Pizza restaurant

Kerrit Hadile Mochet Hugo Heni Beya

Overview

Our website offers a unique experience for pizza lovers! ey’re served at nearby restaurants. You can suggest new pizza creations, add pizzas and restd information about each restaurant where your favorite slices are available. Plus, our search feature lets you quickly find any pizza you're craving.

Dive into the world of pizzas with us and enjoy a seamless, interactive way to enhance your dining and tasting adventures!

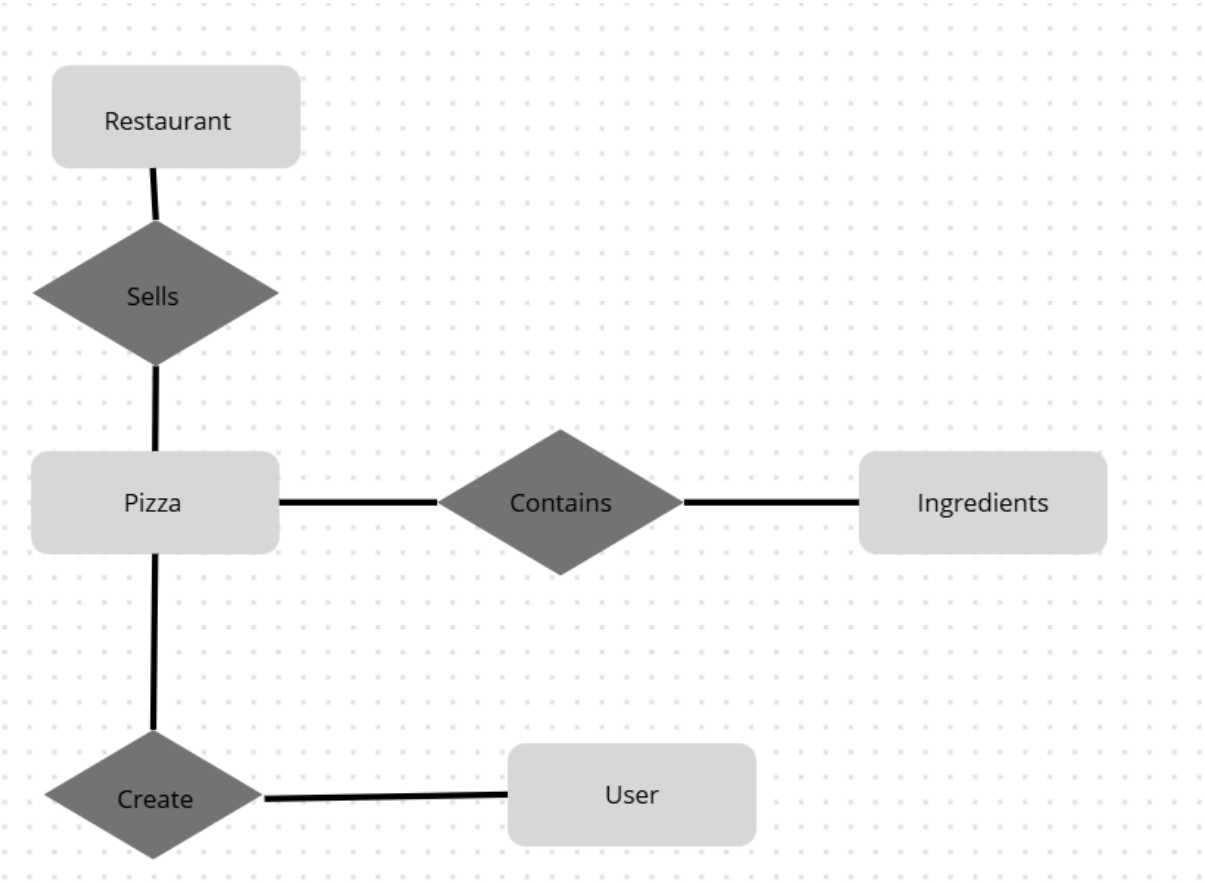
Summary

One Pager

Summary

Date	Sep 1, 2025
01	Database Structure
02	Key Relashionships
03	Table Structure Diagram
04	SQL
05	First Gantt Diagram
06	Use Case Diagrams
07	Wireframe Diagrams
08	Activity Diagrams
09	Sequence Diagrams
10	Component Diagrams
11	Class Diagram
12	Final Gantt Diagram

Database Structure



Relevant Resources
Ingredients : Components of each pizza
Restaurant : Etablishement where clients can make their pizza
User: People who can create their own pizza
Pizza : Represents differents pizza made by clients

KEY RELATIONSHIP

"Contains" tracks which ingredient are required for each pizza.

"Sells" links pizzas to restaurants that sells them.

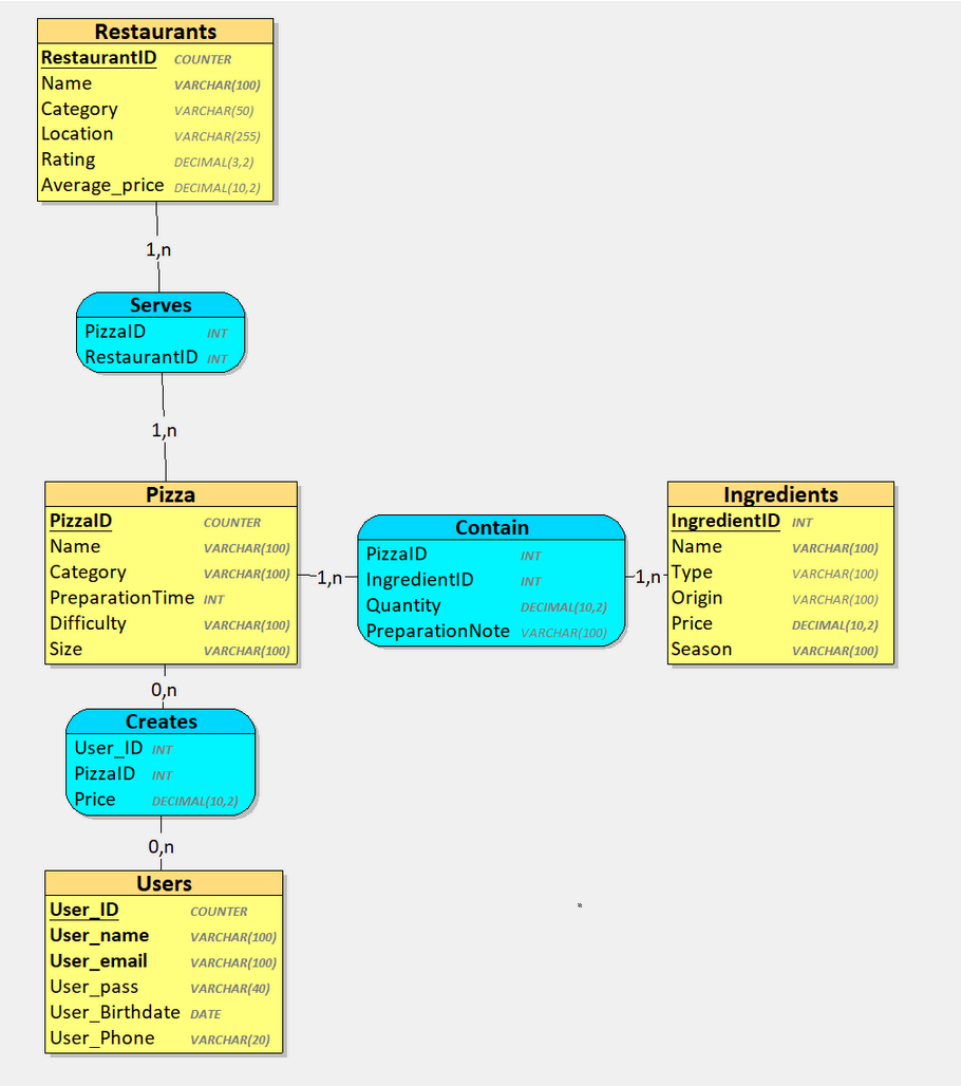
"Create" represents the connection between users and the pizzas they prepare

TABLE STRUCTURE DIAGRAM

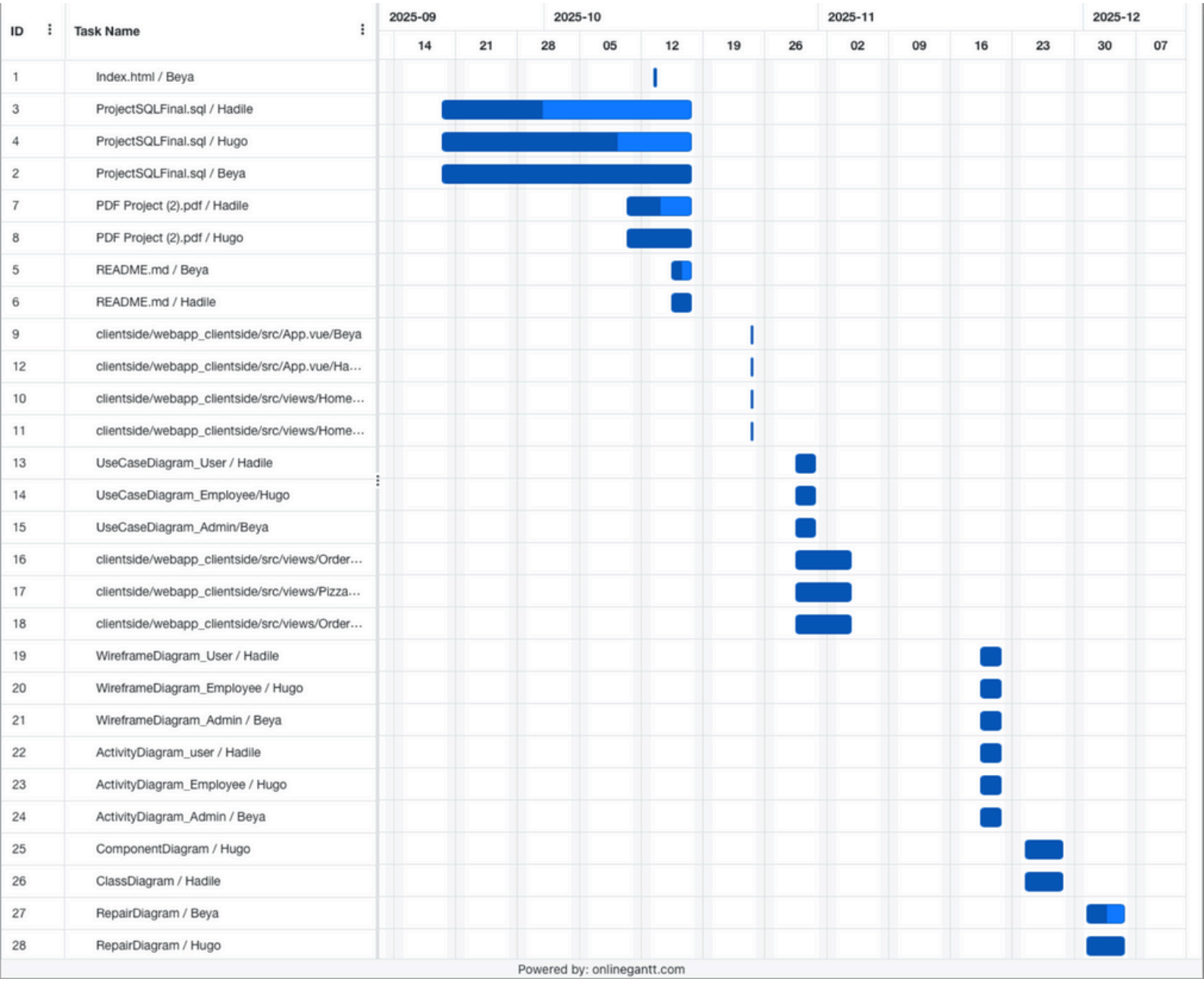
- **Pizza:** Stores details about each pizza including its name, category, preparation time and difficulty.
- **Ingredient :** Keeps information about each ingredient, such as type, origin and price.
- User : Captures user data, including login credentials and profile details.
- **Restaurant :** Contains information about restaurants, such as location, categorie, and rating.

- **Contains :** Links Pizzas and Ingredients with their quantity. The client can delete an ingredient if he wants.
- **Sells :** It shows which restaurant serves which kind of pizza.

OUR TABLES

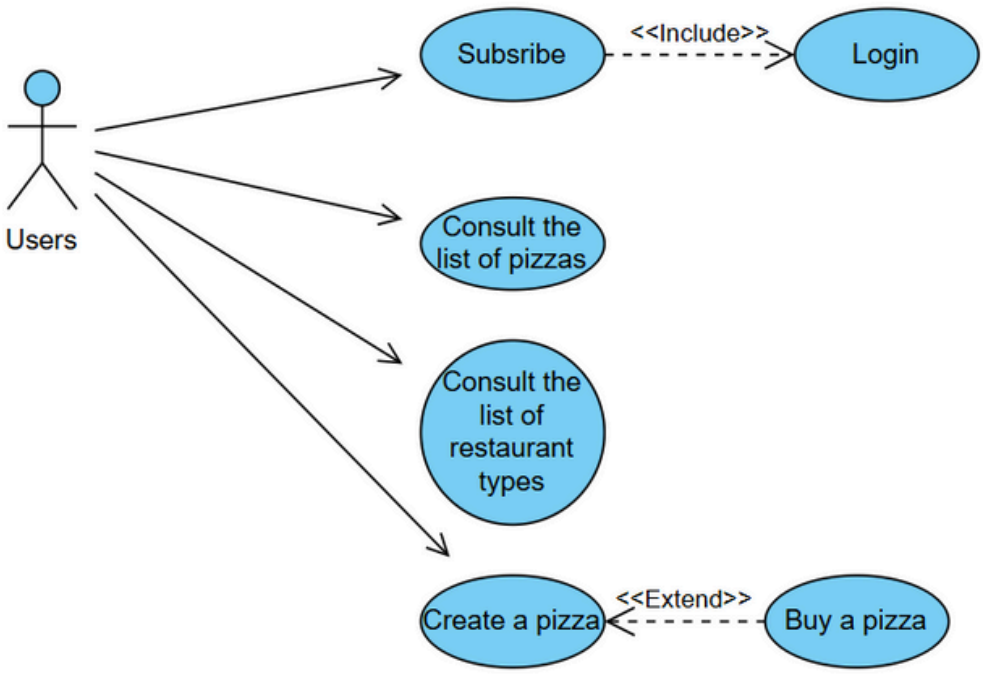

Restaurants
RestaurantID COUNTER
Name VARCHAR(100)
Category VARCHAR(50)
Location VARCHAR(255)
Rating DECIMAL(3,2)
Average_price DECIMAL(10,2)
1,n
Serves
PizzaID INT
RestaurantID INT
1,n
Pizza
PizzaID COUNTER
Name VARCHAR(100)
Category VARCHAR(100)
PreparationTime INT
Difficulty VARCHAR(100)
Size VARCHAR(100)
1,n
Contain
PizzaID INT
IngredientID INT
Quantity DECIMAL(10,2)
PreparationNote VARCHAR(100)
1,n
Ingredients
IngredientID INT
Name VARCHAR(100)
Type VARCHAR(100)
Origin VARCHAR(100)
Price DECIMAL(10,2)
Season VARCHAR(100)
0,n
Creates
User_ID INT
PizzaID INT
Price DECIMAL(10,2)
0,n
Users
User_ID COUNTER
User_name VARCHAR(100)
User_email VARCHAR(100)
User_pass VARCHAR(40)
User_Birthdate DATE
User_Phone VARCHAR(20)

GANTT DIAGRAM



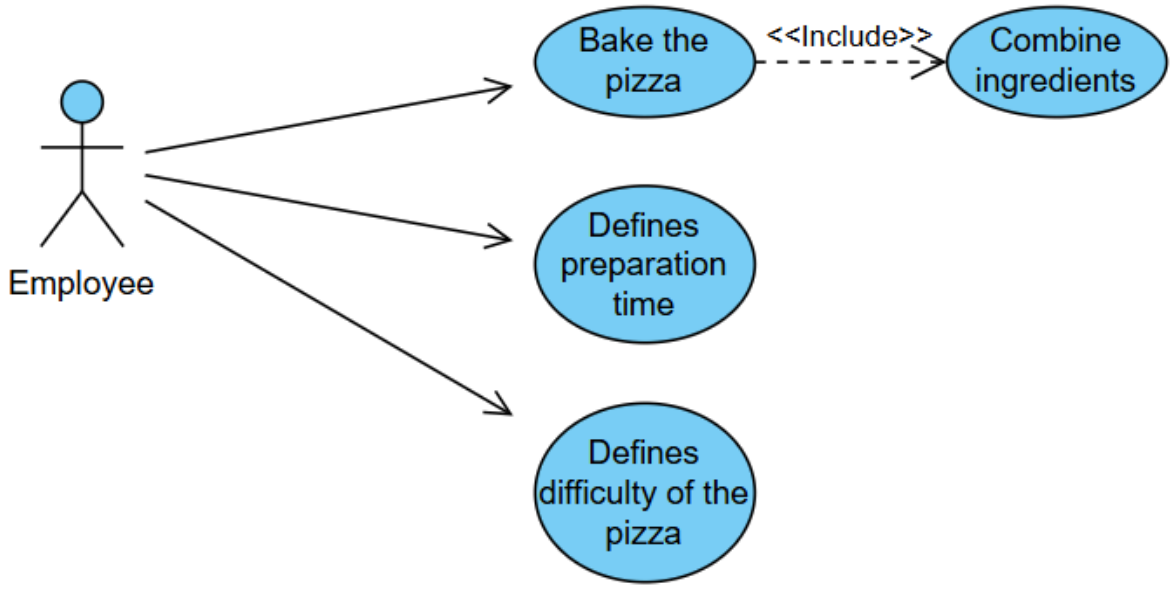
USE CASE DIAGRAM

HADILE



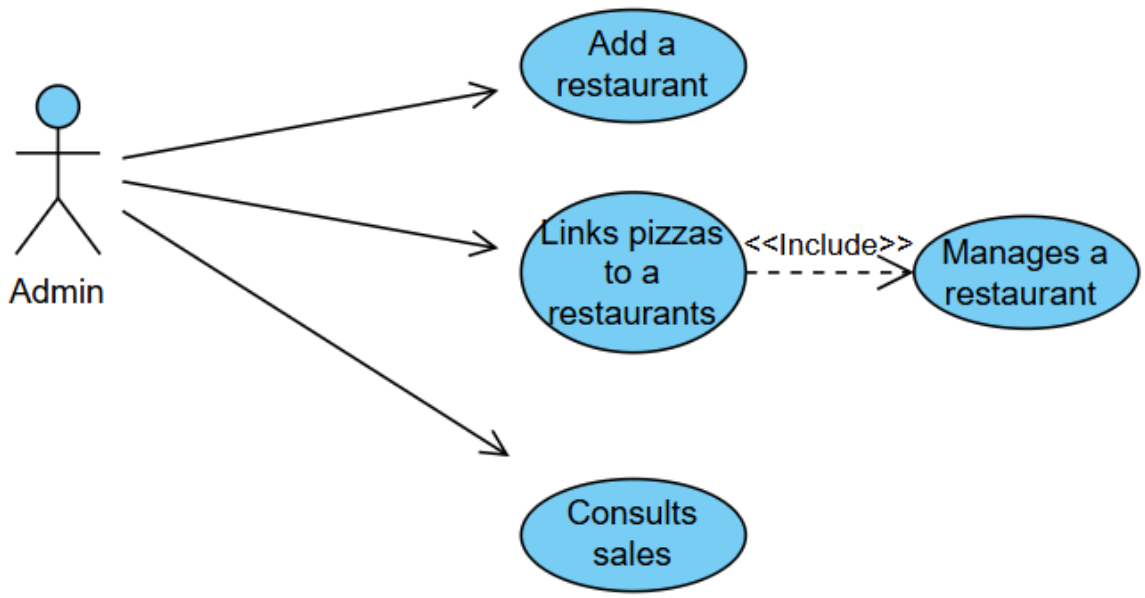
USE CASE DIAGRAM

HUGO



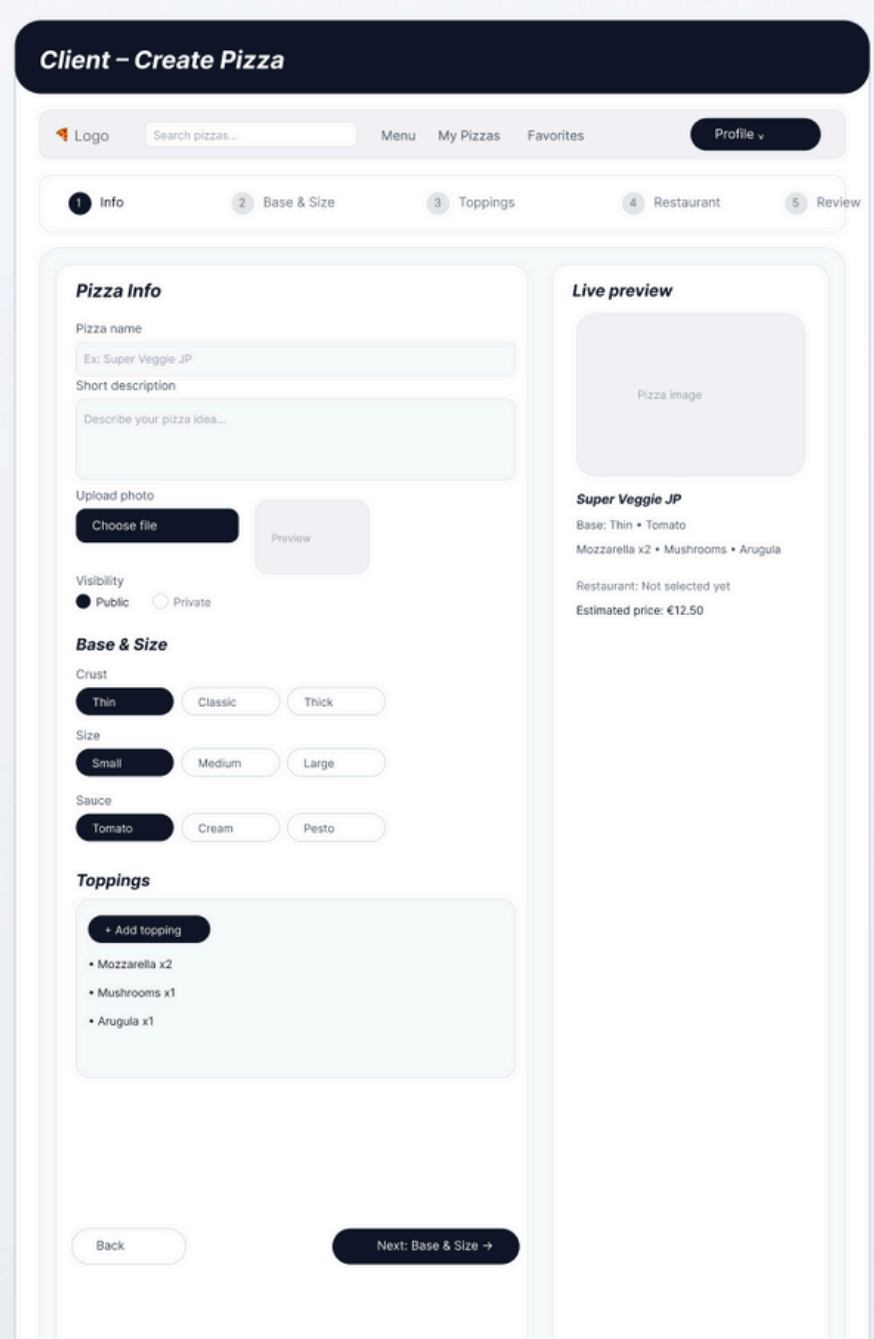
USE CASE DIAGRAM

BEYA



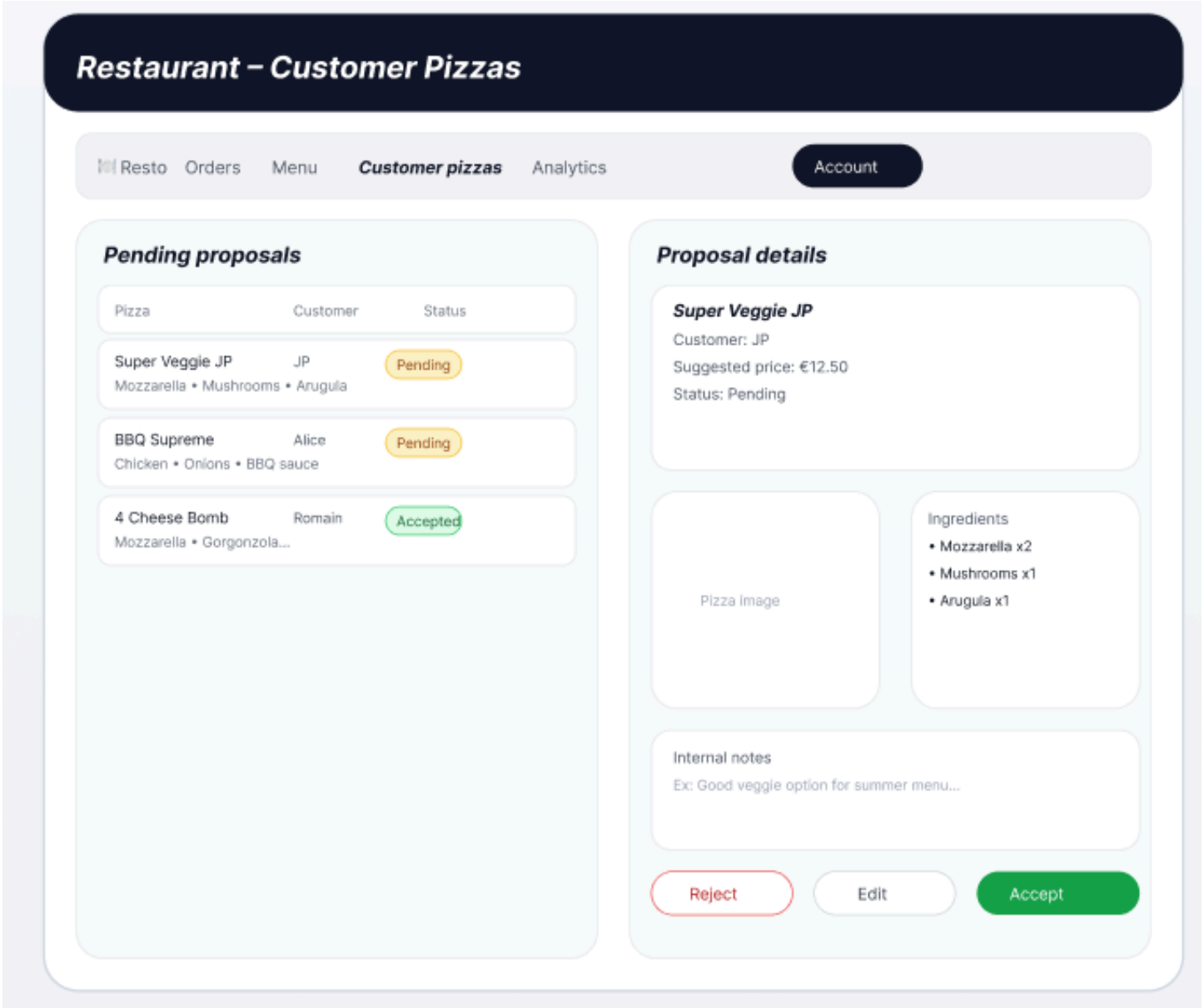
WIREFRAME DIAGRAM

HADILE



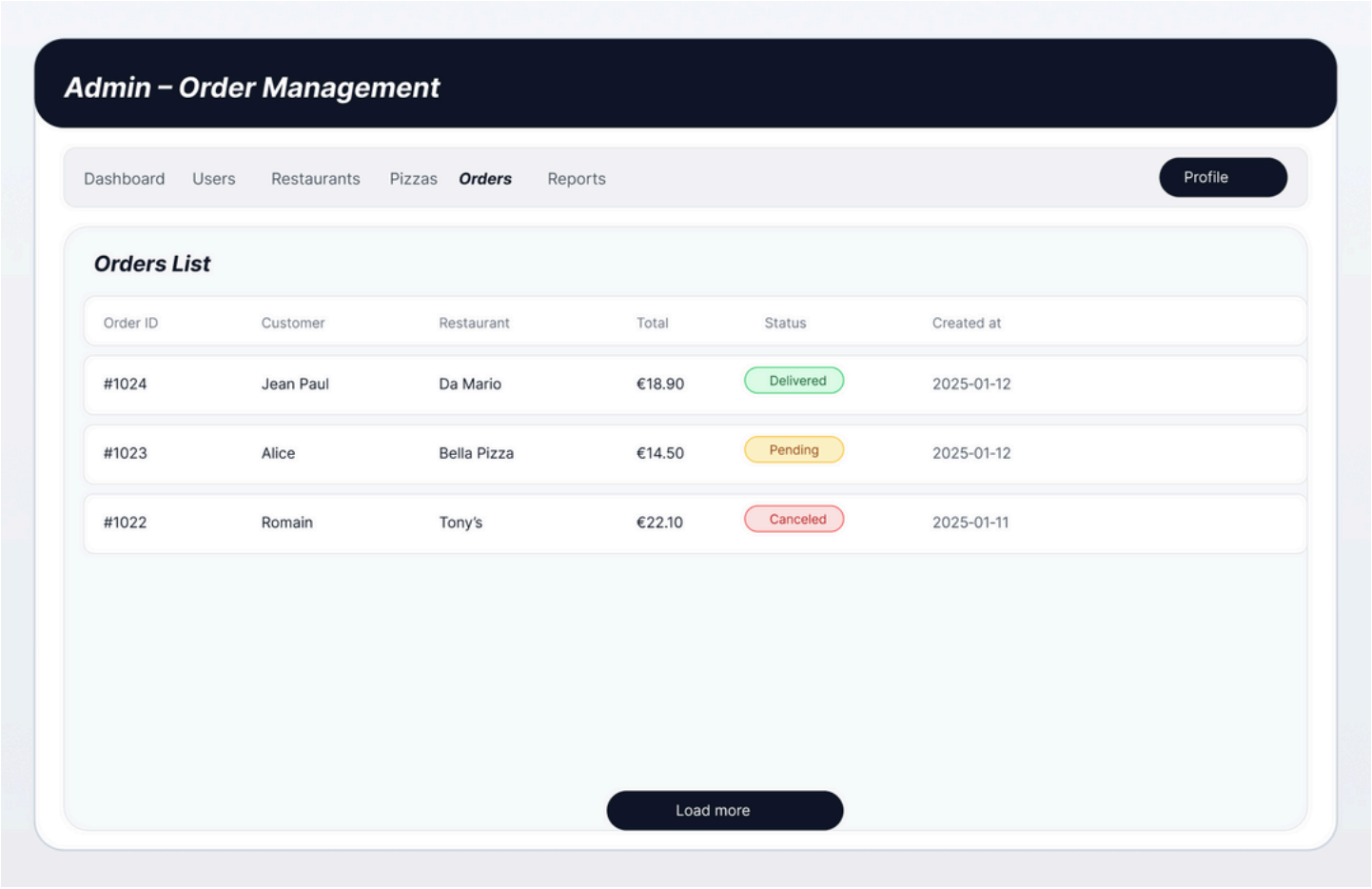
WIREFRAME DIAGRAM

HUGO



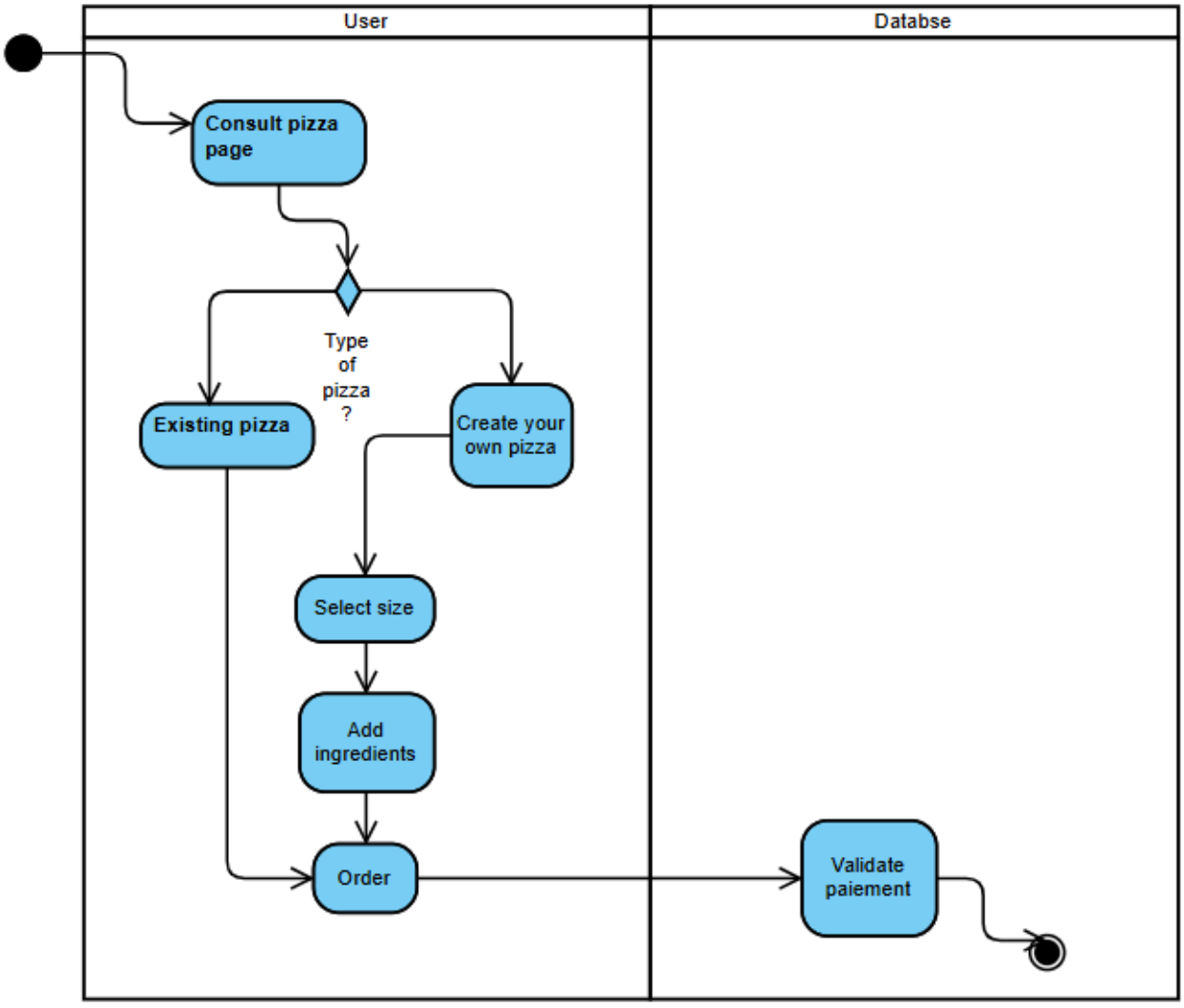
WIREFRAME DIAGRAM

BEYA



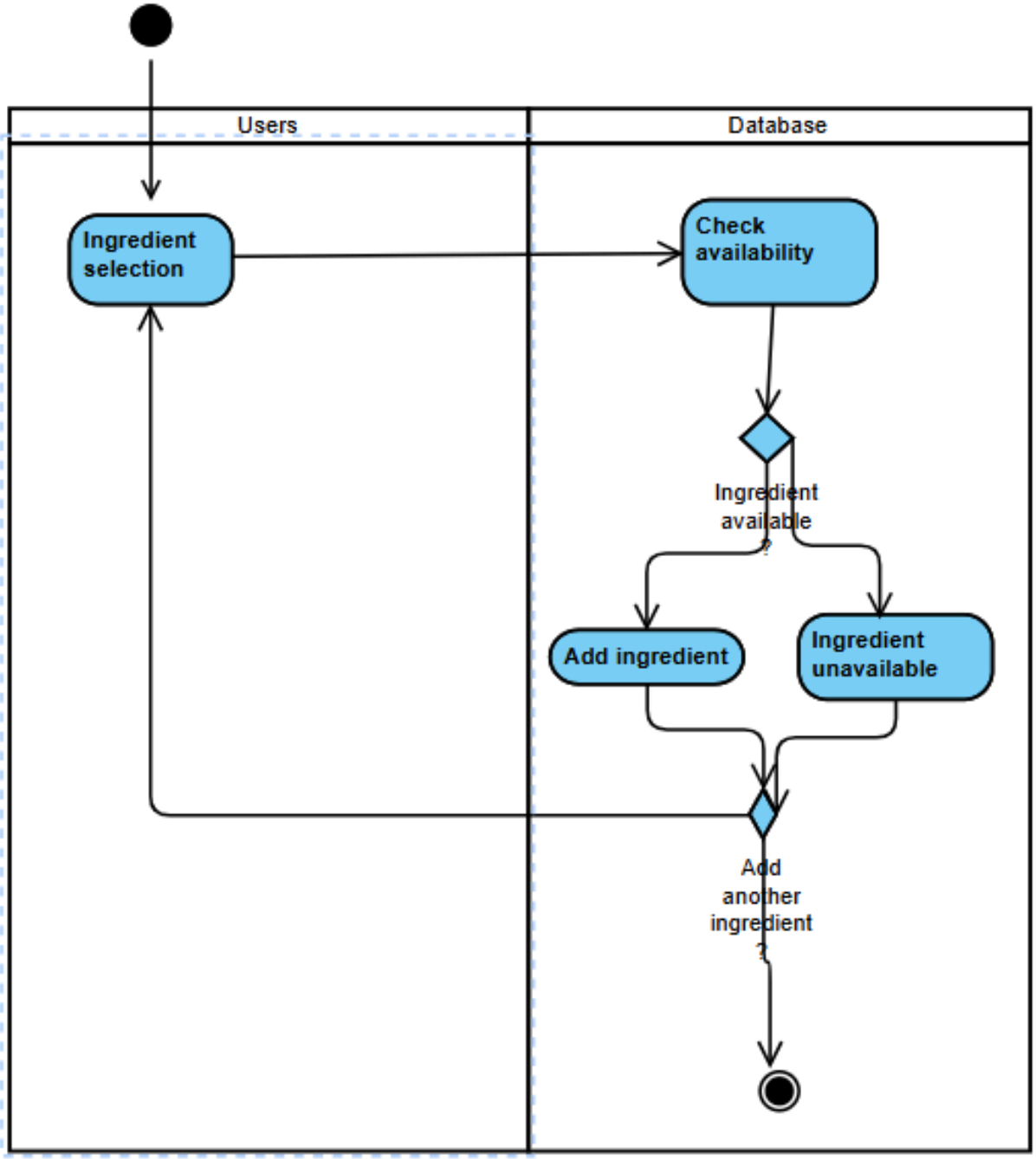
ACTIVITY DIAGRAM

HADILE



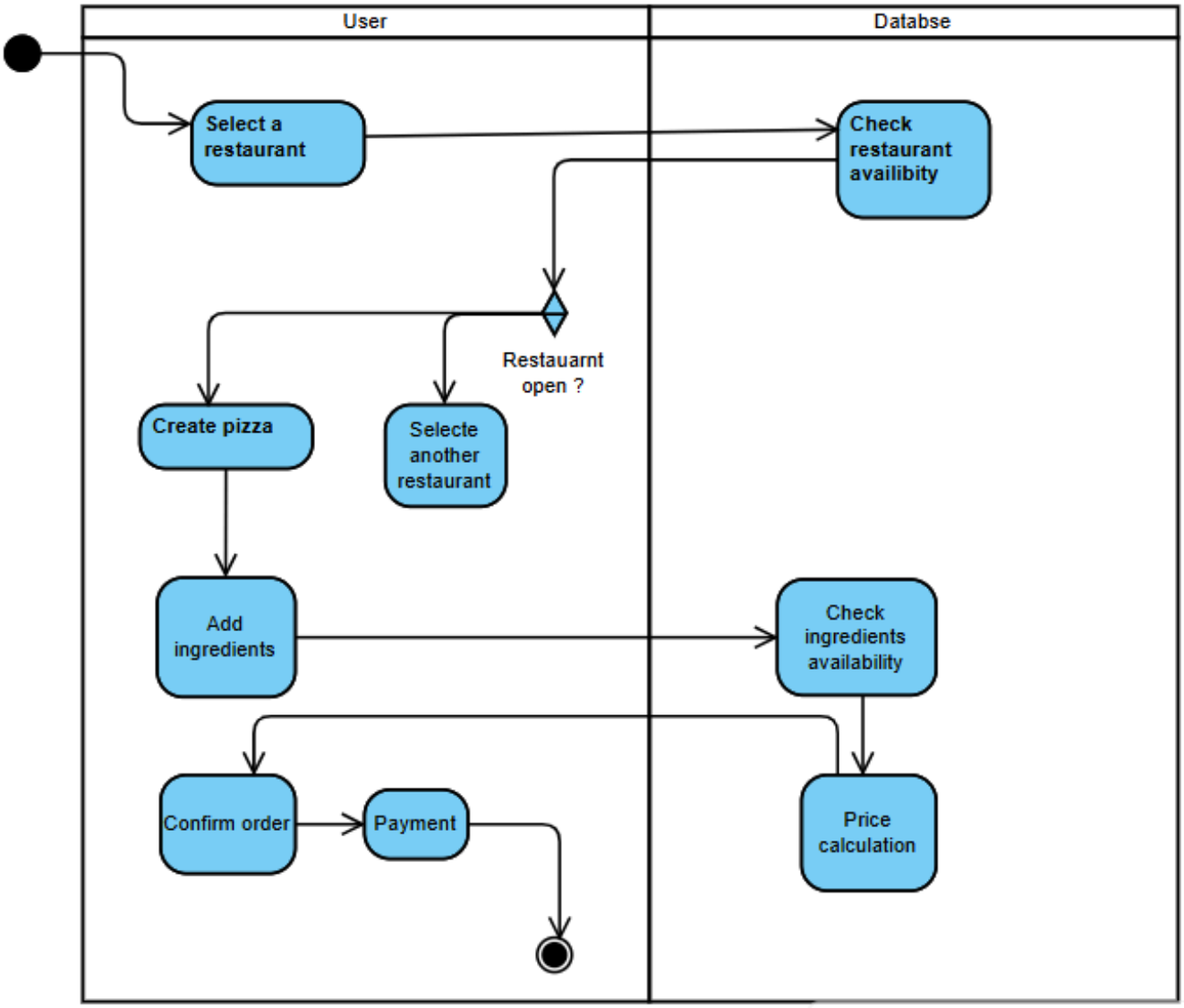
ACTIVITY DIAGRAM

HUGO


Users
Database
Ingredient selection
Check availability
Ingredient available ?
Add ingredient
Ingredient unavailable
Add another ingredient ?

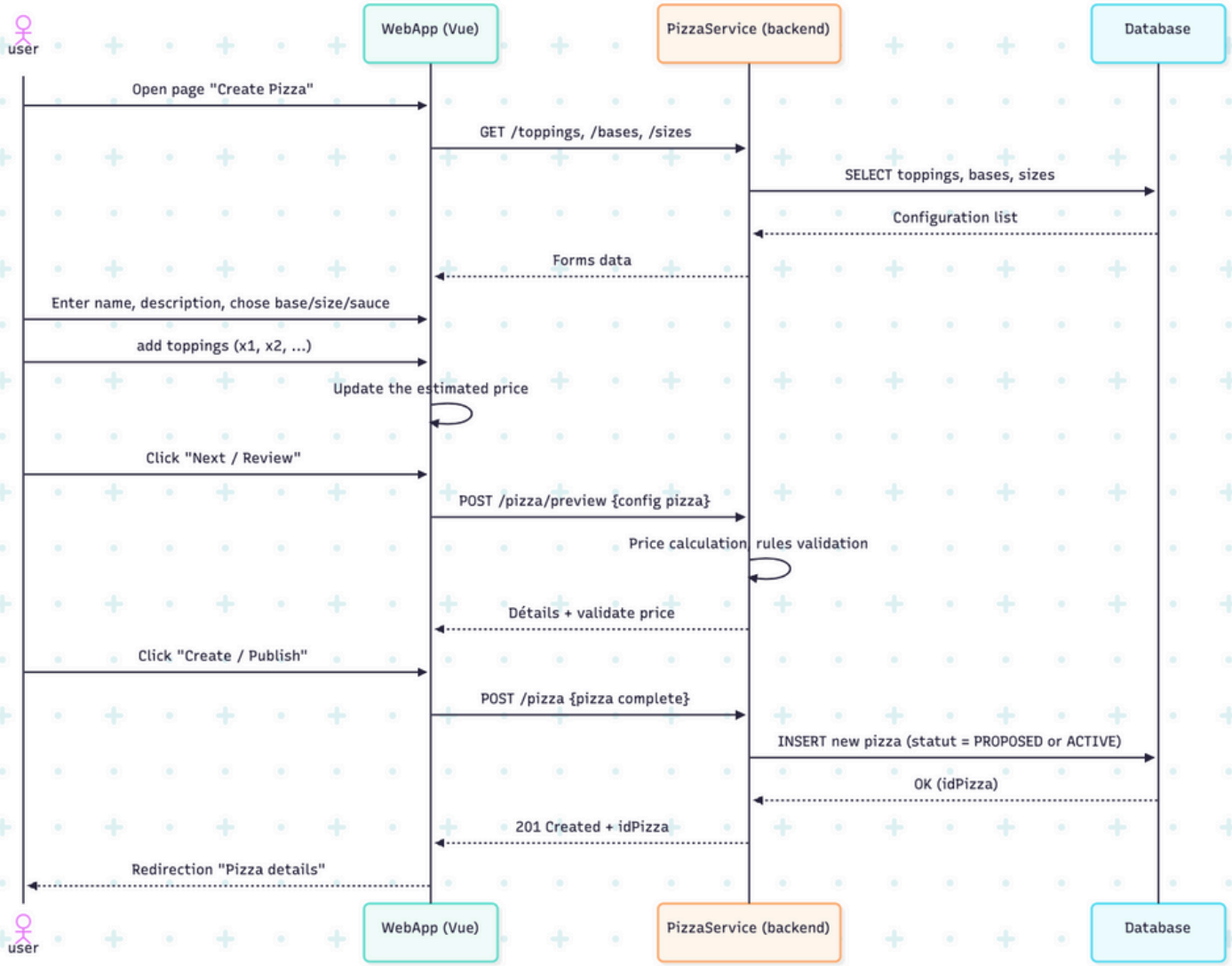
ACTIVITY DIAGRAM

BEYA



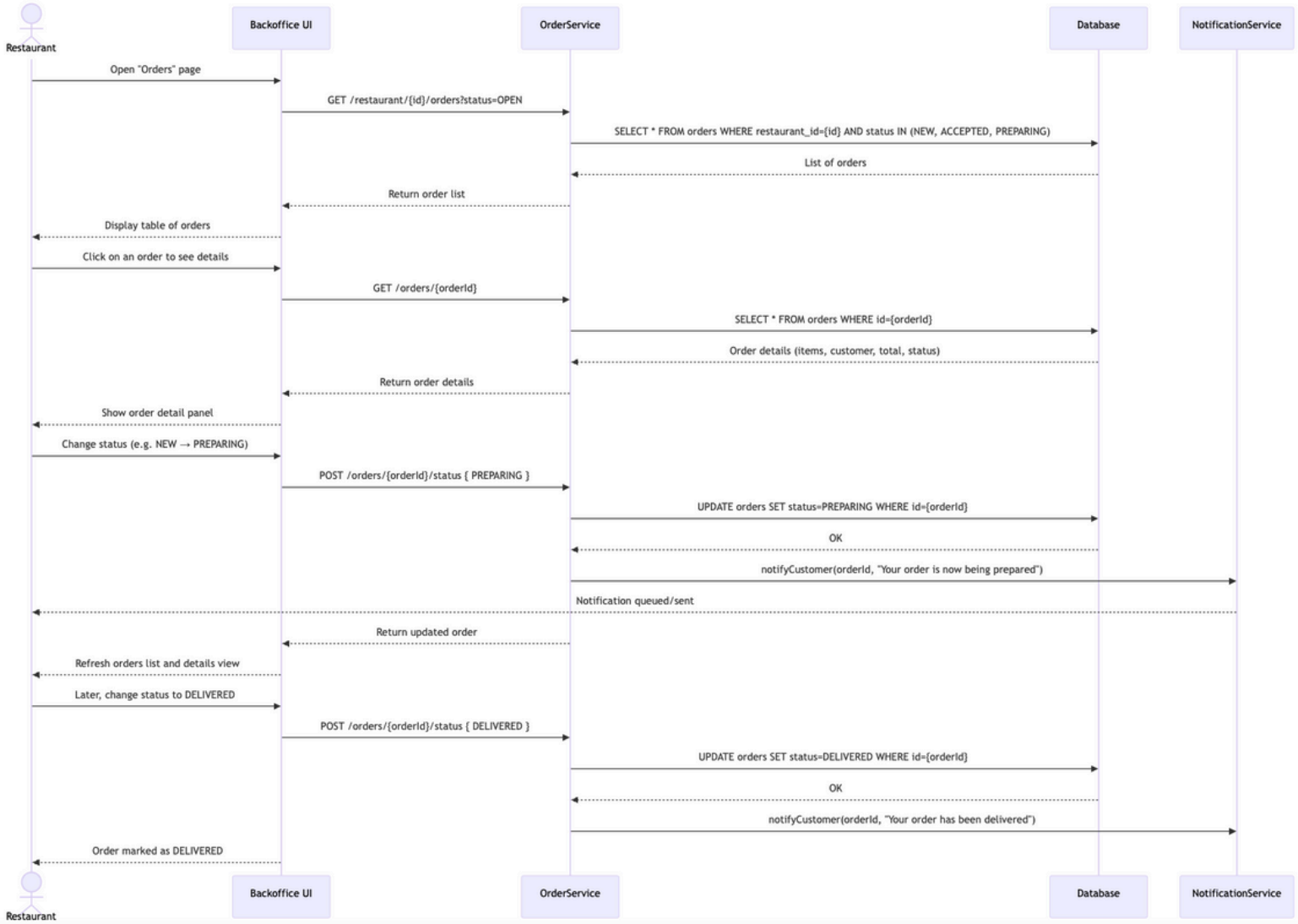
SEQUENCE DIAGRAM

HADILE



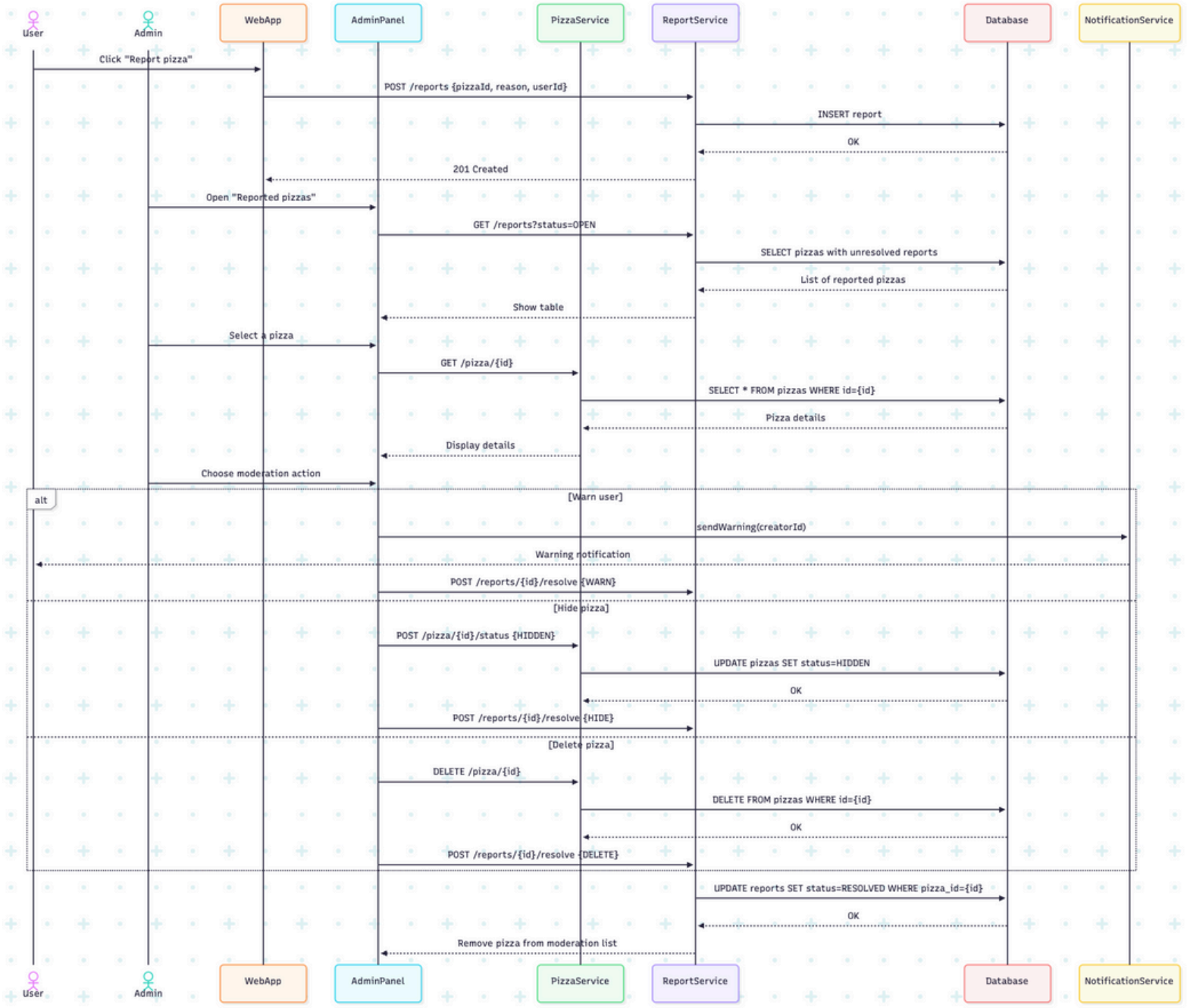
SEQUENCE DIAGRAM

HUGO



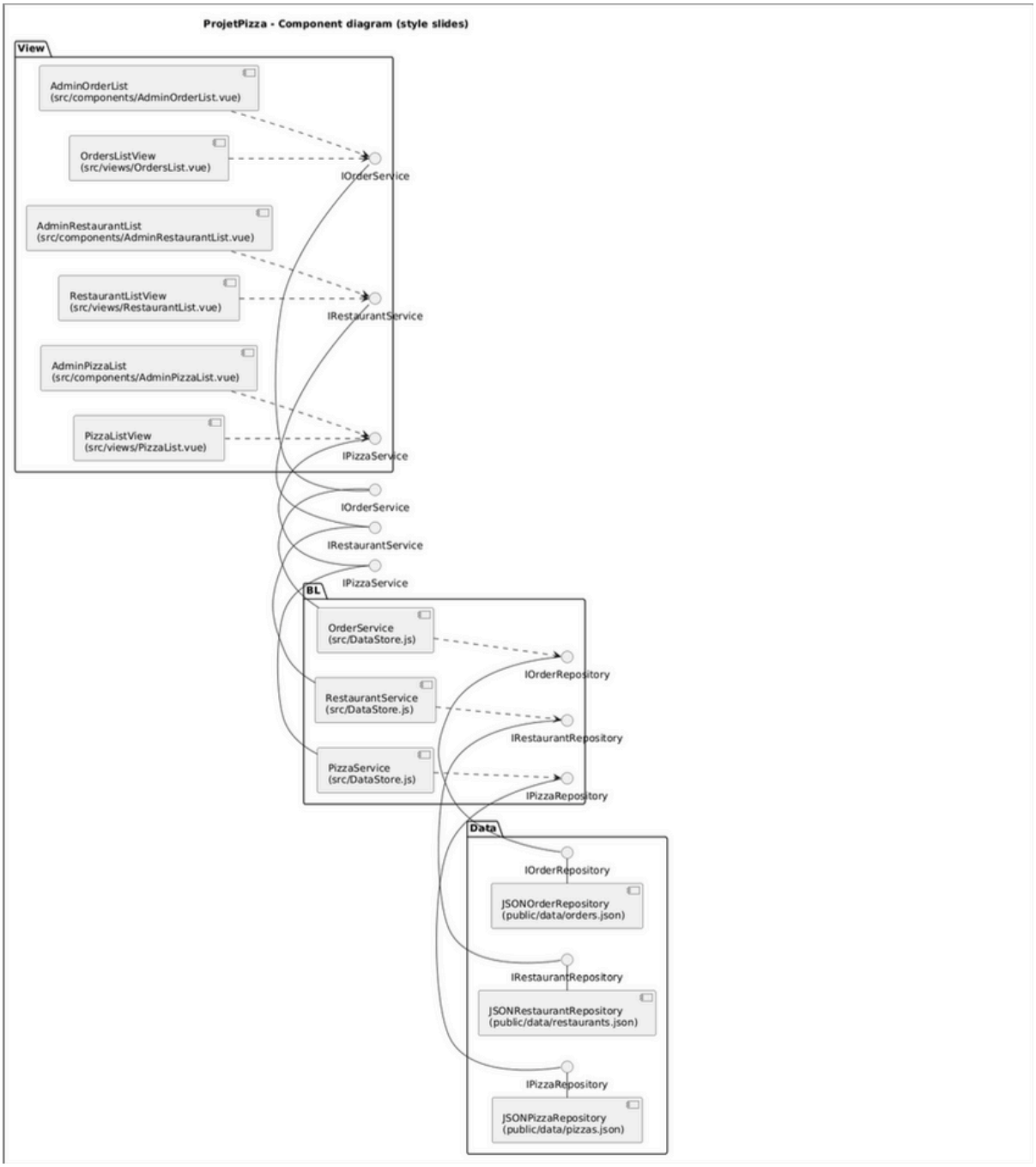
SEQUENCE DIAGRAM

BEYA



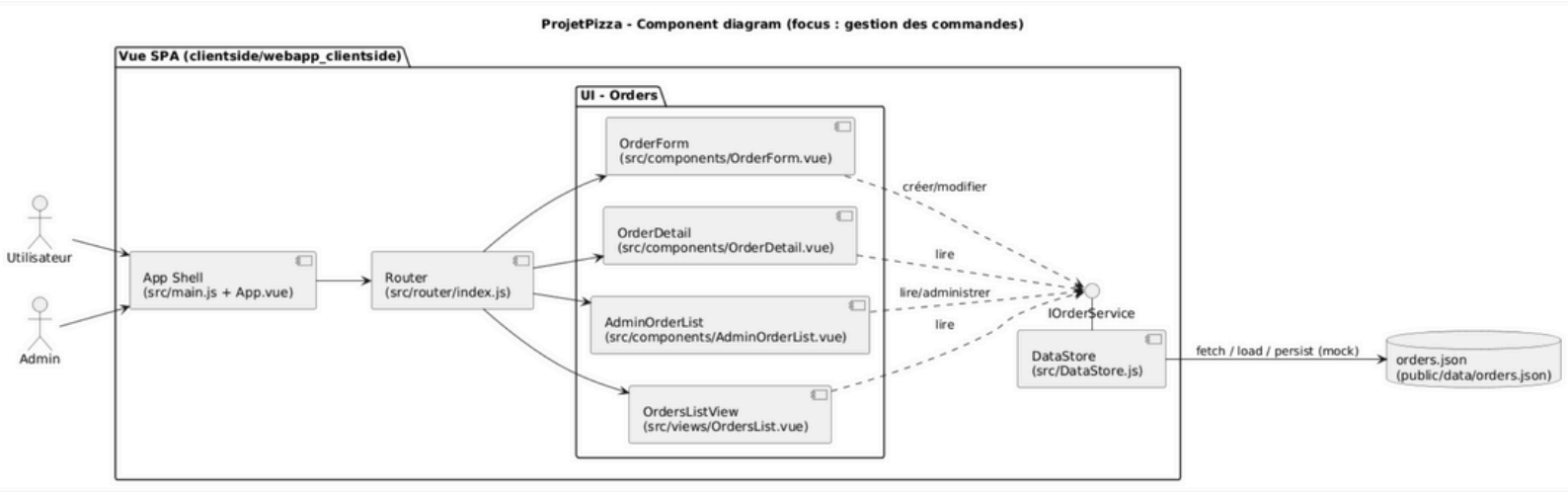
COMPONENT DIAGRAM

HADILE



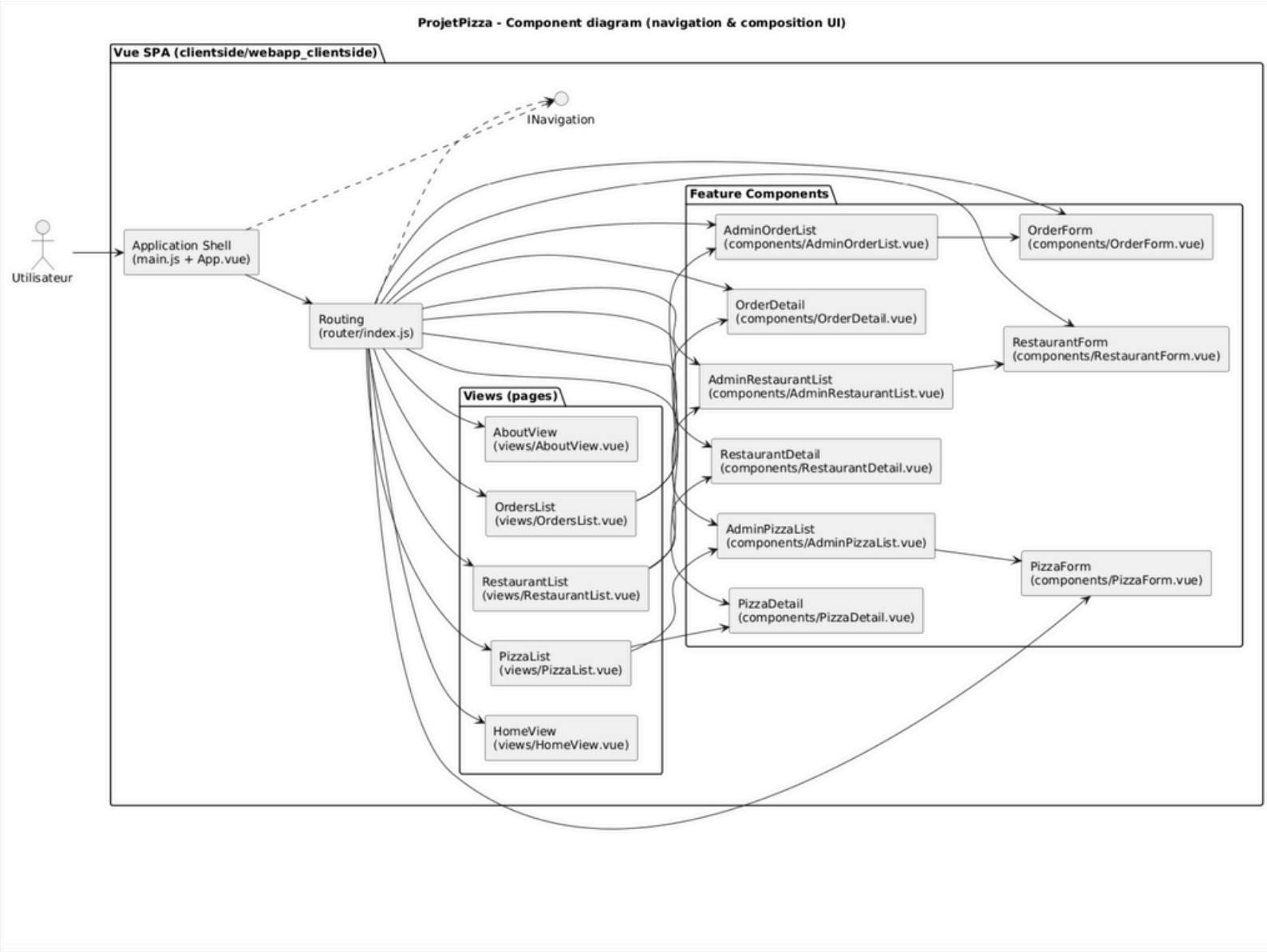
COMPONENT DIAGRAM

HUGO

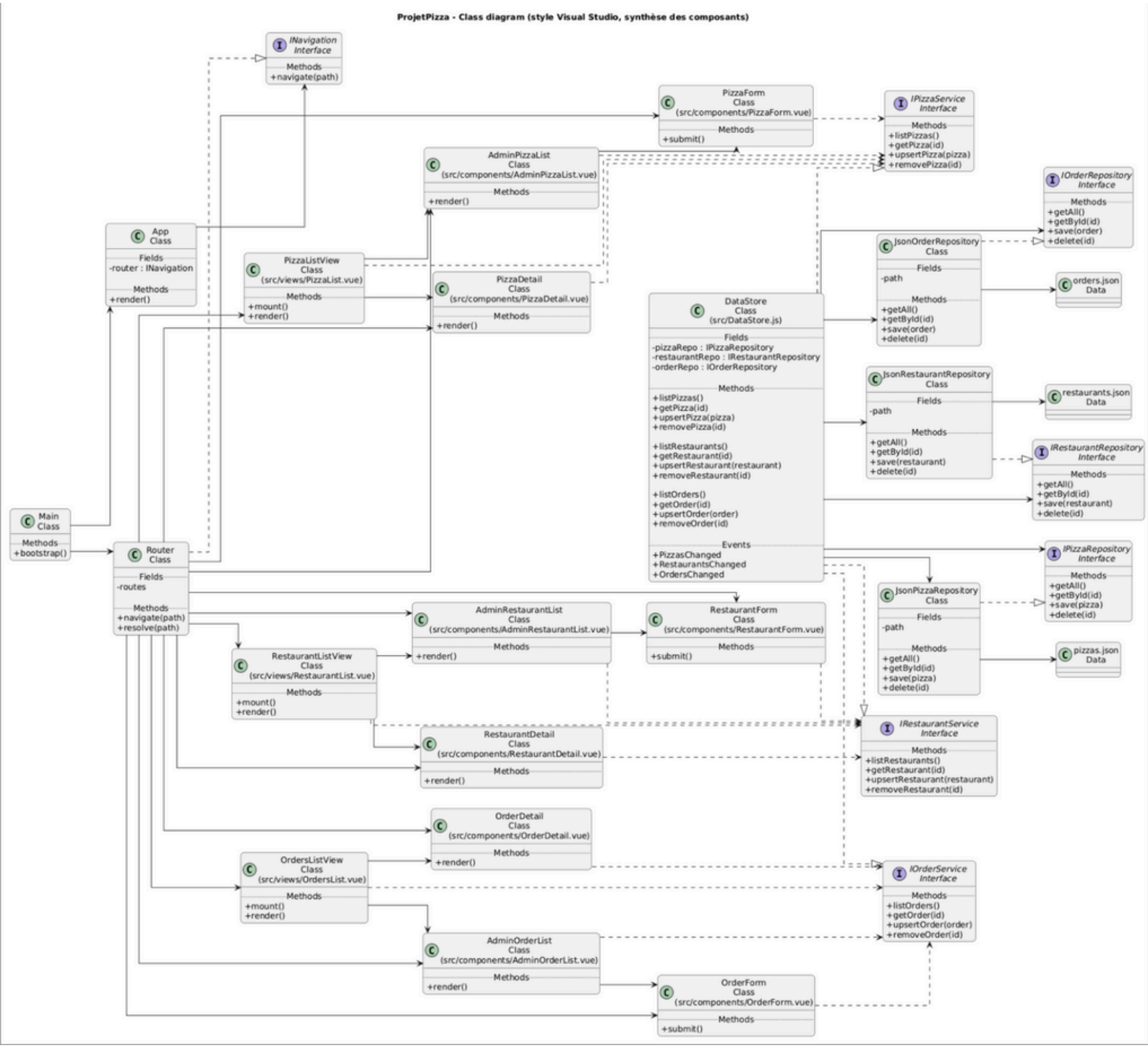


COMPONENT DIAGRAM

BEYA



CLASS DIAGRAM


ProjetPizza - Class diagram (style Visual Studio, synthèse des composants)
INavigation
Interface
Methods
+navigate(path)
PizzaForm
Class
(src/components/PizzaForm.vue)
Methods
+submit()
IPizzaService
Interface
Methods
+listPizzas()
+getPizza(id)
+upsertPizza(pizza)
+removePizza(id)
AdminPizzaList
Class
(src/components/AdminPizzaList.vue)
Methods
+render()
IOrderRepository
Interface
Methods
+getAll()
+getById(id)
+save(order)
+delete(id)
App
Class
Fields
-router : INavigation
Methods
+render()
PizzaListView
Class
(src/views/PizzaList.vue)
Methods
+mount()
+render()
PizzaDetail
Class
(src/components/PizzaDetail.vue)
Methods
+render()
JsonOrderRepository
Class
Fields
-path
Methods
+getAll()
+getById(id)
+save(order)
+delete(id)
orders.json
Data
DataStore
Class
(src/DataStore.js)
Fields
-pizzaRepo : IPizzaRepository
-restaurantRepo : IRestaurantRepository
-orderRepo : IOrderRepository
Methods
+listPizzas()
+getPizza(id)
+upsertPizza(pizza)
+removePizza(id)
+listRestaurants()
+getRestaurant(id)
+upsertRestaurant(restaurant)
+removeRestaurant(id)
+listOrders()
+getOrder(id)
+upsertOrder(order)
+removeOrder(id)
Events
+PizzasChanged
+RestaurantsChanged
+OrdersChanged
JsonRestaurantRepository
Class
Fields
-path
Methods
+getAll()
+getById(id)
+save(restaurant)
+delete(id)
restaurants.json
Data
IRestaurantRepository
Interface
Methods
+getAll()
+getById(id)
+save(restaurant)
+delete(id)
Main
Class
Methods
+bootstrap()
Router
Class
Fields
-routes
Methods
+navigate(path)
+resolve(path)
IPizzaRepository
Interface
Methods
+getAll()
+getById(id)
+save(pizza)
+delete(id)
AdminRestaurantList
Class
(src/components/AdminRestaurantList.vue)
Methods
+render()
RestaurantForm
Class
(src/components/RestaurantForm.vue)
Methods
+submit()
JsonPizzaRepository
Class
Fields
-path
Methods
+getAll()
+getById(id)
+save(pizza)
+delete(id)
pizzas.json
Data
RestaurantListView
Class
(src/views/RestaurantList.vue)
Methods
+mount()
+render()
IRestaurantService
Interface
Methods
+listRestaurants()
+getRestaurant(id)
+upsertRestaurant(restaurant)
+removeRestaurant(id)
RestaurantDetail
Class
(src/components/RestaurantDetail.vue)
Methods
+render()
OrderDetail
Class
(src/components/OrderDetail.vue)
Methods
+render()
OrdersListView
Class
(src/views/OrdersList.vue)
Methods
+mount()
+render()
IOrderService
Interface
Methods
+listOrders()
+getOrder(id)
+upsertOrder(order)
+removeOrder(id)
AdminOrderList
Class
(src/components/AdminOrderList.vue)
Methods
+render()
OrderForm
Class
(src/components/OrderForm.vue)
Methods
+submit()

GANTT DIAGRAM