

摘要

二滩水电站设备状态检修系统利用二滩已建成的各类自动化系统作为数据源，辅助以人工录入数据、点检采集数据，将二滩电厂宝贵的数据财富形成统一的数据仓库，在大量数据中提取设备运行规律，对设备的当前工况进行识别，对未来的运行趋势进行预判，并以图形展示的可视化手段指导运行维护人员及时发现故障、处理故障，最终形成设备检修计划决策支持系统。

本系统主要是采用了 C/S 的设计结构，其开发主要包括后台的 MongoDB 数据库建立与维护。利用现在比较流行 Python 语言对客户提供的不同数据格式和类型文件进行简单方便的文件内容定时收集与统计计算，转成数据库可利用的数据，并且保证数据唯一性和完整性。

前端采用了 C#.net 和现在界面展示比较友好的 DevExpress 工具进行开发，人机交互友好、满足运行维护人员快速调取需要分析的对象和原始数据点，支持曲线、散点图、折线图外观自定义，通过各种图线可以直观的反映数据问题。

本论文记述了从需求分析入手，利用相关技术进行设计，实现系统目标，以及系统测试的情况。

关键词：MongoDb；Python；C#.net；格式化文件自动导入；图形界面展示

目录

1. 绪 论.....	1
2. 开发环境.....	2
2.1 硬件开发环境.....	2
2.2 软件开发环境.....	2
2.3 本章小结	3
3. 需求分析.....	5
4. 系统设计.....	6
4.1 系统功能设计.....	6
4.2 数据库设计.....	6
5. 系统实现.....	11
5.1 用户登陆界面实现.....	11
5.2 主界面实现.....	11
5.3 用户管理界面实现.....	11
5.4 缺陷信息管理界面实现	12
5.5 图形表示界面实现.....	12
5.6 数据导入后台实现.....	14
5.7 本章小结	14
6. 系统测试.....	15
6.1 登录测试	15
6.2 用户管理界面动作测试	15
6.3 反馈界面动作测试.....	16
6.4 数据点的图形展示测试	16
6.5 模拟量数据导入测试	17
6.6 本章小结	18
结 论.....	19
致谢.....	20
参考文献.....	21

1. 绪 论

近年来二滩水电站已建成的各类自动化系统。由于各系统设计目的不同，以至数据分散，各自为营，彼此没有什么关联性，也就是一堆数据而已。除了可以看到各自数据点的变化以外，没有经过任何的数据挖掘，发挥不出更大的价值。

目前电厂希望通过一个统一的数据平台，将各种数据关联起来，通过分析可以找到一些关于设备检修的规律，对设备的故障、损耗可以做到预判，尽可能提前做好设备检修、故障处理，从而减少风险，让电厂更加安全稳定的运行。

作为这个大课题。如何利用电厂宝贵的数据财富形成统一的数据仓库，在海量数据中提取设备运行规律，我们提出了通过可扩充、可组态的算法库，对设备的当前工况进行识别，对各种设备的未来运行趋势进行预判。并以先进的可视化手段帮助运行维护人员及时发现故障、处理故障，最终形成设备检修计划决策支持系统。

在下面的各章中将记述二滩水电站设备状态检修数据可视化平台系统（以下，“本系统”为略称）的系统需求，功能设计，系统实现，系统测试过程中所涉及到的问题及解决方法。

2. 开发环境

2.1 硬件开发环境

1. 服务器 12 台

采用 DELL R720 服务器，参数如表 2.1 下所示：

表 2.1 配置参数表

配置	参数
CPU	Inter Xeon E5-2620 v2 6C 2.6GHz (MAX) CPU 核心 6 个 内核
内存	128G
硬盘	2×1.2TB, SAS 接口, 1 万转, 支持 raid1
网口	2×1000MB 自适应网口
其他	冗余电源及风扇
操作系统	Linux 中文

2. 电源：2 路冗余 220VAC 供电

3. 网络方案

如图 2-1 虚线部分所示, 因为用到了 MongoDB 数据库的分片集群的方式, 所使用了 9 台服务器作为数据库服务器, 2 台作为应用服务器。

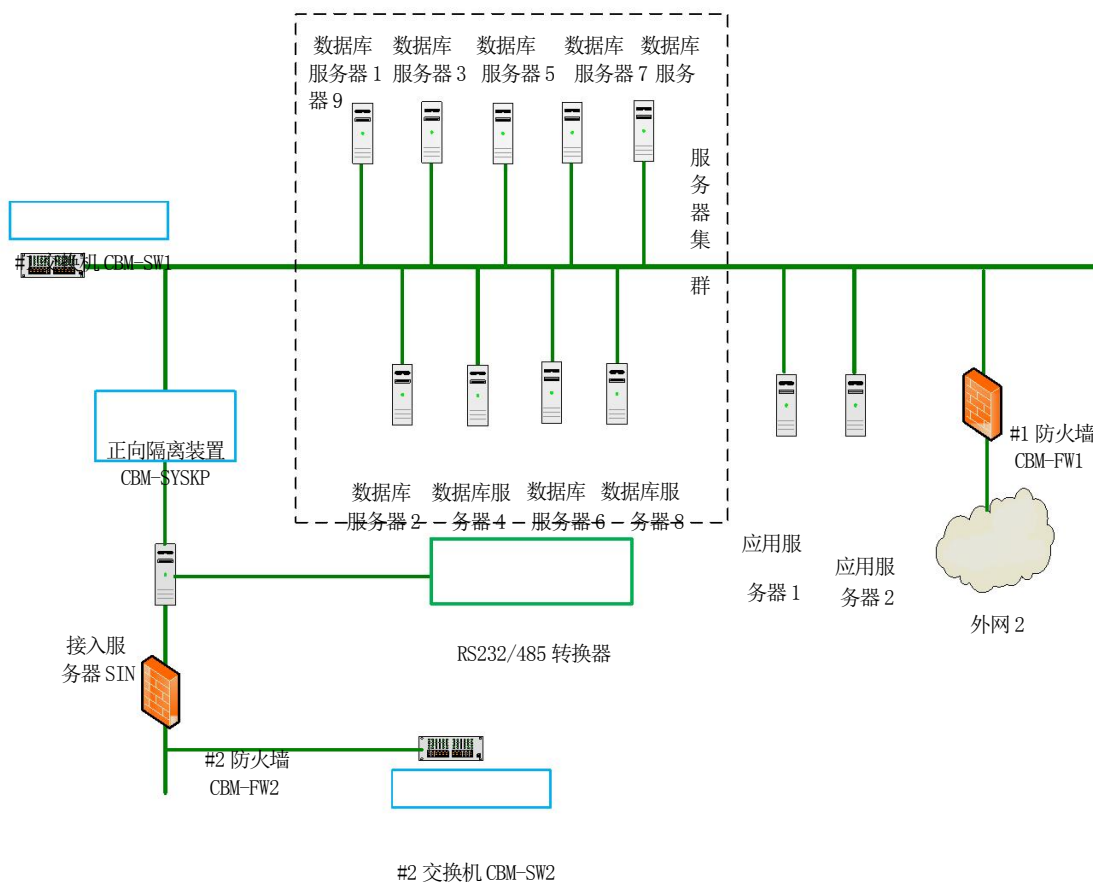


图 2-1 网络方案图

2.2 软件开发环境

1. C#三层架构

C#的三层架构将数据层、应用层和业务层分离，业务层通过应用层访问数据库，保护数据安全，利用负载平衡，提高运行效率。

表示层（UI）位于最外层用于显示数据和接收用户输入的数据，为用户提供一种交互式操作界面。

业务逻辑层（BLL）针对具体问题的操作，也可以说是对数据层的操作，负责对数据业务逻辑处理。

数据访问层（DAL）该层所做事务直接操作数据库，主要负责对数据的增添，删除、修改、更新、查找等操作。

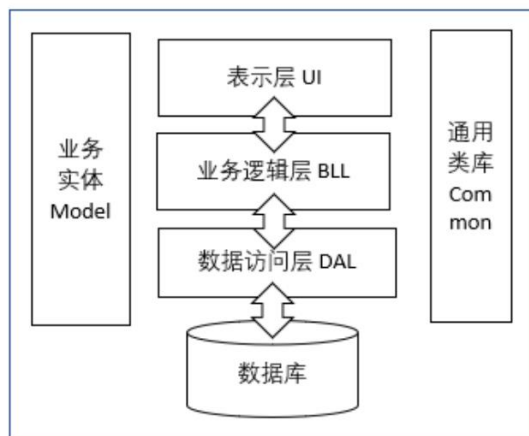


图 2-2 三层结构图

2. DevExpress 控件

DevExpress 开发的控件有很强的实力，不仅功能丰富，应用简便，而且界面华丽，更可方便定制。是一个 .NET 平台的用户界面套装，它包含 Grid、Chart、Reporting、Tree-Grid 等 100 多个功能子控件，同时套包内包含 Winform、WPF、Silverlight、.net 版本和 .NET Application Framework 开发框架。很适用于 Winform 应用程序开发。

3. MongoDB 数据库

MongoDB 是一个介于关系数据库和非关系数据库之间的产品，是非关系数据库当中功能最丰富，最像关系数据库的。它支持的数据结构非常松散，是类似 JSON 的 BSON 格式，因此可以存储比较复杂的数据类型。

它的特点是高性能、易部署、易使用，存储数据非常方便。

文档是 MongoDB 中数据的基本单位，类似于关系数据库中的行（但是比行复杂）。多个键及其关联的值有序地放在一起就构成了文档。集合就是一组文档，类似于关系数据库中的表。集合是无模式的，集合中的文档可以是各式各样的。例如，{“name”：“Mike”}和{“foo”：3}，它们的键不同，值的类型也不同。MongoDB 中多个文档组成集合，多个集合组成数据库。一个 MongoDB 实例可以承载多个数据库。它们之间可以看作为相互独立的关系，每个数据库都有独立的权限控制。在磁盘上，不同的数据库存放在不同的文件中。

4. C/S 结构

二滩水电站设备状态平台系统的体系架构设计中，采用 C/S 架构（Client/Server，客户机 / 服务器模式）。Client 和 Server 常常分别处在相距很远的两台计算机上，Client 的任务是将用户的要求提交给 Server，再将从 Server 返回的结果以特定的形式显示给用户；Server 的任务是接收 Client 提出的服务请求，进行相应的处理，再将结果返回给 Client。

2.3 本章小结

本章主要介绍了本系统的硬件开发环境和软件开发环境。硬件主要介绍了服务器的配置参数和网络配置方案；软件开发环境主要介绍了所用到的编程语言、数据库及架构，通过这些不同的技术使程序完成所需的功能。

3. 需求分析

本系统主要实现大量历史数据存储、数据统计与分析处理功能。以实现水电站主设备运行状态的监测，及时发现设备可能存在的隐患，为检修决策与故障分析提供技术依据。对历史数据库记录的大量设备运行状态数据进行分析 and 追忆，

可为设备检修提供丰富的历史数据信息，以便做出最优决策。

本系统的开发主要包括应用程序系统和数据库。应用程序系统要求界面友好，功能完备，易使用等特点。

系统主要可以满足以下几个功能：

1. 界面展示功能

通过界面把关心的数据点的历史数据进行图形展示、表格查询、用户管理。图形界面基本功能是要做到以下几点：

- （1） 用户管理：不同的用户根据自己的权限登录界面，可以有不同权限对界面进行操作。管理员可以添加用户和确认缺陷信息。
- （2） 设备以树的形状可以展示。可以自定义添加以下目录和节点。
- （3） 对象树上的模拟量和状态量节点可以进行历史数据查询。查询模式以图形曲线方式展示。多点可以通过统一坐标展示或者多图同时方式展示。
- （4） 可以对历史事件进行列表查询

2. 后台数据维护功能

定期导入客户给定的指定格式的数据文件到数据库，做基本的统合计算。

4. 系统设计

4.1 系统功能设计

1. 系统功能设计图，如图 4-1 所示：

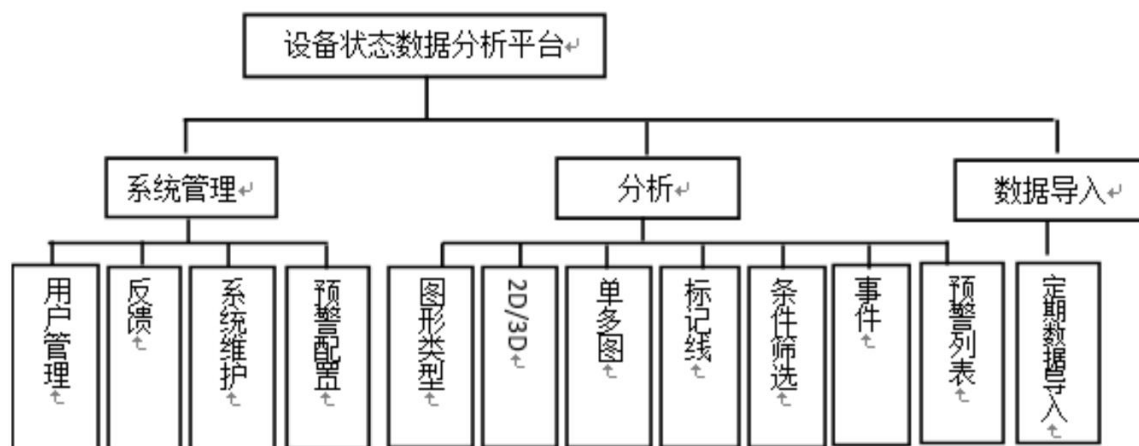


图 4-1 系统功能设计图

2. 功能说明

(1) 用户管理

主要是对登陆用户的信息进行查询、增加、删除、以及修改处理。

(2) 反馈

登陆缺陷信息，并且在完成后登陆确认。

(3) 系统维护

展示数据库的基本状态信息。

(4) 预警配置

需要配置预警信息的点名称和预警参数模型进行维护的界面。

(5) 图形类型

对于主要被查询点的历史信息进行各种图形展示选择。主要可以展示折线图、曲线图、蜡烛棒图、蜡烛线图、柱状图、散点图。

通过选择数据点和查询时间。进行曲线查询，然后再根据选择的图形类型就可以进行不同类型图形的查询。

(6) 2D/3D

部分图可以进行 2D 和 3D 转换的显示。

(7) 单多图

单图或者多图曲线展示。

(8) 事件功能

查询监控系统历史报警事件。

(9) 可以查询一段时间内的预警事件列表。

(10) 数据导入

定时对模拟量和状态量历史数据进行数据导入。

4.2 数据库设计

本系统的数据库设计包含了 10 种 MongoDB 数据库文档用来实现对数据的存储和对数据的调用。

1. event_record 事件集合：事件导入时应做的事件动作频次统计。

2. analog_measure_value_XXX_new 模拟量集合：根据时间最小单位按照分桶封装的方式进行存储，每个小时的数据（最多 60 点）分装入一个桶内打上以小时数结尾的时标，如 2015-09-09T14:00:00 表示 2015 年 9 月 9 日 14 点内的分钟数据。

为了降低 MongoDB 内每个集合的文档的数量，对表格按照系统进行划分集合，集合的名字为 analog_measure_value_XXXX_new, 最后的 XXX 为系统的名字，XXX 取值以数据字段 lname 中的第二段为准。如：

ETN.01LCU.COM.6303_TR_SWI.MVAR_REAL 中以字段 01LCU 为划分依据。进入到集合 analog_measure_value_01LCU_new 中，其他以此类推。

3. data_crawl_info 采集信息集合：主要是为了数据采集程序在数据采集时记录采集的状态，包括已经插入的数据状态，当前插入的状态和最后的更新数据插入状态。每次插入一个文件之前，应该在本文档查询该时间段内，检查是否对该文件名做过插入，如果没有，执行插入操作，否则跳过本文件。同时也可以查询最后插入的时间和文件的名称，并从最后插入的时间段和文件顺序往下做插入操作。

4. equipment_tree 设备树集合：保存设备树节点之间的父子关系和层次关系。

5. analog_measure_meta_data 模拟量数据元集合：保存模拟量数据点的属性。

6. status_measure_meta_data 状态量数据元集合：保存状态量的属性。

7. statistic_job_def 任务定义集合：任务定义文档主要用于定义数据周期处理的作业内容。比如统计操作描述了从什么文档取数据，满足哪些条件进行统计，进行哪些类型的统计，统计结果要写入到哪个文档中等。该文档由负责执行数据统计的后台进程周期扫描并按照统计内容和条件进行统计汇总至相关文档中。

8. statistic_result_info 统计结果集合：统计作业进程按照任务定义集合（statistic_job_def）预定义周期统计好后，统计量按照数据点和时间段（period）逐条形成聚合数据集合。

9. user_info 用户定义集合：保存用户信息。

10. bug_fix 障碍记录集合：保存缺陷修改信息

以下是各个集合的文档定义说明。

表 4.1 event_record 文档定义	
"name" : "六号机励磁电压",	//点的中文描述
"event_time" : ISODate("2015-08-03T23:45:56.820+0000"),	//动作时间
"valunit" : "V",	//单位
"value" : 562.73,	//值
"lname" : "ETN.06LCU.IO_1.PE_07.00_AI.00",	//逻辑名
"state" : "",	//状态值 如复归等（state 和 value&valunit 不能同时设定）
"logininfo" : "",	//
"cmdinfo" : "高限报警",	//值相对意义

```

{
  "name" : "六号机励磁电压",
  "event_time" : ISODate("2015-08-03T23:45:56.820+0000"),
  "valunit" : "V",
  "value" : 562.73,
  "lname" : "ETN.06LCU.IO_1.PE_07.00_AI.00",
  "state" : "",

  "logininfo" : "",
  "cmdinfo" : "高限报警"
}

```

表 4.2 analog_measure_value_XXXX_new 文档定义

```

{
  "_id" : ObjectId("5b32625026eb7f7946a94732"),
  "tmot" : NumberInt(1),           //数据时间最小精度（时间偏移量类型）
                                     0:second,1:minute,2:hour,3:day,4:week,5:month,6:year
  "recCount" : NumberInt(3),       //0, (INT) 该数据桶内的数据个数
                                     1-60个按实际数据个数填写
  "max" : 19.02,                   //本时间段的最大值
  "min" : 19.0,                    //本时间段的最小值
  "var" : 0.000006555555555555274, //本时间段的方差
  "avg" : 19.000333333333334,       //本时间段的平均值
  "lname" : "ETN.01LCU.FILT.STR_HOR_A", //点名
  "pt_id" : NumberInt(21892),       //数据点ID描述预留字段根据读写性
                                     能决定该描述与lname字段中英文描述一一对应
  "time" : ISODate("2018-03-30T19:00:00.000+0000"), //时间
  "mdn" : 19.0,                    //本时间段的中位数数值
  "vals" : [                       //桶装数据数组按照时间变化记录
    不变化不记录的写入方式；
    {
      "k" : "00",                  //最小刻度1（此处按照 "tmot":0为秒）
      "v" : 19.0,                  //0秒时候的数值
    },
    {
      "k" : "04",                  //4秒变位, 1-3秒不记录均与0秒同以下
      "v" : 19.02
    },
    {
      "k" : "05",
      "v" : 119.0
    }
  ]
}

```

表 4.3 data_crawl_info 文档定义

```

{
  "file_name" : "ARV_01M_JC",      //导入文件名
  "hostname" : "ETAPPI",           //执行导入的主机名称
  "file_time" : ISODate("2015-07-12T06:01:00.000+0000"), //文件的导入时间
  "fold" : "//2015.07.12_15H",      //导入文件保存的目录名称
  "insert_time" : ISODate("2017-04-18T09:41:17.471+0000"), //插入时间
  "write_doc_count" : NumberInt(60), //当前文件的数据条数
  "wr_pt_count" : NumberInt(140),   //数据点个数
  "failure_reason" : "",            //导入失败原因
                                     (insert_state=2时才記入)
  "insert_state" : NumberInt(1)     //导入状态1:正常导入
                                     2:代表导入失败
}

```

表 4.4 equipment_tree 文档定义

```

{
  "nodetype" : NumberInt(3),        //节点类型
  "parent" : ObjectId("58a6a3c926eb7fb07b995aad"), //父亲节点ID
  "_nodeid" : ObjectId("58a6a3c926eb7fb07b995abe"), //当前节点ID
  "lname" : "OILL",                 //当前节点英文名称
  "pname" : "油位",                 //当前节点中文名
  "path" : "ET/3F",                 //父节点位置的层次路径
  "ndlevel" : NumberInt(3),         //第几层
  "ndkind" : NumberInt(0),          //节点类型0: 原始节点 2: 手录入节点
  "sectionnum" : "1.7.8"           //节点编号
}

```

表 4.5 analog_measure_meta_data 文档定义

```
{
  "xpath" : "BT/1F/ENERGY",
  "name" : "一号机无功实际值", //描述名
  "serialno" : "11428", //点序号
  "groupid" : "", //分组ID
  "lname" : "ETN.01LCU.COM.6303_TR_SWI.MVAR_REAL", //逻辑名
  "datasource" : NumberInt(0),
  "corgroup" : "", //关联量组ID
  "lowrange" : "", //低限值
  "unit" : "MVar", //单位名
  "hirange" : "", //高限值
  "_parentid" : ObjectId("58a6a3c926eb7fb07b995a78") //父节点ID
}
```

表 4.6 status_measure_meta_data 文档定义

```
{
  "xpath" : "BT/1F/ENERGY", //所在节点路径
  "name" : "一号机正向无功电度", //中文描述
  "serialno" : "11411", //序列号
  "groupid" : "", //组ID
  "lname" : "ETN.01LCU.ENG_COM.P21.VH+", //点名
  "datasource" : NumberInt(0), //0:为系统自动导入1:用户离线输入数据
  "corgroup" : "", //关联量组ID
  "lowrange" : "", //低限值
  "unit" : "GVarh", //单位
  "hirange" : "", //高限值
  "_parentid" : ObjectId("58a6a3c926eb7fb07b995a78") //父ID
}
```

表 4.7 statistic_job_def 文档定义

```
{
  "taskname" : "statistic of status", //0:任务类型 1:暂定
  "lastexecutetime" : ISODate("2017-03-14T12:16:53.244+0000"), //最后更新时间
  "lname" : "ETN.03LCU.IO_1.PE_05.01_DI.05SI", //点名
  "periodType" : NumberInt(1), //0:分 1:小时 2:天 3:月 4:年 此处为按小时统计
  "fromtables" : "event_record", //读取的表名字
  "stat_content" : [ //统计量:统计
    "event_record.state":"运行"
  ],
  {
    "fieldname" : "state",
    "fieldvalue" : "运行",
    "operationtype" : "SUM",
    "updatefield" : "closenum"
  },
  {
    "fieldname" : "state",
    "fieldvalue" : "停止",
    "operationtype" : "SUM",
    "updatefield" : "closenum"
  }
],
  "settotable" : "status_measure_meta_data", //统计结果更新表
  "lastExecuteState" : true //最后一次统计状态
}
```

表 4.8 statistic_result_info 文档定义

```

{
  "lname" : "ETN.03LCU.IO_1.PE_05.01_DI.05SI",           //点名
  "from_time" : ISODate("2017-03-12T08:00:50.433+0000"), //统计开始时间
  "operationtype" : "SUM",                                 //计算类型
  "stat_values" : [                                       //状态值
    {
      "fieldName" : "运行",
      "value" : NumberInt(0)
    },
    {
      "fieldName" : "停止",
      "value" : NumberInt(0)
    }
  ]
}

```

表 4.9 user_info 文档定义

```

{
  "username" : "张梁",                                     //用户名
  "password" : "5fa285e1bebea6623e33afc4a1fbd5",         //密码MD5加密
  "description" : "厂领导",                                //描述
  "groupid" : 1,                                             //对象组
  "permissionCode" : "B11001001010",                       //权限
  "scheme" : {                                              //用户桌面的主题样式
    "UI_Theme" : "Darkroom",
    "fromtime" : ISODate("1970-01-01T00:00:00.000+0000"),
    "totime" : ISODate("1970-01-01T00:00:00.000+0000"),
    "myAnaMeasurePointCounts" : NumberInt(3), //保存点个数
    "myAnaMeasurePoints" : [                               //点名
      "ETN.01LCU.COM.6303_TR_SWI.MVAR_REAL",
      "ETN.01LCU.FILT.BL_V_Y",
      "ETN.01LCU.FILT.GLGO_LEVEL"
    ],
    "myStatusMeasurePointCounts" : NumberInt(2),
    "myStatusMeasurePoints" : [
      "ETN.03LCU.IO_1.PE_04.05_DI.07SI",
      "ETN.DC.YT.DC2.001"
    ]
  }
}

```

表 4.10 bug_fix 文档定义

```

{
  "recorder" : "许明勇",                                   //录入用户
  "dateTime" : ISODate("2017-04-01T10:11:33.356+0000"), //bug创建事件
  "content" : "提交的第一个BUG",                           //bug内容
  "status" : "已解决",                                       //状态
  "bugRecorder" : "许明勇",                                  //修改用户名
  "bugfix_result" : "66"                                     //修改原因
}

```

5. 系统实现

5.1 用户登陆界面实现

点击登陆按钮，根据填写的用户名和密码进行用户信息验证，并且抽取用户界面自定义信息。如图 5-1 所示：



图 5-1 用户登陆界面

5.2 主界面实现

用户登陆以后的第一个展示界面。

根据用户登陆信息确认是否是管理员身份，配置用户权限功能。

从集合 equipment_tree 抽取对象树信息，创建对象树。如有用户自定义信息也将被自动创建。如图 5-2 所示：

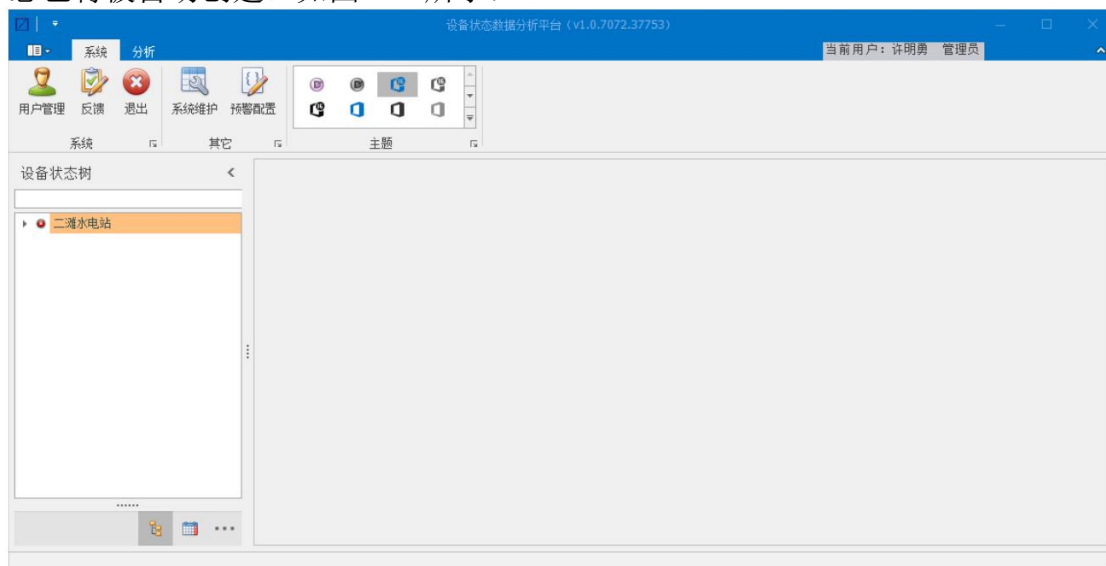


图 5-2 项目主界面

5.3 用户管理界面实现

用户权限分两种，一种是管理员用户，另外一种是一般用户。管理员用户登

陆的情况下第一个按钮显示为“用户管理”，普通用户的情况下显示为“修改密码”。点击“用户管理”按钮，就可以打开用户管理界面。用户管理界面主要是对用户进行添加、删除、密码修改、保存用户信息。用户信息被保存到集合 user_info。用户密码采用 MD5 的加密方式。用户管理界面如图 5-3 所示：



图 5-3 用户管理界面

修改密码界面。如图 5-4 所示：



图 5-4 修改密码界面

5.4 缺陷信息管理界面实现

缺陷信息管理界面是为客户提供一个管理缺陷信息的地方。普通用户提交缺陷信息，管理员有权力确认是否解决。关联集合 bug_fix。界面如图 5-5 所示：



图 5-5 缺陷管理界面

5.5 图形表示界面实现

选择对象树上的模拟量或者状态量节点。点击右键菜单可以查看选择曲线类

型。模拟量节点只有一个“查看曲线”菜单，状态量会出现数个统计曲线查看的菜单。选中菜单选项后右侧会弹出一个标签页，根据录入的查询时间间隔，就可以查询了。客户可以通过图线的变换趋势判断是否存在问题。如图 5-6、5-7、5-8 所示：

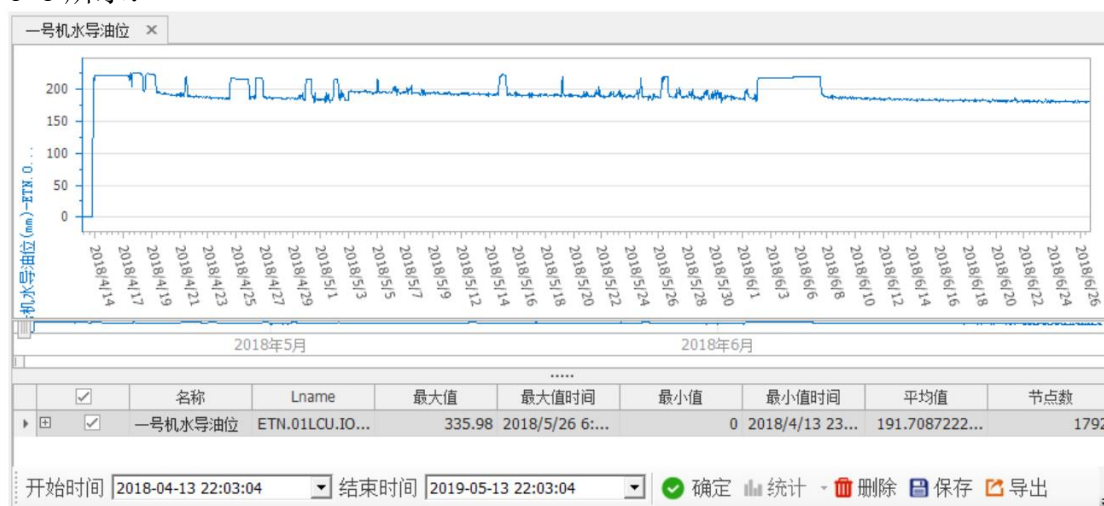


图 5-6 单条曲线图形界面

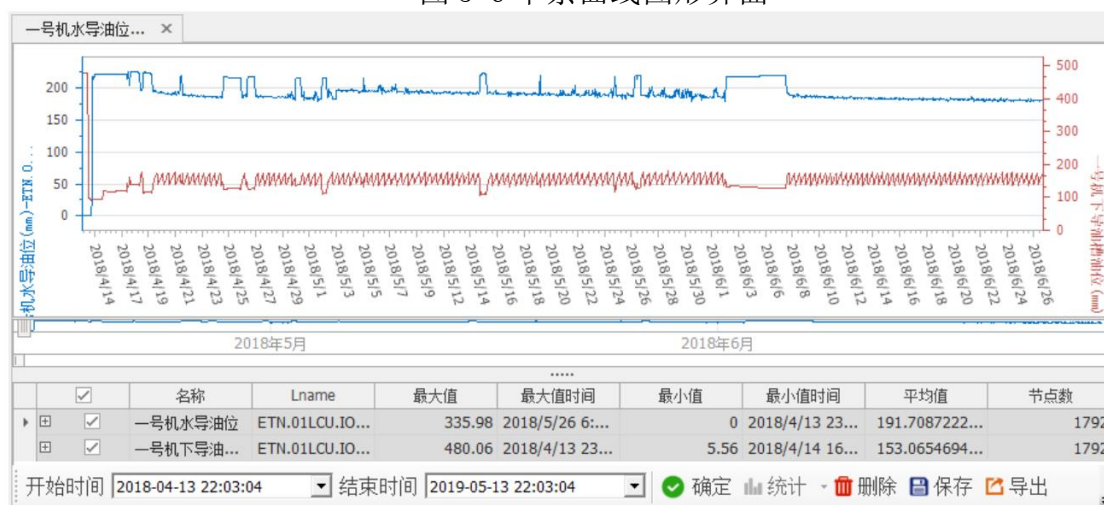


图 5-7 多点叠加图形界面

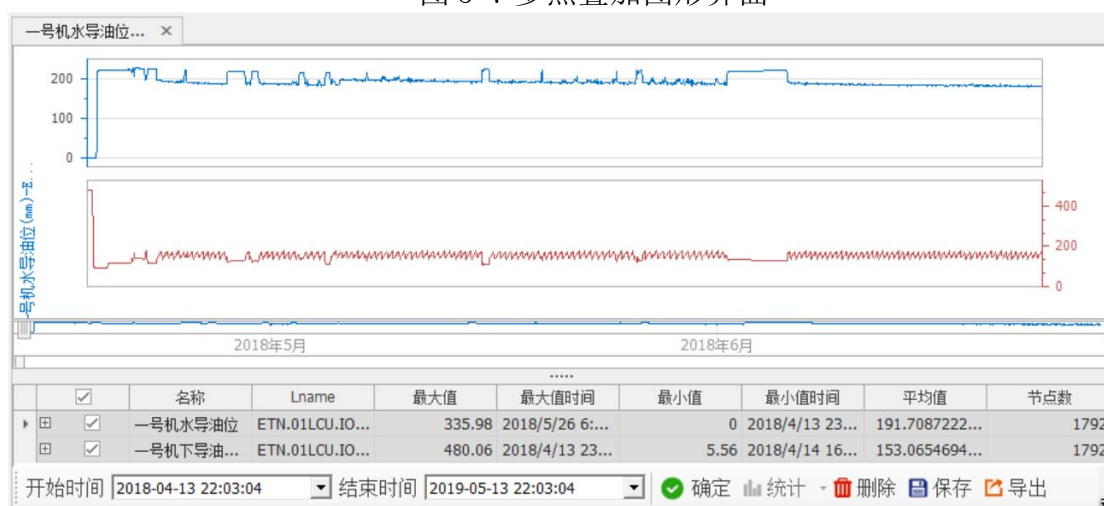


图 5-8 多点多图图形界面

5.6 数据导入后台实现

数据导入利用 Python 语言实现

1. 模拟量文件的定时导入

每晚 23 点开始查询目录/home/SMA2016/data/arv 目录下是否有目录存在，如有文件目录存在，则按照目录下的时间进行目录排序，根据从小到大顺序对每一个目录的文件进行处理。

(1) 抽取文件中第二行第一个点的第二个分割符字段与 'analog_measure_value_' 和 '_new' 组成类似 'analog_measure_value_XXX_new' 定义名备用。

(2) 把文件按照 csv 格式整体读入，对每个数据进行检查，删除空白行。

(3) 读取第一个有效数据的时间字段和文件名、最近一层目录名以及处理状态(正常值)作为检索条件查询集合 data_crawl_info 里是否有同样的记录存在，记录存在表示该文件的数据已经插入过则不再做数据插入处理，若没有同样的记录就做数据插入处理。

(4) 数据都是按照每 0-59 分的数据排列的，按点单位抽出，进行最大值、最小值、均值方差、平均值、以及中位数计算后把结果插入 (1) 拼出的表。

(5) 无论数据插入正常与否把处理结果写入文档 data_crawl_info。

(6) 当前目录下文件做完导入处理后把整个目录移动到 /home/SMA2016/data/bak/插入时间/home/SMA2016/data/arv 下一边做后期确认。

2. 状态量文件的定时导入

每晚 23 点开始查询目录/home/SMA2016/data/dsa 目录下是否有目录存在，如有文件目录存在，则按照目录下的时间进行目录排序，根据从小到大顺序对每一个目录的文件进行处理。

(1) 全文按照 csv 格式读入

(2) 删除空白行

(3) 按照有效数据的时间字段和文件名、最近一层目录名以及处理正常标志作为检索条件查询集合 data_crawl_info 里是否有同样的记录存在，如记录存在表示该文件的数据已经插入过则不再做数据插入处理，若没有同样的记录就做数据插入处理。

(4) 按照顺序把数据插入集合 event_record。

(5) 无论数据插入正常与否把处理结果写入集合 data_crawl_info。

(6) 当前目录下文件做完导入处理后把整个目录移动到 /home/SMA2016/data/bak/插入时间/home/SMA2016/data/dsa 下一边做后期确认。

5.7 本章小结

本章介绍了本系统的前端界面的基本功能。通过图形表示的方式介绍了用户登陆界面、用户管理界面、密码修改界面、缺陷信息界面、图形展示界面的设计。介绍了后台的模拟量以及状态量的数据处理方法。说明了整个系统的基本架构。

6. 系统测试

6.1 登录测试

- 测试流程
1. 输入管理员、普通账号、不存在的账号
 2. 确认登录
 3. 测试结果

登录测试过程如表 6.1 所示：

表 6.1 登录测试表

测试身份	测试账号	预期结果	实际结果	是否通过
管理员	用户名：admin 密码：123456	进入管理员页面	进入管理员页面	是
普通账号	用户名：test 密码：123456	进入主页，第一个按钮表示为修改密码	进入主页，第一个按钮表示为修改密码 错误账号	是
用户名：aaa	在登陆界面显示 密码：111	在登陆界面显示无此用户	是 显示无此用户	

6.2 用户管理界面动作测试

- 测试流程
1. 点击系统导航栏的“用户管理”
 2. 确认用户管理界面用户列表
 3. 添加用户
 4. 删除用户
 5. 修改密码
 6. 保存修改用户
 7. 测试结果

登录测试过程如表 6.2 所示：

表 6.2 用户管理界面测试表

操作	预期结果	实际结果	是否通过
点击“用户管理”按钮	用户管理界面正常表示，已经登陆的用户在一览表内正常显示	用户管理界面正常表示，已经登陆的用户在一览表内正常显示	是
点击“添加”按钮	添加用户界面正常弹出	添加用户界面正常弹出	是
在添加用户界面输入用户信息，点击“确认”按钮	新用户被追加到用户管理界面的列表里边	新用户被追加到用户管理界面的列表里边	是
在用户管理界面，选中一条已存在的用户信息，点击“删除”按钮	跳出是可以删除的确认信息，点击确认，被选中用户不在列表中显示	跳出是可以删除的确认信息，点击确认，被选中用户不在列表中显示	是

选中一个用户，点击“修改密码”按钮	修改密码界面弹出，设定新的密码，点击“确认”按钮，新密码反映到用户列表	修改密码界面弹出，设定新的密码，点击“确认”按钮，新密码反映到用户列表	是
修改用户列表的用户信息，点击“保存”按钮	用户信息被正常保存到数据库。 在登陆界面录入被修改的或者追加的用户名和密码，用户可以正常登陆。 被删除用户无法在登陆。	用户信息被正常保存到数据库。 在登陆界面录入被修改的或者追加的用户名和密码，用户可以正常登陆。 被删除用户无法再登陆。	是
修改用户列表的用户信息，点击“取消”按钮	用户信息没有改变	用户信息没有改变	是

6.3 反馈界面动作测试

- 试流程
1. 点击系统导航栏的“反馈”
 2. 确认 BUG 界面
 3. 提交按钮确认
 4. 保存按钮确认
 5. 测试结果

登录测试过程如表 6.3 所示：

表 6.3 反馈界面测试表

操作	预期结果	实际结果	是否通过
利用普通用户登陆界面，录入内容，点“提交”按钮	内容被正确提交到一览表；退出界面再次打开的时候在反馈界面可以看到上次提交的内容	内容被正确提交到一览表；退出界面再次打开的时候在反馈界面可以看到上次提交的内容	是
利用普通用户登陆界面，双击一览表	不能修改被提交信息	不能修改被提交信息	是
内容被正确提交到一览表；退出界面再交”按钮	内容被正确提交到一览表；退出界面再次打开的时候在反馈界面可以看到上次提交的内容	是 陆，录入内容，点“提交”按钮，再次打开的时候在反馈界面可以看到上次提交的内容	是
利用管理员用户登陆，双击一览表	确认信息部分可以被填写	确认信息部分可以被填写	是
钮 修改信息可以正常保存	修改信息可以正常保存	是 修改信息可以正常保存	是

6.4 数据点的图形展示测试

- 测试流程
1. 选择设备树上的一个节点，双击
 2. 设定一段数据查询时间

3. 点击“确认按钮”
4. 选择图形按钮里的任意类型
5. 测试结果

登录测试过程如表 6.4 所示：

表 6.4 数据点的图形展示测试表

操作	预期结果	实际结果	是否通过
设备树中选取数据点，双击	该点的图形表示界面在右侧打开。	该点的图形表示界面在右侧打开。指定该点的检	是
实时该点的折线图正常展示 间，点击“确定”按钮	该点的折线图正常展示	该点的折线图正常展示	是
在图形显示状态下，点击“图形类型”按钮下拉菜单，选择散点图	该点的散点图正常展示	该点的散点图正常展示	是
点击“保存”按钮	打开文件保存对话框，当前展示线图正常保存成 jpeg 图	打开文件保存对话框，当前展示线图正常保存成 jpeg 图	是
点击“导出”按钮	当前线图的数据以 csv 格式被保存成指定文件	当前线图的数据以 csv 格式被保存成指定文件	是
点击“保存方案”按钮	被选择点将保存在查询方案中。	被选择点将保存在查询方案中。	是

6.5 模拟量数据导入测试

测试流程：1. 在/home/SMA2016/data/arv 目录下放入一个客户给定的文件目录

2. 执行 python /home/SMA2016/script/import_csv 1 00

模拟量数据导入测试过程如表 6.5 所示：

表 6.5 动态成本模块测试表

测试状况	预期结果	实际结果	是否通过
从未插入过数据库的数据	<ol style="list-style-type: none"> 1. 数据被插入 analog_measure_value_XXX_new 的表。XXX 为数据表内第二个字段 2. 数据表 data_crawl_info 内有新数据被插入，处理结果正常 3. 在图形界面根据被插入数据点和时间进行查询，可以查到相应曲 	与预期结果一致	是

	线		
已经插入过的数据文件再次执行插入操作	1. 没有重复数据被插入	与预期结果一致	是
文件格式异常	数据没有被插入，数据表 data_crawl_info 的插入状态记录为异常，并提示插入错误原因	与预期结果一致	是

6.6 本章小结

本章主要对开发完成的系统进行了全面的测试。主要进行测试用例的设计用以检验功能的可实现性，测试内容包含：数据导入、界面登陆、用户管理、数据查询和图形表示的逻辑及功能实现。通过反复测试和对系统的不断完善，本设计已基本完成了预期目的。

结 论

通过开发二滩水电站设备状态检修数据平台，对于系统从需求分析、基本设计、详细设计、单元测试、系统测试等软件开发整体过程有了比较深刻的了解。一个系统的开发除了我们比较擅长的软件开发也要对硬件环境有所了解，很不错的体验。改变了以往只是关心单方面开发界面或者单方面开发后台程序的局限性，对整体数据流处理有了比较好的了解。

通过此次毕业设计，接触了以往不常用到的技术。前台我学会了基本用 C#.net 的三层构架的开发方式，后台学会了用现在比较流行的 Python 语言进行数据文件的简单处理。在数据库方面，打破了以往传统的使用关系数据库的开发模式。使用了目前也有不小话题性的文档型数据库 MongoDB。从 MongoDB 的数据库搭建配置、到表的创建、数据的增加、删除、修改、查询等语法也有了比较深刻的学习。这些技术的学习对今后的工作有很多的帮助。

本系统的设计和开发上基本算是完善了，但是还是有很多细节上的不足。前端的图形模式切换处理还是做的比较粗糙。状态量的曲线图和统计图之间目前还不能比较好的切换。在统计计算方面因为时间不够充分自定义计算处理还有待完善，未来有机会可以在已定义计算和报表创建方面进一步完善。这样对客户来讲也将有更好的体验。

系统不免有错误和待改善之处，欢迎各位老师提出宝贵意见。

致谢

时间一晃而过。充满梦想的大学生活即将结束，在这珍贵的大学生活和学习中，自己憧憬过，失落过，不仅有成功的喜悦，也还有过失败的悲伤，这期间的记忆和收获让人回味无穷。

在我的论文告罄之时，我要特别感谢我的导师 AAA 老师，从我开始选题到最终完成论文，都给了我关怀和指导，在写作过程中，老师细心地指导，使我迈过了一道道的坎。

感谢我的各位任课老师，在学校期间很荣幸可以得到各位老师的指导，让我能尽快融入学院各门课程的学习，顺利完成各门课程。

感谢大家！

参考文献

- [1] Wesley Chun, 孙波翔, 李斌, 李晗, Python 核心编程核心编程（第 3 版）, 人民邮电出版社, 2016 年 6 月
- [2] Eric Matthes, 袁国忠, Python 编程从入门到实践, 人民邮电出版社, 2016 年 07 月
- [3] AI Sweigart, Python 编程快速上手—让繁琐工作自动化, 人民邮电出版社, 2017 年 6 月
- [4] Kristina Chodorow, MongoDB 权威指南（第 2 版）, 人民邮电出版社, 2017 年 4 月
- [5] Kyle, Banker, Peter, Bakkum 等 MongoDB 实战, 华中科技大学出版社, 2017 年 2 月
- [6] Shakutala Gupta Edward, Navin Sabharwal, 蒲成, MongoDB 实战 构架、开发与管理, 清华大学出版社, 2016 年 12 月
- [7] 张甦, 贺磊, MongoDB 运维实战, 电子工业出版社, 2018 年 09 月
- [8] DevExpress 控件中文网
- [9] 克里斯琴. 内格尔, 李明译 C#高级编程（第 11 版）C#7&.Net Core2.0 清华出版社, 2019 年 2 月
- [11] Jon Skeet, 姚麒麟, 深入理解 C#（第 3 版）, 人民邮电出版社, 2014 年 04 月
- [12] 刘锐宁, Visual C# 开发实战 1200 例（第 1 卷）, 清华大学出版社, 2011 年 01 月
- [13] 克利里, C#并发经典实例（影印版）, 东南大学出版社, 2015 年 02 月