基于情感分析的智慧养老系统

开发文档

段智超 16301004

储泽栋 16301002

于佳莉 16301055

路杨 16301042

毕宇航 16301030

2019.7.2

基于情感分析的智慧养老系统

需求分析文档

段智超 16301004

储泽栋 16301002

于佳莉 16301055

路杨 16301042

毕宇航 16301030

2019.7.2

目录

1 引言

1.1 编写目的

本文档阐明项目的背景、意义、应用场景以及项目的创新性等内容。对项目的可行性进行研究，对需求进行分析，明确需求。为后续开发提供文档支持。

1.2 项目概述

基于情感分析的智慧养老系统可以实时显示摄像头拍摄到的画面，通过人工智能算法实时分析老人的情感、分析是否有人摔倒、分析是否有人闯入禁止区域、分析老人是否有和义工互动并追踪义工、分析是否有陌生人出现并追踪陌生人。

1.3 项目背景

随着我国人口老龄化问题的日趋严峻，对专业化养老的需求与日俱增。

由于人口红利消失，适龄劳动力的比例在逐年降低，护理人员的数量相对老人的数量仍然偏少，且护理强度的增大导致护理人员流失，专业护理人员严重不足。

大部分养老机构信息化水平较低，现代化服务装备普遍欠缺。该系统可以摄像头实时拍摄到的画面，通过人工智能算法实时分析，对老人、工作人员、义工进行检测，将信息技术和老人、养老机构、护理人员紧紧地联系在一起，更快速有效地服务老年人。

1.4 术语和缩略语

MTCNN	Multi-task Cascaded Convolutional Networks
CNN	Convolutional Neural Networks
项目干系人	参与该项目工作的个体和组织，或由于项目的实施与项目的成功，其利益会直接或间接地受到正面或负面影响的个人和组织
用例	提供了一个或多个场景，该场景说明了系统是如何和最终用户或其它系统互动
用例图	用例图是外部用户（被称为参与者）所能观察到的系统功能的模型图

1.5 参考资料

［1］基于情感分析的智慧养老系统课程指导手册

1.6 项目定位

1.6.1 应用场景

该系统仅供系统管理员使用。系统管理员使用该系统不仅可以管理老人、工作人员和义工的信息，还可以实时得到警报，如陌生人入侵、陌生人追踪。
该系统本应配备 5 个摄像头。
鉴于项目开发环境有限，仅能采用电脑前置摄像头，所以采用两个摄像头。一个摄像头用于追踪陌生人、检测是否有人摔倒、监控义工与老人的交互。另一个摄像头用于检测禁止区域入侵。

1.6.2 项目干系人

使用人：系统管理员
监测对象：养老院老人、工作人员、义工
地点：养老院
指导老师：张健
开发人员：段智超、储泽栋、毕宇航、路杨、于佳莉
第三方：华为云实践平台

1.7 项目目标

在 2019 年 7 月 10 日之前完成开发基于情感分析的智慧养老系统，实现对老人的情绪变化监测，陌生人闯入检测，禁止区域闯入检测，摔倒检测，老人与义工互动监测，监测画面实时显示。本系统支持双摄像头同时监控，并在陌生人闯入时发出警报。

1.8 项目价值

智能养老系统主要是依托智能科技来实现老年人生活上的辅助，减轻了养老的社会负担，节约了时间、人力成本，提高的工作效率。同时，可以第一时间获警示，提高了安全性和科学性。

2 开发计划

2.1 最终交付物

- 产品:

Web 端交互界面：一个可视化的图形界面，用于给管理员进行与系统的交互，用户可使用的一切功能都在这个界面上体现。
计算机视觉监控：系统对摄像头获取图像处理的模块，该部分对获取图片进行识别，从而获取老人的生理、心里状态等信息，并将这些信息传达给工作人员。
后台管理系统：系统管理人员用来对用户信息管理的系统。

- 文档:

需求分析文档：深入细致的调研和分析，准确理解用户和项目的功能、性能、可靠性等具体要求，将用户非形式的需求表述转化为完整的需求定义，从而确定系统必须做什么。
概要设计文档：把需求分析得到的系统扩展用例图转换为软件结构和数据结构。
测试计划文档：对整个信息系统应用软件组装测试和确认测试，有效预防计划的风险，保障计划的顺利实施。

2.2 运行环境

操作系统：Windows XP、Windows7 及以上，Linux，Unix，macOS
浏览器：Firefox，Microsoft edge，Safari，Chrome

2.3 验收标准

1） 在项目范围内的所有功能都能够实现。
2） 保证系统本系统全年全天 24 小时工作。
3） 系统管理员可以在无人指导下使用系统 3 次后熟悉系统。
4） 系统管理员能够在 10 秒内成功登录系统。
5） 智慧养老系统实时显示监控画面，在网速在 1MB/S 及以上时允许有最多 3 秒的延迟。
6） 智慧养老系统实时显示检测报表内容，在网速在 1MB/S 及以上时允许有最多 3 秒的延迟。
7） 智慧养老系统陌生人检测成功率达到 90%。
8） 智慧养老系统禁止区域闯入检测成功率达到 90%。
9） 智慧养老系统摔倒检测成功率达到 90%。
10） 智慧养老系统老人情绪变化识别率到达 70%。
11） 智慧养老系统老人与义工互动情况检测成功率达到 90%。
12） 智慧养老系统老人识别的情绪为：愤怒、厌恶、恐惧、高兴、伤心、惊讶和无情绪。

2.4 进度安排

任务名称	工期	开始时间	完成时间
启动阶段	**1 个工作日**	**2019/7/1**	**2019/7/1**
确定项目目标、范围、干系人	0.5 个工作日	2019/7/1	2019/7/1
需求分析	0.5 个工作日	2019/7/1	2019/7/1
准备项目计划	0.5 个工作日	2019/7/1	2019/7/1

设计阶段	1.5 个工作日	2019/7/2	2019/7/3
概要设计	0.5 个工作日	2019/7/2	2019/7/2
详细设计	1 个工作日	2019/7/2	2019/7/3
模块设计	0.5 个工作日	2019/7/2	2019/7/2
接口设计	0.5 个工作日	2019/7/3	2019/7/3
开发阶段	5.5 个工作日	2019/7/3	2019/7/8
安装配置开发环境	0.5 个工作日	2019/7/3	2019/7/3
项目编程	5 个工作日	2019/7/4	2019/7/8
前端编程	4 个工作日	2019/7/4	2019/7/7
人员管理系统	1 个工作日	2019/7/4	2019/7/4
实时监测系统	2 个工作日	2019/7/5	2019/7/6
图像管理系统	1 个工作日	2019/7/7	2019/7/7
后端编程	5 个工作日	2019/7/4	2019/7/8
人员管理系统	1 个工作日	2019/7/4	2019/7/4
实时监测系统	3 个工作日	2019/7/5	2019/7/7
图像管理系统	1 个工作日	2019/7/8	2019/7/8
编写开发报告	1 个工作日	2019/7/4	2019/7/4
测试阶段	6 个工作日	2019/7/4	2019/7/9
单元测试	5 个工作日	2019/7/4	2019/7/8
接口测试	0.5 个工作日	2019/7/9	2019/7/9
集成测试	0.5 个工作日	2019/7/9	2019/7/9
编写报告	5.5 个工作日	2019/7/4	2019/7/9
测试报告	5 个工作日	2019/7/4	2019/7/8
用户手册	0.5 个工作日	2019/7/9	2019/7/9
关闭阶段	0.5 个工作日	2019/7/9	2019/7/9
准备 PPT	0.5 个工作日	2019/7/9	2019/7/9

3 可行性分析

3.1 市场可行性分析

1） 人口老龄化问题突出。
2） 养老服务机构服务质量参差不起
3） 养老服务平台发展不足

3.2 技术可行性分析

3.2.1 简述

1. 硬件设备，开发技术
   现在的计算机各方面的技术都非常成熟，相对来说，开发此系统的技术要求比较简单，所以需要的功能都可以实现，因此在技术上是可行的。
2. 系统管理与维护
   需要一定的系统管理和维护的专业人员，在这方面可以雇佣第三方进行维护，也可以通过培训原来的技术人员成为新的需要的技术人员。

3.2.2 技术要素

1. 人脸检测：MTCNN
2. 人脸对比
3. 关键点检测：CNN
4. 表情分类：CNN
5. 服务器搭建：Django
6. Web 前端：Vue.js
7. Web 前端组件：elementUI

3.3 资源可行性分析

1）时间：2019 年 7 月 1 日至 2019 年 7 月 9 日，共计九天开发时间。
2）设备：五台计算机，华为云实践平台
3）地点：逸夫楼 606
4）人员：
   a) 指导老师：张健及助教
   b) 开发人员：段智超、储泽栋、毕宇航、路杨、于佳莉

4 数据描述

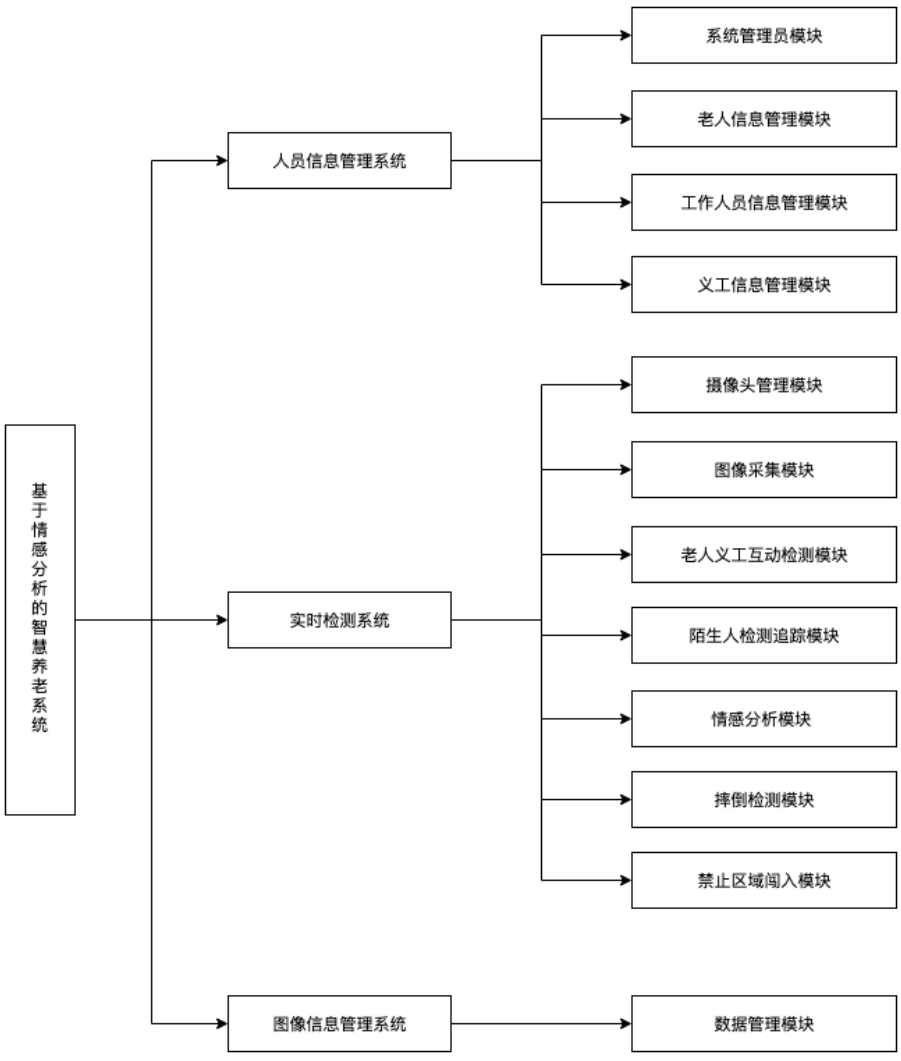
4.1 数据采集

1. 人脸检测训练数据集：WIDER Face
2. 人脸关键点检测训练数据集：Celeba
3. 人体关键点检测数据训练集：COCO 2018 Keypoint Detection Task
4. 表情分类训练数据集：Kaggle Emotion recognition
5. 老人信息：系统管理员录入老人的各项基本信息以及老人的头像

6. 义工信息：系统管理员录入义工的各项基本信息以及义工的头像
7. 工作人员信息：系统管理员录入工作人员的各项基本信息以及工作人员的头像

5 功能性需求分析

5.1 功能模块结构图



5.2 功能模块描述

功能模块	子功能	功能描述	优先级
系统管理员模块	登陆	系统管理员登陆该系统	4
	修改密码	系统管理员可以修改个人密码	4
	维护信息	管理员为系统使用者，可以对显示在该系统上的信息进行相应的操作	2
老人信息管理模块	老人信息增加	录入老人姓名、性别、出生日期、身份证号等信息	3
	老人信息修改	可以对老人姓名、性别、出生日期、身份证号等一项或多项信息进行修改	3
	老人信息查询	可以通过姓名、出生日期、身份证号等对老人进行查询	3
	老人信息删除	删除一条或者多条老人信息	3
	老人头像设置	上传头像	2
	统计分析	对老人的义工互动、情感、是否摔倒、是否传入禁止区域进行分析统计，以图表显示	1
工作人员信息管理模块	工作人员信息增加	录入工作人员姓名、性别、出生日期、身份证号、工作职位等信息	3
	工作人员信息修改	可以对工作人员姓名、性别、出生日期、身份证号、工作职位等一项或多项信息进行修改	3
	工作人员信息查询	可以通过姓名、出生日期、身份证号、工作职位等对工作人员进行查询	3
	工作人员信息删除	删除一条或者多条工作人员信息	3
	工作人员头像设置	上传头像	2
	统计分析	统计分析是否有工作人员摔倒、是否传入禁止区域	1
义工信息管理模块	义工信息增加	录入义工姓名、性别、出生日期、身份证号等信息	3

	义工信息修改	可以对工作人员姓名、性别、出生日期、身份证号等一项或多项信息进行修改	3
	义工信息查询	可以通过姓名、出生日期、身份证号、工作职位等对义工进行查询	3
	义工信息删除	可以删除一条或者多条义工信息	3
	义工头像设置	上传头像	2
	统计分析	统计分析是否有义工摔倒、是否传入禁止区域	1
摄像头管理模块	实时监控	用户可以在网页上实时查看摄像头监控的画面	1
数据管理模块	显示陌生人检测情况	以图表显示闯入的陌生人的照片及检测到陌生人的时间	2
	显示老人情感检测情况	以图表显示老人姓名、老人情感状况、发生情感变化的时间以及图片	2
	显示禁止区域入侵检测情况	以图表显示闯入禁止区域的人员信息和时间	2
	显示摔倒检测情况	以图表显示摔倒人员的信息、发生时间以及图片	2
老人义工互动检测模块	互动检测	检测义工与老人的互动，并且判断义工与哪位老人在互动，以图表显示义工与老人信息、发生时间及图片	1
陌生人检测追踪模块	陌生人检测	检测陌生人的存在	1
	报警	若发现有陌生人，发出报警信息	2
	保存检测情况	存储陌生人截图及时间	3
情感分析模块	情感检测	检测老人情感变化：愤怒、厌恶、恐惧、高兴、伤心、惊讶	1
	保存检测情况	若发现老人发生愤怒、厌恶、恐惧、高兴、伤心、惊讶等情感变化，保存截图及时间	2
摔倒检测模块	摔倒检测	检测是否有人摔倒	1
	保存检测情况	存储摔倒截图及时间	2

禁止区域闯入检测模块	闯入检测	检测是否有人传入禁止区域	1
	保存检测情况	存储闯入人截图及时间	2

6 非功能性需求分析

6.1 性能需求

6.1.1 响应时间

为满足用户的交互要求，以及数据传到服务器，及时完成数据库操作与反馈，在网速流畅的前提下，每项点击的响应时间应小于 1 秒，页面之间跳转的响应时间应小于 2 秒。

6.1.2 负载需求

要求本系统同时在线用户数目可以达到 1000 人。

6.1.3 压力需求

不断增加系统在线用户数目直至同时在线用户达到 1000 人，系统不会发生瘫痪或崩溃，系统的反应能力，响应时间仍然满足前项响应时间的性能需求。

6.2 兼容性

本系统保证浏览器、操作系统、数据库兼容

在测试过程中选择不同浏览器，同一浏览器的不同操作系统版本，不同分辨率的设备，不同网络类型进行测试，考虑是否兼容的问题

避免出现以下情况

1) 程序运行过程中闪退

2) 部分控件显示不完整或者功能失效

3) 屏幕显示异常

4) 图片展示不全

6.3 稳定性

本系统保证在用户使用期间保持稳定，不出现使用中途闪退等情况，在网络情况较好的情况下可以保证持续 24 小时在线。

6.4 易用性

系统的易用性直接关系其市场占有率，易用性要求主要体现在系统的以下方面：

① 导航提示要求明确；导航要求方便、准确、快捷；

② 界面要求布局合理，简洁明了，符合一般网页的制作要求;

③ 某一功能的实现不超过 4 次页面跳转；

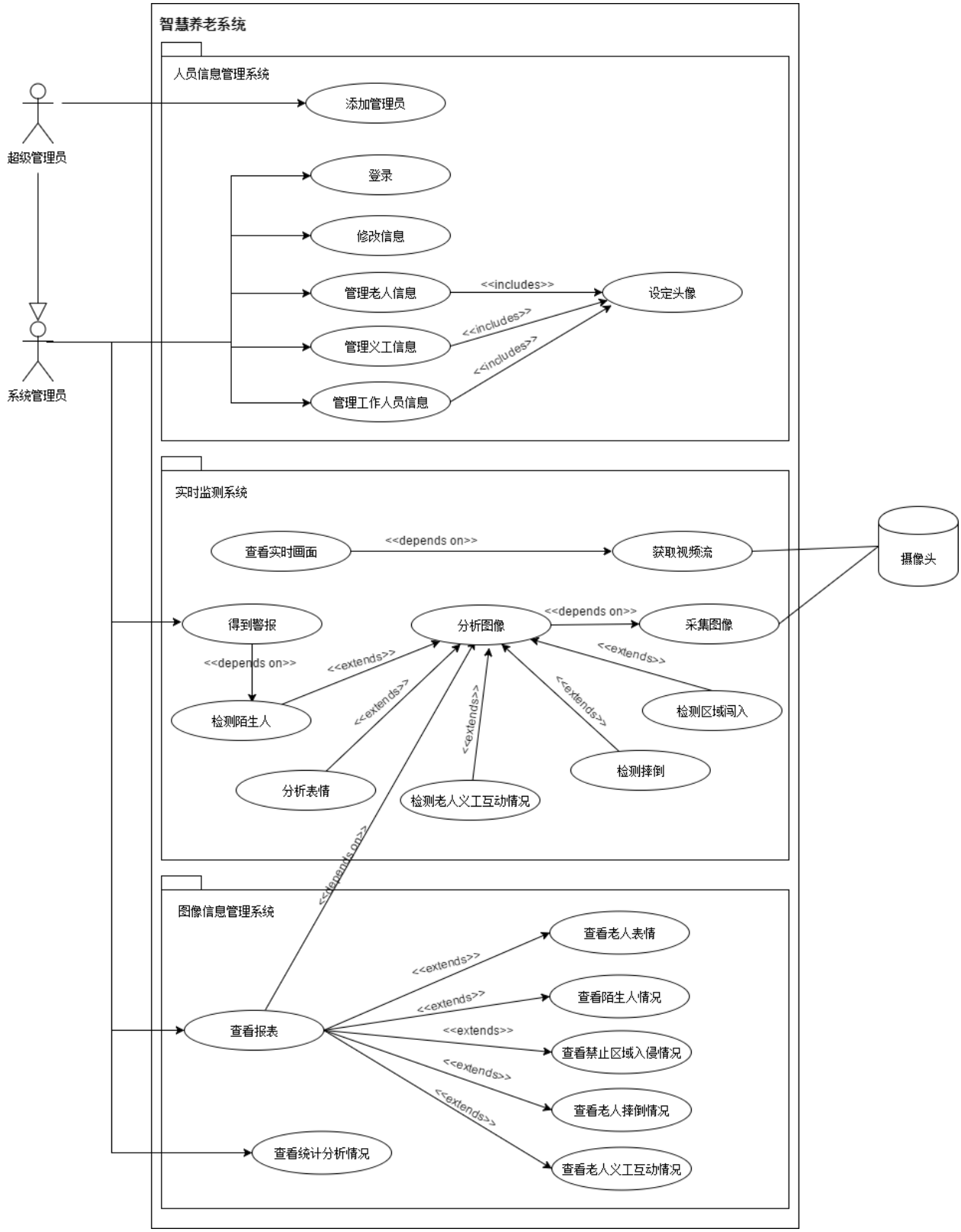
6.5 容错性

本系统要求具有较高的容错性，在用户出现错误操作时，系统应对用户进行提醒并告知用户正确操作流程，不会因操作问题而导致系统无法运行或崩溃。

6.6 可维护性

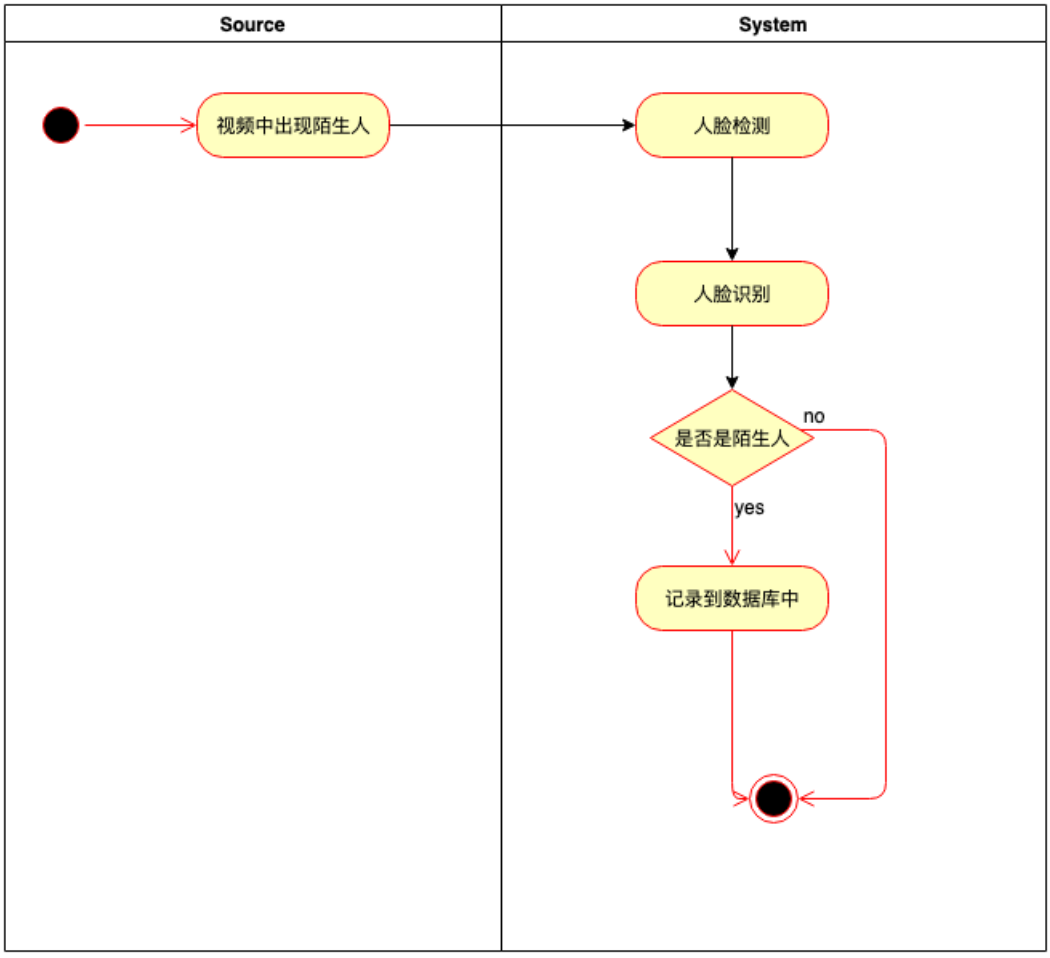
该系统的每个模块应具有一定的独立性，便于改进。具有日志文件，在系统数据丢失，或系统故障时可根据日志文件进行恢复。该系统还应具有完整详细的文档以及健全的代码注释，以方便后期的维护。

7 用例图


智慧养老系统
人员信息管理系统
超级管理员
添加管理员
登录
修改信息
管理老人信息
<<includes>>
设定头像
管理义工信息
<<includes>>
<<includes>>
管理工作人员信息
系统管理员
实时监测系统
查看实时画面
<<depends on>>
获取视频流
摄像头
得到警报
分析图像
<<depends on>>
采集图像
<<depends on>>
<<extends>>
<<extends>>
检测陌生人
<<extends>>
检测区域闯入
<<extends>>
<<extends>>
分析表情
检测老人义工互动情况
检测摔倒
<<depends on>>
图像信息管理系统
查看报表
<<extends>>
查看老人表情
<<extends>>
查看陌生人情况
<<extends>>
查看禁止区域入侵情况
<<extends>>
查看老人摔倒情况
<<extends>>
查看老人义工互动情况
查看统计分析情况

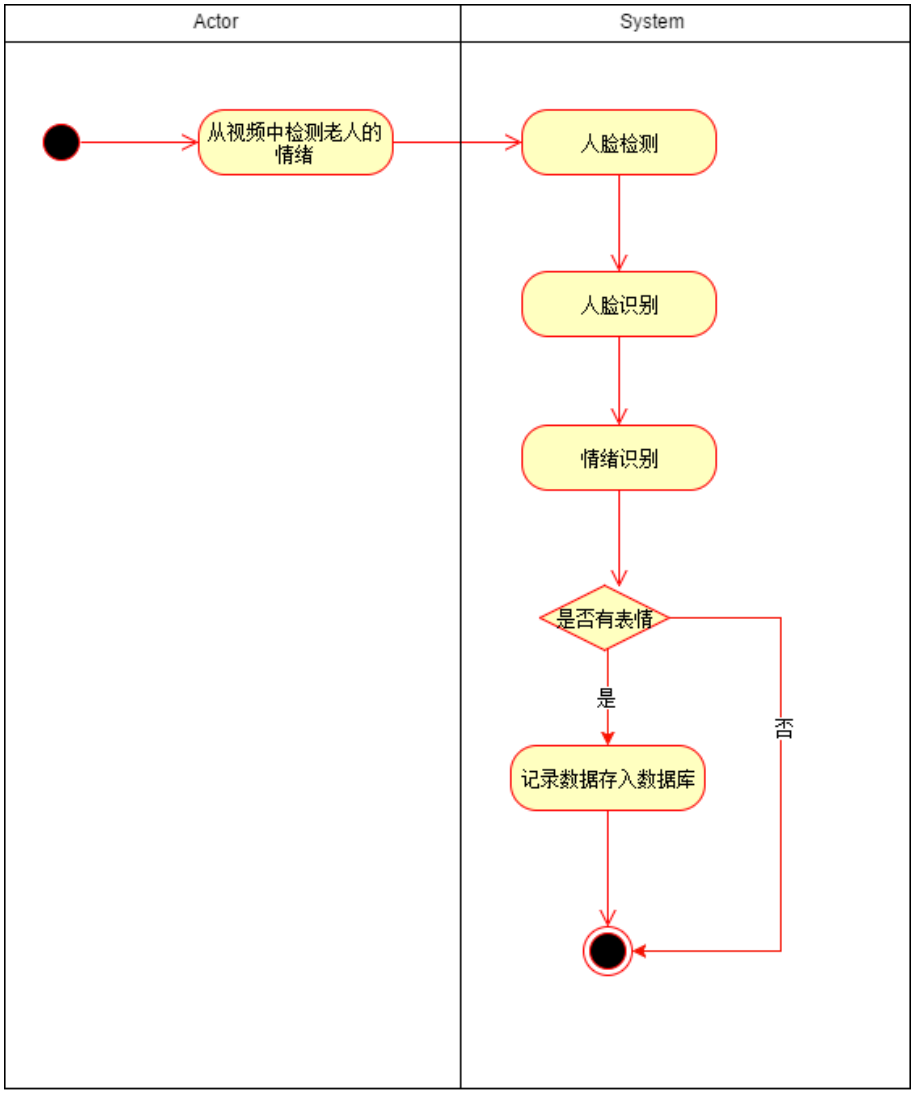
8 核心模块用例规约

Use Case Type	系统分析	
Use Case ID	SAUC-001	
Use Case Name	陌生人检测	
Use Case Description	该用例描述，在摄像头检测区域内，检测摄像头拍摄到的人脸中，是否有在数据库存储的老人、工作人员、义工以外的陌生人，若检测到该陌生人，保存图片并发出警报。	
Priority	高	
Major Business Actors	system	
Major System Actors	system	
Precondition	1、 摄像头功能正常，能够形成一个清晰的视频 2、 网络状况良好，无较明显的网络延迟	
Trigger	实时监测	
Typical Upload Events	**Actor Actions**	**System Responses**
	Step 1: 实时传输图片帧给服务器 **Step 6:** 显示报警信息	**Step 2:** 系统检测分析图片中是否有陌生人 **Step 3**: 系统检测到图片中有陌生人，将图片保存 **Step 4:** 向数据库中插入记录 **Step 5:** 向前端发送报警信息
Alternate Actions	**Alt-Step 3**: 检测图片中无异常，继续检测	
Conclusion	该用例为实时检测，仅当服务器被关闭或者网络间断时，该用例才会结束	
Postconditions	N/A	
Business Rules		
Implementation Constraints and Specifications	陌生人检测准确率应高于 90% 从陌生人出现到发出警报的时间间隔不应该超过 1s	
Assumptions	图片为 jpg 格式，RGB 通道	
Open Issues	N/A	

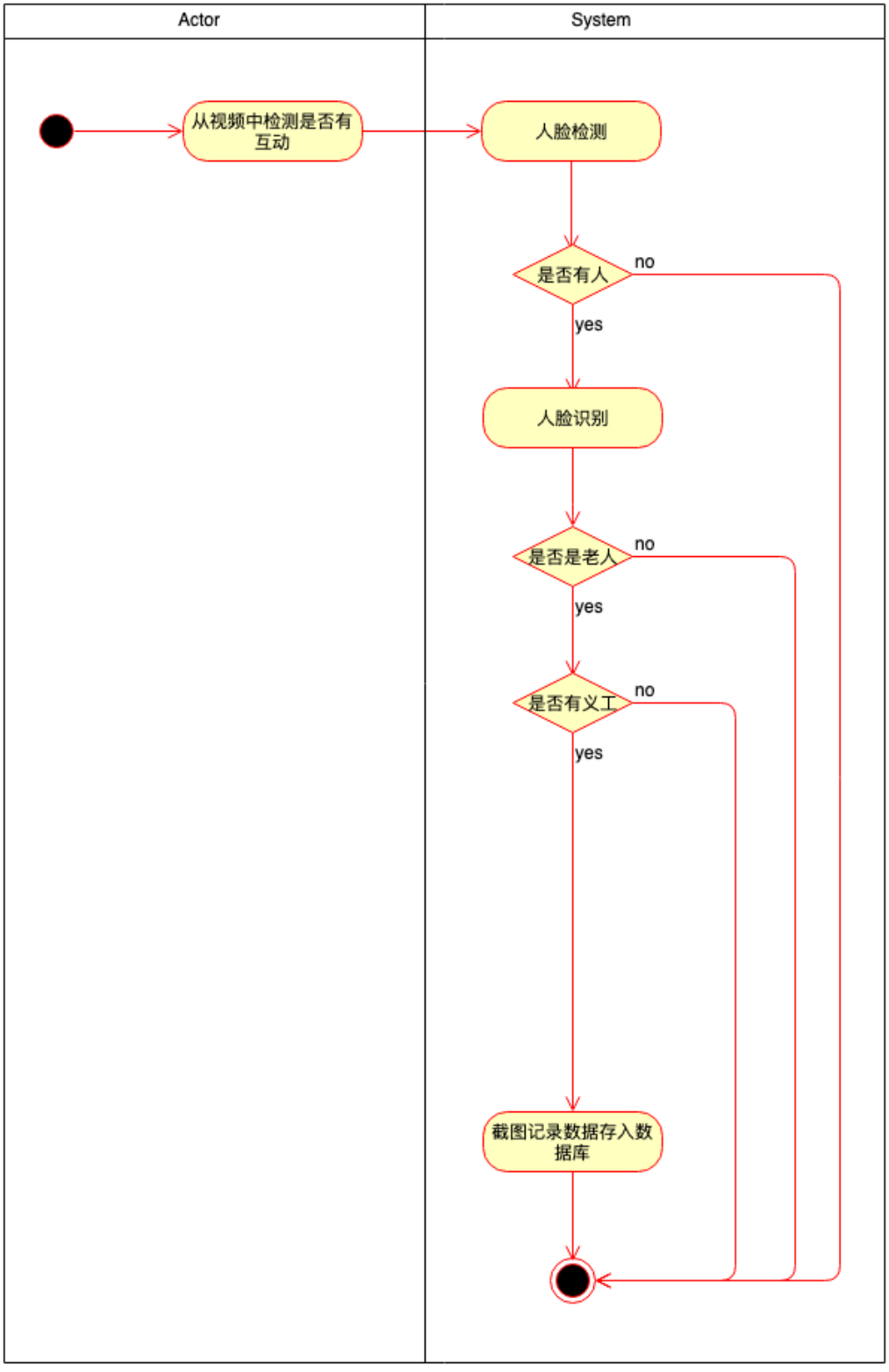


Use Case Type	系统分析	
Use Case ID	SAUC-002	
Use Case Name	情感分析	
Use Case Description	该用例描述，在摄像头检测区域内，检测摄像头拍摄到的老人中，老人出现以下的情绪变化：愤怒、厌恶、恐惧、高兴、伤心、惊讶，保存截图，并且在数据库中记录	
Priority	高	
Major Business Actors	system	
Major System Actors	system	
Precondition	1. 摄像头功能正常，能够形成一个清晰的视频 2. 网络状况良好，无较明显的网络延迟	
Trigger	实时监测	
Typical Upload Events	**Actor Actions**	**System Responses**
	Step 1: 实时传输图片帧给服务器	**Step 2:** 系统检测分析图片是否有老人出现 **Step 3**: 系统检测有老人的图片

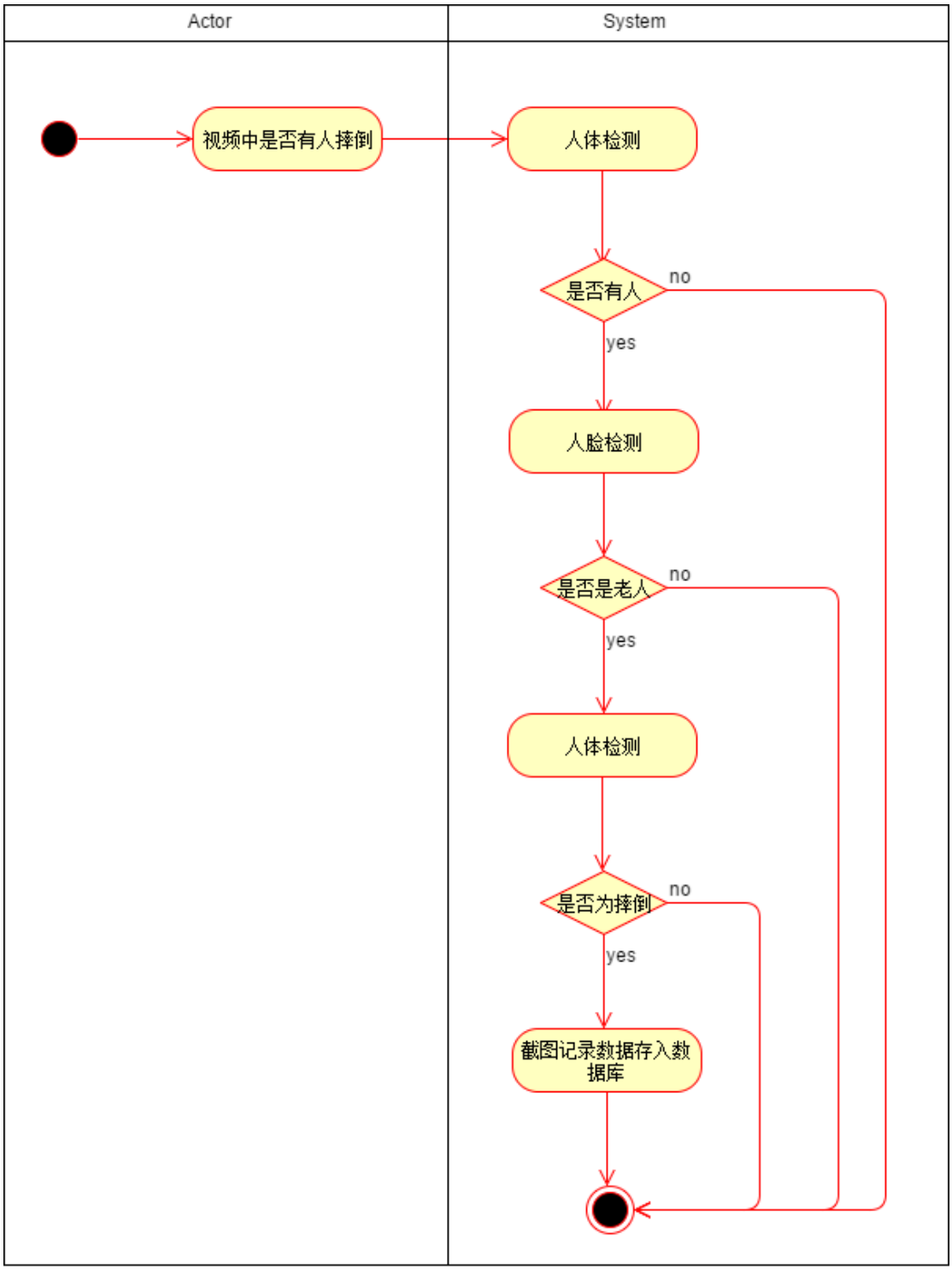
		是否情感出现 **Step 4:** 分析情感类型，并且向数据库记录 **Step 5:** 将该条记录返回
Alternate Actions	**Alt-Step 3**: 检测图片中无情感，继续检测	
Conclusion	该用例为实时检测，仅当服务器被关闭或者网络间断时，该用例才会结束	
Postconditions	N/A	
Business Rules		
Implementation Constraints and Specifications	分类准确度应达到 70%以上 出现情感变化时，到界面出现记录，时间不超过 1s	
Assumptions	图片为 jpg 格式，RGB 通道	
Open Issues	N/A	



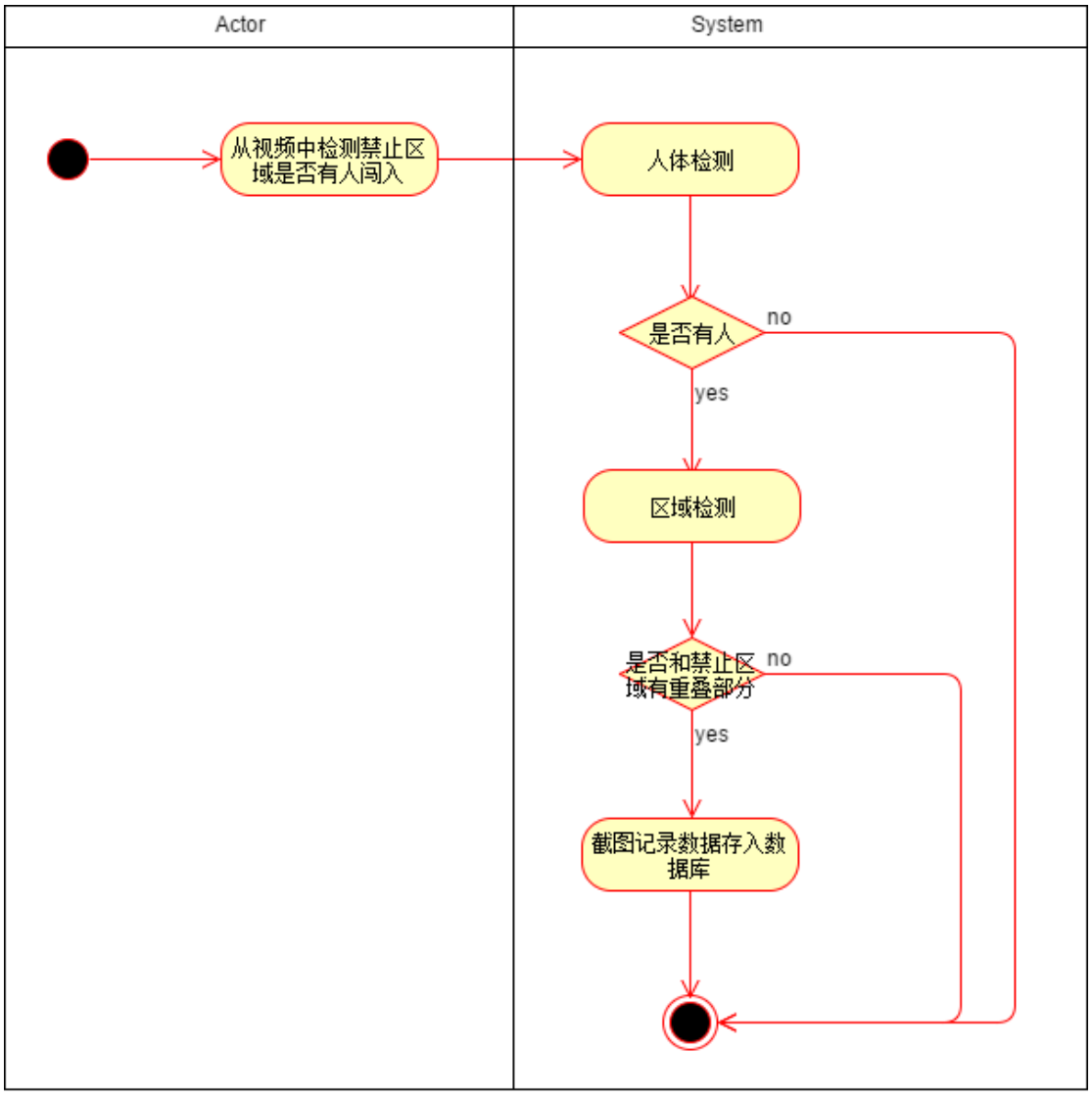
Use Case Type	系统分析	
Use Case ID	SAUC-003	
Use Case Name	检测老人义工互动	
Use Case Description	该用例描述，在摄像头检测区域内，检测摄像头拍摄到的老人中，与义工是非发生互动，若发生互动，与哪位老人互动。发生互动后，保存截图，并记录到数据库中。	
Priority	高	
Major Business Actors	system	
Major System Actors	system	
Precondition	1. 摄像头功能正常，能够形成一个清晰的视频 2. 网络状况良好，无较明显的网络延迟	
Trigger	实时监测	
Typical Upload Events	**Actor Actions**	**System Responses**
	Step 1: 实时传输图片帧给服务器	**Step 2:** 系统检测分析图片中的人脸 **Step 3:** 系统检测分析图片中老人 **Step 4**: 系统检测分析图片中的义工 **Step 5:** 同时存在老人和义工，则发生互动，截图保存 **Step 6:** 将该条记录记录到数据库中
Alternate Actions	**Alt-Step 4**: 检测图片中无互动，继续检测	
Conclusion	该用例为实时检测，仅当服务器被关闭或者网络间断时，该用例才会结束	
Postconditions	N/A	
Business Rules		
Implementation Constraints and Specifications	分类准确度应达到 90%以上 出现互动时，到界面出现记录，时间不超过 1s	
Assumptions	图片为 jpg 格式，RGB 通道	
Open Issues	N/A	


Actor
System
从视频中检测是否有互动
人脸检测
是否有人
no
yes
人脸识别
是否是老人
no
yes
是否有义工
no
yes
截图记录数据存入数据库

Use Case Type	系统分析	
Use Case ID	SAUC-004	
Use Case Name	检测摔倒	
Use Case Description	该用例描述，在摄像头检测区域内，检测摄像头拍摄到有人摔倒就记录当时的画面并存入数据库。	
Priority	高	
Major Business Actors	system	
Major System Actors	system	
Precondition	3、 摄像头功能正常，能够形成一个清晰的视频 4、 网络状况良好，无较明显的网络延迟	
Trigger	实时监测	
Typical Upload Events	**Actor Actions**	**System Responses**
	Step 1: 实时传输图片帧给服务器	**Step 2:** 系统检测分析图片中是否有人摔倒 **Step 3**: 系统检测到图片中有人摔倒，将图片保存 **Step 4:** 向数据库中插入记录
Alternate Actions	**Alt-Step 3**: 检测图片中无人摔倒，继续检测	
Conclusion	该用例为实时检测，仅当服务器被关闭或者网络间断时，该用例才会结束	
Postconditions	N/A	
Business Rules		
Implementation Constraints and Specifications	摔倒检测准确率应高于 90% 检测到有人摔倒的延迟不超过 1s	
Assumptions	图片为 jpg 格式，RGB 通道	
Open Issues	N/A	


Actor
System
视频中是否有人摔倒
人体检测
是否有人
no
yes
人脸检测
是否是老人
no
yes
人体检测
是否为摔倒
no
yes
截图记录数据存入数据库

Use Case Type	系统分析	
Use Case ID	SAUC-005	
Use Case Name	检测禁止区域闯入情况	
Use Case Description	该用例描述，在摄像头检测区域内，检测摄像头拍摄到有人闯入指定的禁止区域就记录当时的画面并存入数据库，。	
Priority	高	
Major Business Actors	system	
Major System Actors	system	
Precondition	5、 摄像头功能正常，能够形成一个清晰的视频 6、 网络状况良好，无较明显的网络延迟	
Trigger	实时监测	
Typical Upload Events	**Actor Actions**	**System Responses**
	Step 1: 实时传输图片帧给服务器	**Step 2:** 系统检测分析图片中是否有人闯入禁止区域 **Step 3**: 系统检测到图片中有人闯入禁止区域，将图片保存 **Step 4:** 向数据库中插入记录
Alternate Actions	**Alt-Step 3**: 检测图片中无异常，继续检测	
Conclusion	该用例为实时检测，仅当服务器被关闭或者网络间断时，该用例才会结束	
Postconditions	N/A	
Business Rules		
Implementation Constraints and Specifications	禁止区域闯入检测准确率应高于 90% 检测到有人闯入的延迟不超过 1s	
Assumptions	图片为 jpg 格式，RGB 通道	
Open Issues	N/A	


Actor
System
从视频中检测禁止区域是否有人闯入
人体检测
是否有人
no
yes
区域检测
是否和禁止区域有重叠部分
no
yes
截图记录数据存入数据库

基于情感分析的智慧养老系统
概要设计

指导老师：张健 孔祥远

小组成员：段智超 储泽栋 毕宇航 路杨 于佳莉

2019.7.9

目录

1. 引言

1.1 编写目的

本文档的目的是阐述基于情感分析的智慧养老系统的概要设计。本概要设计说明书编写的目的在于全面说明该系统第一阶段中的设计考虑，包括系统架构、基本处理流程、程序系统的组织结构、模块划分，为程序的详细设计提供基础。

1.2 项目背景

随着我国人口老龄化问题的日趋严峻，对专业化养老的需求与日俱增。

由于人口红利消失，适龄劳动力的比例在逐年降低，护理人员的数量相对老人的数量仍然偏少，且护理强度的增大导致护理人员流失，专业护理人员严重不足。

大部分养老机构信息化水平较低，现代化服务装备普遍欠缺。该系统可以摄像头实时拍摄到的画面，通过人工智能算法实时分析，对老人、工作人员、义工进行检测，将信息技术和老人、养老机构、护理人员紧紧地联系在一起，更快速有效地服务老年人。

本项目为 web 应用，与其他系统无关联 ，由段智超负责，由段智超小组进行开发。

1.3 定义

实体—联系图（E-R图）	包含实体（即数据对象）、关系和属性。作为用户与分析员之间有效交流的工具。
流程图	由一些特定意义的图形、流程线及简要的文字说明构成，能清晰明确地表示程序的运行过程
系统架构图	是一个系统的草图，用于展示系统分层结构

1.4 参考资料

A. 项目开发计划
B. 项目需求文档
C. 华为云实践平台基于情感分析的智慧养老系统指导手册
D. 《系统分析与设计方法》Jeffrey L. Whitten Lonnie D. Bentley 著

2. 任务概述

2.1 目标

于 2019 年 7 月 1 日启动项目，在 2019 年 7 月 10 日之前完成开发基于情感分析的智慧养老系统，实现对老人的情绪变化监测，陌生人闯入检测，禁止区域闯入检测，摔倒检测，老人与义工互动监测，监测画面实时显示，老人信息管理，义工信息管理，工作人员信息管理以及实时图表显示。本系统支持双摄像头同时监控，并在陌生人闯入时发出警报。

2.2 运行环境

硬件环境：

本软件运行需要计算机局域网与广域网的支持。在同一个局域网与该计算机建立连接，并且传输数据。

软件环境：

操作系统：Windows XP、Windows7 及以上，Linux，Unix，macOS

浏览器：Firefox，Microsoft edge，Safari，Chrome

2.3 需求概述

Web 端：

1. 信息管理模块
   a) 系统管理员信息
      i. 超级管理员：添加普通管理员，设置初始密码以及所有普通管理员能进行的操作。
      ii. 普通管理员：登录系统，修改密码以及使用该系统的所有功能。
   b) 老人信息
      可以对老人的信息进行录入、修改、查询、删除和头像的设定。
   c) 工作人员信息
      可以对工作人员的信息进行录入、修改、查询、删除和头像的设定。
   d) 义工信息
      可以对义工的信息进行录入、修改、查询、删除和头像的设定。
2. 检测事件图像信息管理模块
   a) 统计分析
      对老人的情绪随时间变化的统计，对陌生人入侵次数随时间变化的统计，对老人与义工交互情况的统计，对老人情绪的总体情况的分析统计
   b) 事件检测信息管理
      i. 对摔倒事件的主要信息的查看以及详细信息的查看
      ii. 对老人与义工交互事件的主要信息的查看以及详细信息的查看
      iii. 对老人情绪分析事件的主要信息的查看以及详细信息的查看

iv. 对禁止区域入侵事件的主要信息的查看以及详细信息的查看
v. 对陌生人闯入事件的主要信息的查看以及详细信息的查看

3. 监控画面实时显示

计算机视觉：

1. 老人义工互动检测模块

一旦检测到有义工和老人在交互，将截图保存到硬盘，并实时插入一条数据到数据库。一旦数据库有了新数据，Web 端的实时图表会立即呈现。

2. 陌生人检测模块

一旦检测到有陌生人，将截图保存到硬盘，并实时插入一条数据到数据库。一旦数据库有了新数据，Web 端的实时图表会立即呈现。

3. 老人情感分析模块

分析老人的情绪，将截图保存到硬盘，并实时插入一条数据到数据库。一旦数据库有了新数据，Web 端的实时图表会立即呈现。

4. 禁止区域闯入检测模块

一旦检测到禁止区域有人闯入，将截图保存到硬盘，并实时插入一条数据到数据库。一旦数据库有了新数据，Web 端的实时图表会立即呈现。

5. 摔倒检测模块

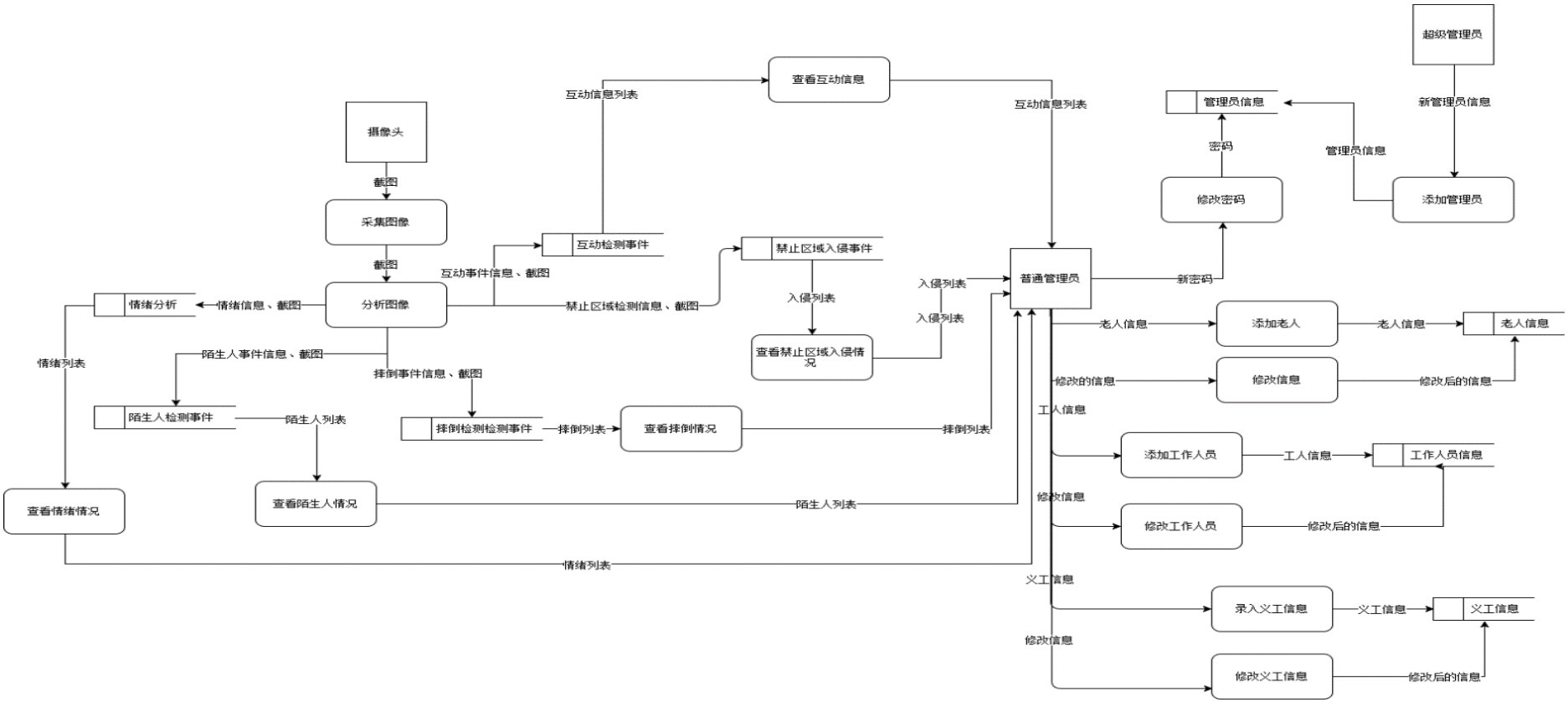
一旦检测到有人摔倒，将截图保存到硬盘，并实时插入一条数据到数据库。一旦数据库有了新数据，Web 端的实时图表会立即呈现。

2.4 条件与限制

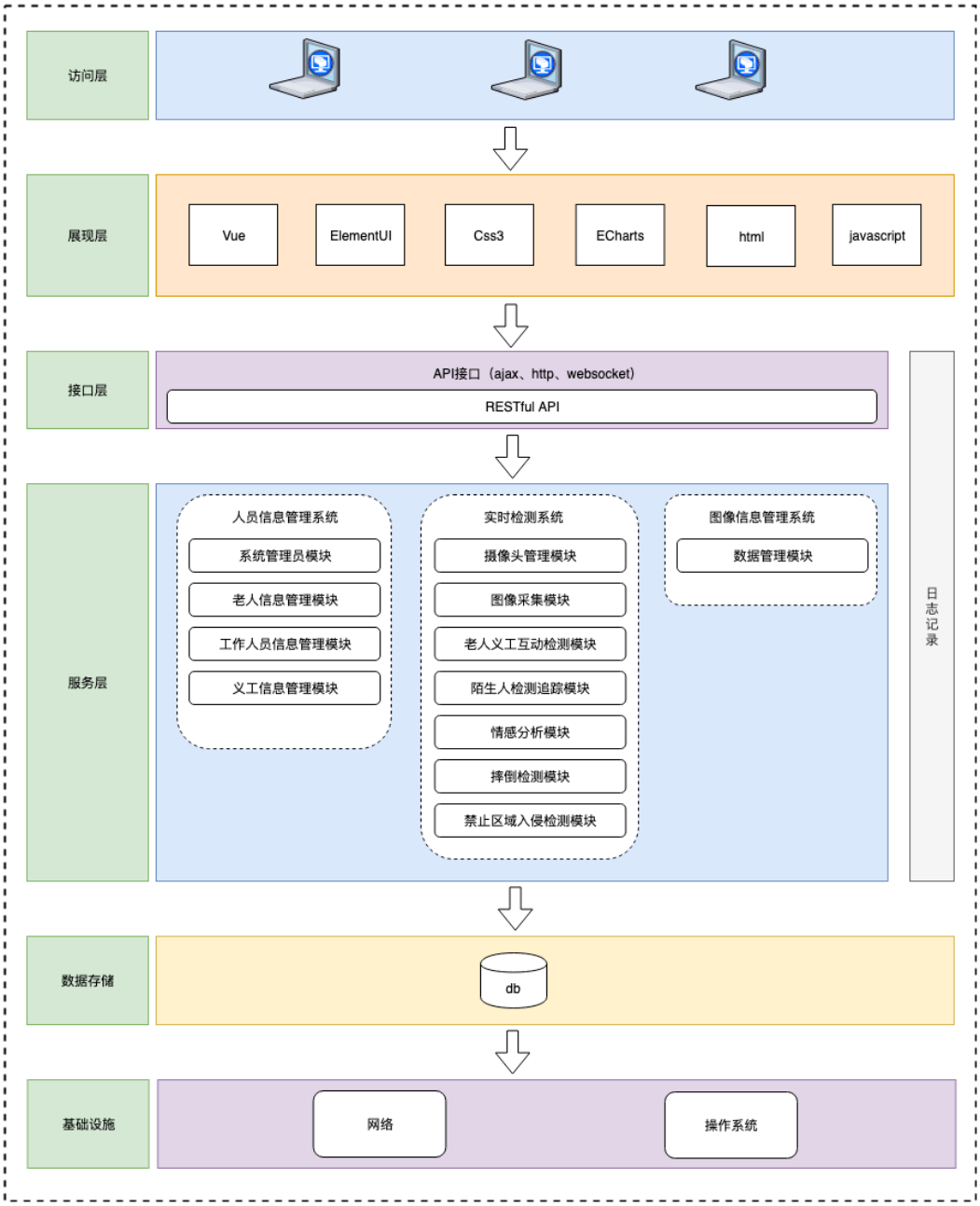
1. 开发周期过短，无法对所有的功能进行优化，无法进行第二次迭代。
2. 缺少摄像头设备，将 5 个摄像头的完成的功能集中到两个摄像头上，一个摄像头完成互动检测，陌生人检测，表情分析，摔倒检测功能，另一个摄像头完成禁止区域入侵检测。
3. 由于开发设备中缺少深度摄像头，无法判断距离，老人与义工的交互简单根据是否出现在同一个摄像头内即可。
4. 由于缺少专业景深摄像头，无法判断距离，只能通过画面截图进行分析判断，会有一些误判，因而会影响检测分析的准确率。
5. 因为受限于硬件的计算能力，只能每 5 秒截一张图进行分析，同时也影响了计算速度。
6. 因开发时间影响，在陌生人检测时，没有进行陌生人信息的记录，对相同的陌生人没有进行整合，多次显示相同陌生人的信息。

3. 总体设计

3.1 处理流程



3.2 总体结构和模块外部设计



将该系统一共分为 6 层，分别是访问层、展现层、接口层、服务层、数据存储层、基础设施层。

访问层为平台直接面向用户的界面，直接与用户交互。

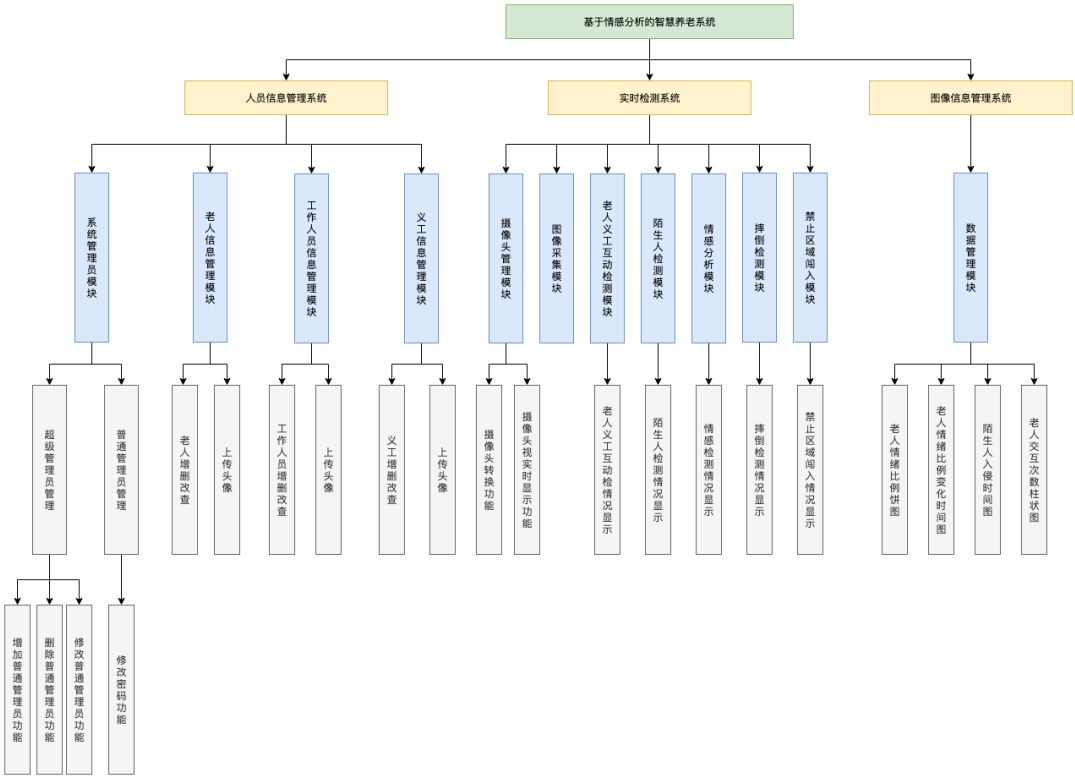
展现层为 web 界面展现。采用了 ElementUI、Vue、css3、ECharts、html、javascript 等技术来展示。主要功能是提供类型的用户操作界面和操作方案，捕捉和收集用高糊的输入信息。用户在该层进行所有功能模块的操作。

接口层为展现层和服务层之间的连接，用于数据传输。

服务层为功能实现层，用于 web 后端业务逻辑的实现并且与数据存储层进行交互。

数据存储层为数据库，用户将后端获得的数据进行存储。该数据库为关系型数据库。包括老人信息数据库、义工信息数据库、工作人员信息数据库、禁止区域入侵事件数据库、陌生人检测数据库、情感检测数据库、摔倒检测数据库、老人义工互动检测数据库。
基础设施层为基本网络和操作系统，是系统运行的基础。

3.3 功能结构



3.4 功能分配

	人员信息管理模块	实时监测模块	图像信息管理模块
老人信息管理	√		
义工信息管理	√		
工作人员信息管理	√		
管理员信息管理	√		
监控画面实时显示		√	
摔倒检测		√	
老人与义工互动检测		√	
陌生人入侵检测		√	
禁止区域闯入检测		√	

老人情绪检测		√	
陌生人入侵报警		√	
统计分析			√
摔倒检测情况显示			√
老人与义工互动检测情况显示			√
陌生人入侵检测情况显示			√
禁止区域闯入检测情况显示			√
老人情绪检测情况显示			√

4. 接口设计

4.1 内部接口

统一的输入输出参数

Response

参数名	类型	是否必须	描述
Success	Status	Y	Success：失败，fail：成功
Code	Int	Y	错误码
Content	Object	Y	消息

必须登录才能访问的接口

接口名称	描述
/account/login	管理员登陆
/elder/add	添加老人信息
/elder/remove	删除老人信息
/elder/update	更新老人信息
/elder/uploadPhoto	上传老人头像

/worker/add	添加工作人员信息
/worker/remove	删除工作人员信息
/worker/update	更新工作人员信息
/worker/uploadPhoto	上传工作人员头像
/volunteer/add	添加义工信息
/volunteer/remove	删除义工信息
/volunteer/update	更新义工信息
/volunteer/uploadPhoto	上传义工头像
/elder/init	初始化老人信息页面
/worker/init	初始化工作人员信息页面
/volunteer/init	初始化义工信息页面
/camera/emotioninit	初始化情感检测页面
/camera/fallinit	初始化摔倒检测页面
/camera/interactinit	初始化互动检测页面
/camera/invadeinit	初始化禁止区域闯入页面
/camera/strangerinit	初始化陌生人检测页面
/report/init	初始化主页面 echart 表格
/account/logout	管理员登出
/account/modifyPassword	管理员修改密码

用户登录（/account/signin）

接口	user/signin
描述	用户登录
验证	Ajax
方法	GET/POST

Request

参数名	类型	是否必须	描述
Email	String	y	用户名
password	String	y	密码

Response

参数名	类型	是否必须	描述
Data	Json	Y	用来判断是否登陆成功的信息

Example

Request

```
{
    Username：”1131898032@qq.com”,
    Password:  ”123456”
}
```

Response

```
1.成功情况
{
    "status": success,
    "code": 200,
}

2.失败情况
{
    "status": fail,
    "code": 400,
}
```

添加老人信息（/elder/add）

接口	/elder/add
描述	添加老人信息
验证	Ajax
方法	GET/POST

Request

参数名	类型	是否必须	描述
Name	String	y	老人姓名
Phone	String	N	老人电话
Birthday	String	N	老人生日
Gender	String	N	老人性别
Fam_name	String	N	老人家属
Fam_phone	String	N	老人家属电话
Id_card	String	N	老人身份证号

Response

参数名	类型	是否必须	描述
Data	Json	Y	用来判断是否成功的信息

Example

Request

```
{
    name：”peter”,
    phone:  ”123456”,
    Birthday:”1950-09-10”,
    Gender:”男”,
    Fam_name:”tony”
```

```
    Fam_phone:”12344441123”,
    Id_card:”12333939494”
}
```

Response

```
1.成功情况
{
    "status": success,
    "code": 200,
}

2.失败情况
{
    "status": fail,
    "code": 400,
}
```

删除老人信息（/elder/remove）

接口	/elder/remove
描述	删除老人信息
验证	Ajax
方法	GET/POST

Request

参数名	类型	是否必须	描述

Id	String	y	老人编号

Response

参数名	类型	是否必须	描述
Data	Json	Y	用来判断是否成功的信息

Example

Request

```
{
    Id：”3000”,
}
```

Response

```
1.成功情况
{
    "status": success,
    "code": 200,
}

2.失败情况
{
    "status": fail,
    "code": 400,
}
```

更改老人信息（/elder/update）

接口	/elder/update
描述	更改老人信息
验证	Ajax
方法	GET/POST

Request

参数名	类型	是否必须	描述
Name	String	y	老人姓名
Phone	String	N	老人电话
Birthday	String	N	老人生日
Gender	String	N	老人性别
Fam_name	String	N	老人家属
Fam_phone	String	N	老人家属电话
Id_card	String	N	老人身份证号

Response

参数名	类型	是否必须	描述
Data	Json	Y	用来判断是否成功的信息

Example

Request

```
{
    name：”peter”,
    phone:  ”123456”，
    Birthday:”1950-09-10”,
    Gender:”男”,
    Fam_name:”tony”
```

```
    Fam_phone:”12344441123”,
    Id_card:”12333939494”
}
```

Response

```
1.成功情况
{
    "status": success,
    "code": 200,
}

2.失败情况
{
    "status": fail,
    "code": 400,
}
```

上传老人头像（/elder/uploadPhoto）

接口	/elder/uploadPhoto
描述	上传老人头像
验证	Ajax
方法	GET/POST

Request

参数名	类型	是否必须	描述

Name	Formdata	y	图片信息

Response

参数名	类型	是否必须	描述
Data	Json	Y	用来判断是否成功的信息

Example

Request

```
{
      Photo:”elder/elder_photo/2.jpg”,
}
```

Response

```
1.成功情况
{
      "status": success,
      "code": 200,
}

2.失败情况
{
      "status": fail,
      "code": 400,
}
```

添加工作人员信息（/worker/add）

接口	/worker/add
描述	添加工作人员信息
验证	Ajax
方法	GET/POST

Request

参数名	类型	是否必须	描述
Name	String	y	工作人员姓名
Phone	String	N	工作人员电话
Birthday	String	N	工作人员生日
Gender	String	N	工作人员性别
Id_card	String	N	工作人员身份证号

Response

参数名	类型	是否必须	描述
Data	Json	Y	用来判断是否成功的信息

Example

Request

```
{
    name：”peter”,
    phone:  ”123456”，
    Birthday:”1950-09-10”,
    Gender:”男”,
    Id_card:”12333939494”
}
```

Response

```
1.成功情况
{
    "status": success,
    "code": 200,
}

2.失败情况
{
    "status": fail,
    "code": 400,
}
```

删除工作人员信息（/worker/remove）

接口	/worker/remove
描述	删除工作人员信息
验证	Ajax
方法	GET/POST

Request

参数名	类型	是否必须	描述
Id	String	y	工作人员编号

Response

参数名	类型	是否必须	描述

Data	Json	Y	用来判断是否成功的信息

Example

Request

```
{
    Id：”3000”,
}
```

Response

```
1.成功情况
{
    "status": success,
    "code": 200,
}

2.失败情况
{
    "status": fail,
    "code": 400,
}
```

更改工作人员信息（/worker/update）

接口	/worker/update
描述	更改工作人员信息

验证	Ajax
方法	GET/POST

Request

参数名	类型	是否必须	描述
Name	String	y	工作人员姓名
Phone	String	N	工作人员电话
Birthday	String	N	工作人员生日
Gender	String	N	工作人员性别
Id_card	String	N	工作人员身份证号

Response

参数名	类型	是否必须	描述
Data	Json	Y	用来判断是否成功的信息

Example

Request

```
{
    name：”peter”,
    phone:  ”123456”，
    Birthday:”1950-09-10”,
    Gender:”男”,
    Id_card:”12333939494”
}
```

Response

```
1.成功情况
{
```

```
    "status": success,
    "code": 200,
}

2.失败情况
{
    "status": fail,
    "code": 400,
}
```

上传工作人员头像（/worker/uploadPhoto）

接口	/worker/uploadPhoto
描述	上传工作人员头像
验证	Ajax
方法	GET/POST

Request

参数名	类型	是否必须	描述
Name	Formdata	y	图片信息

Response

参数名	类型	是否必须	描述
Data	Json	Y	用来判断是否成功的信息

Example

Request

```
{
```

```
    Photo:”worker/worker_photo/2.jpg”,
}
```

Response

```
1.成功情况
{
    "status": success,
    "code": 200,
}

2.失败情况
{
    "status": fail,
    "code": 400,
}
```

添加义工信息（/volunteer/add）

接口	/volunteer/add
描述	添加义工信息
验证	Ajax
方法	GET/POST

Request

参数名	类型	是否必须	描述
Name	String	y	义工姓名
Phone	String	N	义工电话
Birthday	String	N	义工生日
Gender	String	N	义工性别
Id_card	String	N	义工身份证号
Visitdate	String	N	义工来访日期

Response

参数名	类型	是否必须	描述
Data	Json	Y	用来判断是否成功的信息

Example

Request

```
{
    name：”peter”,
    phone:  ”123456”,
    Birthday:”1950-09-10”,
    Gender:”男”,
    Id_card:”12333939494”,
    Visitdate:”2019-07-09”
}
```

Response

```
1.成功情况
{
    "status": success,
    "code": 200,
```

```
}


2.失败情况
{
      "status": fail,
      "code": 400,
}
```

删除义工信息（/volunteer/remove）

接口	/volunteer/remove
描述	删除义工信息
验证	Ajax
方法	GET/POST

Request

参数名	类型	是否必须	描述
Id	String	y	义工编号

Response

参数名	类型	是否必须	描述
Data	Json	Y	用来判断是否成功的信息

Example

Request

```
{
     Id：”3000”,
```

```
}
```

Response

```
1.成功情况
{
    "status": success,
    "code": 200,
}

2.失败情况
{
    "status": fail,
    "code": 400,
}
```

更改义工信息（/volunteer/update）

接口	/volunteer/update
描述	更改义工信息
验证	Ajax
方法	GET/POST

Request

参数名	类型	是否必须	描述
Name	String	y	义工姓名

Phone	String	N	义工电话
Birthday	String	N	义工生日
Gender	String	N	义工性别
Id_card	String	N	义工身份证号
Visitdate	String	N	义工来访日期

Response

参数名	类型	是否必须	描述
Data	Json	Y	用来判断是否成功的信息

Example

Request

```
{
    name：”peter”,
    phone:  ”123456”,
    Birthday:”1950-09-10”,
    Gender:”男”,
    Id_card:”12333939494”,
    Visitdate:”2019-07-09”
}
```

Response

```
1.成功情况
{
    "status": success,
    "code": 200,
}
```

2.失败情况

```
{
    "status": fail,
    "code": 400,
}
```

上传义工头像（/volunteer/uploadPhoto）

接口	/volunteer/uploadPhoto
描述	上传义工头像
验证	Ajax
方法	GET/POST

Request

参数名	类型	是否必须	描述
Name	Formdata	y	图片信息

Response

参数名	类型	是否必须	描述
Data	Json	Y	用来判断是否成功的信息

Example

Request

```
{
    Photo:”volunteer/volunteer_photo/2.jpg”,
}
```

Response

```
1.成功情况
{
    "status": success,
    "code": 200,
}

2.失败情况
{
    "status": fail,
    "code": 400,
}
```

初始化老人信息（/elder/init）

接口	/elder/init
描述	初始化老人信息
验证	Ajax
方法	GET/POST

Request

参数名	类型	是否必须	描述
Get	Ajax	Y	Get 请求

Response

参数名	类型	是否必须	描述
Data	Json	Y	数据库内的老人信息

Example

Request

```
{
    “GET”
}
```

Response

```
1.成功情况
{
    "status": success,
    "code": 200,
    “content”:{
            0：{
                Num:”1”,
                Photo:”../media/elder_photo/2.jpg”,
                Name:”peter”,
                phone:  ”123456”，
                Birthday:”1950-09-10”,
                Gender:”男”,
                Fam_name:”tony”
                Fam_phone:”12344441123”,
                Id_card:”12333939494”
            }
    }
}
```

2.失败情况

```
{
    "status": fail,
    "code": 400,
}
```

初始化工作人员信息（/worker/init）

接口	/worker/init
描述	初始化工作人员信息
验证	Ajax
方法	GET/POST

Request

参数名	类型	是否必须	描述
Get	Ajax	Y	Get 请求

Response

参数名	类型	是否必须	描述
Data	Json	Y	数据库内的工作人员信息

Example

Request

```
{
    “GET”
```

```
}
```

Response

```
1.成功情况
{
    "status": success,
    "code": 200,
    “content”:{
            0：{
                Num:”1”,
                Photo:”../media/elder_photo/2.jpg”,
                Name:”peter”,
                phone:  ”123456”，
                Birthday:”1950-09-10”,
                Gender:”男”,
                Id_card:”12333939494”
              }
    }
}

2.失败情况
{
    "status": fail,
    "code": 400,
}
```

初始化义工信息（/volunteer/init）

接口	/volunteer/init
描述	初始化义工信息
验证	Ajax
方法	GET/POST

Request

参数名	类型	是否必须	描述
Get	Ajax	Y	Get 请求

Response

参数名	类型	是否必须	描述
Data	Json	Y	数据库内的义工信息

Example

Request

```
{
    “GET”
}
```

Response

```
1.成功情况
{
    "status": success,
    "code": 200,
    “content”:{
            0：{
```

```
                    Num:”1”,
                    Photo:”../media/elder_photo/2.jpg”,
                    Name:”peter”,
                    phone:  ”123456”，
                    Birthday:”1950-09-10”,
                    Gender:”男”,
                    Id_card:”12333939494”，
                    Visitdate:”2019-07-09”
                }
    }
}

2.失败情况
{
    "status": fail,
    "code": 400,
}
```

初始化互动检测信息（/camera/interactinit）

接口	/camera/interactinit
描述	初始化互动检测信息
验证	Ajax
方法	GET/POST

Request

参数名	类型	是否必须	描述
Get	Ajax	Y	Get 请求

Response

参数名	类型	是否必须	描述
Data	Json	Y	数据库内的义工信息

Example

Request

```
{
    “GET”
}
```

Response

```
1.成功情况
{
    "status": success,
    "code": 200,
    “content”:{
                Id:”1”,
                Time:”2019-07-09”,
                Photo:”../media/interact/1.jpg”,
                Elder:”peter”,
                Volunteer:”steve”
    }
}

2.失败情况
```

```
{
        "status": fail,
        "code": 400,
        “content”:{}

}
```

初始化陌生人检测信息（/camera/strangerinit）

接口	/camera/strangerinit
描述	初始化陌生人检测信息
验证	Ajax
方法	GET/POST

Request

参数名	类型	是否必须	描述
Get	Ajax	Y	Get 请求

Response

参数名	类型	是否必须	描述
Data	Json	Y	数据库内的义工信息

Example

Request

```
{
        “GET”
```

```
}
```

Response

```
1.成功情况
{
    "status": success,
    "code": 200,
    “content”:{
                Id:”1”,
                Gender:”男”,
                Upper_wear:”上衣”,
                Upper_wear_fg:”T 恤”,
                Cellphone:”有手机”
                Orientation:”侧面”,
                Headwear:”戴帽子”,
                Glasses:”戴眼镜”,
                Bag:”背包”,
                Upper_wear_texture:”有图案”,
                Upper_color:”红色”,
                Lower_wear:”不明”,
                Time:”2019-07-09”,
                Photo:”../media/stranger/1.jpg”,
    }
}

2.失败情况
{
    "status": fail,
```

```
    "code": 400,
    “content”:{}

}
```

初始化情感检测信息（/camera/emotioninit）

接口	/camera/emotioninit
描述	初始化情感检测信息
验证	Ajax
方法	GET/POST

Request

参数名	类型	是否必须	描述
Get	Ajax	Y	Get 请求

Response

参数名	类型	是否必须	描述
Data	Json	Y	数据库内的义工信息

Example

Request

```
{
    “GET”
}
```

Response

1.成功情况

```
{
    "status": success,
    "code": 200,
    “content”:{
                Id:”1”,
                Time:”2019-07-09”,
                Name:”peter”,
                Label:”happy”,
                Photo:”../media/stranger/1.jpg”,
    }
}
```

2.失败情况

```
{
    "status": fail,
    "code": 400,
    “content”:{}

}
```

初始化摔倒检测信息（/camera/fallinit）

接口	/camera/fallinit
描述	初始化摔倒检测信息
验证	Ajax
方法	GET/POST

Request

参数名	类型	是否必须	描述
Get	Ajax	Y	Get 请求

Response

参数名	类型	是否必须	描述
Data	Json	Y	数据库内的义工信息

Example

Request

```
{
    “GET”
}
```

Response

```
1.成功情况
{
    "status": success,
    "code": 200,
    “content”:{
```

```
                    Id:”1”,
                    Time:”2019-07-09”,
                    Photo:”../media/fall/1.jpg”,
    }
}

2.失败情况
{
    "status": fail,
    "code": 400,
    “content”:{}

}
```

初始化禁止区域闯入检测信息（/camera/invadeinit）

接口	/camera/invadeinit
描述	初始化禁止区域闯入检测信息
验证	Ajax
方法	GET/POST

Request

参数名	类型	是否必须	描述
Get	Ajax	Y	Get 请求

Response

参数名	类型	是否必须	描述
Data	Json	Y	数据库内的义工信息

Example

Request

```
{
    “GET”
}
```

Response

```
1.成功情况
{
    "status": success,
    "code": 200,
    “content”:{
                Id:”1”,
                Age:”30”,
                Gender:”男”,
                Upper_wear:”上衣”,
                Upper_wear_fg:”T 恤”,
                Cellphone:”有手机”
                Orientation:”侧面”,
                Headwear:”戴帽子”,
                Glasses:”戴眼镜”,
                Bag:”背包”,
                Upper_wear_texture:”有图案”,
                Upper_color:”红色”,
                Lower_wear:”不明”,
```

```
                    Time:”2019-07-09”,
                    Photo:”../media/invade/1.jpg”,
    }
}

2.失败情况
{
    "status": fail,
    "code": 400,
    “content”:{}

}
```

初始化 echart 表格信息（/report/init）

接口	/report/init
描述	初始化首页表格信息
验证	Ajax
方法	GET/POST

Request

参数名	类型	是否必须	描述
Get	Ajax	Y	Get 请求

Response

参数名	类型	是否必须	描述
Data	Json	Y	数据库内的义工信息

Example

Request

```
{
      “GET”
}
```

Response

```
1.成功情况
{
      "status": success,
      "code": 200,
      “content”:{
                  Interact:{
                       Name:”peter”,
                       Times:”8”
                   },
                  Emotion_pie:{
                       Angry:”4”,
                       Disgust:”2”,
                       Fear:”1”,
                       Happy:”5”,
                       Sad:”4”,
                       Surprise:”6”
                   },
                  Area:{
```

```
                    3
                }
        }
}

2.失败情况
{
        "status": fail,
        "code": 400,
        “content”:{}

}
```

管理员更改密码（/account/modifyPassword）

接口	/account/modifyPassword
描述	更改管理员密码
验证	Ajax
方法	GET/POST

Request

参数名	类型	是否必须	描述
Old_pass	String	y	旧密码
New_pass	String	Y	新密码

Response

参数名	类型	是否必须	描述
Data	Json	Y	用来判断是否更改成功的信息

Example

Request

```
{
    Old_pass:”111222”,
    New_pass:”2334444”
}
```

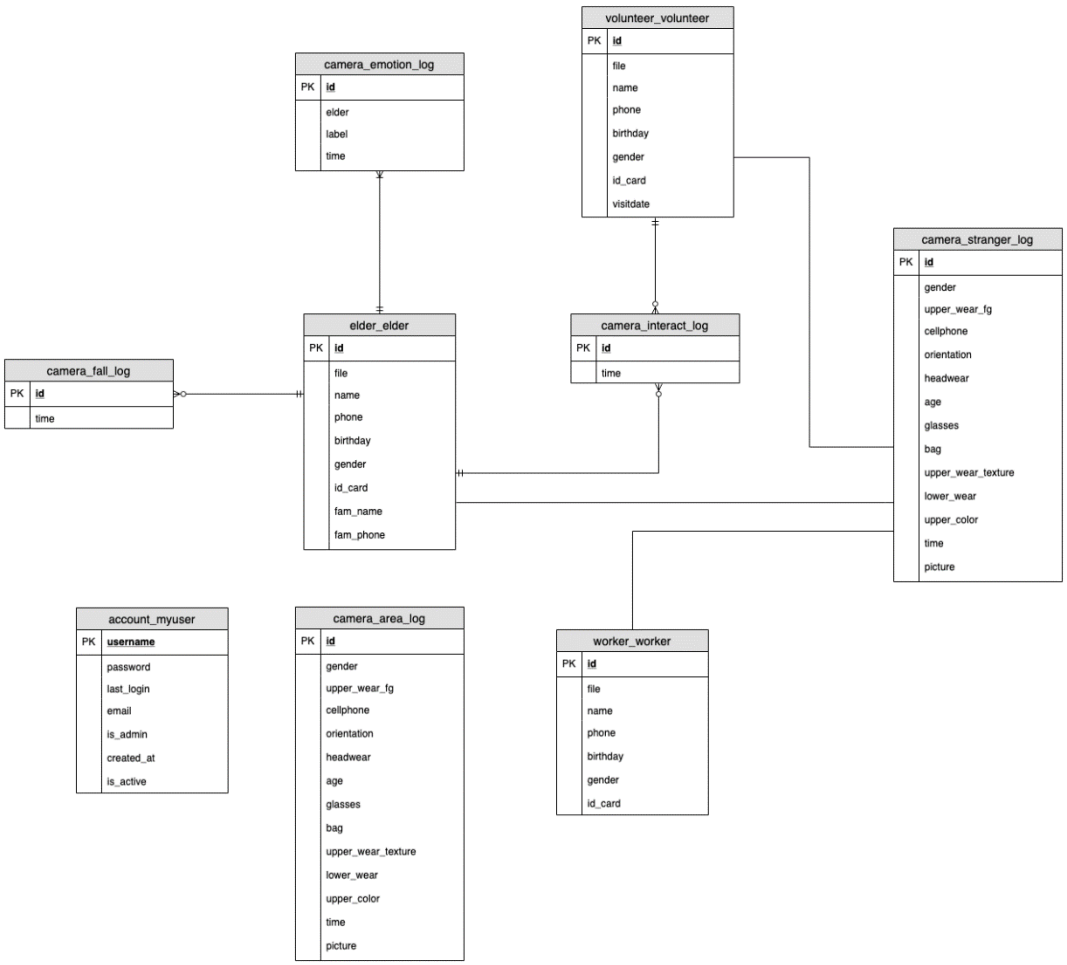
Response

```
1.成功情况
{
      "status": success,
      "code": 200,
}

2.失败情况
{
      "status": fail,
      "code": 400,
}
```

5. 数据结构设计

5.1 逻辑结构设计



5.2 物理结构设计

数据库名称	old_care	
表名	功能	备注
account_myuser	管理员信息表	存放管理员信息
elder_elder	老人信息表	存放老人信息
volunteer_volunteer	义工信息表	存放义工信息
worker_worker	工作人员信息表	存放工作人员信息
camera_area_log	禁止区域闯入事件信息表	存放禁止区域闯入事件信息
camera_emotion_log	老人情绪检测事件信息表	存放老人情绪检测事件信息
camera_interaction_log	互动检测事件信息表	存放互动检测事件信息

camera_fall_log	摔倒检测事件信息表	存放摔倒检测事件信息
camera_stranger_log	陌生人检测事件信息表	存放陌生人检测事件信息

管理员信息表

列名	类型	大小	约束	说明
password	varchar	128	非空	管理员密码
last_login	datetime			上一次登录的时间
email	varchar	40	非空；唯一	邮箱
username	varchar	40	非空；主键	姓名
is_admin	bool		非空	是否是超级管理员
created_at	date		非空	创建时间
is_active	bool		非空	是否在线

老人信息表

列名	类型	大小	约束	说明
id	number	11	非空；主键；自增	老人 id
file	varchar	200	无	头像文件的路径
name	varchar	50	无	姓名
phone	number	50	无	联系电话
birthday	datatime		无	生日
gender	varchar	5	无	性别
id_card	number	50	无	身份证
fam_name	varchar	50	无	亲属姓名
fam_phone	number	50	无	亲属电话

义工信息表

列名	类型	大小	约束	说明
id	number	11	非空；主键；自增	义工 id
file	varchar	200	无	头像文件的路径
name	varchar	50	无	姓名
phone	number	50	无	联系电话
birthday	datatime		无	生日
gender	varchar	5	无	性别
id_card	number	50	无	身份证
visitdate	datatime		无	登记时间

工作人员信息表

列名	类型	大小	约束	说明
id	number	11	非空；主键；自增	工作人员 id
file	varchar	200	无	头像文件的路径
name	varchar	50	无	姓名
phone	number	50	无	联系电话
birthday	datatime		无	生日

gender	varchar	5	无	性别
id_card	number	50	无	身份证

陌生人检测事件信息表

列名	类型	大小	约束	说明
id	number		非空；主键；自增	
gender	varchar	100	非空	性别
upper_wear_fg	varchar	100	非空	是否戴帽
cellphone	varchar	100	非空	是否手持电话
orientation	varchar	100	非空	正面 or 侧面
headwear	varchar	100	非空	上衣类型
age	varchar	100	非空	年龄
glasses	varchar	100	非空	是否戴眼镜
bag	varchar	100	非空	是否背包
upper_wear_texture	varchar	100	非空	上衣花纹
lower_wear	varchar	100	非空	下衣
upper_color	varchar	100	非空	上衣颜色
time	datetime		非空	检测时间
picture	varchar	100		截图

禁止区域入侵检测事件信息表

列名	类型	大小	约束	说明
id	integer		非空；主键；自增	
gender	varchar	100	非空	性别
upper_wear_fg	varchar	100	非空	是否戴帽
cellphone	varchar	100	非空	是否手持电话
orientation	varchar	100	非空	正面 or 侧面
headwear	varchar	100	非空	上衣类型
age	varchar	100	非空	年龄
glasses	varchar	100	非空	是否戴眼镜
bag	varchar	100	非空	是否背包
upper_wear_texture	varchar	100	非空	上衣花纹
lower_wear	varchar	100	非空	下衣
upper_color	varchar	100	非空	上衣颜色
time	datetime		非空	检测时间
picture	varchar	100		截图

互动检测事件信息表

列名	类型	大小	约束	说明
id	integer		非空；主键；自增	
time	datetime		非空	互动时间

情绪变化检测信息表

列名	类型	大小	约束	说明
id	integer		非空；主键；自增	
elder	varchar	100	非空	老人姓名
label	varchar	100	非空	情绪类别
time	datetime		非空	检测时间

摔倒检测事件信息表

列名	类型	大小	约束	说明
id	integer		非空；主键；自增	
time	datetime		非空	摔倒时间

5.3 数据结构与程序的关系

6. 运行设计

6.1 运行模块的组合

客户机程序在有输入时启动接收数据模块，通过各模块之间的调用，读入并对输入进行格式化。在接收数据模块得到充分的数据时，将调用网络传输模块，将数据通过网络送到服务器，并等待接收服务器返回的信息。接收到返回信息后随即调用数据输出模块，对信息进行处理，产生相应的输出。

服务器程序的接收网络数据模块必须始终处于活动状态。接收到数据后，调用数据处理查询模块对数据库进行访问，完成后调用网络发送模块，将信息返回客户机。

6.2 运行控制

运行控制将严格按照各模块间函数调用关系来实现。在各事务中心模块中，需对运行控制进行正确的判断，选择正确的运行控制路径。

在网络传方面，客户机在发送数据后，将等待服务器的确认收到信号，收到后，再次等待服务器发送回答数据，然后对数据进行确认。服务器在接到数据后发送确认信号，在对数据处理、访问数据库后，将返回信息送回客户机，并等待确认。

6.3 运行时间

在软体的需求分析中，对运行时间的要求为必须对做出的操作有较快的反应。网络硬件对运行时间有最大的影响，当网络负载量大时，对操作反应将受到很大的影响。所以将采用高速 ATM 网络， 实现客户机与服务器之间的连接，以减少网络传输上的开销。其次是服务

器的性能，这将影响对数据库访问时间即操作时间的长短，影响加大客户机操作的等待时间，所以必须使用高性能的服务器。硬件对本系统的速度影响将会大于软件的影响。

7. 出错处理设计

7.1 出错输出信息

出错名称	系统输出信息	处理方法
用户名输入错误	“用户名错误，请重新输入。”	进入登录页面
密码输入错误	“密码错误，请重新输入。”	进入登录页面
输入数据格式错误	“输入格式不正确，请检查后再输入。”	返回原输入界面
不允许为空的输入框输入空字符	“请输入。”	弹回原输入页面的输入处
摄像头未开启	“未检测到可用摄像头，无法进行监控”	停止监控
摄像头故障	“摄像头出现故障，无法进行监控”	停止监控
系统故障	“服务器维护中，暂停服务。”	立即启用备用机，恢复故障

7.2 出错处理对策

由于数据在数据库中已经有备份，故在系统出错后可以依靠数据库的恢复功能，并且依靠日志文件使系统再启动，就算系统崩溃用户数据也不会丢失或遭到破坏。但有可能占用更多的数据存储空间，权衡措施由用户来决定。

基于情感分析的智慧养老系统

详细设计

指导老师：张健

项目成员：段智超、储泽栋、毕宇航、路杨、于佳莉

时间：2019.7.9

目录

1.1 登陆功能

1.1.1 功能描述：

管理员登录时输入自己的账户信息进入到主页面

1.1.2 性能描述：

准确的将输入的信息传达到后端并与数据库内的信息进行匹配，及时的将正确的匹配结果返回致前端。

1.1.3 输入：

用户名，密码。

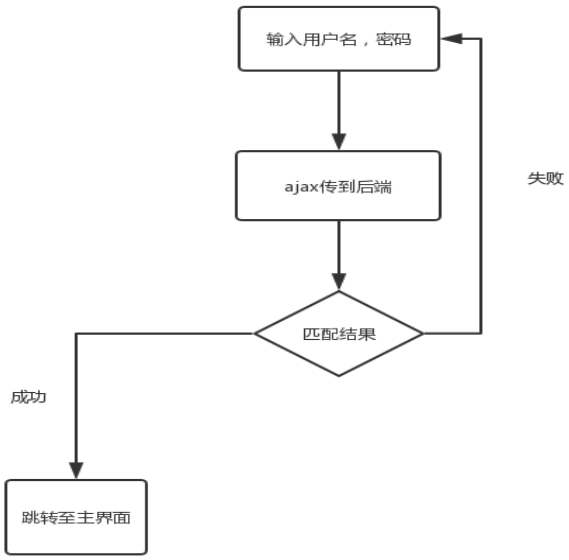
1.1.4 输出：

匹配结果。若成功，跳转至主页面；若失败，提示用户名（密码）不正确，返回登陆界面。

1.1.5 算法：

通过 js 捕获到用户输入信息，以 ajax 的形式传值到后端，获取匹配结果返回到前端。

1.1.6 程序逻辑：



1.1.7 接口：

/account/login

1.1.8 存储分配：

极少

1.1.9 限制条件：

输入用户名必须是邮箱格式

1.2 主界面初始化功能

1.2.1 功能描述：

管理员登录致主界面后，主界面将数据库内现有数据初始化在表格中。

1.2.2 性能描述：

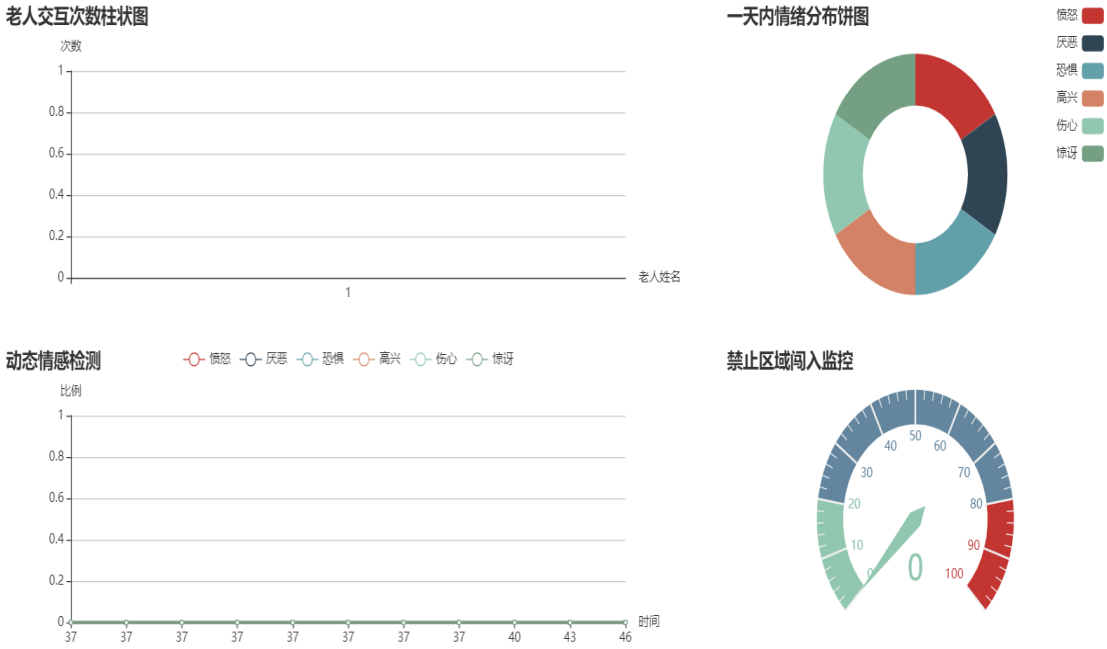
实时的将数据库内信息以表格形式动态加载在界面上。

1.2.3 输入：

无

1.2.4 输出：

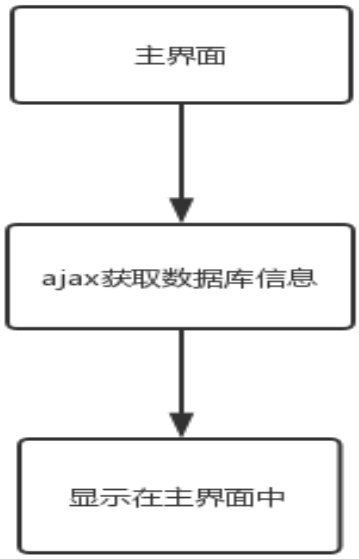
echart 图表



1.2.5 算法：

通过 ajax 调用后台数据库内的数据，并赋值到 vue 内 echart 的 data 部分内变量上。

1.2.6 程序逻辑：



1.2.7 接口：

/report/init

1.2.8 存储分配：

极少

1.2.9 限制条件：

1.3 界面切换功能

1.3.1 功能描述：

管理员登录致主界面后，可以根据自己的需要切换到其他页面。

1.3.2 性能描述：

跳转请求发送后，应该瞬间切换到目标界面，延迟不超过 1s

1.3.3 输入：

点击事件

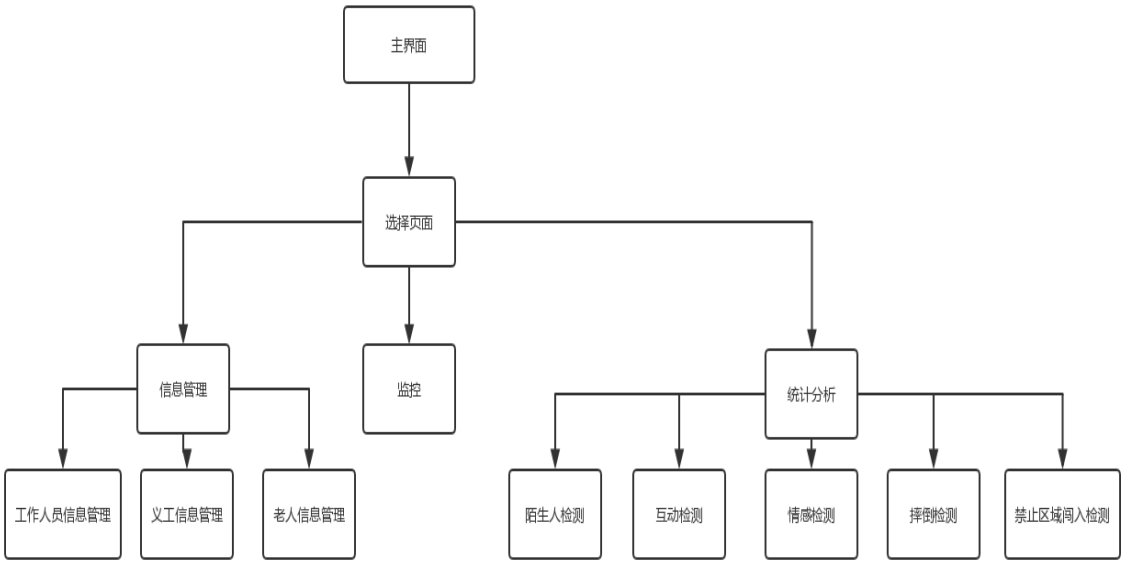
1.3.4 输出：

目标界面

1.3.5 算法：

js@onclick=open（）；函数

1.3.6 程序逻辑：



1.3.7 接口：

无

1.3.8 存储分配：

极少

1.3.9 限制条件：

无

1.4 老人信息管理功能

1.4.1 功能描述：

管理员根据实际情况对老人信息进行增删改查操作。

1.4.2 性能描述：

能够正确地将老人信息传达到后端并保存到数据库中，允许修改头像。

1.4.3 输入：

姓名，手机号，生日，性别，身份证号，亲属姓名，亲属电话号，头像；删除按钮点击；查询条件。

1.4.4 输出：

添加成功/失败；修改成功/失败；删除成功/失败；查询结果。

1.4.5 算法：

ajax 与 vue.js 结合传值到后端，并获取结果，将反馈的信息展示到前端上。

添加：Vue 获取管理员的输入项，以 ajax 的形式 post 到后端，得到一个 json 类型的 data 数据，如果 data 中反馈的是 success 信息，则将输入项信息 push 到框架中 data 数组中，并在页面上新增一行添加的数据。

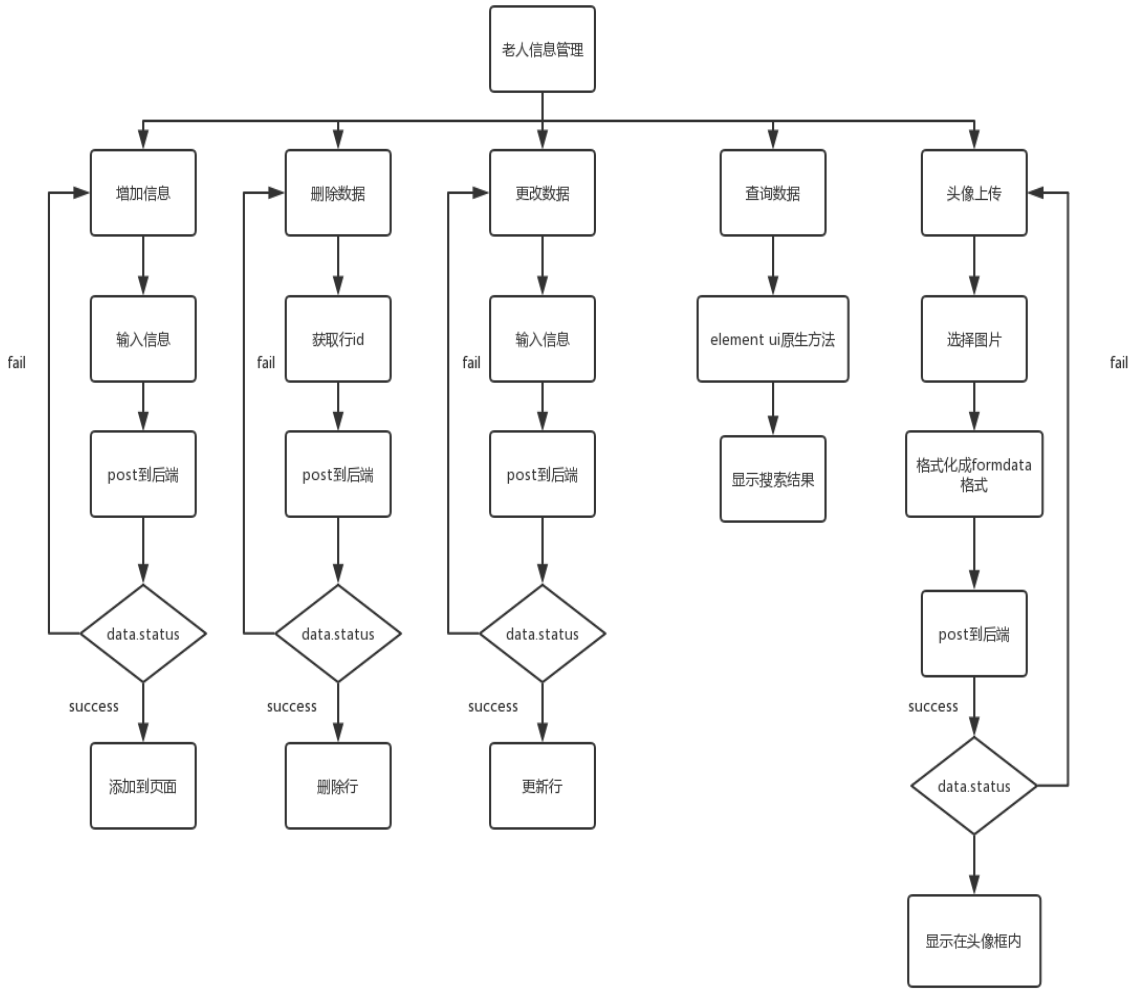
删除：Vue 获取删除行的 id，以 ajax 的形式 post 到后端，得到一个 json 类型的 data 数据，如果 data 中反馈的是 success 信息，则在页面上删除掉该行数据。

修改：Vue 获取管理员的输入项，以 ajax 的形式 post 到后端，得到一个 json 类型的 data 数据，如果 data 中反馈的是 success 信息，则将输入项信息覆盖原数组的数据上，并在页面上更新添加的数据。

查询：elementUI 框架原生的查询框，自动实现根据关键字搜索功能

头像上传：获取当前行的 id，图片信息，以 formdata 的格式命名传递到后端，得到一个 json 类型的 data 数据，如果 data 中反馈的是 success 信息，则将图片显示在页面头像框内。

1.4.6 程序逻辑：



1.4.7 接口：

/elder/add，/elder/remove，/elder/update，/elder/uploadPhoto

1.4.8 存储分配：

极少

1.4.9 限制条件：

图片上传必须是 jpeg 格式

1.5 义工信息管理功能

1.5.1 功能描述：

管理员根据实际情况对义工信息进行增删改查操作。

1.5.2 性能描述：

能够正确地将义工信息传达到后端并保存到数据库中，允许修改头像。

1.5.3 输入：

姓名，手机号，生日，性别，身份证号，头像；删除按钮点击；查询条件。

1.5.4 输出：

添加成功/失败；修改成功/失败；删除成功/失败；查询结果。

1.5.5 算法：

ajax 与 vue.js 结合传值到后端，并获取结果，将反馈的信息展示到前端上。

添加：Vue 获取管理员的输入项，以 ajax 的形式 post 到后端，得到一个 json 类型的 data 数据，如果 data 中反馈的是 success 信息，则将输入项信息 push 到框架中 data 数组中，并在页面上新增一行添加的数据。

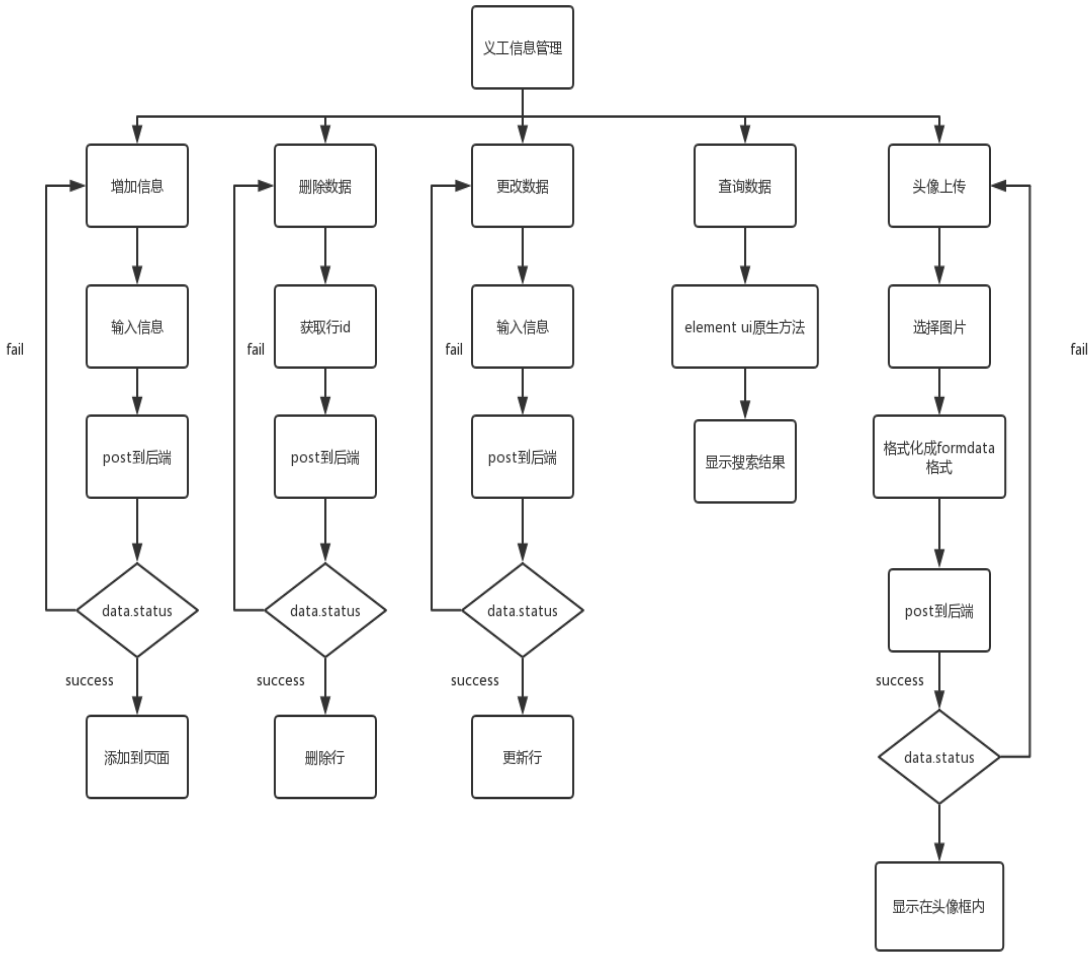
删除：Vue 获取删除行的 id，以 ajax 的形式 post 到后端，得到一个 json 类型的 data 数据，如果 data 中反馈的是 success 信息，则在页面上删除掉该行数据。

修改：Vue 获取管理员的输入项，以 ajax 的形式 post 到后端，得到一个 json 类型的 data 数据，如果 data 中反馈的是 success 信息，则将输入项信息覆盖原数组的数据上，并在页面上更新添加的数据。

查询：element UI 框架原生的查询框，自动实现根据关键字搜索功能

头像上传：获取当前行的 id，图片信息，以 formdata 的格式命名传递到后端，得到一个 json 类型的 data 数据，如果 data 中反馈的是 success 信息，则将图片显示在页面头像框内。

1.5.6 程序逻辑：



1.5.7 接口：

/volunteer/add，/volunteer/remove，/volunteer/update，/volunteer/uploadPhoto

1.5.8 存储分配：

极少

1.5.9 限制条件：

图片上传必须是 jpeg 格式

1.6 工作人员信息管理功能

1.6.1 功能描述：

管理员根据实际情况对工作人员信息进行增删改查操作。

1.6.2 性能描述：

能够正确地将工作人员信息传达到后端并保存到数据库中，允许修改头像。

1.6.3 输入：

姓名，手机号，生日，性别，身份证号，头像；删除按钮点击；查询条件。

1.6.4 输出：

添加成功/失败；修改成功/失败；删除成功/失败；查询结果。

1.6.5 算法：

ajax 与 vue.js 结合传值到后端，并获取结果，将反馈的信息展示到前端上。

添加：vue 获取管理员的输入项，以 ajax 的形式 post 到后端，得到一个 json 类型的 data 数据，如果 data 中反馈的是 success 信息，则将输入项信息 push 到框架中 data 数组中，并在页面上新增一行添加的数据。

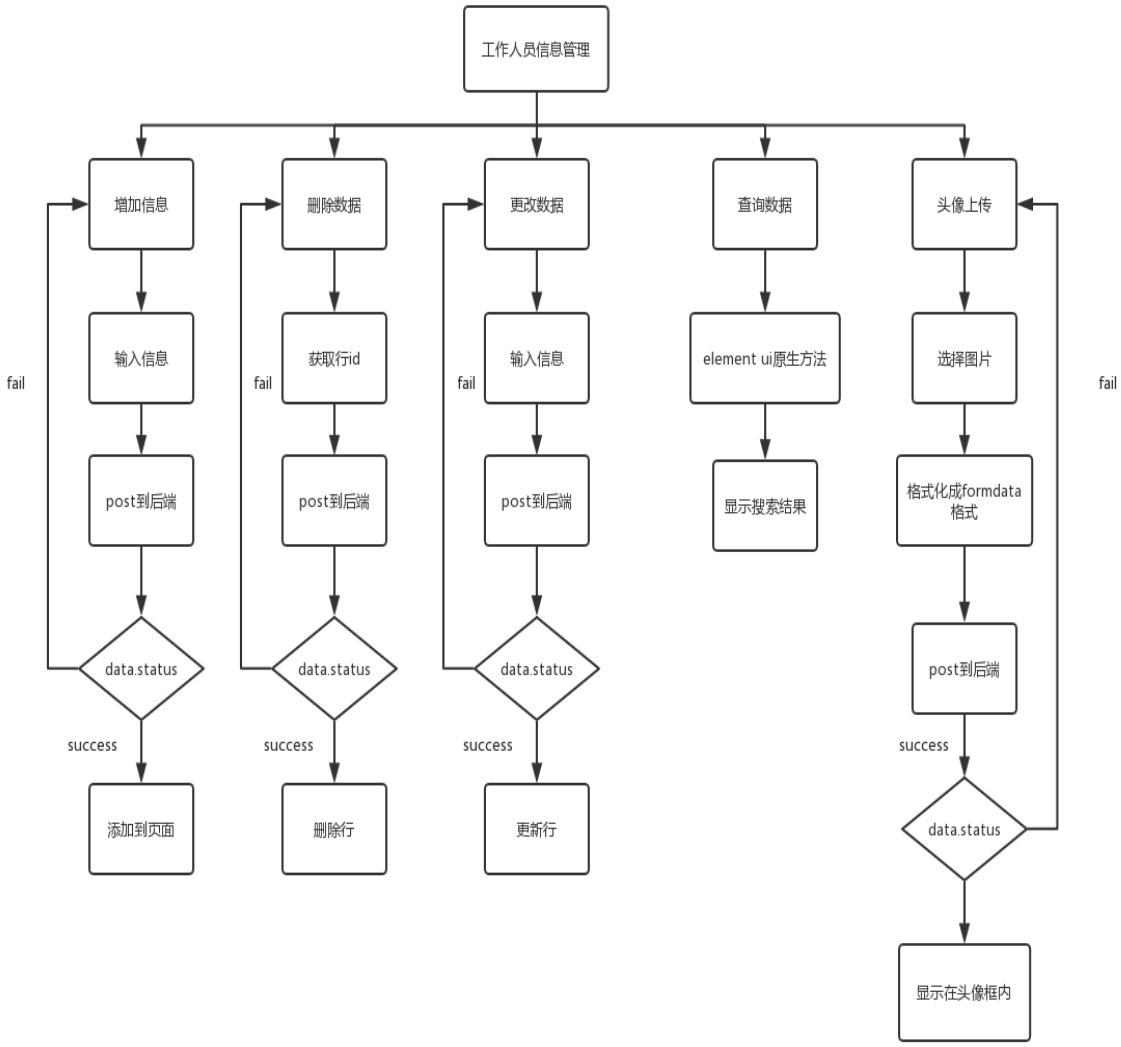
删除：vue 获取删除行的 id，以 ajax 的形式 post 到后端，得到一个 json 类型的 data 数据，如果 data 中反馈的是 success 信息，则在页面上删除掉该行数据。

修改：vue 获取管理员的输入项，以 ajax 的形式 post 到后端，得到一个 json 类型的 data 数据，如果 data 中反馈的是 success 信息，则将输入项信息覆盖原数组的数据上，并在页面上更新添加的数据。

查询：element ui 框架原生的查询框，自动实现根据关键字搜索功能

头像上传：获取当前行的 id，图片信息，以 formdata 的格式命名传递到后端，得到一个 json 类型的 data 数据，如果 data 中反馈的是 success 信息，则将图片显示在页面头像框内。

1.6.6 程序逻辑：



1.6.7 接口：

/worker/add，/worker/remove，/worker/update，/worker/uploadPhoto

1.6.8 存储分配：

极少

1.6.9 限制条件：

图片上传必须是 jpeg 格式

1.7 陌生人检测功能

1.7.1 功能描述：

管理员查看陌生人出现情况。

1.7.2 性能描述：

能够及时的将最新的陌生人信息显示在页面上。

1.7.3 输入：

点击查看按钮。

1.7.4 输出：

陌生人信息。

1.7.5 算法：

ajax 与 vue.js 结合传值到后端，并获取结果，将反馈的信息展示到前端上。

1.7.6 接口：

/camera/starangerinit

1.7.7 存储分配：

极少

1.7.8 限制条件：

有陌生人出现

1.8 情感检测功能

1.8.1 功能描述：

管理员查看老人情感情况。

1.8.2 性能描述：

能够及时的将最新的情感状态信息显示在页面上。

1.8.3 输入：

点击查看按钮。

1.8.4 输出：

情感状态。

1.8.5 算法：

ajax 与 vue.js 结合传值到后端，并获取结果，将反馈的信息展示到前端上。

1.8.6 接口：

/camera/emotioninit

1.8.7 存储分配：

极少

1.8.8 限制条件：

无

1.9 互动检测功能

1.9.1 功能描述：

管理员查看老人与义工情况。

1.9.2 性能描述：

能够及时的将最新的互动情况信息显示在页面上。

1.9.3 输入：

点击查看按钮。

1.9.4 输出：

互动信息。

1.9.5 算法：

ajax 与 vue.js 结合传值到后端，并获取结果，将反馈的信息展示到前端上。

1.9.6 接口：

/camera/interactinit

1.9.7 存储分配：

极少

1.9.8 限制条件：

有交互情况出现

1.10 摔倒检测功能

1.10.1 功能描述：

管理员查看老人摔倒情况。

1.10.2 性能描述：

能够及时的将最新的摔倒信息显示在页面上。

1.10.3 输入：

点击查看按钮。

1.10.4 输出：

摔倒信息。

1.10.5 算法：

ajax 与 vue.js 结合传值到后端，并获取结果，将反馈的信息展示到前端上。

1.10.6 接口：

/camera/fallinit

1.10.7 存储分配：

极少

1.10.8 限制条件：

有摔倒情况出现

1.11 禁止区域闯入检测功能

1.11.1 功能描述：

管理员查看禁止区域闯入情况。

1.11.2 性能描述：

能够及时的将最新的禁止区域闯入信息显示在页面上。

1.11.3 输入：

点击查看按钮。

1.11.4 输出：

禁止区域闯入信息。

1.11.5 算法：

ajax 与 vue.js 结合传值到后端，并获取结果，将反馈的信息展示到前端上。

1.11.6 接口：

/camera/areainit

1.11.7 存储分配：

极少

1.11.8 限制条件：

禁止区域有人入内

1.12 监控功能

1.12.1 功能描述：

管理员实时查看摄像头监视区域内的监控，并能切换摄像头。

1.12.2 性能描述：

能够实时的将监控画面显示在页面上。

1.12.3 输入：

切换摄像头。

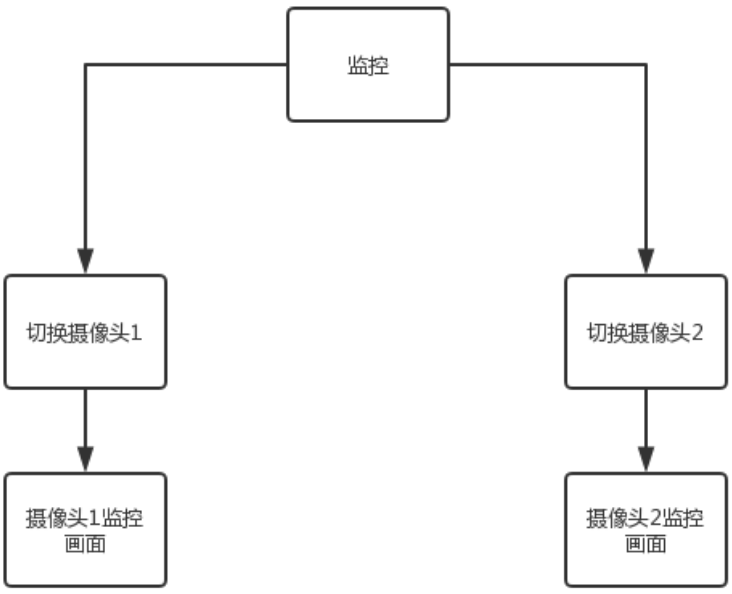
1.12.4 输出：

监控画面。

1.12.5 算法：

ajax 与 vue.js 结合传值到后端，并获取结果，将反馈的信息展示到前端上。Js 函数改变 img 标签内容。

1.12.6 程序逻辑：



1.12.7 接口：

无

1.12.8 存储分配：

极少

1.12.9 限制条件：

无

1.13 Echart 图表展示功能

1.13.1 功能描述：

将各种检测信息以图表的形式动态的加载在页面上。

1.13.2 性能描述：

实时展示数据变化情况。

1.13.3 输入：

无。

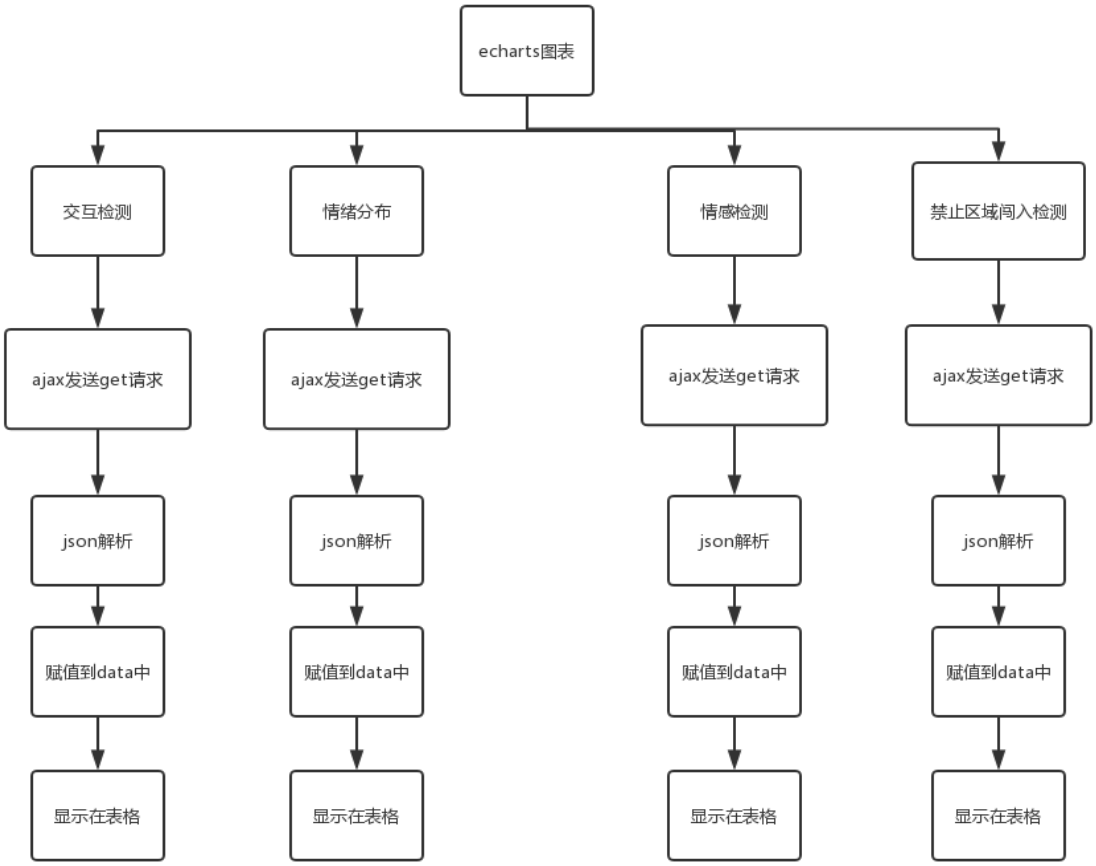
1.13.4 输出：

echart 图表。

1.13.5 算法：

ajax 与 vue.js 结合传值到后端，并获取结果，将反馈的 json 进行解析，通过遍历的方式将其中的信息提取出来并对应的添加到 echart 数据结构内。

1.13.6 程序逻辑：



1.13.7 接口：

/report/init

1.13.8 存储分配：

极少

1.13.9 限制条件：

无

1.14 管理员修改密码功能

1.14.1 功能描述：

管理员修改自己的密码。

1.14.2 性能描述：

准确的将各项信息存到数据库。

1.14.3 输入：

旧密码，新密码，确认新密码。

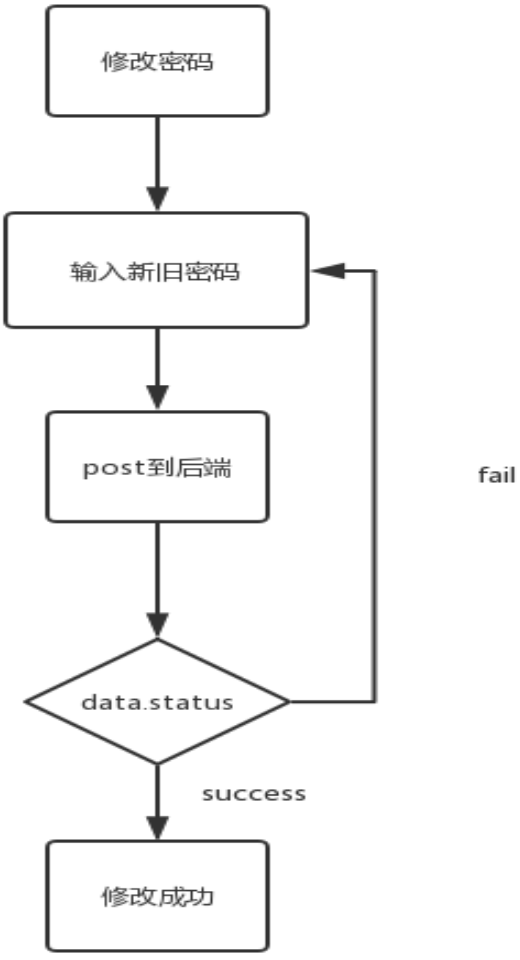
1.14.4 输出：

修改是否成功信息。

1.14.5 算法：

ajax 与 vue.js 结合传值到后端，并获取结果。

1.14.6 程序逻辑：



1.14.7 接口：

/account/modifyPassword

1.14.8 存储分配：

极少

1.14.9 限制条件：

新旧密码，确认密码全部输入，切新密码与确认密码一致

1.15 人脸检测功能（本地）

1.15.1 功能描述

通过给定的图像，能够找出所有的人脸并且记录其位置

1.15.2 性能描述

经测试，在 CPU i7 7700-HQ （2.80GHz），GTX 1060 6G 环境下，平均 648ms 完成一次本地模型计算并返回结果。

1.15.3 输入

要求输入图片的文件名，且文件必须没有损坏。

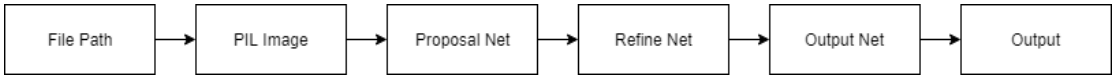
1.15.4 输出

输出 Bounding Box.

1.15.5 算法

MTCNN
通过三层级联网络结构逐层增加识别精度与目标数目最终通过 Output Net 完成最终预测。

1.15.6 程序逻辑



1.15.7 接口

函数名称	参数类型	返回值类型	备注
get_faces	image url of file:	list, list	
nms	list	list	执行 nms 操作需要确保按照目标分类进行，比如人体与狗不应该一起放入 nms

1.15.8 存储分配

2.85M 显存

1.15.9 限制条件

本地模型运行需要 pytorch>=0.4.0, numpy 版本必须为 1.15.4。
图片不能为灰度图。

1.16 目标检测（本地）

1.16.1 功能描述

通过给定的图像，能够找出所有的检测目标，并返回 Bounding Box.

1.16.2 性能描述

经测试，在 CPU i7 7700-HQ（2.85GHz），GTX 1060 6G 环境下，平均 300ms 完成一次本地模型计算并返回结果。

1.16.3 输入

要求输入图片的文件名，且文件必须没有损坏。

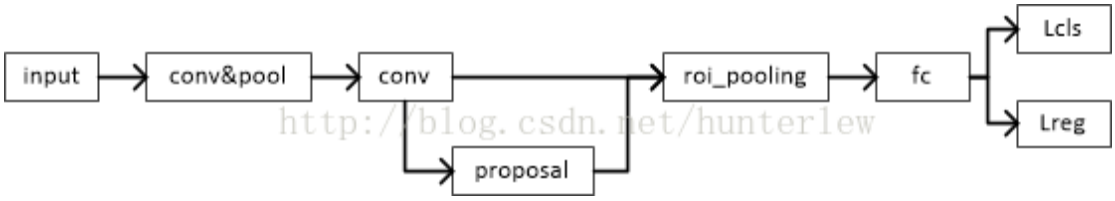
1.16.4 输出

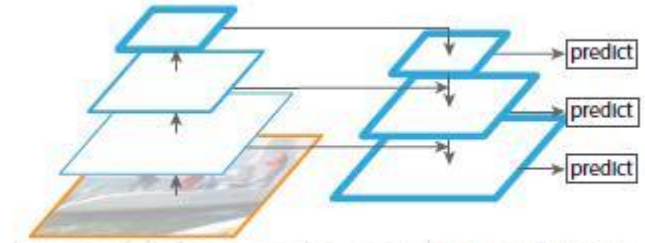
输出若干类别物体的 Bounding Box。

1.16.5 算法

使用 fasterrcnn_resnet50_fpn 作为网络结构，进行目标检测。

1.16.6 程序逻辑





(d) Feature Pyramid Network

1.16.7 接口

函数名称	参数类型	返回值类型	备注
Detect_object.process	img_path of str	list, list	返回值包括裁剪出的图像的文件名以及解析结果

1.17 表情识别（本地）

1.17.1 功能描述

通过脸部区域图像，识别出该张人脸的表情

1.17.2 性能描述

经测试，在 CPU i7 7700-HQ （2.80GHz），GTX 1060 6G 环境下，平均 200ms 完成一次本地模型计算并返回结果。

1.17.3 输入

要求输入图片的文件名，且文件没有损坏。

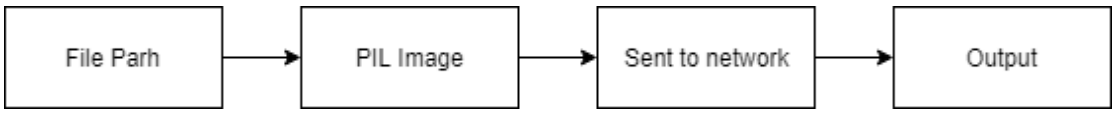
1.17.4 输出

输出表情的类别。

1.17.5 算法

使用 Resnet-18 完成表情分类。

1.17.6 程序逻辑



1.17.7 接口

函数名称	参数类型	返回值类型	备注
Emotion_classification.inference	img_path of str	str	返回值直接返回对应表情类别的字符串

1.18 摔倒检测（本地）

1.18.1 功能

通过人体区域图像，识别出属于该人体的若干关键点，并且根据关键点相对位置，完成摔倒检测。

1.18.2 性能描述

经测试，在 CPU i7 7700-HQ （2.80GHz），GTX 1060 6G 环境下，平均 500ms 完成一次本地模型计算并返回结果。

1.18.3 输入

仅包含单个人体的图像文件名，且图像没有损坏。

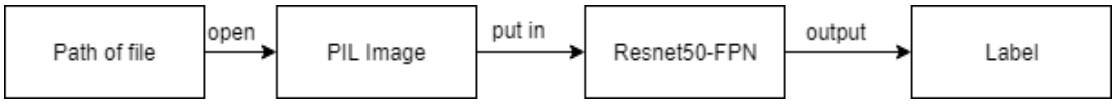
1.18.4 输出

输出是否检测到摔倒，True 为摔倒，False 为没有摔倒。

1.18.5 算法

使用 resnet50-fpn 进行关键点检测。

1.18.6 程序逻辑



1.18.7 接口

函数名称	参数类型	返回值类型	备注
Pose_estimation.process	img_path of str	Boolean	返回值为是否摔倒

1.19 人脸检测功能（网络）

1.19.1 功能描述

通过给定的图像，能够调用百度 AI 开发者平台提供的接口，找出所有的人脸并且记录其位置

1.19.2 性能描述

百度 AI 开发者平台对个人开发者开放 QPS 为 2，经测试，平均 500ms 完成一次网络模型计算并返回结果。

1.19.3 输入

要求输入图片的路径名，且文件必须没有损坏。

1.19.4 输出

字段	必选	类型	说明
face_num	是	int	检测到的图片中的人脸数量
face_list	是	array	人脸信息列表，具体包含的参数参考下面的列表。
+face_token	是	string	人脸图片的唯一标识
+location	是	array	人脸在图片中的位置
++left	是	double	人脸区域离左边界的距离
++top	是	double	人脸区域离上边界的距离
++width	是	double	人脸区域的宽度
++height	是	double	人脸区域的高度

字段	必选	类型	说明
++rotation	是	int64	人脸框相对于竖直方向的顺时针旋转角，[-180,180]
+face_probability	是	double	人脸置信度，范围【0~1】，代表这是一张人脸的概率，0 最小、1 最大。
+angel	是	array	人脸旋转角度参数
++yaw	是	double	三维旋转之左右旋转角[-90(左)，90(右)]
++pitch	是	double	三维旋转之俯仰角度[-90(上)，90(下)]
++roll	是	double	平面内旋转角[-180(逆时针)，180(顺时针)]
+age	否	double	年龄 ，当 face_field 包含 age 时返回
+beauty	否	int64	美丑打分，范围 0-100，越大表示越美。当 face_fields 包含 beauty 时返回
+expression	否	array	表情，当 face_field 包含 expression 时返回

字段	必选	类型	说明
++type	否	string	none:不笑；smile:微笑；laugh:大笑
++probability	否	double	表情置信度，范围【0~1】，0 最小、1 最大。
+face_shape	否	array	脸型，当 face_field 包含 face_shape 时返回
++type	否	double	square: 正方形 triangle:三角形 oval: 椭圆 heart: 心形 round: 圆形
++probability	否	double	置信度，范围【0~1】，代表这是人脸形状判断正确的概率，0 最小、1 最大。
+gender	否	array	性别，face_field 包含 gender 时返回
++type	否	string	male:男性 female:女性
++probability	否	double	性别置信度，范围【0~1】，0 代表概率最小、1 代表最大。
+glasses	否	array	是否带眼镜，face_field 包含 glasses 时返回

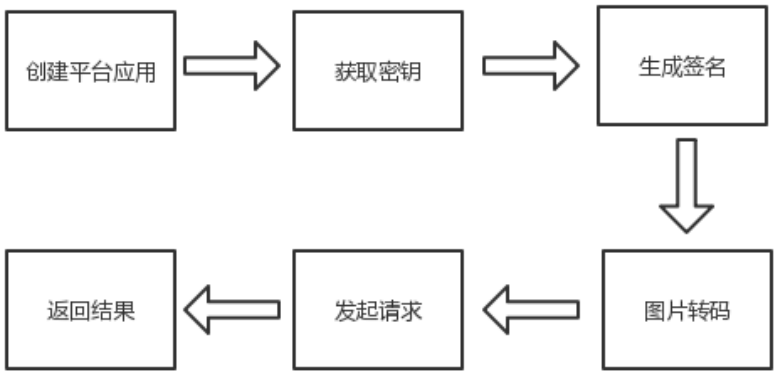
字段	必选	类型	说明
++type	否	string	none:无眼镜，common:普通眼镜，sun:墨镜
++probability	否	double	眼镜置信度，范围【0~1】，0 代表概率最小、1 代表最大。
+eye_status	否	array	双眼状态（睁开/闭合） face_field 包含 eye_status 时返回
++left_eye	否	double	左眼状态 [0,1]取值，越接近 0 闭合的可能性越大
++right_eye	否	double	右眼状态 [0,1]取值，越接近 0 闭合的可能性越大
+emotion	否	array	情绪 face_field 包含 emotion 时返回
++type	否	string	angry:愤怒 disgust:厌恶 fear:恐惧 happy:高兴 sad:伤心 surprise:惊讶 neutral:无情绪
++probability	否	double	情绪置信度，范围 0~1
+race	否	array	人种 face_field 包含 race 时返回

字段	必选	类型	说明
++type	否	string	yellow: 黄种人 white: 白种人 black:黑种人 arabs: 阿拉伯人
++probability	否	double	人种置信度，范围【0~1】，0 代表概率最小、1 代表最大。
+face_type	否	array	真实人脸/卡通人脸 face_field 包含 face_type 时返回
++type	否	string	human: 真实人脸 cartoon: 卡通人脸
++probability	否	double	人脸类型判断正确的置信度，范围【0~1】，0 代表概率最小、1 代表最大。
+landmark	否	array	4 个关键点位置，左眼中心、右眼中心、鼻尖、嘴中心。face_field 包含 landmark 时返回
+landmark72	否	array	72 个特征点位置 face_field 包含 landmark72 时返回
+landmark150	否	array	150 个特征点位置 face_field 包含 landmark150 时返回
+quality	否	array	人脸质量信息。face_field 包含 quality 时返回

字段	必选	类型	说明
++occlusion	否	array	人脸各部分遮挡的概率，范围[0~1]，0 表示完整，1 表示不完整
+++left_eye	否	double	左眼遮挡比例，[0-1] ，1 表示完全遮挡
+++right_eye	否	double	右眼遮挡比例，[0-1] ， 1 表示完全遮挡
+++nose	否	double	鼻子遮挡比例，[0-1] ， 1 表示完全遮挡
+++mouth	否	double	嘴巴遮挡比例，[0-1] ， 1 表示完全遮挡
+++left_cheek	否	double	左脸颊遮挡比例，[0-1] ， 1 表示完全遮挡
+++right_cheek	否	double	右脸颊遮挡比例，[0-1] ， 1 表示完全遮挡
+++chin	否	double	下巴遮挡比例，，[0-1] ， 1 表示完全遮挡

字段	必选	类型	说明
++blur	否	double	人脸模糊程度，范围[0~1]，0 表示清晰，1 表示模糊
++illumination	否	double	取值范围在[0~255]，表示脸部区域的光照程度越大表示光照越好
++completeness	否	int64	人脸完整度，0 或 1，0 为人脸溢出图像边界，1 为人脸都在图像边界内

1.19.5 程序逻辑



1.19.6 接口

函数名称	参数类型	返回值类型	备注
get_faces	image url of file:	Json	

1.19.7 存储分配

未知

1.19.8 限制条件

网络保持畅通，一秒内发出请求不可超过两次。若图片中无人脸应捕捉异常。

1.20 图片分析模块

1.20.1 功能描述

对从监控端发来的图片进行智能分析，做出人脸检测、摔倒检测、交互检测、 情感分析、禁止区域入侵检测的判断，并形成相关数据报告入库。

1.20.2 性能描述

对每张图片的整套分析不超过 1s

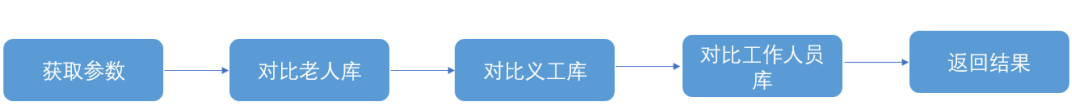
1.20.3 输入

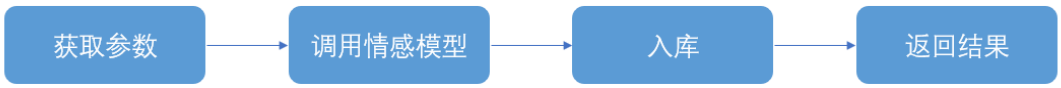
陌生人检测：身体图片、面部图片
情感检测：面部图片、老人 id、分析方法
摔倒检测：身体图片、摔倒角度阈值、分析方法
交互检测：图片中老人及志愿者列表、图片路径
综合分析：图片、分析方法

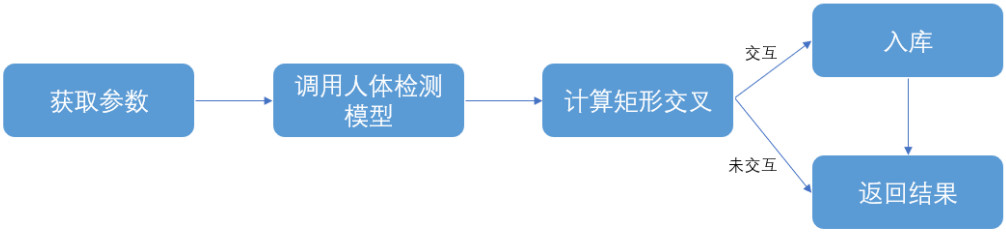
1.20.4 输出

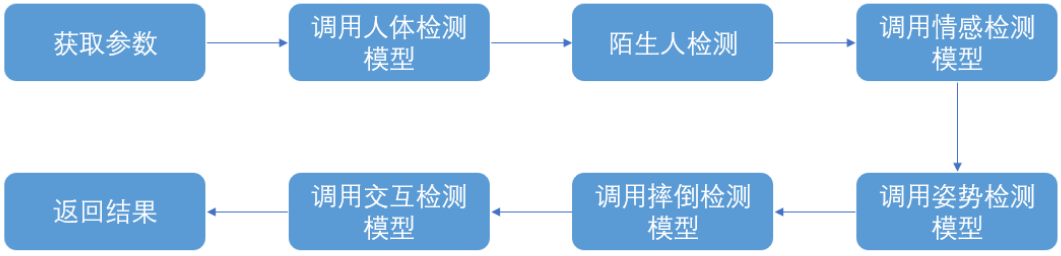
字段	必选	类型	说明
status	是	string	请求返回状态
code	是	int	正确/错误码
content	否	json	必要的返回内容

1.20.5 程序逻辑









1.20.6 接口

函数名称	参数	返回值	备注
strange	body,face	result	
emotion	face,elder,way	None	

fall	body,degree,way	None	
interaction	record,img	None	
analysis	dir,way	None	

1.20.7 限制条件

图片没有损坏

1.21 用户管理模块

1.21.1 功能描述

管理系统用户的账户相关维护操作，如添加用户、登陆、登出、修改密码等。

1.21.2 性能描述

无

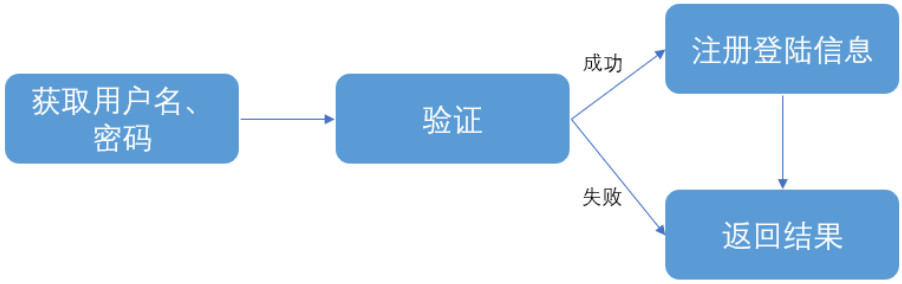
1.21.3 输入

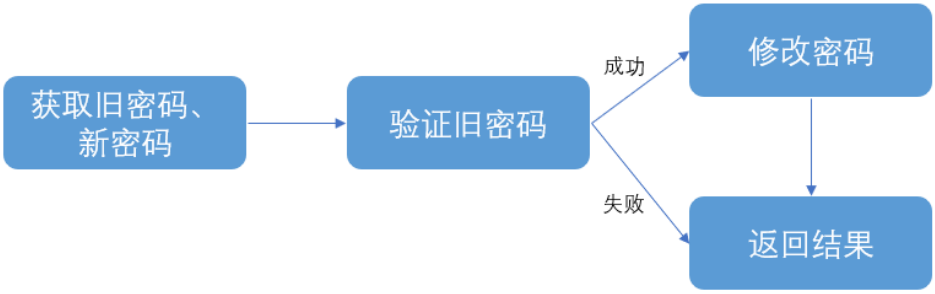
登陆：邮箱、密码
登出：含登陆信息的 request
修改密码：邮箱、旧密码、新密码

1.21.4 输出

字段	必选	类型	说明
status	是	string	请求返回状态
code	是	int	正确/错误码
content	否	json	必要的返回内容

1.21.5 程序逻辑





1.21.6 接口

函数名称	参数类型	返回值类型	备注
Login	用户名、密码	Json	
Logout	用户名	Json	
modifyPwd	旧密码、新密码	Json	

1.21.7 限制条件

无

1.22 老人管理模块

1.22.1 功能描述

对数据库中的老人信息进行增删改查管理。

1.22.2 性能描述

无

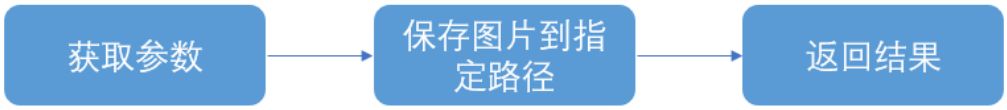
1.22.3 输入

上传照片：老人 id，照片
添加老人：姓名、手机号码、生日、性别、身份证号、家属姓名、家属电话
删除老人：老人 id
更新老人：姓名、手机号码、生日、性别、身份证号、家属姓名、家属电话
初始化信息：无

1.22.4 输出

字段	必选	类型	说明
status	是	string	请求返回状态
code	是	int	正确/错误码
content	否	json	必要的返回内容

1.22.5 程序逻辑



1.22.6 接口

函数名称	参数类型	返回值类型	备注
upload_photo	id,file	Json	
add	name 、 phone 、 birthday 、 gender、id_card、	Json	

	fam_name 、fam_phone		
remove	id	Json	
update	name 、 phone 、birthday 、gender、id_card、fam_name 、fam_phone	Json	
init	None	Json	

1.22.7 限制条件

无

1.23 websocket 模块

1.23.1 功能描述

浏览器与服务器间的 websocket 通讯管理。

1.23.2 性能描述

无

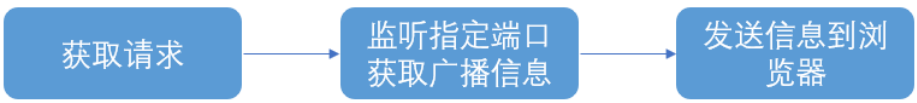
1.23.3 输入

Request

1.23.4 输出

无

1.23.5 程序逻辑



1.23.6 接口

函数名称	参数类型	返回值类型	备注
msg	request	None	建立连接
data	request	None	建立连接

1.23.7 限制条件

客户端主动发送 websocket 连接请求。