# 爱旅行项目总结

1. 项目背景

随着人们生活水平日益提高和闲暇时间增多，旅游业正成为当今世界最具生机活力、永不衰落的“朝阳产业”，成为经济增长的新“亮点”。改革开放以来，我国旅游业已经实现了连续20多年的持续、快速发展，旅游业在国民经济中的地位和作用日益增强，国内旅游人数和效益指标有了百倍以上的增长，国际旅游各项指标比1978年有了四五十倍以上的增长，实现了从“资源大国”到“亚洲旅游大国”的历史性跨越。随着小康社会的到来，我国旅游发展格局和态势正在发生着深刻地变化，从“观光时代”向“休闲度假时代”转变，人们的休闲时间也越来越多，传统工作和休闲概念已经逐步地模糊，人们也更加注重文化精神的消费与追求，这种变化大大刺激了休闲度假:旅游的发展，休闲消费已经成为我国旅游业新的增长点。为了提高消费者的体验度和满意度，遂开发一款集出行，住宿，吃喝，玩乐于一体的优秀软件

1. 使用技术
2. 聚合工程

对于一个大型的项目，如果我们直接作为一个工程开发，由于相互之间的依赖我们只能从头到尾由一组人开发，否则就会出现一个类好多人开发，相互更改的混乱局面，这个时候我们就将项目进行了横向和纵向的拆分。

所谓的横向的拆分就是我们平常说的三层架构，将项目分成了controller层，service层、dao层，可以理解为将一个功能模块的不同调用过程进行了水平方向的拆分。

所谓的纵向拆分就是将一个项目的多个功能模块进行了，可以理解为为了完成一个系统，深度分析需要有哪些功能，然后将这些功能独立出来，进行了拆分。

横向拆分后，每个功能模块进行了单独的开发之后，项目整合的时候就需要有一个能够整合这些项目或者模块的工程。

1. SSM框架开发（SpringMVC-Spring-MyBatis）

Spring

通过Spring的IOC特性，将对象之间的依赖关系交给了Spring控制，方便解耦，简化了开发

通过Spring的AOP特性，对重复模块进行集中，实现事务，日志，权限的控制

提供了对其他优秀开源框架的集成支持

Spring MVC

SpringMVC是使用了MVC设计思想的轻量级web框架，对web层进行解耦，使我们开发更简洁

与Spring无缝衔接

灵活的数据验证，格式化，数据绑定机制

Mybatis

        数据库的操作(sql)采用xml文件配置，解除了sql和代码的耦合

        提供映射标签，支持对象和和数据库orm字段关系的映射，支持对象关系映射标签，支持对象关系的组建

        提供了xml标签，支持动态的sql

1. Git版本管理工具

1、适合分布式开发，强调个体。   
2、公共服务器压力和数据量都不会太大。   
3、速度快、灵活。   
4、任意两个开发者之间可以很容易的解决冲突。   
5、离线工作。 

1. Maven工程

1、原来的项目中需要的jar包必须手动“复制”、”粘贴” 到WEB-INF/lib 项目下，而借助Maven，可以将jar包仅

     仅保存在“仓库”中，有需要使用的工程只需要“引用”这个文件，并不需要重复复制到工程中。

2、原来的项目中所需要的jar包都是提前下载好的，而Maven在联网状态下会自动下载所需要的jar包。首先在本地

     仓库中找，找不到就在网上进行下载。

3、原来的项目中一个jar包所依赖的其他jar包必须手动导进来，而Maven会自动将被依赖的jar包导进来。

4、原来的项目一个项目就是一个工程，而借助Maven可以将一个项目拆分成多个工程。

1. Solr

Solr是一个全文检索服务器，只需要进行配置就可以实现全文检索服务。有效降低频繁访问数据库对数据库造成的压力。

1. Redis

Redis是一个基于内存的高性能键值对数据库，每秒可以出来超过10W次读写操作，是已知的性能最快的Key-Value DB，使用Redis缓存经常使用但不频繁改变的数据，提高用户体验度

1. 短信、邮箱验证
2. 支付宝支付
3. 使用ngrok进行内网穿透
4. 前后端分离开发

通过Nginx反向代理工具与前端代码整合在一起

1. swagger文档测试
2. 使用MD5进行加密
3. 项目流程

（Auth）

邮箱注册:

在进行邮箱注册时，用户输入正确的邮箱，经过验证邮箱格式正确之后，向服务器发送请求，然后以邮件的形式将激活码发送到输入邮箱账号上，用户根据激活码进行激活完成注册。

（激活码<30分钟有效>通过MD5加密然后通过Sping Mail API发送至用户邮箱，同时保存在Redis中，用户进行激活时，调用RedisAPI进行查询，一致则成功否则失败）

手机注册:

在进行手机注册时，用户输入正确手机号码，经过验证手机格式正确之后，向服务器发送请求，然后通过第三方接口发送短信验证码，用户根据验证码完成激活注册

（验证码<1分钟有效>通过第三方平方发送，先保存在Redis中，用户进行激活时，调用RedisAPI进行查询，一致则成功否则失败）

登录:

根据用户输入的用户名和密码进行验证，通过验证则生成Token,并且保存在Redis中，为了保证其一致性，不同客户端生成不同的Token，否则登录失败

PC端：token：PC-用户名[MD5(32位)]-用户ID-当前时间-客户端信息[MD5(6位)] 保存两个小时，若不置换，两小时后自动清除Token

移动端：token：MOBILE-用户名[MD5(32位)]- 用户ID -当前时间-客户端信息[MD5(6位)] 永久保存，修改密码时更换Token

然后在接口中保护Token生成时间和失效时间

注销:

用户验证Token是否有效，在Redis中查看Token是否存在，存在将其remove;

1. 项目总结

因为是第一次接触这种分布式聚合项目，刚开始开发非常不顺利，首先开发之初创建工程就耗费了大量的时间，一些细节没注意总是在重复创建工程，然后开发过程由于分布式项目我们看不到前台页面，所以测试的时候非常不适应，经过老师的讲解，自己的摸索慢慢熟悉swagger测试工具，然后慢慢步入正轨，项目中也用到了大量的新技术点，也都只是大概了解皮毛，还需对技术点进行深刻的学习。总之，这个项目有非常大的收获，同时也看到了自己学习上的不足，今后要继续努力。