**隧道结构健康监测平台系统设计**

**版本号: V1.0**

|  |  |  |
| --- | --- | --- |
| **时 间** | **作 者** | **更改内容** |
| 2024-10-11 | 常爱军 | 创建文档V1.0。 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

本资料为上海霍开光电技术有限公司内部资料， 所有权归霍开光电技术有限公司所有。未经授权许可，不得传播分发。

目录

[1. 引言 3](#_Toc9175)

[2. 总体设计 3](#_Toc14129)

[2.1 需求规定 3](#_Toc30223)

[2.2 系统总体结构设计 3](#_Toc9699)

[2.3 系统架构分层设计 4](#_Toc2962)

[2.4 数据库设计 5](#_Toc2993)

[2.4.1 数据库研究及总体设计 5](#_Toc1475)

[2.4.2 用户数据库 hk\_user的设计 6](#_Toc15982)

[2.4.3 FBG数据库 hk\_fbg的设计 7](#_Toc7482)

[2.4.4 隧道监测报警数据库 hk\_alarm的设计 8](#_Toc11805)

[2.4.5 Web数据库操作类（数据处理层）的设计 9](#_Toc23698)

[2.4.6 数据收集程序数据库操作类的设计(C++类) 11](#_Toc13062)

[2.5 系统运行环境 13](#_Toc15012)

[2.6 系统开发语言及开发工具 14](#_Toc12680)

[3. 系统功能设计 14](#_Toc22423)

[3.1 FBG传感器数据收集程序模块的功能设计 14](#_Toc26983)

[3.1.1 FBG传感器数据收集程序的程序流程图（图5） 15](#_Toc31596)

[3.1.2 程序界面设计 15](#_Toc13484)

[3.1.3 与FBG传感器系统连接的接口协议 16](#_Toc10737)

[3.1.4 接口函数设计 18](#_Toc23113)

[3.1.5 重连机制设计 18](#_Toc32571)

[3.1.6 与mysql数据库的连接、存取 19](#_Toc31523)

[3.2 Socket服务器端类的设计 19](#_Toc24865)

[3.2.1 Socket服务器端类成员变量的设计 19](#_Toc5173)

[3.2.2 Socket客户端类成员函数的设计 20](#_Toc13638)

[3.3 监测管理及展示web应用平台的功能设计 23](#_Toc9947)

[3.3.1功能模块设计 23](#_Toc2227)

[3.3.2 隧道结构健康监测平台数据管理功能设计 24](#_Toc24726)

[3.3.3 账号管理的设计 31](#_Toc15639)

[3.3.4 大屏展示设计 39](#_Toc9668)

[3.3.5 页面框架设计 41](#_Toc12577)

[3.3.6 程序运行设置 44](#_Toc16147)

# 引言

本文档为隧道结构健康监测平台设计说明书，对系统架构、系统层次划分、模块功能、模块实现方式进行了设计。目的是让软件开发人员根据本文档的内容进行程序开发，使得开发的产品符合用户的需求，同时为测试人员提供参考。

产品名称：隧道结构健康监测平台

# 总体设计

## 需求规定

FBG 传感器采用波分复用技术可实现一根光纤上传感多个传感器，实现温度、应力、应变、加速度、位移等多种物理量的实时测量，从而监测被测物的开裂、渗压、沉降、温度变化等。

本软件平台主要用于接受隧道内FBG传感器系统传来的实时数据和隧道开裂、渗压、沉降、温度变化报警数据，将这些数据储存到数据库，并将这些数据以图像或报表形式呈现给用户。用户可以对报警信息进行确认或设置为误报。

## 2.2 系统总体结构设计

平台采用C/S、B/S混合架构，分为后台数据收集模块、前台展示模块、数据库三个部分。系统各模块组件之间、系统组件与FBG传感器子系统组件之间通过工业局域网互联（见图1）

后台数据收集模块功能：运行于数据收集服务器上。该模块作为client端与FBG传感器系统相连，通过传感器生产厂家自定义协议从传感器服务器获取现场采集到的数据或报警信息，并把它们存储到数据库中。

前台展示模块功能：运行于IIS web服务器上。用户通过浏览器登录Web应用程序，查看存储在数据库中采集到的现场数据和报警信息。

数据库系统: 存储从现场FBG传感器收集到的实时数据和报警数据，存储用户数据及电缆信息数据等。

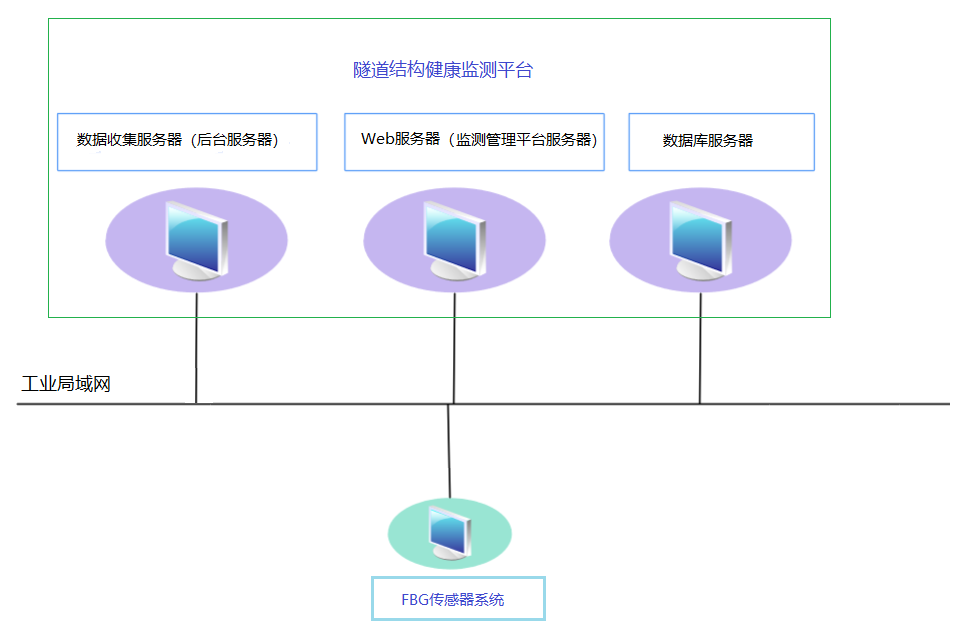


图1 系统结构示意图

## 2.3 系统架构分层设计

系统架构图见图2. 数据管理系统的总体架构包括数据采集层、数据传输层、数据库处理层、逻辑业务处理层、表示层、应用层。

数据采集层：即FBG传感器系统，负责现场数据的采集， 并作为C/S架构的客户端，连接到数据收集模块，通过FBG传感器生产厂家自定义协议将采集到的现场数据和报警数据发给数据收集模块 。该层系统由第三方供应商提供，不属于隧道结构健康监测平台系统开发范畴。

数据传输层：在本系统中就是数据收集程序模块，该模块通过FBG传感器生产厂家自定义协议接受传感器系统采集到的数据，并存储到数据库系统中。

数据处理层和逻辑业务处理层：包括web应用程序和数据库系统。根据用户发出的请求，web应用程序从数据库中查询数据，以图表、统计图、报表等形式返回给浏览器，并按照设计好的逻辑对相关数据作相应的处理后写回数据库。

表示层和应用层: 指web浏览器，提供了用户和系统的交互接口，接受用户输入的信息并发送到逻辑业务处理层，并将逻辑业务处理层返回的数据以页面形式展示给用户。

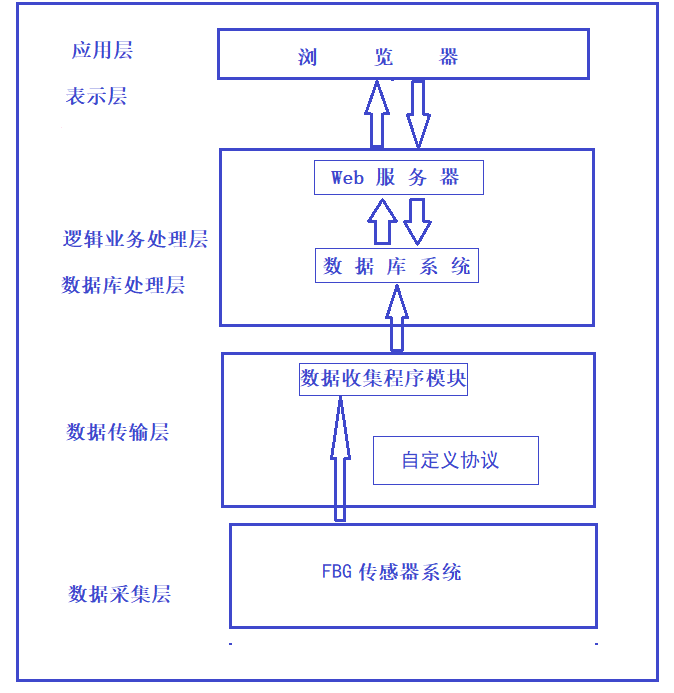


图2 系统架构图

## 2.4 数据库设计

### 2.4.1 数据库研究及总体设计

FBG现场传感器系统有自己的数据库，但隧道结构健康监测平台不能共用传感器系统的数据库，主要有以下几点原因：首先，随着时间的推移，数据库中存储的数据量会不断增长，一段时间之后，数据查询时的负荷有可能会很大，如共用传感器系统的数据库，有可能会FBG传感器设备不能将采集到的数据写到数据库中， 严重的甚至会导致负责现场采集的FBG传感器设备程序崩溃；其次，如果隧道结构健康监测平台与每个传感器子系统的数据库相连，会造成隧道结构健康监测平台的实现变得很复杂；另外，由于FBG现场传感器系统的数据库由传感器供应商开发并实现，对于历史数据，随着数据存储量的不断增加，在累积到一定时间之后，出于对系统性能、存储数据的磁盘空间优化、保证传感器子系统正常运行等方面的考虑，一般都会以一定的算法，对历史数据予以清理，传感器系统自身的数据清理算法很可能并不符合生产实际要求，造成某些不应该被清理的宝贵数据被清理了。因此, 隧道结构健康监测平台必须有自己独立的数据库，将FBG现场传感器传来的数据保存在自己的数据库中。在数据库系统的选择上，可以选择成熟的主流关系数据库系统如SQL Server、Oracle、MySql等。

由于MySQL在系统集群方面的优势，且属于开源项目，在开源社区能找到比较齐全的技术资料。考虑到系统将来的扩展、升级，本系统采用MySQL作为数据库管理系统。

在建立数据库表时，将建立一个实时数据表，以用于实时图形的显示；再分别为开裂、渗压、沉降建三个历史数据表，历史数据库表与实时数据库表结构上完全一致，实时数据表中的数据将定时同步到开裂、渗压、沉降三个历史数据表中，同步后的数据将从实时数据表中删除，这样的设计可大大减轻实时数据表的写入查询压力；另外，还需建立一个报警数据表，用以记录报警数据。

由于从实时数据库向历史数据库复制的工作可以由数据库管理系统自身的job任务来完成，属于数据管理范畴，超出了本系统的开发范围，故在本系统中只考虑实时数据库的设计及存储。

数据库由以下三个库组成：

* 用户数据库： hk\_user

用于存放与用户管理，用户登陆，用户权限相关的数据表。

* FBG数据库： hk\_fbg

用于存放现场采集到的开裂、渗压、沉降等相关的数据表。

* 报警数据库: hk\_alarm

用于存放开裂、渗压、沉降等报警相关的数据表。

### 2.4.2 用户数据库 hk\_user的设计

hk\_user\_info表：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **字段名称** | **中文说明** | **键别** | **是否默认为空** | **数据类型** | **长度** | **备注** |
| id | 序号 | PK | 否 | INT | 20 | 自增 |
| user\_name | 用户名 |  | 否 | varchar | 50 |  |
| pass\_word | 密码 |  | 否 | varchar | 50 |  |
| user\_role | 用户角色 |  | 是 | INT | 10 | 1. 管理用户 2. 一般用户 |
| create\_time | 创建时间 |  | 是 | varchar | 50 | 格式为：  YYYY/MM/DD hh:mm:ss |
| last\_login\_time | 最后登陆时间 |  | 是 | varchar | 50 | 格式为：  YYYY/MM/DD hh:mm:ss |
| is\_active | 是否激活 |  | 是 | tinyint | 1 | 1. 未激活 2. 已激活 |

### 2.4.3 FBG数据库 hk\_fbg的设计

FBG数据库由以下几张表组成:

* 隧道监测实时数据表： hk\_fbg\_real\_data\_info
* 隧道开裂历史数据表： hk\_fbg\_his\_gap\_info
* 隧道渗压历史数据表: hk\_fbg\_his\_shenya\_info
* 隧道沉降历史数据表: hk\_fbg\_his\_chenjiang\_info

传感器位置信息表: hk\_fbg\_sen\_info

2.4.3.1 隧道监测实时数据表：hk\_fbg\_real\_data\_info

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **字段名称** | **中文说明** | **键别** | **是否默认为空** | **数据类型** | **长度** | **备注** |
| id | 序号 | PK | 否 | INT | 20 | 自增 |
| dev\_id | 设备号或设备名称 |  | 是 | varchar | 50 |  |
| create\_time | 接受时间 |  | 否 | varchar | 50 | 格式为：  YYYY-MM-DD hh:mm:ss |
| sor\_code | 传感器号或名称 |  | 是 | varchar | 50 |  |
| data | 数据值 |  | 是 | varchar | 32 |  |

2.4.3.2 隧道裂隙历史数据表：hk\_fbg\_his\_gap\_info

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **字段名称** | **中文说明** | **键别** | **是否默认为空** | **数据类型** | **长度** | **备注** |
| id | 序号 | PK | 否 | INT | 20 | 自增 |
| dev\_id | 设备号或设备名称 |  | 是 | varchar | 50 |  |
| create\_time | 接受时间 |  | 否 | varchar | 50 | 格式为：  YYYY-MM-DD hh:mm:ss |
| sor\_code | 传感器号或名称 |  | 是 | varchar | 50 |  |
| data | 数据值 |  | 是 | varchar | 32 |  |

2.4.3.3 隧道渗压历史数据表：hk\_fbg\_his\_shenya\_info

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **字段名称** | **中文说明** | **键别** | **是否默认为空** | **数据类型** | **长度** | **备注** |
| id | 序号 | PK | 否 | INT | 20 | 自增 |
| dev\_id | 设备号或设备名称 |  | 是 | varchar | 50 |  |
| create\_time | 接受时间 |  | 否 | varchar | 50 | 格式为：  YYYY-MM-DD hh:mm:ss |
| sor\_code | 传感器号或名称 |  | 是 | varchar | 50 |  |
| data | 数据值 |  | 是 | varchar | 32 |  |

2.4.3.4隧道沉降历史数据表: hk\_fbg\_his\_chenjiang\_info

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **字段名称** | **中文说明** | **键别** | **是否默认为空** | **数据类型** | **长度** | **备注** |
| id | 序号 | PK | 否 | INT | 20 | 自增 |
| dev\_id | 设备号或设备名称 |  | 是 | varchar | 50 |  |
| create\_time | 接受时间 |  | 否 | varchar | 50 | 格式为：  YYYY-MM-DD hh:mm:ss |
| sor\_code | 传感器号或名称 |  | 是 | varchar | 50 |  |
| data | 数据值 |  | 是 | varchar | 32 |  |

2.4.3.5 传感器位置信息表: hk\_fbg\_sen\_info

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **字段名称** | **中文说明** | **键别** | **是否默认为空** | **数据类型** | **长度** | **备注** |
| id | 序号 | PK | 否 | INT | 20 | 自增 |
| dev\_id | 设备号或设备名称 |  | 是 | varchar | 50 |  |
| create\_time | 接受时间 |  | 否 | varchar | 50 | 格式为：  YYYY-MM-DD hh:mm:ss |
| sor\_code | 传感器号或名称 |  | 是 | varchar | 50 |  |
| sor\_loc | 传感器位置信息 |  | 是 | varchar | 100 |  |
| sensor\_type | 传感器类型 |  | 是 | INT | 20 |  |

### 2.4.4 隧道监测报警数据库hk\_alarm的设计

隧道监测报警数据库由以下表组成：

* 隧道监测报警数据信息表：hk\_fbg\_alarm\_info

2.4.4.1 隧道监测报警数据信息表：hk\_fbg\_alarm\_info

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **字段名称** | **中文说明** | **键别** | **是否默认为空** | **数据类型** | **长度** | **备注** |
| id | 序号 | PK | 否 | INT | 20 | 自增 |
| alarm\_type | 报警类型 |  | 是 | INT | 50 | 1. 开裂 2. 渗压 3. 沉降 4. 温度 |
| dev\_id | 设备号或设备名称 |  | 否 | varchar | 50 |  |
| alarm\_time | 报警时间 |  | 是 | varchar | 50 | 格式为：  YYYY-MM-DD hh:mm:ss |
| sor\_code | 传感器号或名称 |  | 是 | varchar | 50 |  |
| data | 数据值 |  | 是 | varchar | 32 |  |
| is\_show | 是否已处理 |  | 默认为0 | tinyint | 1 | 0：未处理  1：已确认  2：误报 |

### 2.4.5 Web数据库操作类（数据处理层）的设计

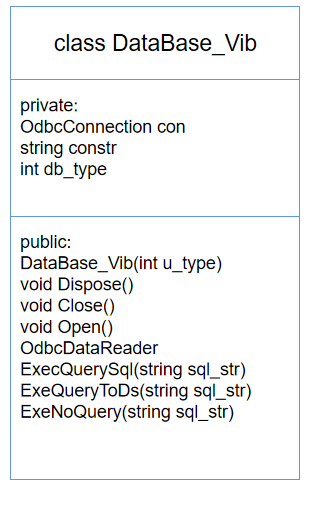


图3 asp.net数据库类

2.4.5.1成员变量

* OdbcConnection con： ODBC连接变量
* string constr：数据库连接字符串
* int db\_type: 数据库类型

1. 振动数据库,
2. botda应变数据库
3. dts温度数据库

4-用户数据库

2.4.5.2成员函数

* 构造函数：

public DataBase\_Vib(int u\_type)

输入：

u\_type: 数据库类型

1-振动数据库,

2-botda应变数据库

3-dts温度数据库

4-用户数据库

* 打开数据库:

void Open()

输入：

无

输出：

无

* 执行数据库查询:

OdbcDataReader ExecQuerySql(string sql\_str)

输入:

sql\_str：查询sql语句

输出:

OdbcDataReader类型的对象

* 执行数据库查询:

DataSet ExeQueryToDs(string sql\_str)

输入:

sql\_str：查询sql语句

输出:

DataSet类型的对象

* 执行数据库修改、插入、删除等无返回操作

ExeNoQuery(string sql\_str)

输入:

sql\_str：sql操作语句

输出:

无

* 关闭数据库

void Close()

输入:

无

输出:

无

* 自析构函数

void Dispose()

输入:

无

输出:

无

2.4.5.3 代码示例:

DataBase\_Vib db\_vib= new DataBase\_Vib(5); //构建一个FBG数据库实例

db\_vib.Open(); //打开数据库

//执行数据库查询，返回OdbcDataReader的实例

OdbcDataReader rd = db\_vib.ExecQuerySql("select count(\*) c from (select

id from hk\_fbg\_alarm\_info where is\_show=0) T;");

if (rd.Read())

{

Label\_vi\_total.Text = rd["c"].ToString(); //读取数据

}

rd.Close(); //关闭OdbcDataReader实例

db\_vib.Close(); //关闭数据库

db\_vib.Dispose(); //释放内存资源

### 2.4.6 数据收集程序数据库操作类的设计(C++类)

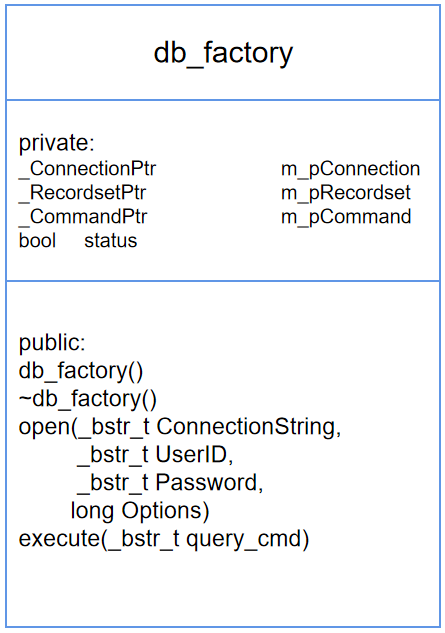


图4 数据库操作C++类

2.4.7.1 成员变量

* \_ConnectionPtr m\_pConnection ：数据库连接对象
* \_RecordsetPtr m\_pRecordset： 数据集变量
* \_CommandPtr m\_pCommand： 命令指针
* Bool status： 数据库连接状态

2.4.7.2 成员函数

* 构造函数：

db\_factory()

输入：

无

输出:

无

* 打开数据库连接：

bool open(\_bstr\_t ConnectionString, \_bstr\_t UserID, \_bstr\_t Password, long Options)

输入:

bstr\_t ConnectionString: 数据库连接字符串

\_bstr\_t UserID: 数据库连接用户名

\_bstr\_t Password: 数据库连接密码

long Options： 连接选项，ODBC方式连mysql时，该值为adModeUnknown

输出:

bool值，连接是否成功

true： 连接成功

false: 连接不成功

* 执行数据库增、删、改、查

bool execute(\_bstr\_t query\_cmd)

输入:

\_bstr\_t query\_cmd： sql语句字符串

输出:

bool值，执行sql语句是否成功

true： 执行成功

false: 执行不成功

* 析构函数

~db\_factory()

输入：

无

输出：

无

2.4.7.3 代码示例:

db\_factory db;

bool con\_status;

con\_status=db.open(con\_string.c\_str(),dbusername,dbpassword,adModeUnknown);

if (!con\_status)

{

string temp="Can not connect to Database. Please check the setting!";

write\_log(temp.c\_str());

return -1;

}

db.execute(my\_sql.c\_str());

## 2.5 系统运行环境

工业局域网: 工业局域网作为系统各组成模块组件之间、系统与各传感器子系统之间相互通讯的载体。因此，系统的开发、测试、运行均需基于工业局域网。

数据库管理系统: mysql 5.7及以上

MySQL ODBC驱动版本: mysql-connector-odbc-5.3.12-winx64及以上

系统各模块的运行环境:

|  |  |  |
| --- | --- | --- |
| 模块 | 操作系统 | 备注 |
| 数据收集程序 | Windows server 2008 及以上 |  |
| web应用程序 | Windows server 2008 及以上 | 系统需按照配置好IIS |
| 数据库管理系统 | Windows server 2008及以上，  或Linux CentOS7及以上 |  |

系统支持的浏览器:

Firefox, IE, Microsoft Edge

硬件要求：

CPU: 8核及以上

内存：8G及以上

## 2.6 系统开发语言及开发工具

数据收集程序：C/C++ （Visual Studio10以上）

Web应用程序： asp.net (Visual Studio2019)

# 系统功能设计

## FBG传感器数据收集程序模块的功能设计

数据收集程序作为服务器端接受FBG传感器设备Socket连接请求，建立连接，接受现场数据和报警信息并存储于数据库中。该程序模块不需要与最终用户直接交互，故可以做成windows服务程序。在首版开发中，暂先开发独立运行的windows 应用程序。

为了防止当收集某个FBG传感器系统数据时产生的异常对其他FBG传感器系统的数据收集产生影响，减少系统的耦合性和复杂度，每个数据收集程序实例只与特定的FBG传感器系统相连，收集特定传感器系统传来的数据。这些单独的运行实例可以最终运行在同一台服务器上，也可以运行在不同的服务器上。

### 3.1.1 FBG传感器数据收集程序的程序流程图（图5）

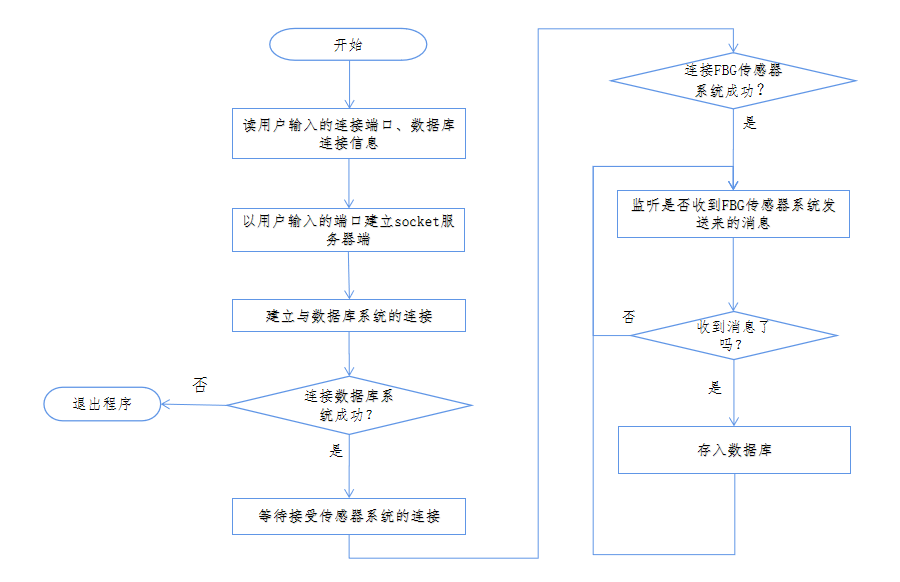


图5 振动数据收集程序的程序流程图

### 3.1.2 程序界面设计

程序界面用于接受用户的输入（图6）。在后续版本中，在将程序做成windows 服务后，可以用一个配置文件来代替，从而取消程序界面。

在程序执行时，用户需输入的输入项有： 开放连接的socket端口，数据库连接的ODBC DSN、数据库用户名、数据库密码，在输入这些参数后，程序将接受FBG传感器系统的连接请求，建立socket连接，接受FBG传感器系统发来的消息存储到数据库中，本机IP由程序自动填写。

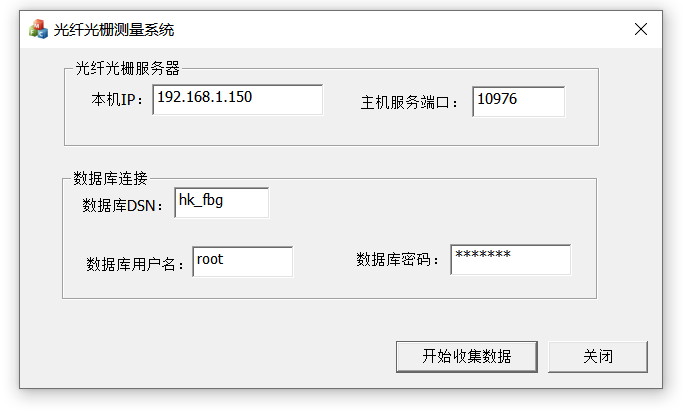


图6 FBG数据收集程序界面

### 3.1.3 与FBG传感器系统连接的接口协议

采用基于基于tcp/ip的基于tcp/ip的FBG传感器厂家自定义协议。消息格式：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **类型** | **参数名称** | **参数类型** | **长度** | **备注** | **参数说明** |
| 包头 | 包头 | uint8 | 2字节 | 0xFE0xEC |  |
| 协议版本 | uint8 | 1字节 | 0x10 |  |
| 报文类型 | uint8 | 1字节 | 0x50 |  |
| 设备编码 | char | 8字节 |  | 设备编码 |
| 包体长度 | uint32 | 4字节 |  | 整包大小 |
| 采集时间 | uint64 | 8字节 |  | 自1970年1月1日零时迄今的毫秒数(北京时间) |
| 采集频率 | Uint8 | 1字节 | 1 - 1Hz  10 - 10Hz  20 - 20Hz  其他值无效 | 数据上送的采集频率 |
| 数据区 | 传感器编码 | char | 6字节 | 传感器编码默认传感器参数设置中传感器编号 | 传感器编码 |
| 传感器值 | float | 4字节 \* 采集频率 | 值的采集时间（毫秒）=包头采集时间 + （n -1）\*1000/采集频率，n代表第n个数据，数据缺失或不够时使用FLT\_MAX补偿，该数值无效 | 传感器值 |
| … |  |  |  | 对编码及值循环 |
| 校验 | Crc校验 | Uint16 | 2字节 |  | 对数据区进行crc16校验 |
| 结束符 | 结束符 | Uint8 | 2字节 | 0D 0A | 0D 0A |

数据包示例：

数据：FE EC 10 50 44 45 4D 4F 00 00 00 00 00 00 00 31 00 00 01 8E 7F 88 5F 9C 01 32 2D 32 00 00 00 92 CD F5 3E 39 2D 31 00 00 00 33 09 B6 40 BF 49 0D 0A

包头：FE EC

协议版本：10

报文类型：50

设备编码：44 45 4D 4F 00 00 00 00 -> DEMO

包体长度：00 00 00 31 -> 长度49字节

采集时间：00 00 01 8E 7F 88 5F 9C -> 时间戳1711536627612 即为2024-03-27 18:50:27

采集频率：01 -> 1Hz传输

数据区：32 2D 32 00 00 00 92 CD F5 3E 39 2D 31 00 00 00 33 09 B6 40

32 2D 32 00 00 00 -> 传感器编码2-2

由于是1HZ传输，一个传感器仅一个数据：92 CD F5 3E -> 0.48008400201797485

(实际为: 0x3EF5CD92,已经是小端优先)

39 2D 31 00 00 00 -> 传感器编码9-1

同上，数据：33 09 B6 40 -> 5.688622951507568

CRC校验：BF 49

### 3.1.4 接口函数设计

* 构建与FBG传感器服务器连接的心跳包

int make\_ping\_request\_msg(unsigned char \*ps,int &msg\_len);

输入:

unsigned char \*ps： 消息buffer

int &msg\_len： 消息长度

输出：

int型变量:

0： 创建成功

-1： 创建失败

* 解析从FBG传感器服务器接受到的消息

int parse\_fbg\_msg(unsigned char \*\* messageBuf, int& msg\_len, char\* dev\_id, char\* rec\_time, int& data\_n, Sensor\_Node\*\* data)

输入:

unsigned char \*\* messageBuf： 消息buffer，当解析完成后，需要从消息buffer中移除该消息

int &msg\_len： 消息长度

char\* dev\_id：设备名称或设备ID

Sensor\_Node\*\* data：解析获得的Sensor\_Node格式的消息体数据

输出：

int型变量:

0： 解析成功

-1： 解析失败

1： 收到的是不完整消息，解析失败

Sensor\_Node的定义：

struct Sensor\_Node

{

char sensor\_id[7];

float data;

struct Sensor\_Node \*next;

};

### 3.1.5 重连机制设计

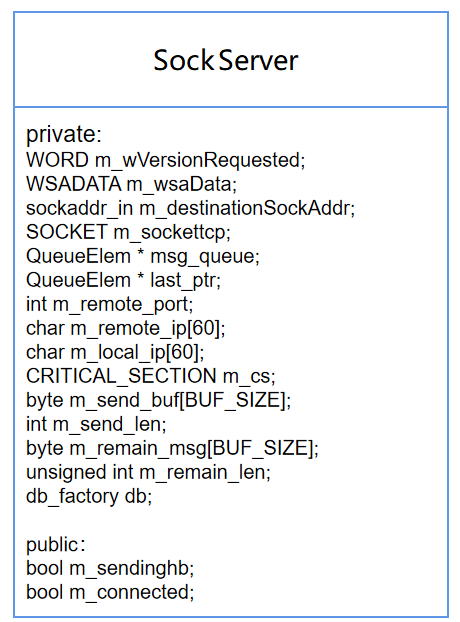
因FBG传感器服务器作为C/S架构的客户端，故FBG传感器数据收集程序无须考虑重连问题。

### 3.1.6 与mysql数据库的连接、存取

采用ODBC连接方式，故开发、测试、运行系统上需先安装mysql ODBC驱动，并在系统的控制面板中配置好数据源。关于数据库存取类的设计，请见2.4.7

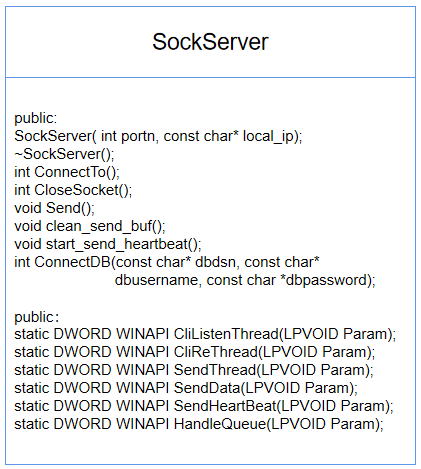
## Socket服务器端类的设计

### 3.2.1 Socket服务器端类成员变量的设计



* WORD m\_wVersionRequested： socket版本信息
* WSADATA m\_wsaData： socket内部数据结构
* sockaddr\_in m\_destinationSockAddr： socket地址数据结构
* SOCKET m\_sockettcp: Socket变量
* QueueElem \* msg\_queue: 解析后的消息队列
* QueueElem \* last\_ptr：消息队列最后一个消息的指针变量
* int m\_remote\_port: socket服务器端的连接端口
* char m\_remote\_ip[60]: socket服务器端的IP
* CRITICAL\_SECTION m\_cs: 发送接受buffer操作临界区变量
* byte m\_send\_buf[BUF\_SIZE]: 发送消息buffer
* int m\_send\_len： 发送消息长度
* byte m\_remain\_msg[BUF\_SIZE]: 剩余未解析消息buffer
* unsigned int m\_remain\_len： 剩余未解析消息长度
* db\_factory db: 数据库实例对象
* bool m\_sendinghb: Socket是否处于发送状态
* bool m\_connected： Socket是否处于连接状态

### 3.2.2 Socket客户端类成员函数的设计



* 构造函数:

SockServer( int portn, const char\* local\_ip);

输入：

int portn: socket服务器端口

char\* local\_ip: 本地IP

输出:

无

* 析构函数

~SockClient();

输入：

无

输出:

无

* 创建服务器，创建线程监听端口

int ConnectTo();

输入：

无

输出:

int型变量

0：创建成功

-1: 创建失败

* 关闭Socket连接

int CloseSocket();

输入：

无

输出:

int型变量

0：关闭成功

-1: 关闭失败

* 将发送buffer中的数据发送到服务器

void Send();

输入：

无

输出:

无

* 将发送buffer清空

void clean\_send\_buf();

输入：

无

输出:

无

* 发送心跳消息

void start\_send\_heartbeat();

输入：

无

输出:

无

* 连接数据库

int ConnectDB(const char\* dbdsn, const char\* dbusername, const char \*dbpassword);

输入：

char\* dbdsn： ODBC数据库连接DSN

char\* dbusername：数据库用户名

char \*dbpassword: 数据库密码

输出:

int型变量:

0: 连接成功

-1： 连接失败

* 发送消息线程函数：

static DWORD WINAPI SendThread(LPVOID Param);

输入:

LPVOID Param: 创建线程时， 需将SocketClient对象自身作为参数传递给线程

输出：

将0返回给系统。

* 发送消息函数， 供发送消息线程调用

static DWORD WINAPI SendData(LPVOID Param);

输入:

LPVOID Param: SocketClient对象指针

输出：

0：发送成功

-1：发送失败

* 发送心跳消息线程函数

static DWORD WINAPI SendHeartBeat(LPVOID Param);

输入:

LPVOID Param: 创建线程时， 需将SocketClient对象自身作为参数传递给线程

输出：

将0返回给系统。

* 处理消息队列线程函数

static DWORD WINAPI HandleQueue(LPVOID Param);

输入:

LPVOID Param: 创建线程时， 需将SocketClient对象自身作为参数传递给线程

输出：

将0返回给系统。

## 监测管理及展示web应用平台的功能设计

设计思想：

* 为保证系统安全，只有登录后才可查看敏感数据。
* 用户分为一般用户和管理员两种用户， 只有管理员才能创建新用户。
* 用户可输入查询条件对历史数据进行查询。
* 系统各页面应有统一的框架和风格，便于操作使用。

### 3.3.1功能模块设计

见图7。系统由账号管理、FBG数据管理和大屏三大部分组成。

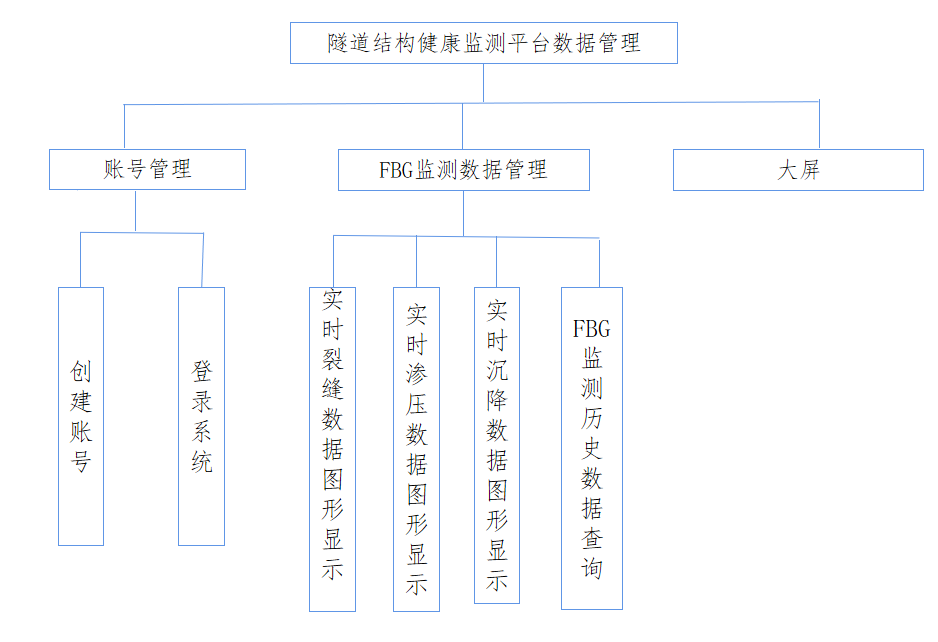


图7 数据管理web应用功能模块

### 3.3.2 隧道结构健康监测平台数据管理功能设计

3.3.2.1 监测平台数据管理主页面设计

监测平台数据管理主界面也就是FBG传感器实时实时数据图形及管理页面。

* UI 设计 （图8）

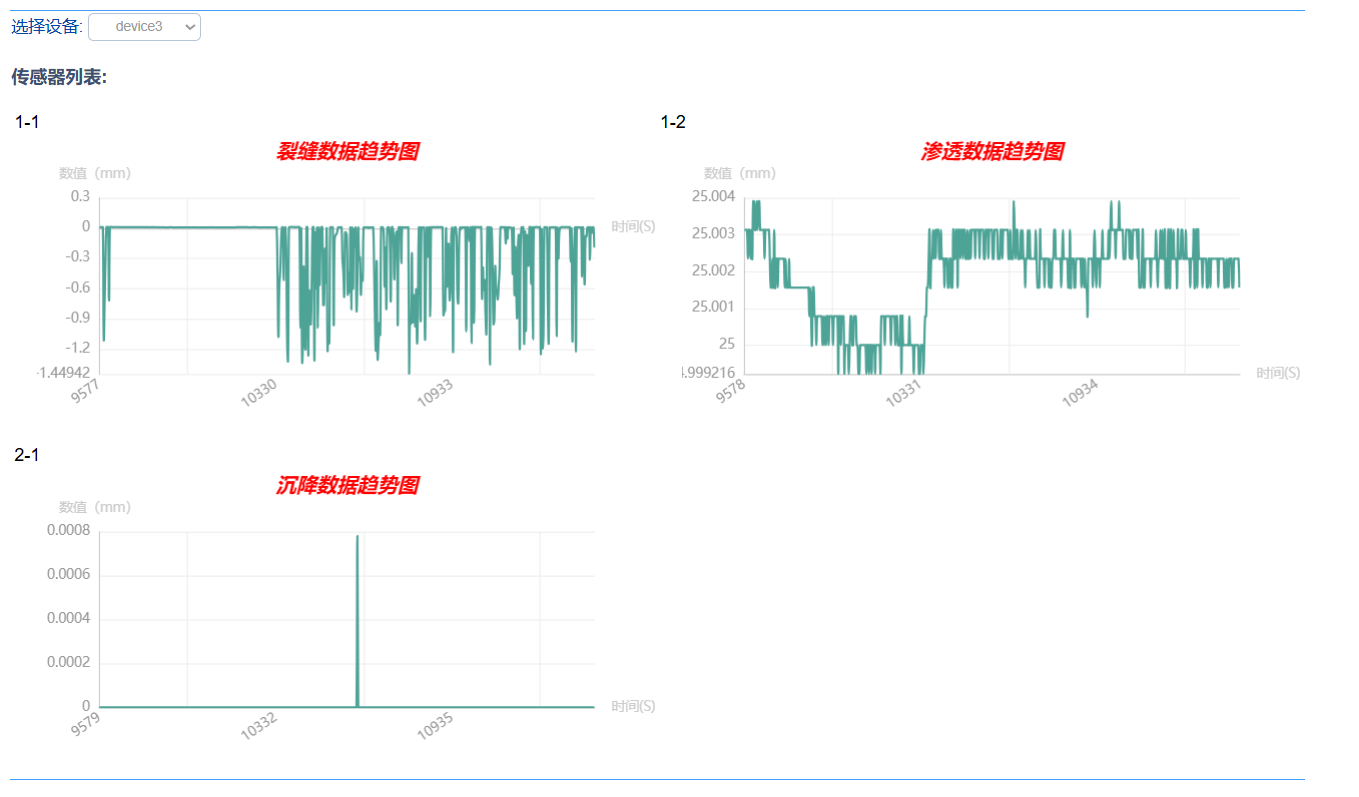


图8 监测平台数据管理主界面

* 功能描述

该页面实时显示FBG传感器系统传来的信号图形，当用户选择传感器设备后，该传感器设备的各传感器信号图形将会显示出来，信号图形实时变化。

由于本页面显示的是实时数据，故优先显示最近的数据，以达到实时显示的效果，该页面分成“选择设备”下拉框和传感器信号图形区两部分。

* “选择设备”下拉框 （图9）

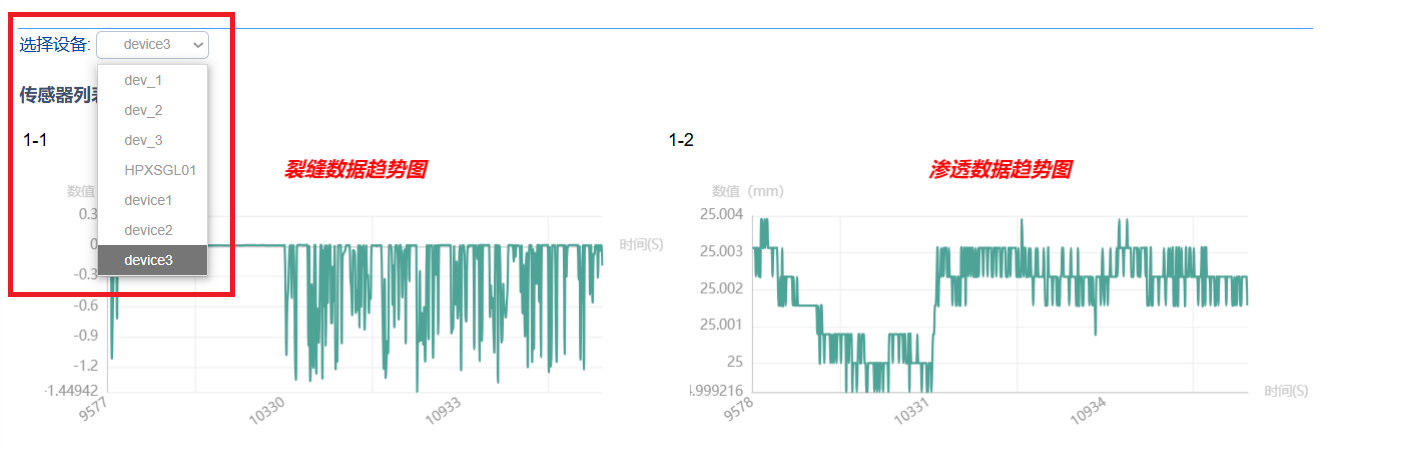


图9 选择设备下拉框

输入输出接口：

|  |  |
| --- | --- |
| **输入操作** | **输出效果** |
| 用户从“选择设备”下拉框中选择某一设备。 | 该设备名称将显示在下拉框的文本框内，该设备所采集到的各传感器数据将以图形的形式展示在图形展示区。 |

* FBG传感器信号图形实时展示区（图10）

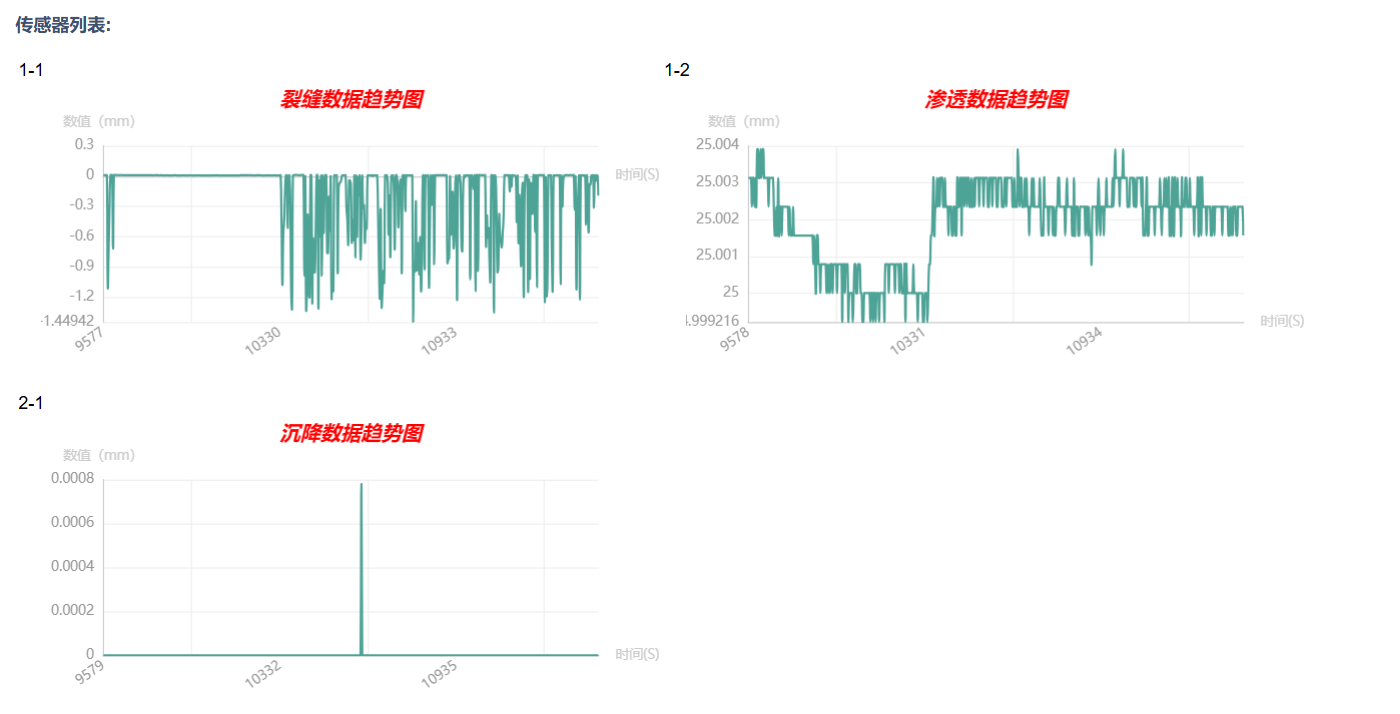


图10 传感器信号图形展示区

展示条件：用户选择了某设备，该设备的所有传感器信号趋势图形将实时展示出来

输入输出接口：

|  |  |
| --- | --- |
| **输入操作** | **输出效果** |
| 用户从“选择设备”下拉框中选择某一设备。 | 该设备所采集到的各传感器数据将以图形的形式展示在图形展示区。  每行只能展示两个传感器图形，如该设备的传感器数量超过两个，超出的传感器图形将显示到下一行。  用户将鼠标放置于任一图形上的某点，该点的坐标将浮现出来（图11）。 |

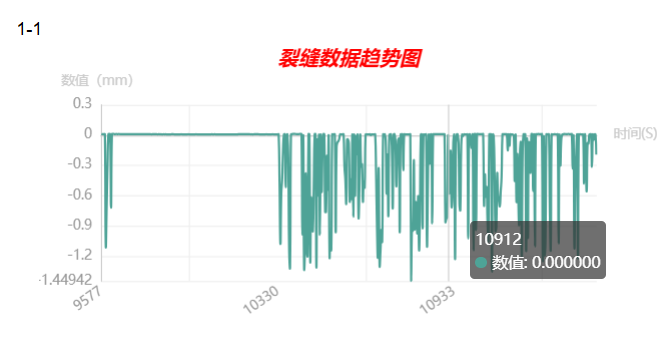


图11 坐标浮现

3.3.2.2 FBG传感器历史数据查询页面

* UI设计 （图12）



图12 FBG传感器信号历史数据查询页面默认界面

没有满足条件的数据时的界面。（图13）



图13 没有满足查询条件时的FBG传感器信号历史数据的界面

查询结果不为0时的界面（图14）



图14 有满足查询条件的FBG传感器信号历史数据时的界面

查询结果超过5000条时，通过分页导航条访问页码高于500的页面时的界面（图15）



图15 访问页码高于500的页面时的界面

* 功能描述

该页面供用户输入查询条件，组合查询’hk\_fbg\_his\_gap\_info’裂隙历史数据表、’hk\_fbg\_his\_shenya\_info’渗压历史数据表、’hk\_fbg\_his\_chenjiang\_info’沉降历史数据表、’hk\_fbg\_real\_data\_info’ FBG实时数据库中数据。各查询条件之间为“与”的关系，因此，如果保持默认查询条件，点击“查询”按钮，将会显示所有记录。查询结果按事件倒序排列，即最近的数据排在前面。

刚进入页面，用户未输入查询条件时，默认显示最近5000条记录。结果按时间逆向排序，最近的记录排在前面。

如查询结果大于十条，每页最多显示十条，通过分页导航条访问其他页面。

如果查询结果大于5000条，前500个页面通过分页导航条可以正常访问，当访问页面的页码大于500时，在查询结果区显示“共查到超过5000条记录，请选择查询条件进行过滤”。

* 输入输出接口
* 查询条件的输入输出接口

输入操作: 用户输入查询条件，点击查询按钮

输出效果: 按照用户输入的查询条件以表格的形式输出FBG传感器数据的历史数据，并显

示总的符合条件的记录数。

查询结果每页显示至多十条数据，如果数据多于十条，可以点击分页页码导航条中的“上一页”、“下一页”、“首页”、“尾页”链接翻页，也可在页码输入框内输入页码，然后点击“Go”按钮跳转到相应页面。

* 查询结果的输入输出接口:

|  |  |
| --- | --- |
| **输入操作** | **输出效果** |
| “首页”按钮 | 如果当前页不是第一页，点击“首页”后，当前页面显示第一页的十条记录。如果当前页面是第一页，点击该按钮页面不改变 |
| “尾页”按钮 | 如果当前页面不是第一页，且查询结果总数不超过5000条时，点击“尾页”，页面跳转到最后一页；如果当前页面是最后一页，点击该按钮页面不改变；如果查询结果超过5000条，点击该按钮，将在结果显示区显示红色的报警文字“共查到超过5000条记录，请选择查询条件进行过滤！” |
| “上一页”按钮 | 如果当前页不是第一页，点击该按钮将显示前一页的查询结果数据；如果当前页是第一页，点击该按钮，页面不改变。 |
| “下一页”按钮 | 如果当前页不是最后一页，且当前页面数加1后不超过500时，点击该按钮，页面跳转到下一页；如果当前页不是最后一页，但当前页面数加1后超过500时，点击该按钮，结果显示区显示红色的报警文字“共查到超过5000条记录，请选择查询条件进行过滤！”；如果当前页是最后一页，点击该按钮，页面不改变。 |
| “Go”按钮 | 在页面输入框中输入页码，如果页码是1~最大页码之间的任意一个页面，且不大于500，点Go按钮后，将跳转到该页; 如查询结果大于5000，且输入框中输入的页码大于500,结果显示区显示红色的报警文字“共查到超过5000条记录，请选择查询条件进行过滤！”;  如在页码输入框中输入的是非法字符，或输入的数字超出了范围，点击Go按钮，页面不改变。 |

* 查询条件的设计

|  |  |
| --- | --- |
| **查询条件** | **说 明** |
| 起始日期 | 接受消息的起始日期，点击输入框后，将弹出日期选择框，从日期选择框中选择日期（图16）。 |
| 结束日期 | 接受消息的结束日期，点击输入框后，将弹出日期选择框，从日期选择框中选择日期（图16）。 |
| 设备名 | FBG监测设备名称或设备号。 |
| 传感器号 | FBG监测传感器名称或传感器号。 |

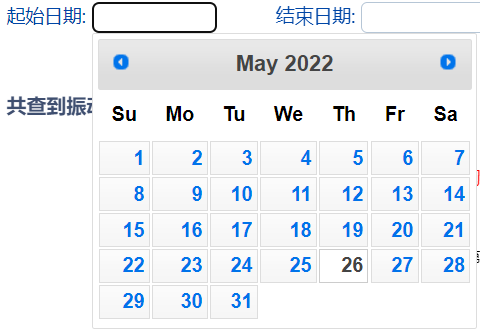


图16 日期选择框

* 查询逻辑的程序流程图 （图17）

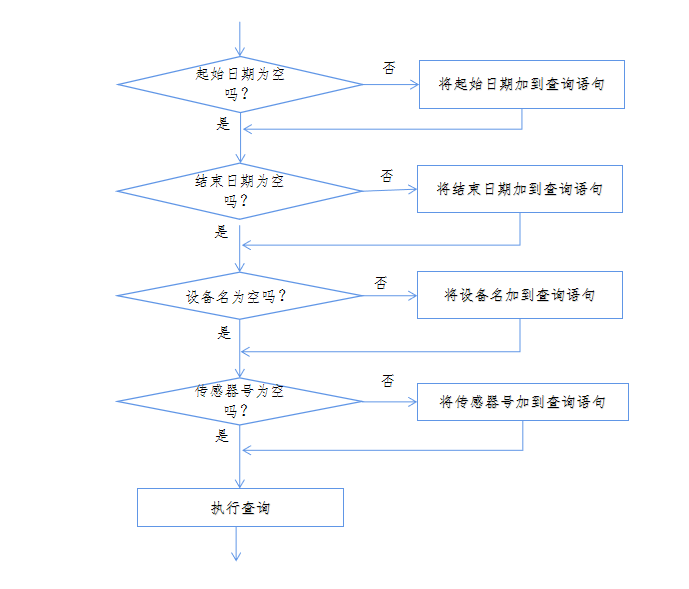


图17 查询逻辑程序流程图

### 3.3.3 账号管理的设计

账号管理主要由登录页面和创建账号页面组成。用户必须先登录系统才能访问各功能页面。在Session超时之后，用户也需重新登录，方可继续浏览各功能页面。

3.3.3.1 登录页面

* UI设计
* 默认登录页面（图18）：



图18 默认登录界面

* 用户名为空，点“登录”按钮后的界面 （图19）



图19 用户名为空时的界面

* 密码为空，点“登录”按钮后的界面 （图20）



图20 密码为空时的界面

* 输入的用户名或密码有误，点“确定” 按钮后的界面 （图21）



图21 输入的用户名或密码有误的界面

* 功能描述

该页面供用户输入用户名、密码登录系统。 在用户登录前或Session超时，访问任意一个功能网页的URL，都将跳转到该页面。

用户点登录后，程序将用户输入的用户名和密码与数据库hk\_user里hk\_user\_info表中的记录进行比对，如与某条记录一致，则允许登录，跳转到系统首页； 否则显示错误信息（图21），提示用户重新输入。

* 输入输出接口

|  |  |
| --- | --- |
| **输入操作** | **输出效果** |
| 用户未输入用户名，点“登录”按钮 | 显示“用户名不能为空”（图19）。 |
| 用户未输入密码，点“登录”按钮 | 显示“密码不能为空”（图20）。 |
| 用户输入的用户名或密码有误 | 显示“您输入的用户名或密码有误”（图21）。 |
| 用户输入正确的用户名和密码，点“登录”按钮。 | 页面跳转到系统首页 |

* 登录页面的处理流程图 （图22）

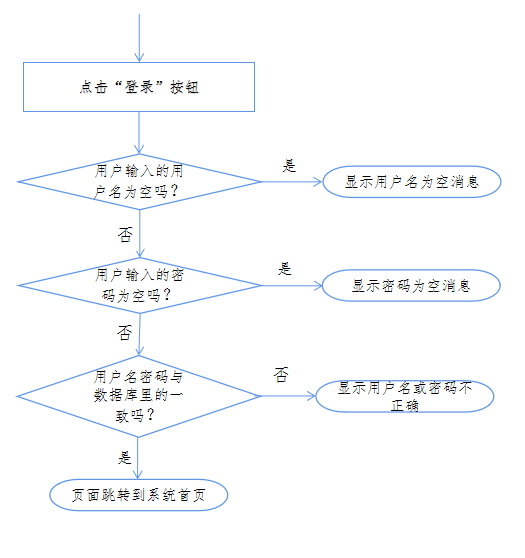


图22 系统登录页面程序流程图

3.3.3.2 创建新用户页面

* UI设计
* 创建新用户页面（图23）：



图23 进入创建用户页面

* 默认创建新用户页面（图24）：



图24 默认创建新用户页面

* 未输入任何用户信息，点“创建新用户”按钮（图25）



图25 未输入用户名

* 未选择用户类型，点“创建新用户”按钮后的界面 （图26）



图26 未选择用户类型创建新用户界面

* 密码和确认密码不一致，点“创建新用户”按钮后的界面 （图27）



图27 创建新用户密码不一致界面

* 用户名在数据库中已存在，点“创建新用户”按钮后的界面 （图28）

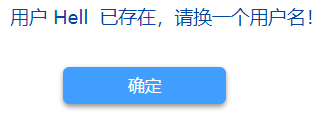


图28 用户名在数据库中已存在，创建新用户界面

* 成功创建新用户界面 （图29）



图29 成功创建新用户界面

* 功能描述

* 只有管理员用户才能访问该页面， 管理员用户是指hk\_user\_info表中user\_role为0的用户，一般用户该字段为1。
* 新添加的用户名不得与数据库中已存在的用户名重名。
* 创建新用户时，必须将新用户选定为管理员或一般用户。
* 输入输出接口

|  |  |
| --- | --- |
| **输入操作** | **输出效果** |
| 用户未输入用户名，点“创建新用户”按钮 | 显示“请填写此字段”（图25）。 |
| 用户输入的密码和确认密码不一致，点“创建新用户”按钮 | 显示“密码和确认密码不一致”（图27）。 |
| 未选择用户类型，点“创建新用户”按钮 | 显示“请选择用户类型”（图26）。 |
| 输入的用户名数据库中已存在，点“创建新用户”按钮 | 显示“此用户已存在，请换一个用户名”（图28） |
| 输入全部合法，且不与数据库中现有用户记录重名，点“创建新用户”按钮 | 成功将新用户添加到数据库，页面显示“添加新用户XXX成功！还要创建一个新用户吗？”（图29）， 如点击“是”按钮，页面返回到“创建新用户”页面， 如点“否”，页面跳转到首页。 |

* 创建新用户的处理流程图 （图30）

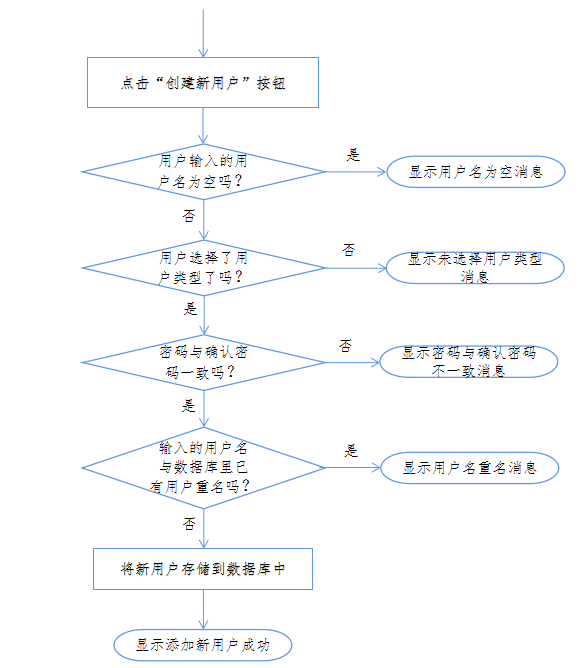


图30 创建新用户流程图

### 3.3.4 大屏展示设计

* UI设计

大屏展示页由数据展示区和图表区两部分组成（图31）。 点击左导航条上的“大屏展示”链接进入大屏展示页。点击页面下方的“全屏显示”按钮，页面进入全屏显示状态，同时“全屏显示”按钮消失。



图31 大屏展示页面

* “全屏显示”按钮（图32）

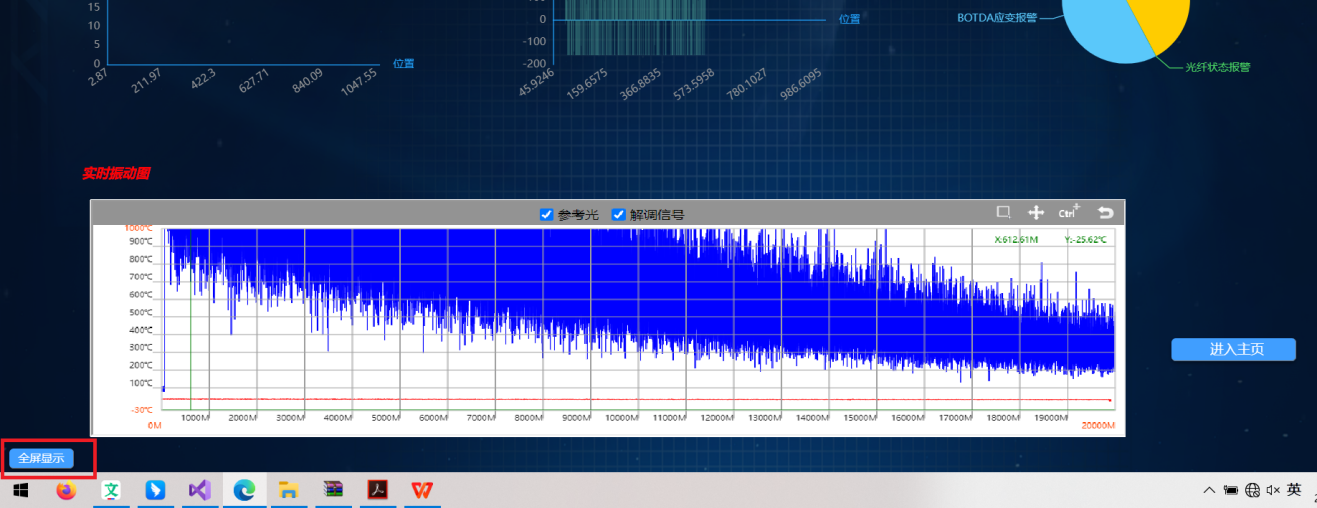


图32“全屏显示”按钮

功能: 点击“全屏显示”按钮，大屏页面将进入全屏模式。

* “进入主页”按钮（图33）

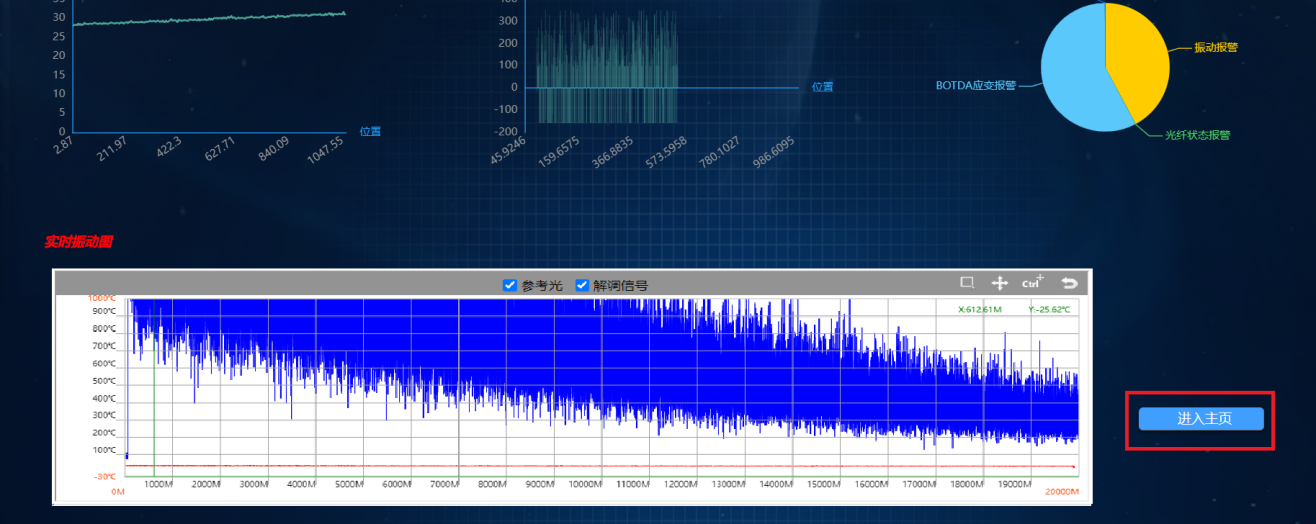


图33 “进入主页”按钮

功能： 点击“进入主页”按钮，页面将进入系统主页。

### 3.3.5 页面框架设计

页面框架由网页标题栏、 左导航栏、 内容显示区三部分构成。（图34）

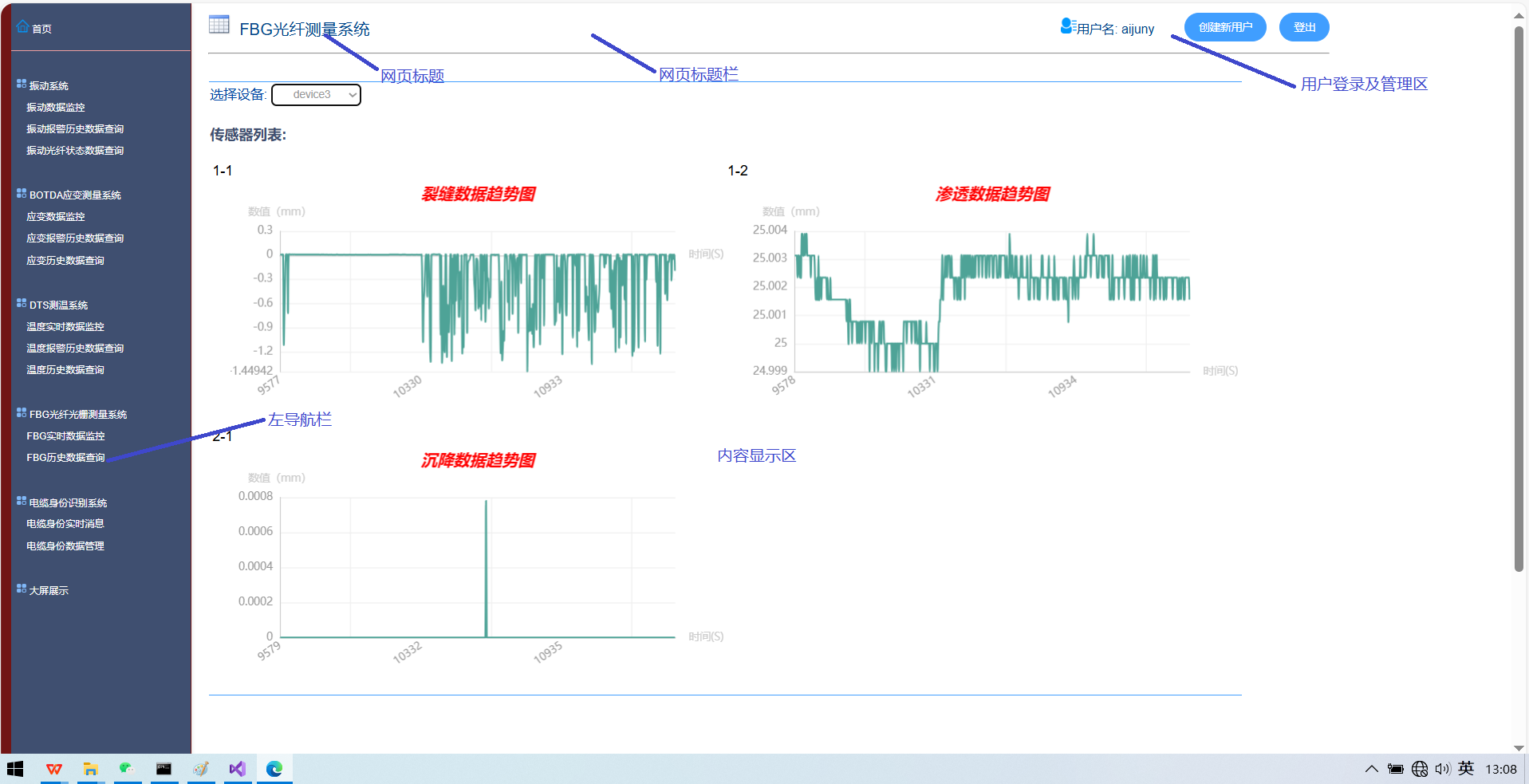


图34 页面框架

3.3.5.1 网页标题栏的设计

* UI设计

网页标题栏由标题栏、登录用户名及登陆管理区两部分组成。（图35）



图35 网页标题栏

主要设计参数:

背景颜色 background-color: #FFFFFF;

高度 height: 70px;

最小宽度 min-width: 1520px;

字体font-family: 华文楷体

* 功能描述：

网页标题: 显示当前内容区的网页的标题

用户名: 显示当前登陆用户的用户名

“创建新用户”按钮:

只有超级用户登陆才显示该按钮，一般用户登陆，不显示该按钮。

点击该按钮，内容区页面将跳转到创建新用户页面（参见图23）。

“登出”按钮： 点击该按钮，将退出当前登陆，页面跳转到登陆界面（参见图35）。

3.3.5.2 左导航条的设计

* 左导航条UI（图36）

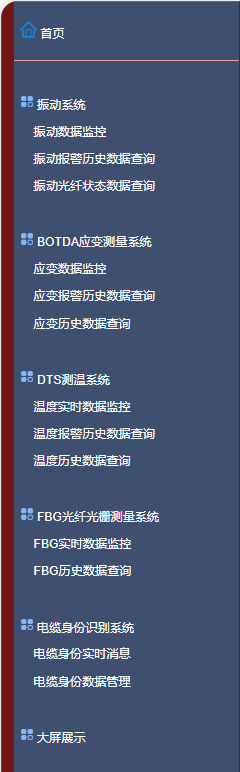


图36 左导航条UI

* 主要设计参数:

背景颜色：background-color: #3F4F70;

最小高度 min-height: 100%;

宽度width: 200px;

高度height: 1100px;

字符颜色color: #FFFFFF

左边框颜色:#711515

* 导航链接URL:

|  |  |
| --- | --- |
| **链 接** | **超链接URL** |
| 首 页 | /Home/Index |
| 振动系统 | /Vib/Index |
| 振动数据监控 | /Vib/Index |
| 振动报警历史数据查询 | /Vib/Vibhis |
| 振动光纤状态数据查询 | /Vib/Fibhis |
| BOTDA应变测量系统 | /Bot/Index |
| 应变数据监控 | /Bot/Index |
| 应变报警历史数据查询 | /Bot/Bothis |
| 应变历史数据查询 | /Bot/Botdata |
| DTS测温系统 | /Dts/Index |
| 温度实时数据监控 | /Dts/Index |
| 温度报警历史数据查询 | /Dts/Dtshis |
| 温度历史数据查询 | /Dts/Dtsdata |
| FBG光纤光栅测量系统 | /FBG/index |
| FBG实时数据监控 | /FBG/index |
| FBG历史数据查询 | /FBG/Fbghis |
| 电缆身份识别系统 | /Cab/Index |
| 电缆身份实时消息 | /Cab/Index |
| 电缆身份数据管理 | /Cab/Cabadmin |
| 大屏展示 | /Home/Bigscreen |

### 3.3.6 程序运行设置

程序部署前，需先修改Web.config进行设置。

3.3.6.1 系统登陆页面的设置

<system.web>

<compilation debug="true" targetFramework="4.5.2" />

<httpRuntime targetFramework="4.5.2" />

<globalization fileEncoding="utf-8" />

<authentication mode="Forms">

<forms name="AuthCookie" loginUrl="login.aspx" protection="All" />

</authentication>

<authorization>

<deny users="?" />

</authorization>

</system.web>

<location path="default.aspx">

<system.web>

<authorization>

<allow users="\*"/>

</authorization>

</system.web>

</location>

3.3.6.2 数据库的设置

<appSettings>

<add key="webpages:Version" value="3.0.0.0" />

<add key="webpages:Enabled" value="false" />

<add key="ClientValidationEnabled" value="true" />

<add key="UnobtrusiveJavaScriptEnabled" value="true" />

<add key="DB\_IP" value="127.0.0.1" />

<add key="DB\_USER\_DSN" value="hk\_ur" />

<add key="DB\_FBG\_DSN" value="hk\_fb" />

<add key="DB\_USER" value="root" />

<add key="DB\_PW" value="sasasa" />

<add key="DB\_USER\_NAME" value="hk\_user" />

<add key="DB\_FBG\_NAME" value="hk\_fbg" />

<add key="ChartImageHandler" value="storage=file;timeout=20;dir=c:\TempImageFiles\;" />

</appSettings>

参数说明:

|  |  |
| --- | --- |
| **参 数** | **说 明** |
| DB\_IP | 数据库服务器的IP地址 |
| DB\_USER\_DSN | 用户数据库ODBC连接DSN |
| DB\_FBG\_DSN | FBG数据库ODBC连接DSN |
| DB\_USER | 数据库用户名 |
| DB\_PW | 数据库密码 |
| DB\_USER\_NAME | 用户数据库的库名称 |
| DB\_FBG\_NAME | FBG数据库的库名称 |