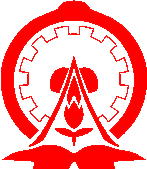
**湖南铁路科技职业技术学院**

**毕业设计成果书**



|  |  |
| --- | --- |
| **课 题：** | **基于ASP.NET Core的骑行者论坛** |
| **专 业：** | **软件技术** |
| **班 级：** | **2002班** |
| **学生姓名：** | **胡威** |
| **所属学院：** | **铁道工程与信息学院** |
| **指导教师：** | **高子童** |

湖南铁路科技职业技术学院教务处监制

目录

[第一章 项目描述及要求 4](#_Toc19767)

[第二章 项目概要设计 5](#_Toc8727)

[2.1 程序用例图 5](#_Toc18923)

[2.2 各类用户各模块功能思维导图 5](#_Toc4035)

[2.2.1 登录注册功能模块思维导图 5](#_Toc13361)

[2.2.2 浏览帖子功能模块思维导图 6](#_Toc6960)

[2.2.3 发布帖子功能 7](#_Toc1738)

[2.2.4 帖子收藏(点赞)功能 8](#_Toc30638)

[2.2.5 帖子评论功能 8](#_Toc11808)

[2.2.6 管理员特权 9](#_Toc7182)

[2.2.7 吧主特权 9](#_Toc27943)

[2.3 数据库设计 10](#_Toc32032)

[2.3.1 用户表设计 10](#_Toc1945)

[2.3.2 帖子表设计 11](#_Toc19027)

[2.3.3 帖子类型表设计 11](#_Toc30009)

[2.3.4 帖子评论表设计 12](#_Toc25399)

[2.3.5 帖子收藏表设计 12](#_Toc30111)

[2.3.6 骑行团表设计 13](#_Toc28657)

[第三章 项目详细设计 14](#_Toc22604)

[3.1 网站设计思维导图 14](#_Toc5496)

[3.2 登录、注册模块 15](#_Toc18096)

[3.2.1 登录、注册模块简介 15](#_Toc22148)

[3.2.2 登录功能模块流程图 15](#_Toc25473)

[3.2.3 注册功能模块流程图 16](#_Toc21069)

[3.2.4 登录、注册功能模块界面 17](#_Toc8970)

[3.2.3 登录、注册功能模块核心代码 18](#_Toc32287)

[3.3 登录成功后主页模块 19](#_Toc22254)

[3.3.1 登录成功后主页模块简介 19](#_Toc22667)

[3.3.2 登录成功后主页模块流程图 19](#_Toc4359)

[3.3.3 登录成功后主页模块界面 20](#_Toc19273)

[3.3.3 登录成功后主页模块主要代码 20](#_Toc24144)

[3.4 浏览帖子功能模块 21](#_Toc22791)

[3.4.1 浏览帖子功能模块简介 21](#_Toc8690)

[3.4.2 浏览帖子功能模块流程图 21](#_Toc24238)

[3.4.3 浏览帖子功能模块界面 21](#_Toc23025)

[3.4.4 浏览帖子功能核心代码 22](#_Toc11248)

[22](#_Toc8718)

[3.5 帖子分类模块 22](#_Toc5397)

[3.5.1 帖子分类模块简介 22](#_Toc11983)

[3.5.2 帖子分类模块流程图 23](#_Toc28770)

[3.5.3 帖子分类模块界面 23](#_Toc2521)

[3.5.4 帖子分类模块核心代码 23](#_Toc17298)

[3.6 帖子排序模块 24](#_Toc25082)

[3.6.1 帖子排序模块简介 24](#_Toc18999)

[3.6.2 帖子排序模块流程图 25](#_Toc5265)

[3.6.3 帖子排序模块界面 26](#_Toc1021)

[3.6.4 帖子排序模块核心代码 26](#_Toc830)

[3.7 帖子搜索功能模块 26](#_Toc12833)

[3.7.1 帖子搜索功能模块简介 26](#_Toc8184)

[3.7.2 帖子搜索功能模块流程图 26](#_Toc9634)

[3.7.3 帖子搜索功能模块界面 26](#_Toc11010)

[3.7.4 帖子搜索功能模块核心代码 27](#_Toc3062)

[3.8 帖子发布功能模块 27](#_Toc14677)

[3.8.1 帖子发布功能模块简介 27](#_Toc5690)

[3.8.2 帖子发布功能模块流程图 27](#_Toc3952)

[3.8.3 帖子发布功能模块界面 28](#_Toc30538)

[3.8.4 帖子发布功能模块核心代码 28](#_Toc31675)

[3.9 帖子审核模块 28](#_Toc22299)

[3.9.1 帖子审核模块简介 28](#_Toc29910)

[3.9.2 帖子审核流程图 28](#_Toc3651)

[3.9.3 帖子审核模块界面 29](#_Toc9101)

[3.9.4 帖子审核模块核心代码 29](#_Toc21163)

[3.10 管理员权限授予功能模块 29](#_Toc32076)

[3.10.1 管理员权限授予功能简介 29](#_Toc1705)

[3.10.2 管理员权限授予流程图 29](#_Toc31554)

[3.10.3 管理员权限授予界面 29](#_Toc9754)

[3.10.4 管理员权限授予功能核心代码 29](#_Toc7064)

[3.11 封号功能模块 30](#_Toc26941)

[3.11.1 封号功能简介 30](#_Toc1382)

[3.11.2 封号功能流程图 30](#_Toc13137)

[3.11.3 封号功能界面 30](#_Toc11598)

[3.11.4 封号功能核心代码 30](#_Toc21324)

[3.12 帖子置顶功能模块 30](#_Toc2017)

[3.12.1 帖子置顶功能简介 30](#_Toc3877)

[3.12.2 帖子置顶功能流程图 30](#_Toc60)

[3.12.3 帖子置顶功能界面 30](#_Toc30339)

[3.12.4 帖子置顶功能核心代码 31](#_Toc23676)

[第四章 系统集成测试 31](#_Toc13547)

[4.1 测试要求和环境 31](#_Toc28862)

[4.2 单元测试 32](#_Toc10335)

[4.2.1 登录功能测试 32](#_Toc16931)

[4.2.2 注册功能测试 32](#_Toc27917)

[4.2.3 主页功能测试 33](#_Toc5407)

[4.2.4 帖子分类导航栏功能测试 34](#_Toc11819)

[4.2.5 帖子浏览功能测试 34](#_Toc19503)

[4.2.6 用户功能测试 34](#_Toc8154)

[4.2.7 骑行团加入功能测试 35](#_Toc8230)

[4.2.8 管理员功能测试 35](#_Toc24585)

[4.2.9 吧主功能测试 35](#_Toc1525)

[4.3 性能测试 35](#_Toc11173)

[4.3.1 性能测试目的 36](#_Toc23467)

[4.3.2 测试结论分析 36](#_Toc29188)

[总结 38](#_Toc27254)

[致谢 39](#_Toc27319)

、

1. **项目描述及要求**

现在许多的年轻人在节假日期间都会选择户外运动,而骑行是一项很好的户外运动，也比较适合年轻人,所以骑行者们的团体非常需要的庞大，因此我决定开发一个专供骑行者的论坛网站RiderBBS,供骑行者们在此网站发布各种各样的帖子,然后还需要开发一个管理员界面和一个站长界面实现对RiderBBS的管理和对管理员的管理操作。

对与骑行者而言,骑行者们可以在登录之前通过RiderBBS查看其它骑行者们发布的各种帖子,在登录之后骑行者可以自行发布自己写好的帖子,并且可以对其他人的帖子进行评论、点赞、收藏。对与管理者而言,可以审核骑行者发布的帖子,只有通过了审核的帖子,才能让其他骑行者看到帖子，还可以置顶帖子。

本项目使用使用ASP.NET Core Blazor+HTML5+CSS3实现对网页进行渐进式开发，使用C#实现前端以及后端的逻辑交互，使用SQL Server进行数据库管理，使用Git+GitHub进行版本控制,使用AntDesign of BlazorUI库，使用阿里云进行服务器托管。

# **第二章** **项目概要设计**

**2.1 程序用例图**

**2.2 各类用户各模块功能思维导图**

**2.2.1 登录注册功能模块思维导图**

登录、注册功能思维导图如下。

给予骑行者个人账号权力

站长账号

有吧主给予的权力

无账号

有账号

**吧主**

**管理员**

**登录**

**注册**

**登录**

**登录**

**用户**

**2.2.2 浏览帖子功能模块思维导图**

浏览帖子功能思维导图如下。

有无账号均可

**用户**

**管理员**

**吧主**

**浏览帖子**

**2.2.3 发布帖子功能**

发布帖子功能思维导图如下。

不需要审核即可

需要有账号，且通过管理员审核

**用户**

**管理员**

**发布帖子**

**2.2.4 帖子收藏(点赞)功能**

帖子收藏(点赞)功能思维导图如下。

只要有相应账号均可

**管理员**

**吧主**

**收藏帖子**

**用户**

**2.2.5 帖子评论功能**

帖子评论功能思维导图如下。

**管理员**

**吧主**

**评论帖子**

**评论帖子**

**评论帖子**

**删除评论**

**骑行者**

**2.2.6 管理员特权**

管理员特权如下。

**管理员**

**审核帖子**

**管理员特殊评论**

**可以看到各帖子统计信息**

**2.2.7 吧主特权**

吧主特权如下。

**审核帖子**

**吧主**

**吧主特殊评论**

**可以看到各帖子统计信息**

**吧主上线通知**

**制订版规**

**对违规用户进行封号**

**2.3 数据库设计**

**2.3.1 用户表设计**

设计用户表，确定建表的数据名称，新建完成后，创建适当约束，添加相关数据。如下表所示。

**表 2.1 UserTable表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段显示** | **字段名** | **数据类型** | **默认值** | **备注和说明** |
| **用户ID** | UserId | Int | - | 主键,为自增列 |
| **用户名** | Username | Varchar | - | 不能为空 |
| **用户密码** | Userpwd | Varchar | - | 不能为空 |
| **是否为管理员** | IsAdmin | Bit | 0 | 默认不为管理员 |
| **骑行团ID** | CircleID | Int | - | - |

**2.3.2 帖子表设计**

设计帖子表，确定建表的数据名称，新建完成后，创建适当约束，添加相关数据。如下表所示。

**表 2.2 PostTable**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段显示** | **字段名** | **数据类型** | **默认值** | **备注和说明** |
| **帖子ID** | PostId | Int | - | 主键,自增列 |
| **帖子名** | PostName | Varchar | - | 不能为空 |
| **帖子类型ID** | PostTypeId | Int | - | 帖子类型表的外键 |
| **用户ID** | UserId | Int | - | 用户表的外键 |
| **点赞数量** | Hits | Int | 0 | - |
| **观看数量** | Views | Int | 0 | - |

**2.3.3 帖子类型表设计**

设计帖子类型表，确定建表的数据名称，新建完成后，创建适当约束，添加相关数据。如下表所示。

**表 2.3 TypeTable**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段显示** | **字段名** | **数据类型** | **默认值** | **备注和说明** |
| **帖子类型ID** | PostTypeId | Int | - | 主键 |
| **帖子类型名称** | PostTypeName | Varchar | - | 不能为空 |

**2.3.4 帖子评论表设计**

设计帖子评论表，确定建表的数据名称，新建完成后，创建适当约束，添加相关数据。如下表所示。

**表 2.4 CommentTable**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段显示** | **字段名** | **数据类型** | **默认值** | **备注和说明** |
| **评论ID** | CommentId | Int | - | 主键，自增列 |
| **评论详情** | CommentDetail | Varchar | - | 不能为空 |
| **用户ID** | UserId | Int | - | 外键 |
| **帖子ID** | PostId | Int | - | 外键 |
| **评论时间** | CommentTime | Datetime | GetDate() | 默认为当前时间 |

**2.3.5 帖子收藏表设计**

设计帖子收藏表，确定建表的数据名称，新建完成后，创建适当约束，添加相关数据。如下表所示。

**表 2.5 CollectionTable**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段显示** | **字段名** | **数据类型** | **默认值** | **备注和说明** |
| **收藏ID** | CollectionId | Int | - | 主键，自增 |
| **收藏者ID** | UserId | Int | - | - |
| **被收藏贴子ID** | PostId | Int | - | - |

**2.3.6 骑行团表设计**

设计骑行团表，确定建表的数据名称，新建完成后，创建适当约束，添加相关数据。如下表所示。

**表 2.6 CollectionTable**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段显示** | **字段名** | **数据类型** | **默认值** | **备注和说明** |
| **骑行团ID** | **CircleID** | **Int** | **-** | **主键,自增** |
| **骑行团团长ID** | **CircleMasterID** | **Int** | **-** | **-** |
| **骑行团名称** | **CircleName** | **Varchar** | **-** | **-** |

**第三章** **项目详细设计**

**3.1 网站设计思维导图**

根据对该网站需求分析与网站功能的描述，建立如下所示的模块功能思维导图。

***RiderBBS***

**用户**

**管理员**

**吧主**

**游客**

**发帖功能**

**评论功能**

**收藏**

**点赞**

**功能**

**浏览帖子功能**

**用户所有功能**

**审核帖子功能**

**后台**

**查看**

**功能**

**授予管理功能**

**封号功能**

**帖子置顶功能**

**制定版规功能**

**浏览帖子功能**

**3.2 登录、注册模块**

**3.2.1 登录、注册模块简介**

RiderBBS登录模块不分管理员登录和吧主登录,因为是账号定义为管理员或者吧主,如果是管理员或者吧主登录以后RiderBBS会有额外的功能供他们使用,类似于视频VIP。RiderBBS注册模块则只注册普通账号,无法直接注册管理员账号。

**3.2.2 登录功能模块流程图**

在登录功能中你需要输入用户名和密码,通过在数据库的查询,我们会动态的显示你输入的信息的正确性,如果用户名不存在会提示你用户名不存在,如果密码不正确会提示你密码不正确，正确则登录成功，具体如下程序流程图所示。

是

否

是

否

**开始**

**输入用户名**

**是否存在**

**输入密码**

**是否正确**

**结束**

**3.2.3 注册功能模块流程图**

在注册功能中你需要输入你要注册的用户名，通过在数据库的查询,确认用户名是否存在,若存在一样的用户名,会提示你用户名已经存在,然后输入密码以及确认密码,如果第二次输入密码与第一次输入的密码不一致,会提示你密码不一致并清空你第二次输入的密码。两次密码一致后,注册成功。具体如下程序流程图所示。

是

否

**结束**

**重复密码**

是

否

**开始**

**输入用户名**

**是否存在**

**输入密码**

**是否一致**

**3.2.4 登录、注册功能模块界面**

登录界面如下图所示。



如用户没有账号，可点击主页注册按钮来切换到注册页面进行账号注册操作。注册页面如下图所示。



**3.2.3 登录、注册功能模块核心代码**

// 用户名框的状态

public FormValidateStatus NameStatus { get; set; } = FormValidateStatus.Default;

private User user=new User();

// 判断用户名是否正确

private void JuageName()

{

// 判断条件

bool hasName=false;

// 获取用户表

DataTable dt=DBHelper.GetData("select \* from UserTable");

// 遍历用户表

foreach (DataRow dr in dt.Rows)

{

// 若不存在输入的用户名,将hasName改为false

hasName = user.UserName != dr["Username"].ToString() ? false : true;

}

// 如果存在用户名

if (hasName)

{

NameStatus = FormValidateStatus.Success;

}

// 如果不存在用户名

else

{

NameStatus = FormValidateStatus.Error;

}

}

**3.3 登录成功后主页模块**

**3.3.1 登录成功后主页模块简介**

因为登录成功后用户部分的功能被激活,所以主页布局也要做相应的变化。

**3.3.2 登录成功后主页模块流程图**

普通用户登录成功后,原主页登录注册部分会被替换为用户的用户名,如果是管理员，用户名是红色的,如果是吧主，用户名是蓝色。具体流程如下。

**开始**

管理员？

吧主？

**普通用户**

**管理员**

**吧主**

否

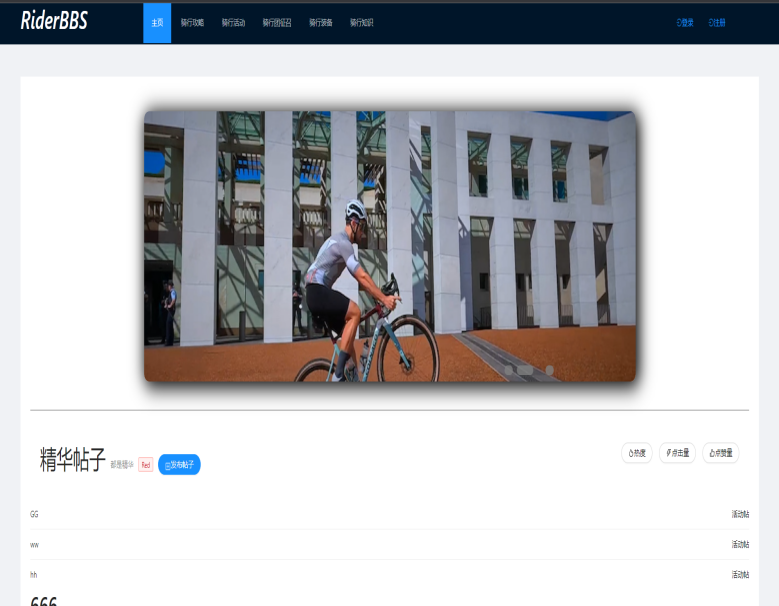
是

是

否

**3.3.3 登录成功后主页模块界面**

登录界面如下图所示。



**3.3.3 登录成功后主页模块主要代码**

// 用户名框的状态

public FormValidateStatus NameStatus { get; set; } = FormValidateStatus.Default;

private User user = new User();

// 判断用户名是否正确

private void JuageName()

{

// 判断条件

bool hasName = false;

// 获取用户表

DataTable dt = DBHelper.GetData("select \* from UserTable");

// 遍历用户表

foreach (DataRow dr in dt.Rows)

{

// 若不存在输入的用户名,将hasName改为false

hasName = user.UserName != dr["Username"].ToString() ? false : true;

}

// 如果存在用户名

if (hasName)

{

NameStatus = FormValidateStatus.Success;

}

// 如果不存在用户名

else

{

NameStatus = FormValidateStatus.Error;

}

}

**3.4 浏览帖子功能模块**

**3.4.1 浏览帖子功能模块简介**

RiderBBS是一个开放性的论坛网站,所以不但对已经注册过的账户开放,游客也可以进入网站浏览骑行者们发布的帖子。

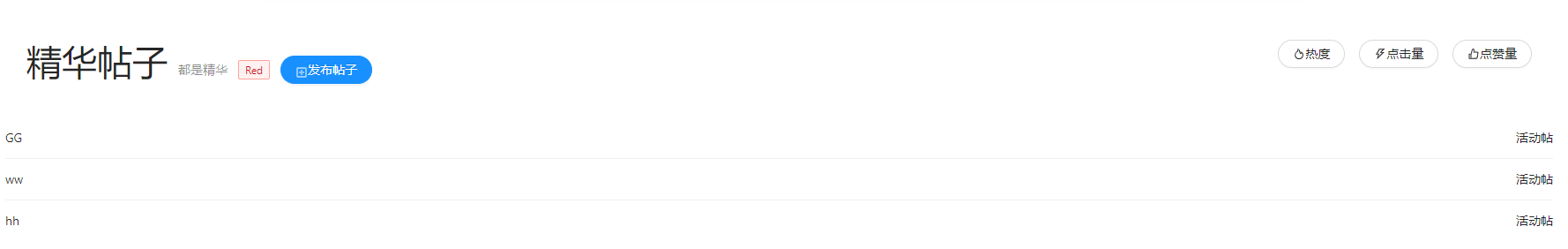
**3.4.2 浏览帖子功能模块流程图**

**开始**

**浏览帖子**

**选择帖子**

**3.4.3 浏览帖子功能模块界面**



**3.4.4 浏览帖子功能核心代码**

// 用户名框的状态

public FormValidateStatus NameStatus { get; set; } = FormValidateStatus.Default;

private User user = new User();

// 判断用户名是否正确

private void JuageName()

{

// 判断条件

bool hasName = false;

// 获取用户表

DataTable dt = DBHelper.GetData("select \* from UserTable");

// 遍历用户表

foreach (DataRow dr in dt.Rows)

{

// 若不存在输入的用户名,将hasName改为false

hasName = user.UserName != dr["Username"].ToString() ? false : true;

}

// 如果存在用户名

if (hasName)

{

NameStatus = FormValidateStatus.Success;

}

// 如果不存在用户名

else

{

NameStatus = FormValidateStatus.Error;

}

}

**3.5 帖子分类模块**

**3.5.1 帖子分类模块简介**

每个骑行者都有各自的需求,因此所需以及发布的帖子都有各自的特点,RiderBBS会在骑行者发布帖子的时候要求他选择自己帖子的类型,向其他骑行者展示时,我们会对骑行者发布的帖子进行分类，从而让骑行者高效且便捷的找到自己所需要的帖子。

**3.5.2 帖子分类模块流程图**

**发布帖子**

**填写类型**

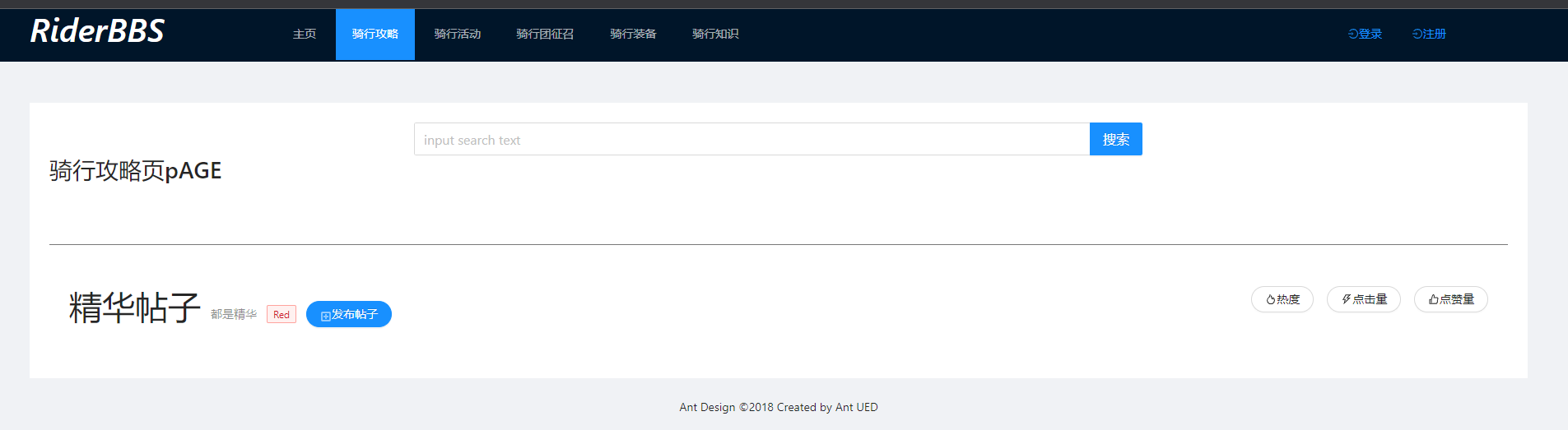
**发布完成**

**浏览帖子**

**选择类型**

**浏览**

**3.5.3 帖子分类模块界面**



**3.5.4 帖子分类模块核心代码**

// 用户名框的状态

public FormValidateStatus NameStatus { get; set; } = FormValidateStatus.Default;

private User user=new User();

// 判断用户名是否正确

private void JuageName()

{

// 判断条件

bool hasName=false;

// 获取用户表

DataTable dt=DBHelper.GetData("select \* from UserTable");

// 遍历用户表

foreach (DataRow dr in dt.Rows)

{

// 若不存在输入的用户名,将hasName改为false

hasName = user.UserName != dr["Username"].ToString() ? false : true;

}

// 如果存在用户名

if (hasName)

{

NameStatus = FormValidateStatus.Success;

}

// 如果不存在用户名

else

{

NameStatus = FormValidateStatus.Error;

}

}

**3.6 帖子排序模块**

**3.6.1 帖子排序模块简介**

很多时候,人们在网上浏览各式各样的信息,判断它是否是一个好的东西的标准都是这个信息它的热度够不够,它的点赞量多不多,它的观看量多不多,帖子也是如此,所以我们通过对帖子进行各种排名,用户可以根据热度，点赞量,以及观看量,实现快速对优质帖子的寻找。

**3.6.2 帖子排序模块流程图**

否

是

是

**选择排序类型**

点赞量》

**按点赞量排序**

**按观看量排序**

**按热度排序**

观看量》

否

**3.6.3 帖子排序模块界面**



**3.6.4 帖子排序模块核心代码**

// 用户名框的状态

public FormValidateStatus NameStatus { get; set; } = FormValidateStatus.Default;

private User user=new User();

// 判断用户名是否正确

private void JuageName()

{

// 判断条件

bool hasName=false;

// 获取用户表

DataTable dt=DBHelper.GetData("select \* from UserTable");

// 遍历用户表

foreach (DataRow dr in dt.Rows)

{

// 若不存在输入的用户名,将hasName改为false

hasName = user.UserName != dr["Username"].ToString() ? false : true;

}

// 如果存在用户名

if (hasName)

{

NameStatus = FormValidateStatus.Success;

}

// 如果不存在用户名

else

{

NameStatus = FormValidateStatus.Error;

}

}

**3.7 帖子搜索功能模块**

**3.7.1 帖子搜索功能模块简介**

当骑行者想要更快的找到自己的想要的帖子时,帖子搜索功能是一个很好的选择,帖子搜索能够定向且高速的找到骑行者所需的内容,我们还会根据骑行者填写的内容通过算法进行模糊查找,解决了当骑行者填写的搜索内容模棱两可的情况。

**3.7.2 帖子搜索功能模块流程图**

否

**提示用户没有此贴**

是

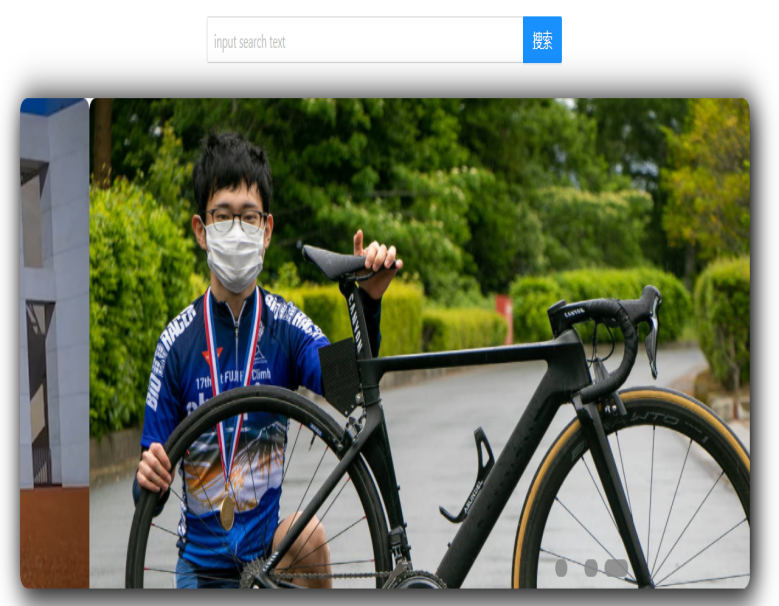
**开始**

是否有内容包含输入值

**显示搜索到的帖子**

**接收搜索框输入**

**3.7.3 帖子搜索功能模块界面**



**3.7.4 帖子搜索功能模块核心代码**

// 用户名框的状态

public FormValidateStatus NameStatus { get; set; } = FormValidateStatus.Default;

private User user=new User();

// 判断用户名是否正确

private void JuageName()

{

// 判断条件

bool hasName=false;

// 获取用户表

DataTable dt=DBHelper.GetData("select \* from UserTable");

// 遍历用户表

foreach (DataRow dr in dt.Rows)

{

// 若不存在输入的用户名,将hasName改为false

hasName = user.UserName != dr["Username"].ToString() ? false : true;

}

// 如果存在用户名

if (hasName)

{

NameStatus = FormValidateStatus.Success;

}

// 如果不存在用户名

else

{

NameStatus = FormValidateStatus.Error;

}

}

**3.8 帖子发布功能模块**

**3.8.1 帖子发布功能模块简介**

作为一个论坛网站,发布帖子的功能是必不可少的,当骑行者需要发布帖子时，首先需要注册一个RiderBBS账号,其次需要有写好自己的标题以及帖子内容,最后将帖子发布请求提交给服务器,服务器再将请求交给RiderBBS管理员，只有当管理员的审核通过了,骑行者的帖子才算正式发布成功了。

**3.8.2 帖子发布功能模块流程图**

**申请发布帖子**

帖子是否通过审核

**发布成功**

是

否

**发布失败**

**3.8.3 帖子发布功能模块界面**



**3.8.4 帖子发布功能模块核心代码**

// 用户名框的状态

public FormValidateStatus NameStatus { get; set; } = FormValidateStatus.Default;

private User user=new User();

// 判断用户名是否正确

private void JuageName()

{

// 判断条件

bool hasName=false;

// 获取用户表

DataTable dt=DBHelper.GetData("select \* from UserTable");

// 遍历用户表

foreach (DataRow dr in dt.Rows)

{

// 若不存在输入的用户名,将hasName改为false

hasName = user.UserName != dr["Username"].ToString() ? false : true;

}

// 如果存在用户名

if (hasName)

{

NameStatus = FormValidateStatus.Success;

}

// 如果不存在用户名

else

{

NameStatus = FormValidateStatus.Error;

}

}

**3.9 帖子审核模块**

**3.9.1 帖子审核模块简介**

为了防止骑行者们发布包含不良信息的帖子,我们加入了帖子审核功能来有效应对这种情况，当骑行者发起发布帖子的请求是，管理员会收到服务器发过来的请求,管理员可以浏览骑行者请求发布的帖子,只有当管理员同意了发布请求，骑行者的帖子才算真正的发布了。

**3.9.2 帖子审核流程图**

**管理员收到需要发表的帖子**

帖子是否符合发布标准

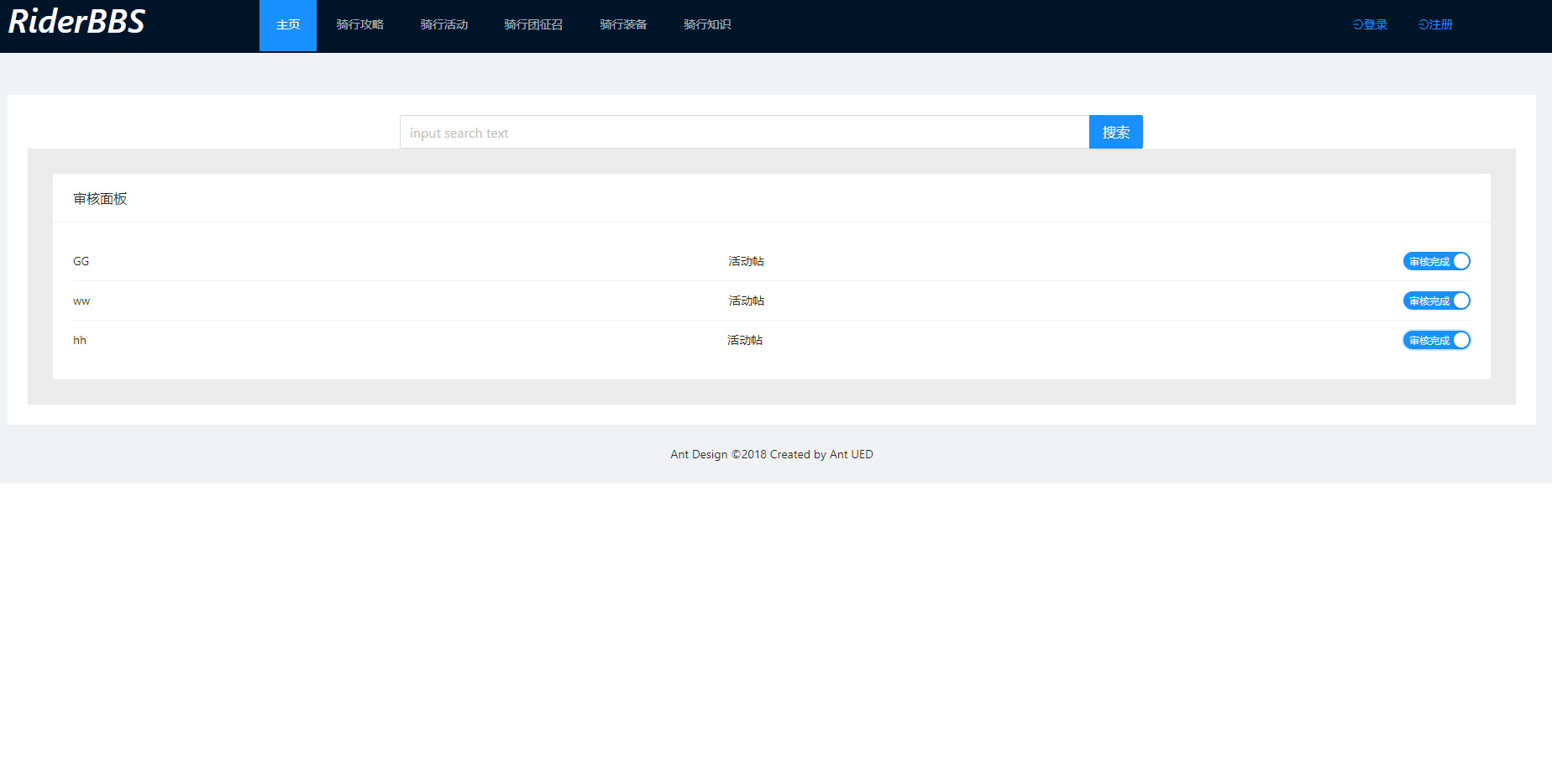
**允许发布**

是

否

**拒绝发布**

**3.9.3 帖子审核模块界面**



**3.9.4 帖子审核模块核心代码**

// 用户名框的状态

public FormValidateStatus NameStatus { get; set; } = FormValidateStatus.Default;

private User user=new User();

// 判断用户名是否正确

private void JuageName()

{

// 判断条件

bool hasName=false;

// 获取用户表

DataTable dt=DBHelper.GetData("select \* from UserTable");

// 遍历用户表

foreach (DataRow dr in dt.Rows)

{

// 若不存在输入的用户名,将hasName改为false

hasName = user.UserName != dr["Username"].ToString() ? false : true;

}

// 如果存在用户名

if (hasName)

{

NameStatus = FormValidateStatus.Success;

}

// 如果不存在用户名

else

{

NameStatus = FormValidateStatus.Error;

}

}

**3.10 管理员权限授予功能模块**

**3.10.1 管理员权限授予功能简介**

需要实现帖子审核功能,那就必须得要一个管理员,本吧主就拥有授予用户管理员的功能,通过吧主的后台页面,可以授予和取消普通用户管理员的权限,普通用户拥有了管理员权限也就成为了管理员,就可以来给我们审核普通用户发布的帖子了。

**3.10.2 管理员权限授予流程图**

**吧主想授予管理员权限**

此人是否值得信任

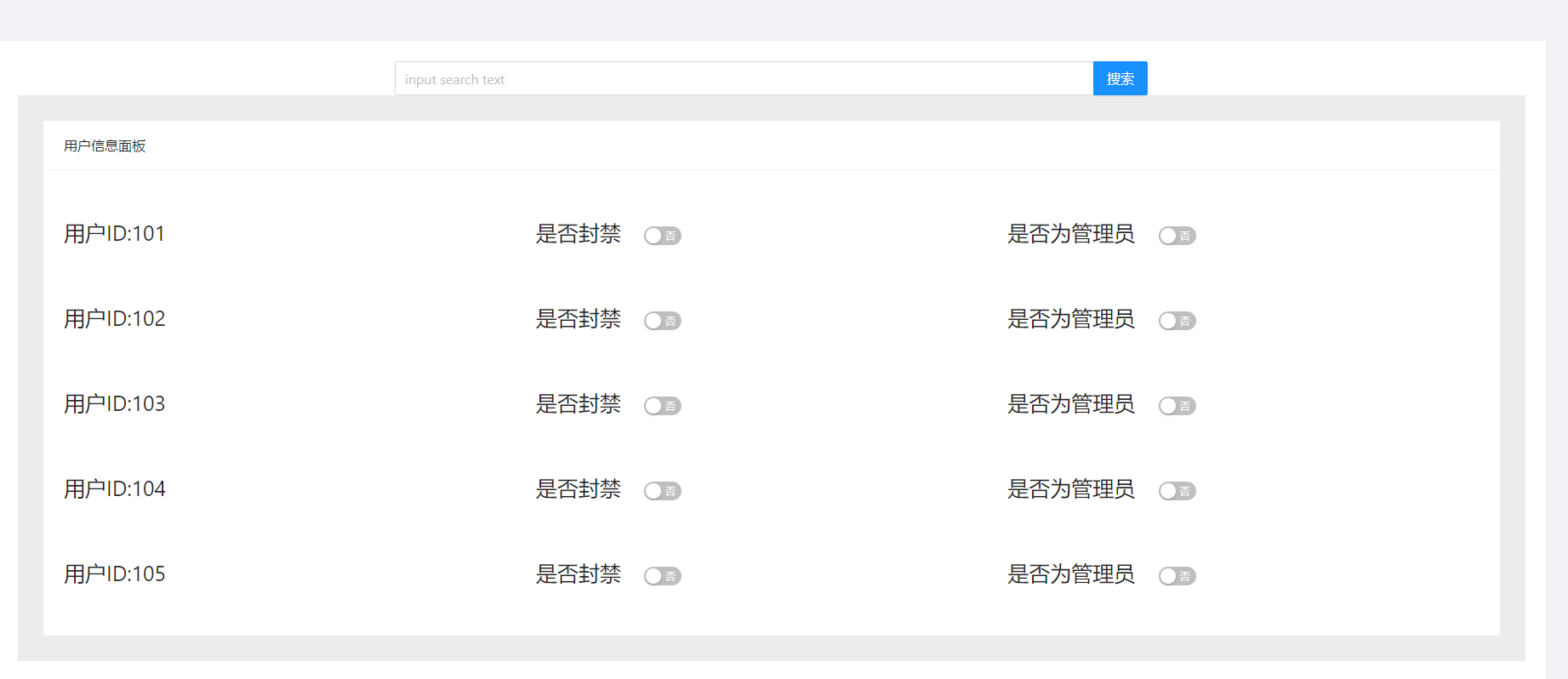
**到后台授予权限**

是

否

**换下一个人**

**3.10.3 管理员权限授予界面**



**3.10.4 管理员权限授予功能核心代码**

// 用户名框的状态

public FormValidateStatus NameStatus { get; set; } = FormValidateStatus.Default;

private User user=new User();

// 判断用户名是否正确

private void JuageName()

{

// 判断条件

bool hasName=false;

// 获取用户表

DataTable dt=DBHelper.GetData("select \* from UserTable");

// 遍历用户表

foreach (DataRow dr in dt.Rows)

{

// 若不存在输入的用户名,将hasName改为false

hasName = user.UserName != dr["Username"].ToString() ? false : true;

}

// 如果存在用户名

if (hasName)

{

NameStatus = FormValidateStatus.Success;

}

// 如果不存在用户名

else

{

NameStatus = FormValidateStatus.Error;

}

}

**3.11 封号功能模块**

**3.11.1 封号功能简介**

为了防止各别用户破坏RiderBBS的网络环境，我给吧主提供了封号功能，当吧主看到不良的评论信息时，可以实现对这些用户的封号处理！

**3.11.2 封号功能流程图**

**吧主浏览帖子评论**

评论是否违规？

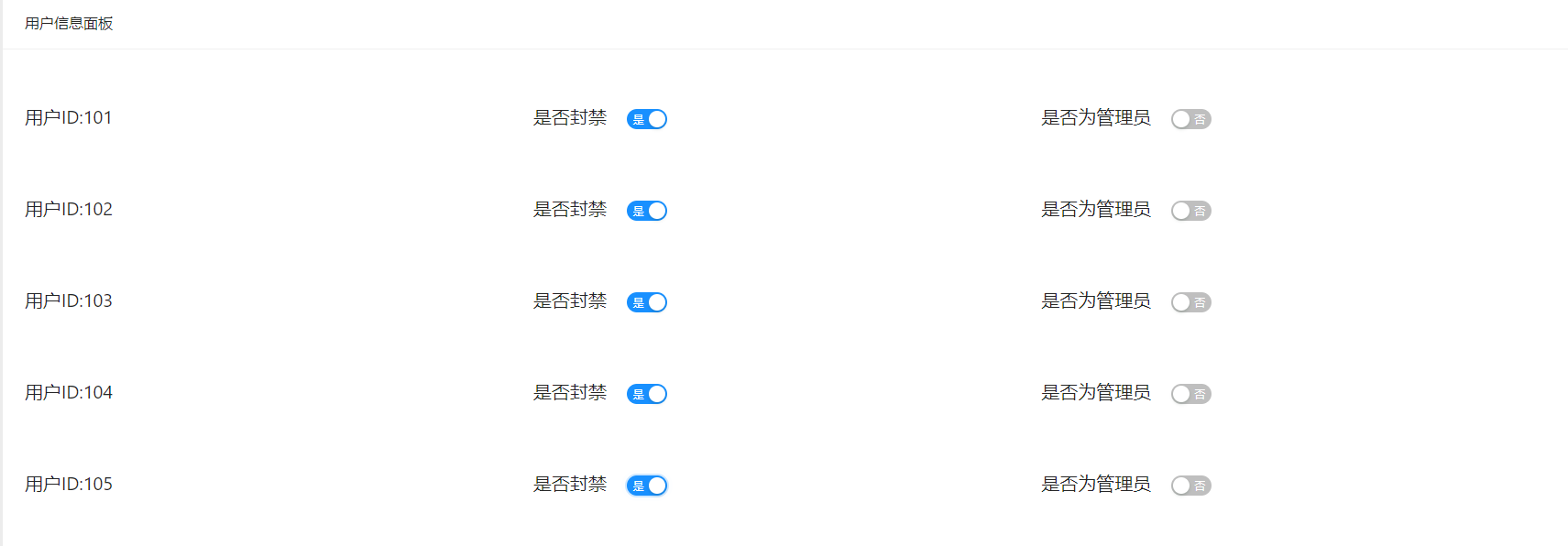
**对评论者进行封号**

是

否

**浏览下一个**

**3.11.3 封号功能界面**



**3.11.4 封号功能核心代码**

// 用户名框的状态

public FormValidateStatus NameStatus { get; set; } = FormValidateStatus.Default;

private User user=new User();

// 判断用户名是否正确

private void JuageName()

{

// 判断条件

bool hasName=false;

// 获取用户表

DataTable dt=DBHelper.GetData("select \* from UserTable");

// 遍历用户表

foreach (DataRow dr in dt.Rows)

{

// 若不存在输入的用户名,将hasName改为false

hasName = user.UserName != dr["Username"].ToString() ? false : true;

}

// 如果存在用户名

if (hasName)

{

NameStatus = FormValidateStatus.Success;

}

// 如果不存在用户名

else

{

NameStatus = FormValidateStatus.Error;

}

}

**3.12 帖子置顶功能模块**

**3.12.1 帖子置顶功能简介**

当吧主发现某个类型的的帖子比较优秀是，可以将此帖子在当前类型的区域进行置顶操作，并且此帖子还会被添加到精华区域板块。

**3.12.2 帖子置顶功能流程图**

**吧主浏览帖子**

评论是否非常优秀

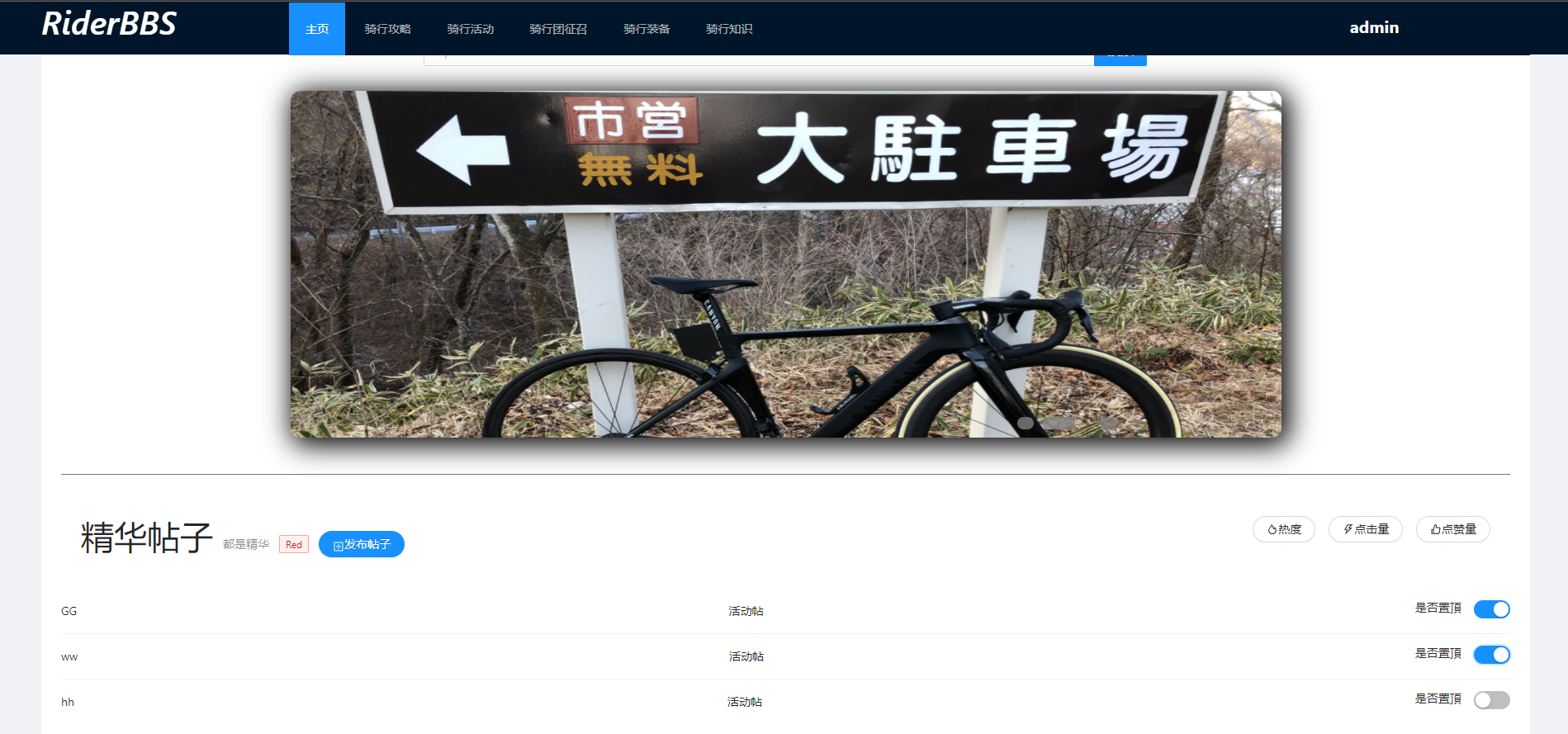
**对帖子进行置顶**

是

否

**浏览下一个**

**3.12.3 帖子置顶功能界面**



**3.12.4 帖子置顶功能核心代码**

// 用户名框的状态

public FormValidateStatus NameStatus { get; set; } = FormValidateStatus.Default;

private User user=new User();

// 判断用户名是否正确

private void JuageName()

{

// 判断条件

bool hasName=false;

// 获取用户表

DataTable dt=DBHelper.GetData("select \* from UserTable");

// 遍历用户表

foreach (DataRow dr in dt.Rows)

{

// 若不存在输入的用户名,将hasName改为false

hasName = user.UserName != dr["Username"].ToString() ? false : true;

}

// 如果存在用户名

if (hasName)

{

NameStatus = FormValidateStatus.Success;

}

// 如果不存在用户名

else

{

NameStatus = FormValidateStatus.Error;

}

}

1. **系统集成测试**

**第四章 系统集成测试**

1. 测试要求和环境

要求：

测试的依据为总体方案中规定的各单元功能与性能；

测试的内容为整个项目按照逻辑划分成的各个单元模块；

测试的条件为各单元模块测试的环境要求；

测试的方法为人工输入数据。

环境：

测试的硬件环境：CPU:酷睿四核,内存4G,硬盘500G；

测试的软件环境：Windows 10，Visual studio2019。

1. 单元测试
2. 登录功能测试

登录功能是进入系统的一个必要的前提，它很大程度的关系到了系统的安全性，现模拟用户身份，测试登录功能是否可以正常运行，具体功能测试用例如下表4.1所示。

表4.1登录测试用例

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 步骤 | 操作描述 | 输入 | 预期输出 | 实际输出 | 是否通过 |
| 1 | 不输入账号和密码 | 账号、密码为空 | 提示账号和密码不能为空 | 提示账号密码不能为空 | 是 |
| 2 | 输入密码但不输入账号 | 账号为空，密码123456 | 提示账号和密码不能为空 | 提示账号密码不能为空 | 是 |
| 3 | 输入账号但不输入密码 | 密码为空，账号admin | 提示账号和密码不能为空 | 提示账号密码不能为空 | 是 |
| 4 | 输入账号和密码 | 账号admin，密码123456 | 提示登录成功 | 提示登录成功 | 是 |

1. 注册功能测试

注册功能指对于管理员实现添加操作，现对该模块进行测试。具体功能测试用例如下表4.2所示。

表4.2登录测试用例

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 步骤 | 操作描述 | 输入 | 预期输出 | 实际输出 | 是否通过 |
| 1 | 不输入信息 | 无 | 提示账号和密码不能为空 | 提示账号密码不能为空 | 是 |
| 2 | 只输入密码 | 密码123456 | 提示账号和密码不能为空 | 提示账号密码不能为空 | 是 |
| 3 | 只输入账号 | 账号admin | 提示账号和密码不能为空 | 提示账号密码不能为空 | 是 |
| 4 | 输入账号、密码和确认密码 | 账号admin，密码123456，确认密码123456 | 提示登录成功 | 提示登录成功 | 是 |
| 5 | 输入账号和密码 | 账号admin，密码123456 | 提示确认密码不能为空 | 提示确认密码不能为空 | 是 |
| 6 | 输入账号、密码和确认密码 | 账号admin，密码123456，确认密码123 | 提示密码和确认密码要一致 | 提示密码和确认密码要一致 | 是 |

1. 主页功能测试

该功能模块包括鼠标悬停导航效果，单击导航选项链接是否可以正确跳转到相应的功能模块等，现对该模块进行测试。具体功能测试用例如下表4.3所示。

表4.3主页功能测试用例

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 步骤 | 操作描述 | 预期结果 | 实际结果 | 是否通过 |
| 1 | 鼠标悬停导航 | 显示成功 | 显示成功 | 是 |
| 2 | 单击导航链接 | 跳转页面成功 | 跳转页面成功 | 是 |

1. 帖子分类导航栏功能测试

该功能模块包括鼠标悬停导航效果，单击导航选项链接是否可以正确跳转到相应的功能模块，选项卡能正常切换，现对该模块进行测试。具体功能测试用例如下表4.4所示。

表4.4帖子分类测试用例

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 步骤 | 操作描述 | 预期结果 | 实际结果 | 是否通过 |

1. 帖子浏览功能测试

表4.5帖子浏览功能测试用例

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 步骤 | 操作描述 | 预期结果 | 实际结果 | 是否通过 |
| 1 | 鼠标悬停帖子标题 | 显示成功 | 显示成功 | 是 |
| 2 | 单击帖子标题 | 跳转页面成功 | 跳转页面成功 | 是 |
| 3 | 查看帖子内容 | 内容与预期一致 | 内容与预期一致 | 是 |

1. 用户功能测试

浏览者身份：该功能模块包括浏览帖子功能,能否发布帖子，

网页用户身份：

表4.6骑行者功能测试用例（浏览者）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 步骤 | 操作描述 | 预期结果 | 实际结果 | 是否通过 |

表4.7骑行者功能测试用例（网站用户）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 步骤 | 操作描述 | 预期结果 | 实际结果 | 是否通过 |

1. 骑行团加入功能测试

表4.8骑行团加入功能测试用例（浏览者）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 步骤 | 操作描述 | 预期结果 | 实际结果 | 是否通过 |

表4.9骑行团加入功能用例（网页用户）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 步骤 | 操作描述 | 预期结果 | 实际结果 | 是否通过 |

1. 管理员功能测试

表4.10管理员功能测试用例

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 步骤 | 操作描述 | 预期结果 | 实际结果 | 是否通过 |

1. 吧主功能测试

表4.11吧主功能测试用例

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 步骤 | 操作描述 | 预期结果 | 实际结果 | 是否通过 |

1. 性能测试

本系统是B/S架构系统，测试的重点是服务器端。具体测试环境如下表4.14所示。

表4.14 软硬件环境

|  |  |  |  |
| --- | --- | --- | --- |
| 环境配置 | 应用服务器 | 数据库服务器 | 客户端 |
| 硬件配置 | CPU:AMD Ryzen 7 4800U with Radeon Graphics 1.80 GHz  Memory: 16.0 GB | CPU:AMD Ryzen 7 4800U with Radeon Graphics 1.80 GHz  Memory: 16.0 GB | CPU:AMD Ryzen 7 4800U with Radeon Graphics 1.80 GHz  Memory: 16.0 GB |
| 软件配置 | Windows 10：Microsoft Visual Studio 2019 | Windows 10 : Microsoft SQL Server Management Studio 18 | Windows 10：Google Chrome |
| 网络配置 | 10MLAN | 10MLAN | 10MLAN |

4.3.1 性能测试目的

1. 评估系统的能力
2. 识别体系中的弱点
3. 系统调优
4. 检查软件中的问题
5. 验证系统稳定性
6. 验证系统可靠性

4.3.2 测试结论分析

经过多次测试和数据报表分析，可以得出如下结论：

1. 当总体并发用户数为250-300时，系统具有最优性能表现；当事务并发数超过300时，事务失败率整体上升，系统到达性能拐点。
2. 在小压力条件下（10并发），最大事务响应时间为查询用户信息事务的2042毫秒，平均在600毫秒左右系统。整体事务微观响应速度较优。
3. 系统资源方面，内存占用率始终处于高位水平（90%以上），磁盘空间由于日志写入而不断被占用。
4. 内存占用处于高位水平，需要进一步探查原因。

**参考文献**

**总结**

**致谢**

这次的毕业设计经历了较长的一段时间，在这段时间中，我得到了很多人的帮助，比如有同学们的无私帮助，也有老师们的细心指导。在此我对他们表示衷心的感谢！

首先，我要感谢我的毕业设计指导老师高子童老师，是他对我的毕业设计的写作提供了很大的指导和帮助，在我编写论文的整个过程中，无论是从组织还是到内容，都有在耐心地指导着我们，自己被认真看待，真的非常感谢程老师这般的敬业，在此向指导老师高子童老师表示深深的谢意。

然后我要感谢我的任课老师蒋燕纯老师，帮我指明了毕业设计的选题，正是因为有了老师先前在课堂上和在平时生活中的细心指导，我才能在最开始的时候就这么顺利的开始设计我的毕业设计，老师的指导会一直影响和激励着我今后的工作与生活。

最后再次向老师们表示感谢！