

# 校园失物招领系统说明文档

姓名(组长): 孔欣然 学号: 2312190201 班级: 计科2302

姓名: 王琰 学号: 2312190223 班级: 计科2302

## 项目背景

校园生活中, 物品丢失与寻找一直是困扰师生的一大难题。传统的信息发布方式分散在公告栏、微信群等不同渠道, 信息不对称、更新不及时、可信度低, 导致失物归还率低、寻找效率差。为此, 我们开发了“校园失物招领系统”, 旨在通过数字化平台整合信息流, 为师生提供一个统一、便捷、可信的在线失物招领服务。

本系统面向全校师生及后勤管理人员, 用户可随时发布丢失或拾获信息, 并通过智能搜索快速匹配; 管理人员则可对信息进行审核与分类管理。

在技术层面, 项目采用前后端分离架构, 前端融合Vue框架与原生开发, 后端基于Spring Boot实现, 涵盖用户认证、跨域处理等实用功能。通过该项目, 我们不仅能切实解决校园实际需求, 也将在全栈开发、团队协作与系统部署等方面获得宝贵的工程实践经验, 是一次技术学习与社会价值结合的有益尝试。

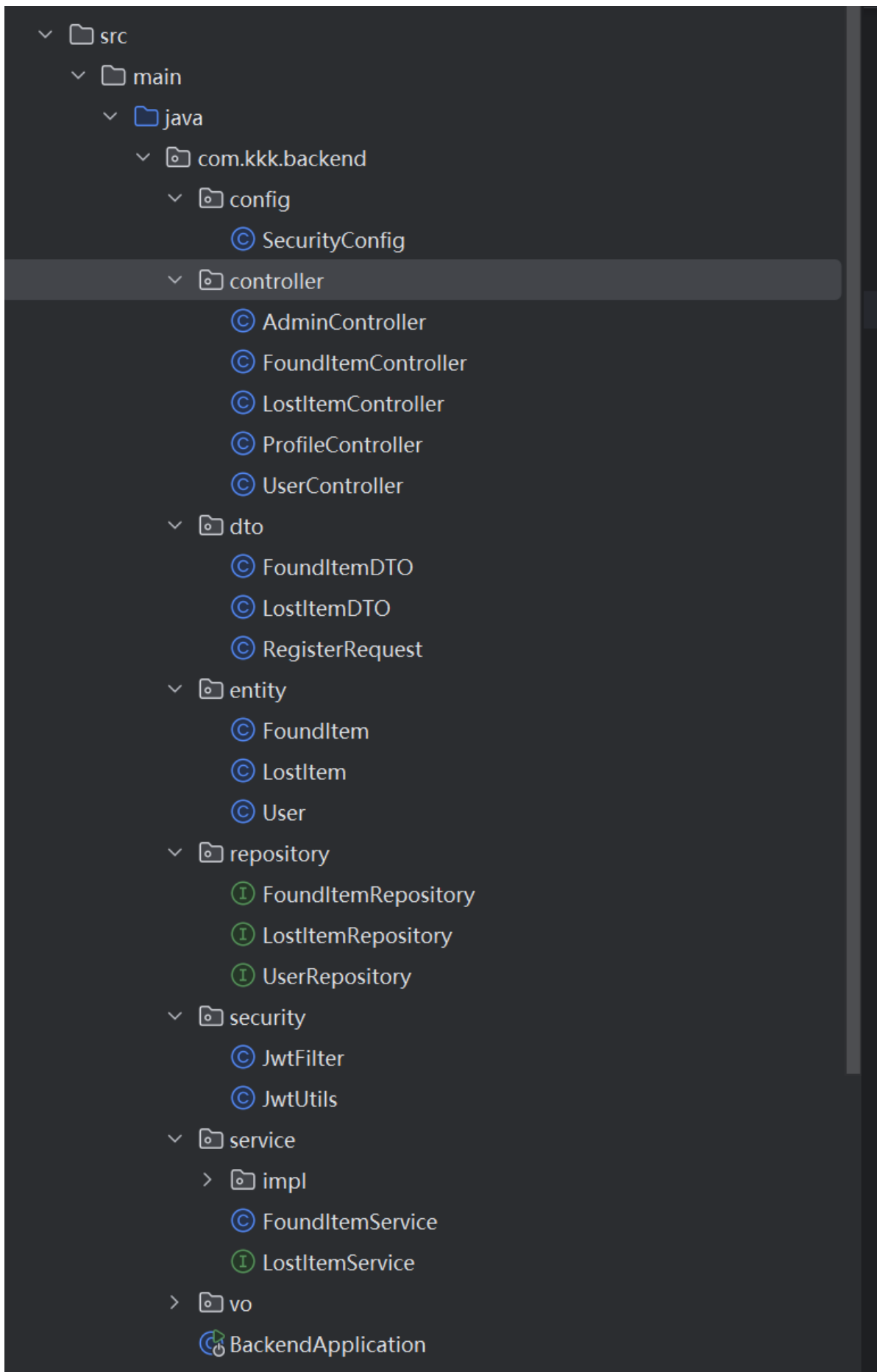
## 技术栈

- 前端:** Vue 3 + vue-router 4+Element Plus + Axios+Vite+JS
- 后端:** Spring Boot (Java)
- 数据库:** MySQL
- 代码托管:** GitHub

前端架构:

> assets	●
> components	●
> router	●
✓ views	●
✓ admin	●
▼ AdminDashboard.vue	U
▼ ItemManagement.vue	U
▼ UserManagement.vue	U
✓ found	●
▼ FoundDetail.vue	U
▼ FoundList.vue	U
▼ FoundPublish.vue	U
✓ lost_item	●
▼ LostItemDetail.vue	U
▼ LostItemEdit.vue	U
▼ LostItemList.vue	U
▼ LostItemNew.vue	U
▼ Home.vue	U
▼ Login.vue	U
▼ Profile.vue	U
▼ Register.vue	U
▼ App.vue	U
JS main.js	U

后端架构:



## 用户功能

## 1. 用户端功能

- 用户注册
- 用户登录
- 发布失物/招领
- 查看所有失物/招领信息
- 管理自己发布的信息
- 修改密码/用户名

## 2. 管理端功能

- 用户管理
- 平台所有失物/招领信息管理

## 关键技术实现

### 1. 实现JWT鉴权机制（孔欣然负责）

为确保系统的安全性并实现用户状态管理，本项目采用 **JWT (JSON Web Token)** 作为用户认证与鉴权的核心机制。

#### 实现方案概述：

- **令牌生成**：用户登录成功后，后端根据用户名和角色生成带有有效期的JWT令牌，使用HS256算法进行签名加密。
- **请求验证**：前端在后续请求的 `Authorization` 请求头中携带Bearer Token，后端通过JwtFilter拦截器对所有非公开接口进行令牌验证。
- **权限解析**：验证通过后，系统从令牌中提取用户身份信息并设置到Spring Security上下文中，供后续接口进行角色权限控制。
- **异常处理**：对于令牌过期、格式错误或未携带的情况，系统返回401状态码并给出明确的错误提示。

#### 核心组件设计：

- **JwtUtils工具类**：负责令牌的生成、解析、过期判断等基础操作，集中管理密钥和有效期的配置。
- **JwtFilter过滤器**：作为Spring Security的入口过滤器，实现对请求的拦截与令牌验证逻辑。
- **SecurityConfig配置**：整合过滤器链、配置跨域支持，并放行注册、登录等公开接口。

#### 关键代码示例：

```
// 登录成功后生成Token
String token = jwtUtils.generateToken(username, role);

// JwtFilter中验证Token
if (authHeader != null && authHeader.startsWith("Bearer ")) {
    String token = authHeader.substring(7);
    if (!jwtUtils.isExpired(token)) {
        // 设置认证信息到Spring Security上下文
        // ...
    } else {
        response.setStatus(HttpServletResponse.SC_UNAUTHORIZED);
        response.getWriter().write("Token 已过期");
    }
}
```

该鉴权机制实现了基本的安全防护，还通过日志输出和错误处理，为开发和调试提供了便利。角色信息的嵌入，为后续管理员与普通用户的权限区分奠定了基础。

## 2.实现前端实时信息检索（王琰负责）

为了实现高效的招领信息查询功能，我们设计了前端实时信息检索系统，支持多维度筛选和即时过滤，提升用户体验。

从技术实现角度看，前端信息检索的几个关键实现点：

- **搜索框设计**：使用Element Plus的 `el-input` 组件实现模糊搜索功能
- **多条件筛选**：通过分类和状态两个维度进行精确筛选
- **前端实时过滤**：利用Vue 3的`computed`属性实现本地数据过滤

多维度筛选的实现如下：

```
<div class="filter-group">
  <span class="filter-label">分类:</span>
  <el-select
    v-model="filterCategory"
    placeholder="全部分类"
    @change="handleSearch"
  >
    <el-option label="证件" value="证件"></el-option>
    <el-option label="钱包" value="钱包"></el-option>
    <el-option label="钥匙" value="钥匙"></el-option>
    <el-option label="电子产品" value="电子产品"></el-option>
    <el-option label="个人物品" value="个人物品"></el-option>
    <el-option label="其他" value="其他"></el-option>
  </el-select>
</div>
```

前端实时过滤的核心算法：

```
const filteredList = computed(() => {
  if (!list.value.length) return [];

  let filtered = [...list.value];

  // 关键词筛选
  if (searchKeyword.value.trim()) {
    const keyword = searchKeyword.value.trim().toLowerCase();
    filtered = filtered.filter(item =>
      (item.title && item.title.toLowerCase().includes(keyword)) ||
      (item.description && item.description.toLowerCase().includes(keyword)) ||
      (item.foundLocation && item.foundLocation.toLowerCase().includes(keyword))
    );
  }

  // 分类筛选
  if (filterCategory.value) {
    filtered = filtered.filter(item => item.category === filterCategory.value);
  }

  // 状态筛选
  if (filterStatus.value !== "") {

```

```
const status = parseInt(filterStatus.value);
filtered = filtered.filter(item => item.status === status);
}

// 按时间倒序排序
return filtered.sort((a, b) => {
  return new Date(b.foundTime || b.createTime) - new Date(a.foundTime ||
a.createTime);
});
});
```

该信息检索方案的特点是：

1. **实时响应**：用户输入后立即显示筛选结果
2. **多维度查询**：支持关键词、分类、状态三种查询方式
3. **性能优化**：前端本地过滤，减少服务器压力
4. **用户体验好**：提供筛选统计和清空功能

## 数据库设计

### 数据库设计原则

在设计数据库时，遵循以下原则：

- 结构清晰，便于理解与维护
- 失物信息与招领信息作为**独立业务模块**分别建表
- 保证基本完整性与扩展性
- 字段命名语义明确，避免歧义

### 数据库整体表结构

本系统数据库共设计 **4 张核心表**：

1. 用户表 `user`
2. 失物信息表 `lost_item`
3. 招领信息表 `found_item`
4. 管理日志表 `admin_log`

### 各表详细设计

#### 1. 用户表 `user`

##### 表用途

存储系统用户的基本信息与角色类型。

##### 字段设计

字段名	类型	说明
id	BIGINT	主键，自增
username	VARCHAR(50)	用户名
password	VARCHAR(255)	密码（加密存储）
contact	VARCHAR(100)	联系方式（微信号）
role	VARCHAR(20)	用户角色（user / admin）
create_time	DATETIME	注册时间

## 2.失物信息表 `lost_item`

### 表用途

记录用户发布的失物信息。

### 字段设计

字段名	类型	说明
id	BIGINT	主键
title	VARCHAR(100)	失物标题
category	VARCHAR(50)	物品类别
lost_location	VARCHAR(100)	丢失地点
lost_time	DATETIME	丢失时间
description	TEXT	详细描述
image_url	VARCHAR(255)	图片路径
status	INT	状态（0 未找回 / 1 已找回）
user_id	BIGINT	发布者 ID
create_time	DATETIME	发布时间

## 3.招领信息表 `found_item`

### 表用途

记录用户发布的招领信息。

### 字段设计

字段名	类型	说明
id	BIGINT	主键
title	VARCHAR(100)	招领标题
category	VARCHAR(50)	物品类别
found_location	VARCHAR(100)	捡到地点
found_time	DATETIME	捡到时间
description	TEXT	描述（可模糊）
image_url	VARCHAR(255)	图片路径
status	INT	状态（0 未认领 / 1 已认领）
user_id	BIGINT	发布者 ID
create_time	DATETIME	发布时间

设计说明

招领信息在描述字段上不强制填写完整特征信息，以防止冒领行为。

4.管理日志表 admin\_log

表用途

记录管理员操作，用于系统维护与审计。

字段设计

字段名	类型	说明
id	BIGINT	主键
admin_id	BIGINT	管理员 ID
operation	VARCHAR(255)	操作内容
create_time	DATETIME	操作时间

数据库关系说明

- user 与 lost\_item：一对多
- user 与 found\_item：一对多
- 失物信息与招领信息互不直接关联，通过人工匹配完成

建表 SQL

```
CREATE TABLE user (  
  id BIGINT PRIMARY KEY AUTO_INCREMENT,  
  username VARCHAR(50) NOT NULL,  
  password VARCHAR(255) NOT NULL,  
  contact VARCHAR(100),  
  role VARCHAR(20) DEFAULT 'user',  
);
```



```
        create_time DATETIME DEFAULT CURRENT_TIMESTAMP
    );

CREATE TABLE lost_item (
    id BIGINT PRIMARY KEY AUTO_INCREMENT,
    title VARCHAR(100) NOT NULL,
    category VARCHAR(50),
    lost_location VARCHAR(100),
    lost_time DATETIME,
    description TEXT,
    image_url VARCHAR(255),
    status INT DEFAULT 0,
    user_id BIGINT,
    create_time DATETIME DEFAULT CURRENT_TIMESTAMP
);

CREATE TABLE found_item (
    id BIGINT PRIMARY KEY AUTO_INCREMENT,
    title VARCHAR(100) NOT NULL,
    category VARCHAR(50),
    found_location VARCHAR(100),
    found_time DATETIME,
    description TEXT,
    image_url VARCHAR(255),
    status INT DEFAULT 0,
    user_id BIGINT,
    create_time DATETIME DEFAULT CURRENT_TIMESTAMP
);
```

## 开发分工与技术职责

---

成员	负责模块	具体任务与职责	技术点
孔欣然	用户与权限管理模块	1. 用户注册、登录流程设计与实现 2. 登录接口与前端交互开发 3. JWT令牌生成与登录状态校验逻辑 4. 用户权限控制与角色管理	JWT鉴权、Spring Security、登录状态管理、密码加密存储
	失物信息管理模块	1. 失物信息数据结构设计 2. 失物发布、编辑、删除功能实现 3. 失物列表与详情页展示开发 4. 失物状态更新逻辑设计	接口设计
	个人中心模块 (失物部分)	1. "我发布的失物"列表展示 2. 个人失物信息管理操作实现	用户关联查询
	文档	1. 编写个人开发日志、系统说明文档	文档编写
王琰	招领信息管理模块	1. 招领信息发布与维护功能开发 2. 招领列表、详情展示实现 3. 招领状态管理逻辑设计	接口设计
	信息检索与筛选模块	1. 搜索接口设计与实现 2. 分类、状态等多维度筛选 3. 前端筛选交互设计	关键词搜索、多条件筛选、模糊查询优化
	系统管理与维护模块	1. 管理员功能设计与开发	权限分级管理
	文档	1. 编写个人开发日志、系统使用说明文档	文档编写

## 附录

GitHub 仓库地址: [gaiyyyy/campus-lost-found-system](https://github.com/gaiyyyy/campus-lost-found-system)