COMM 335 Case Dev

Section 201, Group 13

95970, 37531, 74542, 64813, 95972

Case Narrative

UberEATS is an online platform to order food from a wide range of restaurants to be delivered to their customers. A robust database is required to keep track of the large quantity and variety of deliveries.

Account Management

To login or register, customers are required to provide an email address and a phone number. The registration process also requires a name and a language of communication.

Customers can choose to save up to two delivery addresses in the app and must have at least one primary payment option to use the UberEATS platform, which can be a credit card or debit card. Multiple gift cards can also be added to the payment options in one account.

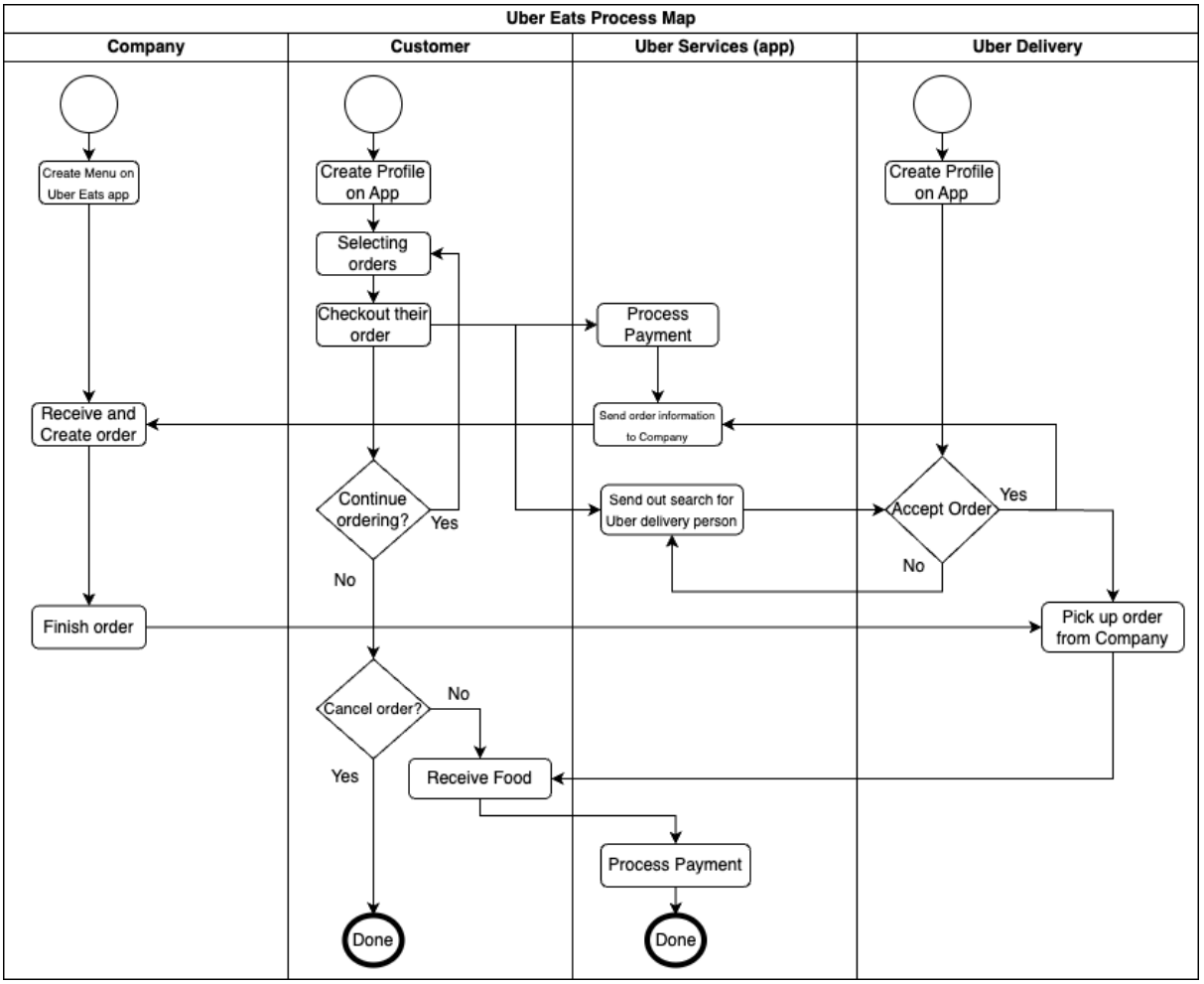
Ordering

Customers can select menu items with their preferences for customizable items to an order. One order can contain multiple menu items from the same restaurant but not from multiple restaurants. If someone would like to order from multiple restaurants, they can place another order from a different restaurant while their current order is being processed.

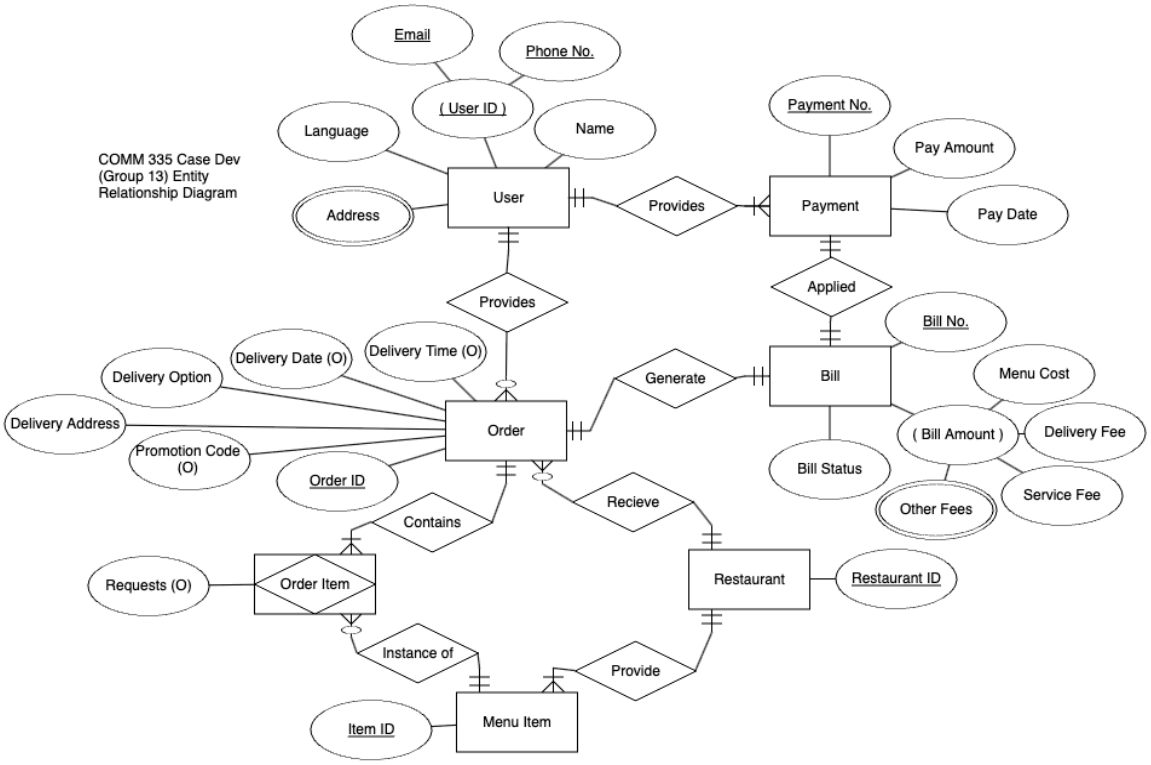
To place the order, they must provide a delivery address, delivery option, and payment method, and can add a promotion code if desired. There are two delivery options: “Deliver as soon as possible” or “Scheduled,” with the latter option requiring a date and time for delivery. Special delivery instructions can be included if so desired.

Generally, the total cost of an UberEATS order will include the ordered menu cost, a delivery fee, a service fee, and other applicable fees, such as a busy area fee and priority fee. All these will be specified in a customer receipt, which may also include discounts from promotions. The receipts of all past orders for a specific user will always be accessible.

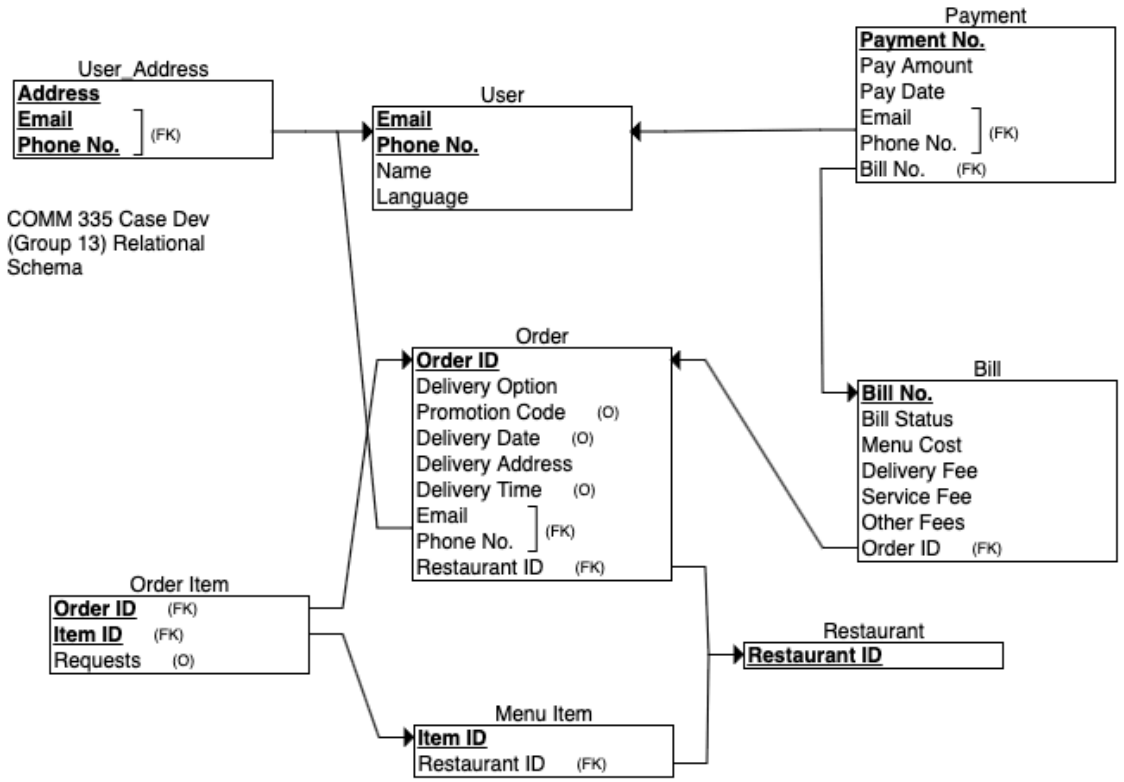
Business Process Map



Entity Relationship Diagram



Relational Schema



SQL Code

```
CREATE TABLE User
(
  Name VARCHAR(100) NOT NULL,
  Language_ 100 NOT NULL,
  Email VARCHAR(400) NOT NULL,
  Phone_No. VARCHAR(20) NOT NULL,
  PRIMARY KEY (Email, Phone_No.)
);


CREATE TABLE Restaurant
(
  Restaurant_ID VARCHAR(10) NOT NULL,
  PRIMARY KEY (Restaurant_ID)
);


CREATE TABLE Menu_Item
(
  Item_ID VARCHAR(10) NOT NULL,
  Restaurant_ID VARCHAR(10) NOT NULL,
  PRIMARY KEY (Item_ID),
  FOREIGN KEY (Restaurant_ID) REFERENCES Restaurant(Restaurant_ID)
);


CREATE TABLE User_Address
(
  Address VARCHAR(200) NOT NULL,
  Email VARCHAR(400) NOT NULL,
  Phone_No. VARCHAR(20) NOT NULL,
  PRIMARY KEY (Address, Email, Phone_No.),
  FOREIGN KEY (Email, Phone_No.) REFERENCES User(Email, Phone_No.)
);


CREATE TABLE Order
(
  Delivery_Option 30 NOT NULL,
  Promotion_Code VARCHAR(15),
  Delivery_Date DATE,
  Delivery_Address VARCHAR(200) NOT NULL,
  Delivery_Time 7,
  Order_ID VARCHAR(10) NOT NULL,
  Email VARCHAR(400) NOT NULL,
  Phone_No. VARCHAR(20) NOT NULL,
  Restaurant_ID VARCHAR(10) NOT NULL,
  PRIMARY KEY (Order_ID),
  FOREIGN KEY (Email, Phone_No.) REFERENCES User(Email, Phone_No.),
```

```
  FOREIGN KEY (Restaurant_ID) REFERENCES Restaurant(Restaurant_ID)
);


CREATE TABLE Bill
(
  Bill_Status 100 NOT NULL,
  Menu_Cost NUMERIC(10, 2) NOT NULL,
  Delivery_Fee NUMERIC(10, 2) NOT NULL,
  Service_Fee NUMERIC(10, 2) NOT NULL,
  Other_Fees NUMERIC(10, 2) NOT NULL,
  Bill_No. INT NOT NULL,
  Order_ID VARCHAR(10) NOT NULL,
  PRIMARY KEY (Bill_No.),
  FOREIGN KEY (Order_ID) REFERENCES Order(Order_ID)
);


CREATE TABLE Order_Item
(
  Requests VARCHAR(1000),
  Order_ID VARCHAR(10) NOT NULL,
  Item_ID VARCHAR(10) NOT NULL,
  PRIMARY KEY (Order_ID, Item_ID),
  FOREIGN KEY (Order_ID) REFERENCES Order(Order_ID),
  FOREIGN KEY (Item_ID) REFERENCES Menu_Item(Item_ID)
);


CREATE TABLE Payment
(
  Payment_No. INT NOT NULL,
  Pay_Amount NUMERIC(10, 2) NOT NULL,
  Pay_Date DATE NOT NULL,
  Email VARCHAR(400) NOT NULL,
  Phone_No. VARCHAR(20) NOT NULL,
  Bill_No. INT NOT NULL,
  PRIMARY KEY (Payment_No.),
  FOREIGN KEY (Email, Phone_No.) REFERENCES User(Email, Phone_No.),
  FOREIGN KEY (Bill_No.) REFERENCES Bill(Bill_No.)
);
```