1

东软睿道

项目实训总结报告

项 目 名 称:LCS实时通讯软件

学 校 名 称:北京理工大学

项 目 讲 师:崔浩

姓 名:陈誉航

学 号:1820201062

实 训 日 期: 2022年8月15日-8月25日

**1 引言**

1.1项目概述

本项目来自于北京理工大学学生在计算机基础上所实践并提升能力的训练计划，目的是在Linux系统下制作一个局域网内的聊天软件。由于Linux操作系统是C语言所编译的，所以对不同架构的CPU都有着良好的兼容性。另外，本项目是建立在QT软件上，因为QT是个简单易用的开发框架的软件。所以，在Linux操作系统下开发QT项目能减小解决所要面对的接口问题。

本项目所涉及的关键技术有TCP/IP协议通讯、数据库Sqlite、QT图形开发工具、Socket“套接字”、MD5等。本报告将详细介绍我在LCS项目中所负责的开发制作过程。

1.2 需求分析

QT是一个跨平台的C++应用程序开发框架。广泛用于开发GUI程序，这可称为部件工具箱。它也可用于开发非GUI程序，例如控制台工具和服务器。而制作实时通讯软件能让整组成员同时认识到许多功能。实时通讯软件也是现今时代人们最常使用的软件，我们能通过本项目了解到许多基础开发通讯软件的技巧。日后对我们非常有帮助。

1.3 运行环境

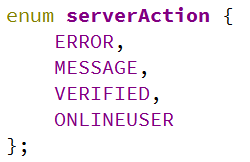
软件环境：QT Creator 4.11.2 + Windows + Linux

硬件环境：Intel 3以上，2GB RAM

使用GitHub进行成员的程序同步

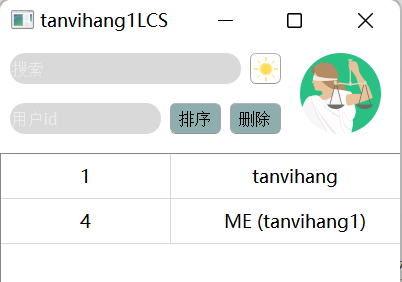
**2 项目设计**（小三黑体，居中）（以个人主要负责内容为主）

2.1 设计思路

主要设计的部分是好友界面，其必须连接数据库才能显示好友列表，并且实现在线功能，这就会连接到服务器来获取在线信息。实现双击点击好友进入聊天界面，此部分需要设计大量信号与槽来实现，因为需要从主界面获取信息并传入聊天界面，并且同步的对聊天对象进行更新。

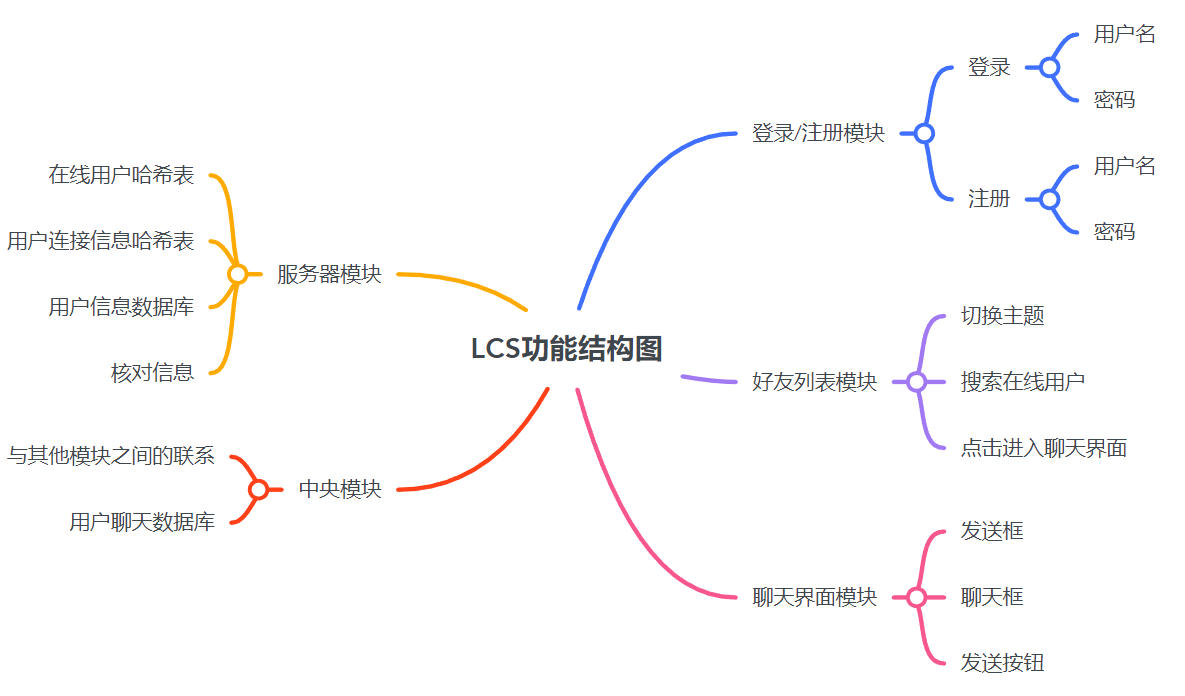
为了方便用户端与服务器之间的发送信息类型，在服务端设置了让用户端快速辨认操作。

同理的，用户端也会有来让服务器快速辨认操作。

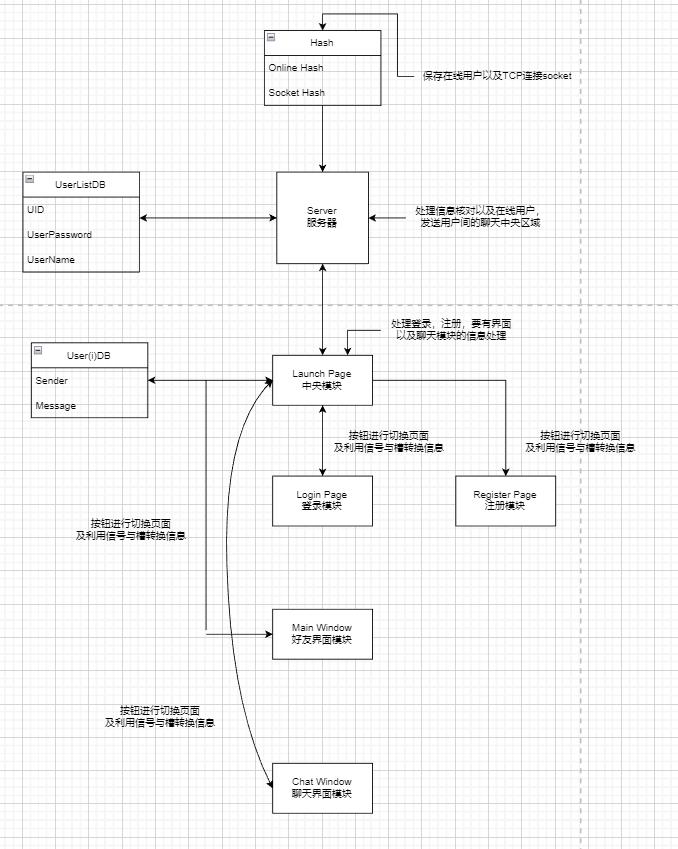
2.2 模块功能介绍

在主界面中，通过更换主题按钮，用户将可以更换主题，也可通过点击好友列表中的好友来进入有他的聊天界面。

2.3 模块结构图



2.4 程序流程图



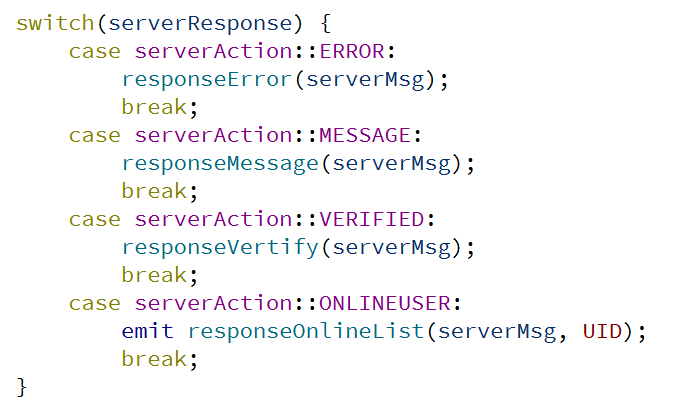
2.5 功能设计分工

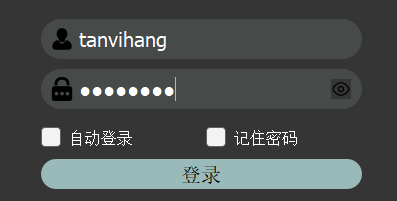
根据本项目的程序流程图，我们组将分工分成五份。分工安排如下：

1. 李沄澺——负责开发UI设计。
2. 刘乐也——负责添加登录与注册界面的控件函数
3. 陈誉航——负责添加主页面控件函数、数据库
4. 刘宇捷——负责聊天界面控件函数
5. 李思净——负责设计服务器、调试整体代码以及页面连接。

**3 详细设计**

我们利用了用户->服务器->用户的方式进行聊天/操作，这样子能够确保实时更新信息。我们在服务器设置了收集用户信息的数据库，进行登入时必须通过服务器验证才能进入主页面。随后我们在客户端里设置了聊天数据库，用以保存历史聊天记录。这里所使用的技术包含tcp连接，以及信号与槽。Tcp与中央模块连接，通过enum来辨认互相传送的信息并做出动作如：登入，获取在线列表，等等的操作。



接下来就是信号与槽的设计来连接中央模块和其他模块（好友列表，聊天界面，登录/注册界面），之间的任何操作都必须经过中央模块来执行，比如说想要从登录界面按下登录按钮，登录界面会发出信号去中央模块则会对这个信息做出动作 &LaunchPage::requestLogin



接下来就发送信息给服务器说有登录用户，获得准证后（密码匹配）服务器发送serverAction::VERIFIED:并再执行responseVertify(serverMsg);

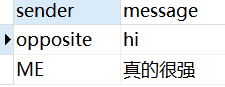
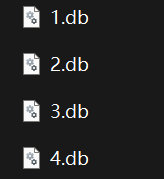
打开聊天窗口。

这里就时我们这个项目中的主要行为，也就是用户->服务器->用户。

**4 实现方案**

1.登录后的操作

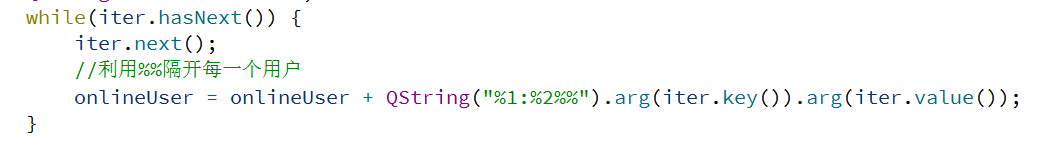
成功登陆后触发函数 void LaunchPage::**responseVertify**(QString data)，其功能包含创建该登录用户自己的数据库用来保存会话记录(该部分将在聊天窗口打开时使用)。db.setDatabaseName(QString("../%1.db").arg(UID));。



并且开启好友窗口（上图），以及向server端发出信息来获取在线用户信息QString onlineList=QString("LCS|%1|%2|").arg(UID).arg(todoAction::ONLINELIST);clientSocket->write(onlineList.toUtf8());

2.获取在线用户显示在表上

当server端接收到信号时，server将从Hash里读取在线用户数据并发送



QString returnOnline = QString("LCS|%1|%2").arg(serverAction::ONLINEUSER).arg(onlineUser);

socketHash.find(UID).value()->write(returnOnline.toUtf8());

回给发送端，并且触发信号case serverAction::ONLINEUSER:emit responseOnlineList(serverMsg, UID);其槽函数为connect(this, &LaunchPage::responseOnlineList, &mainPage, &MainWindow::responseOnlineList);最后主窗口读取数据（serverMsg）并更新在线列表

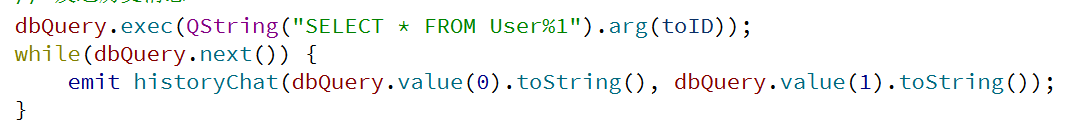


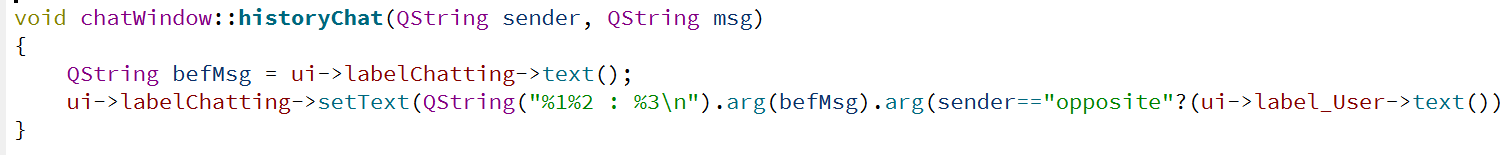
3.双击用户进入聊天页面

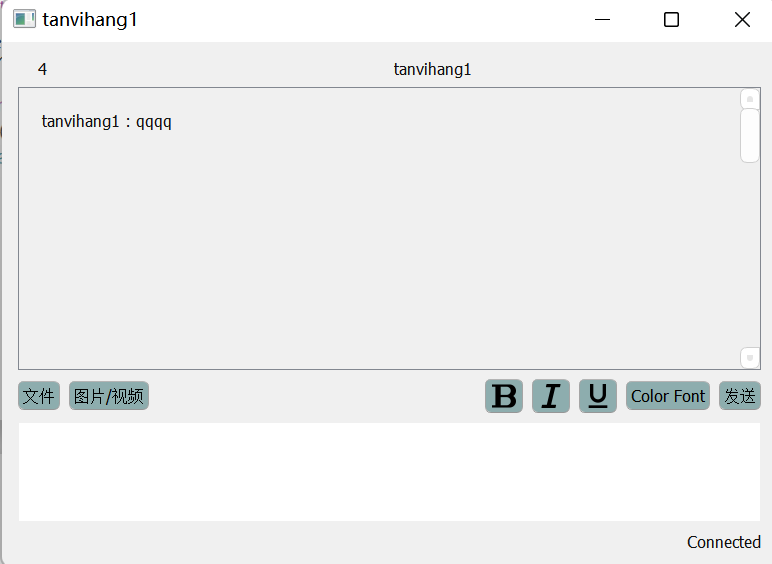
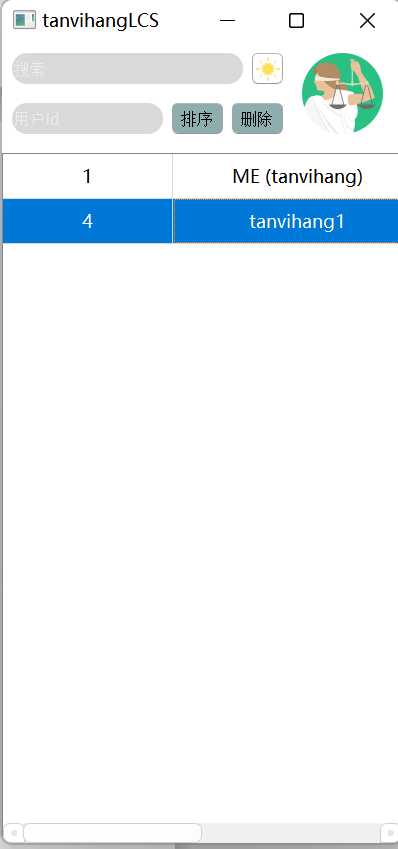
双击在线用户时发送信号emit requestChat(toId, toUsr);向launch page请求打开聊天窗口connect(&mainPage, &MainWindow::requestChat, this, &LaunchPage::requestChat);打开聊天窗口chatPage.show();后

并且发出信号emit responseChat(toID, toUsr);让聊天窗口获取用户以及对方信息。

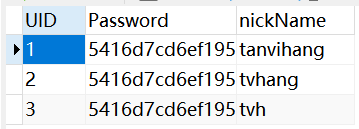


最后读取历史聊天记录





**5 系统测试**

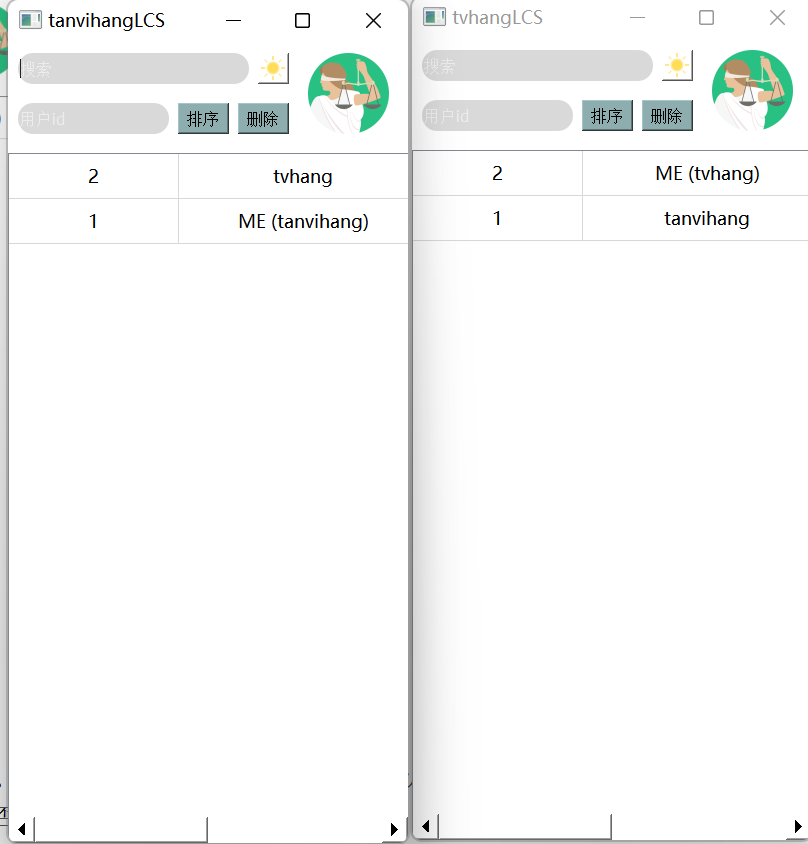


尝试登录（此账号已存在数据库）



登录成功后成功进入好友界面。（由于现在没有同时在线用户，所以没有显示好友）

所以我在登录另一个账号看是否会出现在这里。



可以看到在线用户成功显示在上面了。

**6 心得体会**

这次是第一次开发局域网的程序，并且使用了数据库以及类等技术，让我实际体会到了大一及大二所学习的知识点的实际应用了，也因为这个实践我学习了更多应用方面的知识，比如如何创建一个表来存取用户之间的聊天记录，如何使用类来创建窗口等。这一次实践的难点也非常的多，由于是第一次开发局域网类的程序，并不了解服务端与客户端之间的操作要如何区分，数据库应该存在哪一段，如何通过信号与槽以及网络协议连接服务端获取信息等的模式。最后组长所使用的连接方法让我学习了不少，任何重要的操作必须通过服务端的辨认后才返回给客户端继续操作，这样的设计也很符合真实世界的服务端与客户端的需求。我们这组每一个人都做尽了自己该做的事情，也各自学习更多技术。