Federal University of Technology of Paraná – UTFPR
Academic Department of Electronics – DAELN
Academic Department of Informatics – DAINF
Computer Engineering
Integration Workshop 3 (EEX23) – S71 – 2024/2

Technical Report
ANAmnesis

Flávia Monique C. Rodrigues – flaviamonique@alunos.utfpr.edu.br
Gabriel Leite R. B. Mororó – gabrielmororo@alunos.utfpr.edu.br
Murilo Kenji Unten – murilokenjiunten at alunos.utfpr.edu.br
Robson de Almeida Santana Junior – robsonjunior.2022@alunos.utfpr.edu.br
Victor Damasceno Oliveira – victordamasceno@alunos.utfpr.edu.br

November 2025

Abstract

Medical triage is a critical component of medical care, serving to prioritize patients based on clinical severity. However, in scenarios of high patient throughput, manual screening processes often become a bottleneck, potentially compromising the quality of anamnesis due to time constraints and human fatigue. To address these challenges, this project proposes the development of an automated anamnesis totem. This system is designed to autonomously conduct the complete initial screening routine, integrating the measurement of vital signs with a semantic dialogue interface for patient history collection. All acquired telemetry and textual data are synchronized in real-time with a centralized web dashboard, granting authorized healthcare professionals immediate access to patient profiles. The proposed solution aims to standardize the screening process, reduce waiting times, and optimize resource allocation in hospital environments.

1 Introduction

Emergency departments (EDs) around the world increasingly face operational overload driven by rising patient demand, limited staffing, and the growing complexity of clinical cases. This persistent congestion places significant pressure on healthcare professionals, who must balance rapid patient turnover with the need for accurate and comprehensive anamnesis. Under such conditions, the initial screening process—responsible for collecting vital signs and patient history—often becomes a bottleneck. Time constraints, human fatigue, and variability in clinical practice can compromise the consistency and quality of triage, ultimately affecting patient outcomes and resource allocation within the hospital.

Recent technological innovations have demonstrated the potential to automate parts of the triage workflow, reducing the burden on medical staff while maintaining or even improving screening quality. For example, **Health Pods**, which autonomously measure and record patients' vital signs, have been shown to reduce triage time and increase both patient and nurse satisfaction, proving that self-service physiological data collection is both feasible and effective in real-world clinical settings [1]. These results indicate that automation can play a meaningful role in standardizing the first stage of emergency department flow.

Beyond vital signs, the collection of patient history and presenting complaints—traditionally gathered through verbal dialogue—can also be supported by intelligent systems. The **TriagE-NLU** model provides a robust multilingual natural language understanding framework for interpreting patient-described symptoms in emergency contexts [2]. Similarly, deep learning approaches such as the Deep Attention Model demonstrate that machine learning techniques can reliably process unstructured triage notes to support clinical decision-making [3]. Together, these works showcase the viability of automated or semi-automated systems for capturing the semantic content of patient complaints with high fidelity.

Furthermore, comparative studies between artificial intelligence systems and human clinicians suggest that algorithmic approaches can perform triage and diagnostic tasks with meaningful levels of accuracy. Notably, Razzaki et al. (2018) found that AI-based diagnostic and triage methods can approximate human performance, reinforcing the notion that automated screening tools have legitimate potential in clinical environments [4].

This project presents the development of an automated medical screening system designed to support healthcare professionals in the initial assessment of patients. The system performs a guided anamnesis combined with real time vital sign measurements and a structured voice interaction that helps collect relevant clinical information with consistency and clarity.

The solution integrates hardware, mechanical components, and software to create a fully functional screening station. It measures body temperature, heart rate, blood oxygen saturation, and blood pressure, as well as communicate with the patient through audio and a display with visual cues. The interaction is voice controlled, allowing the patient to answer predefined and adaptive questions that help identify symptoms and complaints. All collected data is processed, stored, and made available through a web platform where authorized professionals can review medical screenings, access patient histories, and manage user accounts.

The specifications guiding the project were defined considering both the system's intended functionalities and the resources available to the team. It also considers the health agents opinions collected in a form [1]. This resulted in a complete set of functional, non functional, and anti requirements that shaped the design and implementation of the solution. The goal is to deliver a reliable

[1] https://docs.google.com/spreadsheets/d/1Ydnd3c1ZTSFZFic_yqLQcPWO6IIGeqcEpSDvRKDTnpc/edit?usp=sharing

and accessible screening environment that improves workflow, reduces manual workload, and standardizes the initial stages of patient evaluation.

2 Overview and Project Specification

2.1 Overview

The *ANAmnesis* has a totem with all the equipment to measure vital signs and communicate clearly with the patients. Besides that, the system contains a website where all the collected information is stored and made available to the previously registered doctors and nurses. Figure 1 indicates a simplified view of the project.

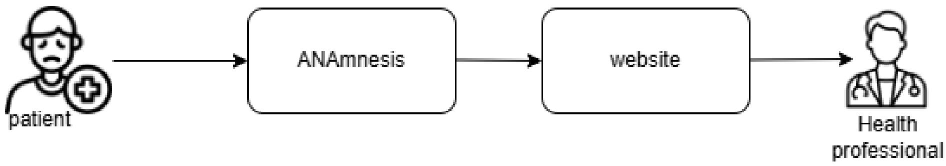

Figure 1: Initial view of the project.

To achieve its objectives, ANAmnesis was developed using the modules displayed in the block diagram in Figure 2.

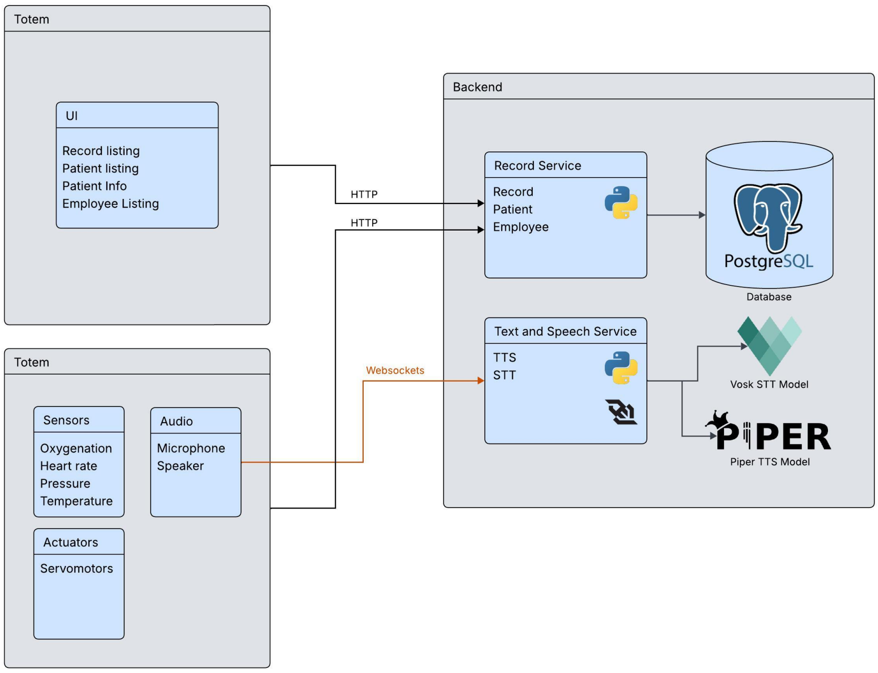

Figure 2: ANAmnesis block diagram.

2.2 Project Specification

The project specifications were defined according to the functionalities that the system must deliver and the resources available to the team. This process resulted in a set of functional requirements, non-functional requirements, and anti-requirements. Additional details about these specifications are available on the team's blog.

Table 1 displays the list of functional requirements specified at the beginning of the project.

Table 1: Functional Requirements.

ID	Requirement
FR01	The system must measure the patient's body temperature.
FR02	The system must measure the patient's heart rate.
FR03	The system must measure the patient's oxygen saturation.
FR04	The system must measure the patient's blood pressure.
FR05	The system must have audio output.
FR06	The system must have audio input.
FR07	The system must control if the doors should open or close.
FR08	The system must have a screen.
FR09	The system must have WiFi connection.
FR10	The system must indicate visually when the patient should speak.
FR11	The system must be plugged into the outlet.
FR12	The system must have space to contain an oximeter and cardiac frequency sensor.
FR13	The system must have space to contain blood pressure equipment.
FR14	The system must have space to contain a temperature sensor.
FR15	The system must have space to contain a microphone.
FR16	The system must have space to contain a sound box.
FR17	The system must have at least 2 spaces to contain a speaker.
FR18	The system must have space to contain a camera (non obligatory).
FR19	The space that contains the blood pressure equipment should have a little door.
FR20	The space that contains the oximeter and cardiac frequency sensor should have a little door.
FR21	Both doors must lock when closed.
FR22	The system must have a space for a screen.
FR23	The drawer opening will be controlled by a servo motor.
FR24	The door opening will be controlled by a servo motor.
FR25	The system must contain a website where healthcare professionals can access patients' information.
FR26	The website must have a login page.

Table 1 continued from previous page

ID	Requirement
FR27	The website must allow users to authenticate to get access.
FR28	The website must allow users to search for patient's medical screenings.
FR29	The website must allow users to see all collected data in a patient's medical screening.
FR30	The website must allow users to see a patient's medical screening history.
FR31	The website must allow users to export a patient's medical record as a PDF.
FR32	The website must manage sessions and keep users logged in.
FR33	The website must have admin users who can manage less privileged users.
FR34	The website must allow new authorized users to register.
FR35	The system must identify patients via name and CPF.
FR36	The system must communicate via audio with the patient.
FR37	The system must be voice-controlled.
FR38	The system must show the questions in text form on the screen.
FR39	The system must start the medical screening after a specific voice command.
FR40	The communication between the system and the patient should be in English.
FR41	The system must ask pre-defined questions at the beginning of the medical screening.
FR42	The system must elaborate questions intelligently based on the patient's complaints.
FR43	The system must give the patient instructions about how to get vital signs monitored.
FR44	The system must show heart rate being measured in real time.
FR45	The system must show oxygen levels being measured in real time.
FR46	The system must show temperature being measured in real time.
FR47	The system must end the medical screening by timeout or user's command.
FR48	The system must keep a history of a patient's medical screenings.
FR49	The system must detect the presence of a patient via camera (non obligatory).
FR50	The system must perform person recognition via camera (non obligatory).

Table 2 displays the list of non functional requirements specified at the beginning of the project.

Table 2: Non Functional Requirements.

ID	Requirement
NFR01	All hinges and locks must be made from non-brittle materials to avoid cracking under stress.
NFR02	All compartments must be accessible for maintenance without requiring specialized tools.
NFR03	The total system dimensions must not exceed 60 × 80 × 20 cm (to ensure portability).
NFR04	The total weight of the system must not exceed 12 kg.
NFR05	Access to sensors must allow easy patient placement without excessive effort.
NFR06	The system must include handles or grips to facilitate safe carrying.
NFR07	Doors and panels must have stops or soft-close mechanisms to prevent sudden slamming.
NFR08	Internal components must be clearly identifiable to facilitate replacement or maintenance.
NFR09	The system must handle authentication using JWTs (JSON Web Tokens).
NFR10	The web application must be in English.
NFR11	The web application must be written in PHP using the Laravel framework.
NFR12	The server backend must be written in Python 3.
NFR13	The Raspberry Pi application must be written in Python 3.
NFR14	The temperature must be measured in Celsius.
NFR15	The system must measure temperature using an infrared sensor (MLX90614).
NFR16	The system must measure blood pressure using a HEM-7142 monitor.
NFR17	The system must measure heart rate and blood oxygen level using a MAX30100 sensor.
NFR18	The speaker must output at least 5 Watts.
NFR19	The microphone must capture audio from a minimum distance of 40 cm.
NFR20	The user shall remain at a minimum distance of 10–20 cm from the temperature sensor when measuring the temperature.
NFR21	The system must use PostgreSQL as the database.
NFR22	The website pages must have at least 4.5:1 contrast ratio for normal text.
NFR23	The website pages must have at least 3:1 contrast ratio for large text.
NFR24	The website must have its text with a minimum size of 16px.

3 Development

The following section details the development of the project.

3.1 System Operational Workflow

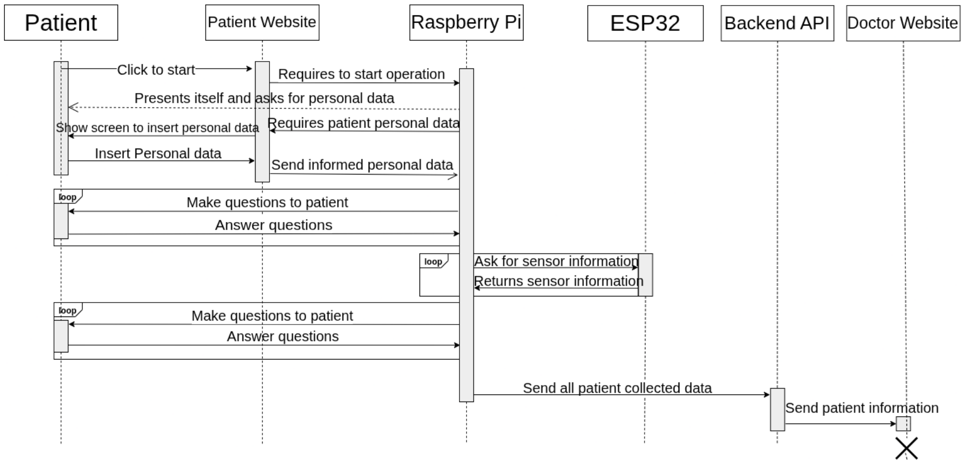

Figure 3: sequence diagram.

To integrate the mechanical, hardware, and software modules described in the following sections, the Anamnesis Totem is architected in a modular way. This ensures that physical actuation is strictly synchronized with the logical requirements of the medical triage. Throughout the whole process, totem talks to the user via speakers by the Piper TTS engine, and the user's voice input is transcribed to text by the Vosk STT engine. The complete operational routine is divided into five distinct stages:

1. **Idle & Detection State:** The user clicks on the start button on the graphical user interface (GUI) on the screen.

2. **Predetermined Questionnaire** The totem initiates the conversation, asking for relevant personal information for both identification and the anamnesis process. For name and CPF, the patient inputs the information by typing into the GUI on the screen. For all remaining questions, the input is done via speech.

3. **Physical Measurement & Actuation:** The triage protocol requires vital signs data. To achieve that, the Raspberry PI sends a command to the ESP32, which handles activating the necessary motors and operating the sensors. Furthermore, the totem guides the user via voice and screen instructions on how to use the sensors. Lastly, the data is retrieved and saved for later storage.

4. **Adaptive Anamnesis Interview:** The core logic utilizes the LLM (Gemini) to process the user's spoken input. Unlike a static questionnaire, the system dynamically generates the next most clinically relevant question based on the previous answer. This loop continues until the AI determines that sufficient context has been gathered or a fixed maximum number of questions has been reached.

5. **Conclusion & Reporting:** Once all necessary data is collected, the system compiles a structured report containing the patient's personal data, the measured vital signs, and a summary of the conversation. This information is then transmitted to the server for medical review, and the totem returns to an idle state.

3.2 Mechanics

The first stage of development focused on designing and assembling the project's mechanical structure. The process began with the creation of the CAD model in SolidWorks, which allowed the team to visualize the full layout of the system and validate the placement of sensors, doors, compartments, and internal components. The first compartment contains the box with the armlet. This box is opened by a servo motor. This compartment also contains a small drawer with an oximeter sensor. The drawer slides out automatically when the system is activated, allowing the user to place their finger on the sensor for measurement. This model provides a clear overview of the mechanical design and guided the construction of the physical structure, as shown in figure 4.

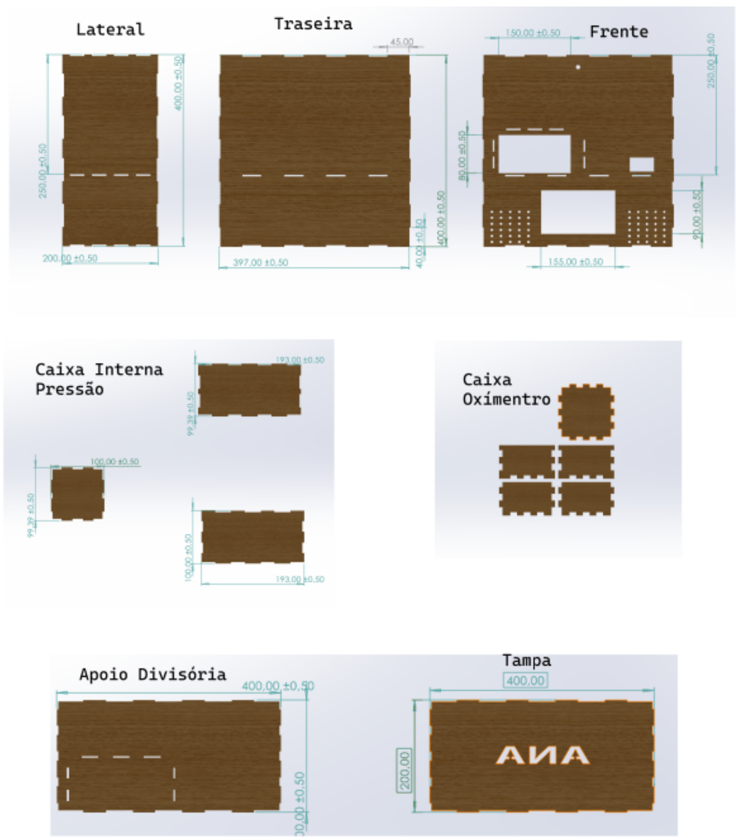

Figure 4: Mechanic overview.

3.2.1 Internal Box

In order to allow the patient to retrieve the armlet without having visibility of the other compartments, an internal box was modeled. Figure 5.

A 3D-printed hinge Design was implemented to operate the door mechanism. This solution enabled smooth and controlled opening from the inside outward, driven directly by a servo motor, ensuring reliable and precise movement.

3.2.2 Small Drawer

The small drawer stores the oximeter and heart rate sensor. Its opening mechanism uses a servo motor connected to the drawer by a wire, which pulls it outward and pushes it back into place. This keeps the devices protected while allowing controlled access during the screening process.

Figure 5: 3D printed parts.

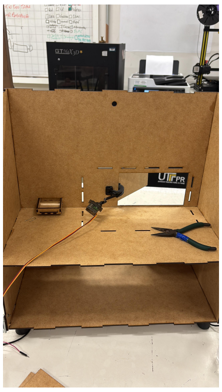

Figure 6: First Image

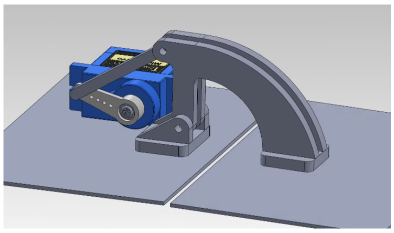

Figure 7: Second Image

3.2.3 Bottom Compartment

At the bottom of the station, a touchscreen tablet provides the main interface for user interaction. From this interface, patients can follow instructions, view ongoing measurements, and confirm or correct information captured by the voice system. Positioned alongside the tablet, the integrated speakers deliver clear audio prompts and feedback throughout the screening. Together, the tablet and speakers form a multimodal interaction layer that enhances usability, accessibility, and patient engagement during the automated assessment.

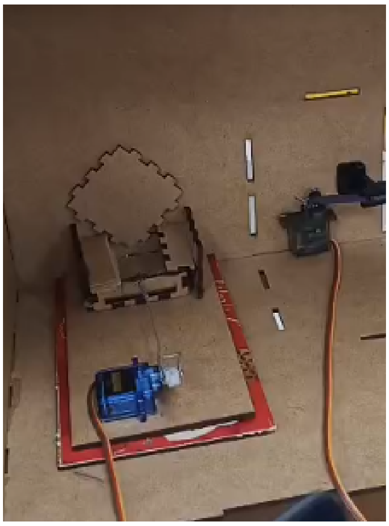

Figure 8: Drawer mechanic overview.

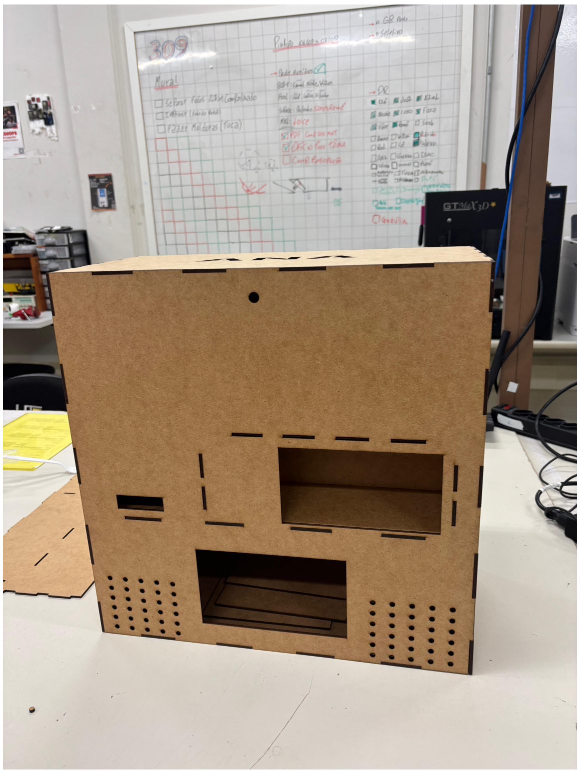

Figure 9: Totem design

3.3 Electronics/Hardware

The circuitry of the project was designed on a schematic and soldered in a universal Printed Circuit Board. The components that could not be soldered to the board were connected to the rest of the circuit using terminal connectors.

Figure 10 shows the schematic for the electronic circuit used in the project.

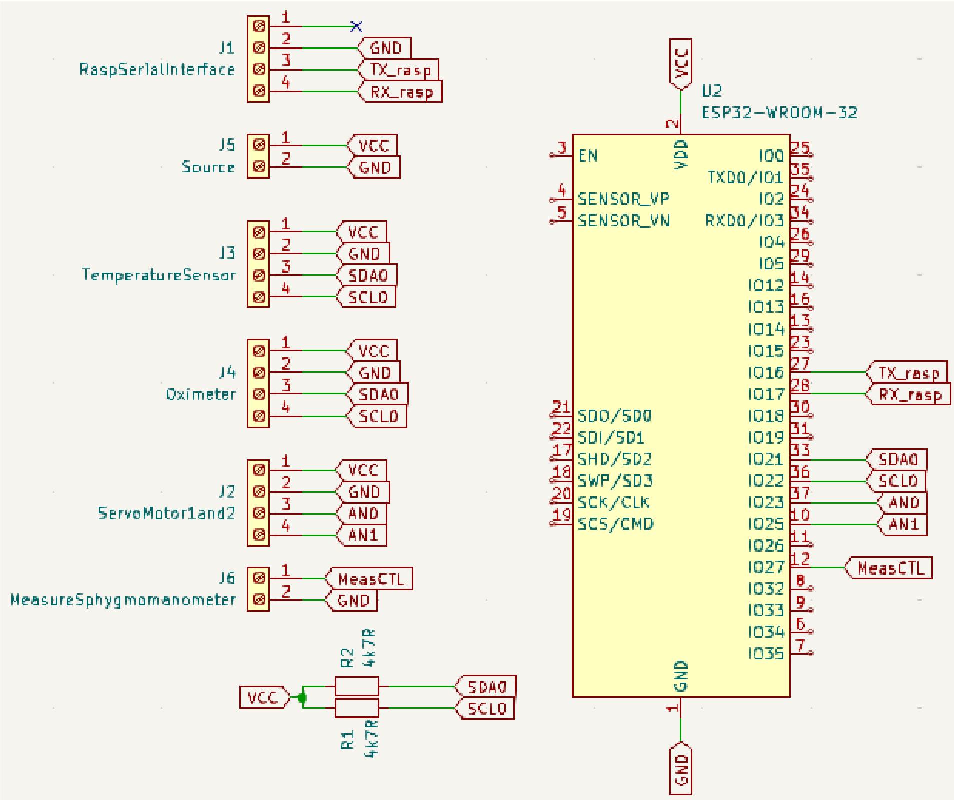

Figure 10: Full circuit schematic.

Figure 12 shows the internal layout of the automated medical screening prototype, where all sensors and control components are arranged inside the enclosure. On the right side, a USB camera is positioned to capture images. Directly above it, mounted at the top of the structure, is the non-contact temperature sensor (GY-906 / MLX90614). On the left side of the enclosure, the oximeter module (MAX30102) is installed to measure blood oxygen saturation and heart rate. Below it, a servo motor is integrated into the mechanical actuation system. In the central region, a relay module is used to activate the operation of the sphygmomanometer used for blood pressure measurement located near the upper left area of the panel. Next to it, a custom prototyping board hosts the ESP32 microcontroller, which acts as the central processing unit. It manages sensor acquisition, device control, communication, and data transmission to the web platform.

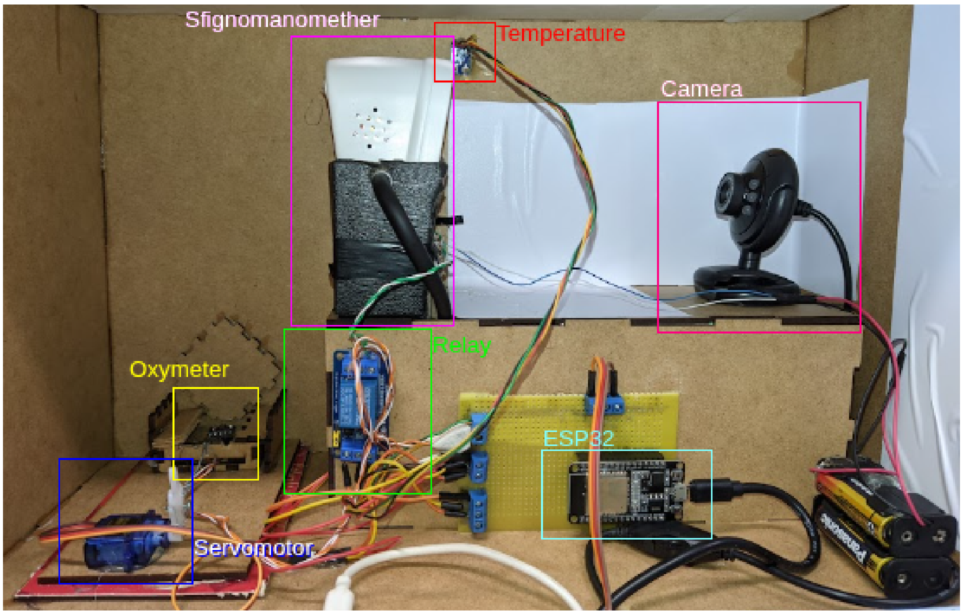

Figure 11: All eletronic components.

3.3.1 ESP32

The ESP32 microcontroller centralizes the instrumentation logic, serving as the direct interface for peripherals. It is responsible for controlling electromechanical components, driving the servomotors, and activating the external sphygmomanometer. Simultaneously, it performs real-time data acquisition from the medical sensors (temperature and oximetry). Communication is established via a physical UART interface, where the ESP32 acts as a transceiver: it listens for high-level instructions from the Raspberry Pi and responds by dispatching structured measurement packets, effectively offloading the hardware management overhead from the main processor.

3.3.2 Servomotors

The system integrates two servomotors to automate physical access to the medical instruments. The first servo actuates the door mechanism of the main enclosure, allowing user access to the blood pressure monitor's armlet. The second servo drives the motion of the retractable drawer housing the digital oximeter. These actuators ensure that sensors remain protected within the totem structure when not in use and are deployed only during the active measurement phase.

3.3.3 Temperature sensor

The system performs non-contact temperature measurement using the GY-906 infrared thermometer module[5] , which is based on the MLX90614 sensor. This device measures temperature by detecting the infrared radiation naturally emitted by objects such as human skin. Its built-in thermopile and signalconditioning eletronics convert the received IR energy into a digital temperature value, allowing accurate reading without physical contact.

3.3.4 Oximeter

The system measures blood oxygen saturation using the MAX30102 optical oximeter module[6]. The sensor operates by emitting red and infrared light into the user's fingertip and detecting the amount of light absorbed by the blood. Variations in absorption correspond to changes in blood volume with each heartbeat. By comparing the absorption rates of the two wavelengths, the modules computes the ratio that correlates with arterial oxygen saturation(SpO_2).

The MAX30102 integrates LEDS, a photodiode, and a low-noise signal processing unit, allowing accurate measurements with minimal external components. The processed data is captured by the microcontroller, filtered , and then used to estimate both SpO_2 and pulse rate in real time. This approach provides a noninvasive, reliable, and fast method for collecting vital cardiovascular information during the screening session.

3.3.5 Relay and Sphygmomanometer

The relay is integrated with the commercial sphygmomanometer to automate the action normally performed by the user pressing the push-button. Connected in parallel with the original switch, the relay briefly closes the circuit to simulate a button press, allowing the ESP32 to trigger the start of the blood pressure measurement automatically.

3.3.6 Camera

The camera is used to extract the blood pressure measurement by visually interpreting the sphygmomanometer display. First, the captured image is converted to grayscale to reduce complexity. An adaptive threshold is then applied to isolate the active illuminated segments from the background. The system checks predetermined coordinate regions corresponding to each seven-segment digit on the display. By identifying which segments are active within these fixed areas, the algorithm reconstructs each digit and maps the detected pattern to its corresponding numerical value, producing the final systolic, diastolic, and pulse readings.

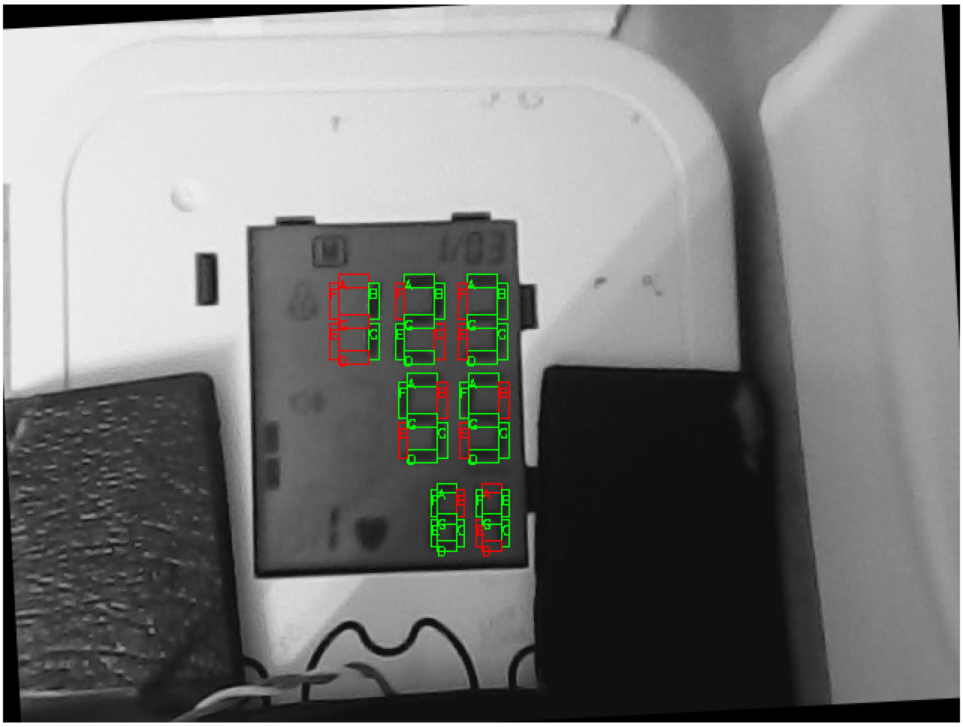

Figure 12: active segment detection.

3.3.7 Power

The system uses a dual power supply approach to support both the microcontrollers and the external medical device. A dedicated 6V supply, provided by four AA batteries, powers the commercial sphygmomanometer to ensure stable operation independent of the digital electronics. In parallel, a 5V power source feeds the Raspberry Pi 4, the ESP32, the speaker unit ant the LED light bar. All sensors, modules and actuators are powered directly from the ESP32 regulated output.

3.3.8 Tablet

A tablet is included in the system to guide the user throughout the screening process. It provides clear visual and textual instructions for each measurement step, ensuring proper positioning of the hand, finger, and forehead when using the oximeter, sphygmomanometer, and temperature sensor. The tablet also informs the user on how to interact with the system's voice interface, indicating when to speak and how to respond to the automated anamnesis questions. This component enhances usability and ensures that the measurements are performed correctly, contributing to the reliability of the overall screening procedure.

3.3.9 Internal LED

The LED light bar is integrated into the enclosure to provide uniform and controlled illumination inside the closed measurement compartment. Because the sphygmomanometer display must be captured by the camera for digital processing, consistent lighting is essential to ensure clear contrast between active segments and the background.

3.3.10 Microphone

Voice acquisition is handled by a USB microphone connected to the Raspberry Pi.

3.3.11 Audio speakers

The system incorporates a speaker set connected to the Raspberry Pi. They are responsible for rendering the synthesized speech, providing auditory feedback to the user throughout the ANAmnesis process.

3.4 Software

The project's software architecture is organized into three distinct operational layers: the **Totem Embedded System**, which encompasses the local middleware on the Raspberry Pi and the firmware on the ESP32; the **Web Application**, an interface designed for healthcare professionals to visualize and manage patient records; and the **Backend Server**, a centralized infrastructure responsible for data persistence (database management) and the orchestration of Speech-to-Text (STT) and Text-to-Speech (TTS) pipelines. The complete source code for all modules is available in the project's repositories [7].

3.4.1 Raspberry Pi 4

The Raspberry Pi 4 serves as the central processing unit and primary orchestrator of the ANAmnesis Totem. Beyond acting as middleware between the physical layer and the server, it acts as the system's host, explicitly managing the control flow of the medical screening routine. The device manages the application's state machine, coordinating when to request user input, trigger sensor readings via the UART protocol, or initiate data transmission. By centralizing these tasks, the Raspberry Pi ensures synchronization between hardware events and high-level logic. After completing an ANAmnesis session, the totem sends the patient's report to the server, which serves the web application where healthcare professionals can access patient data. To support this architecture, the operating system manages six concurrent services:

- **Voice Recognition Pipeline:** Manages audio acquisition via the peripheral microphone and streams the signal to a cloud-based Speech-to-Text (STT) server. To mitigate transcription inaccuracies, the raw text undergoes a semantic post-processing stage utilizing the LLM, which corrects phonetic and syntactic errors based on the context.

- **Audio Output:** Handles the Text-to-Speech (TTS) pipeline. It converts the text into audible speech using an external synthesis service. This audio is then rendered through the system's speakers, allowing communication with the patient.

- **Firmware Interface:** Manages the bidirectional serial (UART) link with the ESP32 microcontroller. This service dispatches control commands to trigger hardware routines (sensors and actuators) and parses incoming payloads. It handles both execution status reports (acknowledgments/errors) and vital signs data, such as temperature and heart-rate measurements.

- **Computer Vision Service:** Implements a deterministic Optical Character Recognition (OCR) pipeline designed to digitize readings from the external blood pressure monitor's LCD screen. This service utilizes the OpenCV library to execute a specific image processing routine: it applies Gaussian blurring and adaptive thresholding to binarize the image and isolate the active display segments. By analyzing pixel density within fixed Regions of Interest (ROI), the algorithm decodes the 7-segment logic to extract systolic, diastolic, and pulse values.

- **LLM Interface:** Manages the HTTP communication with the Google Gemini API[8], explicitly handling context injection and prompt engineering. This service performs two primary functions: first, it executes information extraction from user responses, parsing semantic content into structured data for the local system. Second, it directs the model to determine the most relevant subsequent diagnostic question based on the triage protocol. For instance, if a patient reports a cough, the system dynamically adapts the flow to generate specific inquiries regarding respiratory symptoms, ensuring the investigation remains clinically coherent.

- **User Screen Interface:** Renders the front-end Graphical User Interface (GUI). This web-based application acts as the primary interaction layer, allowing users to manage the session lifecycle (initiation and cancellation) and receive visual guidance for sensor usage. It also displays real-time transcriptions of the system's speech to enhance accessibility. The interface communicates with the backend via a persistent WebSocket connection, ensuring low-latency updates and immediate feedback.

3.4.2 STT Server

The Speech-to-Text (STT) component is deployed as a remote microservice dedicated to transcribing user audio. It is built upon the **Vosk API**[9], a lightweight speech recognition toolkit. Specifically, the system utilizes the *vosk-model-en-us-0.22* model, configured to process audio signals at a sampling rate of 16 kHz. To facilitate real-time processing, the service exposes a **WebSocket interface**. This persistent connection allows the client to stream raw audio chunks continuously, enabling incremental transcription. Furthermore, the server maintains the model in memory, eliminating initialization latency for each request and ensuring rapid text inference from the input stream.

3.4.3 TTS Server

To convert the textual commands and questions into audible speech, the server employs the **Piper TTS**[10] library. The system utilizes the *en_US-amy-medium.onnx* neural network model, chosen for its balance between voice quality and inference speed. The synthesis pipeline processes the text input and outputs uncompressed audio in the WAV container format. To ensure responsiveness, the class structure supports direct binary stream generation, facilitating the immediate transfer of audio data over the network without the latency associated with file system storage.

3.4.4 Report Server

The report server is the backbone of the interaction between the healthcare professionals and patient data. Making use of a PostgreSQL database instance, the server stores patient, report and employee data.

A core requirement for this section of the system is that all the data is deemed sensitive, and therefore access to it must only be granted to employees with proper permissions.

Apart from creating patients and reports, the server also allows for generating a PDF containing the data related to the report.

3.4.5 Tablet Screen

The tablet screen represents the visual user interface of the totem, allowing the patient to start and cancel a session, as well as providing personally identifiable information such as name and the Brazilian *Cadastro de Pessoa Física* (CPF).

In terms of visual cues, the interface provides a transcription of the audio played to the patient on all steps. Moreover, the screen also indicates when the totem's microphone is actively listening for patient voice input. Finally, on steps that require the patient to collect vital signs, GIFs are displayed on the screen as a way to guide the user on how to properly operate the sensors.

The following images show some of the screens shown throughout a screening.

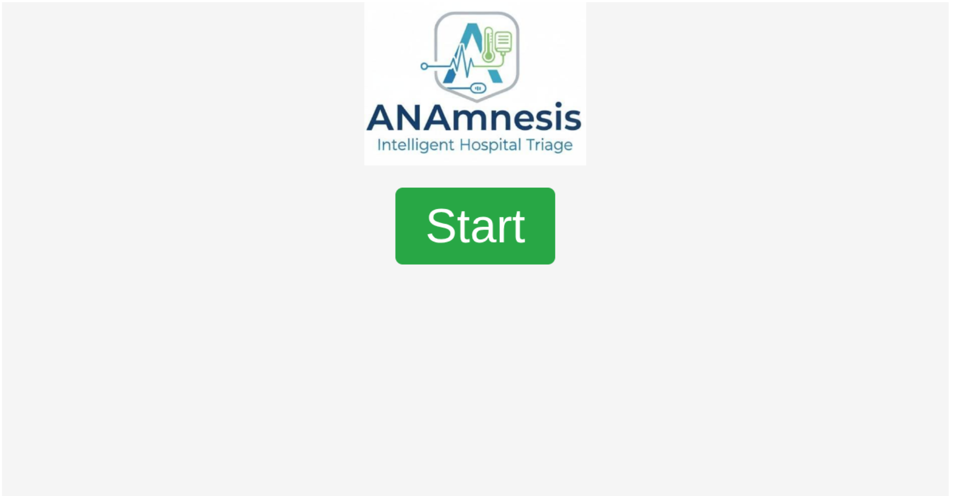

Figure 13: Idle screen.

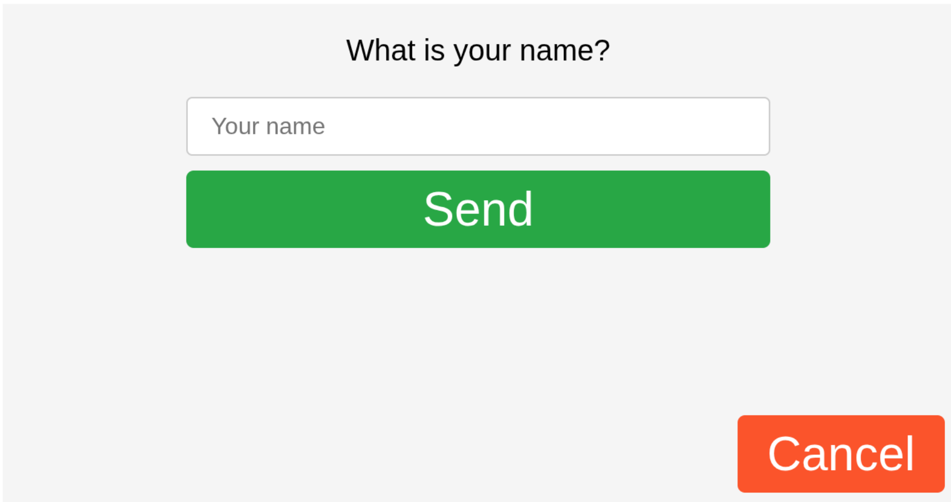

Figure 14: Name input screen.

Figure 15: Date of birth screen.

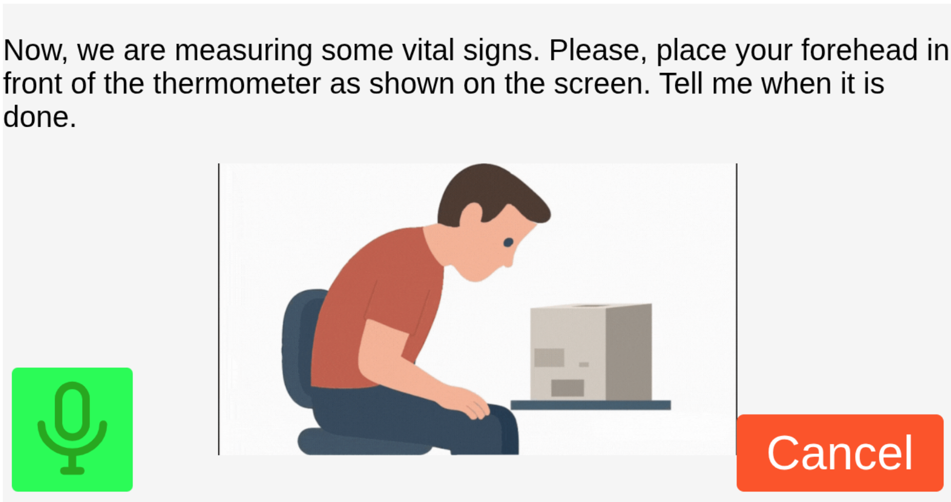

Figure 16: Temperature measuring screen.

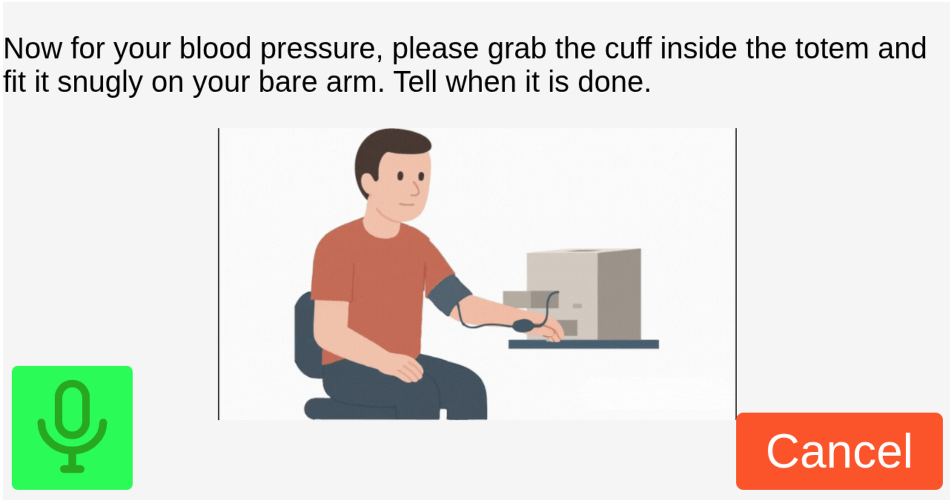

Figure 17: Pressure measuring screen.

3.4.6 Website

ANAmnesis Frontend is a healthcare-oriented web application built with React and modular TypeScript. It is a maintainable interface for hospital operations. It centralizes the management of patient records, consultations, clinical reports, and staff access flows, acting as the presentation layer that interacts with a RESTful backend through a secure connection. The system supports authenticated workflows, allowing authorized health professionals to view longitudinal patient data and export structured summaries (e.g., PDF) for clinical collaboration or archival.

The **login screen** welcomes the user and asks for basic access credentials. It presents a simple form where the person enters an email and a password and then submits. If the information is correct, the user is taken into the system and can see patient and report areas. If something is wrong, a clear message appears inviting the user to try again. The screen also offers a quiet path to the registration page for new users who still do not have access. The experience is direct, with focus on clarity, minimal distraction, and quick entry screens are shown in Figure 18.

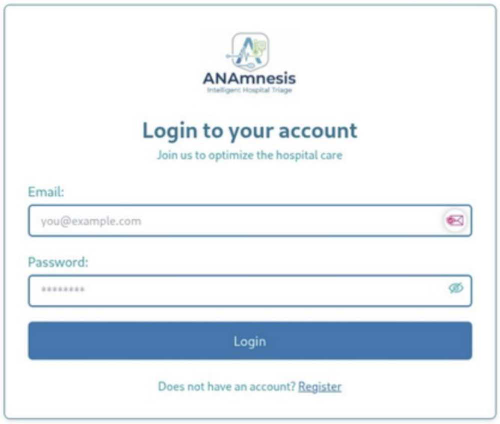

Figure 18: Main screen.

The **registration screen** helps a new user create an account. It asks for essential personal information like name, email, and a chosen password. After sending the form, the user receives feedback: either the account is created and access is granted or a notice explains what needs correction (such as an existing email). The tone is welcoming and supportive, guiding the person smoothly into becoming an authorized user of the application.

The **patient list** screen shows patients waiting for treatment in a clear, scrollable set of cards. A heading sets context, and a prominent button lets the user move to employee authorization. A search bar filters patients instantly by name, keeping interaction simple. Each patient card displays basic identity plus birth date, sex, temperature, oxygen level, and blood pressure for quick situational reading. Visual states handle loading, errors, and empty results gracefully. Clicking a card leads directly to that patient's history. This screen is shown in Figure 19.

Figure 19: Patients list screen.

The **patient history** screen shows a heading with the patient's name and a clean list of past reports in reverse order. Each card highlights the date and a brief summary of temperature, oxygen level, and blood pressure. The styling keeps focus on quick scanning, letting a user sense trends at a glance. A View button on each entry invites deeper inspection of that report's details. The layout balances clarity and compactness for efficient clinical review. It can be seen in 20

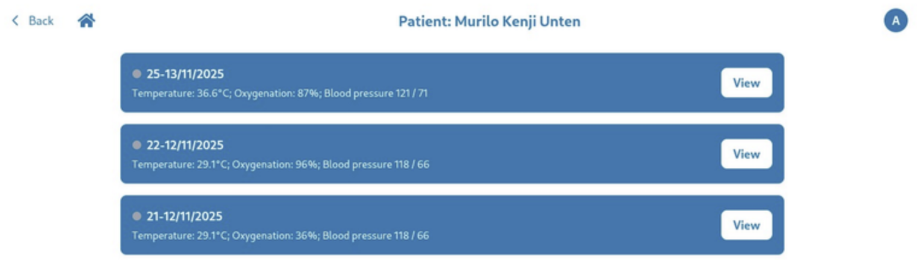

Figure 20: Patient history screen.

The **patient information** screen presents the patient's name with the latest report date and offers two clear actions: start a consultation or download a PDF. Information is laid out in tidy columns showing identity, age, occupation, gender, and basic measurements like weight, temperature, oxygen level, blood pressure, and heart rate. It can be seen in 21. Below, expandable sections list medications, allergies, and a structured interview (questions with answers) when available. The layout favors quick reading and gentle progression from summary to richer clinical context. It is represented in 22.

Figure 21: Patient information screen.

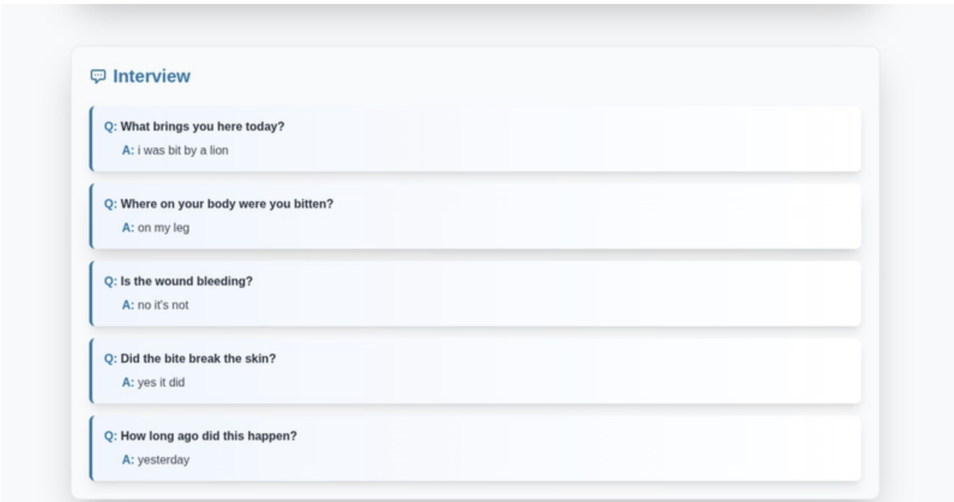

Figure 22: Expandable sections screen.

The **authorize employees** screen lists pending staff who still need their role confirmed. Each entry shows the person's name beside a simple selector where an appropriate role can be chosen. After picking a role, a single button applies the change and briefly shows a success message so the user knows it worked.

Clear states display when data is loading, if an error occurs, or when there are no people waiting. The layout keeps focus on quick approval rather than detailed profiles. It can be seen in 23.

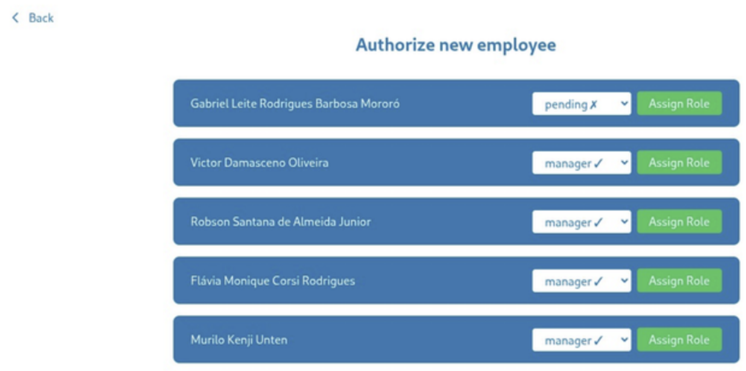

Figure 23: Expandable sections screen.

4 Results

The results of the ANAmnesis project were satisfactory. All the mandatory requirements were fulfilled and the team worked mostly on schedule throughout the development of the project. The entire process of development can be seen in the team's blog [2].

The performance of the totem also met the teams expectations, measuring the patients vital signs and making basic anamnesis, according to what was promised in the Project Charter. It's usability is also satisfactory, being simple and intuitive to the user, added with the application where the information generated by the totem can easily be seen.

4.1 Adaptations

From the planning phase to the execution, only one adjustment was necessary. The initial idea for the retrieval of blood pressure data was to "hack" the blood pressure monitor and get its information through its signals directly from the PCB. However, that proved unfeasible and would take a lot of time. Detecting and interpreting the signals was successful, but it missed some final adjustments. In the end, some trails were burnt on the pressure monitor and it would not be possible to resume this task due to time constraints. So, as written in the

[2] https://anamnesis-blog.netlify.app/

risk plan, a new blood pressure monitor was bought and data is taken using a camera and interpreting the images.

4.2 Schedule

The project development took place between August and December 2025, with some minor adjustments to the computer vision model made during this period. The work was organized around six main deliverables:

- **Deliverable 1:** Mechanical design;
- **Deliverable 2:** Mechanical project and Electronic design;
- **Deliverable 3:** Electronic project and Software design;
- **Deliverable 4:** Software project;
- **Deliverable 5:** Mechanical, Electronic and Software integration;
- **Deliverable 6:** Overall integration

The schedule was, overall, followed as planned; although some tasks took longer to complete—mainly the electronics part previously mentioned, due to its complexity—the total number of hours worked remained below the initial estimate, indicating that the planning and execution phases were successful.

Deliverable	**Estimated hours**	**Worked hours**
Deliverable 1	85:50	91:30
Deliverable 2	23:20	24:00
Deliverable 3	37:30	42:15
Deliverable 4	89:00	97:30
Deliverable 5	99:00	118:00
Deliverable 6	24:00	39:00
Total	**358:40**	**412:15**

Table 3: Work Hours Summary

4.3 Final view of the project

The following pictures show some of the views of the ANAmnesis machine. In Figure 24, it is possible to see the place where the user may get the blood pressure sensor, Figure 25 shows where the patient must put their finger to measure the oxygenation, while in Figure 26 the initial state of the totem is shown.

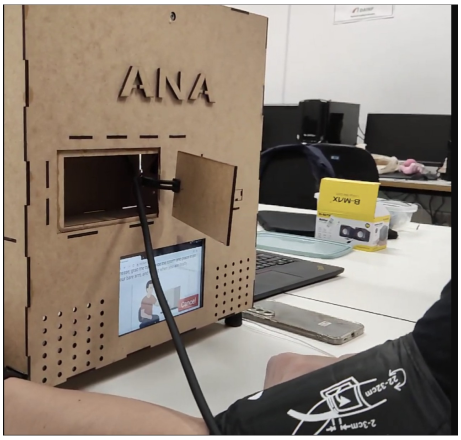

Figure 24: Totem with blood pressure monitor

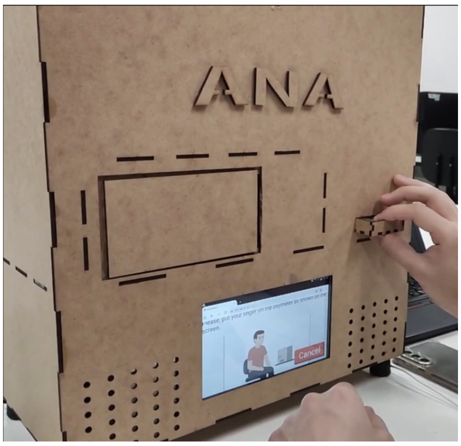

Figure 25: Totem with oximeter

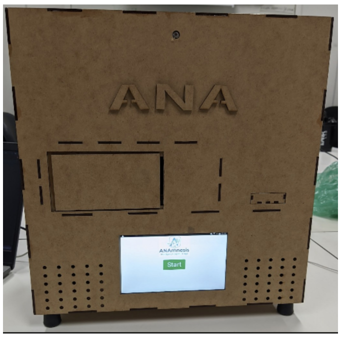

Figure 26: Initial view

4.4 Budget

Our initial plan was to spend R$ 895.00; however, the camera turned out to be cheaper than expected and the new blood pressure monitor also had a lower cost, resulting in a total expenditure below the original budget.

Table 5 shows all of the materials used.

Table 4: Budget

Item	Quantity	Price per unit	Total
Microphone	1	25	25
Speaker	1	20	20
Temperature sensor	1	70	70
Blood pressure monitor	1	60	60
Oximeter/heart rate monitor	1	30	30
Raspberry Pi 4	1	300	300
ESP-32	1	35	35
Camera	1	32	32
MDF	4	12.50	50
Servomotors	2	20	40
MicroSD card	1	40	40
Raspberry Pi power source	1	30	30
Screen	1	80	80
Design Gif Interaction	1	60	60
Cut MDF	-	-	-
Others	1	30	30
Total			**902**

Table 5: Budget

4.5 Risk Analysis and Response Plan

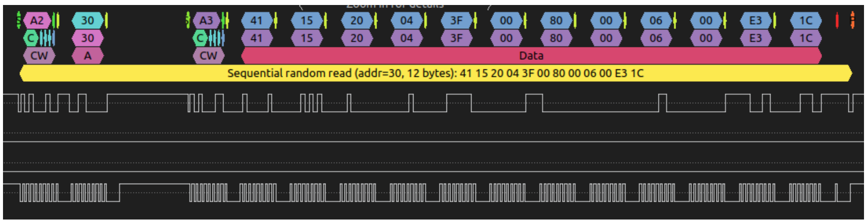

Figure 27: I2C Bus capture.

During the risk planning phase, one strategy initially considered was to obtain the blood pressure values by sniffing the I^2C bus between the sphygmomanometer's microcontroller and the EEPROM that stores the measurement records. However, practical tests showed that the available microcontroller did not have sufficient performance to operate as a reliable sniffer: on every clock cycle of the bus, an interrupt was triggered to buffer the captured data, which resulted in the loss of more than 60% of all transmitted bytes.

Despite this limitation, using a logic analyzer it was possible to correctly decode complete data packets present on the bus, as shown in the Figure 27. These packets had a total size of 14 bytes and followed a fixed structure: the last two bytes corresponded to a CRC-16 checksum; the first byte contained the systolic pressure value plus an offset of 25; the second byte contained the diastolic pressure; the third byte encoded the pulse value; the fourth byte represented the measurement identifier; and the remaining bytes formed a recurring pattern observed in all captures.

Although the packet structure was successfully interpreted, the inability to perform reliable real-time sniffing made this approach unfeasible, leading to the adoption of an alternative method based on image capture and segment recognition.

5 Conclusions and future work

5.1 Conclusions

The development of ANAmnesis was valuable to the enrichment of the team's experience in multiple areas, specially computer vision, machine learning and embedded systems.

The biggest challenges faced by the team were time management, which fortunately was balanced by the scheduling, the mechanics of the machine, especially to open and close the door and the drawer. Also, we had a problem to get signal from the blood pressure monitor, since there was no documentation and it was difficult to get the signals because the trails were to thin. In regards to that, the team had a problem related to the image interpretation due to the low luminosity inside the totem, which hindered the reading of the sphygmomanometer's display. The challenges were overcome by constant team communication and consultation with friends and acquaintances in the area.

The sections of the project that were developed more smoothly were notably the ones the team had planned the most in advance, which shows how crucial the planning phase was.

The team's work on ANAmnesis was satisfactory, since the functionalities worked well and the final result delivered what was promised in the Project Charter.

5.2 Future work

ANAmnesis is open to many improvements in the future, such as:

- A better computer vision model, with a larger sample blood pressure monitor displays to train it, improving its accuracy even further;
- Improvements on human-machine interaction;
- Automated classification of patient risks

6 Acknowledgments

The authors thank all acquaintances and friends who helped in the development of ANAmnesis. Specially people from "O projeto" who offered the lab and a lot of ideas on how to solve a lot of our problems.

References

[1] Giuseppe Andreoni, Alessandra Santangelo, Riccardo Sannicandro, and Alessandro Nizardo Chailly. Health pods for automated triage improve efficiency and satisfaction in nurses and patients. *Applied Sciences*, 15(2), 2025.

[2] Béatrix-May Balaban, Ioan Sacală, and Alina-Claudia Petrescu-Niţă. Triage-nlu: A natural language understanding system for clinical triage and intervention in multilingual emergency dialogues. *Future Internet*, 17(7), 2025.

[3] Djordje Gligorijevic, Jelena Stojanovic, Wayne Satz, Ivan Stojkovic, Kathrin Schreyer, Daniel Del Portal, and Zoran Obradovic. Deep attention model for triage of emergency department patients, 2018.

[4] Salman Razzaki, Adam Baker, Yura Perov, Katherine Middleton, Janie Baxter, Daniel Mullarkey, Davinder Sangar, Michael Taliercio, Mobasher Butt, Azeem Majeed, Arnold DoRosario, Megan Mahoney, and Saurabh Johri. A comparative study of artificial intelligence and human doctors for the purpose of triage and diagnosis, 2018.

[5] GY-906 Datasheet. https://www.melexis.com/-/media/files/documents/datasheets/mlx90614-datasheet-melexis.pdf.

[6] MAX30102 Datasheet. https://www.alldatasheet.com/datasheet-pdf/view/859400/MAXIM/MAX30102.html.

[7] ANAmnesis repositories. https://github.com/anamnesis-project.

[8] https://ai.google.dev/gemini-api/docs, author = Gemini API Documentation., owner = Gustavo, timestamp = 2014.11.01.

[9] Vosk. https://alphacephei.com/vosk/.

[10] Piper-TTS. https://github.com/OHF-Voice/piper1-gpl.