

东软睿道

项目实训总结报告

项 目 名 称: WeTalk 广域网聊天软件

学 校 名 称: 北京理工大学

项 目 讲 师: 崔浩

姓 名: 罗家安

学 号: 1120203088

实 训 日 期: 2022 年 8 月 15 日-8 月 25 日

1 引言

1.1 项目概述

本项目旨在设计一款基于 QtQuick, QML, Qt, SQLite 和云服务器的支持广域网的聊天应用, 为用户提供方便快捷的远程通信和自由交友, 能够跨越距离的限制, 能够在办公, 娱乐, 日常生活等多场景下使用, 时刻保持信息畅通。在项目中我们主要借鉴了 QQ 和微信的相关功能进行开发。

1.2 需求分析

在互联网时代, 依靠电话和短信的通信手段, 远远满足不了人们的需求且效率较低, 成本较高, 在许多场景下, 需要文件和多人信息传输, 此外, 生活水平的日益提高, 人们对于交友和娱乐的追求也随之提高, 为此我们开发了 WeTalk 应用, 支持自由交友, 广域网连接, 建立群聊, 好友单聊, 设计个人名片等能够为人们日常生活, 办公提供便利的功能。

1.3 运行环境

操作系统: 兼容 Windows10、11 和 Ubuntu Server 20.04 LTS

软件: Qt5.12.8

在项目的实际运行中, 客户端可在 Windows 和 Ubuntu 下运行, 服务端在云服务器上运行, 云服务器的操作系统为 Ubuntu Server 20.04 LTS。

2 项目设计

2.1 设计思路

项目分成客户端、服务器端和数据库端, 其中服务器端放在公网下的云端服务器上, 达到公网即时通信的目的。项目系统的整体架构和各架构下的内容如图 2-1 所示, 以服务器端为开发中心, 连接三个部分的通讯。

开发过程采用前后端分离技术, 并写好对应的接口函数和信号, 最后合并时进行连接调整。这有利于我们分别独立进行开发, 分工明确, 开发效率高。

在本项目中, 我主要负责服务端的开发、云服务器的搭建与运维以及服务端与客户端的接口函数的代码编写, 同时撰写了相关的文档。

2.2 模块功能介绍

我们的项目主要分为客户端、服务端和数据库端三大模块。我主要负责服务端的开发、云服务器的搭建与运维以及服务端与客户端的接口函数的代码编写。

我主要负责的服务端作为聊天软件的控制中心, 需要能够接收来自客户端的请求, 对请求进行解析处理。针对不同类型的请求分别设计响应逻辑, 实现与数

数据库的数据交互，并向客户端发回对应的响应。为此，服务端又细分实现了以下九大模块的核心功能。

- 1. 登录注册响应模块：接收客户端发来的登录和注册请求，注册时向数据库写入用户信息，登陆时与数据库进行验证对比，并返回是否登陆成功。
- 2. 查找信息模块：接收来自客户端的搜索请求，按照不同的请求类型（查找群或者查找用户），从服务器调取并向客户端返回查找的信息。
- 3. 更改信息模块：接收来自客户端的个人和群信息，并写入数据库完成更新。
- 4. 消息传输模块：接收来自客户端发出的消息，判断聊天性质（私聊或者群聊），向目标客户端转发消息，并完成聊天记录向数据库的存储。
- 5. 搜索响应模块：接收来自客户端的检索请求，在数据库中查找对应 id 的群或者用户信息，向客户端返回搜索结果。
- 6 创建群聊模块：接收来自客户端的建群请求，创建新的群并存入数据库中，向服务端返回新建群聊的 id 等信息。
- 7. 添加好友或群模块：接收来自客户端的添加好友或群的信息，向客户端返回添加成功或者失败信息，并更新数据库。
- 8. 列表刷新响应模块：接收服务端的刷新请求，解析请求的性质（刷新好友列表或者群列表），并向服务端返回数据库中的最新信息。
- 9. 历史消息相应模块：接收服务端下载群/私聊聊天记录请求，在服务器中提取历史聊天记录，并向服务端返回信息。

2.3 模块结构图

我们的项目主要分为客户端、服务端和数据库端三大模块的结构和各模块之间的关系如下图所示。

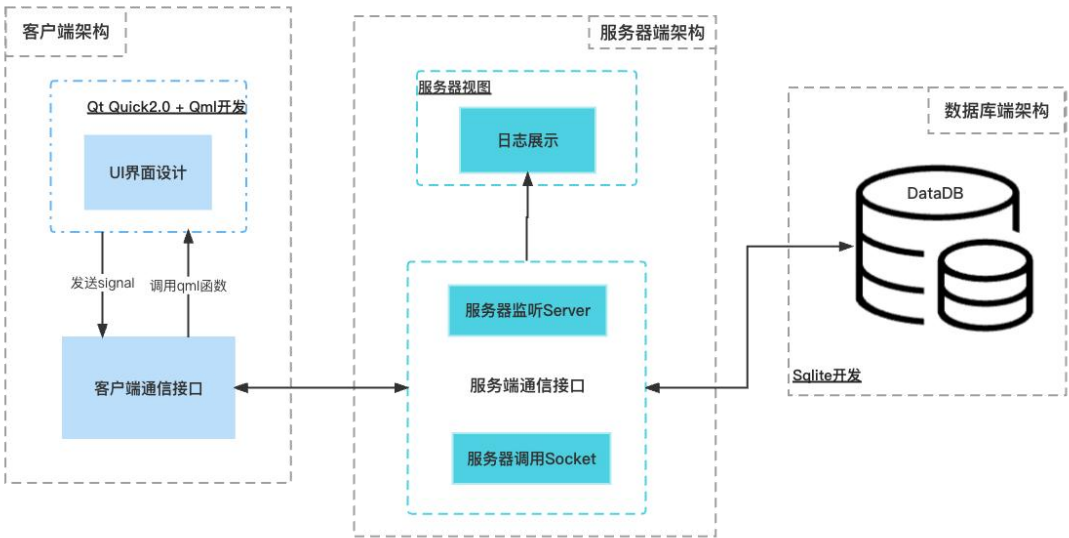
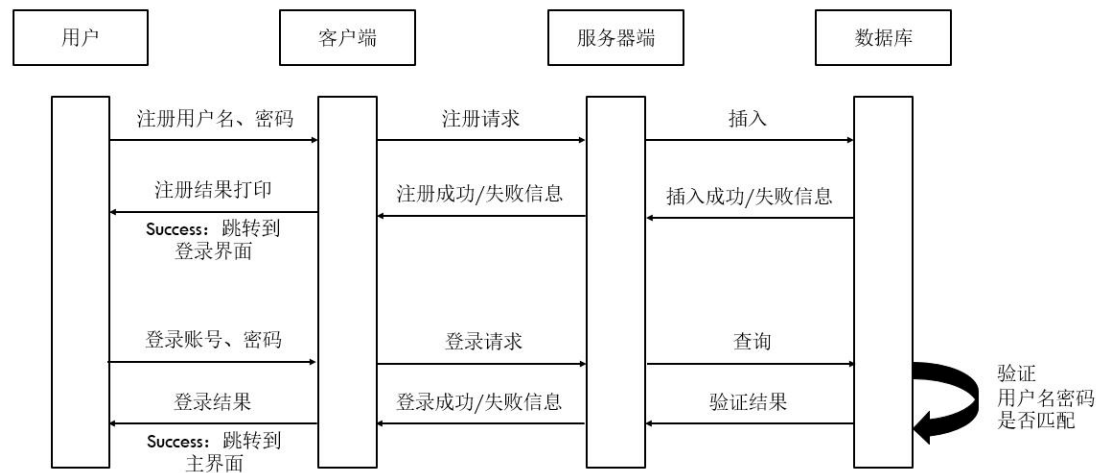


图 2-3 项目系统模块结构图

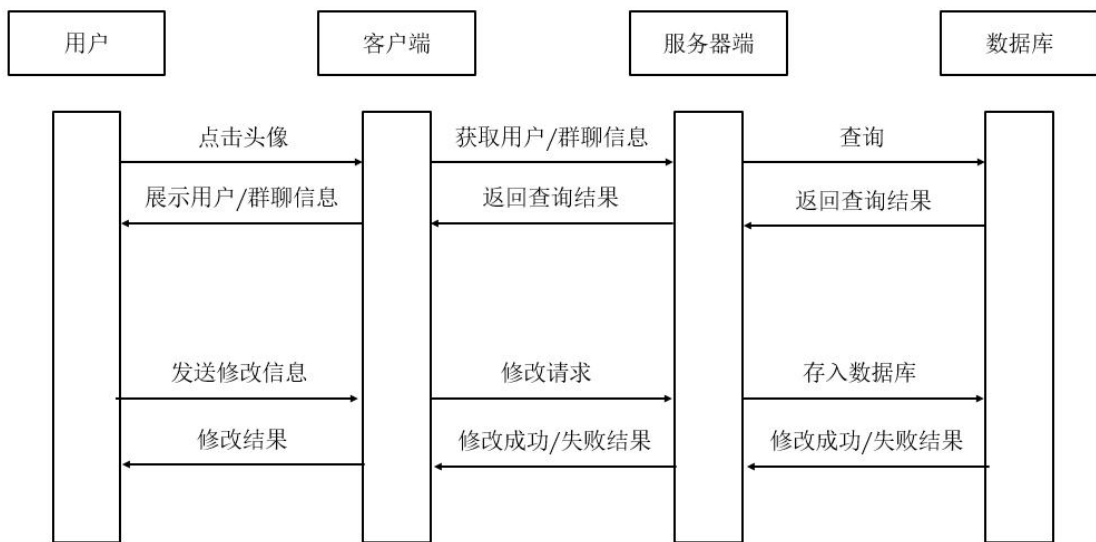
2.4 程序流程图

下面以上面介绍的服务器端的几个模块为例，介绍项目程序的流程图。

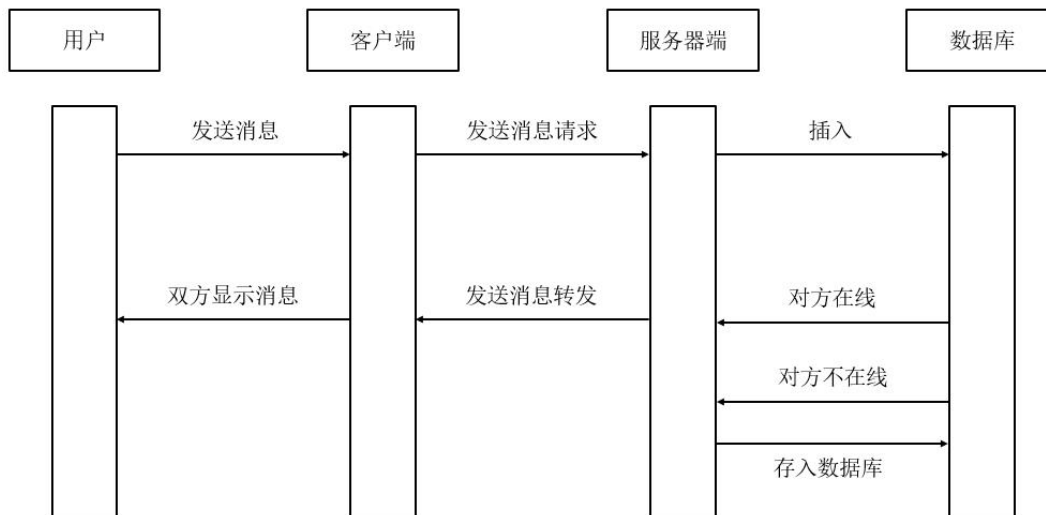
2.4.1 登录注册响应模块



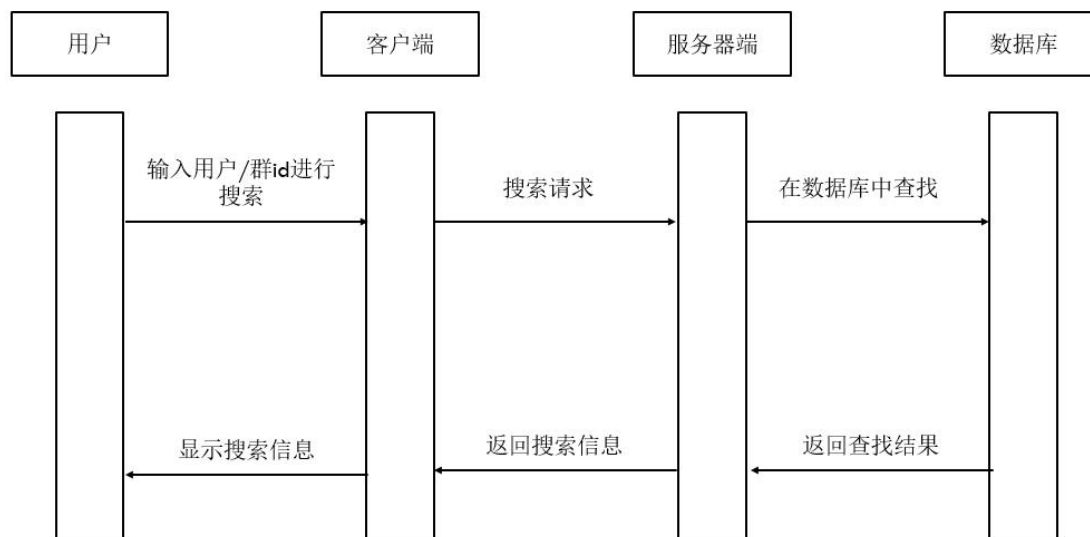
2.4.2 查找和更新信息模块:



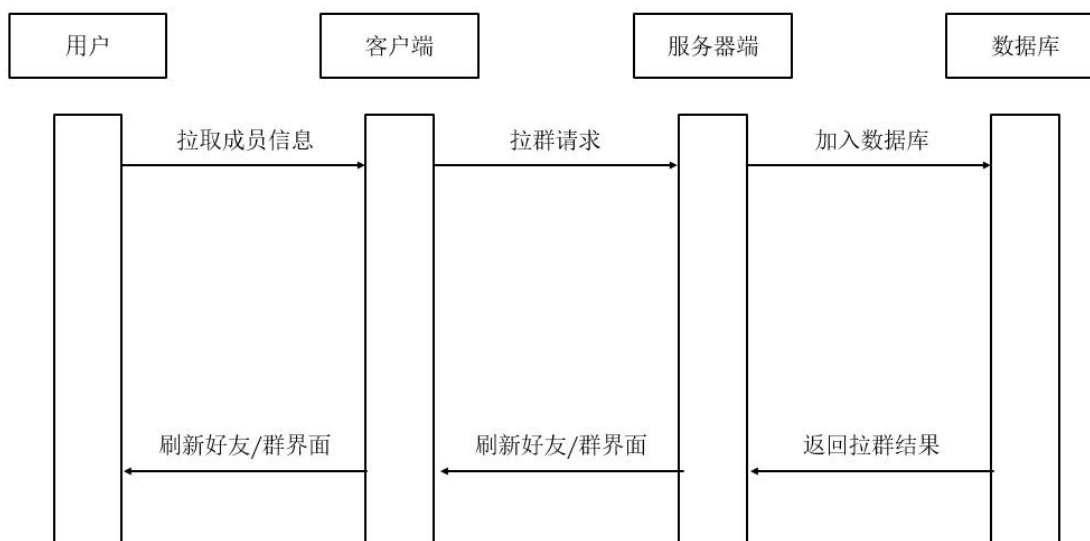
2.4.3 消息传输模块:



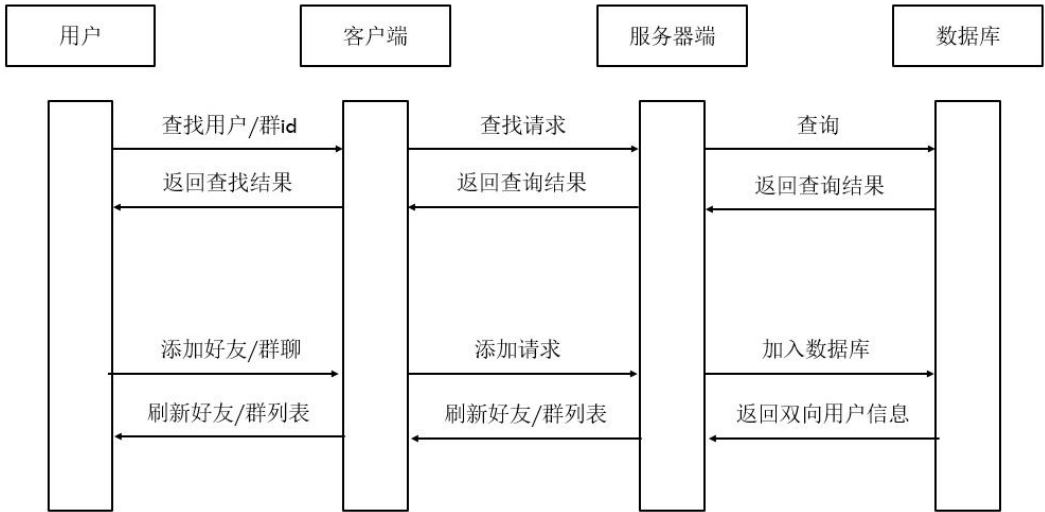
2. 4. 4 搜索响应模块:



2. 4. 5 创建群聊模块:



2. 4. 6 添加好友/群及列表刷新响应模块：



2. 5 功能设计分工

- 服务器端：陈耀宇、罗家安
- 数据库：刘恩泽
- 设计美化客户端 ui 界面：刘翎、沈宇航
- 界面跳转：刘翎、沈宇航
- 自定义控件：刘翎、沈宇航
- 合法性检验：刘翎、沈宇航
- 客户端与服务器端接口函数：陈耀宇、沈宇航、罗家安
- 服务器端与数据库接口函数：陈耀宇、刘恩泽
- 测试：陈耀宇、沈宇航、刘翎、刘恩泽、罗家安
- 项目文档创建与修改：刘翎、刘恩泽、罗家安、沈宇航、陈耀宇

3 详细设计

本项目采用 Qt5 和 QML 开发，采用典型的服务器-客户端架构，依托 Qt 的跨平台特性可以运行在绝大多数 Qt 支持的平台上。服务器放在云端公网上，与客户端之间的通信主要使用 socket 实现。通过服务器启动的各项配置参数可以适配 Qt 支持的主流数据库。核心功能为用户的注册、登录，私聊或群聊中即时消息的收发，历史记录缓存。

在客户端的设计中主要包含两个方面，一个是负责与服务器端通信的客户端通信模块，一个是负责整个客户端的 ui 绘制与逻辑处理的客户端 ui 模块，整个客户端包含多个 ui 界面，每个 ui 界面中会包含自定义控件用来显示各种个人信息，或聊天信息，通过信号与槽的来与客户端通信模块交互往服务器端请求信息。客户端的界面包含登录界面、注册界面、主界面、聊天界面、信息界面、通讯录界面、添加好友/群聊界面。

在我负责的服务器端设计中主要考虑了以下三个方面。第一个方面是负责接收来自客户端的请求，并对该请求进行解析。当客户端发出捕获到用户点击的信号之后，会调用用户端与服务端的接口函数，将传输的信号封装成 json 字段，通过 UDP 传输协议向服务端发送请求。每个 json 字段中的 command 键值对应不同的请求类型。第二个方面是负责与数据库方面的交互，实现从数据库中拉取个人信息、好友信息、群信息和历史消息。服务器与数据库一起部署在云端，服务器通过调用数据库的各种函数接口实现对数据的增删改查操作。第三个方面是负责向客户端传回响应，返回对客户端请求的回应。服务端将传回的数据封装在 json 字段中，通过 UDP 协议传回客户端，以 command 字段存储不同的回应类型，客户端收到 json 字段后，对 json 进行解析之后调用前端不同的 QML 函数来响应用户的请求。

打开客户端时，首先会进行与服务器的连接，若不成功会给出警告；注册界面与登录界面是相关联的，注册界面包含有邮箱验证码和密码匹配提示功能，注册成功后会分配 ID 并自动将 ID 和密码填充入登录框中，点击登录后会与服务器进行通信确认登录信息，进入主界面后显示聊天界面，并显示历史聊天记录。点击聊天界面内个人头像或好友头像，弹出用户详细信息界面，若为个人信息界面，还有设置功能键，点击后可以修改个人信息和头像，点击确认后更新信息和头像。从主界面可以切换到通讯录界面，点击好友/群聊列表的头像，可以进入好友/群聊信息界面；点击对应好友/群聊的聊天按钮，自动切换到对应的聊天界面，进行聊天。聊天时可以选择表情、发送文字。同时，在通讯录界面包含添加好友/群聊，以及创建群聊功能按钮，点击后实现或进入对应功能界面。

本项目使用到了云服务器。我在腾讯云上租用了一台轻量级云服务器，双核 CPU，4GB 内存，足以满足我们聊天软件服务端的需求。云服务器上安装了 Ubuntu Server 20.04 LTS 操作系统，并完成了一系列初始环境以及 Ubuntu 桌面的配置。为了实现网络传输，云服务器也完成了相关网络的配置。在刚开始使用云服务器时，由于防火墙开放程度较高以及 root 密码设置较为简单，服务器多次遭到了来自境外黑客的攻击。于是云服务器又多次升级了安全配置，确保了软件的运行环境安全稳定。

服务器IP/名称	来源IP	来源地	协议	登录用户名	端口	首次攻击时间	最近攻击时间	尝试次数	破解状态	阻断状态	事件状态
10.0.4.2 Ubuntu-mhGR	78.173.68.219	土耳其	ssh	root	22	2022-08-22 21:57:11	2022-08-22 22:14:04	40	破解成功	未阻断 非普通版、非专业版、非旗舰版	已处理

图 3 服务器曾经遭到的网络攻击

4 实现方案

以下是后端服务器的核心函数，QQServer 类完成的是服务器接收客户端请求

功能，QQClient 类完成的是客户端接收服务器响应功能。

1. 登录注册响应模块：
 QQServer::registerRespond
 QQServer::loginRespond
 QQClient::registerBack
 QQClient::loginBack
2. 查找信息模块：
 QQServer::getPersonalInfoRespond
 QQServer::getGroupInfoRespond
 QQClient::personInfoBack
 QQClient::groupInfoBack
3. 更改信息模块：
 QQServer::changePInfo
 QQServer::changeGInfo
4. 消息传输模块：
 QQServer::sendChatMessageRespond
 QQClient::sendChatMessageBack
4. 搜索响应模块：
 QQServer::searchRespond
 QQClient::searchBack
6. 创建群聊模块：
 QQServer::createGroupRespond
 QQClient::createGroupBack
7. 添加好友或群模块：
 QQServer::addRespond
 QQClient::addBack
8. 列表刷新响应模块：
 QQServer::friendRespond
 QQServer::groupRespond
 QQClient::friendBack
 QQClient::groupBack
9. 历史消息相应模块：
 QQServer::messageRespond
 QQClient::messageBack

程序运行图片

注册

用户名

密码

确认密码

邮件地址

验证码

发送验证码

注册

返回

注册页面

用户名

密码

提示

 验证失败

Cancel

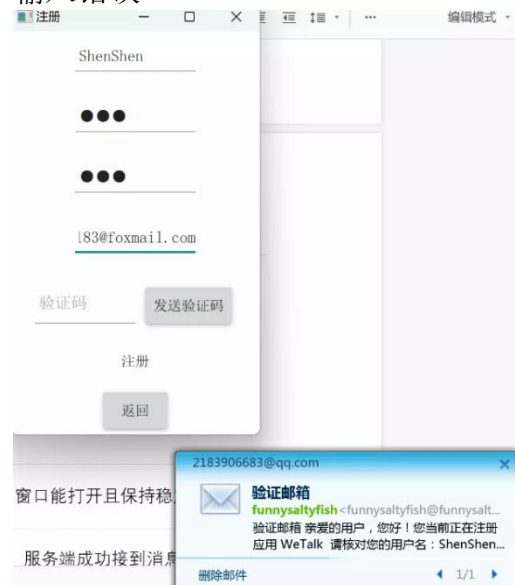
验证码

发送验证码

注册

返回

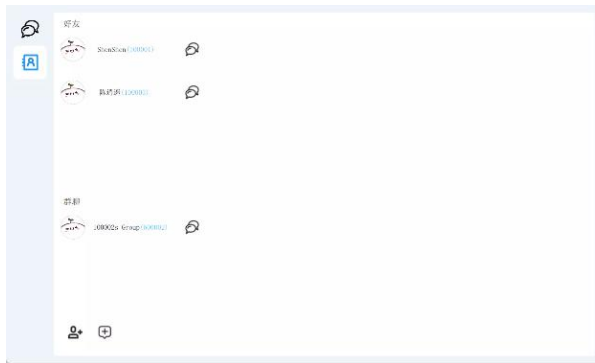
输入错误



验证码发送



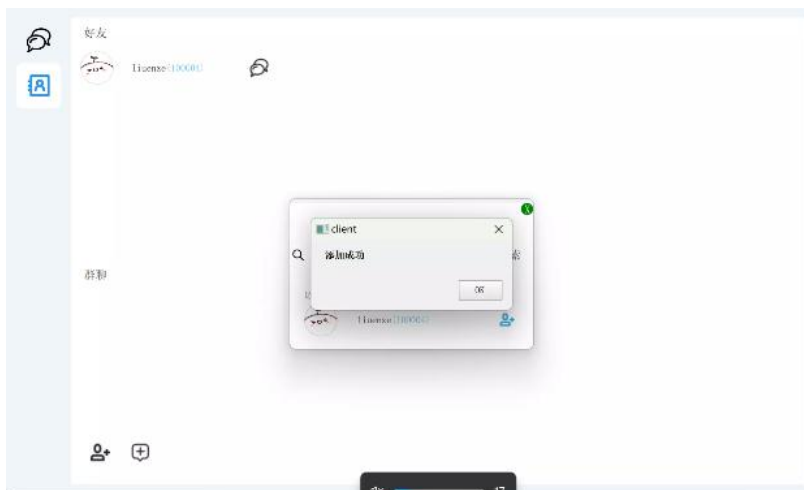
登录页面（自动跳转，自动填充内容）



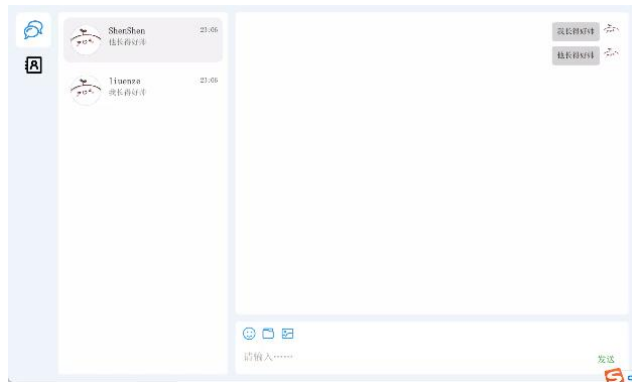
好友和群聊页面



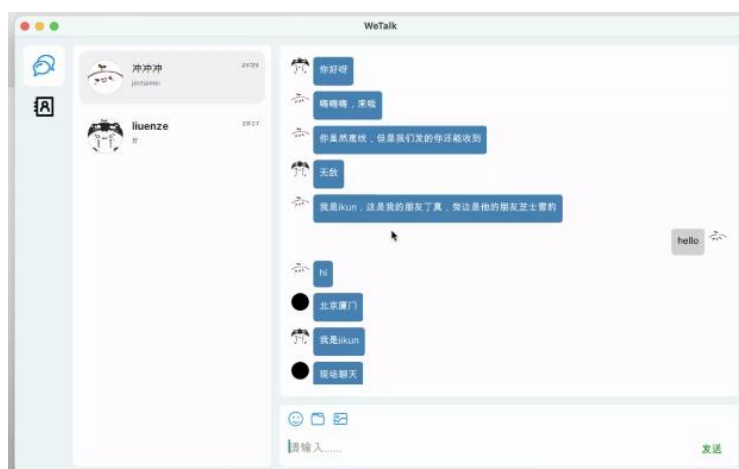
搜索添加好友/群聊



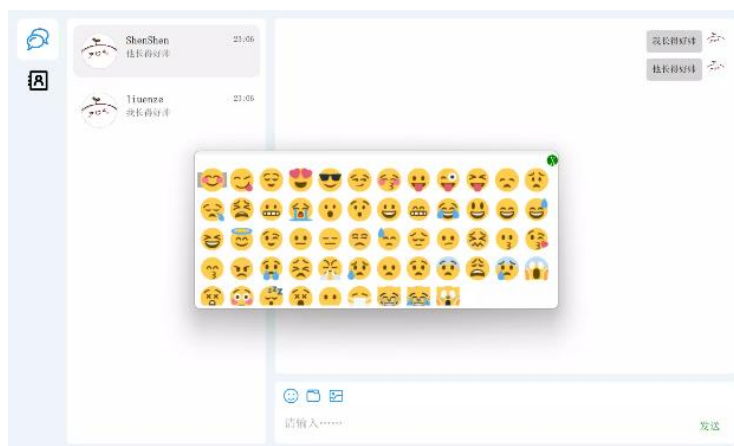
加好友成功弹窗



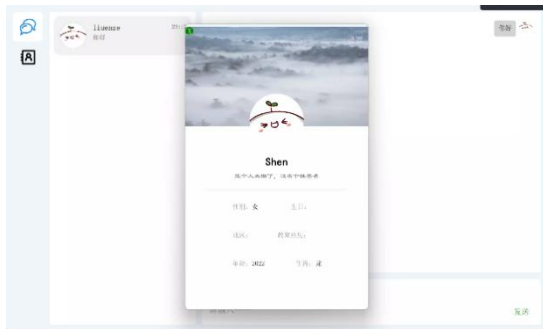
聊天页面和历史消息页面（广域网，支持离线信息收取）



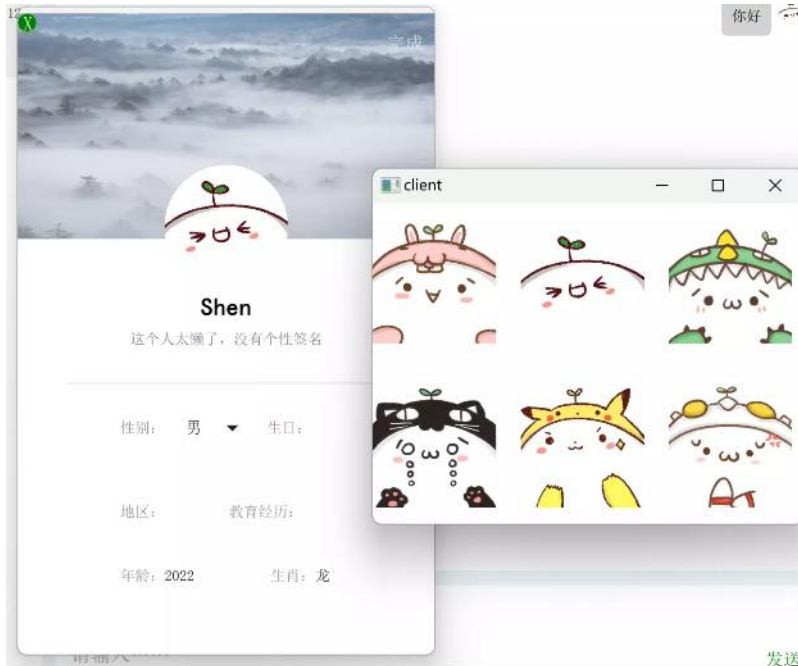
群聊演示



发送表情



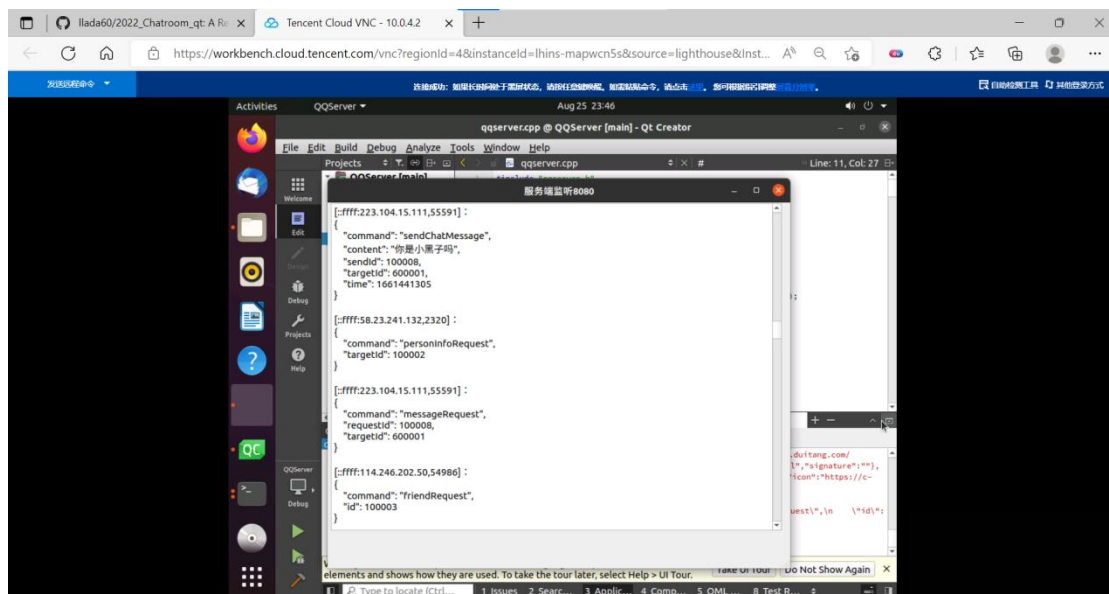
点击头像可以查看个人信息（包括聊天列表、好友列表均可）



个人信息修改（所示信息均可修改，修改生日自动计算年龄和生肖。头像可选择）



云端服务器监控



5 系统测试

测试方法	测试描述	预期结果
客户端注册登录界面是否稳定	打开 client	窗口能打开且保持稳定
注册信息是否能发到服务端	打开 client, 尝试注册, 查看服务端是否接收到消息	服务端成功接到消息
注册信息是否存入数据库	打开 client, 输入注册信息, 查看服务端的数据库是否存入了数据	服务端的数据库内有刚刚注册的相关信息
登录信息是否能发到服务端	打开 client, 尝试登录, 查看服务端是否接收到消息	服务端成功接到消息
服务端能否返回注册、登录正确信息	打开 client, 尝试注册、登录, 查看客户端是否接到消息	客户端成功接到服务端返回的信息
服务端能否返回登录错误信息	打开 client, 输入错误的登录信息, 查看客户端能否接到登录错误的相关信息	客户端成功显示错误信息
客户端能否对错误的注册信息作出回应	打开 client, 输入错误的注册信息, 客户端能否做出回应	客户端弹出错误信息窗口
主界面是否稳定	登录完成后, 是否出现主界面且主界面是否稳定	主界面在不操作的情况下可以出现且能稳定
好友列表能否正常显示	登录完成后, 切换至好友列表界面, 观察是否正常	好友列表正常显示且没有错误
群列表能否正常显示	显示登录界面后, 切换到群列表界面, 观察是否正常	群列表正常显示且没有错误
是否能发送私聊请求	点击好友信息部分, 右端出现与该好友聊天的历史记录	记录显示正常
是否能够发送添加好友请求	点击添加好友按钮, 输入 id, 可以发送添加好友请求	服务端收到添加好友请求
是否可以发送改变头像的信息	客户端点击头像, 可以发送改变头像信息	服务端收到改变头像信息
是否能够更新头像	服务端收到改变头像的信息后, 发送一个 ID, 客户端的头像可以变为指定 ID 的图片	客户端的头像改变
服务端能够返回新的好友列表	服务端收到刷新好友列表的消息后, 返回新的好友列表	客户端收到新的好友列表
客户端能否将信息发至服务端	客户端点击“发送”按钮, 服务端可以收到客户端发送的信息	服务端成功接到客户端发出的消息

服务端能否将一个客户端发送的消息发到另一个客户端	服务端可以将收到的客户端的信息转发到另一个客户端上	客户端可以收到服务端发出的另一个客户端的信息
客户端是否可以发送表情	客户端点击表情按钮选择表情, 另一个客户端可以收到表情	另一个客户端可以接收到表情
客户端是否可以发送加入群组请求	客户端点击加入群组, 输入群组名字, 可以加入群组	服务端接收到客户端发出的加入群组信息
服务端能否发送更新群组列表的信息	服务端发送更新群组列表信息, 客户端可以接到这个消息	客户端的群组列表更新
客户端是否能够发送创建群组请求	客户端点击创建群组信息, 可以被服务端接受	服务端收到客户端发出的创建群组信息
客户端能否在群组中发送消息	客户端发送消息被服务器接收后, 消息可以被服务器发送到信息栏	客户端的历史信息的位置出现客户端发送的消息
服务器可以对客户端发送的不同信息进行不同处理	接收到客户端的信息后, 服务器可以完成不同的响应步骤	客户端可以得到相应的回应
服务器端可以与客户端连接, 连接不成功返回错误信息	客户端可以连接服务端, 若连接错误则会弹出错误信息	客户端可以收到服务端的相应信息

6 心得体会

本次专业实习让我真切地体会到了短时间内高强度开发软件的感受。一方面确实时间紧, 任务重, 经常熬夜写代码。另一方面, 我也感受到了满满的收获。在这次实践中, 我完整地走完了一个软件的开发流程, 系统地考虑了软件的架构方式、模块分解以及模块之间的关联分析。在实际的开发过程中遇到了很多意想不到的问题, 在不断地查找资料、分析资料、尝试解决方法的探索过程中, 我也积累了很多的开发和调试的宝贵经验。同时, 在与其他组员的密切合作中, 我的团队协作意识和沟通能力也得到了提升。最后, 感谢崔老师对我的问题的及时回复和耐心解答!