Author

xx@gmail.com

Library Management System

Abstract

The system differs greatly from the traditional library management system, it covers all functions of the traditional system, and it appeals to students and teachers because of its new concept, living library, also known as human library, also, it is very useful due to its multiple features such as micro-message reminder, used book trade, recommendation, etc.  
You can get a detailed critical review of the project from this document.

Table of Contents

[1 Objectives 2](#_Toc398045981)

[2 Scope and Constraints 2](#_Toc398045982)

[3 Project Details 2](#_Toc398045983)

[3.1 Background 2](#_Toc398045984)

[3.2 Problem Statement 2](#_Toc398045985)

[3.3 Solution 2](#_Toc398045986)

[4 Architecture 2](#_Toc398045987)

[5 User Requirements 3](#_Toc398045988)

[5.1 Use Cases 3](#_Toc398045989)

[6 Detailed Design 3](#_Toc398045990)

[7 Development Process 3](#_Toc398045991)

[7.1 Version 1.0 – Basic Borrow 3](#_Toc398045992)

[8 Testing 3](#_Toc398045993)

[8.1 Functional Test 3](#_Toc398045994)

[8.2 Usability Test 3](#_Toc398045995)

[9 Conclusion 3](#_Toc398045996)

[9.1 Technical Complexity 3](#_Toc398045997)

[9.2 Innovation 3](#_Toc398045998)

[9.3 Future Enhancements 3](#_Toc398045999)

[10 Appendices 3](#_Toc398046000)

[10.1 Reference Guide 3](#_Toc398046001)

# Objectives

The Library Management System aims to provide an online library management for the students and teachers of Jiangsu Second Normal University to search, borrow, return, comment and rate books, also, it aims to build an online community for them to share knowledge, recommend books, as well as trade used books.

The system introduces a new concept called Living Library, also known as Human Library, which started in Denmark that takes human as an education tool or knowledge carrier for people to borrow, you may talk with these borrowed people about their specific areas, skills or experience so that you can get firsthand materials of the theme you want to explore, or fresh impression about the things you’ve learned before, or brand new understanding about a certain kind of people, etc.

The system is designed to make the living library easier to apply, it accepts reservation and allows 2 types of talking with the borrowed people, face-to-face talk and online talk using Skype. It also supports searching, commenting and rating just as the books.

In addition, the system makes it more efficient and convenient for administrators to monitor books, by looking up the rate, comments and borrowed count of the book, they can track users’ needs and take some action to meet the needs. For example, they may collect more demanded books if a book is highly recommended but there are few to lend, or they will consider to buy more books that are the same kind as the hot books.

However, Since massive workload for advanced library management system and less practical to school’s situation according to previous survey(Students expect more useful service), this project will not fully develop functions as traditional book management system and only collect a small amount of books(useful and popular books) for borrowing. For the time limitation, personal storage will be developed in the future enhancement. This project will focus a great deal on user requirements and experience and play a role for living library. All in all, this smart library community will greatly promote usage of the library and bring a total revolution to the library. The project will contain the following functionalities:

1. Users can register member with valid student ID through the website. They are required to fill up some basic personal information(such as name, birthday, gender, student ID, cell phone number, e-mail address, skype ID, interests, password setting, password reminder ) to pass registration. When they successfully register, they will receive an email of confirmation letter .
2. Users can edit personal information.
3. Administrators can add a new user or delete an user.
4. Users can search a book according to its name, category, publisher, publish time, author, book ID and etc.
5. Users can sort books according to borrowing times, the number of comments or rates.
6. Users can borrow a book via the website; they will received a confirmation letter when they have successfully completed a booking, then, they can pick up a book with a valid student ID.
7. Users can renew the borrowed books, however, renewing is only allowed once and duration is a month.
8. Administrators can confirm a booking when users pick up with a valid student ID, however, a booking is only hold up 48 hours before its cancelation.
9. Administrators can confirm an returning of books when they receive them.
10. Overdue notification email will be sent before a week of booking deadline(a duration of booking is two month).
11. Users can rate (one star to five star(best))and comment books (restrict within 500 words) when they has successfully returned books.
12. Administrators can add, delete or edit books
13. Administrators can delete untrue or over-offensive comments of books .
14. Users can post used books sale information by simply clicking “I want to sell used books”.
15. Users can post used books demand information by simply clicking “I want to buy used books”.
16. Users can search used books.
17. Users require to enter certain books’ information(such as name of book, author, publisher, publish time, book id, price and condition of book) before posting.
18. Users can upload pictures of used books.
19. Users can edit or delete their posted information.
20. Users can contact a buyer or seller by clicking “contact” button and they chat by email. For purpose of protection of privacy, real email addresses of buyers and sellers are hidden by using anonymous email addresses.

For example:

1. Administrators can deleted postings irrelevant to the used books.
2. Users can search human books according to different subjects(for example, math, computer, history, physics and etc ).
3. Users can sort human books according to borrowing times, the number of comments or rates.
4. Users can borrow a human book vie the website and choose its available schedule; two reading forms of meeting is optional for users: online meeting(read via Skype) or offline meeting.(read in an arranged study room)
5. Users must submit reading form and meeting topic by filling up a form before booking.
6. Choosing online reading: an user will receive a confirmation letter containing a meeting time, meeting topic and a human book’s Skype ID. Similarly, a human book will receive a confirmation containing a meeting time, meeting topic and an user book’s Skype ID
7. Choosing offline reading: an user will receive a confirmation letter containing a meeting time, meeting topic and meeting place with a number of study room. Similarly, a human book will receive a confirmation containing a meeting time, meeting topic and meeting place with a number of study room.
8. Users can contact wanted human books by simply clicking “contact” button. They communicate by email. For purpose of protection of privacy, real email addresses of buyers and sellers are hidden by using anonymous email addresses.
9. Users can cancel a booking, yet, it must be ahead of six hours before meeting.

Users and administrators will both receive a confirmation letter of cancelation.

1. Users can rate (one star to five star(best))and comment human books (restrict within 500 words) when they has successfully returned books.
2. Users can apply to be human books. (They need to download application form and send finished to librarian committees to review, after they are approved, they will receive an email with an unique invitation code sent by librarian committees and they can use a code to complete human books’ online registration.)
3. Human books can edit personal information.(such as book title, subject belonging, special filed, interests, introduction, available schedule for booking, reading forms(must choose one at least), contact information and etc.)
4. Human books can cancel to be human books.
5. Administrators can add human books, delete human books or edit information of human books.
6. Administrators can delete untrue or over-offensive comments of human books
7. The most interesting part is that users allow to post questions and choose its subjects in “question and help” area. All the human books related this subjected can receive a “help” email with an answering link(for example the question topic is related to Computer subject, then, all the computer subject human books will receive the “help” email), so they can answer the question or give suggestions by simply clicking the link (lead to the place for posting). It will ultra smart and convenient for users’ study. However, users only are allowed to question once per day because of limit of resources.
8. Administrators can backlist users based on following situations:

* Users are often no-shows for picking up a book.
* Users do not return books or return books on time.
* Users often post junk information irrelevant to used books.
* Users are often no-shows for borrowing human books.
* Users often cancel the meeting with human books.
* Users often post malicious comments.
* Users often post malicious questions in “question and help” area.

# 2 Scope and Constraints

The online library community (smart library) system provides service of borrowing books, renewing books and returning books. It also has a publishing platform for used books sale or need. To assist on students’ academic study and development of healthy mental, a living library platform allows users to borrow real human books via its online reservation system. This project is not redesigning a library management system and it will not collect large amount of books as original library due to massive workload(maximum over 200 million books). It only focuses on user’s needs and the most useful and valuable books. Providing a high-quality reading resource is very unique way to immediately attract students and arouse their interest of reading. The most exciting thing is that this project allows me to become the first person who converts concept idea “Living Library” to more practical reservation system.

Development technologies, tools and programming languages used in this project show as follows:

**Development technologies:**

Struts2 formwork

Hibernate formwork

Spring formwork

ExtJs

**Programming languages:**

Java

JavaServer Pages(JSP)

Servlet

MySql 5.5

**Development tools:**

MyEclipse

**Optimizations for system:**

Apache Tomcat web server and load balancer

Cache of Hibernate

Partition and creating index for database

The online library community system contains three main modules: user module, administrator module and Email notification & contact module.

1. User module is divided into five major parts as follows:

* User function
* Book-borrowing
* Used book
* Living library
* Comments & Rating

1. User function will have following functions:

* Registering
* Editing personal information(student ID, birthday, gender, email, Skype ID and etc )
* Login/Logout

1. Book-borrowing will have following functions:

* Searching book information
* Booking books
* Renewing books
* Returning books

1. Used book will have following functions:

* Searching used book information
* Posting used books sale information
* Posting used books demand information
* Editing postings
* Contacting by anonymous email

1. Living library will have following functions:

* Searching human books
* Booking a human book
* Canceling booking
* Registering to be a human book (with valid invitation code)
* Canceling to be a human book

1. Administrator module is divided into four major parts as follows:

* Users management
* Books management
* Used books posting management
* Comments management

1. User management will have following functions:

* Adding a new user
* Editing users information
* Deleting an user
* Blacklisting users
* Promoting a user (become an administrator)

1. Books management will have following functions:

* Adding a new book
* Editing books information
* Deleting books
* Managing human books

1. Used books posting management will have following functions:

* Deleting users’ posting

1. Comments management will have following functions:

* Editing books comments
* Deleting books comments
* Editing human books comments
* Deleting human books comments

1. Email notification system

* Sending a confirmation letter when the users have successfully complete registration.
* Sending the reset password link to the user’s email address if the user forget the password and request to retrieve it.
* Allowing administrators to deliver hold pick-up, overdue by email.
* Sending a confirmation letter to users and human books when the reservation has been made.
* Sending a confirmation letter to users and human books when the reservation has been cancelled.

# 3 Project Details

## Background

Located in Nanjing City, Jiangsu Province, Jiangsu Second Normal University is a full-time undergraduate school. At present, it has 16 institutes and nearly 7,000 full-time students. In recent years, the university is developing rapidly, and continues to enlarge dimensions. However, the original library management system is outdated and in very low usage, and it only provides service inside of library not convenient for students who expect to inquire book information outside the library . Compared with online modern technologies, the system is lack of interaction with users and less functional. It is an old and boring system only used for book-borrowing. After a lot of excellent and innovative ideas talking with my sponsor. I am very glad to be granted an opportunity to develop an online library community for an university.

This project is not for redeveloping a library management system due to massive workload and less practical to school’s situation. This online library community aims to attract more students to participate and interact with each other, sharing reading experience and giving guides to new reader. It also brings ease and convenience to students to store personal data(future enhancement) and trade their used books. Moreover, users can book human books and get the greatest inspiration or tutorials more than ever before because of available of using online living library reservation system. On the other hand, Compared with old fashion ways , now, librarians as system administrators has ability to interact with users, manage and update books more positively and efficiently due to available of books’ comments and rates. I believe with generally increasing participation of students in this online community, more and more students will find interest in reading and studying, finally, if with constant future enhancement of this system, it is possible can generally replace the traditional library.

The project aims to It has four main subsystems as follows:

1. Simple book-borrowing system with robust database structure and data backup.
2. Revolutionary online living library platform, instead of borrowing books it provides borrowing people service.
3. Personal storage space for users.(future enhancement)
4. Used books trading publishing platform.

Sponsor Background:

Located in Nanjing City, Jiangsu Province, Jiangsu Second Normal University is a full-time undergraduate school. Covering an area of about 35 hectare, it currently has three campuses of Caochangmen, Pukou and XiaoYing. Focused on normal education, it also opens some non-normal majors for Jiangsu economic and social development. At present, it has 16 institutes and nearly 7,000 full-time students. In recent years, the university is developing rapidly, and continues to enlarge dimensions. The school has three libraries, collecting about half million covered almost all subjects students need. Library has grown from two locations within 20 staff to its current three locations within 32 staff. Each library provides private learning rooms and public learning areas.

Living Library Background:

Living Library (known as Human Library as well), started in Denmark. The concept idea is taking human as an education tool or knowledge carrier to enhance the people’s participation. These borrowed people may be male and female, old and young, and most of them are with controversial social statuses like transgendered people, right wing of radicals, aids patients, homosexuals, Mohammedans, strippers and alcoholics. With people’s curiosity, they can learn each other better so as to reduce the prejudice and discrimination in the society.

## Problem Statement

Although the library management system is available in Jiangsu Second Normal University, it is out of date. A lot of books are in low rate of usage because of isolation of system and lack of updated statistics. Users can only browse books inside of the library, a lot latest books’ information also need to be acknowledged in the library. Database of system has no backup service and processing speed is low when a lot inquires executed simultaneously. From the survey(see appendix ) which I aim to find out most needs for future school library and drawbacks for current , majority of students think they shall be able to search books outside of the library and get updated and latest information of books. They shall share reading experience and rate books some features similar to Amazon book market. They shall have their own space to store personal electric documents. On the other hand, they complain that current system is boring, less functional, UI design is bad and there is no way to find out the latest or popular books. Library has no activities or any communications between learners. They hope get some tutorial as well. I based all the facts of current system and the result of the survey, I decide to develop the online library community for the JSNU(Jiangsu Second Normal University)which is more suitable for current internet modern society.

Following problems need to be solved:

* How to design a reasonable system architecture that can enable all subsystems works?
* How to build up an optimized database avoid data redundancy?
* How to set up a backup service for database?
* How to handle the solution when massive queries occur?
* How to convert concept idea of “living library” to a real practical online platform?

(There is no existing system example for reference)

* How to enhance and stabilize the living library platform, make it not an reservation system but also the place to ask and solve problems(same great feature will be added similar to Jelly App. ) ?
* How to design an attractive and user-friendly UI interface which changes users’ stereotype of an old original book-borrowing system and makes them love to use?
* How to keep users and living books’ privacy (Email address will be anonymous between email communications, and Skype ID only be noticed when an reservation has been completed. )?
* How to set up personal storage for each user? (future enhancement)
* How to integrate external APIs (Application programming Interface) to make designed system work? (For example, Java Email service or even SMS service)
* How to build up an email reminder to notify information such as books about to overdue, appointment has been made(or canceled)between user and living books etc?
* How to design used books trading platform which can upload pictures and search wanted books information and also with some restrictions and rules to prevent spamming?
* How to sort the books based on rates, popularity, the number of comments and etc?
* How to produce some web security strategies to protect the safety of the system?

Exiting living library’s problems and limitations:

Jiangsu Second Normal University does not exist a living library, it is total new concept for them. My sponsor shows great interests in this idea and is willing to provide me any helpful resources for designing the library(For example, study rooms for communication, some candidate teachers can become living books). In Great Vancouver area, as far as I can find, there are only two organizations providing service of living library, one is Douglas College, the other is Coquitlam Library. The form of these living libraries is simple. They offer some activities, inviting some typical and critical people (for example, ex-offenders, HIV carriers, gays and etc ) and sorting them in different titles of books for people to borrow. After a dialogue, readers learn from their stories and experience, therefore , readers can reduce prejudice and remove stereotype.

However, I believe current living library has serious limitations(see figure). Firstly, there are very few book categories, most of them are centralized in very critical and typical area(for example, homosexual, transgender, HIV, racism etc).On the other hand, living books related to academic and research areas are almost zero. Hence, I believe the mode of the traditional living library is not suitable for requirements in contemporary Universities. It can only offer spiritual assists and therapies for students to develop a healthy self-concept, but zero contribution to their academic study. Secondly, the reservation of traditional living library only stays on massive paper work. Meeting is only face to face, not available for people chatting via online chatting tools such as Skype or MSN, and information of living books can be only inquired in the library. All these facts show that a current living library is inefficient, isolated and limits communications among library administrators, users and living books. It can hardly attract students to participate. Thirdly, the traditional living library is lack of efficiency and availability. Users can only borrow living books during library’s activities. If there is no activity, there is no chance to borrow any living books. It is inconvenient for students to process constant study. Fourthly, it is hard for librarians(system administrators )to track performance of each living book due to lack of users’ feedbacks. Lastly, the form of activity is singleness, when users encounter time or distance limitation, they cannot solve problems.

Figure 1 Current Living Library’s problems

## Solution

# 4 Architecture

**开发技术**

本系统通过整合Java轻量级的开源框架Struts2、Spring和Hibernate，使用MyEclipse集成开发环境，MySql数据库管理系统，Tomcat作为Web服务器，开发了具有耦合度低、维护性好和扩展性高的图书管理系统。实现的技术框架如图所示。系统的服务层采用Spring，持久层的Hibernate通过Spring提供的支持类集成到Spring中，Web层则通过Struts2做页面的控制和数据传递系统严格采取Web层、服务层和持久层三层体系结构，只允许上层的程序可以调用下层的程序，达到层与层之间松耦合的目的。



系统技术框架

1.Struts2框架

Struts2是一种基于MVC的Web应用框架。MVC模式的核心手段是解耦，MVC模式把整个应用程序划分成模型、视图、控制器三个部分，然后严密控制三个部分之间的通信，从而得到一个结构清晰、功能分布合理、可重用、可扩展、可维护的应用程序。使用Struts2能够方便地控制页面的跳转，而不用像Servlet那样需要写复杂的代码才能实现，其强大的值栈和OGNL表达式能够有效地对数据进行传递和控制。对于企业级应用而言，服务器端的验证是必不可少的，任何一个业务逻辑在被调用之前，都应该对用户提供的数据进行严格的校验，传统的数据校验需要程序员手写代码来实现，并且校验代码和业务逻辑混在一起，Struts2为数据校验提供了校验框架，可以轻松地对数据进行校验并将数据校验和业务逻辑分离，不会为业务逻辑带来麻烦。Struts2还有强大的标签库和过滤器，这些都为企业级应用开发提高了效率。

Struts2的运行流程很简单，当用户的请求到达控制器FilterDispatcher，控制器根据用户提交的URL和在struts.xml中的配置来执行相应的Action，Struts2的Action实现了与Servlet API的解耦，使用Action不需要继承和实现任何类和接口。Action处理完用户的请求后，将处理后的结果跳转到Result中预先配置的页面，并把在Action中获取到的数据展现给用户。

2. Hibernate框架

Hibernate是一个基于Java的开放源代码的持久化软件，它对JDBC做了轻量级封装，提供ORM（Object Relation Mapping，对象-关系映射）服务。ORM就是利用描述对象和数据库之间映射的元数据，自动地把Java应用程序中的对象持久化到关系数据库中的表。使用JDBC连接数据库和对数据库的操作需要写大量的代码才能实现，而使用Hibernate只需要配置好Java实体类到数据库表之间的映射关系，以及使用Hibernate提供的数据查询和获取数据的方法，便可以减少大量的代码，提高了开发效率。Hibernate提供的一对多、多对一、多对多等对象关联关系，以及对立即加载和延迟加载的支持，可以方便的对数据进行操纵。其面向对象的HQL查询语句可以根据对象与数据库表之间的映射关系生成相应的SQL语句，简化了复杂的SQL语句的书写。Hibernate还提供一级缓存和二级缓存，合理地使用二级缓存可以有效减少对数据库访问次数，提升系统的整体性能。

3. Spring框架

Spring是分层的Java SE/EE应用的一站式的轻量级开源框架，以IoC（Inverse of Control：控制反转)和AOP（Aspect Oriented Programming：面向切面编程）为内核。控制反转也被称为依赖注入，使用依赖注入，对象是被动接收依赖类而不是自己主动去查找。依赖注入将对象间的依赖关系交由Spring进行控制，而不需要专门的去关心什么时候该实例化对象，可以更专注于业务逻辑的实现。Spring对面向切面编程提供了强大支持，通过将业务逻辑从应用服务中分离出来，实现了内聚开发。

Spring、Struts2和Hibernate三者整合，可以将Hibernate访问数据对象的SessionFactory接口交由Spring的IoC容器去管理，这样只需要在配置文件中配置好便可以在Hibernate访问数据库时不要要手动的去创建SessionFactory实例。还通过Spring的事务机制实现声明试的事务管理。 这可以在不修改源代码的情况下只需要修改配置文件便可以实现不同数据源之间的切换。Spring与Struts2整合后，Struts2的Action的实例化不再由Struts2管理，而是交由Spring的IoC容器去管理，这样可以降低程序的耦合性，将控制器与业务逻辑分离开来，为以后的维护和扩展带来了便利。

**4. ExtJs**

ExtJs是使用JavaScript编写的与后台技术无关的Ajax框架。ExtJs的强大之处就在于它的各种组件，使用它们可以构建丰富多彩的前端页面。ExtJs的表格控件功能非常完善，可以对数据进行排序、缓存、拖动、隐藏和编辑等功能，Form表单控件不仅外观绚丽还支持前端的数据校验，Layout布局可以对整个页面进行布局而不用书写大量的JavaScript代码。使用ExtJs可以尽可能的减少页面的跳转甚至不需要页面跳转便可以完成整个页面下的所有功能。

**开发工具**

MyEclipse是基于Eclipse一款十分优秀的J2EE集成开发工具，MyEclipse比Eclipse有强大的插件集合支持，能够较好的支持开源产品。它是企业级的开发平台，是对Eclipse的扩展。在MyEclipse中，可以整合数据库和J2EE的开发，以及程序编译、运行、测试、部署和发布，这些都可以在MyEclipse集成开发环境中完成。MyEclipse对Java开源框架Struts2、Spring和Hibernate有较好的支持。常用的集成开发工具还有Intellij IDEA，但其在插件方面比较欠缺，并且会占用较大的内存。所以，MyEclipse作为J2EE开发的首选工具，可以使开发变得快捷，加快应用软件的开发效率。

**系统性能优化**

**1. Tomcat集群和负载均衡**

图书管理系统因其特殊性有时用户的并发量很大，单一的服务器会无法满足需要处理的负荷量，或者当服务器出现故障无法继续正常运行时，这时就要使用Tomcat集群技术来解决这些问题。

集群系统由一台或多台服务器计算机上运行的两个或更多服务器软件实例组成，这些服务器计算机彼此协同合作以透明地服务客户端的请求，从而从客户端角度看，整个集群租是一个高可用性服务。网站的集群系统使用Apache的反向代理来实现负载均衡，Apache会将用户的请求分别转发给不同的Tomcat服务器，以此来实现Tomcat的集群。集群部署图如图所示。



Tomcat集群部署

**2. Hibernate缓存**

缓存是位于应用程序与物理数据源之间用于存放复制区域的内存区域，其目的是为了减少应用程序对物理数据源访问的次数，从而提高应用程序的运行性能。当程序需要到数据库中查询数据时，首先会去缓存中查找，如果命中便不会去访问数据库。

**3. 数据库表建索引以及表分区等**

**旧版图书管理系统结构图**



旧图书管理系统结构图

# 5 User Requirements

## 5.1 Use Cases

参考内容，目前系统的所有功能点：

**有关注册：**

用户可以注册成会员，填写基本的信息（姓名，生日，性别，学号，兴趣，email地址，skypeID，进行密码设定，密码找回问题。）

成功注册后，会收到系统的注册成功的确认信。

用户可以修改个人信息

管理员可以添加，删除用户和修改用户信息。

**有关借书还书：**

用户可以根据索引搜索比如：书的种类，书名，书的出版时间，书的作者，书刊号来进行书的搜索。

用户可以对书进行排序比如：按书的借阅次数，按书的评分，按书的评价数。

用户可以网上借书，借书成功后会收到一封系统的确认信，然后凭借学生证可以去图书馆取书。

用户可以网上续约图书，续约只能一次，续约时间为一个月。

确认借书：管理员可以在确认学生身份后，在系统里点借书成功。否则不取，2天后系统会取消这次借阅。

确认还书：管理员可以在确认收到图书，在系统里点还书成功。

图书的借阅期限为2个月，在快过期一周时，系统会自动发送发快到期提醒邮件到用户邮箱。

用户在还书成功的时候，对图书进行评分（1星到5星）和文字评价（400字之内）。

管理员可以添加，删除图书和修改图书的信息（书的种类，书名，出版时间等等）。

管理员可以删除不符合实际对图书负面用户评价。

**有关2手书交易：**

用户可以发布交易帖子，上图。

用户可以修改，删除帖子。

用户可以发帖卖书（直接点击，“我要卖书”）

用户可以发帖求书（直接点击，“我要求书”）

【重点，难点，需要攻克的地方：

在帖子中，用户可以一键沟通，通过点击一个“联系对方”，就可以自动出来一个匿名的邮件地址（中间转发的邮箱地址？），双方可以进行匿名的邮件地址联系。不暴露隐私。

（用户在发布帖子的时候必须要求填写email地址，否则不能发布。）】

管理员可以删除和2手书发布无关的帖子。

想法：介于发布的特殊性，可以强制发帖的模式，必须填写必要的信息。

卖书的时候：

比如必须要填写书的名字，书刊号，作者，新旧程度。出售的价格等。

求书的时候：

只要填写书的名字，作者。

**有关借人的系统：**

用户可以搜素“人书”，根据学科种类。

用户可以对“人书”排序，按书的借阅次数，按书的评分，按书的评价数。

用户可以网上预约想要学科的“人书”。

用户可以根据“人书”的时间表，选择见面的时间。

用户可以选择线上见面（skype）和线下见面（在学习室见面）。

用户在选择线上见面的时候，当预约完成时候，会收到一封确认预约成功信。里面有见面时间， 将要讨论的主题，还有“人书”的SKYPE\_ID。

与此同时，被借阅的“人书”， 会收到一封确认被预约成功信。里面有见面时间， 将要讨论的主题，还有借阅者的SKYPE\_ID。

（注：选择线上见面，用户必须提交讨论的主题，和自己的SKYPE\_ID。）

用户在选择线下见面的时候，当预约完成时候，会收到一封确认预约成功信。里面有见面时间，将要讨论的主题，见面的地址（图书馆学习室的号码）。

与此同时，被借阅的“人书”， 会收到一封确认被预约成功信。里面有见面时间，将要讨论的主题，见面的地址（图书馆学习室的号码）。

（注：选择线下见面，用户必须提交讨论的主题。）

用户还可以选择一键交流功能（邮件地址被系统匿名之后之间的交流，之前2手书里的难点。）

用户可以取消预约（必须提前6小时，否则不能完成！）

用户在取消预约成功后，会收到一封确认预约取消成功信。与此同时，被借阅的“人书”也会收到一封被取消预约的信。

用户可以在成功借阅“人书”后，对“人书”进行评分（1星到5星）和文字评价（400字之内）。

用户可以申请成为“人书”。（需要下载后填写申请表和简历然后发邮件到评审委员会的邮箱，同意申请后，收到一份邮件里面有唯一的邀请码（用过作废）。难点！不是每个“人书”都可以注册的！必须筛选加面试!）

用户可以注册“人书”，必须要用邀请码注册。填写基本信息，如：学科，书名，擅长领域，简介，email地址（用户不可见），skypeid（用户不可以见）。可以借阅的时间段（时间必须大约30分钟）。可供借阅的方式(必须选1个)：线上或者线下？

提交后被自动入库。

“人书”可以修改信息。

“ 人书”可以取消馆藏。

管理员可以修改“人书”信息和取消“ 人书”的馆藏。

管理员可以删除对“人书”不符合的负面用户评价。

管理员可以增加，减少，修改“人书”的借阅表。

（分配学习室的号码给每本“人书”，借阅方式：线上或者线下？或者都行？）

**关于借人系统里的问答帮忙系统：**

用户可以一键提问，提出问题（必须选涉及的学科）（次数限制：一天只能提问2次）

提问后体统会自动把提问的信息发送到所对应学科的每一本“人书”电子邮箱。“人书”可以回复邮件进行解答。

**关于黑名单系统：**

管理员可以根据下面几种情况加用户进黑名单。

1. 借书成功后，经常不去取书的用户。
2. 不还书的用户。

3. 经常发和2手书无关信息的用户。

4． 经常借“人书”，失约不去的用户。

5． 经常借“人书”，平凡取消的用户。

6. 经常恶意评价的用户。

7． 经常恶意提问的用户。

# 6 Detailed Design

功能模块图



图书管理系统总结构图

模块细分

图书管理系统主要分为两部分：用户模块和管理员模块，此外还包括邮件通知/联系模块。

1.用户模块主要有：

(1) 用户功能

(2) 借书

(3) 二手书交易

(4) 借人

(5) 评论功能

a.用户功能

用户注册

用户信息修改

用户登录和注销

b.借书

查询图书信息

借阅图书

图书续约

还书

c.二手书交易

查询二手书信息

发布卖书信息

发布求书信息

邮件匿名联系

d.借人

搜索人书

预约

站内信/邮件通知

取消预约

加入/取消馆藏

e.评价功能

还书时对所借图书发表评分和评论

对所借人书进行评价

2.管理员模块主要有：

(1) 图书管理

(2) 二手书管理

(3) 评论管理

(4) 用户管理

a.图书管理

图书入库

图书信息修改

图书删除

人书管理

b.二手书管理

二手书卖书信息管理

二手书买书信息管理

c.评论管理

图书评价管理

人书评价管理

d.用户管理

添加用户

修改用户信息

修改密码

用户禁用

普通用户升级为管理员或将管理员降级为普通用户

3.邮件通知

图书过期自动邮件通知

二手书交易邮件联系

人书预约/取消预约邮件通知

# 7 Development Process

## 7.1 Version 1.0 – Basic Borrow

**Details of Development**

**Iteration 1: Groundwork**

**Goal**

The Goal for this iteration is creating a foundation for further development. It includes technology choosing, setting up framework and integrating with each other for whole system, and then basic functions of login service to make sure system work stable and well-connected.

**Details**

**Chosen Development technology and tools**

SSH framework

SSH framework is short term for three open source frameworks (Struts+Hibernate+Spring). It is a very handy development method used by many programmers and helps them quickly setup clear, well-reusable and easy maintained web applications. It contains four layers- **P**resentation Layer, Business Logic Layer, Data Persistence layer, and the Data Base Layer. The Struts framework conducts process control and separates pages and code based on MVC (Model-View-Controller) mode in the Presentation Layer, then the Spring framework releases proper decoupling of objects in the Business Logic Layer. Lastly, in the Data Persistence Layer, the Hibernate framework perfectly setups connection to database and easily operates data such as add, delete and modify. The chat bellowed shows division of tasks of SSH framework.



Figure2 Division of tasks of SSH framework

**Integration of Spring framework and Hibernate framework**

It is easy to integrate Spring framework with Hibernate because of its extensibility and openness. Spring framework provides unified management of data source. Instead of configuring configuration file - hibernate.cfg.xml in the Hibernate, it only needs to configure data source and control attributes for the Hibernate in the applicationContext.xml file in the Spring. Meanwhile, in order to easy to use, Spring framework provides Hibernate Template which can easily control database without tedious work. In Spring, database connection and transaction management all begin with setting up SessionFactory. SessionFactory only requires one instance in the application, so the instance can be created by Spring and injected into related dependent objects. The Code for configuration file is as follows:

<!—configure Hibernate database source -->

<bean id="dataSource" class="org.apache.commons.dbcp.BasicDataSource" destroy-method="close"

<!-- get database connection information from configuration file -->

p:driverClassName="${jdbc.driverClassName}"

p:url="${jdbc.url}"

p:username="${jdbc.username}"

p:password="${jdbc.password}" />

<!-- instantiate SessionFactory-->

<bean id="sessionFactory" class="org.springframework.orm.hibernate3.LocalSessionFactoryBean"

<!-- reference data source -->

p:dataSource-ref="dataSource"

<!-- specify mapping files of Hibernate -->

p:mappingDirectoryLocations="classpath:/com/books/domain">

<property name="hibernateProperties"><props>

<!-- set the dialect of Hibernate -->

<prop key="hibernate.dialect">

org.hibernate.dialect.MysqlDialect

</prop>

<!-- background output SQL statements operated by Hibernate and format -->

prop key="hibernate.show\_sql">true</prop>

<prop key="hibernate.format\_sql">true</prop>

</props></property></bean>

<!-- configure HibernateTemplate -->

<bean id="hibernateTemplate"

class="org.springframework.orm.hibernate3.HibernateTemplate"

p:sessionFactory-ref="sessionFactory" />

**Integration of Spring framework and Strus2 framework**

Spring not only offers an outstanding open source MVC framework referred as SpringMVC, but also supports well of integration with other web frameworks. During the integration of Spring and Struts2, the main step is making Spring’s IOC container manage Struts2’s Action, then the Action classes of Struts2 are able to require their instances through Spring. Before the integration, in Struts2’s configuration file - struts.xml, we need to convert Struts2’s request processor to Spirng’s request processor, and upload ApplicationContext when starting Web. In addition, using comment@Controller can realize class injections for these Action classes need to be injected. Configuration code of web.xml is as follows:

<context-param>

<!—configure applicationContext.xml for files upload paths-->

<param-name>contextConfigLocation</param-name>

<param-value>/WEB-INF/applicationContext\*.xml</param-value>

</context-param>

<!—use ContextLoaderListener(from Spring) to create ApplicationContext-->

<listener>

<listener-class>

org.springframework.web.context.ContextLoaderListener

</listener-class></listener>

**Tomcat server**

Tomcat server is a lightweight application server, and it is widely used to handle the normal amount of concurrent accesses in small or medium system. It is also the first choose for debugging JSP program. Moreover, Tomcat server has good compatibility and supported by many well-known software companies. It runs stable and has a good development prospect. Deployment diagram of Tomcat server for this system is as follows:



Figure3 Deployment diagram of Tomcat server

The following steps show the installation of Tomcat server plug in Eclipse:

1. In Eclipse, open display window from window – show view – servers, as shown in figure 4:



Figure4 Display window of Tomcat server

1. In Eclipse, right click New – server in Servers window, it will let you to select version of Tomcat, as shown in figure 5:



Figure5 select version of Tomcat server

1. After done select version, click Next to access the place where to add directory for Tomcat server, then choose the proper directory and change JRE to JRE6, as shown in figure 6:



Figure6 add directory for Tomcat server

1. After adding directory, click Next to the place where to upload project to Tomcat, choose the proper project and click Fish, as shown in figure 7:



Figure7 upload the project to Tomcat server

**Eclipse**

Eclipse is an open source, free and integrated development environment (IDE). It is a development tool to develop applications by mostly programming in Java. It contains a very useful plug-in system, including Java Development Kit, JDK.

The following steps show the installation of Eclipse:

1. Install Eclipse in the desire directory, for example, in E:\eclipse.
2. Unzip and install plugs of Eclipse and find folders named features and plugins, copy them to the same name folders in directory of E:\eclipse respectively.
3. Download JDK and configure system environment variables.

**MySQL database**

MySQL is an open source relational database management system that runs as a server providing multi-user access to a number of databases. The SQL phrase stands for Structured Query Language. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL is a popular choice of database for use in web applications, and is a central component of the widely used LAMP open source web application software stack. Free-software-open source projects that require a full-featured database management system often use MySQL.

The following steps shows the installation of MySQL plugin in Eclipse:

1. In Eclipse, open the Data Source Explorer window from window – show view – data Source Explorer, as shown in figure 8:



Figure8 Date Source Explorer window

1. Right click in Data Source Explorer windows, and open New Connection Profile window, as shown in figure 9:



Figure9 New Connection Profile window

1. In New Connection Profile window, select desire type of database for adding, and access to New Driver Definition window by double clicking. Then, configure all the values according to chosen database, as shown in figure 10:



Figure10 New Driver Definition window

4．After done configuration，click finish to complete.

**Basic functions of login module**

After setting up system frameworks and platforms, login in module will be first priority to concern. Since realizing basic login module is very necessary for test former system frameworks configuration and integration, and also I will more clear and easier explain the concept of SSH by this practical instance. Login module services two actors – normal user and administrator. Normal users can login and perform all the functions provided by the system to meet their demands, conversely, administrators perform their duties to manage system, they login the backstage of system to manage administration module.

State chart diagram of Login Account illustrated by the following figure 11:



Figure11 Login Account state chart diagram

Stat chart description

Actors: Normal User and Administrator

Actors access to login page, and required to input valid user name and password. System will verify User name and password, if is correct, login successfully, otherwise, login failed.

**Database and code design**

The below figure 12 shows design of user table in database. All Users fell mainly into two classes: normal users and administrators, presented by user state 1 and user state 2 respectively. I also add certain attributes (Skype\_ID, email etc) related to further design of book-borrowing and living library system into the table.

User table (figure 12) of database shows as follow:



Figure12 user table of database

The system code structure belongs to three layers: Dao layer, Service layer and Action layer. Dao layer uses Hibernate to operate the underlying database; Service layer mainly process business logic; Action layer main job is controlling front web pages, passing parameters and calling Service layer to process business logic.

System code structure is showed as below:



Figure13 System Code Structure

Since Dao layer uses Hibernate, it is easy to retrorsely generate entity classes though the database table structure. It is unnessary to require mapping files from Hibernate becase of supportive commens of Java. Smilarly, Struts2 and Spring also use comments, in this way, programmer can simplify work of configurate files and easily manage configurations.

Login page for administrators is showed by the below figure 14:

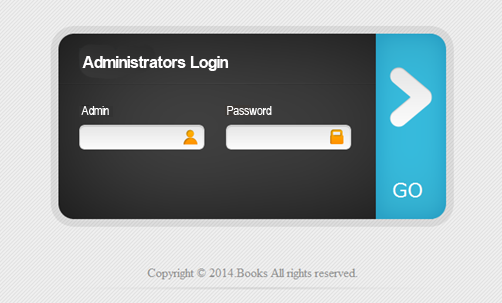


Figure14 Login page for administrators

The following code shows user name and password will be introduced to Action of Structs after their submisson.

@Action(value = "login", results = {@Result(name = "success",type = "redirect", location = "/admin/main.jsp"),

@Result(name = "failure", location = "/admin/login.jsp")})

**public** String doLogin() {

List<User> users = userService.login(user);

**int** ret = users.size();

**if**(ret > 0) {

**return** "success";

} **else** {

**return** "failure";

}

}

Action layer calls service layer to process user login logic, then Action decides actions for login successful or failed. This is how does Struts2 play the role here. Service layer processes user login logic and calls Dao layer to operate database. After verification of information passed, administrators will be able to access to management interface of system backstage.

After successful login, administrators management interface is showed as figure 15:



Figure15 Main management interface of system backstage

I designed simple frameworks for main management interface of system backstage. It combines left function menu and right main operating area, and function menu contains modules (only framework without content) of Book Management, Transaction Management, Feedbacks Management and User Management for managing further design system.

# 8 Testing

## 8.1 Functional Test

## 8.2 Usability Test

# 9 Conclusion

## 9.1 Technical Complexity

## 9.2 Innovation

## 9.3 Future Enhancements

# 10 Appendices

## 10.1 Reference Guide