Technical Report - **Product specification**

FoodFlow

Course:	IES - Introdução à Engenharia de Software
Date:	Aveiro, 7th October 2024
Students:	113144: João Pedro Moreira Viegas 113278: Jorge Guilherme Conceição Domingues 113893: Guilherme Ferreira Santos 114547: João Pedro Ferreira Monteiro
Project abstract:	This project involves the development of a real-time statistics display application for fast food chains, allowing users to monitor popular menu items and track the status of orders within specific restaurants.

Table of contents:

1 Introduction

The pace of modern life has driven an enormous rise in the popularity of quick-service restaurants such as McDonald's and other fast-food chains. Such restaurants live and die by efficiency. That said, as the customer preferences change more frequently today than ever before, keeping track on our favorite items and, at the same time, being capable of seeing real-time status of the order congestion of a restaurant can be very helpful to save time to the customers.

This project aims to develop an application where the user can monitor and see statistics in real-time from fast-food franchises that are already installed.

Users of the app will be able to see live data on what is sold where and in which specific restaurants. It will show users a list of orders that have been placed, are in progress and also those completed. This will not only improve the customer experience by offering a real-time view but also help users and restaurant managers to take informed decisions based on usage data. By making use of the real-time display and tracking ability, this project aims at ensuring that fast-food restaurants are more open about their performance by letting users know what is hot on menus as well as how far gone an order has reached whenever they require to know.

2 Product concept

Vision statement

Project Vision:

Our application is designed to offer real-time visibility into order statuses and popular menu trends within fast food restaurants, providing both customers and restaurant managers with essential data for improving efficiency and decision-making.

Functional (Black-Box) Description of the Application

The application will serve two main purposes:

1. **Customer-Facing Display**:

It will allow customers to view real-time statistics on popular menu items and track the

progress of orders in fast food restaurants like McDonald's. Customers can check the number of orders currently in progress and how close their own order is to being completed (based on the order number they received at the restaurant). However, the app will not include any functionality for placing or managing orders; it is purely informational.

2. **Managerial and Administrative Tool**:

Restaurant managers will use the app to monitor the restaurant's performance in real time. They will be able to see how many orders are to be done, in progress, or completed, helping them allocate resources effectively. Additionally, administrators (like Ana Martins in the personas) will manage access for restaurant managers by creating and assigning credentials.

High-Level Business Problem Being Solved

Fast food restaurants face challenges in managing order efficiently, especially during peak hours. Customers are often left in the dark about wait times and the status of their orders, leading to dissatisfaction. Similarly, restaurant managers struggle to prioritize tasks without access to real-time data, potentially affecting service quality. The business problem being addressed by this system is the **lack of transparency and real-time operational insights**, which affects both customer experience and restaurant efficiency.

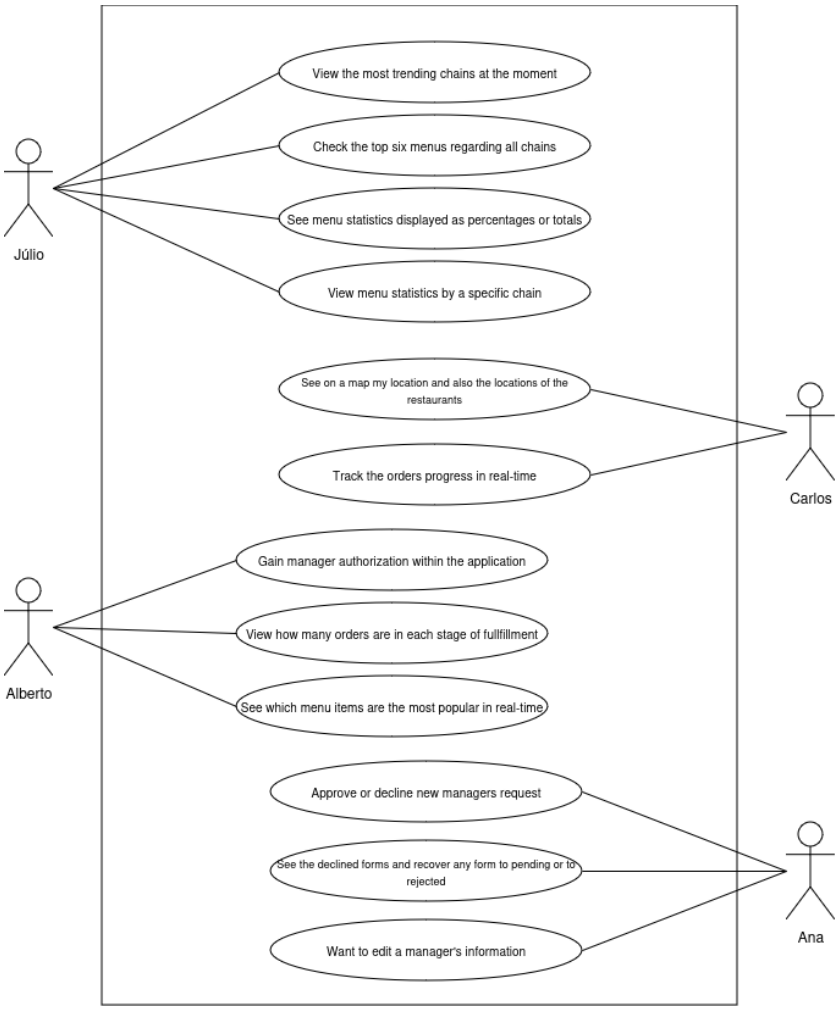
Planned vs. Actual Features

Originally, the system was envisioned to include functionalities for customers to place orders directly through the app. However, we shifted away from this concept and chose a **non-interactive, display-only model**. Now, the app serves as an informational tool, offering real-time order statuses and statistics without the capability to process orders.

Comparison to Other Products

While there are existing apps that allow customers to place orders and track them, such as the McDonald's app or UberEats, our system is **unique** in that it focuses solely on providing **real-time insights into restaurant operations** and does not involve order placement. Apps like these tend to prioritize customer interaction through orders, while our solution focuses purely on giving users (both customers and managers) information about **order status and menu popularity** without the need for an account or order linkage.

UML Use Case Diagram



Personas and Scenarios

Persona – Carlos Fernandes

Carlos is a 19-year-old Economy student studying in Aveiro's University. With the constant growth of the university and with the existence of only two canteens on the campus, eating in the university can be a time-consuming activity. As a student, Carlos must be time-efficient because he does not have much time in between classes or because he has schoolwork to do. Because of this, he occasionally resorts to fast food chains to hurry himself. Despite being a quick solution, sometimes it is not the most effective, since the time he takes to arrive at the restaurant, order, eat and return to the university are not enough.

Motivation: Carlos would like an easy solution to help him understand if he has enough

time to go to a certain restaurant.

Persona – Alberto Joaquim

Alberto is a 40-year-old lawyer and owner of a fast-food restaurant. Despite his time-consuming and successful path in advocacy, he has always been keen on having a business in the restaurant industry. He wants to keep track of his restaurant performance efficiently, comparing it with other restaurants from his or other chains. He needs a place where he can visualize the most sold menus, the clients preferred items and the waiting time of given orders. He wants to maximize his restaurant efficiency and take informed decisions to be able to provide the best experience possible.

Motivation: Alberto wants to understand the habits of the public, optimize the response time and improve the financial results, all while he keeps his work as a lawyer. He is looking for an easy-to-use system that provides useful and easy-to-see data, so he can improve his decisions.

Persona – Júlio Rodrigues

Júlio is a 27-year-old Civil Engineer based in Porto. He has taken a small vacation in Aveiro, and he has the will to go and try a new fast-food chain that previously did not operate in Portugal, and that just opened a restaurant in Aveiro. Since it is a new and unknown restaurant he wants to try the most famous order. He would like to have an application to choose the most desired option from the public, so he can see a real-time graph of the orders that are being requested, which helps him identify the order he wants to place.

Motivation: He wants to be sure that he tries the most popular menu, fulfilling his experience in a new restaurant.

Persona – Ana Martins

Ana Martins is a 45-year-old system administrator responsible for managing access to the restaurant management system across multiple fast-food chains. Ana's primary responsibility is to ensure that restaurant managers have secure and appropriate access to the system so they can view the real-time statistics for their respective locations. Daily, Ana creates new accounts for managers, assigns login credentials, and manages any security issues such as resetting or revoking access when needed.

Motivation for Ana: Ana is motivated by the need to maintain security and organization within the system. She wants to ensure that only qualified managers have access to the

app. Ana's focus is on safeguarding restaurant data while making the system easy to use for managers.

Scenario – Carlos Fernandes

Carlos checks if it is profitable to go to a McDonald's - Carlos opens the app, where he can observe a list of fast-food chains. From this list, he chooses the one he desires (McDonald's). After he does this, he can observe a map centered on his location and where he can see nearby restaurants from that chain. He clicks on the closest one and he will be able to see what the current order flow is, i.e., if that certain restaurant has a lot of orders at that moment or not. This allows him to decide whether it is profitable or not to go to that restaurant, timewise.

Scenario – Alberto Joaquim

Alberto checks on his app while he is at work – Since Alberto is a lawyer and has responsibilities as a lawyer, he cannot always be present at his restaurant. Despite this, he can use the app to check how the restaurant is doing, if everything is working as intended. By seeing the graphs disponible, he can observe that there are orders being taken and served to the customers, which allows him to be more relaxed and focused on his duties as a lawyer.

Scenario – Júlio Rodrigues

Júlio checks the most famous menu in a new chain - Júlio opens the app, where he can observe a list of fast-food chains. From this list, he chooses the one he desires (a brand-new chain that just opened in Aveiro). After he does this, he can observe a map centered on his location and from that map choose the restaurant from that chain. After this, he is presented with a live-time graph that shows what the trending menus have been for the past days. With this graph, he can clearly see what the most famous menu is.

Scenario – Ana Martins

Ana starts her workday by checking that a new manager is set to start at one of the McDonald's branches today, so Ana opens the app's administrative interface to create a new account. She generates login credentials for the manager, assigns the correct

permissions for accessing the restaurant's statistics, and sends the credentials to the manager.

Product requirements (User stories)

Epics

Epic 1: Real-Time Menu Statistics

- **Description:** This epic focuses on displaying real-time statistics of the most consumed menu items across various fast-food chains. The goal is to give users access to live data on menu trends and popularity to inform their dining decisions.

Epic 2: Restaurant-Specific Insights

- **Description:** This epic involves providing users with the ability to view statistics for individual fast-food restaurants, such as the number of orders in the queue, orders in progress, and completed orders. This helps users choose locations based on service efficiency.

Epic 3: Restaurant Management Dashboard

- **Description:** This epic addresses the needs of restaurant managers, offering a management dashboard that displays restaurant-specific operational data. Managers can track order fulfillment status, view which items are popular, and adjust resources accordingly.

Epic 4: Administration Management

- **Description:** This epic addresses the needs of restaurant managers, offering a management dashboard that displays restaurant-specific operational data. Managers can track order fulfillment status, view which items are popular, and adjust resources accordingly.

User Stories for Each Epic

Epic 1: Real-Time Menu Statistics

- **User Story 1.1:** As Júlio, I want to view the most trending chains at the moment.
- **User Story 1.2:** As Júlio, I want to be able to check the top six menus regarding all

chains.

- **User Story 1.3:** As Júlio, I want to see menu statistics displayed as percentages or totals, so I can easily compare the popularity of different items.
- **User Story 1.4:** As Júlio, I want to view menu statistics by a specific chain (e.g., McDonald's, Burger King) or even by a specific restaurant, so I can focus on the popularity of items from my favorite restaurants.

Epic 2: Restaurant-Specific Insights

- **User Story 2.1:** As Carlos, I want to be able to see on a map my location and also the locations of the restaurants of a food chain, so I can choose the closest one.
- **User Story 2.1:** As Carlos, I want to track the orders progress in real-time, so I know whether it's in the "to-do," "preparing," or "done" stage.

Epic 3: Restaurant Management Dashboard

- **User Story 3.1:** As Alberto, I want to gain manager authorization within the application, through a form where I also can register my restaurant.
- **User Story 3.2:** As Alberto, I want to view how many orders are in each stage of fulfillment (to-do, in progress, done), so I can manage staff and resources more effectively during peak hours.
- **User Story 3.3:** As Alberto, I want to see which menu items are the most popular in real-time, so I can adjust kitchen priorities and ingredient stock levels.

Epic 4: Administration Management

- **User Story 4.1:** As Ana, I want to be able to approve or decline new managers' requests, so I can ensure that each manager has individual access to the system.
- **User Story 4.2:** As Ana, I want to be able to see the declined forms, so I can possibly recover any form to pending or to deleted.
- **User Story 4.3:** As Ana, I want to edit a manager's information.

3 Architecture notebook

Key requirements and constrains

1. **Order Generation and Processing**:
    a. Order generation and high-throughput order processing must be handled in real-time by the system.
2. **Scalability and Performance**:

    a. With the large number of orders being handled in real time, the system needs to scale effectively.
    b. The back end needs to minimize direct access to the database
3. **Data Security and User Roles**:
    a. Security measures are required because the system will handle different types of user roles. The system needs to guarantee that users only see and interact with data that they are allowed to interact.
4. **Multiple Services with Defined Responsibilities**:
    a. Architecture needs to have multiple services to distribute responsibilities.
5. **External Services and Interaction**:
    a. There is integration with an external **front-end** developed.
    b. Communication between the front-end and the back-end is done by **REST API** endpoints.

Architetural view

This section explains each architectural component's function and goal. These elements stand for the main structural parts that work together to create a system that is reliable, scalable, and effective.

1. Data Source (Order Generator)

- The **Order Generator** is going to simulate/generate new orders being done in the restaurants.
- It will be implemented in Python.
- Once an order is simulated, it is sent as a message to the **Message Broker (Kafka)** to the restaurant's topic

2. Message Broker (Kafka)

- Between the back-end services and the data source (order generator), the Message Broker acts as a mediator. It guarantees the consistent delivery of orders to the processing system.
- Real-time message queues, which offer fault tolerance, scalability, and durability, are handled by Apache Kafka.
- The produced order messages are queued by Kafka. Until the Order Processing Service consumes them, these messages stay in the queue.

3. Back-End

- The back-end is divided into several services, each with its functionality. These

services include:

Order Processing Service

- It oversees taking in Kafka messages, processing orders (validating, assigning status, etc.), and storing them in the database.
- It is also responsible to add dynamic listeners to each pre-existing restaurant's topic as well each dynamic added restaurant

User Service

- Manages access control, authorization, and user authentication.
- Also used to create, edit and delete credentials to the managers for the administrators to use

Order Service

- In charge of getting all the information relative to the orders since the orders of an entire food chain or orders of a specific restaurant
- It also provides for endpoints of the statistics using the Statistics module

Statistics

- Module that is responsible for formatting raw data coming from the statistics queries. With this, the Order Processing Service is abstracted of the logic of how the data is organized to show in form of statistics

Restaurants Service

- It has the responsibility to retrieve data about the restaurants, food chains and the menus

Data Access Layer

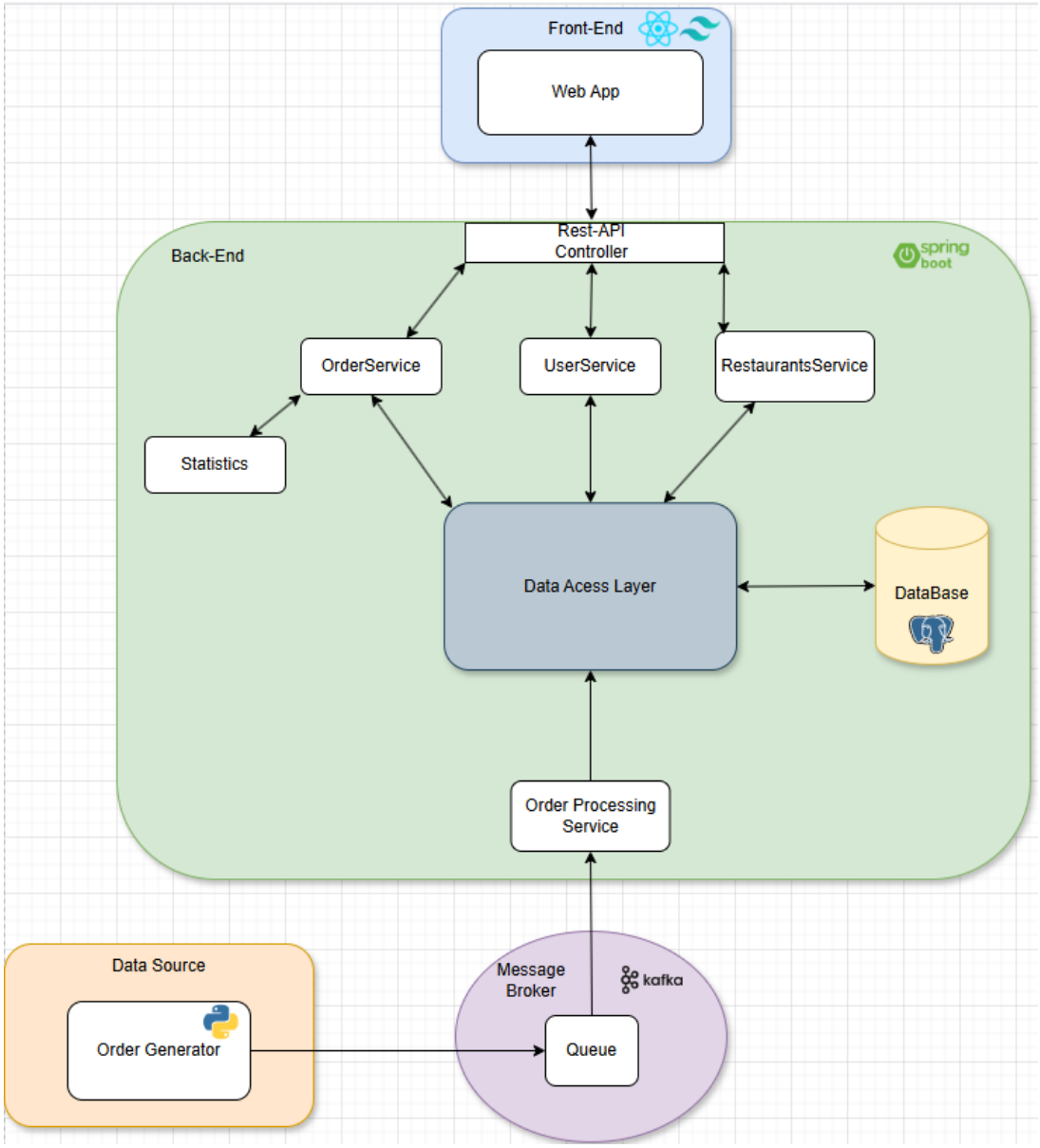
- The service facilitates direct communication between the database and other intermediate services.
- By abstracting direct database contacts, this service enhances security and facilitates database transaction management.
- It is in this layer that will be inserted the Repositories of the entities that will be used by the Services

4. Database

- keeps track of all persistent information about users, orders, and system transactions.
- There are several tables in it, such as ones for users, orders, menus and more.
- The only part that communicates directly with the database is the Data Access Layer, which increases security and reduces the possibility of unauthorized data changes.

5. Front-End (React Application)

- Customers, managers, and administrators interact with the system through the front-end, which is a web application. To retrieve or change data, it uses the REST API to send HTTP queries.
- Interactions with Users:
    - Clients: View restricted statistics and order status.
    - Managers: Examine performance information and full-order trends and statistics.
    - Administrators: Control roles and users.
- REST API Communication: Requests are routed to the relevant microservices by the back-end's REST API, which communicates with the front-end.

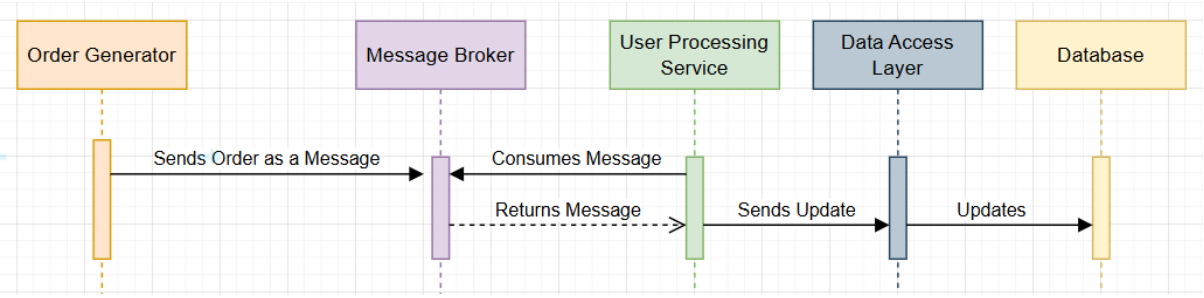


Module interactions

- **Order Generation Reaction**

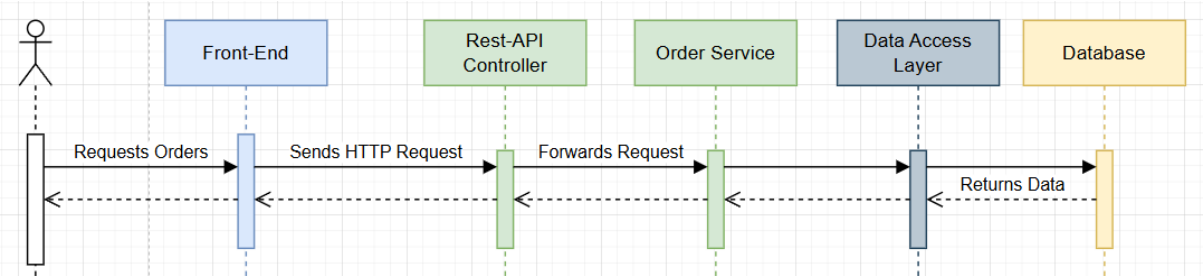
The Order Generator is simulating orders being created in the fast-food chains and, every time an order is created (when order it can be either a new order or an order update), it sends it as a message to the Message Broker that should put this message in the Message Queue in the order's restaurant's topic

From the Back-End, the Order Processing Service is always listening to the queue in each known restaurant's topic ready to consume the messages. In each message, it should process it and see what action should do in the database. When this action is decided, send it to the database through the Data Access Layer and send it also to the WebSocket.



- **User – Orders Status**

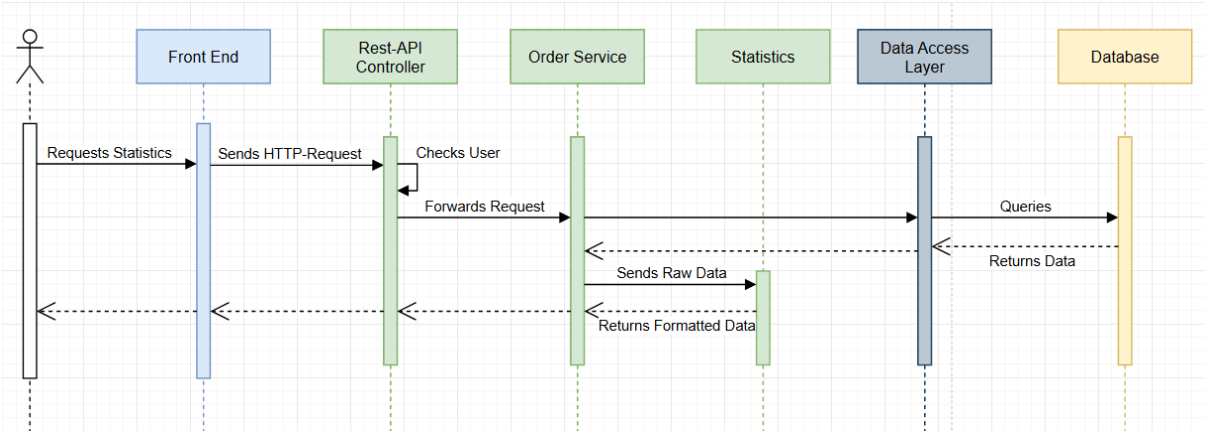
The user chooses a restaurant to see its orders status/congestion. There, the Front-End should make an HTTP Request to the Rest-API where it forwards the request to the Order Service, because the order status is equal to every user, this service just requests the desired orders from the specific restaurant to the Data Access Layer that queries the Database for data. The Database returns the expected data, and this data makes its way to the user to see.



- **User – Statistics Interaction**

The User goes to the page where it shows the statistics, from there, the Front-End should make an HTTP Request to the Rest-API where checks the user role via Spring-Security with his session jwt token.

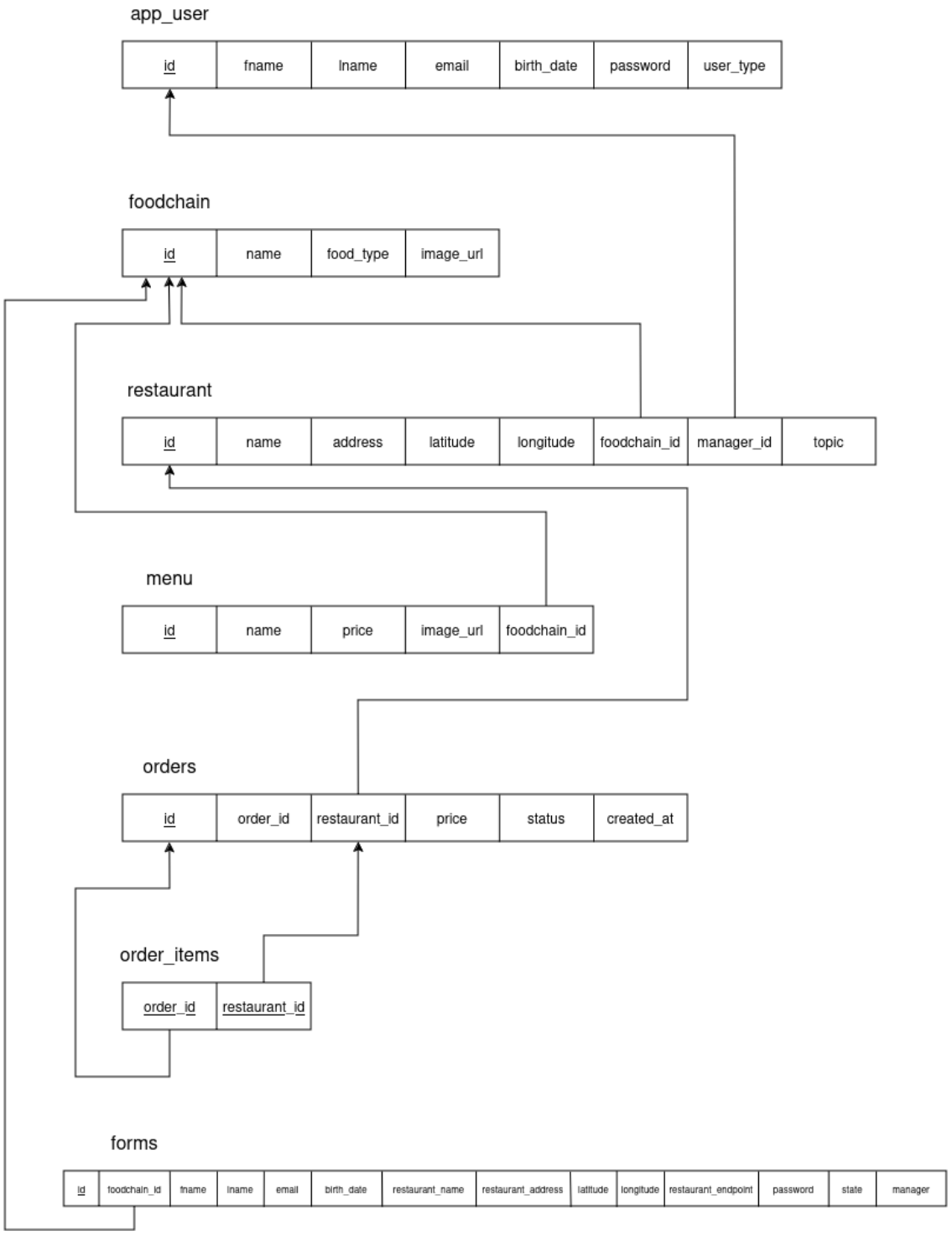
This service should validate the user because it will see different statistics depending on his role. Then the request is made to the Order Service that will, through the Data Access Layer, ask for the raw data of the statistics. After the Database answers the request through the Data Access Layer, the Order Service will use the Statistics module to format the data in a way that the Front-End can easily process and show to the User.



Design Issues:

1. **Data Synchronization**:
    a. Data synchronization between the Order Generator and the Order Processing Service is dependable and scalable thanks to Kafka.
2. **Distributed Workflows**:
    a. Because they are stateless and horizontally scalable, services like Order Service, RestaurantsService and UserService can be replicated and scaled in response to load.
3. **Push Notifications**:
    a. If necessary, web sockets in the REST API can be used to incorporate push notifications, which would enable the front-end to receive real-time order status updates whenever an order goes from "to-do" to "in-progress" or "done."
4. **Updates Distribution**:
    a. Because of the system's microservices architecture, upgrades may be implemented separately for each service (such as Order Processing or User Service) without causing the system as a whole to go down.

4 Information perspetive


app_user
id
fname
lname
email
birth_date
password
user_type
foodchain
id
name
food_type
image_url
restaurant
id
name
address
latitude
longitude
foodchain_id
manager_id
topic
menu
id
name
price
image_url
foodchain_id
orders
id
order_id
restaurant_id
price
status
created_at
order_items
order_id
restaurant_id
forms
id
foodchain_id
fname
lname
email
birth_date
restaurant_name
restaurant_address
latitude
longitude
restaurant_endpoint
password
state
manager

5 References and resources

https://spring.io/projects/spring-boot

https://spring.io/guides/tutorials/rest

https://spring.io/guides/gs/messaging-stomp-websocket

https://docs.spring.io/spring-kafka/reference/quick-tour.html

https://vite.dev/guide/

https://flowbite.com/

https://daisyui.com/

https://tailwindcss.com/

https://docs.docker.com/compose/