**第6章 系统开发与测试**

**6.1 系统开发**

**6.1.1 开发环境与开发工具**

本系统的开发环境和开发工具如下表所示。本系统使用了 Java 作为后端、数据库交互的开发语言；使用了 Python 语言作为论文数据爬取、论文数据分析的开发语言。

在集成开发环境上，使用了 Intellij Idea 作为 Java 语言的开发平台，使用了 PyCharm 作为 Python 的开发环境。

在数据库上，本系统默认使用命令行模式进行数据库的的管理。而系统的界面使用 Google Chrome 作为前端展示平台，而大部分效果都可以在 webkit 内核下的浏览器进行查看。

|  |  |
| --- | --- |
| 软硬件项目 | 配置情况 |
| Java运行操作系统 | 多平台操作系统 |
| Python 运行操作系统 | Windows 没有问题，其它的过后确认 |
| 前后端编程语言 | Java |
| 论文数据方面编程语言 | Python |
| 数据库 | MySQL |
| CPU |  |
| RAM |  |
| ROM |  |
| Client | Webkit 内核浏览器全效果，其它内核浏览器可能（没测试）出现问题 |

**6.1.2 系统注册登录功能的实现**

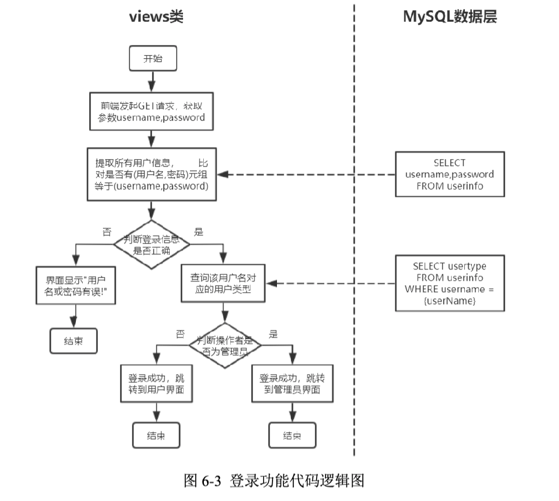
*本系统的注册界面如图6-1所示。系统的注册界面要求用户设置自己的用户名和密码并选择用户类型（一般浏览用户、B站视频创作者）。点击“注册”前需勾选“我同意用户协议”，否则会弹出警告框“必须勾选同意协议”。 三个注册信息都必须填上，如有未填信息则模块下侧会显示红字“不能有输入为空!”。用户名是一条用户信息的主键，也就是说不能有相同的用户名。如果注册了重复的用户名，则模块下侧会显示红字“该用户名已存在!”。如果各种信息都输入无误，则下侧会显示红字“注册成功!!!”，同时数据库中录入新用户信息。点击下方的“返回登录界面”可跳转到登录界面。*

*\*界面截图\**

*本系统的登录界面如图6-2所示。系统的登录界面给用户和管理员提供了系统入口。在页面中输入用户名和密码并点击“登录”键后，如果用户名和密码对应上了，则根据用户类型决定跳转到用户界面还是管理员界面；如果没有对应上，则模块下侧会显示红字“用户名或密码有误!”。点击下方的“进入注册界面”可跳转到注册界面。*

*\*界面截图\**

*登录界面的功能代码处理逻辑如图 6-3 所示。主要步骤为：在前端接收username和password参数，views层验证用户名和密码的正确性，再检查用户类型以决定跳转到用户界面还是管理员界面。用户名或密码有误则返回错误提示。*

*\*代码逻辑图\**

**6.1.3 系统用户界面的实现**

*本系统的用户界面如图6-4所示。用户（一般浏览用户、B站视频创作者）登录后进入的界面为用户界面。该界面的主体部分为视频信息列表。点击视频名称可直接链接到原B站视频网页。点击某视频后的“加入收藏夹”可收藏该视频。页眉处还有“用户推荐视频”和“修改用户信息”的衍生界面跳转*

*\*界面截图\**

*如图6-5所示，点击页眉处的“我收藏的视频”可查看当前用户收藏的所有视频，并且“加入收藏夹”按键变成“移出收藏夹”按键。*

*按分区筛选视频的功能代码处理逻辑如图6-6所示。点击左侧栏的某个视频分区可使右侧的视频表中仅显示该分区的视频信息。主要步骤为：在前端接收video\_type参数，views层筛选出视频信息表中所有视频分区等于video\_type的数据，然后将筛选后的数据返回到前端界面的视频表上，本系统仅研究B站16种视频分区。*

*\*界面截图\**

***\*代码逻辑图\****

*如图6-7所示，选中某个视频（点击视频前的圆点）后，点击“绘制弹幕词云”，页面会弹出该视频的弹幕词云图。同理如图6-8所示，选中某个视频后，点击“弹幕情感分析”，页面会弹出三张与弹幕情感分析有关的统计图（情感柱状图、情感趋势图、情感雷达图）。*

*本系统的弹幕表页如图6-9所示。选中某个视频后，点击“查询弹幕信息”可弹出该视频的弹幕表页。弹幕表页显示的与弹幕有关的内容有：发送时间、发送者ID、弹幕内容、情感分值、情感等级、举报次数。如果查询用户认为某条弹幕的内容违法违规、大量刷屏等，可点击该弹幕后的“举报弹幕”按键，点击后“举报次数”字段值就加1。弹幕表页还提供评论弹幕表的功能，在评论框内输入留言并点击“提交评论”后，留言信息会连同当前登录的用户名实时显示在下方的评论区内并存入数据库中，下次打开该弹幕表页时留言信息依然存在。*

*举报弹幕的功能代码处理逻辑如图 6-10 所示。主要步骤为：在前端接收tableName 和 rowID 参数，views 层通过两个参数查找到该条弹幕的举报次数marked并加一，然后在数据库中更新数据并返回到原弹幕表页上。*

*\*界面截图\**

***\*代码逻辑图\****

**6.1.4 系统管理员界面的实现**

*本系统的管理员界面如图6-11所示。管理员登录后进入的界面为管理员界面。该界面的主体部分也是视频信息列表，展示的信息除了视频名称、视频分区、发布时间外还有视频cid码。点击视频名称也可链接到原B站视频。页眉处还有“管理推荐视频”和“修改情感词典”的衍生界面跳转*

*左侧栏有“添加视频信息”和“修改视频分区”模块。如图6-12所示，在两模块的输入框输入信息并提交后，操作是否成功都会给出红色字提示，并且操作结果都会实时显示在右边的视频表上*

*左侧栏下方还有“追加弹幕数据”、“更新情感得分”和“删除视频信息”三个功能按键。如图6-13所示，选中视频表中的某个视频再点击其中的一个功能按键，即可对该视频进行按键上的操作，操作完毕后会弹出操作成功的提示框。*

*如图6-14所示，点击页眉处的“视频统计图”会弹出各分区已有视频数量和推荐视频数量的柱状统计图。*

*添加视频信息的功能代码处理逻辑如图6-15所示。主要步骤为：在前端接收BV和video\_type参数，views层判断输入是否符合要求，符合要求则运行视频信息爬虫程序，程序运行正常则界面显示“添加成功!”并录入新视频信息。如果是其他情况则界面显示“添加失败!”*

*\*界面截图\**

***\*代码逻辑图\****

**6.1.5 系统衍生界面的实现**

*正如前文的界面图所示，用户界面有“用户推荐视频”和“修改用户信息”两个衍生界面跳转，管理员界面有“视频推荐管理”和“修改情感词典”两个衍生界面跳转。下面是这些系统衍生界面的介绍。 “用户推荐视频”衍生界面如图6-16所示。在右侧输入视频BV号和视频分区并点击“提交推荐”可推荐自己喜欢的B站视频给管理员，左侧则会实时显示各分区的推荐次数。*

*“修改用户信息”衍生界面如图6-17所示。输入新密码或新用户类型并提交，就可更新当前登录用户的密码或用户类型。*

*“管理推荐视频”衍生界面如图6-18所示。用户在“用户推荐视频”界面推荐的视频信息管理员可以在此界面看到。管理员可以根据系统的需要添加用户推荐的部分视频，点击视频信息后的“添加”键即可。*

*“修改情感词典”衍生界面如图6-19所示。点击右侧不同的按键可在左侧查看到不同的情感词表。在左侧下方输入想增加的情感词汇信息可新增相应词表的词汇。在右侧表中点击某词汇后的“删除”键可删除该词汇。*

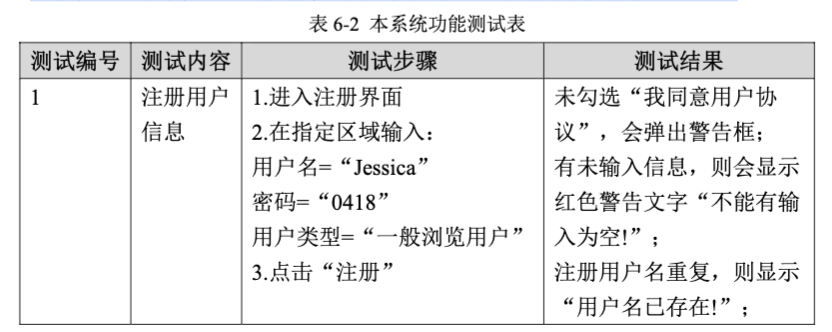
*\*界面截图\**

***\*代码逻辑图\****

**6.2 系统测试**

*在系统开发完成并投入使用之前，必须对系统进行软件测试，以发现系统代码可能存在的bug或运行异常，从而降低未来的系统维护成本。本系统的测试用例主要涵盖：系统注册与登录功能、视频筛选功能、弹幕情感分析功能、视频推荐管理、添加视频功能、用户信息管理。*

*本系统的测试主要以各模块的系统设计方案为基础设计模块测试的内容。本系统采用了动态测试的方法，即通过系统代码的执行结果来测试系统，而非通过直接检查系统代码逻辑来评价系统开发成果。系统功能测试表如表6-2。*



**6.3 系统界面开发难点与解决方案**

在本系统的实现过程中，开发的难点主要有如何从多重认证的网页上爬取数据，比如从学校的数据库；还有如何配置好一个多平台操作的 Java 项目，确保在不同的平台上可以运行； 如何开发一个自适应的网页。

**6.3.1 从多重认证的网页上爬取数据**

待编写

**6.3.2 配置一个多平台的 Java 项目**

难点

操作系统差异：

不同的操作系统（如Windows、Linux、macOS等）具有不同的文件路径、环境变量、网络配置等，这可能导致Java项目在不同平台上出现兼容性问题。

依赖库管理：

Java项目通常依赖于多个外部库和框架。这些库和框架可能在不同平台上有不同的版本要求或特定的安装步骤，增加了配置的复杂性。

数据库兼容性：

如果Java项目涉及到数据库操作，不同数据库系统（如MySQL、Oracle、SQL Server等）之间的SQL语法、数据类型和性能特性可能存在差异，需要针对不同平台进行适配。

内存和性能调优：

不同平台的硬件资源（如CPU、内存、磁盘等）存在差异，这可能需要针对特定平台进行内存分配、垃圾回收等性能调优。

安全性问题：

多平台部署可能增加安全风险，如不同平台的安全策略、防火墙设置、加密方式等可能不同，需要确保Java项目在所有平台上都符合安全标准。

解决方案

使用构建工具：使用Maven、Gradle等构建工具可以自动化管理项目依赖、编译、打包等过程，确保在不同平台上使用相同版本的依赖库和框架。

配置管理：使用Ansible、Chef、Puppet等配置管理工具可以自动化部署和配置Java项目，确保在不同平台上具有相同的配置和环境设置。

编写跨平台代码：避免使用特定于操作系统的API和文件路径，使用Java的跨平台特性（如使用System.getProperty("os.name")来获取操作系统类型）来编写可移植的代码。

数据库抽象层：使用JDBC或ORM框架（如Hibernate、MyBatis）作为数据库抽象层，可以屏蔽不同数据库之间的语法和性能差异，简化数据库操作。

性能监控和调优：使用JProfiler、VisualVM等性能监控工具来监控Java项目的运行状况，并根据需要进行内存分配、垃圾回收等调优。此外，还可以利用Docker等容器化技术来隔离不同平台的环境差异，简化性能调优过程。

安全性措施：确保Java项目遵循最佳的安全实践，如使用HTTPS协议进行通信、对敏感信息进行加密存储和传输、定期更新依赖库以修复安全漏洞等。同时，针对不同平台的安全策略进行适应性调整，确保项目在所有平台上都符合安全标准。

**6.3.3 自适应网页**

难点

多设备适配：不同设备具有不同的屏幕尺寸、分辨率和操作系统，这使得开发者需要考虑到各种设备的特点和差异，以确保网页在各种设备上都能良好显示。

性能优化：自适应网页通常需要加载更多的样式和脚本文件，以适应不同设备的显示需求。这可能导致页面加载速度变慢，影响用户体验。

交互设计：移动设备与桌面设备在交互方式上存在差异，如触摸操作、手势识别等。开发者需要设计出既适合桌面设备又适合移动设备的交互方式。

测试与调试：由于需要适配多种设备，测试和调试的复杂性大大增加。开发者需要在不同的设备和浏览器上进行测试，以确保网页在各种环境下的兼容性。

内容布局：如何在有限的屏幕空间内合理布局内容，同时保持信息的完整性和可读性，是自适应网页设计的关键难点。

解决方案

使用CSS媒体查询：CSS媒体查询可以根据设备的屏幕尺寸、分辨率等条件来应用不同的样式规则。通过使用媒体查询，我们可以为不同的设备设计不同的布局和样式，从而实现自适应显示。

优化图片和文件资源：对图片和文件进行压缩、合并和懒加载等优化操作，可以减少页面加载时间，提高用户体验。同时，使用CDN（内容分发网络）可以加速资源的加载速度。

设计移动端优先的交互方式：在设计交互方式时，优先考虑移动设备的特性，如触摸操作、手势识别等。对于需要用到鼠标和键盘操作的桌面设备，可以采用辅助手段或提供备选方案。

使用自动化测试工具：利用自动化测试工具（如Selenium、Appium等）可以在不同的设备和浏览器上进行自动化测试，提高测试效率。同时，利用跨浏览器测试平台（如BrowserStack、CrossBrowserTesting等）可以测试网页在各种浏览器上的兼容性。

采用流式布局和弹性盒子模型：流式布局可以根据屏幕宽度自动调整元素的宽度和位置，而弹性盒子模型则提供了更加灵活和强大的布局控制能力。通过结合使用这两种布局方式，我们可以在有限的屏幕空间内合理布局内容，同时保持信息的完整性和可读性。

考虑渐进式增强和优雅降级：渐进式增强意味着先为所有用户提供基本的功能和体验，然后再为支持更高级功能的设备提供额外的功能和体验。而优雅降级则意味着先为支持更高级功能的设备提供完整的功能和体验，然后再为不支持这些功能的设备提供降级后的版本。通过这两种策略，我们可以确保网页在各种设备上都能提供相对一致的用户体验。