**Food App Database Design Documentation**

**Functional requirements**

**Customer**

* Customer can search for dishes or food-items
* Customer can add Food Dishes to his favorite items list
* Customer can add items to his cart
* Customer can check his order history
* Customer can make one or many orders
* After getting order customer can give rating to that item

***Food Seller:***

* Seller can sell at least one or many dishes
* Seller can check list of his completed orders
* Seller can add or remove discount to his items
* Seller can update or delete dishes

**Non-Functional requirements**

* The system should be capable of handle a large traffic
* Queries will be efficient
* Customers or Sellers can’t access each other’s data
* The seller cannot cancel an order

**Entities**

* **Customer –** this will hold customer information
* **Seller –** this will hold seller information
* **Dish** – this entity will hold information of dish
* **Orders –** this entity will hold information about orders made by customer
* **Categories –** this entity will hold information of food categories such as “Shakes”, “Juices”, “Ice-creams”

**Relationships Among Entities:**

**Customers and Orders**

* A customer can make one or more orders
* An order would only belong to one customer

**Orders and Dishes**

* An order can contain more than one dishes at a time
* A dish can belong to one or more orders at same time

**Dishes and Sellers**

* A seller can sell more than one dishes at a time
* A dish would belong to only one seller

**Dishes and Categories**

* One category can contain more than one dishes
* A dish would belong to one category

**Attributes**

* **Customer:**
  + *customer\_id, name, address, email, favorite\_items, cart\_items, order\_history, phone\_no.*
* **Food Seller:**
  + *seller\_id, name, address, email, food\_items, completed\_orders*
* **Dish (Food-item):**
  + *dish\_id, name, price, discount, rating, comments, seller\_id, description*