

毕业设计开发文档

—— 智慧社区网格化管理服务平台 ——

一、项目基本概况

项目属性	内容
项目名称	智慧社区网格化管理服务平台
项目性质	本科毕业设计(地理信息科学 - 计算机方向)
开发地点	学校实验室 / 个人工作站
指导模拟	云导师 (Gemini)
核心技术	Spring Boot + Vue 3 + PostGIS + Redis

二、开发进度计划 (甘特图建议)

2.1 准备阶段 (第1-2周)

目标: 完成基础环境搭建与技术预研, 跑通 "Hello World"。

- 环境搭建:
 - 安装配置 JDK 17、Node.js、IntelliJ IDEA。
 - 数据库: 安装 PostgreSQL 并启用 PostGIS 扩展。
 - 中间件: 安装 Redis 并配置本地连接。
- 技术预研:
 - 熟悉 Leaflet 或 Mapbox GL 前端地图库的基本 API。
 - 掌握 PostGIS 核心空间查询 SQL(如 ST_Contains, ST_Intersects)。
 - 了解 Redis 的基础命令(Set/Get)及缓存逻辑。

2.2 核心开发阶段 (第3-8周)

目标: 完成系统核心功能开发, 实现数据流转闭环。

- 数据库设计:
 - 设计并创建网格表(tb_grid)、事件表(tb_event)、用户表(tb_user)。
 - 完成数据库 ER 图建模。
- 后端开发 (Spring Boot):
 - 搭建框架, 集成 MyBatis-Plus。

- 集成 Redis: 配置 RedisTemplate, 实现 Token 管理与热点数据缓存。
- 编写 API 接口: 实现用户登录、网格数据的增删改查 (CRUD)。
- 前端开发 (Vue 3):
 - 搭建脚手架, 封装地图组件。
 - 实现核心功能: 地图加载、网格在线绘制工具、事件上报弹窗。

2.3 联调与优化阶段 (第9-10周)

目标: 提升系统稳定性与性能, 优化用户体验。

- 前后端联调: 确保前端绘制的 GeoJSON 图形能正确保存至 PostGIS 数据库。
- 空间分析实现: 核心算法攻关 —— 实现“点落在哪个网格内的自动判断”。
- 性能优化: 引入 Redis 缓存首页“驾驶舱”的统计数据, 提升页面加载速度。
- UI 美化: 利用 ECharts 开发数据可视化大屏。

2.4 论文与答辩 (第11-12周)

- 整理系统运行截图、录制演示视频。
- 撰写毕业论文, 制作答辩 PPT。

三、开发环境与工具配置

3.1 硬件与系统

- 操作系统: Windows 10/11 或 MacOS
- 开发工具: IntelliJ IDEA (后端), Visual Studio Code (前端)

3.2 运行环境 (Server)

- Java: JDK 17+
- Node.js: 16.x 或 18.x

3.3 数据库与中间件 (Core)

- PostgreSQL 14+: 必须安装 PostGIS 扩展 (用于存储 Geometry 数据)。
- Redis 6.0+: 用于缓存与会话管理 (Session/Token)。

3.4 调试工具

- API 调试: Postman 或 Apifox
- 数据库管理: PgAdmin 4 或 Navicat Premium

四、开发规范与注意事项

1. 空间数据规范:
 - 统一使用 WGS84 (EPSG:4326) 坐标系。
 - 前端采集坐标与后端存储坐标必须保持一致, 避免偏移。
2. 代码规范:
 - 后端接口遵循 RESTful API 设计规范 (GET/POST/PUT/DELETE)。

- 前端 Vue 组件需合理拆分，避免单文件代码行数过长。
3. 异常处理：
- 遇到 PostGIS 空间函数报错时，优先检查 Geometry 数据格式。
 - Redis 连接超时需做好降级处理(即 Redis 挂了不影响主营业务运行)。

五、项目详细设计

5.1 引言

随着城市精细化治理需求的增加，“社区网格化管理”已成为基层治理的重要抓手。传统的社区管理依赖人工记录和纸质地图，存在信息更新滞后、事件位置不直观、指挥调度效率低等痛点。本次毕业设计旨在利用 WebGIS 技术，构建一套“智慧社区网格化管理服务平台”，实现社区管理要素的数字化、事件处置的流程化以及决策分析的可视化。

5.2 项目背景与意义

本项目结合地理信息技术(GIS)与现代 Web 开发技术，解决“最后一公里”的管理难题。通过将社区划分为若干个独立的“网格”单元，并将人、地、物、事、情等信息落实到网格中，实现“底数清、情况明”。系统特别引入 **Redis** 缓存技术，不仅能提升社区工作人员的办事效率，还能在高并发场景下为突发事件(如火灾、防疫)提供快速响应支持。

5.3 技术选型架构

前端技术栈

- **Vue 3**:采用 Composition API, 高效构建单页应用。
- **Leaflet / Mapbox GL**:轻量级开源地图引擎，负责地图渲染与交互。
- **Element Plus**:后台管理界面 UI 组件库。
- **Turf.js**:前端地理空间分析库(如计算面积、长度)。
- **ECharts**:可视化图表库。

后端技术栈

- **Spring Boot**:快速构建微服务应用。
- **MyBatis-Plus**:ORM 框架，简化数据库操作。
- **Redis (企业级亮点)**:
 - 缓存优化:缓存首页“驾驶舱”统计指标，减轻数据库 Count 压力。
 - 会话安全:存储用户 Token，管理登录状态与过期时间。

数据存储

- **PostgreSQL + PostGIS**:利用其强大的空间索引(GIST)和空间函数(ST_Contains)，处理复杂的地理几何运算。

5.4 功能需求列表清单

序号	所属模块	功能名称	功能描述	备注
----	------	------	------	----

1	系统管理	用户权限管理	实现网格员、管理员的增删改查及 Token 认证	关联 Redis
2		组织架构管理	社区、街道办等部门结构管理	
3	GIS一张图	多底图切换	支持天地图(矢量/影像)、OSM、高德底图切换	
4		资源分布展示	在地图上展示楼栋、井盖、消防栓等 POI 分布	
5	网格管理	网格在线绘制	管理员在 Web 端地图上绘制、编辑多边形(Polygon)	GIS 核心功能
6		责任人绑定	将绘制好的网格与网格员进行关联绑定	
7	事件处置	地图选点上报	模拟移动端/PC 端在地图上点击，上报事件及照片	
8		智能自动分派	核心算法：根据坐标自动计算所属网格，推送给对应负责人	PostGIS 亮点
9		任务状态流转	待处理 -> 处理中 -> 已归档的状态管理	
10	数据看板	空间热力图	根据事件密度渲染地图热力图 (Heatmap)	

11		高性能统计	统计各网格处理效率，数据优先从 Redis 读取	性能优化点
----	--	-------	--------------------------	-------

5.5 预期成果展示

1. 社区网格全景可视化: GIS 一张图直观展示社区彩色网格, 不同颜色代表不同预警等级。
2. 在线网格划分工具: 提供类似 CAD 的 Web 绘图工具, 支持网格拆分合并, 实时入库。
3. 事件智能归属: 上报“井盖丢失”后, 系统自动定位所属“A区03号网格”并弹窗通知负责人。
4. Redis 加速驾驶舱: 整合 ECharts 与地图, 关键指标(如今日上报数)采用 Redis 缓存策略, 实现毫秒级加载。

5.6 实验总结与展望 (预填)

通过本次毕业设计, 成功构建了一套基于 Spring Boot + Vue + PostGIS + Redis 的全栈式网格化管理平台。项目不仅实现了基本的 CRUD 业务, 更深入应用了 GIS 的空间分析能力, 解决了传统管理中“位置模糊”的痛点。此外, 通过引入 Redis 缓存技术, 有效解决了海量数据统计时的性能瓶颈问题, 提升了系统的并发处理能力和企业级应用价值。