **??**).

Maintenance management mirrors the receptionist's view: active maintenances are listed with their details and task breakdowns (**??**–**??**). The key distinction is that the manager can add new tasks to an ongoing maintenance, which triggers a client validation process. In addition, the manager has access to the maintenance history, supporting Use Case 4.3 – View history of maintenance performed. Past maintenances can be filtered by client, vehicle, or date, and completed cases allow generating PDF reports (**??** and **??**). Below the history list, statistics such as revenue and working hours are displayed, corresponding to Use Case 4.4 – Develop statistics.

Purchase management also extends the warehouse operator's functionality. Managers review purchase requests submitted by operators and either authorize or reject them (Use Case 4.2 – Authorize purchase). When authorizing, they may also add additional parts and assign the request to a specific operator (**??**–**??**).

The worker management module allows the manager to view dealership employees, edit their roles, and register new workers (Use Case 4.5 – Assign roles to employees). Each worker's

Figura 4.21: Maintenance history list.



details include personal information and role, with options to modify or remove them (Figures **??**–**??**).

Partnership requests are handled through a dedicated view. New partnership proposals from external entities list the entity name, vehicle types, and creation date, and can be either accepted or rejected (Use Case 4.8 – Accept/Reject partnership). Confirmed partners are displayed in a separate table (**??**).

Finally, supplier management is identical to the warehouse operator view, displaying supplier details, contact information, and contract terms for parts provided (**??**–**??**).

Figura 4.22: Purchase request list.

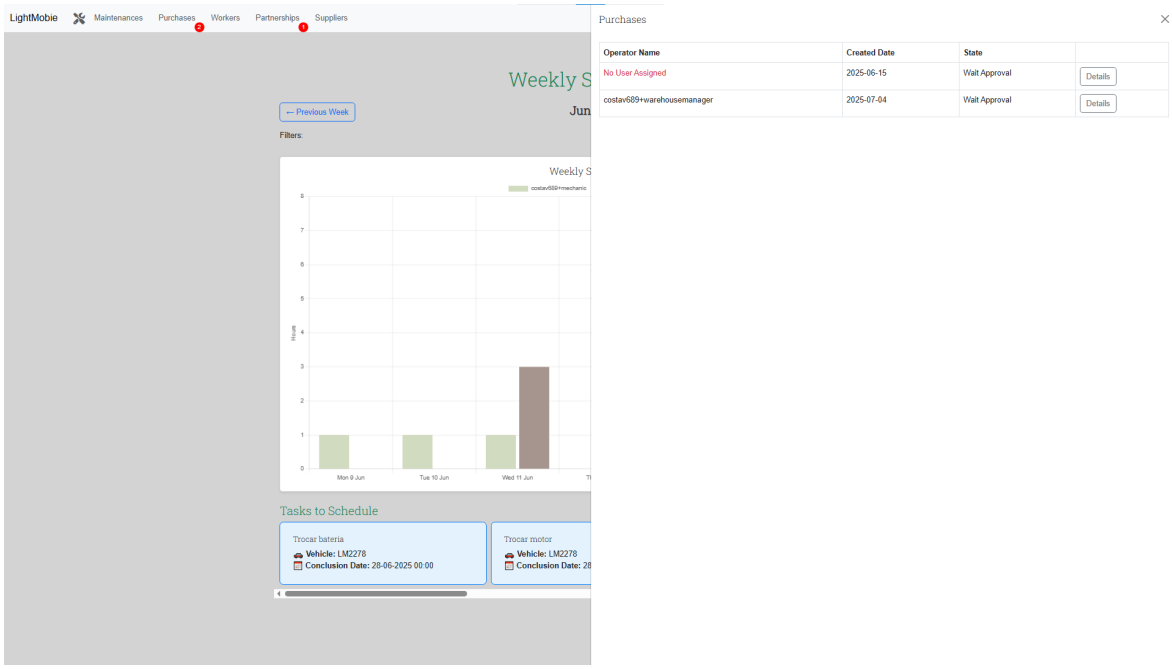


Figura 4.23: Worker list.

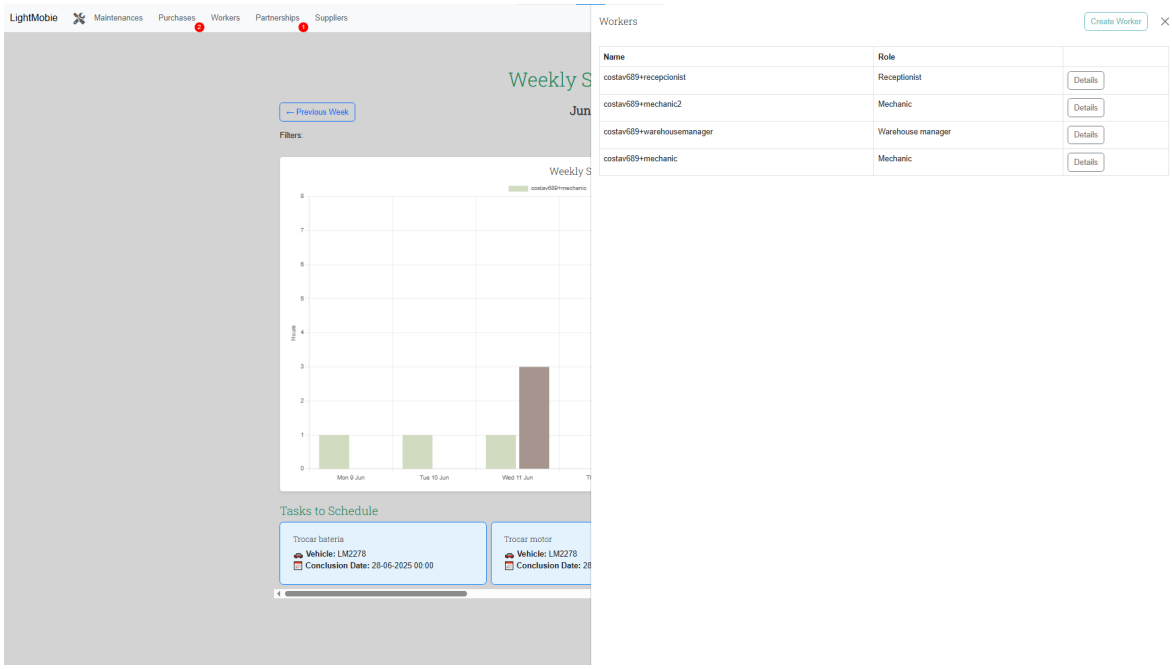
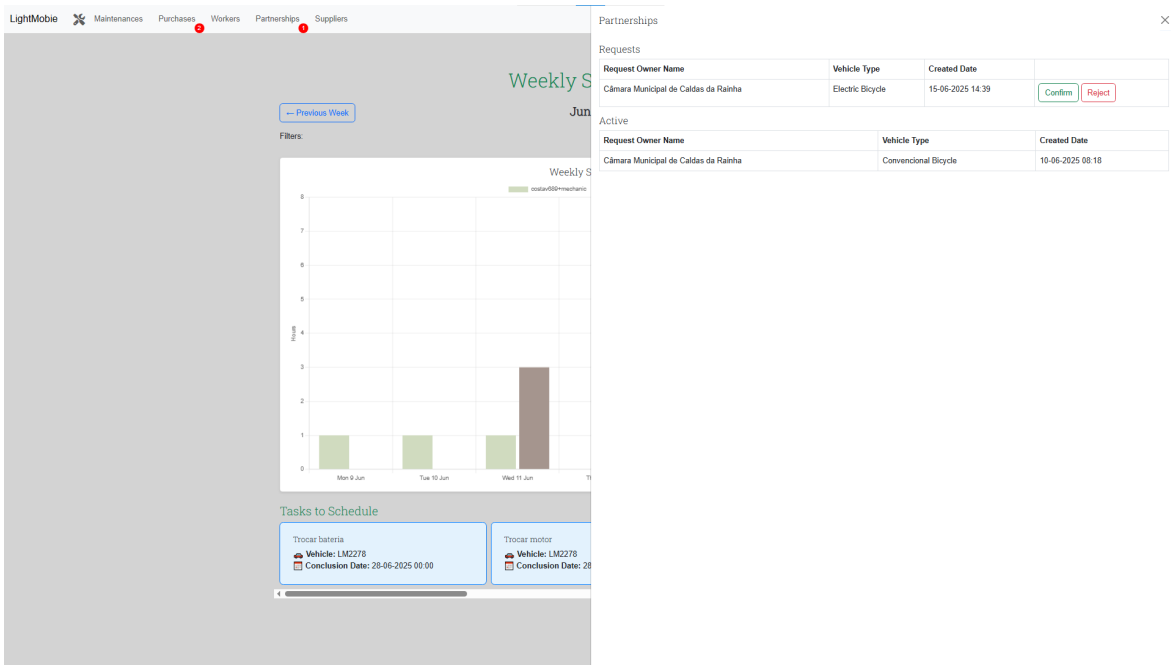
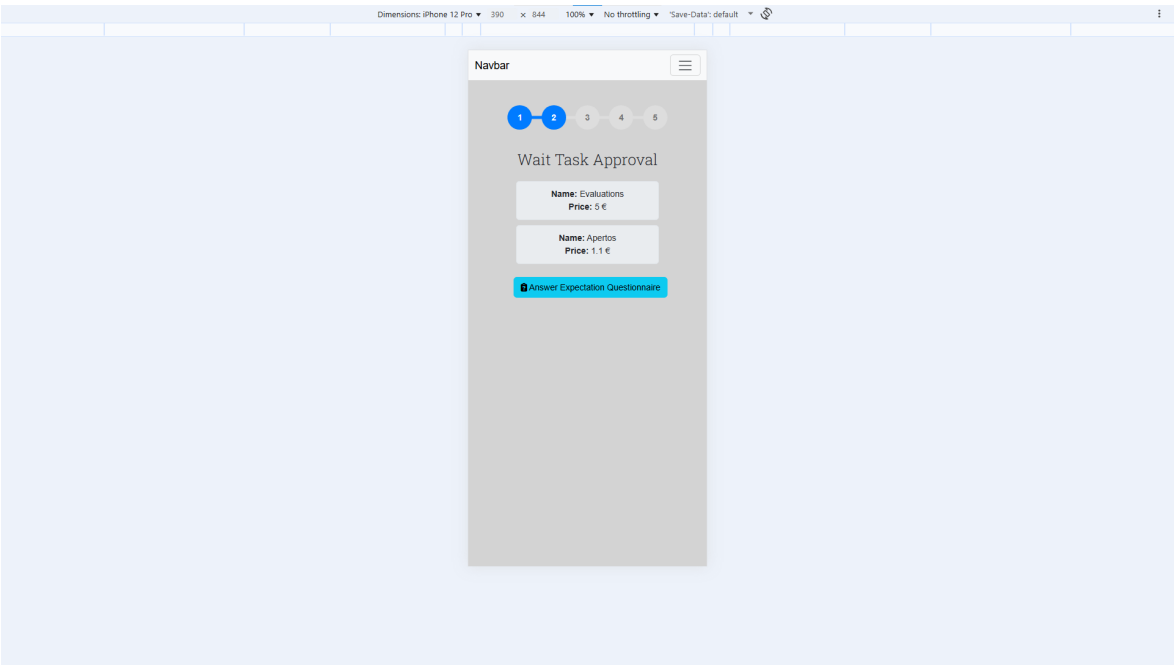


Figura 4.24: Partnership list.



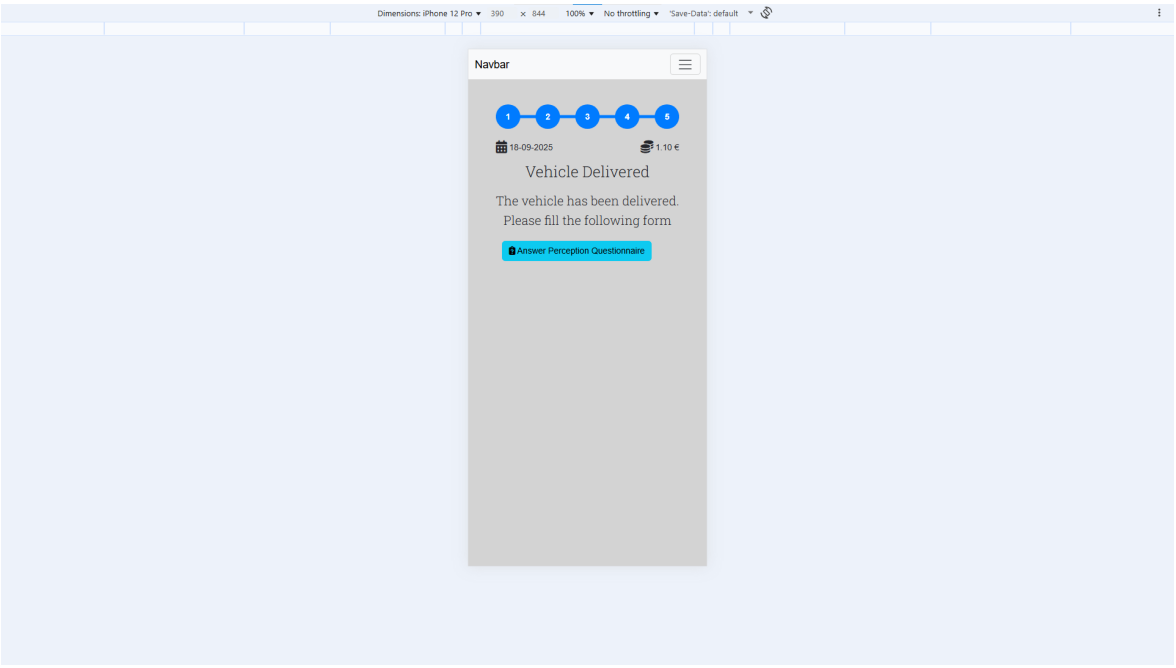
4.2.5 Client view

Figura 4.25: Client Home page when the vehicle evaluation is complete.



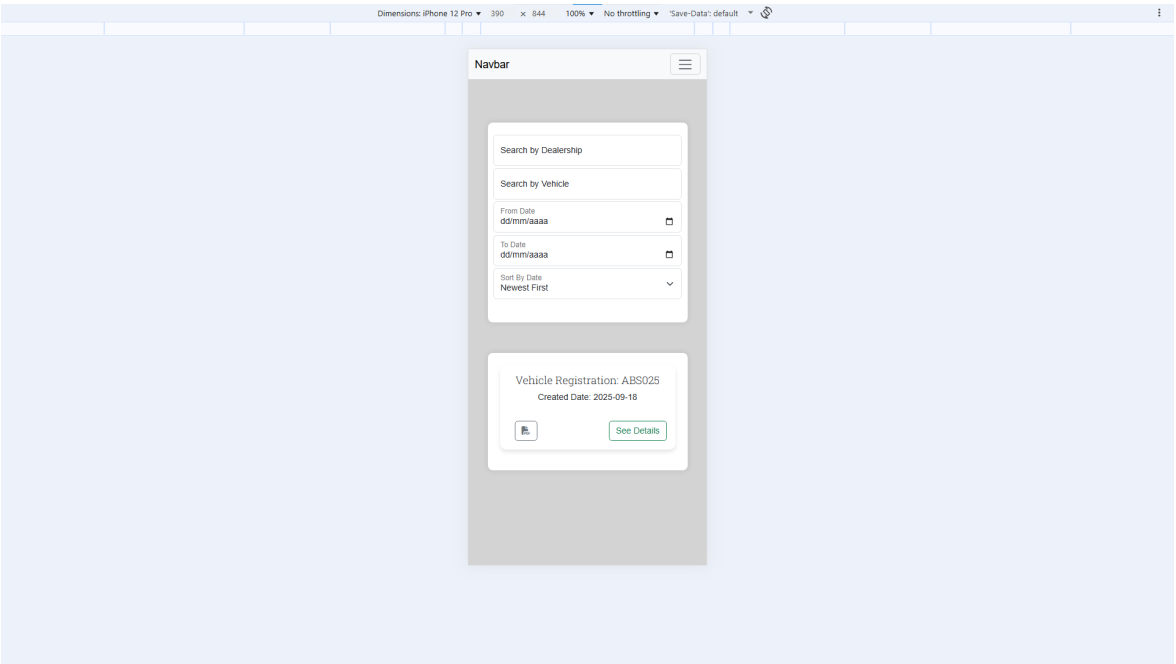
The client interface is designed for mobile use and focuses on tracking the status of a vehicle's maintenance. Its layout differs from the employee views by being streamlined and tailored to end-user needs.

Figura 4.26: Client Home page when the vehicle after the vehicle is delivered to the client.



When a maintenance is scheduled with a dealership, the home page displays the upcoming evaluation date (**??**). Once the evaluation is complete, the client sees the list of proposed tasks, which must be confirmed before the maintenance proceeds (**??**). After approval, the client can monitor task progress, following their completion status step by step (**??**–**??**). When all tasks are finished, the application notifies the client that the vehicle is ready for pickup, and once the delivery is completed, the client can rate the service received (**??**).

Figura 4.27: List of completed maintenances of the client.



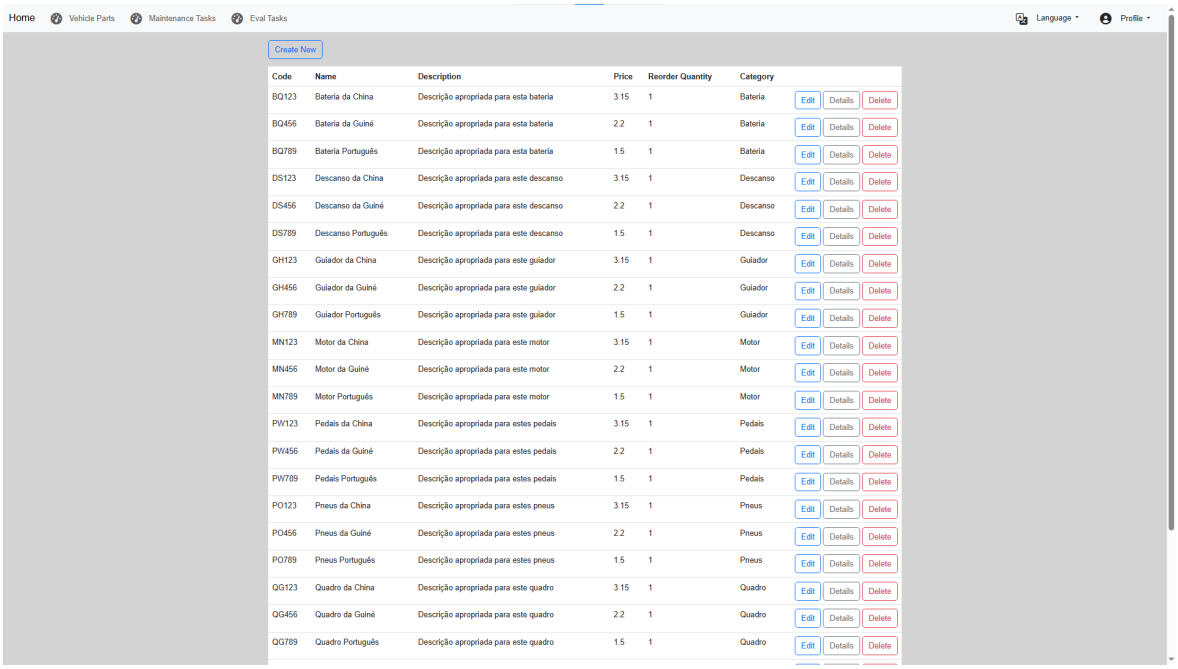
In addition to active maintenances, clients can review their full maintenance history (Use Case 5.5 – View maintenance history). The history view supports filtering by vehicle, dealership, or date (**??**). For each entry, clients may download a PDF report—identical to the one generated by the workshop manager—or open a detailed modal with information and task breakdowns (**??**,**??** and **??**).

4.2.6 Administrator view

The administrator interface is dedicated to configuring the system's core information, namely the catalog of parts, task types, and evaluation tasks. Upon logging in, the administrator is presented with a list of part types (**??**). From this view, parts can be created, edited, deleted, or inspected in detail (Figures **??**–**??**). Each operation follows the same interaction pattern: selecting an action in the table opens a form or details modal, ensuring consistency and efficiency.

The same layout is applied to managing task types (Figures **??**–**??**) and evaluation tasks (Figures **??**–**??**), maintaining uniformity across all system configuration sections. This design allows the administrator to quickly perform updates with minimal effort, ensuring that the dealership's catalog of parts and tasks remains accurate and up to date.

Figura 4.28: Parts type list.



4.2.7 Summary

In summary, the implementation of the different application views demonstrates that the overall system requirements were successfully met. Each user role — receptionist, workshop manager, mechanic, warehouse operator, client, and administrator — was provided with a tailored interface designed to support their specific tasks. Although some views share similarities in layout and functionality, they each incorporate the particular features required to fulfill their responsibilities.

CAPÍTULO 5

Results

This chapter talks about the test made with the users.

5.1 Results

To evaluate the application, I designed a set of user tests tailored to each role-specific interface. These tests consisted of one or more tasks that participants were asked to complete, or sequences of questions they needed to answer. Once a participant finished testing an application, they completed a questionnaire to assess its usability and effectiveness. The same questionnaire, shown in **??**, was used across all applications to ensure consistency in evaluation, while the specific user test tasks are presented in **??**. The goal of these tests was to cover the main requirements of the system and validate the most relevant features of each application.

5.1.1 Receptionist

Overall, participants were able to complete most tasks without major difficulties, although task 6 (maintenance approval) caused confusion, as users struggled to locate the necessary form and required assistance.

The feedback highlighted several usability issues and bugs. The main limitation was that users needed to close the modals for maintenance approval or maintenance detail changes in order to access dealership occupation information, interrupting the workflow. The task filter in the maintenance scheduling modal also behaved unintuitively: clicking on a mechanic removed the filter instead of restricting results to that mechanic. Users suggested changing the behavior so that only the tasks of the selected mechanic are shown.

Other comments included the high sensitivity of the time scroll input, with a preference for keyboard entry, the need to improve graph readability by clarifying labels and adding pointers, and adding a confirmation step when canceling a maintenance. Participants also pointed out that the modal actions could be integrated into the maintenance details tab for

easier access. Finally, inconsistencies were noted in the display of task prices, and the system lacked error messages when canceling maintenances.

5.1.2 Workshop Manager

The workshop manager tests were completed without external help, but several interface improvements were suggested. Participants found the description text in some modals unclear, and the statistics filter for maintenance history was initially malfunctioning, though this was later corrected.

Additional feedback included removing the redundant "active maintenances" category from the history view, ensuring purchase requests rejected by the manager function as intended, and improving form handling by automatically clearing and closing forms after actions such as employee creation or purchase assignment. Users also suggested displaying hours and minutes in statistics instead of decimals, and allowing keyboard input for dates.

5.1.3 Mechanic

For the mechanic interface, users were able to complete tasks independently. However, they raised concerns about the usability of certain elements. The "Ficha técnica" button was not sufficiently visible, and the step-by-step navigation was considered less intuitive compared to a single-page task overview.

Users also suggested moving the "Change Task" button outside the offcanvas panel, ensuring the modal for task changes functions reliably, and adding clearer step names to the task navigation menu.

5.1.4 Warehouse Operator

The warehouse operator interface allowed participants to complete their tasks with ease, but feedback pointed to improvements in search and data entry. Specifically, when creating a purchase, users wanted to be able to search parts by typing rather than selecting from a static list.

Other suggestions included requiring a date when registering delays, repositioning the delay registration button to the main modal body, and refining purchase creation with sets so that redundant total values are not shown.

5.1.5 Client

Finally, the client interface results showed that users could perform tasks without difficulty. However, the evaluation step was found to be unclear, with participants preferring a star-rating system to better express satisfaction.

In the maintenance history, some users perceived the menu option as disabled, indicating a need for clearer design. Additionally, participants suggested that when no active maintenances are present, the screen should display dealership contact information rather than remaining empty, improving usefulness.

5.1.6 Conclusion

Overall, the user tests demonstrated that the main functionalities of each application were usable and allowed participants to complete their assigned tasks. However, the feedback consistently emphasized the importance of improving clarity, visibility, and workflow efficiency across the system. In particular, users identified issues with modal interactions, filter behavior, form handling, and the visibility of certain interface elements. Many of these were relatively small adjustments but had a significant impact on perceived usability. Addressing these points will not only resolve current limitations and bugs but also create a smoother, more intuitive experience for all user roles.

CAPÍTULO 6

Conclusion

This final chapter presents the conclusions of the dissertation. It begins by summarizing the main objectives and results of the project, followed by a discussion of the limitations identified during implementation and evaluation. The chapter closes with proposals for future work that could further enhance and extend the developed system.

The work presented in this dissertation focused on the design and development of a dealership maintenance management platform that integrates the different perspectives of its users: receptionist, workshop manager, mechanic, warehouse operator, client, and administrator. Each role was given a dedicated view, carefully tailored to the requirements identified during the analysis phase, with the objective of ensuring that the entire maintenance workflow could be executed within a single system.

The implementation results confirmed that the majority of the functional requirements were successfully achieved. User tests demonstrated that participants were able to complete their assigned tasks across the different applications, thereby validating the correctness of the main workflows. Furthermore, the tests highlighted that the system provides a consistent and coherent experience across roles, enabling a smooth interaction between the dealership's front office, workshop, and warehouse operations.

Nevertheless, the evaluation also uncovered several limitations. Some interfaces presented usability challenges, such as filters that did not behave as expected, forms that were not cleared after submission, or modal dialogs that forced users to navigate back and forth to gather all the necessary information. Additionally, a few bugs were identified that affected clarity and efficiency, including inconsistencies in pricing information, insufficient visibility of interface elements, and interactions that required unnecessary steps. Although these limitations did not prevent the successful completion of tasks, they showed that further refinement is needed to achieve a more intuitive and user-friendly system.

Overall, this project confirms the feasibility of the proposed approach and delivers a functional prototype capable of addressing the key challenges of dealership maintenance

management. At the same time, it opens the door for future enhancements that can significantly increase the platform's usability, scope, and integration with related systems.

6.1 Future Work

Building on the current results, several directions for future improvements have been identified:

- Develop an internal messaging system to enable direct communication between the receptionist and the mechanics.
- Extend the administrator view to support the creation, removal, editing, and visualization of part categories.
- Allow mechanics to execute tasks that are not explicitly assigned to them.
- Simplify the user creation process by removing the password field and enabling employees to configure their credentials upon email confirmation.
- Provide functionality for assigning multiple tasks to a mechanic simultaneously.
- Add statistical tools for the workshop manager, such as average task completion time and service quality indicators.
- Automatically pause a task if a mechanic logs out of the platform while performing it.
- Integrate a billing module, leveraging the one already developed for the Lightmobie bike-sharing platform.
- Enhance the client evaluation form by allowing users to leave written recommendations or final thoughts in addition to ratings.
- Enable managers to modify the list of parts required for a task.
- Allow mechanics to adjust task parts before starting their execution.
- Require managers to specify a reason when rejecting a purchase request.

These improvements not only address limitations observed during the evaluation phase but also introduce new capabilities that could expand the system's applicability in real-world scenarios. Future iterations of the platform should therefore focus on usability refinements, deeper role flexibility, and stronger integration with dealership and client-facing processes.

APÊNDICE A

Additional content

Figura A.1: Front page of the paper form used by EMEL during vehicle maintenance. Operators use this section to record identified malfunctions.

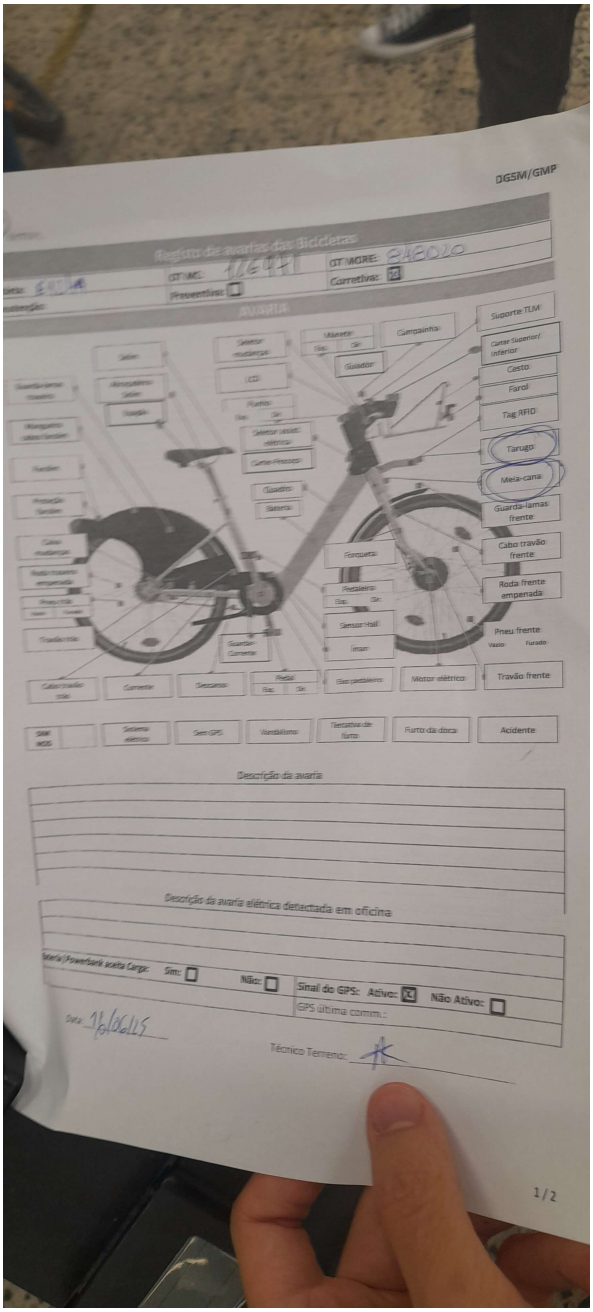
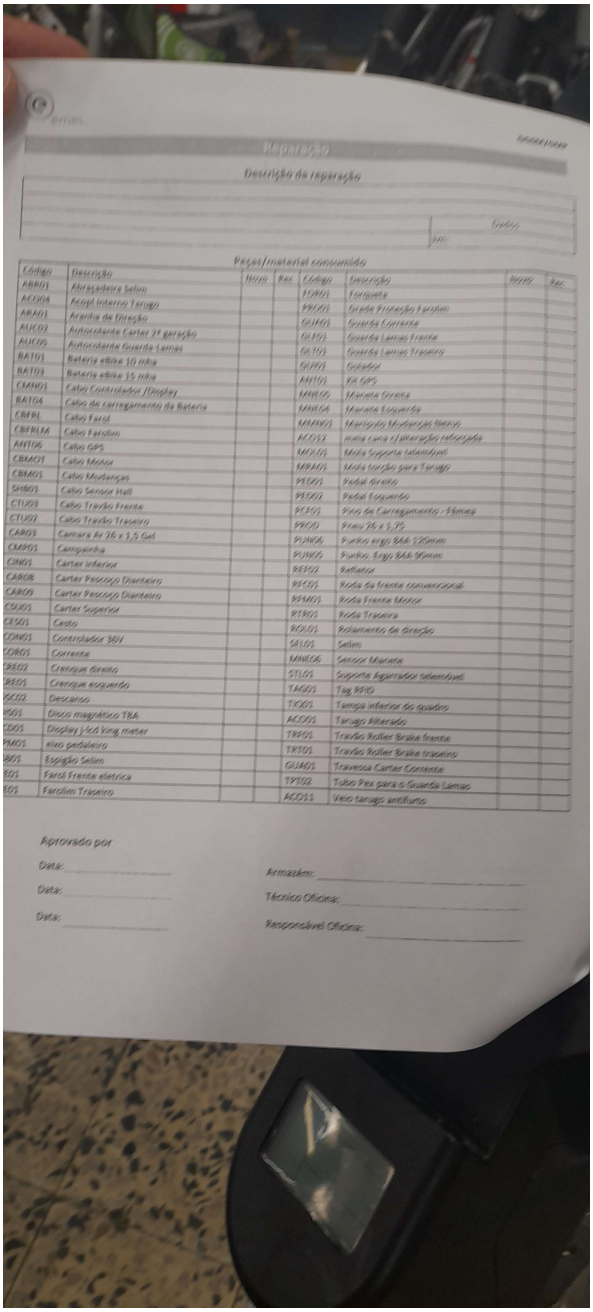


Figura A.2: Back page of the paper form used by EMEL during vehicle maintenance. Mechanics use this section to document the parts replaced during the repair process.



Reparação

Descrição da reparação

Peças/material consumido

Código	Descrição	Horiz	Rec	Código	Descrição	Horiz	Rec
ABR01	Abraçadeira Selim			FOR01	Forqueta		
ACC018	Acopl Interno Tarugo			PRG01	Grade Proteção Farolim		
ARA01	Aranha de Direção			GUA01	Guarda Corrente		
AUC02	Anticolante Carter 2ª geração			GL101	Guarda Lamas Frente		
AUC05	Anticolante Guarda-Lamas			GL T01	Guarda Lamas Traseiro		
BAT01	Bateria eBike 10 mha			GUI01	Guiador		
BAT03	Bateria eBike 15 mha			ANT01	Kit GPS		
CAN01	Cabo Controlador /Display			MAN05	Manete Direita		
BAT04	Cabo de carregamento da Bateria			MAN04	Manete Esquerda		
CBFRL	Cabo Farol			MAN01	Manipulo Mudanças Nexus		
CBFRLM	Cabo Farolim			ACC03	mola cana c/alteração reforçada		
ANT06	Cabo GPS			MOL01	Mola Suporte telemóvel		
CBMOT	Cabo Motor			MRA01	Mola torção para Tarugo		
CBM01	Cabo Mudanças			PED01	Pedal direito		
SHB01	Cabo Sensor Hall			PED02	Pedal Esquerdo		
CTU03	Cabo Travão Frente			PC101	Pino de Carregamento - fêmea		
CTU02	Cabo Travão Traseiro			PIN10	Pino 26 x 1,75		
CAR03	Camara Ar 26 x 1,5 Gel			PUN06	Punho ergo BAK 125mm		
CMP01	Campainha			PUN05	Punho Ergo BAK 95mm		
CIN01	Carter inferior			RAD02	Radiator		
CAR08	Carter Pescoço Dianteiro			RDC01	Roda da frente convencional		
CAR09	Carter Pescoço Dianteiro			RDM01	Roda Frente Motor		
CSU01	Carter Superior			RTR01	Roda Traseira		
CES01	Cesto			ROL01	Rolamento de direção		
CON01	Controlador 36V			SEL01	Selim		
COR01	Corrente			MAN06	Sensor Manete		
CRE02	Cranque direito			STL01	Suporte Agarrador telemóvel		
CRE01	Cranque esquerdo			TAG01	Tag RFID		
DSC02	Descanso			TIC01	Tampa inferior do quadro		
MO1	Disco magnético TBA			ACC01	Tarugo Alterado		
CD01	Display LCD King meter			TRF01	Travão Roller Brake frente		
EIX01	eixo pedaleiro			TRT01	Travão Roller Brake traseiro		
SB01	Espigão Selim			GUA01	Travessa Carter Corrente		
FR01	Farol Frente eletrica			TPT02	Tubo Pex para o Guarda Lamas		
FR01	Farolim Traseiro			ACC11	Veio tarugo antifurto		

Aprovado por

Data: _____ Armazém: _____

Data: _____ Técnico Oficina: _____

Data: _____ Responsável Oficina: _____

Figura A.3: SERVQUAL statements used by the authors in the study to measure the quality of the service at a CMV SA dealership in South Africa (**Measuring_After_sales_Service_Quality**).

Dimensions	Statements
Tangible	1. The service dealers have modern-looking equipments.
	2. The physical facilities are visually appealing in dealer's shop.
	3. Employees appear neatly at work.
	4. Materials associated with service visually appealing at workplace.
Reliability	5. The service dealers always keep their promises to customers.
	6. The service dealers always show interests in customers needs
	7. The service dealers perform their services right at the first time.
	8. The service dealers provide their services in their promised time.
	9. The service dealers endeavor on error-free records
Respon-siveness	10. Employees are aware of the time period a job can be done.
	11. Employees are able to provide efficient service to customers.
	12. Employees are willing to help customers when there is a need.
	13. Employees always try to respond to customers requests promptly.
Assurance	14. Employees' working style enhances customer's confidence.
	15. Customers feel safe and comfortable in their transactions.
	16. Employees always are consistently courteous to customers.
	17. Employees have the knowledge to answer customer's questions.
	18. The service dealers give individual customer attention.
Empathy	19. The service dealers make convenient operating hours to customers.
	20. The dealers will organize employees to attend to individual customer.
	21. Excellent service dealers will have the customers' best interest in heart.
	22. The employees will understand the specific needs of their customers.

Figura A.4: SERVQUAL results of the study to measure the quality of the service at a CMV SA dealership in South Africa. In the table the header "Exp"means the expectation, "Per"means the perception, "Zc"means the Service quality score from the customers and the "Zc-ms"means Service quality score measure by the difference of the expectations of the customer and the expectation of the dealers. (**Measuring_After_sales_Service_Quality**)

Dimensions	No	Customers Exp. & Per. Z			Manager & Staff	Difference
		Exp (1)	*Perc* (2)	*Zc* (3)=(2)-(1)	*Exp* (4)	$Z_{c\text{-}ms}$ (4-1)
Tangible	1	4.82	4.29	-0.53	4.25	-0.57
	2	4.76	4.18	-0.58	4.75	-0.01
	3	4.82	4.35	-0.47	4.5	-0.32
	4	4.59	3.88	-0.71	4.75	0.16
Reliability	5	4.65	3.82	-0.83	4.75	0.10
	6	4.65	3.53	-1.12	4.5	-0.15
	7	4.94	3.35	-1.59	4.75	-0.19
	8	4.71	3.47	-1.24	4.5	-0.21
	9	4.88	3.71	-1.17	4.5	-0.38
Responsiveness	10	4.82	3.94	-0.88	4.75	-0.07
	11	4.88	3.94	-0.94	4.75	-0.13
	12	4.76	4	-0.76	4.75	-0.01
	13	4.71	3.76	-0.95	4.5	-0.21
Assurance	14	4.71	3.94	-0.77	4.75	0.04
	15	4.53	3.65	-0.88	4.75	0.22
	16	4.65	3.82	-0.83	4.5	-0.15
	17	4.82	4.12	-0.7	4.5	-0.32
	18	4.76	3.65	-1.11	4.75	-0.01
Empathy	19	4.65	3.94	-0.71	4.75	0.10
	20	4.71	3.71	-1	4.75	0.04
	21	4.82	3.82	-1	4.75	-0.07
	22	4.88	3.88	-1	4.75	-0.13
Mean		4.75	3.85		4.65	-0.10
Std		0.11	0.25		0.15	0.04

Figura A.5: Active maintenance details Information tab.

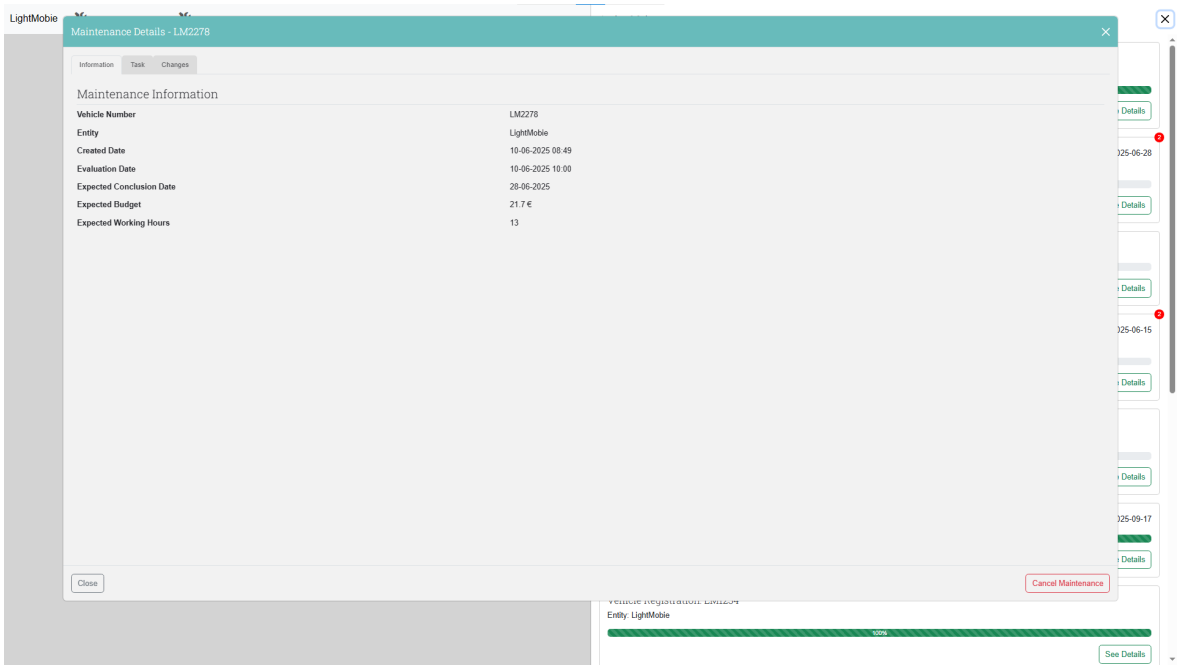


Figura A.6: Active maintenance details task tab.

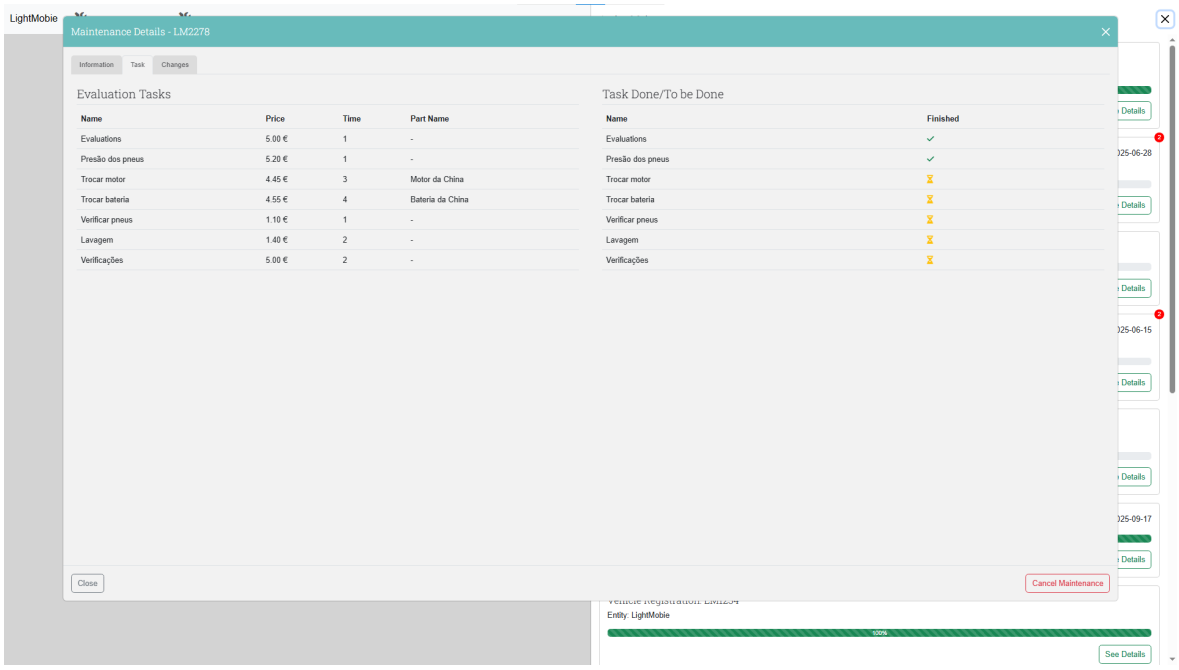


Figura A.7: Active maintenance details change tab.

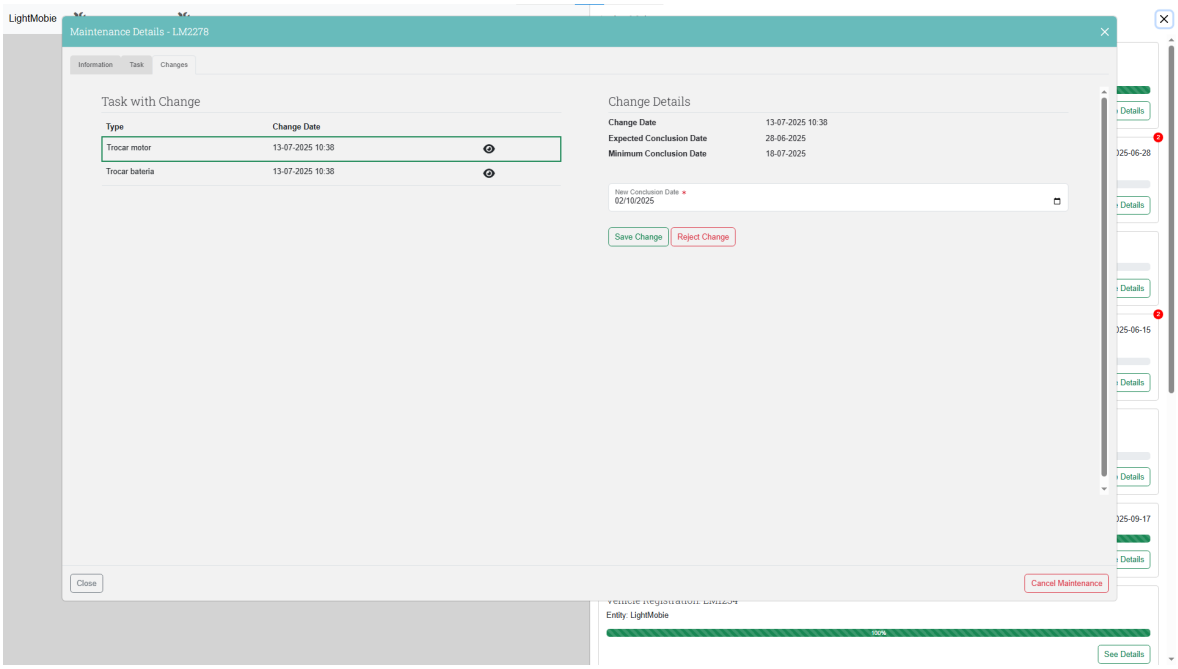


Figura A.8: Maintenance action details.

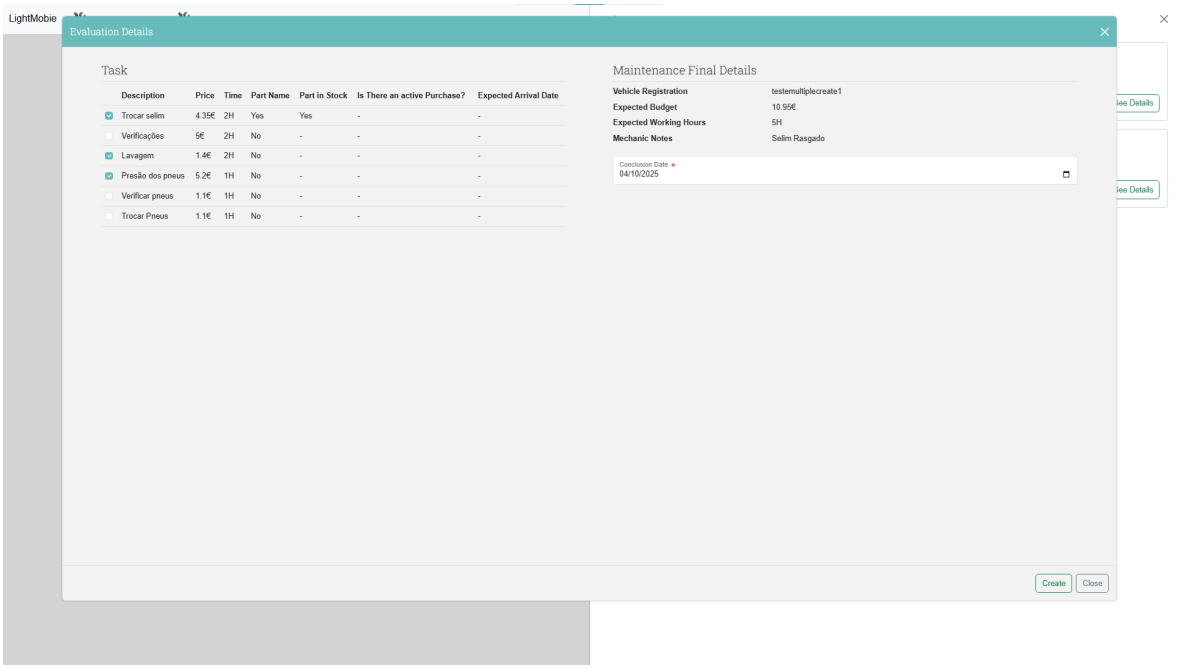


Figura A.9: Mechanic completing simple task first step.

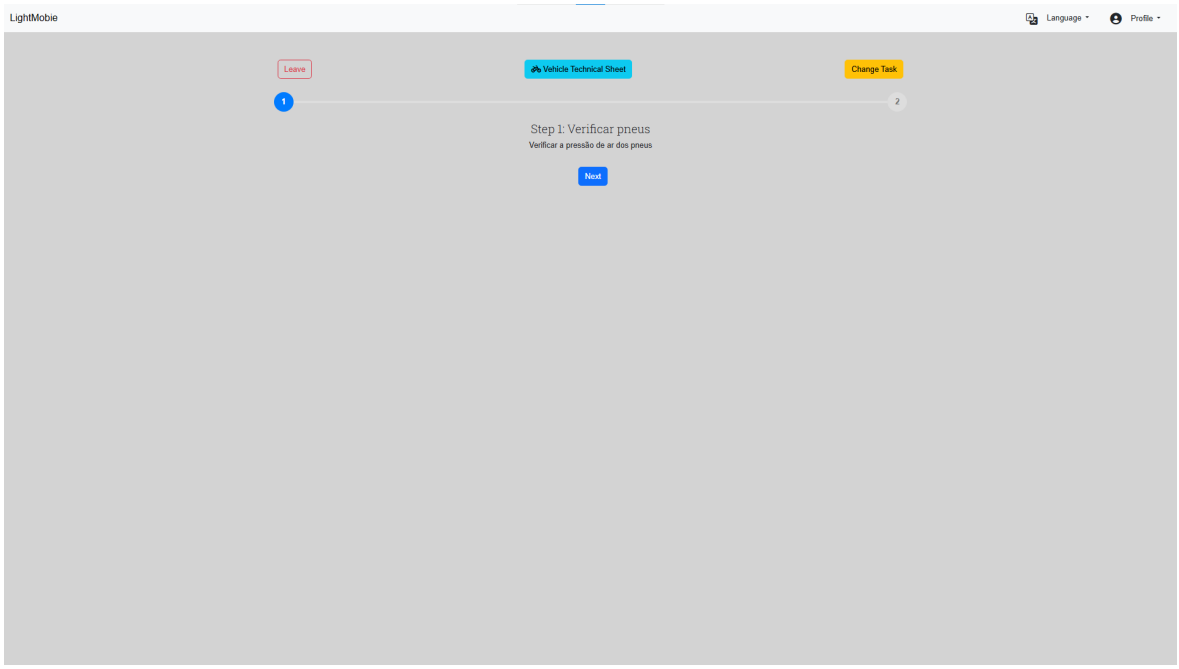


Figura A.10: Change task modal.

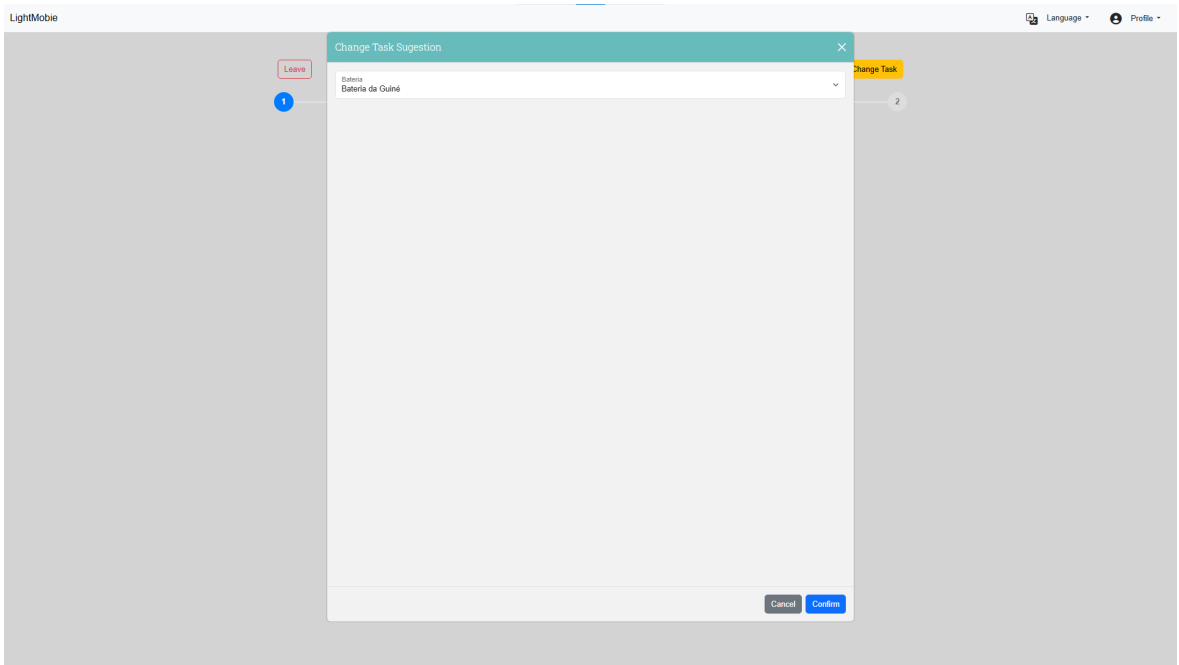


Figura A.11: Mechanic task last step.

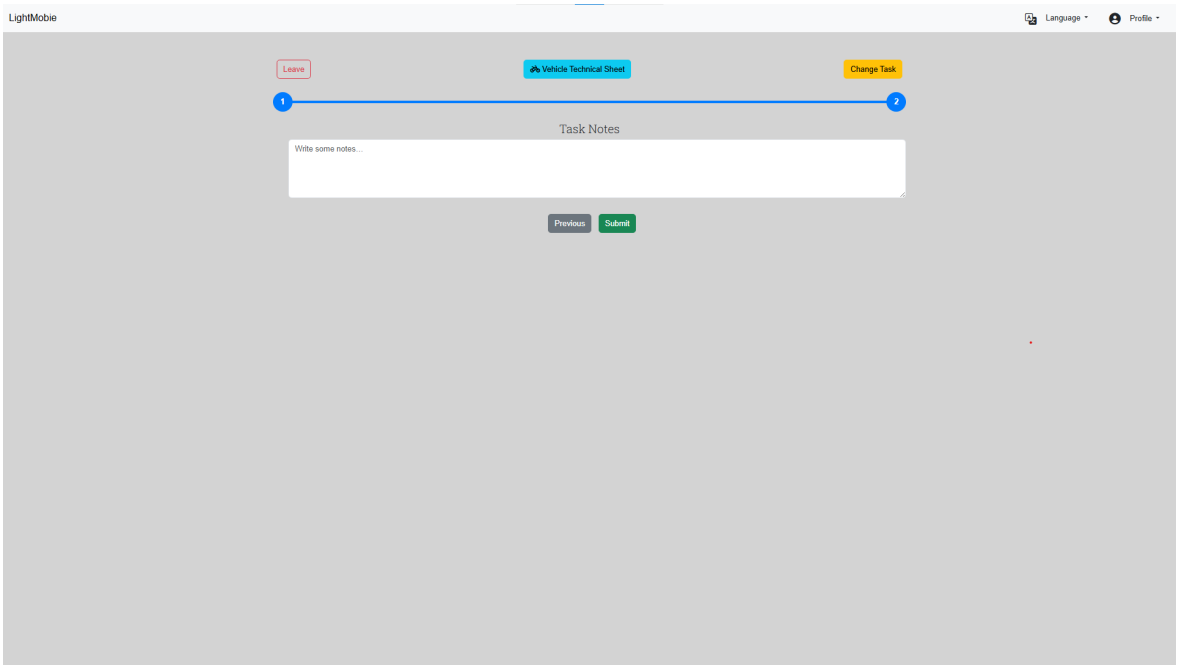


Figura A.12: Mechanic select part modal.

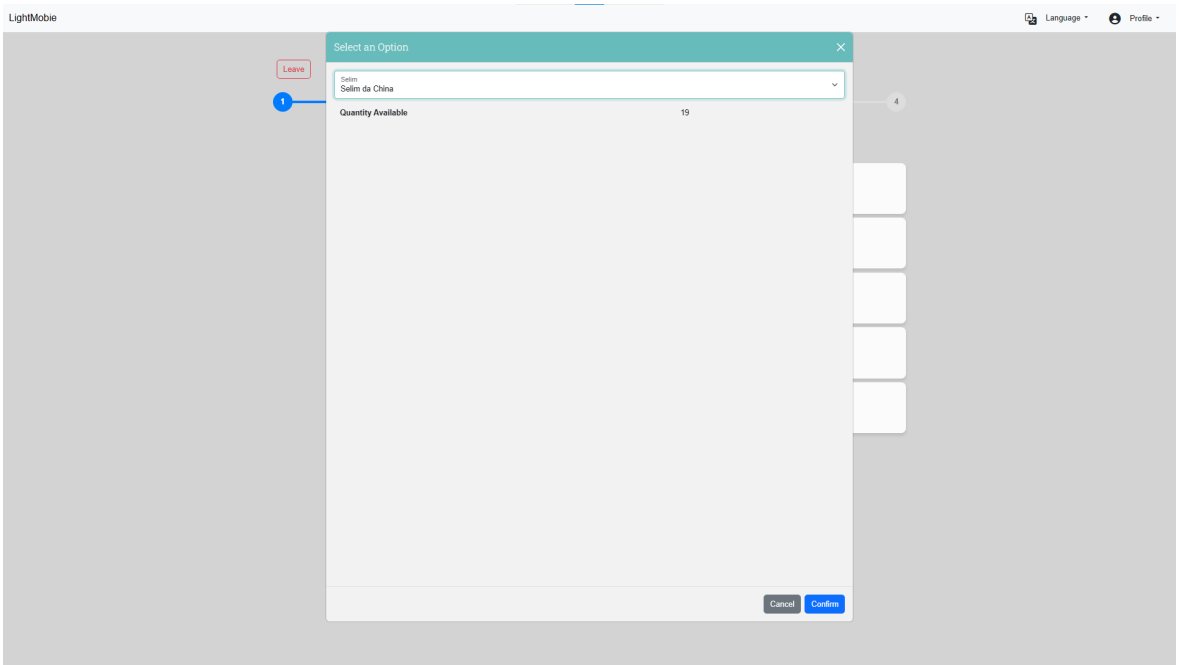


Figura A.13: Mechanic Evaluation task with part selected.

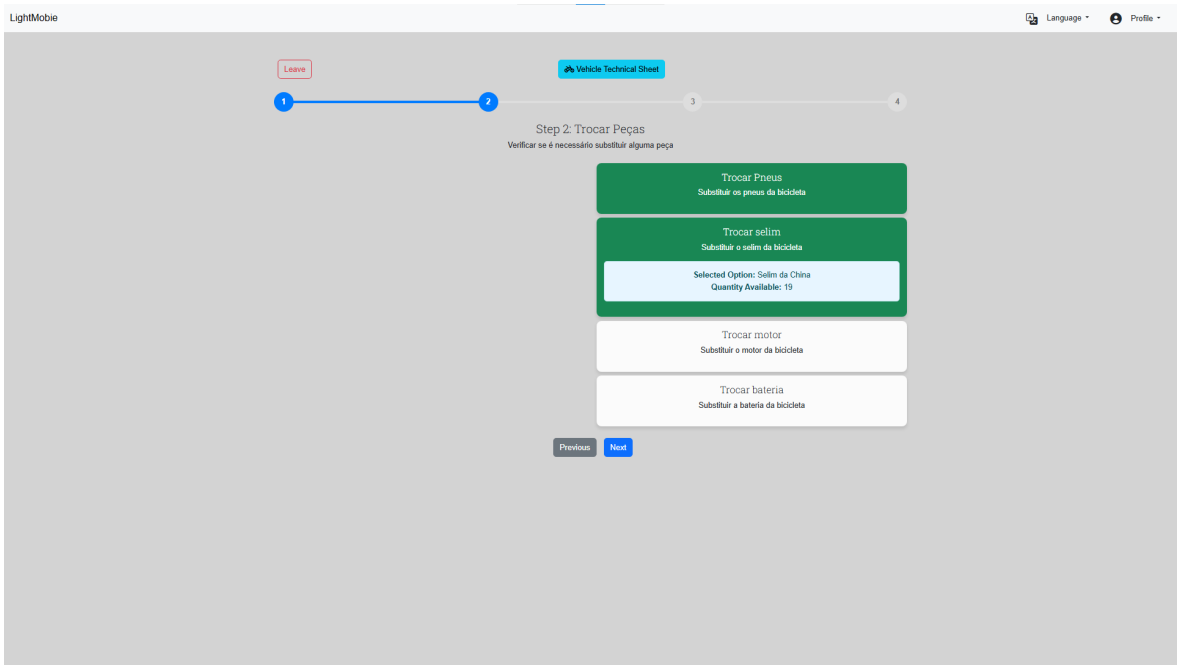


Figura A.14: Mechanic Evaluation last step.

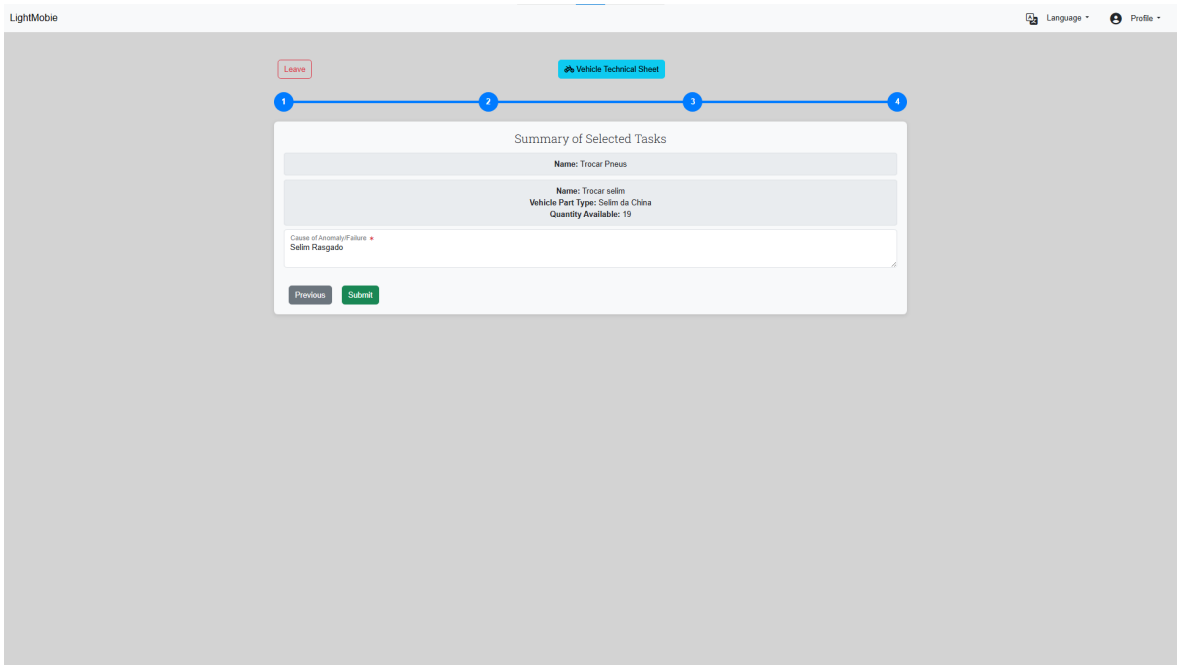


Figura A.15: Inventory editor.

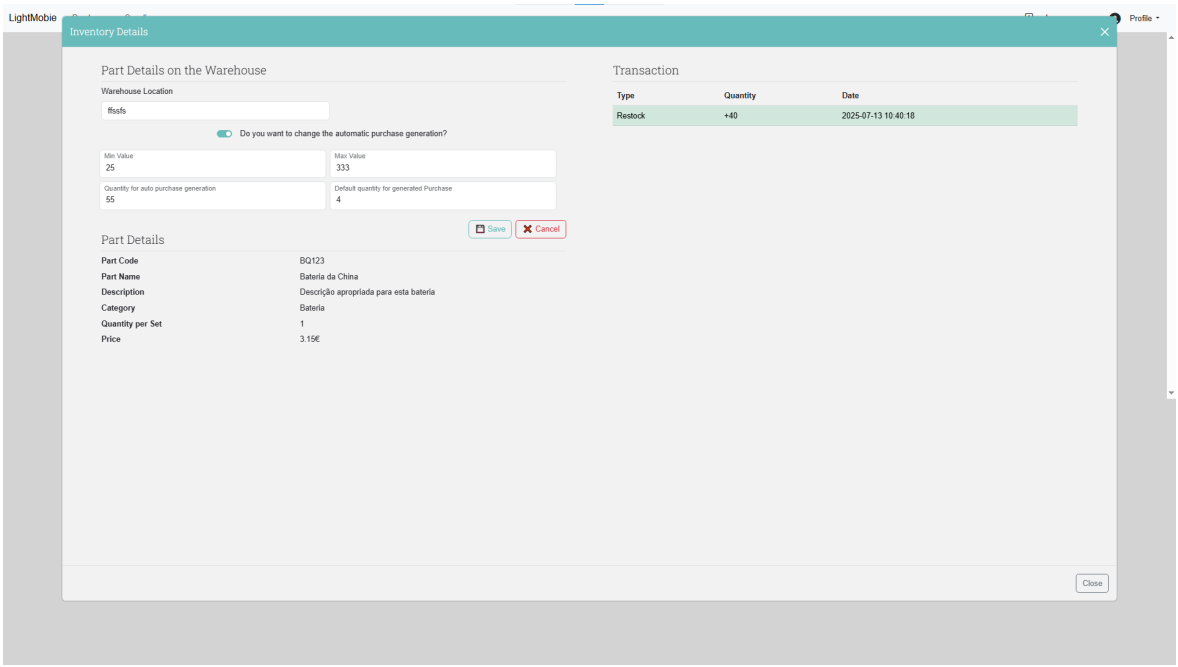


Figura A.16: Assgined purchase details.

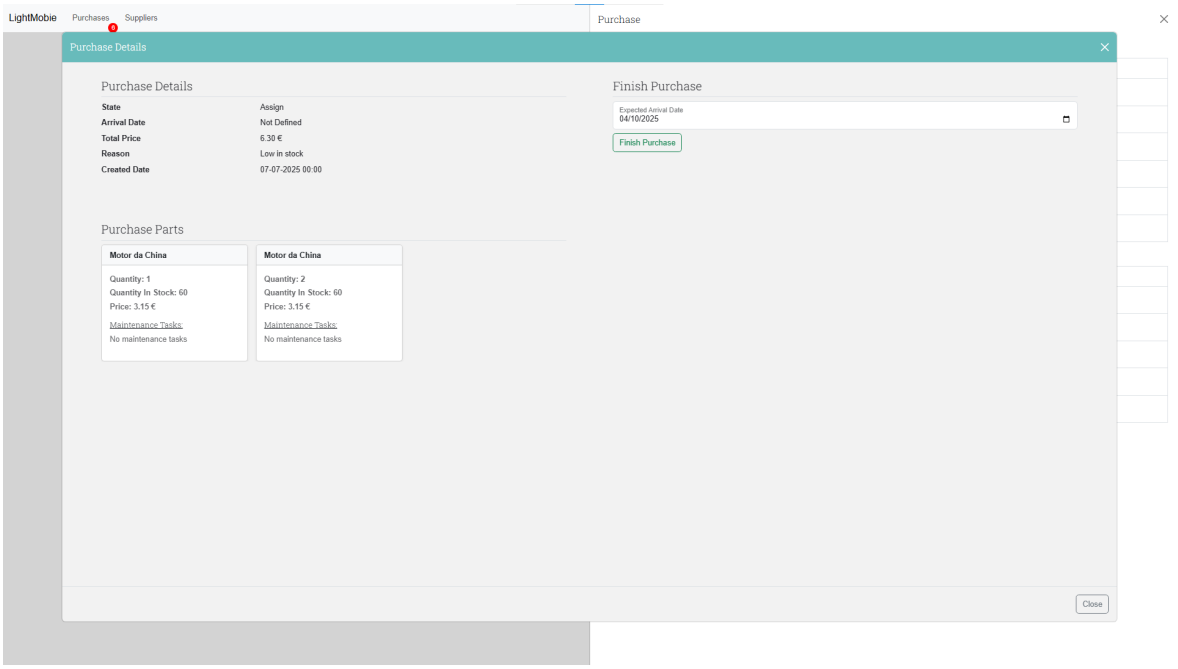


Figura A.17: Waiting delivery purchase details.

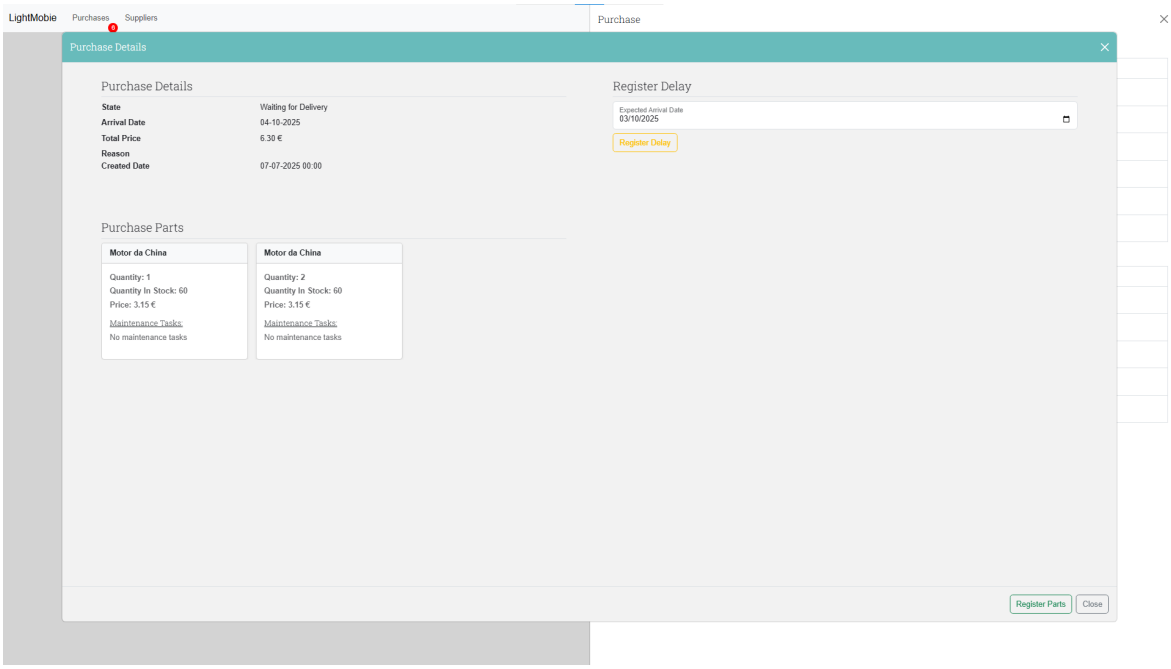


Figura A.18: Delivered purchase details.

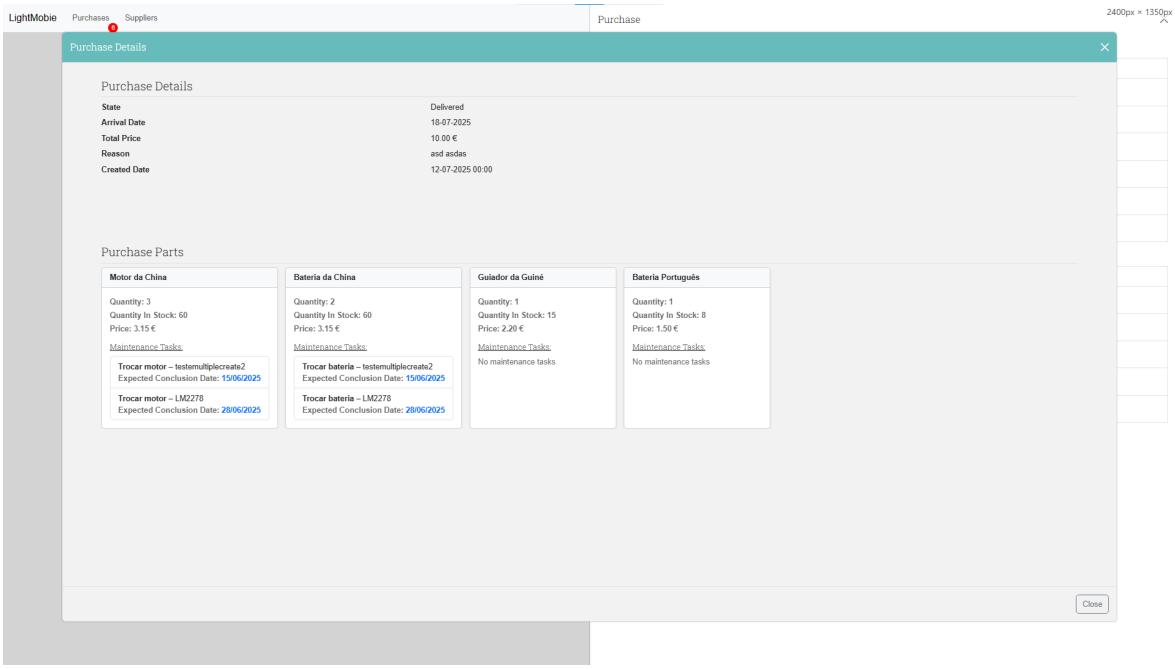


Figura A.19: Supplier details.

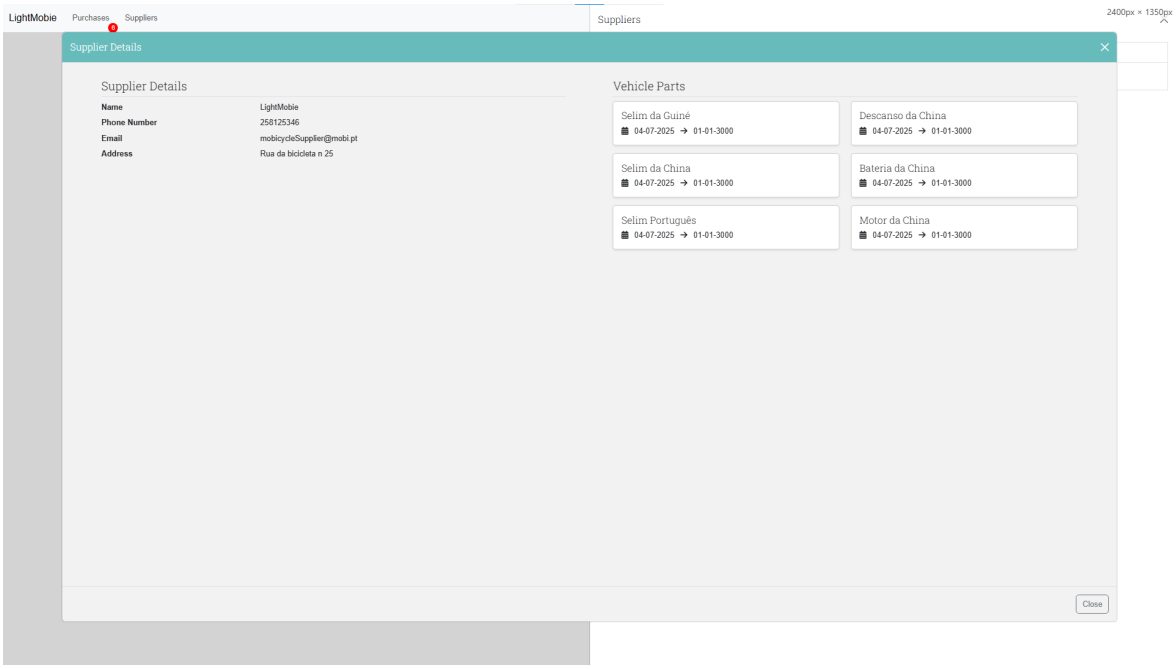


Figura A.20: Assign task to a mechanic.

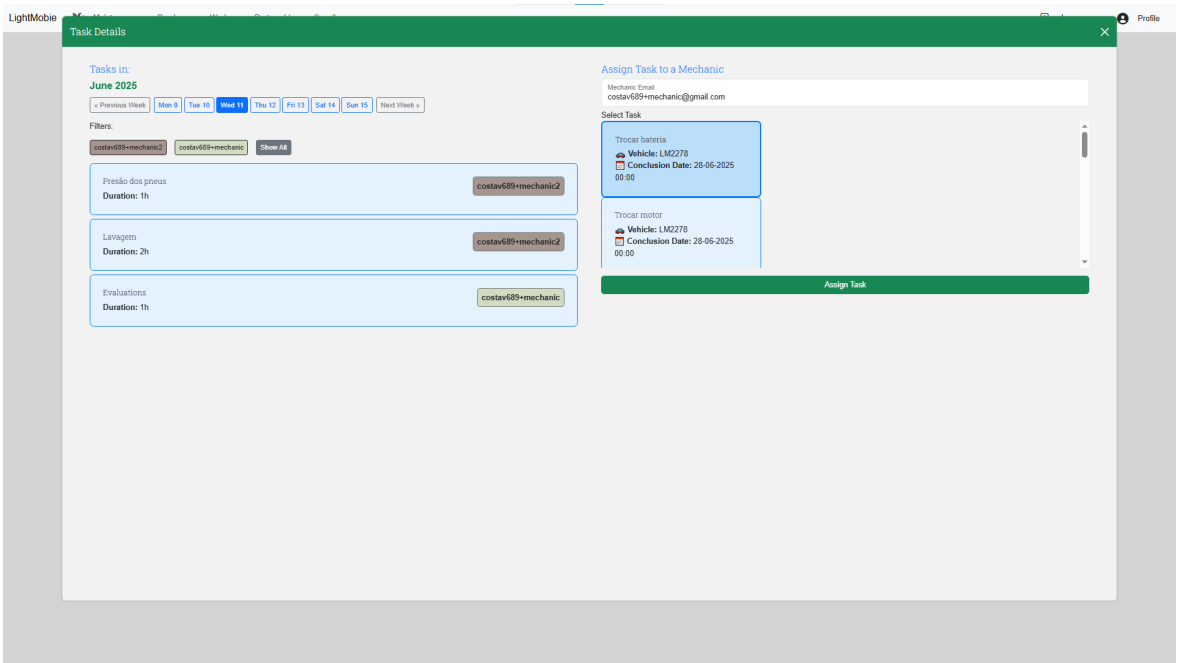


Figura A.21: Active maintenance list.

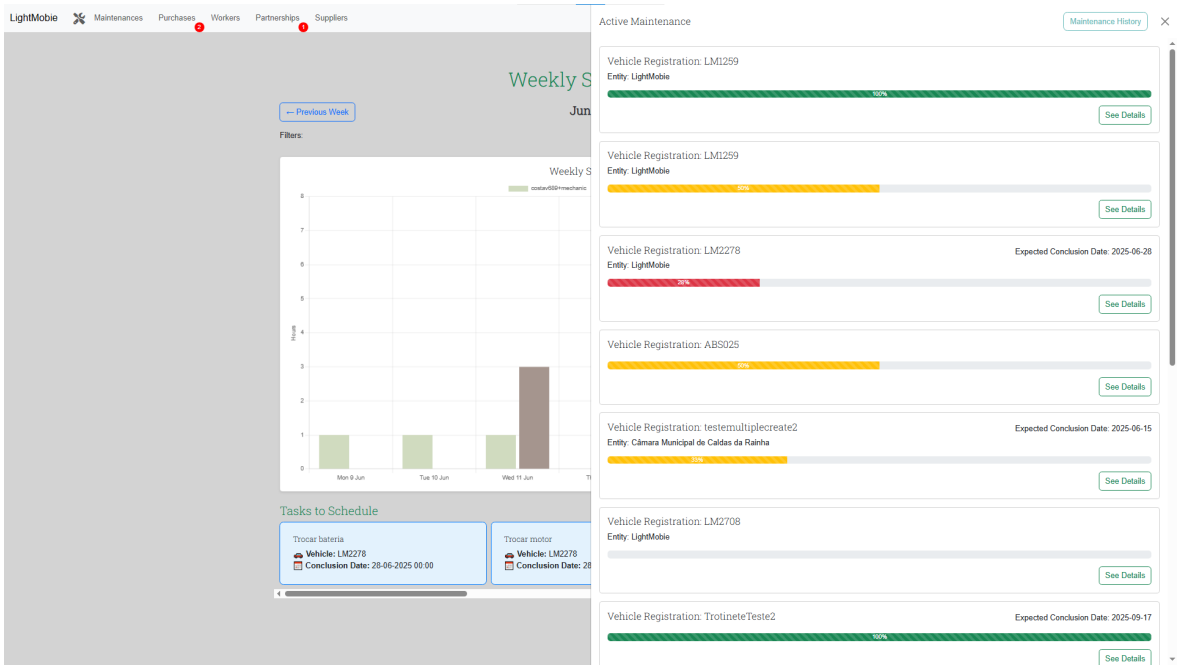


Figura A.22: Maintenance details information tab.

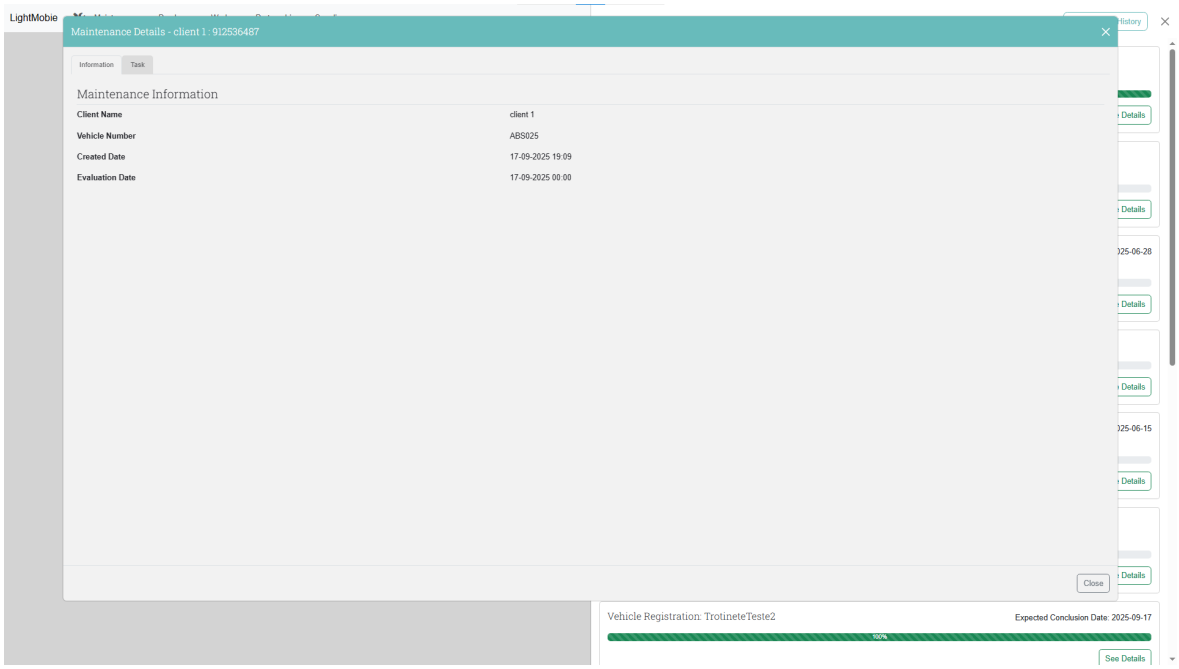


Figura A.23: Maintenance details tasks tab.

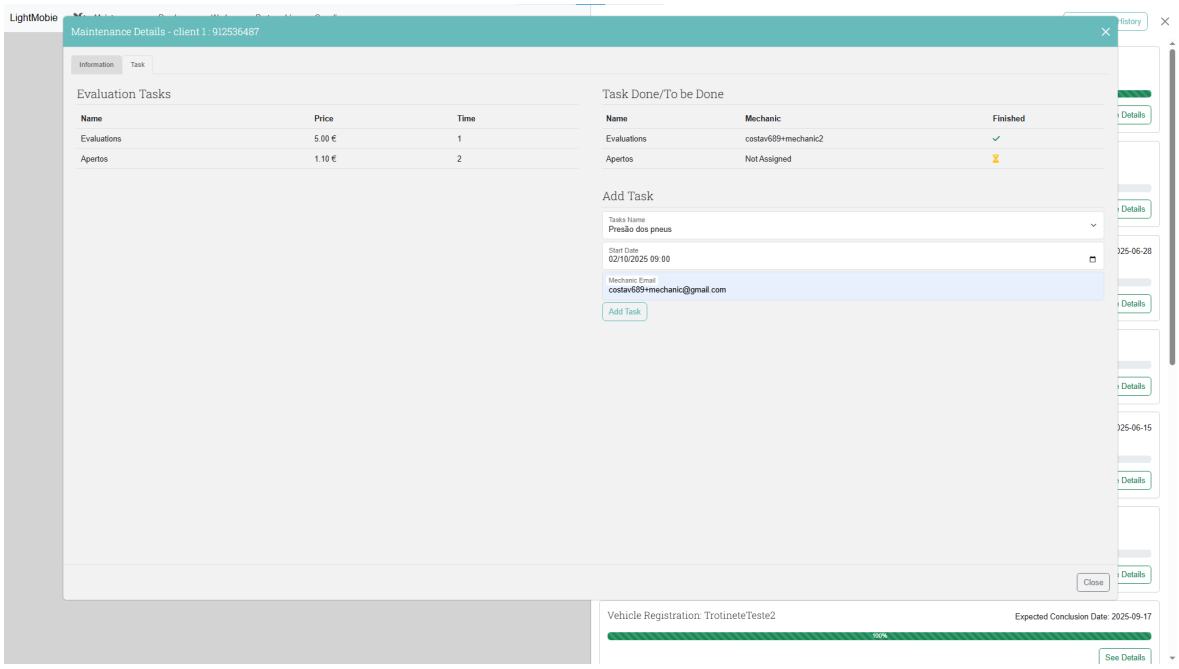


Figura A.24: Report example.

Maintenance Evaluation Report

Client Email: costav689+Client4@gmail.com
Dealership Name: LightMobie
Vehicle Registration: ABS025
Created Date: 2025-09-18
Evaluation Date: 2025-09-18
Cause of Anomaly/Failure: sadsad
Client Notes:
Expected Conclusion Date: 2025-09-18
Actual Conclusion Date: 2025-09-18
Deliver Date: 2025-09-18
Expected Budget: $1.20
Actual Budget: $6.20
Client Rating: 2.00
Expected Working Hours: 2.00
Working Hours: 0.14

Maintenance Tasks

Task Name	Description	Vehicle Part	Quantity	Hours	Price	Status
Verificar óleo	Verificar o nível de óleo da bicicleta	-	-	2.00	$1.20	Concluded
Evaluations	Do Vehicle Evaluation	-	1	1.00	$5.00	Concluded

Maintenance Changes

No maintenance changes available.

Figura A.25: Purchase request details.

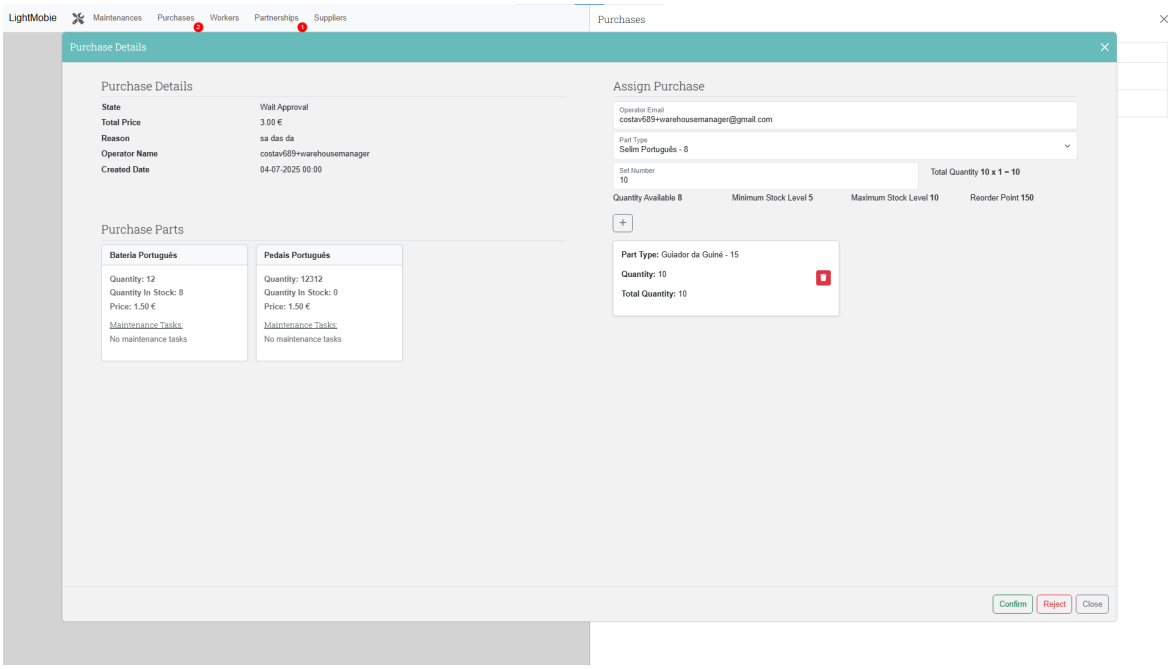


Figura A.26: Supplier list.

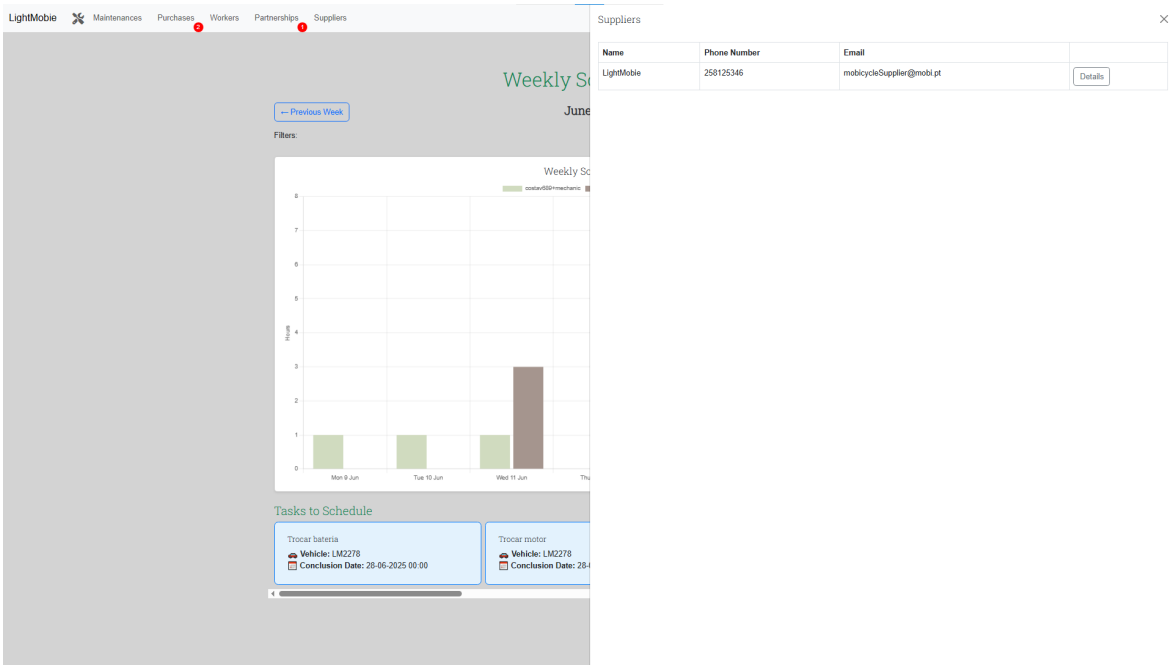


Figura A.27: Supplier Details.

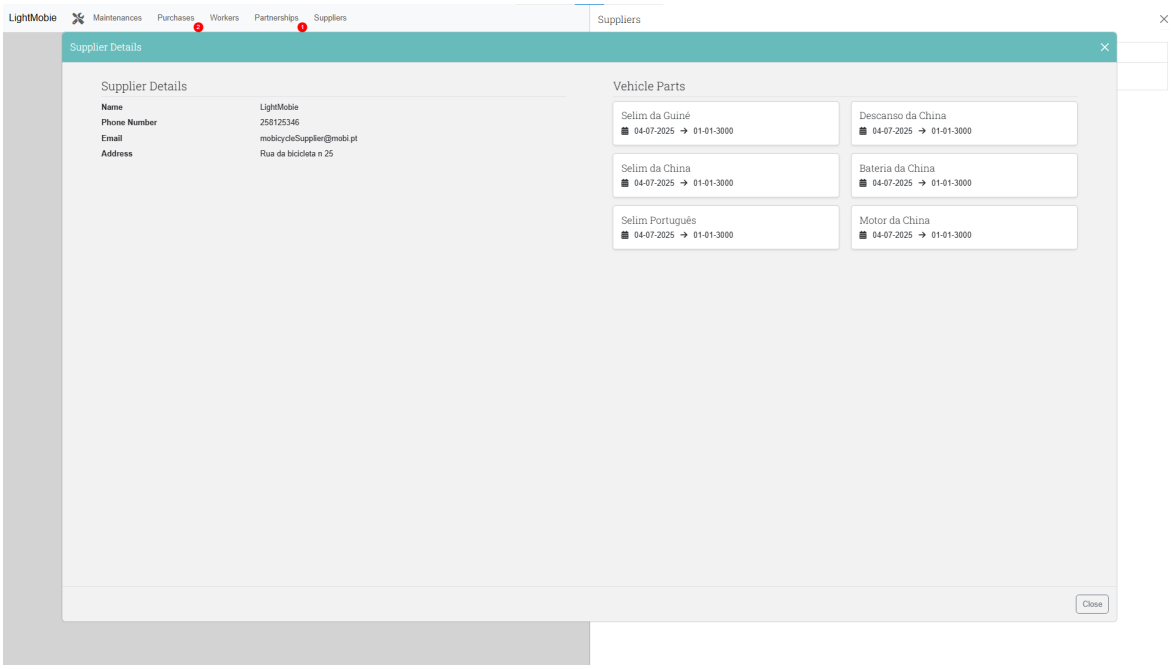


Figura A.28: Worker details.

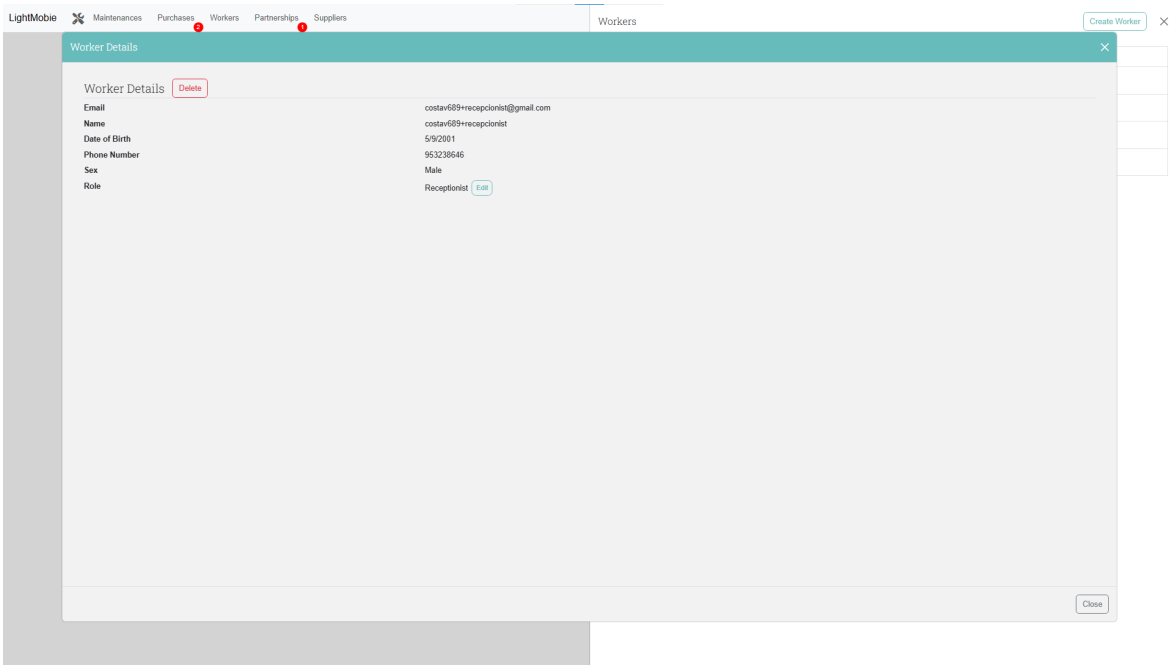


Figura A.29: Create worker.

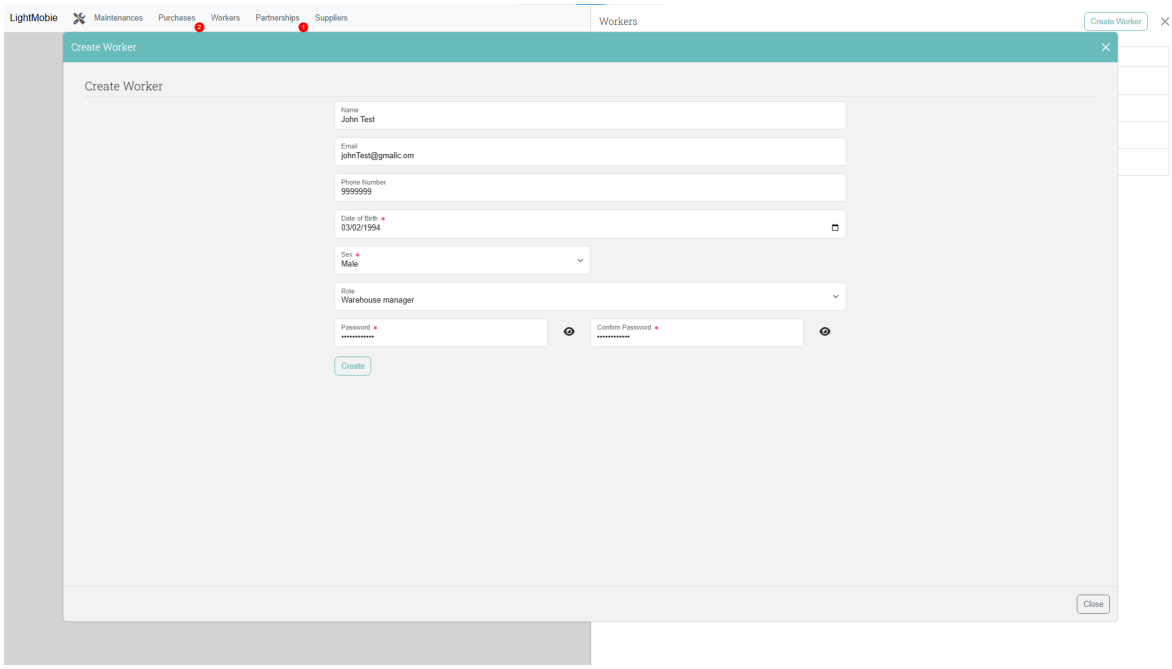


Figura A.30: Client home page with no active maintenance.

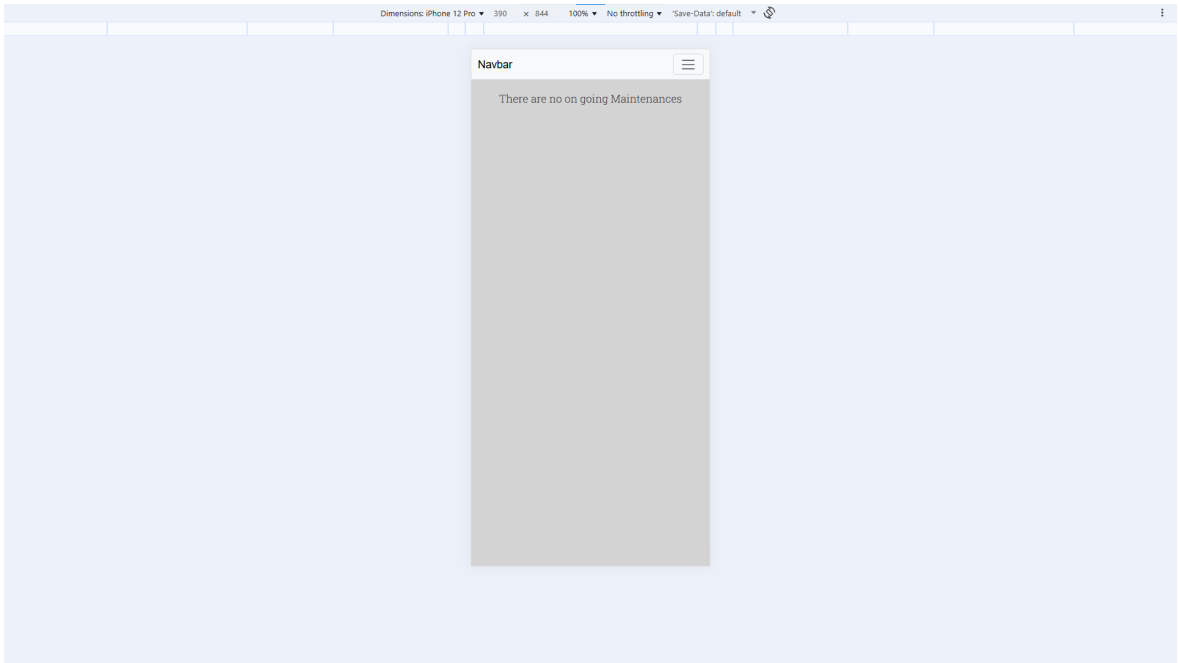


Figura A.31: Client Home page when he schedules a maintenance with the rececionist.

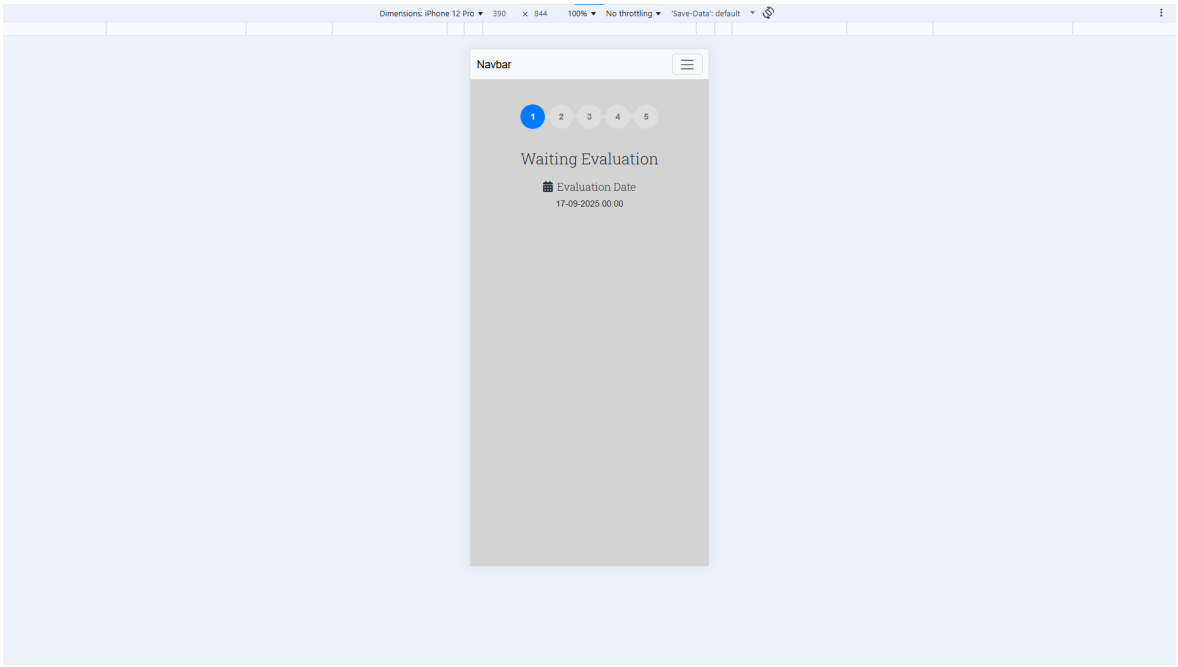


Figura A.32: Client Home page when the maintenance is approved.

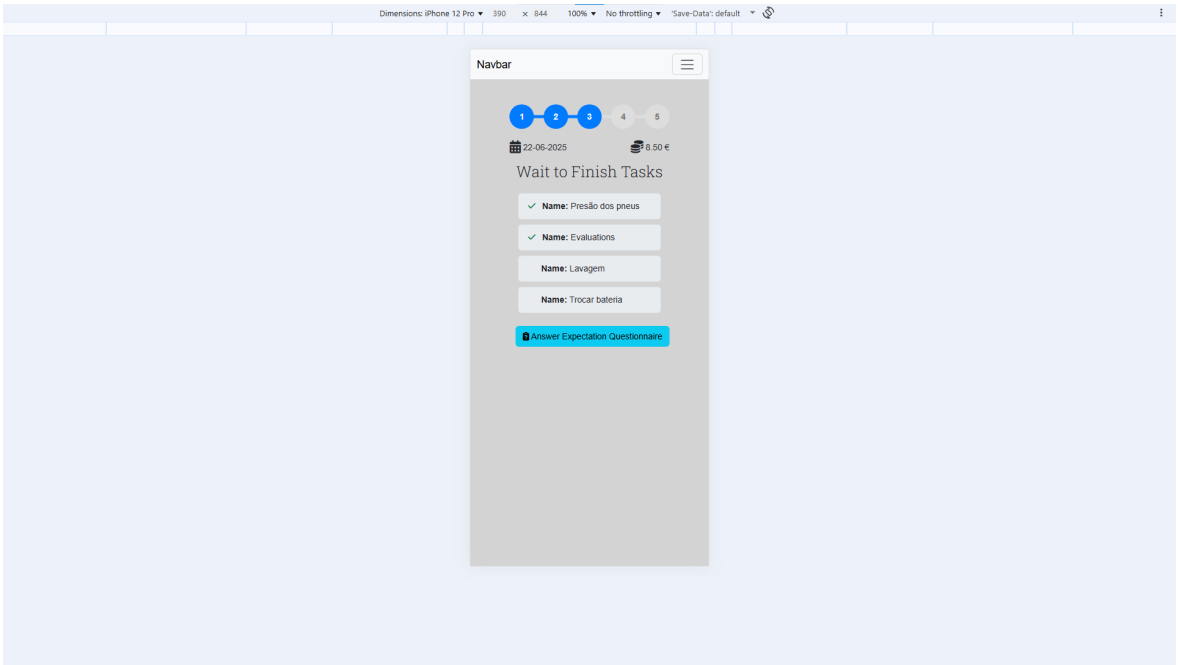


Figura A.33: Client Home page when the maintenance tasks are completed and the client can go take the vehicle.

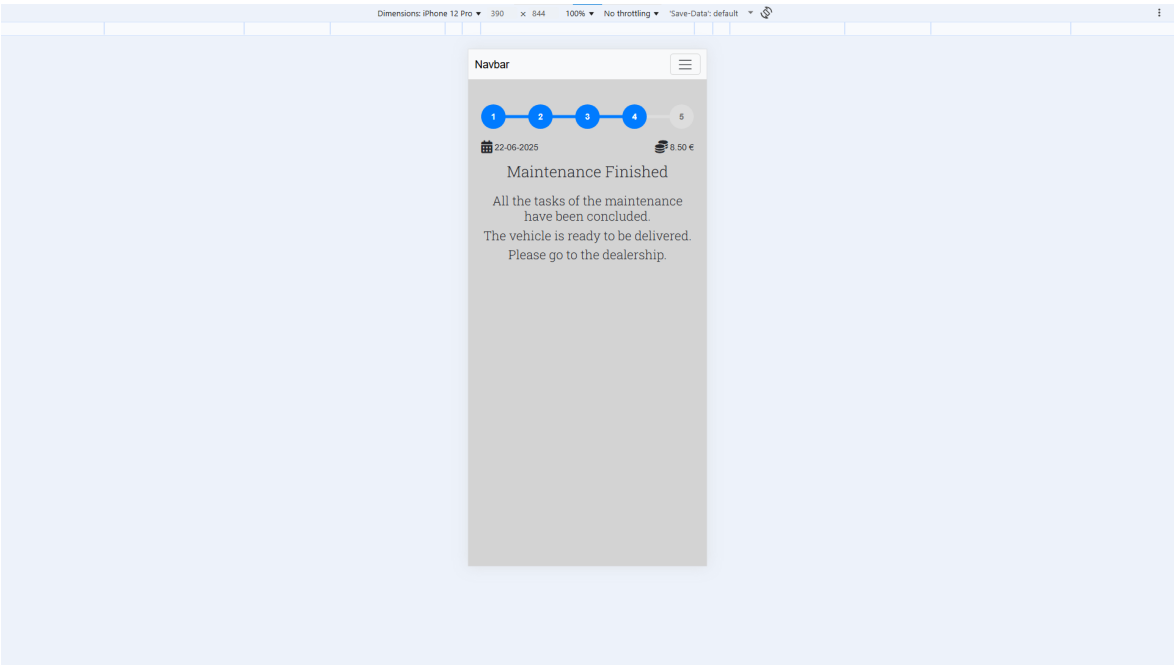


Figura A.34: Details of a maintenance example the tab of information.

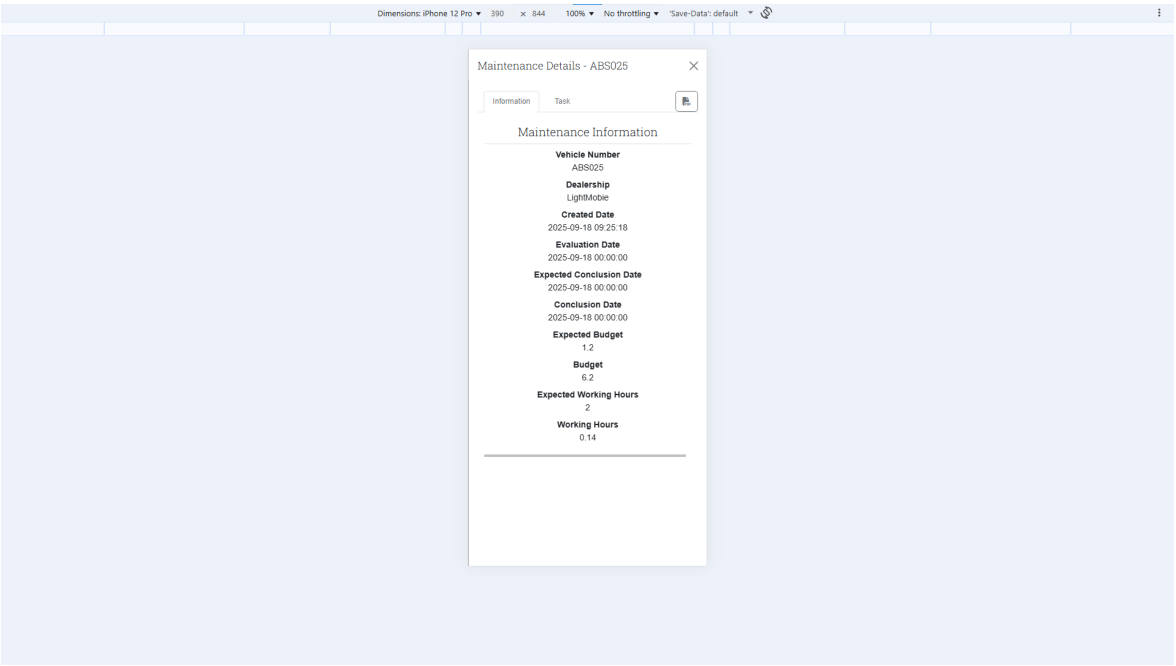


Figura A.35: Details of a maintenance example the list of task.

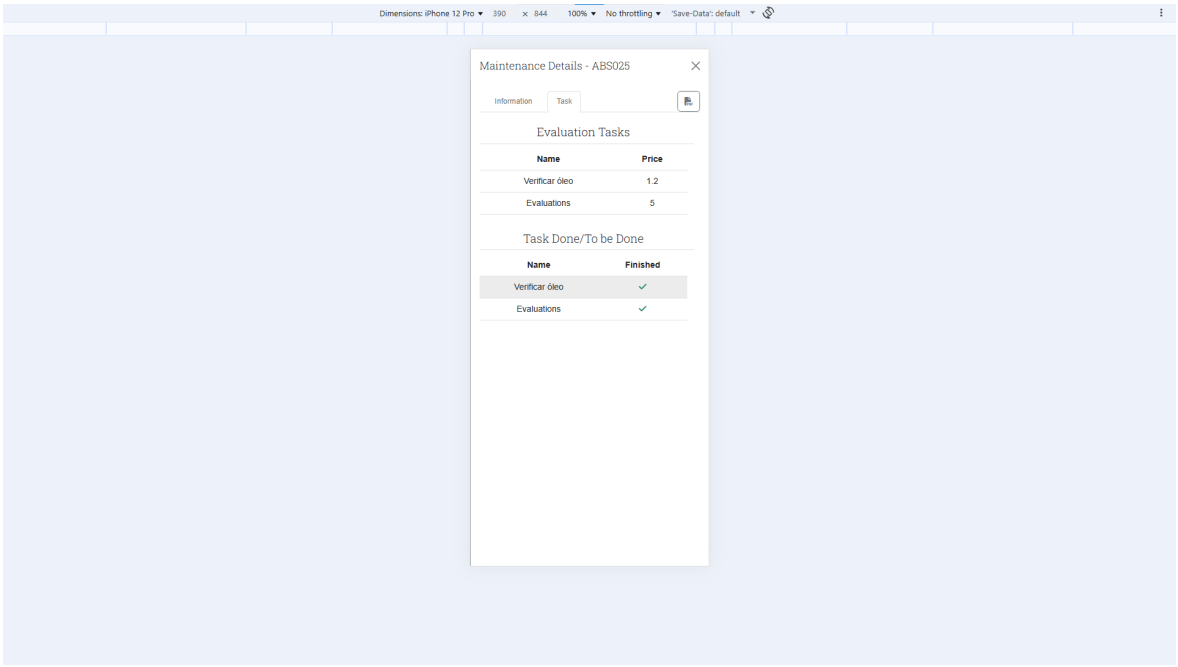


Figura A.36: Parts type details.

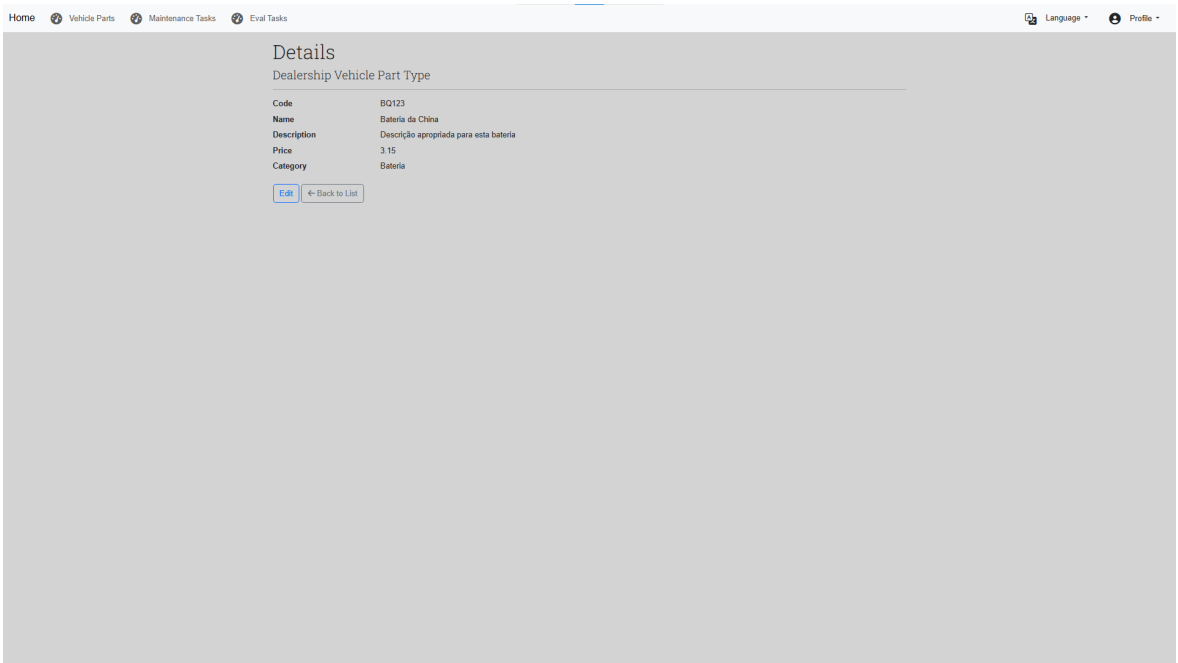


Figura A.37: Parts type edit.

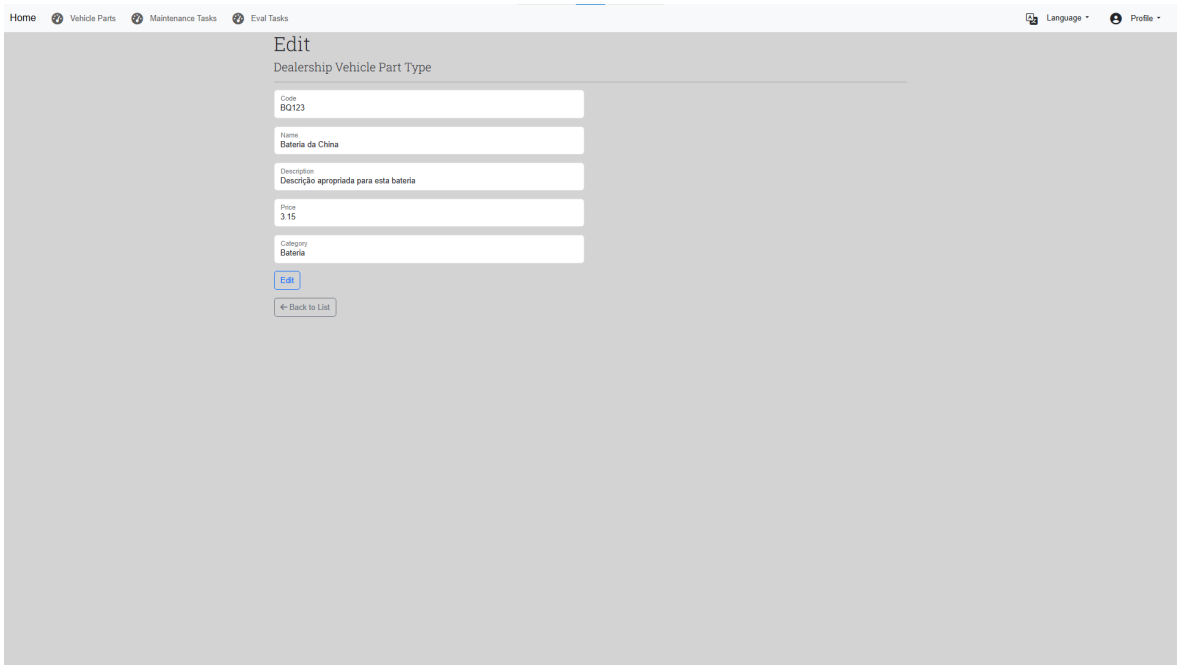


Figura A.38: Parts type delete.

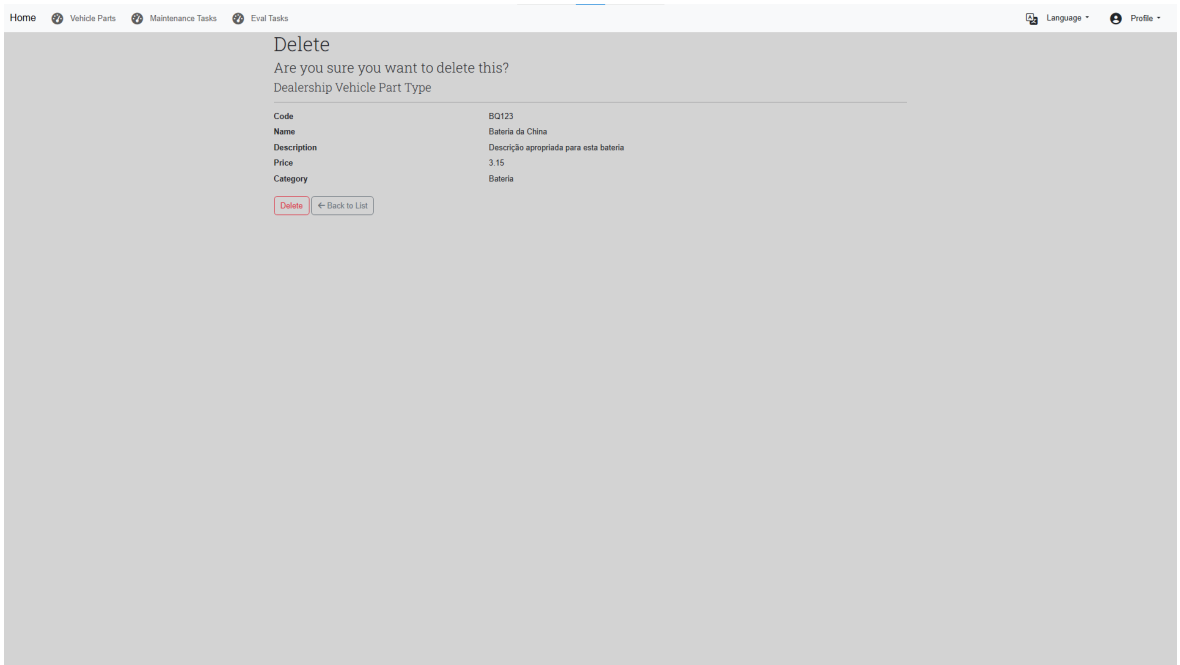


Figura A.39: Parts type create.

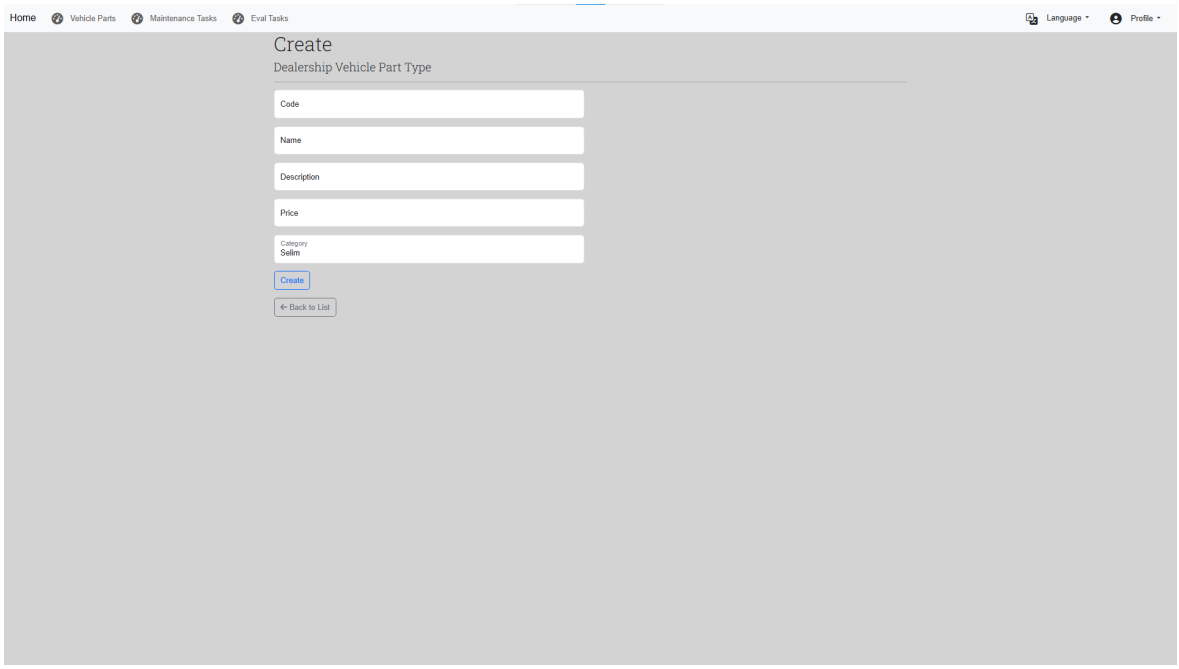


Figura A.40: Task type list index.

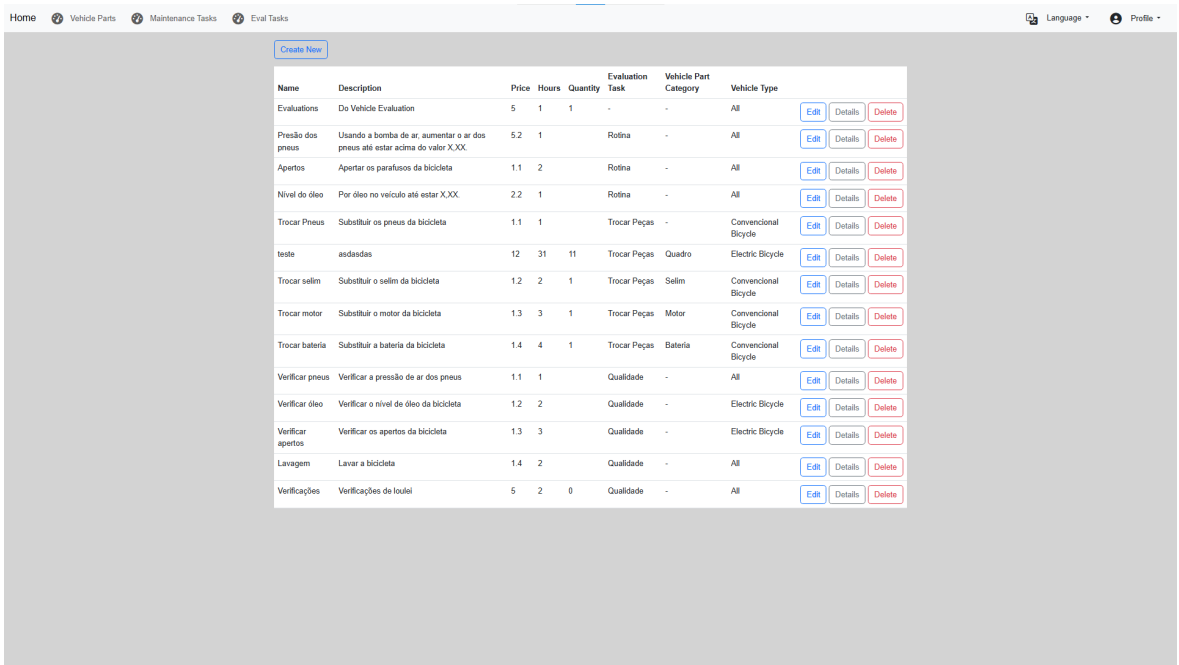


Figura A.41: Eval task list index.

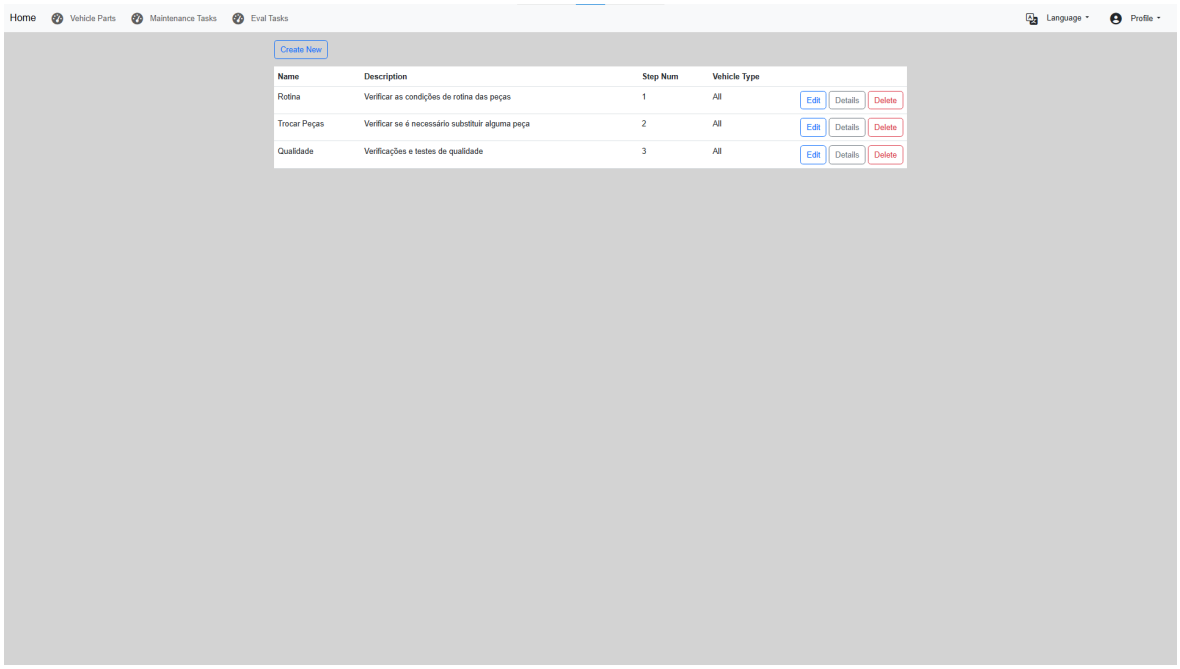


Figura A.42: Eval task edit.

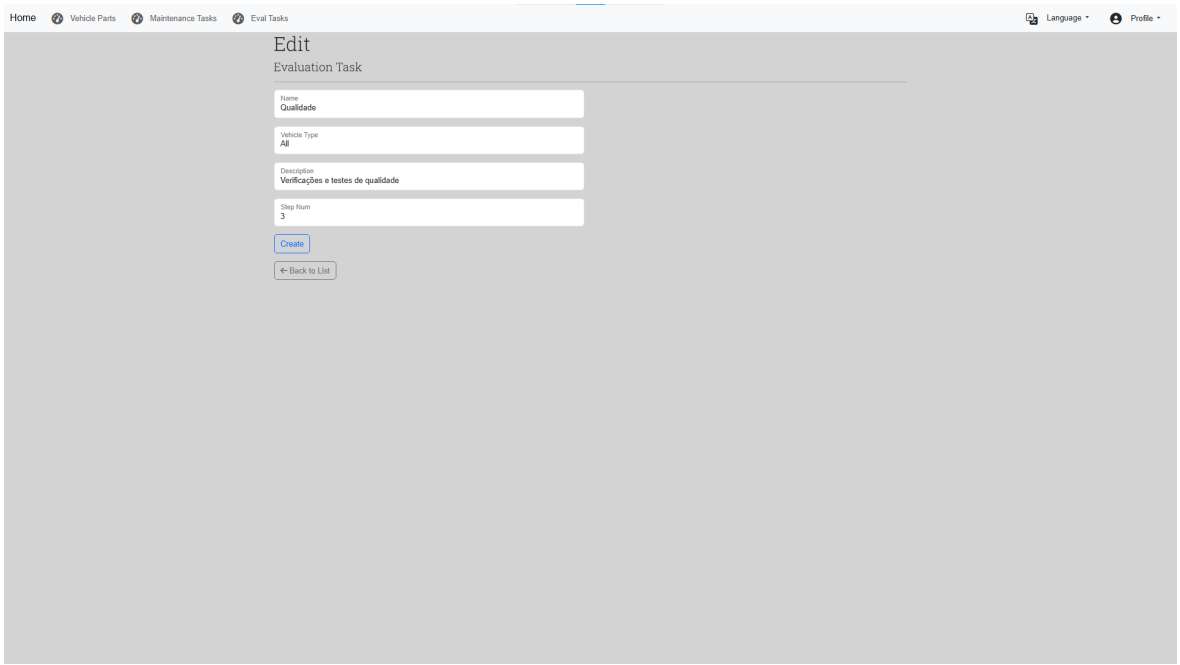


Figura A.43: Task type edit.

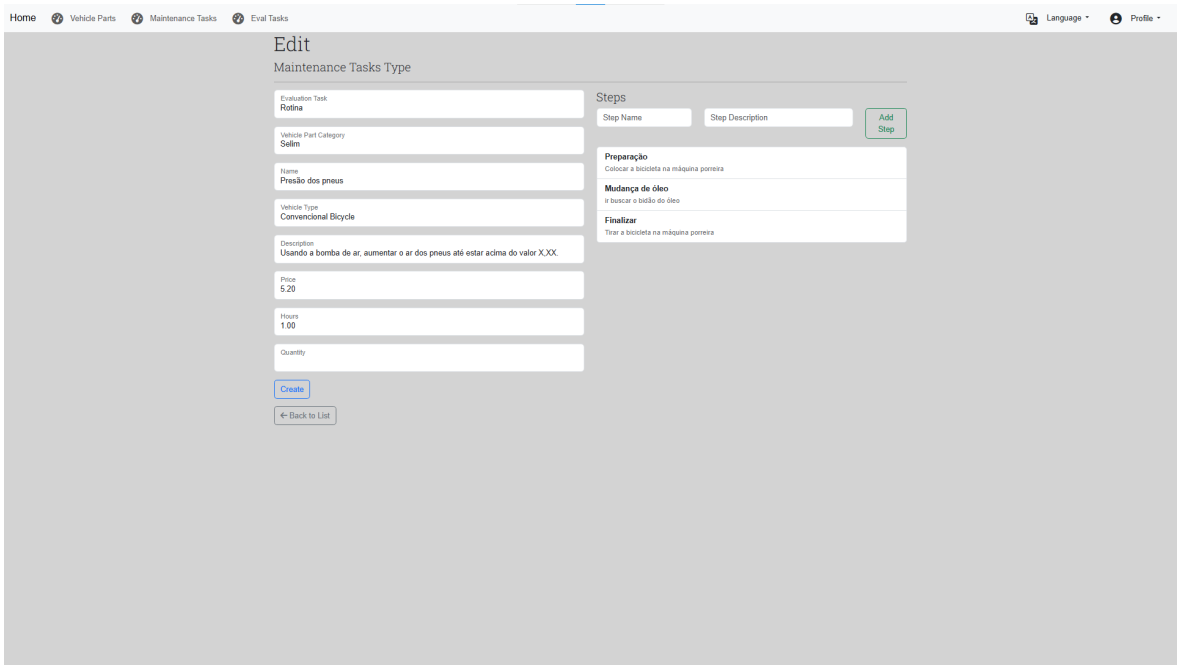


Figura A.44: Task type details.

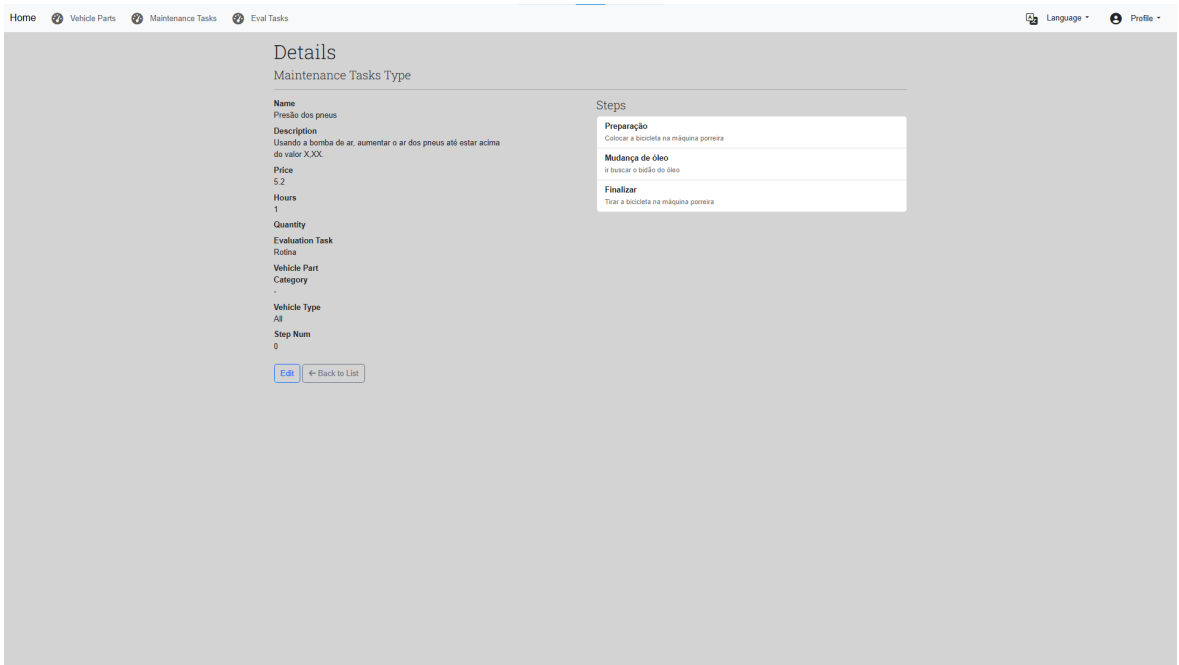


Figura A.45: Task type delete.

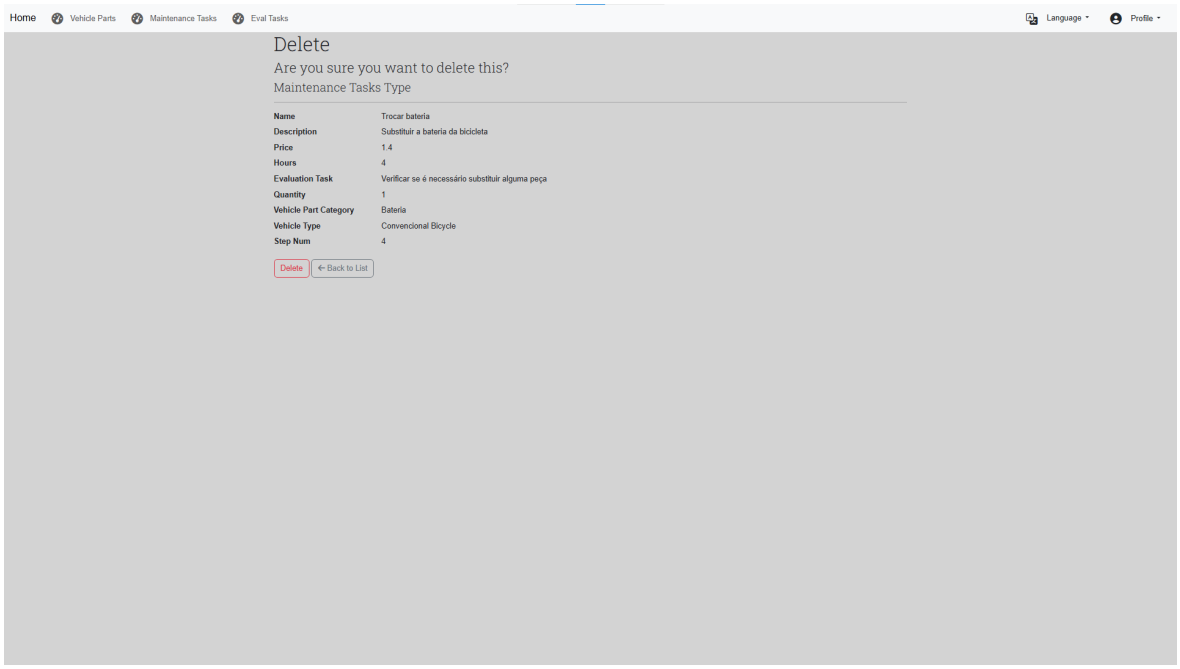


Figura A.46: Task type create.

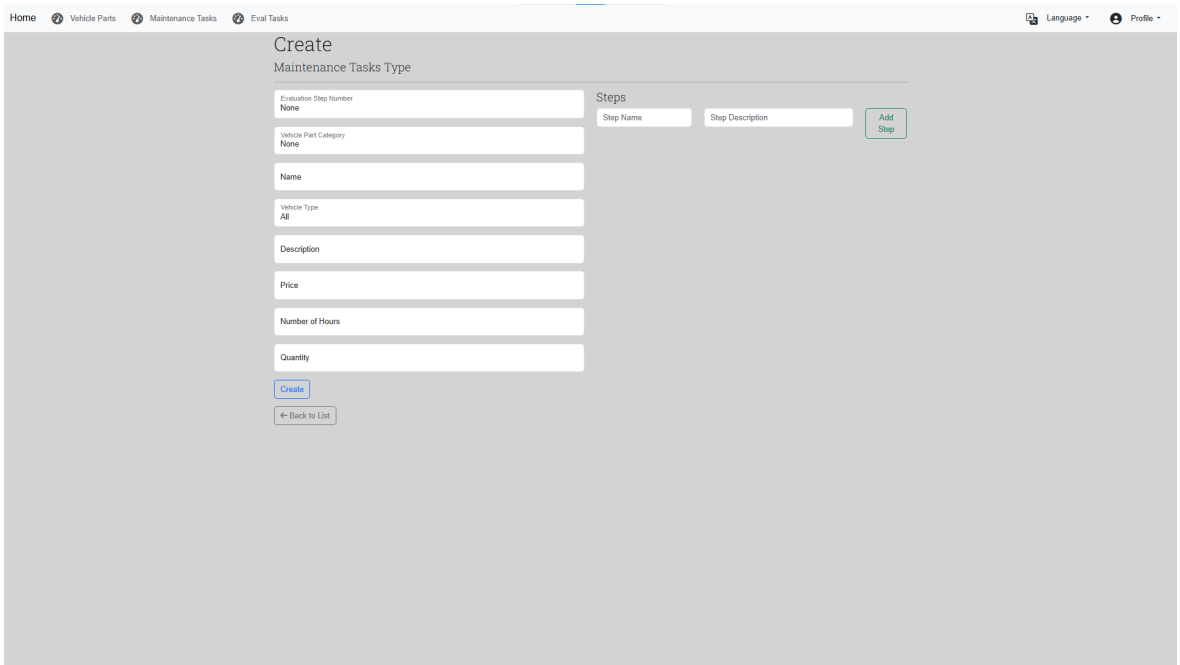


Figura A.47: Eval task create.

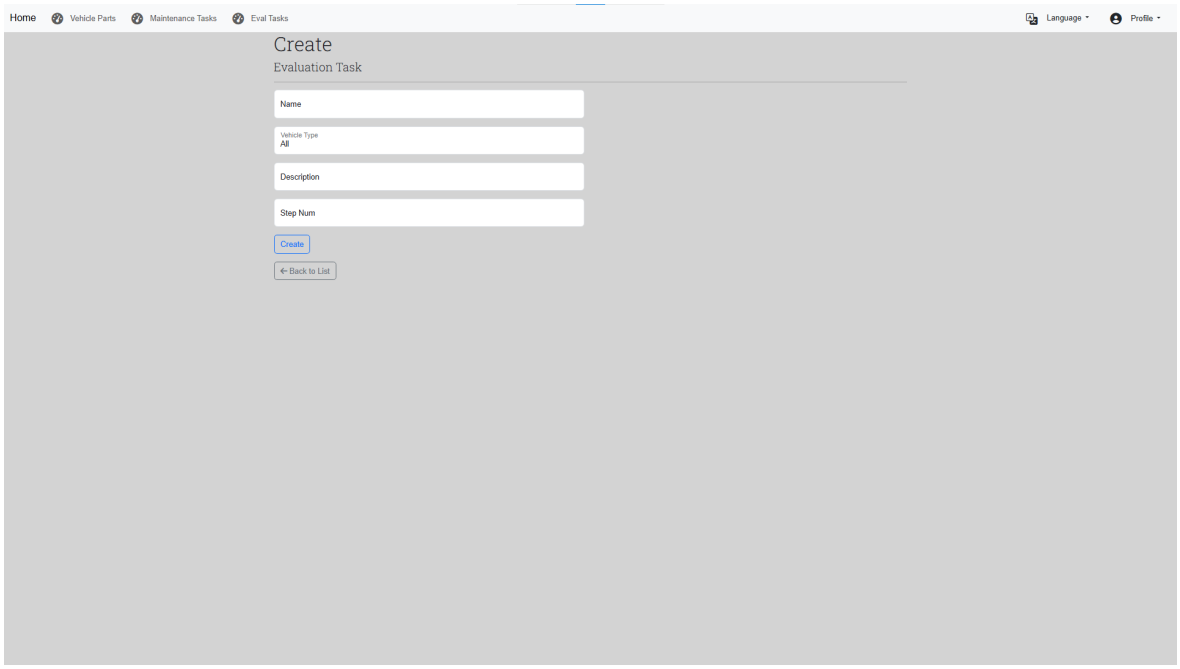


Figura A.48: Eval task delete.

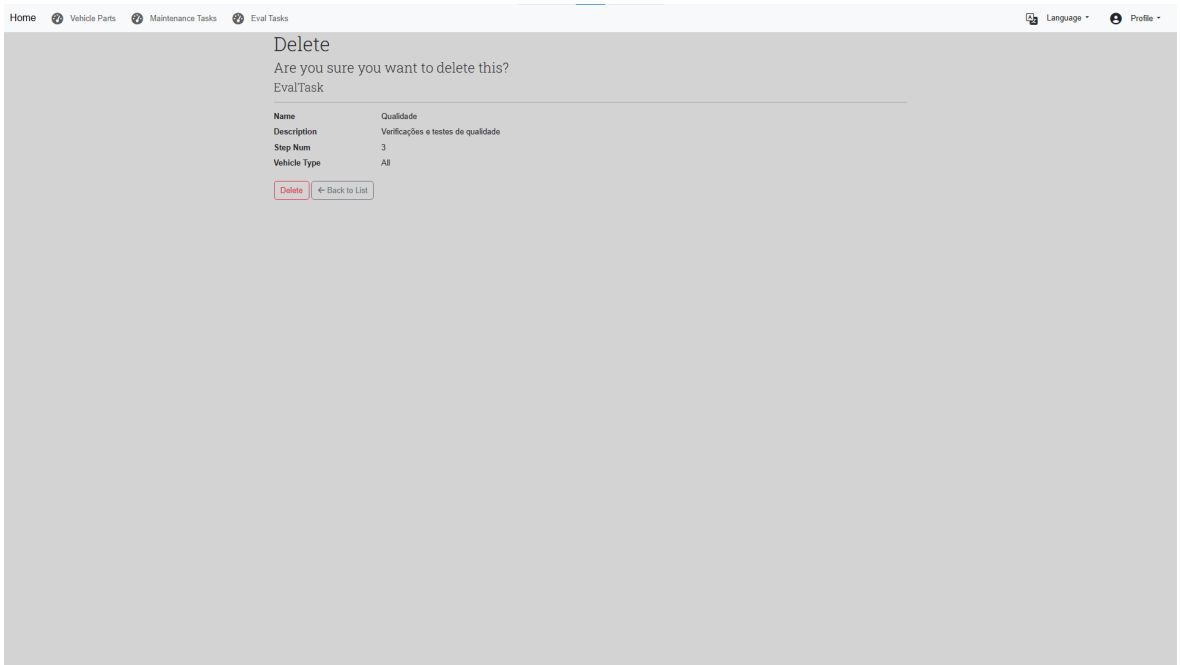


Figura A.49: Eval task details.

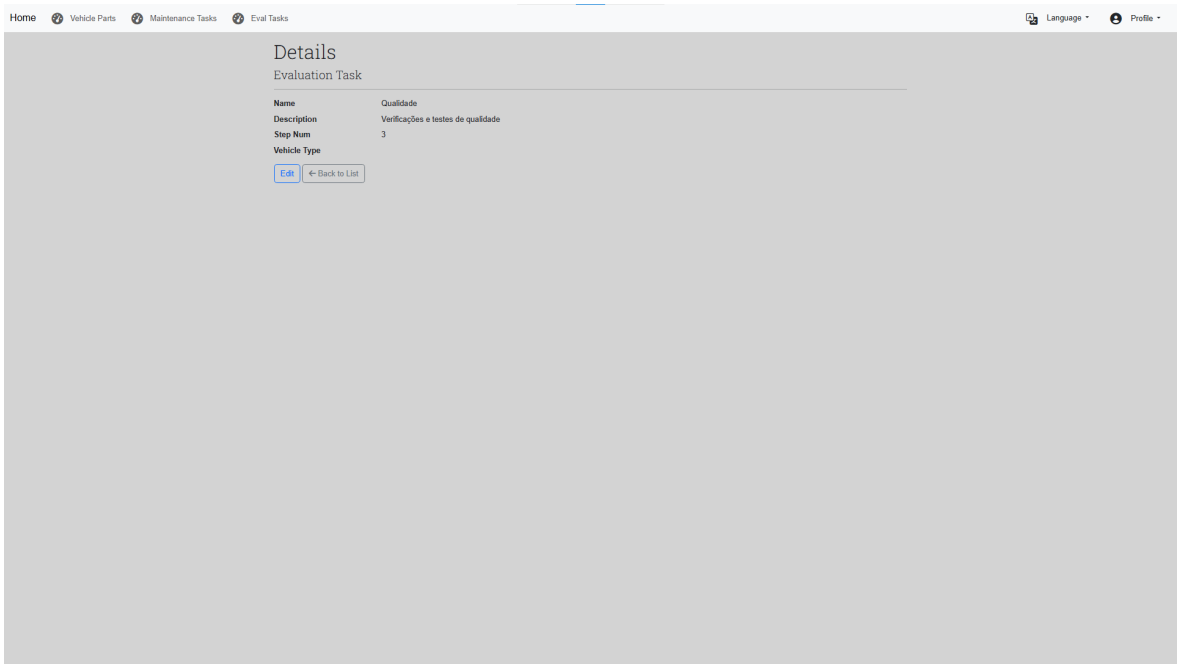


Figura A.50: User tests tasks

Testes de Utilização do Rececionista

Teste 1 - Marcar uma manutenção

Cenário: Um cliente da entidade "**lightmobie**" liga e diz que quer marcar uma manutenção para **Hoje** às **14h00**, pois o veículo "**LM1259**" tem um pneu furado.
Tarefa: Marcar uma manutenção com essa data para a entidade "**LightMobie**", veículo "**LM1259**" e com a tarefa "**Trocar Pneus**".

Teste 2 - Gráfico de tarefas

Cenário: O rececionista quer marcar uma manutenção para o dia mais livre na semana de "**9 de junho**" a "**15 de junho**"
Pergunta: Nesta semana qual é o dia mais dificil para marcar uma avaliação de manutenção?

Pergunta: Nesta semana qual seria um bom dia para marcar uma manutenção?

Pergunta: Quantas horas de trabalho estão atribuidas ao utilizador "costav689+mechanic" no dia 14 de junho?

Pergunta: No dia 14 de junho quais são as tarefas do utilizador "costav689+mechanic" ?

Teste 3 - Recolher informações sobre manutenção

Cenário: O utilizador do veículo "**LM2278**" quer saber quantas tarefas ainda faltam ser realizadas
Pergunta: Na manutenção do veículo "**LM2278**" quantas tarefas faltam ser realizadas?

Pergunta: Qual é a data de conclusão da manutenção

Pergunta: Qual é o preço expectavel da manutenção?

Teste 4 - Alterações de manutenção

Cenário: A manutenção do veículo "**LM2278**" sofreu várias alterações.
Pergunta: Em que tarefas aconteceu essa alteração? E em que dias?

Figura A.51: Aplication questionaire

Usability tests of the mechanic view

After performing the tasks of the mechanic, please answer the following question by rating the statement from strongly disagree to strongly agree.

* Indicates required question

1. I think i would like to use this system frequently. *

   Mark only one oval.

   - ◯ strongly disagree
   - ◯ disagree
   - ◯ Neutral
   - ◯ agree
   - ◯ strongly agree

2. I found the system unnecessarily complex. *

   Mark only one oval.

   - ◯ strongly disagree
   - ◯ disagree
   - ◯ Neutral
   - ◯ agree
   - ◯ strongly agree

3. I thought the system was easy to use. *

   Mark only one oval.

   - ◯ strongly disagree
   - ◯ disagree
   - ◯ Neutral
   - ◯ agree
   - ◯ strongly agree