Daily6+1 系统

详细设计说明书

文件状态： [ ] 草稿 [√] 正式发布 [ ]正在修改	文件标识：	
	当前版本：	1.2
	作　　者：	潘晨宇、邵研、陈炎、陈静、马科学、林泠、林杰
	完成日期：	2020/4/9

版 本 历 史

版本/状态	作者	参与者	起止日期	备注
1.0	潘晨宇		2020/4/3-4/4	开始编写
1.1	潘晨宇		4/4-4/6	添加详细系统设计方案
1.2	陈炎、陈静、马科学、邵研		4/6	

修 改 记 录

日期	修订版本	修改章节	修改描述	作者
2020/4/4	1.0	4	编写系统方案	潘晨宇
2020/4/6	1.1	5	完整系统后端接口细节	潘晨宇

目录

1 引言

1.1 编写目的

说明WebApp------daily6+1软件系统的各个层次以及层次中、模块、程序的 设计考虑。作为组成员编写代码的文档（理论）依据。

重点展示说明功能模块的分解、代码层次结构、前后端接口、系统和模块的 执行流程。

1.2 背景

1. 软件系统名称：Daily 6+1
2. 系统类型：WebApp
3. 系统从属：无从属关系。会关联到第三方系统，如百度地图、谷歌识别等。分为前台子系统与后台子系统
4. 项目提出者：潘晨宇
5. 开发项目组名称：Daily6+1
6. 软件使用硬件背景：支持HTML5网页的所有设备。

1.3 参考资料

《软件系统设计说明书》 作者不详。

《后台管理系统接口需求分析》 --白雪公主960（简书）

1.4 术语定义及说明

Webapp:web application，基于WEB端的应用网页或网站。

PC：personal computer，家用电子计算机。

mac系列操作系统：MAC OS操作系统，指苹果系列电脑上的操作系统。

JDK：JavaSEDevelopmentkit，java开发工具包.

IDE：Integrated Development Environment，集成开发环境。

MVVM：Model-View-ViewModel。框架开发逻辑。

2 设计概述

2.1 任务和目标

2.1.1 需求概述

主要业务需求：可以发布生活分享贴子，有时间轴可以固定动态的webapp。

- **登录部分**：
    - 输入：账号、密码。
    - 输出：登陆成功/登陆失败。
    - 主要功能：用户登录平台。
    - 性能：需求时间时效性较快，占取空间尽可能小。保证信息安全性。
- **平台部分**：
    - 输入：帖子内容/评论内容/搜索内容
    - 输出：新建帖子/新建评论/显示搜索结果
    - 主要功能：与帖子、动态进行交互。
    - 性能：需求时间反应很快。保证内容和权限的安全性。保证并发性。
- **地图部分**：
    - 输入：搜索的tag
    - 输出：地图气泡高亮
    - 主要功能：展示不同地区的帖子数量。
    - 性能：时间反应尽可能快。保证并发性。
- **时间轴部分**：
    - 输入：新建的时间轴动态信息/时间轴筛选信息
    - 输出：新建时间轴动态/展示时间轴动态
    - 主要功能：展示限定条件下的时间轴动态。
    - 性能：筛选时不改变空间占有率。
- **个人主页部分**：
    - 输入：个人信息
    - 输出：更新个人信息

• 主要功能：展示关注列表，更新个人信息。

• 性能：时间耗费要很短，瞬间显示出所需界面。

主要性能需求：

• 所有功能都要求能尽快响应，时间反馈要足够快。

• tag的添加这一功能要求的及时性不高，需要发布者的确定，所以必定 会有一定的延时性。

• 地图部分的气泡染色可能会因为使用人数的增多导致占用空间过大，并 且直观性受到影响，所以要定时清空气泡内数据，保证空间的充足性。

• 软件主要数据是用户的发帖信息，要保证数据安全性，防止越权修改其 他用户的内容。

• 保证用户登录信息的保密性，可以用哈希码加密后保存。

• 其他数据保证空间使用的效率。数据库中数据能达到第三范式，减少数 据的冗余性，保证空间的利用率

• 对所有输入输出，前端应该有足够的细节规范，防止数据库注入攻击和 XSS 前端注入攻击。

2.1.2 运行环境概述

运行硬件：PC机。能够正常运行HTML5网页即可。

运行软件：支持HTML协议的主流浏览器。

操作系统：windows系列操作系统，mac系列操作系统，移动端支持网页浏览的操作系统。

数据库系统：MySql 5或以上

2.1.3 条件与限制

• 小组成员对协议了解不足，对网络防火墙以及HTML以外协议的了解不足。

• 经济条件限制服务器规模，难以满足大量数据的 并发性交互。

• 技术上难以保证网络防毒、internet 互联地区或者内网地区的交互安全。

• 开发时间限制在一个半月内。

2.1.4 详细设计方法和工具

使用UML系列图讲解需求（包括类图、用例图、状态转换图）。使用类图其中的属性和操作设计对应的操作函数，根据类图中不同功能模块和系统的关联设计接口和调用方法。

使用数据流图展示系统内数据的流动。

选用UMLstar作为统一的图模型工具。

HIPO图作为功能模块的整理。

3 系统详细需求分析

3.1 详细功能需求分析

登录部分：

- 输入账号密码，正确时可以登录进入平台界面。
- 输入账号密码错误时给予提示。
- 当用户忘记密码的时候，可以选择点击按钮后，通过验证后重置密码。
- 当用户没有账号密码的时候，可以点击注册按钮，输入相关信息，如用户名，密码等信息后可以注册一个全新的账号。

平台部分：

- 展示动态，分为广场动态和关注列表动态。
- 动态分为实名动态和匿名动态由用户发布，动态允许用户评论，评论与动态的匿名实名状态由用户当前所处的状态决定。
- 只有发帖用户和管理员可以对发出的帖子进行删除和修改的操作。
- 在发动态用户的允许下，可以给予其他用户添加Tag的权限。
- 展示用户最近发的帖子。

广场动态目前是所有用户的动态按时间顺序展示，展示动态管理员审核和举报功能，后期考虑使用热度算法进行排序展示。

关注列表动态只展示关注用户的动态。用户能对动态进行点赞，转发和评论（评论分为实名评论和匿名评论）。

地图部分：

- 数据可视化，显示广地区（全国、省份地图时）的动态发布数量。
- 街道地图被分为地图块。
- 用户发布的动态会以气泡的形式展现在地图上。
- 搜索找到某个地区中有对应内容时，高亮地区。
- 点击气泡可以进入对应地区的搜索结果界面。

时间轴部分：

- 起始时间为建号开始，持续进行，按照时间顺序显示时间轴的动态。

• 初始展示全部分类的集合时间轴。

• 选择不同的分类和筛选时间展示不同类别的时间轴动态，可以查看时间轴动态内容，对时间轴内容进行添加和修改。

• 可以按照时间筛选时间轴展现的内容。

个人主页部分：

• 展示自己的动态，搜索已发布的帖子。

• 关注列表，可以跳转到被关注的地方。

3.2 详细性能需求分析

数据性能

- 除去事务性数据，还要结合百度地图API，支持地图数据。
- 数据量至少能达到400个。
- 数据库要保证使用规范化的MYSQL语言，方便拓展和移动。
- 有可靠的数据安全保密措施以及故障恢复能力。
- 具有完整的安全性（帐号安全，系统级权限，对象安全性，审查等），细粒度化的访问控制，适合于多层环境的安全模式的能力；

并发性

- 至少能够支持100个用户同时访问。
- 并发传输提供尽可能快速的传输效率，支持多图片的上传和下载。
- 气泡功能部分要定时清空以保证数据的实时性和直观性。

响应特性

- 数据查询查询时间应该小于5秒
- 数据添加修改过程不超过15秒，复杂添加修改不超过5分钟。

架构特性

稳定性

- 保证能够提供7*24H的稳定服务
- 保证前端界面和后端数据的稳定性

兼容性

- 前端能够兼容大部分主流浏览器

扩展性

- 便于新业务或者新功能的生成和实现第三方系统与平台的连接。
- 包括扩充性和开放性，保证功能添加和平台、系统连接不会出问 题。

经济性

- 系统应具备高性价比，能对系统资源的使用进行优化，在实现系 统功能的前提下，尽量节省硬件资源的开销。

模块性

• 系统须提供通用的组件支持，能够减少重复开发工作，保证产品　和项目的质量，缩短应用系统的开发周期，有利于系统的扩展。

维护性

•方案和产品的架构须紧密跟踪国家信息安全、业主标准和国际主流技术标准，开放性好，便于系统的升级维护、以及与各种信息系统进行集成。

3.3 详细资源需求分析

硬件资源

- 能够运行主流HTML浏览器的硬件支持。（内存:2G，CPU：i3以上）
- 能够连接因特网的PC机。
- 能够允许一定访问量的服务器（并发100次）
- 能够保存一定数据量的服务器数据储存空间。（至少256G）
- 用以备份的硬盘。（至少256G）

软件资源

- 主流操作系统，包括window 98以上，MAC OS等。
- 1.8版本JAVA的JDK。
- 前端框架Vue以及后端框架Spring boot。
- 能够编写前端以及后端的IDE，如eclipse,webStrom,IDEA。
- 与MySql数据库连接的接口。

3.4 详细系统运行环境及限制条件分析、接口需求分析

系统运行环境：

软件系统：能够运行主流浏览器的系统平台，WINDOS 98以上，MAC 操作系统，LINUX等。

硬件系统：内存2GB以上。

CPU酷睿icore系列 i3 或以上。

鼠标/键盘：可正常交互鼠标及键盘。

数据库系统：MySql 5以上。

系统接口需求分析：

1. 地图功能模块需要地图信息，需要外部地图接口API提供。
2. 搜索功能可能会支持搜图，需要识图智能的API。
3. 前/后台子系统之间会有用户数据流和帖子数据流的传递，需要接口。

4. TCP/IP通信协议接口。

5. 前端渲染所需数据来自于后端，需要后端本地接口返回数据，详细设计见本说明书5.4系统界面设计。

引进软件资源接口：

百度地图：

- 需求百度地图提供各级省市的地图渲染接口
- 需求百度地图-Echars插件的地图数据可视化接口

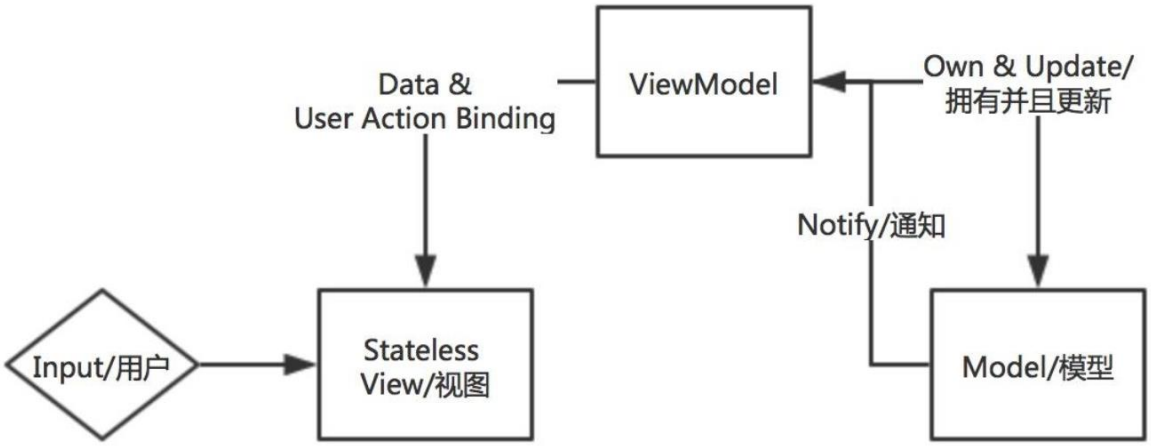
百度识图：

- 需求百度识图“识别主要内容”部分的接口

4 总体方案确认

4.1 系统总体结构确认

- **系统组成**：前台子系统，后台子系统。
- **开发框架逻辑结构**如下



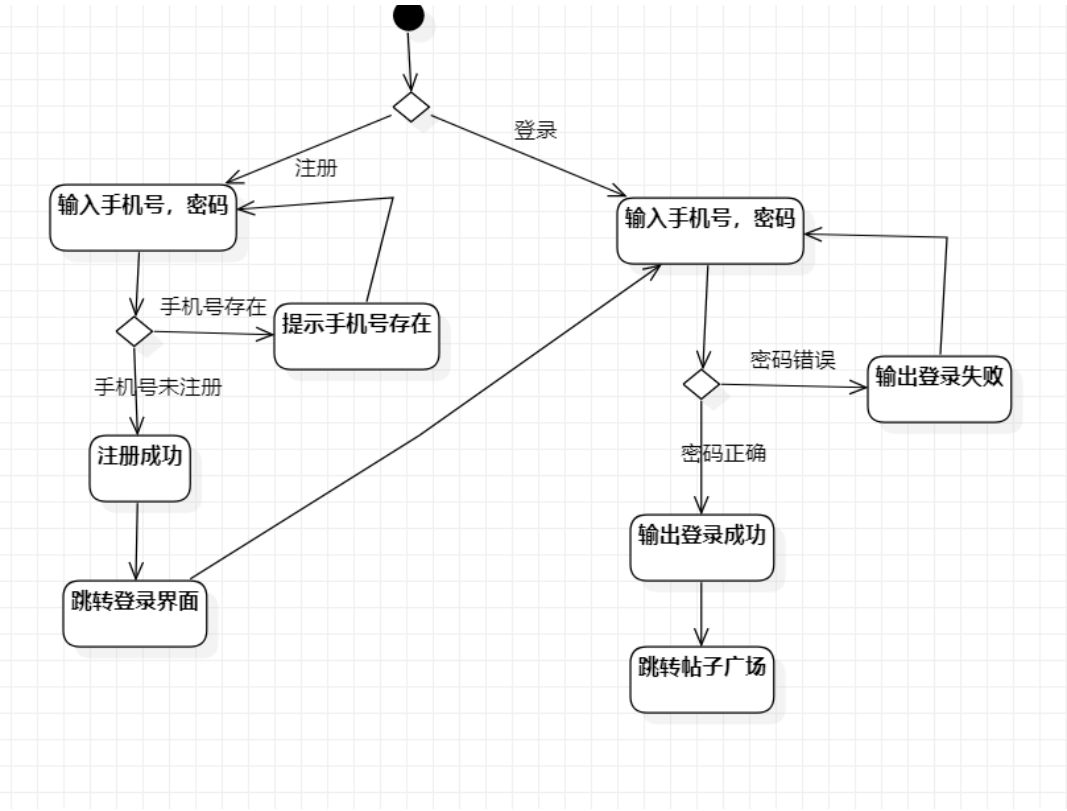
以MVVM框架为主要架构框架。后台系统和前台系统采用相同方式。MODEL层数据由后端逻辑接口得到，前端由ViewModel绑定数据后渲染前台页面(View)。

- **后台子系统**：由登录和审核功能模块构成。为管理员管理帖子和用户提供可视化平台。
- **后台子系统工作流程**：登录->审核->删帖或冻结用户
- **前台子系统**：功能需求中除去后台功能的部分，包括发帖、看帖、回复等等交互功能模块，为用户进行发帖、看帖、回复等功能提供平台。

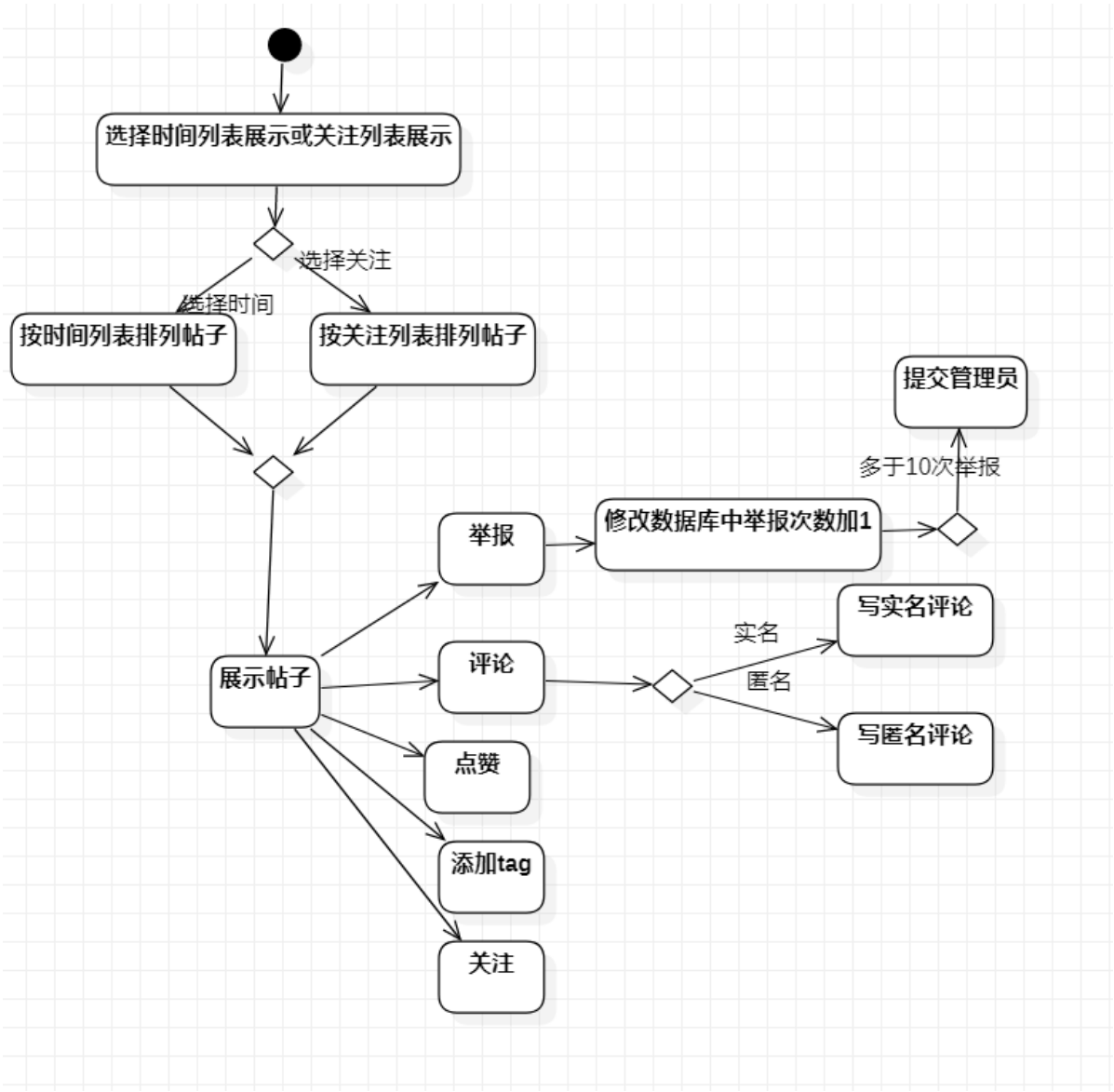
- **前后台系统集成**：通过不同的url地址，进入daily6+1的前/后台登录页面。前后台共用一个数据库数据。

- 前台子系统工作流程：

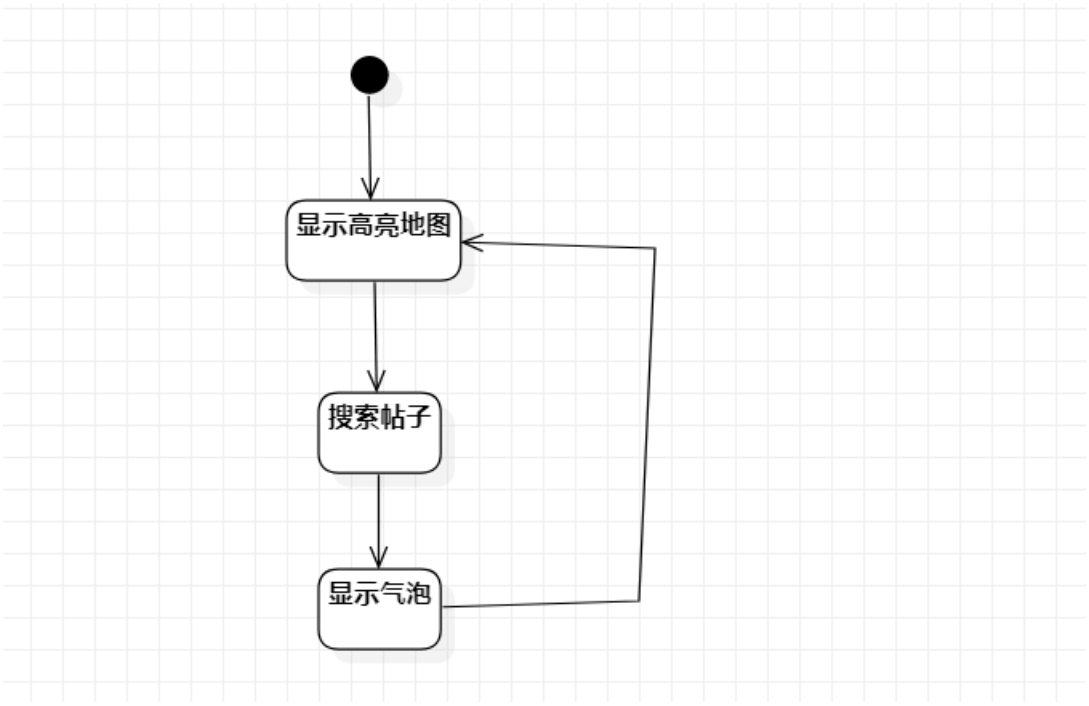
登录：



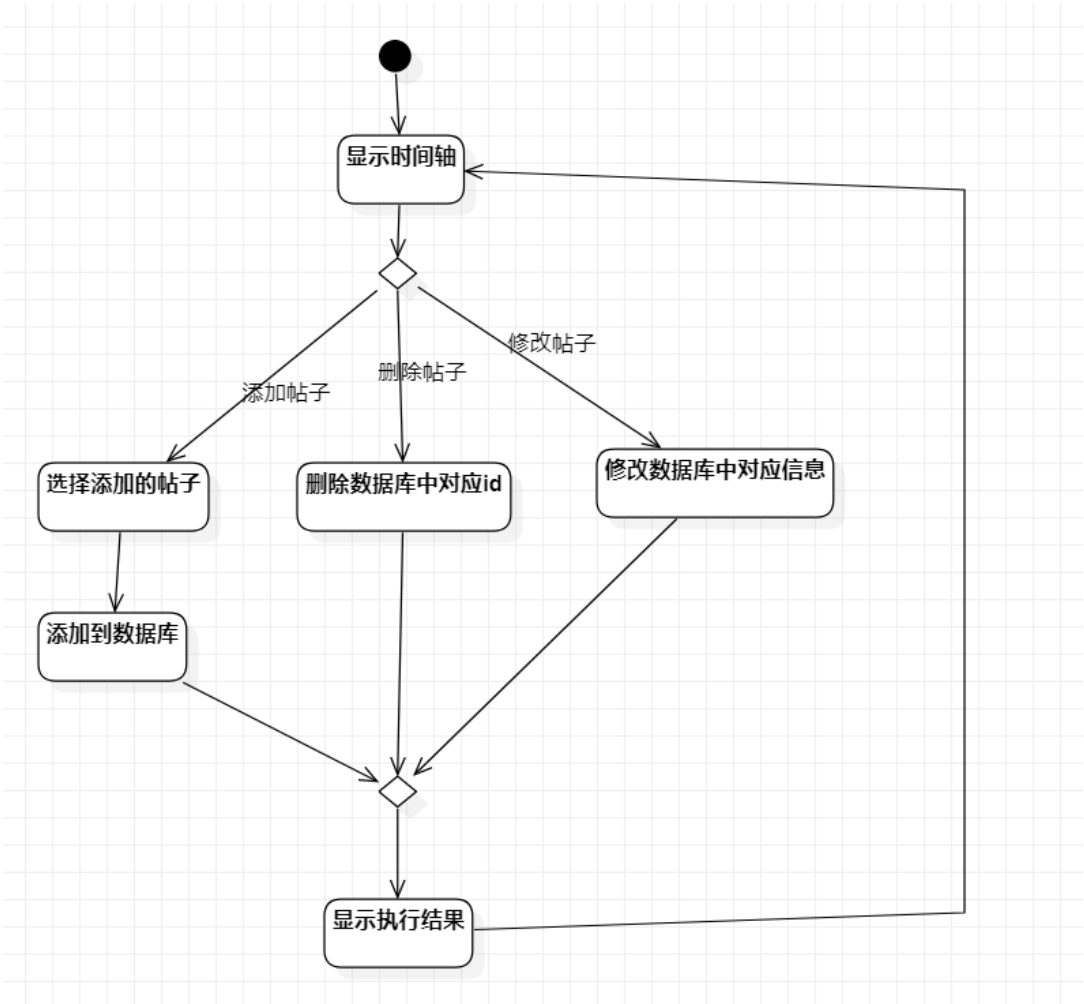
帖子：


选择时间列表展示或关注列表展示
选择关注
选择时间
按时间列表排列帖子
按关注列表排列帖子
提交管理员
多于10次举报
举报
修改数据库中举报次数加1
写实名评论
实名
展示帖子
评论
匿名
写匿名评论
点赞
添加tag
关注

地图：


显示高亮地图
搜索帖子
显示气泡

时间轴：


显示时间轴
添加帖子
删除帖子
修改帖子
选择添加的帖子
删除数据库中对应id
修改数据库中对应信息
添加到数据库
显示执行结果

4.2 系统详细界面划分

4.2.1 应用系统与支撑系统的详细界面划分

本系统前台子系统和后台子系统平行操作。全局数据由数据库存储，统一通过后端调用提供给前台子系统和后台子系统。Daily 6+1应用系统中的数据由数据库系统支撑，应用系统中后端处理数据库中数据，提供给前端用以页面渲染。

数据库中数据不直接填写，而通过前端数据返回给后端后，由后端处理进行更改。特别的，管理员的信息不可申请，而在创建数据库时提供固定数量的管理员数据，不可更新。

数据库系统为MySql5,通过NaviCat作为数据库系统的支撑系统，并通过它对MySql数据库内的数据进行修改。

最终数据库系统和Daily 6+1应用系统部署在服务器上进行。服务器由单一百度云服务器提供服务。服务范围包括webapp的配置以及数据库的配置。

4.2.2 系统内部详细界面划分

- **后台系统以及内部功能模块的交互和调用。**
    - 后台系统覆盖了登录和审查功能模块。
    - **审核功能**应该在登陆功能成功之后才可以执行。
    - 审核功能的数据来源应该来自后端传递的JSON数据文件。
    - 后台系统中的审核功能和登录功能不存在覆盖以及重叠部分。
    - 后台系统中的审核功能模块和登陆功能模块无数据传递和调用。
    - 后台系统允许数据持久储存，要求长久的时效性，在管理员执行完审核之后要求执行时间可以稍长，但不得多于2分钟。
    - 后台登录数据不可申请，只能由数据库内填写，固定分配，保证数据的安全性。
    - **后台管理员权限**不可更改。
    - **冻结用户**需要后台子系统向后端发出请求，调用后端方法，将用户目前的数据状态冻结，并且设为不可登录。
    - **删除帖子**需要后台子系统向后端发出请求，调用后端方法，删除未能通过审核的帖子。
- **前台系统以及内部功能模块的交互和调用。**

- 前台系统的各个功能应该在登陆成功之后才可以执行。
- 地图模块的地图数据来源于百度地图的API。
- **平台功能模块**中，帖子的内容等数据由后端提供。平台功能模块覆盖了展示、新建、删除、查看帖子等功能模块
- 平台功能模块中，新建模块可以向展示模块提供新建的帖子信息。同理删除时也应该提供对应信息，使得展示模块不展示已删除的帖子。
- 平台功能模块中，变更匿名/实名状态时，应该向个人状态模块传递信息，变更个人状态，变更完成时，要使得新建和回复功能模块功能改变。
- 平台功能模块中，查看帖子时，需要向后端发出申请，后端返回对应的JSON文件，并且调用前端方法渲染出对应详情。
- **详情界面**中，可以执行评论、点赞、转发功能。前端调用方法后，后端应执行对应数据库操作，改变相关数据。
- **地图模块**中，高亮地图的渲染通过百度地图API处理数据。
- 地图模块中，搜索TAG关键词时，允许气泡高亮，此时需要与平台功能模块交互，调用平台功能模块的搜索功能，搜索完成后将地区ID返回到地图模块，并使得对应气泡高亮，高亮气泡再关联到搜索页面。
- **时间轴模块**中，覆盖了新增、删除、关联、筛选等功能模块。
- 时间轴模块中，新增、删除等模块与平台功能帖子的新增删除类似。
- 时间轴模块中，新增动态时，允许关联到帖子，此时时间轴模块与个人信息模块进行交互。个人信息模块返回个人已发布的帖子，允许用户进行选择。关联完成后，动态应与帖子发生单方面链接，允许从动态跳转到帖子。
- 时间轴模块中，对时间进行筛选时，不与后端进行交互，直接改变页面的渲染内容，只展现对应类别的时间轴动态。
- **个人页面模块**中，覆盖了查询、修改功能
- 个人页面模块中，允许对个人信息进行修改，此时需要反向向后端传递数据，要求后端修改数据库内容。
- 个人页面模块中，允许查看个人发布的帖子和动态。需要调用相应的查询方法。不对数据进行处理，只通过前端页面展示对应的内容即可。

- **全局数据**
  - 通过数据库进行持久化存储。
  - 全局数据只能被后端修改，前端页面不能直接修改数据库内容。

- 全局数据不能被前端直接读取，需要通过后端调用接口后，处理后端返回的JSON数据后渲染

- **前后台交互**

  - 前后台交互仅当前台提交审核申请后执行。前台系统将被举报贴的数据传递给后台系统以审核。

  - 后台审核完成后，可能执行的冻结用户方法和删帖方法会和后端进行交互后处理并更新数据。

- **系统性能要求。**

  - 前后端数据传递，功能模块间功能调用时，响应时间不能超过5S。

  - 并行操作情况下，允许至少100个用户同时操作。

  - 后端数据应该保证安全性，防止数据库注入攻击。

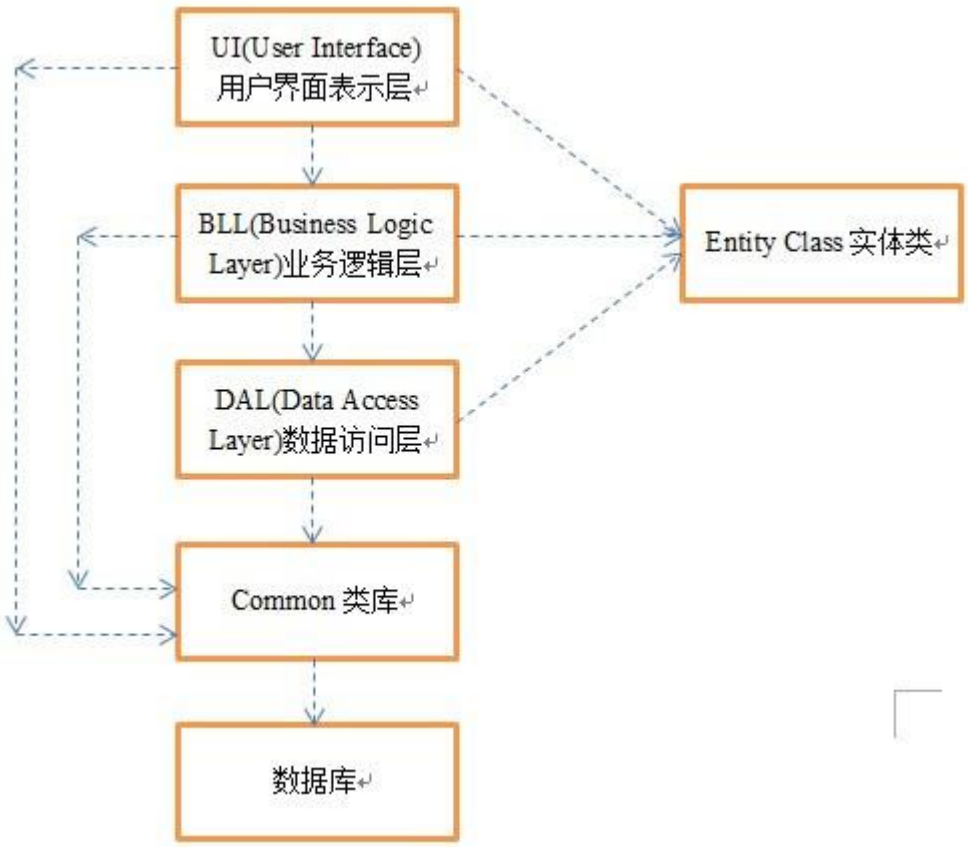
  - 前端页面应该检查输入，防止不合理输入后导致系统崩溃或者数据库数据泄露。

  - 前端代码应该规范，防止XSS的前端注入。

5 系统详细设计

5.1 系统程序代码架构设计

系统采用三层架构模型，将应用系统划分为用户界面表示层、业务逻辑层、数据访问层，以及Entity Class实体类、Common类库组成，各层的关系如下图所示：



UIL内部模块只与BLL(Business Logic Layer)业务逻辑层、Entity Class实体类两个项目发生关联，可能与Common类库发生关联。

BLL层只关联DAL层和实体类，可能关联Common类库。虽然BLL层被U层调用，但是BLL层无需关心UI层的情况。数据库中每个表都对应一个BLL类，为了达到解耦效果，BLL类不能直接调用其他表的DAL类，可以BLL类之间相互调用。

DAL层一般关联Common类库中的最底层，最基础的数据库类（比如：链接数据库），必须关联Entity Class实体类项目。DAL层只是数据库的管理者，但不是访问者，不直接与数据库发生关联。数据库中每个表都对应一个DAL层的接口(访问控制)类。

Common类库用以存放共用类，如数据库连接类等。实体类则是数据库表的实体加强

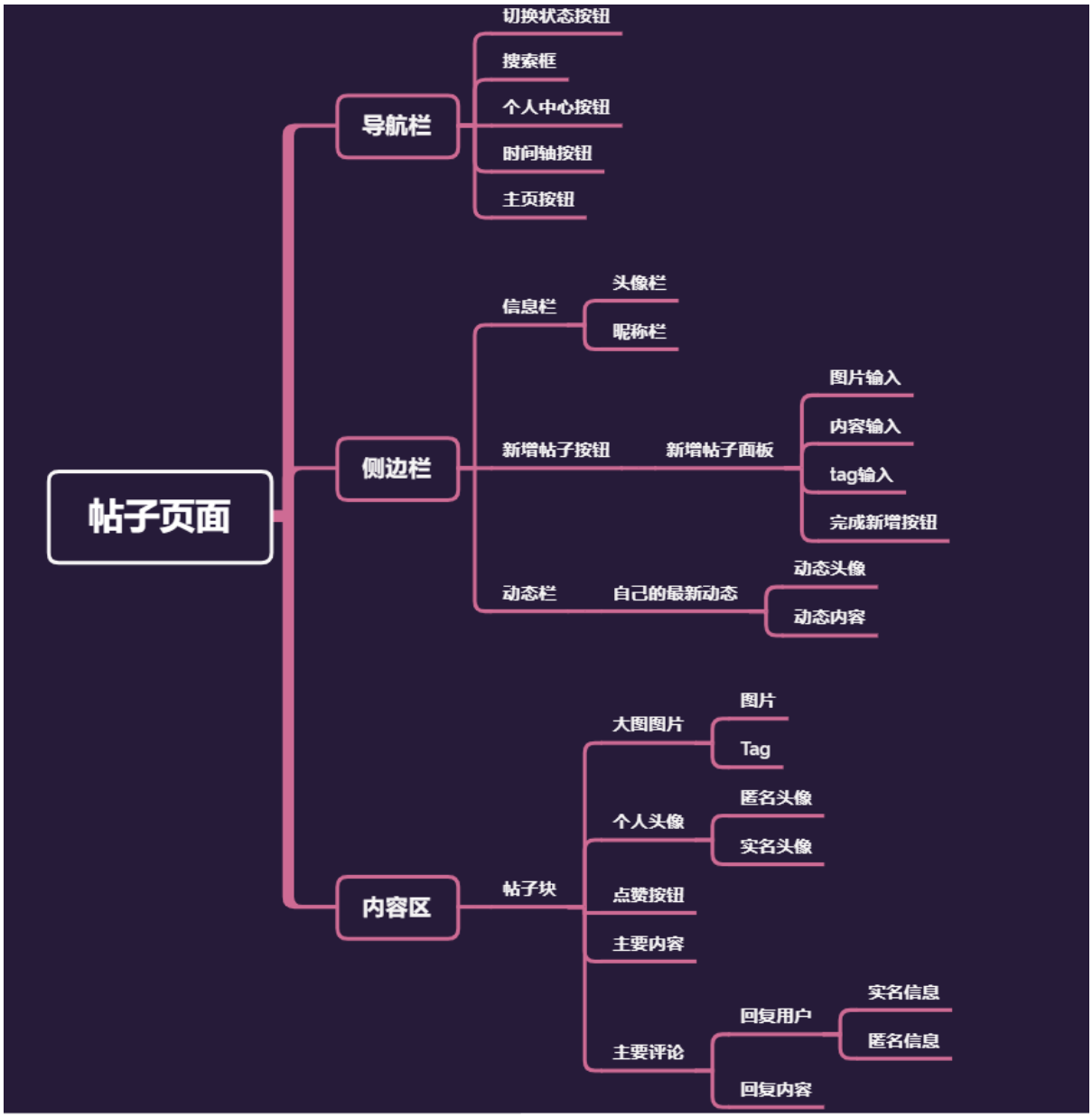
5.1.1 UI(User Interface)用户界面表示层

负责与用户进行交互，显示、接受数据，与此同时，做一些简单逻辑处理，如：输入数据有效性判断、显示各种异常、处理Dataset记录集数据。

本次用户界面表示层，即前端UI表现层，使用VUE框架搭建，以下仅展示本次前端框架中的组件嵌套关系。

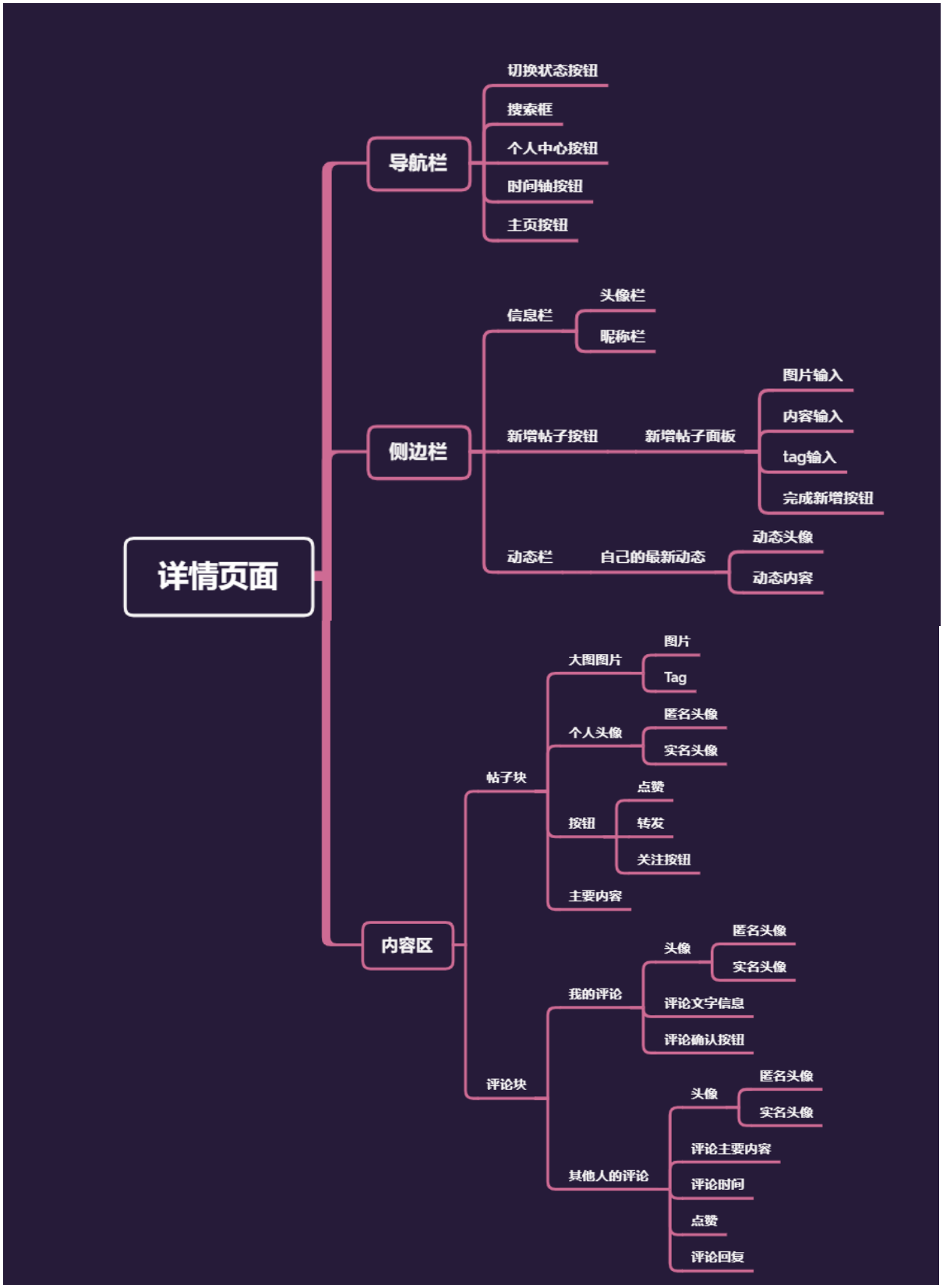
UI层组件类嵌套情况见下：

平台页面

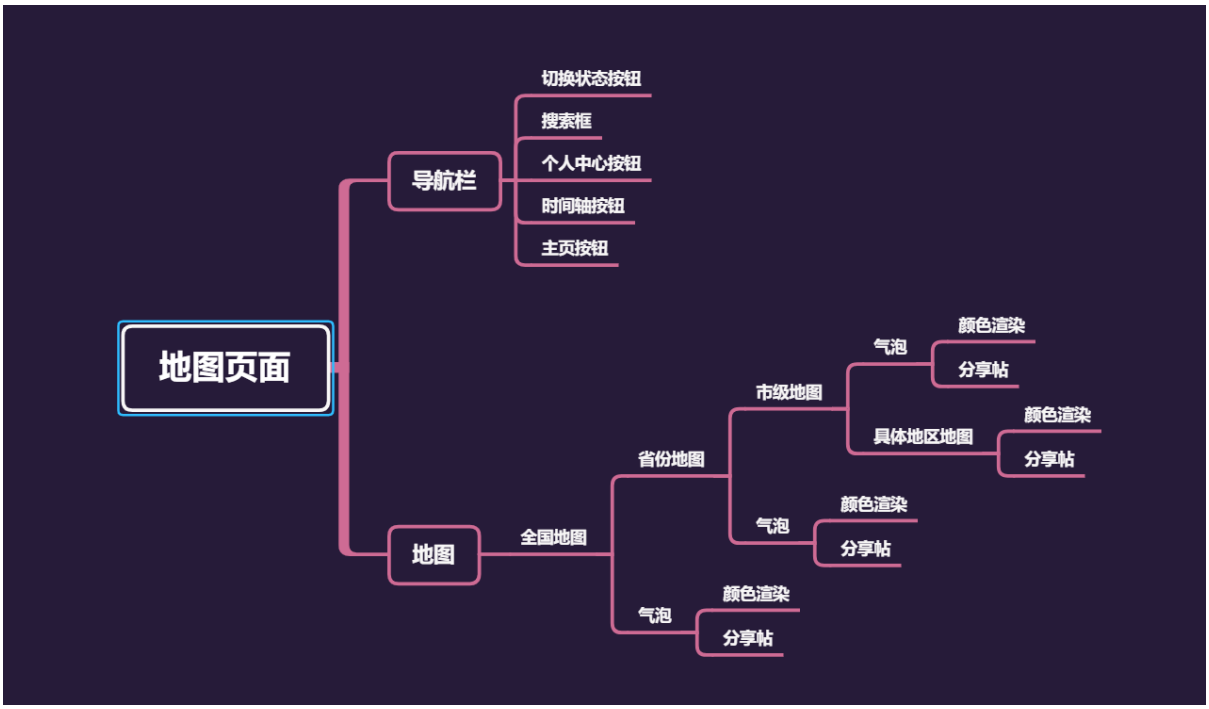


组件可能产生复用情况。

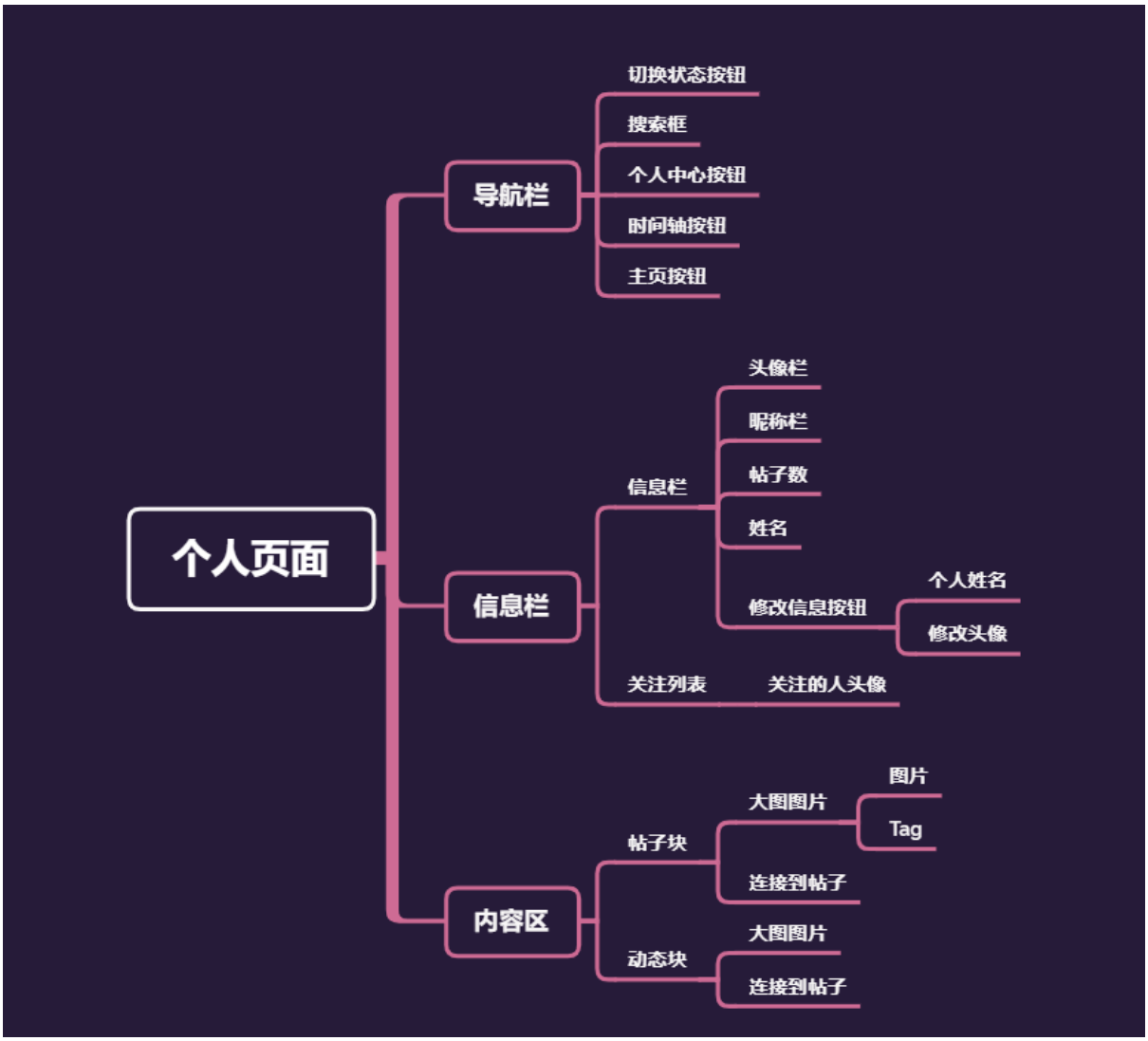
帖子详情页面



图片是拼接完成的，详情需点开大图，重名部分代表组件复用，下不赘述


地图页面
导航栏
切换状态按钮
搜索框
个人中心按钮
时间轴按钮
主页按钮
地图
全国地图
省份地图
市级地图
气泡
颜色渲染
分享帖
具体地区地图
颜色渲染
分享帖
气泡
颜色渲染
分享帖
气泡
颜色渲染
分享帖

地图页面

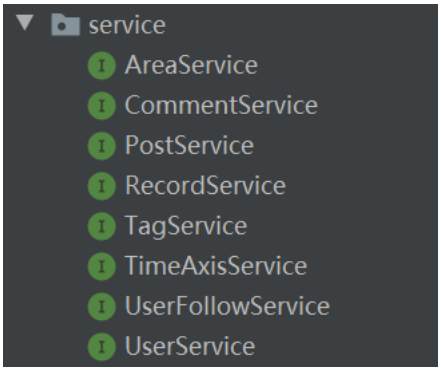

个人页面
导航栏
切换状态按钮
搜索框
个人中心按钮
时间轴按钮
主页按钮
信息栏
信息栏
头像栏
昵称栏
帖子数
姓名
修改信息按钮
个人姓名
修改头像
关注列表
关注的人头像
内容区
帖子块
大图图片
图片
Tag
连接到帖子
动态块
大图图片
连接到帖子

个人主页

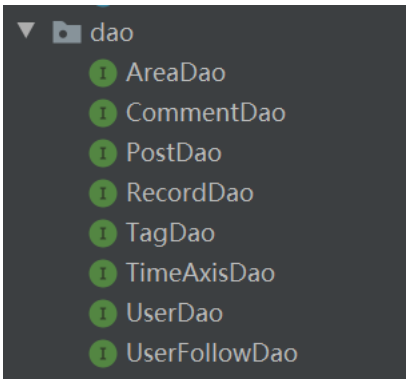
时间轴页面


时间轴页面
导航栏
切换状态按钮
搜索框
个人中心按钮
时间轴按钮
主页按钮
侧边栏
类别栏
全部按钮（默认）
日常按钮
工作按钮
心情按钮
吃喝玩乐按钮
时间栏
日期起始时间输入栏
年输入框
月输入框
日输入框
日期截止时间输入栏
年输入框
月输入框
日输入框
内容区
时间轴主体
动态
日期显示框
内容显示框
图片显示框
删除按钮
查看全文按钮
添加按钮
编辑按钮

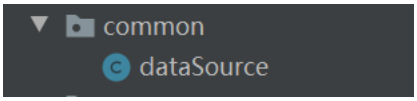
5.1.2 BLL(Business Logic Layer)业务逻辑层



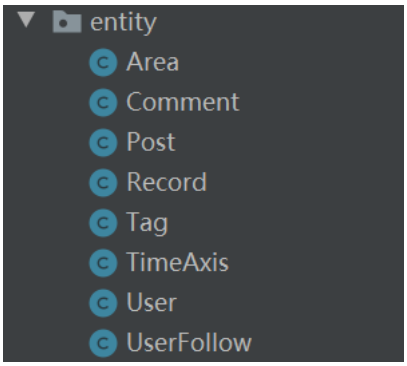
5.1.3 DAL(Data Access Layer)数据访问层



5.1.4 Common类库



5.1.5 Entity Class实体类



Area类：

```
private int areaId; //地区ID
private String areaName; //地区名称
```

Comment类：

```
private int commentId; //评论ID
private String commentContent; //评论内容
private Date commentCreateTime; //评论时间
private Date commentUpdateTime; //评论修改时间
private int postId; //被评论帖子ID
private int userId; //用户ID
private int anonym; //是否匿名评论（是1，否0）
```

Post类：

```
private int postId; //帖子ID
private Date postCreateTime; //发帖时间
private Date postUpdateTime; //更新时间
private int likeNum; //点赞数
private int forwardNum; //转发数
private int commentNum; //评论数
private int tipoffNum; //举报数
private String postImg; //帖子图片
private String postContent; //帖子内容
private int anonym; //是否匿名（是1，否0）
private int areaId; //帖子所属地区ID
private int userId; //发帖人ID
private int forward; //该帖是否为转发帖子（是1，否0）
```

Record类：

```
private int recordId; //动态ID
private String recordImg; //动态图片
private String recordContent; //动态内容
private Date recordCreateTime; //动态创建时间
private Date recordUpdateTime; //动态修改时间
private int TimeAxisId; //动态所属时间轴ID
private int areaId; //动态所属地区ID
```

Tag类：

```
private int tagId; //标签ID
private String tagContent; //标签内容
private int postId; //标签对应的帖子ID
```

TimeAxis类：

```
private int timeAxisId; //时间轴ID
private String timeAxisType; //时间轴类型
private int userId; //用户ID
```

User类：

```
public class User {
    private int userId; //用户ID
    private int userType; //用户类型（普通用户为0，管理员为1）
    private String userName; //用户名
    private String userPwd; //用户密码
    private String userImg; //用户头像
    private String areaId; //用户所属地区ID
    private Date userDate; //用户出生日期
    private String gender; //用户性别
    private String profile; //用户简介
    private int state; //是否可用（是1，否0）
    private int fansNum; //粉丝数
    private int followNum; //关注数
```

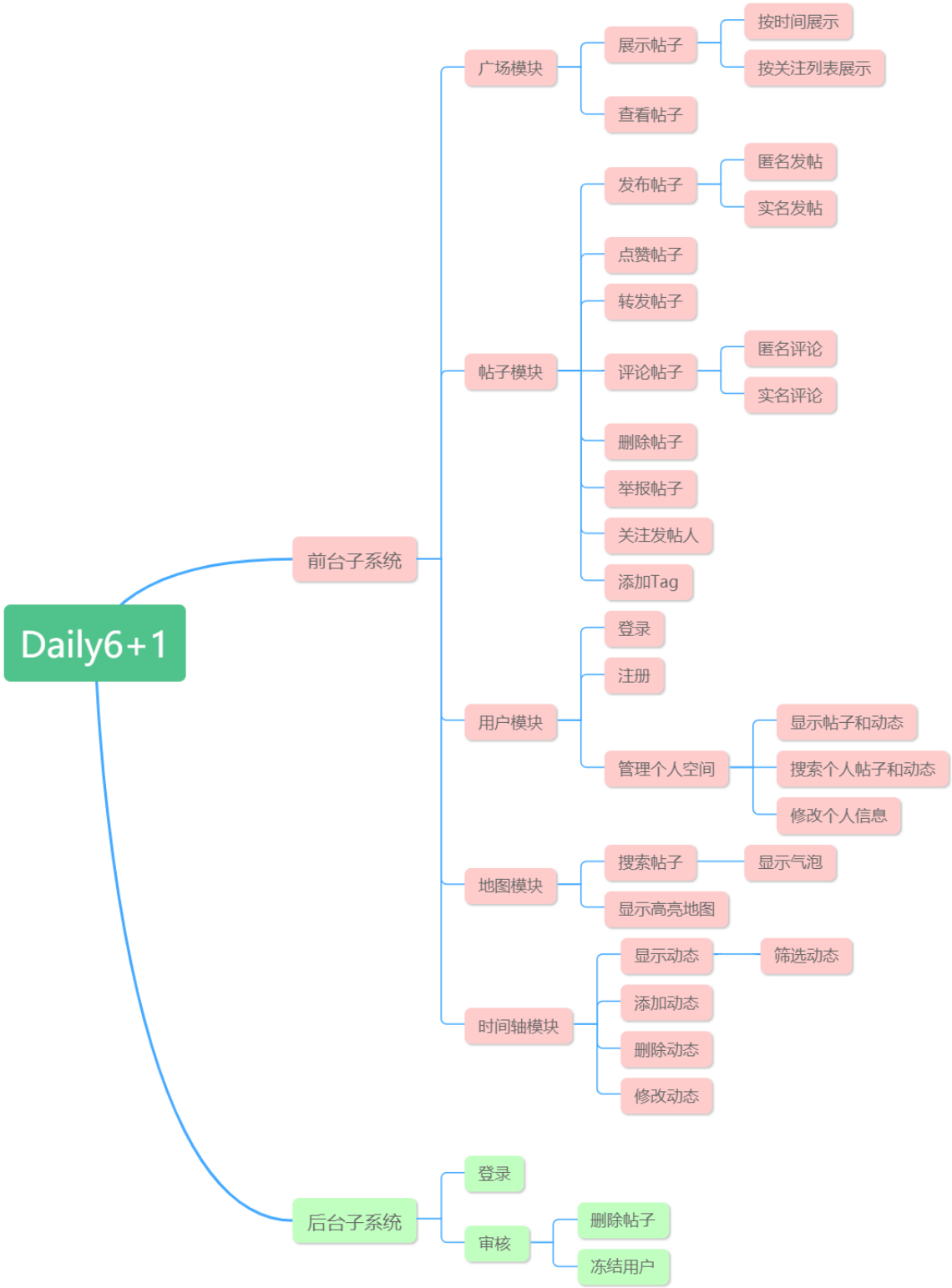
UserFollow类：

```
private int userFollowId; //用户关注表ID
private int userId; //用户ID
private int followId; //关注用户的ID
private Date followTime; //关注时间
```

5.2 系统结构设计及子系统划分

根据业务和功能，将系统的逻辑结构划分为前台管理子系统、后台管理子系统两个子系统，如下图所示：

各个子系统按照功能角度分解，划分出若干不同的功能模块，如下面各图所示：


Daily6+1
前台子系统
广场模块
展示帖子
按时间展示
按关注列表展示
查看帖子
帖子模块
发布帖子
匿名发帖
实名发帖
点赞帖子
转发帖子
评论帖子
匿名评论
实名评论
删除帖子
举报帖子
关注发帖人
添加Tag
用户模块
登录
注册
管理个人空间
显示帖子和动态
搜索个人帖子和动态
修改个人信息
地图模块
搜索帖子
显示气泡
显示高亮地图
时间轴模块
显示动态
筛选动态
添加动态
删除动态
修改动态
后台子系统
登录
审核
删除帖子
冻结用户

（需点开大图查看）

5.3 系统功能模块详细设计

5.3.1 前台子系统

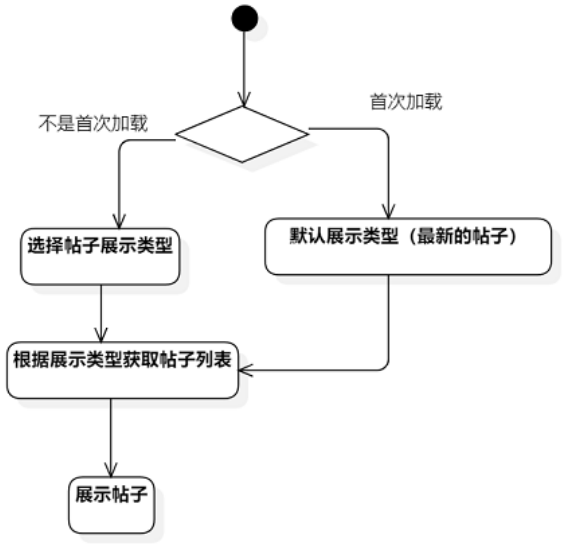
5.3.1.1 广场模块

模块描述：用户在此浏览他人的帖子

主要功能：按用户选择，提供最新/最热门/关注的帖子列表，搜索帖子，查看帖子

5.3.1.1.1

1、流程图



2、输入

展示帖子类型　　String　　非必填

3、输出

成功：按照展示类型的帖子列表，首次加载时默认最新的帖子

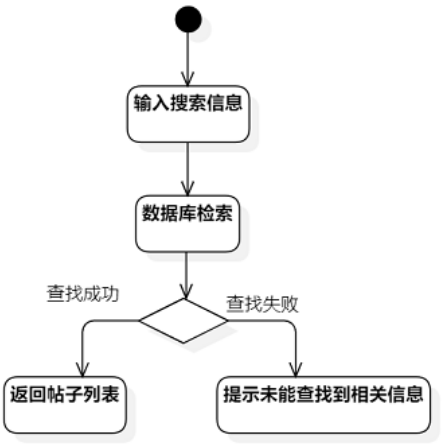
失败：提示暂时无法获取服务器信息

4、算法

从数据库中取出对应的帖子列表

5.3.1.1.2

1. 流程图



2、输入

检索内容　　string　　必填

3、输出

成功：帖子列表

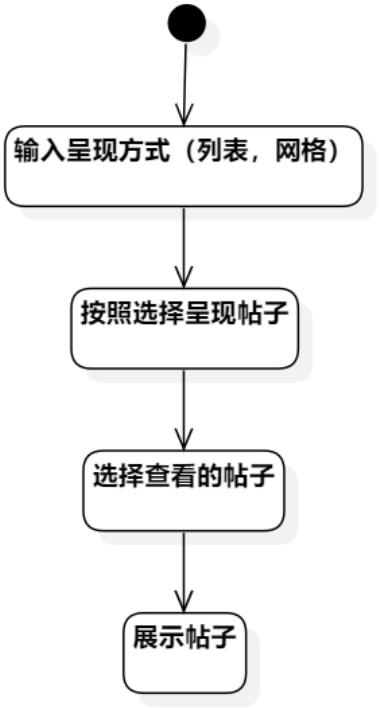
失败：提示未能搜索到结果

4、算法

从数据库中检索待搜索内容，如果有则返回符合检索条件的帖子列表，否则返回false，提示未能找到相关内容

5.3.1.1.3

1. 流程图



2.输入

帖子呈现方式，点击按钮选择

3、输出

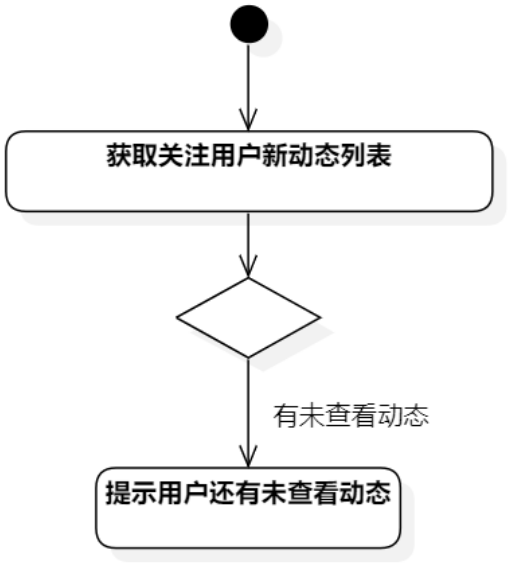
将贴子按展示方式显示给用户

4、算法

用户点击展示方式按钮，切换列表和表格呈现

5.3.1.1.4

1、流程图



2、输入

当前用户登录信息

3、输出

未阅读关注用户的动态列表

4、算法

从记录着未阅读动态的数据库中取出该用户的信息。

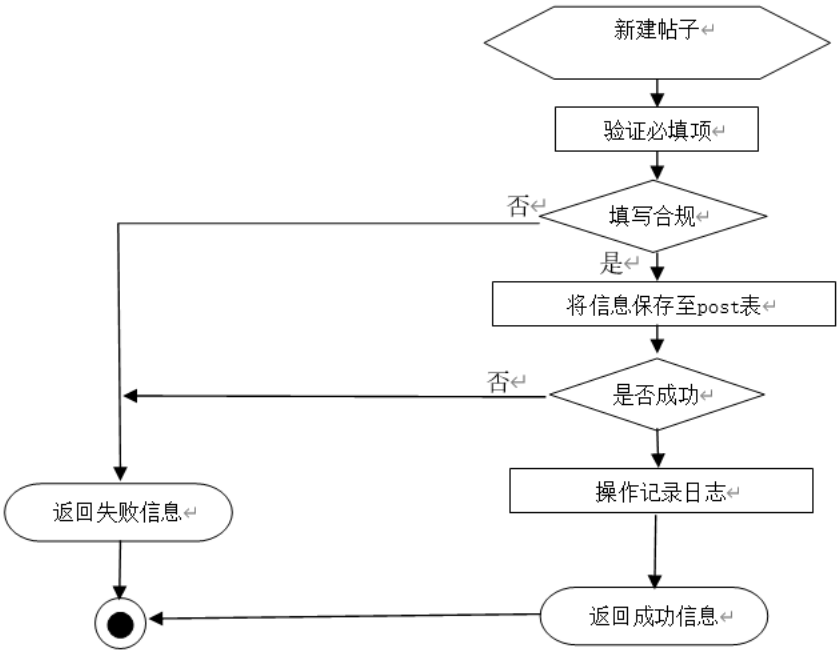
5.3.1.2 帖子模块

模块描述：管理帖子各个功能

主要功能：发布、删除、修改、查找帖子；点赞、评论、转发、举报帖子；加标签

5.3.1.2.1 发布帖子

1、流程图



2、输入

1)	文字	String	必填
2)	图片	String	必填
3)	是否匿名	Integer	必填
4)	是否转发	Integer	必填

3、输出

成功，UI提示发布成功;

失败，UI提示具体信息

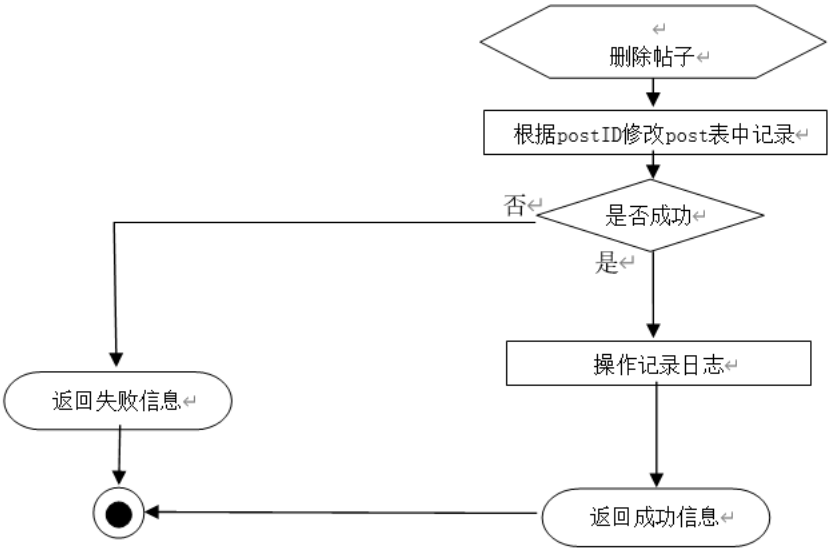
4、算法

1）前端js验证必填项；

2）帖子信息保存至post表。

5.3.1.2.2 删除帖子

1、流程图



2、输入

postId

3、输出

成功，UI提示删除成功;

失败，UI提示具体信息

4、算法

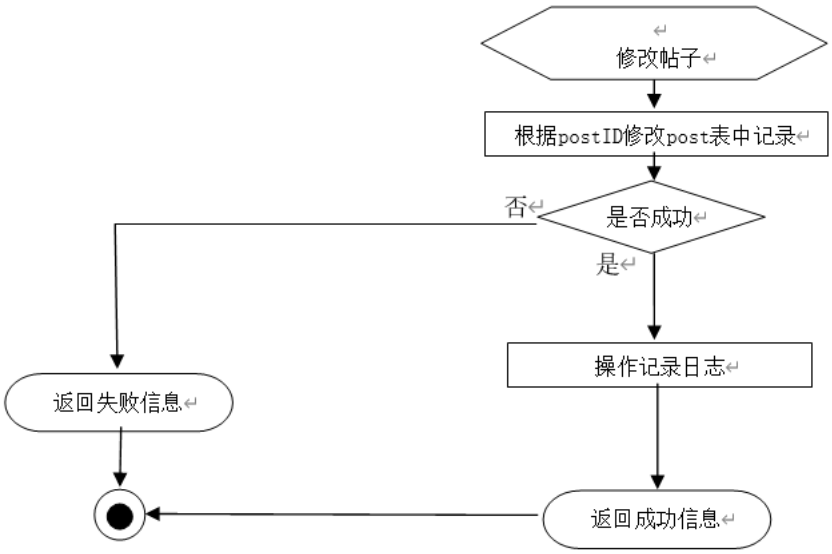
1) 界面获取动态postId

2) 根据postId查找post表，获得相关记录

3) 将该记录从post表删除

5.3.1.2.3 修改帖子

1、流程图



2、输入

帖子ID	Integer	必填
图片	String	
内容	String	
是否匿名	Integer	

3、输出

成功，UI提示修改成功;

失败，UI提示具体信息

4、算法

1) 前端js判断必填项的输入,界面动态获取postId

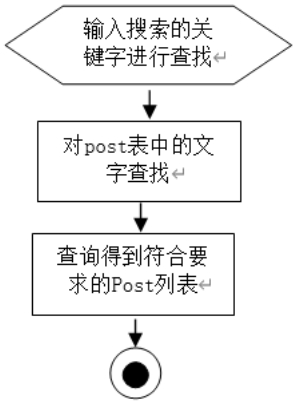
2) 依据帖子id查找post表

3) 获得相关记录后修改post信息

4) post信息存入post表

5.3.1.2.4 查找帖子

1、流程图



2、输入

搜索的关键字

3、输出

成功，输出符合条件的帖子;

失败，UI提示具体信息

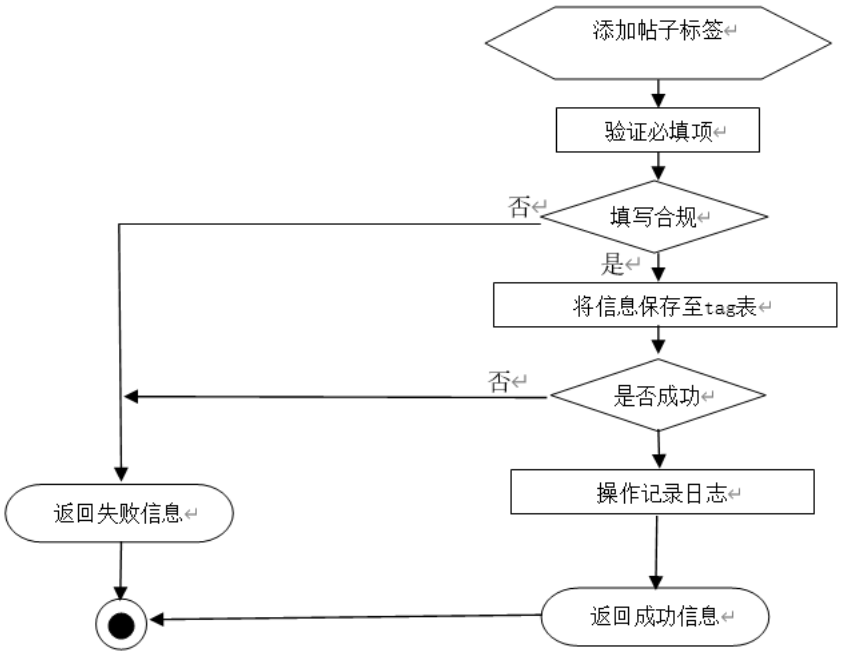
4、算法

1）根据搜索关键字查询对应帖子

2）返回符合需求的帖子

5.3.1.2.5 添加标签

1、流程图



2、输入

TagContent:String,必填

Postid:Integer,必填

3、输出

成功，UI提示添加标签成功;

失败，UI提示具体信息

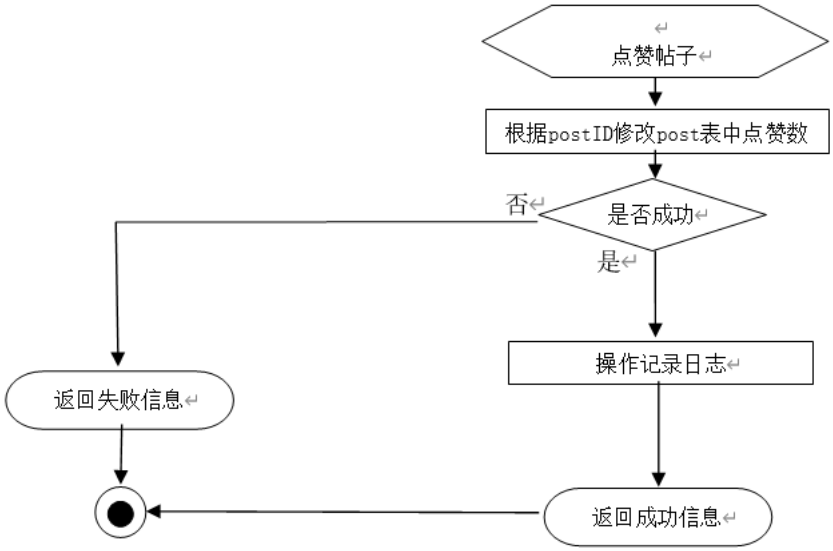
4、算法

1）前端js验证必填项

2）标签信息存入tb_tag表。

5.3.1.2.6 点赞帖子

1、流程图



2、输入

like　boolean 必填

postId Integer 必填

3、输出

成功，UI提示点赞成功;

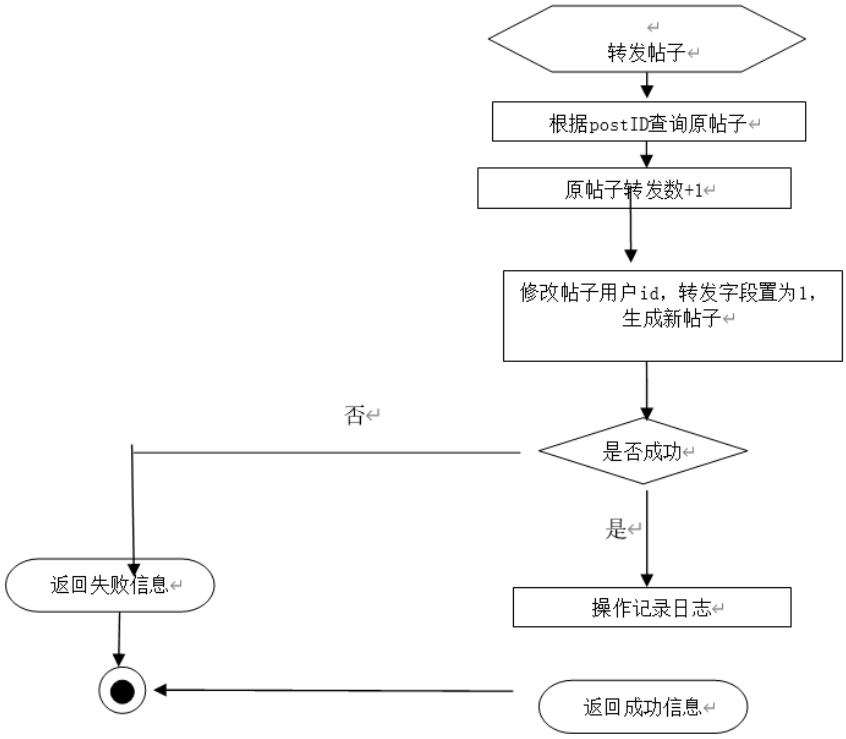
失败，UI提示具体信息

4、算法

1)根据postId将对应帖子的点赞数+1

5.3.1.2.7 转发帖子

1、流程图



2、输入

forward boolean 必填

postId

3、输出

成功，UI提示转发成功;

失败，UI提示具体信息

4、算法

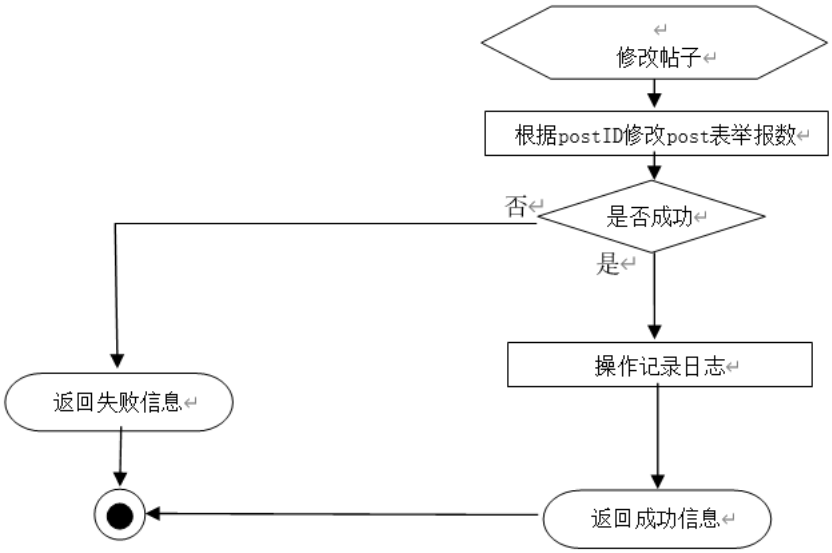
1）根据postId取得原帖子

2）原帖子字段的转发数+1，数据库原帖子进行相应修改

3）修改原帖子的用户id，将是否转发字段置为1,其他字段与原帖子一样，后将该帖子存入数据库。

5.3.1.2.8 举报帖子

1、流程图



2、输入

tipoff boolean 必填

postId 页面传参

3、输出

成功，UI提示举报成功;

失败，UI提示具体信息

4、算法

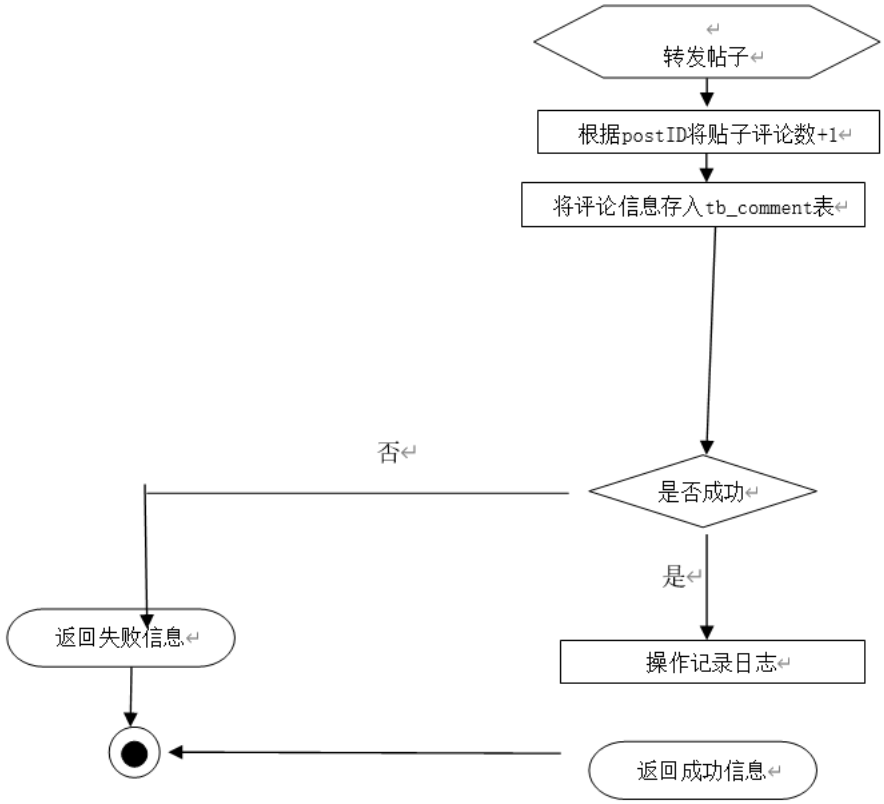
1）根据前端传来的postId取得原帖子

2）原帖子字段的举报数+1

3）数据库原帖子进行相应修改

5.3.1.2.9 评论帖子

1、流程图



2、输入

评论内容:String，必填

是否匿名:Integer，必填

评论者id：Integer，必填

comment：true

postId

3、输出

成功，UI提示评论成功;

失败，UI提示具体信息

4、算法

1）前端js验证必填项

2) 依据postId查找对应帖子

3）评论信息存入tb_comment表。

4）根据postId对帖子的评论数+1

5）对post表做相应修改

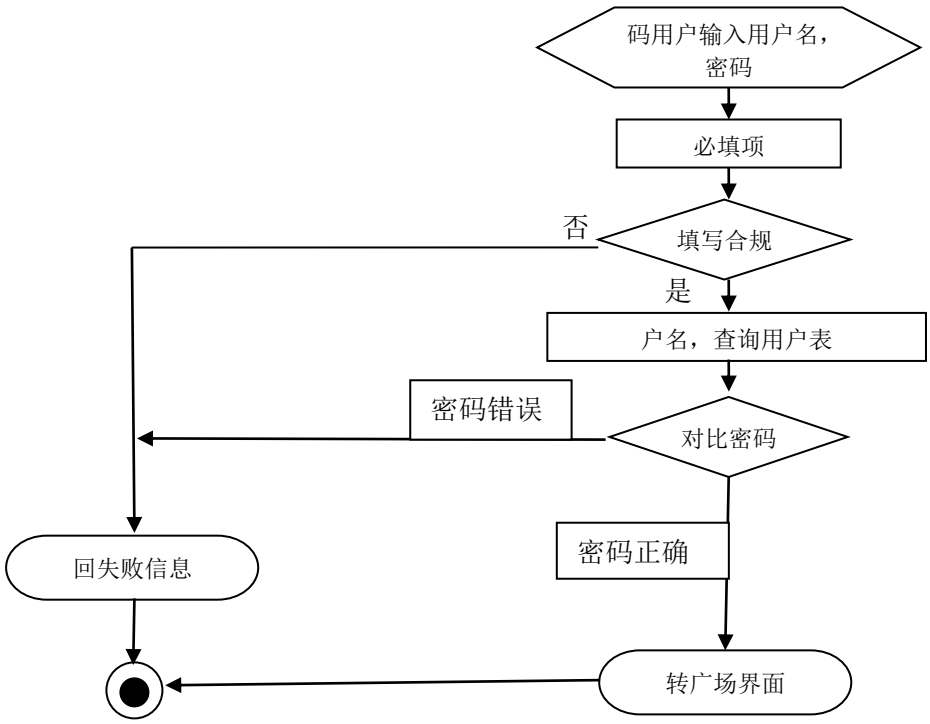
5.3.1.3 用户模块

模块描述：主要针对用户信息、用户个人空间的帖子和动态。

主要功能：登录、注册、显示帖子和动态、搜索帖子和动态、修改个人信息。

5.3.1.3.1 登录

1. 流程图



2. 输入项

用户名、密码。

3. 输出项

1）成功，UI提示登录成功，跳转至帖子广场界面。

2）失败，UI提示密码错误，返回失败信息。

4. 算法描述

1）前端js验证必填项，界面获取用户输入用户名及密码。

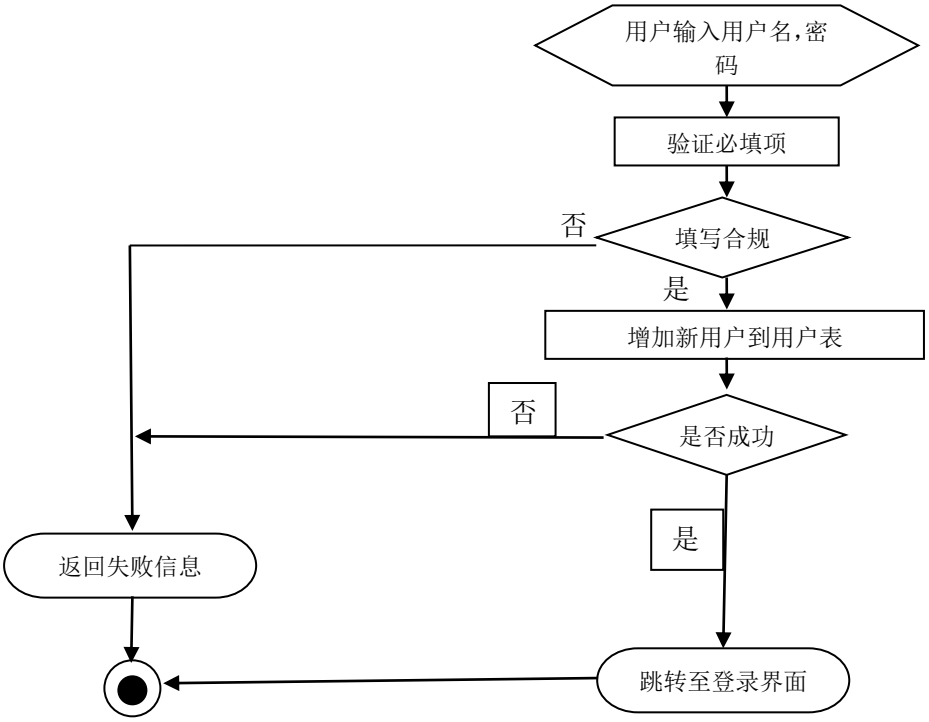
2）通过用户名查找用户表(tb_user)，获取用户密码。

3）密码加密与解密，将用户密码与输入密码做对比。

4）根据密码正确与否，执行流程图相关操作。

5.3.1.3.2 注册

1. 流程图



2. 输入项

用户名、密码。

3. 输出项

1）成功注册，跳转至登录界面。

2）失败，提示用户名已被注册，返回失败信息。

4. 算法描述

1）前端js验证必填项。

2）界面获取用户输入用户名及密码，用户名需唯一。

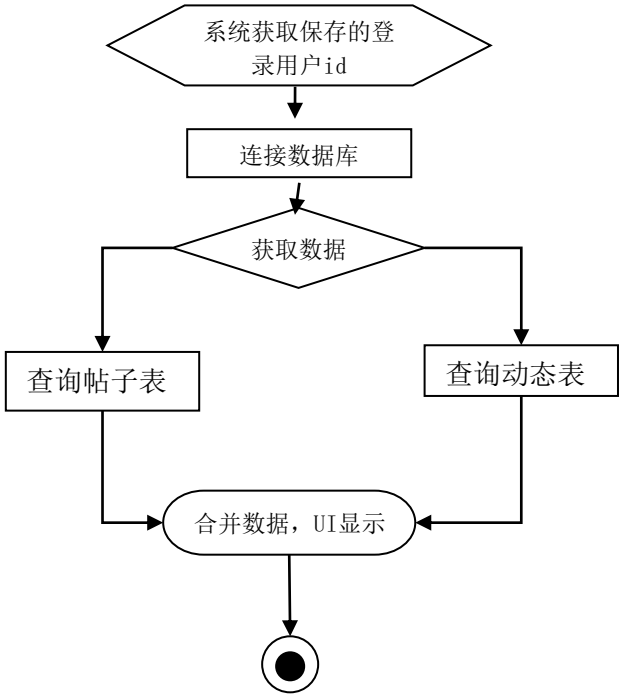
2）通过用户名查找用户表。

3）如果该用户存在，返回失败消息。

4）否则，根据用户名和密码，增加新用户到用户表（tb_user），成功注册。

5.3.1.3.3 显示帖子和动态

1. 流程图



2. 输入项

系统存储的登录用户id

3. 输出项

帖子信息及动态信息

4. 算法描述

1）获取登录用户id。

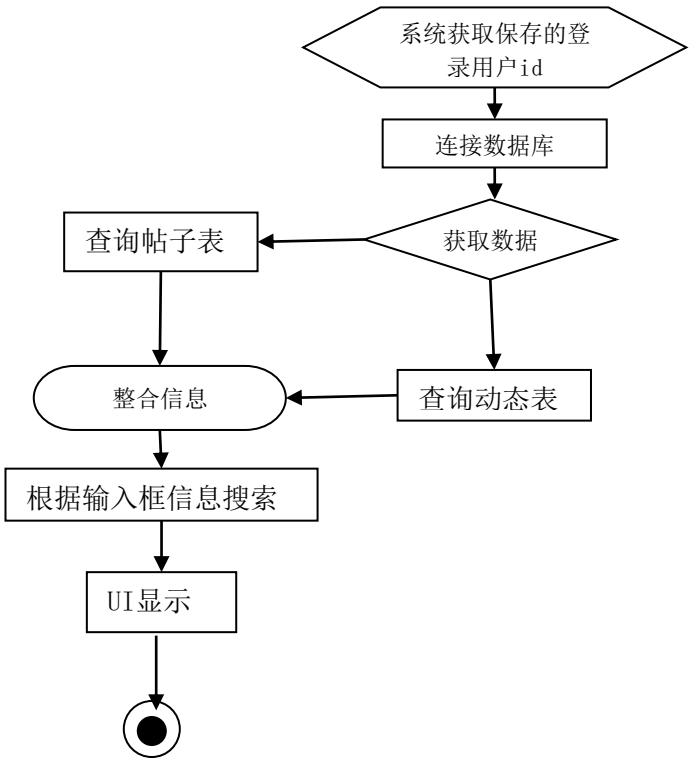
2）根据用户id查找帖子信息（tb_post）及动态信息（tb_record）。

3）数据处理。

4）UI显示。

5.3.1.3.4 搜索帖子和动态

1. 流程图



2. 输入项

系统存储的登录用户id、搜索信息。

3. 输出项

1）成功，显示搜索结果。

2）失败，提示：没有相关信息。

4. 算法描述

1）前端js验证必填项，获取登录用户id。

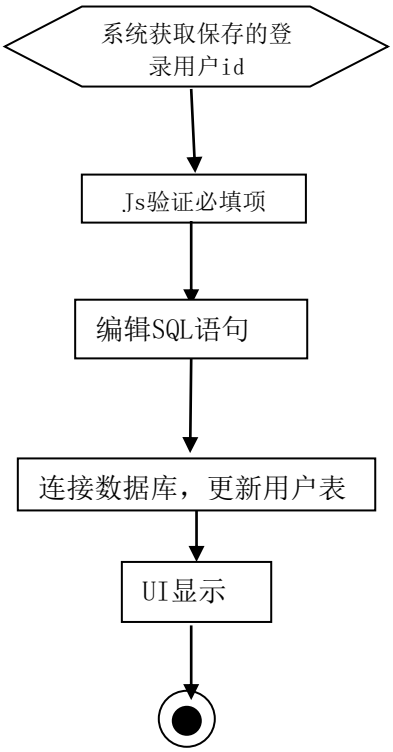
2）根据用户id查找帖子信息（tb_post）及动态信息（tb_record）。

3）根据输入框信息查找相关帖子及动态。

4）显示搜索结果。

5.3.1.3.5 修改个人信息

1. 流程图



2. 输入项

系统存储的登录用户id、用户输入的修改信息。

3. 输出项

显示更新结果，新的用户空间界面。

4. 算法描述

1）前端js验证必填项，获取登录用户id。

2）根据用户id和修改信息编辑更新语句。

3）更新用户表（tb_user）。

4）UI显示。

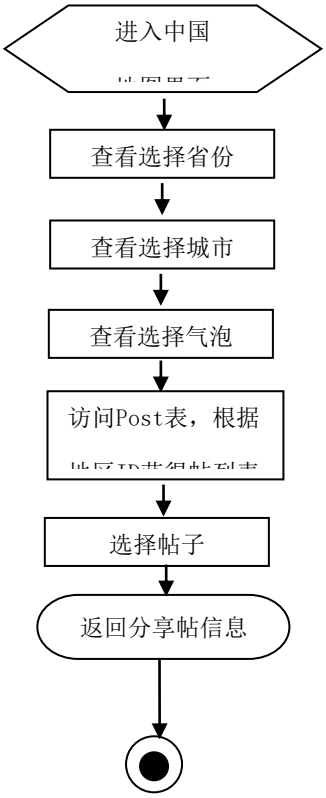
5.3.1.4 地图模块

模块描述：通过地图可视化信息，选择性查看具体地区分享贴信息

主要功能：查看具体地区气泡图、查找相关分享贴的所处地区

5.3.1.4.1 查看具体地区气泡图以及气泡内帖子信息

1、 流程图



2、 输入项

加载时为中国地图界面，页面传参数

3、 输出项

1） 地图上各个省、市区块根据气泡数渲染颜色

2） 气泡根据地区内帖子数渲染颜色深浅

3） 气泡内分享贴列表

4、 算法描述

1） 后端通过地区的气泡数计算获得某省、市区块的气泡数。

2) 根据各个区块的气泡数，排序后根据颜色排列渲染区块颜色。

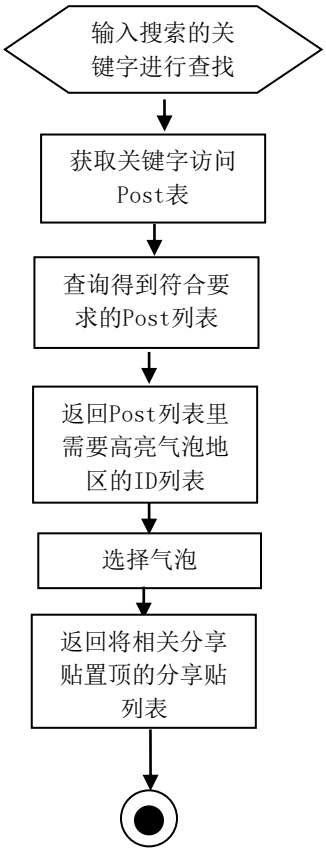
3) 后端通过参数地区ID列表访问Post表，获得符合要求的Post信息数量。

4) 通过各地区的帖子数，排序后根据颜色排列渲染气泡深浅。

5) 后端通过参数地区ID访问Post表，获得Post信息列表，再通过前端一一显示。

5.3.1.4.2 查找分享贴的所处区域

1、 流程图



2、 输入项

搜索的关键字

3、 输出项

1) 成功，符合要求的Post列表，相应气泡高亮，点击气泡，返回相关帖置顶的列表；

2) 失败，UI提示该城市无相关分享贴。

4、 算法描述

1) 根据关键字并访问Post表，查找相关Post信息

2) 获得Post列表后，提取地区ID列表

3) 将选择气泡的地区内帖子显示列表，改为将相关帖置顶的列表

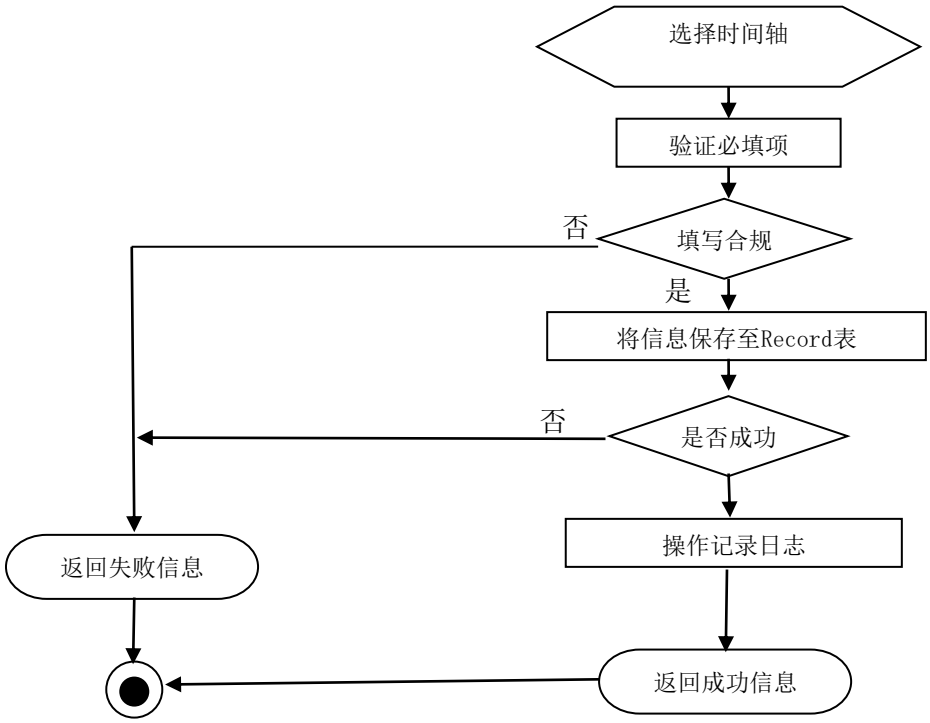
5.3.1.5 时间轴模块

模块描述：管理时间轴，包括添加动态、修改动态、删除动态。

主要功能：添加动态、修改动态、删除动态。

5.3.1.5.1 添加动态

1、 流程图



2、 输入项

动态内容	String	必填
动态图片	String	

页面传参数

3、 输出项

1）成功，UI提示添加动态完成；

2）失败，UI提示具体信息。

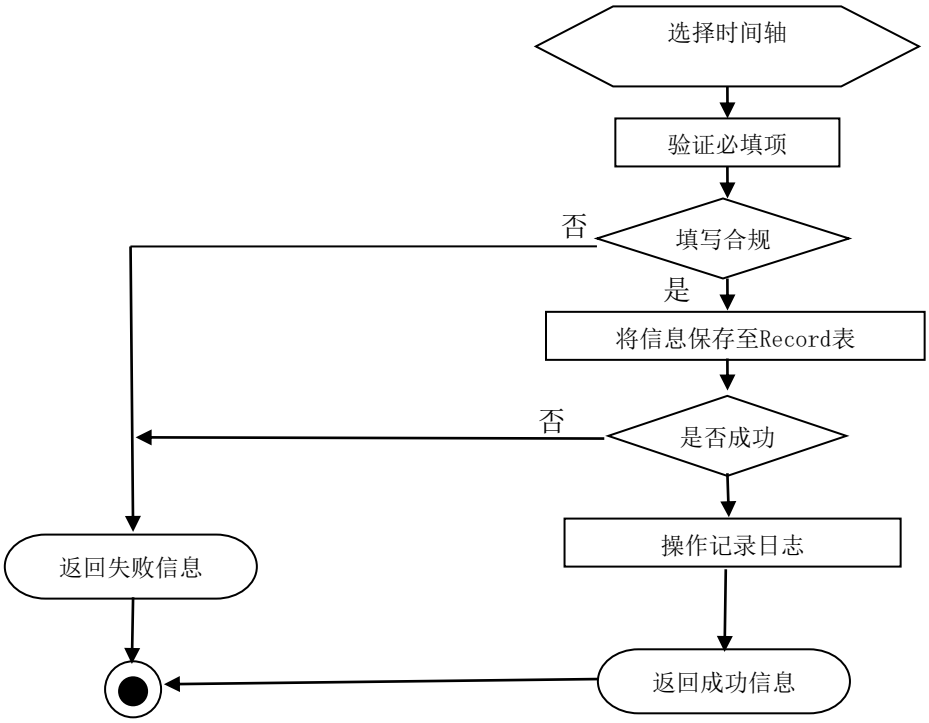
4、 算法描述

1）前端js验证必填项，界面获取用户ID等相关信息；

2）动态信息保存至Record表；

5.3.1.5.2 修改动态

1、流程图



2、输入项

动态内容　　String　　必填

动态图片　　String

页面传参数

3、输出项

2）成功，UI提示修改动态完成；

2）失败，UI提示具体信息。

4、算法描述

1）前端js验证必填项，界面获取用户ID和动态ID；

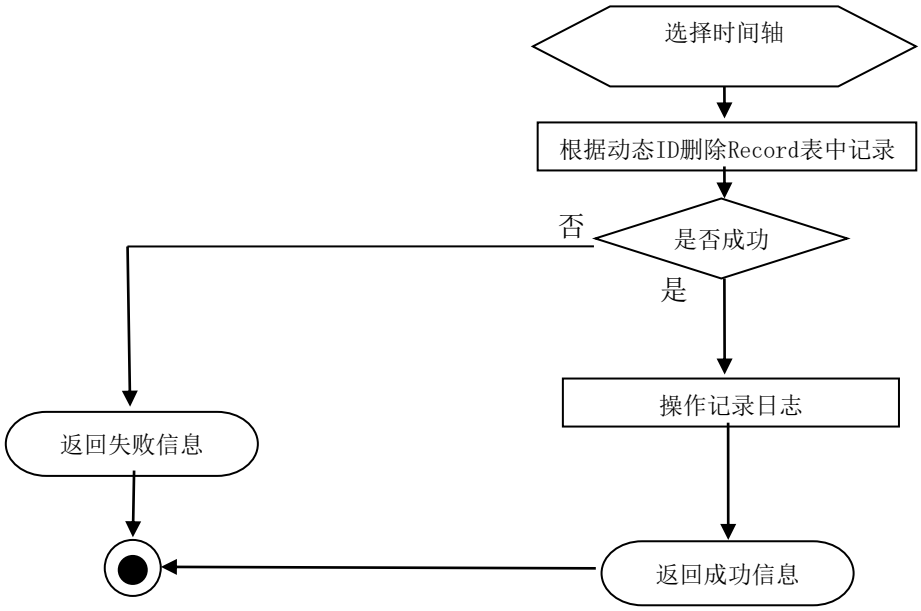
2）根据用户ID和动态ID查Record表；

3）获得相关记录后修改动态信息；

4）动态信息保存至Record表。

5.3.1.5.3 删除动态

1、流程图



2、输入项

动态内容　　String　　必填

动态图片　　String

页面传参数

3、输出项

3）成功，UI提示删除动态完成，刷新时间轴；

2）失败，UI提示具体信息。

4、算法描述

1）界面获取动态ID

2）根据动态ID查Record表，获得相关记录。

3）将该记录从Record表删除

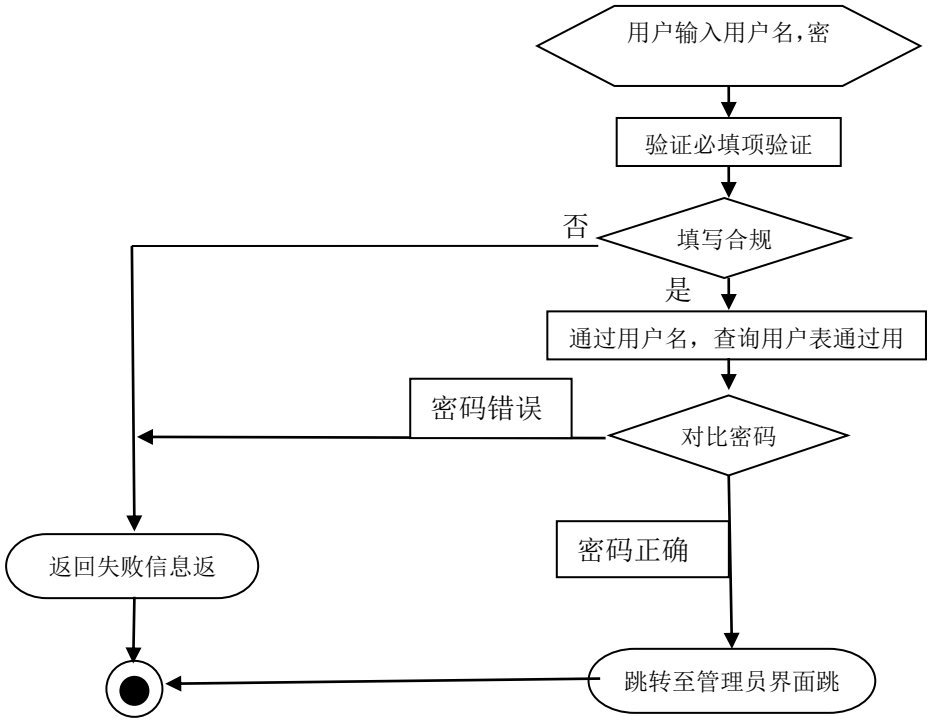
5.3.2 后台子系统

5.3.2.1 登录模块

模块描述：主要针对管理员的登录。

主要功能：登录、登录验证。

1. 流程图



2. 输入项

用户名、密码。

3. 输出项

1）成功，UI提示登录成功，跳转至管理员主页。

2）失败，UI提示密码错误，返回失败信息。

4. 算法描述

1）前端js验证必填项，界面获取用户输入用户名及密码。

2）通过用户名查找用户表(tb_user)，获取用户密码。

3）密码加密与解密，将用户密码与输入密码做对比。

4）根据密码正确与否，执行流程图相关操作。

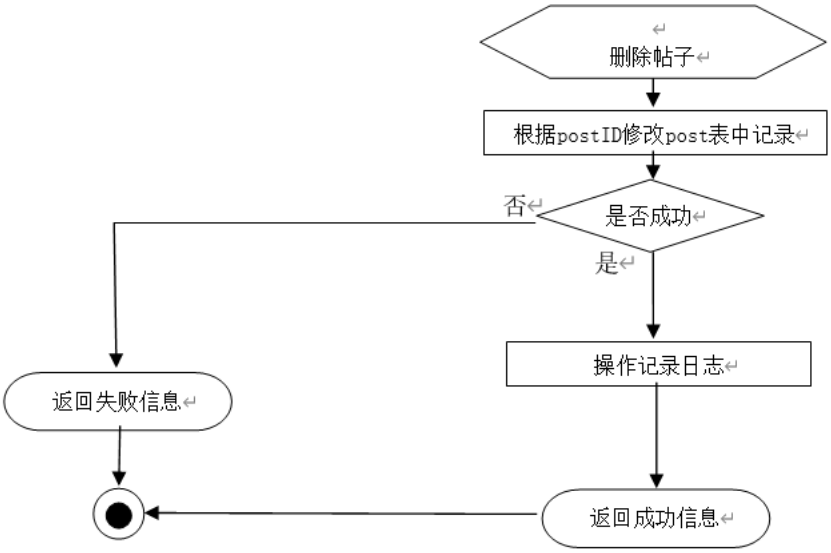
5.3.2.2 审核模块

模块描述：主要针对管理员审核帖子以及后续操作。

主要功能：审核帖子、删除帖子、冻结用户。

5.3.2.2.1 删除帖子

1、流程图



2、输入

postId

3、输出

成功，UI提示删除成功;

失败，UI提示具体信息

4、算法

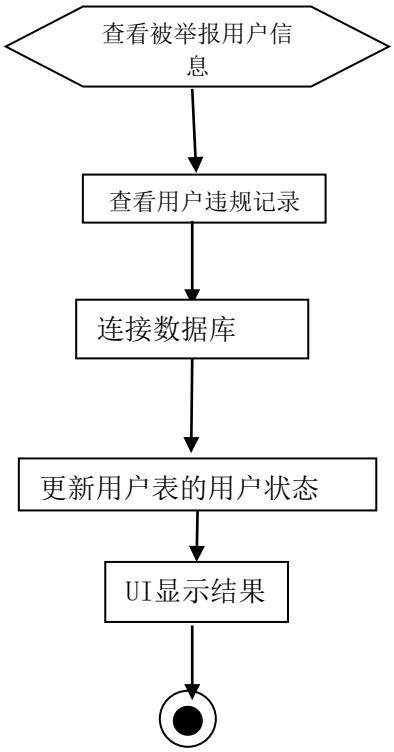
1) 界面获取动态postId

2) 根据postId查找post表，获得相关记录

3) 将该记录从post表删除

5.3.2.2.2 冻结用户

1. 流程图



2. 输入项

被举报人id

3. 输出项

用户属性标记为冻结

4. 算法描述

1）根据用户id查询用户帖子，违规记录等信息。

2）将用户状态（state）标记为已冻结，更新数据库用户表（tb_user）。

3）发送冻结信息。

5.4 系统界面详细设计

5.4.1 外部界面设计

5.4.1.1 广场模块

1）获取广场帖子列表

接口地址：./daily/publist

返回格式：json

请求方式：get

请求示例：./daily/publist?listType=new

请求参数:

名称	类型	说明
listType	String	帖子列表类型

返回参数:

名称	类型	说明
postList	ListPost	帖子列表

2）搜索帖子

接口地址：./daily/search

返回格式：json

请求方式：get

请求示例：./daily/search?q=”最新疫情信息”

请求参数:

名称	类型	说明
q	String	搜索关键字

返回参数:

名称	类型	说明
postList	ListPost	帖子列表

3）获取未读动态列表

接口地址：./daily/unread

返回格式：json

请求方式：get

请求示例：./daily/publist

请求参数:无

返回参数:

名称	类型	说明
postList	ListPost	未读帖子列表

5.4.1.2 帖子模块

1）查询所有帖子信息

接口地址：./daily/listpost

返回格式：json

请求方式：get

请求示例：./daily/listpost

请求参数：无

返回参数:

名称	类型	说明
postList	ListPost	帖子列表

2）查询某一个帖子信息

接口地址：./daily/getpostbypostid

返回格式：json

请求方式：GET

请求示例：./daily/getpostbypostid?postId=4

请求参数

名称	必填	类型	说明
postId	是	Integer	

返回参数

名称	类型	说明

名称	类型	说明
post	Post	帖子

3）查询某一用户的所有帖子

接口地址：./daily/getpostbyuserid

返回格式：json

请求方式：GET

请求示例：./daily/ getpostbyuserid?userId=1

请求参数

名称	必填	类型	说明
userId	是	string	

返回参数

名称	类型	说明
postList	List<Post>	帖子列表

4）增加帖子

接口地址：./daily/addpost

返回格式：json

请求方式：POST

请求示例：./daily/addpost

请求参数

名称	必填	类型	说明
post	是	Post	

返回参数

名称	类型	说明
success	boolean	添加帖子是否成功

5）修改帖子

接口地址：./daily/modifypost

返回格式：json

请求方式：POST

请求示例：./daily/modifypost

请求参数

名称	必填	类型	说明
post	是	Post	

返回参数

名称	类型	说明
success	boolean	修改帖子是否成功

6）删除帖子

接口地址：./daily/deletepost

返回格式：json

请求方式：POST

请求示例：./daily/deletepost

请求参数：无

返回参数

名称	类型	说明
success	boolean	删除帖子是否成功

7）点赞帖子

接口地址：./daily/likePost

返回格式：json

请求方式：get

请求示例：./daily/likePost?like=true

./daily/likePost?like=false

请求参数

名称	必填	类型	说明
like	是	boolean	True:点赞 False：取消点赞

返回参数

名称	类型	说明
success	boolean	是否成功

8）评论帖子

接口地址：./daily/commentpost

返回格式：json

请求方式：POST

请求示例：./daily/commentpost

请求参数

名称	必填	类型	说明
comment	是	Comment	

返回参数

名称	类型	说明
success	boolean	是否成功

9）转发

接口地址：./daily/forwardpost

返回格式：json

请求方式：get

请求示例：./daily/forwardpost?postId=1

请求参数

名称	必填	类型	说明
postId	是	Integer	帖子id

返回参数

名称	类型	说明
post	Post	新帖子

10）举报

接口地址：./daily/tipoffpost

返回格式：json

请求方式：get

请求示例：./daily/tipoffpost?tipoff=true

./daily/tipoffpost?tipoff=false

请求参数

名称	必填	类型	说明
tipoff	是	boolean	True:举报 False：取消举报

返回参数

名称	类型	说明
success	boolean	是否成功

11）添加标签

接口地址：./daily/addtag

返回格式：json

请求方式：POST

请求示例：./daily/addtag

请求参数

名称	必填	类型	说明
tag	是	Tag	

返回参数

名称	类型	说明
success	boolean	是否成功

5.4.1.3 用户模块及管理员模块

1）获取用户所有帖子

接口地址：./daily/getpostbyuserid

返回格式：json

请求方式：GET

请求示例：./daily/getpostbyuserid?userId=1

请求参数

名称	必填	类型	说明
userId	是	Interger	用户ID

返回参数

名称	类型	说明
postList	List<Post>	帖子列表

2）获取用户所有动态

接口地址：./daily/getRecordListByTATAndUser

返回格式：json

请求方式：GET

请求示例：./daily/getRecordListByUser?userId=1

请求参数

名称	必填	类型	说明
userId	是	Interger	用户ID

返回参数

名称	类型	说明
recordList	List<Record>	动态列表

3）登录验证

接口地址：./daily/signInbyUserNameAndPwd

返回格式：json

请求方式：POST

请求示例：./daily/signInbyUserNameAndPwd

请求参数

名称	必填	类型	说明
name	是	String	用户名
password	是	String	密码

返回参数

名称	类型	说明
Success	boolean	登录结果

4)注册账户

接口地址：./daily/signUpbyUserNameAndPwd

返回格式：json

请求方式：POST

请求示例：./daily/signUpbyUserNameAndPwd

请求参数

名称	必填	类型	说明
name	是	String	用户名
password	是	String	密码

返回参数

名称	类型	说明
Success	boolean	注册结果

5)按条件搜索用户空间

接口地址：./daily/searchUserPlaceByCriticalWord

返回格式：json

请求方式：GET

请求示例：./daily/searchUserPlaceByCriticalWord?CriticalWord=锅

请求参数

名称	必填	类型	说明
criticalWord	是	String	关键词

返回参数

名称	类型	说明
Success	boolean	注册结果
recordList	List<Record>	动态列表
postList	List<Post>	帖子列表

6）更新个人信息

接口地址：./daily/updateUserInfo

返回格式：json

请求方式：POST

请求示例：./daily/updateUserInfo

请求参数

名称	必填	类型	说明
name	否	String	用户名
pwd	否	String	用户密码
gender	否	String	用户性别

返回参数

名称	类型	说明
Success	boolean	更新结果

7）审核帖子

接口地址：./daily/getpostbypostid

返回格式：json

请求方式：GET

请求示例：./daily/getpostbypostid?postId=4

请求参数

名称	必填	类型	说明
postId	是	Integer	

返回参数

名称	类型	说明
post	Post	帖子

8）删除帖子

接口地址：./daily/deletepostbypostid

返回格式：json

请求方式：GET

请求示例：./daily/deletepost?postId=4

请求参数

名称	必填	类型	说明
postId	是	Integer	

返回参数

名称	类型	说明
success	boolean	删除帖子是否成功

9）冻结用户

接口地址：./daily/freezeuserbyuserid

返回格式：json

请求方式：GET

请求示例：./daily/freezeuserbyuserid?userid=10

请求参数

名称	必填	类型	说明
userId	是	Integer	

返回参数

名称	类型	说明
success	boolean	冻结用户是否成功

5.4.1.4 地图模块

1）获得地区气泡数

接口地址：./daily/getBubbleNumByAreaID

返回格式：json

请求方式：GET

请求示例：./daily/getBubbleNumByAreaID?areaID=123

请求参数

名称	必填	类型	说明
areaId	是	Integer	地区ID

返回参数

名称	类型	说明
bubbleNum	Integer	气泡数

2）获得地区帖子数

接口地址：./daily/getPostNumByAreaID

返回格式：json

请求方式：GET

请求示例：./daily/getPostNumByAreaID?areaID=123

请求参数

名称	必填	类型	说明
areaId	是	Integer	地区ID

返回参数

名称	类型	说明
postNum	Integer	帖子数

3）获得地区帖子列表

接口地址：./daily/getPostListByAreaID

返回格式：json

请求方式：GET

请求示例：./daily/getPostListByAreaID?areaID=123

请求参数

名称	必填	类型	说明
areaId	是	Integer	地区ID

返回参数

名称	类型	说明
postList	List<Post>	帖子列表

4）获得需要高亮的气泡地区ID列表

接口地址：./daily/getAreaIdListByTag

返回格式：json

请求方式：GET

请求示例：./daily/getAreaIdListByTag?tag=火锅

请求参数

名称	必填	类型	说明
tag	是	String	搜索的关键字

返回参数

名称	类型	说明
areaIdList	List<Interger>	气泡地区ID列表

5）获得需要高亮的气泡地区内将相关帖子置顶的帖子列表

接口地址：./daily/getNewPostListByTag

返回格式：json

请求方式：GET

请求示例：./daily/getNewPostListByTag?tag=火锅

请求参数

名称	必填	类型	说明
tag	是	String	搜索的关键字

返回参数

名称	类型	说明
newPostList	List<Post>	新排序帖子列表

5.4.1.5 时间轴模块

1）添加动态

接口地址：./daily/createRecord

返回格式：json

请求方式：GET

请求示例：./daily/createRecord?timeAT=12&userId=21&recordCt=123&recdImg=123

请求参数

名称	必填	类型	说明
timeAxisType	是	String	时间轴类型
userId	是	Interger	用户ID
recordContent	是	String	动态文字
recordImg	是	String	动态图片

返回参数

名称	类型	说明
success	boolean	判断增加动态是否成功

2）修改动态

接口地址：./daily/updateRecord

返回格式：json

请求方式：GET

请求示例：./daily/updateRecord?recdId=21&recdCtt=123&recdImg=123

请求参数

名称	必填	类型	说明
recordId	是	Interger	动态ID
recordContent	是	String	动态文字
recordImg	是	String	动态图片

返回参数

名称	类型	说明
success	boolean	判断修改动态是否成功

4）删除动态

接口地址：./daily/deleteRecord

返回格式：json

请求方式：GET

请求示例：./daily/deleteRecord?recordId=21

请求参数

名称	必填	类型	说明
recordId	是	Interger	动态ID

返回参数

名称	类型	说明
success	boolean	判断删除动态是否成功

5）获得动态列表

接口地址：./daily/getRecordListByTATAndUser

返回格式：json

请求方式：GET

请求示例：./daily/getRecordListByTATAndUser?timeAxisType=12&userId=21

请求参数

名称	必填	类型	说明
timeAxisType	是	String	时间轴类型
userId	是	Interger	用户ID

返回参数

名称	类型	说明
recordList	List<Record>	动态列表

5.4.2 内部界面设计

1、地图和动态，通过地区ID查找到相应的动态

接口地址：./daily/getRecordByAreaId

返回格式：json

请求方式：GET

请求示例：./daily/getRecordByAreaId?areaId=123

请求参数

名称	必填	类型	说明
areaId	是	Interger	地区ID

返回参数

名称	类型	说明
record	Record	动态

2、地图和分享帖，通过地区ID查找到相应的分享帖

接口地址：./daily/getPostByAreaId

返回格式：json

请求方式：GE

请求示例：./daily/getPostByAreaId?areaId=123

请求参数

名称	必填	类型	说明
areaId	是	Interger	地区ID

返回参数

名称	类型	说明
post	Post	帖子

3、时间轴和动态，通过时间轴ID得到相应的动态列表

接口地址：./daily/getRecordListByTimeAxisId

返回格式：json

请求方式：GET

请求示例：./daily/getRecordListByTimeAxisId?timeAxisId=123

请求参数

名称	必填	类型	说明
timeAxisId	是	Interger	时间轴ID

返回参数

名称	类型	说明
recordList	List<Record>	动态列表

4、时间轴和用户，通过用户ID和时间轴类型得到相应的时间轴ID

接口地址：./daily/getTimeAxisIdByTATAndUser

返回格式：json

请求方式：GET

请求示例：./daily/getTimeAxisIdByTATAndUser?timeAxisType=1&userId=2

请求参数

名称	必填	类型	说明
timeAxisType	是	String	时间轴类型
userId	是	Interger	用户ID

返回参数

名称	类型	说明
timeAxisId	Interger	时间轴ID

5、查找发帖人

接口地址：./daily/getuseridbypostid

返回格式：json

请求方式：GET

请求示例：./daily/ getuseridbypostid?postId=1

请求参数

名称	必填	类型	说明
postId	是	Integer	帖子id

返回参数

名称	类型	说明
userId	Integer	用户id

6、获取帖子举报数

接口地址：./daily/getforwardnumbypostid

返回格式：json

请求方式：GET

请求示例：./daily/ getforwardnumbypostid?postId=1

请求参数

名称	必填	类型	说明
postId	是	Integer	帖子id

返回参数

名称	类型	说明
tipoffNum	Integer	被举报数

7、获取个人信息

接口地址：./daily/viewUserInfo

返回格式：json

请求方式：GET

请求示例：./daily/viewUserInfo

请求参数

名称	必填	类型	说明
userId	是	Interger	用户ID

返回参数

名称	类型	说明
userInfo	User	用户类的对象

8、查询用户账号是否存在

接口地址：./daily/viewUserExistByUserName

返回格式：json

请求方式：POST

请求示例：./daily/viewUserExistByUserName

请求参数

名称	必填	类型	说明
name	是	String	待查询用户名

返回参数

名称	类型	说明
Exist	Boolean	是否存在

5.4.3 用户界面设计

- **人机界面内容与界面风格**
  - **输入输出以及交互方式**

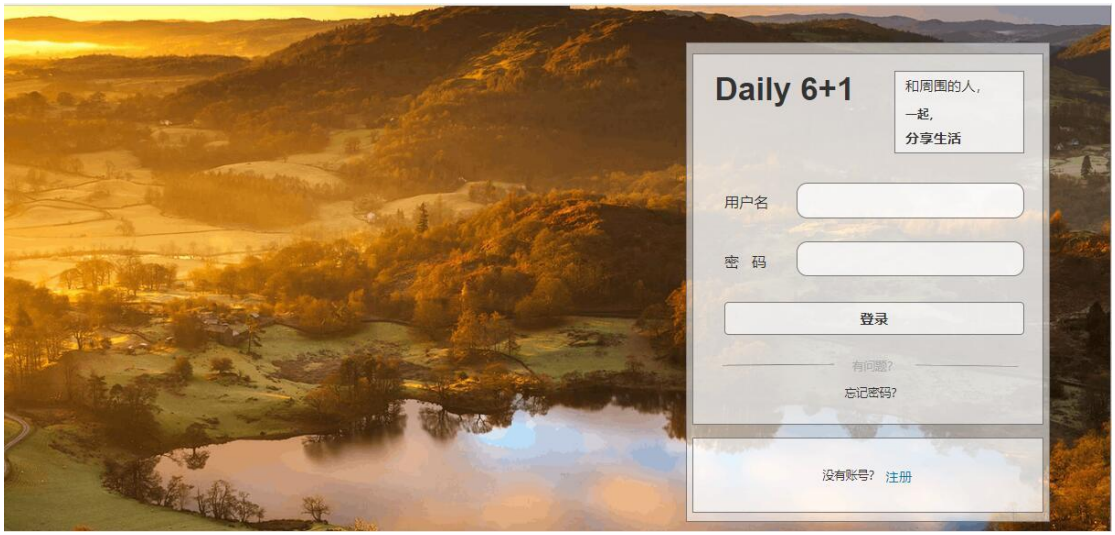
1. 输入主要以键盘、鼠标点击作为输入。
2. 输出主要在屏幕上显示作为输出。
3. 通过点击页面上的按钮以及输入功能对应的信息进行交互。

  - **界面风格**

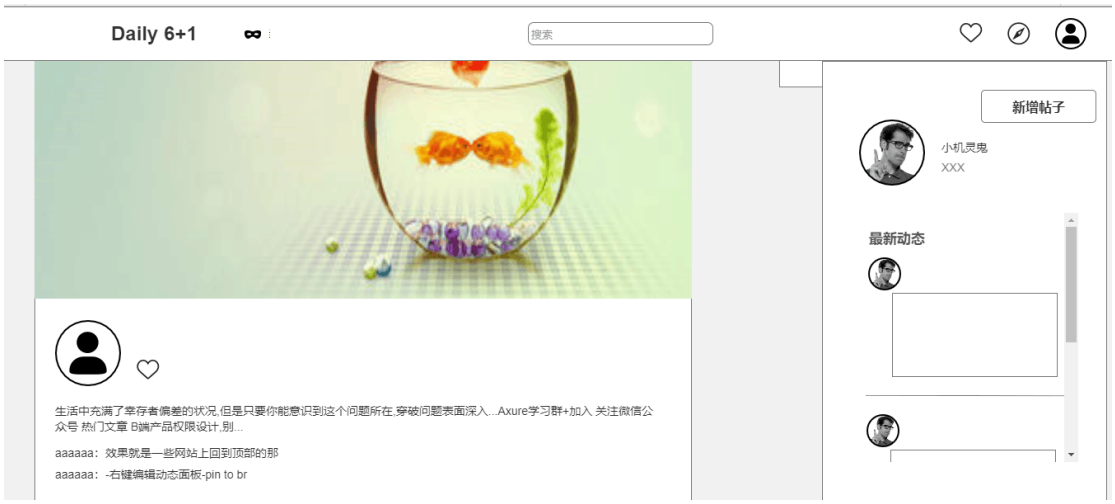
1. 图形用户界面（GUI）交互。
2. 以纯色调为主。黑白色作为主色调，保持界面一致性。
3. 保持界面简洁，无多余的装饰。界面直观、对用户透明：用户接触软件后对界面上对应的功能一目了然。
4. 对齐基础风格为居中，排版风格基本保证对称，如果无法保证对称则尽量使界面协调。

  - **界面内容**

1. WIMP作为人机交互界面的界面，W指窗口(Windows)、I指图标(Icons)、M指菜单(Menu)、P指指点设备(Point Device)
2. 保证tab键的界面跳动不可无序跳动，依照从上到下，从左到右跳动。
3. 作为webapp，不设置按钮对应的快捷键。
4. 具体界面内容见下原型界面概要。


Daily 6+1
和周围的人，
一起，
分享生活
用户名
密 码
登录
有问题?
忘记密码?
没有账号? 注册

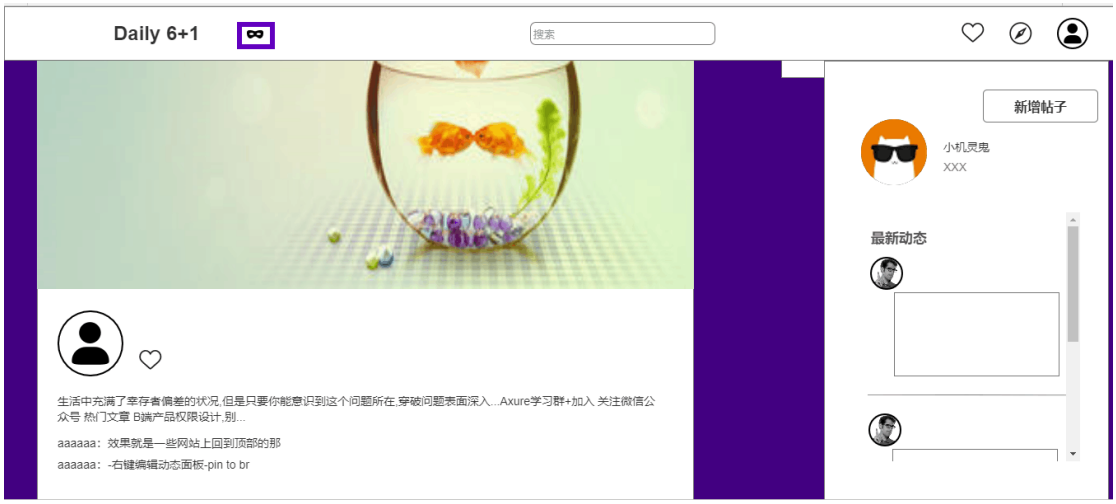
登陆界面


Daily 6+1
搜索
新增帖子
小机灵鬼
XXX
最新动态
生活中充满了幸存者偏差的状况,但是只要你能意识到这个问题所在,穿破问题表面深入...Axure学习群+加入 关注微信公众号 热门文章 B端产品权限设计,别...
aaaaaa：效果就是一些网站上回到顶部的那
aaaaaa：-右键编辑动态面板-pin to br

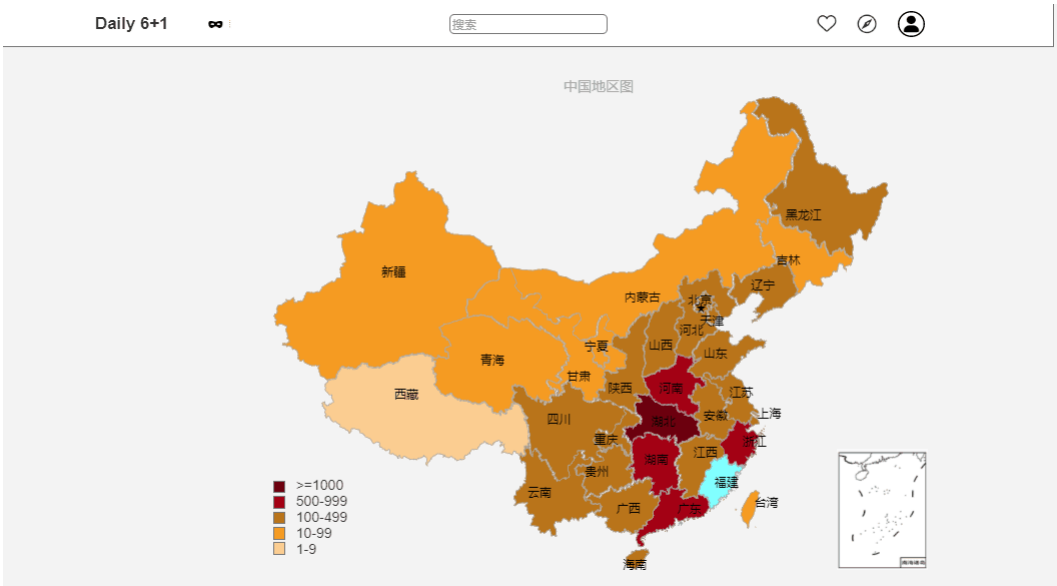
平台界面


Daily 6+1
搜索
killy
今天12：30
+关注
这对海外回来的杭州父女够奇葩的了，不想去隔离区要求回家在小区门口闹事，我一度怀疑这几天闹事的他们是不是在海外接到任务，回国拍视频发外网达到抹黑的目的
XXXXX
XXX
最新动态

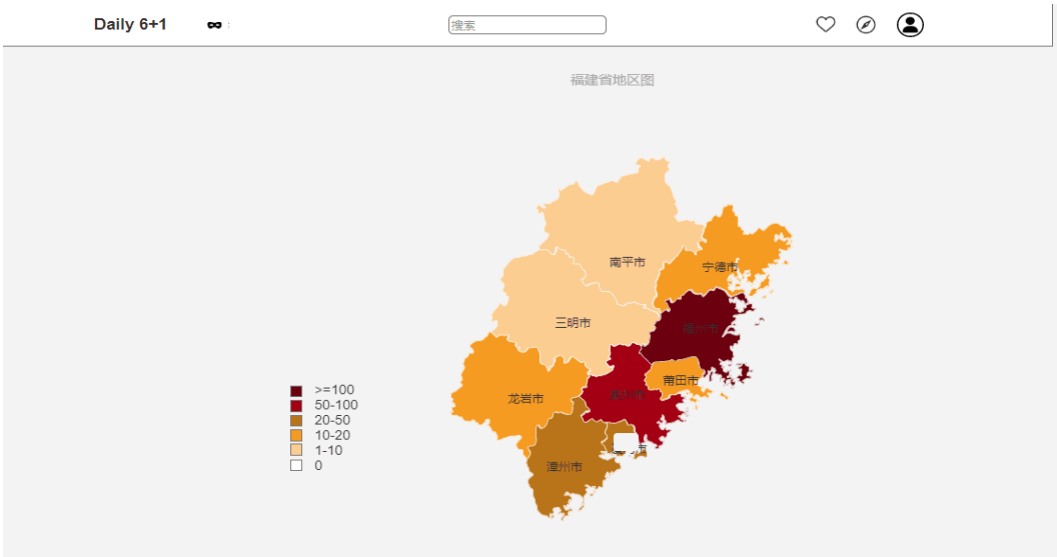
帖子详情界面



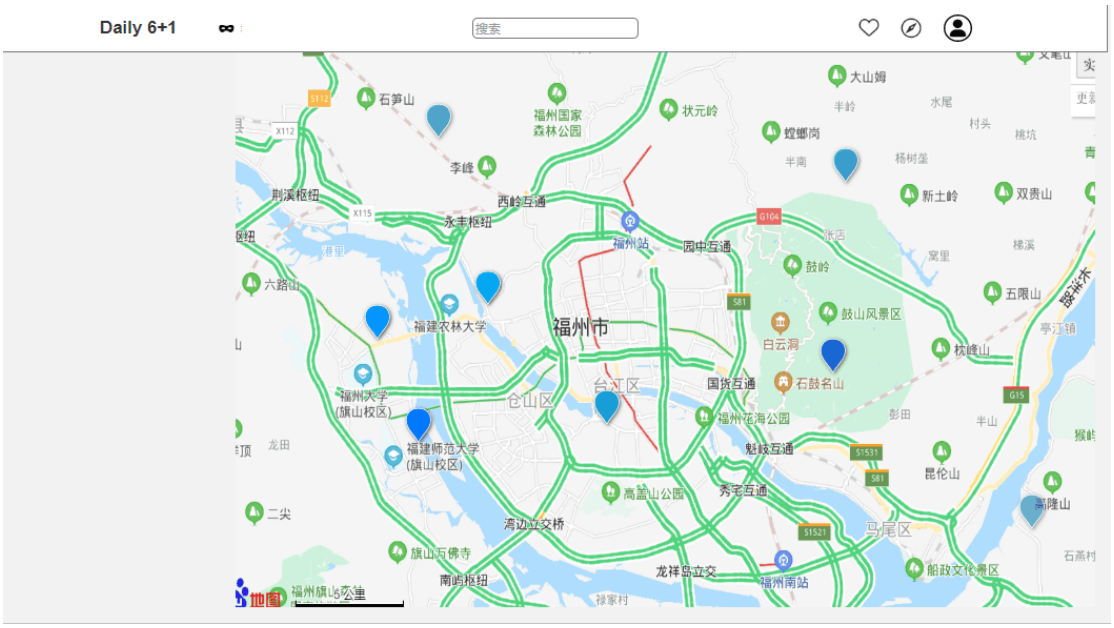
匿名主页



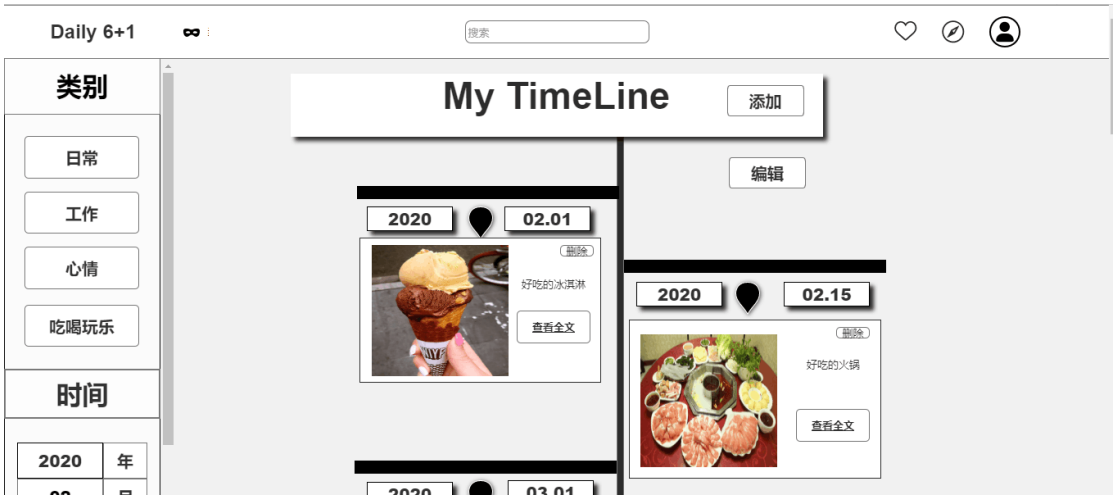
地图模块1


Daily 6+1
搜索
福建省地区图
南平市
宁德市
三明市
南田市
龙岩市
漳州市
>=100
50-100
20-50
10-20
1-10
0

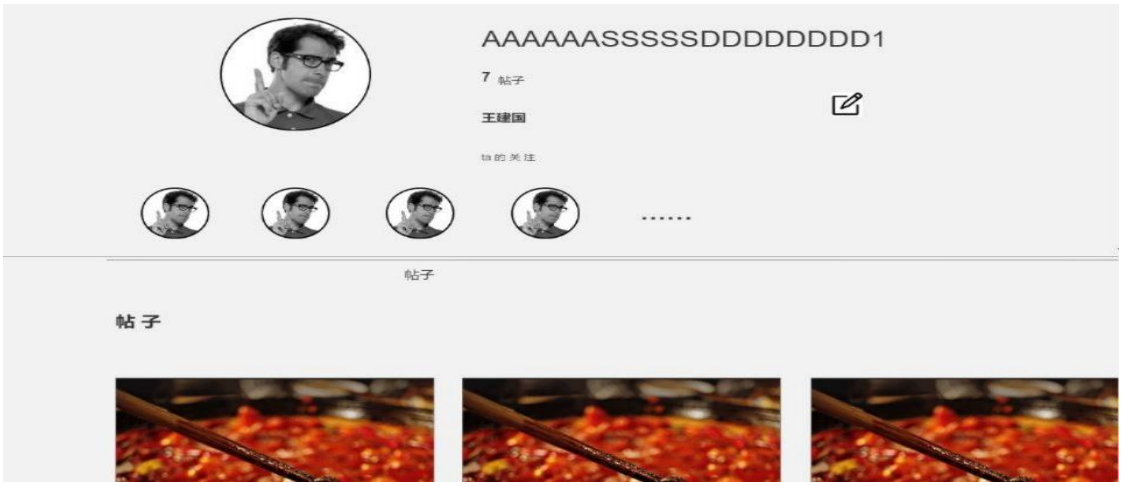
地图模块2


Daily 6+1
搜索
大山姆
石笋山
福州国家森林公园
状元岭
螳螂岗
李峰
西岭互通
荆溪枢纽
永丰枢纽
福州站
园中互通
新土岭
双贵山
鼓岭
五限山
六路山
福建农林大学
福州市
鼓山风景区
白云洞
杭峰山
福州大学(旗山校区)
仓山区
台江区
国货互通
石鼓名山
福州花海公园
福建师范大学(旗山校区)
魁岐互通
昆仑山
高盖山公园
秀宅互通
二尖
湾边立交桥
马尾区
旗山五佛寺
南跨枢纽
龙祥岛立交
福州南站
船政文化景区
福州旗山5公里

地图模块3


Daily 6+1
搜索
类别
日常
工作
心情
吃喝玩乐
时间
2020
年
My TimeLine
添加
编辑
2020
02.01
删除
好吃的冰淇淋
查看全文
2020
02.15
删除
好吃的火锅
查看全文

时间轴模块


AAAAAASSSSSDDDDDDDD1
7 帖子
王建国
.....
帖子
帖子

个人主页

- **调用方式**
    - 前后端之间通过AJAX异步调用后端提供的接口，接受返回数据。
    - 页面内部组件之间通过父子组件，Vue中的$emit方法和prop方法传递参数。
    - 页面之间的调用使用url的get方法实现简单的调用。
- **相关设计（表单设计、报表设计等）**
    - 表单传递信息时，为了防止URL信息泄露，使用POST方式传递。
    - **登录表单**由用户名、密码组成。两者都为String类型。
    - **注册表单**由用户名、密码、手机号码组成。三者都为String类型。
    - **新建帖子表单**由帖子标题、帖子内容、图片、Tag和地区ID组成。其中帖子标题、内容和Tag都以String类型，地图ID为Int类型，图片为img类型。
    - **新建动态表单**由动态标题、动态内容、图片组成，如果选择了关联帖子,则需要获得帖子ID后再传递ID。其中动态标题、动态内容、为String类型，图片为img类型，帖子ID为int类型。
- **输入设计准则**

1. 尽量减少用户输入动作的数量；
2. 维护信息显示和数据输入的一致性；
3. 在当前动作的语境中使不合适的命令不起作用；
4. 用户可以跳过不必要的动作，改变动作顺序，以及在不退出系统的情况下从错误状态中恢复；
5. 消除冗余输入，可能的话提供缺省值。
6. 执行有较大破坏性的动作前要求确认；
7. 提供语境相关的帮助机制。

- **用户所需输出**
    - **登录**

1. 登陆成功时提示登陆成功。
2. 登录失败时，提示登录的信息非法。
3. 注册成功时，提示注册成功。
4. 注册信息填写非法时，提示非法内容。如：用户名长度不合法。
5. 忘记密码填写非法时，提示非法内容，如：手机号不存在。

    - **帖子平台**

1. 新建帖子成功时，通过页面效果提示输出成功，如新建贴滑动显示。
2. 评论成功时，通过页面刷新提示评论成功。
3. 点赞、转发成功时，改变图标颜色提示输出成功。
4. 以上内容失败时，提示错误内容。如：发送评论失败。
5. 匿名/实名切换成功时，改变页面背景提示切换成功。
6. 添加TAG成功时，改变页面背景提示切换成功。
7. TAG在显示时分批显示。
8. 搜索结果应该直接输出在页面上。
9. 搜索结果为空则输出：找不到搜索内容。

- **地图界面**

1. 按照数据的多少在地图上呈现出深浅不一的颜色。
2. 地区模块中的气泡也需要按照数量多少改变颜色深浅。
3. 搜索得到成果后，含搜索内容的气泡应该高亮
4. 没有搜索成果时，气泡颜色不变。
5. 点击高亮气泡时可以跳转到相应的帖子界面。

- **时间轴界面**

1. 新增动态时，窗口提示新增动态成功。
2. 筛选时（包括类别筛选和时间筛选），应该保证时间轴表现形式不变。
3. 时间轴界面不支持搜索。
4. 删除时间轴动态时，应该刷新页面以提示删除成功。

- **个人空间**

1. 个人信息修改成功时，提示修改成功。
2. 个人信息填写非法时，提示非法内容。如：签名过长。
3. 搜索个人的帖子和动态时，应将输出结果直接展示。
4. 搜索结果为空则输出：找不到搜索内容。
5. 点击关注用户时，应该直接跳转到他的个人空间。

- **后台操作**

1. 冻结用户成功后，应该显示出现在用户的状态。如用红字提示。
2. 删除帖子成功后，应该窗口提示删除成功。