基于AR技术的信息投放发布平台

软件架构文档

版本 <1.0>

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| <30/6/2017> | <1.0> | <创建软件架构文档> | <吴新月、杨睿恒、李蓉、陈雪菲> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

1. 简介 4

1.1 目的 4

1.2 参考资料 4

2. 用例视图 4

3. 逻辑视图 5

3.1 概述 5

3.2 在构架方面具有重要意义的设计包 5

4. 进程视图 6

5. 部署视图 6

6. 实现视图 7

软件架构文档 （简化版）

# 简介

## 目的

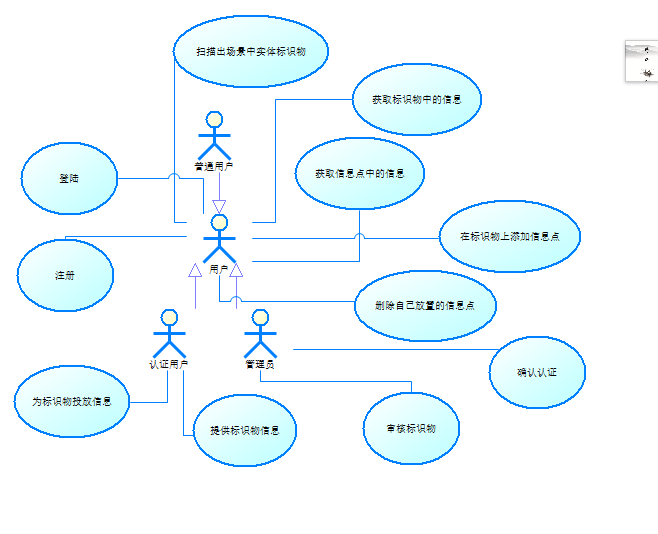
本文档将从构架方面对系统进行综合概述，其中会使用多种不同的构架视图来描述系统的各个方面。它用于记录并表述已对系统的构架方面作出的重要决策。

本文档包含用例视图、逻辑视图、进程视图、部署视图、实现视图。主要浏览此文档的人员为开发人员，此文档帮助他们在开发过程中做出架构方面的决策。

## 参考资料

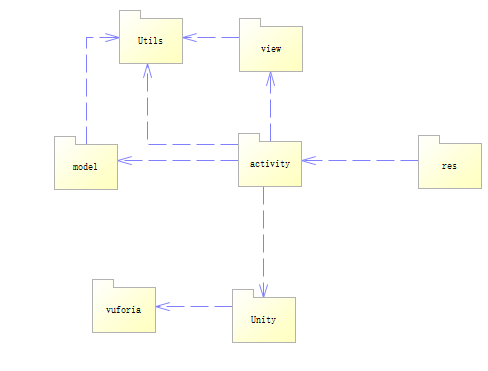
无。

# 用例视图

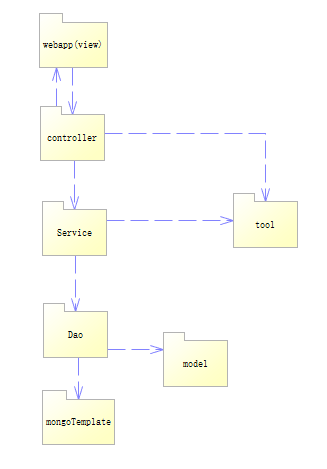


执行者分为3种，普通用户、认证用户以及管理员。他们都继承自用户。

# 逻辑视图



安卓端应用程序包图



服务器端系统包图

## 概述

安卓端系统分为七个部分，从右往左依次是：

res，即实现用户界面的layout,drawable,等文件包；

activities，包括实现View层的用于页面跳转，业务分发, 业务逻辑实现，视图展示的activity类；

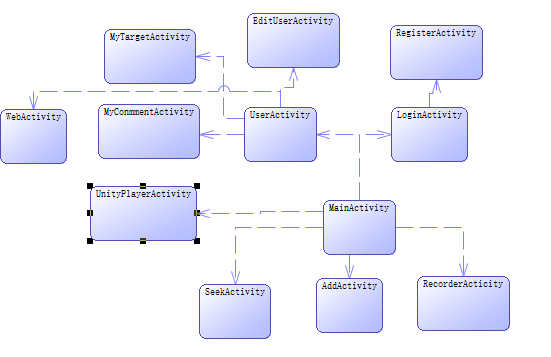
view用于辅助实现页面展示的类；

unity，包括在unity上编写的事件响应脚本和图像渲染；

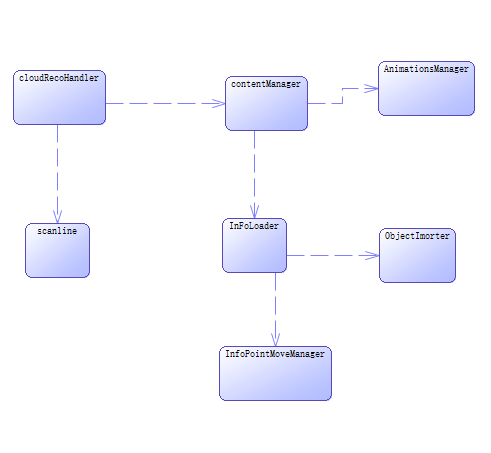
model，用于操作数据层，包含使用Hibernate O/R映射机制实现数据持久性操作的类；Utils包，包含若干贯穿系统各个部分的公共服务类。

## 在构架方面具有重要意义的设计包

Activivy:

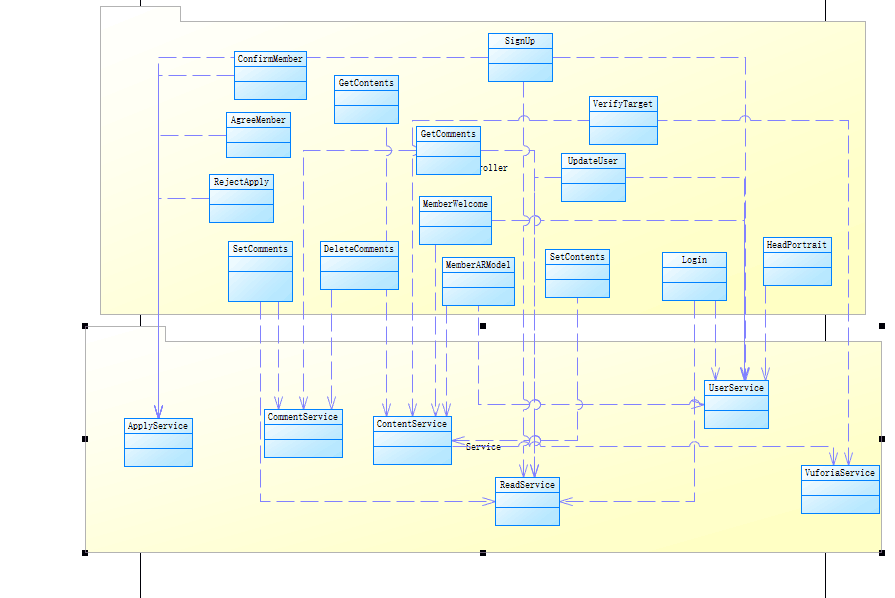
名称:activity包  
简要说明：该包是一个负责与用户交互和通过调用工具包实现业务逻辑的包  
图：  
MainActivity类：  
展示app启动页面和主页面，监听交互事件，通过跳转分发相应的业务。  
LoginActivity类：  
接收用户输入并向服务器发送请求，处理登陆和注册业务。  
RegisterActivity类：  
用来处理login后续的注册业务。  
UserActivity类：  
展示用户个人信息和用户相关功能入口。  
EditUserActivity类：  
用来处理编辑用户个人信息业务。  
MyCommentActivity类：  
用来展示用户的全部历史评论。  
MyTargerActivity类  
用来展示用户创建的信息源。  
RecorderActivity类：  
用来处理用户上传录音评论的业务。  
AddActivity类：  
处理添加文字评论的业务。  
SeekActivivy类:  
处理查看全部评论业务。  
UnityPlayerActivity类：  
处理和unity的交互调用。  
WebActivity类：  
处理从认证用户入口访问服务器网页端的业务。

Unity:

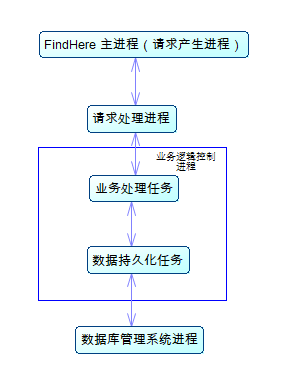
名称：UnityPlayer包  
简要说明：该包负责图像的扫描识别和信息的渲染、展示、交互。  
图：  
CloudRecoHandler类：  
继承于Vuforia的ICloudRecoEventHandler接口，用来进行对图像的识别并根据识别的结果进行不同的处理。  
ScanLine类：  
控制在扫描状态时上下移动的扫描线的显示。ClouRecoHandler类根据识别结果调用ScanLine中的接口，控制扫描线的出现与消失。  
ContentManager类：  
监听各种交互事件，并做出相应的处理，控制各种GameObject的出现、消失以及部分动画效果。CloudRecoHandler类识别成功后调用ContentManager展示识别后显示的GameObject并开始监听各类点击、拖动事件。  
AnimationsManager类：  
提供文字及图片信息从识别图前变换到屏幕前（或相反）的动画效果，变换由点击事件或跟踪识别图事件触发。  
InfoLoader类：  
包含与安卓交互的接口，负责各类信息的加载。  
InfoPointMoveManager类：  
控制信息点加载时的动画效果，加载成功时InfoLoader调用。  
ObjectImporter包：  
模型加载包，大部分复用的他人代码，负责模型的渲染。InfoLoader在从网络获取模型信息后调用其接口生成渲染模型。

Model包:  
简要说明储存实体化信息和后端进行信息交互的实体类  
Comment类  
包含了录音 图片 文字三种格式的评论实体类。  
User类  
储存用户个人信息的用户实体类。  
target类  
储存信息源ID等信息的实体类。  
  
Utils包  
简要说明：处理特定业务的工具类  
ExAudioRecorder类  
用来录音和处理音频的工具类  
RecordButton类  
用来监听录音按钮和呈现录音时按钮的交互效果的类  
Connect类  
用来和服务器进行连接和数据传输的类  
imageUtil类  
进行图片压缩等和图片相关的处理的类

服务器端controller和service包：



# 进程视图



上图中总共有4个进程，

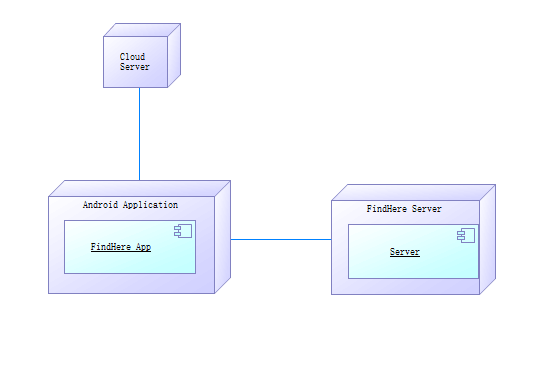
其中，请求产生进程中运行的是用户访问系统时所使用的客户端，它与服务器运行于不同的计算机上，因此需要运行于单独的进程中。

请求处理进程中运行的是各种任务，它用于接收来自客户端的请求，然后转发给业务逻辑控制进程去执行业务逻辑，在接收到业务逻辑控制进程的处理结果后，将其组装成HTTP相应发送给客户端。

业务逻辑控制进程执行的是业务逻辑，运行的任务包含两部分：执行业务逻辑的业务处理任务和执行数据库持久性操作的数据持久化任务。

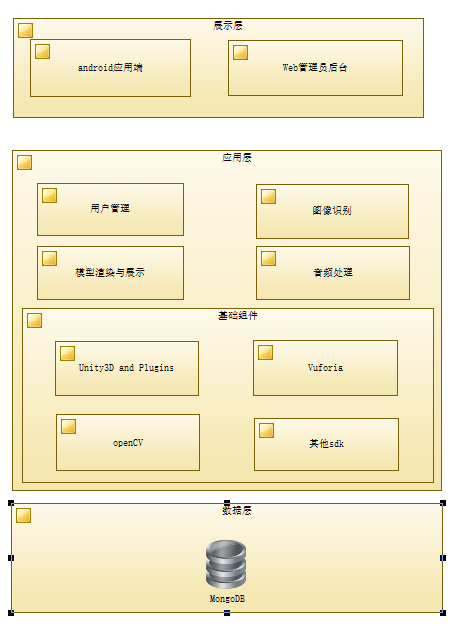
在该软件中，Spring类用来实现业务逻辑，而持久性操作是通过Hibernate类执行的，Spring类和Hibernate类可以运行于不同的进程中，但是考虑到性能，Spring类会通过Hibernate类的本地接口与之交互，而这样做的前提是他们必须在同一个进程中，因此，这两部分任务被设计为运行于同一个进程中。最后是运行数据库管理系统的DBMS Process，负责管理存储在数据库中的数据。

# 部署视图



客户端通过WAN连接Server，其中，关于图像识别的部分交与vuforia的云识别处理，其他数据访问、业务逻辑由自建服务器处理。

# 实现视图



展示层包含Android客户端以及web端的管理员后台。采用MVC结构，Model（模型）封装了应用的一系列数据并定义处理这些数据的逻辑和计算规则；View(视图是用户可以看到的对象，能够响应用户的操作；Controller（控制器）负责处理输入用户交互。

应用层包含用户管理、图像识别、模型展示、音频处理，以及项目依赖的基础组件。

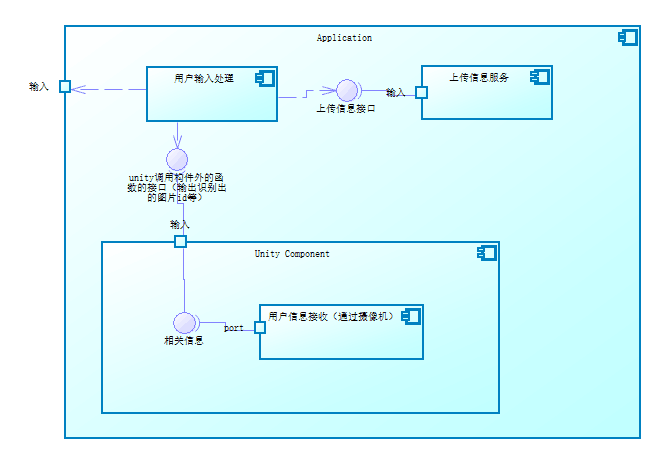
用户管理负责用户的登录状态管理、身份管理；

图像识别负责对获取图像的识别，这一部分采用是Vuforia的云服务以及计算机视觉库OpenCV；

音频模块负责对音频的处理及管理。

基础组件层包含Unity3D的assets、Vuforia 、openCV以及其他SDK。

数据层包含数据访问和数据存储。由于mongo能把文件分成小块储存，在分布式系统中有更大的优势，同时数据格式要求相对较低，数据存储选取了MangoDB数据库，存储在磁盘上。



以上是android应用端的构件。用户输入处理负责响应用户的交互请求，上传信息服务负责处理用户上传的评论等信息，Unity Component负责处理AR的具体实现，比如图像的识别以及模型的投放渲染。Unity player通过摄像头监听获取的图像并比对识别，将结果提供给用户，实现用户信息接收。