# **FOOD ORDERING SYSTEM**

# **(food-lee)**

# CAPGEMINI TRAINING PROGRAMME

# JEE CLOUD

# PREPARED BY

TUSHAR KUMAR PANDEY

# Personal Project

## **PROBLEM STATEMENT**

Develop a commercial Food Ordering System which allows the users to view and order food from restaurants.

**ABSTRACT**

This project is aimed at developing a Food Ordering System. This is a web based application that can be accessed throughout the web. This application consists of two actors. An admin and a user where admin can add restaurant by logging in and add food items offered by every restaurant while the user can view the restaurants and the food items and add items to cart for placing order.

**SCOPE**

**In Scope**

* Actor: 2 types of users exist:
* Admin: login, add/update restaurant, view orders, add new food item to restaurant
* Customer: login, place order, view restaurants, view food items in a restaurant
* Validation:

**Out Of Scope**

* Actor: 4 types of users : Including Restaurant Owner, And Delivery Guy
* Many to Many relationship between food and order not implemented

**CLASSES**

**Class: User**

**Attributes:**

* private Integer userId
* private String username
* private String password

**Methods:**

* **addRestaurant(Restaurant): Restaurant**

This method will prompt admin to enter all the necessary details required to register the new restaurant in the food ordering system.

* **addfood(Food): food**

This method will prompt admin to enter all the necessary details required to add new type of food to the restaurant.

* **viewRestaurants(Restaurant): List<Restaurant>**

It will display all restaurants

* **viewOrders():List<Order>**

It will ask display all orders.

**Class: Customer**

**Attributes**

* private customerId: Integer
* private customerName:String
* private phoneNumber: String
* private email: String

**Class: Restaurant**

**Attributes**

* restaurantId: Integer
* restaurantName: String
* restaurantAddress: String

**Class: Food**

**Attributes**

* foodId: Integer
* foodName: String
* category: String
* available:Boolean
* vegetarian:Boolean
* restaurant:restaurant

**Class: FoodOrder**

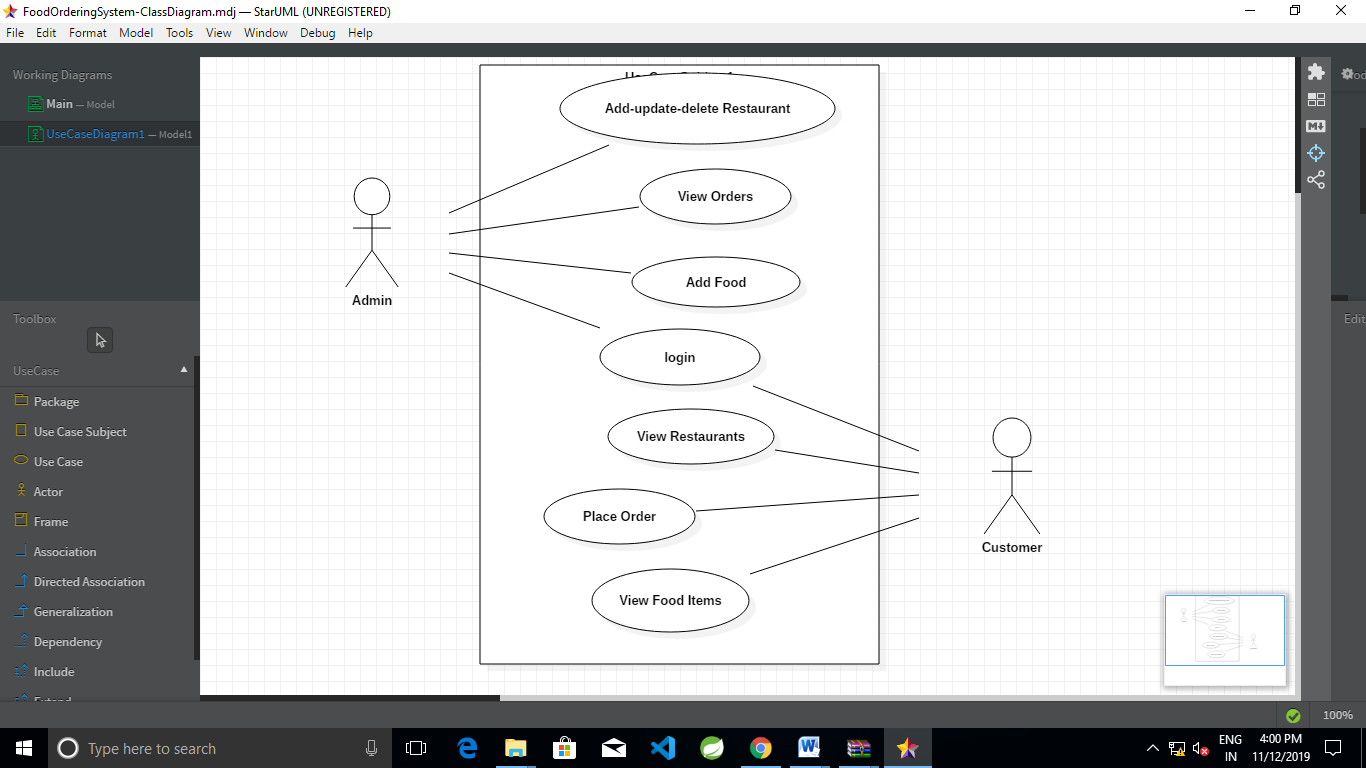
**Attributes**

* orderId: Integer
* orderStatus: String
* orderDate: String
* orderTime:String
* foods:Long
* restaurant:restaurant
* address:String

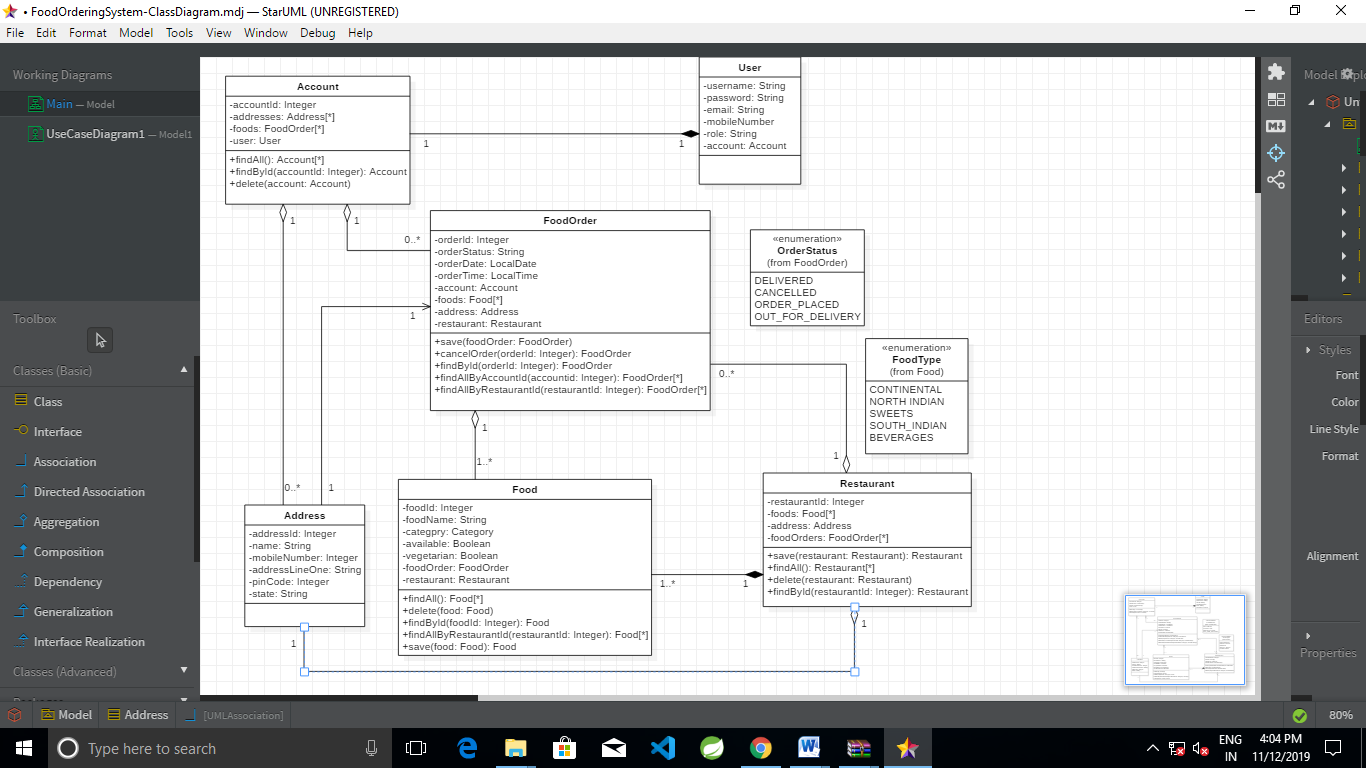
**VALIDATION RULES**

* Every restaurant must have single food item.
* Every order date should be in future.
* As soon as the application is accepted a new customer is added to the system.

## **USE CASE DIAGRAM**



## **CLASS DIAGRAM**



­­

## **SEQUENCE DIAGRAM**

