# 实验报告 网站设计与架构 题目-在线论坛

# 吉林大学 软件学院

## 一小组成员及工作占比

姓名	学号	占比	工作概述
梁镜湖	55170130	40%	组长,后端设计,后端主要业务,服务器编程
钱辂	55170114	30%	数据库设计,部分后端逻辑实现
王李鑫	55170131	30%	交互和页面设计,web前端实现,编写报告

#### 二系统概述

#### 系统设计原则:

- 1. 开放性、可扩充性、可靠性原则。 开放系统是生产各种计算机产品普遍遵循的原则,遵循这种标准的产品都符合一些公共的、可以相互操作的标准,能够融洽的在一起工作。 开放系统使得各种类型的网络和系统互连简单、标准统一,容易扩展升级。从而适应广大用户需求的多变性和产品的更新换代。
- 2. 良好的用户操作界面。 用户操作界面美观、方便、实用, 使用户能在较短的时间内掌握其使用方法。
- 3. 实用性原则。 任何系统的设计都要考虑其实用性,系统开发的目的是为了实现业务处理自动化、规范化,提高工作效率,减轻工作人员的劳动强度,减少开支。 (4)工作平台设计原则。 能适应不同的操作平台,不同的网络。

将功能分为前台和后台两类,因此模块也分为两大类: 前台模块和后台模块。 用户在前台的注册,登录,以及修改个人的注册信息组合成注册登录模块; 用户浏览板块, 浏览主题帖列表, 查看帖子组成浏览模块; 用户发帖, 回帖, 编辑自己发布的帖子组成发帖回帖模块; 管理员编辑帖子, 删除帖子。以上模块组成前台的功能模块。后台模块都是与管理员相关的, 设置论坛参数单独为论坛设置模块; 添加, 删除和设置权限为管理用户模块。

#### 三设计目标

#### 用户功能:

- 1. 用户注册: 只有填写注册信息的用户才能在论坛中发表帖子,新用户注册系统会自动给用户一个提示信息。
- 2. 用户登陆:可以选择在登陆页面登录或者直接在发表帖子处登陆,系统自动记录用户登录信息,只有登录用户才能发表帖子。
- 3. 修改资料: 登录用户可以对自己的原始资料进行修改。
- 4. 发表帖子: 登陆的注册用户可以在论坛内发表帖子。
- 5. 回复帖子: 登录的注册用户可以对帖子进行回复。
- 6. 编辑帖子: 发帖用户可以对自己发表的文章进行修改, 版主和管理员有权限对所有帖子进行编辑。

#### 版主功能:

- 1. 具有全部普通用户功能
- 2. 版内文章管理,包括增删改查
- 3. 取消或恢复用户在版内的发文权

#### 管理员功能:

对于管理员而言,因为是超级用户,登录管理页面可以对论坛,用户信息,论坛的样式等进行管理。管理员具有普通用户和版主的所有功能。

- 1. 帖子、论坛管理:可以对所有论坛帖子进行增加、删除、转移论坛、删除某一用户帖子等操作。对 分论坛进行管理,编辑。
- 2. 用户管理:对论坛的所有用户可以修改其部分信息,以及指定版主,增删用户。
- 3. 屏蔽和封杀功能:可以对指定账号进行屏蔽和永久冻结。

#### 四使用技术和开发环境

- 主要开发语言: golang, HTML
- 运行环境: docker, Ubuntu20.04, web浏览器
- 主要使用库: gin, gorm, bootstrap

## 五程序主要模块介绍

#### 1. model

服务器的持久层,主要用于定义数据的存储方式存储对象,数据的关系以及限定访问方法,主要使用 gorm进行实现

#### 2. control

服务器的业务逻辑层,完成登录管理,逻辑处理等主要业务功能

#### 3. requestHandler

定义处理请求的方法,主要控制服务器的请求响应方式,实现与客户端之间的交互,主要通过goem 进行实现

#### 4. config

负责管服务器配置长的简单包,从本地配置文件或者其他方式载入管理员填写的配置和密钥等

#### 5 template

前段界面模板

## 六 数据库设计

在本设计中,数据库表单和各项配置均采用orm进行生成,下面给出model定义

```
type SubForum struct {
 2
      gorm.Model
 3
                               `gorm:"unique;type:varchar(40)"`
      Name string
      Blocks []Block
                              `gorm:"foreignKey:SubForumName;references:Name"`
 4
 5
   }
 6
   type Block struct {
      gorm.Model
                               `gorm:"type:varchar(40)"`
 8
      SubForumName string
            string
9
      Name
                               `gorm:"unique;type:varchar(40)"`
                  string
10
      MasterID
11
      Master
                       User
                   []Theme
12
      Themes
13
   type Theme struct {
14
15
      BlockID uint
                           `gorm:"primaryKey"`
16
      BlockName string
                           `gorm:"type:varchar(40)"`
                           `gorm:"primaryKey;type:varchar(30)"`
17
      Name
            string
18
      Threads []Thread
    `gorm:"foreignKey:BlockID, ThemeName; references:BlockID, Name"`
19
20
   type BlockFrozenUser struct {
21
       gorm.Model
22
       UserID
                  string
                               `gorm:"unique"`
23
       User
                   User
24
       ThawTime time.Time
25
   }
```

```
26
    type ForumFrozenUser struct {
27
        gorm.Model
                                  `gorm:"unique"`
28
        UserID
                     string
29
        User
                     User
30
        ThawTime
                     time.Time
31
32
    type BanedIP struct {
33
        gorm.Model
34
                string
                             `gorm:"unique;index;type:varchar(12)"`
35
    type Thread struct {
36
37
        gorm.Model
38
        Tittle
                                     `gorm:"type:varchar(100)"`
                     string
        BlockID
39
                    uint
                                     `gorm:"type:varchar(30)"`
40
        ThemeName
                     string
        UserID
                                     `gorm:"type:varchar(11)"`
41
                     string
42
        User
                     User
    `gorm:"constraint:OnUpdate:CASCADE,OnDelete:CASCADE"`
43
        Comments
                    []Comment
    `gorm:"constraint:OnUpdate:CASCADE,OnDelete:CASCADE"`
                    ThreadContent
44
    `gorm:"constraint:OnUpdate:CASCADE,OnDelete:CASCADE"`
45
46
    type ThreadContent struct {
47
        ThreadID
                     uint
48
        Content
                     string
49
50
    type Comment struct{
51
        gorm.Model
        ThreadID
                     uint
53
        UserID
                     string
54
        User
                     User
55
        Content
                     string
56
57
    type User struct {
58
        ID
                     string
                                 `gorm:"primaryKey;type:varchar(11)"`
                                  `gorm:"unique;type:varchar(50)"`
59
        Email
                     string
                                 `gorm:"type:varchar(30)"`
60
        UserName
                     string
                                  `gorm:"type:varchar(30)"`
61
        Password
                     string
62
        LastIP
                     string
                                  `gorm:"type:varchar(16)"`
63
        ThawTime
                     time.Time
    }
```

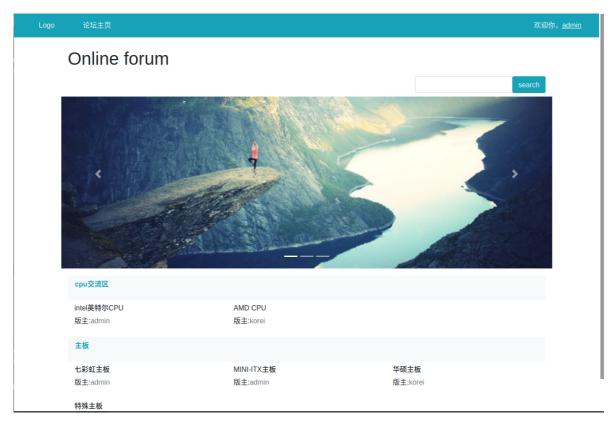
## 七界面功能简要展示

用户登录和注册



注册新用户	×
ID:	
Name:	
Email:	
Password:	
	提交 关闭

论坛主页



## 个人信息页面



## 板块页面



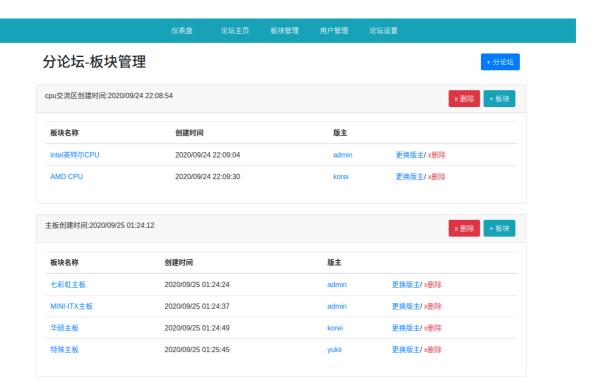
#### 帖子和回复页面



#### 版主主题管理页面



#### 管理员分论坛和板块版主管理页面



#### 用户管理页面



## 八系统遗留问题

#### 尚未完成的功能:

- 1. 禁止用户发言
- 2. IP管理
- 3. 热门帖子推荐
- 4. 设置精华帖子

#### 存在缺陷:

- 1. 部分页面跳转问题
- 2. 表单提交响应