AirChat 即时聊天软件

——需求分析与软件设计

学生姓名	
学号	
专业班级	
课程	学年设计
指导老师	
成绩	

目　录

1. 文档简介

1.1. 文档目的

本文档将描述实现一个类似于 QQ 的简单的即时通讯软件（即：AirChat）的前端和后端的需求分析和架构设计的主要细节以及相应的模型。

1.2. 参考文档

GB856T—88

1.3. 术语与缩写

表 1 术语与缩写

术语、缩写、符号	解释
ORM	Object Relational Mapping(对象关系映射)

2. 产品概述

2.1. 产品简介

本产品将基于 Qt 的 TCP 套接字实现具有登录注册、添加好友、发送文本信息等基本功能的即时聊天软件——AirChat

2.2. 产品范围

本产品的客户端将实现邮箱注册，账号登录，好友添加三个基本的用户管理操作，以及发送文本消息的基本即时通讯功能。

本产品的服务端将以控制台的形式实现。服务端将提供 AirChat 消息服务器的基本的业务处理和转发消息功能。

2.3. 运行环境

表 2 运行环境要求

产品名称	运行环境要求
AirChat 客户端	Win7，Win10 及以上的 x64 平台
AirChat 服务端	Windows Server 2019 标准版 64 位 具有 VCRUNTIME140.dll 及以上运行时环境 具有 MySQL8.0 及以上数据库

3. 需求分析篇

3.1. 客户端

3.1.1. 产品功能需求

3.1.1.1. 用例图

AirChat 的客户端将主要具备邮箱注册，账号登录，添加好友，发送文本信息的功能。

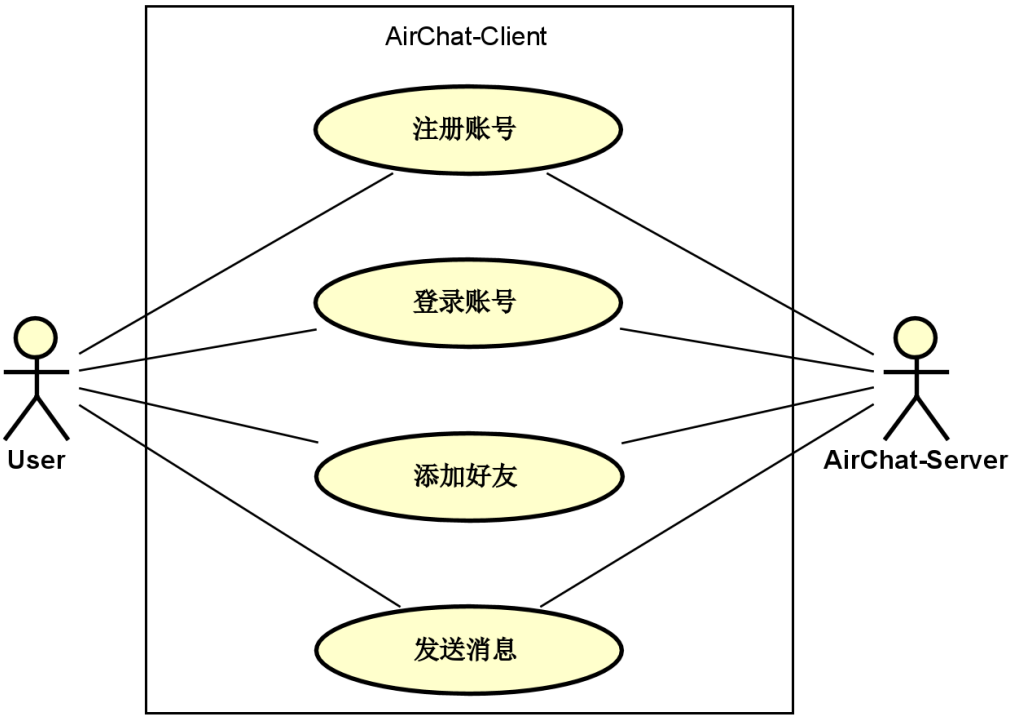

图 1 AirChat 客户端用例图

3.1.1.2. 用例描述

表 3 “注册账号”用例成文描述

ID	RegisterReq-Client	
用例名称	注册账号	
参与者	用户，服务器	
概述	用户可以使用邮箱获取验证码，并设置密码，从而完成注册	
优先级	最高	
前置条件	1. 邮箱正确 2. 邮箱未注册	
后置条件	1. 系统显示注册最终结果	
主要流程		
参与者		系统
1. 用户输入邮箱号		
2. 用户请求验证码		3. 系统发送验证码请求[E3.1]
4. 服务器发送邮件		
5. 用户提交注册请求		6. 系统发送注册请求
7. 服务器回送注册结果		8. 系统显示注册结果[E7.1][E7.2]
异常：E3.1 邮箱格式错误		
		E3.1.1. 系统提示用户
E3.1.2. 用例结束		
异常：E7.1 验证码错误		
		E7.1.1. 系统提示用户
E7.1.2. 用例结束		
异常：E7.2 邮箱已注册错误		
		E7.2.1. 系统提示用户
E7.2.2. 用例结束		

表 4 “登录账号”用例成文描述

ID	SignIn-Client	
用例名称	登录账号	
参与者	用户，服务器	
概述	用户使用账号，并输入密码，从而完成登录	
优先级	最高	
前置条件	1. 邮箱或账号存在	
后置条件	1. 系统跳转至聊天界面或提示登录失败	
主要流程		
参与者		系统
1. 用户输入账号密码登录		2. 系统发出登录请求

3. 服务器回送登录验证数据	4. 系统跳转至聊天界面[E4.1]
异常：E4.1 登录验证失败	
	4.1.1. 系统提示用户
4.1.2. 用例结束	

表 5“添加好友”用例成文描述

ID	AddFriend-Client
用例名称	添加好友
参与者	用户，服务器
概述	用户可以通过账号添加好友
优先级	较高
前置条件	1. 已成功登录
后置条件	1. 系统更新好友列表或提示添加失败
主要流程	
参与者	系统
1. 用户提交好友账号	2. 系统发出添加请求[E2.1]
3. 服务器处理请求并回送数据	4. 系统更新好友列表[E4.1]
异常：E2.1 添加目标已是好友	
	2.1.1. 系统提示用户
2.1.2. 用例结束	
异常：E4.1 目标账号不存在	
	4.1.1. 系统提示用户
4.1.2. 用例结束	

表 6 “发送消息”用例成文描述

ID	Send-Mesg-Client
用例名称	发送消息
参与者	用户，服务器
概述	用户可以发送文本信息到指定的账号
优先级	最高
前置条件	1. 已成功登录
后置条件	1. 系统显示发送的消息 2. 服务器转发消息
主要流程	
参与者	系统
1. 用户发送消息	2. 系统发出消息报文
	3. 系统在对应聊天显示框显示消息
4. 服务器转发消息	

3.1.1.3. 用例序列图

根据上述提到的 AirChat 客户端的用例以及其成文用例的描述，识别基本的系统操作如下。

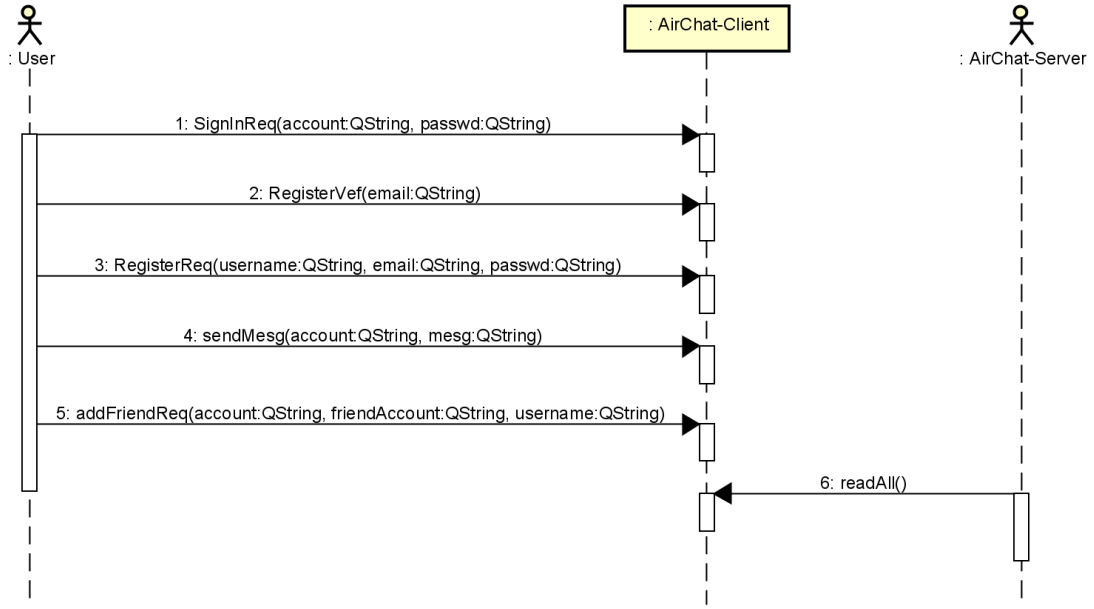

图 2 AirChat 客户端用例序列图

3.1.1.4. 系统规约

根据上述描述的系统操作，其每一个的定义与约束如下表所示：

表 7 signInReq 系统操作规约定义

系统操作名称	signInReq(account:QString, passwd:QString)
目的	验证登录基本信息：账号、密码
前置条件	1. TCP 套接字对象的状态为已连接服务器
后置条件	1. 输入框控件对象的值为所填的值 2. TCP 套接字对象发送了登录请求消息

表 8 registerVef 系统操作规约定义

系统操作名称	registerVef (email:QString)
目的	获取邮箱验证码
前置条件	1. TCP 套接字对象的状态为已连接服务器
后置条件	1. 输入框控件对象的值为所填的值 2. 获取控件状态更新为不可用 3. 30 秒定时器创建并启动 4. TCP 套接字对象发送了验证码请求消息

表 9 registerReq 系统操作规约定义

系统操作名称	registerReq (username:QString, email:QString, passwd:QString)
目的	进行邮箱注册
前置条件	1. TCP 套接字对象的状态为已连接服务器
后置条件	1. 输入框控件对象的值为所填的值 2. TCP 套接字对象发送了注册请求消息

表 10 sendMesg 系统操作规约定义

系统操作名称	sendMesg (account:QString, mesg:QString)
目的	向好友发送消息
前置条件	1. TCP 套接字对象的状态为已连接服务器 2. 已登录系统，主聊天界面对象状态为 show
后置条件	1. 输入框控件的值清空 2. 聊天显示窗口的值追加输入的值 3. TCP 套接字对象发送了消息

表 11 addFriendReq 系统操作规约定义

系统操作名称	addFriendReq (account:QString, friendAccount:QString, username:QString)
目的	根据账号添加好友
前置条件	1. TCP 套接字对象的状态为已连接服务器 2. 已登录系统，主聊天界面对象状态为 show
后置条件	1. TCP 套接字对象发送了添加好友请求消息

3.1.2. 外部接口需求

3.1.2.1. 用户界面

客户端的界面跳转逻辑如下图所示：

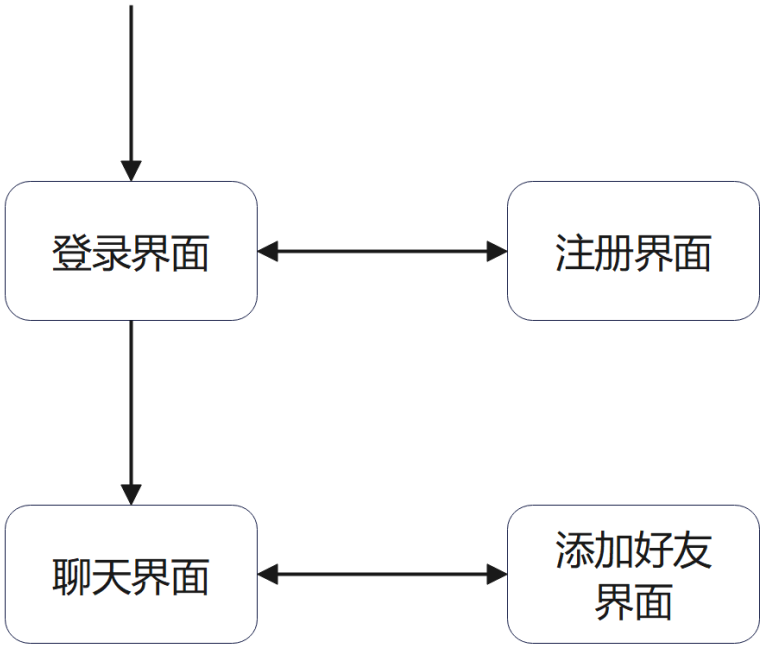

图 3 AirChat 客户端界面逻辑图

登录界面描述：

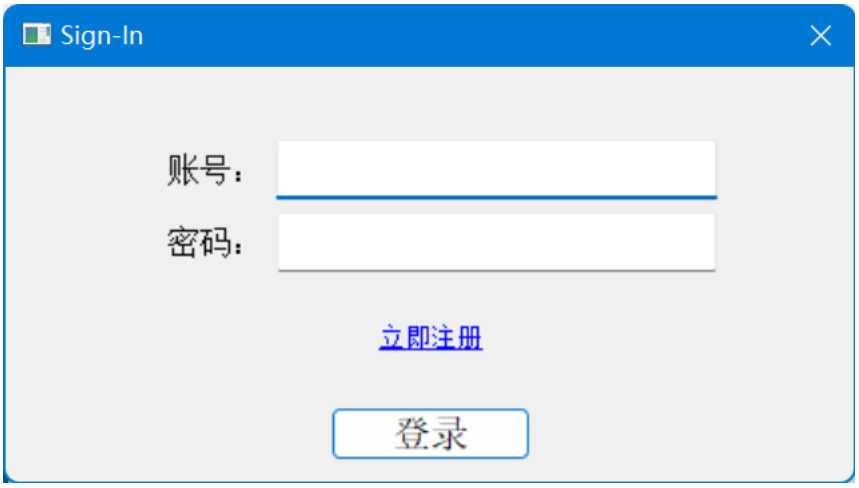

图 4 AirChat 客户端登录界面

可以点击“立即注册”跳转至注册界面

注册界面描述：

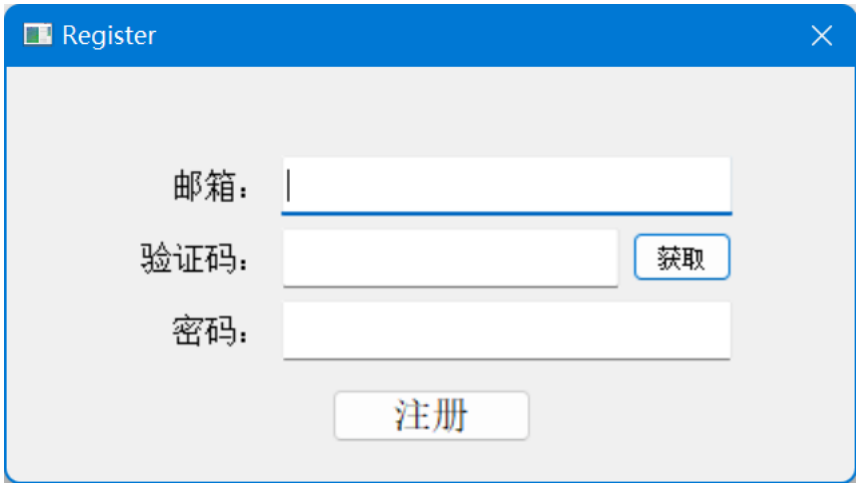

图 5 AirChat 客户端注册界面

点击“获取”控件获取验证码，点击“注册”控件发起注册请求

聊天界面描述：

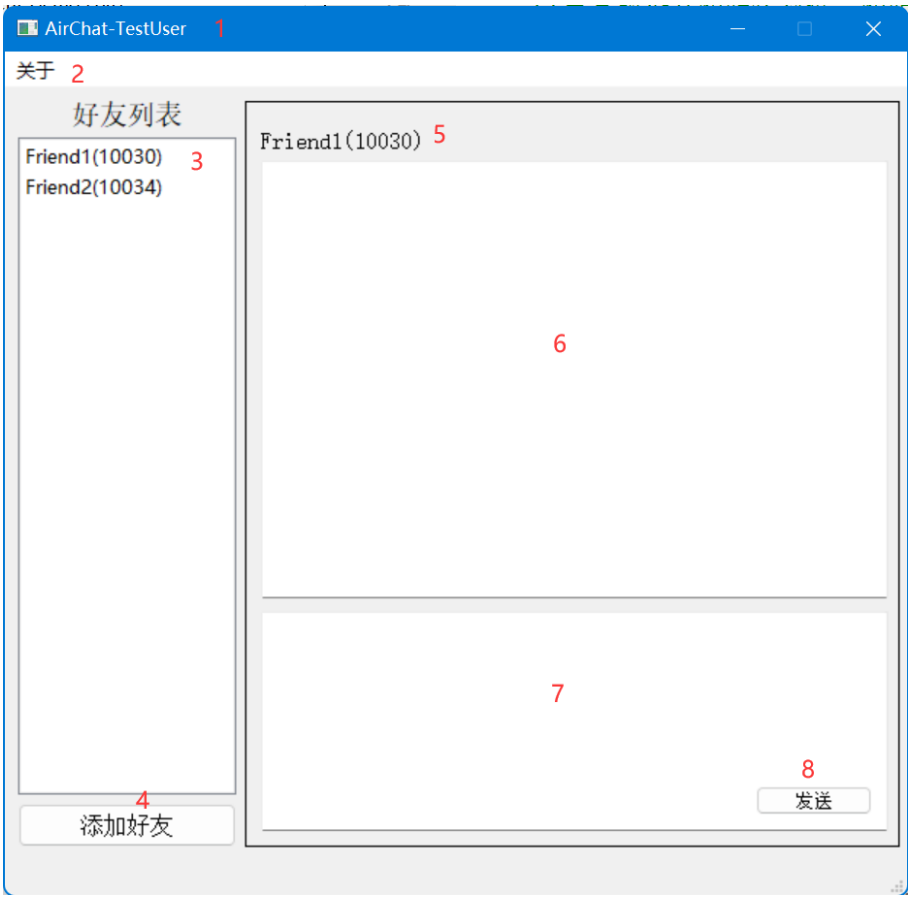

图 6 AirChat 客户端聊天界面

1：显示软件名称、登录用户名

2：显示软件制作信息

3：显示好友列表，点击好友切换至对应的聊天窗口

4：添加好友控件，点击控件弹出添加好友对话框

5：显示当前聊天界面目标好友信息

6：显示接收和发出的聊天内容

7：输入待发送消息

8：发送消息控件，点击发出消息

添加好友界面描述：

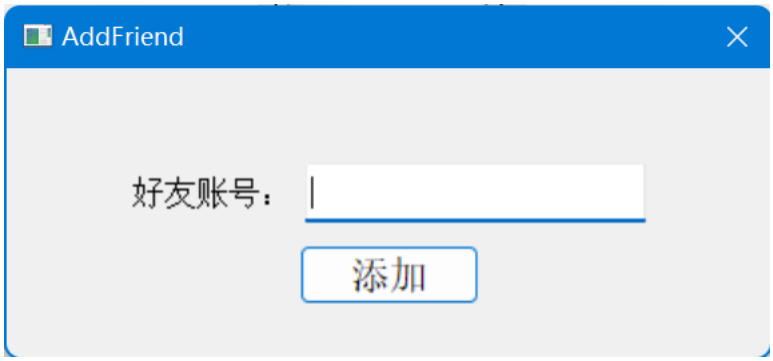

图 7 AirChat 客户端添加好友界面

通过输入好友账号来添加好友

异常界面描述：

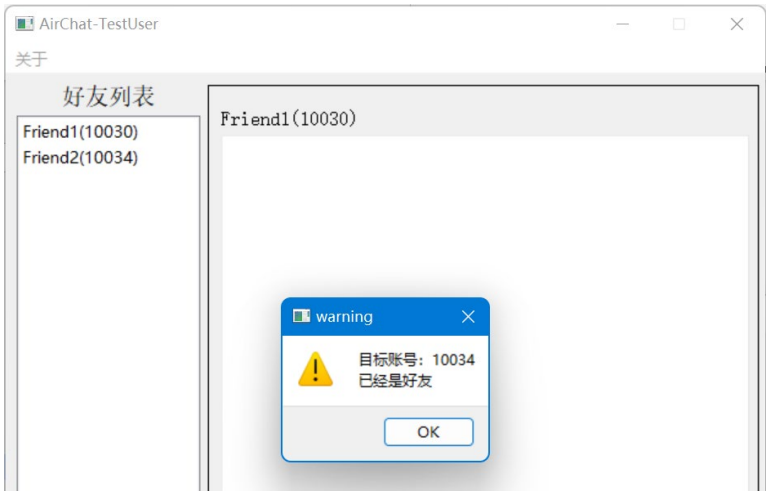

图 8 AirChat 客户端好友添加异常提示

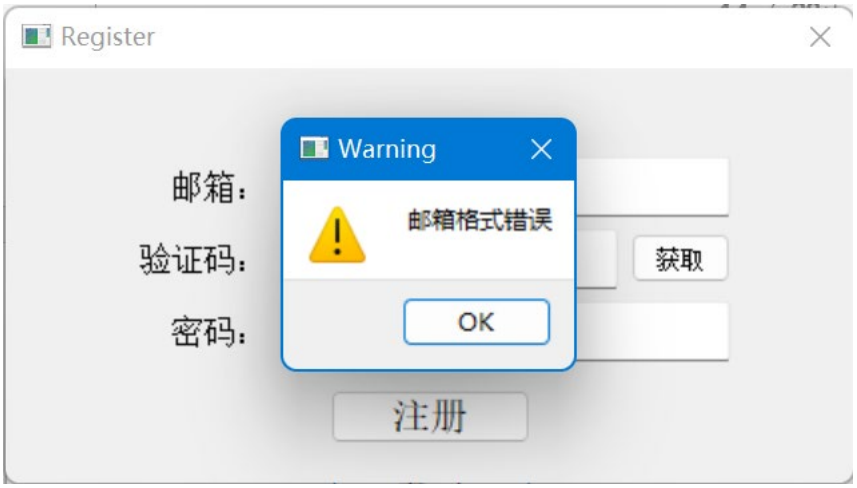

图 9 AirChat 客户端邮箱格式异常提示

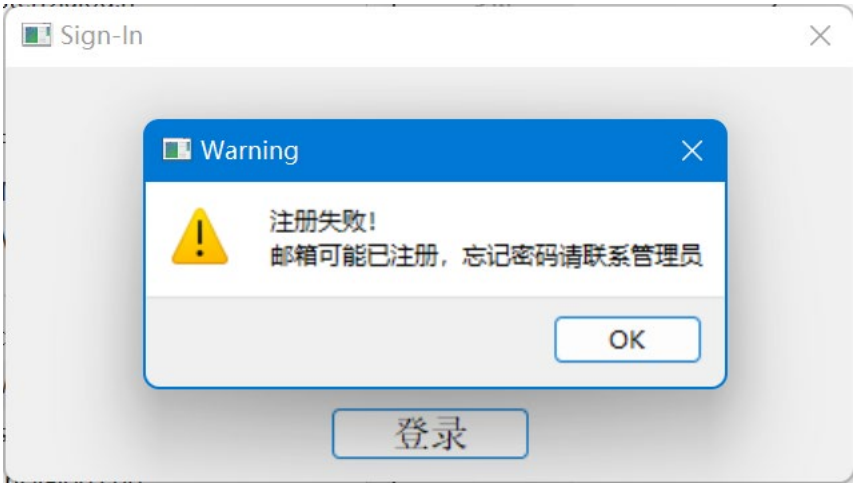

图 10 AirChat 客户端注册失败提示

3.1.2.2. 软件接口

无需其他协作软件

3.1.2.3. 通讯接口

除去用户使用 UI 界面对客户端传递数据外，其他所有的对外通讯均为通过 TCP 套接字完成的网络通信。

该 TCP 通讯套接字不与界面类处于同一线程以避免界面卡死，并且由一个统一的通讯接口类进行管理。

3.1.3. 产品非功能需求

表 12 客户端非功能需求

项目	要求	解决方案
登录时长	5s 内响应	超时提示
注册时长	15s 内响应 （受邮件服务影响）	超时提示
消息到达处理时长	1s 内响应	使用 Hash 表查找来快速定位对应的聊天界面
消息传输	加密传输	使用 AES 加密报文 RSA 加密 AES 密钥

3.2. 服务端

3.2.1. 产品功能需求

3.2.1.1. 用例图

AirChat 的服务器端与客户端相对应，实现对注册的验证，验证码请求的响应，登录账号的验证，好友添加处理，以及将发送的消息转发到对应的在线客户端。目前只设计了客户端双方都在线是的消息转发，没有设计消息的缓存。

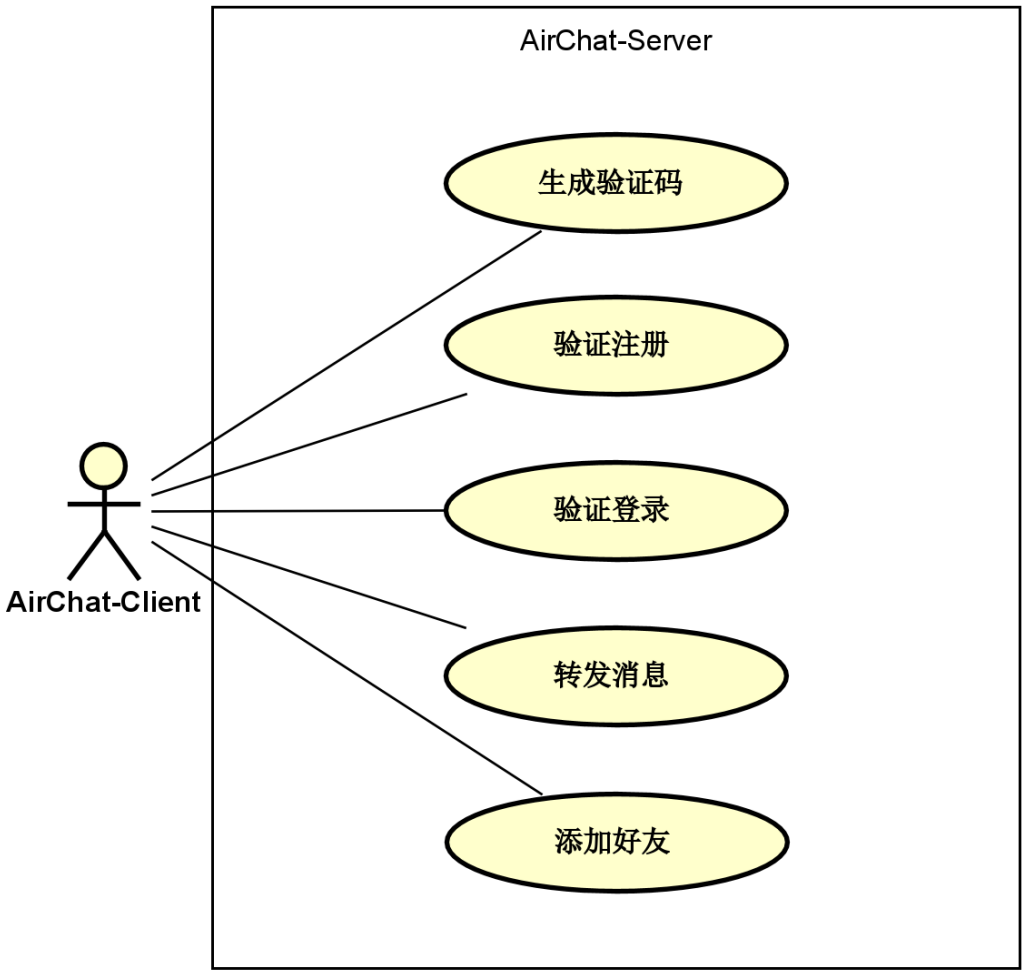

图 11 AirChat 服务端用例图

3.2.1.2. 用例描述

表 13 “生成验证码”用例成文描述

ID	RegisterVef-Server	
用例名称	生成验证码	
参与者	**AirChat 客户端**	
概述	系统生成随机验证码存储到数据库并发送邮件到目标邮箱	
优先级	最高	
前置条件	1. 消息类型标识符为登录请求 2. 报文格式正确	
后置条件	1. 记录到数据库 2. 发送邮件	
主要流程		
参与者		**系统**
1. 发送验证码请求消息		2. 系统解析消息[E2.1]
		3. 系统生成验证码
		4. 系统插入到数据库[E4.1]
		5. 系统发送邮件到目标邮箱
异常：E2.1 报文格式错误或不是登录请求		
		2.1.1. 用例结束
异常：E4.1 数据库已有目标邮箱的表项		
		4.1.1. 系统更新该数据条目

表 14“验证注册”用例成文描述

ID	RegisterReq-Server	
用例名称	验证注册	
参与者	**AirChat 客户端**	
概述	系统验证注册信息的正确性（邮箱与验证码是否匹配，邮箱是否未注册），进而生成账号，并写入数据库	
优先级	最高	
前置条件	1. 消息类型标识符为注册请求 2. 报文格式正确	
后置条件	1. 相应的邮箱、用户名、账号、密码写入数据库 2. 发送注册成功消息	
主要流程		
参与者		**系统**
1. 发送注册请求报文到服务器		2. 系统解析报文[E2.1]
		3. 系统验证邮箱验证码[E3.1]
		4. 系统生成账号

	5. 系统验证邮箱[E5.1]
	6. 系统写入数据库
	7. 系统回送消息
异常：E2.1 报文格式错误或不是注册请求	
	2.1.1. 用例结束
异常：E3.1 邮箱验证码错误	
	3.1.1. 系统回送异常消息
	3.1.2. 用例结束
异常：E5.1 邮箱已注册	
	5.1.1. 系统回送异常消息
	5.1.2. 用例结束

表 15 “验证登录”用例成文描述

ID	SignIn-Server
用例名称	验证登录
参与者	**AirChat 客户端**
概述	系统验证客户端发送的账号和密码合法性，并回送该账户的初始化消息
优先级	最高
前置条件	1. 消息类型标识符为登录请求 2. 报文格式正确
后置条件	1. 发送登录成功消息及初始化消息
主要流程	
参与者	**系统**
1. 发送登录请求消息	2. 系统解析消息[E2.1]
	3. 系统核验账号和密码[E3.1]
	4. 系统查询账户基本信息
	5. 系统回送登陆成功消息及基本信息
异常：E2.1 报文格式错误或不是登录请求	
	2.1.1. 用例结束
异常：E3.1 账号密码不匹配或无此账号	
	3.1.1. 系统回送异常消息
	3.1.2. 用例结束

表 16 “转发消息”用例成文描述

ID	ForwardMesg-Server	
用例名称	转发消息	
参与者	**AirChat 客户端**	
概述	系统根据报文转发消息到目标客户端	
优先级	最高	
前置条件	1. 消息类型标识符为普通消息 2. 报文格式正确 3. 目标账户在线	
后置条件	1. 发送消息到目标账户的客户端	
主要流程		
参与者		系统
1. 发送普通消息		2. 系统解析消息[E2.1]
		3. 系统查询目标账户客户端 TCP 连接[E3.1]
		4. 系统向目标账户发送消息
异常：E2.1 报文格式错误或不是普通		
		2.1.1. 用例结束
异常：E3.1 目标账户不在线		
		3.1.1. 用例结束

表 17 “添加好友”用例成文描述

ID	AddFriend-Server	
用例名称	添加好友	
参与者	**AirChat 客户端**	
概述	系统根据添加好友请求添加好友关系到数据库，并返回客户端更新信息	
优先级	最高	
前置条件	1. 消息类型标识符为好友添加请求 2. 报文格式正确	
后置条件	1. 添加好友关系到数据库 2. 发送客户端更新信息	
主要流程		
参与者		系统
1. 发送好友添加请求		2. 系统解析消息[E2.1]
		3. 系统查询目标账户相关信息[E3.1]
		4. 系统更新数据库
		5. 系统回送更新信息

异常：E2.1 报文格式错误或不是好友添加请求	
	2.1.1. 用例结束
异常：E3.1 目标账户不存在	
	3.1.1. 系统回送异常消息
	3.1.2. 用例结束

3.2.1.3. 用例序列图

服务器不存在有意义的用例序列图，其与参与者的交互只是通过 TCP 通讯完成。

唯一的调用为 TCP 套接字读取到达消息的接口 readyRead()信号。

3.2.1.4. 系统规约

同上

3.2.2. 数据需求

3.2.2.1. 逻辑数据模型[1]

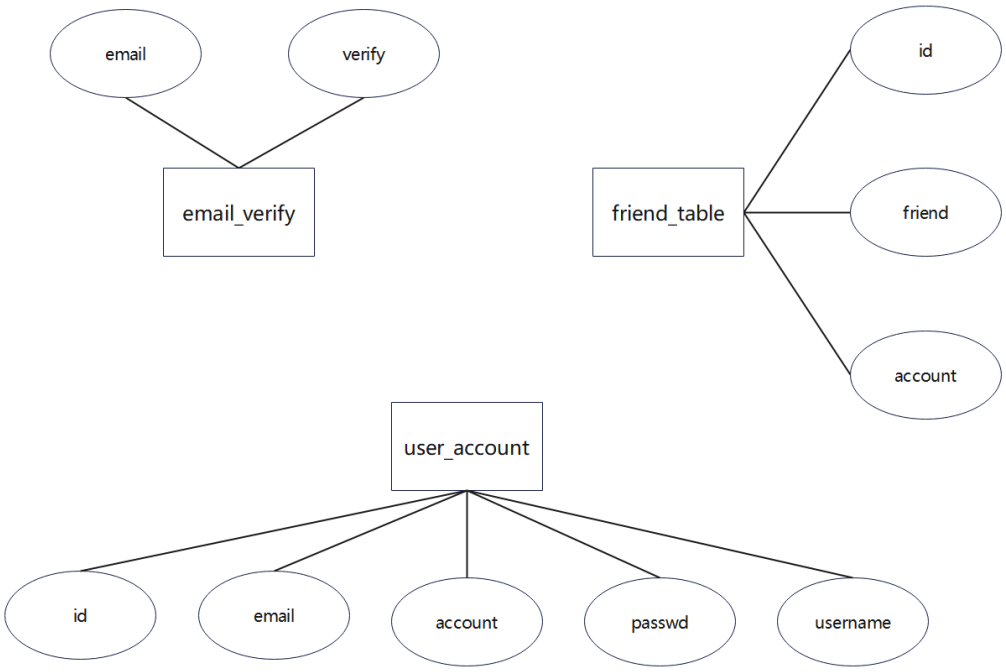

图 12 AirChat 服务端逻辑数据模型

3.2.2.2. 数据定义

表 18 email_verify 表定义

数据项名称	类型	长度	Not Null	主键	其他
email	varchar	50	√	√	
verify	varchar	6	√		唯一

[1] 由于时间和结束能力限制没有采用 C++的某个 ORM 框架来实现实体类，而是采用构造一个管理传统的 sql 查询类的方法实现，所以此处的逻辑数据模型为数据库的模型。

表 19 friend_table 表定义

数据项名称	类型	长度	Not Null	主键	其他
id	int		√	√	自动递增
account	varchar	15			
friend	varchar	15			

表 20 user_account 表定义

数据项名称	类型	长度	Not Null	主键	其他
id	int		√	√	自动递增
email	varchar	50	√		
account	varchar	15			唯一
passwd	varchar	15			
username	varchar	30			

3.2.3. 外部接口需求

3.2.3.1. 用户界面

控制台程序，无 GUI 界面。

3.2.3.2. 软件接口

无需其他协作软件

3.2.3.3. 通讯接口

所有的对外通讯均为通过 TCP 套接字完成的网络通信。

每一个客户端的链接有一个单独的子线程管理。

3.2.4. 产品非功能需求

表 21 服务端非功能需求

项目	要求	解决方案
登录请求处理时长	2s 内响应	数据查询设计
注册时长	12s 内发送邮件 （受邮件服务影响）	无
消息到达处理时长	1s 内响应	使用 Hash 表查找来快速定位在线的目标客户端的 TCP
消息传输	加密传输	使用 AES 加密报文 RSA 加密 AES 密钥
数据库	登录安全	密码加密存储 登录强制使用 SSL 验证

4. 设计架构篇

4.1. 通讯设计

客户端与服务端之间的通讯使用 Qt 封装的 TCP 套接字进行通讯。通讯报文按照如下格式规定。

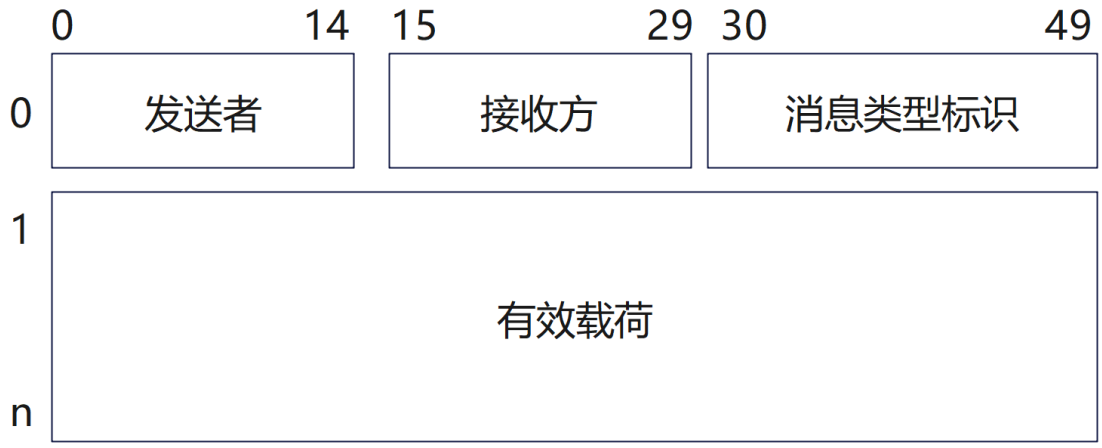

图 13 通讯报文格式

报文由前 50 位（非计算机二进制位，按一个 utf-8 的完整字符为一位计算）的报文头和有效载荷构成。

报文的各个部分的详细说明如下：

表 22 报文格式详细说明

部分	详细说明
发送者	由 15 位的发送方账号或系统消息标识（Server）构成，位数不足 0 填充
接受者	由 15 位的接受方账号或系统处理标识（Server）构成，位数不足 0 填充
消息类型标识	默认为 mesg（20 位，不足补 0）； 当发送者或接受者任意一方为标识 Server 时，该标识不为 mesg，为其他特殊含义标识（见下表）

表 23 消息类型标识取值及解释

消息类型标识取值	解释
Mesg	普通消息标识
Register-Req	账号注册请求标识
Register-Vef	注册验证码请求标识
Register-Suc	注册成功标识
Register-Faild	注册失败标识
SignIn-Req	登录请求标识
SignIn-Init	登录成功初始化数据包标识
SignIn-Faild	登录失败标识
Friend-Add	添加好友请求标识
Friend-New	新增好友数据包标识
Friend-Faild	添加好友失败标识

在注册登录时，客户端将与服务端建立 TCP 长连接。TCP 建立后服务端将向客户端发送登录账户的基本信息（好友信息及在线状态、缓存的消息）.

在进行消息发送（聊天）时，客户端将通过 TCP 将消息发送到服务端。服务端将根据报文头的接收者消息，通过 TCP 来转发消息。

4.2. 客户端设计

4.2.1. 概念类图

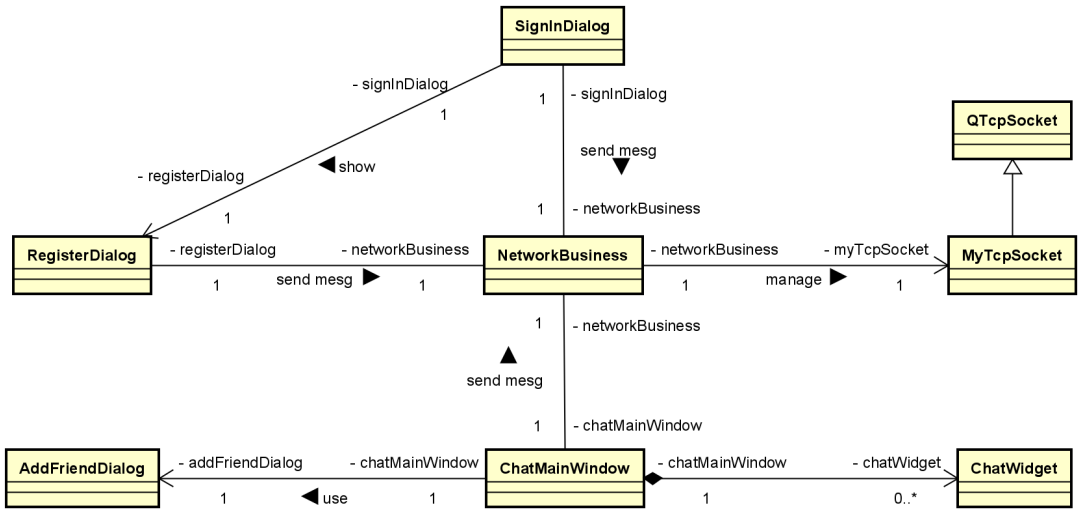

图 14 客户端概念类图

其中 SignInDialog、RegisterDialog、ChatMainWindow 为主要界面类（边界类）负责处理用户的交互以及简单的值的合法性判定（邮箱格式、密码长度、密码复杂度、空值判定）。

AddFriendDialog 和 ChatWidget 同样也是界面类。其中 ChatWidget 作为 ChatMainWindow 成员变量不单独存在。而 AddFriendDialog 属于自定义的有一定复杂业务处理的对话框界面作为 ChatMainWindow 正常使用的基本依赖类。

另外，MyTcpSocket 是 QT 框架提供的 TCP 套接字的封装类 QTcpSocket 的自定义继承类，对一些必要的输入输出和信号槽做出了一定的业务逻辑处理和优化。并将其置于 NetworkBusinesses 类的成员变量来被更好的分成管理。

4.2.2. 对象序列图

“注册账号”

用户通过界面类（边界类）RegisterDialog 与系统交互，传递注册数据。RegisterDialog 类对数据进行简单过滤，使用正则表达式验证邮箱、密码等格式。再提交至 NetworkBusiness 处理生成待发送的格式化的数据报文。NetworkBusiness 控制其成员变量 MyTcpSocket 发送报文到服务器。

如下图所示：

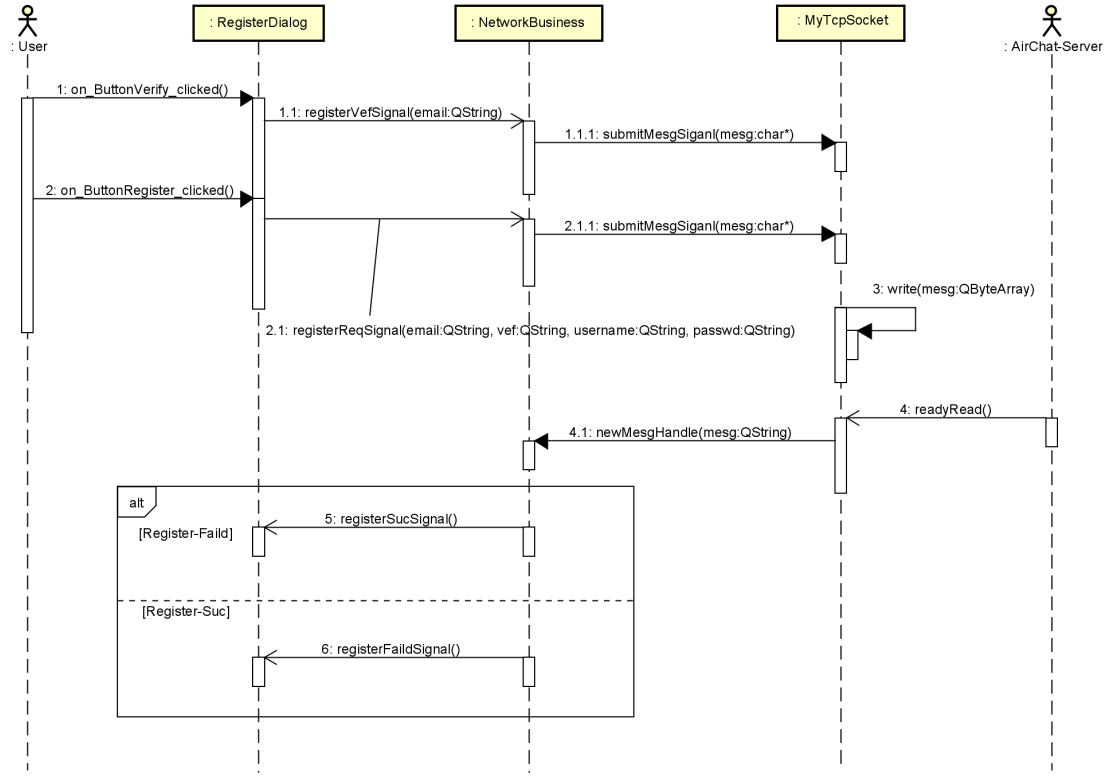

图 15 “注册账号”用例的对象序列图

“登录账号”

用户通过界面类（边界类）SignInDialog 与系统交互，传递登录数据。SignInDialog 类对数据进行简单的判空处理。再提交至 NetworkBusiness 处理生成待发送的格式化的数据报文。NetworkBusiness 控制其成员变量 MyTcpSocket 发送报文到服务器。NetworkBusiness 处理服务器回送数据后，提示用户登录失败或向 ChatMainWindow 类发出初始化信号。使得聊天主界面显示，并隐藏登录窗口。

如下图所示：

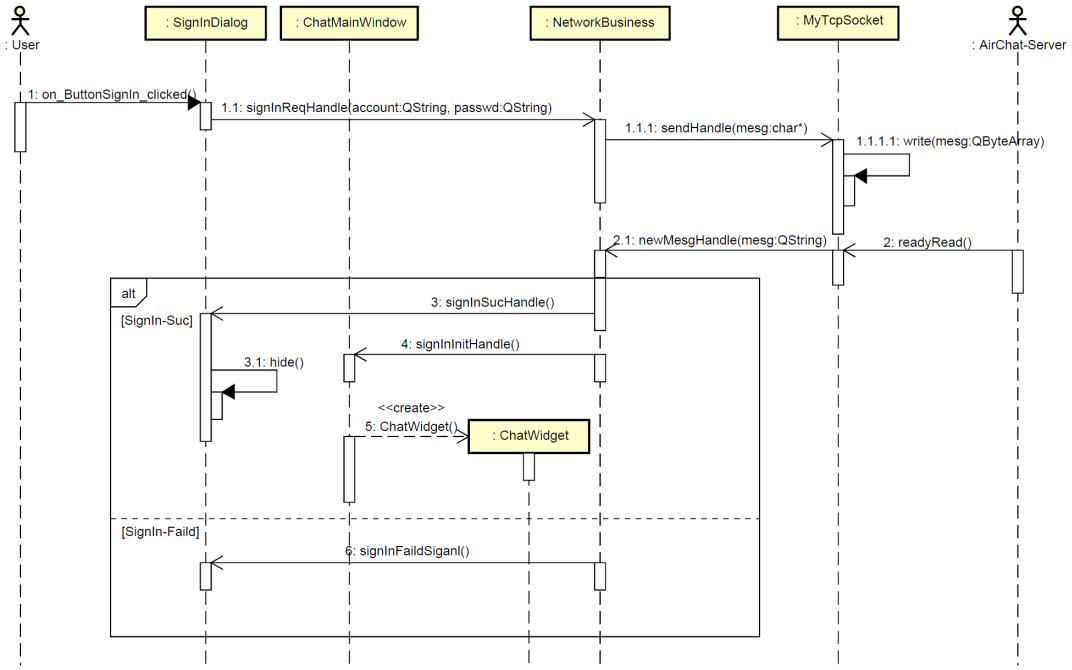

图 16 “登录账号”用例的对象序列图

“添加好友”

用户通过界面类（边界类）ChatMainWindow 与系统交互，传递好友账号数据。ChatMainWindow 类对数据进行简单的判断处理（不是已为好友的账号，不是自己的账号）。再提交至 NetworkBusiness 处理生成待发送的格式化的数据报文。NetworkBusiness 控制其成员变量 MyTcpSocket 发送报文到服务器。NetworkBusiness 处理服务器回送数据后，提示用户添加失败或向 ChatMainWindow 类发出添加信号。使得聊天主界面的好友列表添加新的好友。

如下图所示：

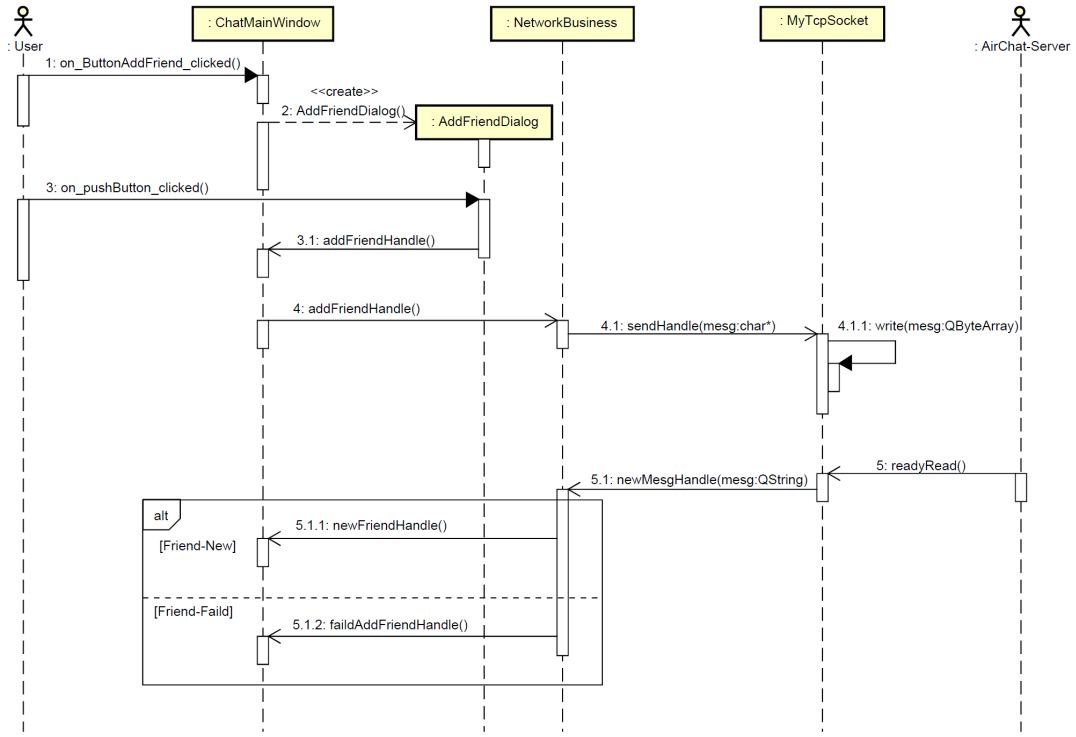

图 17 “添加好友”用例的对象序列图

“发送消息”

用户通过界面类（边界类）ChatMainWindow 与系统交互，传递好友账号数据。ChatMainWindow 类对数据进行简单的判断处理（不是已为好友的账号，不是自己的账号）。再提交至 NetworkBusiness 处理生成待发送的格式化的数据报文。NetworkBusiness 控制其成员变量 MyTcpSocket 发送报文到服务器。NetworkBusiness 处理服务器回送数据后，提示用户添加失败或向 ChatMainWindow 类发出添加信号。使得聊天主界面的好友列表添加新的好友。

如下图所示：

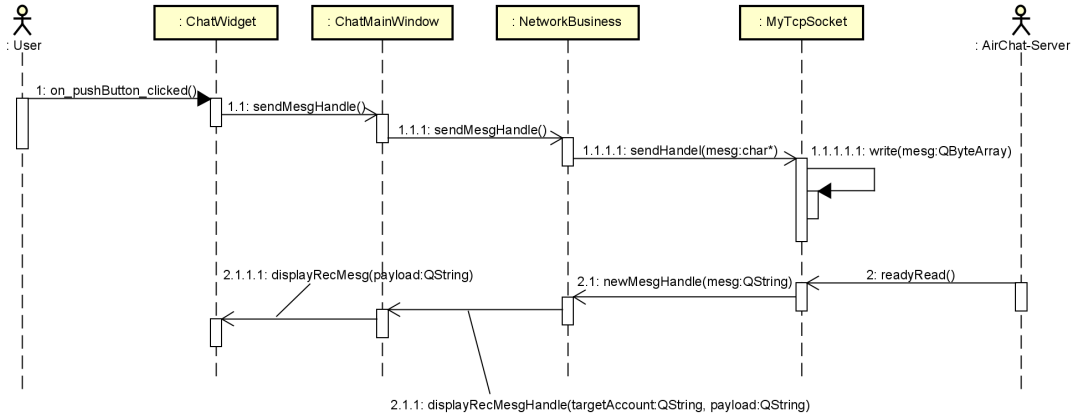

图 18 “发送消息”用例的对象序列图

4.2.3. 核心设计要点

UI 与通讯分离：

将 TCP 通讯套接字与界面交互分离设计避免网络等待阻塞函数导致界面出现未响应。

代码片段 1 通讯子线程启动代码片段

```
//网络通讯子线程
NetworkBusiness *network = new NetworkBusiness();
QThread *thread = new QThread();
network->moveToThread(thread);
/* 信号槽连接代码 */


//开启子线程
QObject::connect(network, &NetworkBusiness::start, network,
&NetworkBusiness::mainBusiness);
thread->start();
```

设置应用属性，使其可以适配不同显示分辨率和系统缩放并保持图像和文字的高清

代码片段 2 应用运行属性设置参数

```
[Platforms]
WindowsArguments=fontengine=freetype
WindowsArguments=dpiawareness=0
```

4.3. 服务端设计

4.3.1. 概念类图

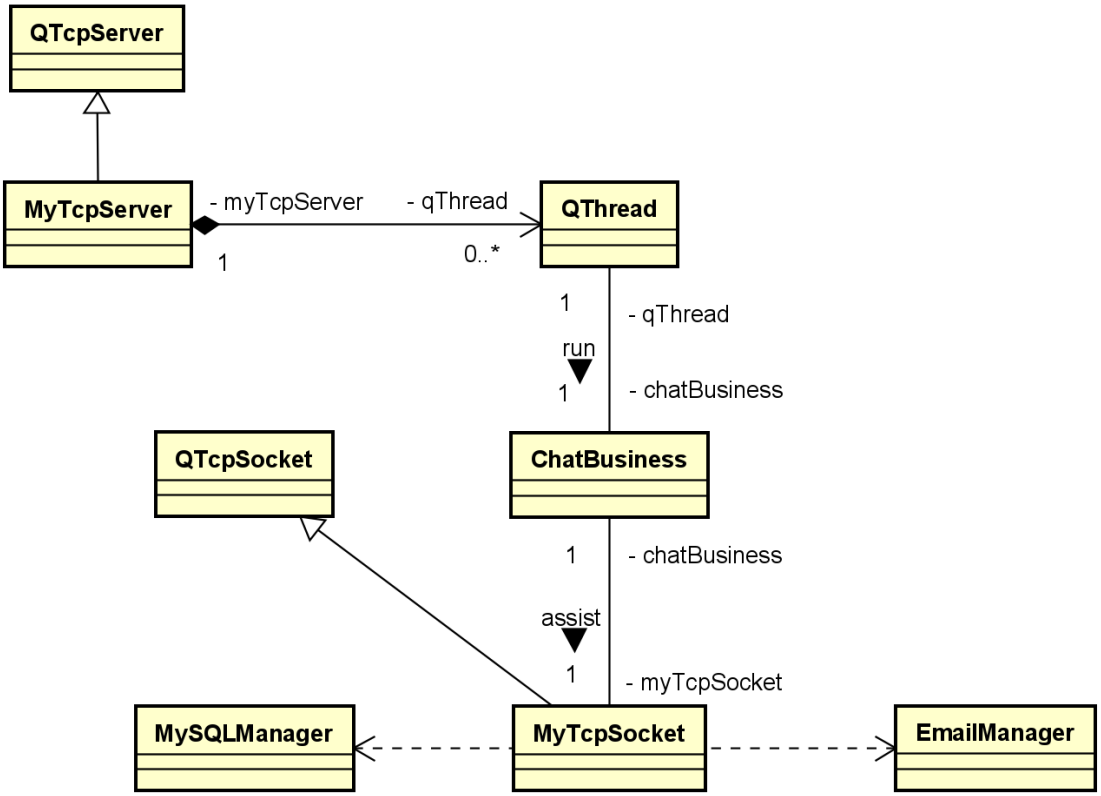

图 19 服务端概念类图

4.3.2. 对象序列图

“生成验证码”

MyTcpSocket 解析收到的报文，并生成随机验证码。然后交给 MySQLManager 设置验证码并创建邮件发送服务发送邮件。

如下图所示：

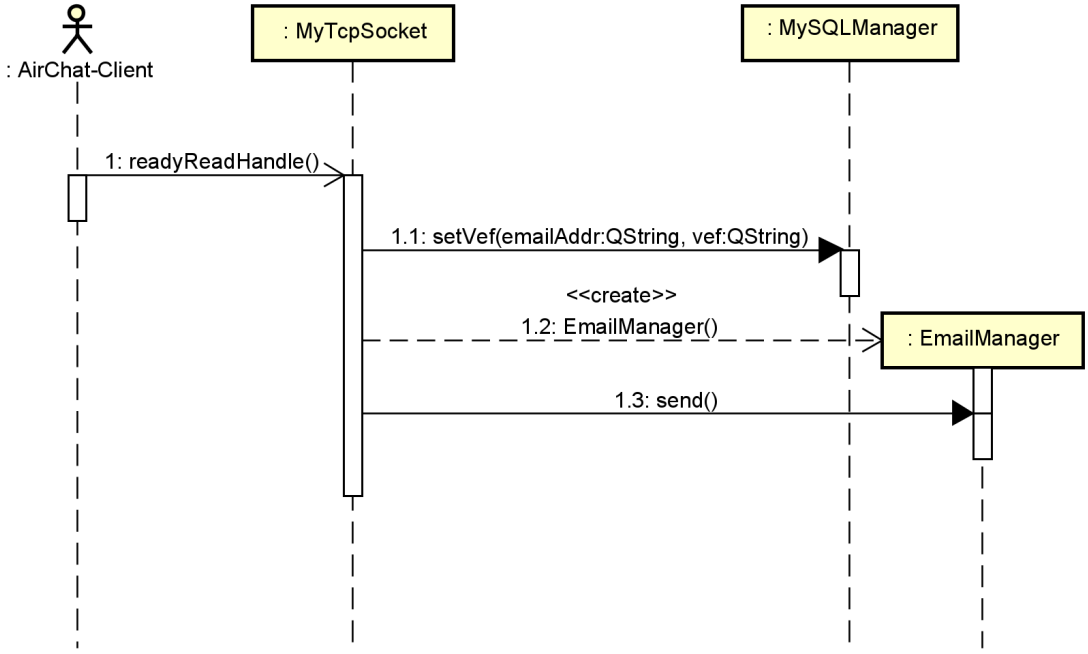

图 20 “生成验证码”用例的对象序列图

“验证注册”

MyTcpSocket 解析收到的报文，并生成随机验证码。然后交给 MySQLManager 生成并设置验证码,然后创建邮件发送服务并发送邮件告知账户密码。若验证不匹配则发送异常消息

如下图所示：

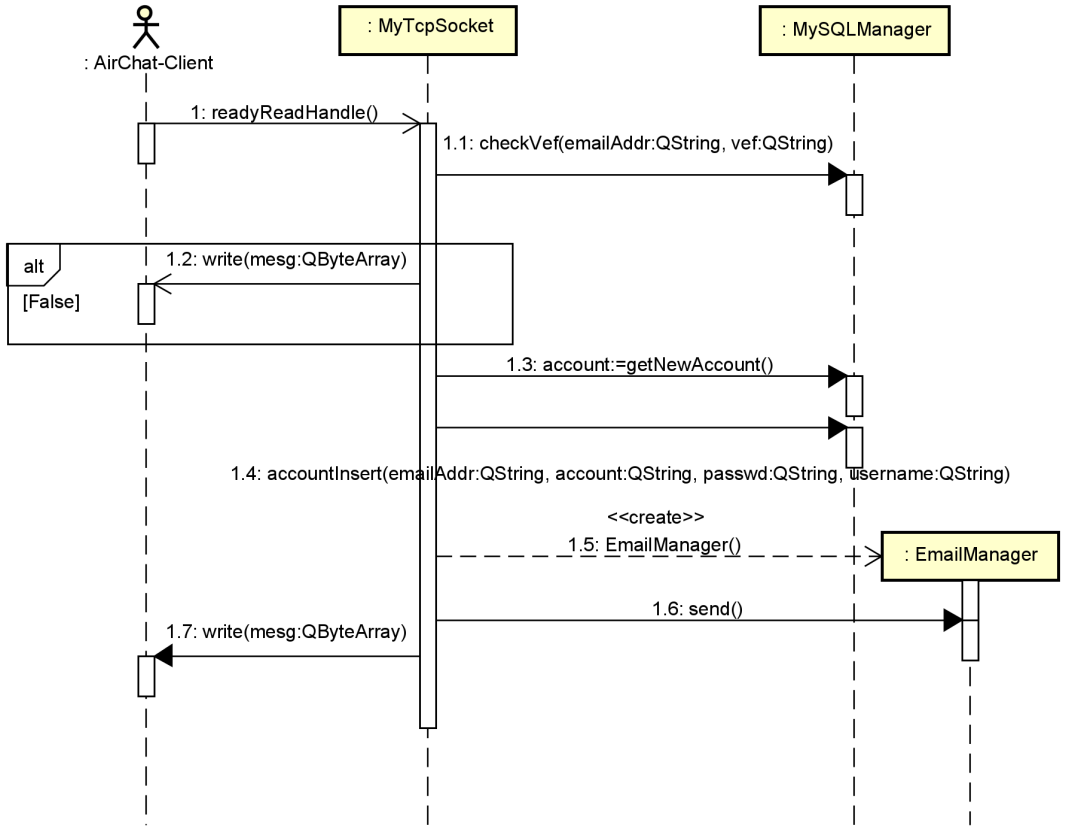

图 21 “验证注册”用例的对象序列图

“验证登录”

MyTcpSocket 解析收到的报文，通过 MySQLManager 先确认账号存在，再去查询并验证密码是否正确。成功验证将层层向上发出信号使得顶层的 MyTcpServer 更新对链接客户端的 TCP 套接字和子线程的 Hash 索引。失败则回送异常消息。

如下图所示：

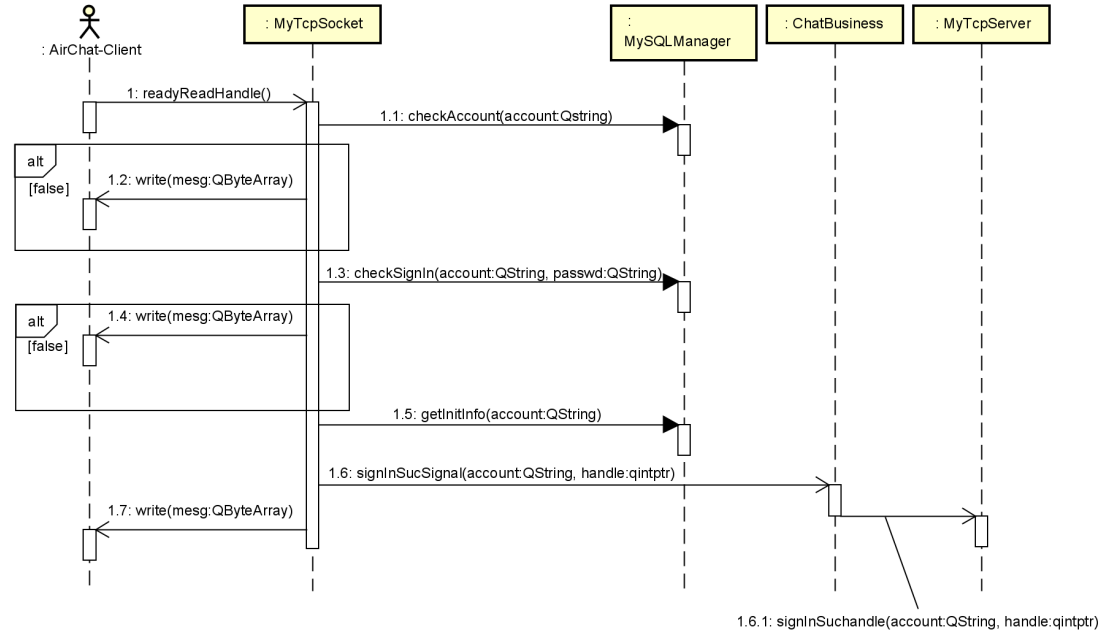

图 22 “验证登录”用例的对象序列图

“转发消息”

MyTcpSocket 解析收到的报文，发送到自己的上层 ChatBusiness 请求转发。ChatBusiness 将其提交到 MyTcpServer，MyTcpServer 根据 Hash 索引转发到对应的 ChatBusiness，如果不存在则放弃。

如下图所示：

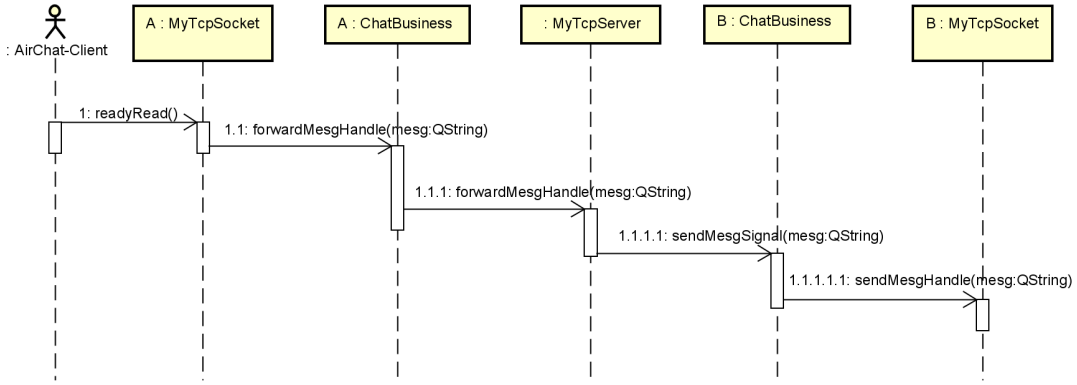

图 23 “转发消息”用例的对象序列图

“添加好友”

MyTcpSocket 解析收到的报文，调用 MySQLManager 验证账号是否存在，不存在则回送异常消息。若存在则更新数据库中的好友关系，并回送好友信息。MyTcpSocket 再将自己的账号消息请求上层转发到目标好友的客户端。若目标好友不在线，则无需传输在线更新数据。

如下图所示：

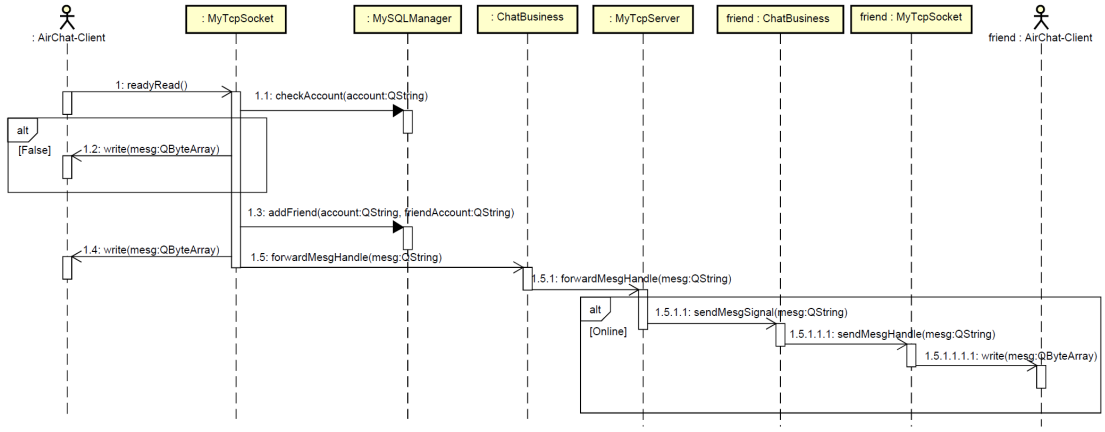

图 24 “添加好友”用例的对象序列图

“TCP 链接过程”

MyTcpServer 重写了 QTcpServer 的 incomingConnection 方法，使得新链接到达时直接创建 ChatBusiness 业务类，并将其移入新的子线程运行。

如下图所示：

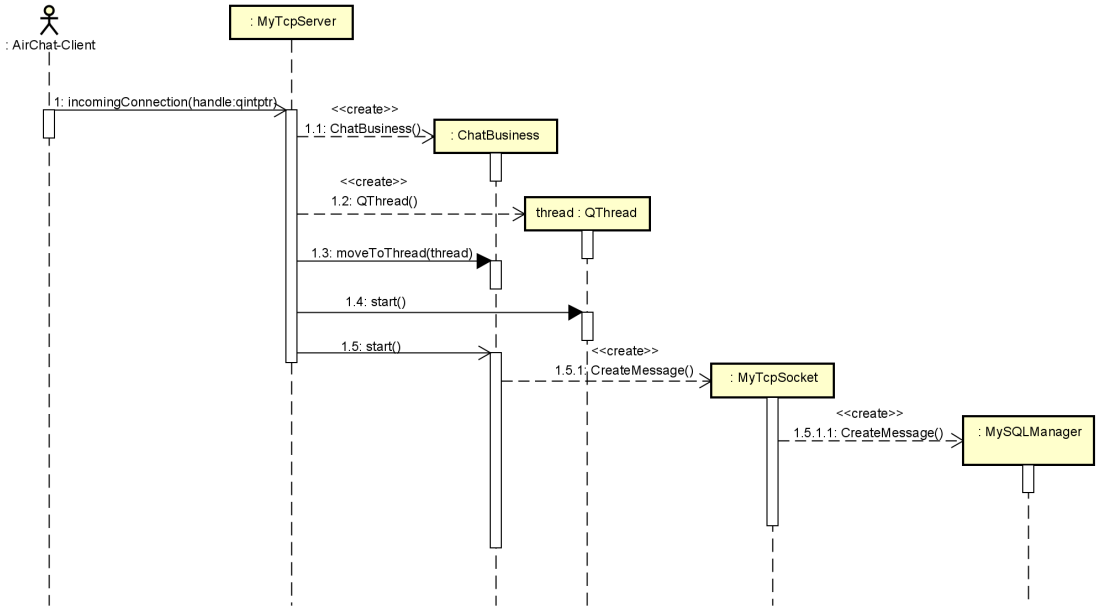

图 25 “TCP 连接过程”对象序列图

4.3.3. 核心设计要点

多线程服务器

为了使得多个客户端通过 TCP 套接字接入，继承了 QTcpServer 类并重写了 incomingConnection 方法，使得每一个新的 TCP 接入都是由一个单独的线程管理。

代码片段 3 多线程实现代码片段

```
//初始化子线程转移点(业务转移点)
ChatBusiness *chat = new ChatBusiness();
QThread *thread = new QThread(this);
chat->moveToThread(thread);
//建立 hash 索引
handleThreadHash[handle] = thread;
handleChatHash[handle] = chat;

/* 信号槽连接代码 */


//启动子线程
qRegisterMetaType<qintptr>("qintptr");
thread->start();
emit chat->start(handle);
```

SMTP 邮件发送

使用 EmailManager 封装了使用 163 邮箱的 smtp 服务，使得邮件的发送更简洁。

验证码生成

验证码使用 Qt 的随机数生成器产生[111111，999999]的整数，并由 MySQLManager 添加入数据库中

账号生成

新账号为当前数据库中账号转换为 int 值时的最大值+1 的 int 值对应的 QString 串。

代码片段 4 账号生成代码 片段

```
/**
 * @brief getNewAccount
 *        获取新的合法账号
* @return  账号
*/
QString MySQLManager::getNewAccount()
{
    QString sql = QString("Select Max(id) "
                          "From user_account");
    QSqlQuery query("", sqlDB);
    query.exec(sql);
    query.first();
    int id = query.value(0).toInt();

    sql = QString("Select account "
                  "From user_account "
                  "Where id=%1").arg(id);
    query.exec(sql);
    query.first();
    int account = query.value(0).toInt();

    return QString::number(++account);
}
```

5. 部署运行篇

5.1. 项目实现情况

表 24 项目功能实现情况

项目	功能	子功能	实现情况
客户端	注册账号	邮箱格式限制	实现
		验证码获取	实现
		输入判空及合法性提示	实现
		注册结果提示弹窗	实现
		成功注册邮件	实现
	登录账号	输入判空及合法性提示	实现
		登录失败提示弹窗	实现
	发送消息	消息高亮区分显示	实现
		好友聊天界面分页	实现
	添加好友	添加好友更新 UI	实现
		添加失败及合法性提示	实现
服务端	验证码生成	验证码随机生成	实现
		邮箱验证码数据库记录	实现
		验证码唯一性	未实现（仅在数据库中做了限制）
		验证码邮件	实现
	验证注册	验证码匹配验证	实现
		账号生成	实现
		账号录入	实现
		回送邮件	实现
		回送异常	实现

	验证登录	账号密码验证	实现
		初始化信息回送	实现
	添加好友	好友账号验证	实现
		在线客户端好友列表同步	实现

表 25 项目非功能实现情况

项目	要求	实现情况
登录时长	5s 内响应	实现
注册时长	15s 内响应 （受邮件服务影响）	实现 5~8 秒内得到响应 （取决于 163 邮箱的速度）
消息到达处理时长	1s 内响应	实现
数据库	密码加密存储	实现
	登录强制使用 SSL 验证	实现
消息传输	使用 AES 加密报文 RSA 加密 AES 密钥	未实现

5.2. 部署方案

服务端使用 Qt 自带的依赖查找程序将 exe 的依赖项汇总并添加 libmysql.dll 一起部署在 Windows server 2019 标准版 x64 上。在服务器上安装 Microsoft Visual C++ Redistribution for Visual Studio2019，然后运行 exe 文件即可。

客户端同样将依赖项汇总后，使用 depends 查找其他必要依赖，再使用 Enigma Virtual Box 打包即可。打包后得到无需安装，可在 win7 及以上环境直接运行使用

5.3. 运行效果

详情参见演示视频

6. 参考文献[2]

[1]. SWULWJ. 基于 Qt 的多线程 TCP 即时通讯软件的设计与实现[EB/OL].(2022-05-24).[2022-05-24]. https://blog.csdn.net/SWULWJ/article/details/124806868

[2]. Qt. QSqlDatabase Class | Qt SQL 5.15.10 [EB/OL].[2022-06-19]. https://doc.qt.io/qt-5/qsqldatabase.html#setConnectOptions

[3]. wangdeyu1994. QT5 使用 163 邮箱发送邮件[EB/OL].(2017-12-02).[2022-06-10]. https://blog.csdn.net/wangdeyu1994/article/details/78693427

[4]. C 语言中文网. Qt QListWidget 列表框用法详解[EB/OL].2012 [2022-06-05].http://c.biancheng.net/view/9418.html

[5]. herr_edoc. Qt--堆栈窗口（QStackedWidget）的使用[EB/OL].(2017-01-11).[2022-06-05].https://blog.csdn.net/trailbrazer/article/details/54344590

[2] 参考文献在本文档中没有体现，从中学习借鉴到的内容在代码中体现