

编号：_____



桂林电子科技大学
GUILIN UNIVERSITY OF ELECTRONIC TECHNOLOGY

软件开发实训 课程设计说明书

题 目： 留言板的设计与实现

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： 兰罗斌

学 号： 1400310415

指导教师： 梁 海

2017 年 12 月 27 日

| | | |
|-------|--------------|----|
| 1 | 详细设计..... | 1 |
| 1.1 | 系统设计..... | 1 |
| 1.2 | 功能设计..... | 1 |
| 1.3 | 数据库设计..... | 3 |
| 2 | 系统实现..... | 4 |
| 2.1 | 数据库连接..... | 4 |
| 2.2 | 登录模块..... | 4 |
| 2.2.1 | 界面..... | 4 |
| 2.2.2 | 功能..... | 5 |
| 2.2.3 | 代码及说明..... | 5 |
| 2.3 | 用户模块..... | 6 |
| 2.3.1 | 界面..... | 6 |
| 2.3.2 | 功能..... | 6 |
| 2.3.3 | 代码及说明..... | 7 |
| 2.4 | 管理员审核模块..... | 7 |
| 2.4.1 | 界面..... | 7 |
| 2.4.2 | 功能..... | 8 |
| 2.4.3 | 代码及说明..... | 8 |
| 3 | 功能测试..... | 8 |
| 3.1 | 测试用例..... | 8 |
| 3.2 | 功能测试..... | 9 |
| 3.3 | 测试结果分析..... | 10 |
| 4 | 总结..... | 12 |

1 详细设计

1.1 系统设计

该留言板登陆的方式有两种，用户直接通过主界面登陆，管理员通过一个独立的网址登陆，用户的主要功能有：留言、删除留言、注销；管理员的主要功能有：对用户的留言进行添加与删除、注销；系统流程图如图 1.1 所示：

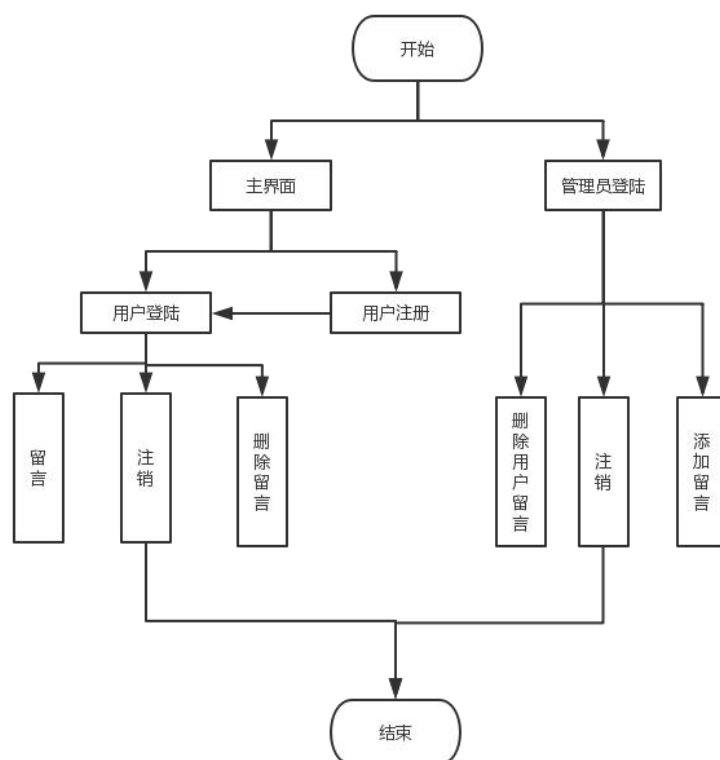


图 1.1 系统流程图

1.2 功能设计

(1) 用户功能中，用户提交留言，先判断留言板内容是否为空，若为空则提示不能为空，若不，则将数据发送到后端，待数据处理之后留言成功；留言流程图如图 1.2 所示：

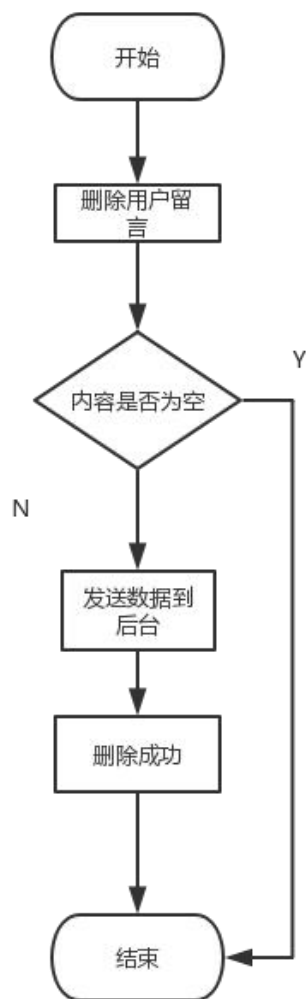


图 1.2 留言流程图

(2) 管理员删除留言功能中，点击删除按键，发送请求到后端，后端则将该条数据从所在的数据库中删除，删除流程图如图 1.3 所示：



图 1.3 删除用户留言流程图

1.3 数据库设计

该留言板的主要对象有两个，一个是用户，包括普通用户和管理员，一个是留言，详细设计如图 1.4 所示，一共有 3 张表，分别用来存放用户的账号密码，管理员的账号密码以及用户的留言。

| 名 | 类型 | 长度 | 小数点 | 允许空值 (| |
|----------|---------|----|-----|--------------------------|---|
| ► ID | varchar | 8 | 0 | <input type="checkbox"/> | 1 |
| Password | varchar | 16 | 0 | <input type="checkbox"/> | |

| 名 | 类型 | 长度 | 小数点 | 允许空值 (| |
|-------|----------|----|-----|--------------------------|---|
| ► N | int | 11 | 0 | <input type="checkbox"/> | 1 |
| User | varchar | 8 | 0 | <input type="checkbox"/> | |
| Ms | text | 0 | 0 | <input type="checkbox"/> | |
| Mtime | datetime | 0 | 0 | <input type="checkbox"/> | |
| Hide | int | 1 | 0 | <input type="checkbox"/> | |
| Pass | int | 1 | 0 | <input type="checkbox"/> | |

| 名 | 类型 | 长度 | 小数点 | 允许空值 (| |
|----------|---------|----|-----|--------------------------|---|
| ► User | varchar | 8 | 0 | <input type="checkbox"/> | 1 |
| Password | varchar | 16 | 0 | <input type="checkbox"/> | |

图 1.4 数据库

2 系统实现

2.1 数据库连接

先新建一个.js 文件并在其中输入数据库相关信息：地址、用户名、密码等。实现代码如下：

```
module.exports = {  
  mysqlInfo : {  
    host : '127.0.0.1',  
    port : 3306,  
    user : 'root',  
    password: '123456',  
    database: 'llb'  
  }  
}
```

再通过 `mysql.createConnection` 方法来连接数据库。

2.2 登录模块

2.2.1 界面

登录界面，本系统设计的登录界面简易并且易操作，有用户登陆、注册的按钮，用户登录界面图如图 1.1 所示：



图 2.1 用户登录界面图

2.2.2 功能

系统的登录模块，若输入密码错误，会出现相应的提示；原始预设值的账号密码单击后将清空输入框中的内容；登陆成功后跳转到留言板主界面。登陆时在前端调用 `API:open("POST", "http://localhost:3000/signin", true);`

登陆的 API 采用 POST 的请求，POST 请求的同时传递参数，其设计为：

```
router.post('/signin', db_handle.signIn);
```

前端只要通过 POST 请求调用该路径即可完成登陆验证工作。

2.2.3 代码及说明

在该功能模块中，主要是进行用户的登录验证的验证，代码设计简易，逻辑清晰，后台先判断是否存在此账号，接着判断密码是否正确。登陆接口主要代码如下：

```
exports.signIn = function (req, res, next) {
    var sql = "select * from tb_user where User=" + "" + req.body.User + "" +
    + " and Password=" + "" + req.body.Password + "" ";i
    Var user={};
    f (req.session.user == null) {
        req.session.user={};
    }

    connection.query(sql, function (err, rows, fields) {
        /*rows 为查询出的数据，可通过 rows[i].字段 访问数据*/
        if (rows.length > 0) {
```

```

        loginUser = req.body.User;
        req.session.user[loginUser] = loginUser;
        return res.send("http://localhost:3000/user/" + req.body.User);
    } else {
        return res.send("账号或密码错误");
    }
}
})

```

2.3 用户模块

2.3.1 界面

用户提交留言界面，界面中的设计了一个文本框一个“提交”按钮和一个“匿名”选中框，底部显示历史留言；需要用户输入留言内容后，点击其中一个按钮完成提交；右上方则显示“注销”、“所有人”“未通过”“历史留言”，用户可以注销以及查看不同类型的留言去操作，用户提交留言界面图如图 1.2 所示：

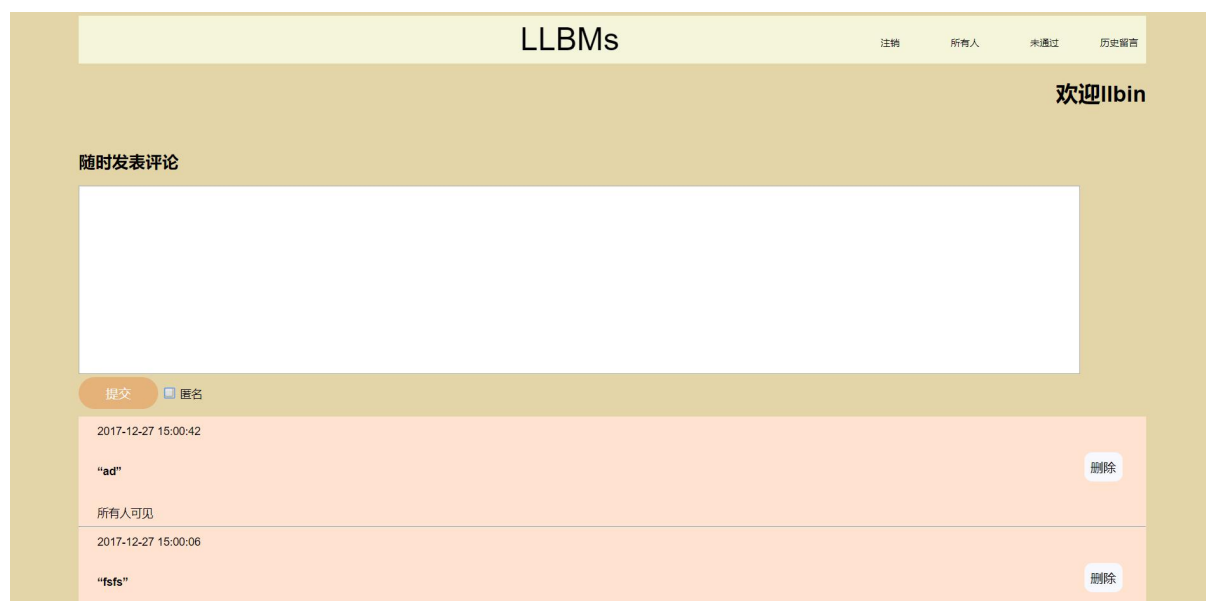


图 2.3 用户提交留言界面图

2.3.2 功能

提交留言模块，在文本框中输入留言后，点击“提交”按钮进行提交，选中“匿名”框则为匿名提交留言；在历史留言中可以删除自己的留言。

2.3.3 代码及说明

在该功能模块中，主要是进行留言的提交和删除，后台接收到数据后，首先设置相应的字段，执行接口中的插入语句，最后将数据插入到数据库中；提交留言接口的主要代码如下：

```
exports.postMessage = function (req, res) {  
    var note = req.body.Ms.replace("","");  
    var sql = "insert into notes(User,Ms,Mtime,Hide,Pass) values('" + req.body.User + "','"  
+ note + "','" + req.body.Mtime + "','" + req.body.Hide + "','" + req.body.Pass + "')";  
    connection.query(sql, function (err, result) {  
        return res.send("留言成功！");  
    });  
};
```

2.4 管理员审核模块

2.4.1 界面

管理员审核留言界面，该界面显示待审核留言，留言右上方显示“所有留言”、“已通过”和“待审核”按键，单击留言的“添加”按钮则该留言通过审核；点击“删除”按钮则删除该留言，对其进行删除操作。管理员审核留言界面图如图 2.4 所示：

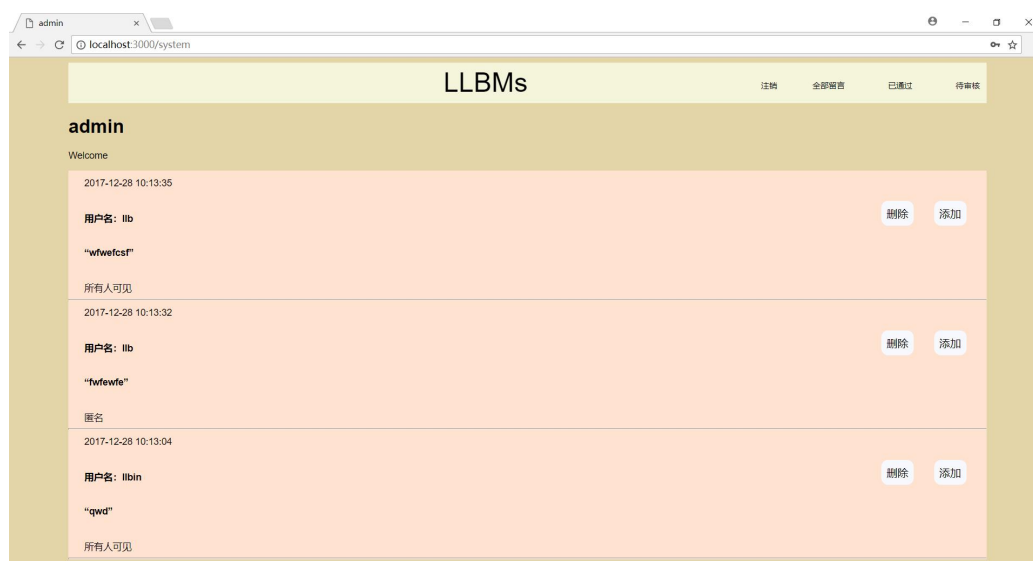


图 2.4 审核留言界面

2.4.2 功能

管理员审核留言模块，管理员登陆后进入审核界面；点击通过“添加”按钮则该留言审核成功，点击“删除”按钮则删除该留言；在右上方可以选择不同类型的留言并对它们进行相应的操作。

2.4.3 代码及说明

在该功能模块中，主要是进行留言的审核，后台先查询出待审核留言的信息，然后管理员进行审核，审核通过，则修改留言的审核状态；审核不通过，则删除该留言。审核留言的接口主要代码如下：

```
exports.addMessage = function (req, res) {
    var sql = "update notes set Pass=2" + " where N='" + req.params.id + "'";
    connection.query(sql, function (err, result) {
        return res.send("添加成功");
    });
};
exports.deleteData = function (req, res) {
    var sql = "delete from User where User='" + req.params.id + "'";
    connection.query(sql, function (err, result) {
    });
    return res.send("DELETE API");
}
```

3 功能测试

3.1 测试用例

为了检验留言板的整体功能是否达到预期的功能要求，各个功能是否满足相应运行结果，对留言板进行测试工作，从而提高软件的质量。

针对用户功能模块，本次软件测试工作的测试用例设计如下：

(1) 用户登录模块。测试能否进入到用户登录界面，测试账号和密码分别为“llb,123”、“llb,000”。

(2) 提交留言模块。测试已注册系统账号的用户能否成功实现提交留言的操作。
用于测试的留言为“留言板”。

(3) 留言审核模块。测试管理员是否能审核留言。

3.2 功能测试

首先针对用户功能模块的主要功能进行测试。

(1) 登录模块：该项测试中，在验证码输入正确的情况下，输入账号和密码为“lib、123”时，登录成功，输入账号和密码为“lib、000”时，登录失败；测试失败时的相关截图如图 3.1 所示：

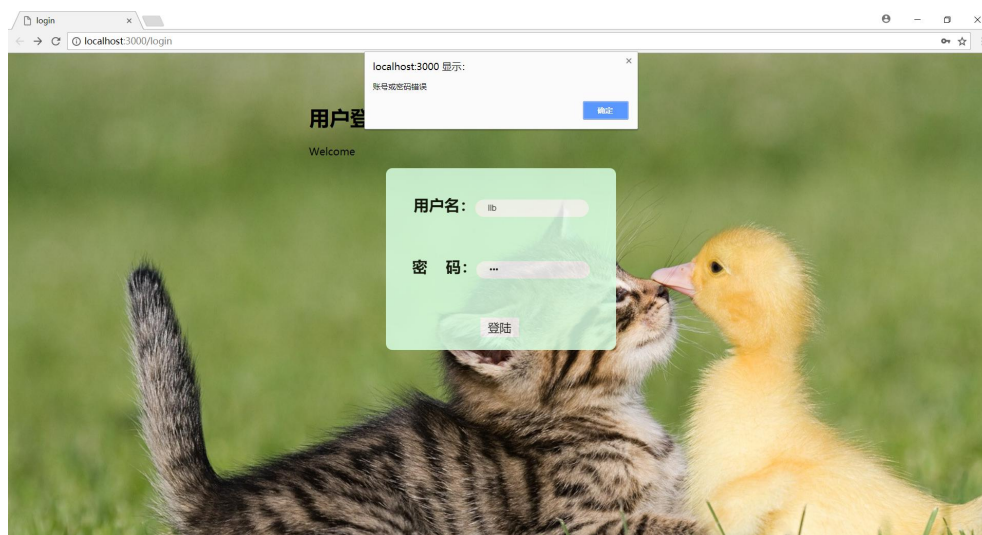


图 3.1 登录模块测试结果图

(2) 提交留言：该项测试中，用户在文本框中输入留言为“留言板”，点击“提交”按钮完成测试；测试时的相关截图如图 3.2 所示：

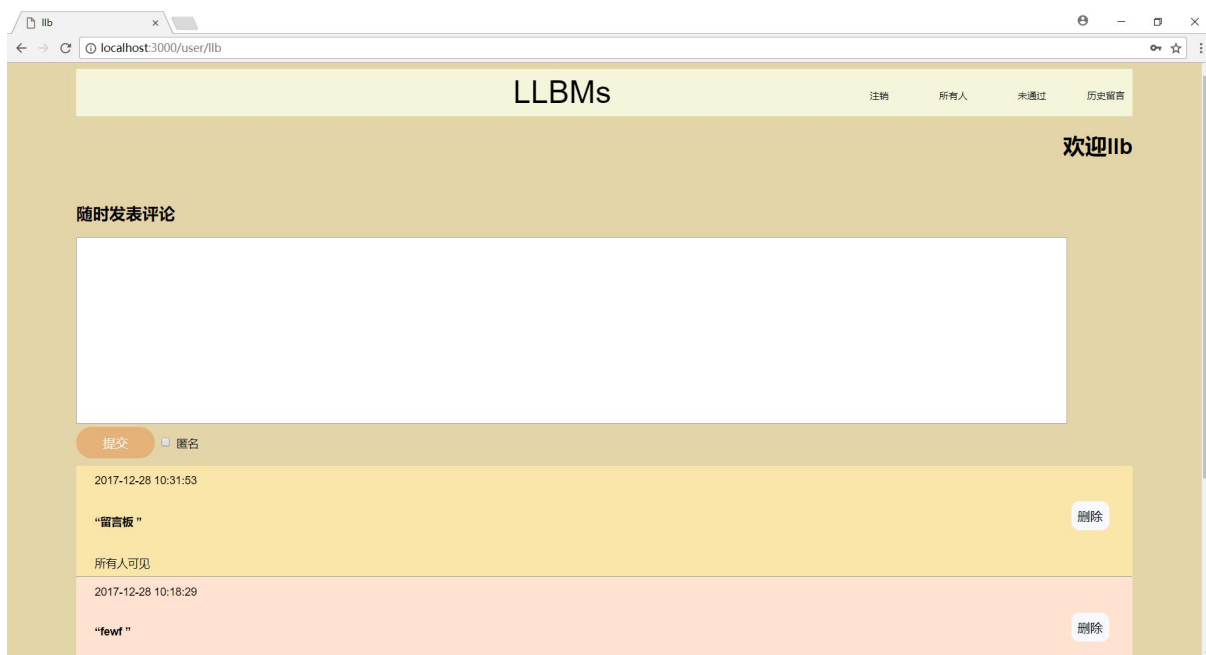


图 3.2 提交留言测试结果图

(3) 审核留言：当管理员审核用户功能模块测试提交的留言，点击“通过审核”，则该留言通过审核；点击“不通过审核”，则该留言被删除；测试截图如图 3.3 所示：

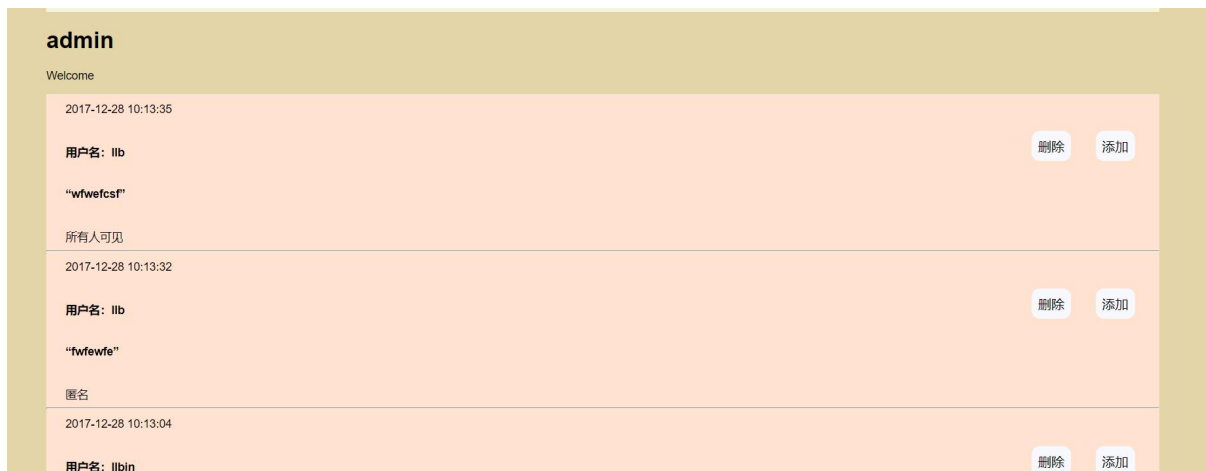


图 3.4 审核留言测试界面

3.3 测试结果分析

(1) 登录模块功能。经测试，该功能实现了用户登录系统的功能，登录验证准确无误，正确的账号密码能登录到相应的主界面，错误的账号密码登录失败，达到了预期效果。

(2) 提交留言功能。经测试，该功能能够成功实现留言提交，这与系统对提交留

言的要求的情况一致，基本达到了预期效果。

（3）审核留言功能。经测试，管理员能完全实现对用户提交的留言进行审核或者删除，与预期结果一致。

4 总结

本留言板在功能设计完成后，能够满足用户的基本需求，普通用户功能模块中已完成的功能有：名留言、匿名留言、查看历史留言、删除历史留言。管理员模块中已完功能有：审核回复留言。本系统具有一定的完整性，并且上述功能已通过相应的测试。

在这次程序开发设计中，我学会了使用 git 保存代码，了解了 restful API 的意义，学会了如何去写接口以及对接口的测试，并使用 webstorm 利用 node.js 来开发网站。虽然难免会遇到困难，有时候一个不经意的微小错误也让我调试很久才能找到。但是这次开发对我们来说可以算是一个全新的尝试，尽管整个过程并不是很顺利，却使我们从中充分体会到学习的快乐，程序也在一步步得到完善，希望能够在后期的学习生活中发现系统的不足以及找到系统更长远发展的可能性与机遇。总之，通过这次课程设计，不仅巩固了所学到的知识，而且还学到了一些新知识，可以说是受益匪浅。