安徽财经大学字

**本科毕业设计**

|  |  |
| --- | --- |
| **题 目** | **基于Python的北上广住房数据分析与可视化** |
| **学 院** | **管理科学与工程学院** |
| **专 业** | **计算机科学与技术** |
| **班 级** | **20计科四班** |
| **学 号** | **20202568** |
| **姓 名** | **韩志鹏** |
| **指导老师** | **张晓春** |

**2021 年 5 月**

安徽财经大学管理科学与工程学院

本科生毕业论文（设计）诚信承诺书

本人承诺：

1.所呈交的毕业论文（设计）《 毕业设计过程管理系统》，是在认真学习理解《安徽财经大学学位论文作假行为处理办法》和《管理科学与工程学院本科毕业论文（设计）工作管理办法》后，保质保量独立完成的，没有弄虚作假，没有抄袭别人的内容；

2.毕业论文（设计）所使用的相关资料、数据、观点等均真实可靠，文中所有引用的他人观点、材料、数据、图表均已注释说明来源；

3.毕业论文（设计）中无抄袭、剽窃或不正当引用他人学术观点、思想和学术成果，伪造、篡改数据的情况；

4.本人已被告知并清楚：学院对毕业论文（设计）中的抄袭、剽窃、弄虚作假等违反学术规范的行为将严肃处理，并可能导致毕业论文（设计）成绩不合格，无法正常毕业、取消学士学位资格或注销并追回已发放的毕业证书、学士学位证书等严重后果；

5.若在省教育厅、学校、学院组织的毕业论文（设计）检查中，被发现有抄袭、剽窃、弄虚作假等违反学术规范的行为，本人愿意接受学院按有关规定给予的处理，并承担相应责任。

学生（签名）：

年 月 日

指导老师（签名）：

年 月 日

**中文标题**

**（宋体 二号 加粗 1.5倍行间距 段前段后0.5行 居中）**

**摘 要**

正文：小四 宋体 1.5倍行距 段落首行缩进2字符

**关键词：xx**；xx

**Title**

**（times new roman 二号 加粗 1.5倍行间距 段前段后0.5行 居中）**

**Abstract**

正文：小四 times new roman 1.5倍行距 首行缩进2字符

**Keywords： xx; xx**

# 目 录

**一级标题三号加粗 行间距1.5倍 段前段后0.5行或6磅**

**中文宋体，英文和字符 times new roman**

# 1 引言

主要写开发该系统的背景、目的和意义（比摘要要详细和全面）、软件的定义、软件的开发工具，阐述课题设计的创新点、实用价值等。

(一级标题 宋体 三号 加粗 1.5倍行间距 段前段后各0.5行 左对齐)

正文：小四 宋体 1.5倍行距 首行缩进2字符

# 2 可行性研究

概述段落

## 2.1 需求概述

北上广城市住房对中国其他城市的住房及以房地产为主导的经济起到定位和引导作用，研究北上广房价关系到中国基层人民的生活质量。

此系统通过收集北上广地区住房信息，结合用户确切的地区需求，通过数据清洗和可视化将当地住房信息具象化展现给用户。（用户需要先进行登录才能操作）。



系统流程图

粗略分析系统操作流程以及功能和特色。

(二级标题 宋体 小三号 加粗 1.5倍行间距 段前段后各0.5行 左对齐)

结合系统流图或业务流程图概述系统需求及物理组成。

## 2.2 系统可行性

### **2.2.1经济可行性**

(三级标题 宋体 四号 加粗 1.5倍行间距 段前段后各0.5行 左对齐)

**（1）工作量估算**

在软件开发阶段需要使用到的人力工作量百分比如下表2-1所示。

表2-1毕业设计过程管理系统各个开发阶段的人力百分比

（表格标题： 中文黑体 英文times new roman 五号 不加粗 表格上方 居中对齐）

|  |  |
| --- | --- |
| 任务 | 人力（%） |
| 可行性研究 | 5 |
| 需求分析 | 10 |
| 概要设计和详细设计 | 25 |
| 编码和测试 | 60 |
| 总计 | 100 |

**（2）成本估算**

在软件开发阶段需要其他一次性支出如下表2-2所示。

表2-2软件开发过程中各个开发阶段的一次性支出

|  |  |
| --- | --- |
| 项目 | 费用（元） |
| 系统前期需求研究 | 500 |
| 开发计划与测试基准研究 | 500 |
| 数据库的建立与数据字典 | 1500 |
| 检查费用和管理性费用 | 1500 |
| 培训费及软件开发人员所需的一次性支出 | 1000 |
| 总计 | 5000 |

表2-3软件开发过程中成本估算

|  |  |  |  |
| --- | --- | --- | --- |
| 项目 | 单价 | 数量 | 费用 |
| PC机 | 5000 | 2 | 10000 |
| 打印机 | 1000 | 1 | 1000 |
| 管理员工资 | 2000元/月 | 12 | 24000 |
| 不可预知费用 | \*\*\* | \*\*\* | 10000 |
| 一次性支出 | \*\*\* | \*\*\* | 5000 |
| 总计 | | 50000 | |

**（3）效益**

表2-4软件开发过程中各个开发阶段的效益

|  |  |
| --- | --- |
| 项目 | 收益（元/年） |
| 一次性收益 | 无 |
| 经常性收益（下载文献） | 80000 |
| 不可定量收益 | 无 |
| 企业定制服务 | 20000 |
| 总计 | 100000 |

**（4）收益/投资比**

一次性支出：26000元

经常性支出：24000元/年

收益：100000元/年

收益/投资比：100000\*5/(26000+24000\*5)=3.424657

**（5）货币的时间价值**

五年预计收益：100000\*5-（26000+24000\*5）=354000（元）

住房数据分析系统的前期软件开发阶段，需要耗费巨大的人力，投资的成本和未来获得的效益会随着软件每年的货币时间价值有所改变。住房数据分析系统五年收益为354000元。实际收益需要通过货币时间价值来进行核算。用利率的形式表示货币的时间价值。假设年利率为i，如果现在存入P元，则n年后可以得到的钱数为：

H:\temp\ksohtml14132\wps1.jpg (1)

这也就是P元钱在n年后的价值。反之，如果n年后能收入F元钱，那么这些钱的现在价值是：

H:\temp\ksohtml14132\wps2.jpg (2)

假定年利率为12%，利用上面计算货币现在价值的公式可以算出系统5年预计收益的现在价值，如下表2-5所示。

表2-5 将来收入折算成现在值

|  |  |  |
| --- | --- | --- |
| 将来值（元） | （1+i)^n | 现在值（元） |
| 354000 | 1.7623 | 200873 |

**（6）投资回收期估算**

住房数据分析系统基础建设成本约为26000元，第一年内收入为100000元，软件投资的回收期约为

 （3)

因此软件的投资回收期约为0.26年，本系统开发成本较低，维修成本适中，软件投资回报期较短可以较快获得利润，用户群体丰富且需求量大，经济利益客观值得投资。

### **2.2.2技术可行性**

爬取技术网络爬虫（Web Crawler）是基于 http 协议中的请求与响应机制，按照一定的规则自动提取 Web 网页的应用程序或者脚本，可以完整爬取整个网页的源代码。开发者通过相关 设备发送请求并传输 url 到目标服务器，目标服务器通过相关处理和优化分析后返回访问者所需的信息，访问者得到相关 HTML 信息后可利用 Beautiful Soup、正则表达式、Xpath 等技术获取想要的网页固定信息。

数据清洗和可视化技术 Python 作为一种新兴语言为开发者们提供了两种常用的数据处理方式，即 Pandas 和 Matplotlib。Pandas 除了可以提取和保存数据以外，还支持不同数据结构间进行相互转换。Matplotlib 可以将数据以图表形式呈现出来，更为直观地展示给用户，进 而将数据的特征显示出来。在爬取数据前，第一步选择进行 数据采集的网站，第二步对选择的目标网站结构进行分析，进 而找到合适的请求方式解析请求返回的响应，同时将爬取的数据保存到本地，从而为进行数据分析做好下一步准备工作。

### **2.2.3 操作可行性**

操作可行性是对开发系统在一个特定的工作环境中能否运行或运行的好坏程度的衡量。该系统采用前端web页面显示可视化结果，界面友好，操作简单，用户无需掌握相关技术，即可进行相关操作，故该系统具有操作可行性。

## 2.3 项目进度计划

示例：

表2-5软件开发过程中各个开发阶段的一次性支出

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **项目阶段** | **时间** | **工作内容** | **成果** | **负责人** | **审核人** |
| 需求调研 | 2022.10.1-2022.10.15 | 对项目需求进行详细调研 | 系统需求规格说明书 | 韩志鹏 | 韩志鹏 |
| 系统设计 | 2022.10.16-2022.10.31 | 在需求调研的基础上对系统构架、安全体系、功能等进行系统设计 | 系统设计说明书 | 韩志鹏 | 韩志鹏 |
| 系统开发 | 2022.11.1-2022.11.15 | 进行各个子系统迭代开发，完成单元测试 | 不同迭代版本的可运行系统 | 韩志鹏 | 韩志鹏 |
| 系统集成 | 2022.11.16-2022.11.30 | 系统集成和对各模块集成测试 | 测试报告，形成可完整运行的系统 | 韩志鹏 | 韩志鹏 |
| 系统初验 | 2022.12.1-2022.12.15 | 项目初验 | 初验报告 | 韩志鹏 | 韩志鹏 |
| 系统试运行 | 2022.12.16-2022.12.31 | 1. 系统试运行 2. 系统持续优化 | 升级版本的可运行系统并安装部署到用户本地 | 韩志鹏 | 韩志鹏 |
| 用户培训 | 2023.1.1-2023.1.15 | 对各级用户进行培训 | 使用报告 | 韩志鹏 | 韩志鹏 |
| 项目终验 | 2023.16-2013.2.1 | 项目终验 | 验收报告 | 韩志鹏 | 韩志鹏 |

# 3 需求分析

概述段落

## 3.1 系统综合需求

### **3.1.1功能需求概述**

分析并完成用户指定地区(北上广)住房数据的可视化

### **3.1.2 系统性能需求概述**

用户能够登录网站并得到分析结果

### **3.1.3 其他需求概述**

无

## 3.2 数据流图分析

### **3.2.1 顶层数据流图**



系统的源点和终点都是用户，用户通过登录进入住房数据分析系统，系统处理用户指令后将分析结果返回给用户

### **3.2.2 数据流图细化与分解**

#### （1）0层数据流图

管理员具有以下功能：

1. 用户信息管理功能：对用户基本信息进行检索、增删改和权限修改；
2. 电影数据系统中所有信息进行查询和增删改管理；
3. 快捷操作和帮助模块：一些系统的辅助操作，便于用户使用系统；

普通用户则可以通过登录该系统，实现电影信息的查询。VIP用户在普通用户的基础上增加住房推荐功能。



#### （2）1层数据流图







#### (3)2层数据流图





## 3.3 数据字典

（由于数据字典定义内容非常多，可针对每一小类选择代表性的示例展示）

### **3.3.1数据流条目定义**

实例：

表3-X 验证用户信息数据字典定义

根据3.2节数据流图分析情况可知，系统主要涉及的数据项包括：xx, xx, xx, xx和xx分别定义如图3-X~3-Y所示。

名称：验证用户信息

别名：无

描述：用户登录信息的输入

数据流组成：验证用户信息=用户名+密码+用户权限

数据流来源：用户

数据流去向：用户登录

（每个类型的数据字典定义前应有一段总结性的开场白。）

### **3.3.2数据存储条目定义**

实例：

表3-X 用户信息数据字典定义

名字：用户信息表

别名：用户信息

描述：唯一标识用户

定义：用户信息表=用户账号+用户密码+用户权限

位置：输入到电脑

### **3.3.3数据项条目定义**

实例：

表3-X 用户账号数据字典定义

名称：用户账号

别名：用户编号

含义：唯一确定用户身份编号

类型：数值

长度：11

取值范围：10000000000~19999999999

## 3.4实体联系分析

### **3.4.1 实体提取及实体图**

表3-18 实体提取结果

|  |  |  |
| --- | --- | --- |
| 实体 | 主码 | 其他属性 |
| 用户 | 用户编号 | 用户编号，用户密码，用户权限，用户类型 |
| 房屋产权人 | 姓名 | 姓名，联系方式 |
| 住房 | 地址 | 地址，价格，户型，面积 |

图 3-2 住房数据分析系统-用户属性图



图 3-3 住房数据分析系统-房屋产权人属性图



图 3-4 住房数据分析系统-住房属性图



### **3.4.2 实体联系图**



# 4软件设计

## 4.1 功能简述

根据前期需求分析可以得到数据分析系统主要有3大功能模块，用户登录管理功能模块、住房数据分析功能模块、使用帮助功能模块，每个子模块再调用更下层的模块从而完成程序的子功能，最下层的模块完成最基本的功能。



图4-1 住房分析系统的一级功能结构图

用户登录管理功能模块包括用户登录、用户注册、找回密码、用户注销四个部分。

1. 管理员可以对用户的基本信息进行检索、增删改和权限修改。
2. 用户则可以通过登录该系统，实现查询电影信息的服务。

（3） 用户注册时，系统将引导用户进行注册，不能使用已存在的用户名，注册成功后返回用户登录界面进行登录。当用户输入账号密码后，系统会与用户信息表中的信息进行比对，比对成功则登录成功。

（4） 登陆成功后，用户可以进行注销以及修改密码等操作，对比失败可以通过找回密码功能再次登录。



图4-2用户登录管理功能模块结构图

电影数据分析功能模块共分为电影信息爬取、电影信息存储、电影数据分析及数据可视化四个部分。

（1）电影信息爬取通过使用Python爬虫工具在各网站爬取大量相关数据，筛选出有效的数据，并进行信息存储。

（2）电影信息存储通过构建数据库来实现信息的安全存储，管理员可以对其进行管理。

（3）电影数据分析将存储在数据库内的数据进行分类整理和分析，将数据划分为许多个以不同主题小的数据集，为可视化提供数据来源。

（4）数据可视化将数据以各种形象的方式显示在用户面前，便于用户了解该电影的相关内容。



图4-3 电影数据分析功能模块结构图

电影数据分析可以分成电影信息分类、影评关键词提取、分析数据存储及同类电影特点分析四个部分。将整理好的电影信息进行不同主题的分类，并提取影评中的关键词，将这些信息存储整理并分析还可以对同类电影的特点进行总结分析，并将分析好的数据提供给数据可视化模块。



图4-4 电影数据分析结构图

电影信息可视化可以分成原始结果展示、图表展示及词云展示。将数据分析后的电影信息通过原始分析数据进行展示，原始分析数据使用更加形象的图表进行展示，也可以使用词云展示可视化的信息。



图4-5 数据可视化结构图

综上所述，将基于Python爬虫的电影数据分析系统进行模块化结构表示，并对每一个子模块在进行模块表示，构建出整体系统结构图，并实现完整的子功能，用户使用电影数据分析系统时结构更加清晰明了，并使用可视化进行图表词云等数据表示，更加方便用户使用该电影数据分析系统。

图4-6 基于Python爬虫的电影数据分析系统整体系统结构图

## 4.2 基于数据流分析构建软件结构

### **4.2.1面向数据流分析**

面向数据流的设计方法把信息流映射成软件结构，信息流的类型决定了映射的方法。

（1）参看图3-2，信息沿输入通路进入系统，同时由外部形式变换为内部形式，进入系统的信息通过变换中心，经加工处理以后再沿输出通路变换成外部形式离开软件系统，可见这些信息流具有交换流的总特征。

（2）确定输入流和输出流的边界，从而孤立出变换中心。

（3）完成各级分解，得到未经精化的软件结构图。



图4-7未经精化的输入结构



图4-8未经精化的变换结构



图4-9未经精化的输出结构

图4-10 基于Python爬虫的电影数据分析系统未经精化的软件结构图

### **4.2.2结构调整与优化**

由图3-6可知，电影信息可视化有明显的事务中心，可以将原本的输出结构调整为事务流。

由事务流映射成的软件结构包括一个接收分支和一个发送分支，映射出接收分支结构的方法和变换分析映射出输入结构的方法很相像，即从事务中心的边界开始，把沿着接收流通路的处理映射成模块。发送分支的结构包含一个调度模块，它控制下层的所有活动模块，然后把数据流图中的每个活动流通路映射成与他的流特征相对应的结构。



图4-11精化后的输出结构

### **4.2.3软件结构图**

图4-12 基于Python爬虫的电影数据分析系统精化后的软件结构图

## 4.3 模块接口、算法设计

### **4.3.1 人机界面**

人机交互过程中人机交互界面处于视感知层面的主要操作区域，是用户与系统直接接触的部分，交互界面的好坏决定着用户体验的直接感受。在人机界面设计的过程中，我们主要考虑以下四个方面的问题：

（1）系统响应时间

系统响应时间是交互式系统中用户抱怨最多的问题。如果系统响应时间过长，用户就会感到紧张和沮丧；但是，当用户工作速度是由人机界面决定的时候，系统响应时间过短也不好，这会迫使用户加快操作节奏，从而可能会犯错误。除了响应时间的绝对长短外，用户对不同命令在响应时间上的差别亦很在意，若过于悬殊用户将难以建立起稳定的工作节奏，在敏感条件下将会难以接受。保证响应时间的稳定，可以大大提高系统的属性。

（2）用户帮助设施

人机交互界面的使用使得每个用户都需要帮助，当遇到复杂问题时甚至需要查看用户手册以寻找答案，我们在设计数据分析系统时也要考虑用户遇到问题时能否很好的解决，提供用户帮助设施，使得用户无需离开用户界面就能解决自己的问题。而在具体设计帮助设施时，我们必须考虑用户在使用帮助设施时出现的问题。用户与系统交互期间，我们可以通过提供部分功能的帮助信息和提供全部功能的帮助信息，以保证用户在任何时候都能获得关于系统任何功能的帮助信息。除此以外，我们也可以通过帮助菜单，特殊功能键或者是HELP命令，帮助用户快速找到帮助入口。显示帮助信息时，可以选择在屏幕固定位置显示简短提示。在用户返回到正常的交互方式方面，可以提供屏幕上的返回按钮和功能键，让用户了解帮助后可以快速回到原界面。用户求助机制宜采用集成式，也避免叠加式系统导致用户求助某项指南而不得不浏览大量无关信息。

（3）出错信息处理

一般来说，由于用户的不熟悉或是误触等一系列操作，使得交互界面发出出错信息和警告信息几乎是必然的。对于错误和警告信息的选择，必须选用用户明了、含义准确的术语描述，同时还应尽可能提供一些有关错误恢复的建议，还有指出错误可能导致哪些负面后果(例如，破坏数据文件)，以便用户检查是否出现了这些问题，并在确实出现问题时方便解决。伴随着听觉上或视觉上的提示，也可以给用户更为直观的感受。信息更不能带有指责色彩，给用户被责怪的体验。

（4）命令交互

作为用户和系统软件交互最常用的方式，命令行依旧是许多高级用户的偏爱。数据分析系统选择菜单与键盘命令并存，在交互界面上提供用户更多的选择，提高系统的实用性。选择“命令宏机制”可以用自己定义的名字代表一个常用的命令序列，保证用户使用命令序列时无须依次输入每个命令，只需输入命令宏的名字就可以顺序执行它所代表的全部命令。除此以外，还要考虑大多数应用软件命令的使用方法，将命令普遍化，保证在软件切换过程中达成一致。

### **4.3.2 模块参数**

由图4-6可知，数据分析系统主要有3大功能模块：用户登录管理功能模块、电影数据分析功能模块、使用帮助功能模块，这三大模块之间还有很多小模块，每个小模块之间相互联系，相互耦合，就能实现大模块的功能，三大主要模块相互联系最终就能可以完成对数据分析系统的实现。

利用爬虫程序来抓取网页这个过程其实跟平时利用各种浏览器看网页是一样的。在浏览网页时，首先要做的就是在浏览器的地址栏输入网页的地址，比如https://movie.douban.com/。这时浏览器作为一个浏览的工具，向服务器发送请求，在把服务器中的内容抓取到本地。并且进行相应的解释和展示。这就是我们通过浏览器打开一个以页面的过程。

html作为一种标记语言，通过标签来标记内容，并且进行区分和解释。而浏览器将获取到的html代码解析转变成我们看到的各种页面。

**（1）用户登录管理功能模块**

1. 用户注册

* URL：<http://127.0.0.1:8000/register>
* 请求方法：POST
* 请求参数：JSON

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 含义 | 类型 | 备注 |
| uname | 用户名 | char | 必填 |
| password | 密码 | char | 必填 |
| phone | 手机号 | char | 必填 |
| captcha | 验证码 | char | 必填 |

* 响应数据说明：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 含义 | 类型 | 备注 |
| code | 状态码 | int | 必填 |
| uname | 用户名 | char | 与error二选一 |
| data | 返回数据[token] | dict | 与error二选一 |
| error | 错误原因 | char | 错误时填写 |

1. 用户登录

* URL：<http://127.0.0.1:8000/load>
* 请求方法：POST
* 请求参数：JSON

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 含义 | 类型 | 备注 |
| uname | 用户名 | char | 必填 |
| password | 密码 | char | 必填 |

* 响应数据说明：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 含义 | 类型 | 备注 |
| code | 状态码 | int | 必填 |
| uname | 用户名 | char | 与error二选一 |
| data | 返回数据[token] | dict | 与error二选一 |
| error | 错误原因 | char | 错误时填写 |

* 1. 找回密码

1. 发送手机号码验证请求

* 页面名：findPass.html
* URL：http://127.0.0.1:8000/password
* 请求方法：POST
* 请求参数：JSON

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 含义 | 类型 | 备注 |
| phone | 手机号 | char | 通过手机号码找回密码 |

* 响应数据说明：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 含义 | 类型 | 备注 |
| code | 状态码 | int | 必填 |
| data | 返回数据[token] | dict | 与error二选一 |
| error | 返回错误原因 | char | 与data二选一 |

* 返回值：JSON

1. 发送手机号码验证码

* 页面名：findPass.html
* URL：<http://127.0.01:8000/v1/users/password/verification/>
* 请求方法：POST
* 请求参数：JSON

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 含义 | 类型 | 备注 |
| phone | 手机号 | char | 必填 |
| captcha | 验证码 | char | 必填 |

* 响应数据：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 含义 | 类型 | 备注 |
| code | 响应状态码 | char | 必填 |
| data | 返回正常数据 | dict | 与error二选一 |
| error | 返回错误原因 | char | 与data二选一 |

* 返回值：JSON

1. 修改密码

* 页面名： findPass.html
* URL：<http://127.0.0.1:8000/v1/users/password/new>
* 请求方法：POST
* 请求参数：JSON

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 含义 | 类型 | 备注 |
| phone | 手机号 | char | 必填 |
| password1 | 新密码 | char | 必填 |
| assword2 | 确认密码 | char | 必填 |

* 响应数据：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 含义 | 类型 | 备注 |
| code | 响应状态码 | char | 必填 |
| data | 返回正常数据 | dict | 与error二选一 |
| error | 返回错误原因 | dict | 与data二选一 |
| message | 具体响应信息 | char | 正确或错误原因 |

* 返回值：JSON

1. 用户注销

* URL：<http://127.0.0.1:8000/v1/users/delete>
* 请求方法：POST
* 请求参数：JSON

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 含义 | 类型 | 备注 |
| uname | 用户名 | char | 必填 |
| password | 密码 | char | 必填 |
| phone | 手机号 | char | 必填 |

* 返回值：JSON

**（2）电影数据分析模块**

1. 电影相关信息爬取

* URL: http://127.0.0.1:8000/subject/movieId
* 请求方式：POST
* 请求参数：JSON

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 含义 | 类型 | 备注 |
| MovieId | 电影编号 | string | 必填 |

* 响应数据：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 含义 | 类型 | 备注 |
| MovieId | 电影编号 | string | 必填 |
| Movie\_Information | 电影信息 | json对象 | 必填 |

Movie\_Information对应的是电影相关信息，是一个json数组，里面存的具体的爬取的电影具体相关信息的格式为：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 含义 | 类型 | 备注 |
| Name | 电影名称 | string | 必填 |
| Id | 电影编号 | string | 必填 |
| ChineseName | 中文名称 | string | 必填 |
| Images | 图片信息 | json对象 | 存放各种大小的电影图 |
| Release | 上映方式 | string | 必填 |
| DatePublished | 上映时间 | int | 必填 |
| Genre | 电影类型 | json数组 | 必填 |
| Language | 语言 | string | 必填 |
| Region | 上映国家 | string | 必填 |
| Rating | 评分信息 | json对象 |  |
| Collect\_count | 观看人数 | int |  |
| Description | 电影简介 | string | 必填 |
| Directors | 导演列表 | json数组 | 必填 |
| All\_actors | 主演列表 | json数组 | 必填 |
| Reviews | 影评 | json数组 |  |

Rating是评分信息，是一个json数组，里面存的是具体的评分信息评分信息的格式为：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 含义 | 类型 | 备注 |
| Max | 最高分 | int | 必填 |
| Average | 平均评分 | double | 必填 |
| Min | 最低分 | int | 必填 |

Directors 是导演列表，是一个 json 数组，里面存的是导演的信息，导演信息的格式为：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 含义 | 类型 | 备注 |
| Name | 名字 | string | 必填 |
| Id | 导演编号 | string | 必填 |
| ChineseName | 中文名 | string | 必填 |
| Images | 图片信息 | json对象 | 存放各种大小的电影图 |
| Sex | 性别 | int | 必填 |
| Birthday | 出生日期 | int | 必填 |
| Birthplace | 出生地点 | int |  |
| OtherName | 曾用名 | string |  |
| FamilyMember | 家庭成员 | string | 必填 |
| Introduction | 人物简介 | string | 必填 |
| Award | 所获奖项 | json对象 | 曾获得的各种奖项 |

All\_actors是主演列表，是一个json数组，里面存的是具体的主演信息，主演信息的格式为：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 含义 | 类型 | 备注 |
| Name | 名字 | string | 必填 |
| Id | 演员编号 | string | 必填 |
| ChineseName | 中文名 | string | 必填 |
| Images | 图片信息 | json对象 | 存放各种大小的电影图 |
| Sex | 性别 | int | 必填 |
| Birthday | 出生日期 | int | 必填 |
| Birthplace | 出生地点 | int |  |
| OtherName | 曾用名 | string |  |
| FamilyMember | 家庭成员 | string | 必填 |
| Introduction | 人物简介 | string | 必填 |
| Award | 所获奖项 | json对象 | 曾获得的各种奖项 |

Reviews是影评列表，是一个json数组，里面存的是具体的影评信息，格式为：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 含义 | 类型 | 备注 |
| Id | 用户编号 | string | 必填 |
| Usename | 用户名 | string | 必填 |
| Place | 所处地点 | string | 必填 |
| Issued\_time | 发布时间 | int | 必填 |
| Detail | 影评详情 | string | 必填 |

1. 电影信息存储
   1. 连接SQL数据库： 前面数据爬取完成后，我们连接My\_SQL，创建数据库和表，发送请求，获取响应。

* 请求方法：POST
* 请求参数：JSON

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 含义 | 类型 | 备注 |
| Number | 编号 | string | 必填 |

* 响应数据：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 含义 | 类型 | 备注 |
| code | 状态码 | int | 必填 |
| data | 返回数据[token] | dict | 与error二选一 |
| error | 返回错误原因 | char | 与data二选一 |

1. 数据信息存储

响应获取成功后，将响应进行解析，获取响应内的数据，并将数据存储到My\_SQL数据库里。

* 请求方法：POST
* 请求参数：JSON

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 含义 | 类型 | 备注 |
| Number | 编号 | string | 必填 |

* 响应数据：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 含义 | 类型 | 备注 |
| Number | 编号 | string | 必填 |
| Message | 编号信息 | json对象 | 编号对应的详细信息 |

1. 电影信息分类

引入数据库，将库里的数据导出，对电影数据信息进行分类处理，将部分属性相同的收集到一起，为后面的电影分析做准备。

* 请求方法：POST
* 请求参数：JSON

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 含义 | 类型 | 备注 |
| Id | 电影编号 | string | 必填 |
| Name | 电影名称 | string | 必填 |
| Genre | 电影类型 | json数组 | 必填 |
| Directors | 导演列表 | json数组 | 必填 |
| All\_actors | 主演列表 | json数组 | 必填 |
| Description | 电影简介 | string | 必填 |
| Collect\_count | 票房 | int |  |
| Review | 影评 | json数组 |  |
| DatePublished | 上映时间 | int | 必填 |
| Release | 上映方式 | string | 必填 |
| Region | 国家 | string | 必填 |

* 响应数据：JSON

按电影类型分类：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 含义 | 类型 | 备注 |
| Genre | 电影类型 | string | 电影类型相同时输出 |
| Id | 电影编号 | string | 必填 |
| Movie\_information | 电影信息 | json对象 | 必填 |
| Number | 数量 | int | 求和 |

按电影上映国家分类：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 含义 | 类型 | 备注 |
| Region | 国家 | string | 上映国家相同时输出 |
| Id | 电影编号 | string | 必填 |
| Movie\_information | 电影信息 | json对象 | 必填 |
| Number | 数量 | int | 求和 |

按电影上映时间分类：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 含义 | 类型 | 备注 |
| DatePublished | 上映时间 | int | 上映年份相同时输出 |
| Id | 电影编号 | string | 必填 |
| Movie\_information | 电影信息 | json对象 |  |
| Number | 数量 | int | 求和 |

后续的分类方法（电影上映方式、导演、演员等）与上面类似。

1. 影评关键词提取

影评关键词的提取我们分为两部分来处理：评论分词模块和关键词相关性分析模块。首先通过分词模块的处理，将目标电影的影评分析为一个个词语，并将词语计数，从而形成一个影评信息表。关键词相关性分析主要通过建立电影关键词库，对影评信息表中的评论进行分析，包括分词数量的统计，关键词的情绪分析结果，以及对演员的评价分析等，为后面的可视化分析提供数据基础。

1. 评论分词模块

为了确保关键词选择的正确，我们选择了基于字符串匹配的分词分析仪IK Analyzer。通过IK的智能切分方法，将电影评论里的关键词正确分开。首先，我们需要先导入数据库内评论表的评论数据，放到一个ArrayList中，ArrayList的每一个元素都是一篇影评。对于这个链表中的每一个分出的词都加以计数并且排序。并在数据库中建立一个影评信息表，将得到的数据导入，以方便后续的使用。

* 请求方法：POST
* 请求参数：JSON

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 含义 | 类型 | 备注 |
| Id | 电影编号 | string | 必填 |
| Name | 电影名称 | string | 必填 |
| Review | 影评 | json数组 |  |

* 响应数据：JSON

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 含义 | 类型 | 备注 |
| Word | 分词 | string | 必填 |
| Number | 数量 | int | 每当这个分词再次出现，数目加一 |

1. 关键词电影相关性处理

数量筛选是对词语提出的要求之一，关键词的筛选还有另外一个要求就是必须是与电影元素相关的词。通过对电影的了解，我们对多部各类型的影评分词进行了统计，得出了一个具体的关键词类型分组，并将其录入数据库。再将影评信息表中跟电影相关的名词并且出现次数多的分词进行分类，然后存入电影要素关键词库，分词中出现较多的形容词进入电影消费者的情绪词库中进一步使用和统计。目标电影影评中的分词与关键词库和形容词库进行比对，形成最后与电影具有相关性的关键词统计数据。

* 请求方法：POST
* 请求参数：JSON

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 含义 | 类型 | 备注 |
| Id | 电影编号 | string | 必填 |
| Name | 电影名称 | string | 必填 |
| Word | 分词 | string | 必填 |
| Number | 数量 | int | 必填 |

* 响应数据：JSON

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 含义 | 类型 | 备注 |
| Keywords | 关键词 | string | 必填 |
| Number | 数量 | int | 必填 |

1. 分析数据存储
   1. 连接SQL数据库

数据分析完成后，我们连接My\_SQL，创建关键词数据库和表，发送请求，获取响应。

* 请求方法：POST
* 请求参数：JSON

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 含义 | 类型 | 备注 |
| Id | 电影编号 | string | 必填 |
| Name | 电影名称 | string | 必填 |
| Review | 影评 | json数组 |  |

* 响应数据：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 含义 | 类型 | 备注 |
| code | 状态码 | int | 必填 |
| data | 返回数据[token] | dict | 与error二选一 |
| error | 返回错误原因 | char | 与data二选一 |

1. 数据信息存储

响应获取成功后，将响应进行解析，获取响应内的数据，并将数据存储到My\_SQL数据库里。

* 请求方法：POST
* 请求参数：JSON

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 含义 | 类型 | 备注 |
| Number | 编号 | string | 必填 |
| Keywords | 关键词 | string | 必填 |
| Number | 数量 | int | 必填 |

* 响应数据：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 含义 | 类型 | 备注 |
| code | 状态码 | int | 必填 |
| data | 返回数据[token] | dict | 与error二选一 |
| error | 返回错误原因 | char | 与data二选一 |

1. 数据可视化

通过前面对爬取数据的分析和整理，我们可以将数据进行可视化分析，使数据以一种更为直观的图表形式展现给用户，方便加深对电影信息概况的了解。

a) 基于各类型电影历年总票房、平均票房可视化

这里我们从电影类型视角出发，展示不同年份各类型电影的总票房、平均票房变化。

* 请求方法：POST
* 请求参数：JSON

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 含义 | 类型 | 备注 |
| DatePublished | 上映时间 | int | 必填 |
| Genre | 电影类型 | string | 必填 |
| Number | 数量 | int | 必填 |

b)基于影评中电影元素的可视化

这里我们选择从电影影评中观察出用户对于电影类型选择上的一个大致倾向。

* 请求方法：POST
* 请求参数：JSON

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 含义 | 类型 | 备注 |
| Id | 电影编号 | string | 必填 |
| Name | 电影名称 | string | 必填 |
| Genre | 电影类型 | string | 必填 |
| Number | 数量 | int |  |

c)基于电影讨论人数的可视化

* 请求方法：POST
* 请求参数：JSON

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 含义 | 类型 | 备注 |
| Id | 电影编号 | string | 必填 |
| Name | 电影名称 | string | 必填 |
| Collect\_count | 观看人数 | int | 必填 |

### **4.3.3 模块内部核心算法**

由前面数据流图以及模块分析可以画出总体以及模块的IPO图，各IPO图如下所示：

反馈登录结果

选择操作类型

反馈分析结果

请求帮助

辅助帮助

用户登录管理

电影数据分析

使用帮助

用户信息

网页信息

用户反馈

图4-13系统总体IPO图

验证用户信息

注册用户信息

用户登录

注册

找回密码

注销

反馈登录结果

引导用户登录

用户信息

返回用户登录

用户名密码

图4-14用户登录管理IPO图

电影信息

分析结果

展示可视化信息

电影信息爬取

电影信息分析

电影信息可视化

爬取需求

网页信息

查询电影信息

图4-17 IPO图

图4-15电影数据分析IPO图

1. 用户登录功能程序流程图

对于用户登录管理，在用户输入账号后，系统会验证账号是否为空，当账号为空，系统提示并返回输入，直至账号不为空，验证账号是否存在，当账号不存在，系统提示并返回输入，直至账号存在；

用户输入密码验证密码是否为空，当密码为空，系统提示并返回输入，直至密码不空，验证密码是否存在正确，密码不正确返回重新输入，直至密码正确；

用户输入验证码后对验证码的正确进行判断，若验证码不正确且验证码未过期则返回重新获取，若不正确且验证码过期则返回重新获取验证码，直至验证码正确则提示登录成功。



图4-16用户登录管理程序流程图

1. 用户注册功能程序流程图：

在注册页面，用户输入联系方式，系统验证联系方式是否为空，当联系方式为空系统提示并返回输入，直至联系方式不空，系统验证联系方式是否正确，当联系方式错误，系统提示并返回输入，直至联系方式正确；

输入密码，系统验证密码是否为空，当密码为空系统提示并返回输入，直至密码不空，再次输入密码，系统验证密码是否为空，当密码为空系统提示并返回输入，直至密码不空，系统判断两次密码是否一致，若一致则可以点击获取验证码，若不一致则返回重新输入密码直至两次一致；

用户输入邮箱，验证邮箱是否为空，若为空则提示并返回输入，直至不为空，待用户选择性别后点击获取验证码，在用户输入验证码后对验证码的正确进行判断，若验证码不正确且验证码未过期则返回重新获取，若不正确且验证码过期则返回重新获取验证码，直至验证码正确则提示注册成功。



图4-17用户注册功能程序流程图

由上述用户登录和用户注册功能程序流程图可以画出下列简略盒图：

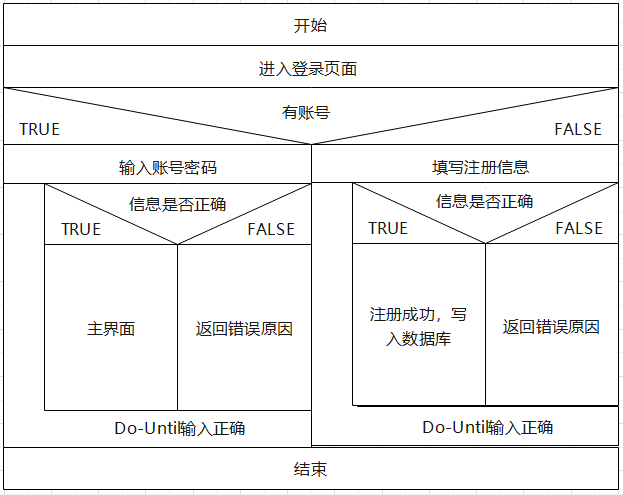


图4-18用户登录以及注册盒图

1. 用户注销功能程序流程图

在用户登录账号后，登录成功进入注销页面，填写注销申请，完成后弹出注销确认，若用户取消提交则返回填写页面，若用户确认则提示注销成功。



图4-19用户注销功能程序流程图

1. 用户找回密码功能程序流程图

用户进入找回密码页面，申请找回密码，输入用户名系统会验证用户名是否为空，当用户名为空，系统提示并返回输入，直至用户名不为空；

输入注册时填写的联系方式系统会验证联系是否为空，当联系方式为空，系统提示并返回输入，直至联系方式不为空；

系统验证验证联系方式是否正确，当联系方式不正确，系统提示并返回输入，直至联系方式正确点击获取验证码，在用户输入验证码后对验证码的正确进行判断，若验证码不正确且验证码未过期则返回重新获取，若不正确且验证码过期则返回重新获取验证码，直至验证码正确；

用户输入新密码，判断密码是否为空，若为空则提示并返回重新输入，若不为空则提示成功找回密码。



图4-20用户找回密码功能程序流程图

由上述各功能流程图可以得到用户登录管理部分的判定表和判定树：

表4-1用户登录管理判定表

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **用户名是否为空** | T | F | F | F | F | F | F | F |
| **输入用户名是否存在** |  | F | T | T | T | T | T | T |
| **输入密码是否为空** |  | F | T | F | F | F | F | F |
| **输入密码是否正确** |  | F | F | F | T | T | T | T |
| **输入验证码是否正确** |  | F | F | F | F | F | T | T |
| **验证码是否过期** |  | F | F | F | T | F | T | F |
| **输入用户名** | X | X |  |  |  |  |  |  |
| **输入密码** |  |  | X | X |  |  |  |  |
| **输入验证码** |  |  |  |  | X | X | X |  |
| **登录成功** |  |  |  |  |  |  |  | X |

图4-21用户登录管理判定树图

1. 用户查询信息功能程序流程图

用户登陆系统后，找到电影信息查询区域，在系统提供的查询框选择想查询的电影信息，数据爬取模块会根据用户的输入，爬取网页信息并将结果展示给用户；如果用户输入不合法，系统会报错提示。



图4-22用户查询信息功能程序流程图

# 5 编码与测试

概述段落

## 5.1 编码

### **5.1.1 编码规则简介**

### **5.1.2代表性模块示例**

## 5.2测试

概述段落

### **5.2.1 白盒测试**

#### 5.2.1.1 xx模块独立路径分析

#### 5.2.1.2 xx模块测试用例设计

#### 5.2.1.3 xx模块测试情况分析

### **5.2.2 黑盒测试**

#### 5.2.2.1 xx模块的等价类分析

#### 5.2.2.2 xx模块测试用例设计

#### 5.2.2.3 xx模块测试情况分析

# 6 系统使用说明

概述段落

## 6.1 系统运行环境和配置

## 6.2 系统操作说明（按照结构图或层次图的框架依次介绍）

### **6.2.1 XX1模块说明**

### **6.2.2 XX2模块说明**

### **6.2.3 XX3模块说明**

### **6.2.4 XX4模块说明**

# 7 总结

# 参考文献

按国家标准GB3469执行。

# 致谢