****

**毕 业 设 计**

**题 目: 基于Django框架的博客的系统设计与实现**

**英文题目: Design and implementation of blog system based on Django framework**

|  |  |
| --- | --- |
| **学生姓名：** | **陈世超** |
| **学位门类：** | **工学** |
| **学 号：** | **2019213637** |
| **专 业：** | **物联网工程** |
| **学 院：** | **软件学院** |
| **指导教师：** | **杨巧合** |
| **教师职称：** | **讲师** |
| **完成时间：** | **2023年05月10日** |

****

**Graduation Project**

**Design and implementation of blog system based on Django framework**

|  |  |
| --- | --- |
| **Student Name：** | **Chen ShiChao** |
| **Degree Type：** | **Bachelor of Engineering** |
| **Student No：** | **2019213637** |
| **Major Filed：** | **Internet of Things Engineering** |
| **Faculty：** | **Software College** |
| **Thesis Advisor：** | **Yang Qiaohe** |
| **Professional Title：** | **Lecturer** |
| **Complete Time：** | **2023年05月10日** |

**作 者 声 明**

本人以信誉郑重声明：所呈交的学位毕业设计（论文），是本人在杨巧合老师的指导下，由本人独立完成的。没有剽窃、抄袭、造假等违反道德、学术规范和其他侵权行为。文中引用他人的文献、数据、图件、资料均已明确标注出，不包含他人成果及为获得东华理工大学或其他教育机构的学位或证书而使用过的材料。对本设计（论文）的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本毕业设计（论文）引起的法律结果完全由本人承担。

本毕业设计（论文）成果归东华理工大学所有。

特此声明。

毕业设计（论文）作者（签字）：

签字日期：202 年05 月30 日

本人声明：该学位论文是本人指导学生完成的研究成果，已经审阅过论文的全部内容，并能够保证题目、关键词、摘要部分中英文内容的一致性和准确性。

学位论文指导教师签名：

年 月 日

摘 要

近年来，随着互联网技术的高速发展，信息获取方式和交流方式正在发生着改变，网络已经成为了获取信息和交流的主要渠道。网络博客是其中一个很重要的分支。本课题旨在设计并实现一个符合当代人习惯，高效易用，功能丰富的博客系统。本课题从用户和管理者两个角度出发设计各个模块，实现了从登录，注册到发布博文、评论等实用功能，为有相关需求的用户提供了一个高效简便的博客平台。

本课题采用MVT(即模型-Model，视图-View，模板-Template)设计模式和B/S(Browser/Server，即浏览器/服务器)应用架构。采用Visual Studio Code编辑器进行开发。利用Git实现版本控制；后端语言采用Python语言进行开发，前端使用Html、Css、JavaScript、Bulma等技术，采用关系数据库Mysql，基于Django框架进行系统的开发。本系统前台面向学生，实现了发布文章、查看文章、查看个人信息、修改个人信息、评论文章等实用功能；Django框架默认为开发者提供了一个后台管理系统，供拥有权限的用户使用。该后台集成了对各个模型的增删改查功能，十分简洁高效，实现了文章管理、评论管理、用户管理、标签管理、分类管理以及日志记录等功能。博客系统前后端的结合满足了用户在大部分场景下使用本系统的业务需求。

本课题所设计和实现的线上考试系统目前已经在阿里云上线部署，经过对各个功能模块的测试，已验证本系统的实现和预期结果一致。

**关键词：** 博客系统； Python； MySQL；Django； Bulma

Abstract

In recent years, with the rapid development of Internet technology, the way of information acquisition and communication is changing, and the network has become the main channel of information acquisition and communication. The web blog is a very important branch. The purpose of this project is to design and implement a blog system which is in line with contemporary people's habits, efficient and easy to use, and rich in functions. This topic from the user and manager two points of view to design each module, the realization of the login, registration to post blog, comment and other practical functions, for users with related needs to provide an efficient and simple blog platform.

This topic uses MVT(namely model-model, view-view, template-template) design pattern and B/S(Browser/Server) application architecture. Using Visual Studio Code editor for development. Use Git to achieve version control; The back-end language uses Python language for development, and the front-end uses Html, Css, JavaScript, Bulma and other technologies. The relational database Mysql is used for system development based on Django framework. The foreground of the system for students, the realization of the release of articles, view articles, view personal information, modify personal information, comment articles and other practical functions; By default, the Django framework provides a backend management system for developers to use by authorized users. The background integrates the function of adding, deleting, modifying and checking each model, which is very simple and efficient, and realizes the functions of article management, comment management, user management, label management, classification management and log recording. The combination of the front and back ends of the blog system meets the business needs of users in most scenarios.

The online examination system designed and implemented by this subject has been deployed online in Alibaba Cloud. Through the testing of various functional modules, it has been verified that the implementation of this system is consistent with the expected results.

**Keywords:** Blog System； Python； MySQL；Django； Bulma

# 第一章 绪 论

## 1.1 课题背景、目的及意义

博客是在Email、BBS、ICQ之后出现的新型网络交流方式，在当下的网络时代，发表博文类似于以前在杂志和报纸等传统纸质媒体上发表文章。博客相对于传统媒体，其方便性不言而喻。博客一经出现便快速发展，如今，不只是互联网工作者需要使用博客来记录或学习专业知识，在互联网领域之外的人士也开始使用博客来分享日常生活。

目前互联网上的博客系统尽管种类繁多，但是安全性无法保障，有些博客系统功能过于臃肿，使用体验不佳。本课题设计的博客系统，以简洁高效为宗旨，进而开发出一个迅捷易用，简洁高效、适用于大多数人群、满足大部分场景需求的的博客系统。本课题设计的博客系统可以保障安全性，有个人博客需求的用户通过本系统可以在网络中发表文章，记录学习或者生活。多个用户可以在本博客系统中交流观点，发表评论。

博客系统目的是在互联网上提供一个供人们交流沟通的平台，尤其是在网络已经成为人们沟通的一种主流方式的当下，一个简洁易用，安全高效的个人博客系统的开发显得更为重要。

## 1.2 可行性分析

程序系统设计和开发之前的可行性分析可以帮助开发人员确定整体的执行方案，明确系统开发的目的和要求，保证系统的设计与实现过程是必要且有意义的。下面从技术可行性、经济可行性、社会可行性三个方面进行分析。以确保系统开发的顺利进行。

1. 技术可行性：本系统基于Django框架进行开发，后端使用Python语言进行开发，Python具有种类丰富的开源库，利用这些库可以较为方便的进行系统开发。前端页面使用Html、Css、JavaScript技术和Bulma框架进行编写。上述技术和框架都是安全可靠的。这些技术有助于安全快速的开发博客系统，同时保证了本系统在技术层面上的可行性。
2. 操作可行性：对用户层面而言，他们可以在博客系统中发表文章、对文章进行评论，这些操作都是在网页中进行，所以用户只需要通过浏览器访问网站，在简单的图形化界面中进行操作，这是相当方便且高效的。而管理员则可以通过Django提供的后台管理界面进行用户管理，用户权限管理等敏感操作。故本系统在操作上具有可行性。
3. 社会可行性：随着互联网技术的不断更新迭代，网络技术已经渐趋成熟，与人们的联系也越来越紧密。而博客也已经成为人们在网络上进行沟通交流的主要渠道之一。本系统致力于构建一个简洁高效，安全易用的博客系统。故本系统具有较好的社会可行性。

## 1.3 论文结构概述

本课题围绕一个博客系统的设计与实现进行，本文的正文主体部分共分为8个章节，每章节的大致内容如下：

1. 第一章 绪论：此章节主要介绍本课题（论文）的研究背景、目的和意义，分析本课题的可行性，并简要说明本论文的章节组织结构。
2. 第二章 核心技术介绍：本章节对博客系统中用到的一些核心技术进行(如程序设计语言Python，框架Django，前端Css组件库Bulma以及开发工具VS Code)详细的说明。
3. 第三章：系统需求分析：此章节主要包括三个方面：总体需求概述、功能性需求分析和非功能性需求分析。本章分别从这三个方面分析了本课题的设计和实现要求，并与相应角色的用例图相结合进行描述。
4. 第四章 系统设计：本章结合了系统数据库的设计，从设计原则和系统结构上考虑对每个模块进行了整体和详细的设计，并对系统中每个模块的一些相关结构图、模型图、数据库E-R图、数据库表结构和整体流程图的设计部分进行了说明。
5. 第五章 系统实现：本章节结合系统界面的截图和部分代码展示，对一些主要功能的设计与实现进行了说明
6. 第六章 系统测试：本章节介绍了本系统使用的测试方法，并就测试结果进行展示和说明。
7. 第七章 系统部署：本章节对项目的部署进行了详细介绍，对项目部署在云服务器中的各个步骤作简要说明。
8. 第八章 总结与展望：本章节对博客系统进行了一个全方位的总结，就不足之处进行反思，并希冀在未来得以完善并修改本博客系统。

## 1.4 本章小结

本章节对本论文的背景、目的和意义进行了介绍，并从技术、操作、社会三个方面进行了可行性的分析，并简要介绍了本论文的章节组织结构。

# 第二章 核心技术介绍

## 2.1 程序设计语言Python

Python语言是一门面对对象的高级程序设计语言，Python采用极简主义的设计理念，加以统一规范的交互模式，这使Python具有极简的代码风格，易于学习，理解和记忆。Python另一个特性是可扩展，Python没有把所有功能和特性都集成到语言核心，相反Python提供了丰富的API，以便程序员根据自身需求调用。Python还拥有大量的第三方库，这在程序开发中可以大大加快开发速度，在第三方库的基础上，可以减少很多复杂的操作，使开发人员集中精力处理业务逻辑。

## 2.2 关系数据库Mysql

MySQL是一种关系型数据库，适用于中小型系统。因其拥有开源、速度较快、规模较小、性能较高等优点，在中小型网站系统开发中，开发人员往往采用MySQL数据库。本博客系统规模适中，开发成本有限，故采用了MySQL数据库，MySQL数据库可以满足本系统所有要求，性能表现较好。

## 2.3 开发环境和工具

### 2.3.1 Visual Studio Code编辑器

Visual Studio Code是微软公司开发的一款跨平台的编辑器，能够在Windows、Linux、MacOS等平台上运行。通过安装一些插件可以使其拥有代码提示，语法高亮等功能，因此可以用来代替集成开发环境。考虑到VS Code的轻量、插件丰富、开发简单、易于使用等优点，本课题采用VS Code来进行开发。

### 2.3.2 Git版本控制

Git是目前最为优秀的分布式版本控制系统之一，任一个项目的开发都可以使用Git来进行版本控制，其分布式的特点使其特别适合多人协作的系统开发，每一个开发人员可以将代码先提交到本地，在通过网络将代码推出到中央仓库。Git还可以管理整个项目的所有版本，在团队开发中可以方便的整合多个开发者的代码，在代码出现问题时可以选择回滚项目，避免因为错误提交影响整个项目的运行。

## 2.4 前端页面核心技术

### 2.4.1 Bulma框架

Bulma是一个免费的、开源的CSS框架，其完全使用CSS进行编写，所以是完全静态的。Bulma采用移动优先的响应式布局，内置了多个可以随时使用的前端组件，使用者只要赋予Html标签对应的类名，便可以使用组件，使用Bulma框架可减少大量的前端样式编写工作，大大提升了开发效率。本系统采用Bulma框架，构建了一个响应性强，美观简洁的web界面。

### 2.4.2 JQuery

JQuery是一个JavaScript的库，利用JQuery进行网页开发可以简化使用JavaScript进行开发的复杂性，JQuery提供对一些常见任务的的自动化，并简化了一些复杂任务。本系统使用Jquery技术，不仅可以将原本需要很多JavaScript代码才能实现的功能缩减为几行简单明了的代码，而且性能不会受到很大影响。故本系统选择使用Jquery。

## 2.5 Web开发框架Django

Django是一个基于Python语言编写的的重量级Web应用框架。Django奉行“包含一切”的设计哲学，即是：Web应用所需的通用功能都应包含在框架内部，例如身份验证、URL路由、模板系统、对象关系映射、数据库的迁移等功能。这些功能都已集成在Django框架内部，具体模块的内部实现细节被封闭，使开发者可以集中精力关注自己的代码。内部集成多种模块功能虽然使Django框架显得相当笨重，但在构建网站时可以获得极高的效率。

Django遵循MTV的设计模式，M（Model）即模型层，负责创建数据的模型，应执行数据库的迁移；T（Template）即模板层，模板层即是展示给用户的界面，模板层可以接受从视图中传来的数据并展现在前台界面；V（View）即视图层，视图负责业务逻辑的处理，是最关键的一层，视图层向内可以连接模型层，间接的访问数据库，对外可以渲染模板，向模板传递从数据库中取得的数据或是从页面中取得数据写入数据库。

此外，Django拥有一个自带的管理后台，拥有超级管理员权限的用户账号可以登录该后台，开发者几乎可以不用写一行代码就可以获得一个功能完备的后台管理界面，并且该管理界面支持自定义，用户可以自己选定该后台可以管理的模型。Django同时也有清晰完备的文档，这使得在开发过程中遇到的疑难问题大都可以在官网寻求到解决方案。

## 2.6 本章小结

本章对博客系统中使用的核心技术进行了介绍，分别从程序设计语言、数据库、开发工具和环境、前端核心技术、后端框架五个方面进行了说明。这五个方面涵盖了本博客系统的大部分的技术点。

# 第三章 系统需求分析

系统需求分析是系统是在正式进行开发之前的关键步骤之一，将系统的需求和功能进行进一步的梳理，不仅将有利于开发人员进行更好的系统设计，而且可以使最终系统模块更加符合用户的要求，进而给用户提供一种良好的用户使用体验。

## 3.1 需求分析总体描述

系统需求分析的具体工作包括基本的功能性需求分析和另外的非功能性需求分析。功能性需求分析需要分析系统中的功能模块，列出核心的模块和基本的功能，因其对整个系统的重要性极高，对这部分模块必须完善其内部的各种细节。非功能性需求分析主要分析对象是在核心模块之外的辅助性功能，这些功能的完善与否不会对整个系统造成决定性的影响，但可以大大提升用户的体验。这一个部分常常由用户和开发者进行沟通或是从用户角度考虑以开发对应功能。

对于本博客系统，基本的功能性需求分析需要从博客系统的核心考虑，包括用户登录、用户注册、发表文章、修改文章发表评论、找回密码等在内的核心功能都应纳入功能性需求分析之中。 而对于其他的非功能性需求，都是在功能性需求之外进一步提升用户体验。

## 3.2 功能性需求分析

本系统提供的核心功能主要包括以下几点

1. 普通用户注册账号并登录系统；
2. 普通用户账号密码忘记后可以通过发送邮件进行密码找回；
3. 普通用户登录后可以进入博客系统浏览所有文章；
4. 普通用户可以在发表文章，发表文章时并支持富文本编辑器和Markdown语法；
5. 普通用户可以进入个人中心修改个人信息；
6. 普通用户可以在文章详情界面发表评论；
7. 管理员用户后台管理用户，文章，评论，分类，标签等模型；
8. 管理员后台查看系统日志信息。

经以上基本功能性需求分析可知，本博客平台系统的用户主要有两类：普通用户、管理员用户。以下是对两种用户角色的权限和功能的需求分析。

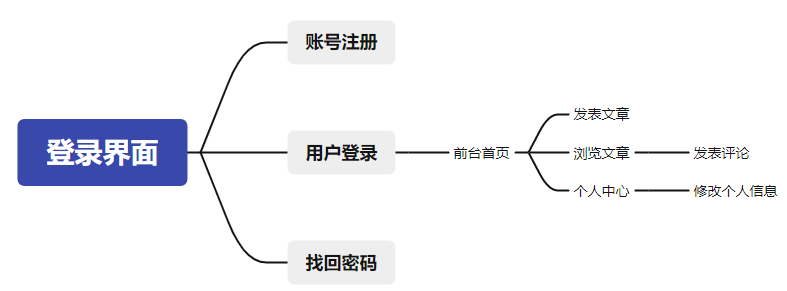


图3-1 学生用户功能性需求用例分析

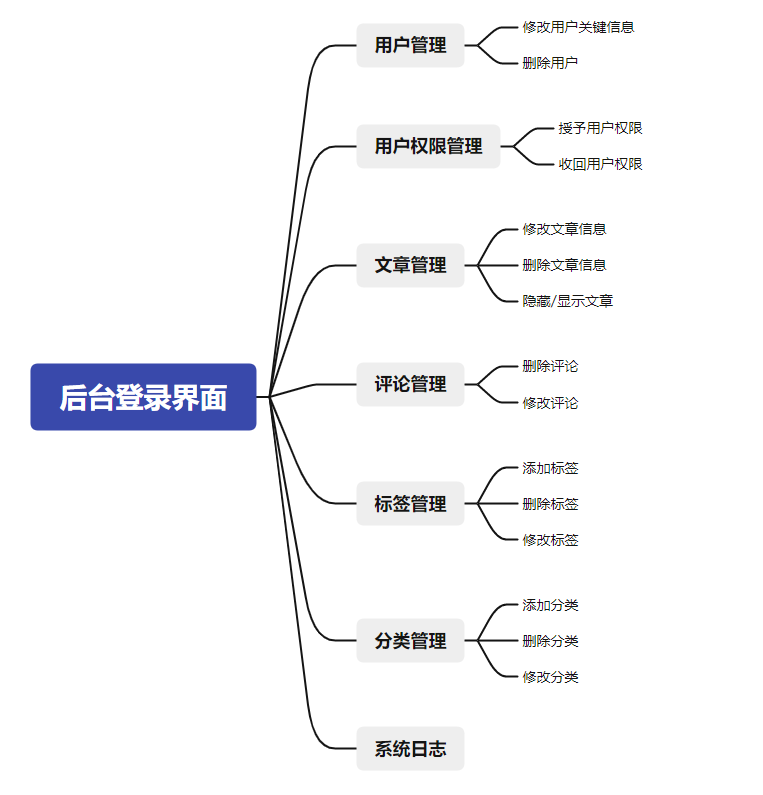


图3-2 管理员用户功能性需求用例分析

## 3.3 非功能性需求分析

本系统的非功能性需求分析主要从以下四个角度进行分析：

1. 系统性能分析：

为避免系统响应速度过慢，开发程序时应尽量考虑效率高的代码，在某些场景下响应页面同时还要兼顾数据的实时更新，故应在不影响用户体验和数据传输速率的前提下减少系统请求。

1. 系统健壮性分析：

在某些情况下，视图层处理逻辑时可能会陷入无限循环或是其他逻辑上的错误，这些都是系统的健壮性分析需要考虑的问题。因此在程序开发中，应作完善的考虑，对异常进行捕获，循环中设置循环上限等都可以使得系统变得更加健壮。

1. 系统并发性分析：

博客系统的本意就是发表文章供人浏览学习，所以系统开发设计之初就应考虑到系统的高并发场景下的性能。提升系统应对高并发的能力，可以避免系统因为请求数过多宕机而造成经济损失。

## 3.4 本章小结

本章主要介绍了系统的需求分析，介绍了系统需求分析的重要性及其在系统开发过程中的作用，阐述了博客系统的核心功能。在系统需求分析中，从普通用户和管理员两种用户的角度分别出发，借由对应的功能模块分析图阐明了本系统的基本功能。并从系统性能、系统健壮性和系统并发性等方面分析了本系统的一些非功能性需求。

# 第四章 系统设计

## 4.1 系统设计原则

系统设计是软件系统开发周期中至关重要的一个环节，贯穿了需求分析和系统实现两大部分，是将理论转化为实际的关键过程。因此，在对本线上考试系统进行设计时，为充分考虑系统逻辑和功能需求以及系统实现的可行性，本系统的设计遵循以下几个设计原则。

1. 系统以用户为中心原则:本系统是一个博客系统，必须始终围绕普通用户的核心功能需求进行设计。在系统开发时不可违背这个核心功能需求进行开发，其他的辅助性功能设计不能违反这一前提。
2. 系统安全稳定原则：安全性对一个面对大众的博客系统十分重要，用户私人数据的泄露将会带来巨大的负面影响。稳定性对用户的体验感也至关重要。所以本系统设计时应充分考虑采取必要措施增强系统的稳定性和安全型，如选择较为安全的MySQL数据库，使用加密数据传输方式，选用安全的服务器厂商。
3. 系统可移植性原则：系统打开并且打包完成后，用户会在各种不同的运行环境下使用本系统，所以应在测试环节充分测试在主流操作系统（Windows、MacOS、Linux等）和浏览器（Google Chrome，Microsoft Edge、Safari、Firefox）中的运行状态，保证系统在上线后能被用户正常访问。
4. 系统可扩展原则：系统在设计完成并上线运行后，可能会进行版本的迭代和维护，所以在设计时应考虑系统将来的拓展性，在开发中为之后的升级预留外部接口，应对未来功能的扩展和增加。
5. 系统高效率原则：系统的响应时间越短，用户的体验越好，所以考虑系统的运行效率对用户来讲也十分重要。编写高效率的代码同时也是提升稳定性的一种方式。

## 4.2 系统架构设计

本博客系统整体采用B/S(Browser/Server，即浏览器/服务器)系统架构和MTV(Model-Template-View，即模型-模板-视图)三层设计模式进行架构设计和开发。下面对B/S架构和MTV设计模式进行介绍：

1. B/S架构：一种浏览器和服务器架构模式，用户通过浏览器发出请求，服务器收到请求后会返回对应的资源，经浏览器渲染后展示给用户。 B/S架构如图4-1所示：

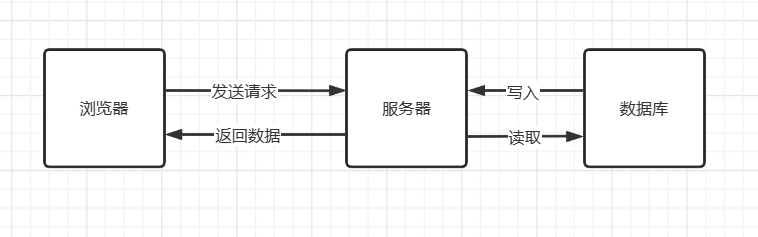


图4-1 B/S架构图

1. MTV三层设计模式：Django采用了MTV的框架模式，即模型（Model）、模板（Template）和视图（Views），三者分别负责项目的不同职责。

* 模型层：数据储存层，负责处理与数据有关的所有事务，例如如何在数据库中存储和取出数据，如何验证数据模型的有效性，处理数据模型之间的关系。
* 模板层：表现层，处理与表现相关的业务，例如如何在HTML页面中显示后台数据，Django框架作为一种Web开发框架，应该提供一种方便的设计方法管理和生成HTML网页。模板包含一些Django提供的特殊语法和HTML的部分代码，特殊语法用来将从视图中传递的数据动态地插入到模板中。
* 视图层：业务逻辑层，视图层是模型层与模板层之间的桥梁，视图负责存取响相应的模型数据，另外视图接受用户的请求，经对应的处理后返回响应的模板给用户，同时可以传递数据给模板用以渲染界面。

## 4.3 数据库设计

数据库存储一个系统的所有用户数据，其重要性不言而喻。一个好的数据库的设计对整个系统的设计具有相当重要的意义。数据库的合理性将会对程序的性能带来较大的提升。

本博客系统采用Mysql数据库进行数据存储，数据库的相关信息如下：

1. 数据库版本：MySQL 5.7.36;
2. 数据库可视化管理软件：Navicat Premiun 15.0.12;
3. 数据库名称：cblog；
4. 数据库基字符集：UTF-8；
5. 数据库排序规则：UTF-8\_General\_Ci;

本博客系统的数据库表设计如表4-1到表4-7所示，其他表的设计均由Django框架自动生成，此处未列出。

表4-1 数据库表设计 user表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段名** | **数据类型** | **字段长度** | **约束** | **说明** |
| user\_id | int | 20 | 主键、非空、自增 | 用户id，自动生成，自增 |
| user\_username | varchar | 255 | 非空 | 用户姓名 |
| user\_account | varchar | 255 | 非空 | 用户登陆账号 |
| user\_password | varchar | 255 | 非空 | 用户登录密码 |
| user\_email | varchar | 255 | 非空 | 用户邮箱 |
| user\_desc | varchar | 20 | 无 | 用户描述 |
| use\_datejoined | varchar | 20 | 非空 | 用户创建日期 |
| user\_is\_superuser | int | 20 | 非空 | 是否是管理员，默认为0（1是，0否） |
| user\_is\_lastlogin | datetime | 6 | 非空 | 最后一次登录日期 |

表4-2 数据库表设计 blog表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段名** | **数据类型** | **字段长度** | **约束** | **说明** |
| blog\_id | int | 20 | 主键、非空、自增 | 文章id，自动生成，自增 |
| blog\_title | varchar | 255 | 非空 | 文章标题 |
| blog\_add\_date | datetime | 6 | 非空 | 文章创建日期 |
| blog\_pub\_date | datetime | 6 | 非空 | 文章最后一次修改日期 |
| blog\_owner\_id | int | 20 | 非空,外键 | 文章创建者id |
| blog\_desc | varchar | 255 | 无 | 文章简介 |
| blog\_category\_id | int | 20 | 非空、外键 | 文章所属分类id |
| blog\_pv | int | 20 | 非空 | 文章浏览量，初始为0 |

表4-3 数据库表设计 category表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段名** | **数据类型** | **字段长度** | **约束** | **说明** |
| category\_id | int | 20 | 主键、非空、自增 | 文章分类id，自动生成，自增 |
| category\_name | varchar | 255 | 非空 | 文章分类名称 |
| category\_desc | varchar | 255 | 非空 | 文章分类描述 |
| category\_add\_date | datetime | 6 | 非空 | 文章分类最后一次修改日期 |
| category\_ pub\_date | datetime | 6 | 非空 | 文章分类创建日期 |

表4-4 数据库表设计 comment表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段名** | **数据类型** | **字段长度** | **约束** | **说明** |
| comment\_id | int | 20 | 主键、非空、自增 | 评论id，自动生成，自增 |
| comment\_content | varchar | 255 | 非空 | 评论内容 |
| comment\_time | varchar | 255 | 非空 | 评论日期 |
| comment\_is\_enable | int | 20 | 非空 | 是否显示评论（1显示，0隐藏，默认1） |
| comment\_article\_id | int | 20 | 非空、外键 | 评论所属文章id |
| comment\_author\_id | int | 20 | 非空、外键 | 发表评论的用户id |

表4-5 数据库表设计 profile表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段名** | **数据类型** | **字段长度** | **约束** | **说明** |
| profile\_id | int | 20 | 主键、非空、自增 | 用户id，自动生成，自增 |
| profile\_image | varchar | 255 | 无 | 用户头像 |
| profile\_birthday | varchar | 255 | 无 | 用户生日 |
| profile\_nickname | varchar | 255 | 无 | 用户昵称 |
| profile\_gender | int | 20 | 非空、默认1 | 用户性别（1男、2女） |
| profile\_email | varchar | 255 | 非空 | 用户邮箱 |
| profile\_desc | varchar | 255 | 无 | 用户描述 |
| profile\_address | varchar | 255 | 无 | 用户地址 |
| profile\_owner\_id | int | 20 | 非空、外键 | 用户简介所属用户id |
| profile\_gexing | varchar | 255 | 无 | 用户个性签名 |

表4-6数据库表设计 tag表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段名** | **数据类型** | **字段长度** | **约束** | **说明** |
| tag\_id | int | 20 | 主键、非空、自增 | 标签id，自动生成，自增 |
| tag\_name | varchar | 255 | 非空 | 标签内容 |
| tag\_add\_date | datetime | 6 | 非空 | 标签创建日期 |

表4-7数据库表设计 sidebar表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段名** | **数据类型** | **字段长度** | **约束** | **说明** |
| sidebar\_id | int | 20 | 主键、非空、自增 | 侧边栏id，自动生成，自增 |
| sidebar\_display\_type | int | 20 | 非空 | 侧边栏类型 |
| sidebar\_content | varchar | 255 | 非空 | 侧边栏内容 |
| sidebar\_sort | int | 20 | 非空 | 侧边栏排序字段 |
| sidebar\_status | int | 20 | 非空、默认1 | 侧边栏状态（1显示，2隐藏） |
| sidebar\_add\_date | datetime | 6 | 非空 | 侧边栏创建日期 |

## 4.4 系统模块设计

为了完成本博客系统的基本功能需求，对用户角色设计了用户注册模块、用户登录模块、发表文章模块、发表评论模块、个人中心模块、文章查询模块等六个模块、以满足用户的基本功能需求。Django自带的管理后台使得不用额外设计后台管理模块、该后台系统的功能完备，同时支持自定义界面。故后台模块在此未列出。

### 4.4.1 用户注册模块设计

注册模块是核心模块之一，注册模块为用户开启相应权限，在数据库中创建用户，在注册之后才能在本系统中发表文章。用户需要在注册页面中填写注册的必要信息，如邮箱，用户名等，视图层对应视图函数接收到请求后会验证提交数据的合法性，从而给出不同的响应。注册模块的流程图如图4-3所示：

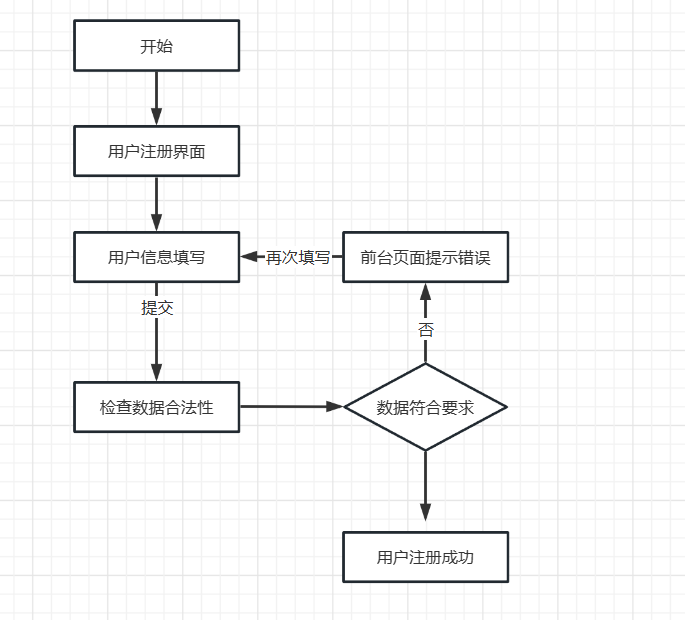


图4-3

### 4.4.2 用户登录模块设计

用户完成注册后，数据库中会创建一个账户，用户进行登录时，后台会在数据库中匹配是否有对应用户名存在，若存在用户名则继续对比密码，密码比对成功则登录成功，否则返回错误信息。登录模块的流程如图4-4所示

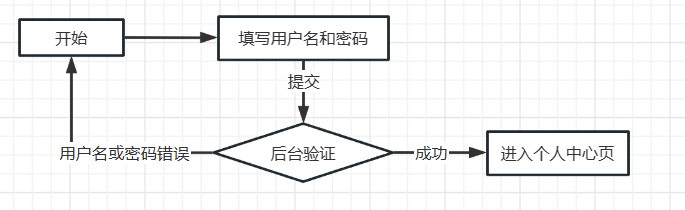


图4-4

### 4.4.3 发表文章模块设计

发表文章是博客系统的核心功能，用户只有在登录成功的状态下才能进入发表文章界面，发表文章的正文框集成了富文本的编辑器，同时支持Markdown语法和语法高亮。发表文章的表单中只有文章标题、文章正文和文章正文三个必填项，其余可以为空。表单验证合法性成功后会在数据库blog表中添加一条文章记录。发表文章流程如图4-5所示：

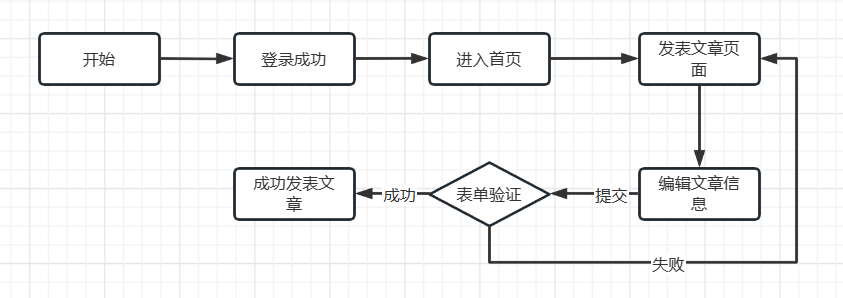


图4-5

### 4.4.4 发表评论模块设计

用户进入一篇文章的详情页时，可以在文章底部发表评论，评论框同样也集成了富文本的编辑器，支持Markdown语法。一篇文章的所有评论都会显示在文章底部，同时会显示评论时间和评论者的用户名。评论模块的流程如图4-6所示：

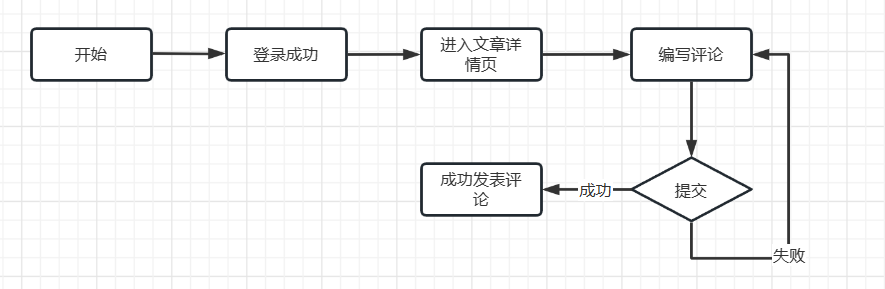


图4-6

### 4.4.5 个人中心模块设计

用户登录成功后会被重定向至个人中心页，在个人中心中用户可以看到自己的个人信息，并可以修改部分个人信息。个人中心模块的流程如图4-7所示

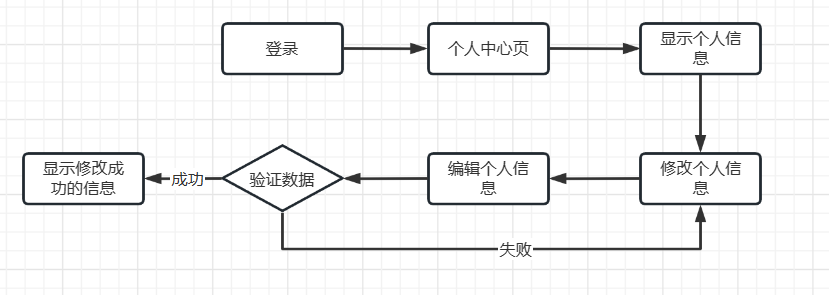


图4-8

### 4.4.6 文章查询模块设计

当博客系统的用户发表文章达到一定数量时，用户想要找到心仪的文章就显得尤为困难，所以文章的查询模块也是十分重要的。文章的查询支持模糊查询，后台会在文章标题、文章正文、文章简介等进行搜索关键词匹配，并将查询出来的结果进行分页展示。文章查询模块的流程如图4-8所示：

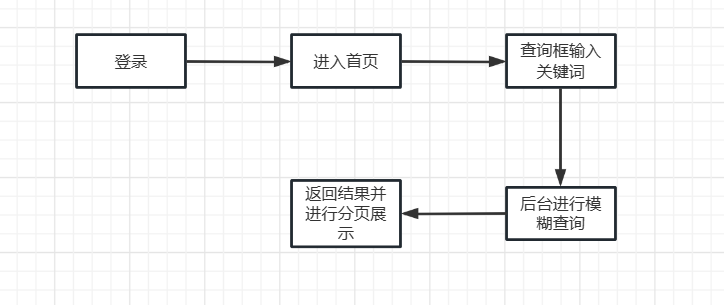


图4-8

## 4.5 本章小结

本章主要介绍了本博客系统的设计过程，介绍了整体的框架结构，明确了开发过程中的设计原则。通过对数据库中的几个重要数据表的详细介绍对本系统的数据库设计进行了说明。对于系统的关键模块分别做了介绍并用图像表示了相关的业务逻辑。

# 第五章 系统实现

## 5.1 登陆及注册模块实现

登录界面如图5-1所示，注册界面如图5-2所示：



图5-1 登录界面截图



图5-2 注册界面截图

登录及注册的核心代码如图5-3、5-4所示

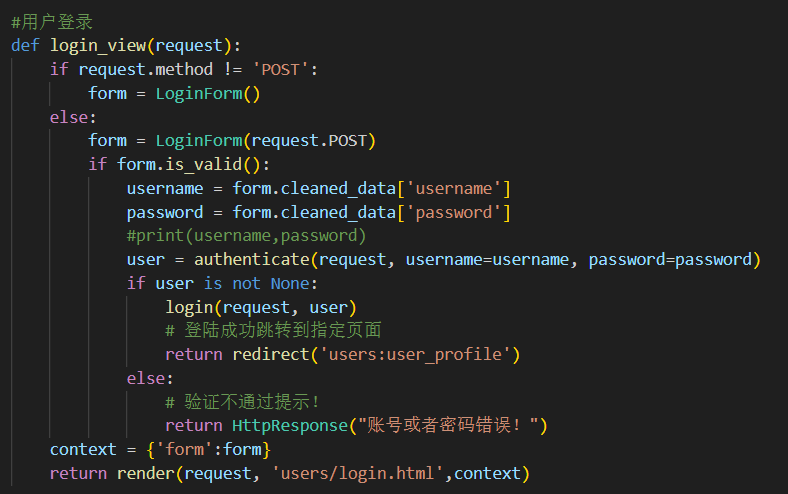


图5-3 登录模块核心源码

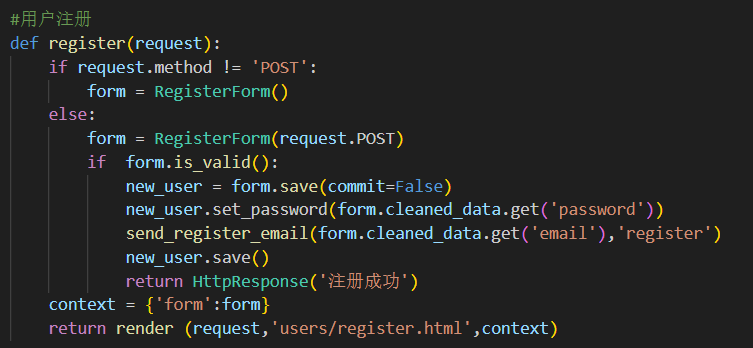


图5-4 注册模块核心源码

## 5.2 发表文章模块实现

用户登录后，可以在文章首页点击右下角的加号图标进行文章发布，文章首页如图5-5所示：



图5-5 文章首页展示截图

点击图标后进入编辑文章页面，文章编辑界面如图5-6所示：

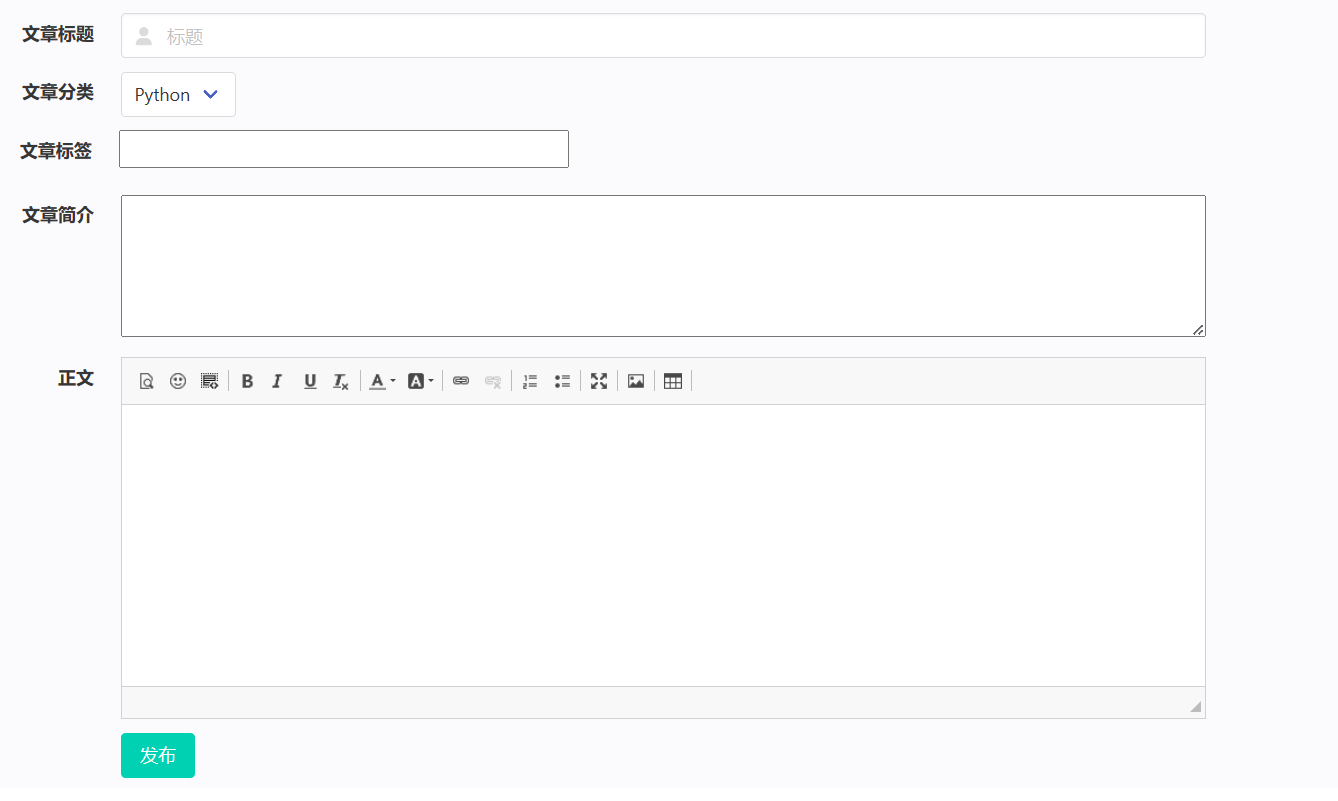


图5-6 文章编辑界面展示截图

输入完毕后点击发布按钮，若表单数据通过校验，则文章发布成功。

文章发布核心源码如图5-7所示：



图 5-7 文章发布模块核心源码

## 5.3 发表评论模块实现

点击一篇文章后会进入该文章详情页，已登录的用户可以在文章下方发表评论，文章详情及评论展示如图5-8、图5-9所示:



图5-8 文章详情页展示截图

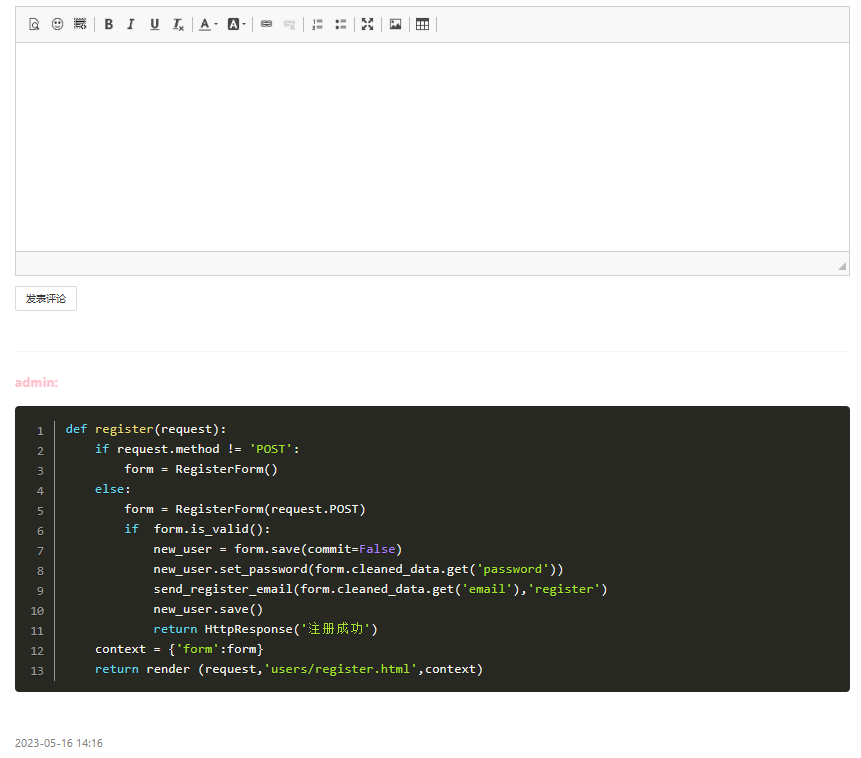


图5-9 评论框及评论展示截图

发表评论模块核心代码如图5-10所示



图5-10 发表评论模块核心源码

## 5.4 个人中心模块实现

个人中心页面和修改界面如图5-11和图5-12所示

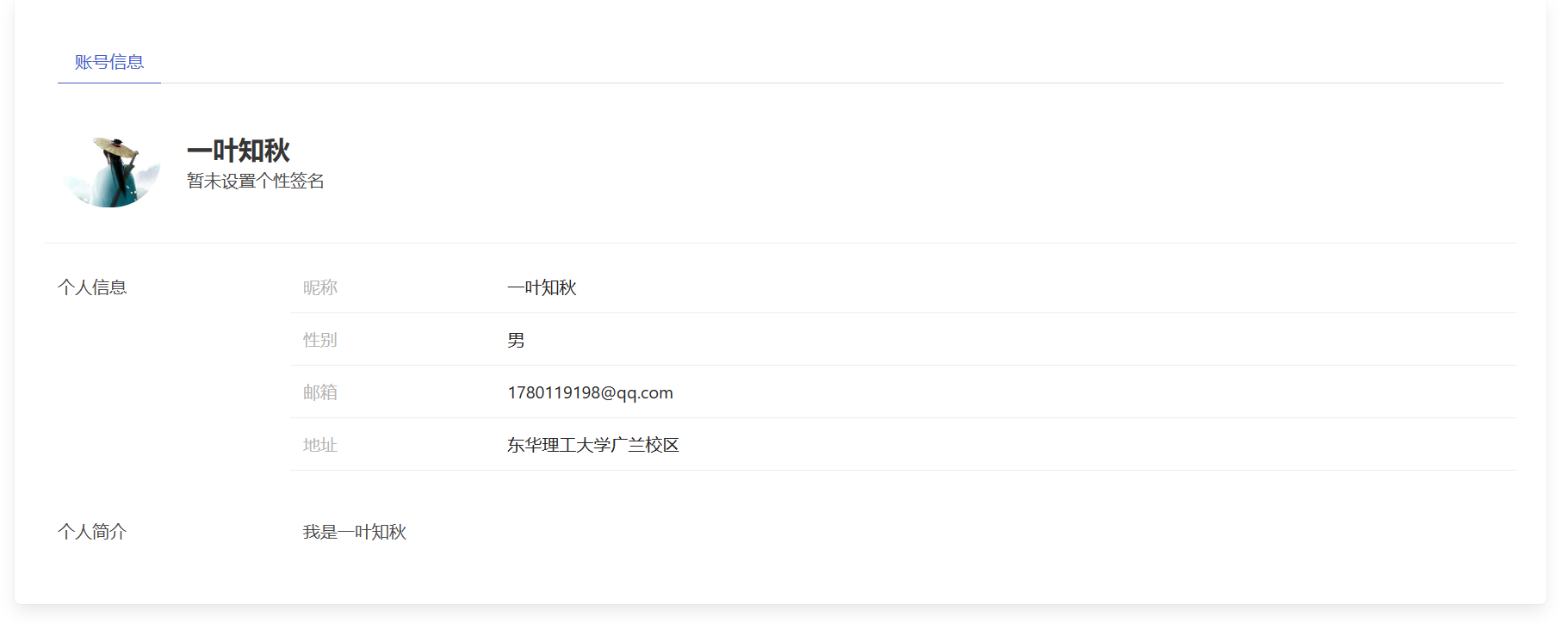


图5-11 个人中心界面截图

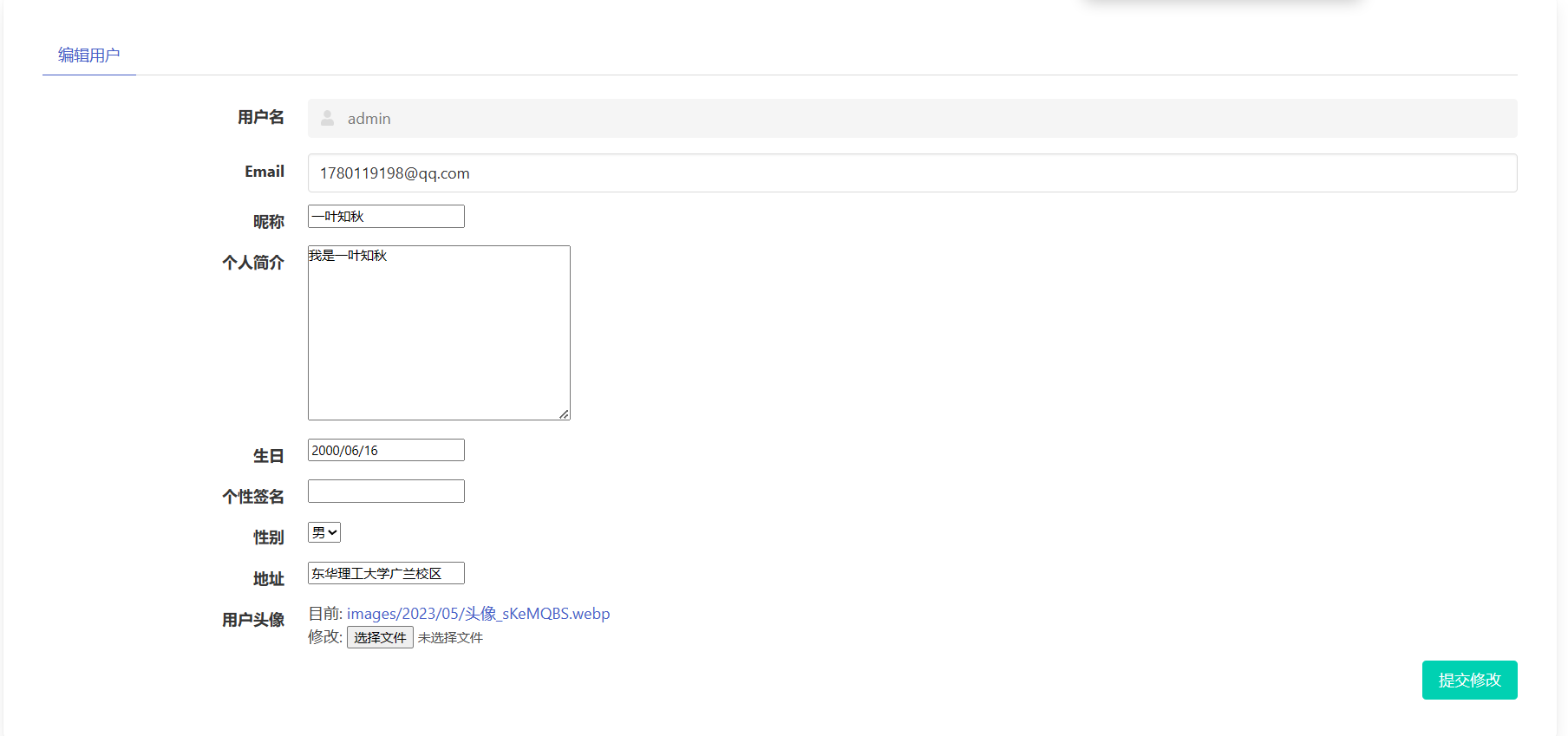


图5-12 修改个人信息页面截图

修改个人信息源码如图5-13所示：

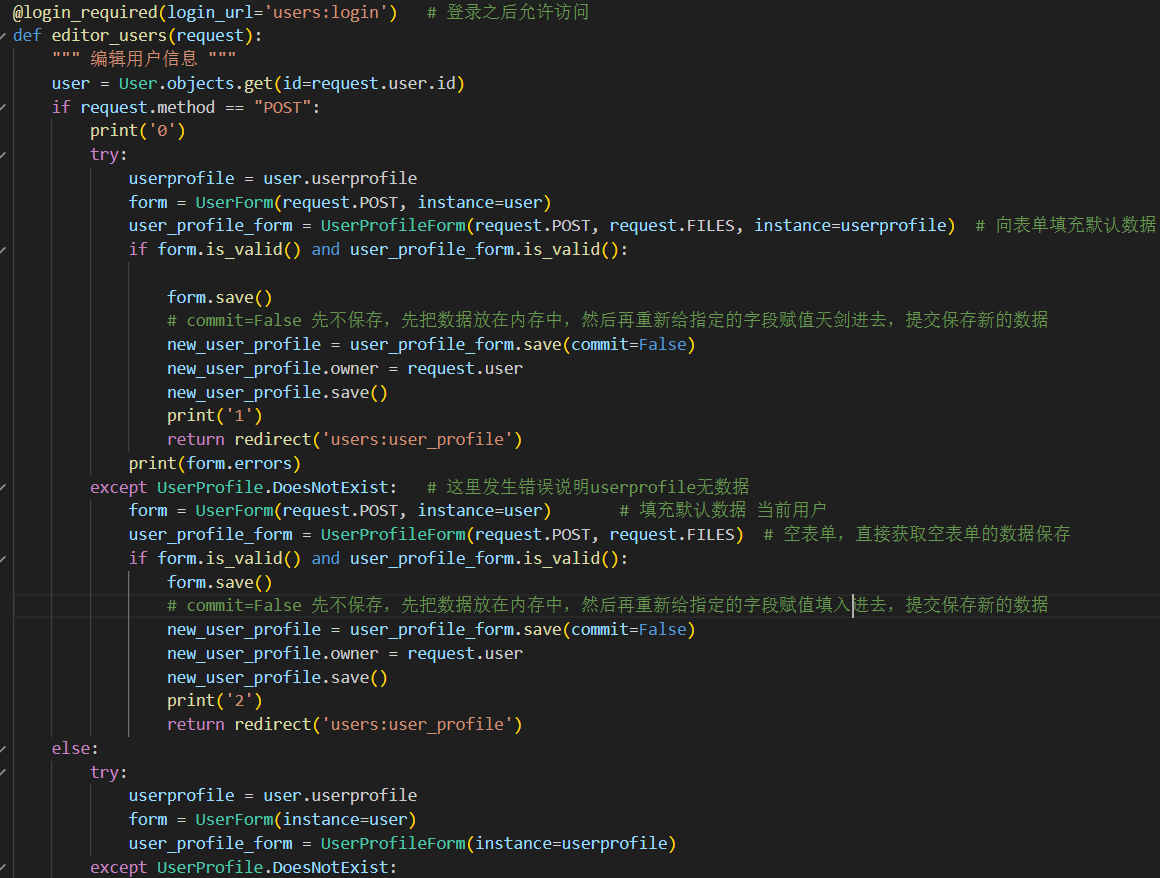


图5-13 修改个人信息源码

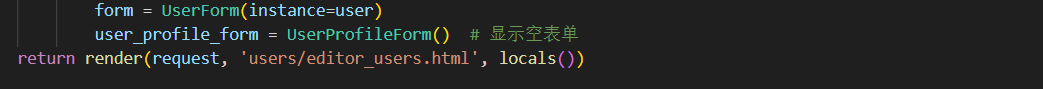


图5-13续图

## 5.5 文章查询模块实现

查询入口在博客首页首部导航栏中，如图5-14所示：



图5-14 搜索框截图

搜索结果会进行分页展示，如图5-15所示：

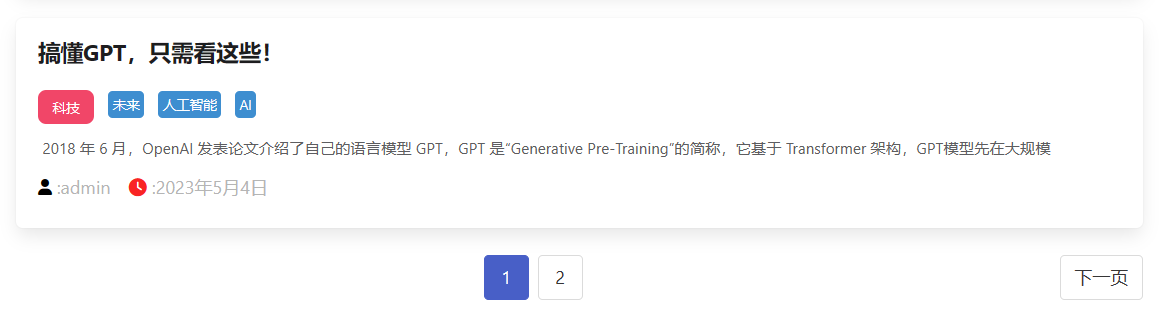


图5-15 搜索分页结果展示

查询模块代码如图5-16所示：

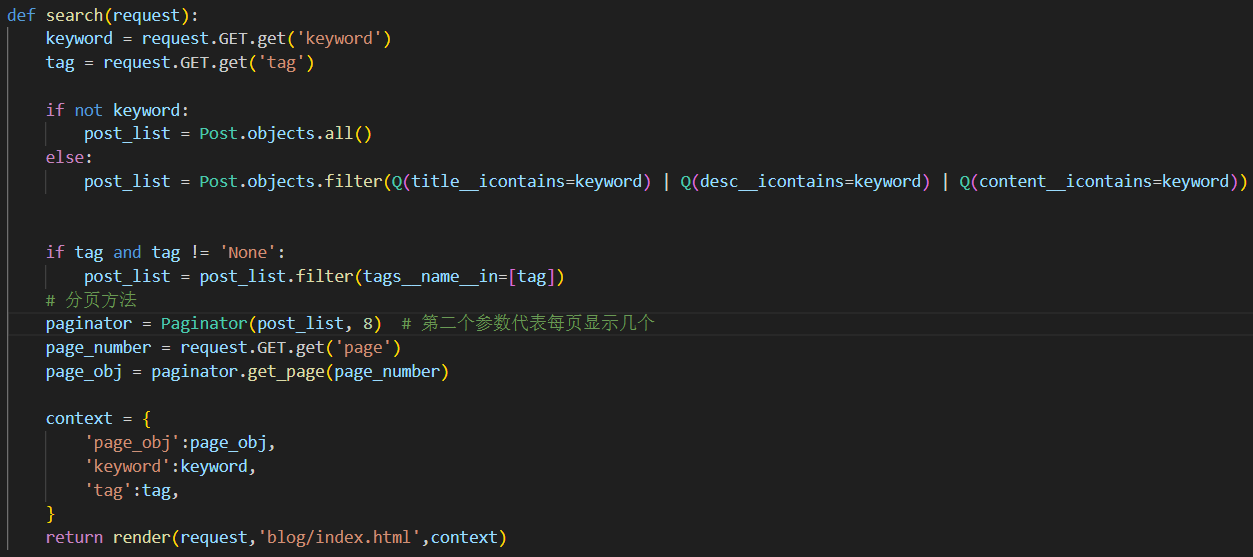


图5-16 搜索模块代码

## 5.6 本章小结

这一章主要是对博客系统中各主要模块的实现进行了描述，用本系统的部分界面截图和部分关键源代码，对系统实现的过程进行了简单的描述，让非开发人员可以简单地了解本系统的实现过程和结果，让他们能够更好地了解本系统的某些细节。鉴于篇幅的限制，我们无法对这一章中的各个模块进行全面介绍，所以对这一章中介绍的内容做了一定程度的精简。

# 第六章 系统测试

对于任何一个软件系统来说，在进行开发的过程中，以及在部署、交付之前，都是需要进行相关的测试的，测试能够帮助开发人员更快更准确的定位到系统中问题所在的地方，而且对开发人员解决问题和提高系统性能都有很大的帮助。对系统进行严格、规范的测试，能够防止在上线后，系统崩溃，或者发生系统异常，从而有效地避免造成重大的经济损失。

## 6.1 测试的目的及意义

开发本博客系统时，对各个模块的关键代码部分展开了大量的单元测试和功能性测试，以保证系统能够按照项目需求来开发，并且能够完成本项目的所有功能要求。系统测试能够帮助我们找到系统中的 Bug，能够保证各个模块的基本功能能够顺利地实现，并且能够减少模块之间错误的关联效应。除此之外，系统测试还能够帮助开发人员培养良好的编程习惯，提高系统健壮性和代码质量。

## 6.2 测试方法

测试方法的分类标准多种多样，按照不同划分标准，可大致分为以下几类：

1. 是否关心程序的内部结构和模块的具体实现：黑盒测试、白盒测试；
2. 是否需要运行程序或模块：静态测试、动态测试；
3. 依据软件工程的开发流程：单元测试(模块测试)、集成测试(组装测试)、确认测试(有效性测试)、系统测试、验收测试、Alpha测试、回归测试、Beta测试

单元测试（又被称作模块测试），它是软件测试环节中最小的单元，它的主要目的是验证模块（函数方法）的正确性，用于发现各模块中可能存在的 Bug。在单元测试中，要根据程序的参数、返回值等内部结构来设计不同的测试用例。单元测试可由单一测试或多个测试组合测试，测试过程中各测试相互独立，测试结果互不干涉。

集成测试（也被称为组装测试、联合测试），一般指的是以单元测试为基础，将每个子模块组装为一个程序系统进行测试。在将各模块进行组合的时候，要对各模块数据是否丢失、各模块间的功能影响、各子功能联合后是否达到父功能的预期目标、各模块各自的问题或缺陷组合后是否对系统造成了更为严重的放大性影响等因素进行考虑。

确认测试（也叫有效性测试）的目的是验证软件的每一个特定功能和系统的性能是否符合用户的需求。通常，验证测试的需求和系统性能指标都会在软件需求规格说明书中有明显而明确的规定。

验收测试，指的是在系统确认测试和软件运行环境配置之后，展开的一种以用户使用为主，开发人员进行辅助的测试，它通常是在用户的实际生产环境中进行的，并将会使用在生产环境中产生的真实数据，除了被测试软件系统的功能和性能之外，还需要对软件系统的兼容性、可移植性、可维护性和错误恢复等附加但十分重要的功能进行考虑。

系统测试，指的是把需要测试的程序或模块包装组合起来，形成一个整体，作为一个元素，加入到整个计算机系统中，并在真实的运行环境下，与计算机系统的各个组件（如硬件、软件、数据、外设、操作使用人员等）一起，对整个的计算机系统进行一系列的装配测试和有效性测试。它的主要目标是找出程序或模组与电脑系统的不一致和冲突的原因。

## 6.3 测试用例

本博客系统在开发、上线及部署的过程中进行了大量的测试，由于测试内容过于繁多，本节选取了用户注册模块和用户登录模块两个模块的测试用例进行展示说明

1. 用户注册模块功能性测试

测试目标：验证用户注册时填写的信息内容是否正确

测试数据(正确值)：姓名：李明 账号：liming 密码：112358 邮箱：[liming@qq.com](mailto:liming11238@qq.com)

模块测试用例如表6-1所示

表6-1用户注册模块测试用例

| **序号** | **测试内容** | **输入数据** | **期望值** | **实际值** | **测试结果** |
| --- | --- | --- | --- | --- | --- |
| 1 | 非空验证 | 漏掉一项信息不输入，直接提交 | 未填写处提示“请填写此选项” | 未填写处提示“请填写此选项” | 测试通过 |
| 2 | 账号格式验证 | 账号：admin，其它数据项格式正确，提交 | 账号处提示“已存在该用户名” | 账号处提示“已存在该用户名” | 测试通过 |
| 3 | 验证密码格式 | 密码：111，其它数据项格式正确，提交 | 密码处提示“请将此文本字符加长到6个字符或更多” | 密码处提示“请将此文本字符加长到6个字符或更多” | 测试通过 |
| 4 | 邮箱格式验证 | 邮箱：liming.com,其它数据项格式正确，提交 | 邮箱处提示“请在电子邮电中添加@符号” | 邮箱处提示“请在电子邮电中添加@符号” | 测试通过 |
| 5 | 两次密码比对验证 | 两次密码输入不一致，提交 | 密码处提示“两次密码输入不一致” | 密码处提示“两次密码输入不一致” | 测试通过 |
| 6 | 正确格式提交 | 全部字段按正确格式填写，提交 | 无错误，提交成功 | 无错误，提交成功 | 测试通过 |

(2）用户登录模块功能性测试

测试目标：登录验证

测试数据(正确值)：账号：admin 密码：123456

此模块测试用例如表6-2所示：

表6-2 用户登录模块测试用例

| **序号** | **测试内容** | **输入数据** | **期望值** | **实际值** | **测试结果** |
| --- | --- | --- | --- | --- | --- |
| 1 | 非空验证 | 漏掉一项信息不输入，直接提交 | 未填写处提示“请填写此字段” | 未填写处提示“请填写此字段” | 测试通过 |
| 2 | 账号验证 | 账号：wangwu，其它数据项正确填写，提交 | 账号处提示“用户不存在” | 账号处提示“用户不存在” | 测试通过 |
| 3 | 密码验证 | 密码：48488，其它数据项正确填写，提交 | 返回密码错误页面 | 返回密码错误页面” | 测试通过 |
| 4 | 用户名与密码格式验证 | 用户名：1780119198@qq.com，密码：1780119198@qq.com | 密码处提示“用户名不能与密码相同” | 密码处提示“用户名不能与密码相同” | 测试通过 |
| 5 | 正确登录验证 | 账号：admin，密码：123456 | 无错误，提交成功 | 无错误，提交成功 | 测试通过 |

## 6.4 测试结论

该系统在开发、部署和上线的过程中，对各种系统进行了测试，并发现了许多问题。与此同时，也正是因为在本系统开发的过程中，进行了严格、规范的各种测试，才可以准确地找出问题的根源，从而可以及时地对各种问题进行修补，从而不会拖慢系统的开发进度，最终帮助整个系统的正确、有效地完成，达到了预期的目标。

在此次系统开发过程中，所用到的各种测试和开发规范，尽管它们的工作量较大，遇到的问题也较多，但是它们确实对我本人造成了积极的影响。我相信，在本课题中所积累的系统测试和规范开发的经验，将会对我未来人生中职业生涯的健康发展起到重要作用。

## 6.5 本章小结

本章介绍了软件开发中的测试方法及其对开发的意义，并介绍了一些常用的测试方法。并选取了两个测试用例阐述了所用测试方法。

# 第七章 系统部署

在项目开发过程中，系统都是在本地环境上运行，只能从本机进行访问，故必须将系统部署在服务器上，供他人访问。本系统采用Django+Nginx+

# 第八章 总结与展望

## 8.1 总结

本课题设计并实现了一个用Python、Html、Css等语言编写的并基于Django框架、Bulma框架和MySQL数据库的博客系统。后台系统使用了Django自带的后台管理系统，前台系统针对普通用户开发，实现了用户登录、用户注册、发布文章、发表评论、修改个人信息、文章搜索、文章分页等实用功能，可以应对大部分场景下的业务需求。

本论文的结构安排依据软件开发流程的顺序，主要的工作总结如下：

首先，本文对课题的时代背景、研究目标和意义进行了说明，并从几个方面作了可行性分析。同时对课题所用技术和框架做了相关介绍，明确了整个课题的技术栈，理清了博客系统的各个功能需求。

其次，通过列出几个关键数据表的具体实现，对数据库相关设计作了详细说明。同时将系统功能统一划分为几个模块，并用模块的流程图和程序结构图对各个模块作了详细的介绍。这一过程为系统程序的发展确定了一个框架，以及所要达到的特定的功能性目标，从而保证随后所进行的全部系统发展都在该框架的范围内，朝着每个特定的功能性目标展开。

再次，对以上各部分的具体实施流程进行了详细的描述，并对以上各部分的功能进行了详细的说明。在这一部分中，主要是利用模块流程图、部分关键代码以及有关界面的截图，来解释各个模块的核心功能的实现过程和结果。

最后，给出了该系统的测试流程，测试结果，以及上线部署的过程和结果。在测试方面，本文首先对系统测试的重要性及意义进行了阐述，然后对系统的几种主要测试方法进行了列举和说明，最后给出了具体的测试步骤。在“系统部署”一章中，详细地介绍了博客系统的部署方法、部署流程。

## 8.2 展望

本课题设计并实现了一个博客平台系统，对一些基本功能已经实现完毕，但由于有限的设计时间和个人能力的限制，仍有许多不足之处以待改进。诸如前台展示界面的美观程度、用户实际体验、界面的人性化程度、系统效率等方面，都是在未来可以优化的部分。

现将未来可以考虑完善升级的地方分点列出：

1. 美化界面布局，提升用户体验。
2. 对评论模块追加二级评论功能，以满足用户的需求。
3. 可以授予某些用户权限，提供接口让其可以更改自己的文章，并显示更改记录。
4. 个人中心中显示自己已发布的文章。
5. 高并发访问的场景随着系统上线的时间越长越有可能发生，故应考虑优化代码和改进应对方法，以让其可以满足高并发场景下的性能需求。
6. 当一篇文章很长的情况下，定位文章中的某一章节对用户来说显得很困难，故可以考虑在文章中添加定位锚点，并生成导航目录便于浏览。