

基于 Android 的农产品质量安全监管软件^①

程 涛¹, 杨晓国², 张春九³

¹(江苏农牧科技职业学院 农业物联网系, 泰州 225300)

²(泰州昊龙信息科技有限公司, 泰州 225300)

³(上海周雨通信技术有限公司, 上海 200120)

摘 要: 面向政府监管部门人员快速查询农产品质量信息和及时将巡查工作信息化登记的需求, 开发了基于 Android 的农产品质量安全监管软件. 介绍了农产品质量安全监管软件的功能模块, 创建了 Android 客户端和服务器的系统架构, 重点设计了客户端界面, 研究了数据读取和写入的关键技术. 该软件在江苏省泰州市农业监管系统应用, 运行效果稳定有效, 达到预期要求.

关键词: Android; 农产品质量安全; 软件设计

Agricultural Products Quality Safety Supervision Software Based on Android

CHENG Tao¹, YANG Xiao-Guo², ZHANG Chun-Jiu³

¹(School of Electronic Science and Technology, Jiangsu Agri-animal Husbandry Vocational College, Taizhou 225300, China)

²(Taizhou Haolong Information Technology Co., Ltd., Taizhou 225300, China)

³(Shanghai Zhouyu Communication Technology Co., Ltd., Shanghai 200120, China)

Abstract: To meet the Government supervision department personnels' demand of quickly checking the quality of agricultural products and timely registering the inspection, the supervision software of agricultural products quality and safety based on Android is developed. This paper introduces the function module of the supervision software of agricultural products quality and safety, creat the Android client and server system architecture, focuses on the design of the user interface of the client, and researches the key technologies of data reading and writing. The software was applied in agricultural regulatory system in Taizhou, Jiangsu Province. The application was stable and effective, and met the expected requirements.

Key words: Android; agricultural products quality safety; design of software

食品安全关系到广大人民群众的身体和生命安全, 关系到社会稳定和经济健康发展, 作为食品主要来源的农产品质量也越来越受到社会关注. 随着互联网的快速发展, 把互联网的创新成果和经济社会各领域深度融合, 推动了技术进步、效率提升和组织变革. 《国务院关于积极推进“互联网+”行动的指导意见》中明确指出“加快互联网与政府公共服务体系的深度融合”、“对生产经营过程进行精细化信息化管理”^[1]. 在此背景下, 项目组基于广泛使用的移动互联设备操

作系统 Android, 开发了手机终端农产品质量安全监管软件. 农产品质量监管部门的用户使用安装在智能手机上的农产品质量监管软件, 通过移动互联网, 读取农产品质量监管平台服务器保存的新闻动态、通知公告、基地信息、被检单位信息、生产基地自检信息、监管部门抽检信息、监管队伍信息、实地巡查登记信息等, 利用互联网提升农业监管水平和效能^[2].

① 基金项目: 2014 年度泰州市科技支撑计划项目(SSF20140118);2015 年度江苏省农业建设项目(SJC2015100230);2015 年江苏农牧科技职业学院科研课题(NSF201507-2)

收稿时间:2015-10-31;收到修改稿时间:2015-12-03 [doi:10.15888/j.cnki.csa.005216]

1 系统总体设计

1.1 功能模块

用客户端(即 Android 手机端)软件主要面向市、县(区)、乡(镇、街道办)农业质量监管部门的监管用户,包括了 8 个业务功能模块: (1)新闻动态: 系统已注册用户成功登录该软件之后, 可以浏览服务器提供的所有新闻列表和新闻详细内容; (2)通知公告: 浏览服务器提供的通知公告; (3)基地信息: 可以查询到在服务器由系统管理员注册的生产基地信息, 包括基地名称、所在地区、工商注册登记号、地址、联系人、联系方式等; (4)被检单位: 可以查询到在服务器由各级监管人员登记的生产基地信息, 作为巡查工作的对象, 是基地信息的有效补充; (5)基地自检: 查阅生产基地用户检测农产品农药残留后上传的检测信息; (6)抽检信息: 查阅生产基地、农贸市场、批发市场将农产品样品交给当地农产品质量监管站进行检测或者监管人员到实地抽样检测完成后上传的质量信息; (7)监管队伍: 查阅辖区内各级监管部门的监管人员信息, 包括姓名、工号、职位、联系方式等; (8)巡查登记: 监管人员对生产基地、农贸市场、批发市场等场所进行巡检时, 可以通过手机对检查场景进行拍照, 将巡检信息录入到手机端, 包括检查地点、检查项目、检查结果、巡检照片、自动获得的 GPS 地址, 即时上传到农产品质量监管平台服务器, 实现监管人员巡视工作的快速电子化登记。

服务器面向客户端提供相应的接口, 供数据读取和写入^[3]。另外服务器还包括了生产基地追溯管理系统、政府监管子系统、公共服务子系统、数据管理中心等功能模块, 系统功能模块结构图如图 1 所示。

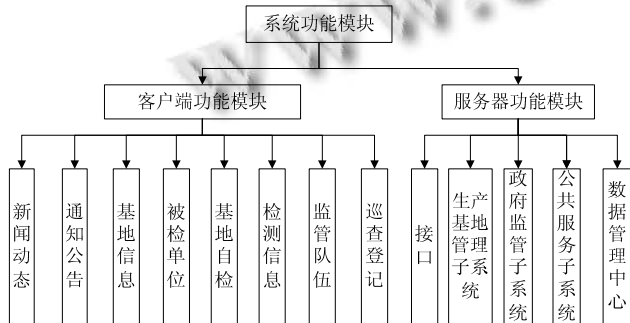


图 1 系统功能模块结构

1.2 系统架构

整个系统的架构采用 Client/Server(客户端/服务器)

架构, 其中 Android 手机端安装的农产品质量安全监管软件作为客户端, 采用 JavaEE 开发的农产品质量安全监管平台作为服务器, 面向客户端提供相应的接口用来读取和写入数据^[4]。

本系统的服务器采用 JavaEE 分层架构, 分为视图层、控制器层、业务逻辑层、数据访问层、持久化对象层。图 2 显示了 Android 客户端与服务器整合的架构。分层体系将业务逻辑、数据访问等放到中间层处理, 客户端不直接与数据库交互, 而是通过控制器与中间层建立连接, 再由中间层与数据库交互^[5]。

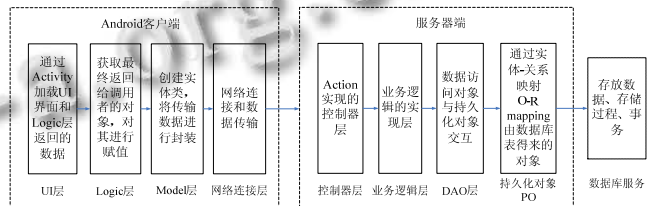


图 2 系统架构

为了增强软件系统的松耦合性, 在各层之间运用框架技术进行分离。其中, 运用 Struts2 将表示层、业务层分离, 创建与业务相关的 Bean 及其之间的依赖注入(DI)关系, 控制各种操作及操作后的转向。运用 Spring 实现业务层与数据访问层的分离, 管理业务层组件。运用 iBatis 实现数据访问层、数据层之间分离^[6]。

Android 端开发时, 分为 UI 层、Logic 层、Model 层、网络连接层, UI 层放入了各个 Activity 和自定义的控件, Logic 层处理业务逻辑, 获取最终返回给调用者的对象, 对对象属性进行赋值。Model 层作为实体类层, 存放了若干实体类, 分别表示用户信息、新闻动态、基地自检信息、监管队伍等。网络连接层实现网络连接和数据传输, Android 应用作为客户端时, 通过 WIFI 或者 4G 与服务器数据交互, Android 应用将会通过 Apache HttpClient 和 HttpURLConnection 向服务器的控制器发送请求, 并获取服务器响应, 客户端和服务器的交互数据格式为 JSON 格式^[7]。

2 软件设计

2.1 界面设计

2.1.1 主功能界面设计

Android 应用的每一个窗体界面都是一个 Activity, 软件运行时, 每一个拥有界面的 Activity 都需要加载一个 XML 布局文件。Android 系统提供了 5 种常见的

布局方式. 图 3 显示的是此界面的布局文件, 以 outline 轮廓图方式显示, 本界面最外层(第 1 层)采用 LinearLayout 线性布局来实现, 其 orientation 属性设置为 vertical 垂直布局, 采用布局的嵌套来排列控件, 从最外层到最内层布局共计 5 层. 在第 1 个 2 层 LinearLayout 中, 分别用 ImageView 控件和 TextView 控件来显示 Logo 和软件的名称, 在第 2 个 2 层 LinearLayout 中, 嵌套了 3 个 3 层 LinearLayout, 分别用来显示欢迎框、登录账号对应的监管单位名称和 8 个功能快捷图标. 在第 5 层即最内层的 LinearLayout 中, 分别用 1 个 ImageView 和 TextView 显示各功能对应的图标和名称. 在最后 1 个 2 层 LinearLayout 中, 用来显示“扫一扫”、“注销登录”、“退出系统”功能的快捷图标. 图 4 所示的是主功能界面加载数据后的效果图.

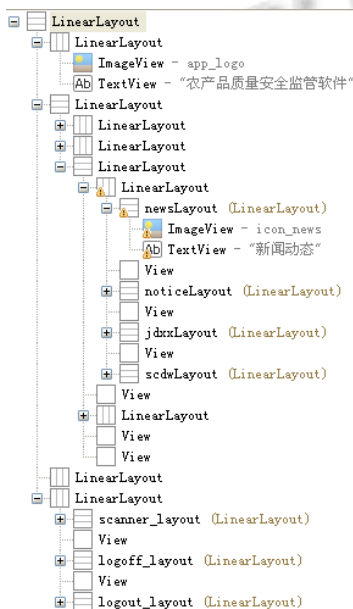


图 3 主功能界面布局

2.1.2 子功能列表界面设计

在图 4 中点击“基地自检”按钮, 在此按钮上注册事件监听器, 当发生单击事件时, 相关的 onclick 处理方法是跳转到 action 名称是“cn.edu.jsahvc.action.JDZJ”所对应的 Activity, 在功能清单文件 AndroidManifest.xml 中, 设置对应的 Activity 是 ui.dashboard.JdzjActivity.

此功能的 UI 布局中, 显示列表区域采用 ListView 控件, ListView 控件能较好的展示列表内容, 并且可以方便的加载数据, 给它赋 id 为 lv_base, 用于在 src 源代码中获取此控件来编程.



图 4 主功能界面

2.2 数据读取

2.2.1 登录数据读取

根据 URL 和文本框得到的账号、密码组合成新的 URL, 采用 GET 方式访问服务器, 如果验证成功后, 服务器返回包括 nodeId 和 dividedFlag 等字段的 JSON 数据流. 内容包括: {"dividedFlag": "321202001", "result": "LoggedIn", "nodeId": "321202001001", "name": "chengtao", "dept": "城东街道监管站", "userType": "7"}. nodeId 表示相应监管站的节点 id, dividedFlag 表示节点所属地区的 id, nodeId 去除后面 3 位顺序号即为 dividedFlag. Android 利用 SharedPreferences 存储类对这些返回的字段值进行参数保存^[8].

如图 5 所示, 乡镇级别监管站 nodeId 的编码结构共有 12 位: 由省级区域编号(2 位)+市级区域编号(2 位)+县级区域编号(2 位)+乡镇级区域编号(3 位)+顺序号(3 位)组成. 其中省级区域包括省、直辖市或自治区, 市级区域包括地级市、自治州, 乡镇级区域包括乡、镇或街道办. 同一个乡镇级区域的不同名称但是具有相同权限的监管机构, 系统会依次分配顺序号. 例如某某镇的监管站和农检中心, 可以分别用 001 和 002 顺序号. 使用城东街道监管站的监管用户账号 chengtao 登录后, 返回 nodeId=321202001001 和 dividedFlag=321202001, 表示的详细信息是江苏省泰州市海陵区城东街道农产品质量安全监管站.

2.2.2 普通数据读取

以“基地自检”功能为例, 将在系统登录成功后服务器返回的 nodeId, dividedFlag 作为 mIBaseTraceLogic.sendForJDZJInfo 方法的参数进行发送, 同时发送的参

数还有 pageNum 作为显示的页次, 采用 URL.openConnection 方法跟服务器连接, 并且通过 URLConnection 类的 setDoInput、setUseCaches、setConnectTimeout、setReadTimeout 等几个方法来设置请求头字段. 通过 URLConnection 类的 getOutputStream 方法构造 OutputStream 类型的输出流对象, 通过此对象来发送请求消息体^[9]. 服务器收到客户端的请求消息后, 通过 HttpServletRequest 类的 getParameter 方法来获得 nodeId、dividedFlag、pageNum 的值, 将这几个值作为到数据库查找数据的条件, 查找成功后将结果集封装成 JSON 格式, 作为一种轻量级的数据格式便于返回. 客户端接受到返回后的 JSON 数据后, 将 JSON 数据进行解析和转换为 JDZJInfo 类型(JDZJInfo 是基地自检结果的实体类), 再转换成基地自检列表所需的数据适配器对象, 在 2.1(2)中的 lv_base 控件转换成可编程的 ListView 对象, 使用 ListView 的 setAdapter 方法将此适配器进行加载, 从而显示出相关的效果.

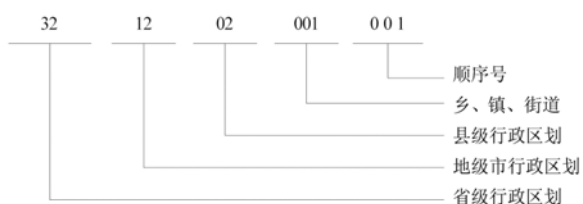


图5 乡镇级监管站 nodeId 的编码结构

2.3 数据写入

涉及到数据写入的是巡查登记模块, 将点击“查询”按钮后获取的被检单位信息列表中, 选择某条作为当时巡查对象的检查地点, 并手动输入检查项目、检查结果, 对当时巡视工作场景拍照, 并自动获取当时所在地的经纬度和地理位置, 一起上传到服务器.

2.3.1 图片转换

对拍摄的检查场景照片, 采用 Base64 编码+Http 的传输方式, Base64 编码适用于在 Http 环境下传递较长的数据, 所有的图片都可以通过算法获取到它的字节数组, 上传时, 服务器接收到后将 Base64 编码存储到数据库服务器. 如果后续客户端要查询之前的巡查详情的时候, 需要将 Base64 字符串返回至客户端, 再由客户端处理获得图片, 从而可以看到以往上传过的所有图片.

代码流程如下:

- ① 读取本地图片, 获取 Base64 字节数据;
- ② 将 Base64 字节数据, 写到另外一份文件当中去;
- ③ 在上传图片之前会先获取服务器的接口地址, 如果地址获取不到, 就不执行上传任务. 如果地址获取完成后, 再获取文件中的 Base64 编码, 而后执行上传任务;
- ④ 服务器获取到 Base64 字符串后, 保存到数据库中. 由于 Base64 编码, 是很长的字符串. 服务器数据库中, 保存 Base64 编码的字段不限制长度.
- ⑤ 客户端查询图片上传历史时, 通过接口访问, 获取到服务器存放的 Base64 编码, 在客户端处理后得到图片.

2.3.2 地图获取

本软件使用的是高德地图进行地址定位, 采用的纯网络定位方式. 首先到高德官网上下载地址定位包, 本软件使用的是 Android_Location_V1.1.2.jar 包, 将其附加到软件项目中 libs 文件夹中, 然后到功能清单文件 AndroidManifest.xml 中添加用户 key, 同时添加 Android 系统允许程序使用到的权限, 具体需要设置 android.permission.INTERNET、android.permission.WRITE_EXTERNAL_STORAGE、android.permission.ACCESS_COARSE_LOCATION、android.permission.ACCESS_NETWORK_STATE、android.permission.ACCESS_FINE_LOCATION、android.permission.READ_PHONE_STATE、android.permission.CHANGE_WIFI_STATE、android.permission.ACCESS_WIFI_STATE、android.permission.CHANGE_CONFIGURATION 共 9 个权限, 分别表示允许程序访问网络、可以读写 SDCARD、允许通过 CellID 或 WiFi 热点来获取地理位置、允许访问有关 GSM 网络信息、允许访问精确位置(如 GPS)、允许访问电话状态、允许改变 WIFI 连接状态、允许访问 WIFI 网络状态信息、允许修改当前设置. 权限配置好后, 在 Activity 中实现 jar 包中的 AmapLocationListener 接口, 并且重写其中的 onLocationChanged(AMapLocation location)方法, 通过读取 location.getLatitude()、location.getLongitude()、location.getExtras().getString("desc")获得当时所在地点的经度、纬度、地理位置.

2.3.3 数据上传

数据上传采用 POST 方式提交, 在客户端将上传数据转化为 JSON 格式后, 服务器的 Action 进行接收、解析和存放。因为每次巡查登记上传图片往往不止一张, 所以系统支持多张图片上传, 在设计实体类的时候, 需要设计两个实体类, 一个是巡查登记类 QtsDayNodesVo, 另外一个为上传图片附件类 QtsDayNodesImageVo, QtsDayNodesVo 中加入 QtsDayNodesImageVo 类的集合作为 QtsDayNodesVo 的属性^[10]。如表 1 和表 2 所示, 在数据库设计两个数据表与实体类对应, 上传图片附件数据表中的 dayNodeId 作为外键, 指向巡查登记数据表的主键 rowId。

表 1 巡查登记数据表(t_ QtsDayNodesVo)

列名	数据类型	数据长度	允许空	说明	是否主键
rowId	varchar	50	否	ID	是
checkAdd	varchar	100	否	地址	否
checkDate	Date		否	检查时间	否
personName	varchar	50	否	检测人员	否
checkProject	varchar	50	否	检测项目	否
checkResult	varchar	50	否	检测结果	否
checkComp	varchar	100	否	被检单位	否

表 2 上传图片附件数据表(t_ QtsDayNodesImageVo)

列名	数据类型	数据长度	允许空	说明	是否主键
rowId	varchar	50	否	ID	是
imagePath	varchar	50	否	图片路径 外键, 指向 t_	否
dayNodeId	varchar	50	否	QtsDayNodesVo 的 主键 rowId	否
Photocode	varchar	max	否	图片的 Base64 编码	否

3 软件运行和应用情况

图 6 所示的是基地自检和巡查登记两个典型功能模块的运行效果。目前该软件在泰州地区农业监管系统进行应用, 运行稳定有效, 监管人员可以随时随地使用该软件浏览新闻动态、通知公告、基地自检等消息, 并且可以上传实地巡查的照片和检查结果, 实现了巡查工作的快速电子化登记。

4 结语

本文通过分层模式设计了一个基于 Android 的农产品质量安全监管软件, 创建了基于 C/S 模式的系统架构, 并对客户端和服务器的开发分别进行分层优化, 设计了手机的界面布局, 研究了软件编程技术。实际

应用表明, 监管人员携带安装有该软件的手机使用方便, 访问速度快。今后还将继续对此软件进行下列几个方面的设计和开发: 1、增加农业执法大队的执法人员角色, 如果监管站(科)的监管人员监管督查时发现农产品质量问题但是没有权力处理时, 通过软件进行移交给执法人员, 服务器同步相关数据; 2、软件增加行政处罚模块, 与手写板、便携式蓝牙打印机等设备集成, 内置处罚案卷模板, 实现处罚案卷的现场标准化制作, 进一步提高执法效率。



图 6 基地自检和巡查登记模块运行界面

参考文献

- 1 国务院. 国务院关于积极推进“互联网+”行动的指导意见. 中国政府网. [2015-07-04].
- 2 陈晓华. 2014年我国农产品质量安全监管成效及2015年重点任务. 农产品质量与安全, 2015, (1): 3-8.
- 3 惠晓威, 王克. 移动视频监控系统的实现. 计算机应用与软件, 2014, (1): 148-150, 179.
- 4 张诚洁. 基于 Android 平台的移动在线测试系统设计与实

- 现. 计算机应用与软件, 2015, (4): 323-325.
- 5 郭健侯, 蔡雪松. 基于 Android 平台的公交信息发布终端软件的设计与实现. 计算机应用与软件, 2014, (9): 322-324, 333.
- 6 陈大鹏, 毛罕平, 左志宇. 基于 Android 手机的温室环境远程监控系统设计. 江苏农业科学, 2013, (9): 375-379.
- 7 李慧, 刘星桥, 李景, 陆晓嵩, 宦娟. 基于物联网 Android 平台的水产养殖远程监控系统. 农业工程学报, 2013, (13): 175-181.
- 8 陈望挺, 林满足, 陈健, 章跃, 傅奇佳, 竺乐庆. 基于 JSP 和 Android 的 C/S 结构问卷系统. 计算机应用, 2013, (3): 886-889.
- 9 吴昊, 何鹏, 杨曼. 基于 Android 的温室大棚监控管理信息系统研究. 农机化研究 2013, (11): 79-83.
- 10 孙小华, 王福顺, 杨会英, 赵艳, 王凤国. 基于智能手机的农业信息服务系统研究. 科技和产业, 2013, (6): 33-36.

WWW.C-S-A.ORG.CN

WWW.C-S-A.ORG.CN