

# TQS: Product specification report

**Gonçalo Ferreira [107853], João Carlos Santos [110555], Matilde Teixeira [108193], Sara Almeida [108796]**

V2023/04/24

<b>1</b>	<b>Introduction .....</b>	<b>1</b>
1.1	Overview of the project.....	1
1.2	Limitations.....	2
<b>2</b>	<b>Product concept .....</b>	<b>2</b>
2.1	Vision statement.....	2
2.2	Personas .....	3
2.3	Main scenarios .....	2
2.4	Project epics and priorities.....	5
<b>3</b>	<b>Domain model.....</b>	<b>5</b>
<b>4</b>	<b>Architecture notebook.....</b>	<b>6</b>
4.1	Key requirements and constrains.....	6
4.2	Architeturual view .....	6
4.3	Deployment architerture .....	7
<b>5</b>	<b>API for developers.....</b>	<b>7</b>
<b>6</b>	<b>References and resources.....</b>	<b>7</b>

## 1 Introduction

### 1.1 Overview of the project

This is a project developed in the context of TQS course (Testes e Qualidade de Software), where the objectives of the project assignment were the implementation of a software quality assurance strategy and the implementation of continuous testing, integration and delivery. To achieve these aspects, the proposal was to build a Minimal Viable Product (MVP) system, with core functionality, architecture and implementation that is coherent with the previously mentioned principles.

With this in mind, our aim is to build the DETICafe, a system for a café service for university students, mainly DETI ones. The uniqueness of the system centralizes in the order process, where when placing orders, the client does so via a kiosk and then wait for the order's number to appear on another dashboard, where he can pick up his order.

To make this happen the system will have 4 different interfaces: one for the clients to place order (kiosk one), one for the cook to see the order's products and ingredients and select when it's ready for the waiter to pick it up, one for the waiter to sinalize the order's dashboard that the order is ready for customer deliver and one for the clients to know if their order's are preparing or ready for them to pick them up.

This service will be a digital transformation of a café and will be very helpful in the management of a café by organizing and streamlining the preparation and delivery of orders. It will make students life easier in the

way they can previously order, for example, their lunches in an easy way but also help bar workers by giving them a system that organizes the orders for them and ends up enshortening waiting lines.

## **1.2 Limitations**

Our system will feature a kiosk interface allowing the client to select the order's items. Once the order is placed, it appears on the kitchen dashboard in a queue-like view, providing the order's ingredients for the cook to prepare. When the order is ready, the cook will signalize that it is ready for pick up. The waiter responsible for the product's pick-up will signal on the platform that the product is ready. By doing this, the corresponding number of order will appear in the order's dashboard and the client will know that he needs to pick it up.

# **2 Product concept and requirements**

## **2.1 Vision statement**

The aim of our system is to provide an easier, faster and contactless delivery for students in university, specifically DETI students, who don't always like communicating with others. Our system will help digitalize this project and provide a contactless delivery, which means, the customer will rarely have to talk with the waiter.

Our system will function in a way that the customer makes the order at the kiosk interface, and when they pay, the order is sent to the kitchen interface. There, the cook can see the order's specific requests and possible alterations in order to cook it. When the status is done, the cook sends a signal to the waiter dashboard for him to pick it up. When the order is at the pickup spot, the waiter changes the status in the order display interface, so that the customer can pick it up.

The main difference from other similar well known products is the fact that our system will make life easier, not only for the bar workers but especially for DETI students who don't have the time to wait in waiting line for their order. Using our application they will be able to previously place their order and only get them when they are ready for pick up. Also during the ordering process, this is separated into different sections, enabling a customer to order from a cafeteria, pastery or even restaurant, etc., making the process easier and quicker.

## 2.2 Personas and scenarios



### D. Alberta, 58, Waitress at DETICafe

Fifty-eight-year-old D.Alberta is a very nice lady that works at the DETICafe and is very cherished by all the DETI students. She is a grandmother of 2 young adults, both in college, so she is very familiar with this age, she loves all her costumers and her job.

D.Alberta is still a very agile and active lady but sometimes the lines of students waiting for their orders are big and it is a bit difficult to manage all the orders and deliver them tho whom indeed ordered them.

**Motivation:** D.Alberta would like a very simple and intuitive system to help her manage the waiting lines, so she would keep track of the orders that were already delivered and those that were not yet and are still being prepared.



### Luís, 29, Cook at DETICafe

Twenty-nine-year-old Luís is one of D.Alberta's sons. Since he was little he used to go with his mom to work during school breaks and always dreamed of working with her and helping her at DETICafe. Therefore, when he grew up he decided to take a cooking course so he would be in charge of the kitchen at DETICafe and make the best menu for the students.

For many years, D.Alberta would write the orders in a paper and take to the kitchen. However, in times where there are a lot of costumers coming to the bar at the same time this method would turn out to be not so efficient. Sometimes

the papers would even get lost or unordered.

Besides, sometimes students ask for personalized menus, without some of the ingredients or more of another some, and the method they used would sometimes make him mistake those.

Luís and D.Alberta have also been looking for a way to make life easier to students and shorten waiting lines by having a system where these could order online.

**Motivation:** Luís needs a more efficient way to keep track of the orders, specially the order in which they were made and those who have customized ingredients requests. He would also like to have a view of which ingredients each order requires so it would be easier for him to quicker serve his costumers.



### **Pedro Santos, 23, LECI student**

Pedro is a determined student and focused on his field of study. He was always interested in electronics and computing from an early age and decided to pursue a career in Computer and Informatics Engineering. He is a quiet and reserved person, but very dedicated to his studies and academic projects.

Despite his busy schedule, Pedro values moments of pause and relaxation during the day. He sees the launch break as an opportunity to recharge and socialize with his course mates. The DETICafe is the ideal place for Pedro, as it offers a variety of quick and practical snack options, perfect for breaks between classes.

At the DETICafe, Pedro usually meets other students on his course and exchanges experiences about the projects they are developing. He also takes advantage of the moment to review class material and discuss doubts with colleagues. Furthermore, the relaxed atmosphere of the DETI bar allows Pedro to relax and unwind before returning to academic activities.

**Motivation:** Pedro has a short break between classes and needs a fast way to choose and buy some food, he doesn't want to wait in a line since he hasn't much time.



### **Ana, 21, LEI student**

Ana is a dedicated and enthusiastic student, passionate about technology and programming since a young age. He is an extroverted and sociable person, always willing to help colleagues with projects and group work. Ana highly values education and is committed to achieving her goal of becoming a successful software engineer.

In addition to her studies, Ana also enjoys maintaining a balanced lifestyle. She understands the importance of taking breaks and resting during the day, and one

of these breaks is lunchtime.

The DETI (Department of Electronics, Telecommunications, and Informatics) bar is the ideal place for Ana, as it offers a variety of quick and healthy meal options at affordable prices. It's a place where she can meet friends, relax, and recharge for the rest of the day of studying.

**Motivation:** Ana wants to have lunch at the DETICafe and she needs a simple way to see the menu, select what to eat and make the request. Ana also needs to know when food is ready to pick it up.

## 2.3 Project epics and priorities

### Epics and respective user stories

#### 1- Client Ordering Interface (Kiosk)

- As a customer, I want to select the section I want to order from, so that it is easier to see the available products from there.
- As a customer, I want to explore all the items available at the DetiCafe in the initial page, so that I can choose what I want for lunch.
- As a customer, I want to search a specific product so that I can order faster and in an easier way.
- As a customer, I want to customize my sandwich, so that I can have a personalized sandwich without eggs since I'm allergic.
- As a customer, I want to review my order before finalizing it, so that I am sure that I'm ordering everything right.
- As a customer, I want to receive my order number, so that I can keep track of my order in the order display interface.

#### 2- Kitchen Dashboard

- As a cook, I want to view incoming orders from the kiosk interface, so that I can start preparing them as soon as I can in the order they were placed.
- As a cook, I want to see the ingredients needed to prepare the incoming order, so that I don't forget anything and deliver the best product to my client.
- As a cook, I want to see if the customer personalized the ingredient of the order, so that I don't mistake the order and deliver the wanted product to my client.
- As a cook, I want to mark orders as ready, so that the waiter knows that they can pick them up.

#### 3- Waiter Interface

- As a waiter, I want to see orders that are ready for pickup in the kitchen, so that I can go get them from there.
- As a waiter, I want to change the status of an order to ready, so that I inform the customer that their order is ready for pickup.
- As a waiter, I want to view organizedly the orders that are ready and the ones that aren't done yet, so that I can see which orders do I need to be ready to pick up.
- As a waiter I want to efficiently change the status of multiple orders, so that the waiting line doesn't get upset.

#### 4- Order Display Interface

- As a customer, I want to be able to keep track of my order to see if it's still preparing, so that I can quickly go to the bathroom and not miss it.
- As a customer, I want to see when my order is ready for pick up, so that I can pick it up and enjoy it faster.

## 3 Domain model

<which information concepts will be managed in this domain? How are they related?>

<use a logical model (UML classes) to explain the concepts of the domain and their attributes>

## 4 Architecture notebook

### 4.1 Key requirements and constraints

<Identify issues that will drive the choices for the architecture such as: Will the system be driven by complex deployment concerns, adapting to legacy systems, or performance issues? Does it need to be robust for long-term maintenance?

Identify critical issues that must be addressed by the architecture, such as: Are there hardware dependencies that should be isolated from the rest of the system? Does the system need to function efficiently under unusual conditions? Are there integrations with external systems? Is the system to be offered in different user-interfacing platforms (web, mobile devices, big screens,...)?

E.g.: (the references cited in [XX ] would be hypothetical links to previous specification documents/deliverables )

There are some key requirements and system constraints that have a significant bearing on the architecture. They are:

- The existing legacy Course Catalog System at Wylie College must be accessed to retrieve all course information for the current semester. The C-Registration System must support the data formats and DBMS of the legacy Course Catalog System [E2].
- The existing legacy Billing System at Wylie College must be interfaced with to support billing of students. This interface is defined in the Course Billing Interface Specification [E1].
- All student, professor, and Registrar functionality must be available from both local campus PCs and remote PCs with internet dial up connections.
- The C-Registration System must ensure complete protection of data from unauthorized access. All remote accesses are subject to user identification and password control.
- The C-Registration System will be implemented as a client-server system. The client portion resides on PCs and the server portion must operate on the Wylie College UNIX Server. [E2]
- All performance and loading requirements, as stipulated in the Vision Document [E2] and the Supplementary Specification [15], must be taken into consideration as the architecture is being developed.>

### 4.2 Architecture view

→ Discuss architecture planned for the software solution.

→ include a diagram ( a package or block diagram)

→ explain how the identified modules will interact. Use sequence diagrams to clarify the interactions along time, when needed

→ discuss more advanced app design issues: integration with Internet-based external services, data synchronization strategy, distributed workflows, push notifications mechanism, distribution of updates to distributed devices, etc.>

### 4.3 Deployment architecture

[Explicar a organização prevista da solução em termos configuração de produção (*deployment*). Anotar, no diagrama, as tecnologias de implementação, e.g.: colo aro simbolo do PostgreSQL na Base de dados,...]

## 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>

[ Base URL: localhost:8080/weather ]

client Regular user of the weather forecast API

GET /now/{latitude},{longitude} get weather forecast of the current day for the given coordinates

GET /recent/{latitude},{longitude}/{days} get weather forecast of the next days starting from today until the given number of days for the given coordinates

GET /period/{latitude},{longitude}/{start},{end} get weather forecast of the given time period for the given coordinates

GET /cached get weather forecasts previously requested and still present in cache

## 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>