



华南农业大学

本科毕业论文

基于 WEB 的华农趣事平台的设计与实现

陈小红
201330350106

指导教师 杜治国 副教授

学 院 名 称 数学与信息学院
论文提交日期 2017 年 4 月 23 日

专 业 名 称 网络工程
论文答辩日期 2017 年 5 月 13 日

摘 要

我们正在进入“互联网+”时代，互联网与我们的生活密不可分，而随着互联网的进步与发展，人们开始在互联网上分享信息，分享学习，分享自己的心情。

基于丰富华农师生校园生活和促进交流沟通以及便于华农师生分享趣事的需求出发，本文设计了一个基于 Web 的华农趣事平台。系统使用了基于 Yaf(Yet Another Framework) 框架的 PHP(Hypertext Preprocessor) 作为后台开发语言和 MVC(Model View Controller) 架构模式，前端使用 MVVM(Model-View-ViewModel) 框架语言 Vue.js 实现完全的前后端分离。以 Apache 作为 Web 服务器，选择采用 MySQL 数据库对数据进行存储，为了保证用户信息安全采用了 MD5(Message-Digest Algorithm 5) 安全加密技术。设计了普通用户使用的趣事平台和供管理员用户对普通用户使用的趣事平台进行管理和规范的后台管理系统。趣事分享平台主要分为趣事展示、趣事操作、用户空间和信息设置四个主要功能模块，而后台管理系统主要分为趣事管理、评论管理、用户管理和管理员管理四个功能模块。

普通用户可以注册登录平台，可以发表、评论、收藏、点赞趣事以及回复他人的评论，也可以进入自己的空间查看自己发表、收藏的趣事以及他人对自己评论的回复内容，同时也可以修改自己的个人信息。管理员可以在管理系统创建管理员，对普通用户以及普通用户发表的趣事和评论进行管理，另外也可以根据用户名、趣事内容和评论内容对用户、趣事和评论进行搜索。平台除了按要求完成之外还增加了功能优化了系统，但是仍然有可以优化提升的问题和地方存在，有待进一步完善和优化。

关键词：信息加密 趣事平台 Yaf 框架 后台管理系统

Design and Implementation of the Web-based Fun Platform of SCAU

Chen Xiaohong

(College of Mathematics and Informatics, South China Agricultural University, Guangzhou
510642, China)

Abstract: We has arrived the "Internet plus" era, so the Internet is inseparable from our life, and with the progress and development of the Internet, people have begun to share information, share learning, and share their feelings on the Internet.

In order to enrich the teachers and students' campus life in South China Agricultural University and promote the communication and share fun, this paper designs a fun platform of SCAU. The system uses the YAF framework which based PHP (Hypertext Preprocessor) as the background development language and MVC (Model View Controller) schema model and use vue.js that is a front-end MVVM (Model-View-View Model) Framework language to achieve separation of front and rear ends and choose to use apache as a web server. In order to ensure the security of user information, it chooses to use the MySQL database to store data, and uses MD5 (Message-Digest Algorithm 5) Security Encryption Technology. The common users are designed to use the Fun Platform and the administrators can manage it. The open platform is divided into four main function modules, which are the interesting display, the fun operation, the user space and the information set up. The management system is also divided into four functional modules, which are fun management, comments management, users management and administrators management.

Ordinary users can register, login and publish, comment, collect, like and reply to other people's comments. They can enter their own space to view their own publish, collections of anecdotes and others comments on their replies to the content, but also can modify their personal information. Administrators can create administrators, manage the fun and comments made by ordinary users, as well as manage ordinary users in the management system. They can also search for users, anecdotes, and comments based on user names, funny content, and comments. In addition to the completion of the platform as required to increase the function of optimizing the system, but there are still some problems that need to be further improved.

Key words: information encryption fun platform Yaf framework background management system

目 录

1 前言	1
1.1 研究背景	1
1.2 内容架构	1
1.2.1 研究内容	1
1.2.2 论文结构	1
2 开发环境和相关技术简介	2
2.1 开发环境	2
2.2 相关技术简介	2
2.2.1 Yaf 框架	2
2.2.2 Vue.js 界面库	2
2.2.3 MVC 架构模式	3
2.2.4 MVVM 架构模式	4
2.2.5 localStorage 本地存储	5
2.2.6 MD5 算法	5
3 需求分析	6
3.1 概述	6
3.2 可行性分析	6
3.3 系统功能性需求	6
3.3.1 游客功能	6
3.3.2 普通用户功能	7
3.3.3 管理员功能	10
3.4 非功能性需求	11
4 概要设计	12
4.1 总体设计	12
4.2 设计思想	12
4.3 系统结构设计	13
4.4 系统功能设计	13
4.4.1 开放平台功能	13

4.4.2 后台系统功能	16
4.5 数据库设计	18
4.5.1 数据库 ER 图设计	18
4.5.2 数据库表设计	19
5 详细设计	24
5.1 存储设计	24
5.1.1 session	24
5.1.2 数据库连接配置	24
5.1.3 localStorage 前端存储	25
5.2 基本设计	25
5.2.1 ajax 发起异步 http 请求	26
5.2.2 后台控制器 controller 接口	27
5.2.3 后台模型 model	27
5.2.4 上传图片	28
5.3 游客功能设计	29
5.3.1 注册	29
5.3.2 查看趣事	29
5.3.3 游客查看评论	30
5.3.4 查看他人的用户空间	31
5.4 普通用户功能设计	31
5.4.1 登陆	31
5.4.2 发布趣事	31
5.4.3 普通用户查看趣事	32
5.4.4 点赞趣事	32
5.4.5 踩趣事	32
5.4.6 评论趣事	32
5.4.7 回复评论	32
5.4.8 收藏和取消收藏趣事	33
5.4.9 修改个人信息	33
5.5 管理员用户功能设计	34

5.5.1 管理员登陆 34

5.5.2 后台系统头部功能 34

5.5.3 用户管理 35

5.5.4 趣事管理 36

5.5.5 评论管理 37

5.5.5 系统管理 38

6 结论与展望 40

6.1 结论 40

6.2 展望 41

参考文献 42

致谢 43

1 前言

1.1 研究背景

随着互联网的日益普及，普通网民逐渐成为互联网内容的书写者和发布者，他们利用网络平台分享自己的观点、心情及各种信息资源（李莹莹，2012）。微信朋友圈、微博、人人网等社交网络平台都是为分享而生，生活中的许多快乐和欢笑都源自于互相分享，而根据 2016 年数据显示，华南农业大学现有全日制在校生 4.2 万余人，其中本科生 3.8 万余人，研究生 4000 余人，来自 27 个国家的留学生 89 人。这么一个庞大的团体需要一个公共的平台去分享自己的快乐和趣事，把自己的欢乐和开心分享给同校的师生，同时也可以增大自己的社交圈子，认识更多同校的同学，丰富自己的校园生活。

基于 WEB 的华农趣事平台的研发是为了丰富华农师生校园生活，传播欢声笑语，促进校园师生的沟通交流，于是它的重要性与必要性对于我们来说就不言而喻了。

1.2 内容架构

1.2.1 研究内容

主要包括两个方面：一是趣事分享平台，主要包括普通用户登陆平台可以发布、评论、点赞、踩趣事以及管理自己的用户信息等。二是平台的后台管理系统，为了过滤掉敏感的和一些没有分享价值的信息，用户发布的趣事需要平台管理员登陆后台管理系统进行审核通过后才可以在公开的趣事分享平台里展示，另外也可以对平台注册普通用户、管理用户发布的评论以及对系统管理员进行管理。

1.2.2 论文结构

本文阐述了基于 WEB 的华农趣事平台的具体实现流程。

第一章介绍了论文以及平台研究的背景和意义。第二章对平台的开发环境和所使用的相关技术作出了详细的介绍。第三章则是对平台进行详尽的需求分析以及列出了各个模块的功能需求。第四章则在需求分析的基础上对系统的实现进行概要设计并设计出合适的数据库表结构。第五章对系统进行了详细的设计，详细说明了每个功能模块的使用方法，另外也对各个模块进行代码优化设计并反复调试改善和修改。最后，第六章阐述了系统开发和论文撰写的心得总结。

2 开发环境和相关技术简介

2.1 开发环境

系统在 Microsoft Windows 10 操作系统环境下使用了基于 Yaf 框架的 PHP (Hypertext Preprocessor) 语言来进行后台开发, 前端则使用了最新的 HTML5 标准、CSS3 层叠样式以及 MVVM (Model-View-ViewModel) 框架语言 Vue.js 来构建页面以及与后台进行数据交互并实现了完全的前后端分离。以 Apache 作为 Web 服务器, 选择采用开源的、可移植性以及运行效率高、丰富的网络支持的 MySQL 数据库对数据进行存储。

2.2 相关技术简介

2.2.1 Yaf 框架

Yaf 全称 Yet Another Framework, 是一个 C 语言编写的 PHP 扩展形式提供的 PHP 开发框架。与一般的 PHP 框架相比, 它更快、更轻便, 它提供了 Bootstrap、路由、分发、视图、插件, 是一个全功能的 PHP 框架 (刘淇峰, 2015)。

相比其他 PHP 框架, Yaf 有很多优点, 如下: 用 C 语言开发的 PHP 框架, 相比原生的 PHP 几乎不会带来额外的性能开销; 所有的框架类, 不需要编辑, 在 PHP 启动的时候加载, 并常驻内存; 更短的内存周转周期, 提高内存利用率, 降低内存占用率; 灵巧的巧的自动加载, 支持全局和局部两种加载规则, 方便类库共享; 高性能的视图引擎; 高度灵活可扩展的框架, 支持自定义视图引擎, 支持插件, 支持自定义路由等等; 内建多种路由, 可以兼容目前常见的各种路由协议; 强大而又高度灵活的配置文件支持, 并支持缓存配置文件, 避免复杂的配置结构带来的性能损失; 在框架本身, 对危险的操作习惯做了禁止; 更快的执行速度, 更少的内存占用 (laruence, 2012)。选择使用 Yaf 框架来进行 PHP 开发的好处和必要就不言而喻了。

2.2.2 Vue.js 界面库

Vue.js 作为一个 Web 界面库, 用于构建数据驱动, 旨在利用尽可能简单的 API 去实现数据的响应式绑定和组件的組合使用, 其工作原理如图 1。

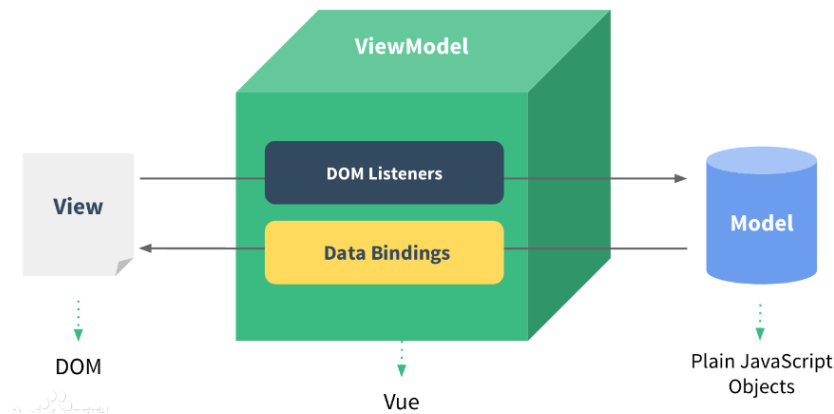


图 1 Vue.js 工作原理图（尤雨溪,2015）

Vue.js 的核心是一个响应的数据绑定系统，绑定被创建后，DOM（Document Object Model）与数据将会保持同步。每当修改了数据，DOM 便会相应地更新，并且数据修改与 DOM 更新实现分离。这使得代码撰写变得更简单，更容易理解。不同于传统页面使用超链接实现页面跳转、切换的方式，Vue 的单页面应用是通过路由设定访问路径，并将其与组件进行映射以实现路径切换，即组件切换（孙宁，赵维平，王宇飞等，2017）。

2.2.3 MVC 架构模式

MVC（Model View Controller 模型-视图-控制器）是 1970 年由 Trygve reenskaug 在在施乐公司的研究中心提出的，最早的参考例程是用 smalltalk80 编写的。它最初是用来解决应用软件中的用户图形界面的交互问题，现已成为 Sun 公司 J2EE 平台的设计模式，它为开发交互式应用系统提供了一个优秀的设计模式。

MVC 是一种软件设计模式，它强制性使应用程序的输入、处理和输出分开，分解成模型、视图、控制器 3 种部件，它们各自处理自己的任务，如图 2 所示。

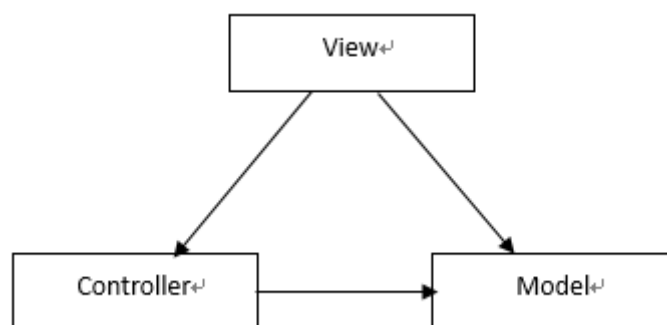


图 2 MVC 模式结构图

视图（View）是用户看到并与之交互的界面。模型（Model）用来表示企业数据和业务规则，数据的处理、逻辑和功能的计算放于其中。控制器（Controller）用来接受用户的操作并调用模型和视图完成用户的需求。

使用 MCV 模式开发 Web 应用的优点主要有：①加快项目的开发效率；②加快项目的开发效率；③利于软件工程化管理。不足主要有：①增加了系统的复杂性；②在提高大型项目开发效率的同时，降低了小型项目的发开效率；③数据处理流程复杂，从而在一定程度上稍微降低了系统的性能（何永太，2010）。

MVC 框架的出现，使系统各层任务明确、逻辑清晰，提高了代码重用性，并降低了后期的维护成本（易剑波，2016）。

2.2.4 MVVM 架构模式

MVVM 是在 MVC 模式上进一步发展的产物，它的全称是 Model View ViewModel，该架构最初是由微软 MartinFowler 提出的，它的关注点在能够支持事件驱动的 UI 开发平台，MVVM 模式的核心是对 View 和 ViewModel 进行双向数据绑定，当 ViewModel 的状态发生变化时，这种变化可以自动传递给视图 View。MVVM 模式架构图如图 3 所示：

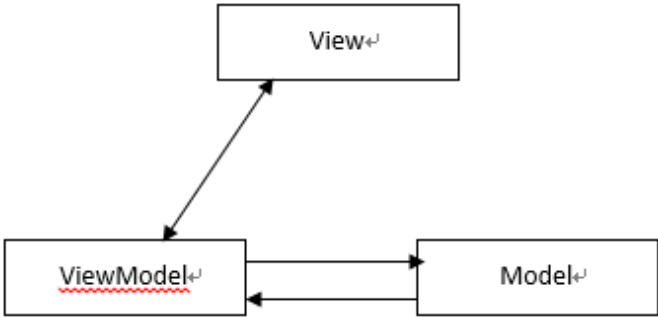


图 3 MVVM 模式架构图

如图 3 所示，其中 Model 是 MVVM 设计模式中用来存放数据和业务逻辑的类，它只是代表应用程序所需的数据信息，它不关注任何行为；View 是软件中与用户进行直接交互的部分，它需要响应 ViewModel 的事件并格式化数据，不负责控制应用的状态；ViewModel 用于封装业务逻辑层，用来存放显示逻辑和状态的类，为视图封装了展示逻辑，负责协调 View 与 Model 之间的数据传递，它可以选择直接把数据模型的变化传递给视图，或者把视图中数据的变化传递给数据模型，即在 Model 和 View 之间建立了双向绑定（易剑波，2016）。

2.2.5 localStorage 本地存储

Web 存储 API 为浏览器可以存储键/值提供了可实现的机制。其中 localStorage 就是 Web 存储 API 接口之一，而另一个就是 sessionStorage，这些接口会被浏览器原生支持，不用以来任何第三方插件。在本系统中采用 localStorage 在用户登录以及用户修改信息时把用户最新信息使用 localStorage 来存储在客户端。localStorage 拥有以下特性：（1）存放数据大小：大小一般被限制在 5MB 左右，但各个浏览器不一致。（2）仅在客户端（即浏览器）中保存，不参与和服务器的通信；（3）数据生命周期：生命周期永久，除非被用户主动清除，否则信息将永久被保存；（4）原生接口可以接受，亦可再次封装来对 Object 和 Array 有更好的支持。不同浏览器无法共享 localStorage 或 sessionStorage，相同浏览器的不同页面间可以共享相同的 localStorage（页面属于相同域名和端口），但是不同页面或标签页面无法共享 sessionStorage 的信息（黄华林，2015）。

2.2.6 MD5 算法

明文密码，指网站在存储用户密码与资料时采用的可以看得懂的明文字符。由于其安全性差、易受攻击，已经渐渐被 MD5 不可逆加密算法取代。

MD5 是 Message-Digest Algorithm 5（信息-摘要算法 5）的缩写，是让大容量信息在用数字签名软件签署私人密钥前被“压缩”成一种保密的格式（就是把一个任意长度的字节串变换成一定长的大整数）。MD5 可以为任何文件（不管其大小、格式、数量）产生一个同样独一无二的“数字指纹”，这个数字指纹就是存储在数据库中的暗纹密码。当用户登录一个使用 MD5 值，然后将此值与保存在文件系统中的 MD5 值进行比较，进而确定输入的密码是否正确（陈震，王艳，2013）。

MD5 加密方式采用的是单向加密的方式进行加密。单向加密方式对应的是双向加密方式，双向加密是加密算法中比较常用的，它除了可以将明文数据加密为密码以外，在需要的时候还可以使用一定的算法将加密以后的密文还原为明文数据。而单向加密方式正好相反，它只能对数据进行加密，而无法进行解密。MD5 加密方式就是这样一种加密方式，经过 MD5 加密后的明文数据，它得到的密文是永远不会变的。同时任意两端不同的明文数据经过 MD5 加密以后，其密文数据也一定是不同的（陈震，王艳，2013）。

3 需求分析

3.1 概述

本华农趣事平台是基于 WEB 供给华农师生分享趣事促进沟通交流的需求而设计的。平台用户主要分为三类，游客、普通用户和管理员。普通用户可通过本平台注册登录平台，可以随时发布趣事，对他人发布的趣事进行评论、收藏、点赞、踩等操作，可以回复他人对自己的评论进行沟通交流，修改个人信息以及重置密码，对自己收藏和发布的趣事进行管理。管理员登陆后台管理系统则可以对普通用户、用户发布的趣事和评论以及对后台管理系统的管理员进行管理，保障趣事平台的合理合法规范地正常运行。

3.2 可行性分析

(1) 系统必要性

华农师生在正常的上课、下课、学习校园生活之外还需要另外的娱乐平台去增添校园乐趣，同时庞大的华农师生群体也需要一个公共的平台去沟通交流，认识更多的校友，扩大自己的交际圈子，构建和谐校园，因此华农趣事平台的研发和存在是必要的。

(2) 技术可行性

本平台采用了 PHP 作为后台开发语言，使用 Yaf 作为后台开发框架同时结合前端 MVVM 框架 Vue.js 进行了以 MVC 为架构并且是目前最新流行的前后端分离开发方式，即后台只需要提供接口，而数据展示则完全交由前端来处理的简明规范化干净利落的应用程序的开发与部署，代码复用性和可维护性高，使用 MVC 架构来设计系统，即用一种业务逻辑、数据、界面显示分离的方法来组织和编写代码，将业务逻辑聚集到一个部件控制器里面，使得在改进和进行 UI 界面设计以及进行用户交互的同时，无需重新编写业务逻辑。同时平台选择采用体积小、速度快、成本较低、运行可靠的 Mysql 数据库，降低成本的同时也保证系统的稳定性和安全性。另外数据库还使用了 MD5 安全加密技术来处理用户信息，以此来保证用户的信息安全，提高系统的安全性。

3.3 系统功能性需求

本平台主要用户角色分为了三种，分别为游客、普通用户和管理员，系统功能性需求主要从这三种角色分别来阐述系统的主要功能性需求。

3.3.1 游客功能

本平台是一个开放的平台，所有人都可以访问，没有登陆的用户以及没有注册过本平台的用户都统称为游客，其具有的功能如下图 4 所示：

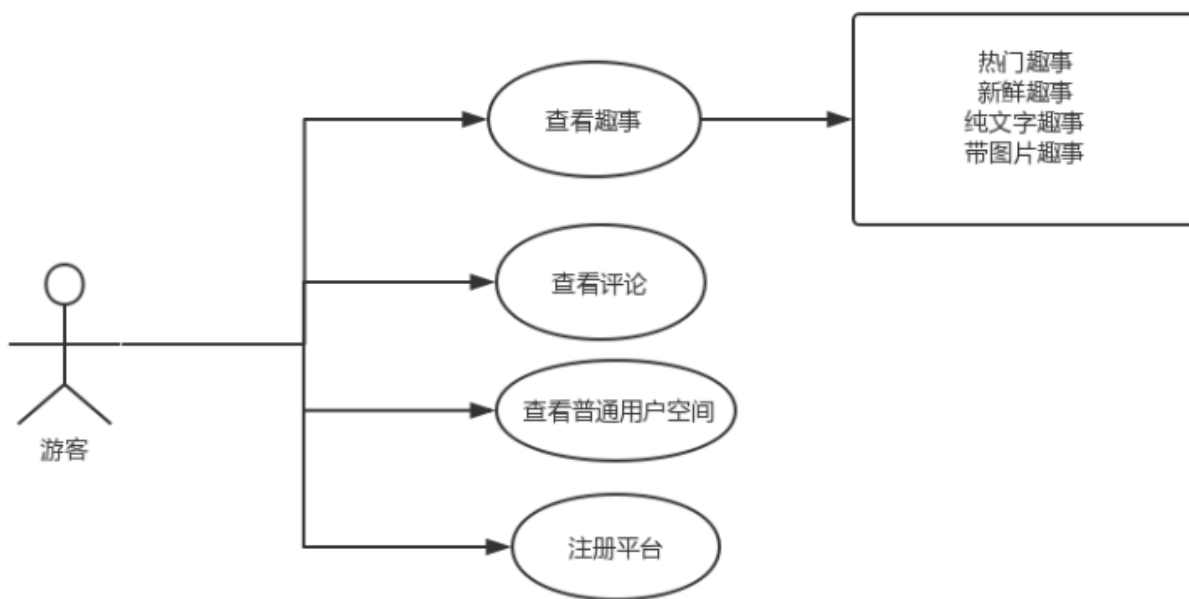


图 4 游客用户用况图

用例说明：

（1）查看趣事

功能描述：所有的用户，在没有登陆时都可以统称为游客，游客不用登陆即可使用本平台。没有登陆时可以在本平台首页上查看趣事，包括查看热门、新鲜、纯文字以及含有图片的趣事。但是游客无法点赞、踩和发布趣事。

（2）查看评论

功能描述：游客可以点击主页的趣事进入趣事详情查看趣事下的其他用户对该趣事的评论，以及。但是游客无法评论趣事。

（3）查看用户空间

功能描述：游客可以点击用户头像或者用户名，进入用户空间查看用户发表、收藏的趣事以及用户发出去的评论。

（4）注册平台

功能描述：游客可以点击首页头部右上角的“登陆/注册”按钮，注册本平台。游客在注册用户时只需要输入没有被使用过的用户名、密码和重复输入密码，即可以使用普通用户功能。

3.3.2 普通用户功能

普通用户作为本平台最基本、最主要也是最重要的平台使用对象人员，整个平台系统都是为了普通用户服务，能使用所有非系统功能。例如，普通用户成功注册平台后，可

以登录平台来发布趣事、评论趣事、点赞趣事、踩趣事、收藏趣事、回复评论、查看自己或者他人的用户空间，修改个人信息，找回密码等功能，具体功能详见下图 5。

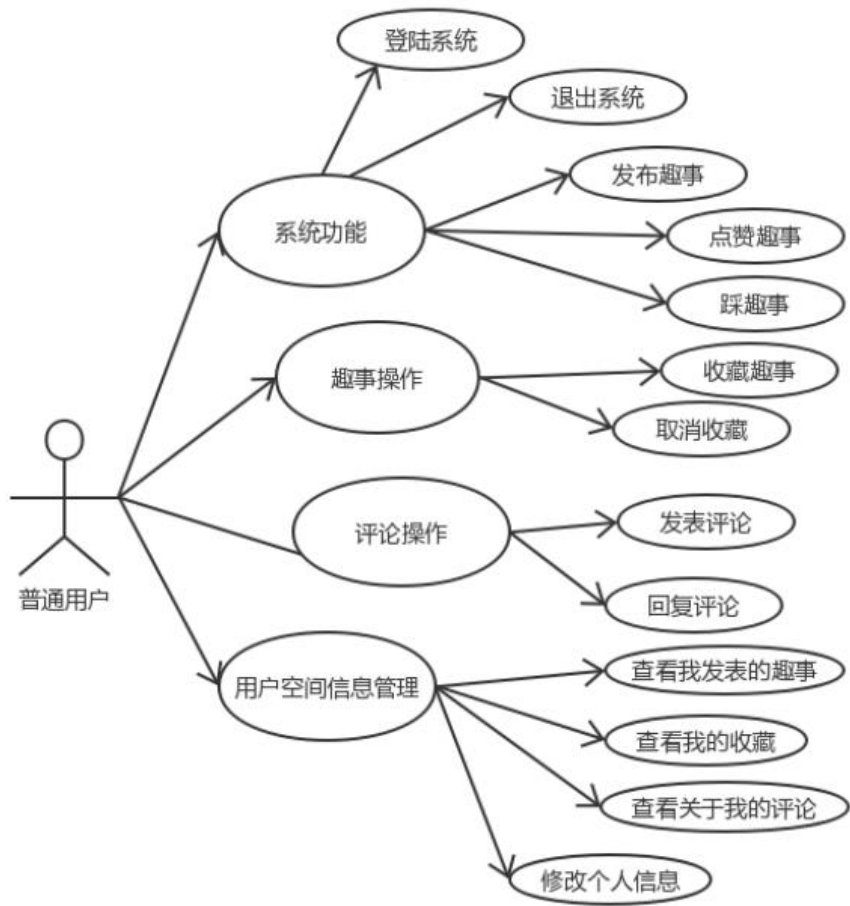


图 5 普通用户用况图

用例说明：

(1) 登陆平台

功能描述：用户点击平台首页头部的“登陆/注册”按钮，平台会弹出登录框供用户输入自己的用户名和密码。后台代码回验证用户的用户名和密码是否正确，校验成功用户就可以成功登陆平台。登陆成功，前端会把用户 id，用户名，用户头像信息存储到 localStorage 中，方便其他地方需用用到这些信息时不需要重新去后台获取，另外后台代码也会把用户 id 存到 session 中并设置有效期，在每一次前端进行 http 请求时都会判断 session 是否过期，如果已经过期则在需要登陆后才可以用到的功能则无法使用，会提示用户先重新登陆，保障用户信息的安全。

（2）重置密码

功能描述：用户如果忘记了自己的用户密码，可以输入自己的用户名，并重新输入密码即可重新重置密码。

（3）发布趣事

功能描述：用户登录成功后可以发布趣事，可以只有文字，也可以配上文字，为了保证趣事质量，以及过滤掉垃圾和违规信息，发布后的趣事需要用户管理员审核通过后方可发布在平台上。

（4）查看用户空间

功能描述：用户登录成功后点击首页右上角自己的头像或者用户名进入自己的用户空间。在自己的用户空间，用户可以看到自己在平台的一些统计时间，如：收藏的趣事总数，收到的赞，注册平台的天数以及发布的趣事所收获到的评论数。同时也可以查看自己发布、收藏的所有趣事以及他人评论自己的趣事和回复自己的评论。

（5）点赞趣事

功能描述：用户登录成功后在查看趣事的过程中，如果觉得某些趣事非常的有趣或者有意义有价值可以对该趣事进行点赞操作，并且不能取消点赞操作。

（6）踩趣事

功能描述：用户在查看趣事时如果觉得该趣事不够有趣或者不认同该趣事观点，可以对该趣事进行踩操作，并且不能取消踩操作。

（7）收藏趣事

功能描述：用户在查看趣事时如果看到了自己喜欢的趣事，可以点击收藏图标收藏该趣事，收藏后的趣事会统一出现在用户空间中的收藏系列，方便用户再次浏览查看。

（8）评论趣事

功能描述：用户登录成功后可以点击趣事或者评论按钮图标进入趣事详情对趣事进行评论，评论会按照评论的时间先后顺序排列，显示在趣事详情的下方。

（9）回复评论

功能描述：用户可以在趣事详情的评论列表中点击想回复的评论对应的“回复”按钮，输入回复的内容，点击提交即可回复评论成功。

（10）取消收藏趣事

功能描述：用户在自己的用户空间-收藏中可以查看到自己所有收藏的趣事，如果觉得某些趣事已经没有继续留在自己的收藏列表中的必要，可以点击取消收藏按钮取消收

藏该趣事，该趣事就会消失在自己的收藏列表中。

（11）修改用户信息

功能描述：用户进入到自己的用户空间-设置即可以修改自己的头像、用户名、密码等信息。

（12）注销退出系统

功能描述：为了保障用户的信息安全，用户在不使用本平台后，保险做法是点击平台首页头部的退出按钮或者用户空间-设置-退出登录即可安全退出系统清除 session 和 localStorage 存储的信息。

3.3.3 管理员功能

管理员用户作为平台的管理者，能够登陆平台的后台管理系统，可以操作后台管理系统中的所有功能，其用况图如下图 6 所示。

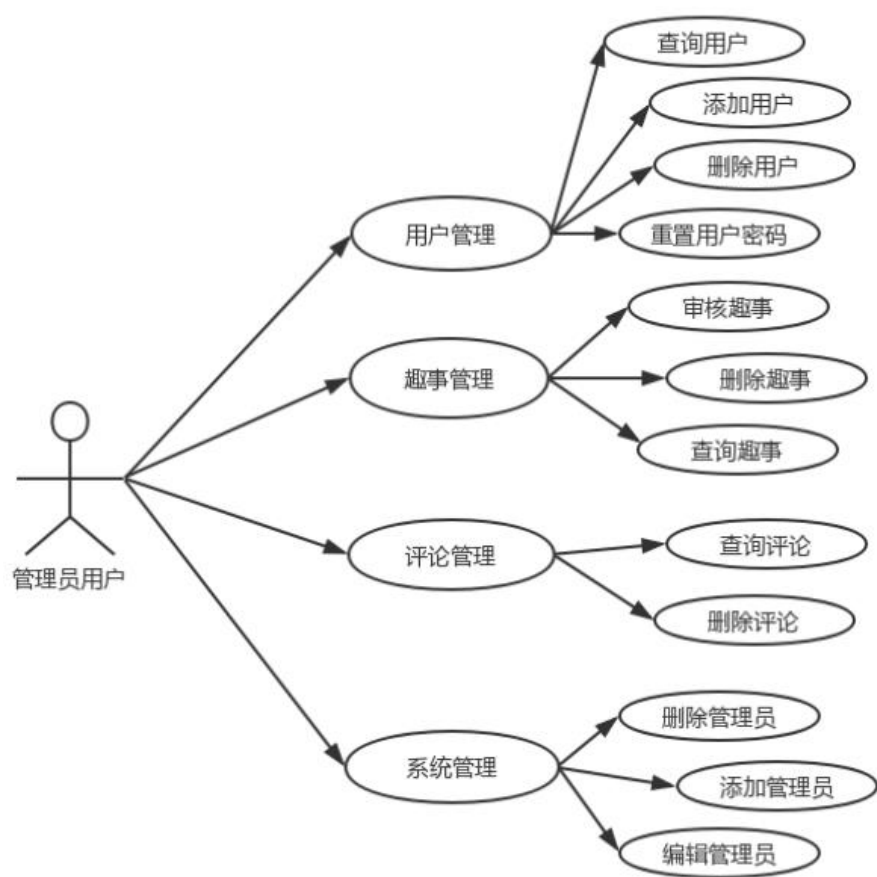


图 6 管理员用户用况图

用例说明：

（1） 用户管理

功能描述：管理员成功登陆后台管理系统后，可以查看所有的普通用户并可以根据用户名称查询用户，也可以为普通用户重置密码，同时还可以添加删除用户。

（2） 趣事管理

功能描述：管理员成功登陆后台管理系统后，可以查看普通用户发布的所有趣事并对普通用户发布的新鲜趣事进行审核，审核通过后的趣事才可以出现在公开平台，也可以删除趣事。

（3） 评论管理

功能描述：管理员成功登陆后台管理系统后，可以查看普通用户发表的所有评论并根据评论内容查询评论，也可以删除用户的评论。

（4） 系统管理

功能描述：管理员成功登陆后台管理系统后，可以对后台管理系统的管理员进行管理，如添加管理员，修改管理员的信息以及删除管理员。

（5） 修改自己的密码

功能描述：管理员成功登陆后台管理系统后，点击后台管理系统的管理员信息-修改密码，可以修改自己的密码。

3.4 非功能性需求

（1）界面友好简洁性：界面设计应美观大方清晰简洁，同时要保证提示信息充足，谨遵易用性的设计原则，让用户易于使用便于操作，界面要显示本用户的信息，以及系统的出入口。

（2）信息安全保障性：为了保证用户的信息安全，本平台对用户的密码等敏感信息进行了加密处理，敏感信息通过使用 MD5 加密技术处理过后才保存进数据库，这样哪怕服务器处于不安全的状态下，用户的重要敏感信息也不会泄露出去。

（3）数据交互即时性：用户在进行每一项操作时都应该能即时做出响应，避免使用户等待过久，如用户发布的评论能马上显示到评论列表。

（4）代码可维护性：平台选择采用 MVC 三层架构强制分离业务逻辑、数据、界面显示的方法来组织代码，让界面和控制器相分离，提高了系统的可维护性和可拓展性，使代码耦合性低，重用性高，生命周期成本低，便于维护和修改。

4 概要设计

4.1 总体设计

基于 Web 的华农趣事平台研发的目的是将为华南农业大学全体师生提供一个公共的平台去互相沟通交流，认识更多的校友，分享彼此的趣事和快乐。根据功能需求，主要讲平台划分为了四个功能模块，分别是求出趣事功能模块，评论功能模块，用户空间功能模块和平台管理模块。

4.2 设计思想

(1) 面向对象设计思想：本平台在设计时遵循着“一切皆是对象”的面向对象的设计思想。把软件系统堪称各种对象的集合，对象就是最小的子系统，一组相关的对象能够组合合成更加复杂的子系统。把所有复杂工程转化成了标准零部件。

(2) MVC 三层架构：本平台系统采用了 MVC 三层架构，Model 层采用 PHP 中的 Yaf 框架来实现处理，View 层采用 HTML 实现，开发人员自定义的 Action 类作为 Controller 层，采用 PHP 来实现。将业务逻辑、数据、界面显示相互分离，让业务逻辑聚集到一个部件控制器里面，使得在进行用户界面优化和改进、数据交互的同时，不需要重新编写业务逻辑。本系统采用面向对象的设计思想，将一切都看作对象，让数据和操作紧密的连结在一起，保护数据不被外界的函数干扰而意外地改变。其中，Action 类通过继承 \Core\BaseControllers 类大幅度地减少冗余的代码，通过数据封装减少外界的干扰。

(3) 组件化程序设计：平台选择使用 Vue.js 这个前端框架来实现前端组件化编程。Vue.js 允许开发者通过尽可能简单的 API 实现响应的数据绑定和组合的视图组件，组件可以扩展 HTML 元素，封装可重用的代码，提高代码可重用性。

(4) 数据库可维护：平台应该具备维护数据库的功能，及时根据用户需求进行数据的增加，删除，修改，即对用户，趣事，评论等数据的维护，从而降低开发成本，提高软件的适应性和可拓展性。

(5) 敏感信息加密存储：为了保障用户的信息安全，对于用户密码这等重要而敏感的信息，须采取 MD5 的加密方式，对信息进行密文处理后才可存储到数据库，这样假使服务器处于不安全的状态情况下，用户的重要信息也不至于泄露出去。

(6) 前后端分离：平台采用前后端分离的思想，使用 SPA (Single-page application) 单页应用模式来开发前端页面，各页面的跳转通过前端路由来进行，使得前后端的数据交互只需要通过接口，让应用干净利落，调试方便，从而提高开发的效率。

4.3 系统结构设计

基于 Web 的华农趣事平台的使用者有三种，游客、普通用户和管理员用户，注册、查看趣事、评论以及可以进入他人的用户空间为游客功能，系统被划分为两个主要的部分，分别是开放的趣事平台以及后台管理系统。开放的趣事平台可以分为查看趣事、趣事操作、评论操作、用户空间四个模块，后台管理系统可以划分为以下四个模块：用户管理，趣事管理，评论管理，系统管理。其中后台管理系统只有管理员用户才可以登陆使用。开放平台和后台管理系统是相对独立的，但使用的是共同的数据，通过数据交互，从而实现完整的平台功能，如下图 7 所示。

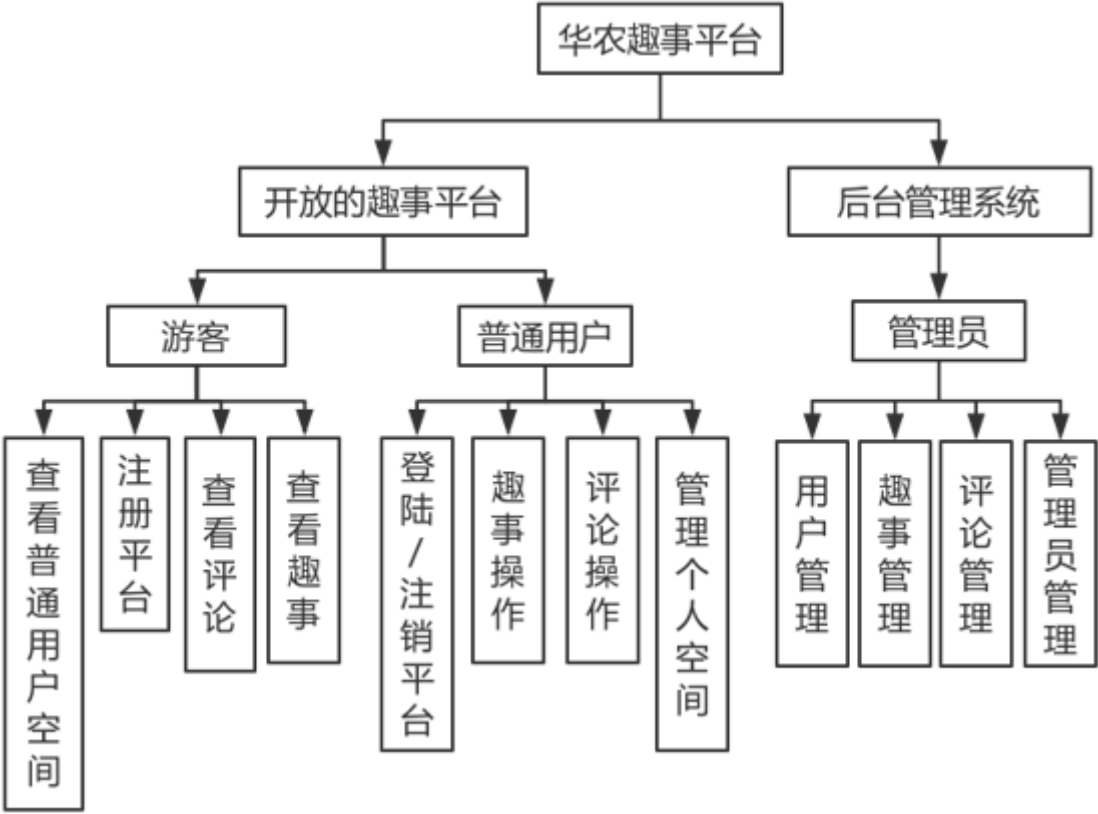


图 7 系统结构设计图

4.4 系统功能设计

4.4.1 开放平台功能

两个主要系统部分都有各自的模块和各自的功能，开放平台和后台管理系统之间相辅相成，根据先前的需求分析，可画出各部分的系统模块功能图。详细系统模块功能如下图 8 所示，

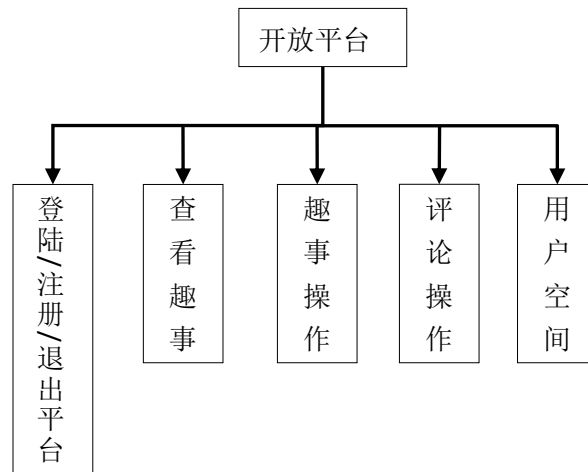


图 8 开放平台功能图

在查看趣事模块中，用户不需要注册也不需要登陆即可以进入平台首页查看平台中已存在的所有已经审核过的趣事列表，趣事分为了四种类型：热门、新鲜、文字、带图。用户可以点击平台首页头部的导航查看对应类型的趣事，每一种趣事都有进行分页显示，用户可以翻页浏览，如下图 9 所示。

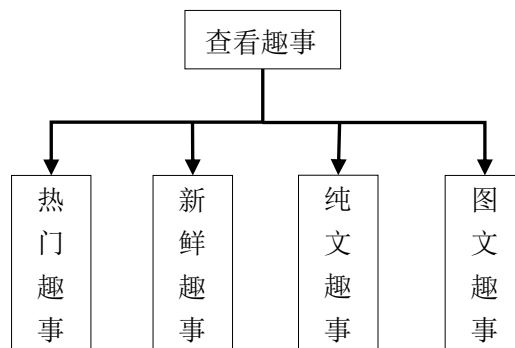


图 9 查看趣事模块功能图

成功登陆的普通用户可以使用趣事操作这一模块功能，普通用户成功登陆后可以对趣事进行点赞、踩和收藏以及取消收藏，如下图 10 所示。

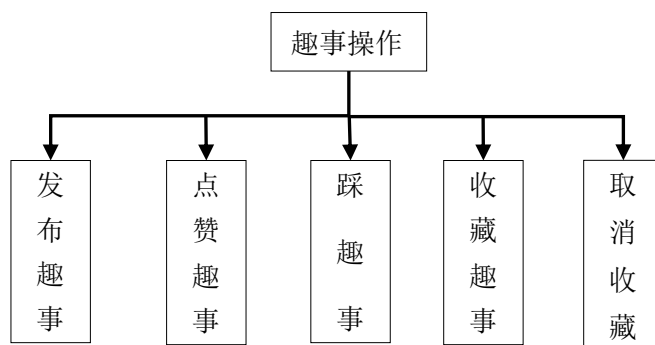


图 10 趣事操作模块功能图

在评论操作模块中，游客可以点击趣事列表的趣事内容或者每一条趣事的评论图标进入到趣事详情页面查看该趣事下的所有评论。如果用户需要评论趣事需要先登录成功后，方可以对趣事进行评论；用户登录成功后，可以在趣事详情页面或者进入自己的用户空间中的评论模块对感兴趣的评论进行回复，如下图 11 所示。

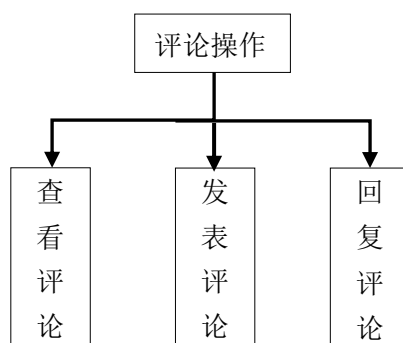


图 11 评论管理模块功能图

成功登陆的普通用户可以点击首页头部自己的头像或者用户名进入自己的用户空间，在用户空间，可以查看到自己在趣事平台的概况，发表的趣事获得的收藏数、获得的赞数、评论数以及在平台注册的时间，可以查看自己发表过的所有趣事，收藏的所有趣事，以及有关于我的评论，具体如下图 12 所示。

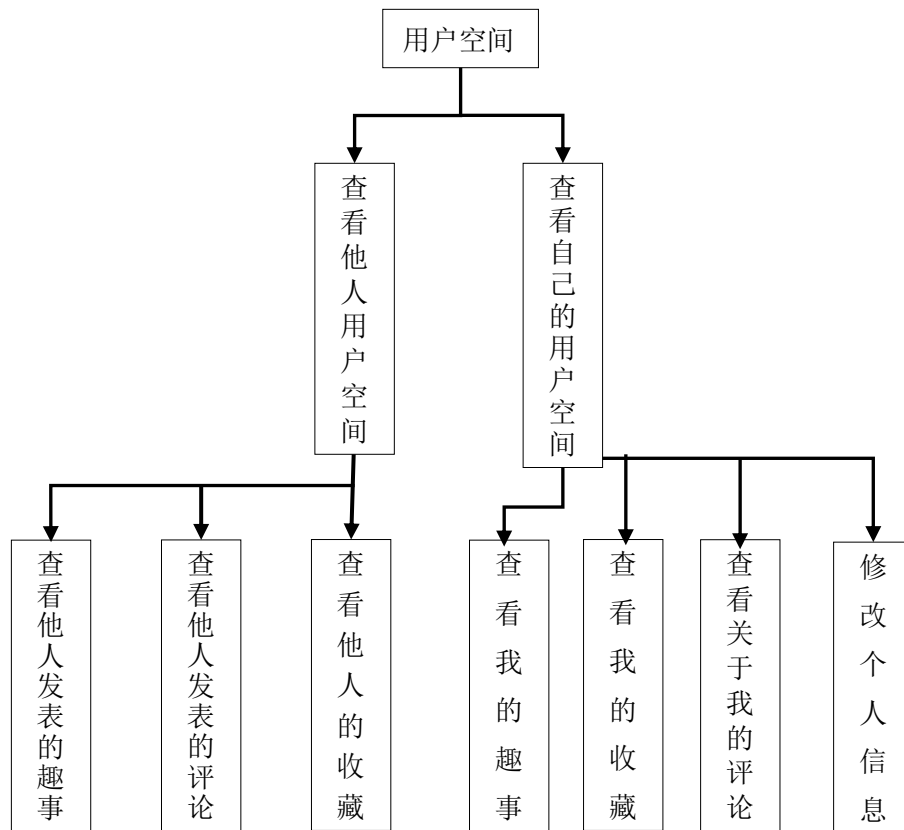


图 12 用户空间模块功能图

4.4.2 后台系统功能

为了保障趣事平台合法规范正常地运行，特设计了专门的后台管理系统来管理趣事平台。管理员用户可以登陆后台管理系统对趣事平台的用户、评论、趣事以及系统管理员进行管理，后台管理系统的功能结构设计图如图 13。

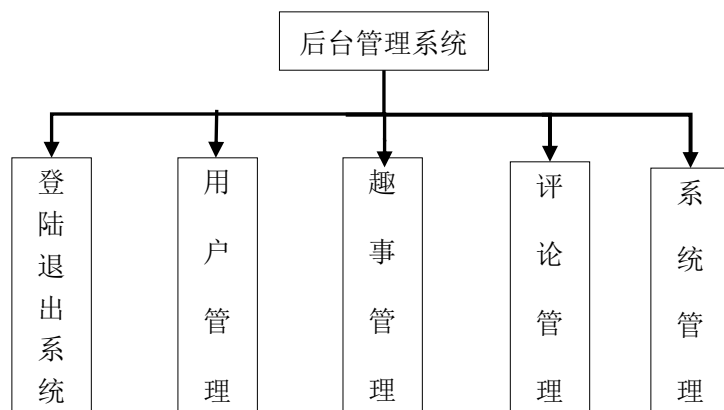


图 13 后台管理系统功能结构设计图

成功登陆后台管理系统的管理员用户对在公开平台注册了的用户进行管理，可以进行添加用户、根据用户名称查询用户、重置用户密码、删除用户功能操作，如下图 14 所示。

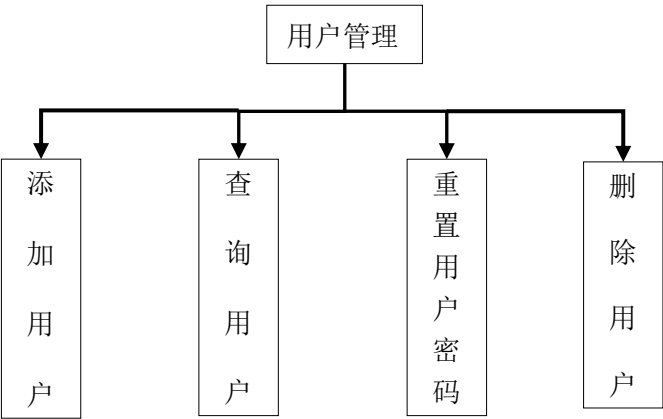


图 14 管理员用户管理模块功能图

成功登陆后台管理系统的管理员可以对普通用户在开放平台发表的趣事进行管理，普通用户发表的趣事必须要经过管理员在后台管理系统进行审核过后才可以显示在开放的趣事平台的趣事列表中；管理员用户还可以删除趣事，根据趣事内容查询趣事，如下图 15 所示。

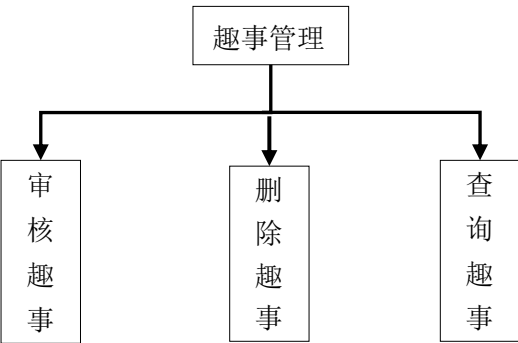


图 15 管理员趣事管理模块功能图

成功登陆后台管理系统的管理员可以对普通用户在开放平台发表的评论进行管理，管理员用户可以查看、删除普通用户发表的评论，还根据评论内容查询评论，如下图 16 所示。

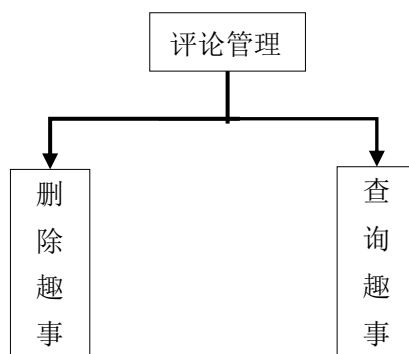


图 16 管理员评论管理模块功能图

成功登陆的普通用户可以点击首页头部自己的头像或者用户名进入自己的用户空间，在用户空间，可以查看到自己在趣事平台的概况，发表的趣事获得的收藏数、获得的赞个数、评论数以及在平台注册的时间，可以查看自己发表过的所有趣事，收藏的所有趣事，以及有关于我的评论，如下图 17 所示。

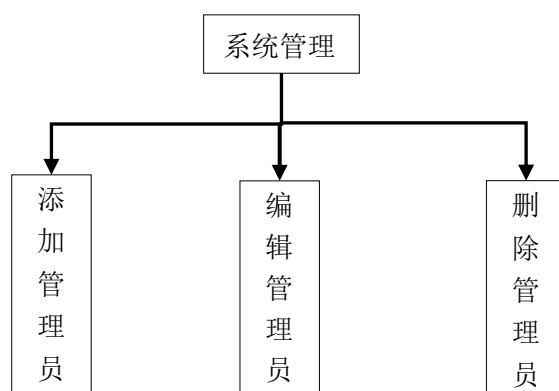


图 17 管理员系统管理模块功能图

4.5 数据库设计

4.5.1 数据库 ER 图设计

如下图 18 数据库 E-R 图所示，在本平台应用中一共有 9 张数据库表，每张表之间的关系以及每张表中所含的字段都可以从数据库 E-R 图中看出。

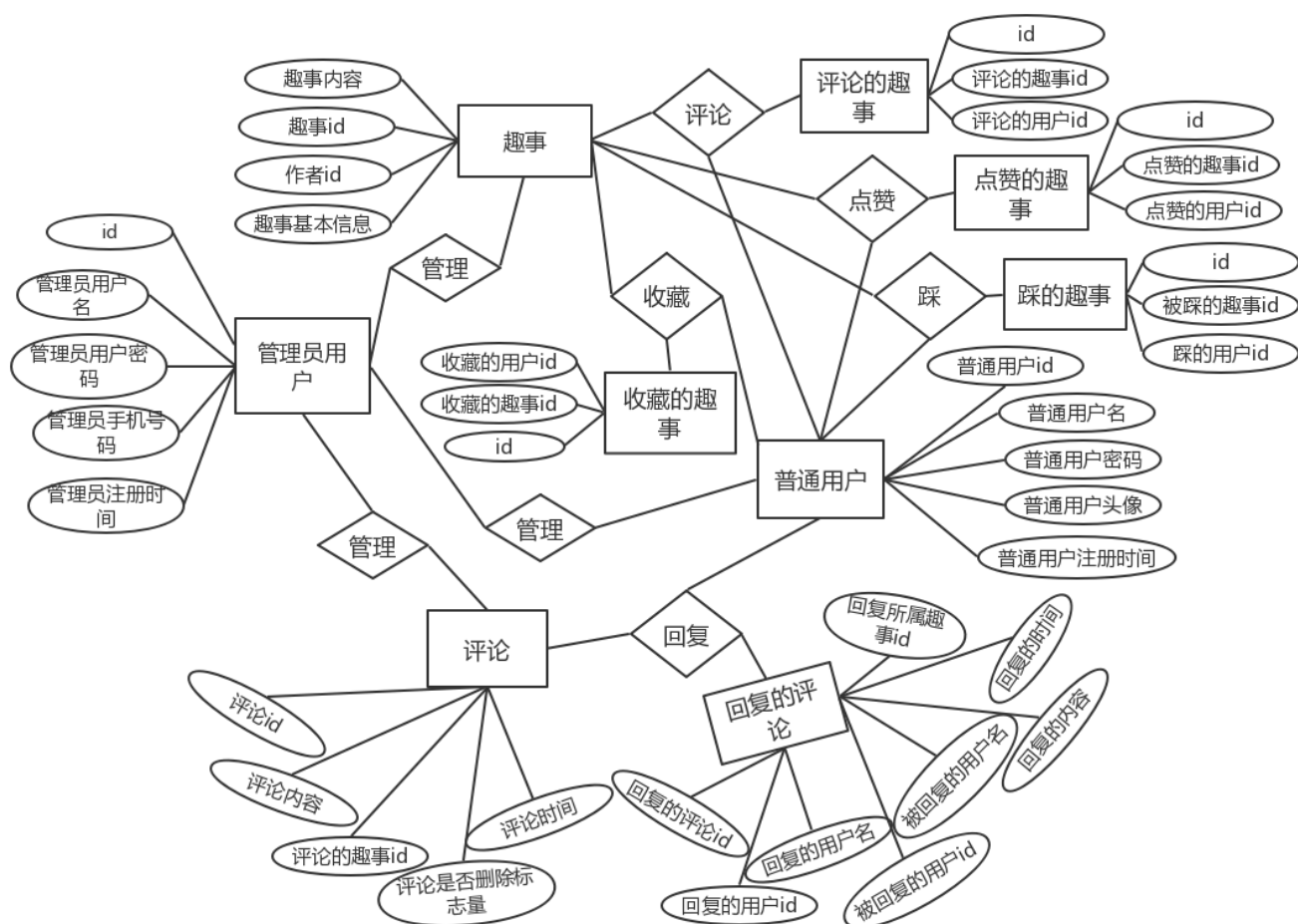


图 18 数据库 E-R 图

4.5.2 数据库表设计

(1) 普通用户 user 表

用户 User 表是用来存放所有普通用户的基本信息，用户注册、登陆、重置密码、修改个人信息，管理员删除用户、添加用户、为用户重置密码均需要对用户 user 表进行增删查改。user_id 为该表的主键，随着用户注册而自动增加；is_delete 标志着该用户是否被删除，0 为用户没有被执行删除操作，1 表示用户已被删除，如下表 1 所示。

表1 用户个人信息表

字段名	数据类型	长度	约束	描述
user_id	bigint	10	Primary Key	用户ID
user_name	varchar	50	Unique Key	用户名称
user_password	varchar	255	Not Null	用户密码
user_photo	varchar	100	Not Null	用户头像
register_time	bigint	20	Not Null	注册时间
is_delete	enum	0	Not Null	删除用户标识量

(2) 管理员 admin 表

管理员 admin 表是用来存放系统管理员的基本信息,添加管理员、修改管理员信息、删除管理员,都需要对 admin 表进行增加、删除、查询、修改。admin_id 为该表的主键,随着管理员添加而自动增加,如下表 2 所示。

表2 管理员信息表

字段名	数据类型	长度	约束	描述
admin_id	int	11	Primary Key	管理员id
admin_name	varchar	255	Not Null	管理员名称
cellphone	varchar	255	Not Null	手机号码
register_time	varchar	20	Not Null	注册时间
password	varchar	255	Not Null	密码

(3) 趣事评论的用户 comment_user 表

评论的用户 comment_user 表是用来存放每一条评论和发表该评论的用户的数据库表。用户发表评论时,需要把用户 id 和评论的趣事 id 存取进去该表。id 为该表的主键,随评论被发表而自动增加,如下表 3 所示。

表3 趣事评论用户表

字段名	数据类型	长度	约束	描述
id	int	11	Primary Key	编号id
user_id	bigint	10	Not Null	用户id
things_id	bigint	10	Not Null	趣事id

(4) 趣事信息 things 表

趣事信息 things 表用来存放所有的趣事的基本信息的数据库表，获取热门、新鲜、纯文、带图趣事，发布趣事以及管理员管理趣事都需要对趣事表进行增删查改操作。things_id 为该表的主键，随趣事的发表则自动增加；has_img 表示该趣事是否带图片，1 为带图片，0 为不带图片；is_delete 表示该趣事是否被删除，0 为没有被删除，1 表示该趣事已被删除，具体如下表 4 所示。

表4 趣事信息表

字段名	数据类型	长度	约束	描述
things_id	bigint	10	Primary Key	趣事ID
publish_time	bigint	20	Not Null	趣事发布的时间
things_content	varchar	255	Not Null	趣事内容
things_image	varchar	255	Not Null	趣事图片
funny_num	int	5	Not Null	趣事点赞数
unfunny_num	int	5	Not Null	趣事踩数
user_id	bigint	10	Not Null	发表趣事的作者ID
comment_num	int	5	Not Null	趣事评论数
is_approval	int	1	Not Null	趣事是否通过审核标志量
favorite_num	int	5	Not Null	趣事收藏数
has_img	int	1	Not Null	趣事是否带图片标志量
is_delete	enum	0	Not Null	趣事是否被删除标志量

（5）被收藏过的趣事 favorite_things 表

被收藏过的趣事 favorite_things 表是用来记录每一条被不同用户收藏过的趣事的数据库表。一个用户收藏了某一条趣事时，该用户 id 和该趣事 id 就会被写入 favorite_things 表，id 为该表的主键，随用户收藏而自动增加，如下表 5 所示。

表5 被收藏过的趣事表

字段名	数据类型	长度	约束	描述
id	int	11	Primary Key	编号id
user_id	bigint	11	Not Null	用户id
things_id	bigint	11	Not Null	趣事id

（6）被点赞过的趣事 funny_things 表

被点赞过的趣事 funny_things 表是用来记录所有被不同用户点赞过的趣事的数据

库表,一个用户点赞了某一条趣事时,该用户 id 和该趣事 id 就会被写入 funny_things 表, id 为该表的主键,随用户点赞而自动增加,如下表 6 所示。

表6 被点赞过的趣事表

字段名	数据类型	长度	约束	描述
id	int	11	Primary Key	编号id
user_id	bigint	11	Not Null	用户id
things_id	bigint	11	Not Null	趣事id

(7) 回复的评论 reply 表

回复的评论 reply 表是用来保存所有回复的评论的数据库表,当用户对某一评论进行回复时,回复的基本信息就会被写入这个表,一条评论可以用很多条回复,但是一条回复只对应一条评论。reply_id 为该表的主键,随着回复操作而自动增加,如下表 7 所示。

表7 评论回复表

字段名	数据类型	长度	约束	描述
reply_id	int	10	Primary Key	评论回复id
reply_user	int	10	Not Null	回复用户id
replied_user	int	10	Not Null	被回复的用户id
comment_id	int	10	Not Null	回复所属评论
reply_content	varchar	255	Not Null	回复的内容
reply_time	bigint	20	Not Null	回复的时间
reply_user_name	varchar	255	Not Null	回复用户的用户名
replied_user_name	varchar	255	Not Null	被回复的用户名
things_id	int	10	Not Null	评论所属的趣事id

(8) 趣事评论 comment 表

趣事评论 comment 表是用来存放趣事评论的数据库表,获取趣事评论、评论趣事以及管理员对趣事进行管理,如删除评论、查询评论都需要对该表进行增加、删除、查找操作。comment_id 为该表的主键,随着评论的发表而自动增加;is_delete 标志着该趣事是否被删除,0 为未执行删除,1 表示该评论已被删除,如下表 8 所示。

表8 趣事评论表

字段名	数据类型	长度	约束	描述
comment_id	int	11	Primary Key	评论ID
content	varchar	255	Not Null	评论内容
user_id	int	11	Not Null	发表评论的作者
things_id	int	11	Not Null	被评论的趣事
comment_time	bigint	20	Not Null	评论的时间
is_delete	enum	0	Not Null	评论是否被删除标志量

(9) 被踩过的趣事 unfunny_things 表

被踩过的趣事 unfunny_things 表是用来记录所有被不同用户踩过的数据库表，用户对趣事进行踩操作时，会把用户 id 和被踩的趣事 id 存入该表，id 为该表的主键，随着踩操作而自动增加，如下表 9 所示。

表9 被踩过的趣事表

字段名	数据类型	长度	约束	描述
id	int	11	Primary Key	编号id
user_id	bigint	11	Not Null	用户id
things_id	bigint	11	Not Null	趣事id

5 详细设计

5.1 存储设计

一个完整的系统平台，需要一个容器来存储信息。本系统选择采用 Mysql 来长期存储主要的数据库信息，而后端使用 session，前端使用 localStorage 来存储用户的登陆信息。因此，存储设计主要阐述 session、数据库连接设置、前端存储 localStorage 这三个数据存储容器。

5.1.1 session

session 又称为会话状态，是 Web 系统中最常用的状态，用于维护和当前浏览器实例相关的一些信息。

在登陆过程中，baseController.php 会将用户 id 保存在 session 中，以判断和维持用户的登陆状态。同时在需要用到用户 id 的 action 方法中也可通过获取 session 中的用户 id 来调用。

```
// 存储session
$_SESSION = [];

$_SESSION[SESSION_LOGGED_USERID] = $data['data']['user_id'];

$this->_uid = $_SESSION[SESSION_LOGGED_USERID];

// 如果session中有存储用户id，则从session中获取用户id，如果this->_uid不为0，表示用户已登录
if(isset($_SESSION[SESSION_LOGGED_USERID]) &&
$_SESSION[SESSION_LOGGED_USERID]>0)
{
    $this->_uid = $_SESSION[SESSION_LOGGED_USERID];
}
```

5.1.2 数据库连接配置

与数据库连接成功后就可以对数据库进行增删查改，无需开启与数据库的连接也无需关闭数据库的连接，数据库连接配置如下所示。

```
db.DRIVER = mysql      ; 使用的数据库类型
db.HOST = 127.0.0.1    ; 数据库主机名
db.NAME = fun_things   ; 数据库名称
db.USER = root         ; 连接数据库的用户名
db.PWD  = root         ; 连接数据库的密码
db.PORT = 3306         ; 数据库端口
db.CHARSET = utf8mb4   ; 数据库使用的字符编码
db.MASTER_HOST = 127.0.0.1
db.SLAVE_HOST = 127.0.0.1
```

5.1.3 localStorage 前端存储

HTML5 中的 Web Storage 提供了两种存储方式：sessionStorage 和 localStorage。sessionStorage 用于本地存储一个会话（session）中的数据，这些数据只有在同一个会话中的页面才能访问并且当会话结束后数据也随之销毁。因此 sessionStorage 不是一种持久化的本地存储，仅仅是会话级别的存储。而 localStorage 用于持久化的本地存储，除非主动删除数据，否则数据是永远不会过期的。同域名下的页面都可以访问 localStorage 中存储的数据。

```
// 使用localStorage存储信息
localStorage.setItem('userInfo',userInfo);
// 获取localStorage中存储的信息 key-value
localStorage.getItem('userInfo');
// 清除localStorage中存储的信息
localStorage.clear();
```

5.2 基本设计

本平台采用前后端分离的设计方式，前端和后端数据交互完全通过接口，前端通过 ajax 发起 http 请求，后端 php 代码中的 Controller 接收处理前端发起的请求，通过 new model，调用 model 中对应的方法来进行返回信息或者数据给 Controller，Controller 再把请求的结果返回给前端，前端对请求的结果进行处理并展示，至此一个完整的数据交互完成。前端会验证必填项用户是否有输入以及输入是否符合规范，如果输入不正确，会弹提示框，也不会发起 http 请求，在前端而不是后台做这些简单的验证可以避免发起不必要的 http 请求，提高性能，优化用户体验。

```

public function registerAction(){ // 示例 - Action

    $oauthData['username']=isset($this->_postData['username']) ? $this->_postData['username']: "";

    $oauthData['password']=isset($this->_postData['password']) ? $this->_postData['password']: "";

    $model = new \Web\ThesisModel();

    $data = $model->register($oauthData);

    echo json_encode($data);

}

public function register($param){ // 普通用户注册 - model

    $options['table'] = 'user';

    $photo = '/uploads/avatar/default-avatar.png';

    $tmpData = array('user_name'=>'?', 'user_password'=>'?', 'user_photo'=>'?', 'register_time'=>'?');

    $options['param'] = array($param['username'], md5($param['password']), $photo, time());

    $status = $this->db->add($tmpData, $options);

    if($status !== FALSE){ return $this->returnResult(200); }

    else { return $this->returnResult(4000); }

}

```

5.2.1 ajax 发起异步 http 请求

前端代码封装了 jQuery 的 ajax 代码，让前端发起 http 请求更方便快捷，代码更少，并把所有公共方法都放置在 util.js 工具文件中。

```

util.ajax = (opt, other) => {

    var defer = $.Deferred();

    if (!opt.dataType) { opt.dataType = 'json'; }

    $.ajax(opt).done((result) => {

        if (opt.dataType === 'json' && result.code !== 0) { defer.reject(result); }

        else { defer.resolve(result); }

    }).fail(() => {defer.reject({msg: '网络出错'})});

    return defer.promise();

}

```

前端使用 ES6 标准中的 promise 对象来实现函数的回调。一个 Promise 对象可以理解为一个将要执行的操作（常常用于异步操作），使用了 Promise 对象之后可以用一种链式

调用的方式来组织代码，让代码更加直观。使用 jQuery 中的用来解决回调函数函数的 deferred 对象来处理发起 http 请求这种耗时的操作，结合 Promise 对象中的 resolve 方法可以使 Promise 对象的状态改变成成功，同时传递一个参数用于后续成功后的操作，而 reject 方法则是将 Promise 对象的状态改变为失败，同时将错误的信息传递到后续错误处理的操作。经过封装改造过后的 ajax 方法，前端把所有的接口调用都写到了 service.js 中，调用后台接口只需要如下示例：

```
service.register = (data) => { // 注册

    return util.ajax({

        url: '/thesis/register',

        type: 'POST',

        data: {

            username: data.username ,

            password: data.password

        }

    });

}
```

5.2.2 后台控制器 controller 接口

前端发起的 http 请求会连接到后台的 controller action 中对应的接口，接口 action 会处理前端发送的请求参数后新建对应的 model 对象进行数据处理，如下示例：

```
// 普通用户登录接口 - Action

public function loginAction(){

    $oauthData['username']=isset($this->_postData['username']) ? $this->_postData['username']: "";

    $oauthData['password']=isset($this->_postData['password']) ? $this->_postData['password']: "";

    $model = new \Web\ThesisModel();

    $data = $model->login($oauthData);

    echo json_encode($data);

}
```

5.2.3 后台模型 model

后台的控制器 controller action 接收了前端发起的 http 请求并新建了一个 model 之后，model 就会根据前端的请求对数据库进行对应的增加、删除、修改和查询操作，如下示例：

```
//注册 – model

public function register($param){

    $options['table'] = 'user';

    $photo = '/uploads/avatar/default-avatar.png';

    $tmpData = array('user_name'=>'?', 'user_password'=>'?', 'user_photo'=>'?', 'register_time'=>'?');

    $options['param'] = array($param['username'], md5($param['password']), $photo, time());

    $status = $this->db->add($tmpData, $options);

    if($status !== FALSE){

        return $this->returnResult(200);

    }

    else {

        return $this->returnResult(4000);

    }

}

}
```

5.2.4 上传图片

在用户修改自己的头像以及发表带图片的趣事时都涉及到了图片上传。因为只允许上传图片，所以要对文件上传的类型格式进行限制。在前端已经使用了 **HTML5** 啊 **accept** 属性限制了 **input** 表情只能上传图片，所以用户在点击上传文件按钮时，只能选择图片，但是客户端的所有东西都是不安全的，因此为了保证安全还需要在服务端进行验证。

```
// 前端限制用户只能选择上传图片

<input id="user_avatar" size="30" type="file" accept="image/*">

// 服务器端验证文件上传格式

$uptypes = array('image/jpg', 'image/jpeg', 'image/png', 'image/gif'); // 图片类型

if(!in_array($_FILES['things_img']['type'], $uptypes)){ // 如果文件类型不在图片类型数组中

    $data['code'] = 1;

    $data['msg'] = '只能上传图片';

    echo json_encode($data);die;

}

}
```

为了防止用户已经上传的图片因为他人新上传的同名图片覆盖掉，因此需要在每次

保存图片时，先将图片按一定的规律重新命名。这里采取将当前上传图片的时间戳添加到原本文件名的末尾作为文件名后缀的方法来防止图片名字重复的情况。

```
$template = $_FILES['things_img']['name']; // 取得上传文件的文件名
$photo = explode('.', $template); // 把文件名与文件扩展名拆分为数组
$time = '_' . time() . '.'; // 取得当前时间戳后缀
$photo_name = $photo[0] . $time . $photo[1]; // 重新组装图片名获取新图片名
$res = move_uploaded_file($tmp_name, 'uploads/things_img/' . $photo_name); // 将上传的文件移动到新位置
```

5.3 游客功能设计

趣事平台首页的菜单栏，游客用户和普通用户所有操作的导航和入口都在该菜单栏中，如下图 19 所示。



图 19 趣事平台首页菜单栏

5.3.1 注册

游客只需要点击平台首页头部右上角的“登陆/注册”平台就会弹出登录框，点击登录框右下角的“注册”即可切换为注册框。

在注册弹框中正确输入用户名和密码以及重复密码点击确定按钮即可以发起注册的 http 请求。如果用户名没有被其他用户注册，就可以注册成功了。

5.3.2 查看趣事

点击图 19 菜单栏中的“热门”或者“新鲜”或者“文字”或者“图片”即可以查看对应类型的趣事。趣事以每页 20 条进行分页显示。

热门趣事是根据趣事评论数量由高到低进行排序；新鲜趣事是根据用户发布趣事的时间先后进行排序，越先发布的趣事越排在后面；文字趣事则是没有带图片的趣事，按照用户发布趣事的时间先后进行排序，越先发布的趣事越排在后面；图片趣事就是带有图片的趣事，用户发布趣事的时间先后进行排序，越先发布的趣事越排在后面。每条趣事都可以查看到发布趣事的时间、作者头像和用户名，以及都有点赞、踩、评论和收藏按钮。但是没有登陆的游客用户是不能进行点赞、踩、和收藏功能，如下图 20 所示

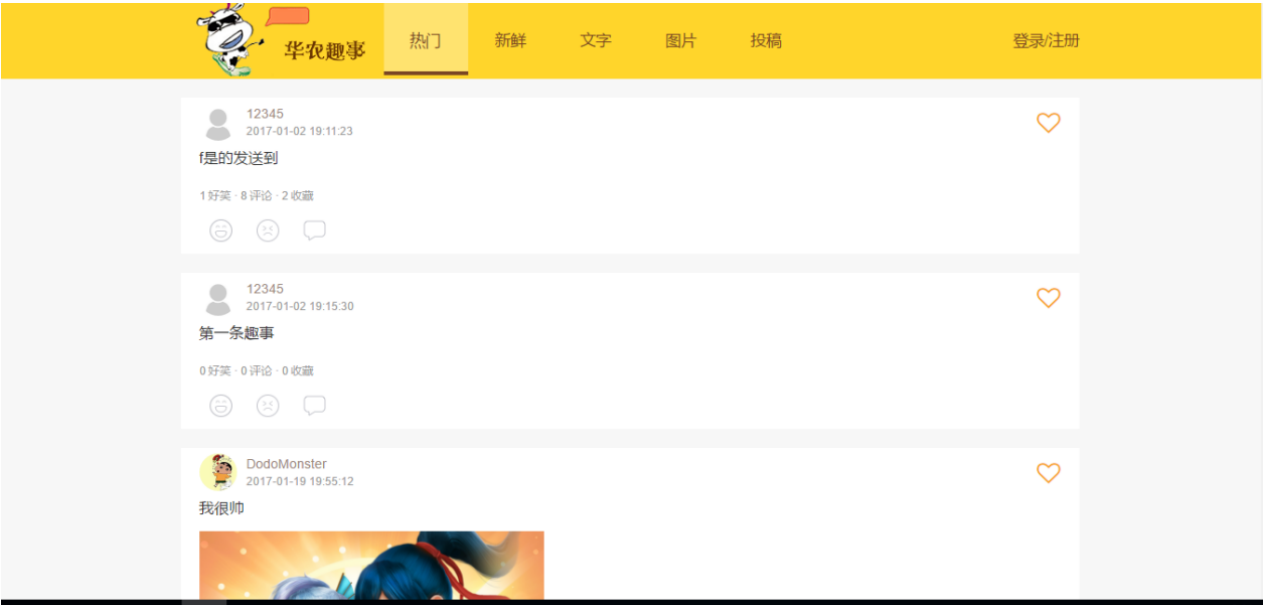


图 20 游客查看趣事

5.3.3 游客查看评论

点击“图 20 游客查看趣事”中的评论按钮或者直接点击趣事内容即可以查看该条趣事对应的评论。由于评论可能会过多，因此分页显示，每页 20 条。但是没有登陆的游客只可以查看评论，不可以进行评论和回复评论功能，如下图 21 所示。

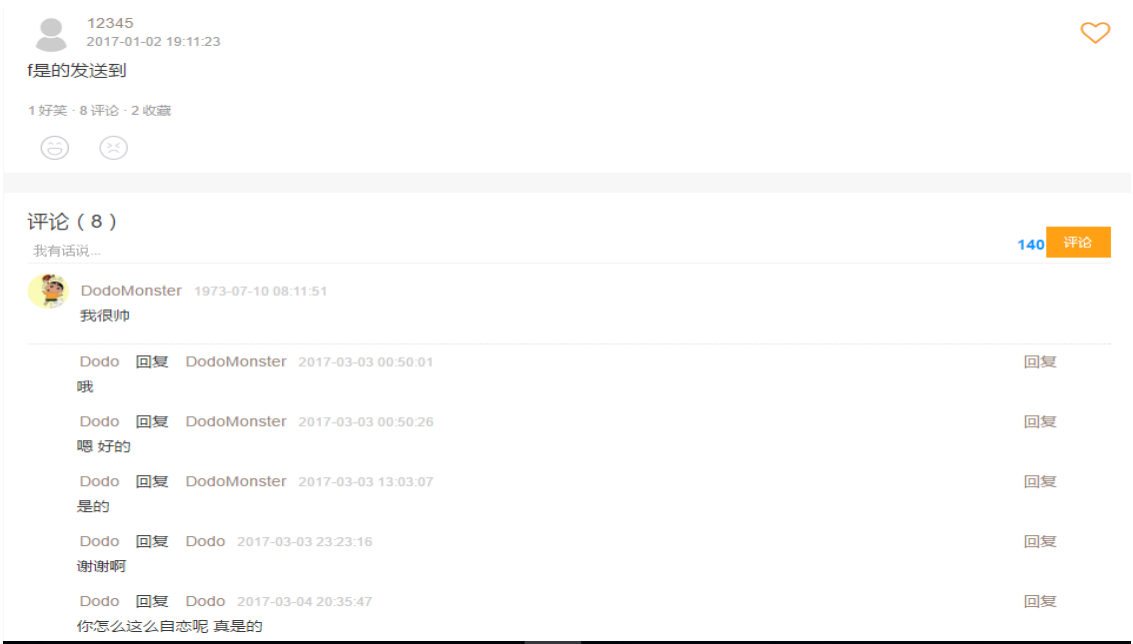


图 21 游客查看评论

5.3.4 查看他人的用户空间

点击“图 20 查看趣事”或者图“图 21 查看评论中”中的趣事作者头像或者用户名，即可以查看他人的用户空间。可以查看作者发表的趣事、作者收藏的趣事，以及他人给这个作者的评论，以及该作者在平台中的基本概况，如其发表的趣事总数、收获的评论数、收获的笑脸数、收获的收藏数、以及注册的平台“年龄”，如下图 22 所示。

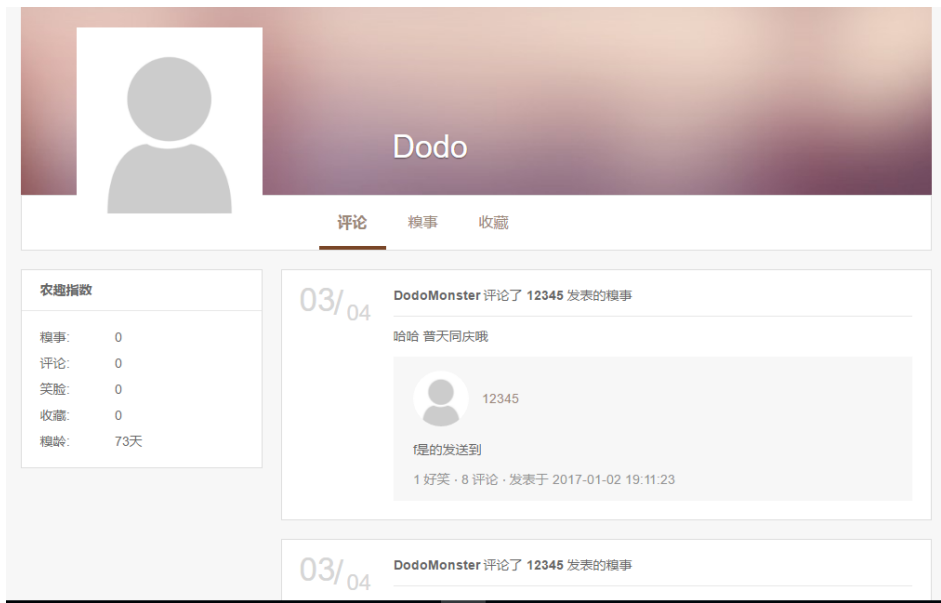


图 22 查看他人的用户空间

5.4 普通用户功能设计

5.4.1 登陆

已经注册的用户想要登陆平台只需要点击平台首页头部右上角的“登陆/注册”平台就会弹出登录框，用户名和密码不能为空，否则会弹框提示，不能正常登陆到平台。正确输入用户名和密码后就可以成功登陆，并且跳转到首页的“热门趣事”页面。后台代码会在核对普通用户的用户名和密码与存储在数据库中的一致时返回登陆成功之前会把用户id 存储到服务器的 session 中，以便识别以及保持用户的登陆状态。

5.4.2 发布趣事

点击“图 19 菜单栏”中的“投稿”，即可跳转至发布趣事页面。

普通用户只要登陆成功，即可以使用发布趣事功能，趣事内容不能为空，否则弹框提示不能成功发布趣事，如需发布带图趣事则可以点击选择文件按钮选择图片，点击“投递”按钮即可以成功发布趣事，并跳转到“新鲜趣事”页面，如下图 23 所示。



图 23 发布趣事

5.4.3 普通用户查看趣事

普通用户登陆成功后看到的趣事列表和游客看到的没有太多区别，但是如果用户对某趣事有进行过点赞、或者踩或者收藏操作，则会高亮显示。

5.4.4 点赞趣事

普通用户登陆成功可对趣事进行点赞操作，点击趣事列表中的点赞图标按钮，即可以点赞操作成功，点赞图标就会高亮显示。

5.4.5 踩趣事

普通用户登陆成功可对趣事进行踩操作，点击趣事列表中的踩图标按钮，即可以踩操作成功，踩图标按钮也会高亮显示。

5.4.6 评论趣事

普通用户登陆成功后点击“图 20 查看趣事”中的评论按钮或者直接点击趣事内容即可以进入查看该条趣事对应的评论并该页面中输入评论后，点击评论按钮就可以进行评论，发表的评论会出现在评论列表中。

5.4.7 回复评论

普通用户登陆成功后进入到“图 21 游客查看评论”中的评论列表，点击想要回复的评论右侧的“评论”就会显示回复输入框，输入回复，点击提交按钮即可回复成功，也可以点击关闭按钮取消回复评论。

普通用户登陆成功后进入到自己的用户空间中的“评论”查看关于我的评论，在输

入框中输入回复的内容，点击回复按钮也可以回复评论成功。

5.4.8 收藏和取消收藏趣事

普通用户登陆成功后在首页查看趣事中每一条趣事中的右上角的收藏图标按钮，即可进行收藏，收藏后的图标会高亮显示，收藏后的趣事可以在我的用户空间中的“收藏”中查看。而收藏后的趣事，可以点击取消按钮图标取消收藏，取消收藏后的趣事便不会出现在在我的用户空间中的“收藏”列表中。

5.4.9 修改个人信息

普通用户登陆成功后点击首页导航右侧自己的用户头像或者用户名即可进入自己的用户空间，在自己的用户空间中的设置即可以修改自己的用户信息，如下图 24 所示：

农趣指数

趣事: 2


评论: 0

笑脸: 3

收藏: 1

粮龄: 74天

更换头像



选择文件 未选择任何文件

确定上传

图片支持JPG格式，尺寸小于200x200像素，文件容量2M以内。

修改用户名

请输入用户名

DodoMonster

确认修改

修改密码

当前密码

新密码

重复新密码

确认修改

帐号

退出登录

图 24 个人空间-设置

普通用户用户注册的时候只自定义了自己的用户名和密码，而系统为新注册的用户设置了默认的头像，用户可以在自己的用户空间中的设置可以修改自己的头像，点击“图 24 个人空间-设置”选择文件按钮，选择文件后再点击确定上传按钮，就可以成功修改用户头像。

普通用户可以在自己的用户空间中的设置修改自己的用户名，在“图 24 个人空间-设置”请输入用户名中的输入框输入新用户名，点击确认修改用户名中的修改按钮，就可以成功修改用户名。

普通用户可以在自己的用户空间中的设置修改自己的密码，在“图 24 个人空间-设置”中的修改密码中的输入框中分别输入自己的当前密码、新密码以及重复密码，只有正确输入了自己以前的密码才能修改密码成功，如果忘记了密码就只能重置密码。

5.5 管理员用户功能设计

为了保证平台能正常、规范地运行下去，因此需要有管理员对平台进行管理。在后台管理系统可以对开放平台的用户、趣事、评论以及系统管理员进行管理。后台管理系统主页菜单栏，如图 25 所示：



图 25 后台系统菜单栏

5.5.1 管理员登陆

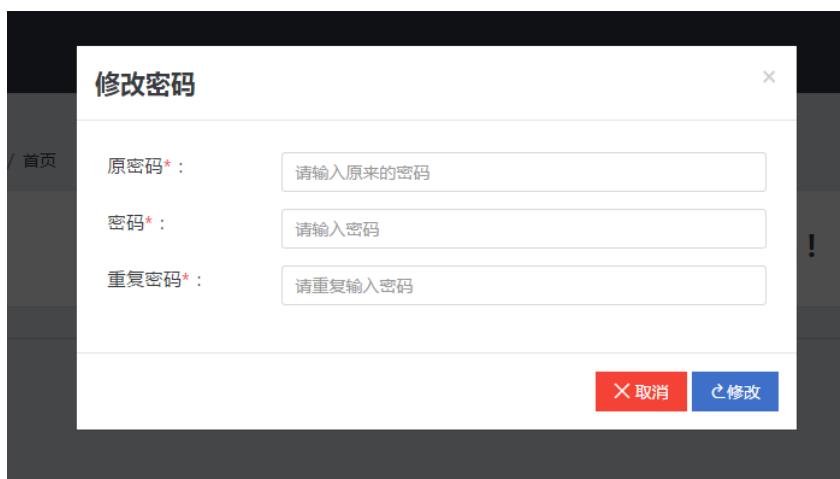
管理员用户可以进入平台后台管理系统登陆页面，输入自己的用户名和密码，点击登陆按钮，就可以成功登陆后台管理系统，进入到后台管理系统首页。用户名和密码不能为空，否则会弹框提示，不能正常登陆。

5.5.2 后台系统头部功能

管理员用户登录进来，系统会根据用户登录时存取到 session 中的 uid 来获取用户信息，然后显示在首页头部的管理员信息。

点击上图中右侧的退出系统图标即可退出系统，清除 session 以及 localStorage 中存储的信息。

点击上图中修改密码按钮，会出现修改密码弹框，如下图 26 所示：

A modal dialog box titled "修改密码" (Change Password) with a close button (X) in the top right corner. It contains three input fields: "原密码*" (Original Password) with placeholder text "请输入原来的密码", "密码*" (Password) with placeholder text "请输入密码", and "重复密码*" (Repeat Password) with placeholder text "请重复输入密码". At the bottom right, there are two buttons: a red "取消" (Cancel) button and a blue "修改" (Modify) button.

Field	Placeholder
原密码*	请输入原来的密码
密码*	请输入密码
重复密码*	请重复输入密码

图 26 后台修改密码弹框

用户只要正确输入自己的原密码、新密码以及重复新密码，就可以成功修改自己的密码。如果用户原密码不正确或者密码和重复密码不一致都会弹出提示框，导致不能成功修改密码。

5.5.3 用户管理

点击“图 26 后台系统菜单栏”中的“用户管理”-“用户列表”进入到用户管理页面。在用户管理页面可以查看所有用户，用户列表进行了分页处理每页 20 条数据，可以查看到普通用户的用户 id、用户名称、注册时间等用户信息，如下图 27 所示。

A screenshot of the "用户列表" (User List) page in a management system. The page has a sidebar with navigation links: "用户管理" (User Management), "用户列表" (User List), "赛事管理" (Event Management), "评论管理" (Comment Management), and "系统管理" (System Management). The main content area shows a table of users with columns: "序号" (Serial Number), "用户id" (User ID), "用户名称" (User Name), "注册时间" (Registration Time), and "操作" (Action). There are three users listed. Above the table is a search bar with the label "用户名称:" and a "搜索" (Search) button. Below the table, it says "总共 3 条" (Total 3 items).

序号	用户id	用户名称	注册时间	操作
1	1	DodoMonster	2017-01-02 19:11:23	编辑资料 删除
2	2	12345	2017-01-02 19:08:03	编辑资料 删除
3	3	Dodo	2017-01-02 18:54:43	编辑资料 删除

图 27 用户管理页面

在用户名称输入框中输入用户名后点击上图中的搜索按钮即可以检索用户。

点击上图中的添加用户按钮，添加用户弹框就会出现。在弹框中输入用户名、密码和重复密码，点击下图的提交按钮即可以成功添加用户。如果用户名或者密码或者重复密码为空或者密码和重复密码不一致都会添加失败，弹出错误提示框，如下图 28 所示。

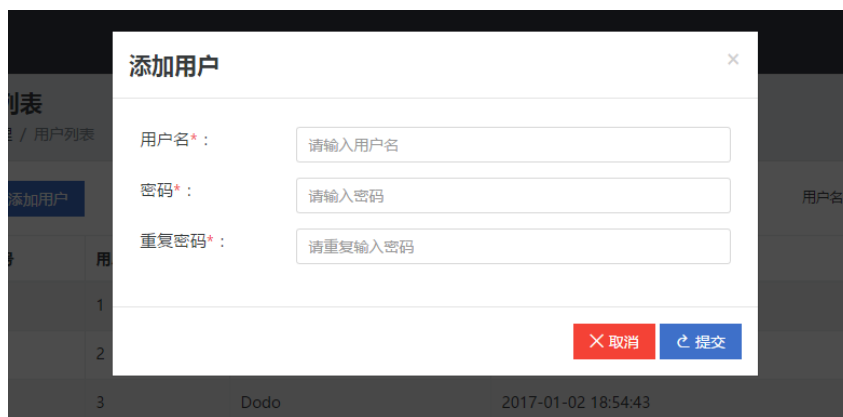


图 28 添加用户弹框

点击“图 27 用户管理页面”的重置密码按钮，为普通用户重置密码的弹框就会出现，如图 29 所示。在弹框中输入普通用户的新密码和重复密码，点击下图的重置按钮即可以为普通用户重置密码成功。如果新密码或者重复密码为空或者新密码和重复密码不一致都会添加失败，弹出错误提示框，如下图 29 所示。



图 29 重置普通用户密码弹框

点击“图 27 用户管理页面”的删除图标按钮，会弹出警告提示框。如果点击了确定按钮，则用户会被删除；点击关闭则取消删除用户并关闭弹框。

5.5.4 趣事管理

点击“图 25 后台系统菜单栏”中的“趣事管理”-“趣事列表”进入到趣事管理页面。在趣事管理页面可以查看所有普通用户发表的趣事，趣事列表进行了分页处理每页 20 条数据，可以查看到趣事的趣事 id、趣事内容、趣事发布的作者、趣事的发布时间、趣事的图片、趣事点赞数、趣事被踩数、趣事被收藏数、趣事评论数等趣事信息，如下图

30 所示。



图 30 趣事管理页面

在趣事内容输入框中输入趣事内容后点击上图中的搜索按钮即可以检索趣事。

普通用户发布的趣事必须要经过管理员审核过后才可以显示在公开平台的趣事列表中。管理员可以点击上图中的审核图标按钮，审核用户警告提示弹框就会出现。如果管理员点击了确定按钮，则该趣事就会被审核通过显示在公开平台的趣事列表中；而点击关闭则取消审核该趣事并关闭弹框。

点击“图 30 趣事管理页面”的删除图标按钮，会弹出警告提示框，再次询问管理员是否确定删除该趣事，如果管理员点击了确定按钮，则趣事会被删除不会出现在趣事列表中；点击关闭则取消删除用户并关闭弹框。

5.5.5 评论管理

点击“图 25 后台系统菜单栏”中的“评论管理”-“评论列表”进入到评论管理页面。在评论管理页面可以查看所有普通用户发表的评论，评论列表进行了分页处理每页 20 条数据，可以查看到评论的评论 id、评论内容、评论发布的作者、评论的发布时间、被评论的趣事 id、被评论的趣事图片、被评论的趣事内容等评论信息，如下图 31 所示。



图 31 评论管理页面

在评论内容输入框中输入评论内容后点击上图中的搜索按钮即可以检索评论。

点击“图 31 评论管理页面”的删除图标按钮，会弹出警告提示框，再次询问管理员是否确定删除该评论，如果管理员点击了确定按钮，则评论会被删除不会再出现在评论列表中；点击关闭则取消删除评论并关闭弹框。

5.5.5 系统管理

点击“图 25 后台系统菜单栏”中的“系统管理”-“管理员列表”进入到系统管理页面。在系统管理页面可以查看系统现有的所有管理员及其信息。我们可以在管理员列表中查看管理员的 id、用户名、手机号、注册时间，如下图 32 所示。



图 32 管理员列表页面

点击上图中的添加管理员按钮，添加管理员弹框就会出现。在弹框中输入管理员用户名、密码、重复密码和手机号码，点击下图的提交按钮即可以成功添加管理员用户，点击取消则取消添加管理员并关闭弹框。带红色*号的为必填项，所以如果管理员用户名、密码或者重复密码为空或者密码和重复密码不一致都会添加失败，弹出错误提示框。

点击编辑管理员图标按钮，编辑管理员弹框就会出现。如果只是想修改用户名、手机号码则可以直接在弹框中输入管理员用户名、和手机号码，点击提交按钮即可以成功编辑管理员用户信息，点击取消则取消编辑管理员的信息并关闭弹框。带红色*号的为必填项，所以如果管理员用户名为空或会添加失败，弹出错误提示框。

而如果想重置管理员密码，则可以重置密码按钮，弹框就会显示重置密码的输入框。输入密码和重复密码而且两者一致点击提交按钮就可以重置用户密码成功了，而点击取消重置密码按钮弹框就会取消重置密码操作，如下图 33 所示。

编辑管理员资料

管理员名称 * : Dodo

密码 * : 请输入密码

重复密码 * : 请重复输入密码

手机号码 : 18814127439

取消重置密码

X 取消 ✓ 提交

图 33 编辑管理员密码用户弹框

点击“图 33 管理员列表页面”的删除图标按钮，会弹出警告提示框，再次询问管理员是否确定删除该管理员，如果管理员点击了确定按钮，则该管理员会被删除不会再出现在管理员列表中也不能再正常登陆后台管理系统对平台进行管理；点击关闭则取消删除该管理员并关闭弹框。

6 结论与展望

6.1 结论

本系统平台是基于 WEB 的华农趣事平台，旨在为广大华农师生提供公共平台去分享趣事，分享快乐，促进沟通交流。在做本系统平台时，本人一直只专注做前端开发，对后台开发不熟悉，因此花了较多时间在学习 PHP 后台编程语言，以及对数据库的设计一直比较迷惑，尤其是在需要把趣事评论用户三者以及对应操作的关系关联起来，因此也请教了不少前辈。

在本华农趣事平台中，后台使用 PHP 语言整合 Yaf 框架，前端使用 Vue.js 这个 MVVM 框架进行完全的前端分离的单页面 SPA 开发方式，采用 MVC 的架构模式强制分离了业务逻辑与、用户的方式来组织代码，将业务逻辑聚集到一个部件里面，使得在改进和个性化定制界面及用户交互的同时，不需要重新编写业务逻辑。以 Apache 作为 Web 服务器，采用体积小、速度快、成本较低、运行可靠的 Mysql 数据库对数据进行存储，减低成本和保证系统的稳定性。另外对于经常需用到的数据，前端采用了 localStorage 这个 HTML5 的存储容器来进行存储查询，避免了经常发起不必要的 HTTP 请求而影响系统的性能。选择采用 Vue.js 这个 MVVM 框架使平台系统是一个单页面应用，利用 Vue.js 数据绑定的特性在无刷新页面的情况下更新数据，而且使用 Vue-Router 这种前端路由在用户无需刷新页面的情况下跳转切换页面，大大提升了系统平台的性能以及让用户体验达到最佳。对于用户重要敏感的信息，如密码，本系统平台采取了 MD5 安全加密技术对其进行处理，数据库表并不直接存储用户设置的密码，而是把用户的原始密码经过 MD5 Hash 运算加密处理过后得出的一段字符串存储在数据库中。这样即使具有数据库管理权限的人员查数据库也无法获取到用户的真正密码得到的也只是 MD5 Hash 运算过后得出的字符串，从而避免普通用户的信息和身份被盗用，而摘要信息向源信息的转变是一个不可逆的过程。而且在平台系统中有关用户信息的查询并不会把用户密码查询出来，防止被有心人利用操作。再者即使服务器所处状态为不安全时，用户的重要信息也不至于泄露，从而真正地保证了用户的信息安全。

在本平台中，一共有三种类型的用户，游客、普通用户和管理员用户。所有人都可以进入注册平台、平台首页浏览查看趣事、评论以及进入普通用户的用户空间查看他人发表的趣事、他人的收藏以及有关他的评论；已经注册平台的用户可以登录平台对趣事进行点赞、踩、评论、收藏趣事和取消收藏趣事以及回复评论的操作，同时进入自己的用户

空间修改自己的基本资料，如头像、用户名和密码；后台管理员可以登录后台管理系统对平台的用户、趣事、评论以及系统管理员进行增删查改，普通用户发表的趣事需经过管理员审核通过才能显示在开放平台的趣事列表中。

6.2 展望

本平台是一个较庞大的系统平台，数据库表虽然只有 9 个，但平台实际是由两个系统相结合组成的。开放的趣事展示平台和后台管理系统是两个相对独立的系统，只是共享数据库数据。平台系统的基本要求和功能都已经被实现了。但是，由于技术和时间有限，平台仍然存在一些问题以及功能等待完善和优化。

（1）趣事的分享功能没有实现，有待进一步实现此功能。

（2）普通用户无法实时知道有人评论了自己的趣事、别人回复了自己的评论等动态消息。

（3）在使用方面，由于平台没有经过开放而长时间的实际使用，所以可能仍然存在一些未发现的 bug 和漏洞，这一方面需要在使用中不断去发现、不断观察并完善其中的不足。

参 考 文 献

- Abraham Silberschatz (美), Henry F. Korth (美), S. Sudarshan (印). 数据库系统概念[M].北京: 机械工业出版社,2012:1-54.
- 陈涛.MVVM 设计模式及其应用研究[J].计算机与数字工程,2014(10):1982-1985.
- 陈震, 王艳.浅谈企业信息系统的密码改进措施[J].科技广场,2013(3): 158-160.
- 何永太.基于 PHP 技术的 MVC 框架的实现与应用[J].安徽水利水电职业技术学院学报,2010,(04):10-12.
- 黄华林.浅谈移动项目的本地存储[J].电脑知识与技术,2015(19):49-51.
- 惠新宸.Yaf 用户手册[EB/OL] .2014.10.22[2017.04.10].
<http://yaf.laruence.com/manual/index.html>.
- LAMP 兄弟连,高洛峰.细说 PHP[M].北京: 电子工业出版社,2013:36-90.
- 刘淇峰, 陈雪蛟.基于 Yaf 的高校教育 CMS 服务平台的设计与实现[J].河北软件职业技术学院学报,2015(02):10-13.
- 李莹莹.网络分享心理研究[D].华中科技大学,2012:1-2.
- Nicholas C.Zakas.JavaScript 高级程序设计[M].第 3 版.北京: 人民邮电出版社, 2012.3:4-41.
- 孙宁,赵维平,王宇飞,陆奇峰,王凌.中国古谱数据库管理平台设计与实现[J].现代计算机(专业版),2017(01):64-69.
- 易剑波.基于 MVVM 模式的 WEB 前端框架的研究[J].信息与电脑(理论版),2016(19):76-77.
- 尤雨溪.Vue.js 教程[EB/OL].<http://v1-cn.vuejs.org/guide/overview.html>.
- Lionel Nkenyereye,Jong-Wook Jang. Performance Evaluation of Server-side JavaScript for Healthcare Hub Server in Remote Healthcare Monitoring System[J]. Procedia Computer Science,2016:90-98.
- Munawar Hafiz, Samir Hasan, Zachary King, et al.Growing A Language: An Empirical Study On How (And Why) Developers Use Some Recently-Introduced And/Or Recently-Evolving JavaScript Features[J]. The Journal of Systems; Software,2016:10-13.
- Yassine Rhazali, Youssef Hadi, Abdelaziz Mouloudi. Model Transformation with ATL into MDA from CIM to PIM Structured through MVC[J]. Procedia Computer Science,2016:83-85.

致 谢

本毕业设计从去年 12 月份就开始动工，白天实习，晚上和周末就抽时间来干活。从最原始的需求分析、原型设计到最后整个平台各个功能都能够顺利地运行操作，自己也从中学习到很多。从对 PHP 一无所知到能够运用这门语言开发系统对数据进行增删查改，也让自己有机会独自一人完成一个完整项目的开发，熟悉了项目的开发流程，让自己更坚定地把软件开发当作自己的工作饭碗。

平台设计能够顺利完成，导师的指导和帮助是必不可少的，真的非常幸运能跟到一个尽心尽责的导师，能够在遇到自己不能解决的问题的时候还能在百忙之中抽出时间来给予帮助和解惑，能够耐心地指导论文撰写和修改。

大学四年就要过去了，马上就要毕业离开学校走向社会，另外也非常感谢大学四年里教授过我学业的老师以及为年级事务呕心沥血的辅导员老师们，感谢他们的耐心教导让自己在大学里茁壮成长让我掌握了很多专业的知识足以走向社会自己谋生；感谢学校提供一个美丽和谐的环境让我能够安心学习同时也认识了很多志同道合的同学朋友，肩并肩学习进步让自己不再孤单。

同时，我也要感谢我的父母，谢谢他们把我生下来，并且把我抚养长大，供书教学，是他们把自己的辛苦汗水化作一滴滴滋润我长大的甘露，真心感恩他们的教导与抚育！

华南农业大学
本科生毕业论文成绩评定表

学号	201330350106	姓名	陈小红	专业	网络工程	
毕业论文题目	基于 WEB 的华农趣事平台的设计与实现					
指导教师评语						
成绩（百分制）： _____ 指导教师签名： _____ 年 月 日						
评阅人评语及成绩评定	成绩评定标准	评分项目			分值	得分
		选题质量 20%	1	专业培养目标	5	
			2	课题难易度与工作量	10	
			3	理论意义或生产实践意义	5	
		能力水平 40%	4	查阅文献资料与综合运用知识能力	10	
			5	研究方案的设计能力	10	
			6	研究方法和手段的运用能力	10	
			7	外文应用能力	10	
		成果质量 40%	8	写作水平与写作规范	20	
	9		研究结果的理论或实际应用价值	20		
评阅人评语：						
成绩（百分制）： _____ 评阅人签名： _____ 年 月 日						

续上表：

	评价项目	具体要求（A 级标准）	最高分	评分				
				A	B	C	D	E
答辩委员会意见与成绩评定	论文质量	论文（设计）结构严谨，逻辑性强；有一定的学术价值或实用价值；文字表达准确流畅；论文格式规范；图表（或图纸）规范、符合要求。	60	55-60	49-54	43-48	37-42	≤36
	论文报告、讲解	思路清晰；概念清楚，重点（创新点）突出；语言表达准确；报告时间、节奏掌握好。	20	19-20	17-18	15-16	13-14	≤12
	答辩情况	答辩态度认真，能准确回答问题	20	19-20	17-18	15-16	13-14	≤12
<div>是否同意通过论文答辩（打√）</div> <div>1.同意</div> <div>2.不同意</div> <div>成绩（百分制）：_____ 答辩委员会主席（签名）：_____</div> <div>年 月 日</div>								
成绩总评	<div>论文总评分数：_____</div> <div>论文成绩总评等级：_____ 学院盖章：_____</div> <div>年 月 日</div>							

注：1、论文成绩评定等级：参考评阅人的评阅、指导教师评阅情况，结合答辩情况，建议按指导教师评分、评阅人评分、答辩评分为4：3：3的比例评定论文总成绩分数，然后按优（90—100分）、良（80—89分）、中（70—79分）、及格（60—69分）、不及格（<60分）给出成绩等级。

2、论文成绩以分数的形式登记到教务管理系统。