Requirements and Analysis Phase

TEAM:

Ramineni Amith Varma - 190050099
Sunkari Pranay Varma - 190050120
Vishnu Vardhan A - 190050130
Yallanki Prathyusha - 190050132

PART A:

Short Description

The restaurant management application basically manages all the restaurant operations in one place. It provides all the requirements of managing a single place. Users and restaurant staff can use the application to make progress of every task they do or every order they complete. Everything with a time log is recorded in the database with operations to keep track of everything. It provides user-based classification with one platform for both customers and the staff.

Why Database

The file system just provides the details of the data representation and storage of data about the restaurant whereas using a Database helps increase organizational accessibility to data which helps to share the data and its analysis quickly and effectively across the application. Management helps get quick solutions to database queries, thus making data access faster and more accurate. Databases help in keeping track of staff actions, orders, purchases, and sales, etc. As we are handling large sets of multiple types of data for our application, it would be better to use a database over other alternatives. Helps us to divide the data into categories and is well structured.

What kind of Database

Relational Database for storing data about the users, items, dishes, ingredients,...etc. GraphDB for having statistics of purchases, food consumed at each time of the day,...etc. Spatial Databases for estimating the time of delivery for the customer location from the restaurant

Domain

A web application with AngularJS frontend and Backend with PostgreSQL database for accessing and storing various types of data in restaurant management

PART B:

Intended Classes of Users

SuperUser, Base Customer, Premium Customer, General Manager, Kitchen Manager, Billing Manager, Delivery Manager, Food Server, Chef, Delivery Person

Use Cases:

1. **LOGIN**
   DESCRIPTION: Logging a user into the application
   TRIGGER: automatic trigger
   ROLE: Base Customer or Premium Customer
   INPUTS: person email, password
   PRE-CONDITIONS: Should be signed up before logging in
   MAIN SUCCESS PATH: Redirecting to the dashboard of the user
   EXCEPTIONS: Email and password invalid
   POST CONDITIONS: checking in the database for email and password matching

2. **ADDING ITEMS**
   DESCRIPTION: Items and ingredients are posted into the database for orders
   TRIGGER: human trigger
   ROLE: Kitchen Manager, General Manager, SuperUser
   INPUTS: item_name, item_type, item_availability, item_price, quantity
   PRE-CONDITIONS: Sufficient ingredients for this item must be present
   MAIN SUCCESS PATH: item entry is added into the items entity and a success message is displayed
   EXCEPTIONS: ingredients present are not sufficient
   POST CONDITIONS: ingredients present are checked for the item to be added

3. **ADDING ADMINS (STAFF)**
   DESCRIPTION: admins are added and deleted frequently
   TRIGGER: human trigger
   ROLE: General Manager, SuperUser
   INPUTS: person_name, person_type, type_from, type_to, address, phone_no
   PRE-CONDITIONS: None
   MAIN SUCCESS PATH: Person is added into the person entity
   EXCEPTIONS: type_from and type_to are invalid
   POST CONDITIONS: type_from and type_to are checked for validity

4. **ONLINE ORDERS**
   DESCRIPTION: orders are placed by users through an online application
   TRIGGER: automatic trigger

ROLE: Base Customer, Premium Customer
INPUTS: item_id, quantity, delivery_address
PRE-CONDITIONS: User must be enrolled as base or premium user and the item must be available
MAIN SUCCESS PATH: Order will be placed and delivery person details are displayed
EXCEPTIONS: Item is not available or delivery is not available
POST CONDITIONS: Item availability and Delivery person availability is checked

5. **ADDING COUPONS**
DESCRIPTION: coupons are added for the sake of premium customers
TRIGGER: human trigger
ROLE: General Manager, Billing Manager
INPUTS: coupon_txt, coupon_type, availability, start_date, end_date
PRE-CONDITIONS: None
MAIN SUCCESS PATH: The coupon is added and a success message is displayed
EXCEPTIONS: start_date, end_date are invalid
POST CONDITIONS: Checks for the validity of start_date and end_date

6. **VIEW STAFF**
DESCRIPTION: The manager can view all the details of staff
TRIGGER: human trigger
ROLE: General Manager, SuperUser
INPUTS: None
PRE-CONDITIONS: The manager account should be active
MAIN SUCCESS PATH: All persons in staff are fetched and displayed
EXCEPTIONS: Manager account is inactive or there is no staff
POST CONDITIONS: Fetches all persons in the staff

7. **ADDING DELIVERY PERSONS**
DESCRIPTION: adding delivery persons for the sake of delivery for online orders
TRIGGER: human trigger
ROLE: General Manager, Delivery Manager
INPUTS: dp_name, phone_no, primary, secondary
PRE-CONDITIONS: The manager account should be active
MAIN SUCCESS PATH: A delivery person will be added and a success message is displayed
EXCEPTIONS: The manager account is inactive
POST CONDITIONS: adds the delivery person to the delivery person entity

8. **OFFLINE BOOKING**
DESCRIPTION: offline booking is done in the restaurant in-person
TRIGGER: human trigger
ROLE: Base Customer, Premium Customer
INPUTS: item_name, quantity, order_date, order_time

PRE-CONDITIONS: None
MAIN SUCCESS PATH: Order is booked and a success message is displayed
EXCEPTIONS: Item is not available
POST CONDITIONS: Checks if the item is available

9. **VIEWING AVAILABLE ITEMS**
DESCRIPTION: customers can view available items in their dashboard
TRIGGER: automatic trigger
ROLE: Base customer, Premium Customer
INPUTS: None
PRE-CONDITIONS: The user must be signed up
MAIN SUCCESS PATH: Items will be fetched and displayed
EXCEPTIONS: No items are available
POST CONDITIONS: Checks if any items are available

10. **POSTING ITEM FEEDBACK**
DESCRIPTION: User can post feedback of item he purchased
TRIGGER: human trigger
ROLE: Base Customer, Premium Customer
INPUTS: feedback_txt, item_name, suggestions, rating
PRE-CONDITIONS: The user must be signed up and purchased the item
MAIN SUCCESS PATH: Feedback will be posted
EXCEPTIONS: The user has not purchased this item
POST CONDITIONS: Checks if the user already purchased this item

11. TABLE RESERVATION

DESCRIPTION: booking a table by a customer
TRIGGER: human trigger
ROLE: Base Customer, Premium Customer
INPUTS: table_type, minimum_capacity
PRE-CONDITIONS: The user must be signed up
MAIN SUCCESS PATH: data in the table_person relationship will be added
EXCEPTIONS: when all the tables are reserved, the user can’t book
POST CONDITIONS: checks if tables are available

PART C:

Entities

1. **Items**

   item_id, int - Primary Key
   item_name, text
   item_type, text
   availability, int
   price, int
   Item can be of food or beverage
   Item_type = food or beverage
   availability = can be bool (yes if available) or int(number of specific items available)

2. **Ingredients**

   ing_id, int - Primary Key
   ing_name, text
   availability, int
   price, int
   availability = can be bool (yes if available) or int(number of specific items available)

3. **Persons**

   person_id, int - Primary Key
   person_name, text
   person_type, text
   type_from, date
   type_to, date
   address, text
   phone_no, int
   person_type = type of role he/she has
   type_from, type_to = subscription from and to dates

4. **Purchases**

   purchase_id, int
   purchase_name, text
   date, date
   time, time
   quantity, int

5. **Tables**

   table_id, int - Primary Key

table_type, text
capacity, int
price, int

6. Item_Feedback

item_f_id, int - Primary Key
feedback_txt, text
suggestions, text
rating, int

7. Dp_Feedback

dp_f_id, int - Primary Key
feedback_txt, text
suggestions, text
rating, int

8. Offline Orders

off_order_id , int
quantity , int
order_price , int
order_date , date
order_time , time

Primary Key - {off_order_id, item_id}

9. Online Orders

on_order_id , int
quantity , int
order_price , int
order_date , date
order_time , time
delivery_address , text
is_delivered , bool
is_cancelled , bool
estimated_time , time
delivery _date , date
delivery_time , time

Primary Key - {on_order_id, item_id}

10. Coupons

coupon_id, int - Primary Key
coupon_txt **,** text
coupon_type , text
availability , int
start_date , date
end_date , date

11. Delivery Persons

dp_id , int - Primary Key
dp_name , text
dp_rating , int
primary , int
secondary , int
phone_no   , int

12. Cancellations

c_id , int - Primary Key
c_reason , text
c_date , date
c_time , time

Relationships

1. Item Ingredients

Connects items and ingredients entities with Many to Many relationships
One item has many ingredients, One ingredient may be in many items

2. Purchased Ingredients

Connects Ingredients t**o** purchases entities with Many to Many relationship
One ingredient may be purchased in many purchase_id's. Onr purchase may contain many ingredients

3. Items sold to user

Connects Items and Persons with Many to Many relationships
One item may be sold to many persons. One person may purchase many items

4. Feedback to item

Connects item_feedback to items with Many to One relationship
One feedback implies only one item. One item may have many feedbacks

5. Feedback by user for item

Connects item_feedback to persons with Many to One relationship
One feedback implies only one user. One user may give many feedbacks

6. **Feedback by user for delivery persons**
   Connects dp_feedback to persons with Many to One relationship
   One feedback implies only one user. One user may give many feedbacks

7. **Feedback to delivery persons**
   Connects dp_feedback to delivery persons with Many to One relationship
   One feedback implies only one delivery person. One delivery person may have many feedbacks

8. **Order delivered by**
   Connects online_orders to delivery persons with Many to One relationship
   One order implies only one delivery person. One delivery person may have many orders to deliver

9. **Online Ordered by User**
   Connects online_orders to persons with Many to One relationship
   One order implies only one user. One user may give many orders

10. **Online Ordered Item**
    Connects online_orders to items with Many to Many relationships
    One order may have many items. One item may be in many orders

11. **Offline Ordered Items**
    Connects offline_orders to items with Many to Many relationships
    One order may have many items. One item may be in many orders

12. **Coupons offered to persons**
    Connects coupons to persons with Many to Many relationships
    One coupon may be given to many persons. One person may have many coupons

13. **Table bookings by user**
    Connects tables to persons with Many to One relationship
    One table should be booked by only one person. One person may book many tables

14. **Cancelled Online Order**
    Connects online_orders to cancellations with One to One relationship
    One order will have only one cancellation. One cancellation will have only one order

PART D:

Forms:

1. **Online Orders**
   Input: order_item, delivery address, quantity
   Output: Delivery Person Details, Estimated time and date, order_price

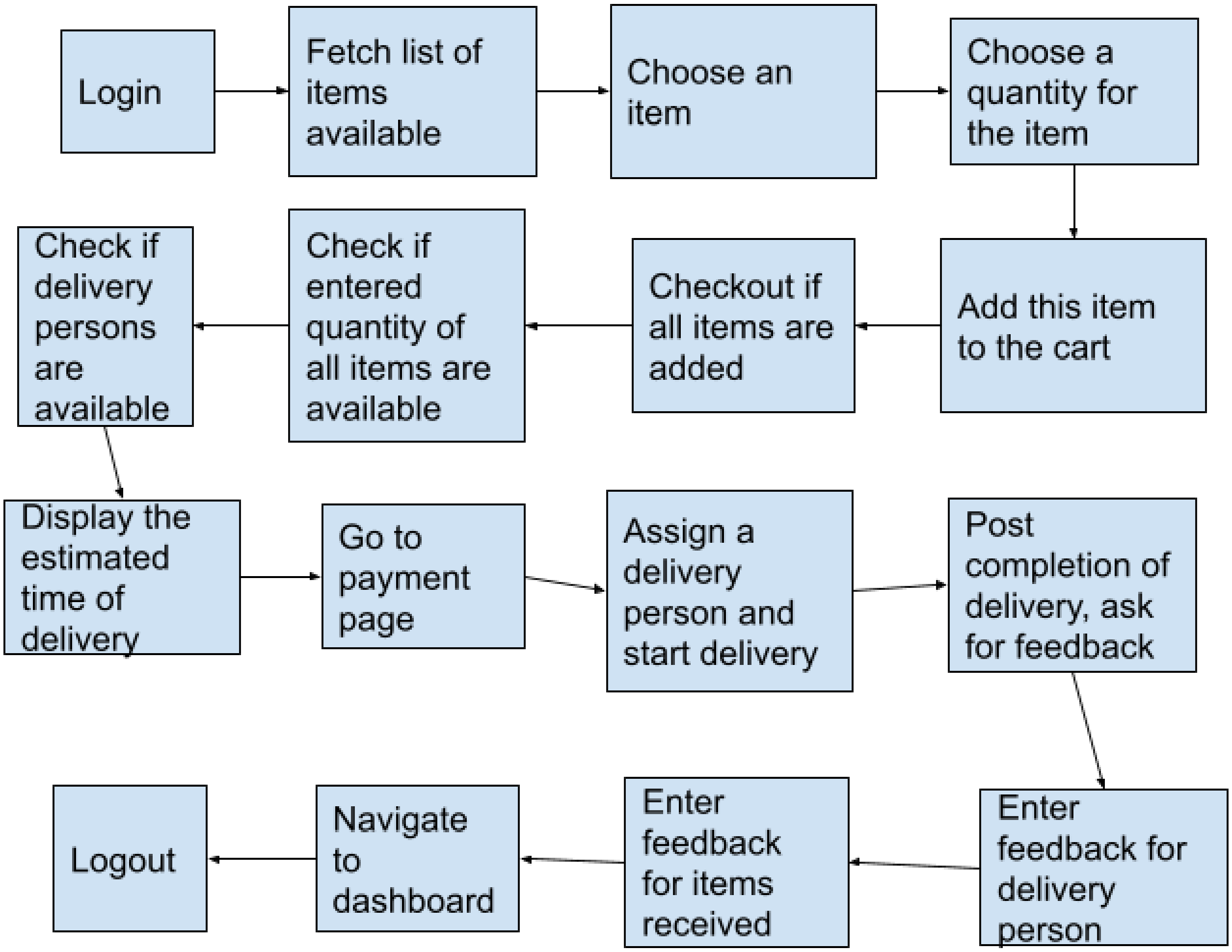
2. **Table Reservation**
   Input: table_type, minimum_capacity, booking_date, booking_from_time, booking_to_time
   Output: table_price, table_id

3. **Order Cancellation**
   Input: on_order_id, cancellation_reason
   Output: Yes if the order is canceled, No if the order cannot be canceled at this stage

4. **Item Feedback Form**
   Input: item_name, feedback_txt, suggestions, rating
   Output: Yes if feedback is submitted, No if any error occurred

5. **Delivery Person Feedback Form**
   Input: dp_name, feedback_txt, suggestions, rating
   Output: Yes if feedback is submitted, No if any error occurred

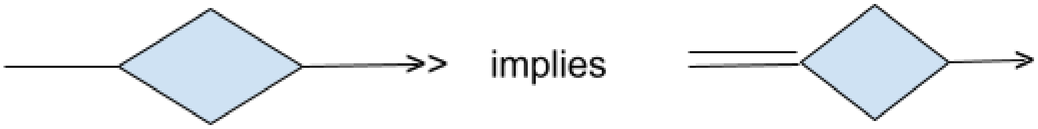
6. **Login Form**
   Input: person_email, password
   Output: Yes if login details are valid, No if details are wrong


Login
Fetch list of items available
Choose an item
Choose a quantity for the item
Add this item to the cart
Checkout if all items are added
Check if entered quantity of all items are available
Check if delivery persons are available
Display the estimated time of delivery
Go to payment page
Assign a delivery person and start delivery
Post completion of delivery, ask for feedback
Enter feedback for delivery person
Enter feedback for items received
Navigate to dashboard
Logout

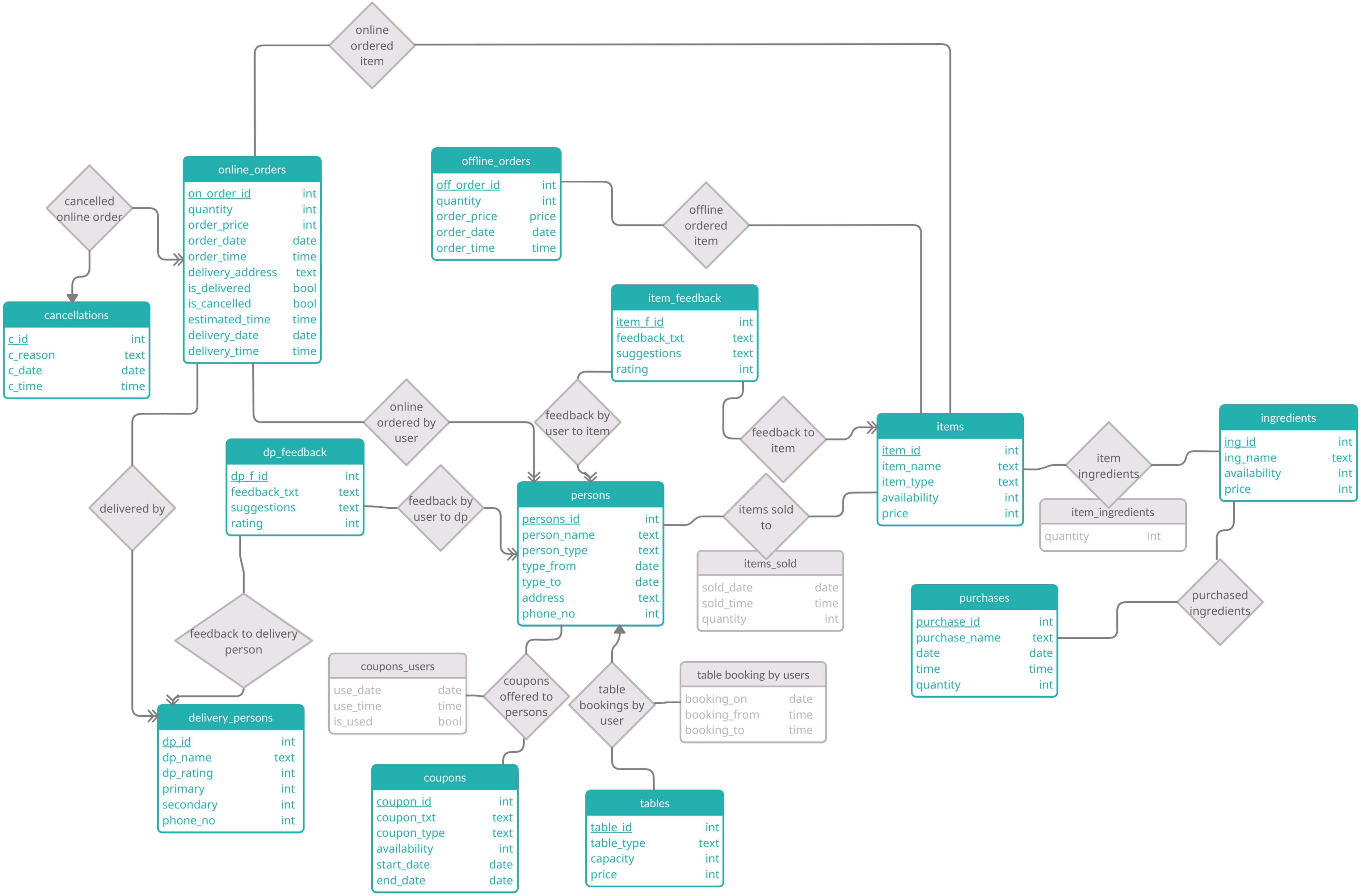
PART E:

Note: Some Indications in the ER diagram will not match the implementations discussed in class. The reason is, the software we used to build ER diagram is a free version and is restricting some shapes in the diagram.

Explanation:



Boxes in Blue color are Entities
Diamond shapes in light brown color are relationships
Boxes in light brown color are attributes for the relationship it got attached



online_orders	
on_order_id	int
quantity	int
order_price	int
order_date	date
order_time	time
delivery_address	text
is_delivered	bool
is_cancelled	bool
estimated_time	time
delivery_date	date
delivery_time	time

offline_orders	
off_order_id	int
quantity	int
order_price	price
order_date	date
order_time	time

cancellations	
c_id	int
c_reason	text
c_date	date
c_time	time

item_feedback	
item_f_id	int
feedback_txt	text
suggestions	text
rating	int

items	
item_id	int
item_name	text
item_type	text
availability	int
price	int

ingredients	
ing_id	int
ing_name	text
availability	int
price	int

item_ingredients	
quantity	int

dp_feedback	
dp_f_id	int
feedback_txt	text
suggestions	text
rating	int

persons	
persons_id	int
person_name	text
person_type	text
type_from	date
type_to	date
address	text
phone_no	int

items_sold	
sold_date	date
sold_time	time
quantity	int

purchases	
purchase_id	int
purchase_name	text
date	date
time	time
quantity	int

coupons_users	
use_date	date
use_time	time
is_used	bool

table booking by users	
booking_on	date
booking_from	time
booking_to	time

delivery_persons	
dp_id	int
dp_name	text
dp_rating	int
primary	int
secondary	int
phone_no	int

coupons	
coupon_id	int
coupon_txt	text
coupon_type	text
availability	int
start_date	date
end_date	date

tables	
table_id	int
table_type	text
capacity	int
price	int