

分类号: TP312

U D C: D10621-408-(2015)2426-0

密 级: 公 开

编 号: 2012124052

# 成 都 信 息 工 程 大 学

## 学 位 论 文

个人学习云记录与展示平台的设计与实现

论文作者姓名: 姜宇

申请学位专业: 物联网工程

申请学位类别: 工学学士

指导教师姓名(职称): 王娟(副教授)

论文提交日期: 2016 年 06 月 12 日

# 个人学习云记录与展示平台的设计与实现

## 摘 要

云计算、云存储、云服务伴随着网络技术的飞速发展，越来越多的用户开始体会到了他们对于不同地点不同终端之间的数据同步服务的强烈需求，云记录平台应运而生。个人学习云记录是其中一种展示自我学习方式，和交流的平台。用户通过平台注册信息，登陆后拥有自己的储存空间，可在发表自己的学习心得，上传需要记录与收藏代码，使用时可添加或通过关键字查询笔记，对记录进行修改删除等操作，随时随地使用网络温习或查询相关记录，提供了一个安全方便的学习平台。

本系统采用 B/S 开发模式，分为服务器端和浏览器端。在服务器端采用 MySQL 数据库使用 MyBatis 技术访问数据库。使用 Spring 框架管理组件，整合 MyBatis，处理一些通用的业务。使用 MVC 分层思想，将代码分为控制层，业务层，持久层和视图层，实现了登陆模块，笔记本模块，和笔记模块。登陆模块的主要包括登陆，注册，修改密码和退出登陆功能。

**关键词：**云记录；学习平台；MVC 模式；MySQL

# **The Design and Implementation of Personal Learning Cloud Record and Display Platform**

## **Abstract**

Cloud computing, cloud storage, cloud services along with the rapid development of network technology, more and more users begin to experience strong demand for their data among different terminals in different locations of synchronization services. Cloud platforms came into meeting this demand. Record Personal Learning cloud is one way to self-learning and communication platform. User registration in and login and he will have his own storage space, can express their learning experience, recording and uploading collections of code. And also can add notes or search by keyword, to modify or delete the records at anytime, anywhere through network for study or queries related records, provides a safe and convenient learning platform.

The system uses B / S development model which is divided into server and browser. On the server side using the MySQL database and using MyBatis technology to access the database. Using the Spring framework for managing components, integrated MyBatis, to deal with some common business. Use MVC hierarchical thinking in which the code is divided into the control layer, business layer, persistence layer and view layer to complete a login module, notebook modules, and notes module. The main login module including login, registration, change password and logout functionality.

**Key words:** cloud records; learning platform; MVC pattern; MySQL

# 目 录

论文总页数：27 页

1 引言.....	1
1.1 课题背景.....	1
1.2 国内外研究现状.....	1
1.3 本课题研究的意义.....	2
1.4 工作总结与论文内容安排.....	2
2 相关技术简介.....	3
2.1 Java 语言概述.....	3
2.2 B/S 和 C/S 体系.....	3
2.3 JavaScript 与 Ajax.....	3
2.5 MVC 设计模式.....	4
2.6 开发工具 MyEclipse.....	4
3 系统需求分析与系统设计.....	4
3.1 系统需求分析.....	4
3.2 系统设计.....	5
3.2.1 系统架构.....	5
3.2.2 系统模块划分.....	5
3.2.3 系统流程设计.....	6
3.2.4 系统数据库设计.....	7
3.2.5 系统 UI 设计.....	9
4 系统实现.....	9
4.1 登陆模块实现.....	9
4.1.1 注册功能.....	9
4.1.2 登录功能.....	10
4.1.3 退出登录.....	12
4.1.4 修改密码.....	13
4.2 笔记本模块实现.....	14
4.3 笔记模块实现.....	14
4.3.1 笔记回收站实现.....	15
4.3.2 分享以及收藏笔记实现.....	18
4.4 统计功能实现.....	19
4.4.1 笔记统计实现.....	19
4.4.2 文件统计实现.....	20

4.5 系统功能小结.....	21
5 系统的展示与测试.....	21
5.1 测试环境.....	21
5.2 系统测试.....	21
5.2.1 注册、登录、修改密码测试.....	21
5.2.2 笔记本界面.....	22
5.2.3 笔记界面.....	23
5.2.4 统计界面.....	24
5.3 测试小结.....	24
结 语.....	24
参考文献.....	25
致 谢.....	26
声 明.....	27

# 1 引言

## 1.1 课题背景

因为互联网的快速发展，人们获得新知识的方法也变得多样化，在线学习应运而生，为了让使用在线学习的人能更方便的记录笔记，云笔记项目因此诞生。云笔记使得知识不能随时随地的记录。云笔记系统贴近用户的使用习惯，期望达到无处不学习无处不笔记的效果，在不远的将来，云笔记必将推向大众，成为人们日常生活不可或缺的一部分。

云笔记主要使用云储存架构，将信息存于网络。云时代的存储系统需要的不仅仅是容量的提升，对于性能的要求同样迫切，与以往只面向有限的用户不同，在云时代，存储系统将面向更为广阔的用户群体，用户数量级的增加使得存储系统也必须在吞吐性能上有飞速的提升，只有这样才能对请求作出快速的反应，这就要求存储系统能够随着容量的增加而拥有线性增长的吞吐性能，这显然是传统的存储架构无法达成的目标。传统的存储系统由于没有采用分布式的文件系统，无法将所有访问压力平均分配到多个存储节点，因而在存储系统与计算系统之间存在着明显的传输瓶颈，由此而带来单点故障等多种后续问题，而云存储正是解决这一问题，满足新时代要求的一剂良药，所以就有了以云储存架构为基础的云笔记的诞生。

## 1.2 国内外研究现状

按照云存储后台以及多种记录方式这两条特征来看，国内类似于 Evernote 的云笔记产品约将近十款，包括有道云笔记、麦库记事、FIT 云笔记、为知笔记、海知笔记、轻笔记等等。Path、Instagram 流行之时，国内同类产品达到二三十款，与其相比，目前云笔记市场的竞争尚未大规模展开。

盛大创新院开发的麦库记事是国内最早的云笔记，依靠先发优势，加上盛大云、搜索、多媒体等技术支持，已经积累了 500 万用户，成为目前用户最多的云笔记产品。

有道云笔记的云存储架构有自身的需求和特点。云存储架构基于自有的底层云平台，实现了高可靠性、高可用性、高扩展性的云端服务。云架构满足了移动互联网时代的新需求，支持多终端、多设备的实时同步需求。针对云笔记，我们的云存储架构有几个特点：包括异构的存储、增量备份、对多版本的原生支持、灵活的分享机制等。

轻笔记利用云存储技术，实现多平台数据同步，通过电脑、手机、平板随时访问和编辑记事本内容。

为知笔记的文档存储，是采用 zip 方式压缩的 html 文件，可以使用任意的解压缩软件打开，例如 WizZip, WinRAR 或者 7z。为知笔记提供了文档加密的

功能，使用加密功能后，为知笔记会把存储的 zip 数据采用 RSA+AES 方式重新加密，从而保证数据内容的安全。是国内目前来说安全性非常高的云笔记产品之一。

### 1.3 本课题研究的意义

随着人们生活水平的提高，人们对于各时各地记录生活也需求越来越大。伴随着云技术的提升，随时随地学习与分享心得是必要的发展途径，生活中的学习，社交都是本系统所关注的方面。

通过对该系统的设计，让我加深了对 MVC 开发模式以及 java,jsp 等相关技术的理解。进一步熟悉数据库管理系统的操作技术。以及懂得了系统设计需要掌握的大局观以及细节，明白查阅资料的重要性。

### 1.4 工作总结与论文内容安排

主要工作小结如下：

- 1.查阅相关云笔记的国内外资料，调研了当前学习软件以及云储存的应响个，发展现状，确定了云笔记学习平台的大体思路以及所用技术等。
- 2.分析现有资源，设计云笔记平台的基本结构。
- 3.搭建框架，设计数据流程，增加系统基础功能。
- 4.实现本系统的所有功能。
- 5.完成对系统的测试。
- 6.撰写论文。

本论文共分为 5 章节，详细安排如下：

第一章为绪论，概述课题背景、通过网上的资料查询云笔记当下国内外的研究情况、本课题的研究意义、主要的研究内容以及论文结构安排。

第二章为系统理论和技术路线，通过对 java, jsp 的书籍查阅，主要对系统的所用的技术路线，系统框架的技术做大致的描述，为云笔记平台设计打下基础。

第三章为系统设计分析，主要论述了云笔记系统的总体设计框图，所需要的 mysql 数据表，以及表的结构等。

第四章为系统详细设计，系统的设计出发，阐述模块功能的实现，以及给出了整体系统测试的演示结果。

第五章为系统测试，从用户角度出发，展示系统使用界面以及测试是否通过。

结语部分是对本文的总结以及工作展望。总结了在本课题期间完成的所有工作以及自我收获。

## 2 相关技术简介

### 2.1 Java 语言概述

Java 是 SUN 公司开发最初命名为 Oak 是一种小型系统的编程语言，随着互联网发展 SUN 公司改造了 Oak 并以“Java”名称正式发布。Java 编程风格接近于 C、C++ 语言，但不同的是 Java 是完全的面向对象程序设计语言，继承了 C++ 语言面向对象技术的核心但是舍弃了指针，增加了垃圾回收机制。

JDK（Java Development Kit）是 Java 开发工具包，除了包含 JRE（Java Runtime Environment）外还包含运行 Java 程序所需要的工具 JDK。所以 JDK=JRE+编译、运行等开发工具。JRE:Java 运行时环境除了包含 JVM（Java Virtual Machine）外还提供了运行 Java 程序所需要的环境。JRE=JVM+Java 系统类库 JVM 是 Java 虚拟机。

Java 开发过程简而言之就是首先使用一个文本编辑器编写扩展名为 .Java 的源文件，然后用 Java 编译器把扩展名为 .Java 的源文件编译为 .class，最后 JVM 加载 .class 并运行 .class。不同的平台使用不同的 JVM 所以，JVM 屏蔽了平台之间的差异，所以实现了跨平台一次编译到处使用。

### 2.2 B/S 和 C/S 体系

B/S 即浏览器/服务器端结构，如果说 B/S 就最好和 C/S 作比较，那具体的项目举例就是网页版的 qq 是 B/S 结构，而需要下载的 qq 客户端属于 C/S 模式。

从安全性来说，C/S 模式对于每个用户都有一个客户端，而 B/S 模式的客户端是被所有用户公用，而且不固定，所以说，C/S 比较 B/S 来说安全性相对较高。

从维护成本角度来讲 C/S 结构中需要用户下载专有的客户端软件，而 B/S 模式，客户端不需要下载任何用户界面，软件系统的改进和升级越来越频繁，B/S 架构的产品明显体现着更为方便的特性。B/S 架构的软件只需要管理服务器就行了，所有的客户端只是浏览器，根本不需要做任何维护。所以客户机越来越“瘦”，而服务器越来越“胖”是将来信息化发展的主流方向。

今后，软件升级和维护会越来越容易，而使用起来会越来越简单，这对用户人力、物力、时间、费用的节省是显而易见的，惊人的。从这一方面来说节约了开发成本。

### 2.3 JavaScript 与 Ajax

JavaScript 是 Netscape 开发的一种在客户端运行的、解释性的脚本语言，最早是在 HTML 网页上使用，用来为网页增加动态功能。为了统一规格，JavaScript 兼容于 ECMA（European Computer Manufacturers Association）标准，因此也称为 ECMAScript。JavaScript 的关键字 var、if、for、switch、break、continue、



while 支持数据类型的自动转换和强制转换。

Ajax (Asynchronous JavaScript and Xml) 是异步的 JavaScript 和 Xml, 是一种用来改善用户体验的技术, 它的实质是利用浏览器提供的一个特殊对象 (Ajax 对象) 异步的向服务器发送请求, 服务器送回部分数据 (一般不再需要返回完整的页面), 浏览器利用这些数据更新当前页面。整个过程, 页面无需刷新, 不中断用户的操作获得 Ajax 对象要区分浏览器。

## 2.5 MVC 设计模式

MVC 一种设计思想, 是根据职责不同将程序中组件分成以下三个部分: M(Model) 模型负责业务处理 (数据和逻辑), V(View) 视图负责与用户的交互界面 (可以接收和呈现数据), C(Controller) 控制器负责协调 M 和 V 工作。MVC 思想现在有很多实现, 可以自己使用 JSP+Servlet+JavaBean 搭建, 也可以使用主流框架, 例如 Spring Web MVC, Struts2, WebWork, JSF 等。

## 2.6 开发工具 MyEclipse

如果想要了解 MyEclipse, 首先应该先了解 Eclipse, Eclipse 是基于 Java 语言开源的并且可以扩展的操作平台。Eclipse 内置了一个标准的插件库, 包括 Java Development Tools 即 JDT, Eclipse 首先来说源码是开放的, 程序员都可以研究使用修改它, 另外, Eclipse 是可以扩展的, 它不仅支持 Java 的开发, 还包括 PHP、C 等很多语言的开发和使用, 只需要下载相应的插件即可。而且很多平台如 Linux, windows 等都支持 Eclipse。MyEclipse 是 Eclipse 的加强版, 它不仅继承了 Eclipse 的所有功能, 而且能很好的处理在 JavaEE 和数据库的开发, 因而我们能利用 JavaEE 中数据库和 servlet 来实现我们软件中后台的一些功能。

# 3 系统需求分析与系统设计

## 3.1 系统需求分析

本系统需要实现用户的学习方便化, 主要模块为笔记本模块, 用户创建笔记本添加笔记, 并能够实现分享以及加入回收站功能, 可上传文件, 并进行学习进度统计。已被分享的笔记可由其他账户搜索得到, 并且实现针对笔记的学习计划定制。

基于上述需求, 本系统必须具备以下功能:

1. 用户的注册与登录。
2. 用户笔记本的添加。
3. 笔记本中笔记添加内容的增加和修改。
4. 分享笔记。
5. 查询被分享的笔记。
6. 上传文件实现统计与下载。

- 7.修改账号信息。
- 8.统计学习进度。
- 9.退出登录。

## 3.2 系统设计

### 3.2.1 系统架构

本系统的主要开发工具是 MyEclipse。

在数据库方面，我所设计的云笔记系统用到的是 MySQL 数据库，大致包括用户表，笔记本类型表，笔记本表，笔记表，笔记状态表，分享笔记表。

对于浏览器器端的设计，主要用 HTML、CSS 来实现，其中对于数据库的请求和获取数据库的信息并显示，主要会用到 Ajax。服务端代码主要采用 Spring MVC（Model View Controller）来实现的。

#### 1. 硬件环境

服务端：有足够大的内存可以运行数据库，MyEclipse 和 Chrome 浏览器。

#### 2. 软件环境

- （1）浏览器端：Chrome 浏览器。
- （2）服务端：MySQL 数据库、MyEclipse 。

### 3.2.2 系统模块划分

此系统设计主要由以下各功能组成，包括用户的登陆、用户对笔记以及笔记本信息的管理、用户退出登录及修改密码。系统所包含的模块如图 3-1 所示：

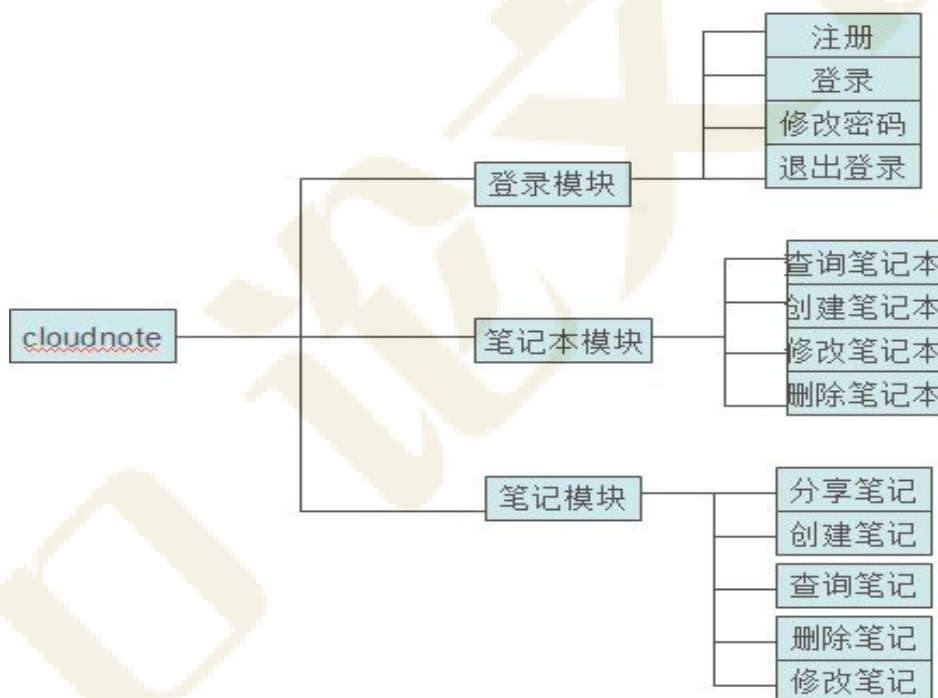


图 3-1 系统模块图

### 3.2.3 系统流程设计

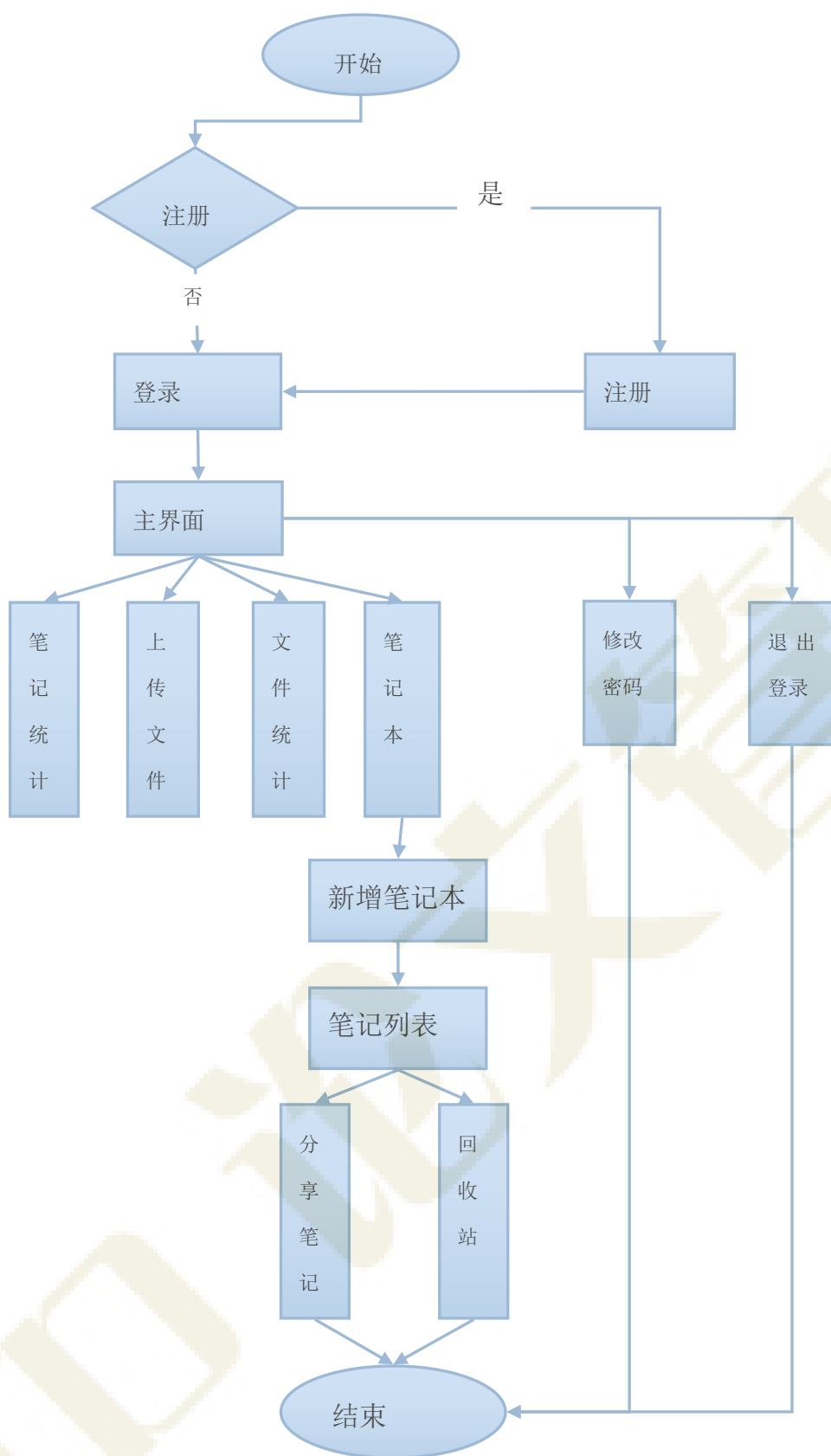


图 3-2 系统流程图

### 3.2.4 系统数据库设计

数据库我们采用 MySQL 数据库，所需要建立的表有：

(1) 用户表 (cn\_user) 存储登录系统的用户信息。其中包括主键用户 ID (cn\_user\_id)、用户名 (cn\_user\_name)、密码 (cn\_user\_password)、说明 (cn\_user\_desc)。

表 3-1 user 表

字段	类型	意义	备注
cn_user_id	varchar(80)	用户 ID	主键 (PK)
cn_user_name	varchar(20)	用户名	不为空
cn_user_password	varchar(40)	密码	不为空
cn_user_desc	Varchar(20)	说明	不为空

(2) 笔记本表 (cn\_notebook) 笔记存储在笔记本中，笔记本存储在笔记本表中。每个用户都可以有很多不同的笔记本，分别存储不同类别的笔记。笔记本表中包括主键笔记本 ID (cn\_notebook\_id)、用户 ID (cn\_user\_id)、笔记本类型 ID (cn\_notebook\_id)、笔记本名 (cn\_notebook\_name)、笔记本说明 (cn\_notebook\_desc)、创建日期 (cn\_notebook\_createtime)。

表 3-2 笔记本表

字段	类型	意义	备注
cn_notebook_id	varchar(80)	笔记本 ID	主键 (PK)
cn_user_id	varchar(20)	用户 ID	不为空
cn_notebook_id	varchar(40)	记本类型 ID	不为空
cn_notebook_name	varchar(40)	笔记本名	不为空
cn_notebook_desc	Varchar(20)	笔记本说明	不为空
cn_notebook_createtime	Varchar(20)	创建日期	不为空

(3) 文件表 (cn\_notebook\_type) 用户可上传文件，文件将在统计中显示，提供下载功能，包括 ID 主键，cn\_user\_id 用户主键，data 文件数据储存, name 文件名, info 备注信息, startTime 上传时间, updateTime 更新时间。

表 3-3 笔记本类型表

字段	类型	意义	备注
ID	varchar(80)	主键	主键 (PK)
data	longblob	文件数据储存区	不为空
name	varchar(40)	文件名	不为空
info	varchar(40)	备注信息	不为空
startTime	varchar(40)	上传时间	不为空
updateTime	varchar(40)	更新时间	不为空

(4) 笔记表 (cn\_note\_id) 是用来存储用户创建的所有笔记, 在保存时需要存储用户的 ID 还有当前笔记所在的笔记本的 ID。笔记表中包括主键笔记 ID (cn\_note\_id)、笔记本 ID (cn\_notebook\_id)、用户 ID (cn\_user\_id)、笔记标题 (cn\_note\_title)、笔记内容 (cn\_note\_body)、笔记创建时间 (cn\_note\_createtime)。

表 3-4 笔记表

字段	类型	意义	备注
cn_note_id	varchar(80)	笔记 ID	主键 (PK)
cn_notebook_id	varchar(20)	笔记本 ID	不为空
cn_user_id	varchar(40)	用户 ID	不为空
cn_note_title	varchar(40)	笔记名	不为空
cn_note_body	Varchar(20)	笔记内容	不为空
cn_note_createtime	Varchar(20)	创建日期	不为空
cn_note_status_id	Varchar(20)	笔记状态	无

(5) 分享笔记表 (cn\_share) 用户对于自己的笔记可以进行分享, 分享后的笔记存储到此分享笔记表中。其中包括主键共享 ID (cn\_share\_id)、共享标题 (cn\_share\_title)、共享内容 (cn\_share\_body)、笔记 id (cn\_note\_id)。

表 3-5 分享笔记表

字段	类型	意义	备注
cn_share_id	varchar(80)	主键共享 ID	主键 (PK)
cn_share_title	varchar(20)	共享标题	不为空
cn_share_body	varchar(40)	共享内容	不为空
cn_note_id	varchar(40)	笔记 id	不为空

### 3.2.5 系统 UI 设计

图 3-3 为本学习平台系统的主要模块简图设计。该模块为笔记本模块，依据此图将开展系统的正式设计。

欢迎进入学习乐园		
笔记统计	文件上传	文件统计
我的笔记本	学习计划	修改密码
退出登录		
全部笔记本 <input type="button" value="新增"/>	全部笔记 <input type="button" value="新增"/>	编辑笔记
笔记本1 笔记本2 笔记本3	笔记本1 笔记1 笔记2 笔记3	笔记1 笔记1的内容 <input type="button" value="保存"/>

图 3-3 笔记模块 UI 简图

## 4 系统实现

### 4.1 登陆模块实现

#### 4.1.1 注册功能

开发步骤：首先在 DAO 中创建表 `cn_user`，创建实体类 `User`，创建 `UserMapper` 创建 `UserMapper.xml`。然后在 Service 中创建 `LoginService`，验证用户名是否重复，创建用户。最后在 Controller 中创建 `LoginController`，并且调用 `LoginService` 处理请求并且给页面返回一些数据，该数据在控制层统一组装，最好所有 Controller 的方法返回的数据都有标准的格式。主要代码如下：

UserMapper.xml:

//mabities 的 mapper 配置文件，直接生成方法以供调用

```
<insert id="save"
      parameterType="com.jy.entity.User">
    insert into cn_user values(
        #{cn_user_id},
        #{cn_user_name},
        #{cn_user_password},
        #{cn_user_token},
        #{cn_user_desc}
    )
</insert>
```

LoginService:

创建用户并校验用户名，校验通过则创建用户，给用户初始化笔记本

```
public boolean createUser(User user) {  
    if(this.checkUserName(user.getCn_user_name())) {  
        this.addUser(user);  
        this.initNoteBook(user.getCn_user_id());  
        return true;  
    } else {  
        //校验失败  
        return false;} } }
```

校验用户名，如果用户名为空则返回提示“用户名不能为空”

```
public boolean checkUserName(String userName) {  
    if (userName == null)  
        throw new BusinessException("用户名不能为空.");  
    User user = userMapper.findByName(userName);  
    if (user == null)  
        return true;  
    else  
        return false; } }
```

初始化笔记本，通过用户的 Id 查找用户所有的笔记本，返回给页面显示笔记本列表，如果用户名为空则显示“参数为空”。

```
public void initNoteBook(String userId) {  
    if(userId == null)  
        throw new BusinessException("参数为空.");  
}
```

LoginController:

传一个 User 实体，调用 serService 里创建方法，将用户信息存入数据库中，返回一个 result 的 json 字符串返回给页面做操做。

```
public Result register(User user) {  
    boolean b = userService.createUser(user);  
    return new Result(b); } }
```

#### 4.1.2 登录功能

开发步骤：首先在 DAO 中的 UserMapper 创建 findByName 方法，然后在 Service 的 LoginService 里增加验证用户名和密码的方法，该方法的逻辑为：根据用户名查询用户，判断查询结果是否为 null，如果查询结果为 null 说明用户名不存在返回错误信息，如果查询结果不为 null，继续判断密码是否正确，密

码正确则登录成功，密码错误则返回错误信息。最后我们在 Controller 中的 LoginController 中调用 LoginService 对用户名、密码进行验证，如果验证成功，将用户信息存入 session。主要代码如下：

UserMapper.xml:

使用 mapper 里的 dao 方法，通过 userName 查找用户名。

```
<select id="findByName"
parameterType="string"
// resultType 是指返回值类型
resultType="com.jy.entity.User">
select * from cn_user where cn_user_name=#{userName}
</select>
```

LoginService:

校验用户名及密码，以及做相关提示，将页面输入的用户名和密码通过 checkUser 方法判断信息是否匹配，用户名和密码非空提示，未输入则在页面做提示。相应的不匹配则在页面做错误提示。

```
public Map<String, Object> checkUser(String userName,
String pwd) {
Map<String, Object> map = new HashMap<String, Object>();
```

判断用户名密码是否匹配，并进行非空判断，如果是空抛出错误提示，匹配成功则提示“登陆成功”。

```
if(userName == null)
throw new BusinessException("用户名为空.");
if(pwd == null)
throw new BusinessException("密码为空.");
User user = userMapper.findByName(userName);
    if(user == null) {
map.put("flag", SystemConstant.LOGIN_PASSWORD_ERROR);
map.put("msg", "用户名错误.");
    }
else if (!user.getCn_user_password().equals(Md5Util.md5(pwd))) {
map.put("flag", SystemConstant.LOGIN_PASSWORD_ERROR);
map.put("msg", "密码错误.");
} else {
map.put("flag", SystemConstant.LOGIN_SUCCESS);
```



```
map.put("msg", "登录成功.");    }return map; }
```

LoginController:

登录校验，成功则跳转进入主界面，调用 service 里的方法，判断数据库信息是否匹配，成功则将信息存入 session，最后返回 result 实体。

```
@RequestMapping("/login.do")
@ResponseBody
public Result login(String userName,
    String password, HttpSession session) {
    Map<String, Object> data =
        userService.checkUser(userName, password);
    if ("0".equals(data.get("flag").toString())) {
        //登录成功，将用户信息保存如 session
        User user = userService.findUser(userName);
        data.put("user", user);
        session.setAttribute("user", user);    }
    return new Result(data); }
```

#### 4. 1. 3 退出登录

开发步骤：Controller：在 LoginController 中增加一个退出登录的方法在该方法中将 session 注销。主要代码如下：

LoginController:

```
@RequestMapping("/login.do")
@ResponseBody
```

调用 service 方法判断，用户输入的用户名和密码是否匹配，resuly 里储存为 0 则是匹配成功，成功后将用户信息存入 session 中。

```
public Result login(String userName,
    String password, HttpSession session) {
    Map<String, Object> data =
        userService.checkUser(userName, password);
    if ("0".equals(data.get("flag").toString())) {
        //登录成功
        User user = userService.findUser(userName);
        data.put("user", user);
        session.setAttribute("user", user);
    }
```

```
        return new Result(data);
    }
}
```

#### 4.1.4 修改密码

开发步骤：首先在 DAO 中的 UserMapper 中增加修改密码的方法，然后在 Service 中增加修改密码的业务需要先校验原密码再修改密码。最后在 Controller 中增加修改密码的方法该方法要接收页面传入的原密码和新密码，调用 Service 进行修改密码。主要代码如下：

UserMapper.xml:

使用 mapper 里的 dao 方法，通过 userid 查找该用户信息，通过传参确定需要更新的信息进行更新操作。

```
<update id="update"
        parameterType="com.jy.entity.User">
    update cn_user set
        cn_user_name=#{cn_user_name},
        cn_user_password=#{cn_user_password},
        cn_user_token=#{cn_user_token},
        cn_user_desc=#{cn_user_desc}
    where cn_user_id=#{cn_user_id}
</update>
```

LoginService:

service 里通过 controller 模块中传来的需要修改的实体和信息，进行更新用户信息操作。

```
public void update(User user) {
    userMapper.update(user);
}
```

LoginController:

```
@RequestMapping("/changePassword.do")
@ResponseBody
```

修改用户密码，通过 Md5 加解密操作，原密码解密后做判断，和数据库一样则更新为新密码并做加密操作，原密码错误则做错误提示，更新成功后提示“更新成功”，调用的是 service 里的更新方法。

```
public Result changePassword(String lastPassword,
    String newPassword, HttpSession session) {
    User user = (User) session.getAttribute("user");
```

```

        if(user.getCn_user_password()
            .equals(Md5Util.md5(lastPassword))) {
            user.setCn_user_password(Md5Util.md5(newPassword));
            userService.update(user);
            return new Result("修改成功.");
        } else {
            return new Result("原密码输入有误.");
        }
    }
}

```

## 4.2 笔记本模块实现

开发步骤：首先在 DAO 代码中增加根据用户 ID 查询普通笔记本的方法。然后在 Service 中增加查询普通笔记本的方法。最后在 Controller 中增加方法处理查询普通笔记本的请求。主要代码如下：

NoteBookMapper.xml:

```

<select id="findNormalNoteBook"
        parameterType="string"
        resultType="com.jy.entity.NoteBook">
    select * from cn_notebook
    where cn_user_id=#{userId}
    and cn_notebook_type_id in (
        select cn_notebook_type_id from cn_notebook_type
        where cn_notebook_type_code
        in ('normal')
    )
    order by cn_notebook_createtime desc </select>

```

NoteBookService:

查询出笔记本名显示在笔记本列表中，将 user 存入 session 中，通过 json 方式将 list 存入 result 实体中，传回页面做页面操作。

```

public Result findNormal(HttpSession session) {
    User user = (User) session.getAttribute("user");
    List<NoteBook> list =
        noteBookService.findNormalNoteBook(user.getCn_user_id());
    return new Result(list);
}

```

## 4.3 笔记模块实现

查询笔记模块的开发步骤是：首先在 DAO 中增加根据笔记本 ID 查询出对应的笔记的方法。然后在 Service 中增加查询笔记方法。最后在 Controller 中增加一个查询方法，处理查询请求。主要代码如下：

NoteMapper.xml:

mppper 里的 dao 方法以供调用，通过 findById 方法，通过笔记本 id 查找数据库中对应的笔记本信息。

```
<select id="findById"
        parameterType="string"
        resultType="com.jy.entity.Note">
    select * from cn_note
    where cn_note_id=#{noteId}
</select>
```

NoteService:

查寻用户的笔记，调用 dao 方法，查询出笔记本信息，如果 Id 为空则抛出错误，提示“参数为空”。

```
public List<Note> findNote(String noteBookId) {
    if (noteBookId == null)
        throw new BusinessException("参数为空.");
    return noteMapper.findByNoteBook(noteBookId);
}
```

NoteController:

在 Controller 里调用 service 方法，将实体 result 传回页面做页面操作。

```
@RequestMapping("/findNote.do")
@ResponseBody
public Result findNote(String noteBookId) {
    List<Note> list = noteService.findNote(noteBookId);
    return new Result(list);
}
```

#### 4.3.1 笔记回收站实现

笔记回收站模块的开发步骤是：每一个笔记都会有一个状态 cn\_note\_status\_id, 首先在 DAO 中增加根据笔记 ID 更新笔记状态的 updateNoteStatus 方法。然后在 Service 中增加删除笔记方法，其实就是做了状态的更新，点击回收站时调用了 findRollBack 方法，查询出状态为“2”的也就是回收站中的笔记本，此时如果点击返回按钮，则该笔记状态变为“1”，也就是被返回原笔记本，点击删除则彻底删除。最后在 Controller 中增加一个查询方法，处理查询请求。主要代码如下：

NoteMapper.xml:

配置文件中的代码，实现更新笔记状态和查询被放入回收站的笔记的 Dao 方法。

更新笔记本状态，当点击删除按钮后实现

```
<update id="updateNoteStatus"
parameterType="map">
updatecn_note
set cn_note_status_id=#{noteStatus} where cn_note_id=#{noteId}
</update>
```

查询状态为“2”的笔记本，点击回收站时实现，也就是已被删除的。

```
<select id="findRollBack"
resultType="com.jy.entity.Note" parameterType="string">
select *
from cn_note where cn_note_status_id='2' and cn_user_id=#{userId}
</select>
```

点击返回按钮，则将回收站中的该笔记返回原笔记本，也就是改变该笔记状态为“1”

```
<update id="recoverRollBack"
parameterType="map">
update cn_note
set cn_notebook_id=#{noteBookId},cn_note_status_id='1'
where cn_note_id=#{noteId}
</update>
```

//在回收站中点击删除按钮，则做彻底删除而不是状态改变

```
<delete id="deleteRollBack" parameterType="string">
delete
from cn_note where
cn_note_id=#{noteId}
</delete>
```

NoteService:

service 类中 updateNoteStatus 方法，将所点击的 noteId 传入 Dao 对应的方法中，将该笔记状态设为“2”，创建 result 对象，将提示信息返回给 Controller 与页面做交互。

```
public NoteResult updateNoteStatus(String noteId) {
    NoteResult result=new NoteResult();
```

```

        Note note=new Note();
        note.setCn_note_status_id("2");
        note.setCn_note_id(noteId);
        noteDao.dynamicUpdate(note);
        result.setStatus(0);
        result.setMsg("已放入回收站");
        return result;
    }

```

service 类中 recoverRollBack 方法，将所点击的 noteId, noteBookId 传入 Dao 对应的方法中，查询出点击恢复的笔记本，调用 dao 的更新方法，更新笔记本状态为“1”，成功则提示“恢复成功”，创建 result 对象，将提示信息返回给 Controller 与页面做交互。

```

public NoteResult recoverRollBack(String noteId, String noteBookId) {
    NoteResult result=new NoteResult();
    Note note=noteDao.findByNoteId(noteId);
    Map<String,String> map=new HashMap<String,String>();
    map.put("noteBookId", noteBookId);
    map.put("noteId",noteId);
    noteDao.recoverRollBack(map);
    result.setStatus(0);
    result.setMsg("恢复成功");
    return result;
}

```

service 类中 deleteRollBack 方法，将所点击的 noteId，传入 Dao 对应的方法中，回收站中点击删除按钮，调用 dao 的删除，则删掉数据库中笔记本信息，成功则提示“删除成功”，将提示信息返回给 Controller 与页面做交互。

```

public NoteResult deleteRollBack(String noteId) {
    NoteResult result=new NoteResult();
    noteDao.deleteRollBack(noteId);
    result.setStatus(0);
    result.setMsg("删除成功");
    return result;
}

```

#### 4.3.2 分享以及收藏笔记实现

收藏笔记的开发步骤是：首先用户点击分享笔记，在 DAO 中增加根据笔记 Id 查询出对应的笔记的方法，然后将此笔记的状态设置为“3”，用户在搜索界面查询出状态为“3”的笔记，也就是已被分享的笔记，这种状态的笔记是任何用户都可查询到的。然后在 Service 中增加收藏笔记方法，也就是点击收藏之后，新建一个笔记本，信息和被点击收藏的一样，保存在该用户收藏夹下，就完成了收藏笔记。主要代码如下：

service 类中 shareNote 方法，将所点击的 noteId 传入 Dao 对应的 findByNoteId 方法中查询。在 share 查询出对应的笔记，若没有查到则将 noteTitle, noteBody 新增加入 share 表中，提示“分享成功”。

```
public NoteResult shareNote(String noteId, String noteTitle, String noteBody) {  
    NoteResult result=new NoteResult();  
    ShareNote share=new ShareNote();  
    //调用 Dao 方法查询笔记信息  
    ShareNote shareNote=shareNoteDao.findByNoteId(noteId);  
    if(shareNote!=null){  
        result.setStatus(1);  
        result.setMsg("该笔记已被分享");  
        return result;  
    }  
    //没查到提示“分享成功”，保存笔记信息到 share 表。  
    share.setCn_share_id(NoteUtil.createId());  
    share.setCn_note_id(noteId);  
    share.setCn_share_body(noteBody);  
    share.setCn_share_title(noteTitle);  
    shareNoteDao.save(share);  
    result.setStatus(0);  
    result.setMsg("分享成功");  
    return result;  
}
```

service 类中 collectNote 方法，将所点击的 noteId 传入 shareNoteDao 对应的 findByShareId 方法中查询。点击收藏按钮后将调用 notedDao 里的储存方法，将该分享笔记变为自己的笔记，并设置该状态为“3”，使其在已收藏笔记列表中出现。

```

public NoteResult collectNote(String shareId,String userId) {
    NoteResult result=new NoteResult();
    ShareNote share=shareNoteDao.findByShareId(shareId);
    Note note=new Note();
    String noteId=share.getCn_note_id();
    Map<String,String> map=new HashMap<String,String>();
    map.put("noteId", noteId);
    map.put("userId", userId);

```

查询笔记是否是自己分享出去的笔记，若是则做提示“该笔记为您自己的笔记，不用收藏”

```

    Note note1=noteDao.findByNoteIdAndUserId(map);
    if(note1!=null){
        result.setStatus(1);
        result.setMsg("该笔记为您自己的笔记，不用收藏");
        return result;
    }

```

点击收藏收新建一个笔记实例，将要收藏的笔记信息存入笔记实例中，并设置状态为“3”，则收藏成功，提示“笔记收藏成功”。

```

    String noteTitle=share.getCn_share_title();
    String noteBody=share.getCn_share_body();
    note.setCn_note_id(NoteUtil.createId());
    note.setCn_note_body(noteBody);
    note.setCn_note_title(noteTitle);
    note.setCn_note_status_id("3");
    note.setCn_user_id(userId);
    noteDao.save(note);
    result.setStatus(0);
    result.setMsg("笔记收藏成功");
    return result;
}

```

## 4.4 统计功能实现

### 4.4.1 笔记统计实现

笔记统计的开发步骤是：在配置文件中增加查询方法，通过 `cn_user_id` 查询出该用户的所有笔记，点击统计界面时调用该方法，将其显示在 `jsp` 页面中，



通过 jquery 拼接成表格统计。

Sql 语句如下：

```
<select  
id="findDetailNote" parameterType="string" resultType="com.jy.entity.NoteOrderBook">  
select*  
from cn_note,cn_notebook where cn_note.cn_notebook_id  
=cn_notebook.cn_notebook_id and cn_note.cn_user_id = #{userId} and  
cn_note.cn_note_status_id = '1'  
</select>
```

NoteService 中的代码：

通过 userId 查询所有用户保存的笔记，并存入 list，新建 result 对象，传回 Controller 与页面交互，页面得到 list 后生成列表。

```
public NoteResult findDetailNote(String userId) {  
    NoteResult result=new NoteResult();  
    List<NoteOrderBook> list = noteDao.findDetailNote(userId);  
    result.setStatus(0);  
    result.setData(list);  
    return result;  
}
```

#### 4.4.2 文件统计实现

笔记统计的开发步骤是：在配置文件中增加查询方法，通过 cn\_user\_id 查询出该用户的上传过的所有文件，点击文件统计界面时调用该方法，将其显示在 jsp 页面中，通过 jquery 拼接成表格统计。

Sql 语句如下：

```
<select  
id="getFileDataById" resultType="com.jy.entity.FileData" parameterType="string">  
SELECT *  
FROM cn_file_data  
WHERE cn_user_id = #{cn_user_id}  
</select>
```

通过 userId 查询所有该用户上传的文件信息，并存入 list，新建 result 对象，传回 Controller 与页面交互，页面得到 list 后生成列表。

```
public NoteResult findFileList(String cn_user_id) {  
    NoteResult result=new NoteResult();
```

```
List<FileData> list = new ArrayList<FileData>();  
list = dao.getFileDataById(cn_user_id);  
result.setStatus(0);  
result.setData(list);  
return result;  
}
```

## 4.5 系统功能小结

本章主要介绍了云笔记平台的主要功能。

云笔记系统具有的功能有：注册、登录、修改密码、笔记本、笔记、回收站、收藏笔记本、统计功能。

用户使用云笔记时，需要先进行注册，注册成功后进入笔记本模块的四个基础笔记：收藏笔记本用来存放用户收藏的笔记，回收站笔记本用来暂存用户删除的笔记，在回收站中，用户可以恢复删除的笔记，也可以将笔记彻底删除。

## 5 系统的展示与测试

### 5.1 测试环境

- 1.硬件:PC 电脑 windows7(64 位)。
- 2.软件: myeclipse2014,mysql 数据库， chrome 浏览器。

### 5.2 系统测试

#### 5.2.1 注册、登录、修改密码测试

- 1、注册帐号，如已存在或不符合规范，则提醒注册失败，成功则返回登录界面。



The image shows a registration form titled '注册' (Register). It contains four input fields: '用户名:' (Username), '昵称:' (Nickname), '密码:' (Password), and '确认密码:' (Confirm Password). Below the fields are two buttons: '注册' (Register) and '返回' (Return).

图 5-1 注册界面

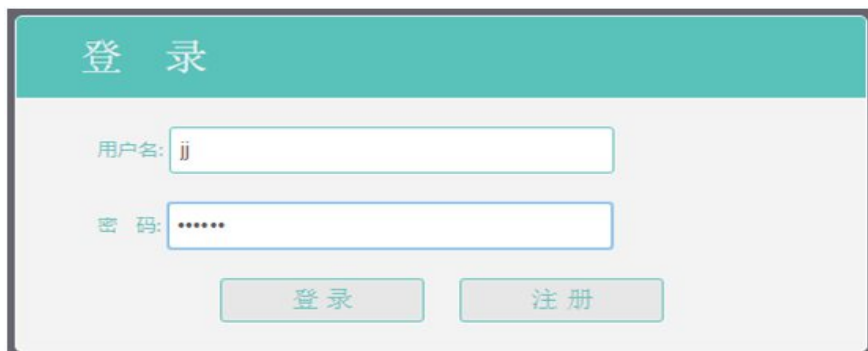


图 5-2 登录界面

2、输入注册的帐号，如错误提示登录失败，成功则跳入主界面。



图 5-3 主界面

3.修改密码，用户输入原密码，两次输入新密码，原密码无错则修改成功，原密码错误则动态提示“原密码错误”。

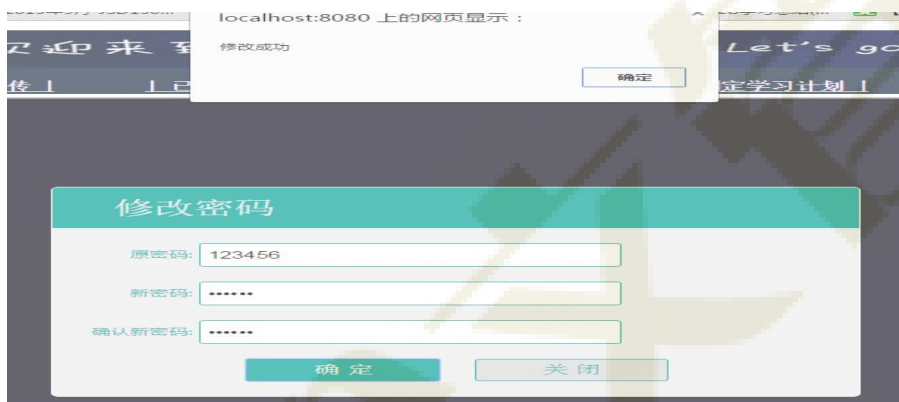


图 5-4 修改密码界面

## 5.2.2 笔记本界面

1.增加笔记本，并增加笔记名和内容并保存。



图 5-5 增加笔记本界面

### 5.2.3 笔记界面

1.用户选择创建好的笔记本，在此笔记本下添加笔记，如果未选择笔记本则添加笔记不成功，提示“未选择笔记本”。



图 5-6 笔记本中增加笔记界面

2.用户点击删除笔记默认放入回收站。



图 5-7 加入回收站界面

3.分享笔记以及收藏笔记，单机分享笔记，并用其他账户登录，可看到已被分享的他人笔记。



图 5-8 分享笔记页面

## 5.2.4 统计界面

1.笔记统计界面，看到创建过的笔记以及所在的笔记本，统计笔记本信息。



图 5-9 笔记统计页面

2.用户可上传学习文件，并在文件统计界面看到详情以及下载。



图 5-10 学习文件统计页面

## 5.3 测试小结

本次测试为流程功能测试，测试了本系统的基本流程功能和各模块的完整性。经测试，本系统所有目标功能都能正常实现。

## 结 语

笔记本模块主要包含新增，修改，删除功能。笔记模块包含查询，新增，修改，删除功能。在浏览器端主要采用 Ajax 处理客户端逻辑以及发送异步请求并实现了系统界面的一些动态效果。

虽然本系统基本功能已经实现，但是还有许多需要改进的地方，如：界面的美化，内存的使用率，选择更优的器材等。同时人机交互还不是非常友好，仍然需要进一步改进。

通过本次毕业设计，加深了对开发框架的理解，对理论与实际有了更深层次的理解，在解决问题方面，学会多重思考，将多种情况考虑周全，避免实现过程中出现问题。

## 参考文献

- [1] 王珊, 萨师煊. 《数据库系统概论》[M]. 北京: 高等教育出版社, 2010. 10
- [2] 明日科技. 《Java Web 从入门到精通》[M]. 北京: 清华大学出版社, 2013. 06
- [3] 张化祥, 陆晶. 《Java 程序设计》[M]. 北京: 清华大学出版社, 2010. 10
- [4] 何玉洁. 《数据库原理与应用教程》[M]. 北京: 机械工业出版社, 2010. 10
- [5] 施瓦茨, 扎伊采夫, 特卡琴科. 《高性能 MySQL》[M]. 北京: 电子工业出版社, 2013. 05
- [6] 刁仁宏. 网络数据库原理及应用[J]. 情报理论与实践, 2004, 8(1), 30-45
- [7] 张基温. 计算机专业课程改革[J]. 无锡教育学院学, 2003, 32(4): 54-55
- [8] 桑新民. 当代信息技术在传统文化教育中引发的革命[J]. 教育研究, 1997, 5(5): 17-23
- [9] 王保罗. Java 面向对象程序设计[M]. 北京: 清华大学出版社, 2003 年
- [10] 贺松平. 基于 MVC 模式的 B/S 架构的研究 [J]. 武汉: 华中科技大学, 2006, 22(4), 12-19
- [11] 耿祥义. JSP 基础编程[M]. 清华大学出版社, 2004

## 致 谢

本文是在王娟老师的热情关心和指导下完成的，他渊博的知识和严谨的治学态度使我受益匪浅，对顺利完成本课题起到了极大的作用。在此向他表示我最衷心的感谢！

在论文完成过程中，本人还得到了辅导员老师和本班同学的热心帮助，本人向他们表示深深的谢意！

最后向在百忙之中评审本文的各位专家、老师表示衷心的感谢！

姓 名：姜宇

出生年月：1994.02.06

E-mail: 447355795@qq.com

性别：男

民族：汉

## 声 明

本论文的工作是 2015 年 12 月至 2016 年 6 月在成都信息工程大学信息安全工程学院完成的。文中除了特别加以标注地方外，不包含他人已经发表或撰写过的研究成果，也不包含为获得成都信息工程大学或其他教学机构的学位或证书而使用过的材料。

关于学位论文使用权和研究成果知识产权的说明：

本人完全了解成都信息工程大学有关保管使用学位论文的规定，其中包括：

- (1) 学校有权保管并向有关部门递交学位论文的原件与复印件。
- (2) 学校可以采用影印、缩印或其他复制方式保存学位论文。
- (3) 学校可以学术交流为目的复制、赠送和交换学位论文。
- (4) 学校可允许学位论文被查阅或借阅。
- (5) 学校可以公布学位论文的全部或部分内容（保密学位论文在解密后遵守此规定）。

除非另有科研合同和其他法律文书的制约，本论文的科研成果属于成都信息工程大学。

特此声明！

作者签名：

2016 年 06 月 日