**即时聊天系统****文档**

****

**学 院\_\_\_\_智能与计算学部\_\_**

**专 业\_\_\_\_\_软件工程\_\_\_\_\_\_**

**组 长\_\_\_\_\_\_\_赵浩喆\_\_\_\_\_\_**

**学 号 3018216249**

**姓 名\_\_\_\_\_ 马骁 \_\_\_\_**

**学 号 3018216235**

**姓 名\_\_\_\_\_\_\_ 郑启俊\_\_ \_**

**学 号 3018216250**

**姓 名\_\_\_\_\_\_\_ 邢思洋\_\_ \_**

**学 号 3018216242**

**班 级\_\_\_\_\_\_ 软工5班\_\_\_\_\_\_**

目录

[1．引言 3](#_Toc42636039)

[1.1编写目的 3](#_Toc42636040)

[1.2项目开发要求 3](#_Toc42636041)

[1.3 文档目的 3](#_Toc42636042)

[2．任务概述 3](#_Toc42636043)

[2.1目标 3](#_Toc42636044)

[2.2 硬/软件设备 4](#_Toc42636045)

[3．用例图说明 4](#_Toc42636046)

[3.1 用例图 4](#_Toc42636047)

[3.2 用例 7](#_Toc42636048)

[4.性能需求 10](#_Toc42636049)

[4.1时间特性 10](#_Toc42636050)

[4.2灵活性 10](#_Toc42636051)

[4.3数据库管理要求 10](#_Toc42636052)

[5.运行需求 11](#_Toc42636053)

[6. 其他需求 11](#_Toc42636054)

[7 类设计 11](#_Toc42636055)

[7.1 确定类 11](#_Toc42636056)

[7.2 确定类之间的关系 14](#_Toc42636057)

[7.3 绘制类图 15](#_Toc42636058)

[8. 详细设计 18](#_Toc42636059)

[8.1 客户端用户登录/注册模块 18](#_Toc42636060)

[8.1.1 用户注册 18](#_Toc42636061)

[8.1.2用户登录 19](#_Toc42636062)

[8.2身份验证模块 21](#_Toc42636063)

[8.2.1 用户登录身份验证： 21](#_Toc42636064)

[8.2.2 用户注册身份验证： 23](#_Toc42636065)

[8.3 好友管理模块 25](#_Toc42636066)

[8.3.1 好友查询 25](#_Toc42636067)

[8.3.2 好友申请 27](#_Toc42636068)

[8.3.3 好友删除 29](#_Toc42636069)

[8.4 客户端聊天会话模块 31](#_Toc42636070)

[8.4.1发送聊天信息 31](#_Toc42636071)

[8.4.2 接收聊天 33](#_Toc42636072)

[8.5 服务器聊天会话模块 35](#_Toc42636073)

[8.5.1 服务器接收聊天请求 35](#_Toc42636074)

[8.5.2 服务器转发聊天 36](#_Toc42636075)

[8.6 服务器端好友管理模块 38](#_Toc42636076)

[8.6.1 查询好友 38](#_Toc42636077)

[8.6.2 申请好友 40](#_Toc42636078)

[8.6.3 删除好友 42](#_Toc42636079)

[9. 人员分工 44](#_Toc42636080)

# 1．引言

## 1.1编写目的

此需求分析文档对“即时聊天系统”做了全面细致的用户需求分析，明确所要开发的软件应具有的功能、性能与界面，使系统分析人员及软件开发人员能清楚地了解用户的需求，并在此基础上进一步提出概要设计说明书和完成后续设计与开发工作。该管理系统主要根据现代人们通过网络即时通讯交流的需求进行设计，并充分了解了传统通讯交流系统的特点及不足。进而对该系统进行功能定位和模块划分。、

编写该文档的目的在于明确即时聊天系统开发的需求，使得软件开发人员与用户对待开发软件的需求有统一的认识。进而使项目余下各个工作流程合理有序进行。同时，本文档可以作为测试成果的依据。

1.2项目开发要求

（1）项目开发规范统一： 模块划分，代码编写均遵照小组命名规范文档； （2）程序优化、安全并要有良好的可扩展性；

（3）用户界面简洁明了、操作简单实用；

（4）与用户保持良好的沟通，及时根据用户新的需求改善系统功能。

## 1.3 文档目的

为明确产品需求，将功能性需求、非功能性需求用准确的描述语言描述清楚，并建立相应的静态模型和动态模型。以便于项目组成员对项目目标有清晰的认识。

# 2．任务概述

## 2.1目标

本项目目的是为了设计一款简易方便的即时聊天系统。利用现代的网络资源优势和技术优势，提供完善的网上聊天系统的管理，来方便满足客户对于同特定好友进行聊天通讯的需求。

为了满足用户需求，该即时聊天系统需要满足，用户不受局域网限制，通过互联网进行好友间的即时通讯，拥有美观的UI界面等用户的显著需求。

## 2.2 硬/软件设备

服务器配置标准：未知

客户端配置标准：未知

所需软件：

数据库：Mysql数据库

开发工具：Eclipse

绘图工具：Microsoft Visio Professional 2013

processOn在线版

# 3．用例图说明

## 3.1 用例图

1. 客户端用户登录/注册用例图



1. 身份验证用例图



1. 好友管理用例图



1. 客户端聊天会话用例图



1. 服务器聊天会话用例图



1. 服务器端好友管理用例图



## 3.2 用例

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 1.1 | 参与者 | 用户 |
| 名称 | 用户注册 | | |
| 描述 | 用户在客户端注册时，向客户端发送请求检查是否该用户已注册。若已注册，返回该用户已注册信息；若没有注册，则进行注册。用户注册需要填写用户名，密码，性别，电话号等信息。 | | |
| 前置条件 | 收到用户注册信息 | | |
| 结束状况 | 1. 用户注册成功 2. 用户注册失败 | | |
| 说明 | 无 |  | 用户 |

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 1.2 | 参与者 | 用户 |
| 名称 | 用户登录 | | |
| 描述 | 用户登录及时聊天系统，输入ID与个人密码，发起登录请求。检查用户信息是否正确，如不正确，返回错误，让用户重新验证，如正确，则成功登录。 | | |
| 前置条件 | 收到用户名密码信息 | | |
| 结束状况 | 1. 密码和用户信息匹配登录成功，返回结果 2. 密码和用户信息不匹配，登陆失败，返回结果 | | |
| 说明 | 陌生用户正常登录 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 2.1 | 参与者 | 服务器 |
| 名称 | 验证登录请求 | | |
| 描述 | 受到验证请求时，通过对数据库内的数据进行比对，判断发送请求中的登录信息是否正确，将验证结果返回 | | |
| 前置条件 | 收到验证请求 | | |
| 结束状况 | 1. 验证正确，返回结果 2. 验证不正确返回结果 | | |
| 说明 | 验证用户信息 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 2.2 | 参与者 | 服务器 |
| 名称 | 验证注册请求 | | |
| 描述 | 受到验证请求时，通过对数据库内的数据进行比对，判断发送请求中的注册信息是否正确，将验证结果返回 | | |
| 前置条件 | 收到验证请求 | | |
| 结束状况 | 1. 查无此人，可以注册  2. 已有此人，拒绝注册 | | |
| 说明 | 验证注册信息 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 3.1 | 参与者 | 用户 |
| 名称 | 查询好友 | | |
| 描述 | 在客户端进行好友查询 | | |
| 前置条件 | 用户登录 | | |
| 结束状况 | 1. 查询操作执行完成，返回结果 | | |
| 说明 | 执行好友的查询操作 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 3.2 | 参与者 | 系统 |
| 名称 | 申请好友 | | |
| 描述 | 在客户端进行好友申请 | | |
| 前置条件 | 用户登录 | | |
| 结束状况 | 1. 申请操作执行完成，返回结果 | | |
| 说明 | 执行好友的申请操作 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 3.3 | 参与者 | 系统 |
| 名称 | 删除好友 | | |
| 描述 | 在客户端进行好友删除 | | |
| 前置条件 | 用户登录 | | |
| 结束状况 | 1. 删除操作执行完成，返回结果 | | |
| 说明 | 执行好友的删除操作 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 4.1 | 参与者 | 用户 |
| 名称 | 发送聊天信息 | | |
| 描述 | 用户在客户端发送聊天信息，判断好友是否在线，如果离线，返回失败，反之则发送给好友，返回成功 | | |
| 前置条件 | 好友列表中找到对应好友 | | |
| 结束状况 | 1. 好友在线，发送信息，返回结果 2. 好友离线，发送失败，返回结果 | | |
| 说明 | 进行即时通讯 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 4.2 | 参与者 | 用户 |
| 名称 | 接收聊天 | | |
| 描述 | 在客户端接收聊天信息 | | |
| 前置条件 | 在好友列表中找到对应好友 | | |
| 结束状况 | 接收到好友发送的信息 | | |
| 说明 |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 5.1 | 参与者 | 服务器 |
| 名称 | 服务器接收客户端发来的聊天请求 | | |
| 描述 | 服务器接收客户端发来的聊天请求，解析聊天信息中的内容和发生对象 | | |
| 前置条件 | 接收聊天信息请求 | | |
| 结束状况 | 返回解析结果 | | |
| 说明 | 无 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 5.2 | 参与者 | 服务器 |
| 名称 | 转发聊天 | | |
| 描述 | 根据解析的信息，向对应的好友客户端转发信息，返回转发的结果。 | | |
| 前置条件 | 成功解析聊天信息请求 | | |
| 结束状况 | 返回转发结果 | | |
| 说明 | 无 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 6.1 | 参与者 | 服务器 |
| 名称 | 查询好友 | | |
| 描述 | 接收到管理请求，根据请求中的信息，执行查询好友操作，返回查询结果。 | | |
| 前置条件 | 收到好友管理的请求，权限为用户 | | |
| 结束状况 | 1.查询操作执行完成，返回结果 | | |
| 说明 | 执行好友的查询操作 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 6.2 | 参与者 | 服务器 |
| 名称 | 申请好友 | | |
| 描述 | 接收到管理请求，根据请求中的信息，执行添加好友操作，返回结果。 | | |
| 前置条件 | 收到好友管理的请求，权限为用户 | | |
| 结束状况 | 1.添加操作执行完成，返回结果 | | |
| 说明 | 执行好友的添加请求操作 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 6.3 | 参与者 | 服务器 |
| 名称 | 删除好友 | | |
| 描述 | 接收到管理请求，根据请求中的信息，执行删除好友操作，返回结果。 | | |
| 前置条件 | 收到好友管理的请求，权限为用户 | | |
| 结束状况 | 1.删除操作执行完成，返回结果 | | |
| 说明 | 执行好友的删除操作 | | |

# 4.性能需求

## 4.1时间特性

1.查询聊天数据：响应时间低于3秒

2.更新信息数据：用户提交信息，查询/添加/修改/删除好友信息等响应时间低于3秒

3.接收数据信息：用户发送给好友消息，对方接收到好友消息响应时间低于1秒

## 4.2灵活性

1. 操作方式上的变化：鼠标点击和键盘输入综合

2．基于java优秀的可移植的性，能够在多种平台上使用

## 4.3数据库管理要求

现代存储技术发达，数据库容量应该为T级别，以保证工作信息交流能够存储以备日后需要。

# 5.运行需求

本系统采用java平台，基于JDBC连接MySQL数据库,利用TCP/IP，Socket 协议实现信息交互。

# 6. 其他需求

1. 操作难易度低，易于使用。各种用户端界面应该时简洁的，可以模拟现在市面上主流的聊天，信息反馈软件系统等界面，来达到此目的。
2. 可靠性高。本系统能在遇到各式错误时做出相应的处理并且安全退出，不导致数据丢失。在实际使用过程中，遇见网络问题，传输文件格式、大小问题，服务器端出错，服务器端不在线或其他不可知的错误时，能够提示用户以正确的操作退出系统或以安全方式自行退出。
3. 对系统代码进行合理的模块化设计，提高本系统的可移植性和可拓展性。不同类的用户端应该有不同的功能，不同的模块支持不同的功能，方便在日后进行系统维护，功能添加等行为。

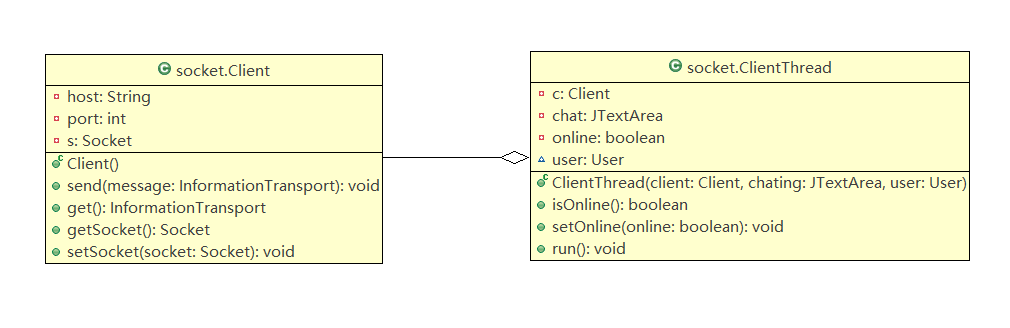
# 7 类设计

## 7.1 确定类

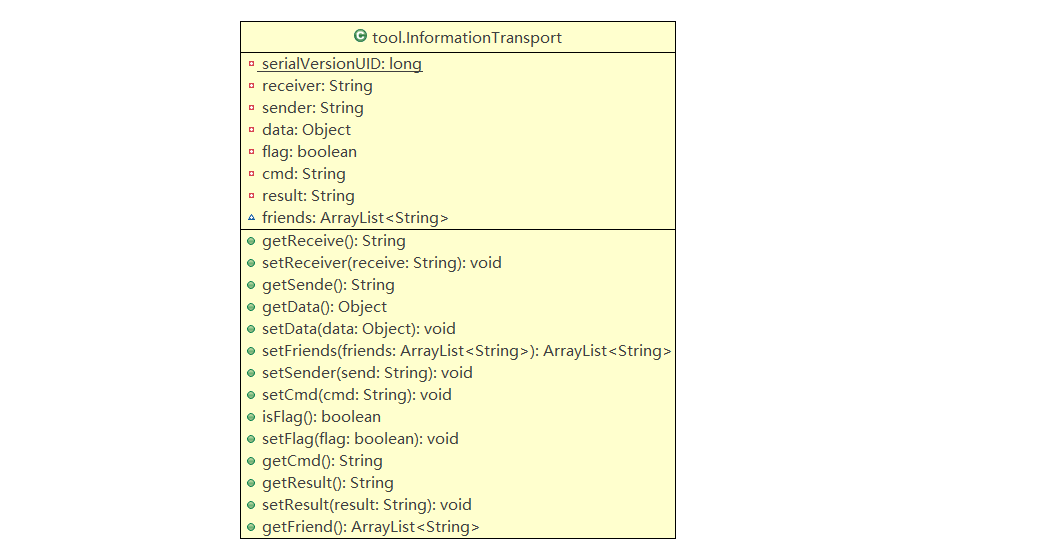
1. 客户端用户类



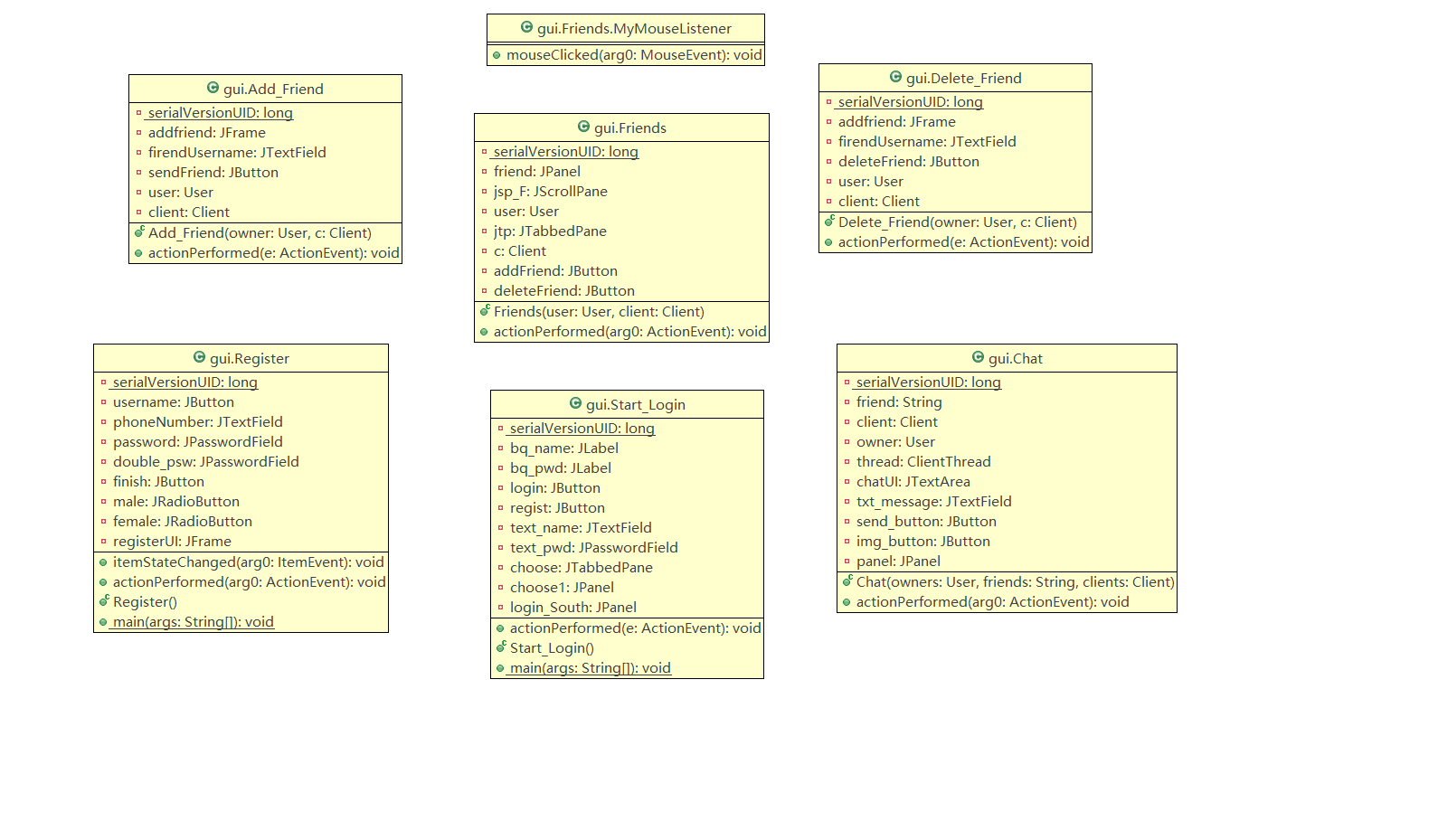
2. 客户端套接字类



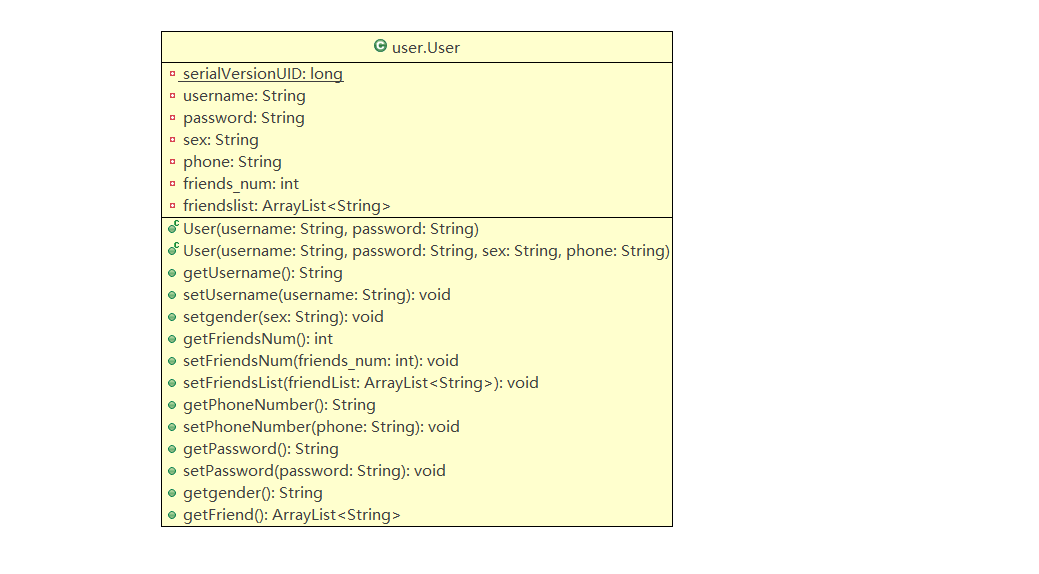
3. 客户端工具类



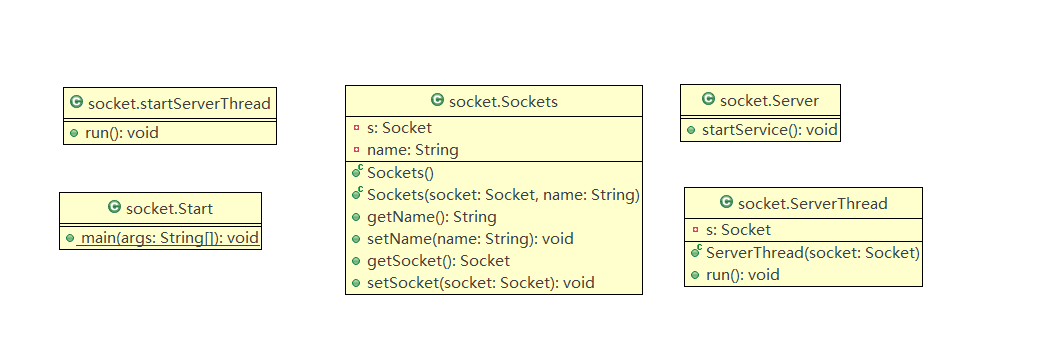
4. 客户端UI界面类



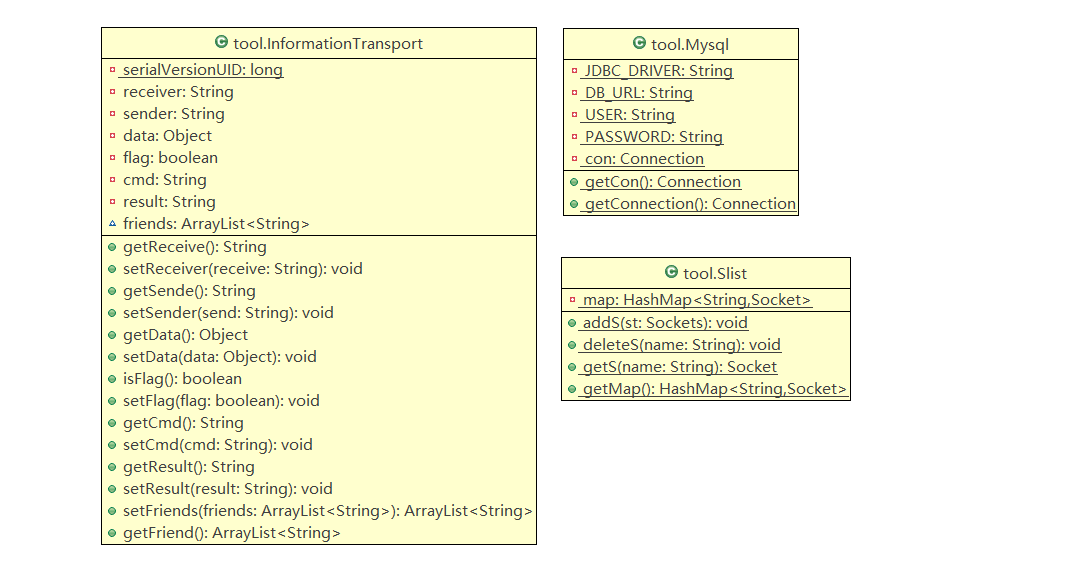
5. 服务器用户类



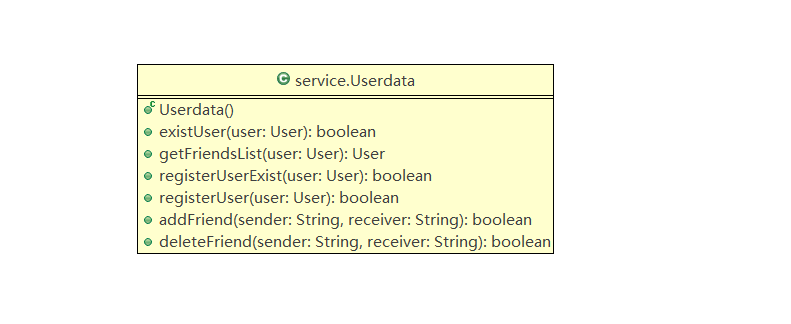
6. 服务器套接字类



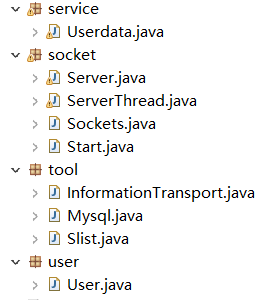
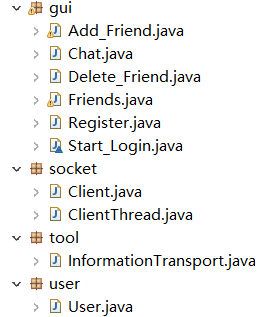
7. 服务器工具类



8. 服务器服务类



## 7.2 确定类之间的关系



1. 客户端gui包（客户端UI界面类）：负责客户端登录的图形界面和实现逻辑：Start\_Login.java是登录界面的实现，在登录界面中，有注册按钮，能够实例化Register.java，负责实现注册账号的界面。Friend.java负责在登录后，生成好友列表的界面，由Start\_Login.java在程序末尾实现。好友列表界面中包含有添加和删除好友按钮，分别实例化了Add\_Firend.java和Delete\_Friend.java类，实现了添加和删除好友的图像界面。Chat.java用于实现与好友聊天的界面，在点击对应好友账号后实现。整体作为View层和用户交互

2. 客户端Socket包（客户端套接字类）：负责实现客户端多线程的socket协议，Client.java生成了socket实例，作为客户端存在。而ClientThread.java继承了Thread类，实现了多线程。同时也负责将客户端的请求命令发送给服务器。整体作为Model层用于处理应用程序数据逻辑。

3. 客户端Tool包（客户端工具类）：包含了InformationTransport.java类，作为信息承载体实例，在客户端和服务器之间传递信息。

4. 客户端 User包（客户端用户类）：包含了user.java类，用于承载用户个人信息在客户端和服务器之间传递信息。

5. 服务器Service包（服务器服务类）：负责进行数据哭操作，包括了Userdata.java类，根据解析后的客户端请求对数据库信息进行操作。

整体作为Controller层，用于处理应用程序中处理用户交互的结果。

6. 服务器Socket包（服务器套接字类）：负责建立服务器的多线程的socket协议包括了Server.java，生成了socket实例。ServerThread.java实现了多线程。同时也负责将客户端的请求进行解析，根据解析结果调用对应的Userdata类的方法，对数据库进行操作编辑。Sockets.java用于保存服务器生产的对应不同客户端的socket实例。Start.java负责实现服务器

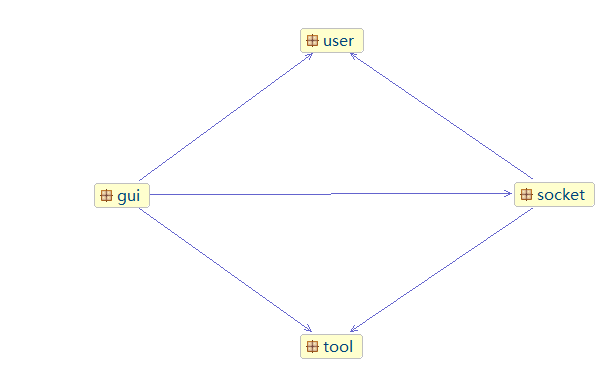
7. 服务器Tool包（服务器工具类）：作为工具包，InformationTransport.java类，作为信息承载体实例，在客户端和服务器之间传递信息。Mysql.java和Slist.java负责实现jdbc和数据库建立联系。

8. 服务器User包（服务器用户类）：包含了user.java类，用于承载用户个人信息在客户端和服务器之间传递信息。

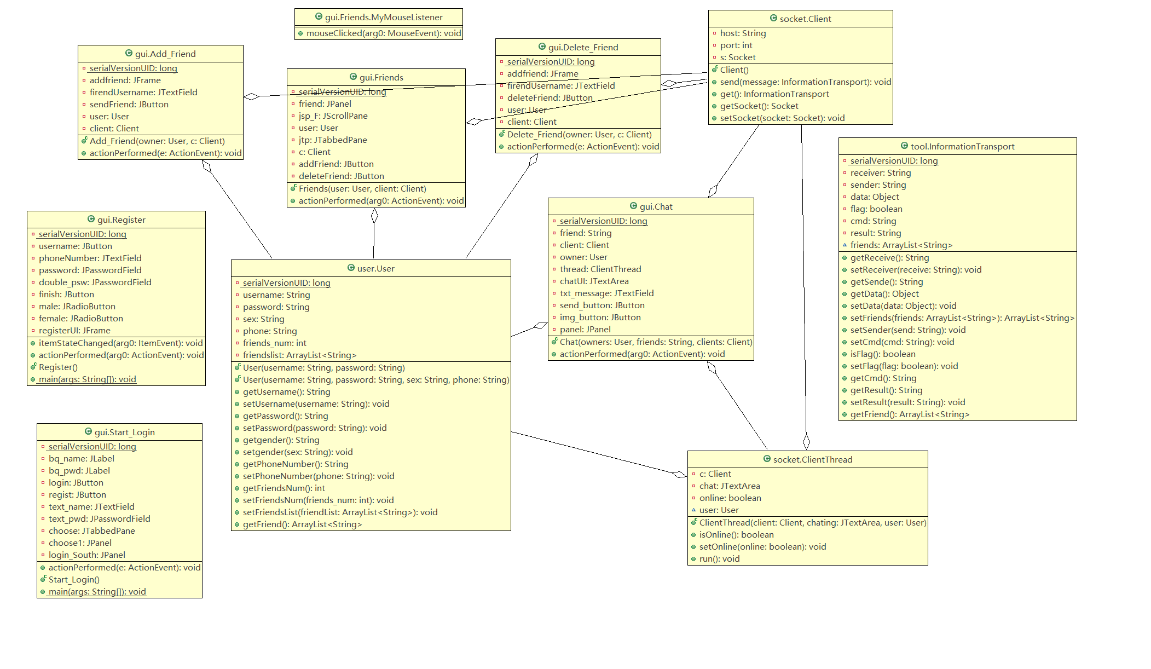
## 7.3 绘制类图

1. 客户端

（1）客户端包依赖图：

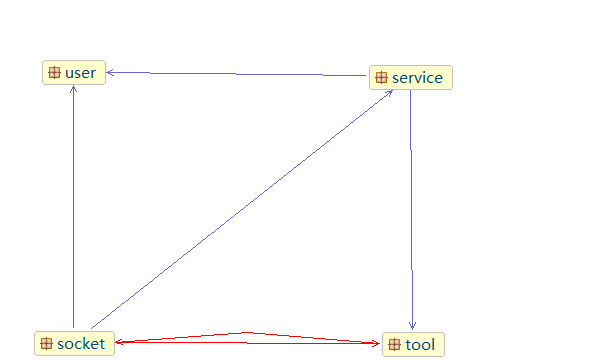


（2）客户端类图

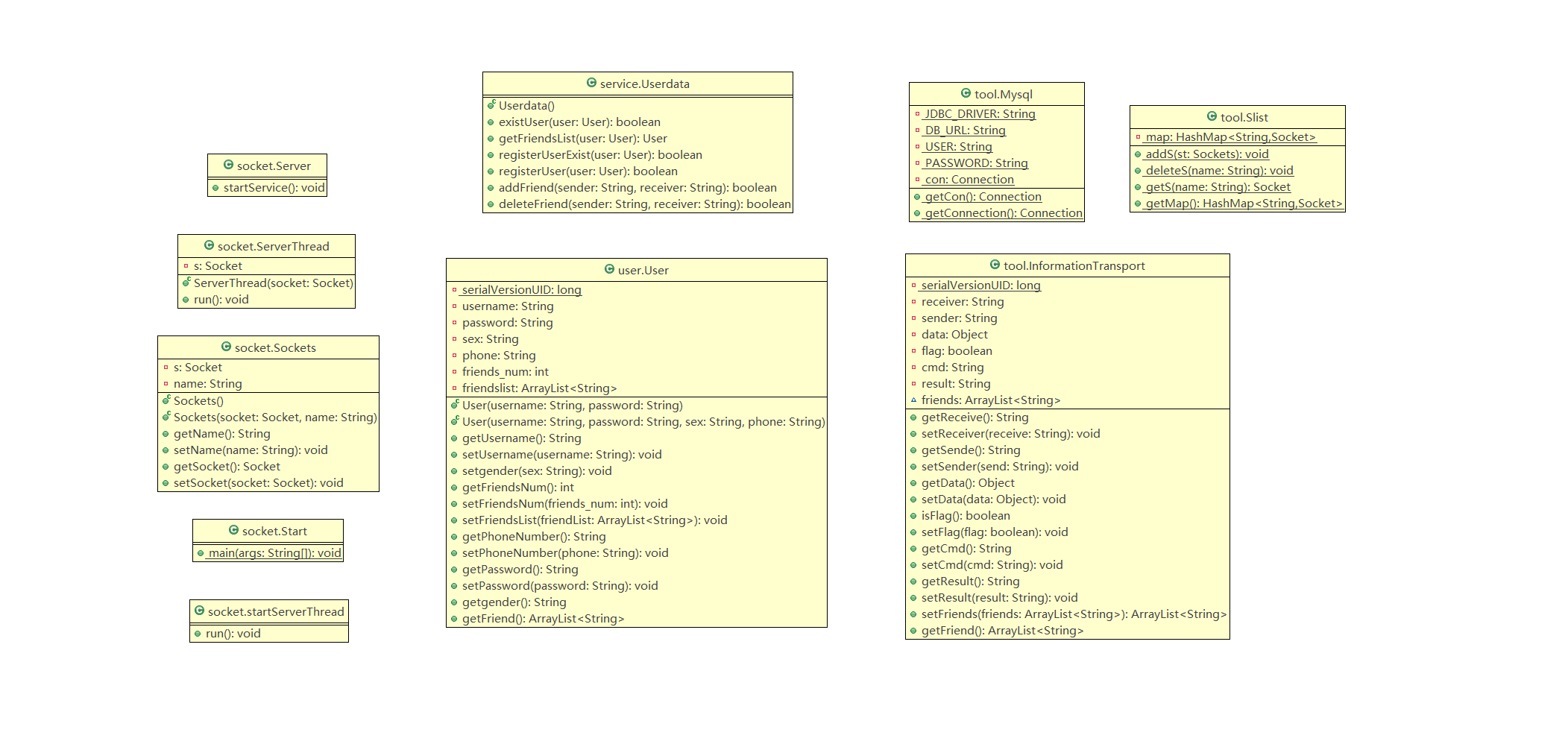


2. 服务器

（1）服务器包依赖图：



（2）服务器端类图

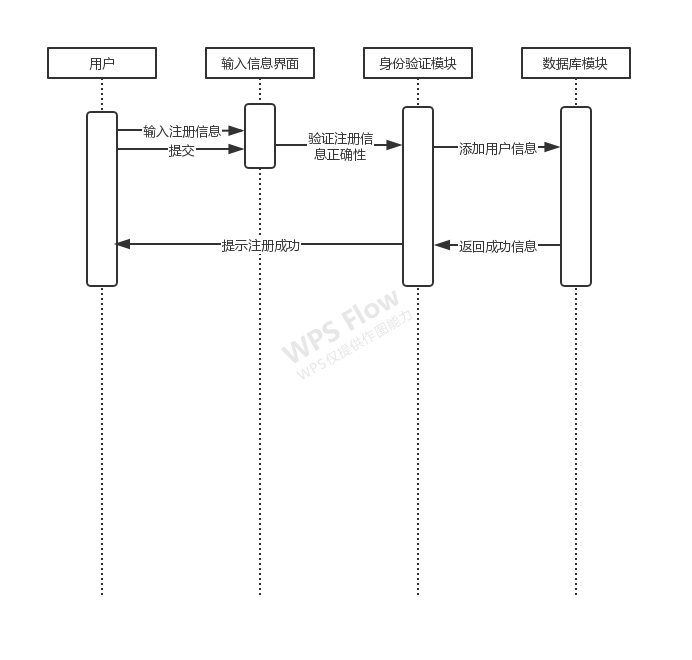


# 详细设计

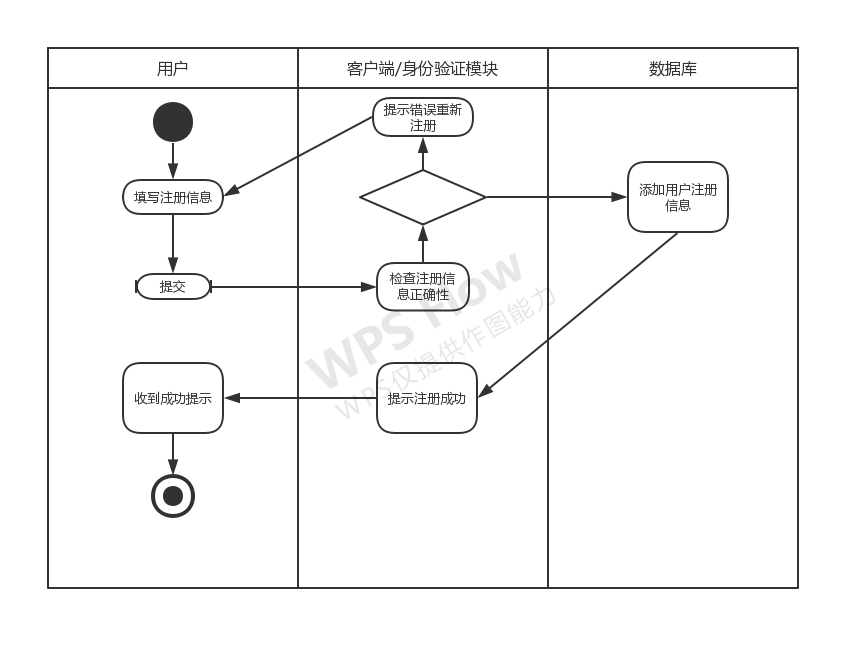
## 8.1 客户端用户登录/注册模块

### 8.1.1 用户注册

顺序图：

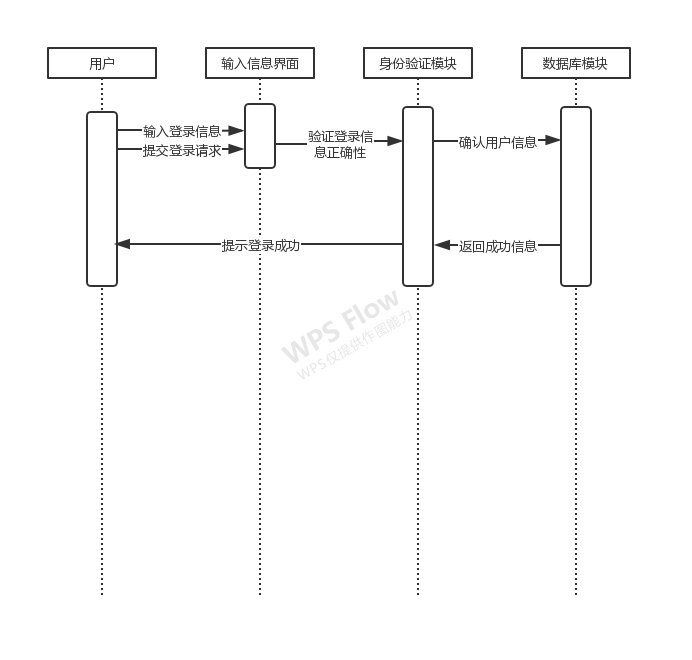


活动图：

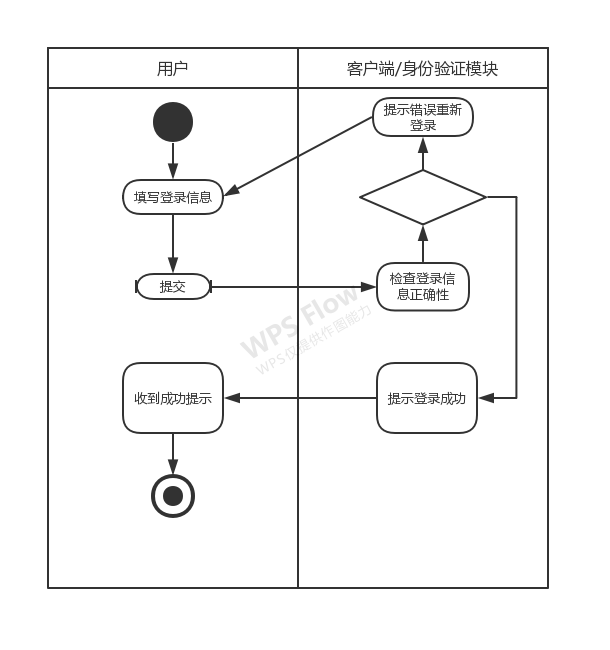


### 8.1.2用户登录

顺序图：



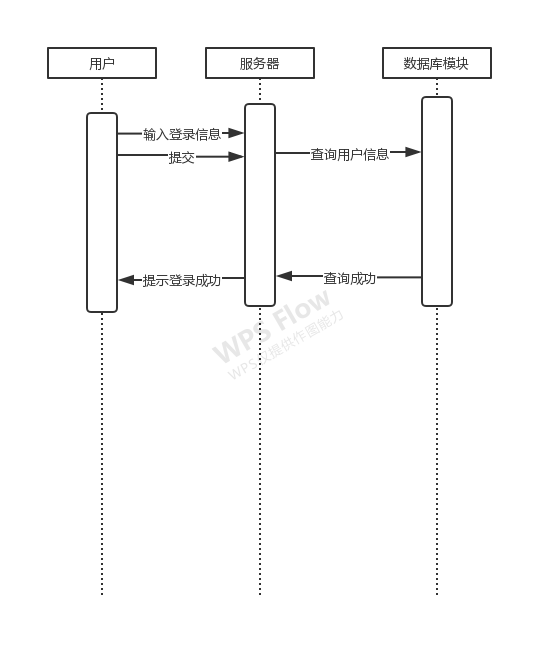
活动图：



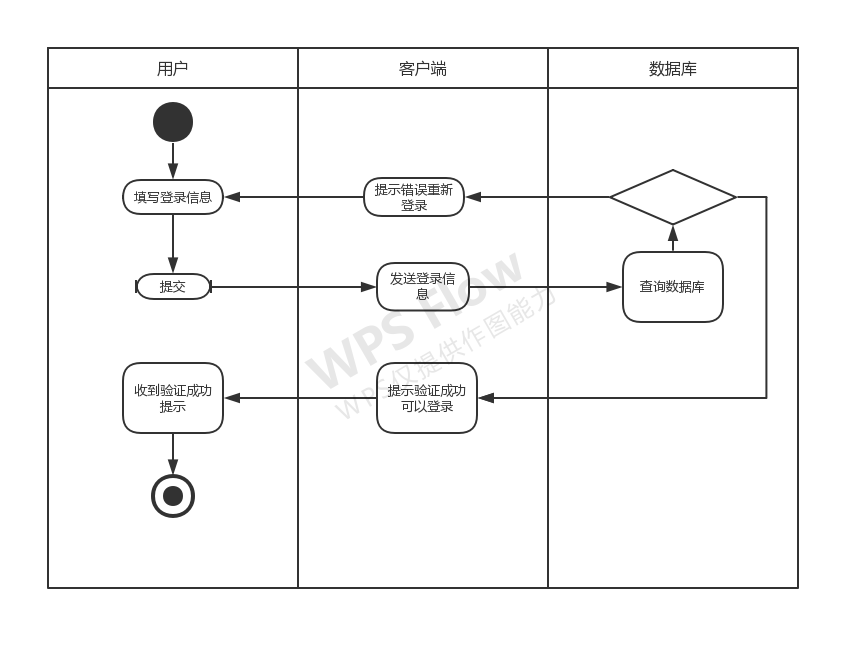
## 8.2身份验证模块

### 8.2.1 用户登录身份验证：

顺序图：

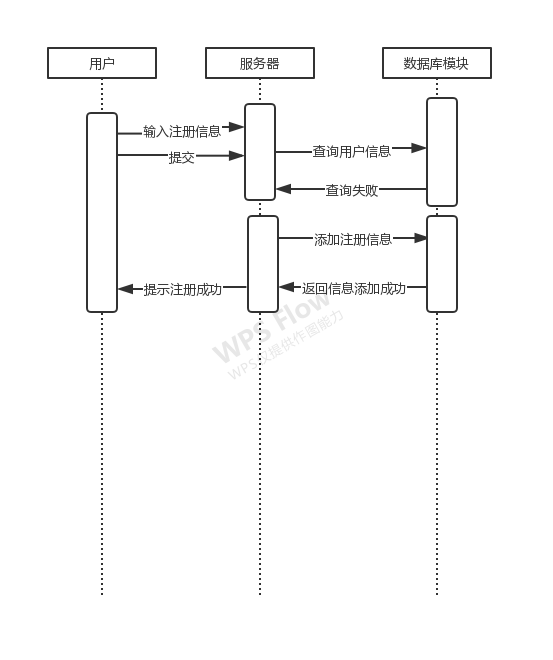


活动图：

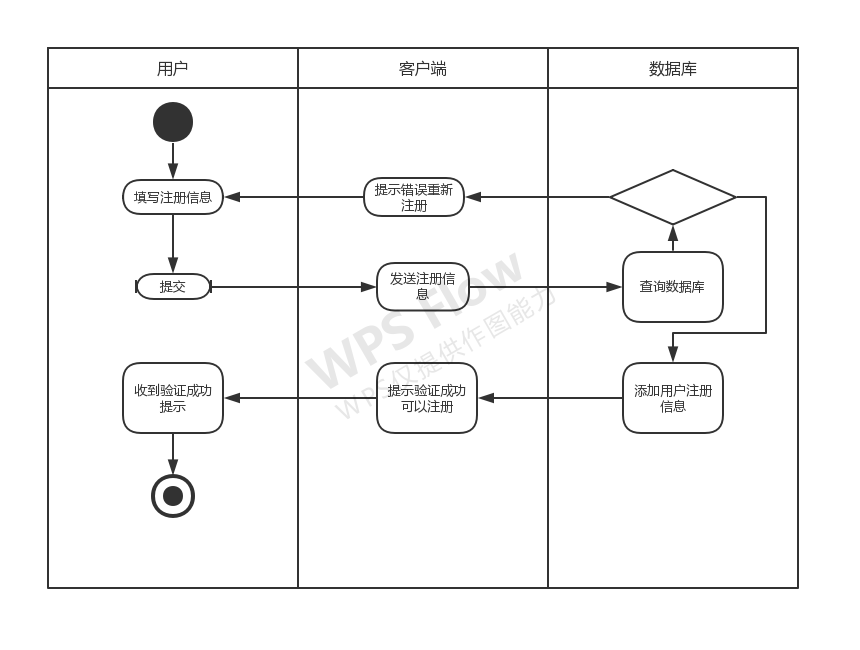


### 8.2.2 用户注册身份验证：

顺序图：



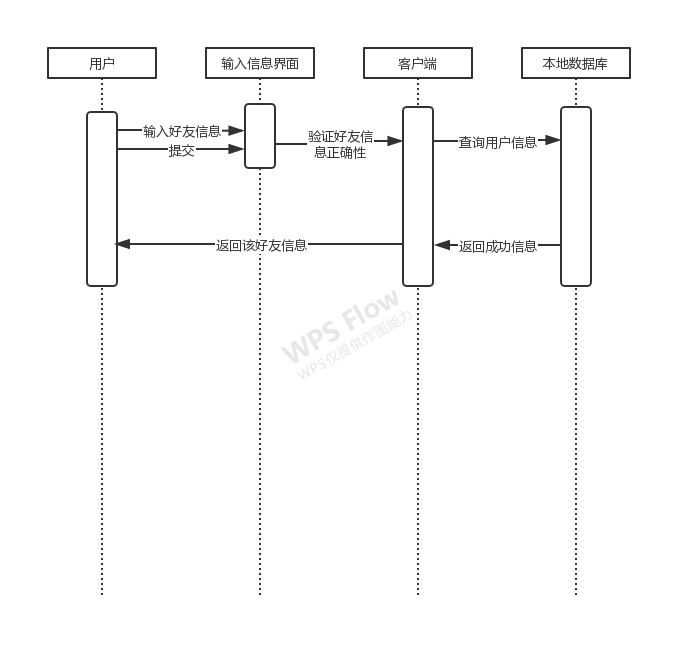
活动图：



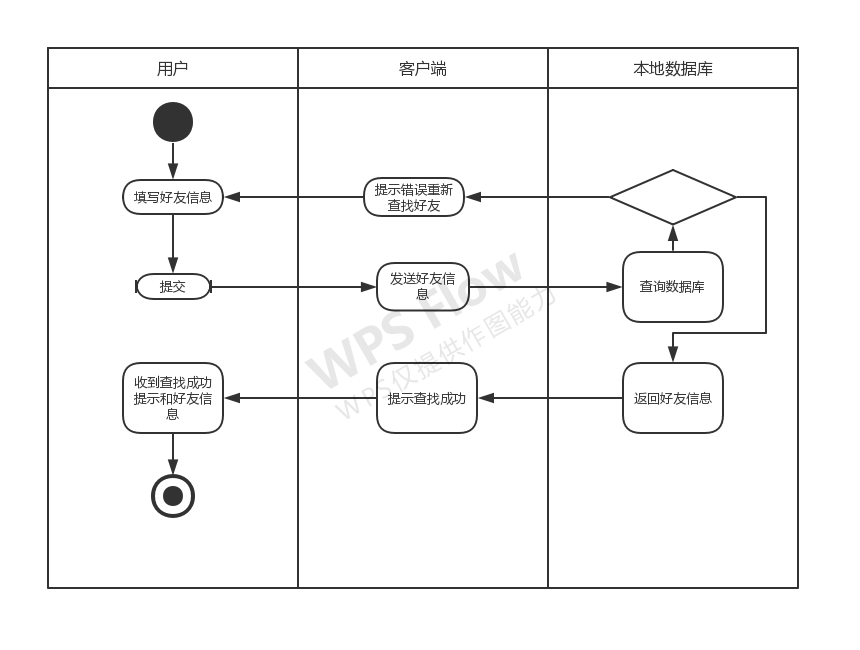
## 8.3 好友管理模块

### 8.3.1 好友查询

顺序图：

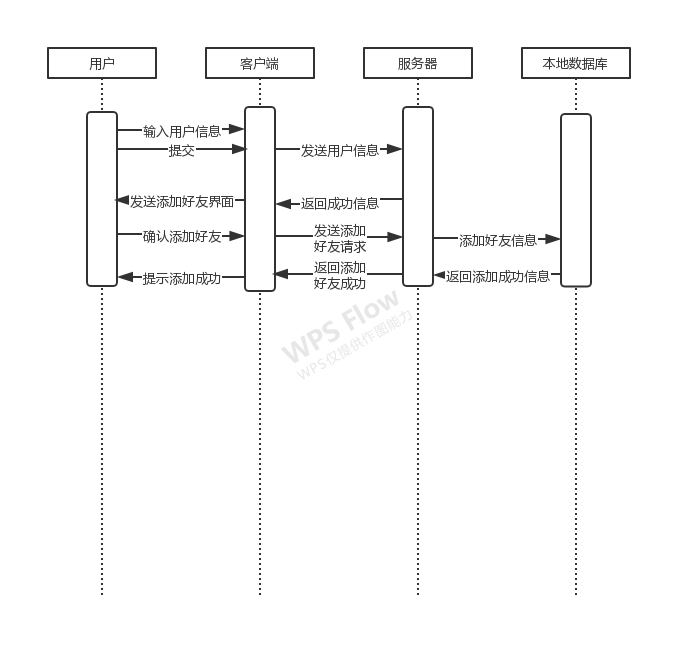


活动图：

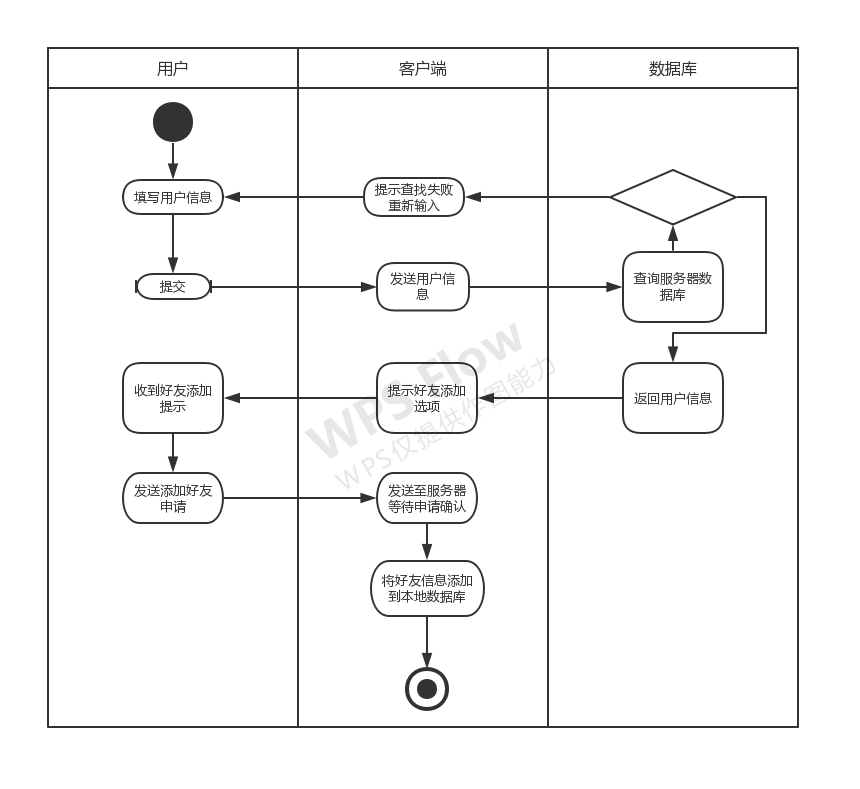


### 8.3.2 好友申请

顺序图：

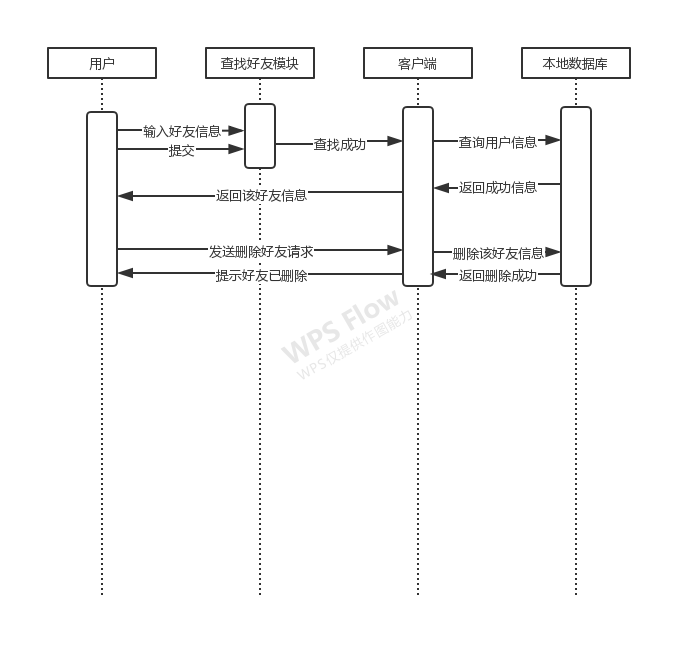


活动图：

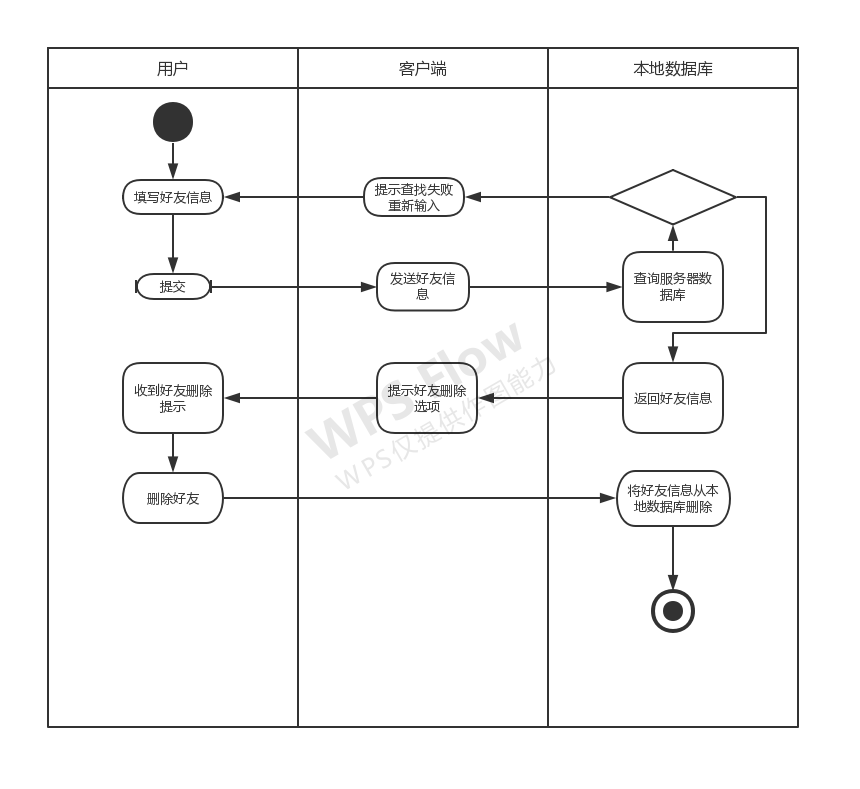


### 8.3.3 好友删除

顺序图：



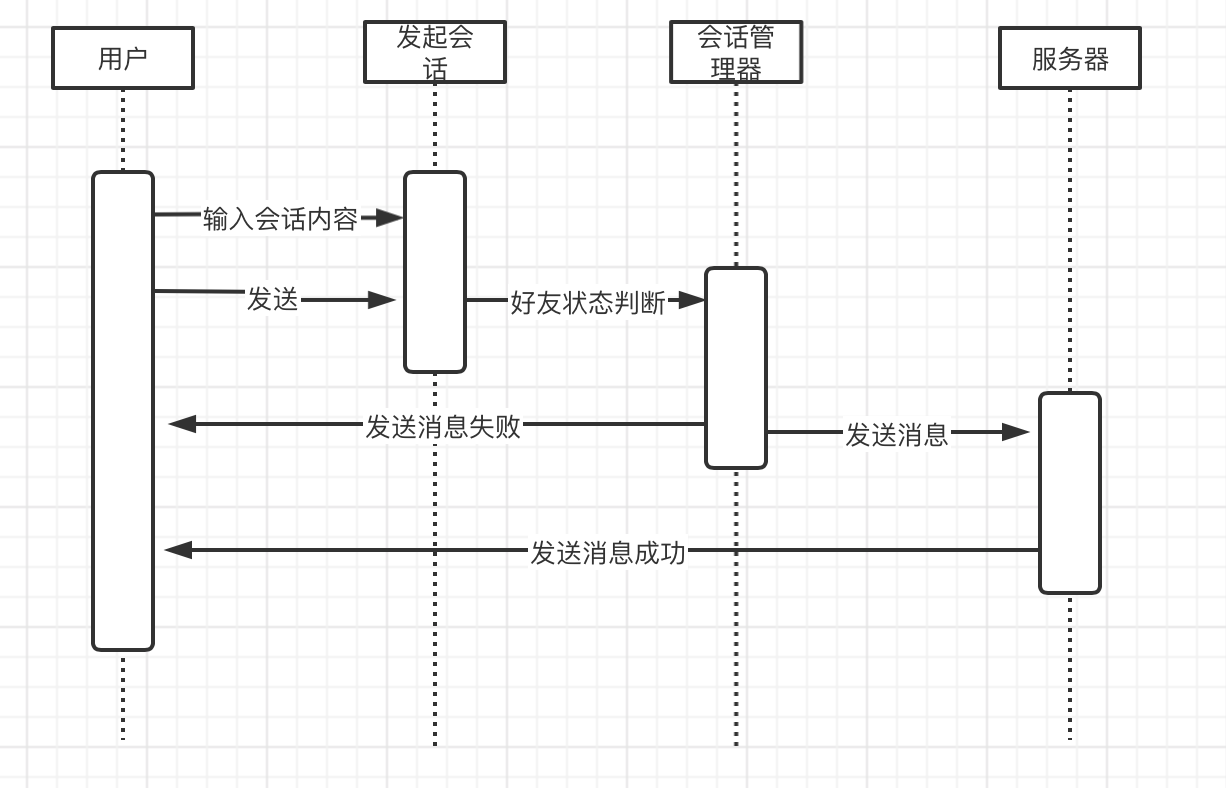
活动图：

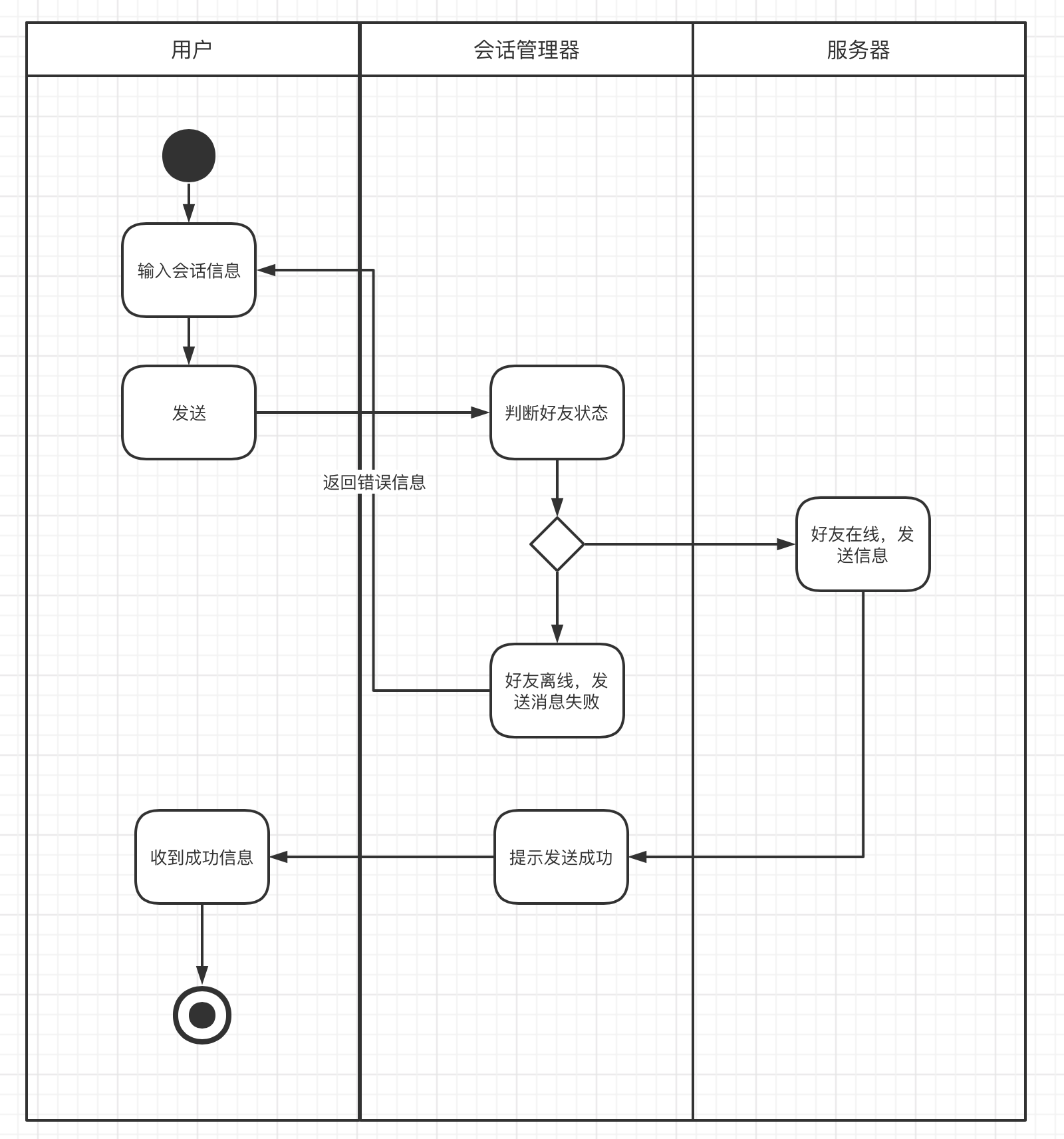


## 8.4 客户端聊天会话模块

### 8.4.1发送聊天信息

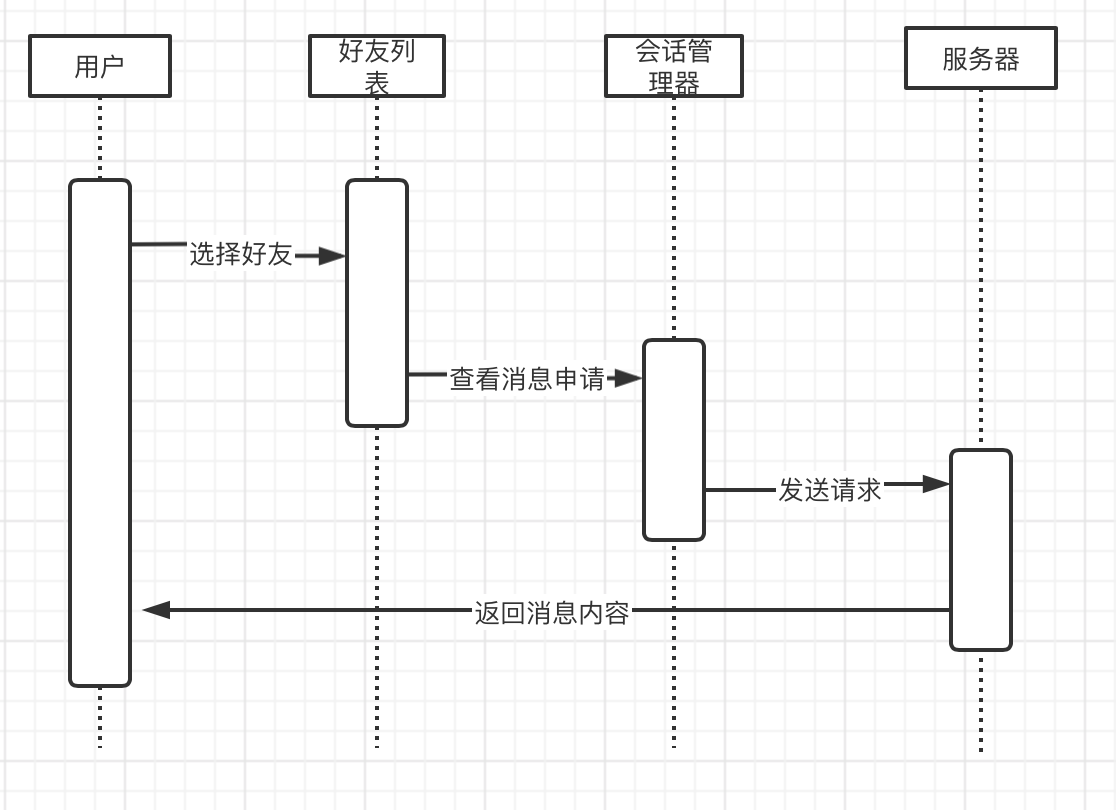
顺序图：

****活动图：

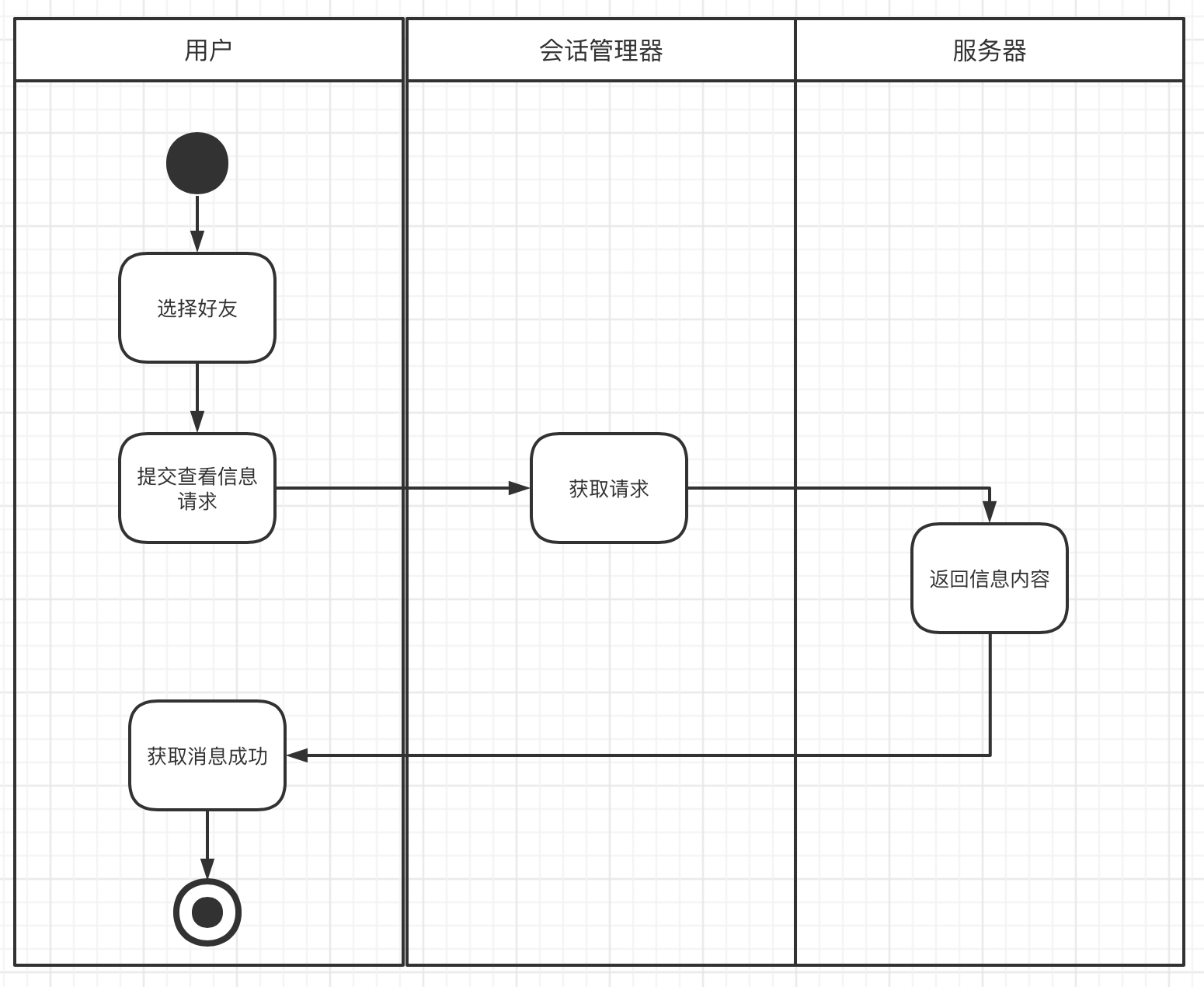
****

### 8.4.2 接收聊天

顺序图：

****

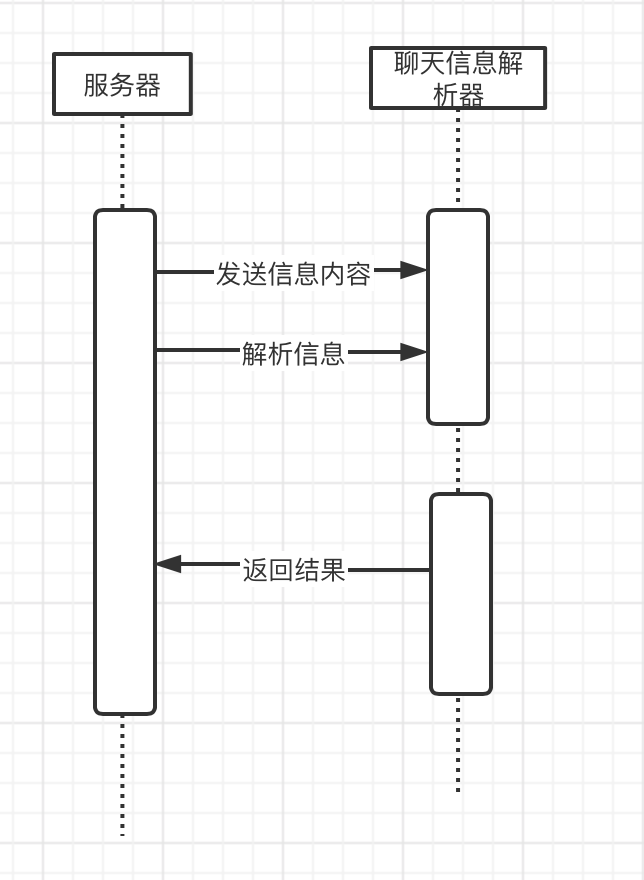
活动图：

****

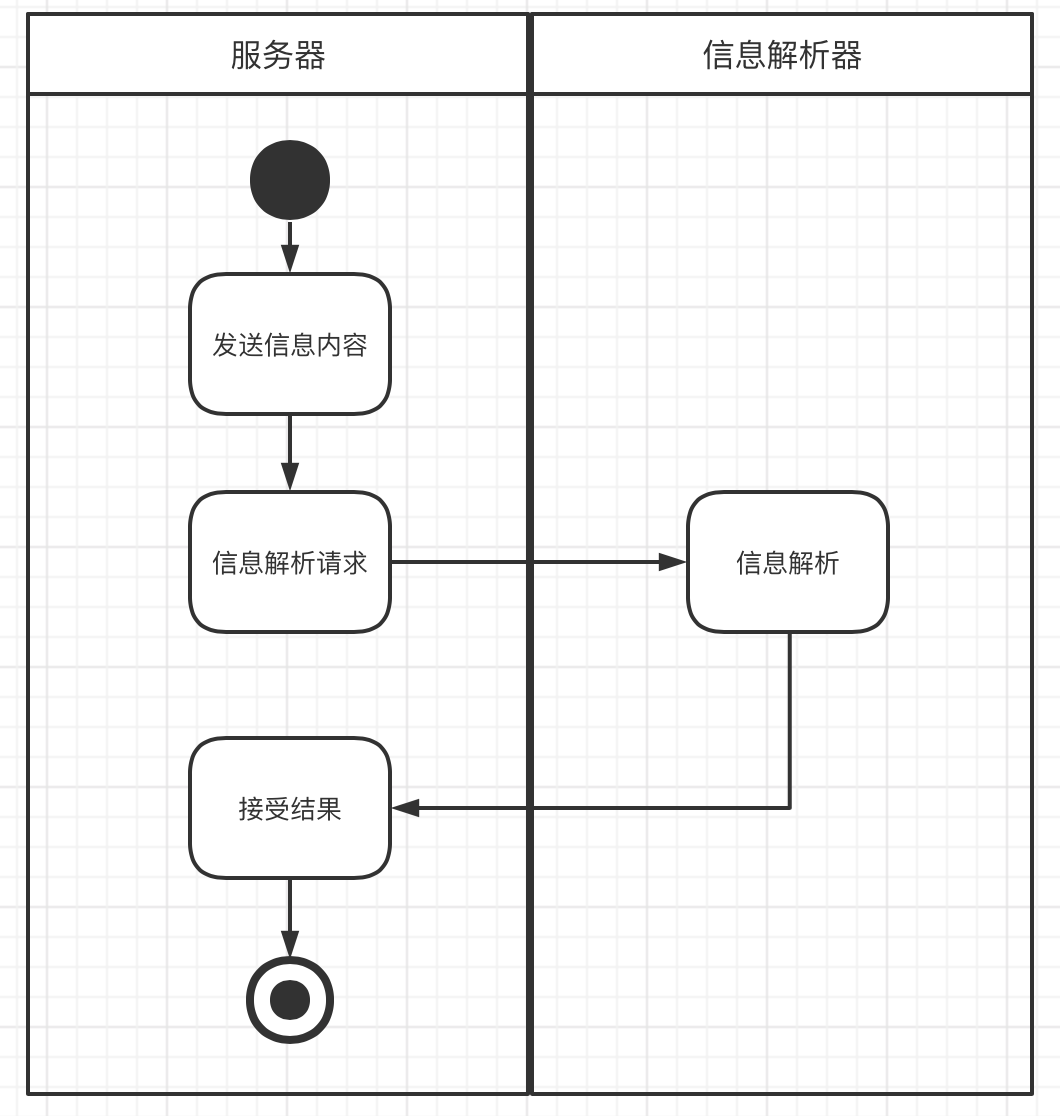
## 8.5 服务器聊天会话模块

### 8.5.1 服务器接收聊天请求

顺序图：

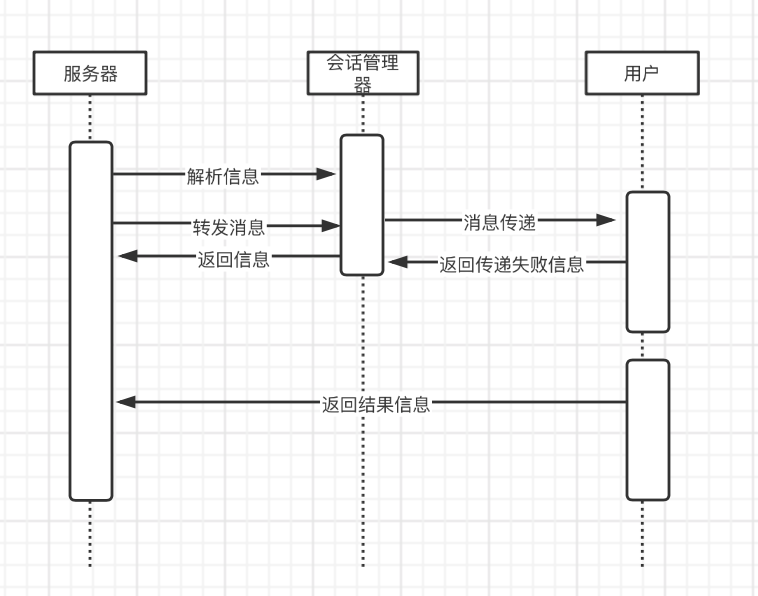
****

活动图：

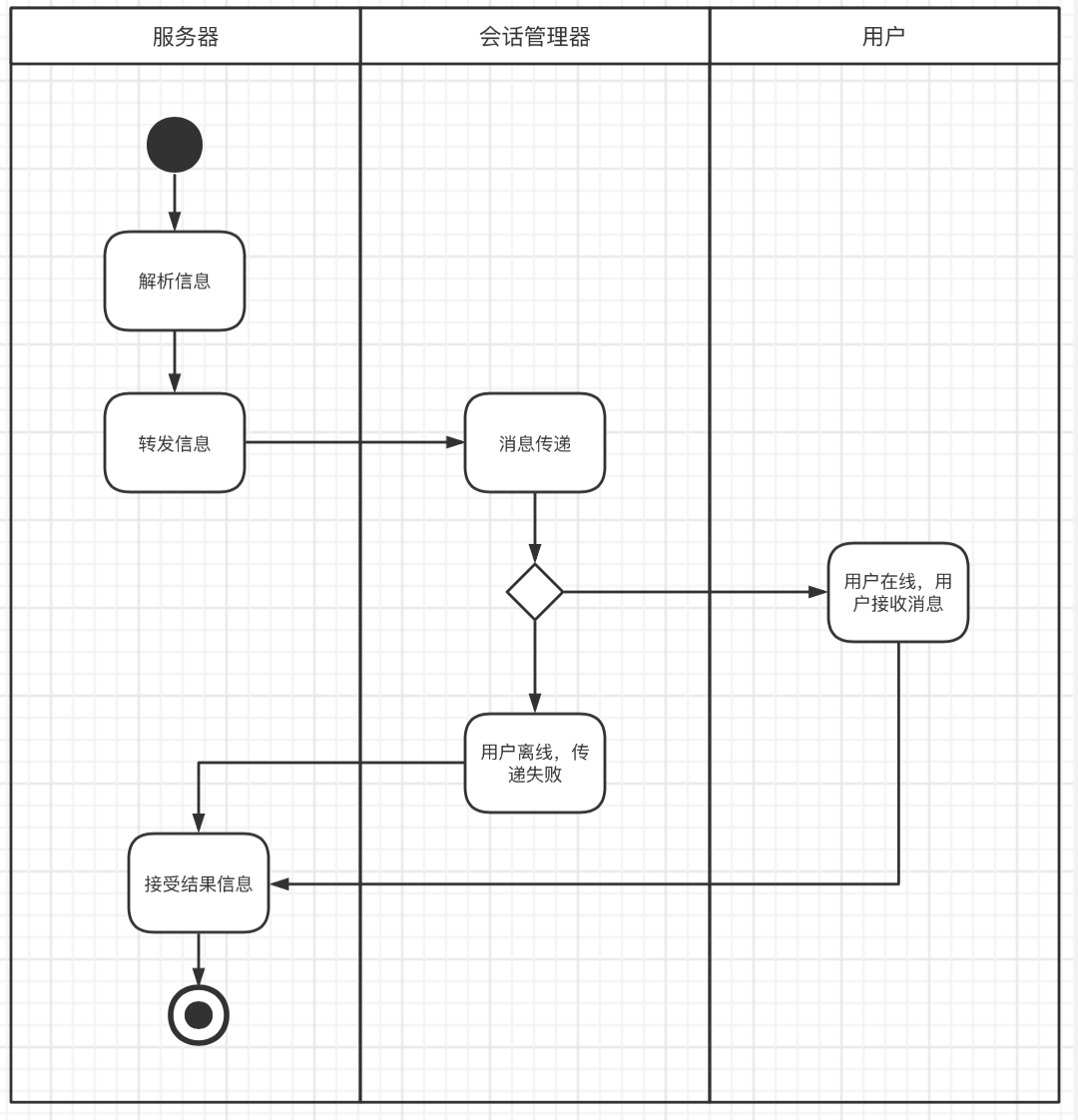
****

### 8.5.2 服务器转发聊天

顺序图：



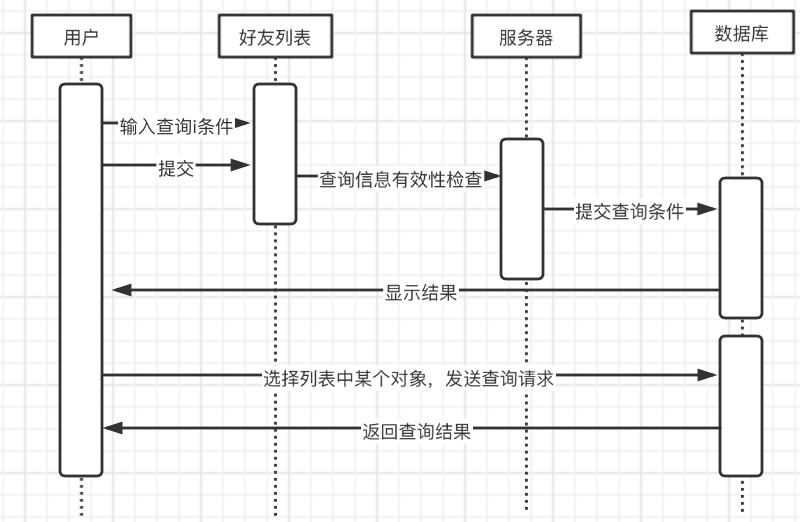
活动图：



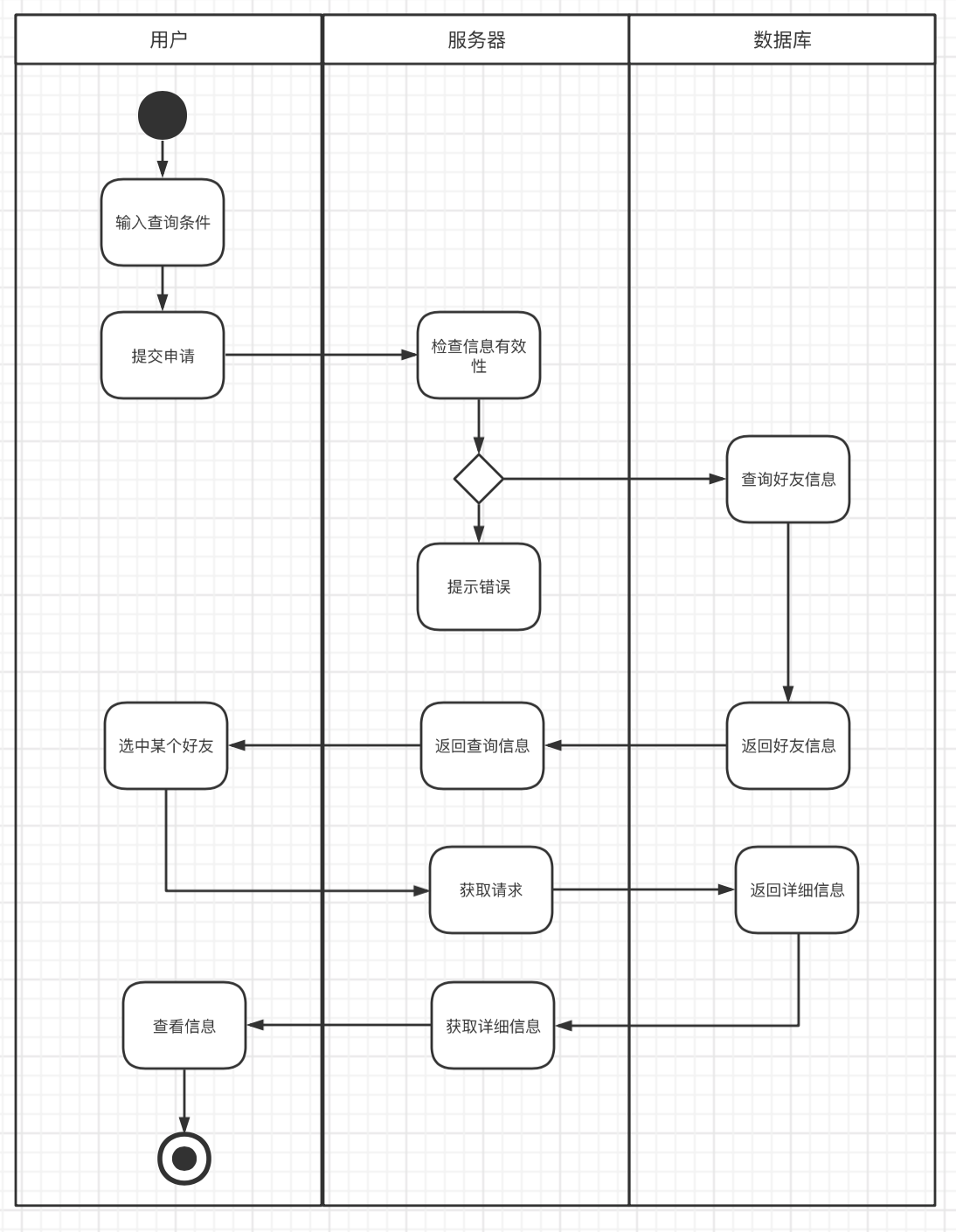
## 8.6 服务器端好友管理模块

### 8.6.1 查询好友

顺序图：

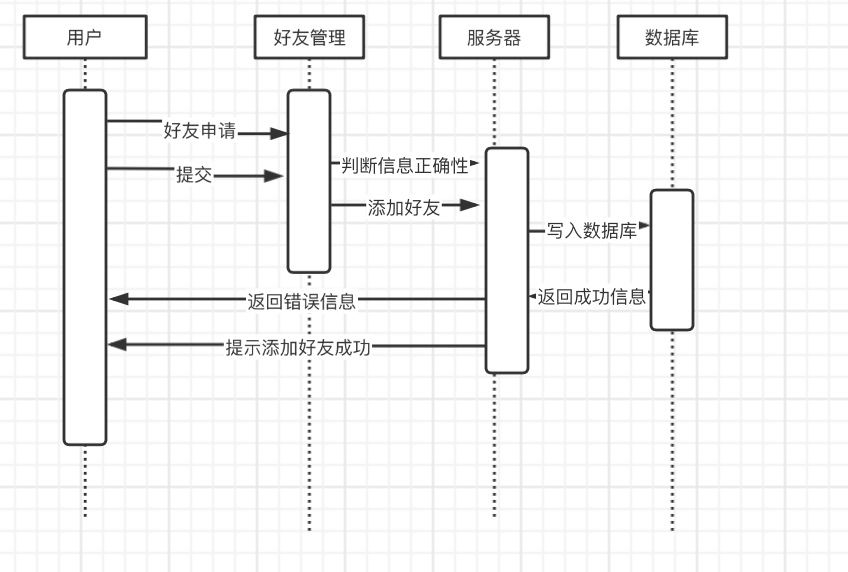


活动图：

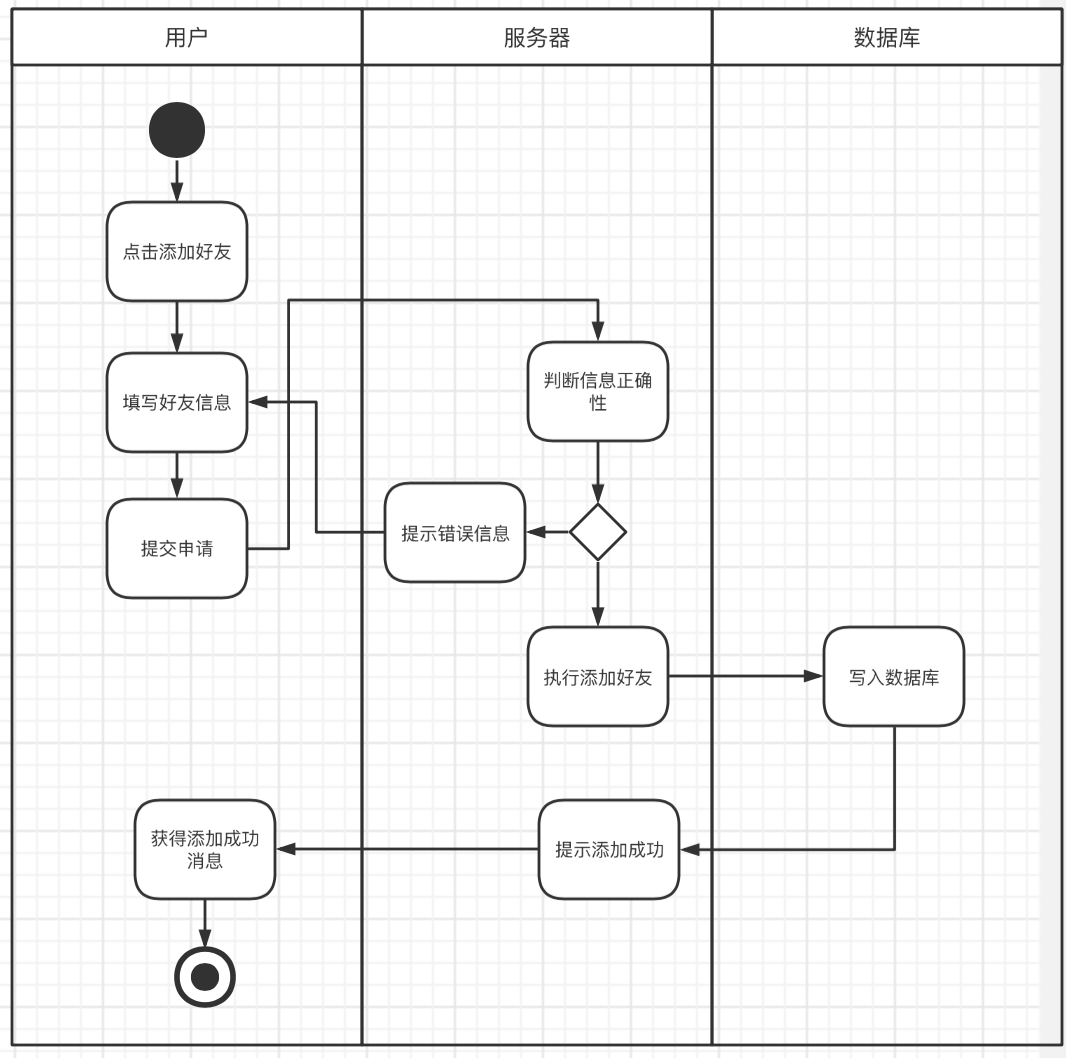


### 8.6.2 申请好友

顺序图：

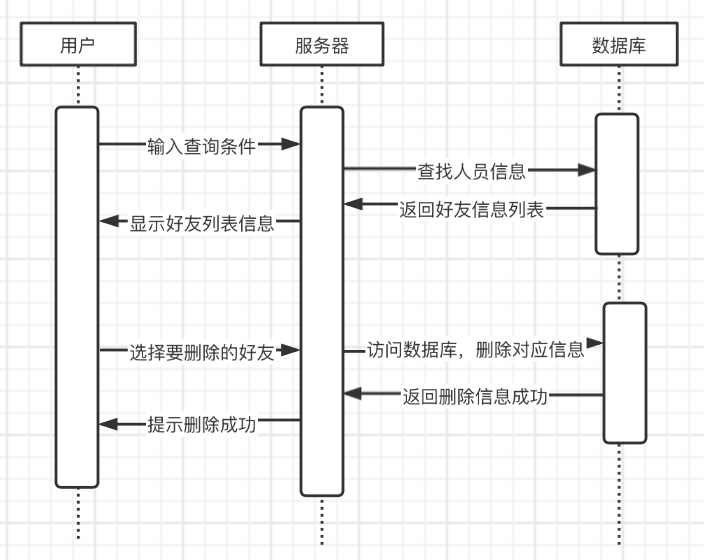


活动图：

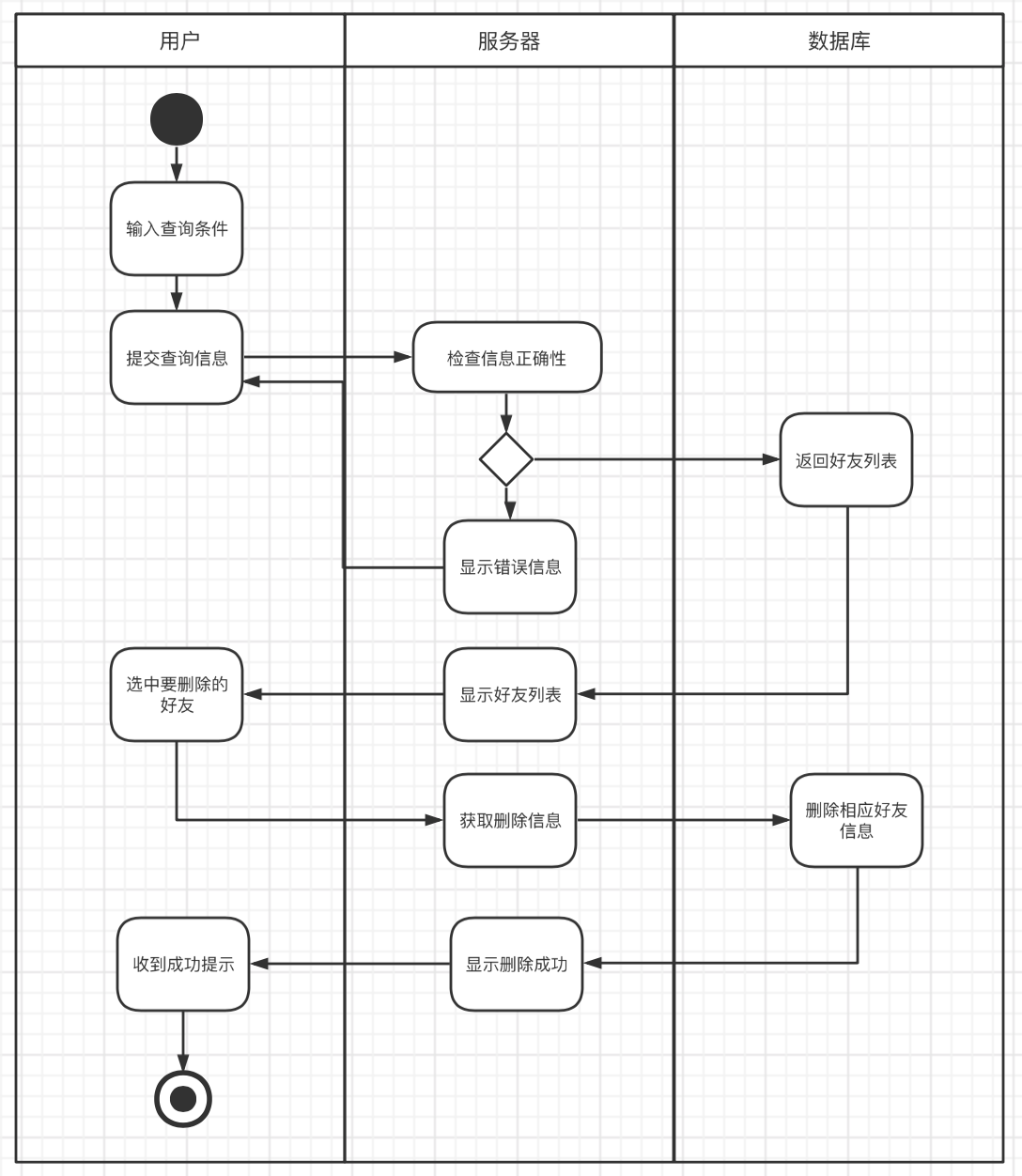


### 8.6.3 删除好友

顺序图：

****

活动图**：**

****

# 人员分工

即时通讯系统人员分工说明：

|  |  |
| --- | --- |
| 赵浩喆 | 负责客户端的图形界面及其实现逻辑的设计：实现了客户端中gui包中的Start\_Login.java是登录界面的实现。在登录界面中，有注册按钮，能够实例化Register.java，负责实现注册账号的界面。Friend.java负责在登录后，生成好友列表的界面，由Start\_Login.java在程序末尾实现。好友列表界面中包含有添加和删除好友按钮，分别实例化了Add\_Firend.java和Delete\_Friend.java类，实现了添加和删除好友的图像界面。Chat.java用于实现与好友聊天的界面，在点击对应好友账号后实现。  负责对最终产生的服务器和客户端之间的连接进行实验和测试，以及对整体项目问题的细微调整。 |
| 郑启俊 | 负责实现客户端和服务器之间的socket通讯协议。实现了客户端Socket包的客户端多线程的socket协议，Client.java生成了socket实例，作为客户端存在。而ClientThread.java继承了Thread类，实现了多线程。以及实现了服务器端的Socket包，用于建立服务器的多线程的socket协议。其中Server.java，生成了socket实例。ServerThread.java实现了多线程。  实现了jdbc功能，负责连接和操作mysql数据库。 |
| 马骁 | 负责实现客户端与服务器端的请求解析和调用。实现了客户端的Socket包中的ClientThread.java中负责将客户端的请求命令发送给服务器的功能。实现了服务器端的Socket包中的ServerThread.java中负责将客户端的请求进行解析，根据解析结果调用对应的Userdata类的方法，对数据库进行操作编辑的功能。  设计并实现了保存用户信息和对应用户好友信息的mysql数据库。 |
| 邢思洋 | 负责对项目的MVC架构进行设计分析，设计了客户端项目的gui包作为View层和用户交互，Socket包作为Model层用于处理应用程序数据逻辑。同时在服务器方面，设计了Service包作为Controller层，Userdata类用于处理应用程序中处理用户交互的结果。实现了Userdata.java，根据解析后的客户端请求对数据库信息进行操作。  设计并实现了Tool包和User包中的InformationTransport.java和user.java作为信息传输的载体进行数据信息和用户个人信息在客户端和服务器之间的传输。 |