

简易版微博需求分析

一、系统概述

1.系统任务

1) 系统目标

1. 设计实现一个简易的类“微博”的社交系统，满足基本的发微博、查看微博、评论微博，用户管理等功能
2. 网页要求界面简洁，平衡恰当，元素对其，具有丰富的用户交互功能，操作符合用户直觉
3. 系统有足够的健壮性，能正确处理各种异常，错误
4. 使用合适的设计模式，合理使用并发、持久化、资源池等技术，架构合理，为未来功能和性能的扩展做相应准备

2) 运行环境

Windows、Linux、MacOS 等主流操作系统上的主流 Web 浏览器，开发环境本地使用可以不用联网，系统上线至生产环境后需要在联网环境下通过域名或 IP 地址访问。

2.需求规定

1.功能需求

1基本功能：

- 有用户注册/登陆/邮件密码找回等用户账号管理功能
- 微博列表，支持热点（如点击数最多/评论最多等规则）微博列表（所有用户可见）
- 可以查看微博内容及评论（所有用户可见）
- 注册用户可以发表微博，可以删除自己已发布的微博
- 注册用户可以对微博评论
- 注册用户可以关注其他用户

2.扩展功能：

- 有话题相关功能
- 增加回复及提醒功能
- 支持发布含有图片，表情的微博
- 搜索功能

2.数据要求

1.用户模型auth_user

- django自带用户模型
- 用户名
- 密码
- 邮箱

2.个人信息模型

- 用户
- 姓名
- 密码
- 个人介绍
- 性别
- 年龄
- 邮箱号
- 头像
- 关注
- 粉丝

3.微博模型

- 用户
- 用户姓名
- 微博内容
- 图片链接
- 发布时间
- 话题选择

4.评论模型

- 对应的微博
- 回复
- 用户
- 用户姓名
- 评论类型
- 发布时间

二、模块清单

1.微博模块

- 发布模块
 - 话题选择
 - 表情选择
 - 图片选择
- 列表模块
- 评论模块
- 回复模块

2.登录注册模块

- 注册模块
- 游客登录模块
- 注册用户登录模块

3.用户模块

- 微博列表
- 粉丝列表
- 关注列表
- 资料模块
- 关注取关模块
- 消息模块

4.话题模块

- 话题列表

5.搜索模块

- 结果列表
- 搜索框模块

6.背景模块

- 背景头部
- 背景底部

三、模块功能概述

1.系统层面

1.安全机制

- `authencate()`用户认证系统
- JSON确保数据在网络传输中不会被篡改

2.微博管理功能

发布微博，添加表情，添加图片，删除微博，返回顺序微博列表，返回热点微博列表

3.评论、评论回复功能

创建评论，回复评论，返回评论和评论回复列表

4.用户管理功能

创建用户，用户登录（游客和注册用户），用户信息维护，用户活动信息维护，消息的创建和分发

5.话题模块

对微博进行话题分类

6.搜索模块

可以搜索微博，话题，用户，并返回搜索结果列表

2.用户层面

1.登录注册

登录时需要填写用户名，个人简介，密码，邮箱号，性别，年龄

2.用户信息管理

注册时自动上传默认头像，个人主页和他人主页信息列表显示

3.关注和取关功能

登录的用户可以在他人主页关注和取关并显示

4.微博功能

登录的用户可以发布微博，需要填写微博的内容并选择话题，可以删除自己发布的微博

5.评论和回复功能

登录的用户可以评论微博，可以对微博下的评论进行回复

6.微博列表

所有用户可以看到微博列表，只有登录用户才能发布微博

7.评论和回复列表

所有用户可以看到评论和回复列表，只有登录用户才能进行评论和回复

8.话题列表

所有用户可以看到话题列表，进入话题可以查看该话题下的所有微博

9.热点功能

微博列表可以按发布顺序排列，也可以按评论回复数目进行热点排序

10.搜索功能

登录用户可以在搜索框中搜索出相关的用户，话题和微博

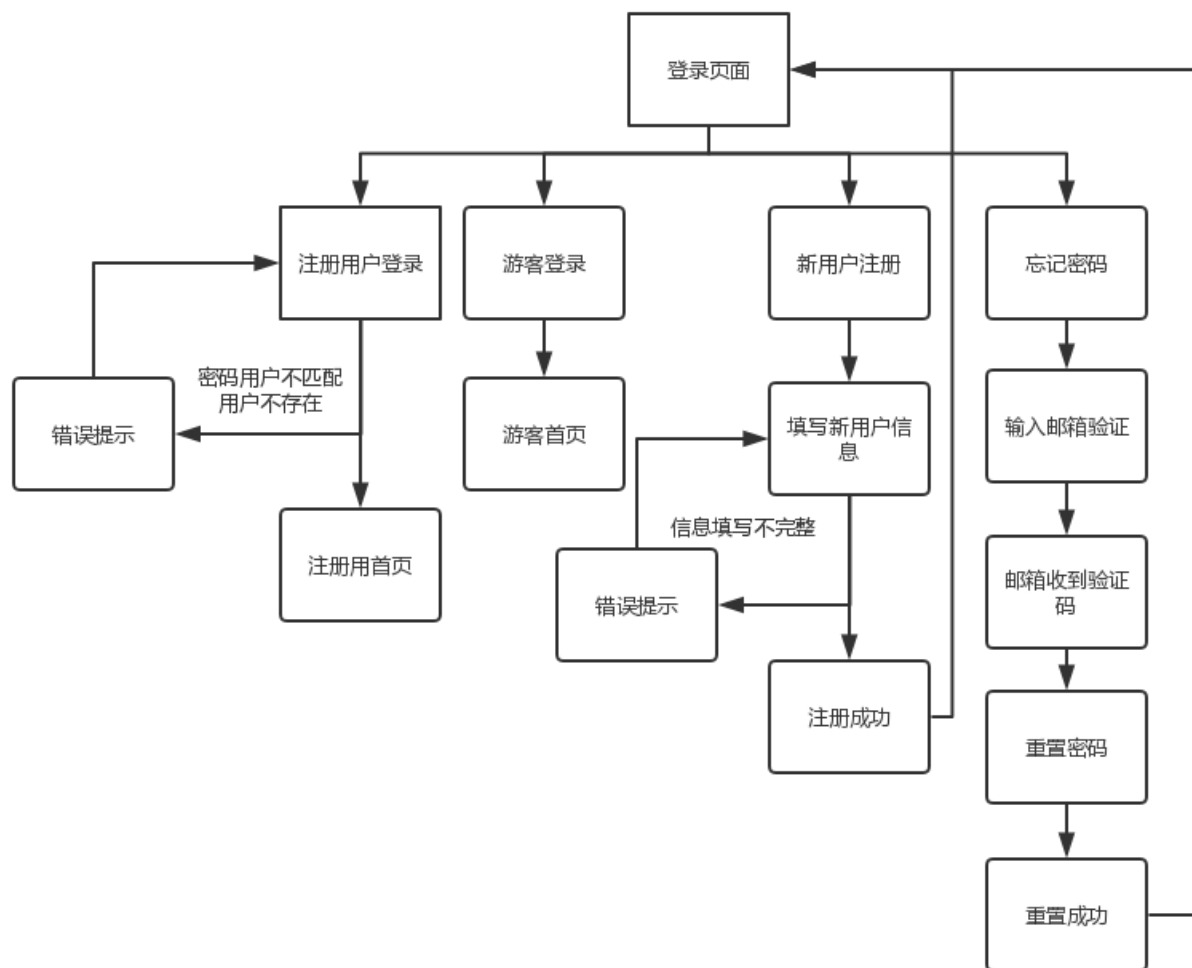
11.消息推送

用户会收到自己发出的微博的评论的消息推送和自己评论的回复的消息推送

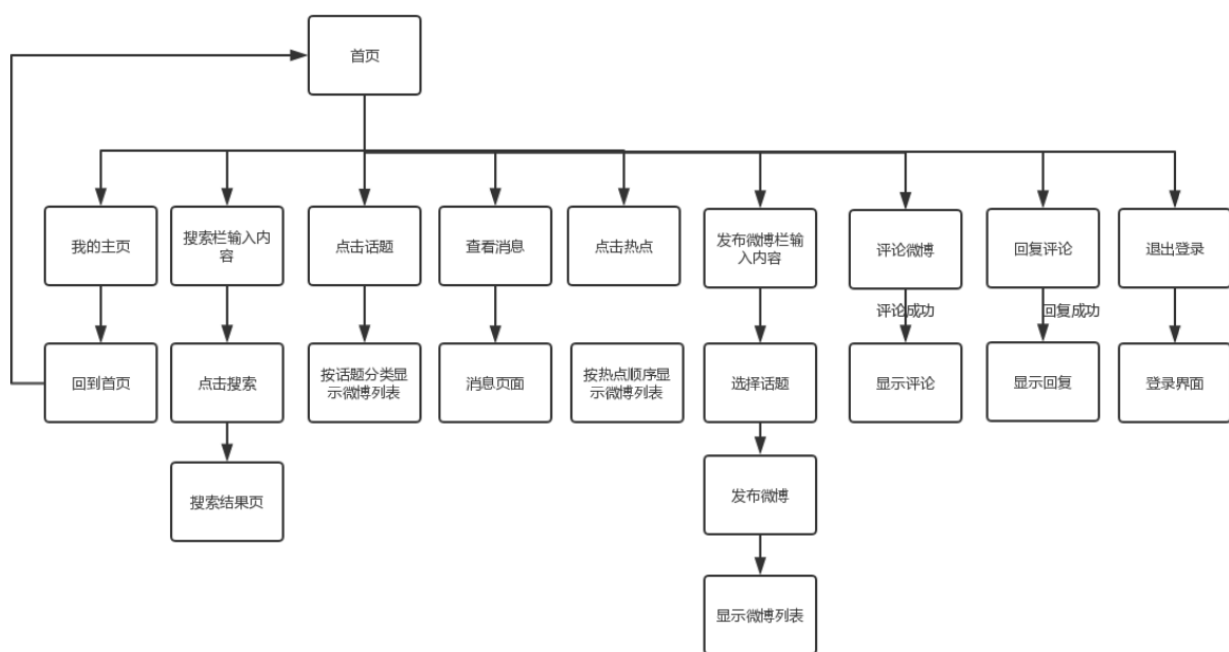
四、程序界面

1.交互流程图

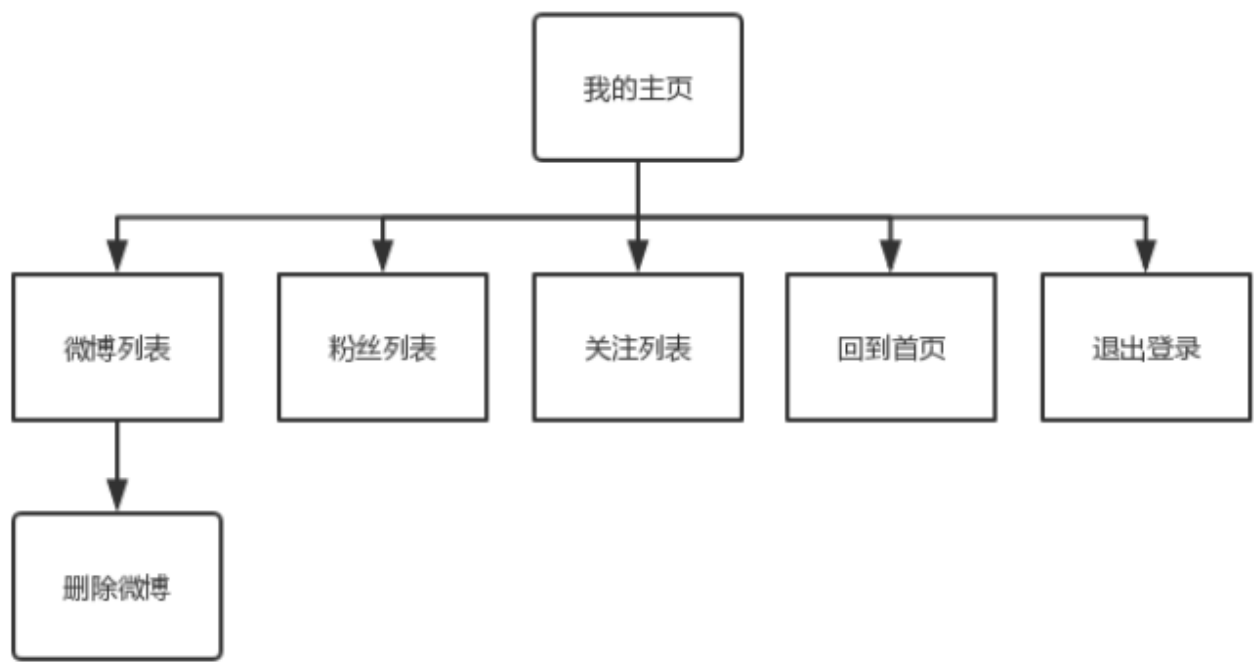
交互流程图1：



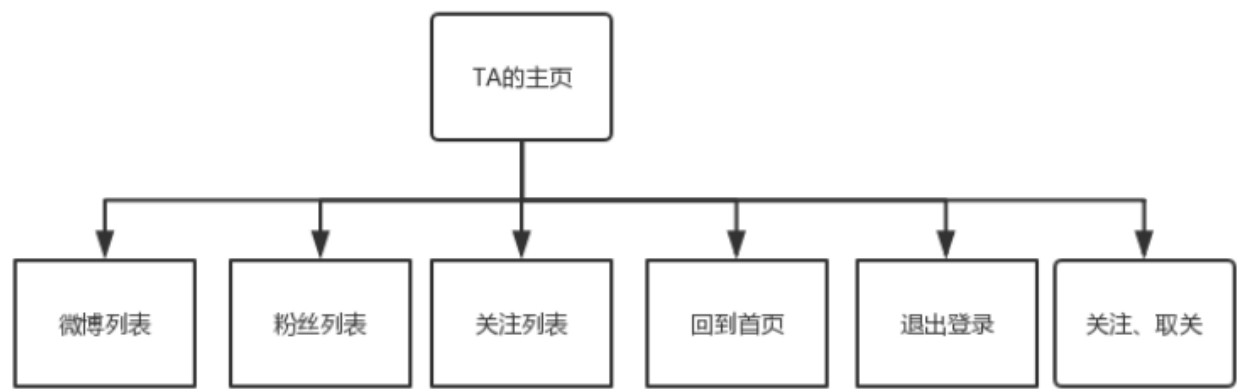
交互流程图2:



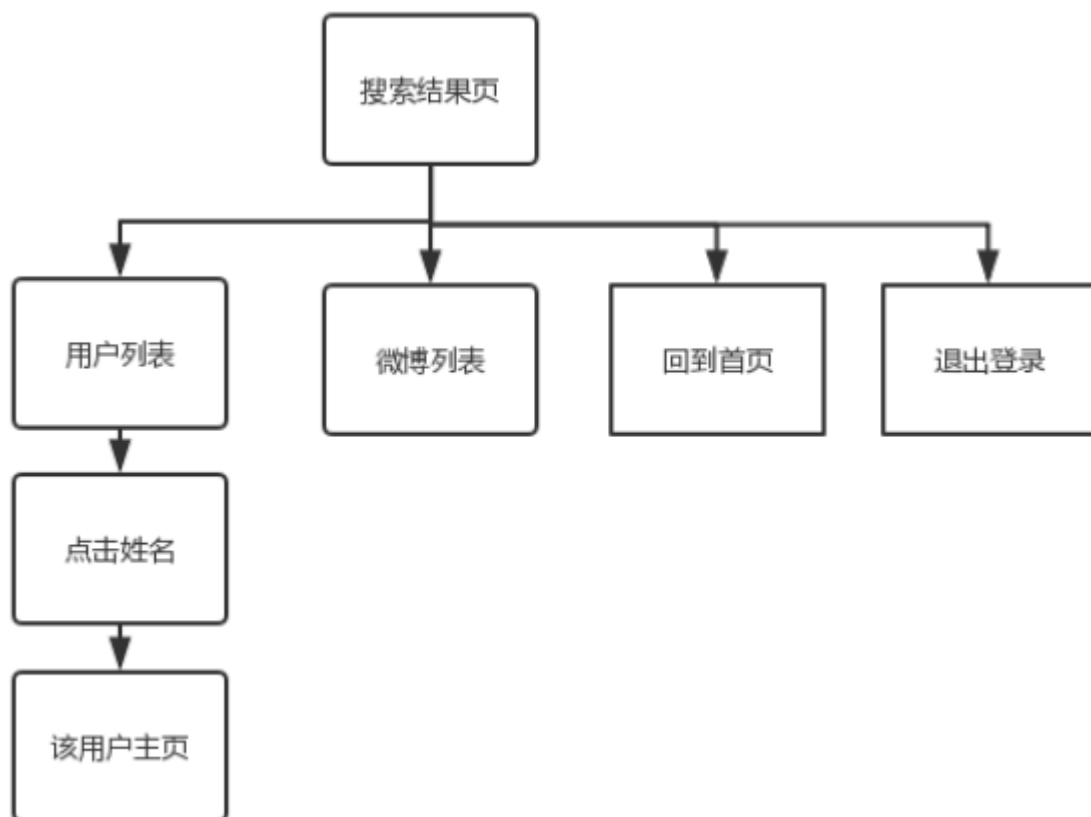
交互流程图3:



交互流程图4:



交互流程图5:



2.页面设计图

页面设计图1:

The screenshot shows a web browser window with the address bar displaying `http://127.0.0.1:8000/login`. The page content is as follows:

用户登录

用户名:

密码:

[新用户注册](#) [忘记密码](#)

页面设计图2:



页面设计图3:



页面设计图4:



页面设计图5:



五、技术架构后端程序

- 后端程序
 - 1. 编程语言: Python3.6
 - 2. 服务器应用框架: Django
 - 3. 数据库选择: MySQL
 - 4. 数据持久化存储: Redis
- 前端程序
 - 1. 编程语言: HTML5 + CSS3 + Javascript + JQuery + JSON + AJAX
 - 2. 前端框架: HTML
 - 3. 打包编译工具: webpack