

Technical Report - **Product specification**

Car Management Application

Course: IES - Introdução à Engenharia de Software

Date: Aveiro, 07, October, 2024

Students: 114646: Francisco Albergaria
113682: Gabriel Santos
113207: Guilherme Amaral
114514: João Gaspar

Project abstract: This project aims to create a Car Management App that uses information retrieved from the ECU (Electronics Control Unit) of a car to display real-time warnings and details about it.

Table of contents:

[1 Introduction](#)

[2 Product concept](#)

[Vision statement](#)

[Personas](#)

[Main scenarios](#)

[3 Architecture notebook](#)

[Key requirements and constraints](#)

[Architctural view](#)

[Module interactions](#)

[4 Information perspective](#)

[5 References and resources](#)

1 Introduction

In the ambit of Introduction to Software Engineering, we (assigned group 204) compromised ourselves to work as a team, develop code, solve tasks and deliver results according to a normal workplace flow expectation. As so we plan to learn from everything this class has to offer, and to follow the templates, ideas and apply the concepts learnt during the practical and theoretical classes into our project, in which we conceptualize and implement a solution to a Car Management Application.

Our group is constituted of four members, in which we all know each other and therefore already have developed a good team spirit, communicative and critical, meaning we won't hold back onto debating and finding solutions to possible problems during this class.

- Francisco Albergaria – Product Owner
- Gabriel Santos – DevOps
- Guilherme Amaral – Team Manager
- João Gaspar – Architect

Some of the goals in mind for this project, in relation to the scope of the IES course is to understand the organization of a software project, managed as an industrial process, select the best software architecture for a given problem/product and build a software system as a team, using a business framework, while using corporate solutions and tools for software development.

Contents to learn and apply: (Software engineering principles; Software process; Software architectures, Cloud models; Development environments; Spring framework and spring boot; Software certifications; Business and ethics).

2 Product concept

Vision statement

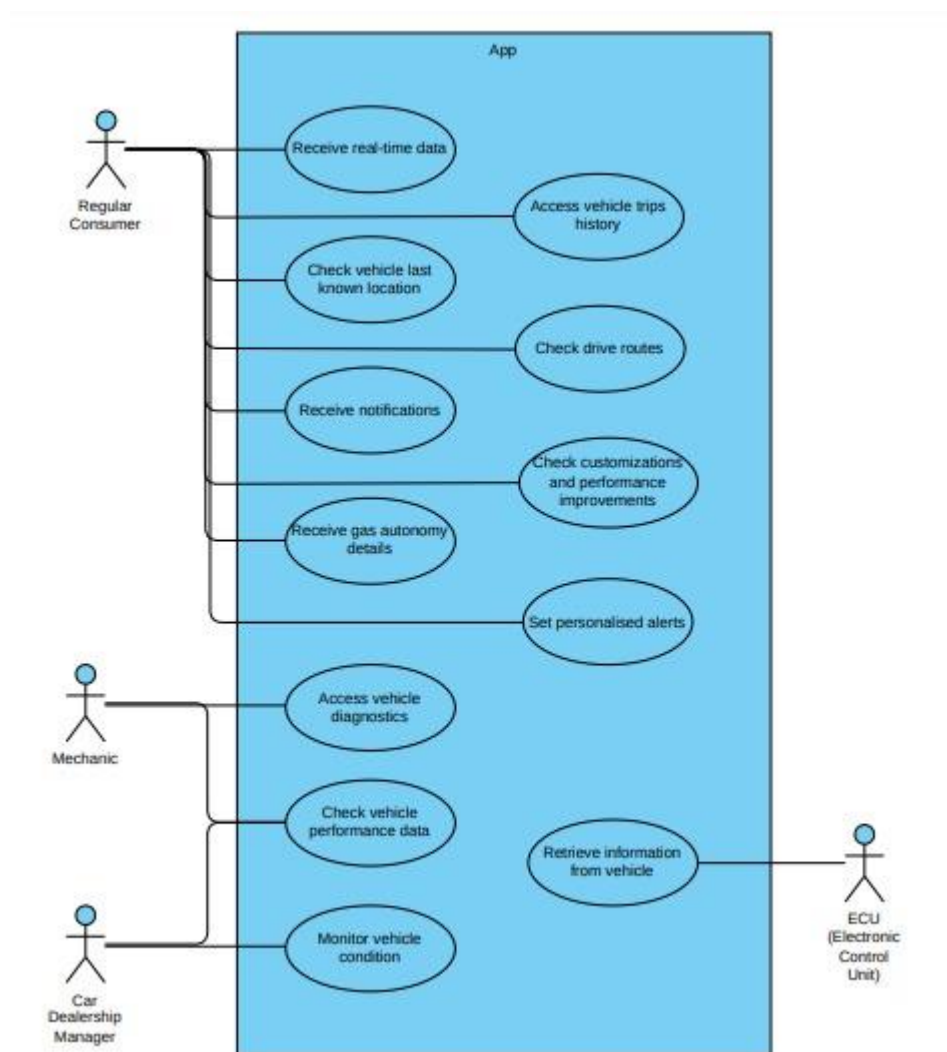
The product envisioned is an application (web/mobile) with the objective of providing valuable information about its user's car in an efficient, concise and simple way. To retrieve such information a custom device, integrated with GPS, is installed in the car, being connected to the ECU (Electronic Control Unit -> gives information regarding everything about the car, such as the stats and status of each component).

With the information provided by the application the user's experience of owning a car would be much facilitated, with the opportunity to get warned about car malfunctions, oil level, breaks/tires states, inspection warning (data both provided by the user so that he can follow it and save it in one place and by the car sensors).

Most modern cars already provide a lot of useful information, for example recent BMW models, which are full of notifications about any problem or details about the car without the need to be turning on the ignition, watching how many times the motor light blinks and

then search in the manual what's the issue. That is what we are aiming for, while also providing the greatest number of details possible, bringing it to all current cars. In that way our app would be able to have the same utility of OBD2 devices, while needing only one installation of the device and providing more information and alerts in a more readable way.

Conclusively, our objective is to make the use of systems like OBD2 more beginner friendly, turning people's attention more towards the importance of having a car well taken care of, since that's the first step for road safety and allowing any entry level user from taking advantage of our app, independently from it being an expert in mechanics or someone who's ignorant of any car concepts. We plan to provide important information that could be used for good, and in an accessible way.



The requirements retrieval was solely made by the members of the group, in which the idea was confirmed by the teacher. For us to accomplish this we used predefined ideas and common sense along with research in different websites related to the functioning of cars as well as the technologies included in one, and the importance of them. (Research links will be posteriorly pointed out in point 5)

Inspiration

As all the members of the group are car users, the concept of the application came from certain needs that we feel that would be fulfilled with certain features, or in this case with our application. Some of us are curious, others have familiars who are mechanics, therefore building requirements was facilitated. Besides, rational thinking was used in requirements, along with overall knowledge and searching.

Personas and Scenarios

Personas (Name, Age, Role/Job):

- Maria, 38 years old, Lawyer
- Ricardo, 52 years old, Taxi Driver
- Albano, 35 years old, Teacher
- Inês, 30 years old, Photographer
- Marta, 37 years old, Journalist
- Diogo, 28 years old, Programmer
- Sofia, 33 years old, Race Driving Technician
- Luís, 27 years old, Car Enthusiast
- Hugo, 37 years old, Test Drive Supervisor
- Mateus, 42 years old, Dealership Manager
- António, 58 years old, Mechanic
- Francisco, 30 years old, Musician

Main Scenarios:

- While working at his shop, António, finds that the OBD2 that him and his employees use has a weak display of information and is often confusing to new workers, so he needs a quick and simple way to get all a client's car information. For that he connects our device to the car ECU and, using our app logs in and links the car to his account.
- Marta has a busy life as a journalist, living on the road, and so, already had problems because of her distraction, causing her to forget to take her car to the maintenance and repair. Because of that, she dreams of a way to be warned when she needs to do such things or when something is wrong with her car. To do so she gets our device installed on her car and, using or app, logs in, links her car to her account and starts getting alerts when something is wrong, or maintenance is appropriate.

Product requirements (User stories)

User Stories:

(The personas used are the ones defined in the previous tab)

User Story: As António, I want to access detailed diagnostics of customer vehicles, so I can quickly identify issues and carry out repairs efficiently.

Acceptance Criteria:

- The system provides a clear dashboard showing all vehicle diagnostics in real-time.
- The diagnostics include detailed information such as engine status, battery level, and error codes.
- The system allows filtering diagnostics by vehicle components (engine, transmission, etc.).
- The system allows the link of several vehicles to the same customer.

Priority: 5

Difficulty: 4

User Story: As Hugo, I want to check the last test drive routes and performance data for cars, to ensure that the vehicles are in good condition before showing them to potential buyers.

Acceptance Criteria:

- The system logs all trip routes with timestamps.
- Performance data (e.g., speed, fuel consumption, engine performance) is available for each trip.
- Alerts are generated if any performance anomalies are detected during drives.

Priority: 3

Difficulty: 4

User Story: As Luís, I want to track all the custom modifications and performance improvements made to my car, so I can optimize it for future tuning and events.

Acceptance Criteria:

- The system allows to log modifications made to the vehicle.
- Performance improvements are tracked with before-and-after data (e.g., horsepower, torque, etc.).
- A summary report of all custom modifications (and their impact on performance) is available

Priority: 1

Difficulty: 3

User Story: As Mateus, I want to monitor the condition of cars in my dealership, so I can ensure they are maintained properly and ready for sale or test drives.

Acceptance Criteria:

- The system shows the maintenance status of all vehicles associated with the client.
- It alerts the user when any vehicle requires servicing or inspection.
- Vehicle history, including past services and test drives, is available in the system.
- Notifications are triggered if a car has been idle/stopped for a long period.

Priority: 4

Difficulty: 3

User Story: As Sofia, I want to receive real-time data on the car's technical performance during races, so I can adjust and ensure the car is running at its optimal level.

Acceptance Criteria:

- The system provides real-time telemetry data during trips (e.g., engine temperature, tire pressure, speed, etc).
- Alerts are triggered when critical performance thresholds are exceeded (e.g., overheating).
- The user can view past trips data for comparison and optimization.

Priority: 4

Difficulty: 5

User Story: As Diogo, I want to receive alerts for oil and coolant levels, so I can avoid issues and plan maintenance in advance.

Acceptance Criteria:

- Notifications are sent when the oil or coolant level falls below the set threshold.
- Historical data on oil and coolant levels is available for monitoring trends.

Priority: 4

Difficulty: 3

User Story: As Marta, I want to be notified when my car's mandatory inspection is approaching, to avoid fines and legal issues.

Acceptance Criteria:

- The system sends reminders 30 days, 7 days, and 1 day before the mandatory inspection date.
- A countdown to the inspection date is displayed on the dashboard.
- Notifications include required documents and checks for the inspection

Priority: 2

Difficulty: 2

User Story: As Inês, I want to get a prediction of how long my car can run with the current fuel level, so I can plan my trips as a freelancer.

Acceptance Criteria:

- The system calculates an estimate of the remaining driving range based on current fuel levels and recent driving habits.
- Notifications are triggered when the fuel level is critically low.

Priority: 3

Difficulty: 3

User Story: As Albano, I want to check the last known location of my car, so I can remember where I parked it when I forget.

Acceptance Criteria:

- The system shows the last recorded location of the car on a map.
- The location data includes a timestamp of when the car was last parked.

Priority: 2

Difficulty: 2

User Story: As Ricardo, I want to access my car's average monthly fuel consumption, so I can better manage my expenses and earnings.

Acceptance Criteria:

- The system calculates and displays the average monthly fuel consumption in litres/gallons and cost.
- Historical fuel consumption data is available for up to 12 months.

Priority: 4

Difficulty: 2

User Story: As Maria, I want to receive a notification whenever my car is moved, to ensure it hasn't been stolen.

Acceptance Criteria:

- The system sends an instant notification to the phone when the car is turned on.
- The notification includes the time and current location of the vehicle.
- A confirmation request is sent to the user to verify whether the movement was authorized, if unauthorized, the system provides an option to alert local authorities.

Priority: 4

Difficulty: 3

User Story: As Francisco, I want to get the location of close mechanics along with my alerts so that I know where to take my car to fix the issue.

Acceptance Criteria:

- The system gives a list of close mechanics or shops along with the alert every time the car has a big problem
- The list has the GPS location associated with each shop

Priority: 2

Difficulty: 5

Functional Requirements:

- Display of car mechanical information
- Logging System
- Display of car electrical information
- Display of car legal information
- Alert notifications
- Access to car last known location
- Access to recent trips information
- Real-Time information
- Allow multiple cars per user
- Simple device linking process
- Display POI locations

Non-Functional Requirements:

- Secure Data Base
- Concise and consistent Data Base
- Optimized data processing to guarantee the shown data is as close to real time as possible
- Usable on both mobile devices and computers
- User interface must be easy to use
- Be able to handle a broad number of users as well as a large influx of users
- Secure message queue
- System should be maintainable and backwards compatible in case of hardware modifications

3 Architecture notebook

Key requirements and constraints

<Identify issues that will drive the choices for the architecture such as: Will the system be driven by complex deployment concerns, adapting to legacy systems, or performance issues? Does it need to be robust for long-term maintenance?

Identify critical issues that must be addressed by the architecture, such as: Are there hardware dependencies that should be isolated from the rest of the system? Does the system need to function efficiently under unusual conditions? Are there integrations with external systems? Is the system to be offered in different user-interfacing platforms (web, mobile devices, big screens,...)?

E.g.: (the references cited in [XX] would be hypothetical links to previous specification documents/deliverables)

There are some key requirements and system constraints that have a significant bearing on the architecture. They are:

- The existing legacy Course Catalog System at Wylie College must be accessed to retrieve all course information for the current semester. The C-Registration System must support the data formats and DBMS of the legacy Course Catalog System [E2].
- The existing legacy Billing System at Wylie College must be interfaced with to support billing of students. This interface is defined in the Course Billing Interface Specification [E1].
- All student, professor, and Registrar functionality must be available from both local campus PCs and remote PCs with internet dial up connections.
- The C-Registration System must ensure complete protection of data from unauthorized access. All remote accesses are subject to user identification and password control.
- The C-Registration System will be implemented as a client-server system. The client portion resides on PCs and the server portion must operate on the Wylie College UNIX Server. [E2]
- All performance and loading requirements, as stipulated in the Vision Document [E2] and the Supplementary Specification [15], must be taken into consideration as the architecture is being developed.>

Architeturual view

- Discuss architecture planned for the software solution.
- include a diagram

Module interactions

- explain how the identified modules will interact. Use sequence diagrams to clarify the interactions along time, when needed
- discuss more advanced app design issues: integration with Internet-based external services, data synchronization strategy, distributed workflows, push notifications mechanism, distribution of updates to distributed devices, etc.>

4 Information perspective

<which concepts will be managed in this domain? How are they related?>

<use a logical model (UML classes) to explain the concepts of the domain and their attributes>

5 References and resources

<document the key components (e.g.: libraries, web services) or key references (e.g.: blog post) used that were really helpful and certainly would help other students pursuing a similar work>