

天津工业大学

毕业设计（论文）

基于安卓平台的 114 生活助手系统的设计与实现

姓 名 钱晓闻
学 院 计算机科学与软件
专 业 软件工程
指导教师 杨晓光 张俊
职 称 副教授 工程师

2015 年 6 月 1 日

第一章 绪论

1.1 本课题的研究目的和意义

生活助手类应用是 Android 中普遍存在的应用。用户通过该类型的应用实时获取周边信息。从衣食住行各个方面获取最及时最准确的信息。应用应当满足简单易用, 高效便捷, 功能强大的需求。然而由于 Android 系统的高速迭代, Android 设备的高速发展, Android 应用的开发难度也不断大增加, 项目的开发进度也不断的减缓。本课题的研究目的就是, 通过实现 114 生活助手 Android 系统, 寻找 Android 的快速开发模式。确保以这样的模式, 可以迅速的开发出一款 Android 应用, 并保证其应有的质量。

课题的研究成果将是一款简单易用, 功能强大, 高效便捷的生活助手应用。通过他可以方便的查询当前位置的周边信息, 可以方便的获取交通信息。并且可以简单的在其平台上面实现 P2P 的在线交易。同时更加重要的是, 通过这个项目, 寻找到一个快速高效开发 Android 应用的模式。使得原先相对复杂的 Android 开发模式变得简单有效。

1.2 本课题的研究方式和手段

课题将会通过实现一个基于 Android 的 114 生活助手来实现课题的研究目的。在 114 生活助手这个项目的开发过程当中, 将会尝试使用最新的开源项目和普遍认同的最佳实践方式进行项目的开发。这些使用到的开源项目有: AndroidAnnotation、Robobinding、Gson、Picasso、Bolts、EventBus。而实用的最佳实践方式是根据 Android 官方最新的开发指南, 以及 Android 社区中最为推崇的开发模式总结而成。这些最佳实践模式包括: 实用依赖注入实现代码解耦, 实用 Fragment 替代 Activity 的部分功能、实用 Genymotion 作为调试工具。另外在开发过程当中, 将会使用最新的开发工具开发。这些最新的开发工具包括: 使用 Gradle 进行编译和打包, 使用 AndroidStudio 替换原来的 Eclipse, 使用 git 作为版本管理工具。

第二章 需求分析

基于安卓系统的 114 生活助手系统，在衣食住行各个方面为用户提供方便可靠的信息咨询。其主要提供的功能可以分为以下三类：

1. 基于位置的周边信息。
2. 覆盖航空，火车，长途大巴的交通信息查询系统。
3. 在线交易市场。

其具体的功能又可以划分为以下几点。

3.1 查询医疗信息功能

用户可以查询到当前选择城市的相关医疗信息。这些信息主要包括周边的医院的地址，电话，介绍。通过这些信息为用户提供医疗服务。用户还可以添加相关的医疗信息，以丰富数据。其用例图如图 3-1 所示。

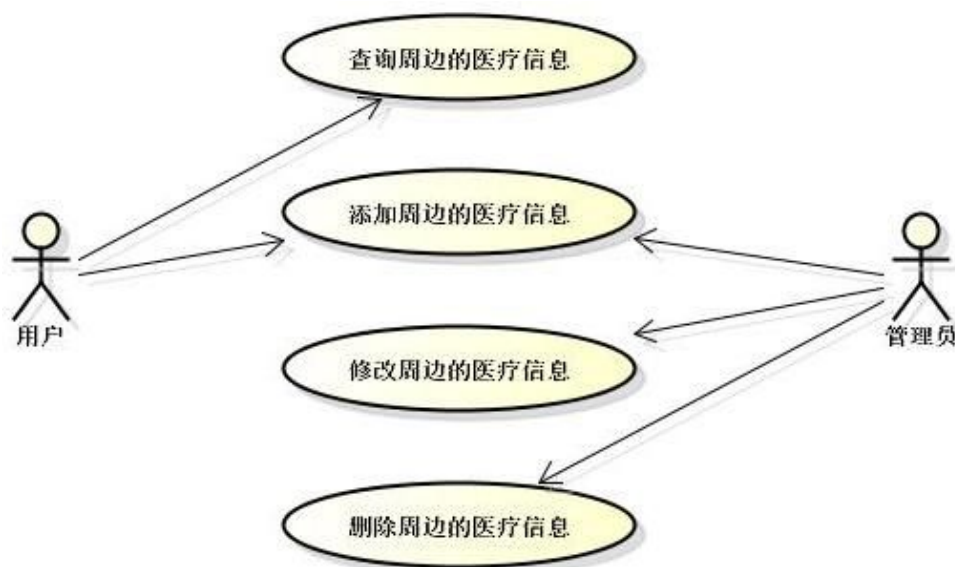


图 3-1 医疗信息的用例图

3.2 查询交通信息功能

用户可以查询当前的实时航班信息，查询到 12306 的实时火车票信息，查询到长途大巴票的信息。根据这些信息为用户提供出行指南。其用例图如图 3-2 所示。

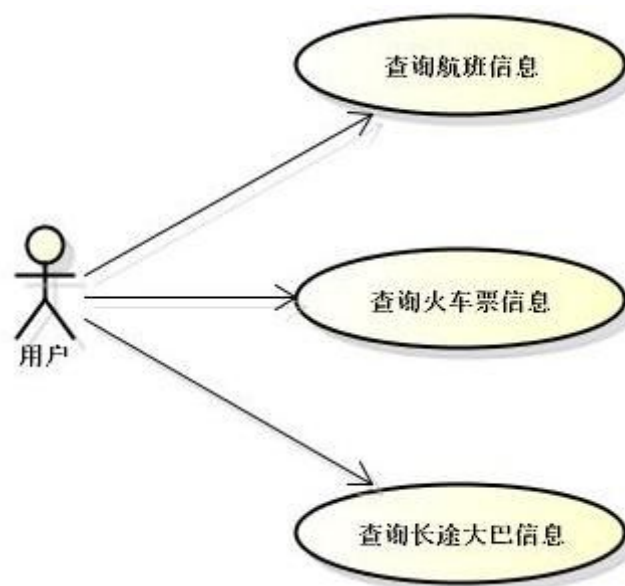


图 3-2 交通查询的用例图

3.3 查询酒店信息功能

用户可以根据当前所在的城市，查询周边的酒店信息。这些信息主要包括酒店的地址，电话，介绍。通过这些信息用户可以随时找到合适的酒店安顿下来。当然，用户还可以添加相关的酒店信息，以丰富数据。其用例图如图 3-3 所示。

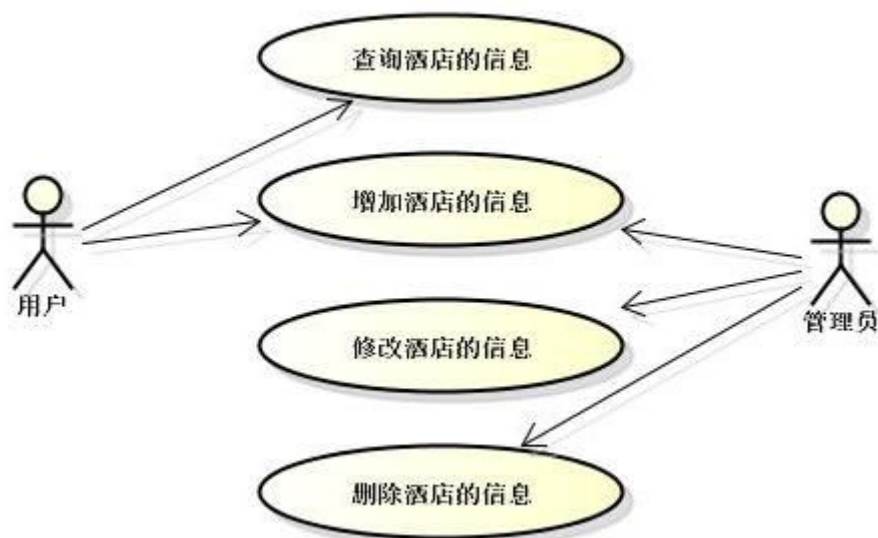


图 3-3 酒店信息的用例图

3.4 查询餐饮信息功能

用户可以根据当前所在的城市，查询周边的餐馆和美食信息。这些信息主要

包括餐馆的地址，电话，介绍。通过这些信息满足用户对于食物和美食的需求。当然，用户还可以添加相关的餐饮信息，以丰富数据。其用例图如图 3-4 所示。

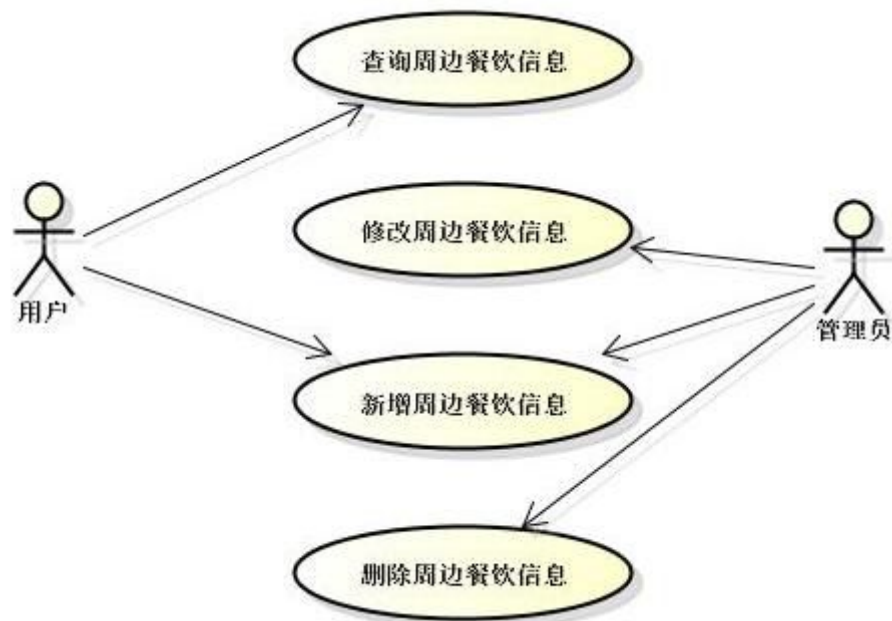


图 3-4 餐饮信息的用例图

3.5 在线市场交易功能

用户可以游览在线市场当中的商品信息。这些信息包括商品的价格，介绍，图等。如果用户需要其中的某一件商品则可以通过线下联系的方式进行交易。当然，用户也可以发布自己的商品信息。从而形成 P2P 的在线交易市场。当然，管理员身份的用户可以审核这些信息，并对其进行管理。其用例图如图 3-5 所示。

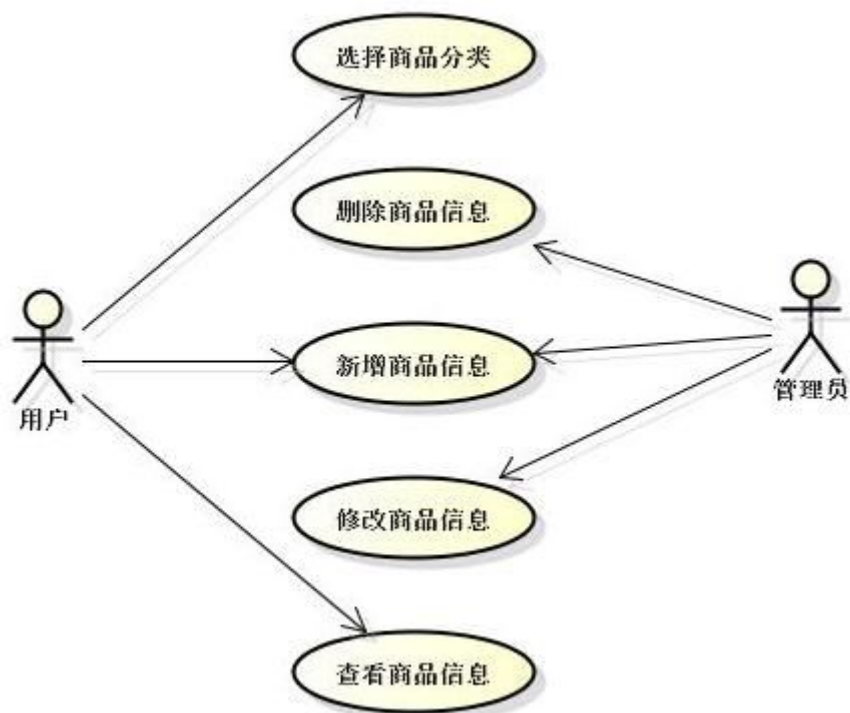


图 3-5 在线交易平台的用例图

3.6 查询家政服务功能

用户可以根据当前所在的城市，查询到关于清洁工，月嫂，搬家公司的信息。通过这些信息用户可以获取到自己所需要的服务。当然，如果用户是相关业务的从业者，也可以在程序当中发布自己的信息。其用例图如图 3-6 所示。

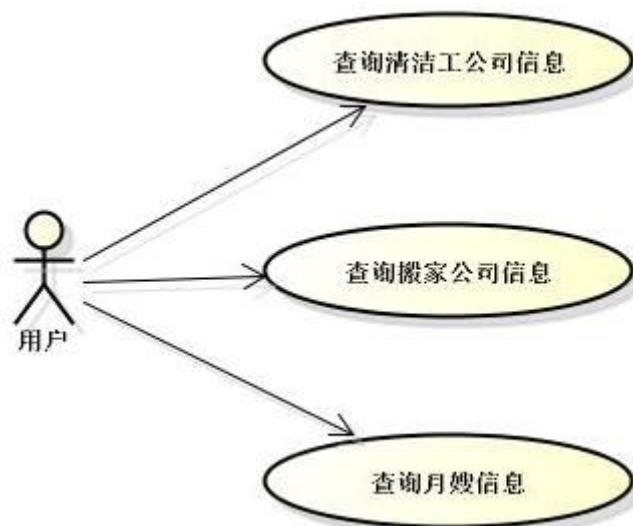


图 3-6 家政信息的用例图

第三章 系统设计

3.1 整体架构设计

系统的整体架构分为三层，如图 3-1 所示，分别有 Activity, Fragment, Model 实现。这样的层次划分，降低了如图 3-2 所示的系统架构。这种过去将大量的业务逻辑全部封装在 Activity 中的实现方式导致的项目臃肿，不易维护，大量重复代码的问题。其中 Activity 主要负责封装一整套的业务逻辑（例如：和交通信息相关的所有业务逻辑），同时负责管理和维护 Fragment 的生命周期。而 Fragment 则负责具体的某个业务逻辑的实现。而 Model 则是真正负责实现该业务逻辑的具体代码。

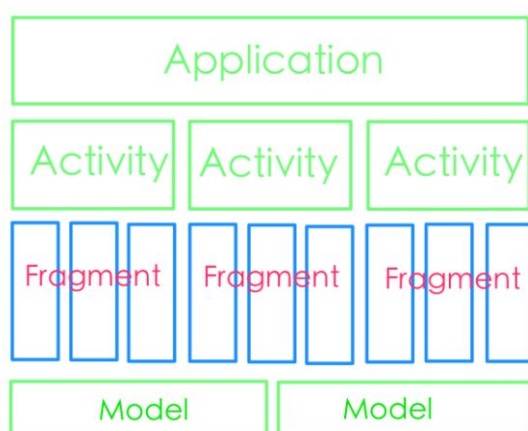


图 3-1 本项目采用的架构



图 3-2 大量 Android 项目中采用的架构

正如上面所说一个模块是由一个 Activity 和若干 Fragment 构成。真正的业务逻辑由 Fragment 实现。而 Activity 作为这些 Fragment 的容器和 Manager。如图 3-3 所示，若干 Activity 构成了整个应用的功能模块，而每个 Activity 由若干 Fragment 构成。

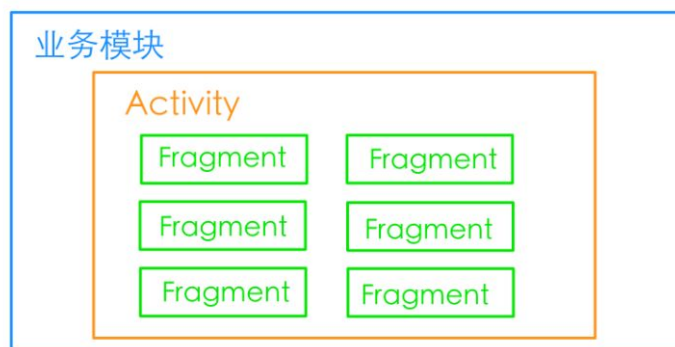


图 3-3 业务模块的实现

而在 Android3.0 之前，系统的功能实现只能由单个 Activity 实现，众多的 Activity 组合成一个模块，就如图 3-4 所示。这种架构的实现方式的缺点在于，Activity 之间的通信方式非常困难。同时也不利于程序代码的解耦工作。使用 Fragment 后的程序架构不仅变的更佳清晰，也有利于后期适配不同尺寸的工作。



图 3-4 业务模块的组成

而各个 Fragment 的具体业务逻辑又由相应的 BusinessModel, ServiceModel, PersistModel 实现。就如图 3-5 所示，若干 Model 支撑着 Fragment 的具体实现。在这些 Model 中，ServiceModel 主要负责与服务器对通信。PersistModel 主要负责数据持续化的工作。BusinessModel 主要负责专业的业务逻辑的工作。这些模块之间相互配合，相互调用，最终使得 Fagment 可以正常使用。

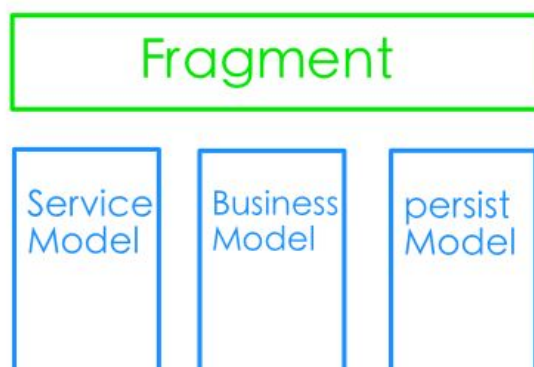


图 3-5 元功能模块的架构

3.2 业务模块的划分

整个系统分为以下 5 个模块。如图 3-6 所示。

1. 功能导航：实现应用的首页，为用户提供功能导航。
2. POI 模块：提供周边信息的检索，管理相应的周边信息。
3. 交通信息模块：提供长途大巴，飞机票，火车票信息的检索。
4. P2P 模块：提供在线交易市场的信息检索与管理。
5. 通用组件模块：提供供其他模块使用的通用模块，包括：城市选择、日期选择、通知弹出、公告信息、用户身份验证。



图 3-6 功能模块划分示意图

3.3 服务端的实现方式

在本项目的实现当中，服务端的数据主要分为三大类。如图 3-7 所示，分为

第三方 WebService、项目 Service、虚拟 Service。其中 WebService 使用的是聚合网的云服务和百度的 LBS 云服务，通过链接他们的 API，获取与周边信息，交通信息相关的全网数据。项目 Service，采用的是 LeanCloud 和 qiniu 的云服务支持。其提供的是项目自身产生的数据，包括用户信息，P2P 信息等。虚拟 Service，则主要是用于项目调试使用的虚拟服务器。他是一个离线的伪服务器，通过他，在开发过程中可以抛弃服务端而直接进行开发，减少了与服务端的耦合。



图 3-7 本项目采用的服务端架构

在选择选择云服务的过程当中，主要考虑的内容有以下几点。

1. 是否能够提供全面准确的数据。
2. API 接口是否足够简单易用。
3. 访问速度是否足够理想。
4. 是否收费。

聚合网云服务和百度的 LBS 云服务，很显然满足以上要求。聚合网拥有千量级别的海量数据，数据类型富含数百种。其中与生活服务相关等数据占据了数十种。另外除了提供封装完好的 SDK 之外，还提供获取 json、xml、jsonp 数据类型的 API。其数据的丰富程度和稳定性都另人满意。百度的 LBS 云服务，基于百度多年的数据积累。数据的丰富程度非常令人满意，唯独遗憾的事情是，百度的 API 接口不是非常完善，很多数据没有办法获取。但是，对于本项目而言已经足够。

LeanCloud 的云服务，是一个 paas 的云服务系统。他过去的称呼是 AvosCloud，后改名为 LeanCloud。他为开发者提供了非常简单的 API 接口，使得开发者可以非常简单方便的为移动客户端提供数据的管理。而 qiniu 云服务则为本项目提供了一个云图床的服务。通过 qiniu 云服务，大大简化了项目在图片上传下载处理的技术瓶颈。本项目选择这些云服务的一个主要的考虑因素在于，Android 客户端的开发过程往往是需要与远程服务器配合的。但是大量的项目证明，完全建立自己的服务器是完全没有必要的。重复造轮子的过程将会大大

增加项目的复杂程度，降低项目的质量，延长项目的周期。而这些云服务技术，为 Android 开发者提供了最为简单和高效的服务端的技术。

第四章 详细设计

4.1 周边信息查询的实现

周边信息查询模块的组成如图 4-1 和 4-1 所示。Fragment 负责对 xml 中的界面渲染，同时调用 POIBusiness 和 POIService 中的方法，实现周边信息的获取和管理。在 POIService 中，他会负责根据当前的地理位置，和用户关注的周边信息关键字返回封装好的 POI 信息数据。这些数据被封装在 POIResultBean 当中，当 Fragment 获取到这些数据之后将会将这些数据渲染到界面当中。

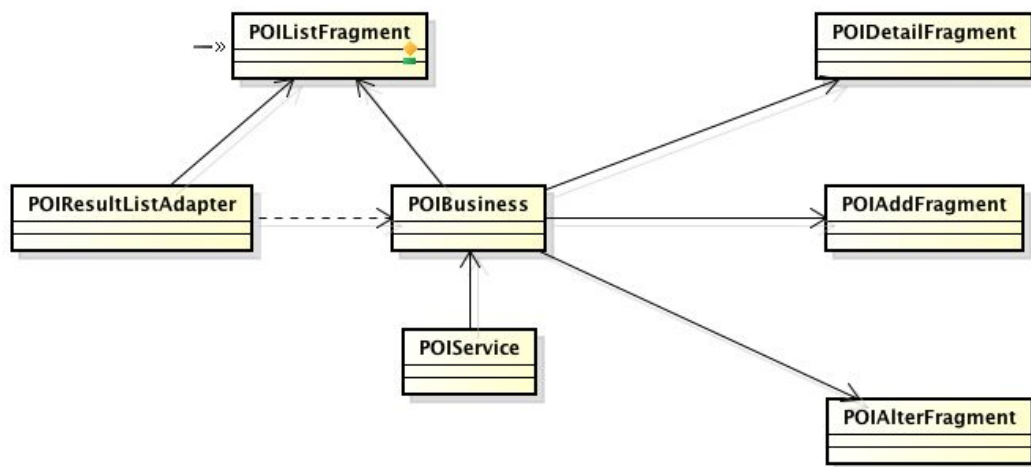


图 4-1 POI 模块的实现

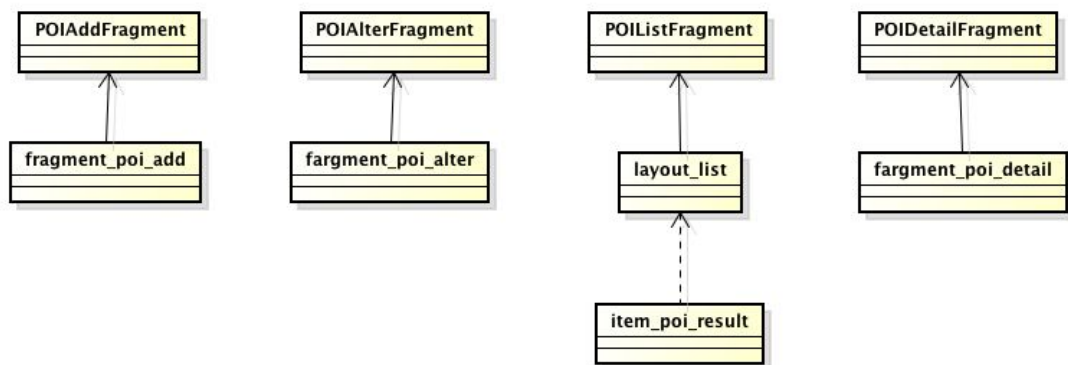


图 4-2 POI 模块的界面组成

值得注意的是 PersonalPOIListFragment 和 POIListFragment，PersonalPOIDetailFragment 和 POIDetailFragment 是继承关系。这两个特殊的 Fragment 除了继承了原本 Fragment 的特性之外，还具有对现有 POI 信息修改的

能力。当然，普通用户只能修改自己产生的 POI 数据，而管理员用户可以修改所有自己产生的 POI 数据。正因为有了这两个 Fragment，程序拥有了数据管理和审核的功能。

4.2 交通信息查询的实现

交通查询模块的组成可以通过图 4-3 和图 4-4 得知。图中 TrainInfoRequestFragment 负责加载 fragment_train_info_request.xml，BusInfoResultFragment 负责加载 fragment_bus_info_request.xml，PlaneInfoRequestFragment 负责加载 fragment_bus_info_request.xml。而所有的 xxxInfoResultFragment 使用的都是 layout_list.xml。

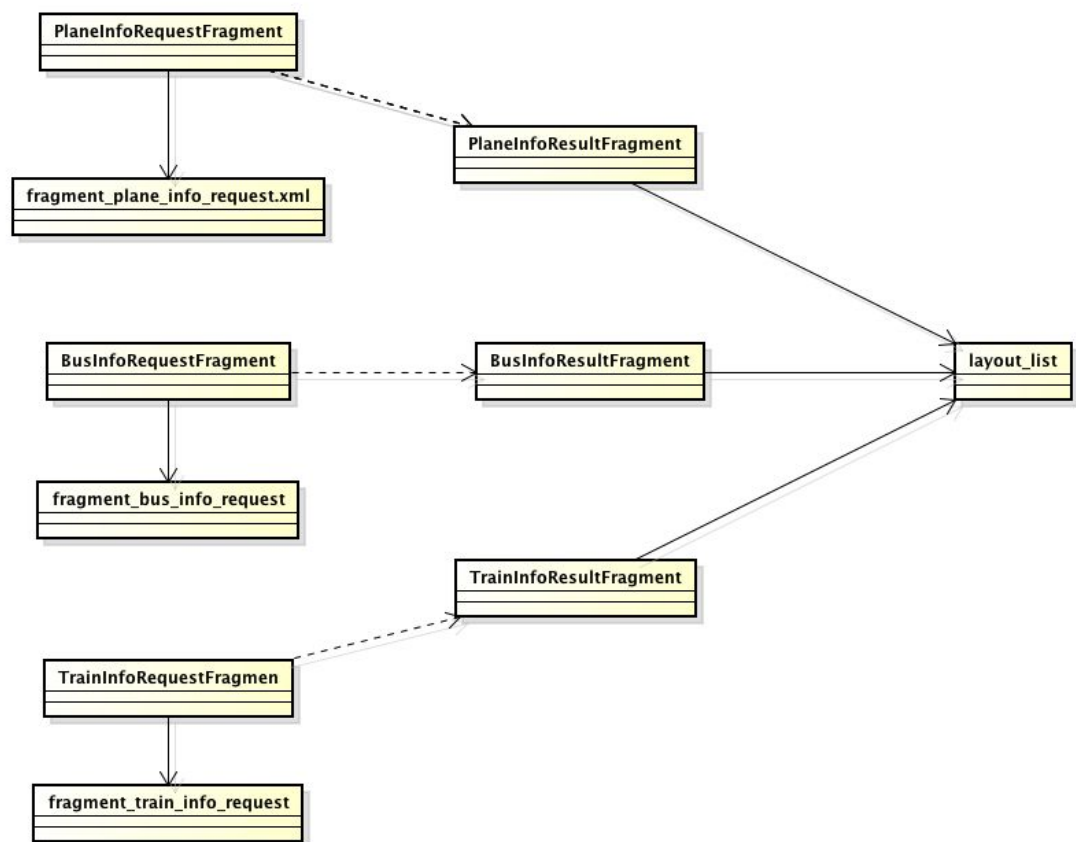


图 4-3 交通查询模块的界面组成

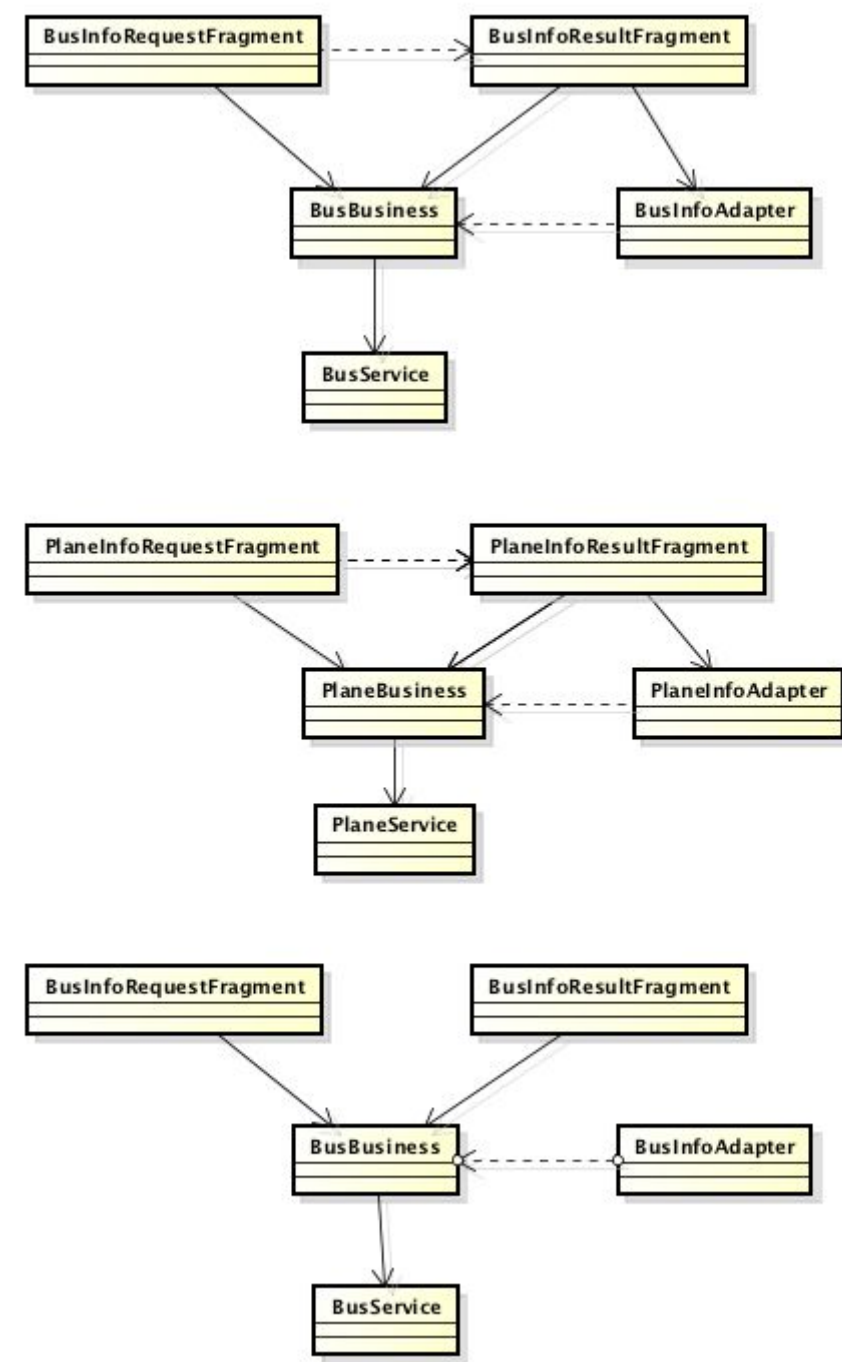


图 4-4 交通查询模块的实现

而在完成界面渲染之后，xxxInfoRequestFragment 通过 BusBusiness，TrainBusiness，PlaneBusiness 封装所要查询的交通数据的请求参数，并跳转到对应的 xxxInfoResultFragment 当中。而在 xxxInfoResultFragment 中则根据传递而来的请求参数数据，BusService，TrainService，PlaneService 请求获取相应的交通信息。

5.4 自由交易市场的实现

在线交易市场的组成如图 4-5 和 4-6 所示。其中 P2PListFragment 负责加载 fragment_p2p_list.xml，P2PDetailFragemt 负责加载 fragment_p2p_detail.xml，P2PAddFragment 负责加载 fragment_p2p_add.xml，P2PAlterFragment 负责加载 fragment_p2p_alter.xml。在各个 Fragment 加载好数据之后，将会调用 P2PService 和 P2PBusiness 获取在线交易市场的数 据，并对这些数据进行管理。

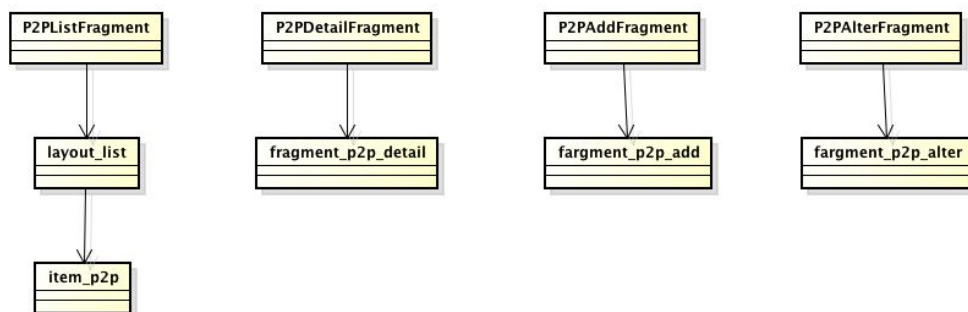


图 4-5 P2P 模块的界面组成

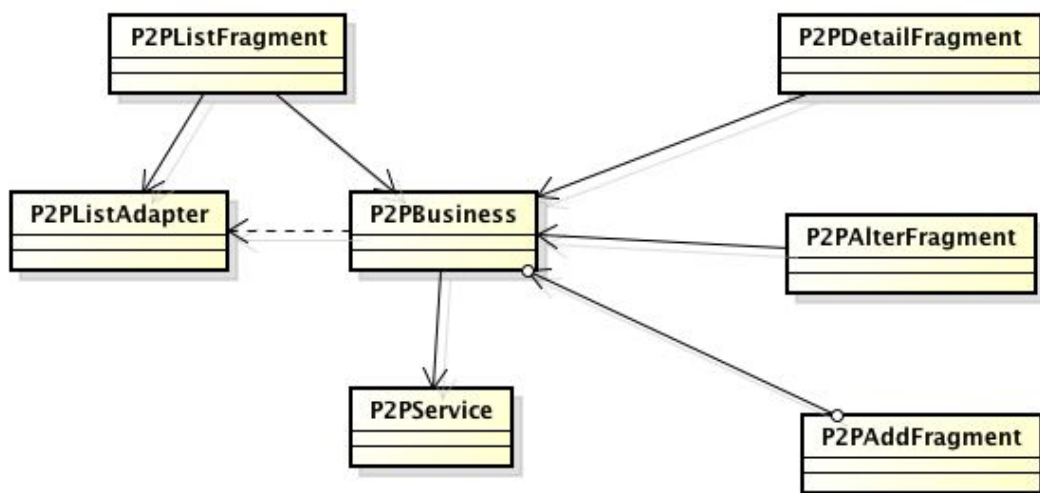


图 4-6 P2P 模块的功能实现

值得注意的是，PersonalP2PListFragment 和 P2PListFragment，PersonalP2PDetailFragment 和 P2PDetialFragment 之间是继承关系。他们在原有父类的基础上增加了对商品数据修改的功能，正是因为这些功能的存在，用户才可以对上述的数据进行修改和管理。当然如果当前的用户是管理员用户，他可以对所有的商品数据进行修改。

在 P2PService 中，将会负责访问服务器，获取相关的商品数据。服务器会根据相应的请求参数返回封装好的 json 数据。之后 P2PService 将会对这些 Json

数据进行解析，并将其封装成为 P2PItemBean 返回给响应的 Fragment。

结论

在本次的毕业设计当中，完成了基于 Android 的 114 生活助手系统的设计与实现。并通过项目的开发，尝试寻找一种快速开发的开发方式。在最终阶段完成了课题任务书当中的所有目标 and 需求，完成并实现了生活助手系统所具备的周边信息查询，交通信息查询，在线交易平台等功能。根据完成的项目你可以查询到航班、火车、长途大巴的实时交通信息。另外你还可以根据自己选择的城市查询到相关的周边信息。这些周边信息包括：酒店信息、餐饮信息、医疗信息、家政信息。

在项目当中，我尝试使用了 Android 开发的最新前沿技术。这些技术有的来自 Andorid 官方，有的来自开源社区。经过这次实际项目的考验，这些原本只是文章中提及到的技术大多被证明是非常有效的。当然其中也有少许的技术并没有如项目开始时预期的那样完美，其中仍然有许多另人遗憾的地方存在。其中值得作为总结的内容如下：

1. 使用 AndroidAnnotation 这样的开发框架，将会大大降低重复的代码，让开发变的无比的顺畅。但是随着项目的无限增长，该框架的使用可能依旧会不可避免的造成代码的冗余。另外 AndroidAnnotation 项目作为 Android 开发的一个整体框架来说，其可能存在的最大的问题在于，其和其他开源框架的配合使用。在本项目的开发当中，已经不仅一次的遇到了 AndoridAnnotation 和其他开源项目的冲突问题。

2. 使用 Fragment 替换原来的 Activity 的部分功能，使得开发变的更加灵活高效。但是同时这也引入了新的问题，而最主要的问题是在于在 Andorid 的官方文档当中，对于 Fragment 的具体使用方式似乎还是相对限制的比较死板的，这造成了在很多的开发当中，开发者一层不变的按照官方文档的思路使用 Fragment。但是根据本次项目的实际经验来看，似乎这样的做法是不值得赞同的。Fragment 应该被更加灵活和丰富的使用。

6. 对项目架构的划分使得项目的架构变的非常的清晰。多层的架构划分，使得 Android 很好的实现了 MVC 的架构逻辑。将 UI 代码放置在 xml 布局文件当中。Model 层主要负责正真的业务逻辑。而 Activity 和 Fragment 作为 Control 层出现。这样的设计，要远远比单独使用 Activity 的要好很多。最最重要的是，这样的划分，也使得项目本身变的可以测试。当然，不能说这样的设计就是非常完美的。由于 Android 项目的特殊性，其层次的划分还不能被非常清晰的划分出来，在各个层次之间依旧存在很多交集的地方。

7. 由于课题的研究时间和经历的现实，本项目在单元测试上面所耗费的精力几乎可以为零。整个项目的开发都是朝着不断向前的步骤进行的，没有按照最佳实践的推荐对项目本身进行单元测试。同时在项目的过程当中，重构和代码回

顾的过程也屈指可数，因此虽然在众多开源框架的帮助下，项目已经变的异常的简洁明了。但是其中依旧存在很多糟糕的设计需要改正。最后，由于设备的限制，项目最终的运行测试只有在魅蓝 note 和 google nexus7 上面进行。因此项目本项目的稳定性和适配性能另人担忧。