AutoCore
System Requirements Specification and Architecture Vision

Authors: Carlos Verenzuela (114597), Lázaro Sá (115884), Isac Cruz (90513), Ricardo Carmo (112657), Rúben Costa (114190), Rui Rosmaninho (113553)

Date: 06/12/2024
Revisions: Version 1.0

Amendment History

Version	Date	Modified By	Reviewed By	Amendment Details
0.1	13-11-2024	Pedro Teixeira	All	Initial version with table of contents.
0.2	22-11-2024	Carlos Verenzuela	Rúben Costa	User stories and table of FR and NFR. Included actors. Began formatting the FR and NFR tables.
0.3	23-11-2024	Rúben Costa	Rui Rosmaninho	Updates made to User Stories 2, 3, and 7. Updates made to FR and NFR tables.
0.4	25-11-2024	Rui Rosmaninho	Isac Cruz	Added the architecture diagram and description. Updates made to the Current Implementation Plan and Status topic and Current Functional Testing Plan topic.
0.5	26-11-2024	Isac Cruz Lázaro Sá	Rui Rosmaninho	Text translation to English. Added Use Cases diagram and updated the descriptions.
0.6	26-11-2024	Carlos Verenzuela	Rúben Costa	Added UI screens and their descriptions.
0.7	27-11-2024	Rúben Costa	Carlos Verenzuela	Added remaining UI screens and their descriptions.
0.8	27-11-2024	Carlos Verenzuela	Rúben Costa	Updated UI screens description. Started UI screens description translation.
0.9	27-11-2024	Rui Rosmaninho	Isac Cruz	Added list of tables & figures. Unified User Stories and updated Requirement tables. Updates made to Actors & Use Cases topics. Updates made to Viable Solution Vision topic.
1.0	28-11-2024	Rúben Costa	Rui Rosmaninho	Update UI screens description translations. Related Work + translation. References. Update in the description of the images in Related Work.

Table of Contents

List of Tables

List of Figures

1. Introduction

1.2. Purpose

This document is a detailed system requirements specification of a vehicle information cluster. The presented requirements are built on law requirements specifications as well as functional and practical needs of general driving requirements.
This document will later be followed by documentation detailing each component and choice regarding the project. Any required change during the development will be mentioned on the last report.

1.3. Overview of the project

The AutoCore system targets a more connected, user-oriented vehicle ecosystem with seamless integration into Smart Cities. It focuses on driver experience, safety, and adaptability with features such as real-time monitoring, intuitive interfaces, and automated responses.

1.4. Definitions and Acronyms

Considered acronyms are defined in Table 1.

Acronym	Definition
UC-X	Use Case X
US-X	Use Story X
CAN	Controller Area Network
5G	Mobile network of the 5th generation
ITS-G5	Wi-Fi-based standard (DSRC) developed for fast and reliable communications
HMI	Human-Machine Interfaces

Table 1 - Considered acronyms and their definitions.

1.5. Related Work

Modern vehicles are increasingly connected, featuring sensors and touchscreen interfaces. This evolution reflects the growing need for communication between devices both inside and outside the vehicle, as highlighted in [1].

This creates new opportunities for software and connected ecosystems, as well as challenges in implementation. For a good driver's experience, it is necessary to ensure intuitive interfaces, incorporating familiar symbols and elements that users can quickly recognize and understand. The design should prioritize simplicity, allowing interactions to feel natural and effortless, which enhances usability and reduces distractions [2]. Furthermore, integrating Smart Cities into the process can provide more relevant information and enhance the safety of vehicle users.

Projects such as [3] have already explored interaction methods in infotainment systems (information + entertainment). Additionally, platforms like Android Automotive and Apple CarPlay simplify the development of vehicular applications, enabling access to vehicle data (e.g., via CAN) and integration with technologies such as 5G and ITS-G5 to support V2X (Vehicle-to-Everything).

Figure 1 - Comparison of current interaction solutions within modern vehicles.

Figure 1 ilustrates the current state of in-vehicles solutions in the market showcasing three distinct approaches to HMI within modern vehicles.

The BMW system emphasizes advanced connectivity with multi-screen layouts and customizable displays. Volvo adopts a vertical touchscreen interface, streamlining navigation and infotainment control. Meanwhile, Honda focuses on simplicity with a wide horizontal layout, offering seamless access to essential vehicle features.

These examples underline the automotive industry's commitment to touchscreen-dominated interfaces, aiming to create visually appealing and user-centric systems. However, the reliance on touchscreens also presents challenges, particularly regarding the driver distraction and usability under varying driving conditions.

Future innovations in HMI should consider integrating complementary technologies such as voice recognition, gesture control and adaptive UI designs. Such advancements will enhance safety.

2. Requirements Elicitation

The Requirements Elicitation process involves understanding the needs, expectations, and constraints of potential stakeholders. This ensures that the system being developed meets the intended objectives and provides value to its users.

2.1. User stories

The following user stories summarize the primary needs identified, focusing on user-centered design principles.

2.1.1. User Story 1

While driving, James, would like to receive real-time intelligent assistance to find the best route to his destination, minimize distractions, and to suggest alternative routes in case of traffic congestion. He would also like to be warned of possible problems with his vehicle, such as low engine oil or coolant, before and during driving in a way that doesn’t cause him to panic.

Apart from this, António must be informed about smart city and road conditions regarding accidents, weather alerts, traffic, and roadworks to adjust his driving accordingly and in case of adverse weather conditions or loss of control, António wants the system to notify other road users instantly to avoid accidents and overall road safety.

2.1.2. User Story 2

Jennifer is a driver who wants to personalize the user interface of the vehicle system to make her driving experience more comfortable and focused. She wants to adjust the display so that only the data relevant to driving is displayed, which would minimize distractions.

She would also like an adjustable theme and brightness of the display, especially at night, considering her photosensitivity, to make her feel comfortable while driving, doing so by utilizing voice commands.

2.1.3. User Story 3

As a company, Ford Motor wants the system to be expandable for future use and in compliance with government regulations. More precisely, Ford Motor wants the system to allow easy integration of additional sensors and communication channels to support new features and ensure compliance with mandatory requirements.

The company wants the system, on the introduction of new regulations and laws, to be flexible to integrate necessary changes that could be done without redoing the entire system to save costs and ensure long-term viability.

2.2. Actors

This section is a general overview of different user roles and their respective characteristics, which will guide the Low-Level Design of the system, ensuring the design of the system answers to the needs of the users and technical environments.

It will be summarized in Table 2, thus giving insights into the kinds of user interfaces needed for data entry, self-service, or trained users, the context in which they occur, global or organizational. It also describes the foreseen user platforms, which include the expected devices, operating systems, and browsers.

Actor	Role
Driver	Consume information (e.g., traffic, vehicle status), provide commands (via voice or touch), and receive visual/audio feedback.
Passenger	Assist the driver in using the system while the car is in motion or independently utilize features on the central display.
Car Service Technician	Perform diagnostics, apply updates, or configure sensors and devices.
Smart City Manager	Provide updated information consumed by the system to benefit drivers and optimize city-wide operations.

Table 2 - Actor Role Description

2.3. Use Cases

The analysis of the user stories results in the following set of use cases. Figure 2 helps in understanding how different actions and system processes are interconnected.

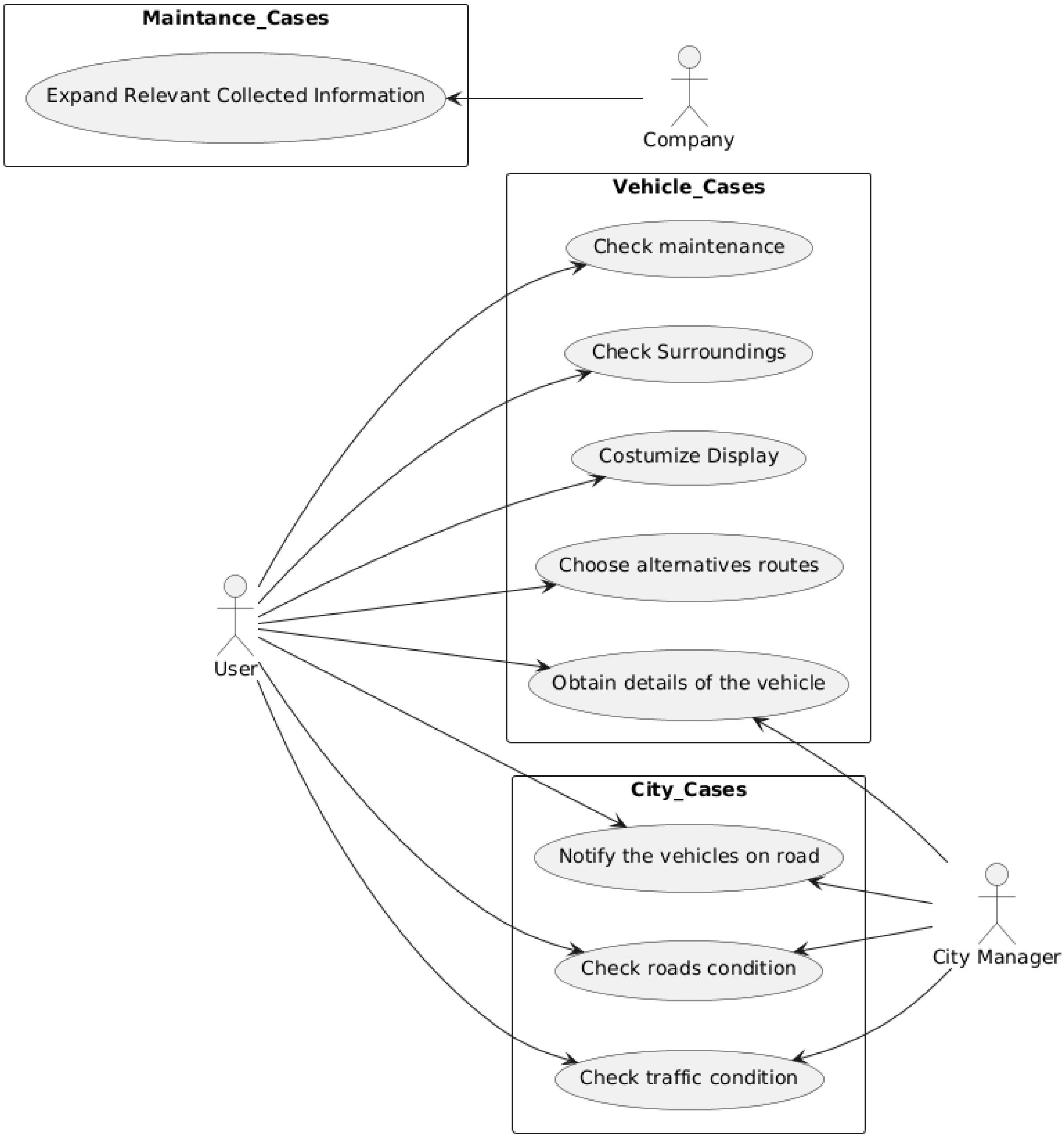


Figure 2 - Use Cases Diagram

Table 3 provides a detailed description of each use case serving as a foundation for the Low-Level Design, ensuring that the system's behavior aligns with user requirements and expectations.

Use Case ID	Use Case Name	Use Case Description
UC-1	See alternate routes	Use the main display to choose a destiny and be able to see and choose from the different routes available to get there.
UC-2	Customize display	Change the location of different elements of the heads-up display and main display to customize them to the user's tastes.
UC-3	Check Surroundings	The user should be able to see the area surrounding the vehicle through the displays.
UC-4	Check maintenance	The user should be able to check the maintenance conditions of their vehicle and be notified of any abnormal values.
UC-5	Obtain details of the vehicle	The user should be able to see information about the state of the vehicle, i.e. speed, fuel level, rpms, etc...
UC-6	Notify the vehicles on the road	Notify other vehicles of emergent situation, propagate relevant messages.
UC-7	Check roads condition	Request road extra information about roads state as works on the roads, closed roads and other road information.
UC-8	Check traffic condition	Request traffic information to the city and display it to the driver in a usable way.
UC-9	Expand relevant collected information	Expand the data structure to allow different types of information.

Table 3 - Use Cases Summary & Description

2.4. Overview - Proposed UI Layout

UI paper prototypes are crucial for gaining insights and exploring various design and implementation options for the interface.

These designs were developed with the objective of presenting essential information to the driver or, in the case of the central display, facilitating driver interaction.

Through these UI screen proposals, we aim to deliver an intuitive and informative interface tailored to real-time needs for safe and efficient driving. The design elements were carefully selected to provide quick access to vital information such as speed, RPM, temperature, and alert notifications, while minimizing distractions.

The following sections outline the key components of the UI and provide the rationale behind each design choice.

Additionally, this process serves to demonstrate our awareness of the available options for each scenario and to justify, from an engineering perspective, the decisions made in implementing the design as represented in the respective drafts.

In the various design prototypes for the display positioned in front of the driver, a white curve is included to represent the space occupied by the car's steering wheel.

2.4.1. UI Screen 1: Initial customization screen

This interface should be the first to appear, allowing the user to customize the UI.

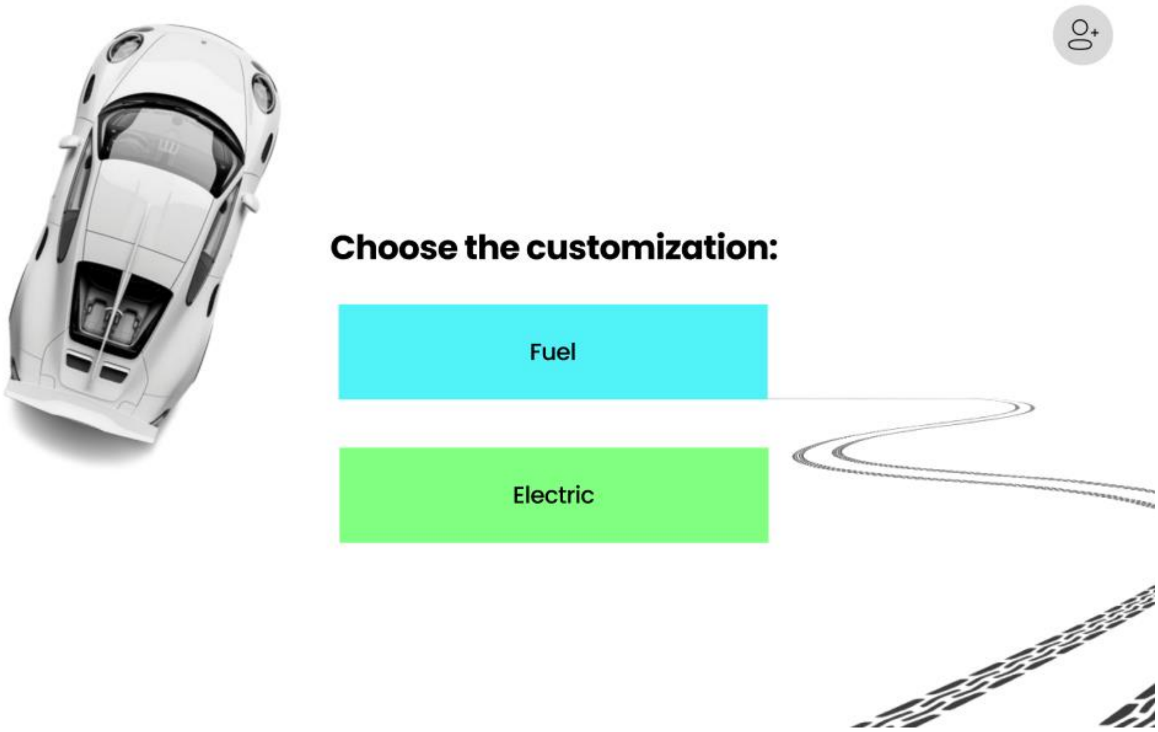


Figure 3 - UI Screen 1: Initial customization screen

When the main user interface (UI) screen is activated, the first screen to appear will allow the user to select the interface that best suits the type of car, such as choosing between a fuel or electric car. However, if it is deemed that this question is unnecessary, this mockup can be repurposed for other types of customizations. For instance, it could be used to enable dark mode, activate a color-blind mode, specify the type of color blindness, or configure other personalization options.

2.4.2. UI Screen 2: Main Driver Screen (initial design)

Figure 4 - UI Screen 2: Main Driver Screen

This is our initial design, focusing on displaying the most essential information on the main screen: gear shifts (depending on whether the vehicle is manual or automatic), warnings, speedometer, RPM, fuel level, engine temperature, and GPS.

Warning icons are positioned on both the right and left sides of the display. These include legally required pictograms, and when an alert is triggered, the corresponding icon changes color to draw attention.

At the center, in the upper-left corner, the engine temperature is displayed. Next to it is a combination of the speedometer and tachometer (RPM): the speedometer displays the speed in numeric form, while the tachometer is represented by a color gradient, indicating rotation levels up to the allowed limit. To the right of these elements, the fuel level is shown.

At the bottom of the display is the GPS, which provides the route and relevant details, such as the estimated time to the destination and the next turn.

On the right-hand side, there is an indicator for new notifications. However, full notification details are not shown on this display; they are instead accessible on the secondary display.

This UI screen prioritizes presenting the most critical information directly in front of the driver, including speed, an overview of the route, and clear indicators, avoiding unnecessary distractions or excessive details.

2.4.3. UI Screen 2: Secondary Screen (initial design)

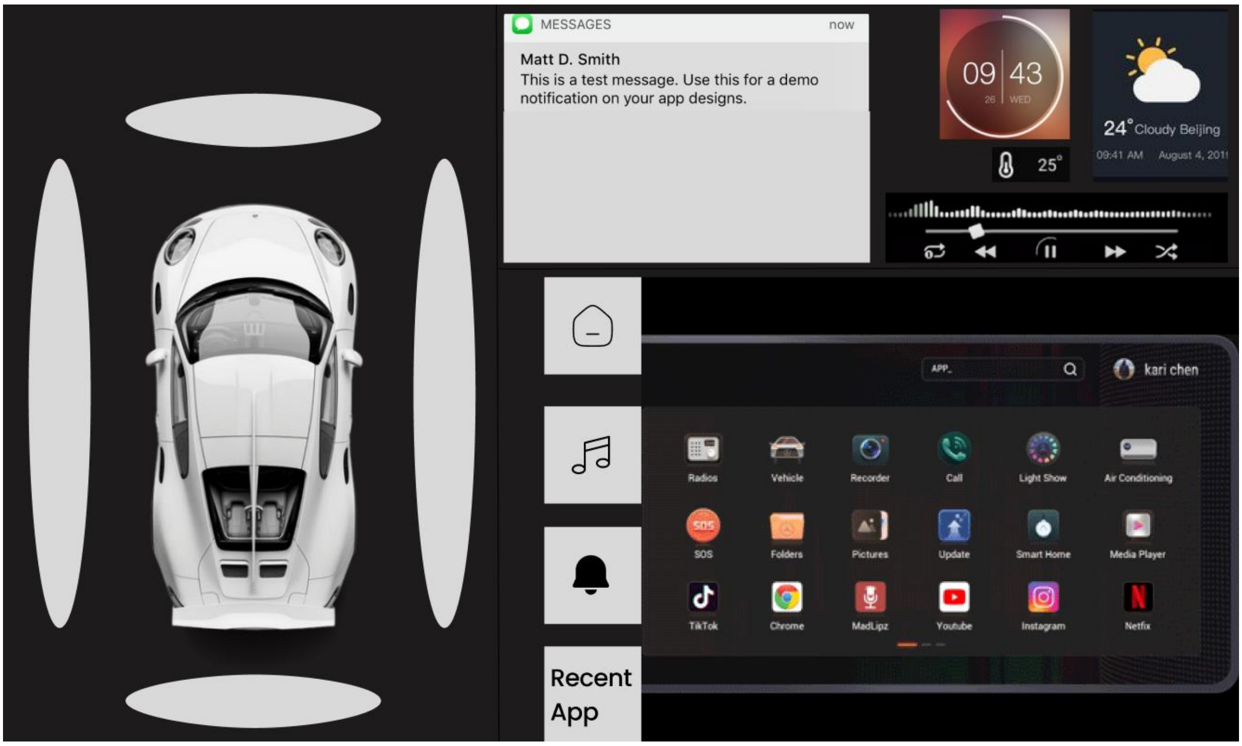


Figure 5 - UI Screen 2: Secondary Screen

On the central display, the left side features a representation of the vehicle that indicates active lights and proximity sensors. This visual element provides information about the proximity of obstacles, such as people, walls, or other objects. One potential approach involves the use of color coding, where zones near the car's exterior shift progressively from white to yellow, orange, and finally red as the object gets closer. This system could also incorporate audio feedback, triggering alerts as the distance decreases.

The upper-right corner is dedicated to the notifications area, allowing users to view full details of any notification. This section also includes controls for music playback, as well as information on the date, time, in-car temperature, and weather conditions.

The lower section of the display is allocated to the cluster's remaining applications, designed to function similarly to a smartphone interface. Key buttons include a "home" button for returning to the main screen, a music button for quick access to the music application, a notifications button for expanding specific notifications, and a dynamic space displaying the most recently used application.

2.4.4. UI Screen 3: Main Driver Screen (second iteration)

Figure 6 - UI Screen 3: Main Driver Screen

Figure 7 - UI Screen 3.1: Main Driver Screen Alternative

For the second design, we adopted a more minimalistic approach, featuring an information cluster positioned at the top of the screen and warnings located on the right side. These warnings appear only when necessary and can also display notifications above them when applicable.

Speed is displayed prominently on the right, alongside a discreet representation of the vehicle.

In this prototype, we decided to divide our ideas into two possibilities, allowing the user to customize the interface based on their preferences.

First UI: Dual Adaptative GPS Mode

In this mode, both GPS displays are shown simultaneously. The larger GPS, highlighted on the screen presents the most relevant information based on the user's current location. Meanwhile, the smaller GPS offers complementary data in a more discreet format.

The two modes dynamically adjust their positions and sizes depending on the context. On highways, the detailed map with streets and directions takes priority, while the secondary display showing surrounding areas is minimized. In urban environments, the map that highlights nearby vehicles and the surrounding area becomes larger, while the detailed map shrinks to a smaller size.

This dual-display configuration ensures that users have a comprehensive view of both navigation and surrounding information, without compromising clarity or usability.

Second UI: Single Adaptive GPS Mode

In this mode, only one GPS display is shown, centered on the screen and it adapts automatically to the driving context. On highways, the interface emphasizes immediate navigation details such as upcoming turns and nearby streets. In urban areas, the focus shifts to the surrounding environment, showing nearby lanes and vehicles, while still maintaining clear navigation instructions.

Additional information is organized around the display. The bottom-left corner shows the engine temperature and fuel level, while the speedometer and tachometer are prominently displayed at the top center of the screen.

2.4.5. UI Screen 3: Secondary Screen (second iteration)

Figure 8 - UI Screen 3: Secondary Screen

In this UI screen, we build upon the previous design with several refinements. The primary button bar has been simplified: instead of icons representing the most frequently used applications, we have replaced them with simple geometric shapes—such as a square, triangle, and circle—similar to those commonly found on smartphones. These shapes serve as faster, more minimalist shortcuts, aligning with modern user preferences.

Furthermore, supplementary information, including music, weather, and date/time, has been slightly reorganized to improve clarity and enhance usability.

2.4.6. UI Screen 4: Main Driver Screen (third iteration)

Figure 9 - UI Screen 4: Main Driver Screen

In this third proposal, the bottom-left corner displays the engine temperature, with the fuel level positioned directly above it. At the center, the map is accompanied by navigational indicators, while the speedometer and tachometer are located near the battery level, optimizing space for navigation.

In the central area, on the opposite side, important messages are displayed. On the right side, notifications are positioned at the top, while warnings appear at the bottom. These warnings do not permanently occupy this space; they appear only when necessary. The remaining area is used to display messages from the SmartCity system.

2.4.7. UI Screen 5: Main Driver Screen (third iteration)

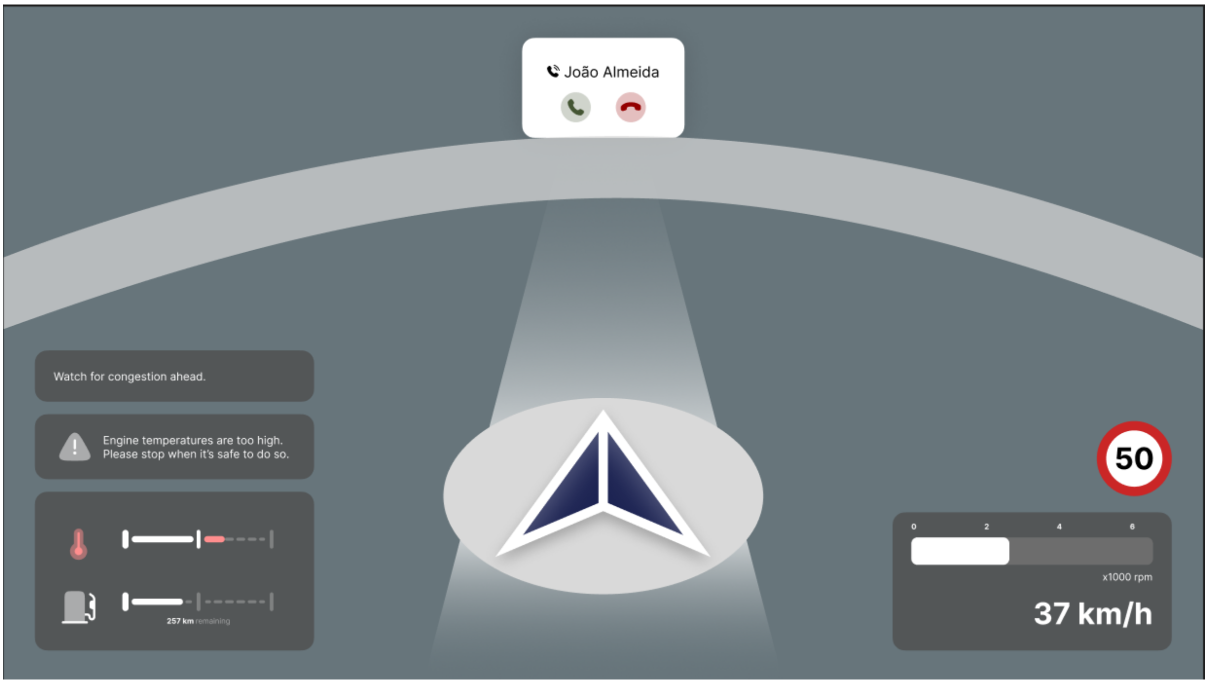


Figure 10 - UI Screen 5: Main Driver Screen

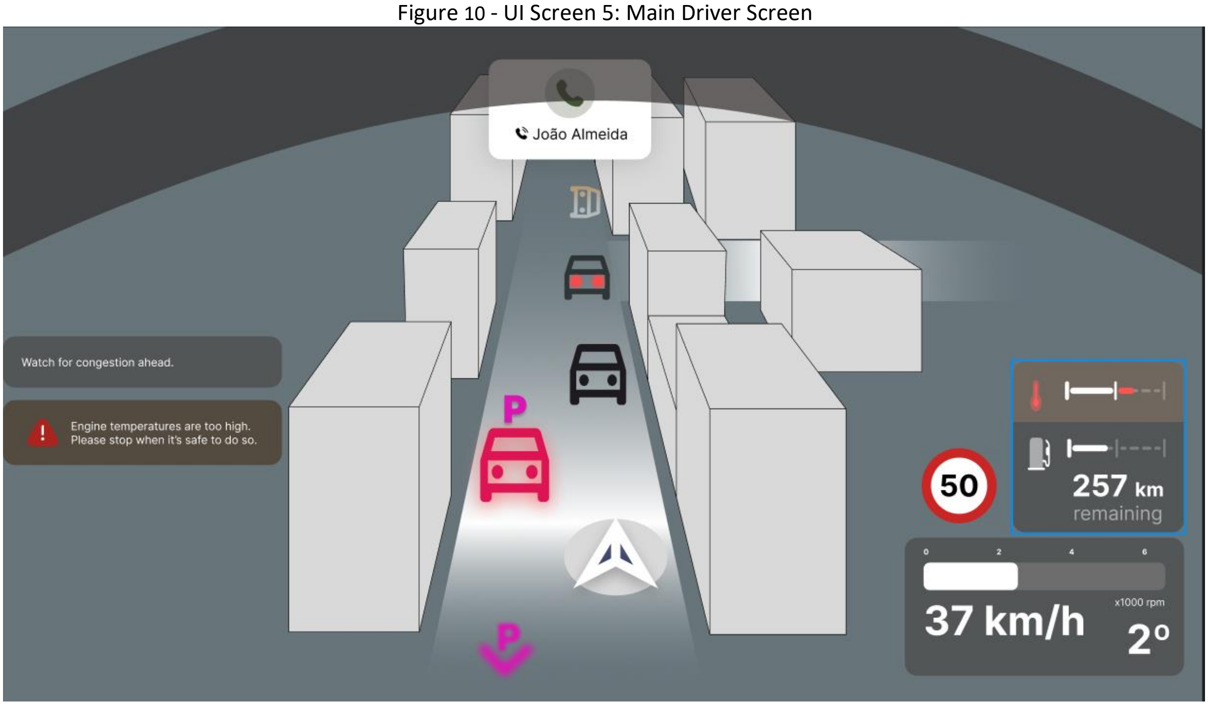


Figure 11 - UI Screen 5.1: Main Driver Screen Alternative

“A clean and functional design that breaks away from traditional standards.”

This dashboard was designed to provide an intuitive and adaptable experience, prioritizing the most relevantinformation for each context. In the top-left corner, messages are displayed. These disappear automatically if they are neither critical nor important. The display is dynamic and adjusts to the context: in the center of the screen, the route is shown — in urban areas, it focuses solely on the necessary directions, while on highways, it also includes surrounding elements.

In the bottom-left section, the tachometer is displayed, accompanied by the speedometer and engine temperature. Above this area, the fuel level is shown. If the engine temperature reaches concerning levels, this information will expand within this section. Additionally, the recommended speed limit is displayed nearby, adjusted according to the traffic signs of the current location.

3. Functional Requirements

This section gives a summarized list of the functional requirements, which have been derived from use cases and user stories outlined in Table 4. These form the base of the design ensuring that the system meets the needs of the users and provides the intended functionalities.

Req. ID	Requirement Name	Requirement Description	Priority	Associated User stories
REQ-F1	Hands-Free Commands	The system must support hands-free operation, such as voice commands	4	US-2
REQ-F2	Smart City Integration	The system must be capable of receiving traffic data from the Smart City infrastructure	1	US-1
REQ-F3	Real-Time Traffic Updates	The system must automatically update traffic information based on data received from the Smart City infrastructure	3	US-1
REQ-F4	Hazard Alerts	The system must display hazard warnings in a manner that ensures users notice them	2	US-1
REQ-F5	Full-Duplex Communication	The system must support full-duplex communication for continuous monitoring and data exchange with the Smart City infrastructure	1	US-1; US-3
REQ-F6	Vehicle Component Monitoring	The system must monitor the vehicle's condition, including key components such as the engine, oil levels, etc...	1	US-1
REQ-F7	Interface Customization	The system should allow users to personalize the interface, including selecting displayed data, theme, etc...	3	US-2

Table 4 - Functional Requirements Summary

4. Non-Functional Requirements

This section outlines a summary of the non-functional requirements, which describe performance, security, usability, and scalability standards that the system shall meet. These requirements, outlined in Table 5 provide assurance that the system will be robust, reliable, and capable of adapting to future needs.

Req. ID	Requirement Name	Requirement Description	Priority	Associated User Stories
REQ-NF1	Hands-Free Interaction Accuracy	The system must ensure precision of hands-free commands, such as voice recognition	4	US-2
REQ-NF2	Data Exchange	The system must exchange information with a response time of less than 1 second	1	US-1
REQ-NF3	Reliability	The system must ensure the reliability of the data.	1	US-1
REQ-NF4	Alert Handling	The system must ensure that alerts are presented in a way that neither jeopardize nor cause panic to the driver.	1	US-1
REQ-NF5	Scalability	The system must be expandable to support new sensors and additional communication methods.	3	US-3
REQ-NF6	User-Friendly	The system must provide a simple and intuitive interface, regardless of customization	2	US-2

Table 5 - Non-Functional Requirements Summary

5. Viable Solution Vision

5.1. Overview of the architecture

This section outlines the structure that the system will take, major modules, and their interaction. It focuses on how the user interface, data handling and communication system can be integrated into one seamless framework. Figure 12 provides an overview of the system's structure, displaying the major modules and their interactions.

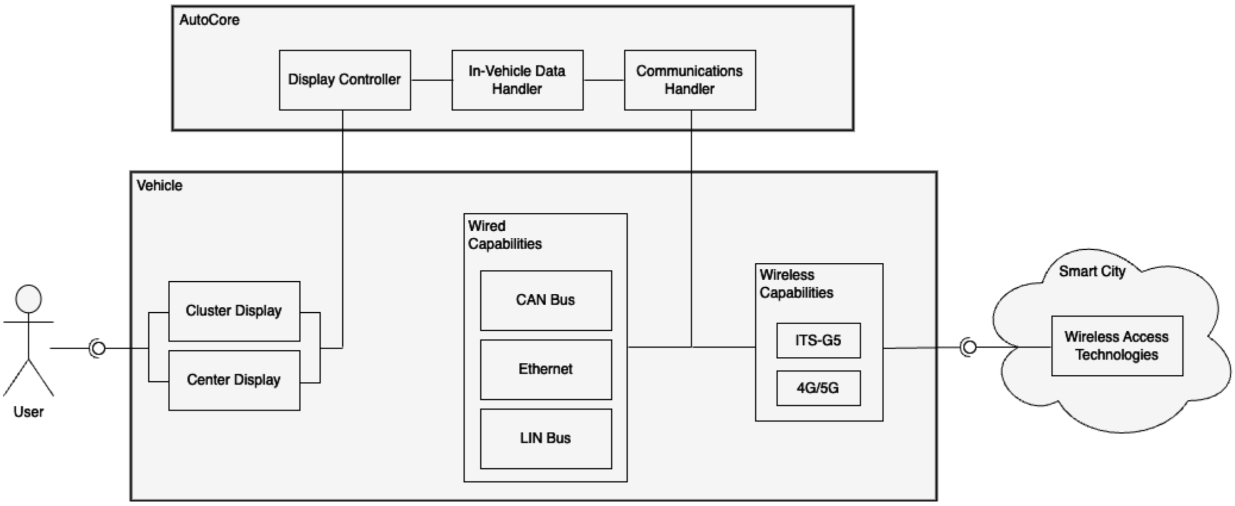


Figure 12 - Global Architecture

Following this, Table 6 offers a detailed description of each module and its submodules, highlighting their respective roles and how they contribute to the system.

Module	Submodule	Description
AutoCore	Display Controller	Acts as the central point for managing all visual outputs, ensuring that they are coherent, responsive, and user-friendly.
	In-Vehicle Data Handler	Enables seamless integration with in-vehicle data systems for access to crucial insights, higher-level system processes and decisions.
	Communications Handler	Handles communications with external networks and smart cities.
Vehicle	This module involves hardware components of the vehicle and internal system interfaces. It provides information to the driver via the Cluster and Center Display.	
	Wired Capabilities	Includes the CAN Bus, Ethernet, and LIN Bus for data exchange between in-vehicle sensors and actuators.
	Wireless Capabilities	Handles the vehicle's external communication, enabling connectivity with external systems. It supports technologies such as ITS-G5 and 4G/5G, facilitating seamless interaction with smart city infrastructure and other entities.
Smart City	Represents the external urban ecosystem, leveraging wireless access technologies for real-time data exchange.	
User	Represents the individuals interacting with the system through the vehicle's displays.	

Table 6 - Architecture's Module Descriptions

5.2. Domain Model

Once working with ITS-G5, receiving messages in CAM and DEN protocols, there is a natural need for data classification and organization. Storing received data allows a close city state monitoring for filtering relevant messages to report to the user. For this purpose, it was created a domain model for external information.

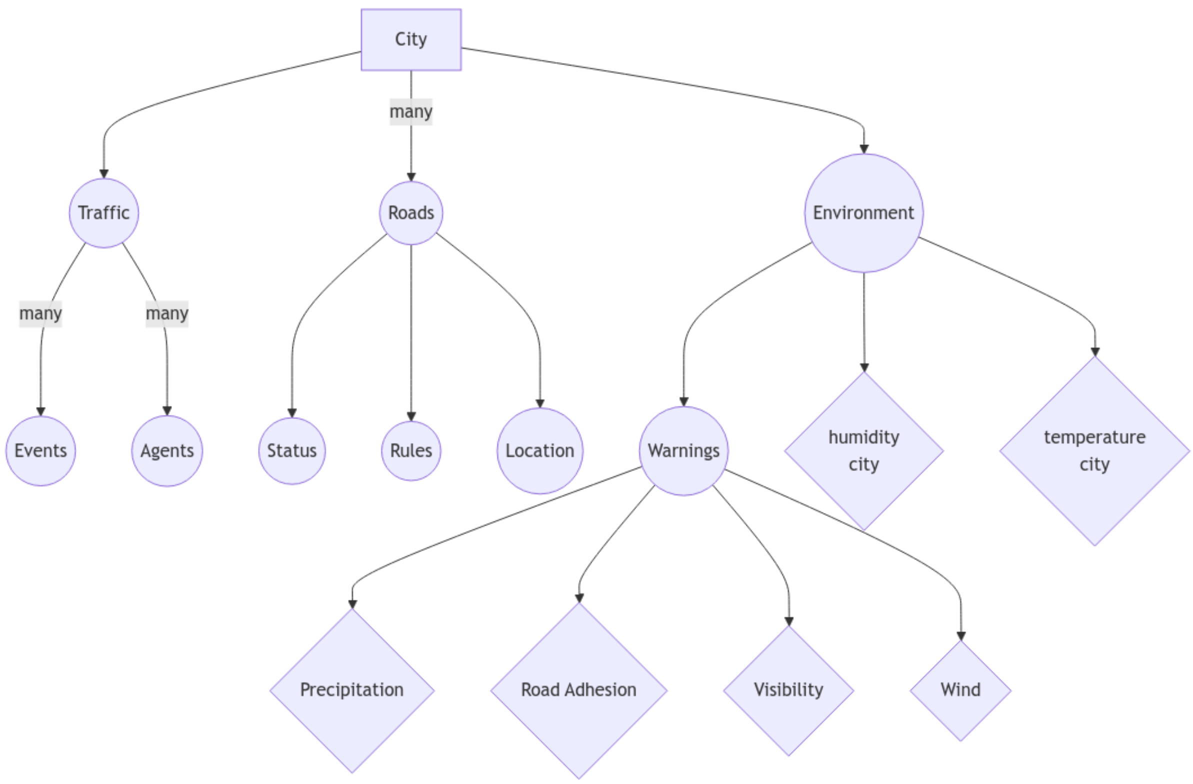
The circles in the following diagrams represent expandable nodes while the losangles represent the main values at the leaves of the branches.

5.2.1. City

For the first iteration of this model the city was categorised by 3 main topics:

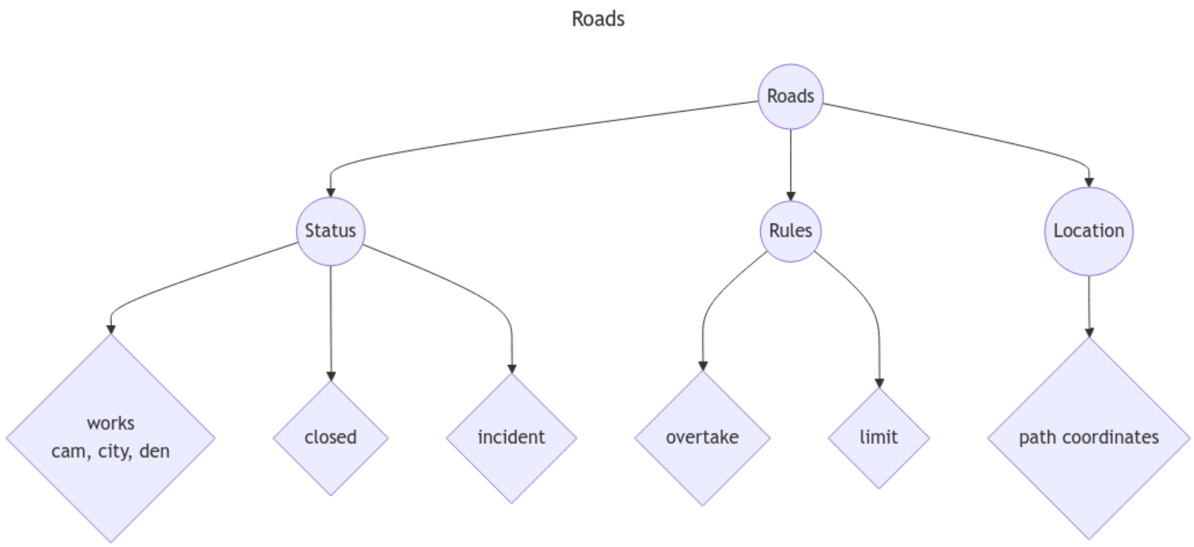
- Roads, containing the condition of each road independent from vehicles travelling by
- Traffic, referencing all traffic information independent form roads, inter-vehicle interactions
- Environment, which modules all climate condition relevant for driving activity

City



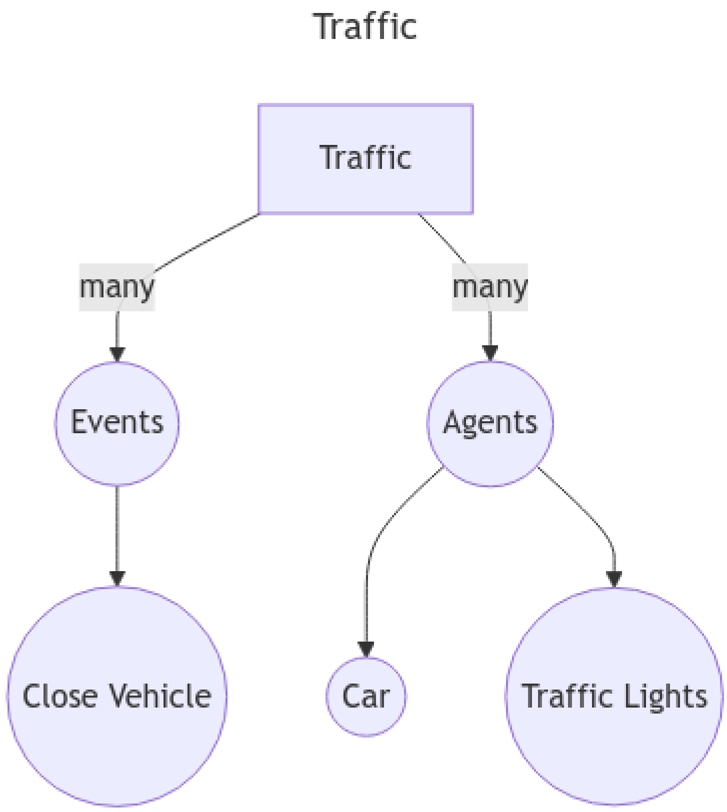
5.2.2. Roads

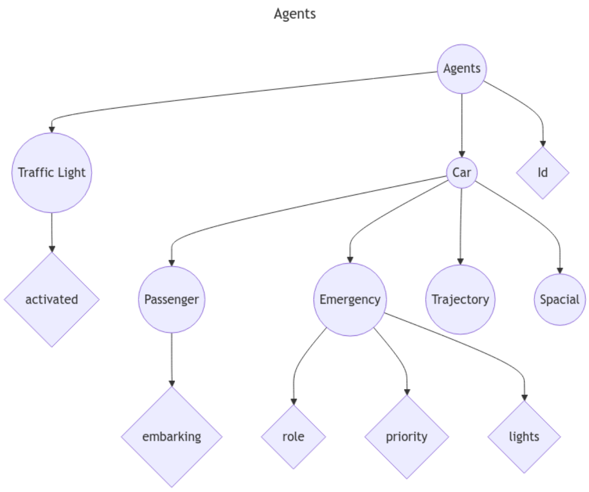
Roads are module in Status, defining road physical state. Rules, defining the legal state of the road (road rules), and Location defining the points belonging the road path.

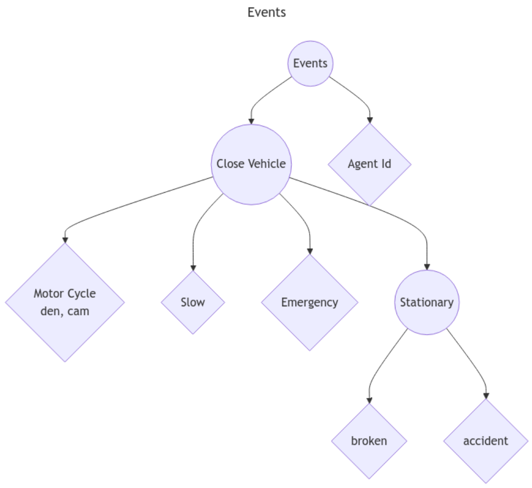


5.2.3. Traffic

Traffic topic is modules into domains: Events regarding to close Agents' behavior relevant to immediate choices, and Agents belonging to the infrastructure of the road, vehicles and other active agents able to communicate in this ecosystem.



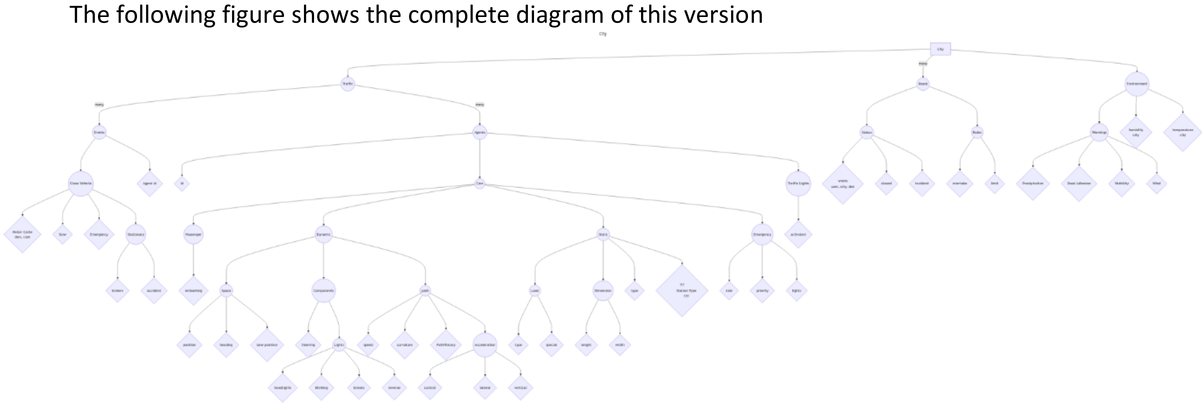




With the data regarding to these tow modules is possible to create a located digital twin of the close environment and use that data to help the user to navigate in the city.

5.2.4. Complete Diagram

The following figure shows the complete diagram of this version



5.3. General Constraints

The development of the system is bound by various constraints that influence the design related to both software and hardware. These constraints emanate from technical, regulatory, and integration requirements that should be met to ensure functionality, reliability, and conformation with the standards of the industry.

On the software side, strict adherence to industry standards and licensing terms provides limited flexibility in leveraging third-party tools. Real-time performance requirements set latency thresholds that guarantee system reliability but limit design choices. Security considerations dictate rigid compliance with cybersecurity standards that ensure the protection against unauthorized access and data breaches, further narrowing development options. Also, reliance on legacy systems complicates things since new features need to fit in smoothly with existing infrastructure and this often means increased complexity for the developers.

Hardware constraints are equally important. Compatibility with existing vehicle interfaces, such as CAN, LIN, and Ethernet buses, limits hardware component choices. Displays also must meet strict physical and technical requirements regarding resolution and refresh rates, which limit the flexibility in choosing appropriate devices. Communication hardware must also meet standards such as V2X and ITS-G5 to ensure interoperability but limits the latitude for other designs or configurations.

5.4. Assumptions and Dependencies

Certain assumptions and dependencies within the system design have important implications for its feasibility and implementation. First, there is a presumption of a smart city ecosystem in which V2X communications can be supported.

The other key dependency is a functional in-vehicle communication bus like CAN, LIN, or Ethernet, where the sensors and actuators can talk to each other. The design also assumes correct network coverage via 4G/5G connectivity and ITS-G5 for external communications and real-time operations.

There is also a dependence on the availability within the vehicle of hardware components, such as displays, sensors, and actuators, that meet various technical specifications for the operation of this system.

Finally, compatibility is assured based on the assumption that all components of the system conform to the industrial standards. Any violation of assumptions or dependencies will result in modification to the system design or additional resources.

5.5. Current Implementation Plan and Status

This section serves as a snapshot of the ongoing development process for the system. Its purpose is to provide an overview of the progress made so far, the proposed solutions for each module, and the rationale behind each decision. Table 7 will be used to clearly organize and present key information.

Module	Proposed Solution	Rationale behind decision	Status
AutoCore	Develop a centralized software to display in-vehicle systems and external networks data.	By centralizing control, AutoCore simplifies data flow management, improves system reliability, and ensures compatibility with legacy and modern vehicle components.	Proof of feasibility achieved through a basic Automotive application with working navigation between Fragments.
Display Controller	Implement a module to manage and control central displays, adaptable to various designs.	This approach ensures that for the different display types, the user will have the same experience and at the same time be flexible to future display technologies and requirements.	Early prototypes built and testing in progress.
Communications Handler	Deploy a Full-Duplex communication with both the city and the car. Abstract communication layer for all modules. It uses ITS-G5,4G/5G and CAN technologies.	The solution utilizes existing communication standards, thus ensuring reliable real-time data exchange. Isolating all communication code in one place, creates a more modular code, allowing adaptation to future hardware.	Currently researching relevant communication standards, such as ETSI, CAM, etc.
In-Vehicle Data Handler	Usage of VSS to store collected information during gathering and to provide data to every other module of the system.	Centralize data providing, abstract data collection, allows data structure expansion, deals with internal network load, uses a data system specifically design to the automotive scenario.	Design under progress and VSS adaptation for the Smart City data structure.

Table 7 - Current Implementation Plan & Status

5.6. Current Functional Testing Plan

This section describes, with the help of Table 8, the strategy and methodology that will be used in the validation of different modules of the system by employing the testing environment provided through a Raspberry Pi 5 setup with Android Automotive.

The tests are focused on ensuring that everything works as required in a real-world scenario, including the application to UI/UX design, communications, and In-Vehicle data handling.

Module	Proposed Testing Plan	Tools & Environment
Application	Developed in Android Studio and deployed on a Raspberry Pi 5 compiled with Android Automotive. Ui Automations[1] tests will be performed to verify the expected behavior of the application under different scenarios.	Android Studio; Raspberry Pi 5 with Android Automotive; Ui Automations[1];
UI/UX	Testing will be conducted using the provided displays to ensure the application's user interface is intuitive, responsive, and well performing. It will also involve iterations of mock-ups based on user feedback and application of the UXH-IVIS Heuristic[2], ensuring the system meets industry standards.	Paper Prototypes; Displays; Iterations based on user feedback; UXH-IVIS Heuristic[2];
Communications	Usage of simulated or logged data from the Smart City's infrastructure and a CAN simulator. Testing will focus on evaluating the reliability and latency of communication.	Raspberry Pi ITS-G5 Hat; Raspberry Pi CAN Hat; CAN Bus Simulator; Wireshark;
In-Vehicle Data	Testing will ensure that all data provided by sensors is properly stored, processed, and accessed using VSS.	Vehicle Signal Specification; Sensor Simulators; Data Integrity Testing Tools;

Table 8 - Current Functional Testing Plan

Figure 13 - Testbed

For testing purposes, a testbed is provided in Figure 13, enabling the observation of various results and the evaluation of the limitations associated with the different functionalities that can be implemented.

[1] Example of the UI Automator tool available at https://developer.android.com/training/testing/other-components/ui-automator

[2] Read about the UXH-IVIS Heuristic at https://www.sciencedirect.com/science/article/pii/S187705092401175X.

References

[1] C. A. DEGUZMAN, B. DONMEZ, J. HARBLUK, C. LAU and P. C. BURNS, "A Technology Scan of Touchscreen Trends in Passenger Vehicles," *A Technology Scan of Touchscreen Trends in Passenger Vehicles,* 22 setembro 2024.

[2] D. A. ROBERTSON and E. L. COX, "Steering Away from Automotive HMI De Facto Standards - The Effect on Control Discoverability | Proceedings of the 16th International Conference on Automotive User Interfaces and Interactive Vehicular Applications," *Steering Away from Automotive HMI De Facto Standards - The Effect on Control Discoverability | Proceedings of the 16th International Conference on Automotive User Interfaces and Interactive Vehicular Applications,* 22 setembro 2024.

[3] D. ZHAO, C. WANG, Y. LIU e T. LIU, “Implementation and Evaluation of Touch and Gesture Interaction Modalities for In-vehicle Infotainment Systems,” *Implementation and Evaluation of Touch and Gesture Interaction Modalities for In-vehicle Infotainment Systems,* 28 novembro 2019.

[4] D. Quiñones, L. F. Rojas e A. Barraza, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S187705092401175X.