Federal Univeristy of Technology - Paraná – UTFPR
Academic Department of Electronics – DAELN
Academic Department of Informatics – DAINF
Computer Engineering
Integration Workshop 3 (ELEX22) – S72 – 2025/1

Technical Report
NeoBell

Alexei R. de Lara – alexeilara@alunos.utfpr.edu.br
Ana L. G. Berlingieri – anaberlingieri@alunos.utfpr.edu.br
Caio M. Raphaelli – caioraphaelli@alunos.utfpr.edu.br
Gabriel A. Spadafora – gspadafora@alunos.utfpr.edu.br
Luiz E. P. de Souza – luiz.2021@alunos.utfpr.edu.br
Raian H. G. Pereira – raian@alunos.utfpr.edu.br

June 2025

Abstract

The NeoBell system is conceived as a holistic, integrated solution designed to modernize the residential entryway experience. It moves beyond the limitations of traditional doorbells and simple package boxes by combining intelligent interaction processing, asynchronous video messaging, and secure automated package handling into a unified physical unit supported by robust software infrastructure. This system does not function as a real-time intercom; instead, it acts as an intelligent gatekeeper and messaging service. The project is implemented using the Radxa Rock 5C SBC(Single Board Computer) and various other electronical parts that work together seamlessly to address the problems outlined previously. We will explore the mechanical structure, the logic driving interactions, the processes for identifying visitors and verifying deliveries, and the digital tools that provide control and information to the resident.

1 Introduction

Brazil's e-commerce sector in Brazil has seen phenomenal growth since the pandemic. B2C revenue soared from around R$170 billion in 2022 to nearly R$186 billion in 2023—a 9.5% jump—after spiking 40% during the COVID-19 lockdowns.[1] Over the longer term, online retail nearly tripled from R$70 billion in 2018 to R$185 billion by 2023. Pandemic-era restrictions accelerated digital adoption among consumers and retailers, and while growth has stabilized (hovering around 7% of total retail sales), projections still point to annual expansion of around 14% through 2026. Despite this boom, Brazil's e-commerce ecosystem faces serious challenges. Cargo theft is one of the biggest concerns.[2]

In today's connected world, the way we interact with our homes and manage deliveries is rapidly evolving, yet traditional home entrance systems like basic doorbells and intercoms often lag behind. These legacy systems remain dependent on the resident's immediate availability, lacking mechanisms for asynchronous communication or secure package handling. This gap results in missed connections, inefficient deliveries, and notable security vulnerabilities, such as package theft ("porch piracy"), which remains a significant concern in the growing online retail landscape.

The NeoBell project is designed to address these deficiencies by providing a unified, intelligent solution. It redefines the residential entryway by integrating local AI-driven interaction management, asynchronous video messaging, and a secure, automated package delivery system. The system operates autonomously, serving as an intelligent gatekeeper that can securely handle a wide range of interactions without requiring real-time intervention from the resident.

2 The Project

2.1 System Components

2.1.1 NeoBell Module & Secure Compartment

The physical unit is the primary interface with the outside world. It is designed for external mounting and houses all necessary hardware, including the SBC, camera, microphone, speaker, and actuators.

NeoBell Control Unit This is the intelligent core of the physical device. It autonomously manages initial interactions using local AI to determine visitor intent (e.g., leaving a message, making a delivery) and guides them through the appropriate workflow.

Secure Delivery Compartment This is a two-stage "airlock" system for package handling.

- **Staging Box (Compartment 1):** The external, temporary holding area. After a delivery is authorized, this compartment unlocks for the delivery person to deposit the package.
- **Secure Box (Compartment 2):** The final, secure storage area accessible only from inside the residence. An internal camera in the Staging Box performs a verification scan before a trapdoor mechanism transfers the package from the Staging Box to the Secure Box, ensuring the correct package was received. Retrieval is handled by the resident using an authorized RFID tag.

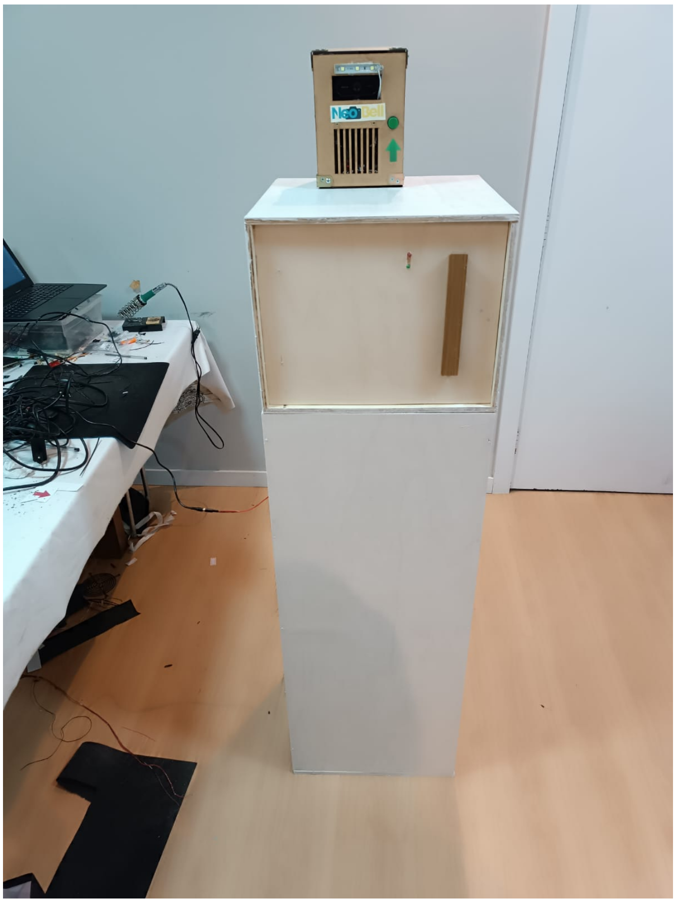


Figure 1: Image of the Full Project

2.1.2 Cloud Backend (AWS)

While NeoBell emphasizes local processing for real-time interactions, the Cloud Backend is indispensable for data persistence, synchronization, and providing scalable services to the mobile app and device.

Role The backend acts as the central data repository, communication orchestrator, and provider of supplementary services. It mediates all communication between the mobile app and the NeoBell module, ensuring data consistency and security.

2.1.3 Mobile Application (Flutter)

The NeoBell Mobile Application, developed for Android using the Flutter framework, is the central hub for residents to monitor, manage, and configure their system.

Functionality The app provides a user-friendly interface to access information gathered by the physical device and control its functionalities remotely. Key features include:

- Viewing and playing back video messages.
- Managing user access for residents and defining permissions for recognized visitors.
- Adding and tracking expected package deliveries.
- Receiving real-time push notifications for important events (doorbell, delivery, etc.).

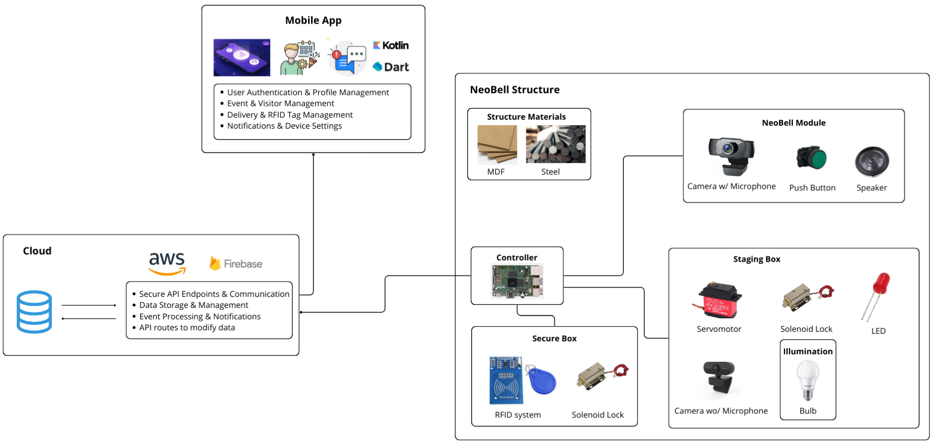

Mobile App
Kotlin
Dart
• User Authentication & Profile Management
• Event & Visitor Management
• Delivery & RFID Tag Management
• Notifications & Device Settings
Cloud
aws
Firebase
• Secure API Endpoints & Communication
• Data Storage & Management
• Event Processing & Notifications
• API routes to modify data
NeoBell Structure
Structure Materials
MDF
Steel
NeoBell Module
Camera w/ Microphone
Push Button
Speaker
Controller
Staging Box
Servomotor
Solenoid Lock
LED
Secure Box
RFID system
Solenoid Lock
Camera wo/ Microphone
Illumination
Bulb

Figure 2: System Diagram of NeoBell Project

2.2 Requirements

This section outlines the essential requirements identified for the system, encompassing both functional and non-functional specifications. These requirements were established based on market research, technical constraints, and user experience considerations. The specifications are categorized to address various aspects of the system and are separated by which part of the system they refer to, whether its mechanical, hardware or software. These requirements guided the system's development and served as benchmarks for evaluating the final implementation. These requirements are categorized into different types, including Functional Requirements (FR), which define the specific functionalities the system must perform, and Non-Functional Requirements (NFR). The tables only show an abridged version of the mandatory requirements of the system, which were all achieved by the team; a full list of the requirements, including some optionals is in our blog. [3]

Table 1: Abridged Key Mechanical Requirements

ID	Description of Requirement
FR-MECH-001	Boxes must fit small/medium packages up to 10 kg with internal dimensions ≥ 25×25×20 cm.
FR-MECH-002	Box 1 must have a motorized 12V lockable door.
FR-MECH-005	Structure must support at least 10 kg without deformation.
FR-MECH-007	Interior must include cushioning for fragile items.
FR-MECH-008	Enclosure must be made of MDF or MDP wood panels.
FR-MECH-010	Hinges must be mounted internally to prevent tampering.
FR-MECH-015	Trapdoor must open within 2 seconds using a servo motor.
FR-MECH-018	Trapdoor gap must be ≤5 mm to avoid pinch hazards.
FR-MECH-021	A top-facing fixed-focus camera must capture package data (10–30 cm range).

3 System Architecture

As it was said earlier the project is divided into five main parts which will be explained in this section. The mechanical structure is the physical foundation of the project, the hardware is all of the electronics parts connected and controlled by the SBC, the firmware is the software that runs embedded in the SBC, controlling all the electronics in its main flow, the AI capabilities of voice and facial recognition and the communication with the cloud. The cloud backend mediates all data between the app and the NeoBell module and the Mobile App is the main way the owner or resident of the house interacts with his product.

3.1 Mechanical Structure

All physical parts of the project were constructed using MDF wood for its durability and cost-effectiveness.

3.1.1 NeoBell Main Module

The NeoBell serves as the primary external interface and the operational core of the NeoBell system, orchestrating all interactions with visitors and delivery personnel. Its front panel, as depicted in Figure 3, is designed for clear user engagement and houses the critical sensors and feedback mechanisms. Housed

Table 2: Hardware Requirements

ID	Description of Requirement
FR-HW-01	The system shall use the Radxa ROCK 5C Lite 4GB as the central processing unit, supporting local AI processing for OCR and Barcode/QR code reading.
FR-HW-02	The Radxa shall provide USB ports to connect two USB cameras for 1080p video capture.
FR-HW-03	The external USB camera with microphone shall capture 1080p video for OCR and Barcode/QR code reading.
FR-HW-04	The internal USB camera, mounted in Box 1, shall capture 1080p images of package labels in a dark environment with LED illumination.
FR-HW-05	The microphone in the cameras shall capture audio from visitors/delivery personnel at up to 1 meter.
FR-HW-06	The system shall include a 1W speaker to play voice instructions audible at 1 meter.
FR-HW-08	A 12V/5A power supply shall power the Radxa, cameras, speaker, servomotor, solenoid lock, and LEDs.
NFR-HW-01	The system shall operate without overheating (Radxa below 70°C).
NFR-HW-02	The push button shall be weather-resistant with clear tactile feedback.

Table 3: Abridged Key Firmware Requirements

ID	Description of Requirement
FR-FW-01	The single-board computer must support a Linux operating system.
FR-FW-02	The system must support AI libraries for tasks like OCR and barcode reading.
FR-FW-04	The system must capture 1080p images from both USB cameras.
FR-FW-05	The system must send 1080p images from both cameras to the cloud.
FR-FW-09	The system must control the internal lock.
FR-FW-12	The system must control the trapdoor mechanism.
FR-FW-14	The system must synchronize data with AWS cloud services.
FR-FW-19	The system must temporarily store data in the OS file system.
NFR-FW-01	The system must complete OCR/barcode processing on a 1080p image in $\leq$ 20 seconds using AI inference.
NFR-FW-05	Time-critical tasks (e.g., lock control, image capture) must be tested under peak load to identify bottlenecks.

Table 4: Abridged Key Cloud Requirements

ID	Description of Requirement
FR-CLD-001	Cloud must provide secure APIs for Mobile App login and data access (events, visitors, deliveries).
FR-CLD-002	Cloud must allow NeoBell devices to sync critical data (logs, settings, permissions).
FR-CLD-004	App must manage "Expected Delivery" records through a cloud API.
FR-CLD-007	Cloud must provide an API to register new NeoBell devices via the Mobile App.
FR-CLD-018	Push notifications must be sent to the App for critical events.
FR-CLD-019	Remote commands from the App must be securely forwarded to NeoBell devices.
NFR-CLD-001	Backend shall be implemented using Amazon Web Services (AWS).
NFR-CLD-010	Architecture must use scalable AWS services (API Gateway, Lambda, DynamoDB).
NFR-CLD-011	Sensitive API endpoints must require authentication.
NFR-CLD-016	APIs must follow RESTful principles and use JSON format.

Table 5: Abridged Key App Requirements

ID	Description of Requirement
FR-APP-001	App must allow registration of new resident accounts.
FR-APP-002	App must allow login with existing credentials.
FR-APP-003	App must provide password reset mechanism.
FR-APP-006	App must show a chronological list of recent events.
FR-APP-008	App must play the video for a selected message.
FR-APP-013	App must allow deletion of a video message.
FR-APP-014	App must display a list of identified visitors.
FR-APP-017	App must allow users to change visitor recording permissions.
FR-APP-018	App must allow users to request deletion of a visitor's data.
FR-APP-019	App must allow registration of expected deliveries.
FR-APP-021	App must show delivery status for each expected delivery.
FR-APP-024	App must receive push notifications from the cloud.
FR-APP-027	App must display operational status of the NeoBell device.
NFR-APP-001	The app must be developed using Flutter.
NFR-APP-016	Auth tokens must be stored securely using secure storage packages.

within a durable enclosure, this unit functions as an automated receptionist. The overall external dimensions of this Module have been precisely calculated to be 15 cm (Width) x 20 cm (Height) x 12,9 cm (Depth). These proportions provide adequate internal space for all integrated functionalities while maintaining a compact and practical form factor suitable for an entryway. When a visitor presses the activation button, the Radxa board and custom PCB work in concert to manage the camera, microphone, and speaker, efficiently coordinating the system's response based on the processed visitor intent.

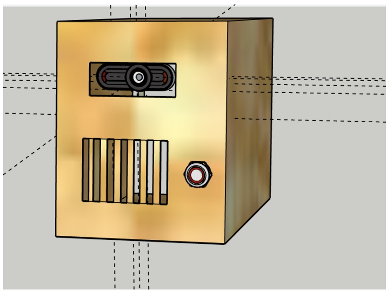


Figure 3: NeoBell Main Module schematic

3.1.2 Staging Box

The Staging Box is a critical component of the NeoBell system, engineered to serve as the initial, secure point of contact for package deposits. Its design balances user-friendliness for delivery personnel with the internal mechanisms required for automated verification.The external appearance of the Staging Box, as depicted in its initial schematic, is a rectangular enclosure with dimensions of 40.0 cm (Width) x 40.0 cm (Depth) x 30.0 cm (Height). This size provides a versatile internal volume for common package sizes. The overall simple rectangular form is chosen for ease of initial construction and integration of internal components.

3.1.3 Trapdoor Mechanism

The crucial transfer of verified packages from Delivery Compartment (Staging Box) to Collect Compartment (Secure Box) is managed by the Trapdoor Mechanism. This automated system is designed for reliability and to maintain the security integrity between the two compartments. The accompanying schematics illustrate its construction and operation. As shown in the view of the underside of Delivery Compartment 1, the trapdoor itself is a panel, approximately 28.0 cm x 26.0 cm, forming the floor of the staging area.

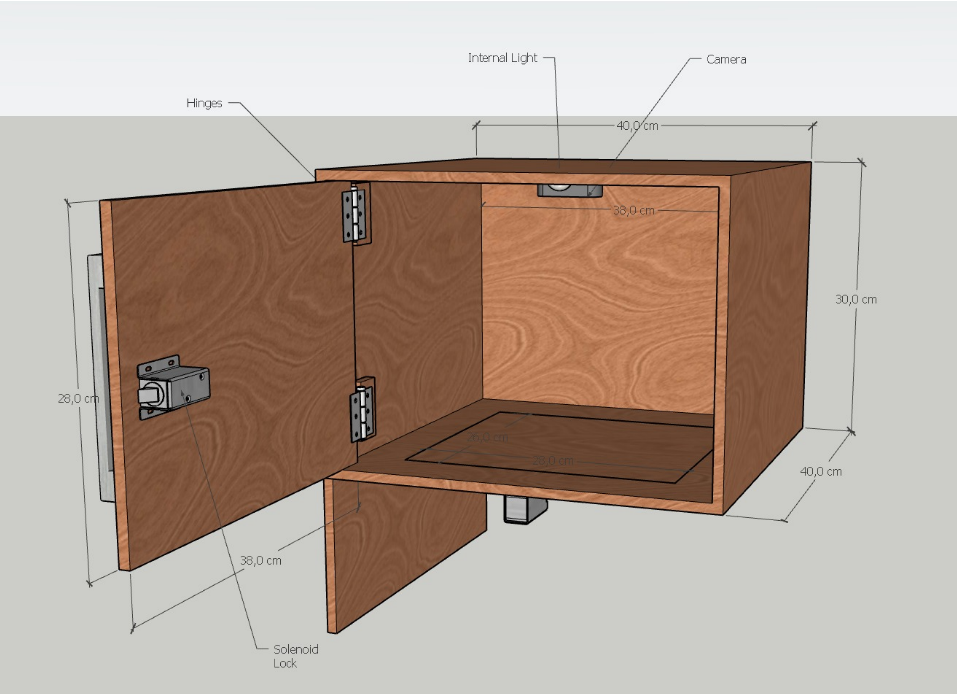


Figure 4: Staging Box Open

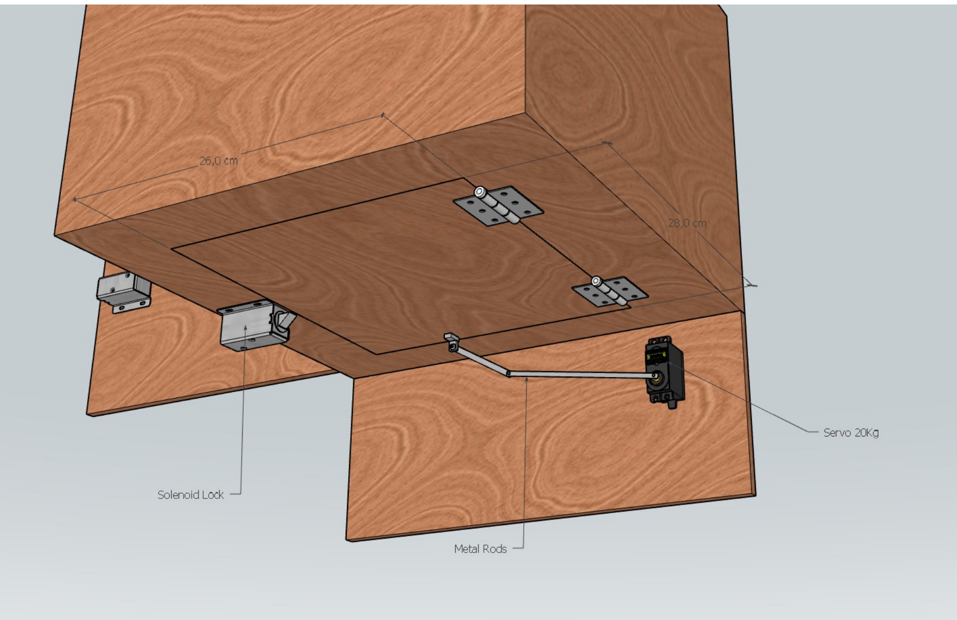


Figure 5: Trapdoor Mechanism

3.1.4 Secure Box

Following the automated transfer via the trapdoor from the Staging Box, packages arrive in Collect Compartment (Secure Box). This compartment is the final, highly secure holding area, designed exclusively for package retrieval by the resident from inside the property. The schematic shows a taller, vertical structure with overall dimensions of 40.0 cm (Width) x 40.0 cm (Depth) x 100.0 cm (Height). This extended height provides a generous internal volume to accommodate multiple packages or larger items that have passed through the initial staging box. This Secure Box, show in Figure 6 completes the automated delivery workflow by providing a safe, resident-accessible final storage location for verified packages, with access conveniently managed via NFC technology.

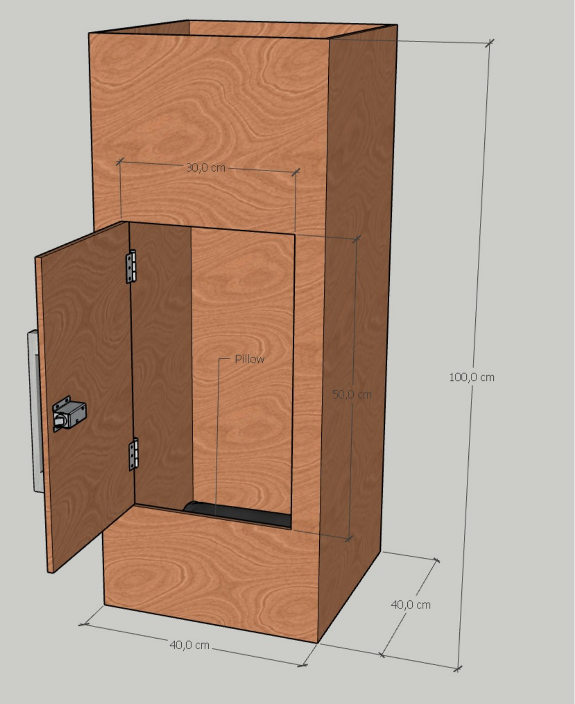


Figure 6: Secure Box

3.2 Hardware Electronics

The electronic architecture of the system is built around a collection of specialized components that work together to control the NeoBell mechanism, as well as the external data acquisition for package handling and security. It integrates both core components and peripherals to ensure seamless operation.

3.2.1 Core Hardware

The hardware controlling the NeoBell mechanism is engineered for precision and reliability, ensuring accurate package handling and security processes.

- **Radxa Rock 5C**: Acts as the central processing unit of the system, managing all hardware and software interactions through its GPIO interface. It processes facial recognition, voice commands, motor control, and manages the compartment logic and security systems.

- **USB Cameras (Full HD 1080p)**: Used for facial recognition and package validation. One camera is positioned at the front for visitor identification, and another is placed inside the delivery compartment to capture images of delivered packages. Both cameras support OCR for reading package labels and verifying tracking numbers.

- **Solenoid Locks**: These are used to secure various compartments:

  - **Delivery Compartment Lock (Staging Box)**: Controls access to the compartment until the user is authorized to deliver a package.
  - **Trapdoor Lock**: Locks the trapdoor until the package is validated.
  - **Collect Compartment Lock (Secure Box)**: Ensures only authorized users with valid RFID tags can access the secure compartment.

  These locks are controlled through GPIO pins with a driving circuit to safely manage the solenoids.

- **RFID Reader**: Verifies RFID tags for access to the collect compartment. It uses multiple GPIO pins for communication and authentication. For the connection, it is using an ESP-32.

- **Servo Motor TD-8120MG**: Used to control the trapdoor. The servo motor is responsible for moving the trapdoor into its horizontal position, with a metal shaft connected to the servo horn. This motor provides enough torque to lift the trapdoor reliably and smoothly.

3.2.2 Peripherals and User Interface Hardware

The user interface hardware consists mainly of external peripherals, all connected to the central microcontroller. These peripherals enhance the user experience by providing seamless interaction and feedback.

- **USB Microphone**: Captures audio input from visitors or delivery personnel, enabling voice interaction with the system.

- **Speaker**: A 1W speaker is integrated into the system to provide audible voice instructions, ensuring clear communication with users.

- **LED Indicators**: External LEDs indicate the operational status of the NeoBell system, such as the status of the door lock or other system conditions. These LEDs help users understand the current system state.

- **Push Button**: Used for user interaction, allowing manual triggering of the system to initiate specific actions, such as unlocking or activating the camera system.

Figure 7 presents the full schematic diagram of the NeoBell system's electronic architecture, detailing the interconnections between power management, sensors, drivers, motors, and the core microcontroller. Power management is handled by one 12V power supplies.

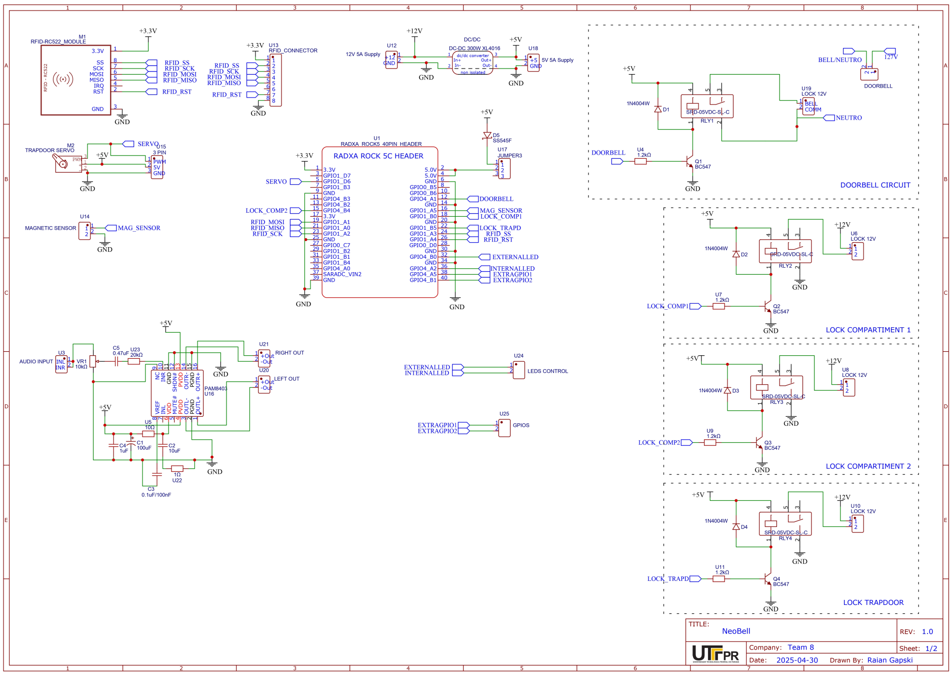


Figure 7: Schematic diagram of the NeoBell system's electronic architecture.

3.3 Firmware

The NeoBell firmware, running on the Radxa Rock 5C, uses a modular, event-driven architecture. This design allows for better resource management for concurrent AI tasks and easier updates for individual components. The firmware uses a multi-process or multi-threaded approach to handle I/O, AI processing, and network communication concurrently.

- **Voice Interaction**: The firmware handles both STT(Speech-To-Text) for detecting what the person interacting with it is saying using **sounddevice** and **cephei** as AI tools and TTS(Text-To-Speech) for the Module to answer the person through the speaker. It uses **subprocess** and **LangPack** as libraries for that. Figure 8 exemplifies the interactions.

- **Visitor Intent Recognition**: The firmware uses **Google Gemini API** for intent recognition based on the voice input by the user, detecting if the user is a visitor, delivery person or a third option which is not recognized.
- **Facial Recognition**: The firmware, through the front facing camera in the NeoBell module detects and identifies visitors using the **OpenCV** and **deepface** libraries. This then gets processed in the Cloud and checks if the visitor is a new visitor or an allowed one.
- **Package Scanning(Internal and External)**:The packages QR code or data matrices are checked by both the inside and outside camera, using a technique called OCR processing and the **QReader** library.
- **RFID identification**: The firmware, through an ESP-32 microcontroller, reads RFID tags that were registered in the app and opens the solenoid lock in the Secure box for package retrieval.
- **Trapdoor and other electronics**: The firmware, using PWM and GPIOs also controls all locks, cameras, LEDs and mainly the servomotor for closing the trapdoor.

3.4 Cloud Backend

The NeoBell cloud backend, built on Amazon Web Services (AWS), serves as the central nervous system for the entire ecosystem. It operates on a serverless, event-driven architecture designed for high scalability, security, and cost-efficiency. Its primary responsibility is to act as the authoritative source for data storage, business logic, user authentication, and as the communication hub between the NeoBell device and the mobile application.

3.4.1 Key Principles & Architecture

The architecture adopts a microservices pattern through serverless functions (AWS Lambda). This choice was driven by the need for modularity and scalability; each function is dedicated to a specific business domain (e.g., user management, device management, video message handling), allowing for independent development, deployment, and scaling. This model significantly reduces operational overhead and ensures that costs are directly tied to usage.

The key AWS services utilized are:

- **Amazon API Gateway:** Serves as the secure "front door" for all HTTP requests from the mobile application, managing endpoint routing, authorization, and traffic control.

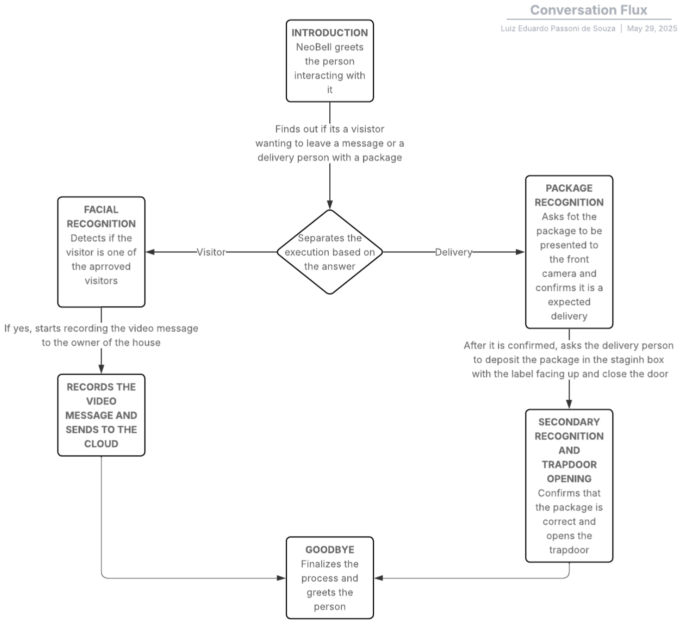

Conversation Flux
Luiz Eduardo Passoni de Souza | May 29, 2025
INTRODUCTION NeoBell greets the person interacting with it
Finds out if its a visistor wanting to leave a message or a delivery person with a package
Separates the execution based on the answer
Visitor
Delivery
FACIAL RECOGNITION Detects if the visitor is one of the aprroved visitors
PACKAGE RECOGNITION Asks fot the package to be presented to the front camera and confirms it is a expected delivery
If yes, starts recording the video message to the owner of the house
RECORDS THE VIDEO MESSAGE AND SENDS TO THE CLOUD
After it is confirmed, asks the delivery person to deposit the package in the staginh box with the label facing up and close the door
SECONDARY RECOGNITION AND TRAPDOOR OPENING Confirms that the package is correct and opens the trapdoor
GOODBYE Finalizes the process and greets the person

Figure 8: Conversation Flow

- **AWS Lambda:** The core of our serverless business logic. Specialized functions (e.g., `UserHandler`, `DeviceHandler`, `MessageHandler`, `SBCHelperHandler`) are triggered by events from API Gateway, AWS IoT Core, Amazon S3, or Cognito to execute specific tasks.

- **Amazon Cognito:** Manages the entire user identity lifecycle for the mobile application, including registration, login, password recovery, and the issuance of secure JWTs for API authentication.

- **Amazon DynamoDB:** Our primary NoSQL database. We utilize multiple tables optimized for specific access patterns, such as `NeoBellUsers` (profiles), `NeoBellDevices` (device registry), `DeviceUserLinks` (linking users to devices with "Owner" and "Resident" roles), `VideoMessages` (video metadata), and `ExpectedDeliveries` (package tracking).

- **Amazon S3:** Used for the durable storage of large binary objects, namely the video message files and visitor registration images. Access is strictly controlled via pre-signed URLs generated by Lambda functions.

- **AWS IoT Core:** The communication hub for our SBC devices. Devices connect securely using X.509 certificates and communicate via the MQTT protocol to send real-time events and receive commands. IoT Rules process these messages and trigger the appropriate backend Lambda functions.

- **Amazon SNS (Simple Notification Service):** Manages the dispatch of push notifications to users' mobile applications. It uses Firebase Cloud Messaging (FCM) as a bridge to deliver these alerts. After a user logs in, the Flutter app sends its unique Firebase device token to the backend via an HTTP API call. A Lambda function then stores this token in DynamoDB and registers an endpoint in SNS for that specific device. This allows the backend to send targeted notifications about important events.
- **AWS IAM (Identity and Access Management):** Enforces security through fine-grained access control. IAM Roles (e.g., `NeoBellLambdaExecutionRole`, `NeoBellIoTRuleActionRole`) grant the minimum necessary permissions for each AWS service to interact with others securely.

3.4.2 Communication Flows

Communication with the backend is handled through two primary channels, creating a separation between user-driven actions and real-time device events:

API REST (via API Gateway) The mobile application uses a RESTful API for all its functionality. All requests are stateless and protected by JWTs issued by Cognito, ensuring that only authenticated and authorized users can access data and perform actions.

MQTT (via AWS IoT Core) The SBC device uses the lightweight MQTT protocol for efficient, real-time, bi-directional communication. This channel is used for sending status updates, logs, and for request/response flows that do not involve large file uploads, such as verifying an NFC tag or checking a visitor's permission.

The complete documentation for the API, database tables, and system flows is maintained on our project's Notion blog, serving as a detailed technical reference for the development team.

3.5 Mobile App

The mobile application, developed using Flutter, is the primary command center through which users interact with and manage their NeoBell system. It communicates exclusively with the backend via the API Gateway after the user authenticates through Amazon Cognito.

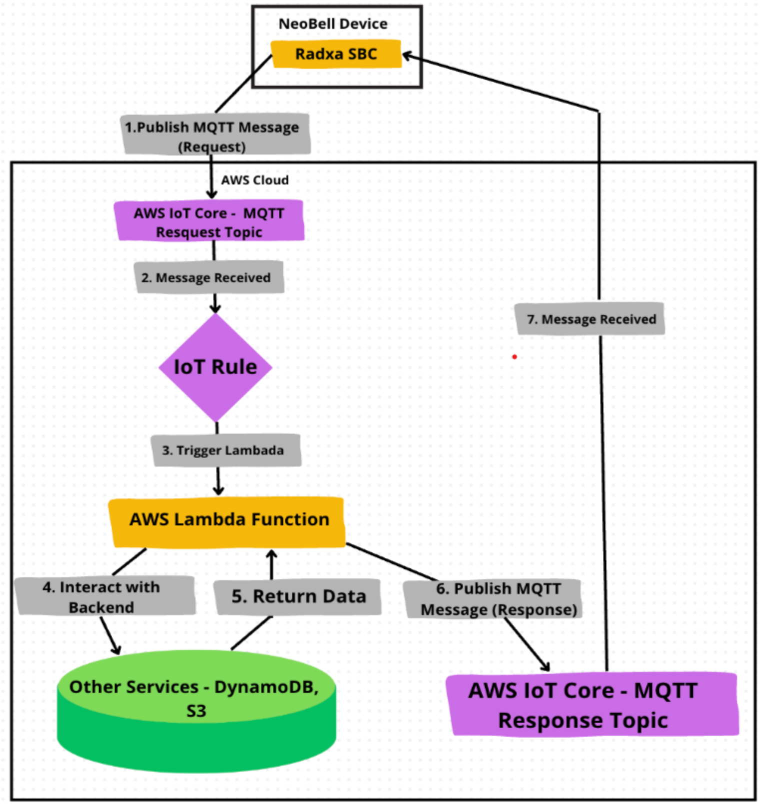

NeoBell Device
Radxa SBC
1.Publish MQTT Message (Request)
AWS Cloud
AWS IoT Core - MQTT Resquest Topic
2. Message Received
IoT Rule
3. Trigger Lambada
AWS Lambda Function
4. Interact with Backend
5. Return Data
6. Publish MQTT Message (Response)
Other Services - DynamoDB, S3
AWS IoT Core - MQTT Response Topic
7. Message Received

Figure 9: AWS Backend viewed by the SBC

3.5.1 Architecture and Technologies

To ensure a maintainable, testable, and scalable codebase, the application's architecture is built on established best practices within the Flutter community:

- **Clean Architecture:** This pattern guarantees a clear separation of concerns by dividing the code into distinct layers: Presentation (UI), Domain (Business Logic), and Data (Data Sources). The core principle is that dependencies flow inwards, making the core business logic independent of UI frameworks and data sources.

- **State Management:** We utilize the BLoC (Business Logic Component) / Cubit library for predictable and scalable state management. This separates business logic from the UI, making the application easier to test and reason about.

- **Navigation:** Screen routing is managed by GoRouter, which provides a robust, declarative API for URL-based navigation, deep linking, and route guarding.

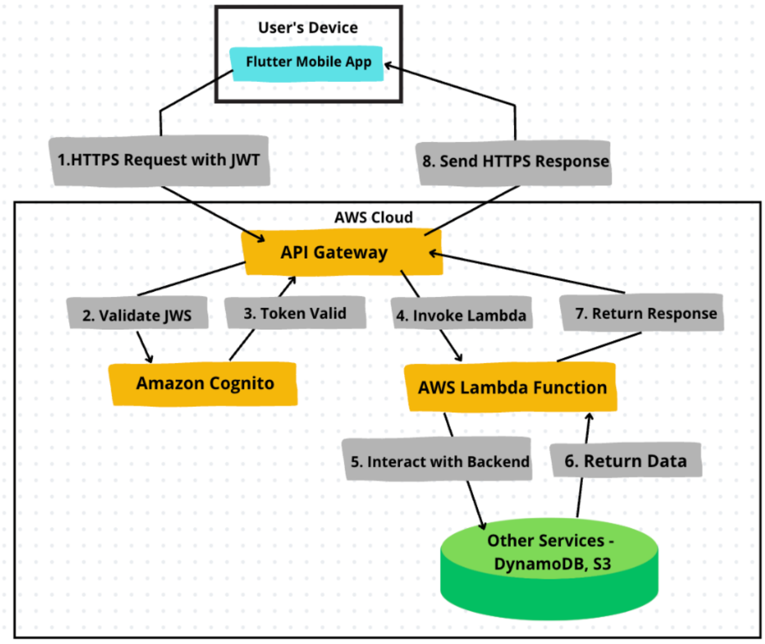


Figure 10: AWS Backend viewed by the App

- **Backend Communication:** Interaction with the NeoBell REST API is handled by a centralized service that uses the `http` package. Authentication is managed seamlessly by the AWS Amplify for Flutter library, which handles the Cognito authentication flows and secure storage of JWTs.

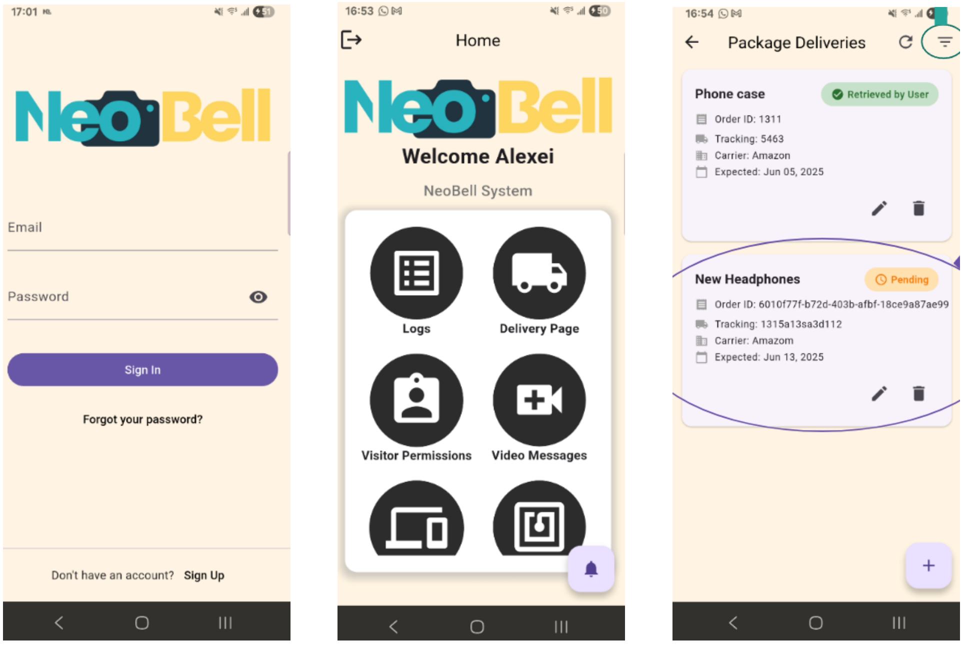
3.5.2 Key Functionalities

The application serves as a comprehensive dashboard for the NeoBell system, allowing the user to:

- **Manage Devices and Users:** Owners can view the status of their linked NeoBell devices, update settings like the friendly name, and manage access for other users ("Residents"). This feature provides homeowners with complete control over their access ecosystem.

- **Interact with Events:** Users can view and play video messages left by visitors, receive real-time push notifications for events like a video message recording or a package delivery, and review a historical log of all system activities.

- **Manage Deliveries and Permissions:** The app provides tools to add and track expected package deliveries and to manage permissions for visitors whose faces have been recognized by the system. Users can also register and name their personal NFC tags for access to the Secure Box.



Phone case
Retrieved by User
Order ID: 1311
Tracking: 5463
Carrier: Amazon
Expected: Jun 05, 2025

New Headphones
Pending
Order ID: 6010f77f-b72d-403b-afbf-18ce9a87ae99
Tracking: 1315a13sa3d112
Carrier: Amazom
Expected: Jun 13, 2025

(a) Login screen (b) Home screen (c) Delivery page screen

Figure 11: Main screens of the app

4 Results

The project met all mandatory requirements and implemented around 60% of the optional ones. Table 6 details a cost analysis, listing components, quantities, unit prices, and total costs, covering all essential items. Although the budget was slightly exceeded, it remained within acceptable limits, and all components performed well. The risk analysis was useful in addressing issues, and the project followed its schedule closely. Delays due to component availability were minimal and promptly managed. Table 7 summarizes team performance, highlighting both timely and delayed deliverables.

5 Conclusion

The project served as an effective learning tool for engineering project development, teaching the team important skills like risk analysis and schedule management using spreadsheets. It also helped improve teamwork, particularly in coordinating tasks and communication under the constraints of a shortened semester.

Regarding the project's core goal, the team developed NeoBell—a system combining a physical module with a secure delivery box. The system success-

Table 6: Project Components and Costs

Item	**Quantity**	**Price (R$)**
Camera USB with Microphone	2	86
Radxa ROCK 5C Lite 4GB	1	416
Speaker	1	15
Wood (m^3)	3	100
Servomotor	1	48
Magnetic Sensor	1	15
Amazon AWS	1	5
Push Button	1	10
Power Supply 12V 5A	1	20
Relay	3	5
SD 16GB	1	32
RFID Reader	1	12
Solenoid Lock	3	45
LEDs	5	5
Cables and Connectors (misc)	1	20
Margin	1	148
Emergency Budget		218
— Additional Items —		
Microphone	1	10
New servomotor 20 kg	1	109.20
EVA	1	15
Paint + Brush	1	158
Amplifier	1	55
Estimated Budget (with Margin + Emergency)		**1,200.00**
Over Budget		**+129.20**
Total Expenses		**1,329.20**

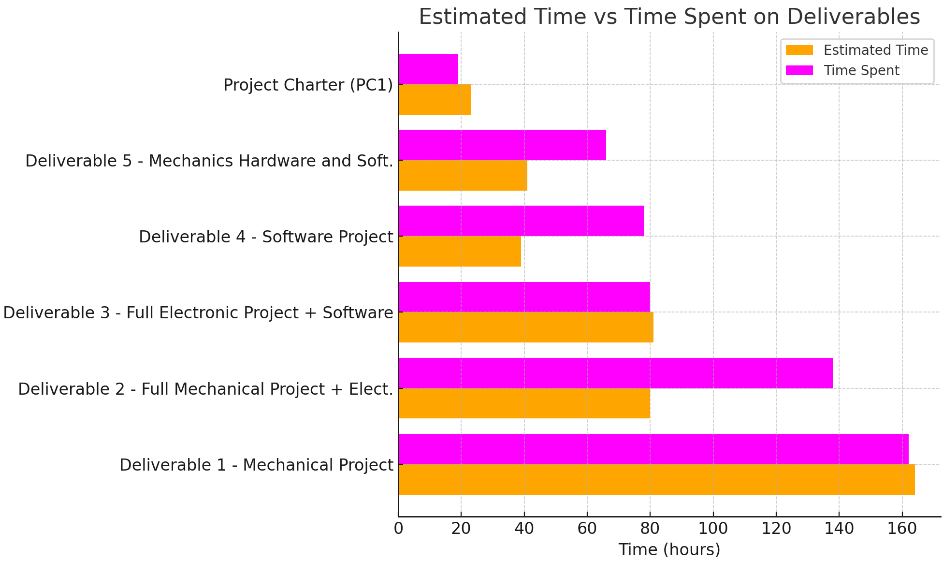


Table 7: Project Schedule

fully integrated hardware and software components, with all parts working together as intended, showing the project's overall success.

References

[1] Statista Research Department. E-commerce sales value in brazil from 2018 to 2023, 2024. Accessed: June 24, 2025.

[2] Times News Global. Challenges and opportunities in brazil's e-commerce market, 2024. Accessed: June 24, 2025.

[3] NeoBell project blog. `https://utopian-asterisk-ca0.notion.site/NeoBell-1deb3933ee48801ea5b1c806137cb59d`, 2025. Accessed: June 24, 2025.