****

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： 黄瑞华

学 号： 1400310414

指导教师： 梁 海

2017年12月25日

目 录

[1 详细设计 1](#_Toc502047280)

[1.1 系统总体流程设计 1](#_Toc502047281)

[1.2 业务流程图 2](#_Toc502047282)

[2 系统实现 5](#_Toc502047283)

[2.1 登录模块 5](#_Toc502047284)

[2.1.1 界面 5](#_Toc502047285)

[2.1.2功能 5](#_Toc502047286)

[2.1.3代码及说明 5](#_Toc502047287)

[2.2 用户提交留言模块 6](#_Toc502047288)

[2.2.1界面 6](#_Toc502047289)

[2.2.2功能 7](#_Toc502047290)

[2.2.3代码及说明 7](#_Toc502047291)

[2.3 用户查看历史留言模块 7](#_Toc502047292)

[2.3.1界面 7](#_Toc502047293)

[2.3.2功能 8](#_Toc502047294)

[2.3.3代码及说明 8](#_Toc502047295)

[2.4 管理员审核留言模块 9](#_Toc502047296)

[2.4.1界面 9](#_Toc502047297)

[2.4.2功能 10](#_Toc502047298)

[2.4.3代码及说明 10](#_Toc502047299)

[3 功能测试 11](#_Toc502047300)

[3.1 测试用例 11](#_Toc502047301)

[3.2 功能测试 11](#_Toc502047302)

[3.3 测试结果分析 13](#_Toc502047303)

[4 总结 14](#_Toc502047304)

# 1 详细设计

## 1.1 系统总体流程设计

本留言板分为管理员功能模块和用户功能模块，管理员功能模块主要功能有：审核回复留言；用户功能模块主要功能有：实名留言、匿名留言、查看历史留言、删除历史留言；系统总体流程图如图1.1所示：

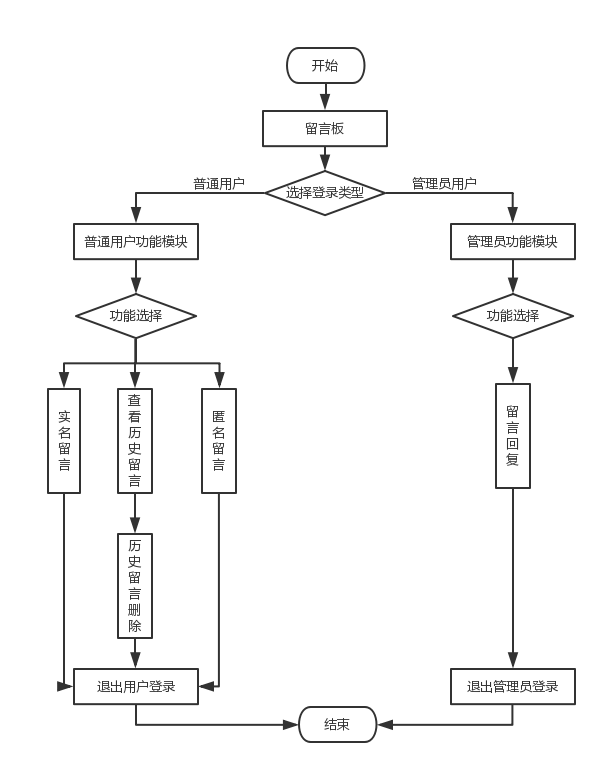


图1. 1 系统总流程图

## 1.2 业务流程图

（1）用户功能模块中，用户选择实名或匿名提交留言，先从前端获取数据，然后后台检查留言内容是否为空，若不，则写入数据，留言成功；若为空，则提示留言不能为空；提交留言流程图如图1.2所示：

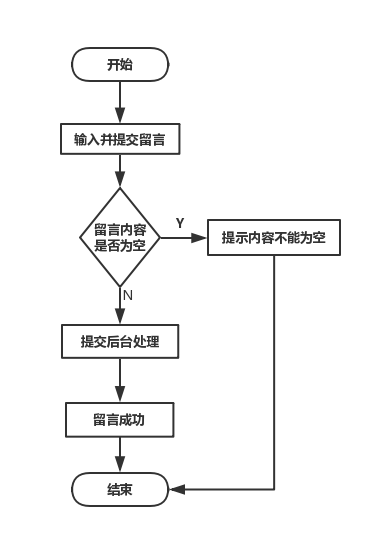


图1. 2 提交留言流程图

（2）用户功能模块中，用户选择查看历史留言,，若存在留言，则显示在界面中；若不存在，则返回数据不存在提示；查看历史留言流程图如图1.3所示：

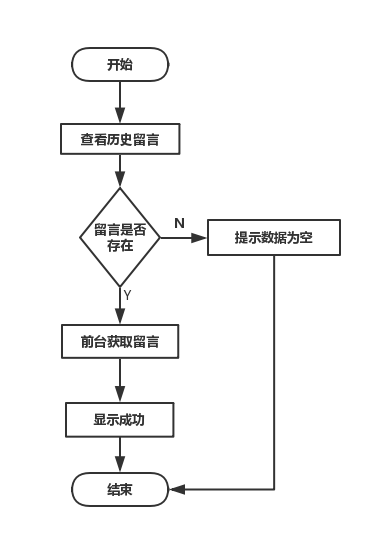


图1. 3 查看历史留言流程图

（3）用户功能模块中，用户选择删除留言，后台先查找出该留言，然后删除选中项，删除留言流程图如图1.4所示：

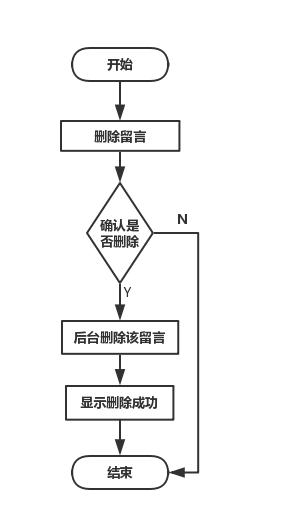


图1. 4 删除留言流程图

（4）管理员功能模块中，管理员可以进行审核留言，若通过审核，则出现相应提示，若不通过审核，则删除该留言；管理员审核留言流程图如图1.5所示：

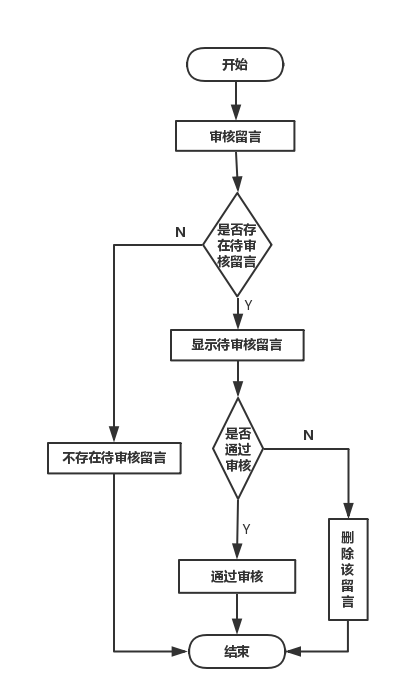


图1. 5 管理员审核留言流程图

# 2 系统实现

## 2.1 登录模块

### 2.1.1 界面

登录界面，本系统设计的登录界面简易并且易操作，，用户登录界面图如图2.1所示：

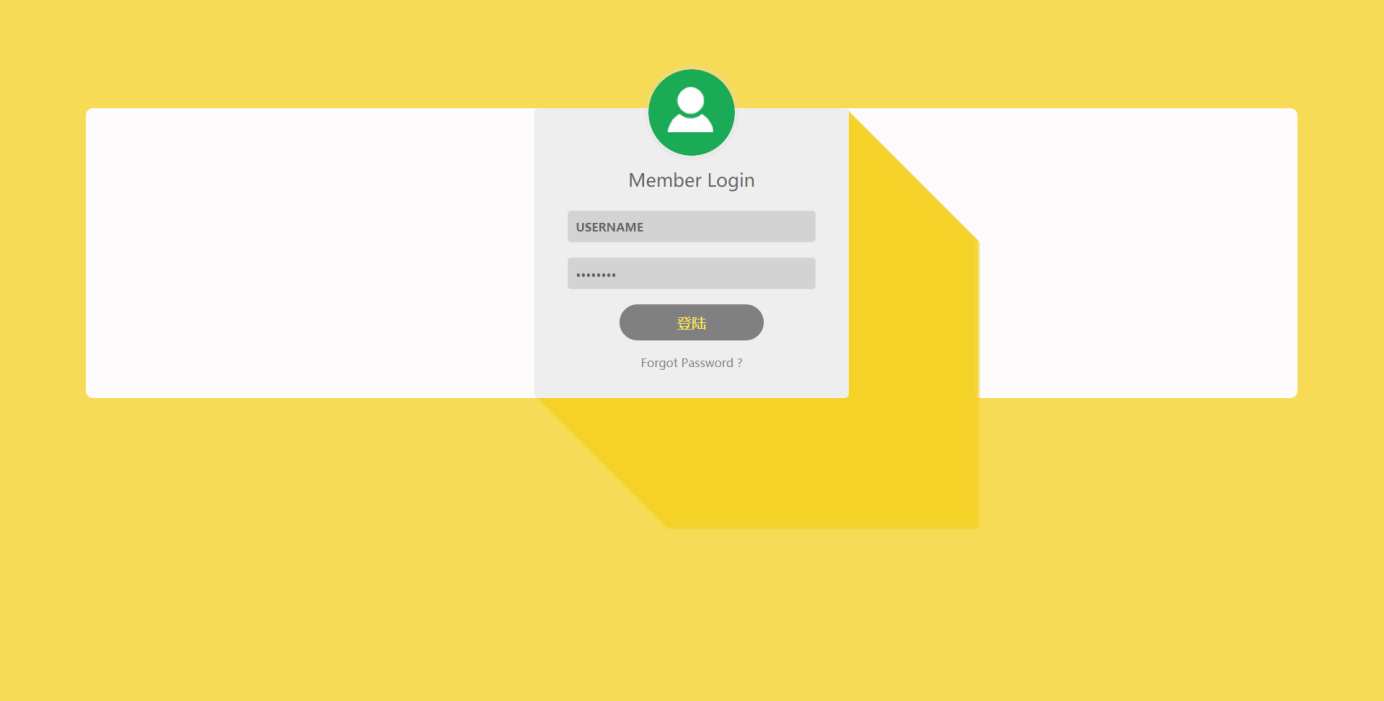


图2. 1 用户登录界面图

### 2.1.2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。

### 2.1.3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) {  
 console.log(req.body);  
 console.log(" 账号： " +req.body.User + " ,密码： " + req.body.Password);  
 var sql = "select \* from tb\_user where User=" + "'" + req.body.User + "'" + " and Password=" + "'" +req.body.Password + "'";  
 console.log(sql);  
  
 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) {  
 //<%=session.getAttribute("username")%>  
 // 设置session传递用户名  
 //session.setAttribute("username", username);  
 return res.send("http://localhost:3000/login\_state/" + req.body.User);  
 } else {  
 return res.send("账号或密码错误");  
 }  
 })  
}

## 2.2 用户提交留言模块

### 2.2.1界面

用户提交留言界面，界面中的设计了一个文本框和两个按钮：“提交”、“匿名提交”，底部显示站内留言；需要用户输入留言内容后，点击其中一个按钮完成提交；用户提交留言界面图如图2.2所示：

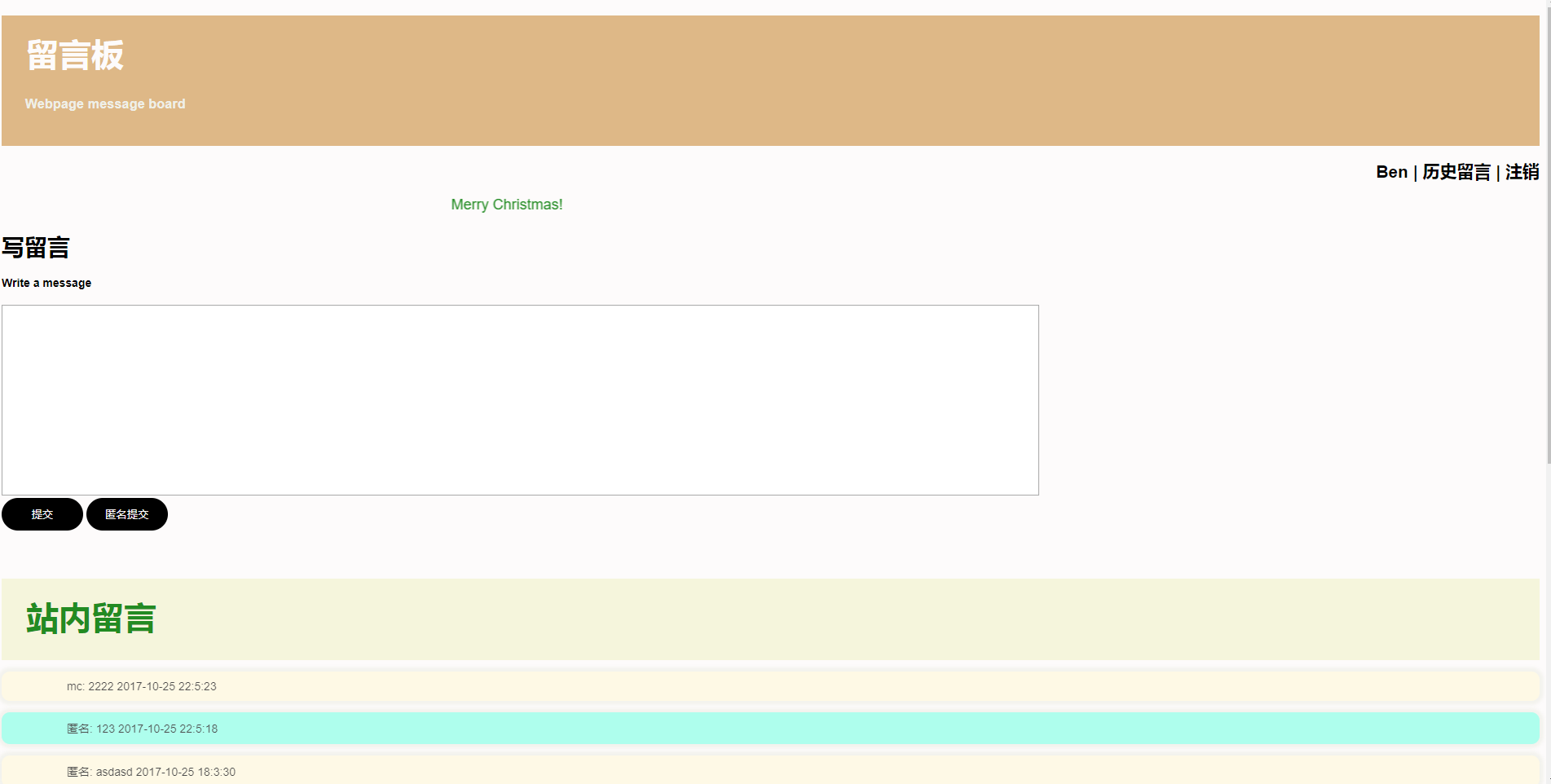


图2. 2 用户提交留言界面图

### 2.2.2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，点击“匿名提交”按钮则为匿名提交留言；右上角有“历史留言”和“注销”的链接，点击后进入相应功能。

### 2.2.3代码及说明

在该功能模块中，主要是进行留言的提交，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.postNote = function (req, res) {  
 console.log(req.body);  
 var note = req.body.Note.replace("'","''");  
 console.log(req.body.Check);  
 var sql = "insert into tb\_notes(User,Note,Date,Param,pass) values('" + req.body.User + "','" + note + "','" + req.body.Date + "','" + req.body.Param +"','"+ req.body.Check + "')";  
 console.log(sql);  
 connection.query(sql, function (err, result) {  
 // if (err) {  
 // return res.send("ERROR");  
 // }  
 return res.send("留言成功！");  
 });  
};

## 2.3 用户查看历史留言模块

### 2.3.1界面

用户查看历史留言界面，界面中显示了该用户个人的全部历史留言，并按时间排序，每条留言后有删除按钮，点击按钮后删除对应的留言；右上角有返回主页的链接；用户查看历史留言界面图如图2.3所示：

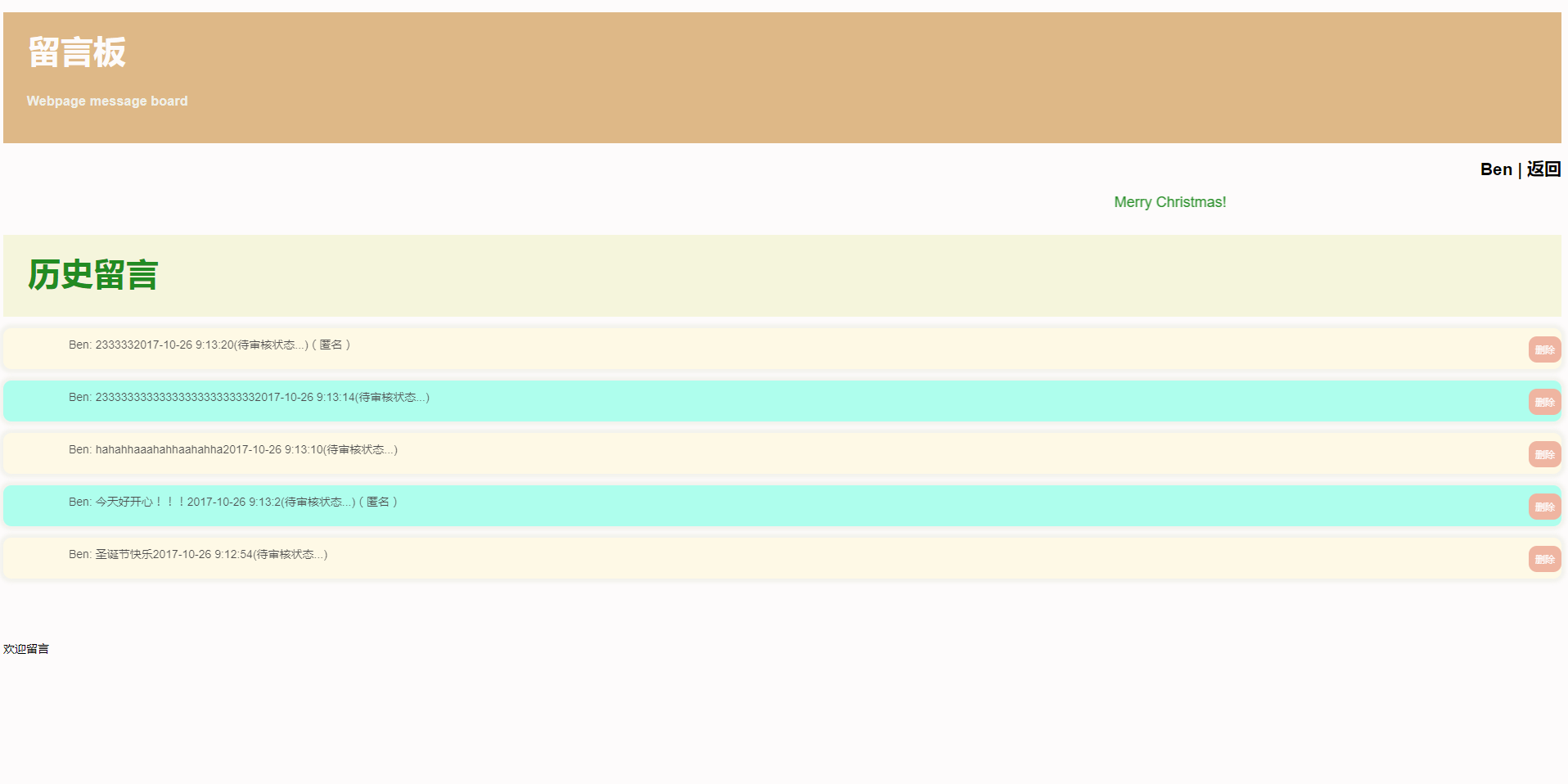


图2. 3 查看历史留言界面

### 2.3.2功能

查看历史留言模块，后台根据用户名字查询数据库，若存在相应的留言则显示出该留言，并显示该留言的审核状态和匿名状态；若不存在，则出现相应提示；查找出来后可以对留言进行删除；右上角有“返回”的链接，点击后返回留言板主页。

### 2.3.3代码及说明

在该功能模块中，主要是进行个人留言的查找和删除，后台接收到数据后，首先判断是否存在此用户的留言；接着将留言显示出来并可执行删除操。查看历史留言接口的主要代码如下：

exports.getNoteByID = function (req, res) {  
 var sql = "select User,Date,Note,Param,pass from tb\_notes where User='" + req.params.id + "' order by Date desc";  
 console.log(sql);  
 var data = {};  
 var date;  
 var getDate;  
 connection.query(sql, function (err, rows, fields) {  
 for (var i = 0; i < rows.length; i++) {  
 date = new Date(rows[i].Date);  
 getDate = date.getFullYear() + '-' + (date.getMonth() + 1) + '-' + date.getDate() + ' ' + date.getHours() + ':' + date.getMinutes() + ':' + date.getSeconds();  
 data[getDate] = {  
 User : rows[i].User,  
 Note : rows[i].Note,  
 pass : rows[i].pass,  
 Param : rows[i].Param  
 }  
 }  
 console.log(data);  
 return res.json(data);  
 });  
};

## 2.4 管理员审核留言模块

### 2.4.1界面

管理员审核留言界面，该界面显示所有待审核的留言，单击“审核通过”按钮则该留言通过审核；点击“删除”按钮则该留言不符合审核要求，对其进行删除操作。管理员审核留言界面图如图2.4所示：

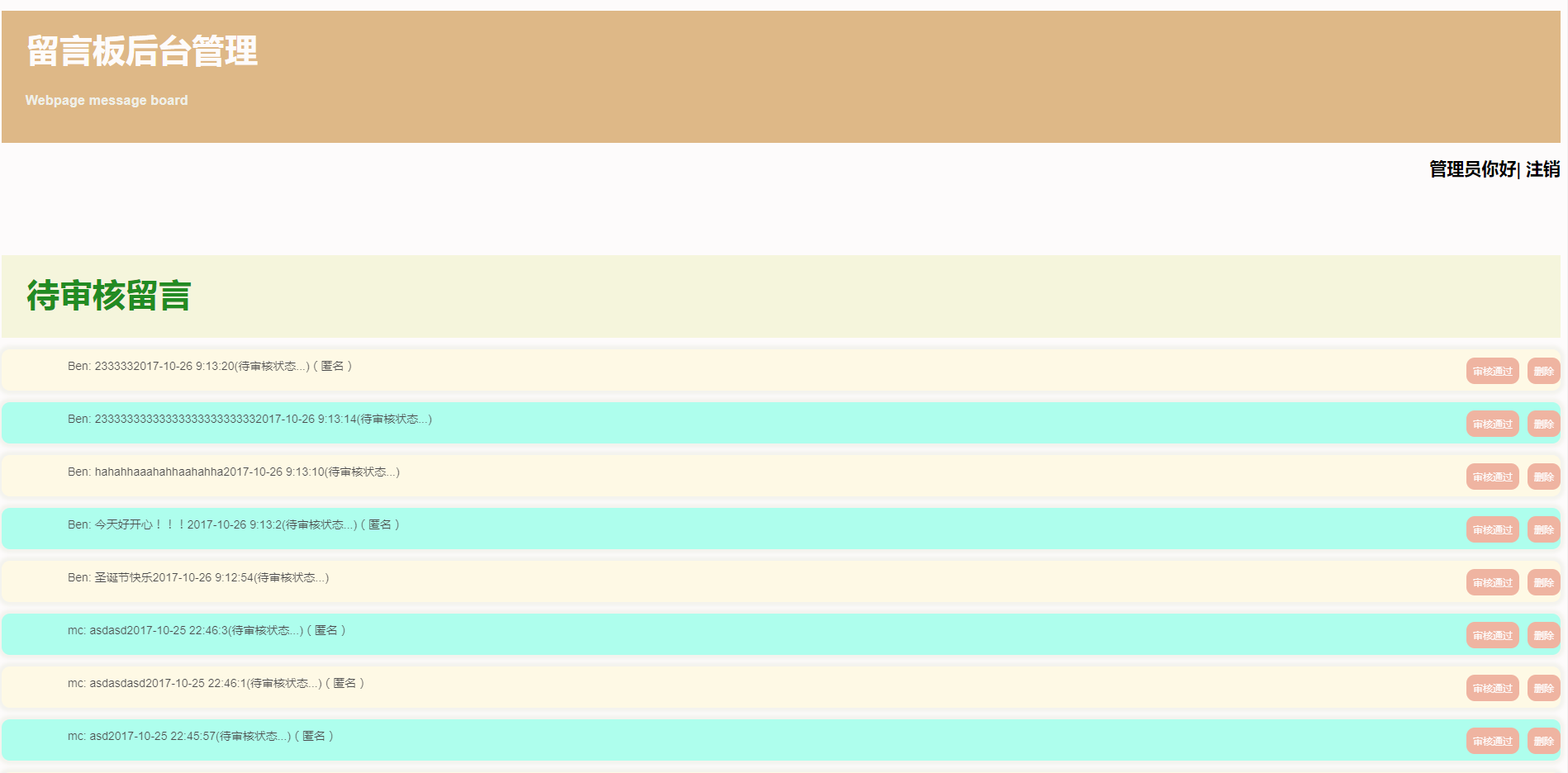


图2. 4 审核留言界面

### 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“通过审核”按钮则该留言审核成功，点击“删除”按钮则删除该该；右上角有“注销”的链接，点击后返回留言板主页。

### 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.updatePass = function (req, res) {  
 var sql = "update tb\_notes set pass=1" + " where User='" + req.params.id + "' and Date='"+ req.params.date +"'";  
 console.log(sql);  
 connection.query(sql, function (err, result) {  
 // if (err) {  
 // return res.send("ERROR");  
 // }  
 return res.send("审核成功");  
 });  
};

exports.deleteNoteByDate = function (req, res) {  
 var sql = "delete from tb\_notes where User='" + req.params.id + "' and Date='" + req.params.date + "'";  
 console.log(sql);  
 connection.query(sql, function (err, result) {  
 // if (err) {co  
 // return res.send("ERROR");  
 // }  
 return res.send("删除成功！");  
 });  
};

# 3 功能测试

## 3.1 测试用例

在经过了对留言板一系列开发设计之后，为了检验留言板的整体功能是否达到预期的设计目标，各个功能模块是否得到预期运行结果，先对留言板进行测试工作，尽可能地发现并排除软件中潜藏的错误，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，测试账号和密码为别为“hrh、123”、“hrh、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“许愿星辰”。

（3）留言审核模块。测试管理员是否能审核留言，用于测试的图片为用户功模块图片上传测试中所上传的图片。

## 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“hrh、123”时，登录成功，输入账号和密码为“hrh、000”时，登录失败；测试时的相关截图如图3.1所示：

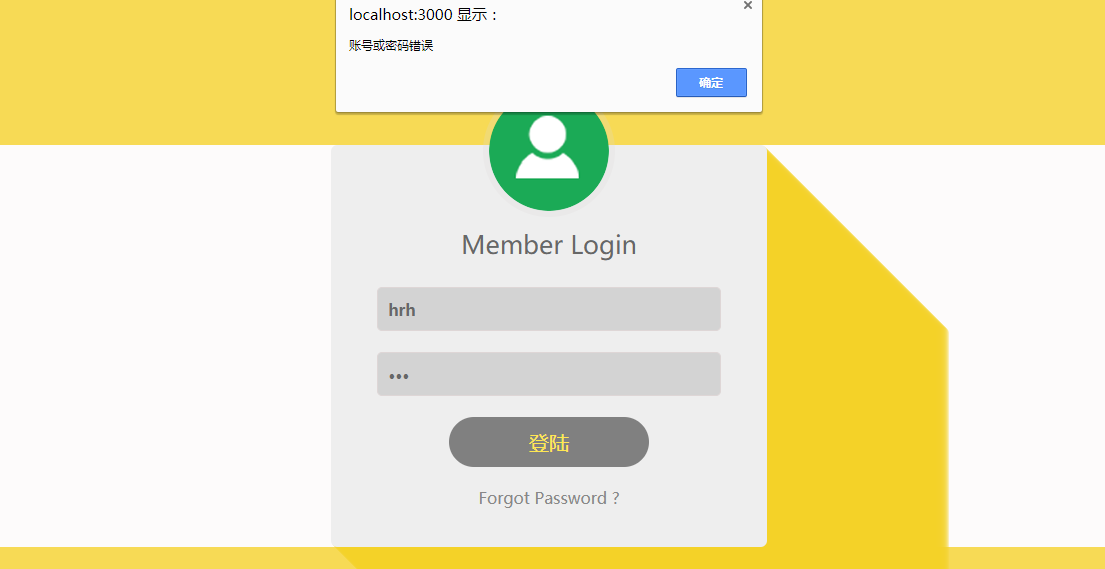


图3. 1 登录模块测试结果图

（2）提交留言：该项测试中，用户在文本框中输入留言为“许愿星辰”，点击“提交”按钮完成测试；测试时的相关截图如图3.2、图3.3所示：

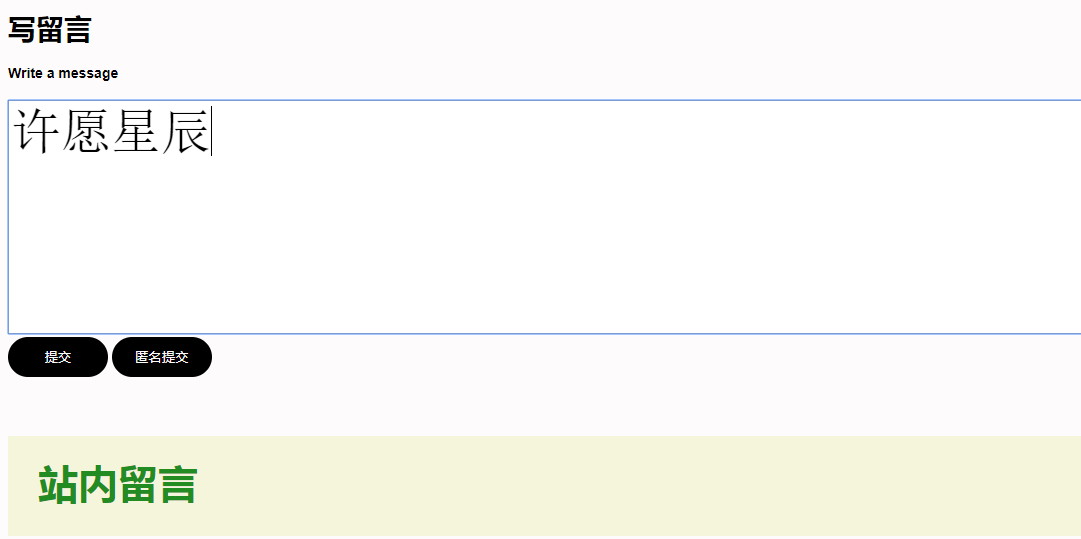


图3. 2 提交留言测试结果图1

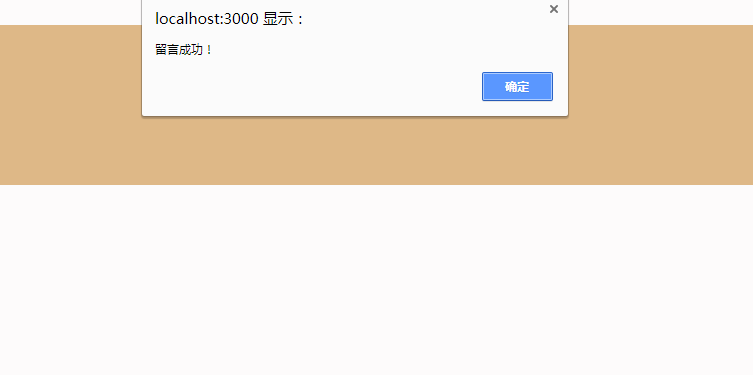


图3. 3 提交留言测试结果图2

（3）审核留言：当管理员审核用户功能模块测试（1）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3.4所示：

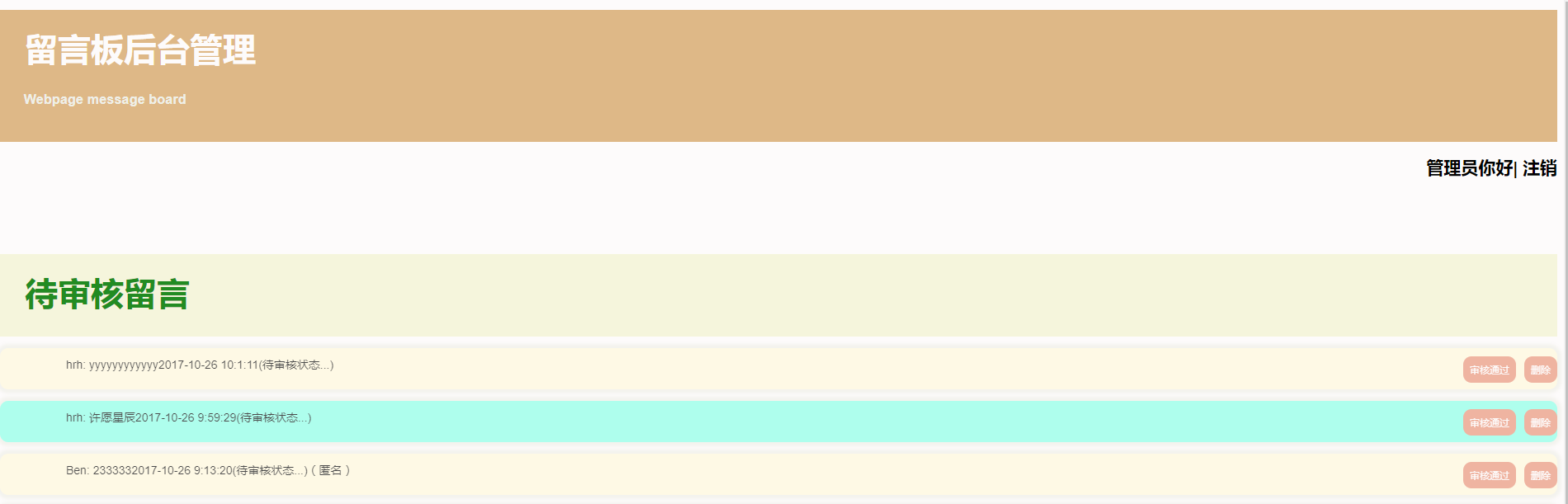


图3. 4 审核留言测试界面

## 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。管理员模块中已完成功能有：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。

在实现Web端各项功能的同时，我掌握了在后台数据库MySQL中建立数据库表和表单的存储过程；学会了使用WebStorm开发工具；以及掌握了在Node.js中利用JS、.ejs和路由设置接口技术的基础知识。通过本次课程设计，我对软件开发的过程有了进一步的认识，提高了自己的动手实践能力，收获颇丰。我认识到在开始一定要提前做好规划，在开发之前一定要明确用户的需求，在做好需求的前提下，再加上详细的设计与计划，同时确保系统功能符合用户需求，这样才可以避免在系统开发过程中出现混乱，并且在对程序进行调试时也能减少附带的错误。

总之，在本次课程设计过程中，我查找了很多书籍和有关资料，并主动和同学讨论在程序中遇到的问题，这使我的编程能力有了不少的提高。由于完成课程设计时间比较紧，加上自身的技术能力有限，所以基于留言板Web端中还有一些不尽如人意的地方，比如系统功能还不是很齐全，系统稳定性比较弱，数据库还不十分安全。同时，由于自身水平有限，系统中还存在少量的不足，系统功能上还是有需要改进的地方，还可以开发更多的功能在界面显示上，也可以做的更简单化，让用户使用起来更加简洁方便，这些不足都将在今后中继续完善。