MIS 6308: System Analysis and Project Management

Assignment 3

1. The description of the use case is as follows:

Use Case Name	Order food from multiple restaurants for same order
Use Case ID	3
Primary Actors	User, Google Maps, Restaurant, Dasher
Stakeholders & Interests	User – wants to order food from different places for same order Restaurant – wants to prepare food to its customers and give it to delivery person on time Dasher – wants to deliver food from restaurants to customer on time
Brief Description	This use case describes the process of ordering food from multiple restaurants for the same order on DoorDash site
Trigger	User goes to more than one restaurants available on DoorDash site for ordering food for the same order
Trigger Type	External Trigger
Relationships	
Association	User, Google Maps, Restaurant, Dasher
Include	Registration/Sign In
Extend	N/A
Generalization	N/A
Normal Flow of Events	

1. The user comes to DoorDash, searches for restaurant within his/her area.
2. Google Maps receives entered area data and executes *“Search Restaurant”* use case.
3. The user finalizes order details from a restaurant and goes to home page where convenient restaurants are displayed.
4. Google Maps receives last selected restaurant’s location and updated route based on restaurants selected till now, executes *“Search Restaurant”* use case and sends restaurant options back to DoorDash system.
5. Convenient restaurants are displayed based on constraint of proximity of restaurants (falling within 1.5-mile radius of each of selected restaurants) or located on updated route to user out of the restaurant options sent by Google Maps. (Route to user gets updated based on addition of restaurants for pickup)
6. Step 3 to 5 are repeated for restaurants the user wants to add (until max. number of restaurants is reached for one order i.e., assumed value is 5)
7. User enters payment details and executes the *“Send Order & Payment”* use case.
8. The restaurants receive order and payment details, prepare food for the order and executes *“Send Order Pickup Request”* use case.

9. The dasher receives the order pickup request, a. Accepts it, confirmation is sent to restaurants. b. Declines it, order pickup request is sent to another dasher and step 9 is repeated till pickup is accepted by a dasher. 10. The restaurants receive confirmed order pickup response from dasher and executes *“Send Order Confirmation”* use case.
Sub Flows
Alternate/Exception Flows
1a – Selects restaurant from the displayed options of restaurants based on previously entered area data and executes the *“Display Restaurant Menu*” use case. 1b – Goes to register/sign in and executes the *“Register/Sign In”* use case.

This use case reuses every data dictionary component present in *“Order Food”* use case.

However, there is one update in the data dictionary where *“Order Details”* would contain details of order from a restaurant along with the restaurant details.
Each order will have at least 1 and up to maximum of 5 *“Order Details”*.

The elements of data dictionary are:

1. Aggregate Data:
   a. Delivery Area = [Street Address | Zip code]
   b. Street Address = Address Line 1 + (City Name) + (State Name) + (Country Name) + (Zip Code)
   c. Restaurant Options = 0{Cuisine Type + 0{Restaurant Details}}
   d. Restaurant Details = Restaurant Name + Restaurant Description + Restaurant Logo + Street Address + Restaurant Website URL + 1{Restaurant Phone Number} + Restaurant Number of Employees + Restaurant Rating + Restaurant Delivery Hours + Restaurant $ Rating
   e. Order Details = Restaurant Name + Restaurant Address + 1{Restaurant Phone Number} + 1{Item}
   f. Order = Order ID + 1{Order Details}5 + Payment Details
   g. Item = {Name + Quantity + (Customizations) + (Additional Instructions) + Issue Instruction}
   h. Payment Details = Payment ID + Card Details + (Promo Code) + Total Amount
   i. Card Details = Card Owner Name + Card Number + Expiry Information + CVV Code + Card Zip Code + (Pin Code)
   j. Total Amount = Sub-total + Taxes + Delivery Fee + Service Fee + (Small Order Fee)
   k. Pickup Request = Request ID + Order + User Address + Delivery Time
   l. Pickup Response = Response + (Reason for response)

2. Data Elements:
    a. Item Name
    b. Item Quantity
    c. Customization
    d. Additional Instructions
    e. Delivery Time
    f. Promo Code
    g. Card Owner Name
    h. Card Number
    i. Expiry Information
    j. CVV Code
    k. Card Zip Code
    l. Sub-total
    m. Taxes
    n. Delivery Fee
    o. Service Fee
    p. Small Order Fee
    q. Response
    r. Reason for response
    s. Zip code
    t. Address Line 1
    u. City Name
    v. State Name
    w. Country Name
    x. Cuisine Type
    y. Restaurant Name
    z. Restaurant Description
    aa. Restaurant Logo
    bb. Restaurant Website URL
    cc. Restaurant Phone Number
    dd. Restaurant Number of Employees
    ee. Restaurant Rating
    ff. Restaurant Delivery Hours
    gg. Restaurant $ Rating
    hh. Order ID
    ii. Request ID
    jj. Payment ID

2. Considering the given class diagram, the answers are as follows:
    2.1. Data elements mentioned below will be stored in DoorDash database:
        i) For *“Order”* class
            (1) Order No
            (2) Order Amount
            (3) Delivery Address
        ii) For *“Order Item”* class
            (1) Item Name
            (2) Item Details
            (3) Quantity
            (4) Additional Instructions
        iii) For *“Feedback”* class
            (1) Review Rating
            (2) Review Comments
        iv) For *“Dasher”* class
            (1) First Name
            (2) Last Name
            (3) Phone No.
            (4) Gender
            (5) License No.
            (6) Dasher Rating
        v) For *“User”* class
            (1) First Name
            (2) Last Name
            (3) Email ID
            (4) Password
            (5) Phone No.
            (6) User Street Number
            (7) User City
            (8) User State
            (9) User Country
            (10) User Zip Code
        vi) For *“Restaurant”* class
            (1) Restaurant Name
            (2) Description
            (3) Website URL
            (4) Street Address
            (5) City
            (6) State
            (7) Country
            (8) Zip Code
            (9) Phone No
            (10) Number of Employees
            (11) Rating
            (12) Number of Ratings
            (13) $ Ratings
            (14) Delivery Hours

vii) For *“Payment”* class
   (1) CCNumber
   (2) CCName
   (3) CVV Code
   (4) Expiration

Photos and logo of *“Restaurant”* class will be stored along with other files for front-end. Similarly, for “Dasher” class, photo of dasher will be stored with files for front-end.

2.2. The database design for above persistent data is as follows:
"Bold text" means Primary Key
"Italic text" means Foreign Key

User (**UserID**, EmailID, Password)

UserDetails (**UserDetailID**, *UserID*, FirstName, LastName, StreetNumber, StreetName, City, State, Country, Zip Code, PhoneNo)

Restaurant (**RestaurantID**, Name, Description, SiteURL, TotalEmployees, Rating, NoOfRatings, CostRating, DeliveryStartHour, DeliveryEndHour, StreetAddress, City, State, Country, ZipCode)

RestaurantContactDetails (*RestaurantID*, PhoneNo)

Order (**OrderID**, OrderAmount, DeliveryTime, ReviewRating, ReviewComments, *LicenseNo, UserDetailID, RestaurantID, PaymentID*)

Item (**ItemID**, *RestaurantID*, Name, Details)

OrderItem (*OrderID, ItemID*, Quantity, AdditionalInstructions)

Payment (**PaymentID**, *UserID*, CCNumber, CCName, CVVCode, Expiration)

Dasher (**LicenseNo**, FirstName, LastName, Gender, PhoneNo, Rating)

- User class is separated into two tables here as a user could have multiple addresses and a different contact number for each address.
- Restaurant class is separated into two tables as restaurant chains within same city can have different addresses and multiple contact numbers for same restaurant.
- Order table has UserDetailID to fetch user address, phone number and name of the user for each order.
- Order table has RestaurantID to fetch restaurant details for dasher and user to contact.
- Order table has LicenseNo to fetch details of Dasher and PaymentID to fetch details of payment mode.
- Order Item class is separated into two tables different orders can have same items and also, there can be items which are not yet ordered from restaurant menus.
- LicenseNo is assumed here to be a unique number and is used to identify dashers.

2.3. Constraints for various tables mentioned in database design will be as follows:

- User (**<u>UserID</u>**, EmailID, Password)
  - (a) Primary Key – UserID
  - (b) Foreign Key – N/A
  - (c) Unique Fields – EmailID
  - (d) Not Null Fields – EmailID, Password

- UserDetails (**<u>UserDetailID</u>**, *UserID*, FirstName, LastName, StreetNumber, StreetName, City, State, Country, Zip Code, PhoneNo)
  - (a) Primary Key – UserDetailID
  - (b) Foreign Key – UserID from User table
  - (c) Unique Fields – N/A
  - (d) Not Null Fields – PhoneNo, FirstName, StreetName, City, State, Country, Zip Code, UserID

- Restaurant (**<u>RestaurantID</u>**, Name, Description, SiteURL, TotalEmployees, Rating, NoOfRatings, CostRating, DeliveryStartHour, DeliveryEndHour, StreetAddress, City, State, Country, ZipCode)
  - (a) Primary Key – RestaurantID
  - (b) Foreign Key – N/A
  - (c) Unique Fields – N/A
  - (d) Not Null Fields – Name, TotalEmployees, DeliveryStartHour, DeliveryEndHour, StreetAddress, City, State, Country, ZipCode
  - (e) Additional Constraints - CostRating > 0 or NULL, Rating should be between 0 and 5 (inclusive) if NOT NULL

- RestaurantContactDetails (*RestaurantID*, PhoneNo)
  - (a) Primary Key – N/A
  - (b) Foreign Key – RestaurantID
  - (c) Unique Fields – N/A
  - (d) Not Null Fields – RestaurantID, PhoneNo

- Order (**<u>OrderID</u>**, OrderAmount, DeliveryTime, ReviewRating, ReviewComments, *LicenseNo, UserDetailID, RestaurantID, PaymentID*)
  - (a) Primary Key – OrderID
  - (b) Foreign Key – LicenseNo from Dasher, UserDetailID from UserDetails, RestaurantID from Restaurant, PaymentID from Payment table
  - (c) Unique Fields – N/A
  - (d) Not Null Fields – OrderAmount, DeliveryTime, LicenseNo, UserDetailID, RestaurantID, PaymentID

- Item (**<u>ItemID</u>**, *RestaurantID*, Name, Details)
  - (a) Primary Key – ItemID
  - (b) Foreign Key – RestaurantID from Restaurant
  - (c) Unique Fields – N/A
  - (d) Not Null Fields – Name, RestaurantID

- OrderItem (*OrderID, ItemID*, Quantity, AdditionalInstructions)
    (a) Primary Key – N/A
    (b) Foreign Key – OrderID from Order table, ItemID from Item table
    (c) Unique Fields – N/A
    (d) Not Null Fields – Quantity, OrderID, ItemID

- Payment (<u>**PaymentID**</u>, *UserID*, CCNumber, CCName, CVVCode, Expiration)
    (a) Primary Key – PaymentID
    (b) Foreign Key – UserID from User table
    (c) Unique Fields – CCNumber
    (d) Not Null Fields – CCNumber, CCName, CVVCode, Expiration, UserID

- Dasher (<u>**LicenseNo**</u>, FirstName, LastName, Gender, PhoneNo, Rating)
    (a) Primary Key – LicenseNo
    (b) Foreign Key – N/A
    (c) Unique Fields – N/A
    (d) Not Null Fields – FirstName, LastName, Gender, PhoneNo
    (e) Additional Constraints – Rating should be between 0 and 5 (inclusive) if NOT NULL