# 基于SSM架构图书管理系统

姓名：陈佰亿 学号：2011030091 班级：20计科二班

1. 背景与意义

图书管理系统是为了有效管理图书馆的资源、提供便捷的图书借阅和归还服务，以及支持图书馆的管理和运营工作。传统的手工管理方法存在许多限制和不便之处，引入计算机技术和信息管理系统可以提高工作效率、减少错误，并为读者和管理者提供更好的服务和体验。

图书馆通常拥有大量的图书、期刊、多媒体资料等资源。通过系统可以对这些资源进行分类、编目和管理，建立准确的馆藏目录，使读者能够方便地查找和借阅所需的资料。系统可以自动记录读者借阅的图书信息，包括借书日期、归还日期、借阅历史等。这样可以减少人工操作，提高准确性，并提供自动化的借还书服务，如自助借还机。系统可以收集大量借阅数据、读者需求、图书流通情况等信息。通过对这些数据的统计和分析，可以帮助图书馆管理者了解读者的阅读习惯、优化馆藏资源、制定采购计划，并为图书馆的决策和规划提供科学依据。通过系统，读者可以在线查询图书馆的藏书情况、预约图书、自助借还书等。这样可以提供更便捷、快速和个性化的服务体验，满足现代人们对高效、智能化服务的需求。

系统可以自动化处理图书馆的各项事务，如图书借还、预约、续借、归还等，减轻人工操作负担，提高工作效率。同时，系统可以生成各类报表和统计数据，简化管理流程，帮助管理者更好地监控和管理图书馆运营。通过系统的资源管理功能，图书馆可以更好地了解图书的流通情况和读者的需求。管理者可以根据数据分析，优化馆藏资源的配置和采购计划，确保图书馆的资源能够满足读者的需求，并降低闲置资源的浪费。系统提供在线查询图书馆的藏书情况、预约图书、自助借还书等功能。读者可以随时随地查询和操作，无需排队等待，提高了借阅的便捷性和效率，增强了用户的满意度。系统收集的借阅记录、读者行为和馆藏需求等数据可以进行深入的分析。通过数据分析，图书馆可以了解读者的阅读偏好、研究热点，优化图书馆的服务和资源规划，提供更符合读者需求的图书馆环境。系统的自动化和集中化管理可以减少人工操作，提高工作效率，减少错误。同时，系统提供了丰富的功能和服务，如在线查询、自助借还书等，为读者提供更便捷、高效的服务体验。通过系统的数据分析和统计功能，图书馆可以了解读者的阅读需求和借阅行为，根据需求调整馆藏资源的配置，有针对性地采购新书，提高资源利用率和满足读者需求。系统提供的数据统计和分析功能为图书馆管理者提供了更科学的依据和参考，帮助他们制定决策、规划图书馆发展、优化服务质量，并提升学术研究和教育教学的水平。图书管理系统是数字化图书馆建设的重要组成部分。通过系统的建设和运营，促进了图书馆的数字化转型，推动信息技术在图书馆行业的应用和发展。

图书管理系统的开发和应用对提高图书馆管理效率、满足读者需求、优化资源管理等方面具有重要的现实意义。它不仅提供了更高效、便捷的图书馆服务，还为图书馆的决策和规划提供了科学依据，推动了图书馆行业的数字化发展。

1. 相关技术

开发图书管理系统所使用到的关键技术包括：

2.1. **框架：Spring MVC、MyBatis**

Spring MVC是一种基于MVC模式的框架，用于开发Web应用程序。它提供了一种分离应用程序的方法，使得开发人员可以更容易地编写可维护和可扩展的代码。MyBatis是一种开源的持久层框架，用于将Java对象映射到关系数据库中的表。它提供了一种简单的方法来存储和检索数据，而无需编写复杂的SQL语句。

劣势：使用框架需要学习一定的知识，学习成本较高；框架也可能会对性能产生一定的影响。

2.2. **编程语言：Java**

Java是一种面向对象的编程语言，被广泛用于Web开发。它具有良好的跨平台性、安全性和可靠性，并且有着丰富的开发工具和库，可以大大提高开发效率。

劣势：Java运行时环境占用的内存较大，可能会对服务器的性能产生一定的影响。

2.3. **数据库：MySQL**

MySQL是一种开源的关系型数据库管理系统，用于存储和管理数据。它具有高性能、可扩展性和可靠性，被广泛用于Web应用程序的开发。劣势：MySQL在处理大量数据时可能会出现性能问题，需要进行优化。

2.4. **网络通信：HTTP协议、WebSocket协议**

HTTP协议是用于Web应用程序之间通信的协议。它是一种无状态的协议，每个请求和响应都是独立的。在图书管理系统中，HTTP协议用于客户端和服务器之间的通信。WebSocket协议一种全双工的通信协议，提供了实时的双向通信能力，适用于实时消息传递和实时更新。

劣势：HTTP协议的无状态特性可能会导致一些问题，如需要在多个请求之间共享数据时需要进行额外的处理。WebSocket提供实时双向通信，适用于实时应用场景，但实现和维护相对复杂。

2.5. **服务器：Tomcat**

Tomcat是一个开源的Web应用程序服务器，用于运行Java Web应用程序。它具有高性能、可靠性和可扩展性，并提供了丰富的功能，如JSP和Servlet容器等。

劣势：Tomcat对内存的占用较大，需要进行优化；同时，Tomcat的配置可能较为复杂，需要进行一定的学习和了解。

1. 需求分析

3.1. 用户需求

在走访调查中，我们与图书馆管理员、工作人员和用户进行了访谈和调研，收集了以下用户需求：

**3.1.1.管理员需求：**

1）能够管理图书馆的图书和资源，包括添加、编辑、删除和搜索图书信息。

2）实现对读者信息和借阅记录的管理和查询。

3）提供统计报表和数据分析功能，以帮助管理员了解图书馆的运行情况。

**3.1.2.工作人员需求：**

1）能够处理读者的借阅和归还图书的操作。

2）生成借阅清单和通知读者归还逾期图书。

3）为读者提供咨询和解答问题的支持。

**3.1.3.读者需求：**

1）能够搜索和浏览图书馆的图书和资源。

2）查看图书的详细信息、可借阅状态和借阅历史。

3）预订图书、管理个人借阅记录和设置提醒。

3.2. 系统功能需求

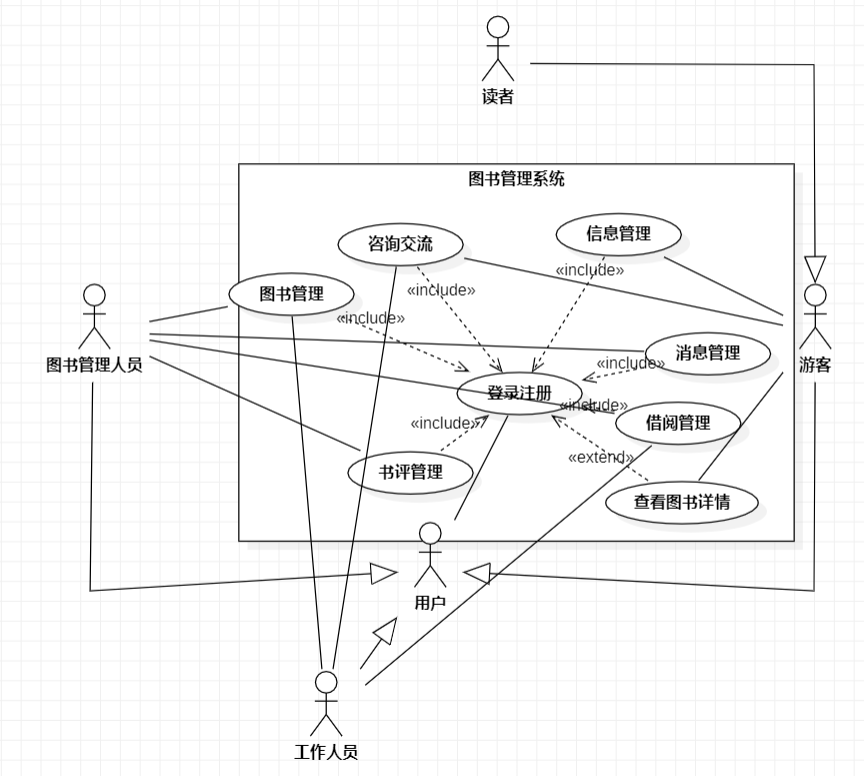


图1-1系统总体用例图

图书管理系统包含以下子功能：

**3.2.1.图书管理功能：**

1）图书信息的增删改查，包括书名、作者、出版社、分类等。

2）图书状态的管理，如可借阅、已借出、预订等。

3）图书搜索和筛选功能，支持关键字搜索和高级搜索。

用例模式图：

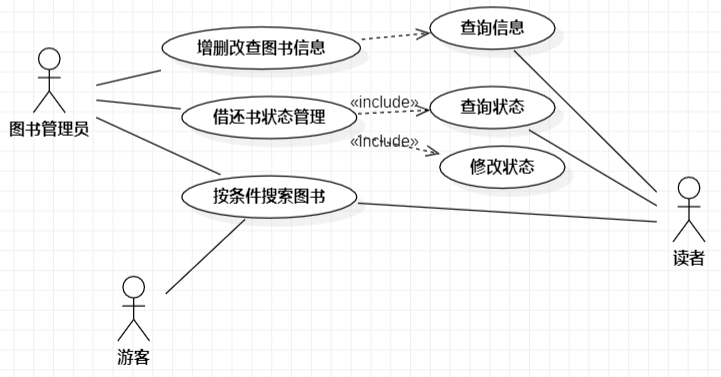


图1-2图书管理用例图

|  |  |
| --- | --- |
| 用例名称 | 图书管理 |
| 用例目标 | 管理员能够添加、修改和删除图书信息。用户能够查询、借阅和归还图书。 |
| 级别 | 子功能 |
| 活动者 | 图书管理员、读者、游客 |
| 前提条件 | 管理员和读者需登录系统 |
| 置后条件 | 数据库中存在正确的管理员与读者信息 |
| 主路径 | 管理员登录系统到后台页面对图书信息进行增删改查，读者登录系统查看图书或者借阅图书或者查看自己的借阅信息。游客在不登陆情况下到主页最热图书页面进行图书检索。 |
| 可选路径 | 无 |
| 例外路径 | 无 |

表格 1.1‑1图书管理用例描述

**3.2.2.借阅管理功能：**

1）读者借阅图书的处理，包括借书、还书和续借等操作。

2）借阅期限的管理和逾期处理。

3）借阅历史记录和个人借阅信息的查看。

用例模式：

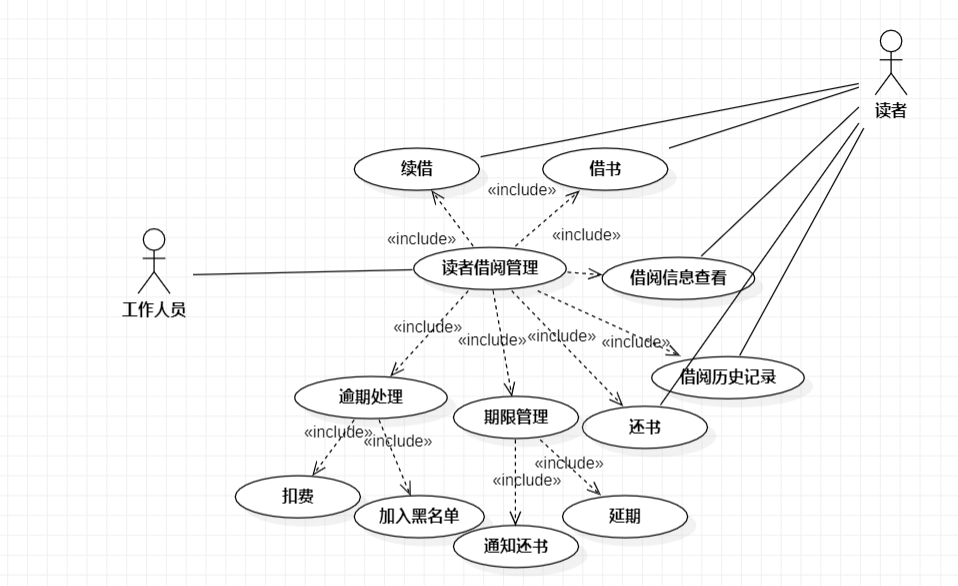


图1-3借阅管理用例图

|  |  |
| --- | --- |
| 用例名称 | 借阅管理 |
| 用例目标 | 管理图书借阅过程，包括将图书借出、还书和续借操作，以及跟踪已借出的图书和超期未还的图书。通过借阅管理，图书馆或者其他类似机构可以更好地维护借阅记录，及时提醒借阅人还书，防止图书丢失或过期。对于借阅人来说，借阅管理可以帮助他们更好地管理自己的借阅记录，及时归还图书并避免超期罚款。 |
| 级别 | 子功能 |
| 活动者 | 工作人员、读者 |
| 前提条件 | 1.必须有一定数量的图书可供借阅，并已经进行图书分类和编目工作。  2.借阅人需要进行注册，获得借阅资格，并填写个人信息、联系方式等登记信息。  3.借阅人需要提供有效证件，并在借书的时候出示借阅证或者其他可验证身份的证件。  4.图书馆或者借阅机构必须建立借阅管理系统，包括记录图书借阅信息、处理还书与逾期等相关事务以及提供查询服务等功 |
| 置后条件 | 1.图书借阅的信息必须被完整、正确地记录下来，包括借阅人的个人信息、借阅时间和归还时间等。  2.图书必须按照规定的流程被借出、归还或续借，确保图书借阅与归还的规范性和准确性。  3.被借阅的图书必须得到妥善的保管，以确保图书状态良好、不受损害且易于借阅。  4.必须设置逾期未还处理方案，确保借阅人不会滞留图书或者损坏图书而不予处理。  5.必须准确计算借阅人的借还情况，及时提醒借阅人还书、缴纳罚款等，并保证归还的图书能够被及时入库并更新借阅状态。 |
| 主路径 | 1.借阅人前往图书馆或借阅机构，提供借阅证或其他可验证身份的证件进行注册和登记。  借阅人查询目标图书，填写借阅单并确认借阅期限；  2.图书管理员处理借阅单并借出图书，包括进行借阅记录、借出图书并进行标记。  3.借阅人带着图书离开并自行安排归还日期，或者进行续借操作。  4.借阅人提前到达图书馆或借阅机构归还图书，交还借阅单并等待管理员的确认，并接受相关催还、逾期提醒等信息。  5.图书管理员处理归还单，并将图书重新上架。  6.定期统计借阅情况并生成统计报告，评估图书借阅的使用状况与借阅人的反馈，并进行相应调整。 |
| 可选路径 | 无 |
| 例外路径 | 无 |

表格 1.1‑2借阅管理用例描述

**3.2.3.用户认证和权限管理：**

1）用户登录和注册功能，确保安全的用户身份验证。

2）不同用户角色的权限管理，如管理员、工作人员和读者的不同访问权限。

用例模式图：

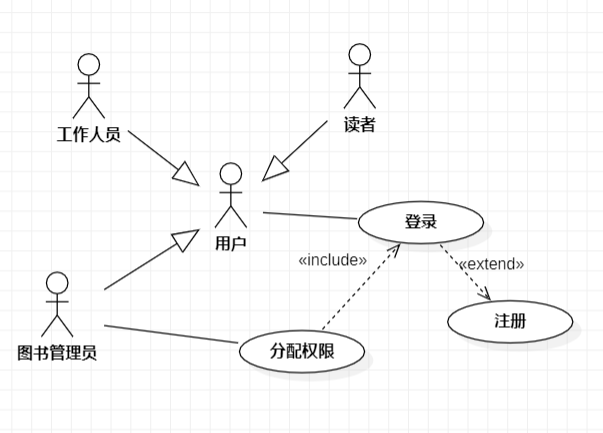


图1-4用户认证和权限用例图

|  |  |
| --- | --- |
| 用例名称 | 用户认证和权限 |
| 用例目标 | 通过用户认证和权限校验，可以检验用户是否是合法的借阅人或管理员，同时也可以确认用户的使用权限和可访问资源范围。 |
| 级别 | 子功能 |
| 活动者 | 工作人员、图书管理员、读者 |
| 前提条件 | 1.必须先创建用户的账户，并保存用户的相关信息，比如用户名、密码、个人信息等。这些信息通常需要经过加密等手段保护用户隐私。  2.系统需要提供多种认证方式供用户选择，比如用户名/密码验证、手机号验证、邮箱验证等。  3.需要在系统中为不同用户角色分配各自的权限，如管理员、借阅人等不同角色拥有不同权限，管理员可以对系统进行管理和设置，借阅人只能进行查询和借阅操作等。  4.需要建立数据库存储用户账户和相应的权限信息，并进行管理和维护，以确保信息系统的正常运行。  5.需要采取相应的安全措施，防止用户的账户信息被盗用或系统被入侵等，如加密存储密码、设置账户锁定和登录错误次数限制等。 |
| 置后条件 | 1.必须保证信息系统在提供用户认证和权限控制服务的同时，系统的安全性能得到保障，防止恶意攻击和入侵。  2.用户认证和权限控制应该尽量简化和优化操作流程，确保用户能够快速、安全、便捷地完成认证和权限控制功能。  3.用户认证和权限控制涉及到系统的核心保障措施，务必保证系统的稳定性和可靠性。如系统崩溃或其他故障问题，将导致系统无法正常工作，进而影响到用户体验和使用体验。  4.用户认证和权限控制的实际使用情况需要得到用户的反馈和信息，以便借阅机构跟进和进行优化与调整。 |
| 主路径 | 1.用户进入系统后，系统要求用户进行身份验证，用户需要输入用户名和密码等信息进行登录。  2.系统验证用户身份和账户信息，如果验证通过，将根据用户角色和权限，跳转到不同的操作界面，如管理员或借阅人界面。  3.在进入操作界面之前，系统会对用户进行权限校验，以校验用户是否有权进行相应的操作，比如管理员才有权进行系统设置和管理操作，借阅人只有借书和归还图书的操作权限。  4.在操作期间，系统会根据用户的角色和权限，快速响应操作请求，执行相应的服务。  5.如果用户在操作期间进行了违规或非法操作，系统会提示用户并进行相应的记录和处理。  6.用户退出系统后，系统会自动销毁用户会话信息和相关缓存数据，保证用户数据的安全性。 |
| 可选路径 | 1.提供用户使用第三方认证服务进行认证登录，如使用社交账号、微信账号等认证。  2.提供用户一个可以通过手机号码、邮箱等信息找回忘记密码的选项。 |
| 例外路径 | 无 |

表格 1.1‑3用户认证和授权用例描述

3.3. 性能需求

**3.3.1.并发用户数**：

系统应支持同时处理多个用户的请求和操作，预计最高并发用户数为10w+。

**3.3.2.响应时间**：

系统对用户请求的响应时间应控制在1~2秒以内，以确保用户体验。

**3.3.3.数据库性能**：

系统应能够高效地处理大量图书和读者数据，保证查询和更新操作的性能。

3.4. 安全需求

**3.4.1.用户身份验证**：

系统应提供安全的用户登录机制，确保只有授权用户能够访问系统。

**3.4.2.数据保护**：

用户个人信息和借阅记录应进行合适的加密存储和传输，以防止数据泄露。

**3.4.3.访问控制**：

不同用户角色应有不同的访问权限，确保敏感信息仅限于授权用户访问。

3.5. 界面设计需求

**3.5.1用户界面风格：**

系统的用户界面应简洁、直观、易于导航，并符合图书馆的品牌形象。

**3.5.2.页面布局：**

信息应合理布局，使用户能够快速浏览和获取所需信息。

**3.5.3.用户反馈：**

系统应提供适当的提示和反馈，以指导用户操作并减少错误并成功使用系统各项功能。

1. 系统设计

4.1. 系统总体架构

基于MVC模式进行设计，系统的总体架构包括：

（1）View层：网页和客户端展示界面。

（2）Controller层：Servlet和Spring MVC，负责Http请求和HTTP响应处理，控制业务流程。

（3）Service层：Spring，业务逻辑层，处理业务操作，调用DAO操作数据库。

（4）DAO层：MyBatis，数据访问层，负责与数据库交互，实现数据的增删改查操作。

4.2. 模块设计

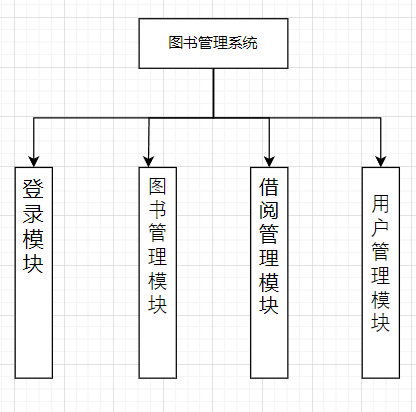


图4.2.1 主要模块图

系统分为四个主要模块：

（1）登录模块：用户通过用户名和密码登录系统，管理员可以通过管理界面进一步进入图书管理、用户管理、借阅管理等模块。

（2）图书管理模块：管理员可以在该模块下进行图书的添加、修改、查询和删除操作。

（3）借阅管理模块：用户可以在该模块下进行借阅和归还等操作。

（4）用户管理模块：管理员可以在该模块下对用户进行管理，包括新增用户、删除用户、修改用户信息等。

4.3. 数据流程设计

系统的数据流程主要包括：

（1）用户和管理员通过登录模块登录系统。

（2）管理员进入各个管理模块进行相关操作。

（3）用户进入借阅管理模块进行借阅和归还操作。

（4）在各个操作过程中都涉及到数据的增删改查操作，数据流转通过DAO层和数据库进行交互。

4.4. 数据结构设计

系统的数据结构主要包括四个主要数据表：

1. 图书表：存储图书基础信息，如图书ID、图书名称、图书作者、出版社、ISBN、价格、剩余数量和总数量等。

数据字典：

| **字段名称** | **数据类型** | **是否非空** | **主键** | **外键** | **字段说明** |
| --- | --- | --- | --- | --- | --- |
| book\_id | bigint | 是 | 是 |  | 图书ID，主键自增长 |
| name | varchar | 是 |  |  | 书名 |
| author | varchar | 是 |  |  | 作者 |
| publish | varchar | 是 |  |  | 出版社 |
| ISBN | varchar | 是 |  |  | ISBN编码 |
| introduction | text | 否 |  |  | 简介 |
| language | varchar | 是 |  |  | 语言 |
| price | decimal | 是 |  |  | 价格 |
| pub\_date | date | 是 |  |  | 出版日期 |
| class\_id | int | 否 |  | 是 | 图书分类ID，关联图书分类表(class) |
| number | int | 否 |  |  | 库存量 |
| cover\_img | varchar | 否 |  |  | 封面图片地址 |
| score | bigint | 否 |  |  | 评分 |

1. 用户表：存储用户基本信息，如用户ID、用户名、密码、用户类型、联系方式和Email等。

数据字典：

| **字段名称** | **数据类型** | **是否非空** | **主键** | **外键** | **字段说明** |
| --- | --- | --- | --- | --- | --- |
| reader\_id | bigint | 是 | 是 |  | 读者ID，主键自增长 |
| name | varchar | 是 |  |  | 读者姓名 |
| sex | varchar | 是 |  |  | 读者性别 |
| birth | date | 是 |  |  | 读者生日 |
| address | varchar | 是 |  |  | 读者地址 |
| phone | varchar | 是 |  |  | 读者电话 |

1. 借阅记录表：存储借阅记录，如借阅ID、用户ID、图书ID、借阅时间、归还时间和借阅天数等。

数据字典：

| **字段名称** | **数据类型** | **是否非空** | **主键** | **外键** | **字段说明** |
| --- | --- | --- | --- | --- | --- |
| ser\_num | bigint | 是 | 是 |  | 借阅编号，主键自增长 |
| book\_id | bigint | 是 |  | 是 | 图书ID，借阅图书的外键关联book表 |
| reader\_id | bigint | 是 |  | 是 | 读者ID，借阅人的外键关联reader表 |
| lend\_date | date | 否 |  |  | 借阅日期 |
| back\_date | date | 否 |  |  | 归还日期 |

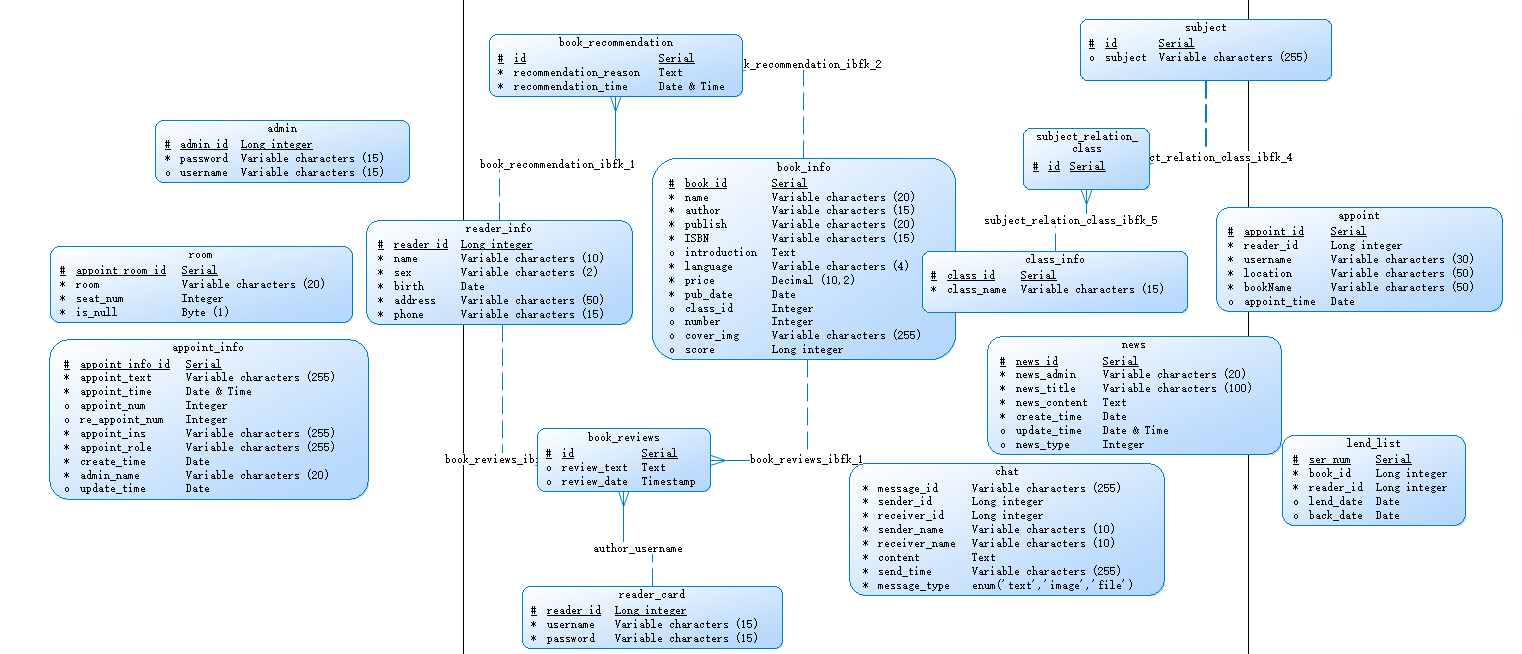
1. 管理员表：存储管理员基本信息，包括管理员ID、管理员用户名、密码、联系方式和Email等。

数据字典：

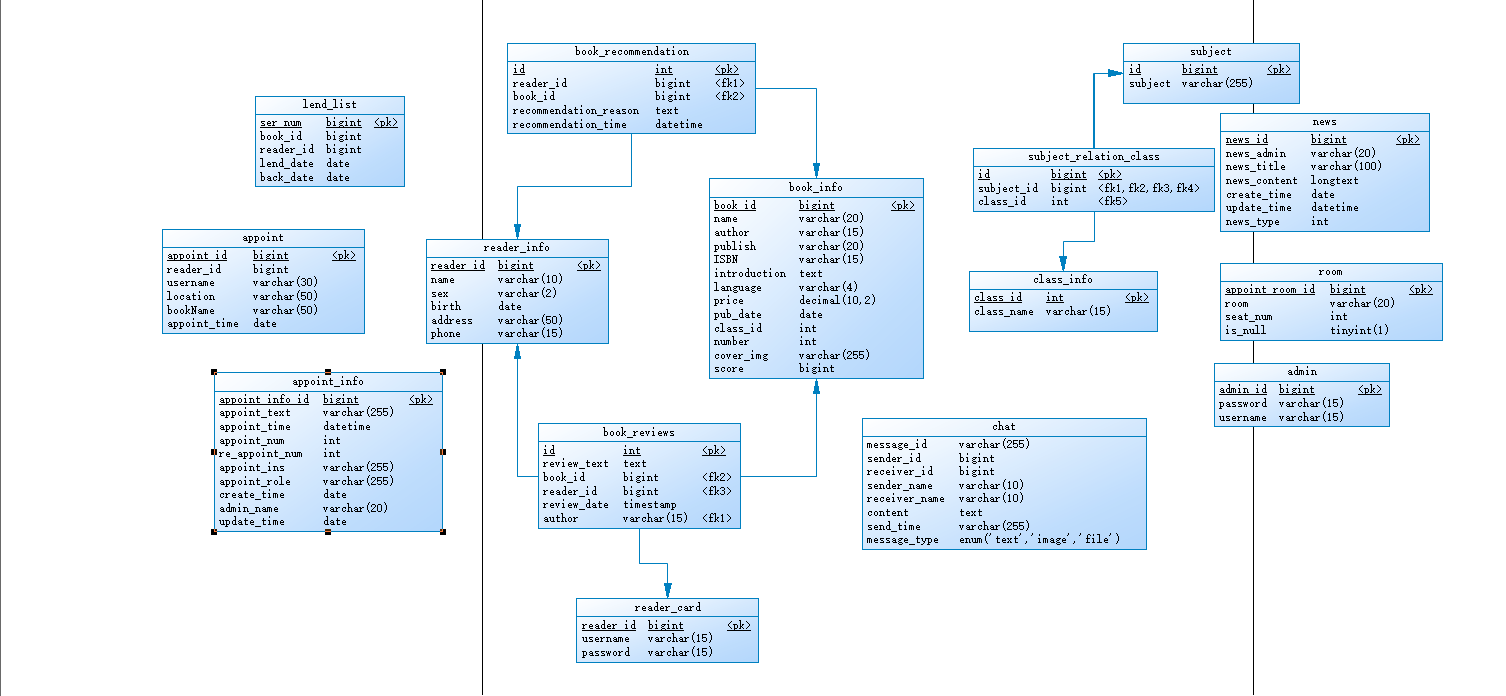
| **字段名称** | **数据类型** | **是否非空** | **主键** | **外键** | **字段说明** |
| --- | --- | --- | --- | --- | --- |
| admin\_id | bigint | 是 | 是 |  | 管理员ID，主键自增长 |
| password | varchar | 是 |  |  | 管理员密码 |
| username | varchar | 否 |  |  | 管理员用户名 |

E-R模型：

1. **数据库概念模型：**

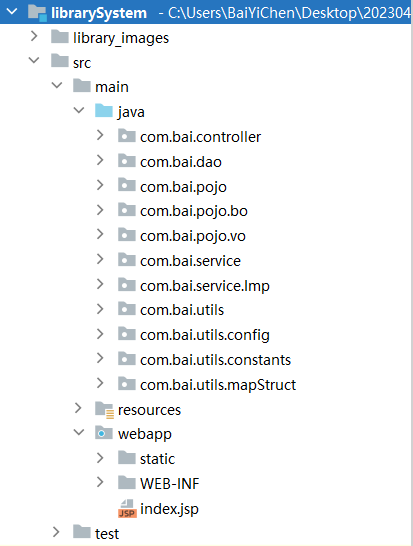


**②数据库物理模型：**



1. 系统实现

工程项目结构及其说明：



项目名称：librarySystem

Package包说明:

|  |  |
| --- | --- |
| com.bai.controller | Controller包，存放所有与前端进行交互的类 |
| com.bai.dao | 存放所有与数据库进行交互的类 |
| com.bai.pojo | 存放于数据库关系映射实体类 |
| com.bai.pojo.bo | 存放所有前端向后端提交数据的业务逻辑类 |
| com.bai.pojo.vo | 存放所有后端向前端传递数据的视图对象 |
| com.bai.service | 存放所有逻辑业务接口类 |
| com.bai.service.Imp | 存放所有逻辑业务接口实现类 |
| com.bai.utils | 存放时间格式化，对象转json等工具类 |
| com.bai.utils.config | 存放所有spring框架配置类 |
| com.bai.utils.constants | 存放所有配置系统的常量类 |
| com.bai.utils.mapStruct | 存放数据库关系映射类向vo、bo转换的接口类 |

5.1.环境准备

在服务器上安装Java、Tomcat、MySQL等相关开发和运行环境，同时建议使用Eclipse等Java开发工具进行编译和调试。

5.2.框架搭建

采用Maven工具创建Maven工程，同时引入Spring、SpringMVC和MyBatis等框架的依赖。

5.3.数据访问层设计和实现

在MyBatis框架中设计和实现数据访问层，包括DAO接口和Mapper映射文件，实现对图书、用户、借阅记录等信息的增删改查操作。

5.4.服务层设计和实现

设计和实现图书管理系统的服务层，通过Spring框架实现依赖注入和控制反转，实现服务层的业务逻辑。服务层负责与数据访问层进行交互、执行业务逻辑、数据校验和用户权限认证等操作。

5.5.控制层设计和实现

设计和实现基于SpringMVC框架的图书管理系统的控制器，实现请求映射、参数绑定、请求响应等功能。控制器负责接收用户请求、调用服务层进行业务处理，并将结果进行封装返回给用户。

5.6.视图层设计和实现

通过JSP、HTML、CSS、JavaScript等技术实现系统的视图层设计和实现，提供用户界面和用户交互功能。

5.7.部署和测试

将应用程序部署到Tomcat服务器上，并进行测试、调试和优化。在测试中还应该包括对系统的安全性、性能、稳定性等多方面进行测试、验收和优化，如SQL注入攻击、数据

1. 总结

图书管理系统对于当代图书馆的使用起至关重要性的作用，为了使用户和图书管理员容易上手，其功能设置不需要十分繁琐，需要有良好的图形界面，让图书管理员和读者可以方便使用。对于现代图书馆而言，拥有大量的书籍，期刊，杂志等，可以快速检索到所需书籍是很关键的问题，所以Mysql数据库的设计是否合理，直接影响到系统的性能。该系统的设计实现比较简单，还有很多地方可以进一步改进，如界面的设计可以更美观，网页加载速度还可以更快一些，由于时间仓促，程序中还有很多需要完善的代码及功能。

参考文献

1. Spring官方文档（https://docs.spring.io/spring/docs/current/spring-framework-reference/）

2. Spring MVC官方文档（https://docs.spring.io/spring/docs/current/spring-framework-reference/web.html#spring-web）

3. MyBatis官方文档（https://mybatis.org/mybatis-3/zh/index.html）

4. 《Web应用开发-基于Spring MVC+MyBatis+maven》（韩冬，电子工业出版社，2018年）

5. 《实用软件工程》（吕云翔，人民邮电出版社，2020年）

6. 《计算机网络》（谢希仁，机械工业出版社，2018年）