桂林電子科技大學
GUILIN UNIVERSITY OF ELECTRONIC TECHNOLOGY

软件开发实训
课程设计报告

题　　目：**留 言 板**

学　　院：计算机与信息安全学院

专　　业：计算机科学与技术

学生姓名：蓝琳琪

学　　号：1400310117

指导教师：梁 海

2017 年 12 月 24 日

1、课设任务

1）本功能：注册、登录、注销、匿名提交留言、实名提交留言、留言管理（回复、删除）。

2）留言只有在管理员回复以后才能显示在页面上，界面设计美观大方。

2、详细设计

2.1 结构设计

本次课程设计所开发的是留言板平台，主要面向一般用户、管理员两类用户，以下，先给出本次系统的总体结构。

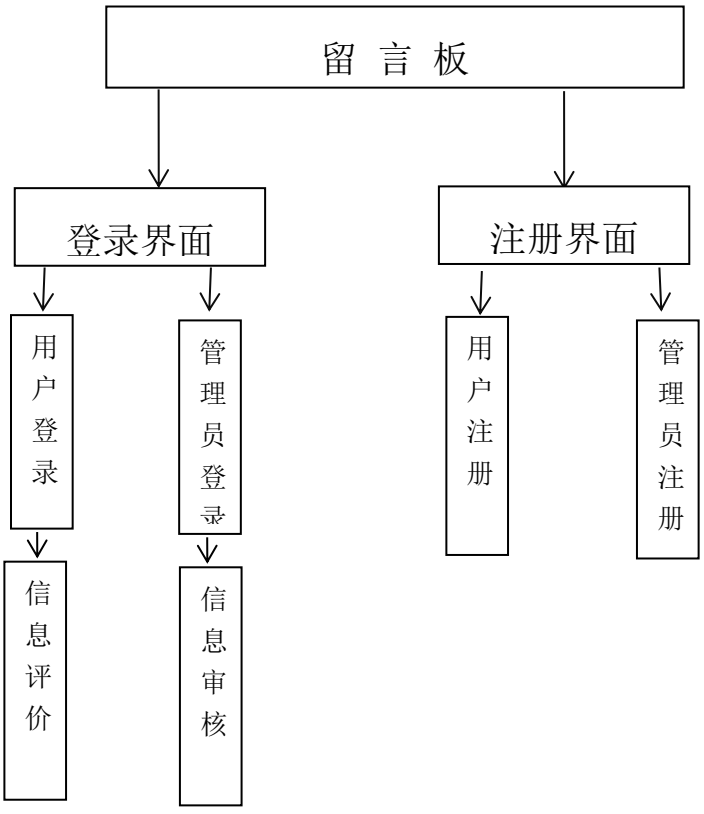

图 2.1 层次方框图

2.2 数据库设计

本系统属于小型网站，主要采用 MYSQL 数据库，原因是采用的数据不是很多，

建的表很少，MYSQL 完全可以胜任，且 MYSQL 操作简单，处理方便，是不二之选。

本系统所需建的表主要有一般用户表、管理员表、信息发布表，分别进行用户数据、管理员数据、发布信息数据存储，表结构如下表 2.2.1，表 2.2.2，表 2.2.3：

表 2.2.1 用户表

字段名	类型	宽度	是否主码	是否为空	描述
idclient	int		PK	否	用户 ID
name	VChar	100		否	用户名
password	VChar	100		否	密码

表 2.2.2 管理员表

字段名	类型	宽度	是否主码	是否为空	描述
idmanage	int		PK	否	管路员 ID
name	VChar	100		否	管理员名
password	VChar	100		否	密码

表 2.2.3 留言信息表

字段名	类型	宽度	是否主码	是否为空	描述
idmessage	int		PK	否	信息 ID
idclient	int			否	用户 ID
idmanage	int			否	管理员 ID
mes	Vchar	200		否	信息体
flag	int			否	审核标记

E-R 图是决定数据库中表的逻辑关系与整体数据库设计的关键一环，详细 E-R 图如下图 2.2.4：

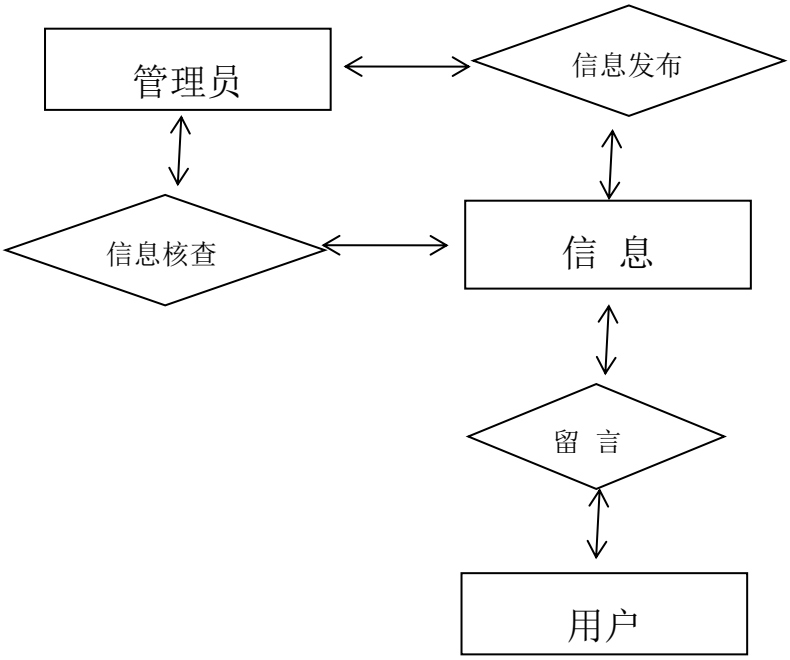

图 2.2.4 总体 E-R 图

3、系统设计

3.1 Dao 层设计

（1）Dao 层实现与数据库的直接相连，实现对数据库数据的直接操作，本项目所需要对用户表、管理员表、信息表的处理，在包含数据库的工程项目开发，DAO 设计的好坏，直接影响到整个工程的结构，一个好的 Dao 设计，可以大大减少工程的代码量。

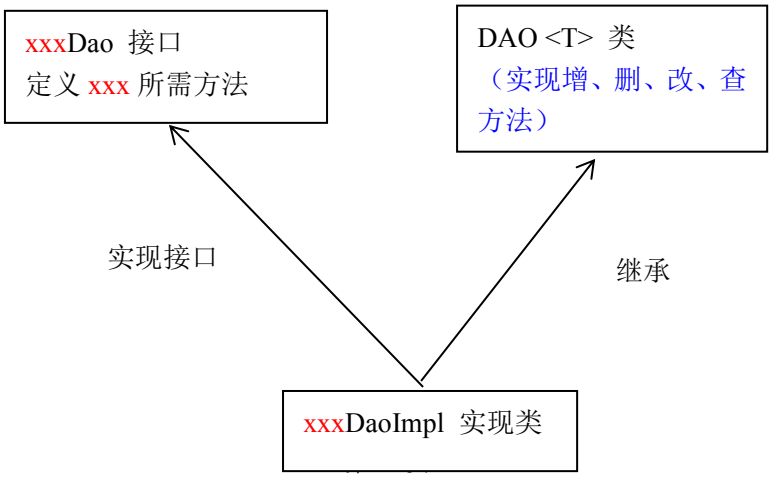

图 4.1 DAO 设计图

DAO<T> 类是利用泛型实现通用于各个实现类的增、删、改、查方法；xxxDao 接口定义所需方法，这些方法主要是对特定表的相关处理；xxxDaoImpl 类调用 DAO<xxx>中通用方法实现 xxxDao 接口的方法。Servlet 层只需调用 xxxDaoImpl 实现类即可，若需要修改，先修改 xxxDao 接口，系统会报 xxxDaoImpl 类的错误，再实现所需方法即可。

（2）**代码实现：**

1、实现对应 UserDA 接口相应方法；

```
@Override
    public void save(User user) throws SQLException{
        // TODO Auto-generated method stub
        String sql="insert into user.message(mes) values(?)";
            update(sql,user.getMes());
    }

    @Override
    public long getCount(Client c) throws SQLException {
        // TODO Auto-generated method stub
        String sql="select count(*) from user.login where name=? and password=?";
        return getForValue(sql, c.getName(),c.getPassword());

    }

    @Override
    public long getCount1(Client c) throws SQLException {
        // TODO Auto-generated method stub
```

```
        String sql="select count(*) from user.login where name=?";
        return getForValue(sql, c.getName());
    }

    @Override
    public void saveClient(Client client) throws SQLException {
        // TODO Auto-generated method stub
        if(getCount1(client)!=0){
            return;
        }
        String sql="insert into user.login(name,password) values(?,?)";
        update(sql,client.getName(),client.getPassword());
    }
```

3.2 中间层设计

在中间层的设计，主要实现与 Dao 层的结合，实现与 View 层的交互，在中间层，我们主要采用 RESTful 实现前端与后端的分离，基于这个风格设计的软件可以更简洁，更有层次，更易于实现缓存等机制。

REST 是 Web 应用都应该遵守的架构设计指导原则，符合 REST 设计标准的 API，即 RESTfulAPI。根据我对其的理解，我认为所谓 RESTfulAPI，应该是指通过具体的 URI 定位符，找到对应的资源，然后以固定的格式返回数据，这样的才是 RESTfulAPI。在本次课设的开发设计中，将严格遵循 REST 的规范来实现 RESTfulAPI 接口。在 Web 应用中客户端访问是通过 HTTP 来进行访问，而 HTTP 的访问必须满足 Rest 规范来实现,HTTP 把对一个 URI 的操作限制在了 4 个之内：GET、POST、PUT 和 DELETE。在本系统中我设计将通过 GET 请求来获取数据，通过 POST 来添加数据，通过 PUT 来更新数据，以及通过 DELETE 来删除数据，在系统中后端人员将 API 接口完成，前端人员只需要通过以上方法来调用 API，便可

以很方便的实现一个前后端分离的项目。下面将详细介绍本系统对 RESTful 的设计与实现。

（1）首先要先了解 RESTful API 的机制，通过获取的 POST 或 GET 请求，来进行相应的处理的操作。

（2）通过 URL 来传输相应的数据，可调用 Dao 层的方法进行操作，也可以直接处理后返回 View 层 XML 数据或 json 数据。

（3）RESTful 代码实现：

```
//@GET 表示方法会处理 HTTP GET 请求
@GET
//这里@Path 定义了类的层次路径。指定了资源类提供服务的 URI 路径。
@Path("/name1/{text}")
//@Produces 定义了资源类方法会生成的媒体类型。
@Produces(MediaType.TEXT_XML)
//@PathParam 向@Path 定义的表达式注入 URI 参数值。
public void login(@PathParam("text") String text) {

String n = text;
User user=new User();
user.setMes(n);
UserDao udi=new UserDaoImpl();
try {
    udi.save(user);
} catch (SQLException e) {
    e.printStackTrace();
}
}

@GET
@Path("/age/{j}&{k}")
@Produces(MediaType.TEXT_XML)
public void userAge(@PathParam("j") String j , @PathParam("k") String k) {

String name=j;
String password=k;
Client client=new Client();
client.setName(name);
client.setPassword(password);
UserDao udi=new UserDaoImpl();
try {
    udi.saveClient(client);
```

```
} catch (SQLException e) {
    e.printStackTrace();
}
}
}
```

3.3 View 层设计

视图的设计，主要决定该系统给用户的感受。本系统采用 HTML 实现的界面，主要通过 JavaScript 与 css 实现界面的动态与样式。通过 POST 与 GET 发送往中间层设计。在 View 层有登录界面与信息发布界面。

（1）登录界面：主要有用户登录与用户注册，登录需要同数据库联系，查询数据库是否存在该用户且密码是否匹配。注册也需要同数据库联系，查询用户是否同名，以判断是否可注册；

（2）信息发布界面：用于用户信息发布，采用 Jqurey 对界面的动态修改，并在 click 事件中采用 location.href=URL 跳转到 URL,即调用中间层 RESTful 进行操作。

（3）代码实现：

登录界面相关：以下仅仅是“注册”点击的功能，获取表单数据后，通过 URL 访问 RESTful 接口，进行相关操作

```
<script type="text/javascript">
    function out1(){
    var oneText=document.getElementById("first").value;
    var twoText=document.getElementById("two").value;
   // alert(oneText);

   location.href="http://localhost:8989/RESTfulWS1/rest/UserInfoService/age/"+oneText+"&"+twoText;
    }
    </script>
```

信息发布界面相关：以下仅仅是“信息评论”点击的功能，获取文本域数据后，通过 URL 访问 RESTful 接口，进行相关数据库操作，之后通过 js 实现界面的动态修改

```
<script type="text/javascript" src="use/main.js"></script>
<script type="text/javascript" src="use/sinaFaceAndEffec.js"></script>
<script type="text/javascript">
    // 绑定表情
    $('.face-icon').SinaEmotion($('.text'));
```

```
    // 测试本地解析
    function out() {
        var inputText = $('.text').val();
        $('#info-show ul').append(reply(AnalyticEmotion(inputText)));
    location.href="http://localhost:8989/RESTfulWS1/rest/UserInfoService/name1/"+inputText;
    }

    var html;
    function reply(content){
        html  = '<li>';
        html += '<div class="head-face">';
        <!--html += '<img src="images/1.jpg" / >';-->
        html += '<img src="use/1.jpg" / >';
        html += '</div>';
        html += '<div class="reply-cont">';
        html += '<p class="username">小小红色飞机</p>';
        html += '<p class="comment-body">'+content+'</p>';
        html += '<p class="comment-footer">2016 年 10 月 5 日  回复  点赞 54  转发 12</p>';
        html += '</div>';
        html += '</li>';
        return html;
    }
</script>
```

4、系统实现

4.1 功能实现

4.1.1 登录模块

留言板的登录模块主要是实现用户的登录与注册，主要操作是在账号、密码中数据，点击“登录”即可，点击“注册”即可进行注册，登录时，若输入数据未能完全匹配，则无法登录；在注册时，若数据库中存在相同的账号（用户名）则无法注册。功能界面如下图 4.1.1：

图 4.1.1 登录注册图

4.1.2 留言板模块

留言板模块主要实现用户的信息留言，主要是面向用户，用户可在留言输入框中输入留言数据，也可在添加表情等，点击“提交评论”即可进行评论，留言的信息会发布到当前界面，并且也会插入数据库进行保存。具体实现主要采用 RESTful 中进行实现，保证在界面无法加载之时，也可通过 URL 进行评论，保证了前端与后端分离，具体的模块实现如下图 4.1.2 与图 4.1.3.

图 4.1.2 信息发布 1

图 4.1.3 信息发布 2

4.2 调试分析与处理

在编程的过程出现过，我往往通过很长时间才能将其处理，这也正说明了我在这方面知识的缺失，以下是我在编程中出现的问题：

（1）在编写 View 层代码，主要 js 与 css,我曾尝试用 Jqurey 获取 input 文本框的 value 值，发现什么弄都获取不了，也上网查了相关资料，问题依旧未能解决，最终只能依靠 JavaScript 获取 value 值，依我判断，Jqurey 本身存在缺陷，Jqurey 虽然简洁，但依旧没有 JavaScript 强大，无法获取表单中的数据，用表单外按钮的点击事件。

（2）在编写中间代码时，主要 GET 和 POST 两种形式 URL，通过对 URL 解析，提取“/”后的数据，在 RESTful 进行处理，开始我以为这只是普通的 java 程序，便设置返回为任意类型，结果不然，应是与@Produces(MediaType.TEXT_XML)相关，且返回的是 xml 数据 json 数据，或者无返回值。

（3）在 Dao 层设计时，没能对反射机制、泛型的准确使用，导致了很多错误，还有在对数据库的相关操作，需用到相应的 SQL 语句，对于不同数据库系统,SQL 语句可能有所不同，我也这一方面吃了很多亏。