车票系统

1. 设计图



1. 待完善以及优化方案
2. 没有使用数据库，只是使用了列表作为数据源
3. 现在的网络系统，很多都是一个后台为多个前台（传统的 web、移动端）服务，因此可以使用 webserice、wcf、webapi（现在比较流行）。
4. 请求并发量增大的时候，增加web服务器的数量，使用ngix将请求分发到到不同的服务器上
5. 数据库压力比较大的时候，可以考虑优化语句、优化索引、分库、分表
6. 算法
7. 每个站点设计一个作为起始站卖出票的数量，一个作为**途径站**被卖出票的数量，一个座位终点站卖出票的数量
8. 每个站点可售票的最大数量不能超过座位数量

* **途径站：假设有站点S1、S2、……、Sn-1、Sn、如果被售出的票是S2->Sn-1，那么S2与Sn-1之间的站均为途径站**
* **可售票的最大数量，a. 如果为起始站或途径站，最大数量=作为起始站点售票量+最为途径站点售票量 b. 如果为终点站，最大数量=作为终点站点售票量+最为途径站点售票量**

1. 关键点
2. 多用户查询票时，不需要对用户进行限制
3. 多用户提交购票信息时，为了防止同时抢购一种票（始发站、途径站、终点站，有部分一直的票），导致库存异常，因此在提交购票信息是增加了 lock