

基于 Android 的无线点餐系统设计

陈晓峰 林正浩

(同济大学 电子与信息工程学院, 上海 201804)

摘要: 近年来, Android 手机的巨大发展, 使得设计新式的无线点餐系统成为可能. 从客户端、服务器端全面讨论了该点餐系统的设计原理和过程, 包括客户端的登陆、点菜、更新、结账等模块, 以及服务器端的 web 服务器、数据库服务器、Web 工程和 MFC 工程等模块. 经实际测试和应用表明: 该系统具有较高的可靠性和实用性.

关键词: 无线点餐系统; Android; 客户端; 服务器端

中图分类号: TP 393.09 **文献标识码:** A **文章编号:** 1000-5137(2012)04-0369-05

0 引言

21 世纪是个科学技术“爆炸性”发展的世纪, 极具科技含量的手机行业更是以跨越式的姿态向前发展. 这其中不得不谈到 Google 公司推出的 Android 手机操作系统. 伴随着 iPhone 刮起的智能手机之风, 从 2008 年起, Android 系统攻城略地, 短短 3 年即超过塞班系统成为全球市场份额最高的智能手机操作系统. 截止 2012 年 2 月统计, 其市场份额已占领了全球半壁江山(全球份额为 52.5%, 在中国更是高达 76.7%, 远超仅有 11.8% 的塞班系统). 如此的风云剧变使得曾经的手机行业老大诺基亚利润大幅缩水, 匆匆忙忙放下塞班系统, 与 Intel 合作开发 MeeGo 系统推出 N9 智能手机, 结果市场反应相当平淡. 于是放弃 MeeGo, 与软件行业巨头微软合作, 推出搭载 Windows Phone 7 系统的 Lumia 手机, 但截止目前为止, 季度出货量依然徘徊在百万级, 远远落后于 iPhone 和 Android 手机. 而曾经是手机个性风向标的索爱和手机行业的鼻祖摩托罗拉的状况则更严重——爱立信的股份被索尼全额回购, 也许不久的将来它的手机名也要改为 Sony 而不是 Sony Ericsson 了; 摩托罗拉则以 125 亿美金的价钱被 Google 全额收购. 反观一直力推 Android 系统的三星和 HTC, 甚至包括中国的中兴、华为, 最近几年其手机出货量一直以两位数增长, 彻底瓦解了原来的诺基亚、索爱和摩托罗拉等老牌厂商的市场地位. 为什么这么短的时间, 可以让手机行业重新洗牌, 根本原因便在于 Android 系统的优异性能和免费开源.

再来讨论餐饮行业的无线点餐系统需求情况. 首先, “民以食为天”, 所以不管哪个城市, 都有各种菜系各种档次的餐馆饭店, 可见市场潜力是非常大的. 传统的点餐过程都是服务员人工完成的, 但近几年伴随着 IT 技术的发展, 也有各种新型的无线点餐系统出现, 如基于 Zigbee 网络的无线点餐系统, 但这种系统需要布局 Zigbee 网络, 并开发对应的手持终端, 成本高而且不方便, 因此大多数饭店仍然不愿意接受这种点餐方式.

既然那么多人手上有 Android 手机, 那么多人需要上餐馆吃饭, 那是否可以开发一款点餐系统, 成本和性能让餐馆可以接受, 同时用餐的客户也会觉得点餐更加便捷和时尚? 答案显然是可以的. 事实上, 只需一个无线路由器, 一台电脑, 和用户手上的一个手机, 就可以完成需求.

收稿日期: 2012-06-25

作者简介: 陈晓峰(1987 -), 男, 同济大学电信学院硕士研究生; 林正浩(1958 -), 男, 同济大学电信学院教授.

1 系统架构

首先给出本设计的整体架构如图 1 所示:

由图 1 可知,Android 手机的客户端软件通过 Apache Http 协议,经路由器访问 PC 机的服务端.PC 机必须配置 Web 服务器和数据库服务器.Web 服务器可以响应客户端的 Http 请求,并根据相应的请求访问数据库服务器,然后将获得的结果反馈给手机客户端.数据库服务器则用于存储菜单信息、订单信息等内容.



图 1 整体架构

2 手机客户端软件设计

2.1 Android 应用程序简介

Android 是一种以 Linux 为内核的开源操作系统,支持使用 Java 作为编程语言来开发应用程序(需下载 Android SDK 开发包). Android 应用程序由一个或多个组件组成^[1]:

(1) 活动(Activity)

Activity 是最重要的也是最频繁使用的组件,在 Android 的应用程序当中,Activity 一般代表手机屏幕的一屏.如果把手机比作一个浏览器,那么 Activity 就相当于一个网页.在 Activity 当中可以添加一些 Button、ListView、ImageView、TextView 等控件.当用户从主屏幕选择一个应用程序时,就会开始一个活动.所有和用户交互的事情都发生在这个组件.比如 2.2 节将要提到的登陆界面,即为一个 Activity.

(2) 服务(Service)

Service 跟 Activity 的级别差不多,它可以和其他组件进行交互,但只能后台运行. Service 用于任何需要持续较长时间的操作,例如网络监视或更新检查.比如后面将提到的更新菜单功能,即用 Service 来实现.

(3) 内容提供者(ContentProvider)

形象地讲,可以将 ContentProvider 看做应用程序的数据库管理员.Android 应用程序最常用的数据存储方式是 SQLite 数据库^[4],它是一个轻量级的数据库,非常适合于嵌入式开发.客户端软件的菜单信息和订单信息等,都会存储在 SQLite 数据库里.ContentProvider 提供了一个接口,使得别的程序(如 Activity)可以方便的和 SQLite 数据库进行交互.

(4) 广播接收器(Broadcast Receiver)

Broadcast Receiver 主要用于监听系统消息,并做出响应.

上面提到的 4 个组件都是相互独立的,共同构成一个 Android 应用程序,而各个组件的通信就是由 Intent 完成.Intent 起着一个媒体中介的作用,专门提供组件互相调用的相关信息,实现调用者与被调用者之间的解耦.

2.2 客户端主程序设计

图 2 给出了客户端主程序流程图,其中登陆、点菜、服务、结账、主菜单都是一个 Activity,各个 Activity 的跳转可以通过 Intent 来实现.

用户打开程序时,是登陆界面(LoginActivity),为了方便管理,用户名和密码均为餐桌号.为了正确登陆,程序需

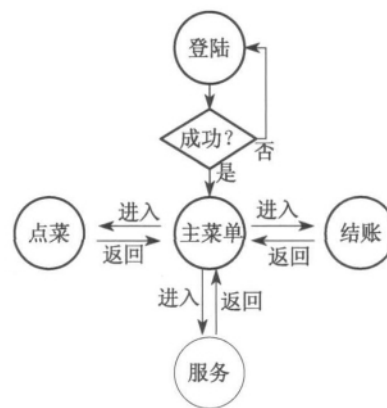


图 2 客户端主程序流程图

要访问后台数据库,Android 已经集成了 Apache 的 Http Client 模块,只需加入相应的驱动包,就能通过局域网络访问后台数据库.

登陆成功以后,进入主菜单(MainActivity).主菜单包括三大功能模块,即点菜(OrderActivity)、服务(ServiceActivity)和结账(CheckoutActivity).图 3 和图 4 给出了本设计的实际登陆界面图和主菜单界面图.

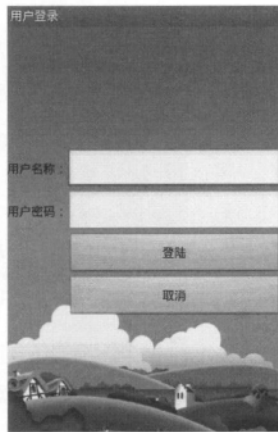


图 3 登陆界面

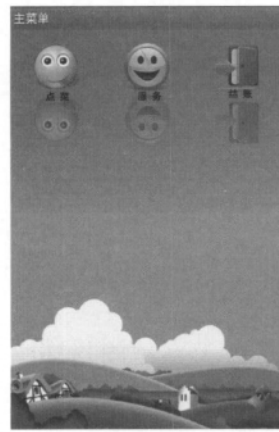


图 4 主菜单界面

下面分别分析点菜、服务和结账模块的实现.

(1) 点菜(OrderActivity)

点菜模块是本设计最重要的模块. UI 界面显示 2 个 Button,即更新和点菜.选择更新,则客户端通过 http 网络从服务器端获取菜单信息,并将信息存储至本地的 SQLite 数据库.选择点菜,则给出菜单列表,菜单列表又分为不同的样式,如饮料区、汤羹区、主食区等,让用户点菜.用户选择完后,即可提交订单,并自动返回主菜单.提交订单后,程序将把本次的点菜信息存储至本地的 SQLite 数据库,以方便后面结账时查询.菜单列表用 ListView 控件来显示,每个菜式包括图片、菜名、菜式简介、客户评价等信息.图 5 是实际设计的主食区菜单.

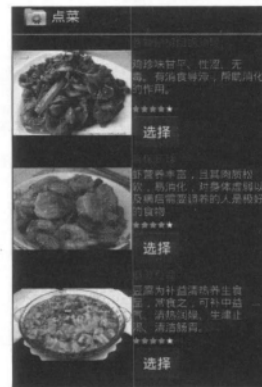


图 5 主食区菜单

(2) 服务(ServiceActivity)

该模块主要发生于用户用餐的过程中,需要查看已点菜单,或者要加菜,或者需要服务员的服务.具体的流程见图 6.

(3) 结账(CheckoutActivity)

选择结账,则 CheckoutActivity 通过 ContentProvider 访问 SQLite 存储的本次用餐信息,并将这些信息整理好,生成本次用餐的明细和费用,同时将该结果反馈给服务器.服务员接到通知,将去对应的桌号完成买单服务.

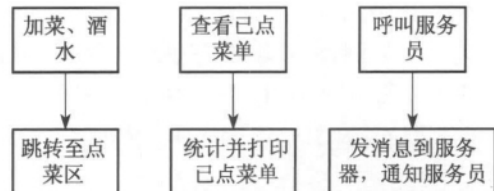


图 6 服务模块流程图

3 PC 服务器端实现

首先给出服务器端的总体设计如图 7 所示.

服务器端最关键的就是开发一个 web 工程(Java 语言实现,用 Eclipse 作为开发平台).开发这个工

程需要为 PC 机配置 web 服务器和数据库服务器^[2],并将两个服务器的驱动包加入到 web 工程.该 web 工程具体实现了如何响应客户端的 http 请求,如何连接数据库,获取数据,并将结果反馈给客户端,它是客户端与服务器的桥梁,也是 web 服务器和数据库服务器的桥梁.

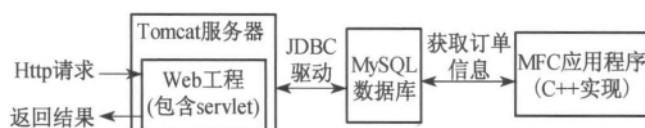


图7 服务器端总体设计框图

Web 服务器选用 Tomcat 服务器, Tomcat 是 Apache 软件基金会(Apache Software Foundation) 的 Jakarta 项目中的一个核心项目, 作为一个轻量级应用服务器, 它在中小型系统和并发访问用户不多的场合下被普遍使用. 简单地说, Tomcat 可以响应 html 页面的访问请求, 实际上就是通过运行 JSP/Servlet^[5] 来实现. 凭着免费、稳定、先进、易用的优势, Tomcat 得到了大部分软件开发商的认可, 成为目前比较流行的 Web 应用服务器.

当正确安装好 Tomcat 到 PC 机后, 则在本地的浏览器或者局域网内的任何移动设备的浏览器(包括手机) 输入“http://10.37.135.215:8080”, 就可以打开一个欢迎网页(10.37.135.215 为当前电脑的 IP, 也可以输入 http://localhost:8080, Tomcat 默认 localhost 变量指向当前电脑的 IP). 设计好 web 工程后, 将其放在 Tomcat 安装目录的 webapp 下, 就可以使用了. 例如 web 工程目录为 DinningServer, 登陆模块的 Servlet 为 LoginServlet, 则在客户端软件的 LoginActivity 模块访问服务器地址为“http://10.37.135.215:8080/DinningServer/LoginServlet? user=10&passwd=10”(这是餐桌号为 10 的用户登陆时程序内部将会使用的地址).

在本设计中, 所谓的 web 工程, 其实就是设计 Servlet. 那什么是 Servlet 呢? 如果把 Tomcat 比作是一座饭店, 那么 Servlet 就是里面的厨师, 要吃大餐, 那么你需要把你想要吃的东西告诉厨师, 然后厨师就会做出满意的菜了. 意思就是说 Tomcat 只不过是一个平台, 一个运行环境, Servlet 才是完成了请求的东西.

Servlet 的框架是由 2 个 Java 包^[6] 组成: javax.servlet 和 javax.servlet.http. 在 javax.servlet 包中定义了所有的 Servlet 类都必须实现或扩展的通用接口和类, 在 javax.servlet.http 包中定义了采用 HTTP 通信协议的 HttpServlet 类. 以 LoginServlet 为例, 它继承 HttpServlet 类, 并在内部实现:

```
String user = request.getParameter( "user" );
String passwd = request.getParameter( "passwd" );
```

从而获取登陆 URL 的用户和密码, 并与数据库存有的用户密码比较, 如果存在, 则返回成功信息, 反之返回失败信息.

接下来讨论数据库服务器. 数据库服务器选用 MySQL. MySQL 是一个关系型数据库管理系统. 由于其体积小、速度快、总体拥有成本低, 尤其是开放源码这一特点, 使得许多关注成本的中小型网站选择了 MySQL 作为网站数据库. 它为多种编程语言提供了 API. 这些编程语言包括 C、C++、Python、Java、Perl 等, 提供 TCP/IP、ODBC 和 JDBC 等多种数据库连接途径. 事实上用 java 编写的 web 工程就是通过 JDBC 访问数据库, 这需要将 MySQL 驱动包 mysql-connector-java-5.0.8-bin.jar 添加到 web 工程的 class-path 下.

MySQL 数据库初始的一些数据信息可以用 MySQL - Front 软件来建立, 它是一款小巧的管理 MySQL 的应用程序, 可以方便的建立、编辑、删除数据库和表格. 本设计需要提前建立的表有:

(1) 用户表(UserTable), 用于登陆系统.

(2) 菜谱分类表(MenuTypeTable), 这是为了将不同的菜式进行分类, 即前面提到的饮料区、汤羹区、主食区等.

(3) 菜谱表(MenuTable) ,用于保存每种菜谱的信息 ,包括名称、价格等.

而 web 程序需要动态建立另一种表 ,即订单明细表(DetailTable) ,每次客户用餐即建立一个表 ,存储所有本次用餐的相关信息 ,包括用餐日期、点菜内容、用餐费用等.

此外 ,为了使服务员和厨师能够实时了解每个客户都点了什么菜 ,或者老板想要统计当天的甚至是当月的营业额 ,还有必要设计一个 windows 下的 MFC 应用程序^[3]来完成这些任务 ,该程序用 visual C++ 开发 ,同样的 ,MySQL 提供了 C++ 语言的操作接口 ,建立工程时 ,只需把驱动包 MySQL Connector/ C++ 的安装目录添加到开发环境的 Additional Include Directory 即可. 限于篇幅 ,这里不详细讨论 MFC 程序的具体实现.

4 总 结

本文作者全面讨论了基于 Android 的无线点餐系统的设计与实现. 当然 ,该系统还有很多提升的空间 ,比如可以让用户每次用餐完后都对所点的各道菜做出评价 ,为了鼓励这项操作 ,评价的用户可获得优惠. 系统将每道菜的用户评价信息存储到数据库 ,用户更新菜单时将可以得到这些详细的评价信息. 这样既有利于餐厅了解自己菜式的优缺点和并进行优化提升 ,也可以让别的用户了解什么菜在本餐厅最受欢迎 ,从而更有选择性的点菜.

总之 ,无线点餐系统是一个方兴未艾的市场 ,基于 Android 的点餐系统将更富时代气息 ,更有优势 ,必然能获得越来越多的市场认可和应用.

参考文献:

- [1] 郭宏志. Android 应用开发详解[M]. 北京: 电子工业出版社, 2011.
- [2] 李兴华. Java Web 开发实战经典[M]. 北京: 清华大学出版社, 2010.
- [3] PROSISE J. MFC Windows 程序设计[M]. 北京: 清华大学出版社, 2007.
- [4] 白文江. 基于 Android 平台的移动应用开发研究[J]. 太原大学学报, 2011, 12(3): 3-4.
- [5] RICH G, COLIN M. Building a web-centric application platform on top of android[J]. IEEE Pervasive Computing, 2011, 10(4): 5-6.
- [6] CHRISTENSEN A S, MOLLER A. Extending java for high-level web service construction[J]. ACM Transactions on Programming Languages and Systems, 2003, 25(6): 2-3.

The design of wireless order system based on android

CHEN Xiao-feng, LIN Zheng-hao

(College of Electronics and Information, Tongji University, Shanghai 201804, China)

Abstract: In recent years, the tremendous development of the Android mobile phone has made the design of new wireless ordering system possible. This article discusses comprehensively from client side and server side the system's designing principles and the processes, including the launcher, a la carte, update, checkout and other modules in client side, as well as the web server, database server, web project, MFC project modules in server side. The actual testing and application show that this system has high reliability and practicality.

Key words: wireless order system; Android; client; server

(责任编辑: 包震宇)