

# Social Web Application Development Based on Data Analysis and Data Mining

Xinlu Du, Zecheng Du, Xingyu Wang, Dongzhe Zhou

North China University of Technology, Beijing  
Email: 943055294@qq.com

Received: Nov. 28<sup>th</sup>, 2017; accepted: Dec. 7<sup>th</sup>, 2017; published: Dec. 14<sup>th</sup>, 2017

---

## Abstract

With economic development, people's living standards improve, and travel has become an important way for people to relax and entertain. In order to make people have a better travel experience in unfamiliar cities, this paper aims to implement an application system for convenient travel and social sharing based on data crawling and analysis. The system includes a web terminal and a mobile application terminal.

## Keywords

Web Development, Data Crawling, Travel Social Applications

---

# 基于数据分析与数据挖掘的 社交类Web App开发

杜新禄, 杜泽成, 王星宇, 周东喆

北方工业大学, 北京  
Email: 943055294@qq.com

收稿日期: 2017年11月28日; 录用日期: 2017年12月7日; 发布日期: 2017年12月14日

---

## 摘要

随着经济的发展、人们生活水平的提高, 外出旅行成为了人们放松娱乐的重要方式。为了让人们在陌生的城市有更好的旅行体验, 本论文旨在实现一款基于数据爬取、分析的关于便捷旅行以及社交分享的应用系统, 系统包含web端以及手机应用端。

## 关键词

Web App开发, 数据爬取, 旅行社交应用

Copyright © 2017 by authors and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## 1. 引言

随着经济的发展、人们生活水平的提高,外出旅行成为了人们放松娱乐的重要方式。为了避免人们在陌生的城市将过多的时间花费在寻找感兴趣的景点、美食上,我们决定开发一款旅行社交应用,用户可以通过关键词搜索来获取第三方网站的数据资源,我们进行汇总以及数据分析来分出优先级然后推送给用户,帮助人们做出合理的出行、游玩规划,同时应用用户之间可以互相展示、分享旅行经历。

## 2. 系统设计与实现

### 2.1. 系统结构设计

本论文实现的应用系统分为两个模块,web端以及移动应用端。

在web端我们采用的是python + flask框架的实现模式,主要功能是实现根据用户需要进行数据搜索、筛选、推荐以及基本的社区功能(发表游记、评论、点赞等)。

本系统应实现的基本功能有:

- 1) 用户在系统中注册、登录。
- 2) 用户可编辑并查看自身资料。
- 3) 用户可以关注其他用户或文章并管理关注列表。
- 4) 查看联系人资料。
- 5) 用户发布、浏览、收藏、评论文章。
- 6) 软件功能分布图如图1,web端、服务器端、移动端实现功能如图1。

### 2.2. 系统功能

- 1) 用户注册、登录。

用户的账户是系统表示用户身份的唯一标准,用户通过自身身份验证才能执行权限内操作、添加好友、进行会话等。用户首次使用需要进行注册,系统将用户的注册信息添加于数据库,此后用户可根据用户名及其密码进行登录,通过系统审核确认身份无误后,用户获得账户信息和权限进行操用户可以在本平台浏览、发布、收藏有关旅游的游记文章。还可以关注其他用户留言交流实现社交的功能(图2)。

- 2) 用户资料列表

用户资料列表用于存储用户的个人信息,如账户名,密码等基础信息,并允许进行修改等基本操作。

- 3) 用户关注列表

用户关注列表用于将用户关注的其他用户或文章进行标记并存储。

- 4) 文章列表

文章列表用于存储每一位用户发布的文章,评论。

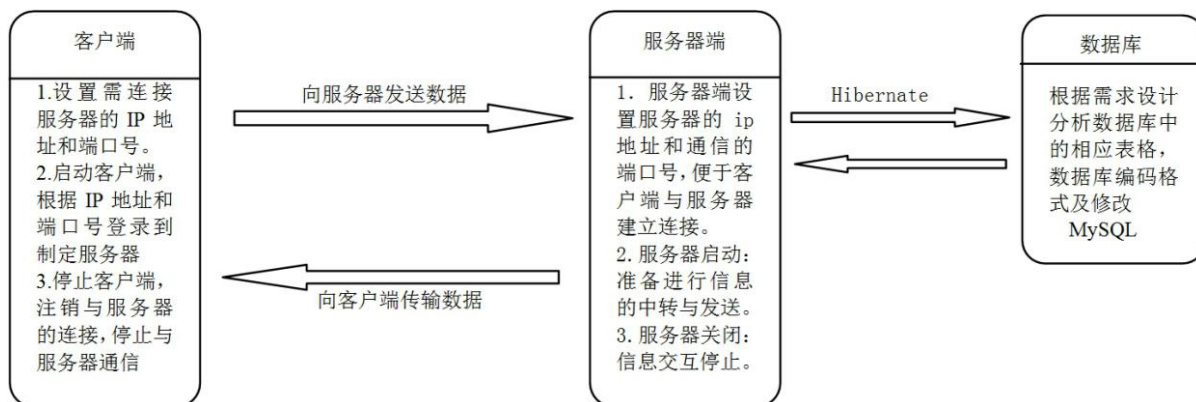


Figure 1. Details of client and server of system

图 1. 系统客户端与服务器端具体功能

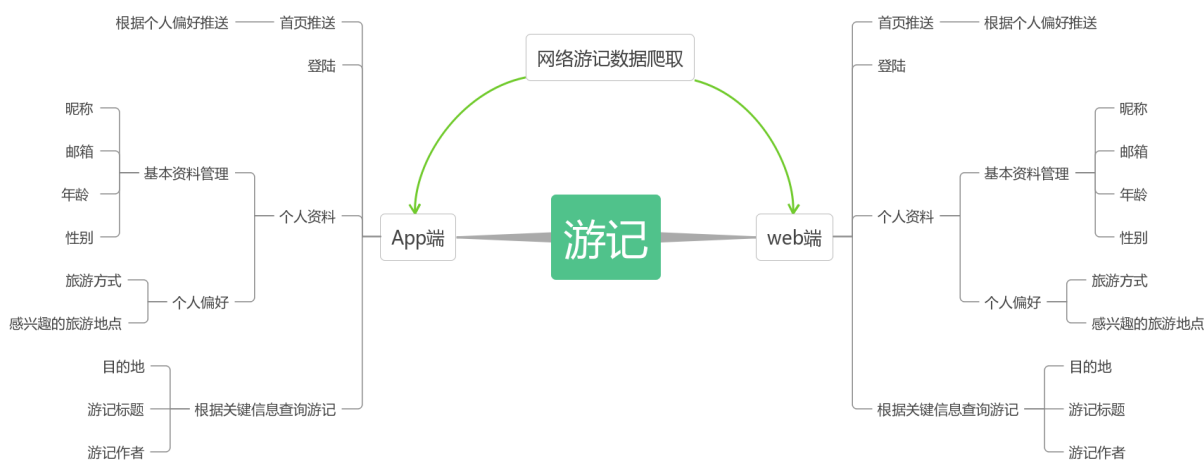


Figure 2. Software function distribution

图 2. 软件功能分布图

## 2.3. 系统模块与实现

### 2.3.1. web 端

web 端是用户的可视化操作应用层, 其视图应美观简洁, 功能齐全, 布局结构合理; 其作用应在于为用户提供浏览、阅读、收藏、评论文章等功能, 与服务器进行数据交互, 为用户提供流畅的、稳定的体验我们采用的是 Python + flask 的实现模式, 采用 mvc 架构(图 3)。

### 2.3.2. 服务器端

Flask: 基于 Python 的服务器应用。

Nginx: 负载的均衡代理服务器。

服务器端分为处理程序运算请求的服务器以及保存数据的数据库两部分。其中服务器起到了执行系统逻辑运算的重要作用, 服务器的正常运行是整个系统稳定运行的关键。服务器采用分块分层、保留接口的设计思想, 保证了块与块之间分别执行不同的功能, 互不干扰, 通过接口来实现数据传递, 减少代码的重复率, 提高了日后对服务器维护升级的效率。

### 2.3.3. 移动端

同样作为用户的可视化操作应用层, 具有与 web 端相同的功能, 用户可以在手机上进行操。作为系

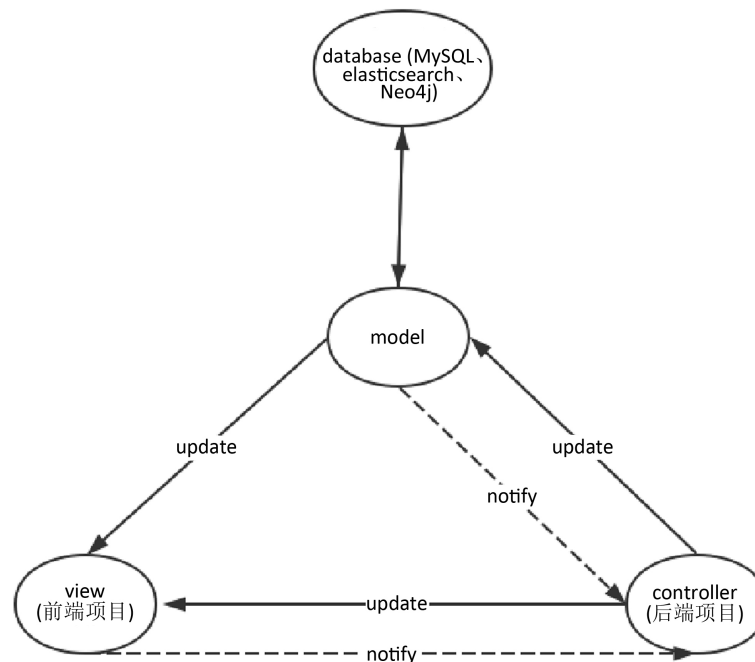


Figure 3. Front-end architecture diagram  
图 3. 前端架构图

统客户端是用户的可视化操作应用层，其视图应美观简洁，功能选项清晰易查，布局结构合理，与服务器进行数据交互，保存本地资源，并保证运行时的流畅性和稳定性。

## 2.4. 关键技术与算法

### 2.4.1. 前端的关键技术与算法

在前端，我们采用了 MVC 框架的软件设计模式。前端网页设计采用 Bootstrap 框架通过 Html，CSS 进行页面布局，并使用 jQuery 与 JavaScript 进行控制部分功能。

**MVC 框架：**是一种采用模型——视图——控制器的方式进行 web app 开发的设计典范。在开发中，我们将逻辑，数据，视图分离进行编程，使得程序在进行优化时能有更加清晰的优化思路，只对需要优化的部分进行修改即可。

**Bootstrap 框架：**bootstrap 框架是一种基于 HTML，css，JavaScript 的前端框架。通过 bootstrap 框架我们可以更好地设计网页的基本元素与布局，如导航栏，字体，图标等基本元素以及通过栅格布局控制网页在不同的大小中能正常的显示网页的布局。

**jQuery：**jQuery 在前端中起到连接作用，用于接收网页所接收到的变量并将他传导至后台。

**JavaScript：**是对网页部分动画效果的控制，如在主页中对不同模态框的访问进行速度的优化，使得跳转不会显得僵硬，是对用户体验优化的主要手段。

### 2.4.2. 服务器端关键技术与算法

在后端，我们选择使用 Python 语言来实现各个功能。主要应用技术有：flask, Nginx, Gunicorn, jinja2, Scrapy 等。

**Flask：**基于 python 的服务器应用。

**Nginx：**负载均衡代理服务器。

**WSGI: (Web Server Gateway Interface):** 作为容器。Web 框架 Flask 的作用是生成 HTML 代码，而

web 服务器(在本应用中即 Nginx)的作用是相应客户端发来的 Http 请求。WSGI 协议即统一 Web 框架和 Web 服务器的接口。在本应用中, WSGI 容器词采用 Gunicorn。因为 Gunicorn 有易安装、易启动、易维护的优点。

Scrapy: Scrapy 是一个为了爬取网站数据, 提取结构性数据而编写的应用框架。应用在包括数据挖掘, 信息处理或存储历史数据等一系列的程序中[1]。其最初是为了页面抓取(更确切来说, 网络抓取)所设计的, 也可以应用在获取 API 所返回的数据(例如 Amazon Associates Web Services)或者通用的网络爬虫[2]。

Gunicorn 安装/启动: Gunicorn 的安装和启动都十分简洁。本应用中, Gunicorn 包部署在虚拟环境 (Virtualenv)中。

#### 1) 安装

```
# pip install gunicorn
```

#### 2) 启动

```
# gunicorn -w 4 -b 127.0.0.1:8080 wsgi:application
```

其中, gunicorn 是启动命令, w 参数指明了开启几个进程来运行 flask 应用, 本应用中采用四个进程同时运行的方式来支持多个访问。b 参数指明了监听的主机地址和端口, 命令的最后跟上文件名: flask 实例的格式, 即可运行 flask 应用。

.Nginx 安装 / 启动

1) 下载 Nginx: # wget http://nginx.org/download/nginx-1.6.2.tar.gz

2) 解压安装包: # tar zxvf nginx-1.6.2.tar.gz

3) 进入安装包目录: # cd nginx-1.6.2

4) 编译安装:

```
./configure--prefix=/usr/local/webserver/nginx--with-http_stub_status_module--with-http_ssl_module--with-pcre=/usr/local/src/pcre-8.35
```

```
# make
```

```
# make install
```

5) 查看 nginx 版本

```
# /usr/local/webserver/nginx/sbin/nginx -v
```

6) 配置和启动

将 Nginx 的配置文件(路径为: /etc/nginx/site-avalidable/default)中内容修改为如下:

```
server {
listen 80;
server_name 127.0.0.1; #服务器 ip 地址 / 域名
location / {
    proxy_pass http://127.0.0.1:8080; # gunicorn 服务所在位置
    proxy_set_header Host $host;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
}
}
```

如上操作之后, 即可启动服务:

```
#/usr/local/webserver/nginx/sbin/nginx
```

Nginx 的安装/启动:

各层之间关系(图 4):

请求过程:

Nginx 监听到客户端的请求, 根据 Nginx 的路由配置将客户端请求转发至 Gunicorn。

Gunicorn 根据响应配置变量, 并请求 Flask 框架。

Flask 根据变量和请求利用 Scrapy 框架获取需要的信息并生成 Html, 作出响应。

### 2.4.3. 移动端

移动端是部署在移动设备上的用户可视化操作端口。使用 jQuerymobile 编写, 将对 web 端进行精简优化移植到移动设备中作为 app。其开发采用响应式设计以及渐进式设计, 能够兼容多种终端, 自动适应不同设备。

1) 具体应用技术与核心类(UML 用例图, 图 5)的构建:

2) 推送内容处理

TravelMsgEntity 类中主要包含了推送信息的各种方法, 通过搜索进行关键字匹配, 或者根据用户选取的爱好进行分类, 同时对于推送的文章根据热度进行优先级的排序[3]。

3) 社交管理

用户在每次登陆系统时, 服务器端都会发送一个最新的联系人列表至客户端。服务器发送的联系人数据格式统一, 因此只需按顺序进行处理, 得到全部联系人。客户端对联系人进行按照姓名全拼先后的自主排序, 或通过用户的主动分组来产生便于操作的联系人列表, 点击联系人可以查看用户的空间展示信息。

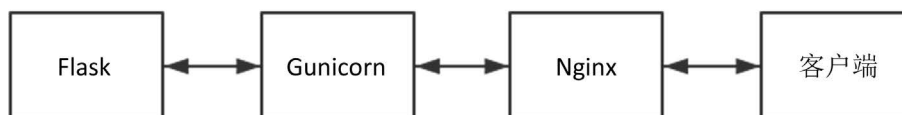


Figure 4. Server-side flowchart

图 4. 服务器端流程图

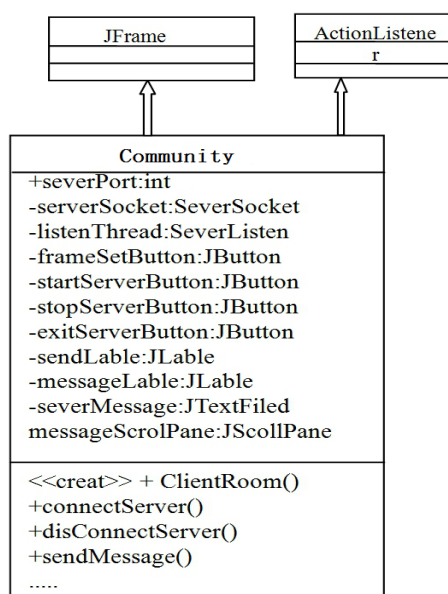


Figure 5. The UML of Community

图 5. Community 的 UML 结构图

4) 游记、文章管理

包含了不同消息形式的接收方法，其中有文字、图片、语音、文件、URL 连接的形式。不同的形式对应着不同的 Java 基本数据类型，通过本类处理后，再与不同功能的界面相结合，实现客户端游记的展示。

### 5) Intent 的使用

系统包含四个不同的层次级功能，而且要求可以使用户灵活地在不用层次之间快速切换，因此使用了 Intent 来整合了不同的功能界面。Intent 对象可以传递给 Content.startActivity()方法或 Activity.startActivityForResult()方法来启动 Activity 或让已经存在的 Activity 来执行其他任务。Intent 对象也可以作为参数将信息返回给调用 startActivityForResult()方法的 Activity。通过这种切换处理机制，保证了用户在不同层次上切换功能的灵活性和使用本系统的流畅性。

## 3. 测试

### 3.1. 测试环境

硬件环境介绍：

#### 1) 服务器端：

处理器：Intel(R)Core(TM)i5-4210M CPU @ 2.60 Hz 2.60 GHz。

已安装内存(RAM)：4.00 GB。

系统类型：64 位操作系统，基于 x64 的处理器。

软件部署：服务器安装了 MyEclipse、TomCat、MySQL。

#### 2) 客户端测试环境：

测试为了模拟真实的软件使用，即多用户同时使用本系统，因此将客户端部署在了三台 Android 手机上，硬件配置如表 1、表 2 所示。

### 3.2. 测试方案

服务器、客户端均进行合理部署与调试后，主要观察系统如下运行情况：

- 1) 客户端与服务器之间数据通信是否正常。
- 2) 服务器能否正确解析数据。
- 3) 服务器运算逻辑是否正常。
- 4) 服务器与数据库的连接情况。
- 5) 服务器对多客户端请求的承受能力。
- 6) Web 端业务访问及逻辑实现是否正常。

### 3.3. 测试结果

测试结果如表 3 所示。

### 3.4. 测试结果及分析

在测试中，三台 Android 手机上的客户端软件运行流程，web 端运行正常流畅。在服务器端启动服务后，客户端“登陆”后，客户端与服务器端数据传接正常。客户端通过服务器向其他客户端发送数据正常。客户端与服务器连接良好；服务器端能良好地接收、解析客户端数据，并准确地进行逻辑运算，在多客户端请求中表现出了稳定的性能和效率，我们认为系统很好地符合了预期。

**Table 1.** Hardware Configuration

**表 1. 硬件配置图**

|      | 测试机 1                                       | 测试机 2                 | 测试机 3                 |
|------|---|-----------------------|-----------------------|
| 处理器  | 8 核, 1.8 GHz                                | 4 核, 2.15 GHz         | 4 核, 2.35 GHz         |
| 运行内存 | 2 GB  | 3 GB                  | 6 GB                  |
| 内核版本 | 3.10.30-00015-g049a08f<br>android@localhost | 3.18.20-perf-gfavbe79 | 3.18.20-perf-g2f72a9d |

**Table 2. Mobile system version****表 2. 移动端系统版本**

|       | Android 版本   |
|-------|--------------|
| 测试机 1 | 6.0.1MXB48T  |
| 测试机 2 | 6.0.1MXB48T  |
| 测试机 3 | Android4.4.2 |

**Table 3. Test results****表 3. 测试结果**

| 功能    | 登陆 | 注册 | 发表 | 关注/收藏 | 评论/留言 |
|-------|----|----|----|-------|-------|
| 测试机 1 | √  | √  | √  | √     | √     |
| 测试机 2 | √  | √  | √  | √     | √     |
| 测试机 3 | √  | √  | √  | √     | √     |
| Web 端 | √  | √  | √  | √     | √     |

## 4. 结语

本系统实现了在 Android 平台上的企业内部即时通讯方案, 功能包括会话、资源共享、文件互传, 客户端与服务器的数据传输采用了 JSON 封装的形式; 后台服务器转接信息; 数据库表实现了三范式的设计标准; 在测试过程中, 本系统运行稳定, 性能良好。本系统的实现具有如下良好性质: 功能满足了对即时通讯的需求; 极大地方便了企业内部员工的沟通, 保障了企业内部信息的安全性, 私密性; 层次清晰, 逻辑简明, 易于开发人员的更新维护。本文详细而准确地介绍了即时通讯类应用程序的开发过程和框架设计, 希望能为同类应用的开发起到抛砖引玉的作用。

## 基金项目

北方工业大学大学生科技活动资助项目。

## 参考文献 (References)

- [1] 钱程, 阳小兰, 朱福喜. 基于 Python 的网络爬虫技术[J]. 黑龙江科技信息, 2016(36): 273-273.
- [2] 姜彬彪, 黄凯林, 卢昱江, 等. 基于 Python 的专业网络爬虫的设计与实现[J]. 企业科技与发展, 2016(8): 17-19.
- [3] 威利娜, 刘建东. 基于 Python 的简单网络爬虫的实现[J]. 电脑编程技巧与维护, 2017(8): 72-73.



**知网检索的两种方式：**

1. 打开知网页面 <http://kns.cnki.net/kns/brief/result.aspx?dbPrefix=WWJD>  
下拉列表框选择：[ISSN]，输入期刊 ISSN：2325-2286，即可查询
2. 打开知网首页 <http://cnki.net/>  
左侧“国际文献总库”进入，输入文章标题，即可查询

投稿请点击：<http://www.hanspub.org/Submission.aspx>

期刊邮箱：[sea@hanspub.org](mailto:sea@hanspub.org)