

---

# 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 ——系统设计说明书

小组：G19

组长：薛伟

组员：梁杰、梁宇坤、陈立华

2018/07/16

---

## 目 录

2018/07/16 .....	1
1 引言 .....	3
1.1 编写目的 .....	3
1.2 背景 .....	3
1.3 定义 .....	4
1.4 参考资料 .....	4
2 总体设计 .....	5
2.1 需求规定 .....	5
2.2 运行环境 .....	5
2.3 基本设计概念及处理流程.....	6
2.5 人工处理过程 .....	7
2.6 尚未解决的问题 .....	7
3 详细设计 .....	8
3.1 登录模块设计说明 .....	8
3.2 编辑个人信息设计说明.....	10
3.3 路线推送设计说明.....	12
3.4 用户推荐模块设计说明.....	14
3.5 旅游路线详情模块设计说明.....	16
3.6 路线搜索模块设计说明.....	18
3.7 爬虫模块设计说明 .....	20
4 接口设计 .....	22
4.1 用户接口（界面设计） .....	22
5.系统数据结构设计 .....	23
6.运行设计 .....	23
6.1 运行模块组合 .....	23
6.2 运行控制 .....	23
7.系统出错处理设计 .....	24
7.1 出错信息 .....	24
7.2 补救措施 .....	24
a. 后备技术 .....	25
b. 降效技术 .....	25
c. 恢复及再启动技术.....	25
7.3 系统维护设计 .....	25

---

# 1 引言

## 1.1 编写目的

本阶段开始将进入正式编码开发阶段。本设计说明书的编写目的，是以项目的需求说明书为依据，从总体设计的角度，明确旅游信息攻略查询搜索引擎的总体架构，流程，数据结构，数据库设计，并且本说明书将为旅游信息攻略查询搜索引擎最终的实现提供重要参考。

本说明的目的在于：

- 为编码人员提供依据
- 为后续的修改和维护提供条件
- 明确模块的外部接口，内部接口，用户接口
- 项目组长按照计划说明书的要求布置和控制整个开发过程

说明书的预期读者包括：

- 项目组长
- 软件开发人员
- 软件质量分析人员
- 软件维护人员

## 1.2 背景

a) 软件系统名称

旅游信息攻略查询搜索引擎

b) 任务提出者

浙江大学软件工程项目实训课程组

c) 开发者

浙江大学软件工程项目实训 2 班 G19 四位同学

d) 用户

搜集旅游攻略的旅游者

旅游信息检索人员

---

#### e) 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:** 一般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。

**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》
- 《G19 需求规格说明书》
- 《项目实训工作描述(案例二)》
- 《CKCEST 重大工程实践分享》
- 《软件工程：实践者的研究方法》

---

## 2 总体设计

### 2.1 需求规定

#### 2.1.1 系统功能

用户界面，包括登陆窗口，注册窗口，主页，搜索，详情等页面

#### 2.1.2 系统性能

本系统的性能将由服务器端数据库，网络数据传输延时，以及并发访问该系统的用户数量决定。

#### 2.1.3 输入输出要求

客户端通过网页展现给用户一个友好的界面，用户可以通过网页界面输入向服务器提供数据的请求命令或者提交表单。经过服务器的处理后展示在用户界面。

#### 2.1.4 数据管理能力要求

安全:服务器将予以数据库最高等级的保护，防止别人从后台下载数据库，防止通过网页 SQL 注入的方式从数据库中获取信息或者破坏数据库。

性能:对于频繁访问数据库的操作，后台需要建立持久的数据库连接，以避免重复连接数据库耗费资源。

### 2.2 运行环境

本项目提供一台阿里云服务器，可以通过公网访问项目网站。

#### 2.2.1 设备

外围设备:

键盘: 能用即可

鼠标: 能用即可

显示器: 能用即可

硬盘: 40G

客户程序硬件要求:

支持 Web 浏览器的计算机或移动设备。

WEB 服务器硬件需求:

---

阿里云服务器

### 2.2.2 支持软件

服务器端：

操作系统：Linux

数据库：MySQL + ElasticSearch

web 服务器：Apache/Spring Cloud

开发工具：IntelliJ IDEA

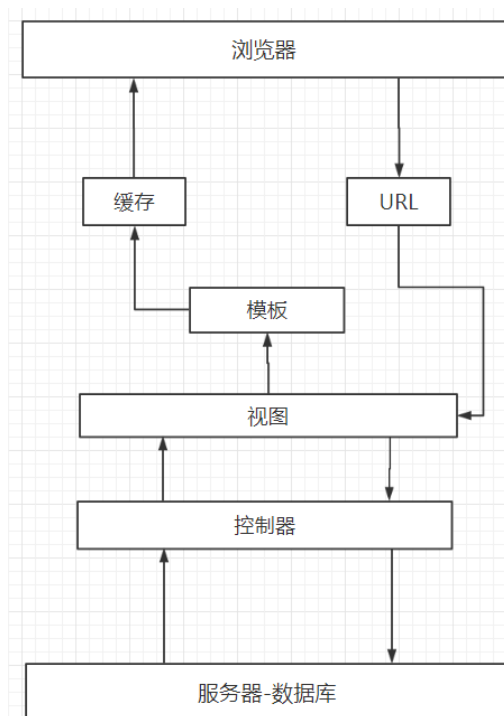
客户端：

谷歌浏览器/Safari

## 2.3 基本设计概念及处理流程

本系统分为两部分，爬虫服务和 Web 搜索服务，爬虫服务主要通过 Java 的 jsoup 来实现，配合 ElasticSearch 存储数据并建立相应的索引。

Web 搜索服务分为客户端和服务端，Java 的 Spring 框架，客户端网页使用 MVC 的开发模式，用户通过视图，以控制器为中介和服务端进行交互。客户端使用 HTML 静态页面和 JavaScript 语言，后端控制器使用 Java 访问数据库，进行数据库的读写操作。



---

## 2.5 人工处理过程

对于旅游路线某些特定信息的索引可能需要通过人工操作来修正。

对于系统的运行状态也需要人工监测和调整。

## 2.6 尚未解决的问题

关于“增量爬取”信息，可以大大地提升爬虫的工作效率，但是暂时还没有办法实现，需要进一步的学习和研究。

## 3 详细设计

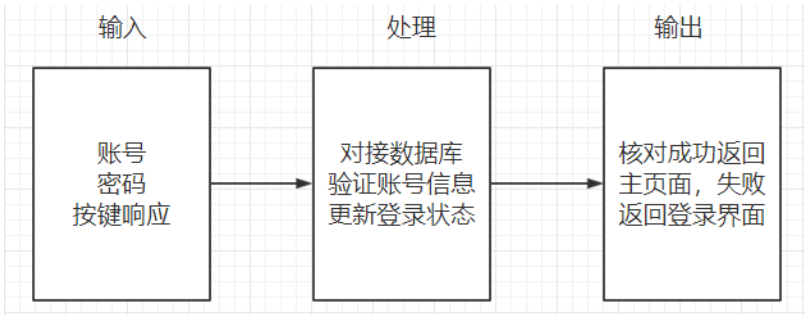
### 3.1 登录模块设计说明

#### 3.1.1 模块描述

此模块以网页表单的形式显示，用户使用客户端网页的第一个界面，在登录成功后才能使用其他的模块和功能，同时以用户的身份，作为之后个性化推荐和搜索的必要内容。模块的目的是对用户进行身份验证。

#### 3.1.2 功能

IPO 图：



#### 3.1.3 性能

用户输入账号后能进行身份验证。

#### 3.1.4 输入项

名称	标识	类型和格式	输入方式
账号	User_id	String	文本输入
密码	Password	String	文本输入
登录	Login	Button	单击按钮

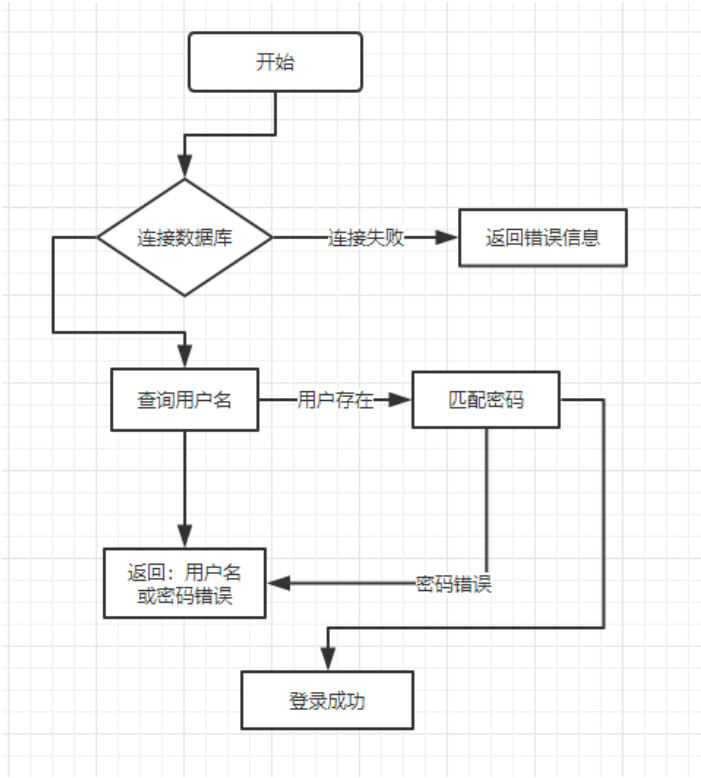
#### 3.1.5 输出项

名称	标识	类型和格式	输出方式
登录结果	Loginresult	Enum{登录成功，	脚本输出



		登录失败}	
--	--	-------	--

### 3.1.6 流程逻辑



### 3.1.7 接口

URL: POST 请求 /login

Body 包含 password 和 username

### 3.1.8 测试计划

输入数据	预期结果
输入正确的账号和密码	登录成功
输入正确的账号名称与错误的密码	登录失败，提示密码错误
输入不存在的用户名和密码	登录失败，提示用户不存在

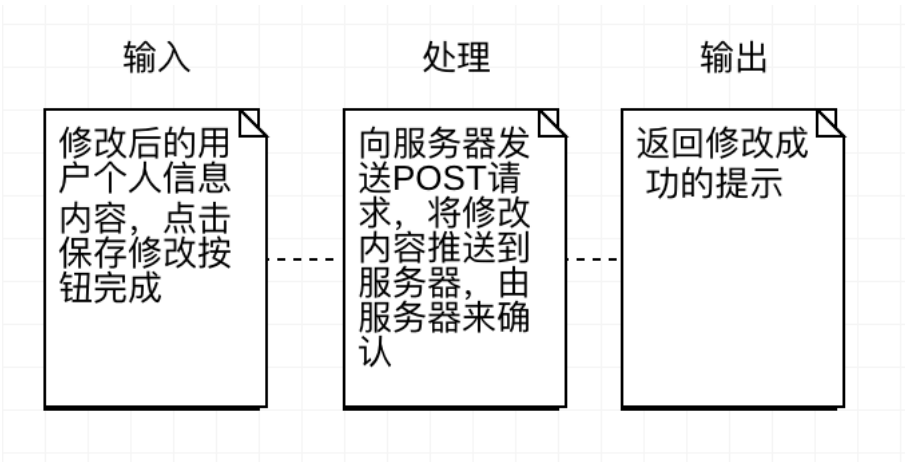
## 3.2 编辑个人信息设计说明

### 3.2.1 模块描述

个人信息显示用户的基本信息，用户可以动态编辑个人信息。包括用户名，个人邮箱，个人 ID，密码，以及密保问题。

### 3.2.2 功能

IPO 图：



### 3.2.3 性能

用户点击编辑按钮，可以进行对个人信息的修改。

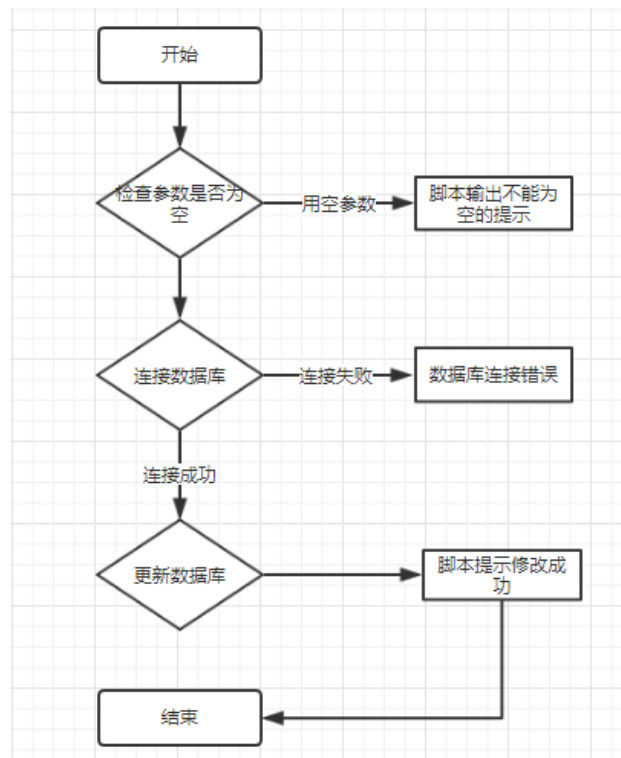
### 3.2.4 输入项

名称	标识	类型和格式	输入方式
用户名	User_name	String	文本输入
密码	Password	String	文本输入
密保问题	Question	String	文本输入
邮箱	Email	String	文本输入
电话	Phone	String	文本输入
保存修改	Save_info	Button	单击按钮

### 3.2.5 输出项

名称	标识	类型和格式	输出方式
修改结果	Modify_info_result	Enum{修改成功， 输入，不能为空， 输入不合法；	脚本输出

### 3.2.6 流程逻辑



### 3.2.7 接口

URL: POST 请求 /updatePersonalInfo

Body 包含 Username, Password, Question, Email, Phone

### 3.2.8 测试计划

输入数据	预期结果
输入完整的个人信息	修改成功
各项内容分别为空	提示输入不能为空，返回修改页面

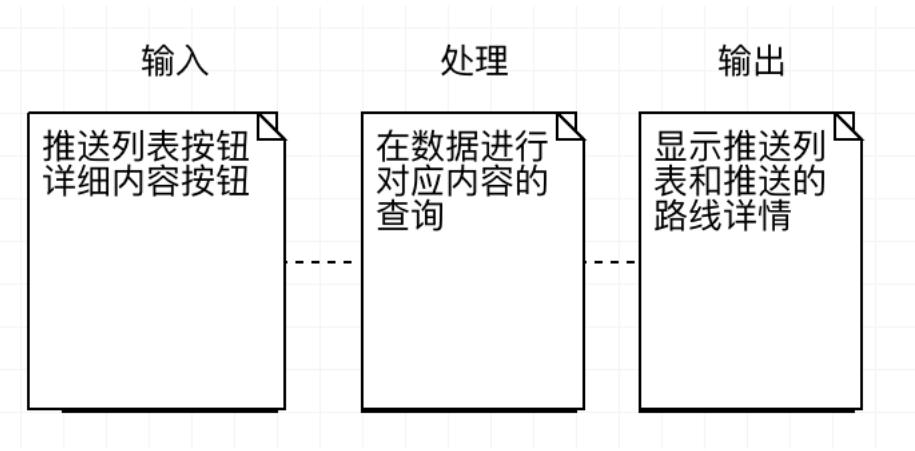
### 3.3 路线推送设计说明

#### 3.3.1 模块描述

用户在导航栏点击推送消息按钮，可以查看最近的旅游线路推送信息。在推送消息列表点击详情按钮，可以看到关于推送路线的所有信息。

#### 3.3.2 功能

IPO 图：



#### 3.3.3 性能

用户点击按钮，可以显示推送内容列表和内容信息，新的通知和已读信息区分开来。

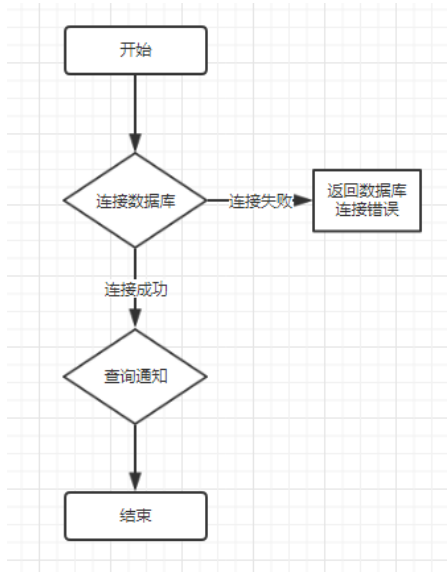
#### 3.3.4 输入项

名称	标识	类型和格式	输入方式
推送通知按钮	Note_list	Button	单击按钮
详情按钮	Note_info	Button	单击按钮

#### 3.3.5 输出项

名称	标识	类型和格式	输出方式
推送列表	List_of_all_note	{note_name,release_time}	文本框
推送详情	Note	URL	跳转到新的详情页面

### 3.3.6 流程逻辑



### 3.3.7 接口

使用 AJAX 进行 GET 请求，服务器进行 Session 身份校验  
返回 JSON 格式数据包

```
{
    'Name': 路线名字,
    'URL': 对应详情链接,
    'isRead': 是否已读
}
```

### 3.3.8 测试计划

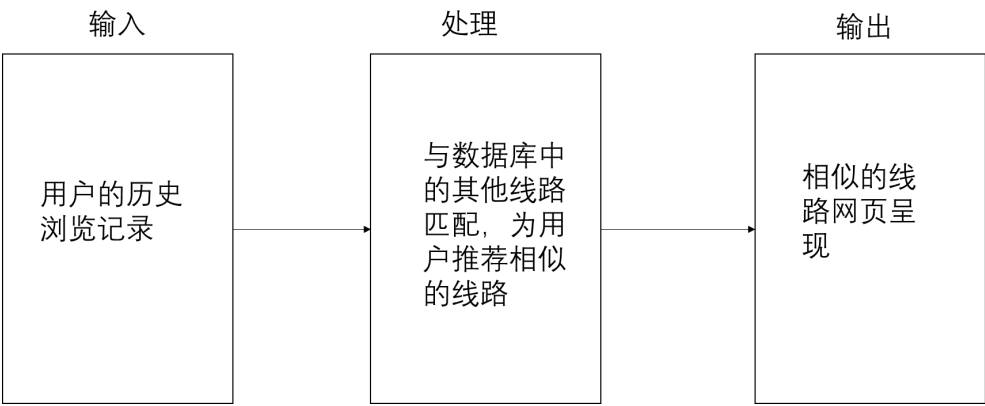
输入数据	预期结果
点击推送列表	显示推送列表
点击推送内容详情	显示推送路线详细内容，并更新推送内容状态为已读

## 3.4 用户推荐模块设计说明

### 3.4.1 模块描述

此模块以逐条呈现的网页形式，能根据用户的历史浏览记录，为用户推荐与用户喜好相关的线路。

### 3.4.2 功能



### 3.4.3 性能

根据用户的浏览记录后能进行推荐线路。

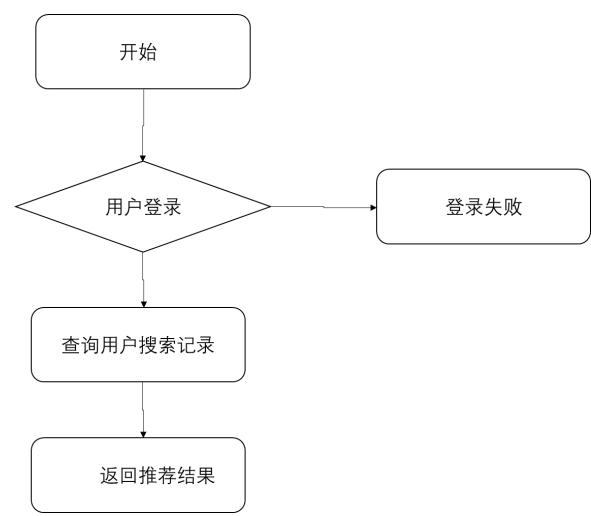
### 3.4.4 输入项

名称	标识	类型和格式	输入方式
用户的历史浏览记录	History	List	数据库记录

### 3.4.5 输出项

名称	标识	类型和格式	输出方式
推荐内容	Recommend	List	脚本输出

### 3.4.6 流程逻辑



### 3.4.7 接口

无外部接口，为服务器内部处理内容

对应函数接口为

```
function GetRecommend(username);
```

### 3.4.8 测试计划

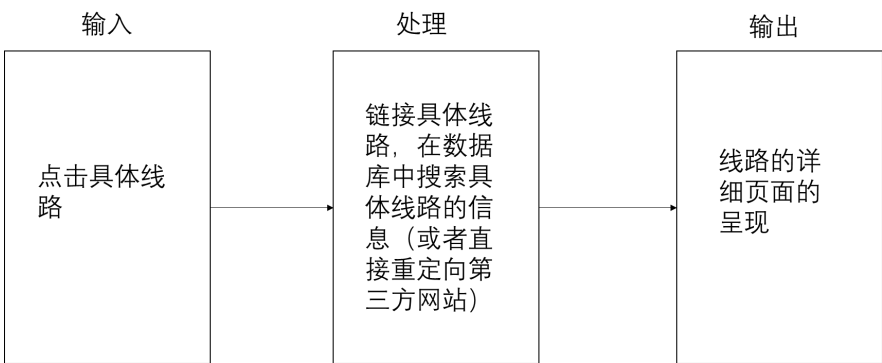
输入数据	预期结果
建立一定的搜索历史，并获得推荐内容	在后续登录时根据用户的搜索记录推荐结果

## 3.5 旅游路线详情模块设计说明

### 3.5.1 模块描述

用户在点击具体路线后，网页呈现出线路的具体信息，包括评论和推荐程度等内容。

### 3.5.2 功能



### 3.5.3 性能

用户点击具体线路后，呈现线路的详情。

### 3.5.4 输入项

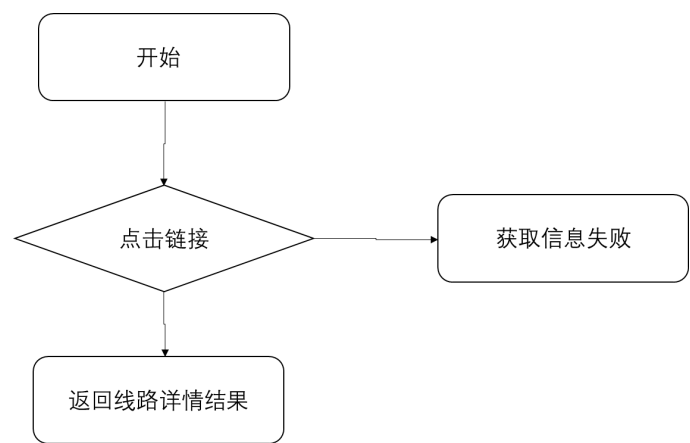
名称	标识	类型和格式	输入方式
点击相关链接	link	链接	鼠标点击

### 3.5.5 输出项

名称	标识	类型和格式	输出方式
线路详情	detail	Web 页面	脚本输出



### 3.5.6 流程逻辑



### 3.5.7 接口

URL: GET 请求 /detail?id=

Id 包含路线的 id 内容

### 3.5.8 测试计划

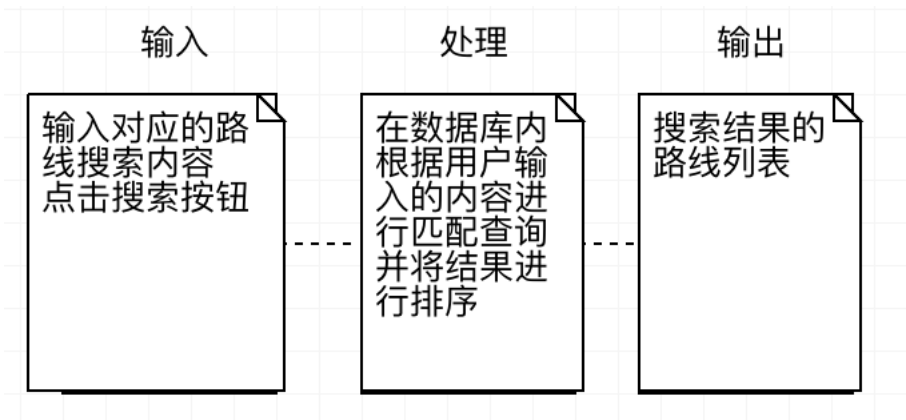
输入数据	预期结果
点击一系列的线路详情链接	返回线路详情界面

## 3.6 路线搜索模块设计说明

### 3.6.1 模块描述

用户在搜索框输入对应的内容，返回对应内容搜索到的路线结果，搜索结果按照推荐度和匹配度进行排序。

### 3.6.2 功能



### 3.6.3 性能

根据用户输入内容进行搜索，搜索时间应该小于网络的延迟时间。

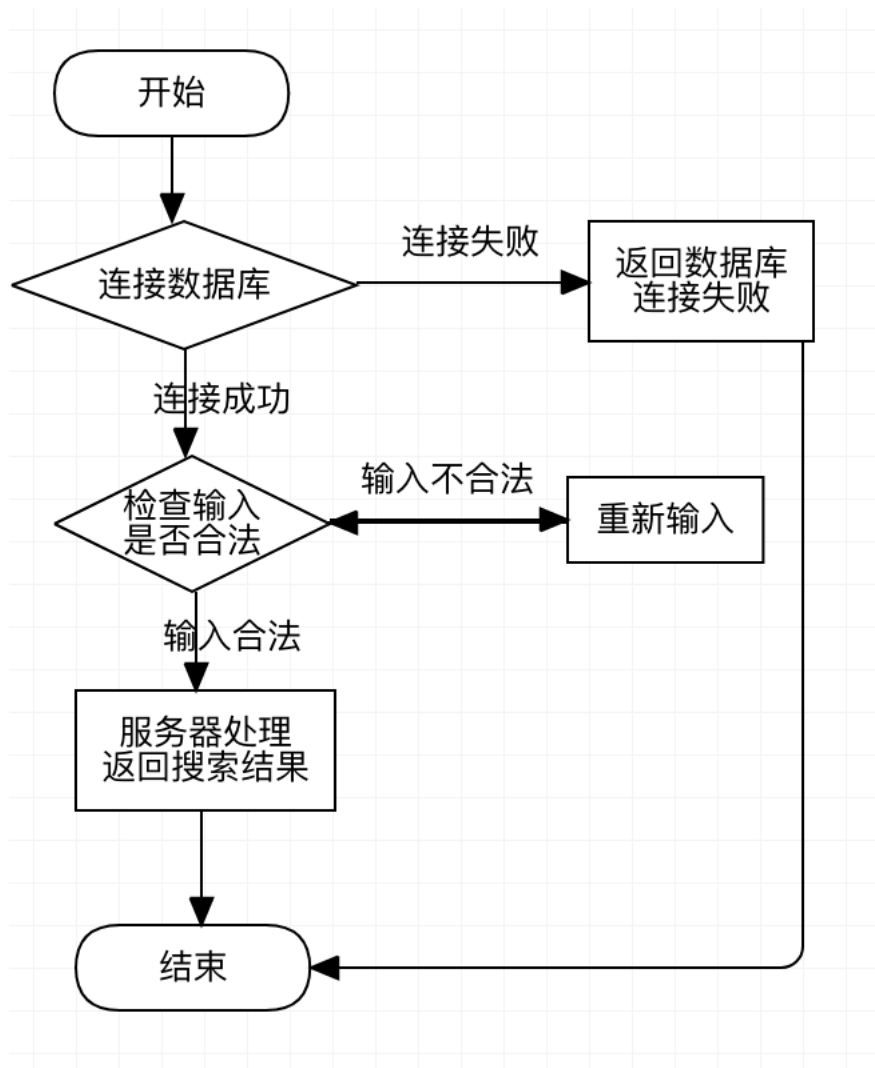
### 3.6.4 输入项

名称	标识	类型和格式	输入方式
搜索关键词	Words	String	文本框

### 3.6.5 输出项

名称	标识	类型和格式	输出方式
搜索结果列表	Result_list	列表形式	脚本输出

3.6.6 流程逻辑



3.6.7 接口

URL: POST 请求 /search

Body 包含 keywords

3.6.8 测试计划

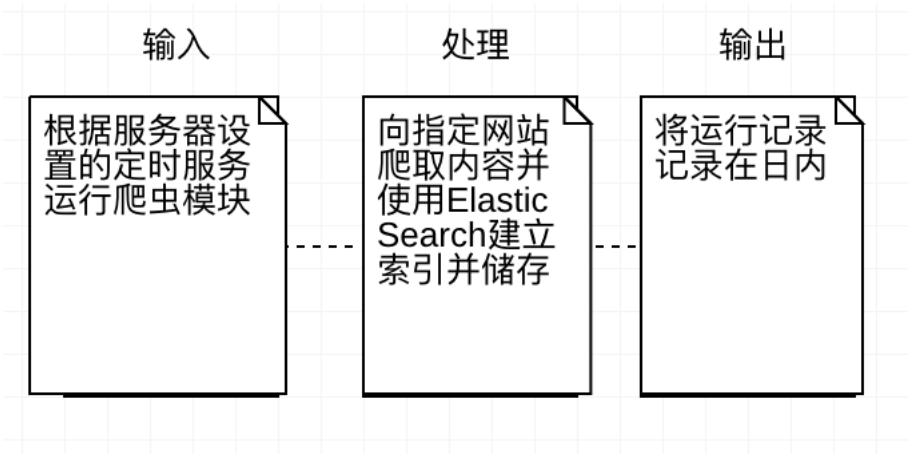
输入数据	预期结果
输入不同的关键词进行搜索	返回对应的搜索结果
输入非法关键词	提示请重新输入

## 3.7 爬虫模块设计说明

### 3.7.1 模块描述

服务器通过定时进行运行爬虫模块，来获取对应网站上的信息，使用 Elastic Search 建立索引并存储，提供给用户进行搜索。

### 3.7.2 功能



### 3.7.3 性能

爬虫运行时间需要保证，初次爬取需要时间较长，之后的每次爬取，时间应该在 10 分钟内完成。

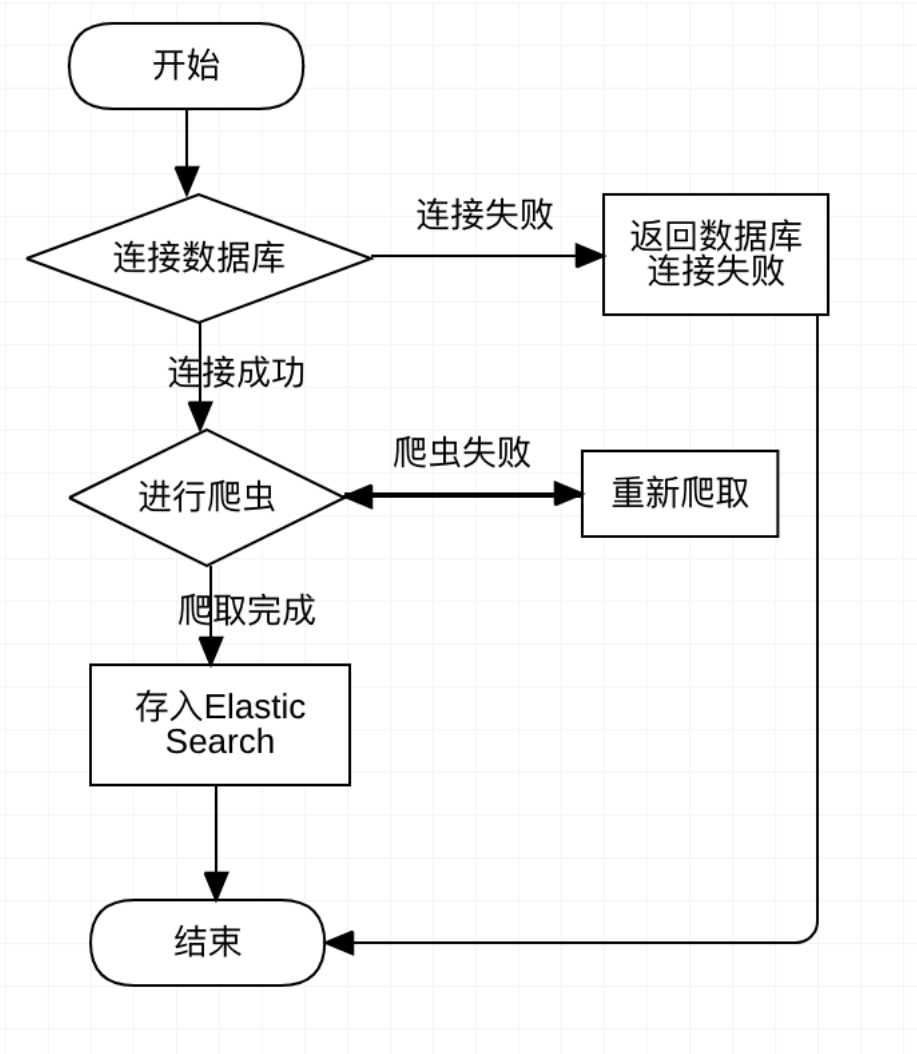
### 3.7.4 输入项

名称	标识	类型和格式	输入方式
爬虫链接	URLS	String	写入代码
爬虫时间	Timeout	Time	写入代码

### 3.7.5 输出项

名称	标识	类型和格式	输出方式
系统爬虫日志	Record	String	脚本输出
Elastic Search 索引记录	Index	记录	脚本输出

3.7.6 流程逻辑



3.7.7 接口

本模块没有外部接口

内部的调用函数是

Function UpdateInfo();

3.7.8 测试计划

输入数据	预期结果
定期检查 Elastic Search 存储内容	内容已经被更新
检查系统运行日志	爬虫按照规定时间运行

---

## 4 接口设计

### 4.1 用户接口（界面设计）

本系统使用网页开发，用户可以通过网页页面上的相关的按钮进行和系统交互，在相关的地方正确填写信息，就能使用此系统提供的各种功能。

### 3.1 外部接口

通过 Elastic Search 提供的接口来进行两个模块之间的交互。

### 3.2 内部接口

	登陆	旅游路线	系统运行状态	路线信息修改	路线推送
爬虫服务	/	共享旅游路线数据库	共享系统日志	共享旅游路线数据库	共享旅游路线数据库
Web 搜索服务	账户信息数据库	共享旅游路线数据库	共享系统日志	共享旅游路线数据库	共享旅游路线数据库



- 运行状态模块：管理员点击运行状态按钮查看运行状态；
  - 用户推送模块：用户在个人列表可以看到个人的推送内容；
  - 留言板模块：用户在菜单栏中评论列表一栏查看和发布留言；
- 用户在所有操作完成后，可点击退出该平台；

# 7.系统出错处理设计

## 7.1 出错信息

用一览表的方式说明每一种可能出错的情况出现时，系统输出信息的形式、含义级处理的方法。 由于输入信息不符合规范（如：商品数量为负数等），称之为软错误； 由于硬件方面的错误（如：网络传输超时、硬件出错等），称之为硬错误； 对于一些关键的操作（如：删除操作），应该提供提示确认机制； 对于数据、测试文档，都是要提供相应的保密措施设置；

表格 7-1：系统出错信息及解决方案

出错信息	解决方法
用户输入信息格式有误	给用户输出错误提示
Sql 语句执行错误	取消本次对于数据库的操作，并提醒用户此次操作失败，重新尝试
平台并发访问量过大	根据需求配置合适的服务器
服务器崩溃	定期对服务器进行维护
磁盘损坏	定期备份硬盘上的文件。
服务器硬盘空间不足	给用户错误提示，扩大服务器硬盘容量，清理垃圾

## 7.2 补救措施

所有的软件系统，不管采用哪种开发方式，都不可避免的存在缺陷。因而系统必须制定合理的意外预防和紧急处理方案，时刻做好准备应对错误和意外的发生。



---

### a. 后备技术

周期性地备份数据库，尤其是相对重要的文件要更加频繁地做好备份工作（比如在本系统中，学生每次作业的数据相比网站的留言，就要重要的多），以防服务器硬盘损坏或者其他原因带来的数据丢失。

### b. 降效技术

使用另一个效率稍低的系统或方法来求得所需结果的某些部分，例如一个自动系统的降效技术可以是手工操作和数据的人工记录。

通常情况下系统的管理员都不是专业的计算机人员，对计算机专业的知识了解相对较少。在面对系统瘫痪等错误时，都不知道如何去解决问题。如今，随着计算机科学技术的飞速发展，为了让系统用户能够自行解决运行过程中遇到的问题，大多采取小型服务器作为后备技术。这项技术在现有的开发系统中使用很广泛，当然备用的服务器从价格上考虑，可能不如主服务器，也可能会降低工作效率，但是从数据的完整性和安全出发，可以让维修人员有更多时间进行系统的维护

### c. 恢复及再启动技术

系统崩溃后，通过系统运行日志记录恢复数据。使软件从故障点恢复执行或使软件从头开始重新运行。

## 7.3 系统维护设计

- 硬件资源维护：定期清理服务器硬盘垃圾，可根据网站实际需求选择升级服务器性能；
- 数据库维护：定期备份数据库文件；
- 系统功能升级：根据用户实际访问平台的需求，对于系统功能进行合理的更新；