**Для составления диаграммы, мы используем одну из диаграмм BPMN – swimlane**

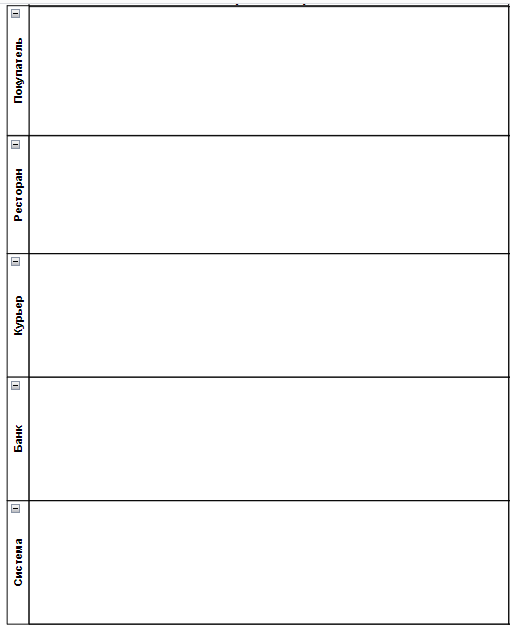
* Пунктирными линиями обозначается BackEnd часть (то что юзер не видит).
* Обычными линиями обозначается то куда юзер переходит и что он лично сам делает
* В логике обработки (система) писать только BackEnd часть (что юзер не видит) – подсчет суммы заказа, показ суммы, …

**Оформление заказа еды**

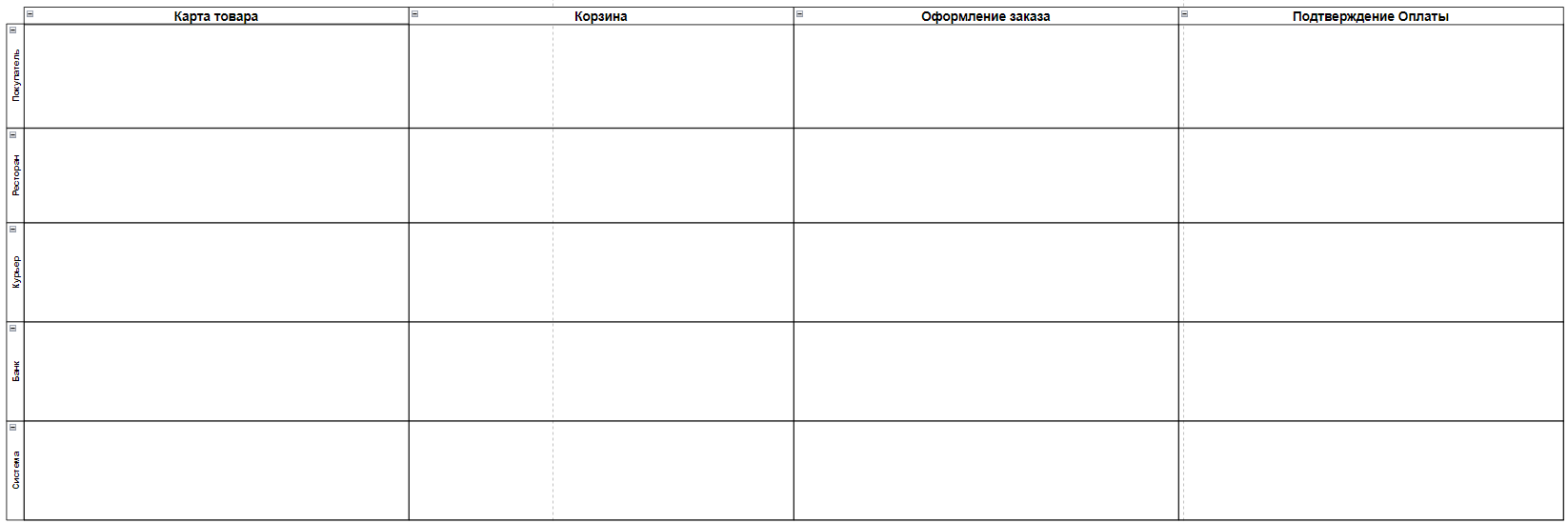
1. Прописать каждую сущность, которая взаимодействует. Прописывать сущности **по вертикали**

Выделять **внешние сущности** (Покупатель, Ресторан, Курьер, Банк), а так же **логику обработки** на нашей стороне (**Система**)

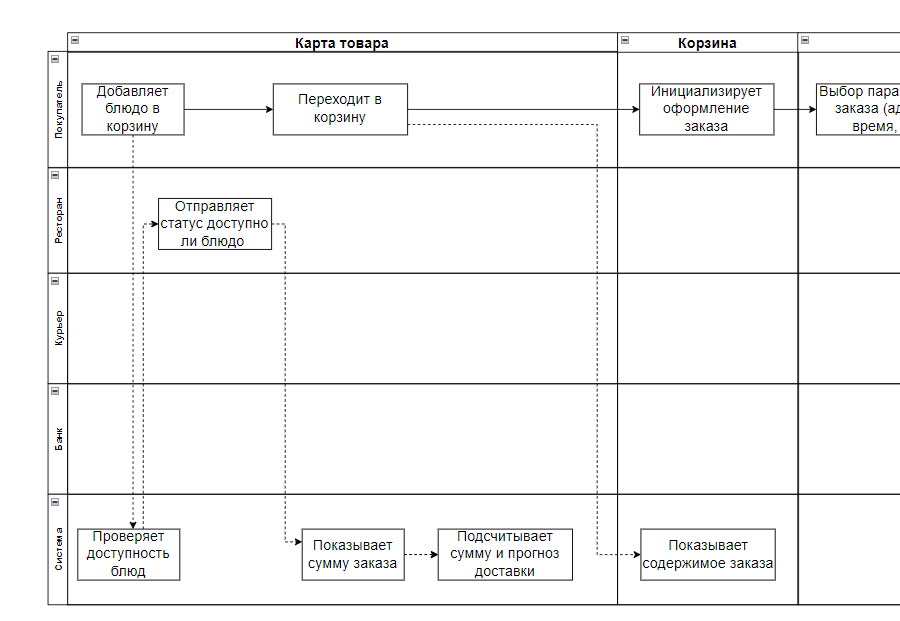
Внешние сущности могут быть каждый Flow разные, но вот логика обработки на нашей стороне всегда будет в диаграмме.



1. Прописать состояния в которых может быть User Flow. Прописывать состояния **по горизонтали**



1. Прописать какая сущность что делает и кому передает “управление дальше”



При тестировании на каждом шаге думать, а может ли действующее лицо сделать другой выбор.

* **Карта товара**: При добавлении блюда в корзину проверить есть ли такое блюдо или нет

…

* **Корзина**: Добавлены блюда из нескольких разных ресторанов, …

…

* **Оформление заказа**: при выборе параметров заказа (адрес, время, …) протестировать а что если ресторан не доставляет по этому адресу, или в это время. А что если время ресторану не хватит времени что бы доставить по этому адресу. Что если не хватает денег для оплаты. Что если оплата не прошла, …

…

* **Пост оплата**: На этапе передачи курьеру продумать, а что если курьер откажется от доставки и передаст другому курьеру.

…