**项目启动报告**

**项目名称： 入侵监测系统**

**小组成员： 周芷平、张钰杰、王俊铭、**

**王地、付柏逢、张业瑞**

**负责人姓名： 周芷平**

目录

1.引言

[1.1编写目的 2](#_Toc13643288)

[1.2项目标识 2](#_Toc13643289)

[1.3术语解释 2](#_Toc13643290)

[2.项目启动 2](#_Toc13643292)

[2.1项目简介 2](#_Toc13643293)

[2.2项目目的 3](#_Toc13643294)

[2.3项目预期 3](#_Toc13643296)

[3.团队成员及分工 3](#_Toc13643295)

[4.项目计划表 3](#_Toc13643302)

[5.风险评估 3](#_Toc13643302)

1. **引言**
   1. **编写目的**

编写本文档的目的是为实施工程和管理软件项目制定合理的计划，为实施和管理软件项目活动提供基础，并根据软件项目资源、约束条件和能力向软件项目的客户提出承诺。

* 1. **项目标识**

项目名称：入侵监测系统

* 1. **术语解释**

|  |  |  |
| --- | --- | --- |
| **序号** | **术语或缩写** | **定义** |
|  | 人脸识别 | 人脸识别，是基于人的脸部特征信息进行身份识别的一种生物识别技术。用摄像机或摄像头采集含有人脸的图像或视频流，并自动在图像中检测和跟踪人脸，进而对检测到的人脸进行识别的一系列相关技术，通常也叫做人像识别、面部识别。 |
|  | 入侵检测 | **系统判定人员是否入侵到监控区域，判定人员是否为未登记人员** |

1. **项目启动**
   1. **项目简介**

该项目旨在开发出功能较为完整的人员入侵检测系统，基于实时的视频流，利用图像识别功能，自动检测视频中的运动人员，检测是否为未登记人员，如是则留存记录且发出警报，管理员可增加和查看入侵记录。包括后端数据处理和前端web页面实时显示结果, 可以应用在小区入侵监测、智能家居的安防等领域。

* 1. **项目目的**

掌握前沿的软件项目管理方法和策略，对实际的软件工程问题进行较为准确的可行性分析与论证；学习将离散数学、数据结构、算法设计与分析等课程掌握的理论方法应用于实训项目，并按照工程化思维方式完成软件开发全部流程；了解软件工程职业道德及履行的社会责任、提高自身对实际问题的准确分析和理解能力；掌握软件测试相关理论，使用合适的测试方法和工具完成软件测试，并能分析评价软件产品周期中项目存在的隐患。进一步体会团队项目开发中担任的不同角色及承担相应工作和责任，包括分析、设计、实现、测试、文档、版本控制、项目负责人等，提升沟通和协作能力。学会清晰地陈述实际项目的需求分析、研发过程、实训结果或结论，准确表述软件相关技术，并进行技术交流。

* 1. **项目预期**

实现实时监控显示、人脸识别判断监控内人员是否为未登记人员、入侵警报、增加和查看入侵记录等入侵检测相关的基础需求；实现用户信息的管理、用户权限管理、用户登录注册等用户管理相关的需求。在以上基础上，尝试添加短信认证、入侵监测事件回放、任意形状入侵区域等其他功能。

1. **团队成员及分工**

项目组长：周芷平

项目成员：王俊铭，张业瑞，张钰杰、王地、付柏逢。

**任务分工：**

1. 张钰杰：后端开发
2. 张业瑞：前端开发、文档编写
3. 王地：监控端开发
4. 王俊铭：前端开发、文档编写
5. 付柏逢：后端开发
6. 周芷平：监控端开发
7. **项目计划表**

|  |  |  |  |
| --- | --- | --- | --- |
| 模块 | 子模块 | 工作量（占比） | 描述 |
| 实时监控 | 入侵判别 | 15% | 系统判定人员是否入侵到监控区域 |
| 入侵记录 |  | 10% | 系统将入侵图像或视频保存至服务器 |
| 人脸检测 |  | 10% | 系统对视频或图像提取人脸信息 |
| 信息采集 | 获取图像 | 10% | 系统获取人脸图像 |
| 建立模型 | 10% | 为每个人建立自己的特征模型 |
| 人脸识别 |  | 10% | 系统对不同的人脸图像进行区分识别 |
| 用户信息管理 |  | 5% | 管理用户信息 |
| 用户权限管理 |  | 5% | 为用户分配权限 |
| 身份信息管理 |  | 5% | 对入侵人员进行信息管理 |
| Web界面 |  | 15% | 用户的主要交互界面和实时监控的现实界面。 |
| 数据库设计 |  | 5% | 为系统存储信息提供 |

1. **风险评估**

技术风险，管理风险和其他风险。

* + 技术风险：部分成员对开发技术使用不熟练、三端开发代码合并不一定顺利。
  + 管理风险：计划不合理，可能不能按照计划执行。
  + 其他风险：丢失数据、不同成员编程逻辑冲突等。