

CAPSTONE PROJECT (FOODTEK PROJECT)

GROUP (16)

Requirements Analysis and Entity Relationship Design for Delivery Management system

Web (.Net ) group

Names:

Gofran Al-Kesbah

Reem Abd-Alhadi

**Abstract:**

**Foodtek Project** is an online food ordering and delivery application that connects admins, customers, and drivers. Admins manage food items, orders, payments, and deliveries within the app. Customers can browse menus, place orders, track deliveries, and leave reviews using the application. Drivers receive orders, update delivery statuses, and view their earnings directly in the app.

**The Mandatory object:**

1. Admin
2. Customer
3. Driver
4. Food

**The Functionality:**

0- Super Admin create system and add the admin

0- Admin Account Should be available before anything

1- Admin Will log in to the system (Email, password)

2- Admin should be able to add Food Categories by

(Name of food, Description, Price, Image, Add note, Availability, Size)

3- Admin should be able Explore Food Categories by

(Name of food, Image)

4- Admin should be able to edit Food Categories by

(Name of food, Description, price, Image, availability, size)

5- Admin should be able to delete Food Categories

6- Admin should be able to Track Order (Name of food, types)

7- Admin should be able to view a list of all registered user

8- Admin should be able to view all orders placed by customers

9- Admin should be able to monitor and update status of the Food item

10- Admin should be able to Add, update, or remove food items from the menu

11- Admin should be able to Set prices.

12- Admin should be able to Generate reports about orders, sales, customer feedback, and delivery Statuses

13- Admin should be able to reject or accept order

14-Admin should be able to show View daily, weekly, or monthly discount.

15- Admin should be able to Manage payment settings

16- Admin should be able to add Driver by (Full Name, Birth of Date, email, password, Driving license, gender)

17- Customer should be able to log in

18- Customer should be able to update personal information and change the password

19- Customer should be able view order history and reorder previous meals

20- Customer should be able Explore Food Categories

21- Customer should be able to get Food item

22- Customer should be able to add food item in order

23- Customer should be able to apply discount and prom code

24- Customer should be able View order status (pending, accepted, in progress, delivered)

25- Customer should be able to manage item on the Favorites

26- Customer should be able to Track the delivery in real time

27- Customer should be able to Rate and leave feedback on meals and delivery service.

28- Customer should be able to View payment history and manage payment details

29- Customer can Pay for orders via multiple methods

30- Customer should be able to logout

31- Driver should be able to log in (Email, Password)

32- Driver should be able to update personal information and change the password

33- Driver should be able to Receive notifications of new orders to deliver

34- Driver should be able to Accept or reject delivery requests based on availability

35- Driver should be able to View customer details and delivery address

36- Driver should be able to Use GPS integration to get directions to the customer's location

37- Driver should be able to Update the status of deliveries (e.g., picked up, in transit, delivered)

38- Driver should be able to View earnings from completed deliveries.

39- Driver should be able to logout.

**Summarize the object:**

**Shared entity for each table**

* (Created By, Updated By, Is Active, Update Date, Creation Date)

1. **Admin**

* (Id, Full Name, Email, Password)

1. **Customer**

* (Id, Full Name, Email, Password, Phone Number, Birth of Date, address, gender)

1. **Driver**

* (Id, Full Name, Email, Password, Phone Number, Birth of Date, gender, driving license )

1. **Food Category**

* (Id, Types ,Description)

1. **Food Item**

* (Id, Name of food, Description, price, Image, add note, availability, size)

1. **Track Progress**

* (Id, Order Status )

1. **Order**

* (id, Order Status , item)

1. **Order Details**

* (id, Quantity)

1. **Payment**

* (id, payment method, Amount, Payment status)

**10- Total Price**

* (id, subtotal, discount, delivery charge, total)

**11- Card Details**

* (Name, Card Number, Expiry Date, CVV)

**12- Review & Rating**

* (id, rating, comment)

**13- Delivery**

* (id, Estimated Time, Delivery Address, status)

**14- History**

* (Id, Status)

**15- Cart**

* (Id, Quantity)

**Generalization and Specialization:**

1-Person is parent

Admin, Customer, Driver is child

The relationships between parent and child 🡪 one to one

2-Table order and table History We collected it in one table (order & history)

3- Table Food Category and table Food item We collected it in one table (Food)

**Relationships between Tables:**

**Admin Table:**

1. (**Admin – Food Category**) 🡪One to Many

* Admin can **create** multiple food category
* Each food category is created by one Admin

1. (**Admin – Food Item**)🡪 One to Many

* Admin can **create** multiple food item
* Each Food Item can **be created** by one Admin

1. **(Admin – Driver**)🡪 One to Many

* Admin can **add** multiple driver
* Each Driver can **be added** by one Admin

**Customer Table:**

1. (**Customer – Order)** 🡪 One to Many

* Customer can **place** many order
* Each order can **be added** by one customer

1. (**Customer – Card Details** )🡪 One to Many

* Customer can **have** multiple card details
* Each card details can **be added** by one customer

1. (**Customer - Review & Rating**) 🡪 One to Many

* Customer can **write** multiple review and rating
* Each review and rating can be **added** by one customer

1. (**Customer – Payment**)🡪 One to Many

* Customer can **make** multiple payment
* Each payment can **be added** by one customer

**Driver Table:**

1. (**Driver – Order**) 🡪 one to many

* A driver can **handle** Many orders
* Each order can **be handled** by one driver

1. (**Driver – Review & Rating**) 🡪 one to many

* A driver can **receive** many reviews and ratings.
* Each review and rating are for one driver.

1. (**Driver – Delivery**) 🡪 One to Many

* One driver can handle multiple deliveries
* Each delivery is assigned to one driver.

**Food Category Table:**

1. (**Food category – Food Item**) 🡪One to Many

* One Food category can **have** multiple food items
* Each food item belongs one food category

**Food Item Table:**

1. (**Food Item – Orders**)🡪 Many to Many via table **Order Details**

* One Order can **contain** multiple Food Items
* Each food item can **appear** in multiple orders

**Order Table:**

1. (**Order – Payment**) 🡪 one to one

* One order **has** one Payment
* One payment is **made** for one order

1. (**Order – Total Price**) 🡪 One to One

* Each order **has** one total price
* Total price **belongs** in one order

1. (**Order – Review and Rating**) 🡪 One to One

* Each Order can **have** one review and rating
* Each Review **belongs** to only one Order.

1. (**Order – History**) 🡪 One to Many

* One order can be linked to many history records.
* Each history record belongs to one order.

1. (**Order – Cart**) 🡪 One to One

* One Cart can belong to One Order
* One Order typically comes from One Cart

1. (**Order** – **Track Progress**)🡪 One to Many

* One order can have many tracks progress entries
* Each track progress entry is associated with one order.

**Payment Table:**

1. (**Payment – Card Details**) 🡪 Many to One

* One Card details can be **used** for multiple Payment
* Each Payment **uses one card details**

**Delivery Table:**

1. (**Delivery – Order**) 🡪 One to One

* Each order **has** one delivery
* The delivery **belongs** for one order

**Cart Table:**

1. (**Cart Table – Customer**) 🡪 One to Many

* One customer can have multiple carts
* Each cart belongs to one customer

**Lookups table**

1.Country Code (id, code, Name)

2-Card Type (id, type)