

浙江大学



软件工程项目实训课程项目
旅游信息攻略查询搜索引擎
——项目总结报告

小组：G19

组长：薛伟

组员：梁杰、梁宇坤、陈立华

2018/07/26

目录

1. 引言	4
1.1 编写目的	4
1.2 项目背景	4
1.3 定义	5
1.4 参考资料	5
2. 实际开发结果	6
2.1 产品	6
2.1.1 程序整体概述	6
2.1.2 程序版本	6
2.1.3 文档列表	7
2.2 项目成果展示	7
2.2.1 登录界面	8
2.2.2 注册界面	8
2.2.3 搜索主页	9
2.2.4 搜索结果——攻略	9
2.2.5 搜索结果——路线	10
2.2.6 搜索结果——全部	10
2.2.7 路线详情	11
2.2.8 攻略详情	11
2.2.9 推荐列表	12
2.3 基本流程	12
2.3.1 搜索用户流程图	12
2.3.3 管理员交互流程图	13
2.4 进度	13
2.4.1 原定计划	13
2.4.2 实际进度	14
2.4.3 对比分析	15
2.5 费用	15

3. 开发工作评价.....	16
3.1 对生产效率的评价.....	16
3.2 对产品质量的评价.....	16
3.3.1 软件缺陷统计.....	16
3.3.2 用例执行情况.....	17
3.3 对技术方法的评价.....	18
3.3.1 Spring	18
3.3.2 Elastic Search.....	19
3.3.3 白盒测试	20
3.3.4 黑盒测试	20
3.4 出错原因分析	21
4. 小组成员工作总结	22
4.1 工作评价	22
4.2 组内评分细则	22
4.3 组内互评结果	23
5. 经验与教训.....	23

1. 引言

1.1 编写目的

随着项目实训课程来到了尾声，项目实训课程组提出的、由我们第 19 小组负责开发的旅游信息攻略查询搜索引擎项目已经全部完成。这份总结报告详细阐述了我们开发的系统的主要功能和性能，并对整个开发过程进行了总结与评价，由此得出经验与教训，以期对项目组未来的各项工作产生指导作用。

预计读者对象：项目组所有成员，课程指导老师，项目评审人员。

1.2 项目背景

➤ 待开发的软件系统的名称：

旅游信息攻略查询搜索引擎

➤ 项目的任务提出：

浙江大学软件工程项目实训课题组

➤ 开发者：

软件工程项目实训 2 班第 19 组 4 人

➤ 用户：

搜集旅游攻略的旅游者

旅游信息检索人员

➤ 实现该软件的计算中心或计算机网络：

基于阿里云的云服务器，可能需要用到多服务器节点

➤ 该软件系统同其他系统或其他机构的基本的相互来往关系：

可需要从多个旅游门户、攻略网站爬去旅游攻略及信息

由项目实训所有小组及老师共同进行评审工作

➤ 相关背景介绍

旅游信息攻略查询搜索引擎的构想主要来自于“垂直搜索引擎”的课程要求，同时由于当前市场上多个旅游网站信息比较混杂，内容也比较凌乱，提供一个较为统一的搜索引擎，同时对信息进行一定的归类、关联和分析，可以方便旅游信

息检索人员、做旅游攻略的人员检索旅游信息，提升工作的效率，获得使用上的方便。

1.3 定义

MySQL: 系统服务器所使用的数据库关系系统 (DBMS)。

SQL: 一种用于访问查询数据库的语言。

Java: 面向对象的编程语言，用以开发平台的后端程序。

UML (Unified Modeling Language): 统一建模语言、是一套用来设计软件蓝图的标准建模语言，是一种从软件分析、设计到编写程序规范的标准化建模语言。

JavaScript: 一种面向对象的动态类型的客户端脚本语言。

ElasticSearch: 新一代非关系型索引数据库。

Apache: web 服务器软件，跨平台，安全性较高的流行 web 服务器软件之一。

MD5 加密算法: 用以提供消息的完整性保护。

1.4 参考资料

- 软件设计文档国家标准操作手册 (GB8567——88)
- 《G19-项目开发计划》
浙江大学 2017-2018 暑期短学期项目实训课程小组—G19
- 《G19-软件需求规格说明书》
浙江大学 2017-2018 暑期短学期项目实训课程小组—G19
- 《项目实训工作描述(案例二)》
浙江大学 2017-2018 暑期短学期项目实训课程小组—G19
- 《软件需求》
Karl Wieggers, Joy Beatty

2. 实际开发结果

2.1 产品

2.1.1 程序整体概述

本程序名为旅游信息攻略查询搜索引擎,用户类别分为信息查询者和网站管理员两种。

其中信息管理员可以可以进行简单的关键词查询,查询包含对应名词的旅游路线,另外也可以通过模糊查询根据某些条件来进行旅游路线的筛选,另外可以根据网站提供的综合统计数据进行检查和分析对比,同时会对用户的信息查询行为进行搜集,构建出个性化推荐/推送的内容策略。

而管理者对网站动态爬取到的数据进行查看和管理,监测网站的运行状态,管理网站的内容和服务器运行状态。调整网站的各项功能参数等。

2.1.2 程序版本

时间	版本	描述
2018.07.21	1.0	系统的第一个版本,完成了大部分功能需求,但界面不够友好,而且 bug 较多,还有一部分重要功能没实现
2018.07.22	1.1	这个版本是在首次测试之后开发出来的,这个版本大幅度修改了前端页面,使页面更加美观,同时完善了各种细节,修复了大部分 bug,并完成了

		需求变更访谈中双方达成一致的新功能
2018.07.24	1.2	修复了部分不明显的、在测试阶段没有查出来的bug

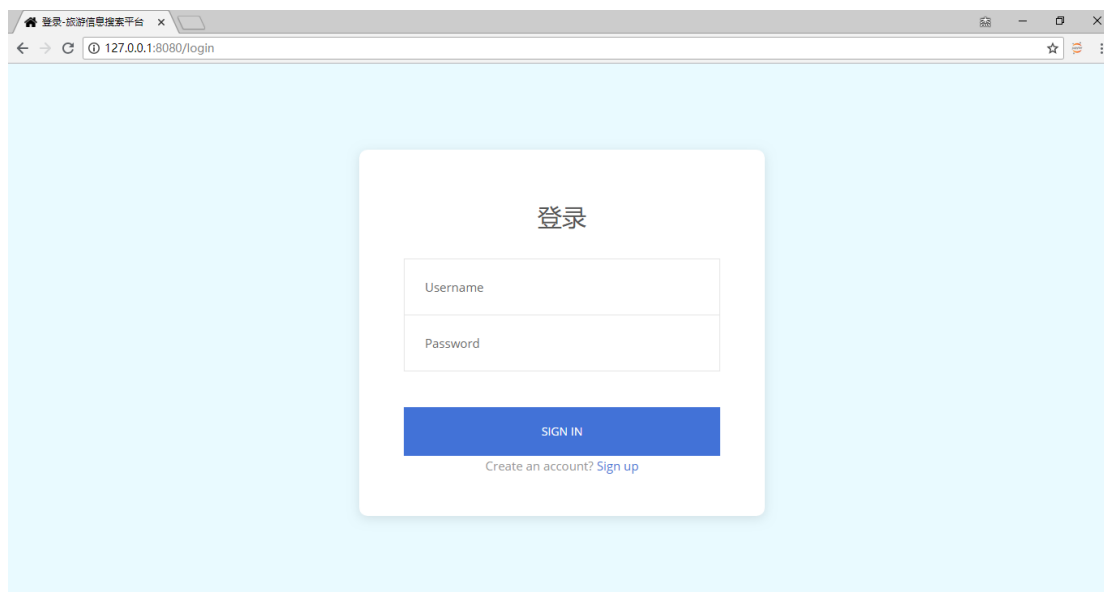
2.1.3 文档列表

- 《项目开发计划》
- 《软件需求规格说明书》
- 《系统详细设计说明书》
- 《系统测试计划》
- 《系统测试报告》
- 《项目总结报告》
- 以及各个阶段工作日报和会议纪要

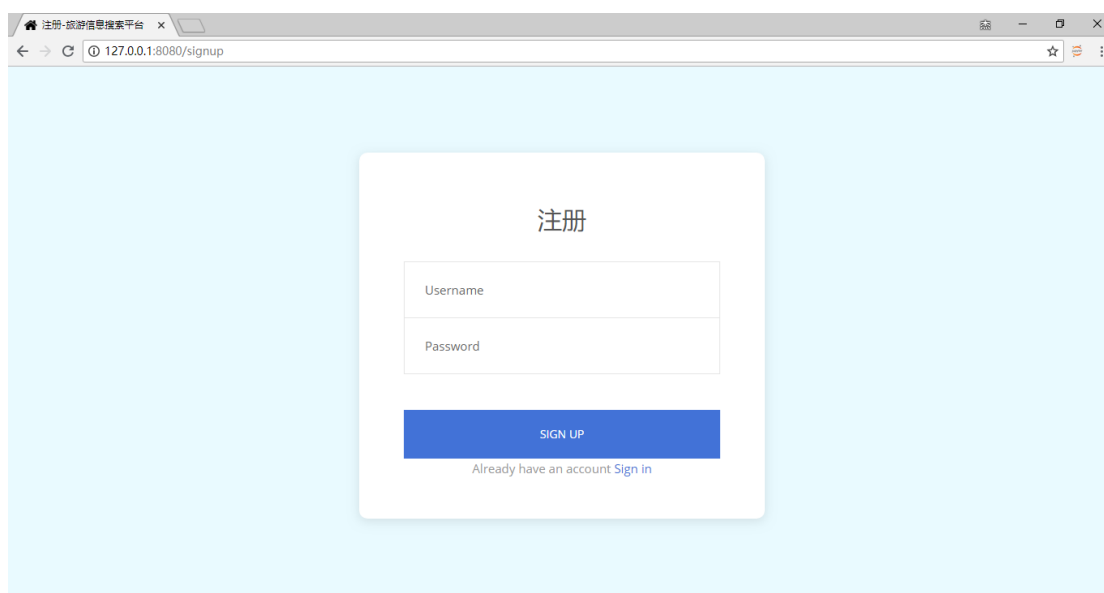
2.2 项目成果展示

本总结报告只列出所实现的功能的界面信息，详细的处理流程见《系统详细设计说明书》

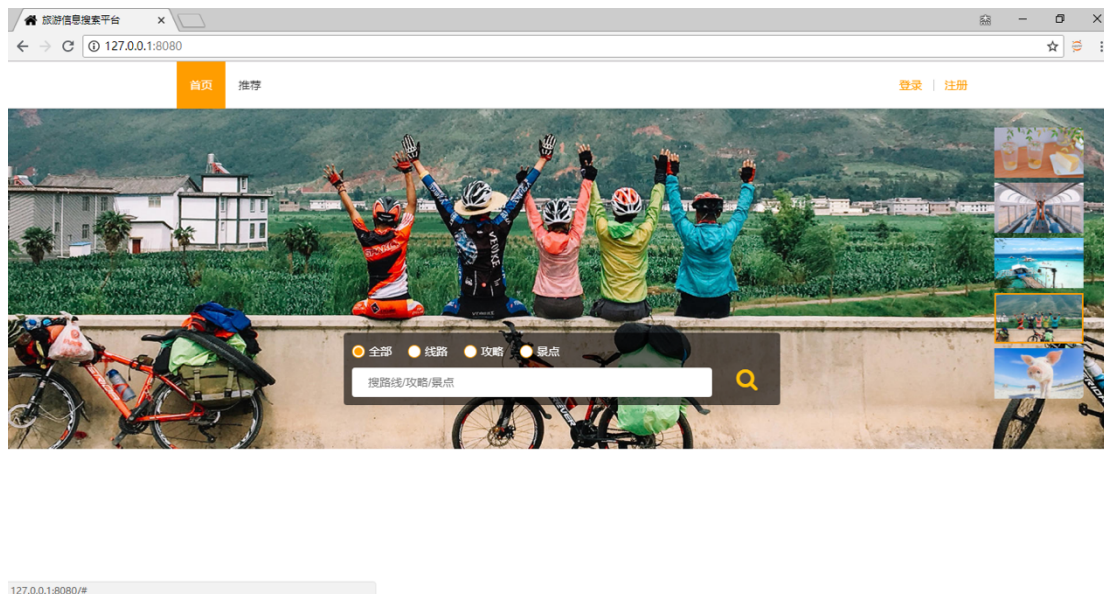
2.2.1 登录界面



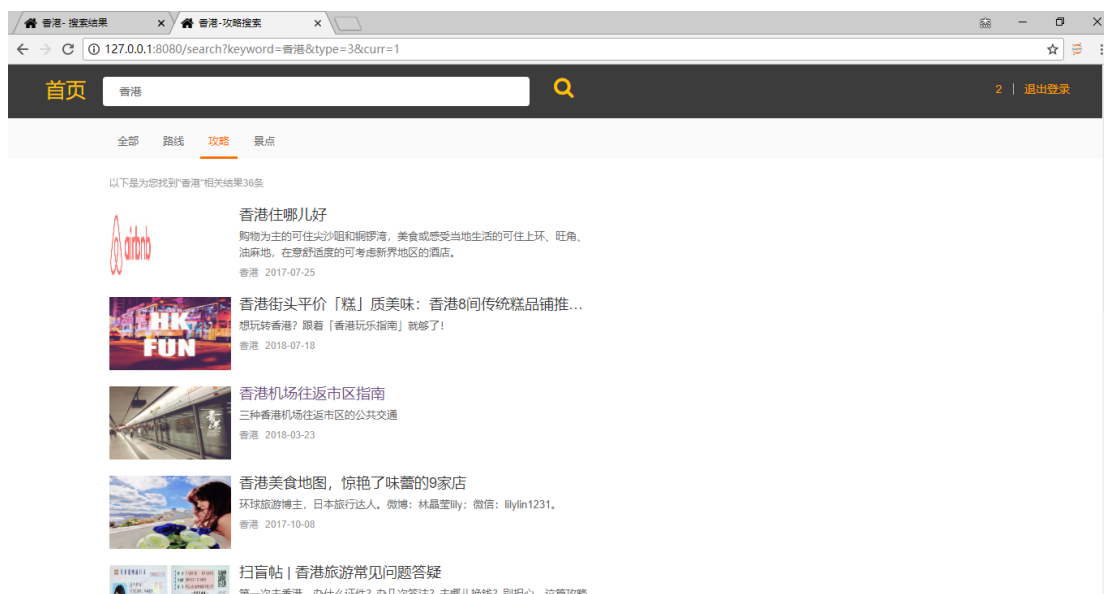
2.2.2 注册界面



2.2.3 搜索主页



2.2.4 搜索结果——攻略



2.2.5 搜索结果——路线

[全部](#)[路线](#)[攻略](#)[景点](#)

以下是为您找到“香港”相关结果1条



即订即用 香港迪士尼乐园电子门票（实时出票+可选餐券+无需打印）

[扫码入园](#)

[咨询客服立享优惠](#)

[非指定日使用](#)

¥ 463~1844 起

[查看详情](#)

[上一页](#)[1](#)[下一页](#)

2.2.6 搜索结果——全部

香港 - 搜索结果

127.0.0.1:8080/search?keyword=香港&type=1

首页 香港


2 | 退出登录

[全部](#)[路线](#)[攻略](#)[景点](#)


以下是为您找到“香港”相关结果37条

旅游攻略


[查看更多旅游攻略 >>](#)




香港住哪儿好
购物为主的可住尖沙咀和铜锣湾，美食或享受当地生活的可住上环、旺角、油麻地，在意舒适度的可考虑新界地区的酒店。
香港 2017-07-25



香港街头平价「糕」质美味：香港8间传统糕点铺推...
想玩转香港？跟着「香港玩乐指南」就够了！
香港 2018-07-18




香港机场往返市区指南
三种香港机场往返市区的公共交通
香港 2018-03-23




香港美食地图，惊艳了味蕾的9家店
环球旅游博主，日本旅行达人。微博：林晶莹lily；微信：lilylin1231。
香港 2017-10-08

附近热门景点


[全部景点 >>](#)




湖心亭




南莲阁




COSTA COFFEE[...]




COSTA COFFEE[...]




COSTA COFFEE



阮泉环碧



湖天一碧



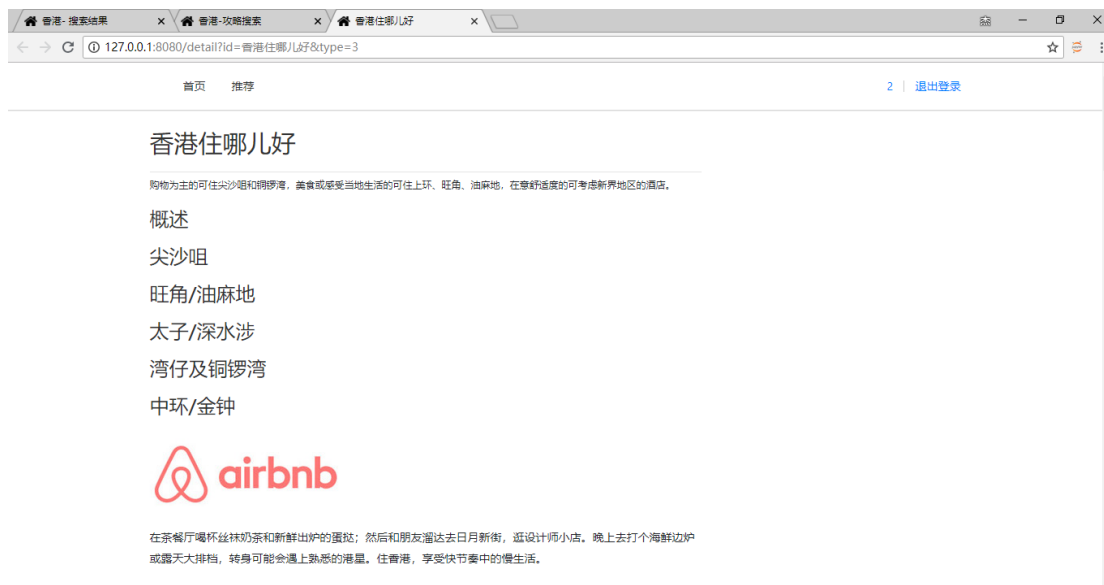
中央美术学院华东...

10 / 24

2.2.7 路线详情



2.2.8 攻略详情

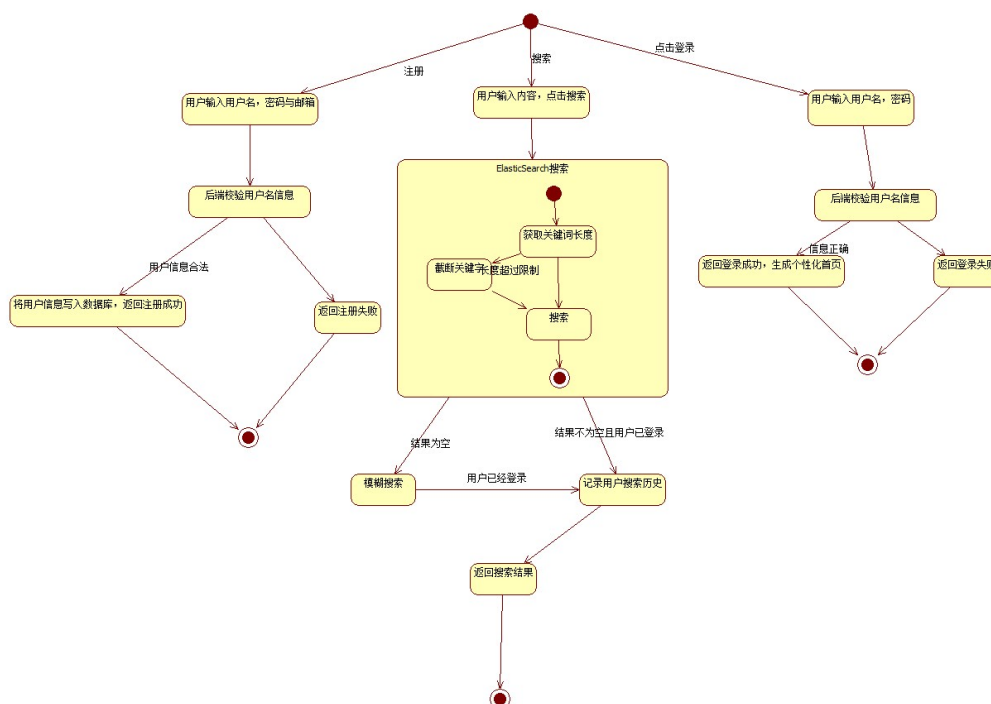


2.2.9 推荐列表

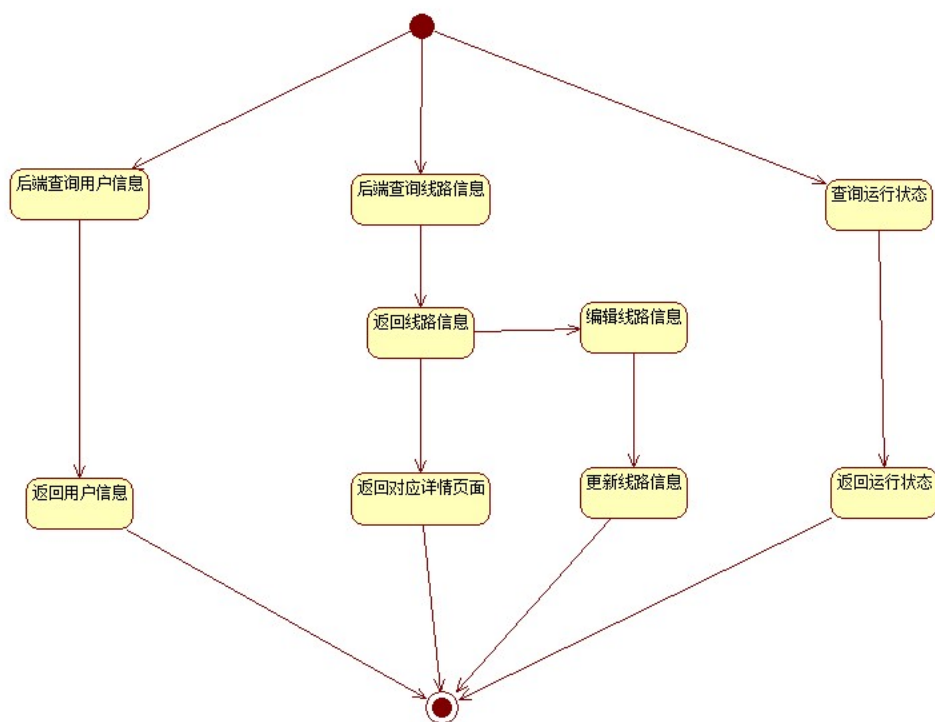


2.3 基本流程

2.3.1 搜索用户流程图



2.3.3 管理员交互流程图



2.4 进度

2.4.1 原定计划

项目阶段	开始时间	结束时间	所需资源
项目启动及撰写 开发计划	2018. 07. 12	2018. 07. 12	项目指导文件
需求分析	2018. 07. 13	2018. 07. 14	需求分析形式、网 站数据查询
系统设计	2018. 07. 15	2018. 07. 16	需求分析、网站构 建知识、服务器环 境、数据库环境
代码编写	2018. 07. 17	2018. 07. 23	《软件设计说明 书》

测试计划	2018.07.21	2018.07.23	《软件需求规格说明书》、《软件设计说明书》、《项目开发计划》
测试分析	2018.07.24	2018.07.25	各模块设计报告、各模块编码、测试环境
项目总结	2018.07.26	2018.07.26	完整项目报告、完整项目代码

2.4.2 实际进度

项目阶段	开始时间	结束时间	主要工作产品
项目启动及撰写开发计划	2018.07.12	2018.07.12	《项目开发计划》
需求分析	2018.07.13	2018.07.14	《需求规格说明书》
系统设计	2018.07.15	2018.07.17	《系统详细设计说明书》
代码编写	2018.07.16	2018.07.25	项目源码
测试计划	2018.07.21	2018.07.22	《系统测试计划》
测试分析	2018.07.24	2018.07.25	《系统测试报告》
项目总结	2018.07.26	2018.07.26	《项目总结报告》

2.4.3 对比分析

从《项目开发计划》中的安排和实际进度表可以看出，项目的前半段我们基本是按照计划来进行工作的，但是到了项目中后期开始，整个节奏基本就跟《项目开发计划》的安排有些脱节了，分析原因如下：

- 在制定《项目开发计划》时，我们团队成员的开发经验不足，难以预估每项工作的工作量和难易程度，之前的项目都开发太过草率，因此这些经验基本无法对我们产生有意义的指导。这也就导致我们在制定《项目开发计划时》只能凭借手头仅有的小部分资料作为参考，导致拟出的计划不切实际；主要工作的难度还是在系统设计及代码实现上。
- 代码的挑战极限部分实现起来较为繁琐，而且没有硬性的衡量指标，在数据量较小的情况下难以对比分析结构，所以在这部分内容上耽误了不少的时间。
- 起初我们以为系统设计、详细设计是在编码之前就要完成的，因此将系统设计、详细设计的进度安排的较早，后来我们知道了前端开发与编写设计文档基本是同时进行的，因此提前了代码编写的时间而延后了编写设计文档的时间
- 由于开发人员经验不足，没有完全按照需求规格说明书来编写，虽然在意识到问题以后及时纠正、与原需求进行了比较，但仍然为整个项目的开发工作造成了一定困扰。
- 代码编写的时间比我们预计的要长很多，这不仅是开发人员未按照第一版的需求来开发程序的原因，另一个更重要的原因是在与老师沟通后发现我们的系统仍然存在较多的问题，因此在需求变更以后又进行了大幅度的功能性修改，也给整合工作造成了一定程度上的麻烦。

2.5 费用

本项目作为课程实践项目，并无费用支出，而且由于项目组部分成员拥有个人云服务器，因此原定的云服务器开支也省掉了。

3. 开发工作评价

3.1 对生产效率的评价

本项目文档编写虽然并未完全按照《项目开发计划》的计划来，但仍然完成了所有项目开发必须的文档的编写，且完成可以运行和测试的软件系统，并满足一定的质量规格标准。

本项目整体开发时间为 15 天，实际开发时间约为 10 天，初版程序开发与测试变更之后的开发各用了 5 天时间，但由于开发技术较新，同时涉及到许多没有学习过的知识，因此实际生产效率较低。

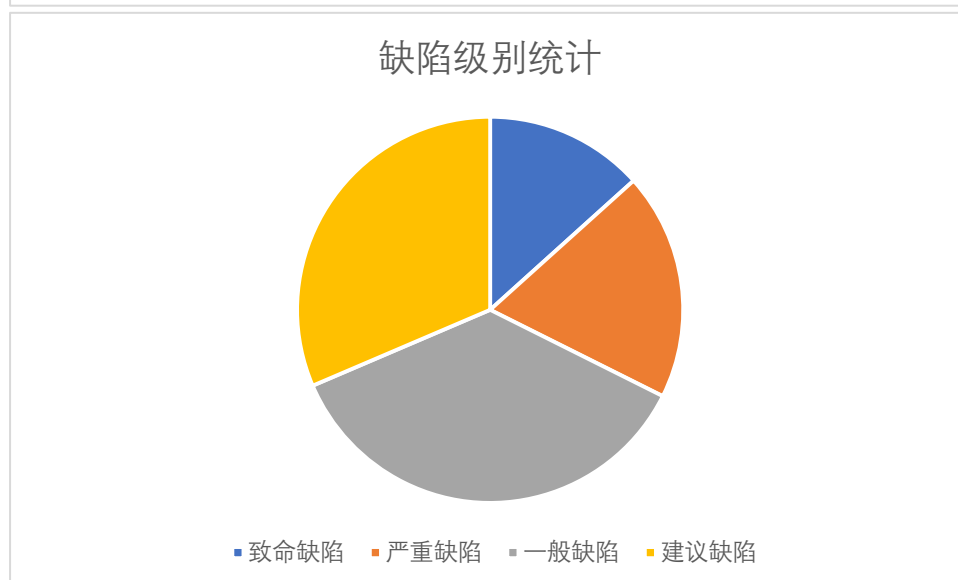
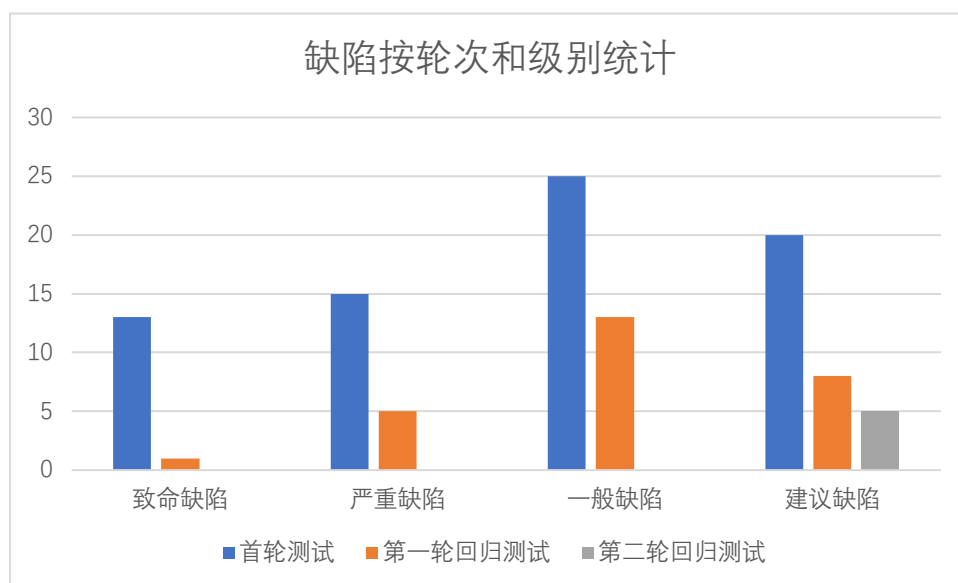
由于在前期制定计划时并未给出生产效率的估计，因此这里无法与计划进行对比。

3.2 对产品质量的评价

3.3.1 软件缺陷统计

缺陷按轮次和级别统计

测试阶段 缺陷级别	首轮测试	第一轮回归测试	第二轮回归测试	总计
致命缺陷	13	1	0	14
严重缺陷	15	5	0	20
一般缺陷	25	13	0	38
建议缺陷	20	8	5	33
小计	73	27	5	105



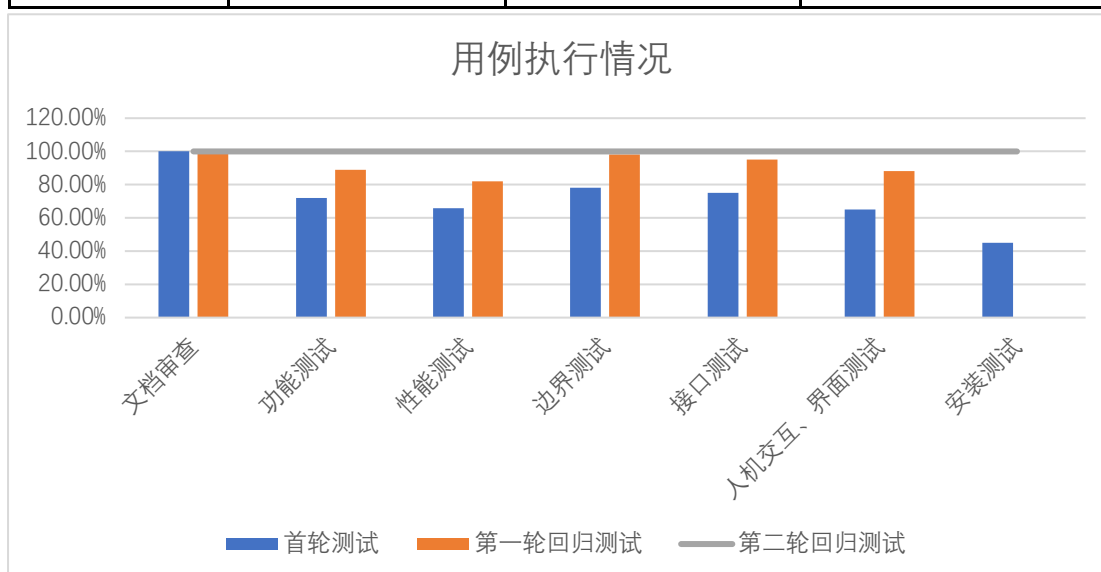
由软件缺陷统计结果可以看出，首轮测试中我们的系统尚存较多缺陷，经过开发人员的共同努力，在第二轮、第三轮回归测试中缺陷数量明显减少。由第三轮测试结果可见，我们的系统缺陷已经几乎没有，系统可以发布。

3.3.2 用例执行情况

用例执行情况

测试阶段 测试类型	用例执行情况		
	首轮测试	第一轮回归测试	第二轮回归测试
文档审查	100%	100%	100%

功能测试	72%	89%	100%
性能测试	66%	82%	100%
边界测试	78%	98%	100%
接口测试	75%	95%	100%
人机交互 界面测试	65%	88%	100%
安装测试	45%	80%	100%



用例执行情况表明，系统的第一版 bug 相当多，存在很大一部分用例无法通过的情况。经过开发人员的修改，系统的第二个版本 bug 已经少了很多。到了第三个版本，所有用例已经全部通过，代表着系统已经可以发布。

3.3 对技术方法的评价

3.3.1 Spring

➤ 简介

Spring 框架是由于软件开发的复杂性而创建的。Spring 使用的是基本的 JavaBean 来完成以前只可能由 EJB 完成的事情。然而，Spring 的用途不仅仅限于服务器端的开发。从简单性、可测试性和松耦合性角度而言，绝大部分 Java 应用都可以从 Spring 中受益。

➤ 优点

- ◆J2EE 应该更加容易使用。
- ◆面向对象的设计比任何实现技术（比如 J2EE）都重要。
- ◆面向接口编程，而不是针对类编程。Spring 将使用接口的复杂度降低到零。（面向接口编程有哪些复杂度？）
- ◆代码应该易于测试。Spring 框架会帮助你，使代码的测试更加简单。
- ◆JavaBean 提供了应用程序配置的最好方法。
- ◆在 Java 中，已检查异常（Checked exception）被过度使用。框架不应该迫使你捕获不能恢复的异常。

3.3.2 Elastic Search

➤ 简介

Elastic Search 是一个基于 Lucene 的搜索服务器。它提供了一个分布式多用户能力的全文搜索引擎，基于 RESTful web 接口。Elasticsearch 是用 Java 开发的，并作为 Apache 许可条款下的开放源码发布，是当前流行的企业级搜索引擎。设计用于云计算中，能够达到实时搜索，稳定，可靠，快速，安装使用方便。

我们建立一个网站或应用程序，并要添加搜索功能，但是想要完成搜索工作的创建是非常困难的。我们希望搜索解决方案要运行速度快，我们希望能有一个零配置和一个完全免费的搜索模式，我们希望能够简单地使用 JSON 通过 HTTP 来索引数据，我们希望我们的搜索服务器始终可用，我们希望能够从一台开始并扩展到数百台，我们要实时搜索，我们要简单的多租户，我们希望建立一个云的解决方案。因此我们利用 Elasticsearch 来解决所有这些问题及可能出现的更多其它问题。

➤ 优点

- 分布式的搜索引擎和数据分析引擎
- 全文检索，结构化检索，数据分析
- 对海量数据进行近实时的处理

3.3.3 白盒测试

➤ 简介

白盒测试又称结构测试、透明盒测试、逻辑驱动测试或基于代码的测试。白盒测试是一种测试用例设计方法，盒子指的是被测试的软件，白盒指的是盒子是可视的，你清楚盒子内部的东西以及里面是如何运作的。"白盒"法全面了解程序内部逻辑结构、对所有逻辑路径进行测试。"白盒"法是穷举路径测试，在使用这一方案时，测试者必须检查程序的内部结构，从检查程序的逻辑着手，得出测试数据。

➤ 优点

- 帮助软件测试人员增大代码的覆盖率，提高代码的质量，发现代码中隐藏的问题

➤ 缺点

- 程序运行会有很多不同的路径，不可能测试所有的运行路径
- 测试基于代码，只能测试开发人员做的对不对，而不能知道设计是否正确，可能会漏掉一些功能需求
- 系统庞大时，测试开销会非常大

3.3.4 黑盒测试

➤ 简介

黑盒测试也称功能测试，它是通过测试来检测每个功能是否都能正常使用。在测试中，把程序看作一个不能打开的黑盒子，在完全不考虑程序内部结构和内部特性的情况下，在程序接口进行测试，它只检查程序功能是否按照需求规格说明书的规定正常使用，程序是否能适当地接收输入数据而产生正确的输出信息。黑盒测试着眼于程序外部结构，不考虑内部逻辑结构，主要针对软件界面和软件功能进行测试。

➤ 优点

- 比较简单，不需要了解程序的内部的代码及实现，与软件的内部实现无关

- 从用户的角度出发，能很容易的知道用户会用到哪些功能，会遇到哪些问题
- 基于软件开发文档，所以也能知道软件实现了文档中的哪些功能
- 在做软件自动化测试时较为方便
- 缺点
 - 不可能覆盖所有的代码，覆盖率较低，大概只能达到总代码量的 30%
 - 自动化测试的复用性较低

3.4 出错原因分析

- 本项目的第一版问题较大，原因是我们的实际开发过程是前后端分开的，即先完成前端再完成后端，因此后端的工作主要是根据前端开发的结果来进行，由于前端开发人员经验不足，在开发前期没有按照《软件需求规格说明书》的要求来开发，因此虽然初版实现了大部分功能，但是仍然存在不少的问题。
- 系统初版部分功能细节并未完善，一方面的是因为在需求分析阶段，对某些需求的描述不够清楚，导致开发人员在参考《软件需求规格说明书》时也无法获取更详细的需求信息，导致细节做的不够好。
- 初版前端界面不够友好，原因是开发人员经验不足，因此页面基本都是靠框架来搭建，而基本没用 css 来美化界面，导致初版界面很难看，这个问题在需求变更访谈时被访谈对象提出，因此在最终版得到解决。
- 导致系统开发周期过长的另一个原因是几位开发人员没有协调好，因此编写代码的时候还要更改原本的数据库设计。

4. 小组成员工作总结

4.1 工作评价

➤ 薛伟（组长）

项目组组长，负责安排各项工作，整理并修改各类文档，在项目进入需求维护阶段时参与代码编写。整体工作认真负责，能在各阶段及时安排任务。同时参与到各项工作中，承担了较大的工作量，完成质量较好。

➤ 梁杰

在项目启动阶段、需求开发阶段参与各类文档的编写，在项目进入需求维护阶段后参与代码编写。工作认真负责，在项目前中期的文档编写任务中交出了质量很高的文档。项目后期的编码阶段，承担了较大部分的开发工作，并参与了系统的测试工作，在测试组长的安排下完成了最重要的测试，带动了整个项目的进程，表现十分优秀。

➤ 陈立华

在项目启动阶段、需求开发阶段参与各类文档的编写，并在需求维护阶段负责测试工作。项目前中期的文档编写质量较高，在项目后期的测试环节中拟制了优秀的测试计划，一起完成了系统测试工作。但由于测试经验较少，因此正式测试阶段出了一定差错，所幸及时补救，造成的损失不大。

➤ 梁宇坤

参与文档编写，并在需求维护阶段参与系统设计。项目前中期的文档编写质量较高，在项目后期主导系统设计，给出了优秀的系统设计方案。承担了主要的爬虫代码的编写，完成质量较好，并参与了全部的测试工作，表现尚可。

4.2 组内评分细则

小组采用完全公平的组内互评方式，由每位组员给其他 3 名成员进行打分，原则上分数在 70 ~ 95 分之间，打完分后发给组长薛伟，组长统计后去掉名字公布所有分数取平均分，即为小组成员的最终得分。

4.3 组内互评结果

姓名	评分 1	评分 2	评分 3	最终得分
薛伟	94	94	95	94
梁杰	92	91	92	92
梁宇坤	89	90	87	88
陈立华	86	86	86	86

5. 经验与教训

本项目给我们最大的教训就是开发过程一定要参考手里已经制定好的文档，特别是《软件需求规格说明书》和《系统详细设计说明书》。本次开发过程中就是由于开发人员对文档的忽视，在代码编写前期的状态完全是“想到什么写什么”，造成系统设计凌乱，导致开发周期延长。

再者，当一个项目有多个开发人员时，开发人员在开发过程中一定要做好沟通工作。本项目进入代码编写阶段时，两位开发人员尽管有定期交流，但交流频繁度还远远不够，很多时间都处于各自闭门造车的状态，很多想法没有及时告诉对方，导致实际开发出来的系统要大幅度修改数据库设计，给开发过程造成了很大困扰。

其三就是文档编写时一定要注意各方面的细节工作。我们在编写《软件需求规格说明书》和《系统详细设计说明书》时，对细节的把握不够好，尽管列出了大部分需求，但仍有部分不可少的需求没有考虑进去，已经写进去的需求也存在着部分描述不够清楚的问题，无法给开发人员准确的需求指导，导致系统的初版存在较多的问题。

当然我们的项目并非从头到尾都是教训而毫无优点。在项目从启动到结束的整个过程里，项目组的 4 名成员都参与了各种文档的编写，每个人的分工都十分明确、合理，大家的工作也都很积极。这些代码编写之外的经验是其他课程、其他专业都学不到的，它们将是我们未来参与实际开发工作的最重要的经验之一。

所幸这是课程项目，暴露出的不足都可以给我们的未来起到良性的导向作用，这些经验与教训对项目组的所有成员、尤其是对两名开发人员都是很有意义的，这能让我们在未来参与到实际的项目开发时，能够避开这些雷区，做出更好的产品。