

酒店视觉AI解决方案

详细方案

|  |  |  |
| --- | --- | --- |
| 文件状态：  [√]草稿  [ ]正式发布  [ ]修改中 | **文件标识：** | 酒店视觉AI解决方案——详细方案 |
| **当前版本：** | V1.0 |
| **作者：** | 王晨旭，张天淼、邓国灏、杨晨光 |
| **完成日期：** | 2019.12.22 |

修改变更记录：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本号 | 更改条款及内容 | 更改人 | 审批人 | 更改日期 |
| v0.1 | 创建初稿 | 王晨旭、张天淼、邓国灏、杨晨光 | 王晨旭 | 2019.12.13 |
| V1.0 | 完善 | 张天淼、王晨旭 | 王晨旭 | 2019.12.22 |

目录

**[1 目标与服务模型](#_Toc1496489473_WPSOffice_Level1)** **[5](#_Toc1496489473_WPSOffice_Level1)**

[1.1赛题价值](#_Toc237994539_WPSOffice_Level2) [5](#_Toc237994539_WPSOffice_Level2)

[1.2解决思路](#_Toc1359666259_WPSOffice_Level2) [6](#_Toc1359666259_WPSOffice_Level2)

**[2组织管理与业务分析方案](#_Toc237994539_WPSOffice_Level1)** **[7](#_Toc237994539_WPSOffice_Level1)**

**[2.1项目来源](#_Toc1359666259_WPSOffice_Level1)** **[7](#_Toc1359666259_WPSOffice_Level1)**

[项目名称：酒店视觉AI解决方案](#_Toc537327286_WPSOffice_Level2) [7](#_Toc537327286_WPSOffice_Level2)

[委托单位：虹软科技股份有限公司](#_Toc690960167_WPSOffice_Level2) [7](#_Toc690960167_WPSOffice_Level2)

[项目用户：酒店经营人员和酒店管理人员](#_Toc1523447440_WPSOffice_Level2) [7](#_Toc1523447440_WPSOffice_Level2)

[项目建设的背景：](#_Toc133600899_WPSOffice_Level2) [7](#_Toc133600899_WPSOffice_Level2)

**[2.2组织管理](#_Toc537327286_WPSOffice_Level1)** **[7](#_Toc537327286_WPSOffice_Level1)**

[2.2.1团队成员](#_Toc1309898378_WPSOffice_Level2) [7](#_Toc1309898378_WPSOffice_Level2)

[2.2.2沟通方式](#_Toc1607173649_WPSOffice_Level2) [7](#_Toc1607173649_WPSOffice_Level2)

[2.2.3安排与进度](#_Toc718206777_WPSOffice_Level2) [8](#_Toc718206777_WPSOffice_Level2)

**[2.3业务分析](#_Toc690960167_WPSOffice_Level1)** **[8](#_Toc690960167_WPSOffice_Level1)**

[2.3.1功能性需求](#_Toc2043204899_WPSOffice_Level2) [8](#_Toc2043204899_WPSOffice_Level2)

[2.3.2非功能性需求](#_Toc1881221963_WPSOffice_Level2) [9](#_Toc1881221963_WPSOffice_Level2)

**[3技术路线和实现方案](#_Toc1523447440_WPSOffice_Level1)** **[10](#_Toc1523447440_WPSOffice_Level1)**

[分为](#_Toc295797360_WPSOffice_Level2) [10](#_Toc295797360_WPSOffice_Level2)

[酒店入住](#_Toc41586715_WPSOffice_Level2) [10](#_Toc41586715_WPSOffice_Level2)

[酒店退宿](#_Toc1015733730_WPSOffice_Level2) [11](#_Toc1015733730_WPSOffice_Level2)

[客房进出](#_Toc1089290107_WPSOffice_Level2) [11](#_Toc1089290107_WPSOffice_Level2)

[酒店用餐](#_Toc400737674_WPSOffice_Level2) [11](#_Toc400737674_WPSOffice_Level2)

[人流分析等模块](#_Toc689369926_WPSOffice_Level2) [11](#_Toc689369926_WPSOffice_Level2)

[3.1酒店入住](#_Toc566070717_WPSOffice_Level2) [13](#_Toc566070717_WPSOffice_Level2)

[3.2酒店退宿](#_Toc597984409_WPSOffice_Level2) [16](#_Toc597984409_WPSOffice_Level2)

[3.3客房进出](#_Toc100494103_WPSOffice_Level2) [18](#_Toc100494103_WPSOffice_Level2)

[3.4餐厅消费](#_Toc1082242579_WPSOffice_Level2) [19](#_Toc1082242579_WPSOffice_Level2)

[3.5用户注册](#_Toc64535163_WPSOffice_Level2) [22](#_Toc64535163_WPSOffice_Level2)

[风险管理](#_Toc163242806_WPSOffice_Level2) [25](#_Toc163242806_WPSOffice_Level2)

[风险类别](#_Toc1285223223_WPSOffice_Level2) [26](#_Toc1285223223_WPSOffice_Level2)

[风险影响定义](#_Toc1356187435_WPSOffice_Level2) [26](#_Toc1356187435_WPSOffice_Level2)

[风险概率定义](#_Toc50790787_WPSOffice_Level2) [27](#_Toc50790787_WPSOffice_Level2)

[实施定性，定量风险分析](#_Toc1089749250_WPSOffice_Level2) [27](#_Toc1089749250_WPSOffice_Level2)

[风险应对](#_Toc1675103134_WPSOffice_Level2) [27](#_Toc1675103134_WPSOffice_Level2)

[风险监督](#_Toc2095244615_WPSOffice_Level2) [27](#_Toc2095244615_WPSOffice_Level2)

[4.5可行性总结](#_Toc339400799_WPSOffice_Level2) [27](#_Toc339400799_WPSOffice_Level2)

**[5总结](#_Toc133600899_WPSOffice_Level1)** **[28](#_Toc133600899_WPSOffice_Level1)**

# 目标与服务模型

## 1.1赛题价值

### 1.1.1概述

随着人脸识别技术越来越成熟，人脸识别不断的被用于各行各业中，如银行、居住区、城市、校园等场景。为了更好的让AI技术在实际生活中落地，让更多的人能够用上AI技术，虹软对外发布了开放平台产品，现已经被各行各业采用。如今在旅游行业，人们生活水平不断的提升，旅游出行或公务出行时，对酒店的服务升级越越来越期待。因此发包方希望能够给出一个视觉AI+酒店的解决方案。

### 1.1.2价值分析

以下分别从酒店与公司产品的用户角度来分析该赛题的价值：

|  |  |  |
| --- | --- | --- |
| 公司 | | |
| 价值 | 方式 | 说明 |
| 经济效益 | 直接获益。通过AI+酒店的方式，提高了入住、退房等效率，提高了收益。 | 随着社会日新月异的发展，人们生活节奏逐渐加快，一个好的智能酒店系统，可以很好的减少人工繁琐的工作，改善酒店入住、退房等环节，便能够给酒店带来经济收益。 |
| 间接获益。AI+酒店的方式能够大大提高酒店运营效率，减少在各个环节所需的人工成本。 | AI+酒店是一个基于人脸识别的智能化系统，大大减少了工作人员的工作量，节省了劳动力，降低了经济开销 |
| 社会效益 | 良好的功能，低廉的开发成本，同时能使酒店的安全性大大提高。 | 快捷，方便的使用产品，给用户舒适方便的使用体验，给用户营造安全的住宿环境，从而对公司有深刻的印象，也服务社会。 |
| 价值 | 说明 | |
| 形象价值 | 酒店的智能化相较于传统酒店效率更高，更加方便，安全可靠，良好的系统有效地提高了公司的形象。 | |
| 方便快捷 | 客户可以通过人脸识别进行入住、退房、进入房间等操作，不存在丢失房卡等问题，而工作人员也可以根据后台数据对人流进行分析。 | |
| 安全可靠 | 基于vue，mysql，java，的开发，保证了系统的安全可靠。 | |

通过前期发放问卷，对用户需求进行调查，我们得到了如下结果：

|  |  |
| --- | --- |
| 用户 | |
| 价值 | 说明 |
| 服务价值 | 智能酒店相比传统的酒店更加的方便快捷，只需要通过人脸识别，便可完成许多操作。提升了用户的工作效率和体验，也方便了酒店方的管理。 |
| 使用价值 | 对于用户来说，有较高的使用价值。减少排队时间，增加安全性，就餐购物更加方便，避免了忘带房卡、丢失餐券等情况。 |
| 方便快捷 | 减少了登记环节，效率更高，同时通过我们的调查也可以发现用户更倾向于使用终端服务机而不是面向前台服务人员。 |

## 1.2解决思路

我方根据“目标，业务，设计，开发，结题”的总体思路安排项目计划，严格在规定时间内完成规定步骤。以下五个阶段为酒店视觉AI解决方案的详细解决方案：

目标阶段：我方从用户酒店应用场景的角度去分析酒店视觉AI解决方案，构思系统蓝图，根据赛题给出的需求，得出基本目标，并分析出系统应具有的核心价值。

业务分析阶段：我方通过深刻分析赛题，初步了解需求，并发放问卷挖掘需求，然后分析出酒店视觉AI系统的功能模块，初步细化每个单元功能模块中的详细功能，再深入挖掘酒店视觉AI系统的核心价值，针对每个价值点进行成本预估和可行性分析，同时做好风险预估和规避方案，进而完成系统的总体设计方案。

系统设计阶段：我方通过业务分析阶段得出的结果，推出系统架构图以对模块界限进行清晰划分，力求能最大限度的保证系统的低耦合高内聚。我方对系统的基本处理流程、系统的组织结构、模块划分、功能分配、接口设计、运行设计、数据结构设计、出错处理设计和性能设计进行了详细的讨论，做到足够的细致，整合出一份稳定可行、可进行编码的技术路线方案。

系统开发阶段：紧扣技术路线方案，对系统进行前后端分离，减轻服务器压力，提高服务器性能。对各个功能做到力求减轻服务器压力，模块和模块之间进行组件化开发，各个模块之间真正的低耦合。尽可能将模块的功能规范，数据采集统一，语言描述统一，资源共享，保证各模块协调一致地工作，将信息的冗余程度降至最低。同时保证各模块之间接口的预留与互通，将系统功能实现并一步步完善。

结题阶段：经过大量测试、完善，发现该系统已基本能够提供稳定的服务，然后将成本可行性方案、目标服务模型、业务方案、技术路线设计方案进行有机整合，形成解决方案，并将系统投入使用。

# 2组织管理与业务分析方案

## 2.1项目来源

**项目名称：**酒店视觉AI解决方案

**委托单位：**虹软科技股份有限公司

**项目用户：**酒店经营人员和酒店管理人员

**项目建设的背景：**

随着人脸识别技术越来越成熟，人脸识别不断的被用于各行各业中，如银行、居住区、城市、校园等场景。为了更好的让AI技术在实际生活中落地，让更多的人能够用上AI技术，虹软对外发布了开放平台产品，现已经被各行各业采用。如今在旅游行业，人们生活水平不断的提升，旅游出行或公务出行时，对酒店的服务升级越越来越期待。因此发包方希望能够给出一个视觉AI+酒店的解决方案。

## 2.2组织管理

### 2.2.1团队成员

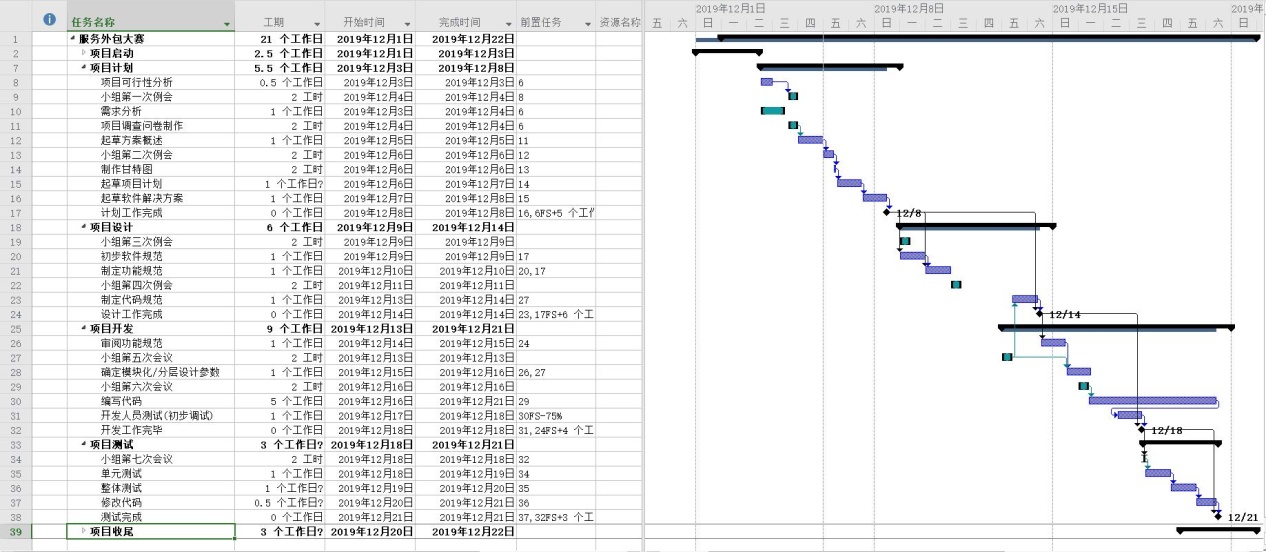
|  |  |
| --- | --- |
| **成员名称** | **任务分工** |
| 王晨旭 | 项目经理 |
| 张天淼 | 技术经理 |
| 杨晨光 | 客户经理 |
| 邓国灏 | UI工程师 |

### 2.2.2沟通方式

|  |  |
| --- | --- |
| **沟通方式** | **优点** |
| 项目会议 | 1. 确立下阶段的目标，明确下阶段每个人的任务 2. 提出上阶段的问题，协同解决，查漏补缺 3. 各个成员之间相互促进、监督 |
| 文档 | 1. 便于管理，查阅各种文档信息 2. 可以记录工作进度 3. 各个阶段的文档输出方便组内成员的评审与查漏补缺 |
| 小组讨论与团建 | 1. 可以加强成员间的默契，可以解决成员间的矛盾与问题 2. 方便大家查漏补缺，互相之间给予建议，使产品更加完善 |
| 微信群联系 | 1. 方便联系，大家工作时都在线 2. 可以更快的文件交流 3. 可以查看历史聊天记录，很好地弥补了信息遗漏的问题 |

### **2.2.3安排与进度**

根据工程项目的进度目标，我方团队按照甘特图进行项目的分析、设计、开发以及测试等步骤，严格按照服务外包项目流程进展，并据以检查工程项目进度计划的执行情况，若发现实际执行情况与计划进度不一致，就及时分析原因，并采取必要的措施。



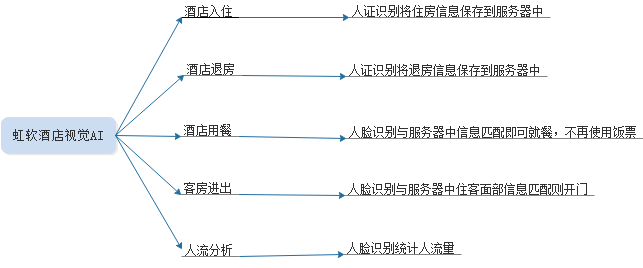
## 2.3业务分析

### 2.3.1功能性需求

酒店视觉AI系统旨在达到以下目标：

利用虹软人脸识别相关技术，以提升酒店服务体验为核心，形成一个完善的解决方案，结合酒店场景，打造不一样的酒店服务。从酒店入住、退房、客房进出、酒店用餐、人流分析等方面进行应用。

|  |  |
| --- | --- |
| **功能** | **分析** |
| 酒店入住 | 酒店视觉AI系统可以通过人证识别扫描房客面部和身份证即可实现入住，不用再通过前台服务人员登记，并将住房信息保存到服务器中。 |
| 退房 | 酒店视觉AI系统可以通过人证识别扫描房客面部和身份证即可实现入住，不用再通过前台服务人员登记，并将退房信息保存到服务器中。 |
| 客房进出 | 通过人脸识别的功能进行刷脸，与服务器中住客面部信息匹配则开门，不再使用房卡 |
| 酒店用餐 | 通过人脸识别的功能刷脸，与服务器中信息匹配即可就餐，不再使用饭票。 |
| 人流分析 | 通过人脸识别的功能统计人流量，统计酒店各个位置每日的人数。 |

****

### 2.3.2非功能性需求

**2.3.2.1安全性**

本系统面向酒店管理人员，所以数据的安全性显得尤为重要，任何数据被盗用都会造成严重的影响。因此，本系统的使用应该具备安全可靠性，进行加密处理，确保数据不泄露，并且具备优良的抵抗外部侵袭的能力，来确保系统能够正常地运作。

**2.3.2.2美观性**

良好的人机交互、严密的操作逻辑和简洁美观、有亲和力的界面整体设计往往更受用户的欢迎，所以该系统应使界面变得有个性化，使操作变得舒适、简单、自由。

**2.3.2.3直观性**

对于一个数据分析系统而言，往往将庞大的数据精简化，以一种直观乐于接受的方式将分析的结果展示出来，所以为了使使用者能够从该系统中直观地获取相应信息，本系统的展现形式应该具备直观性。

**2.3.2.4可变性**

本系统面向的用户主要是酒店管理人员，对于人脸识别、认证识别方面，需要一等的准确性，而系统设计的识别方面上会存在错误和不准确。所以为了长远考虑，本系统的设计必须灵活且可扩展，以便在出现不准确使，及时对系统强化学习，做出调整，不断满足新的筛查需求。

# 3技术路线和实现方案



分为

**酒店入住**

**酒店退宿**

**客房进出**

**酒店用餐**

**人流分析**等模块

本次使用的虹软提供的人脸识别的SDK，此SDK也可根据不同应用场景设计，针对性强。包括人脸检测、人脸跟踪、人脸识别，即使在离线环境下也可正常运行。

Face Detection 人脸检测，使用虹软的人脸引擎，帮助我们检测并定位到影像（视频或图片）中的人脸

Face recognition 人脸识别，该引擎独立运行在终端设备和服务器中，应用端可独立完成算法运行，保证用户数据的私密性，自主运营和保护用户敏感信息。

Face Tracking 人脸跟踪。精确定位并追踪面部区域位置，随着人物脸部位置的变化能够快速定位人脸位置，并且适用于不同表情、性别、年龄、姿态、光照等条件。

AFR\_FSDKInterface engine = new AFR\_FSDKEngine();

    //用来存放提取到的人脸信息, face\_1 是注册的人脸，face\_2 是要识别的人脸

    AFR\_FSDKFace face1 = new AFR\_FSDKFace();

    AFR\_FSDKFace face2 = new AFR\_FSDKFace();

    //初始化人脸识别引擎，使用时请替换申请的 APPID 和 SDKKEY

    AFR\_FSDKError error = engine.AFR\_FSDK\_InitialEngine("APPID", "SDKKEY");

    Log.d("com.arcsoft", "AFR\_FSDK\_InitialEngine = " + error.getCode());

    //输入的 data 数据为 NV21 格式（如 Camera 里 NV21 格式的 preview 数据）；人脸坐标一般使用人脸检测返回的 Rect 传入；人脸角度请按照人脸检测引擎返回的值传入。

    error = engine.AFR\_FSDK\_ExtractFRFeature(data1, width, height, AFR\_FSDKEngine.CP\_PAF\_NV21, new Rect(210, 178, 478, 446), AFR\_FSDKEngine.AFR\_FOC\_0, face1);

    Log.d("com.arcsoft", "Face=" + face1.getFeatureData()[0]+ "," + face1.getFeatureData()[1] + "," + face1.getFeatureData()[2] + "," + error.getCode());

    error = engine.AFR\_FSDK\_ExtractFRFeature(data1, width, height, AFR\_FSDKEngine.CP\_PAF\_NV21, new Rect(210, 170, 470, 440), AFR\_FSDKEngine.AFR\_FOC\_0, face2);

    Log.d("com.arcsoft", "Face=" + face2.getFeatureData()[0]+ "," + face2.getFeatureData()[1] + "," + face2.getFeatureData()[2] + "," + error.getCode());

    //score 用于存放人脸对比的相似度值

    AFR\_FSDKMatching score = new AFR\_FSDKMatching(); error = engine.AFR\_FSDK\_FacePairMatching(face1, face2, score); Log.d("com.arcsoft", "AFR\_FSDK\_FacePairMatching=" + error.getCode()); Log.d("com.arcsoft", "Score:" + score.getScore());

    //销毁人脸识别引擎

    error = engine.AFR\_FSDK\_UninitialEngine();

    Log.d("com.arcsoft", "AFR\_FSDK\_UninitialEngine : " + error.getCode());

在这里，data数据的格式使用NV21格式。

在人脸检测中我们在相机中直接预览，以下的camera回调

    mCamera.setPreviewCallback(new Camera.PreviewCallback() {

        @Override

        public void onPreviewFrame(byte[] data, Camera camera) {

           //这里的data数据就是NV21格式，可以在这里处理人脸检测

        }

    });

此外，android中对图片的处理接口使用BitMap，RGB转NV21：

    public static void encodeYUV420SP(byte[] yuv420sp, int[] argb, int width, int height) throws Exception {

        final int frameSize = width \* height;

        int yIndex = 0;

        int uvIndex = frameSize;

        int a, R, G, B, Y, U, V;

        int index = 0;

        for (int j = 0; j < height; j++) {

            for (int i = 0; i < width; i++) {

                a = (argb[index] & 0xff000000) >> 24; // a is not used obviously

                R = (argb[index] & 0xff0000) >> 16;

                G = (argb[index] & 0xff00) >> 8;

                B = (argb[index] & 0xff) >> 0;

                // well known RGB to YUV algorithm

                Y = ((66 \* R + 129 \* G + 25 \* B + 128) >> 8) + 16;

                U = ((-38 \* R - 74 \* G + 112 \* B + 128) >> 8) + 128;

                V = ((112 \* R - 94 \* G - 18 \* B + 128) >> 8) + 128;

                // NV21 has a plane of Y and interleaved planes of VU each sampled by a factor of 2

                //    meaning for every 4 Y pixels there are 1 V and 1 U.  Note the sampling is every other

                //    pixel AND every other scanline.

                yuv420sp[yIndex++] = (byte) ((Y < 0) ? 0 : ((Y > 255) ? 255 : Y));

                if (j % 2 == 0 && index % 2 == 0) {

                    yuv420sp[uvIndex++] = (byte) ((V < 0) ? 0 : ((V > 255) ? 255 : V));

                    yuv420sp[uvIndex++] = (byte) ((U < 0) ? 0 : ((U > 255) ? 255 : U));

                }

                index++;

            }

        }

    }

## 3.1酒店入住

利用虹软提供的人脸识别、认证验核等功能SDK，android终端机摄像头采集视频流，先进行人脸的预检测，提取人脸特征上传到后端服务器，与服务器中人脸库进行对比

未匹配——未预定房间/未注册该用户

跳转在线订房页面

跳转注册页面

匹配成功，输出本用户的酒店订单，房型、房间号、订购的住宿时间、价格等信息，更改后端数据库订单表的入住状态；

在初始化人脸检测引擎 FT中，首先进行初始化

        Log.d(TAG, "AFT\_FSDK\_InitialFaceEngine =" + err.getCode());

        err = engine.AFT\_FSDK\_GetVersion(version);

        Log.d(TAG, "AFT\_FSDK\_GetVersion:" + version.toString() + "," + err.getCode());

在摄像头预览函数中，首先调用FT的人脸识别函数，再调用FR的人脸特征提取函数

AFT\_FSDKError err = engine.AFT\_FSDK\_FaceFeatureDetect(data, width, height, AFT\_FSDKEngine.CP\_PAF\_NV21, result);

AFR\_FSDKError error = engine.AFR\_FSDK\_ExtractFRFeature(mImageNV21, mWidth, mHeight, AFR\_FSDKEngine.CP\_PAF\_NV21,mAFT\_FSDKFace.getRect(), mAFT\_FSDKFace.getDegree(), result);

在结果中保存人脸特征信息

在人脸检测预览中，我们框出检测到的人脸信息

Canvas canvas = mSurfaceHolder.lockCanvas();

  if (canvas != null) {

     Paint mPaint = new Paint();

     boolean fit\_horizontal = canvas.getWidth() / (float)src.width() < canvas.getHeight() / (float)src.height() ? true : false;

     float scale = 1.0f;

     if (fit\_horizontal) {

        scale = canvas.getWidth() / (float)src.width();

        dst.left = 0;

        dst.top = (canvas.getHeight() - (int)(src.height() \* scale)) / 2;

        dst.right = dst.left + canvas.getWidth();

        dst.bottom = dst.top + (int)(src.height() \* scale);

     } else {

        scale = canvas.getHeight() / (float)src.height();

        dst.left = (canvas.getWidth() - (int)(src.width() \* scale)) / 2;

        dst.top = 0;

        dst.right = dst.left + (int)(src.width() \* scale);

        dst.bottom = dst.top + canvas.getHeight();

     }

     canvas.drawBitmap(mBitmap, src, dst, mPaint);

     canvas.save();

     canvas.scale((float) dst.width() / (float) src.width(), (float) dst.height() / (float) src.height());

     canvas.translate(dst.left / scale, dst.top / scale);

     for (AFD\_FSDKFace face : result) {

        mPaint.setColor(Color.RED);

        mPaint.setStrokeWidth(10.0f);

        mPaint.setStyle(Paint.Style.STROKE);

        canvas.drawRect(face.getRect(), mPaint);

     }

     canvas.restore();

     mSurfaceHolder.unlockCanvasAndPost(canvas);

     break;

  }

}



## 3.2酒店退宿

利用虹软提供的人脸识别、认证验核等功能SDK，android终端机摄像头采集视频流，先进行人脸的预检测，提取人脸特征上传到后端服务器，与服务器中人脸库进行对比

未匹配——未预定房间/未注册该用户

提示未预定房间/未注册

匹配成功，输出本用户的酒店订单，包括本次入住所有消费入住；

酒店系统全部采用无现金、人脸识别记入账单的形式，所以消费在退房时结算。

跳转结算页面，通过支付宝/微信接口跳转支付

支付完毕后

更改该订单状态为完成，提示退房成功；



## 3.3客房进出

利用虹软提供的人脸识别、认证验核等功能SDK，android终端机摄像头采集视频流，先进行人脸的预检测，提取人脸特征上传到后端服务器，与服务器中人脸库进行对比，匹配输出用户信息，将本编号门禁开门时间存入后端数据库从而方便安保管理。并通过配套的智能门锁硬件打开本编号门禁；



## 3.4餐厅消费

利用虹软提供的人脸识别、认证验核等功能SDK，android终端机摄像头采集视频流，先进行人脸的预检测，提取人脸特征上传到后端服务器，与服务器中人脸库进行对比，匹配输出用户信息；

当本用户账户存在可使用餐券时，提示用户是否选择使用餐券，使用扣除餐券数量，不使用将本笔消费记录账单，录入数据库订单表；

支付成功后输出成功提示；



## 3.5用户注册

在终端机上选择注册页面，利用虹软提供的人脸识别、认证验核等功能SDK，android终端机摄像头采集视频流，从而获取注册图像，

先进行人脸的预检测，实现人脸检测功能，

对于静态图片，虹软SDK对应FD，提供方法AFD\_FSDK\_StillImageFaceDetection，检测到的人脸将会被放入此列表，

AFD\_FSDKFaces是人脸识别的结果，定义为

  public class AFD\_FSDKFace {

        Rect mRect;

        int mDegree;

        }

将获取到的NV21图像转化为Bitmap后，调用AFD\_FSDK\_StillImageFaceDetection

提取人脸特征上传到后端服务器，输入注册信息，将该用户保存在人脸库中，

输出提示注册成功；

人脸信息保存在AFR\_FSDKFace类

public static final int FEATURE\_SIZE = 22020;

    byte[] mFeatureData;

进行注册时，我们需要定义另一个类将人脸信息和用户名对应，例如

    class FaceRegist {

        String mName;

        List<AFR\_FSDKFace> mFaceList;

        public FaceRegist(String name) {

            mName = name;

            mFaceList = new ArrayList<>();

        }

    }



# 成本模型及可行性分析

## 4.1引言

本文档主要针对开发人员和项目经理编写的。可行性分析报告的目的，就是能用最小的代价在尽可能短的时间内确定问题是否能够解决，为开始这个项目提供有效的目标和方向。该文档分别从经济可行性，技术可行性和操作可行性上分析。

## 4.2成本模型

成本预测：

|  |  |  |
| --- | --- | --- |
| 类别 | 项目 | 合计（万元） |
| 固定资产，设备投资 | 场地租赁 | 0 |
| 办公设备 | 0.5 |
| 网站建设 | 域名 | 0.05 |
| 云服务器 | 0.6 |
| 云数据库 | 0.3 |
| 网络费 | 0.2 |
| 工资 | 人员工资 | 2.4 |
| 耗材 | 材料费 | 0.1 |

成本分析:

本项目成本主要在于人员成本以及服务器成本上，可通过在项目进行中可使用过程管理手段来进行控制

## 4.3可行性分析

### 4.3.1经济可行性

软件开发均使用开源组件，不需要授权费。软件编写玩抽，后期维护成本较低，且扩展性强。

### 4.3.2技术可行性

软件的开发运用Java、VUE、JavaScript等技术，可用教学资源较多，且能够有效使用虹软所提供的SDK，小组技术人员在该方面也都具有一定学习经验，因此技术方面可行。

### 4.3.3社会因素可行性

如今酒店人脸识别日志已经成为继之前的车牌自动识别一样，被社会所广泛需求。可以极大的提升酒店的安全性和管理的高效性，对于使用者也是更加的便捷。

### 4.3.4人员可行性

指导老师：

罗荣良：本专业资深教师，拥有丰富教学经验，待人友善。

小组成员：

王晨旭：项目经理，拥有带领团队的经验，在团队管理上颇有心得

张天淼：技术经理，拥有丰富的项目工程经验

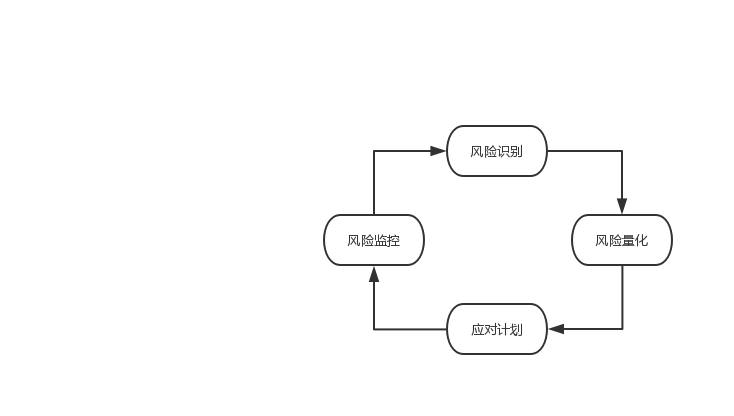
杨晨光：客户经理，与人沟通耐心，友善

邓国灏：UI工程师，能熟练运用PS

## 4.4风险评估：

**风险管理**

项目风险管理是指为了最好的达到项目的目标，识别、分配、应对项目生命周期内风险的科学与艺术。项目风险管理的目标是使潜在机会或回报最大化，使潜在风险最小化。风险管理涉及的主要过程包括：风险识别，风险量化，风险应对计划制定和风险监控，如下图所示。风险识别在项目的开始时就要进行，并在项目执行中不断进行。就是说，在项目的整个生命周期内，风险识别是一个连续的过程。



（1）风险识别：风险识别包括确定风险的来源，风险产生的条件，描述其风险特征和确定哪些风险事件有可能影响本项目。风险识别不是一次就可以完成的事，应当在项目的自始至终定期进行。

（2）风险量化：涉及对风险及风险的相互作用的评估，是衡量风险概率和风险对项目目标影响程度的过程。风险量化的基本内容是确定那些事件需要制定应对措施。

（3）风险应对计划制定：针对风险量化的结果，为降低项目风险的负面效应制定风险应对策略和技术手段的过程。风险应对计划依据风险管理计划、风险排序、风险认知等依据，得出风险应对计划、剩余风险、次要风险以及为其它过程提供得依据。

1. 风险监控：涉及整个项目管理过程中的风险进行应对。该过程的输出包括应对风险的纠正措施以及风险管理计划的更新。

**风险类别**

|  |  |  |
| --- | --- | --- |
| 序号 | 风险类别 | 描述 |
| 1 | 需求风险 | 在项目进展过程中，对于产品的需求存在不确定性 |
| 2 | 相关性风险 | 许多风险都是由外部环境或因素相关性产生 |
| 3 | 技术风险 | 开发人员没有熟悉开发中需要的软件技术 |
| 4 | 管理风险 | 项目管理出现问题 |
| 5 | 产品规模风险 | 项目产品规模越大，需求变更风险也越多 |
| 6 | 安全风险 | 软件开发时保密性也是一个重点 |

**风险影响定义**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 定性描述 | 进度 | 成本 | 质量 | 范围 |
| 影响 | 高 | 进度延期时间在1个月以上 | 成本超支20% | 项目明确表示项目与实际目标偏差非常大 | 每月变更大于8起 |
| 中 | 进度延期时间在半个月至一个月 | 成本超支10%～20% | 客户明确表示项目与实际目标偏差大 | 每月大于3起小于8起 |
| 低 | 进度延期时间在一周以内 | 成本超支小于10% | 客户明确表示项目需要少量修改 | 每月变更小3起 |

**风险概率定义**

|  |  |  |
| --- | --- | --- |
|  | 可能性 | 描述 |
| 概率 | 高 | 发生的概率大于70% |
| 中 | 发生的概率大于30%且低于70% |
| 低 | 发送的概率小于30% |

**实施定性，定量风险分析**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 风险名称 | 优先级 | 影响程度 | 概率 |
| 1 | 需求持续变化 | 中 | 中 | 低 |
| 2 | 软件开发周期过长 | 低 | 低 | 低 |
| 3 | 开发人员技术不足 | 低 | 低 | 低 |

**风险应对**

|  |  |  |
| --- | --- | --- |
| 风险名称 | 应对手段 | 负责人 |
| 需求持续变化 | 由用户访谈员确认后重新修改需求定义 | 项目经理 |
| 软件开发周期过长 | 项目经理更新软件开发计划 | 项目经理 |
| 开发人员技术不足 | 技术经理组织学习 | 技术经理 |

**风险监督**

|  |  |  |
| --- | --- | --- |
| 风险名称 | 控制手段 | 负责人 |
| 需求持续变化 | 由用户访谈员根据访谈结果更新修改需求定义 | 用户访谈员 |
| 软件开发周期过长 | 项目经理设定里程碑定期检查 | 项目经理 |
| 开发人员技术不足 | 技术经理定期检查软件使用情况 | 技术经理 |

## 4.5可行性总结

本系统基本具备可以操作的可行性。

# 5总结

本文档分析了开发中可能出现的问题以及对产品业务进行了思考。根据用户的需求，基于虹软提供的AI接口，在酒店入住、退宿、就餐、客房进出等方面提供了解决方案。本次项目研发过程中，各成员表现出了良好的精神面貌，各 成员之间积极配合，共同完成本项目的研发。