微信小程序文档

Hi洞 介绍

1 成员介绍与分工

姓名	学号	分工
陆昱宽（组长）	191820133	负责小程序逻辑层开发和前后端的联动
陈家志	191250009	负责小程序后端的实现和服务器的部署
顾雯文	191250038	负责小程序的展示和文档完善工作
贺伟	191250044	负责小程序前端界面展示与数据绑定的实现
李若璇	191250072	负责小程序页面设计与美化和文档完善工作

2 项目地址

Github网址： https://github.com/LYK-love/wxProject

开发分支： dev分支

3 项目介绍

3.1 项目背景

“树洞”类小程序在当下愈发衰弱，不仅仅因为树洞的匿名机制让情绪无端放大引发很多无谓的争执，而且，在小程序功能逐渐强大的今天，能够替代“树洞”的小程序越来越多，它们在实现树洞功能的同时有各种新奇的功能，但同时他们的诞生也伴随着无穷无尽的争吵。我们希望有一款将社交性降到最低的树洞，并通过一些其他机制增加树洞的可用性。

3.2 项目简介

"Hi 洞" 是一个简单的匿名树洞。在这里，用户能做的只有写信和读信回信。为了尽可能的降低社交性，我们取消了回复提醒和历史查询的功能，使每一封被用户寄出的信都像是进入了黑洞中，只有有缘才能完成和他的再次邂逅，这也是我们这款小程序的名称“Hi洞”（黑洞）的由来。同时，我们当然也不会让所有的信件真正的随机，无序的随机带来的只能是混乱和差的用户体验，所以我们也设计了一套机制让用户能在保证一定随机性的情况下看到自己从前的帖子。

基于以上的种种机制，这个树洞主要依赖活跃的用户而存在。所以怎么去保证活跃的用活也是我们需要解决的问题，就此我们通过增加24小时无人发帖就自毁的机制，在吸引更多猎奇用户的同时，可以保证用户每日登录使用，增加用户粘性。而每天发帖就能阻止小程序的自毁，也为群众带来正向的引导。

3.2.1 雏形

项目实现了基本树洞所具有的功能，包括但不限于发帖，读帖和回帖。其基本功能如下所示：

1. 使用微信账号登录，进入主页面，在第一次登陆时会出现新手导航，而在之后每次使用中可以重新阅读新手导航。
2. 在主页面显示看见倒计，显示的时间是小程序的剩余生存时间。
3. 在个人中心可以看到发送的帖子数量，以及为这个小程序的贡献的生存时长。
4. 能够发送和阅读帖子。阅读帖子时将看见一个帖子和其他人的回复，可以选择回复帖子或者点击阅读下一个帖子。

3.2.2 展望

通过未来我们团队共同的努力希望能够实现以下功能：

1. 能够对于用户发送的内容进行筛选，通过分词机制将言辞激烈、措辞不当、含有违规词汇的内容屏蔽，营造文明积极的社区环境。
2. 提高帖子的时效性，在发帖量足够的时候只在显示最近一个月的帖子。在某些特殊的日子增加相关性更强的帖子的数量。
3. 通过发帖和读帖的大量数据，为用户打上标签，例如：“积极”“阳光”“消极”“自卑”等，为积极且希望帮助其他用户的人推荐多两成的消极和求助内容，为消极的用户提供更多积极的温暖的内容。
4. 并且最终希望能够与心理健康相关协会联系，为在标签中比较抑郁的人提供言语帮助甚至一些治疗。

3.3 目标用户

在每年都要发布的《国民健康报告》中可以发现，越来越多的人正处于一种心理亚健康的状态，在生活中的种种负面情绪不知道要怎么样正确的去发泄和缓解，我们的目标用户正是这么一群人，我们希望能够用这种“低社交性”的树洞来帮助他们解决心理的问题。

同时我们希望通过24小时自毁的设定能够吸引一些猎奇的用户，这部分用户的心理更加健康积极，可以给予那些处于烦恼中的人一些鼓励。

3.4 应用场景

1. 在生活无趣的时候。
2. 在心情不好，需要吐槽自己的遭遇时候。
3. 在心情舒畅，想要分享美好事情的时候。

3.5 痛点与解决方案

1. 小程序并不存在查看自己曾经发送的帖子的功能，为解决部分用户希望看到回帖的需求，在后端提高刷到带有新的回复的自己的帖子的概率。
2. 匿名发送的机制可能会导致一些激烈言辞的帖子，通过训练模型，提高对分词的认知，将负能量的帖子永久屏蔽，减少消极情绪的传播。

3.6 竞品分析

传统的树洞应用，是基于匿名的社交场所，通过发送帖子记录日常、宣泄情绪、回复他人。很多人希望得到别人的回应，但是也会获得一些不好的反馈。有些人不希望自己的吐槽被注意到，但是可能被当事人看见造成社会性死亡。树洞的匿名性质会将用户的情绪放大，比起实名制的论坛更容易发生争执，社区环境会变得乌烟瘴气。

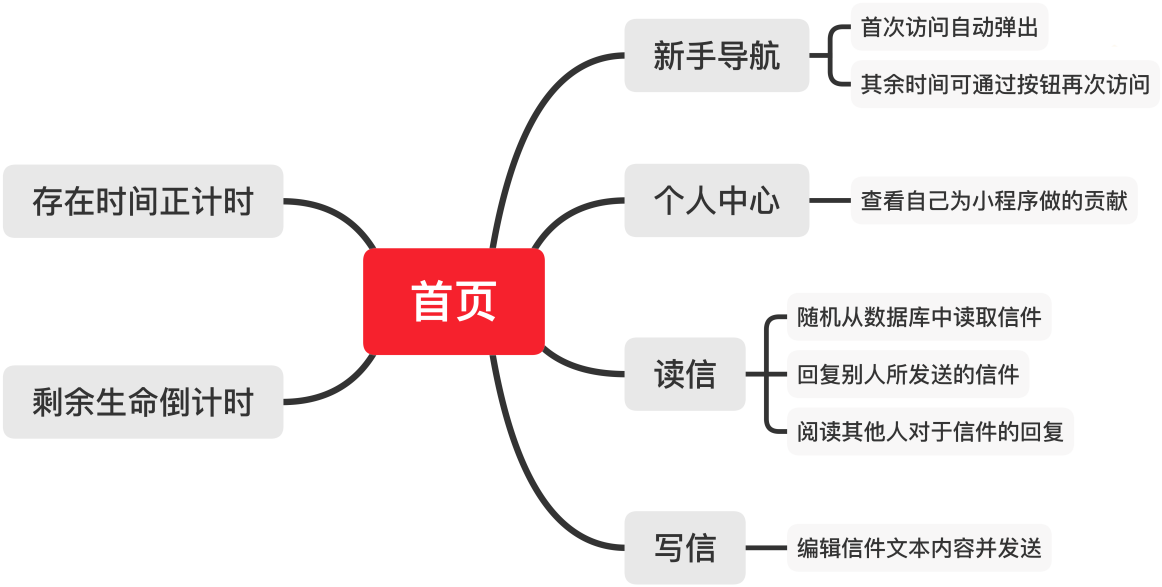
所以我们针对之前的树洞提出了一套新的机制，希望能够解决之前遇到的一系列问题。

4 产品设计

4.1 产品功能

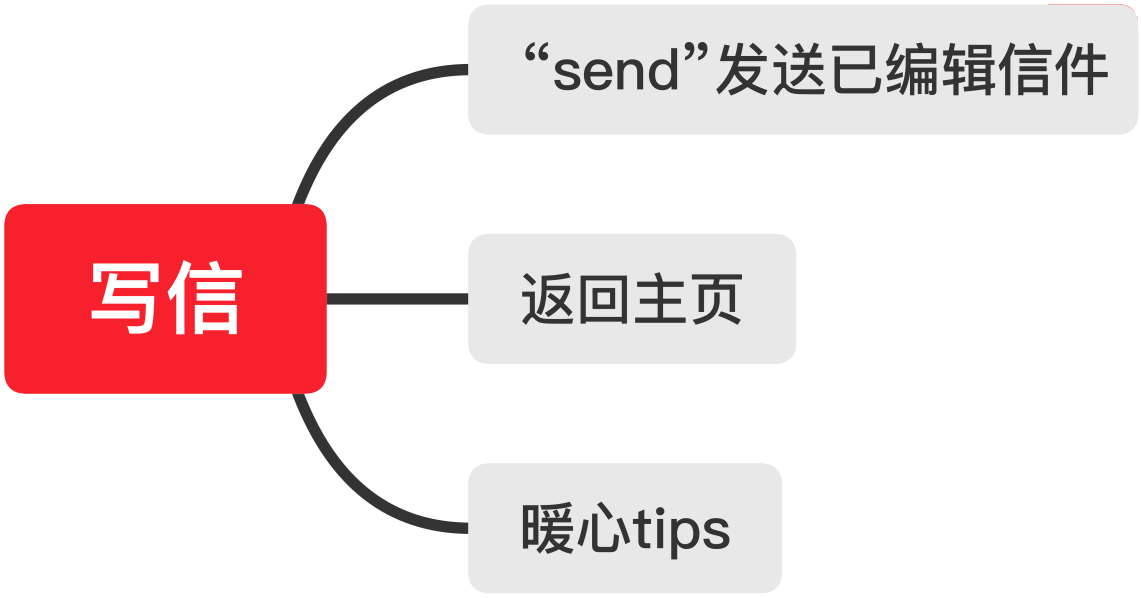
4.1.1 首页

首页会实时展示小程序生命的变化倒计时，同时会显示小程序从运行开始已经存活了多久。而在主页，你可以阅读新手导航、前往个人中心、还可以选择去写一份信或者随机去读一封已有的信件。



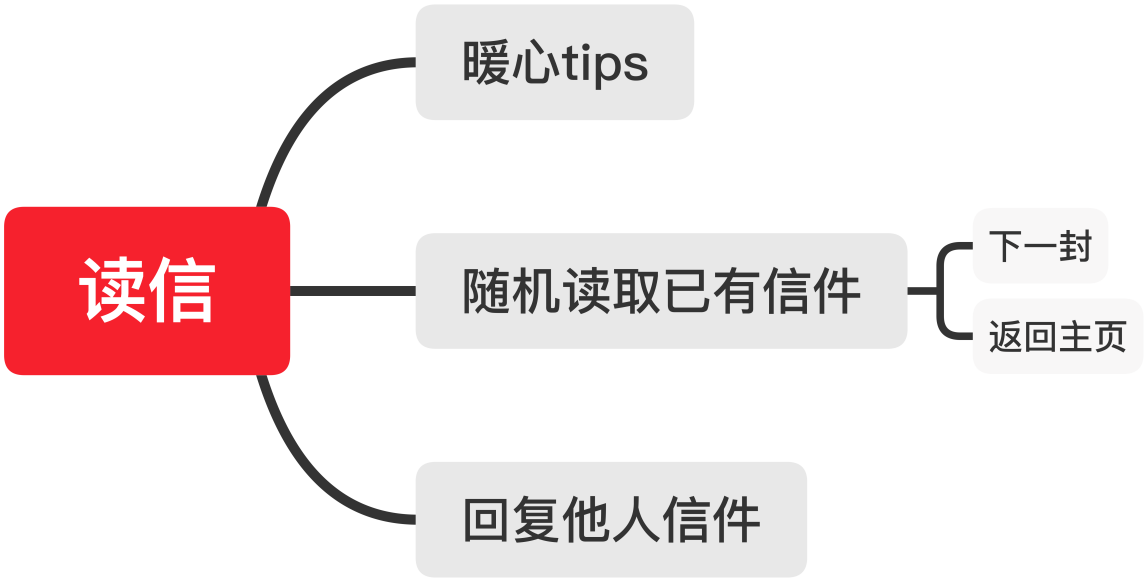
4.1.2 写信

写信界面允许用户编辑所想要表达的内容，通过“send”按钮进行发送；或者可以选择放弃编辑，这时候会弹出二次确认放弃编辑按钮以防止用户误触导致所编辑的内容丢失。在写信界面还可以进入暖心导航页面，以关注用户的心理健康问题。



4.1.3 读信

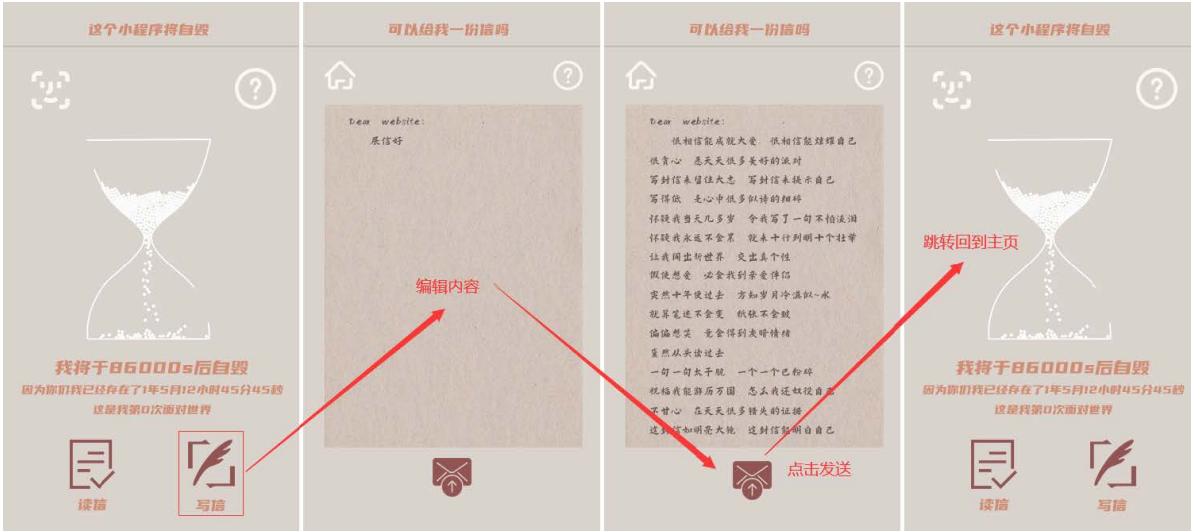
在读信界面用户可以查看从数据库的随机提取的已有信件和其回信，也可以选择编辑是否回复别人的信，并可以同一界面编辑内容进行发送。在读信界面也可以进入暖心导航页面，以关注用户的心理健康问题。



4.2 交互设计

4.2.1 写信流程

用户可以通过小程序写信以表达自己的情绪

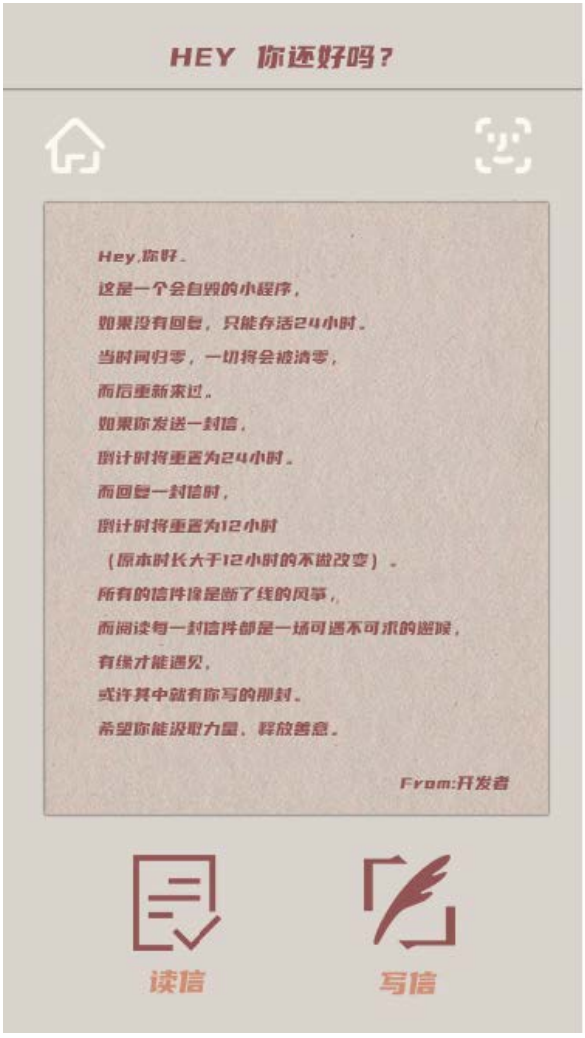


4.2.2 读信流程

用户可以通过小程序阅读信件，同时回复安慰他人

4.2.3 其余页面展示

新手导航（左图） 个人中心（右图）



4.3 运营方案

4.3.1 引流策略

由于小程序有特殊的24小时自毁策略，同时对用户的质量要求很高。所以在平台建立之初，获取具有很强用户粘性的首批用户是重中之重，为此构建了如下引流策略：

1. 激发用户兴趣。“24小时”自毁的机制能够吸引一些猎奇的用户，他们不在树洞类型原本的覆盖群体里，他们对于小程序的存活时长更加感兴趣。同时，该小程序完成基本任务的使用时长非常短，能够在任何碎片化时间里点击诗意，不存在时长的要求。
2. 学校空间推广。学生群体普遍对新奇的事物带有极强的兴趣，通过在大学生的QQ空间和微信朋友圈的推广，能够覆盖相当一群有闲且找乐子的用户使用。

4.3.2 存留策略

在吸引用户后，针对最大化的满足用户需求，增加用户粘性，提高用户在小程序的停留时间，构建如下存留策略：

1. 优化交互策略，提高用户体验。通过简洁清晰明了的交互界面，给用户带来良好的用户体验。同时，界面的操作简明，让功能更加突出，让用户容易上手。

2. 优化读信系统。概率机制。通过用户画像分析用户的性格和爱好，再据此提供更加针对性的信件的呈现，而不是完全随机。同时24小时的机制能够让用户养成每天点击进入小程序浏览的习惯，增加用户的黏性。

5 关键技术

5.1 小程序端

1. 采用NPM对使用的第三方组件库进行包管理
2. 小程序开发主要采用了Vant Weapp第三方组件库进行构建

5.2 服务端

5.2.1 开发框架

使用成熟的Spring Boot框架搭建服务 + Mybatis框架管理数据库，便于后续服务拓展

5.2.2 部署环境

由于项目核心特性是会重置应用，为了避免过长的部署时间导致用户长时间等待，通过服务端直接执行由maven打包的jar包进行部署

6 系统设计

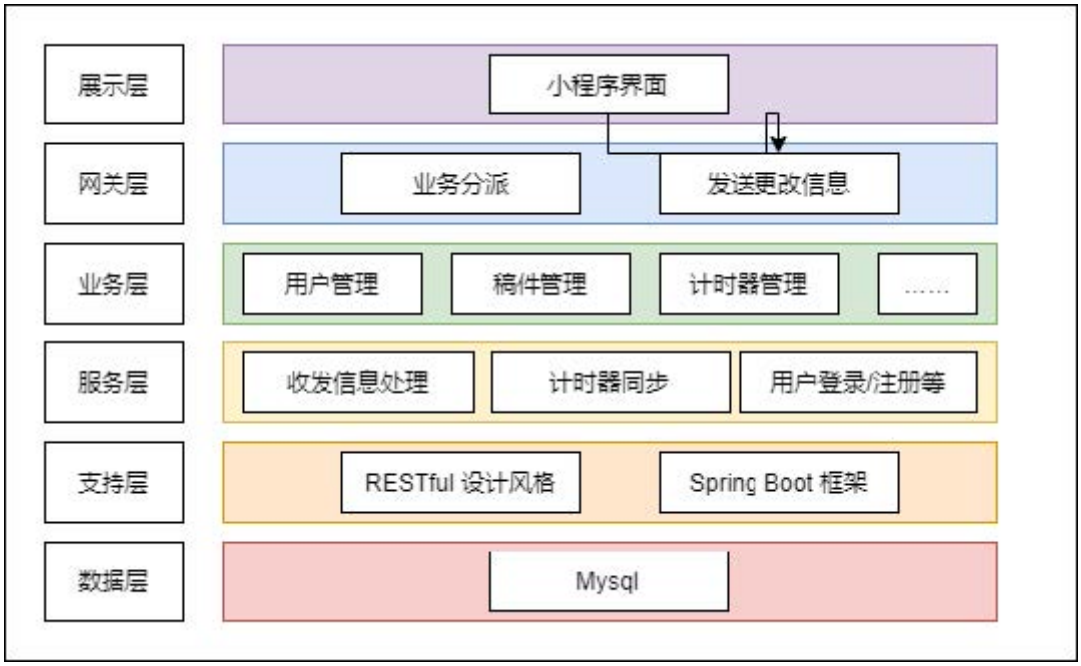
6.1 总结设计

6.1.1 架构设计

本系统的架构采用分层结构，总共划分为六层，其中：

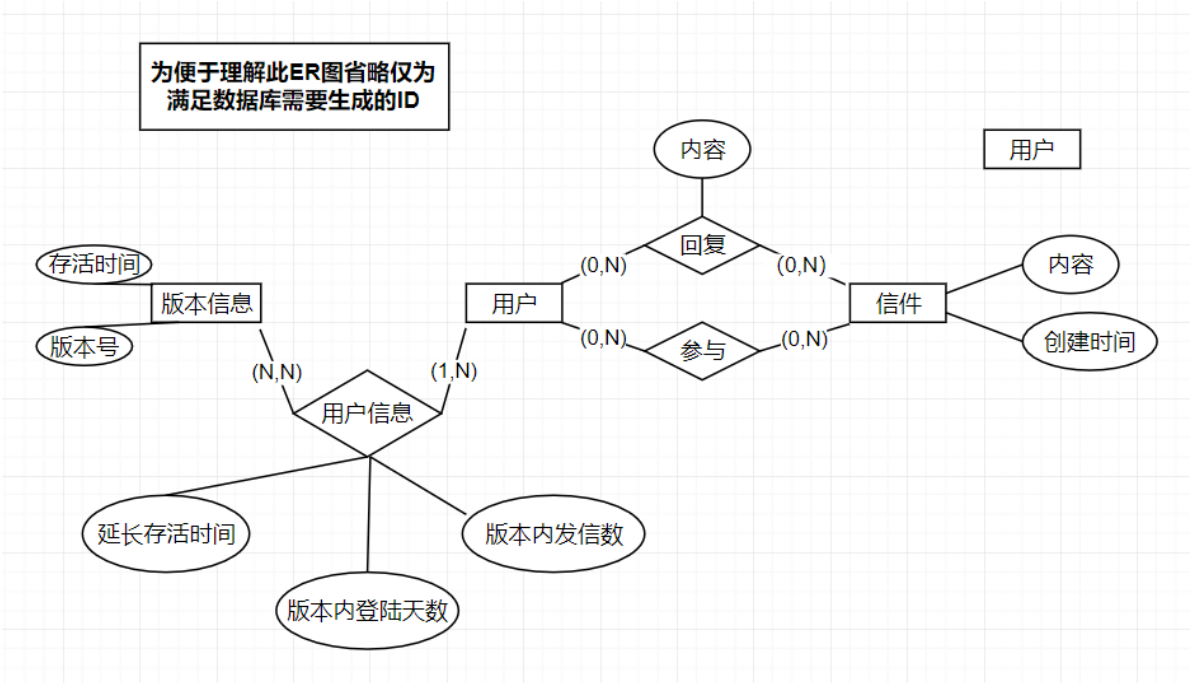
1. 数据层为系统提供数据支持，采用 MySQL 作为主要的数据库进行数据存储
2. 支撑层为系统的设计开发提供支持，服务接口采用 RESTful 设计风格、后端采用 Spring Boot 框架快速构建 Java 应用
3. 服务层包括了系统运行过程中需要的一些公共基础服务，如收发信处理，计时器同步，用户登录/注册等服务；
4. 业务层包含了系统主要的业务服务，包括用户管理、信件管理、计时器管理、概况管理

5. 逻辑层包含了系统的判断逻辑,负责对显示层传入的消息进行业务分派，并向视图层发送更改信息。
6. 显示层负责页面显示,接受用户输入,并向逻辑层传递消息. 并且通过双向绑定,实现实时渲染.



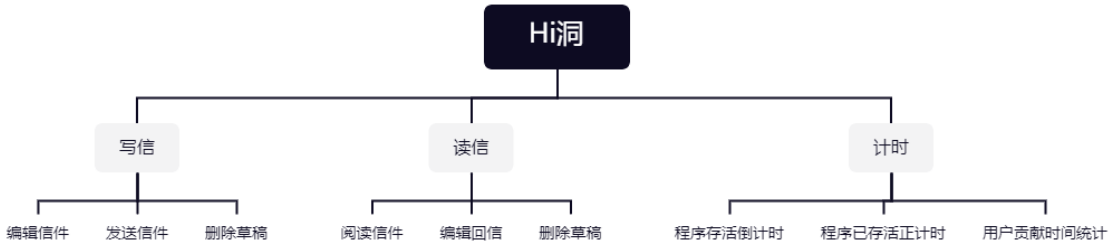
6.1.2 数据库设计

本应用的 E – R 图仅具有三个实体，分别是用户、信件、版本信息
具体E – R 图如下所示.



6.1.3 功能结构设计

为确保系统的可拓展性，对系统的功能结构设计应该采用模块划分的形式实现。小程序的功能结构图如下所示：

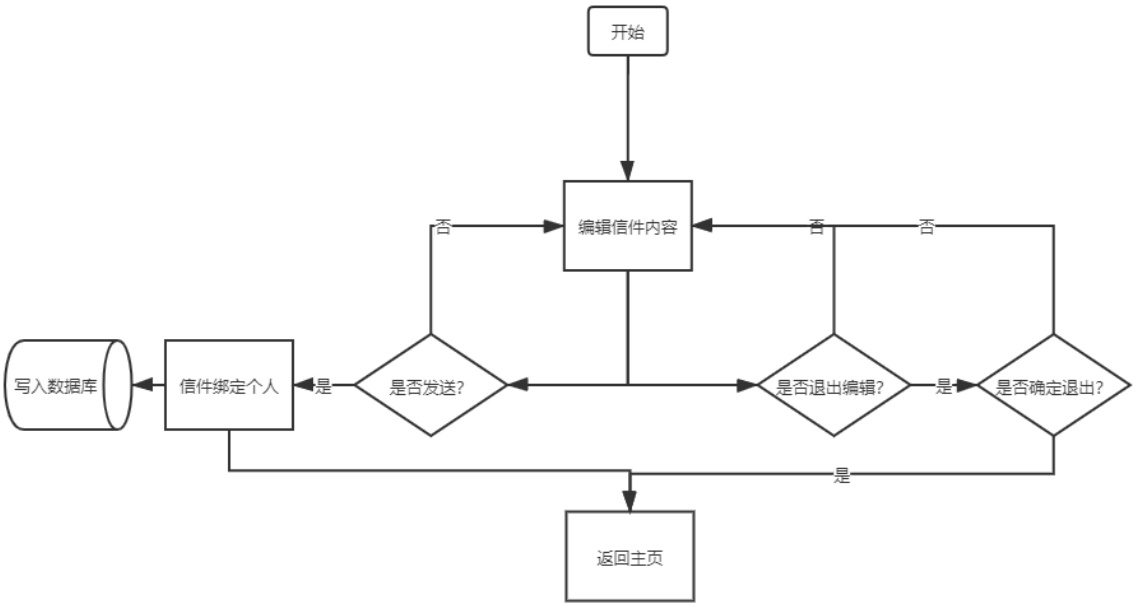


6.2 功能模块设计

根据 4.1.3 小节的设计，将“Hi洞”小程序分成写信、读信、倒计三四个模块，本节将对上述模块的设计与实现进行更深入的阐述说明。

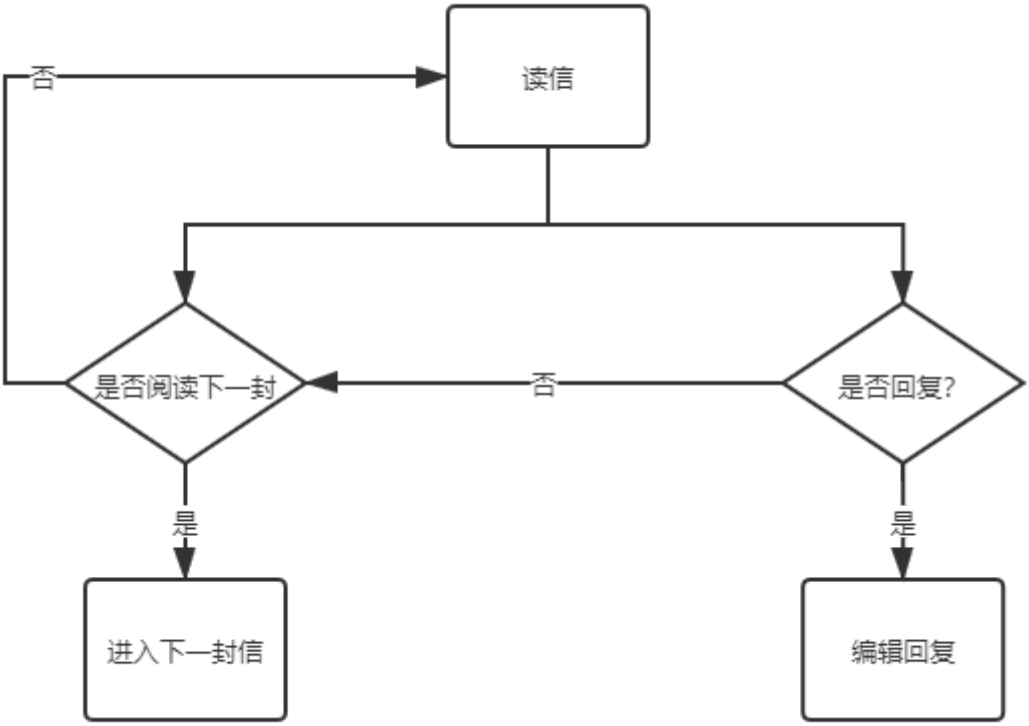
6.2.1 写信模块

写信模块是小程序能够运行起来的基础，其中包括编辑信件，发送信件，删除草稿四个主要功能。用户可以自由编辑页面，在编辑完成后发送信件或者在中途退出。



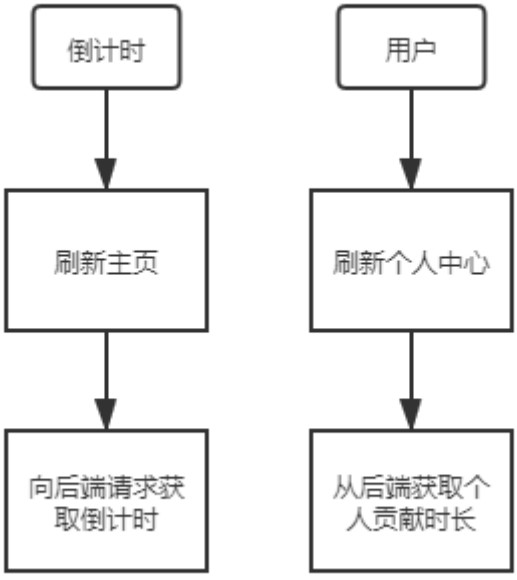
6.2.2 读信模块

读信模块同为小程序能够运行起来的基础，其中包括阅读信件，发送回信，删除草稿四个主要功能。用户可以自由编辑页面，在编辑完成后发送信件或者在中途退出。



6.2.3 倒计时模块

倒计时模块会为用户提供小程序剩余存活时间倒计时、已存活时间计时、个人为小程序贡献时长，分前后端异步显示，只有在刷新页面时候才会再次请求后端返回数据，其余时候前端自我倒计时。



7 微信小程序开发总结

7.1 学习并使用基本的微信开发端开发技术

每一个页面都是四个文件一起渲染的结果，在开发过程中，合理 应用了微信官方文档提供的组件与 API 接口，加快了开发效率

7.2 了解并学习了微信小程序前端界面展示与数据绑定的实现

在微信小程序的的展示层与数据层，二者具有一些微妙的关系，特别是数据同前端界面的渲染，有了进一步的认识。

7.3 使用Vant WeappUI框架进行开发，方便小程序的开发

7.4 了解并学习手机屏幕尺寸以及微信小程序适配问题

在使用微信开发者工具开发好前端界面后，适配到手机端时，出现了严重的长宽失衡、以及组件的显示问题。通过查阅资料，对问题出现的原因进行排查，深度定位。发现手机机型的多款式、px单位的缺点、cover-view与cover-image预览显示以及原生组件嵌套问题。

在深度了解小程序适配手机端的方法后，通过调用微信官方文档提供的 API 接口获取手机的屏幕的宽度和高度，通过比例计算进行适配调整，精确的计算了内外边距、宽度以及更多的使用rpx和百分率替代px，完美的解决了多机型的适配问题。

同时，对于微信小程序中原生组件"view"、"cover-view"与"image"、"cover-image"的展示效果和逻辑有了较多认识，充分的了解到了文档中版本说明的重要性。

7.5 学习并掌握了微信小程序的逻辑层开发

对微信小程序的MINA框架, 双向绑定, 自定义组件等有了新的认识。

7.6 学习了前后端联调的流程

为了减少前后端耦合,降低工作量, 我们采用了rest风格的前后端交互方案, 这是微信小程序自带的方案, 而在拥有postman等工具的情况下, 前端人员与后端的交流和互相了解已经极为方便. 同时为了确保联调成功, 对js的Promise语法等有了更深刻的理解

7.7 体验了协作开发

与美术人员和wxml编写人员的交流是很繁琐也很有成就感的事。协作开发基本是一个持续集成和QA的过程，不仅能带来高效率，也能带来更多的思维碰撞，更多的idea和更少的bug。同时为了保证开发的同步，我们采用了基于git conversion的严格的分支管理策略,，方便项目管控。

7.8 了解了我国现在的网络监管现状

服务器域名需要冗长的备案手续，微信小程序的审核也有诸多掣肘，一方面在开发过程中给我们带来了很多不便，但同时也让我们直观而深刻的认识到了我国现在的网络监管体系之严密，也认识到对全新领域进行作业之前对相应行业环境，法律条文进行早期调研的必要

7.9 学习并掌握了简单的服务器部署

鉴于微信云开发服务不具有普适性，后续的拓展也没有成熟的SpringBoot框架方便，因此在确认需求后，决定使用服务器作为后端，在部署应用的过程中，充分掌握了Linux系统的基本操作，服务器环境的基本配置，以及后端代码的托管与日志信息重定向等常用的技术。

7.10 学习并掌握了POSTMAN工具

为了实现前后端分离并行开发，需要确保接口的一致性，在验证接口输入输出的过程中掌握了POSTMAN对网络接口测试的管理，实现了对接口测试工作的复用，保证了迭代开发时对接口正确性的保障。