

浙江大学计算机学院

Java 程序设计课程报告

2024—2025 学年秋冬学期

题目	HW4：网络聊天室
学号	3220104147
学生姓名	余卓耘
所在专业	软件工程
所在班级	2202

目录

- 1 引言..... 1
 - 1. 1 设计目的..... 1
 - 1. 2 设计说明..... 1
- 2 总体设计..... 2
 - 2. 1 系统架构设计..... 2
 - 2. 2 功能模块设计..... 2
 - 2. 3 数据库设计..... 3
 - 2. 4 流程图设计..... 4
- 3 详细设计..... 5
 - 3. 1 登录界面类的设计..... 5
 - 3. 2 注册类的设计..... 6
 - 3. 3 用户类(User)的设计..... 9
 - 3. 4 数据库工具类(DBUtils)的设计..... 10
 - 3. 5 服务器类(Server)的设计..... 12
 - 3. 6 客户端类(Client)的设计..... 13
 - 3. 7 聊天室界面类(Room)的设计..... 15
- 4 测试与运行..... 16
 - 4. 1 程序测试..... 16
 - 4. 2 程序运行..... 17
- 5 总结..... 22

1 引言

本次大作业是编写一个网络程序。我选择设计并实现一个简易的网络聊天室。我不仅完成了基础要求，使用 swing 库进行 GUI 编程，使用网络编程，支持多组用户同时使用进行聊天消息发送。同时我完成了 Bonus，使用 Mysql 数据库来记录用户的账号，实现并发编程。

1. 1 设计目的

聊天室是生活中常见的网络应用。本文使用 Java 语言编写了一个聊天室，具体功能如下：

- (1) 使用网络编程，连接服务器，启动项目。用 GUI 编程实现服务器连接界面。
- (2) 实现用户注册和登录功能，用户信息存储在 MySQL 数据库中。
- (3) 支持并发编程，可以同时开启多个客户端，多个用户登录进行聊天。
- (4) 主界面左边栏显示个人信息与消息输入框，中间栏是聊天室(上方公屏，下方私屏)，右边栏显示在线好友。使用图形界面，用户点击按钮、文本框打字即可，操作简易友好。
- (5) 用户进入聊天室信息用蓝色字体显示在公屏上，用户点击右上角的×退出聊天室会用灰色字体显示在公屏上。
- (6) 支持群聊和私聊功能，支持多用户同时在线聊天。对所有人发送的消息显示在公屏与私屏，为黑色字体；私聊只显示在私屏，字体为红色。

1. 2 设计说明

本程序采用 Java 程序设计语言，在 VSCode 平台下编辑、编译与调试。具体程序由 3220104147-余卓耘独立完成。运行方式详见压缩包中的 ReadMe.txt 文件。

2 总体设计

2.1 系统架构设计

本系统采用 C/S 架构,分为服务器端和客户端两部分,系统架构如图 1 所示:

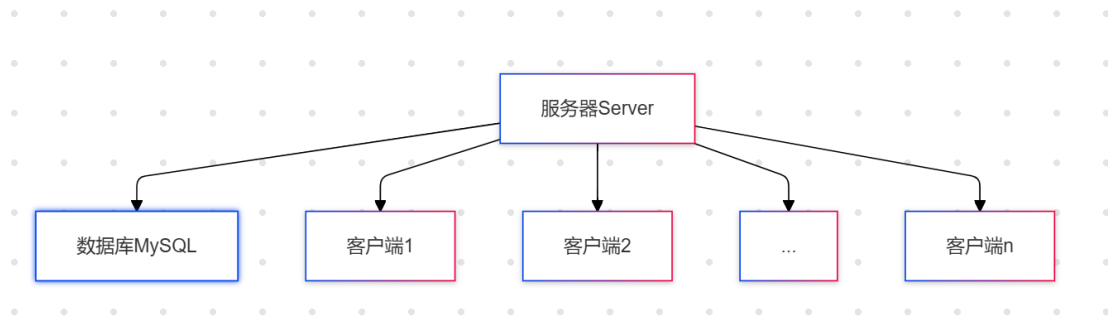


图 1 系统架构图

2.2 功能模块设计

系统的主要功能模块包括:

(1)用户管理

- 用户注册: 允许新用户系统中创建账户。
- 用户登录: 已注册用户通过此功能访问自己的账户。
- 在线用户管理: 管理系统中当前在线的用户。

(2)消息管理

- 群聊消息: 用户可以在此模块中参与群组聊天,发送和接收群组消息。
- 私聊消息: 提供用户之间一对一的私人聊天功能。

(3)界面管理

- 登录界面: 用户访问系统时首先看到的界面,用于用户登录。
- 聊天界面: 用户登录后进入的界面,用于展示聊天内容和进行聊天操作。

程序的总体功能如图 2 所示:

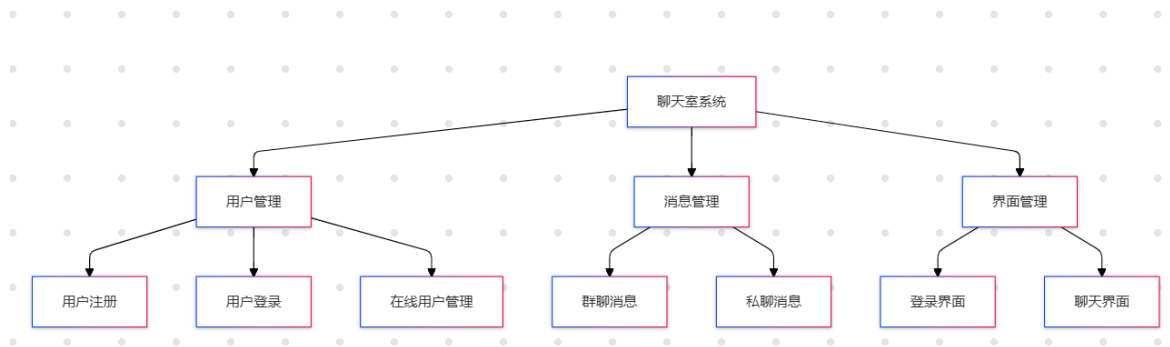


图 2 总体功能图

2. 3 数据库设计

数据表设计如下：string 类型的用户名为主键，string 类型的密码，设计如下图：

USER		
string	username	PK
string	password	

图 3 数据表设计图

建表语句如下(可以在文件的 mysql.txt 里获得)：

```
CREATE DATABASE chat_db;
```

```
USE chat_db;
```

```
CREATE TABLE user (
    id INT AUTO_INCREMENT PRIMARY KEY,
    username VARCHAR(50) NOT NULL UNIQUE,
    password VARCHAR(50) NOT NULL
```

```
);
```

插入初始数据样例如下(可以在文件的 mysql.txt 里获得)：

```
INSERT INTO USER (username, password) values
```

```
('user1', 'pass1'),
```

```
('user2', 'pass2'),
```

```
('user3', 'pass3'),
```

```
('user4', 'pass4'),
```

```
('user5', 'pass5'),
```

```
('user6', 'pass6'),
```

```
('user7', 'pass7'),
```

```
('user8', 'pass8'),
```

```
('user9', 'pass9'),
```

```
('user10', 'pass10');
```

2. 4 流程图设计

程序总体流程如图 3 所示：

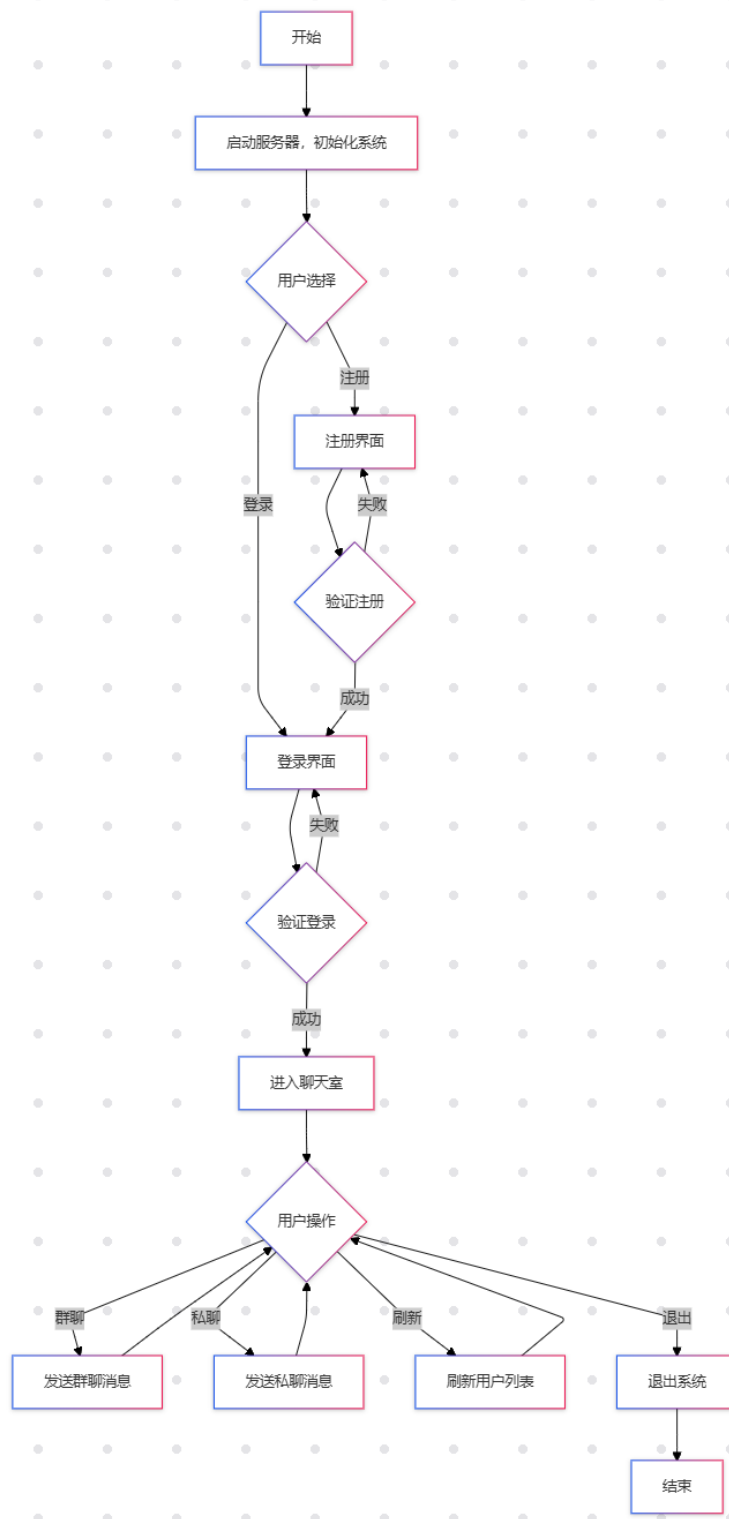


图 3 总体流程图

3 详细设计

3.1 登录界面类的设计

Login 类是 javax.swing 包中 JFrame 容器的子类, 实现了 ActionListener 接口, 负责实现登录界面的显示和登录验证功能。用户输入用户名, 密码后, 可以自由地选择自己的性别。注意, 本聊天室的设计上性别选择不固定。主要成员变量、方法以及和其他类之间组合关系的 UML 图如下所示:

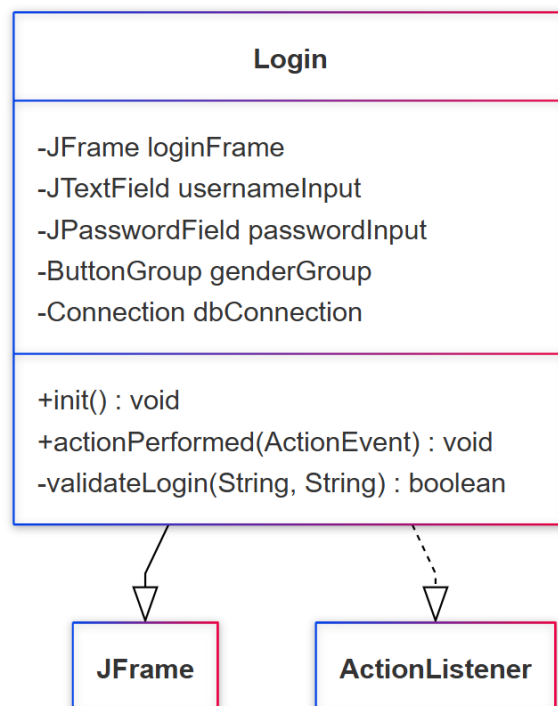


图 4 Login 类的 UML 图

(1) 成员变量

- ①loginFrame: JFrame 类型, 作为登录窗口的主框架
- ②usernameInput: JTextField 类型, 用于用户名输入
- ③passwordInput: JPasswordField 类型, 用于密码输入
- ④genderGroup: ButtonGroup 类型, 用于性别单选按钮组
- ⑤dbConnection: Connection 类型, 用于数据库连接

(2) 主要方法

- ①init() 方法负责初始化登录界面, 设置背景、布局和各组件
- ②actionPerformed() 方法实现 ActionListener 接口, 处理按钮点击事件
- ③validateLogin() 方法负责验证用户登录信息

登录类的设计流程图如下：

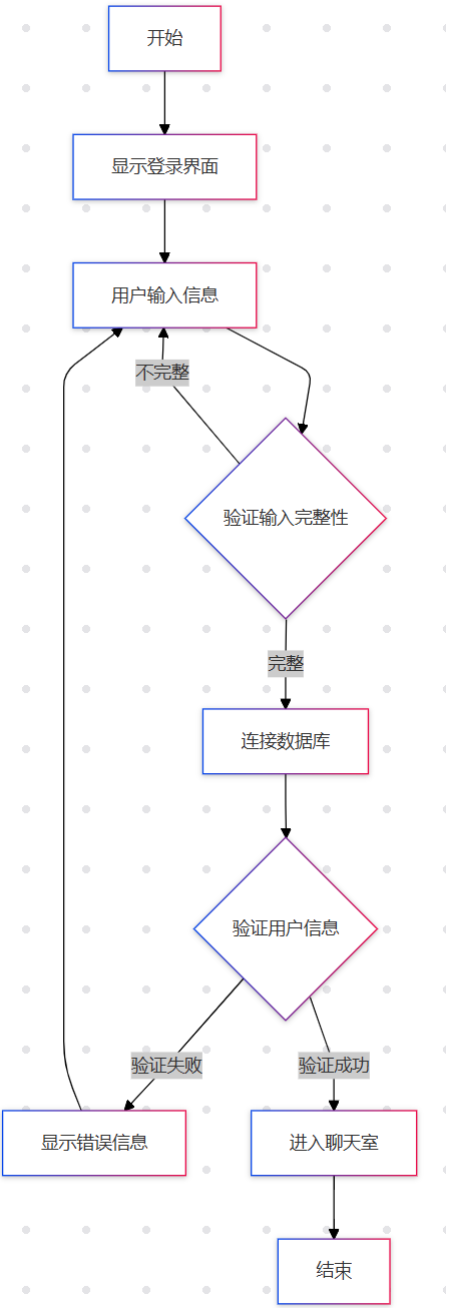


图 5 Login 类的流程图

3. 2 注册类的设计

Register 类负责实现用户注册界面及其功能。连接本地 Mysql 数据库后，注册类主要设计以下数据库操作：

-- 检查用户名是否存在

```
SELECT COUNT(*) FROM user WHERE username = ?
```

-- 插入新用户记录

```
INSERT INTO user (username, password, gender, create_time)
VALUES (?, ?, ?, NOW())
```

标明类的主要成员变量、方法以及和类之间组合关系的 UML 图如图 6 所示：

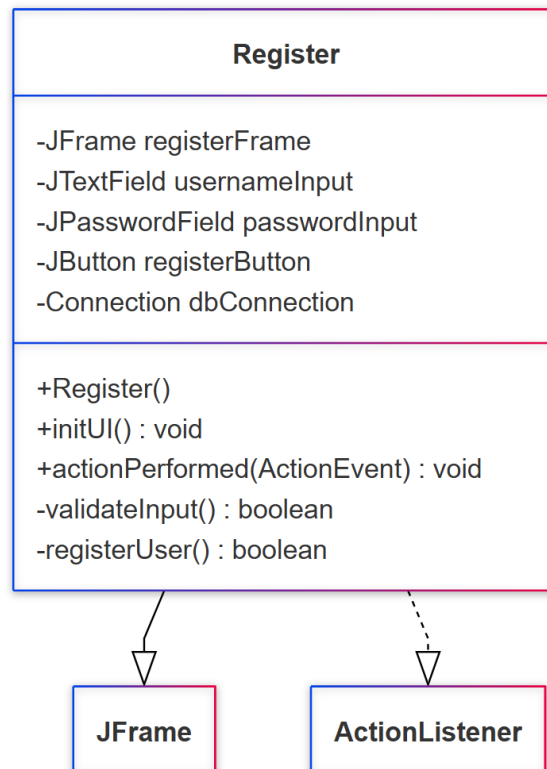


图 6 Register 类的 UML 图

以下是 UML 图中有关数据和方法的详细说明：

(1) 成员变量

- ① registerFrame 是 JFrame 类型的对象，用于创建注册界面的窗口。
- ② usernameInput 是 JTextField 类型的对象，用于输入用户名。
- ③ passwordInput 是 JPasswordField 类型的对象，用于输入密码。
- ④ registerButton 是 JButton 类型的对象，用于触发注册操作。
- ⑤ dbConnection 是 Connection 类型的对象，用于建立与数据库的连接。

(2) 方法

- ① Register() 是构造方法，用于创建 Register 类的实例。

- ② `initUI()` 是一个实例方法，用于初始化用户界面。
- ③ `actionPerformed(ActionEvent)` 是一个实例方法，用于处理按钮点击事件。
- ④ `validateInput()` 是一个实例方法，返回类型为 `boolean`，用于验证输入。
- ⑤ `registerUser()` 是一个实例方法，返回类型为 `boolean`，用于注册用户。

(3)类关系：

`Register` 类继承自 `JFrame` 类。

`Register` 类实现了 `ActionListener` 接口。

注册类设计流程图如图所示：

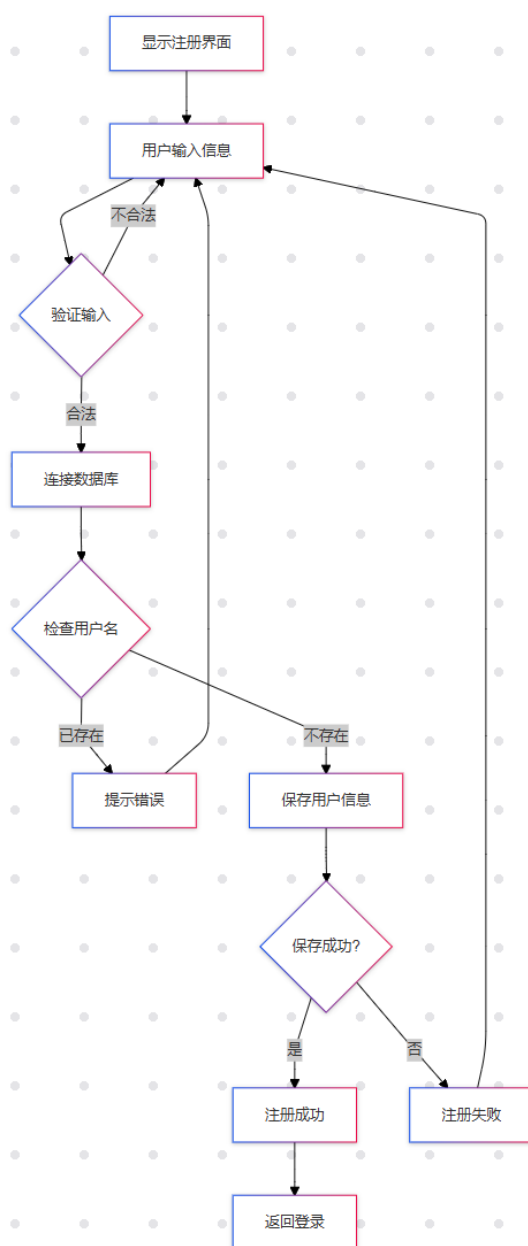


图 6 `Register` 类的 UML 图

3.3 用户类(User)的设计

User 类用于封装用户信息。标明类的主要成员变量、方法以及和类之间组合关系的 UML 图如图 6 所示:

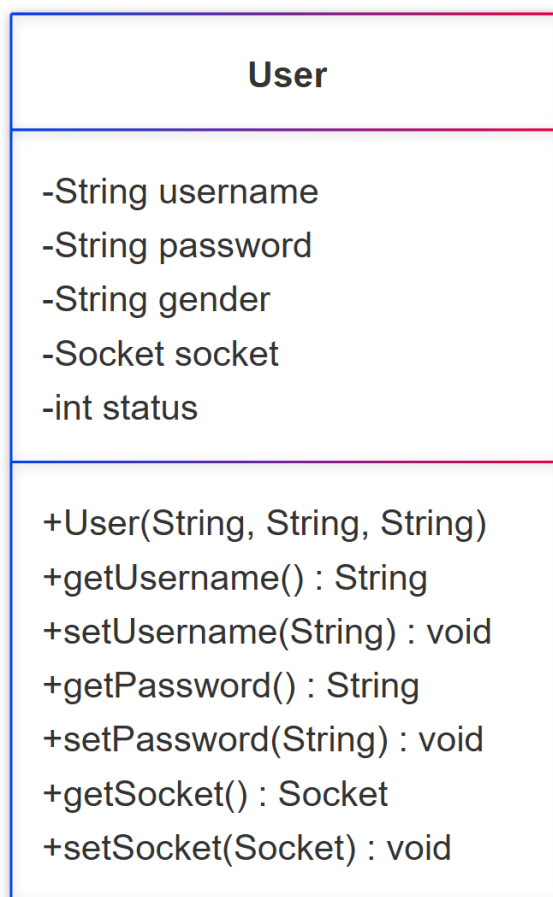


图 7 User 类的 UML 图

以下是 UML 图中有关数据和方法的详细说明:

(1) 成员变量

- ①username: String 类型, 用户名
- ②password: String 类型, 密码
- ③gender: String 类型, 性别
- ④socket: Socket 类型, 用户连接 socket
- ⑤status: int 类型, 用户状态

(2) 主要方法

- ①getter/setter 方法访问私有成员
- ②toString() 方法格式化用户信息
- ③equals() 方法比较用户对象

3. 4 数据库工具类 (DBUtils) 的设计

本聊天室实现了连接本地 Mysql 数据库。DBUtils 类负责数据库连接管理和基本操作。标明类的主要成员变量、方法以及和类之间组合关系的 UML 图如图 8 所示：

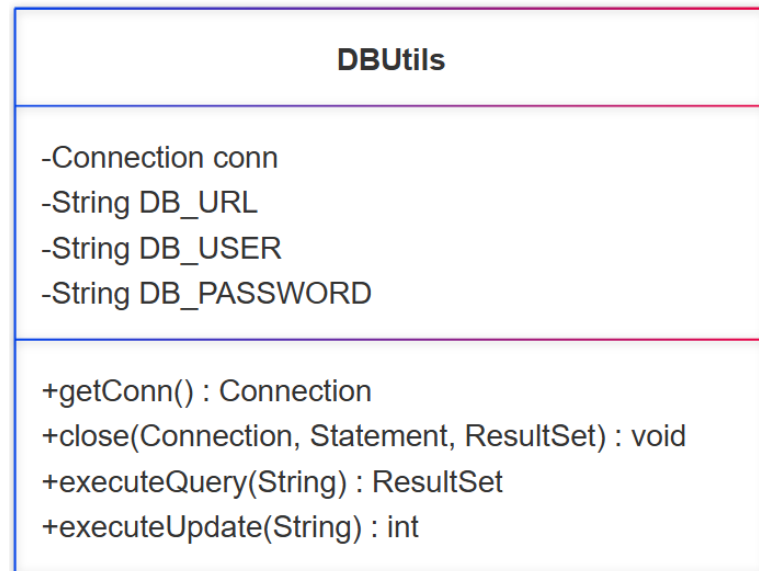


图 8DBUtils 类的 UML 图

以下是 UML 图中有关数据和方法的详细说明：

(1) 成员变量

- ① conn: Connection 类型, 数据库连接对象
- ② DB_URL: String 常量, 数据库连接 URL
- ③ DB_USER: String 常量, 数据库用户名
- ④ DB_PASSWORD: String 常量, 数据库密码

(2) 主要方法

- ① getConn() 方法获取数据库连接
- ② close() 方法关闭数据库资源
- ③ executeQuery() 方法执行查询
- ④ executeUpdate() 方法执行更新

数据库连接流程图如下：

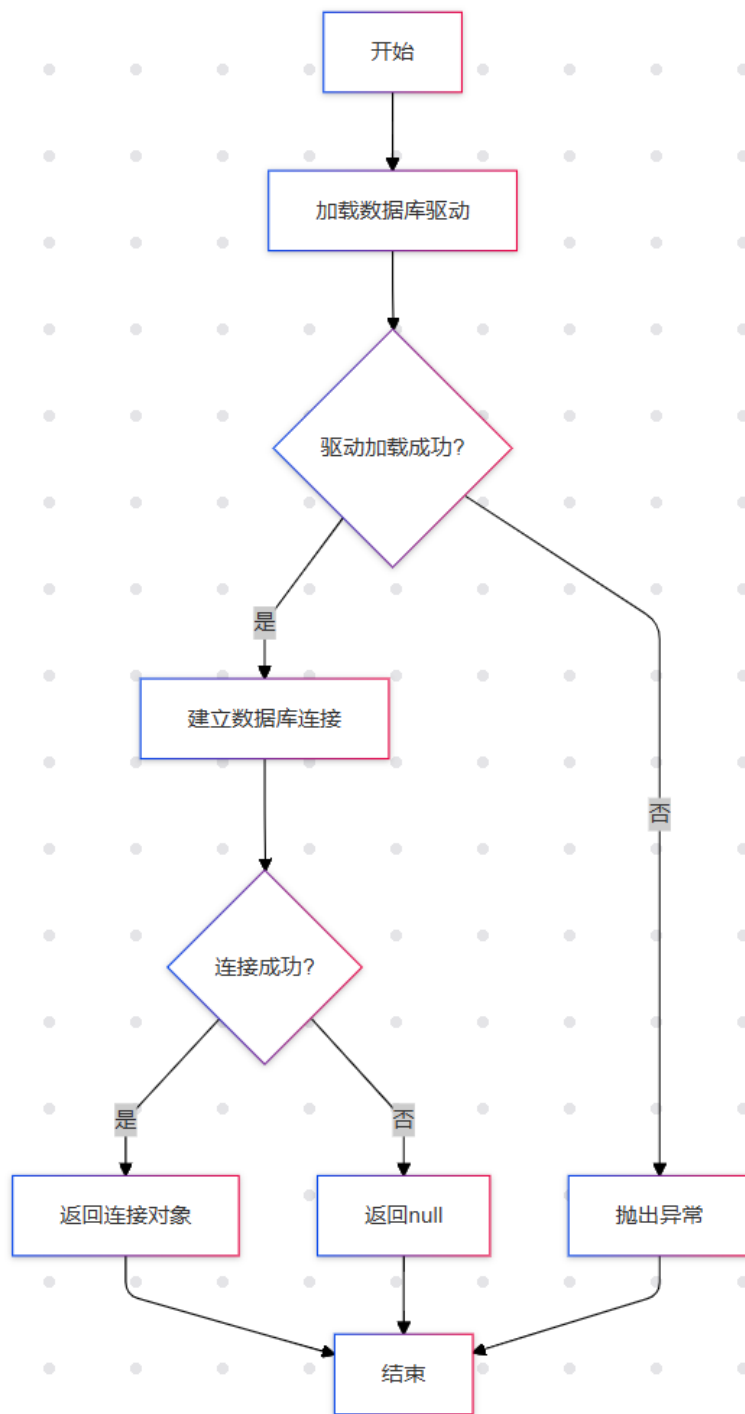


图 9 DBUtils 类的数据库连接流程图

3.5 服务器类(Server)的设计

Server 类继承 Frame 类并实现 ActionListener 接口,负责处理客户端连接请求和消息转发。标明类的主要成员变量、方法以及和类之间组合关系的 UML 图如图 10 所示:

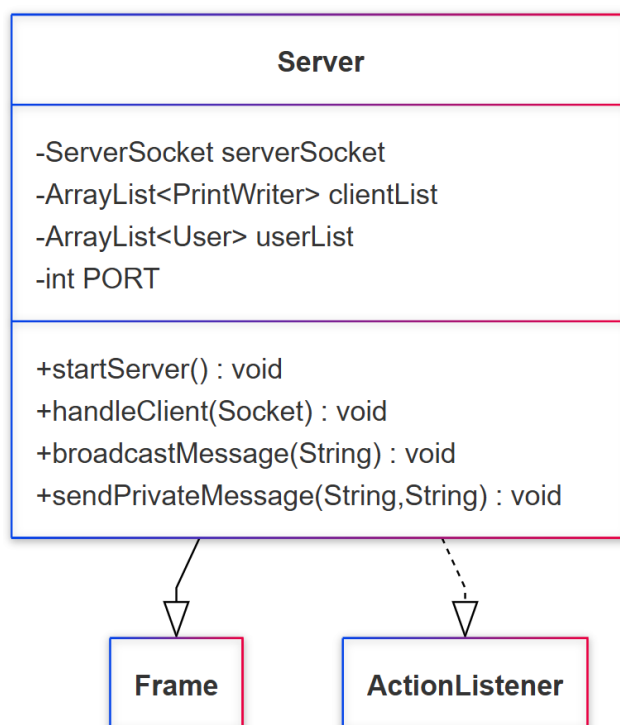


图 10 Server 类的 UML 图

以下是 UML 图中有关数据和方法的详细说明:

(1) 成员变量

serverSocket: ServerSocket 类型,用于监听客户端连接
clientList: ArrayList<PrintWriter>类型,保存所有客户端输出流
userList: ArrayList<User>类型,保存在线用户信息
PORT: int 类型常量,服务器监听端口号

(2) 主要方法

startServer() 方法启动服务器监听
handleClient() 方法处理客户端连接
broadcastMessage() 方法实现消息群发
sendPrivateMessage() 方法实现私聊功能

消息处理流程图如下:

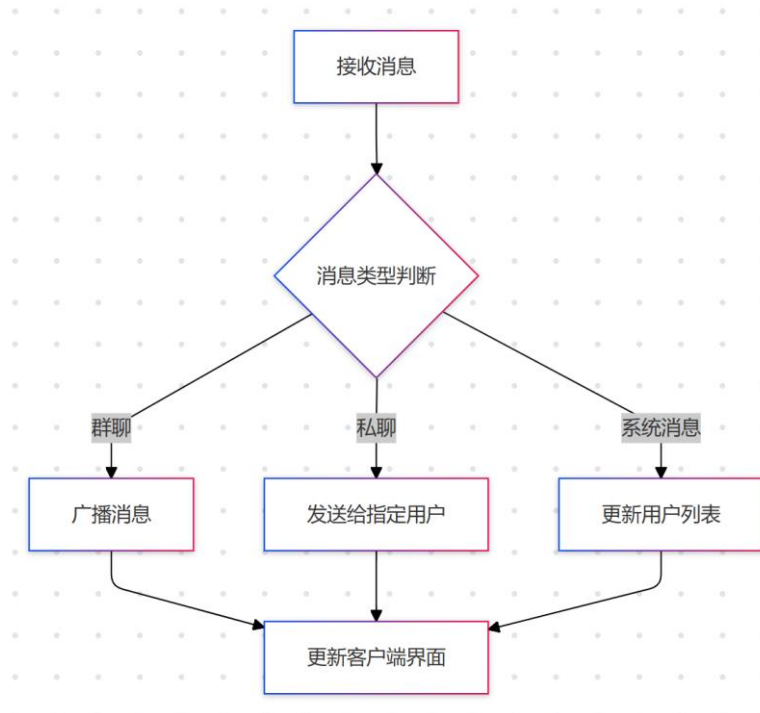


图 11 消息处理流程图

3. 6 客户端类(Client)的设计

Client 类负责与服务器建立连接,发送和接收消息。标明类的主要成员变量、方法以及和类之间组合关系的 UML 图如图 12 所示:

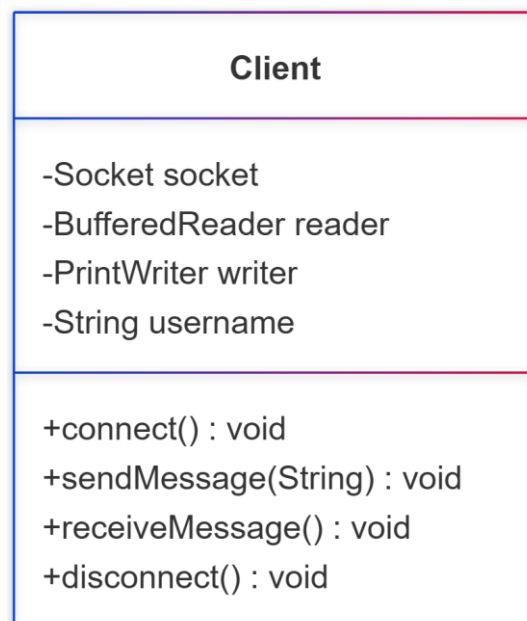


图 12 Client 类的 UML 图

以下是 UML 图中有关数据和方法的详细说明：

(1) 成员变量

socket: Socket 类型, 与服务器的连接

reader: BufferedReader 类型, 读取服务器消息

writer: PrintWriter 类型, 向服务器发送消息

username: String 类型, 用户名

(2) 主要方法

connect() 方法建立与服务器的连接

sendMessage() 方法发送消息

receiveMessage() 方法接收消息

disconnect() 方法断开连接

通信过程流程图如下：

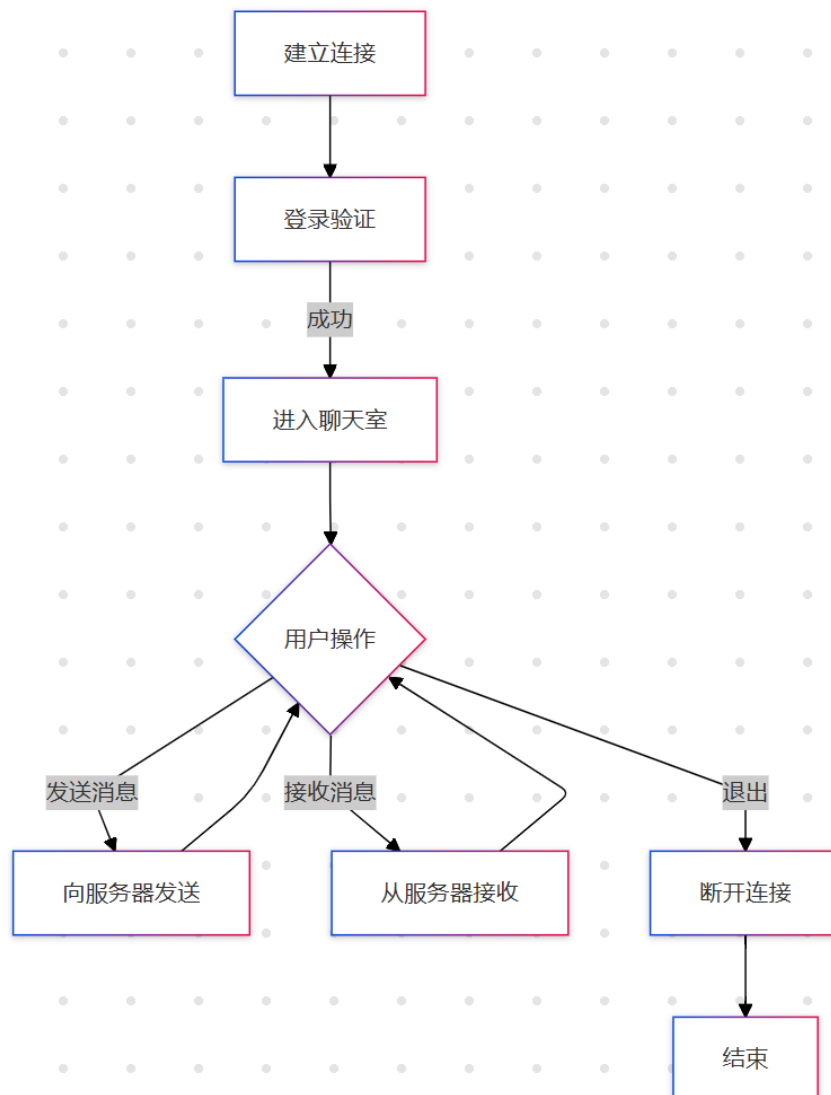


图 13 聊天通信流程图

3.7 聊天室界面类(Room)的设计

Room 类继承 JFrame 类,实现聊天室的图形界面。聊天室界面分为三栏:主界面左边栏显示个人信息与消息输入框,中间栏是聊天室(上方公屏,下方私屏),右边栏显示在线好友。使用图形界面,用户点击按钮、文本框打字即可,操作简单友好。

标明类的主要成员变量、方法以及和类之间组合关系的 UML 图如图 14 所示:

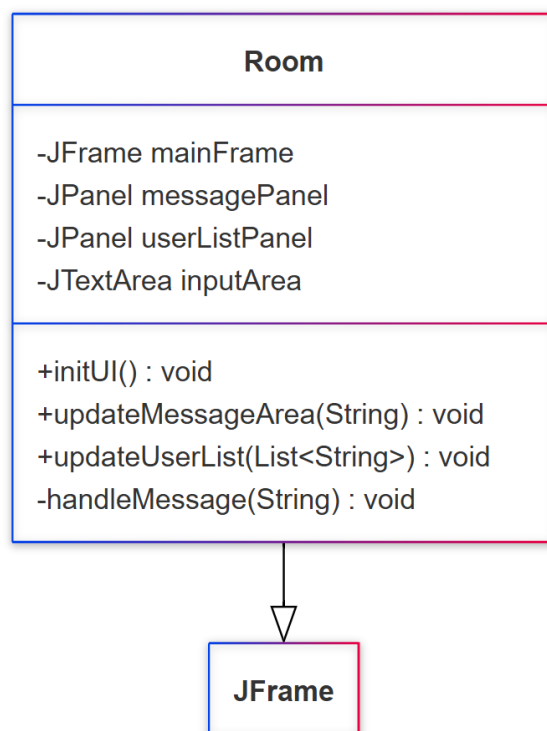


图 14 Room 类的 UML 图

以下是 UML 图中有关数据和方法的详细说明:

(1) 成员变量

mainFrame: JFrame 类型, 主窗口
messagePanel: JPanel 类型, 消息显示区域
userListPanel: JPanel 类型, 在线用户列表
inputArea: JTextArea 类型, 消息输入区域

(2) 主要方法

initUI() 方法初始化界面
updateMessageArea() 方法更新消息显示
updateUserList() 方法更新用户列表
handleMessage() 方法处理消息显示格式

4 测试与运行

4.1 程序测试

在程序代码基本完成后，经过不断的调试与修改。采用集成测试与单元测试的方法进行测试。数据表创建详见压缩包的 Mysql.txt。本次所设计的聊天室能够正常运行，没有出现明显的错误和漏洞。具体界面运行如 4.2 截图详细展示。

本项目具有异常处理功能，测试展示如下：

登录页面用户名、密码输入错误如图

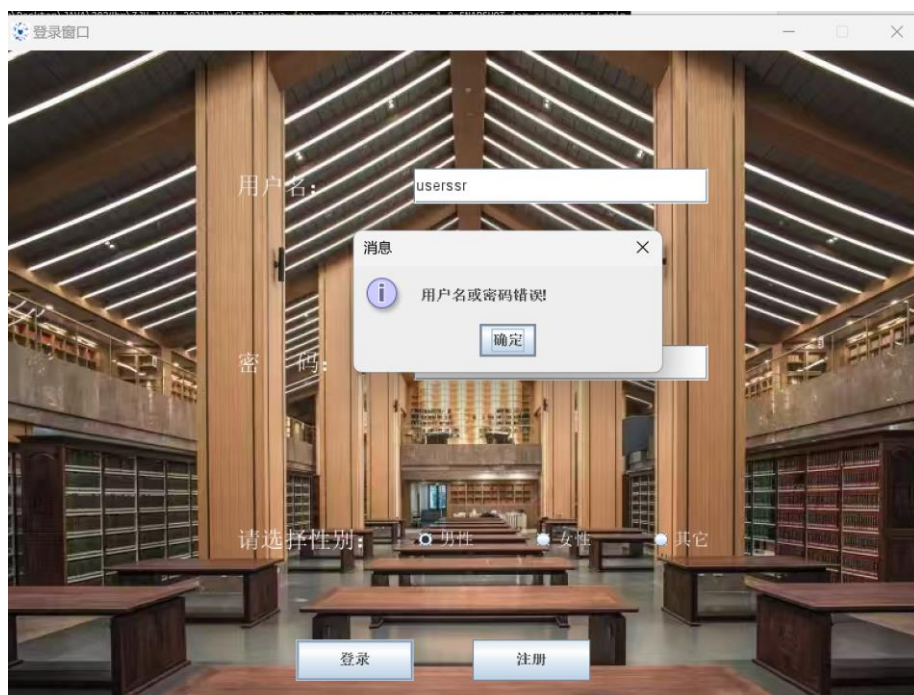


图 14 登录界面输入异常处理图

注册界面用户名冲突异常处理如下图：

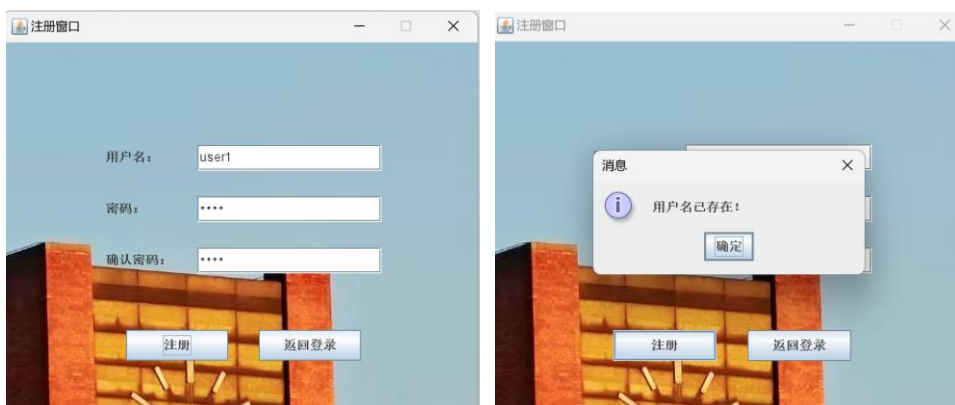


图 15 注册界面重复异常处理图

4.2 程序运行

程序运行方法见 ReadMe.txt:

先运行服务端

java -cp target/ChatRoom-1.0-SNAPSHOT.jar components.Server

```
PS D:\User\Desktop\JAVA\2024hw\ZJU_JAVA_2024\hw4\ChatRoom> java -cp target/ChatRoom-1.0-SNAPSHOT.jar components.Server
服务器启动成功, 监听端口: 1234
```

再运行客户端（从 Login 开始）

java -cp target/ChatRoom-1.0-SNAPSHOT.jar components.Login

支持并发编程，多个客户端同时登录。终端输出如下：

```
PS D:\User\Desktop\JAVA\2024hw\ZJU_JAVA_2024\hw4\ChatRoom>
服务器启动成功, 监听端口: 1234
【user1】进入聊天室
【user2】进入聊天室
【user3】进入聊天室

PS C:\Users\17106> cd D:\User\Desktop\JAVA\2024hw\ZJU_JAVA_2024\hw4\ChatRoom
PS D:\User\Desktop\JAVA\2024hw\ZJU_JAVA_2024\hw4\ChatRoom> java -cp target/ChatRoom-1.0-SNAPSHOT.jar components.Login
点击登录按钮
尝试连接数据库...
数据库连接成功: com.mysql.cj.jdbc.ConnectionImpl@1dc440cc
数据库连接成功!
执行SQL查询: select * from user where username=? and password=?
用户名: user2
密码: pass2
查询完成, 检查结果...
登录成功, 准备打开聊天室...
聊天室界面创建完成, 准备建立连接...
【user2(girl)】欢迎来到聊天室!
聊天室连接建立完成

PS D:\User\Desktop\JAVA\2024hw\ZJU_JAVA_2024\hw4\ChatRoom> java -cp target/ChatRoom-1.0-SNAPSHOT.jar components.Login
点击登录按钮
尝试连接数据库...
数据库连接成功: com.mysql.cj.jdbc.ConnectionImpl@517657a8
数据库连接成功!
执行SQL查询: select * from user where username=? and password=?
用户名: user3
密码: pass3
查询完成, 检查结果...
登录成功, 准备打开聊天室...
聊天室界面创建完成, 准备建立连接...
【user3(non-binary)】欢迎来到聊天室!
聊天室连接建立完成

PS D:\User\Desktop\JAVA\2024hw\ZJU_JAVA_2024\hw4\ChatRoom> java -cp target/ChatRoom-1.0-SNAPSHOT.jar components.Login
点击登录按钮
尝试连接数据库...
数据库连接成功: com.mysql.cj.jdbc.ConnectionImpl@682685ce
数据库连接成功!
执行SQL查询: select * from user where username=? and password=?
用户名: user4
密码: pass4
查询完成, 检查结果...
登录成功, 准备打开聊天室...
聊天室界面创建完成, 准备建立连接...
【user4(boy)】欢迎来到聊天室!
聊天室连接建立完成
```

图 16 启动后终端输出情况

服务器（Server）启动后界面如图，点击“启动服务器”按钮启动

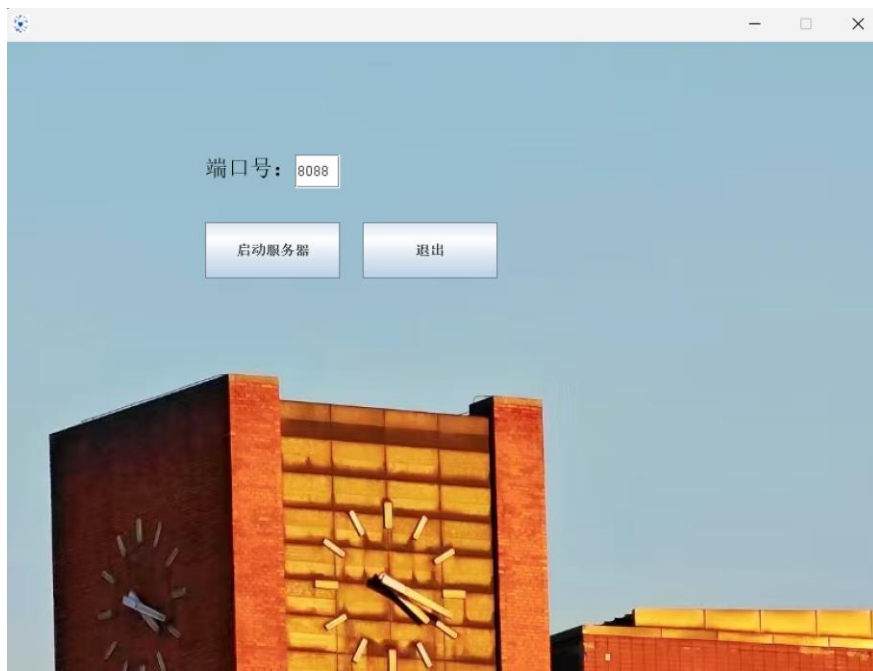


图 17 服务器启动界面图

服务器成功启动后会打开如图所示的窗口：



图 18 服务器成功启动后打开窗口图

然后打开新的终端界面，在根目录下输入：

```
java -cp target/ChatRoom-1.0-SNAPSHOT.jar components.Login
```

启动登录界面。注意此处实现了并发编程，可以同时打开多个终端启动多个登录界面。

登录界面图如下，可以直接输入文字，点击按钮：

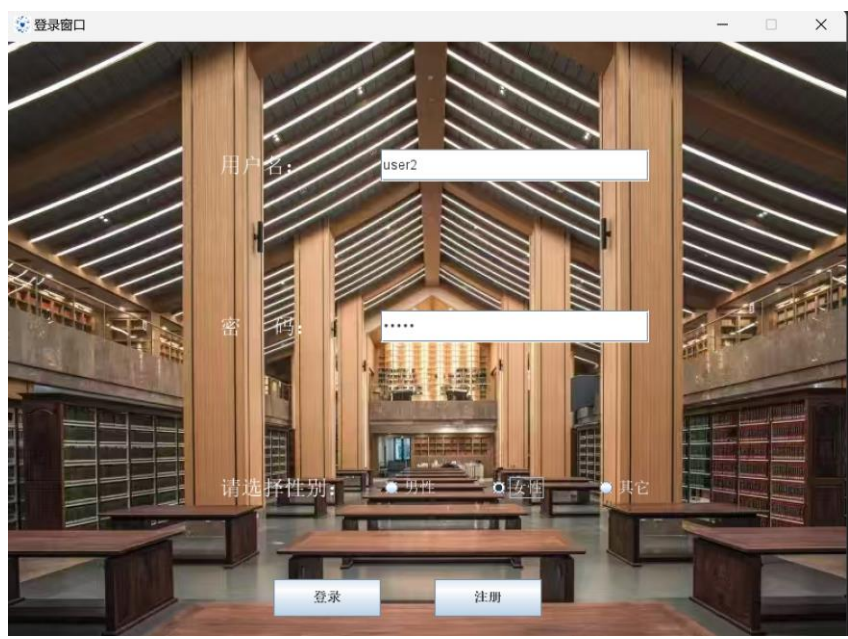


图 19 登录界面图

若输入正确，点击登录按钮即可进入聊天室。

若没有账号，可以先点击注册按钮，在如下注册界面进行注册：

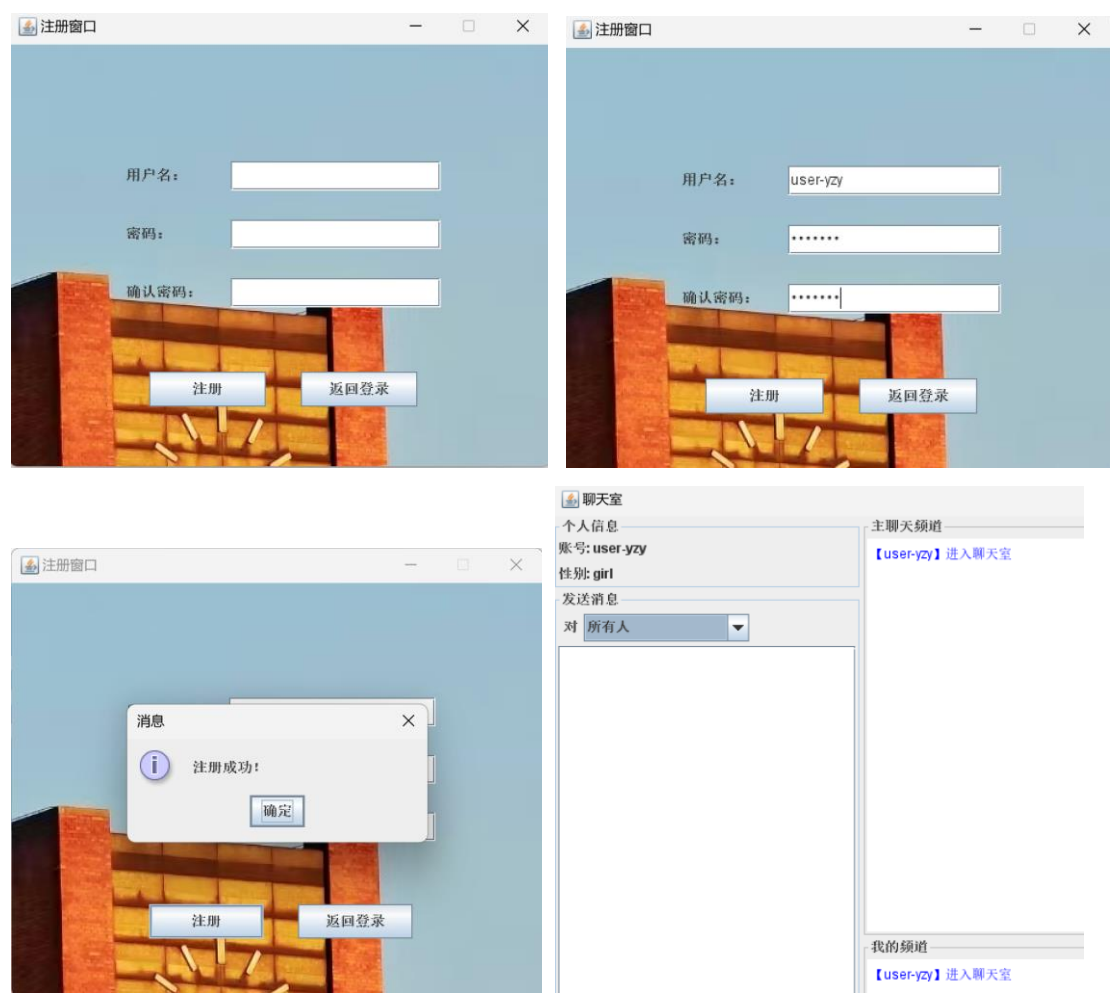


图 20 注册界面图

进入聊天室后主界面如图所示。聊天室界面分为三栏：主界面左边栏显示个人信息与消息输入框，中间栏是聊天室(上方公屏，下方私屏)，右边栏显示在线好友。使用图形界面, 用户点击按钮、文本框打字即可，点击左下角发送按钮即可发送消息，操作简易友好。

用户进入聊天室信息用蓝色字体显示在公屏上，用户点击右上角的×退出聊天室会用灰色字体显示在公屏上。

支持群聊和私聊功能，支持多用户同时在线聊天。对所有人发送的消息在上方“主聊天频道”下方“我的频道”均显示，为黑色字体；私聊只显示在下方“我的频道”，字体为红色。

参考微信的设计，晚进入的用户看不到更早的消息，只看到自己进入之后的消息。下面是最早进入聊天室的 user1 与第三个进入聊天室的 user3 视角的界面展示，可以看到群聊、私聊等功能均正常。

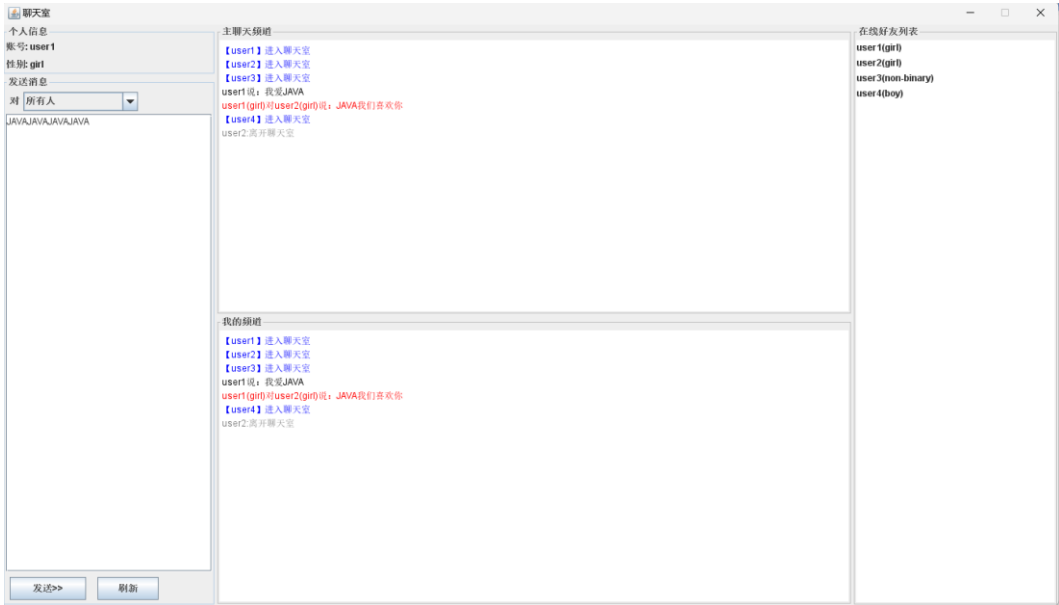


图 21 user1 视角的聊天室主界面图



图 22 user3 视角的聊天室主界面图

刷新按钮功能展示:

如图, 左边是选择了聊天对象, 输入了聊天内容的, 若用户想清空, 恢复默认聊天设置, 点击刷新按钮即可回到右边的状态。

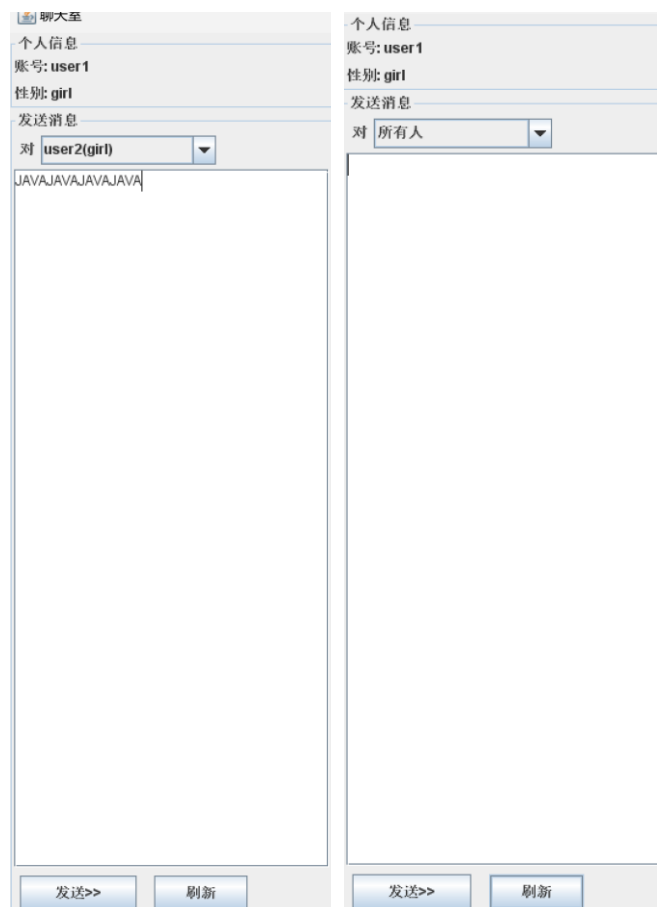


图 23 刷新按钮功能展示图

5 总结

通过本次聊天室项目的开发，我深入理解了网络编程的基本原理，掌握了 Socket 知识，这为我处理网络通信问题打下了坚实的基础。同时，我也探索了多线程编程的应用，这在实现聊天室的多用户并发访问中发挥了关键作用。

在用户界面（GUI）编程方面，我学习了如何设计直观且用户友好的界面，并通过事件处理机制增强了交互性。此外，我还涉足了数据库编程，通过 JDBC（Java Database Connectivity）实现了数据的持久化存储，这为聊天室的消息管理和用户信息存储提供了支持。

项目架构设计的重要性在我心中留下了深刻印象。一个良好的架构设计不仅有助于代码的组织和维护，也是确保项目可扩展性的关键。在开发过程中，我面临了多重挑战，包括多用户并发访问的处理、界面布局的优化以及各个模块之间的协调。通过不懈的努力和问题解决，我的编程能力和项目开发经验得到了显著提升。

尽管这个项目规模不大，但它覆盖了软件开发的多个关键领域，整合了上学期修读的数据库系统，与本学期修读的计算机网络的知识，为我提供了一次全面的实践机会。然而，我也意识到项目还有改进的空间。未来，我计划添加更多功能，如表情支持和文件传输，以丰富用户的交流体验。同时，我将致力于优化界面设计，使其更加美观和易用。此外，提高系统的稳定性和安全性也是我接下来的工作重点，确保用户能够享受到更加流畅和安全的聊天体验。我期待将这些经验应用到未来的项目中，不断进步和成长。

参考文献

- [1] 耿祥义. Java 大学实用教程[M]. 北京：清华大学出版社，2009.
- [2] 耿祥义. Java 课程设计[M]. 北京：清华大学出版社，2008.
- [3] 王鹏. Java Swing 图形界面开发与案例详解[M]. 北京：清华大学出版社，2008.
- [4] 丁振凡. Java 语言实验教程[M]. 北京：北京邮电大学出版社，2005.
- [5] 郑莉. Java 语言程序设计[M]. 北京：清华大学出版社，2006.