**水质分析仪器应用系统软件概要**

**设计说明书**



编制：

校对：

审核：

审定：

批准：

日期： 年 月 日

**西安思坦科技有限公司**

**水质分析仪器应用系统软件概要**

**设计说明书**



**西安思坦科技有限公司**

目录

[1. 项目简介 1](#_Toc492481329)

[2. 系统架构概述 1](#_Toc492481330)

[3. 逻辑架构设计 3](#_Toc492481331)

[3.1. 基于框架的设计 3](#_Toc492481332)

[3.2. 通用业务逻辑的抽象（框架设计） 3](#_Toc492481333)

[3.3. 界面架构 4](#_Toc492481334)

[3.4. 全局唯一性与不同模块对象间的引用（系统设计） 5](#_Toc492481335)

[3.5. 系统逻辑架构 7](#_Toc492481336)

[4. 关键技术 7](#_Toc492481337)

[4.1. 开发环境与开发语言 7](#_Toc492481338)

[4.2. 界面开发技术 8](#_Toc492481339)

[4.3. 数据持久化 8](#_Toc492481340)

[5. 模块设计 9](#_Toc492481341)

[5.1. 框架设计 9](#_Toc492481342)

[5.1.1. 数据管理接口 9](#_Toc492481343)

[5.1.2. 观察者模式接口 9](#_Toc492481344)

[5.1.3. 通信接口 10](#_Toc492481345)

[5.1.4. 其他接口 10](#_Toc492481346)

[5.2. 界面设计 10](#_Toc492481347)

[5.2.1. 顶部模块 10](#_Toc492481348)

[5.2.2. 主内容模块 11](#_Toc492481349)

[5.2.3. 底部模块 11](#_Toc492481350)

[5.3. 系统设计 12](#_Toc492481351)

[5.3.1. 视图管理模块 12](#_Toc492481352)

[5.3.2. 流程控制模块 13](#_Toc492481353)

[5.3.3. 上下文模块（依赖注入） 13](#_Toc492481354)

[5.3.4. 数据模型模块 14](#_Toc492481355)

[5.3.5. 协议模块 14](#_Toc492481356)

[5.3.6. 数据管理模块 15](#_Toc492481357)

[5.3.7. 其他模块 15](#_Toc492481358)

[6. 错误处理 15](#_Toc492481359)

# 项目简介

水质分析仪器是专门用于对水体中所含某种物质成分进行定量分析的一类仪器，主要的仪器种类有COD（化学需氧量）分析仪，氨氮分析仪，总磷分析仪，总氮分析仪等。随着国家对环保领域法律法规的完善确立以及环保产业规模的日益扩大，水质分析类仪器的市场需求也必将逐步扩大。

伴随着计算机技术的发展，软件系统在各种分析类仪器中的重要性日益突出，软件系统不仅作为人机交互的接口，同时也完成了对原始数据的存储分析等功能。为了配合公司水质分析类仪器的研发工作，需要开发与之配套的应用系统软件。

# 系统架构概述

结合系统的功能性需求与非功能性需求，系统整体架构采用分层的设计模式，自上而下划分为交互层、业务逻辑层、数据/通信层。

交互层提供了图形化的用户交互接口，提供了对用户输入数据的收集和下位机返回数据的显示；业务逻辑层对业务逻辑进行封装并向上层提供粗粒度功能接口；通信模块提供了基于具体通信方式的粗粒度接口；数据存储模块提供了数据的持久化与逆持久化相关功能。



# 逻辑架构设计

## 基于框架的设计

框架（Framework）是整个或部分系统的可重用设计，表现为一组抽象构件及构件实例间交互的方法。

一个框架是一个可复用的设计构件，它规定了应用的体系结构，阐明了整个设计、协作构件之间的依赖关系、责任分配和控制流程，表现为一组抽象类以及其实例之间协作的方法，它为构件复用提供了上下文(Context)关系。因此构件库的大规模重用也需要框架。

|  |  |  |  |
| --- | --- | --- | --- |
|  | 功能A | 功能B | 功能C |
| 交互界面 | 子界面A | 子界面B | 子界面C |
| 流程控制 | 子流程控制A | 子流程控制B | 子流程控制C |
| 数据模型 | 数据模型A | 数据模型B | 数据模型C |
| 通信协议 | A功能相关协议 | B功能相关协议 | C功能相关协议 |
| 通信接口 | 公共接口 | 公共接口 | 公共接口 |

## 通用业务逻辑的抽象（框架设计）

根据需求分析说明中对核心业务流程的分析结果，系统框架体系结构如图所示。其中，视图管理接口定义了界面与后台交互的接口，从而实现特定技术开发的前端界面与后端实现之间的解耦和；数据模型接口对相关联的独立数据进行集成封装并提供一致性访问接口；流程控制接口定义了应用功能的方法接口；协议接口定义了协议相关的抽象接口；通信接口定义了通信相关的抽象接口；数据管理接口定义了本地数据管理的抽象接口。

为了解除视图管理接口与流程控制接口之间的循环以来，引入了观察者模式用来解除两个模块之间的循环以来。



## 界面架构



## 全局唯一性与不同模块对象间的引用（系统设计）

为了保证业务逻辑实现的独立性与代码的简洁清晰，实例化后特定功能相关的视图管理对象，数据模型对象，流程控制对象需要保证全局唯一性。即特定的实例化对象在系统运行过程中保持唯一性。

同时，根据业务需求，唯一性实例化对象可能会与多个对象之间形成关联，如图所示：







控制反转即IoC (Inversion of Control)，它把传统上由程序代码直接操控的对象的调用权交给容器，通过容器来实现对象组件的装配和管理。所谓的“控制反转”概念就是对组件对象控制权的转移，从程序代码本身转移到了外部容器。



## 系统逻辑架构



# 关键技术

## 开发环境与开发语言

采用Qt作为基础开发框架主要由以下几个优势：

1）优良的跨平台特性。

2）面向对象。

3）丰富的 API。

C++语言作为Qt框架的原生语言，能够最大限度的发挥出Qt框架的优势，同时C++语言运行时的高效性也保障了应用系统在配置较低的工控机上的流畅运行。

## 界面开发技术

QML是Qt推出的Qt Quick技术的一部分，是一种新增的简便易学的语言。QML是一种陈述性语言，用来描述一个程序的用户界面：无论是什么样子，以及它如何表现。

## 数据持久化

选择SQLite数据库主要基于以下几点原因：

1）使用数据库系统便于数据的存储、读取与查询；

2）SQLite满足系统的数据规模需求；

3）SQLite的特性更适合运行在配置较低的工控机环境下；

4）Qt框架对SQLite数据库实现了集成并提供了友好的编程接口。

# 模块设计

## 框架设计

### 数据管理接口



### 观察者模式接口



### 通信接口



### 其他接口

其他接口由单一接口类或基类构成，主要有CRC校验类，数据模型基类，控制器基类。

## 界面设计

### 顶部模块



### 主内容模块



### 底部模块



## 系统设计

### 视图管理模块



### 流程控制模块



### 上下文模块（依赖注入）



### 数据模型模块



### 协议模块



### 数据管理模块



### 其他模块

数据校验模块，模型协议转换模块。

# 错误处理

|  |  |
| --- | --- |
| **错误类型** | **处理方式** |
| 通信异常 | 重发/抛出 |
| 内部逻辑错误 | 界面提示 |
| 输入错误 | 验证并提示 |