

TQS: Product specification report

André Miguel Antunes Cardoso [108269], Tiago Alexandre Abreu Figueiredo [107263], Alexandre Pedro Ribeiro [108122]
V2023-04-24

1.1	Overview of the project.....	1
1.2	Limitations.....	1
2.1	Vision statement.....	1
2.2	Personas and scenarios.....	2
2.3	Project epics and priorities.....	3
4.1	Key requirements and constrains.....	4
4.2	Architecture view	5
4.3	Deployment architecture	6

1 Introduction

1.1 Overview of the project

QuickServe is an innovative management platform designed to revolutionize the fast-food industry by providing a comprehensive solution for streamlining operations, enhancing customer experience, and empowering kitchen staff with real-time insights. The platform is tailored to meet the unique needs of system managers, customers, and kitchen employees, ensuring a seamless and efficient dining experience.

1.2 Limitations

Ideally the system would provide both the visualization and the algorithms necessary to optimize the workflows of the employees. While in a first instance this won't be available, we plan to encompass this area in a following iteration

2 Product concept and requirements

2.1 Vision statement

QuickServe envisions a future where the fast-food industry is transformed by a management platform that not only streamlines operations but also elevates the customer experience and empowers kitchen staff with real-time insights. Our vision is to create a seamless and efficient dining experience that is tailored to the unique needs of system managers, customers, and kitchen employees. By providing a

comprehensive solution that meets these needs, QuickServe aims to revolutionize the way fast-food chains operate, ensuring that every aspect of the dining experience is optimized for success.

2.2 Personas and scenarios

Alex Smith, the System Manager, brings a wealth of experience in IT and operations management to the role. His background equips him with the skills to tackle the challenges of inefficient order management, difficulty in tracking Key Performance Indicators (KPIs), and the complexities of optimizing work chains. Alex's goals are centered around streamlining operations, reducing costs, and enhancing customer satisfaction through the implementation of efficient management tools. His objectives are to leverage advanced management systems to improve workflows and ensure the smooth running of the fast-food chain.



Alex Smith, the System Manager, is tasked with optimizing the daily operations of the fast-food chain. On a typical day, Alex starts by reviewing the previous day's performance metrics to identify any bottlenecks or inefficiencies. He notices that the order management system is struggling to handle the volume of orders, leading to delays and customer dissatisfaction. Alex decides to implement a new order management tool that promises to streamline the process. He spends the morning setting up the new system, configuring it to handle the chain's specific needs. By the end of the day, the new system is fully operational, and Alex observes a significant reduction in order processing times. This improvement not only enhances customer satisfaction but also allows for better tracking of KPIs, aligning with his goal of streamlining operations and reducing costs.



Emily Doe, the Customer, represents the busy professional seeking a quick and convenient dining experience. Emily's needs are driven by her desire for quick service, variety in menu options, and a seamless ordering process. Her goals and objectives are focused on enjoying a fast, efficient, and enjoyable dining experience, with an emphasis on avoiding waiting in lines.

Emily Doe, a busy professional, is looking for a quick and convenient dining experience. She visits the fast-food chain's website to place an order for her favorite meal. The website's user interface is intuitive, allowing her to easily select her meal and add it to her cart. Emily chooses a delivery option to save time and receives a confirmation email with her order details and an estimated delivery time. As she waits for her order, she appreciates the variety of menu options available, ensuring she can enjoy different meals on different days. Upon receiving her order, Emily finds it to be exactly as she ordered, with the meal arriving on time. This seamless experience meets her needs for quick service and variety, reinforcing her positive perception of the fast-food chain.

John Baker, the Kitchen Employee, is a dedicated worker with a proven track record in fast-food kitchens. John's role involves preparing food orders efficiently and maintaining cleanliness. His

challenges include managing multiple orders simultaneously and keeping the workspace organized amidst the fast pace. John's goals and objectives are aimed at improving work efficiency, reducing errors, and enhancing the overall dining experience by ensuring cleanliness and organization in the kitchen.

John Baker, a kitchen employee, is preparing for a busy lunch rush. He uses the new order management system to efficiently manage multiple orders simultaneously. The system's clear display of orders and their status helps John keep track of what needs to be prepared and when. Despite the fast pace, John ensures that each order is prepared accurately and on time. He also takes a moment to organize the workspace, making it easier for him and his colleagues to work efficiently. By the end of the rush, John feels satisfied with his performance, having contributed to a smooth dining experience for the customers. This scenario aligns with his goals of improving work efficiency and reducing errors, enhancing the overall dining experience.



2.3 Project epics and priorities

For our implementation plan, we present the following epics and timelines:

Order Management System:

- **Objective:** Enhance the system for seamless order placement and real-time tracking.
- **Iteration 3:** Develop a basic order management system that allows customers to place orders and receive basic status updates.
- **Iteration 4:** Implement real-time tracking for orders, enabling customers to know when their food is ready.
- **Iteration 5:** Add a dashboard for system managers to view all orders and manage staff more effectively.

Work Chain Optimization:

- **Objective:** Enable dynamic adjustment of work chains based on real-time data.
- **Iteration 3:** Introduce basic work chain management features, allowing system managers to manually adjust work chains.
- **Iteration 4:** Implement a system that provides real-time data on customer demand and operational efficiency.

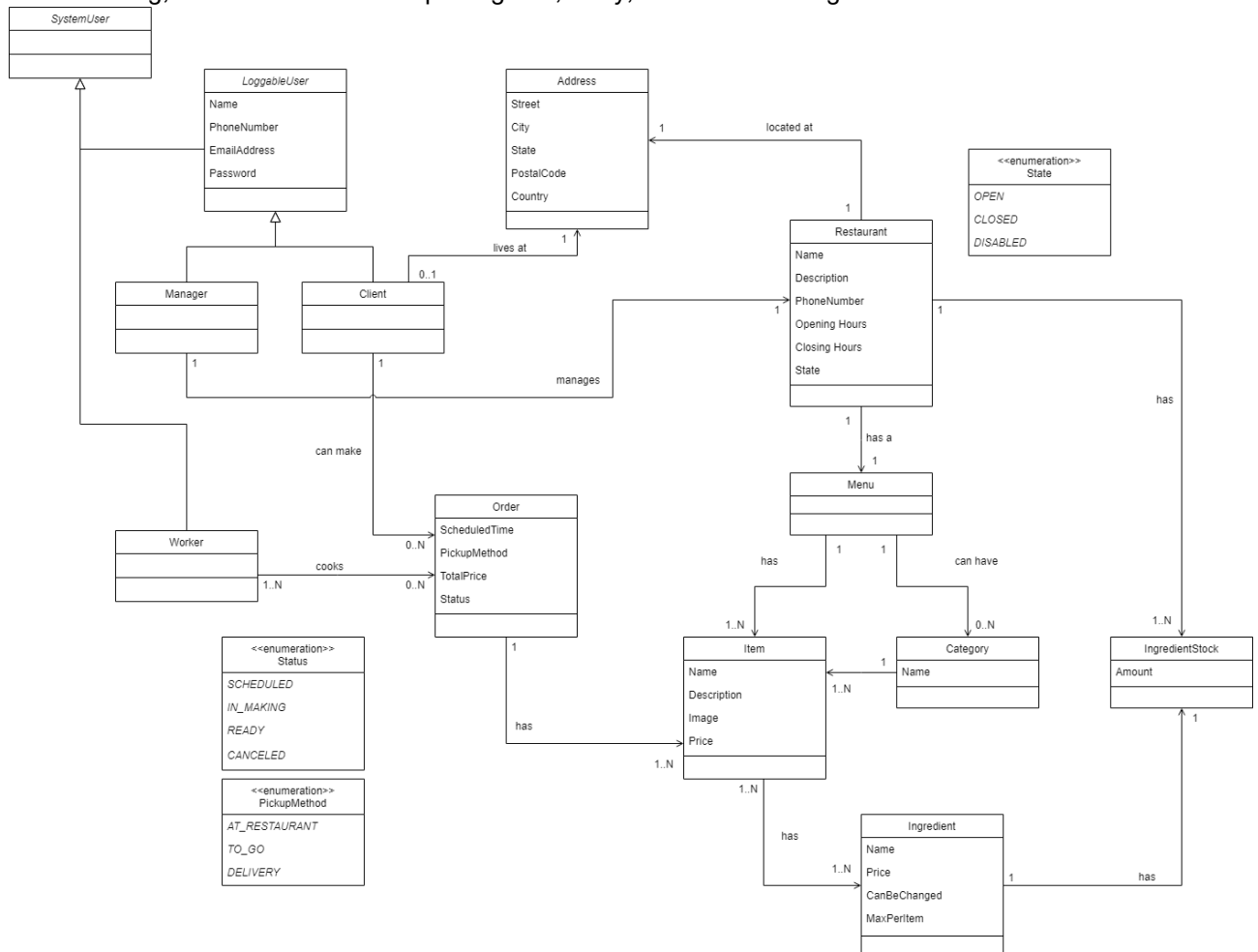
KPI Tracking and Reporting:

- **Objective:** Provide detailed analytics and reporting tools for system managers.
- **Iteration 3:** Set up basic KPI tracking for sales and customer satisfaction.
- **Iteration 4:** Develop detailed reporting tools that allow system managers to analyze operational efficiency.
- **Iteration 5:** Integrate the reporting tools with the order management system to provide a comprehensive view of the business's performance.

3 Domain model

Our system requires an intricate web of entities all working together to make sure all information is correctly stored in the right place, and the access between entities is intuitive.

There are many concepts we want to implement, and so the network of entities must reflect them in a way that inspires order. These concepts include user registration and login, order making and confectioning, menu creation and importing and, lastly, restaurant management.



4 Architecture notebook

4.1 Key requirements and constraints

There are a few key requirements that have a significant bearing on the architecture. They are:

- Efficient handling of usage spikes throughout the day, during lunch and supper.
- The service will be available on various platforms, primarily web, mobile and on big screens.
- The service should handle requests made from kiosks and registers.
- The process of cancelling orders from clients should quickly appear on the chef's screen.
- Menu items might become temporarily unavailable, due to missing ingredients.
- The System Manager must be able to import menus.

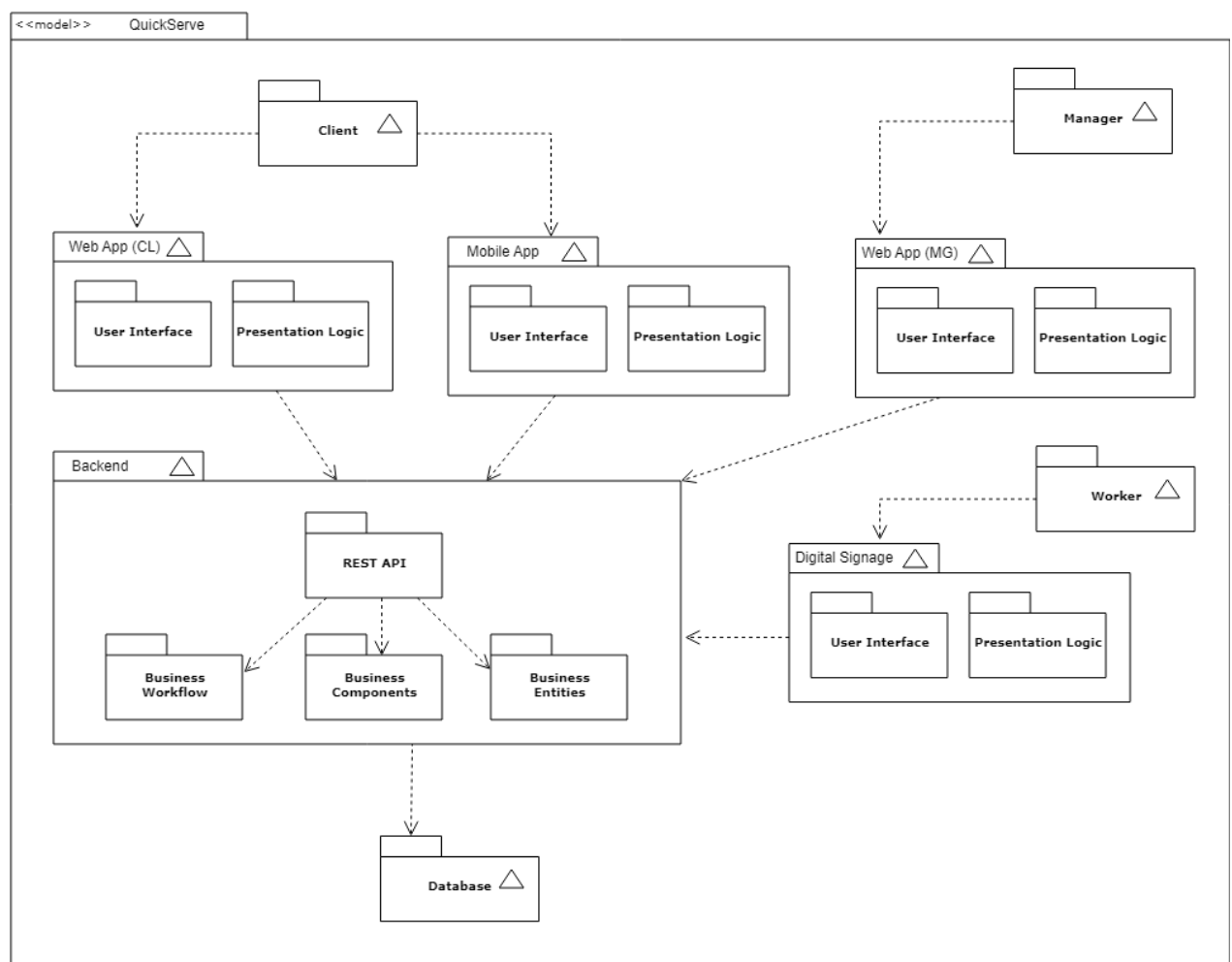
4.2 Architecture view

For the architecture, we need a REST API capable of handling large amounts of requests, coming from both our web app and mobile app. It also needs to relay order information to another web app, which represents the digital signage. Lastly, we need a database to hold all the information regarding restaurants, their menus and their orders, along with user information.

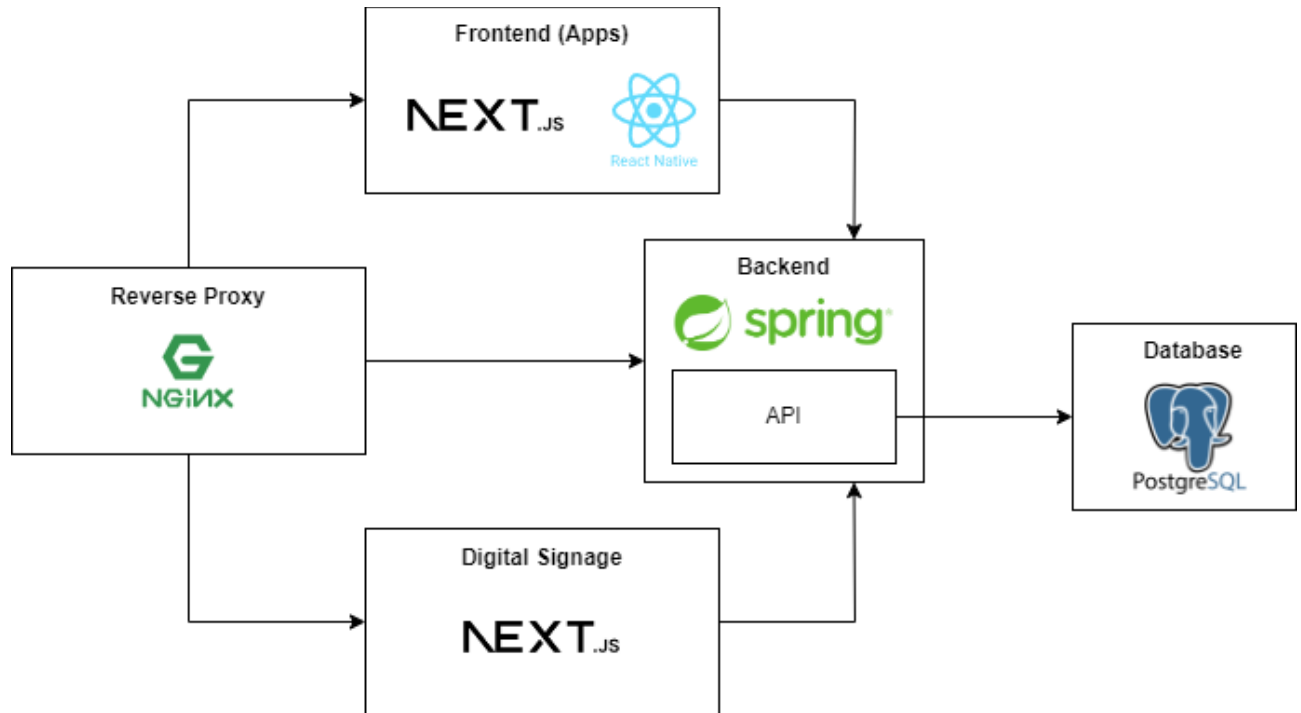
The Business entities include most of the entities described in the Domain Model, with the exception of the Manager, Client, LoggableUser and SystemUser, which will be merged into one entity, User.

The number of services available in the Business layer is still unclear, but regarding controllers, we expect to have at least 4: RestaurantController, MenuController, AuthController (for authentication) and OrderController.

We also want to implement push notification both on the web and in the mobile apps, so that the clients know when their order is ready to pick up.



4.3 Deployment architecture



For the front-end, we used React to develop both the client and the management web applications, and for the mobile app available for clients we used React Native.

Regarding the backend, we used Spring to develop an API capable of providing our service not only to our web apps but also to mobile. Spring's capability for dependency injection makes building APIs a much easier process.

To make the frontend and API available, we use Nginx to serve as a reverse proxy.

To save our data, we need a database that allows for fast access and is also consistent and available, so we chose PostgreSQL.

The deployment is made using docker-compose, with a container for each component: frontend (for the client, manager, and digital signage websites), backend (where the api is), database and Nginx reverse proxy.

5 API for developers

Order Operations pertaining to orders			^
PUT	/api/v1/orders/{restaurantId}/{orderId}/{status}	Update order status	▼
GET	/api/v1/orders	Get all orders	▼
POST	/api/v1/orders	Post an order	▼
GET	/api/v1/orders/{id}	Get order by id	▼
GET	/api/v1/orders/restaurant/{restaurantId}	Get orders by restaurant	▼
GET	/api/v1/orders/inmaking	Get in-making orders	▼
Menu Operations pertaining to menus			^
GET	/api/v1/menus/{id}	Get menu by id	▼
GET	/api/v1/menus/restaurant/{restaurantId}	Get restaurant menu	▼
User Login User login operations			^
POST	/api/v1/auth/signup	Create a new user	▼
POST	/api/v1/auth/login	Login a user	▼
User Order Operations on Orders made by users			^
GET	/api/v1/user/orders/{userId}	Get all orders made by the user	▼
POST	/api/v1/user/orders/{userId}	Place an order	▼
GET	/api/v1/user/orders/{userId}/{id}	Get order by id	▼
Category Operations pertaining to categories			^
GET	/api/v1/categories/{id}	Get category by id	▼
GET	/api/v1/categories/menu/{menuId}	Get categories by menu	▼
Restaurant Operations pertaining to restaurants			^
GET	/api/v1/restaurants	Get all restaurants	▼
POST	/api/v1/restaurants	Create restaurant	▼
POST	/api/v1/restaurants/{id}/menu	Import menu	▼
GET	/api/v1/restaurants/{id}	Get restaurant by id	▼
DELETE	/api/v1/restaurants/{id}	Delete restaurant by id	▼
GET	/api/v1/restaurants/open	Get all open restaurants	▼
Item Operations pertaining to items			^
GET	/api/v1/items	Get all items	▼
GET	/api/v1/items/{id}	Get item by id	▼
GET	/api/v1/items/restaurant/{restaurantId}	Get items by restaurant	▼
GET	/api/v1/items/category/{categoryId}	Get items by category	▼

6 References and resources