

# 南京理工大学计算机科学与工程学院

## 软件工程专业毕业实习报告

班 级\_\_\_\_\_

学生姓名\_\_\_\_\_

学 号\_\_\_\_\_

实习时间\_\_\_\_\_

指导教师\_\_\_\_\_

南京理工大学计算机科学与工程学院制

## 一 毕业实习简况

### 1.1 实习时间和地点

实习时间：8.26-9.12

实习地点：

- 华为软件技术有限公司，雨花台区软件大道 101 号
- （三星南京软件研究所，楚翘城）
- 南京理工大学，玄武区孝陵卫 200 号

### 1.2 实习目的

（1）掌握在计算机科学与技术专业下，合理选择和应用专业软件技术，高级编程语言，应用程序设计调试等的综合能力；

（2）熟练使用多种开发环境，团队协作完成要求系统的设计、开发、实现及调试流程；

（3）掌握社会及行业内对计算机科学与技术人才的专业知识的实际要求，了解和关注专业发展的趋势和重点，适应社会对人才的专业技能的要求。

### 1.3 任务及要求

任务：结合华为 HiAI 开发一个快应用；

要求：基于 HiAI 的美学评分和场景识别、图片分类功能开发美学导向的图片兴趣社区，并包含以下功能：

- 用户个人账户管理，包括注册、登录和修改个人信息等；
- 用户可以创建动态，也可以为其他动态点赞、评论、转发等；
- 系统综合美学评分和点赞数、评论数对图像进行排序，实现排行榜功能；
- 系统针对不同用户进行用户推荐、图像推荐和标签推荐；
- 系统通过场景识别和图像分类对每张图像进行标签，从而支持同质图像的搜索、推荐等功能。

## 二 项目实习技术总结

### 1 概述

针对项目任务及要求，我们完成了 Stack——这一由美学导向的社区快应用的需求分析、设计与架构、编码与测试的全部过程。Stack 社区不仅支持基本的用户管理

和动态管理功能，还支持我们基于华为 HiAI 技术设计排行榜和图像、用户、标签推荐和搜索功能。

该快应用使用了基于前后端分离的分层的软件体系结构，以及基于多云服务器的应用程序、文件、数据库分离服务器架构。其在前端通过华为快应用 IDE 完成代码编写，使用华为快应用加载器完成应用测试；在后端则根据工程师的技术特点完成了 API 分离式的基于 Flask 框架以及基于 Spring Boot 的并行开发，并分别部署于华为云和阿里云轻量服务器中。数据库部分使用华为云数据管理服务对基于华为云关系型数据库 MySQL 实例完成了数据库的可视化操作，并支持高可靠性高安全性的远程连接。此外，使用了对象存储完成了图像为主的文件系统的搭建。

在完成了架构设计、模块划分和代码编写后，我们完成了一系列软件测试，证明了该快应用具有一定可靠性，符合需求预期。

## 2 需求分析

### 2.1 功能需求分析

期望实现注册功能，使用邮箱或手机号验证，注册完成将用户信息存入个人信息数据库，用户登录时，向后端数据库验证用户名密码，验证成功即可登录，登录后可以在个人信息界面中查看自身的信息，查看自己上传的图片，修改自身信息，查看自己的具体有哪些粉丝。流程如图 2.2.1。

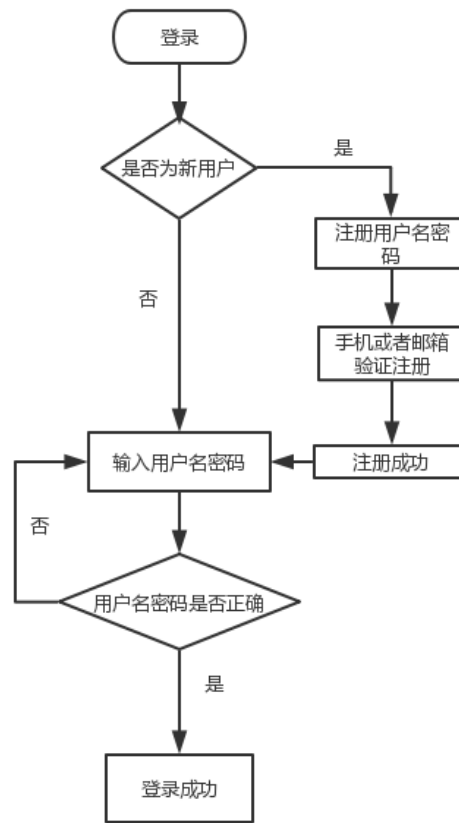


图 2.2.1 用户登录流程

期望实现社区功能，浏览上传的图片，对图片进行评论，或者回复他人评论，也可以对于图片进行点赞。

实现图片排行，包括日榜和周榜。榜单内以评分高低为一日内一周内动态内图片排序，只显示分数最高的前十张图片。这里的评分根据美学评分和点赞数、评论数综合计算。美学评分需要调用 AI 进行评分，在将得到的分数进行排行，在用户每次上传图片的时候，对上传图片进行美学评分，并实时更新美学评分排行榜；点赞数量由社区用户进行点赞，统计数量，更新排行榜。

实现发现的界面设计，主要包括搜索栏以及照片墙，照片墙上推荐的是数据库中权重大于一定阈值的图片（主要考虑排行榜有时限，这里的推荐可以推荐以前的图片）。照片墙只展示单张图片，当用户对照片墙中的某张图片感兴趣时可以点击该图片，随后进入单个动态子模块提供支持的动态详情界面，该界面显示单个动态的具体内容，用户也可以在此进行点赞转发和评论操作。然后上方搜索栏点进去

有账号推荐和标签推荐，然后有实时搜索功能。

实现发布动态的功能，设计一个发布动态的单独 tab，点进去可以从相册里选择图片，也可以把已经选择的图片删除，也能做点简单的编辑。

期望实现三个基础推荐算法，

- 图片推荐。期望实现发现页面，在发现页内随机推荐评分较高的若干图片，支持不断刷新。评分由某种算法进行计算。图片推荐是为了让排行榜以外的、一周以前的高评分图片被用户能够发现；
- 用户推荐。期望在某页面内显示为当前用户推荐的若干用户；
- 标签推荐。由于数据库内每张图片都具有相应标签，期望能够为用户推荐标签或相应图片。

期望实现搜索功能，支持用户搜索和标签搜索。

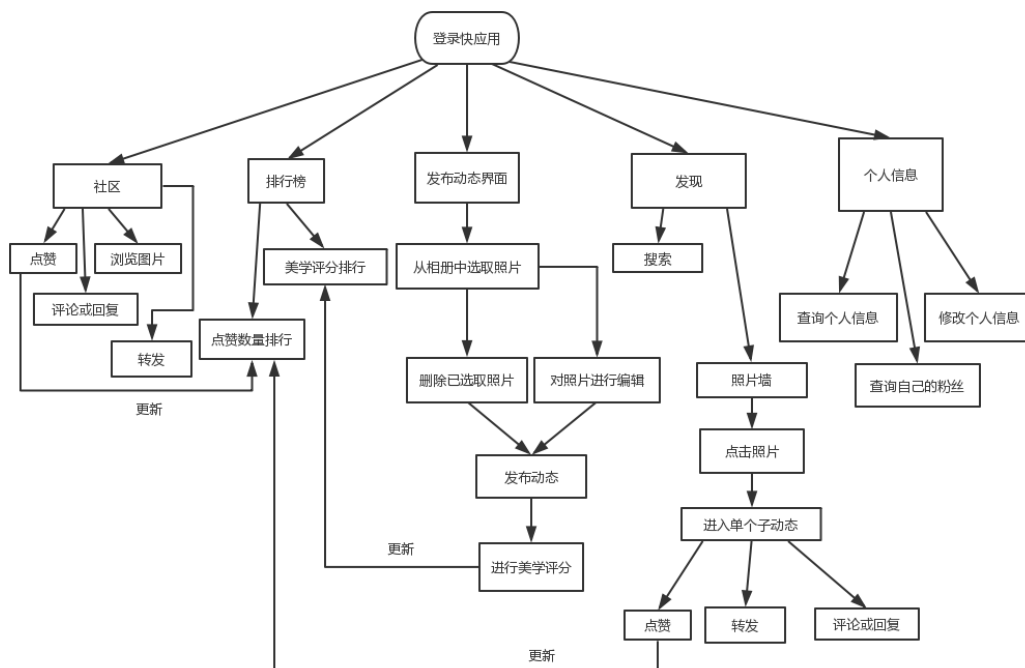


图 2.2.2 整体需求功能逻辑

## 2.2 对性能的需求分析

对应用性能需求主要的是使用频率较高，在应用中关键程度非常高的部分，以及资源占用非常严重的部分。性能需求主要包括，（1）响应速度：API 请求的平均时间，登录响应时间等（2）服务正常运行时能够支持的最大在线人数（3）应用中的某一个接口可以同时被多少用户调用（4）在用户在线人数超过预期的人数时，应用

提供的服务有多少可以继续提供(5)当实际的运行压力超出预期的与运行压力时,应用可以稳定运行多久(6)系统能支持多少虚拟用户(即每天登陆应用的人数)。

对于性能的具体需求主要依据以下几个方面获取:(1)由用户提出明确的对于性能的需求分析(2)依据已有的数据(日志文件)进行分析,得出具体性能需求(3)根据同类型服务应用的数据得出具体性能需求。

我们小组的快应用的需求主要是注册时,使用手机或者邮箱进行验证时,验证信息需要快速的发送给用户,登录时需要迅速地向数据库确认用户名密码,并给出登录成功或者用户名密码错误的信息;在使用服务时,对于接口的调用的请求的响应时间控制在一个很短的时间内;在用户上传图片之后,马上调用 AI engine 对于图片进行美学评分,更新排行榜;在多用户同时在线时,服务能够稳定地提供给每一位用户。

### 2.3 输入输出要求

输入设计的目标:保证向系统输入正确的数据。在此前提下,尽量做到输入方法的简单、快捷、方便。

输入内容就是用户上传的图片,输入方式是在社区中进行直接上传图片,输入的要求是在上传时,可以在手机相册中选取,也可以直接拍照上传,但是必须是图片的格式,而且再上传时,设备必须连接网络,否则会上传失败。

在进行输入之后,后端进行中间处理,调用 AI engine 对图片进行美学评分以及标签分类,为图片添加一个 URL,将图片以及图片的信息使用腾讯云对象存储 COS,依赖于腾讯云对象存储 cos 技术的断点续传、自定义对象头部、静态网站、数据迁移等功能高效便捷的完成图像存储系统的设计与搭建;对图片添加点赞接口、获取点赞数量接口(整合至时间线接口,该接口不能被调用)、取消点赞接口、评论接口、获取评论数目接口(整合至时间线接口,该接口不能被调用)、获取评论内容接口、取消评论接口。中间处理结束之后就是输出。

输出设计的重要性:评价管理信息系统是否能为用户提供准确、及时、适用的内容的标准之一。输出设计的内容:确定用户在使用信息方面的要求包括使用目的、输出速度、使用频率、安全性等要求。

输出的内容就是将经过中间过程处理的图片上传到社区,每一种功能图片会占据手机的全屏幕,以便用户可以清晰的看到全图,下部显示部分评论,点击可以查

看详细的评论；此时图片已经添加了部分接口，点赞与评论的数据也会通过接口传入图片数据库中。

## 2.4 模块要求分析

### 2.4.1 容错性

容错性设计原则：（1）提示与引导，包括尽可能详尽的引导与文字提示，且文字提示的内容简单易于理解，进行不可逆的操作时向用户询问是否继续；（2）限制操作，主要包括加大不可逆操作的难度，限制用户的某些不必要操作，减少出错可能（3）反馈与帮助，是指在错误发生后能够让用户及时反馈错误并提供纠错帮助，错误提示信息需要易懂且精炼（4）错误恢复，能够帮助用户在错误发生后快速回到安全状态。（5）减少负担，主要是减少用户不必要的操作步骤。



图 2.2.3 容错性设计原则

结合上述的容错性设计原则，容错性设计可以分为以下三个阶段：引导、报错、解决。

引导主要针对新手用户，让他们很快的了解设计出的应用的核心功能以及主要的操作方式，帮助他们更快的上手，当然引导功能的实现方式不仅限于此，输入框中的输入提示也是常见的引导样式。输入框是用户完成信息录入的主要途径之一，有录入才有报错，有报错才需要引导。引导归根结底是为了避免用户在操作过程中

犯错，而用户的操作过程又可以看成是不断做决策的过程，要想做出正确的决策必须要消减信息的不对称性。所以引导功能我们可以理解为就是消减信息的不对称性，让用户做出正确的决策。

报错就意味着引导失效，用户还是错误的使用了某一功能，报错主要分为报错方式以及报错时机，报错方式主要就是弹框，而报错的时机就比较重要了，有些错误需要请求数据库或者服务器才可以发现，但是一些简单的可以直接由前端界面进行判断的错误，如果依然滞后报错可能不太合适。

解决即向用户提供纠错的帮助，需要注意的是在给用户提供解决方案的时候，尽量提供跳转链接，方便用户操作。

我们的快应用中错误可能发生的场景包括：注册时的用户名重复，登录时的用户名或者密码错误，修改个人信息时，用户输入的手机号错误，各种需要使用网络的时候，网络连接错误，搜索过程中未找到搜索内容，还有就是可能存在的服务器超载问题等。

对于上述可能发生的问题都会进行反馈，用户根据应用给予的反馈，通过反馈的可能问题尝试解决。对于服务器的超载问题在不升级服务器的情况下，解决方案可以是（1）在服务器中直接形成请求的处理进程，不撤销处理进程，所有的请求成队列等待处理，这一解决办法的问题是响应速度慢，而且在有时间戳的情况下可能会请求失败（2）使用相对于进程而言开销较小的线程来处理请求。

#### 2.4.2 封闭性

开放封闭原则是所有面向对象原则的核心。软件设计本身所追求的目标就是封装变化、降低耦合，而开放封闭原则正是对这一目标的最直接体现。关于开放封闭原则，其核心的思想是：软件实体应该是可扩展，而不可修改的。也就是说，对扩展是开放的，而对修改是封闭的。因此，开放封闭原则主要体现在两个方面：对扩展开放，意味着有新的需求或变化时，可以对现有代码进行扩展，以适应新的情况。对修改封闭，意味着类一旦设计完成，就可以独立完成其工作，而不要对类进行任何修改。

对修改封闭的有两个例外：一是对于缺陷修复的改动，缺陷在软件中很常见，是不可能完全消除的。当缺陷出现时，就需要我们修复现有的代码。软件修复明显倾向于实用主义而不是坚持开放封闭原则；二是对客户端无影响的修改，对于客户端



无影响部分即松散耦合，这些类的变化是不会引起别的类的改变的，所以修改这些类对于用户来说没有直观的变化。

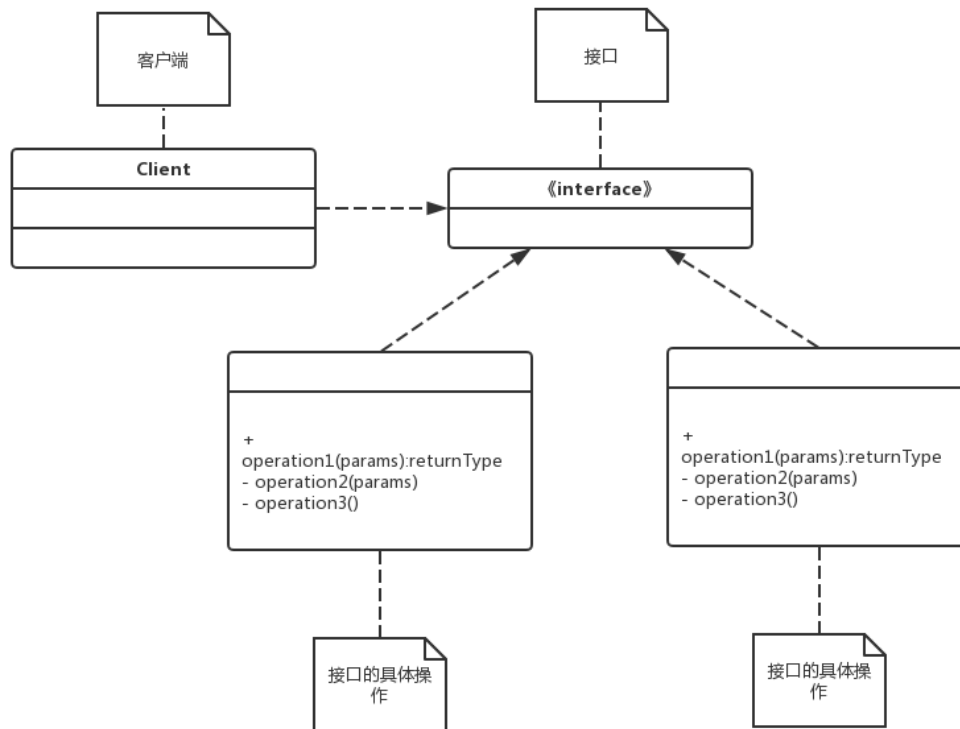


图 2.2.4 接口

在我们小组的项目中封闭性主要体现在了，对于用户所有能使用的功能，都提供了一个调用该功能的接口，用户的动作完全由接口来实现，每设计一个新的功能，就为该功能设计一个接口，供用户调用，而用户程序不需要任何的修改。对于上述的 8 个模块，每个模块的每一个基础功能都需要用户调用接口实现，模块的变化对于客户端没用影响。每当需要开发一个新的功能时，只需要完成功能实现代码以及一个调用的接口，在客户端直接调用即可。

### 2.4.3 安全性

安全性设计主要包括两方面，一是数据库的安全设计；二是接口调用过程中的安全设计。

数据库的安全性设计：数据库的不安全因素主要是（1）无授权用户对于数据库的恶意存取以及破坏（2）数据库中重要的信息的泄露（3）安全环境的脆弱。我们采取的措施有用户身份鉴别，存取控制，授权，加密存储。关于上述的第三个安

全问题主要是系统的安全问题，不作过多讨论。

为了应对无授权用户对于数据库的恶意存取以及破坏问题，主要采用的方法是用户身份鉴别以及存取控制，用户身份鉴别可以是静态口令鉴别，类似于设置一个密码，或者动态令牌的形式进行鉴别；存取控制机制主要包括定义用户权限，并将用户权限登记到数据字典中和合法权限检查。两者机制一起组成数据库管理系统的存取控制子系统。针对于数据库中重要的信息的泄露问题使用的安全方法主要是数据加密存储。

接口的安全性设计：在访问接口的过程中如果没有过滤，可能会出现安全性问题，例如通过模拟请求，对数据进行篡改，因此需要设计一个安全的接口调用方式。具体的方案可以是（1）使用 **token** 串验证，每当用户登录时，服务器生成一个唯一的 **token** 串给用户，在请求调用时，可以通过验证 **token** 串来确保安全性。（2）使用时间戳验证，每次请求发出是可以在请求上附上一个时间戳，在接收端设置一个检测如果超时，就拒绝请求（3）使用 **token** 串与时间戳相结合并进行加密的签名机制。接收端写一个过滤器，只有 **token** 串有效，时间戳未超时，缓存服务器中不存在签名同时满足，才应答此次请求。

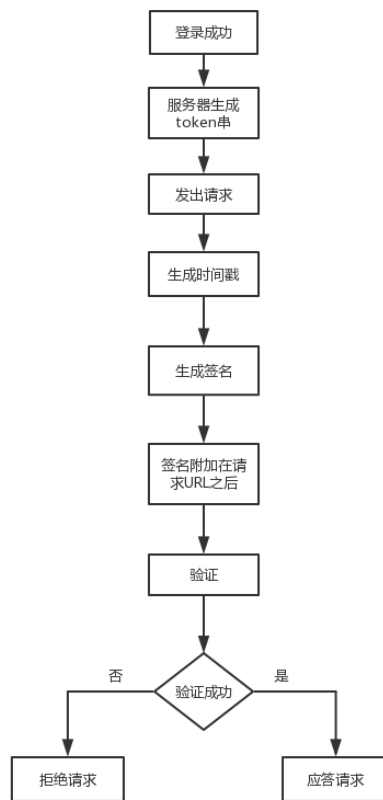


图 2.2.5 token 验证

### 3 面向复杂工程问题的系统设计

#### 3.1 概要设计

##### 3.1.1 总体设计

###### 3.1.1.1 系统概述

本系统是一个美学导向的图像兴趣社区快应用。支持用户发布动态、浏览动态，提供用户图片上传、下载、评论、点赞、转发、修改上传图片等功能，并能够调用华为的 HiAI 进行图片的识别分类，从而供用户按照标签进行搜索，以及美学评分，从而形成基于由美学评分、点赞数、评论数等计算出的权重作为导向的推荐算法，引导用户更好地融入社区，扩大社交圈。

###### 3.1.1.2 设计约束

- 体系结构设计人员需从文档的需求分析部分提取需求约束，保证设计的功能符合需求分析中提出的要求；
- 软件开发人员须遵守开发工具和系统运行环境的约束；

- 前后端开发人员在接口方面须遵循《接口文档》的约束，保证前后端接口的一致性；

#### 3.1.1.3 总体模块设计

基于对 Stack 社区项目的需求分析，结合前后端设计方案和功能性分析，将社区的主要功能拆分为个人模块、时间线模块、动态管理、点赞管理、评论管理、关注管理和推荐、搜索八个模块。用户可以通过快应用 APP（Stack 社区）使用这一系列模块提供的各类功能。

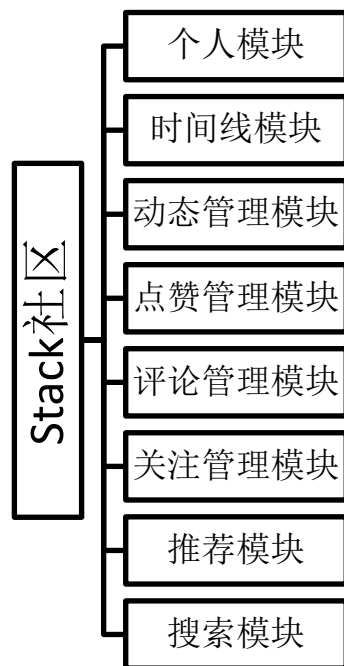


图 2.3.1 整体模块划分

#### 3.1.1.3 技术结构

由于系统采取基于前后端分离的多层架构，技术结构也相应地产生分离：

前端技术：HTML、CSS 和 Javascript 代码编写为主，开发人员须遵循华为的快应用开发文档，使用相应的组件和 API 在快应用开发框架中完成代码开发；

后端技术：

- 以 Java 代码和 Python 代码编写为主，开发人员须遵循模块设计部分文档，按照相应内容设计每个模块对应的 API。
- 数据库部分以 MySQL 代码编写为主，开发人员须根据需求分析部分设计数据库结构，绘制 ER 图，转化为相应的关系并完成表格和相应的约束。
- 服务器维护部分要求服务器维护人员通过 linux 指令完成符合要求的运行

环境的搭建。

3.1.2 架构设计

3.1.2.1 软件架构设计

Stack 社区的系统构架工作是整个系统设计阶段的重要内容，其中对系统软件体系结构和硬件架构的设计尤为重要，需要结合各类软件技术和硬件服务，围绕用户操作、业务逻辑处理和数据信息存储等重要内容进行设计。本平台的系统软件体系结构以分层模式为主体，将系统分成不同的抽象层次，每一层都为上层提供好用的 API，并屏蔽掉下层的细节。这里主要采用四层架构的方式进行实现，详细软件架构设计如图 2.3.2 所示。

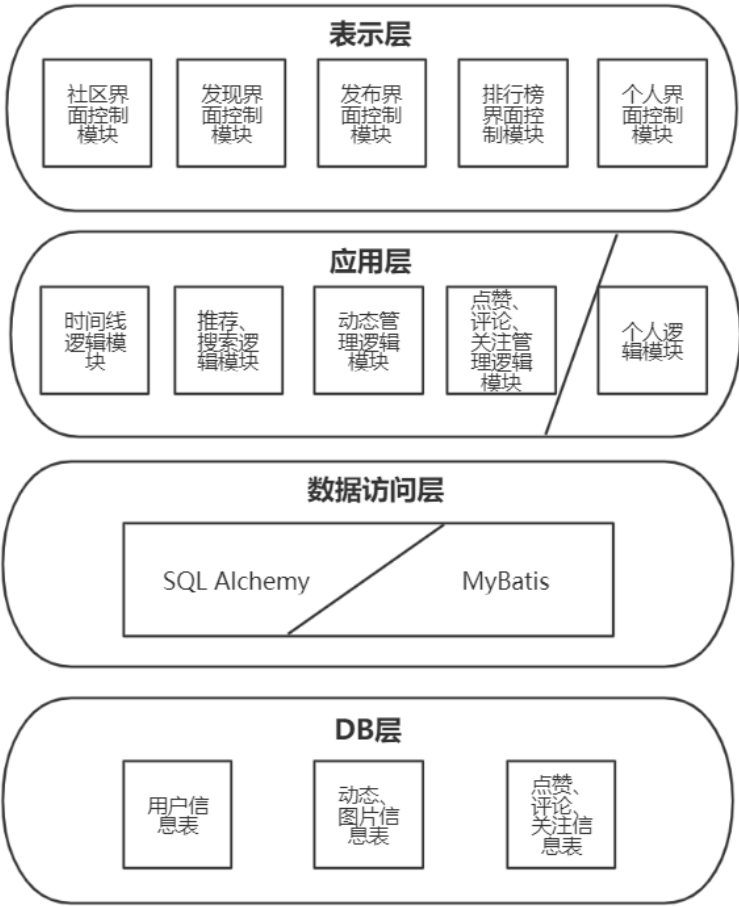


图 2.3.2 软件架构

从图 2.3.2 可以看出四层体系软件架构设计方面的内容，按照业务类型、功能类别、关联关系等方式，对系统每个逻辑层涉及的相关功能和服务进行模块设计，每一层操作的具体内容如下：

- 表示层

表示层利用 CSS 和 JS 等技术在设计的界面控制模块中的 index.ux 文件中完成代码编写，实现用户交互前端界面。交互界面主要包含主 Tab 中可自由左右滑动的五个控制模块：社区界面、发现界面、发布界面、排行榜界面以及用户个人界面。其中发现界面上方包含搜索栏，个人界面上方包含设置界面入口，各子界面都可通过这些主界面来回切入。

- 应用层

表示层和应用层的划分是前后端分离架构的体现，通过将前后端进行有效的解耦，系统才能被模块化，从而为之后的分布式框架、微服务框架等打下坚实的基础。因此表示层的 Javascript 代码除了负责部分界面模块的动态交互，还负责调用应用层在服务器后端提供的接口。后端接收前端发来的以 POST 表单提交为主体的 HTTP 请求，利用数据访问层提供的接口进行相应数据处理，并通过一定包装后将其以 JSON 格式再传回前端。

应用层的主体——服务器后端根据开发人员的特点也存在一定程度的分离。个人逻辑模块部分的 API 接口主要来自基于 Java 语言的 Spring Boot 框架，而其余部分的 API 接口则来自基于 Python 语言的 Flask 框架。

- 数据访问层

API 部分的分离也导致了数据访问层的分离。其中源于 Spring Boot 框架中的部分 API 通过 ORM Mybatis 将面向对象语言程序中的对象自动持久化到关系数据库中，而 Flask 中 API 则主要通过 SQL Alchemy 完成相应功能。尽管使用的插件不同，但二者都是通过对象关系映射，通过连接远程云关系数据库，通过使用 DB 层的接口完成一切增删查改的数据操作。

- DB 层

DB 层采用基于华为云的关系数据库 MySQL 存储数据表，向数据访问层提供数据支撑。具体情况将在数据结构设计中描述。

### 3.1.2.2 硬件架构设计

系统硬件架构是从物理部署的角度对系统的总体框架进行分析。而在云服务非常普遍的今天，综合经济成本、服务器性能、可靠性和安全性等方面考虑，将应用程序、文件和数据库分离，分别部署在独立的云服务器上。如图 2.3.3。

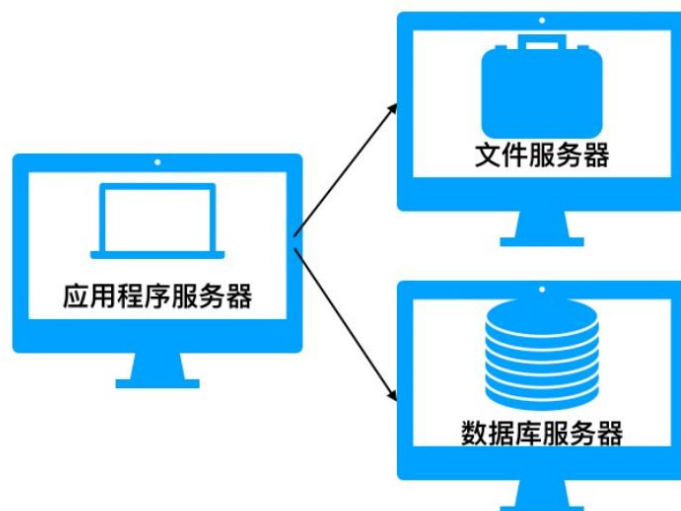


图 2.3.3 硬件架构

由于当前 Stack 社区处于开发的初级阶段，使用用户较少，因此暂未使用缓存、负载均衡、分布式等技术提升整体性能。

以上的基础式应用程序、文件、数据库分离服务器架构将这三个部分分别部署于不同的云服务器中：

- 应用程序服务器：
  - 基于 Java 语言的 Spring Boot 框架部署于阿里云的轻量应用服务器（Simple Application Server）中；
  - 基于 Python 语言的 Flask 框架部署于华为云的弹性云服务器（Elastic Cloud Server）中。
- 文件服务器：部署于腾讯云的对象存储（Cloud Object Storage, COS）中，基于其提供的支持 HTTP/HTTPS 协议访问的分布式存储服务实现社区图片文件的高效存储。
- 数据库服务器：部署于华为云的云数据库（RDS for MySQL）中，利用其高可靠性和高安全性的特点，实现了数据访问层和 DB 层的解耦合。

将以上基础式应用程序、文件、数据库分离服务器架构映射到具体服务器后架构图如 2.3.4。其中上传到对象存储在前端实现，将图像映射为唯一 URL 后通过后端云服务器上传到关系数据库，避免了大文件的二次上传。此外，通过各家云服务器运营商提供的云监控可以实时监测服务器使用情况，可以有效实现数据统计及基于此的服务器调控和流量预测等功能。

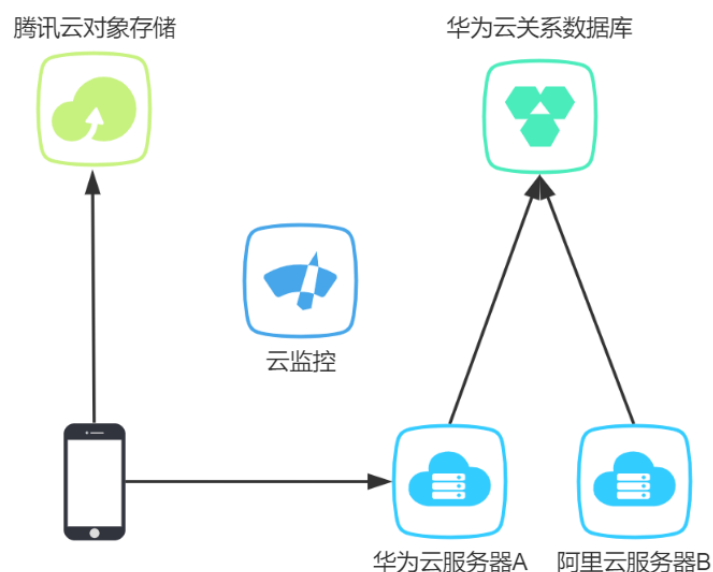


图 2.3.4 具体服务器架构

### 3.1.2 开发工具

- 前端部分：要求使用基于 VSCode 框架深度定制的快速应用 IDE——华为快应用 IDE；
- 后端部分：
  - 基于 Java 的 Spring Boot 框架，使用 jdk8 以上环境下的 IntelliJ IDEA 进行代码开发
  - 基于 Python 的 Flask 框架要求使用 Python 编程语言开发的集成环境 JetBrains PyCharm 进行代码开发；
  - 数据库部分要求使用华为云数据管理服务（Data Admin Service，简称 DAS）可视化地完成数据库部分操作。

### 3.1.4 系统运行环境

- 开发环境

表 2.3.1 开发环境

类别	标准 IDE 配置	标准环境配置
前端开发	DevecoFastAppIDE-V2.0.1	node.js10.16.3
后端开发	JetBrains	python3.6.5



PyCharm2018.1		
IntelliJ		Jdk8 tomcat9
IDEA2018.2.4		
数据库	DAS	MySQL5.7.25
<div>● 生产环境</div>		
表 2.3.2 生产环境		
类别	服务器系统	标准环境配置
后端	centos7.2	python3.6.5
	centos7.3	Jdk8 tomcat9
数据库	/	MySQL5.7.25
<div>● 测试环境</div>		
要求软硬件符合以下要求的手机，并安装有华为快应用加载器，标准版本为HwFastAPPEngine_Loader_Phone_V2.0.1.300。		
软件：EMUI 8.1.0.108(C00)+ 硬件：Kirin 970+		
3.2 系统模块设计		
3.2.1 个人模块		
<div><div>个人模块</div><div><div>账号管理</div><div>设置</div></div><div><div>登录</div><div>注册</div><div>修改个人信息</div><div>注销</div></div></div>		
图 2.3.5 个人模块		
个人模块在前端方面主要为最右侧的显示登录用户个人信息和发布动态的页面，同时支持个人信息的修改、个人动态的删除等功能，其和动态管理模块具有一定的交集。此外用户点击右上方设置按钮可以进入设置页面，内部包含一些产品信息和注销等功能。该模块在后端部分主要覆盖了登录、注册、更新个人信息等接口，因此综合其前后端表现，其从功能性上来说主要包含账号管理和用户设置这两个子		

模块。

个人模块主要包含对用户账号的管理和设置两个部分：

- 账号管理：实现用户账号注册、登录和修改个人信息等功能，同时在后台支持个人信息的获取，显示于登录后应用的各个位置；
- 设置：显示基本产品信息，实现注销功能。

### 3.2.2 时间线模块

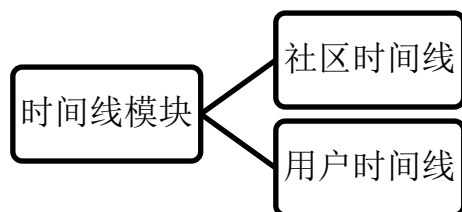


图 2.3.6 时间线模块

时间线模块是该产品的主体模块，包含社区部分的主干时间线子模块和用户部分的枝叶时间线子模块：

- 主干时间线：该部分为社区主体，即按时间倒序展示的用户关注用户和用户自己发布的具体动态，用户可以为每个具体动态进行点赞、评论和转发，这里也是时间线模块和点赞模块、评论模块的交界之处。每条具体动态上方左侧显示发布用户，当前登录用户点击后进入发布用户的个人空间，即进入发布用户的个人时间线；
- 枝叶时间线：进入用户空间后按时间倒序展示的用户自己发布的具体动态，如果该用户即为当前登录用户，即进入个人模块，可以对所有具体动态进行管理。

由于主干时间线上的每条动态的发布用户都具有个人时间线，因此将个人时间线称之为分支/枝叶时间线，其关系如下图 2.3.7。

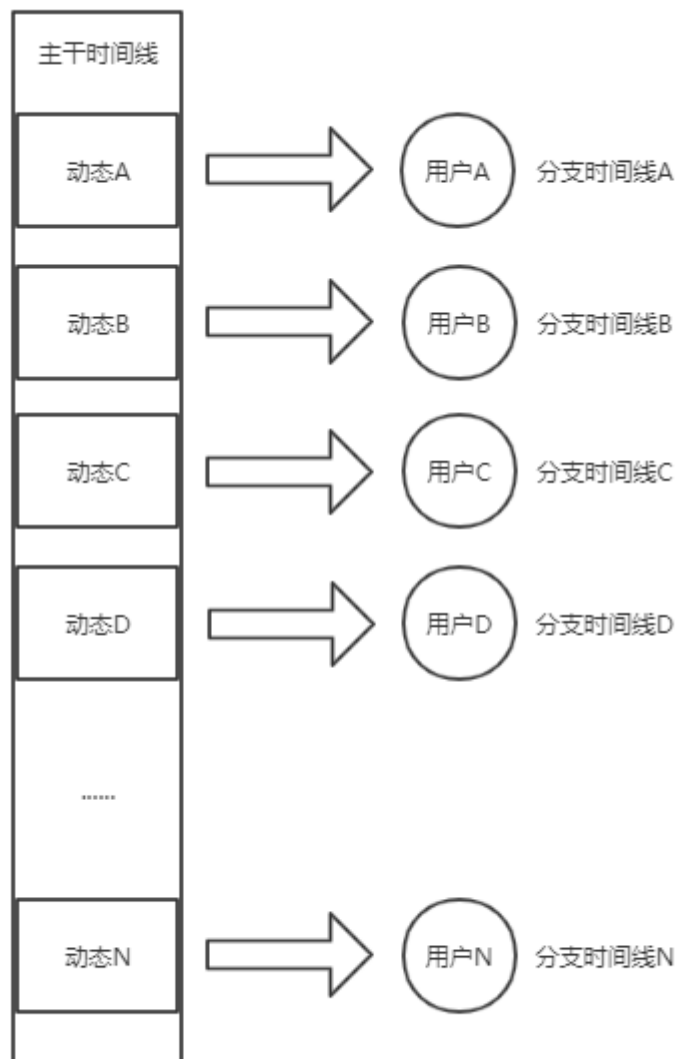


图 2.3.7 主干-分支时间线

### 3.2.3 动态管理模块

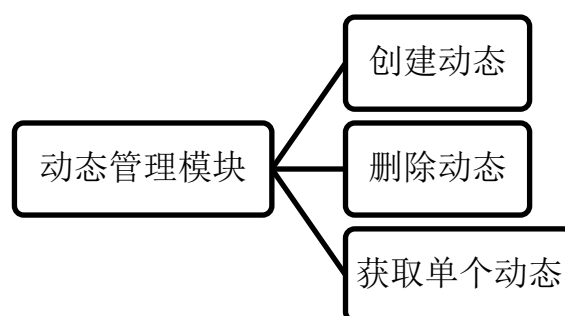


图 2.3.8 动态管理模块

动态管理模块的中心为应用中心 tab 的功能——动态发布功能。

创建动态支持用户即时拍摄或从本地相册中选择心仪的相片。用户可以选择任

意多张相片，并对它们进行排序和裁剪。用户选择上传动态后：

- 相片被上传到对象存储；
- 支持 AI 能力的设备在本地计算出相片的美学评分和标签；
- 不支持 AI 能力的设备的美学评分被替代为随机数，标签被统一设为“其它”；

上传和计算的过程是异步的，其中主要的时间消耗在于上传。用户可以通过 UI 上显示的进度条获悉上传的进度。

当这些预处理过程结束后，一个与本条动态的图片数等长的列表将被作为上传动态成功的消息传送给服务器，其中的元素是相片的 URI、美学评分和标签组成的结构体。

该环节中出现任何错误都将中止整个过程，并将信息反馈给用户。用户随后可以选择重新上传该条动态。

此外，用户个人模块中的动态管理——删除动态也被包含在此模块当中。

在前端的发现和搜索中都会出现照片墙，只展示单张图片，当用户对照片墙或者排行榜中的某张图片感兴趣时可以点击该图片，随后进入由该模块获取单个动态子模块提供支持的动态详情界面，该界面显示单个动态的具体内容，用户也可以在此进行点赞转发和评论操作。

3.2.4 点赞管理模块

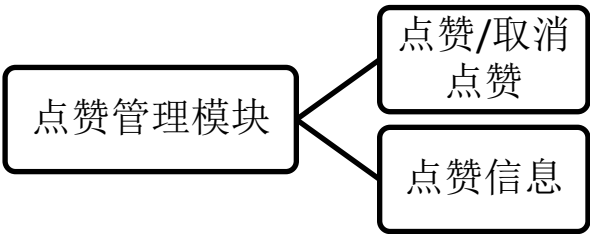


图 2.3.9 点赞管理模块

用户在各类时间线中包含的具体动态、照片墙和排行榜进入的单个动态和排行榜界面中，可以对该动态或排行榜中动态的某张图片（实际对整个动态都进行了点赞）进行点赞或取消点赞操作；在点赞按钮（爱心）的附近一般会显示点赞信息（点赞数），这里由点赞模块的点赞信息子模块提供支持。

3.2.5 评论管理模块

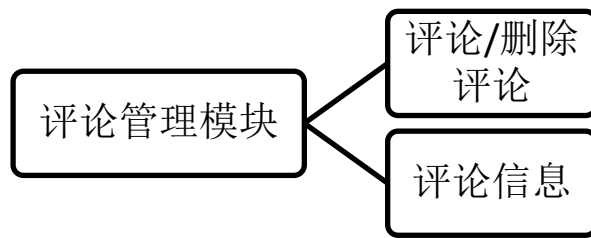


图 2.3.10 评论管理模块

用户在各类时间线中包含的具体动态、照片墙和排行榜进入的单个动态中，点击前端中评论图标或点击“查看全部留言”后，切入评论模块，显示此动态的所有具体评论信息，并且用户可以通过下方的留言框进行评论。评论发布后，评论界面会同步显示用户评论，且用户可以对自已发布过的评论进行管理（删除）。

具体动态下方的评论数目和评论界面中的详细评论内容由评论信息子模块提供支持，而用户输入发布评论和对自己的评论进行管理则由另一子模块作为支撑。

### 3.2.6 关注管理模块

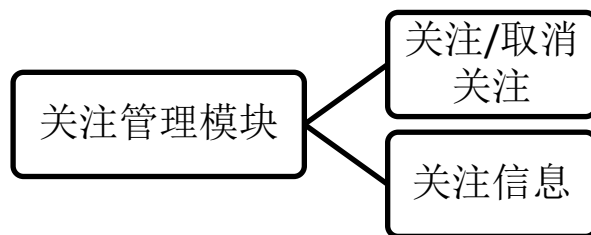


图 2.3.11 关注管理模块

类似于点赞管理和评论管理模块，关注管理模块同样由描述关注内容的关注信息这一数据子模块和关注/取关的动作子模块构成：

- 动作子模块实现关注和取消关注两种功能；
- 信息子模块则为前端显示的关注者和被关注者的数目和具体信息提供一定的数据支撑。

尽管点赞模块、评论模块和关注模块在基础架构上具有极高的相似性，但在具体的前端中的表现形式方面，关注模块与前二者大有不同。首先由个人模块显示用户个人的关注者和被关注者数据，这也是该模块和个人模块的交集。点击个人模块中的关注数据后进入关注信息界面，展示该用户的所有关注者和被关注者的基本信息。此外，个人模块名片部分的下方存在关注中/关注的状态按钮，如果当前登录用户已经关注该用户，则该用户个人模块处会显示“关注中”按钮，否则显示“关注”按钮，点击后进行反向操作：若为“关注中”状态，则点击该状态按钮后实现取消

关注操作，该状态也立即转变为“关注”状态；若为“关注”状态，则点击该状态按钮后实现关注操作，该状态也立即转变为“关注中”状态。

### 3.2.7 推荐模块

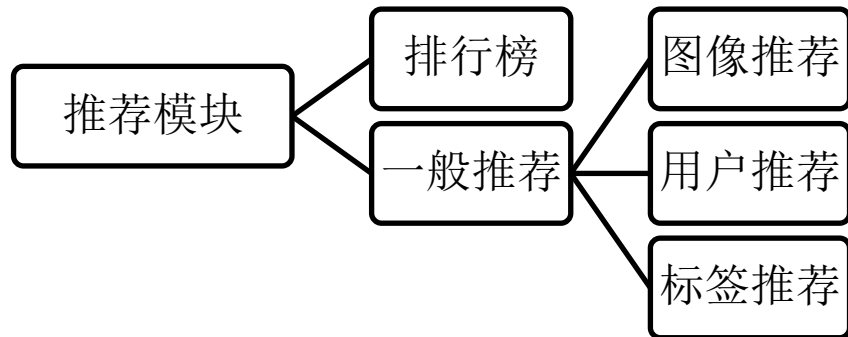


图 2.3.12 推荐模块

推荐模块是该社区项目的一个重点模块，其由排行榜和一般推荐（图像、用户、标签推荐）子模块构成：

- 排行榜：用户点击第四个 **tab** 进入排行榜，排行榜分为日榜和周榜，其根据一定的算法根据用户动态的点赞数、评论数和每张图片的美学评分计算权值，从而为一日内和一周内的所有动态中的图像进行排行，排行主体为图片，点击后进入单个父动态界面。排行榜中每一栏只显示图片、发布人信息和点赞数以及点赞按钮。
- 一般推荐
  - 图像推荐：用户点击第二个 **tab** 进入发现界面，发现界面由上方的搜索栏与主体的照片墙组成，照片墙中随机显示数据库中权重大于某一阈值的一定数目的图片，用户通过刷新可以不断发现新的图片。该推荐主要是为了弥补排行榜中日和周这一时限导致的缺陷，让用户也可以发现一日前、一周以前权重较高的图像；
  - 用户推荐：用户点击搜索栏后进入搜索界面，该界面的第一个 **tab** 会自行列出为当前登录用户推荐的用户，其通过当前用户关注用户的共同关注用户为用户列出推荐用户的基本信息，引导用户拓宽关注圈；
  - 标签推荐：用户上传图片时会自动利用 HiAI 计算该图片的美学评分并通过场景识别为图片打标签。用户点击搜索栏后进入搜索界面，该界面的第二个 **tab** 会首先统计用户动态中最常出现标签，并通过照片墙的形式

式为用户推荐同类标签的图片。

排行榜和各类推荐利用综合计算的权重为用户进行信息推荐，这是该社区“美学导向”这一宗旨的主要体现。

### 3.2.8 搜索模块

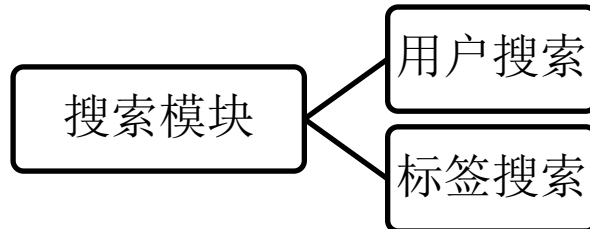


图 2.3.13 搜索模块

搜索模块主要包含用户搜索和标签搜索两个子模块，分别实现以下功能：

- 用户搜索：用户点击搜索栏后进入搜索界面，在搜索栏输入关键词后在账号信息 tab 会实时显示搜索结果，结果为用户名或昵称包含搜索关键词的所有用户的基本信息；
- 标签搜索：用户点击搜索栏后进入搜索界面，在搜索栏输入关键词后在标签 tab 会实时显示搜索结果，搜索结果为包含关键词的所有标签对应的一定数目的图片组成的照片墙。

尽管用户搜索和标签搜索二子模块在功能上是分离的，但在前端调用中是同时调用的，即用户在搜索栏中输入关键词后系统便开始同时搜索用户和标签的搜索结果，并在用户切换 tab 的过程中支持同时显示。此外，搜索支持实时搜索功能，即一旦搜索栏中内容发生改变，系统便进行一次搜索，用户无需点击搜索按钮即可获取搜索结果信息。

## 3.3 流程逻辑设计

### 3.3.1 个人模块

- 用户注册

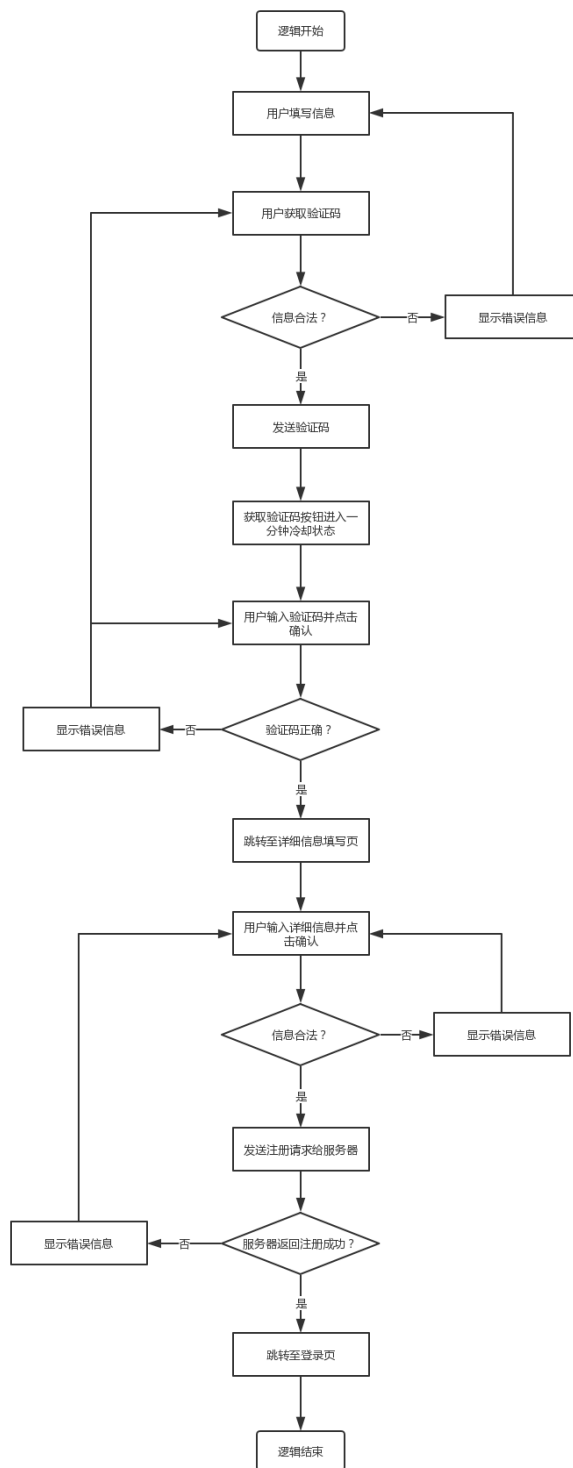


图 2.3.14 用户注册逻辑流程图

其中，验证码按钮的一分钟冷却时间是为了防止用户误以为验证码发送失败而多次点击导致的不必要的开销。

- 用户登录



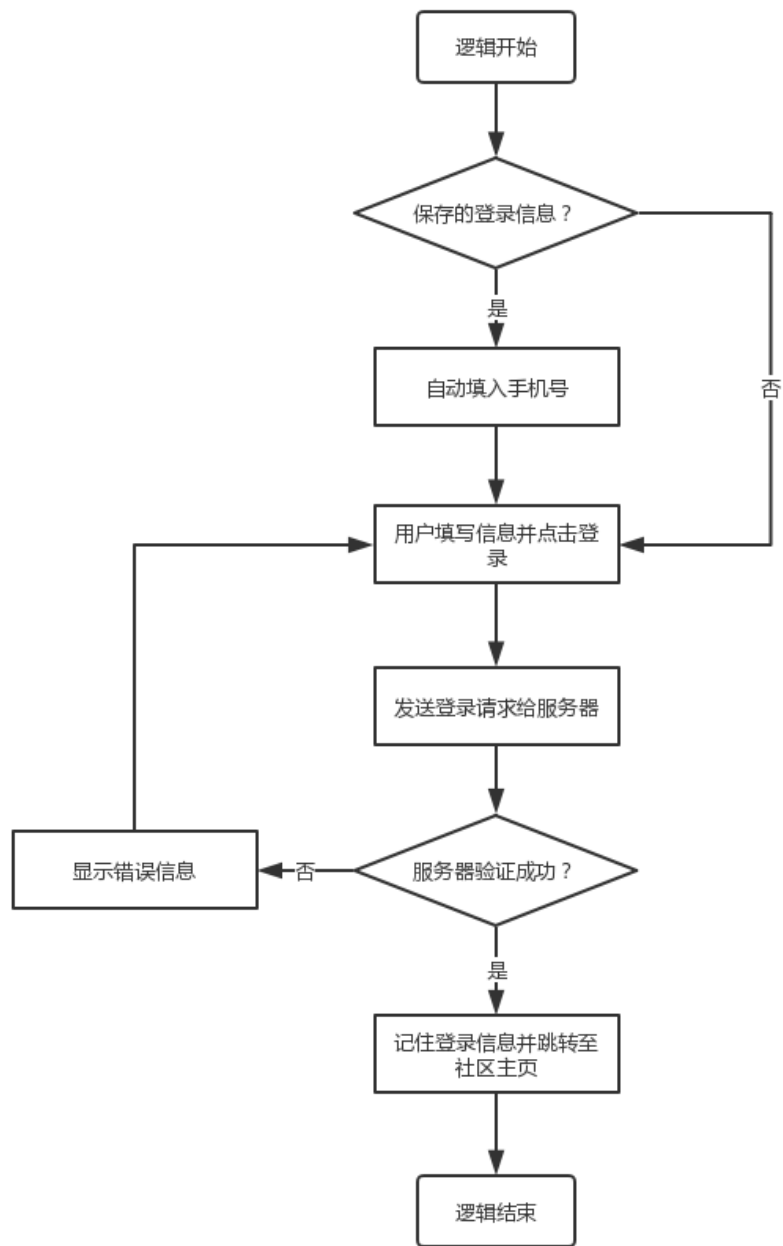


图 2.3.15 用户登录逻辑流程图

此后登录信息将被快应用以全局变量的形式在后台维护。

- 修改个人信息

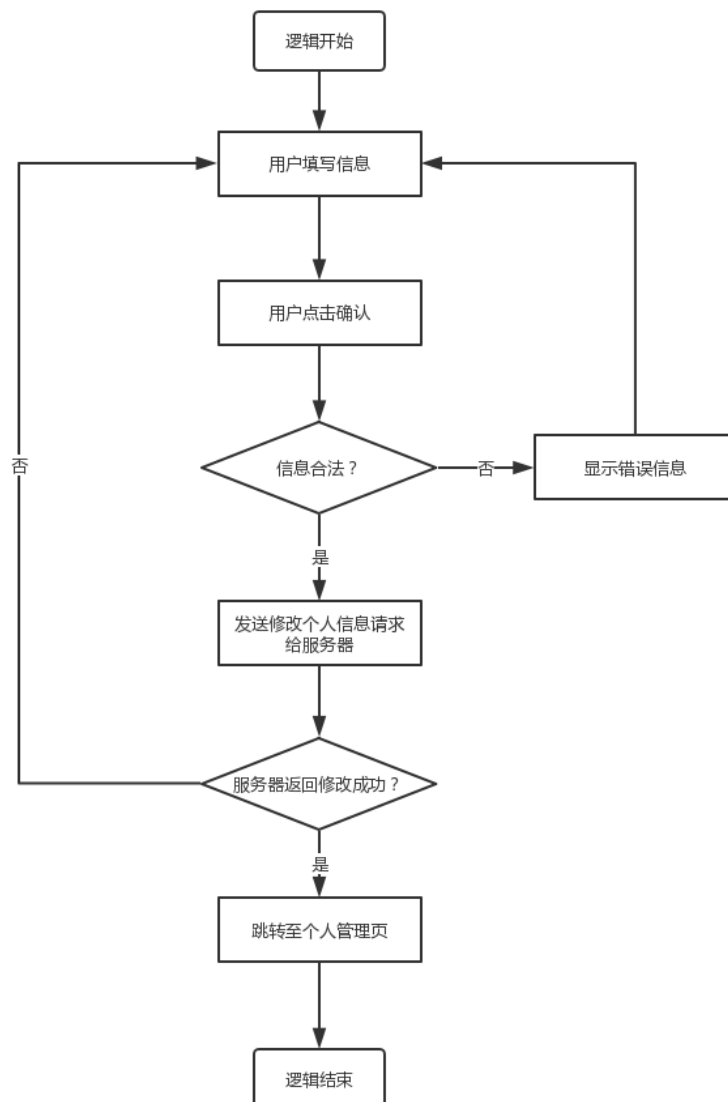


图 2.3.16 修改信息逻辑流程图

无法修改手机号与密码，保证修改信息的合法性。

### 3.3.2 时间线模块

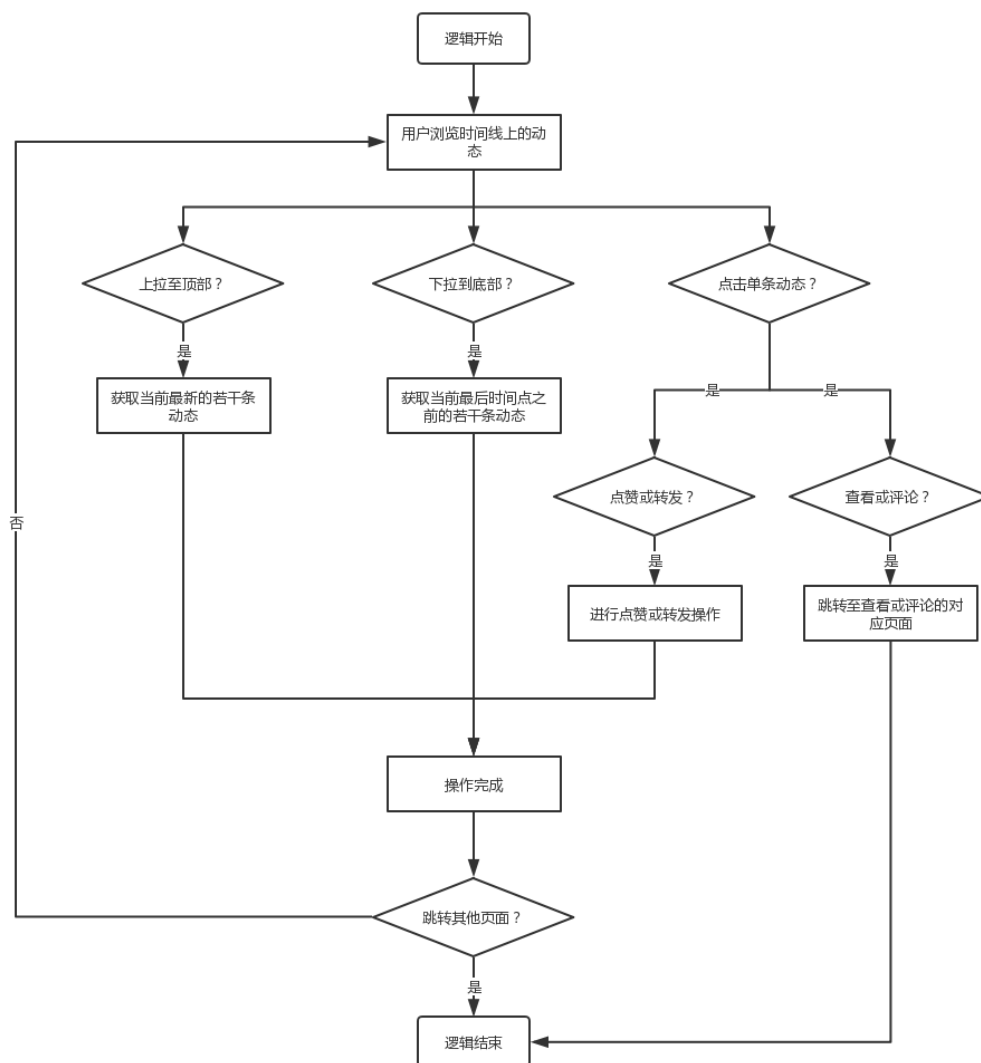


图 2.3.17 时间线逻辑流程图

其中，出现任何失败（HTTP 请求或本地）都将把显示信息给用户。

### 3.3.3 动态管理模块

- 创建动态

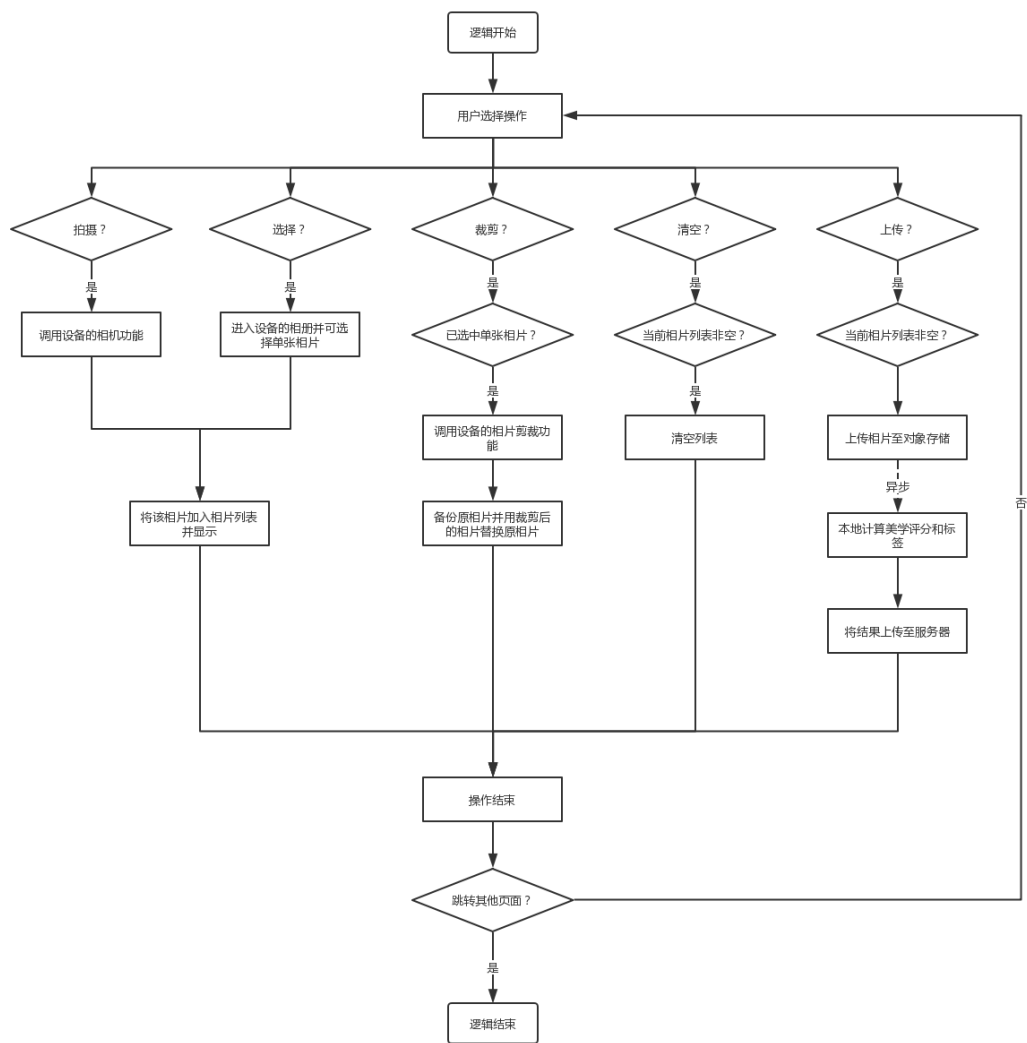


图 2.3.18 创建动态逻辑流程图

其中，出现任何失败（HTTP 请求或本地）都将把显示信息给用户。裁剪操作有备份原图的操作，因此该过程是可逆的。

- 删除动态

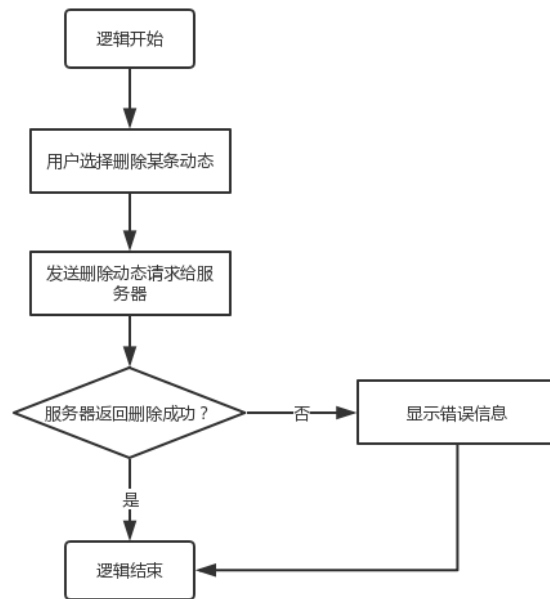


图 2.3.19 删除动态逻辑流程图

删除动态的操作不可逆，并且修改是即时的。

#### 3.3.4 点赞管理模块

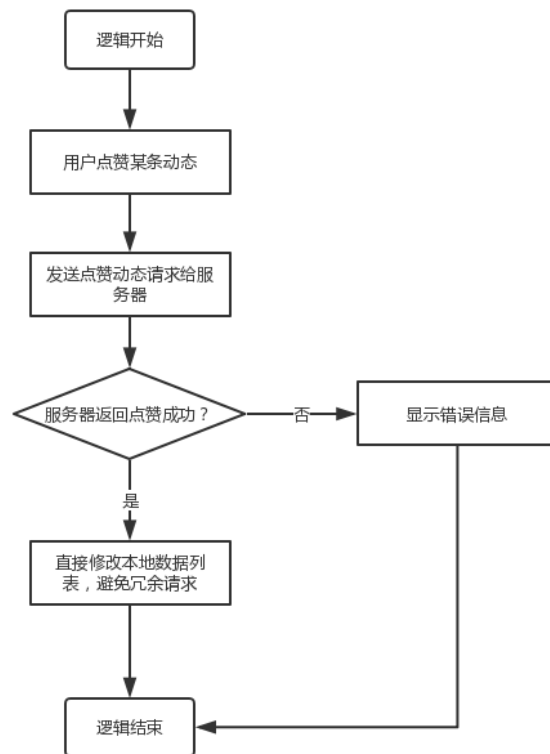


图 2.3.20 点赞逻辑流程图

通过同步修改本地数据列表来防止多次点赞和取消赞的操作使服务器来不及响应。

### 3.3.5 评论管理模块

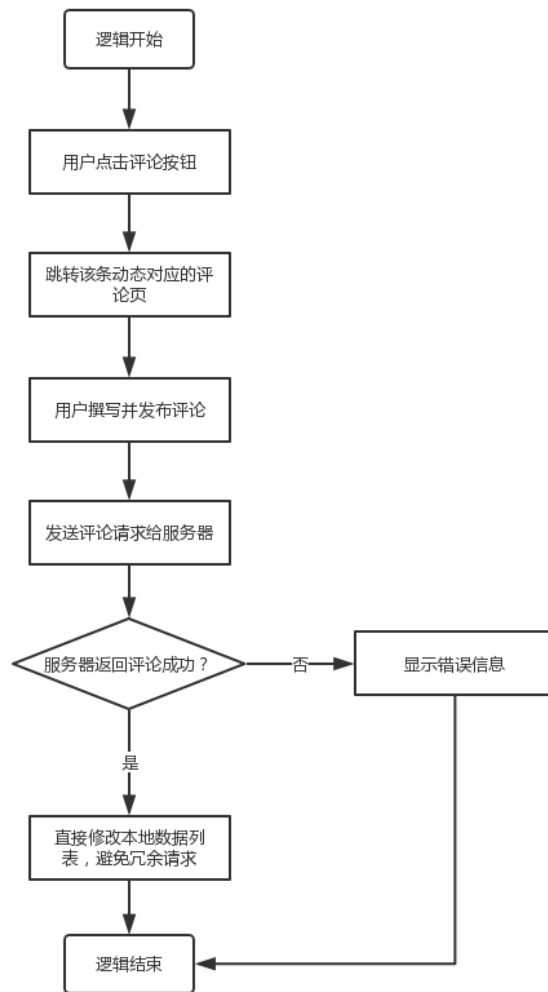


图 2.3.21 评论逻辑流程图

直接修改本地数据列表，防止不必要的 HTTP 请求。

### 3.3.6 关注管理模块

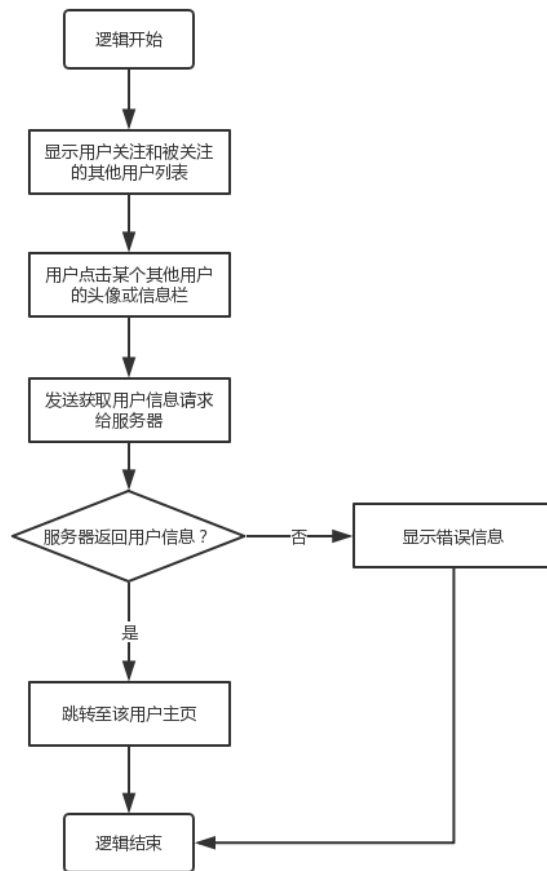


图 2.3.22 关注逻辑流程图

在关注列表中可以进行取消关注操作，与删除动态类似。

### 3.3.7 推荐模块



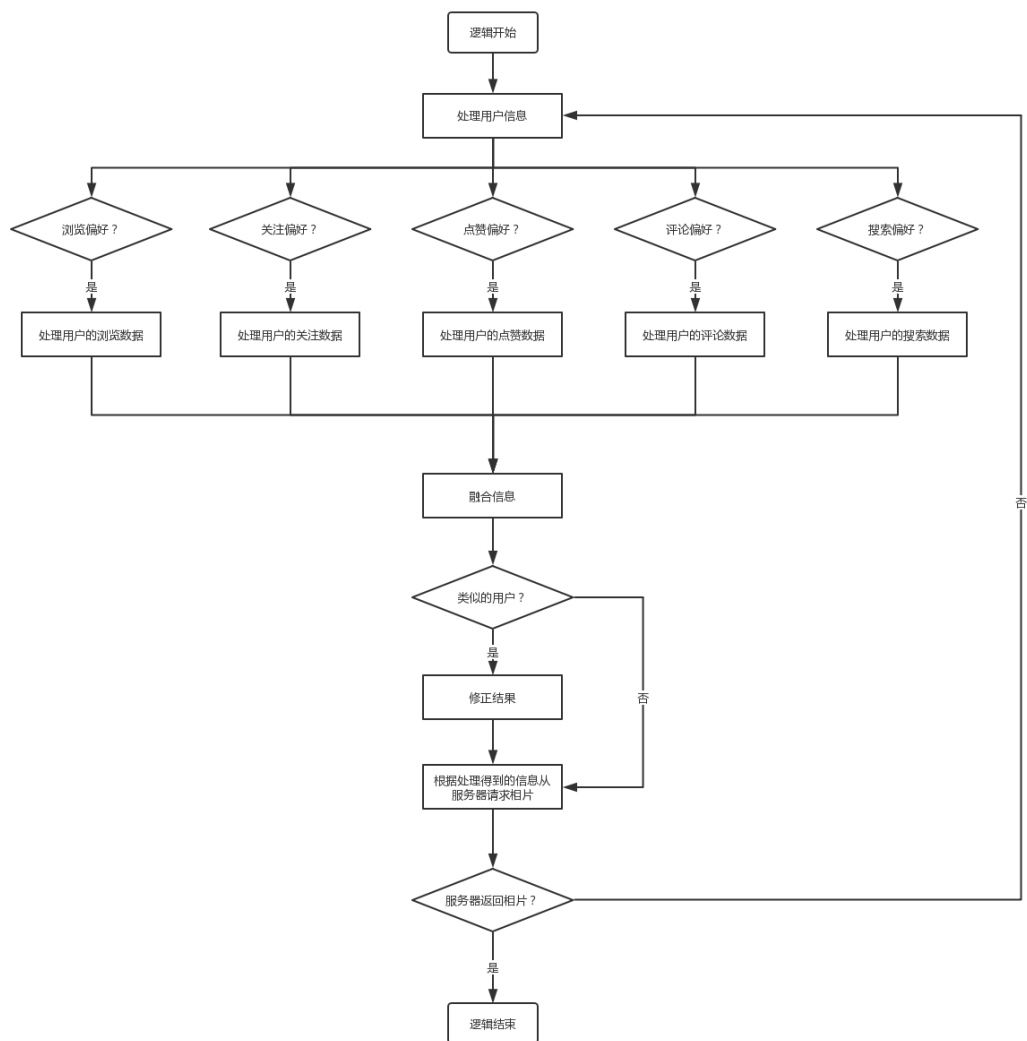


图 2.3.23 推荐逻辑流程图

推荐模块根据用户的各种信息来计算出用户的大致偏好，并以此为基础推荐各种内容。

### 3.3.8 搜索模块

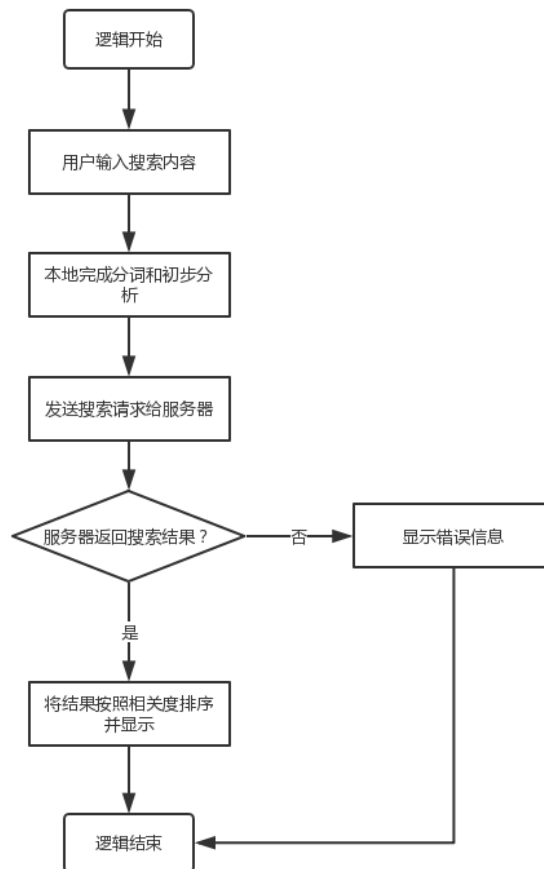


图 2.3.24 搜索逻辑流程图

搜索功能默认提供许多标签供用户选择。当用户选择自行输入搜索内容时，服务器端会完成分析和检索，并返回相关度最高且用户最可能感兴趣的内容。

### 3.4 数据结构设计

#### 3.4.1 数据库系统设计

##### 3.4.1.1 概述

该项目的数据库以华为云关系型数据库（Relational Database Service，简称 RDS）为依托，以这样一种基于云计算平台的即开即用、稳定可靠、弹性伸缩、便捷管理的在线关系型数据库服务为基础，通过创建 Stack 数据库实例，对 innoDB 为存储引擎内核的 MySQL 数据库（5.7.25）进行操作。

云关系数据库具有低成本、高性能、高可靠性和高安全性等特点，且数据库的安全性完全依赖于华为云的安全性。

### 3.4.1.2 基本设计

此数据库主要使用云数据库 MySQL 作为关系型数据库引擎来存储关系型数据，使用 SQL alchemy 和 MyBatis 作为 ORM 框架完成面向对象的对象模型和关系型数据的相互转换。并基于关系型数据库的数据存储，实现一个虚拟的面向对象的数据访问接口，在保证数据的封装性的同时，可以高效快捷地对数据进行访问。

### 3.4.1.3 外部设计

#### 3.4.1.3.1 标识符和状态

本系统使用 MySQL 作为数据库引擎，基于华为云关系型数据库服务，数据库实例中一共建立了 6 张表，如表 2.3.3 所示。

表 2.3.3 数据库标识符和状态表

标识符	表名称	描述信息	使用状态
User	用户信息表	存储用户信息	使用
Post	动态信息表	存储动态内容	使用
Image	图片信息表	存储动态内图片	使用
LikeTable	点赞信息表	存储点赞关系	使用
CommentTable	评论信息表	存储评论内容	使用
FollowTable	关注信息表	存储关注关系	使用

#### 3.4.1.3.2 使用它的程序

使用此数据库的主要程序为 Stack FastAPP，具体情况如表 2.3.4 所示

表 2.3.4 使用程序表

时间	名称	版本号
	Stack FastAPP	1.0

#### 3.4.1.3.3 约定

数据库对象命名约定：

- 名词（实体）类直接以该名词命名数据表，如 User，Post 等；
- 动词（关系）类以动词加”table”命名数据表，如 LikeTable，CommentTable 等。

#### 3.4.1.4 结构设计

##### 3.4.1.3.1 概念结构设计

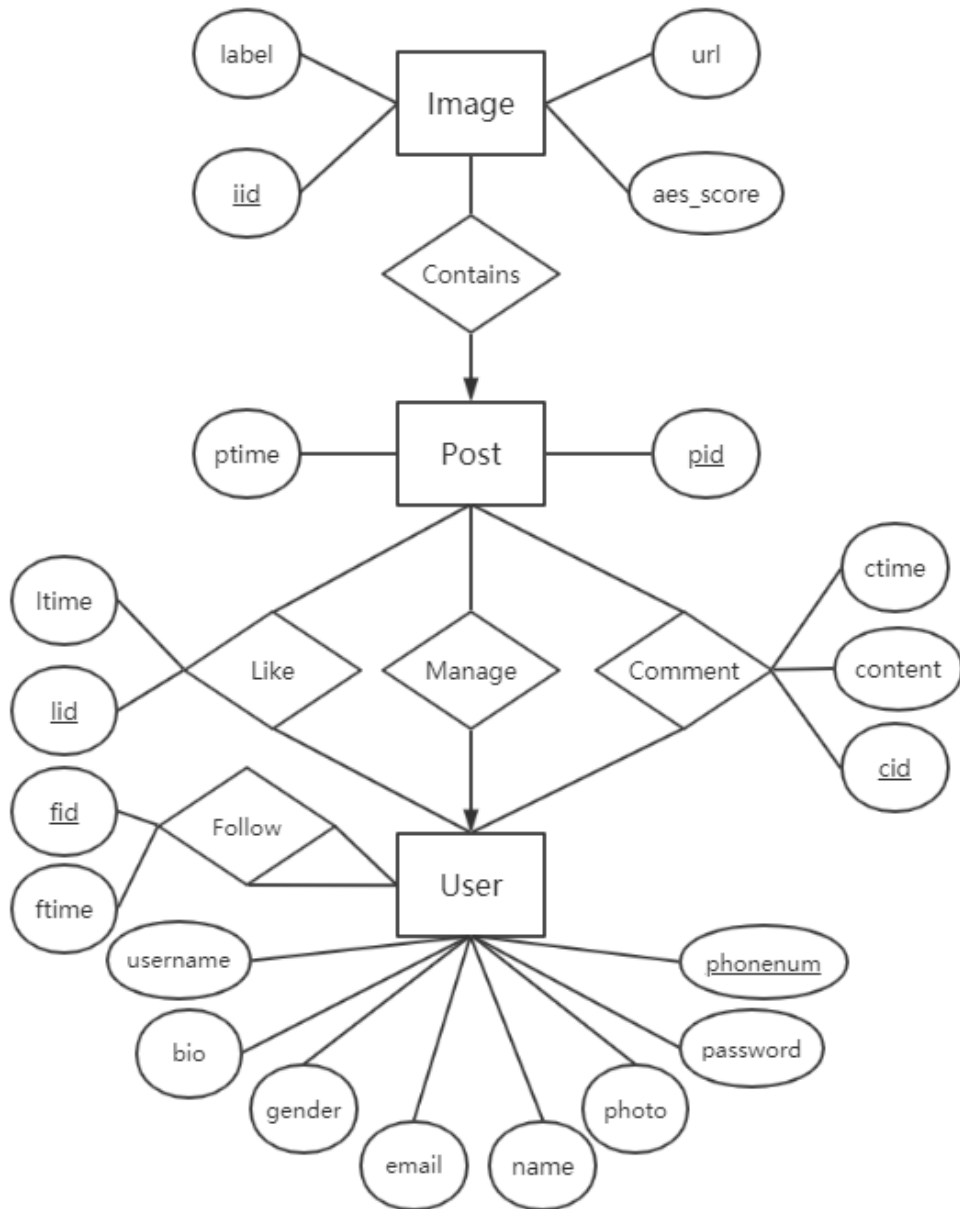


图 2.3.26 ER 图

##### 3.4.1.3.2 逻辑结构设计

将 ER 图转换为关系:

User(phonenum, password, photo, name, username, email, gender, bio)

Post(pid, ptime, phonenum)

Image(iid, url, label, aes\_score, pid)

Like(lid, ltime, pid, phonenum)

Comment(cid, content, ctime, pid, phonenum)

Follow(fid, ftime, follower, followee)

映射到面向对象的对象模型后，如图 2.3.26 所示：

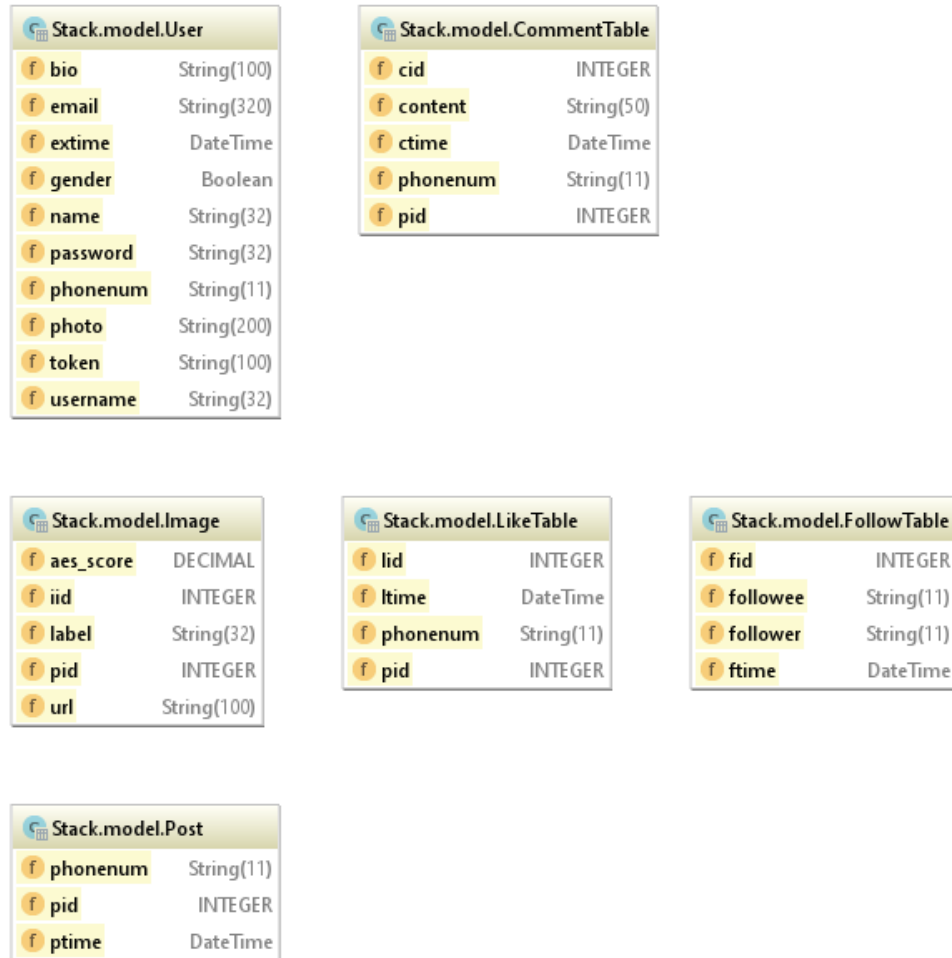


图 2.3.26 SQLAlchemy 对象模型

### 3.4.1.3.3 物理结构设计

在数据库中的存储如下：

表 2.3.5 User Table

字段名	描述	字段类型	可 空	默认值	主 键
phonenum	手机号码	VARCHAR (11)	否	无	是

<b>password</b>	密码	VARCHAR(32)	否	无	否
<b>photo</b>	头像	VARCHAR(200)	否	数据库默认值	否
<b>name</b>	姓名	VARCHAR(32)	是	无	否
<b>username</b>	昵称	VARCHAR(32)	否	随机（前端设置）	否
<b>email</b>	邮箱	VARCHAR(320)	是	无	否
<b>gender</b>	性别	BOOLEAN	是	无	否
<b>bio</b>	个人经历	VARCHAR(100)	是	无	否
<b>token</b>	验证令牌	VARCHAR(100)	是	无	否
<b>extime</b>	过期时间	DATETIME	是	无	否

Photo 默认值为 <https://lucaszhao-1258906334.cos.ap-guangzhou.myqcloud.com/stack/default.jpg>

Token 和 extime 字段用于 token 验证，故不在 ER 图及其关系中显示。

表 2.3.6 Post Table

字段名	描述	字段类型	可空	默认值	主键
<b>pid</b>	序号	INT(5)	否	无	是
<b>ptime</b>	发送时间	DATETIME	否	无	否
<b>phonenum</b>	发送用户手机号	VARCHAR (11)	否	无	否

表 2.3.7 Image Table

字段名	描述	字段类型	可空	默认值	主键
<b>iid</b>	序号	INT(5)	否	无	是
<b>url</b>	资源定位符	VARCHAR(100)	否	无	否
<b>label</b>	标签/分类	VARCHAR(32)	是	无	否
<b>aes_score</b>	美学分数	DECIMAL	否	无	否
<b>weight</b>	美学和点赞权值	DECIMAL(SIGNED)	否	无	否
<b>pid</b>	所属动态	INT(5)	否	无	否

Weight 域的设置是为了便于排行，本质是点赞数和美学分数，因此不在 ER 图中体现。

表 2.3.8 Like Table

字段名	描述	字段类型	可空	默认值	主键
lid	序号	INT (5)	否	无	是
ltime	点赞时间	DATETIME	否	无	否
pid	动态序号	INT (5)	否	无	否
phonenum	点赞用户手机号	VARCHAR (11)	否	无	否

表 2.3.9 Comment Table

字段名	描述	字段类型	可空	默认值	主键
cid	序号	INT (5)	否	无	是
content	评论内容	VARCHAR (50)	否	无	否
ctime	评论时间	DATETIME	否	无	否
pid	动态序号	INT (5)	否	无	否
phonenum	评论用户手机号	VARCHAR (11)	否	无	否

表 2.3.10 Follow Table

字段名	描述	字段类型	可空	默认值	主键
fid	序号	INT (5)	否	无	是
ftime	关注时间	DATETIME	否	无	否
follower	关注者用户手机号	VARCHAR (11)	否	无	否
followee	被关注者用户手机号	VARCHAR (11)	否	无	否

### 3.4.1.5 安全性设计

- 支持网络隔离：通过虚拟私有云（Virtual Private Cloud，简称 VPC）和网络安全组实现网络隔离。通过配置虚拟私有云入站 IP 范围，来控制连接数据库的 IP 地址段。
- 支持访问控制：通过主/子帐号和安全组实现访问控制。在创建数据库实例时，会同步创建一个数据库主帐户，并可根据需要创建数据库实例和数据库子帐户，将数据库对象赋予数据库子帐户，从而达到权限分离的目的。此外，还可以通过虚拟私有云对数据库实例所在的安全组入站、出站规则进行限制，

从而控制可以连接数据库的网络范围。

- 可以防范 DDoS 攻击：数据库处于多层防火墙的保护之下，可以有力地抗击各种恶意攻击，保证数据安全。当华为云关系型数据库安全体系认为用户实例正在遭受 DDoS 攻击时，会首先启动流量清洗的功能，如果流量清洗无法抵御攻击或者攻击达到黑洞阈值时，将会进行黑洞处理，保证华为云关系型数据库整体服务的可用性。

### 3.4.2 文件系统设计

#### 3.4.2.1 概述

该项目的文件系统主要负责存储用户发布动态中的图像，支持图像上传下载等基础操作，并可将上传后的图像映射为唯一的支持独立访问的 URL，便于连同动态其他信息一并存储于数据库中。

考虑到以上需求以及图像文件的特殊性，这里使用了腾讯云对象存储 COS 这样一种支持存储海量数据的分布式存储服务，且可以在任意位置存储和检索任意大小的数据。依赖于 COS 支持的断点续传、自定义对象头部、静态网站、数据迁移等多种功能，可以高效便捷地完成图像存储系统的设计与搭建。

此外，由于 COS 基于腾讯云服务，其可靠性和安全性完全依赖于腾讯云的可靠性和安全性。

#### 3.4.2.2 基本设计

文件系统采用为每个用户设置一个单独的文件夹的方式来存储图像。为了与数据库中以用户手机号码为键（Key）的设计一致，我们将每个用户独有的文件夹命名为用户的手机号。这样可以带来以下好处：

- 避免用户上传多张重复图像而占用过大的存储空间
- 可以根据用户上传的数据的格式、大小等特点来分别优化具体的存储和加密方式
- 当用户进行删除某条动态的操作时，不影响其所发布的其他动态
- 当本地的数据丢失时，该文件夹可以方便的作为备份提供给用户

当用户进行转发动态时，我们保证图像在转发者和被转发者的文件夹中均有备份。这是一种有效的冗余手段，可以为多个用户提供完整的、有效的数据存储和备份。



同时，我们在存储图像时不改变图像原有的名称和扩展名，但对于转发的动态中的图像进行统一的重命名。这是为了尽可能的在单个用户的存储空间内防止冗余，而在多个用户的不同存储空间内促进冗余，以此来节省存储空间，并提高容错率和可恢复性。由于图像存储系统是分布式的，即使某个用户的存储空间完全失效，我们也可以通过其他用户的存储空间和所维护的转发关系恢复出该用户的大部分数据，而对于用户本地数据的重新上传只有非常少的需求。

#### 3.4.2.3 可靠性和安全性设计

由于 COS 基于腾讯云服务，其可靠性和安全性完全依赖于腾讯云的可靠性和安全性，其在可靠性和安全性等方面具有以下设计：

- 支持异地容灾：可以通过配置跨地域复制规则，在不同存储地域的存储桶中自动、异步地复制增量对象，实现数据的异地容灾与备份；
- 支持加密：COS 在数据写入数据中心内的磁盘之前，支持在对象级别上应用数据加密的保护策略，并在访问数据时自动解密；
- 可以通过防盗链功能保障图像的安全：支持防盗链配置，用户可以通过控制台的防盗链功能配置黑/白名单，对数据资源进行安全防护，从而保证只有前端应用可以访问图像。

### 3.5 系统能力毕业要求达成度分析

根据南京理工大学计算机科学与工程学院软件工程专业本科生毕业方案指针要求，每一位本科毕业生都必须拥有进行系统开发的能力。

本次毕业实习进行软件开发是根据华为技术有限公司南京研究所提供的技术要求，在南研所工程师指导下进行软件开发，所以所使用的技术，语言等均和行业发展水平保持一致。

本次开发的快应用是华为同其他九大厂商一起基于硬件平台推出的新型免安装应用，具有安卓生态的应用体验好，功能强的特点，同时网页自动更新，无需占用内存，同时使用华为提供的快应用 ide，开发难度大大降低。

本次毕业实习历经一个月，无论是实习前期进行的项目开发计划制定，以及需求分析，软件设计，还是项目开发中期的程序实现，直到最后的测试工作，所有参与本图片社区项目开发的小组成员都全程参与了此次项目开发，最终项目答辩顺利结束，因此成员均掌握了程序开发的基本知识，具备了进行软件系统开发的基本能

力，达到本科毕业要求。

## 4 系统实现

### 4.1 系统关键模块实现

#### 4.1.1 创建动态模块实现

##### 4.1.1.1 概述

创建动态模块允许用户在本地使用设备的相机拍摄相片，或从设备的相册中选择相片，并对相片进行排序、裁剪后上传相片并创建动态。

该模块具有以下特点：

- 支持任意多张相片和多种格式（如 jpeg、png、gif 等）
- 支持对相片的裁剪和还原
- 过程中即使用户不小心切换页面也不会丢失进度

当用户点击上传按钮后，后台会完成如下操作：

- 将动态中包含的所有相片上传至对象存储
- 支持 AI 能力的设备将在本地计算出每张相片的美学评分和标签
- 不支持 AI 能力的设备将用随机数和“其它”来代替美学评分和标签

上述操作均是异步完成的，主要的时间消耗在于上传图片，而主要的计算量在本地完成。用户可以通过 UI 上显示的进度条获知上传进度。

当这些操作完成后，快应用后台将所有信息组织成一个列表发送给服务器，作为操作成功的消息。列表中每个元素都包含相片的 URI、美学评分和标签。

这样的设计具有以下好处：

- 尽可能的减少了直接将大文件存储在服务器带来的性能开销
- 平衡了上传时间和计算时间，使得任务的总体延时在可接受范围内
- 整体的容错性较强，保证了对服务器的事务的原子性

##### 4.1.1.2 代码实现

上传图片的部分使用了 HMAC SHA1 加密和 Base64 编码，主要包含获取时间戳，加密算法的实现和 HTTP 报头、签名等的格式。

美学评分和标签的部分调用了华为 HIAI 接口，并对返回的评分和标签进行了处理，保证其合法性。

所有子任务完成后都将产生回调，当累加的计数器的值等于子任务个数时，就

代表所有子任务都完成了。这时将调用函数与服务器通信，完成最后的列表上传。

由于以上部分包含较多的回调函数和计时器函数，较为复杂，因此不展示代码。

#### 4.1.1.3 界面

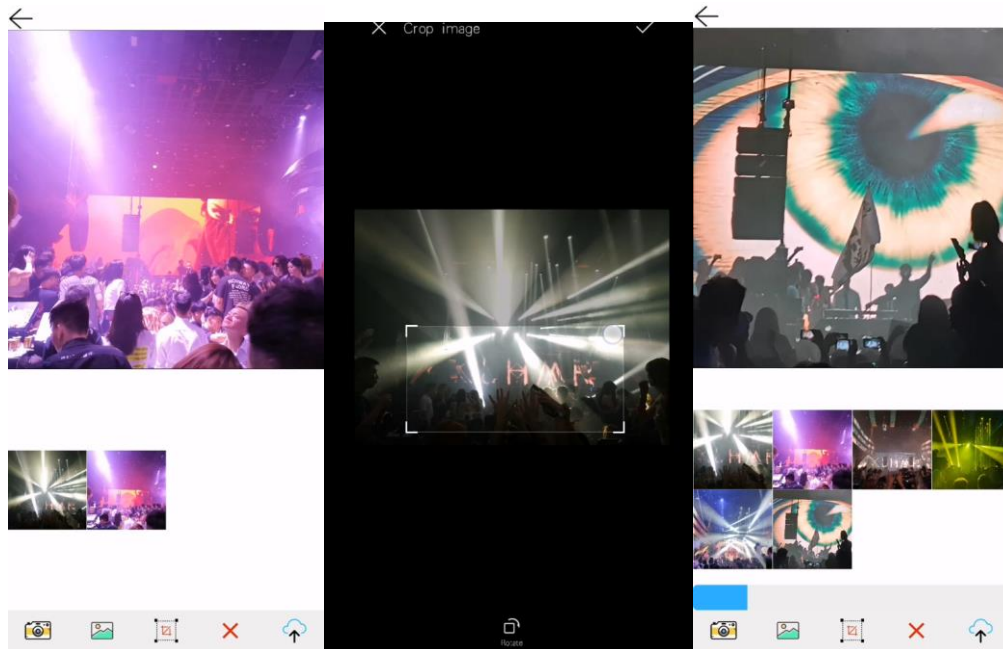


图 2.4.1 创建动态界面

#### 4.1.2 推荐模块实现

##### 4.1.2.1 概述

推荐模块包含的排行榜和图像、用户、标签推荐子模块以综合美学评分、点赞数、评论数计算的权重为导向为用户进行信息推荐，这是该社区“美学导向”这一宗旨的主要体现。

在设计图像权重的过程中，为了保证美学评分这一百分制分数和点赞数、评论数这两个和社区总体用户数目挂钩的数值的平衡，同时让系统在点赞数和评论数过少时去推动美学评分的影响力，利用以下公式对图像权重进行计算：

$$\text{weight} = (\text{Aes\_score} - 50) + 40 * \frac{\text{likes}}{\text{total} * \alpha} + 10 * \frac{\text{comments}}{\text{total} * \beta}$$
$$\text{const } \alpha = 10\%, \beta = 10\%$$

假定当前社区最大用户为 100 人。

这样的权重设计具有以下目的：

- 美学评分最大可占比总权重 50%，保证美学评分的主要导向性；
- 点赞数和评论数占比比例为 4 比 1，因为用户最多可以点一个赞，而可以多

次评论：

- 使用点赞数、评论数和社区总数乘以权的比例来替代点赞数和评论数，有效避免点赞数评论数在无限增长时产生的负面影响；

- 动态中所有图片的点赞数和评论数依赖于动态，都相同，但权重有所差异。

#### 4.1.2.2 排行榜子模块

##### 4.1.2.2.1 核心算法

按图片权重排序获取一周内或一日内所以动态中图片权重排行前十的图片。权重设计见概述。获取函数如下：

```
1. def get_by_weight(self, start, end):
2.     from Stack.config import RANK_LIMIT
3.     return self.query\
4.         .filter(start < Post.ptime, Post.ptime <= end)\
5.         .filter(self.pid == Post.pid)\
6.         .order_by(self.weight.desc())\
7.         .limit(RANK_LIMIT)\
8.         .all()
```

##### 4.1.2.2.2 前端设计

排行榜分为日榜和周榜两个 Tab，每个 Tab 内容相同，内部有总数不超过十的子图像模块，模块包含排行图片、下方的发布人基本信息和点赞数以及点赞按钮。如图。



图 2.4.2 排行榜界面

#### 4.1.2.2.3 后端实现

由于大部分的后端代码相似，这里只展示排行榜部分后端代码。

##### a) 获取表单参数

```
1. @app.route('/rank/get_rank', methods=['POST'])
2. def get_rank():
3.     data = {}
4.     phonenum = request.form.get('phonenum')
5.     period = int(request.form.get('period'))
6.     cur = request.form.get('time')
```

##### b) 后端验证

```
1. try:
2.     schema(
3.         {
4.             "phonenum": phonenum,
5.             "ptime": cur,
6.             "period": period
```

```

7.         }
8.     )
9.     conforms_to_schema = True
10. except MultipleInvalid as e:
11.     data['status'] = 400
12.     conforms_to_schema = False
13.     if "expected" in e.msg:
14.         data['message'] = e.path[0] + " is not in the correct format"
15.     else:
16.         data['message'] = e.msg + " for " + e.path[0]

```

c) 验证成功则获取数据库中信息

```

1. if conforms_to_schema:
2.     try:
3.         images = Image.get_by_weight(Image, start=pre, end=cur)
4.         res = list()
5.         for image in images:
6.             imgdic = Image.out(Image, image)
7.             post = Post.get(Post, image.pid)
8.             imgdic['likes'] = LikeTable.get_count_by_pid(LikeTable, pid=post.pid
9.             )
10.            imgdic['isliked'] = True if LikeTable.get_by_pp(LikeTable, pid=post.
11.            pid,
12.            phonenum=phonenum) is
13.            not None else False
14.            usr = User.partly_out(User, User.get(User, post.phonenum))
15.            imgdic['user'] = usr
16.            res.append(imgdic)
17.            data['message'] = res
18.            data['status'] = 200
19.        except Exception as e:
20.            data['status'] = 406
21.            data['message'] = str(e)

```

d) 返回 JSON 格式结果

```

1. return jsonify(data)

```

#### 4.1.2.3 图像推荐子模块

##### 4.1.2.3.1 核心算法

随机获取权重大于一定阈值的一定数目的所有动态的图片。权重设计见概述。

获取函数如下：

```
1. def get_recommended(self):
2.     from Stack.config import RECOMMEND_LIMIT
3.     return self.query\
4.         .filter(self.weight > 10) \
5.         .order_by(func.rand()) \
6.         .limit(RECOMMEND_LIMIT) \
7.         .all()
```

#### 4.1.2.3.2 前端设计

图像推荐通过如下图的照片墙来展示，由于快应用暂时无法显示瀑布流式设计，采用固定大小的方块瓷砖式排列，通过 `cover` 来显示不同大小的图片。

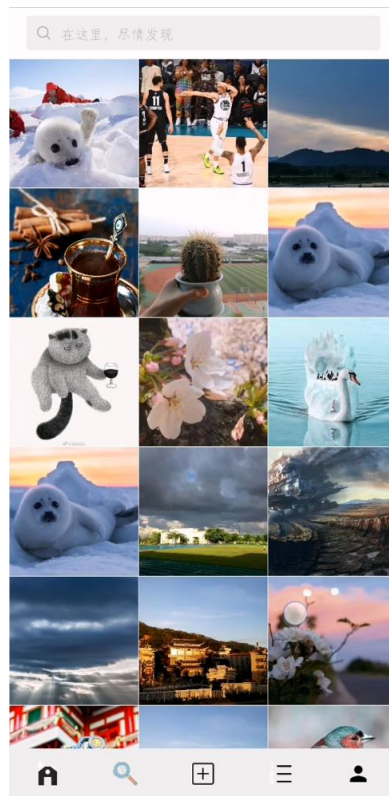


图 2.4.3 照片墙

#### 4.1.2.4 用户推荐子模块

#### 4.1.2.4.1 核心算法

通过统计当前登录用户关注的所有用户的共同关注用户，建立共同关注用户到共同关注数的映射表，依序为用户推荐。具体设计如下：

```
1. followees = FollowTable.get_followees(FollowTable, follower=phonenum)
```

```
2. res = dict()
3. for followee in followees:
4.     # print(FollowTable.out(FollowTable, followee))
5.     user = User.get(User, phonenum=followee.followee)
6.     fs = FollowTable.get_followees(FollowTable, follower=user.phonenum)
7.     for f in fs:
8.         u = User.get(User, phonenum=f.followee)
9.         if u.phonenum not in res:
10.             res[u.phonenum] = 1
11.             res[u.phonenum] += 1
12.
13. res = sorted(res.items(), key=lambda x: x[1], reverse=True)
14. from Stack.config import U_RECOMMEND_LIMIT
15. count = 0
16. ans = list()
17. for pn, cnt in res:
18.     if pn != phonenum and FollowTable.get_by_ff(FollowTable, phonenum, pn) is None:
19.         userdict = User.out(User, User.get(User, phonenum=pn))
20.         userdict['count'] = cnt
21.         ans.append(userdict)
22.         count += 1
23.     if count >= U_RECOMMEND_LIMIT:
24.         break
```

#### 4.1.2.4.2 前端设计

在后台加载推荐，点击搜索栏后自动显示。



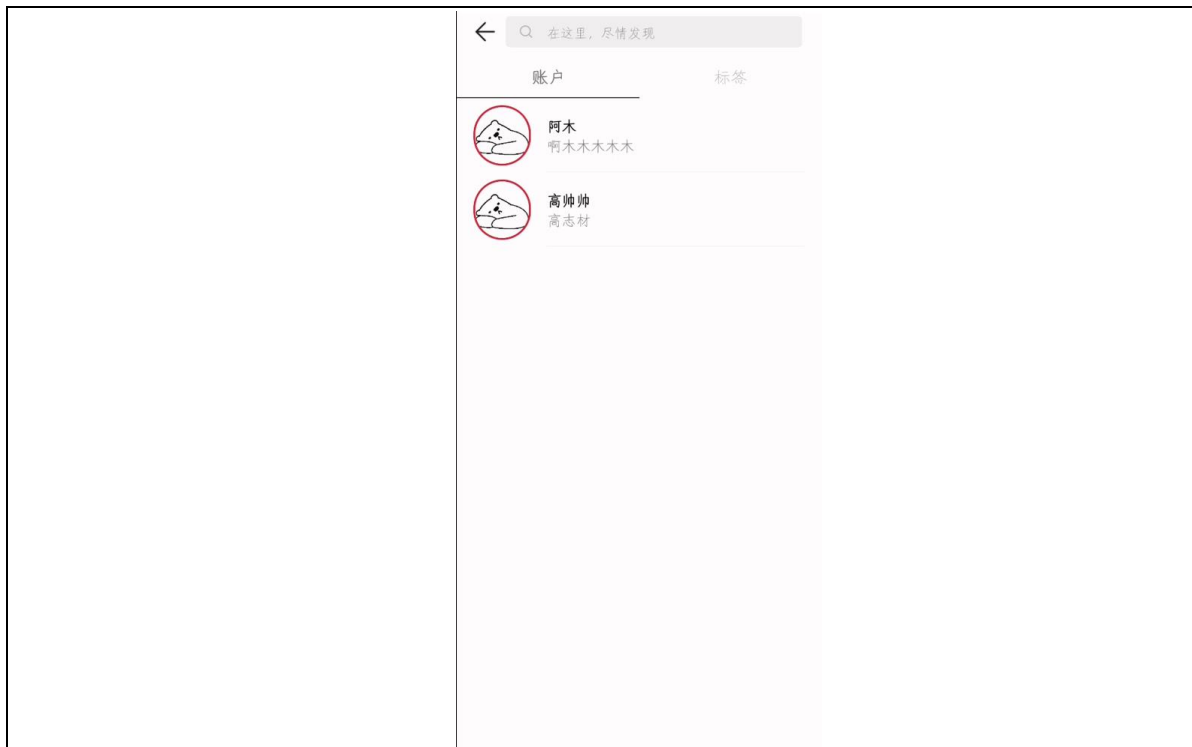


图 2.4.4 用户推荐

#### 4.1.2.5 标签推荐子模块

##### 4.1.2.5.1 核心算法

统计用户已发布动态的最常见标签，为用户推荐同质图像。具体实现如下：

```
1. posts = Post.get_all_posts(Post, phonenum)
2. res = dict()
3. for post in posts:
4.     images = Image.get_by_pid(Image, post.pid)
5.     for image in images:
6.         labels = image.label.split()
7.         for label in labels:
8.             if label not in res:
9.                 res[label] = 1
10.            res[label] += 1
11.
12. res = sorted(res.items(), key=lambda x: x[1], reverse=True)
13. from Stack.config import L_RECOMMEND_LIMIT
14. count = 0
15. ans = list()
16. for label, cnt in res:
17.     if label != '其它':
18.         images = Image.search(Image, keyword=label)
19.         for image in images:
```

```

20.         if Post.get_by_pp(Post, image.pid, phonenum) is None:
21.             ans.append(Image.out(Image, image))
22.             count += 1
23.         if count >= L_RECOMMEND_LIMIT:
24.             break

```

#### 4.1.2.5.2 前端设计

在后台加载推荐，点击搜索栏后自动显示。

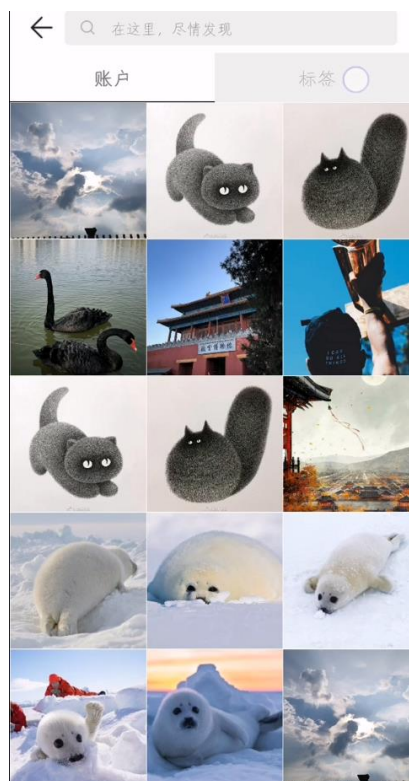


图 2.4.5 标签推荐

### 4.2 系统接口实现

#### 4.2.1 接口概述

接口设计和模块设计以及各级子模块、子系统相适应，基本符合需求分析的要求。接口主要分为个人管理、时间线、动态管理、点赞管理、评论管理、关注管理、推荐和搜索接口 八个部分，由分别部署于两台云服务器上的基于 Sprint Boot 和 Flask 框架的后端提供，并由源于华为云关系数据库 MySQL 实例作为数据支撑。

每个模块的具体接口如图 2.4.6，接口详情可见《Stack 接口文档》。

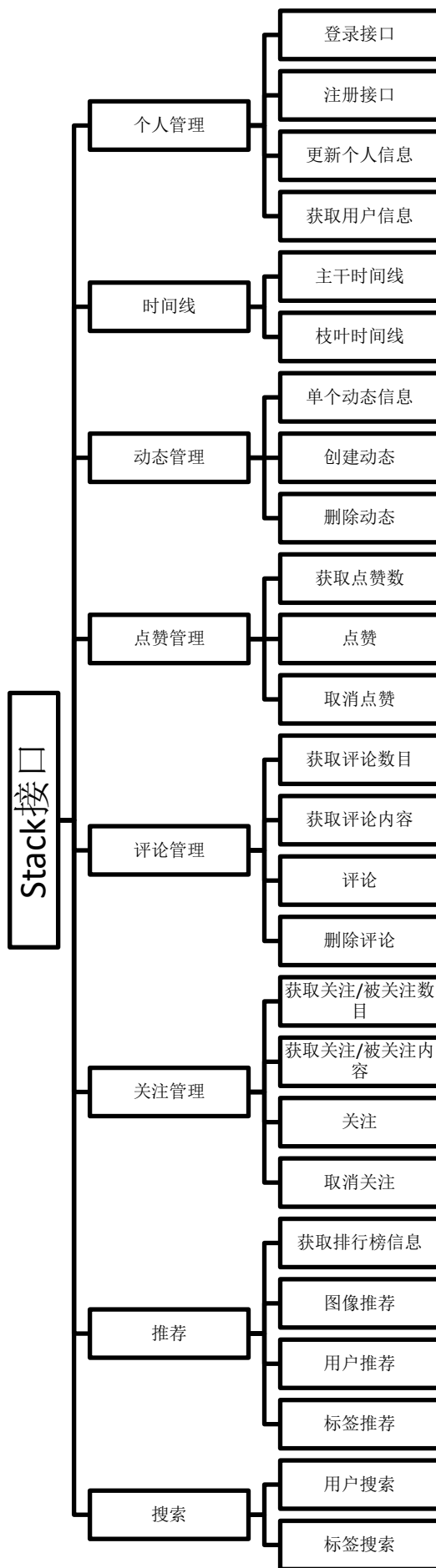


图 2.4.6 接口模块

4.2.2 接口定义

- 交互机制设计
  - 发起: 用户在快应用中进行操作后由前端的后台通过 Javascript 函数调用 API 进行发起;
  - 返回: 同步返回;
  - 错误应对: 后端返回错误码, 前端判断错误码向用户弹出不同 Toast(消息提示框);
- 接口技术选择
  - 格式: 请求头描述报文 Body 格式 (application/json), 报文以 JSON 格式传送内容;
  - 请求类型: 以 POST 表单提交为主体;
  - 安全性: 提交时 Body 中存放 token, 由后端查询对应账号的当前 token, 依此判断是否过期、是否正确。由于系统内部对未登录用户进行了隔离, 因此社区内部 API 暂未做验证处理。
- 接口定义格式
  - 通信协议: HTTP
  - 状态码:

表 2. 4. 1 状态码

状态码	含义
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
405	Bad Request
406	Not Acceptable
500	Internal Server Error
501	Not Implemented
502	Bad Gateway

### 4.2.3 示例接口实现

这里以获取用户评论数接口为例，说明接口实现方法与过程。

#### 1) 通过 POST 获取表单提交内容

```
1. @app.route('/comment/get_comments', methods=['POST'])
2. def get_comments():
3.     data = {}
4.     pid = int(request.form.get('pid'))
```

#### 2) 后端验证格式

```
1. try:
2.     schema(
3.         {
4.             "pid": pid
5.         }
6.     )
7.     conforms_to_schema = True
8. except MultipleInvalid as e:
9.     data['status'] = 400
10.    conforms_to_schema = False
11.    if "expected" in e.msg:
12.        data['message'] = e.path[0] + " is not in the correct format"
13.    else:
14.        data['message'] = e.msg + " for " + e.path[0]
```

#### 3) 如果格式合法，判断动态是否存在

```
1. if conforms_to_schema:
2.     # check post secondly
3.     if Post.get(Post, pid) is None:
4.         data['status'] = 404
5.         data['message'] = "Post {} doesn't exist".format(pid)
```

#### 4) 获取评论数

```
1. else:
2.     try:
3.         lens = CommentTable.get_count_by_pid(CommentTable, pid=pid)
4.         data['message'] = lens
5.         data['status'] = 200
6.     except Exception as e:
7.         data['status'] = 406
```

```
8. data['message'] = str(e)
```

## 5) 返回 JSON 格式结果

```
1. return jsonify(data)
```

### 4.2.4 示例接口文档

其在《Stack 接口文档》中具体内容如下：

#### 1) 接口简介

获取评论数目。

#### 2) 接口详情

- 请求地址：http://114.116.248.233:12345/comment/get\_comments
- 请求类型：Post
- 请求 Header 参数：

表 2.4.2 Header

Key	Value	Description
Content-Type	application/json	

- 请求 Body 参数：

表 2.4.3 Body

Key	Type	Nullabl e	Example
pid	String	no	10005

Body 类型为 form-data

- 返回正确 Json 示例

```
{  
  "message": 0,  
  "status": 200  
}
```

- 返回错误 Json 示例

```
{  
  "message": "Post 999999 doesn't exist",  
  "status": 404  
}
```

}

- 备注

- 3) 修改日志

被整合至时间线接口内部，此接口不调用。

#### 4.3 复杂工程问题能力毕业要求达成度分析

本次的毕业实习是根据华为提供的快应用开发技术进行快应用开发，快应用是华为近几年推出的一项新兴免安装应用，它的开发也是华为联合了其他几大手机厂商共同推出的开发生态，用户即点即用，无需下载安装，同时软件自动更新，这是一项新兴的开发技术，使用新兴技术进行开发本身就具有一定难度。

华为快应用开发 ide 使用 `template` 模板和 `style` 样式作为前端页面渲染的工具，同时使用 `JavaScript` 脚本进行逻辑交互控制，开发中只能使用华为提供给你的组件和 `api` 进行编程开发工作，具有一定难度，在此次图片社区项目开发过程中，用户发布动态，即动态管理，图片的编辑，上传和下载，以及浏览等功能是项目的核心，使用华为提供的 `api` 调用手机完成拍照或图片选择，同时使用 `ai` 能力获取美学评分，完成动态发布工作，以及各个用户的动态管理后端编程实现，这些功能实现起来都比较困难，本小组按时完成了项目开发工作并顺利完成答辩，显示本小组成员也已初步具备解决复杂工程问题的能力，达到毕业要求。

### 5 系统测试

#### 5.1 测试方法

本系统开发主要采用黑盒测试。通过对照需求分析文档，在已知软件系统所应具有的功能上，根据测试来检测每个功能是否都能正常使用。在测试时，不需考虑程序内部结构和内部特性，通过程序接口进行测试，对已经编写好的程序进行运行，查看运行结果，与预期目标进行比对，检查是否满足所要达到的要求，不满足则对代码进行修改或完善。

#### 5.2 测试项目

基于 Huawei FastApp IDE 开发的图片交流社区快应用——堆栈。

#### 5.3 测试约束

- (1) 不同手机设备对应用的兼容程度会对测试结果造成影响；
- (2) 功能中出现的不易发现的无法预料的错误。

5.4 测试环境

表 2.5.1 测试环境

序号	用途	硬件环境	软件环境
1	服务器	CPU: Intel(R)_Core(TM)_i5-7300HQ_CPU_@_2.50GHz 内存: 4.00G 硬盘: 500G	操作系统: Windows 应用软件: Huawei FastApp IDE 数据库: MySql
2	客户机	内存: 4G 硬盘: 64G	android 操作系统

5.5 测试数据项

表 2.5.2 测试数据项

标识符	表名称	描述信息
User	用户信息表	存储用户信息
Post	动态信息表	存储动态内容
Image	图片信息表	存储动态内图片
LikeTable	点赞信息表	存储点赞关系
CommentTable	评论信息表	存储评论内容
FollowTable	关注信息表	存储关注关系

5.6 测试用例

5.6.1 登录模块

测试目的: 验证用户登录的合法性

测试过程:

表 2. 5. 3 登录模块测试用例表

序号	测试用例	预期结果	实测结果	测试状态	错误类型
1	只输入用户名或密码, 点击“登录”	显示警告信息, 登录失败	与预期结果一致	1	无误
2	输入用户名或密码有误, 点击“登录”	显示警告信息, 登录失败	与预期结果一致	1	无误
3	输入正确的用户名和	登录成功, 跳转至	与预期结果	1	无误



图示：

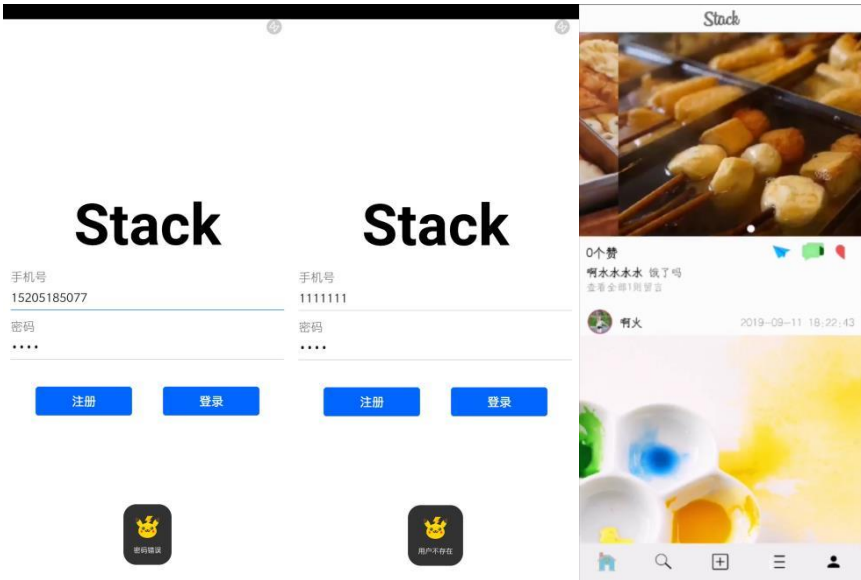


图 2. 5. 1 密码错误    图 2. 5. 2 账户不存在错误    图 2. 5. 3 登录成功，跳转至社区界面

5.6.2 注册模块

测试目的：验证用户注册的合法性

测试过程：

表 2. 5. 4 注册模块测试用例表

序号	测试用例	预期结果	实测结果	测试状态	错误类型
1	注册信息电话号码栏 输入不为11位的电话 号码	提示电话号码有误， 不能进行注册	与预期结果 一致	1	无误
2	两次密码输入不一致	提示两次密码输入 不一致，不能进行注 册	与预期结果 一致	1	无误
3	电话号码正确，两次 密码输入一致，输入 与获取到的不符的验 证码	提示验证码有误，不 能进行注册	与预期结果 一致	1	无误
4	电话号码正确，两次	提示注册成功，跳转	与预期结果	1	无误

密码输入一致，输入 至登录页面 一致  
与获取到一致的验证  
码

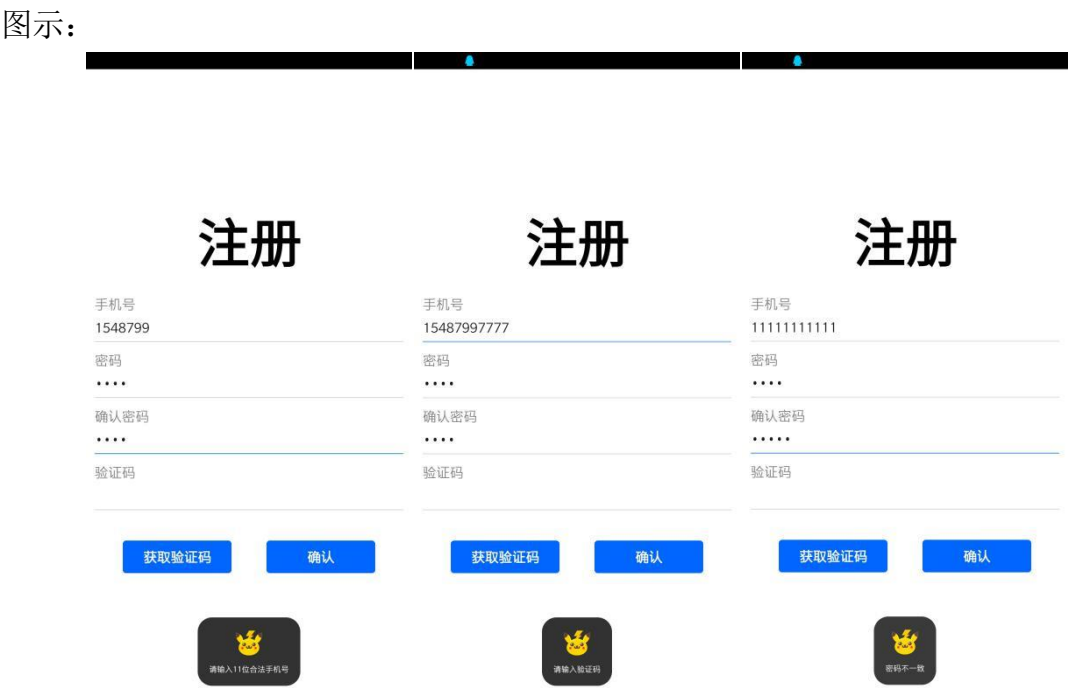


图 2. 5. 4 电话号码格式错误 图 2. 5. 5 未输入验证码 图 2. 5. 6 两次密码不一致

5.6.3 home 社区模块

测试目的：验证点赞、评论、评论数据显示、转发以及点击头像进入个人页面功能

测试过程：

表 2.5.5 社区模块测试用例表

序号	测试用例	预期结果	实测结果	测试状态	错误类型
1	点击“点赞”图标	点赞成功，图标变化，点赞数动态+1	与预期结果一致	1	无误
2	点击“评论”图标	跳转至文字编辑界面，可提交所输入的内容（同一用户可进行多次评论）	与预期结果一致	1	无误

3	点击“show all comments”	展开历史评论数据	与预期结果一致	1	无误
4	点击“转发”图标	显示“转发成功”提示信息,可在个人页面查询到该转发动态	与预期结果一致	1	无误
5	点击用户头像	进入该用户个人主页,显示其动态	与预期结果一致	1	无误

图示:



图 2.5.7 点赞

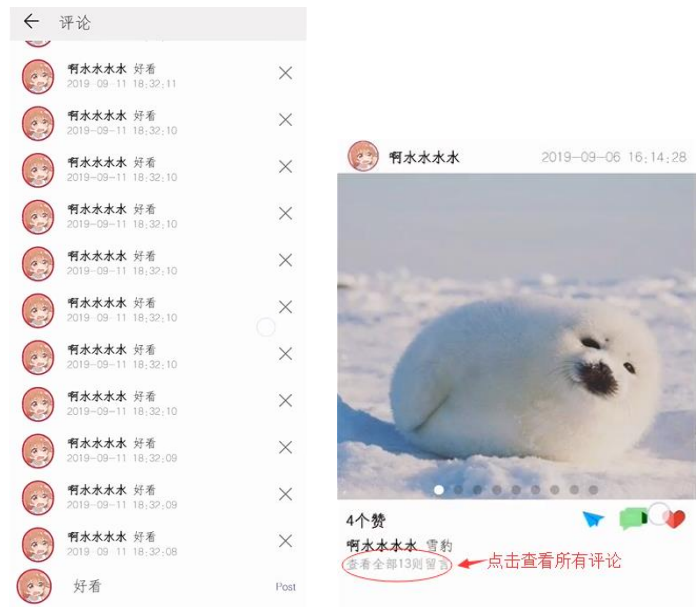


图 2.5.8 评论功能

图 2.5.9 查看历史评论

5.6.4 发现功能模块

测试目的：验证推荐、用户及图片搜索功能

测试过程：

表 2.5.6 发现功能模块测试用例表

序号	测试用例	预期结果	实测结果	测试状态	错误类型
1	点击主页面“+”（发现）图标	系统显示推荐图片，点击图片进入相应动态页面	与预期结果一致	1	无误
2	点击“搜索框”，切换“账户”和“标签”两个选项卡，输入对应搜索内容	根据不同选项卡，页面显示初始推荐内容，有输入搜索内容则显示搜索后所得结果，点击结果即可进入动态或主页	与预期结果一致	1	无误

图示：



图 2.5.10 用户初始“发现”页面

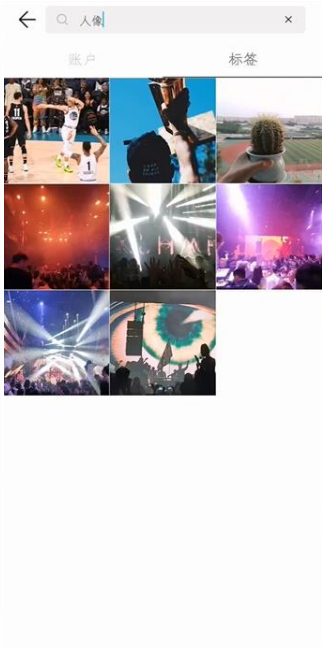


图 2.5.11 点击搜索框初始页面

图 2.5.12 搜索内容

5.6.5 发布动态模块

测试目的：验证用户动态发布及个人页面动态刷新

测试过程：

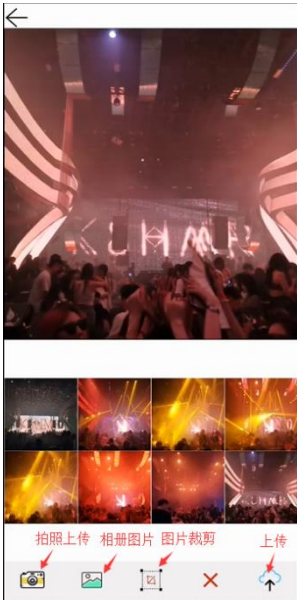
表 2. 5. 7 发布动态模块测试用例表																	
序号	测试用例	预期结果	实测结果	测试状态	错误类型												
1	点击主页面“动态发布”图标，拍照上传图片，相册选择多张图片，对部分图片进行裁剪，点击“上传”图标	上传过程显示进度条，提示上传成功，跳转至 home 社区页面	与预期结果一致	1	无误												
<div>图示：</div> <div></div> <div>图 2. 5. 13 发布动态</div> <div>5.6.6 排行榜模块</div> <div>测试目的：验证动态“日榜”及“周榜”正常显示</div> <div>测试过程：</div> <div>表 2. 5. 8 排行榜模块测试用例表</div> <table><tr><th>序号</th><th>测试用例</th><th>预期结果</th><th>实测结果</th><th>测试状态</th><th>错误类型</th></tr><tr><td>1</td><td>点击主页面“排行榜”图标</td><td>显示动态“日榜”及“周榜”</td><td>与预期结果一致</td><td>1</td><td>无误</td></tr></table> <div>图示：</div>						序号	测试用例	预期结果	实测结果	测试状态	错误类型	1	点击主页面“排行榜”图标	显示动态“日榜”及“周榜”	与预期结果一致	1	无误
序号	测试用例	预期结果	实测结果	测试状态	错误类型												
1	点击主页面“排行榜”图标	显示动态“日榜”及“周榜”	与预期结果一致	1	无误												



图 2.5.14 排行榜

5.6.7 个人页面模块

测试目的：验证个人信息修改及个人历史动态浏览功能

表 2.5.9 个人页面模块测试用例表

序号	测试用例	预期结果	实测结果	测试状态	错误类型
1	点击主页面“个人”图标	浏览个人所有历史动态,可进行个人信息修改	与预期结果一致	1	无误
2	点击“修改个人信息”	进入个人信息页面,对内容修改,提交,个人信息更新	与预期结果一致	1	无误

图示：



图 2.5.15 个人界面

## 5.7 测试结果

### 5.7.1 缺陷情况

(1) 动态中评论预览过少，筛选算法有待优化（参考新浪微博以及 ins 的动态评论）；

(2) 动态中上传的动图预览状态下正常显示，查看大图后变为静态。

### 5.7.2 建议

- (1) 建立图片审核系统，对用户发布的动态进行审核；
- (2) 举报系统，对不文明行为用户进行举报，限制其部分功能；
- (3) 添加用户动态观看权限（参考微信朋友圈之类）；
- (4) 完善缺陷内容。

## 三 企业实习总结和分析

### 1 实习单位及岗位简介

- 实习单位：华为技术有限公司
- 岗位简介：（通用）软件开发工程师

从事 IT 应用层软件、分布式云化软件、互联网软件等的设计开发，可以采用敏捷、Devops、开源等先进的软件设计开发模式，接触最前沿的产品和

软件技术，成为大容量高并发技术的专家；你将参与华为产品的软件研发工作，包括但不限于：

- 完成从客户需求到软件产品定义、架构设计、开发实现、再到上线运营维护等产品生命周期中的各个环节；
- 创造性解决产品在实现过程中的技术难题，应用前沿技术提升产品的核心竞争力，如分布式系统、性能调优、可靠性、数据库等；
- 有机会参与业界前沿技术研究和规划，参与开源社区运作，与全球专家一起工作、交流，构建华为在业界影响力。

## 2 参观企业对本专业人才的具体要求

- 计算机、软件、通信等相关专业本科及以上学历；
- 热爱编程，基础扎实，熟悉掌握但不限于 JAVA/C++/Python/JS/HTML/GO 等编程语言中的一种或数种，有良好的编程习惯；
- 具备独立工作能力和解决问题的能力、善于沟通，乐于合作，热衷新技术，善于总结分享，喜欢动手实践；
- 对数据结构、算法有一定了解；
- 优选条件：
  - 熟悉 TCP/IP 协议及互联网常见应用和协议的原理；
  - 有 IT 应用软件、互联网软件、IOS/安卓等相关产品开发经验，不满足于课堂所学，在校期间积极参加校内外软件编程大赛或积极参与编程开源社区组织；
  - 熟悉 JS/AS/AJAX/HTML5/CSS 等前端开发技术。

## 3 本次实习的个人收获

# 四 毕业要求达成度分析

达成指标	具体的针对性方案	自评等级 (1~5)	综合评定等级 (1~5)
(1.1) 能够在设计环节中体现创新意识；			



<p>(1.2) 在设计/开发解决方案过程中，考虑社会、健康、安全、法律、文化以及环境等因素；</p>			
<p>(1.3) 能够在计算机领域复杂工程问题的预测、建模、模拟或仿真、测试、验证和文档编撰过程中，开发、选择与使用恰当的技术、软硬件及系统资源、现代工程研发工具，提高解决复杂工程问题的能力和效率，并能够理解其局限性。</p>			
<p>(2.1) 针对计算机领域复杂工程问题，能够基于计算机领域科学原理对其进行分析和设计，并能够通过理论证明、软件建模、实验仿真等多种科学方法说明其有效性、合理性；</p>			
<p>(2.2) 能够在计算机领域复杂工程问题的预测、建模、模拟或仿真、测试、验证和文档编撰过程中，开发、选择与使用恰当的技术、软硬件及系统资源、现代工程研发工具，提高解决复杂工程问题的能力和效率，并能够理解其局</p>			

<p>限性；</p>			
<p>（2.3）能够在计算机领域工程实践中理解并遵守工程职业道德和规范，履行责任。</p>			
<p>（2.4）具备多学科背景知识，能够在多学科背景下的团队中与团队成员沟通，了解团队成员想法，并能够协调和组织。</p>			
<p>（2.5）能够就计算机领域复杂工程问题与业界同行及社会公众进行有效沟通和交流，包括撰写报告和设计文稿、陈述发言、清晰表达或回应指令。</p>			
<p>（2.6）能够在多学科环境中应用工程管理原理与经济决策方法</p>			
<p>（3.1）掌握基本的社会、身体和心理健康、安全、法律等方面知识和技能，了解计算机领域活动与之相关性，理解计算机相关领域工程实践中应承担的社会责任；</p>			

<p>（3.2）在计算机相关领域开展工程实践和复杂工程问题解决过程中，能够基于计算机工程领域相关背景知识进行合理分析，思考和评价工程对社会、健康、安全、法律以及文化的影响；</p>			
<p>（3.3）了解信息化与环境保护的关系，能够理解和评价复杂问题的计算机工程实践对环境、社会可持续发展的影响；</p>			
<p>（3.4）对计算机领域及其行业的国际发展趋势跟踪了解，具备一定的国际视野，具有良好的外语听、说、读、写能力，能够在跨文化背景下进行沟通和交流；</p>			
<p>（3.5）了解计算机技术发展中取得重大突破的历史背景，以及当前发展的热点问题，了解信息技术发展的前沿和趋势。</p>			
<p>（3.6）具有自主学习和终身学习的意识，不断学习并适应新的热点或者运用现代化教育手段学习新技术、新知识，具有不断学习和适应计算机技术快速发展的能力。</p>			

## 五 实习心得体会（至少 400 字）

主要完成了 XXX 工作，在工作中遇到了哪些困难，通过上述工作，学习到了 1)。。。 2)。。。 3)。。。 (5 条以内)，认识到还有哪些不足 1)。。。 2)。。。 3)。。。 (5 条以内，包括专业、管理、职业等方面)，今后打算在哪些方面进一步努力。

### 格式说明：

正文小四，中文宋体，西文 Times New Roman；行间距 1.5，首行缩进 2 字符；  
图表中文字：五号宋体，有图名和表名，图名在图下面标注，表名在表上面标注，  
数据表格一律采用三线表。

### 指导教师评语：

成绩评定：

指导教师签字：

年 月 日

注：实习结束时由实习学生填写本表后，交指导教师验收审核，最后交学院归档保管

