密级：秘密

文档编号：D0000-PPC-001100-PPD-2023

项目名称：智能家居管家

项目编号：001100

概要设计模板

版本：0.8.0-0.0.0

2022-08-04

东软集团股份有限公司 初级人才培养中心

(版权所有，翻版必究)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 总页数 | 18 | 正文 | 14 | 附录 | 0 | 生效日期 |  |
| 编制 | 刘林娜 | | | 批准 |  | | |

文件修改控制

|  |  |  |  |
| --- | --- | --- | --- |
| **修改编号** | **版本** | **修改条款及内容** | **修改日期** |
| 1 | 0.8.0-0.0.0 | 创建 | 2023-8-3 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**目 录**

[1 文档概述 5](#_Toc1595)

[1.1 文档目的和范围 5](#_Toc32088)

[1.2 术语/缩略语 5](#_Toc29848)

[2 系统结构图 6](#_Toc11397)

[3 模块详细概述 6](#_Toc20838)

[3.1 客户端模块 7](#_Toc19389)

[3.1.1 客户端模块功能定义 7](#_Toc29941)

[3.1.2 客户端模块结构 7](#_Toc30467)

[3.1.3 客户端模块类图 8](#_Toc25710)

[3.2 服务器模块 9](#_Toc15194)

[3.2.1 服务器模块功能定义 10](#_Toc11065)

[3.2.2 服务器模块结构 10](#_Toc18210)

[3.2.3 服务器模块类图 10](#_Toc5257)

[3.3 网络通信模块 11](#_Toc24034)

[3.3.1 网络通信模块功能定义 11](#_Toc28251)

[3.3.2 网络通信模块结构 12](#_Toc13268)

[3.3.3 网络通信模块类图 12](#_Toc2416)

[4 数据库设计 13](#_Toc13942)

[4.1 数据库引擎概述 13](#_Toc1542)

[4.2 数据库概要设计 13](#_Toc13026)

[5 模块接口设计 14](#_Toc15192)

[5.1 全局变量 14](#_Toc4039)

[5.2 模块间接口函数设计 14](#_Toc14388)

[5.2.1 客户端模块 14](#_Toc16432)

[5.2.2 服务器模块 17](#_Toc3364)

# 文档概述

## 文档目的和范围

本文档是智能家居管家项目的详细设计书，旨在描述项目的系统结构、模块划分、数据库设计、接口设计等方面的详细信息，以便开发人员进行编码、测试和维护。

1. **文档用途**

* 提供系统结构的概述，以便开发者了解系统的组成部分和它们之间的关系；
* 详细描述系统各个模块的功能定义、结构和类图，以帮助开发者理解模块之间的交互方式；
* 提供数据库设计信息，包括数据库引擎选择和数据表结构设计，以指导数据库操作的实施；
* 设计系统的接口，包括全局变量和模块间的接口函数，以确保模块之间的协调和数据传递的准确性。

1. **文档范围**

* 系统结构图：展示系统的组成部分和它们之间的关系 ；
* 模块详细概述：详细描述系统各个模块的功能、结构和类图；
* 数据库设计：选择数据库引擎并描述数据表结构和数据库操作；
* 接口设计：定义全局变量和模块间的接口函数。

本文档适用于项目的开发人员、测试人员和管理人员，以及项目的客户和用户。

## 术语/缩略语

本文中使用的术语和缩略词说明如表1.1所示。

表1.1 本文中术语/缩略语表

|  |  |  |
| --- | --- | --- |
| 序号 | 术语/缩略语 | 说明 |
| 1 | 智能家居 | 利用现代信息技术和网络技术，将各种家居设备有机集成到一起，实现家居设备之间的联动和智能控制 |
| 2 | 服务器 | 运行在Linux服务器上的智能家居管家程序，负责数据接收、处理和分发等功能 |
| 3 | 客户端 | 运行在用户设备上的智能家居管家应用程序，负责用户认证、数据可视化和智能家居控制等功能 |
| 4 | TCP/IP | 一种网络通信的协议，分为四层：应用层、传输层、网络层、数据链路层 |

# 系统结构图

系统的总体结构如图2-1所示，本系统共分为三个大模块：客户端、服务器与网络通信，其中客户端模块包括用户认证、数据可视化、智能家居控制三个小模块，服务器模块包含数据收集与数据分发两个小模块，网络通信分为数据传输与异步通信两个小模块，各个模块分别实现其各自的功能点。

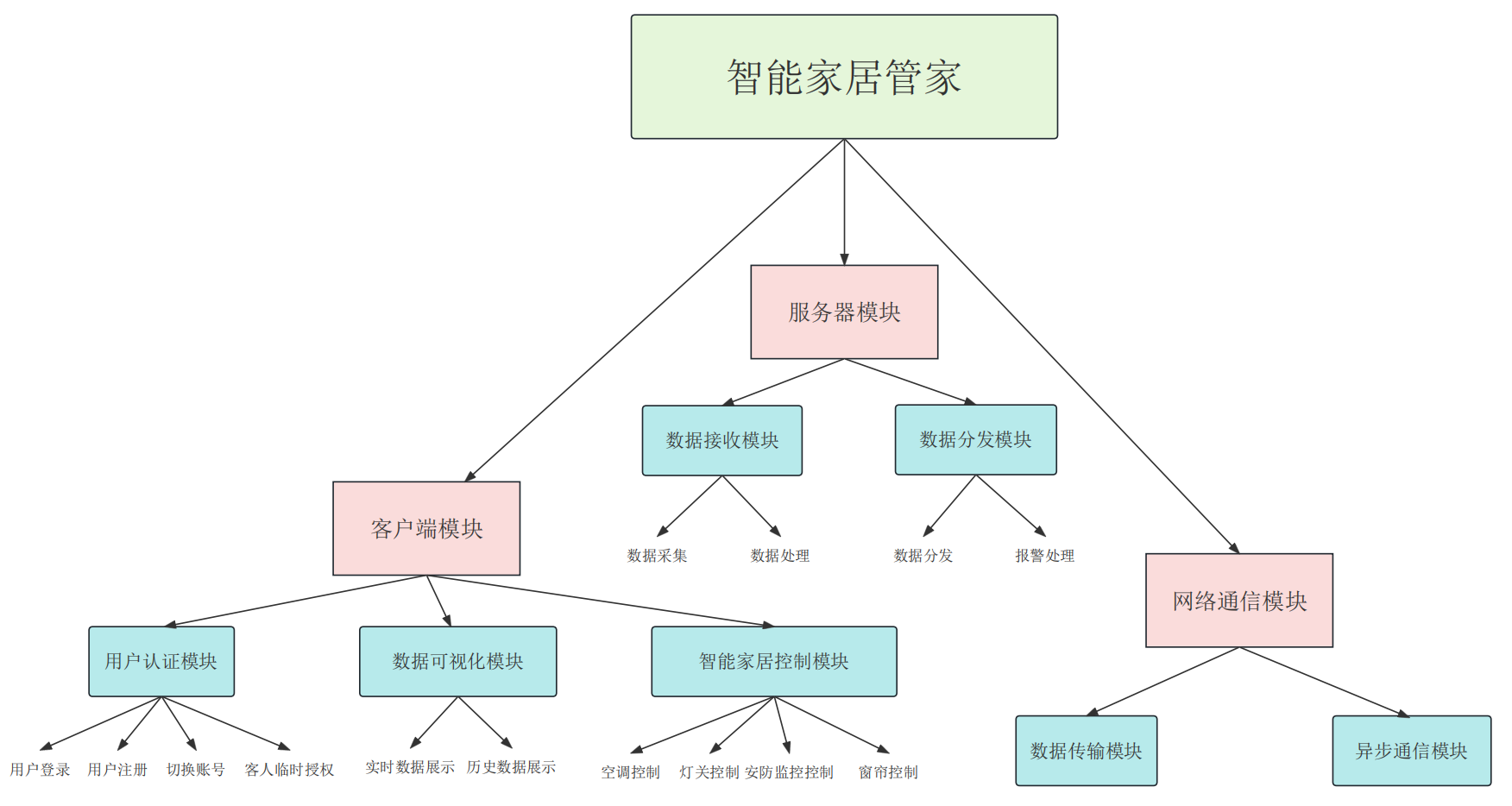


图2-1 系统结构图

# 模块详细概述

该项目中包含了客户端、服务器、网络通信三个主要模块。其中客户端模块负责提供用户界面和交互功能，让用户可以查看和控制智能家居系统；服务器模块负责接收和处理模拟数据，并将处理后的结果发送给客户端；网络通信模块负责处理客户端和服务器之间的通信。这些模块之间相互协调、相互依赖，构成了智能家居系统的完整功能。通过这些模块的协同工作，智能家居系统能够实现数据传输、处理、分析和控制等功能，提高智能化程度和用户体验。

该项目的优点是具有高效、稳定、安全、智能化等特点，能够实现多种设备的智能控制和管理，提高用户的生活品质。不足之处是需要更多的算法优化和设备支持，以提高系统的智能化程度和适用范围。未来的发展方向是在不断优化算法和增加设备支持的基础上，加强系统的可扩展性和智能化程度，实现更加智能化、便捷化和人性化的服务。

以下将分别介绍各个模块：

## 客户端模块

客户端模块是智能家居系统用户界面的核心部分，主要负责用户登录、注册、切换账号、客人临时权限等功能。该模块包括用户认证模块、数据可视化模块和智能家居控制模块。

### **客户端模块功能定义**

用户认证功能让用户可以通过登录、注册、切换账号和设置临时权限等方式进行用户认证，保证系统的安全性和私密性。数据可视化功能让用户可以通过图表形式查看实时或历史的智能家居状态数据，增加系统的可用性和友好性。智能家居控制功能让用户可以通过客户端控制智能家居系统中的灯光、空调、安防监控和窗帘等设备，提高系统的智能性和便捷性。其详细说明如表3-1所示。

表3-1 客户端模块功能定义

|  |  |  |
| --- | --- | --- |
| **序号** | **小模块** | **模块功能点详细说明** |
| 1 | 用户认证 | 用户可以通过登录、注册、切换账号和设置临时权限等方式进行用户认证，输入用户名、密码和邮箱等信息，验证用户身份，保存用户信息到数据库中，为用户分配相应的权限 |
| 2 | 数据可视化 | 用户可以通过图表形式查看实时或历史的智能家居状态数据，从服务器接收数据，展示数据到客户端界面上，支持不同类型的图表，如折线图、柱状图、饼图等，支持不同时间范围的筛选，如日、周、月、年等 |
| 3 | 智能家居控制 | 用户可以通过客户端控制智能家居系统中的灯光、空调、安防监控和窗帘等设备，向服务器发送控制指令，包括开关状态、亮度、温度、湿度、模式和日程安排等参数，从服务器接收控制结果，反馈到客户端界面上 |

### **客户端模块结构**

本模块由用户认证模块、数据可视化模块和智能家居控制模块三个子模块组成。用户认证模块负责处理用户认证相关的逻辑，包括验证用户身份，保存用户信息，分配用户权限等。数据可视化模块负责处理数据可视化相关的逻辑，包括接收数据，展示数据，支持筛选等。智能家居控制模块负责处理智能家居控制相关的逻辑，包括发送控制指令，接收控制结果，反馈结果等。各个子模块之间通过接口函数进行通信。本模块的结构如图3-1所示：

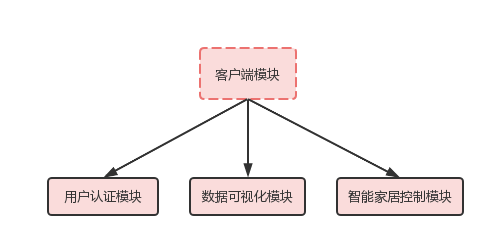


图3-1 客户端模块结构图

### **客户端模块类图**

客户端类图与用户接口说明分别入图3-2和表3-2所示。

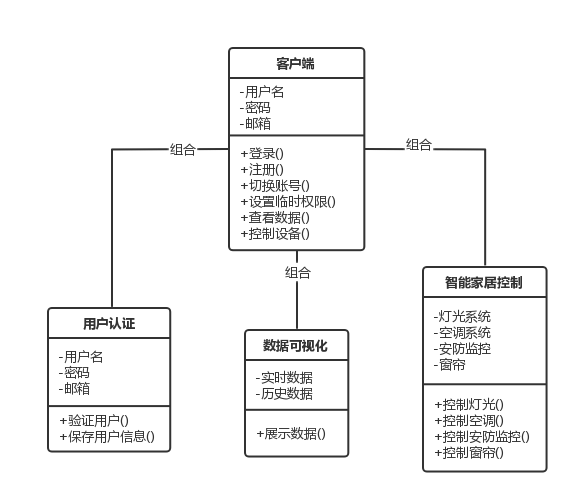
**

图3-2 客户端模块类图

表3-2 客户端模块接口说明

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 模块名称 | 接口函数 | 函数说明 |
| 1 | 用户认证 | bool login(string username, string password) | 用户登录，输入用户名和密码，返回是否登录成功 |
| 2 | 用户认证 | bool register(string username, string password, string email) | 用户注册，输入用户名、密码和邮箱，返回是否注册成功 |
| 3 | 用户认证 | bool switchAccount(string username, string password) | 用户切换账号，输入新的账号和密码，返回是否切换成功 |
| 4 | 用户认证 | bool setTempPermission(string guest, int duration) | 用户设置临时权限，输入客人的用户名和权限持续时间，返回是否设置成功 |
| 5 | 数据可视化 | void showRealTimeData() | 展示实时数据，无输入参数，无返回值 |
| 6 | 数据可视化 | void showHistoryData() | 展示历史数据，无输入参数，无返回值 |
| 7 | 智能家居控制 | void controlLight(bool onOff, int brightness, string mode, string schedule) | 控制灯光，输入开关状态、亮度、模式和日程安排，无返回值 |
| 8 | 智能家居控制 | void controlAirConditioner(bool onOff, int temperature, int humidity, string mode, string schedule) | 控制空调，输入开关状态、温度、湿度、模式和日程安排，无返回值 |
| 9 | 智能家居控制 | void controlSecurity(bool onOff, string mode, string schedule) | 控制安防监控，输入开关状态、模式和日程安排，无返回值 |
| 10 | 智能家居控制 | void controlCurtain(bool onOff, string mode, string schedule) | 控制窗帘，输入开关状态、模式和日程安排，无返回值 |

## 服务器模块

服务器模块是系统的后台部分，负责接收和处理模拟数据，并将处理后的结果发送给客户端。

### 服务器模块功能定义

数据接收功能接收来自传感器或其他来源的模拟数据，并将数据存储到数据库中。数据处理功能对模拟数据进行处理，包括数据分析、特征提取、异常检测等，提高数据的质量和价值。数据分发功能将处理后的数据发送给客户端，满足客户端的需求和期望。其各个模块的详细功能定义如表3-3所示。

表3-3 服务器模块功能定义

|  |  |  |
| --- | --- | --- |
| 序号 | 小模块 | 功能点详细说明 |
| 1 | 数据接收 | 接收来自传感器或其他来源的模拟数据，并将数据存储到数据库中，保证数据的完整性和一致性 |
| 2 | 数据处理 | 对模拟数据进行处理，包括数据分析、特征提取、异常检测等，提高数据的质量和价值，为客户端提供有用的信息 |
| 3 | 数据分发 | 将处理后的数据发送给客户端，满足客户端的需求和期望，为客户端提供良好的服务 |

### 服务器模块结构

本模块由数据接收模块、数据处理模块和数据分发模块三个子模块组成。数据接收模块负责处理数据接收相关的逻辑，包括保存数据到数据库中等。数据处理模块负责处理数据处理相关的逻辑，包括分析数据，提取特征，检测异常等。数据分发模块负责处理数据分发相关的逻辑，包括发送数据给客户端等。各个子模块之间通过接口函数进行通信。本模块的结构如图3-3所示：

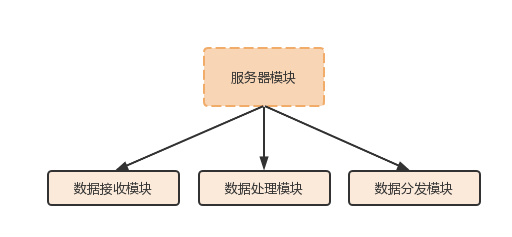


图3-3 服务器模块结构图

### 服务器模块类图

服务器模块类图与接口说明分别如图3-4和表3-4所示。

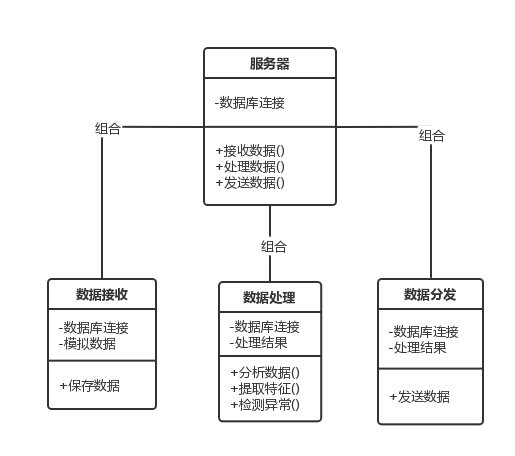
**

图3-4 服务器模块类图

表3-4 服务器模块接口说明

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 模块名称 | 接口函数 | 函数说明 |
| 1 | 数据接收 | void saveData(string data) | 保存数据，输入模拟数据，无返回值 |
| 2 | 数据处理 | void analyzeData() | 分析数据，无输入参数，无返回值 |
| 3 | 数据处理 | void extractFeature() | 提取特征，无输入参数，无返回值 |
| 4 | 数据处理 | void detectAnomaly() | 检测异常，无输入参数，无返回值 |
| 5 | 数据分发 | void sendData(string data) | 发送数据，输入处理结果，无返回值 |

## **网络通信模块**

网络通信模块负责处理客户端和服务器之间的通信。

### 网络通信模块功能定义

数据传输功能负责客户端和服务器之间的数据传输，包括数据的发送和接收。异步通信功能采用异步通信方式，提高数据传输效率和稳定性。其功能定义如表3-5所示

表3-5 网络通信模块功能定义

|  |  |  |
| --- | --- | --- |
| 序号 | 小模块 | 功能点详细说明 |
| 1 | 数据传输 | 负责客户端和服务器之间的数据传输，包括数据的发送和接收，保证数据的准确性和及时性 |
| 2 | 异步通信 | 采用异步通信方式，提高数据传输效率和稳定性，避免阻塞和超时等问题 |

### 网络通信模块结构

数据传输相关的逻辑，包括发送数据和接收数据等。异步通信模块负责处理异步通信相关的逻辑，包括创建线程，使用回调函数等。各个子模块之间通过接口函数进行通信。本模块的结构如图3-5所示：

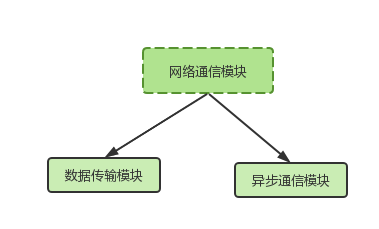


图3-5 网络通信模块结构图

### 网络通信模块类图

网络通信模块类图与接口说明分别如图3-6和表3-6所示。

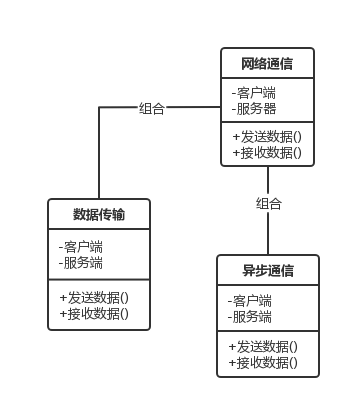


图3-6 网络通信模块类图

表3-6 网络通信模块接口说明

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 模块名称 | 接口函数 | 函数说明 |
| 1 | 数据传输 | void sendData(string data) | 发送数据，输入数据，无返回值 |
| 2 | 数据传输 | string receiveData() | 接收数据，无输入参数，返回数据 |
| 3 | 异步通信 | Void sendDataAsync(string data) | 异步发送数据，输入数据，无返回值 |
| 4 | 异步通信 | string receiveDataAsync() | 异步接收数据，无输入参数，返回数据 |

# 数据库设计

## 数据库引擎概述

数据库引擎概述 本系统使用SQLite作为数据库引擎，因为SQLite是一种轻量级、嵌入式、无需服务器的关系型数据库管理系统，适合用于存储和管理智能家居管家系统的数据。SQLite具有以下优点：

* 简单易用，无需安装和配置；
* 跨平台，支持多种操作系统和编程语言；
* 高效稳定，支持事务和并发控制；
* 自包含，只需要一个单独的文件存储所有的数据。

## 数据库概要设计

本系统的数据库包含的表如表4-1所示。

表4-1 数据库表说明

|  |  |  |
| --- | --- | --- |
| 序号 | 表名 | 表说明 |
| 1 | user | 存储用户信息，包括用户名、密码、邮箱等 |
| 2 | device | 存储设备信息，包括设备编号、设备类型、设备状态等 |
| 3 | data | 存储模拟数据，包括数据编号、设备编号、数据类型、数据值、数据时间等 |
| 4 | permission | 存储用户权限信息，包括用户名、设备编号、权限类型等 |

其ER图如图4-1所示。

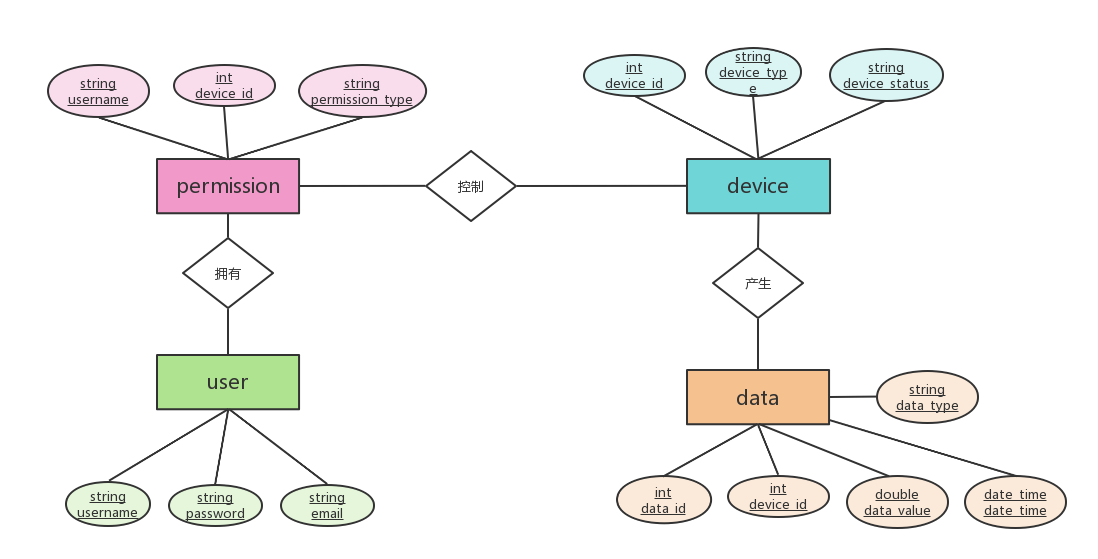


图4-1 数据库ER图

# 模块接口设计

## 全局变量

本系统没有使用全局变量，因为全局变量会增加程序的复杂度和出错的可能性，不利于程序的可读性和可维护性。

## 模块间接口函数设计

### 客户端模块

客户端模块提供了以下四个接口函数，如表5-1~5-4所示，用于与服务器进行通信。

表5-1 函数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **函数名** | void connectServer(string ip, int port) | | |
| **文件名** | client.cpp | | |
| **功能概要** | 连接服务器，输入服务器的IP地址和端口号，无返回值 | | |
| **参数** | | | |
| **类型** | **变量名** | **I/O** | **说明** |
| string, int | ip, port | I, I | 服务器的IP地址和端口号，IP地址为字符串类型，端口号为整数类型 |
| **详细说明** | | | |
| * 算法：使用TCP协议创建一个套接字，使用ip和port作为目标地址，调用connect函数尝试连接服务器，如果连接成功，则返回0，否则返回错误码 * 异常处理：如果connect函数返回错误码，根据错误码的不同，输出相应的错误信息，并终止程序 | | | |
| **使用注意事项** | | | |
| 在调用该函数之前，需要确保ip和port是有效的服务器地址，否则可能导致连接失败或连接错误的服务器 | | | |

表5-2 函数说明

|  |  |
| --- | --- |
| **函数名** | void disconnectServer() |
| **文件名** | client.cpp |
| **功能概要** | 断开服务器，无输入参数，无返回值 |
| **详细说明** | |
| * 算法：使用TCP协议关闭套接字，释放资源 * 异常处理：如果关闭套接字失败，输出错误信息，并终止程序 | |
| **使用注意事项** | |
| 在调用该函数之前，需要确保已经成功连接了服务器，否则可能导致关闭失败或关闭错误的套接字 | |

表5-3 函数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **函数名** | void sendData(string data) | | |
| **文件名** | client.cpp | | |
| **功能概要** | cpp 发送数据给服务器，输入要发送的数据，无返回值 | | |
| **参数** | | | |
| **类型** | **变量名** | **I/O** | **说明** |
| string | Data | I | 要发送的数据，为字符串类型 |
| **详细说明** | | | |
| * 算法：使用TCP协议发送数据，将data转换为字节流，调用send函数将字节流发送给服务器，如果发送成功，则返回发送的字节数，否则返回错误码 * 异常处理：如果send函数返回错误码，根据错误码的不同，输出相应的错误信息，并终止程序 | | | |
| **使用注意事项** | | | |
| 在调用该函数之前，需要确保已经成功连接了服务器，并且data是符合格式要求的字符串，否则可能导致发送失败或发送错误的数据 | | | |

表5-4 函数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **函数名** | string receiveData() | | |
| **文件名** | client.cpp | | |
| **功能概要** | 接收服务器的数据，无输入参数，返回接收到的数据 | | |
| **返回值** | **类型** | string | **说明** |
| **值** | data | 接收到的数据，为字符串类型 |
| **详细说明** | | | |
| * 使用TCP协议接收数据，创建一个缓冲区，调用recv函数从服务器接收字节流，并将字节流转换为字符串data，如果接收成功，则返回data，否则返回空字符串 * 异常处理：如果recv函数返回错误码或0（表示对方关闭了连接），根据情况输出相应的错误信息或提示信息，并终止程序 | | | |
| **使用注意事项** | | | |
| 在调用该函数之前，需要确保已经成功连接了服务器，并且缓冲区大小足够存储接收到的数据，否则可能导致接收失败或接收不完整的数据 | | | |

### 服务器模块

服务器模块提供了以下四个接口函数，如表5-5~5-8所示，用于与客户端进行通信：

表5-5 函数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **函数名** | void acceptClient(int port) | | |
| **文件名** | server.cpp | | |
| **功能概要** | 接受客户端的连接，输入端口号，无返回值 | | |
| **参数** | | | |
| **类型** | **变量名** | **I/O** | **说明** |
| int | port | I | 端口号，为整数类型 |
| **详细说明** | | | |
| * 算法：使用TCP协议创建一个套接字，使用port作为本地地址，调用bind函数绑定套接字和地址，调用listen函数监听端口，调用accept函数等待客户端的连接请求，如果连接成功，则返回一个新的套接字，否则返回错误码 * 异常处理：如果bind、listen或accept函数返回错误码，根据错误码的不同，输出相应的错误信息，并终止程序 | | | |
| **使用注意事项** | | | |
| 在调用该函数之前，需要确保port是一个有效的端口号，并且没有被占用，否则可能导致绑定或监听失败 | | | |

表5-6 函数说明

|  |  |
| --- | --- |
| **函数名** | void closeClient() |
| **文件名** | server.cpp |
| **功能概要** | 关闭客户端的连接，无输入参数，无返回值 |
| **详细说明** | |
| * 算法：使用TCP协议关闭套接字，释放资源 * 异常处理：如果关闭套接字失败，输出错误信息，并终止程序 | |
| **使用注意事项** | |
| 在调用该函数之前，需要确保已经成功接受了客户端的连接，否则可能导致关闭失败或关闭错误的套接字 | |

表5-7 函数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **函数名** | void sendData(string data) | | |
| **文件名** | server.cpp | | |
| **功能概要** | 发送数据给客户端，输入要发送的数据，无返回值 | | |
| **参数** | | | |
| **类型** | **变量名** | **I/O** | **说明** |
| string | data | I | 要发送的数据，为字符串类型 |
| **详细说明** | | | |
| * 算法：使用TCP协议发送数据，将data转换为字节流，调用send函数将字节流发送给客户端，如果发送成功，则返回发送的字节数，否则返回错误码 * 异常处理：如果send函数返回错误码，根据错误码的不同，输出相应的错误信息，并终止程序 | | | |
| **使用注意事项** | | | |
| 在调用该函数之前，需要确保已经成功接受了客户端的连接，并且data是符合格式要求的字符串，否则可能导致发送失败或发送错误的数据 | | | |

表5-8 函数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **函数名** | string receiveData() | | |
| **文件名** | server.cpp | | |
| **功能概要** | 接收客户端的数据，无输入参数，返回接收到的数据 | | |
| **返回值** | **类型** | string | **说明** |
| **值** | data | 接收到的数据，为字符串类型 |
| **详细说明** | | | |
| * 算法：使用TCP协议接收数据，创建一个缓冲区，调用recv函数从客户端接收字节流，并将字节流转换为字符串data，如果接收成功，则返回data，否则返回空字符串 * 异常处理：如果recv函数返回错误码或0（表示对方关闭了连接），根据情况输出相应的错误信息或提示信息，并终止程序 | | | |
| **使用注意事项** | | | |
| 在调用该函数之前，需要确保已经成功接受了客户端的连接，并且缓冲区大小足够存储接收到的数据，否则可能导致接收失败或接收不完整的数据 | | | |