**概要设计**

# —— 基于OPENPAI平台的云智能POSSTER拍照系统

目录

[概要设计 1](#_Toc13749889)

[—— 基于OPENPAI平台的云智能POSSTER拍照系统 1](#_Toc13749890)

[0. 修订历史 2](#_Toc13749891)

[1. 概述 3](#_Toc13749892)

[2. 术语表 3](#_Toc13749893)

[3 设计概述 4](#_Toc13749894)

[3.1 系统结构设计 4](#_Toc13749895)

[3.1.1 系统逻辑架构 4](#_Toc13749896)

[3.1.2 系统物理架构 5](#_Toc13749897)

[3.1.3 系统数据模型 5](#_Toc13749898)

[3.1.4数据库实体表 5](#_Toc13749899)

[3.1.5 数据字典 6](#_Toc13749900)

[3.2 系统接口设计 8](#_Toc13749901)

[3.3 约束和假定 12](#_Toc13749902)

[3.3.1 简介 12](#_Toc13749903)

[3.3.2 采用的语言 12](#_Toc13749904)

[3.3.3 开发工具与平台 12](#_Toc13749905)

[3.3.4 数据库 12](#_Toc13749906)

[3.4 非功能性设计 12](#_Toc13749907)

[3.4.1 性能设计 12](#_Toc13749908)

[3.4.2 维护性设计 12](#_Toc13749909)

[3.4.3 易用性设计 12](#_Toc13749910)

[3.4.4 兼容性设计 13](#_Toc13749911)

[3.4.5 可拓展性设计 13](#_Toc13749912)

[3.4.6 可靠性设计 13](#_Toc13749913)

[3.4.7 灾备设计 13](#_Toc13749914)

[3.4.8 安全性设计 13](#_Toc13749915)

[4. 参考资料 13](#_Toc13749916)

# 0. 修订历史

|  |  |  |  |
| --- | --- | --- | --- |
| 版 本 | 日 期 | 描 述 | 作 者 |
| v1.0 | **2019.6.6** | 完成概述部分 | **游增** |
| V1.1 | **2019.6.9** | 完善系统结构设计 | **林会东** |
| v1.2 | **2019.6.9** | 完善系统数据模型 | **游增** |
| V1.4 | **2019.6.11** | 完善系统接口设计 | **陈梓轩** |
| v1.5 | **2019.6.12** | 完善非功能设计 | **林会东** |
| v1.6 | **2019.6.14** | 根据各部分，补充术语表 | **邱润韬** |
| v2.0 | **2019.6.17** | 小组讨论，对概要设计进行整体修订 | **全体成员** |
| v2.1 | **2019.6.17** | 添加目录，调整排版样式 | **邱润韬** |
| V2.2 | **2019.6.20** | 完善系统逻辑架构图和系统物理架构图 | **王永乐** |

# 概述

《基于OPENPAI平台的云智能POSSTER拍照系统》（下面简称poster）是一个针对特定场景的拍照软件，主要有两个功能。一个是拍摄照片，并优化照片光照，第二个是将照片中的ppt矫正。

软件主要针对场景是光照不均匀的演讲、讲座、讲堂、教室等，这些场景下，普遍存在着人像和ppt区域光照不一致，一般表现为，ppt比人像区域光照要充足一些，因为ppt要么是自发光的屏幕，要么是反射光的投影仪，而为了下面观众在演讲进行时能够看清ppt，演讲台上的光源会调低，导致人像部分光照远低于ppt光照，因此，现有的拍照软件下，如果你调整相机使ppt区域清晰且曝光适度的话，那么人像部分会因为过暗而丢失信息，看不清楚；反过来如果你听过调整参数是人像显得清晰的话，那么ppt区域又会因为过曝而泛白，丢失信息，ppt看不清楚。

因此我们的软件旨在针对这种场景下，将ppt和人像都要拍的清晰，且曝光适度而不突兀，另外我们希望能够将ppt单独提取出来，对它进行矫正，而不让它因为拍摄角度的问题显得不方正，是斜的或者歪的。并且能够让用户单独得到ppt。

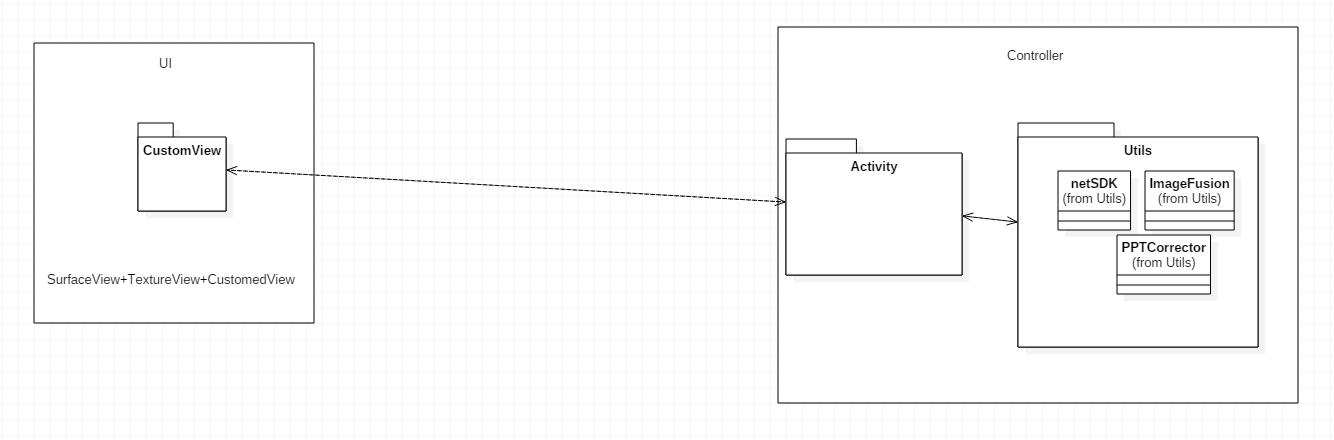
# 术语表

|  |  |  |
| --- | --- | --- |
| 术 语 | 定义和信息 | 别 名 |
| PPT | 演讲用的演示文稿 | slides |
| GUI | 即图形用户[界面](https://baike.baidu.com/item/%E7%95%8C%E9%9D%A2" \t "https://baike.baidu.com/item/GUI/_blank)，是指采用图形方式显示的计算机操作用户[界面](https://baike.baidu.com/item/%E7%95%8C%E9%9D%A2) | 图形界面 |
| 架构 | 有关软件整体结构与组件的抽象描述 | 构架 |
| Mobile net | 指将训练好的神经网络移植到手机等移动平台后的网络结构 | / |
| 接口 | 泛指实体把自己提供给外界的一种[抽象化](https://baike.baidu.com/item/%E6%8A%BD%E8%B1%A1%E5%8C%96/10844295)物（可以为另一实体），用以由内部操作分离出外部沟通方法，使其能被内部修改而不影响外界其他实体与其交互的方式 | / |

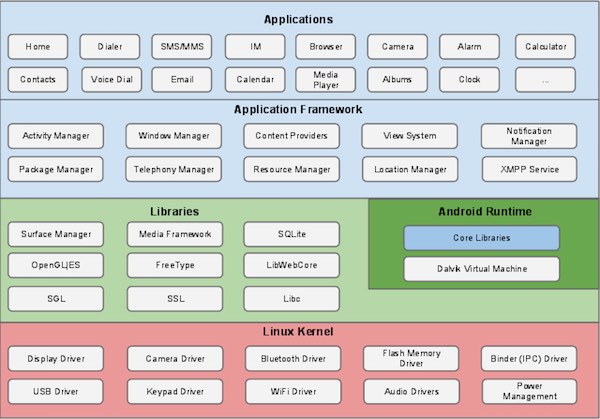
# 3 设计概述

## 3.1 系统结构设计

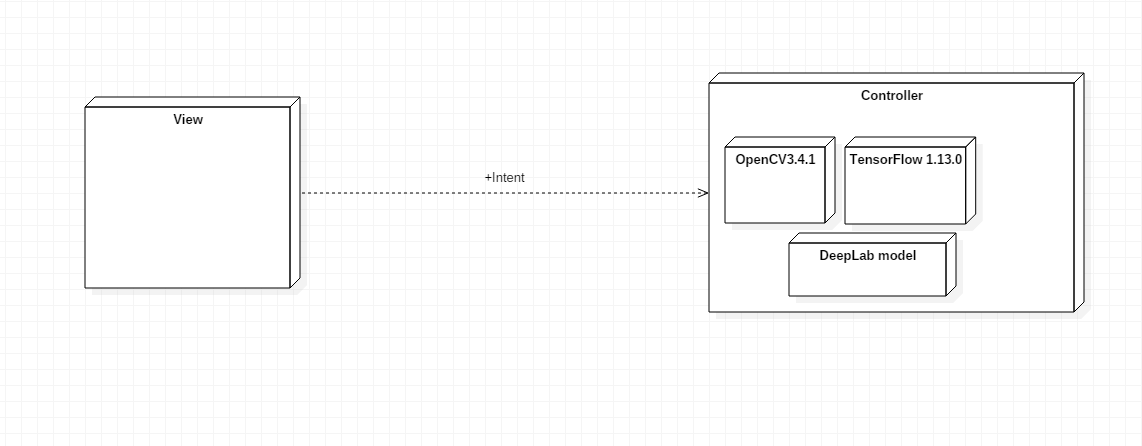
### 3.1.1 系统逻辑架构



Android架构解释：



### 系统物理架构



## 系统接口设计

## UI

## API

## 3.3 约束和假定

### 3.3.1 简介

约束和假定是系统设计中最主要的约束，是客户明确要求的。此部分文档依据需求说明书列出对本软件开发工作的假定和约束。

### 3.3.2 采用的语言

Java

Python

### 3.3.3 开发工具与平台

Android studio

Pycharm

## 3.4 非功能性设计

### 3.4.1 性能设计

* + - 系统响应速度：用户操作系统平均响应时间小于3秒。
    - 支持最大并发数：1

### 3.4.2 维护性设计

* 整个应用软件系统应能够连续7\*24小时不间断工作，应用软件中的任一模块更新、加载时，在不更新与上下模块的接口的前提下，以不影响业务运转和服务为原则；
* 应用软件具有较高的自动化程度，如：自动异常调度、自动故障告警、自动任务恢复等；
* 应用软件具备相应容错手段，能容许操作人员的某些失误操作。

### 3.4.3 易用性设计

* 该系统提供基于图形化的友好管理界面；
* 符合IE界面风格：

 系统风格协调一致，体现人性化的友好性管理界面；

 符合用户习惯或者容易被用户接受的管理风格；

 具有容错能力，包括错误诊断和提示，并具有快速的系统反应的管理界面。

* 具有详细的、易懂的联机帮助，协助用户使用。

### 3.4.4 兼容性设计

* 对X86，X64的系统架构机进行兼容
* 对android5.1-android9的系统都能兼容

### 3.4.5 可拓展性设计

* 对其他光照场景的可拓展性的支持
* 对PPT文字的提取，到处的可拓展性的支持

### 3.4.6 可靠性设计

* 系统满足7\*24稳定运行需要。系统平均无故障时间（MTTF）大于4320小时（180天）；
* 系统容错性强，在外系统故障、用户非法操作、数据内容/格式出错的情况下，仍可正常运行平均失效间隔（MTBF）大于4320小时（180天）；
* 系统故障状态下恢复时间短，平均恢复时间（MTTR）小于10分钟；
* 对人工输入的数据以及来自不同接口的数据进行合法性检查，对错误数据进行自动纠错处理。

### 3.4.7 灾备设计

* 系统做到能支持定期的自动数据备份操作，还提供人工备份的操作。
* 提供多种数据输出格式，该输出格式可以方便快捷的倒回原来系统中。
* 真正做到使系统能在数据损坏，丢失等情况下将备份数据倒回，实现数据恢复。

### 3.4.8 安全性设计

* 用户和权限管理
* 数据范围权限的控制
* 对支付通讯进行加密
* 防止数据篡改、攻击

# 4. 参考资料

《JavaEE 开发实战》

《IT项目管理》

《UML与模式应用》