**Database Systems Project**

Tasmiya Malik, Ifrah Ilyas, Hana Ali Rashid

**Project**

Catering Service Management System

**Description & Assumptions**

The system will function as follows:

* A *Customer* will enter their details in the system.
* They will then choose *FoodItems* and add them to their *Order*.
* Each *FoodItem* will belong to a *category* such as Rice, Burgers, Rolls, Frozen etc.
* Each *FoodItem* will also have a *serving size* (Single/Family) that will determine the *amount* (in grams/number of pieces depending on food) as well as *unit price*.
* A weekly menu will be displayed. Can customers only access items from that day?
* There are also different deals the customer can choose. A deal may be permanent or be only valid for a particular time interval.
* When checking out, the *Customer* will determine the *payment method* (online or cash on delivery). The *total cost* of the order will also be calculated and *delivery charges* will be added as well.
* Delivery charges are fixed regardless of customer location.
* The system will confirm the order once it determines whether there are enough ingredients available for the order, otherwise it will delete the order.
* It will update the *ingredients* left if the order is confirmed.
* The *status* of the order (In Process or Delivered) will also be maintained.
* The *Riders* will be assigned their orders based on the order *region*. They will record the cash they *receive* and any cash they *return* to the customer. Once the order has been delivered and money has been received, the Rider will update the order status and the order will officially be complete.
* Cash returned/received is only applicable for orders where payments are not made online.
* The order status will initially be “In Process” after the order is confirmed. It will be updated to “Delivered” by the Rider when they deliver it.

Feedback:

Riders with minimum assigned orders will be selected.

Rider login

Table for discount → link to order/item e.g. >10 quantity = some discount

Store customer info or check if the email is already in database and retrieve

User account table → select role

If this item chosen then according to ingredient this is max number you can order

Riders can pick order(s) themselves and optimize route themselves

No need for isconfirmed, if not enough then display error message

Dont need ispermanent, just check if validtill is null or not.

Weekly menu selected by admin from drop-downs: dynamically create combo-boxes if “add Item” is selected

Separate table for payment

Separate table for delivery charge if it is fixed, don’t need an attribute, hardcode

Separate region table, tied to order, rider region ID, dont add to customer,

Customer can only order from weekly menu

Link order and deal tables somehow, eg many to many