TECHNICAL UNIVERSITY
OF CLUJ-NAPOCA
ROMANIA

Faculty of Automation
and Computer Science

Food Delivery Application

-Project prepared for the Software Design course-

Student: Borbála Fazakas
Laboratory Assistant: Teodora Vezan
Group 30432
2021-2022

Table of Contents

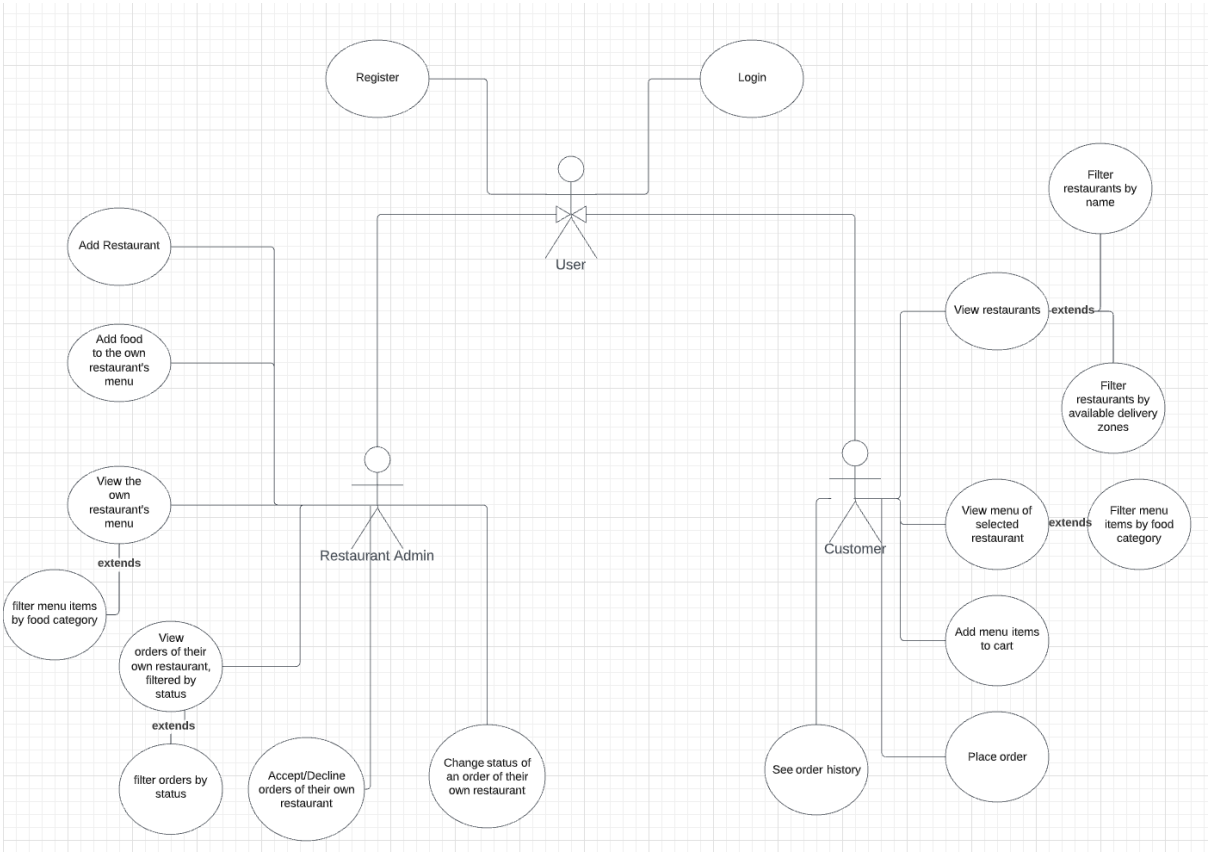
Introduction

Objectives

The primary goal of this project is to implement a desktop application that allows

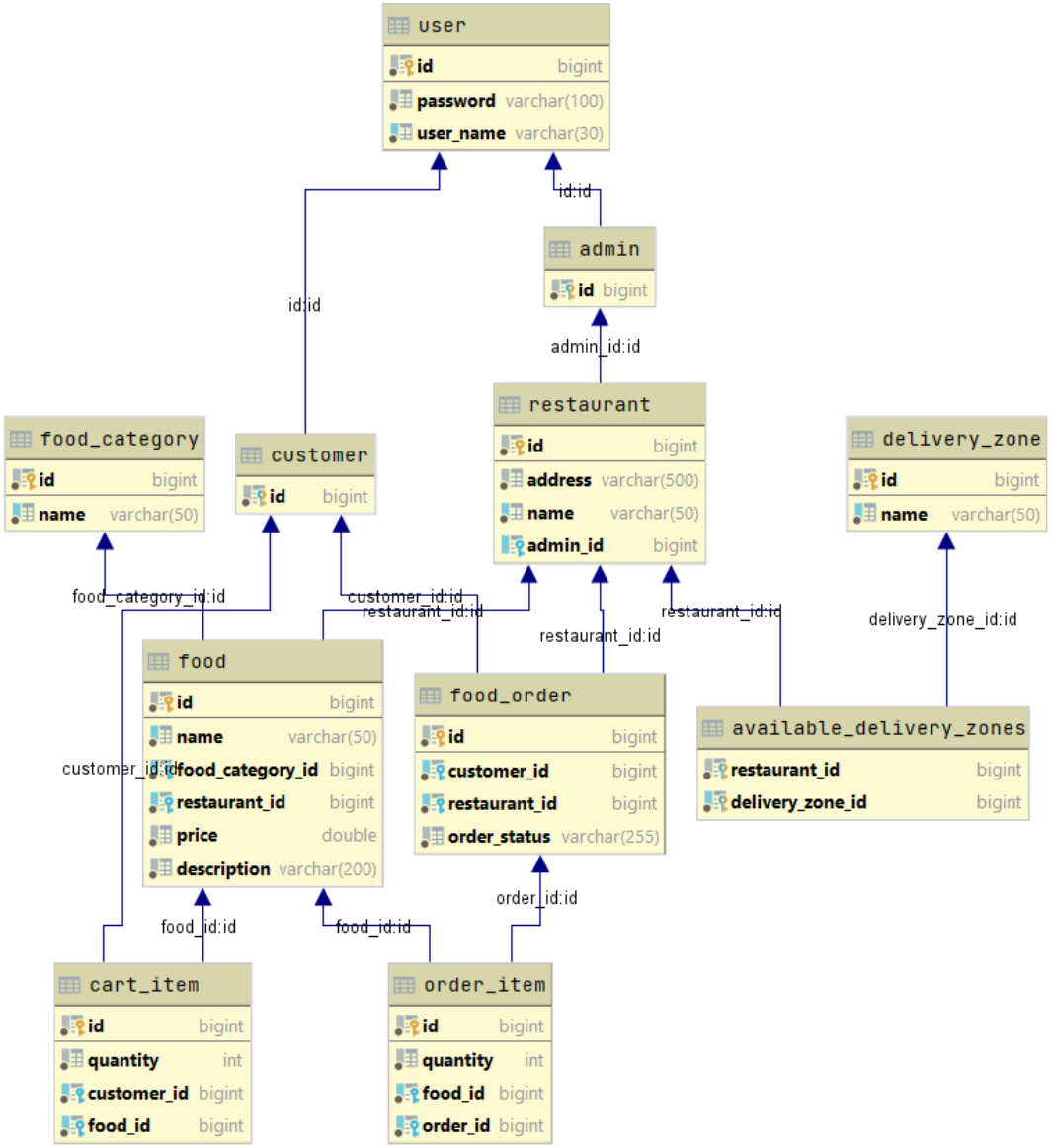
- a restaurant admin
    - to set up a restaurant, with name, address and available delivery zones
    - add menu items to the menu of their restaurant, with name, price, description and category
    - manage the incoming orders, by changing their status (pending/accepted/declined/in delivery/delivered)
- the Customer to
    - search for restaurants
    - view the menu of a selected restaurant
    - put items from one single restaurant into their cart
    - place an order
    - view the status of their orders

Use Cases


Register
Login
User
Add Restaurant
Add food to the own restaurant's menu
View the own restaurant's menu
extends
filter menu items by food category
View orders of their own restaurant, filtered by status
extends
filter orders by status
Restaurant Admin
Accept/Decline orders of their own restaurant
Change status of an order of their own restaurant
Filter restaurants by name
View restaurants
extends
Filter restaurants by available delivery zones
View menu of selected restaurant
extends
Filter menu items by food category
Add menu items to cart
Place order
See order history
Customer

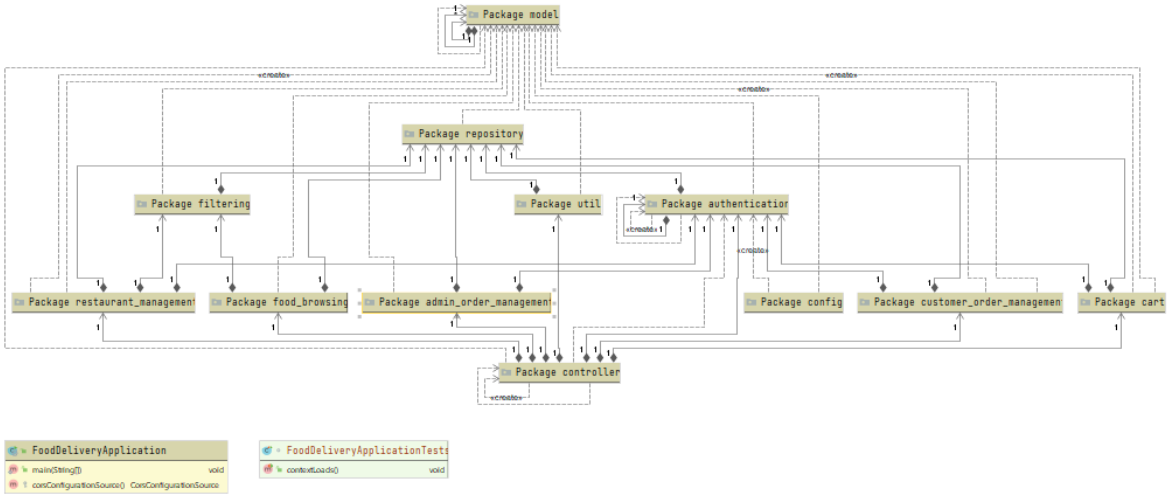
Implementation

Database Design

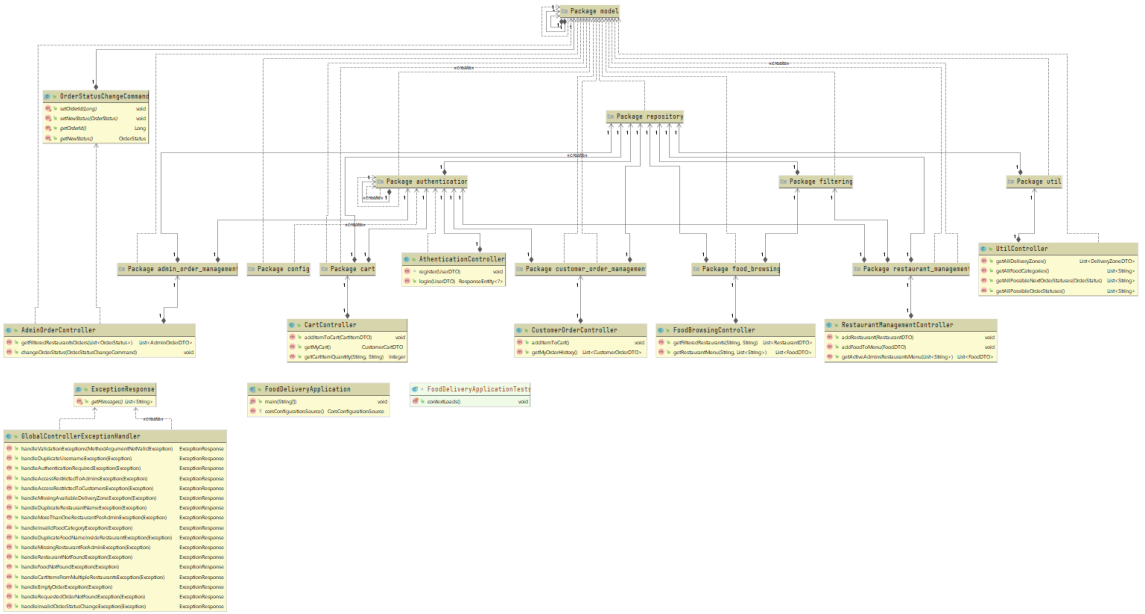


Software Design

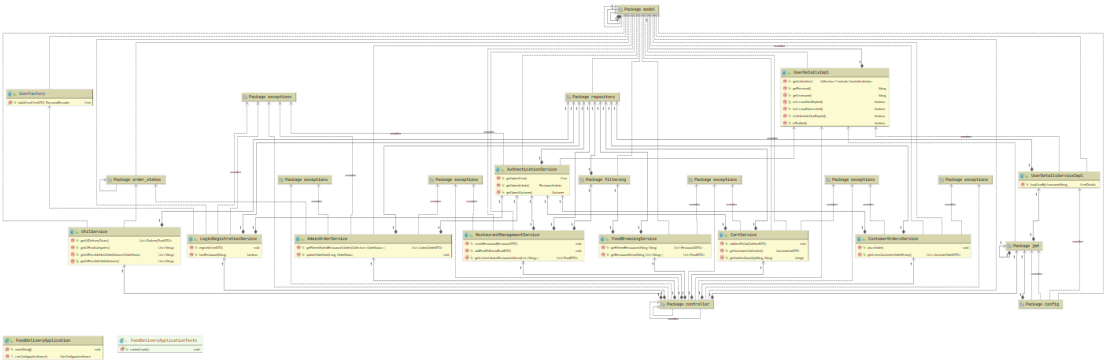
Layered Architecture

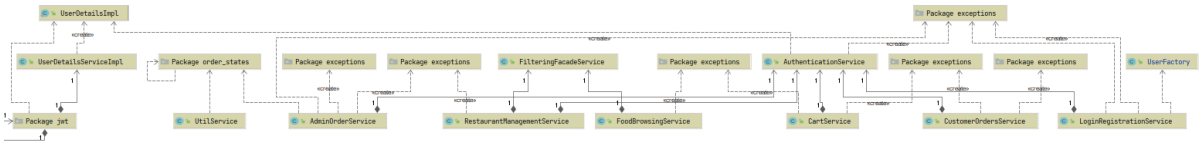


Controller

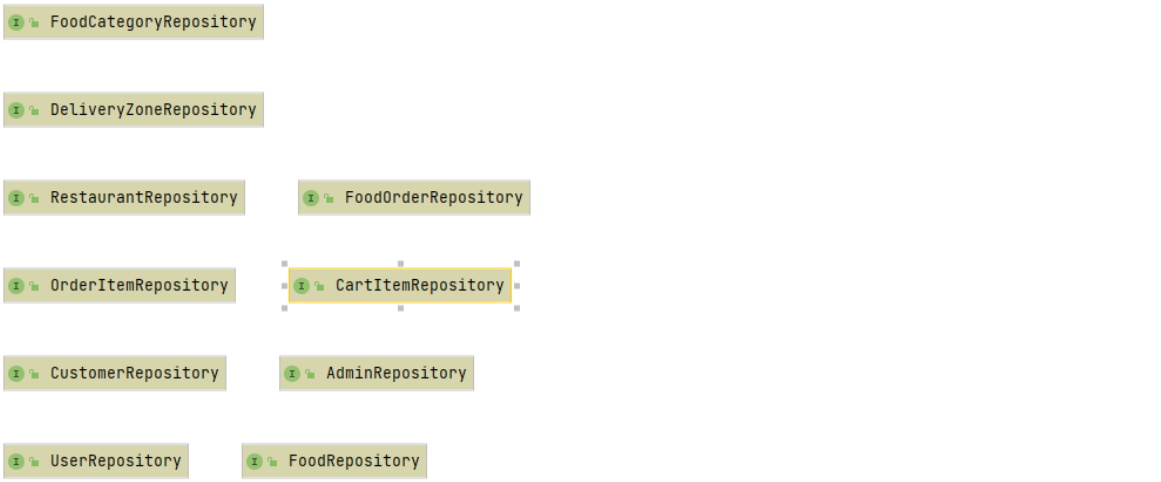


Service

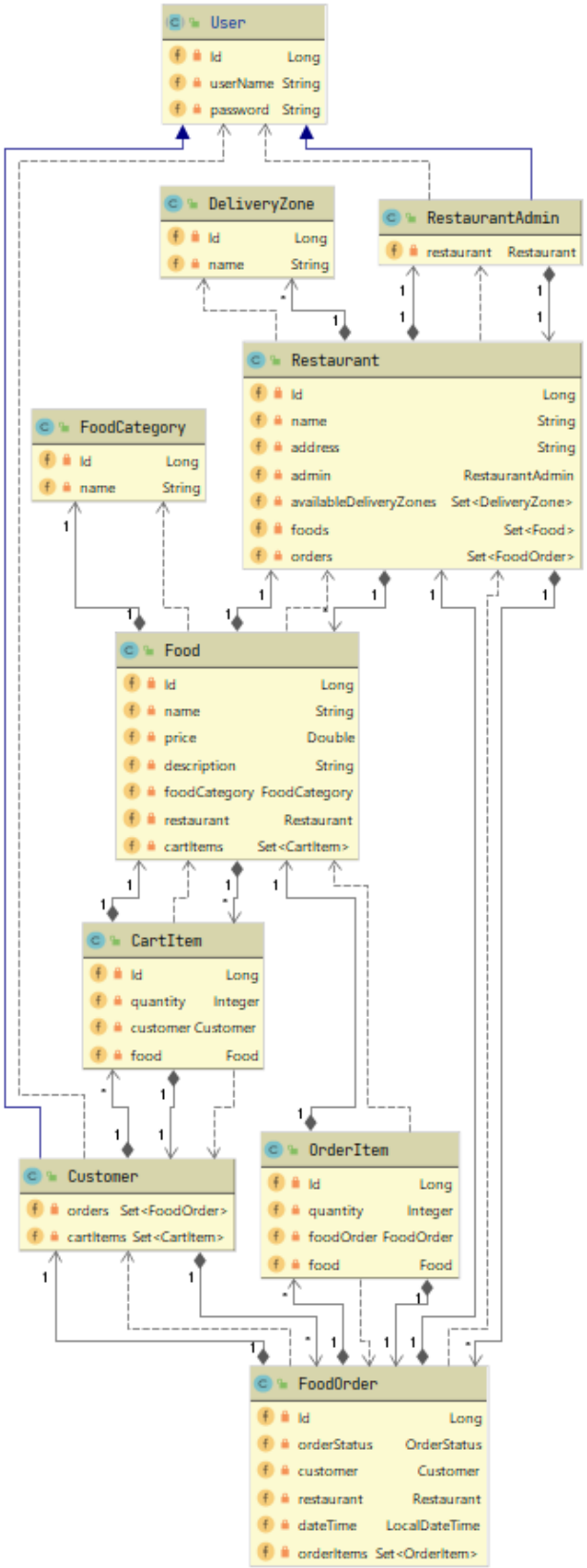




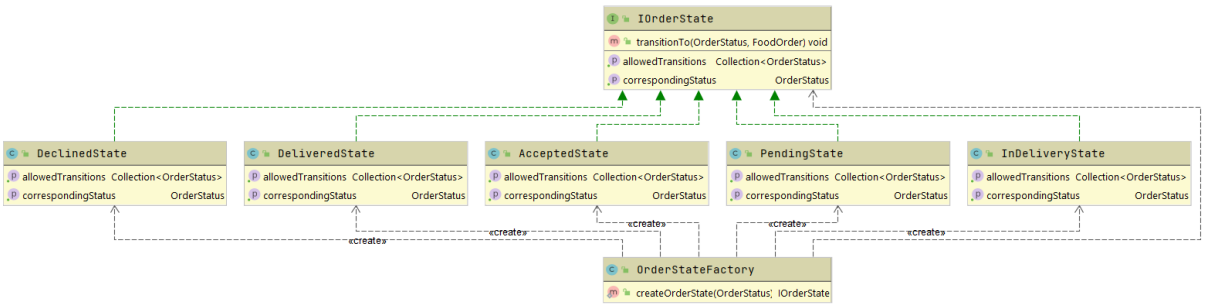
Repository



Model

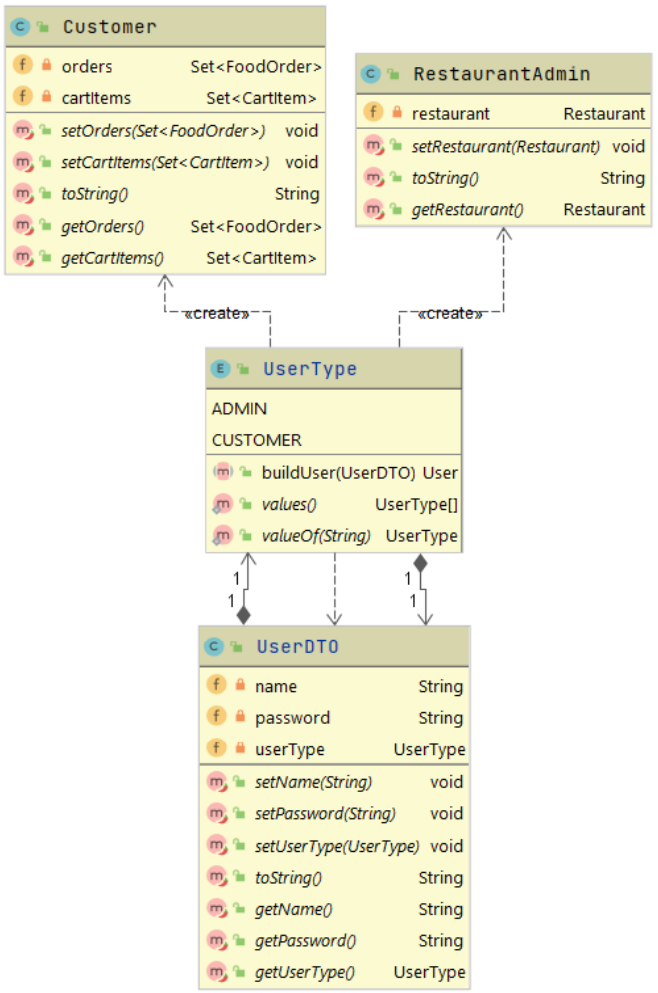

User
Id Long
userName String
password String
DeliveryZone
Id Long
name String
RestaurantAdmin
restaurant Restaurant
Restaurant
Id Long
name String
address String
admin RestaurantAdmin
availableDeliveryZones Set<DeliveryZone>
foods Set<Food>
orders Set<FoodOrder>
FoodCategory
Id Long
name String
Food
Id Long
name String
price Double
description String
foodCategory FoodCategory
restaurant Restaurant
cartItems Set<CartItem>
CartItem
Id Long
quantity Integer
customer Customer
food Food
Customer
orders Set<FoodOrder>
cartItems Set<CartItem>
OrderItem
Id Long
quantity Integer
foodOrder FoodOrder
food Food
FoodOrder
Id Long
orderStatus OrderStatus
customer Customer
restaurant Restaurant
dateTime LocalDateTime
orderItems Set<OrderItem>

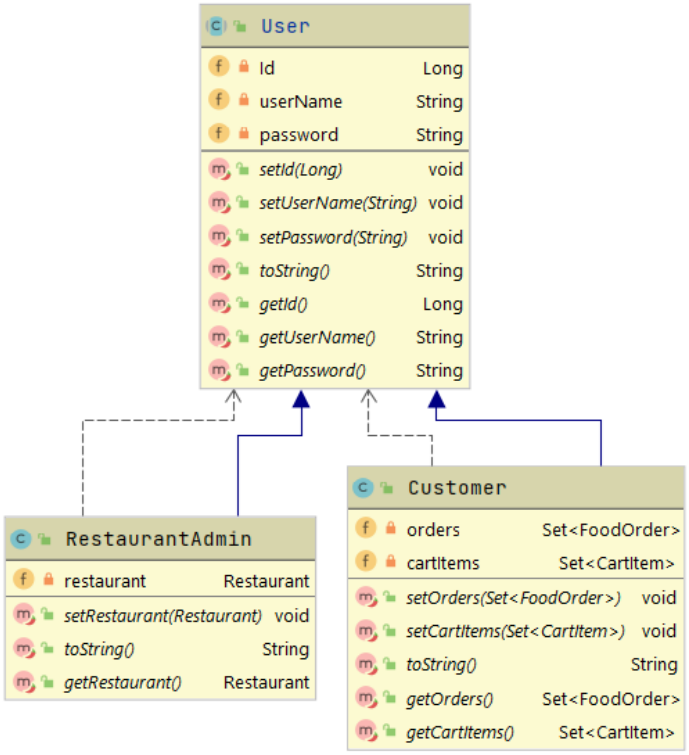
Behavioral Design Pattern: State Pattern + Creational Design Pattern: Factory Method



Usage: Simplifies managing the transitions between different states.

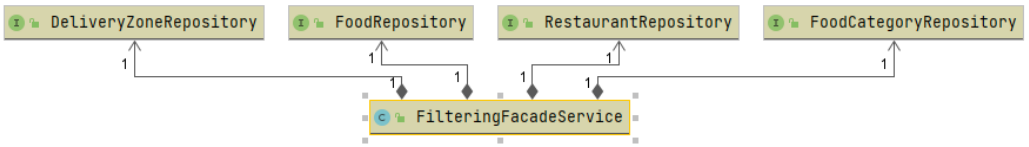
Creational Design Pattern: Factory Method





The client communicates with the server via the UserDTO objects. When a user signs up, a UserDTO is sent with their data, and the UserType enum. The UserType instances behave as the creators of the corresponding Users, through the ‘buildUser()’ method: they know which type of User must be instantiated and how.

Structural Design Pattern: Facade Pattern



Usage: The FilteringFacade provides a simplified interface for the repositories, in case multiple filtering criteria may be built on top of each other.