

**Design Document**

Restaurant reservation and food shopping online

(Tunga Restaurant)

**Index**

[1. Introduction](#_heading=h.gjdgxs) 3

[1.1 Our goal](#_heading=h.30j0zll) 3

[1.2 Our solution](#_heading=h.1fob9te) 3

[2. Customer requirement](#_heading=h.3znysh7) 3

[2.1 User Interface](#_heading=h.2et92p0) 3

[2.2 Manager](#_heading=h.tyjcwt) [Interface](#_heading=h.2et92p0)5

[3. How we design of our system](#_heading=h.3dy6vkm) 6

[4. Functional Decomposition Diagram](#_heading=h.1t3h5sf) 7

[4.1](#_heading=h.4d34og8) [Functional Decomposition Diagram](#_heading=h.1t3h5sf) - Manager7

[4.2](#_heading=h.2s8eyo1) [Functional Decomposition Diagram](#_heading=h.1t3h5sf) - User7

[5. Data Flow Diagram](#_heading=h.17dp8vu) 8

[5.1 Context Diagram](#_heading=h.3rdcrjn) 8

[5.2 Detail Level Diagram](#_heading=h.26in1rg) 8

[6. Entity Relationship Diagram](#_heading=h.lnxbz9) 11

[6.1](#_heading=h.35nkun2) [Entity Relationship Diagram](#_heading=h.lnxbz9)11

[6.2 Entity - Attribute](#_heading=h.1ksv4uv) 12

7. Data design, Data structure 13

The End 22

**About Our Product**

# 1. Introduction

## 1.1 Our goal

* Customer: Help customers buy food online, also help them make table reservations easier with only a small phone that has access to the internet. Easier to use, easier to served
* Restaurant: Make the selling and management process become far easier and cheaper than before.

## 1.2 Our solution

* For customers: Provide an online environment for them to freely shop and make reservations at will. Smart design UI makes customers easy to shop and so on....
* For restaurant: Provide the method that helps you to manage your product, room ,table and customer order way easier with just your smartphone.

# 2. Customer Requirement

## 2.1 User Interface

**2.1.1 Home page**

* Banner to introduce the restaurant.
* Small form to quickly make a reservation.
* List of some products.
* Link to some nice article.
* Restaurant contact information.

### 2.1.2 Menu

* List of products ( sorted by newest ).
* Button to show detail about product
* Sort by vegan food or non vegan, common food or specialties from branche.

### 2.1.3 Product Detail

* The detail information about product
  + Name, image.
  + Category, price.
  + Status, cook duration.
* Add to cart.
* List of related products.

### 2.1.4 Reservation

* Reservation form.
* Best seller products.

### 2.1.5 Cart page

* List of products you have added in your cart.
* Total price.
* Delivery address (Default address would be your address of your account).

### 2.1.6 Checkout

* Notify when order success.
* Button that forwards to the home page and order history.

**2.1.7 Account profile**

* Profile image.
* Form to change profile information

### 2.1.8 Reservation history

* Show reservation history

### m2.1.9 Order history

* Show online order history.

### 2.1.10 Register page

* Form to register new account

**2.1.11 Login page**

* Form to login to your account

## 2.2 Manager Interface

**2.2.1 Account manager**

- Show all user accounts.

- Button to create new user accounts.

### 2.2.2 Branch manager

- Show all restaurant branches all over the world.

* Allow to create, edit and delete branches.

### 2.2.3 Table manager

* Show all table, rooms with their status (Booked, using, cleaning,...)
* Filtered by branch, and room.
* Add new, create, edit and delete.

**2.2.4 Food manager**

* Show all food (newest first).
* Create, edit and delete food

### 2.2.5 Order manager

* Show all order and their information (newest first)
* Only allow edit status of order (Cooked, delivered,...)

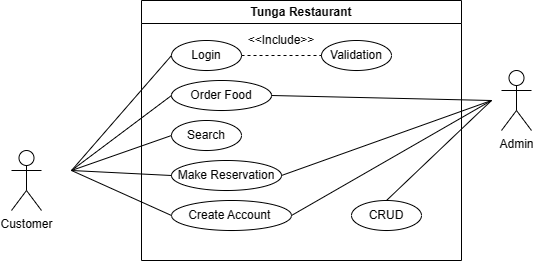
### 2.2.6 Category manager

* Show all Category and their information (newest first)
* Allow create new, edit and delete

# 3. How we design our system

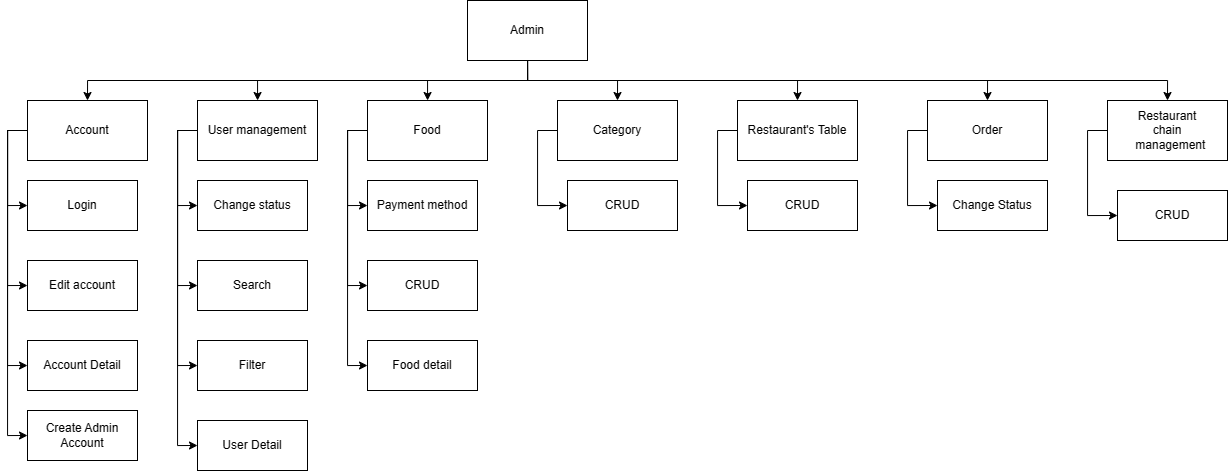


**3.1 Use-case diagram**

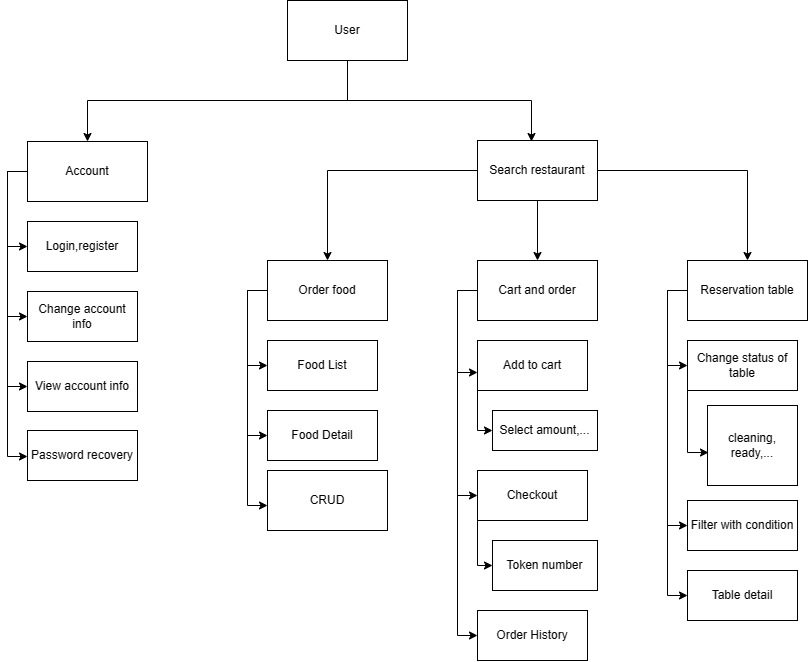


# 4. Functional Decomposition Diagram

## 4.1 Manager side

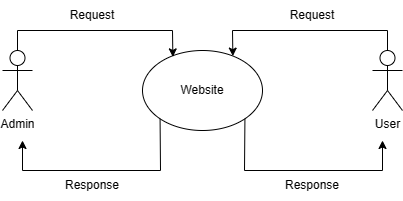


## 4.2 User side



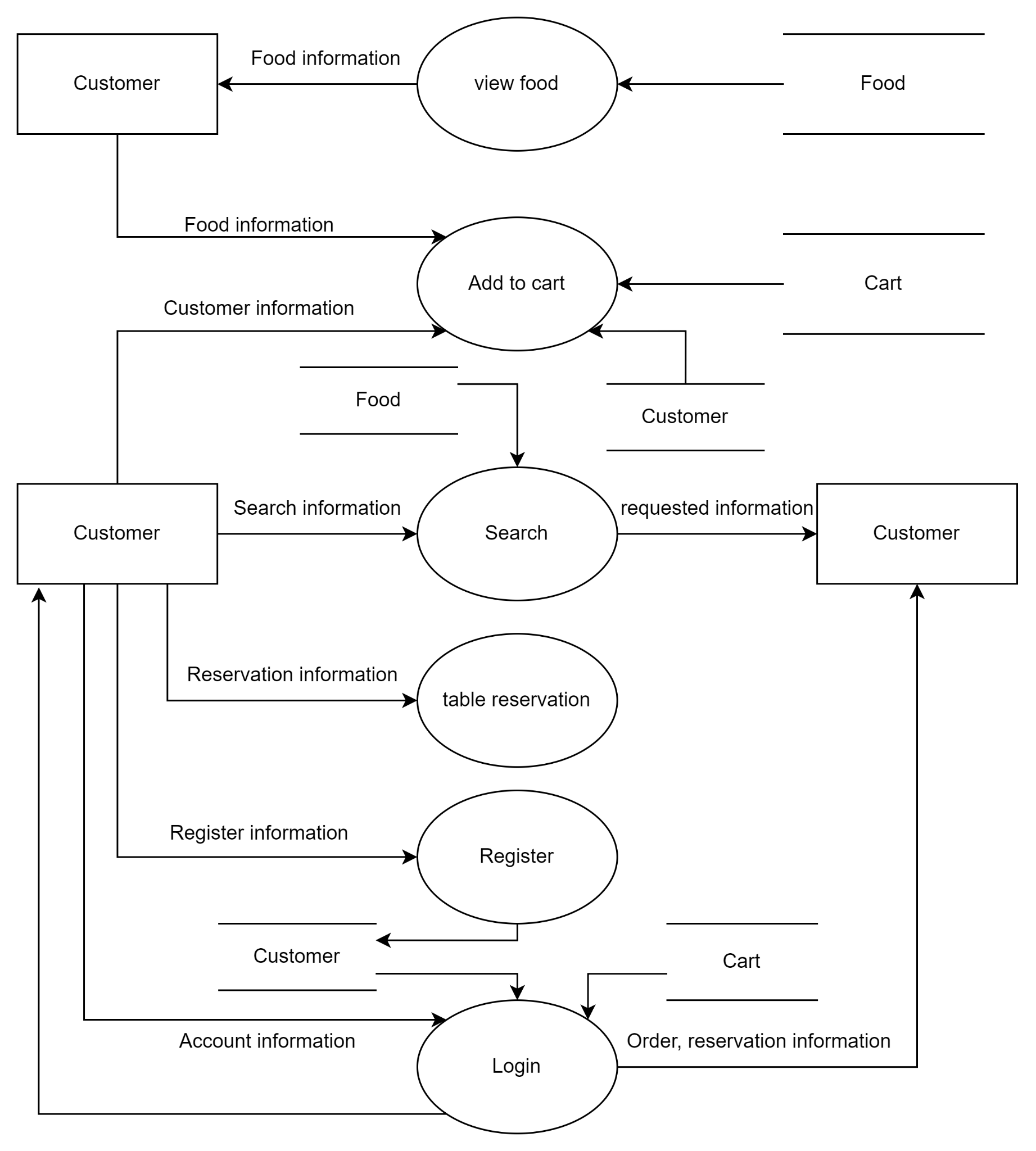
# 5. Data Flow Diagram

## 5.1 Context Diagram

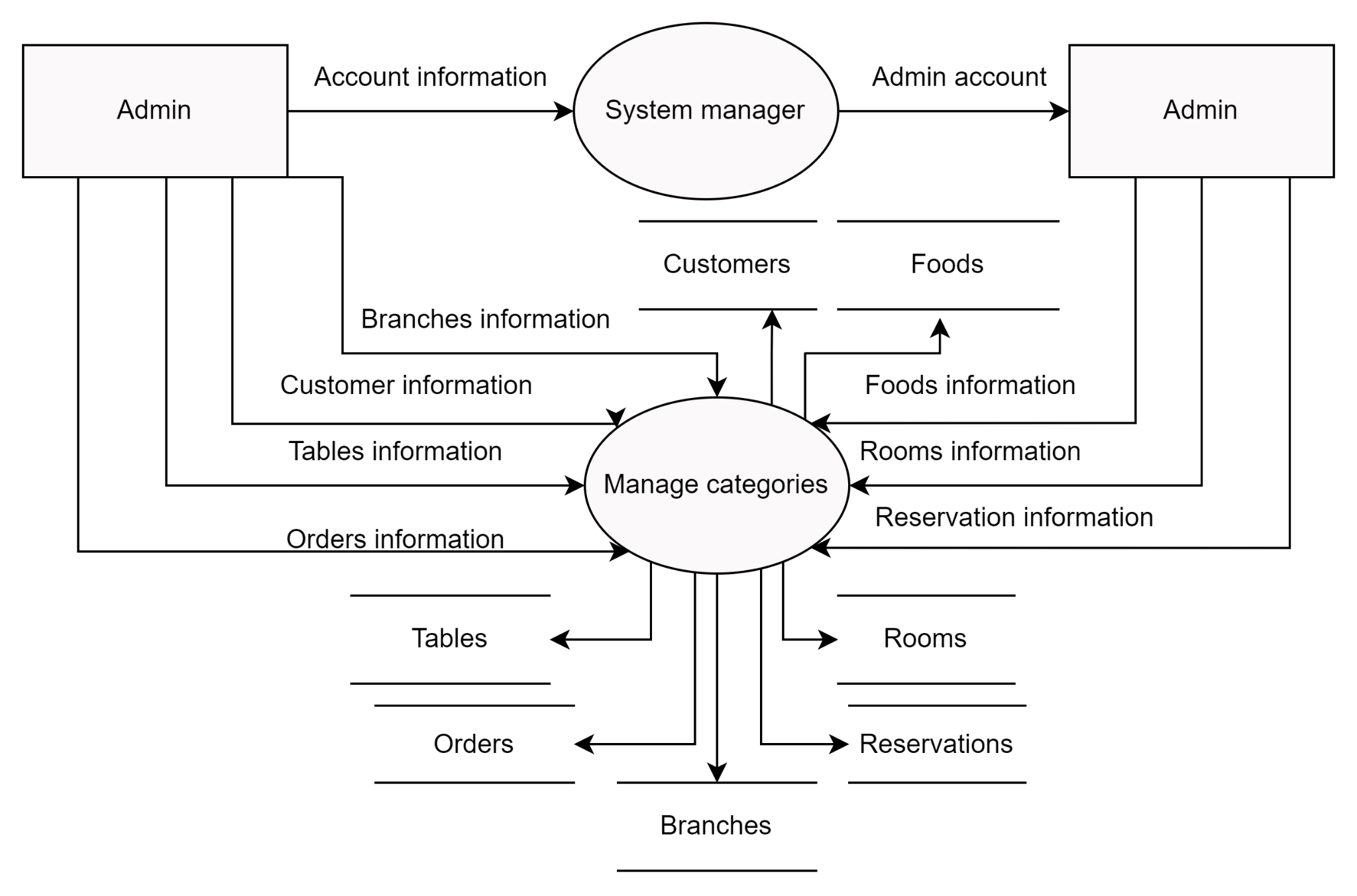


## 5.2 Detail Level Diagram

* User side:



* Admin side:



# 6. Entity Relationship Diagram

## 6.1 Entity Relationship Diagram

## 

## 

## 

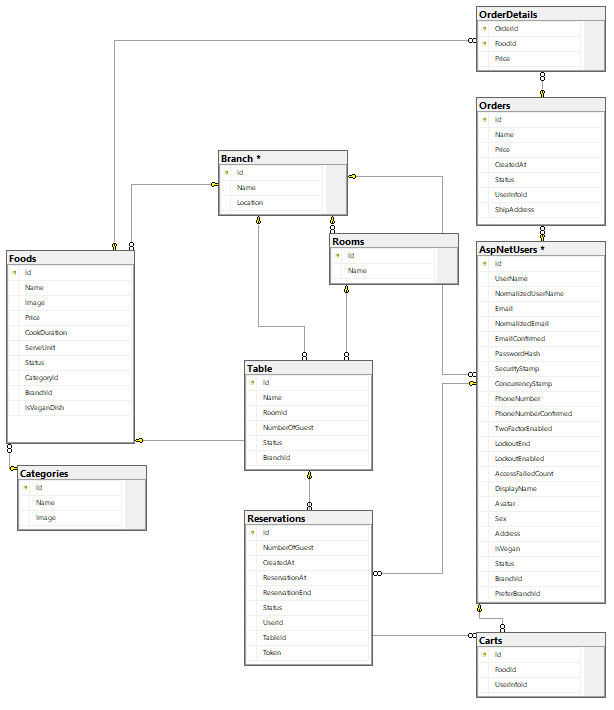
## 

## 

## 

## 

## 6.2 Entity - Attribute



**7. Data design, Data structure**

|  | **Table:Branch** | |  |  |
| --- | --- | --- | --- | --- |
| **Column name** | **Datatypes** | **Nullable** | **Key** | **Description** |
| Id | Int | ◻ | PK |  |
| Name | Nvarchar(max) | ◻ |  |  |
| Location | Nvarchar(max) | ◻ |  |  |
| Status | Int | ◻ |  |  |

|  | **Table:UserInfor** | |  |  |
| --- | --- | --- | --- | --- |
| **Column name** | **Datatypes** | **Nullable** | **Key** | **Description** |
| Id | nvarchar(450)I | ◻ | PK |  |
| UserName | nvarchar(256) | ⌧ |  |  |
| NormalizedUserName | nvarchar(256) | ⌧ |  |  |
| Email | nvarchar(256) | ⌧ |  |  |
| NormalizedEmail | nvarchar(256) | ⌧ |  |  |
| EmailConfirmed | bit | ◻ |  |  |
| PasswordHash | nvarchar(MAX) | ⌧ |  |  |
| SecurityStamp | nvarchar(MAX) | ⌧ |  |  |
| ConcurrencyStamp | nvarchar(MAX) | ⌧ |  |  |
| PhoneNumber | nvarchar(MAX) | ⌧ |  |  |
| PhoneNumberConfirmed | bit | ◻ |  |  |
| TwoFactorEnabled | bit | ◻ |  |  |
| LockoutEnd | datetimeoffset(7) | ⌧ |  |  |
| LockoutEnabled | bit | ◻ |  |  |
| AccessFailedCount | int | ◻ |  |  |
| DisplayName | nvarchar(MAX) | ◻ |  |  |
| Avatar | nvarchar(MAX) | ⌧ |  |  |
| Sex | int | ◻ |  |  |
| Address | nvarchar(MAX) | ◻ |  |  |
| IsVegan | bit | ◻ |  |  |
| Status | int | ◻ |  |  |
| BranchId | int | ◻ |  | foreign key to Branch |
| PreferBranchId | int | ◻ |  |  |

|  | **Table:Room** | |  |  |
| --- | --- | --- | --- | --- |
| **Column name** | **Datatypes** | **Nullable** | **Key** | **Description** |
| Id | Int | ◻ | PK |  |
| Name | Nvarchar(max) | ◻ |  |  |
| BranchId | int | ◻ |  | foreign key to Branch |

|  | **Table:Categories** | |  |  |
| --- | --- | --- | --- | --- |
| **Column name** | **Datatypes** | **Nullable** | **Key** | **Description** |
| Id | Int | ◻ | PK |  |
| Name | Nvarchar(max) | ◻ |  |  |
| Image | Nvarchar(max) | ◻ |  |  |
| Status | Int | ◻ |  |  |

|  | **Table:Table** | |  |  |
| --- | --- | --- | --- | --- |
| **Column name** | **Datatypes** | **Nullable** | **Key** | **Description** |
| Id | Int | ◻ | PK |  |
| Name | Nvarchar(max) | ◻ |  |  |
| RoomId | Int | ◻ |  | foreign key to Room |
| NumberOfGuest | Int | ◻ |  | max guests can serve |
| Status | Int | ◻ |  |  |
| BranchId | Int | ◻ |  | foreign key to Branch |

|  | **Table:Carts** | |  |  |
| --- | --- | --- | --- | --- |
| **Column name** | **Datatypes** | **Nullable** | **Key** | **Description** |
| Id | Int | ◻ | PK |  |
| FoodId | Int | ◻ |  |  |
| UserInforId | Nvarchar(450) | ◻ |  | foreign key to UserInforUserInfor |
| Quantity | Int | ◻ |  |  |

|  | **Table:Foods** | |  |  |
| --- | --- | --- | --- | --- |
| **Column name** | **Datatypes** | **Nullable** | **Key** | **Description** |
| Id | Int | ◻ | PK |  |
| Name | nvarchar(MAX) | ◻ |  |  |
| Image | nvarchar(MAX) | ◻ |  |  |
| Price | real | ◻ |  |  |
| CookDuration | int | ◻ |  |  |
| ServeUnit | int | ◻ |  |  |
| Status | int | ◻ |  |  |
| CategoryId | int | ◻ |  | foreign key to CategoryCategory |
| BranchId | int | ◻ |  | foreign key to Branch |
| IsVeganDish | bit | ◻ |  | Food vegan or not |

|  | **Table:Orders** | |  |  |
| --- | --- | --- | --- | --- |
| **Column name** | **Datatypes** | **Nullable** | **Key** | **Description** |
| Id | Int | ◻ | PK |  |
| Name | nvarchar(MAX) | ◻ |  |  |
| Price | real | ◻ |  |  |
| CreatedAt | datetime2(7) | ◻ |  |  |
| Status | int | ◻ |  |  |
| UserInfoId | nvarchar(450) | ◻ |  | foreign key to UserInfor |
| ShipAddress | nvarchar(MAX) | ◻ |  |  |

|  | **Table:OrderDetails** | |  |  |
| --- | --- | --- | --- | --- |
| **Column name** | **Datatypes** | **Nullable** | **Key** | **Description** |
| OrderId | Int | ◻ | PK | foreign key to Order |
| FoodId | Int | ◻ | PK | foreign key to Food |
| Price | real | ◻ |  |  |
| Quantity | Int | ◻ |  | quantity order of this food |

|  | **Table:Reservation** | |  |  |
| --- | --- | --- | --- | --- |
| **Column name** | **Datatypes** | **Nullable** | **Key** | **Description** |
| Id | Int | ◻ | PK |  |
| NumberOfGuest | Int | ◻ |  | number of pre-booked guests |
| CreatedAt | datetime2(7) | ◻ |  | creation time |
| ReservationAt | datetime2(7) | ◻ |  | Reservation start |
| ReservationEnd | datetime2(7) | ◻ |  | Reservation end time |
| Status | int | ◻ |  |  |
| UserId | nvarchar(450) | ◻ |  | foreign key to UserInfor |
| TableId | int | ◻ |  | foreign key to Tables |
| Token | nvarchar(max) | ◻ |  | token for acknowledgement |

**The end**