Database Design Document

Group Number: 3

Group Members: Dongliang Guo, Pulkit Saharan, Sanchana Mohankumar, Shivani Shrikant Naik, Yuqi Shen

Business Problem:

Virtual Spatula

The system aims to provide an end-to-end cooking experience of different cuisines for app users including recipe selection, grocery shopping and delivery.

By picking a recipe, purchasing the necessary ingredients, and opting for doorstep delivery, users may effortlessly cook at home.

Mission Statement:

Purpose of the database:

1. Provide a list of recipes with ingredients required for the Virtual Spatula app.
2. Maintain a list of product availability for various shop inventories.
3. Increase delivery efficiency by designating the appropriate delivery partner based on location.

Business Rules:

1. One user can have multiple addresses. One address can be shared by multiple users.
2. A person can be a user, a delivery partner or both.
3. Invoice is generated only if payment is successful.
4. If a delivery partner faces issues during delivery, another delivery partner should be assigned to the order.
5. A non-operational store can be present in the database for future collaborations.

Entities and Design Decisions:

Entity Name	Why Entity Included	How Entity is Related to Other Entities
Person	The Person entity can assist in	Person will have a one to one

	maintaining information about users and delivery partners including their first and last name, contact information, and date of birth etc.	mandatory and identifying relationship with User and a one to one optional and non-identifying relationship with DeliveryPartner through its primary key – PersonID.
User	User stores a UserID, PersonID, and their date of joining. This will help keep record of all users. A user can also be a delivery partner, hence we have separate dates of joining for both.	User will have one to many optional and identifying relationships with User_Recipe and UserAddress through its primary key called UserID. User will also have one to many optional and non-identifying relationships with Order and Payment through its primary key, UserID. In addition, User will be associated with Person with the foregin key called PersonID.
Order	Order maintains all details of an order placed by a user. This will keep record of all information that is necessary for an order. It has attributes like OrderID, UserID, OrderDate, Feedback, OrderAmount, OrderPickupTime etc	Order will have one to many mandatory and identifying relationships with Line_Item and DeliveryPartner_Order through its primary key – OrderID. Order will have a one to one mandatory and non-identifying relationship with Address through its foreign key – ShippingAddressID. Order will have a one to one optional and non-identifying relationship with Invoice through its primary key – OrderID. The relationship is optional as there will be no invoice in case of payment failure. Order will have a many to one optional and non-identifying relationship with User through its foreign key – UserID. Each order should have a matching UserID.
Line_Item	Line_Item is an associative entity created to implement many to many relationships between	Line_Item will be related to Order and Ingredient through its composite primary keys – OrderID

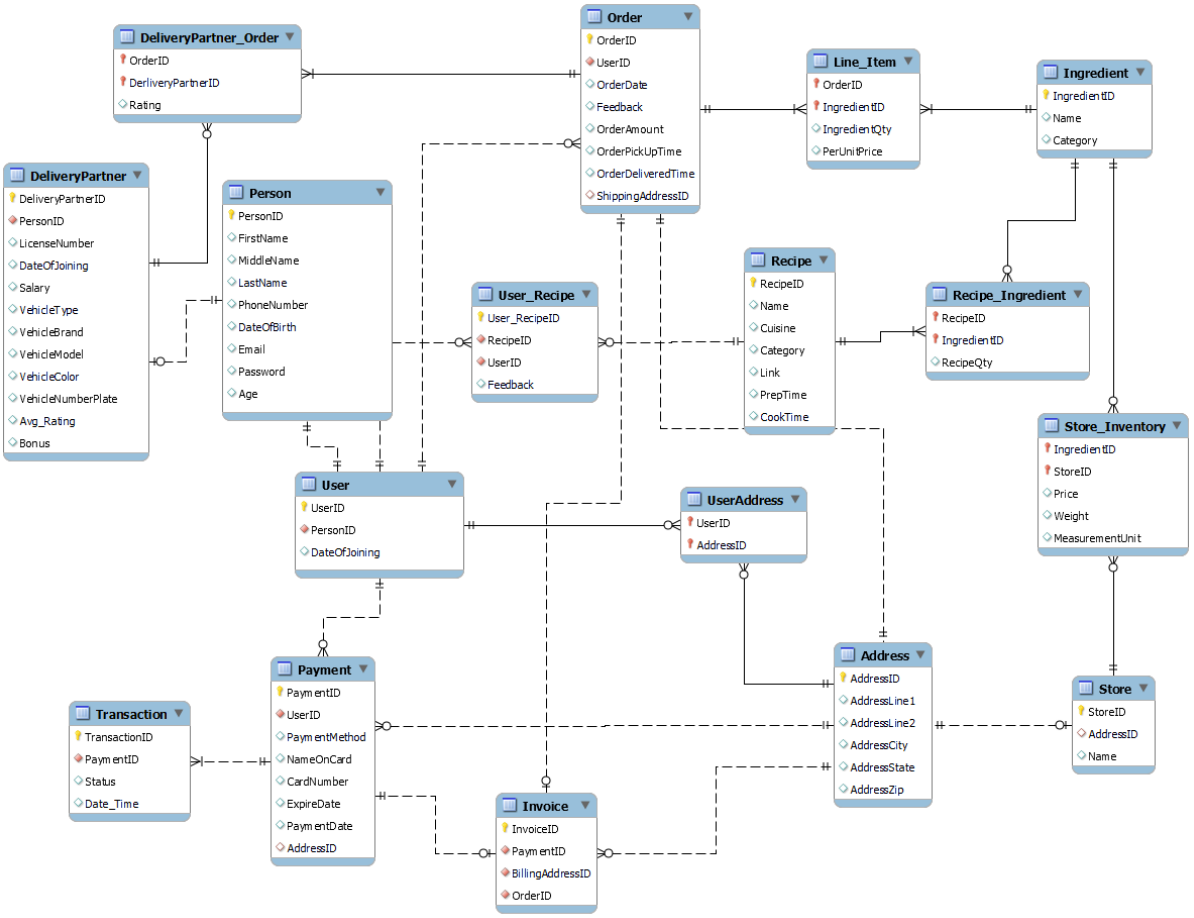
	Order and Ingredient.	and IngredientID. Line_Item will have a many to one mandatory and identifying relationships with both Order and Ingredient. Every record in Line_Item should have a matching record in Order and in Ingredient.
Ingredient	Ingredient maintains the name and category (like dairy, fresh vegetables, meat etc.) details of the items. This will help to keep record of all the necessary ingredients required for the various recipes.	Ingredient will be related to Store_Inventory , Recipe_Ingredient and Line_Item through its primary key , IngredientID. Ingredient will have one to many, mandatory relationship with Line_Item. Ingredient will have one to many optional relationships with Store_Inventory and Recipe_Ingredient. Ingredient will have identifying relationships with Store_Inventory ,Recipe_Ingredient and Line_Item.
Recipe_Ingredient	Recipe_Ingredient is an associative entity created to implement many to many relationship between Recipe and Ingredient. This entity will have composite primary key composed of primary key of Ingredient and Recipe. This will have recipe quantity to maintain the quantity of ingredient required for each recipe.	Recipe_Ingredient will be related to Ingredient and Recipe through its composite primary key (RecipeID and IngredientID). Recipe_Ingredient will have many to one mandatory relationship with Recipe and Ingredient. Every record in Recipe_Ingredient should have a matching record in Recipe and Ingredient.
Recipe	Recipe maintains all the details about the preparation of a meal. It has attributes about the name, type of cuisine, category , hyperlink to recipe, preparation time and cooking time.	Recipe will be related to Recipe_Ingredient having one to many mandatory and identifying relationship. Recipe will also be related to User_Recipe having a one to many optional and identifying relationship.
User_Recipe	User_Recipe is an associative entity created to implement many to many relationship between Recipe and User. This entity will have a composite primary key composed of primary keys of User and Recipe. This will have a feedback attribute as well to maintain feedback from the user	User_Recipe will be related to User and Recipe through its composite primary key (RecipeID and UserID). User_Recipe will have many to one mandatory relationship with User and Recipe. Every record in User_Recipe should have a matching record in Recipe and User.

	for each recipe.	
DeliveryPartner	Delivery Partner contains details to track a delivery partner when order is placed like license number and other vehicle details. It also contains common information like the joining date, Bonus, Avg_Rating and salary of the Delivery Partner. A delivery partner can also be a user of the platform, hence we have separate dates of joining for both.	Delivery Partner Entity is related to Person Entity in order to maintain personal details of a delivery guy like name, phone number and Date of birth. Its relationship with Person is one to one mandatory, non- identifying relationship. It is also related to Order Entity through an associative entity called DeliveryPartner_Order entity. DeliveryPartner is related to the DeliveryPartner_Order Entity with a one to many optional relationship.
DeliveryPartner_Order	DeliveryPartner_Order is an Associative Entity to implement many to many relationship between DeliveryPartner and Order entities. A Delivery Partner can handle multiple orders and an order can be handled by 2 delivery partners in situations where a delivery partner has issues with his vehicle.	DeliveryPartner_Order is related to DeliveryPartner where its relationship is many to one mandatory and identifying relationship. They are both related through the composite primary key of DeliveryPartner_Order called DeliveryPartnerID. Delivery Partner Order is related to Order where its relationship is many to one mandatory and identifying relationship. They are both linked via the composite primary key of Order Entity called OrderID.
Payment	Payment is related to User Entity as each user has to go through the payment process in order to place the order and has attributes like card details, AddressID and Transaction details. Payment is related to a transaction entity which has attributes like status and Date and time.	Payment Entity is related to the User as many to one mandatory relationship using the Primary key of UserID. Payment Entity is related to Transaction with a one to many mandatory relationship using the primary key of PaymentID. Payment Entity is related to Invoice with a one-to-one optional relationship using the PaymentID. Payment Entity is related to address Entity with many to one

Transaction	Transaction entity maintains status of Payment as each payment undergoes a transaction where it has card or 3rd party payment options in order to complete the payment. Transaction Entity has attributes like Status and Date and time.	Transaction is related to Payment with a many to one mandatory, non-identifying relationship as there are many transaction methods that can be used for a payment to be made if there is any transaction failure. Transaction is related to Payment Entity using the foreign key PaymentID.
Invoice	Invoice entity maintains the invoice data for every order placed by an user. This will help in generating invoice for the users. It has attributes like InvoiceID, PaymentID, BillingAddressID, OrderID.	Invoice has a mandatory one-to-one relationship with Order, linked with OrderID to access the order details. Invoice also has a mandatory many-to-one relationship with Address for billing address, linked with the foreign key BillingAddressID. Invoice is linked to Payment with a one-to-one mandatory relationship using the foreign key PaymentID.
Address	Address maintains addresses of customers and stores. Every address will have attributes like Line1, Line2, City, State, Zip etc. These addresses will be utilised to successfully deliver items from stores to users.	Address is related to the User_Address associative entity. A user can have multiple addresses. The relationship is one-to-many and optional, identifying because we are also storing store addresses in Address entity. It will be linked to User_Address through its primary key, AddressID. Address is also related to Store entity with a one-to-one optional and non-identifying relationship. It will be linked through its primary key, AddressID. Address entity also has a one-to-many optional, non-identifying relationship with Invoice as an address can be used as billing address in multiple invoices. It will be linked through its primary key, AddressID. Address is also linked to Order using primary key AddressID with a one-to-one mandatory relationship (shipping address).

User_Address	User_Address is the associative entity that will store the many-many relationship present between User and Address entities. A user can have multiple addresses and an address can be shared by multiple users, this is captured by the User_Address entity.	User_Address is related to User with a many-to-one mandatory, identifying relationship. Every entry in User_Address has to have a matching entry in User entity. It is linked via the primary key attribute UserID. User_Address is also related to Address with a many-to-one mandatory, identifying relationship. It is linked using the primary key attribute AddressID.
Store	Store entity will maintain a list of all stores available to our platform. Every store will have attributes like StoreID, AddressID, Name.	Store has a one-to-many optional and identifying relationship with StoreInventory. The relationship is optional because we can have stores that have not yet registered their inventory with the platform. It is related using the primary key StoreID. Store also has a one-to-one mandatory and non-identifying relationship with Address, linked via the foreign key AddressID.
StoreInventory	StoreInventory maintains a record of all items available in a store. It has attributes like StoreID, IngredientID, Price, Weight, MeasurementUnit. This entity will help in keeping track of product availability across stores and will facilitate efficient delivery.	StoreInventory is related to Store entity with a mandatory, identifying many-to-one relationship. Every entry in the StoreInventory entity needs to have a matching entry in the Store entity. It is linked using the primary key attribute StoreID. StoreInventory is also linked to Ingredient entity with a mandatory, identifying many-to-one relationship. It is linked using the primary key attribute IngredientID.

ER Diagram:


DeliveryPartner_Order
OrderID
DerliveryPartnerID
Rating
Order
OrderID
UserID
OrderDate
Feedback
OrderAmount
OrderPickUpTime
OrderDeliveredTime
ShippingAddressID
Line_Item
OrderID
IngredientID
IngredientQty
PerUnitPrice
Ingredient
IngredientID
Name
Category
DeliveryPartner
DeliveryPartnerID
PersonID
LicenseNumber
DateOfJoining
Salary
VehicleType
VehicleBrand
VehicleModel
VehicleColor
VehicleNumberPlate
Avg_Rating
Bonus
Person
PersonID
FirstName
MiddleName
LastName
PhoneNumber
DateOfBirth
Email
Password
Age
User_Recipe
User_RecipeID
RecipeID
UserID
Feedback
Recipe
RecipeID
Name
Cuisine
Category
Link
PrepTime
CookTime
Recipe_Ingredient
RecipeID
IngredientID
RecipeQty
Store_Inventory
IngredientID
StoreID
Price
Weight
MeasurementUnit
User
UserID
PersonID
DateOfJoining
UserAddress
UserID
AddressID
Payment
PaymentID
UserID
PaymentMethod
NameOnCard
CardNumber
ExpireDate
PaymentDate
AddressID
Address
AddressID
AddressLine1
AddressLine2
AddressCity
AddressState
AddressZip
Store
StoreID
AddressID
Name
Transaction
TransactionID
PaymentID
Status
Date_Time
Invoice
InvoiceID
PaymentID
BillingAddressID
OrderID