TQS: Product specification report

Roberto Rolão de Castro [107133]*, *Mariana Figueiredo Perna [108067]*, *Tiago Caridade Gomes [108307]*, *Rafaela Espírito Santo Dias [108782]

v2024-05-08

[This report should be written as the main source of technical documentation on the project, clarifying the functional scope and architectural choices. Provide concise, but informative content, allowing other software engineers to understand the product and quickly access the related resources.
Tips on the expected content, along the document, are meant to be removed.
You may use English or Portuguese; do not mix.]

1 Introduction

1.1 Overview of the project

The objective of this project within the Test and Software Quality (TQS) course is to synthesize quality assurance practices with agile software development to create a functional software solution. We are tasked with developing an application that not only serves a practical purpose but also embodies the principles of robust software architecture, clean coding practices, and rigorous testing methodologies. This aligns with the TQS course's aim to instill a mindset where quality is not an afterthought but a pervasive aspect of the software development lifecycle.

Our application, CliniConnect, is designed to revolutionize the patient experience in scheduling and managing healthcare appointments. At its core, CliniConnect is a digital platform that serves as a nexus between patients and healthcare providers, offering features such as appointment scheduling,

patient registration, and real-time queue management. Its purpose is to enhance service delivery within healthcare facilities by optimizing appointment booking processes, reducing wait times, and facilitating efficient patient flow and billing post-consultation. CliniConnect stands out by providing a comprehensive, intuitive, and accessible tool for healthcare interaction, laying the groundwork for future expansions like mobile app integration and self-service check-in functionalities.

1.2 Limitations

<explain the known limitations/unimplemented (but planned) features>

2 Product concept and requirements

2.1 Vision statement

Our system is designed to streamline the management of patient appointments and admissions across a network of hospitals in Portugal. It will be used by patients who want to make, see, or cancel doctors' appointments, by healthcare professionals to schedule and manage consultations, and by administrative staff to oversee and facilitate the overall patient flow from check-in to billing.

2.2 Personas and scenarios

Persona 1: Mariana Perna

Mariana Perna, a 20-year-old university student studying environmental science, has been dealing with significant myopia since childhood, necessitating annual ophthalmology check-ups. She is a tech-savvy individual who relies heavily on her smartphone for managing her busy academic and social life. Mariana values the convenience of digital solutions that allow her to maintain her health without disrupting her schedule.

Scenario: Routine Ophthalmology Appointment Booking Mariana receives a reminder notification from the CliniConnect app about scheduling her annual ophthalmology appointment. Within minutes, using her smartphone, she accesses the app to view Dr. Dionisio Cortesão's availability at Hospital da Luz in Aveiro. She selects a slot that fits between her lectures and finalizes the booking. Mariana appreciates the seamless experience and the reassurance that her eye health is being monitored regularly without any hassle.

Persona 2: Adelaide Bicho

At 53, Adelaide Bicho is an experienced receptionist at Trofa Saúde Hospital in Porto. She prides herself on her efficiency and her ability to provide a warm and welcoming environment for patients. Adelaide is the first point of contact for many visitors, and she takes her role seriously, understanding that she sets the tone for their healthcare experience.

Scenario: Managing Priority Queue As the day begins, Adelaide logs into the CliniConnect system at her reception desk, checking the day's schedule and patient appointments. She calls the next priority queue number, D43, which belongs to an elderly patient requiring assistance. With CliniConnect's intuitive interface, Adelaide quickly accesses the patient's details, verifies their priority status due to mobility issues, and smoothly directs them to the appropriate department, ensuring they receive prompt care.

Persona 3: Bernardo Oliveira

Bernardo Oliveira is a vibrant 5-year-old starting kindergarten. Accompanied by his mother, Sofia Oliveira, a 35-year-old architect, they both value timely and effective healthcare services. Sofia, as a working professional and a mother, looks for healthcare solutions that minimize waiting time and provide a stress-free experience for Bernardo's healthcare needs.

Scenario: Pediatric Appointment Check-In Bernardo, holding his mother's hand, enters the pediatric wing of their local clinic for a routine check-up. Sofia uses the self-service kiosk to check in, selecting the normal queue option on the CliniConnect system. They are immediately issued ticket number N26. Bernardo is fascinated by the digital signage that will soon display his number, and Sofia appreciates that the system respects their time, allowing for a smooth and orderly visit.

2.3 Project epics and priorities

The implementation of CliniConnect will be carried out incrementally across several iterations, with each epic representing a broad category of functionality crucial to the application's operation. Below are the defined epics, each composed of several user stories that contribute to a cohesive set of functionalities to be developed and refined in successive releases.

Epic 1: Appointment Management System

User Story 1.1: As Mariana Perna, I want to receive timely notifications to schedule my annual ophthalmology check-up, so that I can maintain my eye health with minimal disruption to my schedule.

User Story 1.2: As a patient, I want to view and select available time slots for appointments, so that I can book appointments at my convenience.

User Story 1.3: As a patient, I want to confirm and cancel appointments through the app, ensuring flexibility in managing my schedule.

User Story 1.4: As a receptionist, I want to view the daily appointment schedule, so that I can effectively manage patient flow.

Epic 2: Queue Management and Prioritization

User Story 2.1: As Adelaide Bicho, I want to efficiently manage priority queues, ensuring that patients with urgent needs receive timely care.

User Story 2.2: As a receptionist, I want to quickly access patient details and verify their priority status, ensuring that I can direct them to the appropriate care promptly.

User Story 2.3: As a patient or caregiver, I want to take a queue number through a self-service kiosk, minimizing wait time and streamlining check-in procedures.

Epic 3: Patient-Doctor Matching

User Story 3.1: As a patient, I want to search for doctors by specialty, location, and availability, so that I can find the right healthcare provider for my needs.

User Story 3.2: As a doctor, I want to manage my availability on the platform, ensuring that patients can book appointments with me based on my real-time schedule.

Epic 4: Notification and Alerts System

User Story 4.1: As a patient, I want to receive reminders for upcoming appointments, ensuring that I don't miss them.

User Story 4.2: As a clinic staff, I want to be alerted about no-shows or cancellations, so I can adjust the schedule accordingly and optimize the use of resources.

Epic 5: User Experience and Accessibility

User Story 5.1: As a young patient, I want an engaging and understandable display system that notifies me when it's my turn, making the waiting experience less daunting.

User Story 5.2: As a working parent, I want to easily check in my child for appointments using a system that respects our time and needs for a stress-free experience.

User Story 5.3: As a patient with accessibility needs, I want an application interface that is easy to navigate and use, regardless of my technical abilities.

Implementation Plan

The project will be structured in a series of sprints, each designed to deliver a working increment of the system:

Sprint 1: Focus on setting up the basic appointment booking and notification functionality, targeting user stories from Epics 1 and 4.

Sprint 2: Develop the queue management features and integrate priority handling as specified in Epic 2.

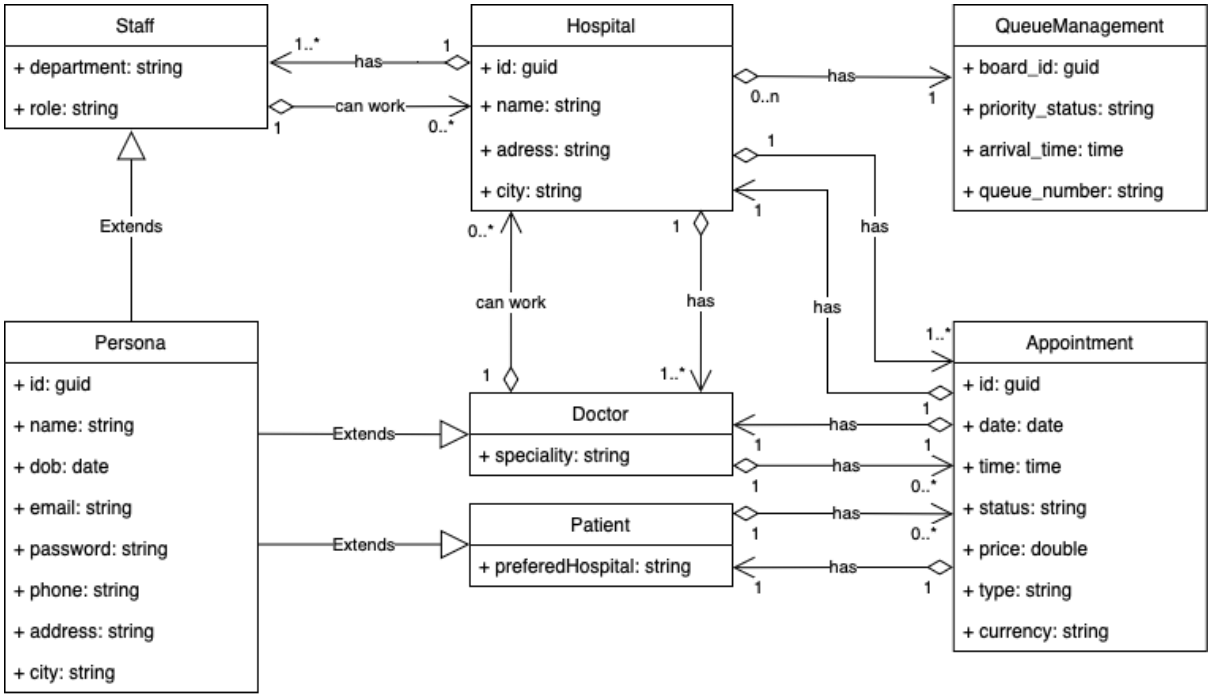
Sprint 3: Enhance the patient-doctor matching features, with emphasis on search functionality and real-time availability updates from Epic 3.

Sprint 4 and beyond: Continuous refinement of user experience, accessibility features from Epic 5, and additional functionalities based on feedback and user testing.

3 Domain model

The class diagram represents a hospital management system with seven entities:

- ***Persona*** is the base class for individuals, containing personal information such as ID, name, date of birth, and contact details.
- ***Staff*** extends ***Persona*** and includes additional attributes for department and function, indicating the function taken in the hospital. Staff can work in multiple hospitals.
- ***Doctor*** also extends ***Persona***, with a specific attribute for specialty. Doctors can work at multiple hospitals and have a one-to-many relationship with ***Patient***.
- ***Patient*** extends ***Persona*** with a preference for a particular hospital.
- ***Appointment*** is associated with ***Patient*** and ***Doctors*** and contains details about the appointment such as ID, date, time, status, as well as financial attributes like price and currency.
- ***Hospital*** is a separate entity with its own attributes like ID, name, address, and city. It has associations with *Staff*, ***Doctor***, and ***QueueManagement***.
- ***QueueManagement*** is linked to Hospital and manages patient queues with attributes such as board ID, priority status, arrival time, and queue number.



4 Architecture notebook

4.1 Key requirements and constrains

For the architecture of CliniConnect, several key requirements and constraints must be addressed to ensure the system is both functional and compliant with expected standards:

- **Accessibility Across Different Platforms:** The application must be accessible from various devices including local PCs at healthcare facilities and remote devices via internet connections. This necessitates a responsive design and possibly a cross-platform mobile application to ensure functionality across all devices.

- **Data Security:** Robust security measures must be implemented to protect sensitive patient data from unauthorized access. This includes secure user authentication mechanisms for all accesses, particularly for remote and mobile interactions.

- **Client-Server Architecture:** The system will be developed using a client-server model where the client software operates on user devices, and the server software runs on a centralized server, likely hosted in a secure cloud environment to facilitate scalability and maintenance.

- **Performance and Scalability Requirements:** Performance benchmarks as outlined in the project's vision and supplementary specifications must be met. This includes ensuring the system can handle a high number of concurrent users and transactions, especially during peak operational hours.

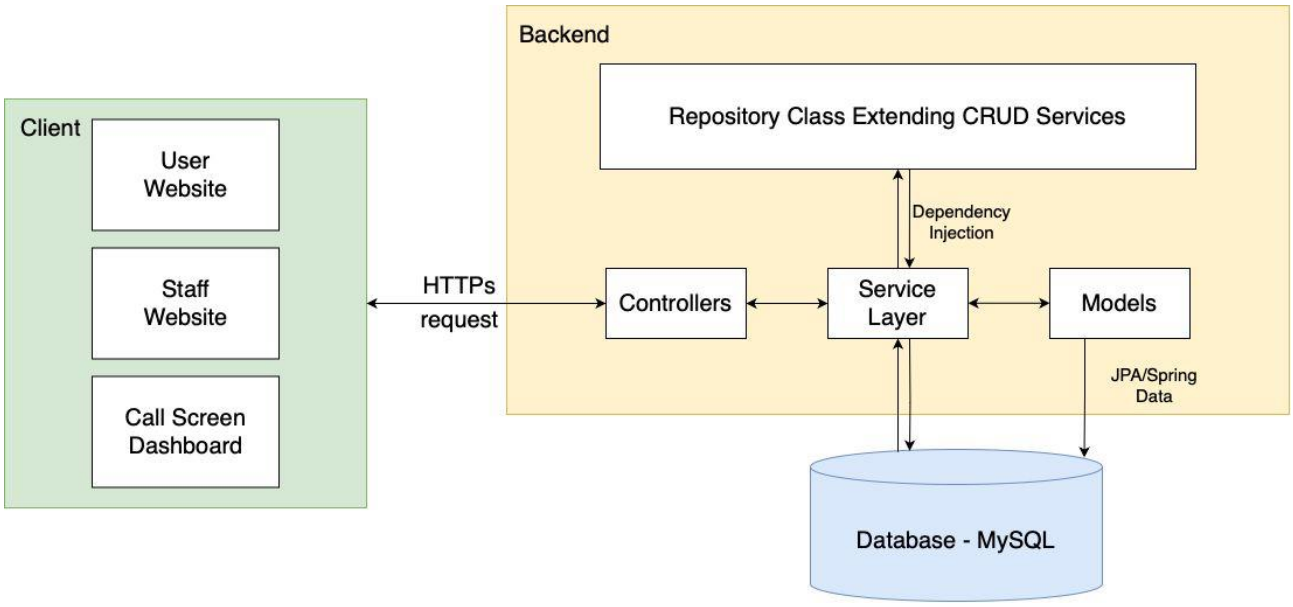
These elements are essential in shaping the architecture of CliniConnect, ensuring it delivers a secure, efficient, and user-friendly experience for both patients and healthcare providers, while accommodating future growth and technological integration.

4.2 Architecture view

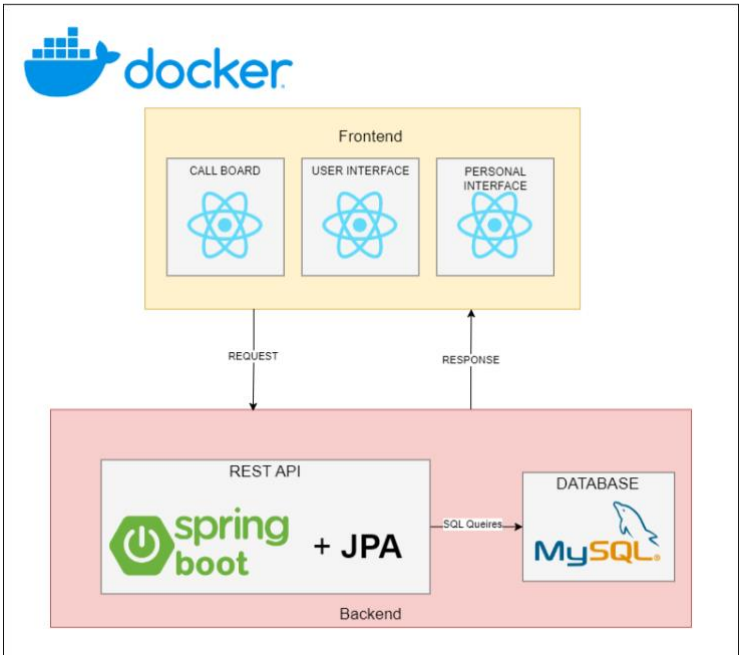
1. **Client Layer**:
   - **User Website**: Interface for end-users to interact with the application.
   - **Staff Website**: Backend interface for staff to manage and respond to user needs.
   - **Call Screen Dashboard**: Specialized interface for managing queues.

2. **Backend Layer**:
   - **Controllers**: Manage the HTTP requests from the client layer, acting as the entry point for all interactions with the backend.
   - **Service Layer**: Handles the business logic of the application, ensuring data manipulation and processing are executed according to business rules.
   - **Repository Class Extending CRUD Services**: Interacts with the database through CRUD (Create, Read, Update, Delete) operations, facilitated by JPA (Java Persistence API)/Spring Data for efficient data management.
   - **Models**: Defines the data structure and possibly the relationships between different data entities.

3. **Database**: Uses MySQL for storing and retrieving data. This is a relational database management system suited for a wide range of applications.

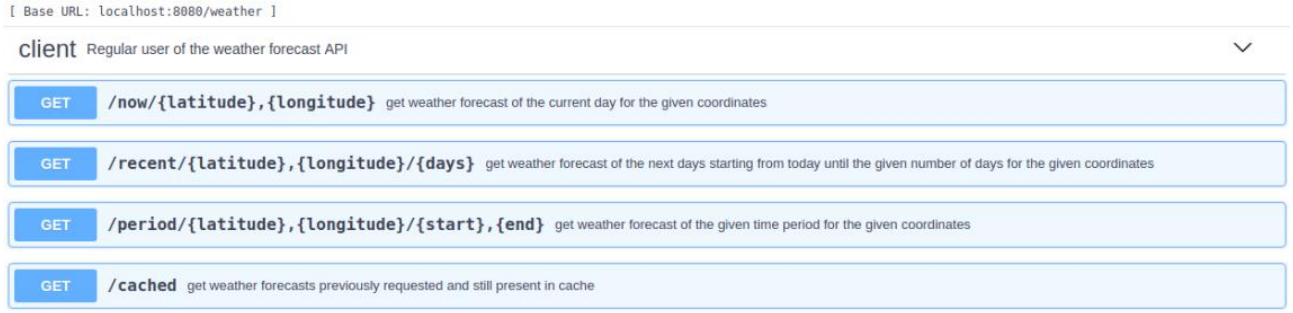


4.3 Deployment architecture



5 API for developers

[Explicar a organização da API. Os detalhes detalhes/documentação dos métodos devem ficar numa solução *hosted* de documentação de APIs, como o Swagger, Postman documentation, ou incluída no próprio desenvolvimento (e.g.: maven site)
<what services/resources can a developer obtain from your REST-API?>
<document the support endpoints>



6 References and resources

<document the key components (e.g.: libraries, web services) or key references (e.g.: blog post) used that were really helpful and certainly would help other students pursuing a similar work>