 **FOSHAN UNIVERSITY**

**本科生毕业设计（论文）**

**基于Django的个性化问答平台的设计与实现**

**学 院：**  **电子信息工程学院**

**专 业： 计算机科学与技术**

**学 号： 20160310321**

**学生姓名： 林上满**

**指导教师： 马莉（副教授）**

**（职称）**

**二〇二〇年四月**

**摘 要**

随着计算机技术的发展和共享经济的普及，网络上的问答平台逐渐成为当前信息与知识共享时代的重要平台。网络问答已经打破了常规传统的面授答疑形式，无数学习者趋向于愿意在互联网平台上分享信息与知识技能，也更愿意在互联网中寻找自己一些问题的解答。在此背景下，有效地利用网络社交问答平台，开展个性化的信息和知识交流活动便成为了值得研究的课题。本文的主要工作如下：

系统整体采用了前端和后端分开的MTV架构，讲数据模型层、用户界面层、业务逻辑层分离开来，以继承API的方式按功能模块进行开发，实现数据模型、业务逻辑分步设计，在此基础上渲染出用户界面，减少了系统内部代码之间的耦合度，同时提高了系统的可扩展性。在功能需求方面，主要突出个性化提问与回答的模块（个性化问答模块），并完善其他基础性功能，如用户注册登录，分享动态、文章以及用户间私信聊天等。在系统设计与实现方面，采取分独立服务器系统的部署方式，以及基于Python语言的Django框架进行快速开发。性能优化方面上，使用Redis缓存，提高热点数据的访问速率，其易扩展性亦可进行分布式的服务器集群形式部署。

**关键词：**个性化问答；知识共享； Django框架；Redis缓存

**Design and Implementation of Personalized Question & Answer Platform**

**Based on Django**

Lin Shang-man

**Abstract**

With the development of computer technology and the popularization of the sharing economy, the question and answer platform on the Internet has gradually become an important platform in the current era of information and knowledge sharing. Internet question answering has broken the conventional form of face-to-face question answering. Countless learners tend to be willing to share information and knowledge skills on the Internet platform, and are more willing to find answers to some of their own questions on the Internet. Under this background, the effective use of online social Q & A platforms to carry out personalized information and knowledge exchange activities has become a subject worth studying. The main work of this article is as follows:

The system as a whole adopts the MTV architecture with separate front-end and back-end. The data model layer, user interface layer, and business logic layer are separated, and they are developed according to function modules in a way of inheriting API. On this basis, the user interface is rendered, which reduces the coupling between the internal code of the system and improves the scalability of the system. In terms of functional requirements, it mainly highlights the personalized question and answer module (personalized question and answer module), and improves other basic functions, such as user registration and login, sharing of dynamics, articles, and private message chat between users. In terms of system design and implementation, a separate server system deployment method and a Django framework based on the Python language are used for rapid development. In terms of performance optimization, Redis cache is used to increase the access rate of hotspot data, and its easy scalability can also be deployed in the form of distributed server clusters.

**Key words:** personalized question and answer; knowledge sharing; Django framework; Redis cache

目录

[1. 个性化问答平台概述 1](#_Toc38923062)

[1.1. 个性化问答平台的研究背景 1](#_Toc38923063)

[1.1.1. 问答形式现状及发展趋势 1](#_Toc38923064)

[1.1.2. 个性化问答平台的竞争力 1](#_Toc38923065)

[1.2. 个性化问答平台的研究目标及意义 1](#_Toc38923066)

[1.3. 个性化问答平台需面对的挑战以及发展策略 2](#_Toc38923067)

[2. 个性化答问平台需求分析 2](#_Toc38923068)

[2.1. 平台设计目标 2](#_Toc38923069)

[2.2. 平台核心需求 2](#_Toc38923070)

[2.2.1. 功能需求 2](#_Toc38923071)

[2.2.2. 平台运行环境需求 3](#_Toc38923072)

[2.2.3. 安全与完整性需求 3](#_Toc38923073)

[2.3. 平台可行性分析 3](#_Toc38923074)

[3. 平台总体设计 4](#_Toc38923075)

[3.1. 平台概述 4](#_Toc38923076)

[3.2. 内部逻辑架构 4](#_Toc38923077)

[3.3. 平台总体部署架构 5](#_Toc38923078)

[3.4. 数据库设计 6](#_Toc38923079)

[3.4.1. Django ORM架构 6](#_Toc38923080)

[3.4.2. 模型类设计 7](#_Toc38923081)

[4. 平台详细设计 7](#_Toc38923082)

[4.1. 基础开发运行环境搭建 7](#_Toc38923083)

[4.2. 功能模块设计 8](#_Toc38923084)

[4.2.1. 用户个人功能模块 8](#_Toc38923085)

[4.2.2. 动态分享模块 8](#_Toc38923086)

[4.2.3. 文章发布模块 8](#_Toc38923087)

[4.2.4. 个性化问答模块 8](#_Toc38923088)

[4.2.5. 用户间私信聊天模块 9](#_Toc38923089)

[4.2.6. 消息通知模块 9](#_Toc38923090)

[4.2.7. 全站搜索模块 9](#_Toc38923091)

[4.3. 平台上线部署 9](#_Toc38923092)

[5. 编码实现及测试 10](#_Toc38923093)

[5.1. 平台模块结构 10](#_Toc38923094)

[5.2. 注册及登录实现 11](#_Toc38923095)

[5.3. 个性化问答 11](#_Toc38923096)

[5.4. 单元测试及系统测试 12](#_Toc38923097)

[5.5. 性能及安全测试分析 13](#_Toc38923098)

[6. 总结与展望 13](#_Toc38923099)

[参考文献 14](#_Toc38923100)

[致 谢 15](#_Toc38923101)

# 个性化问答平台概述

## 个性化问答平台的研究背景

本小节讲分别阐述当代在解决常规性问题的问答形式现状及其在未来五年的发展趋势，还有个性化问答平台在现阶段表现出的独特优势及其竞争力。首先，问答的形式在日常生活中或教育行业中都是非常常见的一种信息交流、知识共享形式，在学习者想到问题疑点的时候，即会希望求其解。结合以上情景，对本个性化问答平台存在的问题研究背景进行阐述。

### 问答形式现状及发展趋势

传统的问答授业解惑形式或仅存在一种形式，即面对面的交流，这可能是最有效且最友好的形式。在传统的问答形式中，学习者为求解一些问题，可能需要付出比较大的代价才能达到面对面的交流，才能获取知识和信息。

但如今是信息化时代，在信息共享和互联网发展十分蓬勃的时候，极需解决问答传统形式的不便。在这时候，新掀起了全民网络学习的高潮，无数网民通过互联网会通过搜素引擎、网络教育平台来获取新的信息与知识。比如，会在百度搜索引擎中检索关键词、中国慕课教育平台进行学习。那么，在互联网上提问或回答别人的问题，就能比较快速的满足学习者的需求，这样的学习代价是非常小的，同时也能使不具备传统问答条件的学习者得到知识和信息的交流，互联网问答便成为了当下必然的发展趋势。

### 个性化问答平台的竞争力

现在互联网上的搜索引擎、网络学习平台[1]均能解决互联网学习者的单个需求，但他们不能在一个平台上满足多方面的需求。在这种情景就会出现一个问题——缺乏一个相对完善的整合型平台。比如，学习者在搜索引擎上能检索关键词得到自己想要的信息，但得到该信息后无法再在同一个搜索引擎上获取其相关联的问答信息，也无法了解其相关性的其他共享信息。这时候，个性化问答平台独特的有点就彰显出来了。个性化问答平台是一个整合型平台，它综合了问答、相关性信息共享的功能，并可以在平常的问答形式中得到标准的答案，也能在平台中能了解到其他用户分享的信息科普。这样个性化问答平台就在一定程度上解决了知识信息共享和问答不相互关联的问题，打破独立平台独立功能的限制，更综合地向学习者提供了多方面信息。

## 个性化问答平台的研究目标及意义

本文研究的目标及意义是通过对个性化问答[2]形式的研究，提出个性化问答的知识信息共享模式，并设计与实现整合型的基于浏览器[3]访问的个性化问答平台，减小传统问答形式的巨大代价，同时满足学习者获取跨平台地综合整合信息与知识需求，使得求学者能够较为方便地在该个性化问答平台中获取多种相关的互联共享。

## 个性化问答平台需面对的挑战以及发展策略

当前“互联网+”时代发展非常迅速，知识共享形式也会在不断进行改善与简化，市场竞争会逐渐更激烈。在多个相关性平台系统整合会是互联网大趋势的前提下，本个性化问答平台需要面临的挑战也是比较大的。互联网中并不缺乏大型系统，初生的、跨系统的平台需要茁壮发展，在独特的整合型以及免费性条件吸引互联网用户，以长期地经营好该个性化问答平台，这样才是一个好的平台发展。

# 个性化答问平台需求分析

## 平台设计目标

本文探讨的是设计与实现基于Django[4]的个性化问答平台，用户可以通过浏览器注册并登录到该平台，按自己的需求可分享动态、发表个人文章、提问问题以及回答问题、对其他用户进行私信聊天等，个性化地获取、分享知识信息以及进行问答。本平台设计应达到以下目标：

1. 用户注册与登录：用户可以自行进行免费注册或者使用社交平台账号登录到本平台，对用户展示的个人信息维护。
2. 动态分享功能：用户可以随时发表自己的学习动态，以分享个人动态的形式进行学习性打卡。所有用户可对动态进行点赞或评论。
3. 发布文章功能：用户可以发布一些知识信息分享性的文章，支持Markdowm格式的编辑与预览，提供其他用户阅览与科普。所有用户可对文章进行点赞或评论。
4. 问答功能：用户可以在平台上进行自由提出问题，也可以为他人的问题作出回答，所有用户均可对问题/回答进行支持或反对的投票，问题的提问者可以对回答进行采纳以标记为最佳回答。
5. 用户间私信聊天功能：用户之间的互动不局限于功能性互动，还可以主动发送私信聊天给其他用户。
6. 消息通知功能：当有其他用户进行互动时，能够即时通知用户。
7. 全站搜索功能：用户可以对平台进行关键词检索，以方便用户查阅获取所需信息。

## 平台核心需求

### 功能需求

平台功能应按照用户实际的个性化需求进行设计，以登录用户作为载体，进行动态分享、文章发布、个性化问答、用户间私信聊天、消息通知、全站搜索等核心功能的使用。

### 平台运行环境需求

* 云服务器端硬件性能需求

服务器系统：Linux 32/64bit操作系统；系统云盘：40GB以上；带宽：2Mbps以上固定带宽；内存：1GB以上

* 平台运行环境需求

Python版本：Python 3.5+运行环境；中间件支持：Mysql、Nginx、Supervisor

* 开发及测试需求

开发继承环境：Pycharm Pro IDE；单元测试：Django-Test-Plus第三方库

* 用户客户端需求

用户登录客户端：允许用户多终端浏览器访问

### 安全与完整性需求

对用户账户数据加密存储的形式，以及使用Auth 2.0进行登录认证，同时使用Django的XSS和CSRF防御系统[5]，对平台数据防泄漏防御；在服务器层面，使用云服务器安全策略组以及服务器防火墙软件进行访问控制配置，保证平台安全性。另外容灾机制上会对数据进行每日增量的异地备份，保证平台数据完整性。

鉴于平台出于初步阶段，数据量等级不会太高，可适当降低服务器集群和负载均衡的标准。

## 平台可行性分析

可行性分析是项目设计开始的初步阶段，是验证一个项目是否在特定环境中具备其能发挥作用和表现意义的重要分析步骤。本文在通过调查了问答平台的现状以及发展趋势，分析了项目背景以及软件关键技术，证实平台确切能从基层上满足当前用户个性化问答的缺口需求。

在经济可行性方面，本平台主要由Linux、Mysql、Python、Nginx等软件均具有开源许可的软件层面上进行开发，开发经费仅集中在云服务器及开发人工成本上，该成本是必要支出费用且成本可控。

在技可行性方面，软件开发使用Django框架搭建及Python Web技术栈进行开发，服务器端技术采用Mysql、Nginx等开源中间进行上线支持，采用Celery框架的进度管理[6]，平台整体技术配置成熟，技术资源强大。

在社会运营可行性上，国内研究现状表明跨系统的知识信息共享存在操作麻烦、信息相关性低等问题，本平台整合为单平台，符合用户实际需求，在市场上具备独特优势竞争力。

在操作可行性上，平台允许用户进行多终端登录，因此用户只需要浏览器访问即可，平台功能设计符合用户需求和使用习惯。

# 平台总体设计

## 平台概述

遵循低耦合、高内聚的思想，本系统采取前后端分离的架构进行开发。后端开发通过继承Django通用开发父类完成API的开发[7]，极大减少了后台功能性接口代码冗余，因此只需要在此基础上完善业务后台逻辑即可。前端采取Django中Template模板形式，由后端业务逻辑对此进行渲染，再展现给用户。平台逻辑层的独立性和前端用户界面交互的简便，使得整体前后端架构清晰，同时也简化了项目后续的运维工序。

图3-1为平台模块架构图，主要包含动态分享模块、文章发布模块、个性化问答模块、用户间私信聊天模块共四大功能性模块，以及用户个人中心、消息通知、全站搜索共三大基础性模块。

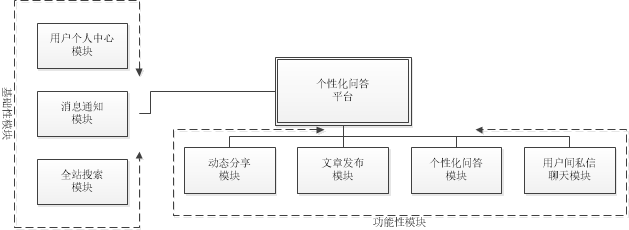


图3-1 平台模块架构图

## 内部逻辑架构

平台内部调用Django的MTV模式，图3-2为平台内部逻辑图。

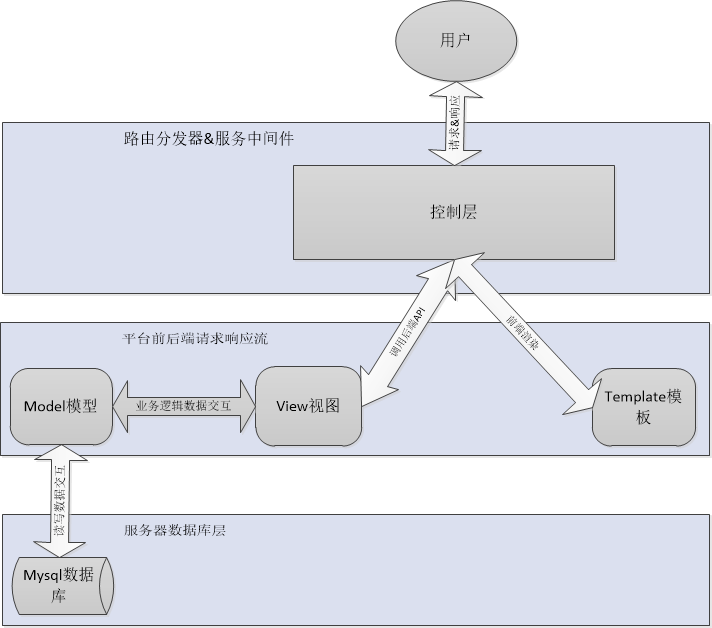


图3-2 平台内部逻辑

其中Model模型作为载体，负责映射业务对象与数据库存储对象（Object Relational Mapping，简称ORM）[8]；Template模板负责前端用户界面的展示；View视图负责业务逻辑处理，封装成后端API接口以供其他前端控制层调用。

## 平台总体部署架构

平台部署设计为B/S架构，图3-3为平台总体部署架构图。

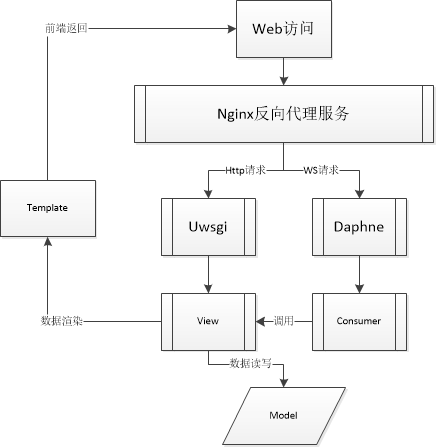


图3-3 平台总体部署架构

平台由用户向平台服务器端发起Web访问请求，服务器端使用Nginx配置反向代理，将用户请求转至平台核心路由控制层，并由核心路由判断请求协议。

如果请求协议是Http协议类型则由Uwsgi服务器处理，如果请求协议是WebSocket协议类型（简称WS协议类型）则由Django官方推荐Asgi服务器Daphne服务器来处理。

平台的核心后端是View层，是业务的逻辑处理层，后端处理时会封装需要构造的数据至Template模板，完成数据渲染，最后返回至用户客户端（B端）浏览器；当接收到的请求为WS协议类型，则会由Consumer层来处理相关同步/异步通讯，其他业务逻辑会调用View核心逻辑层的API接口。

Model是平台数据存取层，该层是Django内嵌的一种对象关系映射框架（Object Relational Mapping，简称ORM）。平台后端的所有与数据库想关的事务均会在Model层向底层Mysql数据库来存取，数据库表不会在平台层呈现，这也是Django快速开发的特性。

## 数据库设计

### Django ORM架构

平台设计采取Django MTV模式，在Model层中使用了Django内嵌的对象关系映射框架（即ORM）。该内嵌ORM架构[9]是一种为了解决面向对象与关系数据库存在的互不匹配的现象的技术。通过使用描述平台所需数据对象和数据库之间映射的元数据，将平台数据对象自动持久化到关系数据库中。在此过程中，数据实体的读写过程对平台后端透明，后端业务逻辑层亦无需关注底层数据的交互实现方式，真实的sql数据语句交互由Django ORM架构来构造并进行数据读写优化，大大简化了开发过程与难度。

因此本平台十分愿意采取该内嵌框架来实现数据底层处理，这对平台开发技术人员是非常友好的，技术人员甚至无需关注底层数据表、无需编写sql交互语句，这样的开发效率是非常高的。

### 模型类设计

本平台按功能模块对数据模型类进行设计，其中全站搜索模块不设计数据交互。其他六大功能模块的数据模型类设计如下：

1. 用户个人中心模块

用户模型使用Django-allauth第三方库实现注册、登录认证（允许使用Github账号授权登录）。本平台在继承第三方库的用户模型类上，扩展了平台用户的元数据，如新增了职业、城市、社交链接等。

1. 动态分享模块

动态分享模块创建News Model，字段包含分享用户、分享内容、点赞关联用户、评论关联动态模型、创建时间、更新时间。当然，使用uuid作为唯一主键。

1. 文章发布模块

文章发布模块创建Article Model，字段包括文章状态、文章标题、文章发布用户、文章封面图片、文章内容、文章标签、创建时间、更新时间。

1. 个性化问答模块

个性化问答模块需要创建两个数据模型类，分别是Question Model和Answer Model。

Question Model字段包括问题状态、问题标题、提问用户、问题内容、最佳回答标记、投票关联模型、问题标签、创建时间、更新时间。

Answer Model字段包括回答用户、关联问题模型、回答内容、最佳回答标记、投票关联模型、创建时间、更新时间。

以上问题/回答数据模型均设置了投票关联模型，因此投票关联模型Vote Model设置字段包括投票用户、投票值、关联问题/回答模型。

1. 用户间私信聊天模块

用户间私信聊天模块创建Message Model，字段包括消息发送用户、消息接收用户、消息体、已读标记、创建时间。

1. 消息通知模块

消息通知模块创建Notification Model，字段包含动作发起用户、通知用户、已读标记、动作标记、创建时间、更新时间。

# 平台详细设计

## 基础开发运行环境搭建

项目构建使用Cookiecutter创建Django项目，Linux服务器端使用pipenv安装虚拟环境作为平台运行环境，平台开发环境使用win 10操作系统软件Pycharm Pro连接服务器进行远程开发，相关服务配置（如创建Django应用配置命令）使用SSH连接服务器进行配置。

## 功能模块设计

### 用户个人功能模块

用户数据模型继承Django-allauth第三方库，使用OAuth 2.0协议进行登录认证，后端View视图创建查看用户详情接口、用户更新个人信息接口，实现用户个人信息维护基本功能。其他接口如更新账号密码、邮箱绑定验证接口由第三方库提供。

### 动态分享模块

动态分享模块主要有三个接口，分别是NewsCreateView、NewsListView、NewsDeleteView，以提供分享动态、查看动态、删除动态三个主要功能，这三个类均会继承Django的接口基本类BaseView，通用类视图能使开发难度降低，也会使代码显得更为简洁、高效。还有其他辅助性接口如点赞动态，则自定构造后端相应处理Ajax请求的Web响应函数即可。另外，动态的评论功能也会调用NewsCreateView接口，但会使用不同的评论关联参数区分该数据为动态或动态评论。

### 文章发布模块

文章发布模块与动态分享模块有一定程度上的相似性，均可创建继承Django的通用视图类的子类视图，来处理新增文章、显示文章列表、查看文章详情、删除文章的基本功能。这里文章会采用Django-slugify和Django-taggit库，分别作用于文章详情页url显示标题、处理文章标签管理。同时，为文章编辑支持Markdown格式，本平台采用Django-markdownx第三方库，本平台在此处设计为编辑Markdown文章实时预览功能，使用用户体验更为友好。文章的评论使用Django第三方模块django-contrib-comments，简单高效地实现文章关联评论功能。

### 个性化问答模块

个性化问答模块的设计会相对较为复杂一些。请求逻辑流按功能可分为：用户发起问题提问、用户向问题投票、用户回答问题、用户向回答投票、问题发起者采纳最佳回答。

为提高代码效率，此处采用ORM框架的GenericRelation关联模型类，即创建投票公共类，关联问题/回答数据模型。

同样，在后端视图逻辑处理层仍使用Django通用视图类来处理问题/回答的创建对象、查看对象列表、查看对象详情、删除对象的请求。对问题/回答投票的逻辑处理，使用前端发起Ajax请求，由后端自定构造后端相应处理函数即可。

对问题发起者采纳最佳回答请求，依然使用Ajax请求处理，但需添加问题提问者判断，当为问题发起者才有权限采纳最佳回答。

### 用户间私信聊天模块

用户间私信聊天模块在视图类中仅需创建继承于通用类视图父类的CreateView和ListView，对单体消息使用DetailView以接收即时通讯。这里设计涉及即时通讯，需使用WebSocket类型协议[10]。在Uwsgi 2.0中引入WS支持，但由于服务架构尚未成熟，在此设计使用官方推荐的成熟的Asgi服务器Daphne来处理WS请求。在Django的Asgi里，使用Consumer作服务提供者，完成同步/异步通讯请求的处理。

当私信聊天双方用户均为在线时，使用第三方库Django Channels实现WS的即时通讯编程。Channels是基于完全异步的层，以同步模式运行Django框架本身，异步处理使用套接字链接。通讯使用聊天用户唯一用户名创建通道层，当聊天消息接收者为用户本身时，使用WS即时通知用户，并将消息体发送至接收者，实现多用户、多聊天通道层的即时通讯私信聊天。

### 消息通知模块

消息通知模块也属于WS响应类型，当用户操作评论动态或点赞文章等会触发消息通知的功能时，在核心后端业务逻辑处理层会调用消息通知接口，同步或异步通知被提醒用户。当被提醒用户在线时，存储消息通知对象实例时会同步实现即时通知；当被提醒用户不在线时，存储消息通知对象实例并不会即时通知用户，而在等待用户上线时检测该用户未读消息，再以未读消息形式通知被提醒用户。

服务器端以Daphne做出响应时，客户端会在javascript脚本中实现即时通讯的展示，呈现给用户界面。

### 全站搜索模块

全站搜索模块可以搜索发布文章、分享动态、个性化问答、注册用户、对象标签等多个数据模型实体数据。使用自安装的第三方库Django-haystack，配合开源的分布式全文搜索引擎Elasticsearch服务[11]，实现对全站数据对象索引类的搜索功能。

## 平台上线部署

平台采用服务器远端开发形式，在开发阶段也会运行在服务器上，因此上线部署会较为简化一些，在开发阶段已经中间件服务配置完成，上线部署只需对中间件进行调优操作即可，如Nginx性能调优。

# 编码实现及测试

## 平台模块结构

在Pycharm Pro IDE开发环境下，平台模块结构如图5-1平台模块结构图所示。

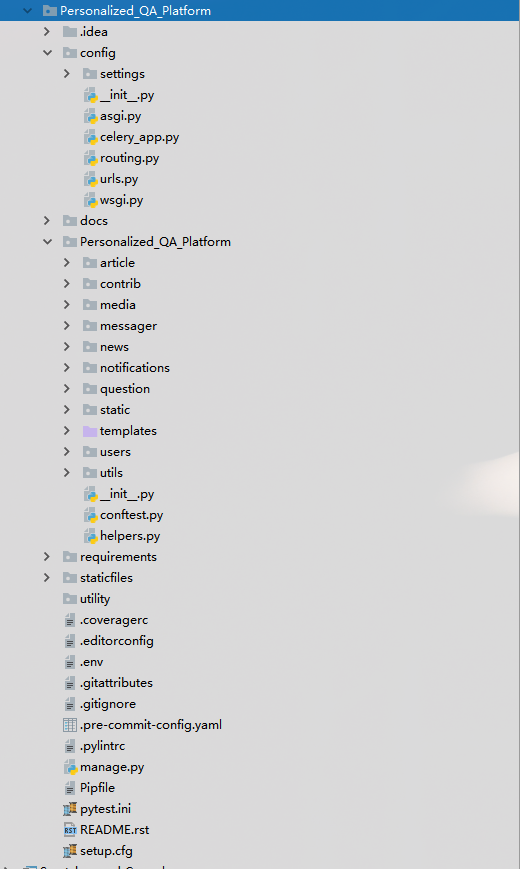


图5-1 平台模块结构图

平台按业务功能划分模块，每个模块均属于Django App应用。平台将用户个人功能模块、动态分享模块、文章发布模块、个性化问答模块、用户间私信聊天模块、消息通知模块、全站搜索模块集成在一起作为个性化问答平台，每个模块也是一个独立功能的App应用，这样的整合型平台在结构上也是十分清晰的，代码架构也是非常完整的，充分地发挥了MTV设计模式的高效性、稳定性。

## 注册及登录实现

通过浏览器访问平台登录入口，用户登录界面如图5-2用户登录界面所示。

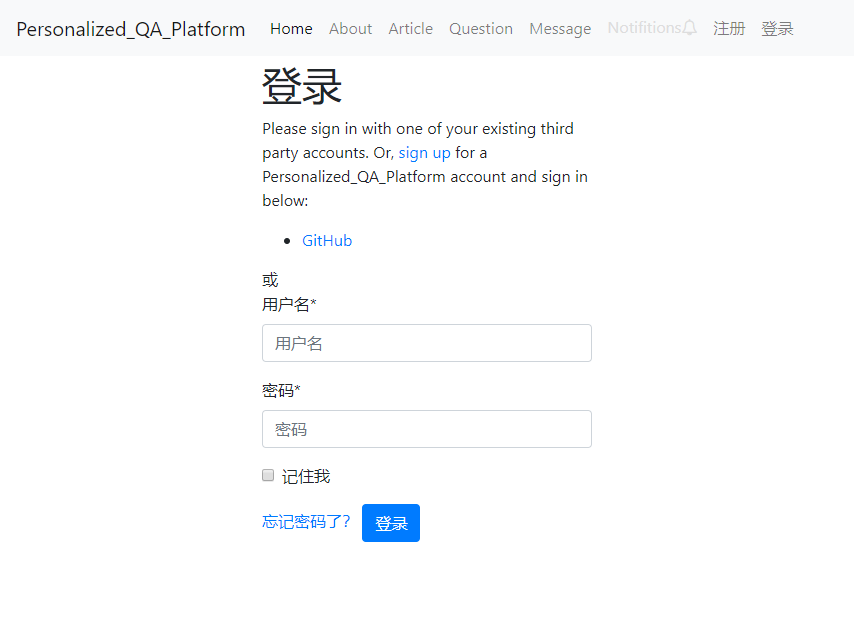


图5-2 用户登录界面

使用第三方平台Github授权登录后，登录重定向跳转到首页（动态页），图5-2为用户登录成功截图。

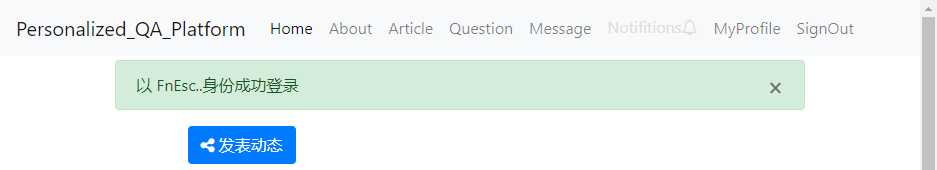


图5-2 用户登录成功

## 个性化问答

根据平台详细设计编码实现，访问个性化问答模块，用户能够得到图5-3为个性化问答详情页。



图5-3 个性化问答详情页

## 单元测试及系统测试

单元测试采用Django-test-plus第三方库增强单元测试[12]，个性化问答模块的单元测试报告如图5-4所示。

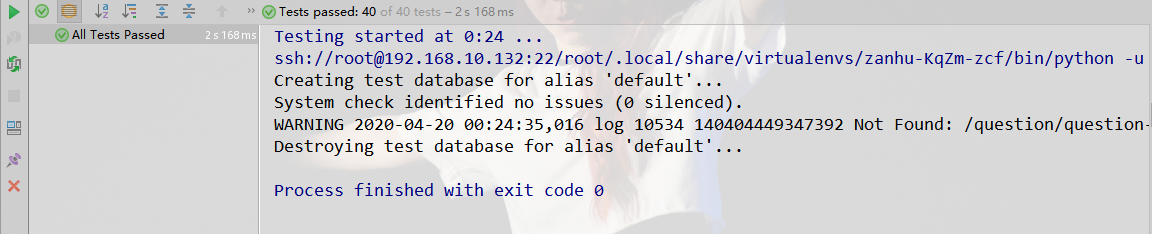


图5-4 个性化问答模块单元测试

系统测试目的是验证系统是否满足了需求规格的定义，模拟真实用户正常使用，找出与需求规格不相符合或与之矛盾的地方，提高系统的高可用性。这里按功能测试、接口测试、上线测试、兼容性测试多维度对平台进行测试。测试分析结果如下：

该平台采用B/S架构的实现形式，实现了用户OAuth认证管理、个性化问答等共七大功能。经系统测试，系统基本满足需求规格说明书明确的各项需求，系统运行正常，且系统的数据处理能力较强，数据读写存储状态良好，Nginx等中间件服务[13]协调性较好，发现缺陷系统解决完毕，软件能力符合发布的要求。

## 性能及安全测试分析

平台服务性能是衡量用户体验的重要标准，安全性是衡量平台保护用户数据安全是否合格的指标之一。在互联网技术发展十分蓬勃的今天，Web方向的网络安全攻击是较为常见的，其中包括sql注入、XSS攻击、CSRF攻击等，本平台使用Django的XSS和CSRF防御系统作为标准安全防护，服务器端使用防火墙做访问控制。经性能及安全性测试，本平台基本符合Web端安全平台，合格上线。

# 总结与展望

当今互联网发展十分快速，它的发展带动了其他行业的共同发展，包括传统工业、电子商务等典型代表。互联网技术使得网民生活更为简单、更具精彩。传统的问答模式已经不能满足现在快节奏的学习生活，人们渴望更快速获取更专业、更标准、更符合自己的知识信息。因此，个性化问答平台将会快速发酵而在互联网生活中普及。个性化问答平台集成跨系统信息，将多方面相关联的知识信息分享整合在本平台上，从而简化人们收获信息的过程、降低获取知识信息的代价，这增强了平台的竞争力，提高了平台的质量。

本文仔细分析了传统的学习问答方式背景，设计并实现了基于B/S架构的个性化问答平台。

从技术上讲，系统采取基于轻量级Python语言的Web框架Django，设计模式采取快速开发的MTV模式，业务逻辑，代码设计层次清晰，耦合性、复用性较高，模块化的设计、快速开发模式的实现及后期维护工作均较为简化。服务器层次上来讲，平台运行较为稳定，允许流量高并发。

在用户体验上，平台功能相对其他单一类型系统来说是比较丰富的，能较高满意度地满足用户个性化问答等需求。平台操作简单，在用户界面、用户体验均较为满意。平台核心功能为个性化问答功能，在上线后将会大力度推广该功能以吸引用户注册、使用。上线后，根据用户反馈，将继续对平台进行完善或增加其他新功能，用户至上。

相信在未来，个性化推广会在互联网平台有着比较重要的地位，因此，个性化相关技术、相关系统平台或会快速成长，占据市场。在此，需要更多的思考来拓宽业务功能，以满足更多用户的需要。

# 参考文献

1. 白相辰. 基于Django框架的Web在线教育平台的设计与实现[D].北京交通大学,2019.
2. 沈闻. 基于问答社区的个性化服务研究[D].扬州大学,2009.
3. 钱志远.基于Web的Python编程环境研究[J].数字技术与应用,2016(10):54.
4. 白昌盛.基于Django的Python Web开发[J].信息与电脑(理论版),2019,31(24):37-40.
5. 韩可彧. 基于Django的XSS和CSRF防御系统的研究与实现[D].武汉邮电科学研究院,2018.
6. 吴斌. 基于Django和Celery框架的进度管理平台的设计与实现[D].华中科技大学,2016.
7. 韦立梅,苏兵.Django框架下Python网站开发过程综述[J].电脑与电信,2019(10):54-56.
8. Zheshu Jia. Marine Integrated Database Management System Based on Improved Object-Relational Mapping Technology[J]. Coastal Education and Research Foundation,2019,98(sp1).
9. 蹇常林.ORM在Django操作数据库中的应用[J].技术与市场,2020,27(01):56-57.
10. Liu, Qigang,Sun, Xiangyang. Research of Web Real-Time Communication Based on Web Socket[J]. ProQuest,2012,5(12).
11. Jean-Marc Andre,Ulf Behrens,James Branson,Philipp Brummer,Olivier Chaze,Sergio Cittolin,Diego Da Silva Gomes,Georgiana-Lavinia Darlea,Christian Deldicque,Zeynep Demiragli,Marc Dobson,Nicolas Doualot,Samim Erhan,Richard Fulcher Jonathan,Dominique Gigi,Maciej Gladki,Frank Glege,Guillelmo Gomez-Ceballos,Jeroen Hegeman,Andre Holzner,Mindaugas Janulis,Michael Lettrich,Audrius Mecionis,Frans Meijers,Emilio Meschi,Remigius K. Mommsen,Srecko Morovic,Vivian O'Dell,Luciano Orsini,Ioannis. A Scalable Online Monitoring System Based on Elasticsearch for Distributed Data Acquisition in Cms[J]. EDP Sciences,2019,214.
12. 尚洁. 基于Django的测试工具平台的设计与实现[D].北京交通大学,2019.
13. 苏翔宇,朱爱群.CentOS7下基于Nginx的反向代理及负载均衡研究与实现[J].现代计算机(专业版),2018(10):61-64.

# 致 谢

时光荏苒，转眼间，大学四年的校园生活即将画上句号。曾记得，站在校门抚摸母校的大门，所有的奋斗和辛劳都已成为往昔美好的回忆，甜美的本科生活也终将尘埃落定。在这里，我要感谢培养教育我的佛山科学技术学院，母校浓厚的学术氛围和舒适的学习环境让我终生难忘。

在这里感谢对我鞭策鼓励、孜孜教导的诸位师长，尤其是本毕业设计论文的指导老师马莉副教授。马莉导师在我做本毕业设计与编写毕业论文给予了我细心的指导和不懈的支持。马莉老师严谨的学术态度和精益求精的教风亦将会是我终身学习的榜样。马导师温和性格、平易近人的人格魅力感染我、激励我，大学跟随马老师的学习生涯，让我受益匪浅。

最后感谢电子信息工程学院的各位老师和同学，在你们的关怀与指导下，我的大学生活称上幸福、开心的时光。

最最后感谢我的家人、朋友，在整个设计开发、写作过程中给予我鼓励、陪伴，无论经历多少学业上的挫折，都让我求学的道路上前进不独孤。希望我的努力，能让你们骄傲。

最最最后，感谢我的偶像邓紫棋，她就是我在黑暗中一直看见的光。