deti universidade de aveiro
departamento de eletrónica,
telecomunicações e informática

TQS: Product specification report

João Sousa [103415], Vânia Morais [102383], Catarina Costa [103696]
v2023-01-02

1 Introduction

1.1 Overview of the project

The objectives of this project include developing a viable MVP using software enterprise architecture standards and applying a Software Quality Assurance (SQA) strategy throughout the software engineering process.

Our project focuses on a collection point network as a service, with Associated Pickup Points (ACP).

The system components include an eStore Web application, a platform backend implemented in Spring Boot, Web interfaces for managing the platform and associated partners.

Technological constraints include choosing suitable frameworks and languages for each component, implementing comprehensive testing strategies, and adopting agile backlog management, feature-branching workflow with code reviews, software testing, CI/CD pipeline setup, and containers-based deployment.

1.2 Limitations

Some of the limitations we had in this project were the implementation of the notification system, where the user would be notified by email or in the eStore when they could collect their order. We also had limitations in testing the code, due to some obstacles we had to overcome regarding the composition of the team. Due to this very problem, we did not have time to run tests on both frontends, but throughout the reports we will talk about how we would apply these tests.
The authentication was incomplete due to lack of time, so it was left in the "auth" branch of the backend repository.

deti universidade de aveiro
departamento de eletrónica,
telecomunicações e informática

2 Product concept

2.1 Vision statement

The PackagePal application is designed to provide a convenient and efficient solution for customers to collect their orders through a network of pickup points. The high-level problem being solved by our system is the need for flexible and reliable delivery options in the e-commerce industry.

Our system allows customers to pick up at Associated Collection Points (ACP) . ACPs are local partner locations such as neighborhood grocery stores or laundries. By offering these pickup options, we simplify the delivery process and provide convenience for customers who may not be available at home during delivery times or prefer alternative pickup locations.

The PackagePal app enhances the customer experience by integrating with eStors, allowing customers to select their preferred pickup point during the ordering process. The system facilitates the communication between eStores and customers, providing them with information about the available pickup points and their locations. While PackagePal does not handle notifications, it collaborates with eStores to ensure timely updates regarding the delivery status of the orders.

By integrating with external partners, PackagePal expands the network of pickup points, offering customers a broader range of options for their convenience. The system streamlines the process of selecting and managing pickup points, making it seamless and user-friendly for both customers and eStores owners. Customers need to visit the respective eStore's platform to stay informed about the status of their orders.

Overall, the PackagePal application aims to provide customers with flexible and convenient pickup options. By simplifying the delivery process and leveraging a network of pickup points, we aim to enhance customer satisfaction and increase the efficiency of order fulfillment in the e-commerce industry.

2.2 Personas and scenarios

Persona 1: Joana - The Enthusiastic Book Lover

Joana is a 28 avid book lover and enjoys spending her free time reading. She finds solace in exploring different genres and expanding her knowledge through books. Joana believes that reading is not only a hobby but also a way to relax and escape from her daily routine. She prefers ordering books online due to the convenience it offers, allowing her to browse a wide selection and have them delivered to her doorstep.

Motivation: Joana's motivation to use the PackagePal application stems from her desire to continue her book-buying hobby. She appreciates the flexibility provided by the app's pickup points, as she may not always be available at home during delivery times. By using PackagePal, Joana can conveniently select a nearby pickup point, such as a local grocery store or laundry, where she can collect her book orders at a time that suits her schedule. This option enables her to ensure timely delivery of her books while maintaining the convenience and flexibility she values.

Scenario: Joana's Convenient Book Pickup

Joana is looking to purchase some new books to continue her hobby. She chooses an eStore (Readly) that is partener of PackagePal.

1. Browsing and Ordering: Joana opens the Readly app on her smartphone and starts browsing through the selection of books available. The user-friendly interface allows her to easily navigate through different genres and authors. After finding a few books that catch her interest, Joana adds them to her cart and proceeds to the checkout.
2. Selecting a Pickup Point: During the checkout process, Joana is prompted to select a pickup point for her order. She sees a list of associated collection points (ACPs), including neighborhood grocery stores and laundries. Joana chooses a grocery store located just a few blocks away from her apartment, as it's convenient for her to stop by after work.
3. Ready for Pickup: Once Joana's order is ready for pickup at the grocery store, she goes to the eStore and checks the status of her order. The page provides her with the necessary information, including the store's address and pickup instructions. Joana can then plan her visit to the store accordingly, ensuring that she can pick up her books at her convenience.
4. Smooth Pickup Experience: Joana arrives at the grocery store and shows the package code, name and email to the store staff, so they can check her identity. They quickly locate her order, hand it over, and Joana verifies the contents. The entire pickup process is smooth and efficient, allowing Joana to leave with her new books in hand.

By using the PackagePal app, Joana enjoys a seamless and convenient book-buying experience. The ability to select a pickup point ensures that she can collect her orders at a location and time that suits her schedule, making her hobby of reading even more enjoyable.

deti universidade de aveiro
departamento de eletrónica,
telecomunicações e informática

Persona 2: Filipa - The Dedicated Collaborator

Background: Filipa is a 35-year-old employee at a small grocery store. She is responsible for processing customer orders and ensuring efficient product delivery. Filipa is an organized and dedicated individual who values customer satisfaction and service quality. She works closely with the company's logistics team to ensure that all orders are carefully prepared and delivered to customers on time.

Motivation: Filipa's motivation for using the PackagePal application is to enhance the customer experience and increase foot traffic to the grocery store. By utilizing the app, Filipa aims to provide a seamless and well-informed delivery process that encourages customers to visit the store for order pickup. The convenience of PackagePal's features, such as selecting pickup points, helps Filipa attract more customers to the physical store, creating opportunities for upselling and providing a personalized experience.

Scenario: Filipa's Efficient Order Checkout

Filipa, an employee at the small grocery store, leverages the PackagePal app to streamline the order fulfillment process and provide a seamless delivery experience.

1. Customer Notification: The customer checks the order status and sees that their order is ready for pickup. The page includes the pickup location, instructions, and operating hours of the grocery store. Filipa appreciates that the app keeps the customer well-informed, ensuring a smooth and convenient pickup experience.
2. Confirmation and Delivery Tracking: As the customer collects their order from the pickup point, the PackagePal app allows Filipa to mark the order as delivered. Filipa also uses the app to track the delivery status and ensure that the order has been handed over to the customer.
3. Efficient Management: Throughout the process, Filipa utilizes the PackagePal app to efficiently manage multiple orders. The app provides a centralized platform where she can easily access order details. This streamlined management system allows Filipa to ensure a high level of service and satisfaction for both the customers.

By using the PackagePal app, Filipa optimizes the checkout process and facilitates a well-informed and efficient delivery experience for customers. The app's features, such as selecting pickup points, enable Filipa to improve customer satisfaction and contribute to the success of the grocery store.

Persona 3: Pedro - The Admin Mastermind

Background: Pedro is 40, administrator of the PackagePal platform, responsible for managing all pickup points and eStores associated with the platform. He is a detail-oriented and proactive individual with a strong focus on optimizing operational efficiency. He values data-driven decision-making and believes that statistics and insights are crucial for improving the overall user experience of the app.

Motivation: As an administrator of PackagePal, Pedro is motivated to use the application to enhance the overall user experience. He aims to efficiently manage all pickup points and associated eStores, ensuring a comprehensive view of operations. Additionally, Pedro values the insights and statistics provided by the application as they allow him to identify areas for improvement and make data-driven decisions to optimize the platform. By utilizing PackagePal, Pedro seeks to deliver an exceptional customer experience, driving operational efficiency, and offering enhanced features based on user feedback and statistical analysis.

Scenario: Pedro's Admin Insights

Pedro, an administrator of the PackagePal platform, is using the PackagePal application's admin dashboard to access valuable insights and make data-driven decisions to improve the app's service.

1. Accessing App Statistics: Pedro logs into the PackagePal admin dashboard and goes to the statistics section. He can see a comprehensive overview of various app statistics, including user orders, order volumes, shops and pickup points. The data is presented in easy-to-understand charts and graphs, allowing Pedro to quickly grasp the key trends and patterns.
2. Identifying Areas for Improvement: Pedro analyzes the statistics to identify areas where the app's service can be improved. For example, he notices that there aren't too many pickup points in a certain region. By pinpointing this issue, Pedro can focus on optimizing the logistics and delivery by trying to have more pickup points in that area.
3. Making Data-Driven Decisions: Armed with the insights gathered from the statistics, Pedro starts formulating data-driven decisions. For instance, he decides to allocate additional resources to high-demand areas to improve delivery times.
4. Collaborating with the Development Team: Pedro collaborates closely with the development team, sharing the insights and decisions derived from the statistics. They discuss potential solutions and improvements to be implemented in future updates of the PackagePal app. Pedro ensures that the team is aligned with the app's objectives and works towards enhancing the overall user experience.

By leveraging the PackagePal admin dashboard and its powerful statistics, Pedro can make informed decisions and drive continuous improvements in the app's service. His goal is to provide a seamless and reliable experience for both users and employees, optimizing the platform's operations and increasing customer satisfaction.

2.3 Project epics and priorities

Each spring we defined a set of tasks, to which we gave a level of priority. This priority is defined by Jira's (the platform used for organizing and managing tasks) standards, which are highest, high, medium, low and lowest.

Spring 1 (20/04 - 04/05)

Task	Priority
Define the product concept	High
Outline the architecture	Hight
Team resources setup	Medium

Spring 2 (04/05 - 11/05)

Task	Priority
Make functional prototype for the main platform	Medium
Make functional prototype for the eStore	Low
Define the SQE tools and practices	High
Product specification report (draft version)	Low

Spring 3 (11/05 - 18/05)

Task	Priority
A few core stories detailed and implemented	High
QA Manual	Low

Task	Priority
CI Pipeline	Medium
User stories to implement	Medium

Spring 4 (18/05 - 25/05)

Task	Priority
Product increment with a couple of user stories	High
Comprehensive API	High
Set up the CD pipeline	Medium
User stories to implement	Medium

String 5 (25/05 - 1/06)

Task	Priority
Stabilize the Minimal Viable Product (MVP)	High
All deployments available in the server	High
Product specification report (final version)	HIgh
Non-functional tests and systems observability	Medium
Comprehensive REST API	High
Quality dashboard	Medium

3   Domain model

For this project one domain model was elaborated and supported with a H2 database. Here we have a package that can be managed by an admin. Each order has a status (PENDING, IN TRANSIT, DELIVERED, COLLECT), a pickupPoint where it will be delivered to the eStore where the order was placed.
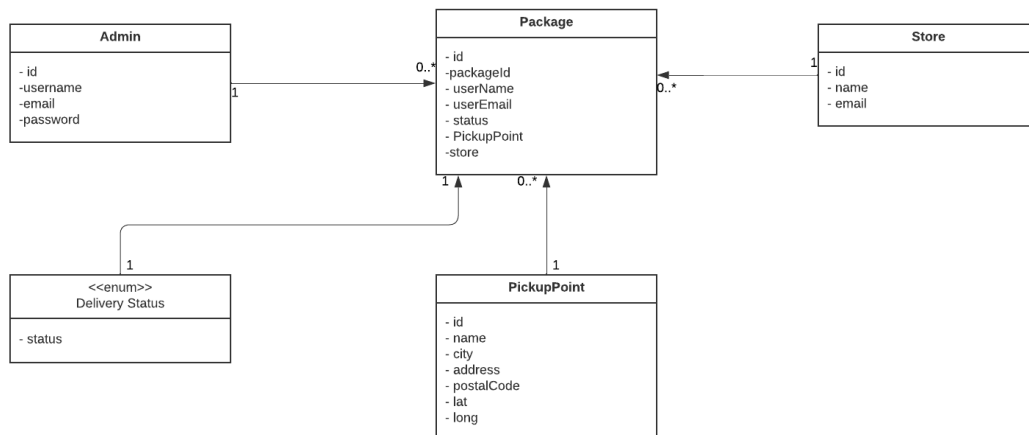

Fig1 - UML class diagram

4 Architecture notebook

4.1 Key requirements and constraints

Our architecture was designed with the following requirements in mind:

- Independence and Scalability: The delivery system should be designed to be independent and capable of hosting multiple eStores with different products. It should be able to handle concurrent requests from these eStores efficiently. The architecture should be scalable to accommodate the growing number of eStores and their delivery requirements.
- Managing Platform: The system requires a comprehensive managing platform that allows control over packages, stores and pickup points. This platform should provide functionalities to manage inventory, monitor delivery status, and generate statistics and reports. It should offer a user-friendly interface for efficient management of the system.
- Pickup Point Management: A separate managing platform is needed to handle the pickup points. This platform should enable the pickup point administrators to control the incoming and outgoing orders, and manage the overall operation of pickup points. It should provide real-time updates on the orders..
- Deployment on Virtual Machine: The system should be deployed on a virtual machine to ensure flexibility, scalability, and ease of maintenance. The virtual machine environment allows for efficient resource utilization, easy deployment, and management of the system components.
- Authentication System: The system should incorporate a robust authentication system to ensure secure access and protect sensitive data. Managing platform administrators and pickup point administrators, should be authenticated with appropriate credentials before accessing the system. This helps to maintain the integrity and confidentiality of the system and its data.

4.2 Architectural view

In our architecture there are two main models, PackagePal and Readly, where the first works independently, while the second depends on the first to work.
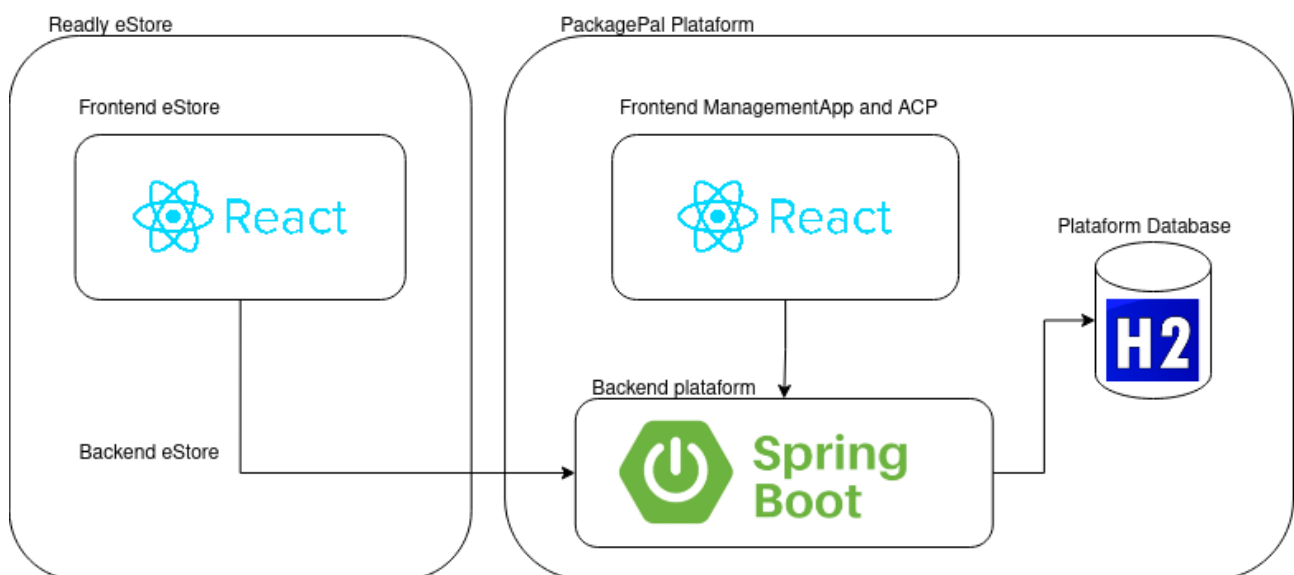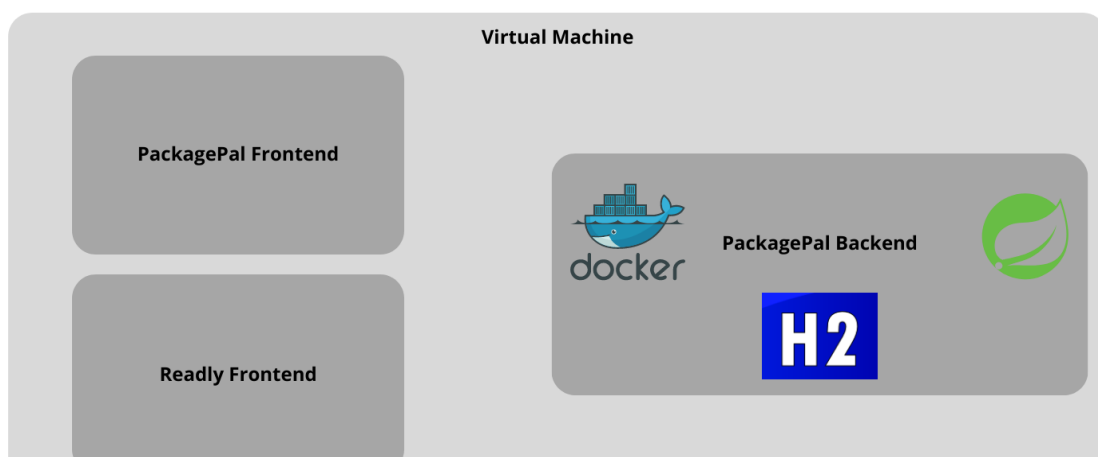

Fig2 - Architecture diagram

4.3 Deployment architecture

11

Fig3 - Deployment Architecture diagram

Our architecture only features a container with docker, but the frontends and backend can be accessed from the following links:
- Readly Frontend: http://192.168.160.234:3000/
- PackagePal Frontend: http://192.168.160.234:3001/
- PackagePal Backend: http://192.168.160.234:8080/api/v1/

5   API for developers

Our API is responsible for providing features and functionality to manage orders and pickup points. It allows users to perform operations such as creating a new order, updating the status of an order, searching for information about collection points and much more.

What developers should know:
- The API follows a RESTful architecture.
- We use JSON as the data format for requests and responses.
- The endpoints are structured according to the resources and operations available.
- Authentication is required to access API functionalities.
- Requests must be made through HTTPS.

API Documentation: https://documenter.getpostman.com/view/24105095/2s93mBxeyU

6   References and resources

What is a deployment diagram?
https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-deployment-diagram/

Generating API documentation
https://learning.postman.com/docs/publishing-your-api/documenting-your-api/

ChatGPT (for better text presentation)