deti universidade de aveiro departamento de eletrónica, telecomunicações e informática

TQS: Product specification report

Diogo Oliveira Magalhães [102470]
Rafael Fernandes Gonçalves [102534]
Leonardo Almeida [102536]
Pedro Henrique Figueiredo Rodrigues [102778]
v2023-06-01

1 Introduction **1**
1.1 Overview of the project 1
1.2 Limitations 2
2 Product concept **2**
2.1 Vision statement 2
2.2 Personas and scenarios 5
2.3 Project epics and priorities 8
3 Domain model **9**
4 Architecture notebook **9**
4.1 Key requirements and constraints 9
4.2 Architectural view 10
4.3 Deployment architecture 12
5 API for developers **13**
6 References and resources **13**

1 Introduction

1.1 Overview of the project

In the scope of "Teste e Qualidade de Software" course, the main objective of this project was to develop a medium-sized project, where we could apply the knowledge acquired in the course, regarding software quality assurance and software enterprise architecture patterns. In order to make the several projects comparable across the different groups, all the projects were expected to follow the same guidelines, which are described in the following sections.

The application developed for this project is called "PickAPoint" and it consists of a network of Pickup Points as a service, where independent businesses can adhere to the network and offer their physical places as collection points to receive orders from customers.

1.2 Limitations

Within the context of this project, we successfully incorporated all the mandatory features outlined in the requirements specification. However, due to time limitations, we were unable to implement certain additional functionalities that we had intended to include. These functionalities are:

- Integration of the physical lockers with the main system
- B2B communication with courier companies

Throughout the deployment phase, we encountered difficulties in deploying the backend applications on the university's Virtual Machine. This was primarily due to network restrictions that hindered the proper implementation of the intended CI/CD pipeline.

2 Product concept

2.1 Vision statement

In recent years, the rapid growth of e-commerce has revolutionized the way we shop. With just a few clicks, we can order products from the comfort of our homes and have them delivered right to our doorstep. However, this convenience comes with its own set of challenges, one of which is the increasing difficulty of delivering packages to customers who are not present at home to receive them. With customers frequently absent, delivery service providers face the dilemma of making repeated delivery attempts. These unsuccessful attempts not only increase operational costs for companies but also cause frustration for customers who have to rearrange their schedules to accommodate subsequent delivery attempts. In some cases, multiple failed deliveries can even lead to packages being returned to the seller, adding further inconvenience to customers.

PickAPoint is a service that aims to solve this problem by providing a network of Pickup Points that the customer can choose when placing an order. These Pickup Points are physical locations, such as stores or even gas stations, where the customer can pick up their order at a time that is convenient for them. This service benefits both customers and businesses. Additionally, the service also provides an API that can be integrated with e-commerce websites, allowing its customers to choose a Pickup Point when placing an order, and track their status progress.

In terms of actors, the system has four main types of users, namely Pickup Points, e-Stores, system administrator, and customers.

From a Pickup Point perspective, the system allows businesses to register their physical places as Pickup Points so that they can receive orders from customers of e-commerce websites. The system also allows Pickup Points to manage their orders, by marking them as delivered or collected by the customer. Additionally, Pickup Points can also monitor the different types of packages, including stored packages, expected packages and cancelled packages.

From an e-Store point of view, the system allows e-Stores to get all the Pickup Points registered in the network. The system also allows e-Stores to create orders, by specifying the customer, the Pickup Point and the package details. Additionally, e-Stores can also track the order status progress, by checking if the order is still pending, delivered or collected.

From an administrator's perspective, the system allows accepting or rejecting the applications of businesses that want to become Pickup Points, and monitor all the pickup points that form the network. Additionally, the administrator can also monitor all the orders and packages that are the responsibility of the infrastructure.

Finally, every time a package arrives at the Pickup Point, the system sends an SMS with the package ID and a security token to the client. This token is used by the customer to collect the package at the Pickup Point.

For a clearer view of our system, a UML Use Case Diagram was developed, where the system's functionalities are presented. This diagram is presented in the following figure.

Figure 1: PickAPoint Use Case Diagram

To facilitate and organize the development process, and considering that we were following an agile methodology, we used the concept of "User Story" as a way to describe the system's functionalities, and as a unit for planning. In the following tables, we present the user stories that were defined for this project.

Pickup Point

Use Case	**User Stories**
Join to the platform as Pickup Point	**As** a business, **I want to** be a Pickup Point **so that** I can receive orders and store them until the client picks them up.
Login	**As** a Pickup Point, **I want to** log into the platform **so that** I can have access to Pickup Point functionalities.

Register received package	**As** a Pickup Point, **I want to** register a received package **so that** the system notifies the client that its package can be picked up.
Register package collection	**As** a Pick-up Point, **I want to** register a package collection **so that** the package is marked as collected by the client.
Monitor stored, expected and cancelled packages	**As** a Pick-up Point, **I want to** monitor stored, expected and cancelled packages **so that** I can manage how many packages will arrive, how many clients will pick their packages and how many orders will not arrive.

e-Store

Use Case	**User Stories**
Get all available Pickup Points	**As** an E-Store, **I want to** get all available pick-up points **so that** I can list all available collection points that the user can choose when placing a new order.
Make a delivery request to a specific Pickup Point	**As** an E-Store, **I want to** make a delivery request to a specific pick-up point **so that** the client can pick its package up in that collection point.
Track order status progress	**As** an E-Store, **I want to** track the progress of an order **so that** the client knows the current situation of its order at any time.
Cancel an order	**As** an E-Store, **I want to** be able to cancel an order **so that** the respective Pickup Points knows that that package will not be delivered.

Administrator

Use Case	**User Stories**
Login/Logout	**As** an Administrator, **I want to** log into the platform **so that** I can have access to administrator functionalities.
Accept/Deny partner join requests	**As** an Administrator, **I want to** accept or deny Pick-up Point joins requests **so that** clients have one more, or not, collection point to pick up their packages.
Check operational statistics	**As** an Administrator, **I want to** check operational statistics, including number of packages and pickup points **so that** I can have a general understanding of service.
Check packages status	**As** an Administrator, **I want to** check packages status **so that** I can have a general understanding of the current situation of the delivery process.

Client

Use Case	User Stories
Receive notification when the package arrives at the pick-up point	**As** a client, **I want to** receive a SMS notification when the package arrives at the pick-up point **so that** I can go there and collect it.

2.2 Personas and scenarios

When following an agile approach, gathering requirements usually entails creating personas. These consist of fictitious entities that represent a certain group of end users of the system, appearing accompanied by realistic characteristics, context, motivations and objectives.

In the scope of this project, we defined four personas, each representing a different type of user:

1. Administrator
2. Pickup Point
3. eStore
4. Client

Administrator

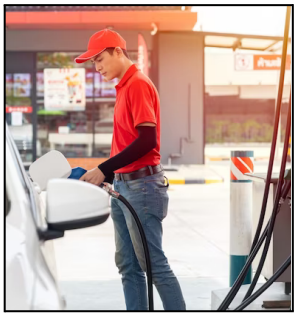
Name	David Miller
Age	58
Sex	Masculine
Job	Administrator of a collection points network
Address	Av. João Jacinto de Magalhães, Aveiro

Context:
David Miller is responsible for managing the operations of a medium size network of collection points. He has more than 10 years of experience in the Logistics sector and stands out for his organization and attention to detail. As he values efficiency and customer satisfaction, he is constantly looking to improve the company's processes.

Motivation:
David Miller would like to have more control and information about the new orders that come from online stores. Additionally, he would like to monitor all the Pickup Points that make part of his medium size network. To facilitate the process of adding a new Business as a Pickup Point, David considered it useful to have a place where he could just accept or deny new applications for Pickup Point.

Pickup Point

Name	Paul Williams
Age	35
Sex	Masculine
Job	Owner of a Gas Station
Address	Avenida Principal, Lisbon



Context:
Paul Williams is an experienced business owner, with over six years of expertise in the gas station industry. He is the proud owner of a Gas Station situated on Avenida Principal in Lisbon.

Motivation:
With the declining number of combustion engine cars and the exponential rise of electric vehicles, Paul Williams recognizes the need to adapt his business to stay relevant. To ensure the survival of his gas station, Paul aims to transform it into a Pickup Point at PickAPoint. By doing so, customers who select his gas station as their preferred collection point can conveniently purchase the products he offers. This strategic move allows Paul to capitalise on the increasing popularity of electric cars while maintaining a valuable presence in the evolving market.

eStore

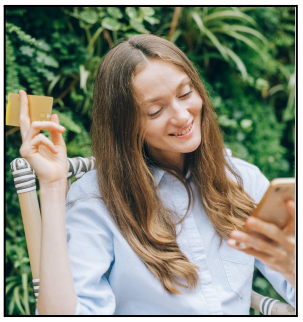
Name	Mary Jones
Age	30
Sex	Feminine
Job	Owner of a online clothes store
Address	Rua das Flores, Porto

Context:
Since 2014, Mary Jones has been the proud owner of an online clothing store. Initially, she contemplated establishing a physical clothing store. However, recognizing the rapid growth and popularity of the internet, Mary made the strategic decision to exclusively launch an online store instead. This choice allowed her to leverage the vast reach and convenience offered by the digital platform, enabling her to tap into a wider customer base and adapt to the evolving trends in consumer behavior.

Motivation:
Mary Jones has observed that a certain portion of people avoid purchasing products online because they are unable to receive them at home. To enhance the value of her company, she intends to enhance its eStore by incorporating a feature that allows users to select Pickup Points as alternative delivery locations. This way, customers will have the flexibility to receive their packages, regardless of whether they are present at home or not. By implementing this update, Mary aims to attract a larger customer base and increase the overall appeal of her company.

Client

Name	Anna Johnson
Age	37
Gender	Feminine
Job	Professor
Address	Avenida da Liberdade, Lisbon



Context:
Anna Johnson, a high-school professor, has a packed schedule with classes throughout the entire week, starting from 8 AM and lasting until 6 PM. Due to her busy timetable, she has no opportunity to go home during the day and is constantly occupied with teaching responsibilities.

Motivation:
Anna, who spends most of her time at school, faces a challenge when it comes to buying products online because she is not at home when the courier delivers her packages. Anna wishes for an alternative option while shopping online: to have her orders delivered to a pickup point. This way, her packages wouldn't be returned, and she could conveniently collect them at her preferred time. Anna also would like to receive an SMS whenever her packages are delivered at the chosen Pickup Point.

Main Scenarios

Scenarios relate concretely how a system will be used, which can help stakeholders better understand end-user needs and expectations. Additionally, they allow identifying potential problems or gaps in system functionality, serving as a basis for validating and prioritizing requirements.

David Miller - **Administrator**

Accept/Deny partner join requests
Recently, several businesses submitted applications to become a partner of PickAPoint. Given this, David Miller:

1. Logs into the system
2. Goes to applications page
3. Checks the information of each business that want to be a Pickup Point
4. Accepts all the applications that he thinks that could give value to its network or denies them

Check packages status
In order to David Miller recognize the current state of each package that his network is responsible of, he:

1. Logs into the system
2. Goes to packages page
3. Checks the state of each package

Paul Williams - Pickup Point

Join to the platform as Pickup Point

Paul Williams want to become a pickup point, so he:

1. Goes to PickAPoint website
2. Goes to registration page
3. Fills the registration form where indicates the business name, email, contact and location
4. Sends the application

Register received package

After Paul Williams' application is accepted he can register in the system every package that arrives. To do this he:

1. Logs into the system
2. Click on the “Register Package” button
3. Enter the package ID
4. Click on the “Register” button

Register package collection

When a client comes to collect his package, Paul Williams':

1. Logs into the system
2. Click on the “Register Pick Up” button
3. Enter the package ID
4. Enter the security token sent to client via SMS
5. Click on the “Set as Collected” button

Anna Johnson - Client

Receive notification when the package arrives at the pick-up point

A few days after the purchase, Anna receives a phone message with its package ID and security token. She drives to the chosen Pickup Point and provides the package ID and security token received. If the ID and the token are correct, she collects its package successfully.

2.3 Project epics and priorities

The development of this project was divided in 4 sprints, each of one with multiple user stories and tasks to complete and major goals with high priority to achieve, these major goals were the following:

- **Sprint 1**: Define the product concept, user stories and tasks definition, system architecture and domain model.
- **Sprint 2**: Prepare a SQE strategy, including the definition of done and the acceptance criteria for each user story, implement a CI pipeline with quality gates and prototype the core user stories for PickAPoint and e-Store.
- **Sprint 3**: Implement the user stories for PickAPoint and e-Store with the CI pipeline and quality gates.
- **Sprint 4**: Implement CD pipeline, deploy the application to production and finish the documentation.

These major goals were the main project epics and to achieve them we needed to complete all the user stories and tasks associated with them, which are specified in the backlog.

3 Domain model

At the beginning of the project, we started by gathering the requirements of the system and defining the main entities and their relationships.

We decided to create the system components based on the different types of users, namely Pickup Points, e-Stores, administrators and customers. Each entity has its own controller and service classes. Both AdminService and PartnerService depend on the MessageSender and TokenGenerator classes, which are responsible for sending SMS messages and generating security tokens, respectively. Additionally, we also created DTOs (Data Transfer Objects), which are used to transfer data between the presentation layer and the controller classes.

To better understand the domain of the system, we developed a domain model, where only the main entities and their relationships are presented. This model is presented in the following figure.
Note: DTOs are not represented in the diagram for simplicity reasons.

Figure 2: PickAPoint Class Diagram

4 Architecture notebook

4.1 Key requirements and constraints

In this project, there are some key requirements and system constraints that have a slight impact on the architecture and design of the system. These requirements and constraints are:

- The administrator and the pickup points can only have access to the system upon authentication and authorization.

- There are different types of users with different functionalities and motivations.
- The system must guarantee the persistence of the data.
- All the interactions between the system and the e-Stores are done through B2B (Business-to-Business) communication, using a REST API.
- Whenever a package is delivered to the Pickup Point, the system must generate a security token that the customer must present when collecting his order.
- The system must be able to send SMS messages to the customers, with the package ID and a security token, every time a package arrives at the pickup point.

4.2 Architectural view

In order to fulfil all the requirements specified in the previous sections, it is extremely important to design an adequate and realistic system architecture.

In the scope of this project, we concluded that the most suitable architecture was a Layered architecture, also known as N-tier architecture. With this type of architecture, the different components that make up the system are organised into horizontal layers, each one of them with a specific role. The main advantage of this design is that it allows us to separate the different concerns of the system, which makes it easier to maintain and extend.

The proposed architecture is composed of 5 layers, namely:

- Presentation Layer
  This layer is responsible for presenting the information to both the administrator and the pickup points. It is also responsible for receiving the information from the users and sending it to the business layer. Regarding the technologies and frameworks, we developed a web application using the **Svelte** framework, following a component-based approach. To facilitate and speed up the development process, we used some CSS libraries and component libraries, including **Tailwind CSS** and **Flowbite Svelte**. Additionally, we used the build tool **Vite**.
- Controller/Boundary Layer
  This layer is responsible for receiving the information from the presentation layer, and sending it to the service layer, and vice-versa. Regarding the technologies and frameworks, we developed a REST API using the **Spring Boot framework**, following a controller-based approach. Taking into account that the system has different types of end users, we decided to create a controller for each type of user, namely Partner Controller (for Pickup Point communication), eStore Controller and Administrator Controller. Additionally, we also created an authentication controller, which is responsible for authenticating the users.
- Service Layer
  This layer is responsible for receiving the information from the controller/boundary layer, and performing the necessary operations to satisfy the user's request. It is also responsible for receiving the information from the data access layer, and sending it to the controller/boundary layer. Following the same approach as the controller/boundary layer, we created a service for each type of user. To send an SMS everytime a new package is delivered at the pickup point, we used the **Twilio API**.
- Data Access Layer
  This layer is responsible for receiving the information from the service layer, and performing

the necessary operations at the database level. Regarding the technologies and frameworks, we used the **Spring Data JPA**, which allows us to easily implement the repository pattern.

- Database Layer
  This layer is responsible for storing the data. Regarding the technologies and frameworks, we used the **PostgreSQL** database management system.

The following Figure illustrates a component diagram that visually presents the different components that compound the system and their relationship.

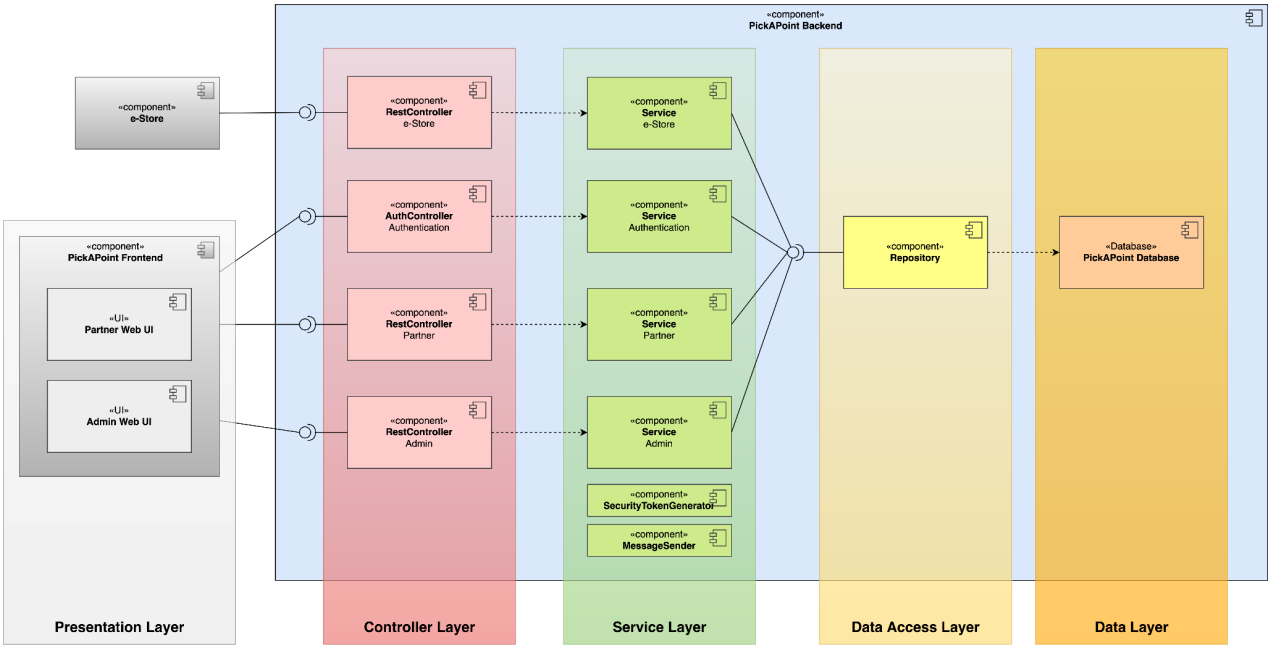


Figure 3: PickAPoint Architecture - Component Diagram

The following Figure illustrates the main technologies and frameworks mentioned before.

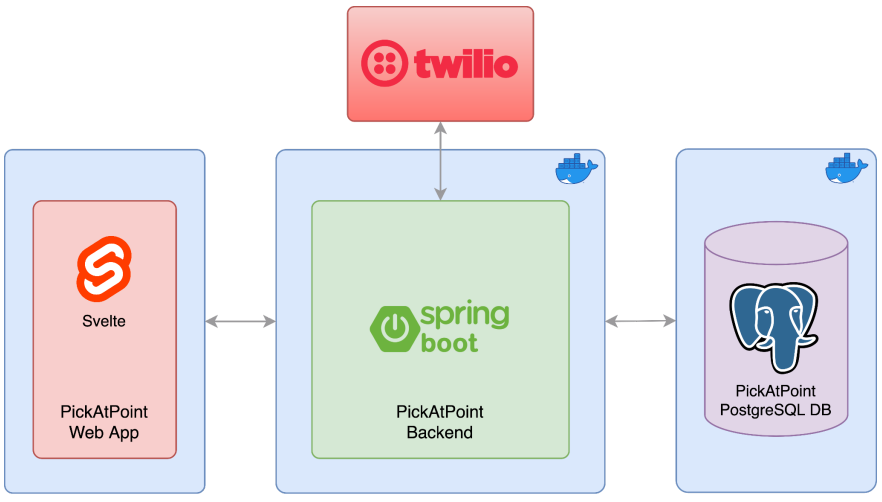


Figure 4: Technologies Stack

Even though the eStore does not belong to the PickAPoint architecture, in this project we used the exact same technologies to develop the eStore platform, except Twilio API.

4.3 Deployment architecture

To make the platform publicly available, we followed the following strategy:

- Frontend
  Both eStore and PickAPoint frontends are deployed on the Google Compute Engine Virtual Machine.

- Backend
  Both eStore and PickAPoint backends (including the databases) are also deployed on the same virtual machine as the web applications. However, to achieve portability, scalability and security, we used Docker to containerize each backend module.

The following figure presents the deployment architecture of the system, including the eStore module.

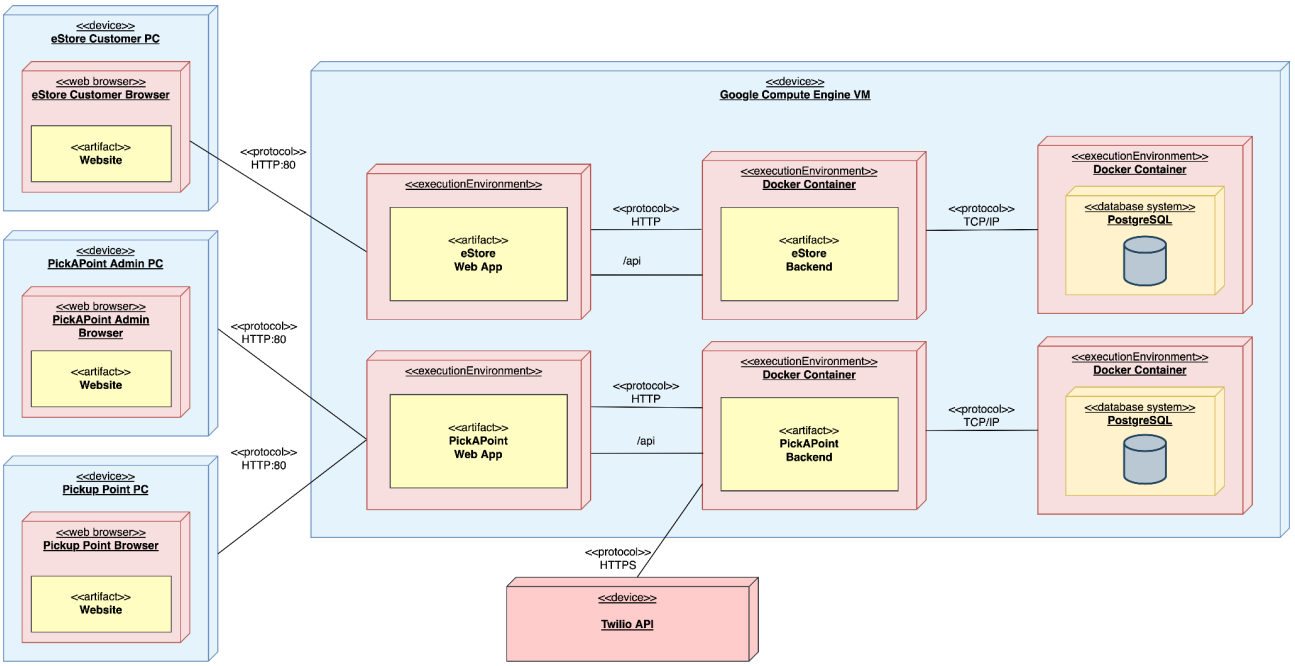


Figure 5: PickAPoint and eStore deployment diagram

5 API for developers

As explained above, we have developed two REST APIs, one for PrintPlate e-commerce and another for the main PickAPoint services. Although the latter was initially designed with our two web applications as the only clients, it can easily be extended to other clients, such as mobile applications, to get all the available pick-up points, track order status, register received or collected packages, and much more.

To document their endpoints, we have used Postman.

DPP_Backend Documentation: https://documenter.getpostman.com/view/27497156/2s93sW8vAK
eStore_Backend Documentation: https://documenter.getpostman.com/view/27497156/2s93sW8vAF

6 References and resources

https://kit.svelte.dev/docs
https://www.twilio.com/en-us
https://github.com/twilio/twilio-java
https://tailwindcss.com/docs
https://www.chartjs.org/
https://flowbite-svelte.com/docs
https://app.diagrams.net/
https://www.postman.com/