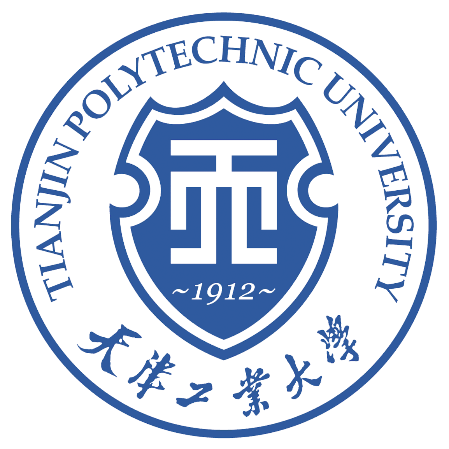
天津工业大学计算机科学与技术学院

实训总结报告



　项目名称: 基于豆瓣资源的个性化信息整合库平台

　合作培养企业：天津市融创软通科技股份有限公司

　专业方向：大数据方向

　项目负责人：何晶

　专业班级: 软件工程Z1801

姓名：孔维一

　实训起止日期：2020.6.15-2020.7.15

**1 引言** （一级标题，小三黑体，居中）

1.1项目概述（二级标题，四号黑体，左对齐）

说明项目目的、背景、意义及参考资料（包括企业提供的素材、DEMO）等；

1.2 需求分析

为什么做这个题目，有什么实际价值

1.3 运行环境

软、硬件环境

（正文，小四号宋体，1.25倍行距）

**2 项目设计**（小三黑体，居中）（以个人主要负责内容为主）

2.1 设计思路

2.2 模块功能介绍

2.3 模块结构图

2.4 程序流程图

2.5 功能设计分工

需详细标注每部分的内容，具体任务及任务安排等。

**3 详细设计**（小三黑体，居中）

详细写明项目的设计（系统设计、数据库设计）等，并写明对所涉技术、核心功能等描述；

**4 实现方案**（小三黑体，居中）

详细写明项目功能实现描述、核心代码及实现结果等；

并对调试好的程序，从执行程序弹出界面开始，每一步操作截一个图，并附加说明（图片格式为JPG格式）。

**5 系统测试**（小三黑体，居中）

详细写明测试方法、测试描述（bug与修正方法等）、预期结果等；

# 1引言

## 1.1项目概述

随着科技时代的发展，我们的社会也在飞速的发展。身处在一个信息化的社会，我们的衣食住行都离不开各种各样的信息。例如我们习惯网购各式各样的衣服，在网上搜索各种各样的美食，然后网站收集你习惯搜索的信息，再为你进行个性化推荐。毫不夸张的说，我们的生活已经离不开各种各样的信息了，我们每一个人每时每刻都在产生大量的数据。而就是为了处理如此海量的数据，大数据这门研究就应运而生了。

我们这个项目叫做基于豆瓣资源的个性化信息整合库平台，其目的就是做一个可以收集大量数据将其插入数据库，并在后台提取出来对数据进行分析整合的平台。这么做的意义在于我们可以收集大量我们想要的数据，然后针对我们想要的方面进行统计分析。例如拿我们所爬取的电影数据来说，我们可以统计哪一年评分最高的前50名的电影，亦或是可以统计五星好评的人是男性多还是女性多，甚至可以精确到某一条评论来估计是正面情绪还是负面情绪。我们将结果以图表的形式展现，这样可以更直观的得到我们想要的结果。通过一系列功能的实现，用户可以更好地检索自己想要的东西，我们也可以通过海量数据更好地了解用户，更好地为用户服务。

参考资料：通过老师对各个模块的讲解，以及老师写的简单例子，加上网上资料的自学，逐步搭建起整个框架。

## 1.2需求分析

我们身处于信息化的时代，不仅我们想要的东西可以快速的检索出来，我们自己无时无刻也在产生大量数据，这就是大数据给我们生活所带来的实际影响。但是杂乱无章的数据是没有任何意义可言的，我们也无法快速找到我们想要的数据，更别提统计了。这就需要我们利用技术对数据加以整合分析，留下我们想要的数据，剔除我们不要的数据，然后将其规范化，最后可以利用得到的数据进行我们想要的分析，将结果进行可视化，这是我们做此项目的初衷。做出这个项目我们可以从任何一个网站使用爬虫爬出我们想要的数据，将其用一张规范的二维表整合进数据库，然后使用BS项目的开发结合前端的开发，将得到的数据提取出来，用前端技术做成一个平台，我们可以在这个平台上检索我们想要的数据，亦或是可以选择一些项目进行分析，用eCharts技术进行可视化，使展示更加直观、美观。这就是此项目的意义。

同时我们在此平台加入用户管理功能，使用此平台前需要登录，如果没有账户则需要注册。此外用户还分为普通用户和管理员用户，二者的权限有所差别，同时管理员用户可以管理普通用户。并且用户拥有标记自己喜欢的或者已看过的图书、电影的功能。

## 1.3运行环境

软件环境：Gitee负责版本管理，提交项目；Pycharm负责搭建scrapy框架，利用爬虫爬取数据；IDEA负责服务器端以及前端代码编写——使用技术为BS开发框架SpringBoot，前端开发技术（HYML + JavaScript + LayUI + ECharts）；连接远程数据库MySQL(Mariadb)；平台开发模式：基于JSON数据的前后端分离式开发。最后使用浏览器打开最终成型的平台。

硬件环境：运行内存8G的PC即可。

# 2项目设计

## 2. 1设计思路

一个软件开发的目的就是面对用户，去完成用户的要求，需要建立合适的思路，开发出能完成用户所需的项目。所以第一点是要明白用户需要什么样的项目，这个项目不仅要满足用户所需要的功能，也要将软件做的人性化，使得界面美观让用户愿意去使用，并将软件使用方法通俗易懂化。

1）第一步我们需要爬取数据，在这一阶段使用的是python相关的scrapy框架，使用的编程软件是Pycharm，我们编写一个能够从豆瓣网站爬取我们所需信息地网络爬虫程序，给其需要爬取的各项数据以及网页，让其自己运行，因为大部分网站设有反爬虫机制，所以要设置限制爬取请求数、延迟、模拟请求浏览器等，因为运行时间较长，考虑增加错误提醒设置以及中断重爬设置。将爬取的数据插入远程数据库，提前在数据库中建好相应的表，运行爬虫即可。

2）接下来进行的是平台web开发，在这一阶段，我们将使用SpringBoot框架搭建一个前后端分离开发而成的数据整合展示平台，

我们的后端服务器就是要实现从数据库端获取数据并进行相应服务的业务逻辑的完成，同时也将满足一些向数据库中进行插入，更新的操作。例如注册用户，添加个人书评，

我们的前端开发主要是html、css、js等搭载layui框架所编写而成，此部分为各个页面的展示，注意各个页面之间的关系、跳转联系。

将二者结合起来便可以完成使用SQL语句从数据库里查出想要的内容的功能，简单的检索利用页面显示，复杂的使用eCharts进行可视化展示。

3）在完成这些基本功能的前提下，对操作功能、页面等进行优化设计，尽量使界面更加美观，使使用更加方便。

由于此次我们为4人一组，一人负责一个模块的搭建，共同实现用户模块的创建，在设计过程中由于使用Gitee以及搭建共通项目的原因，要注意差异的处理，将小框架一步步拼接成大框架

## 2.2模块功能介绍

本次项目分别在两个平台上分为两个阶段进行。Pycharm中数据获取的爬虫程序开发阶段以及在IDEA中的BS项目开发阶段。

第一阶段的开发工作为：在Pycharm开发工具中，使用scrapy框架编写进行图书信息爬取的爬虫程序：此阶段是用于爬取豆瓣网站中所需数据的具体内容，例如每个图书的书名，作者，再深入一层为图书的简介，书评等内容。并将这些数据存储在远端数据库中相对应的关系型数据库表中。此外，为了防止反爬虫机制的屏蔽，应该设置延迟参数，同时处理请求的数量等，并加入错误处理机制，以防爬虫突然的中断。我们也可以将爬取错误的数据写入json文件。并进行二次处理，同时也可以进行数据的一些清洗工作，将一些无用的数据（某些字段为空值的数据）去除掉，以此来保证我们获取数据的有效性。

我本人在这一阶段主要负责图书相关数据的爬取，主要实现的功能有以下几个方面。

1）图书基本信息的爬取。其内容包括以下字段：豆瓣ID，书名，作者，出版社，出版日期，价格，页数，ISBD，分类，简介，图书封面连接，图书sha1哈希值（对应本地图片资源地址），评分，评分人数，五星比率，四星比率，三星比率，二星比率，一星比率。总条数为8255。

2）图书评论信息的爬取。其内容包括以下字段：被评论图书ID，评论人唯一用户名，评论内容，评论时间，评论星级。总条目数为33761.

3）图书评论豆瓣用信息的爬取。其内容包括以下字段：豆瓣用户唯一用户名，常住地，加入豆瓣日期。总条目数为10855条。

4）数据库数据插入异常处理，将错误数据的详情写入json文件，并记录错误位置

第二阶段的开发工作为：在IDEA中进行的的BS项目开发(使用Spring Boot框架）：此阶段除了要用到爬虫爬来的存到数据库里的数据以外，应该是与第一个阶段分开来的。在此阶段我们的项目被分为五大模块，即用户模块，书籍模块，音乐模块，电影模块，电视剧模块。

我本人在这一阶段主要负责用户模块和书籍模块的开发。

用户模块介绍：用户模块主要分为用户注册，用户登录，用户管理，退出登录以及用户个性化显示四个小的模块。

1）用户注册：用户注册需要登记的信息（必填项）有：用户名、昵称、密码、需要再次确认密码以及验证码。注册需要满足的条件为用户名尚未被注册，昵称不为空，填写密码不为空，且两次输入密码一致以及验证码输入正确。验证码为随机生成的忽略字母大小写的字母数字组合。

2）用户登录：已经注册用户输入用户名，密码，以及验证码厚验证无误即可登录，用户登录时需要满足用户在数据库中存在，用户为激活状态，用户为非屏蔽状态，密码无误，且验证码输入正确等必要条件。验证码为随机生成的忽略字母大小写的字母数字组合。

3）用户管理：用户管理分为普通用户的管理以及管理员用户对普通用户进行管理的管理员平台。普通用户可以在个人用户管理页面修改自己的昵称，密码等信息。而管理员用户

4）退出登录：用户可以在主页面的右上角选择安全退出，退出后用户登录状态相关的session会话将被清除。

5）用户个性化显示：主页面中将会根据登录用户的不同，从后端服务器获取当前登录用户的相关信息并进行显示。例如用户头像为从数据库中获取到的相应用户头像资源信息，然后交给前端js来进行动态加载。同时数据分析页面也会动态请求当前用户喜欢和已看数目，并进行加载。

书籍模块介绍：书籍模块分为图书检索，数据可操作和数据可视化三个小模块

1）图书检索：图书检索有精确检索,如按图书ISBN检索，按图书出版日期检索，还有模糊检索，分为按分类检索，按图书作者检索，按书名检索等。检索的结果当中将显示图书的基本信息，并提供相关操作的入口。

2）数据可操作：对图书的操作有查看详情以及编辑两种。

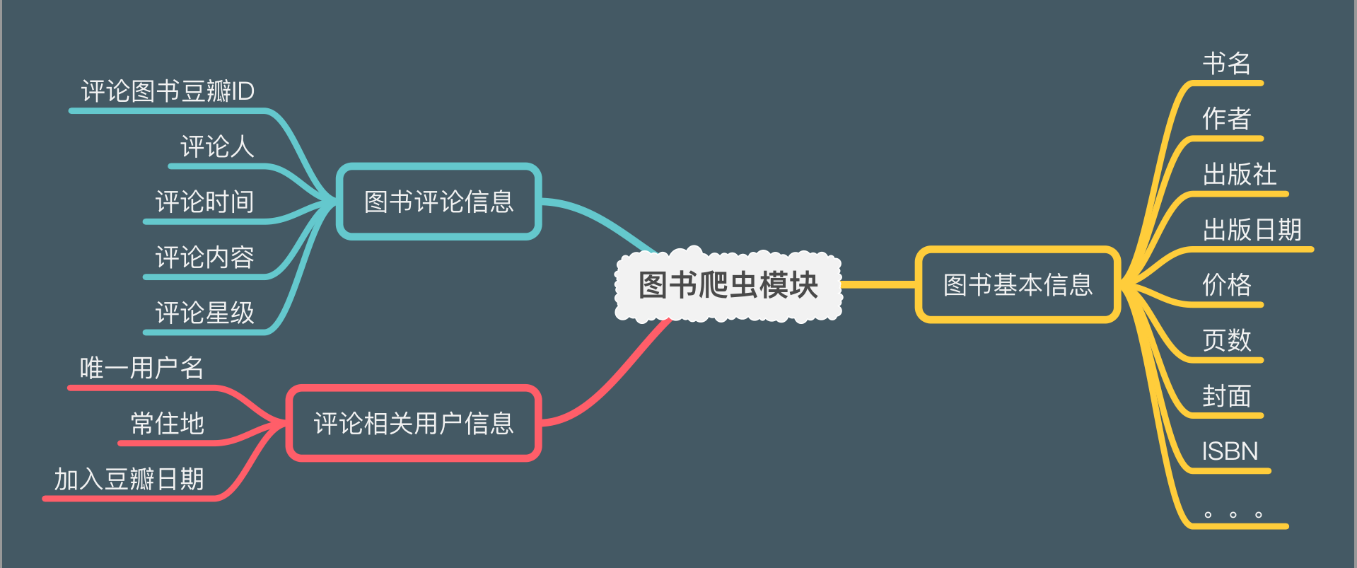
查看详情当中将展示图书的封面，图书的基本信息（书名，作者，价格等）；图书豆瓣评分信息（豆瓣评分，评分人数，各评分星级比例）；书评简介信息，豆瓣网中书籍的简介以及随机爬取的一条相应书籍的评论。我的评分评论：登录用户所进行的用户个人评分以及个人评论。需要登录用户（并且进行过相应编辑才会显示）。

编辑操作包括将书籍加入我喜欢，加入我已读，进行用户对相应书籍的个人评分操作和个人书评添加操作。操作需要在登录状态下进行，非用户的游客登录将不能进行编辑操作。

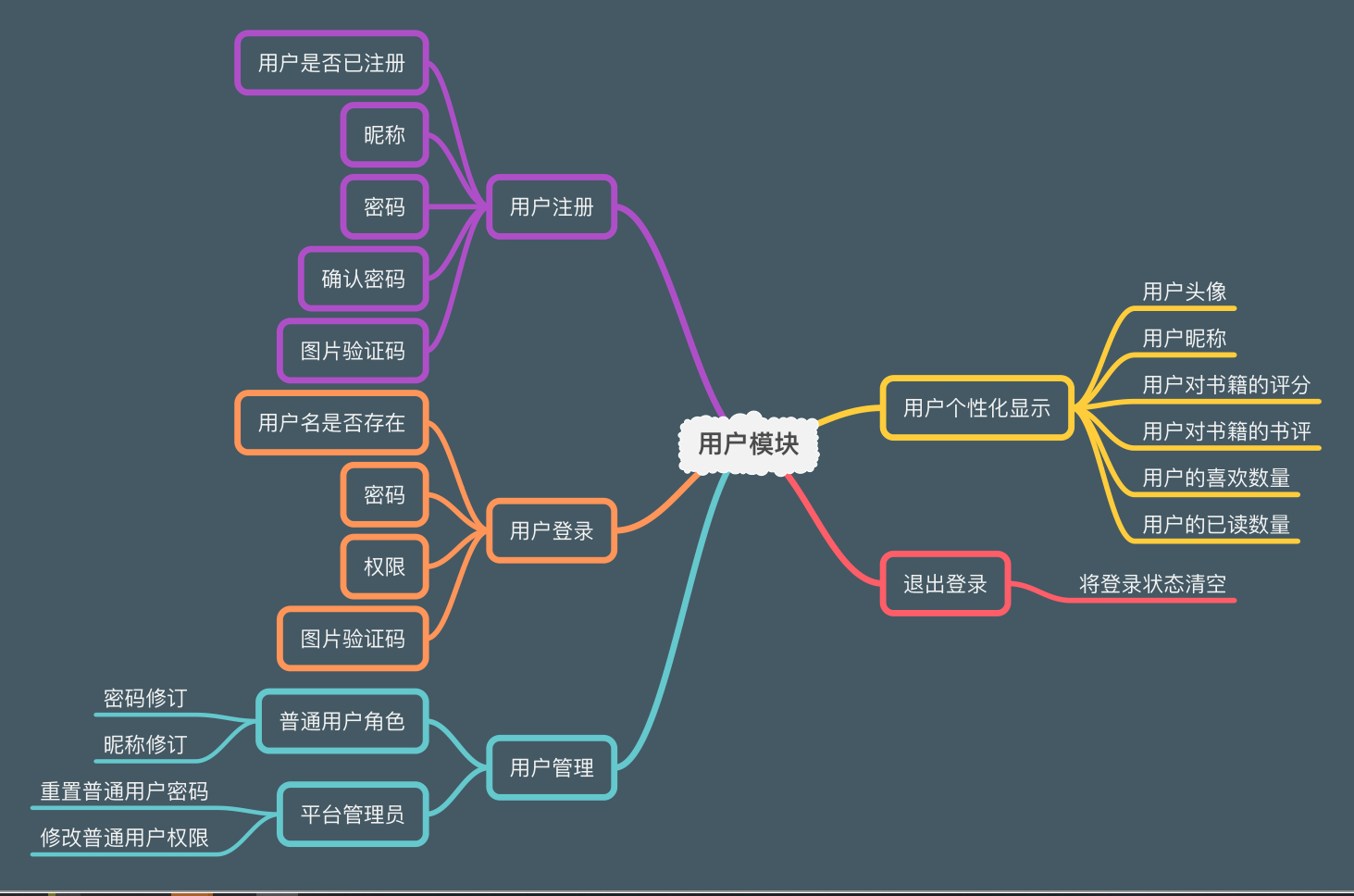
3）数据可视化：此部分包括有数据量显示，如图书数量，作者数量我喜欢，我已读数量等。折线统计图，如图书评分统计，图书数量随年份变化统计；柱状统计图，图书价格统计；扇形统计图，各详情信息的热门比例。

2.3 模块结构图

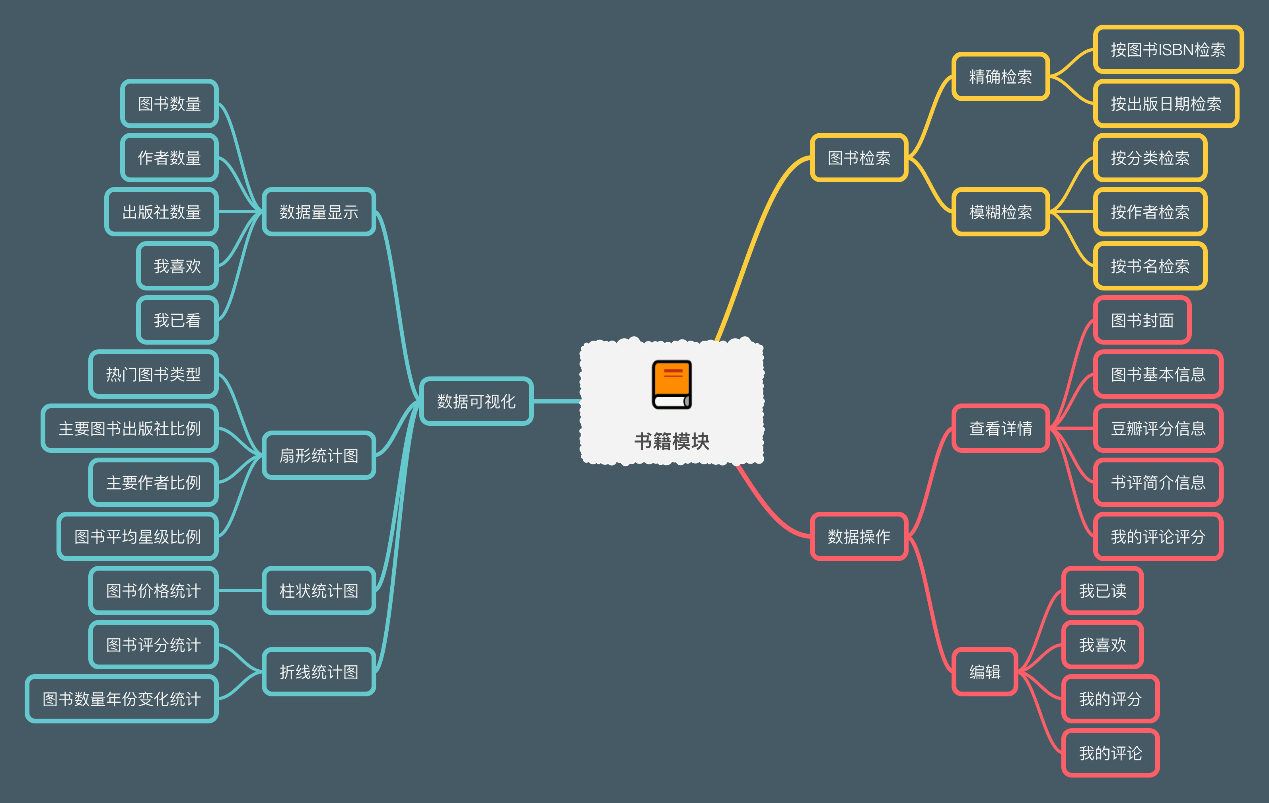
1）数据获取阶段，爬虫程序模块结构图



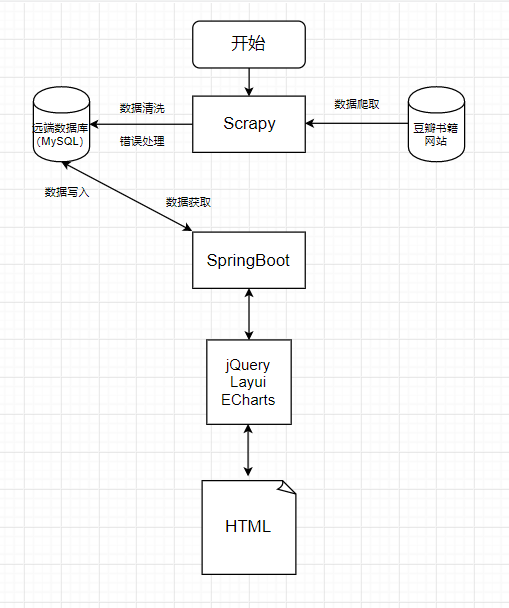
2）BS项目开发阶段：用户模块结构图



3）BS开发阶段：书籍模块结构图



2.4 程序流程图



2.5 功能设计分工

1）scrapy框架当中编写爬虫程序，完成图书数据的获取。

2）SpringBoot项目中，完成了用户登录和注册页面的前端开发和和美化，以及具体注册登录JavaScript逻辑和后端业务逻辑的实现

3）图书模块中完成图书的检索展示，图书的详情展示，图书的编辑（我喜欢，我已看，我的评分，我的评论等功能），以及图书数据分析的可视化展示。

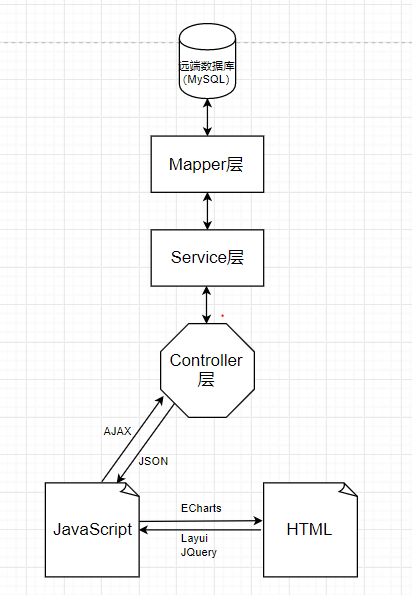
4）用户模块中完成了用户个人管理模块的前后端代码编写，以及根据用户状态开放管理员平台的功能，还有用户密码重置功能的实现，用户状态管理功能。

5）整个主页面的搭建，（使用tab标签展示每个页面）

# 3详细设计

3.1 系统设计

系统详细设计图示如下:



平台系统可以简单地划分前段UI部分和后端Java部分。前后端之间的通信使用JSON数据形式和AJAX请求的技术来实现。系统前端部分通过AJAX技术向后端发送数据请求，系统后端部分则通过JSON形式向前发送数据请求结果。

1）系统前端UI部分使用技术为HTML + JavaScript + Layui + jQuery + ECharts。HTML作为运行在浏览器上的基本静态页面，主要功能为实现相关页面的静态加载，JavaScript技术主要实现对前端页面的动态加载。Layui则为前端开发的美化提供了相当大的便利，利用Layui框架，我们就可以轻轻松松编写出整洁，漂亮的前端UI界面。jQuery技术则用于简化我们的js开发难度。ECharts帮助我们轻松实现数据的可视化展示。

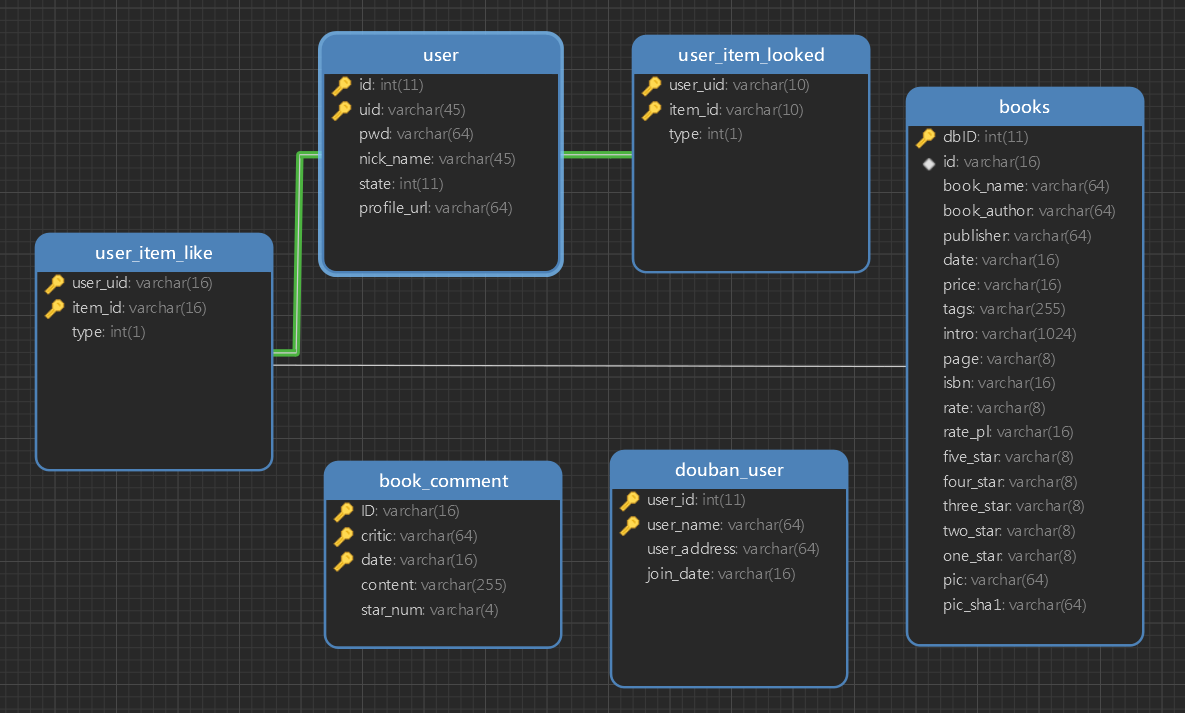
2）系统后端部分使用的是SpringBoot框架技术搭配maven进行相应开发，SpringBoot项目创建时加载Spring Boot DevTools, Lombok, Spring Web, MyBatis Framework, MySQL Driver等组件。

相关组件中，Lombok用于将一个java类转化为可使用的javabean；MyBatis则是一个基于Java的，内部封装了jdbc的持久层框架，他的功能是自动完成数据库加载驱动，创建连接，等操作，可以让我们只需关注sql语句本身即可，不需要花费太多精力在jdbc的具体实现上。

系统后端部分根据业务逻辑可以划分为三层，分别为Controller层，Serivice层以及Dao层。Controller层是管理业务（Serivice）调度和管理跳转的，Controller层只负责管理，并不负责实施，Serivice层负责实现具体的业务逻辑，Dao层负责与数据库打交道，进行数据库的增删改查操作。其余还有entity层（实体类，充当一个对象的模型），以及tools层（负责完成相应需求而设立的工具类）。

3.2数据库设计

数据库设计过程中，我使用了数据库管理工具Navicat，利用可视化环境共创建了六张表，详情如下图所示:



1）books表，用于存储爬取到的书籍详细信息

2）book\_comment表，存储书籍评论信息（我的书评也会插入此表）

3）douban\_user表，存储豆瓣评论用户信息

4）user表，平台用户数据

5）user\_item\_like表，存储用户喜欢的书籍

6）user\_item\_looked表，存储用户已读书籍

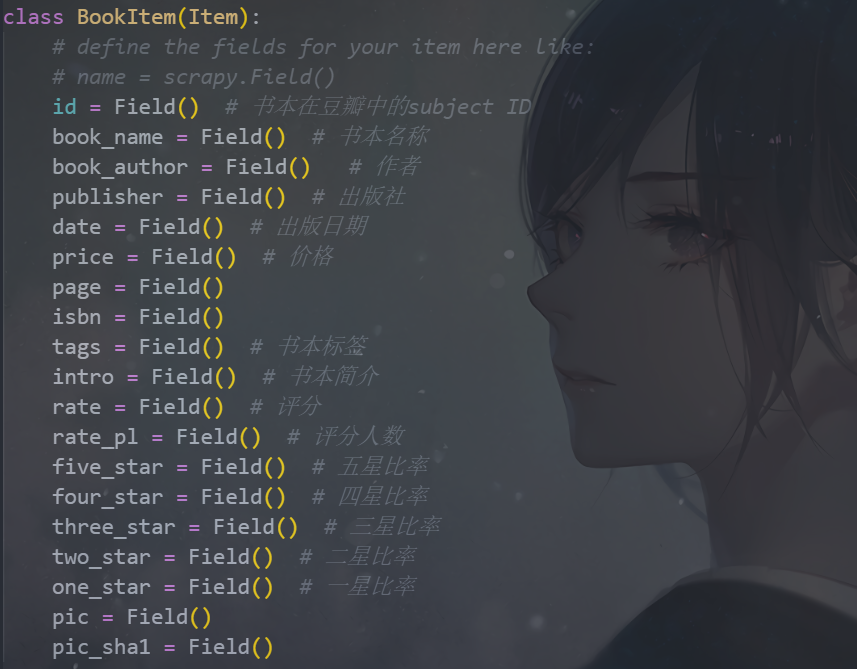
同时为了加快书籍数据可视化部分的查询效率，创建有相应的视图来保存对应查询结果。

# 4 实现方案

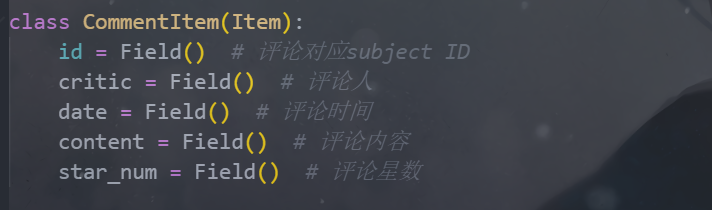
第一阶段，爬虫程序实现

1）定义爬取信息类

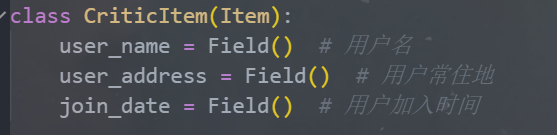
图书详情类：



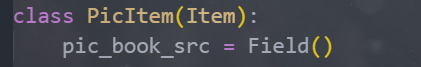
书评类



评论用户类：



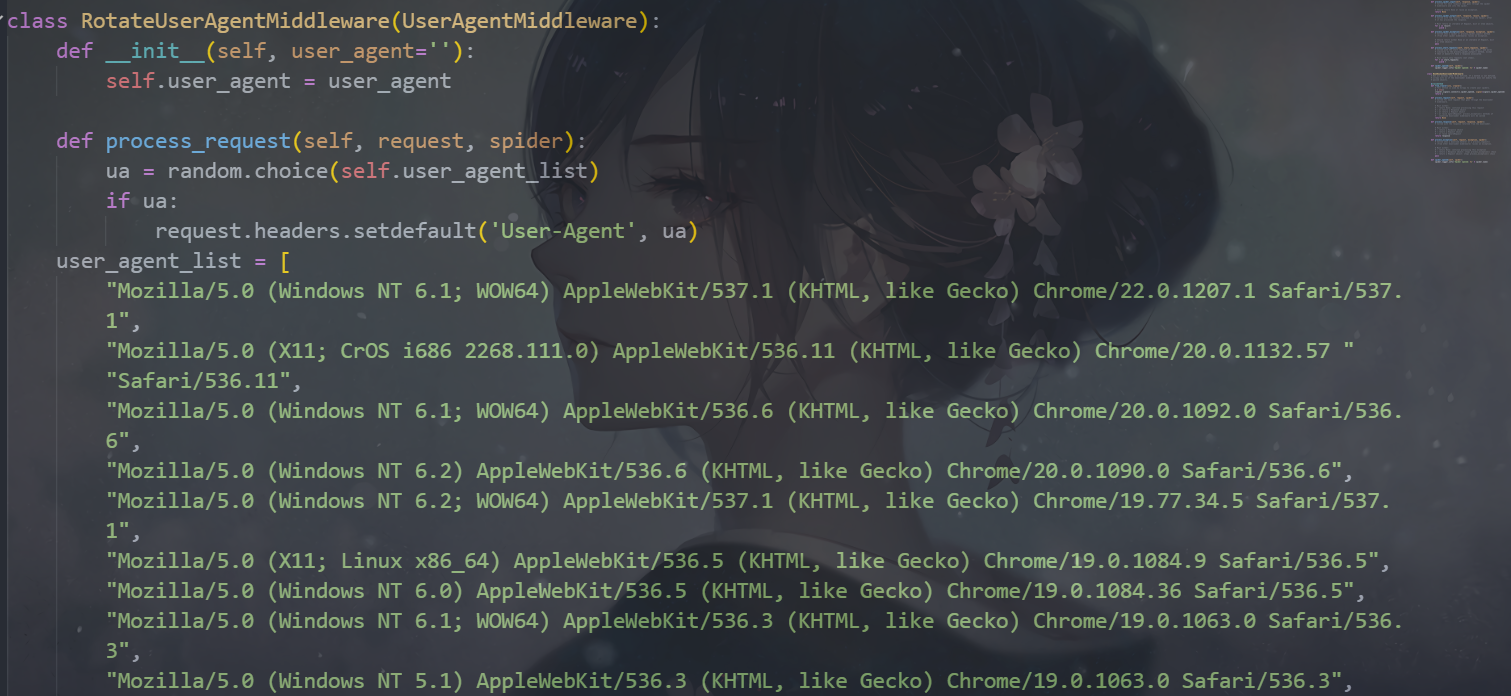
封面路径类：



2）相应setings设置，如设置是否遵守robot.txt，同时处理请求数，延时，爬取深度限制等



3）模拟请求浏览器，在middlewres中实现，利用random.choice实现一个随机的浏览器请求。



4）爬虫spider具体编程实现

爬虫参数设置 name参数为爬虫名称，用于启动爬虫，allowed\_domain参数用于限制爬虫爬取域名在豆瓣网以内，start\_urls则规定爬虫开始请求页面为豆瓣读书的分类页面。



5）利用xpath解析获取所需数据(以下为几个数据实例的获取)



6）在pipeline管道中处理数据（存入远端数据库）

MySQL数据库连接：实际连接的是一个mariadb数据库，但由于两者几乎完全兼容，所以可以使用mysql数据库连接方式进行连接。



具体item处理（以图书详情为例）:在出错情况下将数据写入json.



至此我们的爬虫实现方案已经完成。进入第二个阶段，Springboot项目的具体实现方案。

1）配置完成Springboot项目的搭建，如maven依赖和application.yml的配置，使我们的SpringBoot项目能够正常运行。

