



Python 程序设计

Read-Book “智能知识侦查助手”(9)

2022-2023-1

任课教师: 张涵翠

王旭刚 2020329621074

韦杨婧 2020329621199

池胤杰 2020329621049

计算机科学与技术全英文授课班 20(1)

王旭刚^{1*}, 韦杨婧², 池胤杰³

1* Leader, back-end and webcrawler design and implementation

2 Team member, front-end design and implementation

3 Team member, question data collect, data cleaning

Abstract

Our team's course design is called "Read-Book", an Intelligent Knowledge Scouting Assistant. Our work can be roughly devided to these parts: 1. We used webcrawler to gather data of the top 200 rated books from book.douban.com. and store data into database. 2. Based on those 200 books, we developed different ways to collect and generate question about them. 3. Derive templates to pick questions to form questionnaire. 4. Create GUI interface through Web-App. 5. Analysis, assess and evaluate user statistics from multi angles. 6. Implement an Intuitive Data Visualization for user.

We used *Python* as the main programming language. Use *flask* library to manipulate html, css, javascript for front-end; *requests*, *BeautifulSoup4* library for webcrawling data; and *pymysql* library for back-end programming. For more information, you can click *here* to view our Github Repository and click *here* to view our product page.

Key words: Python, Webcrawler, Question/Questionnaire Generating, Object-Orinted Programming⁽¹⁾, Web-App

课题执行 (文档修订) 记录表

Table 1: 课题执行 (文档修订) 记录表 (按姓氏拼音顺序排列)

| 日期 | 版本 | 修订内容 | 执行人 | 修订人 |
|------------|-----|---------------|-------------|-----|
| 2022.11.06 | 1.0 | 分析需求 | 池胤杰、王旭刚、韦杨婧 | 王旭刚 |
| 2022.11.07 | 1.1 | 系统框架设计 | 池胤杰、王旭刚、韦杨婧 | 王旭刚 |
| 2022.11.08 | 1.2 | 用户、书籍数据库设计 | 王旭刚、韦杨婧 | 韦杨婧 |
| 2022.11.08 | 1.3 | 网页模板修改 | 韦杨婧 | 韦杨婧 |
| 2022.11.08 | 1.4 | 实现登录注册 | 韦杨婧 | 韦杨婧 |
| 2022.11.08 | 1.5 | 查看个人信息功能 | 韦杨婧 | 韦杨婧 |
| 2022.11.09 | 1.6 | 设计书籍页面 | 韦杨婧 | 韦杨婧 |
| 2022.11.09 | 1.7 | 设计知识测试页面 | 韦杨婧 | 韦杨婧 |
| 2022.11.09 | 1.9 | 总结、问题添加 | 王旭刚 | 王旭刚 |
| 2022.11.16 | 2.1 | 修改 README | 王旭刚 | 韦杨婧 |
| 2022.11.17 | 2.2 | 完善摘要 | 王旭刚 | 王旭刚 |
| 2022.11.17 | 2.3 | 问题数据增加至前 29 本 | 池胤杰 | 王旭刚 |
| 2022.11.17 | 2.4 | 前后台数据传递 | 韦杨婧 | 韦杨婧 |
| 2022.11.18 | 3.1 | 成功在服务器部署 | 王旭刚 | 韦杨婧 |
| 2022.11.19 | 3.2 | 问题数据增加至 200 本 | 池胤杰 | 王旭刚 |
| 2022.11.20 | 3.3 | 实现了数据展示界面 | 韦杨婧 | 韦杨婧 |
| 2022.11.20 | 3.4 | 实现了批改试卷 | 王旭刚 | 王旭刚 |
| 2022.11.21 | 3.5 | 总结、问题添加 | 王旭刚 | 王旭刚 |

Contents

| | |
|---|-----------|
| Author | I |
| Abstruct | I |
| 课题执行 (文档修订) 记录表 | II |
| 1 系统设计 | 1 |
| 1.1 概要设计 | 1 |
| 1.1.1 系统框架设计 | 1 |
| 1.1.2 系统功能设计 | 1 |
| 1.2 功能设计 | 2 |
| 1.2.1 功能 1 – 书籍/问题数据的获取 | 2 |
| 1.2.2 功能 2 – 用户注册/登录/登出 | 2 |
| 1.2.3 功能 3 – 全部书籍信息概览 | 2 |
| 1.2.4 功能 4 – 书籍信息测试 | 2 |
| 1.2.5 功能 5 – 测试结果分析以及可视化展示 | 3 |
| 1.3 数据库设计 | 3 |
| 1.3.1 数据表 1 – 书籍 | 3 |
| 1.3.2 数据表 2 – 用户 | 4 |
| 1.3.3 数据表 3 – 题目 | 5 |
| 1.3.4 数据表 4 – 测试结果 | 5 |
| 2 系统实现 | 6 |
| 2.1 前端实现 | 6 |
| 2.1.1 页面 1 – Log/Register/Profile | 7 |
| 2.1.2 页面 2 – Home | 7 |
| 2.1.3 页面 3 – Book | 7 |
| 2.1.4 页面 4 – Test | 8 |
| 2.1.5 页面 5 – Statistics | 9 |
| 2.2 后端实现 – UML 类图 | 11 |
| 2.3 前后端交互 | 12 |
| 2.3.1 用户注册/登录/登出 | 12 |
| 2.3.2 全部书籍信息预览 | 13 |
| 2.3.3 书籍信息测试 | 14 |
| 2.3.4 测试结果分析以及可视化展示 | 15 |

| | |
|---------------------|-----------|
| 3 结束语 | 16 |
| 3.1 总结 | 16 |
| 3.2 不足与展望 | 16 |
| Reference | 17 |

1. 系统设计

1.1 概要设计

1.1.1 系统框架设计

本项目为一个基于 python 的 web 应用，主要由前端和后端组成，前端主要负责用户界面的展示，后端主要负责数据的处理和存储。前端使用 flask 框架，同时使用 html、css、javascript，后端使用 python 语言，数据库使用 mysql。

在书籍数据，问题数据获取方面，使用了 python 的 requests、selenium 库，通过爬虫的方式获取数据，再使用 BeautifulSoup4、re 等库进行数据处理与清洗，最后使用 python 的 pymysql 库将数据存储到 mysql 数据库中。

在后端的全部设计中，我们在动手前先进行了讨论，确定了我们的系统的架构，以及各个模块的功能，最后确定了我们的系统的框架图，如图1所示，整体体现了面向对象的设计思想。

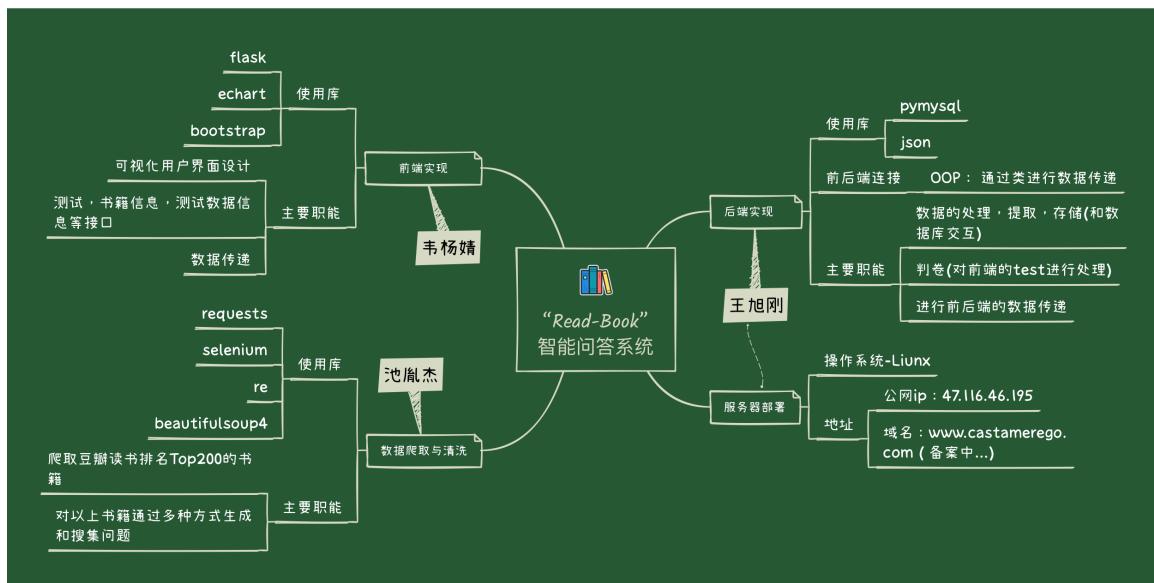


Figure 1: 系统框架设计

1.1.2 系统功能设计

经过需求分析和头脑风暴过后，我们共同确定了我们的系统的功能：

- 书籍/问题数据的获取
- 用户注册/登录/登出
- 全部书籍信息概览

- 书籍信息测试 (按照多种方式: 按照书籍、按照题目类型、全部题库随机抽样等)
- 测试结果分析以及可视化展示

1.2 功能设计

1.2.1 功能 1 – 书籍/问题数据的获取

该部分主要是通过爬虫的方式获取数据，然后进行数据清洗，最后将数据存储到数据库中。首先是爬取豆瓣的 Top200 书籍信息，再根据爬出的书籍通过多种方式获取书籍的问题数据，最后将数据存储到数据库中。

获取问题时，有以下几种方式：

- 通过数据库中的书籍信息，生成概念类问题，如：《三体》的作者是谁？
- 通过手动信息搜集，如搜集百度文库，豆瓣评论等

1.2.2 功能 2 – 用户注册/登录/登出

该部分主要是实现用户的注册、登录、登出功能。主要需要实现的功能有：

- 用户注册 – 用户输入用户名、密码、邮箱等信息，注册成功后，将用户信息存储到数据库中
- 用户登录 – 用户输入用户名、密码，登录成功后，将用户信息存储到当前对话中
- 用户登出 – 用户登出后，将用户信息从当前对话中删除

1.2.3 功能 3 – 全部书籍信息概览

该部分主要实现对书籍信息的概览，主要包括全部 (200 本) 书籍的信息概览，以及每本书的详细信息显示，在书籍的详细信息显示界面将加入针对单本书籍的信息测试，详见功能 4。

1.2.4 功能 4 – 书籍信息测试

该部分主要实现对书籍信息的测试，使用在功能 1 中爬取的书籍信息，通过多种方式进行测试，包括：

- 按照书籍进行测试 – 用户选择一本书，然后进行测试

- 按照题目类型进行测试 – 用户选择一种题目类型，然后进行测试
- 全部题库随机抽样 – 用户选择随机抽样，然后进行测试

1.2.5 功能 5 – 测试结果分析以及可视化展示

该部分主要实现对测试结果的分析以及可视化展示，主要包括：

- 测试结果分析 – 对测试结果进行分析，包括正确率、错误率、各种类别题目的正确、错误条数、答题时长。以及对单次和多次的测试结果梳理
- 测试结果可视化展示 – 对测试结果进行可视化展示，包括雷达图，柱状图，分析文本等

1.3 数据库设计

在本项目中，所有数据库均使用 sqlserver 数据库，以 Python 的 *pymysql* 库进行操作，共有四个表，分别为书籍、用户、题目、测试结果表。

1.3.1 数据表 1 – 书籍

书籍数据库主要用于豆瓣爬取出的 200 本书籍的信息，包括书名、作者、出版社、出版时间、评分、评价人数、标签、简介、题目等信息。种类可大致分为中国与外国书籍，其在部分条目中有些许差异，具体如表2所示。我们将其在豆瓣中的排名作为书籍的 id，以便于后续的操作。

Table 2: 书籍储数据存—数据库设计

| Attribute | Translation | Type | Chinese Book | Foreign Book |
|---------------|-------------|---------|--------------|--------------|
| Id | 排名 | int | ✓ | ✓ |
| Name | 书名 | varchar | ✓ | ✓ |
| Author | 作者 | varchar | ✓ | ✓ |
| Country | 国家 | varchar | ✓ | ✓ |
| Publisher | 出版社 | varchar | ✓ | ✓ |
| Year | 出版日期 | varchar | ✓ | ✓ |
| Page | 页数 | varchar | ✓ | ✓ |
| Price | 定价 | varchar | ✓ | ✓ |
| Frame | 装帧 | varchar | ✓ | ✓ |
| Category | 丛书 | varchar | ✓ | ✓ |
| Isbn | isbn 码 | varchar | ✓ | ✓ |
| Star | 评分 | float | ✓ | ✓ |
| Comment num | 评价数量 | int | ✓ | ✓ |
| Brief | 简介 | varchar | ✓ | ✓ |
| Douban bookid | 豆瓣 id | varchar | ✓ | ✓ |
| Link | 链接 | varchar | ✓ | ✓ |
| Name o | 原作名 | varchar | ✗ | ✓ |
| Trans | 译者 | varchar | ✗ | ✓ |

1.3.2 数据表 2 – 用户

用户数据库主要用于用户的注册、登录、登出等操作，包括用户名、密码、电话、性别、个人简介等信息。具体如表3所示。

Table 3: 用户数据储存—数据库设计

| Attribute | Type | Translation |
|-----------|---------|-------------|
| Id | int | 编号 |
| Name | varchar | 姓名 |
| Gender | char | 性别 |
| Telephone | varchar | 电话 |
| Password | varchar | 密码 |
| Brief | varchar | 简介 |

1.3.3 数据表 3 – 题目

题目数据库主要用于题目的储存，包括题目、选项、答案、问题类别、书籍 id、内容类别等信息。具体如表4所示。其中选项至多为 4 项，问题的种类有单选题和判断题，内容类别有书籍简介、主要人物、作者、情节等，具体如表5所示。

Table 4: 问题储存—数据库设计

| Attribute | Type | Translation | Note |
|--------------|---------|-------------|------------------|
| Id | int | 关联书本的 id | |
| Question | varchar | 问题 | |
| Type | int | 类别 | 0-判断, 1-单选 |
| Question num | int | 选项数量 | |
| C1 | varchar | 选项 1 | |
| C2 | varchar | 选项 2 | |
| C3 | varchar | 选项 3 | |
| C4 | varchar | 选项 4 | |
| Ans | varchar | 正确选项 | 必须与某个选项完全相同 |
| Category | varchar | 问题内容的类别 | 如: 问作者, 书中人物, 情节 |

Table 5: 问题内容类别

| Type | Translation |
|----------|-------------|
| figure | 人物 |
| writer | 作者 |
| main | 主旨 |
| content | 情节 |
| detail | 细节 |
| anecdote | 其他 |

1.3.4 数据表 4 – 测试结果

测试结果数据库主要用于测试结果的储存，包括用户 id、书籍 id、测试结果、测试时间等信息。具体如表6所示。

Table 6: 测试数据储存—数据库设计

| Attribute | Type | Translation |
|-------------|-------|--------------------|
| UserId | int | 用户 id |
| Score | float | 得分 |
| Start | int | 开始时间 (使用 unix 时间戳) |
| End | int | 结束时间 (使用 unix 时间戳) |
| Duration | int | 答题时长 (秒) |
| A1 | int | 人物形象相关答对个数 |
| A2 | int | 作者相关答对个数 |
| A3 | int | 主题相关答对个数 |
| A4 | int | 情节相关答对个数 |
| A5 | int | 细节相关答对个数 |
| B1 | int | 人物形象相关题目总数 |
| B2 | int | 作者相关题目总数 |
| B3 | int | 主题相关题目总数 |
| B4 | int | 情节相关题目总数 |
| B5 | int | 细节相关题目总数 |
| questionnum | int | 问题数量 |
| rightnum | int | 正确数量 |
| wrongnum | int | 错误数量 |
| emptynum | int | 未回答数量 |

2. 系统实现

2.1 前端实现

本次设计基于 Flask 框架实现前后端分离。后端使用 Flask 框架，用路由绑定函数，实现跳转至不同页面时运行对应函数。前端使用 Bootstrap 框架，使用 HTML5 和 CSS3 实现页面布局和样式，使用 JavaScript 和 jQuery 实现页面交互效果。

前端页面按照排版可主要分为三个部分，分别是标签栏，内容栏和尾部。标签栏处设定主页（Home）、书籍页（Book）、测试页（Test）、数据页（Statistics）、个人页（Profile）。内容栏根据不同的页面显示不同的内容。尾部处显示版权信息。而按照功能可分为主页、书籍概览页、测试页、数据页和登录、注册、个人信息页。主页、书籍概览页无需登录即可查看，而测试页、数据页、则需要登陆后才能进行下一步操作。

2.1.1 页面 1 – LOG/REGISTER/PROFILE

进入网站后，点击 login 可进入登录界面，已注册用户可选择直接登录；未注册用户可选择先注册后登录。登陆后可点击标签栏的“profile”查看个人信息。

2.1.2 页面 2 – HOME

主页显示网站的设置了滚动播放。播放的内容分别为每日推荐读物（如图2）和测试入口（如图3）。点击每日推荐读物的详情标签可进入对应书籍的详细信息页；点击测试则进入测试页。



Figure 2: 每日推荐读物



Figure 3: 测试入口

2.1.3 页面 3 – BOOK

书籍页显示豆瓣排名前 200 的书籍排名、书名、作者和对应图片等摘要信息以及跳转按钮（如图4）。点击按钮可进入对应书籍的详细信息页，除了摘要信息外还包含国家、出版社、出版年份、isbn 码、豆瓣评分、简介等（如图5）。

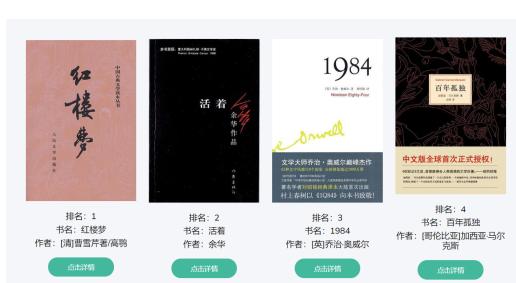


Figure 4: 书籍信息页（摘要）



Figure 5: 书籍详细信息页

未登录状态下，显示返回按钮；登录状态下则在此基础上额外显示测试按钮。点击返回可返回到 200 本书籍的摘要页面；点击测试，可以跳转到测试界面，测试内容仅与该书相关。

2.1.4 页面 4 – TEST

若未登录时直接点击测试页，网页跳转至登录界面，登陆后自动跳转到测试界面。此时用户可以选择测试题目的类型（共分为人物形象类、作者类、主旨类、情节类、细节类五个类别），点击开始测试即可做题。

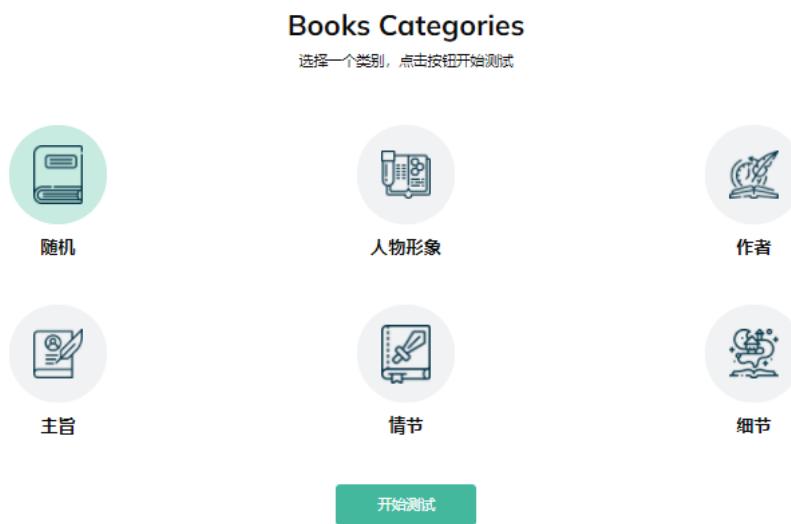


Figure 6: 选择类别进行测试

答题完毕后可以整卷阅览或提交。



Figure 7: 测试完毕可选整卷阅览或提交

提交后将显示批卷结果，用户可以看到自己的答题情况。选择正确则选项显示绿色边框。

24

题目：在《无人生还》中，八个素不相识的人受邀来到海岛黑人岛上。他们抵达后，接待他们的却只是管家罗杰斯夫妇俩。用晚餐的时候，餐厅里的留声机忽然响起，指控他们宾客以及管家夫妇这十人都曾犯有谋杀罪。众人正在惶恐之际，来宾之一忽然死亡，噩梦由此开始了。

A. true

B. false

上一题

下一题

Figure 8: 选择正确

选择错误则用户选择的选项显示红色边框，正确答案显示绿色边框。

题目：《蜀相》的作者是()

21

A. 秦韬玉

B. 杜甫

C. 贺知章

D. 李白

上一题

下一题

Figure 9: 选择错误

未选择则正确选项显示蓝色边框。

题目：在《鼠疫》中，瘟疫城市被重重封锁，无人能够自由进出。被困在城中的人民，朝思暮想着住在城外的亲朋好友。一位到城公干的记者被迫过着无亲无友的生活，只有寄望参与自愿队消磨时间。

23

A. true

B. false

上一题

下一题

Figure 10: 未选择

2.1.5 页面 5 – STATISTICS

与测试页相同，若未登录时直接点击数据页，网页跳转至登录界面，登陆后自动跳转到数据界面。若用户无历史测试数据，网页会提示用户先进行测试，如图11。

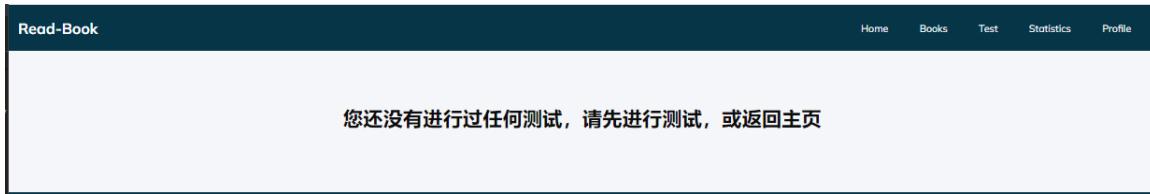


Figure 11: 无历史测试数据

若用户有历史测试数据，网页显示相关数据图与分析文字。其中进度条显示用户多次测试的累计正确率。雷达图则显示用户在五个类别下的累计正确率和最近一次测试的正确率。根据用户的累计正确率，将生成对应分析文本以及名人名言，如图12



Figure 12: 历史数据页面

2.2 后端实现 – UML 类图

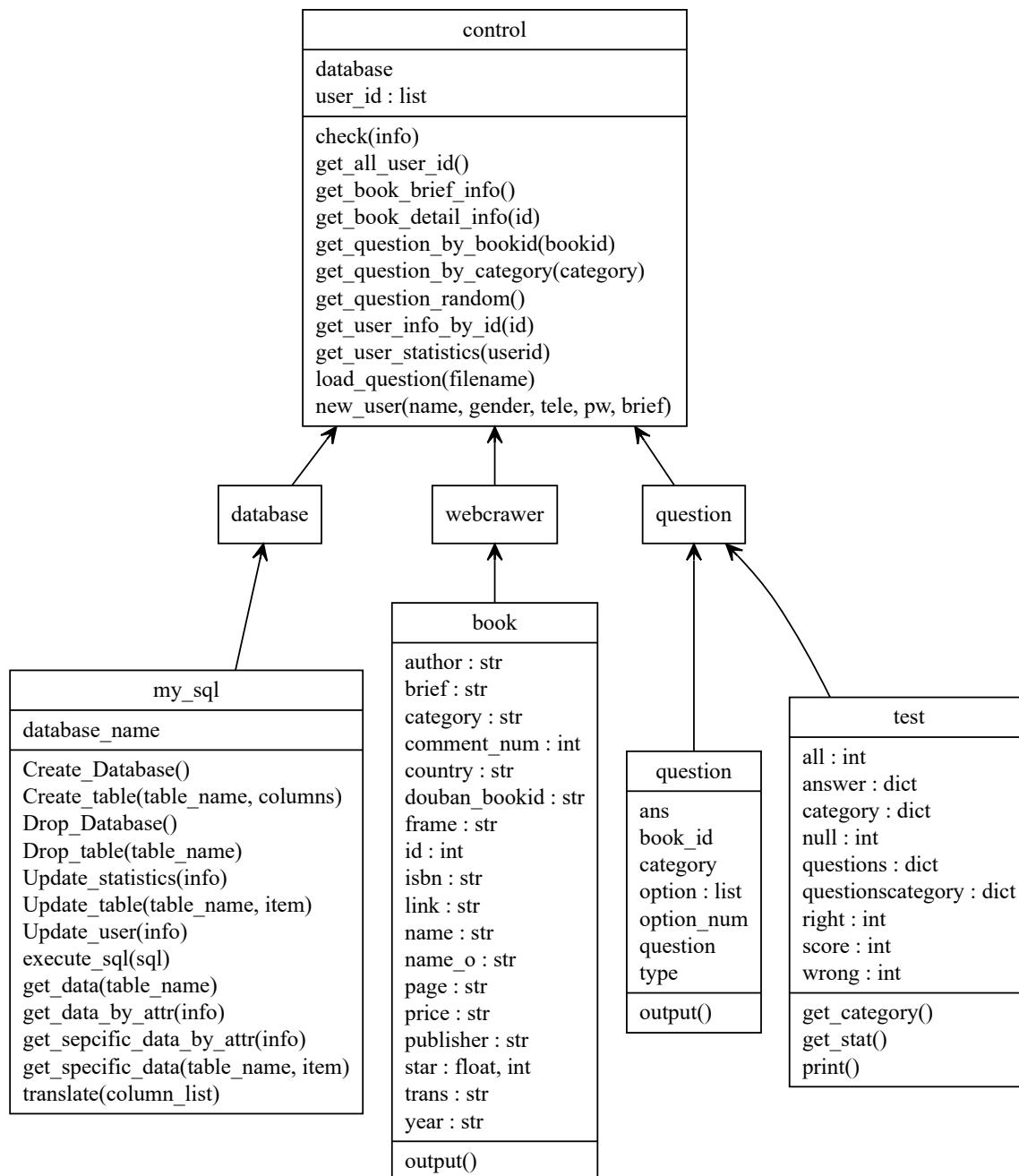


Figure 13: 后端 UML 图

2.3 前后端交互

2.3.1 用户注册/登录/登出

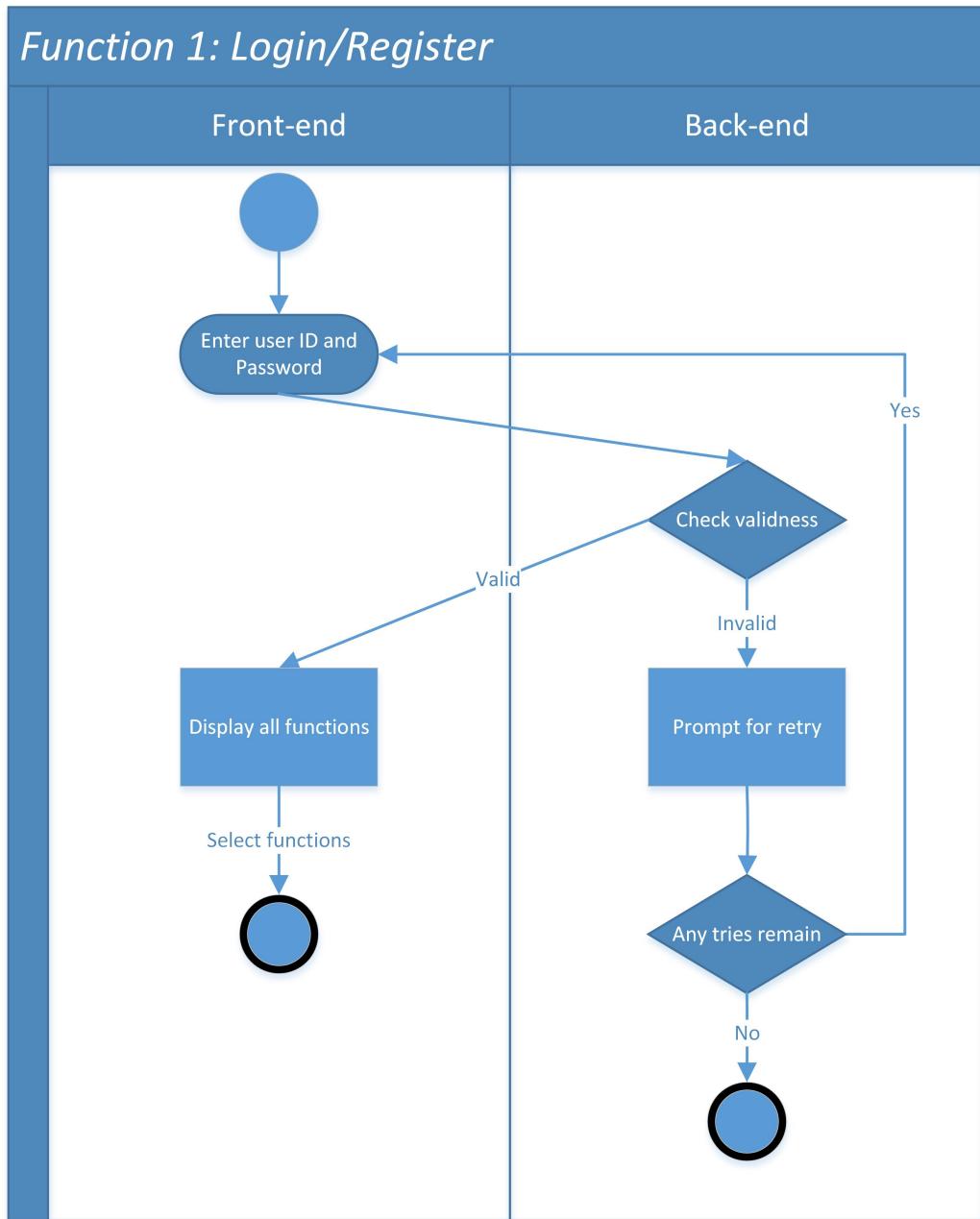


Figure 14: 前后端交互：用户注册/登录/登出

2.3.2 全部书籍信息预览

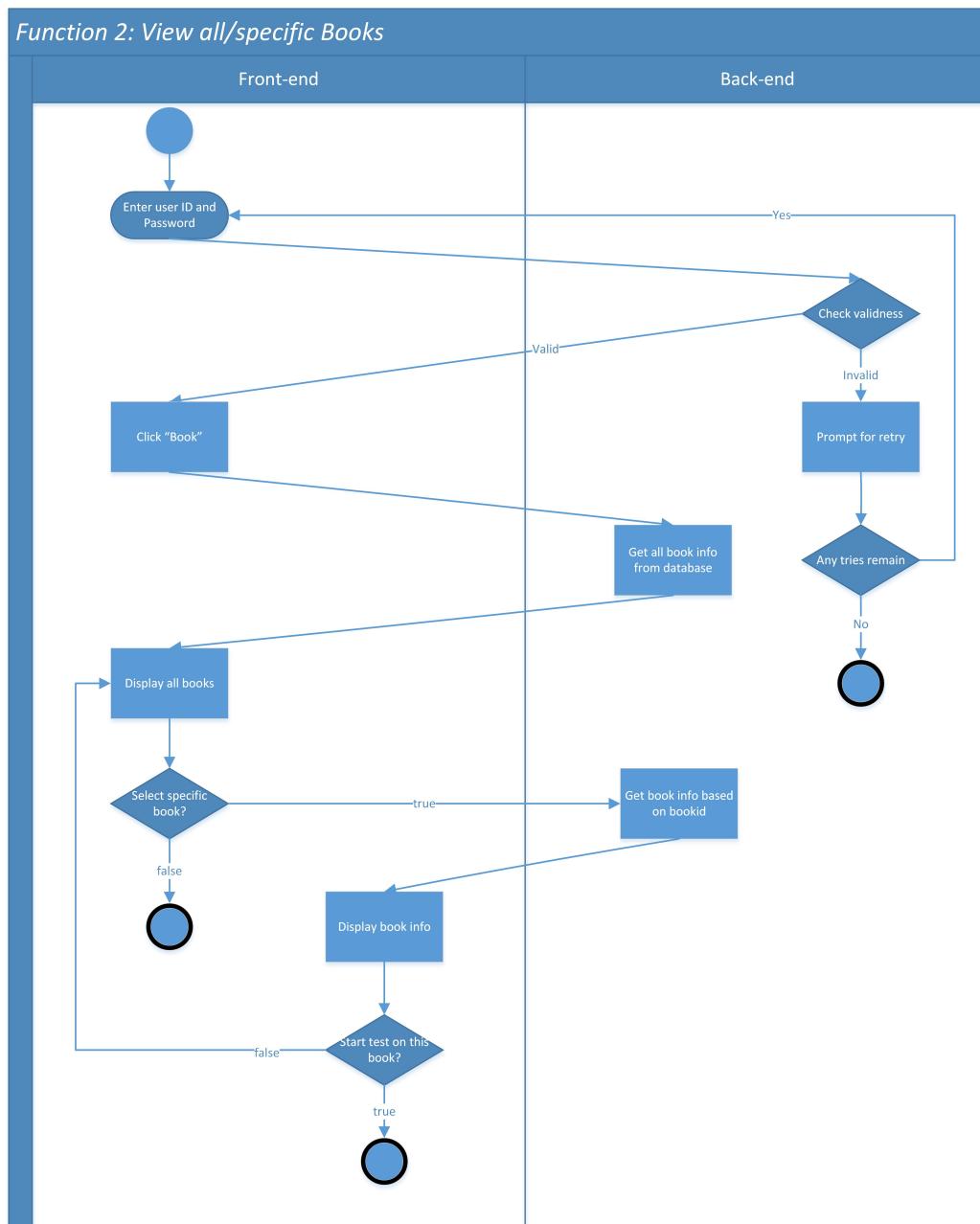


Figure 15: 前后端交互：全部书籍信息预览

2.3.3 书籍信息测试

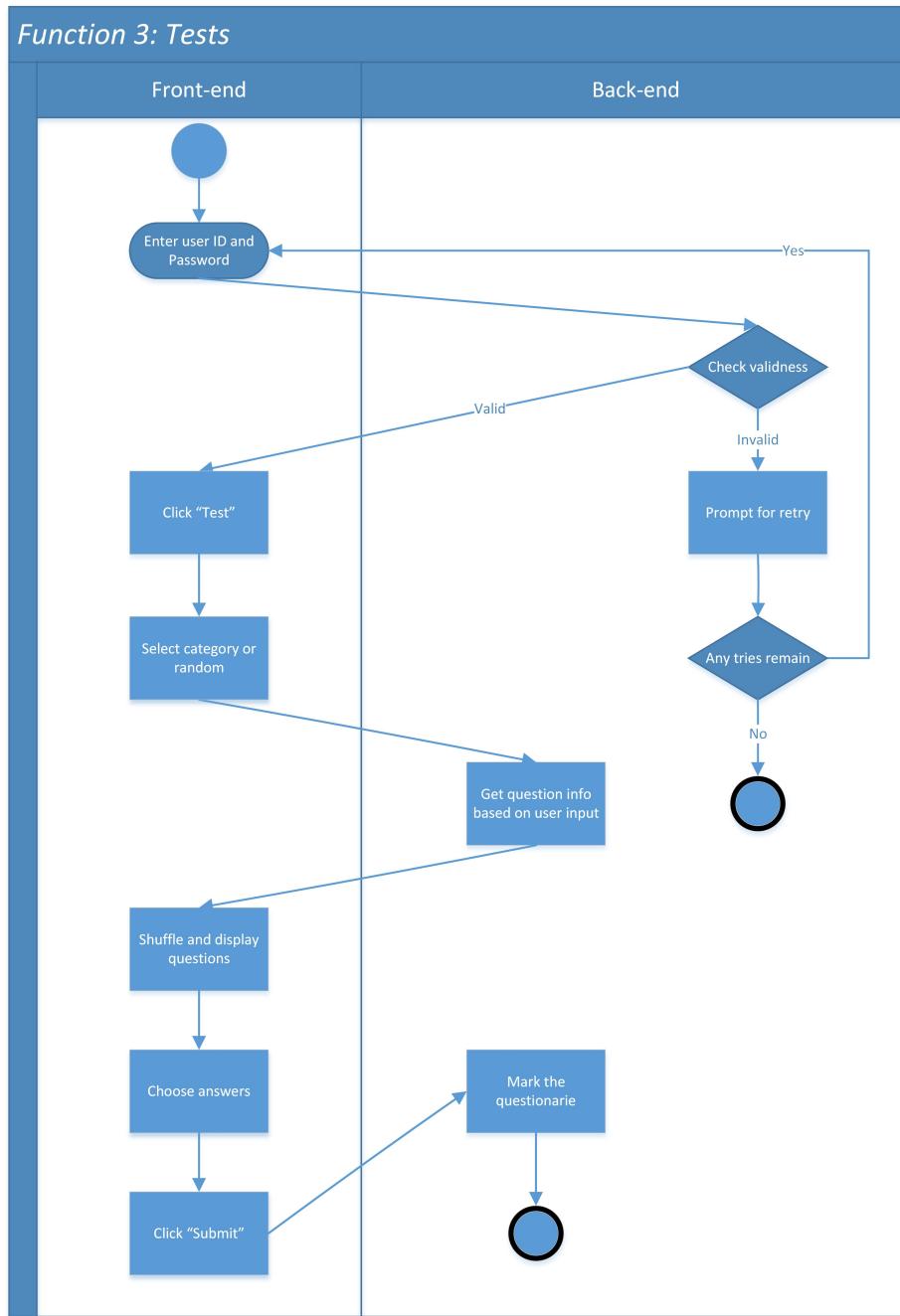


Figure 16: 前后端交互：书籍信息测试

2.3.4 测试结果分析以及可视化展示

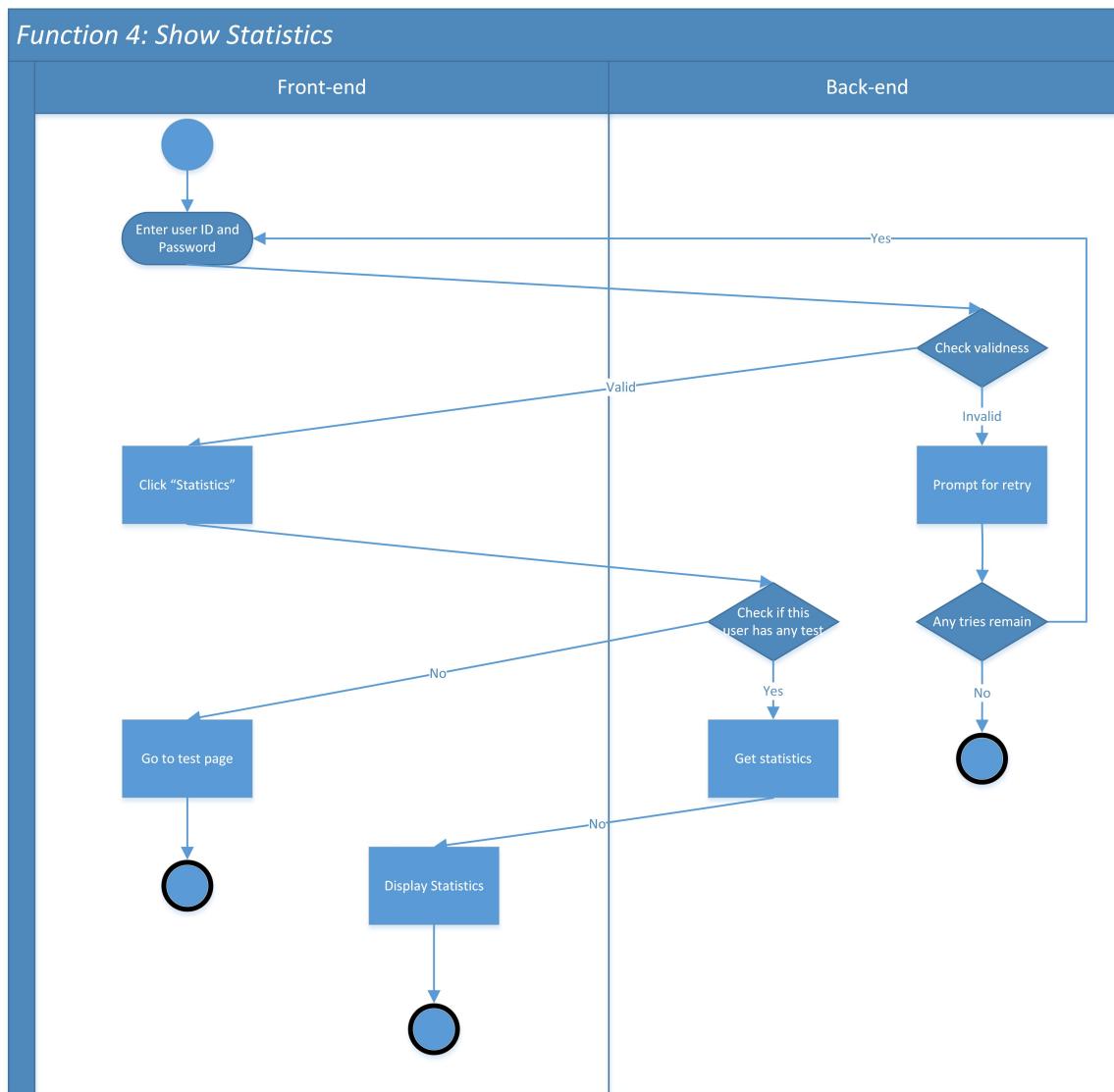


Figure 17: 测试结果分析以及可视化展示

3. 结束语

3.1 总结

1. 在本次 python 课设项目中，是我们对于多人协作 Coding 的一次尝试，我们借助这次机会，去适应社区代码的提交，去了解和使用 git 这个强大的工具，他很好的让我们三人协同工作起来，并且在很多次爆出 bug 时拯救了我们。在我们的前期调研中，我们想把该项目做到较为满意的地步，而且对于程序员来说，合作也是一个在大学中需要主动去培养的技能，无论是在前期讨论分工，中期 coding，后期报告、ppt 制作答辩的哪个阶段，我们三人都学到了相关的知识和技能。
2. 在本次 python 课设项目中，我们也是首次在类似课程设计的较大型项目中使用 LATEX 来进行报告撰写，也算是为了毕设和之后写论文做准备。在过程中也遇到了许多问题，比如：
 - svg 矢量图导入 (Figure 13)
 - 将 Author, Abstract, Reference 加入到目录
 - Reference 使用 BibTex
3. 在本次 python 课设项目中，我们也是首次去了解了 Linux 服务器。在中期的一次讨论中，我们遇到了前后端数据库不同步导致的问题，先是去尝试搭建局域网和云 Sql 服务器，但经过尝试后，遇到了很多问题，最后经过讨论觉得，如果可以全部部署到服务器一定会很炫酷。但过程是十分曲折的，虽然我们有一点点 Linux 的基础知识，但从零开始部署遇到了很多问题，而且在 Linux 下没有 GUI 界面，对于环境变量，软链接等的操作都更麻烦，我们也是 debug 了很久，才取得目前的成果。过程中也用到了计算机网络 (端口控制，IP，域名)、操作系统 (进程控制) 等相关学科的知识。

3.2 不足与展望

1. 在前期准备中爬取了豆瓣 Top200 书籍的信息，但并未爬取完全，结果后期发现，豆瓣的 Top200 是根据评分数据一直在动态变化的。虽然后期通过豆瓣 ID 爬取出了全部的信息，但排名信息无法实时更新，后期会在此基础上继续改进。
2. 在中期讨论中，想到把应用部署到服务器，便开始了对 Linux 服务器的探索，经过一周的尝试，最终有了我们自己的公网 ip **47.116.46.195**，但由于备案时间需要半个月，我们的域名 <http://www.castamerego.com/> 未能在答辩时正式启用。
3. 未能实现多用户登录。由于使用全局变量 globals 在函数间进行传参，多线程共享全局变量导致同时只能一个用户登录。期间尝试过多进程，将全局变量改写为类对象等。但未成功，后期会继续改进。

4. 在过程中还存在很多问题，上面描述的只是部分，我们所遇到的全部问题可以点击[此处](#)，在我们的 Github 仓库中查看。

References

- [1] Inc Wikimedia Foundation. Object-oriented programming. URL https://en.wikipedia.org/wiki/Object-oriented_programming. (2022, Nov 18).