

目录

Java 高级开发课程设计报告	错误!未定义书签。
诗歌网上交流系统	错误!未定义书签。
杨辉	错误!未定义书签。
1. 需求分析	2
1.1 功能模块划分	2
1.1.1 用户模块	2
1.1.2 后台管理	2
1.2 业务实体分析	2
1.3 业务过程分析	3
2. 系统设计与分析	4
2.1 软件体系结构设计	6
2.2 网站整体设计	7
2.3 数据库逻辑设计	7
2.4 主要用户界面设计	8
2.4.1 登录/注册页面	8
2.4.2 主界面	9
2.4.3 CRUD 界面	11
2.5 重要实体类或者业务过程设计	13
3 系统实现	13
3.1 关键技术说明及其应用	13
3.2 关键页面设计：HTML 元素+布局+事件处理	13
3.3 关键 JSP 页面或 Servlet 类设计：输入/输出设计	15
3.4 关键业务类或者业务过程实现说明	16
3.5 数据库物理设计	16
4 系统实现和部署	18
4.1 Web 应用程序部署	18
4.2 数据库部署	18
4.3 其他文件部署及应用程序安装	19

1. 需求分析

当今社会是一个繁忙的社会，人们甚至没有时间静下来“读一首诗”，加上开发者（java高级开发课设小组，杨辉，于佳男）对诗歌有一定的兴趣，我们拟定实现一个网上诗歌在线交流平台，提供好友关注，诗歌在线交流，诗歌种类划分，文件上传等功能，供发展自己的兴趣，同时也提升自己的能力。

1.1 功能模块划分

通过相关经验以及初步的分析，我们的软件系统分为以下两个模块：

1.1.1 用户模块

1. 用户个人中心
包括用户登录注册，以及好友、私信、个人信息维护等信息。
2. 用户发帖/回帖/删帖
登录成功并且有权限的用户能够进行发帖、删帖、回帖等操作。
3. 帖子分类
帖子可以划分为现代诗、古体诗、诗评等类别。
4. 诗集文档、诗歌朗诵音频上传/下载
5. 帖子搜索、查询
6. 用户等级
7. 添加、删除好友
8. 好友私信
用户可以给关注之后的好友发私信。

1.1.2 后台管理

后台管理给管理员用户提供，管理员用户享有以下权限：

1. 帖子管理
帖子管理包括加精、删除、取消加精等操作。
2. 人员管理
管理员可以将用户设置为管理员和拉黑。
3. 资料管理
管理员可以删除任何用户上传的资料。

1.2 业务实体分析

通过相关经验以及抽象建模，抽象出用户、诗歌、诗歌类别、资料、诗歌回复五个实体，加好友、发送消息、上传、下载四个关系。

1. PoemUser
 - a) 用户实体，用户可以有如下操作：添加好友、发送消息、上传/下载资料、发帖/回帖；
2. Updata
 - a) 资料实体，一个资料被一个用户上传，多个用户可以下载多个资料；
3. Reply
 - a) 回复弱实体，通过 id 区别别的回复。
4. PoemPost
 - a) 用户发帖；
5. PoemKind
 - a) 诗歌和资料分类。

1.3 业务过程分析

该系统的业务流程图如图 1.3 所示：

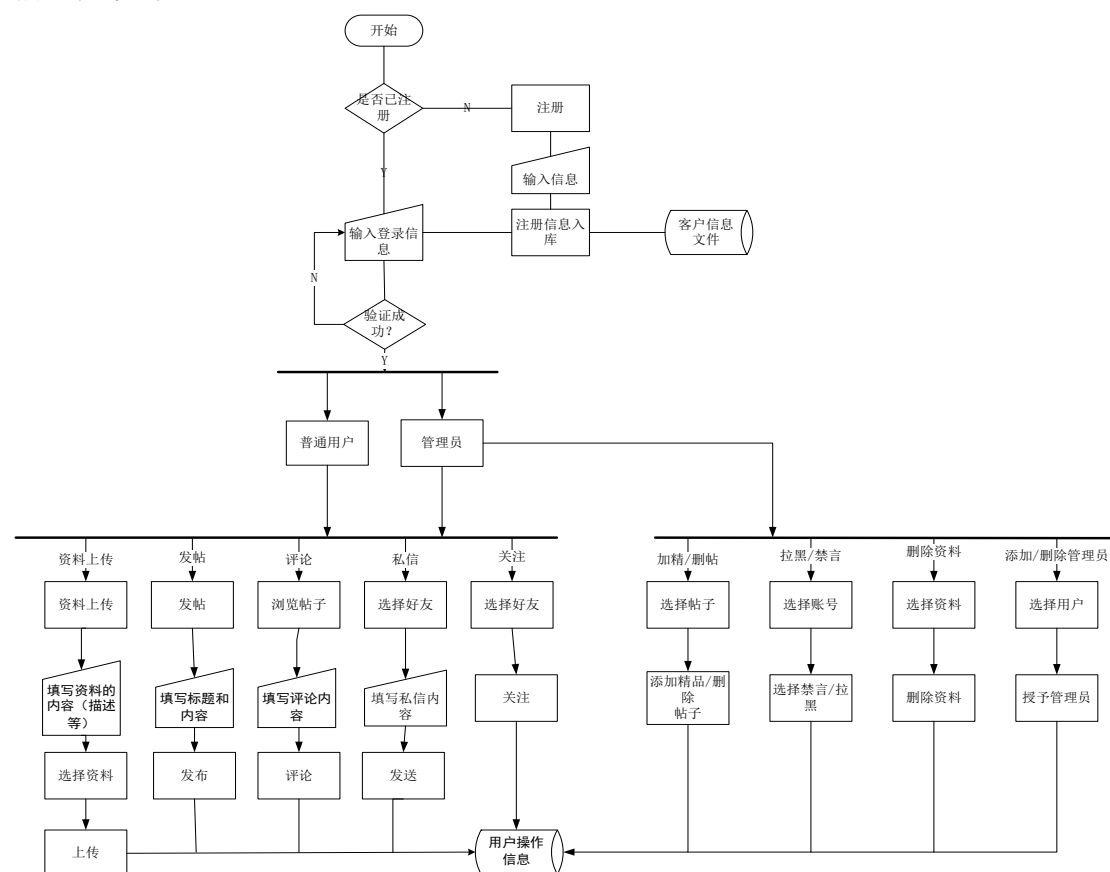


图 1.3 业务流程图

登录成功的用户可以选择管理员和普通用户的操作界面，从而进行不同的功能操作，最后操作的信息返回到数据库中，页面加载的数据再从数据库取出。

2. 系统设计与分析

该系统初步的功能结构图如下图 1.1 所示：

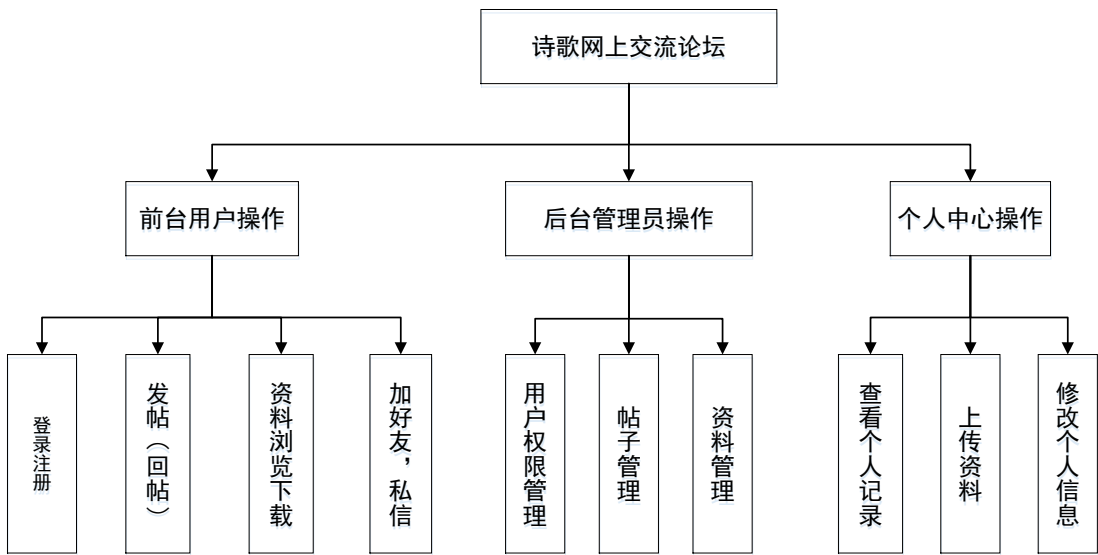


图 2.0.1 功能结构图

该系统分为前台用户操作、后台管理员操作和个人中心三大模块，其详细划分如下：

1. 前台用户操作
 - a) 登录注册：提供用户登录注册的接口，未登录注册的用户只能浏览帖子，并不能评论发帖。
 - b) 发帖/回帖：登录了的用户可以发帖，也能对帖子进行评论操作。
 - c) 资料浏览、下载：用户可以下载论坛上面的一些文档、资料。
 - d) 加好友，私信：拟定实现，但是因为时间问题未实现。用户可以对别的用户进行加好友、私信操作。
2. 后台管理员操作
 - a) 用户权限管理：用户拥有三个权限状态，分别是管理员、普通用户和被禁言的用户，管理员可以改变用户的权限状态。
 - b) 帖子管理：帖子分为精品贴和普通贴，二者有质量的差别，管理员用户可以改变帖子的质量状态。
 - c) 资料管理：管理员可以删除用户上传的资料。
3. 个人中心操作（该模块功能大部分未实现）
 - a) 查看个人纪录：用户可以查看自己的发帖记录、私信记录、上传资料、下载资料、好友列表等内容，目前仅实现了查看发帖记录。
 - b) 上传资料：用户可以上传资料。
 - c) 修改头像：用户可以修改自己的头像。

该系统的流程图如下图 2.0.1 – 图 2.0.5 所示：

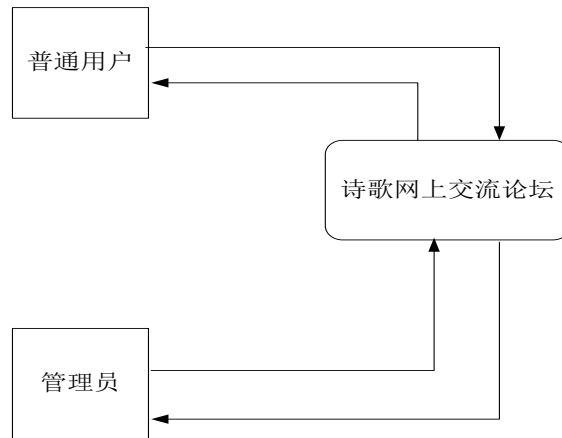


图 2.0.1 顶层数据流图

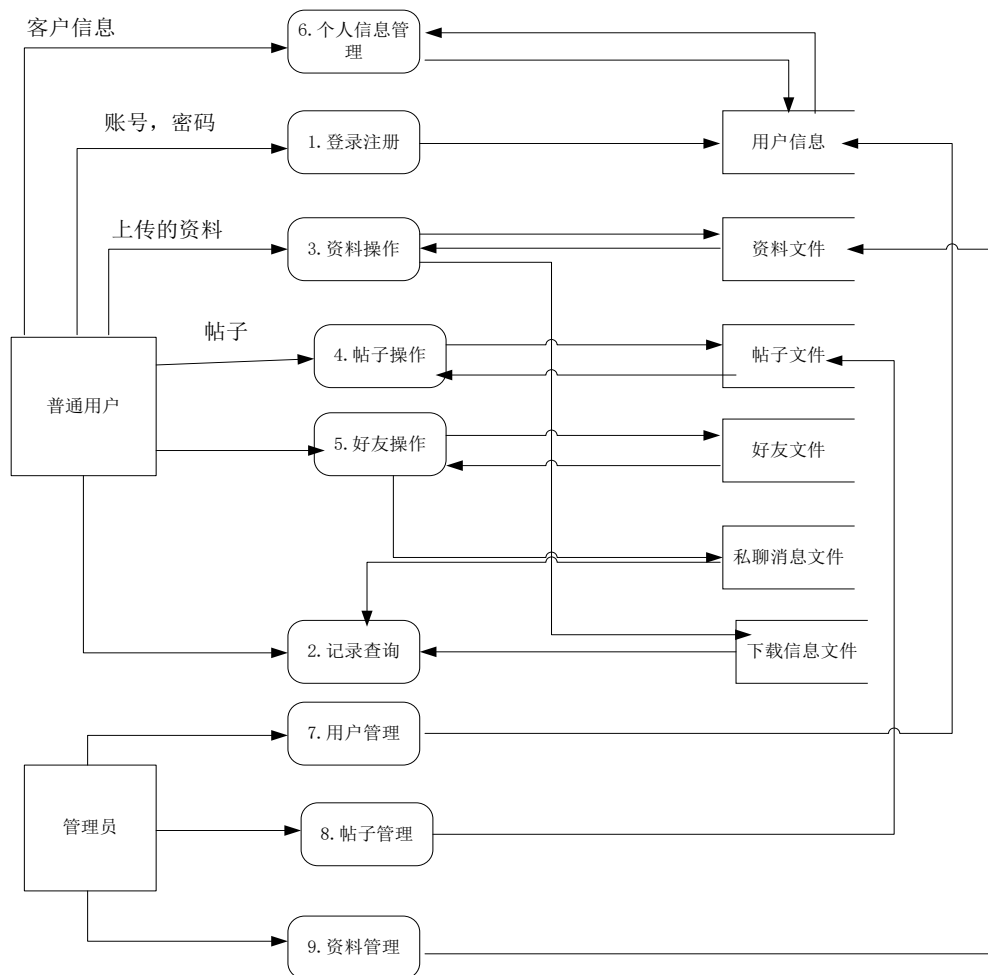


图 2.0.2 0层数据流图

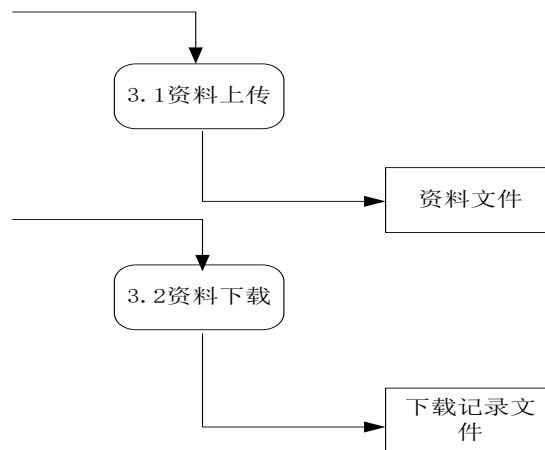


图 2.0.3 1 层数据流图 资料上传和资料下载

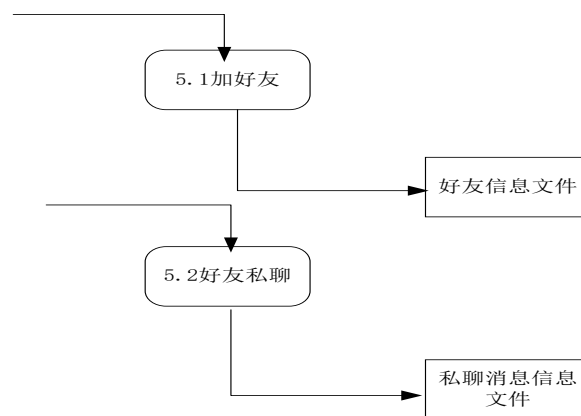


图 2.0.4 1 层数据流图 添加好友和好友私信

2.1 软件体系结构设计

采用 B/S 的结构，其体系结构图如下所示：

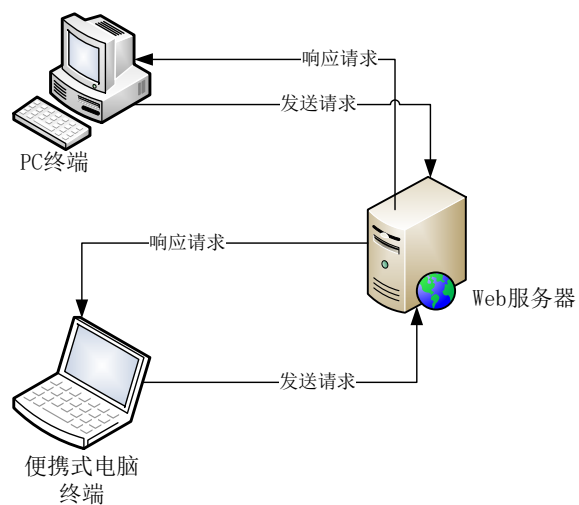


图 2.1 体系结构图

用户通过便携式终端访问 Web 服务器，向 Web 服务器发送请求，Web 服务器解析并

响应请求，并返回响应的数据，便携式终端解析之后展示给用户。

2.2 网站整体设计

网站分为三个模块，分别是浏览信息的前台，个人中心，和后台管理，整个系统的功能根据这三个模块划分。

1. 浏览信息的前台页面：该页面主要是和诗歌交流有关的内容，比如看帖、发帖、回帖。
2. 后台管理：该模块主要是权限控制，网站内容维护的功能。
3. 个人中心：该模块的功能主要是个人信息相关，用户用以处理和自己有关的信息。比如自己的资料管理，帖子管理等内容。

2.3 数据库逻辑设计

数据库文件命名为 Poem，该数据库用来存储用户的操作信息，该数据库用来存储用户操作的各种信息，根据仔细研究和分析以及相关的经验，画出抽象出来的实体以及其关系的 E-R 图如下图 2.3 所示：

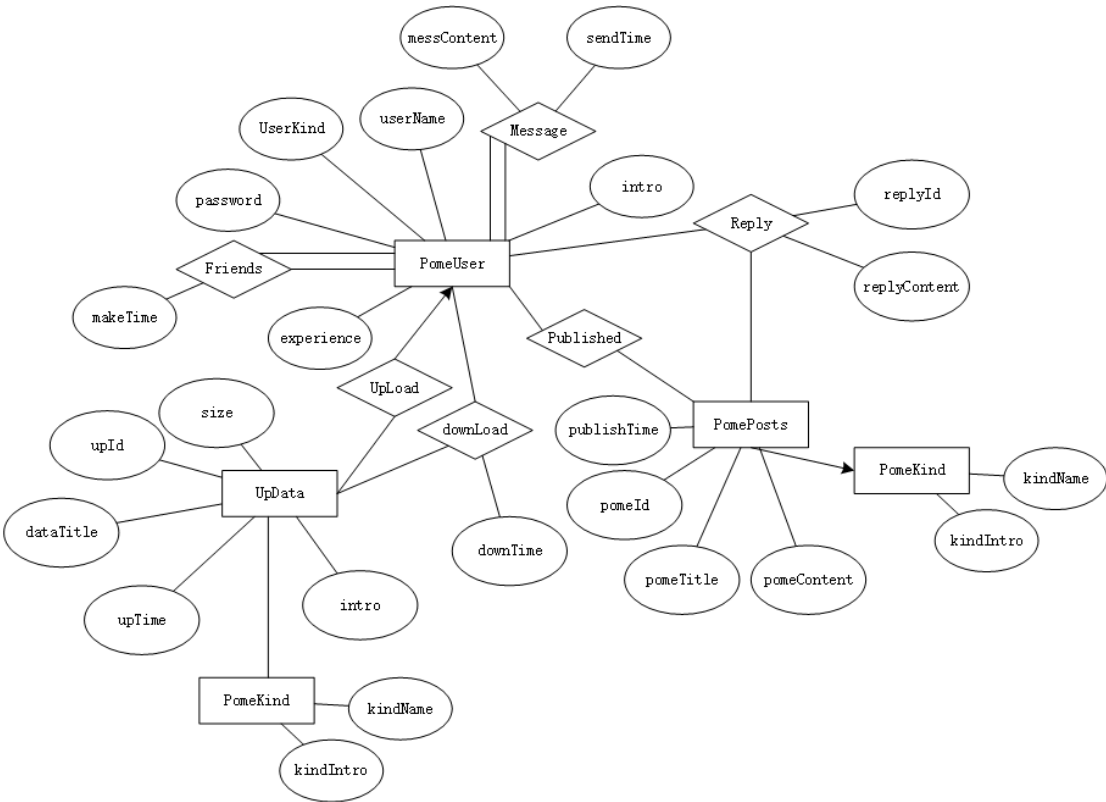


图 2.3 E-R 图

2.4 主要用户界面设计

主要用户界面设计如下所展示。

2.4.1 登录/注册页面

在水一方

输入你的帐号

输入你的密码

登录

图 2.4.1 用户登录界面

用户注册

用户昵称

发呆的淡淡君

✓

该用户名已被注册

密码

.....

✓

确认密码

请确认密码

完成

图 2.4.2 用户注册界面

2.4.2 主界面

2.4.2.1 用户浏览信息主界面

在水一方

看帖

精品

资料中心

后台管理

发呆的淡淡君

按帖子名字搜索

查询

词 | 歌词 | 个人诗集 | 古诗诗 | 散文 | 散文诗 | 现代诗 | 杂文 | 其它

发贴

2

今早八点
还在写代码

发呆的淡淡君
2017-01-06

0

不知道怎么就出bug了
唉????

发呆的淡淡君
2017-01-06

0

格式
天地四方都出bug

淡淡
2017-01-06

1

啊啊啊啊
于佳男测试2333

发呆的淡淡君
2017-01-01

5

人们之所以怀念过去
人们之所以怀念过去并非是因为长久地失

发呆的淡淡君
2017-01-01

精

图 2.4.2.1 信息浏览主界面

2.4.2.2 用户登录浏览帖子以及回帖界面



图 2.4.2.2.1 用户浏览信息主界面



图 2.4.2.2.2 用户回帖界面

2.4.2.3 管理员主界面

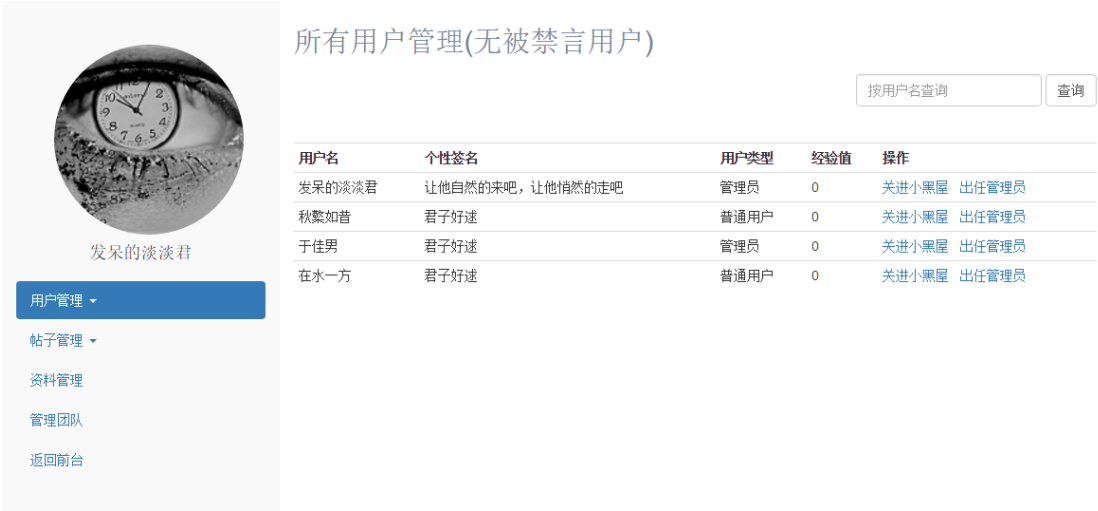


图 2.4.2.3 管理员端主界面

2.4.2.4 个人中心主界面



图 2.4.2.4 个人中心主界面

2.4.3 CRUD 界面

主要的 CRUD 界面如下：

2.4.3.1 用户发帖界面

发表帖子

标题:

</>

B

U

I

S

分类:

词

发表

图 2.4.3.1 用户发帖界面

2.4.3.2 按名字查询用户发帖

在水一方

看帖

精品

资料中心

后台管理

冬天

查询

发帖

0

巴巴的冬天

好冷的冬天

淡淡

2017-01-01

0

冬天

千巴巴的冬天 测试一下

发呆的淡淡君

1994-12-31

精

@在水一方开发团队

上一页

下一页

当前第 1 页

GO

图 2.4.3.2 按名字查询用户发帖

2.4.3.3 上传资料界面

上传资料

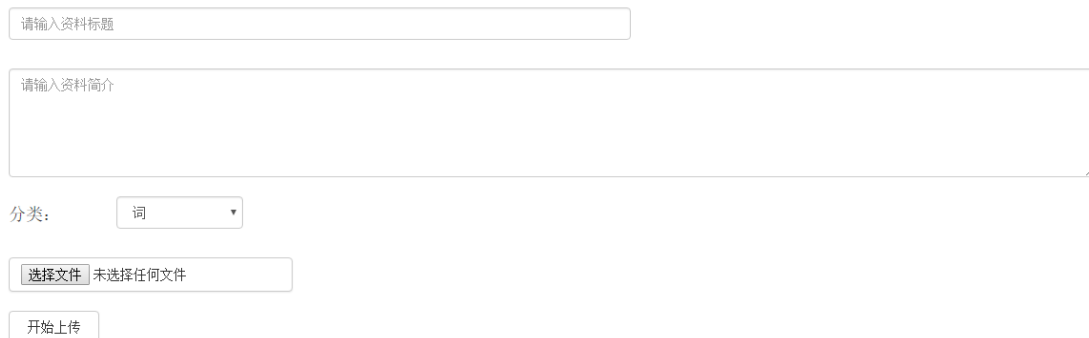


图 2.4.3.3 上传资料界面

2.5 重要实体类或者业务过程设计

3 系统实现

3.1 关键技术说明及其应用

- 该系统使用了 Spring-MVC 的架构，视图层和控制层之间的数据交互采用 ajax 的请求方式，URL 并采用了 REST 的设计风格。采用 JdbcTemplate 和数据库进行交互。
- 前端页面采用 html+少量 css+jQuery 编写前端的响应及框架，使用 bootstrap 进行界面调整及美化。
- 用户登录到服务器，保存密码使用 Md5 加密模式，使得从数据库盗取密码有一定的困难。
- 用户发帖编辑器使用开源 jQuery 插件 wangEditor (<http://www.kancloud.cn/wangfupeng/wangeditor2/>)，能够获取文本格式对应的 html 代码，故而可以直接保存用户格式。
- 用户文件上传至服务器 uploads 文件夹，并将文件路径保存至数据库，文件上传采用的 Spring-MVC 的 MultipartConfig 方式。

3.2 关键页面设计：HTML 元素+布局+事件处理

- 帖子的展示页面布局如下图 3.2.1 所示：

图 3.2.1 帖子展示布局

由于要将帖子规整的以这样的排版大量显示，直接用 js 代码添加页面太过麻烦。所以使用了 template，作为一个模板，最后直接将获取到的内容赋值到相应位置（资料下载展示等页面雷同）。

- 导航栏设计如下图 3.2.2 所示：



图 3.2.1 后台管理导航栏

该页面左侧导航栏用户管理、帖子管理、资料管理、团队管理都是在同一个 html 界面不同的 div 里面，通过控制 div 的隐藏与显示，显示不同的内容（最后内容太多，eclipse 对网页的编程又不太支持，所以很累）。

导航栏在左侧和右侧内容分开显示,使用的使 bootstrap 的栅格系统以及竖版的导航栏，控制不同板块的宽度控制显示内容的大小。

- 以上页面的事件处理

因为使控制不同界面的显示与不显示，而页面元素又很多，所以如果没有规定好，逻辑控制起来就会显得十分混乱，后台管理员端的逻辑控制代码如下图 3.3.3 和图 3.3.4:

```
/** 取消左侧导航栏的Active状态 ... */  
function deleteActive() {...}  
/** 点击"nav_user_manager" 的事件 (人员管理Active) ... */  
function click_UM() {...}  
/** 点击"nav_posting_manager" 的事件 (帖子管理Active) ... */  
function click_PM() {...}  
/** 点击"nva_data_manager" 的事件 (资料管理Active) ... */  
function click_DM() {...}  
/** 点击"nva_team_manager" 的事件 (团队管理Active) ... */  
function click_TM() {...}
```

图 3.2.2 响应代码

```
<div class="row">
  <!-- 这是一个头(左侧导航栏)-->
  <div class="col-xs-3" id="header"...>
  <!--右侧页面-->
  <div class="col-xs-9">
    <!-- 所有用户管理-->
    <div id="user_manager_all"...>
    <!-- 封禁用户管理-->
    <div id="user_manager_banned" style="display: none;...">
    <!-- 帖子管理-->
    <div id="post_manager_all" style="display: none;...">
    <!-- 精品帖子管理-->
    <div id="post_manager_fine" style="display: none;...">
    <!-- 资料管理-->
    <div id="data_manager" style="display: none;...">
    <!-- 团队成员管理-->
    <div id="team_manager" style="display: none;...">
  </div>
</div>
```

图 3.2.3 后台管理 html 代码

3.3 关键 JSP 页面或 Servlet 类设计：输入/输出设计

该页面采用 Spring-MVC REST 设计风格，但是没有使用 JSP 和 Servlet，请求 URL 对应的方法数据如下表所示：

类	URL	方法	输入	输出	功能
DataRest	fileUpload	post	用户提交的表单	上传成功/失败页面	文件上传
	queryData/{title}	get	String	UpData[]（资料信息）	按照资料标题查询资料
	deleteData/{dataId}	get	int	int	删除资料
	showData/{upId}	get	int	DataAndUser	按照id查找资料
KindRest	poemKind	get	null	String	请求帖子的分类
PostRest	publish	post	Json	int	发帖
	publish/{poemId}	get	int	PoemPost	按id查找帖子
	getPost/{title}/{fine}	get	String, int	PostAddReply[]	按照名字和是不是精品查找帖子
	getPost/{kindName}	get	String	PostAddReply[]	按照分类查找帖子
	changePost/{poemId}/{fine}	get	int, int	int	改变诗歌的状态（加精）
	deletePost/{poemId}	get	int	int	按照编号删除帖子
	userPost/{userName}	get	String	PostAddReply[]	根据用户名查找帖子
ReplyRest	addReply	post	json	int	增加回复
	queryReply/{poemId}	get	int	ReplyAndUser[]	根据帖子获取其回复
UserRest	userLogin/{userName}/{password}	get	String, int	PoemUser	检查用户登录是否成功
	userRegister/{userName}	get	String	int	检查用户是否已注册
	userRegister/{userName}/{password}	get	String, String	int	用户注册
	allUser/{userName}	get	String	PoemUser[]	根据名字查询用户
	bannedUser/{userName}	get	String	PoemUser[]	查询被禁言的用户
	changeStae/{userName}/{state}	get	String,int	int	修改用户状态
	changeKind/{userName}	get	String	int	修改用户权限
	querymanager	get	null	PoemUser[]	请求管理员信息
	removeManager/{userName}	get	String	int	按姓名移除管理员

图 3.3.1 Rest 风格 URL 功能表

用户请求数据出了资料上传外，都是使用的 ajax 结合 Spring-MVC Controller 控制器。

3.4 关键业务类或者业务过程实现说明

因为程序比较简单，所以 Service 大多都是直接的转发数据接口层的方法，除了注册中增加了一些默认的配置，使用了 md5 加密之外，其余的没有什么好说的。如下图 3.4.1：

```
/**
 * 用户注册
 * 管理员状态UserKind, 1, 普通用户0
 * 用户权限, UserState 1, 封禁0
 * @param userName
 * @param password
 * @return
 */
public int register(String userName, String password){
    String icon = "resources/img/headimg.jpeg";
    String intro = "君子好逑";
    PoemUser user = new PoemUser(userName, MD5Utils.md5(password), intro, 0, 0, 1, icon);
    return userDao.register(user);
}
```

图 3.4.1 注册用户增加默认配置

3.5 数据库物理设计

数据库的 CDM 图如下图 3.5.1 所示：

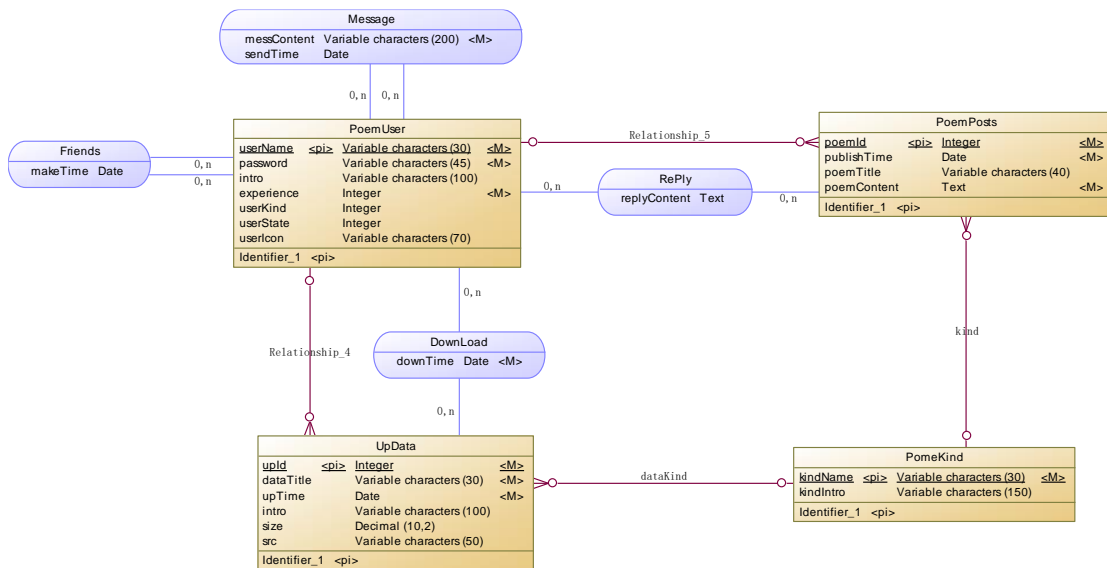


图 3.5.1 CDM 图

生成的 PDM 图如下图 3.5.2 所示：

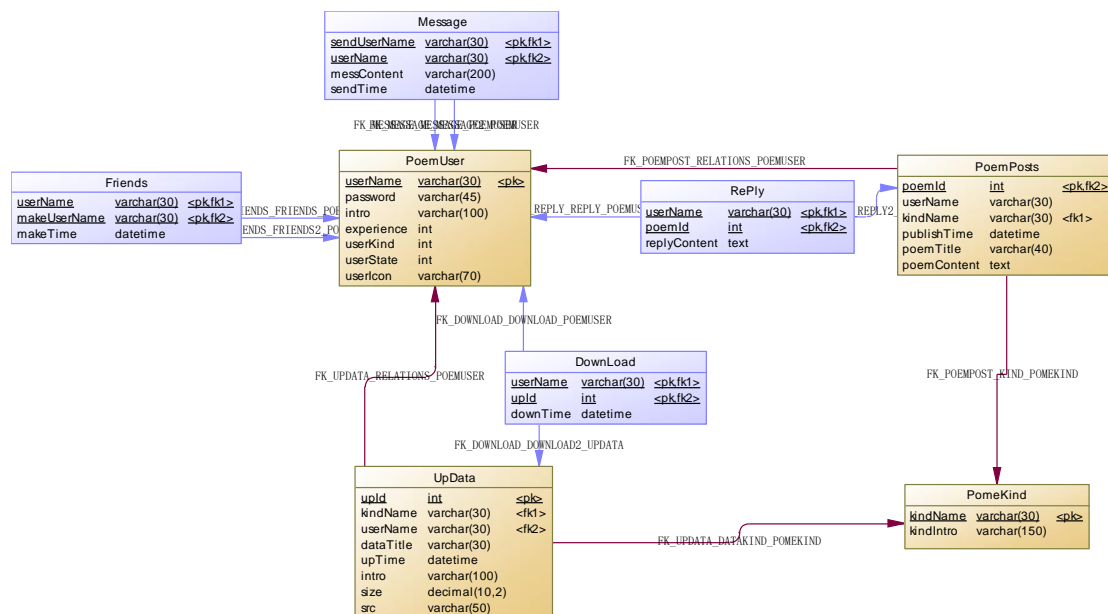


图 3.5.2 PDM 图

数据库图如下表 3.5.1-3.5.8 所示：

表 3.5.1 诗歌帖子信息表

列名	数据类型	可否为空	说明
pomeld	Int	not null	诗歌帖子 ID
username	varchar(30)	not null	用户名
kindname	varchar(30)	not null	诗歌帖子种类
PublishTime	datetime	Not null	发布时间
pometitle	Varchar(40)	not null	发布主题
pomecontent	text	not null	发布内容
precontent	Varchar(100)	not null	
fine	fine	not null	

表 3.5.2 用户信息表

列名	数据类型	可否为空	说明
username	Varchar(30)	not null	用户名
userkind	int	null	用户类型
password	Varchar(45)	not null	密码
intro	Varchar(100)	null	用户简介
userState	Int	null	用户状态
experience	int	not null	用户经验值
userIcon	varchar(70)	null	

表 3.5.3 诗歌类别信息表

列名	数据类型	可否为空	说明
kindname	Varchar(30)	not null	种类名
Kindintro	Varchar(150)	null	种类简介

表 3.5.4 私聊信息表

列名	数据类型	可否为空	说明
sendusername	Varchar(30)	not null	发消息者用户名
username	Varchar(30)	not null	用户名

messContent	Varchar(200)	not null	消息内容
sentTime	datetime	null	信息发送时间

表 3.5.5 加好友信息表

列名	数据类型	可否为空	说明
username	Varchar(30)	not null	用户名
makeUserName	Varchar(30)	not null	加好友的用户名
makeTime	datetime	null	加好友时间

表 3.5.6 下载信息表

列名	数据类型	可否为空	说明
username	Varchar(30)	not null	用户名
upId	int	not null	资料 ID
downTime	datetime	not null	下载时间

表 3.5.7 回帖信息表

列名	数据类型	可否为空	说明
username	Varchar(30)	not null	用户名
poemId	int	not null	诗歌帖子 ID
replyContent	text	not null	回帖内容
replyTime	date	null	回帖时间
replyId	int	Not null	回帖 ID

表 3.5.8 资料信息表

列名	数据类型	可否为空	说明
upID	int	not null	上传资料 ID
kindName	Varchar(30)	null	资料类型名
userName	Varchar(30)	null	用户名
dateTitle	Varchar(30)	not null	用户简介
uptime	datetime	not null	上传时间
intro	Varchar(30)	null	简介
size	bigint	null	大小
src	Varchar(50)	null	路径

4 系统实现和部署

4.1 Web 应用程序部署

将导出的 WAR 程序包拷贝到 tomcat 的 webapp 目录下发布，自动解压运行。

4.2 数据库部署

修改 bean.xml 中数据库连接的 url 的 userName 和 password 设置数据库连接。

4.3 其他文件部署及应用程序安装

因为 bootstrap 引用的有的包是网上的，所以必须保证程序连接到外网。