极客时间架构师训练

闪送架构设计文档

Hanke

2021-02-26

# 背景介绍

公司是由上市公司X投资成立的一家物流快递公司，计划主要进行同城快递业务，希望在两个月内完成后端系统开发，支持同城快递业务的功能需求和快速发展。

# 设计概述

同城快递闪送系统是一个支持同城物流快递下单，送达的系统，是公司扩大市场占有量战略的核心系统，承担着通过线上快捷服务扩大用户群体增加公司总营收的目标任务。

## 功能概述

系统的主要功能包括: \* 支持用户app下单预约快递服务 \* 支持用户app端支付 \* 支持快递员汇报地理信息 \* 支持快递员抢单功能 \* 支持快递员修改订单状态信息 (已收件，已送达) \* 支持用户修改订单状态信息（取消，更改，删除） \* 支持用户查看订单信息

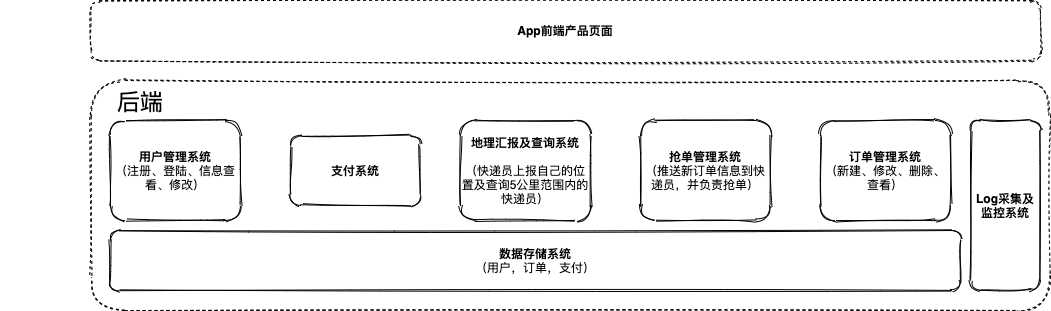
系统的主要用户包括： \* 使用app发送快递的用户 \* 快递员

## 非功能约束

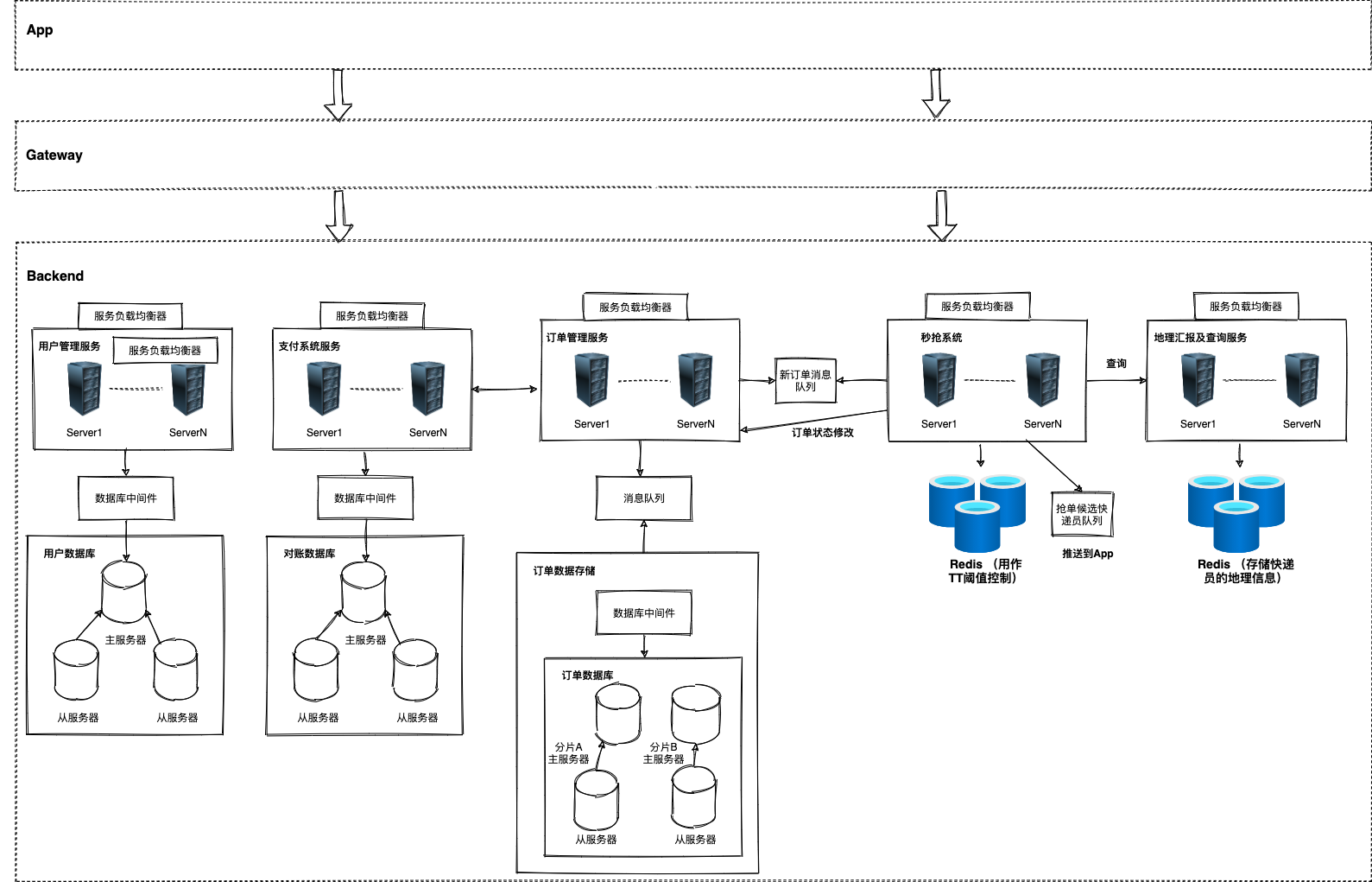
系统预计上线后三个月日单超过1万，一年后日单超过50万。 \* **下单性能目标**：平均响应时间<800ms， 95%的响应时间<1000ms，单台服务器的TPS>30 \* **支付性能目标**：平均响应时间<800ms， 95%的响应时间<1000ms，单台服务器的TPS>30 \* **汇报地理功能性能目标**：平均响应时间<800ms， 95%的响应时间<1000ms，单台服务器的TPS>30 \* **抢单性能目标**：平均响应时间<800ms， 95%的响应时间<1000ms，单台服务器的TPS>30 \* **订单状态修改性能目标**：平均响应时间<800ms， 95%的响应时间<1000ms，单台服务器的TPS>30 \* **查看订单信息性能目标**：平均响应时间<300ms， 95%的响应时间<500ms，单台服务器的TPS>100 \* **系统核心功能可用性目标**: > 99.9% (下单，支付，汇报地理信息，抢单) \* **系统安全性目标**: 系统可拦截恶意下单、盗取数据库、用户数据盗取等攻击，将密码数据进行加密，客户端数据HTTPS加密，外部系统间通信对称加密 \* **数据存储目标**: > 99.99%

# 系统整体设计

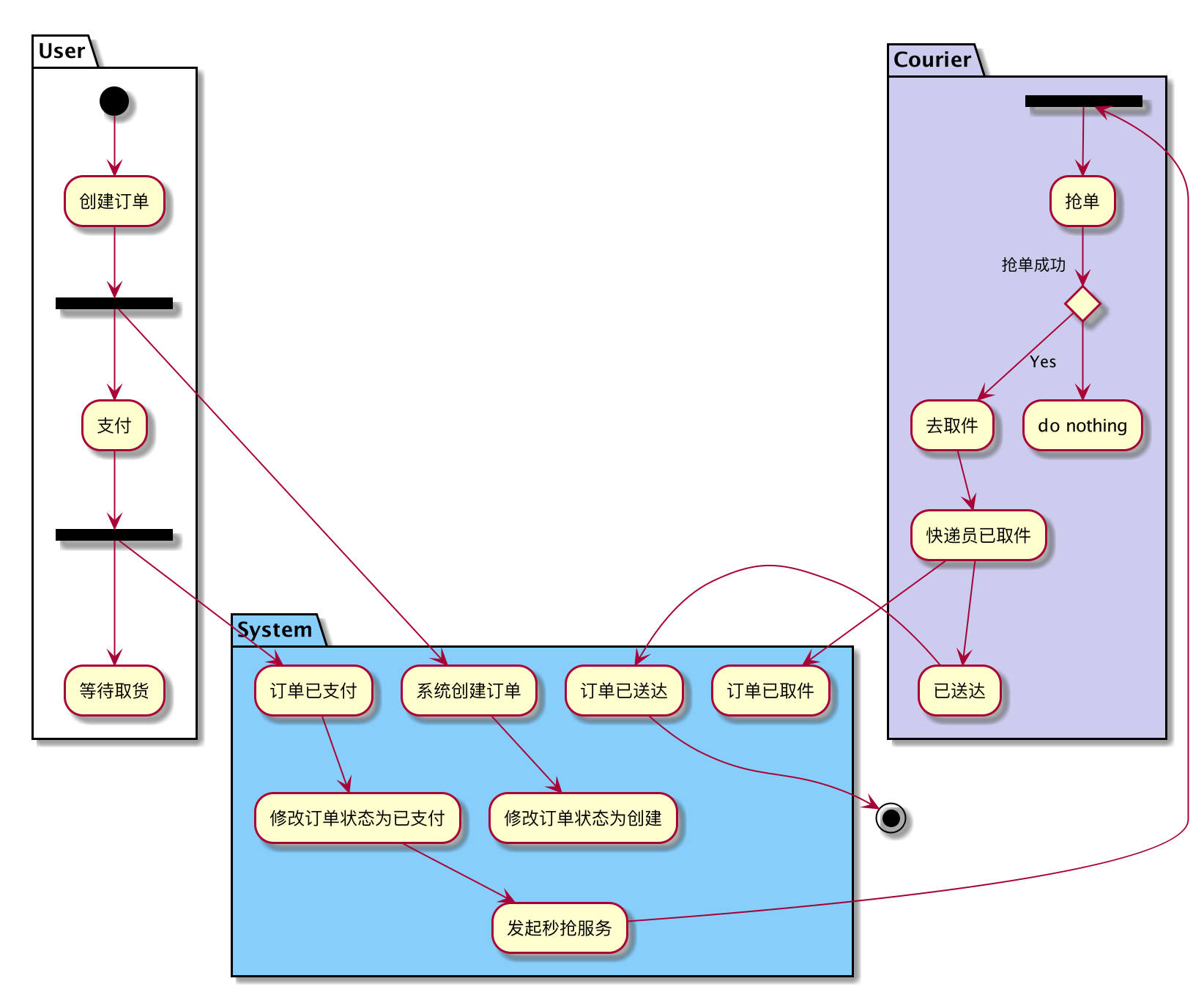
### 逻辑功能图

 主要功能服务模块包括: \* **用户管理模块** \* 主要负责用户的注册，登陆，信息查看，修改 \* **支付系统** \* 主要负责对账支付功能 \* **地理汇报及查询系统** \* 主要负责收集快递员的地理汇报信息，并提供五公里范围内快递员列表查询服务 \* **抢单系统** \* 主要负责发送抢单信息到快递员，并将抢单成功与否信息反馈给用户和订单系统 \* **订单管理系统** \* 主要负责订单的创建，修改，查看服务 \* **Log采集及监控系统** \* 主要整个系统的Log收集及监控

### 系统部署图

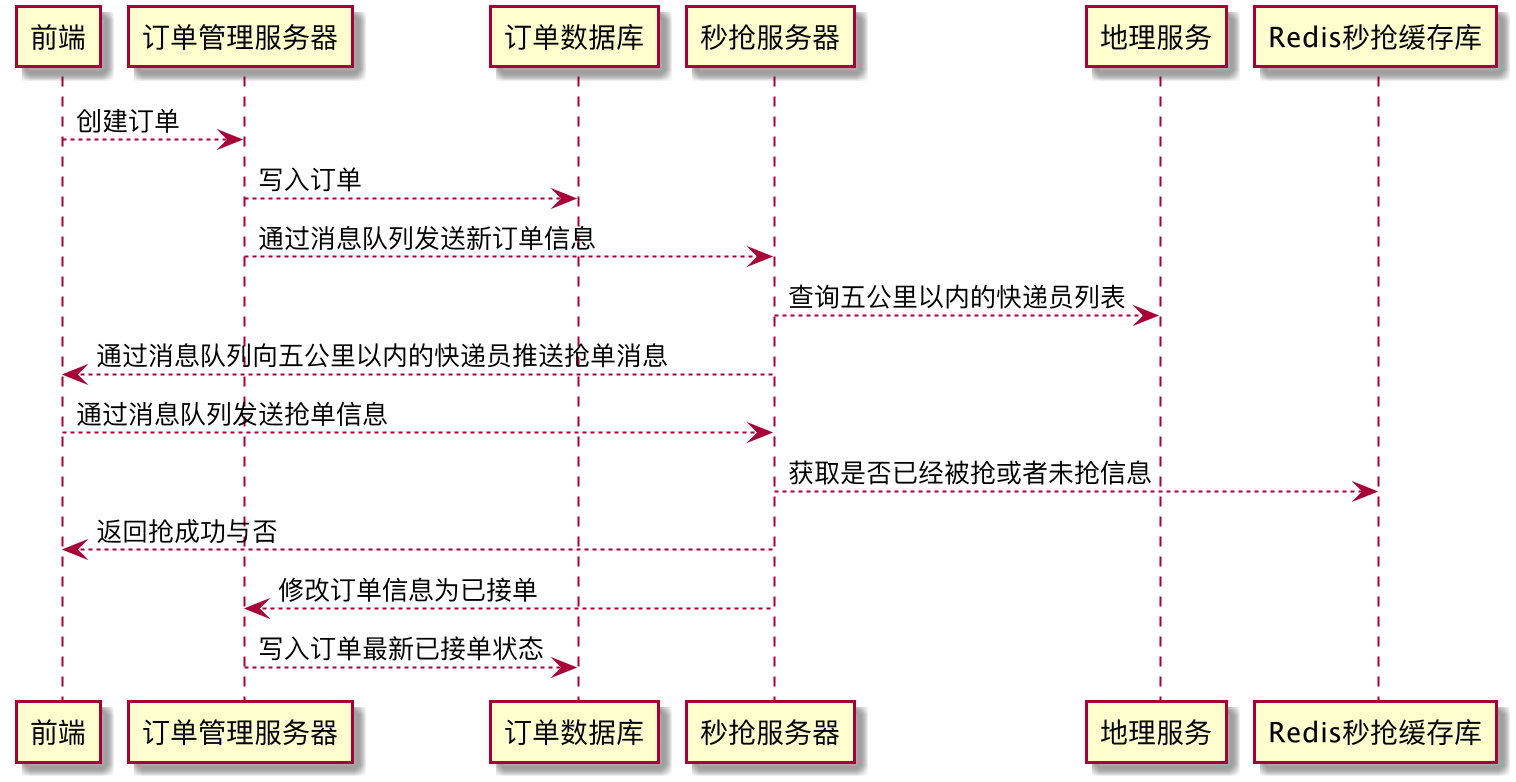
 关键信息包括： \* 采用微服务的方式对系统及服务进行拆解并分别部署，减小依赖和影响 \* 每个微服务采用多个服务器部署+ 服务负载均衡器的访问的方式来保证高性能和HA \* 数据存储层至少是按照主从复制的方式来保证数据的可靠性，并通过读写分离来提升性能 \* 订单存储层一期可采用主从复制+按时间进行分表即可，按时间分表便于后期冷热数据隔离，后期要支撑更大量的并发可能通过按照订单简单的hash进行分库操作来支持更高并发。 \* 订单修改部分可以采用消息队列的方式来进很峰谷削平，减小对数据库的瞬间压力 \* 秒抢系统和订单管理系统部分可以通过消息队列来解耦 \* 秒抢系统可以利用Redis缓存来实现阈值控制，只允许一个快递抢单成功，其他都直接reject返回即可 \* 地理汇报和查询服务用Redis即可，因为快递员的数量也不会太多，万级别的并发足够支撑

### 下单抢单场景的业务活动图, 角色领域泳道模型



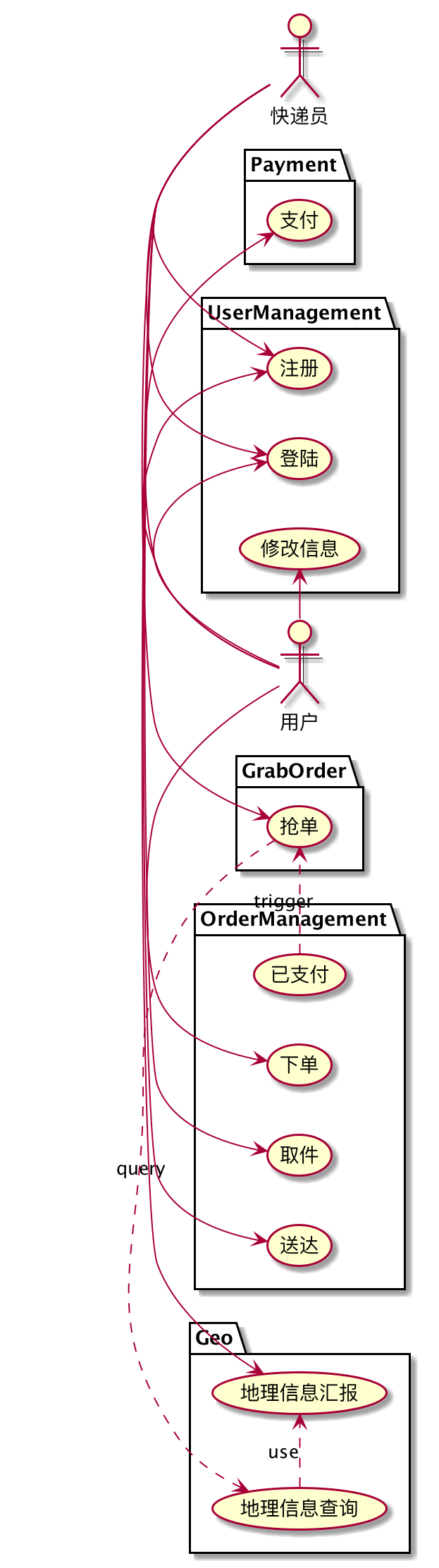
graborder\_activity

### 下单抢单场景的服务器时序模型



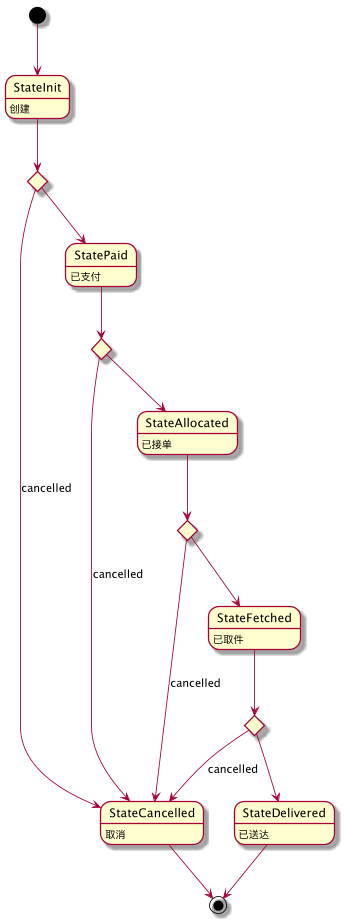
graborder\_seq

### 关键用例图



usecase

### 订单状态图模型



orderstate

# 功能及性能测试

* 功能测试部分需覆盖功能点所有的场景测试
* 性能测试需覆盖所有非功能性约束测试

# 人力和时间安排

* 20 persons \* 2月完成一期开发+测试，支撑上线后三个月日订单1万的需求
* 20 persons \* 2月完成二期系统扩展+监控+数据分析+性能调优，支撑一俱后日订单5万的需求

# 后续系统扩展

* 并发支撑能力扩展
  + 并发支撑可以扩展可以按照订单进行数据的隔离和服务的隔离
    - 数据隔离：可以按照订单进行简单的hash进行数据分库 + 数据库中间件进行访问
    - 服务隔离：可以再建立二级负载均衡服务器+订单Session Mapping服务集群，通过订单进行路由，对订单进行分流，而底层的订单相关的服务可以部多套。
* 监控
  + 监控用户日活，日增，订单日增，收益等防止出现功能及系统问题
* 数据分析
  + 建立关键数据分析报表和仪表盘

本网站的文章除非特别声明，全部都是原创。 原创文章版权归数据元素([DataElement](https://www.dataelement.top))所有，未经许可不得转载!  
**了解更多大数据相关分享，可关注微信公众号“数据元素”** 