POLITECNICO DI TORINO

Master's Degree in
Computer Engineering

Master's Degree Thesis

Studying and Redesigning a Web Application: User-Centered Analysis and Process Optimization

Supervisor
Prof. Stefano Quer

Candidate
Elia Ferraro

Co-supervisors
Dott. Diego Bruno
Ing. Gianpaolo Macario

Academic Year 2024 - 2025

Summary

In modern companies, resource management plays a critical role in the context of operational efficiency, security and cost control. The thesis focuses on the analysis, redesign, and optimization of the AROL S.p.A. IT Portal, a web-based software application useful for hardware and software tracking and management, used by the company's IT department. The project aims to improve usability, facilitate workflows, and support user experience (UX) through a user-centered approach.

The study started from the evaluation of the existing IT portal, highlighting usability issues, based on Jakob Nielsen's 10 Usability Heuristics. The analysis focused on the research of unnecessarily complicated navigation patterns, inefficient data presentation and inconsistencies in interaction design, leading to a poor user experience, increased cognitive load and a slower execution of daily tasks. These limitations significantly affect IT employees' operations that strongly depend on the IT Portal and that critically rely on it.

In order to address these problems, the project follows an iterative yet structured process, that begins with the definition of functional and non-functional requirements. This phase sets the starting point for the work to be performed during the next steps. This document was produced through numerous meetings and interviews with AROL managers and the IT department's users, crucial for understanding both the vision of the company and the detailed daily operations.

The redesign step focuses on the production of a new interface prototype in Figma, exploiting rapid prototyping methodologies to encourage early-stage feedback and iterative interface improvements. The new design concerns intuitive navigation, enhanced data organization and a responsive layout, ensuring a seamless and efficient user experience across different devices, adapting to diverse workflows. Particular attention was paid to hierarchical information architecture, consistency in UI elements, and interaction fluidity, all crucial aspects to optimize daily IT operations.

The implementation phase was carried out using modern web technologies, such as Angular for the frontend framework, combined with PrimeNG for UI components and ngx-scanner for QR code integration. The new system introduces a more structured dashboard, enhanced filtering system and an improved user interface that supports users' workflow. The application also adheres to better accessibility standards, ensuring improved usability for a diverse range of users.

To validate the success of the redesigned portal, many usability tests were conducted, involving IT professionals from AROL S.p.A. The evaluation considered task completion times, operation error rates, and user satisfaction metrics that demonstrated a significant improvement in these fields.

This research highlights the importance of human-centered design in enterprise software

development, providing a replicable approach to optimizing web applications in similar corporate environments. By combining UX principles, modern web technologies, and iterative prototyping, the project sets a reference for improving IT asset management systems through design-driven innovation.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

This section outlines the context of this thesis project, conducted in collaboration with AROL S.p.A. It emphasizes the company's industry relevance, the critical role of its IT department, and the challenges that underpin the project's objectives.

1.1.1 AROL S.p.A.

AROL S.p.A. is a leading company in capping solutions, based in Canelli (AT), Italy. It was born in 1978 with the mission of being recognized as a best-in-class expert in closing systems. Over the years, they have grown by expanding their facilities and opening many plants all over the world, from Europe to America, and also in Asia. Since 2017, AROL has acquired four other companies: LAMFI, MACA ENGINEERING, TIRELLI and UNIMAC-GHERRI forming the AROL group.[5]

Figure 1.1. AROL Quasar F - RF capping machinery for wine and liquors [6]

Today, they produce several machineries models, each developed for a specific type of cap, with the capacity to seal up to 100,000 bottles per hour. Thanks to a competitive and innovative R&D department they can offer a virtually limitless variety of capping solutions, including plastic bottle caps, alluminum caps, and corks for wine bottles. This is supported by data showing that AROL S.p.A. internally produces more than 95% of its mechanical parts, giving them full control over every production process and enabling the creation of new solutions with cutting-edge technologies and ideas.

In order to quantify the actual size of the company, information coming from AROL managmenet state that, as of 2024, the company has about 1000 employees that work from different countries, representing a range of cultures. As a result, finding a way to consistently manage all of them and keep track their work is a real and concrete challenge.

1.1.2 AROL IT department

Like many other companies, AROL has an IT department based at its headquarters, in Canelli, that includes about fifteen employees. Additionally, each facility worldwide has one dedicated IT staff member. In total, approximately twenty people work in IT area (Figure 1.2). Their job consists of managing user accounts, providing technical IT support to AROL employees and handling IT devices assigned to personnel.

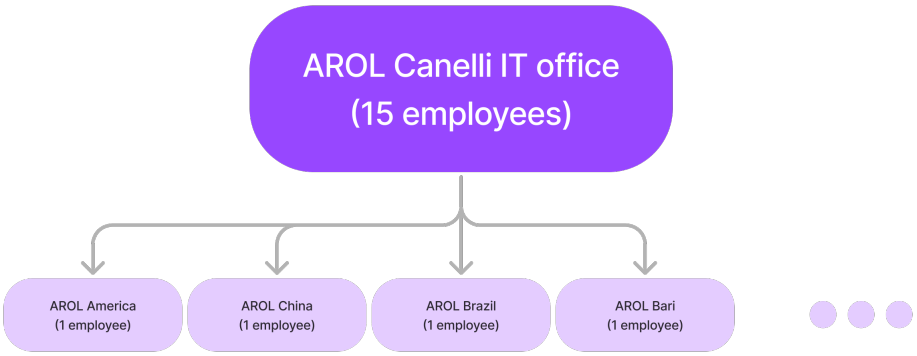

Figure 1.2. AROL IT department organizational graph

Even though AROL S.p.A. is not a software company, the presence of an IT department is crucial in order to be competitive in the market. In addition to the usual IT administration, where employees handle infrastructure and technical problem-solving, AROL's

IT department also develops software applications. This is essential for a company dealing with engineered machinery, as it enables the creation of software solutions tailored to business needs, improving the management of specific processes without the need to rely on an external supplier.

Moreover, the availability of reliable, real-time data supports managerial decisions, allowing the company to base strategic choices on concrete information rather than assumptions. Another key benefit is the potential for innovation in production processes, as IT systems provide the foundation for integrating modern technologies that guide efficiency and modernization. Additionally, human errors are significantly reduced, as automated processes minimize the risks associated with manual operations, leading to improved data accuracy. Finally, all these factors contribute to cost reduction, as resource optimization allows for better allocation of assets, reducing waste and unnecessary expenses.

IT employees use various softwares in order to carry out their duties. These tools must be designed to support their tasks and help them speed up the entire process, because this improvement benefits the production, resulting more profitable for the entire company. One of these software applications is the AROL IT Portal.

1.1.3 Project context

The thesis project focuses on the AROL IT Portal, a web application developed internally by the IT team. This portal serves as an inventory of all software, hardware, and IT resources within the company, allowing AROL IT employees to track and manage them efficiently. For example, daily tasks such as assigning a computer, returning a smartphone, or locating a specific peripheral can be performed using the IT Portal.

In the context of software engineering processes (see Figure 1.3), this project becomes part of the IT Portal evolution phase. Although it is the last step in the software production, it is often underestimated. In reality, it is actually the longest and the most challenging phase. It presents many pitfalls that limit programmer's flexibility. For instance, specific technologies must be used to ensure backward compatibility, and issues must be resolved without altering the existing user workflow to maintain a consistent user experience (UX).

The project is about the renovation of the IT Portal's frontend with the goal of fixing actual UI/UX problems and implement some new features. The portal's current version has many limits for what concern user interaction that will be largely analyzed in the next chapters. These UI problems affect IT tasks by slowing down operations and worsening IT users' productivity on their daily job. Technically simple interactions such as searching for information, updating an item, or assigning a computer are often cumbersome and time-consuming due to an old and inefficient interface, where many important design principles have been violated. In addition, few automations have been implemented, forcing users to perform operations manually. This issue increases the risk of errors and the mental overload of the users, pointing out a poor experience that results in dissatisfaction.

An important aspect will be the adoption of QR codes to streamline the IT resource

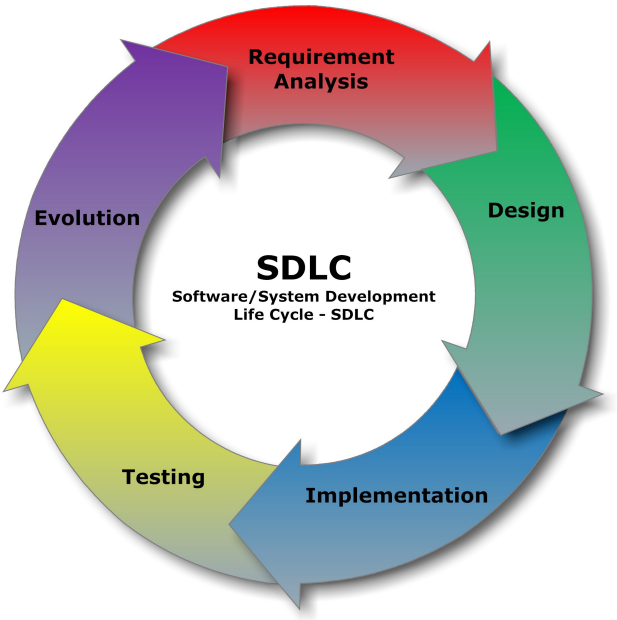

Figure 1.3. Software engineering process cycle [10]

management process, as they eliminate manual operations, which are prone to user errors, and facilitate information retrieval. The foundation for this implementation has been laid by the thesis work of two other AROL students, Chessa Alessio [9] and Santoro Francesco [28], who specifically focused on QR code management. Their work will be valuable for this thesis project, as many challenges have already been addressed, and their results can be leveraged during the implementation phase.

1.2 Objectives

The objectives of the thesis project can be divided into two main categories: business and study.

1.2.1 Business goals

The first category corresponds to practical objectives that must be achieved by the end of the project, which have an actual and significant effect on the company's workflow. These include the design and the development of a new improved version of the AROL IT portal's frontend, fixing its critical issues, which currently hinder IT users from having a seamless and supportive experience with the application. These goals will be reached by enhancing the overall usability and reorganizing distribution of information across pages in the portal, providing users with a clearer view of the system and making it easier to understand and use.

These objectives will be assessed through formal usability tests, collecting data such as task completion times and user feedback. Success will be determined if the general impressions from user are positive and if the recorded times for performing complex, time-consuming tasks show that the new version is significantly faster than the current one.

1.2.2 Academic goals

The second category focuses on academic objectives, aimed at exploring and applying innovative approaches to improve the project's outcomes. First, the thesis will formally analyze the current version of the portal, identifying key design flaws based on the *10 Usability Heuristics* from Jakob Nielsen [23]. These findings will serve as the foundation for creating a new design that addresses and resolves these issues.

The new design will be developed using modern methodologies, including rapid prototyping, to provide a clear and interactive visualization of the proposed solutions before implementation. This approach will not only streamline the design process but also facilitate early feedback and iterative improvements.

1.3 Thesis Overview

The thesis will be structured into eight chapters, with this being the first. Each chapter covers a specific phase of the project, presented in chronological order. This structure allows readers to follow the thesis in the logical sequence of events and deadlines, providing a clearer understanding of the problems encountered and how they were solved based on the available information and the state of the project at the time.

Following this introductory chapter, the state of the art will be presented. This section provides an overview of the previous version of the portal's interface, establishing a foundation for future considerations. The analysis covers various aspects, including the functionalities currently available and the way data is presented. Finally, a formal usability evaluation will be conducted to assess existing issues.

Next, the requirements analysis chapter will focus on the process of drafting the requirements document, highlighting the methodologies used and the challenges encountered. This phase was essential in defining the goals and final objectives of the thesis.

The fourth chapter examines the process that led to the creation of the low/medium-fidelity prototype of the new frontend. In this section, design choices will be explained, aiming to extract the best possible design that meets user needs.

After this initial preparation phase, the actual implementation will be described in the fifth chapter. It begins by outlining the tools used for development, from the code editor to specific libraries, along with the reasons behind these choices. The main challenges of this phase will then be discussed, detailing the solutions adopted. At the end of the chapter, an overview of the final proposed product will be presented, including screenshots from its execution.

The sixth chapter focuses on the verification and validation of the final frontend. It begins with an analysis of the previously defined requirements, explaining how they were assessed. The validation phase will then be conducted by collecting metrics and insights from predefined tasks submitted to users through usability tests. The final evaluation of the new frontend will provide a score that quantifies the improvements achieved with the new solution.

The last two chapters are designed to summarize the entire process, identifying what went well and which areas require further improvement, establishing a starting point for future work.

Chapter 2

State of the Art

This chapter focuses on analyzing the current version of the AROL IT portal's frontend, outlining its main functionalities and formally describing its design issues through the application of Jakob Nielsen's 10 Usability Heuristics. Subsequently, a brief overview of similar solutions addressing the same problem will be presented, with a comparison to AROL's current implementation.

The analysis of the current portal was conducted by examining its main pages and attempting to understand their functionalities without any guidance or support. This approach effectively simulates the experience of a novice IT user interacting with the application for the first time. It is important to note that a well-designed user interface should cater to the needs of both experienced users and novices alike. This implies that the former should be able to expedite their tasks by utilizing shortcuts and hidden features, which might not be immediately apparent to a new user. Meanwhile, the latter should find it intuitive to understand and navigate the interface to achieve their goals.

2.1 IT Portal Analysis

The portal was developed to address a specific need of the IT department: tracking the hardware and software assets of the entire company. This necessity originated from the fact that many AROL employees require devices such as laptops or smartphones, while various departments rely on servers, monitors, and other equipment. As a result, the company often ends up with numerous hardware devices scattered across different departments and facilities, including locations in foreign countries. To quantify the usefulness of the IT Portal, data from the system indicate that the AROL IT department is currently managing more than five thousand devices, including computers, smartphones, and peripherals. In addition to hardware, employees require software licenses to access and utilize third-party applications, such as the Office Suite.

To effectively manage this complexity and ensure real-time knowledge of which employee is assigned to which device or where a particular device is located, the creation of a centralized hub became essential. This hub, used by IT personnel, records every step in the lifecycle of each user-device relationship.

2.1.1 Portal functionalities

Since the AROL IT portal primarily serves as a tracking and inventory tool, its predominant functionality concerns CRUD (Create, Read, Update, Delete) operations, applied to every entity within the application. IT users can create the necessary entities, such as a newly purchased laptop, to add them to the system and begin configuring them assigning the device to an employee, for instance. Once an entity is created, users can access its dedicated page, which contains detailed information specific to the selected item. This allows them to make decisions based on reliable and unambiguous data. At any point, these details can be updated, such as when an IT user modifies a computer's configuration, switches its operating system, or installs a different CPU model. Finally, entities can either be deleted or marked as disabled when they are no longer required or when their lifecycle has ended for any reason.

The following Table 2.1 provides a formal list of the functionalities of the current version of the AROL IT portal.

FN	Functionality Name	Description
F1	**View, filter, and sort users**	View user details after sorting and filtering rows. User data is managed by an external platform, so creation, modification, and deletion are not possible in the IT portal. This data is imported from an external system.
F2	**View, filter, and sort Active Directory groups**	View and filter Active Directory (AD) groups assigned to employees. [1]
F3	**View, filter, and sort computers**	View, sort, and filter the list of computers in a tabular view. Access detailed information for each computer.
F4	**View, filter, and sort user-computer associations**	View, sort, and filter data regarding associations between users and computers.
F5	**Manage MAC and IP addresses for computers**	Perform CRUD operations (Create, Read, Update, Delete) for MAC and IP addresses assigned to computers.

Continues on the next page...

[1] AD is a Microsoft system for centralized management of permissions and policies.

FN	Functionality Name	Description
F6	**View installed software**	View a list of software detected on each computer via automatic scans. Sorting and filtering options are available.
F7	**Manage installed components**	Add, edit, remove, and view PC components such as CPU, RAM, HDD, and GPU for each computer.
F8	**Manage computer peripherals**	View, filter, sort, attach, and detach peripherals like monitors, printers, or external USB HDDs for computers.
F9	**Manage all computer devices**	View, filter, and sort all computer devices available in the system. Add new devices as needed.
F10	**Manage computer device types**	Perform CRUD operations on the list of computer device types, with sorting and filtering options.
F11	**Manage MAC and IP addresses for computer devices**	Perform CRUD operations for MAC and IP addresses associated with computer devices.
F12	**Manage computer statuses**	View, filter, and sort all possible statuses for computers and devices. New statuses can be added or existing ones edited or removed.
F13	**Manage computer types**	View, filter, and sort the list of computer types. Add, edit, or remove entries as needed.
F14	**Manage software**	View, filter, and sort the list of registered software. Add, edit, or delete software entries.
F15	**Manage software licenses**	View, filter, and sort a list of software licenses owned by AROL. Assign licenses to employees or update license details.
F16	**View computers to be renewed**	View, filter, and sort computers due for renewal. Typically, computers older than six years are identified for upgrades or disposal.
F17	**Manage purchase modes**	View, filter, and sort purchase modes such as *Lease*, *Purchase*, and *Loan*. Add, edit, or delete entries.
F18	**Manage mobile device types**	View, filter, and sort the list of mobile device types. Add, edit, or delete entries.

Continues on the next page...

FN	Functionality Name	Description
F19	**Manage mobile devices**	View, filter, and sort the list of mobile devices. Add, edit, or delete entries as needed.
F20	**Manage mobile device assignments**	View, filter, and sort the list of user-mobile device assignments. Assign new devices to users or revoke existing ones.
F21	**Manage SIM cards and assignments**	View, filter, and sort the list of SIM cards. Assign SIM cards to devices or update their details. Add or remove entries as needed.
F22	**Manage mobile operators**	View, filter, and sort the list of supported mobile operators. Add, edit, or delete entries.
F23	**Manage tariff plans**	View, filter, and sort the list of tariff plans available for each mobile operator. Add, edit, or delete entries.
F24	**Manage tariff options**	View, filter, and sort the list of tariff options available for each mobile operator. Add, edit, or delete entries.
F25	**Manage SIM formats**	View, filter, and sort the list of SIM formats. Add, edit, or delete entries.
F26	**Manage mobile device statuses**	View, filter, and sort the list of mobile device statuses. Add, edit, or delete entries as needed.
F27	**Manage manufacturers**	View, filter, and sort the list of manufacturers that produce devices acquired by AROL. Add, edit, or delete entries.
F28	**Manage device models**	View, filter, and sort the list of product models for each manufacturer. Add, edit, or delete entries.
F29	**Manage suppliers**	View, filter, and sort the list of suppliers that provide products to AROL. Add, edit, or delete entries.
F30	**View user assignments**	View all devices assigned to a specific user by entering their name. Select entries to view detailed information about each device.
F31	**Return user devices**	Select devices assigned to a user and mark them for return when no longer needed.

Continues on the next page...

FN	Functionality Name	Description
F32	**Manage user discharge records**	View, filter, and sort the list of discharge records for users. Add, edit, or delete entries.
F33	**Manage user smart working requests**	View, filter, and sort the list of smart working requests submitted by employees. Add, edit, or delete entries.
F34	**Manage web portals**	View, filter, and sort the list of web portals used by AROL. Add, edit, or delete entries.
F35	**Manage applications**	View, filter, and sort the list of applications used by AROL. Add, edit, or delete entries.
F36	**Manage portal settings**	Configure portal settings such as automatic email notifications or the duration of insurance policies.
F37	**Manage user habilitation requests**	View, filter, and sort user habilitation requests. Entries can be categorized by status, such as *Approved* or *Rejected*.

Table 2.1: Functionalities of the AROL IT portal's current version.

2.1.2 Portal data structure

Within the AROL system, the main entities involved in the current functionalities are as follows:

- Users
- Computers
- Mobile devices
- Computer devices
- SIM cards
- Softwares
- Licences

In addition to these core entities, which are the most frequently considered and critical to track, the system includes supplementary entities that serve as supporting information. For instance, catalogues for suppliers, manufacturers, and computer models are maintained to provide a list of available values. However, these supplementary entities are not accessed on a daily basis.

Figure 2.1 illustrates an entity diagram representing the current state of the application. It is important to note that this is not a database schema; rather, it has been created solely based on observations of the frontend, analyzing the page structure and the arrangement of information. Consequently, it is not a comprehensive representation of every minor detail within the portal. Instead, it serves as a general overview to outline the portal's content.

The focal point of the diagram is the *User* entity, representing the employees of all the companies within the AROL GROUP, tracked in the IT portal. For each user, standard personal information is recorded, including identity details, the job location and organizational position, defined by the *Company name*, *Department* and *Office*. Each user is assigned to a set of *Active Directory groups* that implement an authorization system, outlining the actions employees are permitted to perform and the resources they can access.

Since the primary requirement of the application is the hardware tracking, the diagram contains three crucial elements of the system: *Computers*, *Mobile devices* and *Computer devices*. These correspond to physical devices provided by AROL, managed through its IT department. Each item in the these categories is univokely identified by an *hardware code* (HW code) which is an identification system set up by AROL in order to distinguish its hardware assets. These entities also contain essential information such as the *manufacturer*, the *model* and the *type*.

The diagram also presents the *SIM card* entity, linked to mobile devices to record associated phone numbers and tariff plans. Lastly, *Licences* are assigned to Users, enabling them to access and use specific third-party software products within a defined validity period. Additionally, *Software* is linked to Computer entity, as the system provides a dedicated page listing all detected software installed on each computer.

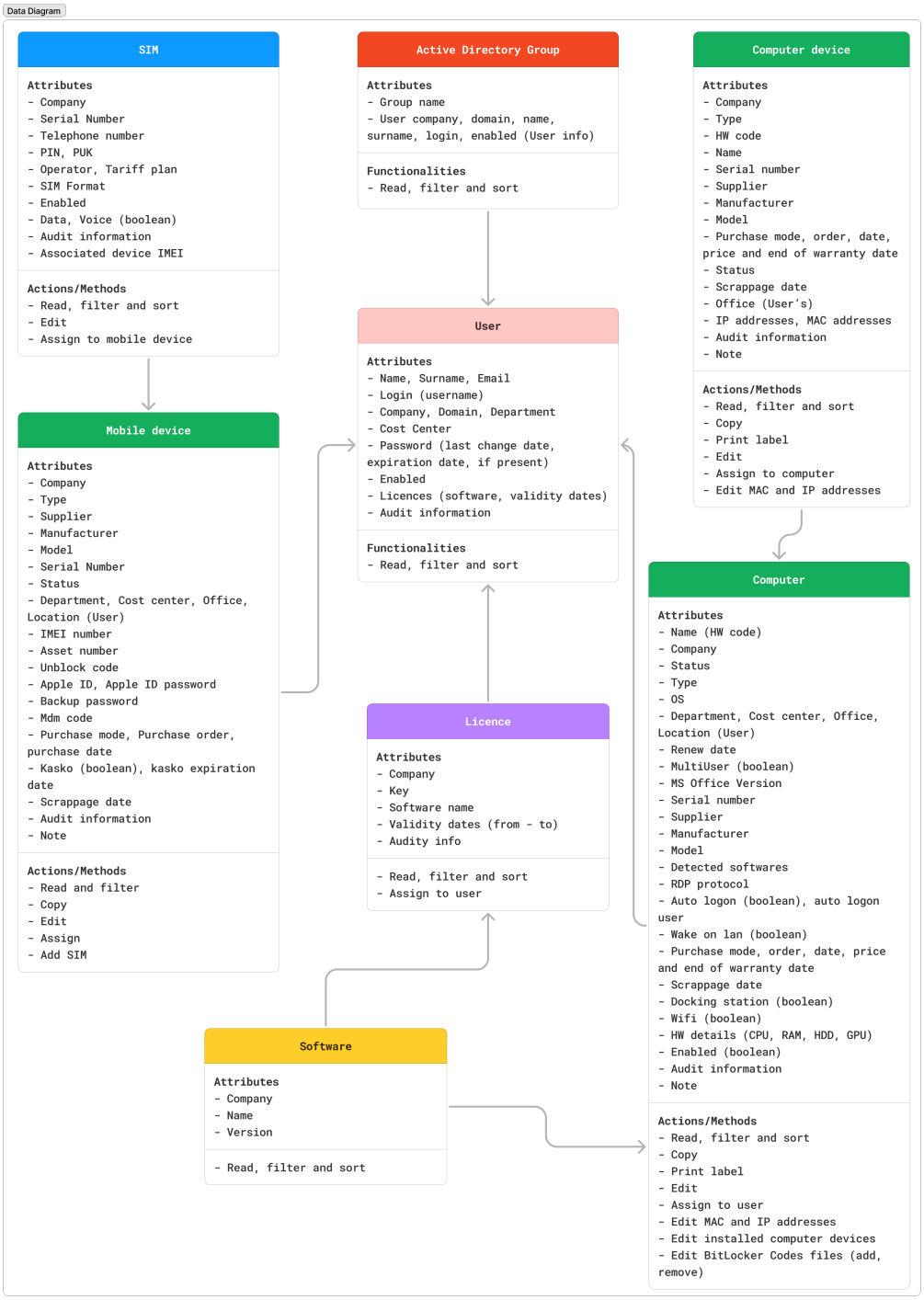

Figure 2.1. Entity diagram (AS-IS)

2.1.3 Portal design and its limitations

This section analyzes the main pages of the AROL IT portal by examining each one, highlighting its components, and identifying critical design issues that prevent users from having a smooth experience. Before starting, it is essential to select a formal method to identify and classify issues by severity, ensuring a systematic approach to determine which parts of the portal require the most urgent and impacting interventions.

Many solutions can be adopted for this kind of analysis. They include evaluating a system by simulating a user's problem-solving process while completing tasks or by implementing surveys for actual portal users to assess their satisfaction. However, this kind of approach are often time consuming, both for the analyst and the users. Also they often reveal that some problem in the design exists, but they don't expose the problem itself. Additionally, other expert heuristic frameworks could have been used, such as those by Ben Shneiderman or Bruce Tognazzini. The problem with these alternatives is that they focus only on specific areas or lack a comprehensive view of the entire human-computer interaction. After careful consideration, the *10 Nielsen's Usability Heuristics* were selected as the primary evaluation framework.

The *10 Nielsen's Usability Heuristics* are a set of user interface design principles created by Jakob Nielsen in 1990. Nielsen, a Danish computer scientist born in 1957, specialized in Human-Computer Interaction (HCI) is largely considered a pioneer in usability and user-centered design. He has authored numerous influential books on web usability, including *Designing Web Usability: The Practice of Simplicity*, which sold more than a quarter of a million copies and has been translated into 22 languages. He also founded, in collaboration with Donald Arthur Norman, the Nielsen Norman Group: a computer UI and UX consulting firm. Due to his experience and contributions to the field, his heuristics are trusted and widely adopted as a framework for usability analysis. The heuristics are concise and complete principles that cover key aspects of usability and interaction between humans and machines. They have been produced for general purposes, so they are not bounded to a specific context or a particular framework.

Figure 2.2. Jakob Nielsen (left) during a public interview [16]

Nielsen's Heuristics were chosen for the analysis of the IT portal because they offer

several important advantages. First of all, they are practical, as their application does not require complex setups or specialized tools, making them ideal for quickly evaluating the portal's usability. Additionally, they are comprehensive, covering a wide range of usability aspects and ensuring that no critical issue is overlooked. Finally, they are highly relevant to this specific context, since they were developed with web design in mind, making them perfectly suited for assessing the usability of the IT portal.

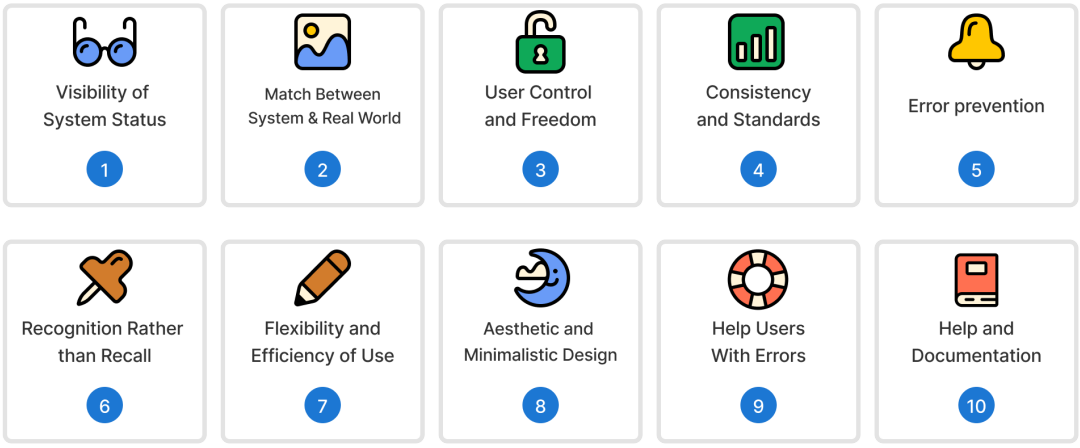

Figure 2.3. Jakob Nielsen's 10 usability heuristics

The following is the list of the ten heuristics, with concise explanations of their key ideas and illustrative examples.

1. **Visibility of System Status**
   The design should always keep users informed about what is going on, through appropriate feedback within a reasonable amount of time.

   Example: A file download progress bar that shows the percentage completed and the estimated time remaining provides clear feedback to users about the current status of their task.

2. **Match Between the System and the Real World**
   The design should speak the users' language. Use words, phrases, and concepts familiar to the user, rather than internal jargon. Follow real-world conventions, making information appear in a natural and logical order.

   Example: An e-commerce website should display product categories like "Clothing" and "Electronics" rather than company internal codes such as "C001" and "C002".

3. **User Control and Freedom**
   Users often perform actions by mistake. They need a clearly marked "emergency exit" to leave the unwanted action without having to go through an extended process.

Example: A "Cancel" button on a form or a "Back" button in a web browser allows users to undo or leave an action easily.

4. **Consistency and Standards**
   Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform and industry conventions.

   Example: Using a trash bin icon for the delete function or a pen icon for edit function, as is standard across many systems, ensures users immediately understand its purpose.

5. **Error Prevention**
   Good error messages are important, but the best designs carefully prevent problems from occurring in the first place. Either eliminate error-prone conditions, or check for them and present users with a confirmation option before they commit to the action.

   Example: A form validation feature that highlights missing or invalid fields before submission prevents errors such as incomplete entries.

6. **Recognition Rather than Recall**
   Minimize the user's memory load by making elements, actions, and options visible. The user should not have to remember information from one part of the interface to another. Information required to use the design (e.g. field labels or menu items) should be visible or easily retrievable when needed.

   Example: A dropdown menu listing all available options allows users to select from visible choices rather than typing from memory.

7. **Flexibility and Efficiency of Use**
   Shortcuts, hidden from novice users, may speed up the interaction for the expert user so that the design can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.

   Example: Keyboard shortcuts like Ctrl+C for "Copy" and Ctrl+V for "Paste" enable experienced users to work faster while still allowing novice users to use the menu options.

8. **Aesthetic and Minimalist Design**
   Interfaces should not contain information that is irrelevant or rarely needed. Every extra unit of information in an interface competes with the relevant units of information and diminishes their relative visibility.

   Example: A mobile banking app displays only essential actions like "Transfer Money" and "View Balance" on the home screen, with less-used features accessible through a secondary menu.

9. **Help Users Recognize, Diagnose, and Recover from Errors**
   Error messages should be expressed in plain language (no error codes), precisely indicate the problem, and constructively suggest a solution.

Example: An error message like "Your password must contain at least 8 characters, including a number and a symbol" is clear and actionable, compared to "Error 400: Invalid Input."

10. **Help and Documentation**
    It's best if the system doesn't need any additional explanation. However, it may be necessary to provide documentation to help users understand how to complete their tasks.

    Example: A search function within a help page that allows users to quickly find "How to reset my password" is more user-friendly than a long, unorganized manual.

Once the main heuristic violations are identified and analyzed, they will be classified into three categories based on their degree of severity. Below is the list of the adopted categories along with a brief description defining how violations are associated with each:

1. *Low* - 1 point
   Violations that have minimal impact on the overall user experience. These issues may cause slight inconvenience or aesthetic inconsistencies but do not significantly hinder the user's ability to interact with the system effectively.

   Example: Using a less familiar but still understandable label for a button, such as "Go Back" instead of "Back.

2. *Medium* - 2 points
   Violations that moderately affect usability. These issues may slow down users, create confusion, or require additional effort to complete tasks, but they do not entirely prevent task completion.

   Example: A dropdown menu with options that are not intuitively ordered, making it harder to find the desired selection.

3. *High* - 3 points
   Violations that severely impact usability. These issues can result in critical user frustration, errors, or the inability to complete essential tasks, making immediate attention and resolution necessary.

   Example: A crucial error message that is either unclear or not displayed, leaving users unaware of how to resolve the problem.

This classification allows for a extensive evaluation of the portal's usability by quantifying each violation's severity and prevalence, contributing to an overall score that reflects the system's design quality. It also helps identify critical areas where the most problematic components or functionalities significantly impact the user experience. By distinguishing between issues of varying severity, this classification enables the prioritization of improvements, ensuring that the most critical problems are addressed first. Furthermore, this classification serves as a reference for future evaluations, providing a way to measure progress and compare the current state of the portal with subsequent improvements.

After reviewing the modalities adopted for the design analysis of the current version of the AROL IT portal, the following section provides a detailed description of the application's design and its main pages. Throughout this presentation, some references to the heuristics will be highlighted by identifying instances where the principles are violated.

Portal pages structure

Before digging into the content of the application iself, it is essential to provide a brief overview of its general structure. This is a fundamental aspect of system design, as it clarifies how users perceive the state of the software. In fact, even minor modifications to the application's structure, even without altering the content of the pages, can result in significant changes to the user workflow and interactions with the system.

Information Architecture (IA) patterns represent the structural models of software applications. A wide variety of patterns, often quite different from one another, are currently used across a wide range of contexts and purposes. Each pattern has distinct features tailored to specific applications, as they promote particular types of interactions. While all these patterns are valid, widely trusted, and supported by leading designers worldwide, selecting the most appropriate one for a given application requires careful and informed consideration. Below are some examples of Information Architecture patterns:

1. *Hub and Spoke*
   This interface features an index page (the "hub") with links to other sections (the "spokes"). Users must return to the hub to move between spokes, promoting a focus on one task at a time. It is often used in multi-purpose applications with diverse functionalities.

2. *Nested Doll*
   A linear pattern where users navigate from an index page with general information to pages with more detailed content. Its clarity and simplicity reduce the likelihood of users getting lost. This pattern is commonly seen in e-commerce websites.

3. *Tabbed View*
   A familiar pattern resembling how browser pages are organized. Users can switch between sections using a toolbar. However, it works best with a small and simple number of sections, as too many tabs can overcrowd the interface.

4. *Dashboard*
   This pattern centers around a homepage displaying multiple elements summarizing related content. Its main advantage is providing an overview at a glance without requiring users to navigate multiple pages. It is widely used in content-heavy applications such as management and financial systems.

5. *Hierarchy*
   This widely used pattern is common across many websites. It is organized with an index page containing links of subpages, which in turn include links to sub-subpages, in that way forming a hierarchical tree of pages.

It is important to note that an application can adopt, and often does, more than one pattern, combining them into a cohesive and complex solution. For this reason, the current version of the AROL IT portal implements two of the patterns above: *Hierarchy* and *Tabbed View*. The first helps users with the organization and compartmentalization of information, by following a logical path from general to specific content, making it easier to locate information if they are familiar with the overall structure. Each level in the hierarchy represents a more detailed subset of information, reflecting a "top-down" approach that guides users step by step toward their desired content. The hierarchy schema of the IT portal's pages is represented in Figure 2.4. The second one, instead, allows them to freely move across many related content without actually leaving the application section in which they are.

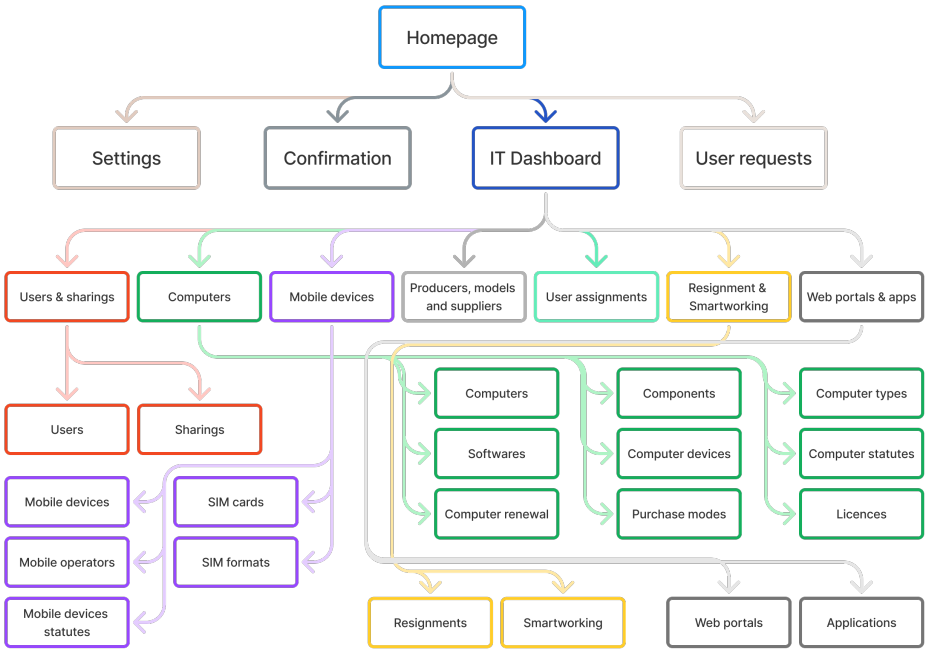

Figure 2.4. AROL IT portal Pages Schema (current version)

The main pages of the AROL IT portal are now presented, highlighting the most important features and especially the violations of the ten usability principles.

Homepage

The first page analyzed during the exploration was the homepage. Typically, this section should provide an overview of the entire application, helping users understand the type of information they can access and the operations they can perform.

Figure 2.5. Homepage, AROL IT portal, current version

As shown in Figure 2.5, the homepage of the IT portal has a simple structure. It consists of a large main container hosting four cards, each corresponding to the following areas:

- Portal settings (*Impostazioni*)
- Material confirmation (*Conferma materiale*)
- IT Dashboard
- User enablings (*Abilitazioni utente*)

The first noticeable aspect is the language used in the portal. The current version of the AROL IT portal adopts Italian. This choice is questionable because, although the majority of portal users are Italian speakers working in the IT office in Canelli, Italy, some employees responsible for international plants speak other languages native to their respective countries (e.g., Brazil, China, the USA, and others). Since software should be inclusive and accessible across borders, adopting English as the system's primary language would have been a more appropriate choice. Even better, the most inclusive solution would allow users to independently select their preferred language, improving usability for those who are not fluent in English.

Another relevant issue is the uniformity of the four cards. They share the same structure, with a bold primary blue color that is highly visible and contrasting, accompanied by a brief description of the respective section. This uniformity conveys the impression that all four sections are equally significant and relevant within the application. As a result, users may expect that the areas accessed through these cards offer similar computational capabilities and usefulness, which may not accurately reflect their actual functionality or importance. This observation was later confirmed during a meeting with IT employees, who indicated that they spend most of their time working within the IT Dashboard section.

If a new homepage design were to be developed, a more effective approach could involve creating distinct cards or, even better, differentiated components with varying hierarchy levels to emphasize the most frequently used section. For instance, the *IT Dashboard* area should be more visible than other components, as it represents the primary action performed by most users from the homepage. This card should be the first element to capture users' attention when launching the application, leading them through the intended workflow path.

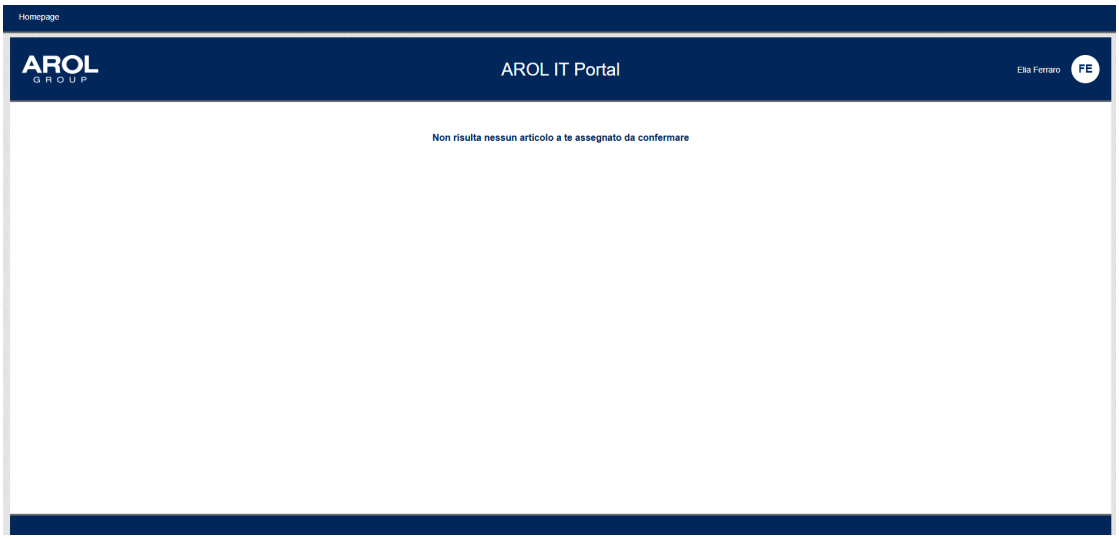

Figure 2.6. Material confirmation page with empty content, current version

Secondly, the *Material Confirmation* card redirects to another page where some actions can be performed, but only if the portal user possesses devices or materials that require receipt confirmation. In all other cases, it displays an almost blank page, as illustrated in Figure 2.6. This is not a real, impacting, problem, but it can be addressed to improve user efficiency by removing distracting components or buttons and displaying them only when necessary. This issue can be classified as a violation of design principle number seven: *Flexibility and Efficiency of use*, with a low grade of severity.

Now, focus on the top navigation bar. Usually, a standard navigation bar (also called *navbar* by designers and developers) in web design is a component located at the top of the screen, designed to support navigation across system's pages. It typically follows a standard structure: a row of elements that correspond to the main sections of the website. users generally expect to see the company or system logo, followed by a horizontal list of links for navigating to other pages, and, on the right, functional buttons such as the user's profile or a language switch. Although this design is not mandatory and many other versions are widely used in trusted and well-structured applications, adhering to standard is often a wise choice, for the reasons mentioned earlier regarding users' familiarity with commonly used components. Famous standard navigation bars examples can be found in Figure 2.7.

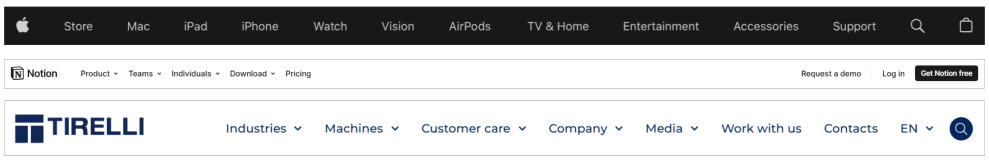

Figure 2.7. Examples of navigation bars (Apple, Notion, Tirelli)

In the IT portal, the logo is displayed below the navbar, along with the application title and the profile button. At the top of the screen, the links section contains only the "Homepage" link, which redirects users to the corresponding page. A problem arises when users click on the AROL logo in the blue bar just below, as this action also redirects them to the homepage. This is a design mistake because, in software design, there should not be multiple elements or components with different appearances performing the same action. Such inconsistencies confuse users, making it harder for them to capture differences between these elements, if any, and increasing the application's learning curve.

Differences in appearance for elements with the same functionality can be classified as a low-severity violation of Nielsen's heuristic number four: *Consistency and Standards*. This violation occurs because the designers interrupted the uniformity of elements and failed to maintain consistency across components serving the same function.

IT Dashboard

By clicking in the *IT Dashboard* card, users are redirected to the corresponding page. As Figure 2.8 shows, the appearance of the interface is very similar to the previous one, keeping a consistent design across multiple pages. Consistency is fundamental in well-designed interfaces, as users, being human beings, are naturally incline to repetition and uniformity.

Familiarity with a software product can be achieved more quickly and effortlessly when the design supports this process by reusing components, adopting established standards already encountered in other applications, and avoiding subtle, confusing changes.

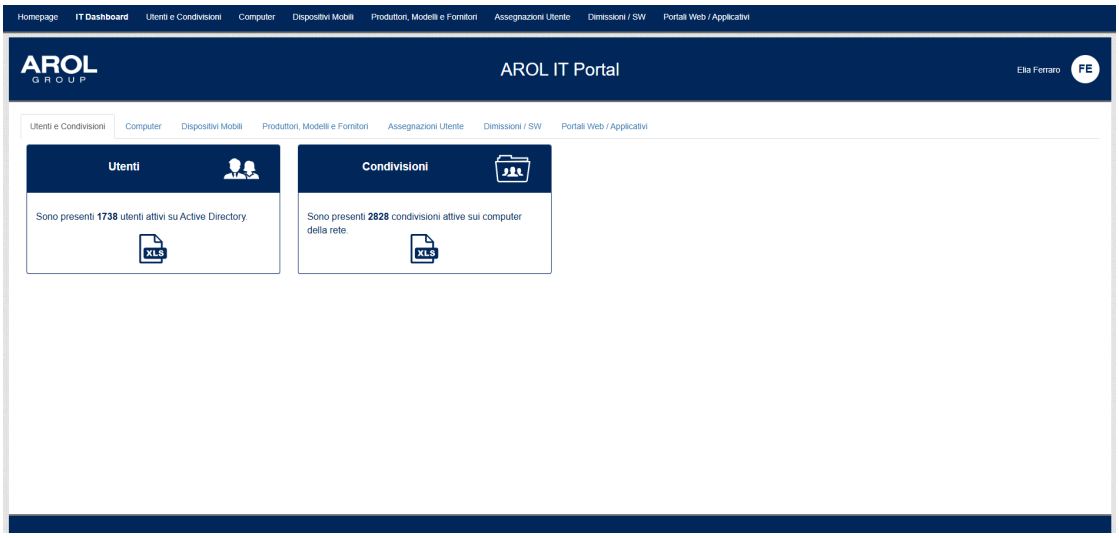

Figure 2.8. IT Dashboard page, current version

At the top of the page, the horizontal, blue, navigation bar has changed with respect to the homepage bar.

The area that, on the homepage, previously only contained the "Homepage" button now houses additional links, such as the "IT Dashboard" link, which refers to the current page, along with seven buttons corresponding to the seven sections in the IT Dashboard area. The issue about it is how these links are displayed. They have the same look: white, small text that turns bold when selected. However, they are part of a page hierarchy (Figure 2.4), that highlights the homepage as the root node, followed by the IT Dashboard page (the current one) and then all the seven sections. This means that they are contained one into the other and this feature must be conveyed by the graphical representation of the list of pages, they cannot be at the same level. This issue can be classified as a medium-severity violation of design principle number 1: *Visibility of System Status* because users cannot clearly understand from it how pages are linked between them and, as consequence, how to reach them.

The same Nielsen design principle is violated, but with a lesser degree of severity (Low), by the navigation links in the navbar when they are in their active state. When users click on a link and are redirected to the corresponding page, the link turns bold. While this is acceptable, it requires users to specifically pay attention to it. The bolding is not very noticeable, and users may need to scan the entire navigation bar to identify slight differences in the appearance of the links in order to determine where they are.

Although the navigation bar and its links have already been discussed, an additional piece of information needs to be addressed. This concerns the duplication of the navigation links from the top blue navbar in the form of tabs within the page content area. This design choice is unusual and problematic for navigation. As previously mentioned, *Hierarchy* and *Tabbed View* are two main Information Architecture design patterns, but they should not be applied simultaneously to represent the same set of data. This approach sends users mixed ideas, leading to confusion. They may wonder: 'Are these pages nested hierarchically within the main one?' or 'Are these sections part of the same page, with content changing according to the selected tab?'. This ambiguity disturbs the user experience, leaving them uncertain about how to navigate the system and preventing them from becoming familiar with the software.

Furthermore, while navigating the application using the navbar links works as intended and provides a smooth navigation experience, a critical issue arises when users interact with the tabs located just below the navbar. Specifically, the active state of the navigation link in the navbar does not align with the selected tab! This discrepancy creates confusion, as the two elements indicate different current locations within the application. For instance, as illustrated in Figure 2.9, the navigation bar suggests that the user is still on the *IT Dashboard* page, whereas the active tab corresponds to the *Computers* page. This inconsistency, which compromises the clarity and reliability of the navigation system, can be classified as a violation of Nielsen's design principle number four. In particular, this problem is quite severe, falling between the medium and high degrees of severity. Since the issue does not completely prevent users from performing any actions, it is categorized as a medium-severity violation.

Figure 2.9. Navbar and tabs section's active states mismatch in IT portal current version

Multiple IA design patterns can coexist within an application only if they are well-designed and seamlessly integrated, with each pattern addressing a specific and distinct set of information.

Now, move the attention to any of the cards of the *IT Dashboard* page. They all share the same structure, featuring a blue top band containing the card's title and an icon representing the content of the corresponding page. Below this, the card's body displays some outlined information, and at the bottom, there is a blue, clickable document icon labeled 'XLS'. Since the acronym stands for (Microsoft) Excel Spreadsheet, users naturally expect it to redirect them to a page displaying tabular data. However, clicking on it downloads an Excel file directly to the IT user's machine.

While this is not a significant issue, since it neither prevents the application's usability nor slows down operations, users will adapt to it after a few interactions. Nevertheless, a more effective approach would involve replacing the current icon with one that includes a downward arrow. This is a standard symbol universally recognized as indicating a download action, reducing confusion and preventing unintentional downloads, especially for users with limited bandwidth.

This kind of violations affect Nielsen's design principle number four: *Consistency and Standards*, with a low degree of severity.

A slightly more significant issue emerges with the 'XLS' icons due to their placement within another clickable element: the cards themselves. This design choice leads to a problematic interaction. When users hover over either the card or the icon, the cursor changes to the pointer style, visually signaling a clickable element. However, there is no visual distinction between hovering over the card as a whole and hovering specifically over the icon. As a result, users might assume that the icon is not an independently clickable element but only a part of the card's overall click functionality, similar to the other static elements on the card.

This lack of clarity can cause confusion and potentially disrupt the user experience, especially for those unfamiliar with the interface. To address this, designers should implement a more intuitive hover effect to clearly indicate the icon's clickable nature. For example, the icon could change color, display a subtle highlight, or feature a tooltip with a descriptive label like 'Download Excel File' when hovered over. These enhancements would provide a clear visual signal, improving both usability and accessibility. By doing so, users would immediately understand the icon's independent functionality without discovering it by chance.

Users

After the click on the *Users* card of the *IT Dashboard* section, IT users are introduced to the corresponding page.

As Figure 2.10 shows, the content of this page is categorized with two tabs: *Users* and *AD Groups*. Both areas, as well as all the other pages that display a large amount of data, adopt a tabular view because this is the best way to group a lot of information in small spaces.

At both the top and bottom of the page, standard pagination elements are implemented to manage the display of user rows on the screen. However, the current design could benefit from a more simplified and minimalist approach, particularly when dealing with a large number of rows. The current paginator[2] displays all available page numbers from the first

[2]A paginator is a tool used in web development to divide large sets of data into smaller, more manageable pages, improving performance and usability. It enables users to navigate through data in chunks rather than loading everything at once, which can slow down page performance and overwhelm the interface. Users can typically move between pages using numbers, arrows, or navigation controls (e.g., "Next" or "Previous").

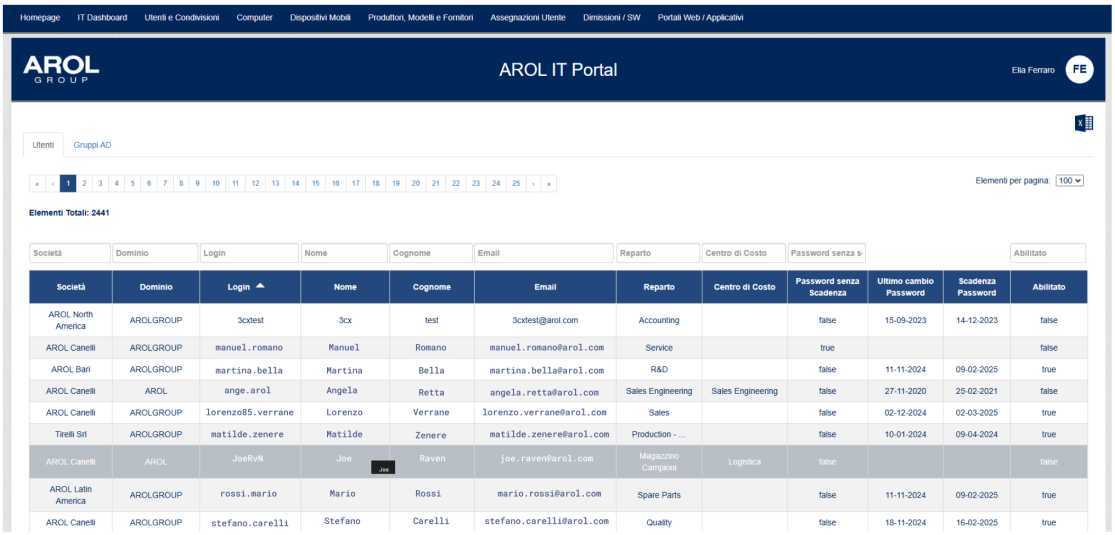

Figure 2.10. Users page, AROL IT portal, current version

to the last, regardless of the total number. This design can overcrowd the interface, as shown in Figure 2.11, taken from the *AD Groups* section.

A more efficient solution would be to adopt a compact pagination design. For instance, a 'condensed' paginator, similar to the one used by Google in its search results, could be employed. Such a paginator typically shows only the current page, a few surrounding pages, and ellipses to indicate additional pages. This design not only reduces visual confusion, but also helps users focus on their current position within the dataset while maintaining easy access to other pages.

Implementing this improvement would enhance usability and provide a cleaner, more professional appearance. It would also align with Nielsen's design principle number eight (*Aesthetic and Minimalist design*), ensuring that users can navigate large datasets without feeling overwhelmed by an unnecessarily complex interface. This is a very low severity degree violation, but it would support a smooth user experience reducing the visual overload of pages.

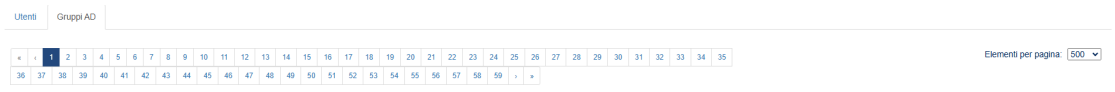

Figure 2.11. Long paginator component in Users/AD Goups page

The main element of the page is the table, which hosts users' data. It has a blue header row, where each field is sortable in both ascending and descending order simply by

clicking on the corresponding column header. Visual arrow icons are correctly displayed to communicate the actual sorting state to users.

Above many of the table columns, text input fields are provided to allow users to filter rows by string matching. This is a wise design choice because it enables users to apply filters without having to open additional windows or divert their focus away from the primary asset of the page: the user data. Moreover, it offers a clear and general view of all applied filters simultaneously.

However, some minor drawbacks may arise when considering visibility. Placing input text fields above the table columns significantly reduces the available width of these fields, making it difficult to read long filter strings at once. Additionally, this design limits the complexity of the filters that can be applied, as it supports only basic string matching. More advanced filtering options, such as dropdown menus, could improve functionality by allowing users to select from predefined values. This approach would not only offer users a complete overview of possible column values but also eliminate the need for them to memorize or guess the values, as they are all quickly available.

The main issue of the users' table lies in the hover behavior of its rows. As shown in Figure 2.10, when the mouse hovers over a row, the background darkens, and the text turns white. Even though this visual effect aligns with standard web design practices for indicating clickable content, it is misleading in this context. The rows are not actually clickable. Clicking on them performs no action. For example, no user detail page is opened, no copy operation occurs, and no options menu is displayed.

This misleading interaction represents a medium-severity violation of Nielsen's fourth heuristic: *Consistency and Standards*. Users expect consistent behavior where visual suggestions such as hover effects indicate interactivity. Failing to meet this expectation can lead to confusion and frustration.

As final step, the analysis of the users page focuses on the icon in the top right corner of the window, which represents the Excel logo in a blue version.

The first issue, though minor, violates Nielsen's *Match Between System & Real World* principle. Users typically expect the Excel logo to appear in its original green color. Even though this is not a critical problem that prevents in any way the usability of the system, users are familiar to this icon in its green, standard version. Although AROL IT portal users will still recognize it, given the logo's widespread accreditation, adhering to the original color scheme would be considered a best practice in software design.

The second, more significant problem about this icon lies in its behavior. After a click on it, an Excel file containing all users information is downloaded to the IT employee's device. This functionality is identical to that of the other Excel icon on the IT Dashboard page. However, similar to the other icon, this one does not include a downward arrow, which would clearly indicate that a download is occurring, instead of opening a page. The main issue here is that the two icons, while visually distinct, perform the same operation. This inconsistency can be considered a medium-severity error, as it creates confusion for users, slowing down their ability to memorize the portal's interface and familiarize themselves with its functionalities. For this reason, not keeping the same icon design

for elements performing the same operation, this issue can be classified as a violation of Nielsen's second principle: *Consistency and Standards*.

AD Groups

This second section of the users page focuses on the associations among the users and their Active Directory groups.

Active Directory (AD) is a directory service developed by Microsoft that facilitates the management of users, computers, and other objects within a network. It operates on the concept of a hierarchical structure, where objects such as users and groups are organized into directories. AD provides centralized authentication, authorization, and resource management, making it a critical tool for IT administrators in medium to large organizations. AROL leverages Active Directory to manage user information and access permissions. The IT portal retrieves user data directly from AD, ensuring consistency and accuracy in user details and group associations. However, it is important to note that the portal itself is not designed to modify AD data directly. Any changes to user information, such as updating group memberships or modifying user attributes, must be performed within the Active Directory system itself. [19]

AROL users may belong to one or more AD groups, which determine the specific permissions and authorizations assigned to them within the AROL's system.

For example, certain groups might provide specific permissions, such as the ability to send an email to all employees with an AROL account, access and manage files stored on OneDrive, or read the content of a particular folder within a system. These permissions define the level of control and accessibility that users have over different company resources, ensuring a structured and secure management of information.

The Active Directory Groups section has the same structure of the Users' one, for this reason, many of the problems already analyzed are still present here. Among them we can find again the Excel icon issue and row hover effect that doesn't match the entries behavior after a click.

In addition to them, a significant data presentation problem arises. As Figure 2.12 shows, each table entry corresponds to an individual association between a user and an Active Directory group. Each entry includes information about the user's company, domain, AD group name, and user details (name, surname, and username). Consequently, for every group a user is associated with, the same user information is repeated across multiple rows. This repetition results in an enormous waste of screen space and creates mental overload for users, who must find the right, relevant information among thousands of nearly identical rows. Moreover, since table rows are independent from each other by definition, this representation fails to accurately reflect the actual association between users and AD groups, where many lines refer to the same user (lines dependency).

This issue violates Nielsen's principle number eight: *Aesthetic and Minimalist Design*, as the interface contains unnecessary redundancies and lacks a simplified, efficient design. The severity of this problem is medium, as it does not prevent users from accessing the required information but significantly hampers their ability to do so efficiently.

The highest degree of severity (high) identified so far has been assigned to a critical

Company	Domain	AD Group	First Name	Last Name ▲	User Login	User Enabled
AROL Canelli	AROLGROUP	Domain Users	Filippo	Tresi	filippo.tresi	true
AROL Canelli	AROLGROUP	ANA.Folder_Lyra_RW	Filippo	Tresi	filippo.tresi	true
AROL Canelli	AROLGROUP	AROL.365_Users	Filippo	Tresi	filippo.tresi	true
AROL Canelli	AROLGROUP	AROL.Folder_FOTO_PDF ...	Filippo	Tresi	filippo.tresi	true
AROL Canelli	AROLGROUP	AROL.Service_Departm ...	Filippo	Tresi	filippo.tresi	true
AROL Canelli	AROLGROUP	AROL_Members	Filippo	Tresi	filippo.tresi	true
AROL Canelli	AROLGROUP	AROL.Folders_FS01	Filippo	Tresi	filippo.tresi	true
AROL Canelli	AROLGROUP	ANA.Folders_ANAFS01	Filippo	Tresi	filippo.tresi	true
AROL Canelli	AROLGROUP	AROL.Folder_JGalileo ...	Filippo	Tresi	filippo.tresi	true
Unimac-Gherri	AROLGROUP	Users	Filippo	Tresi	filippo.tresi	true
Unimac-Gherri	AROLGROUP	MDM-Limited	Giancarlo	Deido	giancarlo.deido	true
Unimac-Gherri	AROLGROUP	MDM-Mail-Office365-M ...	Giancarlo	Deido	giancarlo.deido	true
Unimac-Gherri	AROLGROUP	MDM-WiFi	Giancarlo	Deido	giancarlo.deido	true
Unimac-Gherri	AROLGROUP	MDM-WiFi-UG-WORK	Giancarlo	Deido	giancarlo.deido	true
Unimac-Gherri	AROLGROUP	MDM-Acrobat	Giancarlo	Deido	giancarlo.deido	true
Unimac-Gherri	AROLGROUP	Domain Users	Giancarlo	Deido	giancarlo.deido	true

Figure 2.12. AD Groups table, AROL IT portal, current version

violation of Nielsen's design principle number two: *Match Between System & Real World*. This issue refers to the way Active Directory groups are presented on the screen. As illustrated in Figure 2.12, AD groups are displayed only by their names, which are structured in a format that resembles system keys rather than human-readable identifiers.

This presentation creates a significant barrier for IT users, as they cannot easily discern the meaning or purpose of these group names unless they have direct access to the Active Directory system. Although users can try to guess the groups' purposes based on their names, this approach is both inefficient and prone to error. It has been categorized as high level of severity because users cannot understand what they are working on, by just looking at this page, so their work is compromised.

To resolve this issue, IT employees should be provided with clear, concise descriptions of each group in natural language. These descriptions should explicitly explain the authorizations and permissions associated with each AD group, ensuring that users can quickly and accurately understand their implications without additional system access or guesswork.

Note that from this point forward, many page designs and their corresponding principle violations may repeat, but they will not be discussed in detail unless they show unique aspects or further complications that justify additional consideration. The repetition of similar design issues across different pages highlights a systemic problem in the portal's overall approach to usability. Nevertheless, all violations, including the recurring ones, will be considered for in the final evaluation phase to ensure a comprehensive assessment.

Computers

Switching to the IT Dashboard tab corresponding to the *Computer* area, the user encounters a page similar to the one used for displaying users sections (Figure 2.13). Once again, numerous similar cards are presented, failing to reflect the varying levels of importance or relevance of the corresponding sections. Additionally, the 'XLS' download icons continue

to mislead users, as their design does not clearly indicate their function, contributing to confusion and inefficiency.

Figure 2.13. IT Dashboard - Computer tab, AROL IT portal, current version

By accessing the computers table through a click on the corresponding card, additional issues are revealed (Figure 2.14).

Although the table structure and functionalities are similar to those in the users and AD groups sections, the computers page displays significantly more information per row. As a result, table entries exceed the screen's width, forcing users to horizontally scroll the page to access computer details located on the right side of the table. Even though horizontal scrolling is a standard interaction in certain contexts, it is often discouraged as it is counterintuitive and interfere the user's experience. This can be considered as a violation of the Nielsen's eighth principle (*Aesthetic & Minimalist Design*) with a medium degree of severity. A better design would reorganize the information to fit within a single horizontal row, potentially omitting less relevant columns to provide a more user-friendly view.

Computer assignment

Switching to the *Computer Assignment* tab, users are presented with a list of all associations between computers and users, as illustrated in Figure 2.15. The primary issue lies in how these associations are communicated. The table column labeled "Computer Name" identifies computers solely by their computer name field, which is essentially a code, without providing any additional information.

If users want to access further details about a specific computer, they must navigate to the computers table and, relying on their memory of the computer name, filter the

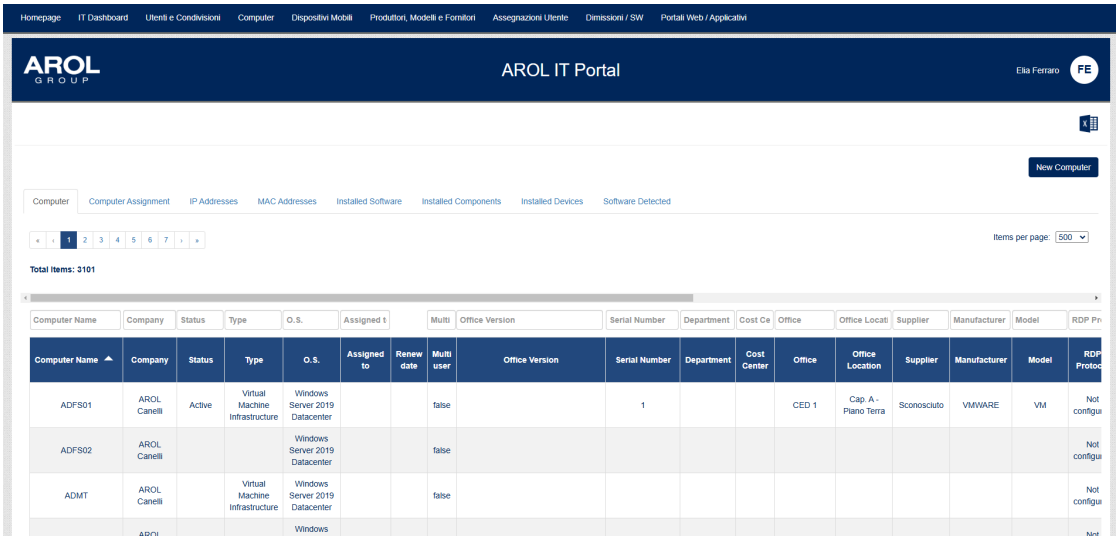

Figure 2.14. Computers page, AROL IT portal, current version

rows to locate the relevant entry. This workflow imposes an unnecessary cognitive load on users, as it forces them to recall information rather than simply recognizing it within the interface.

This design problem worsens the user experience and constitutes a violation of Nielsen's design principle number six: *Recognition Rather than Recall* with a medium degree of severity.

Figure 2.15. Computer assignments, AROL IT portal, current version

The same design issue has been replicated across all other tabs within the computers

section. For instance, similar issues can be observed in the tables listing IP addresses, MAC addresses, and their associations with computers.

View / Edit / Add Computer

This subsection focuses on a specific page that appears in three different but closely related contexts due to the similar structure of the data presented. The page, shown in Figure 2.16, contains the form used to add a new computer to the system. Since information required for creating a new computer are the same as those used for editing an existing one, this form also serves for editing purposes. Additionally, the same form is implicitly used when users only want to view computer details for a specific device.

Although this approach is not considered a best practice, since users searching for specific information about a computer do not need form elements like dropdown menus, input fields, and buttons that may distract them, it can be accepted as a partially effective solution. For this reason, no usability violation has been identified in this case.

Before discussing this page in detail, it is important to address how users navigate to it. The most intuitive and expected method is by clicking on a computer entry in the corresponding table. This interaction is logical and appropriate, as the computer details page is hierarchically linked to the computers overview. Similarly, accessing the page by clicking the "New Computer" button is clear and straightforward, aligning with user expectations.

However, other navigation paths are less intuitive. For instance, clicking on a row in the IP addresses table also redirects to this page. This interaction can be confusing because users might expect to access a page for modifying the selected IP address or managing related settings. Instead, they are redirected to the computer details page associated with that IP address. This inconsistency in navigation occurs in other sections as well, such as the software and devices tabs. Even if this issue does not impact usability, getting a low degree of severity, it still is a violation of Nielsen's principle number three: *User Control & Freedom*.

The page has a simple and well-organized structure that is intuitive, even for novice users. It is divided into sections, allowing users to focus on specific data areas, such as computer details, IP addresses, and BitLocker codes.

However, the page presents some usability issues, particularly in the lower section, where associations between the computer and other entities are managed. When users attempt to add or remove these associations, all input fields accept only plain text, requiring users to manually enter information. For instance, when assigning a user to a computer, a small pop-up window appears (Figure 2.17), asking the user to input the identity of the person to be linked. However, the input field doesn't provides any clue on what specific user information is required, for example a surname, email address, or user ID. A more effective solution would have been the implementation of dropdown menus (as used in the upper sections of the page) that list all assignable users, displaying their names, usernames, or email addresses. This approach would speed up the process and reduce cognitive load.

Figure 2.16. View / Edit / Add Computer page, AROL IT portal, current version

This issue represents a medium-severity violation of Nielsen's principle number six: *Recognition Rather than Recall*. By forcing users to independently determine the correct

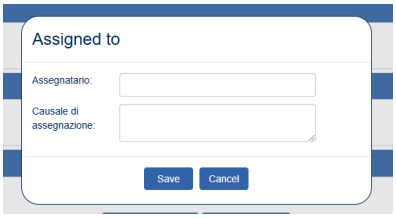

Figure 2.17. Assign computer, AROL IT portal, current version

user to link, the system increases the potential for error and reduces overall efficiency.

Other pages

After discussing these initial pages of the application, it is important to note that the primary goal of this thesis is not only to provide a formal and comprehensive analysis of the current version of the AROL IT portal. For this reason, the remaining pages will not be described in detail. This decision is based on the fact that the other sections share the same structure and design problems previously discussed, including recurring violations of Nielsen's usability principles. However, these sections have been fully considered in the overall evaluation of the portal, which will be presented in the next section.

2.1.4 AROL IT portal (current version) evaluation

To provide a clear and general overview of the IT portal, Table 2.2 summarizes all the usability violations identified during the analysis. Each violation is represented by its corresponding design principle number, based on Nielsen's heuristic system. Additionally, the table includes a description of the specific location where the violation occurred and a brief explanation of the nature of the issue.

PN	Where	What	Severity
P1	Navbar	Active page links are not visually distinct	low
P3	View / Edit / Add pages	These pages are accessible from multiple, sometimes unexpected, points	low
P4	All cards	XLS download icon lacks a downward arrow and hover effect, making its function unclear.	low
P4	Navbar	AROL logo and Homepage link perform the same action but have different appearances	low

Continues on the next page...

PN	Where	What	Severity
P7	Material confirmation	Blank page displayed if no material requires confirmation	low
P8	All paginators	Paginators display all page numbers instead of a compact format	low
P1	All pages	Mixed Italian and English language for English-speaking users only	medium
P1	Navbar	Subpage links are displayed in a plain hierarchy	medium
P1	IT Dashboard	Navbar links and IT Dashboard tabs are duplicated with inconsistent behavior	medium
P2	All pages containing tables	Excel download icon vary in design and color (blue instead of the standard green) without a clear symbol	medium
P4	Users table	Table rows have a hover effect even though they are not clickable	medium
P6	Computer assignment table	Computers are identified only by code, with no direct link to details, forcing users to search manually.	medium
P6	Add / Edit relation in all details pages	Input fields require users to recall names or codes instead of selecting from a list.	medium
P8	AD Groups table	Repetitive entries for the same user overcrowd the table	medium
P8	Computers table	Excessive columns cause horizontal scrolling, disrupting navigation	medium
P2	AD Groups table	Group names are in code format, making them incomprehensible to users	high

Table 2.2: Nielsen's heuristics violations summary (AROL IT portal, current version)

To deliver a transparent and objective evaluation of the current AROL IT portal, it is essential to quantify the overall usability based on the identified design issues. The most effective approach is to calculate a single numerical value that reflects the severity of all detected heuristic violations. This metric offers a brief summary of the portal's usability

state. Established that each severity level corresponds to an integer number (low: 1, medium: 2, high: 3) as previously explained, the computation of the final value follows this mathematical formula, corresponding to the arithmetic average:

$$\text{Average Severity} = \frac{\sum (\text{Severity Weight} \times \text{Number of Occurrences})}{\text{Total Number of Violations}} \tag{2.1}$$

To make the evaluation more explicit, the computed score can be mapped to usability ratings:

- 1.0 – 1.5: Good Usability – Minor issues, overall user-friendly
- 1.6 – 2.0: Moderate Usability – Noticeable issues affecting user tasks
- 2.1 – 3.0: Poor Usability – Critical problems requiring immediate redesign

The result obtained is 1.69, meaning that the overall usability of the AROL IT portal falls into the "Moderate Usability" category. This score suggests that, while the portal is functional, there are noticeable design issues that affect the user experience. These issues, while not immediately critical, still require attention and could prevent users from effectively completing tasks.

To improve the portal, it would be important to prioritize addressing the high-severity issues, as they have the most significant impact on usability. After resolving these, the medium and low-severity issues should also be solved to enhance the user experience and bring the overall score closer to the "Good Usability" range.

Chapter 3

Requirement Analysis

In this chapter, all the project requirements that must be satisfied by the new version of the AROL IT portal will not only be listed but also described, and where the reasoning behind each requirement will be highlighted. The goal is to understand the company's needs, identify what is already provided by the current application, and determine which new features must be implemented from scratch.

Before starting the discussion, a clarification is necessary. Often, when developing new software, whether it is a completely new application or the maintenance of an existing one through the introduction of new functionalities, the organization commissioning the project does not have a clear and well defined idea of the final product. Even though they are experts in their field and understand the processes behind their work, the units involved, and the information needed to execute tasks, they are often unable to provide a complete and exhaustive requirements document for designers and developers to use in creating the new product.

For this reason, several meetings are required to allow the requirements engineer, represented in this thesis project by the candidate, to analyze the context. This involves asking questions to the employees who will actually use the new application and filtering and transforming the answers provided by the company into a set of requirements.

The following is a list of some of the main and most relevant questions posed to company members during meetings:

1. Who are the primary users of the IT portal?
2. Is there any hierarchy among the IT portal users?
3. Which tasks are the most time-consuming and resource-intensive?
4. What are the key tasks currently performed using the IT portal?
5. Can you describe the process for onboarding a new hired employee?
6. How does the process of purchasing a new device work?

7. How is the process of scrapping or returning a device managed?

8. How are software licenses assigned to users?

The answers to these and many other questions have been used to define the requirements discussed in the following sections. These sections are divided into two categories: Functional and Non-Functional requirements [13]. This classification follows standard practices in software development and allows for a clear and organized representation of the system's needs. Despite addressing distinct aspects of the system, both categories work together to provide a complete and integrated foundation for the design and development of the new IT portal.

3.1 Functional Requirements

The first category of requirements addressed is the Functional one. This category is generally the easiest to conceptualize due to its practical and straightforward nature.

Functional Requirements define the core features and capabilities that the application must provide to meet the company's operational needs. Although they are fundamental, they can still involve significant technological complexity. These requirements are typically the ones the company prioritizes most, often offering an initial list that the engineer later refines and structures.

They are characterized by their clear structure, typically involving three key components: the input the system receives, the processes or operations it performs, and the resulting output. This format makes them easily identifiable and directly connected to the system's behavior.

Moreover, functional requirements have a tangible impact on the final product and are relatively straightforward to evaluate. Their successful implementation can be tested and verified, providing clear evidence of whether the system performs as expected.

In the context of the project, functional requirements have been further categorized into three classes described below:

- **High Priority**: Core functionalities that are critical for the system's operation. Without these, the system would fail to fulfill its primary purpose.

- **Medium Priority**: Important features that enhance usability and efficiency but are not essential for the basic functioning of the system. Their absence would not compromise core operations but might reduce user satisfaction and productivity.

- **Low Priority**: Additional, non-essential features that offer extra value or customization. These are considered optional and can be implemented if time and resources allow, without impacting the system's core functionality.

3.1.1 High priority functional requirements

In this subsection, the primary functionalities that must be satisfied by the end of the project will be outlined. These are essential features for the AROL IT portal. The absence of even a single one of these functionalities would render the project a failure, as IT employees would be unable to incorporate the system into their daily workflow. These functionalities are the most fundamental, and typically, they do not involve significant technological complexity. They mainly consist of CRUD (Create, Read, Update, Delete) operations on the entities within the system. Additionally, often the majority of the functionalities of software applications fall under this category.

The new version of the AROL IT portal must include at least the same functionalities as the previous version. This ensures that users can continue performing the tasks they are used to, though potentially through a more revised or improved process. It is essential that these basic, fundamental functionalities remain present, as their absence would disrupt users' established workflows. If these functionalities are considered indispensable, they must be incorporated into the new version without fail.

For this reason, all the previous functionalities have been placed in this priority category, as they form the foundation upon which new, improved, and more specialized features can be built. Note that these functionalities have not been explicitly listed here for shortness, as they are accessible in Table 3.1.

Below are the newly added requirements, which are not present in the current version, that must be satisfied in the new one.

FRN	Name	Description
1	Display user's devices	Display all devices and assets assigned to a specific user in a consolidated view on the user's page
2	Display user asset summary	Show a quick summary of all devices assigned to a user (e.g., 2 computers, 1 mobile, 0 peripherals)
3	Display AD group descriptions	Show descriptions of Active Directory (AD) groups in natural language on the user page
4	Display device location	Display the physical location of a device on its dedicated page for better inventory management
5	View all unassigned devices	Allow users to view all devices that are available for assignment, making it easier to manage inventory

Continues on the next page...

FRN	Name	Description
6	Automatic HW codes when adding devices	Automatically generate hardware codes when adding devices to the system, speeding up the process

Table 3.1: High Priority Functionalities required for the AROL IT portal.

As shown in the table, most of these functionalities focus on displaying previously missing information, such as Active Directory group descriptions, which are crucial for IT users. Additionally, some features will significantly streamline the device assignment process—for example, by directly displaying devices that are available for assignment.

While these features are technically simple to implement, they are essential for improving the system's usability and efficiency.

3.1.2 Medium priority functional requirements

Moving on to the medium priority functional requirements, the focus shifts to features that should ideally be implemented by the end of the project but have less critical importance and urgency compared to previous one. These functionalities are designed to enhance employees' workflow by speeding up tasks, automating repetitive processes, and improving overall efficiency.

Even though the absence of these features would not compromise the core functionality of the system, their implementation would contribute to a more user-friendly and productive experience. These requirements often involve optimizing existing processes rather than introducing entirely new capabilities. Examples include providing summarized overviews, offering more intuitive navigation, and reducing manual input where possible. Their development will depend on available resources and time constraints but should be considered positive additions that can highly enhance the usability and performance of the AROL IT portal.

The following Table 3.2 presents the list of medium priority functional requirements set up for the project.

FRN	Name	Description
7	View remaining software licenses	Show how many licenses are still available for each software package
8	Navigate between device and assignee	Enable navigation from a device's page to its assigned user and vice versa

Continues on the next page...

FRN	Name	Description
9	Add SIM tariffs with a click	Enable users to quickly add tariff options to SIM cards with a simple click
10	Assign a device with a click	Allow users to assign a device to a user with a simple click, rather than manually entering device codes or user identities
11	Assign a license with a click	Allow users to assign software licenses with a simple click rather than manually searching or inputting data
12	Return a product with a click	Enable users to return a product or device with a single click, simplifying the process
13	Add devices from templates	Allow users to add new devices using predefined templates for faster setup
14	Scan QR code to access product page	Allow users to scan a QR code to access a product's page directly

Table 3.2: Medium Priority Functionalities required for the AROL IT portal.

3.1.3 Low priority functional requirements

Finally, the low priority functional requirements consist of optional features that, though not essential to the main functionality of the AROL IT portal, could improve the user experience. These features are considered nice-to-have and will be implemented only if time and resources permit after the high and medium priority requirements have been satisfied.

Their primary purpose is to provide added value by introducing personalization, advanced integrations, or improvements of the quality of employees' work that can make the portal more adaptable and user-friendly. However, their absence would not negatively impact the daily operations of IT employees or the overall functionality of the system.

Low priority functional requirements are often developed for advanced users who are already familiar with the standard functionalities of the tool and can effectively leverage more specialized features to significantly enhance their workflow. Although these features may have a harder learning curve, they offer substantial benefits in terms of efficiency and productivity for experienced users who can fully exploit their potential.

FRN	Name	Description
15	View entity event history	Display a detailed history of actions and events related to each device or asset
16	Generate device QR codes	Generate QR codes for each device to simplify access to device information
17	QR code action menu	Provide a menu of available actions when scanning a device's QR code
18	Batch actions via QR scan	Enable batch processing by scanning multiple devices in sequence to perform the same action

Table 3.3: Low Priority Functionalities required for the AROL IT portal.

These requirements primarily aim to enhance system usability and speed up device management through the use of QR code technology and detailed tracking. Even though not critical for the portal's core operations, these features can significantly improve efficiency by simplifying access to device information and enabling faster execution of repetitive tasks. Their implementation may also evolve based on future user feedback, allowing for further optimization of workflows and user interaction with the system.

After discussing the functionalities expected to be developed by the end of the project, it is now essential to address another crucial component for the quantitative evaluation of the new version of the AROL IT portal: the Non-Functional Requirements.

3.2 Non-Functional Requirements

Non-Functional Requirements define how a system should operate rather than what it should do. While functional requirements specify the application's behavior and features, non-functional requirements focus on the quality of the product, setting performance and usability goals. As previously mentioned, this category of requirements is generally more challenging to define and evaluate compared to functional ones. Functional requirements are directly observable through the use of the application, while non-functional requirements require specific tests, measurements, and complex evaluations to be assessed.

Despite this complexity, non-functional requirements are crucial performance indicators. Their quantifiable nature allows stakeholders to objectively evaluate if the project has succeeded by comparing the measured results against predefined goals. If these requirements are met, the product is considered to have achieved its intended quality standards.

To extensively cover all aspects of application development, non-functional requirements can be divided into three main categories: *UX Quality*, *Frontend Scalability*, and *Maintainability and Code Quality*. Although many classification systems could be equally

valid, the chosen structure best reflects the project goals by highlighting specific quality attributes that are crucial for this type of development.

3.2.1 UX quality

This group of non-functional requirements focuses on the users' perception of the system and how they interact with it. It is one of the most critical categories because it directly influences user satisfaction during interaction with the application. If any of these requirements are not met, users may feel overwhelmed or insecure, making it difficult for them to complete their tasks effectively. This can lead to frustration and ultimately cause users to abandon or dismiss the application.

It is crucial to recognize that even if functional requirements are satisfied, offering a wide range of features and capabilities, allowing users to perform various operations, and providing high levels of freedom with customizable workflows, if the interface is not usable, intuitive, or efficient in helping users find the information they need in a reasonable amount of time, the application will become ineffective. Users will soon abandon it, regardless of the functionalities it offers. Therefore, user experience and usability are vital to the success and longevity of any software.

In the context of the AROL IT portal's frontend, UX quality can be assessed by focusing on the experience of novice users. To consider the project a success, it is essential that these users can quickly locate the information they need and navigate the interface autonomously to identify the correct steps for completing their tasks. Achieving this goal requires careful attention to design principles that prioritize clarity and ease of use.

Several strategies can be employed to improve the user experience, especially when combined. For example, adopting standard icons and terminology that users are already familiar with from widely used applications can help them. Additionally, guiding users through the interface by implicitly suggesting the appropriate path using intuitive design elements, such as appropriate color schemes and clear hierarchy sizing, can significantly improve navigation.

Another critical aspect to consider is accessibility, particularly in the context of the AROL IT portal. Accessibility in this case consists in ensuring that users can access and use the portal from a variety of devices, each with potentially different sizes, computational capabilities, and screen resolutions. Since the portal is designed to be used in various scenarios, it is essential that it functions smoothly across different platforms, whether users are in the office using a desktop computer or in mobility with a smartphone. Although the specific operations they perform may vary depending on the device (e.g., using a desktop may offer more complex tasks, while a mobile device may streamline operations for on-the-go use), the efficiency and ease of use should remain the same.

The following Table 3.4 lists the principal UX quality non-functional requirements.

NFRN	Name	Description
1	Usability	The system must be easy to use for both experienced and novice users. It should require minimal effort to perform basic tasks. In addition, the interface should be self-explanatory and require minimal training. Users should be able to figure out how to navigate and use the system quickly. Users must find the information or the functionality they need in less than 3 minutes.
2	Responsiveness	The system must be compatible with various devices and platforms (e.g., desktop, tablet, mobile). It also should adjust its layout for different screen sizes to provide an optimal user experience across devices. This requirement is considered satisfied if the system carefully displays information without any distorted or strange-looking pages.
3	Consistency	The design and layout must be consistent across all pages and functionalities of the system, ensuring a unified user experience.
4	Visual Standards	The interface should be visually appealing, aligning with modern design trends while ensuring clarity and readability. In addition, best practice design standards must be applied, trying to get close to famous and consolidated applications.
5	Error Management	The system should handle errors smartly and provide informative error messages that guide the user on how to resolve the issue.

Table 3.4: Non-Functional Requirements for UX Quality in the AROL IT portal.

3.2.2 Frontend scalability

By lowering the level of evidence for non-functional requirements, the second category is considered. This category focuses on transitioning the application from a test environment (developer's machine) to a production context. During this phase, the application must be capable of handling real data and the challenges associated with processing such data. It is not as immediately apparent as the UX Quality category, because it becomes more noticeable when users are dealing with complex tasks.

In the case of the AROL project, while the number of users accessing the application at any given time will be limited to the IT department, the data the system will manage

is extended, coming from the entire AROLGROUP. This requires the application to be designed to efficiently handle large datasets while ensuring that users can easily consult and navigate through this massive amount of information. The system should provide a clear and intuitive way for users to find relevant data without feeling overwhelmed.

Additionally, data loading should be handled in a way that allows users to continue interacting with the application while the data is loaded in the background. This approach ensures the platform remains available, avoiding the need for users to wait for all data to load before proceeding with their tasks. This is crucial for maintaining the overall user experience and ensuring that the system remains efficient, even when dealing with large quantities of data.

The following Table 3.5 lists the principal frontend scalability non-functional requirements.

NFRN	Name	Description
6	Data Handling	The system should efficiently handle large datasets without significant delays, ensuring smooth user experience even with massive amounts of data being processed or displayed.
7	Data Loading	The system must load data asynchronously to improve performance, ensuring that the user can interact with the interface without waiting for the entire dataset to be loaded. To improve data loading times and user experience, large datasets must be divided into smaller chunks (pagination) and displayed progressively, reducing the need to load all data at once. Also, lazy loading can be used to load pieces of data only when needed, when users request it.
8	Caching	To avoid loading the same piece of data multiple times, caching some previous values is a good solution.

Table 3.5: Non-Functional Requirements for Frontend Scalability in the AROL IT portal.

3.2.3 Maintainability and code quality

The less evident category of non-functional requirements is represented by the *Maintainability and Code Quality* class. It represents the qualities that concern the code itself and its maintainability phase. It corresponds to the best practices in writing structured code,

by using modular components and following coding standards. Ensuring high maintainability allows the future development team to efficiently and easily introduce updates, fix bugs, and implement new features without compromising the system's general structure.

In the AROL context, the *Maintainability and Code Quality* category is particularly important due to the need for continuous updates and improvements to the IT portal. Since the company is an always-growing firm, AROL IT portal must remain adaptable to future requirements without causing disruptions. Additionally, considering that the development team may change over time, maintaining high code quality ensures that future developers can easily understand, modify, and expand the system.

Table 3.6 shows the main requirements of this section.

NFRN	Name	Description
9	Modular Architecture	The frontend must be developed by separating the code into components that can be later reused declaratively in order to speed up the maintenance phase
10	Design Patterns	Standard Angular design patterns, such as services for data management and lazy loading for optimizing performance, should be implemented. This not only improves scalability but also simplifies future feature expansions, being easier to understand by other developers.

Table 3.6: Non-Functional Requirements for Maintainability and Code Quality in the AROL IT portal.

3.3 Stakeholders and Constraints

The last topic of this chapter focuses on the context around the project, identified by its stakeholders and the constraints set before the starting of it.

3.3.1 Stakeholders

Stakeholders in software development are individuals or groups who are directly or indirectly impacted by the software product. They often belong to different organizational levels and departments, as multiple parties contribute to or are influenced by the software's development and use. Stakeholders have various interests in the project, such as economic benefits, operational efficiency, or usability, depending on their role [11].

Stakeholders are often experts in their respective domains or users whose needs the software is intended to address. Their feedback and suggestions play a critical role in

defining the software's features and guiding the overall direction of the project. Effectively engaging stakeholders ensures that the final product aligns with user needs and business goals.

Generally, stakeholders in a project can be divided into three categories based on their level of interest in the software product. The first category consists of **direct users**, who are the actual users of the application who interact with it regularly. They have the highest interest in the product because it is designed to solve their problems or improve their workflow. The second category includes **secondary users**, who may not use the software directly but interact with its outputs or results. They are important to consider during the requirements phase because they can provide added value, often having a wider and more comprehensive understanding of the project's context. Lastly, there are the **beneficiaries**, who are indirectly affected by the software's products. They are the most distant from the software and, in many cases, may not even be aware of its existence. However, they are significantly impacted by how the other two user groups use the software. Beneficiaries are more concerned with the results produced by the software than with the software itself.

In the context of the AROL IT project, the main stakeholders (Figure 3.1) can be categorized according to their specific roles. IT Employees are the Direct Users, actively interacting with the application to assign devices to users, add new computer models, and manage the scrapping of mobile devices. AROL Management, representing the Secondary Users, do not use the application directly but rely on the information it provides to support strategic decisions and company improvements. Finally, AROL Employees belong to the Beneficiaries category, as they are indirectly affected by the IT department's work, for instance, when receiving devices or being assigned new software licenses. The quality of the IT department's operations, supported by the application, directly impacts their efficiency.

3.3.2 Organizational constraints

Lastly, the discussion shifts towards the analysis of the project's constraints. Although none were explicitly imposed by stakeholders, they naturally emerge from the project's context.

One key constraint is the **alignment with company priorities**: while no strict deadline was set, the project is expected to fit within the company's operational timelines and internal objectives, establishing a tentative maximum duration of five to six months. Effective time management is essential to meet these expectations.

Another significant constraint is the **resource availability** in the IT department, which has limited capacity for continuous support during development. This requires the candidate to work independently, optimizing communication with the IT team and minimizing their involvement to avoid disrupting their ongoing responsibilities. In particular, during the development phase, integrating the frontend with the backend demands a high level of autonomy, as the database could not evolve according to the new frontend.

Finally, the project follows the **Waterfall development methodology**, a structured and sequential approach where each phase (requirements analysis, design, development,

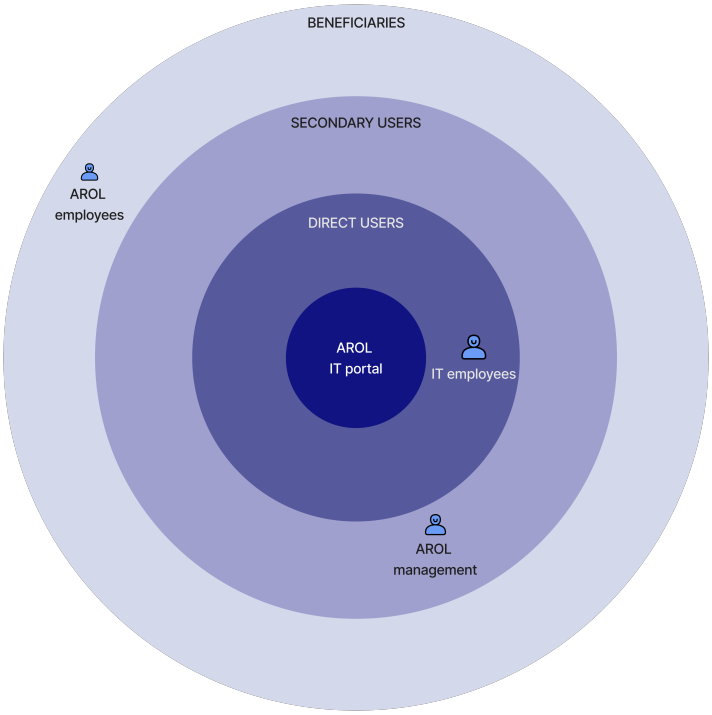

Figure 3.1. AROL IT portal's stakeholders

and testing) must be completed before moving on to the next. This methodology requires detailed planning and careful execution, as implementing changes later in the process can be challenging.

Chapter 4

Design and Prototyping

This chapter focuses on the methods and considerations discussed during the design phase. After defining the functional requirements of the new frontend and identifying how it should support its users, the design phase becomes a critical step in the software development process. It poses the foundations for a more stable and robust product, ensuring that stakeholders approve the design before moving on to the implementation phase. Any changes to the design after implementation begins would require additional work and effort, making the design phase crucial for the project's success.

4.1 System Design

Before proceeding with the design of individual pages and components (such as buttons, cards, and images), it is important to first consider the new Information Architecture.

The new version of the portal's frontend reuses the same Information Architecture as the previous one, as it effectively supports user tasks and objectives. The main patterns, *Hierarchy* and *Tabbed View*, remain because they align well with the portal's goals.

However, these patterns have been refined by reducing the hierarchy depth and addressing inconsistencies from the previous version. Specifically, since the primary actions on the portal are performed in the IT Dashboard section, this section has been removed. Now, the entire portal serves as the IT Dashboard, so users will be directed straight to the relevant data after accessing the portal.

Another important change concerns entities that IT users do not interact with frequently, such as the lists of companies, suppliers, or computer statuses. Since these entities are typically used only for association with more important data and are not modified daily, their placement has been moved. These lists have been relocated from their original positions to the Settings section, where they can still be edited but are now less notable and distracting to users.

Based on these considerations, the following list outlines the most relevant entities for

the IT users' workflow in the new application:

- Users
- Computers
- Peripherals
- Mobile devices
- Sim cards

It is important to note that the entity listed here as *Peripherals* corresponds to the *Computer devices* entity from the previous version of the frontend. This change was made after discussions with IT portal users, who expressed confusion over the term "computer devices" and its meaning. To improve clarity and align with Nielsen's second design principle, *Match between System & Real World*, the nomenclature was updated.

The diagram in Figure 4.1 illustrates the evolved page schema, displaying how the pages are interconnected and how users will navigate through the application to reach their final destinations.

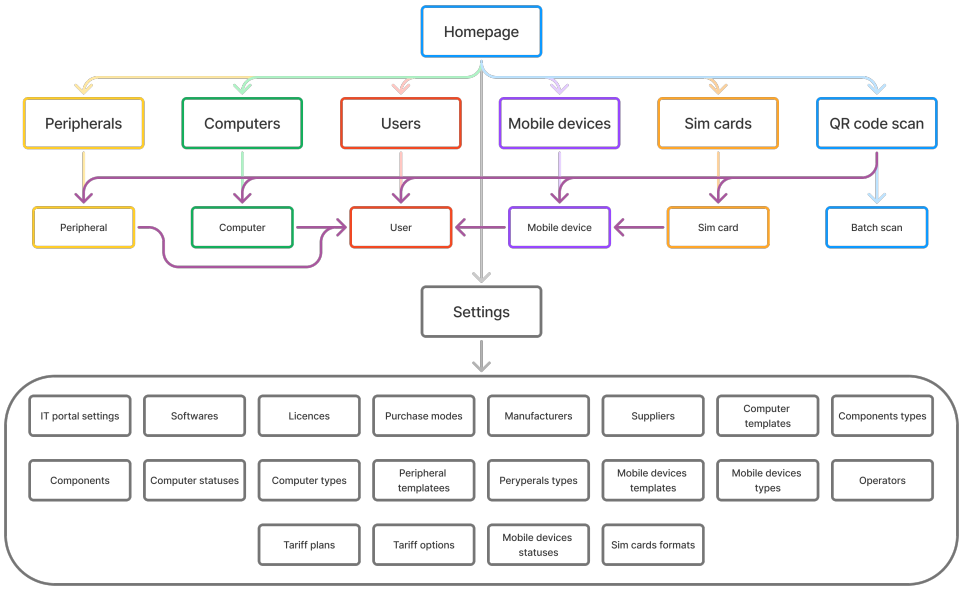

Figure 4.1. AROL IT portal's pages schema, new version

4.2 Prototyping with Figma

Once the basic principles and design strategies have been established, such as outlining the IA pattern, the next step is the actual design phase. In this section, the methodologies used and the critical decisions made will be analyzed, with a focus on the reasons behind them.

4.2.1 Prototyping methodologies

In software design, various tools and techniques can be employed to create a prototype of the future product, depending on the level of detail required and the goals of the design phase. These goes from low-fidelity prototypes, where only the main elements of pages are sketched (often by hand on paper), to high-fidelity prototypes, which closely look like the final product and are typically implemented using real technologies and code.

In the context of the AROL IT portal, since the current application is already in use and contains a lot of functionalities (some of which will suffer significant modifications), the main goal of the design phase is to present a prototype to stakeholders without the need to implement the entire new portal. This approach saves considerable time, especially when considering the inevitable changes stakeholders will probably request.

For this thesis, the applied methodology is *Rapid Prototyping* [18]. This approach involves selecting a tool that allows for quick prototyping, enabling rapid modifications with minimal time investment. The goal is to ensure that the designer can easily edit the prototype by adding new features or adjusting existing ones without the need to code everything again.

Three fundamental steps should be followed cyclically.

The first phase is **Prototyping**, where designers create the prototype using the most suitable tools and technologies for the project's context.

Once the prototype is ready, it moves to the **Feedback** stage, where stakeholders evaluate it from different perspectives, including functionality, requirements conformity, and usability. They then provide feedback, highlighting any issues or problems related to the prototype and the design choices.

Finally, in the **Improvement** phase, the received feedback is carefully analyzed to identify design problems, and necessary modifications are applied to refine the prototype. Once the improvements are implemented, the prototype undergoes another feedback session, repeating the cycle until stakeholders are fully satisfied with the result.

4.2.2 Prototyping tools

The primary tool used to produce the prototype is Figma [12], a cloud-based web application for UX/UI design. Although several alternatives were available, Figma appeared as the best solution for several reasons.

First of all, Figma is one of the most popular and widely supported design tools. Its large user base suggests long-term support and continuous development. Additionally, the wide community ensures the availability of thousands of guides and tutorials online, providing valuable assistance in case of difficulties or uncertainties.

The second reason behind the candidate's choice is Figma's design focus on the needs of designers and developers. Its workflow and interface are made for use in software development contexts, enabling a design approach that mirrors development practices. For example, it allows the definition of components similarly to how they would be created in a development environment.

The third reason is that Figma is not just a design tool for creating and planning user interfaces, it also supports rapid prototyping by allowing users to create simple interactions between screens and components. This feature provides a realistic sense of the product's behavior without requiring a single line of code.

Lastly, Figma is a collaborative tool. This feature allowed AROL members involved in the thesis project to continuously review the progress in real-time and provide immediate feedback, even before formal meetings with the entire project group.

4.2.3 Final approved prototype

In this last section, the last, final, version of the prototype approved by AROL's stakeholders is presented.

The developed mock-up is a medium-fidelity prototype. It is neither a sketch nor a highly detailed and precise representation of the final product. Its purpose is to provide future users and stakeholders with a clear view of the type of information available in the new portal, the components that will structure the pages, and the interactions they will experience. For this reason, no color scheme has been applied, and no images have been included. A plain greyscale design with simple components has been chosen because it is quicker to create and better supports future modifications and various implementations, depending on the chosen technological frameworks.

Homepage

The first page presented is one of the temporarily last sections of the portal designed in the prototype due to its free nature, since developers are more free to experiment and be creative: the *Homepage* (Figure 4.2). For this reason it has been left quite untouched in order to experiment later the best solution for it, also accordingly to the other pages.

This page is designed to provide an overview of the entire system, allowing IT users to quickly identify if any tasks require their attention through a dashboard highlighting pending actions.

The most notable feature on this page is the *Scan Product* button, which is intentionally more visually distinct than other elements to emphasize its primary function,

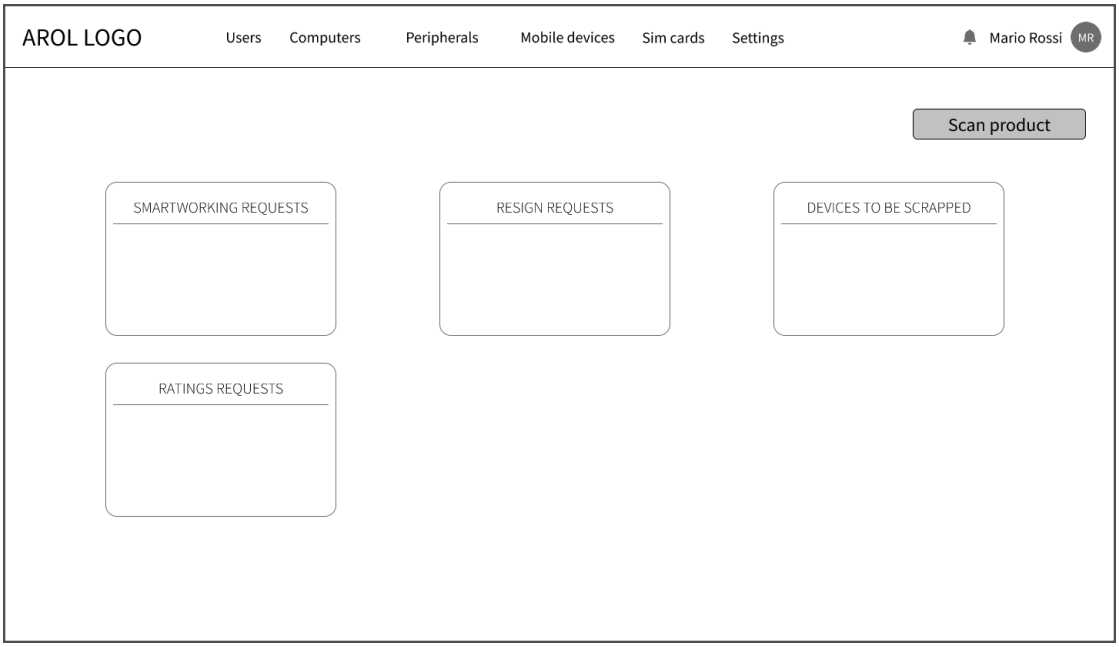

Figure 4.2. Homepage, AROL IT portal, prototype

directing users to the corresponding page for product scanning.

In addition, the navigation bar has been slightly improved to consistently display the main sections of the AROL portal, ensuring they are always accessible from any part of the application.

Users / Computers / Peripherals / Mobile devices / Sim cards pages

All the other pages that represent the list of entities like the users one, showed in Figure 4.3, have been designed following the same, consistent strategy, ensuring IT employees can familiarize with it in the minor time possible.

The main tabular structure has been kept untouched from the previous portal, as it effectively displays large amounts of data. However, only relevant information are now included in the tables, with superfluous details removed. For example, in the users' table, data about users' passwords information has been excluded since it is not relevant in a general tabular view. Additionally, disabled entities are now more distinguishable, not only by a "disabled" tag, but also through a different row background color, enhancing the visual communication of their status.

In the top right corner of the screen, the "Filters" button has replaced the input field row from the previous version to introduce a more sophisticated filtering system. This button shows a badge designed to draw users' attention by displaying the number of

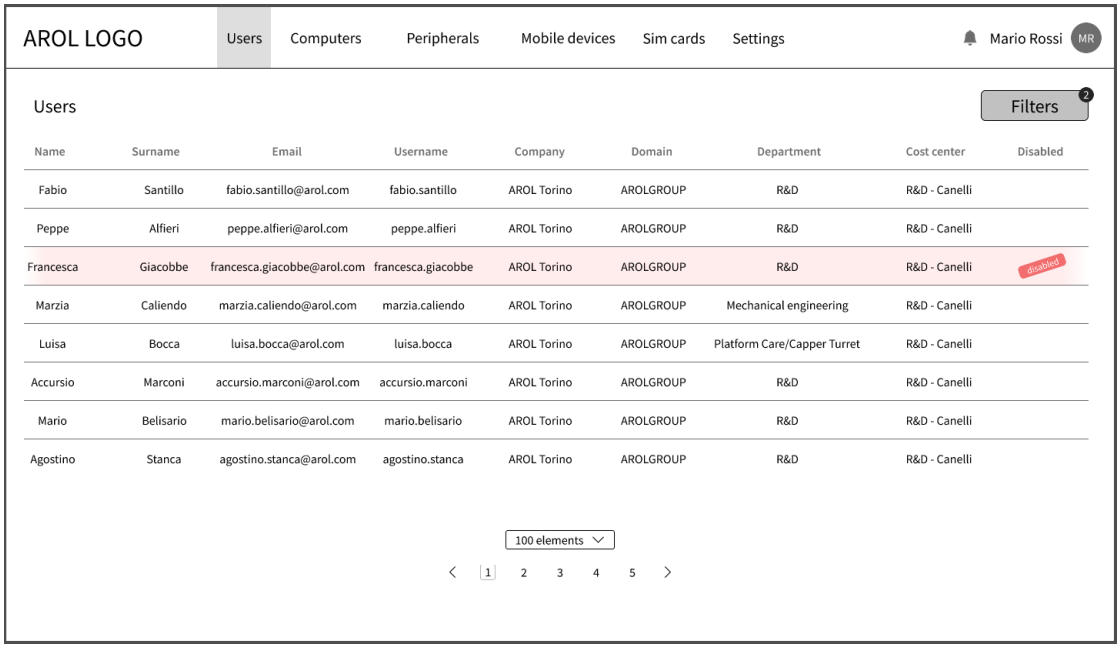

Figure 4.3. Users page, AROL IT portal, prototype

active filters, if any. Figure 4.4 shows the dialog that appears after clicking the "Filters" button. This dialog-based solution offers a clearer and more organized filtering experience compared to the previous inline input fields, as it separates filtering functions from the main table view.

Within the dialog, users can create new filters by selecting table fields, entering desired values, and removing filters using dedicated buttons. Additionally, shortcut buttons under the filter dropdown allow users to quickly apply predefined filters with a single click, speeding up the filtering process.

User / Computer / Peripheral / Mobile device / Sim card pages

By clicking on a table row, users are redirected to the page corresponding to the selected entity. This behavior was already present in the previous version but was not implemented for the user entity. This uniform approach will be appreciated by IT employees, as it ensures consistency across all tables.

As shown in Figure 4.5, the user page is vertically divided into three sections to emphasize the different nature of each.

The first section displays detailed and comprehensive information about the specific user. This section functions like an identity card, presenting key details in an organized manner. It is further divided into subsections grouped by related topics, such as personal information, passwords, and an analytics section. The background color scheme follows

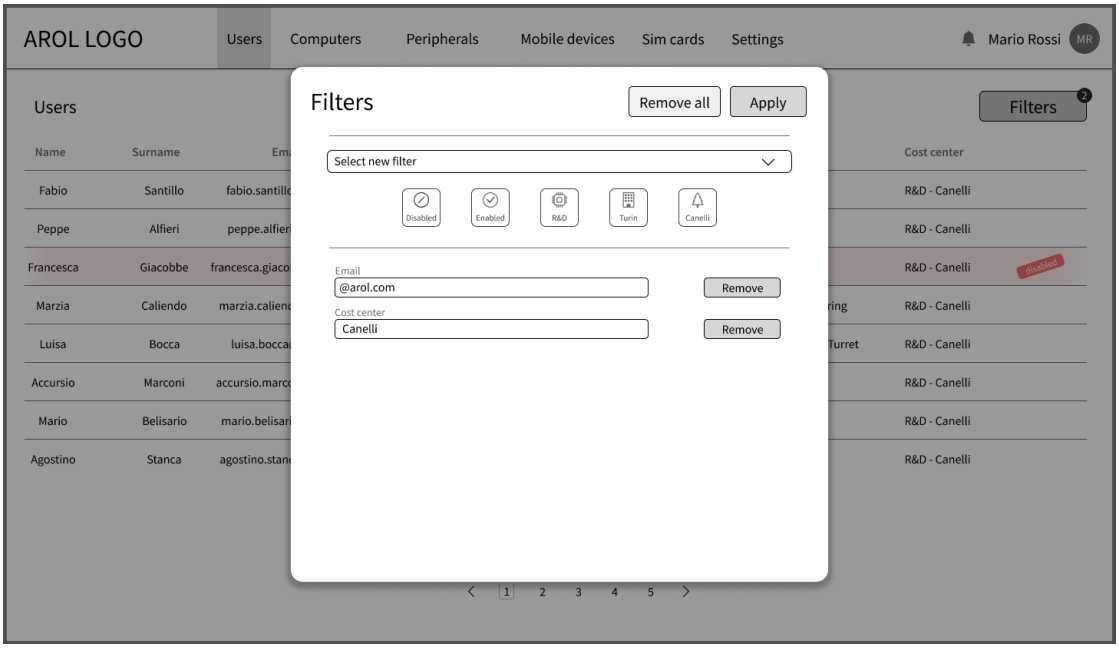

Figure 4.4. Users page, filters, AROL IT portal, prototype

the same pattern as the table rows, maintaining a consistent visual style across pages.

The second, middle section focuses on the relationships between the selected entity and other related entities. For example, on the user page, this section displays information about the devices associated with the user. This design choice aims to help IT employees manage entities with minimal effort. By providing all relevant information about a user, along with their associated devices, on a single page, they can avoid unnecessary navigation between pages. This approach improves workflow efficiency and speeds up data retrieval operations.

Data in this section has been grouped into categories to provide IT users with a clearer and more structured interface. A row of tabs allows users to switch between categories with a simple click. This approach was positively evaluated because it is consistent with the previous interface, which already employed the *Tabbed View* design pattern.

Users are also provided with a set of elements to facilitate data visualization, including a search bar, a sort dropdown menu, and a select button. The latter permit users to choose their preferred display mode, switching between a card view and a simple list view. These two modes are designed to support different workflows, differentiating user interaction based on individual preferences.

From these categorized elements, users can now perform different operations with a single click, such as returning a device assigned to a user.

The final section of the page is dedicated to the entity's history. This section contains

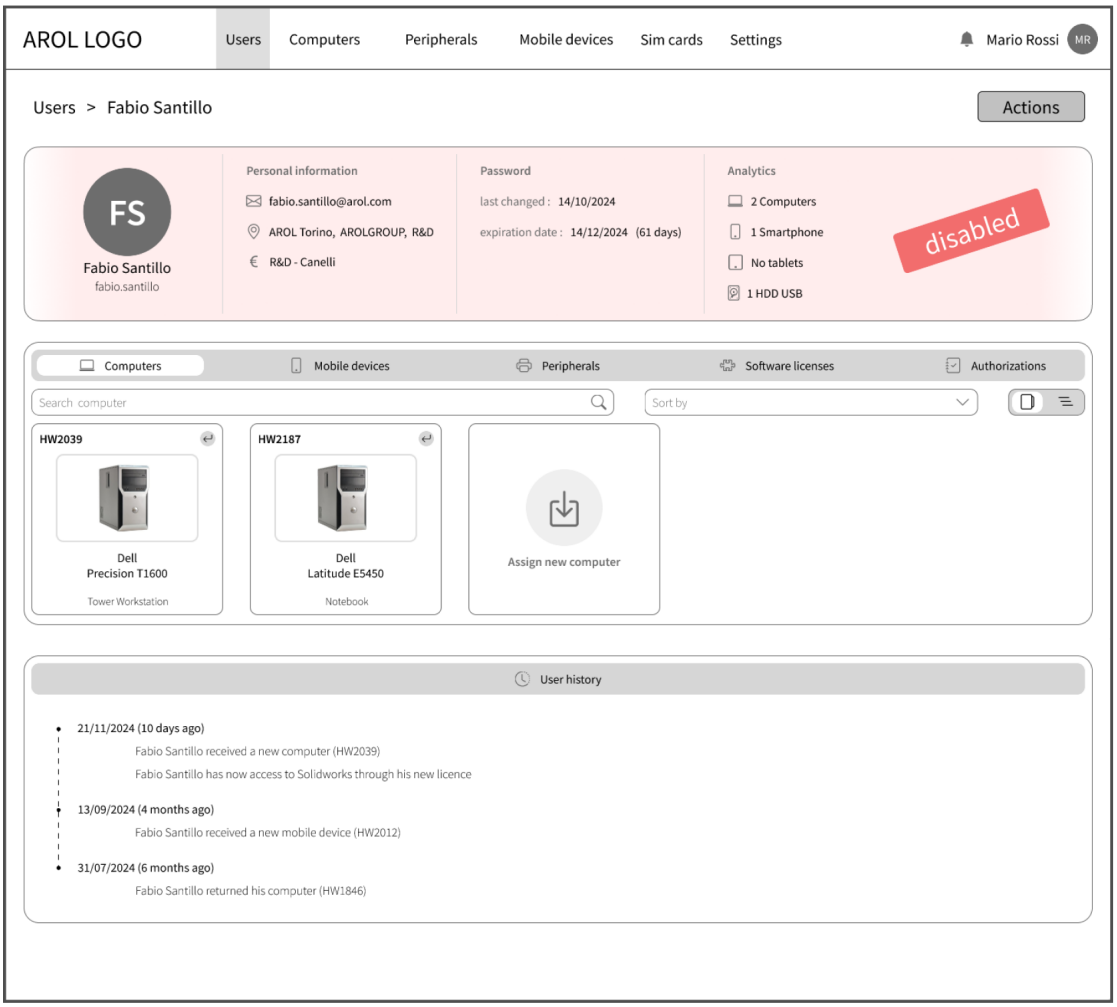

Figure 4.5. User page, AROL IT portal, prototype

information about past events related to the entity, such as assignments, revocations, and modifications. Each event is displayed along with its date and a brief description of what occurred.

This redesign addresses a significant issue in the previous interface: the fragmentation of information. Previously, events were displayed within the relationships themselves, meaning that the only way to view them was by navigating to specific pages, each dedicated to a different type of event (e.g., peripheral assignments, license revocations), and searching for the relevant data within a table. Essentially, there was no way to quickly understand the history of an entity. In contrast, the new design provides a straightforward overview of an entity's lifecycle, reducing effort and improving usability.

The last design improvement is the *Actions* button in the top right corner of the page. It has been designed as a convenient way to perform operations on the entity itself by presenting a small menu where action options are displayed. With a simple click, without requiring users to recall any information or perform any manual operations, the selected action is carried out.

For the sake of simplicity, the design of the other entities' pages will not be explained in this thesis, as they follow the same structure, exploiting the same components and design choices. This consistency also helps users by simplifying the familiarization process with the new application's interface.

Settings page

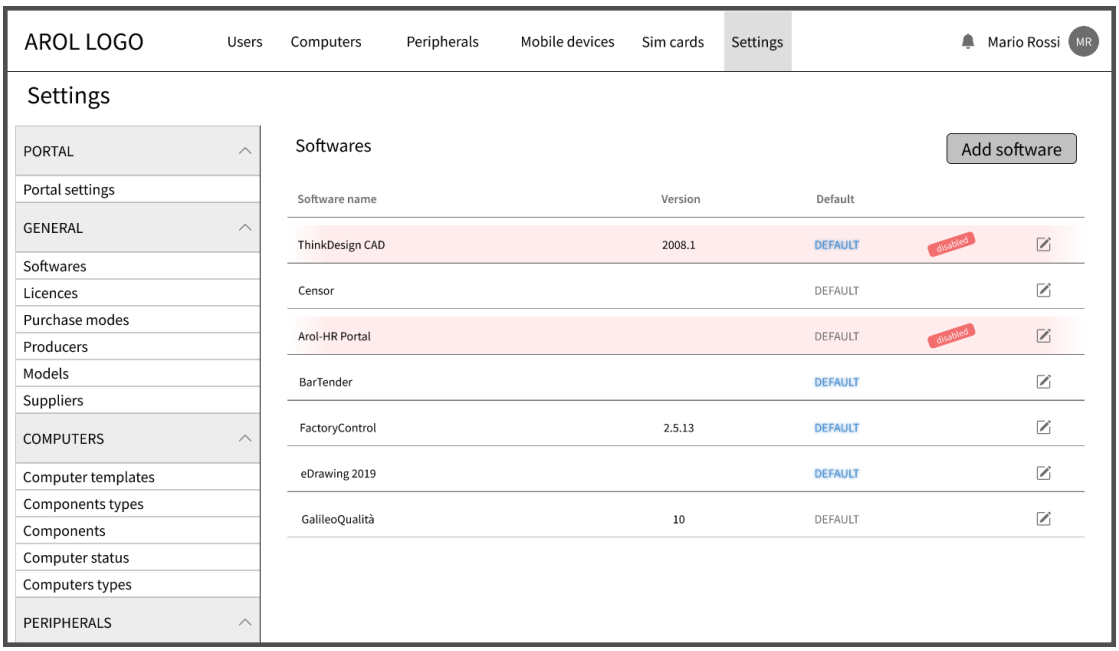

Figure 4.6. Settings page, AROL IT portal, prototype

The goal of the *Settings page* (Figure 4.6) is to provide a centralized location for all elements of the portal that are not main entities. Since these elements are not accessed daily, they do not require a leading position in the design architecture. However, they still need to be available for viewing and modification, just like any other entity in AROL's system.

The page is structured into two horizontally arranged sections.

The leftmost section is dedicated to a hierarchical dropdown menu that lists all available entities, allowing users to navigate through them easily. The decision to implement a hierarchical menu comes from the overall design approach of the application, which organizes portal elements into distinct categories or types. Applying this structure to the settings page ensures that secondary entities are categorized, helping users quickly locate the desired item without the need to scroll through a long and hard to visualize list.

By selecting one of the options from the menu, the corrispondent list appears in the section on the right. Also in this case, a consistent design has been adopted, by exploiting a table as mean to display all the entities that belong to the selected category. From the same page, users have the possibility to add a new instance of that entity or edit a specific one with the dedicated buttons. In each case, a small popup appears to easily complete the desired operation without navigating to a dedicated page, since often these entities require few information that can fit inside a popup (Figure 4.7).

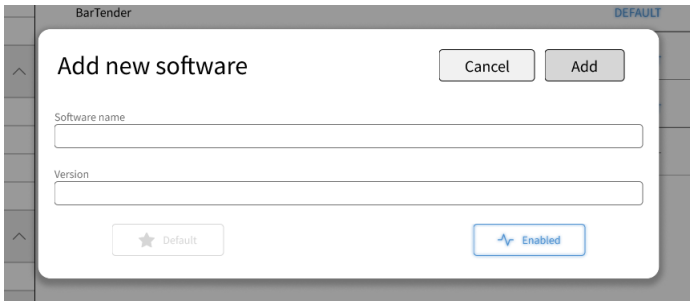

Figure 4.7. Settings page, add popup, AROL IT portal, prototype

QR code scan page

From the homepage, users can access the application's QR code scanner page by clicking the dedicated button.

This feature represents the most significant and impacting improvement in the new interface. Even if not technically complex, it marks a substantial step forward in reducing the IT users' workload. By exploiting QR codes, users can access product information and perform essential operations without manually searching for the desired device.

The page (Figure 4.8) has a simple design, with a central element, the device's camera, used to capture QR codes. This element is displayed in a clear way to emphasize its primary role and purpose. Additionally, it has been designed with a square shape, intuitively guiding users to align QR codes, which are typically square, within the capture area.

At the bottom of the screen, users can find a list of recently scanned products. This feature allows users to scan products using a smartphone and later, from their PC, manage those products without needing physical access or rescanning them.

Moreover, three buttons for batch scanning are placed at the top of the page, providing quick access to the corresponding pages.

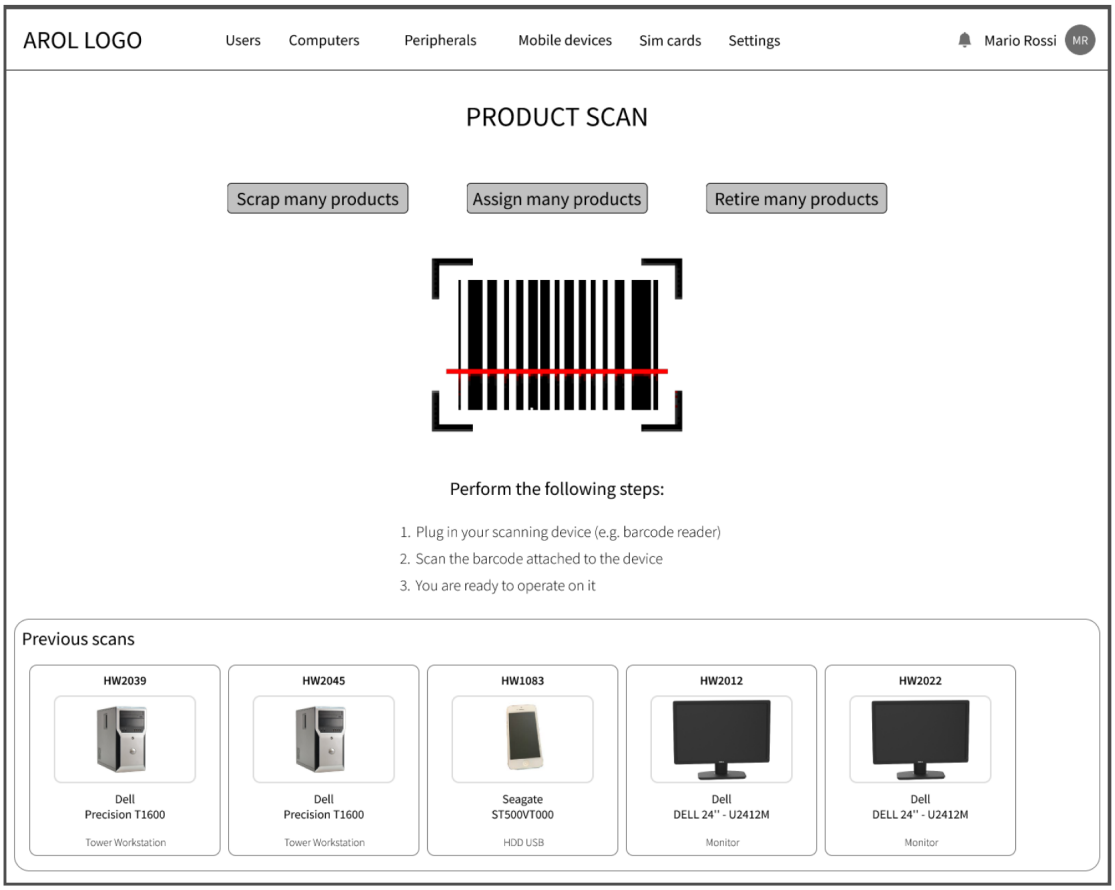

Figure 4.8. QR code scan page, AROL IT portal, prototype

When a QR code is detected by the device's camera, a popup automatically appears, displaying key identity information such as the *HW code* and the product model. Along with this information, the popup presents a list of buttons corresponding to the main operations users can perform on the scanned device. These actions include scrapping the device, assigning it to a user, or simply opening its detail page for further information (Figure 4.9).

QR code batch scan page

The batch scan page (Figure 4.10), accessible via dedicated buttons on the scan page, addresses one of the low-priority functional requirements: the ability to perform the same operation, often complex and tedious, on a group of products, even if they are heterogeneous.

For example, IT users may need to scrap twenty devices (computers, peripherals, and

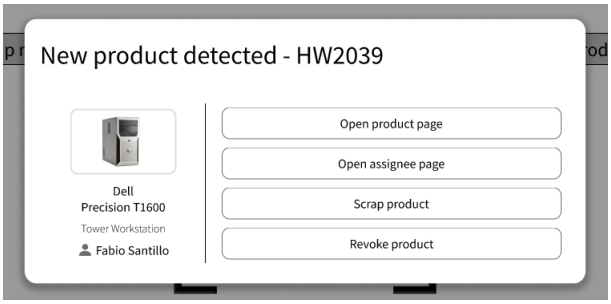

Figure 4.9. QR code scan page, product popup, AROL IT portal, prototype

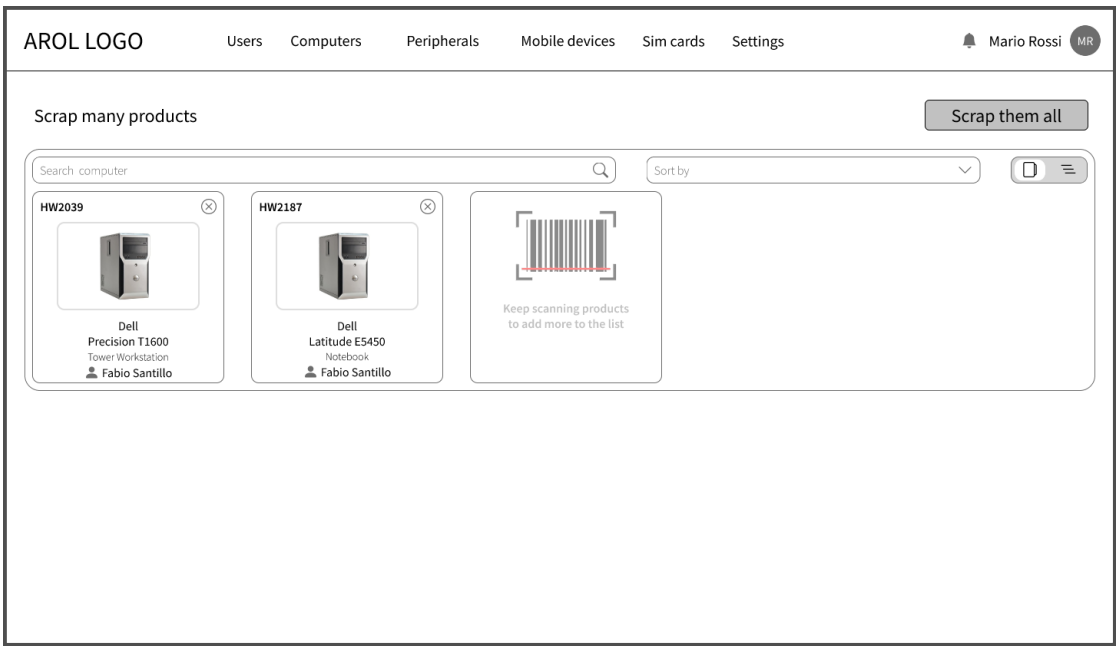

Figure 4.10. QR code batch scan page, AROL IT portal, prototype

smartphones) that were returned by AROL employees. This section of the portal allows them to scan each product quickly from the same screen and, with a single click, apply the operation in batch to all selected devices. This significantly reduces the time and effort required for IT employees, streamlining their workflow.

Chapter 5

Implementation

This chapter describes the process of transforming the design prototype from the previous phase into a real, functional and working application. It highlights the challenges and difficulties met during development, both technological and organizational.

The first section focuses on the software tools used to build the new interface, explaining why each was chosen over its competitors. Then, the description of the implementation phase is covered, highlighting the main components that have been used to build the frontend. Finally, a brief overview of the final application is presented, highlighting its main pages and features.

5.1 Technological Stack

The first step before starting the coding phase, which effectively builds the application, is to carefully analyze and evaluate which software tools will be used for each part of the portal. Since the application consists of multiple sections, heterogeneous in nature and supporting different functionalities, various tools such as libraries, frameworks, and other technologies must be selected based on their specific purposes.

This choice is not trivial, as the long-term stability of the application depends on the availability and support of the companies or developers behind these tools. Selecting a library that becomes deprecated or dismissed just six months after its adoption can significantly impact the application's operation. This would require extensive effort to resolve the issue, such as migrating to a different tool, which implies considerable costs both in terms of workload and financial resources. This situation contributes to what is known as *Technical Debt*: future work required to address problems caused by improper decisions taken during development.

At this point, the software tools adopted are presented, along with the reasons behind these choices.

5.1.1 Visual Studio Code and Bitbucket

The first tool that needs to be chosen is the code editor. This is where the programmers spends most of their time during the implementation phase, so the choice must be made carefully.

One of the most well-known, supported, and trusted editors is Visual Studio Code [20], developed by Microsoft. The choice for the AROL IT portal project fell on this tool for several reasons. First of all, it is widely used by many developers across different contexts, meaning there is an active community that reports bugs and supports its continuous evolution. Additionally, it is highly customizable, allowing users to download and install specific extensions that enhance the base software with additional functionalities. These can help with code visualization, early error detection, and various other features.

For example, in this thesis project, one of the most useful extensions was the Angular Language Service [3], which provides intelligent support by offering completion lists, diagnostic messages, and quick shortcuts.

Finally, since this software is much more than a simple code editor, Visual Studio Code (often abbreviated as VS Code) also supports code versioning through an intuitive graphical user interface (GUI). This interface displays real-time changes in the code and allows users to commit modifications to their repository efficiently.

Speaking of repositories, an essential part of the technology stack is the codebase, where different versions of the application are stored. This is typically cloud-based, allowing multiple stakeholders, including developers, to access the code, track changes, and review audit information.

For this project, Bitbucket [7] was chosen as the version control system. This was the only tool in the technology stack that AROL required to use, as it was already adopted for other projects. Being part of the Atlassian ecosystem, also used for task management with Jira and Confluence, it ensured seamless integration with the company's existing workflow. Additionally, Bitbucket operates with a workflow based on Git [15], a widely recognized standard for version control.

5.1.2 Angular

Since the goal of the project is to develop a frontend interface for a web application, the selection of HTML and CSS is standard and universally adopted. However, using them in their plain form is not scalable for a large-scale project. Additionally, while JavaScript can enhance functionality, developing a complex interface solely with HTML, CSS, and JavaScript is neither practical nor efficient.

For this reason, it is quite common in projects of this kind to use a frontend framework. The main options to choose from are *React*[1] [27], *Angular* [1], and *Vue* [29]. These tools enable developers to manage state and handle the component lifecycle automatically, without dealing with low-level details that would divert attention from more important aspects, such as meeting the application's requirements.

[1] Although React is included in this group, it is technically a library rather than a framework. The term "framework" is used here for simplicity and broader understanding

Regarding this choice, AROL management does not impose the use of any particular framework. However, an important detail should be considered. In fact, both the previous version of the IT portal and other internally developed software were built using the Angular framework. For this reason, the best option is to maintain the same technology, if possible, to facilitate the work of future engineers who will have to maintain the software. If they already have expertise in this field, making modifications to the application will be easier and faster.

Additionally, this choice also allowed the candidate to improve his knowledge of a technology that had never been used in previous projects, gaining new skills and expertise that could be useful in many other scenarios.

5.1.3 PrimeNG, PrimeFlex and PrimeIcons

Once the frontend framework was selected, the next step was to evaluate the possibility of adopting a frontend library to integrate predefined components. Using such libraries allows developers to reuse already developed and tested elements, saving time and resources during the programming phase.

At this stage, several options were available. For instance, *NGX Bootstrap* [21], an Angular-adapted extension of the well-known *Bootstrap* library, offers a robust set of UI components. Similarly, *Angular Material* [4], developed by Google, provides interface elements characterized by a minimal design.

However, the final choice fell on *PrimeNG* [26] for several reasons. First, this library includes a significantly larger number of predefined components, which is particularly useful in applications with multiple areas and sections serving diverse purposes. Additionally, PrimeNG offers components that are missing in the other options. For example, its tabular items support complex operations automatically, simplifying the programmer's work.

Alongside PrimeNG, the project also adopted another library from the same company: *PrimeFlex* [24]. This is a utility library that facilitates the combination of PrimeNG UI components (but not only) with ready-to-use classes for automatic and responsive management of grid systems, spacing, typography, and other key aspects of web interface programming. For instance, it allows developers to define a component's width relative to its parent component, dynamically adapting to the screen size of the user's device with a simple class name.

Lastly, a mandatory choice must be made for what concerns the icons set. Every application need many icons in buttons, popups and menus. For this thesis project, the choice was quite straightforward because *Primefaces*, the same producer of the previous libraries, also developed a collection of icons that have been designed together with UI components. For this reason, adopting a style that comes from the same company is always a good choice in term of coherence and cohesion. The adopted library is called *PrimeIcons* [25].

5.1.4 @zxing/ngx-scanner

An essential aspect of the application is the management of QR codes.

In particular, the scanning functionality is crucial for the success of the new version of the AROL IT Portal, as it supports various tasks that rely on it to be functionally effective. Given its complexity, this feature could not be entirely developed from scratch, as it involves multiple components, interacts with hardware, and requires image recognition capabilities.

For this purpose, the *ngx-scanner* library [22] was selected as the best option. Several factors contributed to this choice. Firstly, it is an open-source library, meaning AROL does not need to purchase any license. It is based on *ZXing*, one of the most reliable libraries for QR code and barcode scanning. Additionally, it was specifically developed for Angular environments, eliminating the need for wrappers or artificial adjustments. Another significant advantage is that this library supports multiple barcode formats, such as Code 128, Code 39, and UPC-A, which AROL may adopt in future versions. Having these formats readily available without requiring additional add-ons translates into significant time savings.

5.1.5 Chart.js

The final technological component of the new application is the *Chart.js* library [8]. As its name suggests, it has been used in the AROL IT Portal to generate graphs displayed on the homepage of the frontend.

Although PrimeNG already provides components for bar charts, line charts, pie charts, and more, the decision to use Chart.js was driven by the fact that PrimeNG charts are built on top of Chart.js. This means that installing Chart.js would have been necessary regardless. By adopting the standalone Chart.js library, the application benefits from a lighter version, as only the essential features are included, which is sufficient given that diagram visualization is not a primary focus of the IT portal.

Finally, if future enhancements place a stronger emphasis on analytics and statistics, this choice will offer greater event control and improved performance.

The following table (5.1) summarizes all the technological choices made fro this thesis project

Name	**Version**	**Scope**
Atlassian Bitbucket	N/A	Repository and code versioning
Angular	18.2.0	Fronted framework
Chart.js	4.4.7	Graph library
Microsoft Visual Studio Code	1.97.2	Code editor
Primefaces PrimeFlex	3.3.1	Frontend utility library
Primefaces PrimeIcons	7.0.0	Frontend icons library

Continues on the next page...

Name	Version	Scope
Primefaces PrimeNG	17.18.13	Frontend UI library
Zxing Ngx-scanner	19.0.0	QR code scanner

Table 5.1: Technological Stack of the new AROL IT portal.

5.2 Frontend Development

Once the main technologies have been selected, the development phase can begin. This is the most technical part of the project, but it is important to note that many crucial decisions must still be made during this stage. Often, these choices directly impact the software's performance from a technical standpoint. For example, an inefficient solution might lead to excessive loading times, or an incorrect route configuration could prevent users from accessing certain areas of the portal.

The following subsections will cover the key aspects of the development phase, outlining the challenges encountered and the corresponding solutions implemented.

5.2.1 Project setup

Before starting to code, it is necessary to configure the development environment. For this purpose, the *Angular Command Line Interface* (CLI) [2] has been used. This choice is standard within the Angular ecosystem, as its CLI is a command-line tool that allows programmers to scaffold, develop, test, deploy, and maintain applications directly from a command shell. This is particularly useful for performing complex operations that involve multiple components, minimizing the risk of errors or inconsistencies.

After installation, the CLI enabled the creation of a new Angular project from scratch and the addition of new components and services with simple commands:

```
npm install -g @angular/cli              ## to install Angular CLI globally
ng new AROL-ITportal                     ## to create a new Angular project
ng generate component breadcrumb         ## to create a new component
ng generate service computers            ## to create a new service
ng serve --port 4200                     ## to start the development server
```

5.2.2 Project structure

Once the project has been correctly generated, it is important to decide how to organize folders and their corresponding files. This is essential for producing a solid and scalable product, ensuring a structured approach to coding. Although this choice may seem less relevant in the context of this thesis project, it is actually crucial, as it saves time when searching for specific files and makes it easier for future developers to understand the project's structure and apply modifications without needing to read the entire code base

first.

As Figure 5.1 suggests, apart from the standard Angular project structure automatically generated by the Angular CLI with the command *ng new AROL-ITportal*, the decisions made during this phase led to the creation of four main folders, each containing files related to logically cohesive areas.

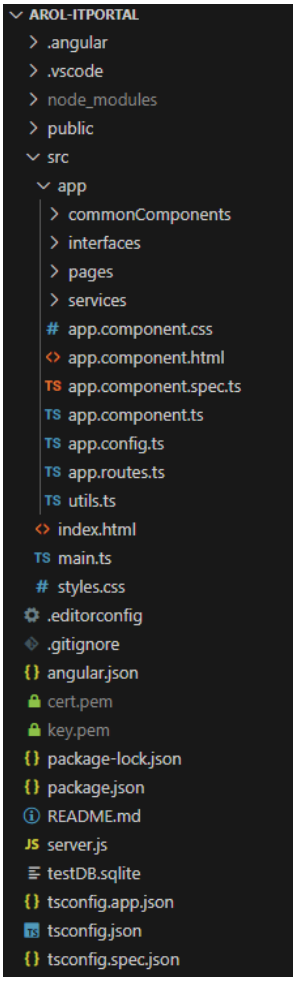

Figure 5.1. Visual Studio Code - Project's structure

The first folder created is *app/pages*, which contains all the screens of the new frontend. Specifically, each child element of this folder is a subdirectory that includes four files related to the same page. These four files, automatically generated by the command *ng generate component pages/pageName*, correspond to: the template file, where the *HTML* structure is defined; the *CSS* file, which contains the styling parameters that customize the appearance of the components; the *Typescript* file, which manages the business logic

of the page; and a test file.

The second essential folder for a well-structured development phase is *app / commonComponents*. This directory is designed to include all the elements of the portal that may be reused multiple times across different parts of the application. These components serve as the building blocks of the frontend. For example, it contains the breadcrumb component, which is used on all pages where nested navigation is required. Each element in this folder follows the same structure as those in *app/pages*, as they are generated using the same command.

Along with the creation of visual components, the application also needs a way to communicate with a backend server. This is where the third folder, *app/services*, comes into play. It is responsible for managing all the files required by the frontend to perform structured REST requests to the database. This directory contains subfolders, one for each logical entity (e.g., companies, computers, peripherals, etc.), each including the corresponding REST request methods. This approach will also be beneficial in the future for developers who need to adapt the backend of the AROL IT Portal to align with the requirements of the new frontend, as these files consolidate all the necessary requests and their formats in a structured manner.

The last folder in this section is *app/interfaces*. It has been designed to store files that define the interfaces used in the portal's frontend. This folder serves as a catalog of variable types, where each interface consists of multiple heterogeneous fields. For example, in the AROL IT Portal, the *User* interface represents an employee, containing fields such as name, surname, email, department, and other relevant information. This directory is essential to ensure consistency across the entire application, guaranteeing that all components use the same data types. Similar to the previous folder, this structure will also help future developers easily identify and work with predefined types and formats.

5.2.3 Routing and navigation

Another important aspect of the development phase, when coding the application's screens, is their logical placement and the relationships between them. Although the navigation behavior has already been described in Chapter 4 while introducing the new design system (Figure 4.1), defining the routes that represent the application's pages remains a crucial topic.

Routes should adhere to specific guidelines and standards for Uniform Resource Identifiers (URI). For instance, best practices recommend using plural nouns to indicate resource containers and avoiding paths longer than four segments. These conventions contribute to a more intuitive and maintainable navigation structure.

The following table (Table 5.2) lists all routes of the new frontend along with a brief description.

Route Name	Description
`/login`	Login page, showed before user is logged

Continues on the next page...

Route Name	Description
/	Homepage of the portal
/users	Contains users table
/users/:id	Specific user's page, showing personal info
/computers	Page dedicated to the set of computers
/computers/:id	Page for a specific computer
/peripherals	Page that lists all the peripherals
/peripherals/:id	Page related to a specific peripheral
/mobiledevices	Shows the list of mobile devices in the system
/mobiledevices/:id	Page dedicated to a specific mobile device
/simcards	Page showing the table of sim cards
/simcards/:id	Page related to a specific sim card
/settings	Page that allow users to manage IT portal's settings
/scanner	Page that hosts the QR code scanner
/scanner/batch	Allow users to perform multiple operations at once
/not-found	Default page for general routing errors

Table 5.2: Routes of AROL IT Portal's new frontend

5.2.4 Key decisions and best practices

During the development phase, programmers can adopt small tricks to enhance technical performance, independently of user interaction, saving both computational resources and development time. The goal is to minimize the number of backend requests, reduce unnecessary computations, and anticipate future work for developers. In the new frontend, three main techniques have been implemented.

First, one of the fundamental techniques in web programming is known as *lazy loading*. As the term suggests, this method is applied in scenarios where data retrieval is a critical operation. In such cases, programmers can opt to load data incrementally, rather than all at once. This approach is beneficial because typically only a limited amount of information fits on a single screen, making it inefficient to request and load all available data at once. In the new version of the AROL IT Portal, this technique has been applied to all tables, which, by definition, contain large amounts of information. By implementing lazy loading, only the necessary portion of data is downloaded from the server at any given time, while users can access additional data by interacting with the paginator.

Another technique that optimizes backend requests is *optimistic update*. This approach predicts the response from the database, improving the user experience by reducing waiting times. For instance, when IT users update information about a computer, they must fill out a form and submit the data. Normally, they would have to wait for the server's response before seeing the updated information in the user interface. With optimistic updates, the interface is immediately modified using locally computed data that simulates the expected response, without requiring the server to send back large amounts of information. In this project, optimistic updates have been implemented for entity editing and device return operations.

The last key method adopted during development is *modularity*. While not strictly a programming technique or algorithm, this is an important organizational choice. It involves structuring the interface by breaking it down into reusable components and generalizing them as much as possible through parameterization. This approach is particularly useful in web applications, where visual elements are often repeated across multiple pages or sections. By creating dedicated components that encapsulate structure, styling, and business logic, developers can avoid redundant code and significantly streamline the development process.

5.2.5 Interface and style

The last topic discussed in this section concerns the management of the application's styling. Although it may seem secondary or superficial, as it mainly deals with colors, borders, alignment, spacing, and other visual aspects, proper style management is crucial for ensuring scalability and modularity. Defining styles in a centralized manner allows them to be declared once and then consistently applied throughout the application. This approach simplifies future stylistic modifications, making it easier for developers to update the portal's appearance without manually adjusting multiple components. Additionally, maintaining styling in a single location improves maintainability by making specific style rules easier to locate and modify.

For the new version of the AROL IT Portal, a dedicated file, *src/styles.css*, has been adopted to contain the majority of the style definition classes. The term majority is used to emphasize that this file is not the sole source of styling, as certain components required their own unique style definitions, distinct within the entire application. For this reason, only those specific components follow the approach of internal style definition. In contrast, for general styling applied to most UI elements, this centralized file has been selected.

The file follows a structured organization, where classes related to the same visual aspects are grouped together. For example, the ones defining background colors are placed consecutively (e.g., bg-disabled, bg-transparent, bg-blue-light).

A crucial decision in the styling process was the definition of the application's color palette. Instead of applying colors directly within styling classes, which are generally scattered throughout the code, a better approach was to define color codes (e.g., `#2e7d32`)

as variables in the root class (Figure 5.2). This allows the use of variable names to reference colors, enabling changes at any time by simply updating the root class, ensuring that the entire application seamlessly adopts the new style guide.

```
:root {
  --primary-blue:      #00256a;
  --secondary-orange:  #ff9f1c;
  --background-light:  #e0e4eb;
  --surface-light:     #f4f6f8;
  --text-dark:         #1c1c1e;
  --success-green:     #2e7d32;
  --warning-yellow:    #ffb020;
  --error-red:         #d32f2f;
  --info-blue:         #1976d2;
  --focus-blue:        #0041a3;
  --disabled-grey:     #a0aab6;
  --blue-light:        #d6deef;
  --blue-dark:         #001a4a;
  --error-dark:        #b22222;
}
```

Figure 5.2. Visual Studio Code - Color palette's definition

5.3 Application Overview

After many hours of coding, aimed at replicating the Figma design previously created, the new version of the AROL IT Portal is now complete. This final section of the Implementation chapter presents the result of the work done, the version that will be effectively used by AROL IT employees in the future. It is important to note that while the original design has been faithfully followed, minor improvements and adjustments were made during the coding phase. However, these modifications did not alter the overall architecture of the portal.

The first screen (Figure 5.3) corresponds to the login page, which was missing in the previous version of the portal. By adding it, users now have a more structured and robust login experience, replacing the simple browser popup with a dedicated login component.

After logging in, the user is presented with the homepage (Figure 5.4), which provides a general overview of the application. It features charts displaying the classification of entities into subgroups such as device types, companies, and formats, along with sections that report devices information and user requests. These sections have been designed and placed on the homepage because they require the IT employee's primary attention upon opening the portal each morning. For example, the list on the left is intended to collect all devices that need to be scrapped as soon as possible.

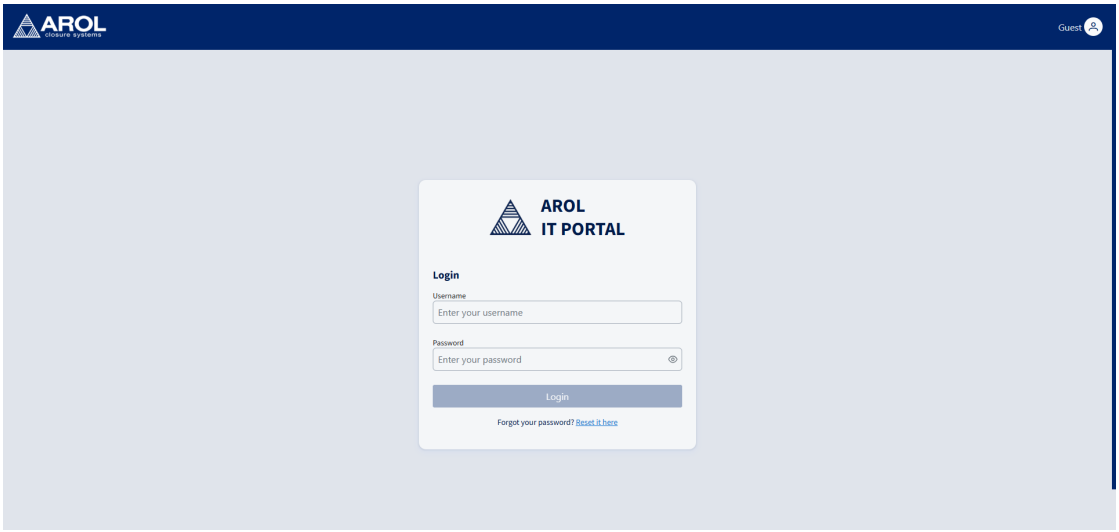

Figure 5.3. Login page - New AROL IT Portal's frontend

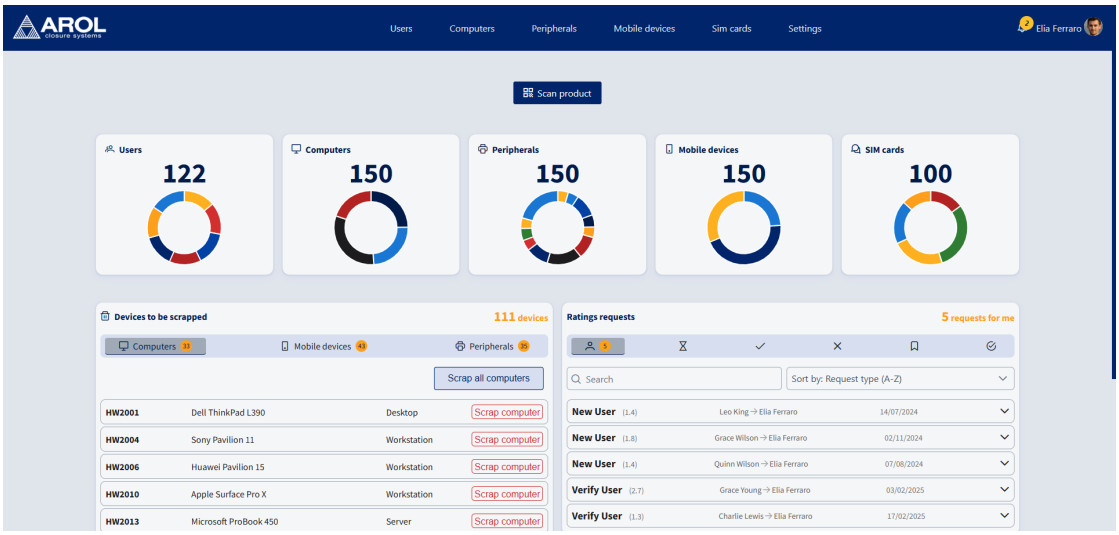

Figure 5.4. Homepage - New AROL IT Portal's frontend

By clicking on one of the five cards on the homepage, IT staff are redirected to the corresponding screen, which displays a table with data about the selected entity (Figure 5.5). From this page, filtering and sorting table rows are simple and intuitive operations

that help IT employees retrieve information easily.

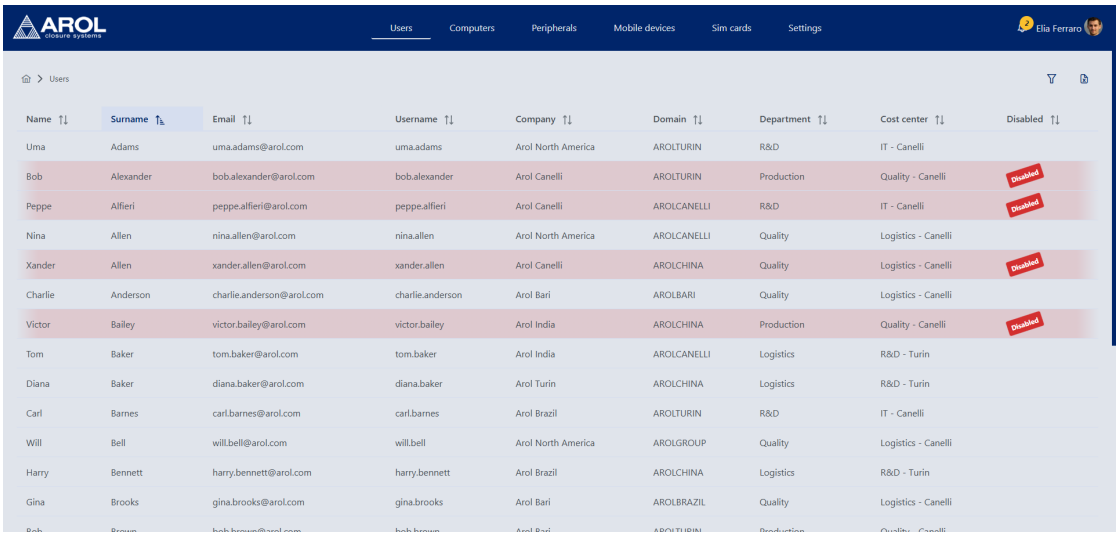

Figure 5.5. Users page - New AROL IT Portal's frontend

If IT users want to access information about a specific person and edit them, they can then click on a row of the table and enter the user's page (Figure 5.6).

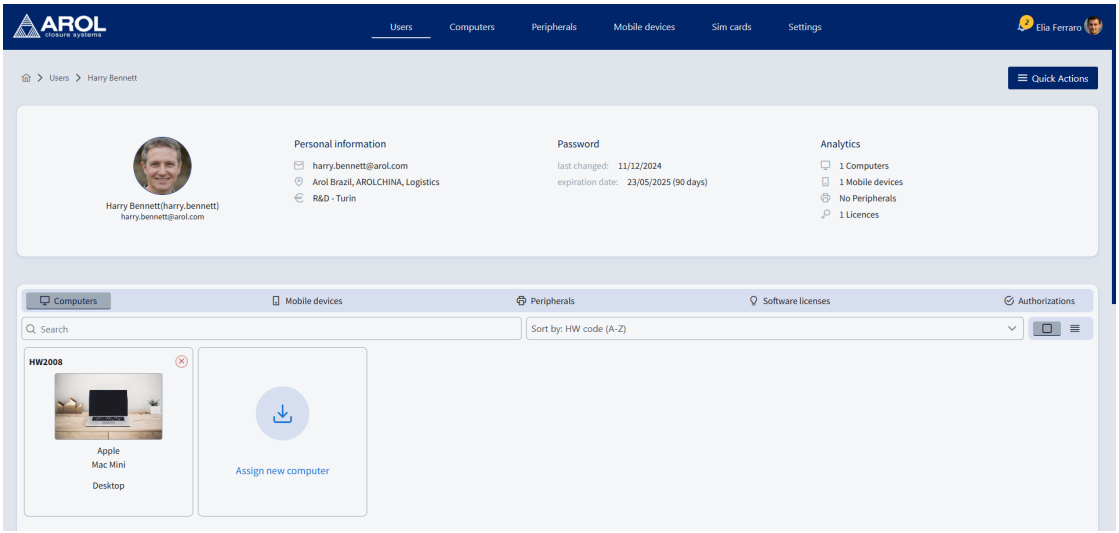

Figure 5.6. User page - New AROL IT Portal's frontend

Another important page is the one related to the scanner (Figure 5.7), shown in its simple and single version, where QR codes can be scanned to perform operations quickly and efficiently.

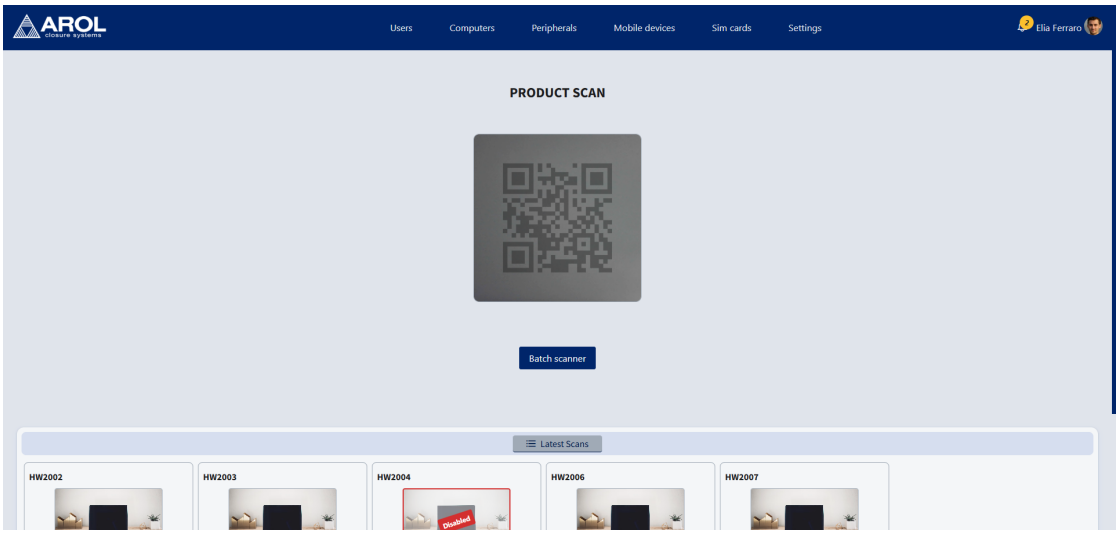

Figure 5.7. Scanner page - New AROL IT Portal's frontend

The last screenshot (Figure 5.8) in this chapter showcases the same users page previously presented, but in its mobile version, optimized for smaller, taller, and narrower screens such as smartphones and tablets. In this version, the table is omitted to prevent screen overcrowding, which would negatively affect usability. Instead, a simple search bar is displayed, allowing users to find a specific user based on their information.

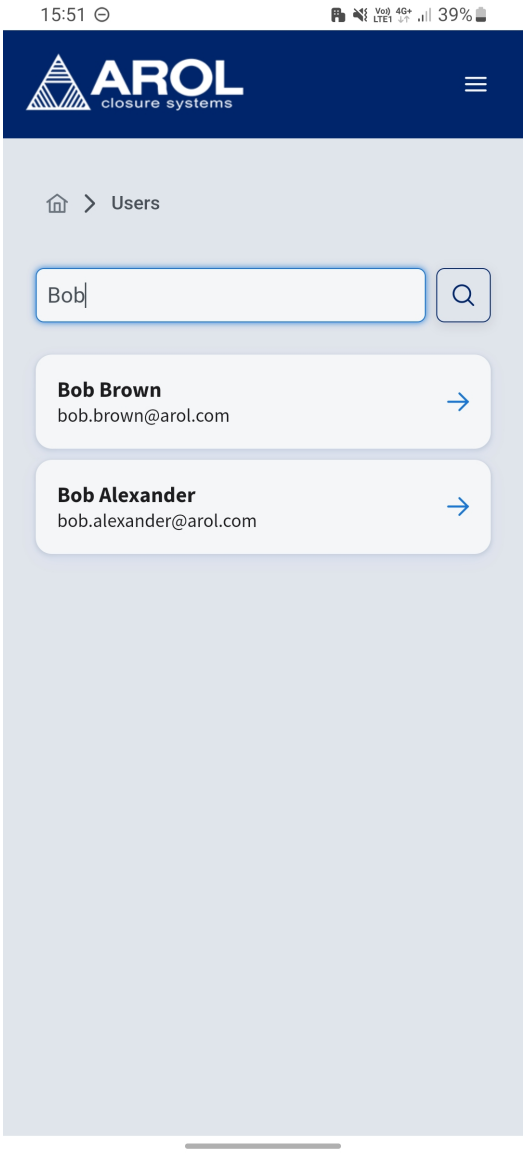

Figure 5.8. Users page (mobile version) - New AROL IT Portal's frontend

Chapter 6

Verification and Validation

After the last line of code has been written, the project cannot be considered complete without a formal verification process to assess whether the results meet the expected goals and to what extent. This chapter, *Validation and Verification*, aims to provide a final evaluation of the project. It also serves to identify areas where the new solution is strong and robust, as well as areas that could benefit from future improvements.

Both verification and validation [14] are essential as they focus on different aspects of software testing, analyzing various scopes and extracting different types of results. Typically, verification is performed first, followed by validation, due to their distinct purposes. The next sections will explain their objectives and methodologies in detail.

6.1 Verification

The first step of this phase is verification. It is the initial operation, both logically and chronologically, performed to evaluate a system, answering the question: *"Are we building the product right?"*. Verification focuses on ensuring that the application meets the design requirements from a static perspective by reviewing documents, code, and design. Often, it is considered part of the *static testing* category because it only deals with static documents and files, such as requirement files, code files, and design projects.

For example, if an application lacks a critical section where users need to perform an essential task, the verification phase can detect this issue. As a result, developers can address the problem and make the necessary corrections to the system.

Often, this phase includes the creation of various types of tests, written in a programming language. These tests are generally classified into *Unit Tests*, *Integration Tests*, and *End-to-End Tests*.

Unit tests focus on testing individual components of the system, where a unit can be defined differently depending on the context, for example, a function, a class, a module, or even a larger element. These tests establish specific rules that determine how a unit should behave: given certain input values, it must return the expected output values.

Integration tests, on the other hand, evaluate multiple units together. This approach is

useful because it verifies not only the logic of each individual unit but also their interactions and how they exchange data. For instance, a test could validate the behavior of multiple functions working in sequence or several interconnected classes.

Finally, end-to-end tests simulate the system from a user's perspective, assessing the entire workflow. This is the most comprehensive testing methodology, as it encompasses all components and interactions, ensuring that any errors affecting the user experience are identified.

Since these tests are written in code, they can be executed repeatedly, allowing for systematic verification after every update. This makes them a highly formal and reliable approach to software verification, both during development and in future iterations.

However, for this thesis project, none of the previously mentioned tests were chosen or implemented during the verification phase. This decision stemmed from the fact that the collaboration's primary focus was not on testing methodologies or pure code production but rather on process optimization and user-centered design. Consequently, less time was allocated to this phase, requiring the team to optimize resources and prioritize other aspects of the project.

On the other hand, completely skipping the verification process was not an option, as it would have left the application in an unclear state, with imprecise results and no formal analysis of the implemented solution. Since the purpose of verification is to ensure that all requirements have been met, the chosen methodology consisted of analyzing the functional and non-functional requirements, highlighting how each of them was addressed (if applicable).

Although this approach is not highly scalable or easily replicable across multiple software versions, it provides a solid level of formal analysis. Moreover, it serves as a valuable resource for future developers, offering a structured document that outlines how the requirements were fulfilled, facilitating future modifications and improvements.

Below, the verification analysis is divided into subsections, distinguishing between functional and non-functional requirements.

6.1.1 Functional requirements verification

The following table (Table 6.1) presents the functional requirements of the new AROL IT Portal interface, as defined in the *Requirements Analysis* chapter. For each requirement, a brief explanation of the corresponding implemented solution is provided.

FRN	Name	Solution's description
1	Display user's devices	In the user page, the central box displays the set of devices assigned to the selected person, with a toggle option to switch between large cards and compact rows. This feature enhances usability by allowing for quick and efficient information retrieval.
2	Display user asset summary	In the first box of the user page, alongside personal data, there is a small analytics section that summarizes the devices currently assigned to the user. This section dynamically updates its values whenever a new device is assigned or an existing one is returned.
3	Display AD group descriptions	In the authorization tab of the user page, instead of displaying plain class names, toggleable areas provide brief descriptions retrieved from the database server, offering a clearer understanding of the meaning of each class name.
4	Display device location	In the device page, when a device is assigned, the department of the assigned user is also displayed.
5	View all unassigned devices	When IT employees want to assign a device, a popup appears, displaying a list of all available devices that are still unassigned, allowing them to select the appropriate one. This helps prevent any user errors.
6	Automatic HW codes when adding devices	When creating a new device and IT employees open the creation popup, the HW code field is automatically filled with an available code generated by the server. This eliminates the need for them to manually select or input the correct code, simplifying the process. The field is still editable.
7	View remaining software licenses	When IT users open the popup to assign a software license to an employee, the list of available licenses is enhanced with the number of licenses still available for each software. This helps the IT user to easily identify when licenses are running low, ensuring better management and timely actions.

Continues on the next page...

FRN	Name	Solution's description
8	Navigate between device and assignee	From the user page, the click on a device card redirects to the device's page, while in the device page, if assigned, a small section indicating the assigned user is displayed. A click on it would navigate to the user page.
9	Add SIM tariffs with a click	From the SIM page, a popup is presented with the list of available tariff options. IT users simply have to choose from the list, facilitated by searching and sorting components, preventing manual errors.
10	Assign a device with a click	From the user page or directly from the device page, a popup is shown with the list of available devices or users. IT users can quickly select a device or a user, aided by sorting and filtering options to avoid mistakes.
11	Assign a license with a click	From the user page, a popup appears with the list of available software licenses. IT users can select the appropriate license easily, with filtering and sorting features to assist in the process.
12	Return a product with a click	From the user page or directly from the device page, IT employees can simply click the *"x"* button to return a product. A confirm modal is also presented to avoid unvoluntary actions.
13	Add devices from templates	From the devices page, a split button opens a popup to create multiple devices at once, based on a predefined template. IT users only need to fill in a few fields related to the company and purchase options, speeding up the process significantly.
14	Scan QR code to access product page	IT users can access the scan page, where a camera box is presented in the middle of the screen. By simply framing a QR code on a device, a popup is automatically shown. There *Open product page* button can be clicked to access device information.

Continues on the next page...

FRN	Name	Solution's description
15	View entity event history	On each page displaying a specific entity, a timeline that collects events related to that entity, organized by date, is presented. For example, a computer being assigned to a user. By hovering over an event, additional details about who performed the action are displayed. On smaller screens, the hover effect is replaced with a small card that directly shows this information.
16	Generate device QR codes	By accessing the *Quick actions* menu, IT users can generate and display the QR code associated with a specific device on the screen.
17	QR code action menu	IT users can access the scan page, where a camera box is presented in the middle of the screen. By simply framing a QR code on a device, a popup is automatically shown, summarizing the main information of the device and providing buttons for quick actions. This significantly speeds up the process of retrieving information and performing management operations.
18	Batch actions via QR scan	IT users can access the batch scanner page, where they can scan multiple QR codes in succession by simply framing them one after the other. After scanning, users can select an action to perform for each device scanned, significantly speeding up operations that typically require a lot of time and are performed on a large quantity of devices.

Table 6.1: Verification of Functional Requirements

Important to note is that these requirements and their solutions have been manually tested both during implementation, while coding, and before the product presentation to the AROL management. These informal tests were performed by simulating users' interactions while completing normal daily tasks.

6.1.2 Non-functional requirements verification

Moving to the non-functional requirements verification step, the following table (Table 6.2) summarizes the quality features that the new interface should have achieved during implementation, along with a short explanation about the decisions behind them.

NFRN	Name	Solution's description
1	Usability	The interface has been implemented following standard design guidelines, adopting recognizable icons and supporting them with labels whenever possible. Basic tasks have been supported by self-explanatory buttons, with key buttons highlighted to catch users' attention.
2	Responsiveness	Pages that display tables adjust the number of columns based on the screen's width. On very small screens, such as smartphones, the table disappears, making room for an intuitive search bar that allows for fast entity retrieval. Complex operations have been removed from mobile versions as they do not align with AROL's mobile workflow.
3	Consistency	Components that repeat across multiple pages maintain the same design and style, making them easier to recognize and remember. The same colors have been used for similar elements to convey their relationships.
4	Visual Standards	The interface follows a modern, minimal layout with flat colors. Recognizable icons have been used to represent actions like closing, saving, and downloading. Color coding is employed to convey information such as success, errors, and disabled states.
5	Error Management	A toast system has been employed to provide effective and immediate communication between the system and the user. In this case as well, colors offer an immediate indication of success or error.
6	Data Handling	Data is managed with an effective design. Tables support large amounts of data, while lists and cards support different visual layouts.
7	Data Loading	Large amounts of data, such as those managed by tables, have been implemented with pagination. This saves resources and ensures that only the required data are effectively downloaded from the server.

Continues on the next page...

NFRN	Name	Solution's description
8	Caching	Previously scanned devices are saved to save resources for future access. On specific entity pages, loaded data is not destroyed between multiple accesses to the same section, avoiding multiple reloads of the same data.
9	Modular Architecture	A specific directory (*app/commonComponents*) has been created specifically to contain components that can be reused across several pages. This ensures modularity.
10	Design Patterns	Services, categorized by logical areas, have been implemented to communicate with the backend server. Best practices and standards have been adopted during coding to facilitate future programmers. Interfaces collect data types in a single directory for fast access and compatibility.

Table 6.2: Verification of Non-Functional Requirements

6.2 Validation

The second step of this chapter is *Validation*. It focuses on the question: *"Are we building the right product?"*. Its goal is to determine whether the developed application meets stakeholders' expectations and, therefore, delivers value. It is also categorized as *dynamic testing* because it involves actively using the application, evaluating it in action in the field, and observing real users interact with it.

In the context of this thesis project, the validation step has been performed through usability tests [17]. By adopting this methodology, a researcher, also called a facilitator or moderator, impersonated in this project by the candidate, asks participants to perform tasks. While they interact with the interface, the researcher observes their behavior and records various data, including both objective, quantitative metrics and subjective, qualitative impressions.

Usability tests are a valuable resource, providing the opportunity to identify issues of any nature, discover areas for improvement, and gain insights into users' behavior and preferences.

For this project, five target users were selected, including IT employees who will use the new frontend as well as some external participants. The inclusion of external users was particularly useful in assessing whether the interface is intuitive and easy to learn for those unfamiliar with the current version.

After exploring the new interface for three minutes, the five users were asked by the

facilitator to complete eigth tasks of varying difficulty, covering all major aspects of the application. For each task, a timer was used to measure completion time, and the number of mouse clicks was recorded. In addition, any errors, whether critical or obstructive to users, were also recorded.

Usability tests often follow the *thinking-out-loud* methodology, in which participants verbalize their thoughts during interactions. However, in this project, this approach was not adopted because time was a key parameter, and verbalizing thoughts while performing tasks could have affected the measurements. Instead, after completing each task, users were invited to share their thoughts, highlighting areas where the interface could be improved.

The following table (Table 6.3) summarizes the five tests by presenting the average results obtained. This is useful as it provides a general overview of the status of the new frontend, along with insights into potential improvements.

N	Task command	Clicks	T op	Err	Notes
1	Tell me how many devices have been assigned to Bob Anderson	7.4	43.4s	0	The user analytics panel is not very visible. Some participants counted the number of cards in the bottom section instead. A good solution, but not the optimal one.
2	Assign a MacBook to Fabio Santillo	9.2	38.8s	1	One user did not notice the pagination component below the table and found an alternative way to access the MacBook page and assign it. This cannot be considered a complete success.
3	Tell me which department the peripheral with code HW4045 is registered in	7.8	60.4s	0	Some initial difficulties with the filtering system due to the new interface, but everything went smoothly afterward.
4	Tell me the total number of devices assigned during Mr. Anderson's career at AROL	15	75.4s	0	One participant gave the wrong answer, but the approach was correct, so it can be considered a partial success.

Continues on the next page...

N	Task command	Clicks	T op	Err	Notes
5	Tell me which components are mounted on this computer (with a QR code attached)	3.6	41.2s	0	One participant initially struggled with the popup after scanning, but no one had trouble finding the answer.
6	Return all of Fabio Santillo's computers	14.2	47.6s	0	Some users performed this operation by accessing the pages of the assigned devices to return them individually instead of returning them from Fabio Santillo's page, which took more time. However, the task was still completed correctly.
7	Create twenty mobile devices with the same features as those described in the "Basic iPhone" template	49.6	63.2s	2	Two users could not complete the task on their own because they misinterpreted the creation popup and ignored the purchase area. This indicates that further improvements should be made to its interface.
8	Assign these five computers to Fabio Santillo as quickly as possible by scanning their QR codes	14	100.8s	0	No one had any issues with this task.

Table 6.3: Validation tests results on the new frontend

In order to have a quantitative comparison between the old and the new versions, the same usability tests have been performed on the current AROL IT Portal's frontend. These (Table 6.4) are the results obtained in the same format as above

N	Task command	Clicks	T op	Err	Notes
1	Tell me how many devices have been assigned to Bob Anderson	5.2	53.5s	0	Devices that are currently assigned and those that have already been returned are in the same table, some users were confused, taking more time to provide the answer.
2	Assign a MacBook to Fabio Santillo	10.8	77.3s	0	No problem on this task.
3	Tell me which department the peripheral with code HW4045 is registered in	9.1	62.7s	0	Device information are not in any hierarchical order, users need to scan the entire page to retrieve the information.
4	Tell me the total number of devices assigned during Mr. Anderson's career at AROL	10.6	50.8s	0	No problem on this task.
5	Tell me which components are mounted on this computer (with a QR code attached)	9.7	67.3s	0	Slow procedure, starting form the serial number on the computer, retrieving the corresponding HW code and then the components.
6	Return all of Fabio Santillo's computers	23.9	106.1s	0	The only way to perform this task is by accessing pages of the devices and manually delete associations, one by one. Being in different pages, it took more time to complete.
7	Create twenty mobile devices with the same features as those described in the "Basic iPhone" template	~700	~40min	0	Boring and error-prone repetition of manual operations. Unfeasible for an efficient company.

Continues on the next page...

N	Task command	Clicks	T op	Err	Notes
8	Assign these five computers to Fabio Santillo as quickly as possible by scanning their QR codes	~70	~10min	0	Even in this case multiple pages must be accessed, requiring a lot of time.

Table 6.4: Validation tests results on the current frontend

Two important considerations must be made regarding this second table, which presents the results of the usability tests conducted on the current version of the AROL IT Portal.

The first concerns tasks five and eight. These tasks implicitly encourage participants to use QR codes on the devices to complete the operations quickly. However, the current frontend does not support code scanning. As a result, users were forced to find alternative methods to complete the tasks, inevitably taking much more time.

Similarly, task seven relies on templates that are not available in the current version. In this case, participants had to create devices manually, one by one, in order to achieve the same result and complete the task.

Lastly, a clarification regarding tasks seven and eight: since these tasks required users to repeat the same operations in a loop multiple times, and given that they were designed to test functionalities available only in the new interface, the tests were truncated after the first iteration. The number of clicks and the time spent were then extrapolated by multiplying individual values. This does not pose an issue, as the goal is simply to provide a general estimate, which is significantly higher than in the first table.

6.3 Frontend Evaluation

This final section aims to summarize the results obtained from the previous tests, attempting to derive a conclusive evaluation of the work done. This is valuable because test results consist of precise numbers tied to specific tasks chosen for the evaluation. However, a meaningful assessment should provide an interpretation of these data, communicating and conveying a general trend that can be applied to the entire application and, more broadly, to the overall work carried out in this thesis project.

In particular, the objective is to determine to what extent the new version of the AROL IT Portal is an improvement (if it is) over the previous version currently used by AROL IT employees. This evaluation is also intended for AROL management, as it provides data to support decision-making and helps assess how IT productivity may change with the adoption of the new interface.

6.3.1 Quantitative results analysis

The first examination is quantitative, as it considers the results obtained from the usability tests conducted on both the current interface and the new one.

One of the most interesting parameters to evaluate is the third column in both tables, which represents the average number of clicks participants needed to complete each task. To provide a clearer and more comprehensive view, a grouped bar chart has been created (Figure 6.1). The horizontal axis represents the different tasks, from the first to the eighth, while each task is visualized with two vertical bars: the blue bar corresponds to the task performed on the old interface, and the orange bar represents the same task executed on the new interface. The bars indicate the number of clicks required for each task in the different interfaces.

It is important to note that the vertical axis uses a logarithmic scale to better represent the data, given the significant variation in values.

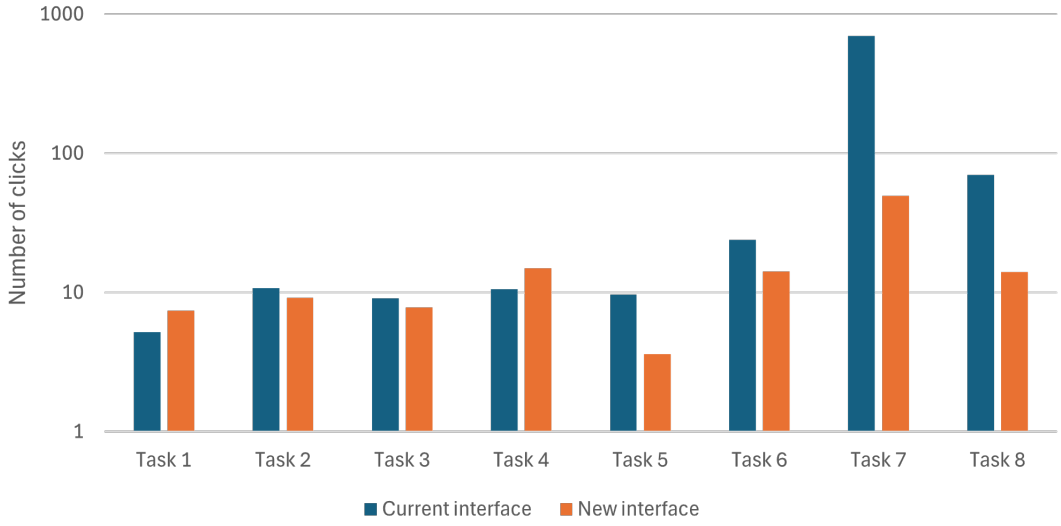

Figure 6.1. Frontend evaluation - Number of clicks per task chart

As shown in the graph, the new interface generally requires fewer clicks, meaning fewer interactions, to achieve the users' goals. This is particularly evident in the fifth, seventh, and eighth tasks, as they were specifically designed to test functionalities available only in the new interface. Since the old application lacked these features, users had to find alternative ways to complete the tasks, demonstrating the significant advantage of the new interface.

Despite this, even the first tasks showed an improvement in efficiency, suggesting that the new interface is faster and more intuitive overall.

However, tasks one and four are the only ones where performance worsened with the new interface. This is primarily due to users' unfamiliarity with the updated design and the fact that currently assigned and previously assigned devices were moved from a single table to separate sections. While this change initially required users to adjust, it ultimately promotes better organization in the long run.

The second parameter worth considering is the time required by participants to complete their tasks. While time itself is a useful metric, it has been transformed to highlight another aspect related to efficiency. In fact, each time value has been converted into the corresponding task completion rate per minute, using the following formula:

$$\text{Task}_i \text{ rate per minute} = \frac{60}{T_i} \tag{6.1}$$

This transformation allows the graph (Figure 6.2) to better highlight not just the time required for each task, but the overall productivity of the IT department when performing different tasks. Each bar represents the number of completed tasks per minute, providing a clearer picture of efficiency improvements.

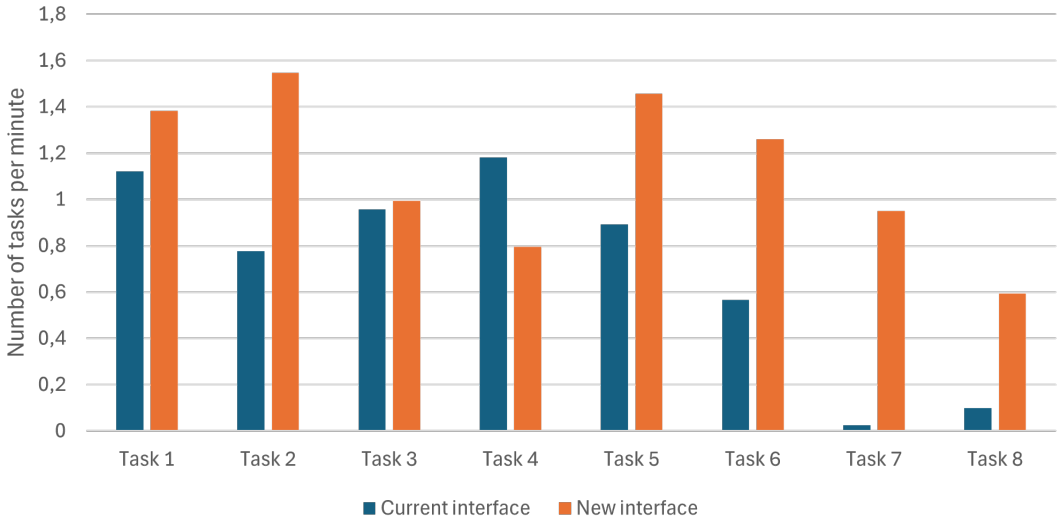

Figure 6.2. Frontend evaluation - Task rate per minute chart

The chart clearly demonstrates, with a significant gap, that the new interface performs considerably better, enabling IT employees to be much more productive in almost every task of the usability test. The only exception was the fourth task, which performed worse than in the old interface. This was because the task required users to gather information about all mobile devices assigned to a specific user throughout their career at AROL. In the previous version of the AROL IT Portal, this data was presented in a large table containing both currently assigned and previously assigned devices. However, in the new interface, this information was split into two separate sections: a panel for currently assigned devices and a comprehensive timeline showing all previously assigned products

for each user. As a result, this specific task was performed more efficiently in the old interface. However, this decline occurred in a task that is rarely performed, whereas daily operations, such as assigning and returning devices, were significantly faster in the new interface.

From the previous chart, the improvements made over the old interface are quite evident, but to better estimate the degree of enhancement, another step forward can be taken.

In fact, the last chart (Figure 6.3) aims to visualize the percentage increase in productivity rate from the current version of the AROL IT Portal to the new one. For each task, the following formula has been applied to calculate the improvement values.

$$\text{Task}_i \text{ productivity improvement percentage} = \frac{\text{Task}_i \text{ rate}_{new} - \text{Task}_i \text{ rate}_{old}}{\text{Task}_i \text{ rate }_{new}} \tag{6.2}$$

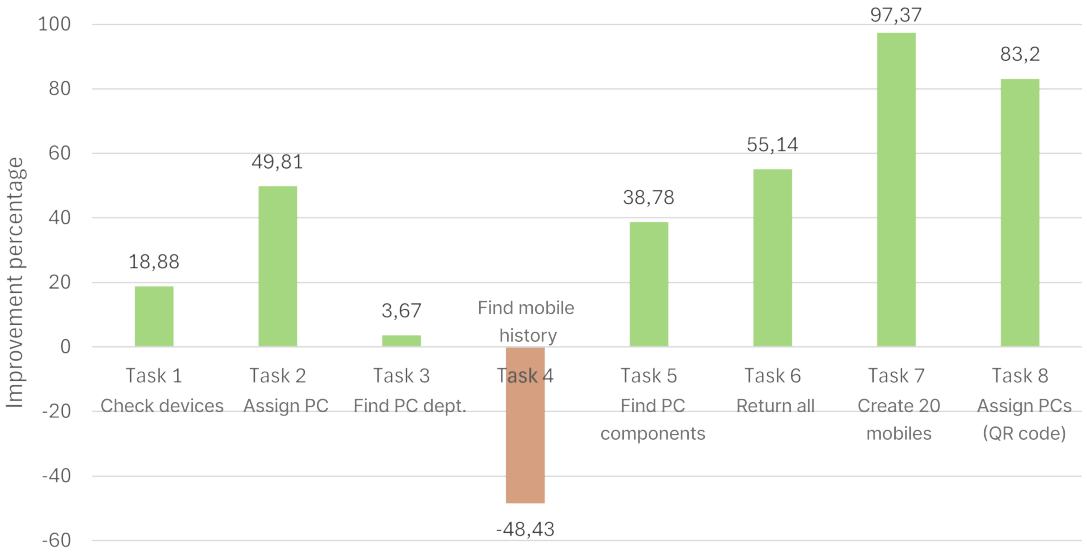

Figure 6.3. Frontend evaluation - Productivity improvement in percentage chart

Apart from task number four, the chart shows an overall productivity improvement of a significant amount, especially for the last tasks, which are more complex and challenging.

By computing the average of the previous values, the new interface proves to be 59.84% more productive than the current one. From AROL management's point of view, this means that by adopting the new interface, the IT department can save a substantial

amount of time. For example, if IT employees normally work eight hours per day, the new interface allows them to perform the same number of tasks as they would in approximately thirteen working hours per day if they continued using the current interface.

This clearly indicates that AROL is saving a significant amount of resources, which can be allocated to other areas, such as technology innovation or the development of additional internal solutions.

6.3.2 Qualitative results analysis

The second type of evaluation, essential to cover all key aspects of system analysis, is a qualitative overview of the new interface. Unlike the previous section, this analysis does not rely on numbers, formulas, or charts. Instead, it focuses on stakeholders' feedback, as their insights are crucial in assessing the system's effectiveness beyond measurable metrics.

The goal of this subsection is to assess their level of satisfaction and overall perception of the new frontend, capturing their thoughts, impressions, and suggestions for potential future improvements.

The overall perception and sentiment regarding the new interface are highly positive. This is evident not only from the usability test results, where completion times were significantly lower compared to the old interface, but also from participants' feedback and remarks after each task. Many expressed their appreciation for the visual appeal of the new design, which they found more uniform and aesthetically pleasing. Moreover, after using it, they highlighted the clarity in information presentation and the ease of use as key advantages.

On the management side, AROL's leadership is extremely satisfied with the new design, recognizing its potential in optimizing processes through a user-centered approach.

However, this does not mean that the new interface is perfect or that the work is entirely complete. Some participants pointed out areas for improvement that could be addressed in future versions.

For instance, several users struggled with the device creation popup, particularly with the vertical column containing tab buttons, as it did not clearly indicate that all forms within the tabs needed to be filled before proceeding.

Additionally, while the filtering system was considered efficient, some users found it too complex for quick and imprecise searches. A possible improvement could be the introduction of a simple search bar to complement the existing filtering system and provide a more intuitive search experience.

Chapter 7

Results and Discussion

This chapter focuses on providing an overview of the work done, aiming to take stock of the project. After completing the practical part of this thesis project, summarizing the team's efforts is very important to understand which goals have been achieved, both in technological and organizational terms, how they have been reached, and which difficulties have been encountered throughout the project.

This is particularly useful for future projects, which can learn from past experiences by identifying aspects that brought value and expertise, trying to replicate them, or using them as a foundation to support new challenges. At the same time, even areas where major problems arose are valuable in this analysis. The more challenging parts of the project can be leveraged to learn from them, addressing and correcting issues while understanding their causes to avoid making the same mistakes in the future and gaining valuable experience.

7.1 Achievements

The most important and evident consideration on the project consists of the achievements that the thesis brought.

The primary achievement, and the most valuable, is the frontend itself. This is an achievement in two senses. Firstly, it is a useful product for the AROL company, which they can put into production to take advantage of its new functionalities and innovative design, ultimately improving IT department productivity. On the other hand, it also serves as valuable experience gained from the process of developing a robust and scalable interface.

The second achievement concerns the set of documentation produced during the thesis project. In fact, for each step of the software development process (current portal analysis, requirements analysis, design phase, etc.), several formal documents have been drafted as a result of the working process that characterized that specific phase. Although these documents were initially considered mandatory steps, merely useful for the final production

of the new frontend, they actually hold greater value in a broader sense.

For example, the analysis of the current frontend can be useful in multiple ways. It can serve as a reference to identify past design mistakes that should be avoided in future projects. Additionally, it can be used as a methodological reference for analyzing other internally developed applications to evaluate them and explore possible improvements.

Similarly, the requirements document is useful for AROL management, as it provides a general understanding of the product through a brief review rather than requiring them to actively use the application to comprehend its functionalities. It can also evolve in future versions if different approaches are considered, building upon the current functionalities listed in this document.

The Figma project, too, can be leveraged for future interface updates without needing to redesign every section from scratch, allowing development to begin from a predefined and already functional design framework.

Finally, the usability testing evaluation serves as a starting point for discussions on setting new goals for future releases. It highlights the strengths and weaknesses of the new interface based on user experiences. The positive aspects can serve as a reference for decisions that proved successful, while the negative aspects provide valuable insights into areas that need improvement to better meet user needs.

A more specific yet significant achievement comes from the way backend server calls have been structured during the development phase of the new AROL IT Portal interface. Since they have been categorized into services that manage different logical entities, with each service containing a dedicated component for backend call functions, retrieving them is straightforward. Each function clearly displays the necessary backend route and, most importantly, the input and output parameters of each call.

This will be extremely useful in future phases, particularly when integrating the new frontend with the existing backend server. The backend will inevitably need updates to align with the new requirements. Having the entire set of backend calls organized in a single place makes them easily accessible, significantly facilitating the integration process.

7.2 Challenges

Equally important is the evaluation and reporting of the difficulties encountered during the development of the project. These are fundamental not only for recognizing the sources of these challenges and understanding how to avoid similar issues in future projects and processes but also for pairing the problems with their corresponding solutions. By doing so, we can extract problem-solving techniques applied in each situation, with the goal of adapting and reproducing them, even in different forms, when similar issues arise in other scenarios.

The first significant challenge that surfaced as soon as the project began, unrelated to any technical or technological issues, was the fact that the thesis was set within an already defined and consolidated context, represented by AROL. The company already had its established habits, processes, and workflows, all developed over years of expertise.

For the candidate, this meant immersing in the company's context and trying to understand and internalize their workflow. Although this might seem straightforward, it was not. Several factors contributed to this, such as the specific terminology the company used to describe processes, which differed from the candidate's background, or the fact that many processes involve other departments, different from the IT one, increasing the complexity and obscurity levels.

At the same time, AROL's long-established processes revealed that many of them were not entirely correct or efficient, even before the implementation of the new frontend. For example, during the analysis of the current interface, inconsistencies between the data presentation on the interface and the backend database emerged. This necessitated additional effort to resolve these issues.

One of the main technological challenges was implementing the responsiveness of the entire web application. This was particularly challenging because, in addition to ensuring the interface adapted to different screen sizes, an inherently difficult task, the application also had to adjust its functionalities based on the working conditions of IT employees in different situations. Employees perform various tasks depending on their environment and goals. For instance, when in the office at their desktop, certain activities are performed, while when mobile and using their smartphones, different tasks are accomplished.

Finally, managing large amounts of data posed another challenge. Not only did the data have to fit the screen, but its presentation played a crucial role in how the software was evaluated by users. If the design failed to meet their needs or support their tasks effectively, users would quickly abandon the application. To address this, best practices in web development, such as lazy loading and optimistic updates, were implemented.

7.3 Lessons Learned

Lastly, it is important to reflect on what the candidate, as well as others, learned from this thesis project. "Others" refers to the fact that both the company and the final reader of this thesis can gain insights from the work done, extracting key considerations and improvements, and, more generally, learning from the errors made throughout the project.

First and foremost, many soft skills and transferable abilities were developed. For instance, the ability to adapt to different contexts and work with diverse people from various backgrounds required the integration of different working procedures and workflows.

In this thesis context, for example, although AROL did not mandate the use of any particular technology, Angular as the frontend framework was the best choice, as described in the Implementation chapter. This choice required the candidate to learn a new technology, a process that presented a series of challenges and inevitably slowed down the overall progress. However, this became a significant exercise in adapting to new scenarios, where the time spent learning new concepts was just as valuable, if not more, than simply developing additional lines of code.

Problem-solving was another critical skill sharpened during this project. It involved

multiple steps, each focusing on different aspects but all aimed at the same goal. Each step brought inherent challenges and problems that needed to be addressed as efficiently as possible, trying to strike a balance between all factors in order to stay on course without being too rigid or overly attached to previous decisions.

A more technical and specific lesson learned was the adoption of rapid prototyping in a real-world context. This approach provided real value and time savings for both the candidate and the company. It allowed AROL to get a general and immediate sense of the final result without waiting for the actual implementation of the interface. Additionally, the frequent meetings allowed AROL to provide feedback on the prototype, enabling adjustments to be made before the coding phase. These meetings involved not only AROL management but also the final users, IT employees, who provided invaluable insights. This collaborative process significantly accelerated the project by capturing both the pros and cons of the design early on, helping to align it more closely with user expectations.

Finally, the thesis' central goal, process optimization through a user-centered approach, emerged as the most valuable lesson to carry forward. It redefined the priorities for any future projects.

When a software engineering team develops an application, like the one in this thesis, it is easy to become overwhelmed by numerous competing objectives. For example, improvements to the design system, a redesign of the database schema, or optimizing asynchronous functions may take up the majority of the team's time, as they try to incorporate the latest, most innovative technologies. While this focus is valid, since technological efficiency can influence the user experience and ultimately benefit the company, it is important to remember the ultimate goal of the project.

The primary aim of software development is to ensure that users can perform their tasks better and/or faster. This is why the development team should prioritize understanding the users' workflows and optimizing their processes before implementing a new interface or system.

Involving end-users in the development process has been crucial to this thesis project. It placed users at the heart of the development process, surpassing any technological choices or tight schedules, and emphasized the importance of a user-centered approach in software development.

Chapter 8

Conclusion and Future Work

The final chapter of this thesis presents the concluding considerations on the work done, offering a broader perspective. It aims to summarize the goals achieved, retrace the steps taken throughout the project, and briefly analyze the obtained results. Additionally, this section will provide an overview of the strengths and limitations of the newly developed frontend.

Finally, potential next steps and areas worth exploring for future improvements will be outlined. These insights serve as a valuable starting point for both AROL and the candidate, guiding the development of additional features and further enhancements to the system, ultimately benefiting both parties.

8.1 Final Considerations

The thesis began with a single but crucial goal: to optimize IT department processes through the redesign of a new frontend for the AROL IT Portal by means of a user-centered approach. As already explained in the previous chapter, this thesis did not focus on the technical implementation of a new system, identified solely by the new user interface, but rather took a much broader approach, prioritizing the final users, who were involved in every phase of the project.

With them, the starting point was the analysis of the current situation, the state of the art regarding the old portal, focusing on usability issues and interaction problems, supported by Jakob Nielsen's ten design principles.

Based on the results obtained, multiple interviews with the final users and AROL management were conducted to draft the requirements document. This was a fundamental phase, as it allowed the entire team to gain a better understanding of user needs and how a new frontend could support their daily tasks.

With a clear idea of the functional and non-functional requirements expected by AROL, multiple cycles of rapid prototyping were necessary to develop a final low/medium-fidelity design prototype of the new interface. By adopting Figma as a design tool, the IT department had, in each cycle, the opportunity to visualize the final result, enabling them

to provide feedback and suggestions.

The most technical part, the central one, concerned the actual implementation of the proposed solution from the previous step in a modular and structured way, following web development best practices. This phase also addressed responsiveness and workflow adaptation to different devices.

Finally, an evaluation of the work was conducted through usability tests, assessing the interface's behavior in real-world scenarios by submitting eight tasks to selected users to gather insights into their interactions. This phase classified the new frontend as a significant improvement in terms of productivity, increasing efficiency by more than 50%, meaning a time saving that could be allocated to other tasks. This was made possible by reducing the number of required clicks to complete a task and minimizing the time users needed to interact with the system.

Since a broader approach was taken, the proposed solution is not only a valuable product for AROL and its IT department but also represents a scalable and replicable process optimization methodology that can be applied to many other scenarios, even those entirely different from AROL's context. By definition, it ensures that users remain at the center of attention, rather than solely focusing on technology or innovation.

However, the proposed solution, the new AROL IT Portal's frontend, has some limitations. For example, it has not been linked to or tested with the real backend. For this reason, some database queries or data types may need to be updated in the future. Additionally, it has not been tested with real data and values but rather with predefined and simulated data. This could pose a problem, as actual data may differ from the expected ones.

For this reason, the following section evaluates possible improvements and future developments for the new frontend.

8.2 Future Work

When discussing projects of this kind, they can never truly be considered finished, as there is always something that can be improved, and new functionalities can always be implemented.

The first natural continuation of the project is undoubtedly the deployment of the new solution. This involves integration with the existing backend by aligning the desired data structures and formats. It will also require the creation of new routes and data storage on the server, as new entities such as templates, histories, and QR codes have been added from scratch.

Another significant improvement would be the introduction of formal verification tests to enhance overall code quality. These could include unit, integration, or end-to-end tests, or ideally a combination of them. This would also aid in refining or correcting requirements, as specific values and metrics could be extracted from the tests.

Regarding future implementations and enhancements, one particularly useful feature concerns user creation. At present, users cannot be created or edited within the portal, as they are managed by an external software application from which the portal simply retrieves corresponding user information. A potential improvement would be the ability to create users directly within the portal, triggering an automated procedure to perform controlled operations on those external tools. This feature would unify and consolidate the entire ecosystem, eliminating the need for manual and independent management across multiple systems and software applications.

At that point, IT employees would be able to fully manage the entire user lifecycle from the portal, covering everything from user creation to deactivation, including permission assignments and other user settings.

The final idea for future development relates to the fact that the application is used by people worldwide, each speaking different languages (English, Italian, Chinese, etc.). A valuable improvement would be the adoption of a tool, possibly AI-based, not only to translate the entire portal into each user's preferred language but also to allow users to write comments and notes in their own language, which would then be automatically translated for other users based on their selected language.

Many possible directions lie ahead for this project, which is far from complete, as it encourages the adoption of new technologies and tools to be integrated into AROL's ecosystem. However, the most important lesson is that, regardless of any new features or functionalities that may be implemented, developers should always keep the ultimate goal in mind: optimizing users' daily work by placing them at the top of the priority pyramid and maintaining a user-centered approach.

Acronyms

- **AD**: Active Directory
- **AI**: Artificial Intelligence
- **CLI**: Command Line Interface
- **CPU**: Central Processing Unit
- **CRUD**: Create, Read, Update, Delete
- **GPU**: Graphic Processing Unit
- **GUI**: Graphical User Interface
- **HCI**: Human-Computer Interaction
- **HDD**: Hard Disk Drive
- **HW**: HardWare
- **IA**: Information Architecture
- **IT**: Information Technology
- **RAM**: Random Access Memory
- **REST**: REpresentational State Transfer
- **SW**: SoftWare
- **UI**: User Interface
- **URI**: Uniform Resource Identifier
- **UX**: User eXperience
- **VS Code**: Visual Studio Code
- **XLS**: (microsoft) eXceL Spreadsheet

Bibliography

[1] Angular. Angular website. url: https://angular.dev/. (Accessed: 19/02/2025).

[2] Angular CLI. Angular cli website. url: https://angular.dev/tools/cli. (Accessed: 20/02/2025).

[3] Angular Language Service extension. Angular language service website. url: https://marketplace.visualstudio.com/items?itemName=Angular.ng-template. (Accessed: 19/02/2025).

[4] Angular Material. Angular material website. url: https://material.angular.io/. (Accessed: 20/02/2025).

[5] AROL Closure Systems. About us. url: https://www.arol.com/arol-canelli/, . (Accessed: 15/12/2024).

[6] AROL Closure Systems. Quasar f - rf. url: https://www.arol.com/it/tappatrici-per-vino-e-liquori/tappatrici-per-tappo-a-fungo/306-quasar-f-rf-it?1153, . (Accessed: 15/12/2024).

[7] Atlassian. Atlassian bitbucket website. url: https://bitbucket.org/product/. (Accessed: 19/02/2025).

[8] Chart.js. Chart.js website. url: https://www.chartjs.org/. (Accessed: 20/02/2025).

[9] Chessa Alessio. Design and implementation of a microservices-based serialization system. url: https://webthesis.biblio.polito.it/33224/. (Accessed: 20/12/2024).

[10] Cliffydcw. Software processors. url: https://upload.wikimedia.org/wikipedia/commons/1/19/SDLC_-_Software_Development_Life_Cycle.jpg. (Accessed: 30/01/2025).

[11] Concepta. How to define stakeholders for your software development project. url: https://www.conceptatech.com/blog/how-to-define-stakeholders-for-your-software-development-project. (Accessed: 12/01/2025).

[12] Figma. Figma website. url: https://www.figma.com/. (Accessed: 14/01/2025).

[13] GEEKSFORGEEKS. Functional vs. non functional requirements. url: https://www.geeksforgeeks.org/functional-vs-non-functional-requirements/. (Accessed: 10/01/2025).

[14] GeeksForGeeks. Differences between verification and validation. url: https://www.geeksforgeeks.org/differences-between-verification-and-validation/. (Accessed: 23/02/2025).

[15] Git. Git website. url: https://git-scm.com/. (Accessed: 19/02/2025).

[16] Juliette Melton. Jakob nielsen and nancy ruenzel. url: https://live.staticflickr.com/2392/1734507382_a3bf8044bd_b.jpg. (Accessed: 30/01/2025).

[17] Kate Moran - Nielsen Norman Group. Usability (user) testing 101. url: https://www.nngroup.com/articles/usability-testing-101/. (Accessed: 24/02/2025).

[18] Mallory Merrill - DevSquad. What is rapid prototyping in ux? step-by-step guide & wise tips. url: https://devsquad.com/blog/what-is-rapid-prototyping-and-why-is-it-used-in-development. (Accessed: 14/01/2025).

[19] Microsoft Learn. Active directory domain services overview. url: https://learn.microsoft.com/it-it/windows-server/identity/ad-ds/get-started/virtual-dc/active-directory-domain-services-overview. (Accessed: 08/01/2025).

[20] Microsoft Visual Studio Code. Visual studio code website. url: https://code.visualstudio.com/. (Accessed: 19/02/2025).

[21] NGX Bootstrap. Ngx bootstrap website. url: https://valor-software.com/ngx-bootstrap/. (Accessed: 20/02/2025).

[22] ngx-scanner. ngx-scanner website. url: https://www.npmjs.com/package/@zxing/ngx-scanner. (Accessed: 20/02/2025).

[23] Nielsen Norman Group. 10 usability heuristics for user interface design. url: https://www.nngroup.com/articles/ten-usability-heuristics/. (Accessed: 20/12/2024).

[24] PrimeFlex. Primeflex website. url: https://primeflex.org/. (Accessed: 20/02/2025).

[25] PrimeIcons. Primeicons website. url: https://primeng.org/icons. (Accessed: 20/02/2025).

[26] PrimeNG. Primeng website. url: https://v17.primeng.org/. (Accessed: 20/02/2025).

[27] React. React website. url: https://react.dev/. (Accessed: 19/02/2025).

[28] Santoro Francesco. Optimizing industrial operations: A web application for qr code-based machinery information management. url: `https://webthesis.biblio.polito.it/31910/`. (Accessed: 20/12/2024).

[29] Vue. Vue website. url: `https://vuejs.org/`. (Accessed: 19/02/2025).