# Table of Contents

Chapter 3

3. System Design

### 3.1 Sequence Diagrams

### 3.2 Class Diagram

### 3.3 Entity Relationship Diagram

Chapter 4

4. Implementation

### 4.1 Tools &Technologies

### 4.2 Project Link

# Chapter 1

1. **Introduction:** Online food ordering is a process of ordering food from a local restaurant or food cooperative through a web page or app. Much like ordering consumer goods online, many of these allow customers to keep accounts with them in order to make frequent ordering convenient. A customer will search for a favorite restaurant, usually filtered via type of cuisine and choose from available items, and choose delivery or pick-up. Payment can be amongst others either by cash, with the restaurant returning a percentage to the online food company.

    **1.1**    Online Food Ordering System

    **1.2**    **Purpose:** In today's era people become so busy with their work for the whole day. So it's become very difficult to manage some time to go out for some hangout & eat something of their own choices. So this project is for all those people who can easily order different types of foods of their own choices in online. This will provide a user friendly environment between the customer and employee thus increasing the efficiency of the food ordering system.

    **1.3 Scope:** Online Purchase, Type of Food Provided, The Foods & Services Offered There.

    **1.4 Why the system is necessary**:
        1. Higher Employee Productivity
        2. Order Accuracy
        3. Grow the Business
        4. Ability to Build a Database

# Chapter 2

2. System Analysis:

2.1 Use Case Model
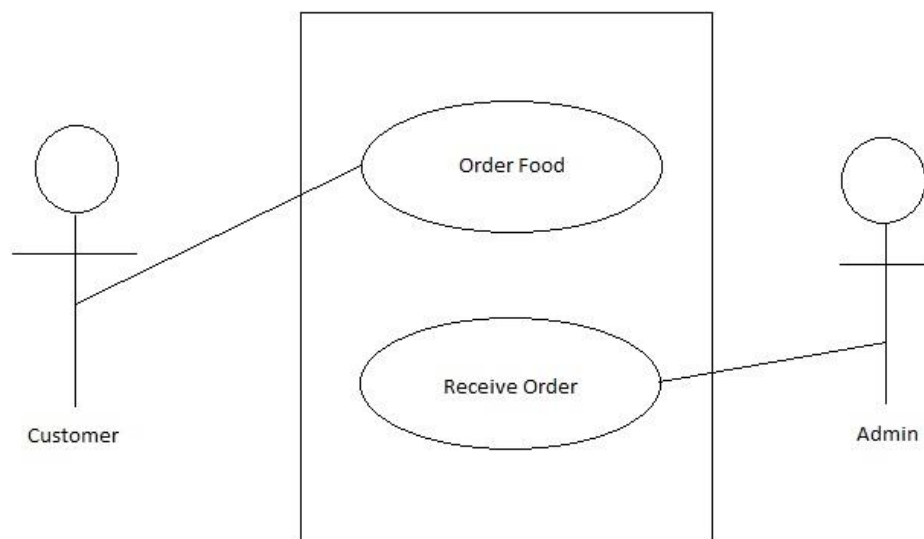


2.2 Actor Goal List

1. Customer can see the food list.
2. Customer can order the food.
3. Admin can see the order.
4. Customer can place the order.

2.3 Use case description

## 2.3.1 Order Food

    1.  After entering into the site, customer can see the foods

    2. Customer can place the order.
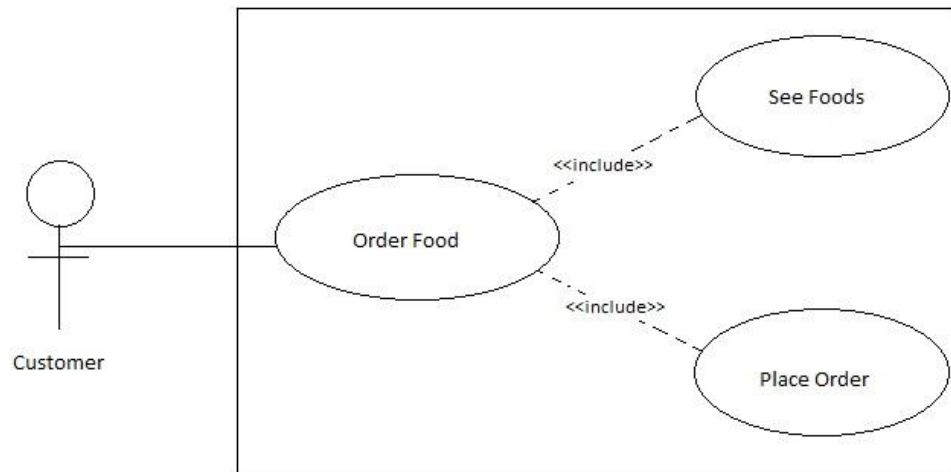


## 2.3.2 Receive Order

    1. Admin can confirm the order.

    2. Admin can cancel the order for unwanted situation.
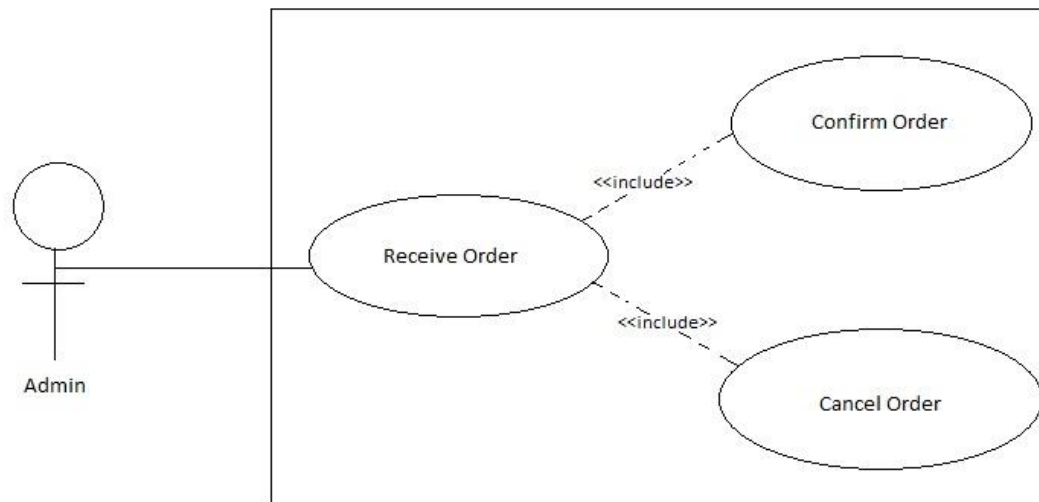
## 2.4 Use Case Description (Detailed)

### 2.4.1 Order Food

| Use Case Id | 1 | |
|---|---|---|
| Use Case Name | Order Food | |
| Primary Actor | Customer | |
| Goal | To order food | |
| Pre-Condition | Need to have selected foods | |
| Post Condition | Need to place order by seeing the desire ordered foods | |
| Main Success | Actor | System |
| Scenario | 1) Customer order the desired food. <br><br> 2) Submit to server. | 1.1) System gives confirmation message. <br><br> 2.1) Server generate confirm message. |
| Failure Scenario | 1) Customer don't able to select foods. | |

| | 2) Customer can't place the order. |
|---|---|

### 2.4.2 Receive Order

| Use Case Id | 2 | |
|---|---|---|
| Use Case Name | Receive Order | |
| Primary Actor | Admin | |
| Goal | Make ensure taking the order | |
| Pre-Condition | Need to have a code number of each orders. | |
| Post Condition | Need to check the food whether it's available or not | |
| Main Success Scenario | Actor | System |
| | 1) Admin takes the order from the customer.<br><br>2) Admin confirms the order | 1.1) System gives confirmation message. |
| Failure Scenario | 1) Don't able to confirm the order | |

## 2.5 System Sequence Diagram

## 2.5.1 Order Food (success scenario)

```
┌──────────┐                              ┌──────────┐
│ Customer │                              │  System  │
└──────────┘                              └──────────┘
     │                                          │
     │         addCart(name,price)              │
     │─────────────────────────────────────────>│
     │                                          │
     │              add done                    │
     │<- - - - - - - - - - - - - - - - - - - - -│
     │                                          │
     │      viewOrderedFoods(name,total)        │
     │─────────────────────────────────────────>│
     │                                          │
     │              view done                   │
     │<- - - - - - - - - - - - - - - - - - - - -│
     │                                          │
     │              confirmOrder                │
     │─────────────────────────────────────────>│
     │                                          │
     │               success                    │
     │<- - - - - - - - - - - - - - - - - - - - -│
     │                                          │
```
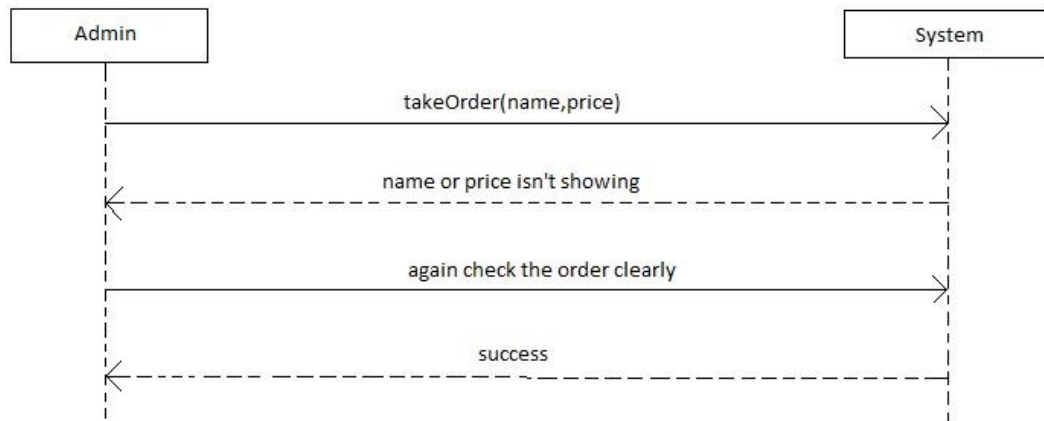
## 2.5.2 Order Food (Failure Scenario)

```
┌──────────┐                              ┌──────────┐
│ Customer │                              │  System  │
└──────────┘                              └──────────┘
     │                                          │
     │         addCart(name,price)              │
     │─────────────────────────────────────────>│
     │                                          │
     │       price or name isn't showing        │
     │<- - - - - - - - - - - - - - - - - - - - -│
     │                                          │
     │           Again addCart clearly          │
     │─────────────────────────────────────────>│
     │                                          │
     │                 Done                     │
     │<- - - - - - - - - - - - - - - - - - - - -│
     │                                          │
```
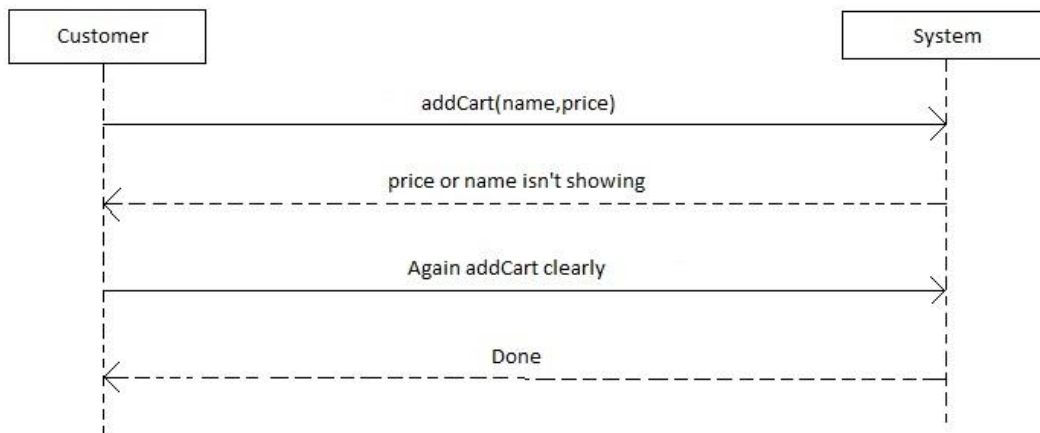
## 2.5.3 Receive Order (success scenario)

| Admin | | System |
|-------|---|--------|
| | takeOrder(name,price) → | |
| | ← take done | |
| | confirmOrder(name,total → | |
| | ← success | |

## 2.5.4 Receive Order (failure scenario)

| Admin | | System |
|-------|---|--------|
| | takeOrder(name,price) → | |
| | ← name or price isn't showing | |
| | again check the order clearly → | |
| | ← success | |

## 2.6 Domain Model

| Customer | | Order Food |
|---|---|---|
| -Name: String<br>-Cell No: String<br>+Address: String | 0..*  give  * | -Food Name: String<br>-Price: String |
| +takeInfo() | | +addCart()<br>+placeOrder() |

| Admin | | Receive Order |
|---|---|---|
| -Name: String<br>-Cell No: String<br>+Address: String<br>-E_mail: String | 0..*  take  * | -Food Name: String<br>-Price: String |
| +givenInfo() | | +chcekOrder()<br>+confirmOrder() |

## 2.7 Activity Diagram

customer add cart food

cancel process

place order

admin take order

confirm order
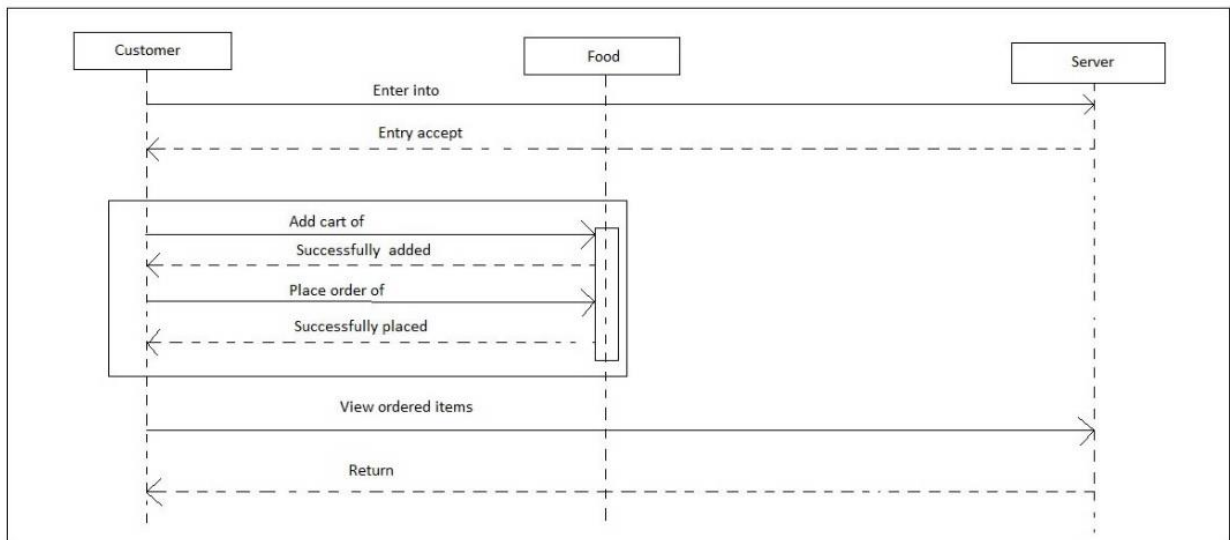
# Chapter 3:

3. System Design:

    3.1   Sequence Diagram

        3.1.1  Order Food



        3.1.2  Receive Order

## 3.2 Class Diagram

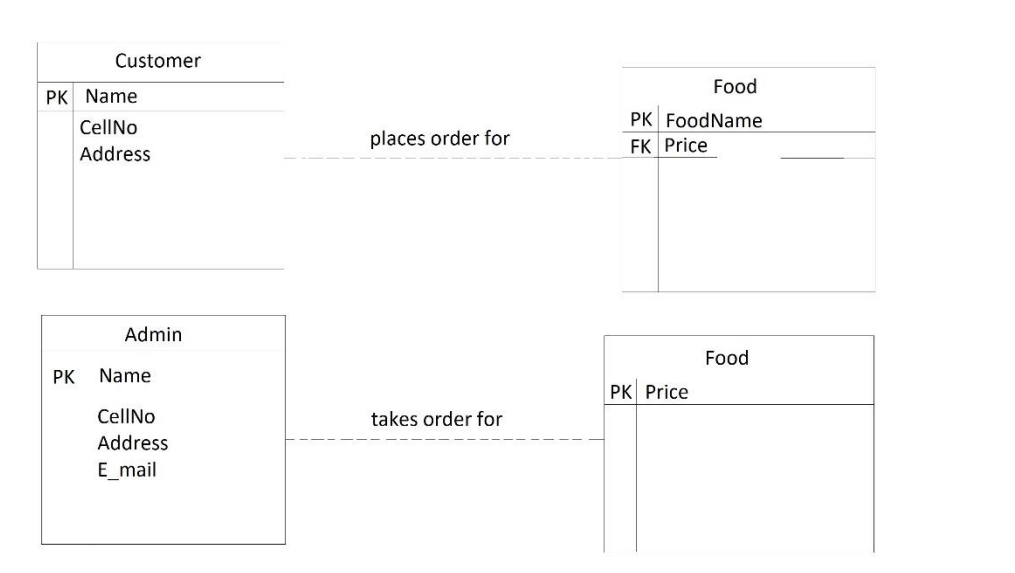| Admin |
| --- |
| -Name: String |
| -Cell No: String |
| +Address: String |
| -E_mail: String |
| |

| Customer |
| --- |
| -Name: String |
| -Cell No: String |
| +Address: String |
| |

| Receive Order |
| --- |
| -Name: String |
| -Price: String |
| |
| +checkOrder() |
| +confirmOrder() |

| Order Food |
| --- |
| -Name: String |
| -Price: String |
| |
| +addCart() |
| +placeOrder() |

## 3.3 Entity Relationship Diagram

| Customer | |
|---|---|
| PK | Name |
| | CellNo |
| | Address |

places order for

| Food | |
|---|---|
| PK | FoodName |
| FK | Price |

| Admin | |
|---|---|
| PK | Name |
| | CellNo |
| | Address |
| | E_mail |

takes order for

| Food | |
|---|---|
| PK | Price |
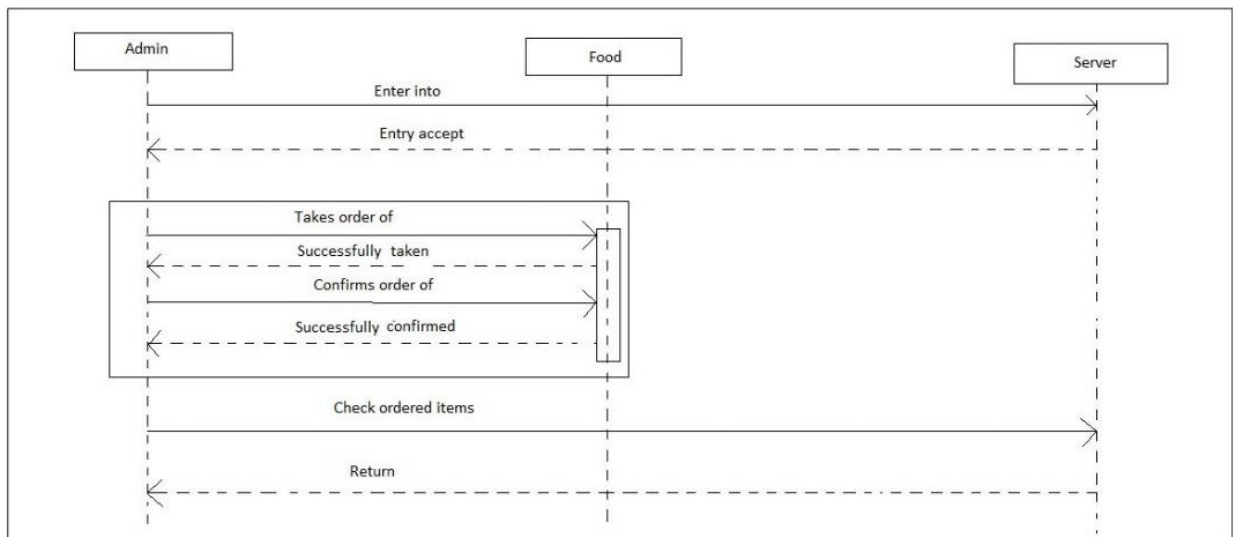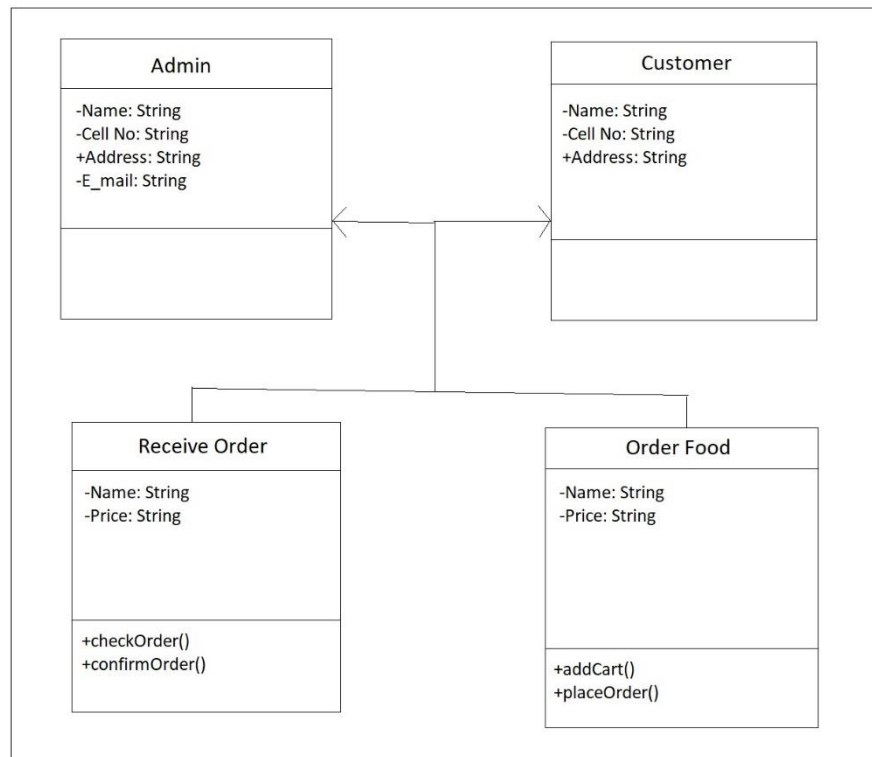
# Chapter 4:

3. **Implementation:** Implementation perspective describes software implementations in a particular technology such as PHP. The implementation phase deals with issues of quality, performance and debugging. The end deliverable of implementation phase is the product itself.

## 3.1. Tools & Technologies

The following tools are using for developing the project

PHP

Sublime Text 3

HTML5, CSS, JavaScript, bootstrap template

## 3.2. Project Link

They provide github repository link after uploading the project there.