

**项 目 报 告**

题 目 图书推荐算法的设计与实现

姓 名 李珏 许佳晨 陈怡

学 号 31501078 31501081 31501106

专业班级 计算1501 计算1502

指导老师 徐煌老师

图书推荐算法的设计与实现

【摘要】 随着互联网的迅速发展，互联网数据正在急速的扩张。在如此大量数据的情况下，用户寻找到对自己有用的数据或者感兴趣的内容的难度就会大大增加。在这个信息过载的时代中，个性化推荐系统是必不可少的。一个好的个性化推荐系统，可以根据用户的喜好和特征来提供比较精准的推荐服务。

本文设计和开发了一个图书推荐系统，实现整个系统从登录注册到个性化推荐。，总体采用了SSM（Spring+Spring MVC +Mybatis）框架、使用maven管理项目依赖、使用基于书籍的推荐算法、基于用户的推荐算法、基于Slope one的算法以及回归算法开发。

【关键词】基于内容，信息过载，个性化推荐系统，SSM框架，mahout算法

目录

[第1章 绪论 1](#_Toc514245334)

[1.1图书推荐系统的简述 1](#_Toc514245335)

[1.1.1图书推荐系统的背景 1](#_Toc514245336)

[1.1.2图书推荐系统的现状和发展趋势 1](#_Toc514245337)

[1.1.3 国内外研究现状 2](#_Toc514245338)

[1.2 图书推荐系统的目标和内容 3](#_Toc514245339)

[1.2.1图书推荐系统的目标 3](#_Toc514245340)

[1.2.2图书推荐系统的内容 4](#_Toc514245341)

[第2章 系统的开发工具和框架技术的介绍 5](#_Toc514245342)

[2.1 系统的开发工具简介 5](#_Toc514245343)

[2.1.1 IDEA工作平台介绍 5](#_Toc514245344)

[2.1.2 Mysql数据库管理系统介绍 5](#_Toc514245345)

[2.1.3 Navicat Premium 12介绍 6](#_Toc514245346)

[2.1.4 Tomcat8.0介绍 6](#_Toc514245347)

[2.1.5 Maven介绍 6](#_Toc514245348)

[2.2 系统框架介绍 7](#_Toc514245349)

[2.2.1 SSM（Spring+SpringMVC+Mybatis）框架介绍 7](#_Toc514245350)

[2.2.2 Mahout介绍 7](#_Toc514245351)

[2.3 系统算法介绍 8](#_Toc514245352)

[2.3.1 基于用户的推荐算法介绍 9](#_Toc514245353)

[2.3.2 基于书籍的推荐算法介绍 11](#_Toc514245354)

[2.3.3 基于Slope one的推荐算法介绍 12](#_Toc514245355)

[2.3.4 回归算法介绍 14](#_Toc514245356)

[第3章 系统的需求分析和整体设计 15](#_Toc514245357)

[3.1 系统的需求分析和整体设计 15](#_Toc514245358)

[3.2 系统的模块设计 15](#_Toc514245359)

[3.2.1 用户登录推荐模块 16](#_Toc514245360)

[3.2.2 用户注册登录功能模块 17](#_Toc514245361)

[3.3 系统的数据库设计 17](#_Toc514245362)

[第4章 系统的总体设计与开发 19](#_Toc514245363)

[4.1 系统总体设计 19](#_Toc514245364)

[4.1.1 登录注册模块 21](#_Toc514245365)

[4.1.2 推荐模块 22](#_Toc514245366)

[4.1.3 推荐算法设计 25](#_Toc514245367)

[第5章 系统测试 29](#_Toc514245368)

[5.1 系统测试目的 29](#_Toc514245369)

[5.2 系统测试方法 29](#_Toc514245370)

[第6章 实验结果检验 30](#_Toc514245371)

[6.1 测试数据集处理 30](#_Toc514245372)

[6.2 具体检验指标说明 30](#_Toc514245373)

[6.2.1 检验指标说明 30](#_Toc514245374)

[6.2.2 具体实验结果检验及结果 31](#_Toc514245375)

[第7章 总结与展望 35](#_Toc514245376)

[7.1 系统项目工作总结 35](#_Toc514245377)

[7.2 系统项目工作展望 35](#_Toc514245378)

第1章 绪论

## 1.1**图书推荐系统的简述**

### 1.1.1图书推荐系统的背景

随着互联网技术的迅速发展，在网络上搜集或寻找自己需要的信息已经成为人们常用的渠道。在当今社会，互联网已经成为我们不可或缺的一部分。而正因为互联网数据的丰富以及数据的多样性，使得很多传统的行业也在不断地变化，促使越来越多的企业、个人将数据电子化、信息化。也正因为如此，出现了数据过载，当数据过于庞大的时候，用户找不到自己所需要的信息或者产品也变成了一种常态，因此需要一个能够减少不必要选项，并可以提供个性化的服务选择。

同时，这样的现象在图书信息中也尤为明显，互联网的发展也使得书籍不断的电子化，同时网络书籍也在井喷式的发展。在这个情况下，读者很难快速有效地选择他们感兴趣的书籍，此时推荐系统就应运而生了。基于内容的图书推荐系统，通过识别和分析读者的阅读偏好，向读者提供建议，帮助读者去了解到自己潜在喜欢的书籍，让读者在更加简洁的书单中去挑选自己喜欢的书籍。在提高效率的同时，使得读者有一个良好的阅读体验。

### 1.1.2图书推荐系统的现状和发展趋势

推荐系统的概念是在20世纪90年代中期提出的。到目前为止，伴随着信息技术的进步以及机器学习技术快速的发展，为推荐系统带来了广泛的关注，提供了良好的机会和发展平台，推荐系统由此进入快速发展的阶段。不仅许多国内外学者正在研究推荐系统，ACM也多次使用推荐系统作为讨论主题，甚至许多国内外期刊也将推荐系统作为专题报道[1]。就用户模型而言，常用的用户建模技术包括向量空间模型，用户评估矩阵和机器学习技术。国际上，已经提出了智能代理技术来分析客户需求，建立用户模型并提供相应的推荐服务。还提出使用神经网络和遗传K-means算法来尝试获得用户偏好。系统中最常用的推荐算法是协同过滤技术。针对协同过滤的改进，尝试使用分解技术来降低特征向量的维数以提高最近邻居的搜索速度，并且对用户最近邻近项赋值权重的方法来优化算法，提高推荐质量[2]。因此，各种数据挖掘技术也开始被应用于推荐系统，以改进推荐系统的精度[3]。学者们也关注推荐系统的可移植性和可信性问题。基于对等网络环境，通过在客户端对用户信息进行控制，以便用户即使在离线状态下也能进行推荐。此外，该系统的推荐实时性，算法的可伸缩性，用户的信任等等融合了信息获取，数据挖掘，人机交互，自然语言处理，心理学，社会行为研究等相关领域都是最近研究的热点。

### 1.1.3 国内外研究现状

由于互联网技术的迅猛发展，推荐系统越来越受到社会的关注。个性化推荐系统的研究内容和研究方向主要包括：推荐系统的推荐准确性和实时性是一对矛盾的研究；推荐质量研究，例如客户评价数据的极端稀疏性，使得推荐系统无法产生有效的推荐，由此推荐质量方面推荐系统就难以保证[4]；研究多种数据和多种技术的整合;数据挖掘技术在个性化推荐系统中的应用以及基于Web挖掘的推荐系统已引起研究人员的越来越多的关注。；由于推荐系统需要分析用户购买习惯和兴趣爱好，涉及到用户隐私问题，如何在提供推荐服务的同时保护用户数据也是关键点[5]。

推荐系统实际上是很多技术的集合和体现。电子商务推荐，电子商务推荐和内容推荐（在许多类别中）都有其自身特点。从近年的主要趋势来看有以下几点：

1、机器学习技术日益成熟。目前，中国的一些大公司正在使用机器学习技术来构建他们自己的推荐系统。通过多种推荐算法来进行混合得出一个推荐得分，最后再利用机器学习的方法来进行参数优化[6]。

2、基于机器学习的推荐系统与计算广告非常相似。它使用非常大规模的稀疏表示功能和大量样本来训练估计预测的用户点击率，浏览时间，赞扬率，购买率等模型。

3、模型本身的技术升级，其中最广为人知的就是LR模型。事实上，依赖于LR的特征工程本身就是一个具有非常高的上限的模型，并且值得在很长一段时间内完成。然后回到添加FFM，使用GBDT构造特征来使用一些技术来转换超稀疏的特征，直接使用GBDT以类似于embedding的方式嵌入到连续的特征中[7]。这种逐步升级将带来实际结果的显着增加。当然，有些人仍然使用DNN做特征来完成功能，甚至直接进行模型制作。但是，由于硬件问题，DNN做模型的推荐系统在经济上不适合在规模庞大，成本敏感的行业中大规模应用。

根据Google学术搜索获得的统计结果，可以更直接地看到。从1992年到现在，与推荐系统相关的已发表论文数量和研究成果显着增加。不仅有国内外许多科学家研究推荐系统，ACM也把推荐系统作为反复讨论的主题，还有很多国内外期刊也多次将推荐系统作为专题报道[8]。

基于内容的推荐系统结合人工智能、概率统计和机器学习等技术进行信息过滤，然后将其表示成一个向量空间中的用户模型来表示用户的兴趣，同时通过注释资源对象的内容特征的方式生成一个对象特征向量作为资源模型[9]。目前，基于内容的推荐系统主要通过分析，获取和过滤对象的文本信息来完成推荐。TF-IDF方法是表征信息获取中文本特征的最常用方法。基于内容过滤的系统具有简单和高效的优点，但由于对项目内容特征的过度细化导致提取能力有限[10]。

## 1.2 **图书推荐系统的目标和内容**

### 1.2.1图书推荐系统的目标

本项目中的图书推荐系统，本课题的设计与开发是基于协同过滤算法以及Tensorflow的深度学习来分析用户对书籍的评分以及用户的个性化特征来实现对用户的个性化推荐。该系统可以实现对已有的数据集进行多个基于不同对象的推荐算法进行计算得出多个推荐分。利用得出的多个推荐分以及用户的个性化特征进行回归模型的训练，从而得出一个最后的较为准确的推荐结果，按照相关度的排名推荐给用户。

### 1.2.2图书推荐系统的内容

本系统使用MVC设计模式进行开发，使用目前比较常见的Maven来进行管理项目依赖，同时采用了Spring+Spring MVC+MyBati联合框架。本系统主要实现了通过分析数据集中的用户对图书的评分，先利用基于书籍的推荐算法进行筛选得出一个数据集得出第一个推荐分，其次利用基于用户的推荐算法进行计算得出第二个推荐分，然后利用Slope one推荐算法进行计算得出第三个推荐得分。最后利用用户的年龄数据以及三个推荐得分进行机器学习，得出最终的推荐图书列表。本系统中通过计算三个推荐分以及用户的年龄数据作为特征，利用逻辑回归模型进行计算得出最后的推荐结果，从而得到一个比较准确的推荐。

# 第2章 系统的开发工具和框架技术的介绍

## 2.1系统的开发工具简介

本系统主要采用以下几种开发软件进行开发与设计：IDEA（IntelliJ IDEA）JAVA语言开发集成环境以及Spyder Python开发集成环境，在系统数据存方面采用Mysql数据库和Navicat Premium 12图形化数据库工具。使用Tomcat8.0来运行整个项目，同时通过Maven来管理项目依赖。

### 2.1.1IDEA工作平台介绍

IDEA全称IntelliJ IDEA，是一个用于java语言开发的集成环境，IntelliJ作为主流的JAVA开发软件，被公认为业界最好的Java开发工具之一。该软件强大之处在于智能代码助手，自动代码提示，重构，J2EE支持，各种版本的工具（git，svn，github等），JUnit，CVS集成，代码分析和创新的GUI设计。大大减少了开发人员编写重复或类似的代码，能够自动搜索出相同或相似名称的变量或组件提供给开发人员使用，这样大大加快了程序员编写代码的速度，从而加快了项目的开发过程。使用IDEAJ作为项目的主要开发工具在数据库和J2EE方面的开发和发布调试有很大的作用，大大提高了开发的工作效率。

### 2.1.2 Mysql数据库管理系统介绍

MySql数据库是目前最流行的关系数据库管理系统之一。它是最受欢迎的关系数据管理系统之一。作为关系数据库管理系统，它将数据保存在不同的表中，而不是将所有数据存储在同一个大型仓库中。这种设计极大地提高了操作的速度和灵活性，也因为它体积小，速度快，总拥有成本低，成为了一般中小型网站的开发首选。Mysql软件使用的是双授权政策，区分为社区版和商业版，这也成为许多高校作为数据库学习的首选软件之一。而且Mysql开发源码这一特点，使得任何人都能在GPL（Genral Public License）的许可下下载和修改其内容来满足自身的个性化需求，这也是Mysql成为当下主流数据库的原因之一。

### 2.1.3 Navicat Premium 12介绍

Navicat Premium是一款简化数据库管理的工具，可以有效简化数据库管理，并有效降低数据库管理成本。它允许用户在同一个程序中连接到MySql，SQLite，Oracle和PostgreSQL等数据库使用户更容易管理不同类型的数据库。更直观的图形界面允许用户以安全和简单的方式创建和访问。

### 2.1.4Tomcat8.0介绍

Tomcat是Apache的Jakarta项目中的一个核心项目。作为一个免费的开源Web服务器，Tomcat是一个轻量级的应用服务器，它通常用于中小型系统和并发访问用户。尽管Tomcat是Apche服务器的扩展，但它在运行上也是独立的，因此当您运行Tomcat时，它实际上是作为与Apache分开的一个进程运行的。

### 2.1.5Maven介绍

Maven项目对象模型（POM），一种软件项目管理工具，可以通过一小段描述性信息来管理项目构建，报告和文档。

除了Maven构建程序的能力之外，它还提供了高级项目管理工具。由于Maven的默认构建规则具有高度可重用性，因此通常可以使用两到三行Maven构建脚本来构建简单的项目。由于Maven的面向项目的方法，许多Apache Jakarta项目在发布时都使用Maven，而使用Maven的公司项目的比例仍在不断增长。使用Maven管理可以提高在项目开发时的效率

## 2.2 系统框架介绍

### 2.2.1（Spring+SpringMVC+jdbc）框架介绍

本系统使用当前比较主流的Java EE企业级框架开发，由Spring、SpringMVC、JDBC三个开源框架整合而成。在Sping框架中，项目的每个组件的依赖关系由运行时的容器决定，当项目运行时，容器动态地将待定的依赖关系注入指定的组件.SpringMVC分离了控制器，模型对象，调度程序和处理程序对象的角色。这种分离使得它们更容易定制。JDBC（Java DataBase Connectivity,java数据库连接）是一种用于执行SQL语句的Java API，可以为多种关系数据库提供统一访问，它由一组用[Java语言](https://baike.baidu.com/item/Java%E8%AF%AD%E8%A8%80)编写的类和接口组成。JDBC提供了一种基准，据此可以构建更高级的工具和接口，使数据库开发人员能够编写数据库应用程序

### 2.2.2 Mahout介绍

Mahout是Apache的一个开源项目，它提供了可扩展机器学习领域经典算法的一些实现，可帮助开发人员更轻松，更快地创建智能应用程序。

Mahout使用Taste来改进协作过滤算法的实现。它是基于Java的可扩展，高效的推荐引擎。Taste不仅实现了最基本的基于用户和基于内容的推荐算法，还提供了扩展接口，以便用户可以轻松定义和实现自己的推荐算法。

## 2.3 系统算法介绍

本系统主要使用三个推荐算法来计算图书对用户的推荐相关度。通过不同的推荐算法来分别得出一位用户对某本没有进行过评分的图书的预测评分，从而将图书按照预测评分的高低来依次推荐给用户。在实际应用中，很少有直接使用某种算法进行推荐的系统。单推荐算法有一些优点，但也有一些缺陷，有一定的局限性。目前比较广泛使用的是，通过不同算法的结果来综合结果，从而得出一个较为准确的结果。所以使用多个推荐算法可以更好地去给用户一个个性化的推荐体验，从而达到更加贴合自己业务的目的。

### 2.3.1 基于用户的推荐算法介绍

基于用户的推荐算法的主主体思想可以体现为“物以类聚、人以群分”这几个字。简单的说就是如果一个用户看了《哈利波特》、《小时代》等几本书，另外也有一个用户也喜欢阅读这些电影，他同时还喜欢《西游记》，那么很有可能用户1也喜欢看《西游记》这本图书。由此我们不难发现，当一个用户1需要个性化推荐的时候，可以先找到与他兴趣相似的用户群体，然后再把用户群体喜欢的、并且用户1没有看过的书籍推荐给用户1[11]。

基于上述基本原理和思想，我们可以将基于用户的推荐算法分为以下几个步骤：

首先，分析每个用户对书籍的评价（本系统中通过用户对阅读过的书籍的评分）；

然后根据用户对本书的评价计算所有用户之间的相似度；

其次，选择与当前用户行为最相似的N个用户；

最后，通过这N个用户评分最高其并且当前用户又没有浏览过的书籍推荐给当前用户。如下图原理所示。

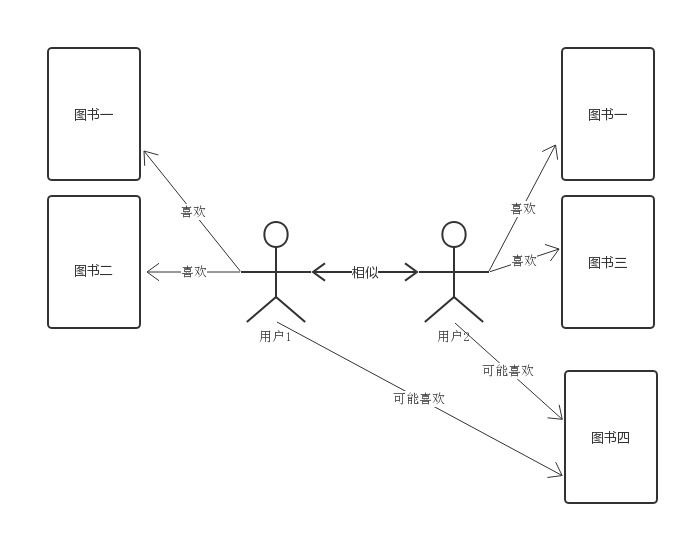


图2.2 基于用户推荐算法介绍

基于用户的推荐算法的计算过程如下：

首先，通过数据集中的用户记录计算用户和图书之间的关联矩阵，如图所示：



图2.3 基于用户推荐算法矩阵示意图

其中，行是不同的用户，列为不同的商品。那么（X，Y）的值就为X用户对Y书籍的评分。此时，我们可以把每一行看做为一个用户对书籍的偏好的向量，然后计算出每两个用户之间的向量距离，这里我们使用余弦相似度来计算。公式如下：

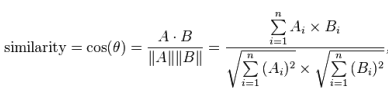


图2.4 余弦相似度公式示意图

从而得出用户向量之间的相似度，如图所示。



图2.5 基于用户推荐相似度计算

这里的数字越接近1，两个用户越相似。最后通过找出与用户1相似度最高的N个用户评价的书籍，删除用户1已经评价的书籍，就是我们需要的推荐结果。

### 2.3.2 基于书籍的推荐算法介绍

上文介绍了基于用户的推荐算法，现在介绍基于书籍的推荐算法。同基于用户的推荐算法相比，基于书籍的推荐算法有所类似，但是在细节上也有所不同。例如用户1经常阅读《西游记》、《水浒传》等图书，那么此时可以预测的是用户1可能也喜欢《红楼梦》这本书[12]。也就是说，我们可以根据一个用户评价的书籍以及评价的分数来进行推荐一个类似的书籍。

由此，我们可以根据以上原理，将基于书籍的推荐算法分为以下步骤：

首先分析各个用户对书籍的评分。

根据分数计算，计算并分析所有书籍之间的相似度。

重新评估计算当前用户的评价高的书籍以找到最相似的N本书籍。

最后向用户推荐这本N书。

原理示意图如下：

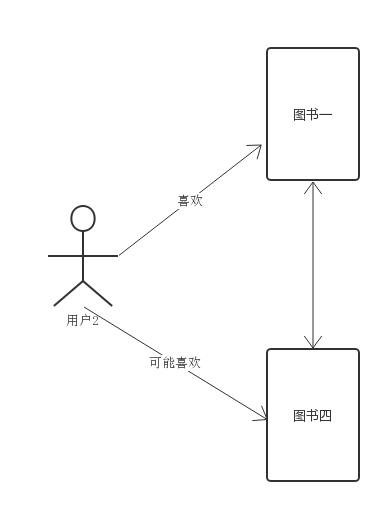


图2.6 基于图书的推荐算法介绍

基于书籍的推荐算法的关联矩阵为书籍与书籍之间的关系，如果用户同时浏览过书籍1和书籍2则（1，1）的值就为1，最后通过计算得出所有书籍之间的关联关系。

从上文中的两种是算法介绍中，我们不难看出基于用户的推荐算法重点在与推荐用户所在兴趣群体中的热点，更加注重社会化。而基于书籍的推荐系统是根据用户的历史评分书籍来进行推荐，更加注重个性化。在书籍推荐方面理论上是基于书籍的推荐能够更好的告诉用户所喜欢看的书籍。但是由于用户的兴趣不同，在某些方面，基于用户的推荐更加擅长[13]。

### 2.3.3基于KNN的推荐算法介绍

我们不用KNN来实现分类，我们使用KNN最原始的算法思路，即为每个内容寻找K个与其最相似的内容，并推荐给用户。相当于每个内容之间都会完成一次两两比较的过程，如果你的网站有n个内容，那么算法的时间复杂度为Cn2，即n(n-1)/2。但是用内容固有属性有一个好处就是因为固有属性一旦创建后基本保持不变，因此算法输出的数据一旦计算好之后不需要重复计算去刷新，也就是对于网站内容而言，原有内容的数据在首次初始化之后可以不断重复使用，只要更新新增内容的数据就可以，数据的统计计算可以使用增量更新的形式，这样可以有效地减少服务器的计算压力。

### 2.3.4 回归算法介绍

本系统利用机器学习领域经典算法。

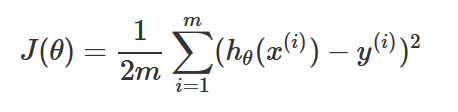
线性回归是利用数理统计中回归分析，来确定两种或两种以上变量间相互依赖的定量关系的一种统计分析方法，运用十分广泛。其表达形式为y = w'x+e，e为误差服从均值为0的正态分布。

线性回归(Linear Regression)是利用称为线性回归方程的最小平方函数对一个或多个[自变量](https://baike.baidu.com/item/%E8%87%AA%E5%8F%98%E9%87%8F)和[因变量](https://baike.baidu.com/item/%E5%9B%A0%E5%8F%98%E9%87%8F)之间关系进行建模的一种[回归分析](https://baike.baidu.com/item/%E5%9B%9E%E5%BD%92%E5%88%86%E6%9E%90)。

线性回归的模型函数如下：



它的损失函数如下：



通过训练数据集寻找参数的最优解，即求解可以得到minJ(θ)的参数向量θ,其中这里的参数向量也可以分为参数和w和b,分别表示权重和偏置值。

求解最优解的方法有最小二乘法和梯度下降法。

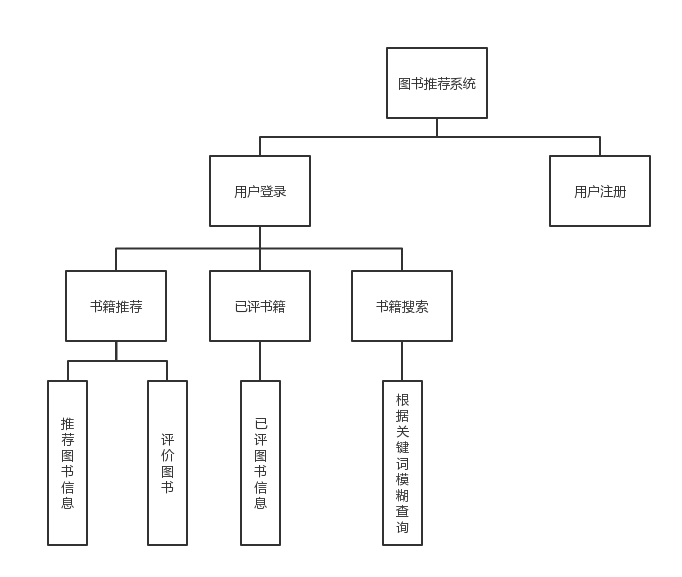
# 第3章 系统的需求分析和整体设计

## 3.1 系统的需求分析和整体设计

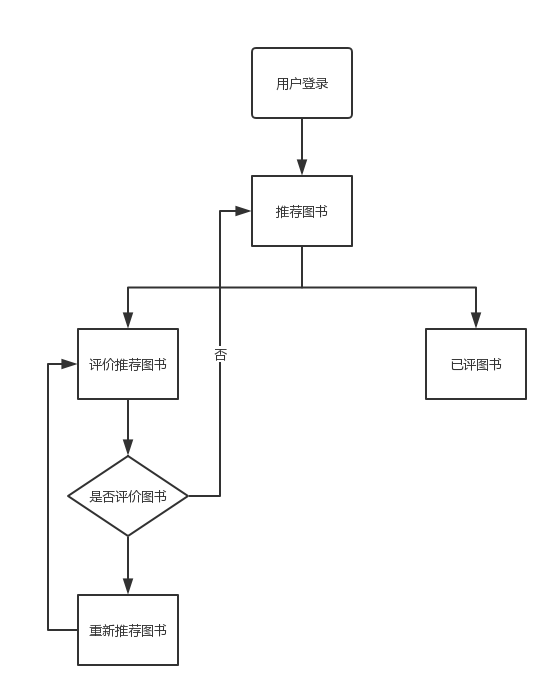
本系统的核心功能就是用户得到个性化推荐图书功能。为实现此功能，需要先获取用户数据以及用户对某些书籍的评分。通过评分数据来进行计算推荐分数，从而得出推荐结果。 若用户再次对书籍进行评分，则再次通过评分数据重新计算推荐结果。

## 3.2 系统的模块设计

本系统的整体框架如下

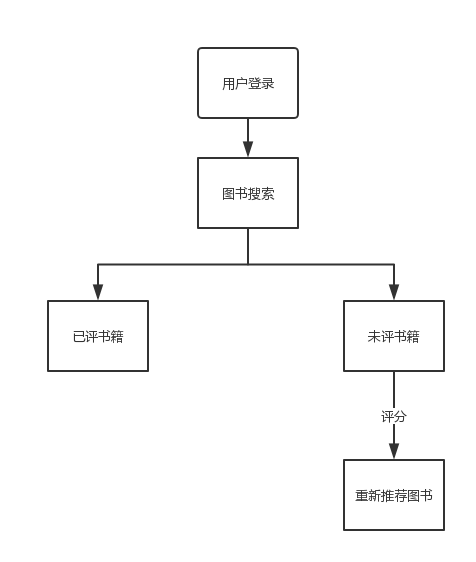


### 3.2.1 用户登录推荐模块



用户登录平台时，通过用户id与密码登录，登陆成功后，首页可以选择对应功能如已评图书、推荐图书、搜索图书。其中通过计算已评图书得出推荐分来给予推荐，在此界面可以选择评价推荐图书，评价后将会重新计算推荐给予推荐。若没有评分则返回。在已评图书界面可以看到自己评价的图书。

### 3.2.2 用户搜索功能模块



用户可以根据书名作为关键词来进行模糊查询，搜到的图书中会有已评分图书和未评分图书，已评分图书会显示评分，对于未评分图书，点击图书，可以进行评分。点击图书会出现书本详情。

## 3.3 系统的数据库设计

本系统一共涉及了3张Mysql数据表。3张数据表中分别存放用户信息、书籍信息以及书籍的评分。

1.用户信息表，表名：bx\_users，具体表结构如下表所示

表3.1用户信息表

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 含义 |
| User\_ID | int | 用户ID（唯一标示） |
| Password | Varchar(255) | 用户密码 |
| Age | Varchar(255) | 用户年龄 |
| Location | Varchar(255) | 用户所在地 |

用户信息表中，UserID自增长作为唯一标示，在注册的时候会自增形成用户ID。用户通过UserID以及Password来进行登录系统。Age为用户的年龄，Location为用户所在地，用于记录用户的个性化特征。

2.书籍信息表，表名：bx\_books，具体表结构如下表所示

表3.2 书籍信息表

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 含义 |
| id | int | 书本ID（唯一标示） |
| ISBN | Varchar(255) | 国际标准书号 |
| Book\_Title | Varchar(255) | 书籍名称 |
| Book\_Author | Varchar(255) | 书籍作者 |
| Year\_Of\_Publication | Varchar(255) | 书籍出版年份 |
| Publisher | Varchar(255) | 书籍出版社 |
| Image\_URL\_S | Varchar(255) | 书籍封面图片S |
| Image\_URL\_M | Varchar(255) | 书籍封面图片M |
| Image\_URL\_L | Varchar(255) | 书籍封面图片L |

其中ID为书籍ID 作为唯一标示，ISBN为书籍国际标准书号。Image\_URL\_S、Image\_URL\_M、Image\_URL\_L三个字段为书籍图标。通过这表我们可以查询到图书的名称、作者、出版年份、出版社以及书籍的封面数据。

书籍评分表，表名：bx\_book\_ratings，具体结构如下表所示

表3.3 书籍评分表

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 含义 |
| User\_ID | int | 用户ID（唯一标示） |
| Book\_ID | int | 书籍ID |
| Book\_Rating | int | 书籍评分 |

其中，Book\_Rating为用户对图书的评分，通过该表我们可以的到用户对某一本图书的具体评分。评分为1-10，其中评分为0的数据为用户隐藏评分。

# 第4章 系统的总体设计与开发

## 4.1 系统总体设计

系统总体采用SSM（Spring+SpringMVC+MyBatis）框架，采用jQuery框架编写。其中main/java文件夹中存放JAVA文件，在resources文件夹中存放项目所需各种属性的配置文件。Webapp文件夹中的static存放前端框架所需文件，views文件夹中存放的jsp页面文件。项目由Maven管理，pom.xml为maven的配置文件。

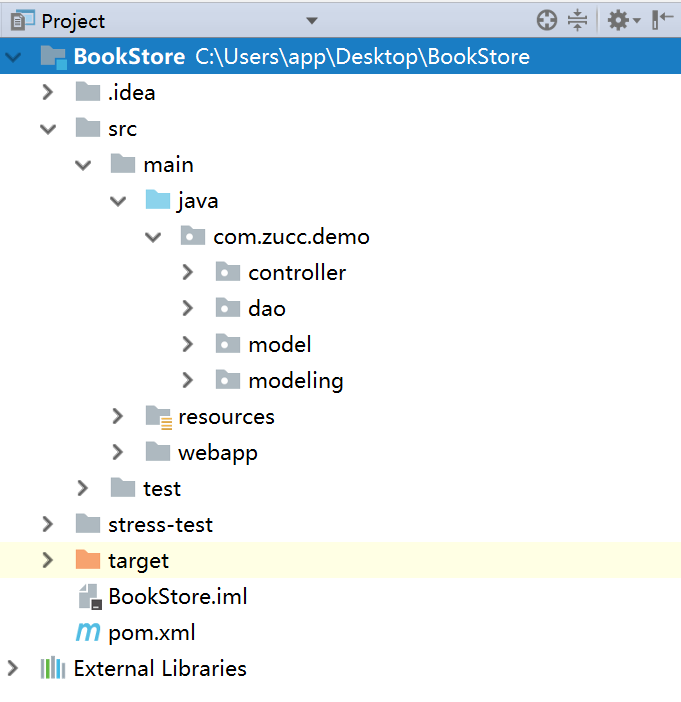
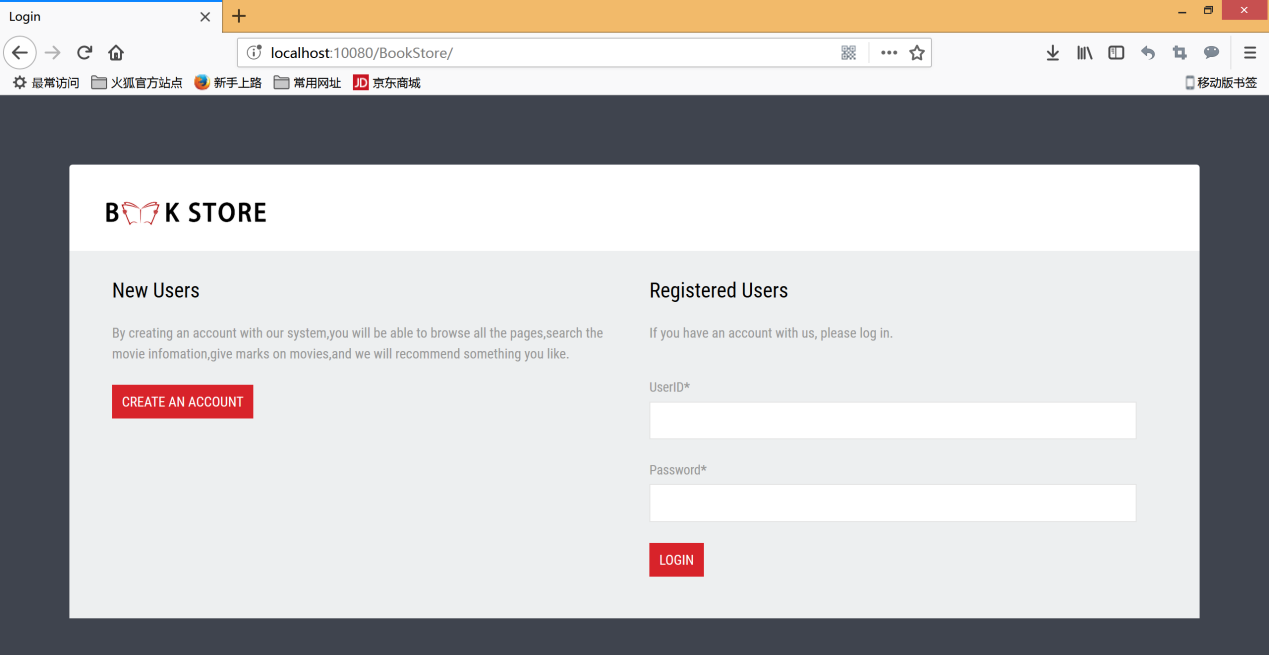
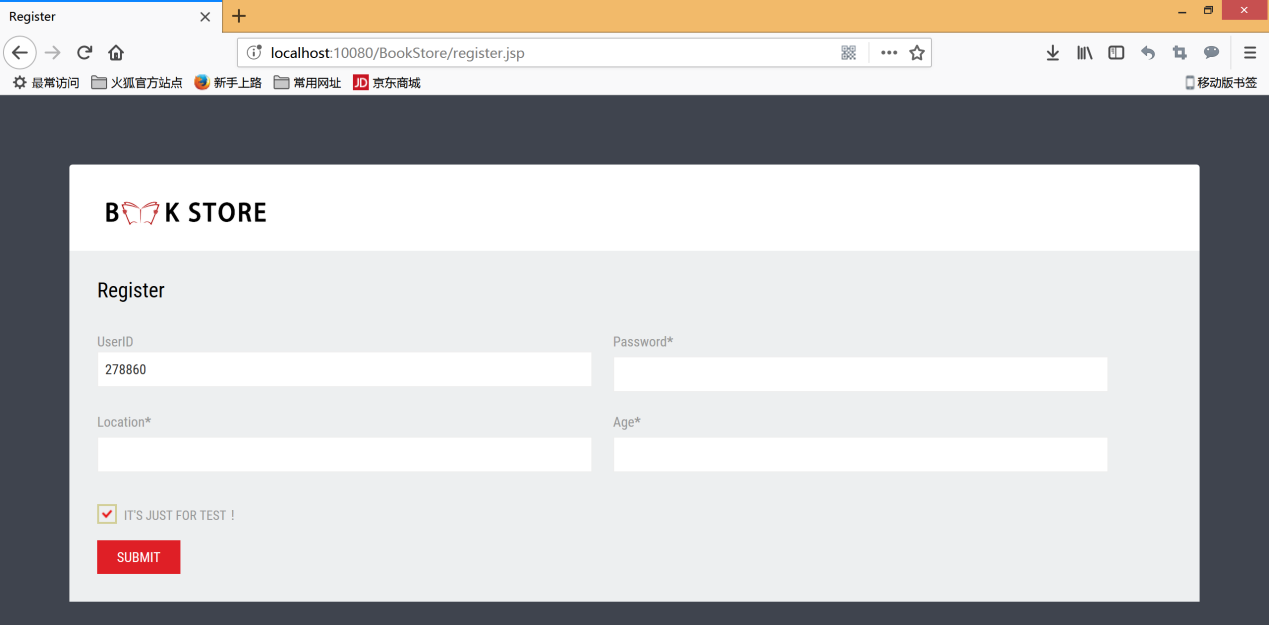


图4.1 整体系统展示图

### 4.1.1 登录注册模块



此为本系统登录以及注册模块.登录界面使用注册时系统自动生成的用户名以及用户自定义的密码进行登录。若没有账号，点击左侧注册用户按钮可以跳转到注册界面。



此为用户注册界面，第一次使用该系统的用户通过填写用户密码、用户所在地。用户年龄并同意测试系统的选项后，点击注册即可完成注册并进入系统。

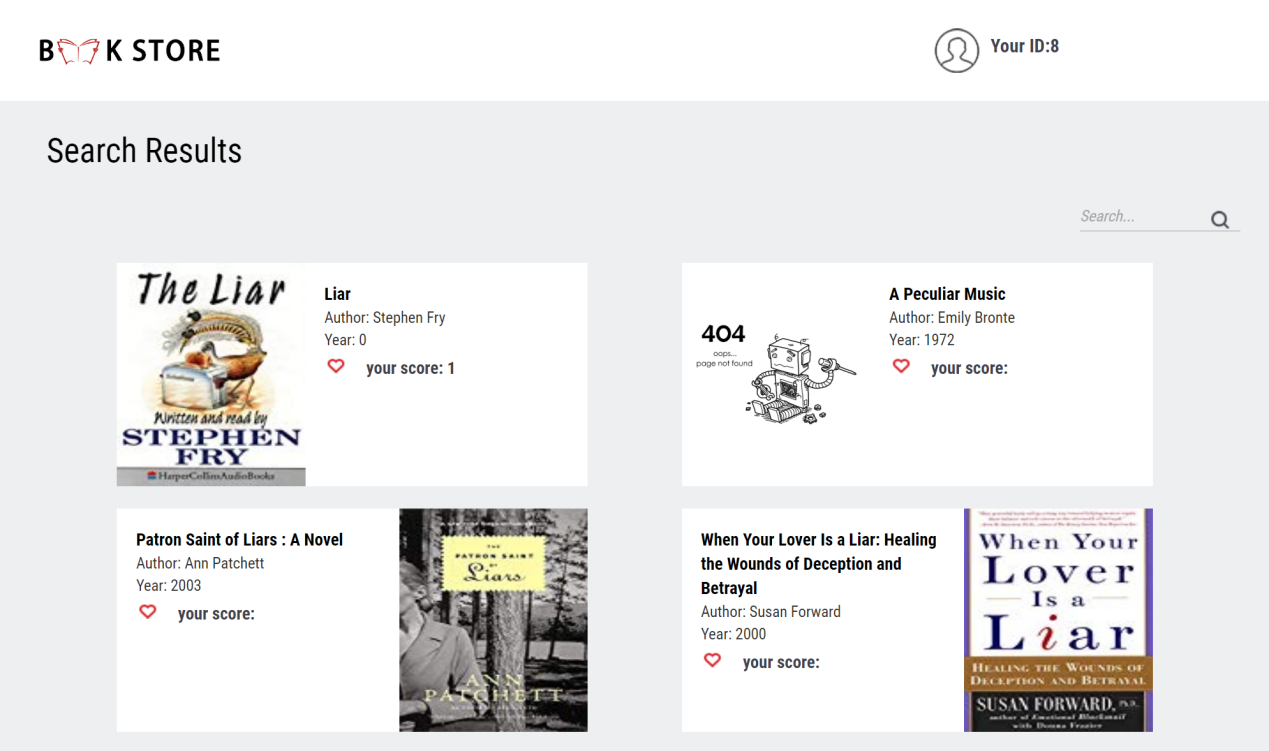
### 

### 4.1.2 推荐模块

### 

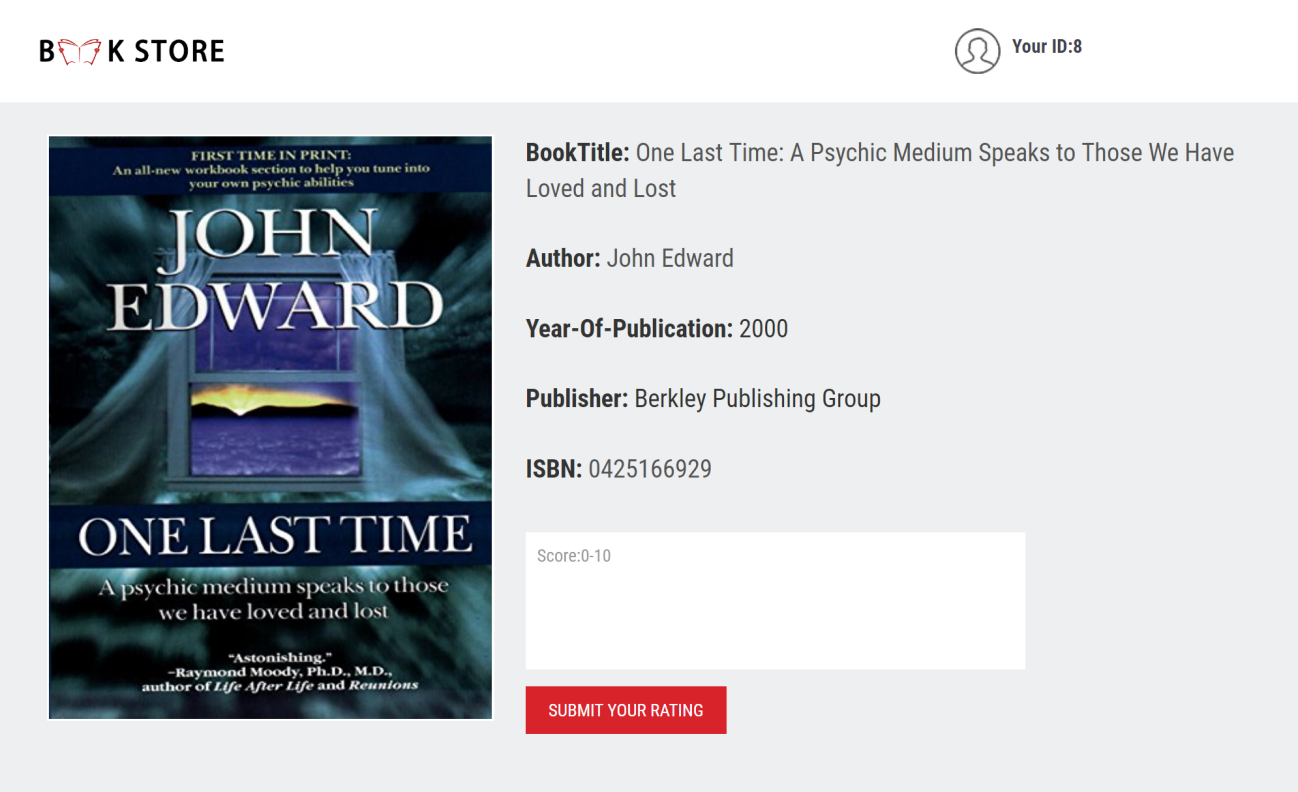
此为用户注册登录后主页，在页面上右上角是用户的用户名，当鼠标靠近用户头像时，会有用户个人信息的图表提示。最右上角的小标可以点击进入用户已评书单页。在主页上会显示推荐书籍。在最上端会有三本评分最高的滚动推荐，下拉则是之后十二本的推荐，若用户为新用户则会推荐当前数据库中排名较高的推荐书籍，并可以进行评分操作。点击评分按钮后（书本）即可进行对该书籍进行评分。在右侧横条输入书本题目关键字点击按钮可模糊查询。

搜索界面如下：



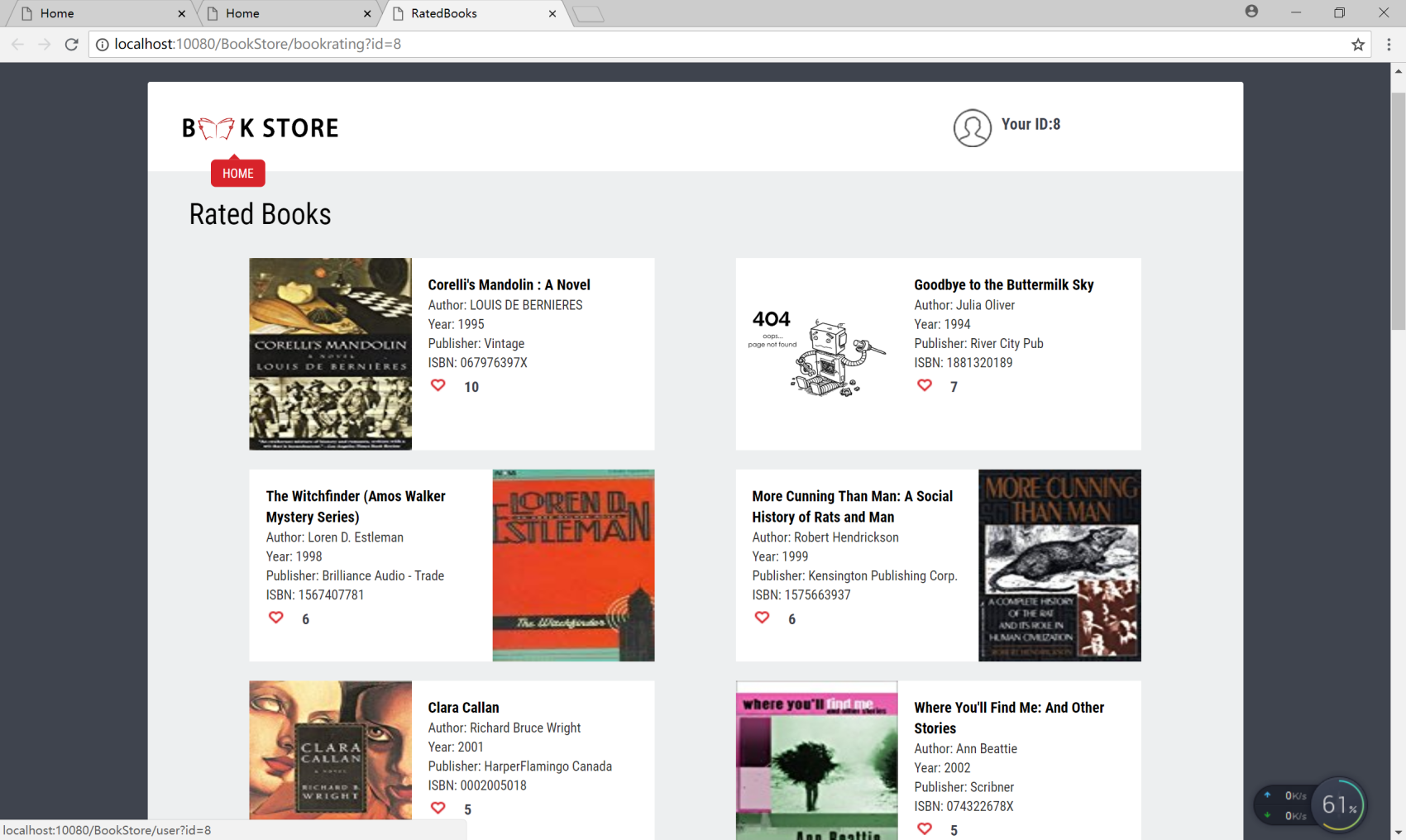
对于搜索出来的书籍，有评分的会显示评分，无评分的可以点击书本题目超链接进入评分界面。

评分界面如下：



每次进行评分后，后台会根据用户评分的数据来进行计算按照新的推荐得分进行推荐书籍。

已评价书单显示如下：

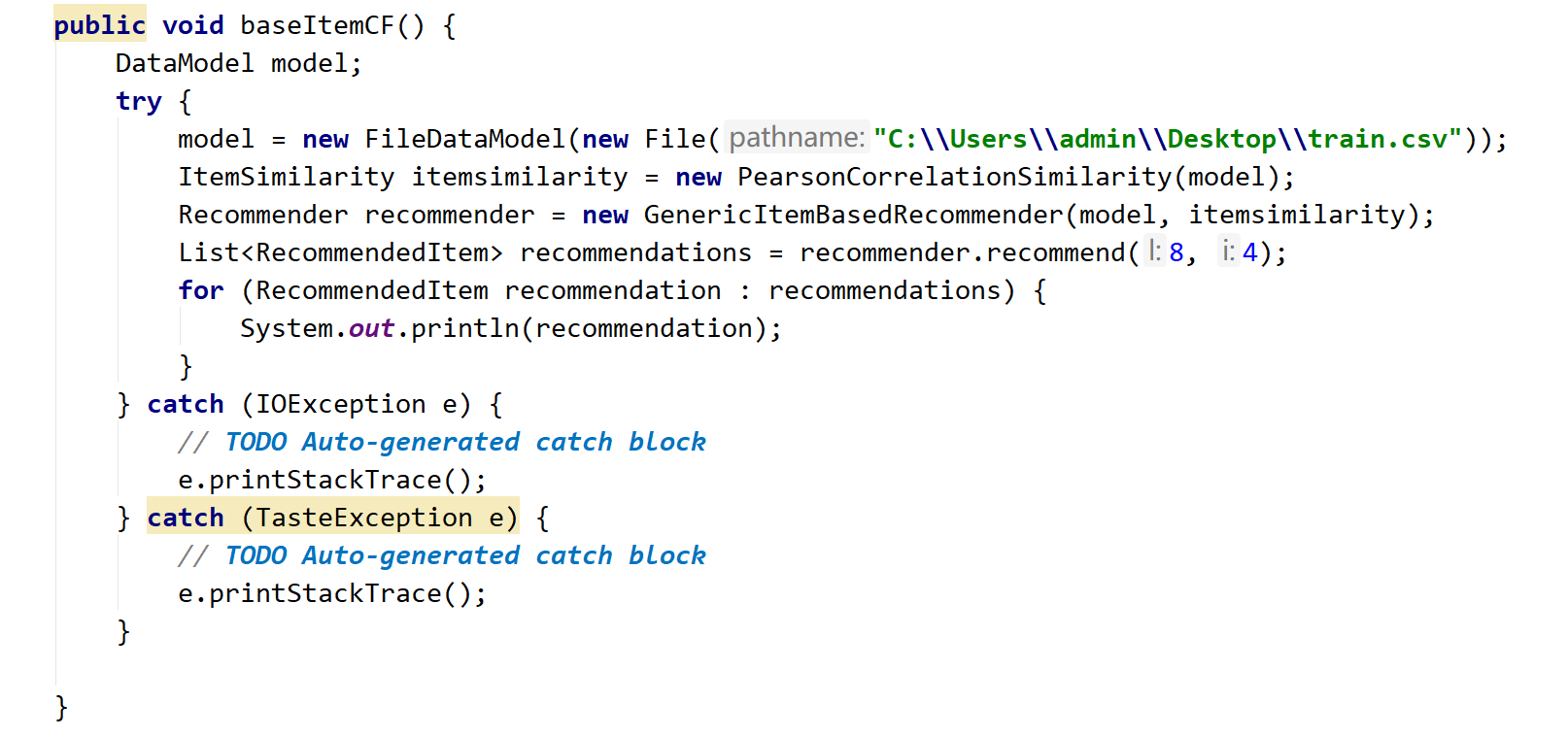


评价后的每一本图书都会在已评图书中显示。用户可以通过此界面查看自己评价过的图书。并可以通过左上角的图表回到首页。

### 4.1.3 推荐算法设计

本系统总体算法设计如下。本系统主要使用了基于书籍的推荐算法。具体设计思路如下，本系统中先通过基于物品的推荐算法对本系统中使用的数据集进行推荐，从而得出基于书籍的推荐结果。系统中的评分数据集以及各个算法计算得出的数据集利用CSV文件进行保存，可以有效减少信息的存储量以及数据计算的运行量。

实现基于书籍的推荐算法的主要函数代码如下：



其中的数据集也是通过文件CSV形式读取文件，CSV文件为基于书籍的推荐算法输出的RATEFILEPATH数据集。然后通过计算不同物品之间的不同评分差来进行线性算法，从而计算得出推荐结果。并将推荐结果书籍以及推荐结果的预测评分输出到CSV文件中。

# 第5章 系统测试

## 5.1 系统测试目的

本系统的测试是为了发现代码中的错误以及在可能出现的问题。使得该系统更加优化，给用户良好的体验。测试也是为了发现各个算法以及数据库方面现有存在的问题。希望能在最短的时间里面花费最好的人力发现各种潜在的错误以及漏洞，确保系统在运行的同时能够更好地提供相应的服务。

## 5.2 系统测试方法

本系统的测试分了以下步骤，通过模块测试来测试系统当前各个模块中是否存在着错误。模块测试要求在各个模块没有出现不必要的错误，在数据读取以及写入数据库方面有没有存在出错。

本系统在测试时，使用大量的已有用户数据进行测试。测试在大量的数据情况下系统是否存在内存溢出等情况。以此来测试系统运行的安全性。通过大量的已有数据集同时来测试推荐算法的合理性，以及整个系统算法设计的合理性。

在该阶段结束后，通过模拟新用户注册来测试新用户注册登录模块，通过此项测试发现新用户登录注册后可能存在的问题来针对性的解决存在的问题。并优化整体系统。并且验证整个系统是否符合需求规格，找出不正确或者矛盾之处并进行优化。

系统测试时必须将系统中的软件以及其所以来的环境资源进行结合，从而在一个较为实际的运行环境下进行测试。这样才能更好地发现系统中可能存在的问题以及可优化的地方。

# 第6章 实验结果检验

## 6.1 测试数据集处理

为了验证本系统中算法的合理性。需要对本系统中所使用以及产生的数据集进行一定的处理，从而对比验证算法的合理性以及优缺点。因此我们从以下几个方面处理数据集：

首先需要清洗初始数据集。初始数据集中为bookCrossing网站中的真实数据，数据中存在着大量的无效数据以及脏数据。因此需要先清洗出初始数据集，从而提高数据集的质量。具体通过用户数据集、书籍数据集以及评分数据集交叉对比，得到标、干净连续的数据。

其次对清洗后的数据集分为两个不同的数据集，通过随机抽取并保存其中的10%的数据作为评价集，利用剩下的90%的数据在系统中进行算法运行。

## 6.2 具体检验指标说明

### 6.2.1 检验指标说明

检验主要利用均方根误差法进行检验，通过推荐分与用户评价的直接偏差的平方和预测次数n的比值的平方根，在系统实际运行过程中，算法推荐得分次数n是有限的，真值使用用户真实的评价数据。方根误差对一组预测数据中的特大或特小误差反映特别敏感，所以，均方根误差能够很好地反映出系统推荐分的精密度。均方根误差，在多次测量一定量时，取该推荐分列的真实误差的均方根误差，称为标准差，用RMSE值表示. RMSE值反映测量数据偏离的程度从真正的价值。RMSE值越小，测量精度越高。因此，RMSE值可以用作评估此测量过程准确性的标准。

使用通过处理后的90%的数据进行系统运行，从而得出一个数据集。然后以此数据集与隐去的10%的数据集做真值对比。

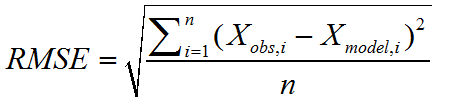
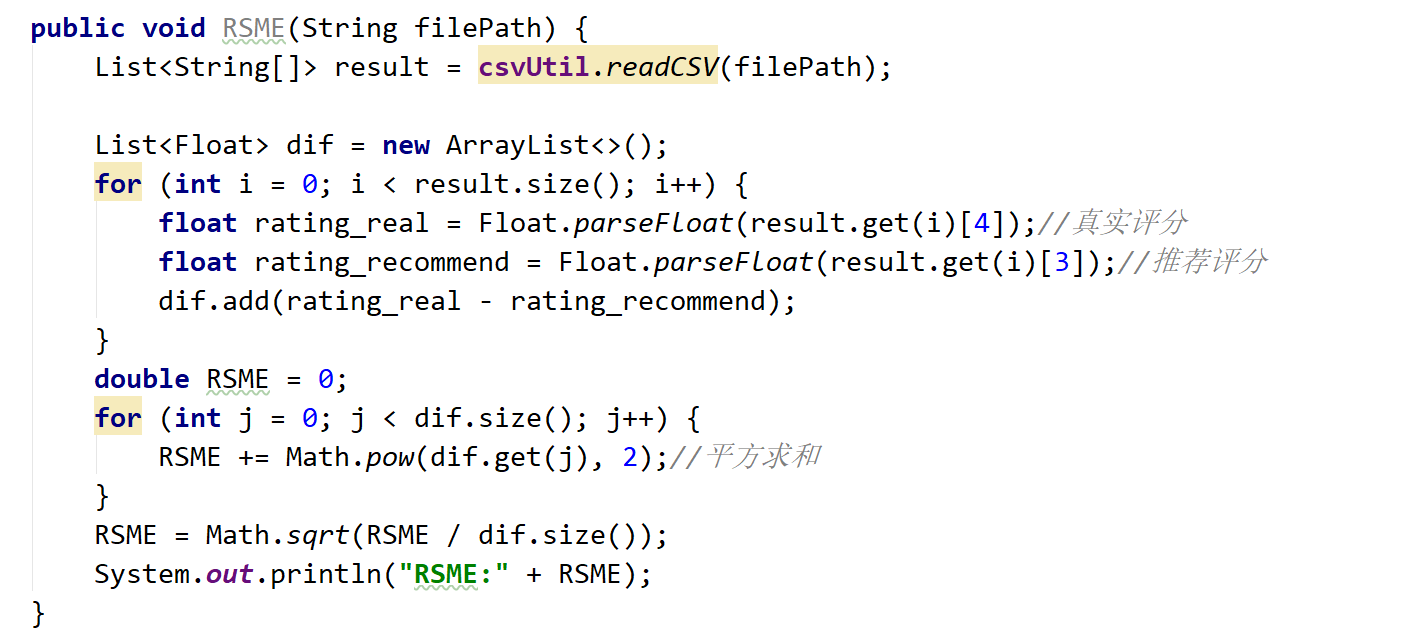
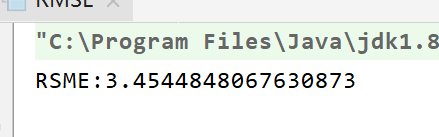


图6.2 RMSE验证检验示意图

主要代码如下：



代码运行结果如下：



# 

# 第7章 总结与展望

## 7.1 系统项目工作总结

本系统项目通过分析在电影、视频以及电子商务上应用广泛的推荐系统，从多个推荐系统的算法入手。对目前运用较为广泛以及实用的三种推荐系统进行了详细的研究和分析。详细介绍了各个算法的思想、程序流程以及各个功能模块的作用，并展示了项目的部分结果。本项目的重点在于通过图书推荐系统的设计与实现，如何高效准确的推荐用户喜欢的图书室整个系统的关键问题所在。由此，本文研究了如何从不同的角度去分析用户数据，并用三种推荐算法来计算得出推荐得分。具体来说，本文主要关注以下几个方面的工作：

1. 学习并研究了基于用户的推荐书算法、基于书籍的推荐算法以及基于Slope One的推荐算法，分析和比较这两种推荐算法的优缺点。分析并处理了数据集中有关数据稀疏度的问题，设计和开发了基于三个推荐算法的图书推荐系统。
2. 研究并学习了Mahout中的一些可扩展的机器学习领域经典算法的实现。并利用其已有资源进行开发设计模型。
3. 研究多种推荐算法的优缺点，并思考设计将多个算法结合在本推荐系统中，学习mahout中的回归模型，并加以实现。
4. 设计并开发了基于SSM框架（Spring+SpringMVC+MyBatis）的图书推荐系统，使用户可以通过web页面访问并使用推荐系统。

## 7.2 系统项目工作展望

尽管在项目的开发上花费了不少的时间和精力，但是在整体上还有存在的着一定的缺憾。系统中还存在着不少可以优化改善的内容。但现阶段由于各方面的限制，本系统还有很大的发展空间，例如对书籍数据的优化，所能获取的书籍数据并不完全，如书籍的类型。今后可以从以下几个方面来提升系统的准确性：

1、准确率的提升。关于推荐准确度的提升可以考虑用到深层次的神经网络进行建模，如采用循环神经网络进行建模，可以更好地剖析用户评分数据集，并生成推荐候选项。

2、系统效率的提升。现阶段使用的Mysql数据库在大量的数据读写上速度有所欠缺，可以考虑使用Nodis数据库（如Redis）存储数据，通过高速缓存技术来提高系统的响应效率。也可以尝试使用Hadoop集群的方式来优化系统。

3、尝试更多的推荐算法。在如今的电商推荐中，往往使用了十几种甚至数十种推荐算法来进行推荐。可以尝试更多的推荐算法加入到系统中，例如上下文感知等，来更好的去优化系统，尝试解决冷启动问题。

4、增加回归模型的特征值。目前的回归模型中的特征值较少，在一些网站对于用户的数据保护比较好。所以对用户个性数据的获取还是有一定的难度，因此在未来的优化中可以和一些网站合作去获取一些用户的特征数据，从而使得机器学习更加完善。