universidade de aveiro
departamento de eletrónica, telecomunicações e informática

TQS: Product specification report

Miguel Miragaia [108317], Cristiano Nicolau[108536], Vasco Faria [107323], André Gomes [97541]

v2024-04-18

1 Introduction 1
1.1 Overview of the project 1
1.2 Limitations 1

2 Product concept 2
2.1 Vision statement 2
2.2 Personas 2
2.3 Main scenarios 2
2.4 Project epics and priorities 2

3 Domain model 2
4 Architecture notebook 3
4.1 Key requirements and constrains 3
4.2 Architetural view 3
4.3 Deployment architeture 4

5 API for developers 4
6 References and resources 4

1 Introduction

1.1 Overview of the project

In the context of the Test and Quality of Software (TQS) course, our project aims to apply course principles to develop and evaluate the quality of a software solution with the integration of automatic tests. Introducing PeTicket, our application revolutionizes the veterinary care experience. It enhances convenience and quality in veterinary care delivery. Our focus is on ensuring robustness, usability, security, and performance, aligning with TQS principles, to deliver a reliable solution meeting user needs

1.2 Limitations

The main defined features of the system were successfully implemented, ensuring a functional and efficient user experience. Except for the QrCode Scan, the QrCode is generated at the backend and passed to the frontend, but it is not validated through scanning, the team had trouble developing the scanning validation phase, so instead it was created a button to bypass this validation and use the endpoint.

2 Product concept and requirements

2.1 Vision statement

The PeTicket application aims to provide a streamlined and modern solution for veterinary care, addressing the high-level business problem of inefficient appointment scheduling, fragmented medical record management, and suboptimal consultation management in veterinary clinics. Our system will be used to facilitate seamless appointment booking, access to comprehensive pet medical records, and efficient consultation management for both pet owners and clinic professionals.

Compared to other veterinary care software, PeTicket stands out with its intuitive mobile application interface, QR code check-in feature, and real-time patient flow monitoring. These features enhance convenience and efficiency, setting PeTicket apart in the market.

2.2 Personas and scenarios

The main personas for the veterinary clinic are Luisa Costa, the receptionist; Paulo Dias, a dog (Max) owner; and Dr. Mariana Santos, a veterinarian.

- **Luisa Costa**

Luisa Costa is 27 years old and works as a receptionist at the veterinary clinic. She considers herself a good listener, hardworking, and welcoming. With her warmth and professionalism, she greets clients and their animals with care and efficiency, ensuring a positive experience from the first contact. Her dedication to making everyone feel welcome is essential for the clinic's welcoming atmosphere..

- **Paulo Dias**

Paulo is a 28 years old active young man who works full-time and has a strong bond with his dog, Max, a loyal and adventurous German Shepherd. Paulo considers Max an integral part of his family and is always attentive to his health and well-being needs. As a dedicated owner, Paulo values the convenience and quality of veterinary services to ensure the best possible care for his canine companion.

- **Mariana Santos**

Dr. Mariana is a veterinarian passionate about animals and dedicated to her work. She has her own veterinary clinic, where she takes care of a variety of animals, from dogs and cats to birds and reptiles. Mariana is recognized for her attentive care and medical skills, and she highly values efficiency and precision in her records and procedures..

2.3 Use Cases

Paulo use cases:

- **Registering on the Application:**

  Paulo registers on the veterinary clinic's application, providing his personal information and Max's details, such as species, breed, and medical history.

- **Scheduling Appointment:**

  Paulo can schedule appointments for Max through the application, selecting an available date and time on the clinic's calendar, and explaining the entire situation of his pet.

- **Viewing Medical History:**

  Paulo can access Max's medical history on the application, where all previous consultations, treatments, and procedures are recorded.

- **Accessing Prescriptions:**

  After consultations, Paulo can access Max's medical prescriptions on the application, where they are stored for future reference.

- **Updating Max's Information:**

  Paulo can update Max's information on the application, and also changes of address, phone number, or important medical information, ensuring that the records are always accurate and up to date.

Luisa use cases:

- **Checking in with a QR code:**

  The customer arrives at the clinic's counter and shows a QR code that represents the appointment, which Luisa reads with a machine and confirms the presence of both the customer and the animal.

- **Manually scheduling appointments:**

  The customer arrives at the clinic and wants to schedule an appointment. Luisa schedules the appointment in the system with the customer's information and their respective animal. After confirming, the appointment will be entered into the system

- **Create Client and Pet:**

  A new customer comes to the clinic for the first time. Luisa collects the necessary information to create a profile for both the client and their pet in the system. She inputs details such as the client's contact information and the pet's information, ensuring that all future visits are seamlessly managed and tracked.

Mariana use cases:

- **View scheduled appointments:**

  Mariana accesses the veterinary clinic's app to view all the scheduled appointments. Mariana can view the date and time of the appointments.

- **View appointment details:**

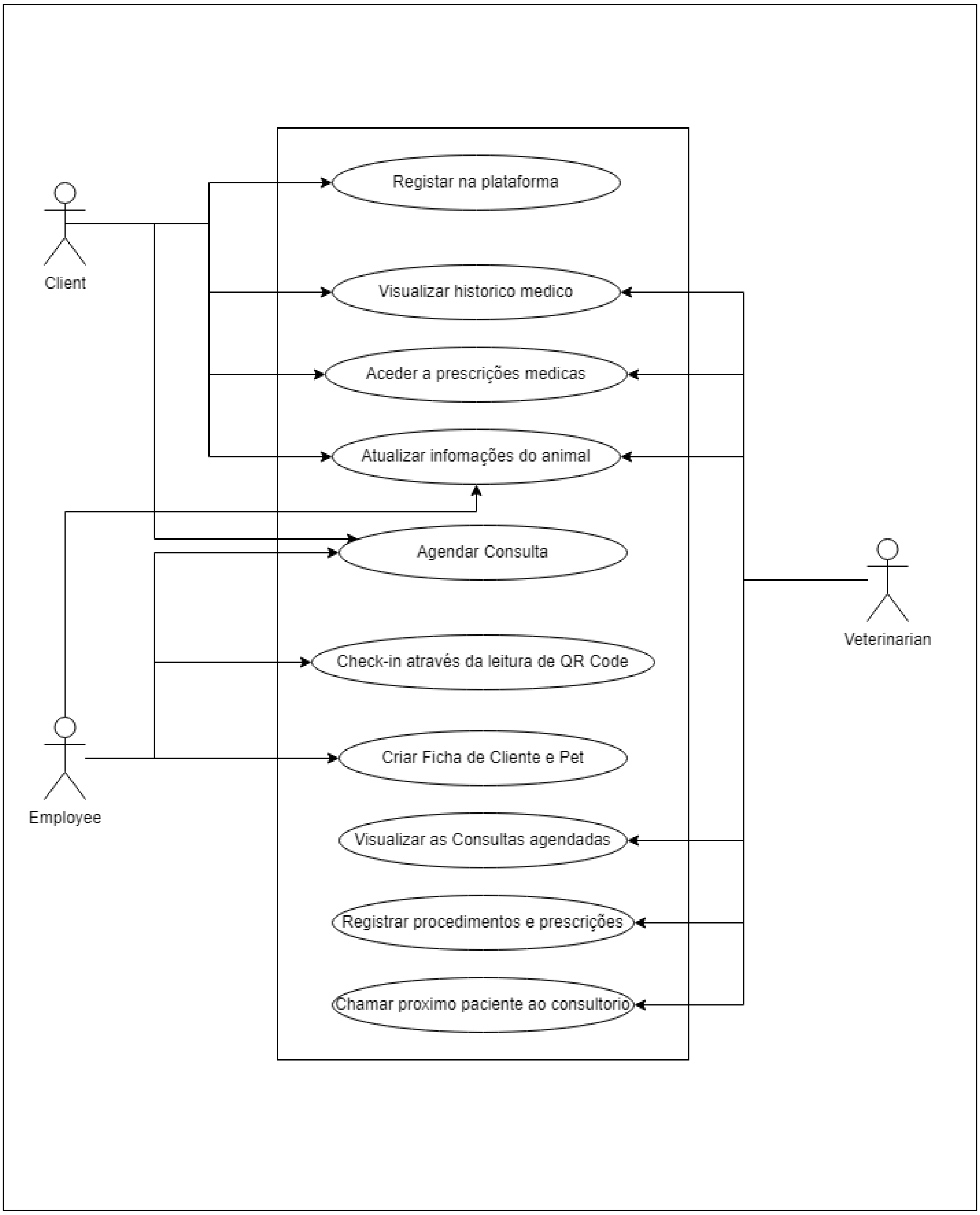
  Mariana accesses the veterinary clinic's app to view all the scheduled appointments. She can select each appointment, in order to check the given pet's details, as well as the owner's details.

- **Register procedures and prescriptions :**

  Mariana accesses the veterinary clinic's app after the appointment is done, and registers the required procedures that the pet needs and/or the medication the pet requires.

- **Call the next appointment to the Consultory**

  When it's time for the next appointment, she ensures a smooth transition by calling the patient into the consultory. The information will be available at a Display shown in the lobby for the clients.


Client
Employee
Veterinarian
Registar na plataforma
Visualizar historico medico
Aceder a prescrições medicas
Atualizar infomações do animal
Agendar Consulta
Check-in através da leitura de QR Code
Criar Ficha de Cliente e Pet
Visualizar as Consultas agendadas
Registrar procedimentos e prescrições
Chamar proximo paciente ao consultorio

2.4 Project epics and priorities

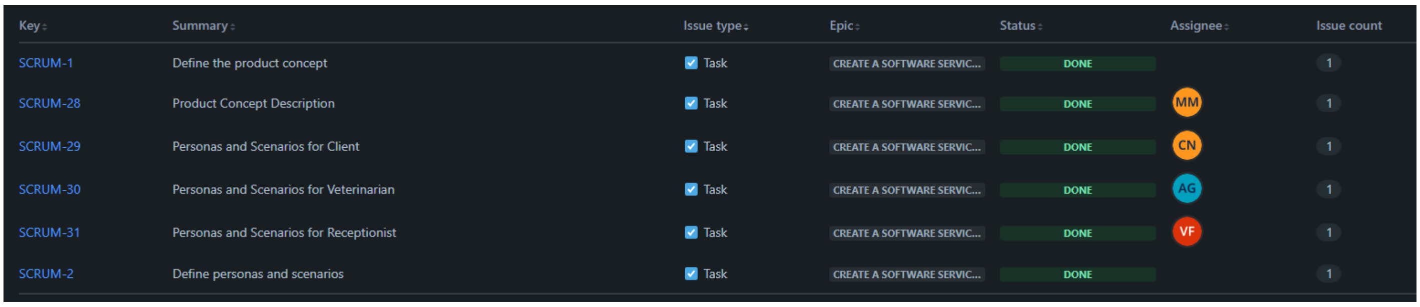
To create a comprehensive software solution for the veterinary clinic, we defined four core epics:

1. Create Software Service for a Clinic Employee
2. Create Software Service for a Clinic Vet
3. Create Software Service for a Clinic Client
4. Create Software Service for a Clinic

Each epic was broken down into issues and subtasks, ensuring a detailed and structured approach to development. When starting the development of new features, we opened a branch named after the Scrum ID and the issue description. Every commit was associated with the subtasks of that issue. Upon completing the development of an issue, we opened a pull request to merge the changes into the dev branch, which was then reviewed and accepted by another group member.

Iteration 1

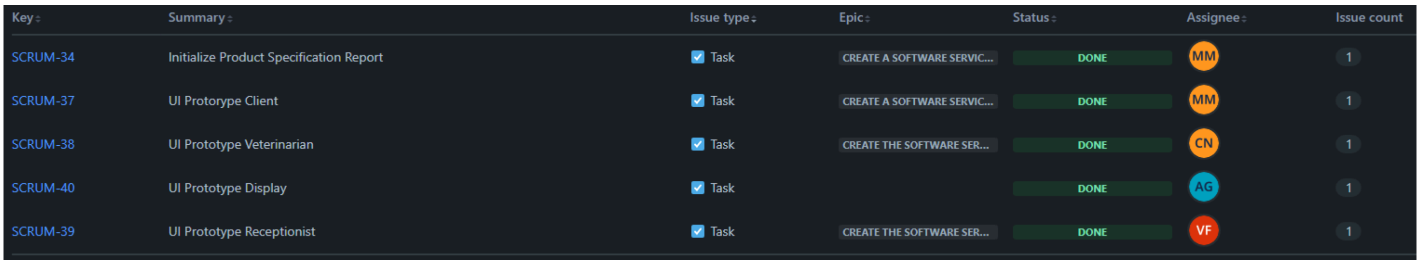
1 - Definition of Product Concept, Personas, User Stories
2 - Team resources setup
3 - Backlog Initialization



Key	Summary	Issue type	Epic	Status	Assignee	Issue count
SCRUM-1	Define the product concept	✓ Task	CREATE A SOFTWARE SERVIC...	DONE		1
SCRUM-28	Product Concept Description	✓ Task	CREATE A SOFTWARE SERVIC...	DONE	MM	1
SCRUM-29	Personas and Scenarios for Client	✓ Task	CREATE A SOFTWARE SERVIC...	DONE	CN	1
SCRUM-30	Personas and Scenarios for Veterinarian	✓ Task	CREATE A SOFTWARE SERVIC...	DONE	AG	1
SCRUM-31	Personas and Scenarios for Receptionist	✓ Task	CREATE A SOFTWARE SERVIC...	DONE	VF	1
SCRUM-2	Define personas and scenarios	✓ Task	CREATE A SOFTWARE SERVIC...	DONE		1

Iteration 2

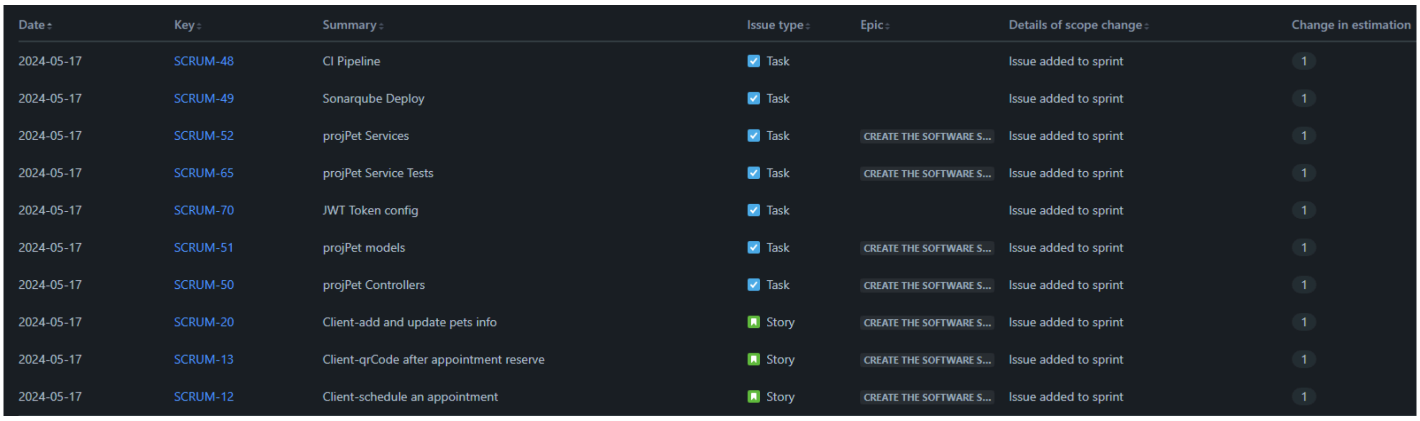
1 - System Architecture definition
2 - SQE Tools and Practices definition
3 - Product Specification Report Initialization
4 - UI Prototype development



Key	Summary	Issue type	Epic	Status	Assignee	Issue count
SCRUM-34	Initialize Product Specification Report	✓ Task	CREATE A SOFTWARE SERVIC...	DONE	MM	1
SCRUM-37	UI Protorype Client	✓ Task	CREATE A SOFTWARE SERVIC...	DONE	MM	1
SCRUM-38	UI Prototype Veterinarian	✓ Task	CREATE THE SOFTWARE SER...	DONE	CN	1
SCRUM-40	UI Prototype Display	✓ Task		DONE	AG	1
SCRUM-39	UI Prototype Receptionist	✓ Task	CREATE THE SOFTWARE SER...	DONE	VF	1

Iteration 3

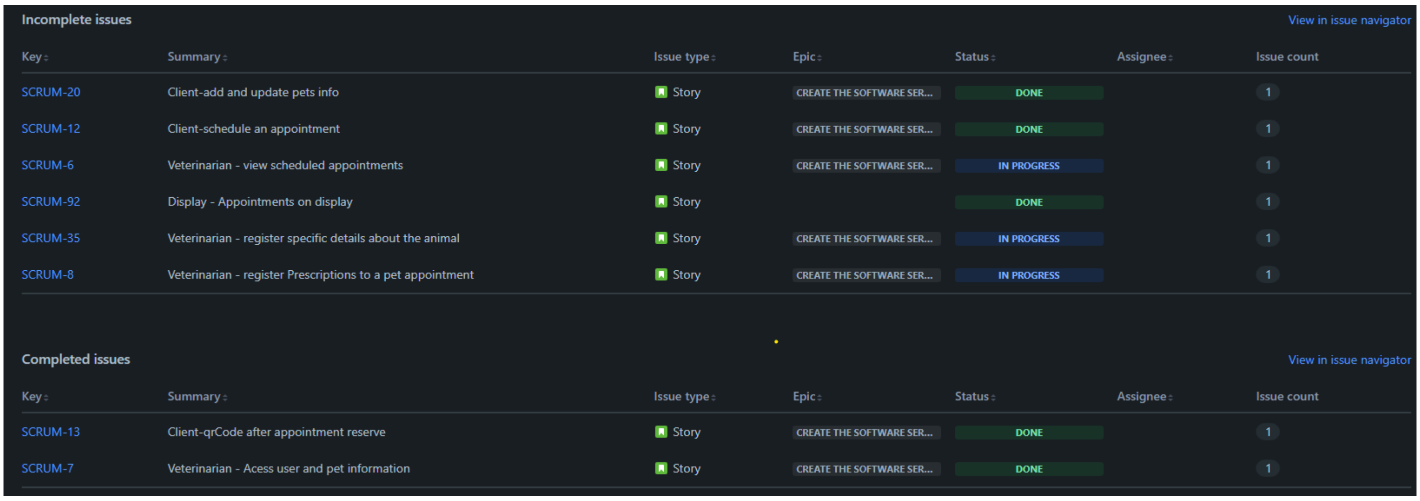
1 - User Stories Development
2 - Definition of Display Usage
3 - QA Manual Initialization
4 - CI Pipeline Implementation
5 - Test Initialization
6 - UI Development



Date	Key	Summary	Issue type	Epic	Details of scope change	Change in estimation
2024-05-17	SCRUM-48	CI Pipeline	Task		Issue added to sprint	1
2024-05-17	SCRUM-49	Sonarqube Deploy	Task		Issue added to sprint	1
2024-05-17	SCRUM-52	projPet Services	Task	CREATE THE SOFTWARE S...	Issue added to sprint	1
2024-05-17	SCRUM-65	projPet Service Tests	Task	CREATE THE SOFTWARE S...	Issue added to sprint	1
2024-05-17	SCRUM-70	JWT Token config	Task		Issue added to sprint	1
2024-05-17	SCRUM-51	projPet models	Task	CREATE THE SOFTWARE S...	Issue added to sprint	1
2024-05-17	SCRUM-50	projPet Controllers	Task	CREATE THE SOFTWARE S...	Issue added to sprint	1
2024-05-17	SCRUM-20	Client-add and update pets info	Story	CREATE THE SOFTWARE S...	Issue added to sprint	1
2024-05-17	SCRUM-13	Client-qrCode after appointment reserve	Story	CREATE THE SOFTWARE S...	Issue added to sprint	1
2024-05-17	SCRUM-12	Client-schedule an appointment	Story	CREATE THE SOFTWARE S...	Issue added to sprint	1

Iteration 4

1 - User Stories Development
2 - Swagger Configuration
3 - Test Development



Incomplete issues

View in issue navigator

Key	Summary	Issue type	Epic	Status	Assignee	Issue count
SCRUM-20	Client-add and update pets info	Story	CREATE THE SOFTWARE SER...	DONE		1
SCRUM-12	Client-schedule an appointment	Story	CREATE THE SOFTWARE SER...	DONE		1
SCRUM-6	Veterinarian - view scheduled appointments	Story	CREATE THE SOFTWARE SER...	IN PROGRESS		1
SCRUM-92	Display - Appointments on display	Story		DONE		1
SCRUM-35	Veterinarian - register specific details about the animal	Story	CREATE THE SOFTWARE SER...	IN PROGRESS		1
SCRUM-8	Veterinarian - register Prescriptions to a pet appointment	Story	CREATE THE SOFTWARE SER...	IN PROGRESS		1

Completed issues

View in issue navigator

Key	Summary	Issue type	Epic	Status	Assignee	Issue count
SCRUM-13	Client-qrCode after appointment reserve	Story	CREATE THE SOFTWARE SER...	DONE		1
SCRUM-7	Veterinarian - Acess user and pet information	Story	CREATE THE SOFTWARE SER...	DONE		1

Iteration 5

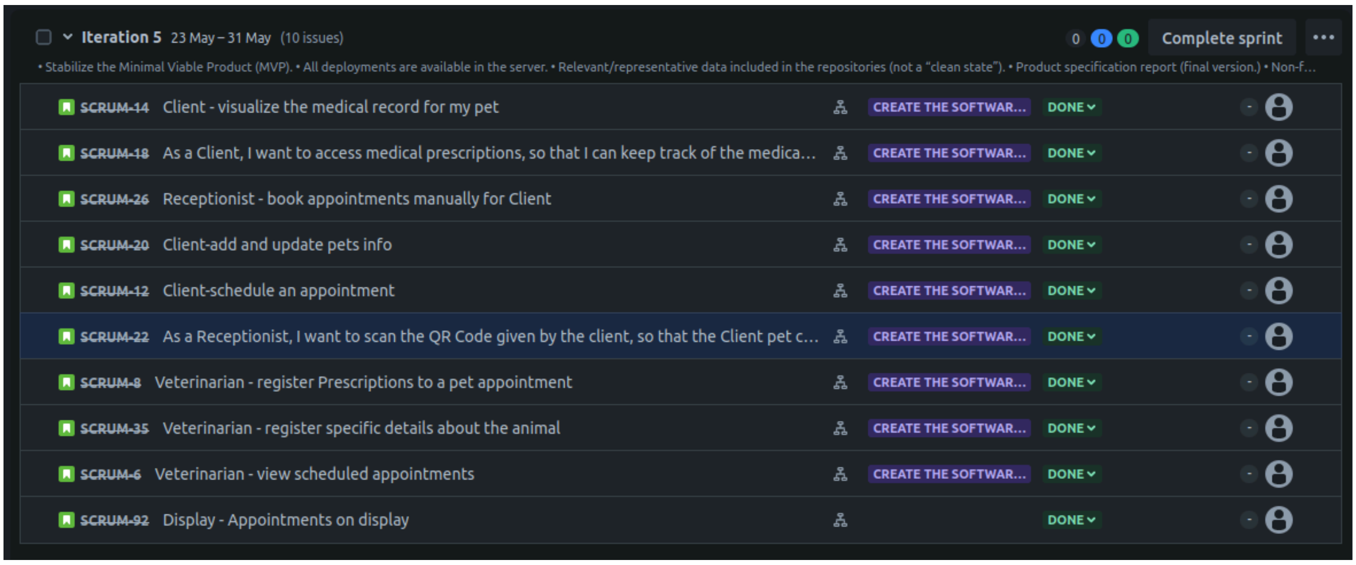
1 - CD

2 - User Stories Finalization

3 - Test Finalization

4 - QA Manual Initialization

5 - Product Specification Report Initialization

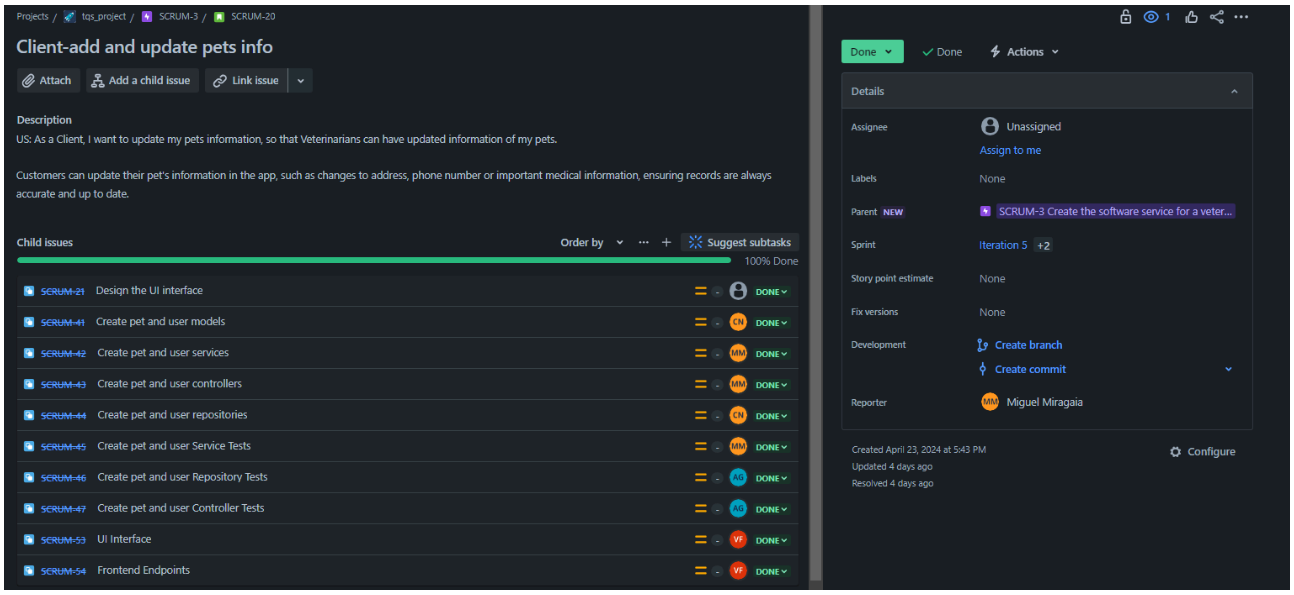


Iteration 5 23 May – 31 May (10 issues) 0 0 0 Complete sprint

• Stabilize the Minimal Viable Product (MVP). • All deployments are available in the server. • Relevant/representative data included in the repositories (not a "clean state"). • Product specification report (final version.) • Non-f...

~~SCRUM-14~~	Client - visualize the medical record for my pet	CREATE THE SOFTWAR...	DONE
~~SCRUM-18~~	As a Client, I want to access medical prescriptions, so that I can keep track of the medica...	CREATE THE SOFTWAR...	DONE
~~SCRUM-26~~	Receptionist - book appointments manually for Client	CREATE THE SOFTWAR...	DONE
~~SCRUM-20~~	Client-add and update pets info	CREATE THE SOFTWAR...	DONE
~~SCRUM-12~~	Client-schedule an appointment	CREATE THE SOFTWAR...	DONE
~~SCRUM-22~~	As a Receptionist, I want to scan the QR Code given by the client, so that the Client pet c...	CREATE THE SOFTWAR...	DONE
~~SCRUM-8~~	Veterinarian - register Prescriptions to a pet appointment	CREATE THE SOFTWAR...	DONE
~~SCRUM-35~~	Veterinarian - register specific details about the animal	CREATE THE SOFTWAR...	DONE
~~SCRUM-6~~	Veterinarian - view scheduled appointments	CREATE THE SOFTWAR...	DONE
~~SCRUM-92~~	Display - Appointments on display		DONE

Issue Example:



Projects / tqs_project / SCRUM-3 / SCRUM-20

Client-add and update pets info

Attach | Add a child issue | Link issue

Description

US: As a Client, I want to update my pets information, so that Veterinarians can have updated information of my pets.

Customers can update their pet's information in the app, such as changes to address, phone number or important medical information, ensuring records are always accurate and up to date.

Child issues — Order by — Suggest subtasks — 100% Done

~~SCRUM-21~~	Design the UI interface		DONE
~~SCRUM-41~~	Create pet and user models	CN	DONE
~~SCRUM-42~~	Create pet and user services	MM	DONE
~~SCRUM-43~~	Create pet and user controllers	MM	DONE
~~SCRUM-44~~	Create pet and user repositories	CN	DONE
~~SCRUM-45~~	Create pet and user Service Tests	MM	DONE
~~SCRUM-46~~	Create pet and user Repository Tests	AG	DONE
~~SCRUM-47~~	Create pet and user Controller Tests	AG	DONE
~~SCRUM-53~~	UI Interface	VF	DONE
~~SCRUM-54~~	Frontend Endpoints	VF	DONE

Done — ✓ Done — Actions

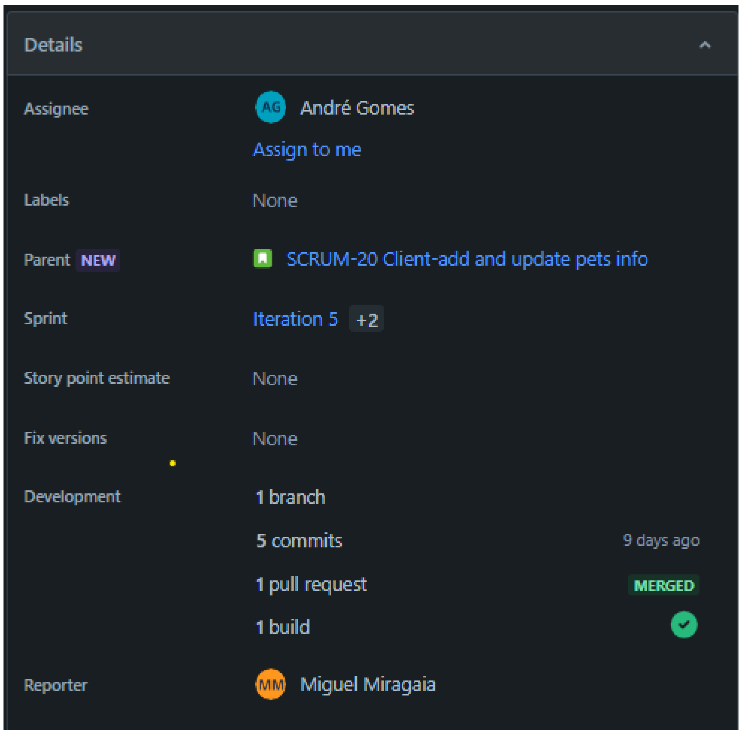
Details

Assignee	Unassigned Assign to me
Labels	None
Parent NEW	SCRUM-3 Create the software service for a veter...
Sprint	Iteration 5 +2
Story point estimate	None
Fix versions	None
Development	Create branch Create commit
Reporter	Miguel Miragaia

Created April 23, 2024 at 5:43 PM
Updated 4 days ago
Resolved 4 days ago

Configure

Subtask Example:

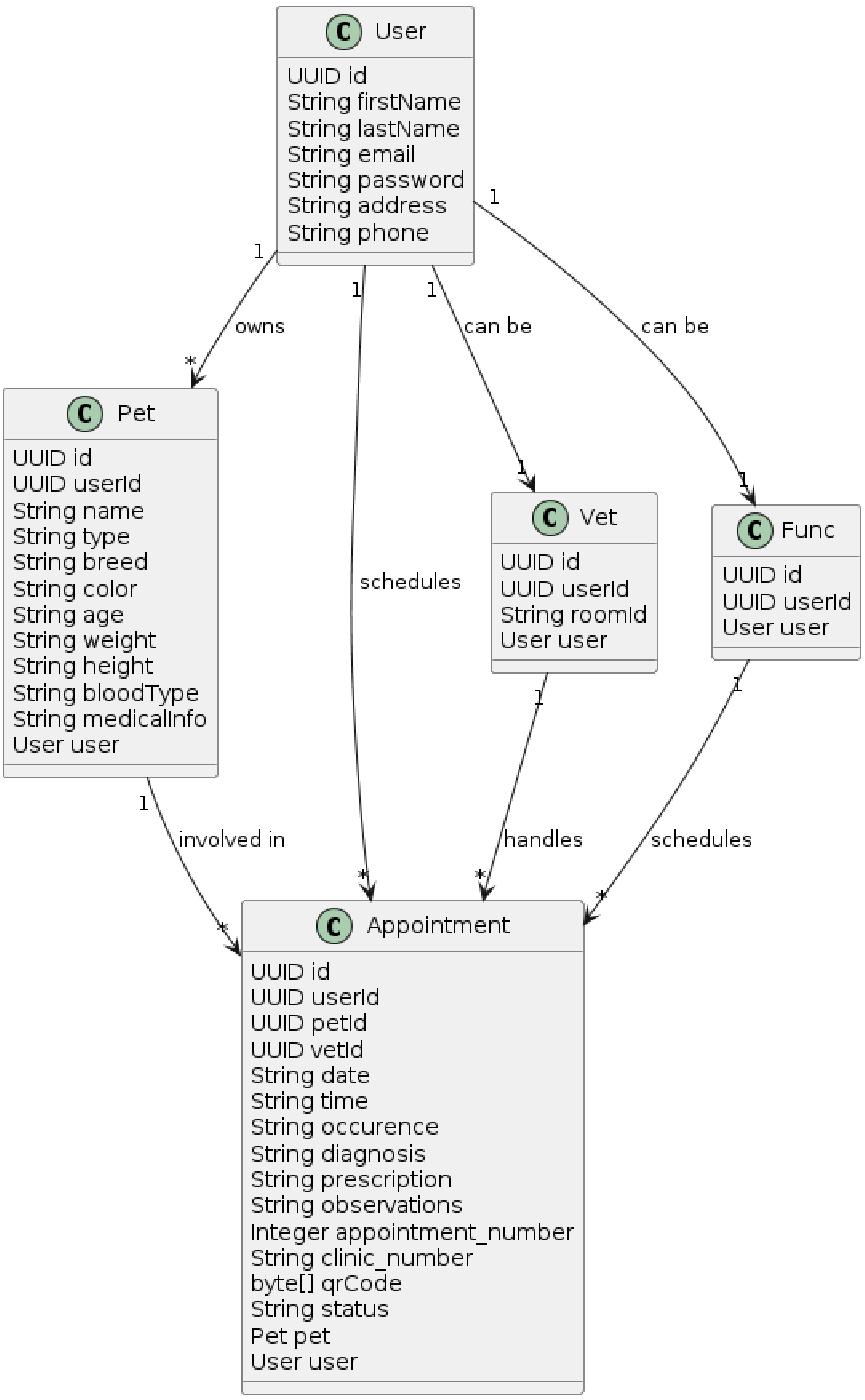


Details

Assignee	AG André Gomes Assign to me
Labels	None
Parent NEW	SCRUM-20 Client-add and update pets info
Sprint	Iteration 5 +2
Story point estimate	None
Fix versions	None
Development	1 branch 5 commits 9 days ago 1 pull request MERGED 1 build ✓
Reporter	MM Miguel Miragaia

3 Domain model

In this system, various entities interact to provide a comprehensive service to pet owners, veterinarians, and clinic staff. The core entities managed in this domain are User, Pet, Appointment, Vet, and Func (clinic functionary). These entities are interrelated, representing the different roles and interactions within the clinic's ecosystem.


User
UUID id
String firstName
String lastName
String email
String password
String address
String phone
Pet
UUID id
UUID userId
String name
String type
String breed
String color
String age
String weight
String height
String bloodType
String medicalInfo
User user
Vet
UUID id
UUID userId
String roomId
User user
Func
UUID id
UUID userId
User user
Appointment
UUID id
UUID userId
UUID petId
UUID vetId
String date
String time
String occurence
String diagnosis
String prescription
String observations
Integer appointment_number
String clinic_number
byte[] qrCode
String status
Pet pet
User user
owns
schedules
can be
can be
involved in
handles
schedules
1
*

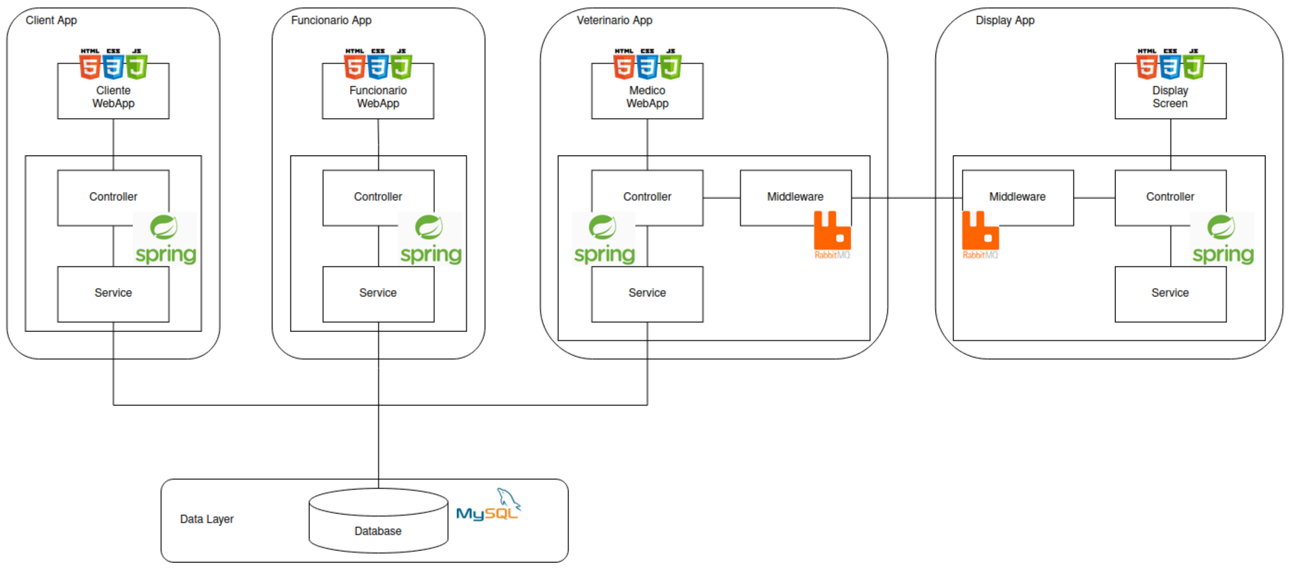
4 Architecture notebook

4.1 Key requirements and constrains

The architecture choice was made taking into account this issues:

- The system must be capable of dealing with several requests simultaneously.
- The system should not permit unknown access to confidential data, needing an authentication system.
- The system must be deployed in a Virtual Machine and run on an external server.
- The system must provide different platforms for the different types of users.
- The system must be available and user-friendly on various devices
- The system must be robust and have fault tolerance mechanisms

4.2 Architecture view



The architecture is composed of various web applications, each intended for a specific group of users and functionalities, all integrated with a central database and using middleware for communication between certain parts of the system.

Client App:

- Web Client App: User interface for clients, developed with HTML, CSS, and JavaScript.
- Controller: Manages HTTP requests from clients.
- Service: Business logic of the application, processing data and interacting with the database.

Employee App:

- Employee Web App: Similar to the Web Client App, intended for employees.
- Controller: Manages HTTP requests from employees.

- Service: Business logic specific to employees.

Veterinarian App:

- Veterinarian Web App: Interface for veterinarians.
- Controller: Manages HTTP requests from veterinarians.
- Service: Processes business logic for veterinarians.

Middleware (RabbitMQ):

- Used for veterinarians to call the next client on the displays.

Display App:

- Display Screen: Displays real-time information, such as clients to be called for consultation.
- Controller: Manages display functionalities.
- Service: Contains business logic for display operations, including a cache to store patients called for each consultation room.

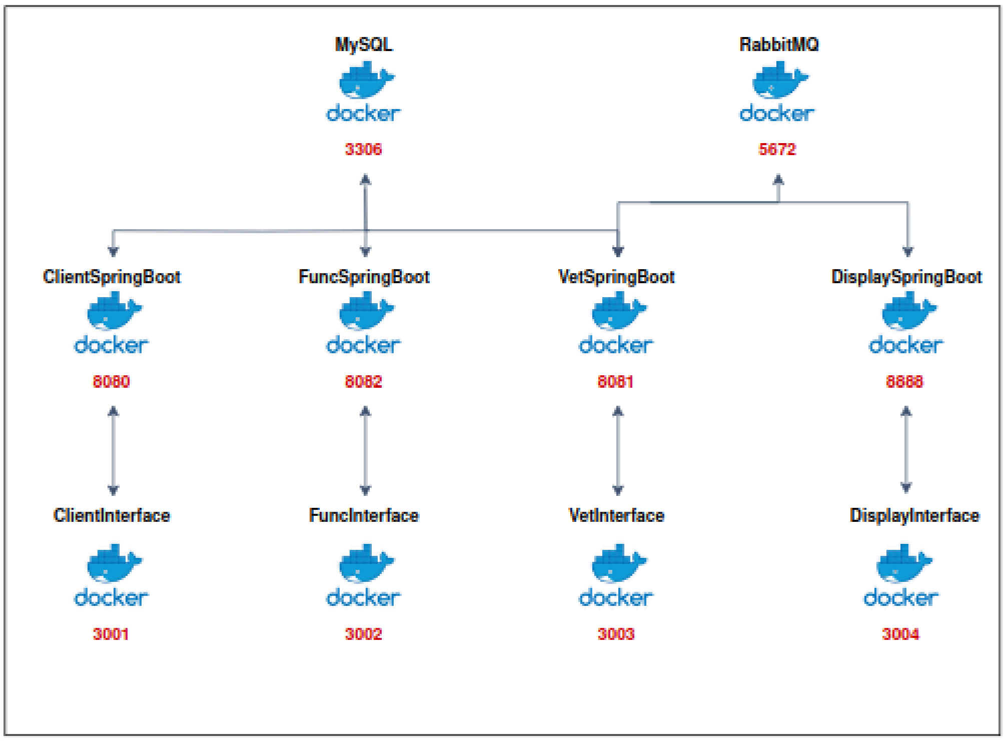
Data Layer:

- Database (MySQL): Central database that stores all the data necessary for the different applications.

Workflow:

- User Interaction: Users interact with the corresponding web applications, sending HTTP requests to the Spring controllers.
- Request Processing: Controllers trigger services to process business logic and interact with the database.
- Communication between Systems: The RabbitMQ middleware facilitates communication between the veterinarian and the display, without the display needing to be connected to the database.

4.3 Deployment architecture

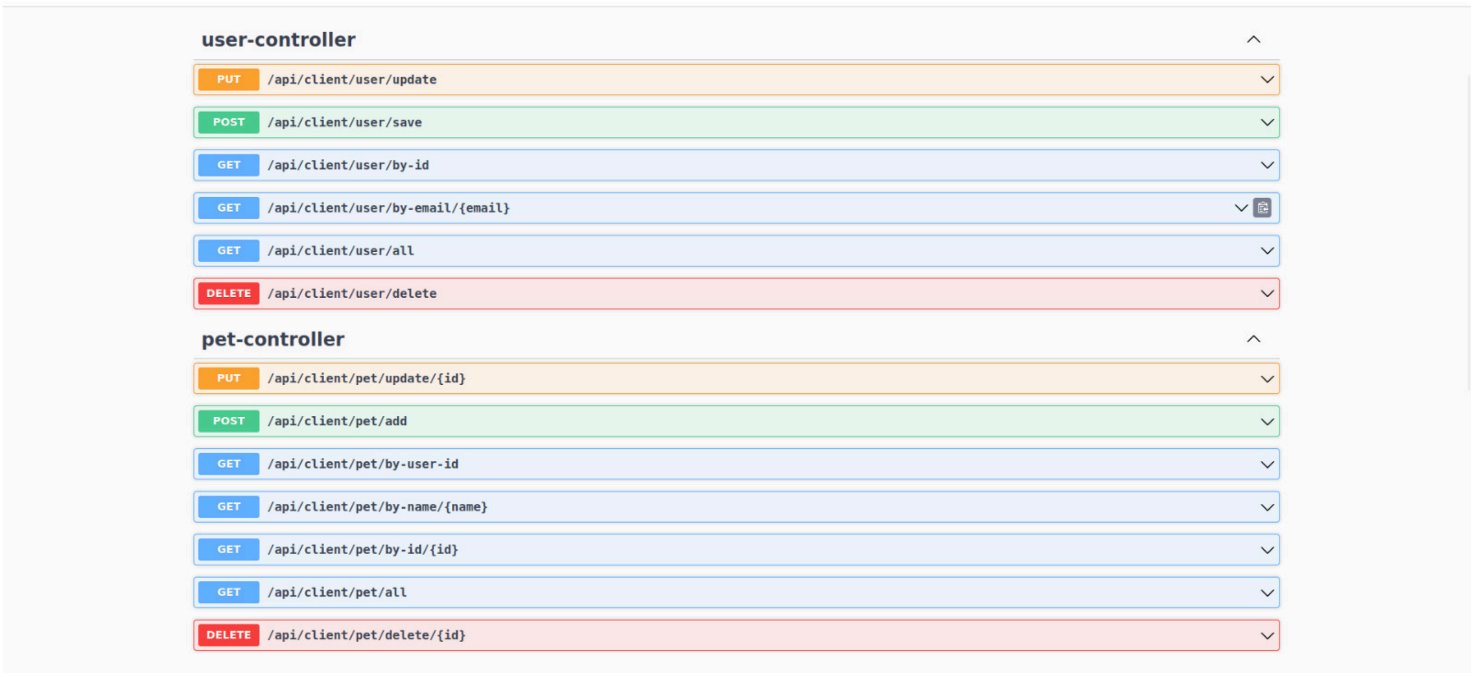


This diagram describes our microservices architecture using Docker containers to encapsulate services and interfaces, ensuring a modular and scalable deployment. Each service interacts with its frontend and the database or middleware as needed. This setup promotes a clear separation of responsibilities and facilitates the maintenance and scaling of services. All components are deployed in our server infrastructure.

5 API for developers

The REST API swagger documentation is available to consult at:

- **Client:** http://deti-tqs-13.ua.pt:8080/swagger-ui/index.html



user-controller

PUT /api/client/user/update

POST /api/client/user/save

GET /api/client/user/by-id

GET /api/client/user/by-email/{email}

GET /api/client/user/all

DELETE /api/client/user/delete

pet-controller

PUT /api/client/pet/update/{id}

POST /api/client/pet/add

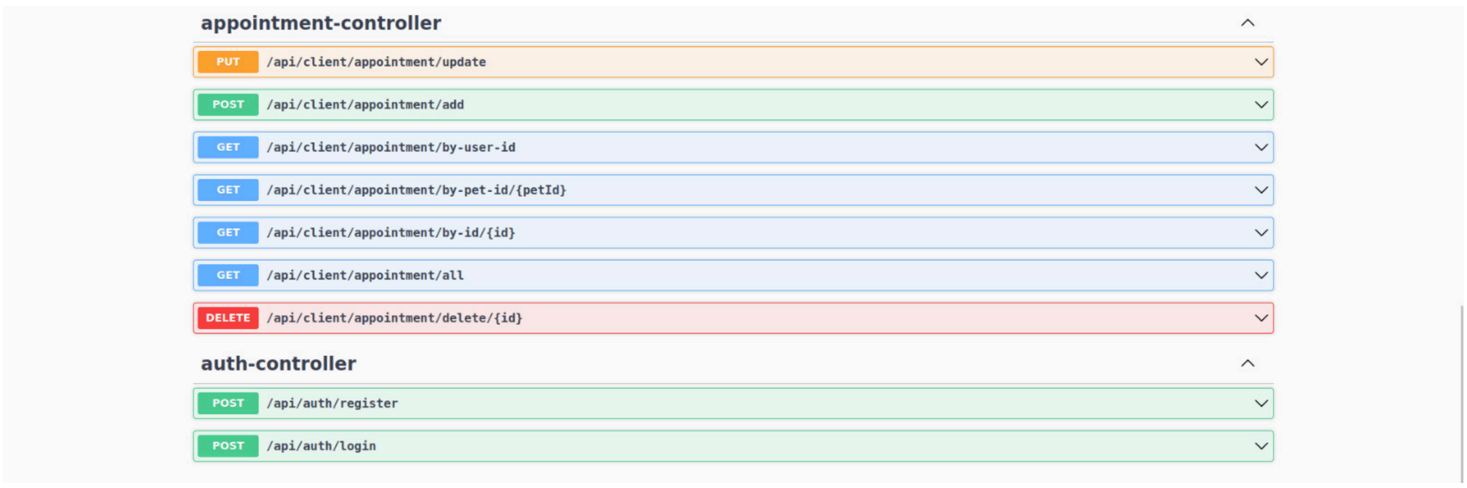
GET /api/client/pet/by-user-id

GET /api/client/pet/by-name/{name}

GET /api/client/pet/by-id/{id}

GET /api/client/pet/all

DELETE /api/client/pet/delete/{id}



appointment-controller

PUT /api/client/appointment/update

POST /api/client/appointment/add

GET /api/client/appointment/by-user-id

GET /api/client/appointment/by-pet-id/{petId}

GET /api/client/appointment/by-id/{id}

GET /api/client/appointment/all

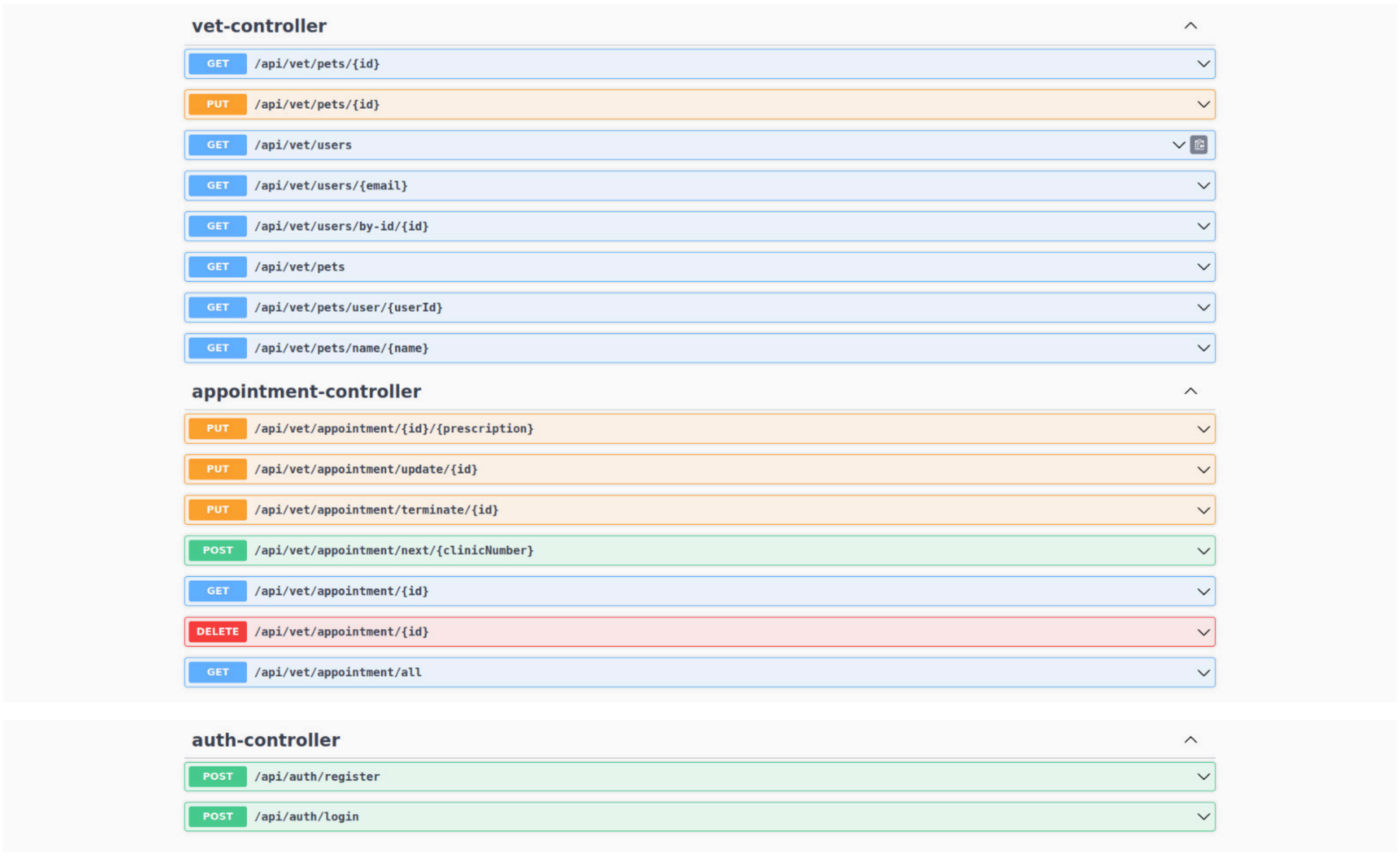
DELETE /api/client/appointment/delete/{id}

auth-controller

POST /api/auth/register

POST /api/auth/login

- **Vet:** http://deti-tqs-13.ua.pt:8081/swagger-ui/index.html



vet-controller

GET /api/vet/pets/{id}
PUT /api/vet/pets/{id}
GET /api/vet/users
GET /api/vet/users/{email}
GET /api/vet/users/by-id/{id}
GET /api/vet/pets
GET /api/vet/pets/user/{userId}
GET /api/vet/pets/name/{name}

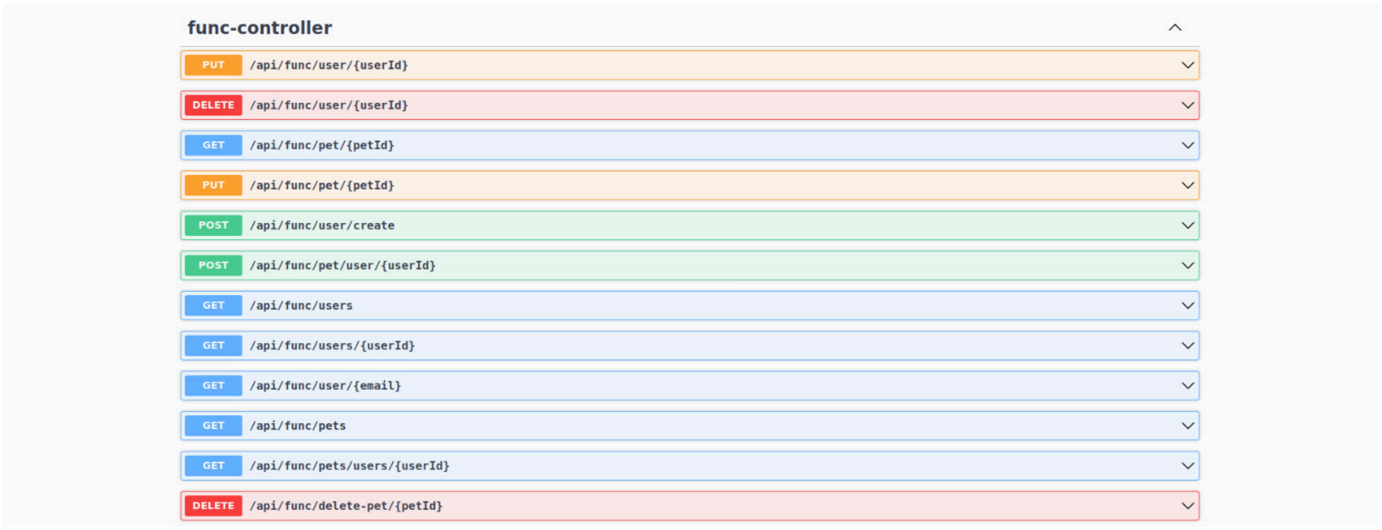
appointment-controller

PUT /api/vet/appointment/{id}/{prescription}
PUT /api/vet/appointment/update/{id}
PUT /api/vet/appointment/terminate/{id}
POST /api/vet/appointment/next/{clinicNumber}
GET /api/vet/appointment/{id}
DELETE /api/vet/appointment/{id}
GET /api/vet/appointment/all

auth-controller

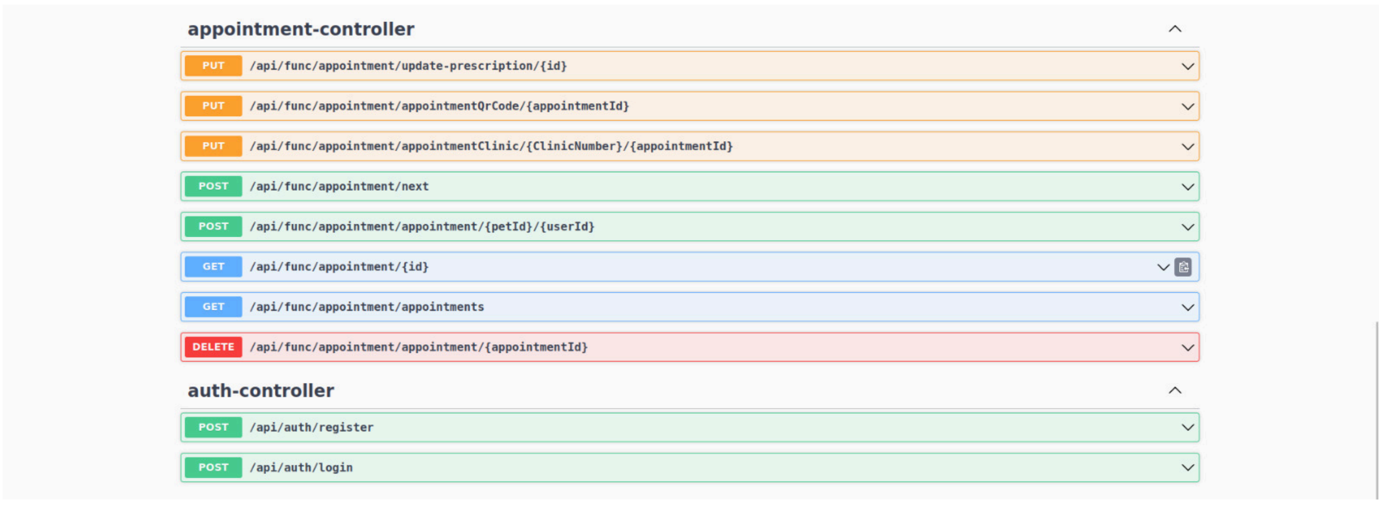
POST /api/auth/register
POST /api/auth/login

- **ClinicFunc:** http://deti-tqs-13.ua.pt:8082/swagger-ui/index.html



func-controller

PUT /api/func/user/{userId}
DELETE /api/func/user/{userId}
GET /api/func/pet/{petId}
PUT /api/func/pet/{petId}
POST /api/func/user/create
POST /api/func/pet/user/{userId}
GET /api/func/users
GET /api/func/users/{userId}
GET /api/func/user/{email}
GET /api/func/pets
GET /api/func/pets/users/{userId}
DELETE /api/func/delete-pet/{petId}


appointment-controller

PUT /api/func/appointment/update-prescription/{id}

PUT /api/func/appointment/appointmentQrCode/{appointmentId}

PUT /api/func/appointment/appointmentClinic/{ClinicNumber}/{appointmentId}

POST /api/func/appointment/next

POST /api/func/appointment/appointment/{petId}/{userId}

GET /api/func/appointment/{id}

GET /api/func/appointment/appointments

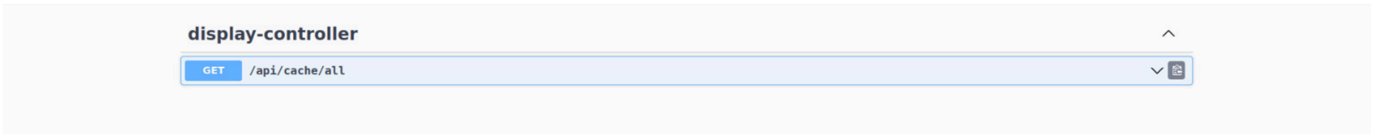
DELETE /api/func/appointment/appointment/{appointmentId}

auth-controller

POST /api/auth/register

POST /api/auth/login

- **Display:** http://deti-tqs-13.ua.pt:8888/swagger-ui/index.html


display-controller

GET /api/cache/all

6 References and resources

Swagger. "API Documentation & Design Tools for Teams | Swagger." Accessed May 27, 2024. https://swagger.io/solutions/api-documentation/