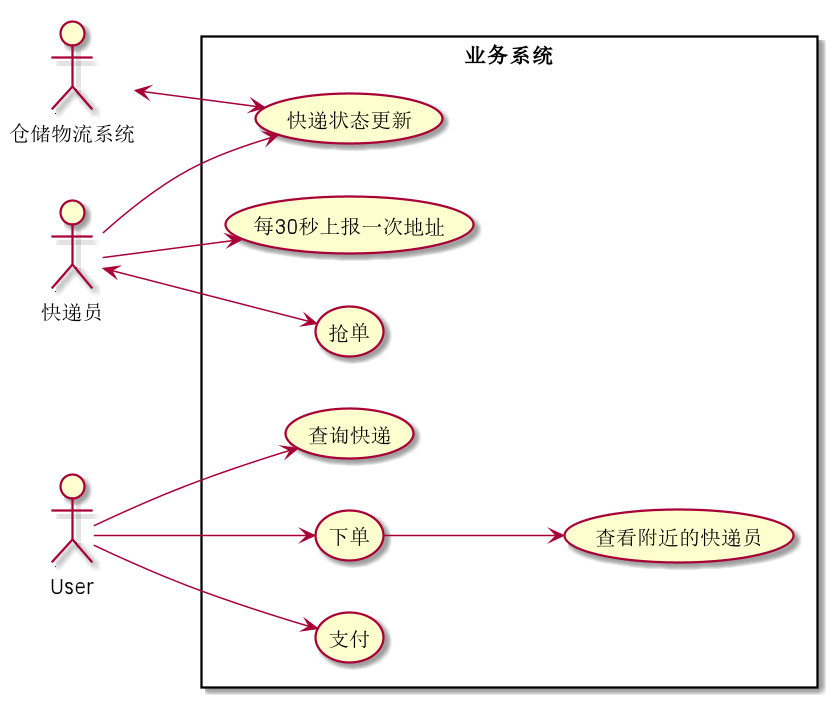
# 架构设计大作业

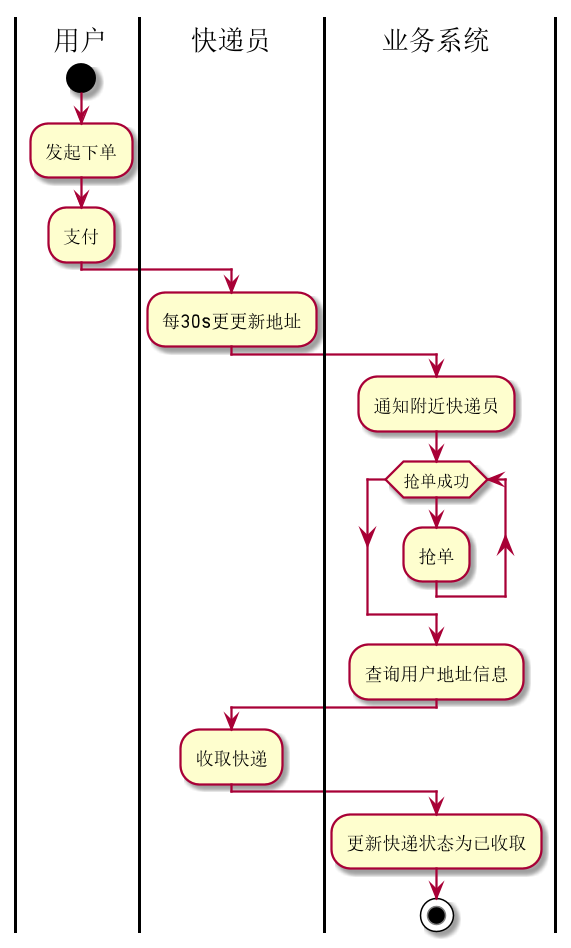
## 1 设计概述

通达业务系统是通达物流快递公司的后端核心系统，承担着通达的同城快递业务的开展工作。预计在两个月内系统上线。

### 1.1 功能概述

通达业务系统主要功能包括下单支付功能, 抢单功能，上报快递员地址，查询快递，更新快递状态的核心功能。系统使⽤者包括用户，快递员，仓储物流系统。.





### 1.2 ⾮功能约束

系统未来预计三个月用户量10万，快递员300人，日单超过1万，一年日单超过50万。

1. 用户下单/支付性能⽬标：平均响应时间<800ms，95%响应时间<1000ms，单机TPS>50；

2. 快递抢单性能⽬标：平均响应时间<800ms，95%响应时间<1000ms，单机TPS>100；

3. 查询快递状态性能⽬标：平均响应时间<300ms，95%响应时间<500ms，单机TPS>100；

4. 更新快递员地址性能目标：平均响应时间<300ms，95%响应时间<500ms，单机TPS>500；

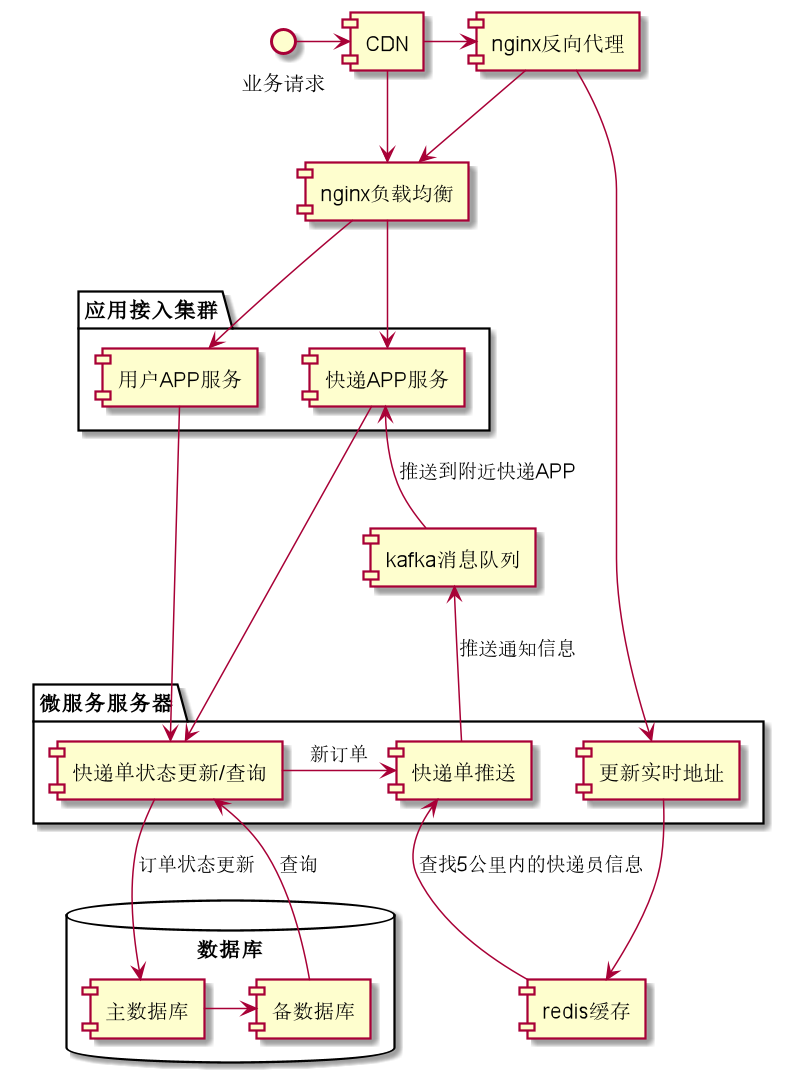
5. 系统核⼼功能可⽤性⽬标：>99.97%；

6. 系统安全性⽬标：系统可拦截XSS攻击、注入攻击、CSRF攻击，密码数据散列加密，客户端数据HTTPS加密，外部系统间通信对称加密；

7. 数据持久化⽬标：>99.99%。

## 2 系统部署图与整体设计

### 2.1 系统部署图



### 2.2 系统的整体需求

1. 用户通过用户APP进行下单，支付。快递通过快递APP查询，抢单，更新订单状态。相关页面使用第三方CDN对静态文件缓存。相关服务通过nginx反向代理转发请求到服务器或者应用接入集群。

2. 应用接入集群为用户APP和快递APP提供后端处理服务。

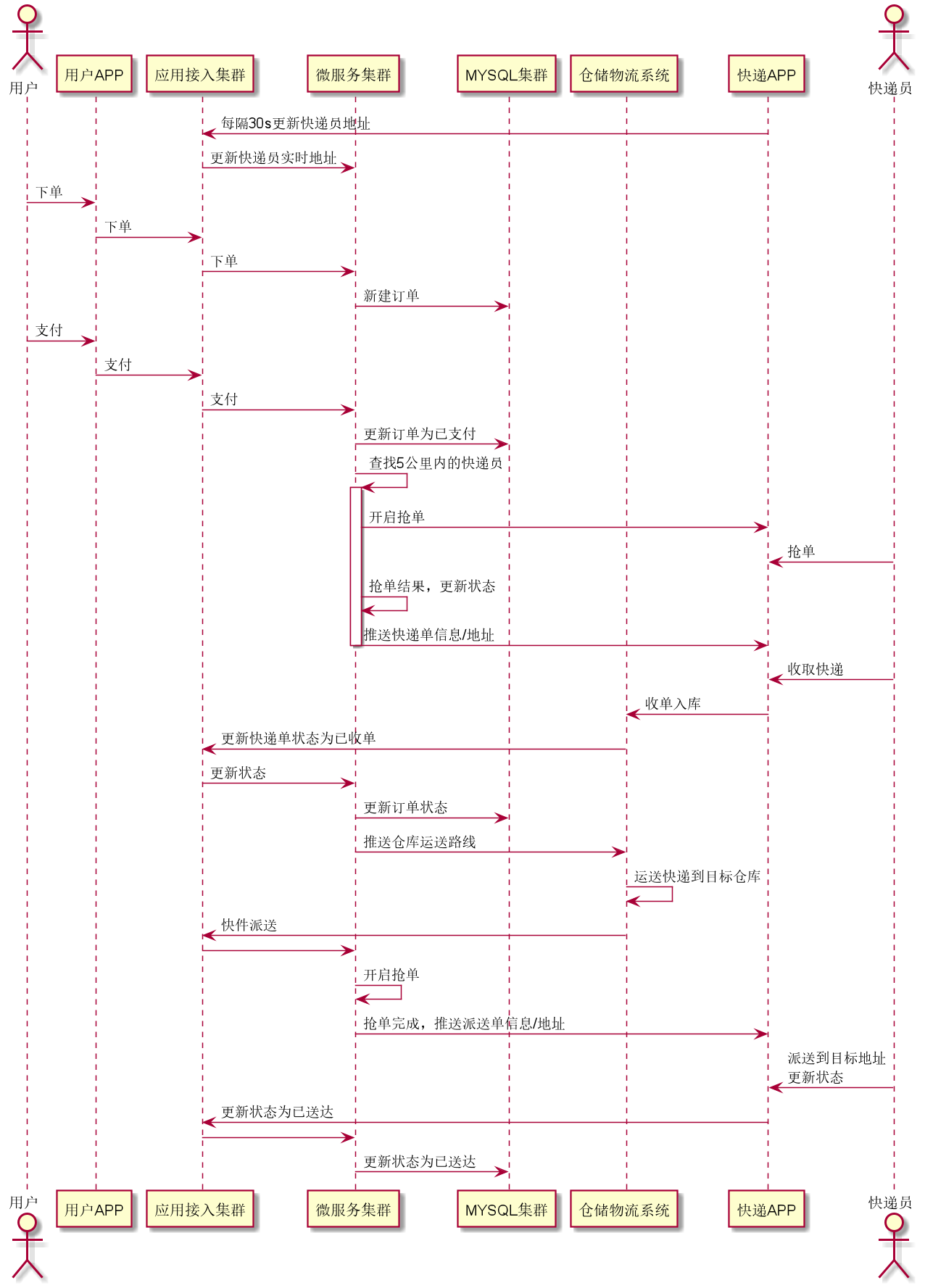
3. 微服务“更新实时地址”：快递APP每30秒上报一次快递员实时地址，实时地址信息在redis中缓存。Redis的存储结构设计为：Key值：城市-区

Value值为hash结构，保存内容：快递员Id，快递员实时地址。

4. 微服务“快递单推送”：订单支付完成后，从redis缓存中获取附近快递员实时地址，计算出5公里内快递员列表，将用户地址通过kafka消息队列向快递员APP推送抢单通知。

5.微服务“快递单状态更新/查询”：提供订单状态更新服务，状态更新写主数据库； 而订单状态查询则从备库中读取。

### 2.3 下单抢单场景序列图



### 2.4 订单状态图

