

# Food Panda

## Project documentation

### Student(s):

➤ Sebastian-Antonio Toma

**Group: 30431-2**

MINISTRY OF EDUCATION



---

**TECHNICAL UNIVERSITY**  
OF CLUJ-NAPOCA, ROMANIA

## I Project specification

The users of this application can be admins (the restaurant administrators) or customers, depending on their role, the main specifications of the project would be: for the customer, they can view restaurants and the menu for the restaurants and order them, also they can see the history of orders and the status of the current placed order; for the admins, they can add restaurants and foods for their restaurant. They can also see the orders, decline/accept them and change their status.

## II Use-Case model

In this section the different types of users will be presented together with the use cases they introduce.

### 2.1 Users and stakeholders

Users of this application can be of two types: the customer and the administrator (of a restaurant).

The administrator can add restaurants and foods for their restaurant. They can also see the orders, decline/accept them and change their status.

The customer/the client can view restaurants and the menu for the restaurants and order them, also they can see the history of orders and the status of the current placed order. They can also register to the app and then log in.

The stakeholders of this application can be the developers and the manager of that application, the restaurants that add their products to this application, which is an online shop. The competition could be other online food delivery applications or other face to face restaurants, that can be negatively impacted by the lack of customers that prefer the commodity of the online shop.

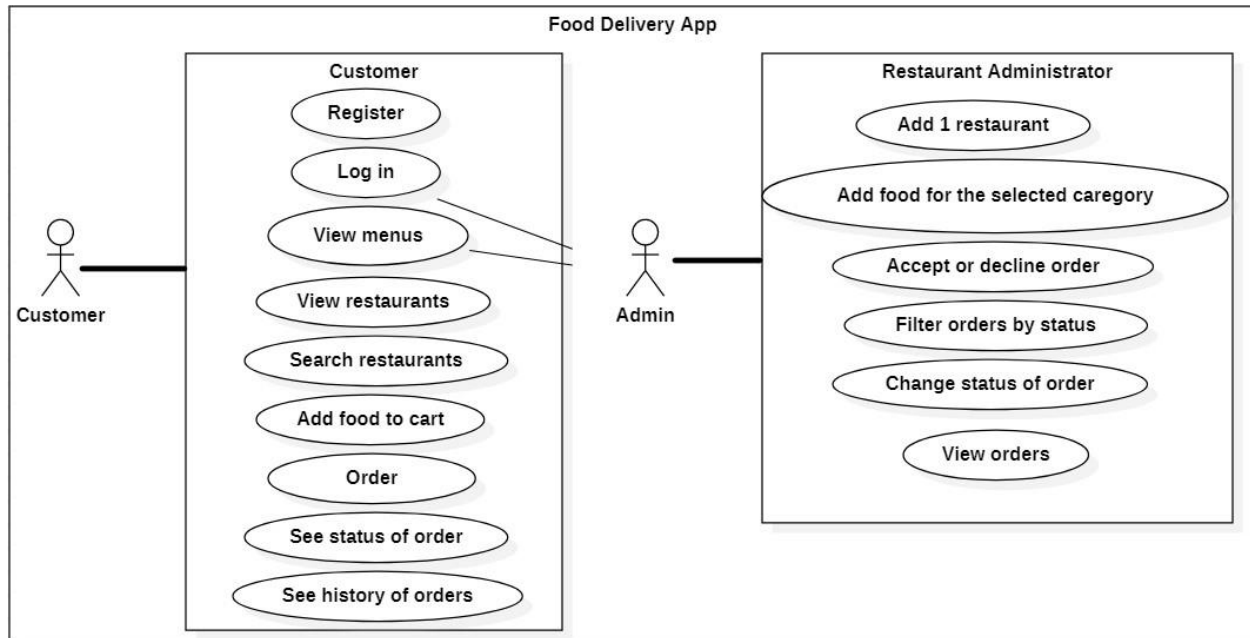
MINISTRY OF EDUCATION



---

**TECHNICAL UNIVERSITY**  
OF CLUJ-NAPOCA, ROMANIA

## 2.2 UML Use-Case diagram



## III Architectural design

I decided to use a Layered Architecture as can be seen from the UML diagrams. A Layered Architecture represents the organization of the project structure into four main categories/layers:

- Presentation layer
- Business layer
- Persistence layer
- Data layer

Each of the layers contains objects related to the particular concern it represents and has a special purpose and calls functions of the layers below it.

MINISTRY OF EDUCATION

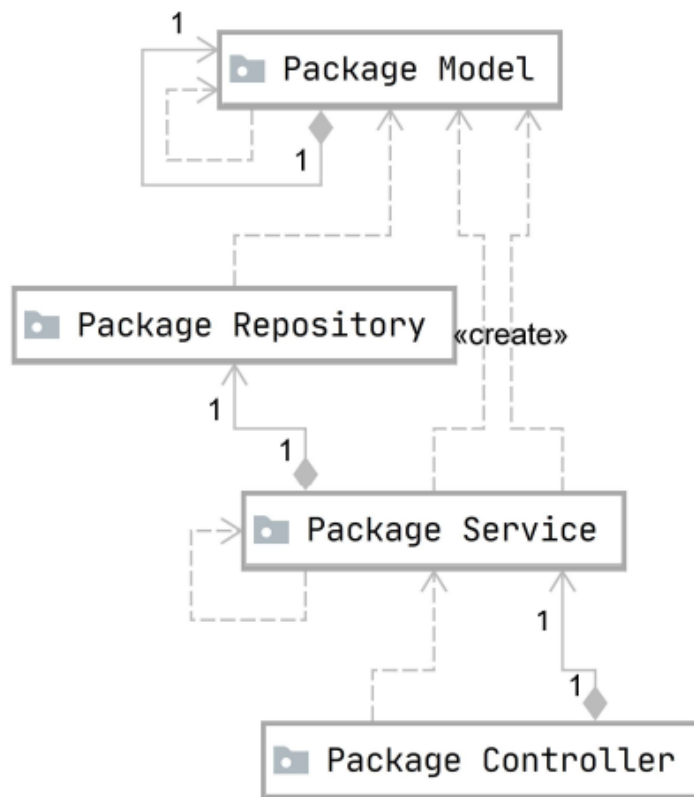


**TECHNICAL UNIVERSITY**  
OF CLUJ-NAPOCA, ROMANIA

### 3.1 Conceptual architecture

The application is a web application (Client-Server) written in Java + Java/Spring/Maven for the backend and React JS for the frontend. It uses a relational database as storage (MySQL). The Data Access Layer is implemented using Hibernate and JPA Repository.

### 3.2 Package diagram



MINISTRY OF EDUCATION

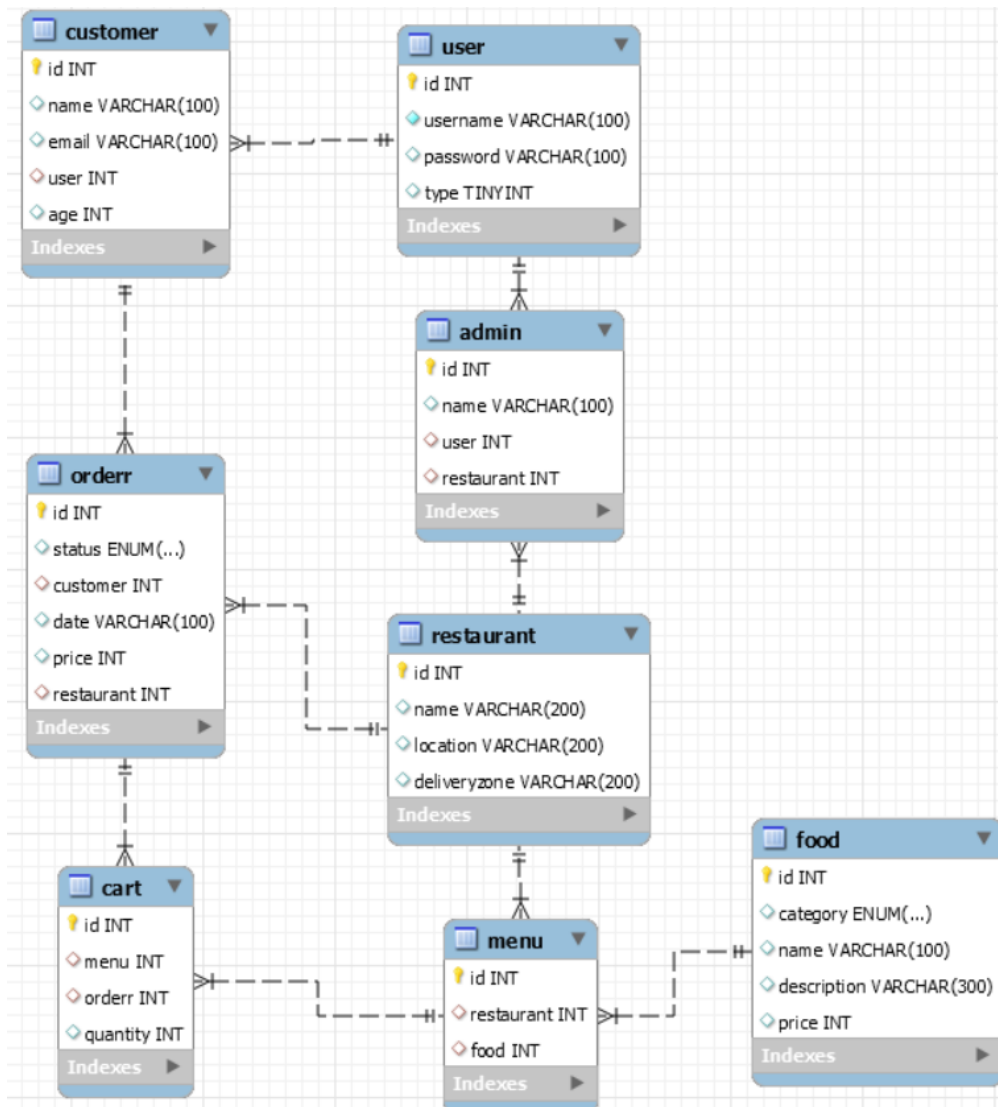


**TECHNICAL UNIVERSITY**  
OF CLUJ-NAPOCA, ROMANIA

### 3.3 Class diagram



### 3.4 Database (E-R/Data model) diagram



MINISTRY OF EDUCATION
























**TECHNICAL UNIVERSITY**  
OF CLUJ-NAPOCA, ROMANIA

## 3.4 Design Patterns

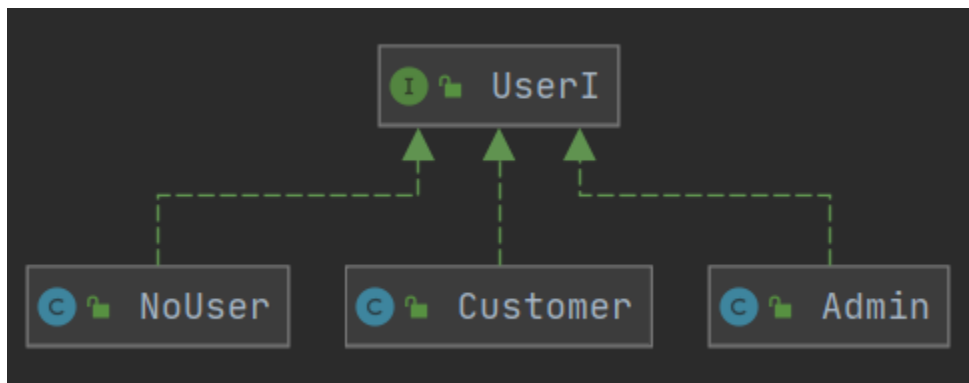
### 3.4.1 Creational Design Pattern

Singleton DP

		ActiveCart	
		CART	ActiveCart
		cart	HashMap<FoodDTO, Integer>
		ActiveCart()	
		getCart()	ActiveCart
		add(FoodDTO)	void
		remove(FoodDTO)	void
		get(FoodDTO)	Integer
		empty()	void
		total	Integer
		restaurant	Integer
		all	List<FoodDTO>

### 3.4.2 Behavioral Design Pattern

Null Object DP



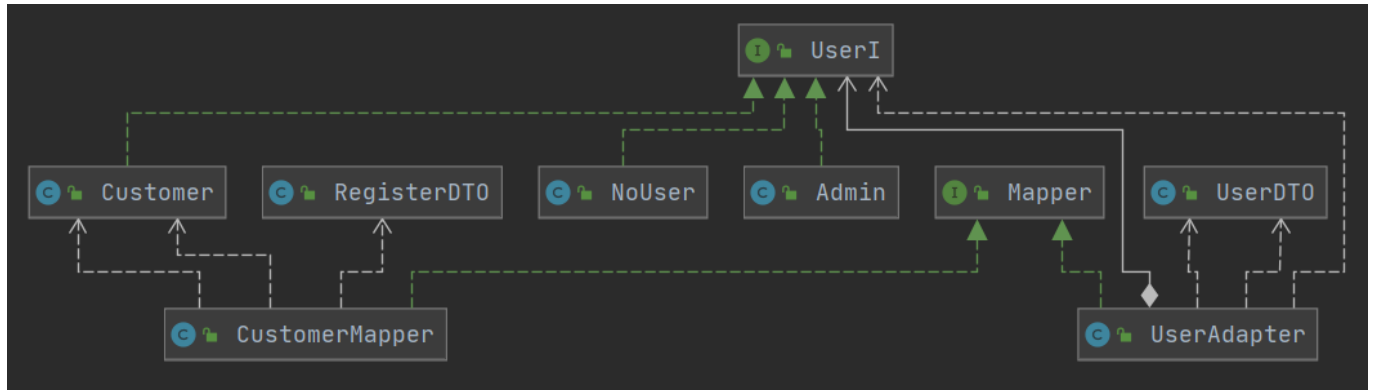
MINISTRY OF EDUCATION



**TECHNICAL UNIVERSITY**  
OF CLUJ-NAPOCA, ROMANIA

### 3.4 Structural Design Pattern

#### Structural DP



### IV Bibliography

- Software Design laboratories

MINISTRY OF EDUCATION



**TECHNICAL UNIVERSITY**  
OF CLUJ-NAPOCA, ROMANIA