ForTalk项目名称：系统

文档编号：1.0

ForChat项目概要设计

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 总页数 |  | 正文 |  | 附录 |  | 生效日期 |  |
| 编制 |  | | | 批准 |  | | |

**修 改 履 历**

|  |  |  |  |
| --- | --- | --- | --- |
| 修改编号 | 日期 | 版本号 | 修改内容 |
| 1 | 08-27 | 0.1 | 创建 |
| 2 | 08-29 | 0.2 | 修改 |
| 3 | 09-01 | 0.3 | 更新完善 |
| 4 | 09-02 | 0.4 | 总结 |
|  |  |  |  |

**目 录**

[1 文档概述 4](#_Toc182323070)

[1.1 文档目的和范围 4](#_Toc182323071)

[1.2 术语/缩略语 4](#_Toc182323072)

[1.3 参考文档 4](#_Toc182323073)

2 系统结构图.........................................................................................................................................................4

[2 模块概述 4](#_Toc182323074)

[2.1 模块功能定义 4](#_Toc182323075)

[2.2 模块结构 5](#_Toc182323076)

[2.3 模块动作时序 6](#_Toc182323077)

[3 接口说明 6](#_Toc182323078)

[3.1 数据结构定义 6](#_Toc182323079)

[3.2 函数 7](#_Toc182323080)

[3.2.1 模块间接口函数 7](#_Toc182323081)

[3.2.2 模块内接口函数 12](#_Toc182323082)

# 文档概述

## 文档目的和范围

本软件实现了利用socket实现了通过TCP连接为基础的网络聊天工具

## 术语/缩略语

|  |  |  |
| --- | --- | --- |
| 序号 | 术语/缩略语 | 说明 |
| 1 | Qt | Qt 是一种函数库是用来帮助制作图形交互界面的。 |
| 2 | Socket | Socket 又称"套接字"，应用程序通常通过"套接字"向网络发出请求或者应答网络请求。 |
| 3 | MySQL | MySQL 是一种关系数据库管理系统，关系数据库将数据保存在不同的表中。 |

## 参考文档

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 文档名 | 作者 | 时间 | 版本 |
| 1 | 概要设计模板.doc | 东软集团股份有限公司 | 2009.11.9 | 0.8.0-0.0.0 |
|  |  |  |  |  |

# 系统结构图

发送端

客户端2

接收端

客户端1

发送端

接收端

# 模块概述

## 模块功能定义

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **模块** | | **功能点** | **功能点详细内容** |
| 1.登录界面 | 1.1窗口标题 | | 显示窗口信息 | 显示当前登陆界面 |
|  | 1.2登录账号 | | 输入用户名 | 点击，验证密码用户名，匹配之后进入首页。 |
|  |  | | 输入密码 | 输入密码，密码由6-12位数字组成。 |
|  |  | | 记住密码 | 点击勾选记住密码 |
|  | 1.3注册新账号 | | 输入用户名 | 输入预注册的用户名，字母和数字组成，8-20位，判断是否合格以及重复，重复则提示输入新的用户名。 |
|  |  | | 输入密码 | 输入密码，密码由6-12位数字组成，判断是否合格。 |
|  |  | | 再次输入密码 | 再次输入密码，判断是否与第一次输入的密码相同。 |
|  |  | | 确认按钮 | 点击，注册完成。 |
|  |  | | 返回按钮 | 点击，返回登录界面。 |
|  | 1.4忘记密码 | | 忘记密码按钮 | 点击，进入找回密码界面。 |
| 2.主界面 | 2.1个人信息 | | 用户名 | 显示用户名 |
|  |  | | 性别 | 显示用户性别 |
|  | 2.2好友列表 | | 分组显示 | 显示用户好友分组 |
|  | |  | 组内成员显示 | 展开或折叠当前组内好友 |
|  | |  | 双击组内成员 | 显示该成员个人信息 |
|  | | 2.3个人信息窗口 | 修改个人用户名 | 左键双击修改用户名 |
| 3.聊天界面 | | 3.1聊天窗口 | 文本输入框 | 输入所要发送给好友的信息 |
|  | |  | 文本输出框 | 输出双方发送过的历史信息 |
|  | |  | 历史记录按钮 | 单击显示历史信息 |
|  | |  | 好友信息显示界面 | 显示好友用户名或IP |
|  | |  | 字体和颜色 | 改变输入文字的字体及颜色 |
|  | |  | 发送按钮 | 发送文本到文本输出框 |
|  | |  | 关闭按钮 | 关闭聊天界面 |
|  | | 3.2图片及文件传输窗口 | 单击图片或文件打开 | 选择要发送的图片或文件 |
|  | |  | 单击图片或文件发送 | 发送选中的图片或文件 |
|  | |  | 下载图片或文件到本地 | 右键保存图片或文件到本地 |
| 4.多线程 | | 4.1创建线程对象 | 创建多窗体进程 | 实现多窗体进程 |
|  | |  | 创建通信进程 | 可以实现双向的通信 |
|  | |  | 创建文件接收进程 | 可以实现文件的双向接收 |
| 5.mysql数据库 | | 5.1设计数据库 | 数据库设置及设计用户信息表 | 可以设置端口，以及一些其他的用户信息 |
|  | |  | 用户好友表 | 可以创建用户好友表的信息 |
|  | | 5.2数据库连接 | 用户与数据库连接 | 实现用户与数据库的连接功能 |
|  | | 5.3数据库操作 | 更新用户信息好友IP地 | 实时更新好友的信息和用户个人的信息 |
| 6.窗口标题栏 | | 6.1窗口标题栏功能键 | 最大化按钮 | 实现窗口最大化 |
|  | |  | 最小化按钮 | 实现窗口最小化 |
|  | |  | 关闭按钮 | 实现关闭窗口 |

## 模块结构

|  |  |  |
| --- | --- | --- |
| 模块名称 | 模块类型 | 概要说明 |
| 登陆注册界面 | 外部模块 | *槽函数的简单运用及数据库的管理* |
| 收发界面 | 外部模块 | *TCP链接的运用* |
| 编辑、阅读界面 | 外部模块 |  |
| 数据库模块 | 内部模块 | 数据库底层函数设计 |
| Socket模块 | 内部模块 | 数据传输函数设计 |

## 模块动作时序

描述模块间动作的时序图。可以在其它文件中记述(这里说明参考「XXX」文件)。

# 接口说明

## 函数

给出模块所用数据结构的说明，以及全局变量和宏常量。

namespace Ui {

class loginDlg;

}

class loginDlg : public QDialog

{

Q\_OBJECT

public:

explicit loginDlg(QWidget \*parent = nullptr);

~loginDlg();

private slots:

void on\_loginBtn\_clicked();

private:

Ui::loginDlg \*ui;

};

class MainWindow;

}

class MainWindow : public QMainWindow

{

Q\_OBJECT

public:

explicit MainWindow(QWidget \*parent = 0);

~MainWindow();

private:

Ui::MainWindow \*ui;

QTcpSocket \*tsocket;

private slots:

void connectHostSlot(); //请求建立连接

void receiveTcpSlot(); //槽函数用于对接受的数据进行处理

void sendTcpSlot(); //发送tcp信号槽（发送按钮触发）

void tcpDisconnect(); //tcp连接中断

};

namespace Ui {

class MainWindow;

}

class MainWindow : public QMainWindow

{

Q\_OBJECT

public:

explicit MainWindow(QWidget \*parent = 0);

~MainWindow();

QImage \*image;

QTimer \*timer;

//线程实验

myThread \*thread;

int pb;

private:

Ui::MainWindow \*ui;

QTcpServer \*tserver;

QTcpSocket \*tsocket;

private slots:

void listenStartSlot(); //开始侦听槽

void newConnectSlot(); //收到新连接

void receiveTcpSlot(); //槽函数用于对接受的数据进行处理

void sendTcpSlot(); //发送tcp信号槽（发送按钮触发）

void tcpDisconnect(); //tcp连接中断

//线程实验

void closeThreadSlot();

};

class myThread : public QThread

{

Q\_OBJECT

public:

myThread(int count);

~myThread();

int count; //自己添加类成员

protected:

void run();

};

### 模块间接口函数

#### Send

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **函数名** | Send | | | |
| **功能概要** | 发送 | | | |
| **记述形式** | int Send (int socket\_fd ,char\* pcBuffer) | | | |
| **参数** | | | | |
| **类型** | **变量名** | | **I/O** | **说明** |
| int | socket\_fd | | I | 套接字标示符 |
| char\* | pcBuffer | | I | 缓冲区 |
| **返回值** | **类型** | Int | **说明** | |
| **值** | 0 | 发送成功 | |
| 1 | 发送失败 | |

#### Receive

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **函数名** | Receive | | | |
| **功能概要** | 接收 | | | |
| **记述形式** | int Receive (int socket\_fd ,char\* pcBuffer) | | | |
| **参数** | | | | |
| **类型** | **变量名** | | **I/O** | **说明** |
| int | socket\_fd | | I | 套接字标示符 |
| char\* | pcBuffer | | I | 缓冲区 |
| **返回值** | **类型** | Int | **说明** | |
| **值** | 0 | 接收成功 | |
| 1 | 接收失败 | |

### 模块内接口函数

#### CheckLoginInput

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **函数名** | CheckLoginInput | | | |
| **功能概要** | 检查用户输入的账号密码时候是否合法（客户端，模块内） | | | |
| **记述形式** | int CheckLoginInput (char \*pcUserID, char\* pcPasswd) | | | |
| **参数** | | | | |
| **类型** | **变量名** | | **I/O** | **说明** |
| char\* | pcUserID | | I | 账户 |
| char\* | pcPasswd | | I | 密码 |
| **返回值** | **类型** | int | **说明** | |
| **值** | 0 | 输入合法 | |
| 1 | 账户含有非法字符 | |
| 2 | 密码不合法(长度或非法字符) | |

#### CheckUserIdExist

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **函数名** | CheckUserIdExist | | | |
| **功能概要** | 调用socket函数判断是用户名是否存在（客户端，模块内） | | | |
| **记述形式** | int CheckUserIdExist (char \*pcUserId) | | | |
| **参数** | | | | |
| **类型** | **变量名** | | **I/O** | **说明** |
| char\* | pcUserId | | I | 用户ID |
| **返回值** | **类型** | int | **说明** | |
| **值** | 0 | 已经被注册 | |
| 1 | 未被注册 | |

#### GetNewMail

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **函数名** | GetNewMail | | | |
| **功能概要** | 获取未读邮件信息 | | | |
| **记述形式** | Int GetNewMail(stMail \* pstNewMail) | | | |
| **参数** | | | | |
| **类型** | **变量名** | | **I/O** | **说明** |
| stMail \* | pstNewMail | | O | 新邮件信息 |
| **返回值** | **类型** | Int | **说明** | |
| **值** | 0 | 获取成功 | |
| 1 | 无新邮件 | |
| 2 | 通信失败 | |