

# TQS: Product specification report

*Joana Amaral Gomes [104429], Artur Denderski [122282], Maria João Machado Sardinha [108756], Mateusz Kubiak [122385]*

v2024-05-06

<b>1.1 Overview of the project</b>	<b>2</b>
<b>1.2 Limitations</b>	<b>2</b>
<b>2.1 Vision statement</b>	<b>3</b>
<b>2.2 Personas and scenarios</b>	<b>4</b>
<b>2.3 Project epics and priorities</b>	<b>6</b>
<b>4.1 Key requirements and constraints</b>	<b>9</b>
Functional Requirements	9
Non-Functional Requirements	9
System Constraints	9
<b>4.2 Architecture view</b>	<b>11</b>

# 1 Introduction

## 1.1 Overview of the project

In the realm of the TQS course, we blend theory with practice to develop practical skills in software testing and quality assurance. Our project aims to apply this learning by creating a digital platform for the restaurant industry using a Software Quality Engineering (SQE) strategy.

Our goal is to implement testing procedures and ensure software quality throughout the project. Our project emphasizes the significance of Software Quality Assurance (SQA) as a foundational element of software engineering. By defining and enforcing a comprehensive SQA strategy, we seek to cultivate a culture of quality consciousness throughout the software development lifecycle. We'll be adopting modern development practices like Continuous Testing, Continuous Integration, and Continuous Delivery, in order to iterate quickly and deliver value consistently.

Our platform, Pollos Hermanos, is designed to enhance customer satisfaction and streamline operations for various restaurants. It's user-friendly for both customers and staff alike.

Through this project, we aim to enhance our learning experience, our knowledge and practical skills on software quality engineering and improve our group work ethic.

## 1.2 Limitations

Known Limitations:

1. Limited Functionality: Initial release focuses on core features, omitting advanced functionalities.
2. User Interface: UI may lack polish due to time constraints.
3. Testing Coverage: While testing is implemented, coverage may not be exhaustive.

Planned Features:

1. Enhanced Reporting: Detailed reporting for restaurant owners.
2. Integration with External Systems: Integration with payment gateways and inventory management.
3. Mobile Optimization: Optimization for mobile devices for improved accessibility.
4. Refined User Experience: Enhancements based on user feedback for improved usability.

## 2 Product concept and requirements

### 2.1 Vision statement

Our goal is to revolutionize the restaurant industry by providing a tailored digital solution that enhances customer satisfaction, streamlining operations, and optimizing order management. Our digital platform, Pollos Hermanos, will feature intuitive components designed to meet the needs of both employees and customers in an effective and user-friendly manner.

We aim to differentiate our solution by offering innovative features that set us apart from existing products in the market (such as McDonald's Mobile App (\*1), Domino's Pizza Tracker (\*2), etc). Some of our strategies to distinguish our product from those apps are:

- Easy Tracking By Customers: After placing an order, the customer receives an "Order Number" and can consult its status in the "history" tab of our website or in the live screen of the corresponding restaurant.
- Easy Tracking By Employees: The employees can easily see which orders they need to prepare and which are ready to deliver.
- Reorder: A customer can easily reorder an order made previously by just checking his order's history.
- Embracing: Many restaurants and options in just one place!
- Security: Secure login and protection of the actions taken by the user.
- Enhanced User Experience: Focus on providing a seamless and intuitive user experience across all touchpoints of the digital platform. Invest in user interface design, navigation, and usability testing to ensure that the ordering process is efficient, enjoyable, and easy to use. We aim to incorporate features such as visual menus, one-click ordering, and saved order preferences to streamline the user journey.

Some key goals for the platform include reducing order processing time and increasing customer retention rates.

The platform will include the following key components:

- **Ordering System** ( $\Phi$ -Patient equivalent):
  - A customer-facing portal allowing diners to place orders, view order history (agenda), and select from various meals and restaurants.
  - Features include fast checkout, intuitive menu browsing, and various options.
  - Key Functionalities: search by cuisine type, preference for specific dishes, and consideration of lately orders.
- **Counter Service System** (our  $\Phi$ -Desk equivalent):
  - Central hub for employees to get and manage orders.
  - Streamlined system for efficient order processing, queue management, and coordination with the kitchen staff.
- **Digital Menu Boards System** (our  $\Phi$ -Boards equivalent):
  - Digital signage screens displaying order status for customers.
  - Effective management of order queues to enhance customer experience.

(\*1) McDonald's offers a mobile app that allows customers to place orders for pickup/delivery, customize their meals, and earn rewards through the loyalty program.

(\*2) Domino's Pizza Tracker is an online ordering system that allows customers to track the status of their pizza orders in real-time, from preparation to delivery.

## 2.2 Personas and scenarios

### Personas:

#### 1. **Fiona** (Family-focused customer)

Background: Fiona is a working mother who takes her family to various restaurants on a regular basis.

Needs:

- Family Meal Options (to satisfy the diverse tastes of her family, Fiona searches for combo meals or family meal deals)
- Efficiency (to reduce stress and guarantee a pleasant meal with her family, she looks for a simplified ordering procedure)
- Order History (to easily reorder her favorite dishes or keep track of previous purchases, Fiona appreciates having access to her order history)

#### 2. **Ethan** (Employee)

Background: Ethan is employed by a restaurant business as a cashier.

Needs:

- Order Management (to effectively process orders, handle payments, so Ethan needs an intuitive user interface)
- Queue Management (to efficiently manage order queues and assign orders based on urgency or specific demands, he requires the right tools)
- Communication

### Scenarios:

#### 1. **Going out with family** (1st Scenario for **Fiona**):

- Fiona decides to go with her family to dinner at her favorite restaurant after a busy day.
- She goes to our website and after checking our selection of restaurants, selects one, and selects various products (and the corresponding quantity) from that restaurant to her family.
- When they arrive, she approaches the counter service system to place her family's order.
- Fiona quickly selects all the products she wishes, adds them to the cart, and pays (and consequently gets an order number).
- With the order number, Fiona checks on the Live Call Screen her order status.
- When her order status changes to "Done", she gets the order.
- The efficient order processing and friendly service ensures a great dining experience for Fiona and her family.

#### 2. **Takeout Order** (2nd Scenario for **Fiona**):

- Fiona wants to order takeout from her favorite restaurant for dinner at home.
- She accesses her order history on the platform to quickly reorder their favorite dishes from previous visits.
- With a few clicks, Fiona places the order she wishes.
- After that, she pays and receives an order number.

- Then she goes to check on the Live Call Screen when the order is ready.
- Thanks to the order history access, Fiona can effectively organize a delicious evening for her family.

### 3. **Busy Lunchtime at the Counter** (1st Scenario for **Ethan**):

- During the busy hour, the restaurant experiences an increase in customer orderings.
- Ethan efficiently manages the counter service system, processing and delivering orders.
- Using the streamlined interface, Ethan quickly gets the orders that need to be done and the orders that are already done, ensuring timely preparation and delivery to customers.
- His effective order management skills help minimize wait times.

## **User Stories:**

### 1. **Creating an account and logging in:**

As Fiona, I want to create an account and log in, so I can place an order.

Acceptance criteria:

GIVEN I'm using the client site and I don't have an account.

WHEN I register using an email and a password and then I login with the same credentials.

THEN I can see that I am logged in and an option to log out.

### 2. **Choosing a restaurant:**

As Brian, I want to choose one of the available restaurants so that I can order from this restaurant.

Acceptance criteria:

GIVEN I'm on the Pollos Hermanos website restaurants tab.

WHEN I choose a restaurant.

THEN I can see my current selection and can still change it.

### 3. **Choosing items from the menu:**

As Brian, I want to search the menu for an item, so that I can order it.

Acceptance Criteria:

GIVEN I am on the 'menu' page

WHEN I enter the name or description of an item in the search bar

THEN the menu dynamically filters to display only the items matching the search query and displays their information

### 4. **Ordering:**

As Brian, I want to place an order so that I can use the platform's main use case.

Acceptance Criteria:

GIVEN I'm logged in on the Pollos Hermanos website menu tab and I have already chosen a restaurant.  
WHEN I choose an item.  
THEN I can see my chosen item in the basket.

**5. Changing order status as an employee:**

As Ethan, I want to change an active order status, so I can work on other orders.

Acceptance criteria:

GIVEN I'm using the employee site and I'm logged on an employee

WHEN I choose an order from the queue

THEN I should be able to change its status to 'done' ('serving') and 'delivered' ('collected')

**6. Payment:**

As a Brian, I want to pay for my order, so that I'm sure it will be completed.

Acceptance criteria:

GIVEN I just placed an order and I'm on the payment site.

WHEN I click on the pay button.

THEN I should see a confirmation screen and my order number

## **2.3 Project epics and priorities**

### **Expected Epics:**

**1. Ordering System Epic:**

Epic Description: Develop a customer-facing platform where diners can place orders, check their order history, and personalize their meals.

User Stories:

- As a customer, I want to easily navigate the restaurant/menu items and add them to my cart for quick ordering.
- As a customer, I want to access my order history to reorder my favorite dishes quickly.
- As a customer I want an easy checkout and payment process.
- As a customer, I want real-time order tracking of my orders and receive estimated pickup/delivery times for better planning.

Priority: High

**2. Counter Service System Epic:**

Epic Description: Develop a central hub for employees to manage orders, process payments, and assist customers.

User Stories:

- As an employee, I want an intuitive interface to process orders efficiently and accurately.
- As an employee, I want tools to easily manage the orders and get their status.

- As an employee, I want to communicate seamlessly with the kitchen staff to ensure timely order preparation and delivery.
- As an employee, I want to easily change status from “Preparing” to “Done” and from “Done” to “Delivered”.

Priority: High

### **3. Integration and Scalability Epic:**

Epic Description: Ensure seamless integration of all components and scalability to accommodate future growth and technological advancements.

User Stories:

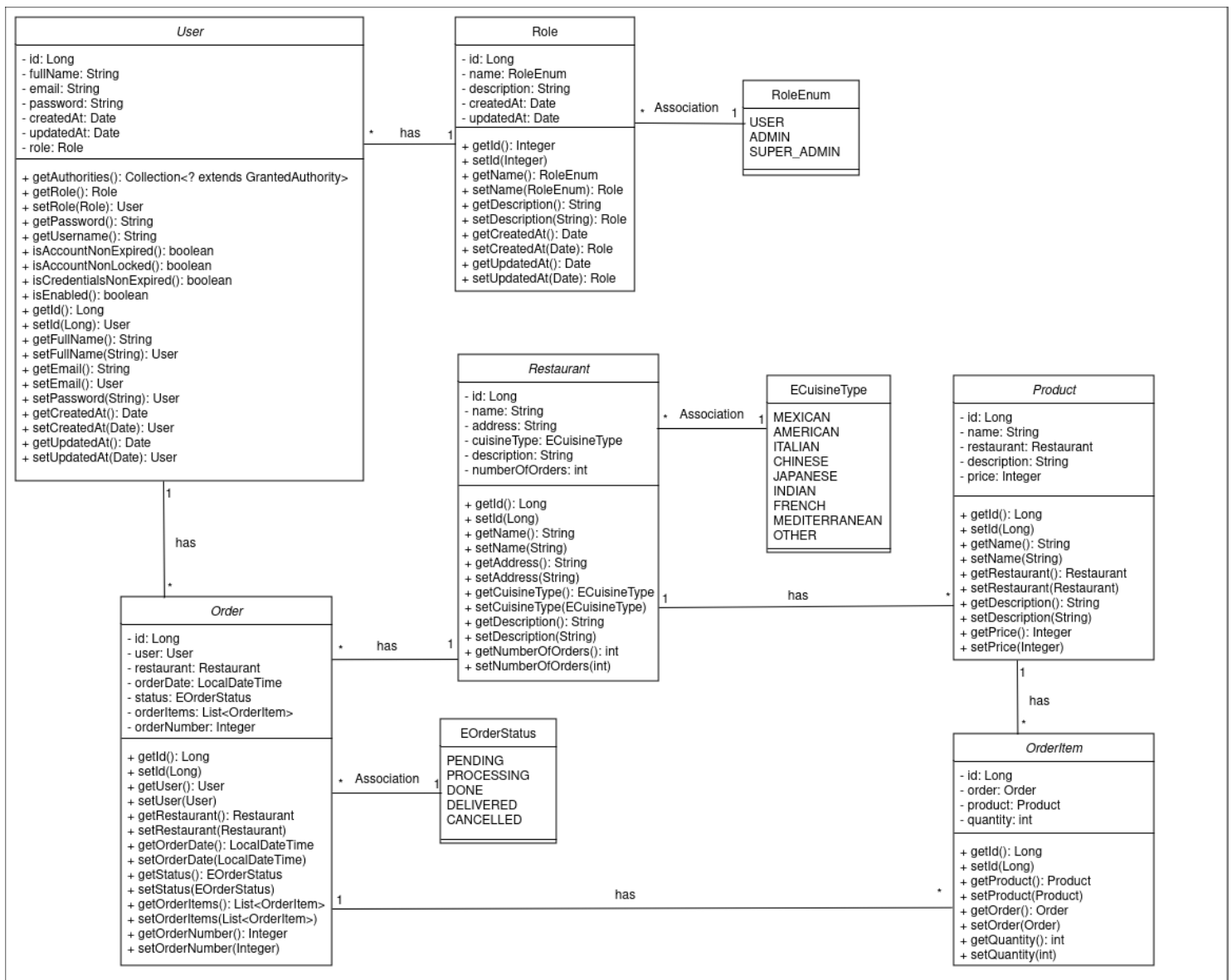
- As a system administrator, I want to ensure that all components of the digital platform integrate seamlessly to provide a unified experience for customers and employees.
- As a system administrator, I want to design the platform with scalability in mind to accommodate increasing demand and future technological advancements.
- As a system administrator, I want to implement robust security measures to protect customer data and ensure compliance with privacy regulations.

Priority: Medium

### 3 Domain model

- Each User has a specific Role, and has multiple Orders.
- Each Role can have multiple Users, and is associated with a specific role type (RoleEnum)
- Each Restaurant has a specific cuisine type (ECuisineType), has multiple Products, and has multiple Orders.
- Each Product is related to only 1 Restaurant, and can be in multiple Orders.
- Each OrderItem has only 1 Product, and has only 1 Order associated with.
- Each Order is associated with an status (EOrderStatus), has only 1 User associated with, has only 1 Restaurant associated with, and can have multiple OrderItems associated with.

Here is a diagram to a better understand of the domain model:





## 4 Architecture notebook

### 4.1 Key requirements and constraints

#### 1. Functional Requirements

1. **User Registration and Login:** Users must be able to create accounts and log in using an email and password.
2. **Restaurant Selection:** Users must be able to choose from a list of available restaurants.
3. **Menu Browsing and Item Selection:** Users must be able to search the menu and select items to add to their cart.
4. **Order Placement:** Users must be able to place orders and view their order history.
5. **Order Status Tracking:** Users must be able to track the status of their orders in real-time.
6. **Reordering:** Users must be able to reorder from their order history.
7. **Employee Order Management:** Employees must be able to manage and update the status of orders.
8. **Payment Processing:** Users must be able to securely pay for their orders.

#### 2. Non-Functional Requirements

1. **Usability:** The platform must provide an intuitive and user-friendly interface for both customers and employees.
2. **Performance:** The platform must be responsive and handle multiple concurrent users efficiently.
3. **Scalability:** The platform must be designed to scale seamlessly to accommodate future growth and increased demand.
4. **Security:** The platform must ensure secure login, data protection, and secure payment processing.
5. **Reliability:** The platform must be reliable with minimal downtime and high availability.
6. **Maintainability:** The platform must be easy to maintain and update.

### 3. System Constraints

#### 1. Hardware Constraints

- a. **Description:** The system must run on the existing hardware infrastructure available in the restaurants.
- b. **Constraint:** Limited computing power and storage on existing POS systems and display screens.

#### 2. Software Constraints

- a. **Description:** The system must integrate with existing software used by the restaurant, such as POS systems and inventory management.
- b. **Constraint:** Compatibility with legacy systems and software versions.

#### 3. Network Constraints

- a. Description: The system must operate efficiently within the restaurant's current network capabilities.
- b. Constraint: Limited bandwidth and potential network latency, especially during peak hours.

#### **4. Regulatory Constraints**

- a. Description: The system must comply with local and international regulations regarding data protection and payment processing.
- b. Constraint: Adherence to GDPR, and other relevant regulations.

#### **5. Security Constraints**

- a. Description: The system must implement robust security measures to protect sensitive customer and business data.
- b. Constraint: Strong encryption for data in transit and at rest, and regular security audits.

#### **6. User Interface Constraints**

- a. Description: The system's user interface must be designed to be easily usable by both customers and employees with varying levels of technical expertise.
- b. Constraint: Limited screen size for certain devices (e.g., POS systems) and need for high accessibility standards.

#### **7. Development Constraints**

- a. Description: The system must be developed within a set timeframe and budget.
- b. Constraint: Strict deadlines and budget limits, requiring efficient use of resources and prioritization of features.

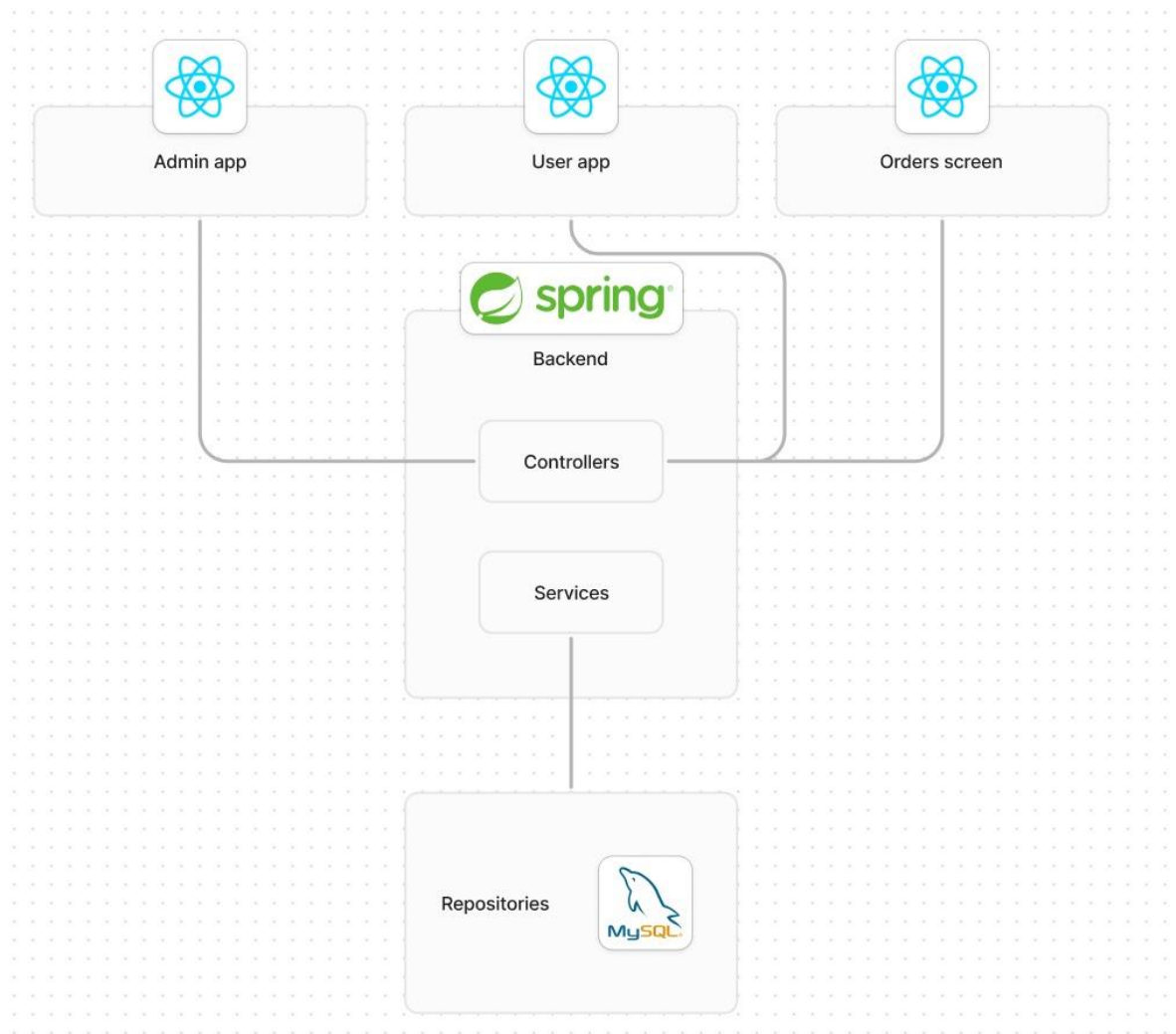
#### **8. Scalability Constraints**

- a. Description: The system must be designed to scale up to accommodate future growth.
- b. Constraint: Potential need for re-architecting components to handle increased load and additional features.

#### **9. Performance Constraints**

- a. Description: The system must perform well under high user load and during peak business hours.
- b. Constraint: Ensuring low latency and high responsiveness even during peak usage times.

## 4.2 Architecture view



## 5 API for developers

Our API is organized around RESTful principles, providing easy-to-use endpoints to interact with our system. Below is an overview of the resources and services available through our API (we used Swagger for documentation - “References and Resources” section):

Admin Operations related to the administrator of the system		^
POST	/admins Create a new administrator of the system. Only accessible to Super Administrators (users with the SUPER_ADMIN role)	📄 🔒 🔗 @ ▼
Authentication Operations related to user authentication		^
POST	/auth/signup Register a new user	📄 🔒 🔗 @ ▼
POST	/auth/login Authenticate and receive JWT token	📄 🔒 🔗 @ ▼
Orders Operations related to orders management		^
GET	/orders Get user's order history	📄 🔒 🔗 @ ▼
POST	/orders Place a new order	📄 🔒 🔗 @ ▼
GET	/orders/{orderId} Get order by ID	📄 🔒 🔗 @ ▼
PATCH	/orders/{orderId} Update order status	📄 🔒 🔗 @ ▼
POST	/orders/payment/{orderId} Pay for an order	📄 🔒 🔗 @ ▼
GET	/orders/in-progress/{restaurantId} Get in-progress orders for a restaurant	📄 🔒 🔗 @ ▼
GET	/orders/done/{restaurantId} Get done orders for a restaurant	📄 🔒 🔗 @ ▼
Products Operations related to the restaurant's products (meals)		^
GET	/products Get all products	📄 🔒 🔗 @ ▼
GET	/products/{id} Get product by ID	📄 🔒 🔗 @ ▼
GET	/products/restaurant/{restaurantId} Get products by restaurant ID	📄 🔒 🔗 @ ▼
Restaurants Operations related to restaurants		^
GET	/restaurants Get all restaurants	📄 🔒 🔗 @ ▼
GET	/restaurants/{id} Get restaurant by ID	📄 🔒 🔗 @ ▼
Users Operations related to users		^
GET	/users/me Retrieve the authenticated user	📄 🔒 🔗 @ ▼
GET	/users Retrieve all users (ADMIN or SUPER_ADMIN only)	📄 🔒 🔗 @ ▼

## 6 References and resources

Jira (Project's Management): <https://mateusz-kubiak.atlassian.net/jira/software/projects/NA/boards/2?atlOrigin=eyJpIjoiZTIzOWE5MTZjMjQ3NDg3M2E2NjJmODM5MTU3MDdmYzliLCJwIjoiajJ9>

Figma (Mockup):  
<https://www.figma.com/file/f0iRnEPgXhjRjCzCGBp8BM/Untitled?type=design&node-id=0%3A1&mode=design&t=gR20IXJPBC2gRh1e-1>

Swagger (API Documentation):  
<https://app.swaggerhub.com/apis/MARIAJOAOMS03/polloshermanos/1>

Libraries:

- React
- SpringBoot
- MySQL