文件类型编号：

*基于GIS地图的可视化智慧安防系统*

**Project开发计划书**

**编写日期：**

**2025/6/5**

**项目负责人（组长）:**

**徐祺翔（***E-mail地址***）**

本项目的项目组长

西北工业大学软件工程实验

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 改版履历 | | 文件名 | | 文件编号： |
| 制定部门： |
| 版数 | 承认/日期 | 查阅/日期 | 作成/日期 | 改版内容 |
| 版本序号 |  |  | 作者名称/作成日期 | 改版的主要内容 |
|  |  |  |  |  |

目录

[1．项目概要 1](#_Toc200014902)

[1.1 项目名 1](#_Toc200014903)

[1.2 开发目的和背景 1](#_Toc200014904)

[1.3 功能概要 2](#_Toc200014905)

[1.4 作业范围 4](#_Toc200014906)

[1.5 作业形态 5](#_Toc200014907)

[1.6 设计输入与设计输出 5](#_Toc200014908)

[1.7 开发规模 6](#_Toc200014909)

[2．体制 7](#_Toc200014910)

[2.1 开发体制 7](#_Toc200014911)

[2.2 委托方联络窗口 7](#_Toc200014912)

[3．整体作业计划 8](#_Toc200014913)

[3.1 工数预测 8](#_Toc200014914)

[3.2 开发模型 9](#_Toc200014915)

[3.3 日程 9](#_Toc200014916)

[3.4 配置管理 10](#_Toc200014917)

[4．进度管理 11](#_Toc200014918)

[4.1 进度报告 11](#_Toc200014919)

[4.2 技术讨论 11](#_Toc200014920)

[5．质量计划（此部分实验课可省略） 12](#_Toc200014921)

[5.1 适用标准和质量记录 12](#_Toc200014922)

[5.2 工程移行基准 12](#_Toc200014923)

[6．交付物品 14](#_Toc200014924)

[6.1 交付物品一览 14](#_Toc200014925)

[7．作业环境 15](#_Toc200014926)

[7.1 开发环境 15](#_Toc200014927)

[7.2 运行环境 15](#_Toc200014928)

[7.3 测试环境 15](#_Toc200014929)

[8．技术学习计划 16](#_Toc200014930)

[9．风险管理（此部分实验课可省略） 17](#_Toc200014931)

[10．对委托方的要求事项 18](#_Toc200014932)

# 1．项目概要

## 1.1 项目名

基于GIS地图的可视化智慧安防系统

## 1.2 开发目的和背景

随着城市化进程的快速推进，城市规模不断扩张，人口密度持续上升，各类安全风险也随之增加。在传统安防体系中，视频监控虽已广泛应用，但主要依赖人工筛查视频画面。这种方式效率极为低下，安保人员长时间面对海量视频数据，极易因疲劳或主观因素产生误判，导致关键信息遗漏。例如在公共安全场景下，火车站、机场等人流密集区域，每天产生的视频数据量巨大，人工筛查难以快速定位异常情况；交通管理中，面对道路上川流不息的车辆，人工查看视频很难及时发现交通违法行为或事故隐患；大型活动监控时，如演唱会、体育赛事现场，复杂的人员流动和场景变化，依靠人工更难做到全方位、无遗漏的监控。

在复杂场景（如公共安全、交通管理、大型活动监控等）中，海量视频数据对监控智能化提出了更高的要求。通过自动检测视频中的异常行为，并生成分类与描述信息，不仅能极大提升监控效率，还能为应急处理提供及时可靠的决策支持。

随着人工智能的快速发展，视觉语言大模型（Vision-LanguageModel，简称VLM）在计算机视觉与自然语言处理领域展现了强大性能。这些模型通过跨模态学习，将视觉信息与语言信息有机结合，能够高效实现图像或视频内容的理解与描述。在视频监控领域，与传统的人工智能技术相比，VLM的应用实现了从视频到异常事件描述文本的跨模态生成。这使得模型的异常检测效果具有可解释性，操作人员能够清晰理解模型的判断依据，从而更容易接受和运用检测结果。例如，当模型检测到视频中有人员奔跑、聚集等异常行为时，能够用自然语言准确描述出事件的类型、发生地点、涉及人员数量等关键信息，帮助安保人员快速了解现场情况，做出准确应对决策。因此，VLM可以广泛应用于视频监控领域，成为提升监控智能化水平的重要推动力。

同时，随着大数据、物联网、人工智能等新一代信息技术的蓬勃发展，各行业数字化转型加速，对安防系统的智能化、可视化、集成化需求日益迫切。智慧安防作为智慧城市建设的关键环节，需要能够整合各类安防数据，实现多源信息的高效融合与深度分析，以提升城市安全防范的整体效能。在此背景下，引入GIS技术成为安防系统升级的必然选择。

GIS（地理信息系统）能够将地理空间数据与各种信息进行有机整合，以直观的地图形式呈现出来。它不仅可以清晰展示地理环境、建筑布局、道路分布等基础地理信息，还能叠加监控设备位置、应急资源分布、人员流动轨迹等安防相关数据。通过GIS强大的空间分析功能，如最短路径规划、缓冲区分析、空间查询等，可在安防事件发生时，快速为应急指挥提供科学决策支持，大大提高安防响应的及时性与准确性。例如在消防救援中，GIS能根据火灾地点快速规划出最佳救援路线，并展示周边消防水源、消防设施等信息，为灭火救援行动争取宝贵时间。

本项目开发基于GIS地图的可视化智慧安防系统，正是为了顺应这一发展趋势，解决传统安防系统存在的弊端。通过将GIS技术深度融入安防体系，结合VLM对视频监控的智能化升级，实现安防数据的地理可视化管理、实时应急指挥调度以及智能风险预测分析，提升安防系统的整体性能与智能化水平，为城市安全运行提供有力保障。该系统预计于2025年9月正式上线，主要面向政府智慧城市建设部门、校园及企业园区管理方、大型商业综合体运营者等对安防系统智能化、可视化有较高需求的客户群体。

## 1.3 功能概要

本开发项目基于Django框架构建，采用经典的MVT（模型-视图-模板）架构模式。在模型层面，主要负责与数据库进行交互，通过Django的ORM（对象关系映射）技术，能便捷地操作诸如MySQL等关系型数据库，实现数据的存储、读取、更新与删除等操作，为整个项目提供数据持久化支持。视图部分则充当请求处理与响应生成的角色，接收用户的HTTP请求，依据业务逻辑调用模型获取所需数据，并决定将哪些数据传递给模板进行展示，它在模型与模板间起到桥梁作用。模板用于生成最终呈现给用户的前端界面，由HTML、CSS和JavaScript构建，结合Django模板语言，将动态数据嵌入静态页面结构中，实现页面的动态展示效果。此外，系统还集成了视觉语言大模型（VLM）用于核心的视频异常检测功能，以及GIS（地理信息系统）相关技术实现地理可视化功能，不同功能模块间通过合理的接口设计与数据交互机制协同工作，共同支撑起整个项目的高效运行。具体的功能概要如下：

**(1)用户管理功能**

用户管理功能为系统的访问与操作提供了基础的身份验证与权限控制体系。在登录环节，系统支持多种登录方式，如用户名与密码组合登录，同时兼容第三方账号登录，例如微信、QQ登录等，极大地方便了用户快速接入系统。对于新用户，注册流程设计简洁且安全，用户需填写必要信息，如用户名、密码、邮箱等，系统会对输入信息进行实时校验，确保用户名的唯一性以及密码的强度符合安全标准，防止弱密码带来的安全隐患。

新用户可以通过提供基本信息（如用户名、密码）完成注册。注册信息将存储在用户表中，用于后续验证，从而实现注册功能。已注册的用户提供用户名和密码进行身份验证。通过验证后，用户可以进入系统主界面，从而实现登录功能。

在权限管理方面，系统采用了基于角色的访问控制（RBAC）模型。系统预定义了多种角色，如管理员、普通用户等，不同角色被赋予不同的权限集合。管理员拥有最高权限，可进行系统全局配置，包括添加或删除用户、修改系统参数、管理视频资源库等操作。普通用户则主要具备视频上传、浏览个人上传视频及查看部分公共视频的权限。通过这种精细的权限划分，既能保障系统核心功能与数据的安全，又能满足不同用户在业务场景中的多样化操作需求。

**(2)视频管理功能**

视频管理功能涵盖了视频从采集录入到存储归档的全生命周期管理。在视频采集与录入阶段，系统支持多种视频源接入方式。对于实时监控视频流，可通过标准的视频传输协议（如RTSP、RTMP）接入，确保视频数据的实时性与稳定性。同时，也支持本地视频文件的批量上传，用户只需将本地视频文件按照系统规定的格式（如MP4、AVI等常见视频格式）整理好，即可通过简洁的上传界面完成批量导入操作。在上传过程中，系统会自动读取视频的元数据信息，如视频分辨率、时长、帧率等，并生成视频缩略图，方便用户快速识别与管理视频内容。

视频存储方面，系统采用分布式存储架构，将视频数据分散存储在多个存储节点上，提高存储的可靠性与扩展性。存储节点可根据实际需求选择不同类型的存储介质，如高速固态硬盘（SSD）用于存储近期频繁访问的热点视频，以保证快速的数据读取速度；大容量机械硬盘用于存储历史视频数据，实现成本与性能的平衡。系统还具备视频数据备份机制，定期将重要视频数据备份到异地存储中心，防止因本地存储故障导致的数据丢失。在视频检索与查询功能上，用户可通过多种条件对视频进行精准检索，如按视频名称、拍摄时间范围、视频标签（用户自定义标签或系统自动生成的标签，如“室内场景”“交通监控”等）、视频内容关键词（通过视频内容分析技术提取的关键信息）等条件进行组合查询，快速定位到所需视频资源。

**(3)视频异常检测功能**

视频异常检测功能是系统的核心智能分析模块，借助先进的人工智能算法与深度学习模型实现。系统采用基于视觉语言大模型（VLM）的异常检测技术，能够对视频内容进行逐帧分析。与传统的基于深度学习的视频异常检测方法相比，我们的基于VLM的方法可以实现提示词设计或者用户自定义规则实现免训练的视频异常检测，从事实现对多种不同异常检测场景的快速适应。如公共场所的打架斗殴、人员跌倒、物品遗留；交通场景中的车辆碰撞、逆行、闯红灯等异常情况等。

在实际检测过程中，用户首先在系统中选定已上传的MP4格式源视频，触发异常检测流程；接着，系统自动调用视觉语言大模型（VLM），对视频内容进行逐帧序列分析，基于跨模态学习能力识别画面中的异常行为模式；模型通过分析连续帧特征，精准定位异常片段的起始与结束时间，生成衡量异常程度的分数（0.0~1.0），并赋予标准化标签（如“斗殴”“物品遗留”）及自然语言描述（如“画面中央区域出现多人聚集推搡”）；最后，系统将包含时间范围、分数、标签、描述的检测结果存入异常片段表（AbnormalClip），同时记录检测时间、模型版本等处理日志，便于后续查询与审计。

**(4)GIS可视化功能**

GIS可视化功能为视频监控数据赋予了地理空间维度，实现了视频资源与地理信息的深度融合。在地图展示方面，系统集成了主流的GIS地图引擎，如ArcGIS、Mapbox等，能够加载高精度的电子地图，包括卫星影像地图、矢量地图等。地图支持多种交互操作，如缩放、平移、旋转，方便用户从宏观到微观视角查看地理区域。用户可在地图上直观地看到各个视频监控设备的分布位置，监控设备以特定图标进行标注，不同类型的监控设备（如固定摄像头、移动摄像头）采用不同的图标样式，便于区分。点击监控设备图标，可弹出详细信息窗口，展示设备名称、设备ID、所属区域、视频实时预览链接（若设备在线且支持实时预览）等信息。

在视频与地理信息关联方面，系统通过为每个监控设备绑定准确的地理坐标信息，建立起视频与地理位置的对应关系。当用户在地图上选择某个区域时，系统能够自动筛选出该区域内所有监控设备拍摄的视频资源，并以列表或缩略图形式展示，方便用户快速查看该区域的视频监控情况。同时，对于视频中发生的异常事件，GIS可视化系统能够在地图上实时标注异常事件发生的位置，以醒目的图标或颜色标识，如红色图标表示紧急异常事件，黄色图标表示一般异常事件。点击异常事件标注点，可查看异常事件详情，包括异常发生时间、异常类型、相关视频片段等信息。系统还支持基于地理信息的统计分析功能，如统计某个区域内一段时间内异常事件的发生次数、分布密度，生成热力图、柱状图等可视化图表，帮助用户直观了解区域内的安全态势与风险分布情况，为决策制定提供数据支持。

## 1.4 作业范围

本次作业范围为“基于GIS地图的可视化智慧安防系统”的全新开发作业，需从零构建集视频监控、异常检测与地理信息可视化于一体的完整系统，实现安防数据的智能化管理与空间化呈现。

本次开发负责的各功能模块在产品整体中形成有机协同体系：用户管理模块作为系统入口，实现用户身份验证与权限控制，为其他模块提供安全访问基础；视频管理模块负责视频数据的全生命周期管理，包括上传、存储与检索，是系统的数据基础；视频异常检测模块通过集成视觉语言大模型（VLM），实现对视频内容的智能分析，生成异常事件的关键信息，是系统的核心智能单元；GIS可视化模块则将视频数据与地理空间信息深度融合，通过地图渲染、路径规划等功能，提升安防事件的空间感知与应急响应能力，是系统的特色功能延伸。各模块通过数据接口与业务逻辑紧密联动，共同支撑系统的完整功能。

开发工程分工方面，我们小组承担全系统的设计与开发任务：在需求分析阶段，完成用户需求调研与功能规格定义，明确系统需实现用户登录、视频上传、异常检测、GIS地图标注等核心功能；在设计阶段，构建基于Django框架的MVT架构，设计用户表、监控表、视频表、异常片段表及GIS相关表（如设备位置表、地理围栏表）的数据库结构，规划GIS地图引擎（如Leaflet.js）与VLM模型的集成方案；在开发阶段，实现用户管理、视频管理、异常检测、GIS可视化等模块的代码开发，包括前端界面交互（如地图操作、视频预览）、后端业务逻辑（如异常检测算法、空间查询接口）及数据库操作；在测试阶段，完成功能测试、性能测试（如视频处理速度、地图加载效率）、安全测试（如数据加密、权限验证），确保系统稳定运行。顾客需提供业务场景的具体需求描述（如安防区域划分、应急资源分布）、地理底图数据（如园区或城市地图）及监控设备部署规划，参与需求评审与验收测试，确认系统功能与业务场景的匹配度。

本次开发将系统拆分为以下独立组件进行作业：1.用户管理组件，实现用户注册、登录、权限分配功能，采用Django内置的用户认证框架，确保用户数据安全；2.视频管理组件，包括视频上传接口、文件存储模块、元数据管理功能，支持MP4格式视频的批量上传与分布式存储；3.异常检测组件，集成VLM模型，对视频帧序列进行分析，生成异常片段的时间范围、标签、描述及异常分数，输出结构化检测结果；4.GIS基础组件，基于Leaflet.js实现地图加载、缩放、标注功能，开发地理数据接口支持设备位置与异常事件的地图可视化；5.地理数据管理组件，设计GIS\_Device表（存储设备坐标）与GIS\_Fence表（存储地理围栏数据），实现设备位置的录入、查询及围栏区域的绘制与管理；6.应急响应组件，集成路径规划算法，根据异常事件位置与应急资源坐标生成最短路径，提供可视化路线指引；7.系统集成组件，开发模块间数据交互接口（如异常检测结果与GIS标注的联动接口），实现视频数据与地理信息的关联查询及界面交互联动（如点击地图标注播放对应视频片段）。通过各组件的协同开发与集成测试，确保系统功能的完整性与操作的流畅性。

## 1.5 作业形态

新开发改造项目

## 1.6 设计输入与设计输出

**(1)设计输入**

设计输入包含项目开发所需的基础条件与约束信息。

需求层面，涵盖用户提出的核心功能需求，主要是用户登录注册、视频上传与管理、异常检测及GIS地图可视化等，同时包含非功能需求，如系统响应时间需满足视频上传后1分钟内启动检测、支持2-5用户并发操作，以及数据安全与权限控制要求。

技术标准与法规方面，遵循《数据安全法》《个人信息保护法》等法规，确保用户数据加密存储与传输安全；参考GIS数据格式标准（如GeoJSON）、视频编码标准（H.264）及Django框架开发规范。

现有资源基于原视频监控系统的数据库结构（用户表、监控表等）、Django项目架构及VLM模型能力，结合GIS扩展需求纳入地理空间数据，如监控设备坐标、地理围栏多边形信息。

软硬件环境约束包括兼容Inteli9/RTX3060及以上硬件配置、Windows/Linux系统、MySQL8.0/PostGIS数据库及Python3.8+与Django4.x框架。

**(2)设计输出**

设计输出为项目各阶段的可交付成果，确保需求转化为实际系统能力。

技术方案层面，形成系统架构设计，在DjangoMVT架构基础上新增GIS模块，采用Leaflet.js/MapboxGLJS实现地图渲染，通过DjangoGIS处理空间数据，构建“视频分析+地理可视化”双核心架构；完成数据库扩展设计，新增GIS\_Fence（地理围栏表）、GIS\_Device（设备地理位置表），定义坐标点、多边形等空间字段及表间关联关系。

功能实现成果包括用户管理模块的权限控制逻辑与界面原型，视频管理模块的上传存储及检索功能代码，异常检测模块的VLM模型集成与检测结果结构化输出（如含时间范围、标签、描述的JSON报告），GIS模块的地图标注、路径规划及热力图交互组件。

开发文档方面，输出接口设计文档，规范前端与后端交互接口的参数与返回格式；制定测试用例集，覆盖功能、性能、安全等多维度测试，如用户登录验证、地图加载速度测试、数据加密传输验证。

部署与运维文档包含环境搭建手册（Python依赖安装、数据库配置、项目启动步骤）、静态与媒体文件配置指南，以及运维手册，明确系统监控指标、故障排查流程（如视频上传失败定位、GIS地图加载异常处理）及版本更新机制。

## 1.7 开发规模

|  |  |
| --- | --- |
| 功能模块 | 开发规模(KL) |
| 用户管理模块 | 1 |
| 视频管理模块 | 4 |
| 异常检测模块 | 4 |
| GIS可视化模块 | 2 |
| 合计 | 10 |

# 2．体制

## 2.1 开发体制

本项目开发由“异常检测部门”负责组织实施，团队成员分工如下：

|  |  |  |
| --- | --- | --- |
| 职能 | 姓名 | 负责业务（应具体化） |
| 项目责任人 | 徐祺翔 | 整体把握项目质量、进度；审核技术方案与提交成果 |
| 项目组长 | 徐祺翔 | 任务协调与进度跟踪；编写项目文档（如计划书、设计书）；参与后端开发 |
| 开发人员1 | 丁宗参 | 负责GIS模块功能开发与测试；地图服务API接入与空间分析调试 |
| 开发人员2 | 张浩东 | 实现视频异常检测模块；集成大模型接口（Qwen-VL）；撰写测试用例与报告 |
| 开发人员3 | 唐鑫臣 | 负责数据库建模与前后端接口逻辑；辅助用户界面功能实现与优化 |

## 2.2 委托方联络窗口

|  |  |  |  |
| --- | --- | --- | --- |
| 职能 | 公司·部门 | 姓名 | 联系办法 |
| 负责人 |  |  | 电话： |
| FAX： |
| E-mail： |
| 技术联络窗口 |  |  | 电话： |
| FAX： |
| E-mail： |

# 3．整体作业计划

## 3.1 工数预测

**(1)预测依据**

[1]项目复杂度

融合GIS地图可视化、视觉语言大模型（VLM）异常检测、Django前后端开发，需跨领域技术整合，存在多模块协同开发难点。

[2]团队规模与技术水平

项目团队共4人，具备Python开发、深度学习模型部署及GIS开发经验，但VLM模型优化与GIS空间分析功能存在技术挑战。

[3]硬件与软件环境

需配置NVIDIAGPU（RTX3060+）、MySQL/PostGIS数据库、Leaflet/Mapbox引擎，环境搭建与调试需额外工时。

[4]风险因素

VLM模型推理效率可能受视频分辨率影响，需预留模型优化时间；

GIS地图服务API密钥申请及跨平台兼容性（如移动端适配）可能导致延期；

多用户并发测试时可能出现数据库性能瓶颈，需提前进行压力测试。

**(2)工数预测表**

|  |  |  |  |
| --- | --- | --- | --- |
| 任务阶段 | 任务内容 | 工数（人月） | 时间分配依据 |
| 需求分析与设计 | 需求调研、功能规格书编写、数据库设计、GIS功能规划 | 1.00 | 基于文档复杂度及团队4人协作，需1周完成核心设计 |
| 环境搭建与开发 | Django框架搭建、MySQL/PostGIS配置、VLM模型部署（Qwen2-VL）、Leaflet地图集成 | 1.50 | 包含GPU驱动调试、模型权重下载（10GB）、地图服务API对接等耗时任务 |
| 模块开发 | 用户认证模块、视频上传与检测模块、GIS可视化模块、异常片段管理模块 | 3.00 | 按4人分工，每个模块平均1.5人月，含接口联调与单元测试 |
| 集成测试与优化 | 多模块联调、压力测试（16路视频并发）、模型精度优化（如误检率降低）、GIS热力图性能调优 | 1.00 | 需模拟真实安防场景测试，处理视频断流重连、GPU显存溢出等异常情况 |
| 文档与交付 | 用户手册编写、API文档生成、操作视频录制、部署包制作（Docker镜像） | 0.60 | 4人协作完成，需4天左右整理所有交付物 |
| 合计 | - | 7.10 | - |

**(3)备选方案**

[1]增加开发人员（+2人）：工数可压缩至5.00人月，重点分配至模块开发与测试阶段，加速并行开发；

[3]精简功能（如暂不支持移动端适配）：工数降至6.00人月，优先保障核心功能（视频检测）交付。

## 3.2 开发模型

**敏捷开发（Scrum）**

理由：(1)项目涉及多技术栈迭代（如GIS引擎更新），需灵活应对需求变更；

(2)采用短周期冲刺（2周/迭代），每周召开站会同步进度，及时调整开发方向；

(3)核心里程碑可分阶段交付，确保项目可控性。

## 3.3 日程

总日程表（2025.5-2025.6）

|  |  |  |  |
| --- | --- | --- | --- |
| 阶段 | 起止时间 | Milestone | 交付物/关键节点 |
| 需求与设计 | 2025.05.15-2025.05.22 | 完成需求规格书与数据库设计 | 《02\_FDesign.docx》终稿、ER模型图 |
| 环境搭建 | 2025.05.22-2025.05.24 | 完成开发环境部署 | Django项目框架、GPU推理环境、MapboxAPI密钥配置 |
| 模块开发-冲刺1 | 2025.05.24-2025.05.31 | 用户认证与视频上传功能上线 | 登录/注册接口、视频上传模块（支持MP4/AVI） |
| 模块开发-冲刺2 | 2025.05.25-2025.05.31 | 异常检测模块初步运行 | 基于Qwen2-VL的基础检测功能，输出异常标签与分数 |
| 模块开发-冲刺3 | 2025.05.31-2025.06.04 | GIS地图基础功能上线 | 设备标注、异常点位可视化、地图缩放/拖拽交互 |
| 集成测试 | 2025.06.04-2025.06.07 | 完成端到端测试与优化 | 支持16路视频并发检测、GIS热力图性能达标（渲染延迟＜500ms） |
| 用户验收 | 2025.06.07-2025.06.10 | 中间版本提交与评审 | 可运行Demo（含用户管理、检测、GIS展示） |
| 文档与部署 | 2025.06.10-2025.06.14 | 完成最终交付物 | 安装手册、API文档、Docker部署包 |
| 项目交付 | 2025.06.15 | 交付评审会议与正式交付 | 系统正式版本 |

**关键节点说明：**

(1)设计审查会议：2025.05.22，评审需求规格书与数据库设计；

(2)中间版本提交：2025.06.10，提交可演示的Alpha版本；

(3)交付评审会议：2025.06.15，确认功能验收与部署方案。

## 3.4 配置管理

**内容：**

(1)管理对象：代码（Django后端、前端JS/CSS）、模型文件（Qwen2-VL权重）、配置文件（.env、settings.py）、文档（需求与功能规格书/设计说明书、用户手册）；

(2)变更控制：需求变更需通过评审会确认，代码变更需经2人以上审核。

**方法：**

(1)版本控制：采用Git分支管理（主分支/开发分支/功能分支），每次提交需附带Issue编号；

(2)环境隔离：开发/测试/生产环境分离，通过DockerCompose管理容器化部署。

**工具：**

(1)代码托管：GitHub；

(2)持续集成：GitHubActions，自动运行单元测试与代码质量检查（如Flake8）；

(3)文档管理：Confluence，实时同步需求与设计变更。

**软件产品版本的标识方法：**

(1)版本格式：主版本号.次版本号.修订号

主版本号：重大功能变更（如新增VLM模型版本，GIS引擎升级）；

次版本号：新增模块或功能扩展（如支持新异常类型、GIS路径规划）；

修订号：bug修复或性能优化（如数据库查询优化、地图加载速度提升）。

(2)示例：V1.0.0（初始版本）、V1.1.0（新增GIS热力图）、V1.1.1（修复视频上传bug）。

# 4．进度管理

## 4.1 进度报告

为确保项目按照预期目标有序推进，项目组将采用以下方式开展进度报告：

**报告方式：**通过电子邮件（E-mail）提交书面进度报告，并在必要时组织电话会议或在线会议（如腾讯会议、Zoom）进行口头说明和答疑。

**使用语言：**中文为主要沟通语言，必要时可附英文摘要或术语对照，确保专业术语在技术资料中准确表达。

**报告时间：**每周五下午提交一次进度报告，内容需在周五中午前完成内部确认。

**报告内容：**(1)本周已完成的主要工作项（按模块或子任务划分）；

(2)当前开发或设计中遇到的问题与风险提示；

(3)下周计划安排及资源需求；

(4)对其他成员或组间接口的依赖事项；

(5)附录：代码提交记录、模型调试日志、设计文档更新情况等（如有）。

**发布范围：**项目核心成员（包括技术负责人、接口人）、指导教师，必要时抄送其他相关人员（如GIS模块协作者、数据库接口开发人员）。

## 4.2 技术讨论

为确保项目开发中技术问题能够快速响应与协同解决，项目组将安排定期的技术讨论，并在必要时随时响应：

**主要手段：**(1)日常问题沟通优先使用QQ小组与E-mail；

(2)跨模块接口或需求变化讨论通过线上会议进行；

(3)对于代码、流程图或复杂系统架构变更，也可使用白板或手写笔记扫描件（PDF）通过群共享方式分发。

**使用语言：**中文为主要讨论语言，必要时在讨论纪要中补充英文术语说明，以便于资料后续归档。

**讨论时间：**技术会议安排为每周三晚上20:00在线会议，每次会议由不同模块负责人轮流主持，议题提前1天提交；若遇紧急需求或接口问题，可由项目负责人发起临时会议。

# 5．质量计划（此部分实验课可省略）

## 5.1 适用标准和质量记录

为确保“基于GIS地图的可视化智慧安防系统”项目在开发与交付过程中满足既定的质量标准，项目组制定如下质量目标与控制指标：

**(1)适用标准**

采用《软件工程—产品质量要求与评价》（ISO/IEC 25010）中提出的质量特性，包括功能适用性、可靠性、可用性、安全性、性能效率、可维护性等。

遵循《GB/T 16260-2006软件工程产品质量》（等同ISO/IEC 9126）作为验收测试参考依据。

单元测试覆盖率、集成测试通过率、BUG密度等评价参考《CMMI Dev》模型三级实践要求。

**(2)质量目标与说明**

|  |  |
| --- | --- |
| 项目 | 指标值/目标说明 |
| 潜在BUG预测值 | 每千行代码不超过3个BUG（基于项目采用Django框架+Python，BUG密度中等） |
| BUG摘除率 | 每阶段至少摘除90%以上已知BUG（从测试到上线，每次迭代至少完成一轮回归） |
| 测试密度 | 单元测试≥80%；集成测试≥95%；功能测试全覆盖异常处理路径 |
| 生产效率 | 每人每周平均提交≥500行有效代码（不含注释/生成代码），评审合格率≥85% |

**(3)质量记录方式**

所有BUG通过项目组Jira（或语雀任务列表）进行记录与跟踪；

所有测试报告、缺陷清单、修复记录均归档于项目文档中心；

每次阶段验收前需完成质量审核会议并形成《阶段质量评估表》。

## 5.2 工程移行基准

本项目采用自定义工程阶段划分（需求分析、概要设计、详细设计、编码实现、系统集成测试、上线部署），各阶段的完成判定基准如下：

|  |  |
| --- | --- |
| 工程阶段 | 判断基准 |
| 需求分析 | 完成《需求与功能设计规格书》评审，所有功能需求条目经用户确认并冻结，无重大遗漏或变更。 |
| 概要设计 | 系统架构图、模块划分、主要数据结构、接口关系形成文档，评审通过，关键模块设计达成一致意见。 |
| 详细设计 | 所有模块的内部逻辑流程、关键类/函数/数据库结构定义完成，文档可追溯需求并通过技术评审。 |
| 编码实现 | 模块代码完成80%以上，核心模块单元测试通过率≥85%，通过CodeReview并符合命名与结构规范。 |
| 系统集成测试 | 所有模块已完成集成，GIS地图联动、视频上传与异常检测、用户交互流程完整，测试通过率≥95%。 |
| 上线部署与验收 | 项目部署至目标环境，主要功能可用，Bug等级A类为0，用户验收测试完成并签署验收确认单。 |

# 6．交付物品

## 6.1 交付物品一览

本项目交付物品涵盖文档类成果、源代码、部署系统、可视化模块、数据库及测试资料等，详见下表：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 项目编号 | 成果物名称及版本 | 交付方式及媒体 | 预定交货期 | 使用语言 |
| 01 | 《需求与功能设计规格书》V1.0 | 电子文档（PDF/Word） | 2025年6月15日 | 中文 |
| 02 | 《详细设计说明书》V1.0 | 电子文档（PDF/Word） | 2025年6月15日 | 中文 |
| 03 | 源代码包V1.0（前端+后端+模型接口） | Git仓库压缩包/USB介质/网盘链接 | 2025年6月15日 | Python/HTML/CSS |
| 04 | Django+MySQL数据库结构及初始数据脚本 | SQL脚本文件/文本格式 | 2025年6月15日 | SQL |
| 05 | 视频异常检测模块（Qwen-VL接入版） | 模型部署包（PyTorch模型+API接口） | 2025年6月15日 | Python |
| 06 | GIS地图可视化功能模块 | 前端代码包+配置文件 | 2025年6月15日 | JavaScript |
| 07 | 测试用例集与测试报告V1.0 | PDF测试报告+Excel用例文档 | 2025年6月15日 | 中文 |
| 08 | 部署说明书&用户操作手册V1.0 | PDF文档/电子手册 | 2025年6月15日 | 中文 |
| 09 | 系统演示视频 | MP4视频文件 | 2025年6月15日 | 中文解说 |
| 10 | 项目总结报告V1.0 | PDF文档 | 2025年6月15日 | 中文 |

# 7．作业环境

## 7.1 开发环境

* 硬件

处理器为Intel Core i5或AMD Ryzen 5及以上多核处理器，内存8GB及以上，256GB及以上固态硬盘，显卡需16GB显存以上。

* 软件

集成开发环境用PyCharm，版本控制系统为Git，数据库管理工具是MySQL Workbench，Python环境为3.8版本，虚拟环境管理工具使用Anaconda。

* OS

支持Windows 10/11、Ubuntu等操作系统。

## 7.2 运行环境

* 硬件

处理器采用Intel Xeon或AMD EPYC系列，内存16GB及以上，512GB及以上企业级固态硬盘，显卡需16GB显存以上，网络连接高速稳定。

* OS

可选用Linux（如Ubuntu Server、CentOS）或Windows Server操作系统。

* 软件

Web服务器用Nginx或Apache，应用服务器采用Gunicorn或uWSGI，数据库为MySQL，Python环境为3.x版本。

## 7.3 测试环境

* 硬件

处理器为Intel Corei3或AMD Ryzen3及以上，内存4GB及以上，128GB及以上固态硬盘，显卡需16GB显存以上。

* 软件

测试框架使用Django自带测试框架，自动化测试工具为Selenium，性能测试工具是JMeter，数据库管理工具为MySQLWorkbench。

* OS

可使用Windows10/11或Ubuntu操作系统。

# 8．技术学习计划

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **作业者** | **必需的技术、知识** | **学习方法** | **学习时间** | **OUTPUT** |
| 产品经理 | Axure原型设计工具，用户体验设计原则，市场需求分析方法，敏捷开发流程 | 学习Axure官方教程，阅读软件工程课件等资料，参与产品需求调研项目，与开发团队沟通交流 | 每周8小时 | 能够用Axure绘制高保真原型，准确分析市场需求，熟悉敏捷开发流程并有效推进产品开发 |
| 前端开发 | HTML5、CSS3新特性，JavaScript高级编程，Django中前端框架的实现 | 观看B站优质前端教学视频，阅读对应官方文档，参与开源项目实践，在CodePen上做练习 | 第1-4周，每周8小时 | 能够用HTML5和CSS3搭建美观页面，用JavaScript实现交互逻辑，熟练使用一种主流前端框架开发组件 |
| 后端开发 | Django5.0.8框架，Python3.x高级特性，MySQL数据库高级管理与优化，Nginx服务器配置 | 阅读Django、Python官方文档，参考开源项目代码，在本地服务器进行配置实践，参加技术交流社群讨论 | 第1-5周，每周10小时 | 能基于Django开发复杂后端业务逻辑，优化Python代码性能，熟练管理MySQL数据库并进行性能调优，完成Nginx复杂配置 |
| 算法开发 | 视觉语言大模型的使用，索引增强生成（RAG）技术，Agent技术 | 阅读阿里云官方文档，掌握大模型的API调用以及相应的技术使用 | 第1-3周，每周6小时 | 能基于API调用结合RAG，Agent技术实现项目的主要功能 |

# 9．风险管理（此部分实验课可省略）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **风险** | **发生可能性**  **(高/中/低)** | **影响范围及**  **影响程度** | **可能的预防方法** | **风险发生时预计采取的对策** |
| 硬件设备采购延迟或不符合要求 | 中 | 影响开发进度，可能导致整体项目交付延期。若硬件性能不达标，会影响软件运行和测试效果。 | 提前与供应商签订详细合同，明确交付时间和质量标准；定期跟进采购进度。 | 与供应商协商加快供货，或寻找替代供应商；对已到货但不符合要求的设备，及时退换货，调整开发计划，优先开展不依赖该硬件的任务。 |
| 软件环境配置出现问题 | 中 | 导致开发、测试无法正常进行，影响项目进度。 | 在项目开始前，进行软件环境的预搭建和测试；制定详细的环境配置文档。 | 组织技术人员分析问题原因，参考配置文档和技术论坛解决方案进行修复；若问题复杂，考虑重新搭建环境，并做好数据备份。 |
| Django框架相关技术难题 | 中 | 阻碍功能开发进度，影响代码质量和稳定性。 | 开发团队提前进行技术储备，  学习相关文档和教程；  建立技术交流群，遇到问题及时讨论。 | 邀请技术专家进行指导，查阅官方文档和开源社区解决方案；必要时调整功能实现方案。 |
| 数据库性能优化困难 | 中 | 影响系统响应速度和数据处理能力，降低用户体验。 | 在数据库设计阶段进行合理规划，采用性能优化工具进行模拟测试。 | 分析数据库性能瓶颈，调整数据库结构、索引等；采用缓存技术减轻数据库压力；必要时寻求数据库专家协助。 |
| 团队成员技术水平差异导致进度不一致 | 中 | 部分模块开发进度缓慢，影响整体集成和测试。 | 项目开始前评估成员技术水平，进行针对性培训；合理分配任务，技术强的成员帮带技术稍弱的成员。 | 对进度落后的成员提供额外指导和资源支持；调整任务分配，必要时从其他项目组借调人员协助。 |

# 10．对委托方的要求事项

**(1)资料、数据和程序清单及提供时间：**

在项目启动初期（项目开始第1周内），委托方需提供项目相关的业务流程说明文档、现有数据（如历史订单数据、用户信息数据等）以及可能涉及的已有程序清单。这些资料将为项目的开发和设计提供基础依据，确保开发方向符合实际业务需求。

**(2)作业HW・SW环境费用支持承诺：**

委托方需承诺对项目所需的硬件（HW）和软件（SW）环境费用给予支持，明确资金支持的范围和额度，以便我方能够及时采购和搭建稳定、合适的开发及运行环境，保障项目顺利推进。

**(3)顾客分担的作业及其完成时间：**

委托方需在项目需求调研阶段（第2-3周），配合完成业务需求梳理和确认工作，及时提供业务细节和需求变更信息。在项目测试阶段（第7-8周），协助进行业务场景的真实测试，提供实际业务操作反馈。

**(4)review和review结果的提示时间：**

在项目的关键节点，如原型设计完成（第4周）、功能开发完成（第6周）、集成测试完成（第8周），委托方需在收到相关交付物后的5个工作日内进行review，并及时反馈review结果，以便我方根据反馈进行调整和优化。

**(5)设计变更或设计追加要求的期限：**

在项目整体计划的前期阶段（前5周内），委托方若有设计变更或设计追加要求，需及时提出。随着项目推进，后期变更可能会对成本和工期产生较大影响，因此请尽量在前期明确需求，避免后期不必要的变更。

**(6)使用顾客机器的办法：**

若项目需要使用委托方机器，需委托方提前安排技术人员与我方对接，提供机器的使用权限、网络配置、系统环境等相关信息。双方需共同制定机器使用规范和安全保障措施，确保机器使用过程中数据安全和系统稳定。

**(7)顾客进行验收检查时要求检查的内容：**

验收检查时，委托方需检查项目功能是否符合需求规格说明书，包括各业务流程的完整性、数据处理的准确性、界面操作的便捷性等；检查系统性能，如响应时间、并发处理能力等是否满足预期；检查系统的稳定性和兼容性，在不同环境和设备上能否正常运行。

**(8)顾客验收检查结果通知的内容和期限：**

委托方在验收检查完成后的5个工作日内，需向我方通知验收结果。通知内容应包括验收通过或不通过的结论，若不通过需详细列出存在的问题和缺陷，以便我方及时进行整改和完善。

**保护页**