****

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： 梁振彰

学 号： 1400310120

指导教师： 梁 海

2017 年 12月 24 日

目录

[1 课程设计任务 3](#_Toc502215896)

[2 详细设计 3](#_Toc502215897)

[2.1 数据库设计 3](#_Toc502215898)

[2.1.1 数据表设计 3](#_Toc502215899)

[2.1.2 相关SQL语句 3](#_Toc502215900)

[2.1.3 E-R图设计 4](#_Toc502215901)

[3 系统设计 5](#_Toc502215902)

[3.1 连接层设计 5](#_Toc502215903)

[3.2 中间层设计 6](#_Toc502215904)

[**4.3 View层设计** 7](#_Toc502215905)

# 课程设计任务

1. 基本功能：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。
3. 界面设计美观大方。

# 详细设计

* 1. 数据库设计
     1. 数据表设计

表2.1.1 用户表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **字段名** | **类型** | **宽度** | **是否主码** | **是否为空** | **描述** |
| **id** | int |  | 是 | 否 | 用户ID |
| name | VChar | 20 |  | 否 | 用户名 |
| password | VChar | 100 |  | 否 | 密码 |
| Role | Int |  |  | 否 | 是否管理员 |

表2.1.3 留言信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **字段名** | **类型** | **宽度** | **是否主码** | **是否为空** | **描述** |
| **Id** | int |  | PK | 否 | 信息ID |
| Name | Vchar | 20 |  | 否 | 用户名 |
| Title | vchar | 200 |  | 否 | 留言标题 |
| Message | Vchar | 200 |  | 否 | 信息体 |
| Time | Datetime |  |  | 否 | 留言时间 |
| flag | int |  |  | 否 | 审核标记 |

* + 1. 相关SQL语句

用户表：

CREATE TABLE `User` (

`name` char(20) NOT NULL DEFAULT '',

`password` char(20) DEFAULT NULL,

`administrator` char(1) DEFAULT NULL,

`ID` bigint(20) DEFAULT NULL,

PRIMARY KEY (`name`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

留言表：

CREATE TABLE `messagess` (

`id` int(11) NOT NULL,

`name` varchar(20) NOT NULL,

`time` datetime NOT NULL,

`title` varchar(20) NOT NULL,

`message` varchar(20) NOT NULL,

`flag` int(11) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

* + 1. E-R图设计

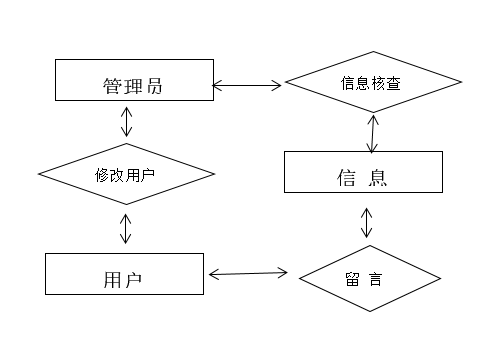


图2.1.3 总体E-R图

**2.2 结构设计**

留 言 板

注册

用户注册

登录

用户登录

管理员登录

留言

留言审核

用户修改

图2.2 层次方框图

# 系统设计

* 1. 连接层设计

连接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，一个好的连接设计，可以大大减少工程的代码量。

将浏览器作为前端，服务器作为后端，结构可以表示如图4.1

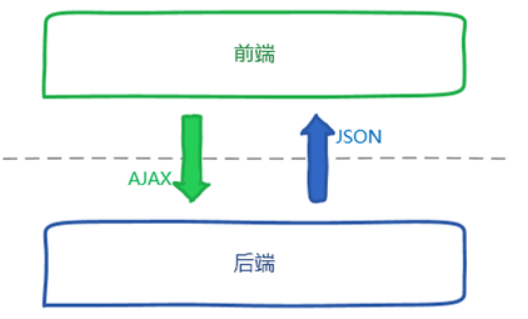


图4.1 前后端设计图

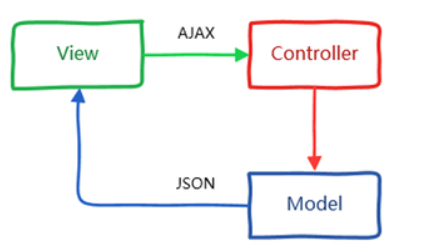


图4.2 数据交互示意图

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。

* 1. 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。

* 1. 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JSP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请求与后台进行数据交互。

前端采用框架：frameset对界面进行分区，主要分为导航功能菜单区和内容显示区。具体设计如下：

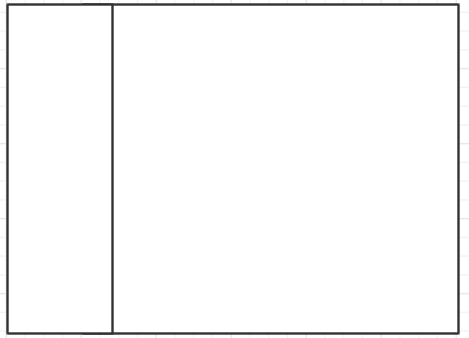


图3.3 前端设计图

# 编码实现

* 1. 后台代码

根据数据库的设计，可得实体类

* + 1. 用户类：

**public** **class** User {

**private** String name;

**private** String password;

**private** String administrator;}

* + 1. 信息类：

**public** **class** MessBoar {

**private** **int** id;

**private** String name;

**private** Date time;

**private** String title;

**private** String message;}

并根据实体设置相应的方法。

* + 1. 操作类：

留言添加：

**public** **boolean** addInfo(MessBoar mb) {

**try** {

*pstmt* = *conn*

.prepareStatement("insert into messagess values(?,?,?,?,?)");

*pstmt*.setInt(1, mb.getId());

*pstmt*.setString(2, mb.getName());

*pstmt*.setDate(3, mb.getTime());

*pstmt*.setString(4, mb.getTitle());

*pstmt*.setString(5, mb.getMessage());

*pstmt*.executeUpdate();

**return** **true**;

} **catch** (Exception e) {

**return** **false**;

}

}

获取留言：

**public** **static** ArrayList<MessBoar> findMbInfo(){

**try**{

ArrayList<MessBoar> al=**new** ArrayList<MessBoar>();

*pstmt*=*conn*.prepareStatement("select \* from messagess");

ResultSet rs=*pstmt*.executeQuery();

**while**(rs.next()){

MessBoar mb=**new** MessBoar();

mb.setId(rs.getInt(1));

mb.setName(rs.getString(2));

mb.setTime(rs.getDate(3));

mb.setTitle(rs.getString(4));

mb.setMessage(rs.getString(5));

al.add(mb);

}

**return** al;

}**catch**(Exception e){

e.printStackTrace();

**return** **null**;

}

}

* 1. 前端代码
     1. 登录设计



图4.1 登录界面

输入框采用JS验证信息是否有效输入，无效输入提示用户。

**function** check(form) {

**with** (form) {

**if** (name.value == "") {

alert("名称不能为空");

**return false**;

}

**if** (password.value == "") {

alert("密码不能为空");

**return false**;

}} }

* 1. 菜单设计

登录后跳转到功能界面

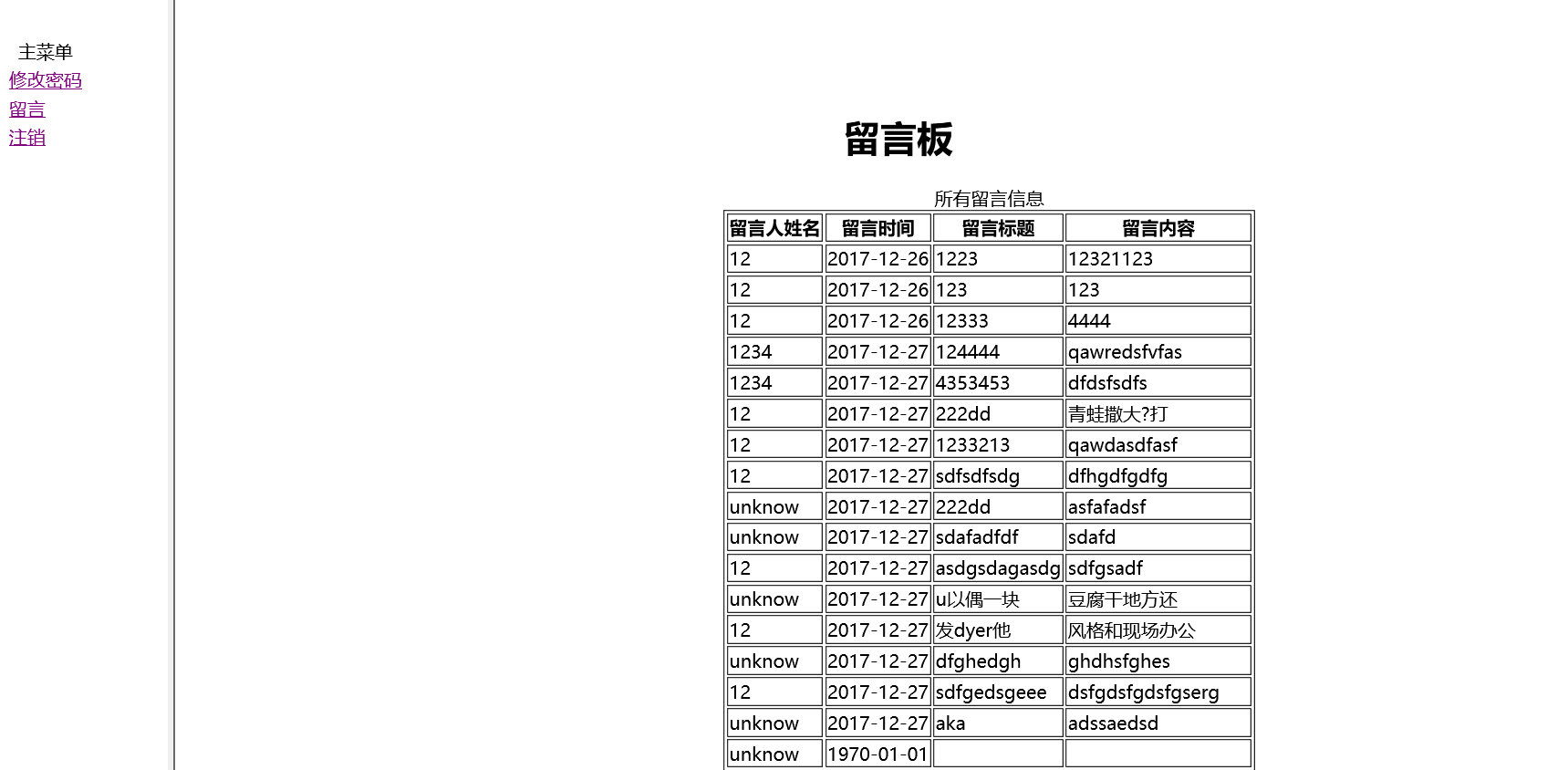


图4.2 主界面

* 1. 留言板设计

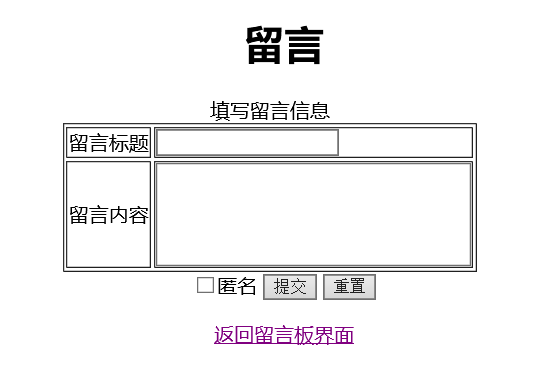


图4.3 留言界面

与登录采用相同的JS验证输入。

# 成果总结

* 1. 完成的设计

本课程设计完成了登录、注册、修改密码、留言功能，都采用了JS进行校验输入，完成了课程设计的主要要求，实现了一个留言板的基本功能。

* 1. 未完成的设计

本课程设计未完成管理员留言审核，restful api的设计。

* 1. 个人心得

本次课程设计完成度一般，仍有许多不完善之处，由于时间紧迫，并且有新内容需要区学习，所以最后很遗憾没能很好的完成本次课程设计。在本次学习过程中，学会到使用JS的方法，后台与前台的数据交互方法，有一定成果。