**DOKUZ EYLÜL UNIVERSITY**

**ENGINEERING FACULTY**

**DEPARTMENT OF COMPUTER ENGINEERING**

**CME 2210**

**Object Oriented Analysis and Design**

**ONLINE FOOD SERVICE SYSTEM**

**by**

**Melisa Beysümengü**

**İrem Çalmaz**

**İrem Okur**

**Contents**

[**CHAPTER ONE** 3](#_Toc34675128)

[**INTRODUCTION** 3](#_Toc34675129)

[**1.1.** **Purpose** 3](#_Toc34675130)

[**1.2.** **Scope** 3](#_Toc34675131)

[**CHAPTER TWO** 4](#_Toc34675132)

[**REQUIREMENTS** 4](#_Toc34675133)

[**2.1. Object Oriented Model** 4](#_Toc34675134)

[**2.1.1 Address** 4](#_Toc34675135)

[**2.1.2 Phone** 4](#_Toc34675136)

[**2.1.3 Management** 4](#_Toc34675137)

[**2.1.4 User** 4](#_Toc34675138)

[**2.1.5 Customer** 5](#_Toc34675139)

[**2.1.6 Admin** 5](#_Toc34675140)

[**2.1.7 Restaurant** 5](#_Toc34675141)

[**2.1.8 Food** 6](#_Toc34675142)

[**2.2. Dependencies** 6](#_Toc34675143)

# **CHAPTER ONE**

# **INTRODUCTION**

## **Purpose**

Online food service system is created to help customers who wants to order food and the system keeps the name of the restaurants which has a package delivery option. In the past years, when a customer wanted to order a food, they would go to the restaurant or they would call the restaurant directly but this online food service system saves the customers’s time and assists them to choose the best food option according to their budget.

## **Scope**

The System is going to provide a safe, fast and easy online ordering system for customers. Every costumer has their own profiles and they can access to the system with given password. Costumers can choose a restaurant according to their budget and also they can see if a restaurant is full or not, if it is not full then they can see the number of customers in front of them. Payments will be done at the doorstep.

Furthermore, the system contains admin section. This section allows restaurant owners to add new food or remove an existing food on the current menu and admins can see the total orders. Moreover, admin can shut down the system to service because of the unexpected circumstances for instance occurness of over ordering.

# **CHAPTER TWO**

# **REQUIREMENTS**

## **2.1. Object Oriented Model**

The system has 8 classes. Such as Restaurant, Food, Customer, User, Address, Phone, Admin and Management.

### **2.1.1 Address**

Address class has 4 attributes. For instance streetname, town, city, description. These all are String. And this class includes their setters and getters.

### **2.1.2 Phone**

Phone class has 2 attributes. For instance country\_code, number. These all are String. And this class includes their setters and getters.

### **2.1.3 Management**

Management class created for managing the entire system. Functions will be added according to system’s needs.

### **2.1.4 User**

User class has 6 attributes. This class has: two String attribute called name, surname; an address attribute defined in Address class; an phone attribute defined in Phone class; a String attribute called password; customerlist and adminlist attribute as an ArrayList type.

This class contains above attributes’s getters and setters ; toString() method which returns String.

### **2.1.5 Customer**

Customer class has 3 attributes. Such as two integer attribute called ID\_customer, count; an orders attribute as an ArrayList type.

This class has above attributes’s getters, setters and toString() method; DisplayAllOrders method which display all orders; CreateCustomer method which takes User class’s variables and add the new customer to the customerlist ArrayList;Order method takes 2 parameters from Restaurant and Food classes.

### **2.1.6 Admin**

Admin class has 2 attributes. Such as an integer attribute called ID\_admin, count .

This class has above attributes’s getters, setters and toString() method ; CreateAdmin method which takes User class’s variables and add the new admin to the adminlist ArrayList.

### **2.1.7 Restaurant**

Restaurant class has 7 attributes. This class has: two String attribute called restaurant\_name, ship\_min; an address attribute defined in Address class; an phone attribute defined in Phone class; and food attribute defined in Food class which type as a ArrayList; a customerqueue attribute using Queue structure.

This class contains above attributes’s getters and setters, toString(); Displayqueue method which shows all customers that ordered food from certain restaurant; addFood method which takes a parameter from Food class and add them to food ArrayList; RemoveFood method which removes it from food ArrayList.

### **2.1.8 Food**

Food class has 4 attributes. This class has: an String attribute called food\_name; an restaurant\_list attribute defined in Restaurant class which type as a ArrayList and lastly String attribute called ingredients using ArrayList structure.

This class contains above attributes’s getters and setters, toString(); addRestaurant method takes a parameter from Restaurant class and add them to restaurant\_list ArrayList.

## **2.2. Dependencies**

Admin and Customer Classes extends User Class. Also the User class implements Userinterface interface.

New functionalities can be added or remaining functionalities can be changed during the development process.