The paper “**Clairvoyance: Smart multimodal assistant for the visually impaired**” is almost finished and will be published **next week**.

In the meantime, the document below provides a detailed report of the work conducted during this project.

Final Project Management Report

PIE Project – **Clairvoyance**

Github Repo: https://github.com/puushtab/clairvoyance

Academic Year 2024–2025

ENSTA

PIE Team Clairvoyance
Ayoub B., Amine M., Gabriel D.,
Hussein R., Ismail E., Lara P.,
Louis M., Luana G., Yasmin V.,
Supervisor: Mr. Franck Taruffi

Contents

1 Introduction

1.1 Project Context

Visually impaired individuals encounter numerous obstacles in their homes when performing daily tasks, a situation aggravated by a progressive loss of autonomy. These difficulties reduce their independence and negatively impact their quality of life, particularly when they must adapt to technologies often perceived as complex. According to INSERM, **1.7 million people in France suffer from visual impairment, including 1.3 million elderly people losing autonomy**. These figures highlight a growing issue, raising a major challenge: designing suitable, accessible solutions that respect specific needs to restore functional autonomy.

Many technologies have been developed to support people with disabilities, but they are frequently judged as unsuitable, intrusive, or difficult to use. Current devices, while innovative, struggle to offer an intuitive interface that is quick to master and personalized for daily use. This observation reveals an urgent need for solutions that not only compensate for functional limitations but also integrate harmoniously into the users' domestic environment.

The ***Clairvoyance*** project aims to address these issues by developing a pair of connected glasses equipped with a camera, a microphone, and a speaker, assisted by artificial intelligence. This innovative device will describe the visual environment to the user in real-time, while being controllable via voice commands. Furthermore, it will integrate with connected home objects to offer advanced functionalities, such as locating objects or controlling appliances. Beyond its technological aspect, ***Clairvoyance*** aspires to propose an accessible interface and restore true autonomy to visually impaired people by placing their needs at the heart of the design.

Table 1: Context and challenges for visually impaired people

Aspect	**Description**
Target population	1.7 million people with visual impairment in France, including 1.3 million elderly people losing autonomy (INSERM).
Main challenges	Loss of autonomy, difficulties in daily tasks, adaptation to complex technologies.
Limitations of current solutions	Unintuitive interfaces, lack of personalization, insufficient integration into the home environment.
Clairvoyance project goal	Offer an accessible, intuitive, and integrated solution to restore autonomy via AI-assisted connected glasses.

Figure 1: Use of Clairvoyance glasses in a home environment

1.2 Project Objectives

This project aims to meet concrete needs through innovative and accessible technical solutions. The defined objectives guide development and ensure alignment with user expectations.

- **Create a functional product**: Using LEAN and AGILE methods, the project adopts an iterative approach to develop and progressively improve functionalities. This ensures continuous adaptation to user feedback.
- **Start from the needs of potential clients**: An in-depth analysis of end-user needs guides technical and functional choices, ensuring the product effectively meets their expectations.
- **Design a low-cost, low-tech, and accessible product**: The goal is to develop an affordable solution, emphasizing simplicity and efficiency, to reach a broad population, including people with limited resources, targeting a market different from pre-existing solutions.
- **Develop a modular and adaptable product**: The product's modularity will allow personalization for different types of disabilities, increasing its relevance for a variety of users.
- **Minimize adaptation time for the elderly**: An intuitive interface and ergonomics designed for users unfamiliar with recent technologies will facilitate quick adoption.
- **Promote an open-source project**: By making technical resources (code, designs, etc.) available, the project encourages collective innovation and facilitates improvement or adaptation by other actors.

1.3 Methodology and Approach

The *Clairvoyance* project adopts an Agile method (iterative and flexible), including Scrum, Kanban, and Extreme Programming, combined with classic tools like WBS and the FAST diagram to structure the V-Model. This approach is supported by Notion for organization and tracking. The tools used are described below:

Table 2: Description of tools used in the Clairvoyance project

Tool	**Description**
Scrum	Management by sprints with defined roles (Product Owner, Scrum Master, Team).
Kanban	Visual workflow management via a Kanban board.
Extreme Programming	Practices for continuous code improvement (tests, pair programming).
WBS	Hierarchical structure of tasks to organize the project.
FAST Diagram	Representation of functions to structure the V-Model.
Notion	Collaborative tool to manage sprint backlog, Kanban, meetings, and tracking.

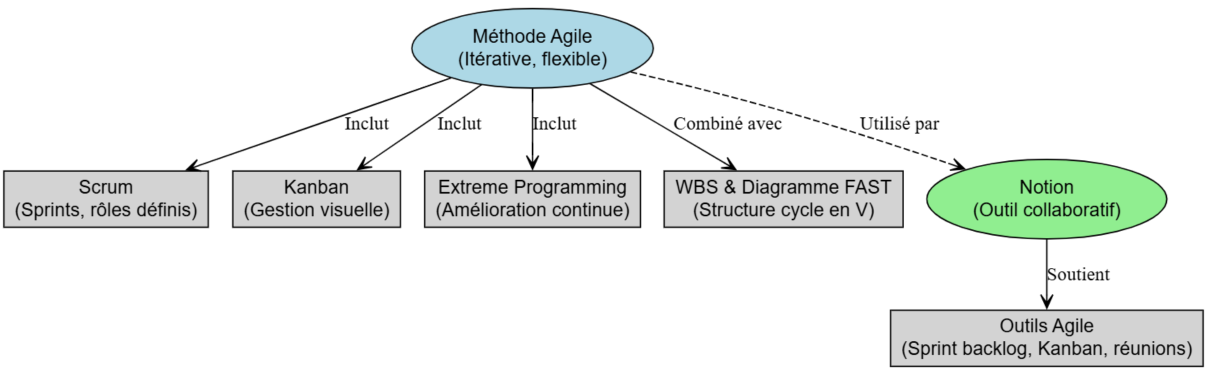


Figure 2: Tools and methods used

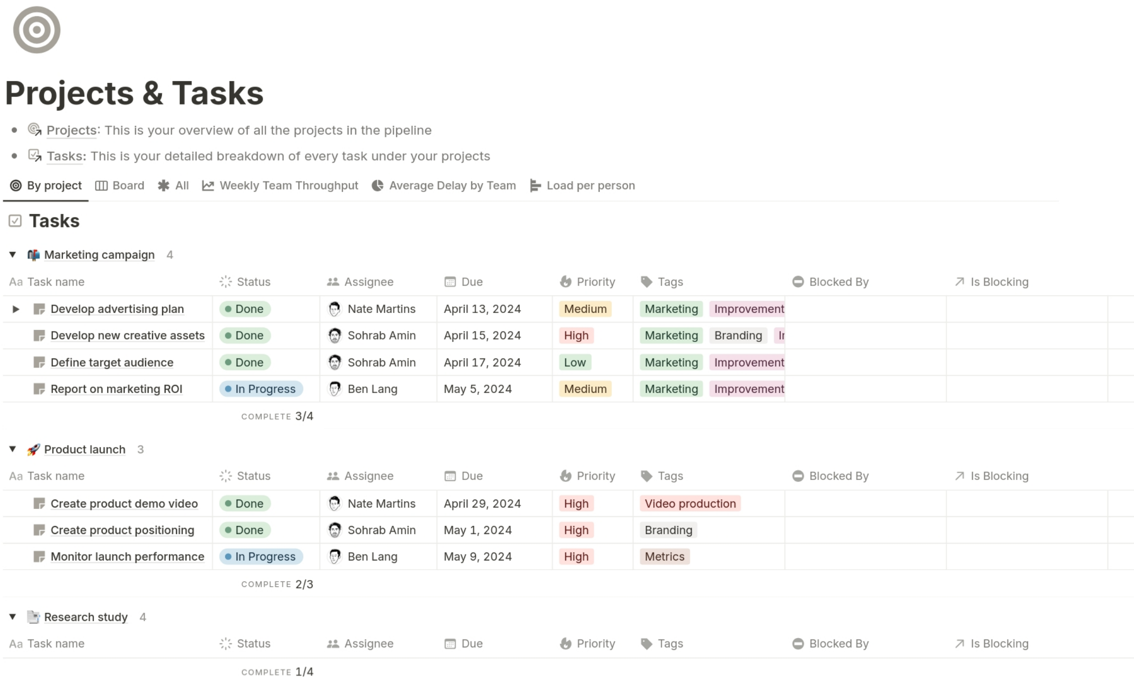


Figure 3: Notion Platform

2 Need Description, Objectives, Scope & Stakeholders

2.1 Description of Needs

Visually impaired people face major challenges in their daily lives, such as loss of autonomy, difficulties in performing domestic tasks, adapting to complex technologies, and a decline in their quality of life. These obstacles, exacerbated by a population of 1.7 million people with visual impairment in France (INSERM), require adapted solutions. The *Clairvoyance* project identifies the following needs:

- **Restore autonomy**: Allow users to manage their daily tasks independently through a real-time description of their environment.
- **Facilitate interaction**: Propose an intuitive interface via voice commands, adapted to all levels of technological proficiency.
- **Integrate the solution into the home**: Connect the glasses to domestic objects to locate items or control devices.
- **Improve quality of life**: Reduce frustration by offering an accessible and non-intrusive solution, harmoniously integrated into their daily life.

These objectives aim to transform the experience of visually impaired people by offering a practical and human solution, while meeting their specific needs.

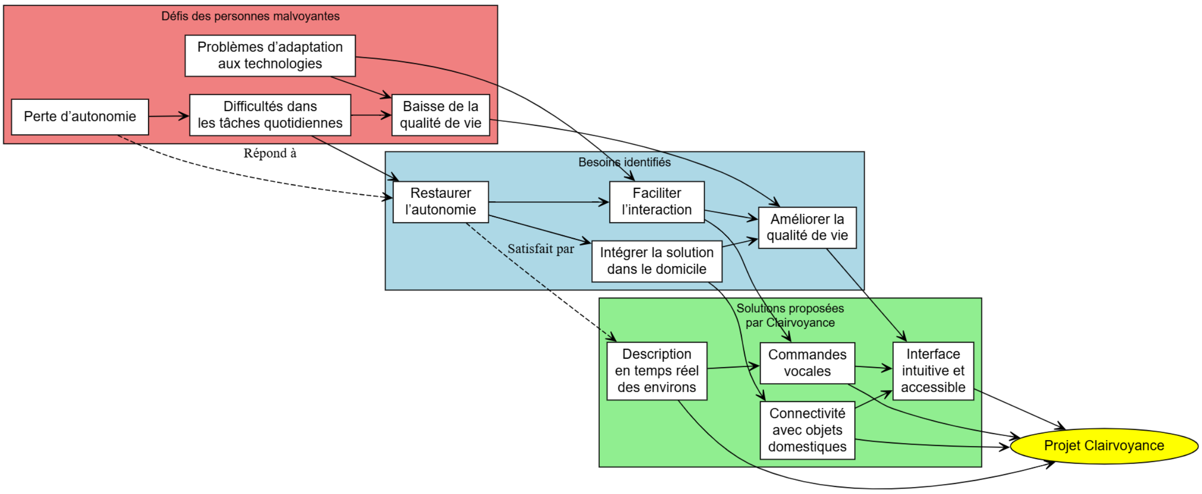


Figure 5: Description of need linked to the proposed solution

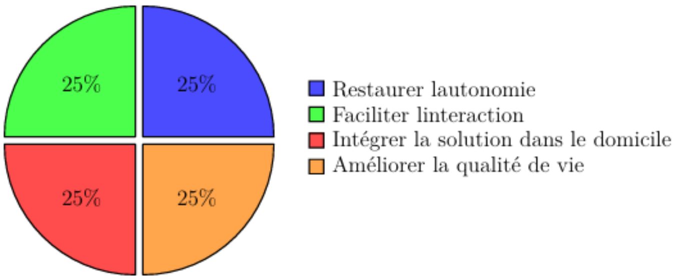


Figure 4: Distribution of objectives by importance

To address these needs, *Clairvoyance* proposes connected glasses with artificial intelligence, capable of describing the environment, being controlled by voice, and integrating with connected objects, thus offering a user-centered solution.

2.2 Identification of Needs and Product Backlog Elaboration

At the beginning of the project, our team participated in several collective reflection sessions (brainstorming) to identify the potential needs of end users. This phase was complemented by the distribution of a form addressed to relevant associations, with the aim of gathering concrete feedback on the daily needs of the people targeted by our solution. The information collected during these two steps allowed us to build an initial *product backlog*. This document, which groups all envisaged functionalities as well as associated priorities, constitutes the basis for defining the project requirements. The *product backlog* presented below is divided into several categories representative of the expressed needs, and has not been modified since its creation to preserve a faithful record of the initial ideation phase.

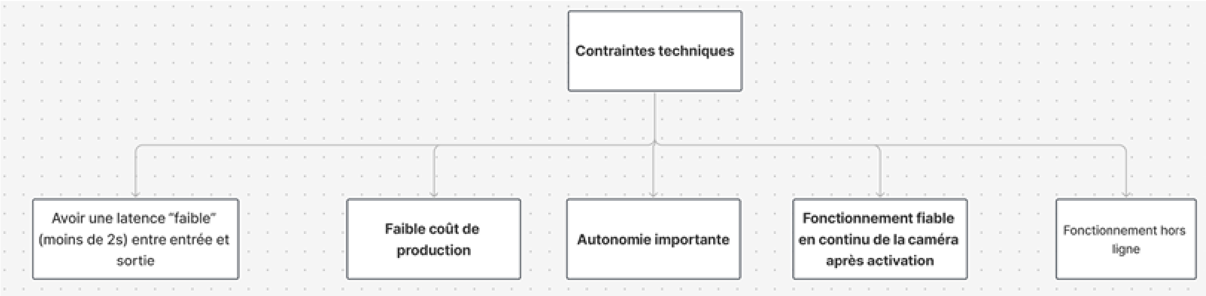


Figure 6: Product Backlog of technical constraints

Figure 7: Product Backlog of textual and voice management

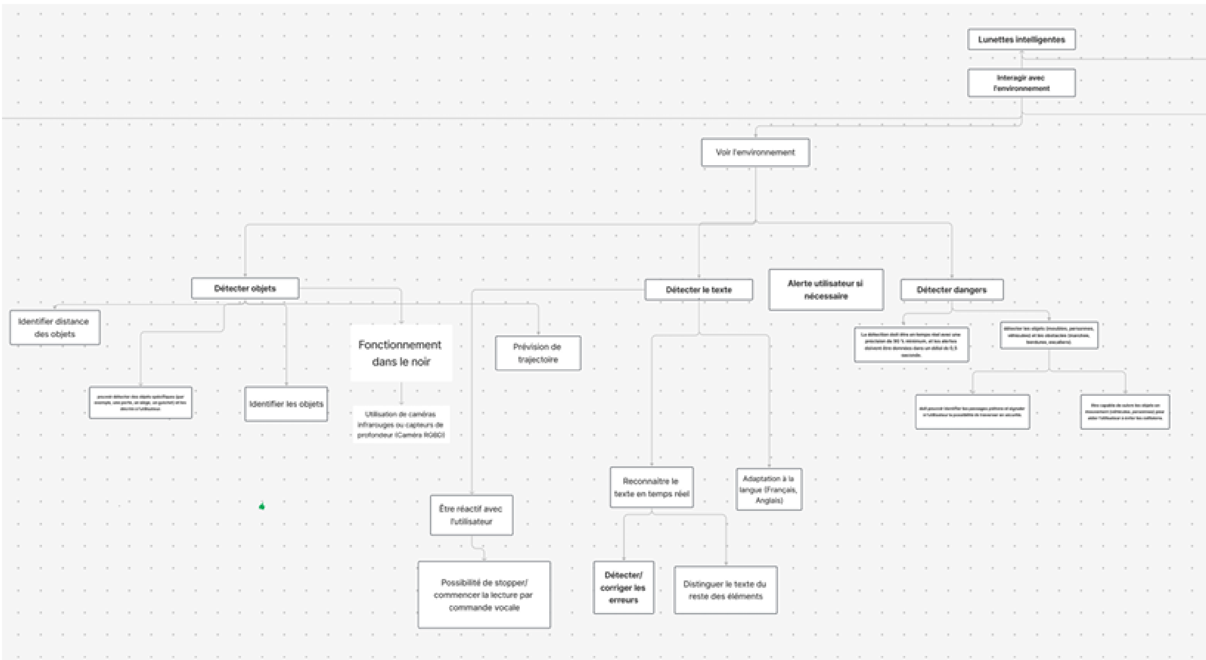


Figure 8: Product Backlog of vision

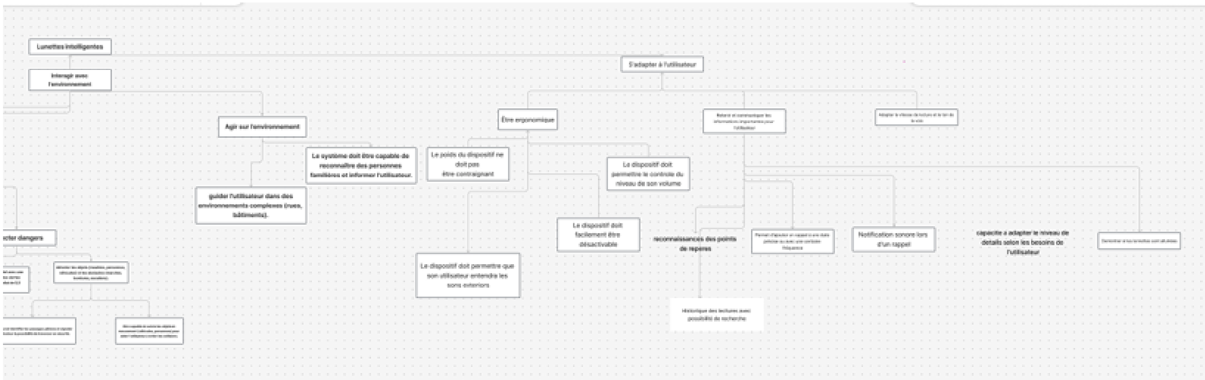


Figure 9: Product Backlog of adaptability constraints

2.3 Project Stakeholders

The realization of the project relies on the involvement of several actors with complementary roles:

- **Project Team**: Composed of students, it is responsible for the design, development, and implementation of technical solutions.
- **ENSTA Paris**: The school offers an environment conducive to the project's realization thanks to its infrastructure, financial support, and pedagogical supervision.
- **End Users and Associations**: Visually impaired people and their representatives constitute a valuable source of feedback. Their needs and expectations guide the design of a relevant and accessible product.
- **Project Management Supervision**: Ms. Laure Letellier accompanies the team in the structuring, organization, and tracking of the project, providing her methodological expertise.
- **Fablab of l'X**: This laboratory provides prototyping tools necessary for the manufacture and testing of the first versions of the product.
- **Sector Start-ups**: Studying similar existing solutions allows identifying areas for improvement and considering potential collaborations.

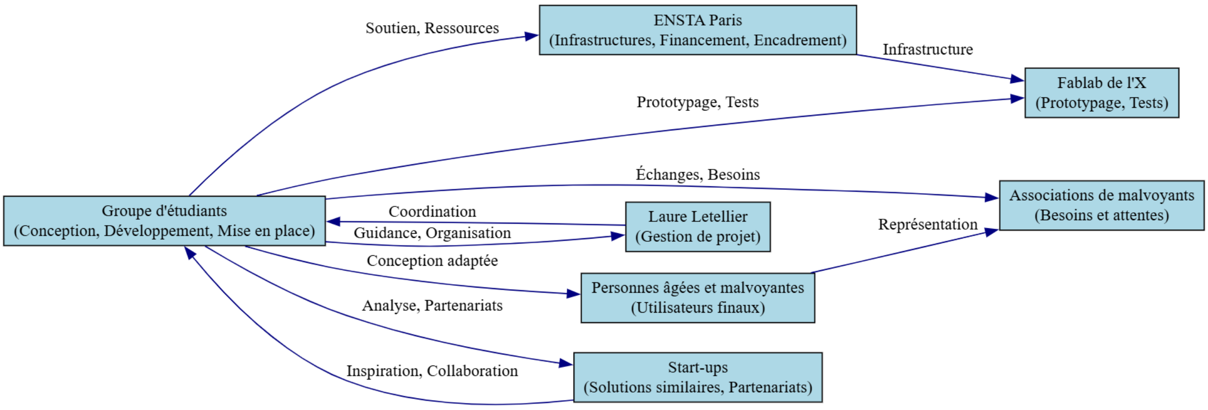


Figure 10: Description of stakeholders

2.4 Requirements & Evolution

At the beginning of our project, we established requirements regarding our product and classified them into different categories. These then allowed us to create our FAST diagram as well as a coverage matrix to know how to break down our tasks according to product functions. These have evolved throughout the project, and below is the table of requirements with their status in the final prototype and the explanation. The color code is validated, close to validation, not achieved or modified.

Requirement ID	Description	Evolution
R1	The glasses must be able to turn on and off by voice	Abandoned due to consumption and logistical problems
R2	The glasses must be able to turn on and off via power button	Turning on Arduino and application
Hardware		
R3	The glasses must be lightweight (<400g)	Current weight distributed: 50g
R4	The weight of the glasses must be distributed evenly on the skull	Conclusive user test
R5	The glasses must have a power button	Arduino power on
R6	The glasses must not obstruct the view of a visually impaired person	Camera placed on the forehead
R7	The glasses must be equipped with an RGBD camera	Abandoned in favor of monocular depth detection technology to save budget
R8	The glasses are powered by an internal battery	On-board Li-ion battery

Alerts		
R9	The glasses must alert the user of their charge status	Color indicator on the battery
R10	The glasses must alert the user of their operating mode	Audio feedback implemented
R11	The glasses must alert the user of an upcoming collision	Collision detection on demand, not in real-time
R12	The glasses must alert the user of a dangerous object nearby	Elaborate classification system combined with text reading
Perception of the Environment		
R13	The glasses must be able to see the environment	Operational camera
R14	The glasses must perceive depth	Depth data obtained with acceptable precision according to benchmarks by MiDaS technology
R15	The glasses must classify objects by type and priority order	Functional YOLO classification algorithm
Interaction with the User		
R16	The glasses must be able to communicate with the user in French with a human voice	Integrated voice synthesis but more effective in English
R17	The glasses must be able to describe the environment on command	Voice command not yet active
R18	The glasses must be able to refine the description of the environment on command	Description satisfactory but improvable according to our dataset
R19	The glasses must be able to read text on command	OCR module successfully tested and benchmarked with suitable precision
R20	The glasses are connected to the phone	Dependency problems led to rethinking the entire application for mobile, choice of a remote PC application ported to mobile. Existence of a semi-functional mobile prototype.
Financial and Technical Constraints		
R21	The cost of building the glasses must not exceed the budget assigned by ENSTA	Budget respected to date
R22	The glasses must not have a latency greater than 3 seconds	Average latency: 3.2 s

R23	The glasses must be able to function offline	Choice of Client-Server operation corresponding to domestic usage
R24	The glasses must have an autonomy greater than 30h	Current autonomy: 20h, optimization possible
R25	Sound control must be possible by voice	Sound control on the computer
R26	The glasses must function in the dark	Abandonment of this function to simplify the application

3 Project Organization, Risks & SWOT

3.1 Team Structure & Roles

3.1.1 Organizational Chart

Our team consists of 10 members, 8 from the STIC track and 2 from Applied Mathematics. This diversity gives us the necessary skills to design and prototype an ambitious project. Most members possess solid experience in language models, machine learning, and collaborative work, valuable assets for the project's success. Initially, we adopted the Agile method to organize our work, distributing responsibilities among all members and dividing the team into two groups aligned with our product's functionalities: the Textual Processing group and the Image Processing group. Each group was led by a head in charge of coordinating tasks and communicating progress to the project manager, who assumed the role of Scrum Master.

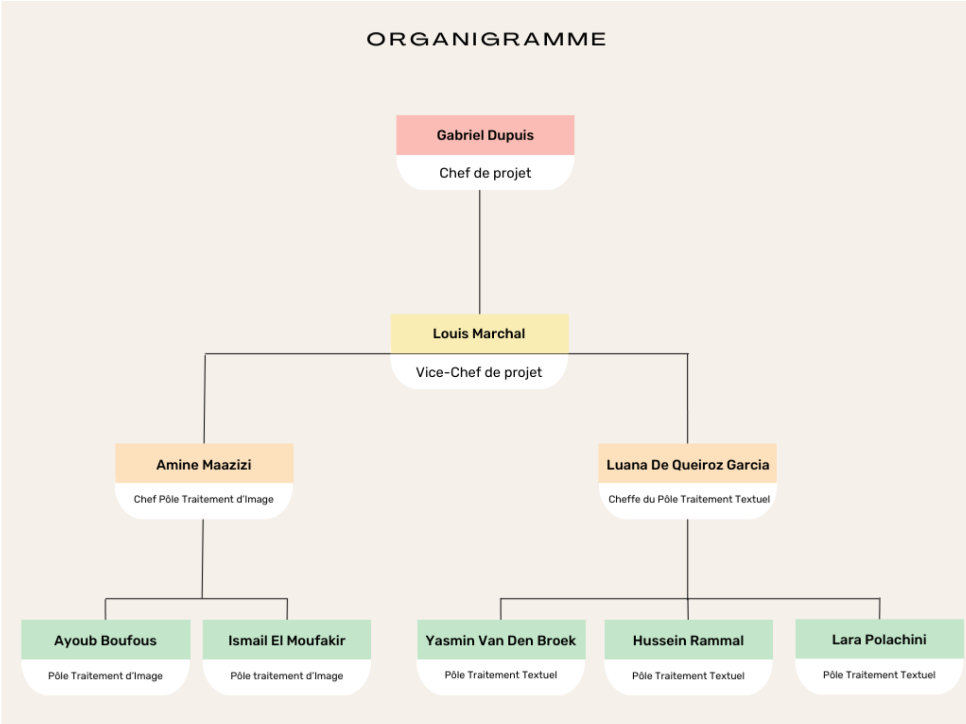


Figure 11: Organizational Chart 1: Initial Team Organization

After developing our first operational MVP, new responsibilities emerged, requiring a team reorganization. These new missions included the Hardware group, the Software group, and an Agentic group acting as the "brain" of the project. The latter integrates decisions, rules, and the use of Large Language Models (LLM) to optimize the use of tools available to the client. Similarly, each group was led by a head in charge of coordinating tasks and transmitting progress to the project manager, who assumed the role of Scrum Master. We also eliminated the position of vice-project manager, as direct communication with the project manager proved sufficient for the scale of this project.

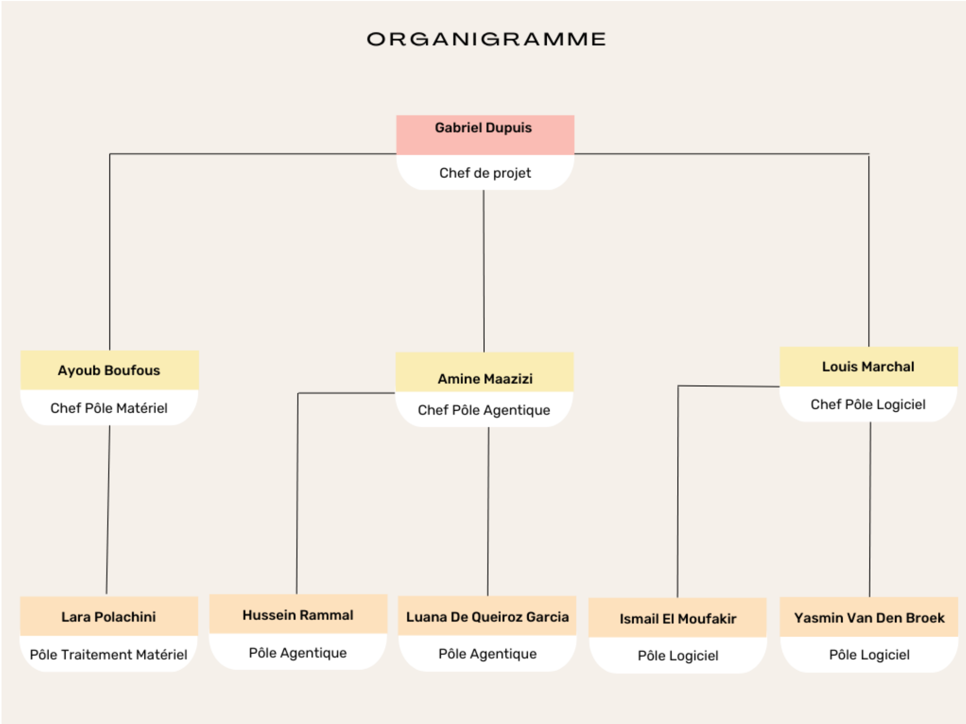


Figure 12: Organizational Chart 2: Organization after mid-term

3.1.2 RACI Matrix

As specified in the team presentation section, our group was divided into sub-groups with managers for each of the six parts of the project represented above. This allowed us to create a RACI Matrix and assign responsibilities on our Notion workspace. However, this matrix remains flexible throughout the project, as exchanges between sub-teams are very frequent in the Agile method.

Table 4: RACI Table of Activities (Periods 1 and 2)

Activity	Amine Maazizi	Gabriel Dupuis	Louis Marchal	Ayoub Boufous	Ismail El Moufakir	Hussein Rammal	Luana Garcia	Yasmin Van Den Broek	Lara Polachini
Period 1									
Determine when to capture user voice	I	I	I	I	I	I	A/R	C	R
Process audio to isolate voice	I	I	I	I	I	C	A/R	R	R
Translate user voice to text using pre-existing tools	C	I	I	I	I	C	A/R	R	R
Interpret and extract relevant information from user inputs using an LLM	C	I	I	C	I	R	A/R	R	I
Process the JSON file from input processing using an LLM to extract an environment description text by priority	C	A	C	R	I	I	R	R	R
Process response text into human voice audio file	I	I	I	I	I	C	A/R	R	R
Determine when to retrieve video	R	A	R	R	R	R	R	R	R
Identify objects on video using YOLOv11	C	A	C	R	R	C	I	I	I
Extract information from object detection	A/R	I	C	R	R	C	I	I	I
Couple information with depth detection	A/R	I	C	R	R	C	I	I	I

Table continued on next page

Activity	Amine Maazizi	Gabriel Dupuis	Louis Marchal	Ayoub Boufous	Ismail El Moufakir	Hussein Rammal	Luana Garcia	Yasmin Van Den Broek	Lara Polachini
Establish a means of file transmission to text management	R	A/R	R	R	R	R	R	R	R
Adapt to real-time detection	A/R	I	C	R	R	C	I	I	I
Detect depth using AI model without special camera	A/R	R	I	C	C	C	I	I	I
Handle inputs from RGBD camera	A/R	I	C	R	R	C	I	I	I
Extract text from a photo using OCR tool	A	I	C	C	R	C	I	I	I
Adapt OCR to video	A	I	C	C	R	C	I	I	I
Correct errors on text	A	I	C	C	R	C	I	I	I
Manage real-time text reading	A	I	C	C	R	C	I	I	I
Design a 3D model of glasses with global component insertion scheme	I	I/R	I	I	I	C	A	R	R
Manage weight distribution	I	I/R	I	I	I	C	A	R	R
Choose embedded electronics	I	I/R	I	I	I	C	A	R	R
Order necessary electrical components	I	A/R	I	I	I	C	I	R	R
3D print glasses prototype	I	A	I	I	I	R	A	C	C
Assemble glasses by implementing pre-existing software	R	R	A/R	R	R	R	R	R	R
Test each code part elementarily	R	R	A/R	R	R	R	R	R	R
Test communications between different software parts	R	R	A/R	R	R	R	R	R	R
Test operation on phone	R	R	A/R	R	R	R	R	R	R
Test glasses model ergonomics	R	R	A/R	R	R	R	R	R	R
Test final prototype by giving it to users	R	R	A/R	R	R	R	R	R	R
Elaborate different possible commands	C	R	A/R	C	C	C	C	C	C
Establish a hierarchy of possible commands	C	R	A/R	C	C	C	C	C	C
Plan different meetings	I	A/R	R	I	I	I	I	I	I
Take into account the progress of each group	R	A/R	R	I	I	I	R	I	I
Provide feedback on each sprint for product integration improvement	R	A/R	R	R	R	R	R	R	R
Period 2									
3D printing of camera support	I	I	I	A/R	I	I	I	I	R
Order and retrieve hardware	I	I	I	A/R	I	I	I	I	R
Create FastAI code anticipating server responses	I	I	I	A/R	I	I	I	I	R
Make code work on hardware	I	I	I	A/R	I	I	I	I	R
Integrate call, email, SMS features into app	I	R	R	I	A/R	I	I	I	I
Perform unit tests of features	I	R	R	I	A/R	I	I	R	I
Integrate features into app	I	A/R	R	I	R	I	I	R	I
Migrate features to a server	I	R	A/R	I	R	I	I	R	I
Ensure features work on server	I	R	A/R	I	R	I	I	R	I
Create a database of scenarios envisaged by the project	A/R	I	I	I	I	R	R	I	I
Train neural network models on these scenarios	A/R	I	I	I	I	R	R	I	I
Integrate an agent to interpret unforeseen scenarios	R	I	I	I	I	A/R	R	I	I
System integration into application	A/R	I	I	I	I	R	R	I	I
Perform application unit tests	R	A	I	R	R	I	I	I	I
Perform application integration tests	R	A	R	R	R	I	I	I	I
Realization of carbon footprint assessment	I	I	I	I	I	I	I	A/R	I
Create GDP report	R	A/R	I	I	R	I	R	I	I
Film project demo video	R	R	A/R	R	R	I	I	I	I
Perform video editing	I	I	A/R	I	I	I	I	I	I

3.2 Functional Decomposition

The FAST (Function Analysis System Technique) diagram is a tool that decomposes project functions to clarify objectives and structure development steps. Based on our Product Backlog, we were able to break down the various essential functions of our product into a FAST diagram, which provides a response to each of the requirements found in the previous step.

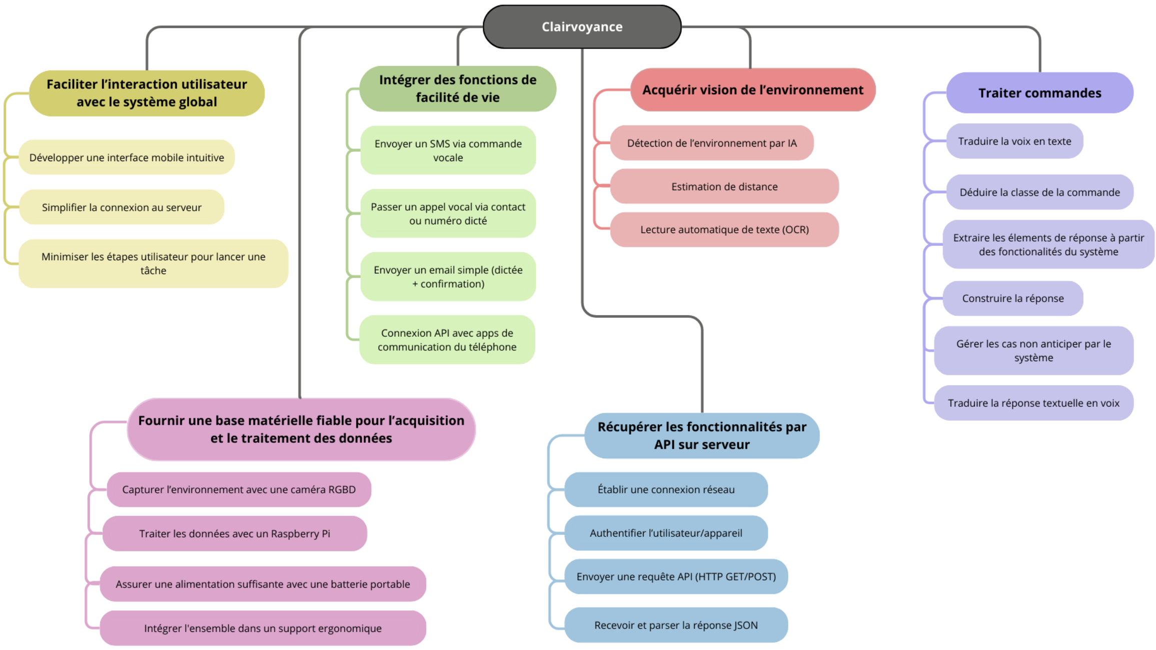


Figure 13: FAST Diagram

3.3 SWOT Matrix

During the first period, we identified the following elements in our SWOT analysis.

Strengths	Weaknesses
• Advanced competence in software development and knowledge of effective group tools (Github, Discord, Notion) • Motivated team for a concrete project • Funding from ENSTA Paris and Fablab of l'X • Agile Organization	• Large team • Absence of benchmarks from previous groups or supervisor to move forward • Conflicting exams and schedules
Opportunities	**Threats**
• Possibility to start from need by directly contacting visually impaired and elderly people • Recent "low-tech" market and need for affordable product for elderly people • Recent availability of very high-performance and open-source artificial intelligence tools	• Loss of motivation, team scattering • Product not adapted to user need • Product too ambitious and non-functional

During the second period, our project reached a decisive stage with the achievement of a functional **MVP** (Minimum Viable Product). The main objective then became to **complete, stabilize, and make this MVP usable**, in a much more constrained context than during the first phase. Time constraints (internship search, exams), logistical constraints (materials not available on time), and human constraints (lack of response from target associations) strongly impacted our ability to iterate quickly. Below is a new SWOT matrix adapted to this new phase, compared to the previous version.

Strengths	Weaknesses
• Operational MVP serving as a clear basis for development. • Experience acquired during the first phase (code, integration, project management). • Skill upgrading on hardware, embedded AI, and application integration. • Clear task distribution with continuous update of the RACI matrix.	• Very strong time constraints due to personal deadlines (exams, internships). • Dependence on hardware that is difficult to access or broken. • Lack of user feedback (few or no returns from contacted associations). • Personal burdens (financial, logistical) outside the initial budget.
Opportunities	**Threats**
• Integration of new features (calls, intelligent scenarios, interpretation outside foreseen cases). • Opportunity for field tests in the final phase. • Demo video to showcase the product. • Potential to generalize the system to other audiences (blind, seniors, mild cognitive impairment).	• Blockages linked to unavailability or delay of key components. • Potential flaws in application reliability if not tested enough. • Loss of motivation at the end of the project linked to accumulated fatigue.

Key Shifts

- **From exploration to implementation**: The first period was centered on ideation, R&D, and prototyping. The second period was more demanding in terms of concrete production, testing, and reliability.
- **From agility to constraint management**: While agility favored experimentation in the initial phase, material, logistical, and academic reality imposed more rigid management.
- **From enthusiasm to resilience**: Initial energy was partly consumed by delays, but the team showed adaptability and professionalism to achieve a coherent final product.

3.4 Risk Analysis

Risk analysis is a key step in project management, allowing anticipation of potential obstacles and definition of strategies to mitigate them. Based on a rigorous methodology (detailed in the Appendix), this analysis was carried out using collaborative workshops within the team. Each identified risk was evaluated according to two main criteria: probability of occurrence (rated 1 to 5) and potential impact (rated 1 to 5). The criticality of each risk was then calculated to prioritize necessary actions, ensuring better mastery of uncertainties and improvement of project resilience. **The entire table and risk analysis can be found in the appendix so as not to hinder the reading of the report.**

Several risks anticipated during the initial phase of the project did not ultimately materialize, thanks to the measures put in place:

- **Software/Hardware integration failure**: Initially judged critical, this risk was mastered thanks to a modular architecture and frequent exchanges between technical sub-groups. Progressive integration tests helped avoid major blockages.
- **Project manager overload**: Although the centralized role carried a bottleneck risk, the implementation of stand-up meetings by sub-group and the empowerment of referents helped fluidify daily management.
- **Component incompatibility**: This risk was avoided by validating each component before purchase (thanks to preliminary tests in sprint 2), and by orienting towards simple and compatible choices (like abandoning the Bluetooth module).
- **Poor workload distribution**: Regular adjustments of sprints and role flexibility allowed effective effort distribution, thus avoiding significant imbalances or task abandonment.

The anticipation of these critical points and the team's reactivity ensured project continuity without rupture, even during periods of tension or material delay.

3.5 Lessons Learned on Organization

During the *Clairvoyance* project, the organization of our ten-member team was put to the test by the complexity of a one-year engineering project, combining technical, logistical, and human challenges. By adopting an Agile approach and relying on tools like Notion, Scrum, and the RACI matrix, we drew several valuable lessons on managing a large team and coordinating an ambitious project. These lessons, stemming from our collective experience, are summarized below:

- **Agility requires rigorous discipline**: The Agile approach, with its sprints and Kanban board, allowed us to iterate quickly, but its effectiveness depended on strict meeting planning and constant backlog updating. Without this discipline, we risked delays, as observed during the second period's constraints (exams, internships).

- **Communication is the pillar of coordination**: Regular meetings and the use of Notion reinforced transparency, but internal communication problems (Risk-10-Team) sometimes slowed progress. We learned that proactive communication, with clear minutes, was essential to align a large team.
- **Role flexibility improves resilience**: The RACI matrix, although structured, benefited from flexibility in task assignment. Reorganization after the first MVP, with the creation of new groups (Hardware, Software, Agentic), allowed better response to emerging needs, reducing the impact of turnover (Risk-11-Team).
- **Work overload threatens motivation**: Time and logistical constraints of the second period (Risk-32-Team, Risk-33-Team) revealed the importance of distributing tasks fairly and recognizing individual efforts. Team-building sessions helped maintain cohesion in the face of accumulated fatigue.
- **Anticipation of logistical risks is crucial**: Material delivery delays (Risk-09-Delay) and unforeseen breakdowns (SWOT weakness) taught us to plan alternative solutions, such as identifying multiple suppliers or testing components upon receipt.
- **Involvement of external stakeholders is difficult but necessary**: Lack of feedback from associations for visually impaired people (SWOT weakness) limited user validation. We understood the importance of diversifying external communication channels and planning these interactions from the start.
- **Transition from ideation to production requires organizational adaptation**: Moving from exploration (first period) to implementation (second period) required more rigid constraint management. Removing the vice-project manager position and focusing on the project manager simplified decision-making.
- **Managing expectations maintains coherence**: Scope creep (Risk-08-Delay) taught us to clearly define the MVP perimeter and regularly communicate on priorities to avoid unplanned additions.
- **Collective skill upgrading strengthens the team**: The diversity of profiles (STIC and Applied Mathematics) and experience acquired in embedded AI and prototyping (SWOT strength) allowed the team to overcome complex technical challenges, such as FastAI code integration or 3D printing.
- **Resilience in the face of constraints forges unity**: Despite academic and logistical pressures, the team's adaptability, supported by a common vision of the project (restoring autonomy to the visually impaired), transformed obstacles into opportunities for learning and cohesion.

4 Synthetic Presentation of the Technical Solution

4.1 Functional Technical Specifications

Within the framework of this project, we developed a prototype of smart glasses intended to assist visually impaired people in understanding their immediate environment. The main objective is to propose an accessible technological solution, capable of providing essential information on the user's close environment, in a simple and intuitive way. These are not traditional glasses, but rather *pseudo-glasses*: a device composed of a camera and a microphone, fixed on a glasses frame and connected to both a distant server and a mobile phone. The camera allows capturing images of the environment, while the microphone can be used for vocal interactions. Data processing is performed on the server, and results are transmitted to the user via the phone. The predefined functionalities of the system are as follows:

- **Environment detection**: The camera analyzes objects and elements present around the user to provide a description of their surroundings.
- **Text reading**: When a text is placed in front of the user (for example, a sign, a document, or a label), the system is capable of recognizing it and retransmitting it to the user in audio form.
- **Distance estimation**: The device can detect objects located in front of the user and estimate their distance, which helps avoid obstacles and facilitates safe movement.

In addition to these predefined functionalities, our device also integrates so-called improvisation functions, allowing for more flexible and personalized interaction between the user and the system. The system's improvisation functionalities are as follows:

- **Voice assistant**: The user can ask free questions through the microphone to create a sort of conversation with the glasses. This allows, for example, asking for details about the environment, contextual information, or executing certain voice commands.
- **Interconnectivity**: The system is interconnected with the user's mobile phone functions. Thus, it is possible to make calls, send SMS or emails directly by voice command, offering greater autonomy and better integration into daily life.

Thanks to these functionalities, our solution aims to improve the autonomy and quality of life of people with visual impairment by bringing them practical help daily.

4.2 Global Architecture

Thanks to the simplified vision image (figure 14), we can visualize data flows between the front-end and back-end. We observe that the user gives a vocal instruction, which is then received by the application. The application sends it to the Google Cloud API

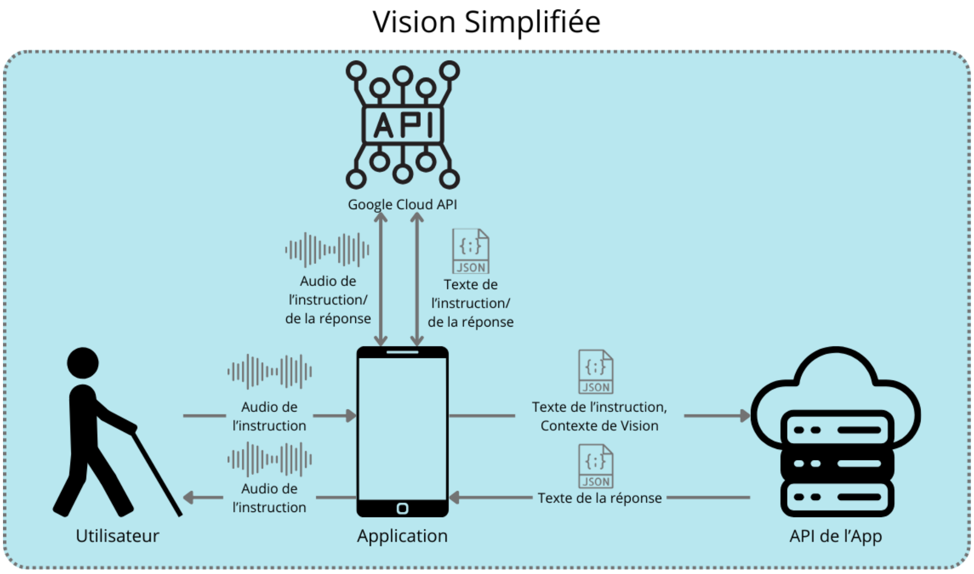


Figure 14: External view of the application

to obtain a text transcription. The transcribed text is then transmitted to the application API. After response generation by the application API, a textual response is sent to the application, which transmits it to the Google Cloud API to generate the audio response intended for the user.

4.3 Hardware (HW)

For the realization of the prototype, we selected lightweight, compact hardware components compatible with each other to ensure fluid and functional integration on a wearable frame. The system base consists of a Raspberry Pi 5, chosen for its processing performance adapted to embedded tasks and its extensive connectivity. It is protected by a specific case, ensuring both support and cooling. Power is provided by a 1300mAh 3S FPV battery, offering reasonable autonomy while maintaining a reduced format. A 32GB Lexar SD card is used for operating system and data storage. To avoid overheating, a 5V fan is integrated into the transparent case, ensuring thermal dissipation for the Raspberry Pi. A voltage converter (330V to 5V - *Note: Likely 11.1V/12V to 5V in reality based on 3S battery, but translating source text*) is used to guarantee stable power supply to components from the battery. The device is mounted on a GoPro strap, allowing the assembly to be fixed comfortably and stably on the user's head. Finally, visual capture is ensured by a 3-camera module for Raspberry Pi, allowing wider or multi-angle vision, essential for environment recognition and scene analysis functionalities. We can see the exemplification of hardware components in figure 15 and how each component will be positioned.

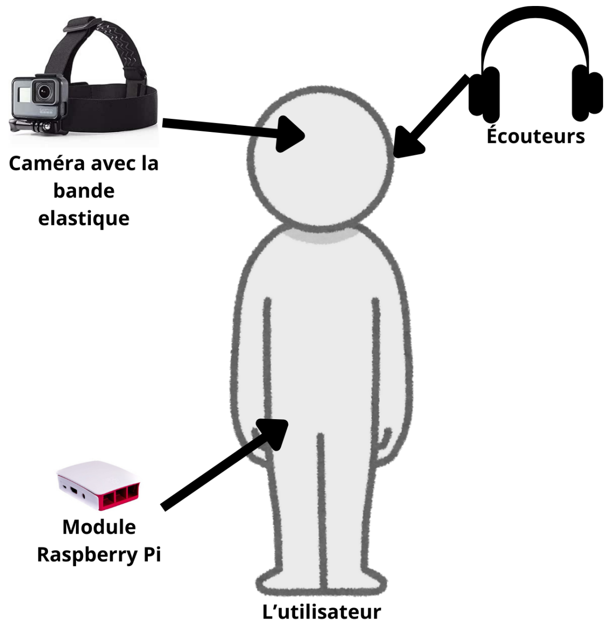


Figure 15: Hardware x User Relation

Component	Function
Raspberry Pi 5	Embedded central processing unit
Raspberry Pi Case	Physical protection and component fixation
FPV Battery 3S 1300mAh	Portable power source
Lexar SD Card 32GB	System and data storage
5V Fan	Raspberry Pi cooling
Converter 330V → 5V	Voltage regulation for stable power supply
GoPro Strap	Portable mounting support for the user
Raspberry Pi 3-camera Module	Visual capture for environment analysis

Table 5: Hardware components of the prototype and their functions

4.4 Software (SW)

Image 16 illustrates the detailed operation of the system by highlighting data exchanges between different components. When a vocal instruction is given by the user, the application first receives its textual transcription (obtained via Google Cloud API). Then, it captures camera data corresponding to the user's visual environment. The application starts by interpreting the textual instruction to associate it with a command class. If this class is identified as an improvisation function, the request is transmitted to the Agentic model, designed to manage this type of interaction. Otherwise, if the instruction corresponds to a predefined command, it is transmitted to the LLM model, responsible for processing these specific functions. Predefined commands include:

- environment description,

Vision detaillée

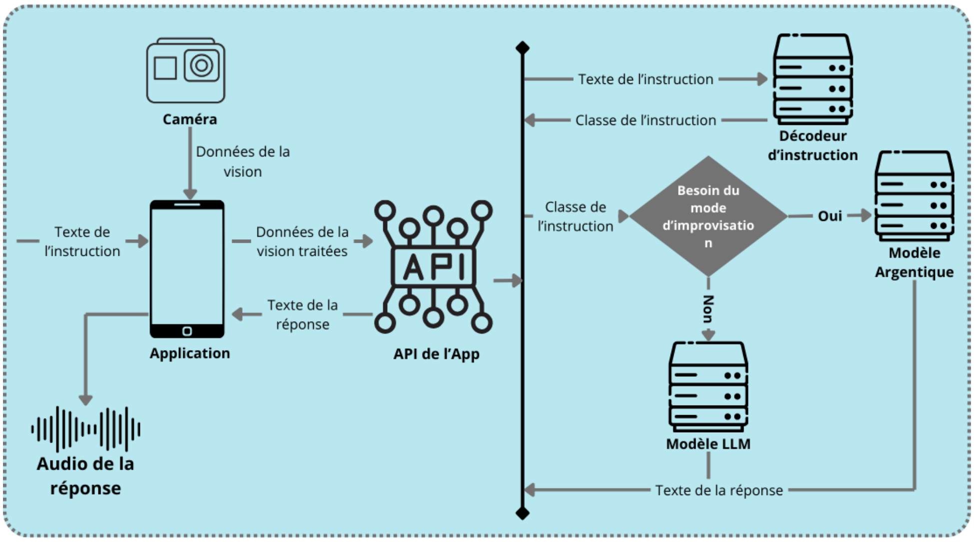


Figure 16: Internal view of the application

- text reading,
- depth detection,
- sending emails,
- sending SMS,
- and making calls.

In all cases, the concerned model generates a textual response, which is sent back to the application. Finally, this response is converted to audio via Google Cloud API and played back to the user.

4.5 Tests & Validation

To guarantee system reliability, we adopted an incremental approach for the testing phase. Each module was first tested separately, in a controlled environment, to validate its proper functioning in isolation. Thus, we performed unit tests on the three major components of the project: the language module (voice processing and response generation), the vision module (image capture and interpretation), and the mobile module (communication between the application and external services via smartphone). Once each module was individually validated, we proceeded to their progressive integration into a complete functional prototype. This step allowed us to detect potential interoperability conflicts between components and adjust communication interfaces. Throughout the process, we emphasized the use of open-source modules and accessible technologies, always aiming for minimal processing time to guarantee a fluid experience for the user. This performance constraint guided our technical choices, particularly regarding the libraries used and the software structure of the application. In addition, we also performed benchmarking of key functions (voice recognition, image description, voice synthesis, etc.) to compare several alternatives and choose the most performant solutions in terms of speed, precision, and compatibility with our architecture. These comparative tests played a determining role in the selection of final tools used in the prototype.

4.6 Configuration Management and Traceability

To ensure good configuration management and facilitate development traceability, we used GitHub as the central platform for code management and task coordination. Each team member worked in personal branches, which allowed developing and testing functionalities in isolation without disrupting others' work. Once developments were validated, personal branches were merged into the `dev` branch, which served as the basis for continuous integration. The `main` branch was reserved for the most stable version of the project, reflecting deliverables tested and ready for use. We also tried to apply a clear naming convention for branches (for example, `feature/function-name`, `fix/bug-name`, etc.) and used Git tagging to mark important versions or development milestones (`v1.0`, `v1.1`, etc.), which facilitates tracking project evolution.

4.7 Gap between Design and Realization

During development, although we respected the broad outlines defined in the design phase, certain important evolutions were introduced to better meet technical constraints and functional needs. We kept obtaining precise results as a priority while relying on accessible technologies. The main gaps concern the introduction of new modules and adjustments in software architecture:

- The addition of an improvisation module, allowing processing of free questions not initially planned in the list of functionalities. This flexibility proved essential to enrich interaction with the user.
- The development of a custom tokenizer, trained on datasets we built, to interpret user voice instructions more accurately.
- The integration of an LLM (Large Language Model) specifically dedicated to processing predefined commands, ensuring more coherent and rapid responses.

These choices improved modularity, response precision, and user experience, at the cost of slight complexification compared to the initial plan.

Planned Element	Final Realization	Justification
Fixed voice commands only	Addition of an improvisation module (LLM)	Allows answering free questions and enriching user interaction
Use of Google Gemini	Tokenizer trained on custom datasets	Better interpretation of complex or ambiguous instructions
Hard-coded responses for functions	Dynamic generation of responses via LLM model	Greater system flexibility and scalability

Table 6: Gaps between initial design and final realization

5 Project Control: Communication, Budget, Planning

5.1 Communication

5.1.1 Summary of All Meetings

During the second phase of the project, the team strengthened coordination by continuing **mandatory weekly meetings**, involving all members. These meetings allowed us to make groups communicate with each other and correct our flaws at the end of Sprints. In parallel, **stand-up meetings by working group** were organized with the project manager, particularly during rush periods, to follow the progress of project subparts (hardware, software, design) more finely and validate technical decisions made by each group. At the end of each main meeting, a **structured report** was written, systematically containing:

- a summary of the **meeting proceedings** and collectively **validated choices**,
- an **updated schematic** of product architecture,
- a reminder of **upcoming GANTT deadlines**,
- an updated **role distribution** by group,
- a **list of tasks** to be performed in the following week or fortnight.

These reports were **shared directly on the project's Facebook group** to ensure visibility by all members. This method proved more effective than Discord (little used for formal reports), or Notion, which required everyone to remember to connect manually. All these documents can be found below.

Réunion 25/10: Définition du projet	Réunion hebdomadaire		October 25, 2024 6:00 AM
Réunion 08/11: Recherche documentaire	Réunion hebdomadaire		November 8, 2024
Réunion 15/11: Mise en place GdP	Réunion hebdomadaire		November 8, 2024
Réunion de gestion 20/11: Définition des deadlines	Réunion hebdomadaire		November 20, 2024
Stand-up meeting 1 06/12: Développement logiciel	Stand-up meeting	Développement logiciel	November 20, 2024
Stand-up meeting 1 06/12: Communication	Stand-up meeting	Communication utilisateur	December 6, 2024
Stand-up meeting 1 06/12: Détection d'objet	Stand-up meeting	Détection d'objets	December 6, 2024
Stand-up meeting 1 06/12: Groupe vision	Stand-up meeting	Détection d'objets	December 6, 2024
Stand-up meeting 2 13/12: Groupe Voix	Stand-up meeting	Communication utilisateur	December 13, 2024
Stand-up meeting 2 13/12: Groupe Vision	Stand-up meeting	Détection d'objets	December 13, 2024
Réunion 08/01: Rappel premier sprint	Réunion hebdomadaire		January 8, 2025
Réunion 13/01: Mise au point et suite du projet	Réunion hebdomadaire		January 13, 2026
Réunion 17/01: Début Sprint 2	Réunion hebdomadaire		January 17, 2026
Réunion 21/01: Avancée sprint 2	Réunion hebdomadaire		
Stand-up meeting 3 23/01: Retour Hussein	Séance de réflexion		
Réunion 24/01: Feedback Sprint 2	Réunion hebdomadaire		January 24, 2025
Réunion 07/02: Fin Sprint 2/Début Sprint 3	Réunion hebdomadaire		February 7, 2025
Réunion 14/02: Benchmark & Phase de tests	Réunion hebdomadaire		February 14, 2025
Réunion 21/02: Retour fin de phase d'intégration et benchmark + LLM Hussein	Réunion hebdomadaire		February 21, 2025
Réunion 07/03: Fin Sprint 3/Départ sprint 4	Réunion hebdomadaire		March 7, 2025
Stand-up meeting 4 12/03: Groupe par groupe	Stand-up meeting		March 13, 2025
Réunion 14/03: Retour travail groupes hardware, database, … & Nouvelles étapes	Réunion hebdomadaire		March 13, 2025
Réunion 21/03: Retour bref visio et explications suite	Réunion hebdomadaire		March 21, 2025
Réunion 28/03: Début sprint 5	Réunion hebdomadaire		March 28, 2025
Réunion 04/04: Démonstration sprint 5	Réunion hebdomadaire		April 4, 2025
Réunion 11/04: Fin du sprint 5 & POC terminé	Réunion hebdomadaire		April 11, 2025
Réunion 01/05: Retour livrables et deadlines finales	Réunion hebdomadaire		May 1, 2025

Figure 17: List of meetings and meeting types

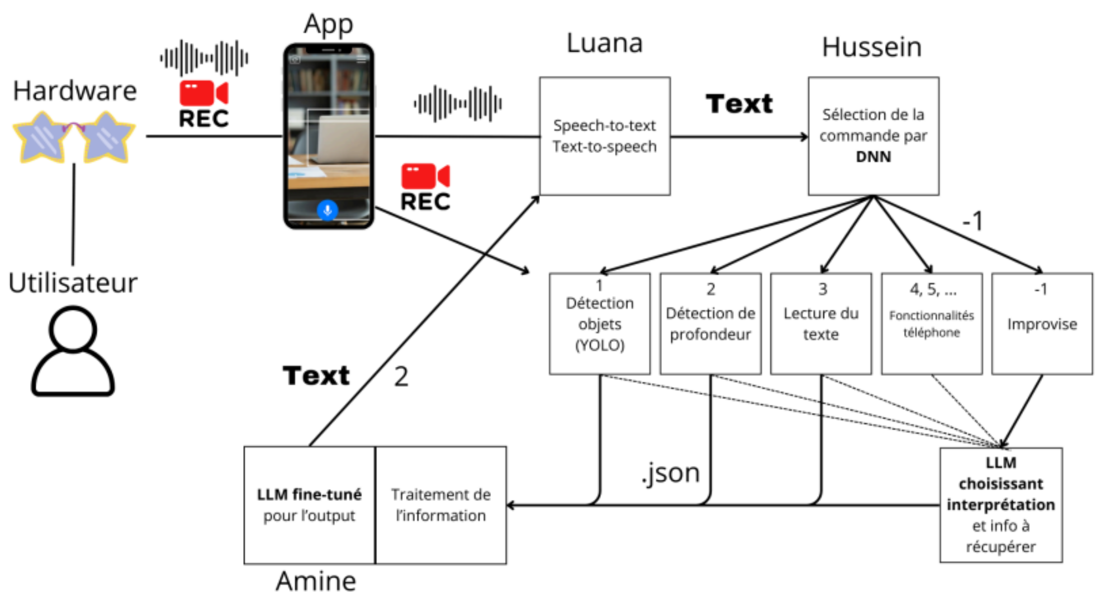


Figure 18: Example of schema associated with a meeting report

Retour de la réunion:

Hardware:

- Caméra -> Choix d'une caméra classique plutôt que RGBD par contrainte de coût et pour itérer sur nos efforts précédents
- Micro-controlleur -> Utilisation du Rasberry Pi 5 du prof pour le moment
- Micro+Haut-Parleurs -> Utilisation d'écouteurs simples et cheap
- Batterie -> Batterie classique Arduino OU Batterie à piles pour permettre une meilleure autonomie
- Design ->
    - Abandon de l'idée de broches pour lunettes, ou de bandeau pour lunettes: problème d'inconfort et de mauvaise répartition du poids très rapide. Problématique pour les lunettes percées.
    - **Choix d'un bandeau imprimé en 3D en plastique avec des joints/coussins qui se clipse sur le front avec la caméra dessus et d'un boîtier sur la ceinture relié à la caméra par un câble.**
    - **Choix de ne pas se concentrer sur le support pour le moment et de déjà utiliser les composants qu'on a déjà pour faire des tests**
    - **Tout le calcul se ferait sur un serveur externe au portable et le casque interagit en réseau avec le portable ou le serveur, pas de bluetooth**
- Ventilateur possible si rasberry chauffe trop

Figure 19: Example of linear meeting feedback

Feedback réunion 07/03

TODO pour la semaine:

Groupe hardware:

1. Trouver des modèles pré-existants de bandeau pour clipser une caméra sur la tête, imprimable en 3D ou pas. Explorer site de créations communautaires pour cela, ne pas tout refaire soi-même
2. Contacter l'U2IS + Taruffi pour récupérer:
   a. Rasberry Pi
   b. Caméra
   c. Batterie
   d. Câbles et connectique nécessaire

Groupe fonctionnalités:

1. Choisir 2 commandes **n'ayant aucun rapport avec la vision** importantes à réaliser
2. Etudier possibilité API Google/Alexa pour réaliser commandes
3. Commencer la réalisation du code des 2 commandes de manières modulaires, intégrable à l'app Kivy et les faire tourner sur portable. (à finir pour le 28 mars)

Groupe dataset:

1. Penser à tous les scénarios et les lister
2. Constituer 2 datasets exhaustifs:
   a. Dataset pour les commandes que on a déjà
   b. Dataset pour les commandes pas encore existante pour anticiper scénarios
3. (Fine-tuning + important possible mais dépendant du dataset, pour plus tard)

Groupe test:

1. Trouver plus de contacts
2. Contacter les associations et fixer rencontres
3. Fixer dépendances et essayer d'importer l'app sur phone (pour le 28 mars)

Objectifs par personne:

Répartition libre par groupe.

Figure 20: TODO from meeting report

5.1.2 Significant Successes

1. Effective pivot from sprint 2 to sprint 3 for hardware realization thanks to the good conduct of the meeting
2. Good general flow of information bilaterally between the team and the project manager through stand-up meetings and meeting reports with graphic explanations
3. Good use of feedback from each end of sprint to allow project evolution and good redistribution of groups

5.1.3 Difficulties Encountered

1. Numerous meetings exceeding the initial time set due to divergence of vision addressed at the wrong time
2. Problem of parallel progress of the hardware team dependent on other groups because tasks distributed over too short a timeframe
3. Temporal distribution of tasks not anticipated enough in advance

5.1.4 Assessment and Lessons

Tools (Discord, Notion, Google Meet) were adapted to our needs. However, the lack of formalization of decisions during the first months and the problem of getting used to Notion led to errors that were resolved by systematic sending of synthetic reports to the whole group on Messenger. Documentation in Notion proved very useful once it was systematized.

5.2 Financial Report

In collaboration with Frank Tarrufi, we borrowed equipment from PIE and many necessary parts were provided to us. This helped reduce PIE costs. However, for technical, financial, and client alignment reasons, between our provisional budget and the final budget, many things evolved. The most notable are:

1. **Change from RGBD camera to classic camera**: modern depth analysis techniques in monocular vision allow very satisfactory results, and to not impact design cost and final product complexity, we chose a classic camera, compatible with Arduino.
2. **Non-use of Bluetooth module for Arduino**: such a module would only be useful for wireless earphones but would add a layer of complexity so it was abandoned.
3. **Addition of a battery**: this will allow the glasses to last more than 8h without being recharged.
4. **Addition of push button**: This will allow manual command passing without using voice.

The material not mentioned remains very close to what we initially had.

Planned Item
RGBD Camera
Bluetooth for Arduino
Arduino Board
Head strap
Cables and resistors
Earphones
3D printer spool
Total

Item	Price
Camera (non-RGBD)	20.58€
Bluetooth for Arduino (abandoned)	–
Arduino Board 5 (Raspberry Pi 5)	138.00€ (provided)
Arduino (RPi) Case and Fan	11.40 € (provided)
Lexar 32GB SD Card	11.90 € (provided)
Head strap (GoPro Mount)	29.99 €
Cables and resistors	(provided)
Earphones	(provided)
3D printer spool	(provided)
Battery 330V	23.99€
Converter 330V - 5V	13.99€
USB C Cable	4.51€
Push button	6.29€
Total not provided	99.35€

5.3 Sprint Backlog

5.3.1 Initial Sprint Backlog

First of all, the sprint backlog allows visualizing each of the sprints determined at the beginning of the project over time. The objectives of each sprint are:

- **Sprint 0**: Define the entire project, get informed on the state of the art and user needs.
- **Sprint 1**: Have functionalities that work individually.

- **Sprint 2**: Refine Sprint 1 solutions and integrate different functionalities into a single Proof of Concept. This POC will be an application on mobile.
- **Sprint 3**: Design the 3D model of glasses and revisit the POC to determine aspects to change and improve. Determine necessary electrical components.
- **Sprint 4**: 3D print a first physical prototype, test its ergonomics and make a first physical model.
- **Sprint 5**: Integrate everything together and revisit the previous sprint to improve features. Define new features to add.
- **Sprint 6**: Finish the product after final test phases, guarantee stability of new features.

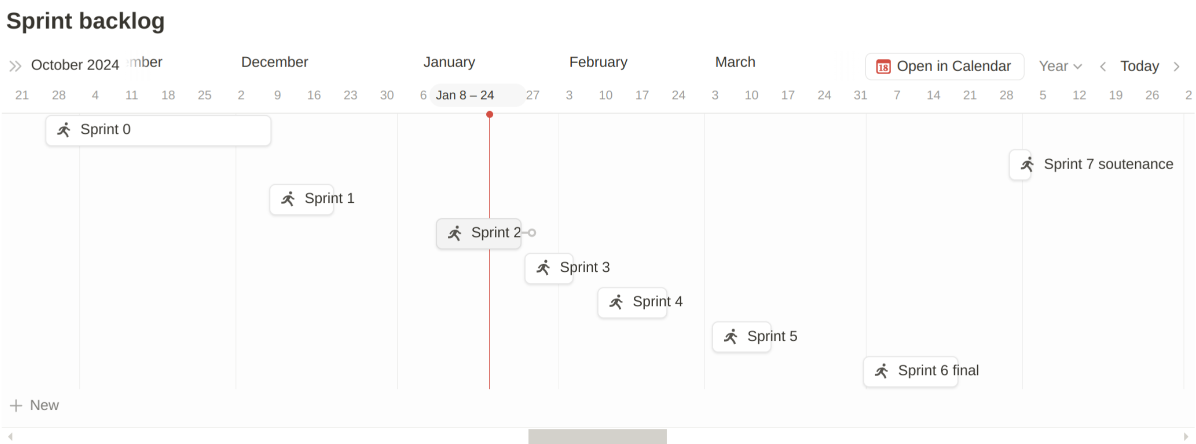


Figure 21: Sprint backlog

5.3.2 Final Sprint Backlog

After continuing the project, the Sprint Backlog evolved significantly following various delays taken during the project and changes in direction called "pivots". Thus, at the end of each Sprint, we evolved the following sprints to adapt according to the Agile method. However, as the sprint breakdown was done based on the school schedule, the dates remain exactly the same. Thus, achievements at the end of each sprint were:

- **Sprint 0**: Define the entire project, get informed on the state of the art and user needs.
- **Sprint 1**: Have functionalities that work individually.
- **Sprint 2**: Refine Sprint 1 solutions and integrate different functionalities into a single Proof of Concept. This POC will be an application on mobile. -> Delay taken on Sprint 2 due to hardware part and too rough anticipation of auxiliary workload.

- **Sprint 3**: Design the 3D model of glasses and revisit the POC to determine aspects to change and improve. Determine necessary electrical components. -> End of Sprint 2 on Sprint 3 and task distribution of Sprint 3 during Sprints.
- **Sprint 4**: 3D print a first physical prototype, test its ergonomics and make a first physical model. -> We were restricted by hardware availability. During this phase, we therefore created a training database for our model, unit tests, and performed Sprint 3 tasks.
- **Sprint 5**: Integrate everything together and revisit the previous sprint to improve features. Define new features to add. -> Integration of new improvisation and call functionalities with first hardware test and realization of a software POC integrating new functionalities 100% functional and a hardware POC 50% functional. Hardware delivery delay prevented us from finishing the physical prototype.
- **Sprint 6**: Finish the product after final test phases, guarantee stability of new features. -> End of hardware prototype and integration of new features into physical prototype, unit tests and integration tests.
- **Sprint 7:** End of auxiliary deliverables, user test and project post-mortem assessment.

You can find the detailed task decomposition in the following section.

5.4 Planning Management and Critical Path

5.4.1 Initial Team Backlog

As can be seen below, each task belongs to a "project" corresponding to a distinct team. Task progress is updated based on whether a task has been marked as finished or not. This visual allows primarily quantifying progress by team rather than time management.

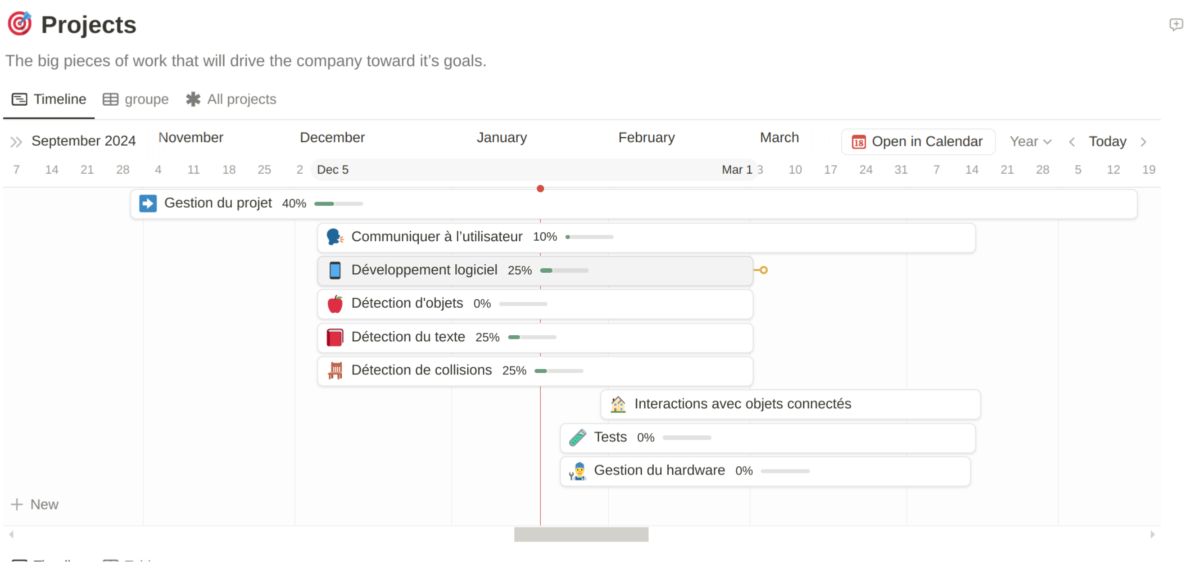


Figure 22: Team Backlog

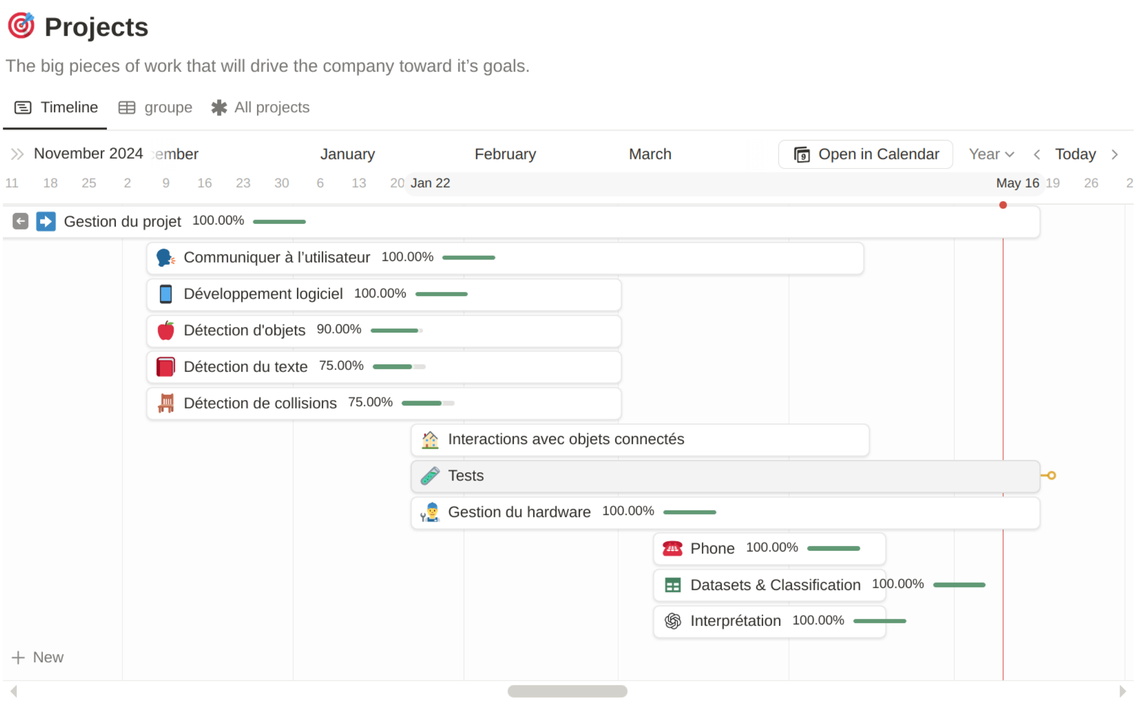


Figure 23: Final Team Backlog

5.4.2 Final Team Backlog

After 5 retrospective Sprint reviews, we adapted the time management of each working group to match our progress and deadlines. We also added new working groups from recent sprints called "pivots". We then obtain the following Team Backlog.

It will be noted that a part is not completed since it is the interaction with connected objects part. We chose to set this part of the project aside in favor of call and SMS functionalities located just below. We also added the dataset and classification group handling the AI that classifies tasks, and the Interpretation group handling an agentic and independent AI able to interact with the internet if a command is not recognized. Finally, it will be noted that the hardware management part was extended due to delivery and design delays we faced.

5.4.3 Initial GANTT

Here we have the initial GANTT chart generated by Notion representing all tasks distributed over time. These are linked to a project but also and especially a Sprint, and also have owners. Everything is linked by a single database in the Notion management tool and thus the Kanban board allows indicating task progress in real-time on the GANTT. Below is the GANTT chart from our first two sprints before adding new tasks and adapting previous tasks. This diagram allowed us to track progress on elementary tasks and paths to follow as we went along. Much information is useful here. At the very top are the **dates**, tasks are distributed according to their start date and deadline, this is the small **blue block**. Their **progress** from Kanban is indicated at the very beginning (Done/Not done/In progress) and the task name is on the left and overflows at the task level. Then, we can see task **assignments** (small letters in a circle). Finally, yellow arrows indicate **dependencies** between tasks. This is only a visual representation of our task database representing all properties and assignments which you can find in the Appendix.

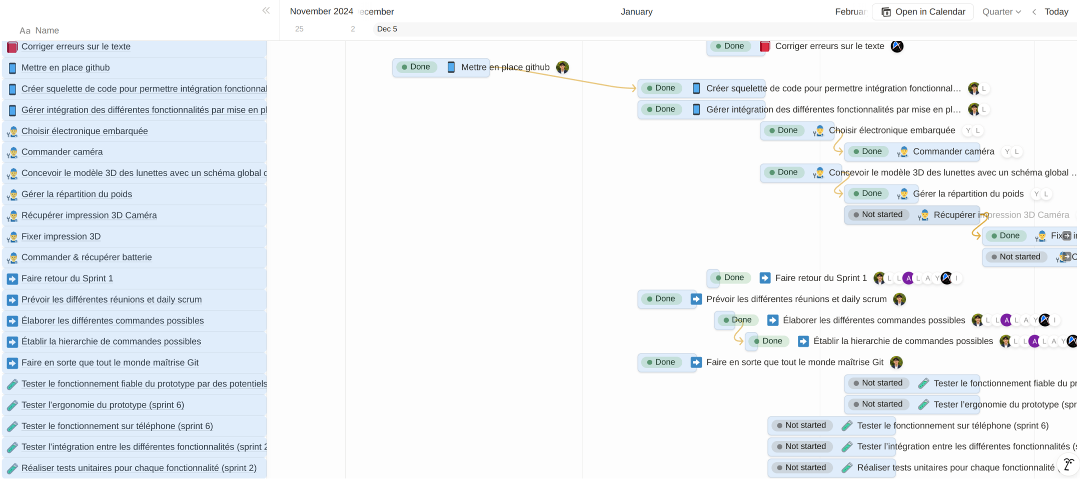


Figure 24: GANTT Chart Sprint 1 & 2

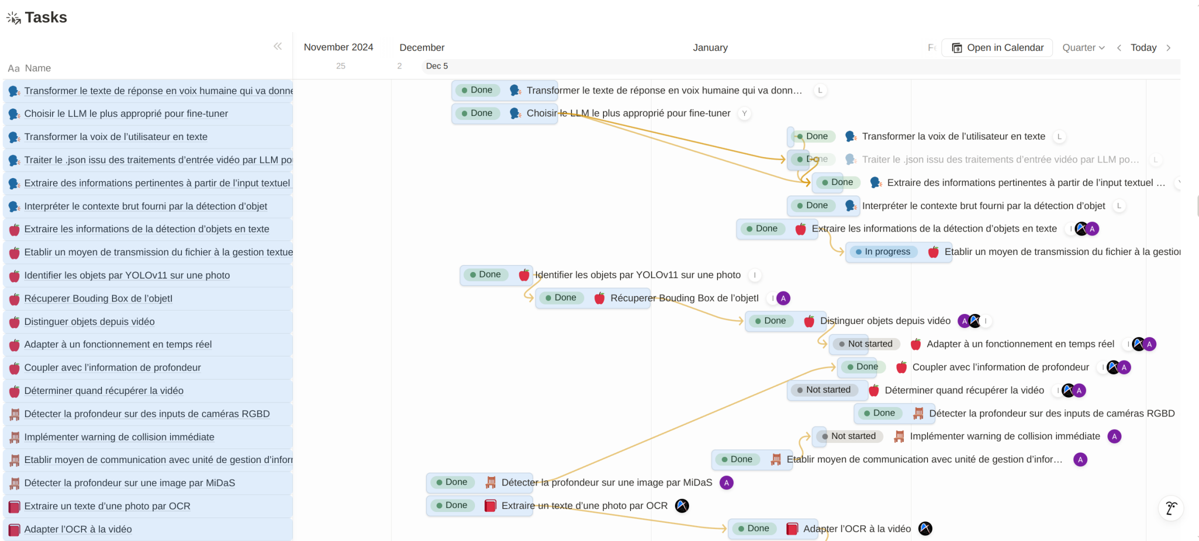


Figure 25: GANTT Chart Sprint 1 & 2

5.4.4 Final GANTT

You will find below the GANTT chart associated with Sprints between March and May. Notable modifications are justified in the following section.

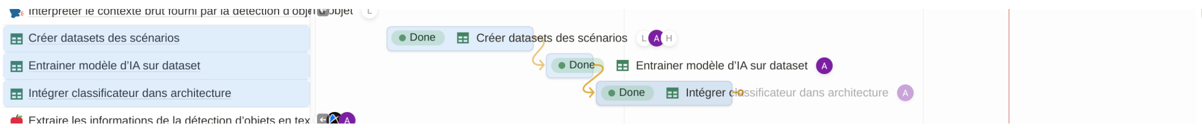


Figure 26: GANTT Chart Sprint 4-7 1

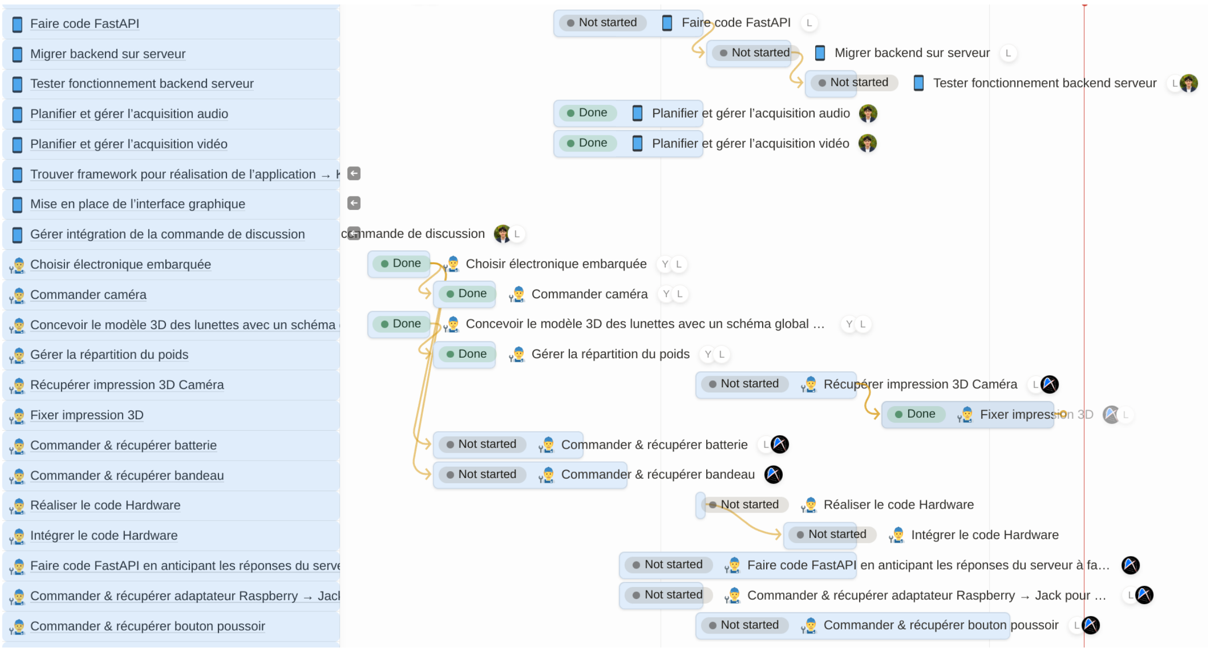


Figure 27: GANTT Chart Sprint 4-7 2

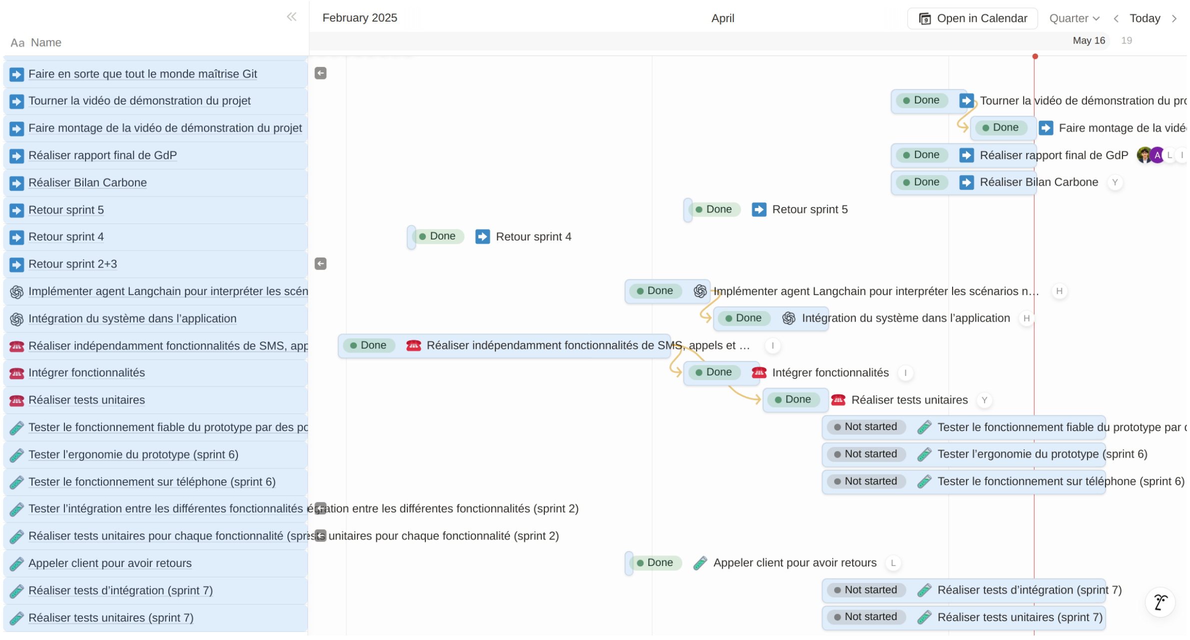


Figure 28: GANTT Chart Sprint 4-7 2

5.4.5 Gap Analysis

Several gaps between the two GANTTs can be mentioned here:

1. First, **hardware management** constitutes the critical path that caused us significant delay, which is why tasks from previous sprints were shifted in time to allow smooth operation.
2. During new sprints, three new working groups were created with their associated tasks after pivoting post-Sprint 3. Applying this Agile method allowed us to be flexible while focusing on long-term objectives by proposing new approaches during the project, like the interpretation/textual processing working group.
3. We did not follow the GANTT Chart to the letter because each week, we reviewed project progress and **weekly tasks were distributed according to weekly progress**. This choice made by the project manager allowed members to have a clear short-term objective to **increase intensity** and **decrease reluctance to tools used** (a Facebook message worked practically better than a link to Notion).

In conclusion, task breakdown by week transmitted via reports and end-of-Sprint meetings aiming to approach the project with a new perspective by redefining roles allowed us to improve the efficiency and relevance of our group work.

5.5 Global Assessment

The systematic implementation of synthetic reports shared on Messenger compensated for a difficult start on Notion. The combination of weekly meetings and targeted stand-up meetings strengthened inter-group coordination, especially during rush periods. However, some meetings suffered from time overruns and poorly framed divergences, reminding us of the importance of keeping a strict agenda. Replacing expensive or useless components (RGBD camera, Bluetooth module) with simpler solutions allowed optimizing cost without harming functional quality. The final budget (99 € for items not provided) remains controlled and consistent with initial ambitions. Continuous adaptation of sprints and working groups after the prototyping phase was decisive in refocusing efforts on achievable goals. The weekly backlog, shared in meetings, facilitated a more dynamic breakdown, more motivating, and better adapted to student schedules. It taught us to structure our exchanges, anticipate technical and human unforeseen events, and leverage an evolving management method. The diversity of deliverables and the ability to document them make this project transferable, durable, and improvable today. The project found a good balance between initial planning and continuous adaptation. Management in sprints, supported by regular meetings and reports shared on Messenger, improved coordination despite difficult beginnings on Notion. The budget was controlled thanks to adjusted technical choices, such as replacing the RGBD camera and abandoning the Bluetooth module, while integrating useful additions (battery, push button). Delays related to hardware were partially compensated by agile task reorganization. This experience allowed us to learn to structure our exchanges, manage uncertainty, and produce clear and reusable documentation for the project continuation.

6 Global Synthesis Feedback

6.1 List of Produced Deliverables

This section presents a summary of deliverables produced within the framework of the *Clairvoyance* project. The table below details each deliverable, its version, owners, location (link to repository or document), and status (delivered, in progress, or abandoned). These deliverables reflect the team's efforts to develop a functional and documented solution, including software, hardware, environmental analysis, project management report, and presentation video.

Table 7: Summary of produced deliverables

Title	**Version**	**Owner**	**Location**	**Status**
Open-source Software	2.0	Entire team	GitHub	In progress
Hardware Prototype	1.0	Ayoub Boufous	Google Drive	In progress
Carbon Footprint	1.0	Yasmin Van Den Broek	Google Drive	Delivered
GDP Report	1.0	Gabriel Dupuis, Amine Maazizi, Ismail El Moufakir, Luana Garcia	Google Drive	Delivered
Presentation Video	1.0	Louis Marchal	YouTube	Delivered

6.2 Qualitative and Quantitative Assessment

The project allowed delivering a functional prototype, well documented and aligned with initial expectations. Of the 7 defined sprints, all were achieved, even if some experienced partial delays, notably on hardware aspects. Nearly **90% of backlog tasks were completed according to Notion tracking**, more than **500** code versions (commits) exist, and **80 %** of initial requirements were totally or partially respected. From a budgetary point of view, expenses not covered by PIE equipment amount to approximately **99€**, a controlled cost thanks to judicious technical choices (RGBD camera replacement, Bluetooth abandonment). The quality of documentation produced (minutes, diagrams, notices, commented code) as well as the flexibility of our organization also make this project easily reusable and scalable.

6.3 Key Learnings

Technical: we learned to design an electronic prototype, integrate embedded software modules, manage software dependencies, and manipulate 3D design tools. We also learned to fine-tune AI models, use an API interface, use vision models, text detection, depth detection, and exploit mobile phone features. **Organizational**: the breakdown into sprints, regular meetings, and task tracking via Notion and Messenger allowed us to progress over time even during exam periods. **Human**: the project confronted us with concrete situations of teamwork, tension management, collective decision-making, and adaptation to technical unforeseen events. Github code organization in particular was a major project issue from which each member learned lessons.

6.4 Recommendations for the Future

For teams who will take over the project:

- Do not underestimate delays related to hardware (orders, 3D printing, testing).
- Centralize communication on a platform consulted regularly.
- Document each sprint and each technical choice in a synthetic format.
- Finalize the "interaction with connected objects" part, left aside due to lack of time.
- Fully integrate the project into a mobile application for communication features.
- Push the improvisation part further.

6.5 Conclusion

This project allowed us to develop key skills in project management, technical design, and team coordination. It strengthened our rigor, adaptability, and autonomy. More so, this project imagined, designed, and carried out this year by our team is a demonstration of the strength of collective work and shows that even the wildest ideas can be realized with a team that fervently believes in its project. We hope that the work accomplished will serve as a solid basis for future teams, and that the tools and documents transmitted will allow them to go further in the development of *Clairvoyance*.

6.6 Testimonials

" This was not my first experience leading a team for a project but it was my first time as project manager of such a large team on such a big project and such a duration. I loved the experience from start to finish, from the conceptualization phase to the realization of the final prototype; it was really satisfying to see my project materialize with such a motivated team. This project stems from frustration I felt after being around my grandfather daily. I perceived that it was impossible for him with his visual and motor handicap to be autonomous in his own home despite the fact that necessary technology already exists. That is why I wanted to set up Clairvoyance, to provide a solution to all people living the same daily life. Other simpler pre-conceived PIEs existed but I knew how to surround myself well. Amine, Ayoub, Ismail, Hussein, Lara, Luana, Louis, and Yasmin believed in the project as much as I did and it was very pleasant to work with them. Obviously, not everything went as planned during the project but that's part of the adventure, and if I were asked to do it again, I would do it with pleasure. "

Gabriel Dupuis

"The Clairvoyance project was initially intimidating because it was a brand new project imagined by Gabriel. We had neither feedback from previous years nor a dedicated tutor to guide us. However, this proved to be a wonderful opportunity to put into practice the concepts covered in project management during this 2nd year at ENSTA. I understood how a balanced distribution of efforts among team members can move a large-scale project forward. Technically, Clairvoyance offered me the opportunity to apply my computer skills acquired at school and via self-training. I was head of the computer vision team in the first semester, where I developed an image depth estimation tool using machine learning. In the second semester, I led the "agentic" team, generating scenarios to train language models and neural networks capable of classifying, decomposing, and reconstructing questions and answers. This team leader role also allowed me to develop my skills in leadership, listening, and collaboration. Overall, this experience was extremely formative at several levels, and I sincerely thank my teammates for their commitment."

Amine Maazizi

"When Gabriel contacted me to set up a team around a new PIE subject, I was super motivated by the idea of discovering the world of AI a bit more. We started almost from scratch, with very little experience, or even none for some. And honestly, I am proud of what we managed to do together. It's not perfect, far from it, but that's not what matters most to me. What I retain above all is everything I learned along the way: text recognition, hardware, project management... And then the team atmosphere, the evolution between the first meetings where we didn't know each other too well, and the moment when we really formed a cohesive group. We built a team spirit that, I think, will remain engraved in my memory for a long time. Obviously, we made quite a few mistakes, in organization, time management, code too. But that's precisely how you learn. It's totally different from what we see in class, where we sometimes have the impression that everything works on the first try if we follow the steps well. Here, we were confronted with reality: it crashes, it blocks, you have to retry, adapt, distribute tasks, bounce back. And in the end, it's hyper formative.

A big thank you to the whole team — Gabriel, Amine, Ismail, Louis, Hussein, Luana, Lara, Yasmin — for this super adventure we lived together. This project really marked me, as much for what I learned there as for the human experience."

Ayoub Boufous

"Working with artificial intelligence was a good experience, both in terms of investigation and knowledge reinforcement. Unfortunately, we don't have much time for the PIE, as we would like, and it seems to me that the school doesn't give it all the importance it deserves. Moreover, we didn't have a professor to ensure project tracking. But it's great to give life to a project like Clairvoyance."

Luana Garcia

"My participation in the Clairvoyance project, which aims to develop smart and accessible glasses for visually impaired people, particularly motivated me. I was able to concretely experiment with project development, its organization as well as challenges encountered. My work mainly focused on language processing and its integration with application features, while also contributing to the elaboration of the project's carbon footprint assessment. This experience allowed me to strengthen my technical skills and better understand the different stages of a technological project."

Yasmin Van Den Broek

"This project allowed me to apply what I had learned in class, both technically and non-technically, like in project management. Realizing such a project in a team, over such a long period, allowed me to confront concrete engineering issues that I will face later."

Louis Marchal

"Participating in the Clairvoyance project was a stimulating and memorable experience. I was in charge of the artificial intelligence part, and more specifically the "improvisation" aspect: ensuring that smart glasses could answer all sorts of user questions naturally and fluidly. I dove into the heart of the most advanced methods in Python, mobilizing cutting-edge tools in the field. Throughout the project, I had the chance to be integrated into several teams, which allowed me to discover varied approaches and enrich my global vision. Weekly meetings organized by our project manager were real spaces for exchange where every idea counted; it was a rare and precious group dynamic. Technical interdependencies sometimes made things complex, but thanks to solid organization and brilliant teammates, challenges became a learning opportunity. But beyond technical aspects, it is especially the human adventure that marked me. By building glasses intended to help others see better, I paradoxically gained clarity myself."

Hussein Rammal

"Within the framework of this project, I had the opportunity to work on the design of smart glasses intended to identify environments to help people with disabilities. During the initial phase of the project, I was responsible for developing code for object classification. This step was particularly enriching for me, as it allowed me to deepen my knowledge on complex code structure, a field in which I did not yet have much experience. I was thus able to progress significantly in programming and computer vision. In the final phase of the project, I took charge of the hardware part. It was a very satisfying experience because I was able to put into practice my skills in electrical engineering, actively participating in component integration and system assembly. This project allowed me to develop varied technical skills while contributing to an initiative with a positive social impact."

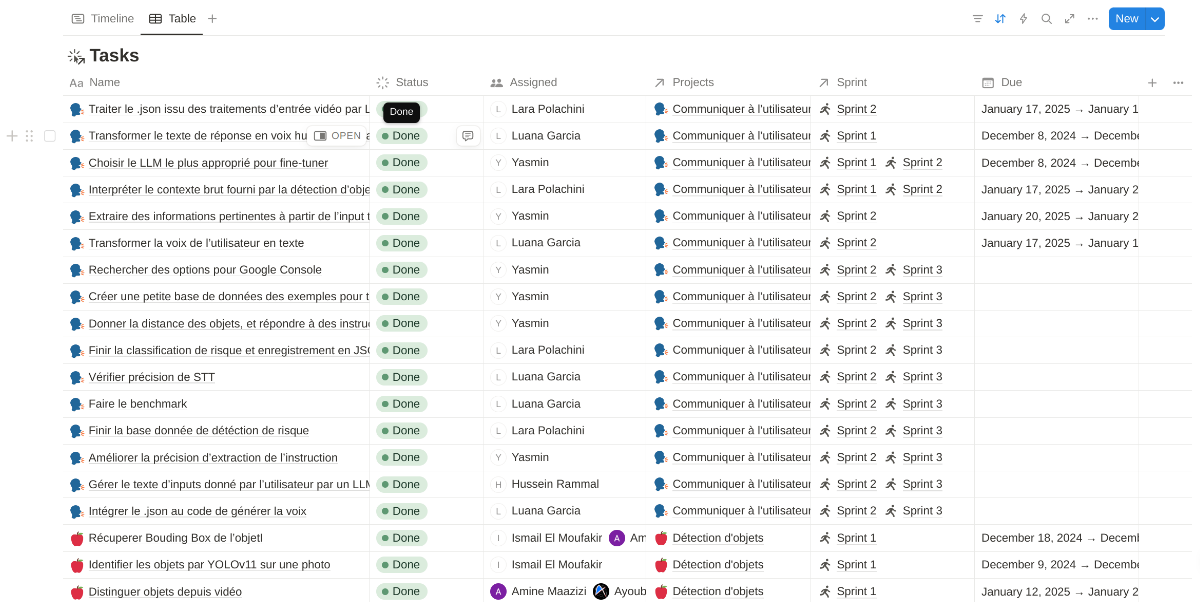
Lara Polachini

Acknowledgements

We wish to express our deep gratitude to all the people who contributed to the realization of our PIE Clairvoyance project aiming to assist visually impaired people in their daily lives. We particularly thank Mr. **Franck Taruffi** for his attentive guidance and constant support throughout the year. His supervision was an essential pillar of the project's progression. We also address our warm thanks to **Damien Spudic** and **Alexis Scotto d'Apollonia**, for their valuable contributions to the "carbon footprint" part of the project, which allowed us to integrate a rigorous environmental dimension. Finally, we thank **Laure Letellier** for her enlightened advice on project management, which allowed us to effectively structure our work and respect key milestones throughout the year.

Appendix

Task List

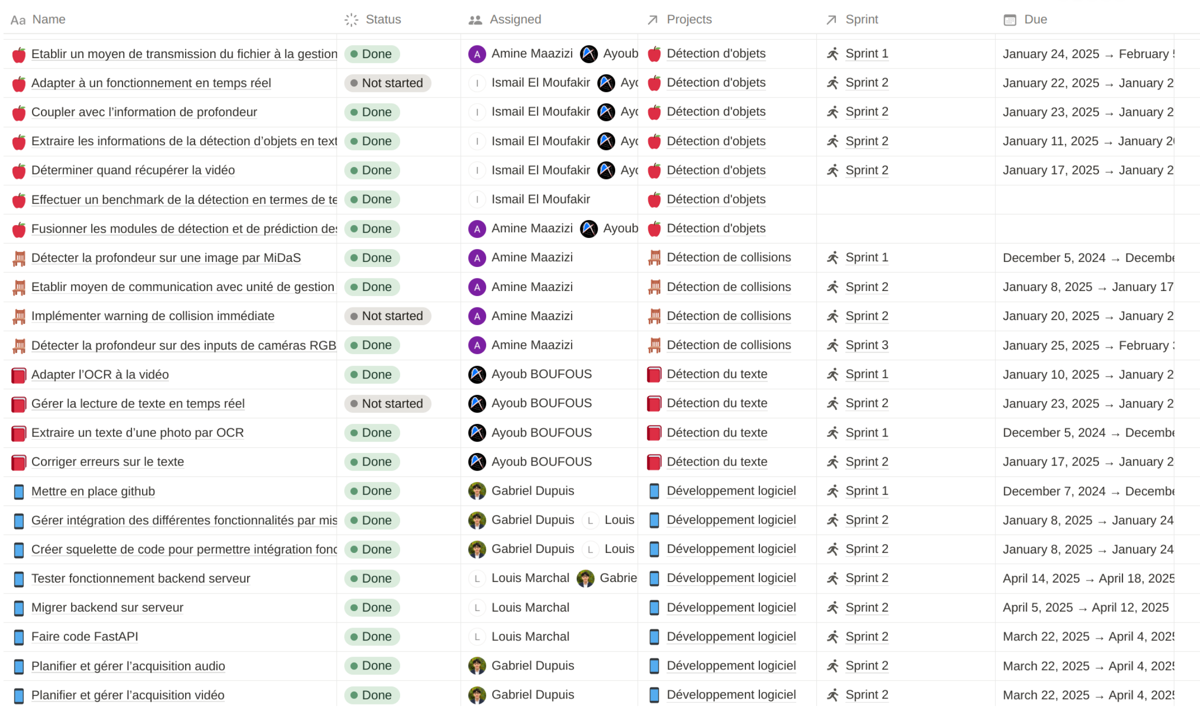


Timeline | Table | New

Tasks

Name	Status	Assigned	Projects	Sprint	Due
Traiter le .json issu des traitements d'entrée vidéo par L	Done	Lara Polachini	Communiquer à l'utilisateur	Sprint 2	January 17, 2025 → January 1
Transformer le texte de réponse en voix hu	Done	Luana Garcia	Communiquer à l'utilisateur	Sprint 1	December 8, 2024 → Decemb
Choisir le LLM le plus approprié pour fine-tuner	Done	Yasmin	Communiquer à l'utilisateur	Sprint 1, Sprint 2	December 8, 2024 → Decemb
Interpréter le contexte brut fourni par la détection d'obje	Done	Lara Polachini	Communiquer à l'utilisateur	Sprint 1, Sprint 2	January 17, 2025 → January 2
Extraire des informations pertinentes à partir de l'input t	Done	Yasmin	Communiquer à l'utilisateur	Sprint 2	January 20, 2025 → January 2
Transformer la voix de l'utilisateur en texte	Done	Luana Garcia	Communiquer à l'utilisateur	Sprint 2	January 17, 2025 → January 1
Rechercher des options pour Google Console	Done	Yasmin	Communiquer à l'utilisateur	Sprint 2, Sprint 3	
Créer une petite base de données des exemples pour t	Done	Yasmin	Communiquer à l'utilisateur	Sprint 2, Sprint 3	
Donner la distance des objets, et répondre à des instru	Done	Yasmin	Communiquer à l'utilisateur	Sprint 2, Sprint 3	
Finir la classification de risque et enregistrement en JS	Done	Lara Polachini	Communiquer à l'utilisateur	Sprint 2, Sprint 3	
Vérifier précision de STT	Done	Luana Garcia	Communiquer à l'utilisateur	Sprint 2, Sprint 3	
Faire le benchmark	Done	Luana Garcia	Communiquer à l'utilisateur	Sprint 2, Sprint 3	
Finir la base donnée de détéction de risque	Done	Lara Polachini	Communiquer à l'utilisateur	Sprint 2, Sprint 3	
Améliorer la précision d'extraction de l'instruction	Done	Yasmin	Communiquer à l'utilisateur	Sprint 2, Sprint 3	
Gérer le texte d'inputs donné par l'utilisateur par un LLM	Done	Hussein Rammal	Communiquer à l'utilisateur	Sprint 2, Sprint 3	
Intégrer le .json au code de générer la voix	Done	Luana Garcia	Communiquer à l'utilisateur	Sprint 2, Sprint 3	
Récuperer Bouding Box de l'objetl	Done	Ismail El Moufakir, Am	Détection d'objets	Sprint 1	December 18, 2024 → Decemb
Identifier les objets par YOLOv11 sur une photo	Done	Ismail El Moufakir	Détection d'objets	Sprint 1	December 9, 2024 → Decemb
Distinguer objets depuis vidéo	Done	Amine Maazizi, Ayoub	Détection d'objets	Sprint 1	January 12, 2025 → January 2

Figure 29: Tasks 1



Name	Status	Assigned	Projects	Sprint	Due
Etablir un moyen de transmission du fichier à la gestion	Done	Amine Maazizi, Ayoub	Détection d'objets	Sprint 1	January 24, 2025 → February
Adapter à un fonctionnement en temps réel	Not started	Ismail El Moufakir, Ay	Détection d'objets	Sprint 2	January 22, 2025 → January 2
Coupler avec l'information de profondeur	Done	Ismail El Moufakir, Ay	Détection d'objets	Sprint 2	January 23, 2025 → January 2
Extraire les informations de la détection d'objets en text	Done	Ismail El Moufakir, Ay	Détection d'objets	Sprint 2	January 11, 2025 → January 2
Déterminer quand récupérer la vidéo	Done	Ismail El Moufakir, Ay	Détection d'objets	Sprint 2	January 17, 2025 → January 2
Effectuer un benchmark de la détection en termes de te	Done	Ismail El Moufakir	Détection d'objets		
Fusionner les modules de détection et de prédiction de	Done	Amine Maazizi, Ayoub	Détection d'objets		
Détecter la profondeur sur une image par MiDaS	Done	Amine Maazizi	Détection de collisions	Sprint 1	December 5, 2024 → Decemb
Etablir moyen de communication avec unité de gestion	Done	Amine Maazizi	Détection de collisions	Sprint 2	January 8, 2025 → January 17
Implémenter warning de collision immédiate	Not started	Amine Maazizi	Détection de collisions	Sprint 2	January 20, 2025 → January 2
Détecter la profondeur sur des inputs de caméras RGB	Done	Amine Maazizi	Détection de collisions	Sprint 3	January 25, 2025 → February
Adapter l'OCR à la vidéo	Done	Ayoub BOUFOUS	Détection du texte	Sprint 1	January 10, 2025 → January 2
Gérer la lecture de texte en temps réel	Not started	Ayoub BOUFOUS	Détection du texte	Sprint 2	January 23, 2025 → January 2
Extraire un texte d'une photo par OCR	Done	Ayoub BOUFOUS	Détection du texte	Sprint 1	December 5, 2024 → Decemb
Corriger erreurs sur le texte	Done	Ayoub BOUFOUS	Détection du texte	Sprint 2	January 17, 2025 → January 2
Mettre en place github	Done	Gabriel Dupuis	Développement logiciel	Sprint 1	December 7, 2024 → Decemb
Gérer intégration des différentes fonctionnalités par mis	Done	Gabriel Dupuis, Louis	Développement logiciel	Sprint 2	January 8, 2025 → January 24
Créer squelette de code pour permettre intégration fonc	Done	Gabriel Dupuis, Louis	Développement logiciel	Sprint 2	January 8, 2025 → January 24
Tester fonctionnement backend serveur	Done	Louis Marchal, Gabrie	Développement logiciel	Sprint 2	April 14, 2025 → April 18, 2025
Migrer backend sur serveur	Done	Louis Marchal	Développement logiciel	Sprint 2	April 5, 2025 → April 12, 2025
Faire code FastAPI	Done	Louis Marchal	Développement logiciel	Sprint 2	March 22, 2025 → April 4, 202
Planifier et gérer l'acquisition audio	Done	Gabriel Dupuis	Développement logiciel	Sprint 2	March 22, 2025 → April 4, 202
Planifier et gérer l'acquisition vidéo	Done	Gabriel Dupuis	Développement logiciel	Sprint 2	March 22, 2025 → April 4, 202

Figure 30: Tasks 2

Name	Status	Assigned	Projects	Sprint	Due
Mise en place de l'interface graphique	Done	Louis Marchal, Gabrie	Développement logiciel	Sprint 2	January 21, 2025 → January 2
Gérer intégration de la commande de discussion	Done	Gabriel Dupuis, Louis	Développement logiciel	Sprint 2	January 27, 2025 → February !
Gérer intégration commande de distance des objets	Done	Gabriel Dupuis, Louis	Développement logiciel	Sprint 2	
Gérer intégration commande de description d'environne	Done	Louis Marchal, Gabrie	Développement logiciel	Sprint 2	
Trouver framework pour réalisation de l'application → K	Done	Gabriel Dupuis	Développement logiciel	Sprint 2	January 18, 2025 → January 2
Choisir électronique embarquée	Done	Yasmin, Lara Polachin	Gestion du hardware	Sprint 3	March 5, 2025 → March 10, 20
Gérer la répartition du poids	Done	Yasmin, Lara Polachin	Gestion du hardware	Sprint 3	March 11, 2025 → March 16, 2
Concevoir le modèle 3D des lunettes avec un schéma (	Done	Yasmin, Lara Polachin	Gestion du hardware	Sprint 3	March 5, 2025 → March 10, 20
Commander & récupérer batterie	Done	Lara Polachini, Ayoub	Gestion du hardware	Sprint 4	March 11, 2025 → March 24, 2
Fixer impression 3D	Done	Ayoub BOUFOUS, La	Gestion du hardware	Sprint 4	April 21, 2025 → May 6, 2025
Récupérer impression 3D Caméra OPEN	Done	Lara Polachini, Ayoub	Gestion du hardware	Sprint 4	April 4, 2025 → April 18, 2025
Commander caméra	Done	Yasmin, Lara Polachin	Gestion du hardware	Sprint 4, Sprint 5, S	March 11, 2025 → March 16, 2
Commander & récupérer bandeau	Done	Ayoub BOUFOUS	Gestion du hardware	Sprint 5, Sprint 6 final	March 11, 2025 → March 28, 2
Intégrer le code Hardware	Done	Ayoub BOUFOUS	Gestion du hardware	Sprint 5, Sprint 6 final	April 12, 2025 → April 18, 2025
Réaliser le code Hardware	Done	Ayoub BOUFOUS	Gestion du hardware	Sprint 5, Sprint 6 final	April 4, 2025
Faire code FastAPI en anticipant les réponses du serve	Done	Ayoub BOUFOUS	Gestion du hardware	Sprint 5, Sprint 6 final	March 28, 2025 → April 18, 20:
Commander & récupérer adaptateur Raspberry → Jack	Done	Lara Polachini, Ayoub	Gestion du hardware	Sprint 5, Sprint 6 final	March 28, 2025 → April 4, 202!
Commander & récupérer bouton poussoir	Done	Lara Polachini, Ayoub	Gestion du hardware	Sprint 5, Sprint 6 final	April 4, 2025 → May 2, 2025
Faire retour du Sprint 1	Done	Gabriel Dupuis, Lara F	Gestion du projet	Sprint 2, Sprint 5, S	January 17, 2025 → January 1
Prévoir les différentes réunions et daily scrum	Done	Gabriel Dupuis	Gestion du projet	Sprint 2, Sprint 5, S	January 8, 2025 → January 15
Établir la hierarchie de commandes possibles	Done	Gabriel Dupuis, Lara F	Gestion du projet	Sprint 2, Sprint 5, S	January 22, 2025 → January 2
Élaborer les différentes commandes possibles	Done	Gabriel Dupuis, Lara F	Gestion du projet	Sprint 2, Sprint 5, S	January 18, 2025 → January 2
Faire en sorte que tout le monde maîtrise Git	Done	Gabriel Dupuis	Gestion du projet	Sprint 2, Sprint 5, S	January 8, 2025 → January 15

Figure 31: Tasks 3

Faire montage de la vidéo de démonstration du projet	Done	Louis Marchal	Gestion du projet	Sprint 5, Sprint 6 final	May 3, 2025 → May 9, 2025
Tourner la vidéo de démonstration du projet	Done	Gabriel Dupuis, Amine	Gestion du projet	Sprint 5, Sprint 6 final	April 25, 2025 → May 2, 2025
Réaliser rapport final de GdP	Done	Gabriel Dupuis, Amine	Gestion du projet	Sprint 5, Sprint 6 final	April 25, 2025 → May 9, 2025
Réaliser Bilan Carbone	Done	Yasmin	Gestion du projet	Sprint 5, Sprint 6 final	April 25, 2025 → May 9, 2025
Retour sprint 5	Done	Gabriel Dupuis	Gestion du projet	Sprint 5, Sprint 6 final	April 4, 2025
Retour sprint 4	Done	Gabriel Dupuis	Gestion du projet	Sprint 4, Sprint 5, S	March 7, 2025
Retour sprint 2+3	Done	Gabriel Dupuis	Gestion du projet	Sprint 2, Sprint 5, S	February 7, 2025
Intégration du système dans l'application	Done	Hussein Rammal	Interprétation	Sprint 5, Sprint 6 final	April 7, 2025 → April 18, 2025
Implémenter agent Langchain pour interpréter les scéna	Done	Hussein Rammal	Interprétation	Sprint 5, Sprint 6 final	March 29, 2025 → April 6, 202!
Intégrer fonctionnalités	Done	Ismail El Moufakir	Phone	Sprint 5, Sprint 6 final	April 4, 2025 → April 11, 2025
Réaliser tests unitaires	Done	Yasmin	Phone	Sprint 5, Sprint 6 final	April 12, 2025 → April 18, 2025
Réaliser indépendamment fonctionnalités de SMS, app	Done	Ismail El Moufakir	Phone	Sprint 5, Sprint 6 final	February 28, 2025 → April 2, 2(

Figure 32: Tasks 4

Table 1 – Risk Identification and Treatment Plan

Table 8: Risk Identification and Treatment Plan

Activity	Code	Cause	Action
Technical Risks			
AI System Calibration	Risk-01-Tech	Poor calibration of AI algorithms for audio description	• Develop a rigorous calibration protocol • Perform iterative tests with end users
Power Management	Risk-02-Tech	Battery overheating or insufficient autonomy	• Selection of high-quality batteries adapted to needs • Integration of temperature sensors and intelligent energy management
Glasses Durability	Risk-03-Tech	Fragility of the frame or integrated components	• Use robust and lightweight materials • Perform resistance tests in real conditions
Electronic Component Failure	Risk-04-Tech	Sensor or processing system failures	• Choose proven and reliable components • Eventually implement a redundancy system for critical components
Software Compatibility	Risk-05-Tech	Incompatibility with different operating systems or connected devices	• Develop modular and standardized interfaces • Perform multi-platform compatibility tests
3D Printing of Camera Support	Risk-22-Tech	Production delays, support quality, compatibility with camera	• Plan 3D printing in advance • Test support quality and check compatibility with camera

Continued on next page

Table 8 – continued

Activity	Code	Cause	Action
Hardware Acquisition and Integration	Risk-23-Tech	Delivery delays, component quality, compatibility	• Select reliable suppliers • Test components and check compatibility upon receipt
FastAI Code Development	Risk-24-Tech	Code complexity, performance, security	• Develop code in modules to facilitate tests • Perform regular performance and security tests
Integration of New Features (calls, emails, SMS)	Risk-25-Tech	Integration complexity, compatibility with existing app, security	• Develop features in modules • Test integration with existing application • Implement security measures for communications
Migration to a Server	Risk-26-Tech	Server availability, scalability, data security	• Choose a reliable and scalable server • Test availability and scalability • Implement data security measures (encryption, authentication)
Creation of a Scenario Database	Risk-27-Tech	Data quality, scenario relevance, data security	• Collect quality data and validate scenarios with users • Implement security measures for the database
Neural Network Model Training	Risk-28-Tech	Data quality, model performance, overfitting	• Use cross-validation techniques • Regularize models to avoid overfitting • Test models on real data

Continued on next page

Table 8 – continued

Activity	Code	Cause	Action
Integration of an Agent to Interpret Unforeseen Scenarios	Risk-29-Tech	Agent complexity, performance, reliability	• Develop agent in modules to facilitate tests • Perform regular tests and validate with users
Unit and Integration Tests	Risk-30-Tech	Discovery of bugs or incompatibilities late in the project	• Plan regular tests throughout development • Implement automated tests • Quickly correct identified problems
		User Risks	
User Interface	Risk-06-Client	Complexity of voice interface for non-tech-savvy users	• Simplify voice commands • Provide tutorials and effective user support
User Acceptance	Risk-07-Client	User resistance to adopting new technology	• Involve end users from the beginning of development • Collect and integrate user feedback in product iterations
		Schedule Risks	
Scope Creep	Risk-08-Delay	Unplanned addition of features beyond MVP	• Clearly define MVP scope • Prioritize essential features
Supplier Delays	Risk-09-Delay	Delivery delays of electronic components or hardware	• Select reliable suppliers with good references • Implement contractual clauses regarding delays

Continued on next page

Table 8 – continued

Activity	Code	Cause	Action
Missed Deadlines	Risk-31-Delay	Increased time constraints due to personal deadlines (exams, internships)	• Prioritize critical tasks • Adjust schedule based on availability • Communicate regularly on progress
		Team Management Risks	
Internal Coordination	Risk-10-Team	Lack of communication and coordination between team members	• Implement effective project management tools (e.g., Trello, Slack) • Organize regular follow-up meetings
High Turnover	Risk-11-Team	Loss of key team members during project	• Offer incentives and a good work environment • Organize everyone's workload well
Work Overload	Risk-32-Team	Task accumulation due to time constraints	• Distribute workload fairly • Identify tasks that can be delegated or postponed
Team Demotivation	Risk-33-Team	Fatigue and constraints affecting motivation	• Organize team-building sessions • Recognize and value team efforts • Ensure good work-life balance
		Financial Risks	
Budget Overrun	Risk-12-Budget	Underestimation of development and production costs	• Establish a detailed and realistic budget • Plan a budget margin for unforeseen events

Continued on next page

Table 8 – continued

Activity	Code	Cause	Action
Component Price Fluctuation	Risk-13-Budget	Unforeseen increase in electronic component costs	• Negotiate fixed-price contracts with suppliers • Search for less expensive component alternatives
		Environmental Risks	
Use of Non-Eco-Friendly Materials	Risk-15-Env	High environmental impact of materials used	• Select recyclable and sustainable materials • Minimize production waste
Non-Compliance with Environmental Regulations	Risk-16-Env	Non-respect of ecological norms in force	• Perform continuous regulatory monitoring • Collaborate with experts in environmental compliance
		Security Risks	
Software Vulnerabilities	Risk-17-Security	Software flaws compromising user data	• Implement robust security protocols • Perform regular security audits
User Physical Safety	Risk-18-Security	Accidents or injuries caused by glasses usage	• Design ergonomic and safe glasses • Perform safety tests in various usage scenarios
		Quality Risks	
Non-Compliance with Quality Standards	Risk-19-Quality	Final product not meeting established quality standards	• Implement a quality management system • Perform quality controls at each development stage
		Communication Risks	

Continued on next page

Table 8 – continued

Activity	Code	Cause	Action
Poor External Communication	Risk-20-Comm	Ineffective communication strategy impacting public perception	• Develop a clear and targeted communication strategy • Use diverse communication channels adapted to end users
		Legal Risks	
Non-Compliance with Data Regulations	Risk-21-Legal	Non-respect of GDPR and other data protection laws	• Integrate data protection by design (Privacy by Design) • Ensure compliance with GDPR and other relevant regulations

Table 2 – Residual Risk Analysis

Table 9: Residual Risk Analysis

Activity	Code	Residual Risk	Probability (1–5)	Impact (1–5)	Criticality (P×I)
		Technical Residual Risks			
AI System Calibration	Risk-01-Tech	Reduced precision of audio descriptions	3	4	12
Power Management	Risk-02-Tech	Slightly decreased battery autonomy	2	4	8
Glasses Durability	Risk-03-Tech	Premature wear of components	2	3	6
Electronic Component Failure	Risk-04-Tech	Intermittent sensor failures	2	4	8
Software Compatibility	Risk-05-Tech	Incompatibility with certain domestic devices	3	3	9
3D Printing of Camera Support	Risk-22-Tech	Inferior quality or incompatible camera support	3	3	9

Continued on next page

Table 9 – continued

Activity	Code	Residual Risk	Probability (1–5)	Impact (1–5)	Criticality (P×I)
Hardware Acquisition and Integration	Risk-23-Tech	Defective or incompatible components	3	4	12
FastAI Code Development	Risk-24-Tech	Complex code leading to bugs or mediocre performance	4	4	16
Integration of New Features	Risk-25-Tech	Malfunctions of new features or incompatibilities	3	4	12
Migration to a Server	Risk-26-Tech	Server availability or security problems	3	5	15
Creation of a Scenario Database	Risk-27-Tech	Poor quality data or irrelevant scenarios	2	3	6
Neural Network Model Training	Risk-28-Tech	Underperforming or overfitted models	3	4	12
Integration of an Agent	Risk-29-Tech	Agent unable to correctly manage unforeseen scenarios	3	4	12
Unit and Integration Tests	Risk-30-Tech	Undetected bugs or late corrections	4	4	16
		User Residual Risks			
User Interface	Risk-06-Client	Usage difficulties despite simplification	3	3	9
User Acceptance	Risk-07-Client	Persistent resistance to adoption	2	4	8
		Schedule Residual Risks			
Scope Creep	Risk-08-Delay	Additional delays to integrate secondary features	4	4	16
Supplier Delays	Risk-09-Delay	Minor delays in component delivery	3	3	9
Missed Deadlines	Risk-31-Delay	Delays due to time constraints	4	5	20

Continued on next page

Table 9 – continued

Activity	Code	Residual Risk	Probability (1–5)	Impact (1–5)	Criticality (P×I)
Team Management Residual Risks					
Internal Coordination	Risk-10-Team	Persistent communication problems	3	3	9
High Turnover	Risk-11-Team	Occasional loss of key members	2	4	8
Work Overload	Risk-32-Team	Team burnout	3	4	12
Team Demotivation	Risk-33-Team	Productivity drop	3	3	9
Financial Residual Risks					
Budget Overrun	Risk-12-Budget	Moderate overrun of initial budget	3	4	12
Component Price Fluctuation	Risk-13-Budget	Slight increase in component costs	3	3	9
Environmental Residual Risks					
Use of Non-Eco-Friendly Materials	Risk-15-Env	Carbon footprint slightly higher than expected	2	3	6
Non-Compliance with Environmental Regulations	Risk-16-Env	Minor non-compliance with ecological norms	1	4	4
Security Residual Risks					
Software Vulnerabilities	Risk-17-Security	Minor software flaws	2	4	8
User Physical Safety	Risk-18-Security	Minor accidents related to glasses usage	2	5	10
Quality Residual Risks					
Non-Compliance with Quality Standards	Risk-19-Quality	Sporadic failures in product quality	2	4	8

Continued on next page

Table 9 – continued

Activity	**Code**	**Residual Risk**	**Probability (1–5)**	**Impact (1–5)**	**Criticality (P×I)**
		Communication Residual Risks			
Poor External Communication	Risk-20-Comm	Negative public perception despite efforts	1	3	3
		Legal Residual Risks			
Non-Compliance with Data Regulations	Risk-21-Legal	Minor legal infractions related to data	1	2	2