

需求与设计文档

组长：张后斌 2018010858

组员：吴雨娴 2019013303，黄碧婷 2019010481

选题：校园论坛应用

一、项目背景

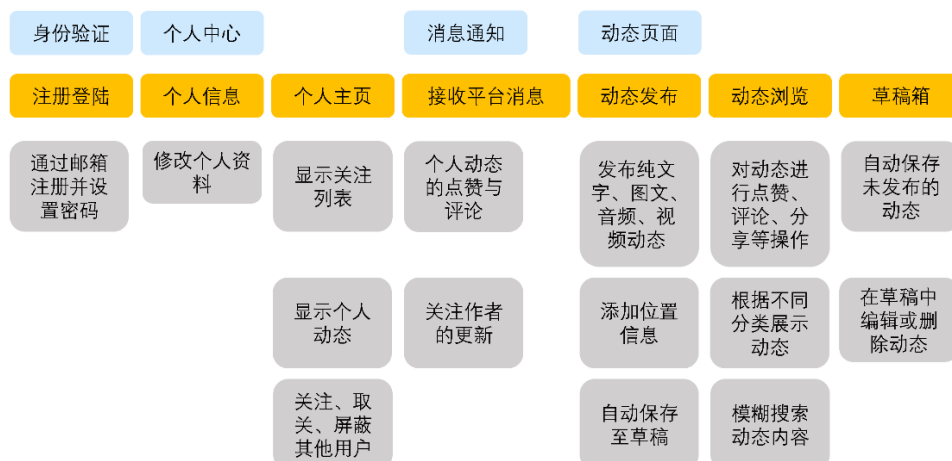
当今社会是一个信息化丰富的社会，虽然树洞已经离我们远去了，但是树洞的出现依旧展现出了广大学生对一个校园论坛的热烈渴求。同在一个园子下生活，我们往往可能会对其他人的经历更有共鸣，发帖者也会在这种共鸣下可能更好的解决自己的问题。很多时候，有些问题是只有有着相同的经历才能感受更加深刻，所以这样一个可供广大学生使用的校园论坛是迫在眉睫的。即便有了微博、知乎等论坛，但是很多时候思维方式已经受教育程度等等因素，导致我们可能无法从中得到共鸣，或者说在对某些问题的看法上，出现很多认知上的偏差。所以，综合来说，创建一个独属于我们的校园论坛是一件十分有意义的事情，通过这样一种模式，或许我们可能更加加深彼此之间的联系，更有彼此都在同一个园子里的身份认同感。

二、功能需求

1. 介绍系统角色

本次我们小组的选题是建立一个校园论坛，理论上我们需要管理员和用户两种身份的角色；但是由于在评分细则中没有对管理员相关功能的描述，因此目前我们暂不考虑管理员的身份，如果后续开发过程中仍旧有时间，我们会考虑补充管理员这样一个角色。所以目前我们的系统角色只有用户这样一种身份。由于我们是一个校园论坛的应用，因此主要的使用群体就是学生，学生可以在这个论坛中发布自己想要创作的内容，也可以看到其他用户发布的各种内容，本质是提供一个平台，希望学生能够通过这样一种媒介分享自己的生活，让校园生活变得更加美好。

2. 用户故事



3. 功能点清单

(1) 注册与登录:

1. 通过邮箱注册账号, 设置密码
2. 修改用户名, 头像, 简介, 密码
3. 对用户进行关注和取关, 显示关注列表, 在动态页面的用户名后边显示“已关注”
4. 对用户进行屏蔽, 并不显示其动态

(2) 二级菜单:

1. 个人主页显示本人发布的所有动态
2. 从作品、评论等界面进入他人的个人主页
3. 草稿页面保存未发布的作品草稿
4. 进入一篇草稿的编辑界面继续进行编辑或者删除

(3) 发布:

1. 发布纯文字动态
2. 发布图文混合(包括调用相机进行拍照)
3. 发布音频动态(包括调用麦克风进行录音)
4. 发布视频动态(包括调用相机进行录像)
5. 发布动态时添加位置信息(调用手机定位)
6. 编辑过程中自动保存草稿, 发布完成后自动从草稿箱删除

(4) 查看:

1. 选择根据时间顺序或者点赞数量进行动态展示
2. 选择根据所有人或者已关注的人进行动态展示
3. 对动态进行点赞(展示所有的点赞人信息)
4. 对动态进行评论(展示所有的回复信息)
5. 对动态进行分享(分享到任何其他可展示内容的 APP 中)
6. 可以取消点赞, 删除评论

(5) 通知:

1. 接收到来自平台的通知信息, 包括自己作品的点赞与评论, 以及关注作者的更新
2. 用户从个人中心界面进入通知消息列表

(6) 搜索:

1. 根据动态名称, 动态内容, 用户名称, 作品类型(文图音视, 在动态发布时选择)四种方式搜索内容
2. 搜索时实现模糊搜索

三、非功能需求

(1) 性能需求:

- 应用应当能够在以下设备上正常运行:
 - Android 虚拟机 Pixel 3A, 屏幕分辨率 2220*1080, 系统版本: Android 11.0
 - Redmi Note 9 Pro, 屏幕分辨率 2400*1080, 系统版本: MIUI 12.0.11 (Android 10)

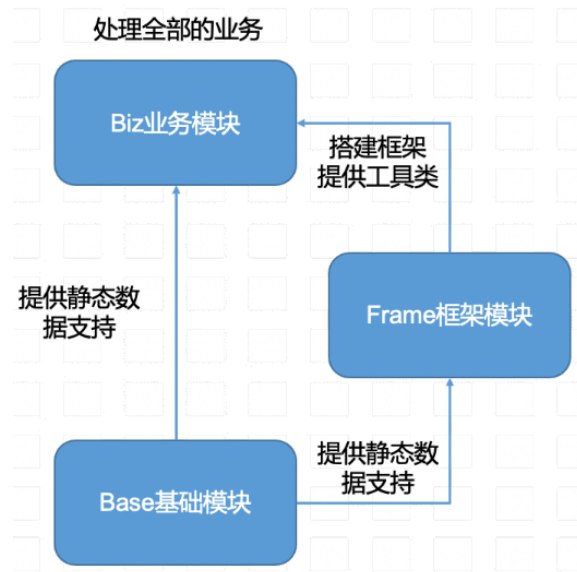
- Redmi Note 8, 屏幕分辨率 2340*1080, 系统版本: MIUI 12.0.11 (Android 10)
 - 应用应支持多客户端同时操作。
 - 在正常网络环境下, 应用的响应速度应该控制在 5 秒以内。
 - 应用应具有良好的用户体验, 充分体现校园论坛的交互特点。
- (2) 质量属性:
- 鲁棒性: 应用应具有鲁棒性, 在出现问题时不会使应用崩溃。
 - 可扩展性: 应用应该具有良好的架构设计, 可扩展性强。
 - 安全性: 应用应该保证安全可靠, 不随意泄漏信息。
- (3) 前端非功能需求:
- 应用应具有简洁、大方的外观。
 - 应用在运行过程中不应出现组件遮挡、显示不全的情况
 - 所交付的应用源代码应格式规范、风格统一, 易于阅读和维护。
- (4) 后端非功能需求:
- 后端应符合业务逻辑规范, 满足不可违背和推荐性两大块的要求。
 - 后端代码整体组织结构需要按照 Base 基础模块、Frame 框架模块和 Biz 业务模块的三大模块原则来进行, 且每个类应该按照相关性放在对应的包中。
- (5) 整体代码规范:
- 代码应该书写工整, 利用换行符、空格符保证代码的可读性, 并避免出现冗长的单行代码。
 - 每个类及类中的字段和方法应有对应的注释说明, 方法中如果代码量较多, 也应该分块予以注释说明。
 - 对于大括号的使用、函数的注释风格等应该进行统一。

四、初步技术设计

1. 后端技术设计

(1) 后端框架: 后端采用课程提供的代码框架, 结合 Java 服务器和 SpringBoot 框架进行开发;

(2) 业务处理: 后端通过 Base 模块、Frame 模块、Biz 模块实现业务的处理, 其处理逻辑如下:



(3) 数据存储：通过 Redis 和 Mongoddb 数据库实现数据的存储和增删改查等功能。

2. 前端技术设计

前端主要采用 Java 进行编写，通过 websocket 和 http 与后端进行数据交互，主要会通过 Android studio 平台进行开发和测试。