



**本科毕业论文**

|  |
| --- |
| **仿QQ空间完成个人主页设计与实现** |
|  |

**朱丽旋**

**201525050227**

|  |  |
| --- | --- |
| 指导教师 | **林旭东 副教授** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 学院名称 |  | **数学与信息学院** | 专业名称 |  | **网络工程** |
| 论文提交日期 |  | 2018年4月15日 | 论文答辩日期 |  | 2018年5月20日 |

# **摘 要**

随着互联网技术的发展迅速，人类科学技术不断进步，计算机科学也越来越成熟，并且一直不断的发展，成为推动时代发展的工具，其带来的便利影响着人们生活的方方面面。过去笔墨纸砚是人们记录的工具，再是现代的一纸一笔便已足够，而如今，更多人选择的是用一台联网的电子设备便可取而代之。以网络形式进行记录生活的点滴，除了记录时的方便，网络形式存储数据更安全不易丢失，且方便查找，更是过去纸质版不可比拟的。

本文主要分析了个人主页系统的实现。该系统使用浏览器/服务器模式（B/S）进行开发，用python和vscode、sublime作为开发平台及工具，采用了Flask+JQuery+boostrap框架， 用来存储数据的数据库是mysql,是主要采用Html、CSS、Javascript技术来实现前端页面的静态与动态显示，使用ajax技术来处理前端与后台数据的交互。

本系统使用邮箱号来注册用户信息，邮箱是用户的唯一标识，一个邮箱不能重复注册。在注册时需要两次次输入相同密码来增强用户对该密码的记忆。该系统主要由账号管理模块、片语模块和日记模块这三大模块构成。其中账号管理模块包括注册、登录、个人设置和切换账号功能，而个人设置还包括修改用户名和修改密码两个功能，切换账号功能使得用户可以在该浏览器登录其他账号；片语模块包括发表片语、查看所有片语、评论片语、查看片语评论功能；日记模块则包括发表日记、查看所有日记、添加分类、查看某一分类日记、评论日记、查看日记评论等功能。设计该产品初衷是希望留下当时最真的痕迹，记录的不管是好的坏的，想法或是愚蠢幼稚，所以片语、日记和评论都不支持删除、修改功能，只能通过评论。

关键词：个人主页 Flask JQuery Boostrap

**The Analysis and Design of Homepage System**

Zhu Lixuan

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

**Abstract:** With the rapid development of Internet technology, human science and technology continue to advance. Computer science is becoming more and more mature, and has been continuously developed, becoming a tool to promote the development of the times, and the convenience brought by it affects all aspects of people's lives. In the past, writing brush and ink，and paper and platform was a tool for people to record, and then a modern a paper and a pen was enough. Nowadays, more people choose to use a networked electronic device instead. Recording life in the form of a network, besides the convenience of recording, the network form to store data is more secure and not easy to lose, and easy to find, which is incomparable in the past paper version.

This article mainly analyzes the implementation of the personal homepage system . The system uses the browser / server mode (B / S) developed with python and vscode, sublime as a development platform and tools, using Flask + JQuery + boostrap frame.A database for storing data is mysql. Mainly use Html, CSS, Javascript technology to achieve static and dynamic display of front-end pages, and use ajax technology to handle the interaction between front-end and background data .

The system uses the mailbox number to register user information. The mailbox is the unique identifier of the user , and one mailbox cannot be registered repeatedly. When registering, you need to enter the same password twice to enhance the user's memory of the password . The system consists of account management module,mind module and diary module three major modules. The account management module includes registration, login, personal settings and switch account functions, and the personal settings include two functions of modifying the user name and changing the password. The switch account function enables the user to log in to other accounts in the browser; the mind module includes delivering mind , viewing all minds, commenting on the mind, view the minds comment; the diary module includes the function of publishing a diary, viewing all diaries, adding categories, viewing a category diary , commenting a diary , and viewing a diary comment . The original intention of designing this product is to hope to leave the most true traces at that time . The records are good or bad, ideas or stupid, so the words , diaries and comments do not support deletion , modification , and only comments .

**Key words:** personal homepage Flask JQuery Boostrap

目 录

[1 引言 1](#_Toc510791115)

[1.1 选题背景 1](#_Toc510791116)

[1.2 开发本系统的意义 1](#_Toc510791117)

[1.3 本系统的设计思路 1](#_Toc510791118)

[1.4 本系统开发采用的研究方法 1](#_Toc510791119)

[2 需求分析 3](#_Toc510791120)

[2.1 系统架构 3](#_Toc510791121)

[2.2 功能模块用例分析 3](#_Toc510791122)

[2.2.1 注册用例 4](#_Toc510791123)

[2.2.2 登录用例 4](#_Toc510791124)

[2.2.3 添加分类用例 5](#_Toc510791125)

[2.2.4 修改分类用例 5](#_Toc510791126)

[2.2.5 删除分类用例 6](#_Toc510791127)

[2.2.6 记账用例 6](#_Toc510791128)

[2.2.7 查询账单用例 7](#_Toc510791129)

[2.2.8 修改账单用例 7](#_Toc510791130)

[2.2.9 删除账单用例 8](#_Toc510791131)

[2.2.10 查看收支分析用例 8](#_Toc510791132)

[2.2.11 查看对账表用例 9](#_Toc510791133)

[2.2.12 找回密码用例 9](#_Toc510791134)

[2.2.13 修改密码用例 10](#_Toc510791135)

[2.2.14 修改个人信息用例 10](#_Toc510791136)

[3 数据库设计 11](#_Toc510791137)

[3.1 实体E-R图设计 11](#_Toc510791138)

[3.1.1 用户信息实体E-R图 11](#_Toc510791139)

[3.1.2 记账分类实体E-R图 11](#_Toc510791140)

[3.1.3 账单信息实体E-R图 12](#_Toc510791141)

[3.1.4 验证码实体E-R图 12](#_Toc510791142)

[3.2 数据库表设计 13](#_Toc510791143)

[3.2.1 用户信息表 13](#_Toc510791144)

[3.2.2 记账分类信息表 14](#_Toc510791145)

[3.2.3 账单信息表 14](#_Toc510791146)

[3.2.4 验证码信息表 15](#_Toc510791147)

[4 系统设计 16](#_Toc510791148)

[4.1 系统目标 16](#_Toc510791149)

[4.2 系统的功能 16](#_Toc510791150)

[4.3 系统开发环境 16](#_Toc510791151)

[4.3.1 Python简介 16](#_Toc510791152)

[4.3.2 PyCharm简介 17](#_Toc510791153)

[4.3.3 MySQL简介 17](#_Toc510791154)

[4.3.4 Django简介 17](#_Toc510791155)

[4.4 系统的设计 18](#_Toc510791156)

[4.4.1 总体流程设计 18](#_Toc510791157)

[4.4.2 注册流程设计 20](#_Toc510791158)

[4.4.3 登录流程设计 21](#_Toc510791159)

[4.4.4 记账流程设计 21](#_Toc510791160)

[4.4.5 总体模块的设计 22](#_Toc510791161)

[4.5 前端界面模块主要设计与实现 22](#_Toc510791162)

[4.5.1 表单设计 22](#_Toc510791163)

[4.5.2 表单校验 22](#_Toc510791164)

[4.5.3 日期选择 24](#_Toc510791165)

[4.5.4 图表绘制 24](#_Toc510791166)

[4.5.5 网页动态刷新 25](#_Toc510791167)

[4.6 后台逻辑模块主要设计与实现 25](#_Toc510791168)

[4.6.1 处理请求 25](#_Toc510791169)

[4.6.2 发送邮箱验证信息 26](#_Toc510791170)

[4.6.3 分页显示 27](#_Toc510791171)

[4.6.4 拦截器 27](#_Toc510791172)

[4.6.5 数据库交互 28](#_Toc510791173)

[5 系统运行效果及功能说明 29](#_Toc510791174)

[5.1 注册 29](#_Toc510791175)

[5.2 登录 29](#_Toc510791176)

[5.3 找回密码 30](#_Toc510791177)

[5.4 修改密码 30](#_Toc510791178)

[5.5 修改个人资料 31](#_Toc510791179)

[5.6 分类设置 31](#_Toc510791180)

[5.7 记账 32](#_Toc510791181)

[5.8 账单管理 32](#_Toc510791182)

[5.9 收支分析 33](#_Toc510791183)

[5.10 对账表 34](#_Toc510791184)

[6 结论与展望 35](#_Toc510791185)

[参 考 文 献 36](#_Toc510791186)

[致 谢 37](#_Toc510791187)

引言

选题背景

在网络几乎全民覆盖的时代下，互联网的发展为人们提供了更便捷的生活方式。点餐可以叫外卖，买票、购物也可以在网上点点手指鼠标就能完成，那么，发表心情、写日记为何不利用互联网科技带来的便利呢？每当我们打开自己的个人主页, 浏览其中的内容, 我们都仿佛再次经历我的成长历程。（杨昭涛 周英，2012）

国内一些个人主页, 滥用特效, 给浏览者留下混乱、无序的印象。这是因为个人主页制作中可供参考的专业技术实在太多, 个人可利用不同的网页制作技术为个人主页增色。如可将个人主页划分为很多个细小的版块, 可在界面上随意增添很多动画的效果, 以及很多弹出式广告, 有的界面很花哨, 随着鼠标的移动会有很多五颜六色的画面跟随, 虽然色彩绚丽,但在整体上影响了人们的浏览效果,造成许多使用上的障碍。网页制作技术在为个人主页的完善提供多种可行途径的同时, 也造成了个人主页因过度使用技巧而适得其反。这是个人主页的不足之处。（燕辉 2015）在人们接触过五花八门又功能繁杂的社交平台后，更倾向于想要有一个既简单，又能满足存放记录个人生活的平台。而从平台上的简单，带给人以生活上的简单化；同时在结束一天的奔波后，可以直面一个人的心灵生活，记录下所思所感，成长路上的点滴。

开发本系统的意义

开发一个可以存放个人时光照片，也可以抒写短条片刻心情，又可以记录日记心得的平台。

本系统的设计思路

1. 通过参考QQ空间相册、说说及日志的功能设计，结合自身及调查周边同学对基本功能的需求，对设计题目进行全面的需求分析；
2. 根据需求分析结果做出概要设计；
3. 根据概要设计进行详细设计，做出E-R图(概念结构设计)；
4. 在E-R图基础上进行关系模式设计（逻辑结构设计）；
5. 根据需求分析中的各种数据，得出各种视图以及各种约束、规则、触发器脚本描述；
6. 根据(5)中的内容在mysql中设计生成对应的数据库；
7. 根据需求分析中的各种数据，得出各种视图以及各种约束、规则、触发器脚本描述；
8. 用vscode和sublime text3作为开发平台及工具完成设计。

## 本系统开发采用的研究方法

该系统是基于B/S模式进行开发。开发过程中前端使用jQuery + Bootstrap框架，以及Ajax进行数据交互，服务端使用Python3 Flask框架，数据库使用MySQL5.70。

（1）整体设计采用Flask+Boostrap框架，结合运用Ajax和JQuery等技术实现。

（2）Flask是一种用Python语言编写的轻量级的web框架，它的显著特点是小，几行代码便能跑出一个web系统，核心简单且容易扩展，有内置的模板和数据库，但你完全可以随心所欲使用其他技术进行替换，因此flask较其他框架更容易上手。

（3）Bootstrap的强大之处在于它对常见的CSS布局小组件和JavaScript插件都进行了完整且完善的封装，使得开发人员（不仅是前端开发人员）信手拈来，轻松使用（深入理解Bootstrap 徐涛，2016）

（4）利用jQuery Validate插件来实现对表单数据进行验证，高效率、几乎与用户输入时同时进行验证，同时使用Ajax技术实现与服务器异步请求，提升用户使用体验。

（5）本系统实现前后端分离思想，为避免前端Ajax和服务器出现跨域请求情况，在服务端Flask引入flask\_cors设置允许跨域访问。

（6）采用MySQL数据库管理系统。逐步用上存储过程，使运行效率更高点。

（7）开发工具使用vscode和sublime text3，他们两者具有相似的功能风格，且启动快，使用vscode插件live server，充当本地调试时前端的服务器。使用自己熟悉的开发工具，来提升编程效率。

需求分析

系统架构

系统主要包括“账号管理”，“片语”，“日记”共3个模块，其中账号管理包括了注册，登录，个人设置，切换登录账号，而个人设置中又包括了修改用户名和密码；片语包括发表片语、查看所有片语、评论片语和查看片语评论；日记包括发表日记、查看所有日记、添加分类、查看某一分类日记、评论日记和查看日记评论。系统的总体架构如图1所示。

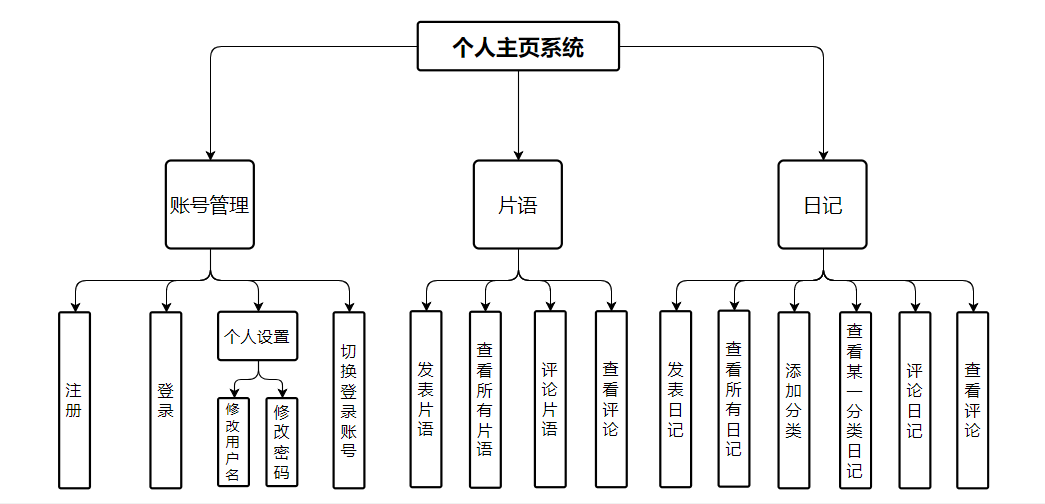


图1 系统总体架构图

功能模块用例分析

表1 个人收支管理系统功能表

|  |  |  |
| --- | --- | --- |
| 子模块 | 子功能 | 编号 |
| 账号管理 | 注册 | F1.1.1 |
| 登录 | F1.1.2 |
| 修改用户名 | F1.1.3 |
| 修改密码 | F1.1.4 |
| 切换登录账号 | F1.1.5 |
| 片语 | 发表片语 | F1.2.1 |
| 查看所有片语 | F1.2.2 |
| 评论片语 | F1.2.3 |
| 查看片语评论 | F1.2.4 |
| 日记 | 发表日记 | F1.3.1 |
| 查看所有日记 | F1.3.2 |
| 添加分类 | F1.3.3 |
| 查看某一分类日记 | F1.3.4 |
| 评论日记 | F1.3.5 |
| 查看日记评论 | F1.3.6 |

注册用例

注册：用户通过注册成为该系统用户，才能使用该系统

用例ID：F1.1.1

用例名称：注册

参与者：用户

用例描述：用户注册成为系统用户

前置任务：无

基本事件流： a、访问系统网址，进入系统主页，

b、依次hover头像 – 点击切换账号 – 点击去注册

c、正确填写相关信息

d、点击Sign Up按钮注册

e、显示注册结果

f、用例结束

说明：邮箱不能重复注册，否则会显示错误信息

登录用例

登录：验证用户身份是否存在，用户登录成功后才能使用该系统功能。

用例ID：F1.1.2

用例名称：登录

参与者：用户

用例描述：用户通过系统的登录界面填写账号密码登录

前置任务：必须先成功注册

基本事件流： a、 进入系统登录页面

b、 输入账号密码并提交

c、 显示登录结果

d、 用例结束

说明：账号和密码都正确才能登录

修改用户名用例

修改用户名：用户可以修改其用户名。

用例ID：F1.1.3

用例名称：修改用户名

参与者：用户

用例描述：用户修改其用户名

前置任务：成功登录系统

基本事件流： a、 hover个人头像

b、 点击下拉框“个人设置”链接

c、 输入更改后的用户名

d、 点击确定

e、 显示修改用户名结果

f、 用例结束

修改密码用例

修改分类：用户可以修改其用于登录的密码。

用例ID：F1.1.4

用例名称：修改密码

参与者：用户

用例描述：用户修改密码

前置任务：成功登录系统

基本事件流： a、 hover个人头像

b、 点击下拉框“个人设置”链接

c、 点击“修改密码”tab

d、 输入原密码、新密码和确认密码

e、 点击“确定”按钮

f、 显示修改密码结果

g、用例结束

切换登录账号用例

切换登录账号：用户可以切换不同的账号使用该系统。

用例ID：F1.1.5

用例名称：切换登录账号

参与者：用户

用例描述：用户切换其他账号登录该系统

前置任务：无

基本事件流： a、 hover个人头像

b、 点击下拉框“切换账号”链接

c、 输入账号密码

d、 点击“确定”按钮

e、 显示登录结果

f、 用例结束

发表片语用例

发表片语：用户发表片语。

用例ID：F1.2.1

用例名称：发表片语

参与者：用户

用例描述：用户发表短篇瞬间心情、情绪、思绪

前置任务：成功登录系统

基本事件流： a、 在主页点击加号 – 点击第一个图标，进入片语界面

b、 在输入框中输入文字

c、 点击submit按钮提交

d、 弹框提示发表结果

e、 输入框下方展示发表的片语

f、 用例结束

查看片语用例

查看片语：用户查看发表过所有片语

用例ID：F1.2.2

用例名称：查看片语

参与者：用户

用例描述：用户查看已发表过的所有片语

前置任务：成功登录系统

基本事件流： a、 在主页点击加号 – 点击第一个图标，进入片语界面

b、 显示用户发表过的所有片语

c、 用例结束

评论片语用例

评论片语：用户评论已发表的片语。

用例ID：F1.2.3

用例名称：评论片语

参与者：用户

用例描述：用户评论已发表的片语

前置任务：成功登录系统，已发表的片语

基本事件流： a、 在主页展示出来的片语或者进入片语页面

b、 在某一条片语下面点击“写评论”

c、 在弹出的模态框表单中输入评论内容

d、 弹框显示评论结果，评论按钮评论数+1

e、 用例结束

查看片语评论用例

查看片语评论：用户查看片语的评论。

用例ID：F1.2.4

用例名称：查看片语评论

参与者：用户

用例描述：用户查看已发表的片语的评论

前置任务：成功登录系统，已被评论的片语

基本事件流： a、 在主页展示出来的片语或进入片语页面

b、点击评论按钮

c、 显示评论内容

d、 用例结束

发表日记用例

发表日记：用户发表日记。

用例ID：F1.3.1

用例名称：发表日记

参与者：用户

用例描述：用户发表日记

前置任务：成功登录系统

基本事件流： a、在主页点击加号 – 点击第二个图标，进入写日记界面

b、输入标题、日记内容，选择分类，submit按钮提交

c、弹框显示发表结果

d、用例结束

查看日记用例

查看日记用例：用户查看已发表过的所有日记。

用例ID：F1.3.2

用例名称：查看日记

参与者：用户

用例描述：用户查看已发表过的所有日记

前置任务：成功登录系统，日记多于3篇

基本事件流： a、 在主页点击“日记”tab页的“查看更多”，进入日记列表界面

b、 显示用户发表过的所有日记

c、用例结束

说明：日记小于4篇的仅在主页显示

添加日记分类用例

添加日记分类：用户在发表日记选择分类时新增分类名。

用例ID：F1.3.3

用例名称：添加日记分类

参与者：用户

用例描述：用户在发表日记选择分类时可以新增分类名

前置任务：成功登录系统

基本事件流： a、 在主页点击加号 – 点击第二个图标，进入写日记界面

b、 输入标题、日记内容，点击“新增分类”按钮

c、在弹出的表单中填写分类名，点击提交

d、点击submit按钮提交日记

e、返回日记提交的结果

f、用例结束

说明：日记提交成功则分类添加成功

查看某一分类日记用例

查看某一分类日记：用户查看已发表的某一分类日记。

用例ID：F1.3.4

用例名称：查看某一分类日记

参与者：用户

用例描述：用户可以查看指定分类的日记

前置任务：成功登录系统，已发表过日记

基本事件流： a、 在主页点击“日记”tab页的“查看更多”，进入日记列表界面

b、 点击左边栏任一类别tab，或者点击文章标题下“类别”旁边的链接

c、返回只显示该类别文章

d、 用例结束

评论日记用例

评论日记：用户评论已发表的日记。

用例ID：F1.3.5

用例名称：评论日记

参与者：用户

用例描述：用户登录后进行密码的修改

前置任务：成功登录系统，已发表过日记

基本事件流： a、 在主页展示的日记，或者进入日记详情页面

b、 在日记下方点击“写评论”按钮

c、 在弹框中写评论内容

d、 点击“提交”

e、 提示修改成功

f、 用例结束

、

查看日记评论用例

查看日记评论：用户查看已被评论的日记的评论。

用例ID：F1.3.6

用例名称：查看日记评论

参与者：用户

用例描述：查看日记评论

前置任务：成功登录系统，已被评论的日记

基本事件流： a、 在主页，或者进入日记详情页面

b、 点击评论

c、 显示评论内容

d、 用例结束

数据库设计

## 实体E-R图设计

用户信息实体E-R图

系统需要用户注册输入所有个人信息才能成为该系统用户，从而进行使用系统。在用户登录时需要使用到“邮箱”和“密码”字段，用户信息实体包括用户ID、用户名、密码、邮箱等属性，用户信息实体E-R图如图2显示。

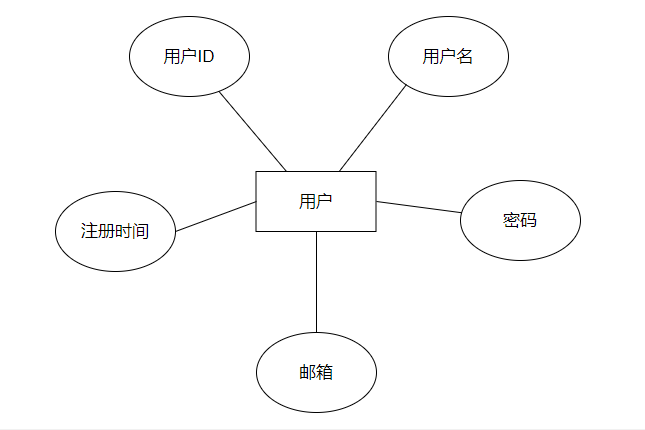


图2 用户E-R图

片语实体E-R图

用户发表片语只需要提交片语的内容即可。每一条片语产生，在数据库便生成一个片语ID，该ID为片语的主键，还根据当前时间生成发表时间字段，评论数字段计数每条片语评论数，且用用户ID字段记录该片语是由哪个用户产生。片语实体包括片语ID、发表时间、内容、评论数和用户ID属性，记账分类实体E-R图如图3所示。

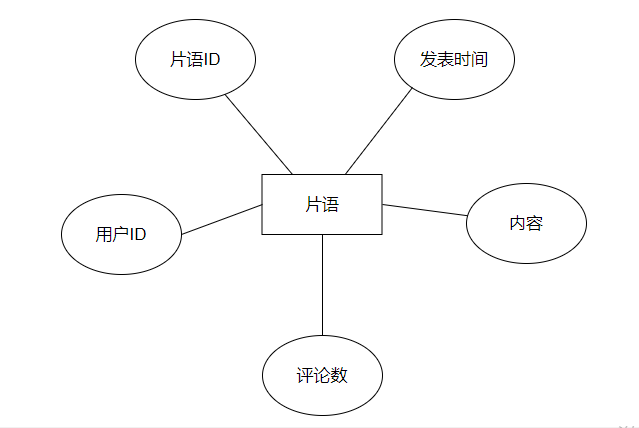


图3 片语E-R图

片语评论实体E-R图

片语评论表根据片语ID关联到片语表，每当用户发表一条评论时，该片语的评论数属性值会加1。片语评论实体包括评论ID、发表时间、内容、片语ID、评论者ID、评论者用户名等属性，片语评论实体E-R图如图4所示。

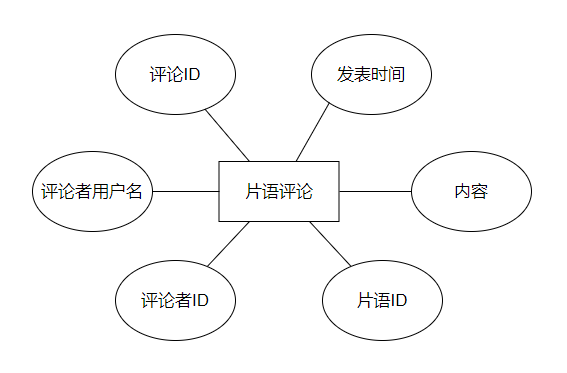


图4 片语评论E-R图

日记实体E-R图

每一篇日记产生，在数据库便生成一个日记ID，该ID为日记的主键，还根据当前时间生成发表时间字段，评论数字段计数每篇日记评论数，且用用户ID字段记录该日记是由哪个用户产生。日记表通过分类ID字段关联到日记分类表。日记实体包括日记ID、发表时间、标题、内容、评论数、用户ID和类别ID等属性，日记实体E-R图如图5所示。

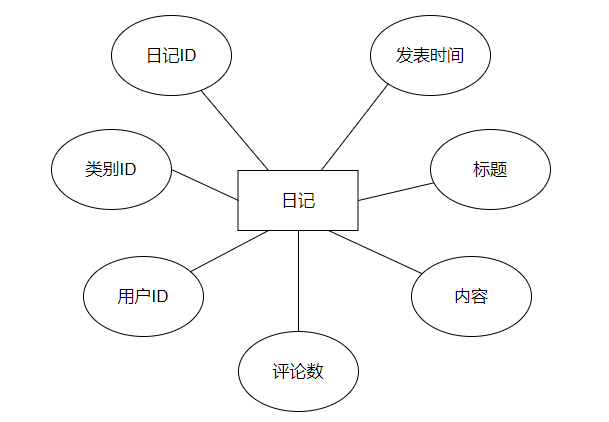


图5 日记E-R图

日记评论实体E-R图

日记评论表根据片语ID关联到片语表，每当用户发表一条评论时，该日记的评论数属性值会加1。日记评论实体包括评论ID、发表时间、内容、日记ID、评论者ID、评论者用户名等属性，日记评论实体E-R图如图6所示。

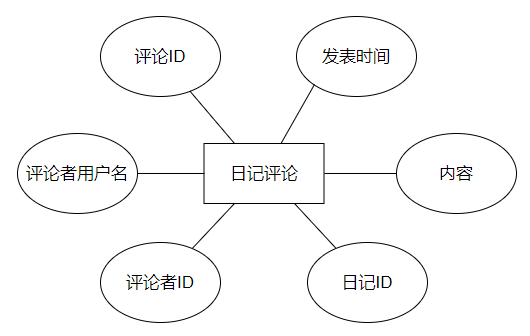


图6 日记评论E-R图

日记分类实体E-R图

每篇日记都有属于自己的类别，用用户ID属性关联到用户表，使得知道该类别属于哪个用户的分类。日记分类实体包括类别ID，类别名称，文章数和用户ID。日记评论实体E-R图如图7所示。

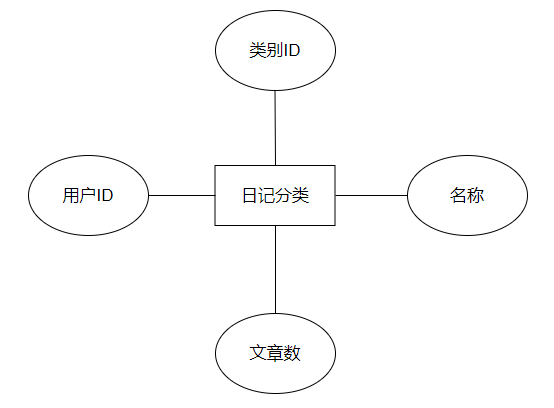


图7 日记分类E-R图

## 数据库表设计

用户信息表

用户信息表属性：用户ID、用户名、密码、邮箱、注册时间。其中用户ID为主键，注册时间为插入数据时间。表结构如表2所示。

表2 用户信息表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | | 域名 | 域名描述 | 数据类型 | 长度 | 备注 |
| 1 | ID | | 用户ID | int | 11 | 主键 |
| 3 | username | | 用户名 | varchar | 20 | default null |
| 4 | password | | 密码 | varchar | 20 | default null |
| 5 | email | | 邮箱 | varchar | 20 | default null |
| 6 | registeTime | | 注册时间 | timestamp | 0 | current\_time |

片语表

片语表属性：片语ID、发表时间、内容、评论数和用户ID，其中片语ID为主键。表结构如表3所示。

表3 记账分类信息表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | | 域名 | 域名描述 | 数据类型 | 长度 | 备注 |
| 1 | ID | | 分类ID | int | 11 | 主键 |
| 2 | uploadDate | | 发表时间 | timestamp | 0 | current\_timestamp |
| 3 | content | | 内容 | varchar | 200 | default null |
| 4 | commentNum | | 评论数 | int | 11 | default null |
| 5 | user\_id | | 子分类 | int | 11 | default null |

片语评论表

片语评论表属性：评论ID、评论时间、内容、片语ID、评论者ID和评论者名称。其中评论ID为主键。片语评论表结构如表4所示。

表4 片语评论表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | | 域名 | 域名描述 | 数据类型 | 长度 | 备注 |
| 1 | ID | | 评论ID | int | 11 | 主键 |
| 2 | postDate | | 评论时间 | timestamp | 0 | current\_timestamp |
| 3 | content | | 内容 | varchar | 200 | default null |
| 4 | mind\_id | | 片语ID | int | 11 | default null |
| 5 | critic\_id | | 评论者ID | int | 11 | default null |
| 6 | critic\_name | | 评论者名称 | varchar | 20 | default null |

日记表

日记表属性：日记表ID、发表时间、标题、内容、评论数、用户ID和类别ID。其中验证码ID为主键。日记表结构如表5所示。

表5 日记表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | | 域名 | 域名描述 | 数据类型 | 长度 | 备注 |
| 1 | ID | | 日记ID | int | 11 | 主键 |
| 2 | uploadDate | | 发表时间 | timestamp | 0 | current\_timestamp |
| 3 | title | | 标题 | varchar | 20 | default null |
| 4 | content | | 内容 | longtext | 0 | default null |
| 5 | commentNum | | 评论数 | int | 11 | default ‘0’ |
| 6 | user\_id | | 用户ID | int | 11 | default null |
| 7 | category\_id | | 类别ID | int | 11 | default null |

日记评论表

日记评论表属性：评论ID、评论时间、内容、日记ID、评论者ID和评论者名称。其中验证码ID为主键。日记评论表结构如表6所示。

表6 日记评论表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | | 域名 | 域名描述 | 数据类型 | 长度 | 备注 |
| 1 | ID | | 评论ID | int | 11 | 主键 |
| 2 | postDate | | 评论时间 | timestamp | 0 | current\_timestamp |
| 3 | content | | 内容 | varchar | 200 | default null |
| 4 | dia\_id | | 日记ID | int | 11 | default null |
| 5 | critic\_id | | 评论者ID | int | 11 | default null |
| 6 | critic\_name | | 评论者名称 | varchar | 20 | default null |

日记分类表

日记分类表属性：类别ID、类别名称、文章数和用户ID。其中类别ID为主键。日记分类表结构如表7所示。

表7 日记分类表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | | 域名 | 域名描述 | 数据类型 | 长度 | 备注 |
| 1 | ID | | 日记ID | int | 11 | 主键 |
| 2 | title | | 标题 | varchar | 20 | default null |
| 3 | diaNum | | 文章数 | int | 11 | default ‘0’ |
| 4 | user\_id | | 用户ID | int | 11 | default null |

系统设计

系统目标

开发一个精简的、易于操作的又能够满足于人们记录个人信息的个人主页，希望能够让人们因为网络科技的发达而带来生活的便利。这里没有花哨的页面展示，也不需要你花精力用各种主题样式去装扮，分散你记录的注意力，有的只是为你开辟一片敞开心扉无所畅谈、直面心灵而不受其他干扰简简单单的天地。

运用所学的知识，去开拓未尝试过的知识领域，并将所学所获理论知识付诸于实践，开发出一款在生活实际中实用的系统，为生活服务。

## 系统的功能

（1）支持用户注册成为该系统用户；

（2）支持用户登录使用该系统；

（3）支持用户更改用户名和登录密码及切换登录账号；

（4）支持用户发表片语及查看片语；

（5）支持用户发表日记及查看日记

（6）支持用户给日记进行分类

（7）支持评论功能，包括评论片语、日记以及查看评论。

## 系统开发环境

本文系统在windows10下开发，前端使用HTML+CSS+jQuery+Bootstrap架构，后端使用Python语言的web框架Flask。数据库存储数据采用MySQL5.7，服务器使用Flask内置的WSGI服务器。前后端使用json数据格式交互。开发工具使用vscode和sublime text3。

Vscode、Sublime text3简介

Vscode和sublime text3都并不是一个集成开发环境（IDE），而是一个文本编辑器。但他们都可以使用各种插件来满足开发需求，比如语法高亮、代码格式化插件，比起IDE他们轻便，对于开发小型应用完全是能够胜任开发需求的。

Flask简介

Flask是一种用Python语言编写的轻量级的web框架，它的显著特点是小，几行代码便能跑出一个web系统，核心简单且容易扩展，有内置的模板和数据库，但你完全可以随心所欲使用其他技术进行替换，因此flask较其他框架更容易上手。

MySQL简介

MySQL是一个关系型数据库管理系统。它是世界上最流行的开源数据库。尽管MySQL具有强大的功能，但它易于设置且易于使用。其性能优良, 经济成本小, 操作简便等特点, 逐步成为中小企业, 甚至一些大型互联网公司搭建自己数据库系统的首选。（李洪生 李一萌，2019）

如果想要开发一个网站或web应用程序，MySQL是一个值得强烈推荐的、一个不错的选择（Abraham Silberschatz，Henry F.Korth，S.Sudarshan，2015）。

jQuery简介

jQuery是一个js库，里面封装了许多js方法方便我们开发使用，操作简便，功能强大，其主旨内涵就是“Write Less，Do More”。使用jQuery，会比使用JavaScript简便很多。

Bootstrap简介

Bootstrap是前端的一个UI框架。Bootstrap的强大之处在于它对常见的CSS布局小组件和JavaScript插件都进行了完整且完善的封装，使得开发人员（不仅是前端开发人员）信手拈来，轻松使用（深入理解Bootstrap 徐涛，2016）

## 系统的设计

总体流程设计

总体设计流程图由个人主页系统主页界面开始。如果用户没有登录，则会自动跳转到登录页面，如果用户没进行注册过，可以通过登录界面的注册链接跳转到注册页面进行注册。

登录/注册成功后会跳转到用户个人主页的主页界面。在该界面可以查看到最新的前3条片语和日记（如果有的话）。该首页包含个人设置、切换账号登录、查看所有片语、写片语和写日记以及评论功能入口。

点击个人设置链接，即跳转到修改用户名和修改密码页面。

点击写片语按钮即进入片语页面，点击写日记按钮则进入写日记页面。写日记页面包括发表日记和添加日记分类功能。

如果用户已发布过片语或者日记，会在主页显示出来；如果片语数多于三条，只显示最新三条，其他隐藏，会有查看更多链接出现，点击查看更多会跳转到片语页面；日记同片语作处理，点击查看更多会跳转到日记页面。

片语页面包含写评论功能，和展示该用户所有的片语，每条片语下面跟主页片语一样，都会有评论和写评论按钮，可以进行查看和评论片语。

而日记页面用来展示该用户所有的日记，还包含显示日记分类，日记默认是按发表时间倒序显示，分类是按创建时间排序显示。用户可以点击任一分类，则该页面只显示该分类日记。点击日记标题或阅读全文则进入日记详情页面，日记详情页面则只显示该日记。

在显示每条片语下面都会有评论和写评论按钮，而日记则在主页和日记详情显示下面才有评论和写评论按钮。用户可以进行查看该评论和写评论。

整体架构如图8所示。

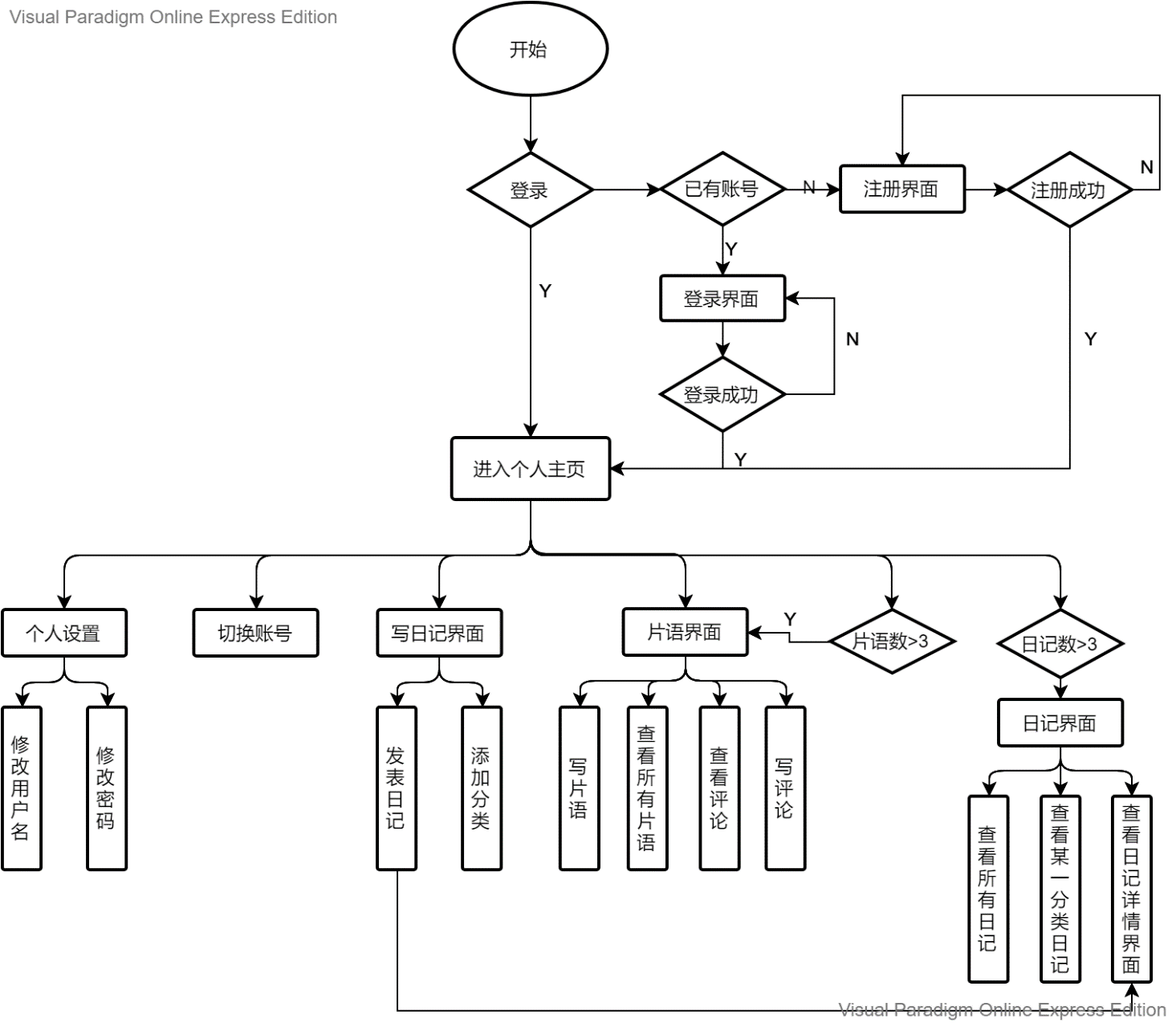


图8 总体流程图

注册流程设计

用户在注册页面输入用户名、邮箱、密码和和确认密码后点击“submit”，验证通过则注册成功。

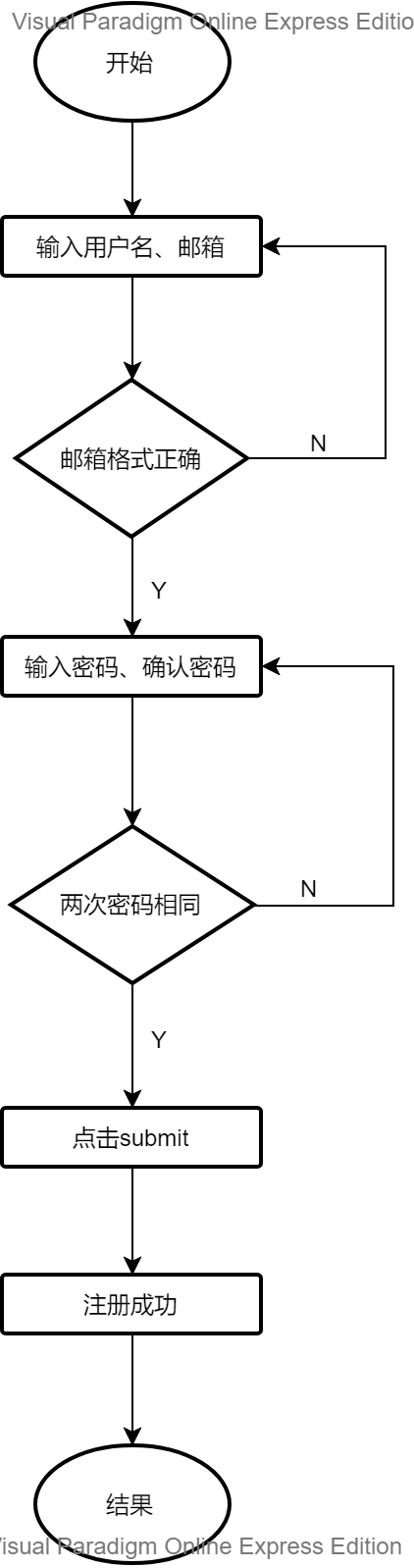


图9注册流程图

登录流程设计

用户在登录页面输入账号（注册邮箱）和密码后点击登录，账号和密码都正确则登录成功，否则登录失败。

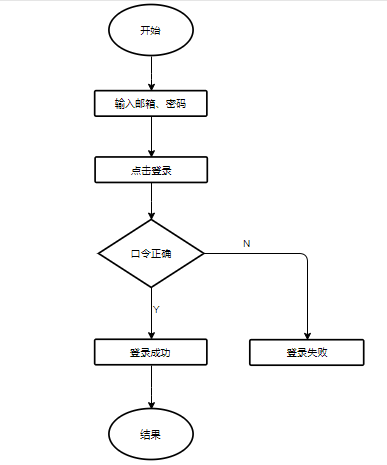


图10 登录流程图

记账流程设计

点击写日记按钮进入写日记模块，输入日记标题和日记内容后，可以选择已有的分类给文章分类，也可以新增一个分类，然后发表日记。

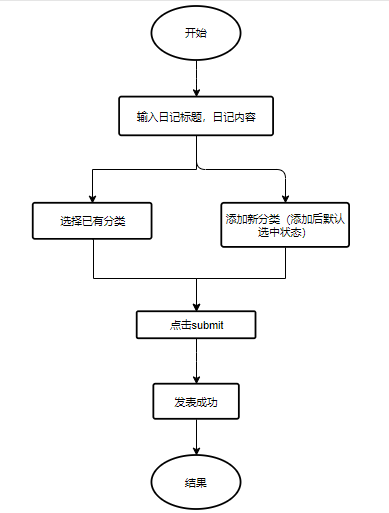


图11 发表日记流程图

总体模块的设计

本系统主要是由两个模块构成的：前端界面模块和后台逻辑模块。

（1）前端界面模块

前端界面模块实现了用户直观可操作的页面，包括了个人主页面、登录页面、注册页面、个人设置页面、片语页面、写日记页面、日记页面等。

（2）后台逻辑模块

后台逻辑模块是系统的关键，前端与数据库的交互都是要通过后台逻辑模块处理，起到相当于中央处理机制的作用。

## 前端界面模块主要设计与实现

统一样式

引进Bootstrap，使得按钮样式统一；设计系统主题色；统一hover样式和颜色，使整体保持风格统一。

//HTML文件中

//需要导进Bootstrap框架

<!-- Bootstrap 核心 CSS 文件 -->

<link rel="stylesheet" type="text/css" href="../lib/bootstrap/css/bootstrap.css">

<!-- Bootstrap 核心 JavaScript 文件 -->

<script src="../lib/bootstrap/js/bootstrap.js"></script>

//注册按钮

<button type="submit" class="btn btn-yes" id="signupSubmit">Sign up</button>

//登录按钮

<button type="submit" class="btn btn-yes" id="loginSubmit">Sign in</button>

//个人设置中确定按钮

<button type="button" class="btn btn-yes">确定</button>

//个人设置中取消按钮

<button type="button" class="btn-cancel btn btn-default col-sm-offset-7">取消</button>

//写评论按钮

<button type="submit" class="btn btn-yes" id="mindSubmit">Submit</button>

//发表日记按钮

<button type="submit" class="btn btn-yes" id="diarySubmit">Submit</button>

//CSS文件中

/\*\*

\* body:#e1f3da;

\* 主题色（用户名，签名，评论，导航）：#58d231

\* 主题色hover：#4b9e2e

\* container 背景色：#fffcf4

\*/

.btn-yes{

color: #fff;

background-color: #58d231;

border-color: #4bca22;

}

.btn-yes:hover{

color: #fff;

background-color: #4b9e2e;

}

表单校验

通过前端jQuery Validator插件对表单进行检查，能够有效地验证信息的合法性，避免了后台的无效请求，同时减少了数据库的非法操作。

//引入jQuery Validator插件

<script src="../lib/validate/jquery.validate.js"></script>

//注册表单的校验

$(function () {

$("#signupForm").validate({

rules: {

username: {

required: true,

minlength: 2},

password: {

required: true,

minlength: 5},

confirm\_pwd: {

required: true,

minlength: 5,

equalTo: "#password"},

email: {

required: true,

email: true

}

},

messages: {

username: {

required: "请输入用户名",

minlength: "用户名必需由两个字母组成"

},

password: {

required: "请输入密码",

minlength: "密码长度不能小于 5 个字母"

},

confirm\_pwd: {

required: "请输入密码",

minlength: "密码长度不能小于 5 个字母",

equalTo: "两次密码输入不一致"

},

email: "请输入一个正确的邮箱",

}

});

});

时间格式化

该系统很多地方都需要展示时间，例如片语日记的发表时间，以及评论的发表时间。从后台接收到的时间格式类似于Thu Aug 18 20:38:54 CST 2016，并不符合中国人的阅读习惯，因此便在前端封装了个函数处理时间格式化，转换为2016-08-19 10:38格式。

// 日期时间格式化

function dateFormat(date, format) {

date = new Date(date);

date.setHours(date.getHours());

var o = {

'M+': date.getMonth() + 1, //month

'd+': date.getDate(), //day

'H+': date.getHours(), //hour

'm+': date.getMinutes(), //minute

's+': date.getSeconds(), //second

'q+': Math.floor((date.getMonth() + 3) / 3), //quarter

'S': date.getMilliseconds() //millisecond

};

if (/(y+)/.test(format))

format = format.replace(RegExp.$1, (date.getFullYear() + '').substr(4 - RegExp.$1.length));

for (var k in o)

if (new RegExp('(' + k + ')').test(format))

format = format.replace(RegExp.$1, RegExp.$1.length == 1 ? o[k] : ('00' + o[k]).substr(('' + o[k]).length));

return format;

}

//调用

postDate = dateFormat(postDate, 'yyyy-MM-dd HH:mm');

写评论接口处理

由于本系统多处涉及评论，因此将评论封装成一个function，此处摘取写评论接口代码。该接口所需要判断是片语的还是日记的评论，然后通过点击写评论按钮获得被评论的ID。通过ajax与后台接口交互数据，评论成功后动态显示评论数+1。

$(".write-comment").click(function (e) {

e.preventDefault();

var parentId = $(this).parent().parent().attr('id');

arr = parentId.split('\_');

Id = arr[1];

if (arr[0] == 'diary') {

isDiary = true;

mindOrDiary = 'diary'

} else {

isDiary = false;

mindOrDiary = 'mind'

}

})

$("#saveComment").click(function (e) {

e.preventDefault();

var textArea = $("#txt").val()

$(this).attr('data-dismiss', 'modal')

$.ajax({

type: 'post',

datatype: 'json',

url: host + mindOrDiary + '/write\_comment',

xhrFields: {withCredentials: true},

crossDomain: true,

data: { Id: Id, content: textArea },

success: function (data) {

if (data) {

if (data.errorcode == 0) {

alert('评论成功');

var selector = $('#' + mindOrDiary + '\_' + Id + ' .get-comments');

var commentNum = selector.val().split(/[()]/)[1];

commentNum++;

selector.val("已评论(" + commentNum + ")")

} else {alert('评论失败'); }

}

}

})

})

## 后台逻辑模块主要设计与实现

### 处理请求

后台用视图函数统一接受、处理前端的请求以及将结果返回前端页面，在url.py中设置视图函数相关的URL，即是不同的请求对应不同的处理函数。当前端发出一个请求时，系统会根据请求路径发送数据到相应的视图函数进行处理。

def login(request): //登录页面的后台处理代码

if request.method == 'GET':

return render\_to\_response('login.html')

state = None

if request.method == 'POST':

email = request.POST.get('email', '')

password = request.POST.get('password', '')

user = User.objects.filter(email=email, password=password)

if user:

response = HttpResponseRedirect(reverse('home'))

request.session['email'] = email

return response

else:

state = 'not\_exist\_or\_password\_error'

content = {

'active\_menu': 'index',

'state': state,

'user': None

}

return render(request, 'login.html', content)

### 发送邮箱验证信息

在注册和找回密码的时候需要进行邮箱验证，系统生成验证码发送到用户提供的邮箱，用户输入验证码，后台校验成功才能成功注册和找回密码。

//给邮箱发送验证码

code\_list = []

for i in range(10): # 0-9数字

code\_list.append(str(i))

for i in range(65, 91): # A-Z

code\_list.append(chr(i))

for i in range(97, 123): # a-z

code\_list.append(chr(i))

myslice = random.sample(code\_list, 6) # 从list中随机获取6个元素，作为一个片断返回

verification\_code = ''.join(myslice) # list to string

new\_email = Verification\_code.objects.create(email=email, code=verification\_code, action='1')

new\_email.save()

email\_title = "记账本验证码"

email\_body = "您正在找回记账本密码，这是您的验证码：{0}".format(verification\_code) # 发送邮件

status = send\_mail(email\_title, email\_body, EMAIL\_FROM, [email])

### 分页显示

从数据库中提取数据，设置每页显示的数量，分页计算页面数量，再返回到前端分析显示（齐金刚，李滔，李晋军，2014）。

//记账列表分页显示

after\_range\_num = 5 # 当前页前显示5页

befor\_range\_num = 4 # 当前页后显示4页

try: # 如果请求的页码少于1或者类型错误，则跳转到第1页

page = int(request.POST.get("page", 1))

if page < 1:

page = 1

except ValueError:

page = 1

paginator = Paginator(accounts, 10) # 设置books在每页显示的数量

try: # 跳转到请求页面，如果该页不存在或者超过则跳转到尾页

accounts\_list = paginator.page(page)

except(EmptyPage, InvalidPage, PageNotAnInteger):

accounts\_list = paginator.page(paginator.num\_pages)

if page >= after\_range\_num:

page\_range = list(paginator.page\_range)[page - after\_range\_num:page + befor\_range\_num]

else:

page\_range = list(paginator.page\_range)[0:int(page) + befor\_range\_num]

template = 'searchaccounting.html'

return render(request, template, {"accounts": accounts,"page\_range": page\_range})

### 拦截器

用户需要登录了才能使用该系统，未登录的用户将被拦截到登录页面。

//拦截未登录的用户

from django.shortcuts import HttpResponseRedirect

from django.shortcuts import render\_to\_response

try:

from django.utils.deprecation import MiddlewareMixin # Django 1.10.x

except ImportError:

MiddlewareMixin = object # Django 1.4.x - Django 1.9.x

class LoginMiddleware(MiddlewareMixin):

def process\_request(self, request):

if not request.path.startswith(('/online/login', '/online/registered','/online/retrievepassword', '/online/send\_retrievepassword\_verification\_code', '/online/send\_registered\_verification\_code')):

if request.session.get('email', None):

pass

else:

return HttpResponseRedirect('/online/login')

### 数据库交互

后台直接使用Django封装好的函数来进行数据库操作，只需要简单的配置就可以使用MySQL数据库，创建或修改类只需要同步数据库就能修改对应的表设计。

//配置MySQL数据库

DATABASES = {

'default': {

'ENGINE': 'django.db.backends.mysql',

'NAME': 'accountingsystem',

'USER': 'root',

'PASSWORD': '\*\*\*\*\*\*',

'HOST': '127.0.0.1',

'PORT': '3306',

}

}

//创建类，同步数据库即可生成对应的表

class Verification\_code(models.Model):

id = models.AutoField(primary\_key=True)

email = models.CharField(max\_length=30)

create\_date = models.DateTimeField(auto\_now\_add=True)

code = models.CharField(max\_length=6)

action = models.CharField(max\_length=6)

//数据库操作

//查询操作

Verification\_code.objects.filter(email=email, action='0').values("code").order\_by('-create\_date')[:1][0]['code']:

//添加操作

User.objects.create(email=email, username=username, password=password, )

//更新操作

User.objects.filter(id=userid).update(password=password)

系统运行效果及功能说明

## 注册

（1）注册界面



图10 注册界面图

（2）注册功能说明

如图10所示，用户根据提示填写个人信息，当信息不符合要求时，将无法点击注册按钮；注册过的邮箱无法重复注册，两次输入的密码不相同也无法进行注册，必须保持一致。

## 登录

（1）登录界面



图11登录界面图

（2）登录功能说明

如图11所示，用户输入账号和密码登录，没有注册的用户可以点击“注册账号”进行注册，忘记密码则可以点击“忘记没密码”找回密码。登录成功将跳转至系统首页。

## 找回密码

（1）找回密码界面



图12找回密码界面图

（2）找回密码功能说明

如图12所示，用户输入注册的邮箱，输入密码和确认密码，点击“发送验证码”，系统将验证码发送到邮箱中，用户输入验证码后点击“找回密码”，找回密码成功则跳转至系统首页。找回密码成功的条件时按照要求输入相关信息。

## 修改密码

（1）修改密码界面



图13 修改密码界面图

（2）修改密码功能说明

如图13所示，用户要登录才能修改密码，输入的旧密码要正确，两次输入的新密码符合要求并且保持一致才能修改成功。

## 修改个人资料

（1）修改个人资料界面



图14 修改个人信息界面图

（2）修改个人资料功能说明

如图14所示，用户个人资料中的邮箱是不能修改的，其他信息可以在输入框中编辑完后点击“保存”即可修改个人资料。

## 分类设置

（1）分类设置界面



图15 分类设置界面图

（2）分类设置功能说明

如图15所示，用户可以选择添加收入、支出、借入、借出、还款、收债六种分类，点击相应的分类弹出输入框输入分类名称点击添加即可，不能重复添加相同的分类。点击具体分类后面的编辑图标可对该分类进行编辑，点击删除图标可删除该分类。

## 记账

（1）记账界面



图16 记账界面图

（2）记账功能说明

如图16所示，用户登录系统，点击导航菜单栏的“记账”进入记账功能界面。用户可以选择收入、支出、借入、借出、还款、收债六种分分类分别进行记账，输入日期、金额、账户、备注，选择分类点击“保存”即可添加一条记账记录。其中日期默认选择当前日期，同时用户也可以选择一个自己想要的日期；金额显示用户只能输入数字，不能输入其他字符；分类只能选择该大分类下面的子分类，不支持手动输入其他的分类，如果想要增加分类，可以先通过分类设置功能设置；账户是指用户此次使用的收入或支出的金融账户，比如支付宝、信用卡等。

## 账单管理

（1）账单管理界面

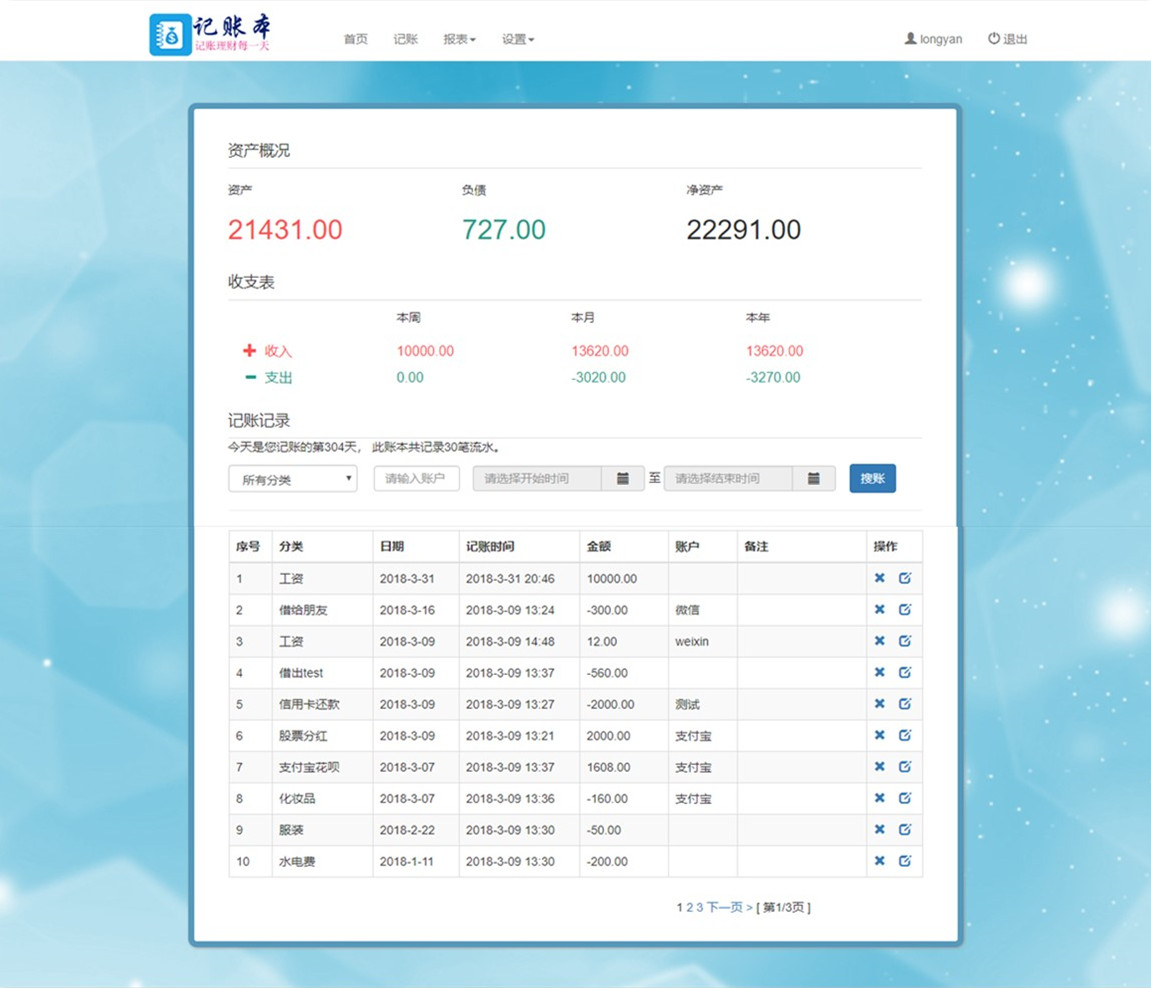


图17 账单管理界面图

（2）订单管理功能说明

如图17所示，显示当前用户的所有账单记录，可以进行多条件查询、修改删除账单操作。

## 收支分析

（1）收支分析图界面



图18 收支分析界面图

（2）收支分析功能说明

如图18所示，用户可以选择一个时间段，查看这段时间内的收支图表，对该段时间内的财务行为一目了然。

## 对账表

（1）对账表界面



图19 对账表界面图

（2）对账表功能说明

如图19所示，对账表包括日对账表和月对账表，直观显示每天、每月的财务流入流出的情况。

结论与展望

本论文介绍了课题的选题背景、需求分析、数据库设计到系统设计。该系统后台部分使用了Django框架，将程序划分为视图、模板、模型三部分，提高了业务处理效率。前端部分引进bootstrap、jQuery框架，使用了html、css、javascript技术可以轻易实现页面效果。

当然，本文系统还有很多地方可以改进：

（1）本文系统功能相对单薄，没有预算设置、超额提醒等功能，可以增加一些功能让系统更加满足用户使用。

（2）代码复用率很低，很多相似功能的代码无法做到复用。

（3）本人系统的收支分析统计的维度相对较少，可以增加一些维度和多种分析图来让用户更直观地了解自己日常地财务行为。

（4）前端html代码兼容性做得不够好，当屏幕大小变化了，页面不能做到完全自适应。

参 考 文 献

陈颂丽, 李海良. 基于MVC模式的个人财务管理系统的设计与实现[J]. 科技信息. 2010, 2:2-4

单东林, 张晓菲, 魏然. 锋利的jQuery（第2版）. 人民邮电出版社. 2015, 1:15-30

钱乐秋, 赵文耘, 朱俊珏. 软件工程. 清华大学出版社. 2014, 5:56-74

齐金刚, 李滔, 李晋军. Django框架Web数据查询分页技术研究[J]. 电子设计工程. 2014, 3:8-15

孙智勇, 面向高收入群体财务管理系统建设及应用研究[D]. 湖南大学. 2015, 10:2-3

王冉阳. 基于Django和Python的Web开发[J]. 电脑编程技巧与维护. 2009,1:45-50

徐涛. Web开发技术丛书 深入理解Bootstrap. 机械工业出版社, 2014, 2:156-157

Abraham Silberschatz, Henry F.Korth, S.Sudarshan. 数据库系统概念. 机械工业出版社. 2015, 10:30-35

Adam Freeman. HTML5权威指南. 人民邮电出版社.2014,8:20-130

Baron Scbwartz, Peter Zaitsev, vadimTkacbenko. 高性能MySQL. 电子工业出版社. 2013, 5:170-230

David Cochran, Ian Whitley.Bootstrap实战. 人民邮电出版社. 2015, 9:1-189

Eric Matthes. Python编程从入门到实践. 人民邮电出版社. 2016, 9:1-230

Jeff Forcier, Paul Bissex, Wesley Chun. Django Web开发指南. 机械工业出版社. 2009, 7:230-235

Jeremy Keith. JavaScript DOM 编程艺术（第2版）. 人民邮电出版社. 2011, 1:48-73

Wesley Chun. Python核心编程（第3版）. 人民邮电出版社. 2016, 5:60-75

致 谢

就要毕业告别学校了，在这里我首先要感谢我的指导老师林旭东老师，是他给予我帮助和指导，让我在规定地时间内完成了毕业设计。我觉得他是一名很认真负责地好老师，谢谢他。

其次，我要感谢学校这四年来地栽培，让我学到了很多专业知识，学会了如何做人做事。

接着，我要感谢所有遇到地同学、老师，是他们陪伴我走过了大学四年的美好时光。

然后，我要感谢我的家人，是他们给予我无限地爱和支持。

最后，非常感谢百忙之中抽出珍贵时间来参加论文审阅和答辩的各位老师和评委。谢谢大家。

**华南农业大学**

**本科生毕业论文成绩评定表**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 学号 | 201430340117 | | | | | 姓名 | | 龙燕 | 专业 | 软件工程 | | |
| 毕业论文题目 | | | 基于Django与bootstrap个人收支管理系统 | | | | | | | | | |
| 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 | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 答  辩小组  评语及  成  绩  评  定 | 评价项目 | 具体要求（A级标准） | 最高分 | 评分 | | | | |
| A | B | C | D | E |
| 论文质量 | 论文结构严谨，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 |
|  |  |  |  |  |
| 答辩小组评语  是否同意通过论文答辩（打√）   1. 同意 2. 不同意   成绩（百分制）：　　　　　　　　 答辩小组成员（签名）：  　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　 年 月 日 | | | | | | | |
| 成绩总评 | 论文总评分数：  教学院长签名：  学院盖章：  年 月 日 | | | | | | | |

续上表：