# 基于人证核验的共享酒店住宿信息管理系统——用户端APP

# 摘要

国际酒店预订客户端是无线互联网和旅游行业快速发展、相互融合的产物，随着人民生活水平的不断提高，人们对于商务出行、旅游出行等的需求也越来越强烈，酒店预订是人们计划出行所必须经历的事情之一，具有广阔的市场前景，也可以给旅游公司与酒店带来丰厚的商业机遇，设计出一款方便好用的酒店预订客户端具有重要的意义。

本人参与设计并实现了一款国际酒店预订客户端系统，其基于平台，使

用语言编写并釆用结构设计，进而提高了系统的灵活性与可扩展性，后台服务器釆用语言编写，本人主要负责客户端开发部分。经过对产品特点以及用户特征的分析，本文提炼出了系统的需求，该系统主要包括用户登登录、酒店搜索、酒店详情、订单管理、系统设置等五大功能模块。用户登录模块提供了用户登录与用户注册等功能。酒店搜索模块提供了酒店的搜索功能，用户可根据入住退房日期、目的地城市、关键字等信息对酒店信息进行检索，并可根据位置、价格等条件进行排序。酒店详情模块提供了酒店详情信息的查看，用户可对酒店进行点评、预订酒店，通过订单管理模块用户可査看已定好的酒店的订

单信息。系统设置模块实现了第三方应用推荐、意见反馈、查看软件版本信息等功能。

关键词：酒店；预订；旅游；

# 1 引言

## 研究背景及意义

在这个移动互联的时代，人们急切需求一款可以足不出户、随时随地通过自己的手机来浏览并预订快捷酒店的住宿信息管理系统。然而在当前市场上的一些应用及酒店入住流程太复杂，用户体验和隐私效果较差，为此开发一款简单、实用的基于人证核验的共享酒店住宿信息管理系统具有实际意义。

本系统基于共享经济和人工智能，利用共享经济的优势，让用户可以快速，方便入住酒店。然后结合人工智能的人脸识别算法，验证人脸和身份证信息来保证安全，可靠。最终实现方便用户的人证核验的共享酒店住宿信息管理系统。

## 本选题在国内外的研究状况及发展趋势

共享经济，一般是指以获得一定报酬为主要目的，基于陌生人且存在物品使用权暂时转移的一种新的经济模式。其本质是整合线下的闲散物品、劳动力、教育医疗资源。有的也说共享经济是人们公平享有社会资源，各自以不同的方式付出和受益，共同获得经济红利。此种共享更多的是通过互联网作为媒介来实现的。在北京、广州、杭州等多个城市，继共享单车、共享汽车之后，共享充电宝、共享篮球、共享雨伞等共享经济新形态不断涌现，并成为新一轮资本蜂拥的“风口”。仅以共享充电宝为例，短短40天时间就获得11笔融资，近35家机构介入，融资金额约12亿元人民币。

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脸部的一系列相关技术，通常也叫做人像识别、面部识别。人脸识别系统的研究始于20世纪60年代，80年代后随着计算机技术和光学成像技术的发展得到提高，而真正进入初级的应用阶段则在90年后期，并且以美国、德国和日本的技术实现为主；人脸识别系统成功的关键在于是否拥有尖端的核心算法，并使识别结果具有实用化的识别率和识别速度；“人脸识别系统”集成了人工智能、机器识别、机器学习、模型理论、专家系统、视频图像处理等多种专业技术，同时需结合中间值处理的理论与实现，是生物特征识别的最新应用，其核心技术的实现，展现了弱人工智能向强人工智能的转化。

将人脸识别和共享酒店结合起来的状况并不多见，所以这是一个新的课题。

共享酒店是传统酒店的升级版，在保留传统酒店的同时开发新大陆，同时互联网智能平台也是是大势所趋。将共享经济和人脸识别结合起来，将会擦除新时代不一样的火花。随着酒店市场竞争推动，竞争越来越激烈，只有酒店赶上新形势，只有更新改造、升级换代，才能不落伍。而共享酒店的人脸识别没有身体接触，不像指纹机那样要接触皮肤；人脸识别让人有自尊感，按手指那是强制式的，会让人感觉到是被操作的。

# 2 业务分析

# 3 总体设计

## 3.1 界面设计

界面作为一个系统的门户，是直接和用户息息相关的，所以一个APP的最重要的就是界面设计，也就是UI设计。本系统本着简介大方，实用的基本原则，APP大致分为主界面，酒店列表，酒店详细，登录界面，注册界面，实名认证界面，订单界面。其中主界面采用底部导航，分为三个模块，分别是首页订单和我的。对应界面如下图1-1所示：然后就是酒店列表界面，酒店列表界面，主要给用户展示搜索到的酒店列表，然后用户可以在此界面进行搜索和排序，界面如图1-2所示

### E:\QQDownloads\1308341987\FileRecv\MobileFile\IMG_20180417_171040.pngE:\QQDownloads\1308341987\FileRecv\MobileFile\IMG_20180417_165408.png3.1.1 主界面

图1-1 主界面 图1-2 酒店列表界面

### E:\QQDownloads\1308341987\FileRecv\MobileFile\Screenshot_2018-04-25-22-39-06.pngE:\QQDownloads\1308341987\FileRecv\MobileFile\Screenshot_2018-04-25-22-39-25.png3.1.2 登录注册界面

### E:\QQDownloads\1308341987\FileRecv\MobileFile\Screenshot_2018-04-25-22-40-02.pngE:\QQDownloads\1308341987\FileRecv\MobileFile\Screenshot_2018-04-25-22-39-47.png3.1.3 实名认证界面

## 3.2 业务设计

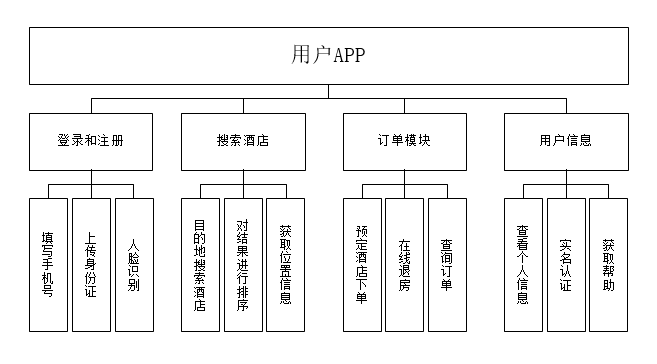
通过对业务需求的详细分析，对系统的业务进行分析，主要分登录注册模块，这一模块主要进行用户信息的身份验证，包括实名认证和人脸识别，其次就是酒店搜索模块，这一模块用户可以通过目的地，关键字，酒店名称进行搜索，然后可以用户可以对搜索结构进行排序。接着就是订单模块，用户选择酒店房间进行下单，然后查看订单信息等。最后就是用户信息模块，在此模块用户可以查看自己的信息，获取帮助等。没有进行实名认证的用户可以在此进行实名认证。进行退出登录等操作。模块分析图如下图1-3所示

图1-3 系统业务模块图

# 4 系统实现

## 4.1 登录注册

用户登录和注册是一个系统必不可少的一部分，用户只有进行登录之后才可以下单预订酒店。用户填写用户名和密码。如果登录成功服务器就会返回一个token，APP对此token保存到本地，以后检验身份下单等操作就可以使用该token进行操作，避免了频繁传输用户名密码带来的安全问题，减少了频繁查询数据库造成的压力。对于token的操作，为了减少每次请求都要都要手动带上token，所以APP实现了一个全局的网络请求拦截器，如果返回的信息带有token，就把该token保存到SharePreference。然后每次请求的时候，都判断SharePreference中的token是否为空，如果不为空就在该请求加上token。

clientBuilder.addInterceptor(new Interceptor() {

@Override

public Response intercept(Chain chain) throws IOException {

Request request = chain.request();

SharedPreferences sharedPreferences = context.getSharedPreferences("config",Context.MODE\_PRIVATE);

Response response = chain.proceed(request);

if (response.header("Authorization")!=null){

SharedPreferences.Editor editor = context.getSharedPreferences("config",Context.MODE\_PRIVATE).edit();

editor.putString("token",response.header("Authorization"));

editor.commit();

}

return response;

}

});

clientBuilder.addInterceptor(new Interceptor() {

@Override

public Response intercept(Chain chain) throws IOException {

Request request = chain.request();

SharedPreferences sharedPreferences = context.getSharedPreferences("config",Context.MODE\_PRIVATE);

String token = sharedPreferences.getString("token", "");

if (!token.isEmpty()){

Request authorization = request.newBuilder().addHeader("Authorization", token).build();

return chain.proceed(authorization);

}

return chain.proceed(request);

}

});

## 4.2 用户注册

用户首先要填写手机号，判断手机号是否正确和是否注册过，然后给改手机号发送验证码（该功能暂未实现），如果用户输入验证码正确就可以进行下一步，填写用户名和密码。在密码界面，要验证密码的复杂性和两次输入密码一直，如果都通过就提交用户信息到服务器。注册成功就返回token，相当于用户直接登录，注册失败提示对应的信息。注册成功之后就可以进行实名认证，首先用户拍摄身份证正面照片和反面照片，拍摄之后，对身份证正面照片进行ORC识别，返回用户身份证信息，验证身份证信息的完整性，成功的话进行下一步，人脸识别，打开用户手机前摄像头，获取上一步的身份证的人脸信息，然后获取摄像头中的人脸信息，对比两个信息的完整性，如果正确，就给人脸信息身份证信息上传到服务器，完成实名认证。

## 4.4 搜索酒店

用户进入首页后填写目的地，入住时间，搜索关键字等条件搜索酒店，Button触发点击事件后将这些条件封装成搜索对象通过Bundle对象传递给下一个Activity，如果输入的目的地与高德SDK获取到的用户当前位置信息一致，将当前位置的经纬度发送给服务器，如果用户没有输入目的地，目的地默认为当前位置，入住时间默认为当前时间，离店时间推后一天，搜索关键字可以为空。

HotelListActivity 在初始化方法中接收到搜索对象，将该对象传递给P层方法，P层方法调用M层方法将该对象发送到服务器，P层将服务器返回的查询结果解析成List<Hotel>对象，并将数据传递给ReclyclerView的适配器中初始化。

## 4.5 对酒店列表进行排序

列表页顶部搜索框使用谷歌原生的SearchView控件，设置监听器，重写提交方法，在提交方法里将搜索框的文字封装进搜索对象中，调用P层方法请求服务器并返回结果。

筛选下拉控件使用数组数据填充，在点击事件中通过Item的位置坐标查询筛选条件，并封装到搜索对象中，调用P层方法请求服务器并返回结果。

final ListView zhpxView = new ListView(this);

zhpxAdapter = new DropDownAdapter(this, Arrays.asList(zhpx));

zhpxView.setDividerHeight(0);

zhpxView.setAdapter(zhpxAdapter);

final ListView jgpxView = new ListView(this);

jgpxView.setDividerHeight(0);

jgpxAdapter = new DropDownAdapter(this, Arrays.asList(jgpx));

jgpxView.setAdapter(jgpxAdapter);

final ListView wzpxView = new ListView(this);

wzpxView.setDividerHeight(0);

wzpxAdapter = new DropDownAdapter(this, Arrays.asList(wzpx));

wzpxView.setAdapter(wzpxAdapter);

popupViews.add(zhpxView);

popupViews.add(jgpxView);

popupViews.add(wzpxView);

zhpxView.setOnItemClickListener(new AdapterView.OnItemClickListener() {

@Override

public void onItemClick(AdapterView<?> parent, View view, int position, long id) {

zhpxAdapter.setCheckItem(position);

mDropDownMenu.setTabText(position == 0 ? headers[0] : zhpx[position]);

mDropDownMenu.closeMenu();

key.setSortKey(szhpx[position]);

key.setSort("DESC");

mPresenter.getHotelData(key);

}

});

jgpxView.setOnItemClickListener(new AdapterView.OnItemClickListener() {

@Override

public void onItemClick(AdapterView<?> parent, View view, int position, long id) {

jgpxAdapter.setCheckItem(position);

mDropDownMenu.setTabText(position == 0 ? headers[1] : jgpx[position]);

mDropDownMenu.closeMenu();

key.setSortKey("MINPRICE");

key.setSort(sjgpx[position]);

mPresenter.getHotelData(key);

}

});

wzpxView.setOnItemClickListener(new AdapterView.OnItemClickListener() {

@Override

public void onItemClick(AdapterView<?> parent, View view, int position, long id) {

wzpxAdapter.setCheckItem(position);

mDropDownMenu.setTabText(position == 0 ? headers[2] : wzpx[position]);

mDropDownMenu.closeMenu();

if(key.getCoordinate()!=null){

key.getCoordinate().setRadius(swzpx[position]);

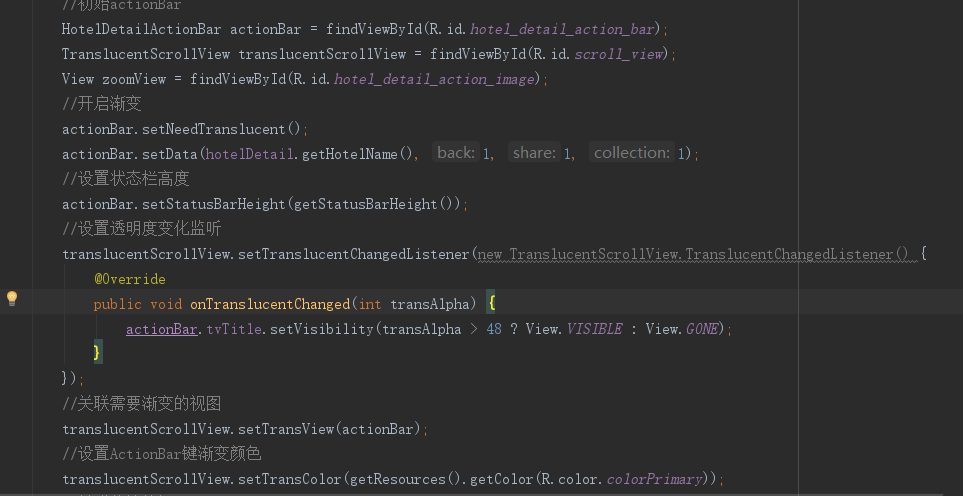
mPresenter.getHotelData(key);

}

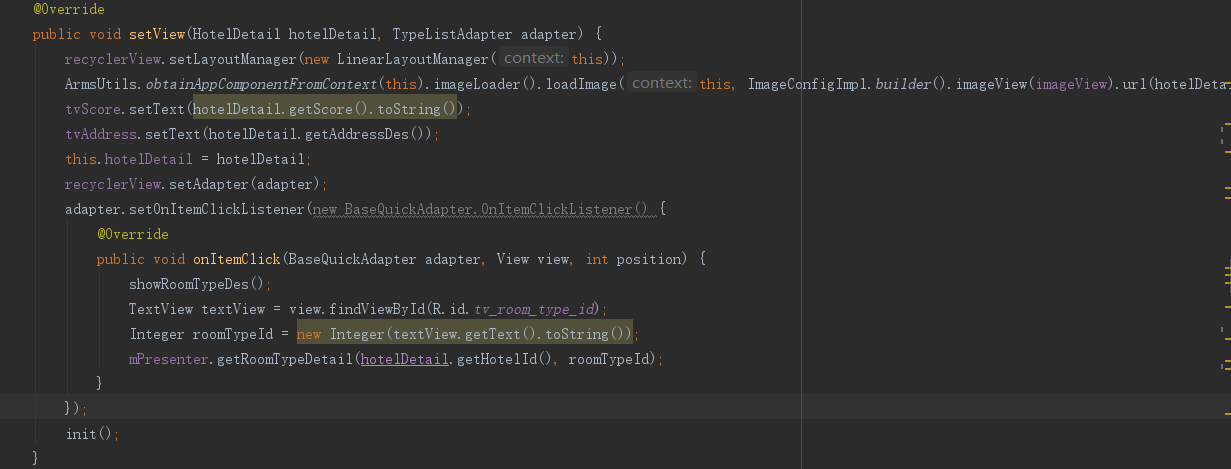
}

});

## 4.6 查看酒店详细信息

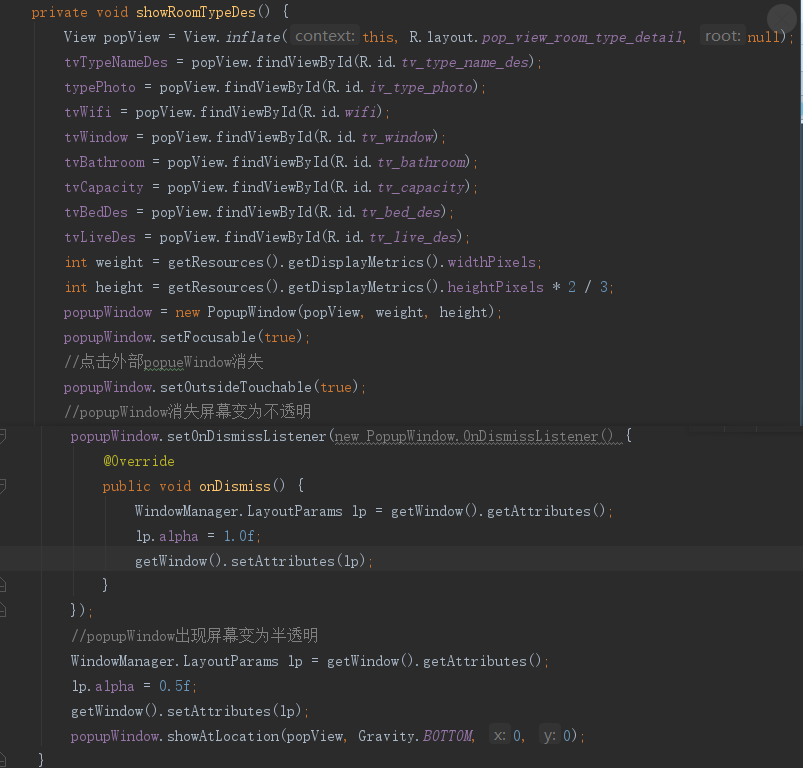
用户通过点击酒店列表，进入酒店详情页面。该页面主要用来展示该酒店的详细信息和已有的房间类型列表。首先设计了酒店详细界面，做了一个滑动特效，就是首先实现酒店图片沉浸到状态栏。然后用户向上滑动改变状态颜色和显示ActionBar和酒店名字。（具体效果参看视频该特效是自定义一个继承ScrollVIew的布局，监听其中的滑动事件，然后来改变状态栏和ActionBar的状态。接着获取酒店ID，向服务器发送请求，把收到的数据显示在界面。

## 4.7 获取酒店房型列表

用户进入详细界面之后，向服务器发送请求获取该酒店的房型列表。返回的数据用RecyclerView显示（相当于Listview）。

## 4.8 获取酒店房型详细信息

当用户点击房型列表的时候，出现一个弹出框（PopWindow）来显示房型的详细信息。首先让PopWindow显示出来，然后想服务获取数据，最后回显到PopWindow上。



## 4.9 在线预订

用户查找到指定酒店之后，选择对应的房型信息，点击预订按钮，会弹出该房型对应的可用房间号，用户选择房间号如下图2-2所示，就进入订单界面如下图2-4所示（如果没有登录，是进不去的会被拦截到登录界面）让用户确认订单信息，如果信息无误的话，就可以点击确认订单，后台就会生成对应的订单信息。

图2-2 选择房间界面 图2-3 确认订单界面

## 4.10 查询订单

用户点击订单会向服务器发送消息，查询改用户的信息，最终显示到界面。如下图2-4所示：



## 4.11 实名认证

实名认证是一个非常重要的阶段，在本系统中实名认证分为两个部分，第一部分就是拍摄身份证照片信息，会对拍摄的照片进行OCR识别，识别其中的姓名的身份证号，然后对图片进行人脸检测，最后把检测和识别到的信息保存下来。把这些数据传到下一个界面，在第二个界面中使用SurfaceView控件显示相机预览画面，在人脸识别页面实现SurfaceHolderde的回调方法，并在SurfaceView的生命周期中绑定人脸识别引擎，当SurfaceView创建时初始化引擎，当View销毁时销毁人脸识别引擎。在此页面实现PreviewCallback回调接口，在回调方法里实时检测预览帧中的人脸数据，如果检测到，立即停止预览并保存人脸数据。然后和上个界面传输过来的界面进行比对。如果对比成功就把数据传输到后台进行保存，如果不成功就提示用户重新认证。