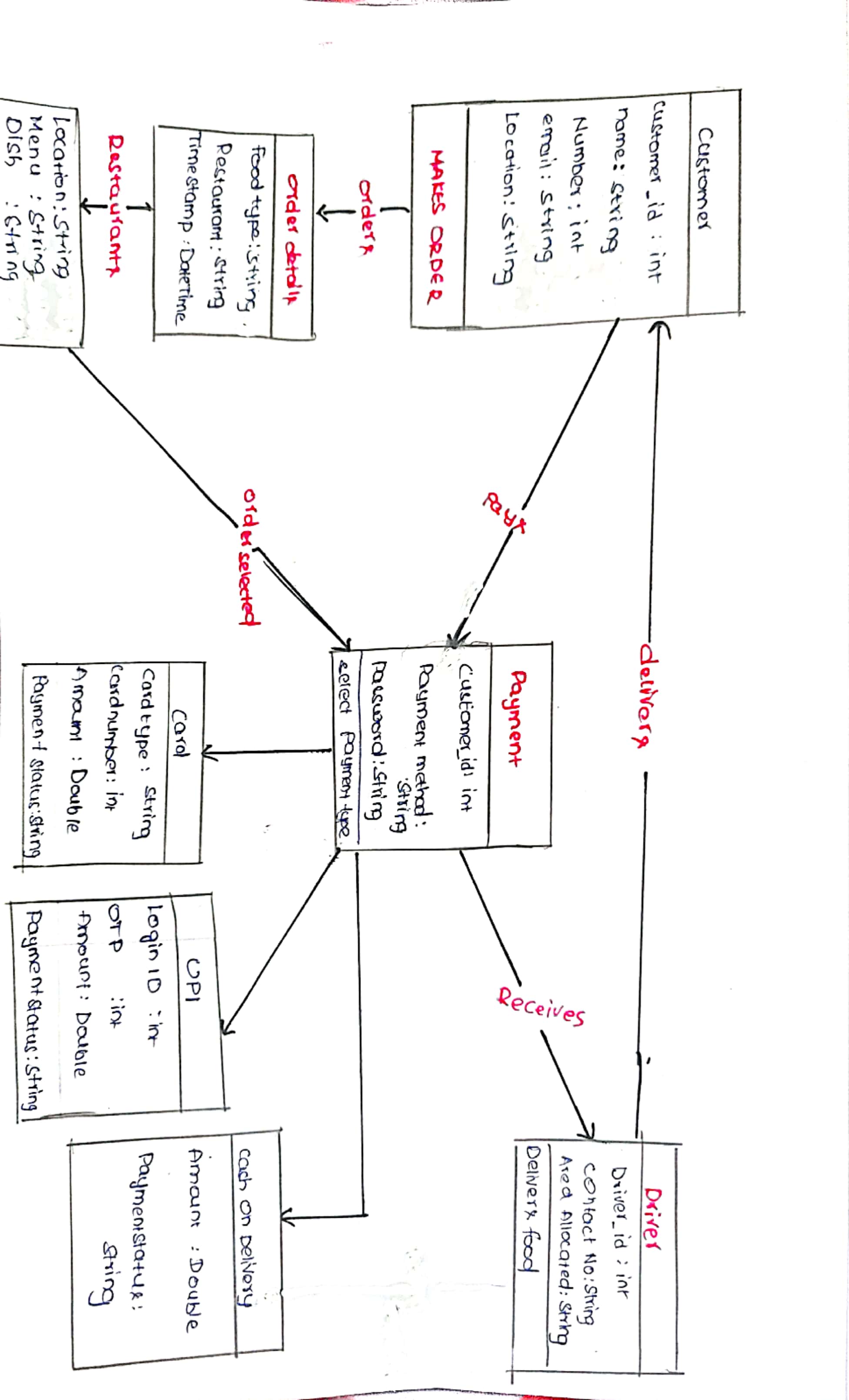
**LLD DESIGN**

****

**1. Authentication Service:**

* **Trivial/Novel:**
  + Trivial: Authentication services are common in many applications, and basic functionalities like user registration, login, and password management are well-established and standardized.
  + Small/Large:
    - Small: Implementing authentication involves standard practices and libraries. While crucial for security, the core functionalities are well-understood and relatively straightforward to implement.

**2. Restaurant Service:**

* **Trivial/Novel:**
  + Novel: While restaurant listings and search functionalities are common, managing a diverse range of restaurant data, ensuring accuracy, and integrating external sources (like APIs for restaurant details) present novel challenges.
  + Small/Large:
    - Large: Managing restaurant data involves complex data modelling, integration with external APIs, and ensuring data consistency. Handling a large volume of data and real-time updates adds to the complexity.

**3. Order Service:**

* **Trivial/Novel:**
  + Trivial: Order management functionalities like placing orders, tracking status, and managing transactions are common in similar applications.
  + Small/Large:
    - Medium: While the core functionalities are standard, ensuring real-time updates, scalability, and reliability can increase complexity, requiring a medium effort.

**4. Payment Gateway Service:**

* **Trivial/Novel:**
  + Novel: Integrating with third-party payment gateways, ensuring secure transactions, and handling various payment methods involve novel challenges, especially in terms of compliance with industry standards and security.
  + Small/Large:
    - Large: Implementing a payment gateway service involves complex integration, encryption, and compliance with regulations like PCI DSS. Ensuring reliability and security while supporting multiple payment methods require a large effort.

**5. Review Service:**

* **Trivial/Novel:**
  + Novel: Managing user reviews, ratings, and sentiment analysis present unique challenges, especially in handling large volumes of data, spam detection, and ensuring data integrity.
  + Small/Large:
    - Large: Building a robust review service involves complex algorithms for sentiment analysis, moderation workflows, and data storage. Addressing spam, abusive content, and maintaining trust in the review system require a large effort.

**Explanation:**

* **Trivial/Novel:**
  + Determines whether the feature involves common practices or presents unique challenges not commonly encountered.
* **Small/Large:**
  + Indicates the scope and complexity of implementing the feature. Large efforts are needed for features with extensive functionalities or complex integrations.

**Conclusion:**

By categorizing each feature into these quadrants, development efforts can be prioritized and appropriate resources allocated to ensure successful implementation of the Zomato-like application, considering both its novelty and complexity.