

福州大学

《系统设计说明书》

小组：第九组-Mobius

班级：软件工程 W 班

年级：2019 级

专业：软件工程

系统设计说明书

0. 文档介绍

0.1 文档目的与范围

0.1.1 目的

编写莫比乌斯软件系统设计说明书的主要目的是对莫比乌斯进行详细设计，在概要设计的基础上进一步明确系统结构，详细的介绍系统各个模块，并对项目功能进行了具体描述，对前后端界面和功能实现体会出了具体要求以指导前后端开发，让开发人员能够更好的投入到软件开发中。

0.1.2 范围

莫比乌斯主要的面向群体是当代年轻人，现在年轻人的压力越来越大，导致他们的身心都越来越疲惫，由于大家的社恐，倾诉对象越来越少。为了能够让大家有一个分享自己的不快乐的地方，莫比乌斯推出了一下几个业务。

1. 向陌生人发送信件
2. 查看来自陌生人的信件
3. 和陌生人成为笔友
4. 和笔友交谈

0.2 读者对象

读者对象有用户，需求分析人员，项目开发人员，测试人员，文档编辑人员等。

0.3 参考文献

[1]邹欣, 构建之法 (第三版), 人民邮电出版社, 2017

1. 系统概述

现在的社会竞争越来越激烈, 年轻人的生活压力越来越大, 生活中产生了很多忧虑和不良情绪。然而由于忙于工作, 年轻人线下的社交在逐渐减少, 大家都想通过社交软件将内心的忧郁倾诉出来, 但是因为身处在一个快节奏的时代中, 人们很难在网络世界中找到真正的知心朋友。所以在此基础上我们提出了一个社交软件——解忧杂货店, 它通过的书信的方式与他人交流, 从而抒发内心的忧郁, 使人们能在这个快节奏的时代找到真正可以倾诉心声的人。

2. 文档设计规范

一级标题: 默认字体小二

二级标题: 默认字体三号

正文: 宋体小四

3. 开发、测试与运行环境

类别	标准配置	用途
开发环境	IDEA2019	后端开发
	Android Studio 7.0+	移动端开发
	Visual Studio Code	前端开发
测试环境		后端测试
	谷歌、火狐浏览器	前端测试
		移动端测试
运行环境	CentOS 7	服务器部署

4. 软件系统结构图

4.1 体系结构设计图

4.1.1 系统结构描述

系统采用 MVC 模式进行设计, 前后端分离。当后端团队与前端团队共同设计协调

好接口后，可实现前后端并行开发，提高开发效率。由于视图层与业务层分离，耦合性低，业务层的各层次职责单一，所以更易于后期维护。

4.1.2 各层次关系职责说明

(1) Controller 层：

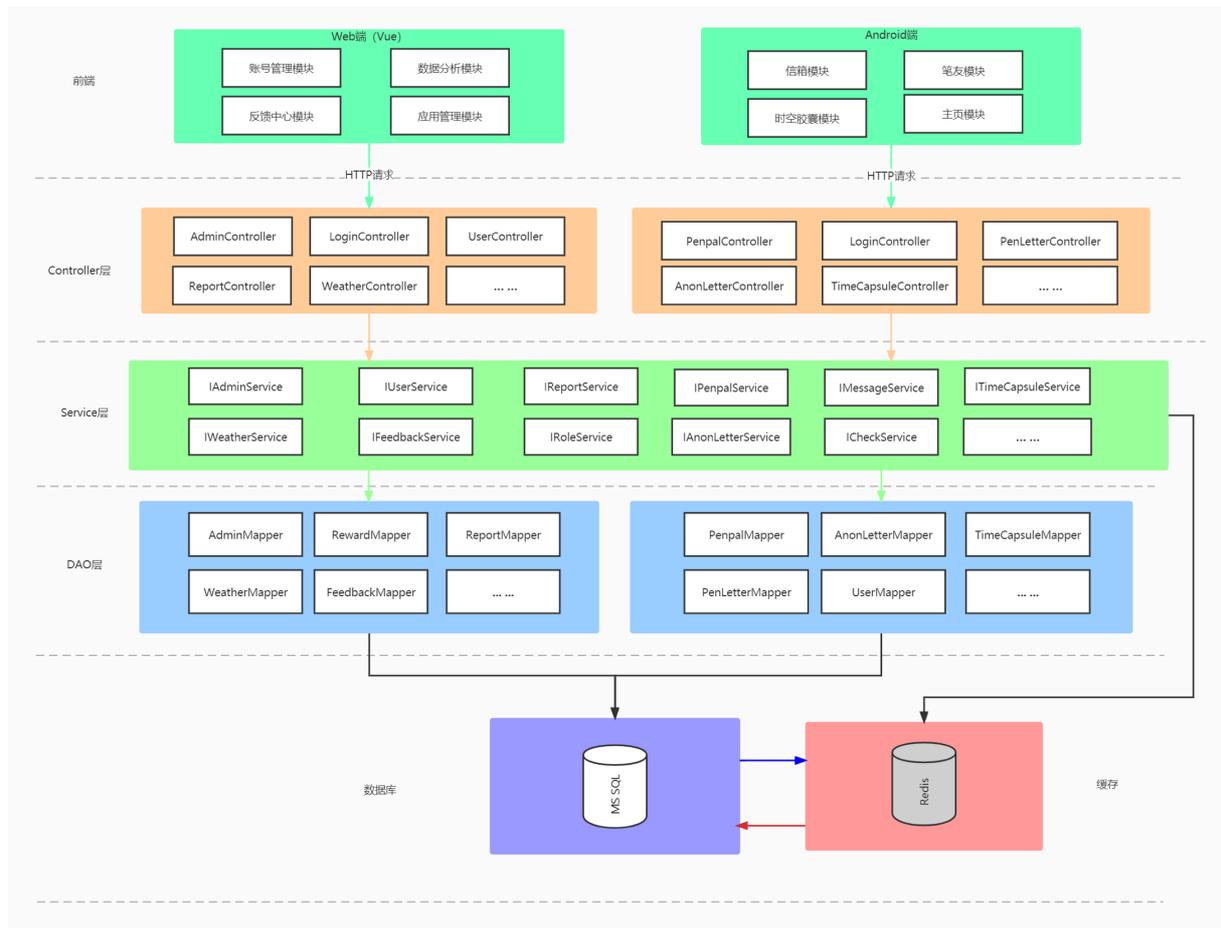
负责接收前端的 Http 请求，根据请求路径匹配相应控制器进行处理，并调用 Service 层获取所需数据，将数据返回给前端展示。

(2) Service 层：

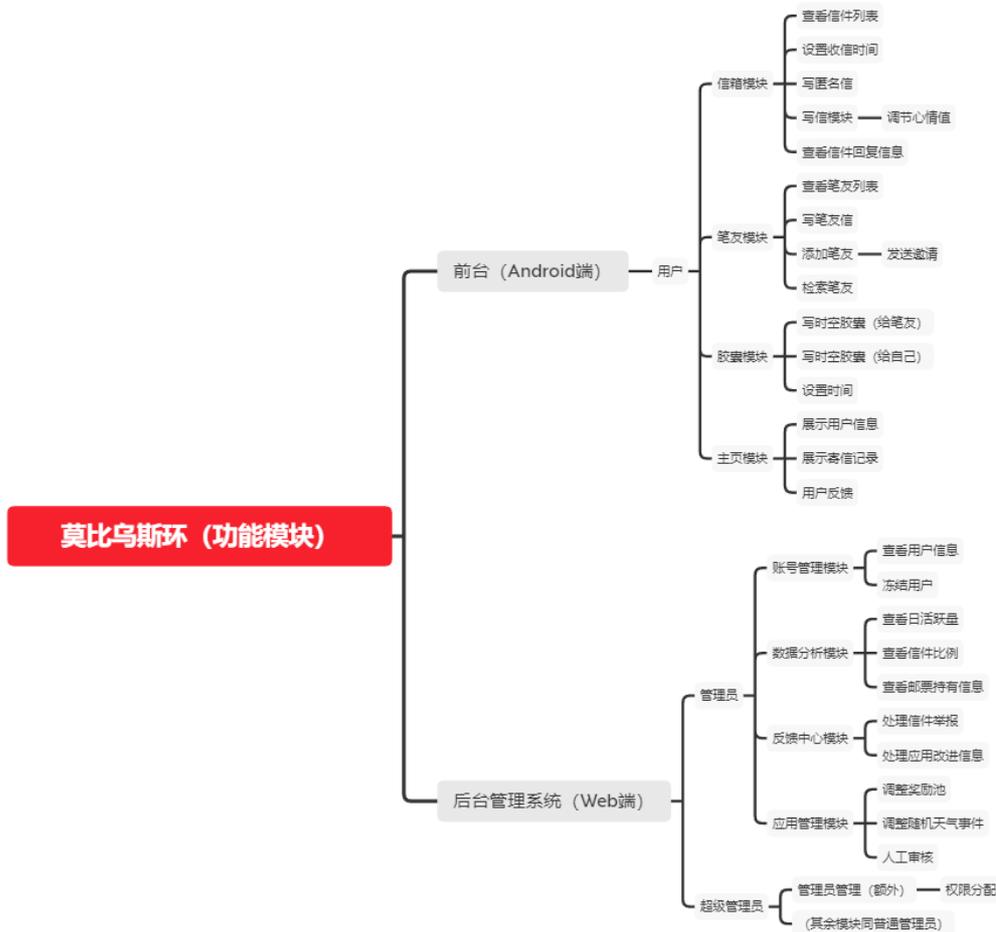
由 Controller 调用，调用 DAO 层获取所需的基本数据，主要进行数据的处理和封装，并将处理完成的数据交给 Controller 返回给前端。

(3) DAO 层：

由 Service 层调用，定义一系列操作数据库的方法来与数据库进行交互，根据 Service 层的需求访问数据库，获取数据，并封装为模型对象或对象集合交予 Service 层进行进一步加工处理。



4.2 功能模块层次图



4.2.1 划分依据与原因

本项目前台(Android端)主要分成4个模块(信箱模块、笔友模块、胶囊模块、主页模块)。用户可以通过四个模块进行写信、收信操作;

后台管理系统(Web端)主要分为“管理员”、“超级管理员”,

其中“管理员”主要拥有4个功能模块,分别是“账号管理模块”、“数据分析模块”、“反馈中心模块”、“应用管理模块”,

而“超级管理员”主要拥有“管理员管理”和“管理员拥有的全部功能”两个功能模块。

划分的主要依据是按照实际的功能划分,每个模块的功能单一,操作对象也尽可能单一,其目的是为了使得各个模块之间尽可能相互独立,在开发分工中可以并行开发不同的模块,避免因耦合度太高而带来合作难度的提高;

4.2.2 模块介绍

4.2.2.1 前台(Android端)模块介绍:

(1) 信箱模块

该模块主要是实现“查看信件列表”、“设置收信时间”、“写匿名信”、“写信模块”、“查看信件回复信息”等功能。让用户体验写信的快乐。

1) “查看信件列表”功能介绍:

该功能可以实现“信箱”的作用，其可将用户收到的信件存入信箱并以列表的形式展现。

其中信件的类型分为:“普通信件”、“陌生信件”、“笔友信件”、“匿名信件”、“时空信件”。

2) “设置收信时间”功能介绍:

该功能可以设置发送信件的时间和对方接收信件的时间。

3) “写匿名信”功能介绍:

“匿名信”就是隐藏个人信息的信件；该功能可以设置自己的信件为匿名信；该匿名信可以发送给附近的朋友、笔友、陌生人。

4) “写信模板”功能介绍:

“写信模板”顾名思义就是写信；该功能可以将自己写的信件发送给附近的朋友、笔友、陌生人。

信件可以获取到写信人的心情，收信人可以了解到寄信人的心情状态。

5) “查看信件回复信息”功能介绍:

“查看信件回复信息”；该功能可以获取到朋友、笔友、陌生人等收到的信件。

(2) 笔友模块

该模块主要是实现“查看笔友列表”、“写笔友信”、“添加笔友”、“检索笔友”等功能。让用户体验笔友的快乐。

1) “查看笔友列表”功能介绍:

该功能可以实现“QQ 好友”的作用，展示好友列表

2) “写笔友信”功能介绍:

“写笔友信”顾名思义就是写信给笔友；该功能可以将自己写的信件发送给笔友。信件可以获取到写信人的心情，收信人可以了解到寄信人的心情状态。

3) “添加笔友”功能介绍:

“添加笔友”就是添加好友；该功能可以邀请你觉得聊的来的朋友

4) “检索笔友”功能介绍:

该功能可以实现查找笔友进行添加。

(3) 胶囊模块

该模块主要是实现对未来写信，将信发送给未来的自己或者笔友。

1) “写时空胶囊”功能介绍:

该功能可以实现“对未来写信”，可将信件发送给未来的自己或者笔友

2) “设置时间”功能介绍:

“设置时间”；该功能可以设置未来的自己收到信件的时间。

(4) 主页模块

该模块主要是实现“展示用户信息”、“展示寄信记录”、“用户反馈”等功能。展示用户个人信息。

1) “展示用户信息”功能介绍:

“展示用户信息”用于展示用户个人信息。

2) “展示寄信记录”功能介绍:

“展示寄信记录”可以展示自己寄出去的信件信息。

3) “用户反馈”功能介绍:

“用户反馈”可以向系统发送对该 App 的建议。系统接到信息后会及时进行反馈和修

改。

4.2.2.2 后台管理系统(Web 端)模块介绍:

管理员:

(1) 账号管理模块

该模块主要是实现“查看用户信息”、“冻结用户”等功能。管理账号信息。

1) “查看用户信息”功能介绍:

管理员可以查看用户的个人信息;

2) “冻结用户”功能介绍:

管理员可以冻结用户 (在某用户存在风险时);

(2) 数据分析模块

该模块主要是实现“查看日活跃量”、“查看信件比例”、“查看邮票持有信息”等功能。管理员可以通过以上功能进行数据分析。

1) “查看日活跃量”功能介绍:

系统会统计前台 App 发送信件数量等来判断日活跃量, 而管理员可以查看日活跃量;

2) “查看信件比例”功能介绍:

系统会统计不同信件的类别进行分类处理, 最后得到信件的比例。而管理员可以查看信件的比例

3) “查看邮票持有信息”功能介绍:

系统会统计每个信件里的信息反馈到后台管理系统。而管理员可以查看邮票持有信息;

(3) 反馈中心模块

该模块主要是实现获取前台的用户对 app 内事件的反馈并进行处理。

1) “处理信件举报”功能介绍:

系统会从前台收到不一样的反馈, 如果为信件举报, 我们会将其分到这类功能进行处理;

2) “处理应用改进信息”功能介绍:

系统会从前台收到不一样的反馈, 如果为应用改进信息, 我们会将其分到这类功能进行处理;

(4) 应用管理模块

该模块主要是实现“调整奖励池”、“调整随机天气事件”、“人工审核”等功能。这里主要为应用的管理, 由管理员和系统进行操作;

1) “调整奖励池”功能介绍:

我们会通过统计发信件的次数等信息, 设置奖励池功能; 而管理员可设置奖励池里的奖品;

2) “调整随机天气事件”功能介绍:

寄信的过程中会受到天气的影响导致送达时间会改变; 而此功能便是设置天气事件; 可通过系统随机生成天气事件; 而管理员可以添加天气;

- 3) “人工审核”功能介绍：
此功能主要是处理一些需要审核的功能；

超级管理员：

(1) 管理员管理模块

该模块主要是实现“权限分配”等功能。

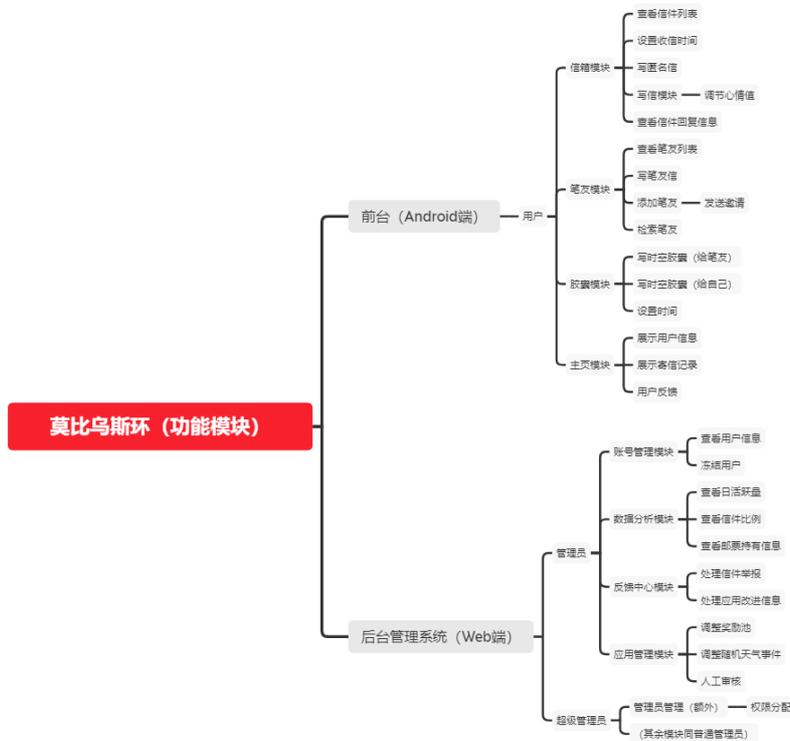
1) “权限分配”功能介绍：

超级管理员可以设置管理员用户的权限；

(2) 管理员拥有的全部模块

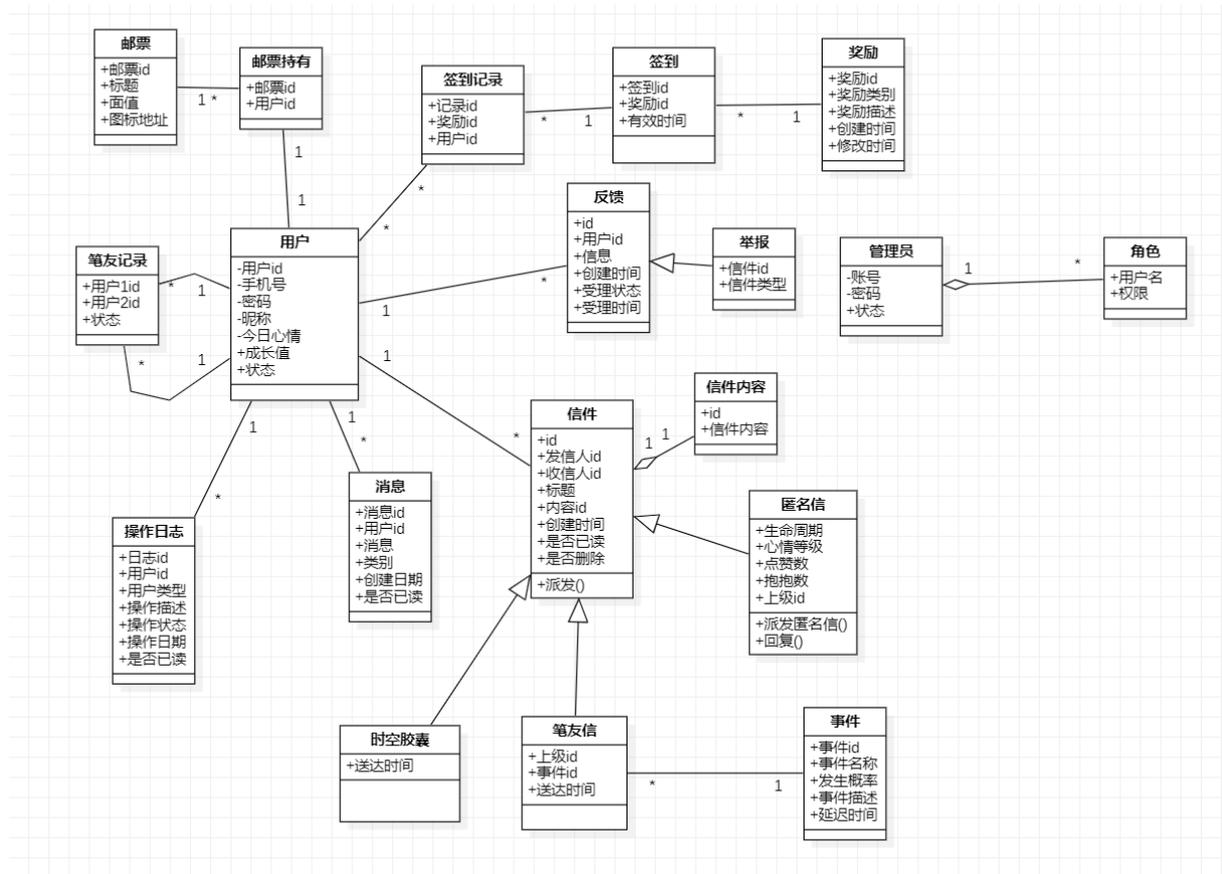
超级管理员拥有管理员的全部功能权限；

功能模块层次图



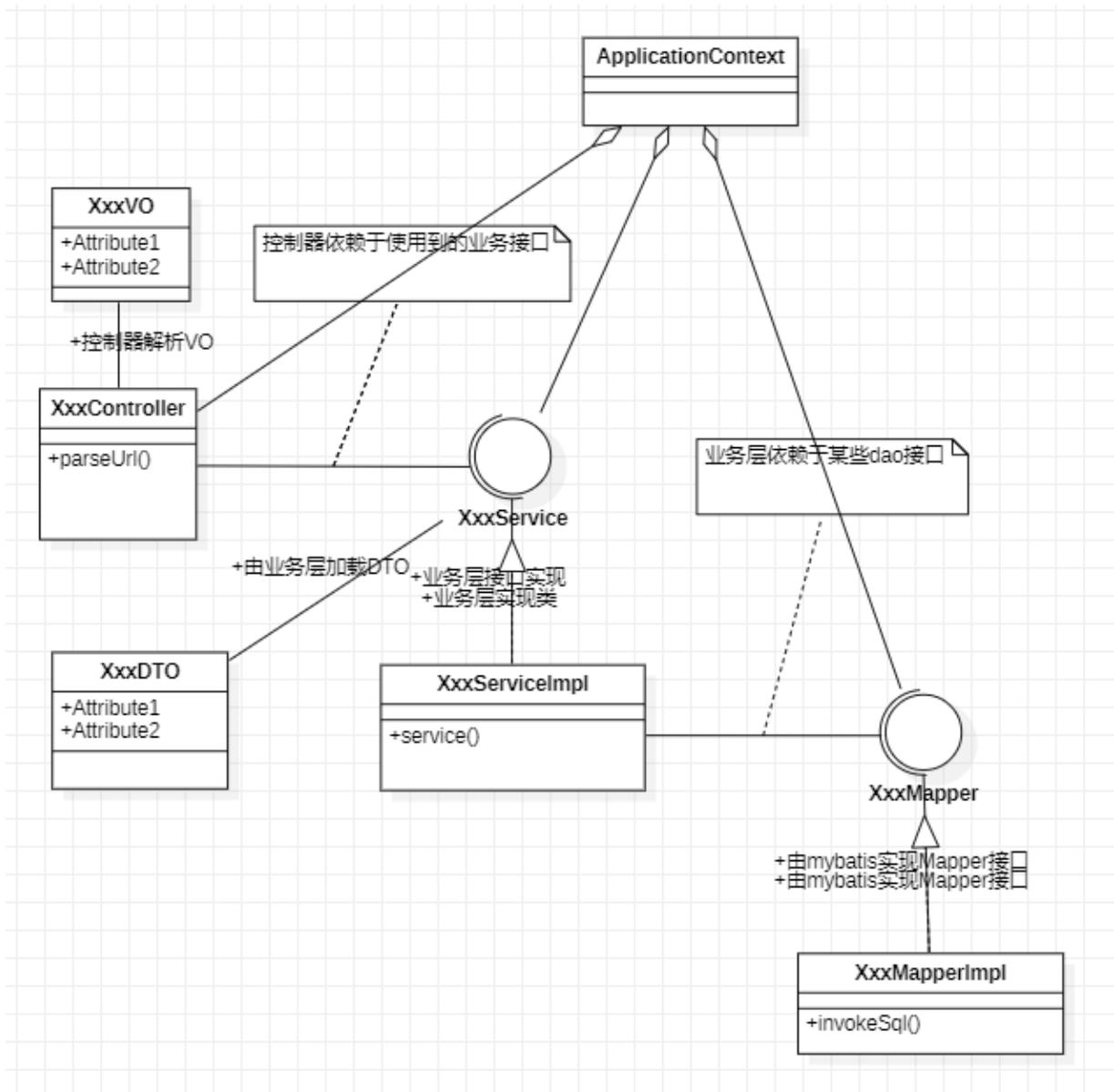
4.3 类图设计

4.3.1 实体类

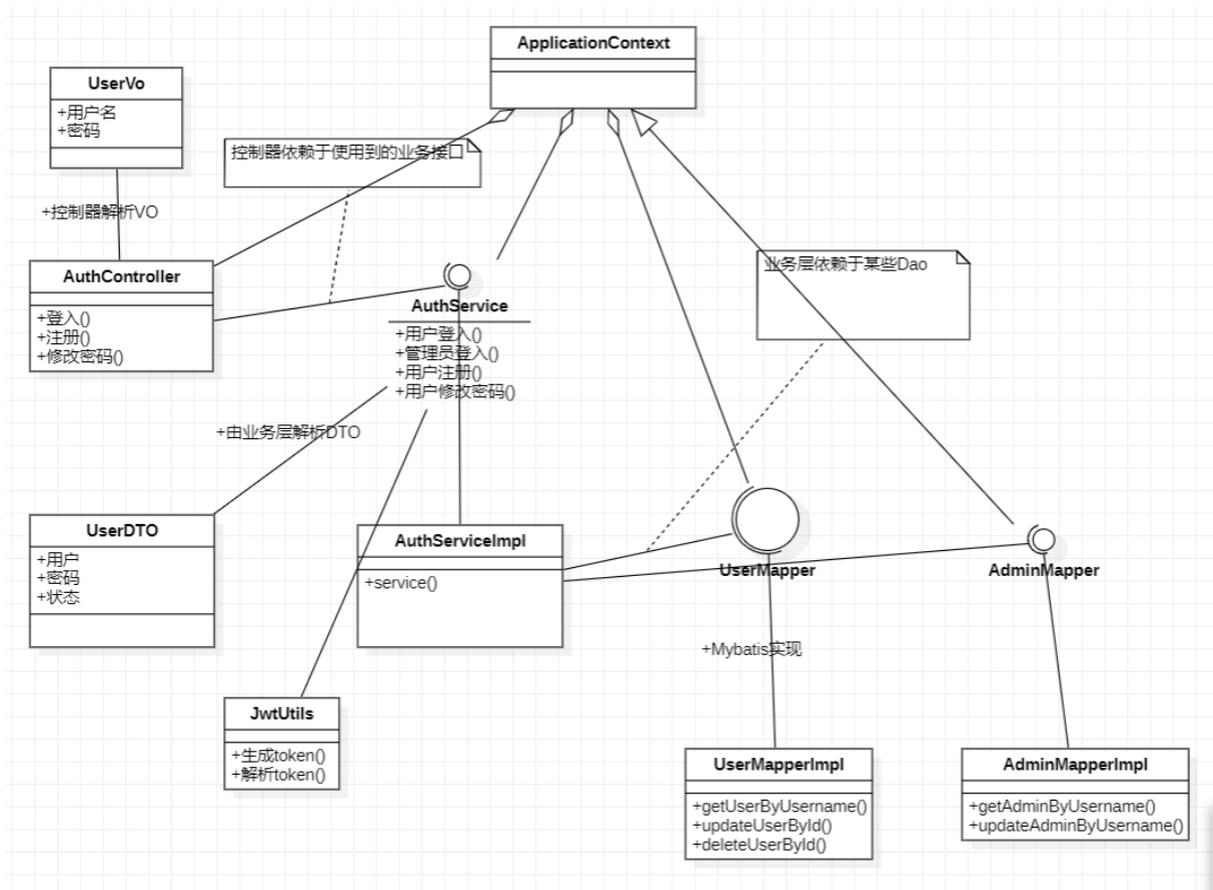


4.3.2 MVC 架构类

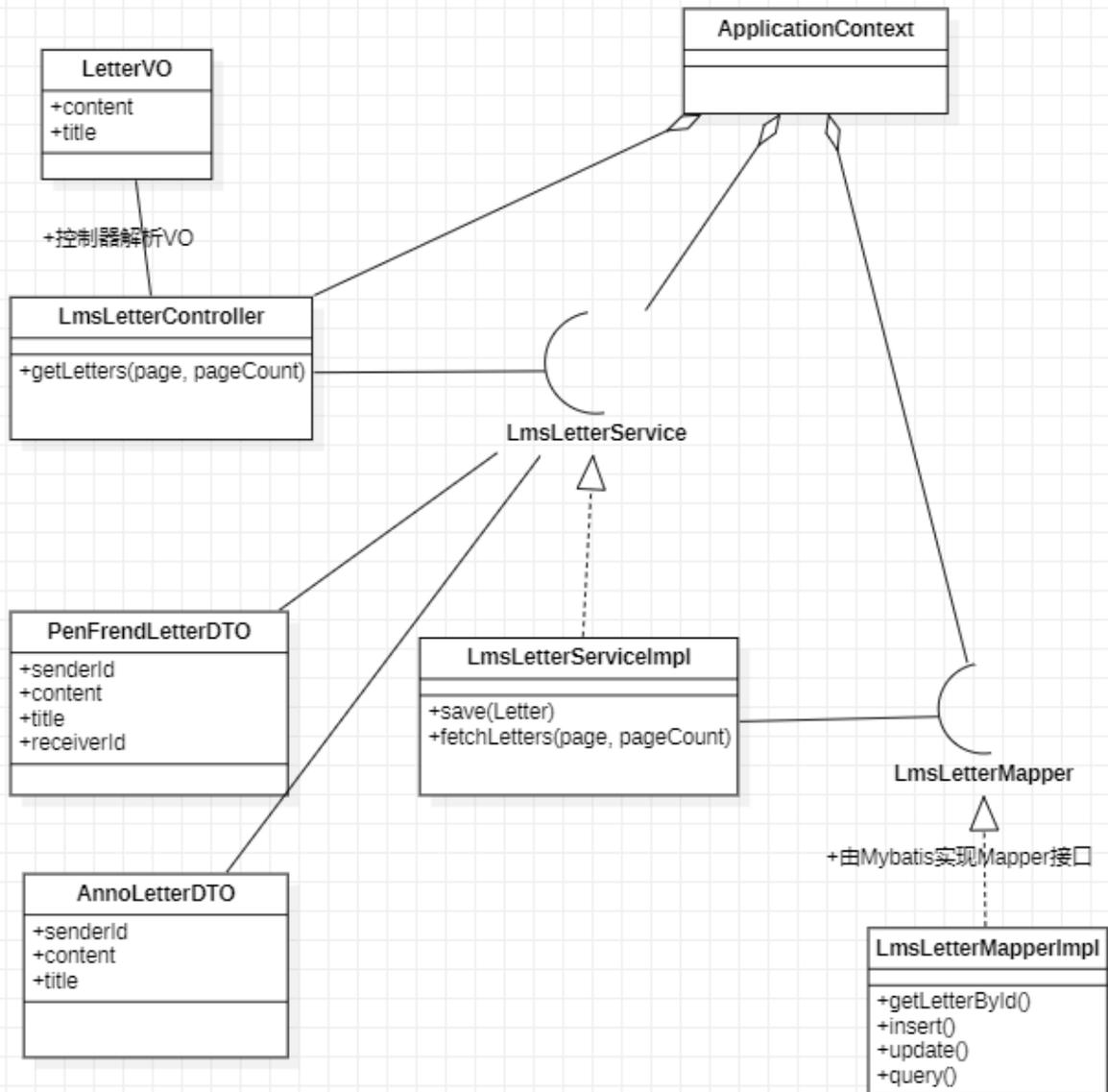
通用业务架构:



用户认证架构：



信件业务架构：



4.4 用例分析

4.4.1 移动端

用户

信件相关

- 写信
 - 写匿名信
 - 写笔友信
 - 写时空胶囊
- 查看信件
 - 查看匿名信
 - 查看笔友信
 - 查看时空胶囊
- 回复信件
- 举报信件
- 查看寄信历史记录

笔友相关

- 邀请成为笔友
- 添加笔友
- 删除笔友
- 检索笔友
- 查看笔友列表

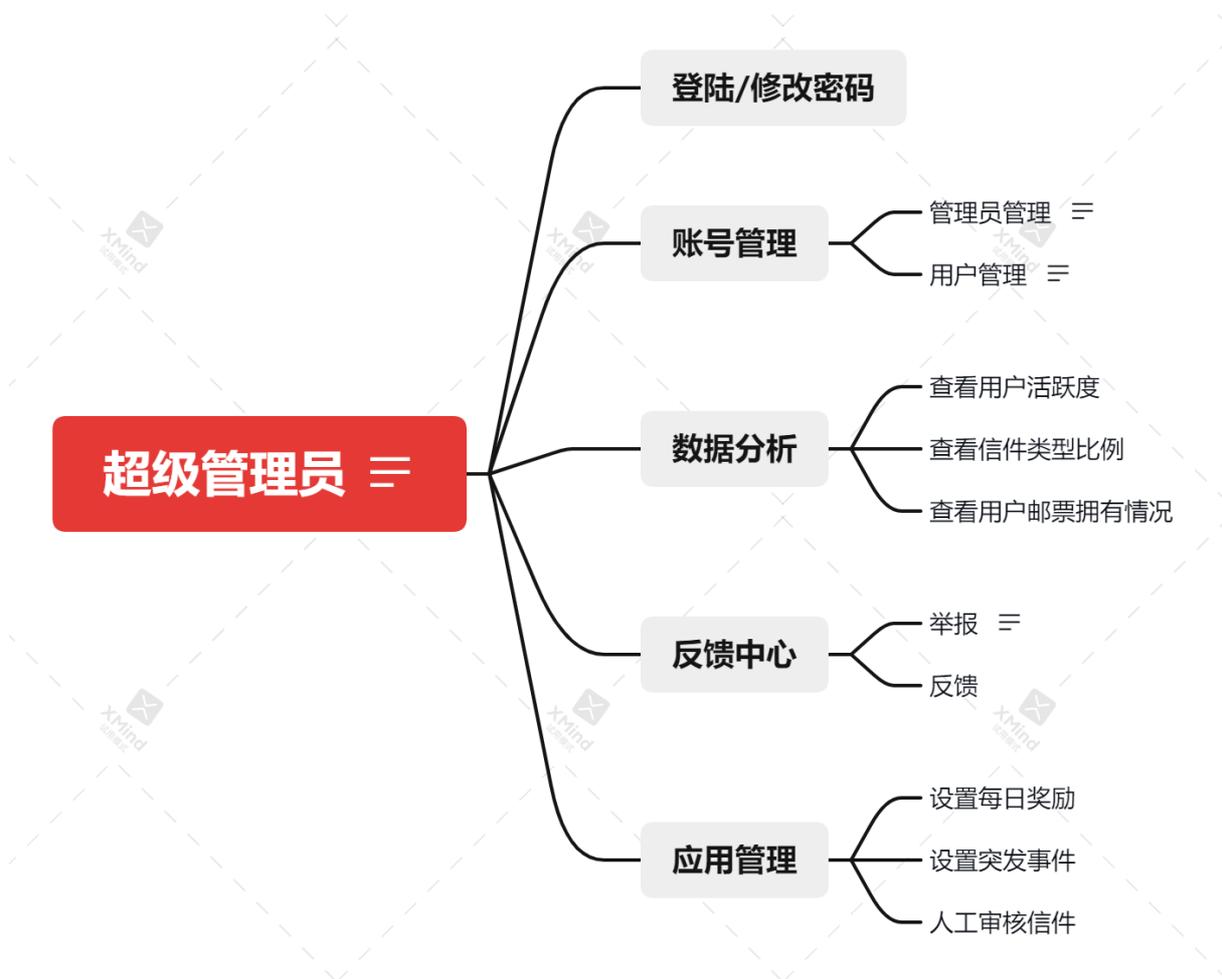
个人信息相关

- 查看个人信息
- 修改个人信息

系统相关

- 设置收信时间
- 设置心情
- 反馈

4.4.2 后台



说明:

- 超级管理员比普通管理员多了一个“管理员管理”的权限。
- 管理员管理：对管理员账号进行增删改查操作。
- 用户管理：对用户进行搜索操作，同时可以冻结违规用户。
- 举报：管理员根据举报详情，判断是否解冻被举报的用户。

4.5 算法结构设计

4.5.1 垃圾邮件分类

主要采用了深度学习中 CNN 来实现分类，通过 Pytorch 实现 TextCNN 来进行分类。网络结构简单导致参数数目少，计算量少，训练速度快，在单机单卡的 v100 机器上，训练 165 万数据，迭代 26 万步，半个小时左右可以收敛。

1. 先用 wordEmbedding 分词构建词向量
2. Conv 卷积

3. 通过一个 max-pooling 进行下采样。
4. 最后使用 softmax 进行分类

5. 系统安全性和权限设计概述

5.1 权限设计与鉴权说明

认证策略: 服务器采用 SpringSecurity 框架+jwt 令牌以及使用 redis 辅助的认证方式。Jwt 通过签名加密的方式保证不被其他人伪造颁发 jwt。

鉴权: 采用 SpringSecurity 框架对每一个接口限制权限，只有具有相应权限的用户才能调用相应的接口且通过 jwt 存储的用户信息对私人内容进行鉴权。同时 jwt 是无状态的，因此通过 redis 维护黑名单使 jwt 主动失效。

权限设计: 项目分三种角色 用户，管理员，超级管理员，每个角色有相应的权限，具有相应的权限才能访问相应的方法

用户可通过前台 App 操作相应的笔友模块，信箱模块，时空胶囊模块和主页模块。

管理员可通过后台 Web 对用户账号管理，平台数据分析，以及反馈举报处理，以及应用管理。

超级管理员: 超级管理员在管理员的基础可以对管理员账号进行管理。

5.2 安全性设计说明

针对系统可能产生的安全问题进行分析与说明

CSRF(跨站请求伪造)

CSRF 用户 A 请求了正常的网站 C，之后产生了 cookie 保存在客户端，之后用户 A 又浏览的恶意网站 B，网站 B 接收到用户请求后，返回一些攻击性代码，并发出一个请求要求访问第三方站点 C，恶意网站 B 产生恶意的请求去请求网站 C，由于 cookie 是浏览器自动携带的结果 C 只能看到 cookie 是正确的，以为就是用户 A 的行为，从而产生损失。

目前防御 CSRF 有四种方案：

- 验证 HTTP Referer 字段
- 在请求地址中添加 token 并验证
- 在 HTTP 头中自定义属性并验证
- 验证码

服务器采用 jwt 鉴权方式，其内部实现用户通过账号密码登入，登入成功后服务器会颁发 token 给前端，前端存储 token 并且在每次请求手动把 token 加在 http 的请求头上，服

务器对 jwt 解析获取相应信息进行鉴权, 服务器没有采取传统的 session-cookie 机制进行鉴权, 同时 token 为前端在每次请求手动的添加上就不存在跨站请求伪造的问题。

XSS 攻击

跨站脚本攻击 (Cross Site Script 为了区别于 CSS 简称为 XSS) 指的是恶意攻击者往 Web 页面里插入恶意 html 代码, 当用户浏览该页之时, 嵌入其中 Web 里面的 html 代码会被执行, 从而达到恶意用户的特殊目的

服务器不会直接信任用户传输的数据, 会设计一个 Xss 过滤器专门来过滤可能存在的 XSS 攻击。

SQL 注入

sql 注入是指在 sql 语句预编译过程中混入了恶意 sql 代码导致 sql 错误执行, 以不正当的手段多数据库信息进行操作。

后端会对要执行的 sql 语句进行预编译, 保证预编译过程没有用户传来的参数, 且执行的时候只需传参不会再次进行编译, 不会导致恶意 sql 执行。

数据加密

对用户密码等数据采用 SpringSecurity 中 BCryptPasswordEncoder 类通过 SHA-256+随机盐+密钥对密码进行加密。

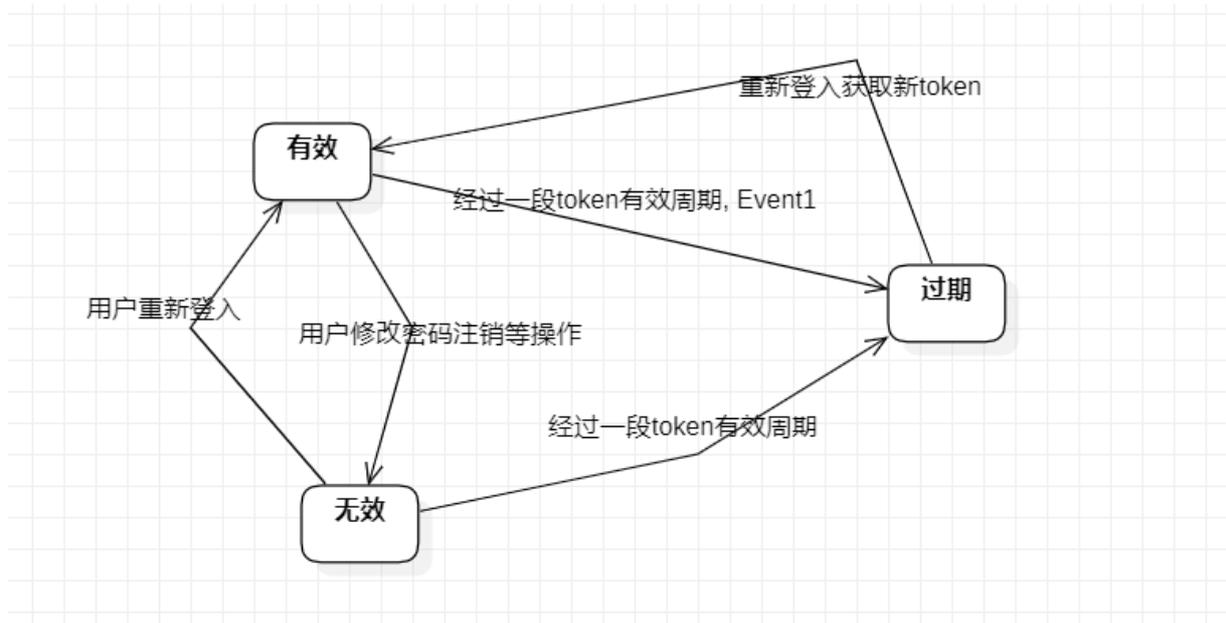
特点: 通过 hash 算法加密每次加密后的密文都不相同且不可逆, 保证了用户密码不能被解密。且通过匹配算法匹配未加密密码与加密后密码是否相同。

JWT 主动失效

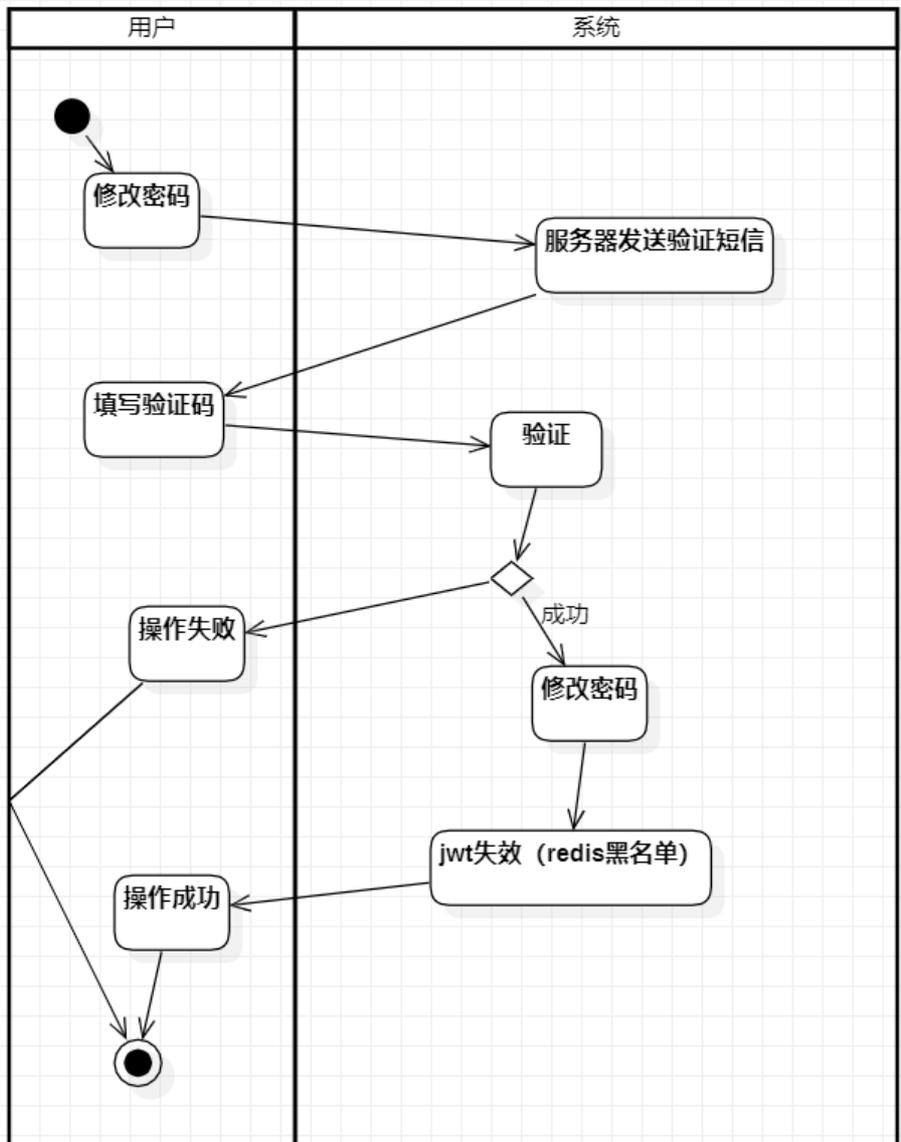
使用 Jwt 有许多好处但缺点也相应而来, 由于 Jwt 的无状态性服务器只知道这个 token 是否合法以及是否过期, 存在一定程度局限性, 因为服务器不知道此时 token 是否有效, 例如如果用户账号密码被盗, 由于 Jwt 的无状态性, 服务器会在一段时间内都认为 jwt 有效, 此时用户修改密码, 并不能使之前以及颁发过的 token 失效, 就存在安全漏洞。

我们在 项目中的方案使用 Jwt 黑名单策略。例如在注销等场景将该 token 加入黑名单。当用户发出请求后, 如果该 token 在黑名单中, 则阻止用户的后续操作, 返回 Invalid token 错误。对于名单的维护使用 redis, token 的过期时间和 redis 中数据的存活时间保持一致。

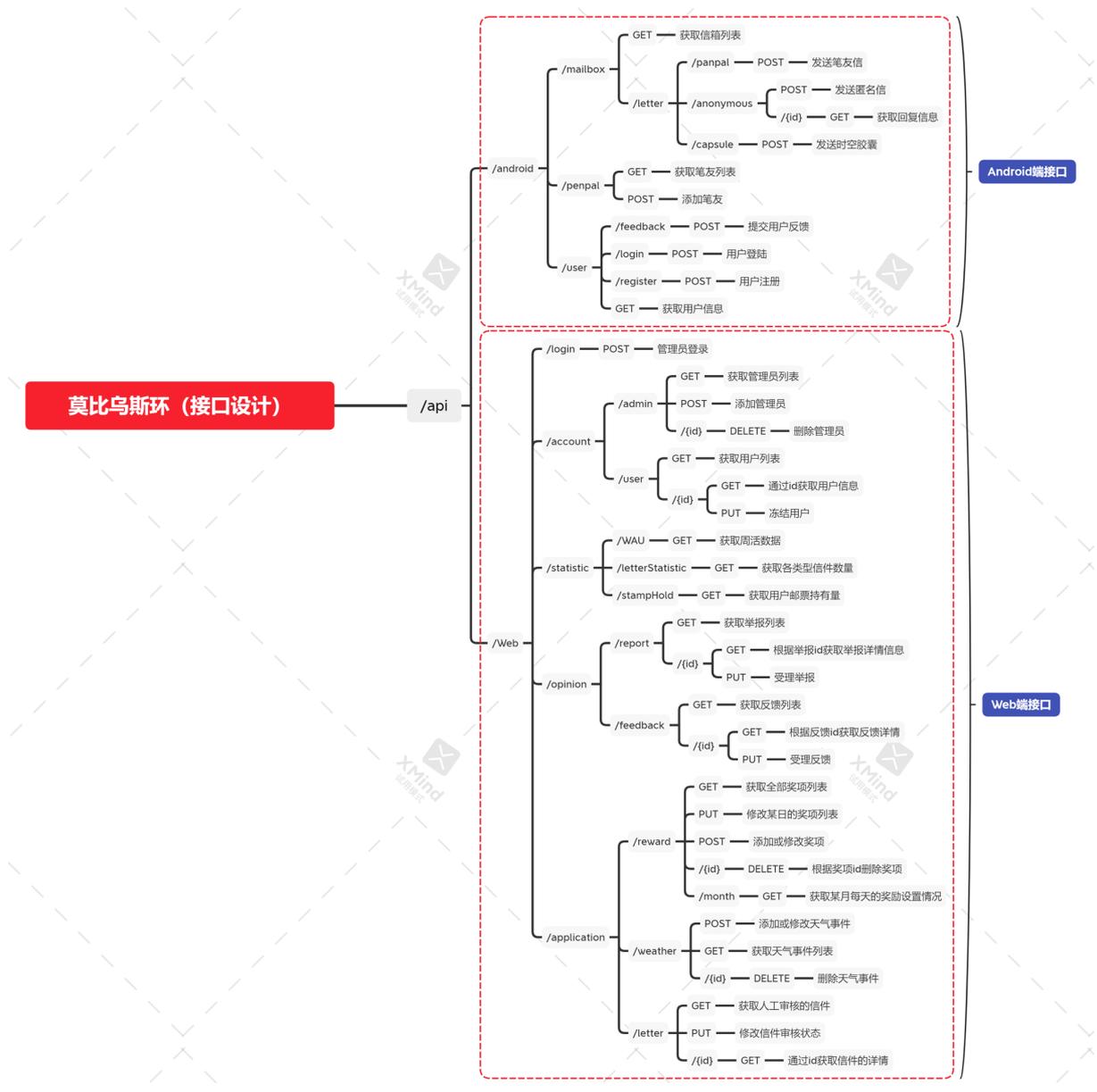
Jwt 的状态图：



模拟用户修改密码 Jwt 失效活动图：



6. 接口设计概述



根据项目接口，本项目的接口分为 Android（前台），Web（后台）部分设计。

6.1 移动端部分接口

6.1.1 信箱管理部分

获取信箱列表



简要描述

- 获取信箱列表

请求URL

- `http://xx.com/api/android/mailbox`

请求方式

- GET

返回示例

```
{
  "error_code": 0,
  "data": {
    "mailbox": [
      {
        "id": "5dc78582-2764-47f8-97f4-7bo62d5703ab",
        "sender_id": "",
        "title": "神秘来信1",
        "like_num": 2,
        "hug_num": 2,
        "is_read": false,
        "arrival_time": "2022-04-15 16:17:08",
        "type": 0
      },
      {
        "id": "4dc78582-2764-47f8-97f4-7bo62d5703ab",
        "title": "神秘来信2",
        "sender_id": "",
        "like_num": 2,
        "hug_num": 2,
        "is_read": false,
        "arrival_time": "2022-04-15 16:17:08",
        "type": 0
      },
      {
        "id": "4cc78582-2764-47f8-97f4-7bo62d5703ab",
        "title": "好友来信1",
        "sender_id": "2764-47f8-97f4",
        "like_num": ,
        "hug_num": ,
        "is_read": false,
        "arrival_time": "2022-04-15 16:17:08",
        "type": 1
      },
      {
        "id": "4ccc8582-2764-47f8-97f4-7bo62d5703ab",
        "title": "时间胶囊1",
        "sender_id": "2764-47f8-97f4",
        "like_num": ,
        "hug_num": ,
        "is_read": false,
        "arrival_time": "2022-04-15 16:17:08",
        "type": 2
      }
    ]
  }
}
```

返回参数说明

参数名	类型	说明
mailbox	array[letter]	信件列表

letter

参数名	类型	说明
id	string	信件ID
title	string	信件标题
sender_id	string	发送用户ID
like_num	int	喜欢人数
hug_num	int	抱抱人数
is_read	bool	是否阅读
arrival_time	Date	送达时间
type	int	信件类型 (匿名信: 0, 笔友信: 1, 时间胶囊: 2)

备注

- [更多返回错误代码请看首页的错误代码描述](#)

获取回复信息



简要描述

- 获取回复信息

请求URL

- `http://xx.com/api/android/letter/anonymous/{id}`

请求方式

- GET

返回示例

```
{
  "code": 200,
  "msg": "获取成功",
  "data": {
    "reply": {...}
  }
}
```

返回参数说明

参数名	类型	说明
reply	array[anon]	回复列表

备注

- 更多返回错误代码请看首页的错误代码描述

简要描述

- 获取回复信息接口

发送匿名信



简要描述

- 发送匿名信接口

请求URL

- `http://xx.com/api/android/mailbox/letter/anonymous`

请求方式

- POST

参数

参数名	必选	类型	说明
user_id	是	string	用户唯一标识码
anon	是	object	匿名信

返回示例

```
{
  "msg": "发送成功",
  "code": 200,
  "data": null
}
```

返回参数说明

无

备注

- 更多返回错误代码请看首页的错误代码描述

发送笔友信



简要描述

- 发送笔友信接口

请求URL

- `http://xx.com/api/android/mailbox/letter/penpal`

请求方式

- POST

参数

参数名	必选	类型	说明
user_id	是	string	用户唯一标识码
pen	是	object	笔友信

返回示例

```
{
  "msg": "发送成功",
  "code": 200,
  "data": null
}
```

返回参数说明

无

备注

- 更多返回错误代码请看首页的错误代码描述

简要描述

- 发送时空胶囊接口

请求URL

- `http://xx.com/api/android/mailbox/letter/capsule`

请求方式

- POST

参数

参数名	必选	类型	说明
username	是	string	用户名
capsule	是	object	匿名信

返回示例

```
{
  "msg": "发送成功",
  "code": 200,
  "data": null
}
```

返回参数说明

无

备注

- 更多返回错误代码请看首页的错误代码描述

6.1.2 笔友管理部分

简要描述

- [获取笔友列表接口](#)

请求URL

- `http://xx.com/api/android/penpal`

请求方式

- GET

返回示例

```
{
  "code": 200,
  "data": {
    "penlist": [
      {
        "id": "5a57-4c35-bfe1",
        "nickname": "大聪明",
        "mood": "开心",
        "create_time": 2022-04-15 19:26:00
      },
      {
        "id": "5a57-4c35-bfe2",
        "nickname": "大聪明1",
        "mood": "开心",
        "create_time": 2022-04-15 19:26:00
      }
    ]
  }
}
```

返回参数说明

参数名	类型	说明
penlist	array[pen]	笔友列表

pen

参数名	类型	说明
id	string	笔友id
nickname	string	笔友昵称
mood	string	笔友心情
create_time	date	成为笔友的时间

备注

- [更多返回错误代码请看首页的错误代码描述](#)

添加笔友



简要描述

- 添加笔友接口

请求URL

- `http://xx.com/api/android/penpal`

请求方式

- POST

返回示例

```
{
  "code": 200,
  "msg": "添加成功"
  "data": null
}
```

返回参数说明

无

备注

- 更多返回错误代码请看首页的错误代码描述

6.1.3 用户管理部分

获取用户信息



简要描述

- 获取用户信息接口

请求URL

- `http://xx.com/api/android/user`

请求方式

- GET

返回示例

```
{
  "code": 200,
  "msg": "获取信息成功",
  "data": {
    "user_id": "e92208d5-5a57-4c35-bfe1-85832c767adb",
    "phone": "13553113513",
    "nickname": "大聪明",
    "mood": "开心",
    "grow_value": 2,
    "status": "正常"
  }
}
```

返回参数说明

参数名	类型	说明
user_id	string	用户唯一身份标识码
phone	string	用户手机号
nickname	string	用户昵称
mood	string	用户心情值
grow_value	int	用户成长值
status	string	用户状态

备注

- 更多返回错误代码请看首页的错误代码描述

提交用户反馈



简要描述

- 用户注册接口

请求URL

- `http://xx.com/api/android/user/feedback`

请求方式

- POST

参数

参数名	必选	类型	说明
user_id	是	string	用户唯一标识码
feedback	是	object	反馈信息

返回示例

```
{
  "code": 200,
  "msg": "反馈成功",
  "data": null
}
```

返回参数说明

无

备注

- 更多返回错误代码请看首页的错误代码描述

用户登陆



简要描述

- 用户登陆接口

请求URL

- `http://xx.com/api/android/user/login`

请求方式

- POST

参数

参数名	必选	类型	说明
phone	是	string	用户手机号
password	是	string	用户密码

返回示例

```
{
  "code": 200,
  "msg": "登陆成功",
  "data": {
    "token": "eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJhZG1pb1IsImNyZWl0eSI6ImV4MjZwQjE2NDk4NjA3MjY3ODMsImV4cCI6I6MTY1MDQ2NTUyNn"
  }
}
```

返回参数说明

参数名	类型	说明
token	string	Token

备注

- 更多返回错误代码请看首页的错误代码描述

用户注册



简要描述

- 用户注册接口

请求URL

- `http://xx.com/api/android/user/register`

请求方式

- POST

参数

参数名	必选	类型	说明
phone	是	string	用户名
password	是	string	密码
nickname	是	string	昵称

返回示例

```
{
  "code": 200,
  "msg": "注册成功",
  "data": null
}
```

返回参数说明

无

备注

- 更多返回错误代码请看首页的错误代码描述

6.2 部分后台 web 端接口

获取天气事件列表



简要描述

- 获取天气事件列表

请求URL

- `/api/web/application/weather`

请求方式

- GET

参数

参数名	必选	类型	说明
page	是	int	页码
pageSize	是	int	页面大小

返回示例

```
{
  "msg": "",
  "code": 200,
  "data": {
    "weatherList": [
      {
        "id": 0,
        "incident": "晴天",
        "delay_time": 20h,
        "ratio": 50,
        "description": "今天是晴天，信件马上会飞奔到你身边啦"
      },
      {
        "id": 1,
        "incident": "阴雨天",
        "delay_time": 4h,
        "ratio": 50,
        "description": "今天有点小雨，信鸽在努力飞回来啦"
      }
    ]
  }
}
```

返回参数说明

参数名	类型	说明
id	int	天气事件唯一标识
incident	string	事件名称
delay_time	string	延迟时间
ratio	int	事件发生概率
description	string	事件具体描述

获取所有反馈



简要描述

- 获取所有反馈接口

请求URL

- `/api/web/opinion/feedback/`

请求方式

- GET

参数

参数名	必选	类型	说明
desc	是	boolean	是否按反馈日期降序

返回示例

```
{
  "msg": "",
  "code": 200,
  "data": {
    "total": 10,
    "data": [
      {
        "id": 1,
        "user_id": 2,
        "user_nickname": "张三",
        "feedback_title": "",
        "create_time": "",
        "deal_time": "",
        "is_deal": 0,
      },
    ]
  }
}
```

返回参数说明

参数名	类型	说明
id	int	反馈id
user_id	int	反馈者id
user_nickname	string	反馈者昵称
feedback_title	string	反馈标题
create_time	string	反馈时间
deal_time	string	受理时间
is_deal	int	受理状态

备注

- 更多返回错误代码请看首页的错误代码描述

获取信件详情



简要描述

- 通过id获取信件的详情

请求URL

- `/api/web/application/letter/{id}`

请求方式

- GET

参数

参数名	必选	类型	说明
id	是	int	信件ID

返回示例

```
{
  "msg": "",
  "code": 200,
  "data": {
    "id": 002,
    "sender_id": 001,
    "title": "今天吃饭了",
    "content": "今天吃了两碗饭",
    "state": 0,
    "update_time": "2022-05-05 02:03:00"
  }
}
```

返回参数说明

参数名	类型	说明
id	int	信件id
sender_id	int	发送者用户id
title	string	信件标题
content	string	信件内容
state	int	信件状态, 0未审核, 1已通过, -1未通过
update_time	date	信件更改时间

