

**本科毕业设计（论文）**

题 目

学院名称 计算机科学与技术学院

专业名称 物联网专业

年级班级

学生姓名

指导教师

2019年 5 月

# 摘要

二十一世纪，随着社会的进步和经济的快速发展，人们对煤矿能源的需求越来越大，对煤矿的开采需求也越来越大。各个地方的煤矿如雨后春笋般涌出，传统的信息获取管理方式也渐渐不能满足需求。煤矿信息化逐渐成为如今煤矿行业的主要业务，煤矿数据检测系统也成为了其中比较重要的部分。目前各个煤矿上采用的数据获取方式各种各样，但都存在各种不足。如今的煤矿监控系统一般都是采用RS232、RS485总线或CAN总线实现实时传输，这几种方式不仅无法满足传输长距离的需求，而且成本较高，小型煤矿可能并不需要花费这么高的成本。所以需要一套传输距离长，稳定性高，成本相对较低的系统。

针对以上的问题，本课题所设计的煤矿顶板离层检测分析系统采用了Android端APP+基于WCF+C#的桌面客户端程序。因为桌面程序稳定性高，图像化界面使得用户交互简单易懂。本系统采用移动APP取通过蓝牙端获硬件数据保存成TXT文本文件，通过FTP（File Transfer Protocol，文件传输协议）上传TXT文件至服务器，用C#编写的客户端与服务器交互获取数据，并存入数据库，数据库使用的是SQL Server 2008 R2服务器引擎。各地的煤矿数据通过移动端APP上传数据至FTP服务器，FTP服务器和桌面程序的服务端部署在服务器上（可以使用在云服务器）。本系统使用MD5算法的加密功能设计实现了软件注册用户功能，以保护煤矿数据信息安全，同时保护了软件知识产权。

**关键词**：FTP传输； MD5 加密；WCF架构；桌面程序

**ABSTRACT**

ln the 21st century, with the progress of society and the rapid development of the economy, people's demand for coal mine energy is increasing, and the demand for coal mining is also growing. Coal mines in various places have sprung up, and the traditional methods of information acquisition and management have gradually failed to meet the demand. Coal mine informationization has gradually become the main business of the coal mining industry today, and the coal mine data detection system has become a relatively important part. At present, the data acquisition methods used in various coal mines are various, but there are various deficiencies. Today's coal mine monitoring systems generally use RS232, RS485 bus or CAN bus to achieve real-time transmission. These methods can not only meet the long-distance transmission requirements, but also cost a lot. Small coal mines may not need to spend such high cost. Therefore, a system with long transmission distance, high stability and relatively low cost is needed.

In response to the above problems, the coal mine roof separation detection and analysis system designed by this subject uses the Android APP+ WCF+C#-based desktop client program. Because the desktop program is highly stable, the graphical interface makes user interaction easy to understand. It mainly solves the problem of system versatility, enabling users to access the system in different regions and at different times. The system uses the mobile device to obtain the TXT text file through the Bluetooth terminal, and uploads the TXT file to the server through FTP (File Transfer Protocol), and the client written by C# interacts with the server to obtain data and deposits it. Database, the database uses the SQL Server 2008 R2 server engine. The coal mines everywhere upload data to the FTP server through the mobile terminal, and the server of the FTP server and the desktop program is published on the total server (which can make the cloud server). The system uses the encryption function of the MD5 algorithm to design and implement the software registration user function to protect the mine data information security and protect the software intellectual property.

**Keywords**: FTP transfer; MD5 encryption; WCF architecture; desktop program

目录

[1 绪论 1](#_Toc5834)

[1.1开发背景 1](#_Toc23776)

[1.2研究意义 1](#_Toc1900)

[1.3论文主要工作 2](#_Toc28813)

[1.4本论文的结构安排 2](#_Toc7497)

[2 开发环境以及关键技术 4](#_Toc14185)

[2.1 Android Studio概述 4](#_Toc15867)

[2.2 Visual Studio概述 4](#_Toc18849)

[2.3 开发语言简介 5](#_Toc11791)

[2.3.1 java和Android简介 5](#_Toc31317)

[2.3.2 C#简介 6](#_Toc14788)

[2.4 C/S结构叙述 7](#_Toc7138)

[2.5 WCF详解 7](#_Toc11445)

[2.5.1 WCF简介 7](#_Toc19298)

[2.5.2 WCF框架 8](#_Toc22490)

[2.6 FTP简介与工作原理 10](#_Toc9515)

[2.6.1 FTP简介 10](#_Toc8319)

[2.6.2 FTP工作模式 11](#_Toc4698)

[2.6.3 FTP传输方式 12](#_Toc10287)

[2.7 SQL Server数据库 12](#_Toc8113)

[2.8 本章小结 13](#_Toc23706)

[3 需求分析 14](#_Toc4165)

[3.1 系统开发目标 14](#_Toc30717)

[3.2 系统需求分析 14](#_Toc13018)

[3.3 可行性分析 16](#_Toc32162)

[3.4 本章小结 16](#_Toc30429)

[4 设计与实现 17](#_Toc23397)

[4.1 系统总体设计 17](#_Toc26195)

[4.1.1 总体功能划分 17](#_Toc24680)

[4.1.2 系统模块设计 17](#_Toc5186)

[4.2 系统数据库设计 19](#_Toc31319)

[4.2.1 数据库概念设计 19](#_Toc19293)

[4.2.2 数据库逻辑设计 19](#_Toc30193)

[4.3 数据采集APP各功能的具体实现 20](#_Toc26351)

[4.2.1 APP主页面的实现 20](#_Toc9393)

[4.2.2 数据采集的实现 21](#_Toc20583)

[4.2.3 数据浏览的实现 24](#_Toc6338)

[4.2.4 数据上传的实现 26](#_Toc11757)

[4.2.5 参数设置的实现 28](#_Toc29099)

[4.4 上位机服务端实现 29](#_Toc29410)

[4.5 顶板离层数据分析系统的具体实现 30](#_Toc18266)

[4.5.1 用户管理的实现 31](#_Toc18758)

[4.5.2 数据接收的实现 33](#_Toc28274)

[4.5.3 数据查询的实现 34](#_Toc17233)

[4.5.4 数据报表的实现 37](#_Toc26625)

[4.5.5 系统设置的实现 41](#_Toc32445)

[4.5.6 数据库维护的实现 43](#_Toc8211)

[4.5 本章小结 44](#_Toc28575)

[5 软件测试 45](#_Toc31168)

[5.1 功能测试 45](#_Toc30396)

[5.2 测试用例及测试结果 46](#_Toc23707)

[5.3 本章小结 49](#_Toc18358)

[6 总结与展望 50](#_Toc22193)

[6.1总结 50](#_Toc1422)

[6.2 展望未来 50](#_Toc16252)

[致谢 51](#_Toc28749)

[参考文献 52](#_Toc27513)

# 1 绪论

## 1.1开发背景

.NET技术是[微软](https://baike.baidu.com/item/%E5%BE%AE%E8%BD%AF" \t "https://baike.baidu.com/item/.NET/_blank)公司为了实现[XML](https://baike.baidu.com/item/XML" \t "https://baike.baidu.com/item/.NET/_blank)（Extensible Markup Language，即可拓展标记语言），[Web Services](https://baike.baidu.com/item/Web Services" \t "https://baike.baidu.com/item/.NET/_blank)，[SOA](https://baike.baidu.com/item/SOA" \t "https://baike.baidu.com/item/.NET/_blank)和敏捷性而发明的架构性技术。对于技术人员来说，要学习.NET，一定要先学习XML,Web Services 和 SOA技术，一定要先学习.NET技术的来源和能用.Net解决的问题。[微软](https://baike.baidu.com/item/%E5%BE%AE%E8%BD%AF" \t "https://baike.baidu.com/item/.NET/_blank)公司搭建[技术平台](https://baike.baidu.com/item/%E6%8A%80%E6%9C%AF%E5%B9%B3%E5%8F%B0" \t "https://baike.baidu.com/item/.NET/_blank)，然后技术人员在这个平台上建立自己的应用系统。换个角度，.NET可以解释成这样：.NET是[微软](https://baike.baidu.com/item/%E5%BE%AE%E8%BD%AF" \t "https://baike.baidu.com/item/.NET/_blank)新开发的[平台](https://baike.baidu.com/item/%E6%8A%80%E6%9C%AF%E5%B9%B3%E5%8F%B0" \t "https://baike.baidu.com/item/.NET/_blank)，为商务便捷而构造的相互连通的应用系统，这些系统依赖于标准的，互联的，高稳定性的和高性能的。从技术方面来看，.NET程序是一个依赖于.NET Framework框架的[应用](https://baike.baidu.com/item/%E5%BA%94%E7%94%A8%E7%A8%8B%E5%BA%8F" \t "https://baike.baidu.com/item/.NET/_blank)程序。如果一个[应用程序](https://baike.baidu.com/item/%E5%BA%94%E7%94%A8%E7%A8%8B%E5%BA%8F" \t "https://baike.baidu.com/item/.NET/_blank)不依赖于.Net Framework框架，就不是.NET程序。具体的说，[XML](https://baike.baidu.com/item/XML" \t "https://baike.baidu.com/item/.NET/_blank)，[Web Services](https://baike.baidu.com/item/Web Services" \t "https://baike.baidu.com/item/.NET/_blank)，[SOA](https://baike.baidu.com/item/SOA" \t "https://baike.baidu.com/item/.NET/_blank)三者缺一不可。.NET框架是基于Windows的操作平台，使用在互联网开发上的平台分布式技术。

## 1.2研究意义

在科技日益发达的二十一世纪，社会对能源的需求越来越重，煤矿能源也变得越来越稀缺，人们对煤矿能源的开采变得越来越密集；各个矿井遍布在全国各地，但是随着人工矿业的发达，也带来了越来越多的难题；矿井下各种安全系数一直都是被高度重视的，而在各种安全系数中，煤矿下顶板离层数据一直是比较重要的一部分。到现在为止，各种矿井数据监测系统层出不穷。矿井底下数据监测系统一般都是通过传感器采集，各种总线实时传输，在地上系统实时接收数据查看。可是这样的方式成本较高，一般规模较小的煤矿公司根本没有必要花费这么大的成本去完成这种复杂庞大的实时检测系统，可是光靠人工去测量又太大材小用，而且人工测量不仅难度系数太高，而且危险系数也高。

本课题所研发的顶板离层数据采集系统是通过传感器检测数据，移动APP采集数据，上位机存储分析数据的方式完成对数据的采集、存储、分析的过程。也就是说，这个系统由传感器、APP、数据库、服务器、客户端软件组成，成本较低，在局域网内可以随时完成数据的上传与接收，传感器与移动APP之间是通过蓝牙传输数据，进一步降低了成本；在APP采集数据时，只需要打开手机的蓝牙，连接传感器的蓝牙模块，就会自动完成数据的传输，易于操作；更方便的，如果将服务端发布在服务器上，就能随时随地上传数据。

## 1.3论文主要工作

论文研究了比较成熟的技术和算法，如MD5加密，FTP传输协议，蓝牙传输，WCF框架等技术，并结合Android技术和.Net技术框架，开发了移动端软件和基于C#语言和C/S架构的矿井顶板离层数据检测系统。

本系统已开发实际项目，在与客户交互的过程中不断地修改和完善各种功能，不足的地方在客户的需求下也在慢慢改进，实现了大多数矿下数据监测必须要的功能。

目前已实现的功能有：移动APP与传感器硬件连接接收数据，将数据以规定格式保存成TXT文档，并可以随时查看，可以随时上传至FTP服务器，上位机能实现数据的接收，并且实现了绘制数据折线图和导出数据报表等功能。

## 1.4本论文的结构安排

论文总共分为六章，内容以及章节安排如下：

第1章，绪论。大致阐述了系统的开发背景和研究意义，以及本系统涉及到的基本技术与框架。

第2章，详细描述本系统所用的关键技术，以及简单介绍一下本系统所用的开发环境。

第3章，介绍系统开发目标、需求分析以及可行性分析。

第4章，介绍系统各个功能模块的设计，以及各个模块的详细实现过程。

第5章，使用设计好的测试用例测试本系统各个功能模块并生成报告。

第6章，总结与后续的修改。

# 2 开发环境以及关键技术

顶板离层数据检测系统是由移动端APP和上位机软件组成。本章主要介绍APP开发工具Android Studio以及数据采集APP中用到的技术，和上位机软件开发工具Visual Studio以及C/S上位机软件中用到的关键技术。

## 2.1 Android Studio概述

Google 公司为了方便程序员开发Android，推出了一款适用于Android编程的集成开发工具——Android Studio，在[IntelliJ IDEA](https://baike.baidu.com/item/IntelliJ IDEA" \t "https://baike.baidu.com/item/Android%20Studio/_blank)上做了改进 ，有点相似于[Eclipse](https://baike.baidu.com/item/Eclipse/61703" \t "https://baike.baidu.com/item/Android%20Studio/_blank) [ADT](https://baike.baidu.com/item/ADT" \t "https://baike.baidu.com/item/Android%20Studio/_blank)，Android Studio 不仅提供了集成的 Android 开发环境，而且比较方便调试。

在IDEA的基础上，Android Studio做了如下改进：

·提供了对[Gradle](https://baike.baidu.com/item/Gradle" \t "https://baike.baidu.com/item/Android%20Studio/_blank)工具的构建与方法的支持

·增加了Android 代码的自动生成与快速提示

·提示工具的捕获性能、可扩展性、兼容性等问题都得到了改进与提高

·支持使用[ProGuard](https://baike.baidu.com/item/ProGuard" \t "https://baike.baidu.com/item/Android%20Studio/_blank)优化字节码文件以及应用签名

·根据系统提供的向导来规范使用常用的 Android 应用设计和组件

·强大的布局设计编辑器，可以随意拖拉Android组件并随时预览效果

谷歌在2015年5月29日的I/O开发者大会上，发布了AndroidStudio 1.3版，这一版增加了基于C++的编辑和调试功能。Android Studio 1.3版本开发Android程序变得更方便，编辑代码的速度提升，而且支持C++本地语言的编辑和调试使用。程序员在设计布局界面的时候还可以看到自己的应用前端界面在不同尺寸屏幕中的样子，比较方便于Android开发。

## 2.2 Visual Studio概述

Microsoft Visual Studio(VS)是微软公司推出的开发工具包系列产品。Visual Studio IDE中几乎包含了各种编程语言的编写环境，在本系统中使用C#语言作为上位机软件的开发语言，C#语言与Visual Studio的兼容性特别高。不仅如此，Visual Studio 中创建的项目可以自带各种框架，本系统所用的WCF框架就在其中。Visual Studio中整合了各种创建可视化界面所需用到的控件，Visual Studio将各个控件封装成一个个类，所有事件的框架代码都是自动生成的，用户只用在指定的事件方法中编写自己的逻辑代码就行了。

## 2.3 开发语言简介

### 2.3.1 java和Android简介

Java是近20年来最受程序员喜欢的编程语言之一。java语言之所以广受欢迎，是因为其特有的优势：编写结构简单、独立型好、能够推动面向对象的发展、多线程、安全性高[3]。

Java是一门基于面向对象语言思想的高级编程语言，入门简单易懂，没有C/C++中晦涩难懂的指针概念，与此同时也没有C++中的多继承思想，这使得java建构比较简单，没有C++那么复杂；对比同是具有面向对象思想的C#而言，没有委托概念，但java中强大的多线程机制弥补了这一缺陷。Java文件编译之后得到的字节码文件可以在任意有JVM（java虚拟机，java运行环境）的系统上运行，这样使得java有了跨平台的优势。JVM可以进行各种调优，使得java运行环境更加优化。目前java版本已经更新到java12，本系统使用的是java8（java8比较稳定）。Java中的annotation（注解）这个概念是java5之后提出来的，Annotation定义了一种标准的描述元数据的方式，元数据就是描述数据的数据。

Android应用程序使用JAVA语言写的，在许多方面利用了java的优势，多线程机制在Android开发中发挥出了相当大的作用，各种主线程（UI线程）不能完成的任务都可以分配给子线程来完成，比如计时功能。

Android系统分为四层，从上往下依次是Application（应用层）、framework（框架层）、libraries/runtime（类库/运行层）、Linux Kernel（Linux内核层），如下图2.1所示：

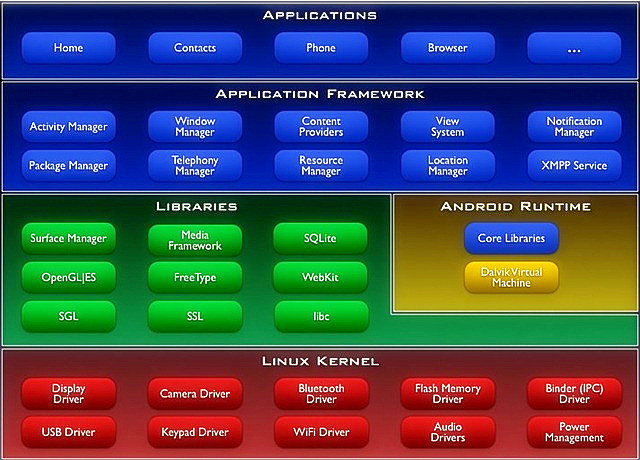


图2.1 Android系统底层框架

Application：应用层主要是包括上层应用程序，一般的APP都是安装在应用层。

Framework：框架层是一个应用程序的框架，提供各种Android四大组件——Activity（活动，代表应用程序的一个页面）、Service（服务，一直在后台运行，不需要提供界面交互）、Broadcast Receiver（广播接收器，用于接收各种广播，如收到短信的广播、开机广播）、Content Provider（内容提供者，用于应用程序之间交换数据）。这一层也为开发者提供各类API。

Libraries/Android RunTime：本层分为两个部分，一部分是各种库，还有一部分是Android 运行环境。

Linux内核层：包括各种硬件驱动，是硬件和软件之间的交互层。

### 2.3.2 C#简介

C#与java特别相似，也是一种面向对象的编程语言，是用来编写运行在微软平台上的.Net Framework 平台上的应用程序的一种高级编程语言。

与java相比，C#少了字节码文件这一特点，意味着C#没有跨平台的优势。然而，微软为C#提供各种方便开发的控件，这意味着C#开发UI变得非常方便，事实上，我们只要拖动控件，在Visual Studio提供的自动生成的事件代码中加入自己的逻辑，就能实现想要的功能，C#开发使得程序员不用重复写无用代码，只要写关键代码就够。

## 2.4 C/S结构叙述

C/S架构，即Client/Server (客户端/服务端) 架构，是开发人员都比较熟悉的软件体系架构，通过客户端和服务端的配合，提高了软件系统的通讯效率，客户端程序需要安装才能进行各种操作。

用户可以操作的程序全部都在客户端，服务端提供的服务主要有数据存储、数据管理、数据维护和并发控制等，客户端的任务是通过访问服务端获取数据从而完成具体的逻辑业务。

C/S架构的优点是开发比较容易，用户交互界面简便友好；缺点是应用程序的扩展和维护比较困难，如图2.1所示是最基本的C/S两层架构。

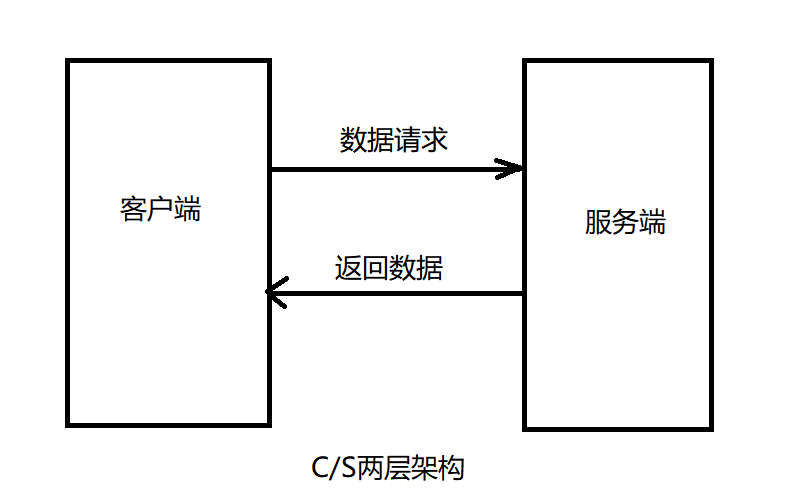


图2.2C/S两层架构

## 2.5 WCF详解

### 2.5.1 WCF简介

WCF的全称是Windows通信基础（Windows Communication Foundation）。WCF是微软推出的，基于SOA的.Net平台下的框架产品，SOA全称是Service Oriented Architecture，即面向服务的体系结构，基于“软件变服务”的思想[2]。在WCF框架下开发SOA非常方便，它可以创建和运行分布式系统，是一种强大的编程模型，这种模型是面向服务的。

WCF服务端通过暴露终端节点（图2.2中的EndPoint），将客户端需要的Message传给客户端终端节点，实现客户端业务逻辑。WCF基本框架如图2.2：

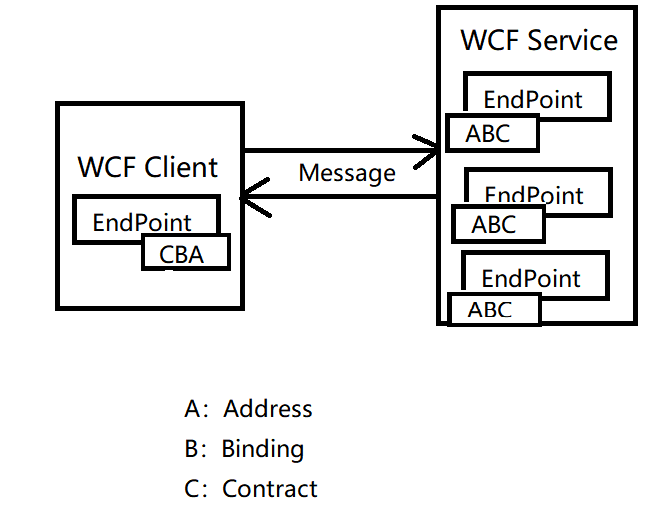


图2.3 WCF基本框架图

### 2.5.2 WCF框架

WCF是一个分层的体系架构，它的结构从上往下依次是：

（1）应用层（Application）：是开发应用程序的一层，与用户交互的程序就是在这一层开发的，应用层是用户能够接触的唯一一层。

（2）契约层（Contracts）：这一层包含各种契约，指定客户端以什么样的服务和信息访问服务端，服务端按照约定返回指定格式的信息。契约层基本包含以下四种类型的约定：

Service contract - 服务契约，是用来描述服务操作的。是一系列操作的集合，每一个操作都描述了一种消息模式，服务契约用一种特殊的渠道提供对于服务的描述，把基于.NET平台的CLR类型变成不基于平台的描述。

Data contract - 数据契约，定义与服务交换的数据格式。客户端和服务端需要约定协议方便交互。

Message Contract - 消息契约，用来控制数据契约。消息契约的主要目的是制定SOAP消息的参数类型格式（在这里，WCF采用通信格式是的SOAP格式，SOAP代表简单对象访问协议）。

Policy and Binding - 策略和约束契约，也可以说是一个服务，客户端和服务端通信的先决条件，通信时需要遵循这一约定。

（3）运行时服务（Service Runtime）：在契约层下面的一层，决定在运行时的各类服务型操作。可以操作以下类型的服务：

Throttling Behavior - 统计处理消息的数量

Error Behavior – 处理发生内部服务错误时

Metadata Behavior – 指定元数据暴露的可能性

Instance Behavior – 统计客户端需要创建需要的实例的数量

Transaction Behavior – 在事务状态变化的情况下出现的任何故障

Dispatch Behavior - 调度行为，WCF处理消息的基础方式

Concurrency Behavior - 服务器在通信过程中使用该功能控制客户端

Parameter Filtering - 服务参数在被调用之前的验证的过程

（4）消息层（Messaging）：这一层主要由消息通道构成，通道的功能是在两个端点之间传递消息内容，一组通道是由通道堆栈组成的，通道堆栈包括以下两种类型：

Transport Channels - 传输通道的位置在通道堆栈底部，负责发送和接收各种协议的消息，如（HTTP、TCP和P2P）。

Protocol Channels - 协议通道位于通道堆栈的顶部，也被称为层状通道，按照各种通信协议规定的格式封装消息。

（5）寄宿和托管

WCF框架的最后一层是托管，托管其实就是将服务端的程序寄宿到某个地方以便服务端的程序正常运行，方便于客户端交互，托管的方式有以下几种：

IIS - Internet Information Service的缩写，使用IIS寄宿WCF服务，这里的原理其实和Web Service一样，会形成一个URL链接，客户端通过这个URL访问服务端获取各种服务。

Windows激活服务 - 通常缩写为WAS，是IIS7.0开始提供的，是使用TCP或Namedpipe协议来进行HTTP和非HTTP通信。

Self-Hosting - 这是WCF服务获取自托管控制台应用程序的机制，为自己的解决方案提供了非常灵活的协议和设置选择。其实是将服务端程序打包成一个应用程序，在服务器上安装这个应用程序，需要程序保证一直运行。

Windows Service - 这个寄宿WCF服务的方法是比较好的，因为服务一直保持运行状态并与可以随时客户端联系，不像其他的寄宿方法一样需要客户端激活才能提供服务。

上述WCF分层的体系架构如图2.3所示：

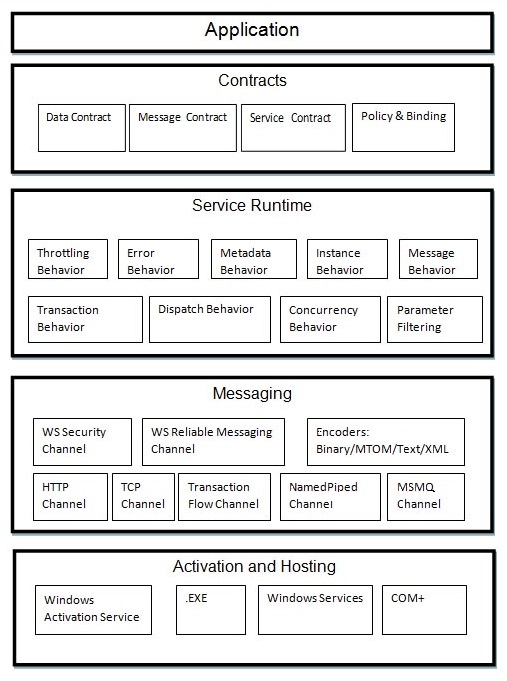


图2.4 WCF分层体系框架图

## 2.6 FTP简介与工作原理

### 2.6.1 FTP简介

FTP是文件传输协议（File Transfer Protocol）的缩写，是一个应用层的协议，专门用于在[客户端和服务器](https://zh.wikipedia.org/wiki/%E4%B8%BB%E5%BE%9E%E5%BC%8F%E6%9E%B6%E6%A7%8B" \o "主从式架构)之间传输文件。文件传送（file transfer）和文件访问（file access）之间的区别在于：前者由FTP提供，后者由如[NFS](https://zh.wikipedia.org/wiki/%E7%BD%91%E7%BB%9C%E6%96%87%E4%BB%B6%E7%B3%BB%E7%BB%9F" \o "网络文件系统)等应用系统提供[4]。FTP是一个客户端-服务器协议，能在不进行任何处理的情况下传输各种类型的文件，就意味着FTP非常安全，基本不会发生文件损坏或者丢失的情况。

不好的是，FTP的延时特别高，这就造成了，从发送请求到接收到文件之间中间的这段时间非常长；而且需要执行一些冗长的登录程序，加剧了FTP的延时。不过本系统的数据量不是很大，完全能够忽视这些延时。

FTP服务分别用20和21端口来传输数据流和控制流，21是通向ftp服务器的进口，当控制流传输时，数据流端口是空闲的，而当数据流传输时，控制流端口处于空闲。

### 2.6.2 FTP工作模式

（1）主动模式

在这种模式下，客户端从一个大于1023的端口连接到服务器的21端口（这里是为了告诉服务端自己的端口——这里客户端连接的端口不是传输端口，只是一个临时端口），FTP服务器再将自己的20端口连接到客户端的指定端口（之前客户端告诉服务端的端口号），开始传输数据，过程如图2.4所示：

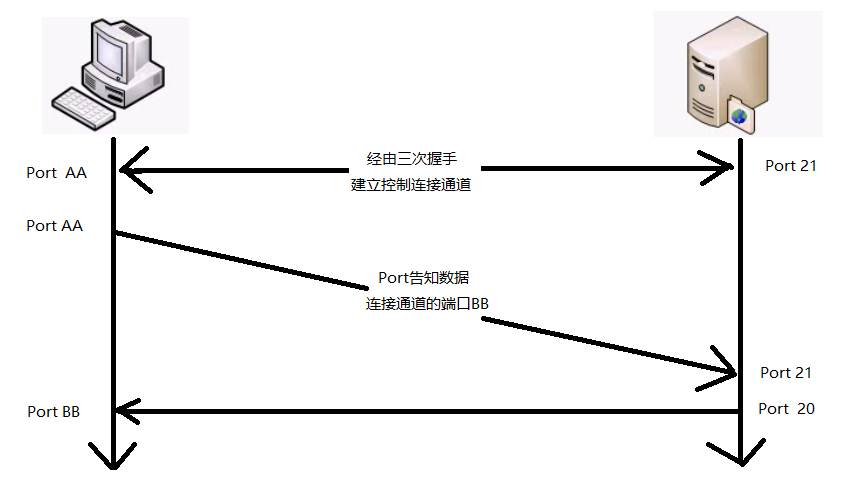


图2.5 FTP主动模式传连接过程

1. 被动模式

客户端从一个大于1023的端口连接到服务器的21端口，告诉服务端自己将会使用某个端口连接服务端，然后服务端处于等待连接状态，等待客户端向服务端的某个端口（被动模式的数据端口不是20，这里的端口号是大于1023的随意一个端口）发起连接，连接完成之后开始传输数据，如图2.5所示：

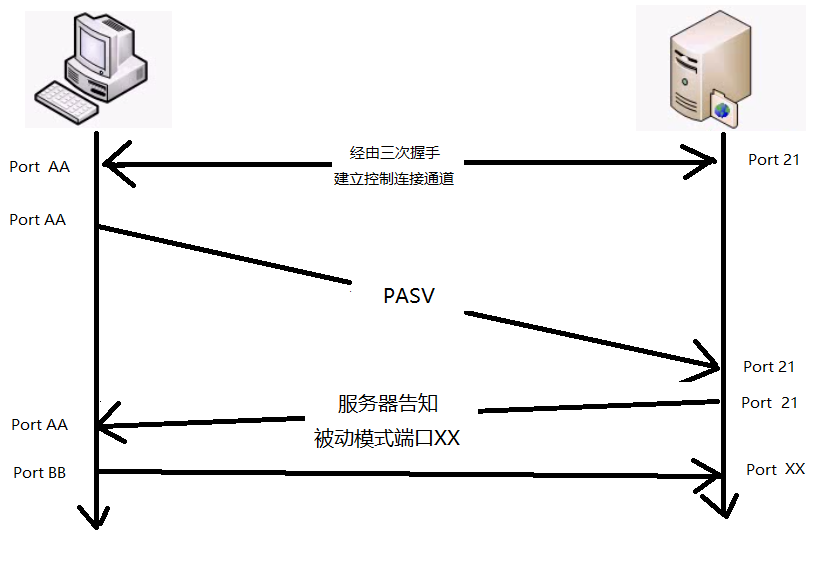


图2.6 FTP被动模式连接过程

### 2.6.3 FTP传输方式

（1）ASCII传输方式

当客户端和服务端之间传输的是简单的文本文件时，可以使用ASCII码传输。这是，当接收的一端机器使用的系统与发送文件的操作系统不一致，比如发送文件的系统是Windows，而接收的系统是Linux，此时就会自动将文本内容调整成Linux存储的文本格式。

（2）二进制传输方式

当客户端与服务端之间传输的是二进制文件（如.exe文件，程序或者压缩文件等连续文件）时，二进制传输方式可以对文件进行连续拷贝，而不会对传输的文件进行任何处理，就能保证文件的完整性。

## 2.7 SQL Server数据库

SQL全称[结构化查询语言](https://baike.baidu.com/item/%E7%BB%93%E6%9E%84%E5%8C%96%E6%9F%A5%E8%AF%A2%E8%AF%AD%E8%A8%80/10450182" \t "https://baike.baidu.com/item/SqlServer/_blank)（Structured Query Language）。SQL语言被ANSI([美国国家标准协会](https://baike.baidu.com/item/%E7%BE%8E%E5%9B%BD%E5%9B%BD%E5%AE%B6%E6%A0%87%E5%87%86%E5%8D%8F%E4%BC%9A" \t "https://baike.baidu.com/item/SqlServer/_blank))规定为所有[关系型数据库系统](https://baike.baidu.com/item/%E5%85%B3%E7%B3%BB%E5%9E%8B%E6%95%B0%E6%8D%AE%E5%BA%93%E7%AE%A1%E7%90%86%E7%B3%BB%E7%BB%9F/696511" \t "https://baike.baidu.com/item/SqlServer/_blank)的标准语言，SQL语句最主要的功能是操作各种数据库，得到想要的数据。

SQL Server数据库引擎是一款[关系数据库管理系统](https://baike.baidu.com/item/%E5%85%B3%E7%B3%BB%E6%95%B0%E6%8D%AE%E5%BA%93%E7%AE%A1%E7%90%86%E7%B3%BB%E7%BB%9F/11032386" \t "https://baike.baidu.com/item/SqlServer/_blank)（RDBMS），一开始是由微软、Sybase和Ashton-Tate三家公司共同开发的，并在1988年推出了O/2版本，今年以来不断推出新版本，现在的最新版本是在2017年发布的SQL Server 2017。

以SQL Server 2008为例，介绍SQL Server各个版本:

SQL Server 2008企业版:

这一版本将为开发者提供更大规模在线逻辑处理模型和更强大的服务器引擎，是一个全方位的应用处理和数据智能存储的平台，为开发者开发应用程序提供了各方面的支持包括数据仓库、高扩展性、高安全性、数据报表和智能分析等。

SQL Server 2008标准版:

是一个应用处理和数据智能存储的平台，为企业部门级应用程序开发提供了最大限度的友好性和可靠的管理特性。

SQL Server 2008工作组版:

这一版很容易就能升级到标准版或企业版,因为具备管理核心数据库的引擎特性，这是一个非常可靠的应用处理和数据智能存储的平台，拥有以实现既可靠又安全的发布、远程同步运行和对分布式运行程序为目的的管理能力。

SQL Server 2008 Web版:

这一版提供了可以实现支持实用性强、规模较大、成本不高的web程序或为业务托管解决方案的工具，是为了实现Windows服务器中的需求而设计了高实用性、面向网络服务的运行环境。

SQL Server 2008开发者版:

这一版本也可以轻易地升级到企业版。开发者使用SQL Server 2008开发者版可以开发任何关于SQLserver的应用程序。这一版具备所有企业版的功能，但是只能用来开发和测试，不能给基于任何产品的发布。

SQL Server 2008 Express版:

尽管这一版只是SQL Server中的一个小版本，但这是SQL Server所有版本中唯二的免费版本，这一版本和工作组版本一样具有所有核心的管理数据库的功能，而且具有SQL Server 2008中最新的DateTime数据类型。这一版只是为了学习或开发小型应用程序设计的。

SQL Server Compact 3.5版:

SQL Server Compact版本可以运行在各个微软平台上。这是SQLserver另一个免费版本，这一版适用于嵌入式应用程序，可供移动应用、小型桌面应用、小型web程序使用。

## 2.8 本章小结

本章首先介绍了APP和上位机软件的开发环境，以及简单介绍开发语言；然后是关键技术的概述，包括C/S架构基础、WCF详解，以及FTP简介和工作原理和SQLServer数据库的介绍。

# 3 需求分析

需求分析是软件生命周期中特别重要的一个环节，这一部分是在不考虑怎么去实现系统功能的前提下，分析系统需要什么样的功能。需求分析是根据用户提出的需求进行分析，确认要开发的软件需要实现的功能有哪些。

## 3.1 系统开发目标

煤矿井下顶板离层数据检测系统是为在矿井工作人员提供的一款检测系统，

需要实现的功能是数据采集、数据上传、数据接收、数据分析、数据存储等功能。对煤矿工作人员的技术要求是：熟悉计算机基础操作。

## 3.2 系统需求分析

本系统主要包括顶板离层数据采集APP和顶板离层数据分析系统软件两部分。

APP主要包括以下功能：采集处理、数据浏览、上传数据、参数设置。

采集处理：与传感器硬件的蓝牙连接，自动接收传感器传输过来的数据，将数据按照约定的格式封装成TXT文件，保存在手机存储空间中，同时将接收过来的数据进行展示，使用ListView控件，将数据列出来。数据接收完之后蓝牙自动断开（为了解决硬件的功耗问题）。

数据浏览：数据保存在手机内存中，可以随时进行浏览，浏览数据前选择日期和设备ID，数据浏览与上面数据采集时浏览数据的方式是一样的。

参数设置：与传感器硬件的蓝牙模块连接，硬件自动发送此时的各个参数（设备ID、设备时间、采集周期、xx值），APP接收之后，用户根据此时的参数对传感器参数进行修改，修改完成之后蓝牙连接自动断开。

上传数据：这一部分功能在第一次使用的时候需要用户手动输入FTP IP、用户名和密码,保存完成之后就可以进行数据上传。另外上传完成之后为了保证用户误触上传按钮再次上传造成上传失败，上传完成之后将按钮禁用，同时显示提示数据已上传，请勿重复上传。还有上传完所有数据之后，用户可以选择将所有数据清空，以防止下次重复上传。

APP目前具备上述所有功能，后期可能会因为用户需求进行扩展和更改。

上位机软件包括用户登录、数据接收、数据查询、历史曲线、参数设置、创建报表和备份还原数据库等功能。

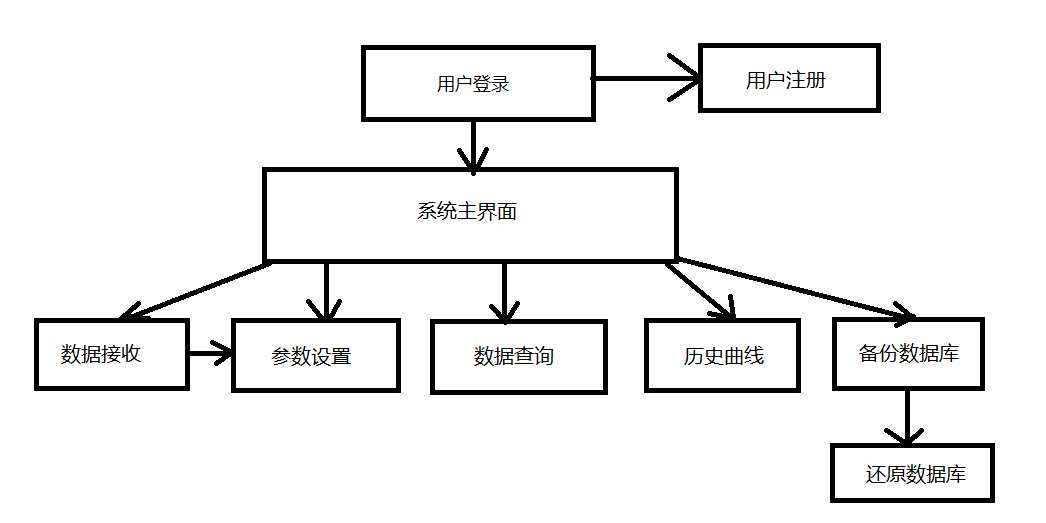


图3.1顶板离层数据检测系统功能图

用户登录：这一模块只需用户输入对应的用户名和密码，用户名和密码正确，登录成功，进入主界面，否则显示登录失败，并提示错误信息。

用户注册：用户需要输入用户名和设置对应的密码，点击注册，系统检查用户名的合法性，用户名合法，显示注册成功，否则显示注册失败，并提示错误信息。

登陆成功之后进入系统主界面，主界面有数据接收、数据查询、历史曲线、参数设置、创建报表和备份还原数据库等的选项：

选择数据接收，程序开始从FTP服务器下载数据并自动解析数据存入数据库，并提示下载进度。

参数设置：用户需要手动设置输入设备所属的矿井名、巷道名、位置以及深基和浅基的报警值。

数据查询：用户选择开始和结束日期和设备ID，将从开始到结束这一段时间此设备所有的数据都列出来。

历史曲线：用户选择开始和结束日期和设备ID，将从开始到结束这一段时间所有的数据绘制成折线图展示给用户看，并加上之前所设置的报警值作为参考。

创建报表：自动生成报表包括日数据报表和月数据报表。日报表将这一整日的数据绘制成一张简单的小型折线图，并在折线图下方加上一个表格列出这一日每个小时的深基浅基平均值、最大值、最小值、最大最小值之间的差值；月报表列出这一整月每天的数据平均值、最大值、最小值以及最大最小值之间的差值。并计算出本月和上月数据平均值、最大值、最小值做成对比。最后应客户要求加上一个实时报表，显示某一条巷道所有的设备最后一条数据，模拟实时数据报表。并在日报表的基本功能上添加一个修改报表的功能，用户修改某个时间段的平均值、最大值、最小值，根据用户输入的数据在折线图和表格上进行修改数据。

备份还原数据库：这个功能与用户无关，是开发者为了保证数据安全，所必须实现的功能。而在每次数据接收完成之后就会自动备份一次数据库。

## 3.3 可行性分析

系统可行性分析是需求分析中比较重要的部分，一定程度上决定软件是否能够实现。本系统的可行性分析分为三部分：经济可行性、技术可行性和操作可行性。

（1）经济可行性

本系统采用的是APP+C/S软件的形式，APP开发成本几近于零；C/S软件需要一个服务器，由于是给小型煤矿企业开发的，数据量并不多，可以把服务端部署在局域网内，客户端只要在同一网段之内的设备都能运行，还可以将服务器部署在云端，这样就能实现随时随地上传下载数据的功能，从经济角度看，本系统可行。

（2）技术可行性

APP开发工具：Android studio，所用语言java，关键技术：广播机制、套接字编程、流、页面布局等。

APP运行只需要在Android4.0以上的有蓝牙功能手机上都能运。

上位机开发工具：visual studio、SQLServer数据库，所用语言C#，关键技术：WCF架构、连接SQLserver数据库操作数据、visual studio插件report viewer的使用等。

上位机部署环境：Windows Server2012R2、Windows Server 2008和Windows 10、Windows 7。

从技术层面上看，本系统可行。

（3）操作可行性

本系统是针对煤矿工作人员设计的软件，一切功能站在用户角度出发，并且是桌面程序，功能齐全，简单易懂，界面简洁，对用户非常友好，操作可行。

## 3.4 本章小结

本章一开始主要确定了系统所适用的用户人群，然后详细总结了系统需要实现的具体功能，接着是可行性分析，从经济、技术、用户操作三个方面进行具体可行性分析研究。

# 4 设计与实现

## 4.1 系统总体设计

### 4.1.1 总体功能划分

矿井顶板离层数据检测系统是通过数据采集APP，将各个矿井的数据以文本文档的形式通过网络上传至FTP服务器，将FTP服务器作为数据转换的中介，将各个矿井的数据信息(传感器数据)存储到总服务器的数据库中，当客户端访问时,从服务器获取数据。

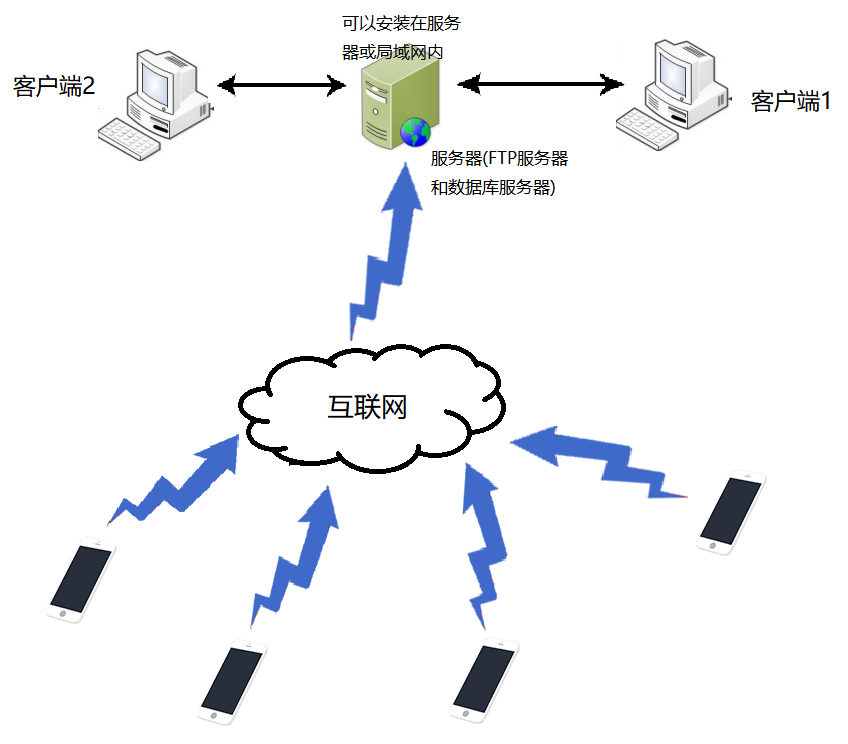


图4.1顶板离层数据检测系统拓扑图

### 4.1.2 系统模块设计

按照系统需求分析，将整个系统划分为数据接收、系统设置、数据库维护、数据查询、系统设置和用户管理六个模块，各个模块的功能如下图4.2所示：

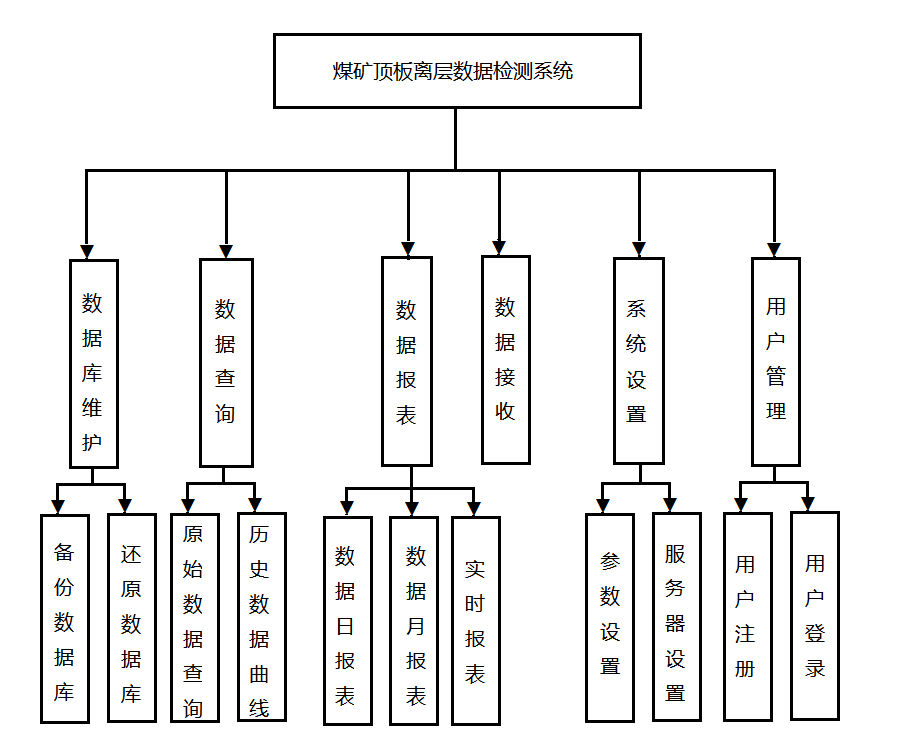


图4.2顶板离层数据检测系统功能模块图

数据采集APP的分为数据采集、数据浏览、参数设置、数据上传四大模块，各模块具体功能如图4.3所示：

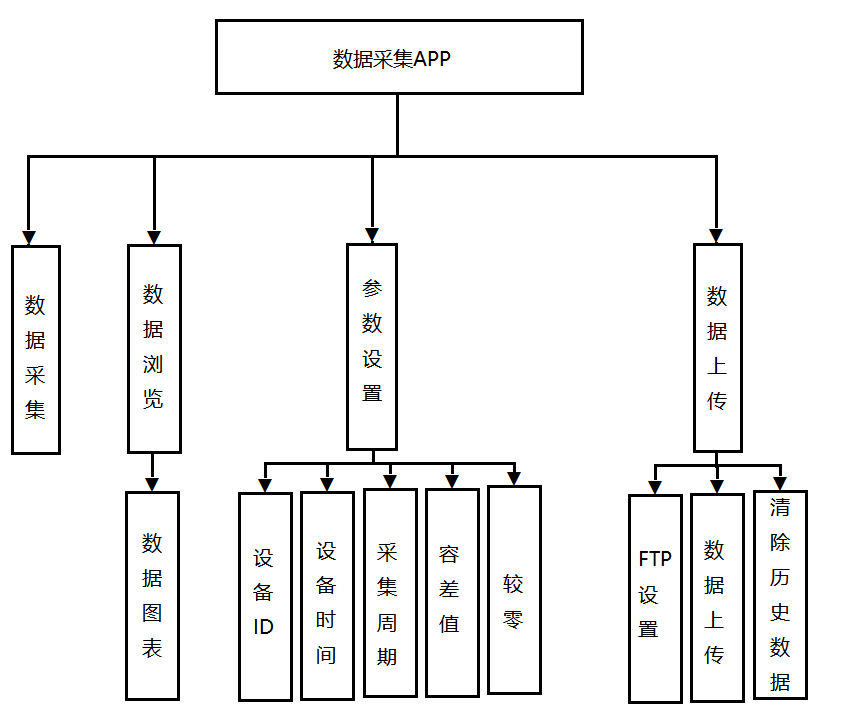


图4.2数据采集APP功能模块图

## 4.2 系统数据库设计

本系统采用的是SQL Server数据库，实现的是对矿下顶板离层数据的操作，所以需要一类表是用来存储数据的，这类表以设备ID作为表名，总共有四列，分别是主键ID、时间日期、深基、浅基；传感器设备信息也需要存储，比如所在矿井名、巷道名、位置；本系统还有用户管理功能，所以还有一张用户信息表，包括主键ID，用户名，密码。

### 4.2.1 数据库概念设计

根据顶板离层数据检测系统的数据库需求，设计出以下实体-关系模型图（E-R图）：

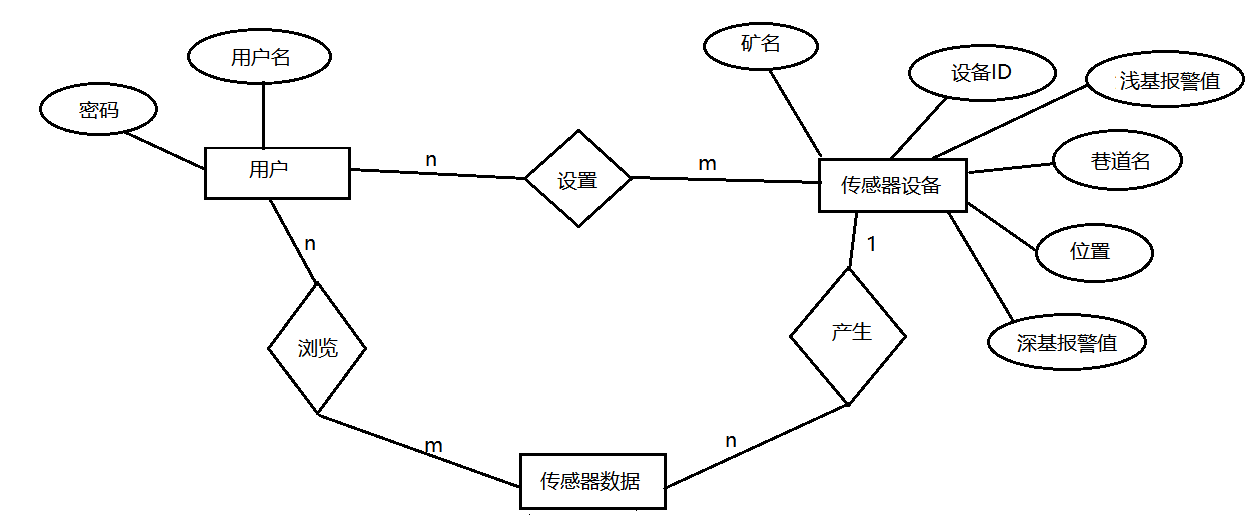


图4.3 系统E-R图

### 4.2.2 数据库逻辑设计

## 4.5 本章小结

本章一开始描述了系统总体设计，包括系统总体功能的划分和系统功能模块的设计，然后实现了数据库的设计，并详细地描述了数据采集APP和顶板离层数据监测系统的各个功能的具体实现。

# 5 软件测试

软件测试属于软件设计过程中的最后一步，其目的是测试软件的功能是否符合预期的效果、界面是否与预期的符合、软件有无错误。软件只有通过了最后的测试才能上线。

软件测试主要包括4个层次，如此下图所示

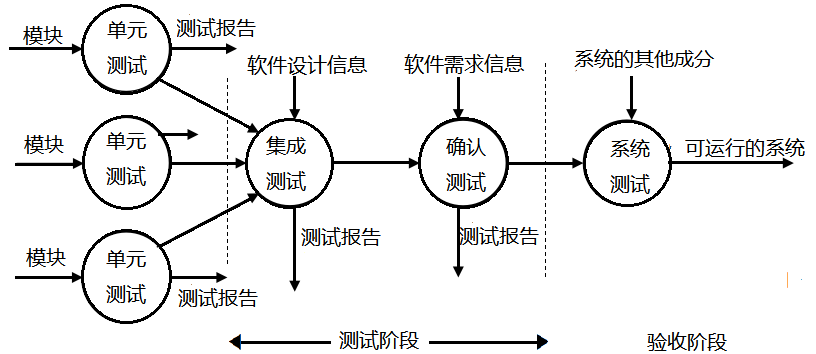


图5-1层次测试信息流程图

## 5.1 功能测试

功能测试也叫黑盒测试，指的是站在用户的角度测试软件，开发人员在忽视软件内部结构和代码的情况下对软件的功能进行测试。对软件的进行一个整体评估，再对各个功能进行测试，将预期的结果与实现效果进行对比，进行改进，使软件达到预期的功能。再撰写测试报告，完成一整个测试的过程。

功能测试需要测试软件的功能有没有错误，是否达到预期的效果，功能测试表如下：

表5-1功能测试表

|  |  |
| --- | --- |
| 测试目标 | 软件性能是否达标，系统是否没有出现异常。  软件是否与需求分析的一致，各功能模块是否正确。 |
| 测试范围 | 需求分析中的各个功能模块。 |
| 技术 | 使用黑盒测试的技术按照各个测试用例的标准来进行测试。 |
| 开始标准 | 点击系统各个功能页面，正常进行使用 |
| 完成标准 | 测试结果与预期的效果一致 |
| 测试重点 | 数据库数据和软件的正确性 |
| 需要考虑的特殊事项 | 软件运行时是否出现异常停止运行的情况 |

## 5.2 测试用例及测试结果

（1）“数据采集”测试结果

表5-2“数据采集”测试结果

|  |  |  |
| --- | --- | --- |
| 测试要点 | 测试路径 | 测试结果 |
| 蓝牙搜素 | 传感器蓝牙模块蓝牙名为“CCCC” | 成功搜索到“CCCC” |
| 蓝牙连接 | 连接蓝牙名为“CCCC”的传感器 | 提示“连接成功” |
| 数据传输 | “CCCC”设备在2019年2月24号采集到199条数据 | 准确采集到199条数据 |

（2）“数据浏览”测试结果

表5-3“数据浏览”测试结果

|  |  |  |
| --- | --- | --- |
| 测试要点 | 测试路径 | 测试结果 |
| 日期选择 | 手机中存储了2019年3月1日到3月8日的数据 | 自动补齐输入框中有190301-190308 |
| 设备选择 | 手机中2019年3月3日有“34A1”和“34A3” | 选择190303，下拉列表框中出现“34A1”和“34A3” |
| 数据 | 选择“34A3” | 列出“34A3”设备2019年3月3日的所有数据 |
| 折线图 | “34A3”的2019年3月3日的数据 | 成功绘制出“34A3”的2019年3月3日的数据折线图 |

（3）“数据上传”测试结果

表5-4“数据上传”测试结果

|  |  |  |
| --- | --- | --- |
| 测试要点 | 测试路径 | 测试结果 |
| FTP设置 | 输入FTP的IP、用户名、密码 | 提示“设置成功” |
| 数据上传 | 将手机中的数据都上传到FTP服务器中 | 提示“上传成功” |
| 删除历史数据 | 把手机中的所有数据全部删除 | 提示“成功删除全部数据” |

（4）“参数设置”测试结果

表5-5“数据上传”测试结果

|  |  |  |
| --- | --- | --- |
| 测试要点 | 测试路径 | 测试结果 |
| 设置设备ID、设备时间、采集周期、容差值、校准 | 设备ID设置成“ABCD”、设备时间设置成此刻时间、采集周期设为3秒、容差值设为2 | 提示“设置成功” |
| 清零 | 点击“清零”按钮 | 提示“清零成功” |

（5）“用户登录”测试结果

表5-6“用户登录”测试结果

|  |  |  |
| --- | --- | --- |
| 测试要点 | 测试路径 | 测试结果 |
| 用户名、密码 | 用户名或密码为空 | 输入框后显示“请输入用户名”或“请输入密码” |
| 用户名、密码 | 用户名为真、密码为假；输入：用户名“123”； 密码“12” | 提示“密码错误” |
| 用户名、密码 | 用户名为真、密码为真；输入：用户名“123”；密码“123” | 登录成功，进入系统主界面 |

（6）“用户注册”测试结果

表5-7“用户注册”测试结果

|  |  |  |
| --- | --- | --- |
| 测试要点 | 测试数据 | 测试结果 |
| 用户名、密码等 | 已存在的用户名；输入已经注册过的用户名“123” | 提示“用户名已存在” |
| 密码一致性验证 | 密码输入：hhh  确认密码输入：aaa | 提示“两次密码不一致,请重新输入” |
| 用户名、密码 | 不存在的用户名“黄珍”  密码：hhh  确认密码：hhh | 提示“注册成功” |

（7）“数据接收”测试结果

表5-8“数据接收”测试结果

|  |  |  |
| --- | --- | --- |
| 测试要点 | 测试数据 | 测试结果 |
| 数据接收 | FTP中有190301-190308的数据 | 数据库中新增加了所有数据 |

1. “数据查询”测试结果

表5-9“数据查询”测试结果

|  |  |  |
| --- | --- | --- |
| 测试要点 | 测试数据 | 测试结果 |
| 原始数据查询 | 查询2019年3月3日到3月8日设备34A3设备的所有数据 | 成功显示2019年3月3日到3月8日设备34A3设备的所有数据 |
| 历史数据曲线 | 查询2019年3月3日到3月8日设备34A3设备的曲线 | 成功2019年3月3日到3月8日设备34A3设备的数据折线 |

（9）“系统设置”测试结果

表5-10“参数设置”测试结果

|  |  |  |
| --- | --- | --- |
| 测试要点 | 测试路径 | 测试结果 |
| 参数设置 | 设置34A3设备为“成庄矿”、“50124巷道”、“200米”、深基报警30、浅基报警50 | 设置成功，下方的参数设置明细中多出34A3的设备信息 |

（10）“数据报表”测试结果

表5-11“参数设置”测试结果

|  |  |  |
| --- | --- | --- |
| 测试要点 | 测试数据 | 测试结果 |
| 日报表 | 查看34A3设备2019年3月3日的日报表 | 正确显示2019年3月3日的日报表 |
| 修改日报表 | 修改34A3设备2019年3月3日报表的00:00:00-01:00:00  的数据为，平均值：深50浅100；最大值：深50浅100；  最小值：深50浅50 | 成功修改00:00:00-01:00:00  的数据 |
| 月报表 | 查询设备F101的2019年2月的月报表 | 正确显示F101的2019年2月的月报表 |
| 实时报表 | 查询成庄矿“50124巷道”的高于“深20浅30”的设备 | 成功显示成庄矿“50124巷道”的高于“深20浅30”的设备 |

（11）“数据库维护”测试结果

表5-11“数据库维护”测试结果

|  |  |  |
| --- | --- | --- |
| 测试要点 | 测试路径 | 测试结果 |
| 数据库备份 | 数据库备份 | 提示“备份数据库成功” |
| 数据库还原 | 数据库还原 | 提示“还原数据库成功” |

## 5.3 本章小结

本章主要对数据采集APP和顶板离层数据分析系统上位机软件进行了各个功能的测试，通过测试发现APP和软件都达到了预期的结果。

# 6 总结与展望

## 6.1总结

在整个毕业设计的过程中，我学到了很多课堂上从来没有接触过的理论知识。同时，也使自己的动手能力有了很大的提升。通过对数据采集APP的开发，我巩固了java和Android方面的知识，并接触到了关于蓝牙扫描的知识，并熟练掌握了java的套接字编程与输入输出流的使用。并且，通过对顶板离层数据分析系统软件的开发，我熟悉了系统开发流程，对C#面向对象有了比较深刻的认知，特别是掌握了C#中多线程更新UI的知识。

本次毕业设计让我认识到软件开发是一件有趣的事，特别是在功能模块的测试的时候，发现问题之后解决问题是一件让人特别自豪的事情。同时，我也觉得软件开发是一个比较艰难的过程，在开发之前需要特别多的准备工作，像需求分析和系统功能模块的划分。

在本次毕业设计中我也遇到一些困难，在各个功能实现的过程中都多多少少遇到一些问题，有些问题通过自己调试很快就解决了，而有些问题需要上网查找资料才能进行参考着解决，比较困扰的问题就需要请教导师了。在解决问题的过程中，不仅是我的耐心得到了很大的提升，同时也锻炼了自己独立解决问题的能力，并且充分认识到了我所学的知识是很匮乏的、很片面的。在以后的工作中，我会努力提高自己的动手能力，同时与理论知识相结合。

## 6.2 展望未来

在老师的知指导下，系统初步形成了。当可是因为时间比较短与技术能力有限，系统比较粗糙并且存在很多不足，在很多方面还有比较大的改进空间。在代码方面，很多代码都是比较多余的，不够简洁明了，导致了系统的性能不是很完美。

代码重构是一件比较有挑战性的事情，需要有足够的耐心和充分的心里承受能力，但是代码重构一旦成功，系统的性能与结构都能有一个质的跳跃，一个优秀的程序员，是敢于并且可以重构自己以前的代码的。

# **致谢**

历时两个月的毕业设计终于结束了，我心里有很多的情绪杂糅在一起，有欣喜、欣慰、还有放松，还有一丝心酸。因为这两个月过完了之后就要毕业了，回想起这四年的大学时光，我不禁感叹时光飞逝，美好的时光总是过得飞快的。

我要感谢我的导师XXX，倪老师对待学生的态度是认真负责的，在整个毕业设计的过程中，XXX一直给我指导与帮助。在论文的撰写过程中，XXX不厌其烦地指出我的各种错误并不停地指导我进行改正。在论文的时间安排上，倪老师也给了我很大的指导意见，到后来我如期能完成毕业设计和毕业论文，离不开倪老师的帮助。

同时，我还要感谢我的同组同学，无论是毕业设计还是毕业论文过程中，我们一直互相鼓励，相互讨论，很多的错误都是我们一起探讨，一起改正的。在后面写论文的过程中，我们互相督促，这才有了我们如期完成毕业论文的结果。

# 参考文献

1. 马小军, 李伟, 杨爱军. 基于WCF的分布式机场安检设备远程管理研究与设计[C]. 中国计算机用户协会网络应用分会2018年第二十二届网络新技术与应用年会. 0.
2. 苏敏. 基于计算机软件开发的JAVA编程语言研究[J]. 数码世界, 2018.
3. 于磊. 基于WCF的数据库应用开发[J]. 电脑知识与技术, 2018, v.14(6Z):17-18.
4. 周政尹. 基于WCF的分布式缓存系统设计[J]. 电子技术与软件工程, 2018.
5. 胡鑫. 基于代理的FTP安全审计技术的研究与实现[D]. 2015.
6. 林伟婷. C/S与B/S架构技术比较分析[J]. 科技资讯, 2018, v.16；No.514(13):21-22.
7. DerekBeyer, Beyer, 龚小平. C＃ COM+编程指南[M]. 中国水利水电出版社, 2002.
8. 叶核亚.Java程序设计实用教程（第5版）