

System Design for a Food delivery service

Main actors in our system:

- Customers/consumers
- Restaurant (Merchant)
- Delivery Agents (drivers)
- System Admin (from the company)

Functional Requirements:

Customers should be able to:

- Create a cart, add menu items to it, and then place an order.
- Once an order has been placed, customers should be receiving notifications of its status.
- On the app, customers may check the status of their order.
- Cancel the order.
- Make a payment for the order.
- Create an account with their contact details and update them.

Restaurants should be able to:

- Set up their profile (onboarding), create/refresh/add new menu items, and upload photos.
- Receive notifications regarding orders placed, delivery agent assigned, and order progress updates, among other things.
- Take feedback about their orders from customers.
- Off board: When a restaurant closes its doors or decides to stop accepting online orders.

Delivery Agents should be able to:

- Receive updates about available orders in their region, which they can choose from.
- When the order will be ready for pickup, they should be notified.
- Any difficulties encountered during order pickup/delivery should be reported to the customer/restaurant.
- If they don't wish to continue working, they can de-register.

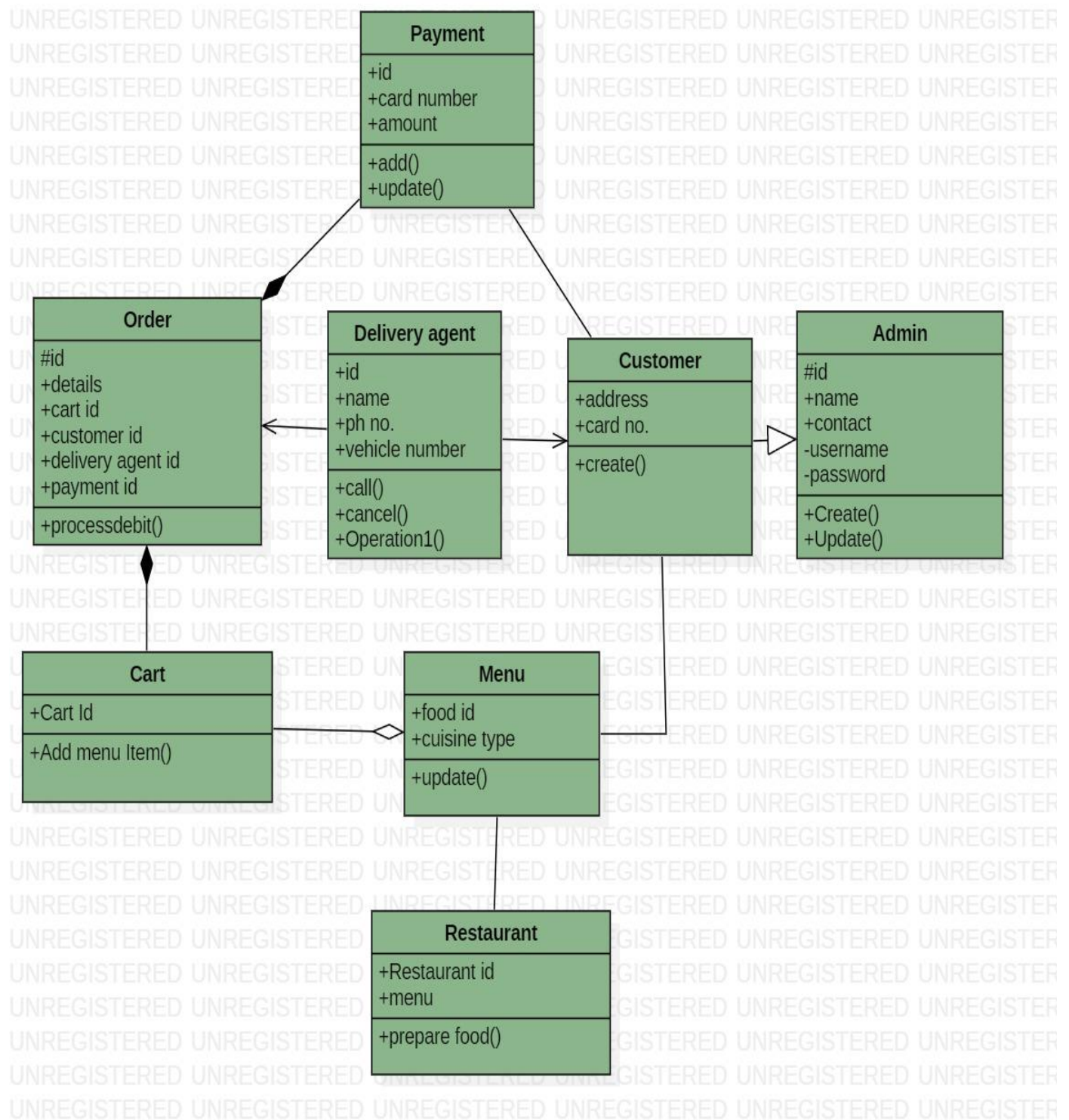
System Admin should be able to:

- Select which restaurants to include in the system.
- Manage customer's info and status.
- Monitor orders and info.
- Provide interactive interface between all the actors.
- Record order payments and deliveries.
- Take feedback from restaurants and delivery agents.

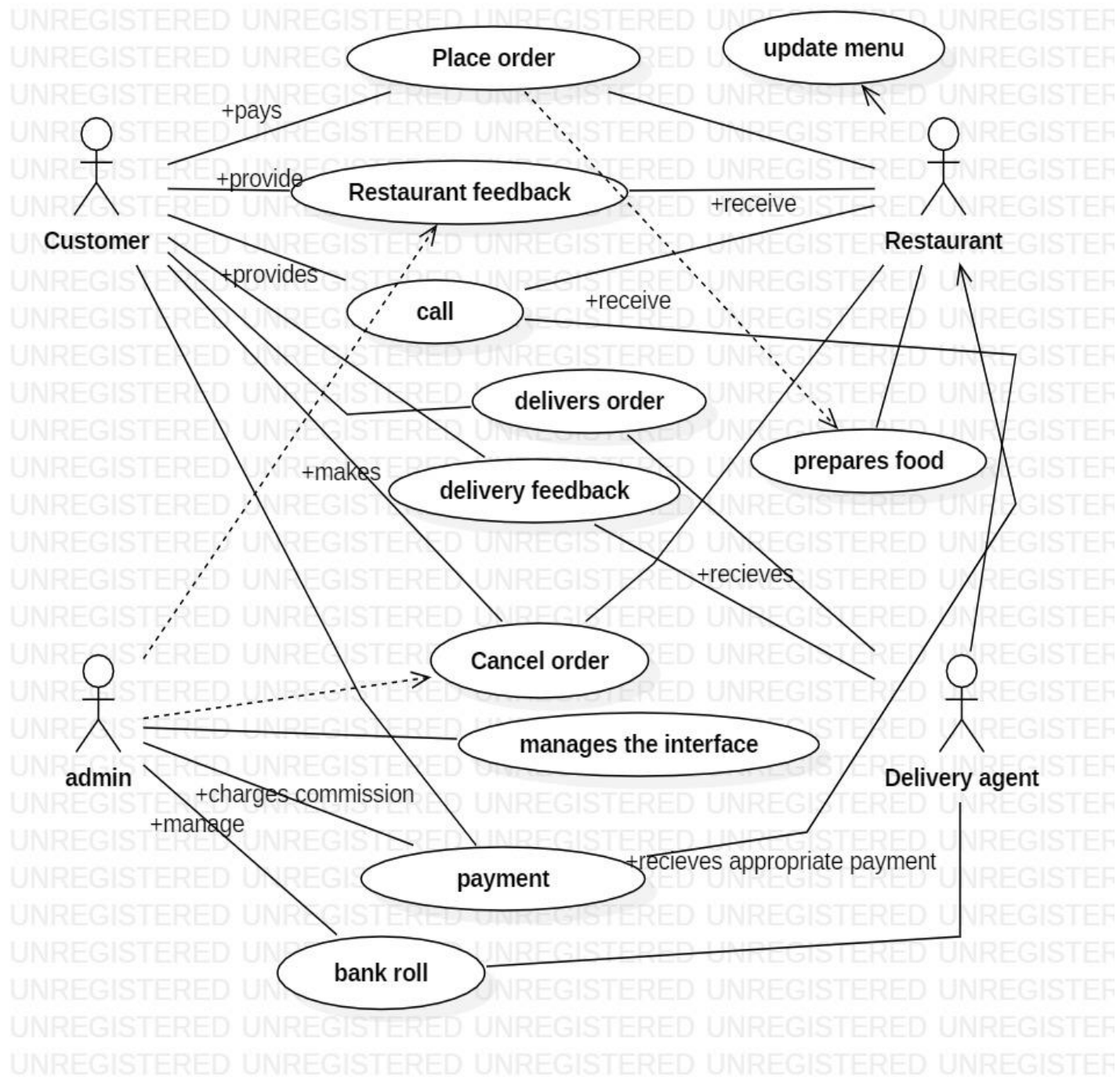
Non Functional Requirements:

- **Latency:** Hungry users would not want to wait to see the menu or restaurant specifics, so the search feature should be quick. The ordering experience should also not have high latency and must be seamless and fast. Also, because the restaurant/menu-related data will be entered through a separate service, the time between when the data is entered and when the result appears in the search should be acceptable but not excessive.
- **Consistency:** When a new restaurant or menu is added, the information does not have to be available right away. It's preferable to have a consistent outcome. When an order is placed, however, the client, the restaurant, and the door dasher should all see the same order. As a result, in this situation, consistency is critical.
- **Seamless User Interface:** The user should be able to use this delivery system without encountering any technical glitches.
- **Availability:** High availability is desirable for the optimal customer experience, as well as for the restaurants handling the order and the food delivery agent. No restaurant wants their business to suffer because of a system failure as that's a significant financial loss.
- **High Throughput:** The system should be able to handle high peak load without problems or failures.

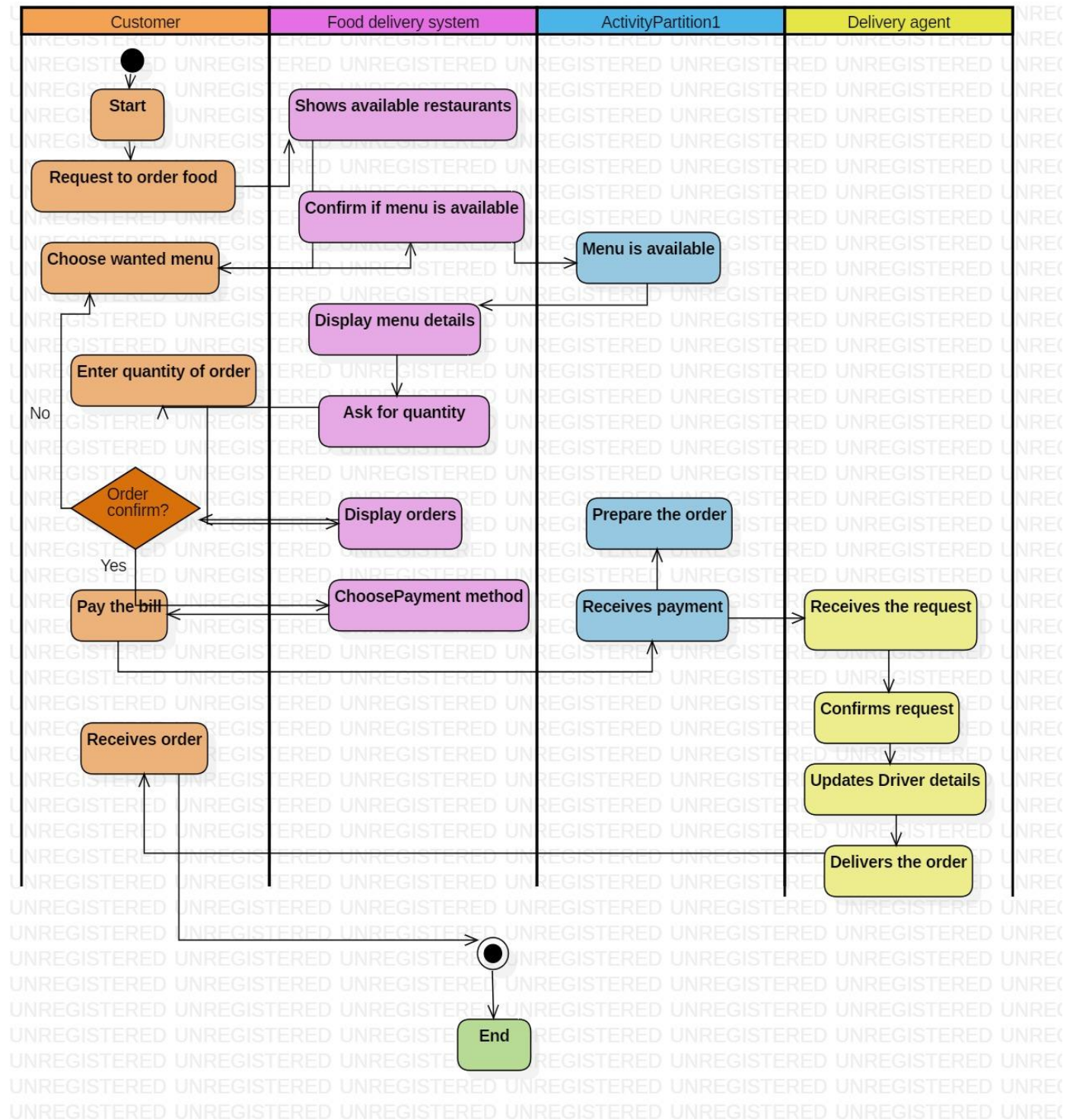
Online Food Ordering System Class Diagram:



Online Food Ordering System General Use case Diagram:



Online Food Ordering System Activity Diagram:



Online Food Ordering System Sequence Diagram:

