浙江大学计算机科学与技术学院

Java 程序设计课程报告

2024—2025 学年 秋冬学期

题目	线上聊天室的设计与开发
学号	3220103450
学生姓名	姜雨童
所在专业	计算机科学与技术
所在班级	计科 2202

目　　录

1 引言

本次项目使用 GUI 和网络编程开发一个线上聊天室程序，要求能支持多组用户同时使用。该项目巩固了在 Java 课上学习的 GUI 和网络编程相关知识，让我对 Java 语言有了更深入的理解和更熟练的掌握，同时也提高了我的编程水平，为今后的工作学习打下基础。

1．1 设计目的

随着互联网的普及，即时通讯软件已成为人们日常生活中不可或缺的工具。本项目旨在设计并实现一个基于 Java 的聊天室应用，该应用允许用户注册、登录，并在聊天室内与其他用户进行实时交流。具体功能如下：

(1) 初始界面为注册/登录界面，允许用户以不重复的用户名注册聊天室账号并进行登录。注册信息存储在 MySQL 数据库中。

(2) 成功登录后跳转群聊界面，用户可以在主窗口（群聊窗口）与聊天室内其他人员进行聊天。

(3) 群聊窗口左侧为当前在线用户列表，点击用户名可以开启额外的小窗视图，与之进行私聊；被私聊方的界面会主动弹出私聊窗口。

1．2 设计说明

本项目采用 Java 编程语言开发，使用 Eclipse 作为开发环境。项目由多个模块组成，包括客户端、服务器端和数据库操作模块。本项目由笔者一人完成，不涉及成员分工。

2 总体设计

2．1 功能模块设计

本聊天室应用的主要功能模块包括：

• 用户注册与登录模块：允许用户创建新账户或使用现有账户登录聊天室。用户注册时需要输入用户名和密码，系统将这些信息存储到数据库中；登录时，系统会验证输入的用户名和密码是否正确。

• 聊天功能模块：用户可以在聊天室内发送和接收消息，支持私聊和群聊。私聊功能允许用户与特定的用户进行一对一的交流，而群聊功能则允许所有在线用户看到彼此的消息。

• 用户列表显示模块：实时显示当前在线用户列表。当有用户登录或退出时，系统会自动更新用户列表，以便其他用户了解当前在线的用户情况。

• 数据库操作模块：存储用户信息，支持用户注册和登录验证。使用 MySQL 数据库，通过 JDBC 连接实现对数据库的操作。

程序的总体功能如图 1 所示：

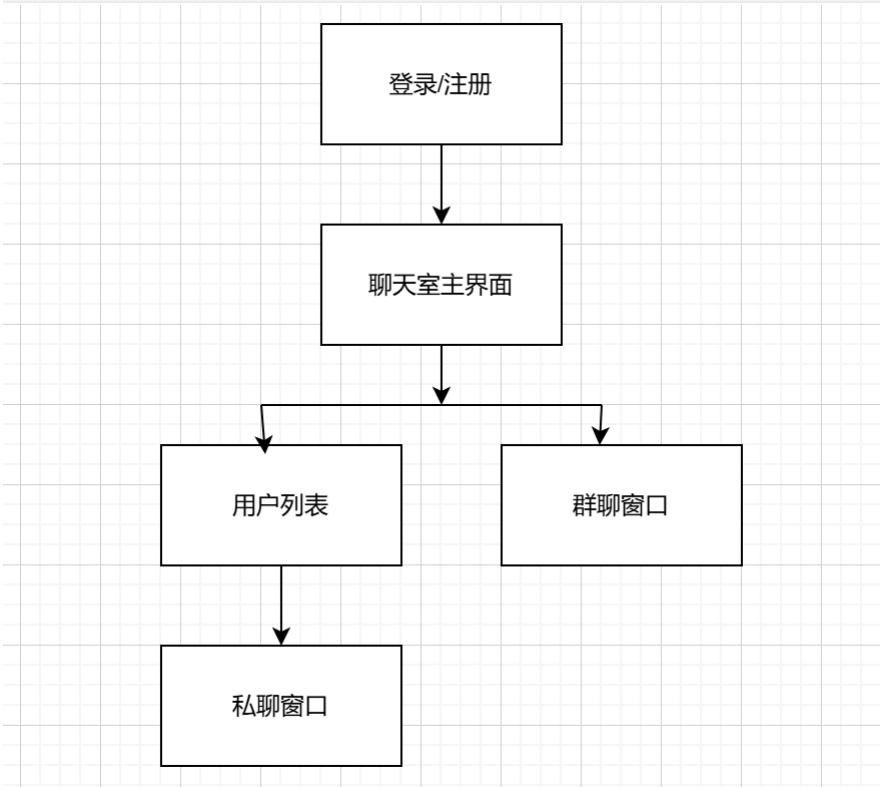

图 1 总体功能图

2. 2 流程图设计

程序总体流程如图 2 所示：

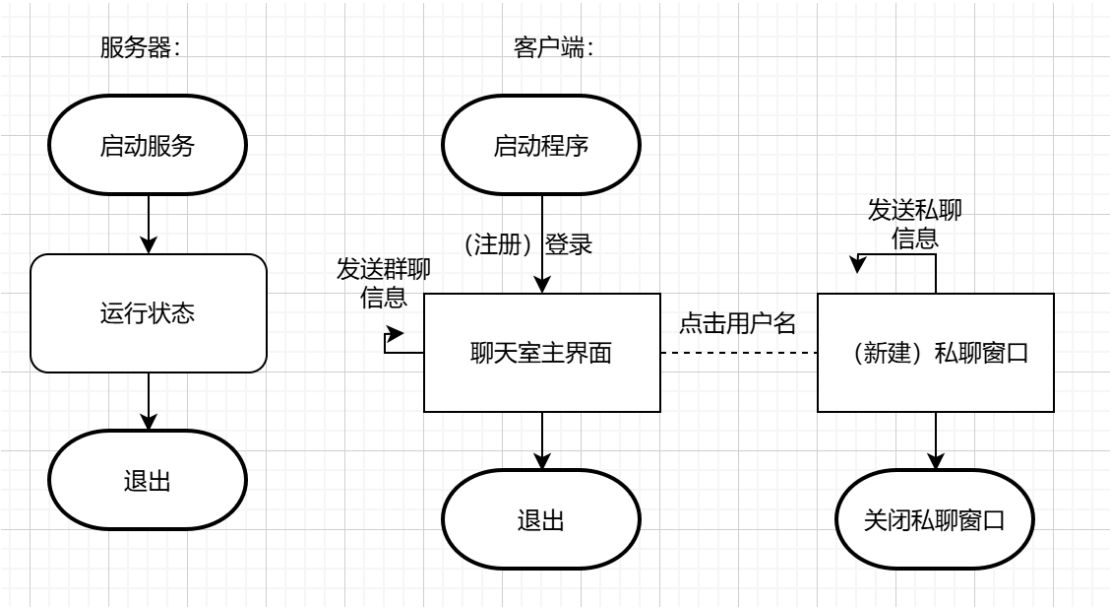

图 2 总体流程图

3 详细设计

3.1 客户端设计

客户端模块负责与用户交互，提供图形界面供用户登录、注册和聊天。主要类包括（后附主要功能类的 UML 图和数据/方法分析，其他模块设计同理）：

• ChatClient：负责创建和管理聊天窗口，处理用户输入和显示聊天信息。使用 Swing 组件构建图形界面，实现用户登录、注册、发送消息等功能。

• PrivateChatWindow：用于实现用户之间的私聊功能。当用户选择与某个用户私聊时，会弹出一个新的窗口，用户可以在该窗口中与对方进行私密交流。

• ChatMessage：定义了发送的消息，由信息发送者、信息类型和信息内容三部分组成。

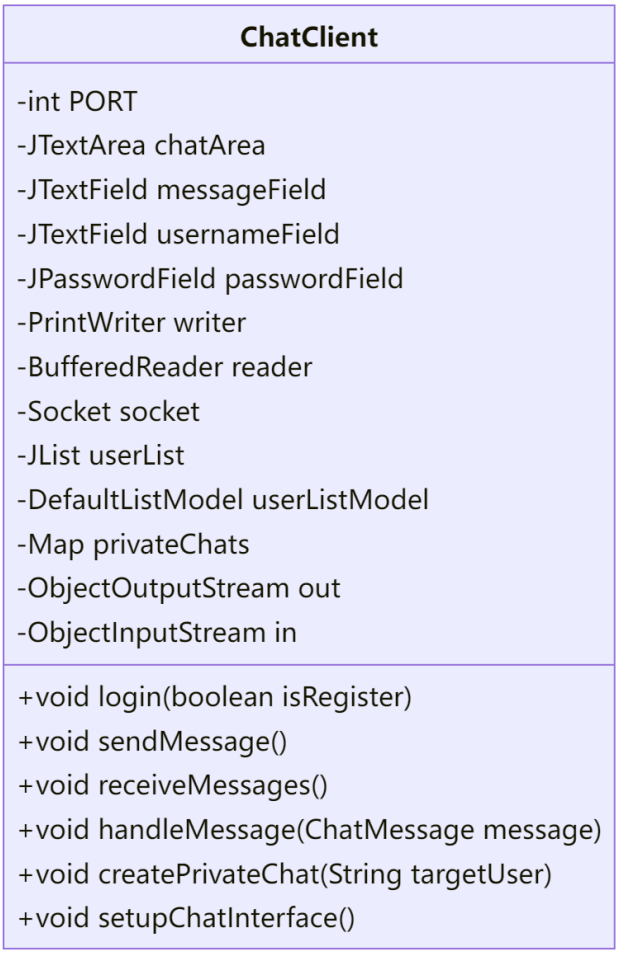

图 3 ChatClient 类的 UML 图

以下是 UML 图中有关数据和方法的详细说明：

变量说明

- PORT：聊天室服务器的端口号，客户端将通过这个端口连接到服务器。
- chatArea：显示聊天信息的文本区域，用户可以看到聊天历史记录。
- messageField：输入聊天消息的文本框，用户在此输入要发送的消息。
- usernameField：输入用户名的文本框，用户登录时需要输入用户名。
- passwordField：输入密码的密码框，用户登录时需要输入密码。
- writer、reader：用于与服务器进行数据传输的输出流和输入流。
- socket：与服务器建立连接的套接字对象。
- userList：显示在线用户列表的列表组件。
- userListModel：用户列表的数据模型，用于管理用户列表的数据。

• privateChats：存储私聊窗口的映射，键为私聊对象的用户名，值为对应的私聊窗口对象。

• out、in：用于与服务器进行对象传输的输出流和输入流。

方法说明

• login(boolean isRegister)：登录或注册的方法。根据 isRegister 参数决定是登录还是注册，通过与服务器交换消息来完成用户的身份验证或注册。

• sendMessage()：发送聊天消息的方法。获取用户输入的消息文本，通过输出流向服务器发送一个包含消息内容的 ChatMessage 对象。

• receiveMessages()：接收消息的方法。在一个循环中不断从输入流读取服务器发送的消息对象，并调用 handleMessage()方法处理这些消息。

• handleMessage(ChatMessage message)：处理接收到的消息的方法。根据消息的类型（如用户列表更新、聊天消息、私聊消息等）执行相应的操作，如更新用户列表、显示聊天消息或处理私聊消息。

• createPrivateChat(String targetUser)：创建私聊窗口的方法。如果与指定用户的私聊窗口已存在，则显示该窗口；否则，创建一个新的私聊窗口并添加到私聊窗口映射中。

• setupChatInterface()：设置聊天界面的方法。在用户登录成功后，移除登录界面，创建并显示聊天界面，包括聊天区域、消息输入框、发送按钮和用户列表等组件。

3．2　服务端设计

服务器端模块负责处理客户端的连接请求,转发消息,并管理在线用户列表。主要类包括：

• ChatServer：负责启动服务器，监听客户端连接，并管理客户端处理器。使用 ServerSocket 监听指定端口，当有客户端连接时，创建一个新的 ClientHandler 来处理该客户端的请求。

• ClientHandler：负责处理单个客户端的连接，接收和发送消息。使用 ObjectInputStream 和 ObjectOutputStream 实现与客户端的消息传输，根据接收到的消息类型（如登录、注册、聊天等）执行相应的操作。

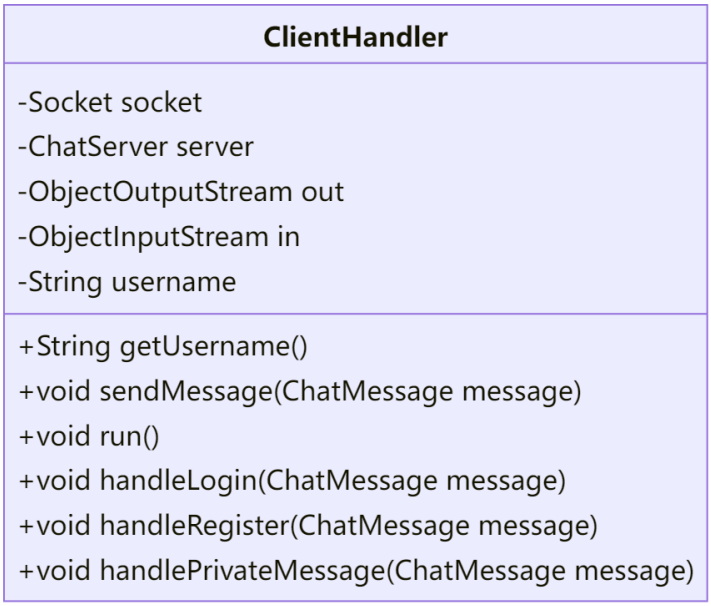

图 4 ClienHandler 类的 UML 图

以下是 UML 图中有关数据和方法的详细说明：

变量说明

• socket：与客户端建立连接的套接字对象。

• server：所属的聊天室服务器对象。

• out、in：用于与客户端进行对象传输的输出流和输入流。

• username：客户端用户的用户名。

方法说明

• getUsername()：获取客户端用户用户名的方法。

• sendMessage(ChatMessage message)：向客户端发送消息的方法。通过输出流向客户端发送一个 ChatMessage 对象。

• run()：客户端处理器的运行方法。在一个循环中不断从输入流读取客户端发送的消息对象，并根据消息类型调用相应的方法处理这些消息。

• handleLogin(ChatMessage message)：处理登录请求的方法。验证用户输入的用户名和密码是否正确，如果正确则允许用户登录，并通知其他用户；否则拒绝登录。

• handleRegister(ChatMessage message)：处理注册请求的方法。将用户输入的用户名和密码存储到数据库中，如果注册成功则允许用户登录；否则拒绝注册。

• handlePrivateMessage(ChatMessage message)：处理私聊消息的方法。将私聊消息发送给指定的接收者。

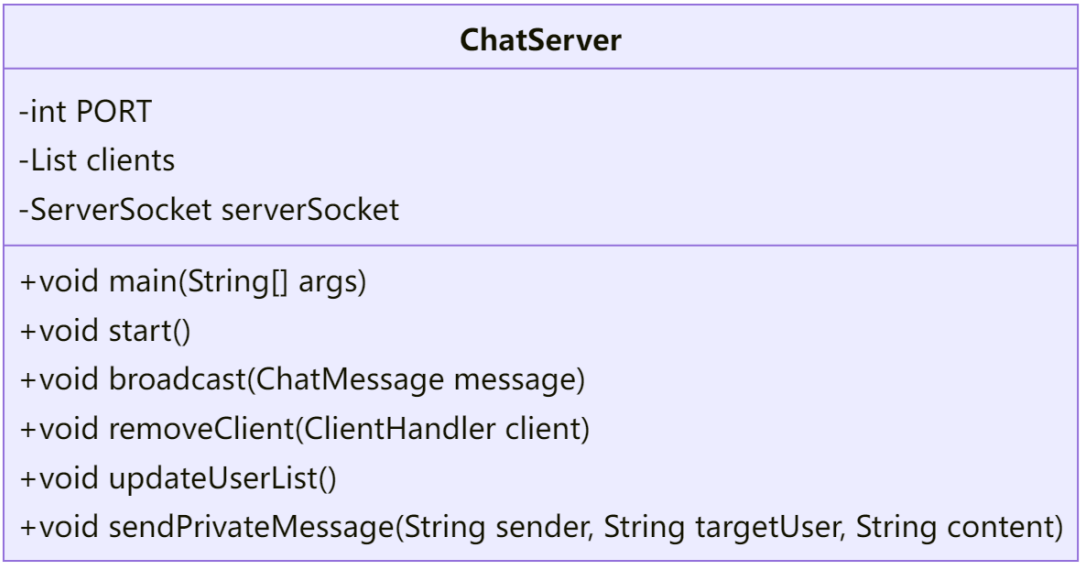

图 5 ChatServer 类的 UML 图

以下是 UML 图中有关数据和方法的详细说明：

变量说明

• PORT：聊天室服务器监听的端口号。

• clients：存储所有在线客户端处理器的列表。

• serverSocket：用于监听客户端连接请求的服务器套接字对象。

方法说明

• main(String[] args)：程序的主入口方法。创建一个 ChatServer 对象，启动服务器，并监听指定端口的客户端连接请求。

• start()：启动服务器的方法。在一个循环中不断接受客户端的连接请求，为每个连接创建一个新的 ClientHandler 对象，并启动一个新的线程来处理该客户端的请求。

• broadcast(ChatMessage message)：广播消息的方法。将消息发送给所有在线的客户端处理器，使所有用户都能收到该消息。

• removeClient(ClientHandler client)：移除客户端处理器的方法。从在线客户端处理器列表中移除指定的客户端处理器，并更新用户列表。

• updateUserList()：更新用户列表的方法。获取所有在线客户端处理器的用户名，生成一个新的用户列表，并通过广播消息通知所有客户端更新用户列表。

• sendPrivateMessage(String sender, String targetUser, String content)：发送私聊消息的方法。将私聊消息发送给指定的发送者和接收者，使只有他们能看到该消息。

3．3　数据库设计

数据库模块负责存储用户信息，支持用户注册和登录验证。主要类为：

• DatabaseUtil：提供数据库连接和用户数据操作的方法。使用 JDBC 连接 MySQL 数据库，实现用户注册时将用户信息插入数据库，登录时从数据库中查询用户信息进行验证。

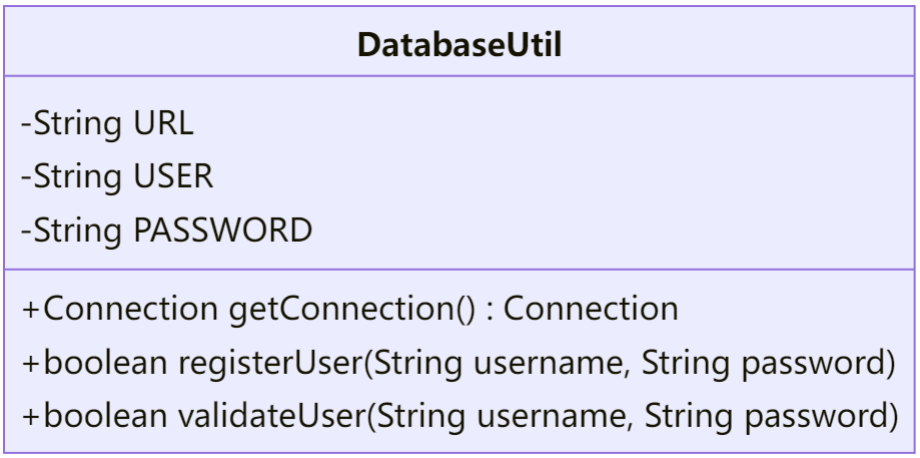

图 6 DatabaseUtil 类的 UML 图

以下是 UML 图中有关数据和方法的详细说明：

变量说明

• URL：数据库连接的 URL，指定了数据库服务器的地址、端口、名称以及时区。

• USER：数据库的用户名。

• PASSWORD：数据库的密码，用于验证登录数据库的权限。

方法说明

• getConnection()：获取数据库连接的方法。使用DriverManager.getConnection()方法，传入数据库的URL、用户名和密码，返回一个Connection对象，表示与数据库的连接。

• registerUser(String username, String password)：注册用户的方法。首先获取数据库连接，然后准备一个插入用户信息的SQL语句，将用户名和密码作为参数设置到SQL语句中，执行更新操作，如果成功则返回true，否则返回false。

• validateUser(String username, String password)：验证用户的方法。获取数据库连接后，准备一个查询用户信息的SQL语句，将用户名和密码作为参数设置到SQL语句中，执行查询操作，如果查询结果有数据（即用户存在且密码正确），则返回true，否则返回false。

4 测试与运行

4.1 程序测试

在开发过程中，笔者对各个模块的功能进行了详细的测试，包括单元测试和集成测试。测试内容：

• 用户注册和登录功能的正确性：测试用户注册时能否成功将信息存储到数据库，登录时能否正确验证用户名和密码。

• 聊天消息的发送和接收：测试用户在聊天室内发送的消息能否被其他用户正确接收，私聊功能是否能正确地将消息发送给指定的用户。

• 用户列表的更新和显示：测试当有用户登录或退出时，用户列表是否能及时更新，并正确显示当前在线的用户。

经过不断的调试修改，最后测试认为本次项目所设计的线上聊天室程序能够正常运行，且支持注册登录、群聊私聊等功能，没有出现明显的错误和漏洞。总体来看，虽然聊天室程序在细节上还有改进空间，但是总体设计在功能上已经基本达到要求。

4．2　程序运行

程序运行时，首先启动服务器，然后客户端连接到服务器。用户登录后，可以在聊天室内与其他用户交流。在 Windows 系统下可以直接在根目录使用批处理文件进行编译和运行：

编译：compile.bat

服务器运行：run-server.bat

客户端运行：run-client.bat

启动服务器后，启动一个客户端程序：

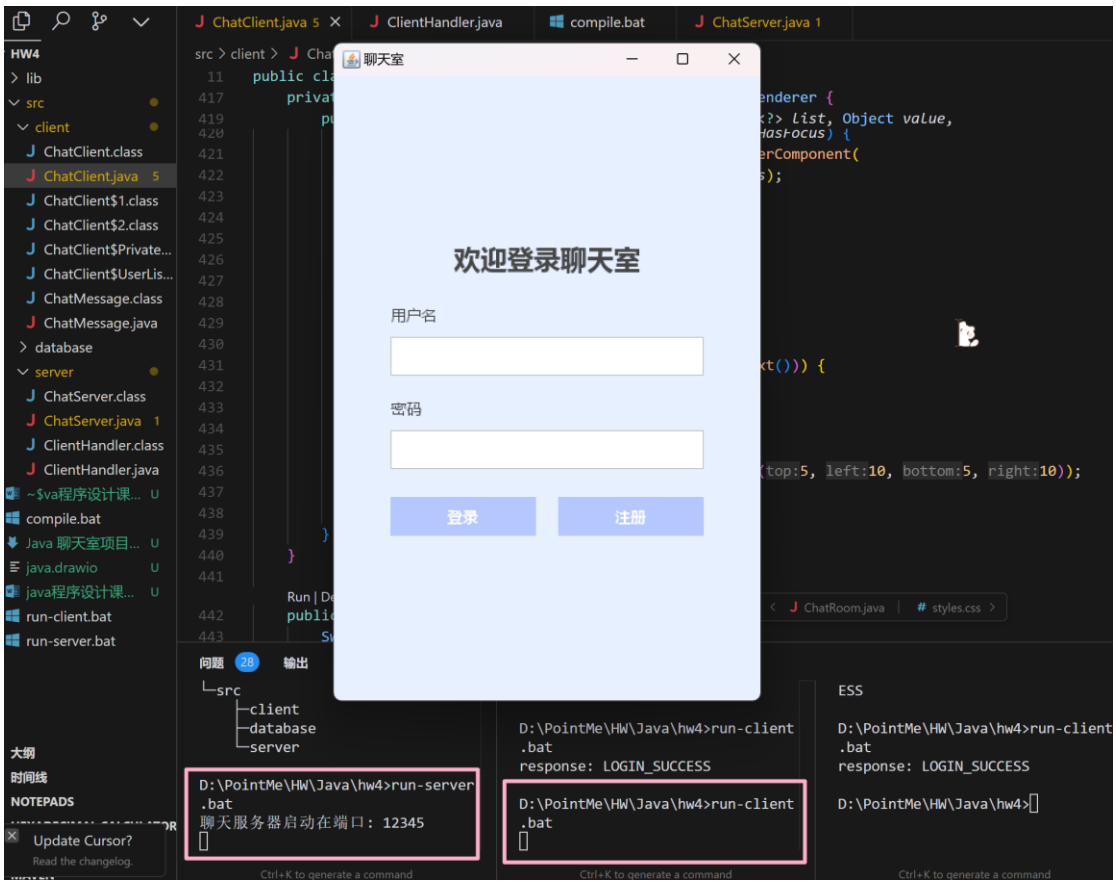

图 7 聊天室注册/登录页面

随后进行注册与登录的测试：

测试前数据库内数据如下：

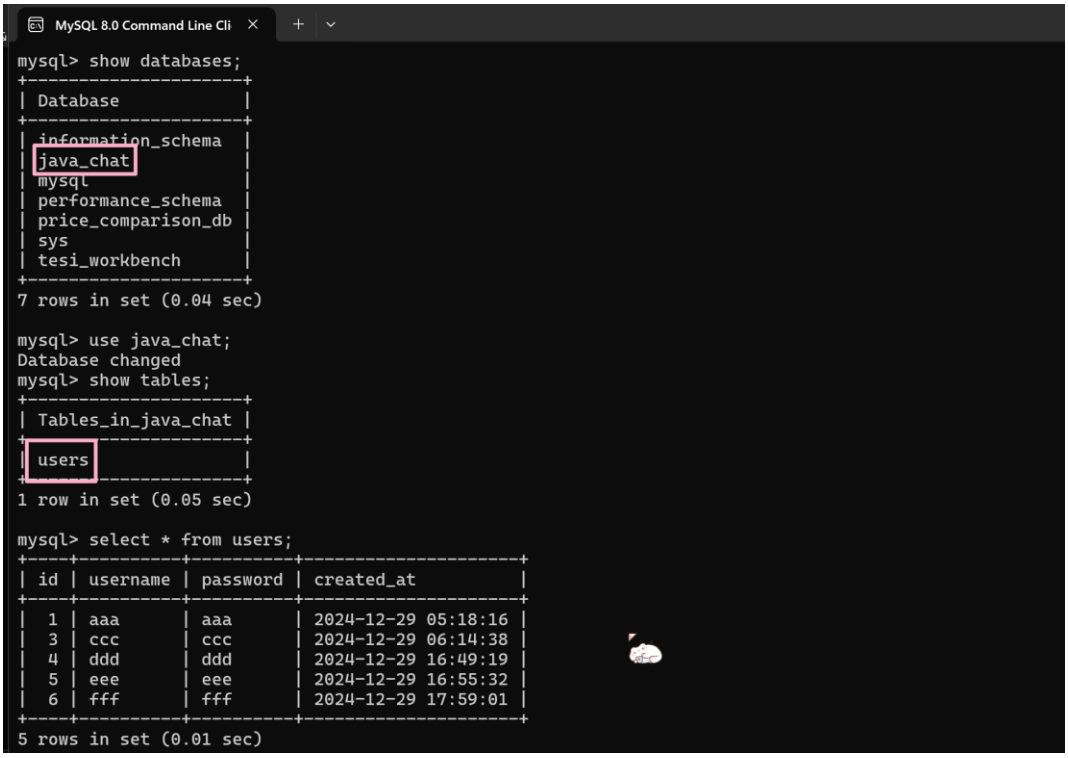

图 8 测试前数据库截图

使用用户名“aaa”和“bbb”进行注册，由于用户名唯一性，前者失败，而后者成功且程序跳转至聊天室界面：

图 9 注册测试结果图

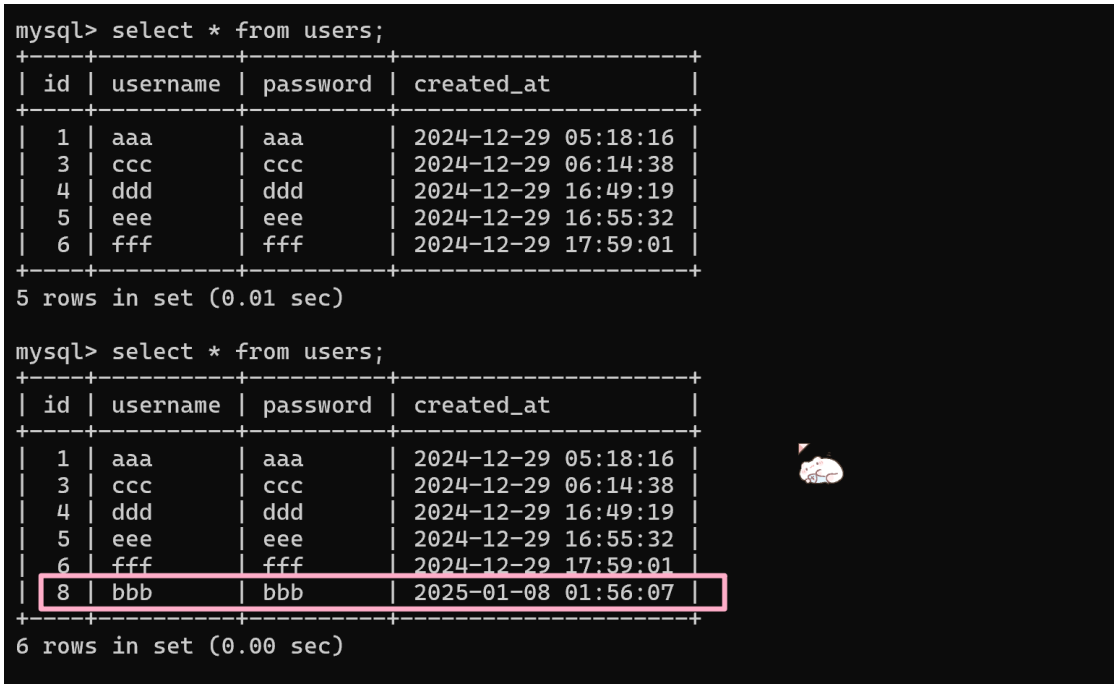

图 10 测试后数据库截图

进入聊天室主界面，系统提示用户加入聊天室，且能够在群聊发送消息：

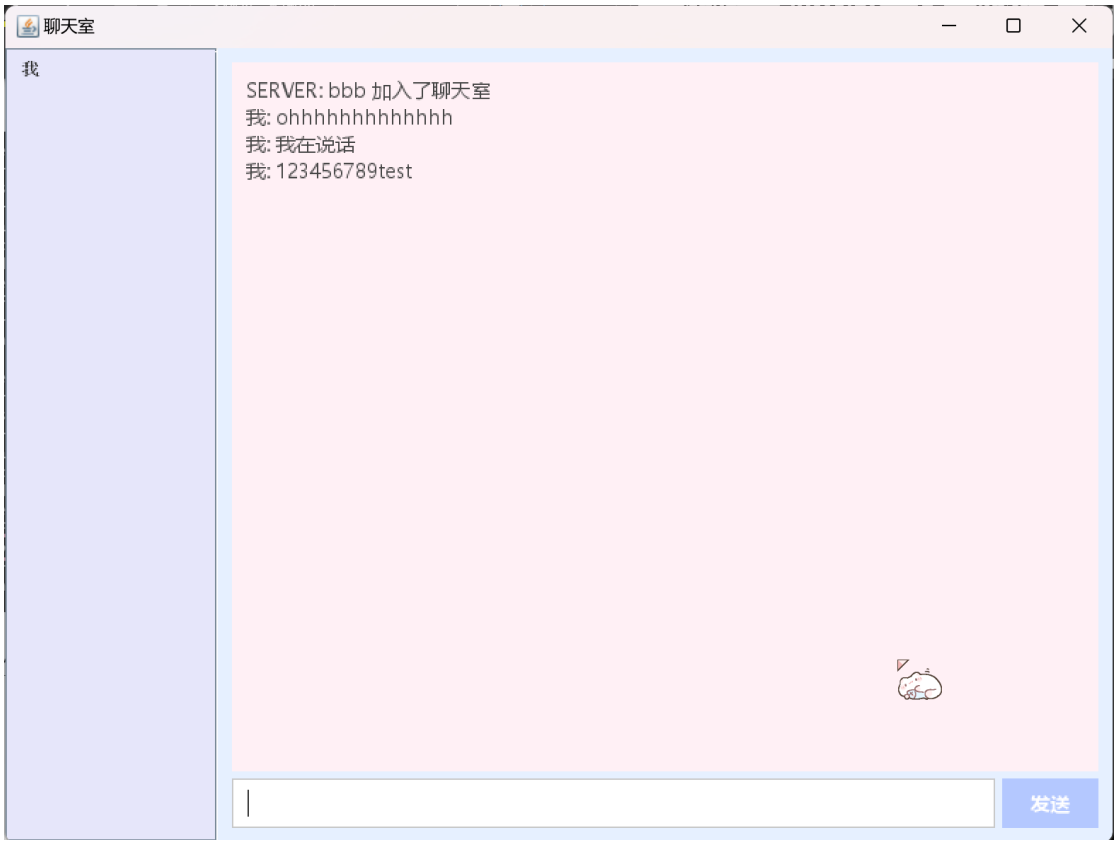

图 11 聊天室主界面/群聊

另外启动一个客户端程序，观察用户列表的更新情况：

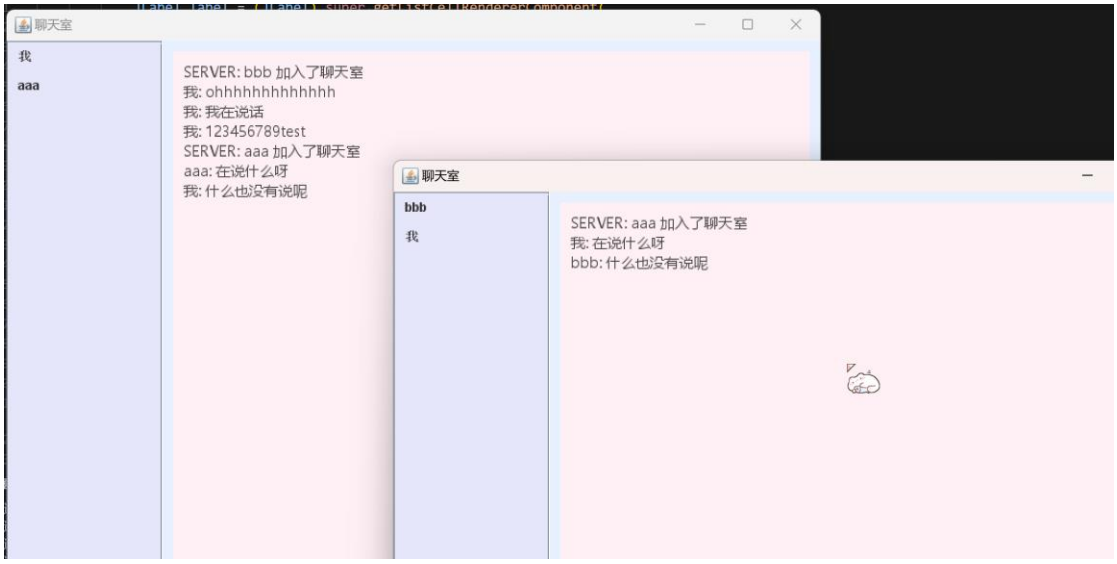

图 12 用户列表更新

再次启动一个新客户端，并点击左侧用户名使其进行私聊：

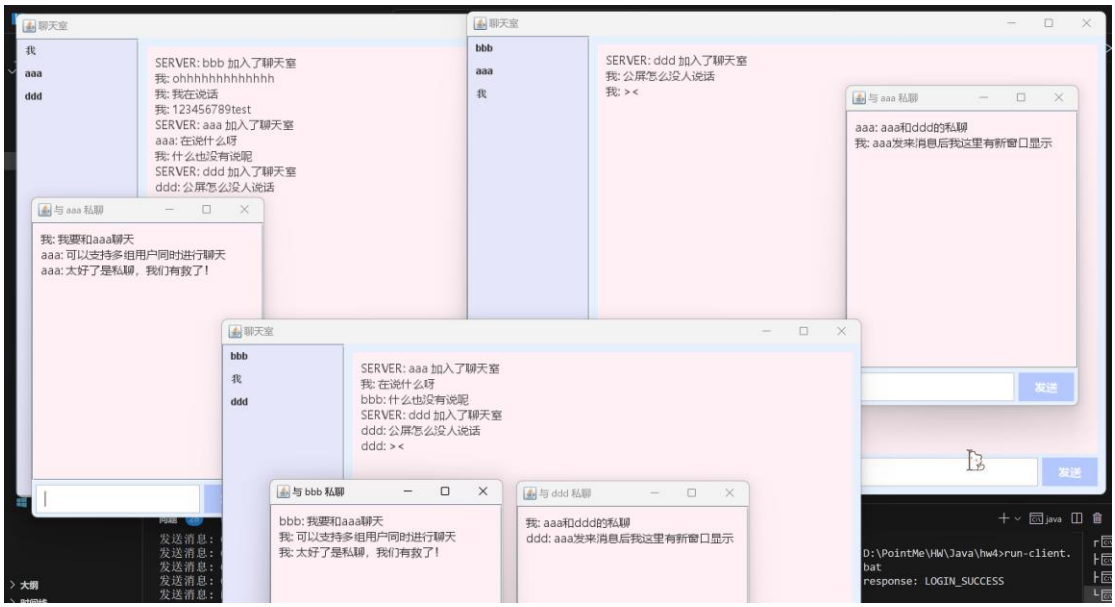

图 13 私聊界面

5．总结

这是第一次在实践中使用 GUI 相关知识编写 Java 程序，有一定难度，编程过程中也遇到很多问题，需要不断查阅相关资料调整代码。本来基于交互性考虑，设计的是不同消息发出者的消息用不同颜色显示，比如系统消息用深紫色，自己

的消息用玫红色，用户列表也同理。但是实操中发现现在的消息面板/用户列表属于一个 Panel，更改前景色时所有文字的颜色都会一起修改（比如系统消息发出来时显示深紫色，但是用户自身发出消息后，两条消息一起显示为玫红色）。要实现上述功能需要把每条消息细分成一个 Panel 并进行排版等方面的设计，较为麻烦。囿于时间和精力原因，我最后放弃了这个设计。

而在网络编程方面，本学期的另一门课计算机网络也涉及到了 socket 和相关内容并做了实验，因此较为简单。

总体来说，通过本次项目的设计与实现，笔者对 Java 网络编程和 GUI、数据库操作（JDBC）有了更深入的理解，这也为笔者今后的学习与工作打下基础。虽然仍有一些细节可以进一步优化，例如增加更多的用户交互功能、提高系统的稳定性和安全性等，但是本聊天室项目在功能上基本达到了预期目标，也具有较好的用户交互性，整体而言是一个较为完整的项目。