# Barker树洞系统

### 大致介绍

1. 作者：Teager
2. 需求与UI：Teager
3. 前端负责：Teager
4. 后端与数据库：Teager
5. 技术栈：
6. UI：Figma
7. 前端：uniapp，Vue3，Pinia，React Native
8. 后端：Node.js，Koa
9. 数据库：MongoDB
10. 服务器：Nigix
11. 系统介绍：该系统为私人项目，开发一个可以让用户浏览和发表一些生活中的吐槽或对事物与事件的感想等的树洞系统。用户可以在系统中浏览由自己和其他用户发表在树洞里发表的各种碎碎念，也可以发表自己的一些对生活或事物等的感想与吐槽进行发表公开

### 数据结构

1. 语句：用户发表公开和浏览的句子实体
2. 序列号：用于区别其他的语句，使用nanoid生成保证每个语句唯一
3. 内容：语句的内容，限制100字以内
4. 发表人：语句的创建者，与管理员帐号无关
5. 点赞量：语句的点赞数量，为number类型，>=0
6. 公开状态：语句是否公开，只能管理员修改，0为未公开，1为公开
7. 类别：语句的感情类别，0为正能量，1为负能量
8. 标签：语句的题材，由数组组成
9. 时间：语句的时间（年/月/日）
10. 每日语句：用于储存每日发表并审核通过的语句，以日期树的叶节点储存于日期树中
11. 时间：每日语句结构的时间标记（年/月/日）
12. 语句列表：在上述时间发表并审核通过的语句列表，元素结构为语句
13. 管理员：负责审核语句发表的帐号，可以对语句实体进行操作（通过，删除，修改）
14. 帐号：管理员帐号，字符串格式，新建时需要验证是否重复
15. 密码：管理员密码，md5加密，只能字母，数字并限制在8-16位

注意：管理员只能由数据库添加

### 三、功能说明

#### （一）普通用户

1. 软件进入：普通用户进入不需要登录，默认进入浏览模块，有浏览、发表和本地管理三个·大功能

2. 浏览功能：用户进入浏览模块后，默认随机展示语句，每个语句包含内容与作者，有按钮可以刷新语句，也可在30秒后自动刷新语句。可以根据类别筛选想看的语句

3. 发表功能：用户进入发表模块后，可以在内容框内输入想发表的语句，在作者框内输入名字，并选择类型和分类后，按发表按钮可以发表语句，但语句需要管理员审核后才能在浏览模块展示出来

#### （二） 管理员

1. 登录功能：在发表模块后点击登录按键可以跳到登录界面，输入密码和帐号后可以登录进管理员系统，页面模块也变成审核模块和管理模块

2. 审核功能：管理员进入审核模块后，可以查看根据发表时间顺序展示的待审核的语句，可以看到内容，时间，作者与分类，其中不能过审的可以删除，也可以修改分类，通过的话可以选择完全公开或匿名

3. 管理功能：管理员进入管理功能后，默认查看3天内已过审的语句，也可以自行选择时间查看语句。对已过审的语句管理员可以删除，也可以进行修改，但只能修改分类以及是否匿名

### 四、架构解析

#### （一）前端

1. 视图层：负责展示界面以及对界面上的数据与元素进行控制，包括组件，路由，布局等，数据只能控制与显示有关的数据，技术栈为Vue3，uniapp
2. 控制层：负责对数据层的数据进行操作，作为视图层和数据层之间的沟通桥梁，依据数据分类，技术栈为JS
3. 数据层：储存需要用到的数据，依据实体分类，技术栈为Pinia

#### （二）后端

负责对数据库进行查询与操作，并将操作与查询结果返回给前端，使用框架为Koa

1. 查询：对数据库数据进行查询并返回结果，可传入参数限制查询条件，一般查询先根据时间沿树结构查询到对应的DayChats，然后再根据条件进行查询
2. 删除：对数据库数据进行删除操作，需要传入语句时间与序列号，先根据时间找到DayChats，然后根据序列号找到语句并删除
3. 增加：对数据库数据进行增加操作，先根据时间寻找对应DayChats，若没有则会新建DayChats，然后插入对应语句
4. 修改：对数据库某个数据进行修改，先进行查询操作，然后对数据中指定属性值进行修改

接口请求：

1. 获取语句：get请求，获取语句，不传参则随机获取，传参则根据所传参数获取，只能获取一条语句（注，若获取语句的state为2，则需要把author置空后再返回）
2. 获取语句列表：get请求，需要验证token，获取语句列表，传入参数为时间和列表类型（0为过审1未过审），先判断获取哪种列表，若未过审列表则返回前10条，若过审列表则根据时间筛选出对应时间的chatlist，若未传时间则传最近的一天
3. 更新语句：post请求，传入参数为语句的id，state，tag，time，请求需要验证token是否有效，根据time和id寻找对应的语句然后更新传参对应的参数值，若找不到语句则返回修改无结果
4. 增加语句：post请求，传入参数为content，author，tag，后端自动生成语句实体对象后根据生成对象时的时间（后端生成）储存进数据库auditList（未过审列表）对应时间树叶节点的chatlist中
5. 登录：post请求，传入name和md5加密后的password，先根据name寻找对应账户，然后取出数据库中的账户的password进行aes解密，与请求中password进行md5解密后的结果进行比对，若相同则返回成功，同时生成token前后端各留一份，后端token设置过期时间
6. 过审语句：post请求，需要验证token是否过期，传入参数为完整的语句实体对象，先根据id和time执行删除语句操作去除未过审列表里的语句，然后将参数的语句对象根据time放入对应的chatlist里
7. 删除语句：delete请求，需要验证token是否过期，传入参数为id和time，先根据time寻找对应的chatlist再根据id找到对应语句并删除

#### （三）数据库

储存语句数据，分为过审，未过审与管理员三个数据库，使用数据库为MongoDB