Case

Browns Crafthouse

Browns Crafthouse is a locally focused, casual restaurant that redefines the dining experience from when patrons enter to the moment they leave. Browns Crafthouse has approached you to help them plan and envision a business process that improves their front-house management.

Reservations

To accept a reservation, Browns Crafthouse requires customers to input information on party size, date and time, and name as a reservation ID.

Service

Upon entering Browns Crafthouse, customers are greeted by a host, who checks for reservations and guides them to their chosen seats. Once seated, guests are presented with their menus from their assigned server. Customers are then free to browse the menu at their own pace and, once ready, make one or multiple orders to their assigned server.

Order Preparation

As selections are made, servers will generate a ticket with information such as the item quantity, item price, and specific instructions, which would be sent to the kitchen where talented chefs begin preparing the meals and drinks for the patrons. It is important to note that a ticket may contain one or many menu items that reflect the current menu selection.

Payment

After the completion of the prime business process, patrons have the choice to repeat the process by ordering more dishes or choose to conclude the dining experience, at which patrons may request their bill from the assigned server. Payment methods are versatile, offering the utmost convenience with payments in Cash or Card.

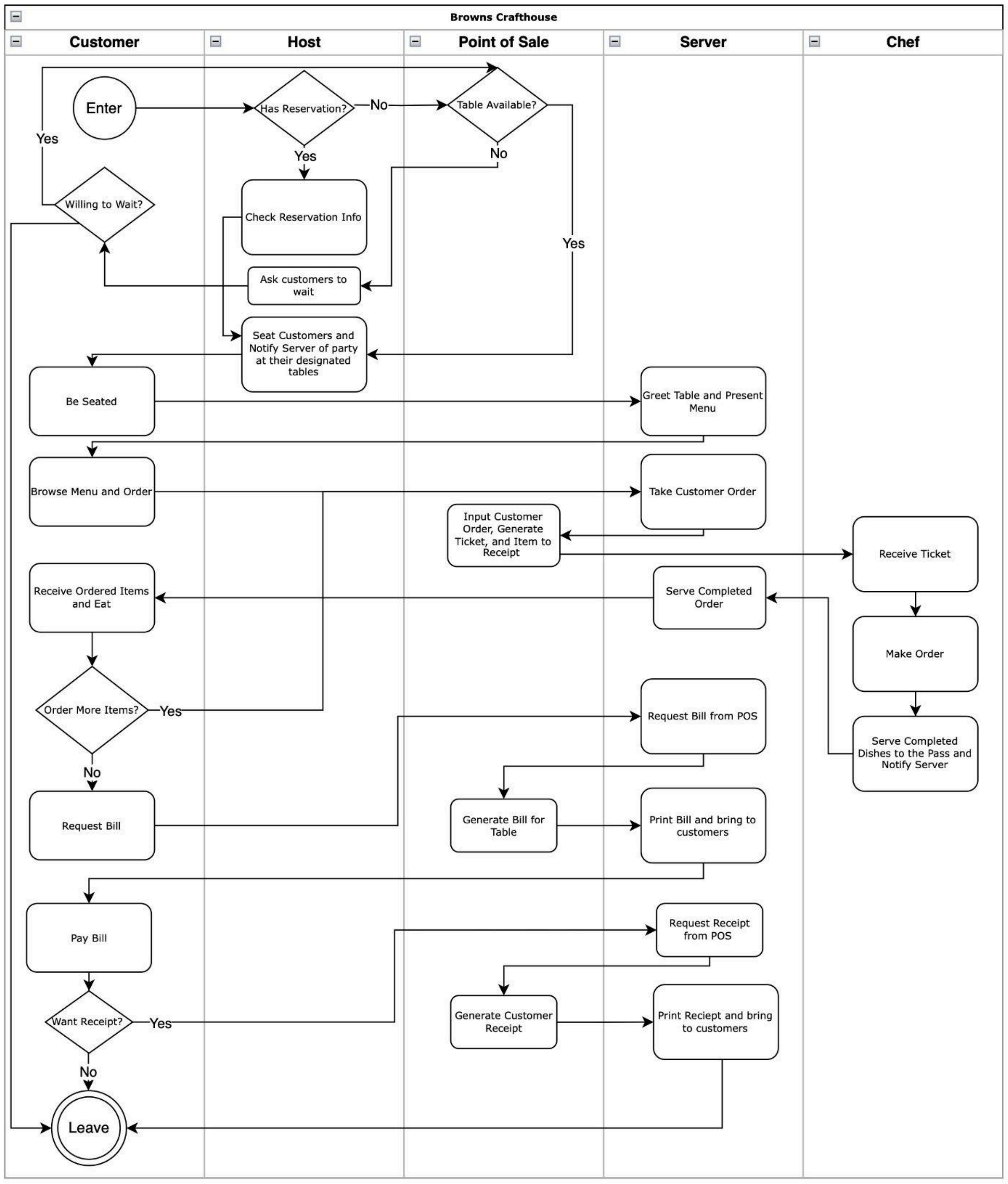
Database Design Decisions

Before developing the database, it was decided the POS system would be excluded as it is a front-end application that will use all aspects of the database (the backend) to create a user-friendly interface for staff to interact with. Thus, it was exempted from the Entity Relationship Diagram and Relational Schema. Additionally, the designed database handles customer processing, meaning it does not include elements such as food inventory and detailed staff information. It is assumed that these business functions are controlled via a separate database to improve the overall efficiency of the databases.

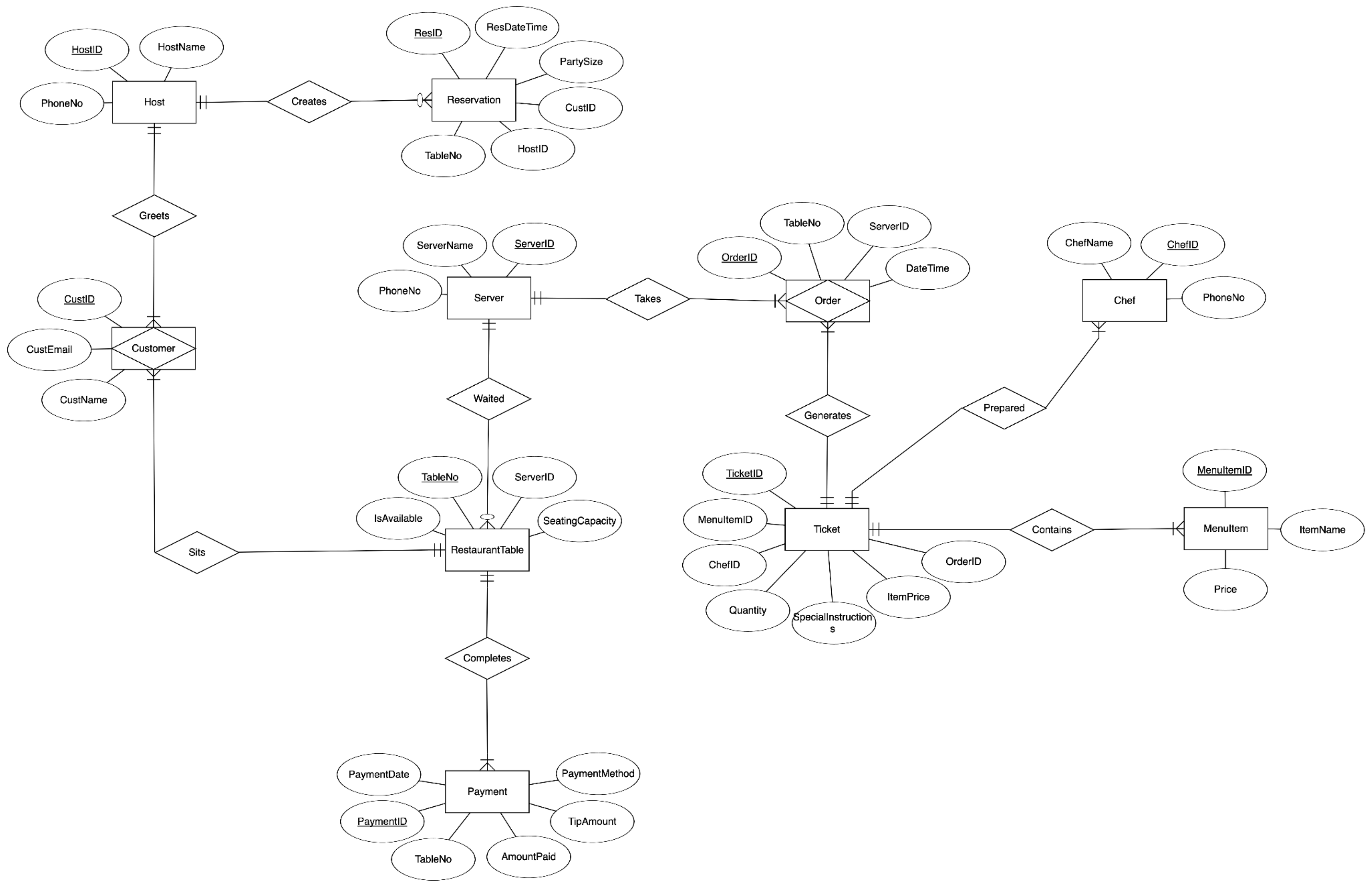
Additional Assumptions

Browns Crafthouse considers take-out orders as a separate business process from dine-in customers. The process of ensuring that food is ready to be served is handled by the kitchen staff and is also considered a separate business process. Furthermore, it is assumed that the food inventory is always up to date via the Point of Sale system (POS). Thus, meal options with missing food items will never be ordered.

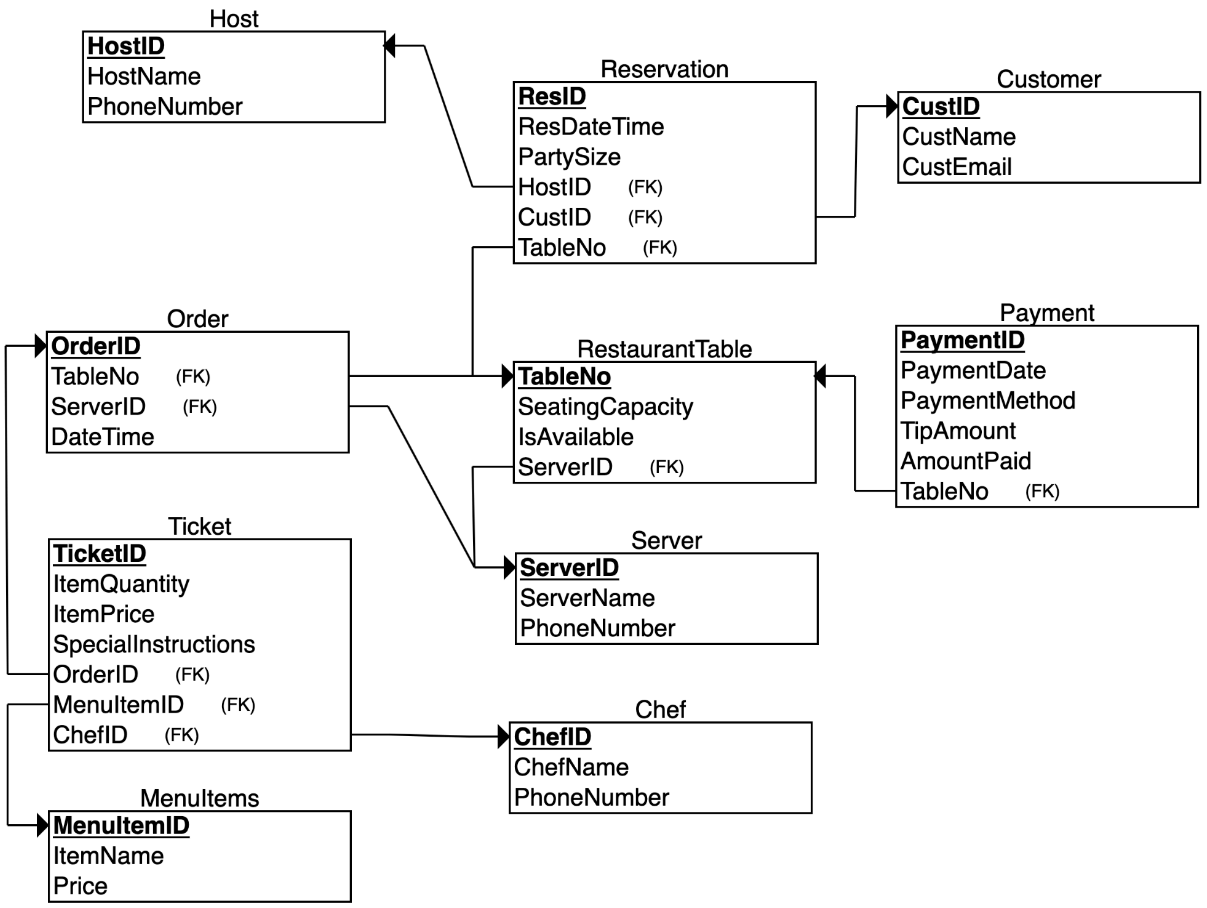
Business Process Map



Entity Relationship Diagram



Relational Schema



SQL Code

```
CREATE TABLE Customer
(
  CustID INT NOT NULL,
  CustName VARCHAR(100) NOT NULL,
  CustEmail VARCHAR(100) NOT NULL,
  PRIMARY KEY (CustID)
);

CREATE TABLE Chef
(
  ChefID INT NOT NULL,
  ChefName VARCHAR(100) NOT NULL,
  PhoneNumber VARCHAR(17) NOT NULL,
  PRIMARY KEY (ChefID)
);

CREATE TABLE Server
(
  ServerID INT NOT NULL,
  ServerName VARCHAR(100) NOT NULL,
  PhoneNumber VARCHAR(17) NOT NULL,
  PRIMARY KEY (ServerID)
);

CREATE TABLE Host
(
  HostID INT NOT NULL,
  HostName VARCHAR(100) NOT NULL,
  PhoneNumber VARCHAR(17) NOT NULL,
  PRIMARY KEY (HostID)
);

CREATE TABLE MenuItems
(
  MenuItemID INT NOT NULL,
  ItemName VARCHAR(100) NOT NULL,
  Price FLOAT NOT NULL,
  PRIMARY KEY (MenuItemID)
);

CREATE TABLE RestaurantTable
(
  TableNo INT NOT NULL,
  SeatingCapacity INT NOT NULL,
  IsAvailable BOOLEAN NOT NULL,
  ServerID INT NOT NULL,
  PRIMARY KEY (TableNo),
  FOREIGN KEY (ServerID) REFERENCES Server(ServerID)
);

CREATE TABLE Payment
(
  PaymentID INT NOT NULL,
  PaymentDate DATE NOT NULL,
  PaymentMethod VARCHAR(50) NOT NULL,
  TipAmount FLOAT NOT NULL,
  AmountPaid FLOAT NOT NULL,
  TableNo INT NOT NULL,
  PRIMARY KEY (PaymentID),
  FOREIGN KEY (TableNo) REFERENCES RestaurantTable(TableNo)
);

CREATE TABLE Order
(
  OrderID INT NOT NULL,
  DateTime DATETIME NOT NULL,
  TableNo INT NOT NULL,
  ServerID INT NOT NULL,
  PRIMARY KEY (OrderID),
  FOREIGN KEY (TableNo) REFERENCES RestaurantTable(TableNo),
  FOREIGN KEY (ServerID) REFERENCES Server(ServerID)
);

CREATE TABLE Reservation
(
  ResID INT NOT NULL,
  ResDateTime DATETIME NOT NULL,
  PartySize INT NOT NULL,
  HostID INT NOT NULL,
  CustID INT NOT NULL,
  TableNo INT NOT NULL,
  PRIMARY KEY (ResID),
  FOREIGN KEY (HostID) REFERENCES Host(HostID),
  FOREIGN KEY (CustID) REFERENCES Customer(CustID),
  FOREIGN KEY (TableNo) REFERENCES RestaurantTable(TableNo)
);

CREATE TABLE Ticket
(
  TicketID INT NOT NULL,
  ItemQuantity INT NOT NULL,
  ItemPrice FLOAT NOT NULL,
  SpecialInstructions VARCHAR(255) NOT NULL,
  OrderID INT NOT NULL,
  MenuItemID INT NOT NULL,
  ChefID INT NOT NULL,
  PRIMARY KEY (TicketID),
  FOREIGN KEY (OrderID) REFERENCES Order(OrderID),
  FOREIGN KEY (MenuItemID) REFERENCES MenuItems(MenuItemID),
  FOREIGN KEY (ChefID) REFERENCES Chef(ChefID)
);
```