我去了一些他应用的结构，但是也没有很多，大致就是微盘，小米阅读，网易新闻，久记日记，暂时就总结一下，得出一个我们自己的通用的工程结构。  
     小组名为Transcend，目前准备开发活动助手应用，主要实现活动发布，活动报名，活动签到。

第一时间我们应该得出如上图这种整个的架构

base是一个android库，也是一个完整的android工程其中BaseApplication做一些所有app都会用到的基础初始化或者配置。之后其他应用的application应该都继承此BaseApplication。feedback包下存储一些用户反馈等通用功能模块。其实，很多情况下，upgrade模块也可以添加到base工程下，制定统一的软件升级机制

而common只是一个jar文件，oauth为权限开放，view存放一些最通用的自定义view，比如对话框，定制的列表等，utlis为工具类。当然可以根据需要作为android库来开发。

config包存储一些系统配置，比如名称，应用参数等系统级的常量或者静态变量。AppConfig.java存放应用程序的根配置，比如版本，目录配置等等。当然我们有其他大模块的配置，拥有较复杂的用户管理模块的话，完全可以增加一个UserConfig.java中存储用户的一些配置信息等等。

model包，里面不仅存放了实体类xxx.java，而且存放了更高一级的实体类的相关类，比如xxxManager.java,xxxFactory.java.关于这一点，我们参考一下android.jar中结构，发现，Activity.java和ActivityManager.java，View.java和ViewManager.java，Bitmap.java和BitmapFactory.java等等N多类似的一对类都在同一个包下，觉得实体包下存放实体类相应的Manager和Factoty类也是正确的，是我们可以采纳的一种结构。

constant包，此包下存放的都是public static final常量，定义状态，类型等等。

注意：

1. .功能模块和类型模块均可以划分，如果没有需要的话，模块的划分都可以省略。  
   2).activity和service这类组件划分，如果没有需要的话，组件的划分都可以省略。

utlis

oauth

View

upgrade

Base activity

ad

feedback

xxxFactory.java

xxxManage.java

xxx.java

xxxType.java

xxxstate.java

model

Constant

Public static final int HTTP\_CNNMACT\_TIMEOUT=20000;

Public static final int HTTP\_REQUEST\_TIMEOUT=20000;

Public static final int HTTP\_UPLOAED\_TIMEOUT=20000;

Public static String appName=”actHelper”;

commonn

App config

user.config

Config

R.java

Com.Transcend.actHelper

base