Idly, Vadai, Sambar!

Engineering Design Document

Table of Contents

1.0 High Level Design

On a high level, the solution being offered can be divided into three essential components:

- Front end
- Application Programming Interface
- Back end

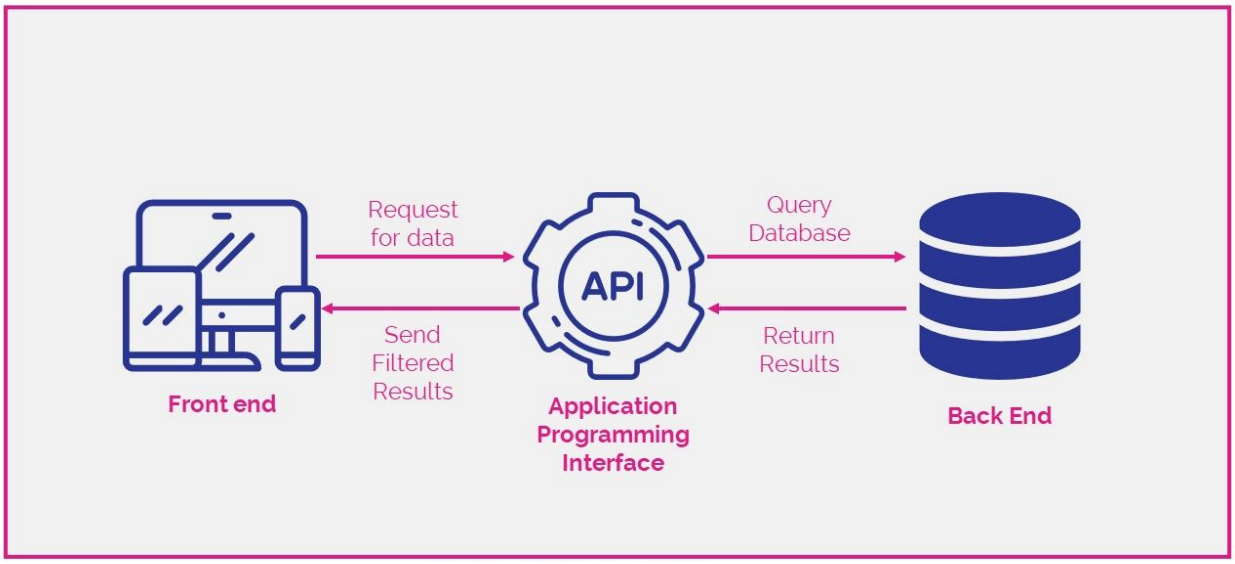

Fig 1 : High Level Diagram

The front end is composed of all the views that a user will see. A user will be interacting with the front-end, without having to understand the implementation of the backend. The solution encompasses a front-end in the form of a Web Application that is written in **React.js**.

The Application Programming Interface (API) is a layer that is meant to act as an abstraction to obtain that the front-end requests from the database(s). The front-end makes all requests to the API using appropriate URLs. The API has implementation logic that queries the database, filters the results and returns the response. The solution uses an API written in **Node.js**.

The backend consists of database(s) that handle that are source of data storage. This layer, along with the API layer handles the structure of storage of data and is responsible for retrieval / creation of new documents as and when instructed to do so. The solution uses a **MongoDB database** to serve as a backend.

2.0 Component Level Design

The components were created in-sync with the user stories. The components are:

1. Place Order Component
2. View Menu Component
3. Update Menu Component
4. Get Items from Menu Component
5. Update Items from Menu Component
6. Manage Online Orders component
7. Update Delivery Status component
8. View Order History component

2.1 Place Order Component

The place order component is a mockup of the customer-side logic as the data prescribed by the QEats application had to be restructured, making the logic written in QEats unusable. For the sake of uniformity, the routes were rewritten to accommodate the needs of the solution.

While this component has no front end as it does not come under the scope of the solution, a working command-line API is written with error-checking, et al.

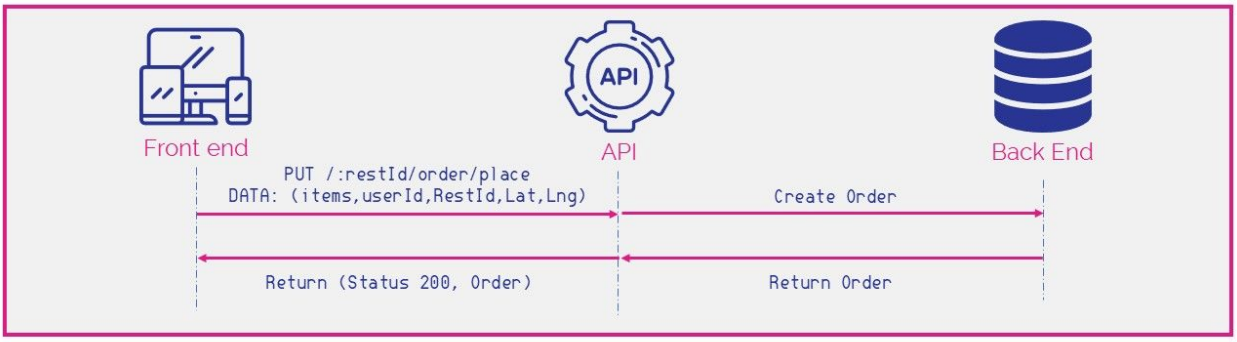

Fig 2 : Placing Order Interaction Diagram

2.2 View Menu Component

The View Menu component helps a restaurant manager keep a tab on the menu. It helps in identifying quantities of items on the menu in order to manage the inventory of items to feed the demand. It also gives an idea about trends so that appropriate offers can be put in place to push items that are not fast moving.

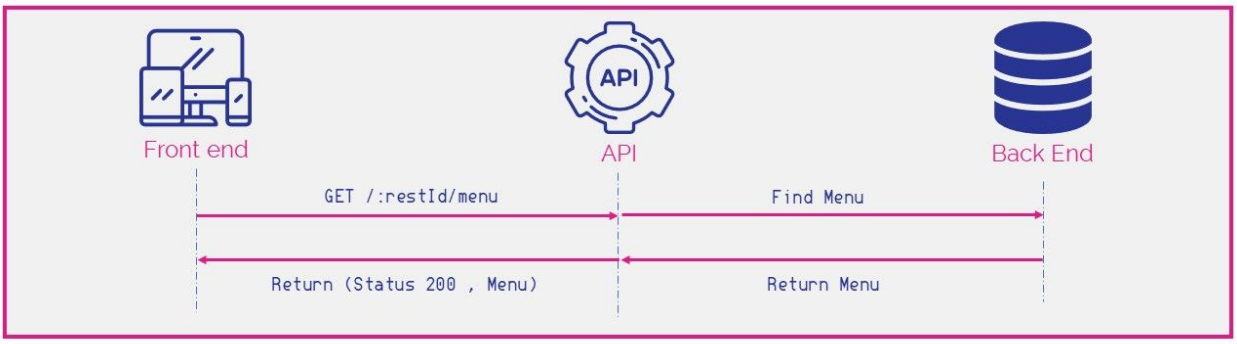

Fig 3 : View Menu Component Interaction Diagram

2.3 Updating Menu Component

The solution gives the restaurant manager an option to add new items to their menu. The added item must have details like name, price, image, quantity available and time to prepare the item.

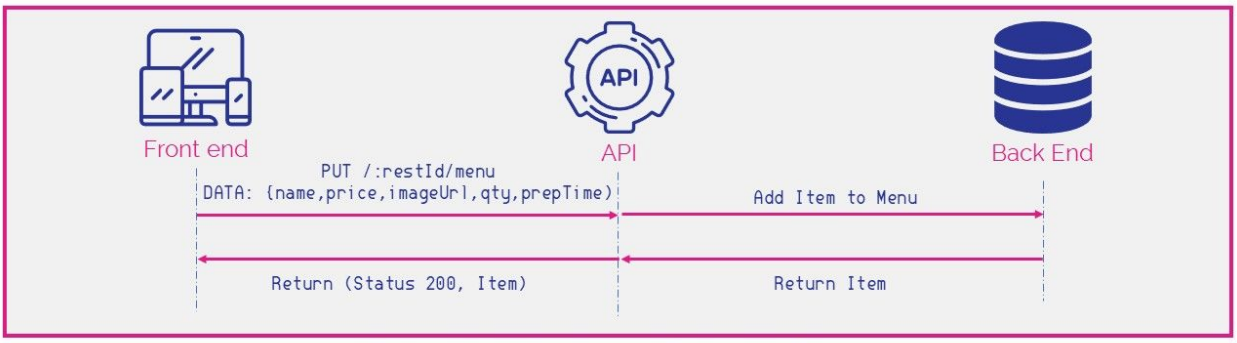

Fig 4 : Update Menu Component Interaction Diagram

2.4 Get Items from Menu Component

This component allows a manager or customer get an item from the menu to view details like name, price, time taken to prepare, and image of the item.

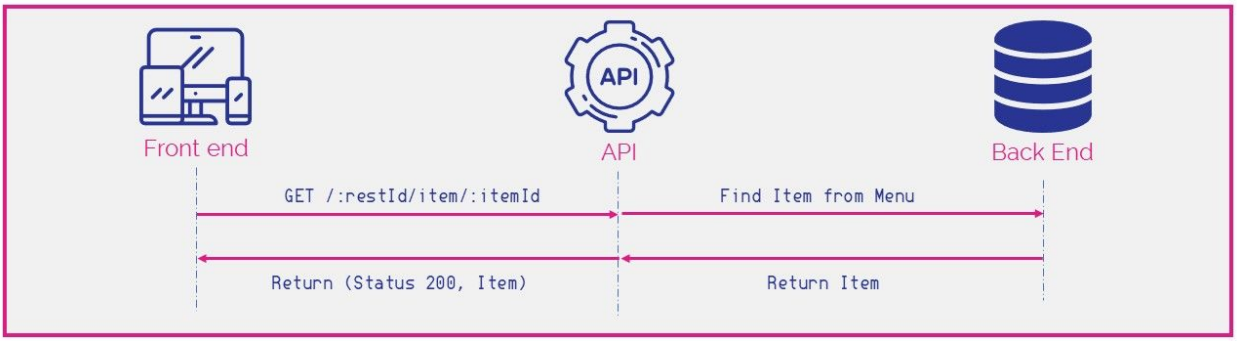

Fig 5 : Get Item From Menu Component Interaction Diagram

2.5 Update Items from Menu Component

This component allows a restaurant manager to edit an item to signify a change in price, quantity available, preparation time or imageUrl. It also helps to make some items temporarily unavailable. This component can be used to change one or many fields of an item or delete the item.

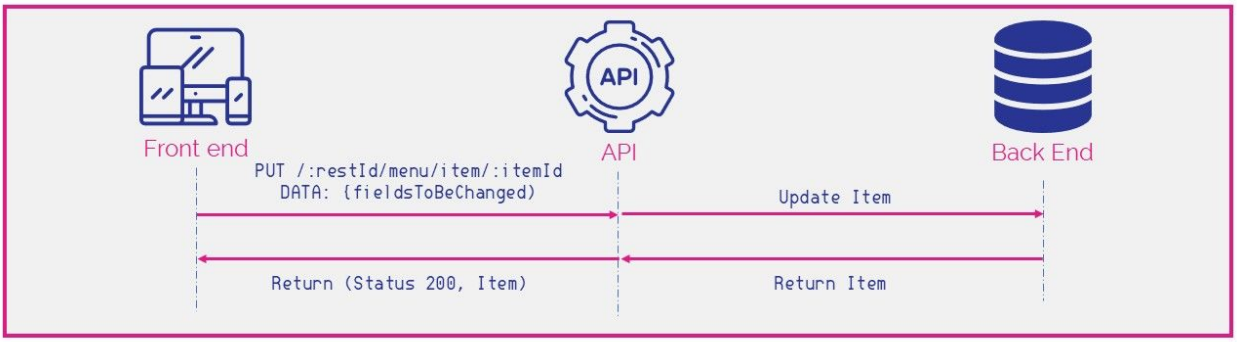

Fig 6 : Update Items from Menu Interaction Diagram

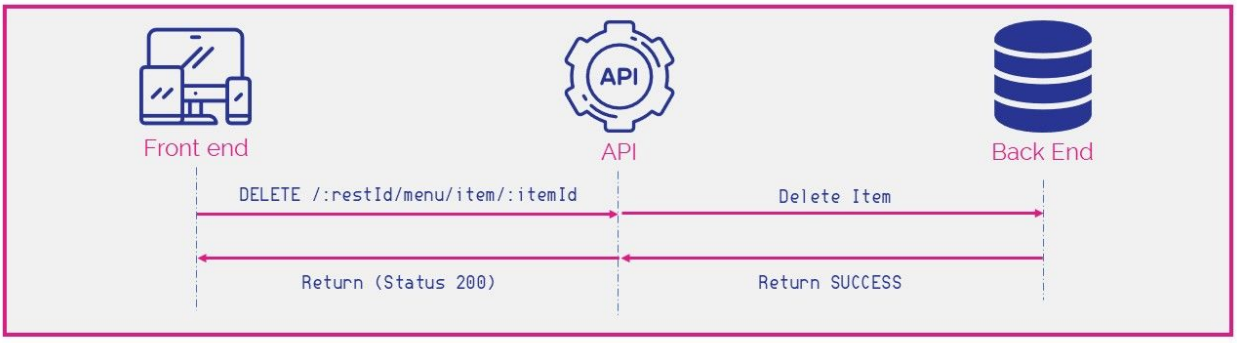

Fig 6b. Delete Item from Menu Component Interaction Diagram

2.6 Manage Online Orders Component

Extending the QEats food ordering application solution that was worked upon during the Crio Summer of Doing micro-experience, this solution allows a restaurant to act upon a customer's order. Details of the order along with distance from the customer can be used as a metric to decide if an order can be served or not. The order service is in-sync to the inventory so only those items that can are available can be ordered because unavailable items won't be visible in the customer View of the Restaurant Menu.

A restaurant can either accept or reject a customer's order.

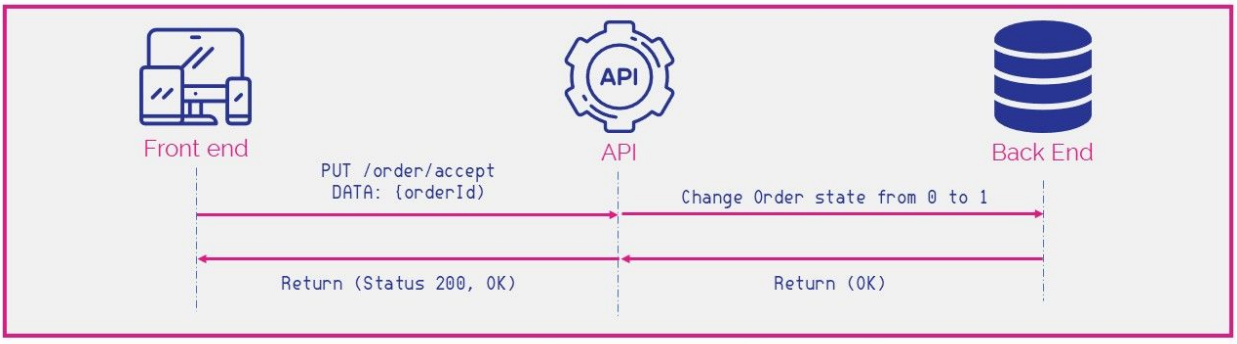

Fig 7a : Restaurant Accepting an Order Interaction Diagram

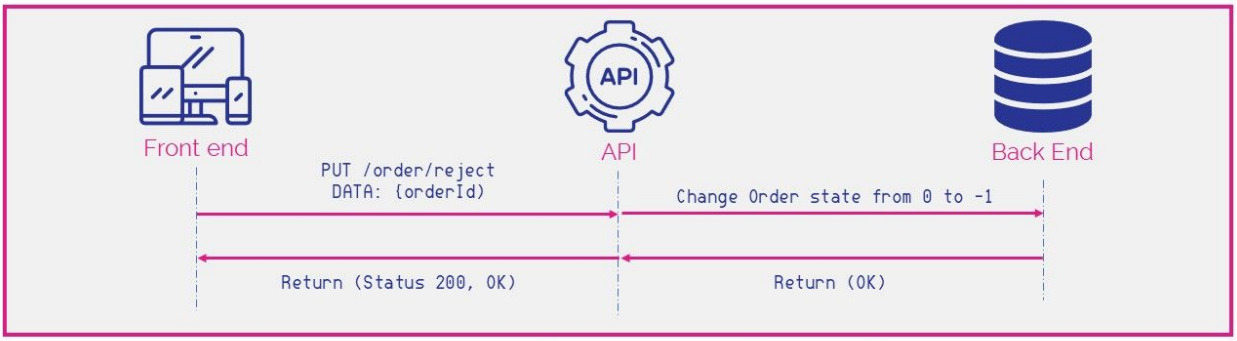

Fig 7b : Restaurant Accepting an Order Interaction Diagram

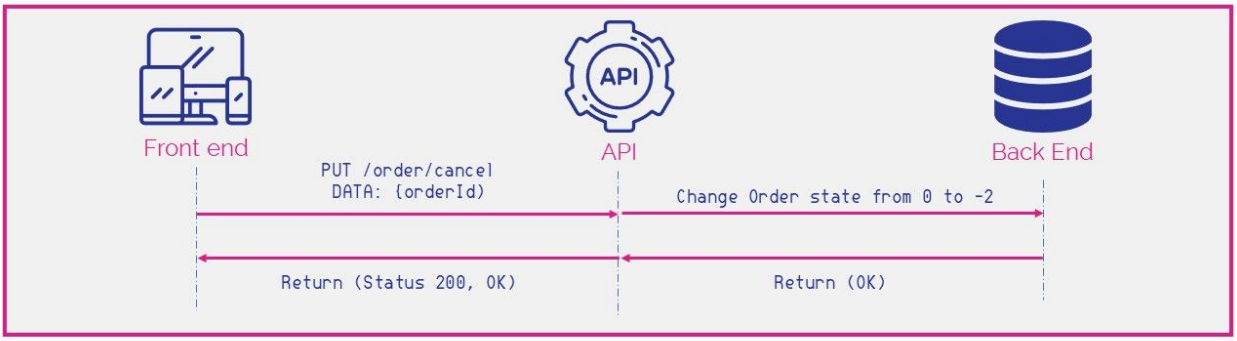

Fig 7c : User cancelling an Order Interaction Diagram

2.7 Update Delivery Status Component

Once a restaurant accepts an order placed by a customer, the order is considered to be in ACTIVE state until it leaves the restaurant. Between placing the order and the order being picked up, a restaurant has the liberty to change order status to notify the customer about the status of their order.

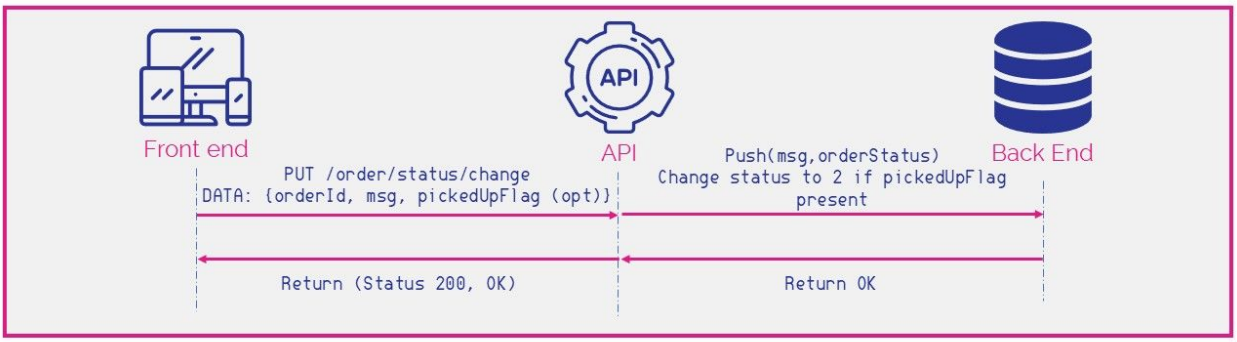

Fig 8 : Update Delivery Status Interaction Diagram

2.8 View Order History Component

This component helps a restaurant keep track of the orders that they've served.

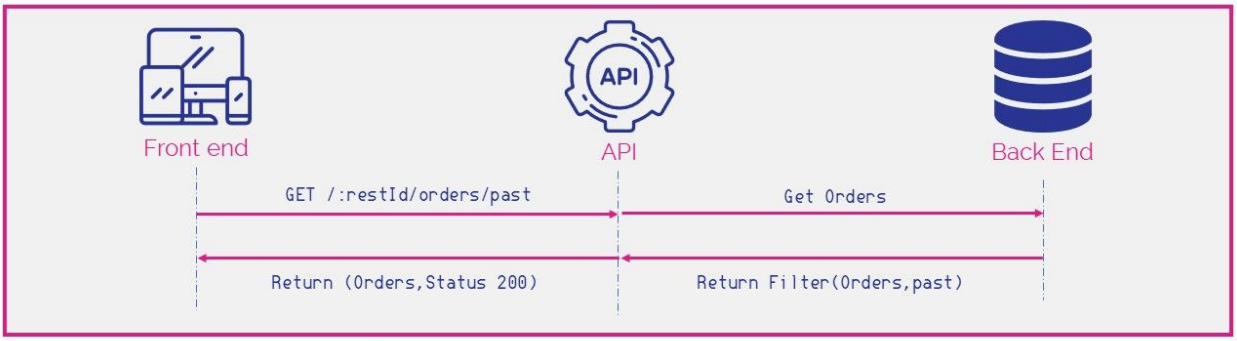

Fig 9 : View Order History Component Interaction Diagram