**基于Android终端的鞋业产品质量系统**

**软件学院 软件工程专业**

**123012009137 张龙德 指导教师：吴垠**

**【摘要】质量是产品永恒不变的主题，保证产品质量就是坚持企业的永续发展。近年来，许多企业为了使产品质量的管理更加智能化、更加有效率，纷纷引进计算机辅助系统来管理产品质量。然而，生产车间的条件限制必然导致难以将计算机作为采集数据的终端，这是像制鞋企业这类传统制造业企业面临的又一困难。而本文研究的产品质量管理系统，是一套基于Android终端的产品质量系统，以制鞋行业作为实现载体。它从品管员的角度考虑，采用有利于品管员携带的手机客户端，以便于采集质量数据。从数据分析人员以及企业经营者的角度考虑，采用便于质量数据分析和展示的网页客户端。从系统维护人员、开发人员以及经营成本的角度考虑，采用了软硬件裁剪的专门嵌入式设备作为管理数据的服务器，适合嵌入式开发的体积小且运算速快的SQLite3数据库。首先，本文先对项目需求进行分析，分别从市场、技术、经济三个角度综合考虑本系统实现的可行性。然后，从设计模式的角度加以分析本系统的架构，重点描述了本系统的功能模块，并利用用例图和流程图直观地反应本系统的主要功能。之后，简单描述了本系统实现的关键技术，主要包括多线程技术、IO多路复用技术、SQLite数据库技术、Socket通信技术以及ASP.Net的若干种技术。接着，本文花了大量笔墨介绍产品质量系统的实现，分别从手机客户端、数据库服务器端、网页客户端三个方面，阐述函数设计、编码实现以及工程三层架构的具体实现。最后，本文简单介绍了系统的使用说明并进行项目总结。**

**【关键词】产品质量管理系统;品管系统;Android终端;鞋业产品质量管理系统**

**目 录**

[1 绪论 4](#_Toc356147714)

[1.1项目背景 4](#_Toc356147715)

[1.2项目意义 4](#_Toc356147716)

[2 需求分析 4](#_Toc356147717)

[2.1功能性需求 4](#_Toc356147718)

[2.2非功能性需求 5](#_Toc356147719)

[3 可行性分析 5](#_Toc356147720)

[3.1市场可行性 5](#_Toc356147721)

[3.2技术可行性 5](#_Toc356147722)

[3.3经济可行性 6](#_Toc356147723)

[4 系统概要设计 6](#_Toc356147724)

[4.1系统用例图 6](#_Toc356147725)

[4.2系统功能模块描述 6](#_Toc356147726)

[4.2.1手机客户端各功能模块描述 7](#_Toc356147727)

[4.2.2数据库服务器端各功能模块描述 7](#_Toc356147728)

[4.2.3网页客户端各功能模块描述 7](#_Toc356147729)

[4.3系统流程图 7](#_Toc356147730)

[4.3.1质量数据采集 7](#_Toc356147731)

[4.3.2数据库服务器 8](#_Toc356147732)

[4.3.3数据展示与分析 8](#_Toc356147733)

[4.4 系统运行环境 9](#_Toc356147734)

[5 系统详细设计 9](#_Toc356147735)

[5.1系统架构 9](#_Toc356147736)

[5.2系统设计模式 10](#_Toc356147737)

[5.2.1传统两层客户机/服务器模式(C/S) 10](#_Toc356147738)

[5.2.2浏览器/服务器模式(B/S) 10](#_Toc356147739)

[5.3数据库设计 11](#_Toc356147740)

[5.3.1表格设计 11](#_Toc356147741)

[5.3.2数据库配置优化 13](#_Toc356147742)

[5.4网络连接设计 14](#_Toc356147743)

[6 系统实现 15](#_Toc356147744)

[6.1 系统实现关键技术 15](#_Toc356147745)

[6.1.1多线程服务器 15](#_Toc356147746)

[6.1.2面向连接的Socket编程模型 16](#_Toc356147747)

[6.1.3 IO多路复用 17](#_Toc356147748)

[6.1.4 SQLite3数据库 17](#_Toc356147749)

[6.1.5 ASP.Net 17](#_Toc356147750)

[6.2数据库服务器的实现 18](#_Toc356147751)

[6.2.1概述 18](#_Toc356147752)

[6.2.2主程序模块 18](#_Toc356147753)

[6.2.3业务处理模块 19](#_Toc356147754)

[6.2.4 Socket通信模块 19](#_Toc356147755)

[6.2.5线程池模块 19](#_Toc356147756)

[6.2.6字符串处理模块 19](#_Toc356147757)

[6.2.7数据库访问模块 19](#_Toc356147758)

[6.2.8调试模块 20](#_Toc356147759)

[6.2.9 编译、移植和运行 20](#_Toc356147760)

[6.3手机客户端的实现 21](#_Toc356147761)

[6.3.1概述 21](#_Toc356147762)

[6.3.2界面实现 21](#_Toc356147763)

[6.3.3面向连接的Socket通信模块 21](#_Toc356147764)

[6.3.4本地缓存模块 22](#_Toc356147765)

[6.3.5业务逻辑模块 23](#_Toc356147766)

[6.3.6工程代码架构 23](#_Toc356147767)

[6.3.7编译与运行 23](#_Toc356147768)

[6.4网页客户端的实现 24](#_Toc356147769)

[6.4.1概述 24](#_Toc356147770)

[6.4.2网页前端 24](#_Toc356147771)

[6.4.3网页后端 24](#_Toc356147772)

[6.4.4工程代码架构 25](#_Toc356147773)

[6.4.5编译与运行 26](#_Toc356147774)

[7 系统使用说明 26](#_Toc356147775)

[7.1运行数据库服务器 26](#_Toc356147776)

[7.2采集质量数据 27](#_Toc356147777)

[7.3显示分析数据 29](#_Toc356147778)

[8总结 37](#_Toc356147779)

[致谢 38](#_Toc356147780)

[参考文献 38](#_Toc356147781)

# 1 绪论

## 1.1项目背景

质量是产品永恒的主题。在现代企业中，能够在市场竞争中得以生存和发展的重要因素是产品质量。可靠的产品质量是企业持续发展、赢得竞争力的战略武器。因此，产品质量信息得到了大中小企业的重视[1]。

然而，在传统产品质量管理流程中，品管员主要以手工方式，把产品质量信息逐一记录在纸质材料上，然后面对大量的纸质信息进行数据整理、统计分析等后期工作，操作起来不仅繁琐而且效率低下，整个周期也相对较长[2]。这样下来，一方面将耗费大量人力物力在质量数据的录入和对纸质材料的整理上；另一方面，产品质量信息的获得及时性较差，从而无法快速地对大量质量信息进行统计分析，进而导致公司高层做出决策的偏差，给公司带来不必要的损失。

## 1.2项目意义

随着计算机技术的日益成熟，企业向智能型、管理型的现代化企业发展[3]，由此计算机辅助质量系统也被引入传统品质管理过程。国内外的计算机质量辅助系统林林种种，但总的发展趋势是更加智能化、网络化，更加便捷、及时。然而，许多生产车间借助计算机终端采集质量数据比较不方便，尤其像制鞋企业这种制造业。因此，借助手机终端采集数据就应用而生了，即将传统产品质量管理系统中的质量数据录入这一终端改造成用Android手机终端以方便生产车间质量信息的录入。采用Android手机作为采集质量数据的终端为什么会比较适合这些制鞋生产车间员工的使用了？试想，只要手中的智能手机是Android系统的，安装一个产品质量管理信息系统的手机客户端，就可以轻松方便地进行质量信息录入，从而提高效率，及时性得以保障。

# 2 需求分析

本系统的需求分为功能性需求和非功能性需求[4]。其中，功能性需求是为满足产品质量管理系统用户的业务需求而必须实现的、具有实质性内容的要求；而非功能性需求是旨在更好发挥功能性需求，系统必须具备的某些特性。

## 2.1功能性需求

（1）质量信息的录入

品质管理员登录Android手机中的产品质量管理系统客户端，根据相应的生产车间，选择员工，建立班次，对于产品做出质量评价后，直接通过手机的触摸屏准确无误地录入产品质量信息，上传到数据库服务器，存储到数据库服务器上的数据库中，到下班点的时候，可以进行下班操作。当生产中遇到紧急情况或者产品质量出现严重问题的时候，可以下达停产整顿、查明原因等指令。

（2）质量信息的查询统计分析

经理等管理人员通过登录客户端网页，间接访问数据库服务器，了解近期产品生产情况，产品生产过程中存在的问题、经过BI（商务智能）处理的解决方案，查询质量数据的统计分析结果。简单的统计分析过程是通过计算机按照一定的业务流程直接以饼状图、条形图、折线图等形式直观地呈现在网页界面中，更加复杂的统计分析结论是通过质量数据分析人员登录到客户端，进行数据挖掘之后得到的。

（3）质量信息的数据挖掘

质量数据分析人员通过客户端登录到本系统后，根据公司需求、产品生产情况以及产品质量问题，有针对性地而非机械性地提出有利于公司发展的、可以供公司管理层作为参考的依据和策略。

（4）系统配置维护

系统管理员登录到本系统后，不仅可以进行数据库维护，如数据库备份等操作，还必须能够管理员工信息，如修改密码，以及系统设置等常见系统操作。

## 2.2非功能性需求

（1）产品质量管理系统的及时性

品质管理员通过手机端采集到质量数据录入手机的时间不超过2秒。在局域网内，将质量数据上传到服务器的时间不能超过3秒。

（2）产品质量管理系统的响应时间

网页页面加载时间低于2秒，网页页面响应时间低于5秒。

手机页面加载时间低于1秒，手机页面响应时间低于3秒。

（3）产品质量管理系统的负载能力

产品质量管理系统必须满足整个品质管理流程的需要，就应该能够具备一定并发数量的服务器，能够承载各个车间品管员、经理、数据分析员同时访问而正常运行，并发量为500基本能满足要求。

（4）产品质量管理系统的容错能力

服务器对于非法访问、非法数据请求、错误操作、以及超负荷的操作应能够及时处理，继续良好运行。

（5）产品质量管理系统的数据安全性

由于产品质量数据用于了解产品生产情况，提供决策参考，属于公司内部资料信息，必须保证数据库服务器和数据库以及局域网内通信不被非法窃取，仅对相关人员开放其权限范围内的数据信息，以保证数据安全性。

（6）产品质量管理系统的可扩展性

可以根据公司不同时期的需求，扩展系统功能，比如修改统计分析指标，增加统计项目等。

# 3 可行性分析

## 3.1市场可行性

一方面，从工作人员的角度考虑，由于生产车间与进行数据统计分析人员办公地点的间隔性，目前大部分企业采用纸质记录质量信息，然后电脑录入，鲜有手持嵌入式设备录入和智能手机录入的方式。加之，智能手机触摸屏的普及更是如虎添翼，避免了诸多纸质录入的烦恼。

另一方面，从企业老板的立场考虑，由于只需要局域网内的通信（当然可以注册域名，购买服务器空间，把网页端挂到广域网上），我们可以安装嵌入式设备作为数据库服务器、轻量级的数据维护数据、一台电脑作为局域网内的网页服务器，节约成本，安全性也更高。如果推出该质量管理系统，必将会受到关注。

## 3.2技术可行性

采用BS（browser/server）和C/S(client/server)的模式，技术成熟，实施起来比较简单，而且有利于系统的安装维护和更新。将所有数据都保存到数据库服务器端，网页的访问放到网页服务器端，各个平台通过Socket通信，降低了它们之间的耦合度，只要通信协议统一一致，就可以将整个系统划分为用C语言开发的服务器端、用Android开发的手机端和用ASP.Net开发的网页端，三个方向可以同时进行。另外，轻量级的数据库SQLite3.0维护起来更加方便。

## 3.3经济可行性

首先智能手机端安装一个质量管理系统APP端几乎是零成本。数据库服务器可以采用嵌入式软硬件，进行裁剪、量身定做，成本也不会高。网页服务器只要一台电脑即可完成，算下来，经济实惠。

# 4 系统概要设计

## 4.1系统用例图

本小结的用例图描述了鞋业产品质量管理系统的四个参与者：品管员、数据分析员、经理和系统管理者与本系统众多用例之间的关系，如图4-1-1所示。

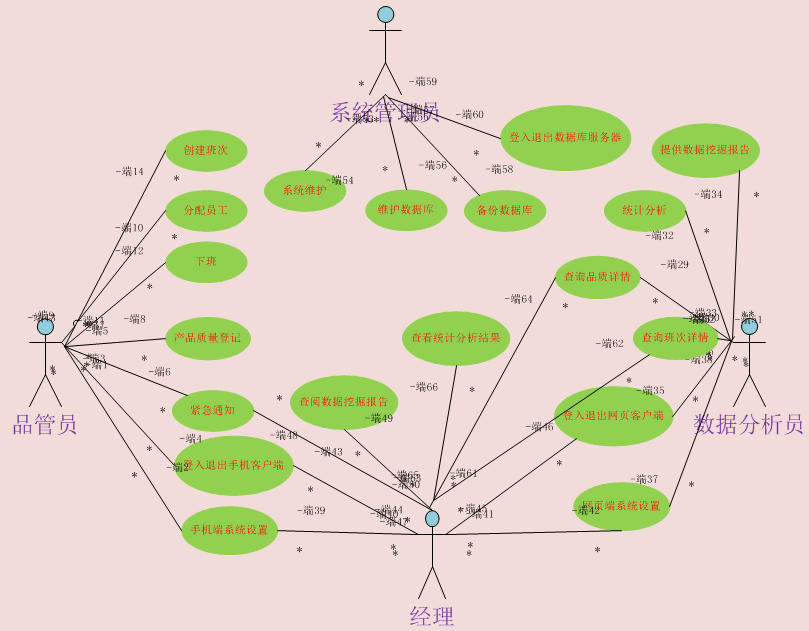


图4-1-1 鞋业产品质量管理系统用例图

## 4.2系统功能模块描述

系统功能模块分别从手机客户端、数据服务器端、网页客户端三大模块加以描述各自子模块的功能点。

### 4.2.1手机客户端各功能模块描述

表4-2-1 手机客户端各功能模块描述

|  |  |
| --- | --- |
| **模 块 名** | **描述** |
| 登录  建立班次 | 品管员选择生产车间登录手机端  选择员工建立一个班次 |
| 产品质量登记 | 对于某个员工的生产的产品进行产品详情记录 |
| 品管等级 | 记录员工加工某产品的完成度。 |
| 本机存储 | 本机数据保存，用来保存用户配置 |
| 系统设置 | 修改服务器IP及端口号 |

### 4.2.2数据库服务器端各功能模块描述

表4-2-2数据库服务器功能模块描述

|  |  |
| --- | --- |
| **模 块 名** | **描述** |
| 网络连接模块（Android） | 网络Socket通讯模块，使用TCP/IP，遵循自定义通信协议 |
| 网络连接模块（C#网页） | 网络Socket通讯模块，使用TCP/IP，遵循自定义通信协议 |
| 字符串处理模块 | 字符串处理模块，用来处理网络传输的字符串 |
| SQLite3数据库模块 | 访问数据库，增删改查数据库 |
| Socket通信模块 | 绑定、监听以及进行Socket通信的基本读写 |
| 线程池模块 | 开辟一定数量的线程，维护线程 |

### 4.2.3网页客户端各功能模块描述

表4-2-3 网页客户端端各功能模块描述

|  |  |
| --- | --- |
| **模 块 名** | **描述** |
| 实时显示班次 | 以闪动的形式，显示最新班次情况 |
| 时事现实品质  班次查询  品质详情查询 | 以闪动的形式，显示最新详细记录情况  对历史班次进行查询  对历史品质详情进行查询 |
| 数据分析模块  数据挖掘 | 以表格以及饼状图、折线图等统计分析图形式直观显示统计分析结果  对统计分析的数据进一步分析 |
| 系统设置 | 设置刷新时间，进行密码修改 |

## 4.3系统流程图

### 4.3.1质量数据采集

本小节的流程图清新地描述了手机客户端从登录、选择员工到建立班次、进行品质登记整个采集质量数据的过程，如图4-3-1所示。



图4-3-1 质量数据采集的流程图

### 4.3.2数据库服务器

本小节的流程图清新地描述了数据库服务器从接受客户端数据请求、建立连接到完成数据交互的整个具体操作流程，如图4-3-2所示。



图4-3-2 数据库服务器运行的流程图

### 4.3.3数据展示与分析

本小节的流程图清新地描述了网页客户端从登录系统到查看品质量情况、统计分析、提供数据挖掘解决方案的整个操作流程示意，如图4-3-3所示。



图4-3-3 数据展示与分析的流程图

## 4.4 系统运行环境

（1）用于质量信息录入的手机端：Android 2.2以上系统

（2）运行服务器的硬件：Tiny6410Arm板

（3）进行数据查询、统计分析的网页客户端：IE 6.0以上浏览器（或者Firefox、Chrome浏览器）。

（4）运行网页服务器：装有IIS7.0或者Apache 2.4.2的计算机（局域网）

# 5 系统详细设计

## 5.1系统架构

本文研究的鞋业产品质量管理系统的整体架构分为采集质量数据的手机客户端架构、提供数据请求的数据库服务器和显示分析数据的网页客户端架构，如图5-1-1所示。

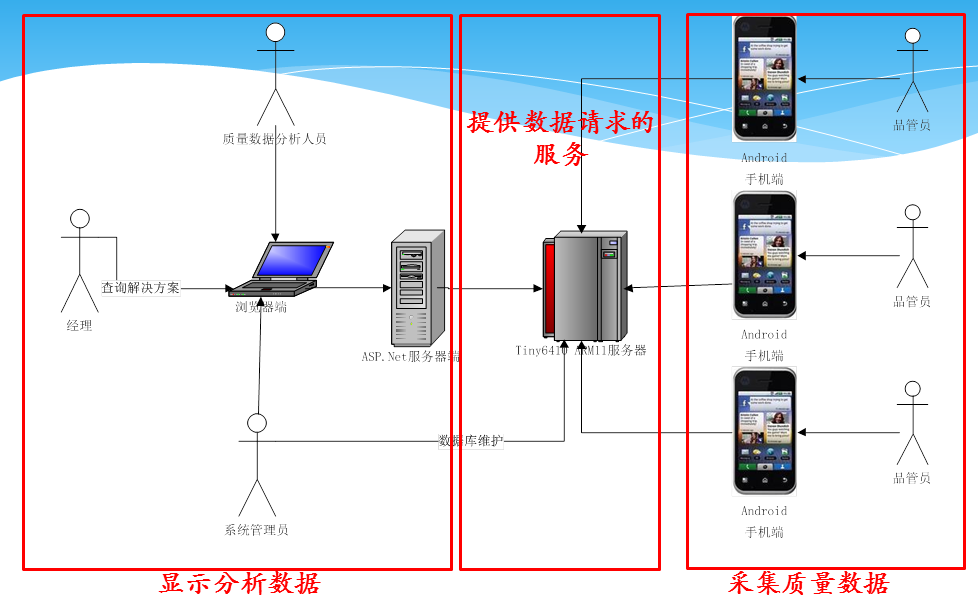


图5-1-1 产品质量管理系统架构图

## 5.2系统设计模式

### 5.2.1传统两层客户机/服务器模式(C/S)

客户机/服务器系统有三个主要部件:数据库服务器、客户应用程序和网络[5]，具体结构如图5-2-1。



图5-2-1 经典C/S模式图

由于采集质量数据的地点主要限于生产车间内，只要在局域网内通信即可，故采用C/S模式来设计采集质量数据的手机客户端与数据保存维护的数据库服务器，具体结构图5-2-2。



图5-2-2 产品质量管理系统手机-数据库服务器C/S模式图

### 5.2.2浏览器/服务器模式(B/S)

我们知道经典三层客户机/服务器模式可以分为：功能层、数据层、表示层，而浏览器/服务器架构模式实际上是它的一种实现方式[6]，具体结构为：浏览器/Web服务器/数据库服务器，采用B/S结构的应用系统的基本框架如图5-2-3所示。



图5-2-3 经典B/S模式图

考虑到数据显示可以随时随地的特性，且需要展示各种统计分析特效，采用B/S模式来设计数据显示和操作的网页客户端和网页服务器比较合适，具体如图5-2-4所示：



图5-2-4 产品质量管理系统网页-数据库服务器B/S模式图

## 5.3数据库设计

### 5.3.1表格设计

首先创建一个数据库SQLite3 qm.db,该数据库共有8张表，它们分别是Employee、quality、workType、class、detail、plant、position，这8张表的结构如表5-3-1至5-3-8。

表5-3-1员工信息描述

|  |  |  |  |
| --- | --- | --- | --- |
| **列名** | **数据类型** | **可否为空** | **描述** |
| employeeNo（PK） | integer | Not NULL | 员工号 |
| name | text | Not NULL | 员工姓名 |
| pwd | text | Not NULL | 密码 |
| position | text | Not NULL | 职位 |
| plantNo | integer | Not NULL | 车间号 |
| workTypeNo | integer | Not NULL | 工种号 |

表5-3-2 产品质量信息描述（quality）

|  |  |  |  |
| --- | --- | --- | --- |
| **列名** | **数据类型** | **可否 为空** | **描述** |
| qualityNo（PK） | integer | Not NULL | 品质号 |
| drawback | text | Not NULL | 具体缺陷名称 |
| workTypeNo | integer | Not NULL | 工种号 |

表5-3-3工种信息描述（workType）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **列名** | **数据类型** | | | **可否为空** | | **描述** |
| workTypeNo（PK） | | integer | Not NULL | | 工种号 | |
| productName | | text | Not NULL | | 产品名称 | |

表5-3-4班次信息描述（class）

|  |  |  |  |
| --- | --- | --- | --- |
| **列名** | **数据 类型** | **可否为空** | **描述** |
| classNo | integer | Not NULL | 班次表号 |
| qmEmployeeNo | integer | Not NULL | 品管员号 |
| beginTime | text | Not NULL | 开始时间 |
| endTime | text | Not NULL | 结束时间 |
| classRange | text | Not NULL | 班次 早/中/晚 |
| plantNo | integer | Not NULL | 车间号 |
| crStatus | text | Not NULL | 班次状态：是否下班 |

表5-3-5产品质量记录详情描述（detail）

|  |  |  |  |
| --- | --- | --- | --- |
| **列名** | **数据**  **类型** | **可否为空** | **描述** |
| classNo（PK） | integer | Not NULL | 班次表号 |
| employeeNo | integer | Not NULL | 被登记的员工号 |
| workTypeNo | integer | Not NULL | 工种号 |
| concreteData | text | Not NULL | 具体数据 |

表5-3-6车间信息描述（plant）

|  |  |  |  |
| --- | --- | --- | --- |
| **列名** | **数据类型** | **可否为空** | **说明** |
| plantNo（PK） | integer | Not NULL | 车间号 |
| plantName | text | Not NULL | 车间名称 |
| plantPosition | text | Not NULL | 车间位置 |
| managerNo | text | Not NULL | 负责人员工号 |

表5-3-7职位信息描述（position）

|  |  |  |  |
| --- | --- | --- | --- |
| **列名** | **数据类型** | **可否为空** | **说明** |
| positionNo（PK） | integer | Not NULL | 职位号 |
| positionName | text | Not NULL | 职位名称 |

表5-3-8商务智能信息描述（res）

|  |  |  |  |
| --- | --- | --- | --- |
| **列名** | **数据类型** | **可否为空** | **说明** |
| resNo（PK） | integer | Not NULL | 解决方案号 |
| resMakerNo | integer | Not NULL | 分析员员工号 |
| resText | text | Not NULL | 解决方案 |
| positionNo | integer | Not NULL | 可以查看该 解决方案的职位 |

### 5.3.2数据库配置优化

由于SQLite3数据库是一款开源的软件，可以根据开发人员的需要，修改源代码，编译生成定制的数据库，比如说SQLite3中没有密码设置，这就需要开发人员进行手工定制。另一方面，要想使用性能最优的SQLite3还要根据具体项目进行性能优化方面的配置。本系统需要的数据库，我也进行了配置。

（1）增大数据库内存页大小，如图5-3-1（该选项必须在创建数据之前设置）



图5-3-1 增大数据库内存页大小的操作截图

（2）打开数据库的自动清理功能，如图5-3-2（该选项必须在创建表之前设置）



图5-3-2打开数据库的自动清理功能的操作截图

（3）增大cache的大小，如图5-3-3



图5-3-3增大数据库cache的操作截图

（4）打开文字敏感选项，如图5-3-4，否则搜索中文和大小写的时候会出现错误。

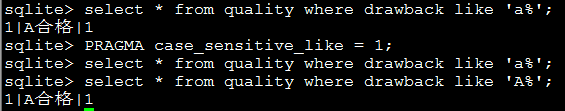


图5-3-4打开文字敏感选项的操作截图

（5 ）设置正确的编码方式如图5-3-5

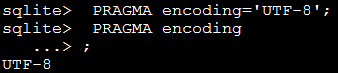


图5-3-5设置编码方式为UTF-8的操作截图

（6）备份还原数据库

方法一：通过导出导入命令语句来备份恢复数据库

导出数据库SQL语句（有关创建表、插入数据的一个事务）：

[root@EmbedSky db]# SQLite3 qm.db ".output qm.sql"

通入之前备份的命令以还原数据库：

[root@EmbedSky db]# SQLite3 qm.db ".dump" ） qm.sql

此方法备份的数据库是经过清理的，大小明显小于原数据库。（大小与原数据库经过VACUUM之后差不多）

方法二：使用SQLite3内部命令来备份恢复数据库

使用SQLite3内部命令“.backup”备份数据库

[root@EmbedSky db]# SQLite3 qm.db ".backup main qmBackup2.db"

此时备份的数据库大小与原数据库一样大（偏大，并没有清理VACUUM）

使用SQLite3内部命令“.restore”还原数据库

[root@EmbedSky db]# SQLite3 qmBackup3.db ".restore main qm.db"

## 5.4网络连接设计

本系统主要分为手机客户端、数据库服务器和网页客户端三大模块，这三大模块是通过网络连接起来的，那它们是如何通过网络连接起来的了？其中，手机客户端与数据库服务器是通过Wifi局域网连接起来的，而网页客户端与数据库服务器是以Web服务作为桥梁连接起来的。数据库服务器是为手机客户端和网页客户端的数据连接请求提供服务器的，数据库服务器中保存着客户端的数据。它们之间的连接都是基于TCP/IP的Socket通信，由于本系统网络连接的复杂性，故自定义一套通信协议实现手机客户端、数据库服务器和网页客户端的网络连接。自定义的通信协议为三大模块开辟了交互的接口，定义好通信协议同时也为后续开发避免了模块交互的麻烦。

通信协议的定义可以采用很多种方式，这里介绍三种。第一种方式，整个通信包的大小限定，将其划分包头、包尾和核心部分，限定各部分的数据长度。包头格式固定，主要为数报的属性内容，包尾则大多为CRC检验码。如果需要传输的数据长度不够包长，则将多余部分填充至满；如果需要传输的数据超过限长，则拆分成多个通信包传输。当然，更高级一点的方式可以设定一个调整字段，有效增加核心部分的长度，比方说数字电视中的传输TS包就设定有调整字段adapter\_filed。可想而知，此方式虽然比较完美，但是实现起来过于复杂，不利于实现，还会浪费一定的通信资源。第二种方式，稍微简单一点，将具有一定组织逻辑的数据封装成传输简单结构体，然后通过recvmsg和sendmsg这两个系统函数进行传输，此方式虽然说可以直接以二进制数据进行网络IO读写，但是设定好数组的长度（数据定长），势必要浪费一定的空间。第三种方式就是本系统要采用的通信协议，这里着重介绍。本系统采用的通信已协议，包头、包尾和数据部分以分隔符进行分割，如果分隔符为具体数据，则通过转义字符进行转义。包头固定，但是数据部分的长度灵活调整，不会浪费控件和传输资源。包头为简单的通信标志，有3个字节，第一个字节为通信类型，有两类：‘/’和‘\*’。后面连个字节标识本包的通信意义，有25种，7种保留代实现。数据部分又根据包头的具体意义进行分割，具体如图5-4-1和5-4-2所示。



图5-4-1 通信协议定义详细截图第一部分



图5-4-2 通信协议定义详细截图第二部分

# 6 系统实现

## 6.1 系统实现关键技术

### 6.1.1多线程服务器

通常情况下，多线程的程序是根据需要，调用相应线程函数，执行某一任务后，销毁该线程。纵然说，通过pthreadcreate函数来创建一个新线程是最直接的，但这样创建出来的线程分布在程序的各个位置，管理这些线程比较困难。另一方面，如果我们需要创建大量线程，并且执行一个相对比较简单的任务之后销毁该线程，由于创建线程本身也是需要一定开销的，这样一来，本身简单的任务需要的开销反而变得更大，进而造成多线程的低效率[8]。因此，这种实现方法并不能满足我们的要求，尤其在网络通信编程中，我们的应用需要面对数目巨大的连接请求，同时，这些请求执行的任务却又比较简单，占用的时间很少。就比如说，本系统的数据库服务器，主要负责数据存储维护的，它需要同时面对多个手机客户端和网页客户端频繁的数据存储请求的连接和数据查询请求的连接，但是执行的数据存储任务和数据查询任务，无非是SQL语句的增删改查，消耗的任务时间本身就比较短，这样我们可能就会处于不停的创建线程并销毁线程的状态，进而多线程的并发性也必将降低。

由于上述情况的存在，程序员们便采用了一种线程预创建的技术，即在程序启动的时候，创建一个管理线程，然后开辟一个线程池，在里面存放已经创建好的一定数量的但未激活的空闲线程，由管理线程自行负责管理维护的技术。具体表现为，根据程序需要，可以向线程池申请线程，管理线程便会将事先创建好的空闲线程分配出去，让其执行任务之后，再由管理线程回收，再次成为空闲线程，进入线程队列，等待其它任务，这样便能有效的降低频繁创建销毁线程所带来的额外开销。简单地说，就是线程池中存放了一定数量的线程半成品[9]，具体如图6-1-1所示。

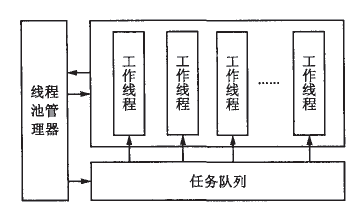


图6-1-1 线程池工作流程示意图

### 6.1.2面向连接的Socket编程模型

Socket接口是网络编程中最为通用的API。Socket接口提供了一个用于通信的端口，程序可以通过这个端口与任何一个具有Socket接口的程序通信,应用程序在网络上传输，接收的信息都通过这个Socket接口来实现[10]。而Linux系统中，提供了两种Socket通信系统调用分别为面向非连接的UDP和面向连接的TCP。面向非连接的UDP通信是一种不可靠的网络传输，由于本系统的质量数据传输是非常重要的，故采用面向连接的可靠的TCP通信，其基本步骤如图6-1-2。

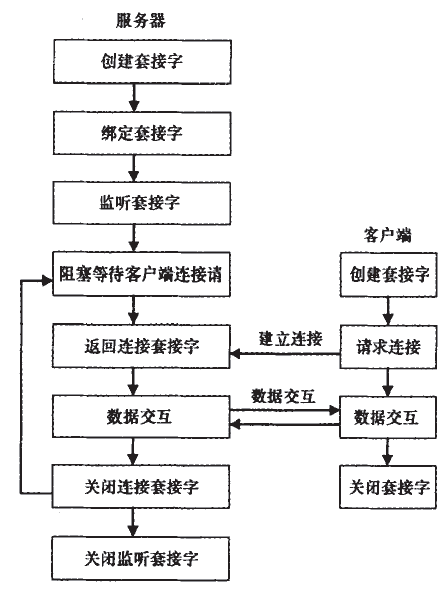


图6-1-2 基于TCP的Socket通信过程图

本系统通过Socket通信，将三个不同模块：网页、手机、数据库服务器之间架起一座通信桥梁，实现多平台下的通信。

### 6.1.3 IO多路复用

采用IO多路复用，我们可以在利用TCP 三次握手时间间隙并发处理其它连接请求、建立多个连接，这样不仅提高了服务器的并发性还增强了服务器的负载能力。目前，在Linux上开发的非阻塞Socket IO多路复用主要有三种：select、poll和Epoll。select和poll用于受并发数限制和轮训结构的低效率以及内存拷贝的问题，效率和性能能不如基于建立链表和红黑树的基础上实现的Epoll[11]。

向来在Linux下设计并发网络程序的方法非常多，例如典型的apache模型、nigix模型和TPC模型以及select模型和poll模型，那为什么要使用Epoll IO多路复用呢？ 我们可以从以下三个方面加以分析：

首先，采用 Epoll IO多路复用不会受到最大并发连接数的限制，其上限是Linux系统最大可打开的文件数。第二，效率方面，Epoll仅负责所谓活跃连接，与连接总数无关，这一点与传统的select和poll相比，Epoll的效率明显大大提高。最后，我们都知道频繁地在用户态和内核态切换用是非常耗性能的，而Epoll通过共享内存的方式实现内存拷贝，并不需要在内核态与用户态之间进行拷贝。

综上所诉，本系统的数据库服务器端采取效率最高、性能最优的Epoll IO多路复用技术。

具体而言，Epoll的实现通过C库封装的3个Epoll系统调用，它们分别是：

（1）int Epoll\_create(int iSize); //创建Epoll文件描述符

（2）int Epoll\_ctl(int iEpfd, int iOp, int iFd, struct Epoll\_event \*event); //Epoll控制函数，控制Epoll事件的增删改

（3）int Epoll\_wait(int iEpfd, struct Epoll\_event \*events,int iMaxevents, int itTimeout); //等待Epoll事件发生

### 6.1.4 SQLite3数据库

本项目分为Android开发的手机端、Linux C开发的服务器端和用ASP.Net开发的网页端的跨平台产品质量管理系统，功能比较单一集中，主要是针对产品质量管理方面的。因而存储的数据并不会很大，但是对于跨平台的支持性要求比较高和访问数据库的速度要求比较高。这些特点就需要一种支持小产品的、易于安装管理的、支持跨平台比较好的数据库[11]。对于开发人员来说，SQLite3拥有易于使用的API，容易上手。对于数据库的维护来说，SQLite3是一款体积小、轻量级的数据库，因而几乎不需要人员管理。SQLite数据库不仅可以通过配置来优化数据库，还具备其他数据所没有的特点： SQLite可以创建内存数据库，即当我们使用数据库时，可以在内存中创建一份备份数据库，当关闭数据库前再将内存数据导出到磁盘数据库，这必然会加快数据的引擎速度。

### 6.1.5 ASP.Net

(1)DIV+CSS布局

DIV+CSS是我们常用的符合W3C标准的网页布局方式。DIV+CSS布局将控件和页面的样式代码写于CSS样式文件中，将JavaScript脚本写在js文件中，使得页面体积容量大幅度变小，将所有页面或所有区域统一用CSS文件控制，能够有效地避免不同区域或不同页面体现出的效果偏差，使得页面效果比较一致，从而有利于维护和管理网页代码。相对于表格嵌套的方式，DIV+CSS将页面划分为更多的区域，页面逐层加载，这样无形中加快了浏览速度[14]。另外，我们把样式文件置于网页代码的头部，将脚本文件置于网页代码的尾部，这样用户打开页面的时候便能先看到整个页面的渲染效果，增强了用户体验。

(2) jQuery

JavaScript目前被广泛地应用于Web开发中，随着HTML5技术的发展，JavaScript在未来还将有更大的发展和应用空间。而jQuery是时下最流行的JavaScript库，据调查，互联网中近一半的网站都使用了jQuery[15]。一方面，使用jQuery，开发者的编码工作将大大减少，而大量的jQuery插件，特别是jQuery-ui，使得开发者可以轻易实现很多绚丽的效果[16]。另一方面，学习jQuery不需要开发者投入太多，就能够迅速开始开发工作，而且使用jQuery实现了脚本与页面的分离，有利于代码管理维护。考虑到，产品质量管理系统网站的开发量并不大，实现的是一些基本功能，并需要开发一些原生的、具有特殊功能的插件，主要是网站整体布局，按钮效果更复杂一点的统计分析图标以及实施刷新的动态效果，jQuery库以及jQuery-UI库的插件已足够使用[17]。

(3) Ajax

总所周知，如果没有Ajax，客户端响应某个服务器控件时，就必须请求整个页面，服务器会根据一定的流程生成页面，发往客户端.。而客户端浏览器进行编译后，呈现出网页页面。整个过程是一种同步的方式，即客户端必须等待服务器响应之后，在收到服务器页面回发之前，客户端会一直处于阻塞。如此一来，不仅效率低下而且用户体验极其不好。我们能有什么方式解决了？Ajax框架就应用而生了。Ajax是一种前端技术，通过Ajax可以轻易实现异步无刷新通信。Ajax主要用于整个网页的局部更新，不需整个页面的全局刷新，请求服务器的时候仍然可以异步进行其他操作。而且使用Ajax的页面，往返数据量小，不会出现屏幕抖动，刷新效果好,具有良好的用户体验。Ajax的实现有很多种方式，而本系统所使用的是VStudio2010集成的Ajax控件。

## 6.2数据库服务器的实现

### 6.2.1概述

服务器按照功能可以大致分为如下几个模块：主程序模块：server.c，业务处理模块：process.hc、process.h，Socket通信模块：mySocket.c、mySocket.h，SQLite3数据库访问模块：SQLite3Helper.c、mySocket.h，线程池模块：Threadpool.c、Threadpool.h，字符串处理模块：parseString.c、parseString.h，调试模块：Debug.c、Debug.h，Makefile文件与程序执行的shelll命令文件ss.sh。服务器运行在嵌入式环境中（本项目主要是在Arm开发板上），需要以下条件支持：Source Insight 3.5、VMWare 7.1.0虚拟机、Fedora 14操作系统、Tiny6410 Arm11开发板。代码工程如图6-2-1所示。

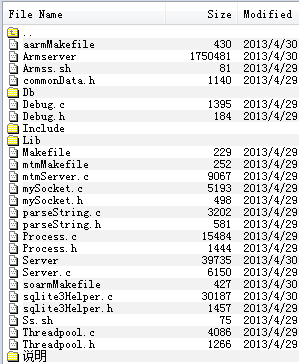


图6-2-1 数据库服务器Linux C开发的代码工程截图

### 6.2.2主程序模块

首先设置允许打开的最大文件数，建立一个SOCK\_STREAM套接字描述符，并设置为套接字接描述符为非阻塞模式，然后将这个套接字描述绑定上服务器端的IP地址和端口号，接着监听队列中的客户端TCP通信连接请求。由于服务器端用的是Epoll多路复用，我们必须创建 一个Epoll 句柄，把监听套接字加入到 Epoll 集合里，然后通过Epoll\_wait函数等待事件发生，当Epoll集合中的套接字接收到客户端连接请求时，触发一个Epoll事件。此时，对被触发的对应套接字进行判断，如果该套接字是之前的监听套接字，我们将其与发出连接请求的客户端进行绑定，建立一个服务器端与客户端通信的新套接字，否则将该套接字作读写状态切换，或者销毁。我们根据服务器端与客户端通信的新套接字，从线程池中获得一个子线程，处理相应的业务流程，处理完业务操作后，线程池将该线程回收再利用，主线程任然处于等待事件的无限循环中。

### 6.2.3业务处理模块

业务处理模块主要是调用其他各个模块的函数来完成一定的业务操作，具体有：调用线程池模块的函数接口，为相关业务分配线程；调用字符流处理函数，截取头部标志信息、解析字符流各个参数；调用数据库访问模块的函数实现与SQLite3的数据交互；对非法访问屏蔽、容错处理等操作。其中，最重要的任务是对获取的数据包按照制定通信格式进行解密解包和对将要发往目标客户端的数据进行封包加密处理。

### 6.2.4 Socket通信模块

Socket通信模块分为Socket、Bind、Listen、Accept、setSockNonBlock、myRead、myWrite这几个函数分别实现，建立套接字、绑定套接字、监听套接字、接受客户端Socket连接请求、设置套接字为阻塞方式以及读取客户端传来的字符数组存放在缓冲区中与将缓冲区中的字符数组写到Socket中。

### 6.2.5线程池模块

线程池模块首先开辟一个线程池管理者与一个线程池：

typedef struct worker

{…} Thread\_worker;

typedef struct

{…} Thread\_pool;

然后是维护线程的基本操作，依次有：线程池初始化与销毁、线程管理者与线程池的绑定、分配线程。

### 6.2.6字符串处理模块

字符串处理模块包括按照自定义的通信协议分析缓冲区中字符流的参数合法性函数、分析字符流头部信息的合法性、截取头部信息、截取字符流中指定字段的函数以及字符流出错处理函数。

### 6.2.7数据库访问模块

SQLite3数据库访问模块作为数据库服务器的底层实现了数据的存储维护等基本操作，为业务逻辑提供可以直接调用的函数接口，分别从以下几点加以说明：

（1）密码设置

为了确保数据的安全性，必须进行密码设置。

SQLite3\_key(qm.db,"zld",3); //设置密码为zld

SQLite3\_rekey(qm.db,"zld",3); //重新设置密码为zld

（2）创建内存数据库

SQLite3可以创建内存数据，先将磁盘数据库读出来，然后在内存中备份一个，数据库操作结束后，再将内存数据库的数据，备份到磁盘数据[18]。本系统的内存数据库的实现通过一个函数加以实现:

MemDb(SQLite3 \*sqlMemDB, const char \*cpcDiskDB, int iSaveOrLoad);

其中sqlMemDB为内存数据库的索引，cpcDiskDB为磁盘数据的存储路径，iSaveOrLoad为读入内存和存会磁盘的选项（0代表读入内存，创建内存数据；1代表存会磁盘数据库）

（34）数据增删改查等基本操作

由于这些操作比较和其他数据的操作相似，详见代码，此处仅对一些特殊的操作进行阐述。

（A）获得自增字段的插入值

在增加员工的时候，由于员工号字段设置为字段；在增加一个班次表的时候，由于班次号设置为自增值，则也需要马上获得该自增值；类似地，品质表中的品质号、工种表中的工种号……，这在SQLite3数据库编程中中必须通过“SQLite3\_last\_insert\_rowid()”这个API函数获得。

（B）释放内存

由于本系统网页客户端，需要频繁地获得大量数据用以处理和显示，这就需要调用的数据库函数：SQLite3\_get\_table，这个函数必须及时释放内存，否则会很快耗尽数据库，最终导致SQLite数据崩溃。这两个释放内存的函数分别是：

SQLite3\_free(Msg);//用于释放存储错误信息的内存

SQLite3\_free\_table(dbResult);//用于释放存储查询结果的内存

当调用SQLite3\_exec函数（该函数主要用于增、删除、修改数据）时，也需要执行SQLite3\_free(Msg);释放错误信息的内存。

（C）获得本地时间

在创建本次表的时候，有一个班次创建时间和班次结束时间用到。

具体函数是datetime(CURRENT\_TIMESTAMP,'localtime')不要用datetime('now')（UTC时间，没有本地化）。

### 6.2.8调试模块

由于服务器是在交叉编译环境下，进行开发的，因此调试信息只能通过回显的形式打印输出，然而过多的打印信息势必会影响调试的效果，因此需要一个通过统一的调试函数来开关回显信息，函数设计如下：

//Debug.h Debug.c

//打印当前时间

void DisplayCurTime();

//打印调试信息

void DebugString(char \*);

//打印变参函数的调试信息

void Debug(char \*fmt, ...);

### 6.2.9 编译、移植和运行

考虑到交叉编译比较麻烦，故在VMware上开发数据库服务器，编译运行成功后，再进行交叉编译，最终，将数据库服务器和数据库移植到Arm板上面运行。然而，实际操作中，考虑到两种环境的差异性，需要不断在虚拟机上和Arm板上面进行尝试和调试。

我的做法是：在虚拟机上，针对虚拟机和Arm板2两种环境分别写好Makefile文件，编译生成数据库服务器可执行文件，然后由shell文件执行生成的可执行数据库服务器文件，这样会避免较多麻烦。其中， 虚拟机上面运行数据库服务器端需采取动态编译。

在嵌入式环境下，通常采用静态编译的方式，虽然软件体积与动态连接方式相比要大得多，但是方便移植，不需要外部文件的支持，可以独立运行。下面介绍一下，交叉编译的过程。首先，安装交叉编译工具arm-linux-gcc-4.3.3，执行两条shell命令：(1)cd /opt/ (2)tar xvzf arm-linux-gcc-4.3.2.tgz –C / ，便能在/opt/FriendlyARM文件下面获得我们所需要的arm-linux-gcc交叉编译工具。然后，到SQLite官网上，下载一个源码包（比如说SQLite-autoconf-3070603.tar.gz），解压源码包到/opt目录下，建立一个文件夹（比如说SQLiteCross），进入解压后的文件夹，执行如下configure命令：./configure –-host=arm-linux –prefix=/opt/SQLiteCross，生成相应的Makefile文件后，接着执行make命令、make install命令，然后便会在之前所建的文件夹/opt/SQLiteCross下生成include、lib、bin、share这4个文件夹。交叉编译时，我们主要用到的文件有bin文件夹下面的SQLite3、include文件夹下的SQLite3.h和lib文件夹下面的库文件。其中，bin文件夹下的文件SQLite3是SQLite3数据库的可执行应用程序，include文件夹下面的SQLite3.h文件是我们编译编译的时候所需要的数据库头文件，而lib文件下面包括静态编译所需要的库文件：libSQLite3.a、libSQLite3.la，动态编译所需要的库文件：libSQLite3.so.0、libSQLite3.so、libSQLite3.so.0.8.6。最后，把libSQLite3.a和SQLite3.h这两个静态编译需要的文件拷贝到工程目录下面，并建立相应的文件夹存放。

## 6.3手机客户端的实现

### 6.3.1概述

从用户的角度看，手机端的客户端主要是品管员在生产中采集质量数据的简易APP，必须符合：操作便捷、反应灵敏、准确性高、界面简单美观等特点。从软件设计的角度看，手机端的开发并不会太复杂，采用传统的三层架构足以满足开发要求，但是容错性高，快速反应等特点又对开发提出了较高的要求。针对高容错性的特点，需要增加异常处理模块，专门处理各种异常情况，例如数据传输丢包、网络不通达、操作失误、手机没电等棘手情形。另外，在设计函数、实现函数的时候，必须注意返回值、传递参数、初始化、类型兼容等问题，反复验证之后才能算完成，以确保对各种异常的周到考虑。针对软件的反应速度，鉴于本系统数据传输受网络情况的制约较大，结合软件开发的特点，本系统的手机客户端在设计Socket通信模块的同时，还采用了多线程以及本地缓存的方式以加快页面的加载速度。多线程是将与数据库服务器交互的部分剥离出来，开辟专门的线程处理数据。本地缓存是将不经常变动的数据保以CSV文件的形式存储在手机本地，对于将要上传到数据库服务器上的大数据暂时保存在本地。如此一来，忙碌中的品管员便不会感觉到“卡”了。下面分别从界面模块、Socket通信模块、本地缓存模块、业务逻辑等四大模块分别进行介绍。

### 6.3.2界面实现

手机客户端的界面共有5张，分别是：登录界面LoginForm、系统设置界面SystemConfigForm、员工列表界面EmployeeList、建立班次后的员工列表SelectedEmployeeList、产品质量登记界面RecordProductDetail，还包括手机客户端LOGO。整个手机客户端的色调为蓝色，所有成功操作弹出非模态提示框，错误操作弹出模态提示框。

（1）登陆界面使用白色背景的文本控件来输入用户名和密码，下拉框选择车间信息，透明蓝的登录按钮，并设置跳转到系统设置界面的入口。密码错误将会弹出模态提示框，登录成功则弹出非模态提示框2秒后消失。

（2）系统设置界面使用文本控件作为IP和端口的载体。

（3）员工列表界面、建立班次后的员工列表和产品质量登记界面都选用ScrollList作为载体来呈现员工信息，如果手机屏幕无法全部列出数据项，则用户可以通过上下滑动来查看数据。

### 6.3.3面向连接的Socket通信模块

鉴于用户的需求以及用于体验的考虑，Socket通信设计成非阻塞方式，数据通信的同时，用户可以进行其他操作。延时超过两秒，放弃本次通信，重新发包，数据传输失败次数达到3次，提示用户操作失败。该模块作为手机客户端的底层，负责业务逻辑层与数据库服务器之间的数据通信，对于数据仅负责传输，并不负责解析等业务操作。主要处理流程是：先将IP与端口号与套接字绑定起来，然后对数据进行网络字节序与主机字节序进行转换后，得到目标格式的数据后交由业务逻辑层处理或者发往数据库服务器。主要数据成员包括strServerIP（IP信息）、iServerPort（端口号）、iTimeOut（超时设置）以及strToServer（发往数据库服务器的字节流）、strFromServer（来至目标数据库服务器的字节流），主要函数是负责通信的函数communicate具体如下：

'手机客户端与服务器通信1次

'备注：通信后，自动关闭套接字

'传入参数1:data,传给数据库服务器端的数据 类型：String

'传入参数2:s,已经与服务器端建立通信的套接字 类型：Socket

'返回值：从服务器端接收到的数据信息 类型：String

Sub communicate(dataToServer As String ,connectedSocket As Socket) As String

### 6.3.4本地缓存模块

本地数据与CSV文件交互主要是通过StringUtils对象以及它的4个函数实现的，分别如图6-3-1、6-3-2、6-3-3、6-3-4。

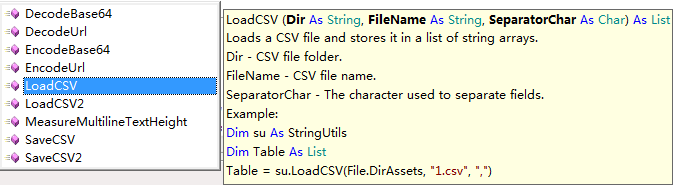


图6-3-1 带表头的保存函数使用截图

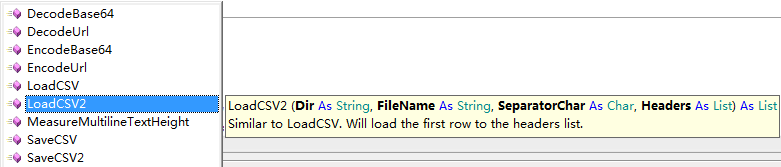


图6-3-2 不带表头的保存函数

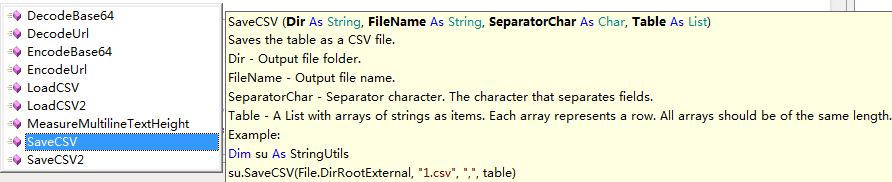


图6-3-3带表头的存储函数

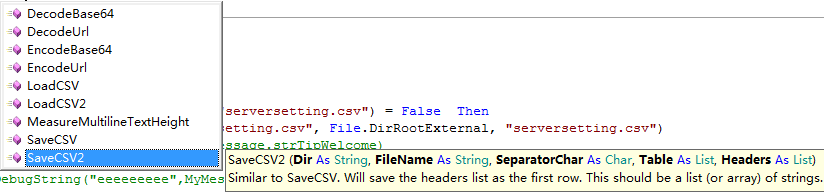


图6-3-4不带表头的存储函数

这四个函数的特点就是：设计好的API接口、适合便捷存储简单的数据，正好适合本系统手机端本地缓存的需求。可以通过分隔符即表头设计，灵巧有序地存储数据。主要缓存函数如下：

Dim StringUtils1 As StringUtils '用于配置文件交互

'从本地保存的CSV文件中读取车间信息，并显示到下拉框控件中

Sub LoadTableFromCSV(Dir As String,Filename As String,HeadersExist As Boolean)

'保存用户配置文件

Sub save\_user\_setting

'读取用户配置文件

Sub load\_user\_setting

'保存系统设置

Sub save\_setting

'载入系统设置

Sub save\_setting

### 6.3.5业务逻辑模块

由于手机客户端的功能仅仅是采集质量数据，这就决定了业务逻辑层比较简单，其主要任务是根据自定义的通信协议进行封装数据包和解拆数据包，去掉包头包尾部分，按照一定格式拼接成目标格式的字符串，显示到界面层和传递给下一层：Socket通信模块或者本地缓存模块，具体的通信协议参见网络连接设计部分。该墓开具体实现函数有三大块：一块是与通信底层交互，另一块是与本地缓存交互，还有一部分是为满足界面层显示需要的。

### 6.3.6工程代码架构

本系统的手机端采用三层架构。数据访问层包括：本地缓存的CSVHelper、Socket通信的Myscoket。业务逻辑层：包括Process模块，字符串处理模块stringProcess、消息控制模块MyMessage、异常处理模块MyException。表现层包括：登录界面LoginForm、系统设置界面SystemConfigForm、员工列表界面EmployeeList、建立班次后的员工列表SelectedEmployeeList、产品质量登记界面RecordProductDetail。最后，设计了各层交互的重要数据模块CommmonData以及调试模块MyDebug。具体代码工程文件如图6-3-5。

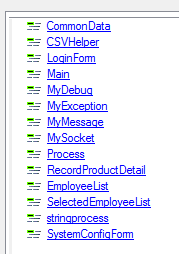


图6-3-5手机客户端代码工程截图

### 6.3.7编译与运行

编译生成apk文件之前，先设置好以下参数。

（1）Package name: qm1.0.zld

（2）Choose Icon:选择一张事先制作好的图片作为应用程序图标

（3）Application Label：产品质量管理系统1.0

（4）Application Version：1.0

## 6.4网页客户端的实现

### 6.4.1概述

优秀的网页不仅需要具备精致合理的设计、完善的功能、漂亮的前端，还需要较高的性能，良好的用户体验。为达到这些效果，这就需要采用一个好的架构和策略加上关键性技术以及细致地实现才能完成。首先，我将产品质量管理系统的前后端的功能独立开来,这样一来，整个架构层次清晰、便于代码管理与维护。

### 6.4.2网页前端

前端是展示层，主要负责数据显示的职责，采用DIV+CSS布局，结合jQuery和Ajax技术实现清晰的布局、优质画面和良好用户体验。本系统的网页客户端共有三个aspx页面：Login.aspx、qmDefault.aspx、Administrator.aspx。

（1）Login.aspx页面是登录页面，该页面布局简单明了，并对用户名与密码的合法性进行核查，以提示用户。

（2）Administrator.aspx是系统管理员的界面，对网页布局要求不高，主要通过选项卡将员工信息、车间信息等系统信息分块，然后各个模块选择适合的gridview控件展示数据，因为与repeater等控件相比，gridview不仅能够方便地分页展示数据，还轻松地能够对数据进行增加删除、编辑的常见操作。（3）qmDefault.aspx页面是本系统网页客户端的主页面，这里重点介绍，其设计思想具体如下：

1）用DIV+CSS布局一个经典的自适应的左中右结构，中间部分又分为上中下。上部为产品质量管理系统的Logo，通过photoshop设计好后置入网页头部，中间部分用jQuery的选项卡插件布局2）用Repeater控件显示产品质量数据的统计分析指标如合格率等，配合饼状图、条形图、折线图等插件显示质量数据，并增加了导出数据到Microsoft Excel文件等附加功能3）将Repeater控件的内容嵌入Ajax控件UpdatePanel的ContentTemplate标签。在UpdatePanel容器内部安置一个定时器，设置定时器的刷新周期，然后通过一个ScriptManager来统一管理脚本，就可以局部刷新，实时展示质量数据情况了。以此方式实现的局部刷新同样适用于实时显示班次情况。4）由于该页面实现的功能较为复杂，为便于开发和代码管理以及重复利用，本系统使用用户控件对具有独立模块的部分页面进行封装，以独立的ascx页面存在，使用的时候，在aspx页面中注册之后便可以使用了。

### 6.4.3网页后端

后端文件放在APP\_Code文件夹 中，包括字符串解析和Socket通信基本操作以及业务操作，可以将Socket通信获得数据部分看作传统三层架构中的数据访问层（只是访问数据库换做Socket通信获得数据）；其他部分看作业务逻辑层，这一部分主要包括统计分析的策略与算法封装的函数和类。

#### 6.4.3.1通信底层

所有业务逻辑部分与服务器交互都是基于底层通信实现的，该层自定义一个communicate函数，主要负责接收从业务逻辑层传递过来的数据，然后根据设定的IP和端口号发往目标服务器。操作过程：首先，读取配置文件web.config中配置的IP和端口号节点信息与一个套接字绑定起来。然后是典型的TCP/IP通信过程，将根据通信协议处理过的字符串转化为字节流通过send函数发往服务器，通过receive函数接收数据，更具体的实现不加以赘述了。

#### 6.4.3.2业务逻辑层

该层主要任务有两个：一是根据一定业务逻辑处理从界面层获得的数据，交接给通信底层，其中业务逻辑包括封包和解析、拼接字符串以及满足合格率、次品率等各种统计指标的要求；二是从通信底层获得从服务器端传递过来的原始数据，进行解包，解析、拼接字符串，建立和绑定repeater、gridview等控件的数据源以满足界面层的显示要求，并核算分类各种统计数据。该层主要函数有两类：

（1）ct()、cg()、cb()、ci()、cw()、starCw()

（2）parseCtStr()、parseCgStr()、parseCbStr()、parseCiStr()、parseCwStr()、parseStarCwStr()

其中，以”parse”打头的一类函数主要负责字符串的处理并按照通信协议解包、封包；另一类则主要负责按照一定统计指标计算数据以及绑定数据源等工作。具体函数意义参见网络连接设计部分。

### 6.4.4工程代码架构

根目录下，存放lib文件、CSS文件、网页aspx文件以及配置文件。lib文件下面，存放jQuery库文件、插件脚本文件以及自编写的脚本文件。CSS文件下面，存放样式文件和图片文件。具体如图7-12。

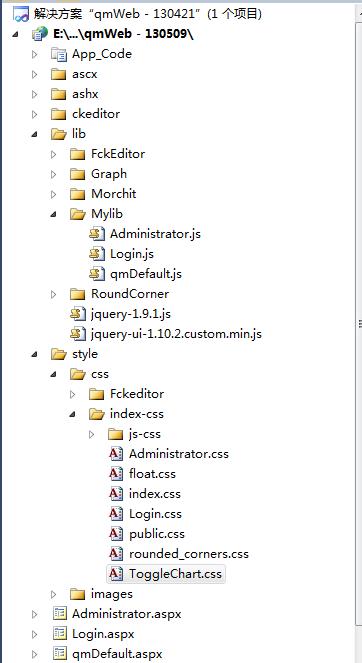


图7-12网页客户端代码工程截图

### 6.4.5编译与运行

（1）通过web.config文件的一个节点来配置服务器的IP地址和端口号以及系统设置参数等，具体如下：

<appSettings>

<add key="IP" value="172.17.200.18" />

<add key="port" value="1664" />

<add key="freshInterval" value="60" />

</appSettings>

（2）编译VS2010下的项目文件，发布网站到网页服务器中，配置好网页服务器以及局域网，就可以通过浏览器访问网站了。

# 7 系统使用说明

## 7.1运行数据库服务器

由于数据库服务器在虚拟机上的运行效果与Arm板上的效果基本类似，为方便说明，这里仅仅针对虚拟机上运行数据库服务器进行截图说明。具体步骤如下：

（1）首先，配置好网络。然后，在shell命令窗口中，运行服务器端命令：/bin/bash ss.sh

（2）运行服务器端将会出现图7-1-1至7-1-4情况。

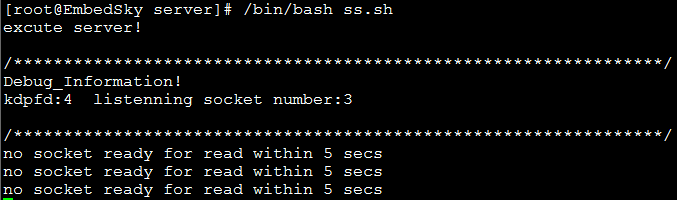


图7-1-1没有客户端访问时数据库服务器运行截图

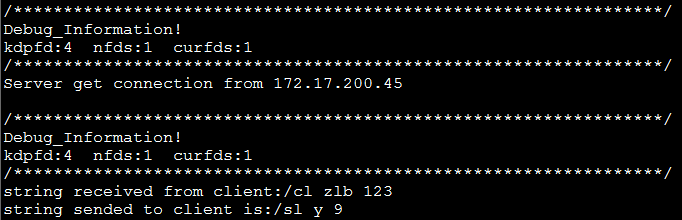


图7-1-2系统管理员发出连接向数据库服务器发出连接请求时的运行截图

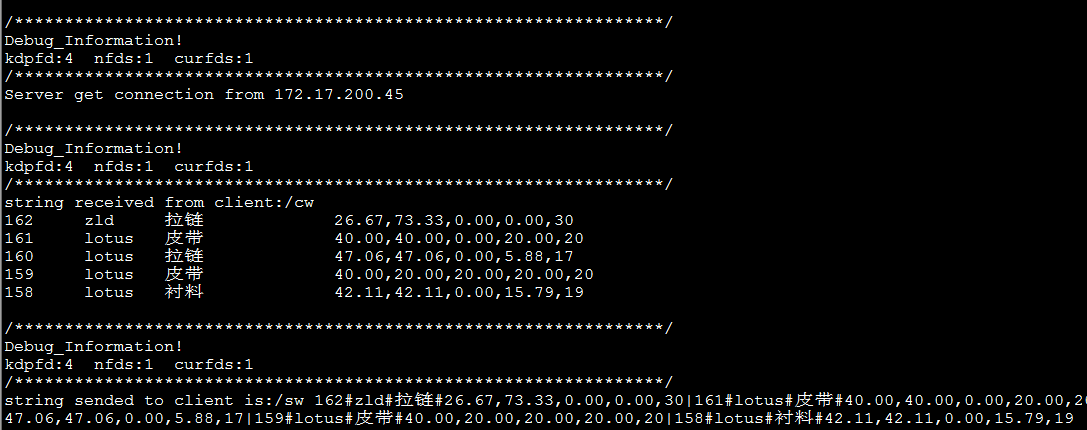


图7-1-3数据分析员发出连接向数据库服务器发出连接请求时的运行截图

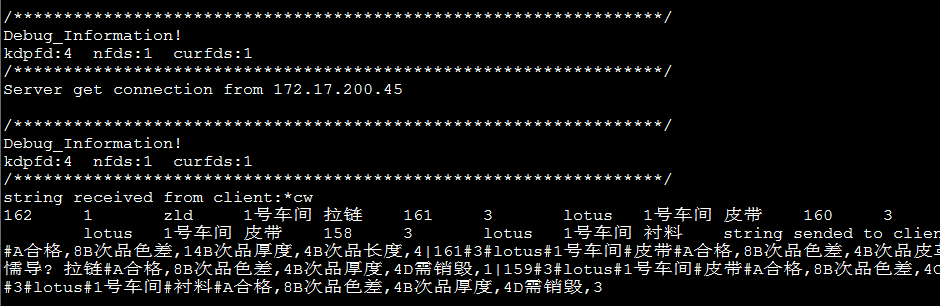


图7-1-4经理发出连接向数据库服务器发出连接请求时的运行截图

数据库服务器图7-1-1至7-1-4的打印截图仅仅是为了显示运行效果，在实际运行的时候必须要将这些打印信息屏蔽掉，因为IO操作必然占用较大的服务器资源，这对服务器来说往往是致命的。

## 7.2采集质量数据

首先产品质量管理员必须打开手机的Wifi，连接到公司的无限局域网，然后产品质量管理员打开手机APP客户端后，APP客户端的图标如图7-2-1所示。



图7-2-1 手机APP客户端图标

输入用户名、密码，选择车间，点击登陆按钮后，就可以登陆产品质量管理系统，登录界面如图7-2-1和7-2-3。



图8-2-3手机客户端登录成功界面截图



图8-2-2手机客户端登录界面截图

当车间信息、品质信息、职位信息以及服务器IP与端口号等发生变化时，必须更新以CSV形式存储在本地的车间表、品质表、职位表，重新设置服务器IP与端口号，截图如图7-2-4、7-2-5、7-2-6。



图8-2-5更新手机客户端系统设置截图

图8-2-6更新手机客户端品质信息截图



图8-2-4更新手机客户端车间信息截图

品质管理员通过下拉框选择某一车间后，可以进入员工列表界面如图7-2-7，在该界面可以上下滑动界面进行选择将要创建班次的员工。

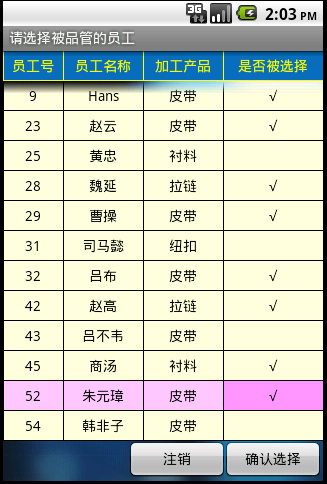


图7-2-7员工列表界面截图

选择员工建立班次后，跳转到已建立班次的员工列表界面如图7-2-7。



图7-2-7班次员工列表界面截图

品质管理员针对某员工生产的产品进行品质鉴定后，选择班次员工列表的该员工行，则跳转到产品质量表界面如图7-2-9。在该界面中，品管员可以根据实际品质情况，填写品质情况上传到数据库服务器。



图7-2-9 品质列表截图

等到下班的时间，品质管理员可以进行下班操作已结束该班次，如图7-2-10。

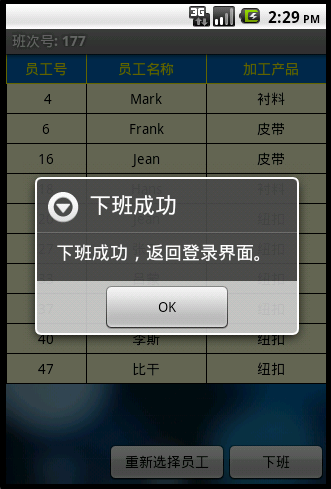


图7-2-10下班界面截图

## 7.3显示分析数据

数据分析人员、经理、系统管理员可以通过不同浏览器登录到产品质量管理系。下面以Firefox浏览器为例访问本系统网页客户端。

首先，本系统不同角色访问登录界面，登录界面如图7-3-1。然后，输入正确的用户名和密码后进入相应页面。



图7-3-1登录界面截图

根据用户的操作，点击登录按钮时，可能会遇到“登录成功”、“登录失败”、“密码不能为空”的提示框，分别如图7-3-2、7-3-3、7-3-4，0.5秒后自动跳转。



图7-3-2 “登陆成功”截图



图7-3-3“密码错误”截图



图7-3-4“密码不能为空”截图

下面分别从系统管理员、数据分析人员、经理这三种角色来介绍使用本系统网页客户端的操作步骤。

**（1）系统管理员**

系统管理员登录到本系统后，将会出现管理员工、车间、品质、工种信息、数据库备份的5个不同选项，如图7-3-5。



图7-3-5 系统管理员主界面截图

其中员工、车间、品质、工种信息界面基本类似，下面例举员工管理界面来说明。员工界面展示的是员工列表的分页列表如图7-3-6（红色格子为异常信息），系统管理员可以对员工进行增加、编辑、删除、排序。

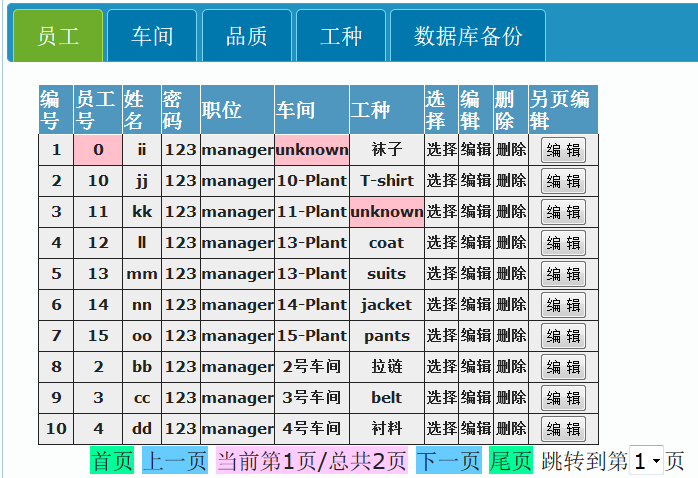


图7-3-6员工信息管理界面截图

系统管理员还可以进行数据库备份，操作截图如图7-3-7、7-3-8、7-3-9



图7-3-7数据库备份界面截图

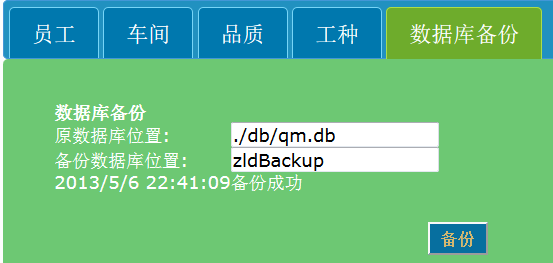


图7-3-8数据库备份成功界面截图



图7-3-9数据库备份路径为空或者非法界面截图

**（2）数据分析人员**

数据分析人员登录到本系统将会出现一个选项卡，分别有“BI”、“今日班次”，”今日品质“、”分析“、”查询“、”设置“和“主页“这6项功能。

数据分析人员做出数据分析后，可以在“BI”页面编写数据挖掘报告，然后上传到数据库服务器供经理等人查阅。“BI”截图如图7-3-10。



图7-3-10 BI界面截图

“今日班次”是实时显示当日所建立班次的状况，该页面每4秒刷新一次，如图7-3-11，如果还没有建立班次，将会出现如图7-3-12。



图7-3-11 “今日班次”显示班次情况截图



图7-3-12 “今日班次”没有班次情况截图

”今日品质“是实时显示当日产品质量登记情况，该页面每4秒刷新1次，如图7-3-13。如果还没产品质量登记情况，将会出现如图7-3-14。



图7-3-13 “今日品质”显示产品质量登记情况截图



图7-3-14 “今日品质”没有产品质量登记情况截图

”分析“主要包括四部分，第一部分是以表格形式呈现产品质量的合格率、次品率、可维修率、废品率、总数等统计分析数据，且可以导出Excel数据，如图7-3-15。



图7-3-15 统计分析报表截图

第二部分为动态折线图，用户可以根据需要，选择目标数据的折线图走势，第一张折线图为所有统计数据的折线图，如图7-3-16。用户可以通过右上角的标签进行选择，就可以得到第二张关于合格率和总量的折线图如图7-3-17，其他走势图类比7-3-17。

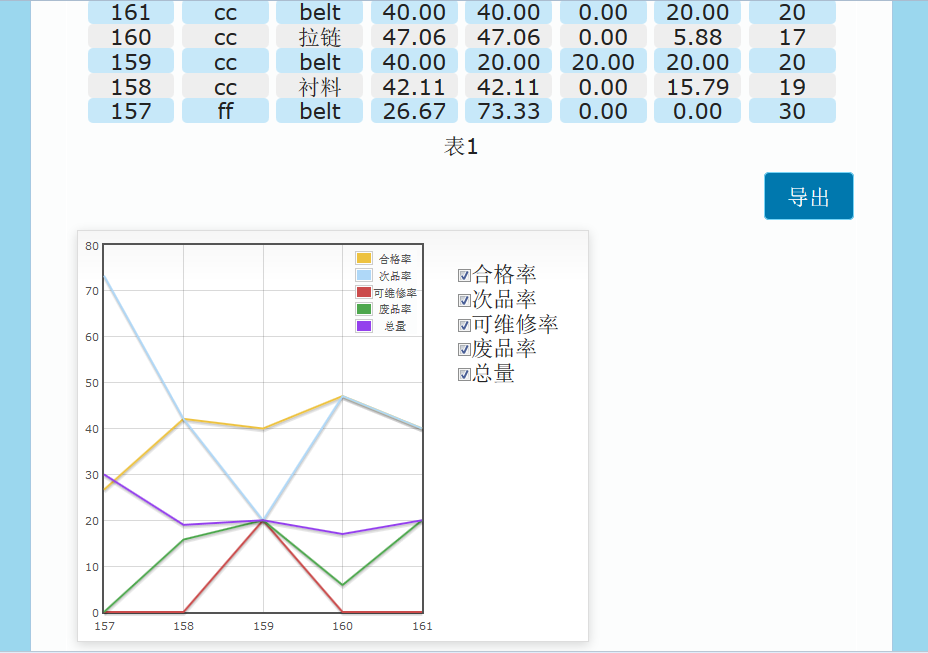


图7-3-16 所有统计数据折线图

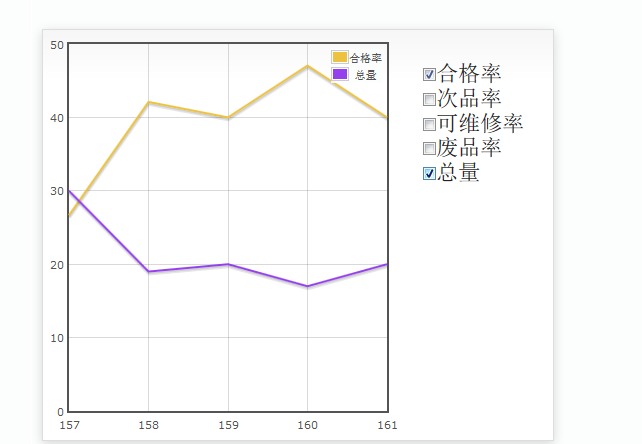


图7-3-17 合格率-总量走势图

第三部分是关于合格率、次品率、可维修率、废品率的饼状图，如图7-3-18。

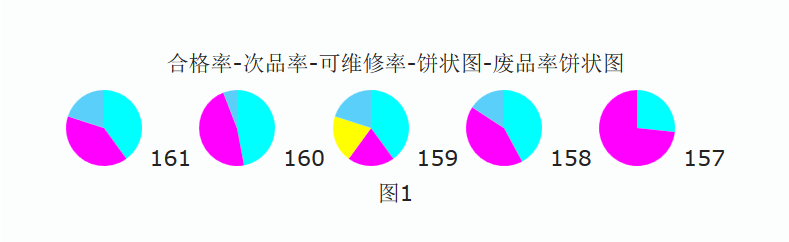


图7-3-18 合格率-次品率-可维修率-废品率饼状图

第四部分为总量的柱状图，如图7-3-19。

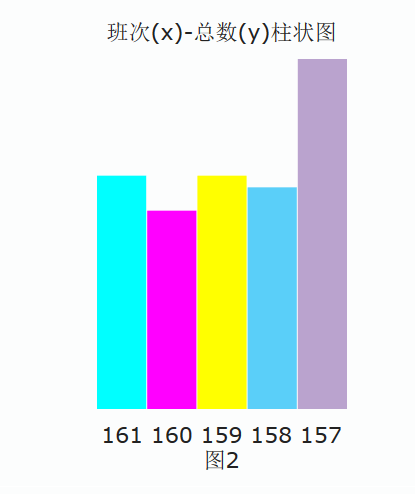


图7-3-19 总量柱状图

”查询“界面的功能主要是以班次、产品、日期为查询条件对产品质量记录情况进行查询，如图7-3-20。



图7-3-20查询后数据呈现截图

“设置”包括刷新时间间隔设定、密码修改等基本操作。



图7-3-21 设置界面截图

主页包括新闻、通知、公告、公司介绍等信息，如图7-3-22。



图7-3-22主页截图

**（2）经理**

经理登录后进入的主界面与数据分析人员的界面大同小异，区别在于经理有最高权限，可以查看机密的BI信息，其BI界面截图，如图7-3-22。其他界面与数据分析人员相似，请参照数据分析人员部分。



图7-3-23 经理BI界面截图

# 8总结

通过这个项目我觉得项目需求及整体架构至关重要，如果开始着手进行项目时，需求还不明确对于后期的开发工作影响是非常大的，由于之前产品质量管理系统的需求不是很明确，导致后来反复修改，严重影响开发效率。

服务器端系统性能要求感受很深。首先，服务器端最大的挑战就是服务器多并发的要求，并针对各种客户端做出差异性服务。然后，服务器端必须确保不能崩溃，这就要要求服务器内部具备相应的预防机制，例如如何预防多客户端同时请求以及大数据请求造成的服务器拥堵，并预防这种情况下造成的错误。由于是把服务器放在Arm板上运行，所以应尽量提高服务器运行速度，减小程序大小，对于代码质量要求也比较高。让我感受最深的是：开发时，环境搭建及网络配置很重要也比较麻烦，还有就是Socket通信时，协议的制定也很重要，这样才能与客户端很好地交接。再者，服务器端必须面对多客户端，周详考虑到各客户端的差异性与一致性，调试程序与功能测试时必须开启多种工具及开发平台，这是非常麻烦的。最后，服务器的内存管理尤为重要，申请的内存，一定要记得释放，否则随着时间的推移，将会“后患无穷”。

对于数据库，我碰到一个非常棘手的问题，就是Sqlite经常崩溃，这个问题同时也一度让我非常崩溃。Sqlite提示的错误信息是：“the database disk image is malformed”。上Sqlite官网查询了一下，主要原因有三个，其一是磁盘空间不足，这个可以很快否定；其二是访问数据的时候，没有及时释放空间，我重新审查了一遍代码，发现确实有几处调用了sqlite3\_gettable()函数却忘了sqlite3\_free(errmsg);sqlite3\_free\_table( dbResult );其三是数据库没有进行清理，这是Sqlite数据库的缺点，后来通过“PRAGMA auto\_VACUUM”来配置数据库，终于解决了问题。

对于ASP.Net网页开发，由于之前开发过几个项目，所以比较容易上手。虽然说做出来是没问题，做得好却并不容易，而且对于Socket通信还是头一次。总体来说，并没有多大难度，主要是处理Socket通信从客户端传来的字符串做相应的处理后，要比较清楚美观地以网页形式呈现，并且符合客户需求，并且网页反应速度不能太慢，这对于代码总体设计、网页布局也需要花费不少心思。

Android端和ARM服务器通讯的时候不知为何总是被服务端告知传输的字符串格式错误，通过使用WPE和WireShark网络截包工具，最后找到问题所在，其实是Android端Java编写的程序字符串末尾是\n\0,而ARM端的C语言编写的程序则是\0,因此ARM端会多读一个\n出来，导致字符串的匹配失败。

然后就是Android端文件存储问题，使用编译器的时候，可以使用自带的File.DirAssets目录下文件，但是实机运行的时候该目录是只读的，不能进行写操作，因此后来经过调试和多方查证，改为如果File.DirRootExternal目录下目标文件存在则不拷贝直接对那个文件操作，不存在则File.DirAssets的文件拷贝到File.DirRootExternal目录下，这样就能实现本地默认文件以及文件读写功能。

手机客户端把字符串解析成表的时候会出现表的第一行或者最后一行全部被置空的情况，结果多次对比测试，找到问题是字符串头尾出现空格字符，即\32，只需要对需要处理的字符串进行Trim操作去掉字符串头尾的空格符后再处理即可。

# 致谢

感谢吴垠老师对我的指导和帮助，感谢我的家人和同学们对我的支持和帮助。

# 参考文献

[1] [陈祥兵](http://jour.fulink.superlib.net/searchJour?sw=%E9%99%88%E7%A5%A5%E5%85%B5&ecode=utf-8&channel=searchJour&Field=2), 论企业质量管理中的竞争优势[J].科技管理研究，2003,23(1)：1-5

[2] 戴小军，质量管理系统中产品实现过程的研究与软件实现[D].东北大学硕士学位论文，2005：2-30

[3] 百度百科.质量管理[EB/OL], <http://baike.baidu.com/view/58026.htm,2013-04-21>

[4] 郭灵，基于J2EE的产品质量管理系统的开发与研究[D]，河北农业大学硕士学位论文，2011：10-17

[5][6] 徐宏喆 侯迪，实用软件—设计模式教程[M]. 北京:清华大学出版社,2009:248-254

[7]史蒂文斯 拉戈著 尤晋元 张亚英 戚正伟译，UNIX环境高级编程[M] .北京:人民邮电出版社，2000：147-505

[8] 李昊 刘志镜，线程池技术的研究[J].现代电子技术, 2012,31(3):1-5

[9] 赵海 李志蜀 韩学为 叶浩，线程池的优化设计[J].四川大学学报, 2005,42(1):1-4

[10][11] 梁明刚，陈西曲,Linux下基于Epoll+线程池高并发服务器实现研究[J].武汉工业学院学报, 2012,31(3):1-5

[12][13] SQLite中文社区[ EB/OL]，<http://www.SQLite.com.cn/，2013-04-02>

[14] 俸学文 傅汉，DIV+CSS网页布局研究[J] .计算机与信息技术，2012，（2）：1-2

[15] 赵增敏 李惠敏，基于jQuery框架的Ajax应用开发[J] .制造业自动化, 2012,34(19)：1-4

[16] 李志秀 张军 陈光 杨丽红，JQuery Ajax 异步处理JSON 数据在项目管理系统中的应用[J].云南大学学报( 自然科学版)，2011，33(2)：247-250

[17] 曹刘阳，编写高质量代码—Web前端修炼之道[M] .北京:机械工业出版社，2010：2-200

[18] （美）艾伦 （美）欧文斯，SQLite权威指南[M]. 北京：电子工业出版社，2012：10-250

**Product quality management system**

123012009137 Tutor: Wu Yin

(Faculty of Software, Fujian Normal University, Fuzhou)

**Abstract:**Product quality is the eternal theme, to ensure product quality is to adhere to the sustainable development of enterprises. In recent years, many companies in order to make the management of product quality more intelligent, more efficient, have introduced computer-aided system to manage product quality. However, the conditions of the production workshop will inevitably lead to difficult to computer as a data collection terminal, which is like shoe-making enterprises such traditional manufacturing enterprises are facing another difficult. This paper, the product quality management system is a set of product quality system based on Android terminal, as the realization of the carrier to the shoe-making industry. It is considered from the point of view of the quality control staff, is conducive to quality control staff to carry mobile client, in order to collect quality data. Considered from the point of view of data analysis and business, to facilitate quality data analysis and presentation in the web client. Maintenance personnel, developers, and operating costs from the system point of view, the use of specialized embedded hardware and software to crop equipment as management data server, suitable for the small size and fast computing speed SQLite3 database embedded development. First, this paper first analyze the needs of the project, considering the three angles from the market, technical, economic feasibility of the system. Then, from the point of view of the design pattern analysis of the structure of the system, focusing on description of the function of the system module, and use case diagrams and flow charts intuitively reflect the main functions of the system. After a brief description of the key technologies in the system, including the multi-threading technology, the IO multiplexing technology, SQLite database technology, Socket communication technology and ASP.Net certain kinds of technology. Then, we spent a lot of ink on product quality system implementation, respectively, from the mobile client, the database server side, the web client three aspects of the function design, coding, and the concrete realization of the project three-tier architecture. Finally, the paper briefly describes the instructions for use of the system, system testing, and project summary.

**Key words：**Product quality management system; quality control system; Android terminal; footwear product quality management system