

# 微信小程序应用开发赛

## Hi洞 介绍

### 1 项目介绍

#### • 1.1 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。

#### • 1.2 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。

##### - 1.2.1 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。

### • 1.2.2 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。

## • 1.3 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。

## • 1.4 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。

## • 1.5 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。

## • 1.6 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。

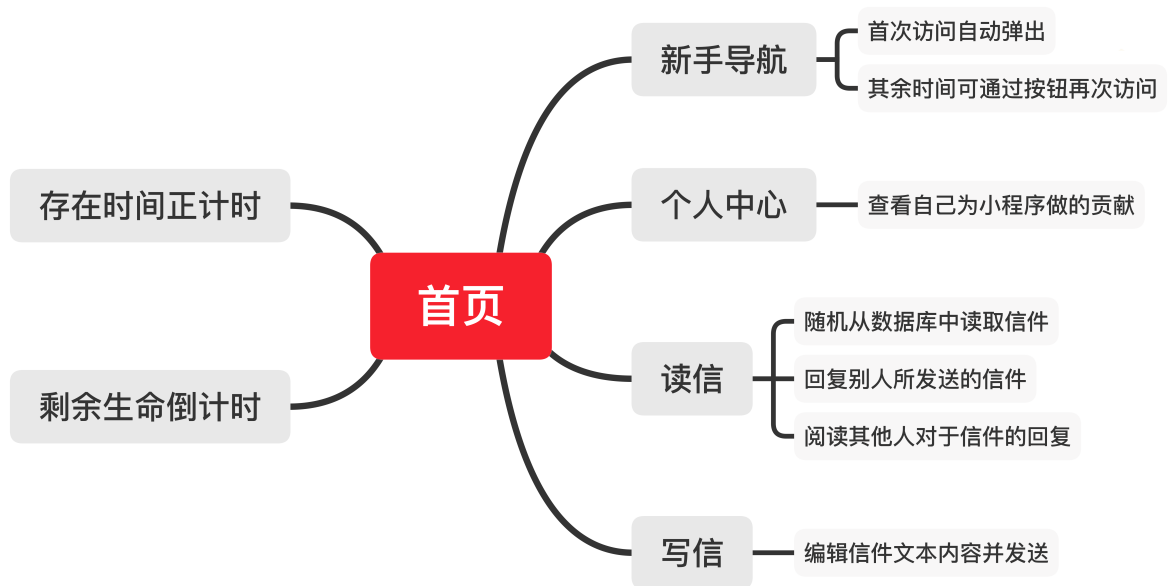
所以我们针对之前的树洞提出了一套新的机制，希望能够解决之前遇到的一系列问题。

# 2 产品设计

## • 2.1 产品功能

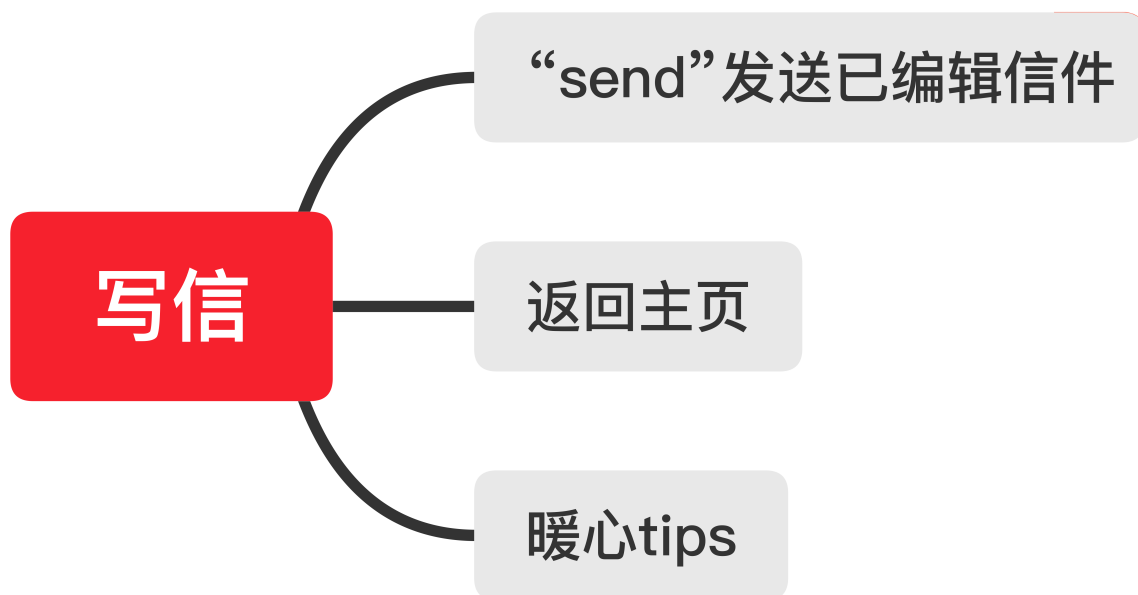
### - 2.1.1 首页

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。



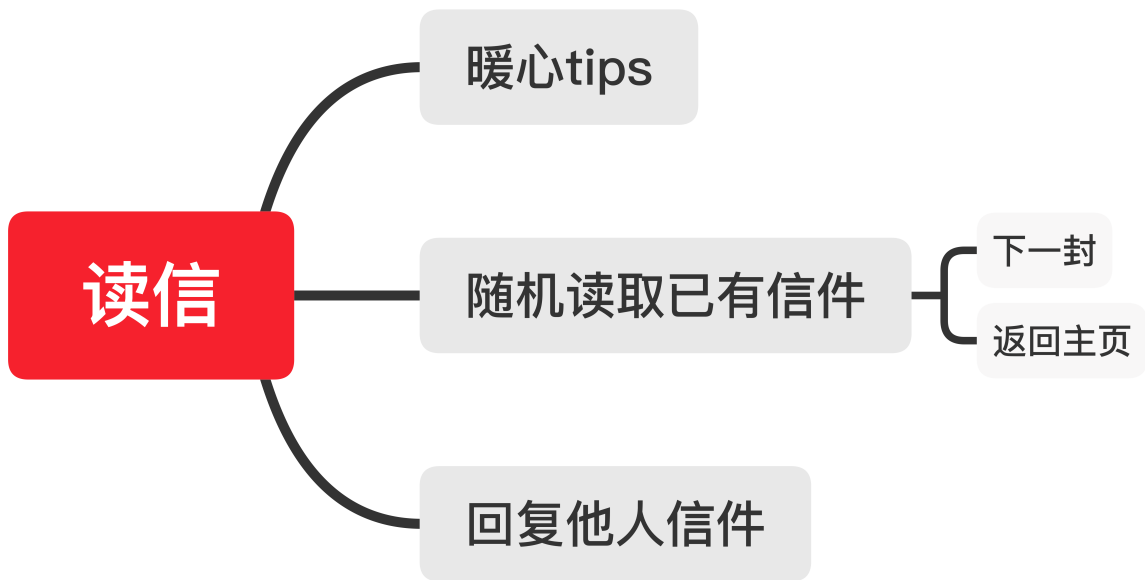
### 2.1.2 写信

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。



### 2.1.3 读信

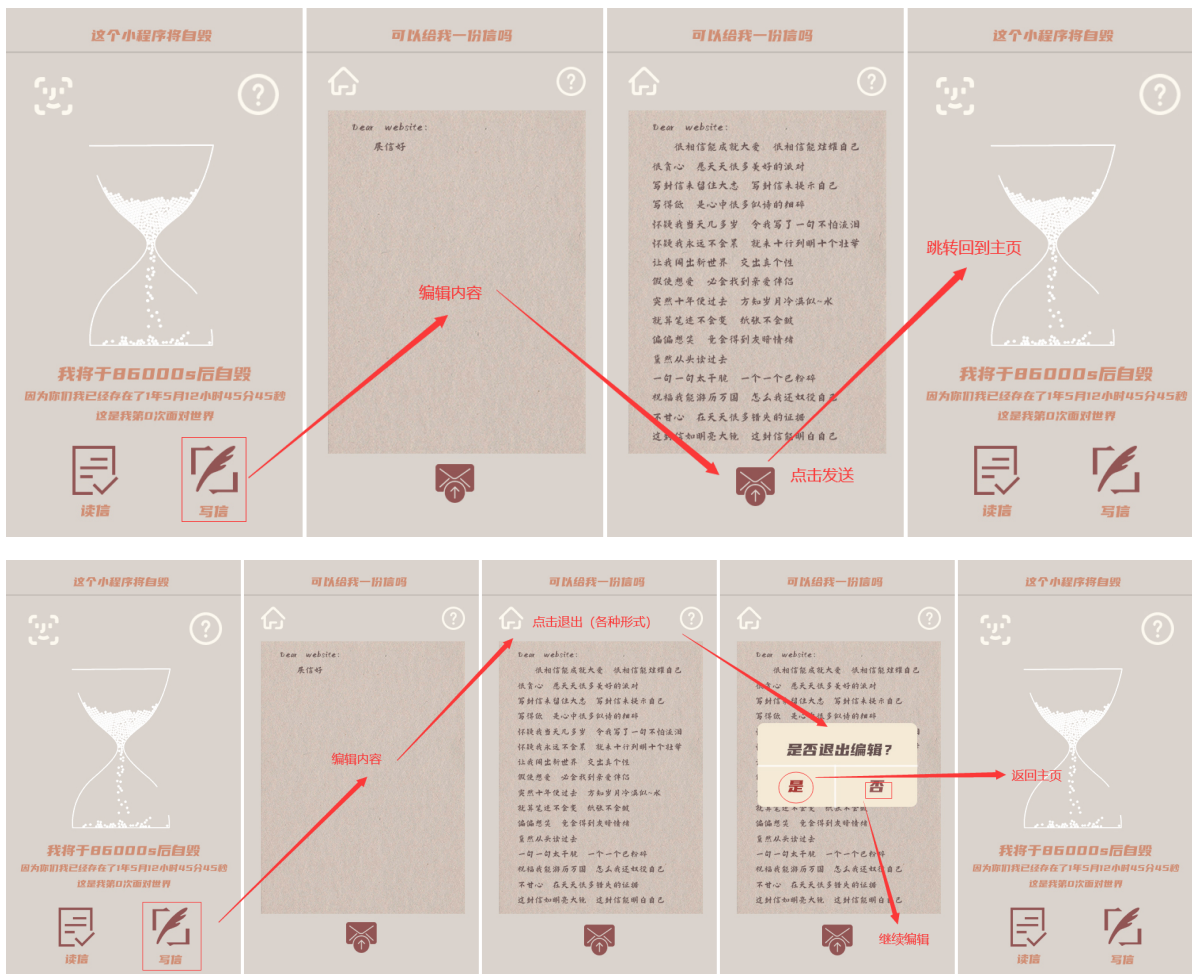
在读信界面用户可以查看从数据库的随机提取的已有信件和其回信，也可以选择编辑是否回复别人的信，并可以同一界面编辑内容进行发送。在读信界面也可以进入暖心导航页面，以关注用户的心理健康问题。



## • 2.2 交互设计

### - 2.2.1 写信流程

用户可以通过小程序写信以表达自己的情绪



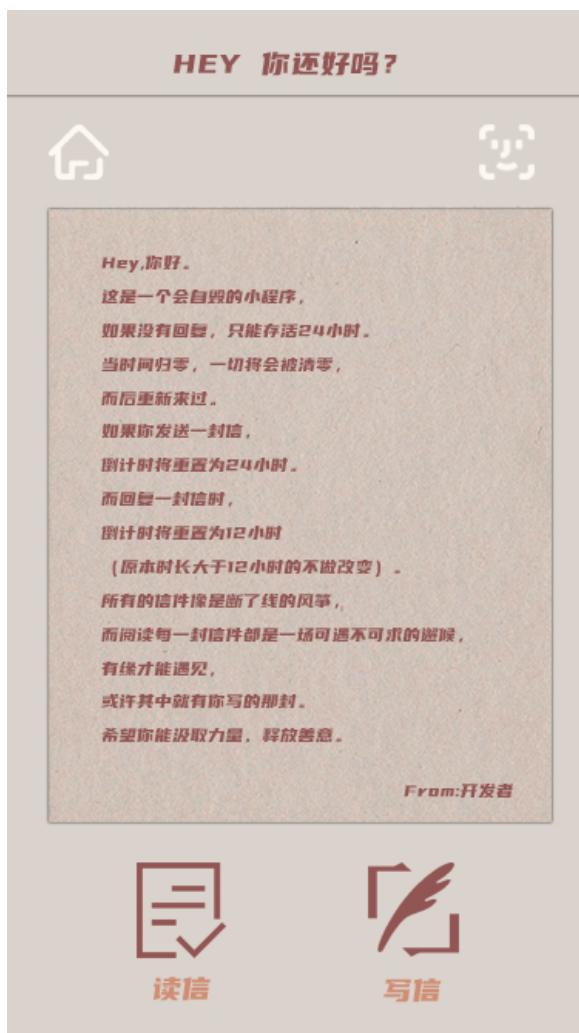
## 2.2.2 读信流程

用户可以通过小程序阅读信件，同时回复安慰他人



## 2.2.3 其余页面展示

新手导航（左图） 个人中心（右图）



## • 2.3 运营方案

### - 2.3.1 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。



### - 2.3.2 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。

## 3 关键技术

### • 3.1 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

### • 3.2 服务端

#### - 3.2.1 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

#### - 3.2.2 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

## 4 系统设计

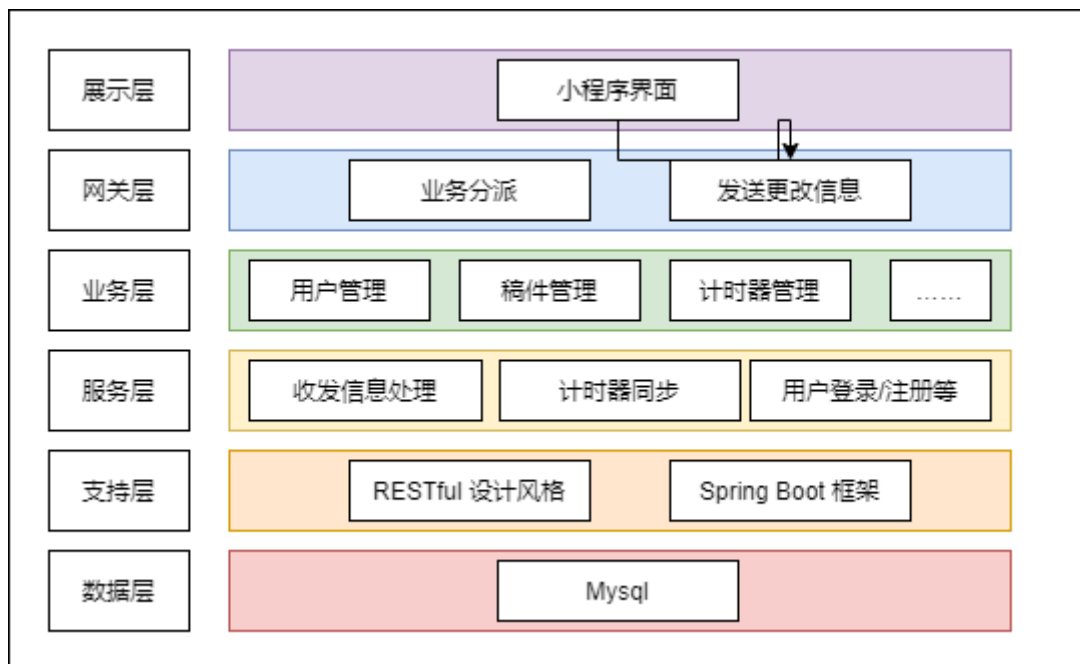


## • 4.1 总结设计

### - 4.1.1 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：

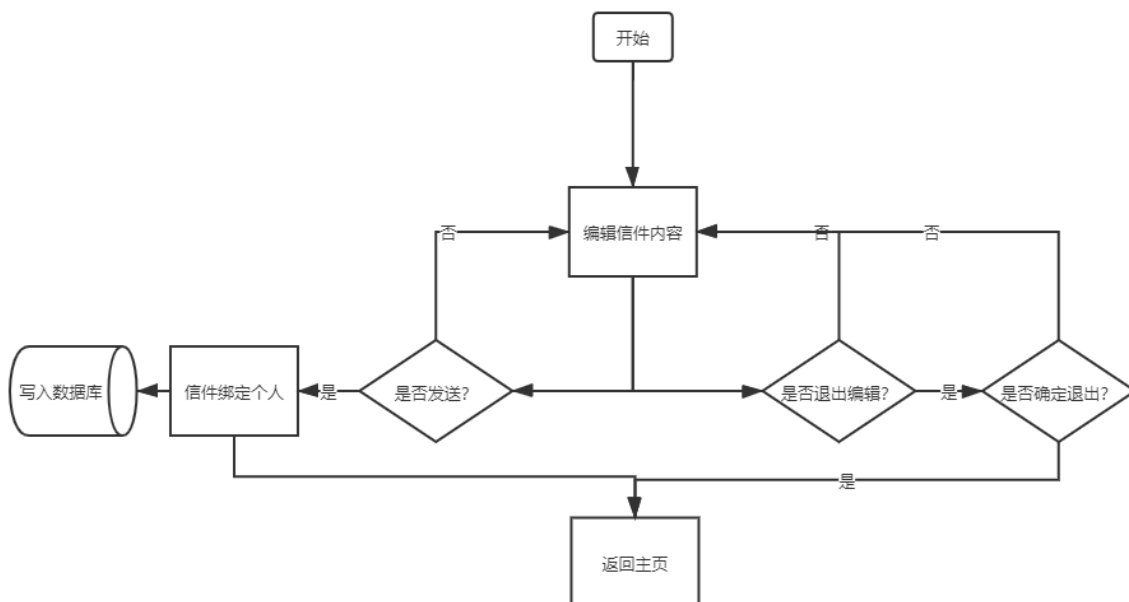
1. 数据层为系统提供数据支持，采用 MySQL 作为主要的数据库进行数据存储
2. 支撑层为系统的设计开发提供支持，服务接口采用 RESTful 设计风格、后端采用 Spring Boot 框架快速构建 Java 应用
3. 服务层包括了系统运行过程中需要的一些公共基础服务，如收发信处理，计时器同步，用户登录/注册等服务；
4. 业务层包含了系统主要的业务服务，包括用户管理、信件管理、计时器管理、概况管理
5. 逻辑层包含了系统的判断逻辑,负责对显示层传入的消息进行业务分派，并向视图层发送更改信息。
6. 显示层负责页面显示,接受用户输入,并向逻辑层传递消息. 并且通过双向绑定,实现实时渲染.



### - 4.1.2 数据库设计

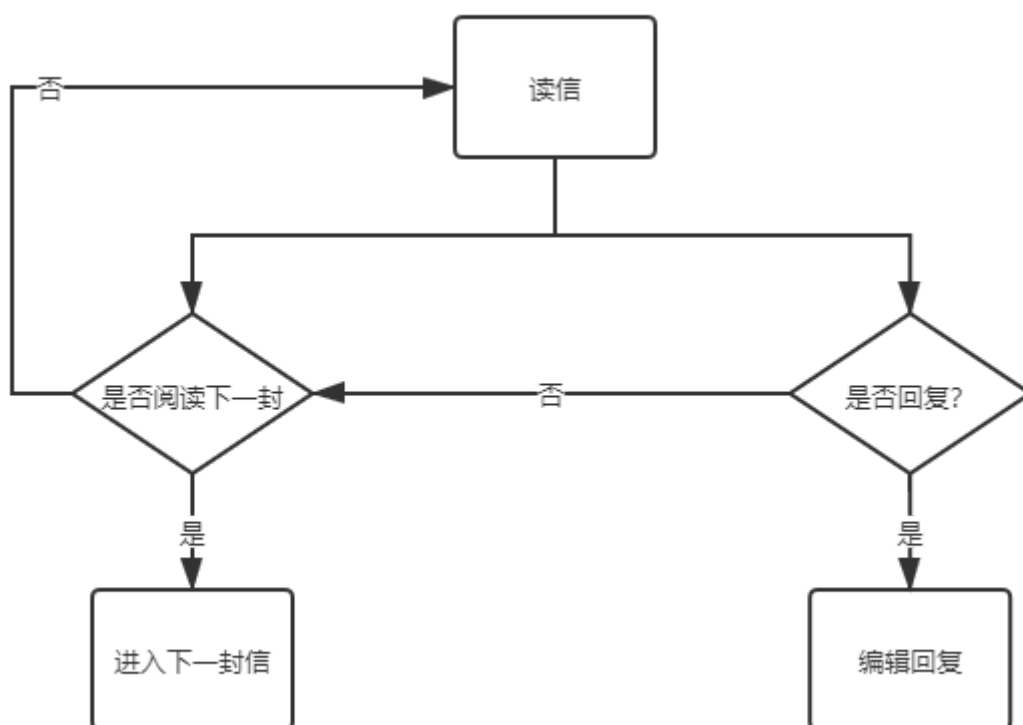
本应用的 E-R 图仅具有三个实体，分别是用户、信件、版本信息  
具体E-R 图如下所示.





#### 4.2.2 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。



### 4.2.3 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。

