

PROJ 11

按 RUP4+1 表格，给出架构设计关注的功能与非功能性需求。然后将需求分配到 RUP 架构模型。

一、对于海报墙，功能性需求和非功能性需求如下：

功能性需求	非功能性需求
用户登录	性能需求
发布资讯	安全性需求
浏览查看海报	记录日志
分类	
发现	

二、RUP4+1

4+1 指的分别是逻辑视图、进程视图、部署视图、数据视图和用例视图

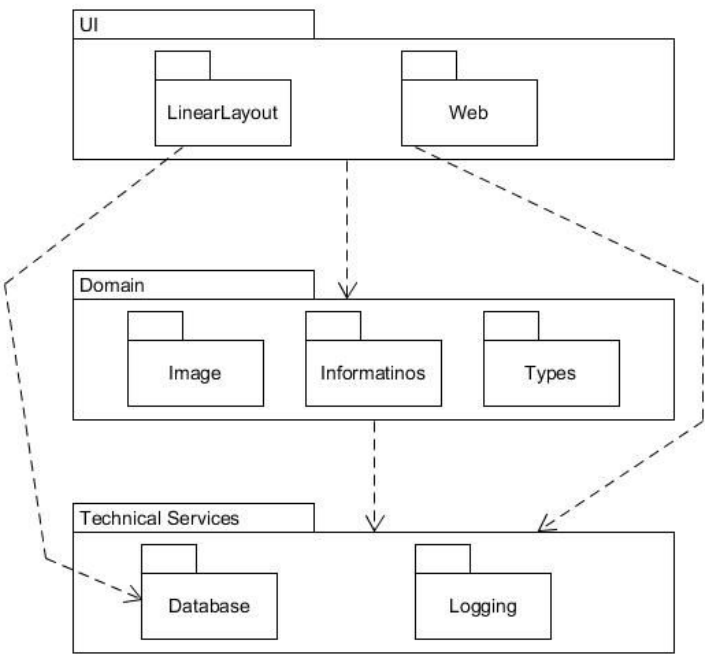
1、逻辑视图

讨论和动机

使用了宽松的分层架构，包括用户界面、领域对象和技术服务。

用户界面层包括了安卓端的 LinearLayout 和 Web 端，他们依赖于领域对象层而实现应用需求，例如浏览海报，查看海报信息等，最下面一层是技术服务层，除了领域对象层依赖于该层外，UI 层的 LinearLayout 依赖于该层的 Database，因为在手机端要浏览海报需要从服务器下载图片；同时，Web 也要依赖于该层，因为发布者要在 Web 端进行登陆和上传海报及其信息到数据库等操作。

包图如下所示。



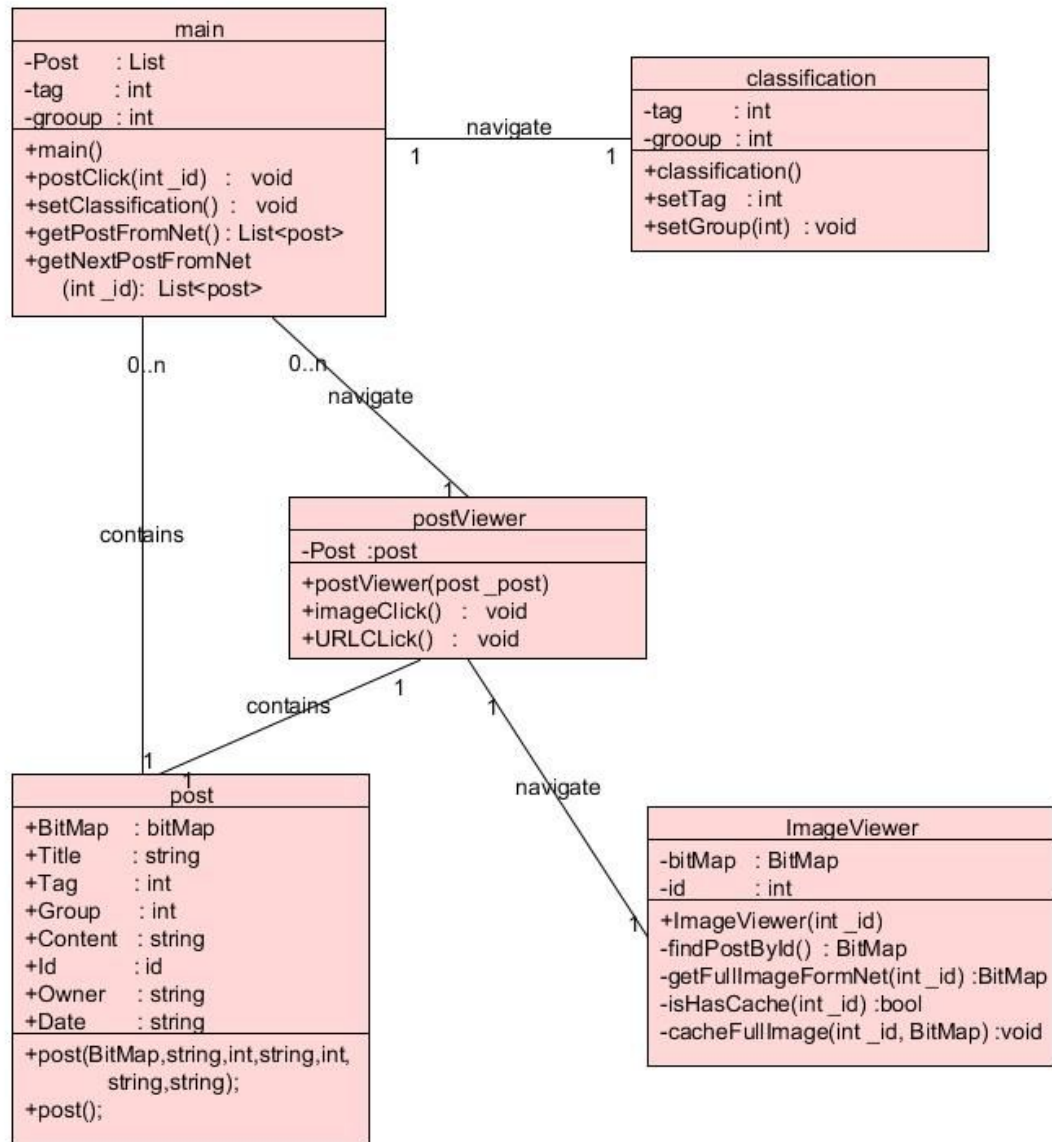
2、进程视图

讨论和动机

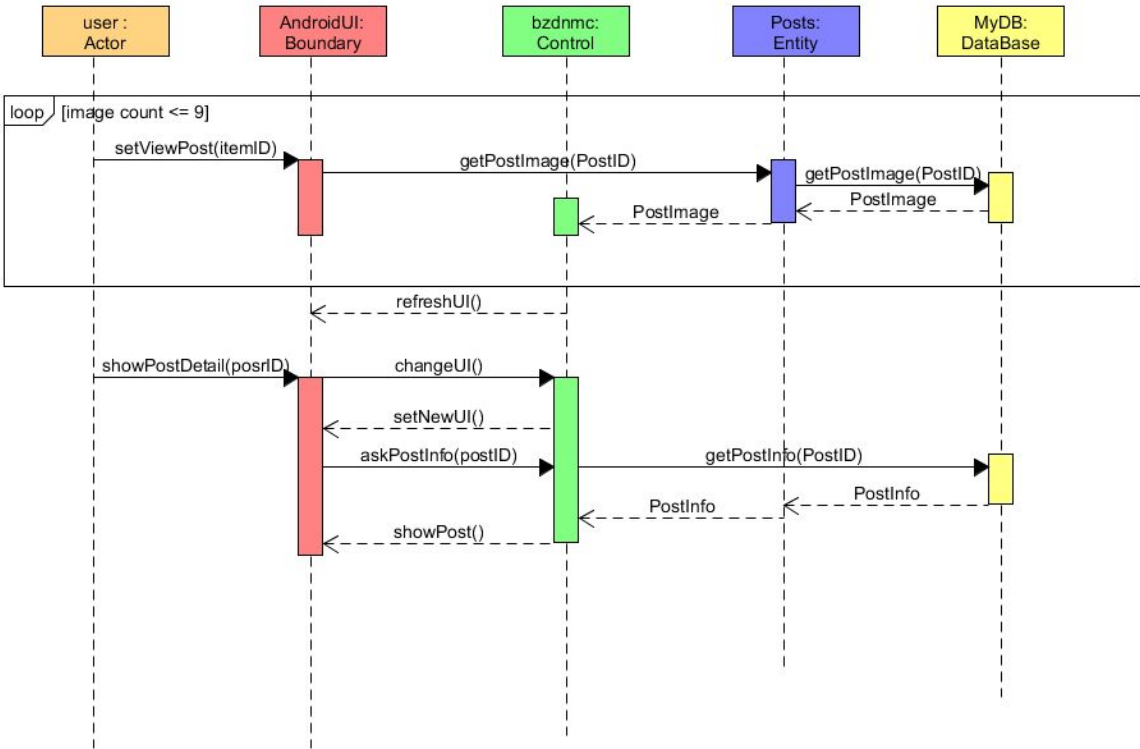
海报墙的流程为：发布者从 web 端上传海报和详情链接 → 数据被存到数据库 → 用户在手机端从服务器读取图片和链接进行浏览及查看详情

类图和顺序图如下所示：

类图：



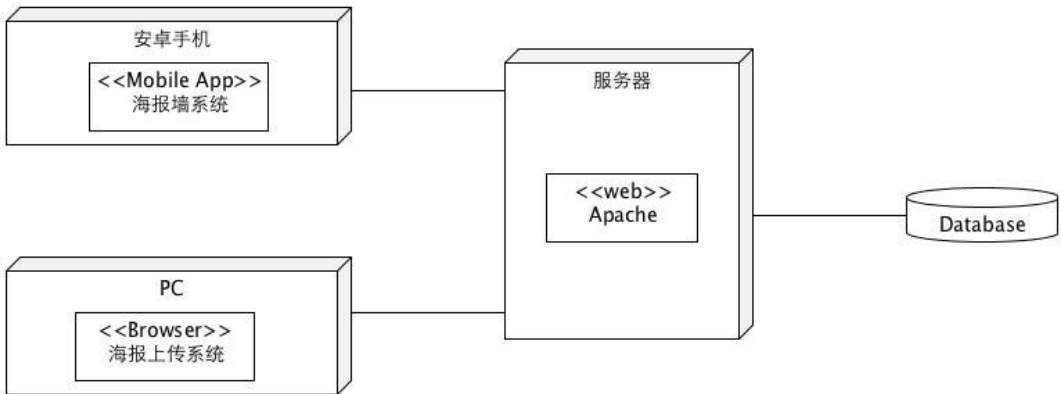
顺序图：



3、部署视图

讨论和动机

为了提高效率和可靠性，服务器、海报上传系统、安卓系统被部署在不同的部分，考虑到制作海报时一般在电脑上进行，所以我们设计了一个 PC 端的海报上传系统，以此方便发布者上传海报及相关链接。部署图如下所示：



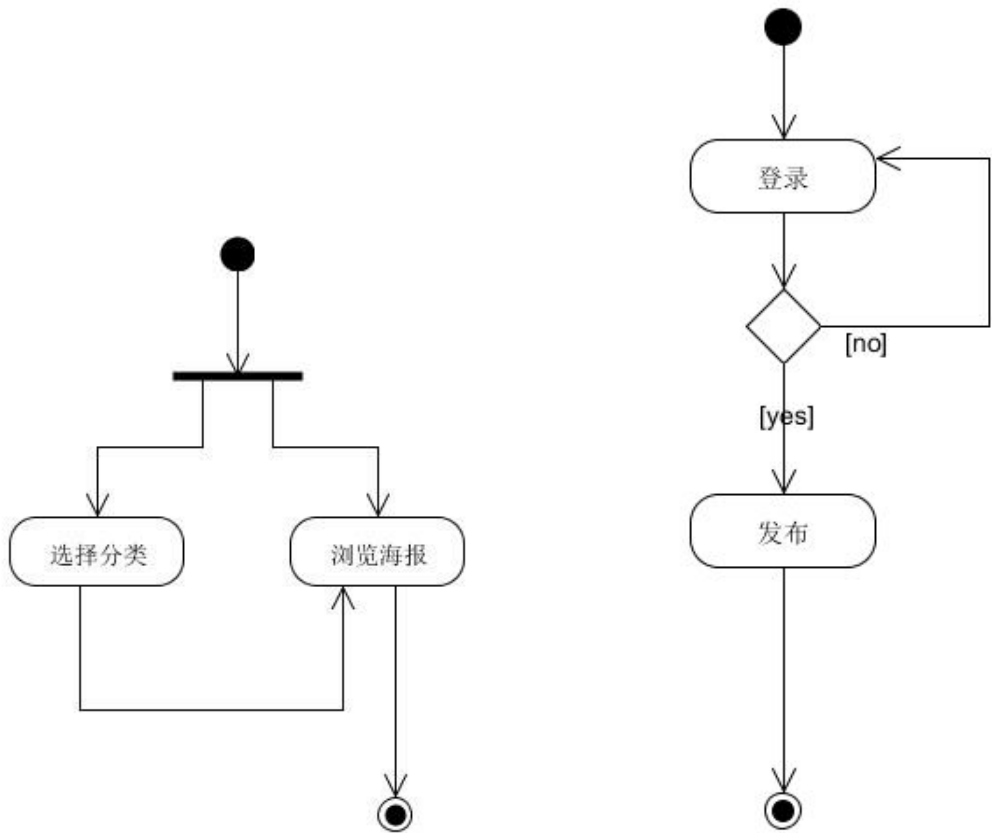
4、数据视图

讨论和动机

在 Android 端，用户可以进行简单的海报浏览也可以进行分类后再浏览；在 Web 端，用户需进行登陆后再进行海报及其信息的上传。使用 TCP 链接将图片传入服务器，同时将对数据插入数据库；根据客户端的请求，再使用 TCP 链

接将请求发送到服务器，服务器根据请求内容获取对应信息并使用 TCP 传回客户端。

基于 Android 和基于 Web 端的活动图依次如下：



5、用例视图

讨论和动机

海报墙中有三个主要用例：浏览海报、查看海报详情、发布海报，这三个用例要在大部分非功能性需求得到满足时才能使用户拥有较良好的用户体验，例如性能需求对应浏览海报用例，安全性需求对应发布海报用例。

其中，最基础最重要的用例是浏览海报，通过实现此用例，可以解决大部分关键架构问题。用例图如下所示：

