Business Requirements for Pre-prepared Dishes Ordering System

By CS5200 Fall 2024 semester Qiong Wu Project 1

Inspired by Locale https://www.ycombinator.com/launches/IgH-locale-we-re-making-restaurant-delivery-affordable

Highlight nouns and verbs

Introduction

Compared to the flexibility and diversity of China's food delivery industry, the Bay Area's system feels limited and cumbersome. Inspired by Locale's model and the experience of sharing Uber rides, the concept of consolidating multiple orders into a single delivery trip offers a tangible solution. Grouping pre-prepared dishes from different suppliers into one optimized delivery route reduces operational expenses, fuel costs, and provides an eco-friendlier system, while offering flexibility to customers and easing the pressure on local restaurants during peak hours.

Problem Statement

Local restaurants face challenges from surging demand during peak hours and quality issues caused by long-distance, real-time delivery. These factors increase pressure on suppliers and lead to higher costs for both customers and restaurants. Current platforms focus on real-time deliveries, requiring multiple orders from nearby restaurants, which inflates delivery fees and complicates the process. Meanwhile, each meal is typically ordered from one restaurant, resulting in a high proportion of the cost being attributed to delivery fees. If customers want to order from another nearby restaurant, they are required to place a separate order, increasing costs further.

My Solution

My pre-prepared dishes ordering system will focus on meals from local suppliers, consolidating orders and offering flexible delivery windows to optimize restaurant workloads. By avoiding peak traffic hours and using the best possible delivery routes, the system ensures an enhanced experience for drivers and restaurants. The key features of the database design include:

1. **Flexible Meal Bundles**: Customers can create their own meal bundles from various local restaurants.
2. **Flexible Delivery Windows**: Customers select flexible delivery windows, allowing restaurants to prepare meals without the rush of real-time orders.
3. **Optimized Route Delivery**: The system consolidates orders from multiple suppliers into a single delivery trip, using Google Maps API and OR-Tools, reducing fuel and time costs for drivers. (idea from ChatGPT)
4. **Cost Reduction for Restaurants**: By spreading out delivery and allowing restaurants to prepare meals at less congested times, overhead costs are lowered, leading to better margins.

Conclusion

By combining ideas from Locale's innovative model and the ride-sharing concept, this system creates a streamlined, cost-effective, and environmentally friendly meal delivery service. It optimizes logistics, enhances supplier efficiency, and improves the overall customer experience, offering a competitive advantage over traditional food delivery platforms.

Collect the nouns and verbs

Nouns

- food delivery industry
- the Bay Area
- Locale's model
- sharing Uber rides
- multiple orders
- orders
- single delivery trip
- pre-prepared dishes
- suppliers
- optimized delivery route
- operational expenses
- fuel costs

- eco-friendlier system
- customers
- local restaurants
- peak hours
- demand
- quality issues
- long-distance, real-time delivery
- pressure
- higher costs
- platforms
- delivery fees
- proportion
- pre-prepared dishes ordering system
- delivery windows
- workloads
- experience
- drivers
- Meal Bundles
- Google Maps API
- OR-Tools
- fuel
- time costs
- logistics
- supplier efficiency
- advantage

Verbs

- consolidating
- offers
- Grouping
- reduces
- easing
- face
- caused
- increase
- focus on
- inflates
- complicates
- resulting
- attributed
- optimize

- avoiding
- using
- ensures
- create
- select
- prepare
- spreading
- combining

Reorganize the Nouns and Verbs. Classify them

Nouns

- food delivery industry
  - the Bay Area
- orders
  - optimized delivery route
  - operational expenses
  - fuel costs
  - long-distance, real-time delivery
- suppliers
  - local restaurants
  - supplier efficiency
- customers
  - demand
  - orders
- delivery
  - fees
  - proportion
- pre-prepared dishes ordering system
  - single delivery trip
  - pre-prepared dishes
  - delivery windows
  - Meal Bundles
  - Google Maps API
  - OR-Tools
- drivers
  - fuel
  - time costs
  - logistics

Verbs

- create

- offers
- grouping
- optimize
- avoiding
- select
- prepare
- spreading
- combining

Rules of Business

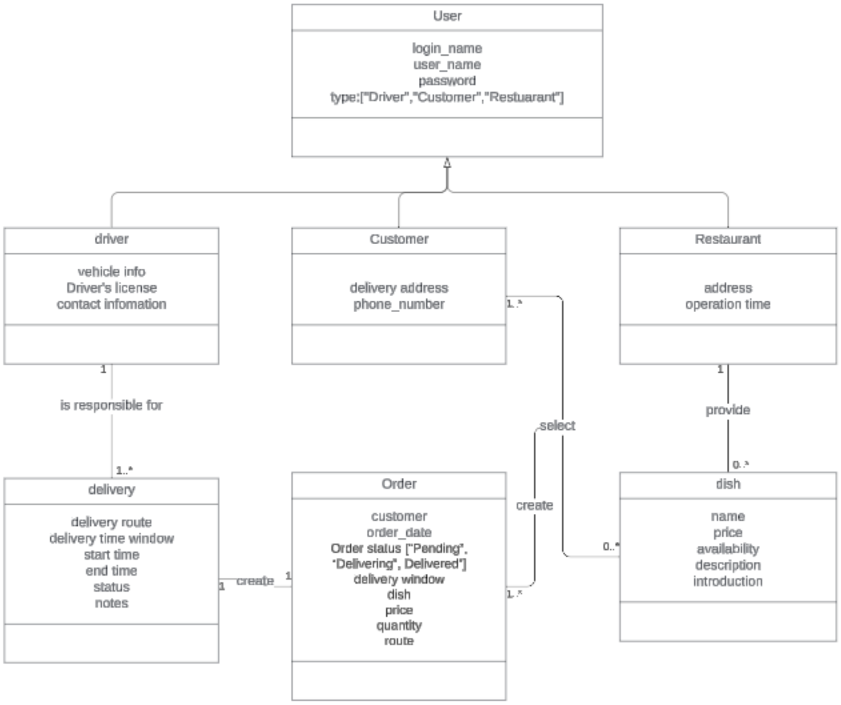
- Any user can create accounts as customer, but they must provide valid address and phone number.
- The accounts for supply restaurants will be created after approval following a partnership agreement.
- Individuals can create delivery accounts, but they must provide a valid driver's license and vehicle information.
- Each supplier can only provide one menu but can have multiple dishes with price and availability quantity.
- Customers can place an order consisting of items from multiple suppliers.
- All selected items from different suppliers are combined into a single order for delivery.
- The system will analyze traffic patterns and supplier preparation times to suggest optimal delivery windows.
- Drivers can select orders, and once the driver selects the orders, a suggested pickup sequence is created.
- Suppliers can update their menu offerings, including prices, availability, and descriptions. But the price of existing orders will not change.

Extract classes

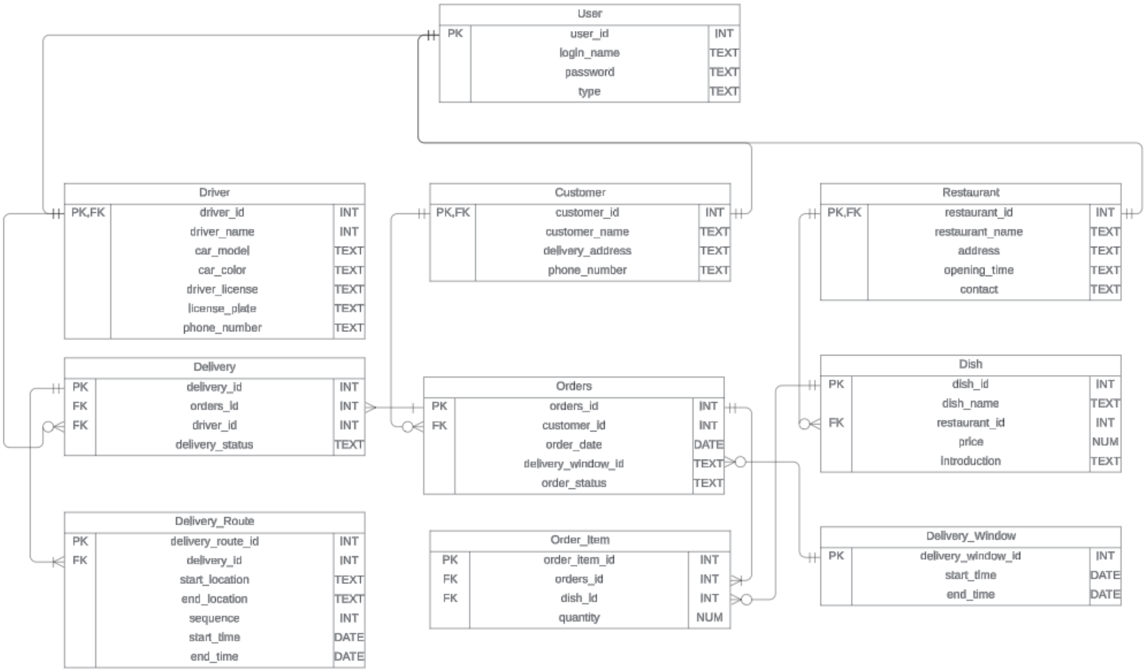
- Customer
    - Login
    - Password
    - Name
    - Address
    - Phone number
- Suppliers
    - Name
    - Address
    - Opening hours

    - Menu
- Supplier Menu
    - Supplier
    - Price
    - Availability
    - Description
    - Introduction
- Order
    - Customer
    - Order date
    - Order status [“Pending”, “Delivering”, Delivered”]
- Order items
    - Item
    - Price
    - Quantity
- Delivery
    - Order
    - Driver
    - Pickup Time
    - Delivery Time
- Delivery route
    - Start location
    - End location
    - Stops
    - Sequence
- Delivery Windows
    - Date
    - Start time
    - End time
- Driver
    - Name
    - Vehicle Info
    - Driver’s license
    - Phone Number

UML Class Diagram



ERD



Relational Schema

- User (user_id: INTEGER, login_name: TEXT, password: TEXT, type: TEXT)
- Customer (*customer_id*: INTEGER, customer_name: TEXT, delivery_address: TEXT, phone_number: TEXT)
- Restaurant (*restaurant_id*: INTEGER, restaurant_name: TEXT, restaurant_address: TEXT, opening_time: TEXT, contact: TEXT)
- Dish (dish_id: INTEGER, item_name: TEXT, price: REAL, introduction: TEXT)
- Orders (orders_id: INTEGER, *customer_id*: INTEGER, order_date: TEXT, order_status: TEXT)
- Order_Item (order_item_id: INTEGER, *orders_id*: INTEGER, *dish_id*: INTEGER, quantity: INTEGER)
- Delivery (delivery_id: INTEGER, *orders_id*: INTEGER, *driver_id*: INTEGER, delivery_status: TEXT, delivery_date: DATE)
- Driver (*driver_id*: INTEGER, car_model: TEXT, car_color: TEXT, driver_license: TEXT, license_plate: TEXT, phone_number: TEXT)

- Delivery_Route (delivery_route_id: INTEGER, *delivery_id*: INTEGER, start_location: TEXT, end_location: TEXT, sequence: INTEGER, start_time: DATE, end_time: DATE)
- Delivery_Window (delivery_window_id, start_time: DATE, end_time: DATE)

Functional dependencies:

- **User:**
    - user_id -> login_name, password, type
    - user_id is a primary key
- **Customer:**
    - customer_id -> delivery_address, phone_number
    - customer_id is a primary key
- **Restaurant:**
    - restaurant_id -> restaurant_name, restaurant_address, opening_time, closing_time, contact
    - restaurant_id is a primary key
- **Dish:**
    - dish_id -> item_name, price, introduction
    - dish_id is a primary key
- **Orders:**
    - orders_id -> customer_id, order_date, order_status
    - orders_id is a primary key
- **Order_Item:**
    - order_item_id -> orders_id, dish_id, quantity
    - (orders_id, dish_id) -> quantity
    - order_item_id is a primary key
    - (orders_id, dish_id) is unique
- **Delivery:**
    - delivery_id -> orders_id, driver_id, delivery_status, delivery_route_id
    - delivery_id is a primary key
- **Driver:**
    - driver_id -> car_model, car_color, driver_license, license_plate, phone

  - driver_id is a primary key
- **Delivery_Route:**
  - delivery_route_id -> start_location, end_location, sequence, start_time, end_time
  - delivery_route_id is a primary key
- Delivery_Window:
  - delivery_window_id -> start_time, end_time
  - delivery_window_id is a primary key