分类号：TP311.5UDC：D10621-408-(2019)2378-0

密　级：公开编号：2015081121

成都信息工程大学

学位论文

基于WEB的游戏分享平台的设计与实现

|  |  |
| --- | --- |
| 论文作者姓名： | 宋钦 |
| 申请学位专业： | 软件工程 |
| 申请学位类别： | 工学学士 |
| 指导教师姓名（职称）： | 王嘉昀（讲师） |
| 论文提交日期： | 2019年5月20日 |

基于WEB的游戏分享平台的设计与实现

**摘　要**

随着全球游戏行业的高速发展与游戏产品质量的不断提高，越来越多的人将游戏作为丰富业余生活的娱乐方式之一 ,游戏产业的发展甚至带动了电子竞技职业的发展。不过目前游戏种类繁多，同时游戏的质量也是参差不齐，人们很难独自发现一款优质的游戏。

“linklink游戏分享平台”是专门为游戏爱好者提供的一个玩家之间相互分享，鉴赏游戏的第三方平台，玩家可以通过该平台发现和分享自己喜欢的游戏，搜索获取最新的游戏资讯，并且通过该平台，玩家与玩家，玩家与游戏官方之间可以相互讨论游戏相关的话题。平台旨在统一收集各渠道游戏资讯，为玩家们提供发现好游戏的机会和良好的游戏体验环境。

本系统基于B/S模式，使用MySQL作为数据库,后端框架则是使用了时下流行的Java开源框Spring Boot。平台实现了以游戏分享为核心的多个功能。玩家可以浏览或者搜索游戏信息及游戏相关文章，管理个人信息，发布及管理评论，发布及管理评价，同时也可以在相应的游戏版块发布文章，分享游戏心得和游戏攻略。管理员的功能主要包括：账号管理，游戏信息管理，评论管理，评价管理，文章管理，以及主页展示信息管理等等。

**关键词：**游戏分享；游戏资讯 ；开源；Spring Boot；MySQL

Design and Implementation of Game Sharing Platform Based on WEB

**Abstract**

With the rapid development of the global game industry and the continuous improvement of the quality of game products, more and more people regard games as one of the entertainment ways to enrich amateur life, and the development of the game industry has even led to the development of e-sports profession. However, there are many kinds of games at present, and the quality of the games is uneven. It is difficult for people to find a high-quality game.

The "linklink Game sharing platform" is a third-party platform for game enthusiasts to share each other and appreciate games, through which players can discover and share their favorite games and search for the latest game information. And through the platform, players to players, players to game officials can discuss game-related topics. The platform aims to collect game information from all channels and provide players with the opportunity to find good games and a good game experience environment.

The "linklink Game sharing platform" is based on B / S mode, using MySQL as the database, while the back-end framework uses the popular Java open source box Spring Boot.. The platform realizes many functions with game sharing as the core. Players can browse or search game information and game-related articles, manage personal information, publish and manage reviews, publish and manage evaluation, and also publish articles in the corresponding game section to share game experience and game introduction. The functions of the administrator mainly include: account management, game information management, comment management, evaluation management, article management, and home page display information management and so on.

**Key words:** Game sharing; Game information；Open Source；Spring Boot；MySQL；

**目 录**

论文总页数：33页

[1 引言 1](#_Toc9202832)

[1.1 课题背景与意义 1](#_Toc9202833)

[1.2 国内外研究现状 1](#_Toc9202834)

[1.3 本课题的研究方法 1](#_Toc9202835)

[2 需求分析 1](#_Toc9202836)

[2.1 系统需求概述 1](#_Toc9202837)

[2.2 系统用户角色分析 2](#_Toc9202838)

[2.3 系统功能性分析 4](#_Toc9202839)

[2.4 可行性分析 4](#_Toc9202840)

[3 系统总体设计 5](#_Toc9202841)

[3.1 系统总体架构 5](#_Toc9202842)

[3.2 系统数据库设计 5](#_Toc9202843)

[3.2.1 数据库逻辑设计 5](#_Toc9202844)

[3.2.2 数据库物理结构模型 6](#_Toc9202845)

[3.2.3 数据库表结构设计 7](#_Toc9202846)

[3.3 系统功能设计 11](#_Toc9202847)

[3.4.1 个人信息管理模块 12](#_Toc9202848)

[3.4.2 游戏信息管理模块 12](#_Toc9202849)

[3.4.3 游戏评价管理模块 14](#_Toc9202850)

[3.4.4 评论管理模块 16](#_Toc9202851)

[3.4.5 文章管理模块 17](#_Toc9202852)

[3.4.6 搜索模块 17](#_Toc9202853)

[3.4 系统技术难点及关键技术 18](#_Toc9202854)

[3.5 系统优点 19](#_Toc9202855)

[4 系统实现 19](#_Toc9202856)

[4.1 系统开发环境 19](#_Toc9202857)

[4.2 个人信息管理模块的实现 20](#_Toc9202858)

[4.3 游戏信息管理模块的实现 22](#_Toc9202859)

[4.4 游戏评价管理模块的实现 23](#_Toc9202860)

[4.4 文章管理模块的实现 25](#_Toc9202861)

[4.5 评论管理模块的实现 25](#_Toc9202862)

[5 系统测试 27](#_Toc9202863)

[5.1 测试环境 27](#_Toc9202864)

[5.2 测试用例及过程 27](#_Toc9202865)

[5.2.1 游戏信息管理模块测试 27](#_Toc9202866)

[5.2.2 游戏评价管理模块测试 28](#_Toc9202867)

[5.2.3 评论管理模块测试 28](#_Toc9202868)

[5.2.4 文章管理模块测试 29](#_Toc9202869)

[5.3 测试结果分析 29](#_Toc9202870)

[结 束 语 30](#_Toc9202871)

[参考文献 31](#_Toc9202872)

[致　　谢 32](#_Toc9202873)

[声　　明 33](#_Toc9202874)

# 引言

## 课题背景与意义

随着全球游戏行业的高速发展与游戏产品质量的不断提高，电子游戏尤其是网络游戏越来越被大多数人作为娱乐的第一选择[1]。我国的游戏产业发展较好，在最新的游戏产业收入报告中 ，仅我国的游戏产业就创造了379亿美元的收入，占全球游戏市场收入的28%[2]。不过，虽然国内游戏行业蒸蒸日上，但国内很少有平台对游戏进行分享与鉴赏，多为以盈利为目的相互独立的游戏渠道或平台[3]。

“linklink游戏分享平台”是一个基于WEB的游戏资讯平台，方便玩家使用和获取游戏资讯。可以让玩家省去四处寻找发现游戏的时间和精力，为玩家提供更优质的游戏体验。平台旨在统一收集各渠道游戏讯息，为玩家们提供发现好游戏的机会和良好的游戏体验环境，玩家可以通过该平台发现和分享自己喜欢的游戏，寻找志趣相投的小伙伴，同时可以获取最新的游戏资讯。

## 国内外研究现状

国内缺乏像国外的IGN这种专业的游戏评分机构，而国外有许多专业的游戏评分机构和玩家交流平台和论坛[4]，虽然国内游戏论坛在近些年也如雨后竹笋，随着国内游戏行业的发展而逐渐增多--第三方论坛如NGA游戏论坛，游戏网，3dm游戏网以及各大游戏公司自家的游戏论坛如网易游戏论坛，多玩游戏论坛等等，但是还是没有专业的评分机构[5]。

在国外，游戏的发行通常只有Google PlayStore与AppStore这两个渠道，但在国内，同一个游戏需要发行多达10多种不同的渠道版本[6]，使得玩家为了寻找和发现高品质的游戏不得不往来于各个游戏推广渠道和平台之间，游戏开发人员也需要对各个渠道版本进行适配，浪费了许多人力和物力，长此以往，游戏研发成本会逐渐增加，十分不利于游戏行业的良性发展。

## 本课题的研究方法

平台基于Spring Boot 框架开发，后台使用Java作为开发语言，前端页面使用Freemaker + JQuery进行数据交互，使用插件LayUI作为页面模板进行数据展示，使用LayUI的LayEdit插件作为富文本编辑器，Video-JS插件提供视频播放支持。

# 需求分析

## 系统需求概述

本系统主要用户为普通玩家，游戏管理员，网站管理员。

主要需求分析如下:

1. 玩家信息管理，玩家对个人的信息管理和查看。
2. 游戏信息管理，玩家可以通过游戏详情页了解游戏相关信息，并且可以

通过详情页直接下载或者跳转到游戏官网进行下载。管理员可进行游戏信息管理。

1. 游戏评价功能，用户可以对所有的游戏进行评分，评分会直接影响游戏排名的名次。
2. 评论功能，用户可以对游戏，文章，评价进行评论。
3. 文章发布及管理，玩家可以发布与游戏相关的文章进行游戏内容分享和推荐。
4. 检索功能，根据关键字全文检索。

## 系统用户角色分析

本系统主要使用角色分为普通玩家，游戏管理员，网站管理员。普通玩家可以浏览游戏信息，发布评论评价。游戏管理员可以管理相关游戏的评论及文章。网站管理员可以进行账号管理，主页展示信息管理，文章，评论以及评价管理[7]。

根据用户角色分析，得出如下角色用例图。玩家用例图如图2-1所示，

游戏管理员用例图如图2-2所示，网站管理员用例图如图2-3所示。



图2-1玩家用例图



图2-2游戏管理员用例图



图2-3网站管理员用例图

## 系统功能性分析

本系统主要功能划分包含个人信息管理，游戏信息管理，游戏评价管理，

评论管理以及文章管理和搜索功能。

（1）个人信息管理

当玩家登录成功之后，玩家可以对自己的个人资料进修改，也可以上传图片修改个人头像。

（2）游戏信息管理

普通玩家可以根据类型或者主页上的推荐查看游戏。游戏管理员和网站管理员可以修改游戏信息。

（3）游戏评价管理

玩家可以对游戏进行评价，评价包含评价内容与评分两个部分，评分将会影响游戏的排行榜名次。网站管理员可以进行管理。

（4）评论管理

玩家用户可以对游戏评价，文章添加评论并管理。管理员能对评论进行管理。

（5）文章管理

用户可以在游戏社区内发布游戏心得分享，游戏推荐等文章。用户与管理员可对文章进行管理。

（6）搜索功能

根据输入的关键字进行检索，在搜索页面查看搜索结果。

## 可行性分析

为了避免资源的浪费，在平台系统搭建之前必须进行全面的可行性分析。本系统的可行性分析主要包含以下两个方面：

1.经济可行性

本平台开发所使用的框架，插件，软件为免费或者开源的。整个项目的计划与实现都由本人独立完成，无额外的开销。平台也只允许在本地计算机上，不需要额外购买服务器搭建平台，所以综上分析，开发该平台的成本极低，故经济上可行[8]。

2.技术可行性

该平台的后端开发使用了当下流行的Java开发框架Spring Boot，该框架简单易学，学习成本低，上手快，没有如Spring MVC框架那么繁杂的配置要求。

数据库则是使用了MySQL，此数据库比较适合中小型项目，对于本系统而言十分合适[9]。

前端的页面设计完全参考TapTap游戏网站[10]，使用LayUI前端框架复现页面，使用Video-JS提供视频播放支持，使用LayUI提供的富文本编辑器对评论，文章进行编辑。所有的插件都能即学即用[11]。故技术方面的可行性是完全没问题的。

# 系统总体设计

## 3.1 系统总体架构

平台采用的Spring Boot框架整合了Spring MVC，MyBatis，Tomcat。

由Model层负责视图的数据传递，Controller进行逻辑判断，Controller调用Service接口进行业务逻辑处理，Service调用Mapper进行数据持久化处理[12]。

系统后端总体架构图如图3-1所示：



图3-1 系统总体架构图

## 3.2 系统数据库设计

### 3.2.1 数据库概念设计

根据需求分析，一共设计了6个实体：

（1）用户（User）：描述用户信息的实体，属性包括编号，昵称，密码，头像，个人介绍，性别，联系电话等等。

（2）评论（Comment）：描述评论信息的实体，属性包括评论id，评论者id，回复者id，所属游戏id，所属评价id，所属文章id，评论内容，时间等等。

（3）评价（Evaluate）：描述游戏评价信息的实体，属性包括评价id，评价者id，所属游戏id，内容，评分，评价时间，更新时间。

（4）文章（Topic）：描述文章信息的实体，属性包括文章id，发布者id，所属游戏id，标题，内容，时间等等。

（5）游戏（Game）：描述游戏信息的实体，属性包括游戏id，游戏名称，标签，图标，类型，开发者，发行者，厂商，得分，发布时间，更新介绍，更新时间等。

（6）厂商（Manufacturer）：描述游戏厂商信息的实体，属性包括厂商id，名称，图标，介绍，得分，所有游戏id。

根据实体间的联系，得出数据库整体E-R图如图3-2所示。



图3-2 系统整体E-R图

### 3.2.2 数据库物理结构模型

本系统一共设计了8张数据库表，数据库物理结构如图4-1所示：



图3-3 数据库物理结构图

### 3.2.3 数据库表结构设计

本系统使用MySQL数据库，一共有8张数据库表。各表信息如下：

1. 用户信息表（User）

存放用户信息，如头像，昵称，密码等等

用户信息表（User）的设计如表3-1所示：

表3-1 用户信息表（User）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 中文名称 | 列名 | 数据类型 | 主键 | 非空 | 外键 |
| 1 | 编号 | uid | int | √ | √ | × |
| 2 | 姓名 | username | varchar | × | √ | × |
| 3 | 密码 | password | varchar | × | √ | × |
| 4 | 头像 | iconpath | varchar | × | √ | × |
| 5 | 签名 | sign | varchar | × | √ | × |
| 6 | 状态 | state | int | × | √ | × |
| 7 | 等级 | lv | varchar | × | √ | × |
| 8 | 经验 | exp | varchar | × | × | × |
| 9 | 电话 | phone | varchar | × | × | × |
| 10 | 性别 | sex | int | × | × | × |
| 11 | 邮箱 | email | varchar | × | × | × |
| 12 | Qq | QQ | int | × | × | × |
| 13 | Ip | IP | varchar | × | × | × |
| 14 | 注册时间 | regtime | datetime | × | √ | × |
| 15 | 最后登录时间 | lastlogin | datetime | × | √ | × |
| 16 | 游玩的游戏 | played\_uids | varchar | × | √ | × |
| 17 | 最喜欢的游戏 | most\_played\_uids | varchar | × | × | × |
| 18 |  | reserved\_uids | varchar | × | × | × |
| 19 | 评论数 | comment\_num | int | × | × | × |
| 20 | 发布文章数 | posts\_num | int | × | × | × |

1. 管理员表（Admin）

用于记录管理成员的信息以及权限的信息。

管理员表（Admin）的设计如表3-2所示:

表3-2 管理员表（Admin）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 中文名称 | 列名 | 数据类型 | 主键 | 非空 | 外键 |
| 1 | 编号 | uid | int | √ | √ | × |

续表3-2

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 2 | 姓名 | username | varchar | × | √ | × |
| 3 | 密码 | password | varchar | × | √ | × |
| 4 | 是否为超级管理员 | isSuper | Int | × | √ | × |
| 5 | 上次登录时间 | Slastlogin | Datetime | × | √ | × |

1. 游戏置顶表（Top）

用于记录首页的游戏信息展示

游戏置顶表（Top）的设计如表3-3所示:

表3-3 游戏置顶表（Top）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 中文名称 | 列名 | 数据类型 | 主键 | 非空 | 外键 |
| 1 | 编号 | uid | int | √ | √ | × |
| 2 | 游戏编号 | Gid | Int | × | √ | × |
| 3 | 是否推荐 | Recommend | Int | × | √ | × |
| 4 | 是否主页滚动显示 | Home\_page\_top | Int | × | √ | × |
| 5 | 上次登录时间 | Update\_time | Datetime | × | √ | × |

1. 评论表（Comment）

用于记录评论信息，包括评论内容时间，评论者uid，评论所属版本等等。

评论表（Comment）的设计如下表3-4所示：

表3-4 评论表（Comment）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 中文名称 | 列名 | 数据类型 | 主键 | 非空 | 外键 |
| 1 | 编号 | uid | int | √ | √ | × |
| 2 | 评论者uid | commenter\_uid | int | × | √ | × |
| 3 | 被回复者uid | reciver\_uid | int | × | √ | × |
| 4 | 游戏uid | game\_id | int | × | √ | × |
| 5 | 评价uid | evaluate\_uid | int | × | × | × |
| 6 | 文章uid | topic\_id | int | × | × | × |
| 7 | 厂商uid | manufacturer\_uid | int | × | × | × |
| 8 | 内容 | content | text | × | √ | × |
| 9 | 时间 | time | datetime | × | √ | × |
| 10 | 类型 | type | int | × | √ | × |

续表3-4

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 11 | 点赞 | greate\_num | int | × | × | × |
| 12 | 踩 | diss\_num | int | × | × | × |
| 13 | 状态 | status | int | × | × | × |

1. 评价表（Evaluate）

用于记录评价信息，包括评价内容，评分，评价所属游戏，评价者等信息。

评价表（Evaluate）的设计如表3-5所示：

表3-5 评价表（Evaluate）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 中文名称 | 列名 | 数据类型 | 主键 | 非空 | 外键 |
| 1 | 编号 | id | int | √ | √ | × |
| 2 | uid | uid | int | × | √ | × |
| 3 | 游戏uid | game\_id | int | × | √ | × |
| 4 | 内容 | content | text | × | √ | × |
| 5 | 评分 | score | float | × | √ | × |
| 6 | 点赞数 | great\_num | int | × | × | × |
| 7 | 踩 | diss\_num | int | × | × | × |
| 8 | 评价时间 | evaluate\_time | datetime | × | √ | × |
| 9 | 修改时间 | update\_time | datetime | × | √ | × |

1. 文章表（Topic）

记录了文章的内容，包括文章包含的图片，视频，所属模块等信息

文章表（Topic）的设计如表3-6所示：

表3-6 文章表（Topic）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 中文名称 | 列名 | 数据类型 | 主键 | 非空 | 外键 |
| 1 | 文章uid | Topic\_uid | int | √ | √ | × |
| 1 | 发布者uid | topicer\_uid | int | √ | √ | × |
| 2 | 所属游戏 | topic\_game\_uid | int | × | √ | × |
| 3 | 标题 | title | varchar | × | √ | × |
| 4 | 内容 | content | text | × | √ | × |
| 5 | 是否为官方 | is\_office | int | × | × | × |
| 6 | 是否置顶 | is\_top | int | × | × | × |
| 7 | 点赞数 | great\_num | int | × | × | × |
| 8 | 踩数 | diss\_num | int | × | × | × |
| 9 | 关注数 | concer\_num | int | × | √ | × |

续表3-6

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 10 | 评论数 | comment\_num | int | × | √ | × |
| 11 | 视频地址 | video\_path | varchar | × | √ | × |
| 12 | 时间 | time | datetime | × | × | × |
| 13 | 更新时间 | update\_time | datetime | × | × | × |

1. 游戏信息表（Game）

用于记录游戏信息，包含游戏名称，icon，介绍，厂商，等等信息。

游戏信息表（Game）的设计如表3-7所示：

表3-7 游戏信息表（Game）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 中文名称 | 列名 | 数据类型 | 主键 | 非空 | 外键 |
| 1 | 游戏uid | uid | int | √ | √ | × |
| 2 | 名称 | name\_cn | varchar | × | √ | × |
| 3 | 名称 | name\_en | varchar | × | √ | × |
| 4 | 标签 | tags | varchar | × | √ | × |
| 5 | 图标 | icon | varchar | × | × | × |
| 6 | 游戏类型 | game\_type | int | × | × | × |
| 7 | 开发者 | developer | varchar | × | × | × |
| 8 | 发布者 | publisher | varchar | × | × | × |
| 9 | 发布时间 | publish\_time | datetime | × | × | × |
| 10 | 备注 | remarks | varchar | × | × | × |
| 11 | 安装人数 | installation\_num | int | × | √ | × |
| 12 | 关注数 | concer\_num | int | × | √ | × |
| 13 | 下载地址 | andorid\_download\_url | varchar | × | √ | × |
| 14 | ios\_download\_url | varchar | × | √ | × |
| 15 | 最后评论时间 | latest\_comment\_time | datetime | × | √ | × |
| 16 | 评价数 | evaluate\_num | int | × | × | × |
| 17 | 评论数 | comment\_num | int | × | × | × |
| 18 | 开发者的话 | developer\_words | text | × | × | × |
| 19 |  | communication | varchar | × | × | × |
| 20 | 视频地址 | game\_video\_path | varchar | × | × | × |
| 21 | 宣传图片 | main\_img | varchar | × | × | × |
| 22 | 宣传图 | game\_imgs\_path | varchar | × | × | × |

续表3-7

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 23 | 介绍 | introduction | text | × | √ | × |
| 24 | 大小 | app\_size | varchar | × | √ | × |
| 25 | 版本 | app\_version | varchar | × | √ | × |
| 26 | 更新时间 | update\_time | datetime | × | √ | × |
| 27 | 厂商 | manufacturer | varchar | × | √ | × |
| 28 | 官网地址 | office\_url | varchar | × | √ | × |
| 29 | 专业评分数 | major\_score | float | × | √ | × |
| 30 | 评分 | score | float | × | √ | × |
| 31 | 最近评分 | latest\_score | float | × | √ | × |
| 32 | 是否置顶 | top | int | × | √ | × |
| 33 | 是否推荐 | recommend | int | × | √ | × |
| 34 | 管理者id | manager\_uid | varchar | × | × | × |

1. 厂商表（Manufacturer）

主要用于记录厂商信息，包含厂商介绍，所有游戏，icon等等

厂商表（Manufacturer）的设计如表3-8所示：

表3-8 厂商表（Manufacturer）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 中文名称 | 列名 | 数据类型 | 主键 | 非空 | 外键 |
| 1 | uid | uid | varchar | √ | √ | × |
| 2 | 游戏id | games\_uid | varchar | × | √ | × |
| 3 | 厂商名称 | name | varchar | × | √ | × |
| 4 | 图标 | icon | varchar | × | √ | × |
| 5 | 信息 | info | varchar | × | × | × |
| 6 | 评分 | score | float | × | × | × |
| 7 | 关注数 | fans | int | × | × | × |

## 3.3 系统功能设计

根据需求分析，本平台按使用角色可以划分出以下功能结构图。

系统功能结构图如图3-4所示。



图3-4 系统总体功能模块图

### 3.4.1 个人信息管理模块

个人信息管理模块包含修改个人资料，修改个性签名，上传头像三个子功能。用户可以通过此功能修改如qq，电话，邮箱等资料，也可选择本地图片上传作为用户头像。

用户需登录后，通过点击头像下拉选项框进入个人中心，点击修改资料按钮进入资料修改页面，在此页面，用户可以选择修改个人头像，点击头像打开本地上传图片窗口，在选择正确的图片文件后点击上传即可成功修改个人头像。个人资料的修改在所输入的数据得到校验成功后才能继续进行。

个人信息管理流程图如图3-5所示。

### 3.4.2 游戏信息管理模块

游戏信息管理模块为游戏管理员和网站管理员使用的功能，该功能包含添加新游戏，修改游戏信息，删除游戏信息，检索游戏信息，浏览游戏等子功能。

游戏信息管理功能结构图如图3-6所示。

游戏信息管理需在用户登录后进行，需要游戏管理员或网站管理员权限才能对游戏信息进行修改。管理员通过后台登录界面登入，进入游戏信息管理页，首先通过游戏id或者游戏名称查询游戏信息，选择相应游戏点击修改按钮进入游戏信息修改页面。若游戏不存在，可以点击发布新游戏按钮添加游戏。

游戏信息管理流程图如图3-7所示。



图3-5 个人信息管理流程图



图3-6 游戏信息管理功能结构图



图3-7 游戏信息管理流程图

### 3.4.3 游戏评价管理模块

游戏评价管理包含两个子模块，分别为发布评价，历史评价。一个用户对同一游戏只能发布一个评价，不允许多次评价。游戏评价包含评价内容与游戏评分两个部分组成，评价内容描述用户对该游戏的试玩体验，评分则为玩家对游戏的整体试玩感觉打分，游戏的评分会影响该游戏的排行榜名次。游戏评价管理功能结构图如图3-8所示。

玩家登录后，进入游戏详情页面，若未对游戏进行过评价，则可以发布评价，若已发布过，则只能对评价进行修改和删除操作。游戏评价管理流程如图3-9所示。



图3-8 游戏评价管理功能结构图



图3-9 游戏评价管理流程图

### 3.4.4 评论管理模块

评论管理包含两个子模块，分别为发布评论，历史评论。其中历史评论模块包含查看评论，删除评论两个子功能。

游戏评论管理功能结构图如图3-10所示。

玩家登录后，进入个人中心，通过浏览历史评论页面对历史评论进行查看，删除操作。评论管理流程图如图3-11所示。



图3-10评论管理功能结构图



图3-11 评论管理流程图

### 3.4.5 文章管理模块

文章管理包含两个子模块，分别为发布文章，历史文章。其中历史文章模块包含查看评论，修改文章，删除评论三个子功能。

文章管理功能结构图如图3-12所示：



图3-12 文章管理功能结构图

玩家登录过后，可以通过游戏信息页进入对应的游戏社区，在社区内，点击发布帖子按钮可以进入文章编辑页面，数据输入无误后点击发布按钮即可发布新文章。玩家也可以通过个人中心查看历史文章，并对文章作先用的修改或删除操作。文章管理流程图如图3-13所示。

### 3.4.6 搜索模块

搜索模块提供关键字检索功能，玩家可以通过输入关键字搜索相关讯息，包括游戏信息，文章信息，厂商信息等等。

该功能无需登录即可使用，在每个页面的顶部导航栏的搜索框中输入相应的关键字，点击搜索按钮后，系统在搜索结果列表列出根据关键字检索出的游戏，帖子或者厂商等信息。



图3-13 文章管理流程图

## 3.4 系统技术难点及关键技术

本系统的关键技术难点在于评论的展示和文件上传两个方面。

评论展示：本平台中很多模块都有评论功能，比如玩家的游戏评价可以被评论，文章里可以有评论，厂商页面也有评论，评论中也有评论。关键在于评论的数据结构的设计，评论包含了评论内容，评论者的信息与被评论者的信息，涉及到了三个实体数据，处理起来稍微麻烦。评论中的评论嵌套展示也是个问题，在参考TapTap的评论展示方式后，决定采用TapTap的展示方法，及评论中的评论单独显示，不以嵌套的方式展示，这样页面相对于评论嵌套展示的页面简洁清晰。

文件上传：在个人信息中，玩家可以通过选择本地图片上传作为头像，游戏信息中也可以选择多图上传与视频上传。单图片的上传后端可以使用Java的MultipartFile接收数据，前端可以调用LayUI自带的文件上传接口。同理，视频上传功能也可如此实现。难点在于多图上传中，如何取消已选择的图片与多图片的上传路径的存储。取消已选择的图片通过前端JS代码实现，图片路径采用字符串存储，各个图片的路径间使用分号分隔开，在使用时，后端会进行字符串处理，将各个图片的路径分隔开成为单独的数据。

## 3.5 系统优点

本系统主要有适用性高，开发及维护成本低，安全性高的优点。

(1)适用性高。该系统为游戏分享平台，现如今，游戏已经成为年轻一代的首选娱乐方式，并且玩家还会越来越多。大量的玩家需要一个共同交流沟通的平台，而该系统恰好提供这样一个平台，所以，对于该系统来说，潜在的用户群体以及用户数量是十分巨大的。

(2)开发及维护成本低。该系统的设计与实现均由一个人独自完成，没有额外的软硬件开销，开发成本极低。项目上线后的维护工作也很少，一个网站管理员就能处理完所有工作。

(3)安全性高。该系统有严格的权限控制，普通玩家用户只能对本身的数据进行操作，没有特别权限无法进入后台管理，游戏管理员则只拥有相对应的游戏社区的权限，无法对非本游戏的数据进行操作。所有权限的设置由网站管理员进行。同时，系统会对输入数据进行校验，防止用户插入恶意数据或代码。

# 系统实现

## 4.1 系统开发环境

本平台在Windows10专业版开发环境下进行，编辑器使用了当下最流行的Java EE编辑器之一的Intellij IDEA 2018。该编辑器简单易学，许多公司也都以该开发软件作为公司的第一编辑器。

为了更加快速的进行项目搭建，本平台使用了Spring Boot框架，Spring Boot是一款十分优秀的快速构建项目的开源框架，该框架省去了Spring MVC框架所需的复杂配置，仅需少量的配置文件就能解决。同时，Spring Boot集成了Tomcat服务器与MyBatis持久层框架，所以开发人员只需配置数据库就能立即进行开发调试。同时，为了更有效率的进行代码开发，使用了Maven项目管理工具来对繁多的jar包进行管理，Maven的使用大大缩减了项目的搭建耗时，也更加利于开发人员对项目jar包的使用。

在优化代码调试方面，使用了DevTools插件，该插件是一个热部署插件，当程序代码发生了修改时，该插件会更快速自动重启服务，编译被修改部分的代码，这样就无需开发人员进行手动重启服务器，更加省时省力。

系统总体开发环境清单如下表4-1所示：

表 4-1 系统开发环境清单

|  |  |  |
| --- | --- | --- |
| 硬件环境 | 处理器 | Intel(R) Core(TM) i7-8750M CPU @ 2.60GHz  2.60 GHZ |
| 内存 | 16.00GB |
| 硬盘 | 1TB |
| 屏幕分辨率 | 1920x1080 |
| 软件环境 | 操作系统 | Windows 10 专业版 |
| 浏览器 | Chrome ,QQ浏览器 |
| 开发工具 | IntelliJ IDEA 2018 |
| 热部署插件 | DevTools |
| 项目管理工具 | Maven 3 |
| 服务器 | Tomcat |
| 数据库 | MySQL8 |
| 数据库可视化工具 | Navicat |
| JDK版本 | 1.8 |

## 4.2 个人信息管理模块的实现

个人中心会显示用户的一些基本信息，比如展示一些用户玩过的游戏，已经发布的评价和评论。

在用户登录过后，进入个人中心，点击编辑资料进入信息修改页面，信息修改包含了用户的一些基本信息，如头像，密码，性别等等。玩家可以在此页面上传图片，图片格式不作限制，大小不得超过100M，不作裁剪，系统会自动以圆形图片显示图片。

（1）头像上传的实现

头像上传的核心就是实现文件上传功能。前端使用layui的文件上传接口，后台以MultipartFile作为文件的数据结构进行接收，多文件的上传需要多次调用后台接口。后台成功接收到文件后，会修改原始文件的名字，以上传的日期的时间戳作为该文件的名字，存放入以用户uid命名的文件夹中。未登录时无法使用上传文件的功能，文件上传核心代码与注释如下所示。

public static List<String> Upload(List<MultipartFile> files, String path, Boolean isMultip) throws IOException {

List<String> fileList = new ArrayList<>();

for(MultipartFile file : files){

//获取文件原始名字

String name = file.getOriginalFilename() ;

String suffixName = name.substring(name.lastIndexOf(".")) ;

Date date = new Date();

//自定义名字，用时间做名称

String hash = String.valueOf(date.getTime());

String fileName = hash + suffixName ;

File tempFile = new File(path + fileName);

if(!tempFile.getParentFile().exists()) {

tempFile.getParentFile().mkdirs() ;

}

//存放文件

tempFile.createNewFile() ;

file.transferTo(tempFile);

fileList.add(tempFile.getName());

if (!isMultip)

break;

}

return fileList;

}

（2）个人信息管理的实现

用户个人的信息在前台使用form表单来展示，在对个人信息进行修改时，layui-form组件会对表单信息进行验证，如非空的验证，手机数据格式和邮箱格式的验证。表单信息在经过序列化后，以json数据格式通过ajax异步提交，后台Spring Boot的@RequestMapping注解会把接收到的json数据自动封装成一个User对象，该User对象包含了当前用户所需修改的字段及数据。调用Service层的用户信息更新接口即可完成信息更新。个人信息管理时序图如图4-1所示。



图4-1个人信息管理时序图

个人信息管理页如图4-2所示。

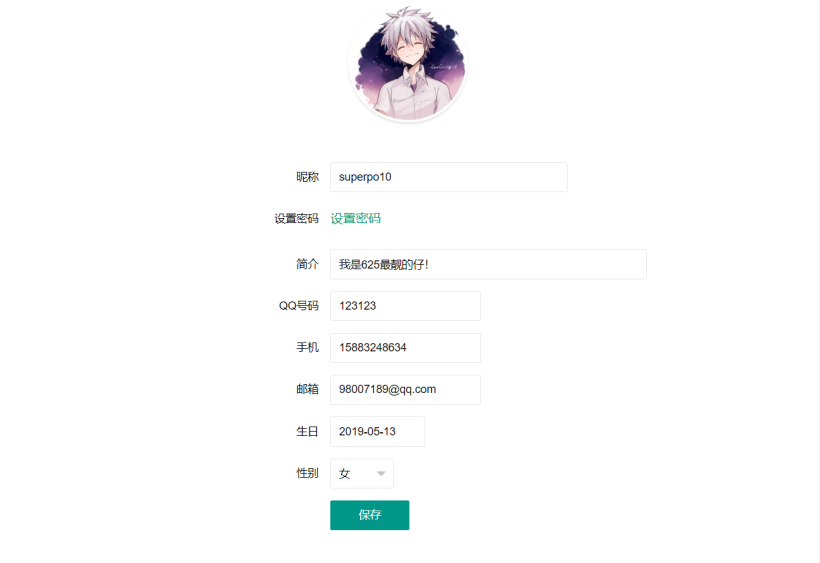


图4-2 个人信息管理

## 4.3 游戏信息管理模块的实现

游戏信息管理由游戏管理员和网站管理员共同维护，不过，游戏的信息大部分都由游戏管理员负责维护，一个游戏可以拥有多个游戏管理员。通常，一个游戏所包含的信息十分繁多，包括但不限于游戏的名称，版本，大小，厂商信息，发布日期等等。

在功能实现方面，前端使用layui-form组件进行数据校验，图片的上传同样使用文件上传接口。后台在接收到前端传递的json数据后，将会自动封装成Game对象，调用GameService的游戏信息更新接口即可完成添加或更新。游戏信息管理时序图如图4-3所示。



图4-3 游戏信息管理时序图

游戏信息更新核心代码与注释如下所示。

public JsonResponse UpdateGameInfo(Game game, HttpSession session){

//游戏uid是否存在

if (game.getUid() == null)

return new JsonResponse(1, Lang.get("Tips\_NullUid"));

//游戏是否存在

if (gameService.GameIsEmpty(game.getUid()))

return new JsonResponse(1,Lang.get("Tips\_Game\_SearchError\_Null"));

//进行更新操作

if (gameService.UpdateGameInfo(game))

return new JsonResponse(0, Lang.get("Tips\_Update\_Success"), game);

return new JsonResponse(1, Lang.get("Tips\_Update\_Error"));}

游戏信息页面如图4-4所示。

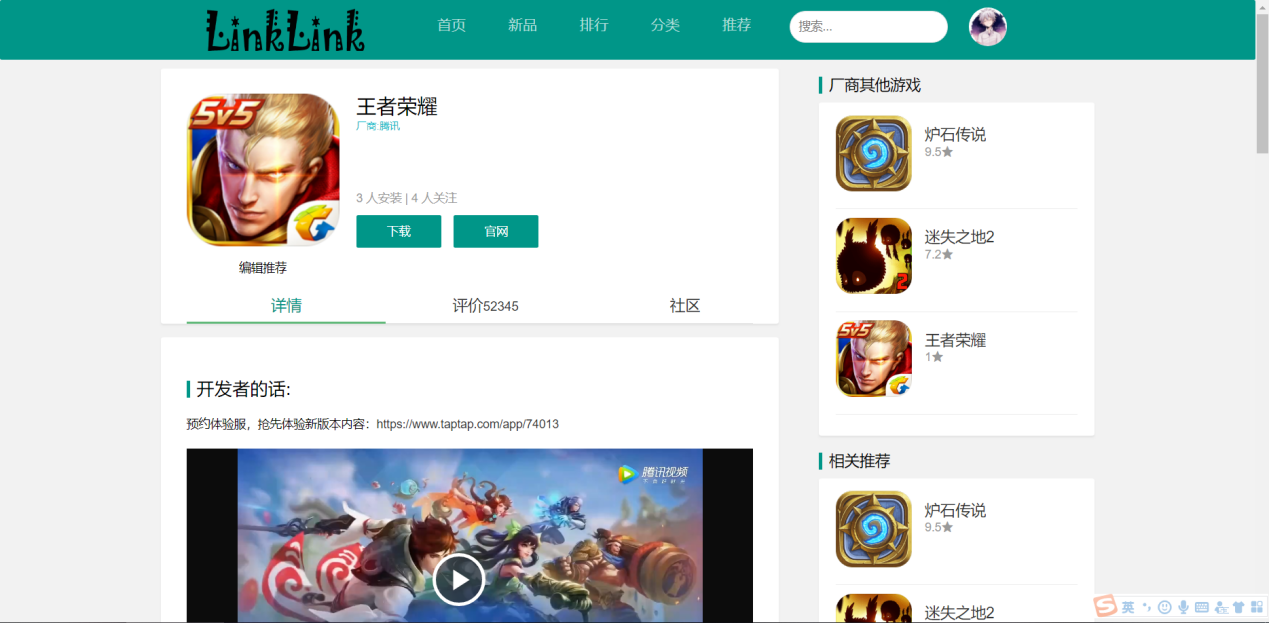


图4-4 游戏信息页面

## 4.4 游戏评价管理模块的实现

对于普通用户来说，可以在个人中心的评价页面查看已经发布的评价。点击评价可以查看评价并对评价进行修改或者删除操作。

游戏评价涉及到前端评分功能的实现，为了美观及便捷，前端的评分使用了layui-rate评分组件，用户可以通过该组件选择相应的分数。在后端以evaluate对象接收到评分以及评价的内容数据后，首先会对该用户的评价进行重复性检查，避免出现一个游戏有多个评价的情况，以确保游戏评价的唯一性。游戏评价管理时序图如图4-5所示。



图4-5 游戏评价管理时序图

评价更新核心代码与注释如下所示。

public JsonResponse UpdateEvaluate(Evaluate evaluate){

//判断评价是否存在

if (evaluateService.UpdateEvaluate(evaluate))

return new JsonResponse(0, Lang.get("Tips\_Update\_Success"), evaluate);

//返回结果

return new JsonResponse(1, Lang.get("Tips\_Update\_Error"));

}

评价页面如图4-6 所示

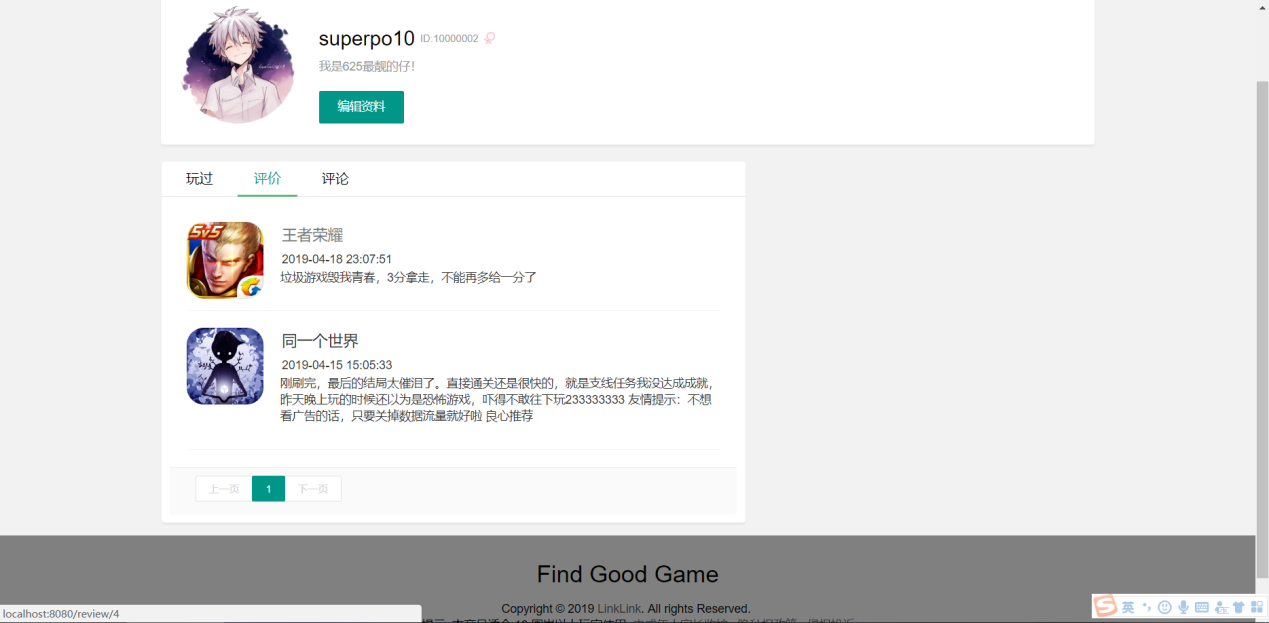


图4-6 评价页面

## 4.4 文章管理模块的实现

在前端，使用了LayUI自带的富文本编辑器作为文章的编辑工具，用户可以使用该编辑器编辑文章内容，插入图片等，插入的图片会通过相对应的的上传接口存放入本地磁盘中。文章的数据也使用json数据格式，通过ajax异步提交到后台。文章管理模块时序图如图4-7所示。

图4-7 文章管理模块时序图

文章发布核心代码与注释如代码下所示。

public JsonResponse AddTopic(Topic topic, HttpSession session){

//添加文章

if (topicService.AddTopic(topic))

return new JsonResponse(0, Lang.get("Tips\_Add\_Success"), topic);

//返回结果

return new JsonResponse(0, Lang.get("Tips\_Add\_Error"));

}

## 4.5 评论管理模块的实现

用户可在个人中心查看已发布的评论。查看页面会显示玩家最近的评论，

玩家可以点击该评论进入文章评论区查看评论，在评论展示页面可以对评论进行删除操作，前端数据同样以json数据通过ajax异步提交到后台，

评论管理时序图如图4-8所示。

图4-8 评论管理时序图

评论发布核心代码与注释如下所示。

public JsonResponse AddComment(Comment comment, HttpServletRequest request){

//添加评论

if(commentService.AddComment(comment))

return new JsonResponse(0, Lang.get("Tips\_Add\_Success"), comment);

return new JsonResponse(1, Lang.get("Tips\_Add\_Error"));

}

评论展示页面如图4-9所示

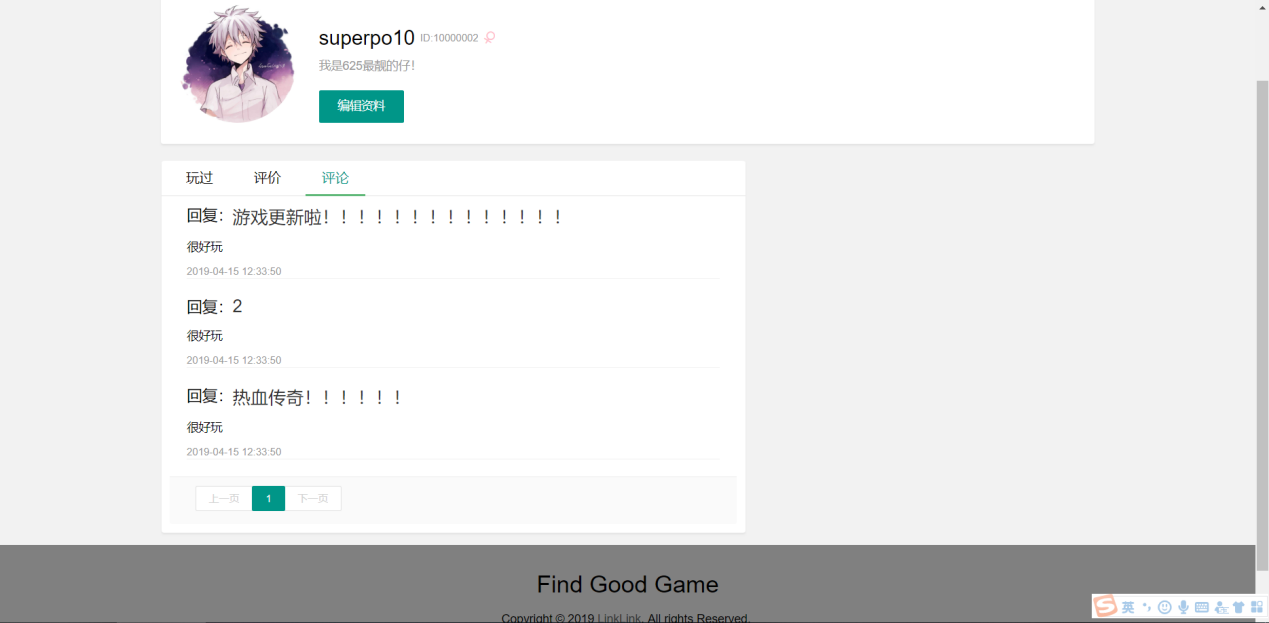


图4-9 评论展示页面

# 5 系统测试

系统开发完成之后，需要对各个功能模块进行相应的测试，从而找出功能的未知问题，以保证能够在系统上线运行前顺利解决问题，使得功能能够正常的使用。本次测试使用黑盒测试方法对各个功能模块进行功能验证。

## 5.1 测试环境

本系统的部署及测试环境如表5-1所示。

表5-1 测试环境

|  |  |  |
| --- | --- | --- |
| 硬件 | 处理器 | Intel(R) Core(TM) i7-8750M |
| 内存 | 16.00GB |
| 硬盘 | 1TB |
| 屏幕分辨率 | 1920x1080 |
| 软件 | 操作系统 | Windows 10 专业版 |
| 浏览器 | Chrome ,QQ浏览器 |
| 服务器 | Tomcat7.0 |
| 数据库 | MySQL8 |

## 5.2 测试用例及过程

以下列举游戏信息管理，评价管理，评论管理，文章管理四个模块的部分测试用例。

### 5.2.1 游戏信息管理模块测试

游戏信息管理测试内容如表5-2所示。

表5-2 游戏信息管理测试内容

|  |  |  |  |
| --- | --- | --- | --- |
| 测试用例编号 | Test001 | 测试用例名称 | 游戏信息管理模块 |
| 测试描述 | 对游戏信息发布或修改 | | |
| 前驱条件 | 账号已登录，且具有修改权限 | | |
| 测试步骤 | 1.选择图片上传；  2.点击上传；  3.输入游戏信息；  4.点击修改按钮； | | |
| 期待输出结果 | 1.当所选上传文件的格式有误或文件大小超过100m时会弹窗提示；  2.当有必填项没有填写时，点击修改按钮会弹窗提示‘xxx不能为空’；  3.发布时，若该游戏已存在会进行提示；  4.当所有数据输入正确，点击修改按钮，提示修改或发布成功； | | |
| 测试结果 | 通过 | | |

### 5.2.2 游戏评价管理模块测试

评价管理测试内容如表5-3所示。

表5-3发布游戏评价测试内容

|  |  |  |  |
| --- | --- | --- | --- |
| 测试用例编号 | Test002 | 测试用例名称 | 评价管理模块 |
| 测试描述 | 发布与修改游戏评价 | | |
| 前驱条件 | 账号已登录 | | |
| 测试步骤 | 1.进入想要评价的游戏详情页；  2.移动到评价区；  3.选择评分；  4.输入评价内容；  5.点击发布按钮； | | |
| 期待输出结果 | 1.未选择评分时点击发布按钮会提示‘尚未评分’；  2.评价内容没有填写时，点击发布按钮会弹窗提示‘内容不能为空’；  3.发布时，若该游戏已存在评价会进行提示‘评价已存在’；  4.当所有数据输入正确，点击发布按钮，提示发布成功； | | |
| 测试结果 | 通过 | | |

### 5.2.3 评论管理模块测试

评论管理测试内容如表5-4所示

表5-4 评论管理测试内容

|  |  |  |  |
| --- | --- | --- | --- |
| 测试用例编号 | Test003 | 测试用例名称 | 评论管理模块 |
| 测试描述 | 修改评论，删除评论 | | |
| 前驱条件 | 账号已登录 | | |
| 测试步骤 | 1.进入个人中心；  2.点击评论页签，进入评论展示页；  3.点击相应评论进入评论详情页；  4.点击修改按钮进入修改页；  5.输入修改内容；  6.点击发布按钮； | | |
| 期待输出结果 | 1.未登录无法进行操作；  2.评论内容没有填写时，点击发布按钮会弹窗提示‘内容不能为空’；  3.当所有数据输入正确，点击发布按钮，提示发布成功 | | |
| 测试结果 | 通过 | | |

### 5.2.4文章管理模块测试

文章发布测试内容如表5-5所示。

表5-5 文章发布测试内容

|  |  |  |  |
| --- | --- | --- | --- |
| 测试用例编号 | Test004 | 测试用例名称 | 文章管理模块 |
| 测试描述 | 发布文章 | | |
| 前驱条件 | 账号已登录 | | |
| 测试步骤 | 1.进入游戏社区；  2.点击发布文章按钮进入文章编辑页面；  3.输入文章内容；  4.点击发布按钮，若数据校验通过则提示发布成功 | | |
| 期待输出结果 | 1.未登录无法进行操作；  2.内容没有填写时，点击发布按钮会弹窗提示‘内容不能为空’；  3.当所有数据输入正确，点击发布按钮，提示发布成功； | | |
| 测试结果 | 通过 | | |

## 5.3 测试结果分析

经测试，大部分功能模块均可以正常使用，没有出现较严重的问题，一些小的问题比如Test04测试用例中出现使用LayUI富文本编辑器发布文章时，出现数据接收错误的Bug，排查后发现为后台接口接收请求格式错误；进行多图片上传时，出现图片返回路径错误的问题，经排查，发现是由ajax多次请求导致数据重复。大部分小问题已经得到解决，可能还存在未发现的问题，到目前为止，系统能够正常运行。

结 束 语

本系统旨在为所有热爱游戏的玩家提供一个更加高效发现优质游戏，分享游戏的平台，并且希望越来越多的玩家拥有一个体验游戏的良好环境。

本系统的整体完成度虽然没有计划中的高，不过基本的功能模块都已经实现。以TapTap网站为原型实现的前端页面简洁清晰，有着良好的视觉体验。缺点在于有些模块的子功能尚未实现，如对游戏的收藏，预约未上线游戏，用户关注以及点赞功能。

通过本次系统的设计与实现，提升了自身的编码能力与项目开发软件的使用熟练度，并对项目整体架构有了更进一步的认知，在编码各个功能模块时，更加注重各个功能模块之间的联系和耦合性，培养了对项目全局把握的意识。除此之外，通过此次的课题研究，更加了解到了需求分析与功能设计对前期构建项目的重要性。

参考文献

1. 刘晓. 网络游戏行为及其影响因素研究[D].武汉大学,2018.
2. 乔伊娜. 游戏销售与交流平台设计与实现[D].黑龙江大学,2017.
3. 梁金财. 高性能、高灵活度的游戏发行平台设计与实现[D].华南理工大学,2016.
4. 牛舜君,高志军.大学生手机游戏现状调查与启示[J].中国教育技术装备,2018(23):14-16.
5. 魏玮. 互联网产品游戏化设计策略研究[D].江南大学,2018.
6. 周世明,徐昕仪.浅析手机游戏行业的现状与发展趋势[J].计算机产品与流通,2019(04):86.
7. 许亮. 梁晔，马楠 游戏资讯和交流平台的设计与实现 科技广场 2017 第6期
8. 补剑琪. 基于SSM框架游戏玩家交流平台设计与实现[D].北京交通大学,2017.
9. 李畅. 基于iOS系统的游戏社区设计与实现[D].哈尔滨工业大学,2017.
10. 唐玥玚. 基于Spring与iBatis的游戏活动商城平台设计与实现[D].北京交通大学,2016.
11. 李天明. 豆玩手机游戏平台的设计与实现[D].吉林大学,2015.
12. 付星. 网络游戏官方网站的设计与实现[D].电子科技大学,2012.

致　　谢

在论文的选题与撰写过程中，本人得到了指导老师尽心尽力的指导和帮助。导师认真负责，知识技能丰富，帮我解决了很多在撰写论文过程中出现的问题和遇到的困难，并且及时指出并纠正了我的错误。

在此，我向指导老师表示由衷的感谢！

其次，感谢大学时期所有教导过我的所有老师！感谢他们的细心教导与尽职尽责，让我在这四年的大学生活中能够学有所成。

最后向在百忙之中评审本文的各位专家、老师表示衷心的感谢！

**作者简介：**

|  |  |  |  |
| --- | --- | --- | --- |
| 姓 名： | 宋钦 | 性别： | 男 |
| 出生年月： | 1996年5月17日 | 民族： | 汉 |
| E-mail： | 980047129@qq.com |  |  |

声　　明

本论文的工作是2018年10月至2019年5月在成都信息工程大学软件工程学院完成的。文中除了特别加以标注地方外，不包含他人已经发表或撰写过的研究成果，也不包含为获得成都信息工程大学或其他教学机构的学位或证书而使用过的材料。

关于学位论文使用权和研究成果知识产权的说明：

本人完全了解成都信息工程大学有关保管使用学位论文的规定，其中包括：

（1）学校有权保管并向有关部门递交学位论文的原件与复印件。

（2）学校可以采用影印、缩印或其他复制方式保存学位论文。

（3）学校可以学术交流为目的复制、赠送和交换学位论文。

（4）学校可允许学位论文被查阅或借阅。

（5）学校可以公布学位论文的全部或部分内容（保密学位论文在解密后遵守此规定）。

除非另有科研合同和其他法律文书的制约，本论文的科研成果属于成都信息工程大学。

特此声明！

作者签名：

2019年 5 月 31日