本科毕业设计（论文）

基于 020 的线上线下手机壳定制系统的设计与实现

Design and Implementation of Phone Shell Customization System based on O2O

学　　院：软件学院

专　　业：软件工程

学生姓名：ＸＸＸ

学　　号：ＸＸＸ

指导教师：ＸＸＸ

北京交通大学

2016 年 5 月

学士论文版权使用授权书

本学士论文作者完全了解北京交通大学有关保留、使用学士论文的规定。特授权北京交通大学可以将学士论文的全部或部分内容编入有关数据库进行检索，提供阅览服务，并采用影印、缩印或扫描等复制手段保存、汇编以供查阅和借阅。

（保密的学位论文在解密后适用本授权说明）

学位论文作者签名： 指导教师签名：

签字日期： 年 月 日 签字日期： 年 月 日

中文摘要

摘要：如今使用互联网获取信息的人越来越多，根据最新的统计数据显示，截止到目前，使用互联网上网的总人群数达到 8.5 亿，人们使用网络获取最新信息、在线支付、查找地理位置等实用技术，在这样的环境下，产生了 O2O 这样一个新的概念。O2O 不同于传统的 C2B，在 C2B 模式下，工厂会事先根据设计文档进行产品的大规模生产。而 O2O 更多的是按照顾客需求定制，满足顾客的个性化需求[13]。而手机壳定制系统就是这样的一个平台应用。

本文描述的主体，即手机壳定制系统，是根据产品经理的需求，由本人负责实现，经过近六个月的实现与测试，已上线试运行，目前状态良好。该系统采用一种更加新颖的依赖用户的方式，由用户设计手机壳，然后工厂端进行制作，最后将手机壳安全地送到用户手中。该系统包含三个部分，分别是“魅印”手机壳定制 APP、手机壳工厂后台管理系统和后端服务器。其中“魅印”手机壳定制 APP 包括 5 个模块——定制模块、用户管理模块、订单管理模块、支付模块和社区分享模块；手机壳工厂后台系统包括 4 个模块——订单管理模块、用户权限、数据统计、物流管理模块；后端服务器提供两大类接口，分别是客户端相关接口、工厂端相关接口。

本文详细描述了该手机壳定制系统的实现过程，从最初的需求分析到概要设计、详细设计以及所使用的相关技术，符合软件工程规范的开发流程。在本文中，使用了大量的用例图，流程图和时序图以及类图对系统的具体模块业务进行说明。

本文包括绪论，主要介绍项目的来源，主要内容和个人工作，然后是需求分析，分模块进行功能描述，然后是系统架构设计，包括总体架构设计、数据库结构设计和设计模式，然后是系统模块设计与实现。论文的最后是对毕业设计论文各项工作的总结。

关键词：O2O；手机壳；定制

ABSTRACT

ABSTRACT: Nowadays with the rapid development of Internet, Chinese Internet users may reach 850 million people while the real-time online processing, GIS and mobile payment technology become mature in 2015, which is the basis for the development of O2O application, according to Chinese government statistics. O2O, different from C2B which create a relationship with the customers in a way of enlightenment, is more customized to meet the individual needs of customers. The Phone Shell Customization System is such a platform.

The subject of the essay is the Phone Shell Customization System, which took me six months to complete it from general design to testing phase, 4.5 months of developing and 1.5 months of testing respectively. There are three parts in the systems, which are mobile APP, Factory Client and Service. Each part has several module, as for mobile APP, five modules are included, and there are four modules in Factory Client part. The Service provides interfaces for both of them.

This paper describes the implementation of the platform from the initial requirement analysis to general design, detailed design with related technologies used, which complies with the standard engineering software development process. In this paper a lot of user case diagrams, flowcharts and sequence diagrams and class diagrams of the specific module are included. There are 72 pictures, 17 tables in it.

Firstly, the paper introduces the topic related background, the present situation, the content and research significance, respectively, from the demand analysis, structure analysis, detailed design and implementation of aspects of the system are discussed in this paper. In the last part of the thesis is the summary of the thesis and the work to be.

KEYWORDS： O2O; Customization; PhoneShell; C2B

目　　录

1 绪论

1.1 课题研究背景

如今使用互联网获取信息的人越来越多，根据最新的统计数据显示，截止到目前，使用互联网上网的总人群数达到 8.5 亿，人们使用网络获取最新信息、在线支付、查找地理位置等实用技术，在这样的环境下，产生了 O2O 这样一个新的概念。020 的字面含义是线上和线下[1]，通过将线上与线下结合在一起，免去了中间环节，直接对接的好处是增加了用户的参与度和灵活度，使用户可以直接与商家对接。

手机壳定制系统将用户与工厂直接对接，用户设计，工厂生产并配送给用户，是本课题研究的主体。

1.2 国内外研究现状

目前国内各大电商陆续推行应用 020 模式，从发展来看，020 模式被证实已经很快被广大顾客接受。在国外 020 序幕也早已开启，不管是创新公司，如 Groupon，还是商业巨头 Google 纷纷陆续推行 020 战略，同时也涌现了一批 020 模式的新生企业。

近观国内 020，发现有这么几点要遵循[2]：

- 诚信与信誉：每一个推行 020 的商业品牌都必须讲究信誉。
- 商家资质： 商家必须努力提高自己的产品水平和质量，以满足顾客的需求。
- 不断发掘创新：提高创新意识，努力向走在前沿的企业看齐。

虽然 020 离我们很近，但是要成为每个人都希望的样子，必须保证服务质量和适时创新。而手机壳定制系统就是在保证产品质量的情况下的一个创新应用平台。

1.3 课题研究内容及意义

该课题想要搭建一个定制手机壳的服务平台，让客户端用户可以定制自己的个性化手机壳图案，在线支付进行购买，并可以选择将自己的作品分析到社区或其他的社交空间，同时将客户端用户与工厂端直接连接起来，形成一个顾客订购，工厂直接生产发货

的生产链，这样不仅满足了顾客购买手机壳的个性化需求，同时工厂端可以投入更多的人力物力去提高产品的质量[14]。这样的平台让每一个用户在使用的过程中都有一种顾客至上的感觉，用户共享也无形中带来了更多的新用户，手机壳的销量也将逐日攀升。

1.4 个人的主要工作

在手机壳定制系统中，我完成了“魅印”手机客户端，手机壳工厂后台系统和服务器三部分的所有功能实现。

在“魅印”手机壳客户端，我实现了定制模块、用户管理模块、订单管理模块、支付模块和社区分析，五个模块功能的介绍如下：

● 定制模块提供用户编辑自定义手机壳图片的功能，用户可以选择手机相册的图片，进行缩放、旋转、加入第三方特效等。

● 用户管理模块提供用户信息管理、注册、登陆、登出、用户偏好管理，包括手机型号信息、用户默认地址信息等。

● 订单管理模块类似淘宝、京东的订单处理界面，让用户可以新增、查看和修改订单。

● 支付模块目前接入支付宝，用户可以选择立即支付或者下订单后在一定时限内支付。

● 社区分析模块按照热门、最新两种方式进行排序，用户可以发布自己的作品，也可以为其他用户的作品点赞，同时系统支持将作品分享到常用的社交平台，如 QQ 空间、微信朋友圈、新浪微博、人人等。

● 在手机壳工厂后台系统，我实现了订单管理、用户权限管理、数据统计和物流管理，四个模块的功能介绍如下：

● 订单管理模块：工厂后台的订单管理模块根据订单的三个状态，未制作、制作中、已完成来综合处理订单。工厂端做出处理后，订单状态会向后迁移，同时可以及时查看订单的详细信息和顾客的相关开放信息。

● 用户权限管理：由于不同工厂后台的用户，职责不一样，能够浏览的信息权限也不一样，系统允许管理员给后台用户分配或撤销权限，相应用户登陆后，只能查看自己的相关内容。

● 数据统计模块：该模块属于商务智能，顺应大数据的趋势，以数据可视化的方式统计订单的相关信息。

● 物流管理：该模块与物流第三方应用对接，工厂给制作完成的订单进行发货，查看物流状态。

● 服务器提供两类接口，我开发了客户端相关接口和工厂端相关接口。接口的具体定义请参看后文具体介绍。

1.5 论文组织结构

论文是以一款名为“魅印”手机壳定制 APP 为研究对象，从需求分析到概要设计，再到详细设计和具体的系统设计，主要分为六个模块，分别是绪论、系统理论和技术支持、系统需求分析、系统架构设计、系统模块详细设计与实现、总结。

● 绪论：五个小节，分别介绍课题研究背景、国内外研究现状、课题研究内容和意义、个人的主要工作和论文结构设计。

● 理论与技术支持：主要介绍了系统开发过程中主要用到的技术，同时解释了一些论文中出现的缩略词的含义，帮助了解论文的内容。

● 系统需求分析：从三个方面，分别是需求分析、功能分析和质量分析阐述系统的要求。

● 系统架构设计：对整个系统进行架构分析，详细描述系统的数据结构、设计模式、模块处理流程以及整个系统实现的理论支持。

● 系统模块设计与实现：涉及到具体的模块设计，对每一个模块进行详细的分析和建模。

● 总结：结束语，对整篇论文的撰写进行总结并对所有给予我支持和帮助的人士表示感谢。

2 理论和技术支持

2.1 “魅印”手机壳定制 App 理论基础

2.1.1 XML

XML 全称为 Extensible Markup Language，即可扩展标记语言，是一种用于标记电子文件使其具有结构性的标记语言，类似的还有 JSON 等。

XML 文档定义的方式有：文档类型定义（DTD）和 XML Schema。DTD 定义了文档的整体结构以及文档的语法，XML Schema 用于定义管理信息等更强大、更丰富的特征。

在开发 Android 应用时，XML 文件的使用大大简化了界面的绘制和基础配置参数设置的工作。

2.1.2 Android 布局

Android 布局用来设计界面，目前使用的 IDE（如 eclipse，Android Studio1.5）支持拖拽式可视化界面设计。目前 Android 有四种基本布局，分别是 LinearLayout，RelativeLayout，FrameLayout 和 TableLayout。在这些布局的内部除了可以放置控件外，也可以在布局中嵌套布局，这样无论多么复杂的界面都能实现[3]。先分别介绍这四个基本布局，然后说明它们之间的关系。

- LinearLayout：称为线性布局，这个布局会将它所包含的控件在线性方向上依次排列，并可以通过 android:orientation 属性指定排列方向是 vertical 还是 horizontal。
- RelativeLayout：称为相对布局，和 LinearLayout 的排列规则不同，它可以通过相对定位的方式让控件出现在布局的任何位置。
- FrameLayout：称为帧布局，这种布局没有任何定位方式，定义在该布局里的控件都会呈现在左上角，且彼此覆盖在一起。
- TableLayout：表格布局，允许使用表格的方式来排列控件。

对于布局与控件之间的关系：

Android 的 UI 界面都是由 View 和 ViewGroup 及其派生类组合而成的。AndriodUI 界面的一般结构可以参考如下的示意图 2-1

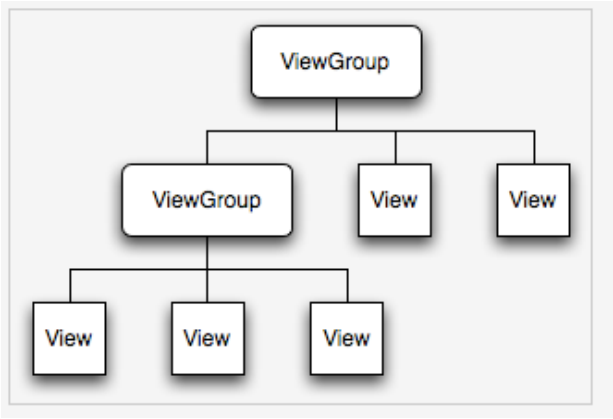

图 2-1 AndriodUI 界面结构

Android 的布局里可以嵌套子布局，也可以包含基本控件。如下的示意图 2-2 展示了布局与控件之间的关系

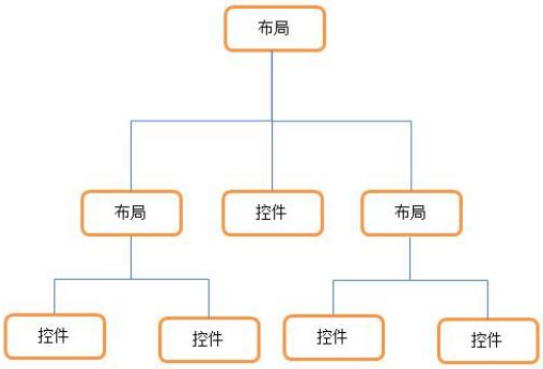

图 2-2 Android 布局与控件的关系

2.1.3 Gson 与 Json

Gson 是 Google 开发的 Java API，用于转换 Java 对象和 Json 对象[4]。它使用实体类字段名称作为 Json 字段的名称，并赋予对应的值。Gson，不同于 Json 类库，不需要使用 annotation 来标识需要序列化的字段，同时又可以通过使用 annotation 来灵活管理，也可以使用接口 JsonSerialiser，其中 serialize()方法返回的类型必须是一个 JsonElement 类型的实例。JsonElement 有四种具体的实现类型：

- JsonPrimitive——例如一个字符串或整型。
- JsonObject——以 JsonElement 名字（类型为 String）作为索引的集合。类似于 Map<String, JsonElement>集合。
- JsonArray——JsonElement 的集合
- JsonNull——值为 null

如图 2-3 展示

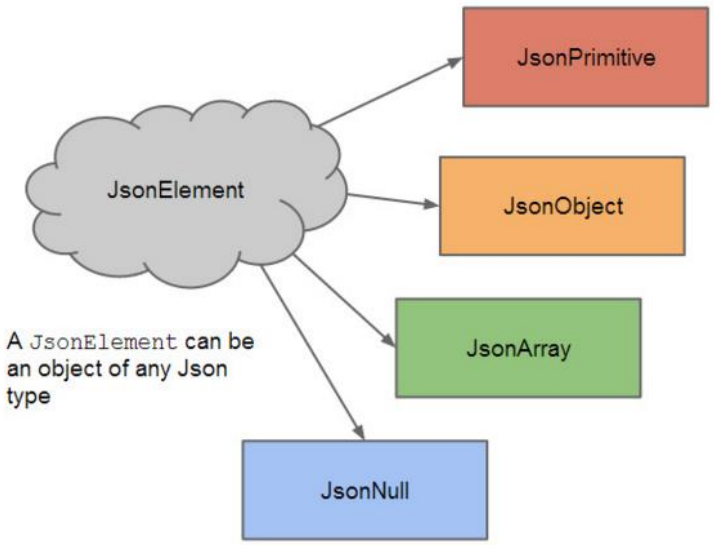

图 2-3　JsonElement 的四种类型

2.2 手机壳工厂后台系统

2.2.1 AJAX

AJAX 全称为 Asynchronous JavaScript And XML，即异步 Javascript 和 XML，是指一种创建交互式网页应用的网页开发技术。使用 AJAX 意味着可以在不重新加载整个网页就可以对网页的某个部分进行刷新[5]。如图 2-4 展示了传统网页浏览的过程

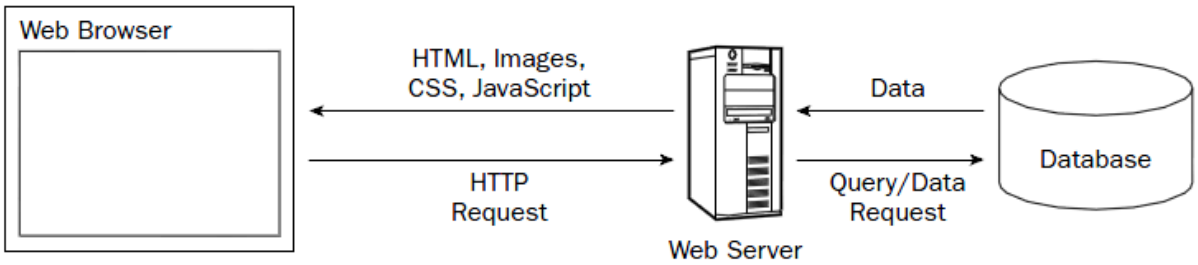

图 2-4　传统网页浏览过程

使用 AJAX 之后，通过 AJAX 在用户交互方面有了很大的改进，用户可以不用为提交表单而长时间等待服务器的应答。如图 2-5 展示了使用 AJAX 技术之后的过程

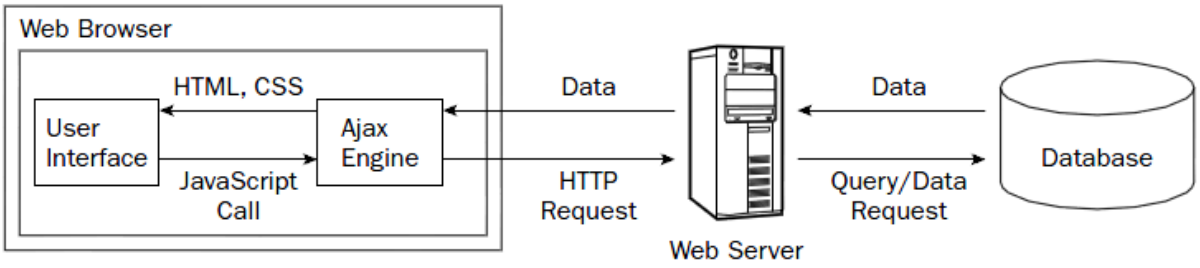

图 2-5 AJAX 引入后的过程

2.2.2 JQuery

JQuery 是继 prototype 之后又一个优秀的 Javascript 库。JQuery 里自带有很多动画效果，且与 AJAX 技术兼容，可以很方便的使用 AJAX 来更新页面局部的内容。同时 JQuery 特有的提取元素的方法，使之成为越来越火的前端脚本语言库[6]。

JQuery 能够使 HTML 页面与 js 脚本分离开来，这样简化的 HTML 页面的内容，也将前端页面的逻辑处理部分从显示代码中抽离出来了。总的来说它是对 javascript 脚本语言的一次封装，可以更加灵活、方便地去实现前端的脚本代码。

2.2.3 Bootstrap

Bootstrap 由 Twitter 的设计师 Mark Otto 和 Jacob Thornton 合作开发，是一个 CSS/HTML 框架。

使用 Bootstrap 可以有预编译的 CSS 文件，这样提高了开发的效率，而且还可以通过修改源码，来定制自己的样式、风格。同时 Bootstrap 提供的界面栅格化处理和流式界面的技术[7]让使用它的网站能够在多台不同尺寸的显示器下正常显示，可以很好的适配各种移动设备[8]，最后它提供了了丰富的说明文档让使用者查阅，可以很快的学习掌握。

2.3 服务器

2.3.1 MySQL5.0

MySQL 是一个开源的，目前使用最多的关系型数据库管理系统，因为它安装方便，维护成本低，在性能上也可以满足中小型网站对数据存储的要求。

2.3.2 Filter

过滤器技术是 servlet2.3 新增加的功能。它使用户可以改变一个 request 和修改一个 response。Filter 不是一个 servlet，它不能够产生一个 response，但它可以在一个 request 到达 servlet 之前预处理 request，也可以在 response 离开 servlet 时处理 response。换一种说法，Filter 其实是一个“servlet chaining”，包括

1. 在 servlet 被调用之前截获；
2. 在 servlet 被调用之前检查 servlet request；
3. 根据需要修改 request 头和 request 数据；
4. 根据需要修改 response 头和 response 数据；
5. 在 servlet 被调用之后截获；

3 手机壳定制系统需求分析

手机壳定制系统是一个让用户定制个性化手机壳的系统，其主要的功能是能够让用户通过各种手势操作绘制照片或图片，并可以将自己的作品分享到社区或主流的社交平台上。用户提交订单并支付完成后，工厂后台管理系统会收到订单信息，并开始制作，用户可以实时的查询目前订单的状态，工厂后台管理系统的物流模块也会实时更新物流状态，确保手机壳最终安全地送到用户手中。

3.1 项目需求概述

3.1.1 需求概述

目前各大电商都陆续加入 O2O 的大阵营，如团购网，电影票预定网站，手机销售网，服装电商等等。由于商家将服务和各种优惠信息主打到互联网上，人们对手机等可上网的移动设备的需求量越来越大。很多人为了追求潮流，选择 5000 元以上的智能手机比比皆是，但在保养，选购手机壳的时候，却普遍发现市场上的手机壳质量良莠不齐，虽然电商购物网站上有提供各种品牌的手机壳，但价格虚高，图案单一，很难满足用户个性化的需求。手机壳制作目前存在如下几个严重的弊端，首先手机壳样式设计者由于自身的审美限制和有限区域的市场调查，基本无法获取到普适的样式设计和图案风格，可想而知，做出来的手机壳难以吸引顾客的眼球。然后工厂生产手机壳没有成规的销售计划，存在生产富余，滞销库存的可能，最终导致新的手机壳样式无法及时上市，市场上的手机壳种类也就越来越来少，导致一个销售恶链，进入恶性循环。

现阶段的手机壳定制系统的难度在于手机壳定制方面，手机壳定制系统采用一种更加新颖的依赖用户的方式，由用户设计手机壳。因此首先“魅印”手机壳定制客户端必须满足定制手机壳的各种个性化需求，同时允许用户分享自己的作品，另外工厂端也要实时的统计数据和获取订单的最新状态。

该系统包含“魅印”手机客户端，手机壳工厂后台系统和后端服务器三个部分。系统功能模块分别对各部分进行划分。对于“魅印”手机客户端，可以划分为以下五个模块：定制模块、用户管理模块、订单管理模块、支付模块、社区分享模块。对于手机壳工厂后台系统，可以划分为以下四个模块：订单管理模块、用户权限管理模块、数据统

计模块和物流管理模块。后端服务器主要为手机端和工厂端提供访问数据信息的接口。系统的模块分解图见图 3-1。

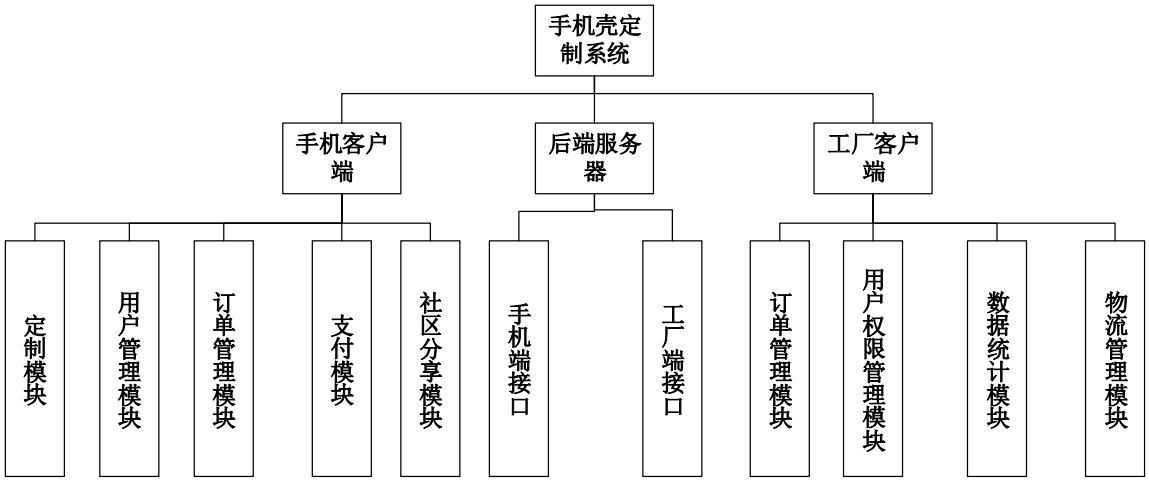

图 3-1 系统模块分解图

3.1.2 业务流程图

手机壳定制系统的业务流程图如图 3-2 所示：

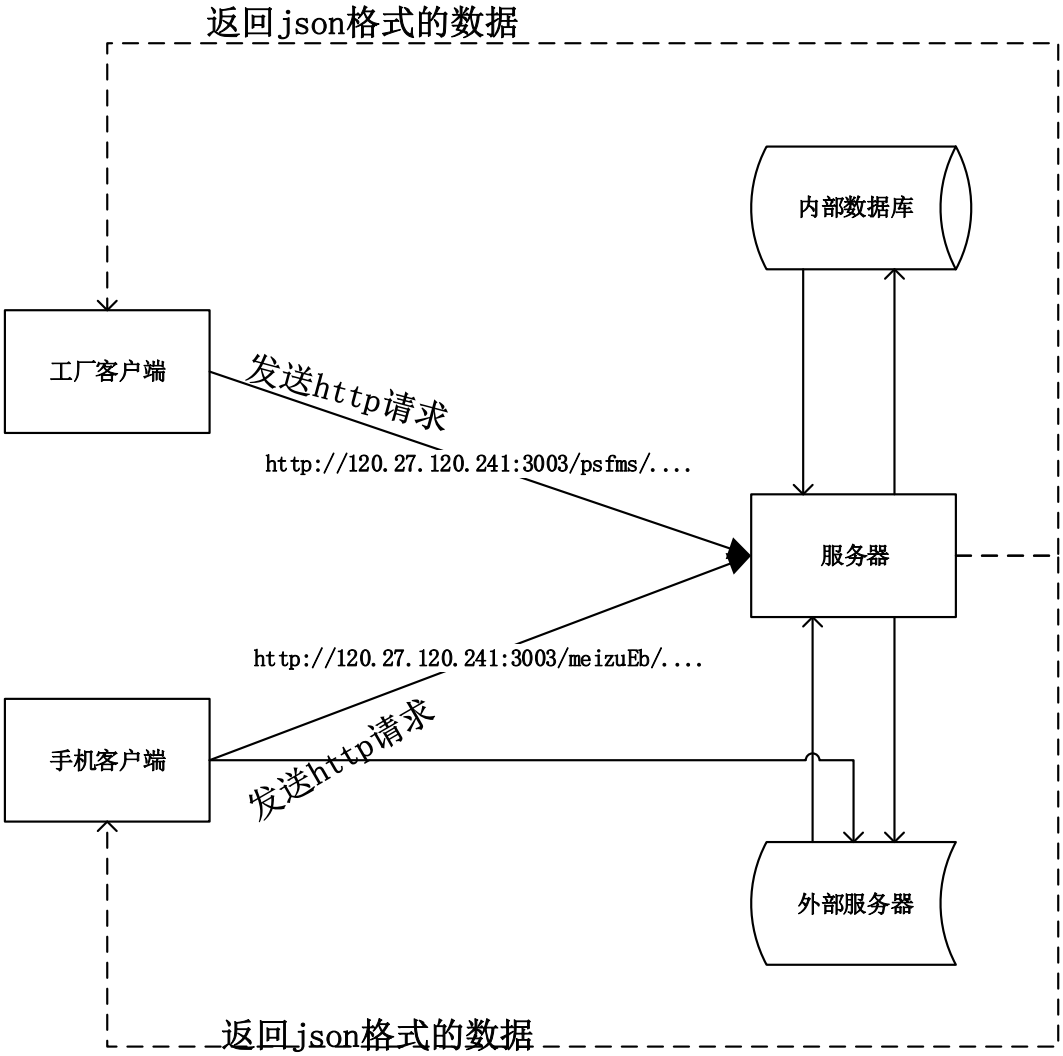

图 3-2 手机壳定制系统的业务流程图

手机壳定制系统由三个部分组成，分别是“魅印”手机客户端、工厂客户端和后台服务器。“魅印”手机客户端与工厂客户端与服务器建立连接，发送 http 的 get 或 post 请求提交数据或者获取显示数据。当发送 get 命令时，服务器与内部数据库进行交互，将数据库返回的数据已 Json 的格式发回到客户端，客户端取到数据后使用 Gson 转化为对象，作为数据源初始化或者更新控件内容。在支付部分，手机客户端还需要与外部服务器进行交互，外部服务器与内部服务器进行数据交换，最终由内部服务器返回支付的结果给手机客户端。

手机客户端以 URI 为 http://localhost:8080/GetOrderById?orderId=144879193185 发送请求，服务器返回一个 Json 串 {id: “123”, name = “hello”} ，最后客户端根据服务器返回的结果初始化或更新控件。

3.2 系统手机客户端功能模块划分

3.2.1 定制模块

1. 模块用例图

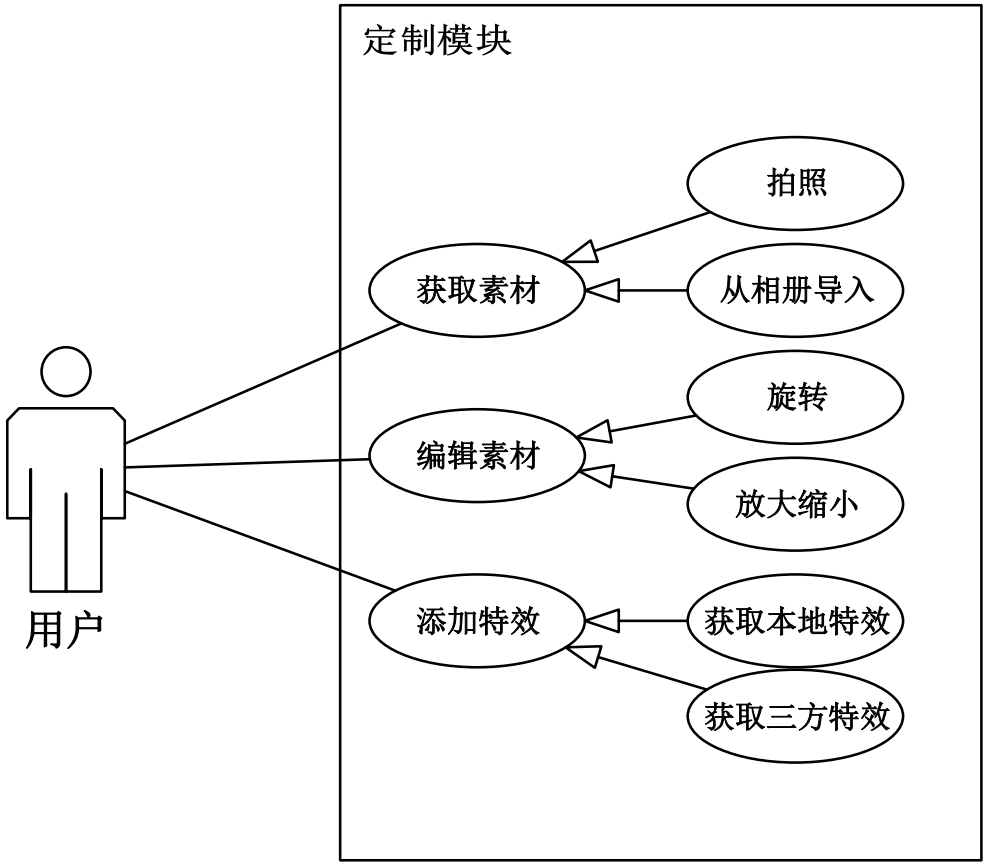

图 3-3 定制模块用例图

2. 模块用例描述

● 获取素材用例

1) 用户选择手机型号，系统给予两种获取素材的方式——拍照和从相册导入。

2) 用户选择拍照，系统调用照相机，并将最新拍的照片渲染在照片编辑区域。

3) 用户选择从相册导入，系统显示所有相册列表，点击某个相册，以缩略图的形式显示所有的照片。

4) 用户点击想要编辑的照片，系统将所选照片渲染在照片编辑区域。

● 编辑素材用例

1) 用户在照片编辑区域可以双手缩放，来改变照片的大小，系统自动剪裁区域之外的部分。

2) 用户在照片编辑区域可以通过旋转，来改变照片的方位，系统自动剪裁旋转过程中出界的部分。

● 添加特效用例

1) 用户点击特效按钮，可以选择加入本地特效，其中包含常规的滤镜、添加文字特效。

2) 除了本地特效，用户可以选择加入第三方特效，目前系统接入 360 Camera 的特效素材，如气泡，卡通图案，动态图片。

3) 编辑完成后，用户点击确认，系统将展示最后的效果，并计算图片质量水平，生成价格和待发服务器的图片。

3.2.2 用户管理模块

1. 模块用例图

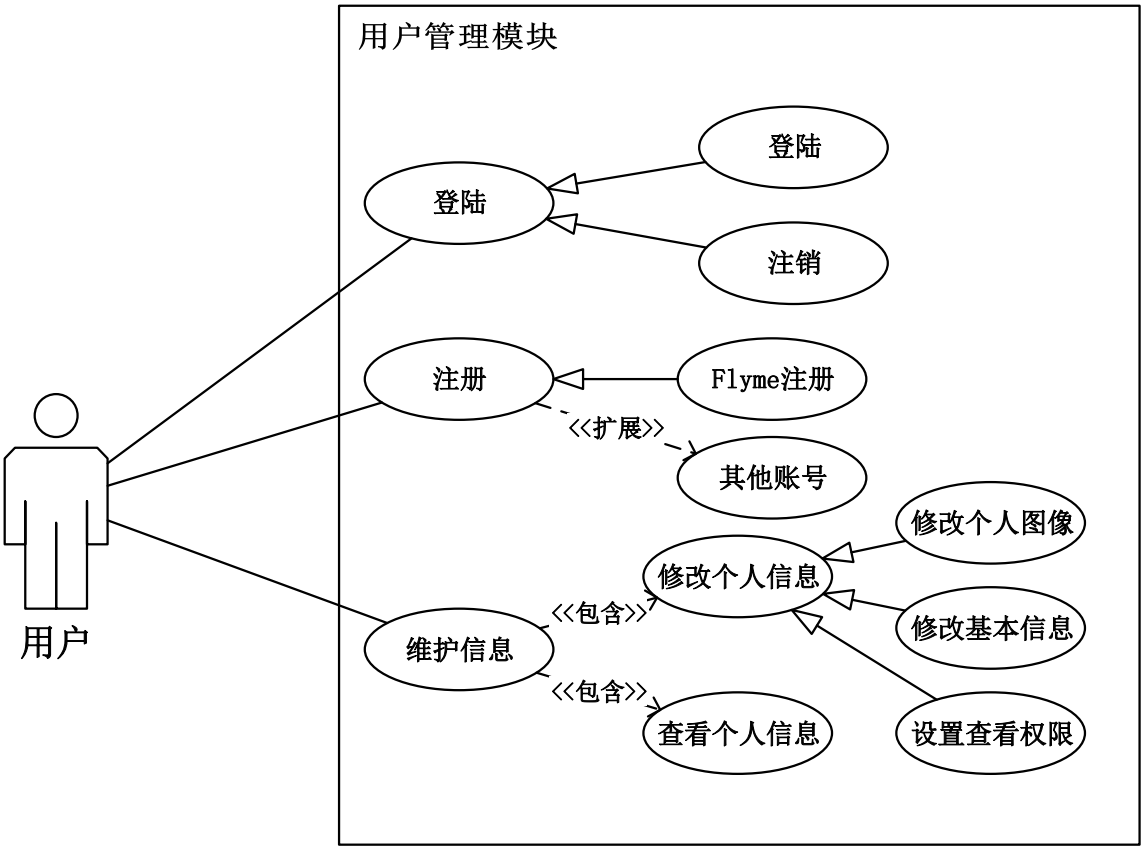

图 3-4 用户管理模块用例图

2. 模块用例描述

● 登陆用例

1) 用户在个人界面上点击登陆，系统跳转到 flyme 登陆页面，输入 flyme 的账号和密码进行登陆。
2) 用户登陆之后，可以在我的界面上点击注销，系统将删除用户的相关信息，并通知服务器。

● 注册用例

1) 用户进入 flyme 登陆页面，如果没有注册，可以点击注册，页面将跳转到 flyme 的注册页面。
2) 用户可以选择通过手机号注册，flyme 将会发送验证码到用户手机，验证成功后，用户输入两次密码，即完成注册。
3) 用户可以选择通过第三方完成注册，目前可选的有 QQ、WeChat 注册，系统跳转到相应应用的授权界面，并自动给用户分配账号。
4) 注册完毕，用户重新进入 flyme 登陆界面进行登陆。

● 修改个人基本信息用例

1) 用户的个人信息包括用户图片、用户昵称、生日、手机号，刚注册的用户需要完善以上信息。
2) 用户可以修改个人基本信息，并设置查看权限，其中包括允许所有人查看、仅自己可见、仅朋友可见。

● 查看个人基本信息用例

1) 用户进入我的界面，可以查看自己的基本个人信息，包括用户图片、昵称、生日和手机号。
2) 没有完善以上信息的用户，系统将提示用户填写信息，方便其他用户找到你。

3.2.3 订单管理模块

1. 模块用例图

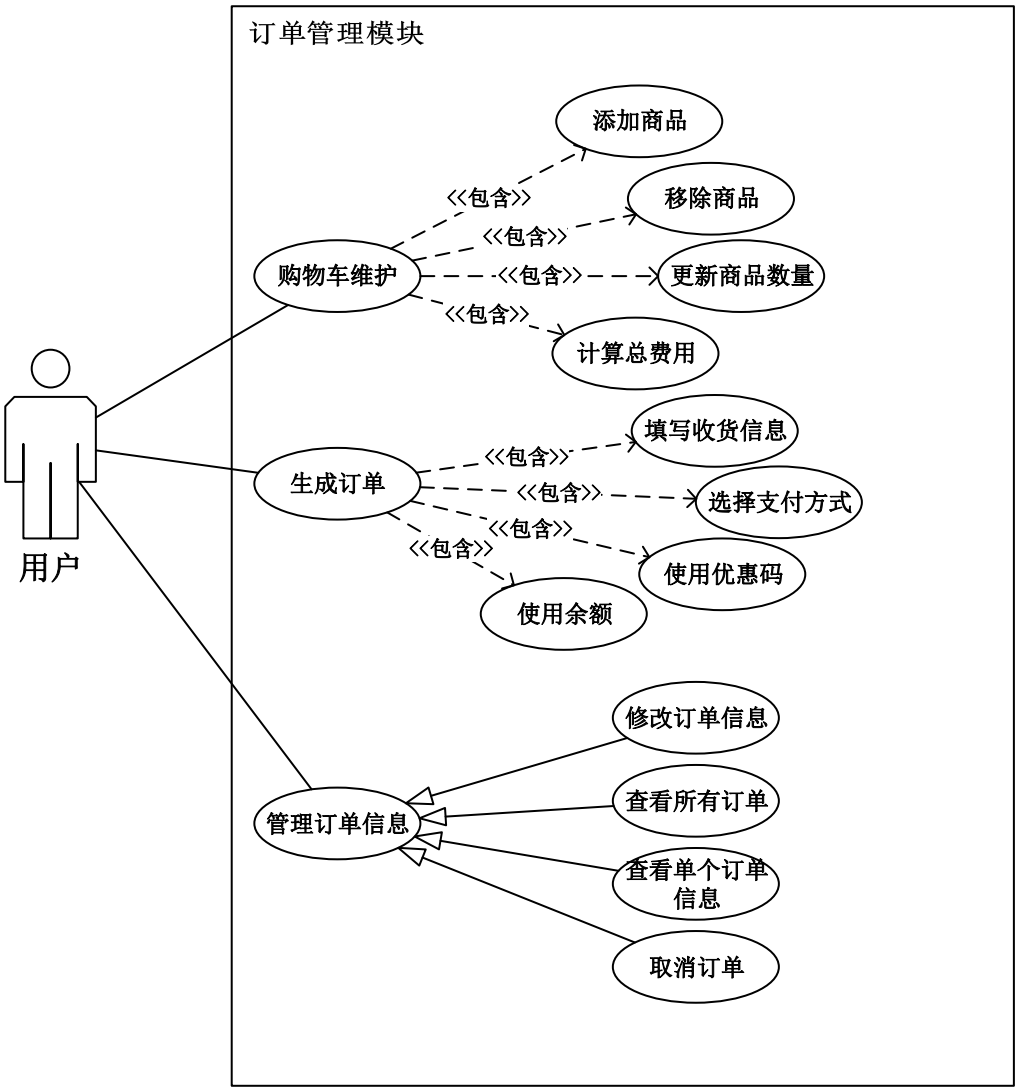

图 3-5 订单管理模块用例图

2. 模块用例描述

● 购物车维护用例

1) 用户可以再次制作手机壳，将新的手机壳添加到购物车列表。

2) 用户在购物车界面可以点击删除按钮，取消某个商品。

3) 用户可以上调或下调某一个商品的数量，系统默认商品的数量为 1。

4) 系统在用户编辑购物车的时，会实时计算所有商品的总费用。

● 生成订单用例

1) 用户确定好所有商品后，点击购买按钮，系统将跳转到订单生产界面。

2) 对于初次下单的用户，需要用户填写收货信息，其中包括用户姓名、电话、手机号、收货地址、备注。信息录入成功后，系统将本地保存用户的地址信息，便于下次购买。

3) 已录入收货信息的用户，系统会根据用户使用地址的频度，以首选项的方式默认一个地址，用户可以进入地址编辑界面，选择其他录入的地址。

4) 用户需要选择支付方式，目前仅支持在线支付，包括支付宝和微信支付。

5) 系统会定期给购买达到一定次数和金额的用户发送优惠码，用户可以输入优惠码，这将在一定程度上减免部分费用。

6) 用户可以使用余额的方式支付，余额可以通过充值。

7) 用户完善所有信息之后，即可下单，系统将会把订单信息发送给服务器，同时更新用户的订单列表。

● 修改订单信息用例

1) 在我的界面，系统将列出所有的订单，按照时间的先后顺序进行展示。

2) 目前系统无法支持用户修改订单。订单一旦生成，将无法再次编辑订单的信息。

● 查看所有订单用例

1) 在我的界面，系统将列出所有的订单，按照时间的先后顺序进行展示。

2) 用户可以通过下拉和下滑，查看所有的订单，系统会根据用户的动作更新订单列表和加载更多。

● 查看单个订单用例

1) 在我的界面，系统将列出所有的订单，按照时间的先后顺序进行展示。

2) 用户可以点击单个订单的 item 项，查看具体的订单信息和物流信息。

3) 系统将会在每次加载时，更新订单状态和物流信息。

● 取消订单用例

1) 对于已支付的订单，系统目前不支持取消订单功能，将自动隐去删除订单的按钮。

2) 对于未支付的订单，系统检查订单如果在五分钟内没有支付成功，将会自动删除。

3) 对于未支付的订单，用户可以点击删除订单按钮来取消订单。

3.2.4 支付模块

1. 模块用例图

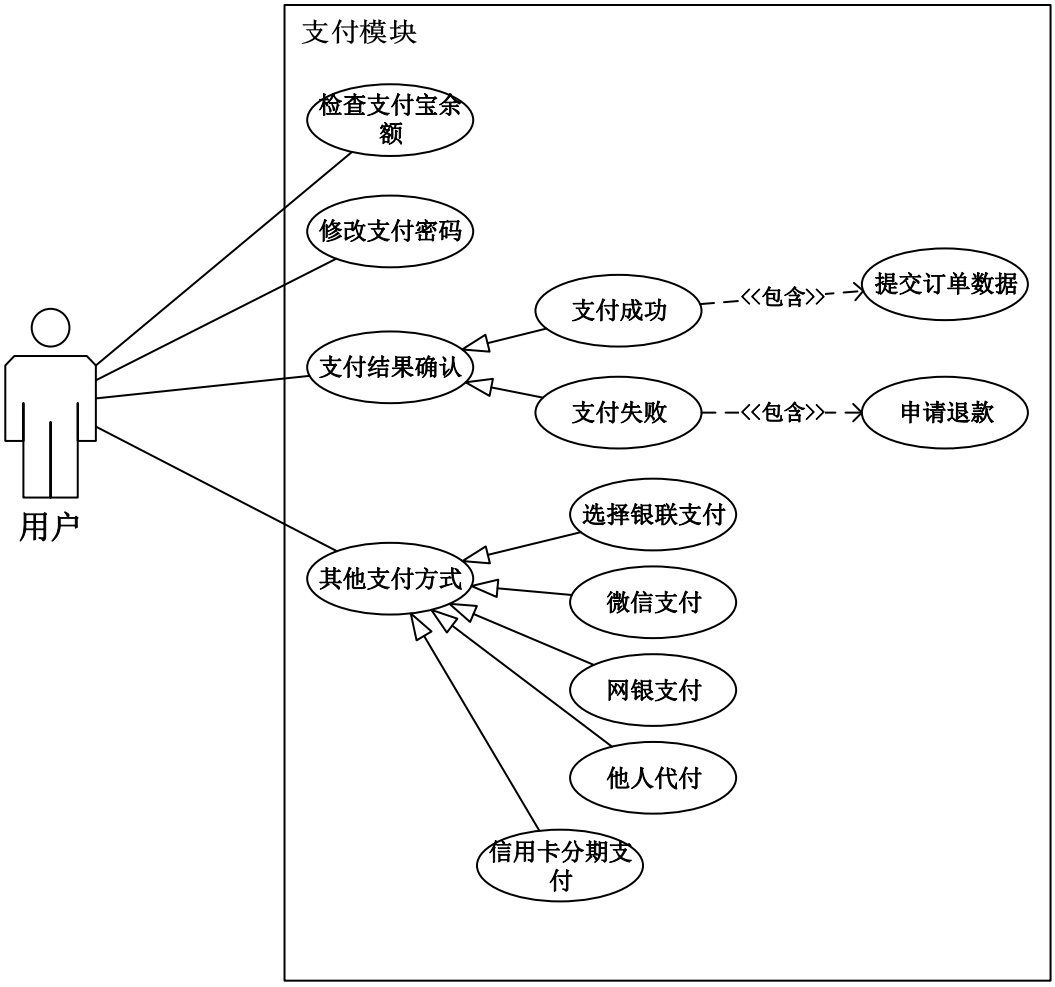

图 3-6 支付模块用例图

2. 模块用例描述

● 检查余额用例

1) 在支付状态下，系统会与支付宝交互，获取用户支付宝的余额，判断是否可以直接通过余额进行支付。

2) 如果无法通过余额进行支付，系统将会自动隐去余额支付该项。

● 修改支付密码用例

1) 用户需要先输入旧的支付密码通过验证。

2) 验证通过后，需要输入新密码两次，系统自动提交新密码。

● 确认支付结果用例

1) 用户完成支付后，系统将检查支付结果，等待服务器返回支付结果码。

2) 如果支付成功，将提交订单数据到服务器。

3) 如果支付失败，将提示用户支付失败，并请求用户尝试重新支付。

● 选择其他支付方式用例

1) 用户可以在支付状态下，选择其他支付方式，其中包含银联支付、微信支付、网银支付、他人代付。

2) 对于信用卡分期支付，购买金额需要达到一定金额，否则无法分期支付。

3.2.5 社区分享模块

1. 模块用例图

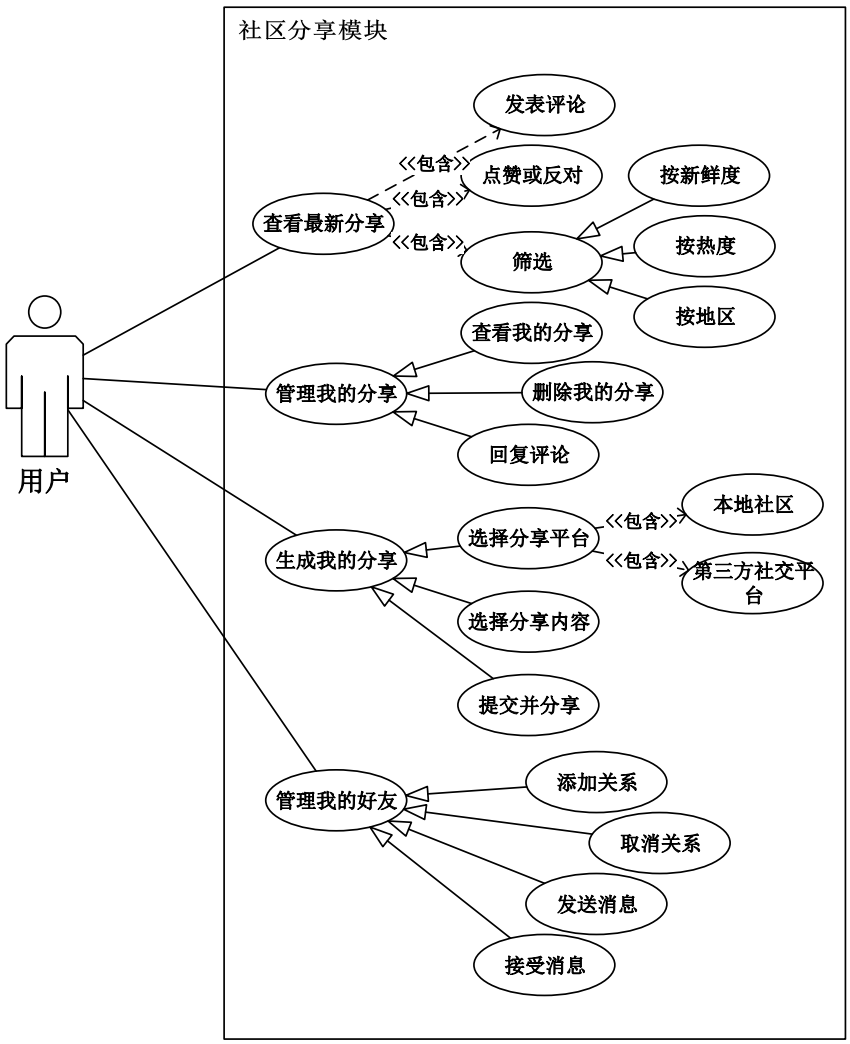

图 3-7 社区分享模块用例图

2. 模块用例描述

● 查看最新分享用例

1) 用户进入社区分享界面，可以以三种方式进行查看所有用户的分享，分别是按新鲜度、按热度和按地区。

2) 在其他用户的分享下，用户可以发表文字评论，可以点赞和反对。

3) 可以点击某个用户的分享，查看该分享的所有评论和其他信息。

● 管理我的分享用例

1) 在社区界面的我的分享分页下，用户可以查看自己的所有分享。

2) 同时用户可以删除自己的分享，系统将会同步服务器的相关数据。

3) 用户可以查看其他用户的评论，并进行回复。系统会在收到评论时以气泡的方式提示用户。

● 生成我的分享用例

1) 用户购买成功后，可以分享自己的作品，系统提供两种分享方式，包括本地社区和第三方社交平台。

2) 用户选择本地社区，填写分享内容，系统自动置入手机壳图片，点击确认分享即可。

3) 用户选择第三方社交平台，目前系统支持分享以下社交平台，包括 QQ 空间、微信朋友圈、新浪微博、豆瓣和花瓣。

● 管理我的好友用例

1) 用户可以搜索其他用户，与之建立关系。

2) 用户可以删除已建立的关系。

3) 用户可以向建立关系的朋友发送消息和接受该朋友的回复。

4) 系统自动将建立关系朋友的分享推送给用户，并以气泡的方式通知用户。

3.3 系统工厂客户端功能模块划分

3.3.1 订单管理模块

1. 模块用例图

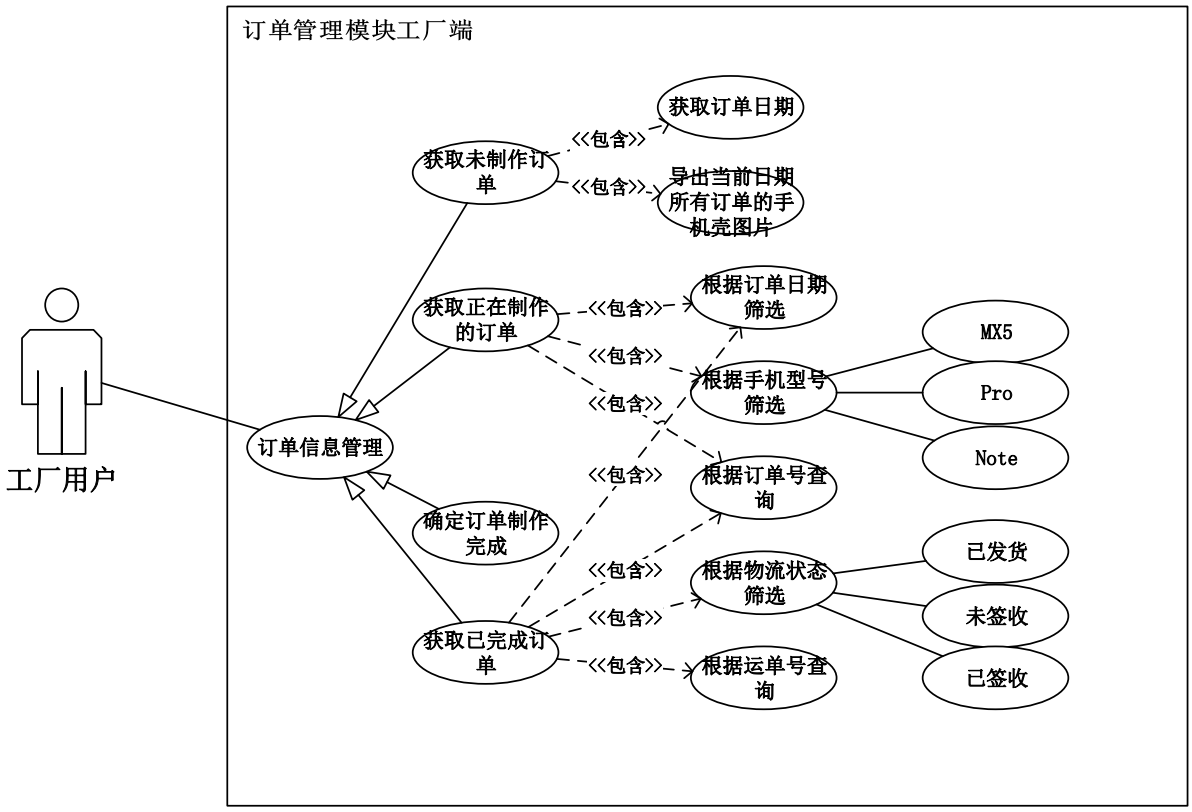

图 3-8 订单管理模块用例图

2. 模块用例描述

● 获取未制作订单用例

1) 工厂用户进入未制作订单页面，系统获取所有的未制作订单日期供用户选择。

2) 工厂用户选择某一日期，系统加载当日所有的未制作订单的手机壳图片，以列表的形式展现给用户。

3) 工厂用户点击导出手机壳图片，系统从服务器取到当日所有订单，并统一按订单号+序号命名图片文件，最终打包发送给前端页面。

4) 用户确定导出成功，系统将所选日期和对应的所有订单状态改为正在制作。

● 获取正在制作的订单用例

1) 工厂用户进入正在制作订单页面，系统获取所有正在制作订单的信息，以列表的形式展现给用户。

2) 工厂用户可以通过订单日期对正在制作的订单进行筛选，也可以通过手机型号进行筛选，如果有单独的订单号，可以通过订单号查询某个订单的相关信息。

● 确定订单制作完成用例

1) 工厂用户对导出的订单图片进行制作，完成后即可开始进行发货。

● 获取已完成订单用例

1) 工厂用户进入已完成订单页面，系统获取所有已完成的订单信息，已列表的形式展现给用户。

2) 工厂用户可以通过订单日期对已完成的订单进行筛选，也可以通过订单的物流状态进行筛选。

3) 如果工厂用户知道订单号或者订单的运单号，可以通过订单号和运单号进行信息查询，获取该订单的详细信息。

（注：手机型号目前包含三种，分别是 MX5，Pro 和 Note。物流状态包括已发货，未签收和已签收）

3.3.2 用户权限管理模块

1. 模块用例图

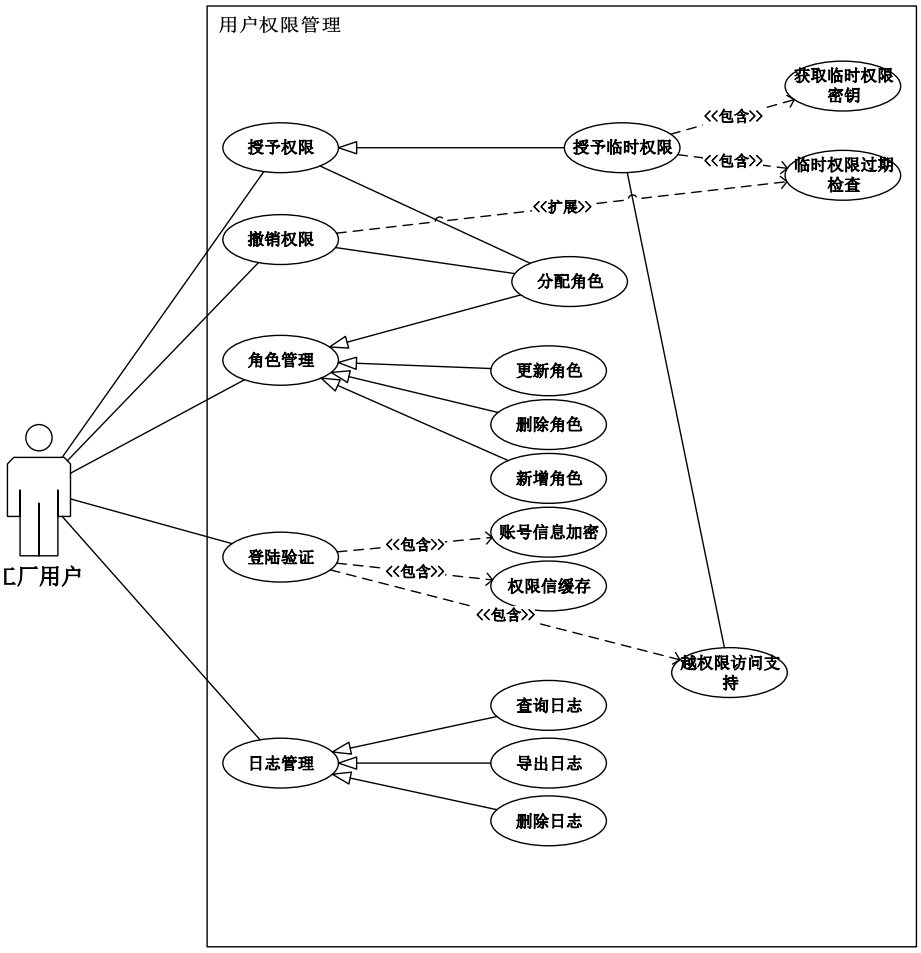

图 3-9 用户权限管理模块用例图

2. 模块用例描述

● 授予权限用例

1) 管理员用户登陆系统，可以给所有工厂端用户进行权限授予。

2) 管理员用户可以给其他用户指定相应的角色，不同的角色职能不一样，对系统数据获取的权限也不一样。

3) 系统支持授予临时权限，即“Run as”，其他用户必须获取对应用户的临时权限密钥，才能在一定时间段内以该用户的身份进行数据访问。同时系统会检查临时权限是否过期，如过期，将撤销该临时权限，并将当前用户的权限切换为原始设定。

● 撤销权限用例

1) 管理员用户可以撤销其他用户的角色，让该用户失去当前角色的职能和数据访问权限。

2) 对于临时权限过期的问题，系统将会自动撤销用户的临时权限，并将当前用户的权限切换为原始设定。

● 角色管理用例

1) 如工厂用户有权限进行角色管理，该用户可以新建角色，给改角色设定数据访问和功能职责范围。

2) 该用户可以删除某个角色，系统将会通知服务器移除该角色相关信息。

3) 同时该用户可以更新当前已有的角色内容，如变更数据访问或者功能职责的范围。

● 登陆验证用例

1) 为了系统的安全，所有用户的账号信息都将使用 MD5 加密方式进行存储。

2) 系统会保存当前用户的权限，避免每次访问时重复查询，提高操作的流畅度。

3) 系统支持越权限访问，即“Run as”，临时权限的授予与撤销有系统自动维护和管理。

● 日志管理用例

1) 类似于数据库的审计功能，所有用户的操作都将记录到日志中，包括时间，操作内容，访问数据和变更数据。

2) 有权限的用户可以查看日志内容，或者删除日志。

3) 系统支持将日志已 Excel 格式导出。

3.3.3 数据统计模块

1. 模块用例图

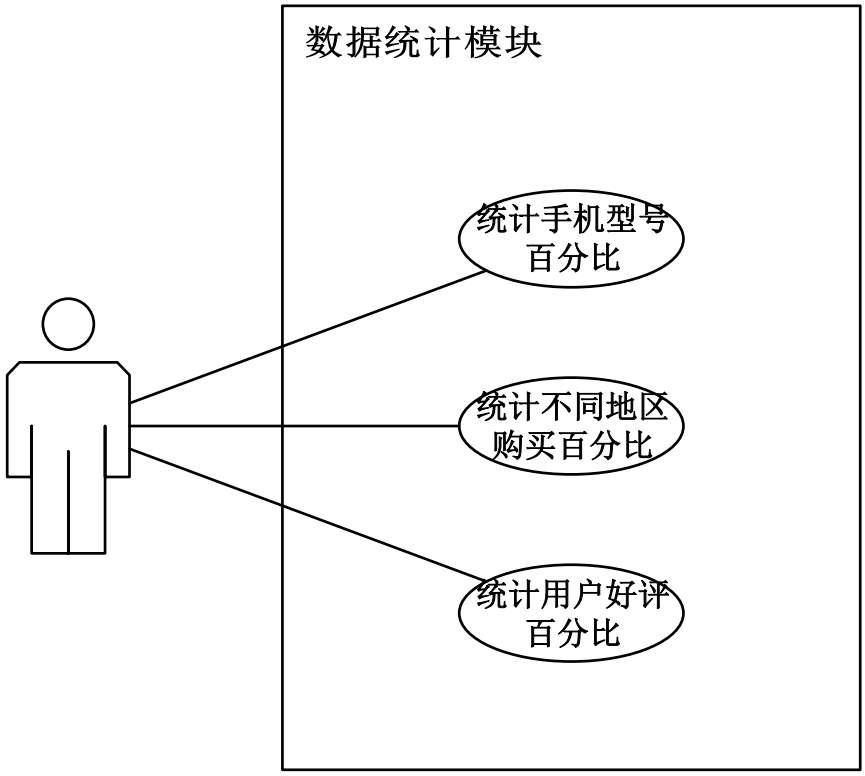

图 3-10 数据统计模块用例图

2. 模块用例描述

● 统计手机型号百分比用例

1) 系统在数据统计页面中，获取数据库的销售的数据信息，按照以下三个时间段进行统计，当月、当季和当年。在以手机型号为指标的统计中，分别计算当前时间段内不同手机型号的销售情况，计算出百分比，并以饼状图的形式和表格的形式展示数据。

2) 用户切换不同的时间周期，系统刷新数据，显示当前时间周期的结果。

● 统计不同地区购买百分比用例

1) 系统在数据统计页面中，获取数据库的销售的数据信息，按照以下三个时间段进行统计，当月、当季和当年。在以地区为指标的统计中，分别计算当前时间段内不同地区的手机壳销售情况，并以条状图的形式和表格展示数据。

2) 用户切换不同的时间周期，系统刷新数据，显示当前时间周期的结果。

● 统计用户好评百分比用例

1) 系统在数据统计页面中，获取数据库的销售数据信息，按照以下三个时间段进行统计，当月、当季和当年。在以用户好评率为指标的统计中，分别计算当前时间段用户的好评和差评比例，并以饼状图和表格的形式展示数据。

2) 用户切换不同的时间周期，系统刷新数据，显示当前时间周期的结果。

3.3.4 物流管理模块

1. 模块用例图

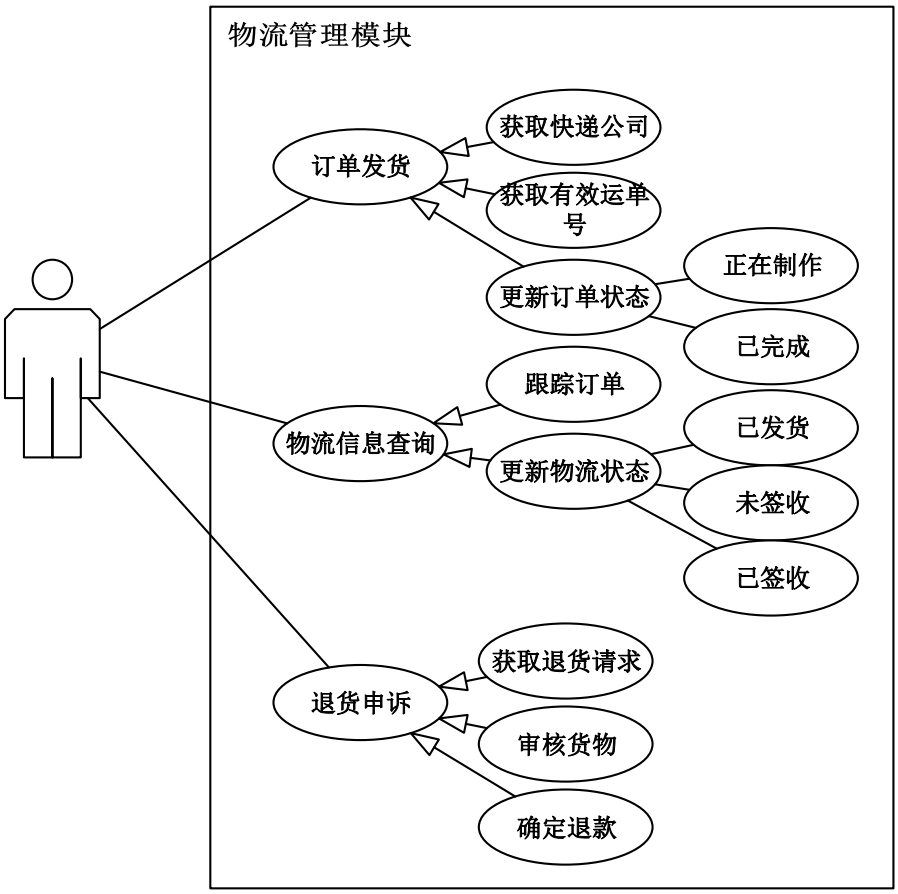

图 3-11 物流管理模块用例图

2. 模块用例描述

● 订单发货用例

1) 工厂用户确定订单制作完成之后，可以开始发货。

2) 发货时，系统获取所有可选的快递公司，并为该订单生成唯一的运单号。工厂用户确认信息后，点击确认发货。

3) 确认发货后，系统将当前订单的状态由正在制作变更为已完成。

● 物流信息查询用例

1) 工厂用户可以实时查询订单的物流状态信息，跟踪物流的位置信息。

2) 系统通过第三方物流 app 获取订单的进展状况，默认每 3 小时更新一次物流信息。同时变更订单的物流状态，由已发货变为未签收，然后由未签收变为已签收，至此，订单完成。

● 退货申述用例

1) 系统收到退货请求，会以气泡的形式通知有权限的用户接受查看。

2) 工厂收到退返的货物之后，检测有问题的，将通过系统通知用户退货成功。

3) 对于退货成功的情况，工厂将发送确定退款通知。

3.4 后端服务器接口划分与介绍

3.4.1 手机端接口

后端服务器定义的手机端接口是供定制系统手机客户端进行访问获取信息使用的。手机端通过 HttpClient 发送 GET 或 POST 请求，然后服务器与数据库进行交互返回信息给前端。表 3-1 统计了系统定义的所有手机端的接口。

表 3-1 手机端接口列表

接口名	请求类型	描述
POSTORDER	POST	上传订单接口
GETORDERS	GET	获取一个用户的所有订单接口
GETORDERBYID	GET	通过订单号获取订单信息接口
GETORDERBYTRACKINGNO	GET	通过运单号获取订单信息接口
GETORDERBYDATEINTERVAL	GET	获取指定日期段内所有的订单信息接口
GETORDERBYDATE	GET	获取指定日期所有订单信息接口
NOTIFYORDERPAID	GET	通知订单付款成功接口
NOTIFYORDERSHIPPED	GET	通知订单已发货接口
PRODUCTDEF	POST	产品信息的添加/修改/获取接口
ALIPAYNOTIFY	GET	服务器异步信息保存接口

3.4.2 工厂端接口

后端服务器定义的工厂端接口是供定制系统工厂客户端获取信息使用的。工厂前端通过 Ajax 发送 GET 或 POST 请求，然后服务器与数据库进行交互返回信息给前端。表 3-2 统计了所有工厂端接口。

表 3-2 工厂端接口列表

接口名	请求类型	描述
GETPHONESHELLORDERSSTATE	GET	获取手机壳订单统计信息接口
GETPHONESHELLNONPRODUCE	GET	获取未制作手机壳订单接口
GETPHONESHELLPRODUCING	GET	获取正在制作手机壳订单接口
GETPHONESHELLPRODUCED	GET	获取已制作手机壳订单接口
POSTPHONESHELLPIC	POST	上传手机壳定制图接口
GETPHONESHELL	GET	获取手机壳定制图
PHONESHELLORDEREXPORT	GET	批量获取手机壳定制图
NOTIFYPHONESHELLORDEREXP	GET	通知手机壳订单已导出
PHONESHELLORDEREXPORTAV	GET	查询指定条件的日期判断订单是否能够导出
STATISTICSBYCONDITION	POST	大数据统计相关接口
RIGHTS	GET	用户权限管理接口

3.5 质量需求分析

3.5.1 性能

- 系统各功能运行结果与预期一致
- 可与任何有网页的机器共同操作。
- 可在各种移动设备及系统上打开网页操作。
- 系统通过网关，基站设备与用户进行连接，同一个用户只能占用同一个网关，一个网关最多连接 10 个用户，一个用户可以与多个基站设备相连，一个基站设备最多连接 500 个用户。
- 系统网关受限于请求吞吐量，每秒不多于 100 个用户请求。

3.5.2 安全性

- 用户未登录，系统拒绝一切操作。
- 系统拒绝一切该角色没有的权限。
- 至少 99.9%以上的时间，系统可以保护用户之间的消息不被非法授权增加、修改或删除。
- 当受到病毒攻击时，系统能够检测到攻击操作并立即停止服务器运行。
- 应用程序必须扫描所有进入的或更改的数据及软件，以发现所有被公布的计算机病毒，蠕虫以及木马。
- 当黑客入侵时，系统能够检测到伪造数据的更改。

3.5.3 可维护性

- 分析普通应用故障的平均工作量小于 1 人/天
- 分析基站设备损坏导致系统紊乱的问题小于 1 人/天
- 分析升级添加新功能需要更改的模块时间小于 1 人/周
- 分析应用瘫痪或由于过多繁琐故障导致的系统破坏问题小于 1 人/周
- 完成一次数据恢复的平均工作量小于 1 人/天
- 完成一次小版本升级的平均工作量小于 1 人/周
- 完成一次重大版本升级的工作量小于 1 人/月
- 修复问题的平均工作量小于 1 人/周
- 系统可删除不要的功能，优化或简化现有系统功能
- 系统可添加新的功能，改进已有的功能或修复系统中的缺陷

3.5.4 可靠性

- 服务器正常运行的可能性>99.99%
- 服务器平均正常运行时间>1 年
- 服务器平均故障恢复时间<1 小时
- 重大故障在 12 小时内回复服务的可能性，在 24 小时内回复历史数据

- 系统可进行数据备份，固定时间进行一次完全备份

3.5.5 易用性

- 专业人员透彻操作系统所需时间 2H
- 管理员操作熟练时间 1H
- 外部用户操作熟练时间 1H
- 专业人员学习该软件所需时间 20MIN
- 有软件基础的人员学习该软件所需时间 30MIN
- 无软件基础学习软件所需时间 50MIN
- 专业人员理解系统所需时间 1DAY

3.5.6 效率

- 点击响应时间<2 秒
- 注册响应时间<5 秒
- 查询响应时间<5 秒
- 用户注册频率每分钟 10 次
- 用户更新信息频率每小时一次
- 用户查询信息频率每分钟一次
- 系统用户数≤1,000,000
- 同时在线人数≤1,000
- 信息传递数≤1,000,000,000

4 系统架构设计

4.1 设计约束条件

4.1.1 设计的标准和规范

- 模块化、松耦合。
- 面向接口，机制与策略分离。
- 插件式结构。

4.1.2 软硬件约束

- 手机客户端：Android OS 4.0 及以上
- 工厂客户端：IE、Chrome、Firefox、Safari 等常用浏览器。
- 系统使用 MySQL 5.0 数据库。
- 服务器 Linux + jetty + MySQL 5.0

4.1.3 运行环境约束

- 手机客户端：需要用户在联网状态下使用，可以是 Wi-Fi 网络或者 2G 及以上网络。
- 工厂客户端:推荐使用Chrome浏览器进行操作,比较证明该浏览器兼容性更好。

4.2 项目性能规范

4.2.1 图片渲染性能

对于“魅印”手机壳定制客户端，其最重要的特点是“定制”，它在基于 Android 自带的对图片的处理方法外，加入第三方素材，如水印效果，3D 效果等，都是为了满足用户在编辑图片是需要的多种个性化需求。所以这就对系统的图片渲染功能提出了一定的要求，在图片处理和发送最终待制作的图片到服务器的过程中，要尽可能的保留图片的

像素密度和像素质量，并按照不同手机型号的规格确定最终照片的尺寸大小。

对于用户导入的图片，能够自动优化清晰度，锐度，亮度等参数的，要自动帮助用户去完成，并实时的检测图片的质量，如果图片质量过于低，系统要提示用户更换图片。总的来说，系统提高图片渲染性能要做到如下几点：

- 兼容各种第三方特效，如水印，3D 等，并保证不与系统自带特效冲突。
- 严格控制图片的质量，对于在可调整范围的图片，由系统自动优化。
- 对于优化后图片质量仍然没有改变的，提示用户替换当前编辑的图片。
- 系统根据手机型号规格，确保最后发送的图片像素密度和像素质量在能够正常制作的范围内。

4.2.2 信息处理性能

对于“魅印”手机壳定制客户端，同时也要强调信息处理的性能，保证用户提交和获取的信息效率。用户在很多应用场景下都需要通过网络与服务器进行数据的交换，如果一次性同步提交很多数据，肯定会导致上次提交的数据或获取的数据还没有取到，新的提交或获取的数据只能在消息队列里等待上一次的操作完成。这样的实现方式最终造成信息更新瘫痪，应用无法快速响应用户的请求。显然这样的结果是可怕的。为了提高系统信息处理的性能，需要做到以下几点：

- 信息的提交使用异步提交数据，系统预先改变客户端的结果，待提交的数据进行异步提交。为防止用户在系统提交数据的过程中终止进程，系统需要动态的提示用户后台的同步操作，如果数据提交失败，提示用户失败。
- 消息的获取使用同步获取数据，即用户触发了获取订单信息的操作，系统使用同步的方式即时与服务器交互，将最新的数据展现给用户。
- 在同步与异步的互相配合下，即降低了网络压力，又能够在用户容忍的限度内将数据显示。

4.2.3 消息通知性能

对于“魅印”手机壳定制客户端，都需要及时的将最新消息推送给用户，如用户的共享有了最新的评论，工厂端收到了退货通知。仅仅靠人为的去获取、刷新，肯定会造成

延误。为了让用户能够及时的收到最新消息，手机客户端和网页端都必须考虑消息通知的性能。为了提高该性能，需要做到以下几点::

- 当有最新消息时，系统自己唤起，并推送给用户，不管用户是否启动了应用程序，用户可以选择关闭推送最新消息。
- 系统以醒目的方式提示用户，在手机客户端，消息推送以 Notification 的方式；在工厂客户端，消息以气泡的方式。

4.2.4 服务器并发处理性能

对于服务器，当多用户并发访问时，要考虑提高其并发处理的能力。避免造成服务器长时间无法访问或服务器崩溃。

使用分布式服务器和数据库技术，搭载和处理后端服务器的业务流程，进而提高服务器并发处理的能力[9]。

4.3 设计模式

引入设计模式这一概念，是为了提高代码的复用性，降低不同模块的耦合度，在一定程度上实现高内聚，低耦合。

首先系统使用了 MVC 的架构，将 Model，View 和 Controller 彼此分离开来，也就是将数据，视图和后台的业务处理逻辑分离。这样做的好处是，不同的部分可以单独实现，需要耦合的地方以接口的形式给出。

在具体的模块实现里，主要使用了如下的设计模式来提高代码的质量，为后期需求的变更和代码复用做准备。

4.3.1 观察者模式

社区分享模块使用了观察者模式，用户添加朋友关系和取消朋友关系与模式中的订阅与取消类似，当被朋友的状态更新时，消息将自动发送给好友。模式的结构图见图 4-1。

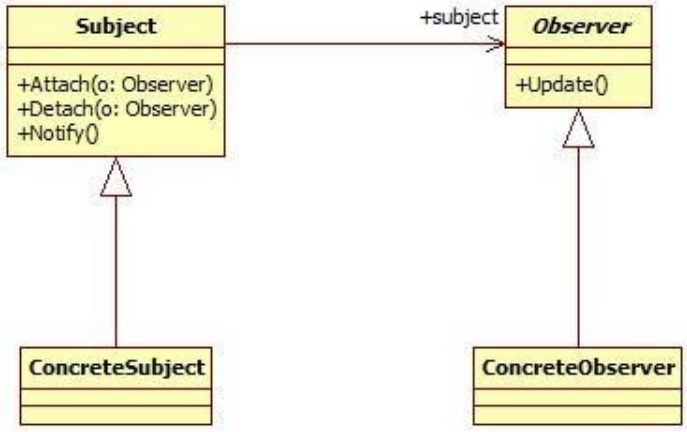

图 4-1 社区分享模块使用观察者模式的结构图

4.3.2 适配器模式

订单管理模块使用了适配器模式，Android 很多控件的数据绑定都是继承 Adapter 来实现的，通过自定义 Adapter 的方法，来显示数据和更新数据。手机壳定制客户端的订单处理的数据源来着内部服务器返回的 Json 串，通过使用 Gson 的 fromJson 方法，转换为 Model 中对应的对象。然后重写自定义 Adapter 的 getCount、getItem 和 getView 方法。模式实现的结构见图 4-2。

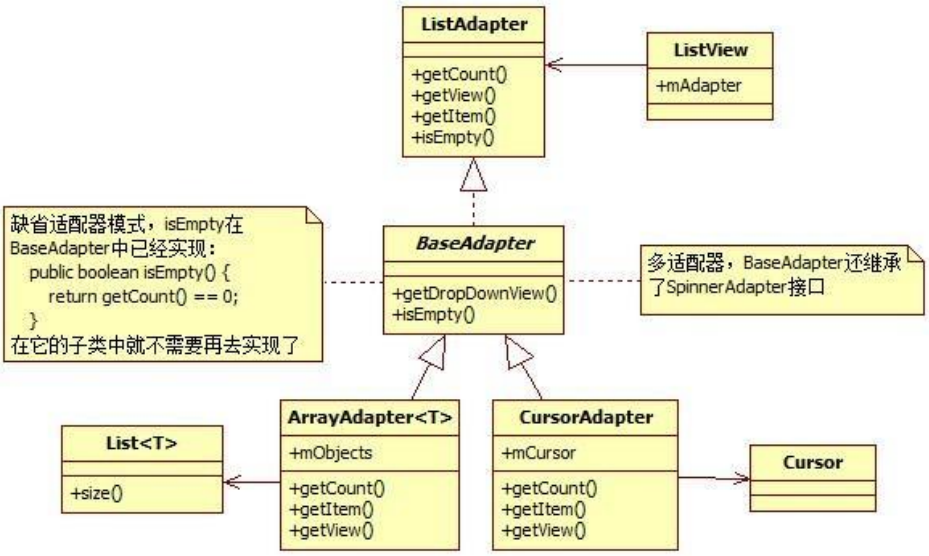

图 4-2 订单管理模块使用适配器模式

4.4 系统结构总体设计

手机壳定制系统包含三个部分，分别是工厂客户端、手机客户端和服务器。

4.4.1 数据交互设计

三部分的数据交互方式见图 4-3。

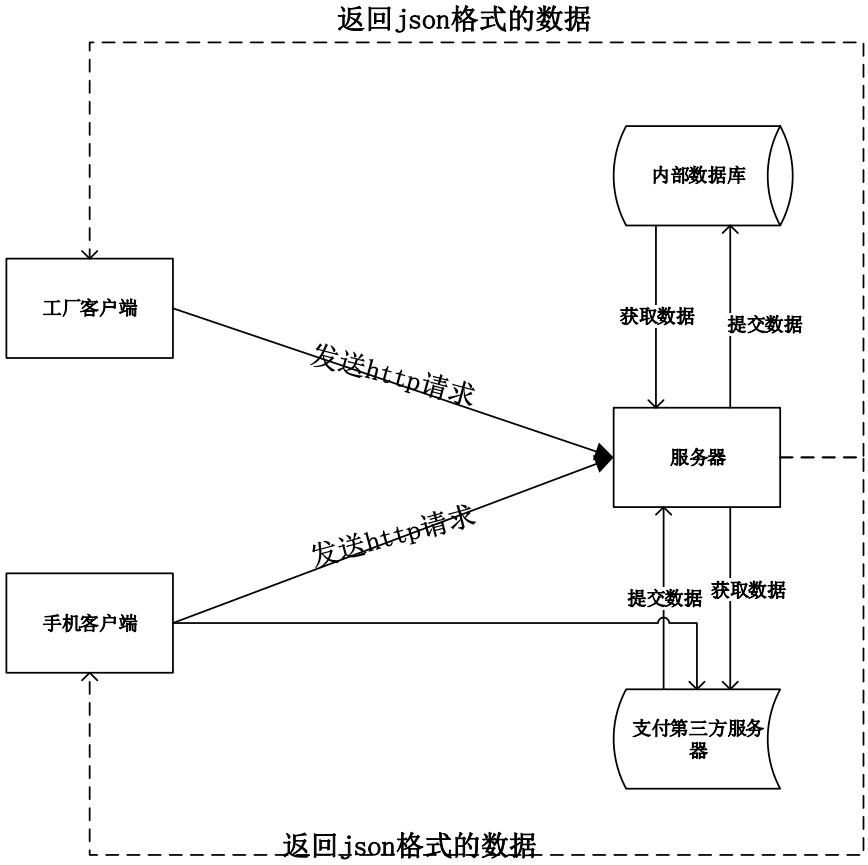

图 4-3 客户端与服务器数据交互方式

工厂客户端和手机客户端向服务器以异步或同步的方式，发送 HTTP 的 GET 或 POST 请求，然后将服务器返回的 Json 串进行解析。

4.4.2 整体架构设计

系统的总体架构可以分为数据显示层，业务操作层和网络交互层。这三个层次是基于 MVC 架构划分的，为系统的运行提供了一个结构清晰的运行环境。如将网络交互层单独划分为一个层次，是因为在 Android 客户端，需要与网络打交道的地方太多，如

果只是简单地与业务逻辑层耦合在一起，不仅结构混乱，而且后期网络发生变化时，很难迅速修改和定位错误。所以通过封装一个 Data Retriever 层，很好的解决的这个问题，并可以灵活定义不同业务操作的网络请求方式，如同步或是异步，提交还是获取（即 POST 还是 GET）。对于数据显示层，不同的客户端代表的内容不一样，Android 客户端的数据显示层主要是活动（activity）的定义，而工厂后台系统是 jsp 页面的实现，以及与业务操作层的相关数据绑定部分。业务操作层，顾名思义，就是对系统内部业务流程的定义与实现，通过操作获取的数据，将处理后的数据与相应的数据显示层进行绑定[10]。

手机壳定制系统的服务器使用 Servlet 作为业务处理层，使用 Hibernate 作为数据持久层[15]。总的系统结构图见 4-4。

图 4-4 系统整体结构图

在系统整体架构设计图中，数据显示层包含客户端需要与用户交互的所有 UI，其

中手机客户端的 UI 是 Activity，如图片编辑 UI、支付 UI、社区分享 UI 等，而工厂客户端的 UI 是 html 或 jsp 页面，如数据统计 UI、物流管理 UI 等。用户在相应界面上进行操作，发送的请求交由系统的业务操作层来处理，在业务操作层，定义了所有的用例实现，如账号注册、生成订单、数据统计、生成分享等。业务操作层位于网络交互层之上，当用户触发某一业务请求时，需要通过网络交互层与下层的数据持久层进行沟通。网络交互层是一个过渡层，定义了请求的各种接口方法与实现，目的是将网络请求与业务逻辑分离，业务操作层通过调用网络交互层（亦为网络接口层）与服务器进行通信。数据持久层包含数据库中的各种表信息和数据信息，是服务器提供服务的基本要素，如用户账号信息、订单信息、好友信息、分享信息、权限信息、物流信息等，具体的数据库设计在下一节将会详细描述。

4.4.3 系统部署设计

用户使用手机壳定制系统时，手机客户端可以在 Android 手机和平板上使用，通过设备接入 2G、3G、4G 或者 Wi-Fi 网络与服务器进行交互。在无网的状态下，系统无法正常使用，图 4-5 展示了系统的使用结构，可以看到手机客户端和工厂客户端与网络交互层进行交互（为了系统的安全，我们可以使用加密或者防火墙技术[11]），然后网络交互层通过网络接入，可以是 2G、3G、4G 或 Wi-Fi，与数据持久层进行通信，数据持久层与数据库进行交互，即提交和获取数据。

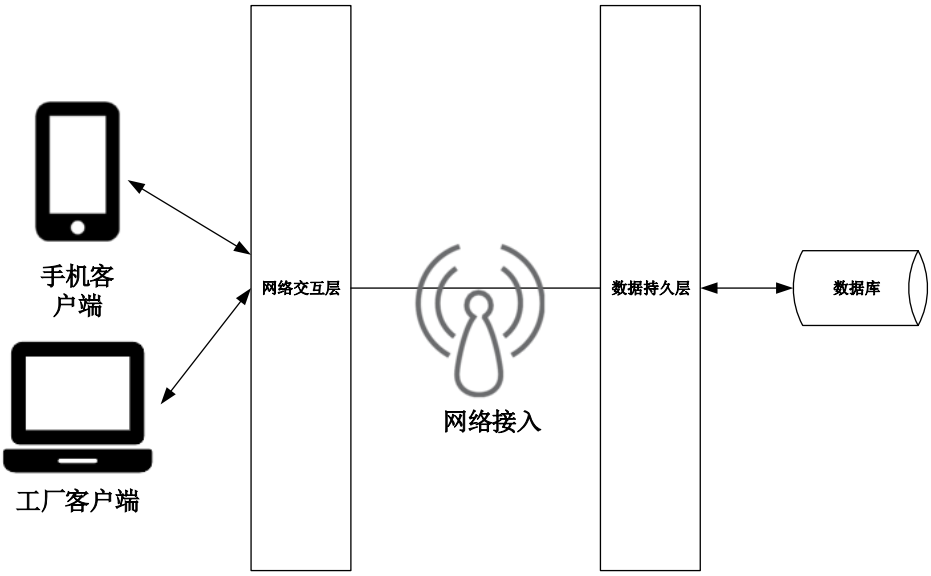

图 4-5 系统部署设计结构

4.5 数据库设计

4.5.1 数据库表设计

数据库共涉及 14 张表，表 4-1 列出了数据库中所有手机壳系统相关的表。

表 4-1 数据库表设计

表名	中文解释	描述	备注
TAB_ACCESSIBLEIP	可访问白名单	对不在 IP 表的用户限制访问	无
TAB_CONSIGEE	收货人信息表	存储收货人的具体信息	无
TAB_EXPRESSINFO	物流信息表	存储订单的物流信息	无
TAB_EXPRESSTRACE	物流路径表	存储物流配送的过程信息	无
TAB_NOTIFY	发货信息表	存储发货信息	无
TAB_ORDER	订单信息表	存储所有订单的具体信息	无
TAB_ORDERSCHEDULE	订单日程表	存储订单重要的日期节点	无
TAB_PRODUCT	产品数量信息表	存储待制作的产品数量和规格	无
TAB_PRODUCTDEF	产品定义表	存储产品规格信息	无
TAB_USER	用户信息表	存储用户信息	无
TAB_PHONESHELLPIC	手机壳图片表	存储所有手机壳图片的二进制	无
TAB_FRIENDS	好友信息表	存储用户的好友信息	无
TAB_SHARE	分享表	存储用户发表的分享信息	无
TAB_RIGHTS	权限信息表	存储工厂端用户的权限信息表	无

4.5.2 数据库概念模型图

该数据库概念模型图对手机壳系统的所有数据表之间的关系进行了建模，其中一对一的关系有 tab_order—tab_expressinfo、tab_order—tab_notify、tab_order—tab_consigee、tab_order—tab_orderschedule；

一对多的关系有 tab_user—tab_order、tab_user—tab_friends、tab_user—tab_share、tab_order—tab_expresstrace、tab_order—tab_phoneshellpic、tab_order—tab_product、tab_product—tab_productdef；

多对多的关系有 tab_user—tab_rights。因为数据库概念模型可以完全包含 ER 图的信息，所以本文就使用更专业的概念模型来描述数据库实体之间的关系和实体所包含的数据信息，具体信息见图 4-6。

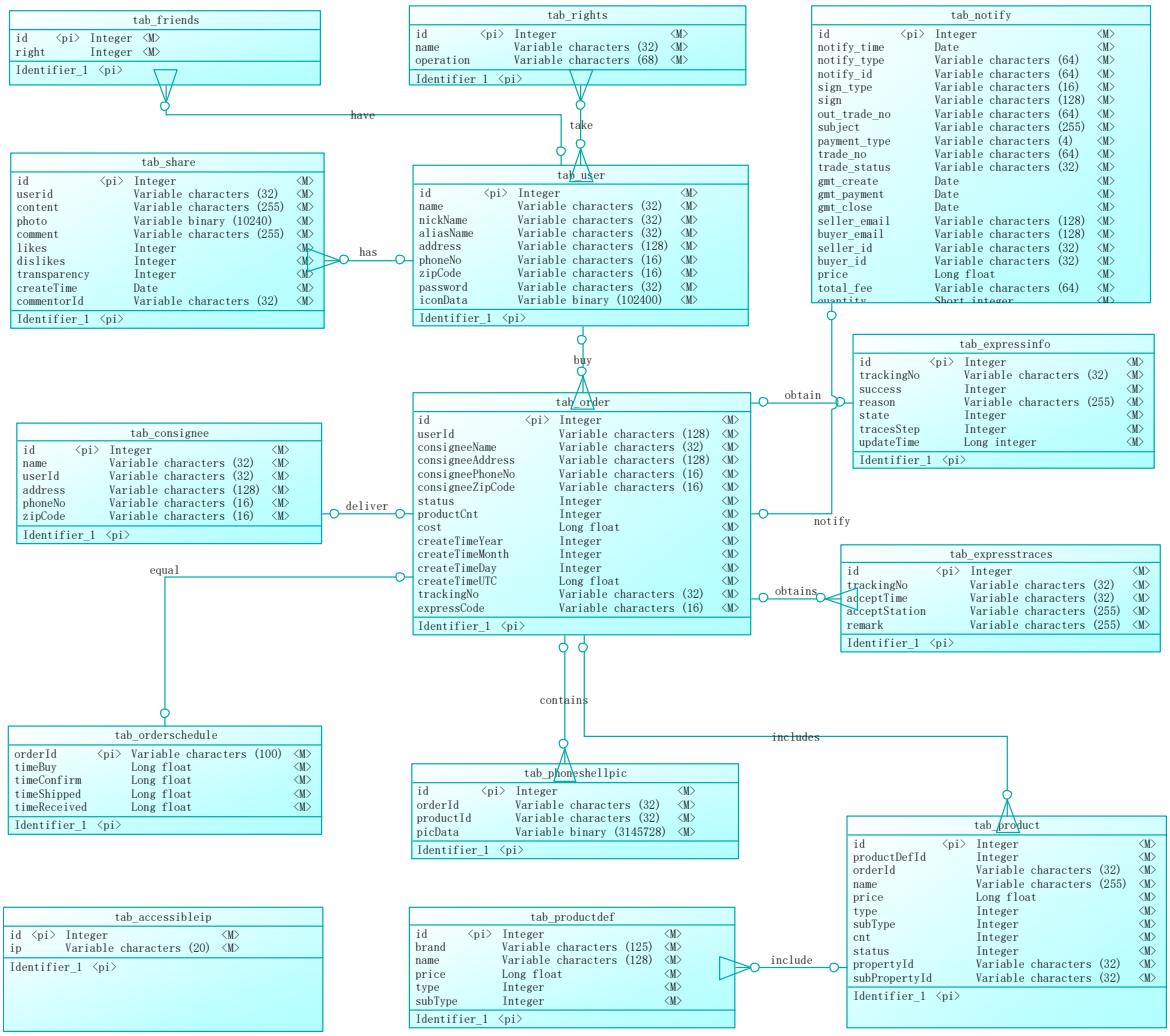

图 4-6 数据库概念模型图

4.5.3 数据库逻辑模型图

数据库逻辑模型图是对数据库概念模型图的进一步具体化，与上幅图不一样的是，不同的实体由于彼此的关系，加入了外键，同时对于多对多关系，产生了中间表。这样保证了数据库的信息在最少冗余的情况下，既节省了空间，又便于数据的维护与备份。逻辑模型图请见图 4-7。

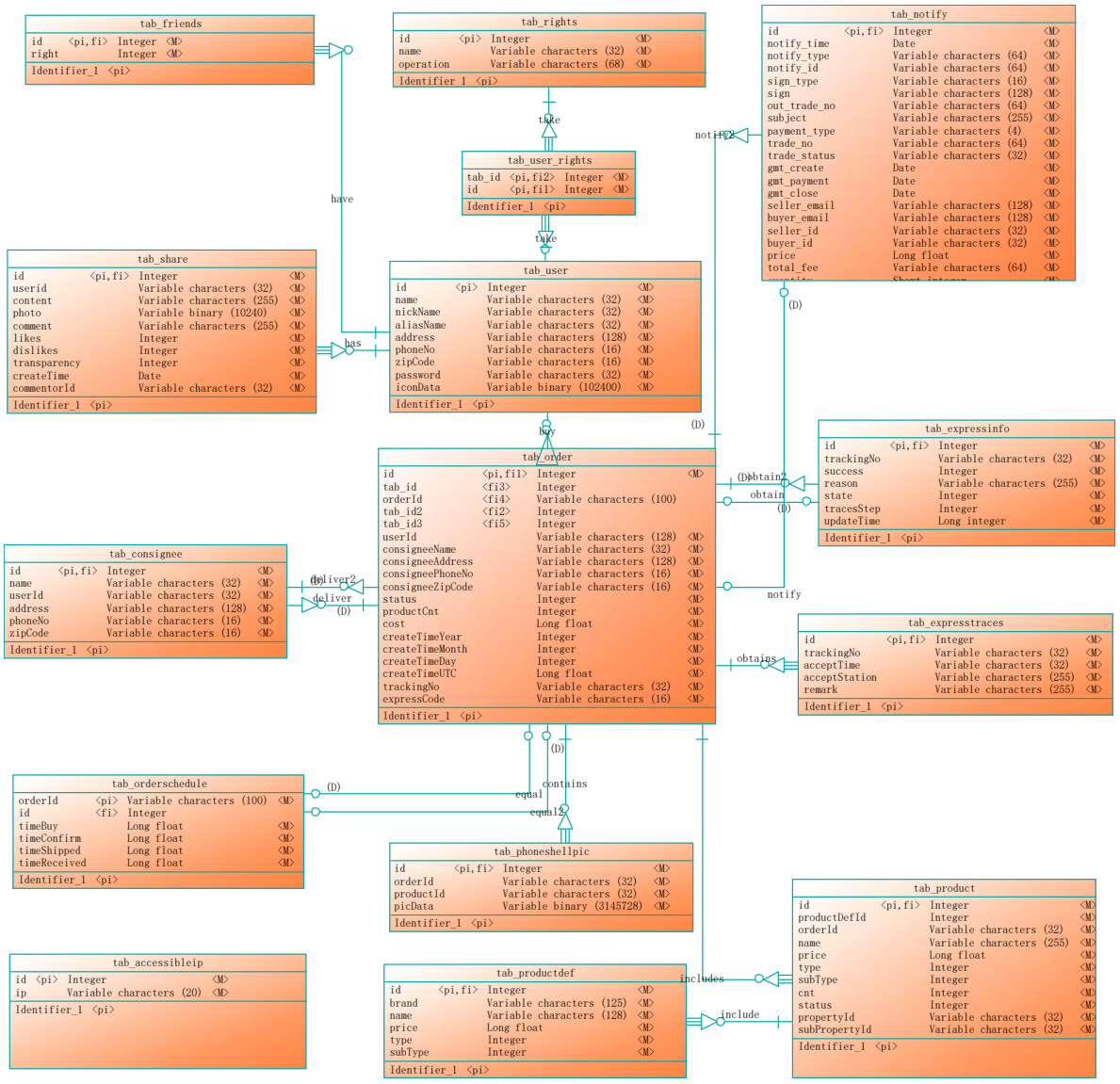

图 4-7 数据库逻辑模型图

5 手机壳定制系统模块设计与实现

手机壳定制系统的手机客户端一共有五个模块：定制模块、用户管理模块、订单管理模块、支付模块和社区分享模块；工厂客户端一共有四个模块：订单管理模块、用户权限管理模块、数据统计模块和物流管理模块。

定制模块提供用户编辑自定义手机壳图片的功能，用户可以选择手机相册的图片，进行缩放、旋转、加入第三方特效等。

用户管理模块提供用户的信息管理，包含注册、登陆、注销、用户偏好管理等信息的维护。

订单管理模块类似淘宝，京东的订单处理页面，让用户可以新增，查看和修改订单。

支付模块目前接入支付宝，用户可以选择立即支付或者下订单后在一定时限内支付。

在社区分享模块中，平台自带一个社区板块，按照热门、最新两种方式进行排序，用户可以发布自己的作品，也可以为其他用户的作品点赞，同时系统支持将作品分享到常用的社交平台，如 QQ 空间、微信朋友圈、新浪微博、人人等。

工厂端的订单管理模块根据订单的三个状态，未制作、制作中、已完成来综合处理订单。工厂端做出处理后，订单的状态会向后迁移，同时可以及时查看订单的详细信息和顾客的相关开放信息。

在工厂端的用户权限管理模块中，由于不同工厂后台的用户，职责不一样，能够浏览的信息权限也不一样，系统允许管理员给后台用户分配或撤销权限，相应用户登陆后，只能查看自己的相关信息。

工厂端的数据统计模块属于商务智能，顺应大数据的趋势，已数据可视化的方式统计订单的相关信息。

工厂端的物流模块与物流第三方应用对接，工厂给制作完成的订单进行发货，查看物流状态。

5.1 系统手机客户端功能模块设计与实现

5.1.1 定制模块

5.1.1.1 定制模块流程图

手机客户端的定制模块是提供手机壳制作的核心模块，该模块包含获取素材、编辑素材和添加特效三个部分。用户在定制手机壳之前，需要先选择手机的型号，目前系统提供三种手机型号，分别是 MX5、Pro 和 Note，点击相应型号之后，系统将会询问用户是直接拍照还是从相册中获取，如果用户选择拍照，系统将调用安卓自带的拍照工具，如果系统有多个拍照 app，系统会询问用户，选择其他拍照工具。用户拍照完毕之后，系统会自动将原图载入图片编辑区域，用户可以选择是否继续或返回重新拍照。如果用户选择从相册中获取，系统将加载用户的所有相册，用户点击进入某个相册之后，系统接着加载相册内部所有图片的缩略图。用户只需选择某一张图片，即可将该图片加载到图片编辑区域。在该区域，系统会检测用户的手势进行相应的操作，如放大缩小、旋转等，同时用户可以添加本地内置的特效或第三方特效。图 5-1 为定制模块的处理流程图。

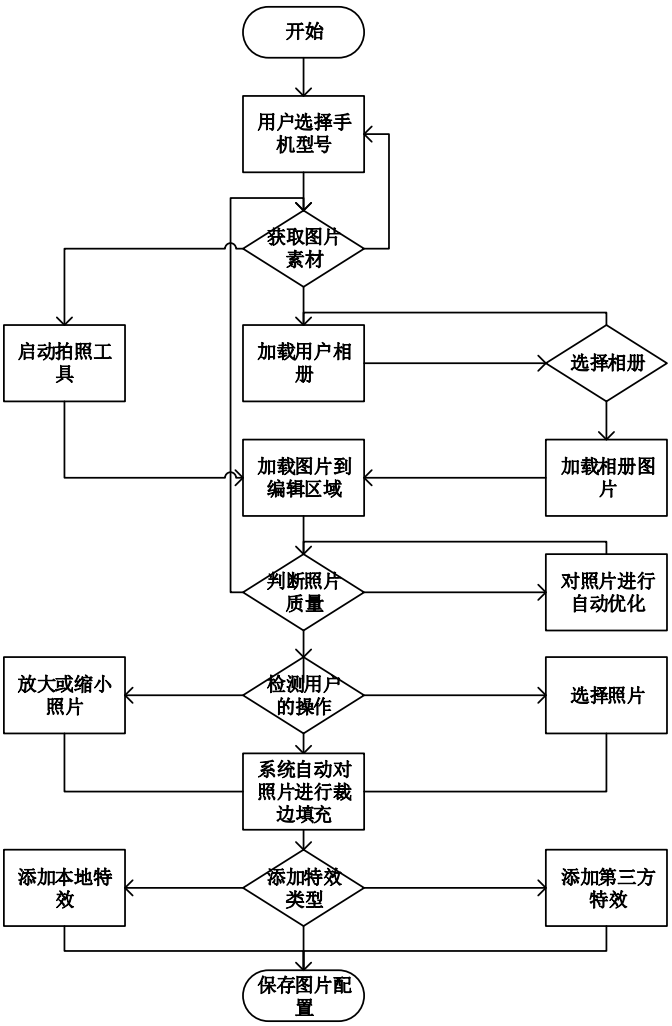

图 5-1 定制模块流程图

5.1.1.2 定制模块时序图

图 5-2 描绘了用户从最初绘制到完成的过程。首先在 MainActivity 上，APP 用户选择手机型号，然后选择获取素材图片的方式，如果系统接收到拍照消息，将打开系统照相机，用户拍摄完毕后返回所有相册，否则系统直接返回所有相册。用户选定好照片之后，系统接收到编辑照片请求，将跳转到 ShellEditActivity，该活动不断发送手势信息给 GestureDetector，然后系统不断重绘和刷新界面，直到系统接受到编辑完成信息。最后系统跳转到 EditFinishActivity。

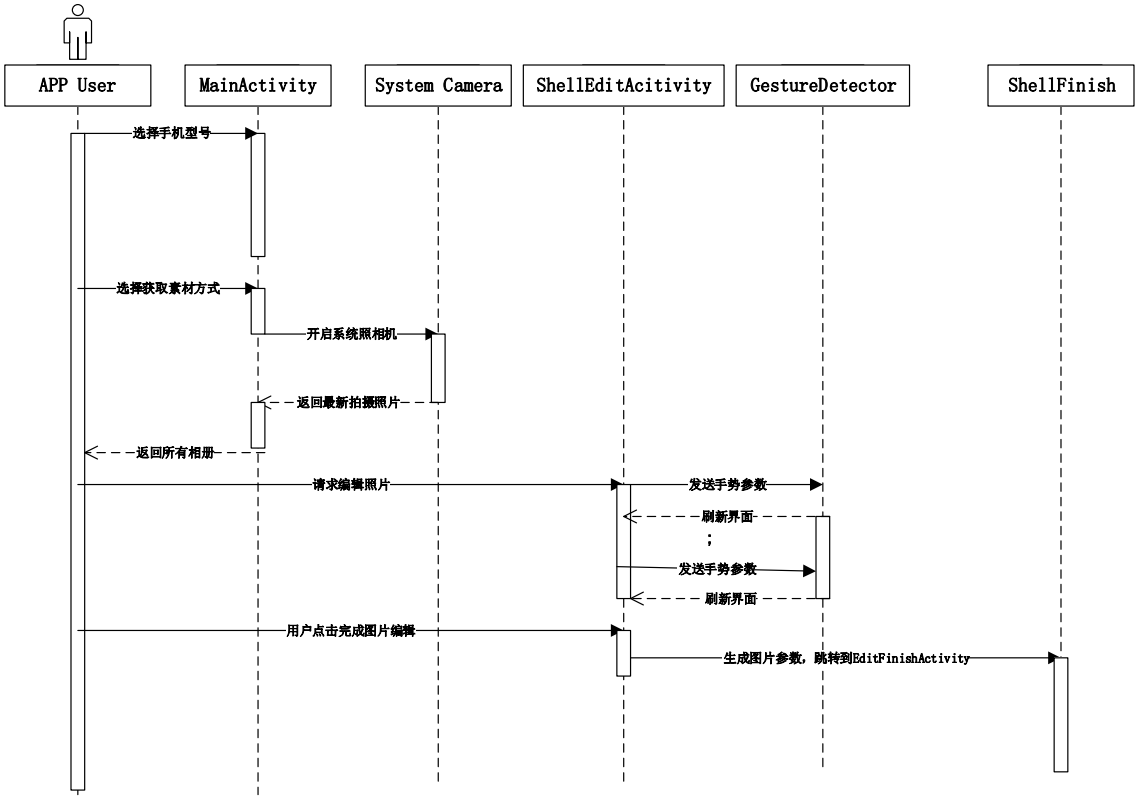

图 5-2 定制模块时序图

5.1.1.3 定制模块表结构设计

1. 手机壳图片表 tab_ phoneshellpic

表 5-1 手机壳图片表 tab_phoneshellpic

字段名	类型	允许空	描述	备注
ID	INTEGER	N	对每个图片的识别 id	主键
ORDERID	VARCHAR2(32)	N	订单号	
PRODUCTID	VARCHAR2(32)	N	产品号	
PICDATA	BIOB (3145728)	N	图片二进制存储	

该表保存手机壳图片二进制文件，且每一个图片都被唯一的订单号和产品号标识，并有自增主键 ID。

5.1.1.4 定制模块实现

1. 定制模块类图

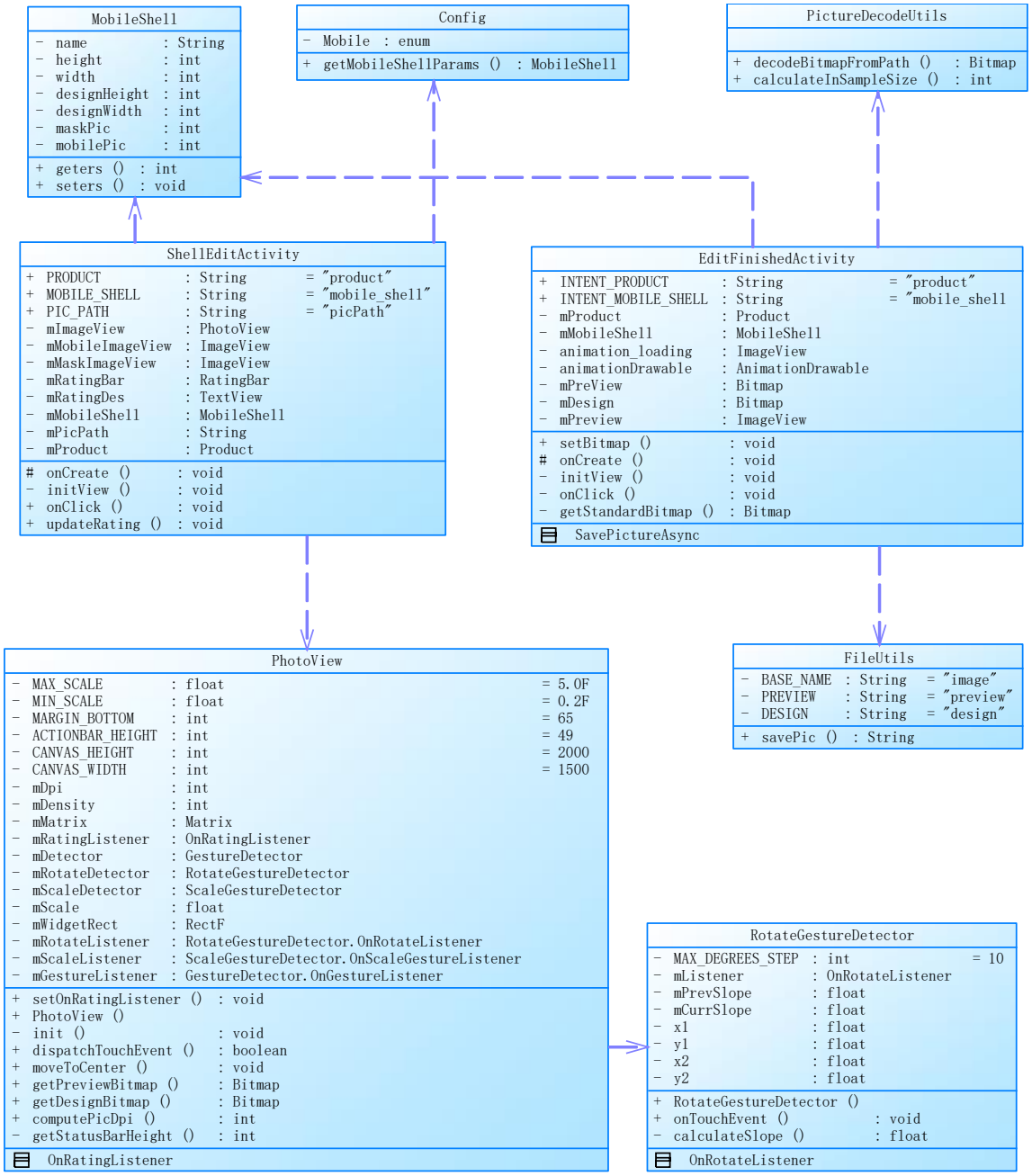

图 5-3 定制模块类图

2. 类图说明

首先介绍 PhotoView 类，该类继承 ImageView，自定义了图片区域显示的长和宽，并包含 3 个监听器的实现，分别是 mRotateListener、mScaleListener、mGestureListener，即手势选择监听器，放大缩小监听器和手势监听器。该类依赖于 RotateGestureDetector 类，该类的职责是根据用户手指在屏幕上的坐标变化，计算图片的相关参数，并传递给 PhotoView 类。在手机定制模块中，最重要的是 ShellEditActivity 和 EditFinishedActivity，其中前者为用户编辑图片的活动，它依赖于其他的 model 类和 PhotoView 工具类，来处

理图片的渲染和最终图片的参数设定。当用户在 ShellEditActivity 中结束绘制之后，Config 对象中图片参数已为赋值，并传给 EditFinishedActivity，该活动将评估图片的质量，并自动优化图片的渲染效果，同时调用 FileUtils 工具类的 savePic()方法保存最终的图片。

3. 运行效果图展示

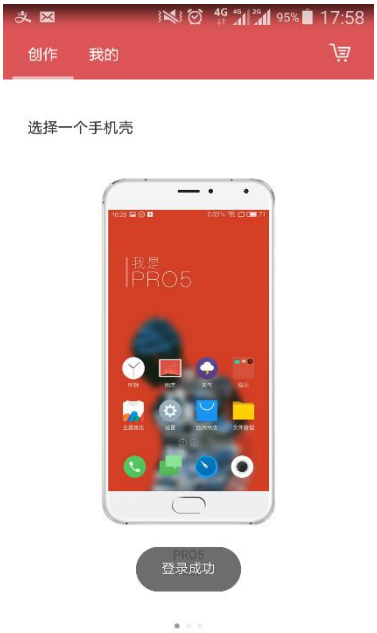

图 5-4 选择手机壳

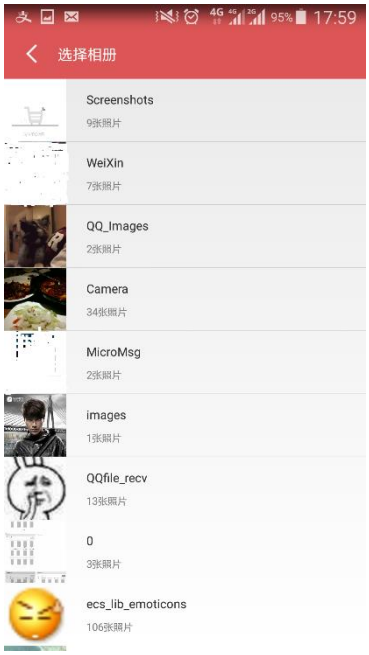

图 5-5 选择相册

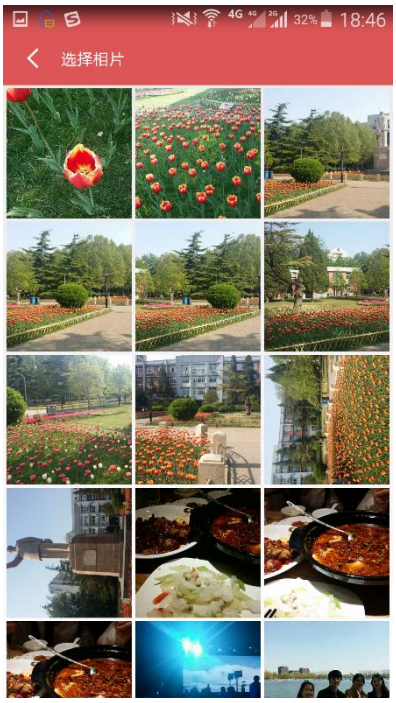

图 5-6 选择照片

图 5-7 照片编辑界面

5.1.2 用户管理模块

5.1.2.1 用户管理模块流程图

用户管理模块提供维护用户信息和验证用户登陆、注册的功能块。在用户进入应用之后，如果是初次使用，需要先注册账号，目前系统支持使用 flyme 账号和第三方账号进行授权注册。注册完毕之后用户登陆进入系统。在我的界面上，用户可以完善个人的相关信息，如上传图像，编辑信息和设置查看权限。如图 5-8 为用户管理模块的处理流程图。

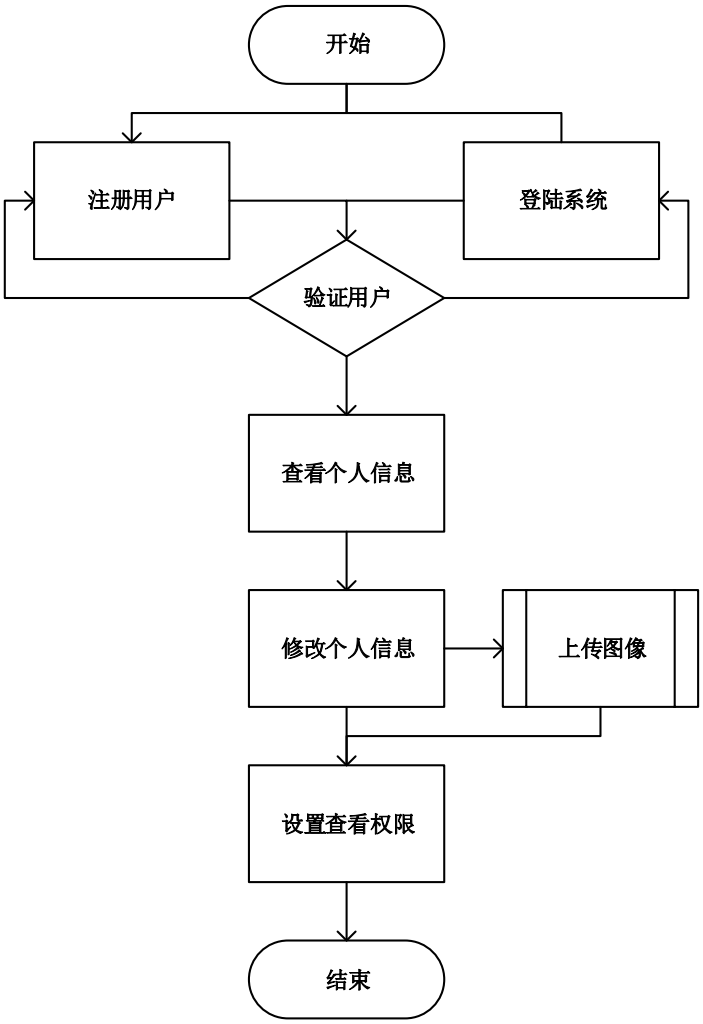

图 5-8 用户管理模块流程图

5.1.2.2 用户管理模块时序图

1. 用户登陆时序图

图 5-9 描绘了用户登录系统的过程，首先用户在 html5 界面上输入用户名和密码，并点击登录按钮，系统接收到登录请求之后，检查用户列表中是否有该用户且用户信息是否正确，如果通过验证，则返回登录成功的信息，并在 html5 界面上返回相关的用户

信息。

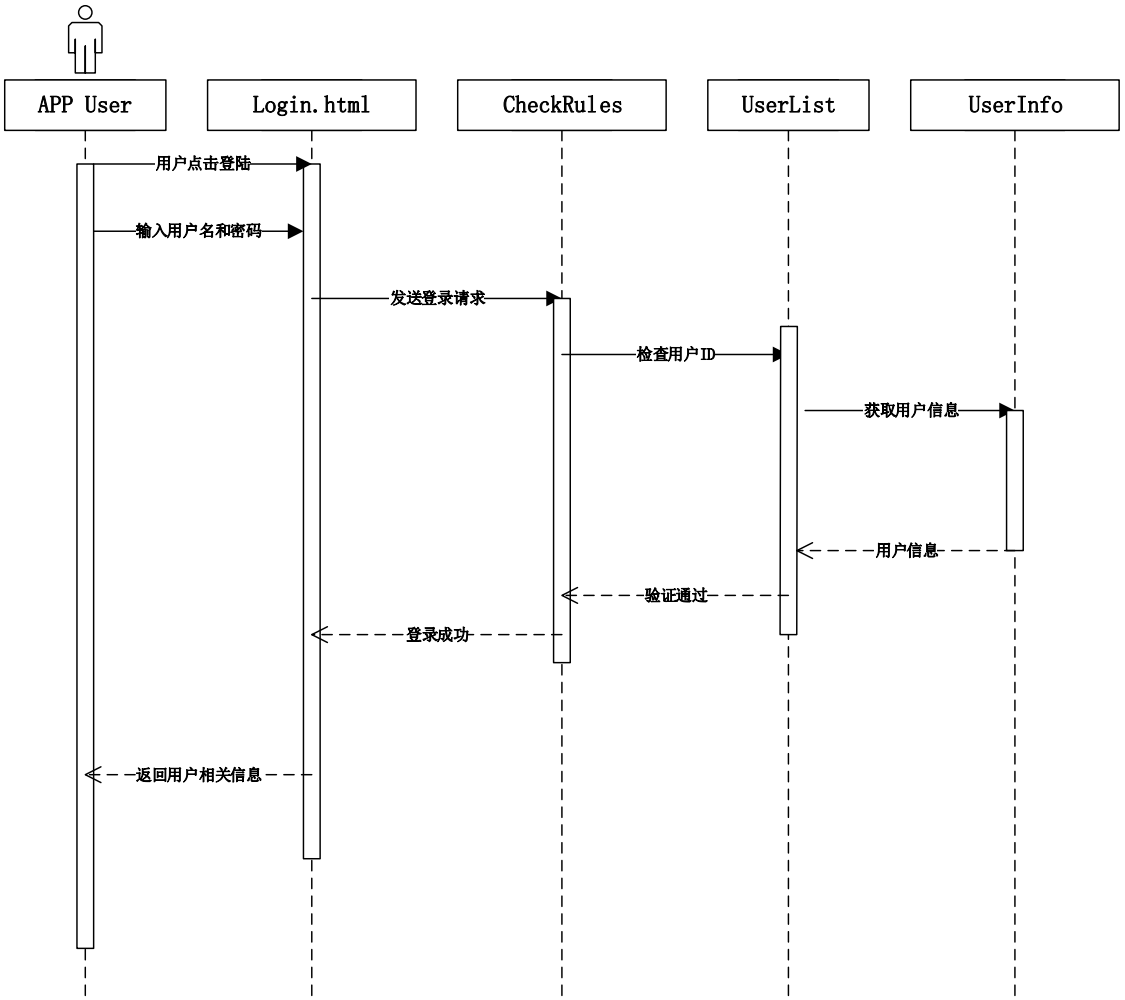

图 5-9 用户登录时序图

2. 用户注册时序图

图 5-10 描绘了用户注册系统的过程，首先用户在 html5 界面下选择注册方式，系统接收到注册命令后，给第三方账号发送授权请求，用户在 html5 界面点击授权，系统接收到授权成功信息后，发送授权后的账号信息，并写入数据库，如果注册成功，将返回注册信息给用户，并跳转到登录页面。

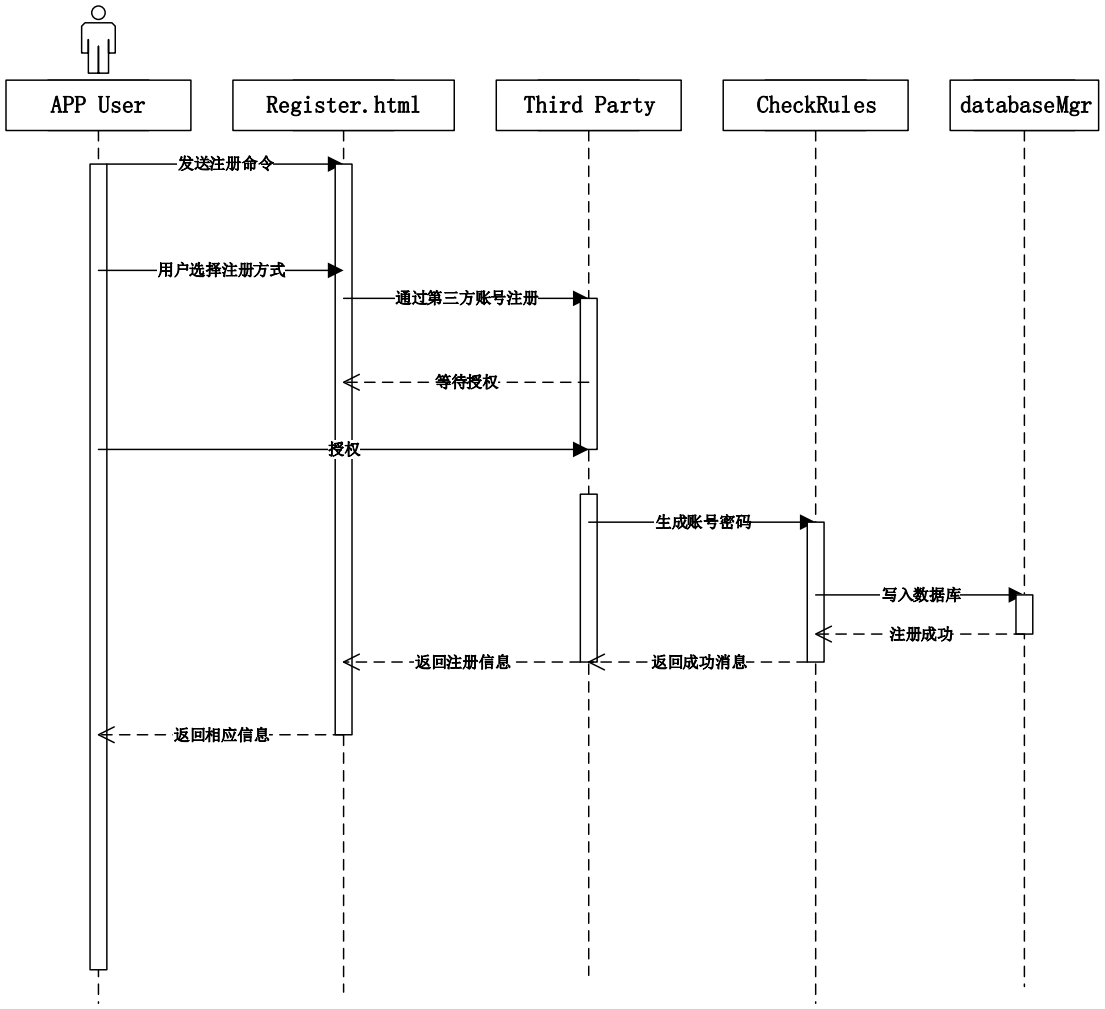

图 5-10 用户注册时序图

3. 用户管理个人信息时序图

图 5-11 描绘了用户管理个人信息的过程。首先用户在 View 层，即 UI 界面上请求更新信息，控制器 Controller 收到请求之后，更新个人信息的 Model 层，并写入数据库。由于是异步提交信息，所以 View 层先完成更新，待数据库更新完成后，系统将会返回数据库信息更新成功的信息。

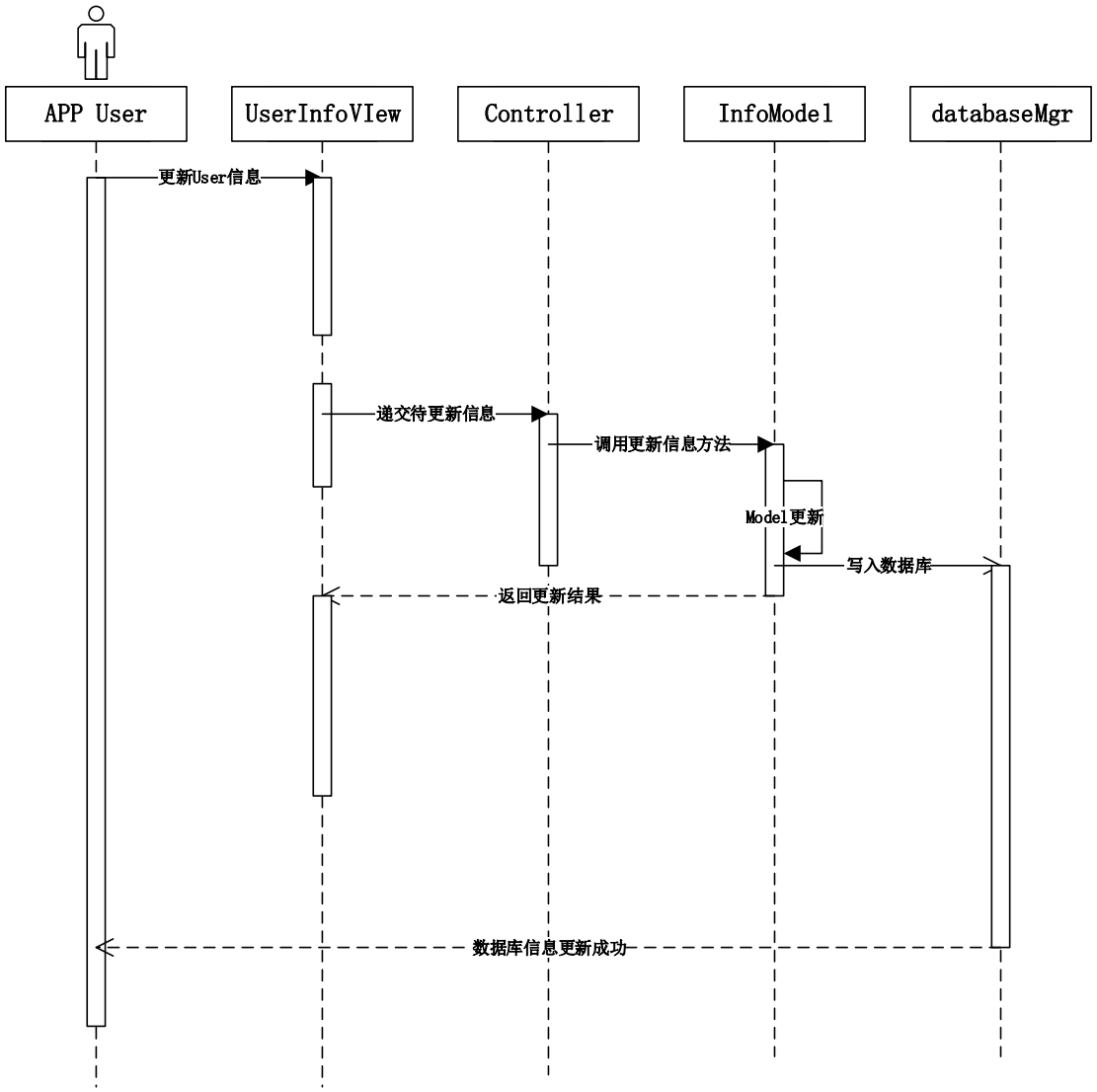

图 5-11 用户管理个人信息时序图

5.1.2.3 用户管理模块表结构设计

1. 用户信息表 tab_user

表 5-2 用户信息表 tab_user

字段名	类型	允许空	描述	备注
ID	INTEGER	N	对每个用户的识别 id	主键
NAME	VARCHAR2(32)	N	用户名	
NICKNAME	VARCHAR2(32)	N	用户昵称	
ALIASNAME	VARCHAR2(32)	N	名称缩写	

表 5-2 用户信息表 tab_user<续>

ADDRESS	VARCHAR2(128)	N	用户地址	
PHONENO	VARCHAR2(16)	N	手机号	
ZIPCODE	VARCHAR2(16)	N	邮编号	
PASSWORD	VARCHAR2(32)	N	密码	
ICONDATA	BIOB(10240)	N	用户图像二进制	

5.1.2.4 用户管理模块实现

1. 用户管理模块类图

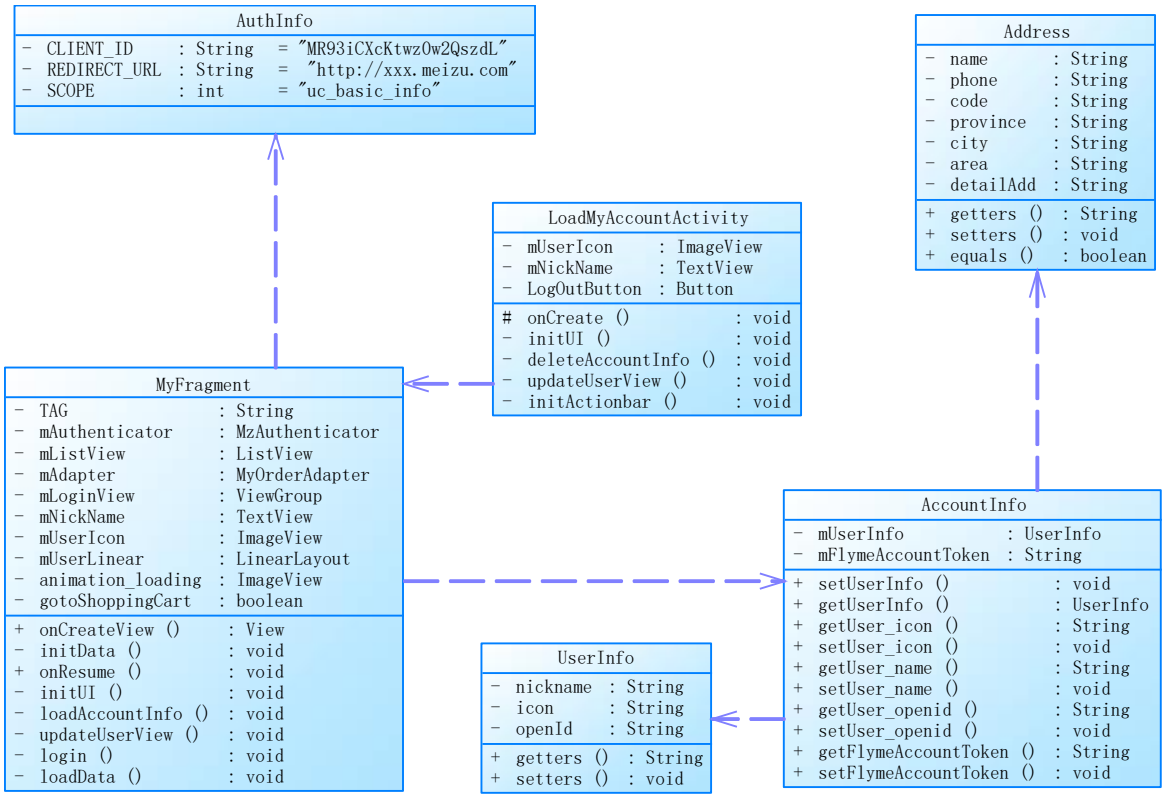

图 5-12 用户管理模块类图

2. 类图说明

用户管理模块涉及到四个实体类，分别是 UserInfo、Address、AuthInfo 和 AccountInfo。AccountInfo 类是对用户基本信息的再一次封装，持有 UserInfo 和 Address 类的引用。该模块包含 LoadAccountActivity 活动类，负责注销用户账号，修改用户信息登录注册职能。该活动依赖于 MyFragment 类，该类封装了关于用户管理模块用例的所

有实现，并通过网络与服务器进行交互，来完成登陆注册和更新用户信息的请求。

3. 效果展示

图 5-13 选择登录方式

图 5-14 注册界面

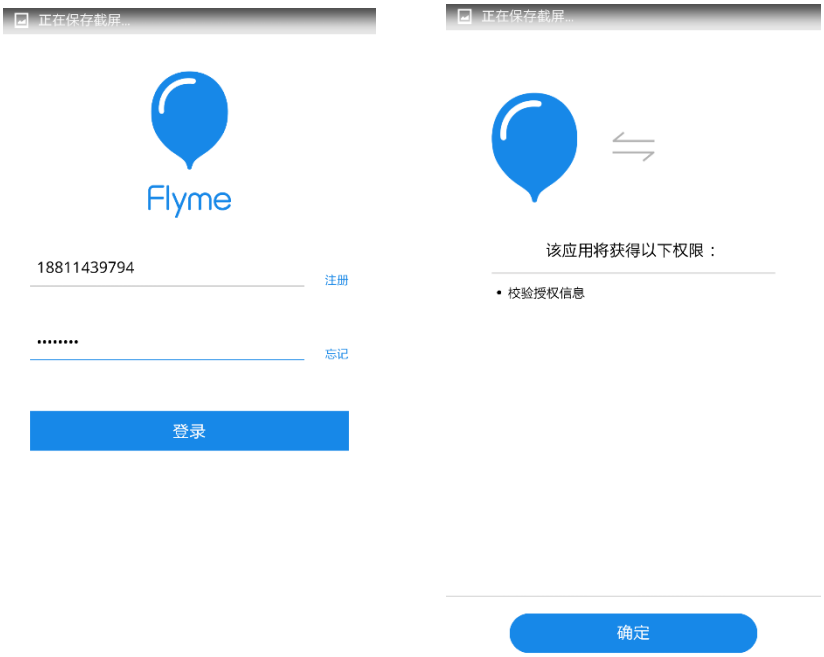

图 5-15 登录界面（flyme 账号）

图 5-16 更新个人信息界面

5.1.3 订单管理模块

5.1.3.1 订单管理模块流程图

订单管理模块是用户处理和查看订单信息的核心功能块。用户在定制模块部分完成

了手机壳图片的编辑之后，点击加入购物车，系统根据不同手机型号的配置信息生成一个产品信息对象，待提交给服务器。在购物车界面，用户可以编辑一个产品的数量，系统默认的产品数量为 1。用户也可以返回主界面，制作新的手机壳再次加入购物车，购物车会根据用户的相关操作，实时的统计产品信息，计算出总的价格。当用户确定好购物车的产品明细之后，可以点击提交订单。对于第一次下单的用户，系统需要先录入收货信息。用户在录入收货信息界面，需要填写真实姓名，手机号，详细地址和邮编，填写保存，系统会记录并保存至服务器，同时也会在本地备份，作为用户偏好设置的信息来源，当用户第二次下单时，基于以前的收货信息，会默认获取本地信息，并确认。收货信息确认之后，用户还需要选择支付方式，目前系统支持支付宝、微信支付和银联支付，同时系统会检测是否有可用优惠券，并更新待支付金额。最后用户点击提交订单，系统将生成订单，并提交至服务器。用户可以在我的界面下查看所有的订单信息，并可以点击单个订单，查看该订单的详细信息。对于未支付的订单，用户可以取消订单。图 5-17 为订单管理模块的流程图。

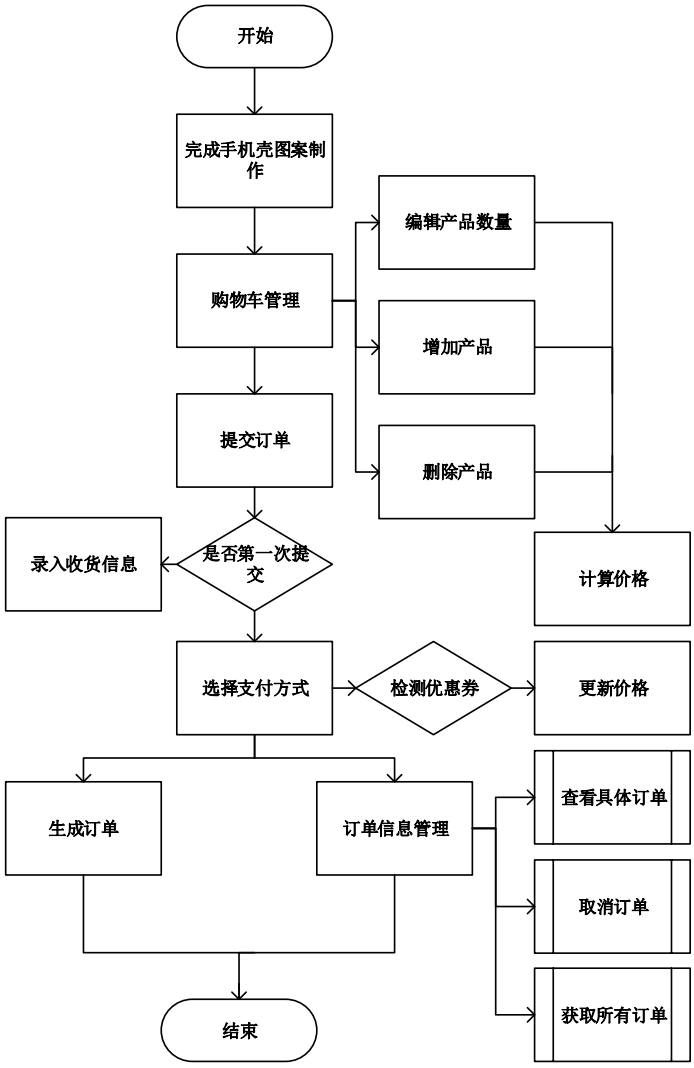

图 5-17 订单管理模块流程图

5.1.3.2 订单管理模块时序图

1. 购物车维护时序图

图 5-18 描绘了用户更新购物车，管理购物车内容的过程。首先用户在 ShellFinish 界面请求编辑完成，并加入购物车，系统将产品信息发送给 ShoppingCartActivity，并返回添加成功信息，在 ShoppingCartActivity 活动上，用户发送编辑产品数量请求后，系统将计算更新后的价格信息，CheckPrice 组件收到计价请求之后，会想数据库请求是否有可用的优惠券，并根据数据库返回的信息，进行计算，最后返回价格信息。如果用户发送删除产品请求，系统同样会更新价格信息，处理流程同上。当用户确定购物车之后，系统将生成订单信息并进行跳转，同时会清空当前的购物车内容。

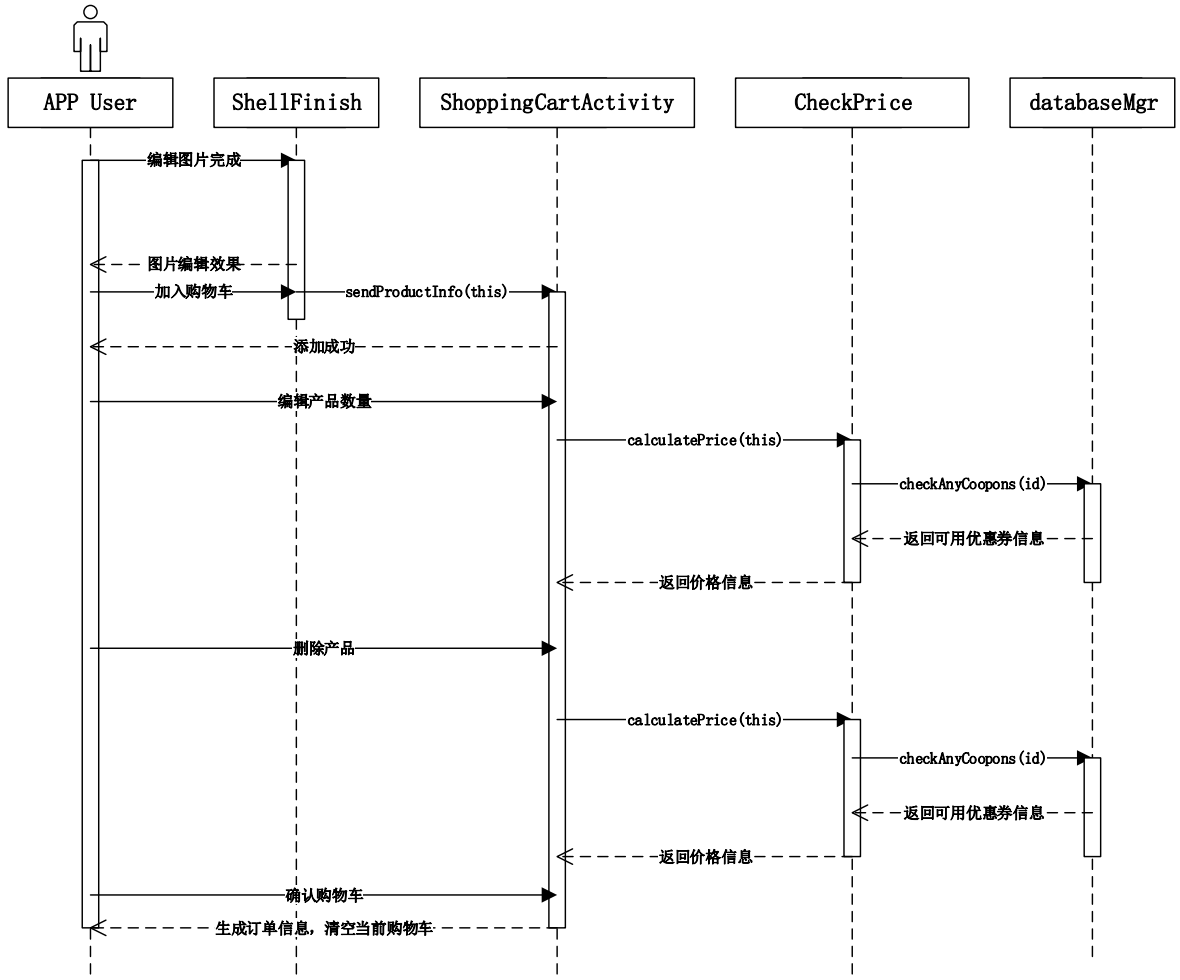

图 5-18 购物车维护时序图

2. 生成订单时序图

图 5-19 描绘了系统生成订单的过程。首先用户发送确定购物车信息之后，系统将跳转到 MyFrameActivity，并检查用户是否第一次下单，如果不是，AddressMgr 组件将会返回用户常用收货地址信息否则，系统将跳转到 AddressActivity 让用户输入收货信息，输入完成之后，返回收货信息给 AddressActivity，并刷新 MyFrameActivity，然后用户请求选择支付方式，系统将查询可用的支付方式，并返回给用户。确认完毕订单信息之后，用户发送确认下单请求，系统请求将订单信息写入数据库，并返回下单成功的信息给用户。

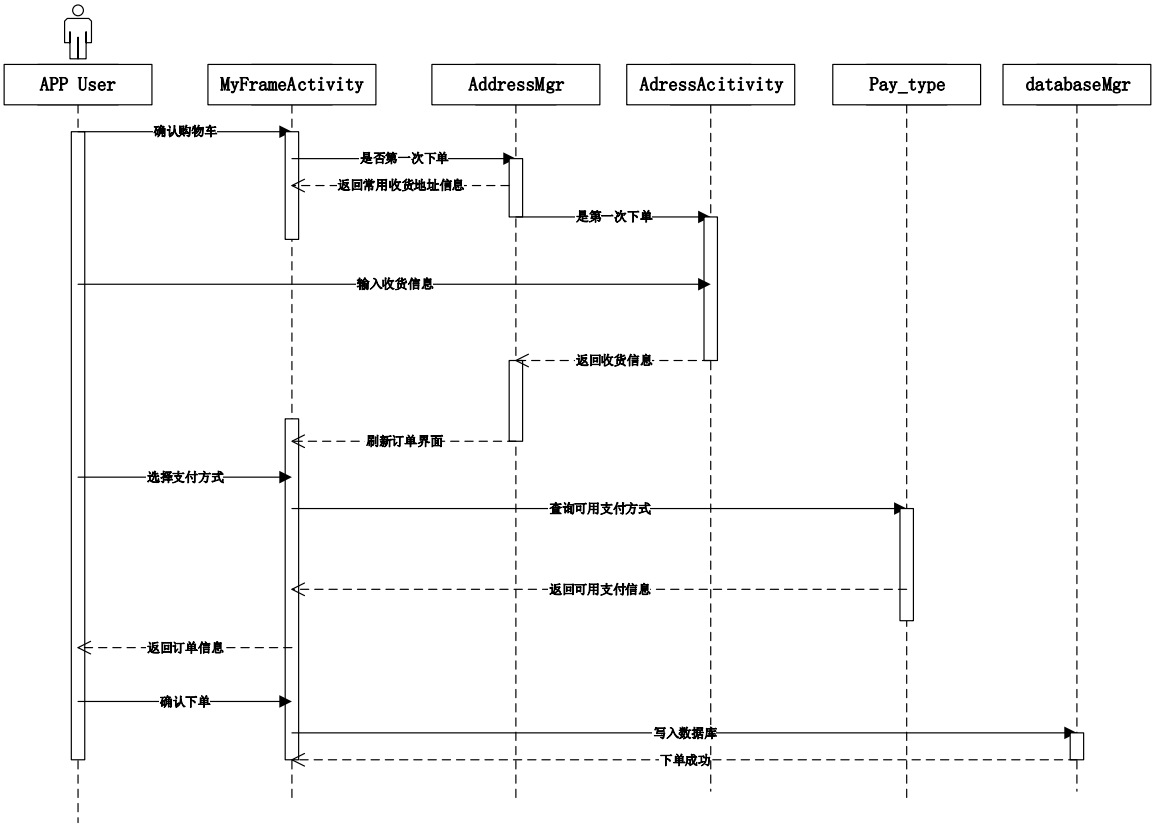

图 5-19 生成订单时序图

3. 订单信息管理时序图

图 5-20 描述了用户管理订单信息的过程。用户管理订单信息，包含获取所有订单、查询单个订单信息、取消某个订单。系统处理过程是在 View 层提交请求给 Controller，即控制层，控制层向数据库提交请求并等待数据库返回数据，更新信息 Model 层，Model 层更新后，View 层将会被刷新，最终用户收到请求的信息。如用户在 MyFrameActivity 请求获取所有订单，MyFrameActivity 发送 handleGetAllOrders()命令给 Controller，然后 Controller 发送 getAllOrders(id)命令给 databaseMgr，返回的订单信息更新 InfoModel，然后更新 MyFrameActivity，用户就获取到所有订单的信息了。

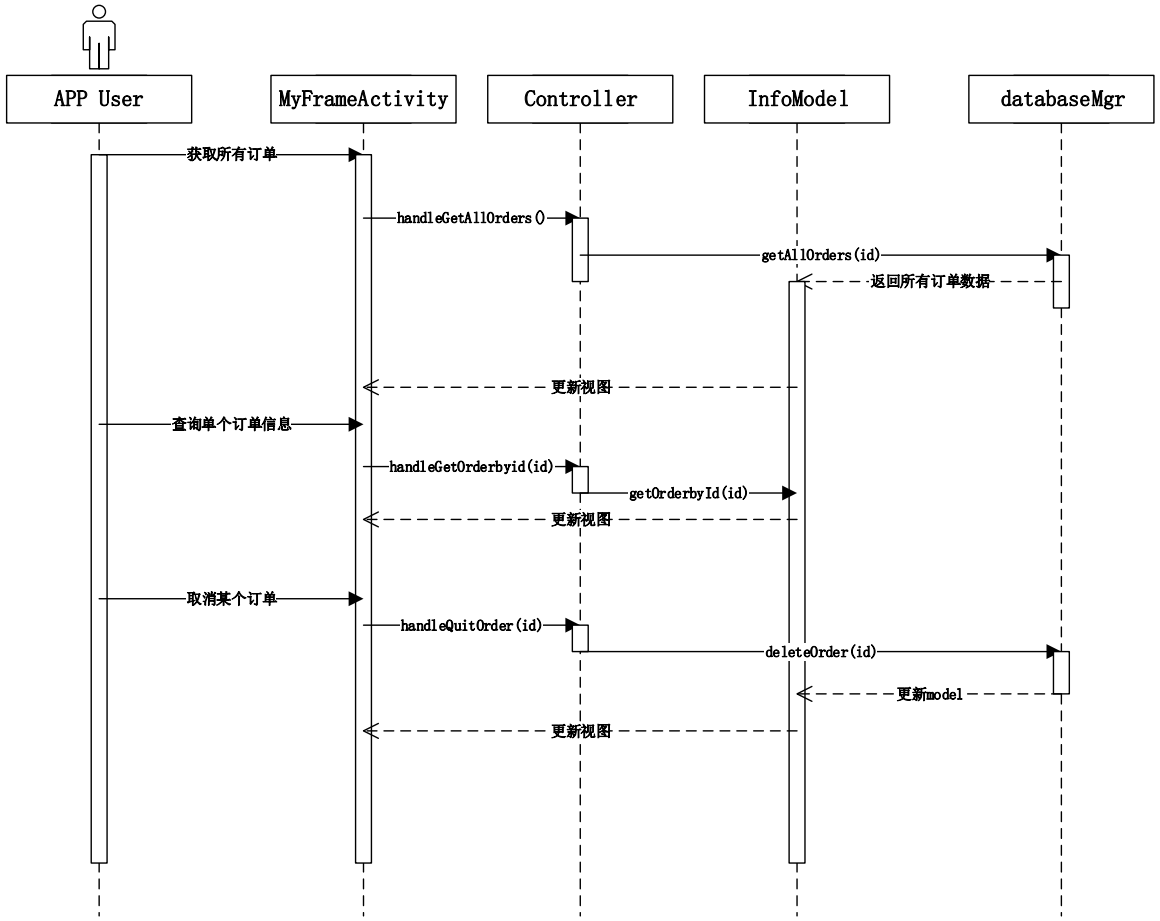

图 5-20 订单管理模块时序图

5.1.3.3 订单管理模块表结构设计

1. 收货人信息表 tab_ consigee

表 5-3 收货人信息表 tab_consigee

字段名	类型	允许空	描述	备注
ID	INTEGER	N	对每个用户的识别 id	主键
NAME	VARCHAR2(32)	N	用户名	
USERID	VARCHAR2(32)	N	用户 ID	
ADDRESS	VARCHAR2(128)	N	用户地址	
PHONENO	VARCHAR2(16)	N	手机号	
ZIPCODE	VARCHAR2(16)	N	邮编号	

注：表 tab_consigee 存储收货人的具体信息。

2. 订单信息表 tab_order

表 5-4 订单信息表 tab_order

字段名	类型	允许空	描述	备注
ID	INTEGER	N	对每个订单的识别 id	主键
USERID	VARCHAR2(32)	N	用户 ID	
CONSIGNAME	VARCHAR2(32)	N	收货人姓名	
ADDRESS	VARCHAR2(128)	N	收货人地址	
PHONENO	VARCHAR2(16)	N	收货人手机号	
ZIPCODE	VARCHAR2(16)	N	收货人邮编号	
STATUS	INTEGER	N	订单状态	
PRODUCTCNT	INTEGER	N	产品数量	
COST	LONG FLOAT	N	价格	
CREATETIMEY	INTEGER	N	订单创建年份	
CREATETIMEM	INTEGER	N	订单创建月份	
CREATETIMED	INTEGER	N	订单创建日期	
CREATEUTC	LONG FLOAT	N	订单创建 UTC	
TRACKNO	VARCHAR2(32)	Y	运单号	
EXPRESSNO	VARCHAR2(16)	Y	物流号	

注：表 tab_order 存储订单的具体信息。

3. 订单日程表 tab_orderschedule

表 5-5 订单日程表 tab_orderschedule

字段名	类型	允许空	描述	备注
ORDERID	VARCHAR2(100)	N	订单号	主键
TIMEBUY	LONG FLOAT	N	下单时间	
TIMECONFIRM	LONG FLOAT	N	确认订单时间	
TIMESHIPPED	LONG FLOAT	N	订单发货时间	
TIMERECEIVED	LONG FLOAT	N	订单签收时间	

注：表 tab_orderschedule 存储订单的重要时间日程信息。

5.1.3.4 订单管理模块实现

1. 订单管理模块类图

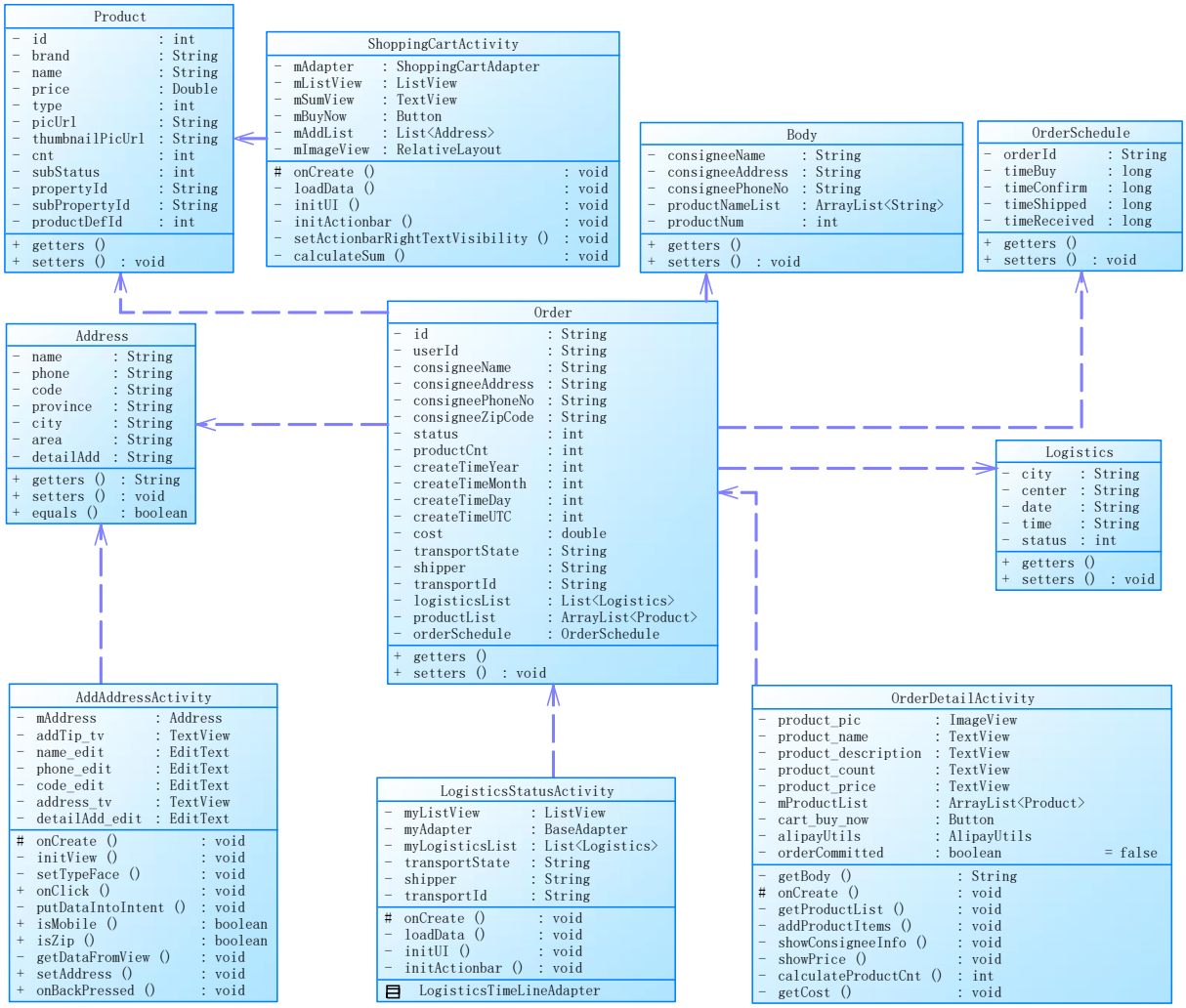

图 5-21 订单管理模块类图

2. 类图说明

先介绍该模块涉及到了几个实体类，Product 类为产品实体类，定义了用户购买手机壳商品的相关信息，如手机品牌、手机价格等；Body 类为收货人信息实体，定义了收货人的相关信息，如收货人姓名、收货人手机号等；Logistics 类为物流实体类，定义了配送物流的相关信息；Order 和 OrderSchedule 类分别是订单实体类和订单日期信息类。在订单管理模块中，用户在 ShoppingCartActivity 活动里对购物车上的商品进行编辑确认后，跳转到 OrderDetailActivity 上，该活动持有 Order 实体类的引用，该引用的实例化信息来源为 ShoppingCartActivity 通过 intent 传递过来的订单信息。用户在提交订单之前，系统会检查 SharedPreference 里是否保存的收货地址信息，如果没有，用户需要跳转到 AddAddressActivity 上进行编辑添加，完成后系统保存地址信息到本地，并与服务器数据库同步信息，最后返回 OrderDetailActivity 界面上，等待用户确认支付。

3. 效果展示

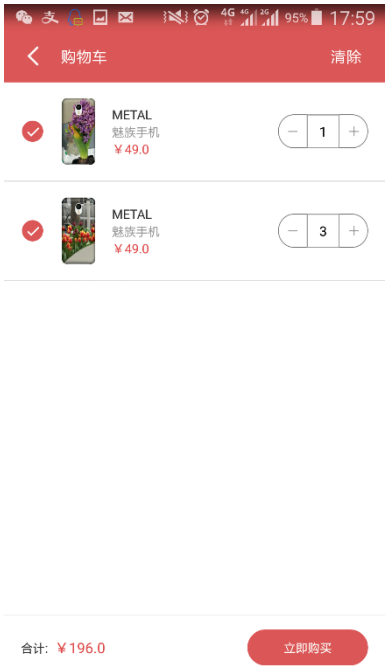

图 5-22 购物车界面

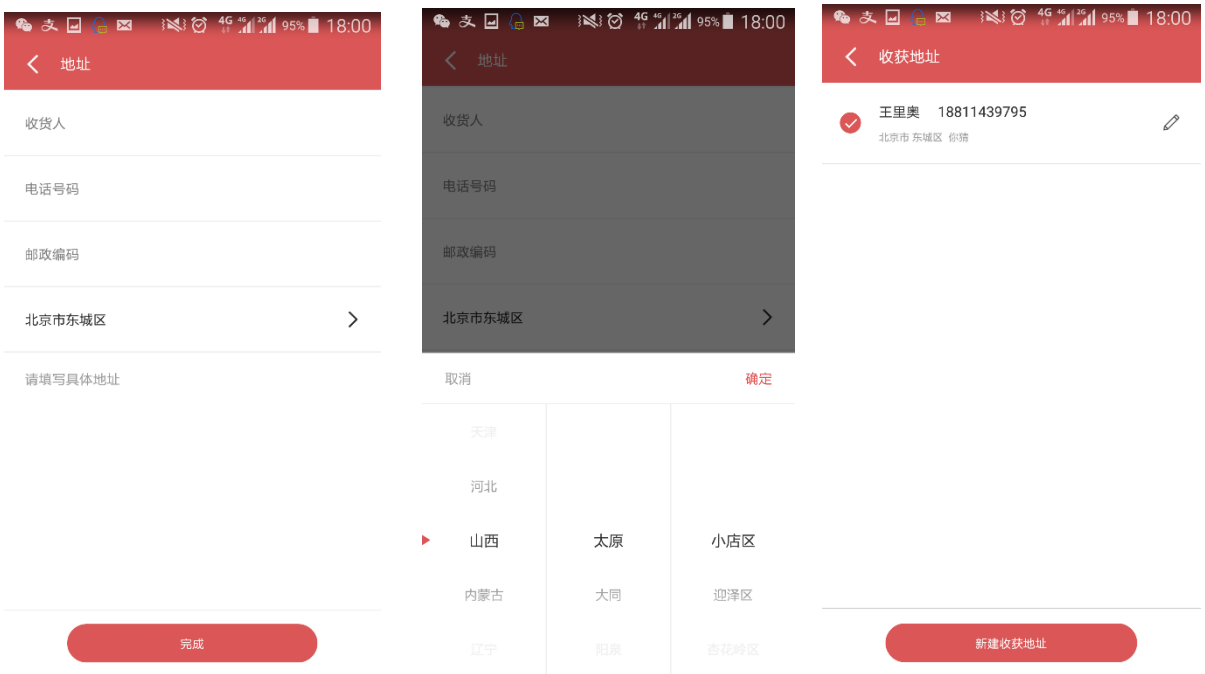

图 5-23 添加收货地址信息界面

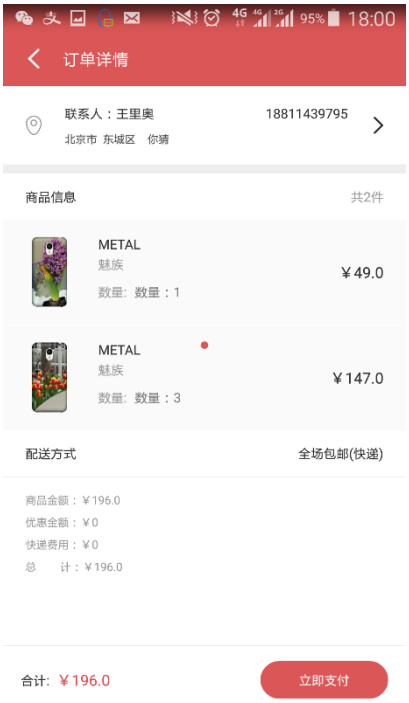

图 5-24 下单界面

5.1.4 支付模块

5.1.4.1 支付模块流程图

支付模块是用户网上支付，完成订单的最后一步功能块。目前系统调用第三方支付方式的 API，如支付宝 API，微信支付 API。当用户点击提交订单，系统生成订单信息，并等待用户进行支付。用户可以选择立即支付，也可以稍后支付。点击立即支付，系统会根据用户选择的支付方式，调用相应的 API；点击稍后支付，系统将离开当前界面，进入我的界面，在该界面下，用户可以查看所有的订单信息。如果下单后，超过一定时限仍然没有完成支付，系统会自动删除订单，用户也可以在我的界面下手动删除相应未支付的订单。用户完成支付后，系统将根据服务器返回的支付状态，确定支付结果，如果支付成功，将跳转到我的界面，显示当前订单的具体信息和状态；如果支付失败，系统将提示用户失败原因，并尝试再次跳转到支付页面。图 5-25 为支付模块的流程图。

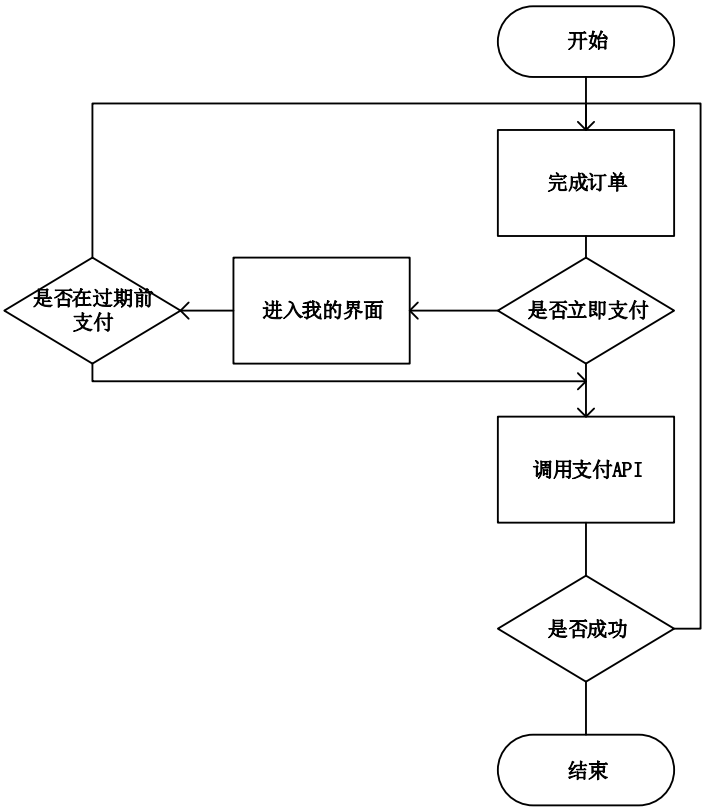

图 5-25 支付模块流程图

5.1.4.2 支付模块时序图

图 5-26 描绘了支付订单的过程。首先在 ShoppingCartActivity 活动上用户发送立即支付请求，系统将向第三方支付平台发送支付请求，第三方平台处理完毕后，将支付结果发送给数据库，数据库写入信息成功后，将支付结果返回给前端，支付完成。如果用户想支付等待支付的订单，需要向 MyFrameActivity 发送获取未支付订单的请求，然后对返回的未支付订单进行现在支付，同样的，系统向第三方支付平台发送支付请求，根据返回的支付结果，系统刷新界面并显示当前支付订单的状态信息。如果订单支付成功，订单将由未支付变成等待工厂制作的状态。

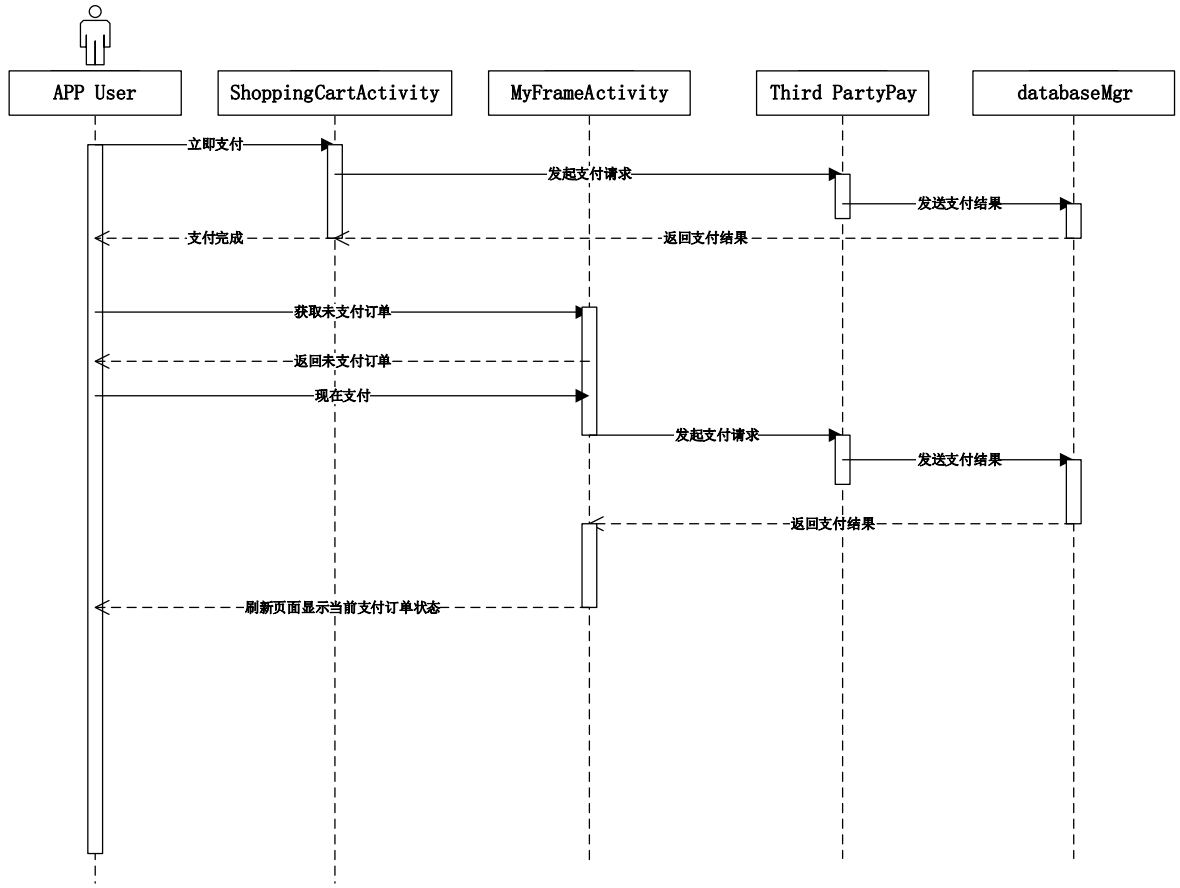

图 5-26 支付模块时序图

5.1.4.3 支付模块表结构设计

1. 产品数量信息表 tab_product

表 5-6 产品数量信息表 tab_product

字段名	类型	允许空	描述	备注
ID	INTEGER	N	对每一个产品的识别 id	主键
PRODUDEFID	INTEGER	N	产品定义 ID	
ORDERID	VARCHAR2(32)	N	订单号	
NAME	VARCHAR2(255)	N	产品名称	
PRICE	LONG FLOAT	N	产品价格	
TYPE	INTEGER	N	产品类型	
SUBTYPE	INTEGER	N	子类型	
CNT	INTEGER	N	产品数量	

表 5-6 产品数量信息表 tab_product<续>

STATUS	INTEGER	N	产品制作状态	
PROPERTYID	VARCHAR2(32)	Y	属性 ID	
SUBPROPERTY	VARCHAR2(32)	Y	子属性 ID	

注：表 tab_product 存储产品的具体信息，如产品数量，名称等。

2. 产品定义表 tab_productdef

表 5-7 产品定义表 tab_productdef

字段名	类型	允许空	描述	备注
ID	INTEGER	N	对每一个产品的识别 id	主键
BRAND	VARCHAR2(125)	N	产品型号	
NAME	VARCHAR2(128)	N	产品名称	
PRICE	LONG FLOAT	N	产品价格	
TYPE	INTEGER	N	产品类型	
SUBTYPE	INTEGER	N	产品子类型	

注：表 tab_productdef 存储产品的定义信息。

5.1.4.4 支付模块实现

1. 支付模块类图

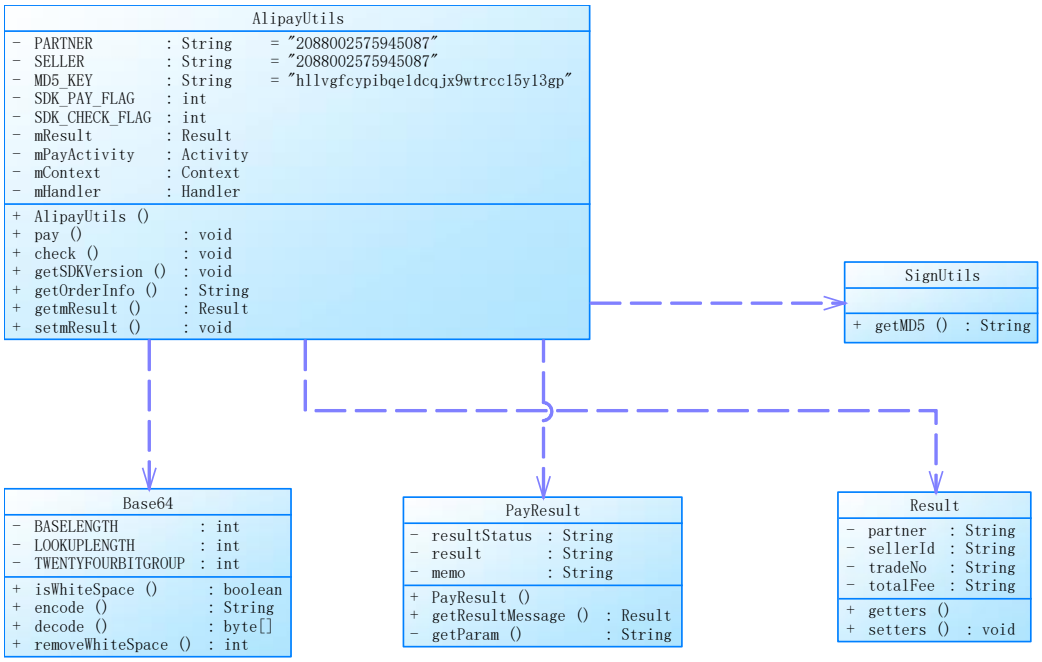

图 5-27 支付模块类图

2. 类图说明

支付模块涉及到两个实体类，PayResult 类为支付结果信息实体类，Result 类为支付信息类。虽然它们两个都有 Result，但 PayResult 类是实例化是在系统与第三方支付服务器交互之后进行的，而 Result 则在系统与第三方支付服务器交互之前实例化的，它包含买方支付 id、卖方接收款项 id、交易号和支付总费用。可以看出支付使用 MD5[12] 加密方式，使用 SignUtils 工具类获取 MD5，然后通过 Base64 进行加密和解密。系统最终是通过 AlipayUtils 工具完成支付的，支付流程为：

- 系统先通过方法 getOrderInfo()方法获取订单信息，实例化 Result 成员变量。
- 使用 pay()方法请求支付。
- 通过异步回调的方式获取服务器返回的信息，实例化 PayResult 引用。
- 使用 check()方法检测支付是否正常完成。

3. 效果展示

图 5-28 发起支付界面

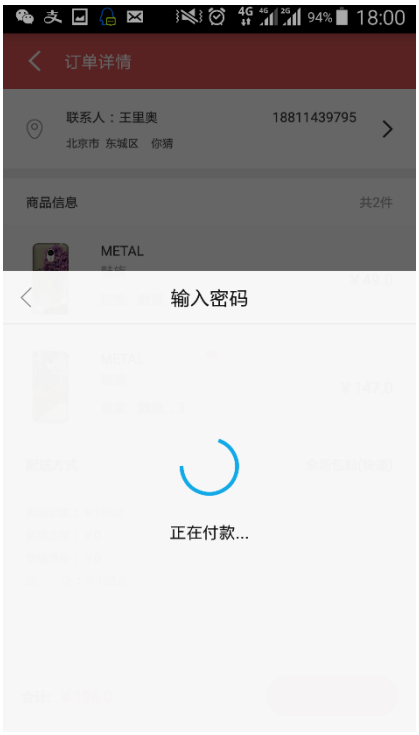

图 5-29 支付完成

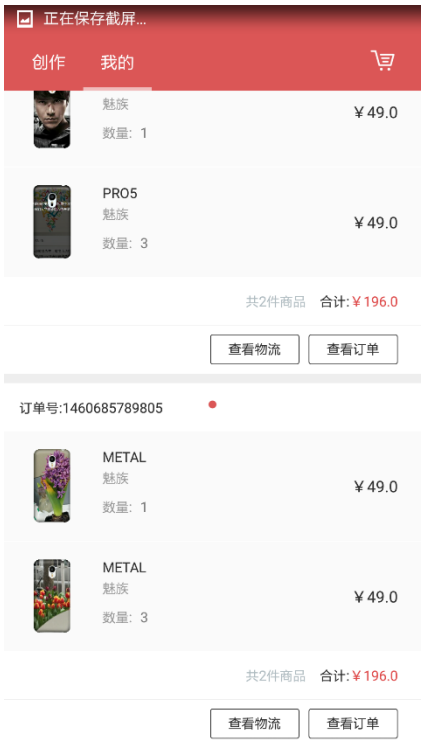

图 5-30 订单完成支付

5.1.5 社区分享模块

5.1.5.1 社区分享模块流程图

社区分享模块是用户在购买支付成功后，可以在本地社区或第三方社交平台上分享自己的作品。该模块包含四个部分，分别是查看我的分享、管理我的分享、生成我的分享和管理我的好友关系。用户完成订单之后，点击分享即可进入分享界面，系统给出两个选择，分享至本地社区和分享至其他社交平台。如果用户点击分享至本地社区，系统将自动生成分享的部分内容，其中包含待分享的手机壳图案，用户完善分享内容之后，点击确认，即完成分享的生成。在社区界面，用户可以查看所有用户的最新分享，可以按照时间、热度和地区排序，用户可以对别人的分享进行评价或点赞。用户可以在我的分享界面中查看自己的分享，其中包括分享内容，评价信息和点赞数量，并对评价进行回复。对于不再需要的分享可以删除。用户同时可以添加好友，系统将实时推送其他好友的分享内容给用户，如不需实时推送，可以选择关闭。图 5-31 为社区分享模块的流程图。

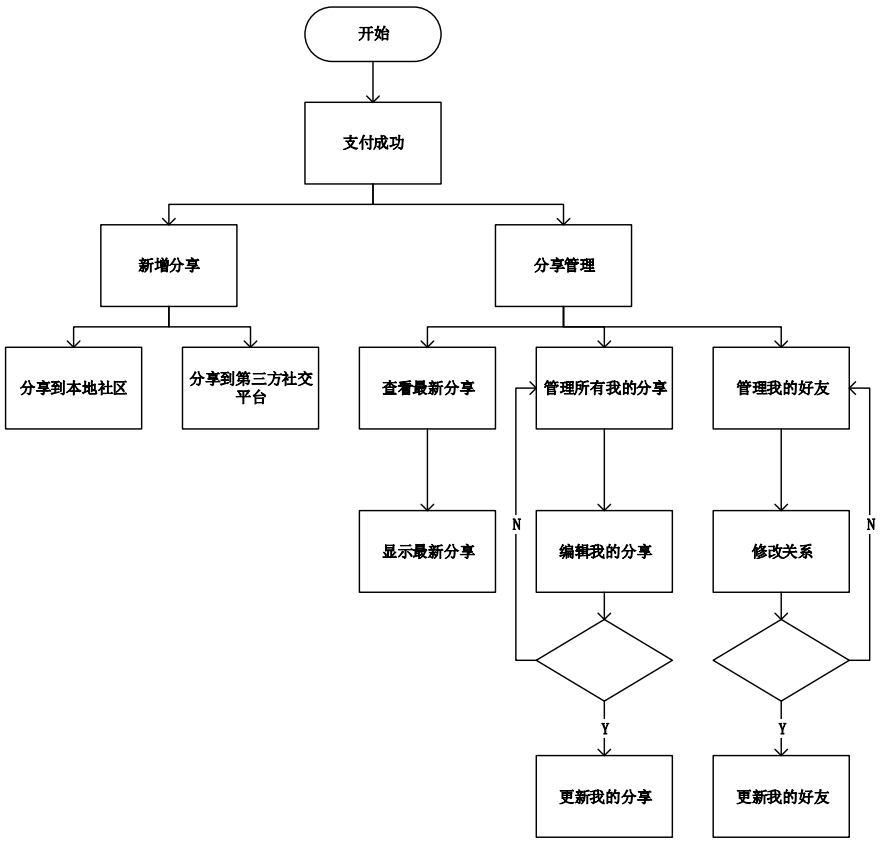

图 5-31 社区分享模块流程图

5.1.5.2 社区分享模块时序图

1. 新增分享时序图

图 5-32 描绘了用户新增分享的过程。首先用户在 MyFrameActivity 上请求获取所有未分享但已支付的订单，然后对系统返回的订单选择要分享的并请求分享，系统收到信息后，弹出分享界面，用户填写分享信息并设置完查看权限，然后选择分享的平台，最后点击确认分享。系统收到确认分享信息后，将新上传的分享信息写入数据库，并跳转到 MyShareListView 界面上，返回用户的所有分享信息。同时如果用户有好友，且未屏蔽的情况下，系统将会通知其他好友，推送该条新的分享。

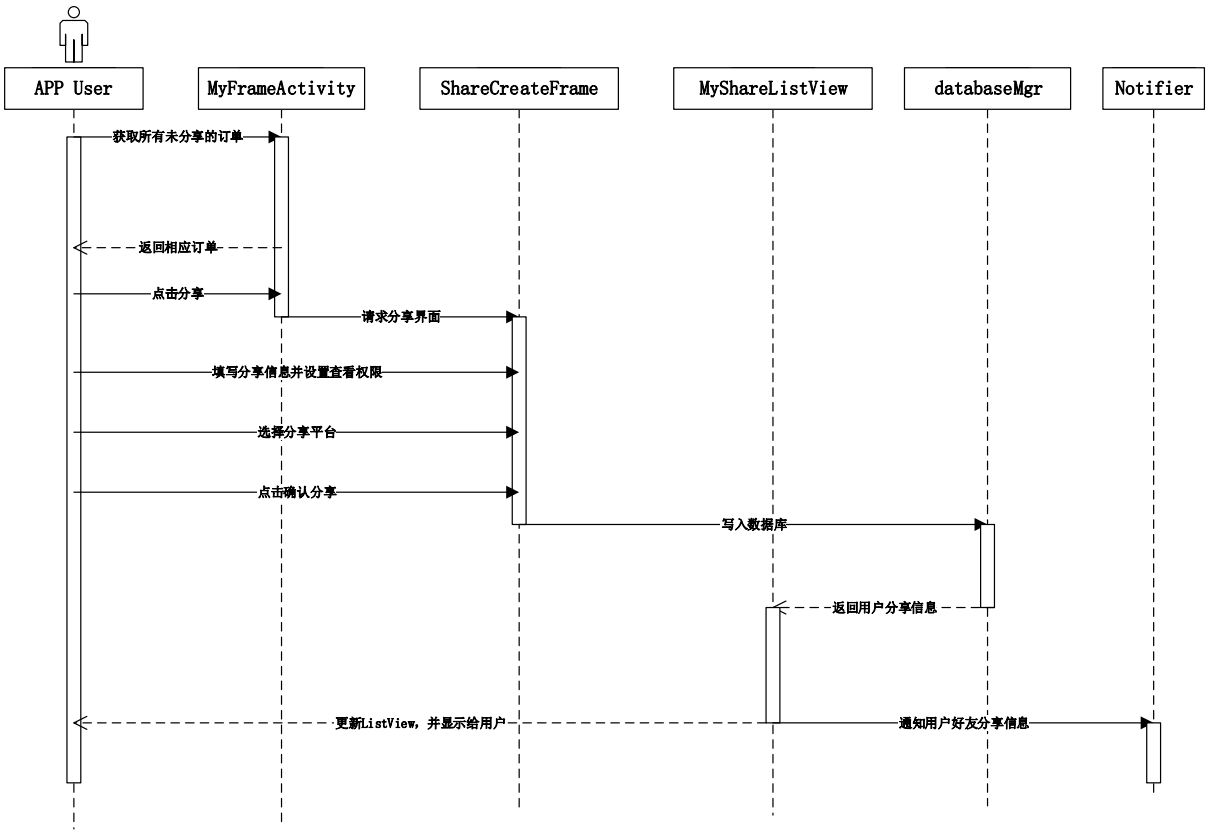

图 5-32 新增分享时序图

2. 管理分享时序图

图 5-33 描绘了用户管理自己所有分享的过程。首先用户在 MyShareListView 界面上请求获取所有分享，系统返回所有分享信息，并更新界面。对应某一个分享下的评论，用户可以选择回复，系统将用户的回复写入数据库，成功后更新用户界面，并通知回复的对象。如果用户请求删除某一个分享，系统将会同步数据库的信息，并返回更新后的视图。用户也可以管理所有的好友关系，如添加好友关系，删除好友关系。

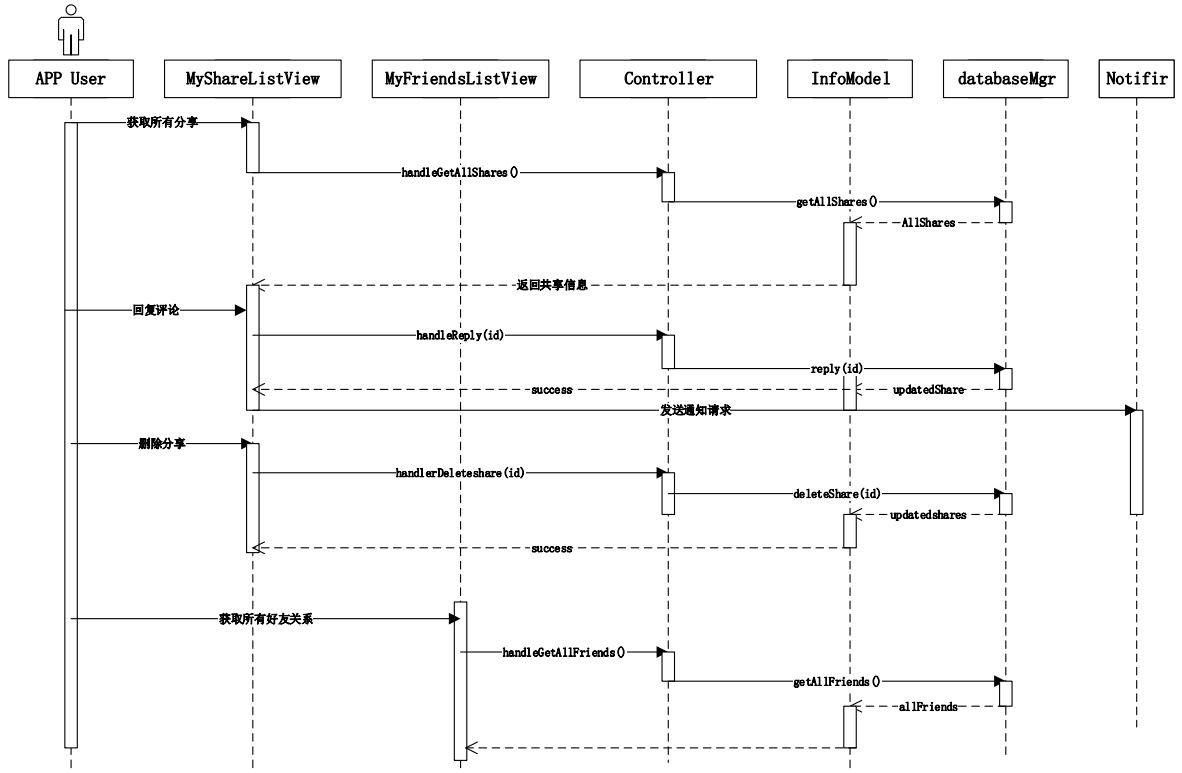

图 5-33 管理分享时序图

5.1.5.3 社区分享模块表结构设计

1. 好友信息表 tab_friends

表 5-8 好友信息表 tab_friends

字段名	类型	允许空	描述	备注
ID	INTEGER	N	对每一个好友的识别 id	主键
RIGHT	INTEGER	N	好友权限	

注：表 tab_friends 存储好友信息。

2. 分享表 tab_share

表 5-9 分享表 tab_share

字段名	类型	允许空	描述	备注
ID	INTEGER	N	对每一个分享的识别 id	主键
USERID	VARCHAR2(32)	N	用户 id	
CONTENT	VARCHAR2(128)	N	分享内容	
PHOTO	LONG FLOAT	N	分享图片	

表 5-9 分享表 tab_share<续>

COMMENT	INTEGER	N	评论	
LIKES	INTEGER	N	点赞数	
DISLIKES	INTEGER	N	反对数	
TRANSPARENT	INTEGER	N	是否公开	
CREATETIME	DATE	N	创建时间	
COMMENTRID	VARCHAR2(32)	Y	评论人 id	

注：表 tab_share 存储用户分享的具体信息。

5.2 系统工厂客户端功能模块设计与实现

5.2.1 订单管理模块

5.2.1.1 订单管理模块流程图

工厂端的订单管理模块是工厂后台系统的核心功能块，它直接关系到手机壳的制作与配送。工厂用户进入订单处理后，可以选择进入三个子菜单，分别是未制作，正在制作和已完成。这三个子菜单分别对应订单的不同状态，当手机端用户提交订单之后，系统默认当前订单为未制作状态，工厂用户进入未制作子菜单，系统先统计所有未制作订单的日期范围，然后按照日期先后顺序进行展示，用户选择某一日期，系统将会加载当前日期的所有订单中的手机壳图片，用户可以分页浏览所有的手机壳图片。待工厂用户确定好后，点击导出图片，系统会自动从服务器取出当前日期下的手机壳图案，并按照一定的规律进行命名，最后打包压缩，传给前端开始制作，此时当前日期的所有订单都将变为正在制作状态，且所有订单的手机壳产品将有未制作变成已导出状态。如果工厂用户需要导出以往订单的手机壳图案，点击导出以往订单，键入日期，即可重新导出。工厂用户进入正在制作子菜单，系统会统计所有正在制作订单的日期范围，用户可以调节日期范围，勾选手机型号来对正在制作的订单进行筛选，也可以直接输入订单号进行查询。订单从正在制作迁移到已完成需要对订单继续发货处理，具体的操作将在物流管理模块详细介绍。工厂用户进入已完成子菜单，系统会统计所有已完成订单的日期范围，同正在制作子菜单一样，用户可以调节日期范围，勾选物流状态来对已完成的订单进行

筛选。图 5-34 为工厂端订单管理模块的流程图。

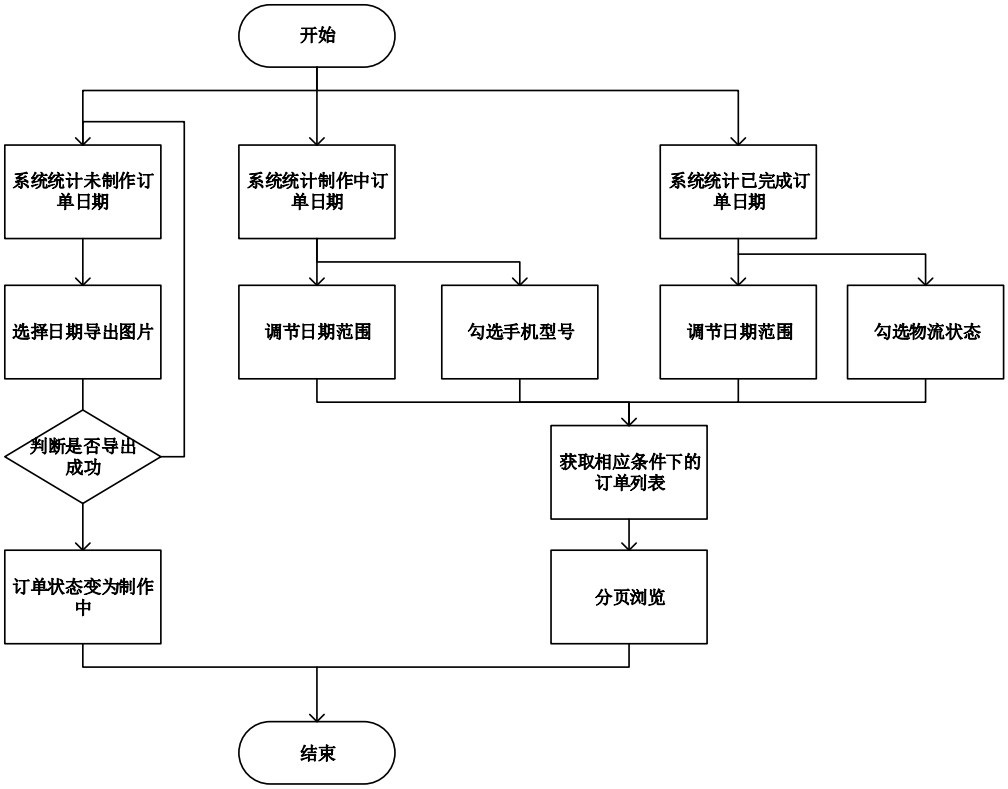

图 5-34 工厂端订单管理模块流程图

5.2.1.2 订单管理模块时序图

图 5-35 描绘了工厂不同角色的人共同管理订单的过程。目前工厂端的工作人员分为三类，分别是接单开始制作的人、发货人和售后员，他们分别处理不同状态下的订单。首先处理未制作人员请求获取所有未制作订单的日期，请求交由 Controller，调用数据库查询返回结果，然后处理未制作人员选择要导出手机壳图片的日期，系统返回当日所有未制作订单的图片并打包，这时工厂可以使用手机壳图片开始抛光打磨了。接着处理制作中人员发出获取所有正在制作订单的请求，系统返回订单信息，然后工作人员进行发货，发货成功后，系统将会通知处理已完成人员。

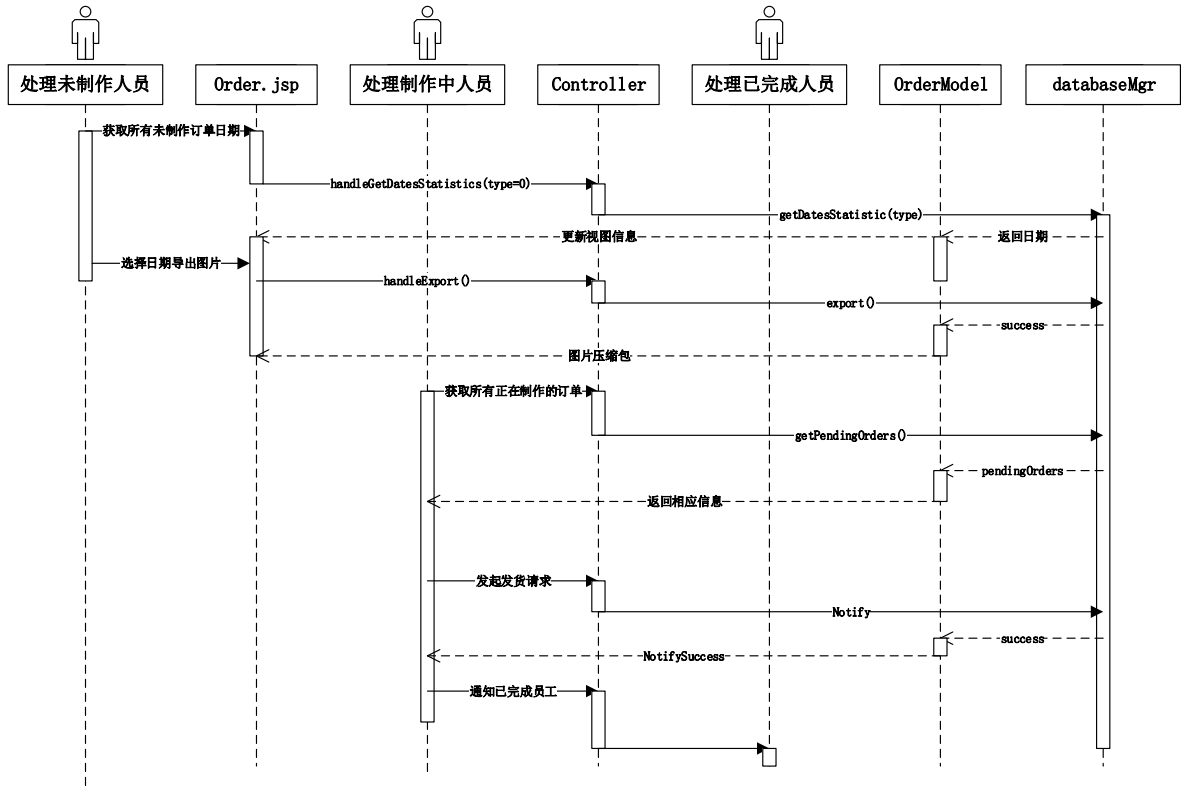

图 5-35 工厂端订单管理模块时序图

5.2.1.3 订单管理模块表结构设计

1. 收货人信息表 tab_ consigee

表 5-10 收货人信息表 tab_consigee

字段名	类型	允许空	描述	备注
ID	INTEGER	N	对每个用户的识别 id	主键
NAME	VARCHAR2(32)	N	用户名	
USERID	VARCHAR2(32)	N	用户 ID	
ADDRESS	VARCHAR2(128)	N	用户地址	
PHONENO	VARCHAR2(16)	N	手机号	
ZIPCODE	VARCHAR2(16)	N	邮编号	

注：表 tab_consigee 存储收货人的具体信息。

2. 订单信息表 tab_order

表 5-11 订单信息表 tab_order

字段名	类型	允许空	描述	备注
ID	INTEGER	N	对每个订单的识别 id	主键
USERID	VARCHAR2(32)	N	用户 ID	
CONSIGNAME	VARCHAR2(32)	N	收货人姓名	
ADDRESS	VARCHAR2(128)	N	收货人地址	
PHONENO	VARCHAR2(16)	N	收货人手机号	
ZIPCODE	VARCHAR2(16)	N	收货人邮编号	
STATUS	INTEGER	N	订单状态	
PRODUCTCNT	INTEGER	N	产品数量	
COST	LONG FLOAT	N	价格	
CREATETIMEY	INTEGER	N	订单创建年份	
CREATETIMEM	INTEGER	N	订单创建月份	
CREATETIMED	INTEGER	N	订单创建日期	
CREATEUTC	LONG FLOAT	N	订单创建 UTC	
TRACKNO	VARCHAR2(32)	Y	运单号	
EXPRESSNO	VARCHAR2(16)	Y	物流号	

注：表 tab_order 存储订单的具体信息。

3. 订单日程表 tab_orderschedule

表 5-12 订单日程表 tab_orderschedule

字段名	类型	允许空	描述	备注
ORDERID	VARCHAR2(100)	N	订单号	主键
TIMEBUY	LONG FLOAT	N	下单时间	
TIMECONFIRM	LONG FLOAT	N	确认订单时间	
TIMESHIPPED	LONG FLOAT	N	订单发货时间	
TIMERECEIVED	LONG FLOAT	N	订单签收时间	

注：表 tab_orderschedule 存储订单的重要日程信息，如发货时间，签收时间。

5.2.1.4 订单管理模块实现

1. 效果展示

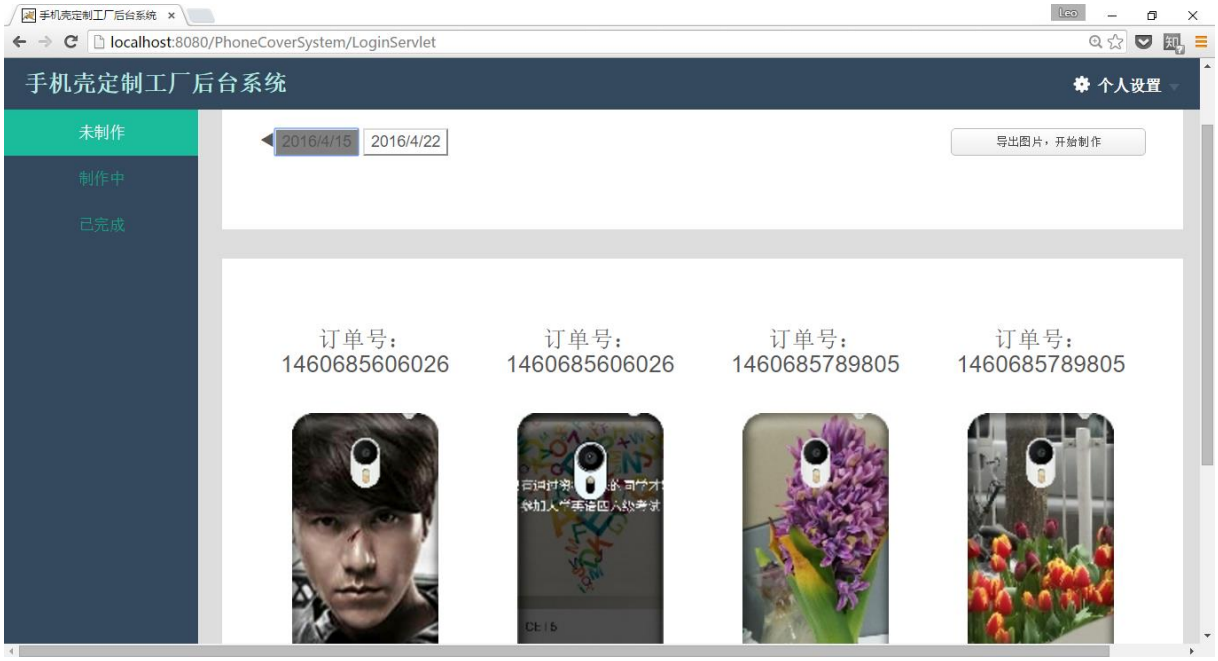

图 5-36 工厂端未制作页面 1

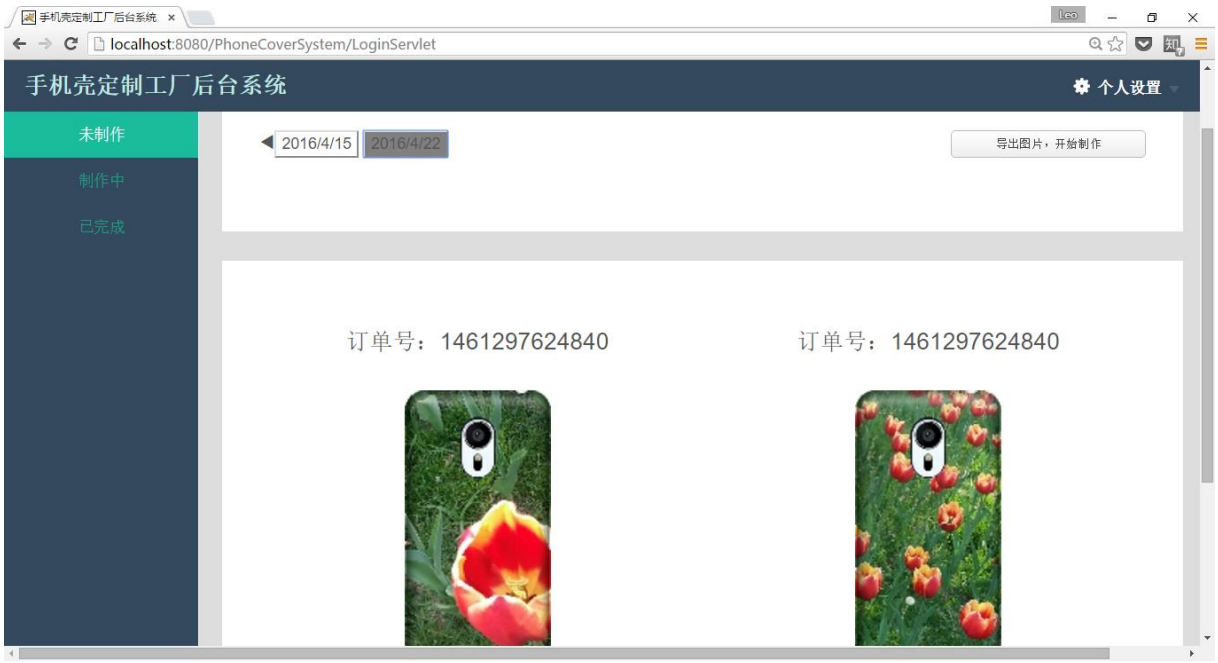

图 5-37 工厂端未制作页面 2

图 5-38 工厂端制作中页面 1

图 5-39 工厂端制作中页面 2

图 5-40 工厂端已完成页面 1

图 5-41 工厂端已完成页面 2

5.2.2 用户权限管理模块

5.2.2.1 用户权限管理模块流程图

用户权限管理模块是工厂端客户端特有的一个模块，它限制了不同用户在系统中的行为，在一定程度上使系统运行井然有序，也保障了系统数据的安全。首先工厂端用户的账号密码都是加密的，每一个用户会被分配一个或多个角色，不同角色的工作职能不同，获取数据和进行操作的权限也不同。如 A 用户是制作员，那么他负责未制作区域的操作，如导出图片，查看未制作订单等，其他模块的操作，如发货，查看完成订单信息等是没有权限进行的。在必要的时候，系统支持越权限访问，但需要密钥进行配置，同时系统会在一定时间之后关闭该通信渠道，还原用户的权限。为了最大程度监督不同用户的操作，系统设置了日志功能，管理员可以对日志进行编辑查看。图 5-42 为用户权限管理模块的流程图。

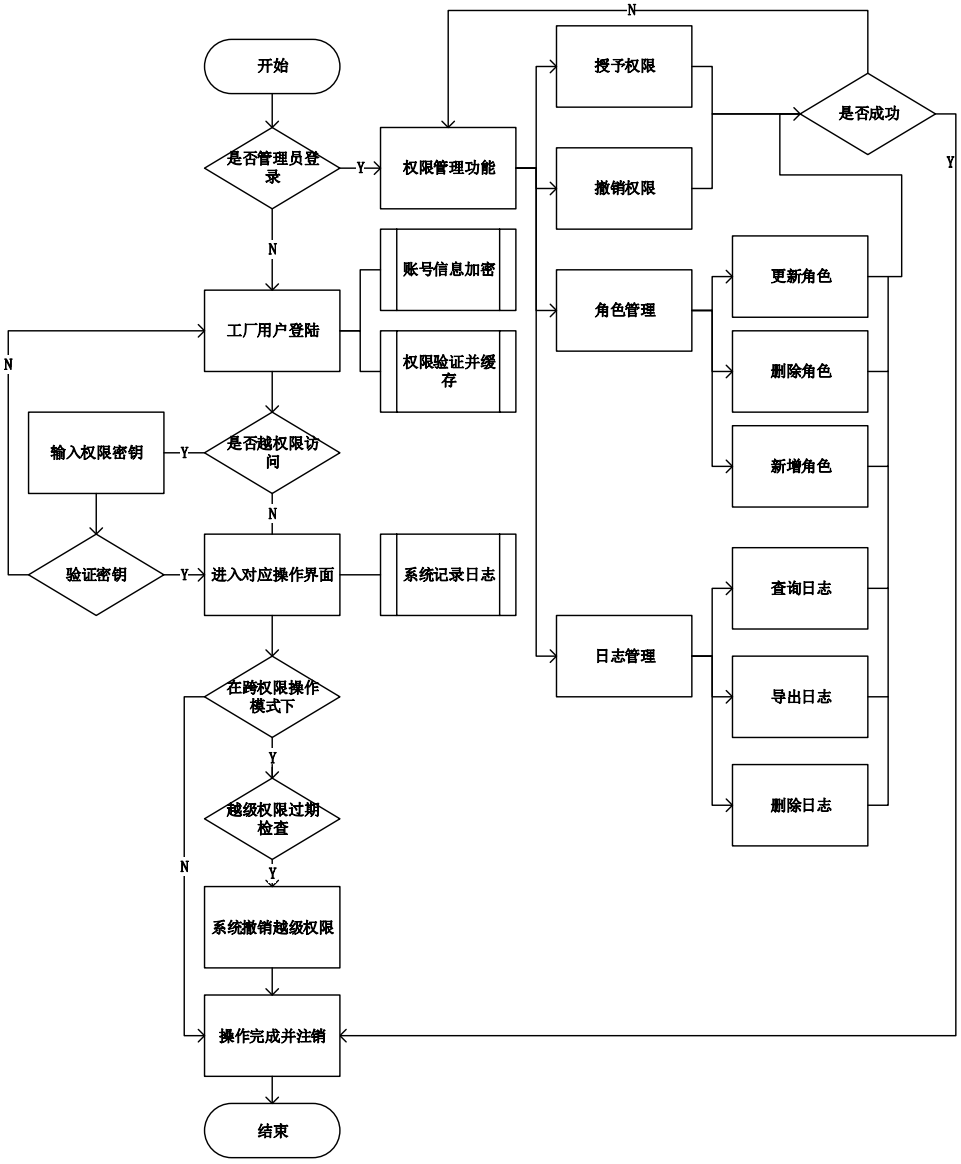

图 5-42 用户权限管理模块流程图

5.2.2.2 用户权限管理模块时序图

图 5-43 描绘了工厂端管理员管理用户权限的过程。首先工厂端管理员对不同的角色进行管理，可以为某一个用户添加角色或撤销角色，操作成功后，工厂端一般用户的浏览权限会更新。当一般用户登录系统时，会查询数据库，根据角色返回允许操作的内容，最终显示相应的界面。

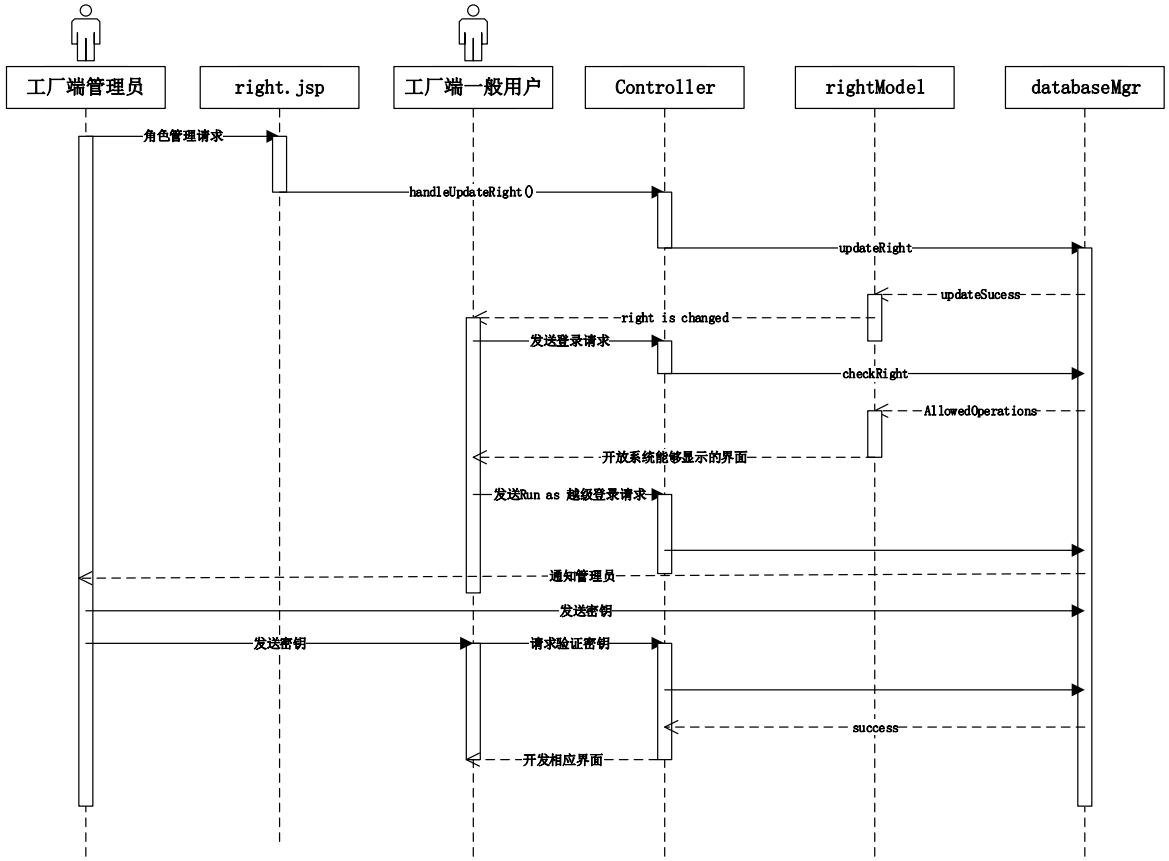

图 5-43 用户权限管理模块时序图

5.2.2.3 用户权限管理模块表结构设计

1. 权限信息表 tab_rights

表 5-13 权限信息表 tab_rights

字段名	类型	允许空	描述	备注
ID	INTEGER	N	对每一个权限的识别 id	主键
NAME	VARCHAR2(32)	N	权限角色名称	
OPERATION	VARCHAR2(68)	N	该角色允许的操作	

5.2.3 数据统计模块

5.2.3.1 数据统计模块流程图

数据统计模块顺应大数据的流行趋势，对手机壳销售情况进行阶段性的统计，目前系统支持月份数据统计、季度数据统计和年份数据统计。统计指标分别为手机型号，不同地区和用户好评。工厂用户进入统计功能页，选择统计时间跨度和统计指标，系统会根据条件，将服务器返回的数据已饼状图、条状图和表格的形式呈现给用户。用户可以通过该模块迅速了解手机壳的销售情况。图 5-44 为数据统计模块的流程图。

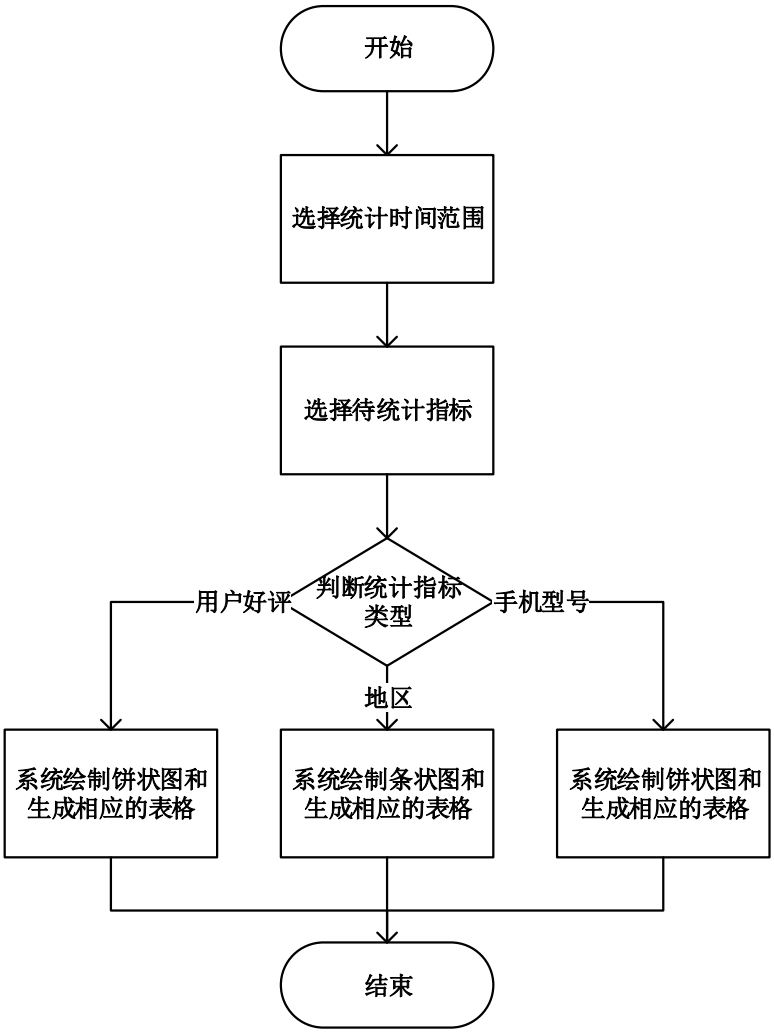

图 5-44 数据统计模块流程图

5.2.3.2 数据统计模块时序图

图 5-45 描绘了对所有订单数据进行统计的过程。首先用户需要选择统计周期和统计指标，提交请求后，系统会根据条件计算参数，并返回结果到界面。如果用户变更某一条件，系统会重新请求数据，并再次计算后更新界面显示。

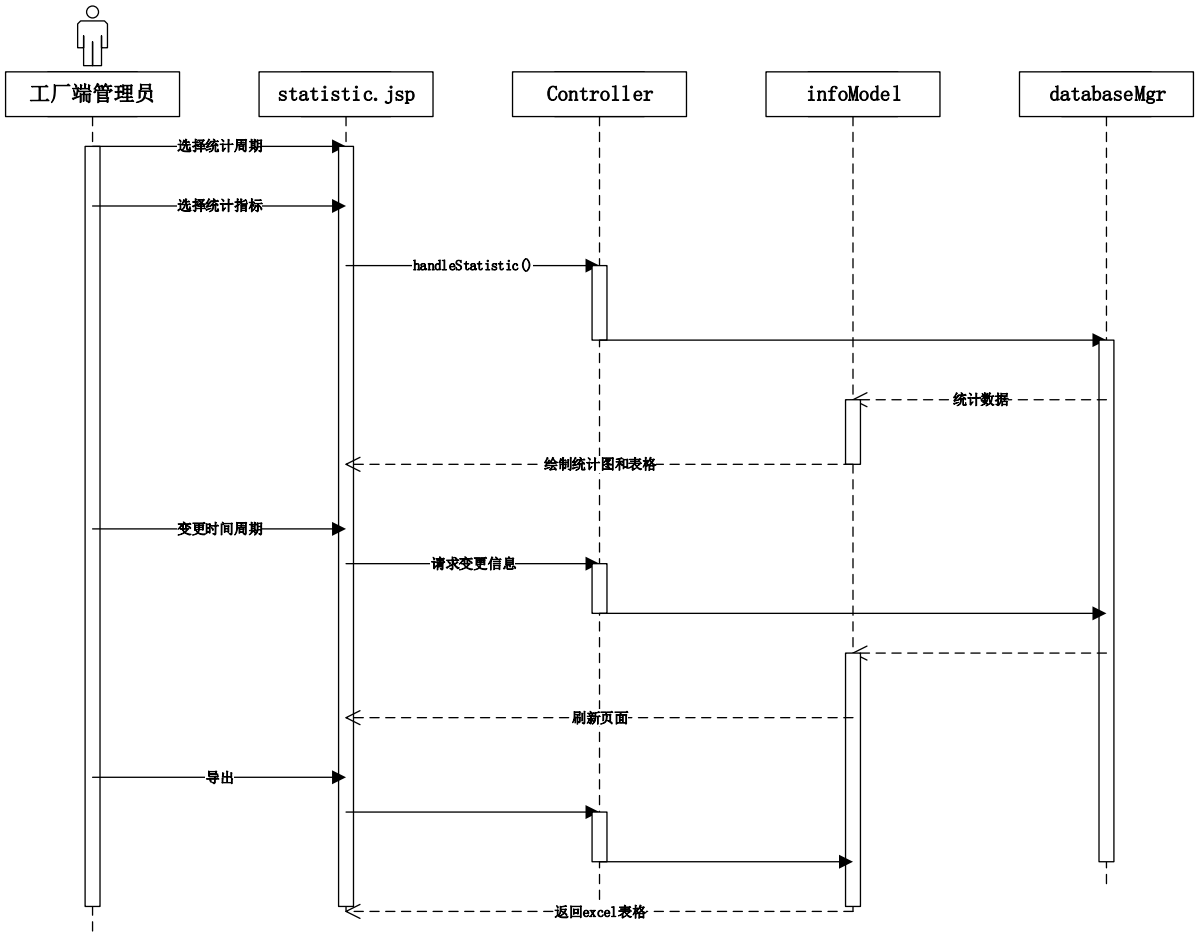

图 5-45 工厂端数据统计模块时序图

5.2.4 物流管理模块

5.2.4.1 物流管理模块流程图

物流管理模块是即订单管理模块之后的，工厂客户端的又一核心功能块。该模块为制作完成的订单提供发货，查询物流状态和跟踪物流信息的服务。授权的工厂用户在制作中子菜单页面上可以进行发货操作，首先通过系统获取快递公司，然后系统生成有效的运单号和物流号，点击发货，系统将会与第三方物流服务器进行交互，发出揽件请求。后期物流状态的更新和查询，都将由系统从第三方服务器端获取，并返回给本地服务器。

图 5-46 是物流管理模块的流程图。

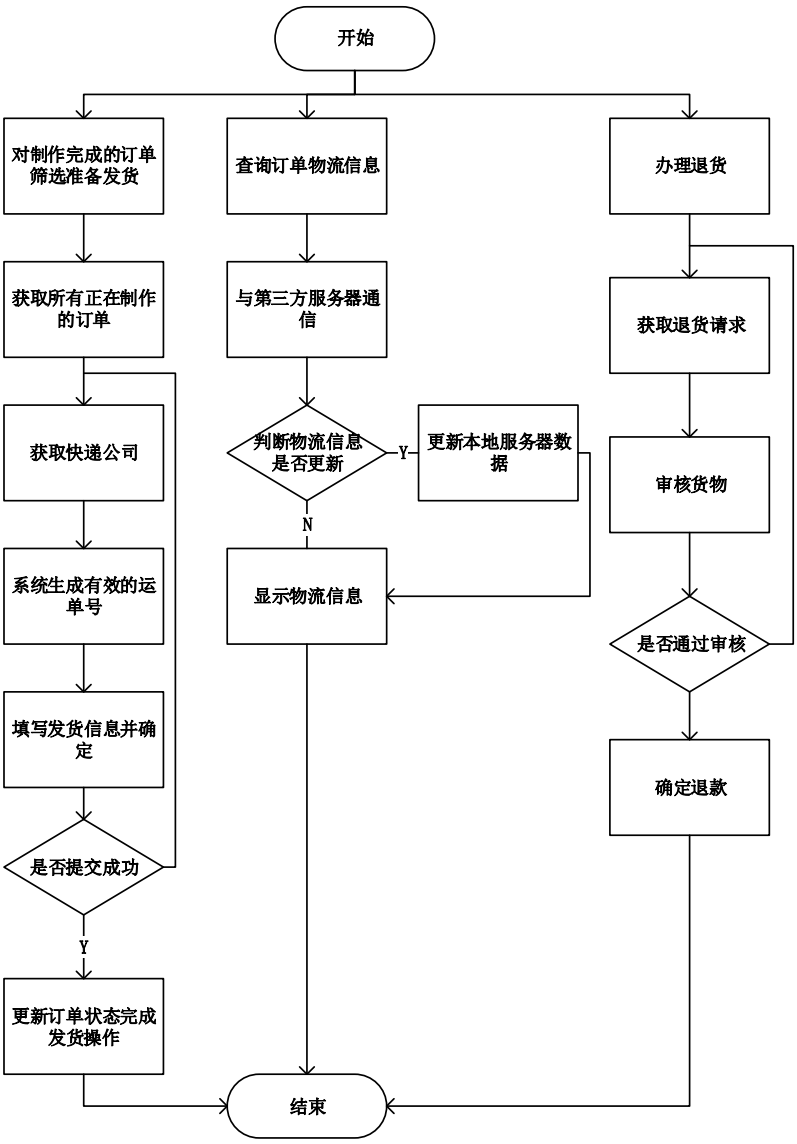

图 5-46 物流管理模块流程图

5.2.4.2 物流管理模块时序图

1. 发货时序图

图 5-47 描绘了发货的处理过程。用户请求获取所有制作完成的订单，然后选择要发货的订单，填写快递公司信息、发货信息并确定发货，系统处理完成后会返回发货成功的信息。

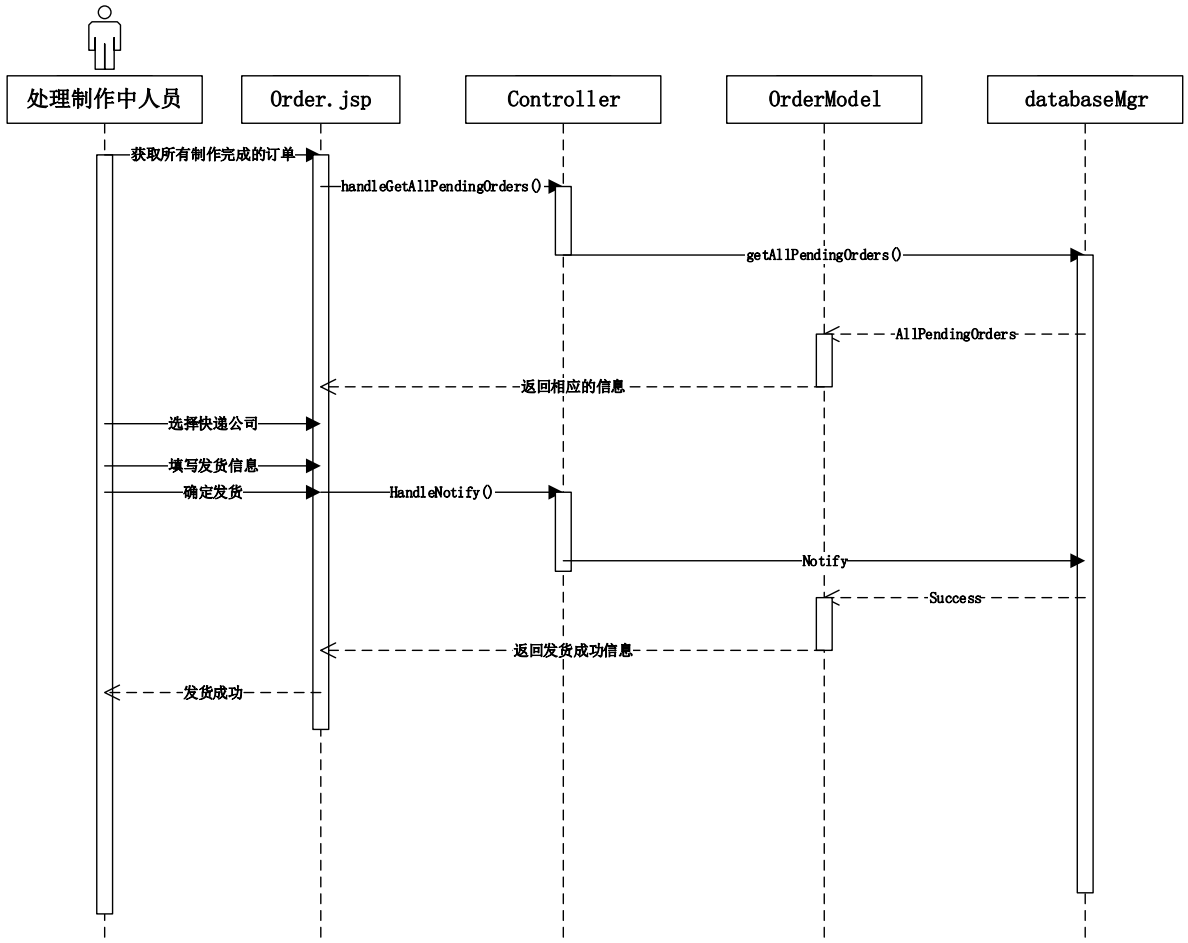

图 5-47 发货时序图

2. 办理退货时序图

图 5-48 描绘了后台用户处理退货的请求过程。该业务的办理由处理售后人员和手机端用户共同协作完成。首先处理售后人员获取所有的退货请求，如果收到的退货的确是残次品，则发起退款请求。手机端用户收到款项后，需要进行确认，确认完毕后，退货流程完毕。

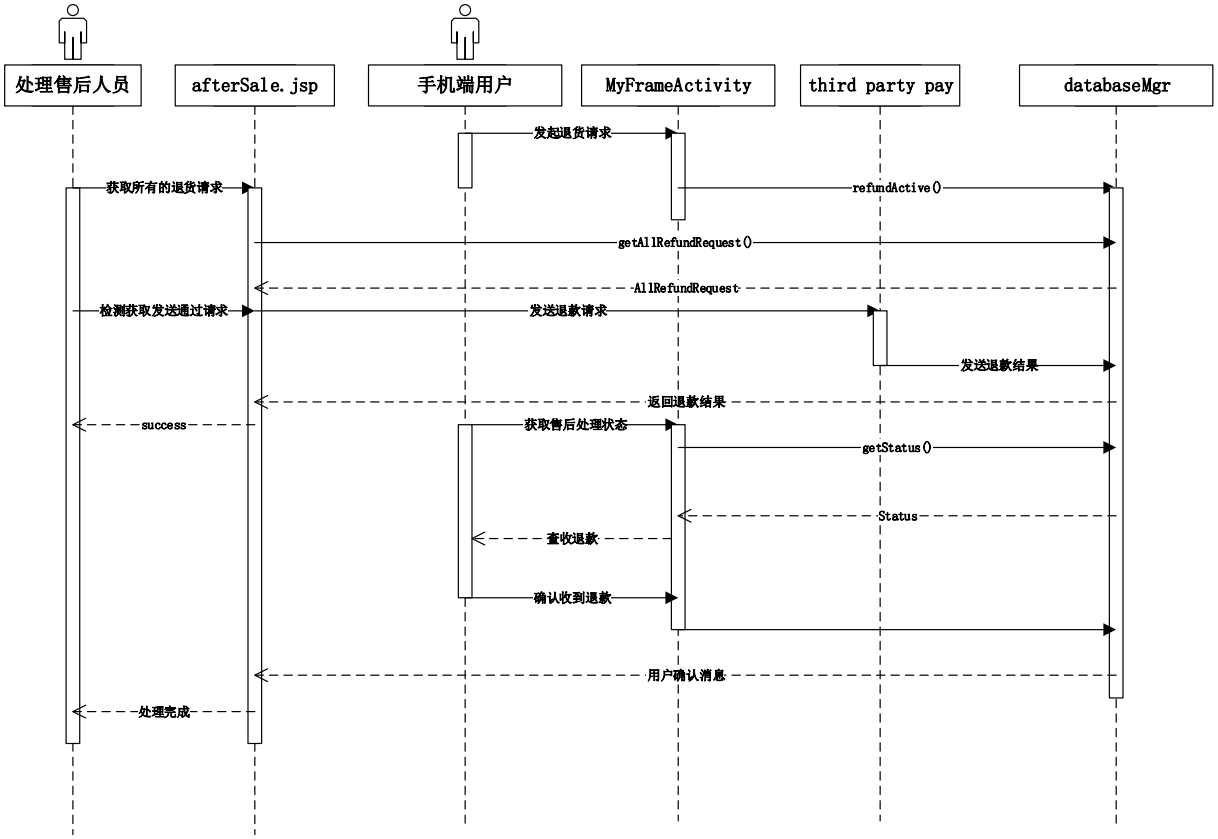

图 5-48 办理退货时序图

5.2.4.3 物流管理模块表结构设计

1. 物流信息表 tab_expressinfo

表 5-14 物流信息表 tab_expressinfo

字段名	类型	允许空	描述	备注
ID	INTEGER	N	对每一个物流信息的识别 id	主键
TRACKINGNO	VARCHAR2(32)	N	物流号	
SUCCESS	INTEGER	N	是否成功	
REASON	VARCHAR2(255)	N	失败原因	
STATE	INTEGER	N	物流状态	
TRACKSTEP	INTEGER	N	所处阶段	
UPDATETIME	LONG INTEGER	N	信息更新时间	

2. 物流路径表 tab_expresstrace

表 5-15 物流路径表 tab_expresstrace

字段名	类型	允许空	描述	备注
ID	INTEGER	N	对每一个物流路径的识别 id	主键
TRACENO	VARCHAR2(32)	N	物流路径号	
ACCESTATION	VARCHAR2(255)	N	接受站名	
REMARK	VARCHAR2(255)	N	接受意见	
ACCEPTTIME	DATE	N	接受时间	

3、发货信息表 tab_notify

表 5-16 发货信息表 tab_notify

字段名	类型	允许空	描述	备注
ID	INTEGER	N	对每一个发货信息的识别 id	主键
NOTIFYTIME	DATE	N	发货时间	
NOTIFYTYPE	VARCHAR2(64)	N	发货类型	
NOTIFYID	VARCHAR2(64)	N	发货运单号	
SIGNTYPE	VARCHAR2(16)	N	签名类型	
SIGN	VARCHAR2(128)	N	签名信息	
OUTTRADENO	VARCHAR2(64)	N	发货失败号	
SUBJECT	VARCHAR2(255)	N	主题	
PAYTYEP	VARCHAR2(4)	N	支付类型	
TRADENO	VARCHAR2(64)	N	物流号	
TRADESTATUS	VARCHAR2(32)	N	物流状态	
GMTCREATE	DATE	N	交易发起时间	
GMTPAYMENT	DATE	N	交易支付时间	
SELLEREMAIL	VARCHAR2(128)	N	商家邮箱	
BUYEREMAIL	VARCHAR2(128)	N	买家邮箱	
SELLERID	VARCHAR2(32)	N	商家 id	
BUYERID	VARCHAR2(32)	N	买家 id	
PRICE	LONG FLOAT	N	价格	

表 5-16 发货信息表 tab_notify<续>

TOTALFEE	VARCHAR2(64)	N	总费用	
QUANTITY	SHORT INTEGER	N	产品数量	

5.2.4.4 物流管理模块实现

1. 效果展示

图 5-49 工厂端发货界面

图 5-50 发货成功，查看物流信息

6 总结

本毕设论文以我在魅族科技有限公司的实习项目“魅印”手机壳定制系统为载体，从该系统的需求分析到架构分析再到详细设计，使用图文兼有的方式进行了细致的描述。论文的绪论部分通过分析现有O2O平台的优势和手机壳定制C2B销售方式的不足，引入了O2O的手机壳定制系统开发的必要性和可用性。可以发现开发所用的技术和理论都是常用的，为了让非专业的人士也能够更加清楚了查阅本文，在理论和技术支持模块对所有用到的技术进行了简要的讲解。在论文的后续章节部分，主要对系统的模块和开发过程进行阐述。从系统的需求分析到系统的架构设计，再到系统模块设计与实现效果展示，涵盖了系统开发过程的全部内容。

具体到需求分析模块的内容，该部分对整个系统的九个模块，分别绘制了对应的用例图，并给出了相应的用例描述。在章节的最后，对系统的非功能性需求，即质量需求也进行了概括和分析。

具体到系统架构分析模块的内容，该部分首先对开发设计存在的约束条件进行了阐述，然后详细叙述了项目性能要达到的规范要求，最后从设计模式、总体设计和数据库设计三个方面对系统架构进行了详细的建模展示。

具体到模块设计与实现的内容，该部分对系统的九个模块，分别绘制了对应的流程图，时序图，并展示了后台数据库的表结构，最后对重要模块的实现类图进行展示，并附上了最终的实现效果。

该系统目前已经上线，但在用户使用的过程中，也不断在发现新的问题，并需要进一步的完善和升级。我相信手机壳定制系统会在以后的维护过程中，变得越来越好。

当然了，在论文的撰写过程中，我也发现自己的很多不足，在不断的修改和完善的过程中，提高了自己排版，绘图和叙述的能力。希望自己在以后的研究生学习中，更加严格的要求自己，比现在的自己更加的优秀。

参考文献

[1]廖腾宏.O2O 的最终目标--实现 C2B 和无界服务.决策与信息,2015,0(3);42-43

[2]艾瑞咨询.2015 中国社区 O2O 行业研究报告.中国连锁,2016,0(2);66-70

[3]申晋祥[1].Android 系统中的布局研究与实现.山西大同大学学报：自然科学版,2014,30(5);15-17

[4]沈美[1].浅析在 Android 系统中 JSON 和 GSON 的用法.电脑编程技巧与维护,2014,0(24);81-83

[5]靖伟.Ajax 技术的研究与应用.中国传媒大学学报：自然科学版,2015,22(6);50-55

[6]林振文[1].网页美工中 jQuery 技术的应用研究.计算机光盘软件与应用,2015,18(1);224-225

[7]陈员义.基于 Bootstrap 响应式 Web 前端研究.福建电脑,2015,31(12);72-73

[8]薛鹏飞[1].Bootstrap 在 Web 前端开发中的应用与研究.科技致富向导,2015,0(14);254-254

[9]刘孝国[1],田晶[2],李太浩[1].基于分布式网络负载服务器设计研究[J].制造业自动化,2012,34(15):22-25.

[10]刘平; Android 手机访问服务器的一种数据交互方法[J];电子设计工程,2010,(09):96-98,102.

[11]张义[1],彭科[1],王国银[1].计算机网络通信安全中数据加密技术的研究与应用.中国新通信,2015,17(21);70-70

[12]于志刚.数据库 MD5 加密系统的设计与实现.科研,2015,0(49);178-178

[13] Mourtzis D, Doukas M, Psarommatis F. A multi-criteria evaluation of centralised and decentralised production networks in a highly customer-driven environment. CIRP Ann Manuf Technol 2012;61(1):427–30.

[14] Stillstro¨m C, Jackson M. The concept of mobile manufacturing. J Manuf Syst 2007;26:188–93.

[15] Hibernate Framework, JBoss; 2013. URL: www.hibernate.org.

致　　谢

在论文撰写的过程中，我遇到了不少困难，感谢那些帮助我，给我提各种建议的人。

首先，我要衷心的感谢我的毕设导师 XXX，XXX 老师利用平时的休息时间给了我很大的帮助和指导。从选题的确定到中期答辩的把关，再到论文内容的审核和修改，她给我提了很多建议，并在论文撰写的过程中，主动与我沟通，帮助我把整套论文顺利完成。

然后我要感谢实习公司的同事和领导，是你们热心的帮助和经验的传授，让我收获良多，并顺利完成我的毕设项目——手机壳定制系统的开发实现。

最后，在大学四年，所有帮助过我的老师和同学，我都要真诚的感谢。你们的情谊将是我以后永远美好的回忆。

附　　录

附录 A 用户定制与生产网络设计的移动端应用——外文原文

Mobile apps for product customization and design of manufacturing networks

Abstract

Manufacturing enters a new era, where companies, exploiting mass customization practices, base their business on mobility and customer integration in product design. These two utterly important activities can be supported by applications deployed on mobile devices, namely apps. However, apps in the manufacturing domain have yet to be widely adopted. Towards that end, the proposed work focuses on the integration of the customer in product personalization, and aims to support the design of manufacturing networks on the move, through the development of apps for Android devices. The applicability of the developed mobile apps in an automotive pilot case is presented.

Keywords: Mass Customization, Personalization

1. Introduction – State of the art and motivation

Nowadays, manufacturing industries are called to rapidly and effectively respond to ever-changing market demands generated by globalization, economy and customer saturation towards mass produced products [4]. Reasonably accurate forecasts and build-to-stock policies are no longer viable solutions, even for traditional mass producing industries like automotive [1]. Mobile and decentralized decision-making on core company activities is becoming the standard practice enabled by Information and Communication Technology. Moreover, value cocreation and customer integration in product design [2] arise as promising means towards capturing the market's pulse and supporting bridging the gap between mass production and mass customization [3].Mobile technology evolves rapidly; the last decade the use of mobile applications has outpaced traditional PC-based web-browsing. Nearly 70% of all Internet users access it using

smartphones and tablets [6]. The exploitation of mobile devices and applications (apps) is eventually finding its application in a scientific context [7,10]. Apps are being developed for biomedical research [7], photovoltaic installations [8], calculation of sealant O-ring dimensions [9] and pharmaceutics [11]. In the manufacturing sector, a number of mobile agent-based systems have been reported [12–15], together with some consumer-oriented apps [16,17]. However, the adoption of apps focused on core manufacturing processes remains limited [18]. The EU funded project Apps4ME envisages the creation of knowledge enriched apps for supporting various phases of the product and factory lifecycles, ranging from the requirements collections phase, to the product design phase, and up to the short-term production scheduling and even execution monitoring [20]. The necessary components of apps in order for them to be fully leveraged in manufacturing are presented in [19], where architecture, development, infrastructure, security, portfolio and privacy issues are investigated. In the short-term, such "role-based" apps can boost productivity by 5–10% as experts believe [17].

Although the term mobility may be interpreted as movement of physical manufacturing resources [21], here it is defined as "the ability of a stakeholder to remain constantly and interactively connected to the company's fundamental activities and make remote decisions".

Motivated by the above, the proposed work presents two mobile apps integrated on the back-end with a web based platform that share product design and production information. The first app concerns the customer integration in the products design phase and allows product personalization and visualization over mobile devices. The second app concerns the inter-organizational interaction of the company. It provides information regarding the company's partners and the design of efficient manufacturing networks for accommodating customized orders. It aims to enable the OEM to remotely manage the configuration of manufacturing networks in order to serve mass customization satisfying cost, time, quality, reliability and flexibility performance indicators.

2. Mobile apps description

The overall framework and workflow of the developed apps is depicted in Figure 1. The apps are the mAR (mobile Augmented Reality) and the mDEMAP (mobile Decentralized Manufacturing Platform).

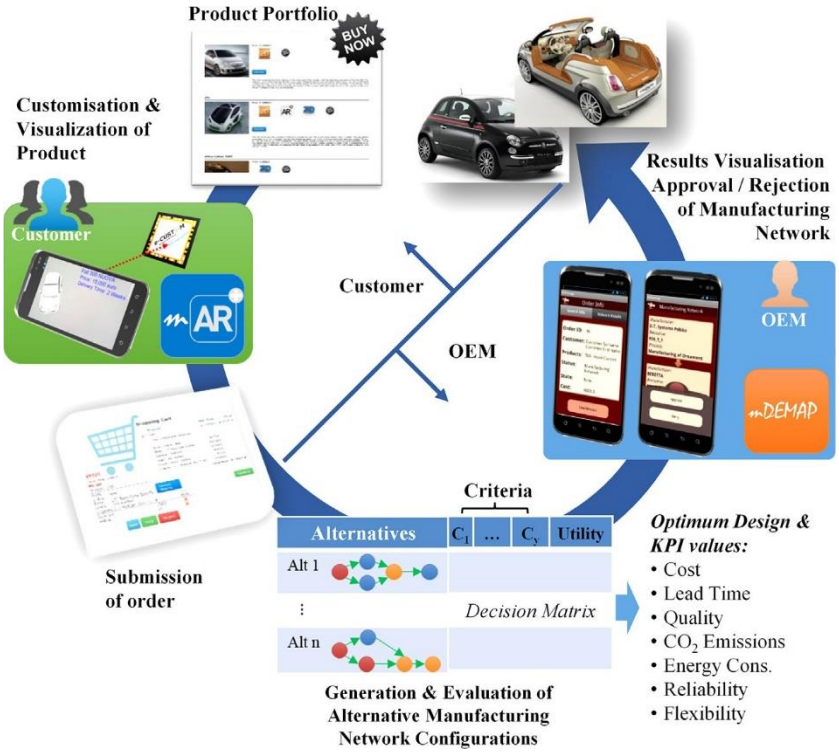

Figure 1. Framework of the mDEMAP and mAR apps.

The OEM logs into the application and is presented with a context menu. Primarily, a list of the manufacturing facilities of suppliers, dealers and OEM Plants is visualized. New facilities can be added and existing ones can be modified. The information included under each facility includes contact details (phone, email, etc.) and facility capabilities (type, number, capacity and processing capabilities). The mobile services allow the OEM to contact each partner by telephone call, e-mail or SMS message. The location of each facility is manually selected and stored through the integrated Google Maps API and is utilized during the evaluation of the manufacturing network alternatives for the calculation of travel distance and deriving environmental impact and lead time by the mDEMAP decision making algorithms. The location of each plant is shown with a flag on the map and directions are offered to the OEM from his current location to the plant through the OS's built-in GPS navigator (Figure 4). The activity diagram of Figure 2 depicts a workflow of stepwise actions provided to end-users.

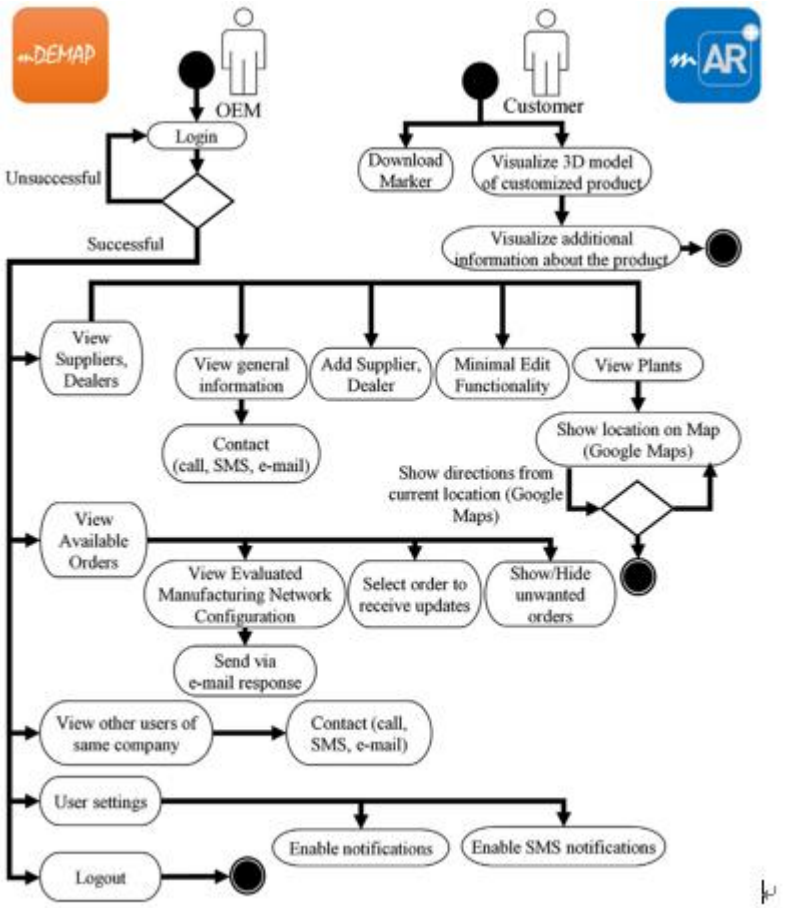

Figure 2. UML activity diagram of the mDEMAP and mAR apps.

Moreover, the OEM can visualise all the submitted orders. Once a specific order is selected, information such as selected customisation options, estimated cost and customer location, along with the criteria values of the optimum manufacturing network configuration become available. The evaluation of the alternative manufacturing network configurations is handled by the underlying webbased platform described in [3,5]. The best/"good" network according to the selected decision making criteria and their weights is visualised as a sequence of operations. For each operation, the facility, the resource assigned to each task and the transportation routes are calculated. The OEM can approve/reject the generated manufacturing network configuration. For important orders, the OEM can also receive notifications regarding their status and environmental impact. Finally, a set of miscellaneous settings are programmed such as enabling location services and enabling/disabling notifications and managing the list of other subscribed users (stakeholders, customers, etc.). The mAR app allows the customer to customise a selected product using the 3D configurator and then visualise the product using the AR viewer. The application provides a fiducial marker associated to each customised product, required for image tracking, which is handled by the AR libraries. Except for AR visualisation, mAR overlays additional information about the product name, customisation

price and delivery time in the scene, which remain visible while rotating the device (Figure 4).

3. Software design and implementation of the mobile apps

Both applications are developed for mobile devices with ARM-based processors, 512 MB minimum memory, 200 MB or more free storage, Android 2.3.3 (Gingerbread) or later and Google Play Services.

The software design was performed using Unified Modelling Language diagrams (UML 2.4.1). Eclipse J2EE v3.6 was used for programming and debugging of the apps, based on a 3-Tier architecture (Figure 3), which includes the data, business and presentation layers. The presentation layer consists of the Graphical User Interfaces (GUIs) of the app. Moreover, the application components of the developed apps, which are essential building blocks of any Android application, fit in the Business Layer, which handles the information exchange between Data and Presentation. A Service and a Broadcast Receiver are used for establishing a connection between the apps and the Google Cloud Messaging (GCM) that handles the apps' notifications. Finally, the Data Layer achieves data retrieval from the database through a web service following the SOAP protocol and integrates the apps at the back-end. Finally, the queries and results' retrieval are handled by the Hibernate framework [22].

The AR application was developed with Unity 3D [23] and UART [24], a set of plugins for Unity 3D that allow users to easily develop and deploy AR applications. The free library Vuforia [25] was used for the creation of the customised marker and for image recognition.

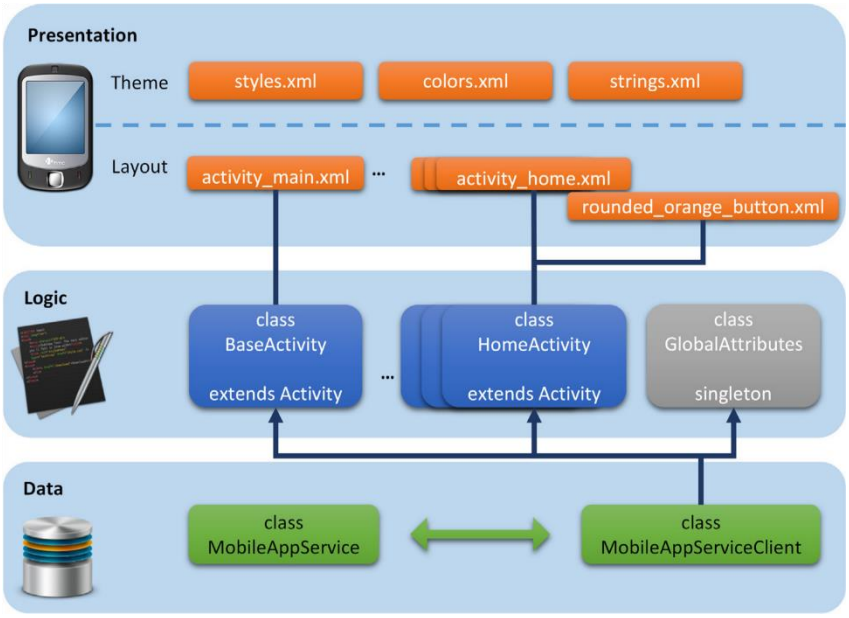

Figure 3. The 3-Tier architecture of the developed mobile apps.

Figure 4. Screenshots of the mDEMAP and mAR apps.

4. Industrial pilot case application

The applicability of the apps in a real industrial context is currently being tested to a major European automotive manufacturer. The apps have been installed to personal smartphones of the company's employees who are testing it by assuming different roles (customer, OEM) and report software bugs and other functional issues. The testing, which is based on a formal process targeted at end-user functionality testing, is undergone for almost one year now. The pilot application focuses on a typical production engineering scenario, where the OEM receives a customised product order from the customer and designs the manufacturing network to carry out the production. The details of the partners of the manufacturing network (equipment, capabilities, KPIs, locations) and of products (Bill of Materials, Bill of Processes, parts and subassemblies) are also synchronised with the web-based collaborating software platform described in [3].

5. Conclusions and future work

The apps developed in this research work support two fundamental business functions. Firstly, the mAR app enables the intuitive integration of the customer in the design phase of highly customised products and secondly, mDEMAP, enables the design of the manufacturing network to handle the production and transportation of the customised product for a customer.

A limitation of the proposed apps is the requirement for Internet connection in order for communication with the server to be established and for the data to be synchronised, however, the constant expansion both in speed and coverage of wireless and 3G data networks is envisioned to diminish such issues.

Future work will focus on the direction of exploiting the grid and cloud technologies for distributing the computation burden. The alternatives' generation and evaluation services will run on a computer grid, and the decision-making procedure results will be aggregated, and visualised through the mobile device alleviating the need for massive local processing power, thus allowing a full decoupling of the mDEMAP

app from the underlying web-based platform.

Acknowledgement

The work reported in this paper has been partially supported by the EC funded FP7 project “e-CUSTOM – A web-based collaboration system for mass customisation” (260067).

References

[1] Mayer H. Supply chain planning in the German automotive industry. OR Spectrum 2004;26:447–70.

[2] Vargo SL, Maglio PP, Akaka MA. On value co-creation: a service systems and service logic perspective. Eur Manage J 2008;26(3): 145–52.

[3] Hu SJ, Ko J, Weyand L, ElMaraghy HA, Kien TK, Koren Y, Bley H, Chryssolouris G, Nasr N, Shpitalni M. Assembly system design and operations for product variety. CIRP Ann Manuf Technol 2011;60(2):715–33.

[4] Mourtzis D, Doukas M, Psarommatis F. A multi-criteria evaluation of centralised and decentralised production networks in a highly customer-driven environment. CIRP Ann Manuf Technol 2012;61(1):427–30.

[5] Mourtzis D, Doukas M, Psarommatis F. Design and operation of manufacturing networks for mass customisation. CIRP Ann Manuf Technol 2013;63(1):467–70.

[6] Accenture. Mobile web watch survey; 2012.

[7] Young HA. Scientific apps are here (And more will be coming). Cytokine 2012;59:1–2.

[8] Fernandez-Pacheco DG, Molina-Martmnez JM, Ruiz-Canales A, Jimenez M. A new mobile application for maintenance tasks in photovoltaic installations by using GPS data. Energy Convers Manage 2012;57:79–85.

[9] Mobile ‘‘apps’’ provide parts data and calculate O-ring dimensions. Sealing Technol 2012;(2):3.

[10] Nee AYC, Ong SK, Chryssolouris G, Mourtzis D. Augmented reality applications in design and manufacturing. CIRP Ann Manuf Technol 2012;61(2):657–79.

[11] Williams AJ, Ekins S, Clark AM, Jack JJ, Apodaca RL. Mobile apps for chemistry in the world of drug discovery. Drug Discov Today 2011;16(21–22):928–39.

[12] Shen W, Hao Q, Yoon HJ, Norrie D. Applications of agent-based systems in intelligent manufacturing: an updated review. Adv Eng Inform 2006;20:415–31.

[13] Cimino M, Marcelloni F. Autonomic tracing of production processes with mobile and agent-based computing. Inf Sci 2011;181:935–53.

[14] C.Y. Huang, C. Pattinson, Using mobile agent techniques for distributed manufacturing network management. In: Proceedings of the 2nd annual symposium of postgraduate networking conference (PGNET'01), Liverpool, UK; 2001.

[15] Huang CY, Kai Cheng K, Holt A. An integrated manufacturing network management framework by using mobile agent. Int J Adv Manuf Technol 2007;32(7–8):822–33.

[16] Katz J. Mobile apps break into manufacturing. Industry Week 2012. http://www.industryweek.com/companies-amp-executives/mobileapps-break-manufacturing.

[17] M. Lunani, Role-based mobile apps transform manufacturing; 2013. http://www.cognizant.com/insights/perspectives/role-based-mobileapps-transform-manufacturing [accessed 08.08.13].

[18] Clevenger NC. IPad in the enterprise. Developing and deploying business applications. Indianapolis: Wiley; 2011.

[19] Gro¨ger C, Silcher S, Westka¨mper E, Mitschang B. Leveraging apps in manufacturing. A framework for App technology in the enterprise. Proc CIRP 2013;7:664–9.

[20] Apps4aME Project, Engineering apps for advanced manufacturing engineering, FP7 FoF.NMP.2012-6, Grant Agreement No: 314156.

[21] Stillstro¨m C, Jackson M. The concept of mobile manufacturing. J Manuf Syst 2007;26:188–93.

[22] Hibernate Framework, JBoss; 2013. URL: www.hibernate.org.

[23] Unity 3D; 2013. URL: http://unity3d.com.

[24] Unity AR Toolkit; 2013. URL: https://research.cc.gatech.edu/uart.

[25] Vuforia; 2013. URL: http://www.qualcomm.com/solutions/augmented-reality.

附录 B 用户定制与生产网络设计的移动端应用——翻译

用户定制与生产网络设计的移动端应用

摘要：制造业进入了一个新的时代，现在公司开始发掘大众自定义的做法，在产品设计上将他们的业务基于移动性和顾客的共同创作上。部署在移动设备上的应用有力地支撑着这两个重要的活动。然而，移动应用在制造领域还没有被广泛的采用。一直到文章最后，论文的主要焦点放在顾客在产品个性化的集成，旨在支持这样的制作网络设计工作的推进工作，通过开发设计安卓设备上的移动应用，论文通过一个开发完成的自动飞行器的移动应用的可用性来展示这一点。

关键字：定制、大众自定义、个性化

1. 介绍

如今，制造业需要快速高效地应对不断变化的市场需求，这些需求由全球化，经济和顾客对大规模生产的产品的满意度产生。一般来说精确的预测和按照库存生产的策略已经不再是可行的了，即使对于传统的大众制造行业如汽车。对公司核心活动的移动而非集中的决策正在演变为由信息和通信技术支持的标准做法。而且，价值共创和用户参与设计产品的方式崛起为抓住市场命脉和沟通大众制造和大众定制之间间隙的桥梁。

移动技术发展迅速，最近 10 年移动应用的使用超过了传统 PC 端网页浏览应用。接近百分之七十的互联网用户使用智能手机或平板。对移动设备和应用的使用在科技领域最终找到了应用的一席之地。为生化研究等各种行业的应用相继被报道，同时也有一些面向消费者的应用。但是，这些以制造过程为核心的应用还是很有限的。欧盟筹资的一个项目，扩大了移动应用在创作过程中各个领域的应用范围，从需求的收集，到产品的设计，到短期生产日程指定，甚至执行监管。对于这样的基于角色的应用，在短期类，能够提高百分之五到百分之十的生产率。

尽管移动这个术语可能会被解释为实际制造资源的移动，但是在这里它是一种让利益相关人保存参与公司重要活动和制定远程决策的能力。

收到以上的激励，本论文展示两个移动应用，它们基于网站平台，共享产品设计和生产信息。第一个应用让顾客参与到产品的设计过程，允许产品自定义和通过移动设备实现虚拟预览。第二个应用设计公司不同组织之间的沟通交互。它提供关于公司合作伙伴和处理自定义订单的制造网络高效设计的信息。它旨在让原始设备制造商能够远程的管理制造网的配置信息，为了满足大众自定义，成本，时间，质量，可靠性和灵活性等性能指标。

2. 移动应用描述

图一展示了该应用的整体框架和工作流。该应用是移动端的增强现实和移动端的非集中化的制造平台。原始设备制造商登录应用，会有一个内容菜单。首先，供应商、分销商、制造商的制造设备的列表会展示出来。新的设备可以添加，已存在的设备可以进行修改。在每一个设备下的信息包含联系详情（电话，邮件等）和设备的属性（类型，数量，容量，处理能力）。这个移动服务让原始设备制造商可以通过电话，邮件或短信与每一个合作方进行联系。每一个设备的地址都是人为选择和存储，通过 Google Maps 的 API，而且通过移动端的非集中化的制造平台的决策算法计算货运距离和环境影响因素和时间来选择和评估的时候，会被使用。每一个工厂的地址在地图上用一个小红旗标出，同时通过系统内置的 GPS 导航器，会有从原始设备制造商到工厂的路径方向。图二的活动图展示了提供给最终用户的过程流。

Figure 1. Framework of the mDEMAP and mAR apps.

而且，原始设备制造商能够查看看所有提交的订单。一旦选择了一个具体的订单，一些信息如自定义选项，估价，顾客位置还有最优的制造网配置的数值信息会被显示出来。可选的制造网络配置的评估是由底层的网站平台处理的。根据所选的决策标准和考虑因素而获得的最佳网络是以一组操作展现的。对于每一个操作，设施，分配给每一个任务的资源还有运输路径都会被计算。原始设备制造商能够同意或否决产生的制造网络配置。对于重要的订单，原始设备制造商能够收到关于订单状态，环境影响的通知。最后，一系列的设置如打开位置服务，打开或关闭通知，管理其他订阅用户列表等等也被实现。移动端的增强现实应用让顾客可以使用 3D 配置器来自定义一个产品，然

后通过 AR 展示器查看产品效果。该应用提供一个标签来关联每一个自定义的产品，是图片跟踪需要的，并交由 AR 库进行处理。除了 AR 查看的功能，移动端的增强现实还包含其他的信息如产品的名称，自定义价格，配送时间等，在旋转设备的时候也可以显示在界面上。

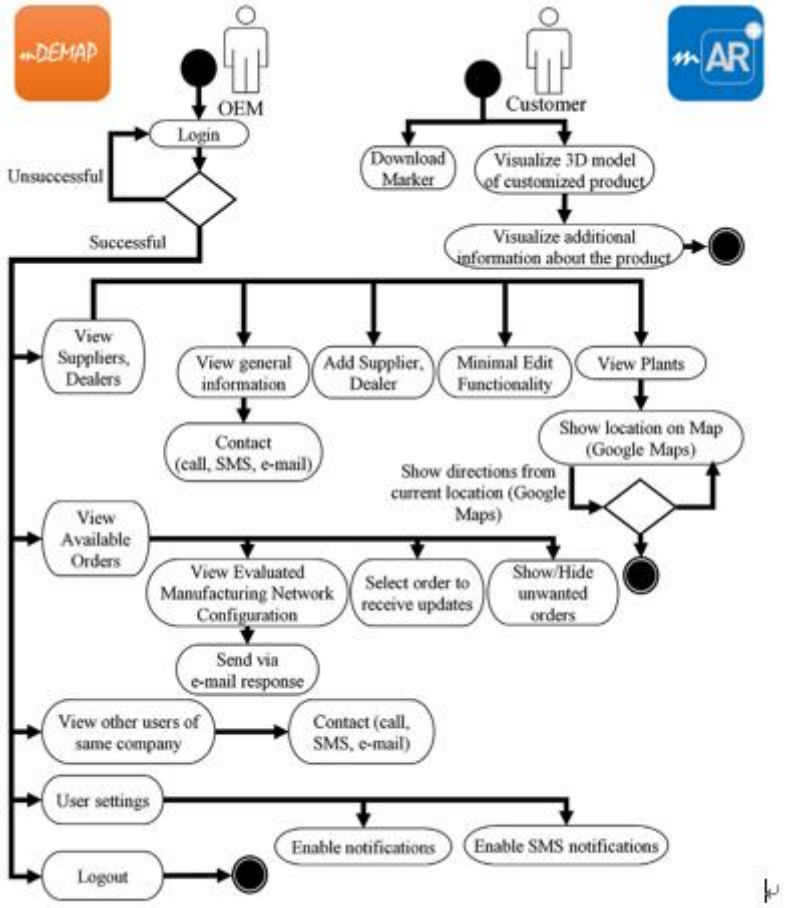

Figure 2. UML activity diagram of the mDEMAP and mAR apps.

3. 软件设计与实现

这两个应用都是为 ARM 处理器，512M 最小内存，200M 或有更多空间，Android2.3.3 以上和带有 Google Play Service 的移动设备开发的。

软件的设计使用的 UML，使用 Eclipse J2EE v3.6 进行开发实现，并基于三层架构（图三），该架构包含数据层，逻辑层和展示层。展示层包含图像用户界面。而且，软件最重要的部分在逻辑层，它负责处理数据层与展示层的信息交换。一个服务和一个广播接收器用来建立应用于 Google Cloud Messaging 建立连接，它是用来分发应用通知的。最后，数据层通过一个 web service 从数据库获取数据，它是基于 SOAP 协议的，数据的搜寻通过 Hibernate 框架实现的。

AR 应用使用 Unity 3D 和 UART 开发，它是一组 Unity 3D 的插件，方便用户开发和部署 AR 应用。免费的库 Vuforia 被用来创建自定义标签和图片识别。

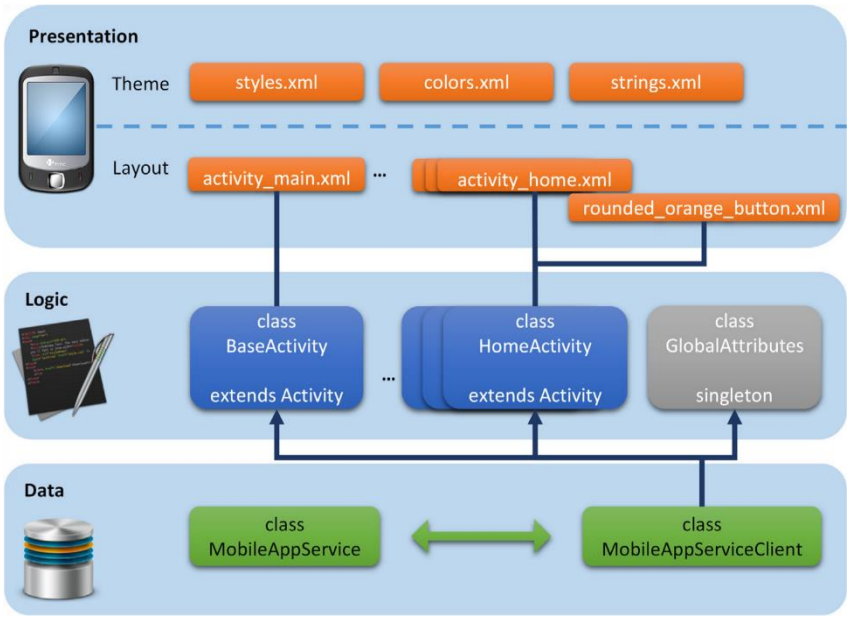

Figure 3. The 3-Tier architecture of the developed mobile apps.

Figure 4. Screenshots of the mDEMAP and mAR apps.

4. 工业案例应用

在真实工业场景下该应用正在由一家大型欧洲汽车制造商测试。该应用被安装在个人智能机上，测试它的公司员工扮演不同的角色（顾客，制造商）然后提交 bug 和其他功能性问题。该测试工作，是基于正式过程以最终用户的功能性测试为目的，到目前已经进行了快一年。该应用有一个典型的应用场景，当制造商收到一个来着顾客的自定义订单后，制造网络上所有可选的合作商的具体信息还有产品的信息会与网站平台进行同步。

5. 总结和以后的工作

在该论文中介绍的应用支持两个基本的业务功能。首先移动端的增强现实应用让用户能够参与到产品的设计，自定义个性化的产品。第二，移动移动端的非集中化的制造平台可以给用户提供最优的制造和配送方案。

但目前该应用也存在局限，需要联网后才能使用。由于无线网络的覆盖和各大运营商 3G 数据网的扩大，这已经不是一个问题了。

未来的工作将聚焦在发掘图与云技术下的分配计算下面。新的一代应用将会运行在电脑网格上，决策结果将会被整合并展现在移动设备上。这样就减轻了大规模本地处理器的需要，因此能够降低耦合，与底层的网站平台分离开来。

致谢

这篇论文的工作由 EC 资助的项目“e-CUSTOM”提供支持。