

Car-Rental

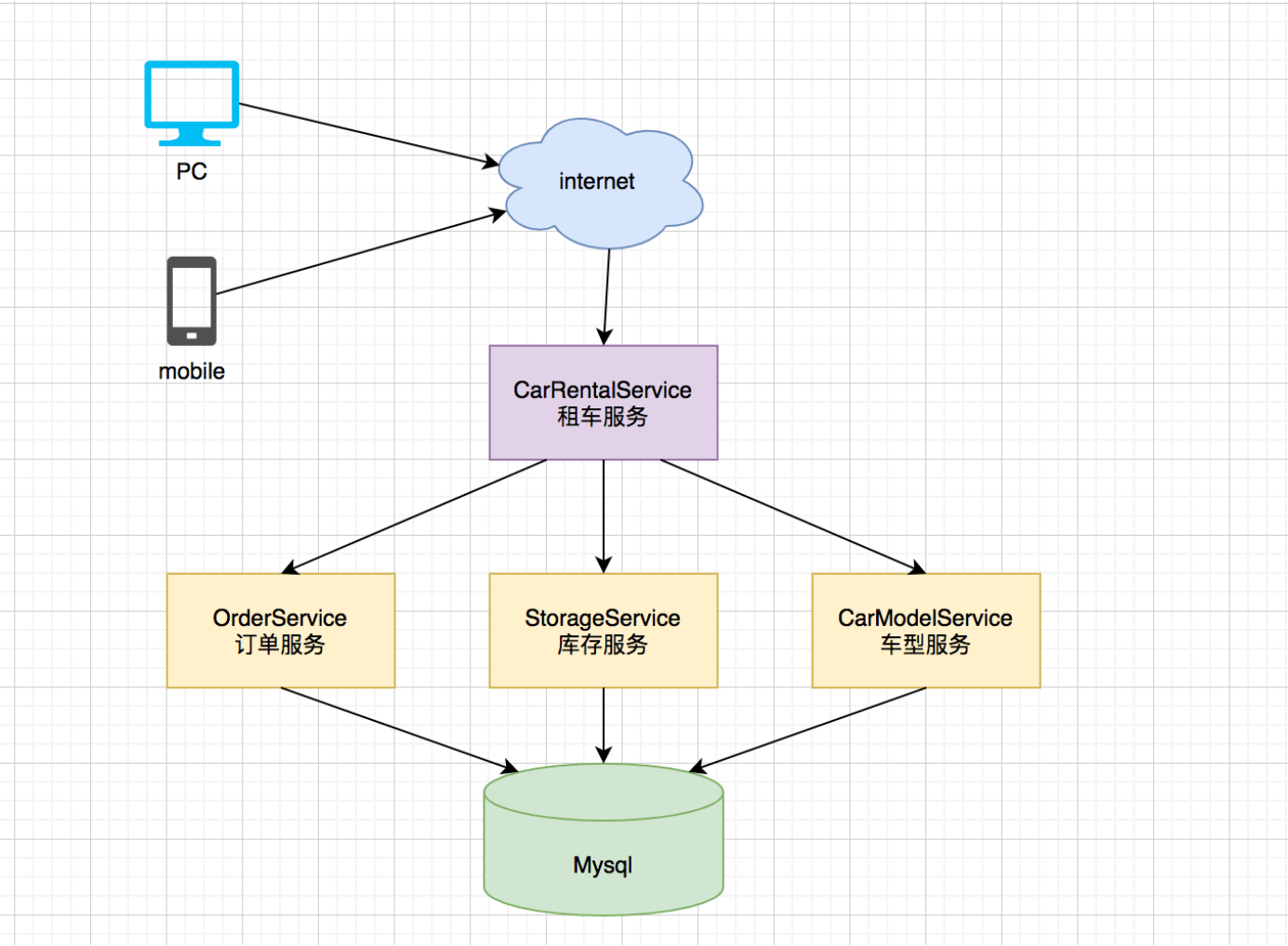
1. 概述

这是一个租车服务demo，包括的基本功能有车型预览，预约下单，我的订单查看。

代码仓库见：<https://github.com/moyuer91/Car-Rental>

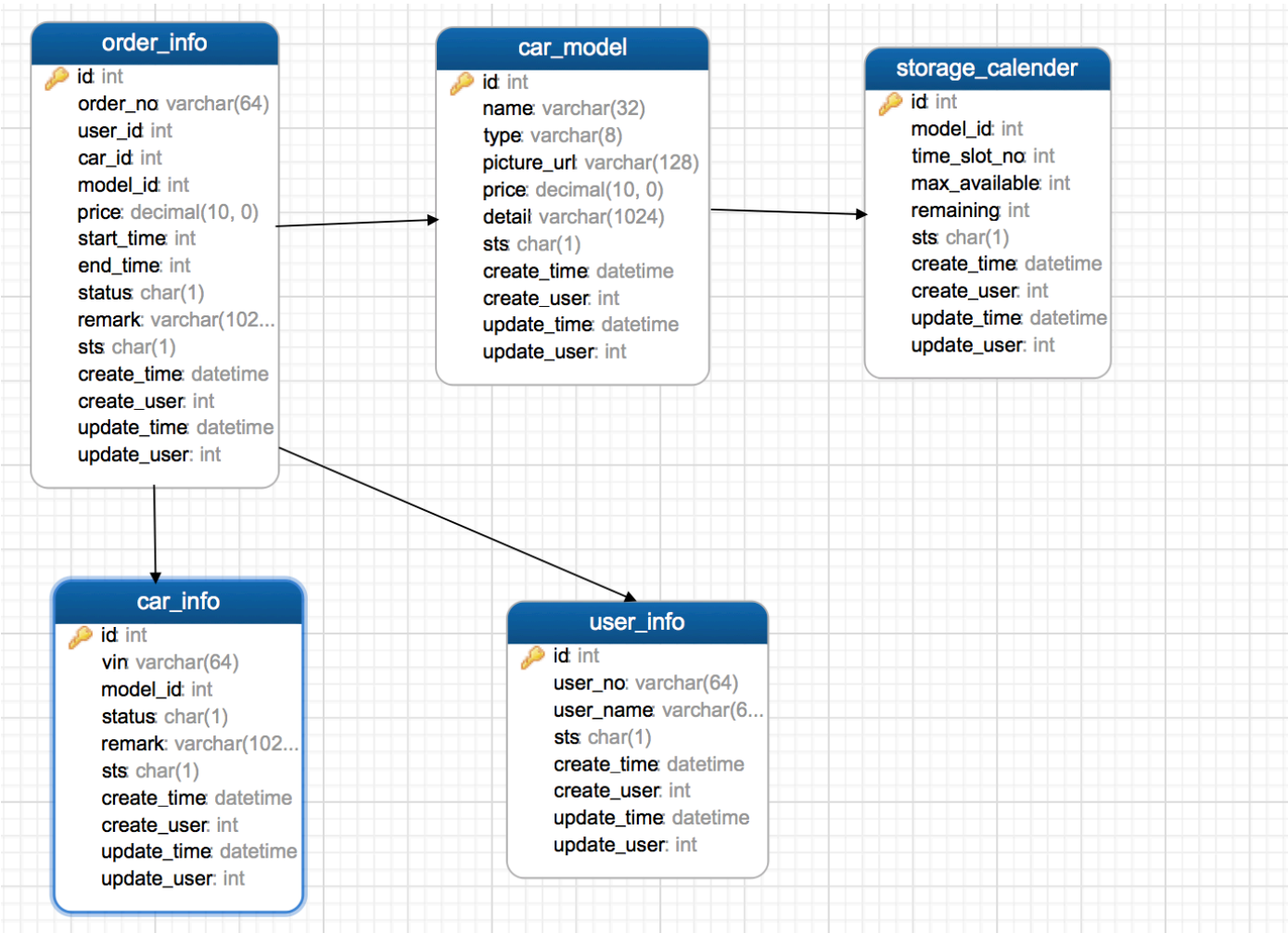
2. 系统架构

2.1 功能模块



- CarRentalService：租车服务，作为租车服务的统一入口，调用订单服务，库存服务，车型服务完成租车功能。
- OrderService：订单服务，完成订单创建和取消等功能。
- StorageService：库存服务，管理车辆信息，以及库存日历，每日定期根据当日库存情况开放新的库存日历，运营人员也可车辆情况调整库存日历。
- CarModelService：车型服务，管理了客户租车页面展示的可租车型列表，以及车型的详细情况，价格等信息。运营人员可根据库存情况来调整车型信息。

2.2 数据库设计



1. order_info: 订单表
2. car_model: 车型信息表，保存车型信息如toyota camery
3. car_info: 车辆信息表，保存了租车公司的所有库存车辆信息，作为库存日历生成的依据。
4. storage_calender: 库存日历表，根据当前的库存情况生成未来一段时间内可用车辆日历。
5. user_info: 用户表，仅保存简单的用户编号和名称

2.3 接口设计

使用swagger进行接口登记管理，具体可以查阅<http://localhost:8801/swagger-ui.html#/>

3. 功能模块设计

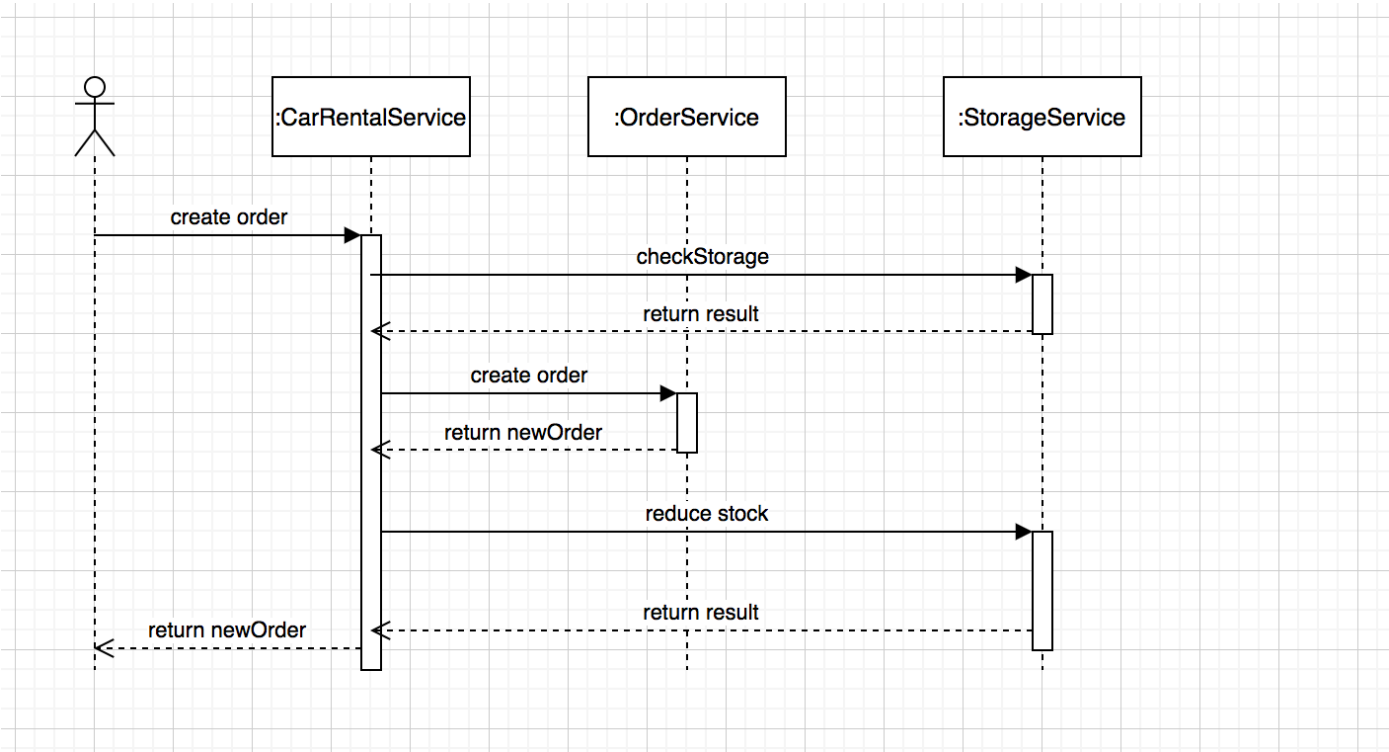
3.1 车型预览

3.1.1 流程设计

3.1.2 实现要点

3.2 预约下单

3.2.1 流程设计



- 1. 提交下单请求到后台。
- 2. 校验对应车型下的车辆日历库存。
- 3. 如果订单期间存在一个时间槽库存不足，则提示库存不足。
- 4. 否则创建订单，扣减对应时间槽的库存。

3.2.2 实现要点

- 1. 库存校验
库存使用库存日历的形式进行管理，定期开放可预约时段的库存情况，创建订单时校验订单涉及的时间槽的库存情况，如果存在一个时间槽库存为0，则表示库存不足。否则表示库存充足。
- 2. 原子性
订单创建和库存扣减必须保持原子性，否则可能出现数据不一致的情况。本方案中使用可重入锁来保证原子性，为保证并发访问时不出现超卖现象，使用切面进行上锁和解锁，同时该切面的order要在spring事务切面前面。

4. 非功能性需求

暂无设计