

合肥工业大学

《网络工程师综合训练》课程设计报告

学 号 2017218007
姓 名 文 华
专 业 班 级 物联网工程 17-2 班
选 题 网络留言簿系统
指 导 老 师 郑淑丽

合肥工业大学

2021 年 1 月

目 录

1	任务描述及设计要求	1
1.1	任务描述	1
1.2	设计要求	1
2	开发环境与工具	2
2.1	硬件	2
2.2	软件	2
3	设计原理	3
3.1	网上留言簿系统简介	3
3.2	系统结构分析	3
3.2.1	B/S 结构	3
3.2.2	C/S 结构	4
3.2.3	B/S 结构与 C/S 结构的比较	5
3.3	开发技术简介	6
3.3.1	Apache 简介	6
3.3.2	PHP 简介	6
3.3.3	HTML 5 简介	7
3.3.4	MySQL 简介	8
3.3.5	PhpStorm 简介	9
3.3.6	Chrome 简介	11
4	系统功能描述及软件模块划分	13
4.1	系统主要功能	13
4.1.1	管理员权限	13

4.1.2	普通用户权限	13
4.1.3	通用权限	13
4.1.4	管理员后台	14
4.1.5	普通用户后台	15
4.2	软件模块划分	15
4.2.1	公共模块	16
4.2.2	数据库模块	19
4.2.3	用户模块	20
4.2.4	注册模块	21
4.2.5	登陆模块	22
4.2.6	留言模块	23
4.2.7	用户管理模块	25
4.2.8	留言管理模块	27
5	设计步骤	28
5.1	数据库设计	28
5.2	创建模块目录	30
5.3	编程与调试	31
5.3.1	编程	31
5.3.2	调试	33
5.3.3	系统测试与部署	33
6	关键问题及其解决方法	37
7	设计结果	40
8	软件使用说明	48
8.1	网上留言簿系统安装	48

8.1.1	软件要求	48
8.1.2	安装步骤	48
9	参考资料	49
10	验收时间及验收情况	50
11	设计体会	51

1 任务描述及设计要求

1.1 任务描述

网上留言簿系统的开发主要包含两部分，一部分是后台的设计与制作，一部分是前台的设计与制作。在后台的设计中，能够实现留言系统模块配置设置选项，例如是否允许游客留言（未实现），是否需要审核（未实现），是否启用验证码，是否启用显示用户信息，是否默认显示等等；能够实现对留言的管理，包括回复留言信息、删除留言信息、批量删除留言信息、批量回复留言信息（未实现）、批量管理所有未回复留言信息（未实现）等。

在前台的设计中，能够实现在线提交留言，要能够记录留言者是谁，留言的内容是什么，留言的时间，留言是否被回复等内容。要能够根据后台配置显示留言的样式，是否分页，是否显示校验码、是否需要审核等信息。

1.2 设计要求

- （1）进行网上留言簿的需求分析和功能设计；
- （2）在数据库中构建数据库、表或视图，熟悉数据库开发流程；
- （3）根据网上留言簿的功能，设计各页面和脚本，掌握动态网页的制作技术。

2 开发环境与工具

2.1 硬件

Dell G3579 笔记本电脑

2.2 软件

操作系统: Windows 10 Pro N for Workstations

平台工具: PhpStorm 2019.3.2、MySQL 8.0.20、Apache 2.4.6、PHP 7.2.34、
Chrome (Version 87.0.4280.66 [Official Build] [64-bit])

编程语言: PHP 7、HTML 5、CSS、JavaScript

3 设计原理

3.1 网上留言簿系统简介

网上留言簿简单说就是一个交流平台，在网络用户交流中起很大的作用，每个人都可以将其个人资料和问题等信息保留在页面上，大家可以相互讨论、相互交流。网上留言簿的操作相对是比较简单的、在您进入网站后、进入创建留言页面都可以看到有输入框，输入后提交即可。网上留言簿不但可以提供完备的信息发布功能，而且在网络用户交流中起到很大的作用、版主可以通过网上留言簿收集反馈信息，也有助于用户通过网络收集信息。因此网上留言簿的存在早已不是什么新奇的事了。本系统主要是考虑到数据库的安全性以及制作网页的优越性，将利用 PHP 编程语言以及 MySQL 数据库制作网上留言簿的系统。

3.2 系统结构分析

3.2.1 B/S 结构

随着网络技术的发展和浏览器功能的完善，上世纪 90 年代末出现了 B/S(Browser/Server)结构，并成为当今 web 开发的主流 B/S 结构是指客户端(不需要专门软件，主要由 Web 浏览器组成)通过 TCP/IP 协议和 HTTP 协议的支持，通过浏览器访问 Web 服务器和后台数据库的体系结构。



图 3.1 B/S 结构示意图

图 3.1 所示为 BS 结构的三层示意图，Web 浏览器作为客户端，可以认为是应用服务器，是表示层，Web 服务器作为应用层，数据库服务器为数据层。客户端可以在不使用任何用户程序的情况下访问系统，完成对数据的查询、添加、删除和修改。系统发现有涉及到数据库的相关指令时，首先将指令发送到 Web 服务器，数据库执行 Web 服务器发送过来的指令，执行完毕后将结果返回 Web 服务器，用户就可以看到结果。

B/S 体系结构的优点：不需要专门的软件，对操作系统也没有要求，只要有浏览器就可以，提高了用户可选择性，降低了客户机的存储空间和内存应用；在客户端实现了零维护，容易进行系统扩展；维护简单，只需要在服务器端进行就可以，不会因为用户规模

影响工作量；具有更好的网络支持，扩展程度高。

当然，B/S 体系结构也有其自身的劣势：服务器负担较重，所有的逻辑业务和数据业务都在服务器端进行，服务器性能成为系统瓶颈；建立在广域网上，面对不可知的用户，安全风险控制能力相对较低。

3.2.2 C/S 结构

各类管理系统在实际应用中，最初是单机型(数据和应用程序在同一台主机上)，随着网络技术发展出现了分布式(数据和应用程序跨越网络节点)，比较有代表性的是 C/S 结构(Client/Server 简写，客户端/服务器模式，图 3.2)并在上世纪八九十年代得到快速发展。在这种模式中，服务器作为数据核心，存储和提供数据服务，客户端(需要安装专门的软件)进行人机交互和逻辑解释，属于典型的两层体系结构。

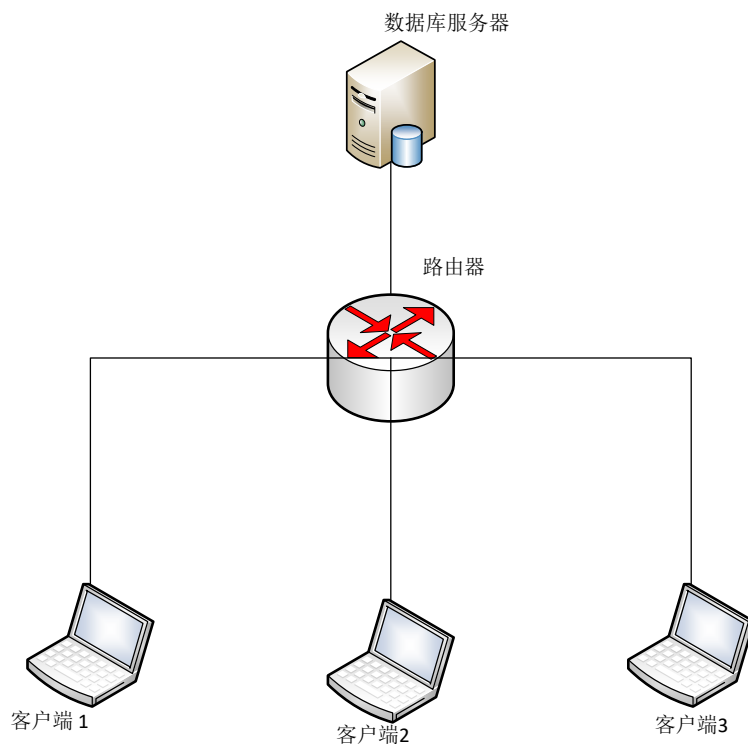


图 3.2 C/S 结构示意图

C/S 结构优点明显：将任务进行分发，在客户端对数据进行处理后提交服务器，响应速度快，有利于充分发挥硬件上的优势；功能分离，数据全部存储于服务器上，客户端只在需要时提取相应的数据，确保了数据的安全性和完整性；客户端是针对性开发，功能完善，界面美观。

由 C/S 结构自身特点决定，其局限性同样明显：维护成本高，系统升级需在所有客户端操作；客户端的软件需要根据客户机系统专门开发，没有通用性，可移植性不高；因为需要安装客户端，用户群相对固定，相对于未知用户的推广度不高。

3.2.3 B/S 结构与 C/S 结构的比较

C/S 和 B/S 结构都有自身的优势和缺点，从技术角度进行比较，如表 3.1 所示。

表 3.1 B/S 和 C/S 结构在技术上的对比

比较项目	C/S	B/S
交互性	一般采用专门局域网，交互性好	有一定交互能力，与 C/S 相比有限
数据一致性	总服务器需要对区域服务器进行数据整合，一致性低	所有数据集中存放，一致性高
实时性	所有数据都是事后数据，不能提供决策	实时数据
数据溯源性	难以查到原始数据	可追溯到客户端原始表单
系统维护速度	需要对每个节点进行维护，速度慢	只进行服务器端维护，速度快
服务器负荷	部分逻辑运算在客户端进行，服务器负荷低	所有逻辑和数据运算都在服务器端进行，服务器负荷高
数据响应速度	服务器和客户端同时工作，响应速度快	数据以表单网页格式传递，响应速度慢
数据安全性	采用局域网网络协议；用户固定，信息安全控制能力强	采用开放性协议，依靠密码管理数据库保证安全；安全控制能力相对弱
网络适用性	只适用于局域网内部用户和宽带用户	任何网络结构，特别是宽带达不到的地方

因本系统属于高效课程设计，对于商用性的要求不高，故在此不在商业运用上对两者进行比较。

综合比较 C/S 和 B/S 两种结构，并结合项目实际，确定系统采用 B/S 体系结构。本系统设计采用“浏览器+Web 服务器+中间层+数据库”多级分布体系结构，来源于对 B/S 体系结构的优化，继承其优点，同时对完善了不足。本项目的系统结构，如图 3.3 所示。

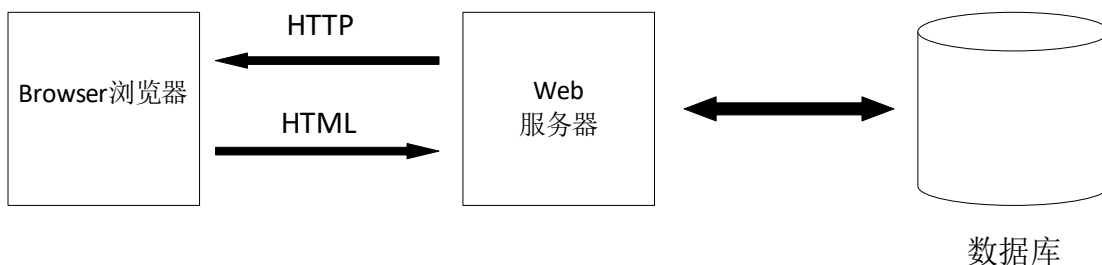


图 3.3 系统结构示意图

3.3 开发技术简介

3.3.1 Apache 简介

Apache 服务器，最早是由伊利诺伊大学香槟分校的国家超级电脑应用中心(NCSA)开发，由开放源代码团队不断维护和加强，现在已经成为主流的 Web 服务器软件。截至 2021 年 1 月 9 日，最新版本为 2020 年 8 月 7 日发布的 2.4.46。Apache 服务器市场占有率目前排在第一位，份额远超微软的 IIS，其主要的优点有如下：

- ① 支持多种计算机平台，可移植性高，几乎支持所有的 Unix，Windows，Linux 平台。
- ② 软件是免费的，建站和后期升级几乎没有成本。
- ③ 系统运行稳定，设置完毕后运行稳定，不会出现假死现象，这也是很多大型的网站才用 Apache 作为服务器的原因。
- ④ 安全性高。系统是开源的，任何人都可以分析代码，查找漏洞并进行维护升级，安全性较高。
- ⑤ 支持语言多，对 ASP，PHP，JSP 都很好的支持。

3.3.2 PHP 简介

PHP（Hypertext Preprocessor，超文本处理器）语言诞生于 1995 年，由 Rasmus Lerdorf 发布。由于代码的开源性，很多程序员对其进行扩展，现在已经发展到 PHP 7。PHP 语言借鉴了 C、JAVA、PERT 等语言的部分语法并有自己的特点，能够使 Web 开发者快速写出动态生成页面的脚本。

PHP 语言具有很多优点，主要有：

- ① 与其他语言比较，相对简单，易于新手掌握，可以嵌入到 HTML 语言中，编程简单，运行快。
- ② 代码是开源免费的，降低成本。
- ③ 由于 PHP 是运行在服务器端的脚本，可以运行在 Unix、Linux、Windows、Mac OS、Android 等平台。
- ④ 在 PHP 7 中，面向对象方面已经完善并不断更新，PHP 完全可以用来开发大型商业程序。
- ⑤ 几乎支持所有的流行数据库。

值得一提的是，PHP 内置 MySQL 函数，支持通过 PHP 语言对数据库进行操作。

3.3.3 HTML 5 简介

HTML5 是 HTML 的最新标准，HTML5 的草案已经于 2008 年发布，目前 W3C(万维网联盟)正在对此进行进一步完善。十多年来，HTML 一直没有进行真正的升级，由于 HTML 和 XHTML 标准已力不从心，它们以文档为中心的理念无法有效地满足现代 Web 应用的需要。1982 年，Tim Berners-Lee 为使世界各地的物理学家能够方便地进行合作研究，创建了 HTML1.0。这个版本只能显示文字，并且仅包含了少量的<TAG>标签。1996 年 W3C 公布了 HTML 的第二个版本 HTML3.2。这个版本扩充了 HTML 语言的特性，增加表格、文字排列和复杂元素显示等功能。一年后，HTML4.0 发布。HTML 技术开始与 CSS 技术相结合，增加了 HTML 语言的表现能力，其改进版本 HTML4.01 一直沿用至今，其发展将被 HTML5 所取代。

HTML5 技术是最近几年才兴起的新网络技术。根据 w3C 的发言稿称：“HTML5 是开放的 Web 网络平台的基石。”

2013 年 5 月 6 日，HTML5.1 正式草案公布。该规范定义了第五次重大版本，第一次要修订万维网的核心语言：超文本标记语言 (HTML)。在这个版本中，新功能不断推出，以帮助 Web 应用程序的作者，努力提高新元素互操作性。

HTML5 是 HTML 技术的最新标准，它的目的是取代 HTML4.0 和 XHTML1.0 标准，适应互联网近几年飞速发展的需要。HTML5 技术的发展，主要包括 HTML5 JavaScript 和 CSS3.0，一般而言，可以理解为 HTML5 标准包含超过三个技术。它存在的意义是为了简化在浏览器的富互联网应用模式的发展不同的公司处理分析的过程。

新版本提供了网页的 HTML 元素，新的属性和行为，从而增加基于互联网浏览器的经验模式，如<视频>和<声音>标签的使用不需要安装插件可以在浏览器中浏览多媒体内容;HTML5 还放弃了 HTML4 标准件，如<字体>标签，这部分功能将实施 CSS3.0。此外，提供了相当多的支持 HTML5 标准开发丰富的互联网应用，如及时的二维图纸，当地的离线存储，将历史记录功能和控制。

HTML5 的不仅仅是一个简单的升级版的 HTML4，这是二十年来的一个大的飞跃，互联网技术的发展。这使得 Web 应用的成熟，平台，互联网的使用模式的变化。

HTML5 是网络的核心语言规范的 HTML，我们看到在浏览器的 HTML 格式。HTML5 主要有原始的视频和音频的离线存储，绘图画布，流支持，几何位置，智能表单和 Web 应用浓度特征。由于没有需要使用一个浏览器插件，Flash 和 Silverlight 相比，这

些丰富的 Web 应用程序(RIA)是速度的先天优势。特别是在移动平台,其硬件性能和传统的电脑有一定的差距,所以速度是更加明显的优势。

3.3.4 MySQL 简介

数据库是整个系统的核心,用来存储用户数据,合理的数据结构设计能够减少数据存储占用的空间,方便管理和查询,有利于提高系统效率。MySQL 是最流行的关系型数据库管理系统,在 Web 应用方面是最好的好的 RDBMS(Relational Database Management System, 关系数据库管理系统)应用软件之一。

MySQL 是一种关联数据库管理系统,关联数据库将数据保存在不同的表中,而不是将所有数据放在一个大仓库内,这样就增加了速度并提高了灵活性。由于其体积小、速度快、总体拥有成本低,尤其是开放源码这一特点,一般中小型网站的开发都选择 MySQL 作为网站数据库。

MySQL 数据库中存在 root 用户和一般用户,不同的用户被赋予不同的操作权限 [91]。在 mysql 5.5 中,对数据库用户提供 5 种不同的操作权限:全局级、数据库级、表级、列级、存储过程和函数级(指定存储过程和函数),权限依次递减。权限划分由系统中存储的六张表进行划分:

① user 表,验证数据库登录,也称为超级用户表,默认 root 用户具有所有数据库操作权限。

② db 表,指定用户对指定的数据库中的对象具有的操作权限,包含用户信息列 (Host, Db, User),权限列 (数据库级权限 19 列)。

③ host 表,作为 db 表的辅助,一般不使用。

④ tables_priv 表,指定用户对某个表格所具有的权限。包含用户信息列 (Host, Db, User, Table_name),权限列 (表级权限 Table_priv, Column_priv 用户所拥有对该表的所有列级权限,如果该列为空,那么 columns_priv 表中用户就没有对该表的列级权限记录),其他 (授权时间 Timestamp, 授权者 Grantor)。

⑤ columns_priv 表,指定用户对某个字段所拥有的权限。包含用户信息列 (Host, Db, User, Table_name),权限列 (column_priv 列级权限),其他列 (授权时间 timestamp)。

⑥ procs_priv 表,指定用户对某一个单独的存储过程或函数。包含用户信息列 (Host, Db, User, Routine_name, Routine_type),权限列 (Proc_priv 存储过程权限),

其他列（授权时间 Timestamp，授权者 Grantor）。

结构化查询语言 SQL（Structured Query Language），由 IBM 圣约翰研究实验室为其关系型数据库管理系统 SYSTEM R 开发。SQL 简单易学，结构简单，功能强大，经推出后得到广泛的应用，像 SQL Server、Oracle、Sybase 这类大型数据库管理系统还是像 PowerBuilder、Visual Foxpro 等数据库开发系统都支持 SQL 作为查询语言。

SQL 完成的功能主要有：查询数据；

在表中插入、修改和删除数据；建立、修改和删除数据对象；控制对数据和数据对象的存取；保证数据库的一致性和完整性。SQL 语言主要包含四部分：

SQL 完成的功能主要有：查询数据；

在表中插入、修改和删除数据；建立、修改和删除数据对象；控制对数据和数据对象的存取；保证数据库的一致性和完整性。SQL 语言主要包含四部分：

- ① 数据定义语言：creat, alter, drop;
- ② 数据查询语言：select;
- ③ 数据操纵语言：insert, update, delete;
- ④ 数据控制语言：commit work, rollback work。

3.3.5 PhpStorm 简介

PhpStorm 是 JetBrains 公司开发的一款商业的 PHP 集成开发工具，旨在提高用户效率，可深刻理解用户的编码，提供智能代码补全，快速导航以及即时错误检查。

PhpStorm 可随时帮助用户对其编码进行调整，运行单元测试或者提供可视化 debug 功能。

（1）PhpStorm 主要功能

i. 智能 PHP 编辑器

- PHP 代码补全；
- 智能的重复编码检测器；
- PHP 重构；
- 支持 Smarty 和 PHPDoc；
- 支持多语言混合。

ii. JavaScript 编辑器

- 基于 DOM/指定浏览器完成；

- 代码导航和用法查找；
- JavaScript 重构；
- JavaScript 调试器。

iii. HTML/CSS 编辑器

- 支持 HTML5；
- 支持 Zend 编码；
- 检验和快速修正；
- 显示应用的外观；

提取嵌入样式。

iv. 轻量级 IDE

- 易于安装；
- 可在 Windows、Mac OS X、Linux 上运行；
- 项目配置简单——可在任意地方打开代码开始工作；
- 性能优先。

v. 智能的环境

- 可视化 Php 单元测试运行期；
- VCS 支持 SVN、Git、Mercurial 等；
- 支持 FTP 和远程文件同步；
- 可记录本地修改；
- 可视化调试；
- 无需任何配置的调试器；
- 支持在 PHP、JS、HTML 中设置断点；
- 观察变量，窗口；
- 批量代码分析；
- 集成分析器。

(2) PhpStorm 优点

- 跨平台。
- 对 PHP 支持 refactor 功能。
- 自动生成 phpdoc 的注释，非常方便进行大型编程。

- iv. 内置支持 Zencode。
- v. 生成类的继承关系图，如果有一个类，多次继承之后，可以通过这个功能查看他所有的父级关系。
- vi. 支持代码重构，方便修改代码。
- vii. 拥有本地历史记录功能（local history 功能）。
- viii. 方便的部署，可以直接将代码直接 upload 到服务器。

3.3.6 Chrome 简介

2008 年，Google 公司以苹果开源项目 Webkit 为内核创建了 Chromium 项目，该项目的目标是创建一个快速、支持众多操作系统的浏览器，并基于 Chromium 发布了自己的浏览器产品 Chrome。Chrome 发布后以其卓越的性能和优异的用户体验迅速占领浏览器市场，和 Microsoft IE、Mozilla Firefox 成为桌面系统上最流行的三款浏览器。目前 Chrome 浏览器已经发展成市场占用率最高的 Web 运行环境，据 2019 年 3 月 NetMarket-Share 数据显示，Chrome 浏览器的市场份额不低于 68.88%，是微软 Edge 的 10 倍以上。Chrome 浏览器的成功得益于其技术上的大胆创新以及对 Web 最新标准的支持。Chromium 作为 Chrome 的开源版本，众多创新技术会议先在 Chromium 上发布，待技术稳定后会应用于 Chrome 浏览器上，当然，Chrome 也会加入一些私有的解码器、音视频支持技术和 Google 服务等 Chromium 没有的功能。图 1 所示为 Chrome 多进程架构及核心模块图。Chrome 主要包含两种进程：主进程和渲染进程。渲染进程包含 JS 引擎、HTML 解释器和 CSS 解释器等模块，负责接收网页内容，对其进行解释执行，并进行布局和渲染，主进程负责将渲染进程的渲染结果进行显示，以及子进程管理、多窗口管理等。

Chrome 支持 Process-per-site-instance、Process-per-site、Process-per-tab 三种多进程策略和 Single process 一种单进程策略，多进程策略适用于 PC 端，不同 Web 应用或网页由多个渲染进程独立渲染，单个 Web 应用或网页的崩溃不会影响到其他进程，多进程策略有利于 Web 运行环境的稳定运行，但这样会导致资源开销大，因此在资源受限的智能终端设备上，一般都采用 Single process 单进程模式，多个 Web 应用或网页的渲染是由单个进程内的不同线程完成的。Chrome 的主体架构，如图 3.4 所示。

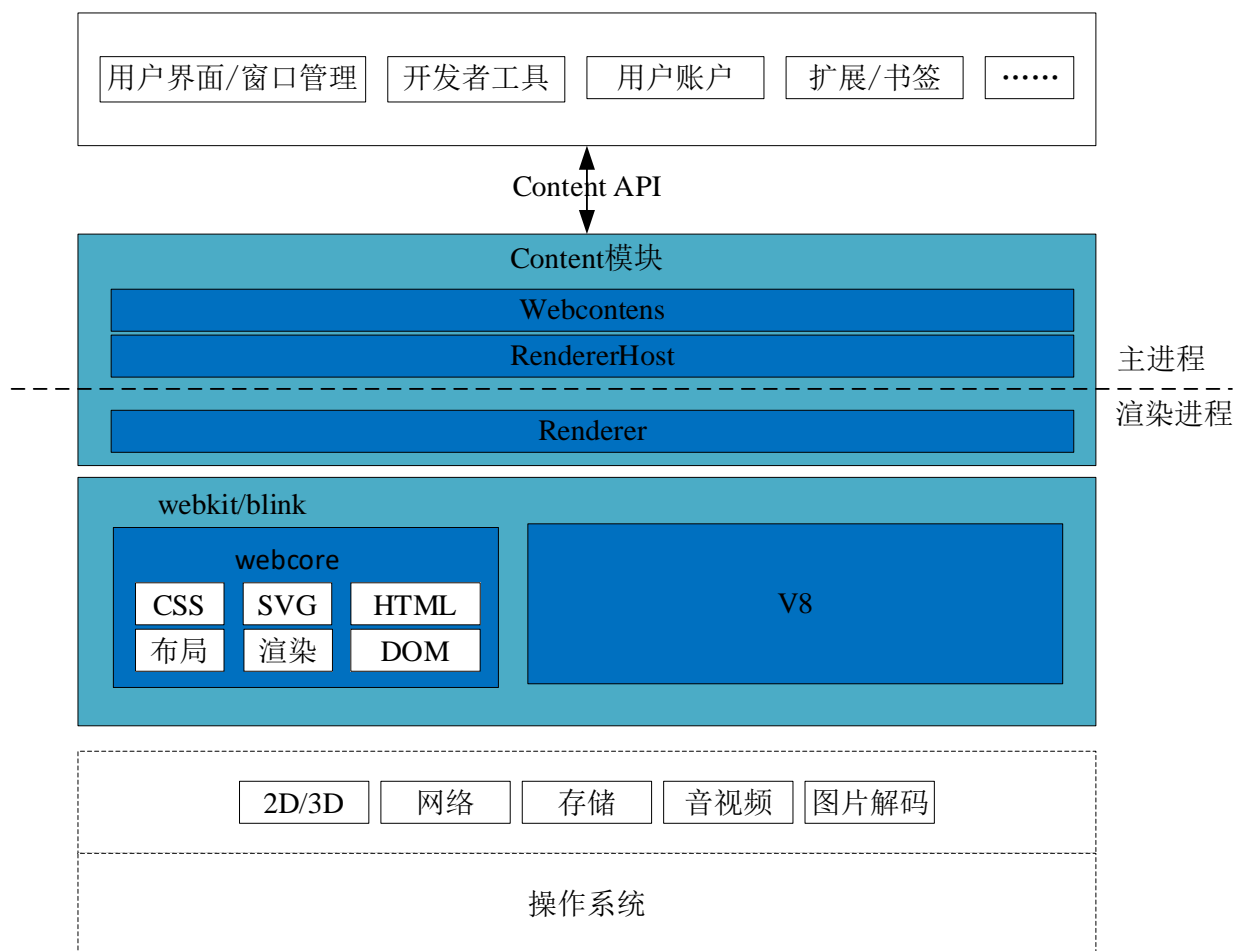


图 3.4 Chrome 架构图

4 系统功能描述及软件模块划分

4.1 系统主要功能

4.1.1 管理员权限

登录

禁言普通用户（可设置禁言期限或永久禁言，见**管理员后台**）

删除普通用户账号（见**管理员后台**）

发表留言（**管理员**发表留言时**不需要**填写**数字字母**验证码）

回复留言（可以回复所有留言）

评论留言

删除留言（见**管理员后台**）

删除评论（见**管理员后台**）

留言内容搜索

时间戳筛选留言

4.1.2 普通用户权限

注册（用户名不能重复、注册邮箱不能重复、填写**数字字母**验证码、不设置邮箱验证）

登录

发表留言（**普通用户**发表留言时**需要**填写**数字字母**验证码）

回复留言（只能回复其他用户给自己的留言）

评论留言

删除留言（只能删除自己的留言）

删除评论（只能删除针对自己留言的评论）

留言内容搜索

时间戳筛选留言

4.1.3 通用权限

留言时可以设置**临时昵称**

留言内容

Please leave your please!

昵称

图 4.1 留言与临时昵称示例

留言时可以设置：所有人可见、登录可见、个人可见

管理员登录后可以查看所有不同隐私权限的留言

普通用户点击留言的昵称时可以查看对应用户的公开信息

管理员点击留言的昵称时可以查看对应用户的所有信息

用户信息包括（每一项都可设置为隐私或者公开）：

用户名（ID）

邮箱

真实姓名

手机

博客

QQ

微信

微博


4.1.4 管理员后台

管理员后台可以查看的信息：

所有注册的用户；

点击某一用户，可以查看其具体信息以及所有留言、回复或评论

发表用户	发表内容	发表性质	发表时间
123	爱吃西红柿	留言	2020-11-08 17:57
123	爱吃芒果	回复（123—>456）	2020-11-08 18:05
123	爱喝牛奶	评论（123—>789）	2020-11-09 12:03

表示用户 A 对用户 B 的回复或者评论

删除留言：

管理员在查看网络留言簿主页时右键可删除留言

管理员也可在后台针对某用户的某一条留言进行删除（支持批量操作）

删除评论：

管理员在查看网络留言簿主页时右键可删除某条留言的评论

管理员也可在后台针对某用户的评论进行删除（支持批量操作）

禁言用户：

管理员在查看网络留言簿主页时右键可禁言用户

管理员也可在后台针对某用户进行禁言（支持批量操作）

删除普通用户账号：

管理员可在后台删除普通用户，删除的内容包括该用户在数据库中的注册信息、留言、回复内容等所有与该用户有关内容（支持批量操作）

4.1.5 普通用户后台

删除留言：

普通用户在查看网络留言簿主页时右键可删除个人留言

普通用户也可在后台针对某个人的某一条留言进行删除（支持批量操作）

删除评论：

普通用户在查看网络留言簿主页时右键可删除个人留言的评论

普通用户也可在后台针对某用户给个人留言的评论进行删除（支持批量操作）

4.2 软件模块划分

根据 4.1 所述的系统主要功能，本项目将系统分为 8 个模块来实现。

系统的结构图，如图 4.2 所示。

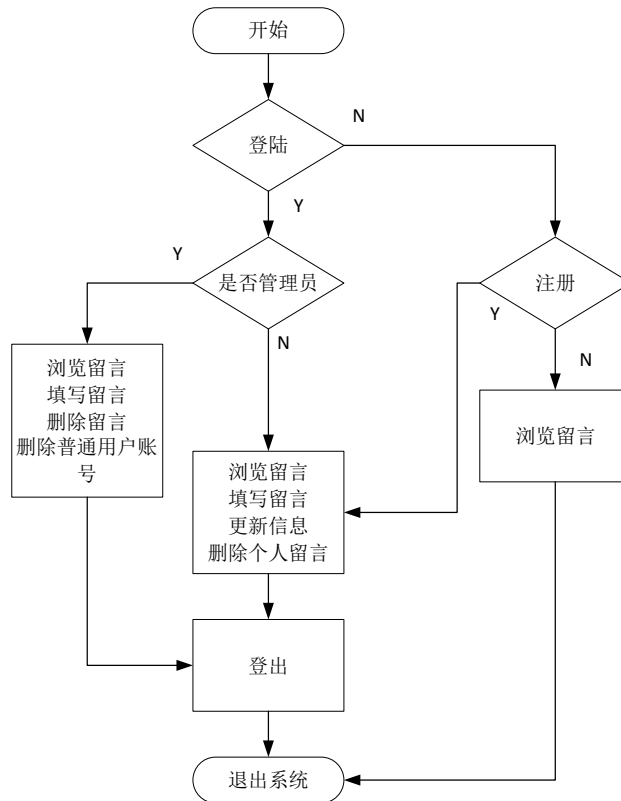


图 4.2 网络留言簿结构

4.2.1 公共模块

这个模块包含了若干个系统管理员与普通用户都可以访问的功能或者对于登陆者和游客都开放的页面。

common.php 的作用是在脚本出错时格式化输出结果、使便于调试，验证来自前端的用户输入是否为 post 提交，验证当前的请求是否为 ajax 请求。具体实现代码如下所示。

```

<?php
function dd($arr)
{
    echo '<pre>';
    var_dump($arr);
    die();
}

function dump($arr)
{
    echo '<pre>';
    var_dump($arr);
}
  
```

```

/**
 * 是否是 POST 提交
 * @return int
 */
function isPost()
{
    return ($_SERVER['REQUEST_METHOD'] == 'POST' &&
(empty($_SERVER['HTTP_REFERER']) || preg_replace("~https?:\/\w+([\^:\/]+).*~i", "\\1",
$_SERVER['HTTP_REFERER']) == preg_replace("~([\^:\/]+).*~", "\\1",
$_SERVER['HTTP_HOST']))) ? 1 : 0;
}

```

```

/**
 * 当前请求是否 ajax 请求
 *
 * @access public
 * @return bool
 * PHP 判断 ajax 请求的原理:
 * 在发送 ajax 请求的时候, 我们可以通过 XMLHttpRequest 这个对象, 创建自定义的
header 头信息。
 * 就是在 header 一层判断是否是 ajax 请求, 对应的 php 根据
$_SERVER['HTTP_X_REQUESTED_WITH'] 判断。
 */

```

```

function isAjax()
{
    return isset($_SERVER['HTTP_X_REQUESTED_WITH']) &&
strtolower($_SERVER['HTTP_X_REQUESTED_WITH']) == "xmlhttprequest";
}

```

header.php 的作用是为增添顶部页面, 根据 session 里存储的数据判定显示哪些重定向的选项。具体实现代码如下所示。

```

<div class="layui-header header header-doc" winter="">
    <div class="layui-main">
        <a class="logo" href="/message_board/index.php">
            留言板
        </a>
        <ul class="layui-nav">
            <?php if ($user):?>
                <li><a href="messages.php">留言管理</a></li>
                <li><a href="user.php">用户后台</a></li>
                <li><a href="profile.php"><? = $user['email']?></a></li>
                <li><a href="logout.php">退出</a></li>
            <?php else:?>

```

```

        <li><a href="login.php">登陆</a></li>
        <li><a href="register.php">注册</a></li>
    <?php endif;?>
</ul>
</div>
</div>
<!--
顶部页面
-->

```

index.php 的作用是根据 session 里存储的数据显示网络留言簿的内容（有些留言是全员包括游客可见，而有些是登陆用户可见，还有仅留言用户个人可见的内容），留言内容的填写与 post 传递（仅限登陆用户），筛选功能的前端窗口与按钮，数字字母验证码机制的实现等。具体实现的核心代码如下所示。

```

if (!$user) {
    $where[] = 'visibility=1';
} else {
    if ($user['admin'] != 1) {
        $where[] = '(visibility in (1,2) || (visibility = 3 and from_user = ' . $user['id'] . '))';
    }
}

```

```

$time = $_GET['time'] ?? '';
if ($time) {
    $start_time = date('Y-m-d 00:00:00', strtotime($time));
    $end_time = date('Y-m-d 23:59:59', strtotime($time));
    $where[] = '(time>=\' ' . $start_time . \' and time<=\' ' . $end_time . '\')';
}

```

```

<table class="layui-table">
    <thead>
        <tr>
            <th>发表用户</th>
            <th width="60%">发表内容</th>
            <th width="10%">发表性质</th>
            <th width="10%">发表时间</th>
            <th width="10%">操作</th>
        </tr>
    </thead>
    <tbody>
        <?php foreach ($message as $v): ?>
            <tr>
                <td><a href="<?=$user ? 'info.php?id=' . $v['from_user'] : 'javascript:;'?>">?>

```

```

$v['nickname'] ? $v['nickname'] : $v['from_user']?></a></td>
    <td width="60%"><?=$v['content']?></td>
    <td><?=$v['type'] == 1 ? '留言' : (($v['type'] == 2 ? '回复' : '评论') . '(' .
$v['from_user'] . '->' . $v['to_user'] . ')') ?></td>
    <td><?=$v['time']?></td>
    <td>
        <?php if ($user && $user['id'] != $v['from_user'] && $v['type'] == 1): ?>
            <a href="javascript:;" data-id="<?=$v['id']?>" class="comment">评论</a>
        <?php endif;?>
        <?php if ($user && ($user['admin'] == 1 || $user['id'] == $v['to_user']) &&
$v['type'] == 3): ?>
            <a href="javascript:;" data-id="<?=$v['id']?>" class="reply">回复</a>
        <?php endif;?>
        <?php if ($user && ($user['admin'] == 1 || $v['from_user'] == $user['id'])): ?>
            <a href="javascript:;" data-id="<?=$v['id']?>" class="delete">删除</a>
        <?php endif;?>
    </td>
</tr>
<?php endforeach; ?>
</tbody>
</table>

```

```

if ($('#canvas').length > 0) {
    code_draw();
    // 点击后刷新验证码
    ($('#canvas').on('click', function() {
        code_draw();
    }));
}

```

4.2.2 数据库模块

这个模块主要包括了系统对数据库的初始化连接的设置。

config.php 的作用是完成数据库的配置。具体实现代码如下所示。

```

<?php
//数据库配置文件
$config = [
    'db' => [
        'database_type' => 'mysql',
        'database_name' => 'message',
        'server' => '127.0.0.1',
        'username' => 'root',
        'password' => '123',
        'charset' => 'utf8',
    ],
];

```

```
]
];
```

4.2.3 用户模块

这个模块包括了用户个人信息的设置与展示，以及管理员对普通用户的管理，如禁言、删除账号等。

profile.php 的主要功能是让用户可以自主设置个人信息以及其可见性，并通过 post 提交给后端，用户修改完成后会有提示信息并刷新页面，接着重定向至主页面。具体实现的核心代码如下所示。

```
if (isPost()) {
    $id = $_POST['id'];
    unset($_POST['id']);
    unset($_POST['email']);

    $database = new \Medoo\Medoo($config['db']);
    $res = $database->update('user', $_POST, ['id' => $id]);
    $user = $database->get('user', '*', ['id' => $id]);
    if (!session_id()) session_start();
    $_SESSION['user'] = json_encode($user);
    $alert = '修改信息成功!';
    //exit(json_encode(['code' => 200, 'msg' => '修改信息成功!']));
}
```

user.php 的主要功能是根据用户对个人信息可见性的设置进行展示，并提供了管理员对普通用户的禁言、删除账户权限。具体实现的核心代码如下所示。

```
<?php foreach ($users as $v): ?>
    <tr>
        <td width="20%">
            <?php if (!$v['admin']): ?>
                <input type="checkbox" name="ids" value="<?=$v['id']; ?>">
            <?php endif; ?>
        </td>
        <td width="10%"><?=$user['admin'] == 1 || $v['id_public'] ? $v['id'] : '私密' ?></td>
        <td width="10%"><?=$user['admin'] == 1 || $v['email_public'] ? $v['email'] : '私密' ?></td>
        <td width="10%"><?=$user['admin'] == 1 || $v['realname_public'] ? $v['realname'] : '私密' ?></td>
        <td width="10%"><?=$user['admin'] == 1 || $v['telephone_public'] ? $v['telephone'] : '私密' ?></td>
        <td width="10%"><?=$user['admin'] == 1 || $v['blog_public'] ? $v['blog'] : '私密' ?></td>
        <td width="10%"><?=$user['admin'] == 1 || $v['qq_public'] ? $v['qq'] : '私密' ?></td>
    </tr>
</foreach>
```



```
' ?></td>
    <td width="10%"><?=$user['admin'] == 1 || $v['wechat_public'] ? $v['wechat'] : '秘密' ?></td>
    <td width="10%"><?=$user['admin'] == 1 || $v['weibo_public'] ? $v['weibo'] : '秘密' ?></td>
    <td width="10%"><?=$v['admin'] == 1 ? '管理员' : '普通会员' ?></td>
    <td width="10%"><?=$v['forbid'] == '-1' ? '永久' : ($v['forbid'] >= date('Y-m-d')) ? $v['forbid'] : ''; ?></td>
    <td width="10%">
        <?php if ($user['admin'] == 1 && $v['admin'] != 1): ?>
            <a href="javascript:;" class="forbid" data-id="<?=$v['id']; ?>">禁言</a>
        <?php endif; ?>
    </td>
</tr>
<?php endforeach; ?>
```

// 禁言

```
$('#forbid').on('click', function () {
    let id = $(this).data('id');
    let title = '禁言用户' + id;
    $('#id').val(id);
    layer.open({
        type: 1,
        title: title,
        content: $('#forbid')
    });
});

$('#delete').on('click', function (e) {
    e.preventDefault();
    var ids = [];
    $('input[name="ids"]:checked').each(function () {
        ids.push($(this).val());
    });
    if (!ids) {
        layer.msg('请选择要删除的用户', {icon: 5});
        return;
    }
}
```

4.2.4 注册模块

这个模块向游客提供了用户注册的窗口，并完成用户注册的一系列工作。

register.php 的主要功能是在游客注册时完成邮箱验证、验证码验证，注册完成后引

导用户修改个人信息等。具体实现的核心代码如下所示。

```
$('#btn-register').on('click', function (e) {
    e.preventDefault();
    var form = $(this).parents('form');
    var data = form.serializeArray();
    var d = {};
    $.each(data, function() {
        d[this.name] = this.value;
    });

    if (!/^[a-zA-Z][0-9](\w|-)+@[a-zA-Z0-9]+\.[a-zA-Z]{2,4}$/i.test(d.email)) {
        layer.msg('邮箱格式不正确', {icon: 5});
        return false;
    }

    if (!d.password) {
        layer.msg('密码不能为空', {icon: 5});
        return false;
    }

    if (d.password != d.confirm_password) {
        layer.msg('两次密码不一致', {icon: 5});
        return false;
    }

    if (d.code != $('#canvas').attr('data-code')) {
        layer.msg('验证码错误', {icon: 5});
        code_draw();
        return false;
    }

    $(this).prop('disabled', true);
    form.submit();
});
```

4.2.5 登陆模块

这个模块辅助完成用户的登陆操作，并完成用户登陆的一切工作。

login.php 的主要功能是在用户登陆时进行账号与密码验证、验证码识别认证，用户登陆后重定向至主页面。具体实现的核心代码如下所示。

```
$('#btn-login').on('click', function (e) {
    e.preventDefault();
    var form = $(this).parents('form');
    var data = form.serializeArray();
    var d = {};
```

```

$.each(data, function() {
    d[this.name] = this.value;
});

if (!d.email) {
    layer.msg('邮箱格式不正确', {icon: 5});
    return false;
}

if (!d.password) {
    layer.msg('密码不能为空', {icon: 5});
    return false;
}

$(this).prop('disabled', true);
form.submit();
});

```

4.2.6 留言模块

这个模块是系统的核心模块，实现了留言内容的写入、展示、发表性质的识别，删除留言，同时识别用户是否具有留言权限等。

message.php 的主要功能是识别用户是否被禁言，留言内容是否符合规范，留言完成提示用户等。具体实现代码如下所示。

```

<?php
//留言内容组件功能模块
require_once 'config.php';
require_once 'common.php';
require_once 'medoo.php';
require_once 'auth.php';

$data = $_POST;

$db = new \Medoo\Medoo($config['db']);
$user = $db->get('user', '*', ['id' => $user['id']]);

// 禁言用户不能留言
if ($user['admin'] != 1) {
    if ($user['forbid'] == -1 || ($user['forbid'] > 0 && $user['forbid'] <= date('Y-m-d'))) {
        exit(json_encode(['code' => 201, 'msg' => '被禁言']));
    }
}

$data['from_user'] = $user['id'];

```

```

$data['time'] = date('Y-m-d H:i:s');

if (!$data['content']) {
    exit(json_encode(['code' => 201, 'msg' => '留言内容不能为空']));
}

if (isset($data['message_id']) && $data['message_id']) {
    $message = $db->get('message', '*', [
        'id' => $data['message_id']
    ]);
    if ($message) {
        $data['to_user'] = $message['from_user'];
    }
}
if (isset($data['code'])) unset($data['code']);
$db->insert('message', $data);
exit(json_encode(['code' => 200, 'msg' => '留言成功']));

```

messages.php 的主要功能是根据留言的可见性与当前用户的身份显示用户留言，留言内容的筛选以及日期筛选，向用户提供删除留言的前端按钮并将请求以 post 方式传递给后端等。具体实现的核心代码如下所示。

```

<?php foreach ($message as $v): ?>
    <tr>
        <td width="20">
            <?php if ((isset($user['admin']) && $user['admin']) || ($user['id'] ==
            $v['from_user'])): ?>
                <input type="checkbox" name="ids" value="<?=$v['id']; ?>">
            <?php endif; ?>
        </td>
        <td><a href="<?=$user ? 'info.php?id=' . $v['from_user'] : 'javascript:;' ?>"><?=$
            $v['nickname'] ? $v['nickname'] : $v['from_user'] ?></a></td>
        <td width="60%"><?=$v['content'] ?></td>
        <td><?=$v['type'] == 1 ? '留言' : (($v['type'] == 2 ? '回复' : '评论') . '(' .
            $v['from_user'] . '->' . $v['to_user'] . ')) ?></td>
        <td><?=$v['time'] ?></td>
    </tr>
<?php endforeach; ?>

```

```

e.preventDefault();
var type = $('input[name="type"]:checked').val();
var date = ($('#date').val());
if (!type) {
    layer.msg('请选择类型', {icon: 5});
    return;
}

```

```

}

if (type == 2 && !date) {
    layer.msg('请选择时间', {icon: 5});
    return;
}

$('.btn-message').on('click', function (e) {
    e.preventDefault();

    $.ajax({
        url: 'message.php',
        type: 'post',
        data: $(this).parents('form').serialize(),
        dataType: 'json',
        success: function (res) {
            if (res.code == 200) {
                window.location.reload();
            } else {
                layer.msg(res.msg, {icon: 5});
            }
        }
    });
});
});

```

4.2.7 用户管理模块

这个模块完成用户个人信息的更新，以及管理员禁言普通用户的操作。

info.php 的主要功能是向用户开放了更新个人信息的权限，更新完成后提示用户信息更新成功并刷新当前页面。具体实现的核心代码如下所示。

```

<form class="layui-form" action="" method="post">
    <div class="layui-form-item">
        <label class="layui-form-label">用户名</label>
        <div class="layui-input-block">
            <input type="text" name="id" class="layui-input" readonly value="<?=$user['admin'] == 1 || $u['id_public'] ? $u['id'] : '私密';?>">
        </div>
    </div>
    <div class="layui-form-item">
        <label class="layui-form-label">邮箱</label>
        <div class="layui-input-block">
            <input type="text" name="email" class="layui-input" readonly value="<?=$user['admin'] == 1 || $u['email_public'] ? $u['email'] : '私密';?>">
        </div>
    </div>
</form>

```

```

</div>
<div class="layui-form-item">
  <label class="layui-form-label">真实姓名</label>
  <div class="layui-input-block">
    <input type="text" name="realname" class="layui-input" readonly value="<?=$user['admin'] == 1 || $u['realname_public'] ? $u['realname'] : '私密';?>">
  </div>
</div>

```

forbid.php 的主要功能是向管理员提供了禁言普通用户的权限，并确定禁言时长。具体实现代码如下所示。

```

<?php
//管理员禁言普通用户功能模块
require_once 'config.php';
require_once 'common.php';
require_once 'medoo.php';
require_once 'auth.php';

$id = $_POST['id'];
$type = $_POST['type'];
$date = $_POST['date'];

if (!$user['admin']) {
    exit(json_encode(['code' => 201, 'msg' => '没有操作权限']));
}

$db = new \Medoo\Medoo($config['db']);
$user = $db->get('user', '*', ['id' => $id]);
if (!$user) {
    exit(json_encode(['code' => 201, 'msg' => '用户不存在']));
}
if ($user['admin']) {
    exit(json_encode(['code' => 201, 'msg' => '不能禁言管理员']));
}

if ($type == 1) {
    $data['forbid'] = '-1';
} else {
    if (!$date) {
        exit(json_encode(['code' => 201, 'msg' => '请输入时间']));
    }

    $data['forbid'] = $date;
}

```

```
$db->update('user', $data, ['id' => $id]);
exit(json_encode(['code' => 200, 'msg' => '禁言成功']));
```

delete.php 的功能之一是给予了管理员删除删除用户账号的权限，处理前端 post 传递来的删除请求，删除用户完成后刷新当前页面。具体实现的核心代码如下所示。

```
<?php
//管理员删除普通用户及其留言功能模块
require_once 'config.php';
require_once 'common.php';
require_once 'medoo.php';
require_once 'auth.php';
$data = $_POST;
$id = $data['ids'];
$table = $data['table'];
$db = new \Medoo\Medoo($config['db']);
if ($table == 'user') {
    $db->delete($table, [
        'AND' => [
            'id' => $id,
            'admin[!]' => 1
        ]
    ]);
```

4.2.8 留言管理模块

这个模块的作用是给予用户自主管理留言的权限，处理前端传递来的留言删除 post 请求。

delete.php 的功能之一是允许用户删除自己的留言（留言、回复或评论），删除留言成功后刷新当前页面。具体实现的核心代码如下所示。

```
$db->delete('message', [
    'OR' => [
        'form_user' => $id,
        'to_user' => $id,
    ]
]);

else if ($table) {
    $db->delete('message', [
        'id' => $id
    ]);

exit(json_encode(['code' => 200, 'msg' => '删除成功']));
```

5 设计步骤

5.1 数据库设计

根据系统分析和功能上的需要，规划出留言本中数据表的实体 E-R 图。用户信息实体，包括用户 ID、邮箱、姓名、手机号、博客、QQ、微信、微博、用户 ID 是否可见、邮箱是否可见、姓名是否可见、手机号是否可见、博客是否可见、QQ 是否可见、微信是否可见、微博是否可见、密码、是否管理员标识、是否被禁言标识。用户信息 E-R 图，如图 5.1 所示。留言信息实体，包括留言用户 ID、留言用户昵称、留言内容、留言时间、留言可见性、类型、留言内容 ID。留言信息 E-R 图，如图 5.2 所示。评论信息实体，包括评论内容用户 ID、评论内容昵称、评论内容、评论者、回复者、评论时间、类型、评论内容 ID。评论信息 E-R 图，如图 5.3 所示。回复信息实体，包括回复留言用户 ID、回复用户昵称、回复内容、评论者、回复者、回复时间、类型、回复内容 ID。回复信息 E-R 图，如图 5.4 所示。

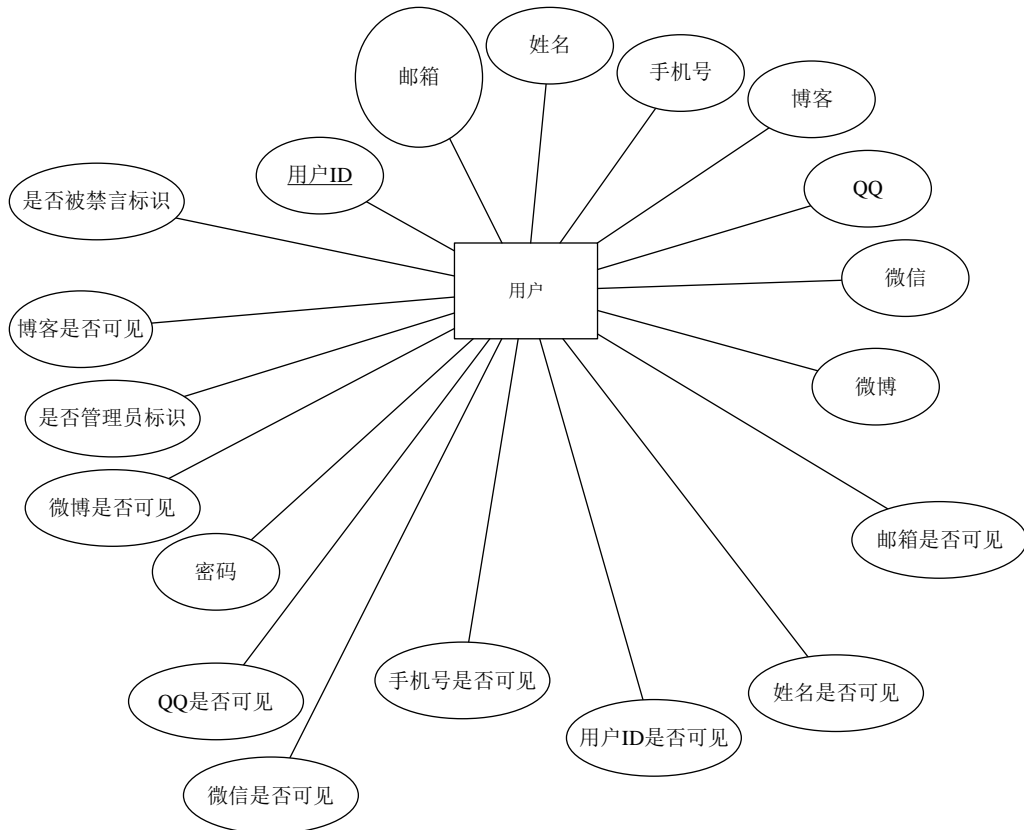


图 5.1 用户信息 E-R 图

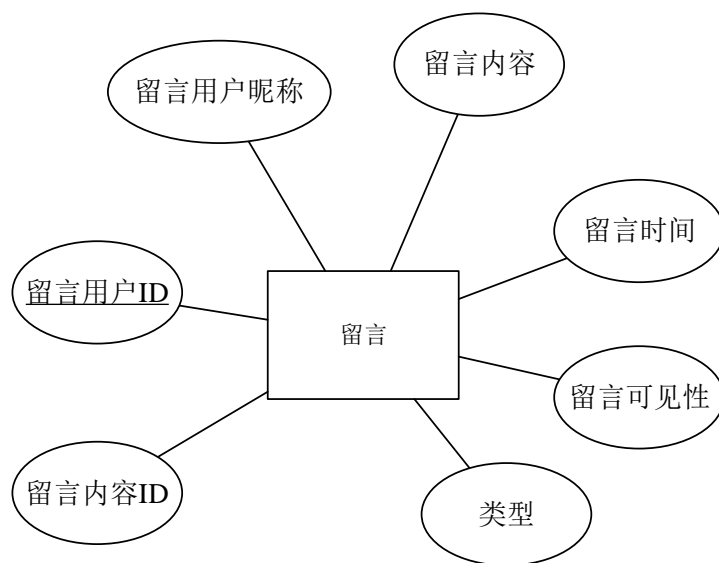


图 5.2 留言信息 E-R 图

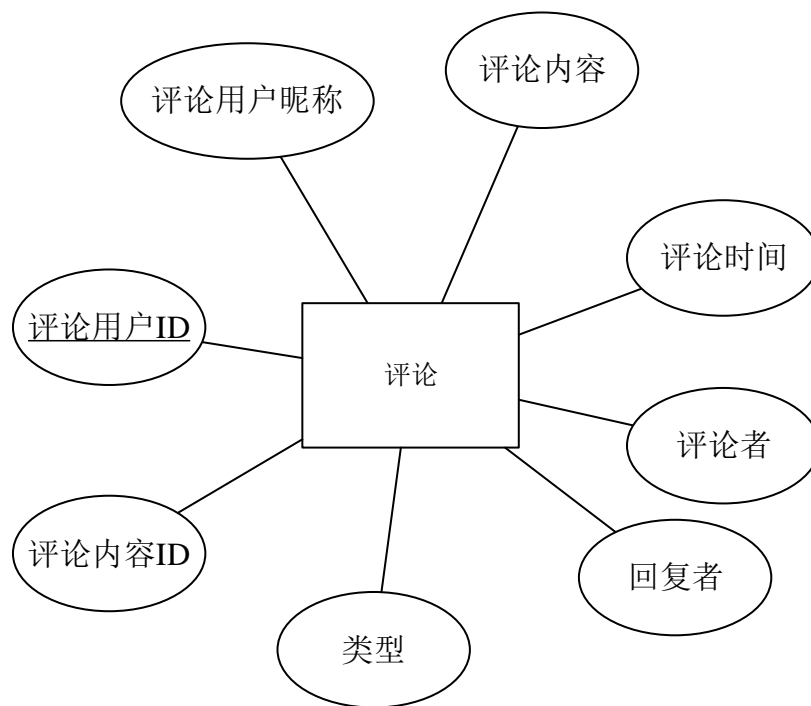


图 5.3 评论信息 E-R 图

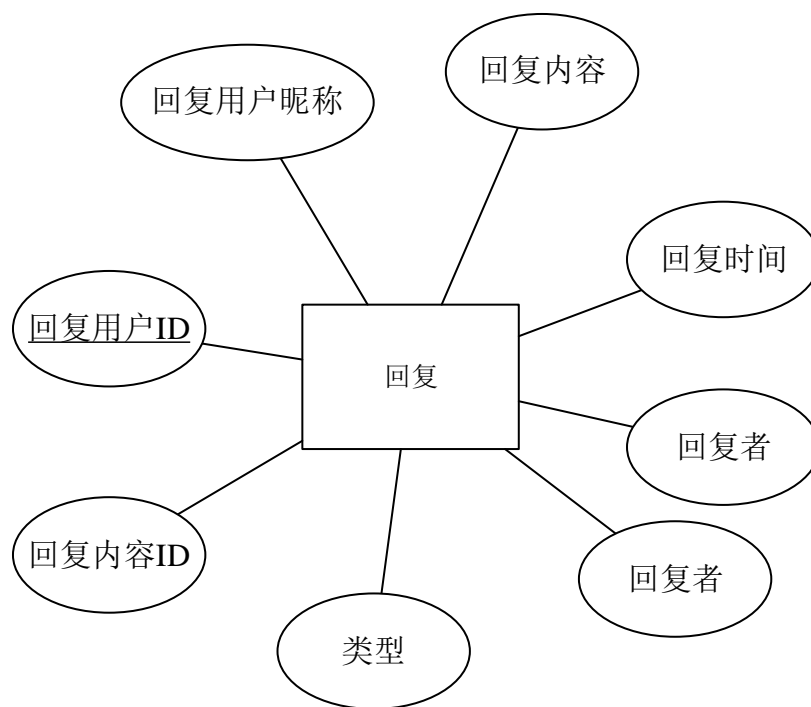


图 5.4 回复信息 E-R 图

5.2 创建模块目录

创建好数据表后进行模块的创建，每一个独立的内容都可以称之一个模块，如：公共模块、数据库模块、用户模块、注册模块、登陆模块、留言模块、用户管理模块、留言管理模块等。每个模块都可放置在单独的目录下，因本系统开发工作量较小，故所有模块均放置在同一目录下。系统开发的模块目录，如图 5.5 所示。

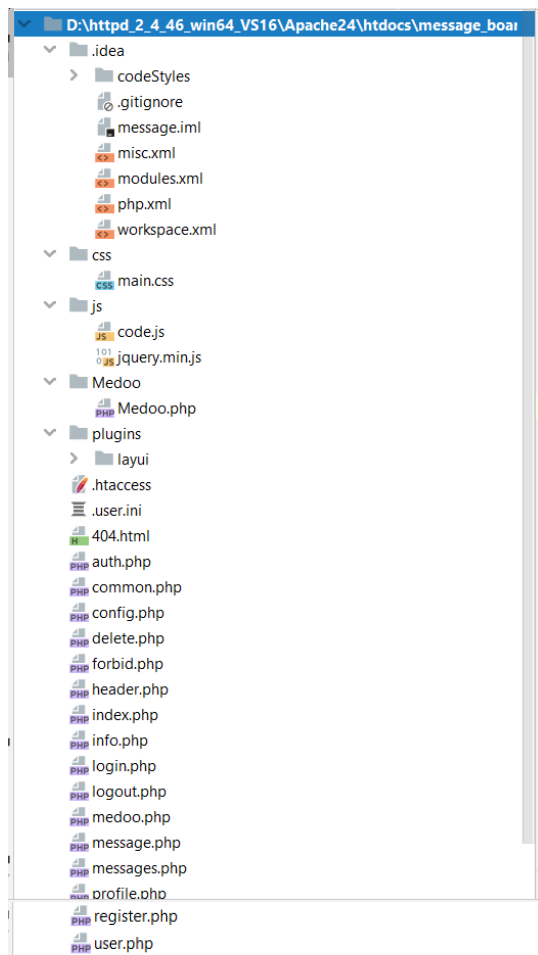


图 5.5 系统开发模块

5.3 编程与调试

在完成数据库设计与模块创建后，开始使用 PHP 进行编程实现，编程结束后对系统进行测试。

5.3.1 编程

系统开发所使用的类与方法，如图 5.6 所示。本项目所开发的网络留言簿模块有两个控制器，一个是涉及前台功能的前台页面控制器，默认的调用控制函数为 `init()`，控制器的名称为 `index.php`。另外还有一个是涉及后台功能的后台管理控制器，默认的调用控制函数为 `init()`，控制器的名称为 `message.php`。控制器的开发是关键也是难点，现在主要针对前台控制器(`index.php`)开发作详细的介绍。

前台控制器主要涉及前台留言显示、在线留言的提交处理等功能函数。在程序设计时，一定要考虑程序的健壮性和安全性，要考虑到脚本被调用所引起的被入侵，植入的现

象，所以在程序的开始加入 `echo '<pre>';var_dump($arr);die();` 代码（即 `index.php` 的第 3 行代码 `require_once 'common.php';`）。

判断一下下面的这段程序是否处于 `ph-pcms` 的运行范围，如果在运行范围，则继续运行本段程序，如果不在运行范围，则给出提示信息并结束程序的运行。

留言信息的处理是网络留言簿系统中最关键的一部分，主要是留言页面的信息是如何存入数据库，数据库的信息怎样能在网络留言簿前端显示，并且能够根据条件进行查询。这些功能可以写在提交留言的处理函数 `insert` 中。首先当点击留言按钮的时候要判断留言是真的要提交么，如果是，直接调教留言事件，如果不是，返回并给出提示信息“您没有登录，或者没有权限，请您通过正确的方式提交”。如果是正常的留言，先 `if ($('#code').val() != $('#canvas').attr('data-code'))` 来判断一下传递过来的变量是否存在，存在是说明页面上应用了验证码，然后调出验证码的程序进行验证。这个变量不存在，说明页面上没有应用验证码，继续往下一步执行。

在 `PHP 7` 中验证码的传递是通过 `session` 的，主要是将页面提交上来的验证码同存贮在 `session` 中的验证码值进行比较，如果相同，则提示验证码正确，如果不同，则提示验证码不正确。由于 `PHP 7` 是支持多站点的，模块可以应用到每一个网站上，因此一定要记录要在模块中有站点的记录，用来区分到底是哪个站点在调用这个模块，消息是哪个站点传递的。另外，用户名和用户 `ID` 的记录也是必要的，因为要记录留言者的信息。

最后，利用程序调用 `medoo` 框架中的函数，提取表单里与留言相关信息，然后经其后台的处理直接存入数据库中。相应的留言信息也就存入 `message` 表中，留言信息被记录。如果需要调用留言信息，或者对留言信息进行查询、更改、删除等操作，直接操作数据库中的相关信息就可以了。留言的显示模版和控制器的定义两者是密切相连的，两者是息息相关的，在显示模块的定义时一定要考虑到后台的控制器的设置，在控制器的设置中有关于前台显示的控制，例如是否开启留言设置，是否显示验证码，是否显示提交表单等等。

后台控制器的开发，也是调用 `medoo` 框架中的函数完成的，在此不赘述。



图 5.6 系统的类与方法

5.3.2 调试

代码调试与编程实现几乎是并行的，在实现后，着手对实现效果进行测试，如果同预期有出入，则需要反复调试代码，直至成功。调试的既需要技巧，也需要细致的安排，这就要求能够做到尽量完善的测试。

5.3.3 系统测试与部署

(1) 系统测试环境

系统测试环境应当尽量与使用环境一致，避免因环境差异造成的测试误差，使一些 bug 无法发现。表 5.1 和表 5.2 分别是系统对客户端和服务器的软硬件要求。

表 5.1 系统测试的硬件环境

硬件名称	详细要求
服务器	CPU 2 Ghz 以上，内存 2G 以上，硬盘 300G 以上
客户端	具有网络功能的 PC
网络	基于单位内部局域网，含路由交换等设备

表 5.2 系统测试的软件环境

软件名称	详细要求
服务器端操作系统	Ubuntu 18.04
服务器端数据库系统	MySQL 8.0
服务器端应用程序服务器	Apache 2.4
服务器端脚本解释器	PHP 7.2
客户端操作系统	Windows 10
客户端浏览器	Chrome 86 及以上版本

(2) 系统测试计划

① 功能测试阶段。对于网络应用软件系统来说，其功能测试阶段一般分为用户连接测试、界面测试、链接测试和数据库测试等方面。用户连接测试用于测试是正确用户信息能否登入系统，错误用户信息是否被拒绝登陆。界面测试主要分为两个方面，一方面，各个窗口页面实现功能是否与功能模块划分对应，是否能准确完成页面功能；另一方面，测试页面所包含的所有元素，例如按钮、图表和文字等是否正确显示，页面提交后表格数据是否正常，与设计初衷是否保持一致等。数据库测试用于检测页面操作与数据库的同步，检查页面提交内容是否如实记录在数据库表中，数据库表之间的信息是否一致等。在这一阶段，可以请实际应用系统的人参与测试，体验系统操作界面和操作流程，提出改善意见，完善系统开发。

② 集成测试阶段。集成测试阶段主要分为三个部分，集成测试阶段需要完成功能集成测试、模块内部的集成测试和整个系统集成测试三个部分的测试任务，由于三者是相互关联甚至互相包含的，因此，在实际测试过程中，可能一个测试用例会包含三个测试部分的内容，无法严格区分单独用于某一个部分的测试用例。

功能界面集成测试指在单个界面功能测试的基础之上，增加测试互相影响的数据是否正确的测试用例进行测试。模块集成测试指模块通过了各自的集成测试，按模块组合执行测试用例，在覆盖界面操作功能的同时，加入各个界面之间跳转以及数据存储逻辑的测

试用例进行测试。系统集成测试按系统组合执行测试用例，基于模块集成测试的基础，加入不同模块之间业务逻辑关联的测试用例进行测试。

在上一阶段的测试工作中，界面功能测试主要着眼于单个页面的图表、按钮等的测试，而在功能界面的集成测试中，加入了不同界面相互之间可能有影响的测试用例。模块内的集成测试主要在同一业务逻辑处理模块的内部，完成各自的集成测试内容，不涉及与其他模块之间的操作相关的测试内容。系统整体集成测试主要关注各个模块之间的相互影响，按照模块之间的松散耦合相关性，以及系统需求分析和设计中的业务操作流程，设计响应的测试用例，验证系统的正确性。

与系统功能测试阶段一样，系统集成测试也采用黑盒的方式，该方式将系统认为是密闭的盒子，不需要关系内部结构，只看是不是能正确执行系统功能。

(3) 测试结果

基本的功能测试表如表 5.3 所示。

表 5.3 功能测试表

序号	测试方法和过程描述	预期结果	实际结果
1	管理员用户登录，输入正确的用户名、密码	成功	成功
2	管理员登录，输入错误的用户名、密码	失败	失败
3	普通用户登录，输入正确的用户名、密码	成功	成功
4	普通用户登录，输入错误的用户名、密码	失败	失败
5	切换左侧菜单，看右侧页面是否显示正常	正确	正确
6	修改密码，两次输入密码不一致	失败	失败
7	修改密码，两次输入密码一致	正常	正常

在测试过程中，每项操作都给出了预期结果和实际结果，当两者不一致时，系统出现错误，需要进行调试。

测试过程中，预期结果和实际结果一致。通过测试发现系统界面友好，易于上手，提高了用户查看留言以及进行相关操作的效率。

表 5.4 是系统集成测试表。

表 5.4 系统集成测试表

测试功能	测试具体描述	预期结果	正确
管理员设置某用户不能登录系统	管理员设置某用户不能登录系统，	正确	正确

	查看数据库中的表数据是否正确		
	该用户登录系统，能否登入	失败	失败
仅登陆可见留言	普通用户登陆查看留言	正确	正确
	游客查看留言	正确	正确
填写/评论/回复留言	普通用户填写留言不输入验证码	失败	成功
	用户评论/回复留言	成功	成功

从表中可以看到，系统测试有预期结果和实际结果不一致的地方，需要进行调试，确保系统的正确性。经过修改程序反复调试后解决了系统 bug。通过功能测试和集成测试，发现系统在操作界面和数据方面不存在太大问题，可以投入使用。

测试完成后，直接部署到服务器即可。

6 关键问题及其解决方法

1、注册与登陆的问题。

解决方法：

注册：

- 状态判断：如果是已登录状态（通过 COOKIE 验证），该页面跳转至网站首页；否则显示该注册页面。
- 表单填写：前端页面一个 Form，主要包括邮箱、密码、重复密码、验证码等。
- 前端判断格式：通过 JavaScript 正则表达式判断邮箱是否符合邮箱格式、密码是否符合一定长度和难度等。
- 判断是否已注册：AJAX 判断该邮箱是否已被注册，如果是，则提示登录；否则继续。
- 表单提交：POST 表单。
- 后台判断格式：防止前端 JavaScript 被修改或者直接 POST 数据，需要后台再次验证数据格式。
- 插入数据库：在表单各数据正确无误的情况下将新用户的相关信息插入数据库（密码需加密处理，不能明文显示），并返回新用户的用户 ID 等相关资料。
- 跳转到登录状态：通过用户 ID 等资料给用户添加 COOKIE 保持其登录状态。

登录：

- 状态判断：如果是已登录状态，该页面跳转至网站首页；否则显示该登录页面。
- 表单填写：前端一个 Form，包括邮箱和密码。
- 前端判断格式：判断邮箱是否符合邮箱格式、密码是否符合长度等。
- 表单提交：POST 表单。
- 后台匹配：查找是否存在该用户；匹配该用户密码；返回是否匹配。
- 页面跳转：匹配成功，则添加 COOKIE 保持登录状态；否则提示相关错误，并跳回登录状态。

2、前端留言数据如何与数据库保持高效同步？

解决方法：

使用 medoo 框架的相关 API，增删查改的 `inser`、`delete`、`selectContext (query)`、`update` 等方法都进行了封装与优化，较之直接操作效率更高。

3、在服务器负载一定情况下，系统如何快速响应？

解决方法：

也是使用 medoo 框架的相关 API，medoo 不论是代码体量或是允许内存都很小，非常适合服务器负载教宝贵的用户。

4、如何实现留言信息的内容筛选与时间戳筛选？

解决方法：

还是使用 medoo 框架的相关 API，`selectContext (query)` 查找起来简直不要太香。

5、验证码机制如何实现？

解决方法：

简单地用 CSS 设置一下验证码框背景，再用 JavaScript 实习一个 `code_draw` 函数，其主要功能就是产生一个含字母与数字的随机字符串。具体实现代码如下所示。

```
function code_draw() {
    var canvas_width = $('#canvas').width();
    var canvas_height = $('#canvas').height();
    var canvas = document.getElementById("canvas"); //获取到 canvas 的对象，演员
    var context = canvas.getContext("2d"); //获取到 canvas 画图的环境，演员表演的舞台
    canvas.width = canvas_width;
    canvas.height = canvas_height;
    var sCode = "A,B,C,E,F,G,H,J,K,L,M,N,P,Q,R,S,T,W,X,Y,Z,1,2,3,4,5,6,7,8,9,0"; //取值范围
    var aCode = sCode.split(",");
    var aLength = aCode.length; //获取到数组的长度
    var value = [];
    for (var i = 0; i <= 3; i++) {
        var j = Math.floor(Math.random() * aLength); //获取到随机的索引值
        var deg = Math.random() * 30 * Math.PI / 180; //产生 0~30 之间的随机弧度
        var txt = aCode[j]; //得到随机的一个内容
        value[i] = txt.toLowerCase();
        var x = 10 + i * 20; //文字在 canvas 上的 x 坐标
        var y = 20 + Math.random() * 8; //文字在 canvas 上的 y 坐标
        context.font = "bold 23px 微软雅黑";

        context.translate(x, y);
        context.rotate(deg);
    }
}
```

```

    context.fillStyle = code_randomColor();
    context.fillText(txt, 0, 0);

    context.rotate(-deg);
    context.translate(-x, -y);
}
// 将生成的值以属性的方法添加到元素
value = value.join('');
$('#canvas').attr('data-code', value)
//验证码上显示线条
for (var i = 0; i <= 5; i++) {
    context.strokeStyle = code_randomColor();
    context.beginPath();
    context.moveTo(Math.random() * canvas_width, Math.random() * canvas_height);
    context.lineTo(Math.random() * canvas_width, Math.random() * canvas_height);
    context.stroke();
}
//验证码上显示小点
for (var i = 0; i <= 30; i++) {
    context.strokeStyle = code_randomColor();
    context.beginPath();
    var x = Math.random() * canvas_width;
    var y = Math.random() * canvas_height;
    context.moveTo(x, y);
    context.lineTo(x + 1, y + 1);
    context.stroke();
}
}

function code_randomColor() { //得到随机的颜色值
    var r = Math.floor(Math.random() * 256);
    var g = Math.floor(Math.random() * 256);
    var b = Math.floor(Math.random() * 256);
    return "rgb(" + r + "," + g + "," + b + ")";
}

```

7 设计结果

本部分主要以截图介绍为主。

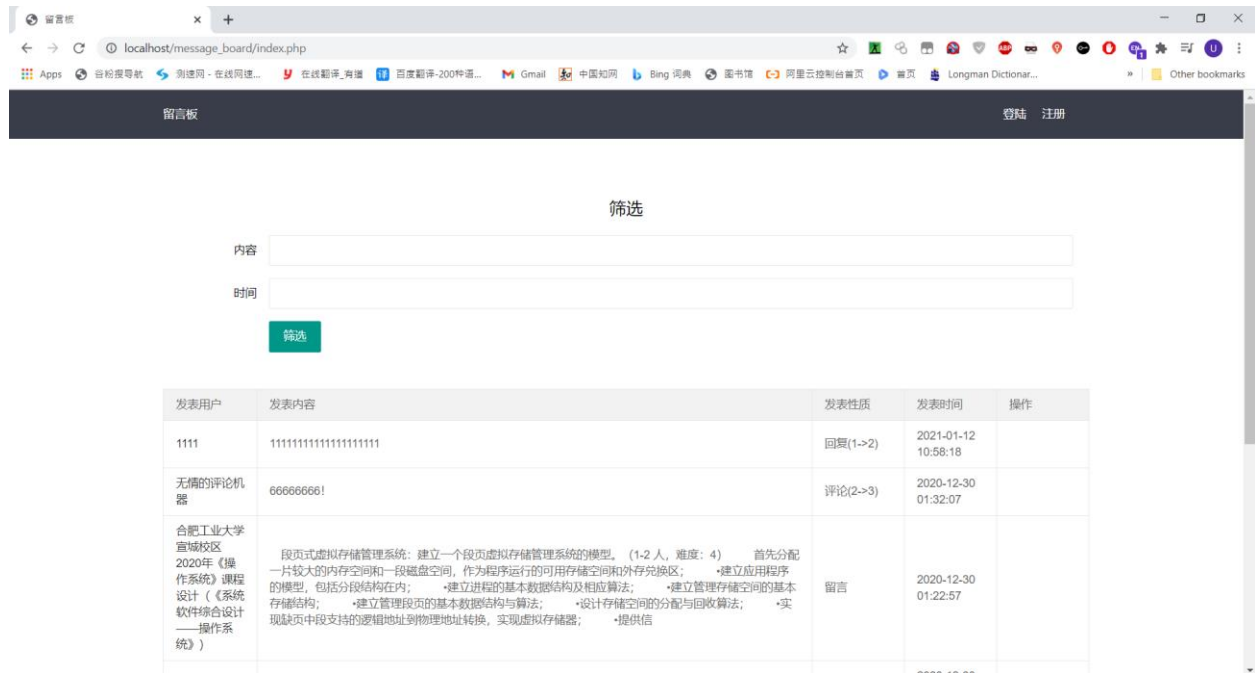


图 7.1 主页面（未登录）

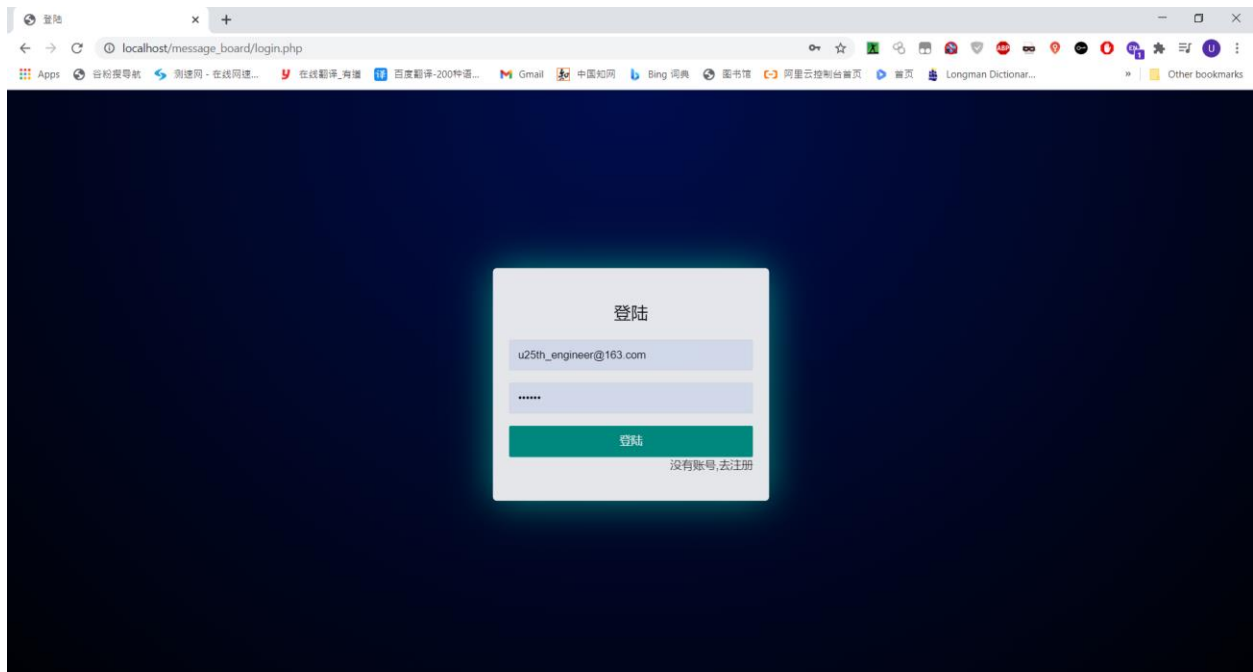


图 7.2 登陆页面

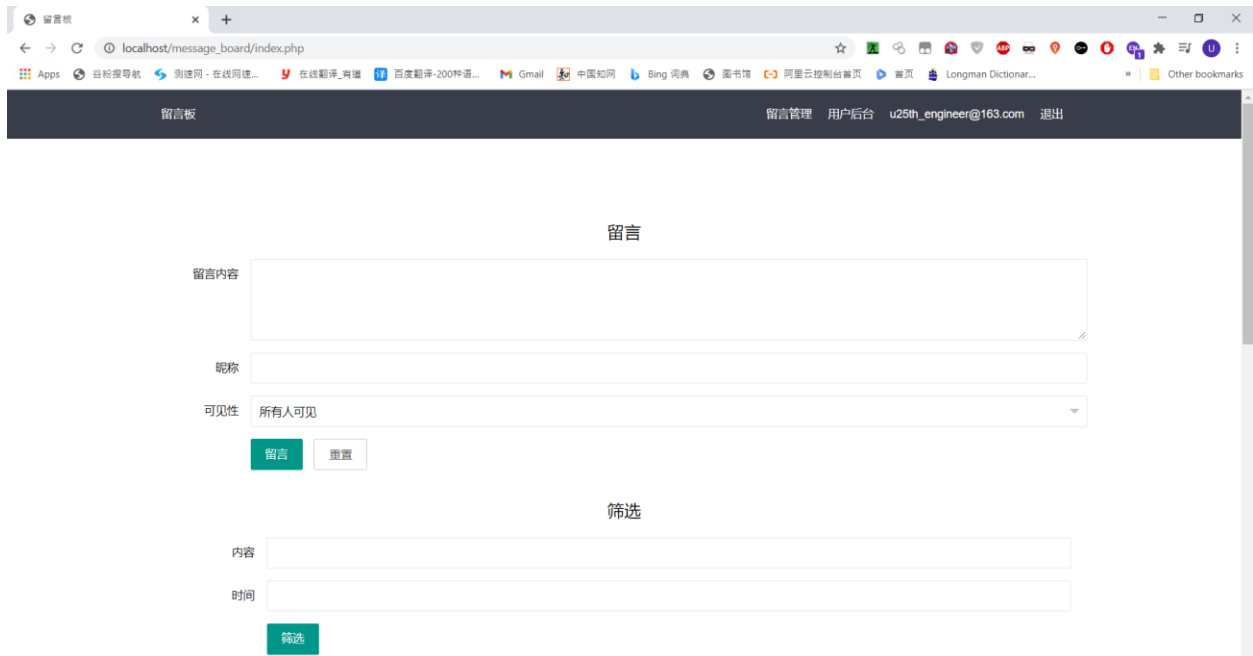


图 7.3 主页面 (管理员登陆)

时间				
筛选				
发表用户	发表内容	发表性质	发表时间	操作
1111	11111111111111111111	回复(1->2)	2021-01-12 10:58:18	删除
无情的评论机器	66666666!	评论(2->3)	2020-12-30 01:32:07	回复 删除
定理3(归纳证明原理)	(仅登录可见) 设 $P(n)$ 是关于自然数 n 的一种性质或命题. 如果当 $n=e$ 时, $P(e)$ 成立以及由 $P(n)$ 成立必可推出 $P(n+1)$ 成立,那么 $P(n)$ 对所有的 $n \in \mathbb{N}$ 都成立.	留言	2020-12-30 01:24:43	评论 删除
合肥工业大学宣城校区2020年《操作系统》课程设计(《系统软件综合设计——操作系统》)	段页式虚拟存储管理系统: 建立一个段页式虚拟存储管理系统的模型。(1-2人, 难度: 4) 首先分配一片较大的内存空间和一段磁盘空间, 作为程序运行的可用存储空间和外存交换区; 建立应用程序的模型, 包括分段结构在内; 建立进程的基本数据结构及相应算法; 建立管理存储空间的基本存储结构; 建立管理段页的基本数据结构与算法; 设计存储空间的分配与回收算法; 实现缺页中断支持的逻辑地址到物理地址转换, 实现虚拟存储器; 提供信	留言	2020-12-30 01:22:57	评论 删除
验收啊	马上就要验收了!	留言	2020-12-30 00:26:22	评论 删除
plk	66666	评论(1->2)	2020-12-29 13:47:30	回复 删除
1	又测试	留言	2020-12-29 13:47:01	删除

图 7.4 留言内容展示

筛选				
内容	6			
时间				
筛选				
发表用户	发表内容	发表性质	发表时间	操作
无情的评论机器	66666666!	评论(2->3)	2020-12-30 01:32:07	回复 删除
plk	66666	评论(1->2)	2020-12-29 13:47:30	回复 删除
UCI soybean 数据集	从实验结果可知, 对于UCI soybean数据集, 各项指标与初始化随机数的关系大致为: 初始化随机数从1增长至2时, 各项指标均上升明显, 超过20%; 初始化随机数从2增长至3时, 各项指标均有明显程度的下降; 其中准确率下降较小; 初始化随机数从3增长至4时, 除准确率大致持平外, 其余三项指标均有3%~5%的上升; 初始化随机数从4增长至5时, 各项指标均有超过明显的下降, 其中召回率下降达13%; 初始化随机数从5增长至6时, 各项指标均上升, 其中准确率上升程度为8.1%, 精确率上升程度是19.5%, 召回率与f1上升几乎持平	留言	2020-11-14 07:22:40	删除

图 7.5 留下内容筛选（筛选关键词：“6”）

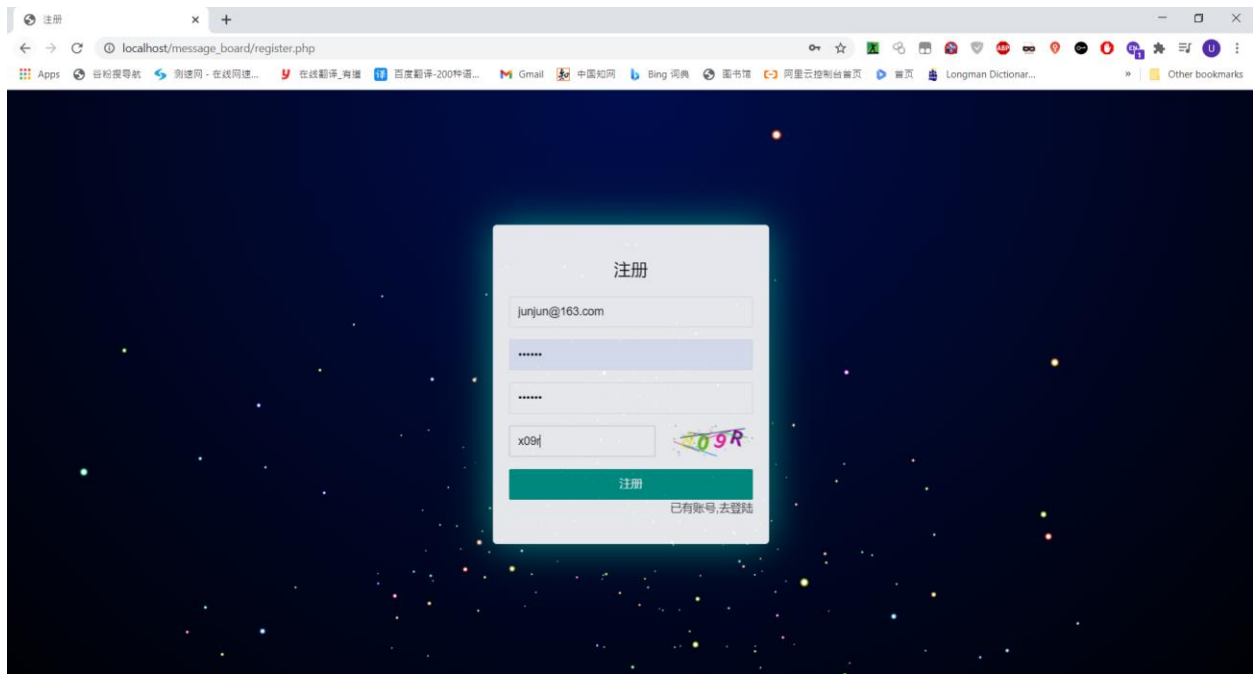


图 7.8 注册页面



图 7.9 验证码输入错误时提示

留言板

留言管理 用户后台 junjun@163.com 退出

用户ID

4

公开

私密

邮箱

junjun@163.com

公开

私密

真实姓名

公开

私密

手机

公开

私密

博客

公开

私密

博客

公开

私密

QQ

公开

私密

微信

公开

私密

微博

公开

私密

保存

重置

图 7.10 用户个人信息页面

用户

localhost/message_board/user.php

留言板 留言管理 用户后台 junjun@163.com 退出

用户ID	邮箱	真实姓名	手机	博客	QQ	微信	微博	类型	禁言	操作
1	私密	Utah Xef	18856302551	123456789012345123456789012345	2046195761	u25th_engineer	https://weibo.com/25thEngineer	管理员		
<input type="checkbox"/>	2	uath@123.com	私密	私密	私密	592551037	私密	私密	普通会员	
<input type="checkbox"/>	3	私密	私密	私密	私密	2046195761	iShi_qian	私密	普通会员	
<input type="checkbox"/>	私密	私密	私密	私密	私密	私密	私密	私密	普通会员	

图 7.11 普通用户的用户后台

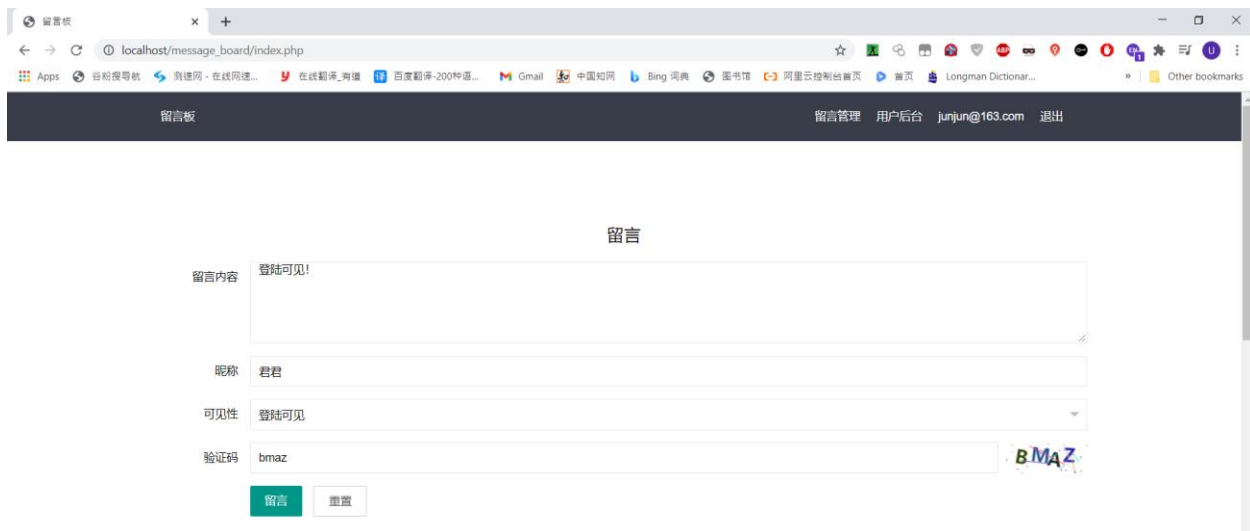


图 7.12 留言设置“登陆可见”

发表用户	发表内容	发表性质	发表时间	操作
君君	登陆可见!	留言	2021-01-13 07:39:10	删除
1111	11111111111111111111	回复(1->2)	2021-01-12 10:58:18	
无情的评论机器	66666666!	评论(2->3)	2020-12-30 01:32:07	
定理3(归纳证明原理)	(仅登录可见) 设 $P(n)$ 是关于自然数 n 的一种性质或命题.如果当 $n=e$ 时, $P(e)$ 成立以及由 $P(n)$ 成立必可推出 $P(n+1)$ 成立,那么 $P(n)$ 对所有的 $n \in \mathbb{N}$ 都成立.	留言	2020-12-30 01:24:43	评论

图 7.13 “登陆可见”留言浏览

用户

localhost/message_board/user.php

留言管理 用户后台 u25th_engineer@163.com 退出

删除

	用户ID	邮箱	真实姓名	手机	博客	QQ	微信	微博	类型	禁言	操作
	1	u25th_engineer@163.com	Utah Xef	18856302551	123456789012345123456789012345	2046195761	u25th_engineer	https://weibo.com/25thEngineer	管理员		
<input type="checkbox"/>	2	uath@123.com				592551037			普通会员		禁言
<input type="checkbox"/>	3	aoleman@123.com	奥特曼			2046195761	iShi_qian		普通会员		禁言
<input type="checkbox"/>	4	junjun@163.com							普通会员		禁言

图 7.14 管理员的用户后台

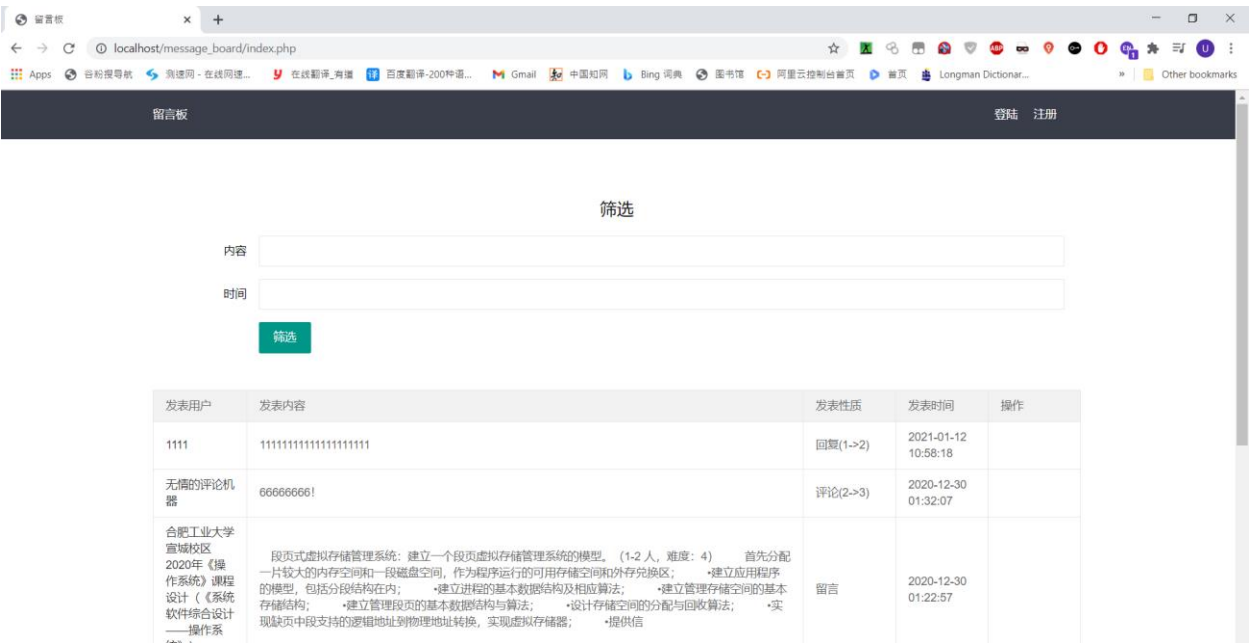


图 7.15 游客无法浏览“登陆可见”留言内容

8 软件使用说明

8.1 网上留言簿系统安装

8.1.1 软件要求

服务器端数据库系统	MySQL 8.0
服务器端应用程序服务器	Apache 2.4
服务器端脚本解释器	PHP 7.2

8.1.2 安装步骤

启动 MySQL 服务；

将源代码目录下 sql_data 的 message_board.sql 文件导入数据库，并修改 config.php 的相应配置；

将项目文件放置在 Apache 工作目录下；

启动 Apache 服务。

此时打开浏览器，输入地址即可浏览网络留言簿的内容并进行一系列的操作。

9 参考资料

- [1]苏红丽. 基于 PHPCMS V9 的留言板系统开发[J]. 办公自动化: 2015.8: 56 – 58.
- [2]张慧. 基于 Web 的留言板系统设计与实现[J]. 科技广场: 2014.11: 92 – 95.
- [3]王聪. PHP 留言板管理系统的设计与实现[J]. 中国新通信: 2015. 14
- [4]陈泽鹏. 基于 HTML5 技术的移动数字校园的研究与应用[D]. 湖南大学硕士学位论文: 2014: 2 – 3.
- [5]王昭, 邓浩江, 朱小勇, 胡琳琳. 终端 Web 运行环境及其相关优化技术研究综述[J]. 网络新媒体技术: 2020.1: 2 – 3.
- [6]张玉清, 贾岩, 雷柯楠, 吕少卿, 乐洪舟. HTML5 新特性安全研究综述[J]. 计算机研究与发展: 2016: 2165 – 2166.

10 验收时间及验收情况

验收时间：2020 年 12 月 30 日上午。

验收情况：演示了系统的使用，认真回答老师所问的问题，老师问的最后一个问题是“这个程序怎么实现留言内容的搜索的？”，这项目的年代有点久远了，我一下子没想到是直接用的 medoo 里面的 API 实现的，搞得有点久。

11 设计体会

自己的动手能力较弱，很多基础的知识都还不熟，在这次课程设计中没少绕弯路。查阅了大量资料。网络实训不仅锻炼了我的编程能力，也增强了我的学习能力，为将来进一步学习这方面的知识夯实了基础。