CS 387 Project - Restaurant Management System

Project Design document

04 April, 2022

Team YARA:

190050107 Satwik M

190050124 Thivesh M

190050038 Akash G

190050026 Karthikeya B

Table of contents:-

a)TABLES

b)INTEGRITY CONSTRAINTS

d,f)Transactions SQL queries

i) Business Logic Controller

h) User forms

a)TABLES:

```
1)item(item_id,item_name,cost,qua_inv,unit)
2)dish(dish_id,dish_name,recipe,time_taken,dish_type,cost,rating,photo)
3)dish_items(dish_id,item_id,quantity)
4)area(area_id,loc,city)
5)customer_type(c_type_id,c_type,min_num_dishes,max_num_dishes)
6)customer(c_id,name,username,pswd,ph_no,addr,num_orders,num_dish)
7)cart(c_id,dish_id,quantity)
                     8)employee(e_id,name,username,pswd,salary,ph_no,addr,e_type,join_date,status,left_dateprim_area_id,sec_area_id)
9)offer(offer_id,name,discount)
10)day(dat,day)
11)offer_valid(offer_id,dat,dish_id,c_type_id)
12)day_to_day_dishes(dat,dish_id,dish_count)
13)day_to_day_items(dat,item_id,used,bought,left_inv)
                     14)orders(order_id,c_id,area_id,table_id,dat,received_time,finished_time,delivered_time,deliver_person,status,order_type)
15)order_dishes(order_id,dish_id,quantity,offer_id)
16)table_status(table_id,loc,status)
```

There are no functional dependencies in our shema. Hence it is 3NF form.

b)INTEGRITY CONSTRAINTS:

1)item:

primary key(item_id)

2)dish:

primary key(dish_id)
dish_type in ('Veg Starter','Nog-Veg Starter','Veg Main','Non-Veg Main','Deserts')

3)dish_items:

Primary key(dish_id,item_id)
Foreign key(dish_id) references dish
Foreign key(item_id) references item

4)area:

Primary key(area_id)

5)table_status:

Primary key(table_id)
Status in ('O','E') - 'O' is for Occupied and 'E' is for Empty

6)customer_type:

Primary key(c_type_id)
C_type in ('Platinum','Gold','Silver','Normal')

7)customer:

Primary key(c_id)
unique(username) - each customer should have unique username
Username not null
Pswd not null

8)cart:

Primary key(c_id,dish_id)
Foreign key(c_id) references customer
Foreign key(dish_id) references dish

9)employee:

Primary key(e_id)
unique(username) - each employee should have unique username

Username not null
Pswd not null
E_type in ('Chef','Waiter','Head Waiter','Delivery','Manager')
Status in ('Working','Leave','Left')
Foreign key(prim_area_id) references area
Foreign key(sec_area_id) references area

10)offer:

Primary key(offer_id)

11)day:

Primary key(dat)

Day in ('Mon','Tue','Wed','Thu','Fri','Sat','Sun')

12)offer_valid:

Unique(offer_id,dat,dish_id,c_type_id)
Foreign key(offer_id) references offer
Foreign key(dat) references day - Can be null, null value means that this offer is valid on all days
Foreign key(dish_id) references dish - Can be null, that means offer is valid for all dishes
Foreign key(c_type_id) references customer_type - Can be null, that means offer is valid for all customer types
offer_id not null

13)day_to_day_dishes:

Primary key(dat,dish_id)
Foreign key(dat) references day
Foreign key(dish_id) references dish

14)day_to_day_items:

Primary key(dat,item_id)
Foreign key(dat) references day
Foreign key(item_id) references item

15)orders:

Primary key(order_id)

Foreign key(c_id) references customer

Foreign key(area_id) references area - null value means it's a Dine-in order

Foreign key(delivery_person) references employee - null value means it's a Dine-in order

Foreign key(table_id) references table_status - null value means it's a Dine-in order

Foreign key(dat) references day

Status in ('Preparing', 'Out for delivery', 'Delivered', 'Served')

Order_type in ('Online','Dine')

Dat not null

16)order_dishes:

Primary key(order_id,dish_id)

Foreign key(order_id) references orders

Foreign key(dish_id) references dish

Foreign key(offer_id) references offer - null value means no offer was applicable

c)Not using any views.

d,f) Transactions in DB for each use case and the corresponding SQL query

For Everyone:-

Use Case: Login

1. If the role is Customer then we search for the corresponding row in the customer table -

SELECT * FROM customer where username = u_name and pswd = pwd

2. If the role is any other then we search for the corresponding row in the employee table -

SELECT * FROM employee where username = u_name and pswd = pwd

For Owners:-

Managing Employees -

Use Case: View Employee details

1. Query all existing employee records's name

SELECT * from employee

2. Return details of a single selected employee

SELECT * from employee where e_id= employee_id

Use Case: Delete Employee details

1. Delete employee record details from the table

DELETE FROM employee WHERE e_id =employee_id

Use Case: Edit Employee details

Edit employee record details from the table
 UPDATE employee SET status=new_state where e_id = employee_id

Use Case: Add Employee details

1. Add a new entry into the employee table

INSERT INTO employee

VALUES (name, salary, ph_no, addr, username, pswd, e_type, join_date, status, left_date, prim_area_id, sec_area_id)

Managing Dishes -

Use Case: View Dish details or Menu

- Query all records' names from the dish table SELECT * from dish
- Return details of a selected dish SELECT * from dish where dish_id = dish_id

Use Case: Delete dish

 Remove the particular dish record from database DELETE FROM dish WHERE dish_id = dish_id; DELETE FROM dish_items WHERE dish_id = dish_id;

Use Case: Add Dish details

Insert a new entry to the dishes table
 INSERT INTO dish VALUES (dish_id, dish_name, recipe, time_taken, dish_type, cost, rating, photo);
 INSERT INTO dish_items VALUES (dish_id, item_id, quantity);

Restaurant Details

Use Case: View and edit Restaurant details

1. These details will be stored as a JSON file.

Managing Items

Use Case: View Item details

1. Query all item names from the item table.

SELECT * from item;

2. Query a particular item record from the item table.

SELECT * from item WHERE item_id = item_id;

Use Case: View Items purchased and used up on that day

Fetch all items from day-to-day-items relation.
 SELECT * from day_to_day_items where day = d;

Use Case: Update item count in the inventory

Update the qua_inv attribute for that item in the item table
 UPDATE item SET qua_inv=new_quantity WHERE item_id = item_id;

Managing Tables

Use Case: View Table Status

1. Fetch all tables from the table relation SELECT * from table_status;

2. Fetch further details about a table from the table relation SELECT * from table_status where table_id = t_id

Use Case: Edit Table Status

1. Edit the status attribute of a table UPDATE table_status SET status=new_state WHERE table_id = table_id;

Managing Orders

Use Case: View order history details

1. Fetch all order's details from the orders relation SELECT * from orders

2. Fetch complete details of a particular order from the orders relation SELECT * from orders where order_id = order_id

Use Case: Add an order

1. Adds a row to the orders table, adds dishes in the orders to the order_dishes table

INSERT INTO orders
VALUES (c_id, area_id, table_id, dat, received_time, finished_time, delivered_time, delivery_person, status, order_type)
INSERT INTO order_dishes

INSERT INTO order_dishes
VALUES (order_id, dish_id, quantity, offer_id)

Manage Customers

Use Case: View customer details

- 1. Fetch details of all customers from the customers relation SELECT * from customer
- Fetch complete details of a particular customer.SELECT * from customer WHERE c_id = c_id

Use Case: Add a new customer

1. Adding a new row to the customer table

INSERT INTO customer VALUES (c_id,name,username,pswd,ph_no,addr,0,0)

Use Case: Edit customer details

 The corresponding row in the customer table is updated UPDATE customer
 SET name=name AND ph_no = ph_no AND addr=addr
 WHERE c_id = customer_id

Manage Offers

Use Case: View offer details

1. Fetch all offers from the offers relation SELECT * from offer

2. Fetch details about a particular offer SELECT * from offer WHERE offer_id = offer_id

Use Case: Add an offer to the offer table and offer_valid tables

 A new row is added to the order table and rows are added to the offer_valid tables INSERT INTO offer VALUES (offer_id,name,discount)

INSERT INTO offer_valid VALUES (offer_id,dat,dish_id,c_type_id)

Manage Delivery People

Use Case: View delivery person details

1. Fetch all delivery person from the employees table / delivery person table SELECT * from employee where e_type = 'Delivery'

Use Case: Edit delivery person details

Change the area codes for a particular Delivery person
 UPDATE employee SET prim_area_id=p_id and sec_area_id = s_id WHERE e_id = e_id

For Customers:- (View Menu,

Use Case: View cart and confirm the order

- 1. Viewing rows corresponding to the customer in the cart table SELECT * from cart where c_id = c_id;
- 2. On confirmation, insert a new order into the orders relation. Already written

Use Case: View previous orders

- Fetch orders from orders table, filter on a particular customer.
 SELECT * from orders where c_id = c_id;
- 2. Fetch a particular order detail, on selection. Already written

Use Case: Change account details - Already written

- 1. Fetch existing account details from Customers table and login table.
- 2. Modify customer details record in the customers table.

For Chef:-

The relevant subset of Owner use cases - (Manage Dishes, Menu, Recipe of dishes)

For Head Waiter:-

The relevant subset of Owner use cases - (Manage Table Status)

For Inventory Manager :-

The relevant subset of Owner use cases - (Manages inventory of items)

For Billing Manager:-

The relevant subset of Owner use cases - (Managers Orders and Customers)

For Delivery Manager:-

The relevant subset of Owner use cases - (Manages Delivery Persons and Area Codes)

Analytics Use cases

- best dishes according to user rating SELECT * from dish ORDER BY rating limit 10;
- 2. Most ordered dish

SELECT dish_name FROM

((SELECT dish_id from order_dishes) as o_
UNION

(SELECT dish_id from cart) as c_)as d_,
dish

where dish.dish_id =d_.dish_id

GROUP BY dish_id

ORDER BY count(dish_id) DESC
limit 1;

3. Best day with most orders

SELECT dat from day_to_day_dishes GROUP BY dat ORDER BY count(dat) DESC limit 1;

I. Business Logic Controller :-

For Everyone:-

Use Case: Login

Users need to sign up at first, before logging into the web app. From next time onwards, they can directly sign in.

For Owners:-

Use Case: View Employee details

All employees previously added via the "Add Employee" option and those already existing in the database will be visible. A particular employee's details can be brought into focus by clicking on it.

Use Case: Delete Employee details

Employee records will be deleted. There is no way to recover these details.

Use Case: Edit Employee details

An existing employee's details such as name, ph.no, job can be edited.

Use Case: Add Employee details

A new employee has to be added to the database using this option. The new employee can have the same name as another employee, since we will have a unique employee ID.

Use Case: View Dish details

All dishes added previously via the "add dish" option and those already present in the database will be displayed. A particular dish can be viewed in focus by clicking on it.

Use Case: Delete dish

This particular dish will be removed from the database. This operation is irreversible.

Use Case: Add Dish details

A new dish is added into the menu using this option. The dish name needs to be unique, to avoid confusion for customers.

Use Case: View and edit Restaurant details

Restaurant details can be edited and saved. We need a photo, address, ph.no and a neat description.

Use Case: View Item details

All items added previously via the "add item" option and those already present in the database will be displayed. A particular item can be viewed in focus by clicking on it.

Use Case: View Items purchased and used up on that day

Items used up will be reflected in the dishes-orders served. Items purchased will be managed by the Inventory Manager.

Use Case: Update item count in the inventory

Item count will be updated in the database. This has to reflect the orders placed and inventory bought.

Use Case: View Table Status

The status will be free/occupied/reserved.

Use Case: Edit table status

Can update the status of tables. Free tables can be made busy/reserved. Busy table can be freed.

Use Case: View order history details

All previous orders are displayed. Click on a particular order to view it in focus.

Use Case: Add an order

Create an order manually after taking orders from dine-in customers. Online customers will place orders under their own name.

Use Case: View customer details

All current customers are displayed. Click on a particular customer to view them in focus.

Use Case: Add a new customer

This option is used to register new customers offline.

Use Case: Edit customer details

This option is used to edit details offline.

Use Case: View offer details

All current offers are displayed. Offers typically allot discounts on total bill.

Use Case: View delivery person details

Delivery persons details are displayed. Emphasis is kept on whether they are free or currently doing a delivery.

For Customers:-

Use Case: View menu and select items

All items are displayed to the user. They can select a particular item for ordering.

Use Case: View cart and confirm the order

After viewing their items again in the cart, they can apply for an offer and confirm their order.

Use Case: View previous orders

All previous orders of that user are visible. Clicking on a particular order will open it in focus.

Use Case: Change account details

Users can change their details like name, phone no., Address etc.

For Chef:-

The relevant subset of Owner use cases - (Manage Dishes, Menu, Recipe of dishes)

For Head Waiter:-

The relevant subset of Owner use cases - (Manage Table Status, Manage Waiters)

For Inventory Manager :-

The relevant subset of Owner use cases - (Manages inventory of items)

Use case: Update about items

Details about how much of which item was bought will be updated here. This will also update the attributes of that item in the Items relation.

For Billing Manager:-

The relevant subset of Owner use cases - (Managers Orders and Customers)

For Delivery Manager:-

The relevant subset of Owner use cases - (Manages Delivery Persons and Area Codes)

Use Case: Assign a delivery person

Delivery manager will assign a delivery person to make a delivery to an online customer. Manager needs to take care of assigning the right delivery location to that delivery person.

H. User forms:

Login	Page
Username	
Password	
Role	Dropdown
Lo	ogin
New Customer?	

New Cu	ustomer
Username:	
Password:	
Name:	
Phone No:	
Address:	
S	ubmit
	Login Page

Customer Home page

New Or	der Cart	Prev Orders My Details
	Dish Type	Dropdown
List of di	shes	
Photo	Dish Name	Cost Offer - 0 +
Photo	Dish Name	Cost Offer - 0 +
Photo	Dish Name	Cost Offer - 0 +

Cart

Dishes in cart	
Photo Dish Name Cost Offer - 0	+
Photo Dish Name Cost Offer - 0	+
Total Cost Delivery Charges	
Offer Final Amount	
Pay	

		Cusomter Details		
Username: [Name: [Ph. No: [Address: [Type: No. of orders: No. of dishes:	
Edit De	etails		Cl	hange password

	Dish Name	
	Name : Ingredients:	
Photo	Cost:	
	Rating:	
	Offer:	
	- 0 +	

Previous Orders

List of orders Order # Dishes in Order	
Photo Dish Name Cost C	Offer - 0 +
Photo Dish Name Cost C	Offer - 0 +
Total Cost Delivery Cha	
Delivery Time Delivered By	

Employees Page

List of Er	nployees		
ID	Name	Role	Edit Remove
ID	Name	Role	Edit Remove
	Add	Employee	

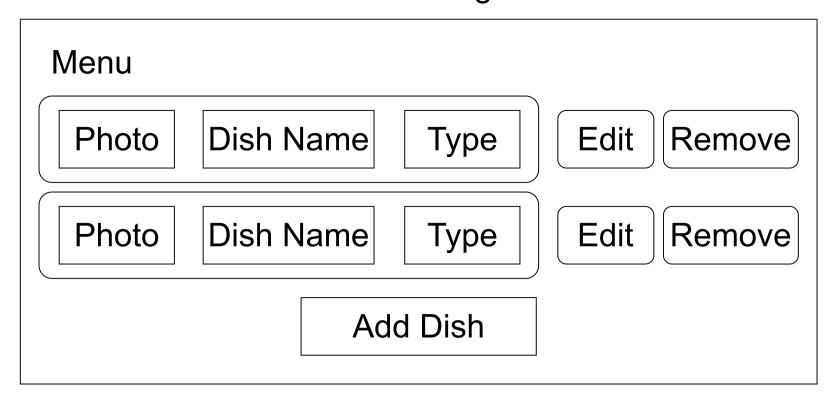
Add Employee Page

Username:	Address:	Primary Delivery
Name:	Role:	Area
Ph. No:	Pwd:	Secondary Delivery Area
	Add	, 00

Edit Employee Page

Username:	old username	Address:	old address
Name:	old name	Role:	old role
Ph. No:	old ph.no	Pwd:	old pwd
status:	Working		
		Edit	

Dishes Page

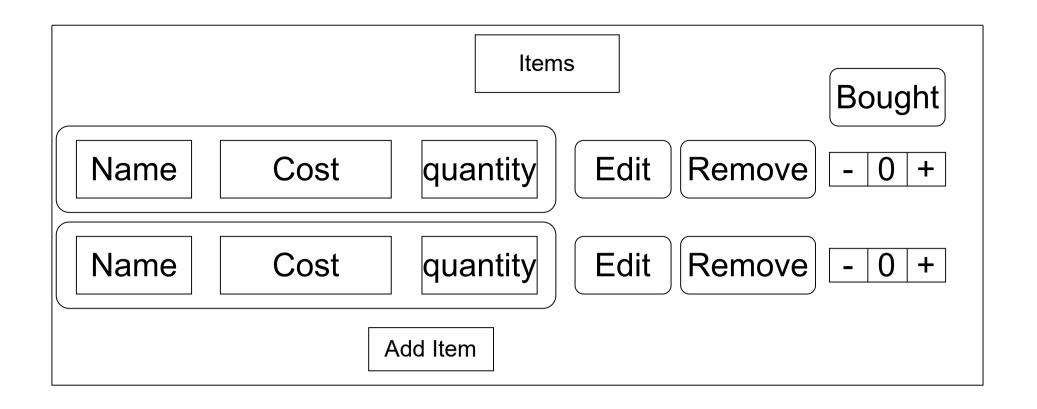


Add Dish Page

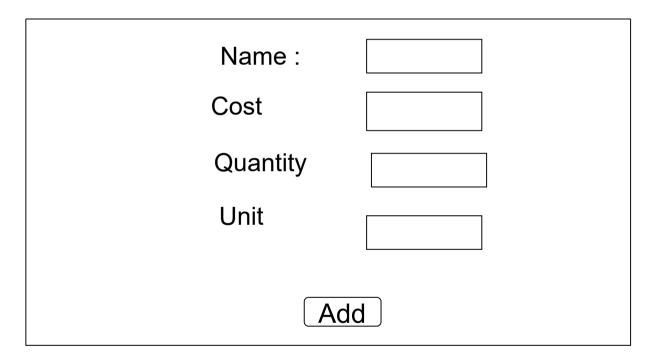
Γ		Name :
	Photo	INAITIE .
		Ingredients:
		Recipe:
		Type:
		Time Taken:
Add		

Edit Dish Page

	Photo	Name : old name
		Ingredients: old ingridents
		Recipe: old recipe
		Type: old type
		Time Taken: old time
		Edit



Add Item Page



Edit Item Page

Name :	old name
Cost	old cost
Edit	

Offers Page

Offer Name	Discount	Edit	Remove
Offer Name	Discount	Edit	Remove
	Add Offe	er	

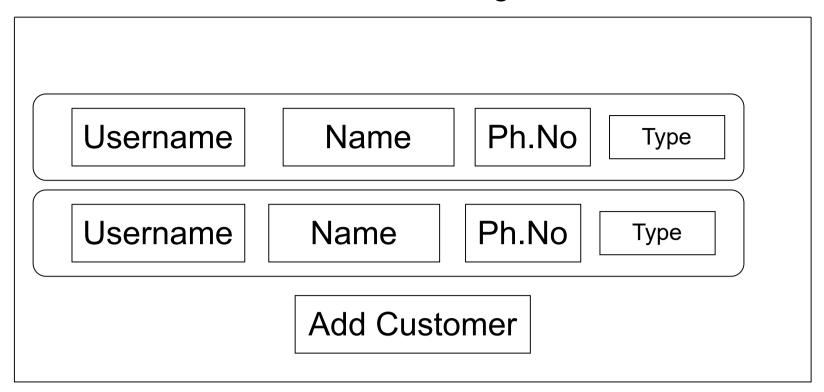
Add Offer	
Name:	
Discount:	
Dates: - date +	
Dishes: - dish +	
Add	

Edit Offer	
Name: name	
Discount: percentage	
Dates: - dates +	
Dishes: - dishes +	
Save	

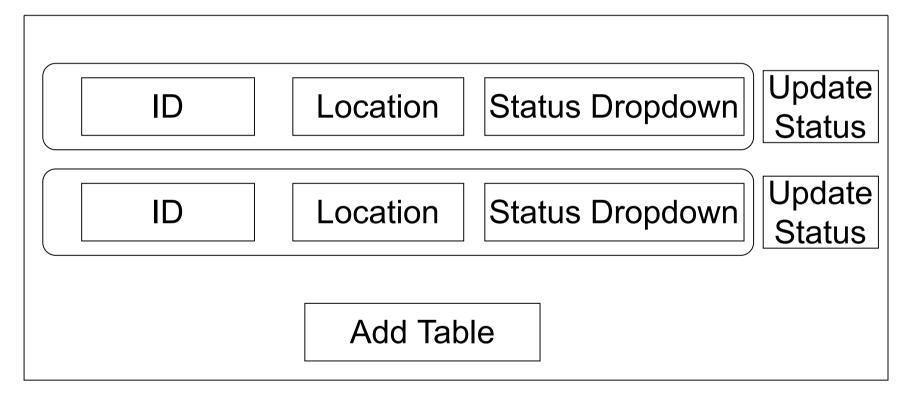
Per Day Page

Date:		
Items List		
Item	Used Count Bought Count	
DIshes List		
	Dish Count	
Sales Made Amount Spent		
	Net Profit	

Customers Page



Tables Page



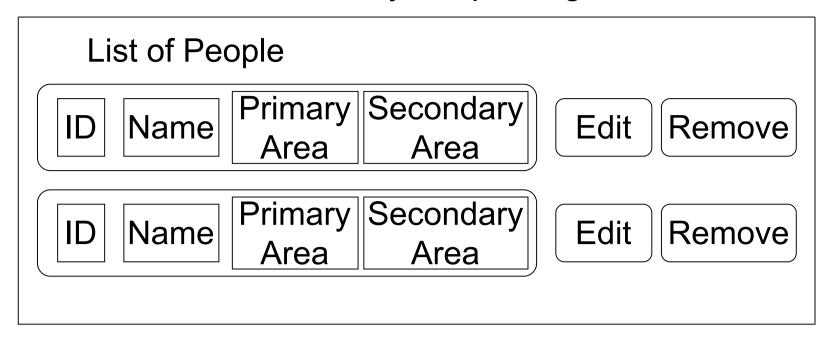
Add Table Page

Location	
Add	

Orders Page

Date Customer Name Order Type Bill status		
Date Customer Name Order Type Bill status		
Add Order		
Add Order Page Customer: Order Type: Dishes: - dish,quantity + Place Order		
Analytics Page Best Dish by ratings: Best Dish by no. of orders: Most often visiting cutomers: Best delivery boy by no. of deliveries: Most used type of order:		

Delivery People Page



Delivery Person Page

