# 1 系统测试

## 1.1 测试方式

因为本系统的体积过于庞大，所以系统测试大多是在开发过程中进行的，而且大多数都采用了手动操作测试的方式来进行。

## 1.2 测试结果

在开发的过程中有遇到过一些系统问题，但是在一段时间的DEBUG之后本系统当前无BUG运行正常，各个模块的功能均可以正常的运行，且能够达到预期的运行效果。

# 2 体会与自我评价

## 2.1 技能提升

通过这次课程设计的实践，自己加深了对于之前很多知识的理解，这次的实践是对自己之前所学习过的知识的一次总结和应用，同时也让自己学习到了更多新的技术。在这次课程设计中，自己独立完成了数据库系统的需求分析、数据库概念结构设计、数据库逻辑结构设计、数据库物理结构设计等一些流程的开发，大幅度的提升了自己的编程能力、独立思考能力、软件开发能力等一些综合素质。

在系统的开发中，我首次实践了之前所学习过的Spring + SpringMVC + MyBatis组合框架来进行系统的开发，在这套开发模式中，Spring提供了IOC的系统容器，SpringMVC提供了结构更加清晰、更易维护的MVC开发架构模式，MyBatis提供了封装JDBC后更易操作的数据库操作方式。

在之前的学习过程中，自己更多的就是读一些相关的资料和书籍，或者做一些小demo来进行实践，但其实在这个过程中自己对这个技术的掌握是不到位的，大多数情况下是仅仅限于了解的，古话说得好“纸上得来终觉浅，绝知此事要躬行”，实践是获得真知的唯一途径，许多的技术只有当你真正的亲身去实践过了，才会有更加深入的理解。因此这次的系统开发给了我一个很好的机会对其进行实践，而且对于这样较大规模的系统开发，在开发过程中会遇到许多在做小demo中没有遇到过的问题，在不断的遇到问题，解决问题的过程中使自己对于这项的技术又更深了一步。

同时在这次的系统的开发中，我也通过SpringMVC来使用了MVC的这种架构模式，其实之前在做安卓开发的时候，也有了解过MVC、MVP和MVVM这三种开发模式，并在一个较大规模的安卓开发过程中实践使用了MVP这种架构模式，并在小程序的开发过程中实践过MVVM这种架构模式，但是对于后端开发，使用MVC这种架构模式来进行开发还是第一次，因此这次的系统开发也是自己对于MVC架构模式的又一次深入探索与学习。

在这个系统的开发过程中，我是秉承一种大规模的系统开发的思想来进行开发，并集中的考虑了关于并发问题、容灾问题和数据读并发读写性能等一些高并发场景下的问题。在购物车的设计过程中，我的第一个版本是使用数据库来直接进行存储的，每次购物车状态的更新，我都会同步到数据库。但是在后来的思考和资料查询的过程中，我发现这样的方式在高并发的场景下存在着巨大的性能瓶颈，因此我选择了使用Redis内存来进行购物车的数据存储，并在特定的时间来使用单独的工作线程将Redis内存中数据同步至数据来进行持久化存储，保证用户在每次退出登录后再重新登录时可以再次获取到自己数据库中的数据，同时也保证了当主机发生了意外关闭或其他重启现象而清空内存时，用户的购物车信息也能够被持久化存储至数据库中。

针对商品列表和商品和商品详情这一系列页面的加载，我也采用了Spring集成Redis的方式来进行优化，通过测试可以发现没有优化之前，商品的信息直接从数据进行获取时每次操作的时间大概在500ms左右，而这还只是在数据量较少的情况下，那如果当数据量高达至上万时，这样操作的延迟会给用户带来很不好的体验。但是当我使用Redis内存进行优化后，成功的将这一操作的延迟降低到10ms~40ms性能优化进百分之几百，页面的相应速度明显得到大幅度的提升。

图7.1.1

上图即该操作的整个流程，通过Spring对于Redis的集成，我们在每次需要获取商品信息之前都先到Redis内存中进行查找，如果Redis内存中存在的话则直接返回数据，不存在时我们就到数据库中去进行查询，并将查询后的数据返回同时保存至Redis内存中，当管理员在后台进行修改商品信息的操作时，我们就同步的更新Redis内存中的数据。

我之所以想到这种方式来对数据存储进行优化，其主要原因就是对于商品详情这个数据查询的次数是明显的多于修改的次数的，所以我们可以将其保存至Redis内存中，以对内存的读取来代替对于文件的读取。

对于商品的定时上架功能，我采用了Spring集成Quartz的方式来进行实现，并通过管理员设置的上架时间来动态的生成CronExpression表达式，并将生成的任务动态的添加至Scheduler中进行管理。而我之所以选择Quartz而没有选择Timer的主要考虑是，Quartz相对来说使用更加的灵活，同时它是以多线程并发的方式来进行任务的执行，也就是当多个任务同时需要执行时，他会采用多开多个线程的方式来并发的进行执行，而对于Timer则是单线程的来进行的，会出现阻塞操作。因此使用Quartz可以大幅度的提升系统的性能。

同时对于用户注册和图片文件上传等一些可以进行异步操作的功能模块，我都使用RabbitMQ消息队列方式来进行异步的执行，通过这样的方式可以大幅度的提升系统的性能同时提升用户的使用体验。

## 2.2 能力提升

通过这次的系统开发，我觉得对于我最大的提升除了技术方面，更多的是在能力方面的提升。在开发的过程中，不断的遇到问题，遇到挫折，解决问题，收获成功的喜悦。在开发的过程中养成了独立思考、发现问题和解决问题的能力。

在开发的过程中，无数次的在漆黑的夜晚遇到莫名其妙的BUG，开始的时候整个人都会很焦躁不安，很惶恐，手足无措不知道怎么去解决它，但是随着开发的进行，经验的积累，自己解决问题的能力逐渐上升，DEBUG的能力也不断的提升，遇到问题的时候更加的冷静，能够在更短的时间里定位问题、发现问题并在思考后解决问题，我觉得这才是对于最大的影响。

同时在开发的过程中，随着自己学习能力的不断提升，面对新的技术时有了更好的学习方法，学习的效率也越来越高。