

汇报人：潘晨宇 汇报时间：04/11

DAILY

6+1

一个关注用户极简、直观体验，希冀纯享功能的私密交流WEBAPP



CONTENT



01. 类图、用例图等图展示

Focus on the charts

02. 系统设计部分

Details in the design of the programming system

03. 数据库设计部分

Details in the design of the database

04. 开发与分工

Development and division of labor



01

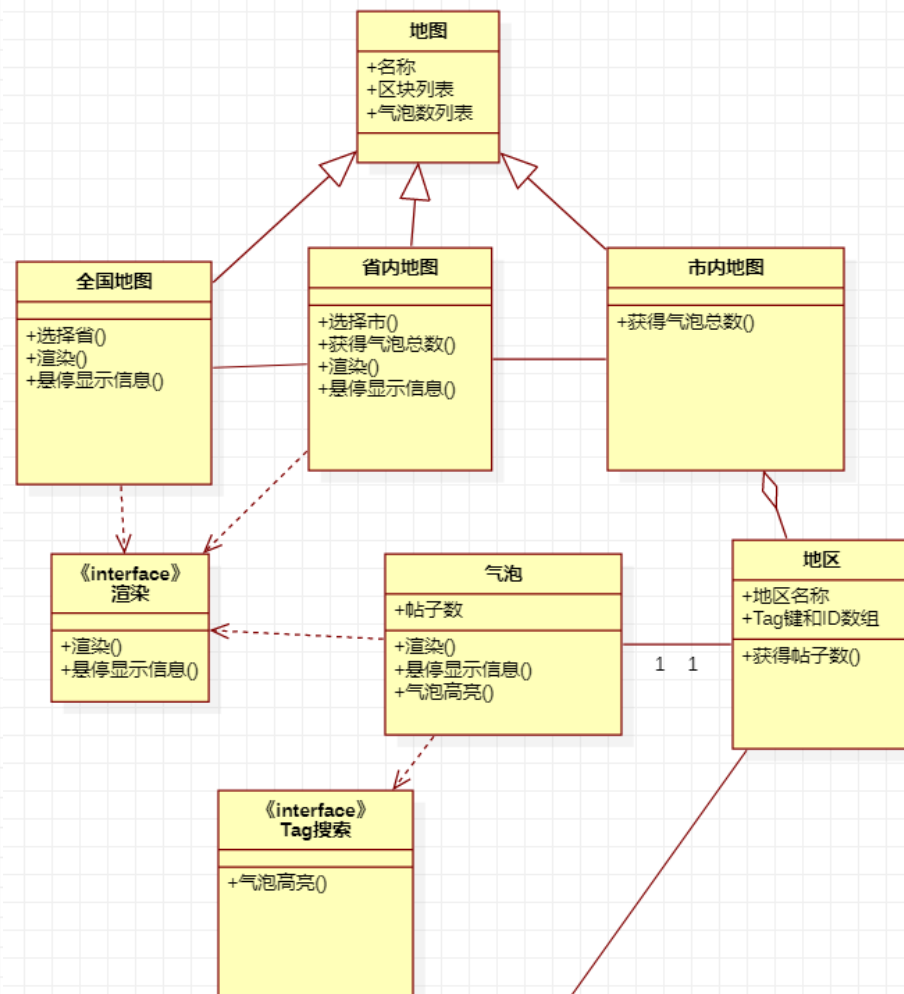
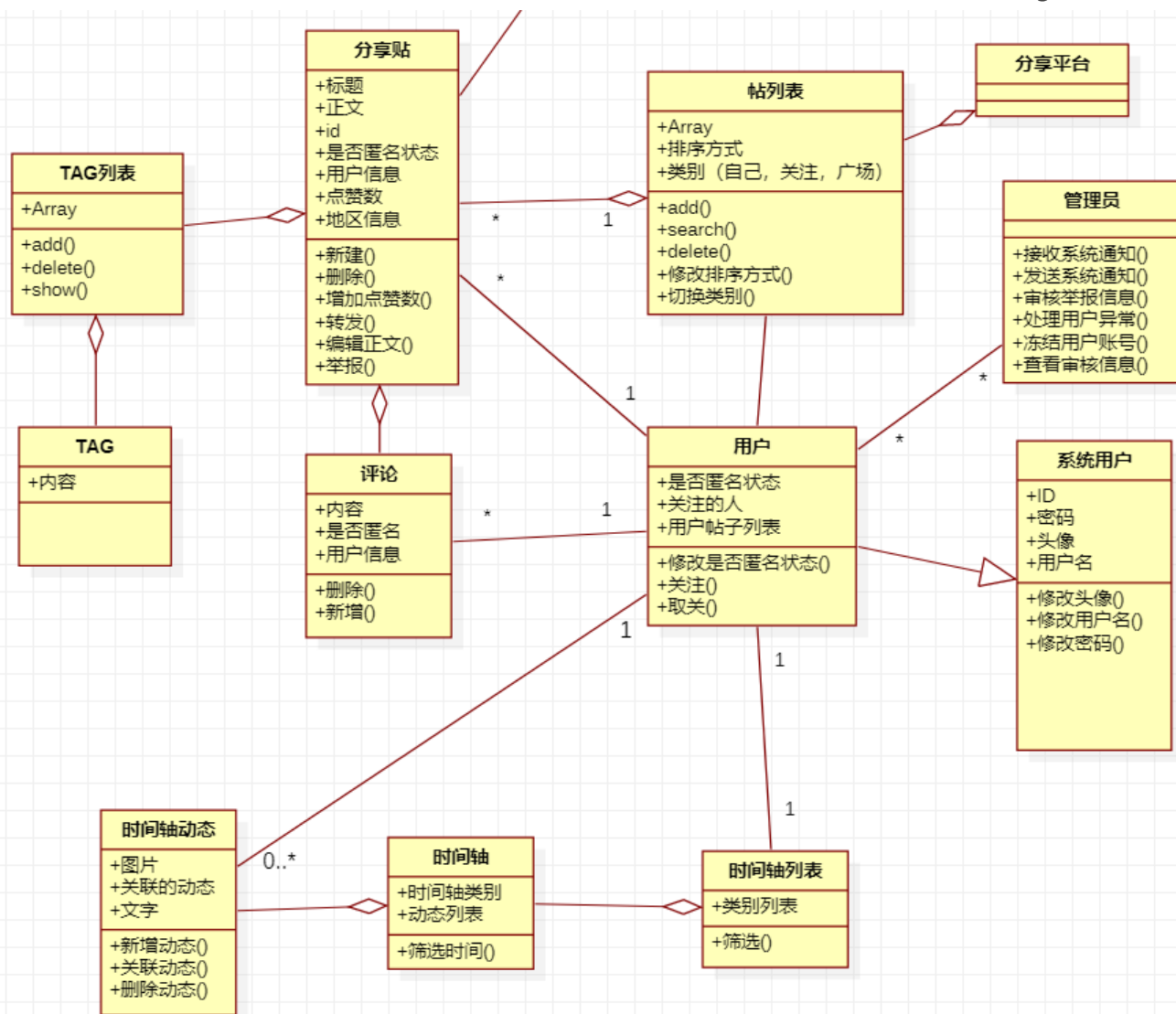
工程图展示

Focus on the charts



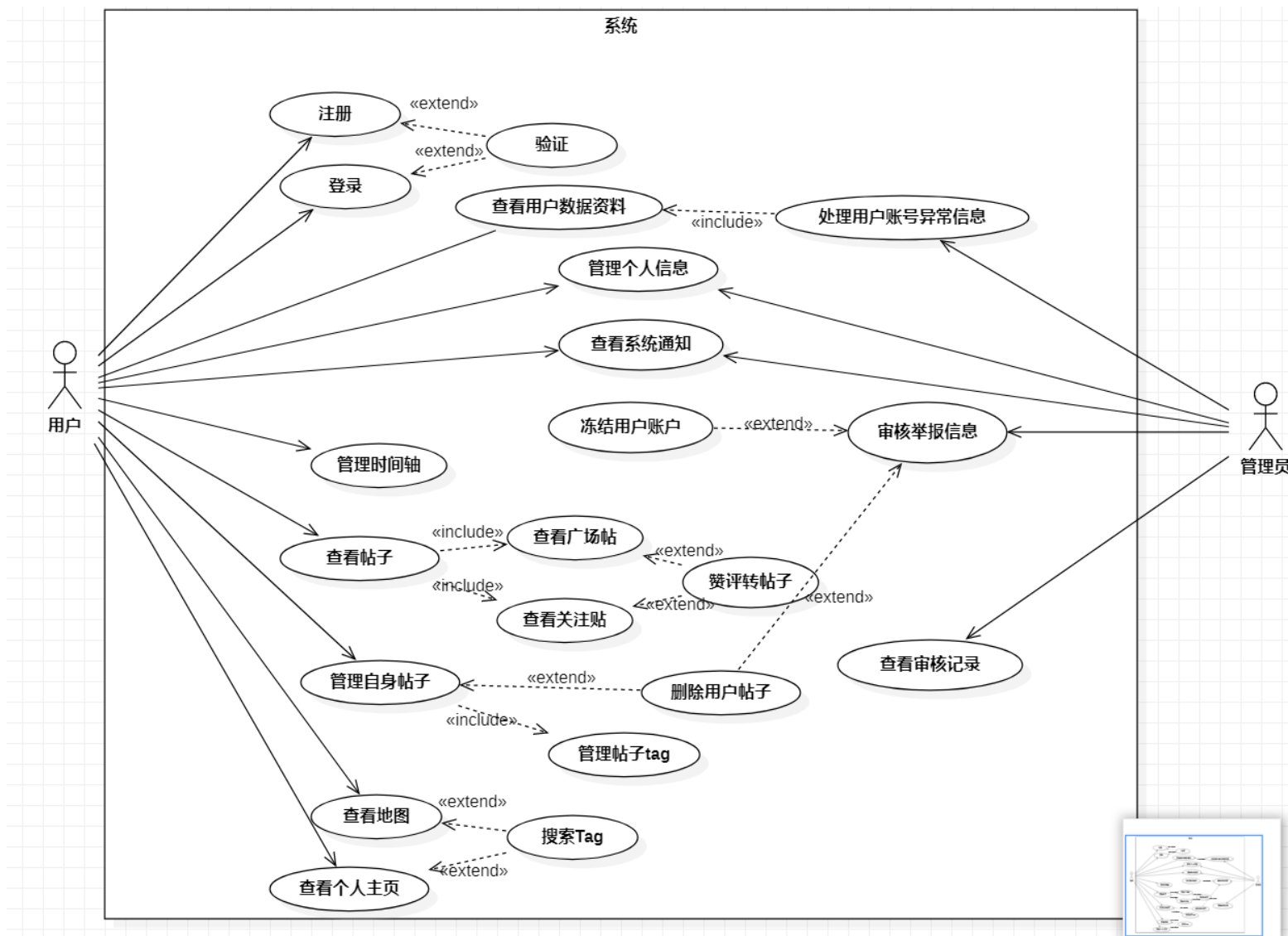
类图

design details



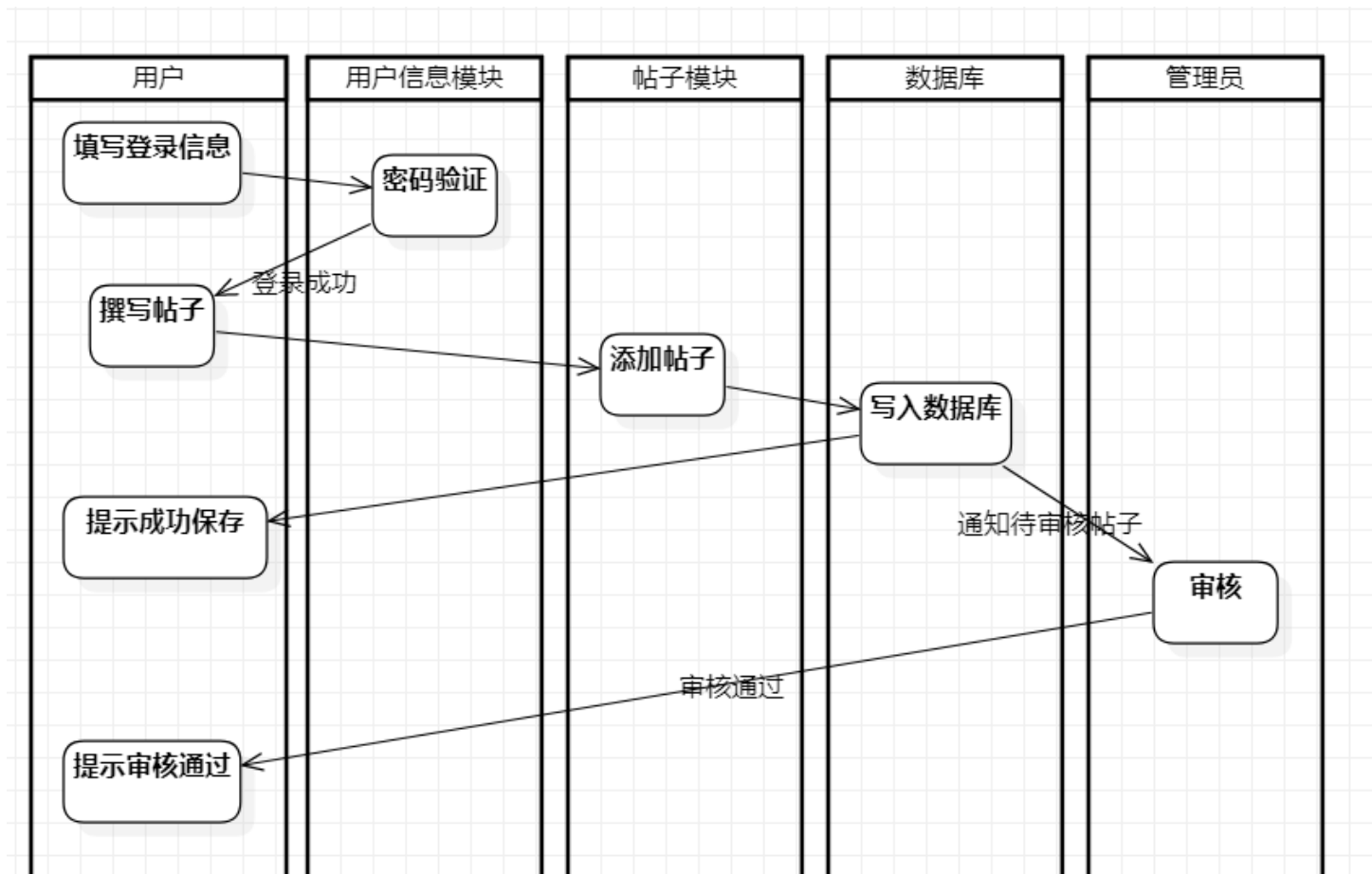
用例图

design details



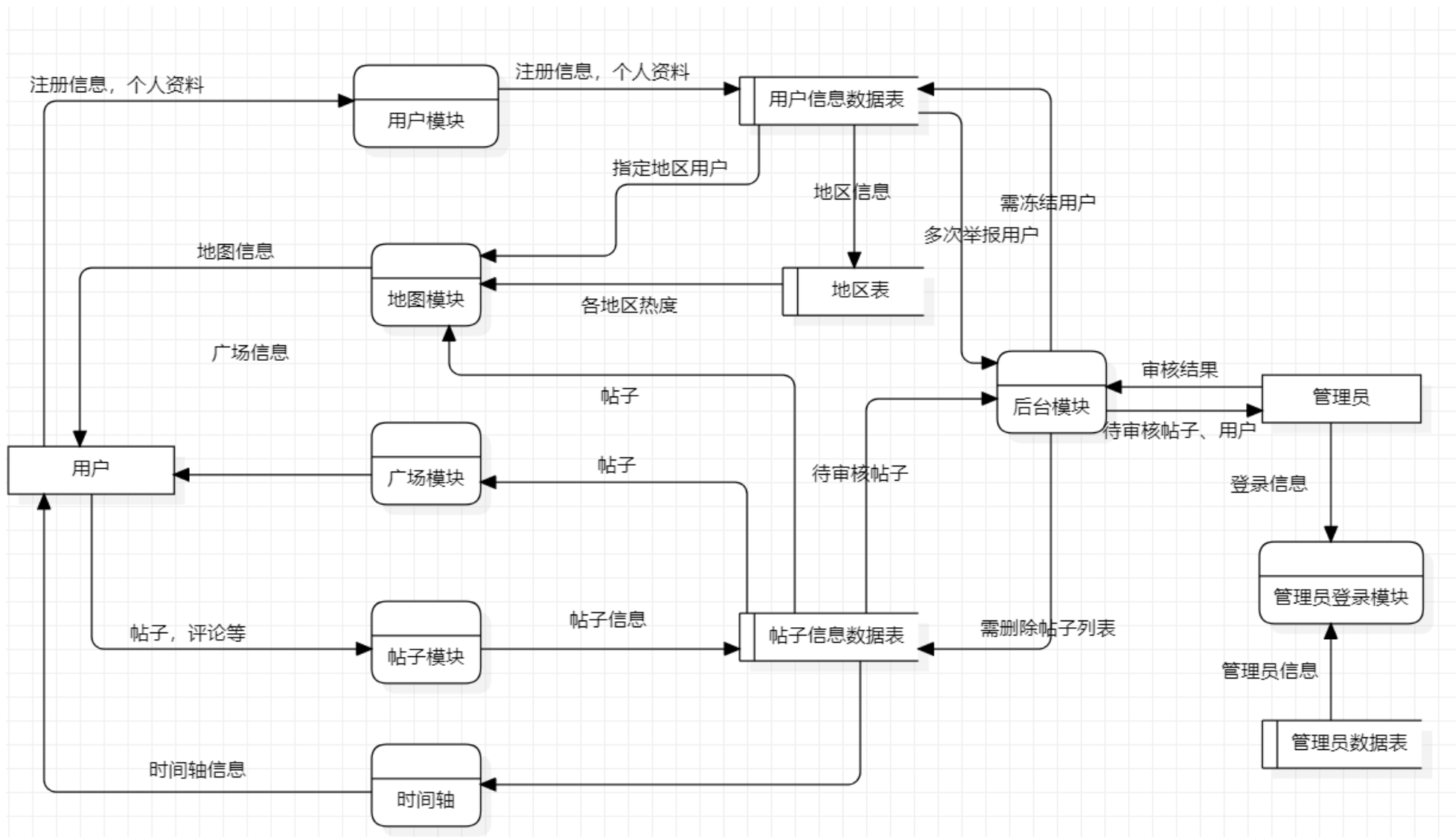
泳道图

design details



数据流图

design details





02

系统设计部分

Details in the design of the system



1 引言.....4

1.1 编写目的.....5

1.2 背景.....5

1.3 参考资料.....5

1.4 术语定义及说明.....5

2 设计概述.....6

2.1 任务和目标.....6

2.1.1 需求概述.....6

2.1.2 运行环境概述.....7

2.1.3 条件与限制.....7

2.1.4 详细设计方法和工具.....7

3 系统详细需求分析.....8

3.1 详细功能需求分析.....8

3.2 详细性能需求分析.....9

3.3 详细资源需求分析.....10

3.4 详细系统运行环境及限制条件分析、接口需求分析.....10

4 总体方案确认.....11

4.1 系统总体结构确认.....11

4.2 系统详细界面划分.....15

4.2.1 应用系统与支撑系统的详细界面划分.....15

4.2.2 系统内部详细界面划分.....15

5 系统详细设计.....17

5.1 系统程序代码架构设计.....17

5.1.1 UI(User Interface)用户界面表示层.....18

5.1.2 BLL(Business Logic Layer)业务逻辑层.....22

5.1.3 DAL(Data Access Layer)数据访问层.....22

5.1.4 Common类库.....22

5.1.5 Entity Class实体类.....22

5.2 系统结构设计及子系统划分.....24

5.3 系统功能模块详细设计.....26

5.3.1 前台子系统.....26

5.3.1.1 广场模块.....26

5.3.1.2 帖子模块.....28

5.3.1.3 用户模块.....36

5.3.1.4 地图模块.....41

5.3.1.5 时间轴模块.....43

5.3.2 后台子系统.....46

5.3.2.1 登录模块.....46

5.3.2.2 审核模块.....47

5.4 系统界面详细设计.....48

5.4.1 外部界面设计.....48

5.4.1.1 广场模块.....48

5.4.1.2 帖子模块.....49

5.4.1.3 用户模块及管理员模块.....53

5.4.1.4 地图模块.....56

5.4.1.5 时间轴模块.....58

5.4.2 内部界面设计.....59

5.4.3 用户界面设计.....62

系统设计文档



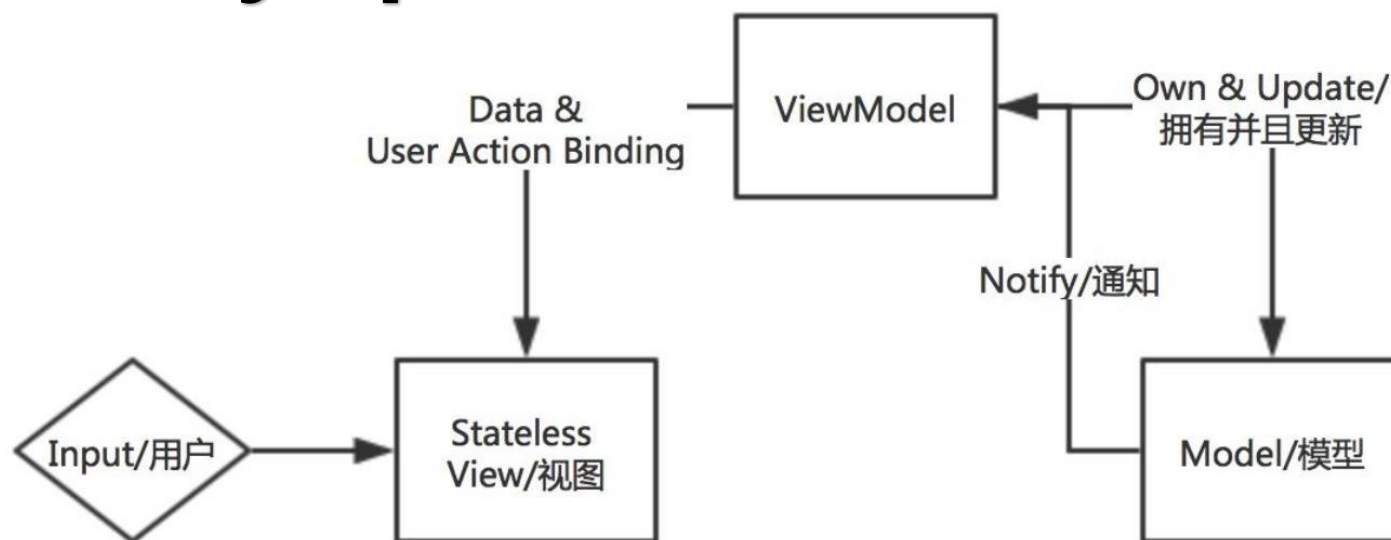
系统整体结构

Details about design of the programming system

MVVM结构

它本质上就是MVC 的改进版。
MVVM 就是将其中的View 的状态和行为抽象化，让我们将视图UI 和业务逻辑分开。

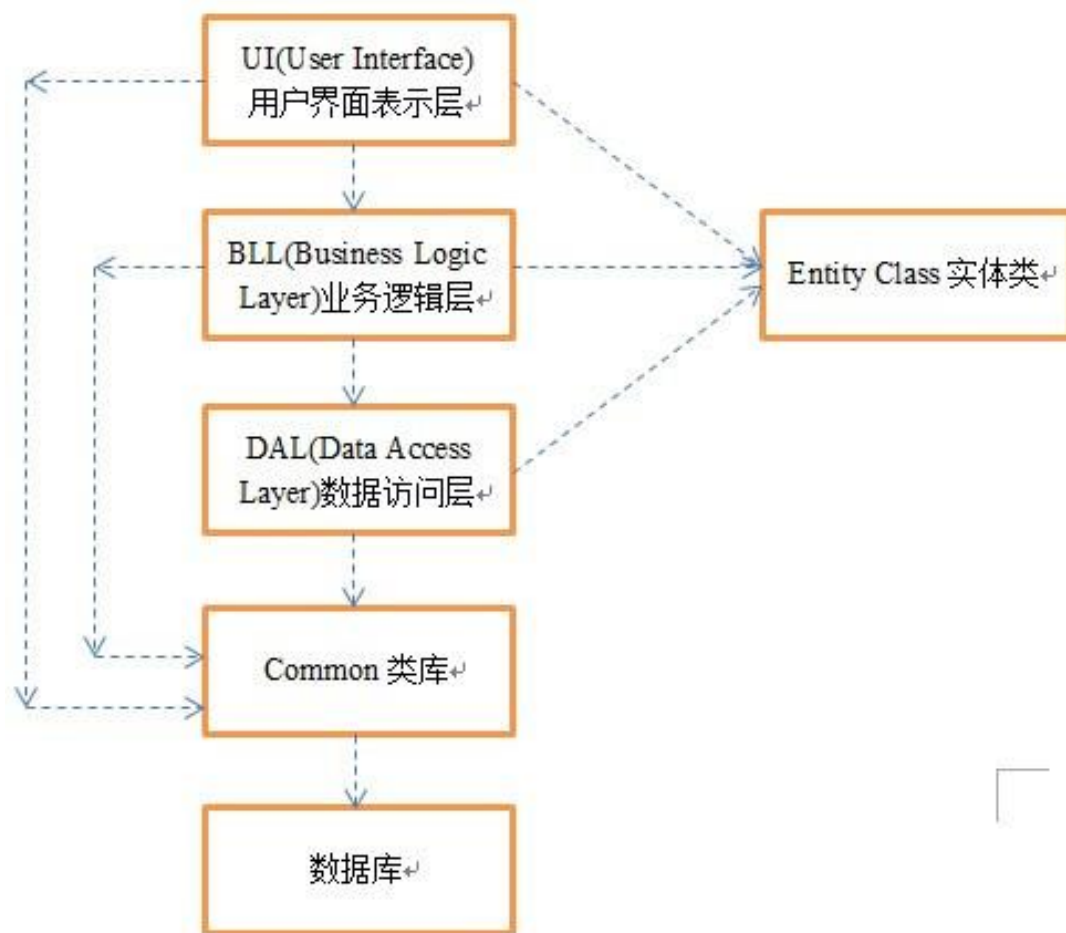
Vue + SpringBoot + Mybatis + Mysql



除了你需要一个为View量身定制的model，这个model就是ViewModel。ViewModel包含所有由UI特定的接口和属性，并由一个 ViewModel 的视图的绑定属性，并可获得二者之间的松散耦合，所以需要在ViewModel 直接更新视图中编写相应代码。

系统代码结构

Details about design of the programming system



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。

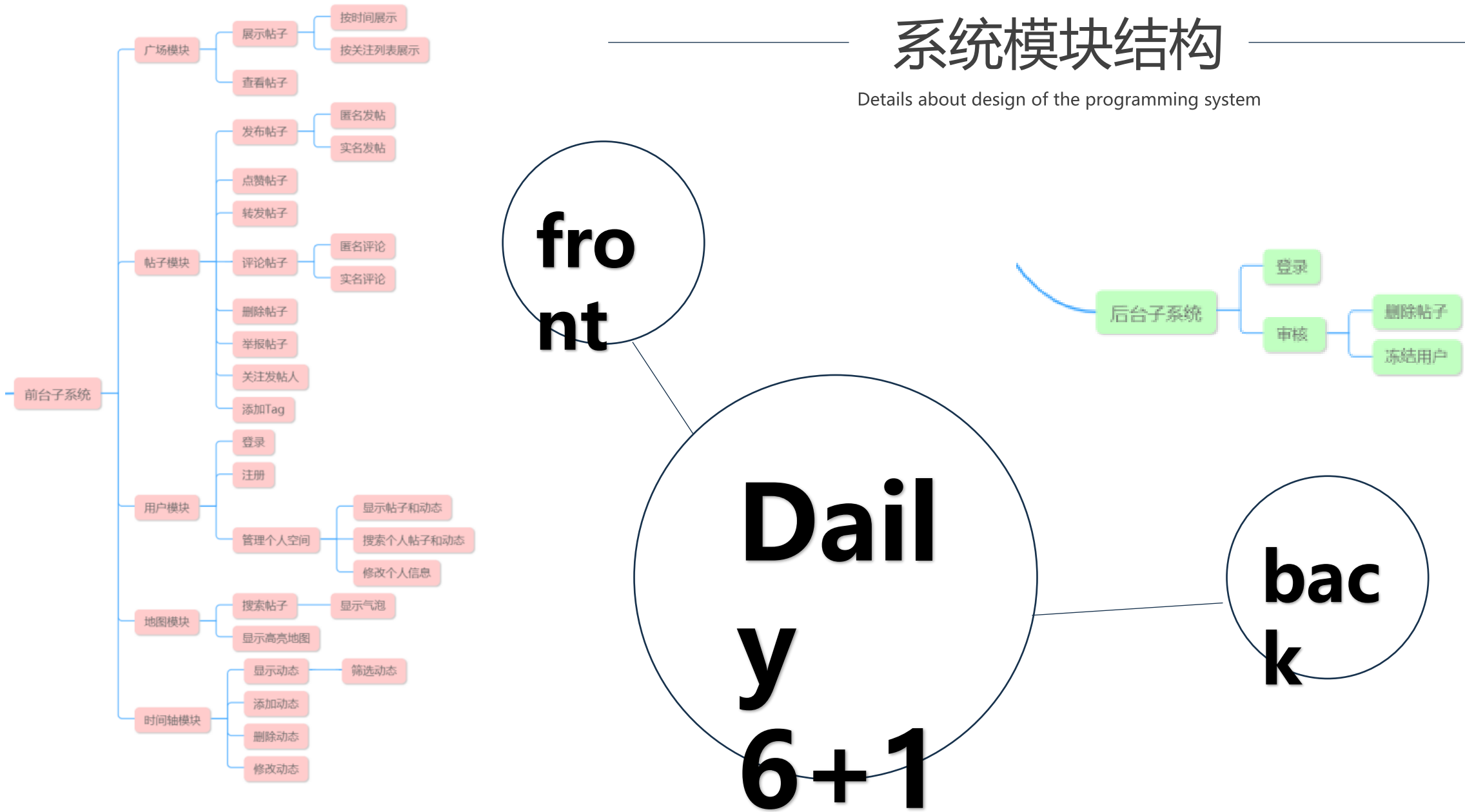
BLL层只关联DAL层和实体类，可能关联Common类库。虽然BLL层被U层调用，但是BLL层无需关心UI层的情况。数据库中每个表都对应一个BLL类，为了达到解耦效果，BLL类不能直接调用其他表的DAL类，可以BLL类之间相互调用。

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

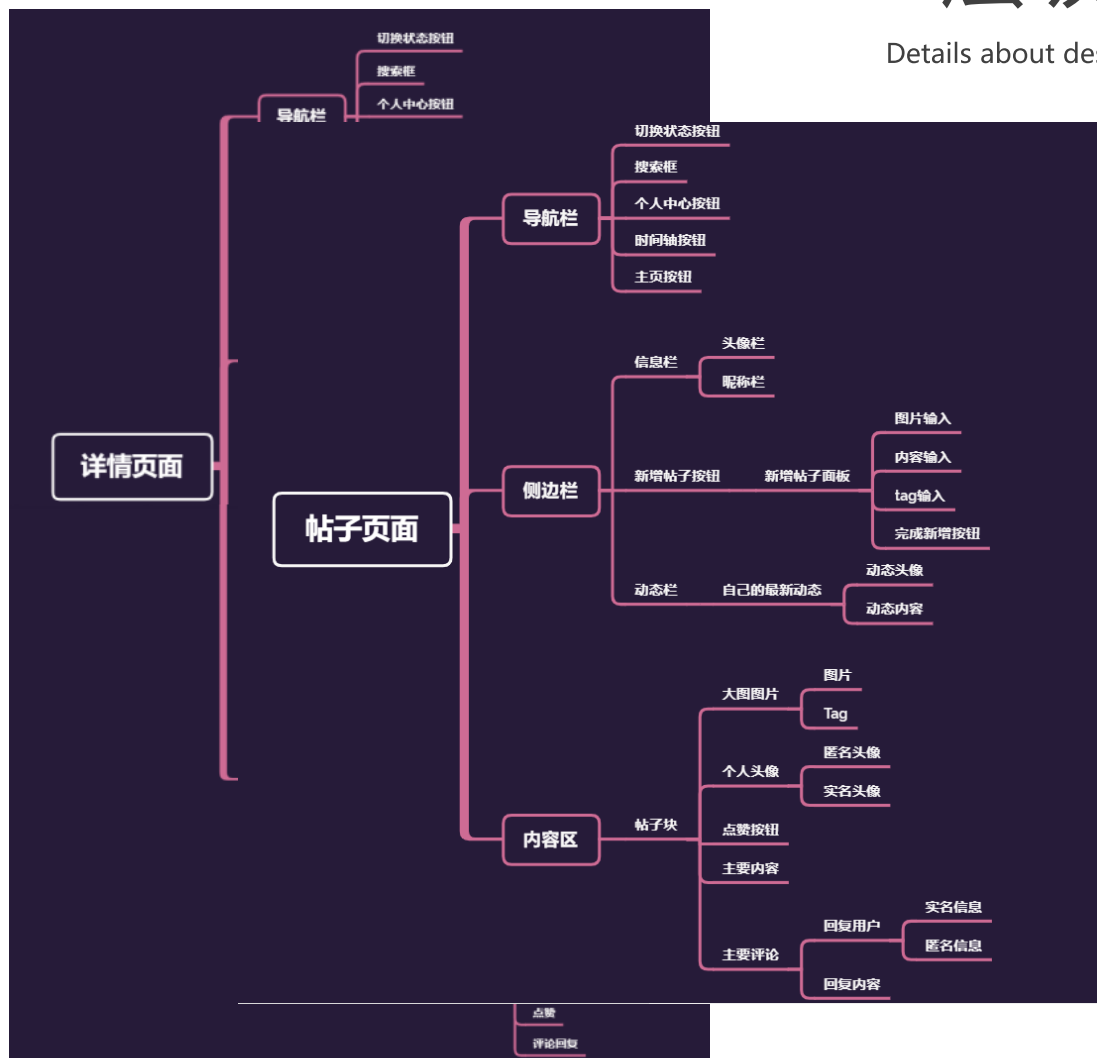
系统模块结构

Details about design of the programming system



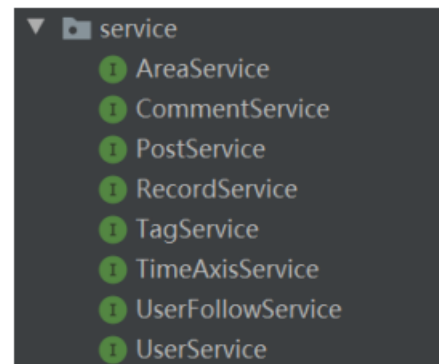
层级模块结构

Details about design of the programming system

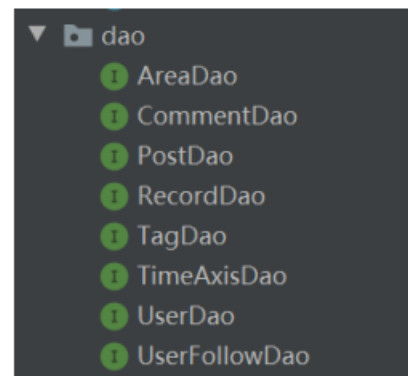


前端

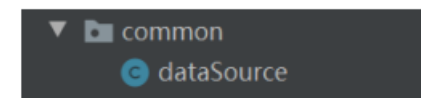
5.1.2 BLL(Business Logic Layer)业务逻辑层



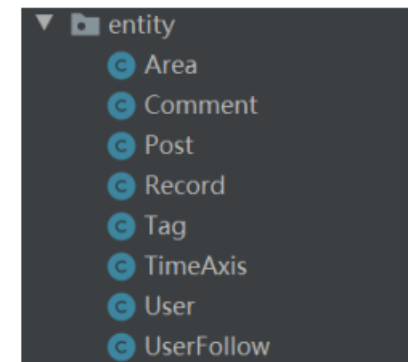
5.1.3 DAL(Data Access Layer)数据访问层



5.1.4 Common类库



5.1.5 Entity Class实体类



后端

模块交互情况

Details about design of the programming system

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。

HIPO图与模块交互

Details about design of the programming system

外部界面数据接口

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

2) 失败，提示

4. 算法描述

1) 前端js验证

2) 通过用户

3) 如果该用

4) 否则，根

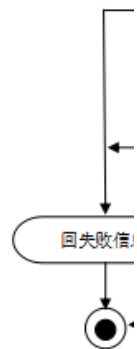
5.3.1.3 用户模块

模块描述：主要针对

主要功能：登录、

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

2) 失败，UI提示

4. 算法描述

1) 前端js验证必填

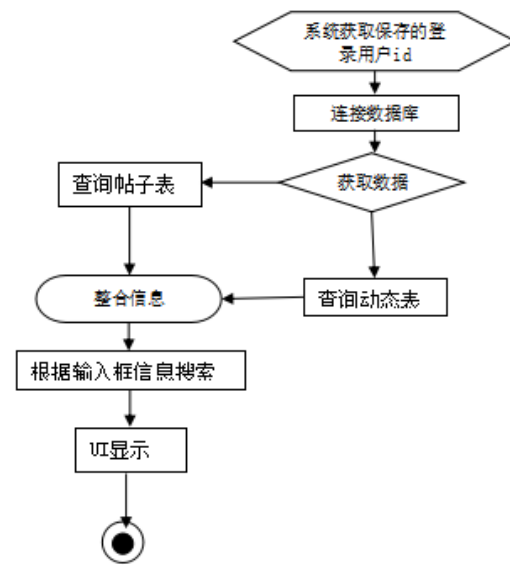
2) 通过用户名查

3) 密码加密与解

4) 根据密码正确

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。

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

内部界面数据接口

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	Integer	用户ID

返回参数

名称	类型	说明
timeAxisId	Integer	时间轴ID



03

数据库设计

Details in the design of the database



外部设计

Details in the design of the Database

表名	主键	描述
tb_area	area_id	地区表
tb_comment	comment_id	保存发布的评论
tb_post	post_id	保存发布的帖子
tb_record	record_id	保存发布的动态
tb_tag	tag_id	保存帖子的标签
tb_time_axis	time_axis_id	保存时间轴
tb_user	user_id	保存用户信息
tb_user_follow	user_follow_id	保存用户之间关注的关系

标识符和状态

数据库对于外部展现的表名以及表内部的状态

外部使用和约定

数据库表在被外部系统——即Daily6+1得前后台系统使用的对应情况表以及数据库建立的约定。

G. 2. 2 使用它的程序

应用程序	访问的数据表
登录	用户表
审核帖子	帖子表
冻结用户	用户表
广场显示帖子	帖子表
发布、点赞、转发、评论、举报、删除帖子	帖子表
搜索帖子	帖子表、地区表
为帖子添加标签	帖子表、标签表
关注发帖人	帖子表、用户关注表
显示高亮地图	帖子表、地区表
时间轴显示动态	动态表、时间轴表
添加动态	动态表
删除动态	动态表
修改动态	动态表
修改个人信息	用户表

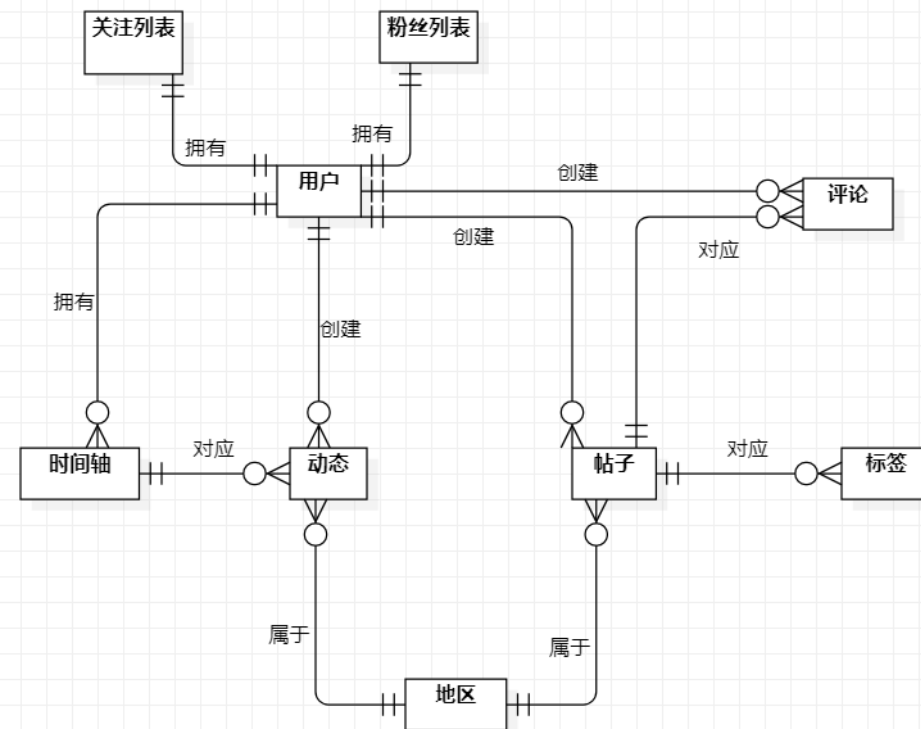
概念与实现

Details in the design of the Database

E-R图也称实体-联系图 (Entity Relationship Diagram)，提供了表示实体类型、属性和联系的方法，用来描述现实世界的概念模型。

数据库整体的ER图应该是什么样的呢？

ER



概念与实现

Details in the design of the Database

用户模块

实体：用户

主要功能：显示帖子和动态，搜索帖子和动态，修改个人信息，登录。



表 tb_user 的结构（用于保存用户的相关信息）

字段名	数据类型	长度	主键	非空	描述
user_id	整数	/	是	是	自增
user_type	整数	/	否	是	用户类型（普通用户0, 管理员1）
user_name	字符串	45	否	是	用户名
user_pwd	字符串	45	否	是	用户密码, 采用某种算法加密后存储。
user_img	URL	1024	否	是	用户头像, 上传到数据库转换成对应的 URL 格式
area_id	整数	/	否	是	用户所属地区 ID
user_date	时间格式	/	否	是	用户出生日期
gender	字符串	2	否	是	用户性别
profile	字符串	1024	否	是	用户简介
state	整数	/	否	是	是否可用（可用1, 不可用0）
fans_num	整数	/	否	是	粉丝数
follow_num	整数	/	否	是	关注数

安全与权限

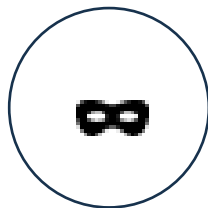
Details in the design of the Database

安全措施



后台过滤

后端设置过滤机制，使用过滤器对没有注册登录用户的请求进行拦截，不予放行，防止非法用户恶意操作，只有经过常规途径注册并登录的用户才能使用系统。



数据加密

对存储和传输的数据进行加密处理，从而使得不知道解密算法的人无法获知数据的内容。



视图控制

为不同的用户定义视图，通过视图机制把要保密的数据对无权存取的用户隐藏起来，从而自动地对数据提供一定程度的安全保护。



防止注入

前端对输入进行验证，防止诸如“SELECT USER”等语句的输入导致后台执行数据库操作泄露数据库中数据。。

保密



04

开发与分工

Development and division of labor



开发与分工

Development and division of labor

前端

前端UI的展示，页面的跳转，
地图API以及其他视觉效果的配置

221701104 潘晨宇
110700516 林泠
221701132 林杰



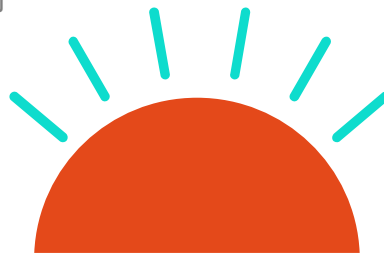
页面美化和逻辑

调整页面的跳转、整理UI
让界面更加友好



算法逻辑

将后台的部分逻辑转化为
逻辑代码，连接数据库和
前端的桥梁



灵感分工！



页面逻辑编写

API的调用，后端数据的
调用，能让数据展现出来



数据库

编写数据库逻辑代码，提
供数据库接口，让前端能
够使用数据。

221701104 邵
研
221701116 陈
炎
221701141 马
科学
091700403 陈
静 后端

数据的整理筛选，视频、图片、
文字的存放，用户的管理，提
供接口给前端使用

开发与分工

Development and division of labor

091700403 : 时间轴页面和平台页面
功能的后端接口与逻辑代码编写

221701141 : 个人主页、时间轴和地
图页面的后端接口与逻辑代码编写

221701104 : 帖子页面、个人
主页页面的前端编写。

221701132 : 时间轴页面、地
图页面的前端编写

110700516 : 时间轴页面、地
图页面的前端编写

221701105 : 个人主页与管理员操作
后端接口与逻辑代码编写。

221701116 : 平台页面和地图页面功
能的接口与逻辑代码编写

第四阶段

第四周，UI美化与效果渲
染，热门算法更新

第三阶段

第三周，功能链接完成，后端
接口实现测试无误，执行前后
端链接

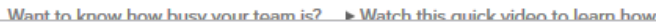
第二阶段

第二周，前端页面UI完成逻辑，
完成基本跳转，后端接口实现

第一阶段

第一周，前端UI基本结构建立
与美化效果，后端读取与简单
操作完成

Development and division of labor

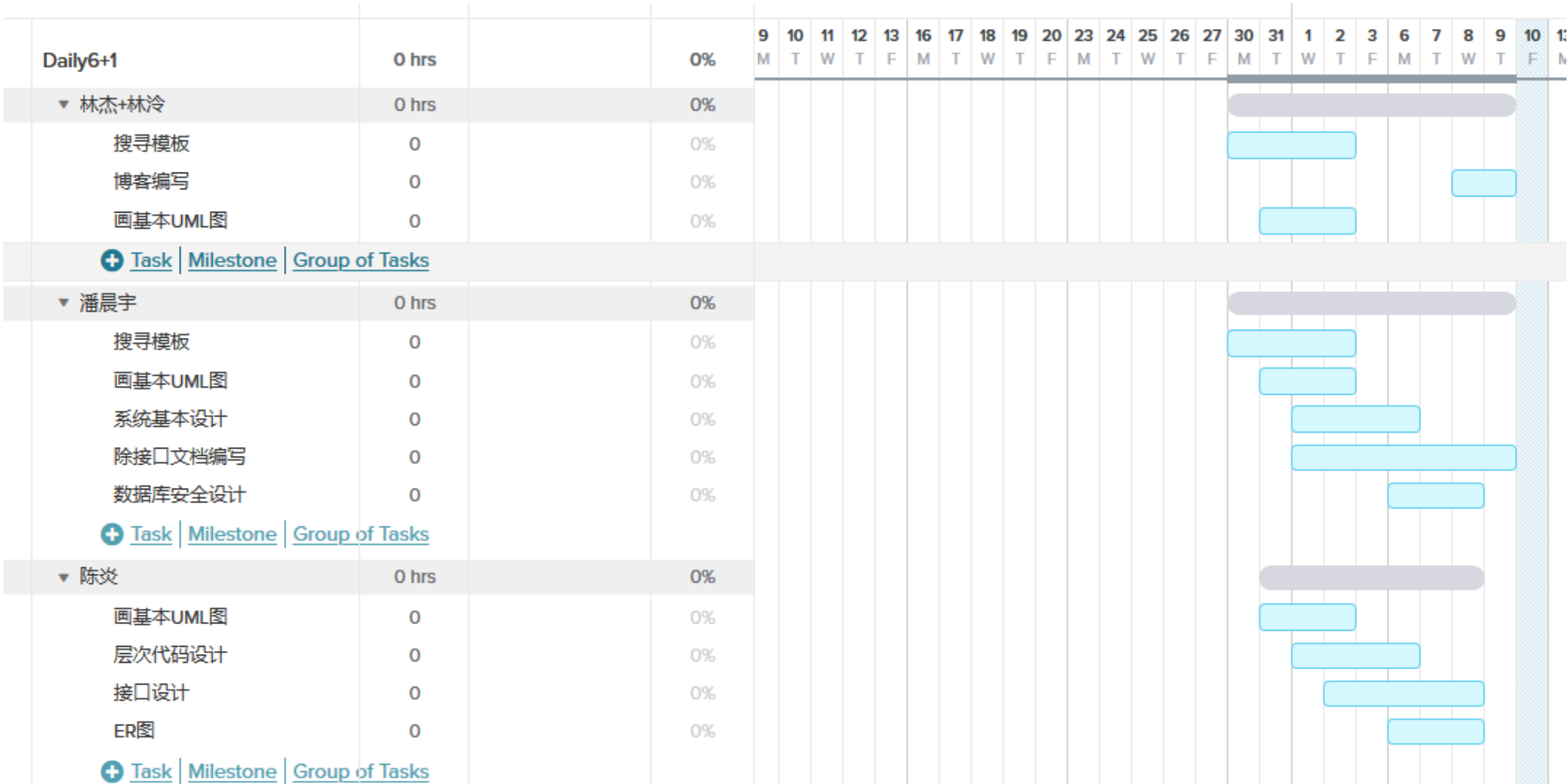


panchenyu

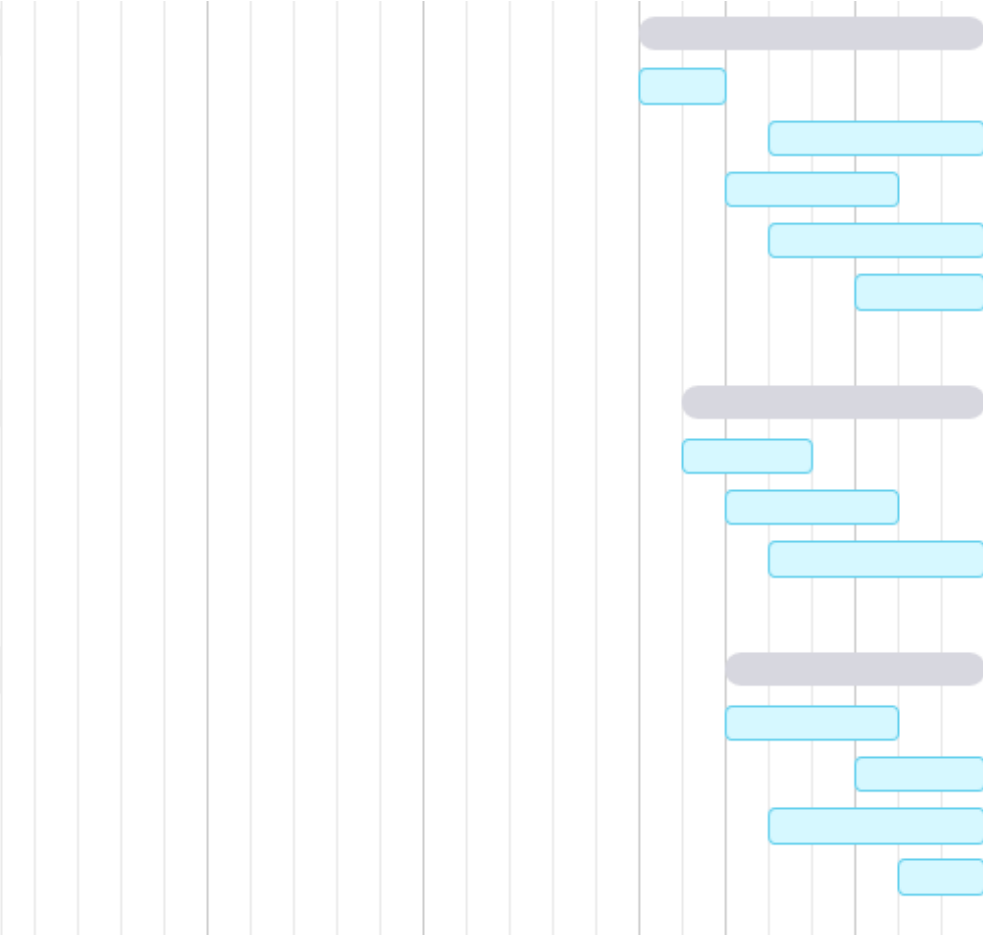
Unassigned	0	0	0	3	6	6	9	8	7	7	7	7	6	5	5	6	6	8	8	7	8	7	5	4	4	4	3	6	4	4	4	4	4	4	4	3	3	3	3	3	3	2	2	3	3	2	2	2	2	3	2	2	2	2	0	0	0
------------	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

开发与分工

Development and division of labor



▼ 陈静	0 hrs	0%
层次设计	0	0%
数据库基本设计	0	0%
层次代码设计	0	0%
接口设计	0	0%
数据库安全设计	0	0%
+ Task Milestone Group of Tasks		
▼ 邵研	0 hrs	0%
画基本UML图	0	0%
层次代码设计	0	0%
接口设计	0	0%
+ Task Milestone Group of Tasks		
▼ 马科学	0 hrs	0%
层次代码设计	0	0%
ER图	0	0%
接口设计	0	0%
后端文档整合	0	0%
+ Task Milestone Group of Tasks		



汇报人：潘晨宇汇报时间：04/11

谢谢观看

一个关注用户极简、直观体验，希冀纯享功能的私密交流WEBAPP