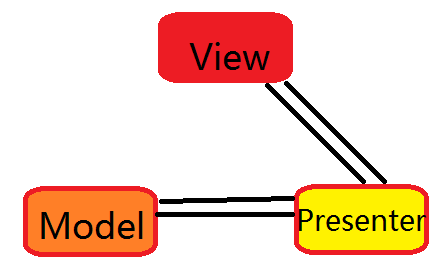
##APP开发采用MVP模式

###图解：



• M(Model) 数据相关层

• V(View) 视图层，如Activity上的布局

• P(Presenter) 纽带层，用来连接Model与View.

###MVP开发在Android中的基本流程

1. View层定义View.interface，用来定义View的行为。一般由Activity或者是Fragment来实现这个接口，它定义了View视图的各种变化，如设置Textview,加载对话框，更新进度条等。

2. Model层定义Modle.interface,这个是用来定义数据层发生变化时的通知接口，因为Model不能直接与View交互，所以它与Presenter交互，然后再通过Presenter间接达到与View的交互。

3. Presenter翻译的意思是主持人，也就是主持场合，控制节奏的意思。Presenter负责具体的业务逻辑，请求数据，把数据送到Model，或者监听Model的数据变化，接受View层的动作，负责通过通知View层的视图变化。

##后端采用基于CodeIgniter框架的LNMP环境开发。

\*\*选用CodeIgniter框架的原因\*\*

- 它一个小巧的框架，但有出色的性能；

- 它一个几乎零配置的框架；；

- 不需要被迫学习一种新的模板语言（当然如果你喜欢，你可以选择一个模板解析器）；

- 有着清晰、完整的文档。

- 容易上手

CodeIgniter框架 的开发基于MVC（模型-视图-控制器）设计模式。MVC是一种用于将应用程序的逻辑层和表现层分离出来的软件方法。

\*\*模型\*\* 代表存储的数据结构。通常来说，模型类包含对数据库进行增删改查的方法。这些方法是对数据处理的重要部分。

\*\*视图\*\* 是要展现给用户的信息。一个视图通常就是一个网页，但是我们的APP中，使用到的视图主要是在web端部分。

\*\*控制器\*\*是模型、视图以及其他任何处理 HTTP 请求所必须的资源之间的中介。安卓端我们将会使用到CI的M/C。

- 前台使用okhttp获取后台数据

- 后台使用php编写API接口