**MIS 6308**

**Fall 2017**

**Assignment 3**

**Question 1 (10 points)**

The current Doordash system does not allow a customer to place menu items from multiple restaurants in the same order. Clearly, such a feature can be very useful to customers. On the other hand, Doordash does not want to allow the user to provide complete freedom in choosing restaurants. For example, it may not be too difficult to assemble items from restaurants that are near to each other or those that are on the way to the delivery address.

Assume we want to design a new process that will help with ordering items from multiple restaurants in the same order. However, the process should only allow the user to choose from restaurants that are “convenient” from DoorDash’s perspective. We would like to model this process as the use case ‘Order from Multiple Restaurants’ in our model.

1. Develop a use case description for it. Identify the actor and trigger. Underline data used within this use case.

**Answer 1:**

|  |  |
| --- | --- |
| Use Case Name: | Order from Multiple Restaurants |
| Primary Actor: | User |
| Brief Description: | User wants to order food from multiple restaurants via Doordash.com |
| Stakeholders: | User: Wants to order food from multiple restaurants Dasher: Offers food delivery to user |
| Trigger: | User requests to order food from multiple restaurants |
| Normal flow of events: | 1. Login/ sign up using user details 2. Receive order request with restaurant type 3. Send location details to google maps 4. Receive nearby restaurant details 5. Send restaurant details to user that are convenient (within 1 mile or on the way to destination) to Doordash.com 6. Receive order confirmation for multiple restaurants and payment details from user 7. Send order details to multiple restaurants 8. Receive order confirmation from all restaurants 9. Send ride request to dasher to collect food from multiple restaurants 10. Receive dasher confirmation 11. Send dasher details to customer 12. Receive feedback for service |
| Alternate/Exception flow: | 1.a1. If the login fails, redo this step  6.a1. If user cancels the order, go back to place order  12.a1. If dasher cancels the request, go back to send ride request to dasher |

1. Document the data used in this use case using the data dictionary notation.

**Answer 2:**

Data Dictionary:

Use Case 2: Order from Multiple Restaurants

User Details= First Name + Last Name + {Phone No} + Email Id + Password + User Country +User Zip code + Payment Details

Payment Details =Order Amount + Card Details

Card Details=Card Number + CVV + Expiry Month + Expiry Year

Restaurant type = {[Chinese| Indian | Pizzeria]}

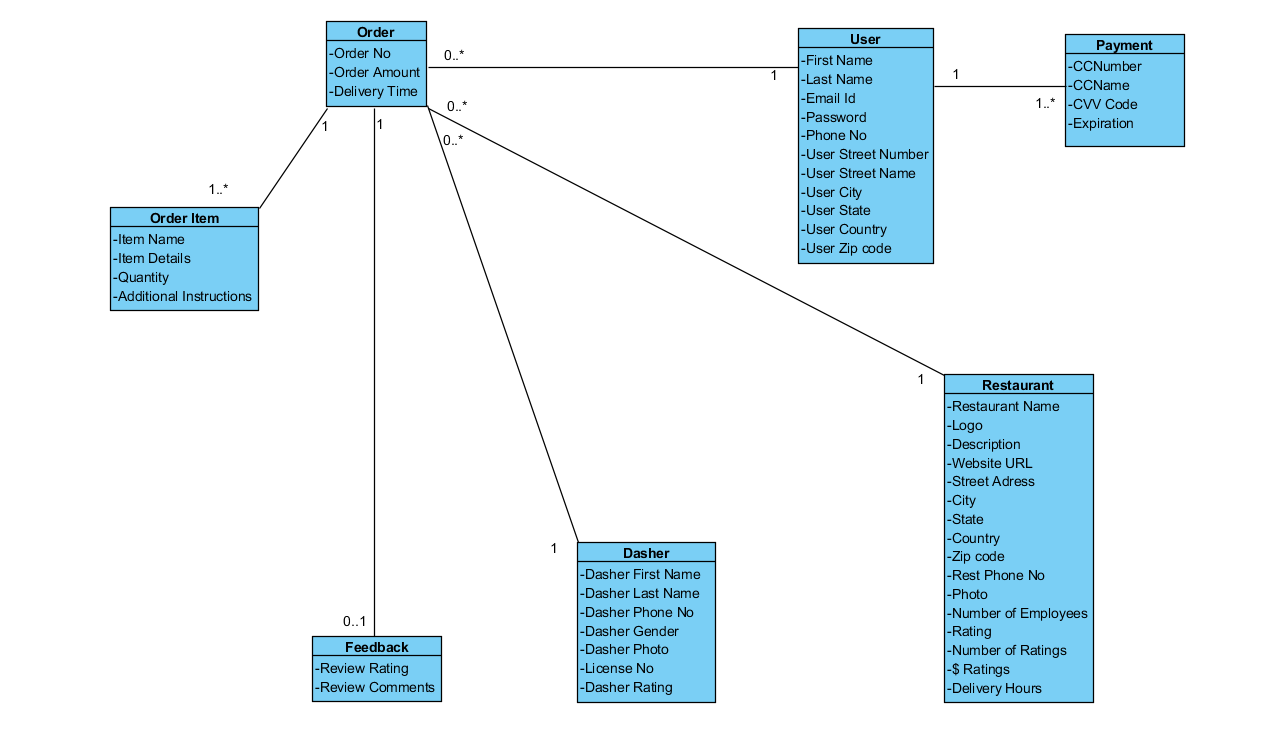
Restaurant Details = {Restaurant Name + Description + Logo + Website URL + {Phone No} + Photo + Street Address + City + State + Country + Zip code + Rating + Number of Ratings + $ Ratings + Delivery Hours + Number of Employees}

Dasher Details = Dasher First Name + Dasher Last Name + {Dasher Phone No} + Dasher Gender + Dasher Photo + License No + Dasher Rating

Order Details = Order No+ {RestaurantOrderNo + No of Items + Item Name + Restaurant Details}

**Question 2 (10 points)**

Consider the following class diagram.



1. Identify the data elements that Doordash will store in a database as persistent data.

**Answer 1:**

All given data need to be stored in the database

1. For the persistent data you identified in step 1, design a database in III Normal Form.

**Answer 2:**

Primary key- **Bold**

Foreign key -*Italics*

User (**UserId**, First Name, Last Name, Email Id, Password, Phone No, User Street Number, User Street Name, User City, User State, User Country, User Zip code,)

Restaurant (**RestaurantId**, Restaurant Name, Logo, Description, Website URL, Street Address, City, State, Country, Zip code, Rest Phone No, Photo, Delivery Hours,)

Dasher (**DasherId**, Dasher First Name, Dasher Last Name, Dasher Phone No, Dasher Gender, Dasher Photo, License No)

Payment (**PaymentId**, CCNumber, CCName, CVV Code, Expiration, *UserId*)

Order(**OrderNo**, OrderAmount, DeliveryTime, *UserID*, *RestaurantID*, *DasherId*)

OrderItem(***OrderNo*, ItemName**, ItemDetails, Quantity, Additional Instructions)

Feedback(**Feedback#**, Review Rating, Review Comments, *OrderNo*)

1. Specify the primary key, foreign key(s), and constraints for each table in your design.

**Answer 3:**

PK- Primary Key

FK-Foreign Key

**User**

UserId- PK, UNIQUE, NOT NULL

**Restaurant**

RestaurantId- PK, UNIQUE, NOT NULL

**Dasher**

DasherId- PK, UNIQUE, NOT NULL

**Payment**

PaymentId- PK, UNIQUE, NOT NULL

UserId- FK, NOT NULL, should exist in user table

**Order**

OrderNo - PK, UNIQUE, NOT NULL

UserId- FK, NOT NULL, should exist in user table

RestaurantId- FK, NOT NULL, should exist in restaurant table

dasherId- FK, NOT NULL, should exist in dasher table

**OrderItem**

OrderNo - PK, UNIQUE, NOT NULL

ItemName - PK, UNIQUE, NOT NULL

Order No - should exist in order table

**Feedback**

Feedback# - PK, UNIQUE, NOT NULL

Order No - FK, NOT NULL, should exist in order table