

教育部高等学校大学计算机课程教学指导委员会

# 中国大学生计算机设计大赛



## 软件开发类作品文档简要要求

作品编号： 66616

作品名称： MeetingManager

作 者： 张效凯、徐倩、张若兰

版本编号： V1.0.0

填写日期： 2019 年 5 月 16 日星期四

### 填写说明：

- 1、 本文档适用于**所有**涉及软件开发的作品，包括：软件应用与开发、大数据、人工智能、物联网应用；
- 2、 正文一律用五号宋体，一级标题为二号黑体，其他级别标题如有需要，可根据需要设置；
- 3、 本文档为简要文档，不宜长篇大论简明扼要为上；
- 4、 提交文档时，以 PDF 格式提交本文档；
- 5、 本文档内容是正式参赛内容组成部分，务必真实填写。如不属实，将导致奖项等级降低甚至终止本作品参加比赛。

## 目 录

第一章 需求分析 .....	3
第二章 概要设计 .....	3
第三章 详细设计 .....	4
第四章 测试报告 .....	15
第五章 安装及使用 .....	20
第六章 项目总结 .....	23

# 第一章 需求分析

会议室的预定、管理等问题，在企业、学校、园区等人员组织众多、会议众多的地方非常普遍。如何确定会议室在什么时间是空闲的，如何让会议室的预订更加方便、使用更加高效灵活，是一个急需解决的问题。我们以此为思路，思考怎么让会议室的管理更加高效，思考怎么让会议室的预定更加方便，于是就有了我们的作品“会议室管理系统”。

本作品主要面向企业，为企业一站式解决会议室管理问题。整个系统由 Web 端、App 端、门禁端和后台处理端组成，用户可以在 Web 预定端和 App 端查询、预定会议，并在门禁端开启会议室。系统嵌入虹软人脸识别 SDK，采用人脸识别验证用户身份，高效准确。系统使用智能化算法推荐会议室，用户可以更加准确的定位到自己的意向会议室。Web、App、门禁三端联合，会议室信息管理、订单管理、人员管理一体化。

市面上已有的产品与本系统对比如下：

已有产品	缺点	本系统解决方案
纯电脑端预定系统	使用不便 线下无匹配硬件	三端灵活预定 线下搭载智能硬件终端
指纹识别签到系统	线下硬件死板 验证偶尔失败	智能终端展示信息 人脸识别准确高效
门禁卡签到系统	卡片易丢失	生物信息验证，无额外物品
人工管理	效率低下	管理网络化、智能化

# 第二章 概要设计

系统按照功能可以分为预定端、管理端和后台管理端；按照技术可以分为 Web 端、Android APP 端、后台管理端、门禁端。为了让叙述更加清晰，我们以技术分类的角度介绍智能会议室管理系统。

## \*Web 端

Web 端分为 Web 预定端和 Web 管理端，两者使用统一的平台登录，根据权限的不同显示不同的页面。预定端主要功能有查询可用会议室、预定会议室、更改预定信息、修改用户个人信息等；管理端主要功能有处理用户注册信息、处理会议室预定信息、管理、维护数据库、管理会议室注册信息等。Web 端通过 HTTP 协议直接与后台管理端的 Web 服务器通信。

## \*Android APP 端

Android APP 端用于用户预定会议室，拥有上述 Web 预定端的全部功能，除此之外，还有人脸数据采集的功能。Android APP 端采取 Android+HTML5 的开发模式，也应用 HTTP 协议直接与后台管理端的 Web 服务器通信。

## \*后台管理端

后台管理端运行着一个 Web 服务器和一个数据库。此端的作用是处理其他三个端口的请求，并转换成数据库操作语言来操作数据库，更新数据库。

## \*门禁端

门禁端是硬件终端，安置在会议室门口。此端功能有管理会议室门的开关/灯/投影仪

等硬件信息、展示会议室预订信息、人脸识别验证身份信息等。门禁端通过直连数据库的方式与后台管理端通信。

会议室管理系统架构如图所示：

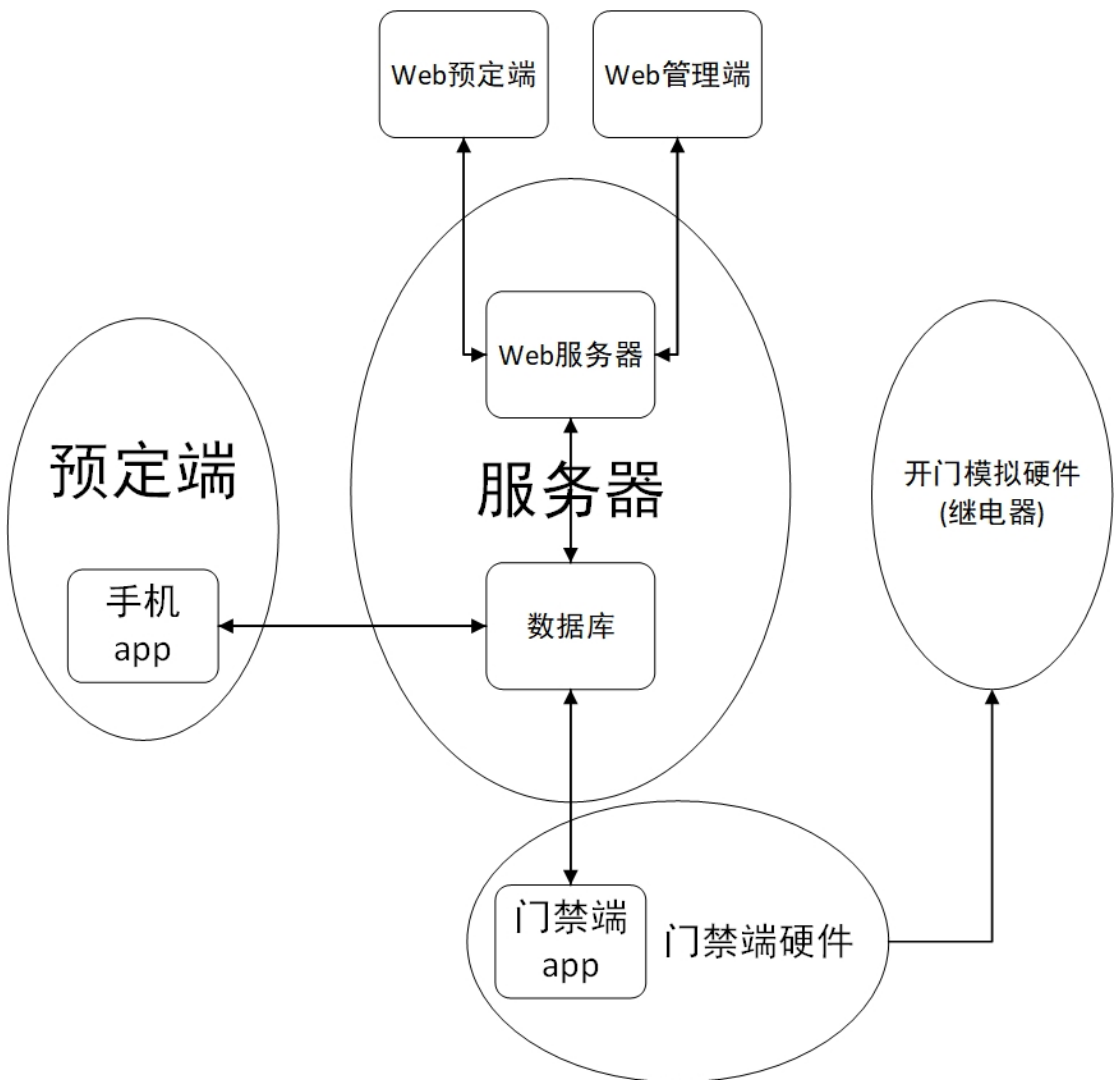


图 1 会议室管理系统架构

## 第三章 详细设计

### 1. Web 端详述

#### 1.1 Web 端综述

Web 端主要完成系统的线上注册和预定操作(如注册账户，注册人脸信息，预定会议室和检索订单等操作)。Web 端部署在搭载 Ubuntu-16.04.5 的 Linux 服务器上，服务器拥有公网 IP，项目部署后可以直接访问。Web 端人脸识别算法采用虹软公司的 SDK，允许用

户在网页上传照片注册人脸信息。

Web 端采用 Mysql 数据库，使用 Django 内建 ORM 进行读写数据，极大地方便了开发者对数据库表信息的增删改查。

Web 端系统架构如下：

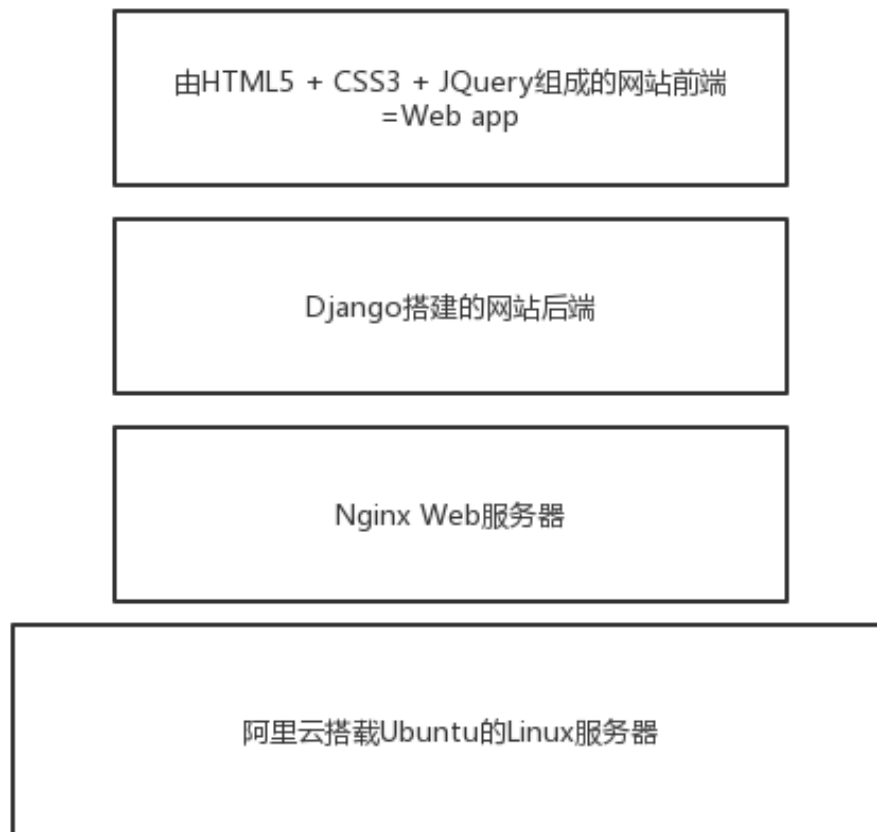


图 2 Web 端系统架构

## 1.2 Web 端详细模块

### 1.2.1 服务器

服务器采用阿里云的搭载 Ubuntu-16.04.5 的 Linux 服务器，该服务器带有公网 ip，有外部 ssh 接口，方便开发者登录服务器进行调试开发。

### 1.2.2. 前端

前端采用 HTML5 与 Semantic Css 前端框架进行布局开发与润色，采用原生 Javascript + JQuery 实现网站的逻辑操作，使用 JQuery 的 Ajax 与后端的接口进行 json

数据交互。

前端提供用户注册，登录，用户注册人脸信息，预定会议室，订单查询和加入会议等功能模块。

**注册模块：**

用户输入账户用户名与密码完成账户注册。

**登录模块：**

用户输入账户用户名与密码完成账户登录进入系统导航界面。

**注册人脸信息：**

用户登录后，可以在我的档案处查看或修改本人信息，并且可以在此页面通过上传免冠的清洗照片进行人脸信息的注册。在人脸注册页面出现注册成功提示后，就可以使用预定功能，并且在门禁前端可以通过刷脸进入会议室。

**会议室预定：**

用户可以通过点击相应的时间与会议室房间号进行房间预定。

会议室预定作为系统的核心功能，我们对用户的输入做了严格的限定，用户不能选择以前的时间作为预定日期，预定时间也不能超过两个小时，预定最小时间不能小于 15 分钟，并且只能在 8:00-23:00 的区间内进行预定。

**订单查询：**

在这里用户可以查询自己的所有订单信息(如：会议日期，房间号，开始时间，结束时间，参与人员和订单号等信息)，并且可以取消已经生成订单。

**加入会议：**

用户可以通过点击加入会议来加入指定会议，当预订人同意你的加入后，你可以以你的人脸信息来开启门禁端的闸门。

这意味着预定人可以自由分配其已经预定的会议室的权限，可以把开门的权限交给与会人员。

### 1.2.3 后端

前后端采用 MVVM 框架进行架构系统。MVVM (Model - view - viewmodel) 是一种软件架构模式，有助于将图形用户界面的开发与业务逻辑或后端逻辑（数据模型）的开发分离开来，这是通过置标语言或 GUI 代码实现的。MVVM 的视图模型是一个值转换器，这意味着视图模型负责从模型中暴露（转换）数据对象，以便轻松管理和呈现对象。

后端数据库采用 Mysql 关系型数据库，开发者使用 Django 的 ORM 来对数据库进行增删改查。对前端暴露 HTTP GET, POST 和 Ajax 接口在安全的前提下与前端进行数据交互。数据库设计部分 ER 图如下图所示：

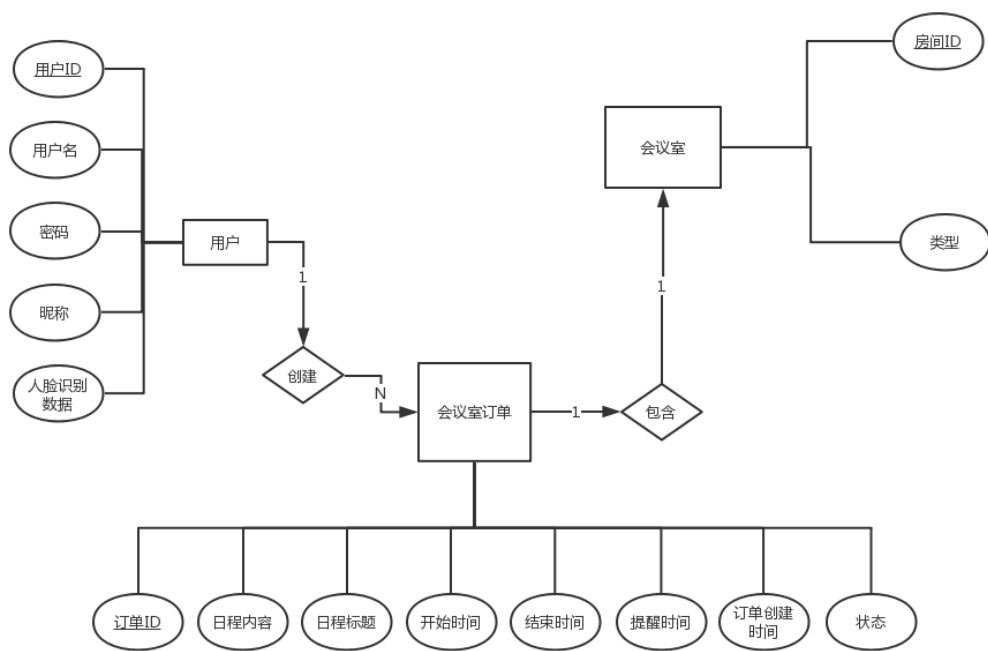


图 3 数据库设计部分 ER 图

Web 端效果图如下：

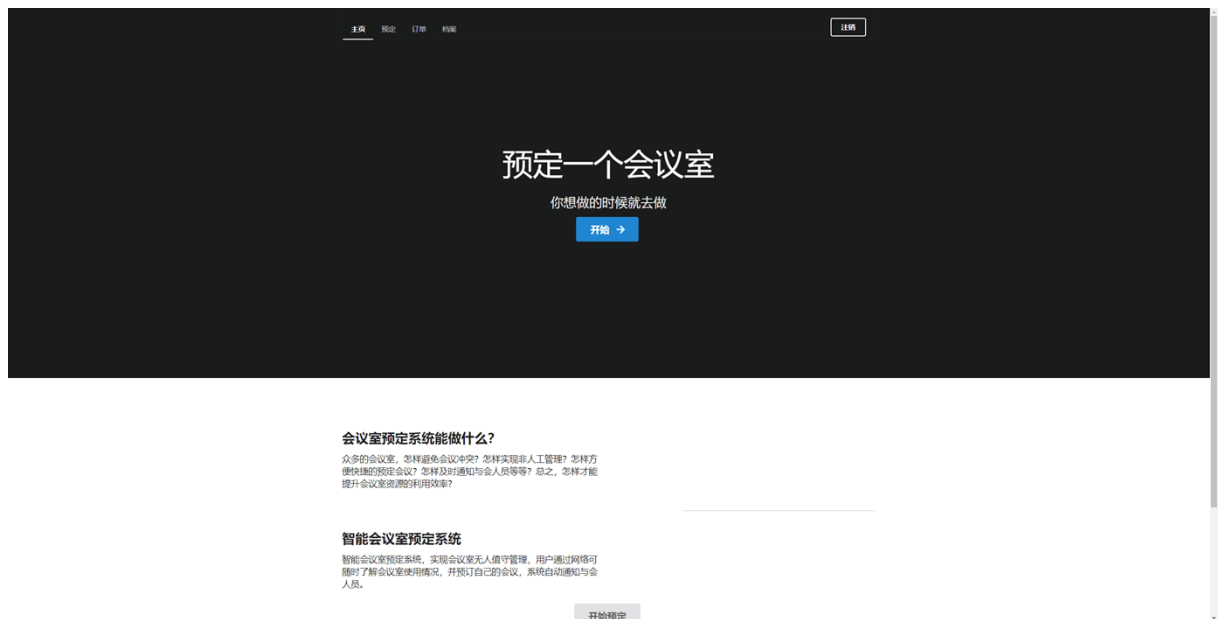


图 4 Web 端效果图 1

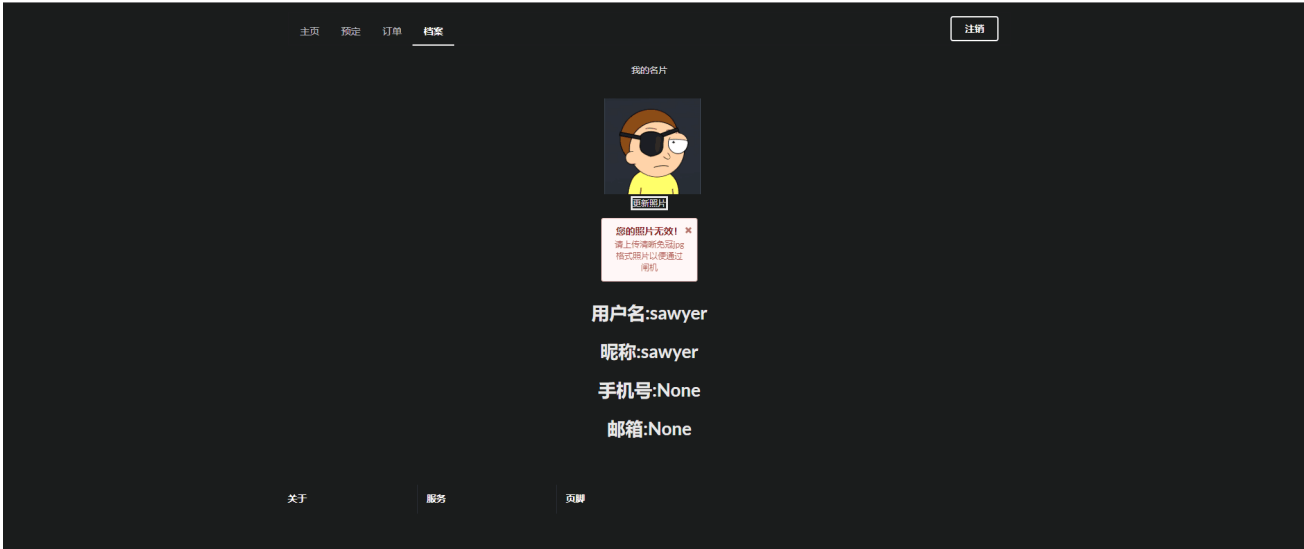


图 5 Web 端效果图 2

会议室预定															
#	8:00	9:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00	19:00	20:00	21:00	22:00
101															
102															
103															
104															
105															
106															
107															
108															
201															
202															
203															
204															
205															
206															
207															
208															
301															
302															
303															
304															
305															

图 6 Web 端效果图 3

## 2. 后台处理端详述

后台处理端的架构较为简单，设置有 Web 服务器和数据库，用来给其他三个端口提供后台服务支持、数据服务支持。这两个模块在介绍 Web 端的时候就已经说过了，这里不在赘述。

## 3. Android App 端详述

### 3.1 Android App 端综述



App 端主要完成用户线上注册、登录（通过用户上传图片/打开摄像头注册人脸信息）；会议室的线上预定/取消预定、订单查询/删除、个人档案修改；把会议的实时情况向服务器反映等工作。程序设计部分是基于 Android 平台、IDE: Anroid Studio 3.2.0, Language: Java 进行开发，人

脸识别算法则采用虹软公司的 SDK，运用 http 协议与数据库进行通信。其关键技术指标在于使用 mvp 设计模式划分 app 视图与逻辑与数据模型、异步 http 网络请求框架进行网络请求以及整合虹软公司 SDK 来对人脸图片抓取转特征码储存数据库。App 端系统架构如下：

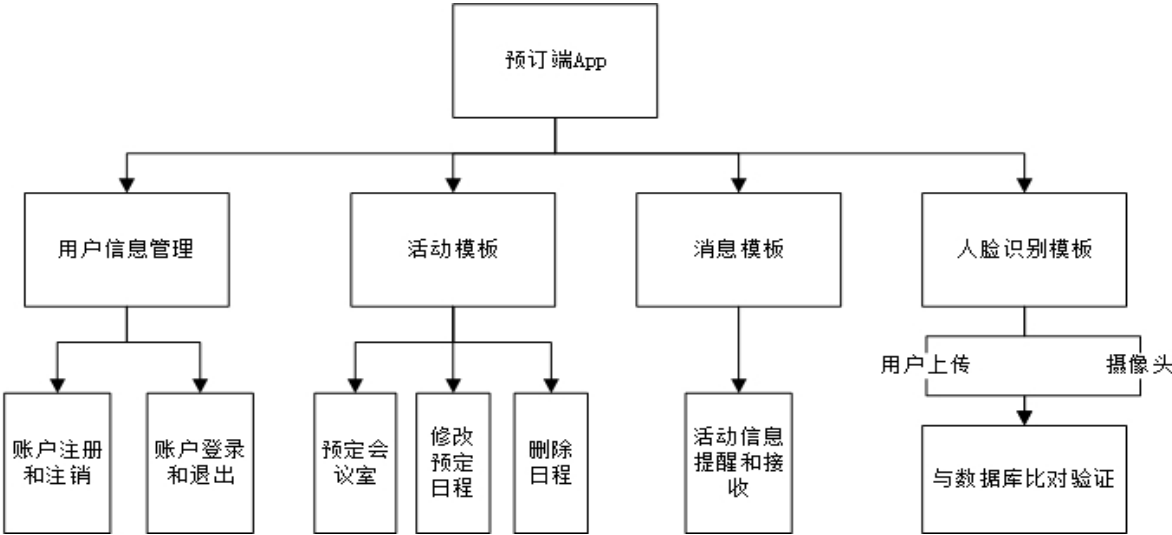


图 7 App 端系统架构

3.2 Android App 端详细模块

功能描述：App 端主要为公司会议人员而设计，作为会议室的线上预定端。主要完成会议室的预定及预定信息查询与展示、会议日程的添加与删除、通过人脸识别数据的采集和比对进行用户的注册与登录、与后台数据库实时连接进行用户个人档案的修改等工作，故 App 预定端的主要模块有以下几部分：

3.2.1 MainActivity 模块

App 登录后的主要模块，包含 HomeFragment, OrderFragment, MineFragment 三个模块

\*HomeFragment

App 默认的显示界面，用于打开会议室预约页面, 可以一键前往预订记录页面

\*OrderFragment

展示用户预订的记录一览, 可以取消预订/修改添加会议日程

\*MineFragment

用户的个人信息页面，显示用户的基本信息，如用户姓名，电话号码，性别，用户头像。同时具有退出登录、注销账户和修改个人资料功能。

### 3.2.2 LoginActivity 模块

登录注册页面, 可选择上传头像或拍摄。登录注册模块主要用于用户的首次注册与使用应用的登录, 点击登录或注册按钮时先进行网络连接检测, 需在有网络的情况下进行操作。连接数据库, 然后检测账户的有效性。默认选择记住密码, 方便下次登录。同时有修改密码提示文字, 点击后弹出修改密码的对话框。

注册

#### 1) 号码/邮箱

对数字和字符的检测、手机号码 11 位的限制、是否之前注册过的检查（要和服务器对接）；

#### 2) 密码

位数的限制, 例如最少 6 位、至少含英文字母、数字、特殊符号任两种等；

#### 3) 短信验证

服务端写好接口, 移动端通过 get 或 post 手机号码过去, 然后接口调用 API 发送/重复发送的倒计时

手机登录

第三方登录: 微信登录（使用的是微信开放平台的 SDK, 注意要先判断用户是否有安装微信）/新浪微博登陆（使用新浪开放平台的 SDK, 新浪 SDK 会自动判断用户是否有安装新浪 APP）

### 3.2.3 MeetingRoomActivity 模块

会议室预订页面, 表格形式展示所有时段所有会议室的预订状态

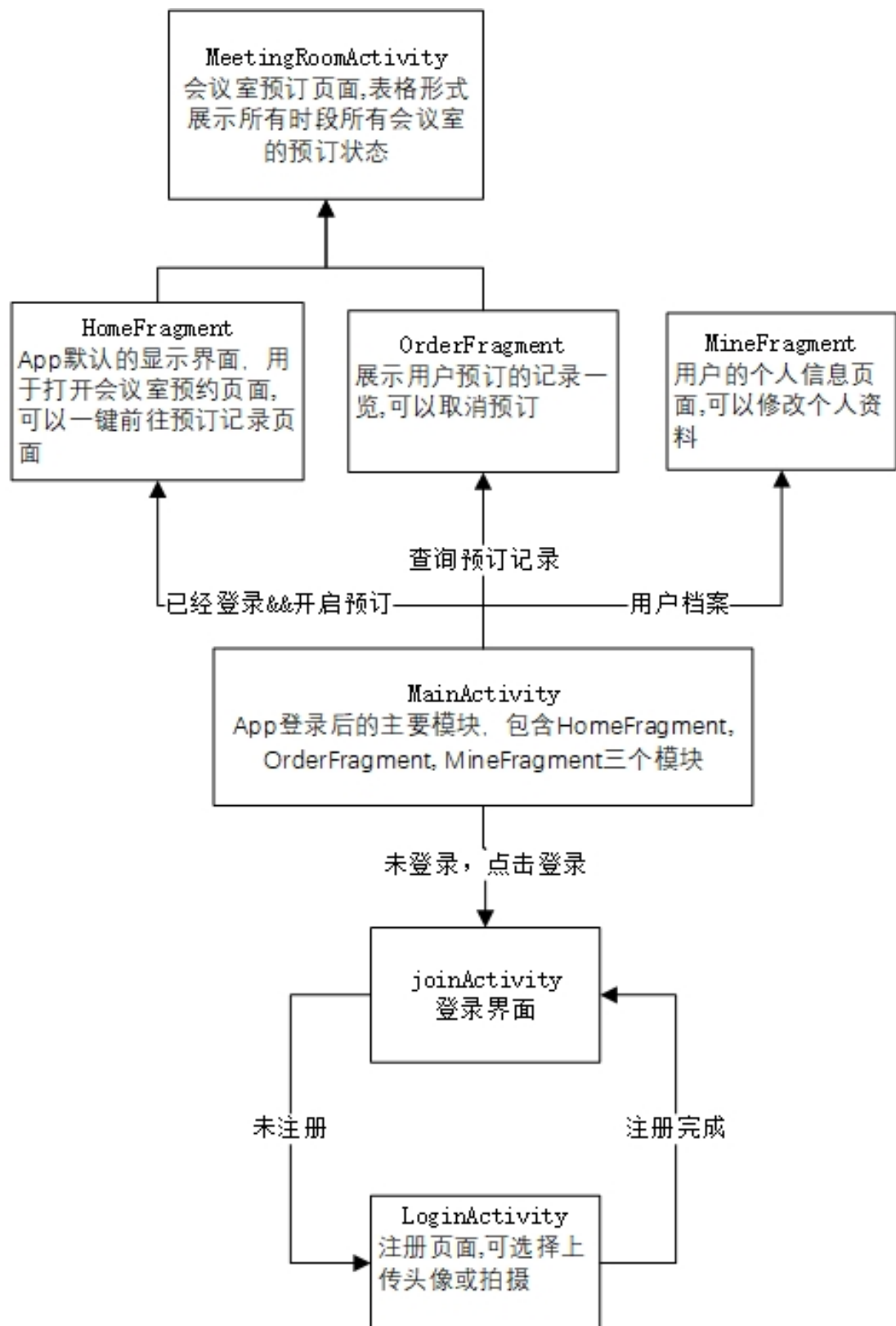


图 8 Android App 端详细模块

Android APP 端效果图如下：

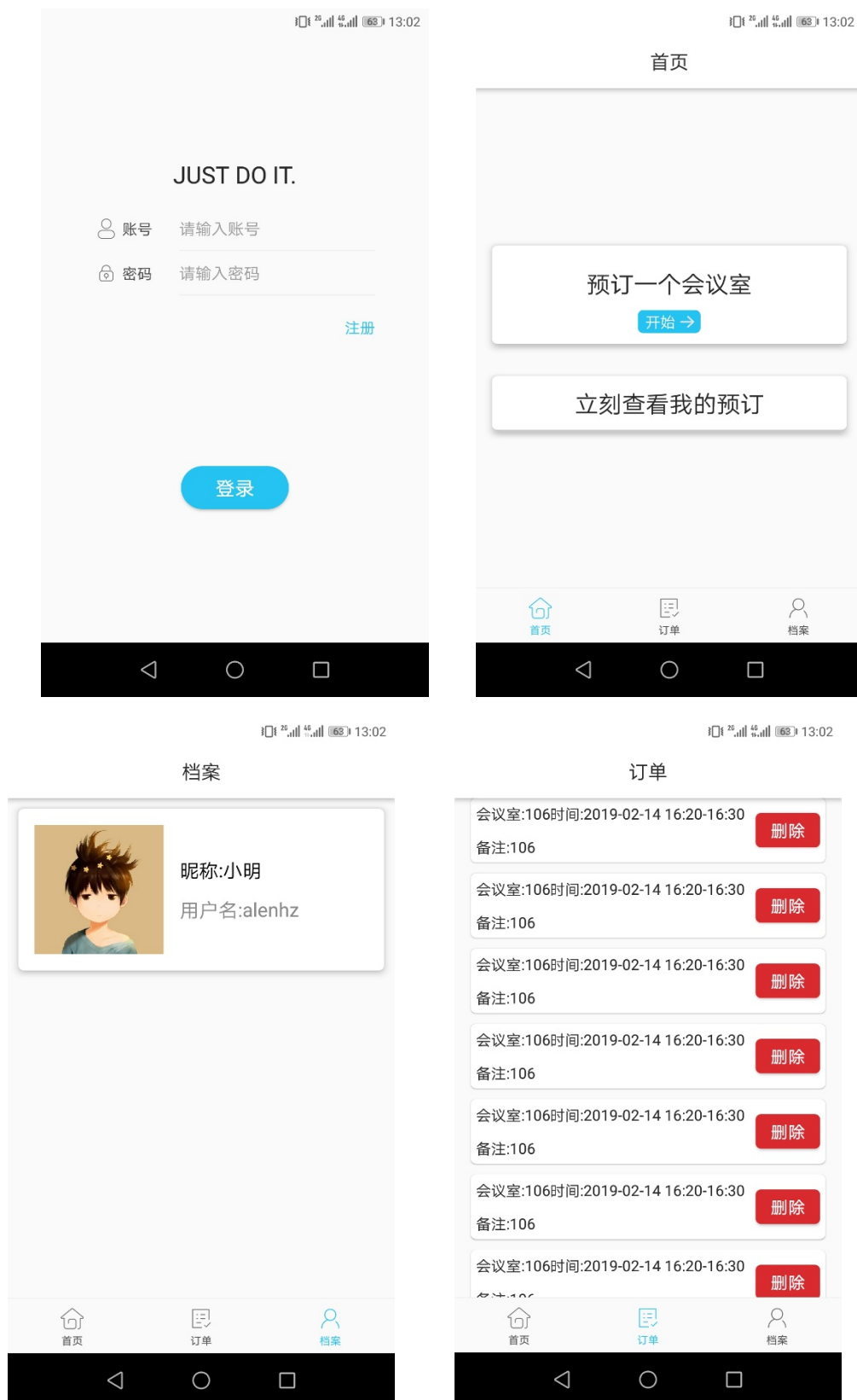


图 9 Android APP 效果图

## 4. 门禁端详述

### 4.1 门禁端综述

门禁端主要完成线下会议室的管理工作(如开门、身份验证等)和把会议的实时情况向服务器反映。硬件部分采用 NanoPi-M4 开发板, 外接电容触摸屏和 500 万像素摄像头, 搭载 Android Oreo 系统。软件部分采用 java 语言开发 Android 程序, 人脸识别算法采用虹软公司的 SDK, 运用 TCP/IP 协议与数据库进行通信。门禁端系统架构如下:

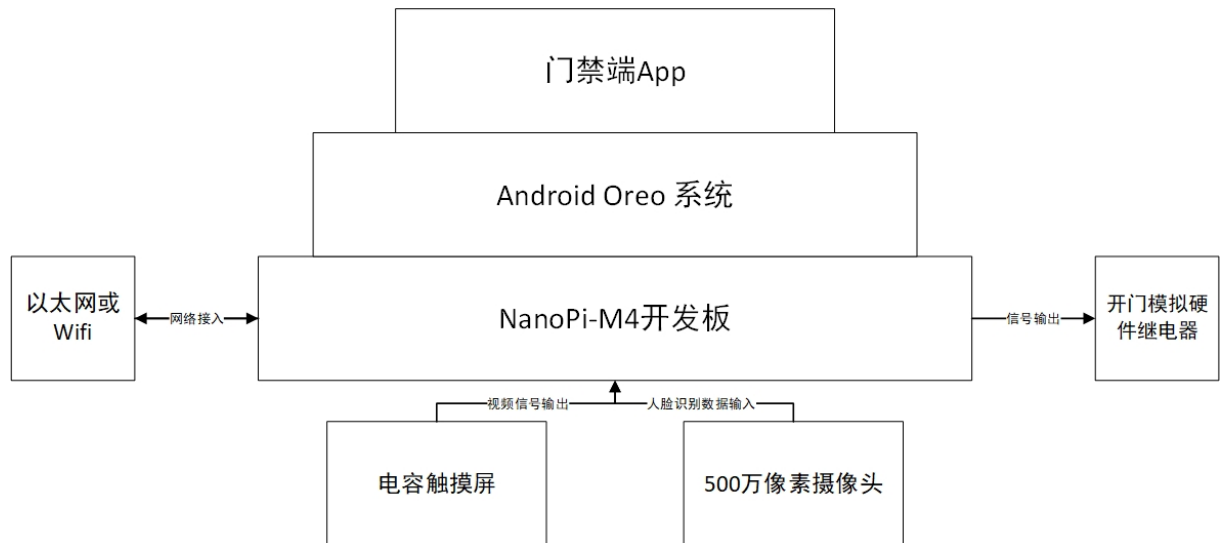


图 10 门禁端系统架构

### 4.2 门禁端详细模块

#### 4.2.1 硬件模块

硬件选用友善科技 NanoPi-M4 开发板, 外接 USB 电容触摸屏和 USB 摄像头。触摸屏的显示信号通过开发板 HDMI 接口接入, 触摸信号通过 USB 接口输出到开发板, 并且需要外接 5V 电源。摄像头用来采集人脸数据, 进而提供给软件层进行人脸比对, 因此选择了 500 万像素。开发板带有以太网和 wifi 接口, 可以通过这两个接口来与服务器通信。与其他会议室硬件例如门的通信, 采用 USB 输出的继电器硬件进行模拟。具体架构见“图 2 门禁端系统架构”。

#### 4.2.2 软件模块

软件端主要完成预订信息的展示、人脸识别数据的采集和比对、验证人员身份等工作, 主要的模块有以下几部分:

##### \*MainActivity

App 默认的显示界面, 用来显示当前会议室的预订信息。此页面可以完成用户的登录操作, 登录后点击开启按钮, 验证成功后即开启会议室。

### \*OpenActivity

此页面展示当前登录用户的预订信息，并可对预定的会议室进行一系列的操作，比如开启、延期、取消等等。

### \*SignActivity

此页面为签到页面，显示签到人数和会议开始时间。页面右侧设有人脸识别窗口，可以通过人脸验证来签到，正常情况下验证时间小于 1s，经过后台数据缓存后，还可实现离线验证。

### \*MettingActivitty

会议中默认的显示界面，展示会议的相关信息，还设有提前结束按钮，身份验证成功后即可提前结束会议。

### \*LoginActivity

登录界面，用于用户通过用户 ID 和密码登录。

### \*VerifyActivity

验证界面，通过摄像头采集用户人脸数据，提交给后台进行人脸数据比对，验证用户身份。

门禁端软件架构如下图所示：

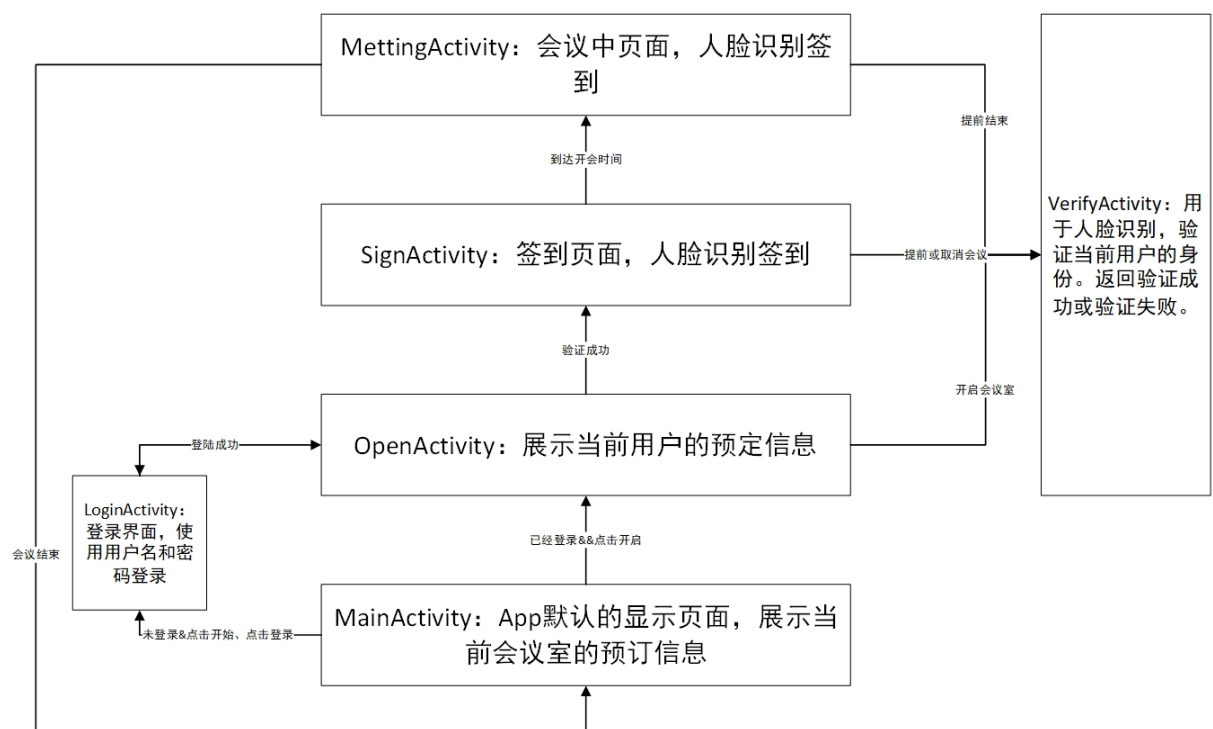


图 11 门禁端软件架构

门禁端效果图如图所示：



图 12 门禁端效果图

## 第四章 测试报告

### 一、Web 端测试报告

#### 1. 测试人员

Web 端负责人 唐嘉宗

#### 2. 测试目的

2.1 测试能否处理用户上传肖像，如以下几种情况：(1)用户上传了没有人脸的图片  
(2)用户没有上传图片。本次测试用来确保程序能正确处理这些意外情况。

2.2 测试预定功能是否合理，如用户预定往日的会议室，用户对往日的无效订单进行操作或未登录用户进行注册等操作。站在用户的角度上思考需要给用户展示那些信息，展示形式以及展示的效果。

2.3 测试加入会议与分配权限功能。

#### 3. 测试时间

2019 年 3 月 20 日

#### 4. 测试环境

4.1 软件运行环境

硬件：采用远程搭载 Ubuntu 的 Linux 服务器。

#### 4.2 软件编译测试环境

开发 IDE：本程序采用 Pycharm 作为开发、测试的 IDE，同时集成 MySQL、软 SDK 等第三方软件开发包。

运行系统：开发环境为 Linux

### 5. 测试内容

#### 5.1 个人档案页面

此界面可以让用户修改自己的信息，尤其是个人肖像。

点击更新图片，选择正常人脸图片后，点击提交。提示正在处理图片…请稍后…

随后提示图片有效！您可以顺利通过门禁闸机。

点击更新图片，选择一棵树图片，点击提交。提示正在处理图片…请稍后…

随后提示图片无效！您不能顺利通过门禁闸机。

新用户注册，进入档案页面会直接看见图片无效！您不能顺利通过门禁闸机警告消息。

#### 5.2 预定界面

此界面为登录前后都能查看的页面。

未登录用户：点击某一会议室进行预定，出现请登录提示框。

已登录用户：点击某一会议室进行预定，出现预定提示界面，输入备注后点击保存成功预定。

已登录用户：在往日的日期进行预定，出现不能预定过期的会议提示。

#### 5.3 我的会议界面

此界面为管理会议界面，主要有两个功能。一是管理自己预定的会议室，二是分配他人的权限。

经过测试，预定能正确被取消，权限能正确分配。

#### 5.4 加入会议页面

此页面用来让用户加入会议，用户根据时间来选择会议进行加入。随机挑选会议点击加入按钮，提示已经发送加入请求。登录被请求账户，发现有新的未处理请求，点击我的会议进行权限分配，勾选用户。登录请求账户，点击我的会议，可以看见我加入的会议请求已经通过。

### 6. 存在问题

#### 6.1 人脸识别模块



人脸识别模块打包好后用 Python 来调用运行速度较慢，用户在 Web 端提交肖像之后，需要 2-3 秒才能完成运行完成。

## 6.2 界面的展示问题

会议室预定情况展示界面总体上没有太大问题，可以正常地展示会议室在不同时间的预定情况，但是结束时间不能正确的展示，只能取精确到小时展示。

## 二、门禁端测试报告

### 1. 测试人员

门禁端负责人 张效凯

### 2. 测试目的

2.1 测试人脸识别模块的识别准确率和识别效率，确保人脸识别模块的稳定性。因为整个系统的身份验证机制是通过人脸识别模块进行，因此测试人脸识别模块，确保识别的快速、准确是必须的。

2.2 测试各个模块的信息展示是否合理，站在用户的角度上思考需要给用户展示那些信息，展示形式以及展示的效果。

2.3 加强用户体验，提升界面的美化水平和界面的响应速度，确保用户能够流畅舒适地使用此产品。

2.4 验证各个模块之间的交流机制和接口设计，确保各个模块之间能够流畅的转换，能够正常的传递数据，提高每个模块的错误数据的处理水平和容错能力。

### 3. 测试时间

2019 年 3 月 20 日

### 4. 测试环境

#### 4.1 软件运行环境

硬件：开发板选用友善之臂公司研制的 RK3399 NanoPi-M4 主板，外接 500w 像素摄像头用于人脸识别。视频输出到树莓派电容触摸显示屏，可以方便的进行各项测试操作。与其他端的通信通过以太网接口或 wifi 来进行。

软件：本程序运行在 Android Oreo 系统。

#### 4.2 软件编译测试环境

开发 IDE：本程序采用 Android Studio 作为开发、测试的 IDE，同时集成 mysql、软 sdk 等第三方软件开发包。

运行系统：开发环境为 win10

## 5. 测试内容

### 5.1 MainActivity 页面

此页面是登录 APP 后的默认显示页面，测试的内容主要有联网下测试、断网下测试。经测试发现联网下信息显示正常，断网后不显示信息，APP 运行正常，但没有提示信息。点击开启按钮可以跳转到下一页面，联网、不联网跳转均正常。

此页面有两个功能按钮，分别是开启下一界面的按钮和登录按钮。点击开启按钮可正常跳转到下一界面，并传递会议室、登入人员的相应信息。点击登录按钮可正常进入人脸验证登录页面，验证成功后成功返回人员身份信息到主页面。

### 5.2 OpenActivity 页面

此页面为登陆后的显示页面，展示当前用户的个人信息及预订信息。经测试预订信息显示正常，个人信息显示正常。会议信息展示的会议室状态，包括已结束、等待中、待开启三个状态，用不同的时间（开始时间前 30 分钟、距离开始时间不到 30 分钟、已到达会议时间未开启、未到结束时间、已过结束时间）测试，逻辑判断正常。

点击一条会议后的操作按钮（删除和开启），均执行相应正确的操作（删除过期信息和开启会议室）。

### 5.3 SignActivity 页面

此页面为签到页面，主要有两个功能。一是展示会议信息，二是人脸识别签到。

经测试，会议信息能够正确显示，会议倒计时显示正确，倒计时结束后也能执行开始会议的操作。下方三个功能按钮取消、延期、开始均能触发相应的操作。

人脸识别签到模块引擎初始化正常、联网情况下获取人脸信息正常、用户签到成功后给予相应的欢迎信息。

### 5.4 Meeting 页面

此页面为会议中页面，展示当前开会的信息和会议倒计时以及当前会议室的预订信息。经测试会议信息、预订信息、倒计时均能正常显示。结束会议的功能按钮也能正常执行结束会议操作。

## 6. 存在问题

### 6.1 人脸识别模块

人脸识别模块在数据库内数据少量时运行良好，识别速度快，精准度高。当存入大量数据时，检索时间变慢，人脸识别速率也会变慢。另外运行时多开程序会导致系统崩溃重启，原因可能触及了开发板的运行上限。

对于人脸识别模块在大量数据时表现出性能下降的问题，后期会对数据获取模块和人脸信息筛选模块做出优化，启动程序后请求获取数据库数据并保存到本地或内存，不再重复请求。对于识别到的人脸信息，先用识别到的年龄和性别信息对数据进行第一次筛选，再进行人脸识别比对，可以大大加快比对速率。

## 6.2 信息展示问题

信息展示总体上没有问题，可以正常的展示会议室信息、用户信息和订单信息。但部分页面的字体显示、美化还存在问题，后期继续优化。

另外每切换一个页面都会重新请求数据库信息，造成信息展示存在一定的延迟，后期也要进行修改。

## 三、Android APP 端测试报告

### 1. 测试人员

Android app 端负责人 张若兰

### 2. APP 版本类型

Android Nougat 及以上版本

### 3. 测试内容

#### 3.1 MainActivity 页面

此页面是登录 APP 后的默认显示页面，测试的内容主要有联网下测试、断网下测试。经测试发现联网下信息显示正常，断网后不显示信息，APP 运行正常，点击开启按钮可以跳转到下一页面，联网、不联网跳转均正常。

此页面有两个功能按钮，分别是开启下一界面的按钮和登录按钮。点击开启按钮可正常跳转到下一界面，并传递会议室使用的相应信息。

#### 3.2 HomeFragment

App 默认的显示界面，用于打开会议室预约页面，可以一键前往预订记录页面。点击一键前往按钮能够成功跳转之订单界面，历史订单一览无遗。

#### 3.3 OrderFragment

此界面用以展示用户预订的记录一览，已预定的会议室相同时间段不能再次预定，实现了取消预订/修改添加会议日程的功能，用户可以清晰明了的了解会议室预订情况，状态展示简洁，具有实时性。

#### 3.4 MineFragment

经测试，用户的个人信息页面，能成功显示用户的基本信息，如用户姓名，电话号码，性别，用户头像。同时能够完成退出登录、注销账户和修改个人资料功能。

#### 3.5 LoginActivity 模块

登录注册页面，可选择上传头像。登录注册模块主要用于用户的首次注册与使用应用的登录，点击登录或注册按钮时先进行网络连接检测，需在有网络的情况下进行操作。连接数据库，然后检测账户的有效性。经测试，检测到账户有效之后即能成功登录，修

改个人信息模板功能正常。

#### 4. 软件功能

注册账号、登录有效账号

修改个人信息，包括用户昵称、可上传头像、修改电子邮箱和电话号码

预定任意空闲时间会议室并备注会议信息

展示历史订单、我的日程，可添加和删除预定订单

快捷键实现一键查看预定信息

注销、退出当前账号

#### 5. 软件优势

界面简洁、扁平化、实用性强

软件使用性强，会议发起者与参会人员黏性增强，互动性增长

用户可提前了解会议室空闲时段，避免预定等待问题

企业职员对会议室预订情况可随时了解，随时预定

#### 6. 软件劣势

验证机制不完善会造成软件的使用程度弱

服务器存在不稳定性，需不定期检测

目前功能相对较少，不能完全满足用户全部业务需求，需进行深度开发

#### 7. 改进建议

增强服务器稳定性

对软件功能进行二次开发，将人脸签到功能添加入 APP 中，完善验证机制，提高签到效率，增强安全性

APP 功能更简洁明了，便于企业职员使用

## 第五章 安装及使用

作品有三个用户端，环境搭载如下：

**App 端：**

1. 安装相关工具及下载文件

Android studio

<https://developer.android.google.cn/studio/>

2. 启动 Android studio, 本文使用的是 3.2.0 版本
3. 准备需要在 Android Studio 中打开的项目。请将 zip 文件下载下来并解压缩
4. 修改项目中所需的 Gradle 软件包需求。

找到 EdgeEffectOverride-master\gradle\wrapper 下面的 gradle-wrapper.properties 文件, 使用 Notepad++ 之类的文本文件工具打开它, 请将红色部分改为你的机器上已安装的 Gradle 版本, 比如我这里是 gradle-1.12-all.zip。

5. 点击 Import Project, 定位到项目所在的目录, 找到其中的 build.gradle 文件, 然后点击 OK 按钮。
6. 导入成功之后即可编译运行。
7. 将编译好的 android studio 项目打包成 apk

(1) 首先在 Android studio 上方的工具栏找到 build, 点击查看 bulid 工具;

(2) 在 bulid 工具栏的下拉栏, 找到倒数第四行的 Build Bundle(s)/APK(s), 点击进入该设置-Build APK(s)即可导出成功;

(3) 这样, 在刚才设置的路径, 我们就可以看到已打包成功的 apk 了, 可以在手机上直接安装。

### 门禁端:

1. 下载并安装 Android Studio
2. 新建项目

①Company domain 填写 example.com

②API 水平选择 API21

③样式选择 Empty Activity

④Activity name 填写 MainActivity

3. 打开项目地址, 替换文件

用本文件夹 libs 文件夹      替换      项目目录->app->libs 文件夹

用本文件夹 src->main 文件夹      替换      项目目录->app->src->main 文件夹

下载虹软 Android Arc\_Face SDK 2.0, 将下载文件夹中的      libarcsoft\_face.so 和 libarcsoft\_face\_engine.so 两个文件      复制到      项目目录->app->src->main->jniLibs->armeabi-v7a 文件夹内

4. 打开项目, 重新 Sync Project
5. 填写启动信息

- ①打开项目 java->common-Constants
- ②填写申请到的虹软 SDK 激活码
- ③填写 web 端已搭载的服务器信息
6. 连接终端设备或开启虚拟机
7. 编译程序并执行
8. 若提示缺少程序包，请百度相应的包，并下载到本地，添加入本地环境或项目中

#### **Web 端:**

方法一：浏览器输入 47.106.81.165:8000 直接登录已经部署的站点

方法二：自行搭建环境：

```
# MeetingRoom
A meetingRoom reservation system
# 创建项目目录
mkdir Meeting
cd Meeting
# 创建一个 virtualenv 来隔离我们本地的包依赖关系
virtualenv env
source env/bin/activate # 在 Windows 下使用 `env\Scripts\activate`
# 在创建的虚拟环境中安装 Django , Django REST framework 与 Mysql ORM
# 安装 django
$python3 -m pip install django
# 安装 rest framework
$python3 -m pip install djangorestframework
# 安装 pymysql
$python3 -m pip install pymysql
# 自动初始化会议室信息存入数据库
$python3 auto_create_rooms
# 运行
$python3 manage.py runserver
# 管理数据库
$python3 manage.py dbshell
```

## 第六章 项目总结

智能化、共享化是未来经济发展的方向，会议室管理也必然会朝着这两个方向发展。人工智能应用到生活的各个方面，那些重复繁琐、毫无技术的工作必然会被智能化的机器取代。应用智能化技术改革会议室管理系统，我们不去做，也终会有人去做。我们有幸成为踏上行业朝阳路的第一批开拓者。

这个项目大概耗时 5 个月左右，这五个月来真真切切的体会到自己在成长。从一个什么也不懂的小白到开发出一套完整的系统，收获的不仅是技术上的，更是一种能力。小组成员也从之前的单打独斗到现在的合作分工，确实不容易。期间考虑了很多别的方案，也出现过很多问题，但最后在小组成员的合力下也是解决了。学会的越多越感觉自己学得太少，以后还要继续加油！