**网上沟通交流系统**

需求分析报告

目录

[1引言 2](#_Toc433443386)

[1.1编写目的 2](#_Toc433443387)

[1.2背景 2](#_Toc433443388)

[1.3定义 2](#_Toc433443389)

[1.4参考资料 3](#_Toc433443390)

[2系统结构模型 3](#_Toc433443391)

[2.1初始对象表 3](#_Toc433443392)

[2.2初始类图 4](#_Toc433443393)

[2.3细化类图 5](#_Toc433443394)

[2.3.1属性的识别 5](#_Toc433443395)

[2.3.2 操作的识别 7](#_Toc433443396)

[2.4三层精化类图 11](#_Toc433443397)

[3系统行为模型 15](#_Toc433443398)

[3.1交互模型 15](#_Toc433443399)

[3.2状态模型 16](#_Toc433443400)

概要设计说明书

# 1引言

## 1.1编写目的

需求说明书目的是使用户和软件开发者双方对该软件有一个共同的理解，使之成为整个开发工作的基础。便于用户、开发人员进行理解和交流，反映出用户问题的结构，可以作为软件开发工作的基础和依据，并作为确认测试和验收的依据。

预期读者：软件设计者和测试者。

## 1.2背景

说明：

1. 待开发的软件系统的名称：网上沟通交流系统；
2. 本项目的任务提出者： 柴华溢
3. 本项目的任务开发者： 柴华溢
4. 用户：具有网上沟通交流需求的用户；
5. 实现该软件的计算中心或计算机网络：南京航空航天大学计算中心；

## 1.3定义

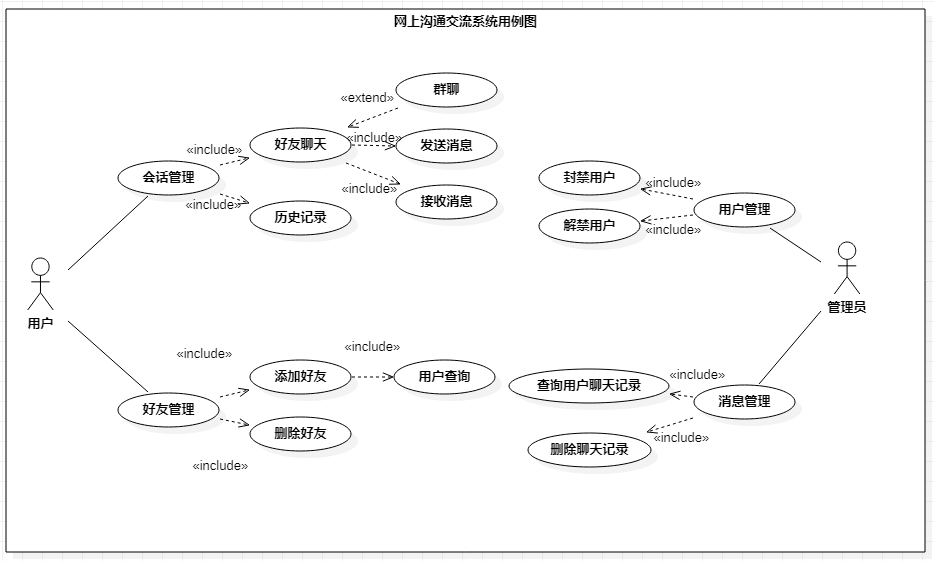
|  |  |
| --- | --- |
| 名称 | 说明 |
| 帐号 | 用户注册时登记的用户名及密码 |
| 用户分类 | 用户、管理员 |
| 登录 | 用户用以注册的账号和密码登入系统 |
| 注册 | 用户注册个人账户 |
| 聊天 | 两用户之间进行交流 |
| 群聊 | 多用户间组建群聊，在群内进行交流，聊天内容群内成员均可见 |
| 历史记录 | 显示好友以及群聊内之前的聊天内容 |
| 添加好友 | 通过查询，发起添加好友申请，该用户可以拒绝或者同意，同意后加入互相的好友列表 |
| 删除好友 | 用户可以删除已添加的好友 |
| 禁止（解禁）用户账号 | 管理员可以禁止（解禁）用户账号 |
| 聊天记录管理 | 管理员可查看任意用户的聊天信息，并可对其删除 |

## 1.4参考资料

|  |  |  |
| --- | --- | --- |
| 编号 | 参考资料 | 作者 |
| 1 | 《软件工程（第三版）》，高等教育出版社，2012 | 齐治昌等 |
| 2 | 《构建之法（第二版）》，人民邮电出版社，2015 | 邹欣 |
| 3 | 《Test-Driven Development with Python》，O'Reilly，2014 | Harry J. W |

# 2系统结构模型

## 2.1初始对象表

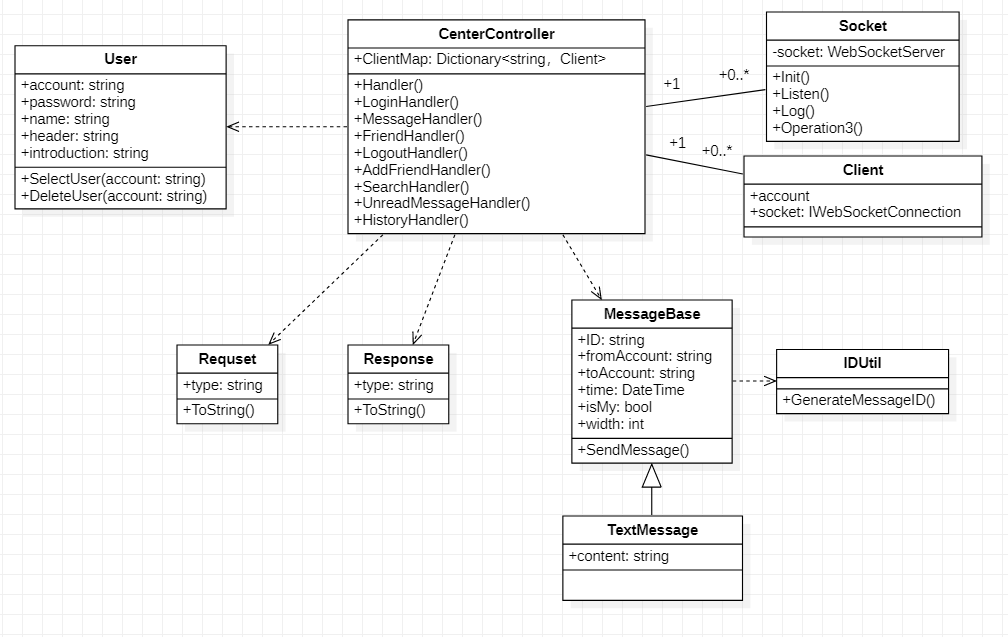


网上沟通交流系统用例图

初始对象/类表：

|  |  |  |
| --- | --- | --- |
| 可能的类 | 英文类名 | 简单描述 |
| 用户 | User | 该类是用户类，提供对用户信息的相关操作 |
| 好友 | Friend | 该类是好友类，提供对好友信息的相关操作 |
| 消息基类 | MessageBase | 消息基类，提供消息的基本信息 |
| 文本消息 | TextMessage | 最简单的文本消息，只有文字 |
| 图片消息 | ImageMessage | 图片类型消息 |
| 文件消息 | FileMessage | 文件类型消息 |
| 客户端套接字 | ClientSocket | 用于处理客户端Socket通信 |
| 服务端套接字 | ServerSocket | 用于处理服务端Socket通信 |
| 中央控制器 | CenterController | 中央处理器，用于处理各种类型的请求和响应 |
| 请求 | Request | 请求消息的基类 |
| 响应 | Response | 响应消息的基类 |

## 2.2初始类图



网上沟通交流系统用例图

## 2.3细化类图

### 2.3.1属性的识别

类名：User（用户类）

属性表如表2.1.1所示：

表2.1.1 User类属性表

|  |  |  |
| --- | --- | --- |
| 中文名 | 英文名 | 数据类型 |
| 用户ID | account | unsigned |
| 用户账号 | account | String |
| 用户密码 | password | String |
| 真实用户名 | name | String |
| 个人介绍 | introduction | String |
| 电子邮箱 | Email | String |

类名：Friend（朋友类）

属性表如表2.1.2所示：

表2.1.2 Friend类属性表

|  |  |  |
| --- | --- | --- |
| 中文名 | 英文名 | 数据类型 |
| 用户账号 | account | String |
| 用户密码 | password | String |
| 用户名 | name | String |
| 头像 | header | String |
| 个人介绍 | introduction | String |

类名：MessageBase（消息基类）

属性表如表2.1.3所示：

表2.1.3 MessageBase类属性表

|  |  |  |
| --- | --- | --- |
| 中文名 | 英文名 | 数据类型 |
| 消息ID | messageID | String |
| 发送者账号 | fromAccount | String |
| 接收者账号 | toAccount | String |
| 发送时间 | time | Date |
| 是否本人发送 | isMy | bool |
| 消息宽度 | width | int |

类名：TextMessage（文本消息类）

属性表如表2.1.4所示：

表2.1.4 TextMessage类属性表

|  |  |  |
| --- | --- | --- |
| 中文名 | 英文名 | 数据类型 |
| 文本内容 | content | String |

### 2.3.2 操作的识别

1.操作的分类

基本操作：包括数据库检索和更新，如增加、删除、修改、选择、查询等

关键操作：必须由对象提供的、在算法上复杂的业务操作

2.操作的识别

类名：User

操作表如表2.2.1所示：

表2.2.1 User类操作表

|  |  |  |
| --- | --- | --- |
| 中文名 | 英文名 | 返回值类型 |
| 登录 | login() | bool |
| 注册 | regist() | bool |
| 返回用户账号 | getAccount() | String |
| 设置用户账号 | setAccount() | bool |
| 返回用户密码 | getUserPassword() | String |
| 设置用户密码 | setUserPassword() | bool |
| 返回用户真实姓名 | getUserRealName() | String |
| 设置用户真实姓名 | setUserRealName() | bool |

类名：Friend

操作表如表2.2.2所示：

表2.2.2 Friend 类操作表

|  |  |  |
| --- | --- | --- |
| 中文名 | 英文名 | 返回值类型 |
| 查询好友 | search() | bool |
| 添加好友 | addFriend() | bool |
| 删除好友 | delFriend() | bool |
| 返回用户账号 | getAccount() | String |
| 设置用户账号 | setAccount() | bool |
| 返回用户密码 | getUserPassword() | String |
| 设置用户密码 | setUserPassword() | bool |
| 返回用户真实姓名 | getUserRealName() | String |
| 设置用户真实姓名 | setUserRealName() | bool |

类名：MessageBase

操作表如表2.2.3所示：

表2.2.3 MessageBase类操作表

|  |  |  |
| --- | --- | --- |
| 中文名 | 英文名 | 返回值类型 |
| 发送消息 | sendMessage() | bool |
| 查询历史 | getHistory() | bool |
| 删除消息 | deleteMessage() | bool |

## 2.4三层精化类图

在初始类图的基础上，运用三层体系结构思想对基本类模型进行划分。根据各个对象的具体情况，将对象分成界面、事务和数据层。

1.对客户端的三层划分

界面层：

|  |  |
| --- | --- |
| 名称 | 描述 |
| LoginWindow.xaml | 登录页面 |
| FriendWindow.xaml | 好友页面 |
| ChatWindow.xaml | 聊天页面 |
| HistoryWindow | 历史消息页面 |
| FindFriendWindow.xaml | 添加好友页面 |

事务层：

|  |  |
| --- | --- |
| 名称 | 描述 |
| CenterController.cs | 中央处理类 |
| FriendController.cs | 好友处理类 |
| MessageController.cs | 消息处理类 |
| UserController.cs | 用户处理类 |

数据层：

|  |  |
| --- | --- |
| 名称 | 描述 |
| LoginViewModel.cs | 登录视图模型 |
| FriendViewModel.cs | 好友视图模型 |
| ChatViewModel.cs | 聊天视图模型 |
| HistoryViewModel.cs | 历史记录视图模型 |
| FindFriendViewModel.cs | 添加好友视图模型 |

2.对student类的三层划分

界面层：

|  |  |
| --- | --- |
| 名称 | 描述 |
| MainWindow.xaml | Log页面 |
| ChatRecordWindow.xaml | 用户聊天记录管理页面 |

事务层：

|  |  |
| --- | --- |
| 名称 | 描述 |
| CenterController.cs | 中央处理类 |
| FriendController.cs | 好友处理类 |
| MessageController.cs | 消息处理类 |
| UserController.cs | 用户处理类 |

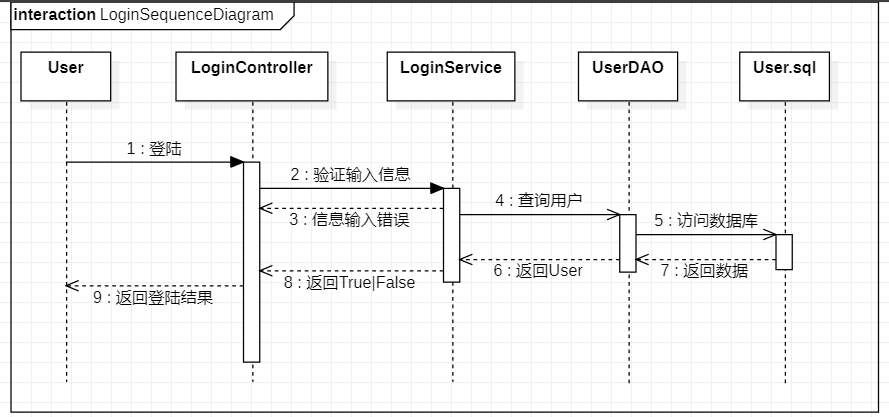
数据层：

|  |  |
| --- | --- |
| 名称 | 描述 |
| MainViewModel.cs | Log视图模型 |
| ChatRecordViewModel.cs | 用户聊天记录视图模型 |
| BaseDAO.cs | DAO基类 |
| UseDAO.cd | 用户信息数据访问类 |
| FreindDAO.cs | 好友信息数据访问类 |
| MessageDAO.cs | 消息信息数据访问类 |

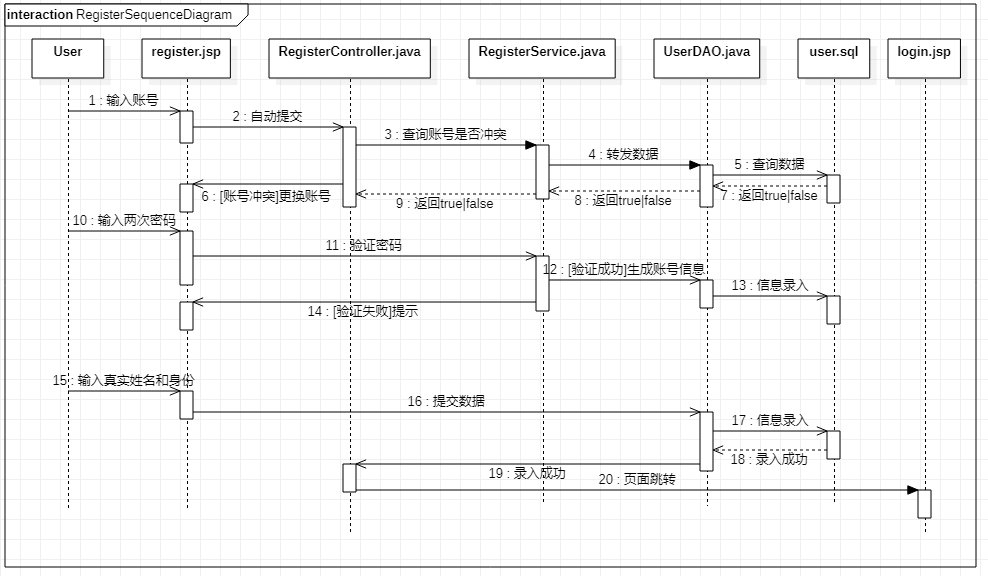
# 3系统行为模型

## 3.1交互模型

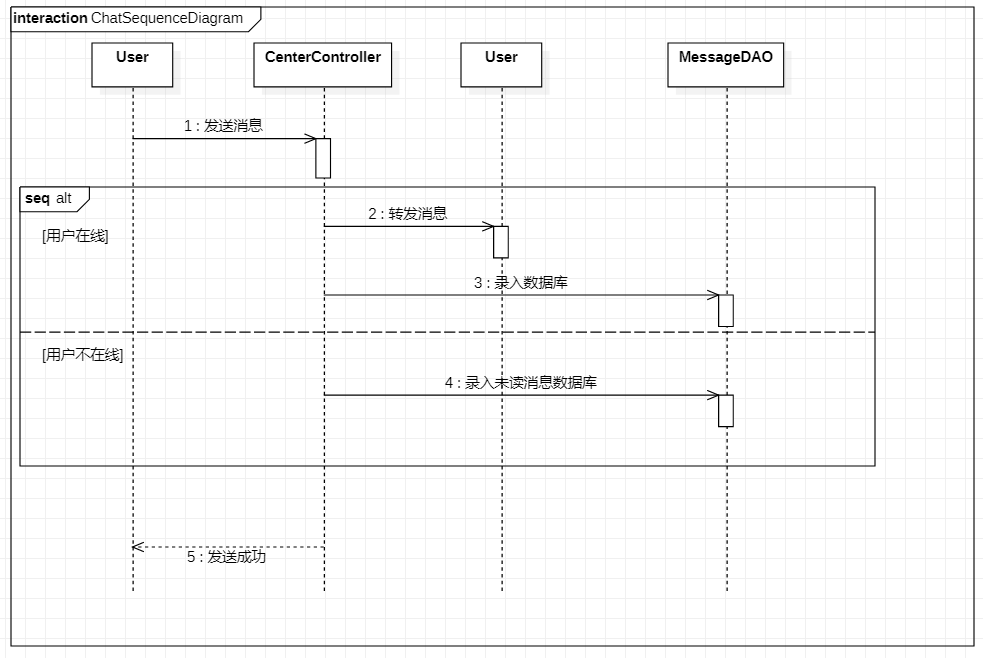
登录



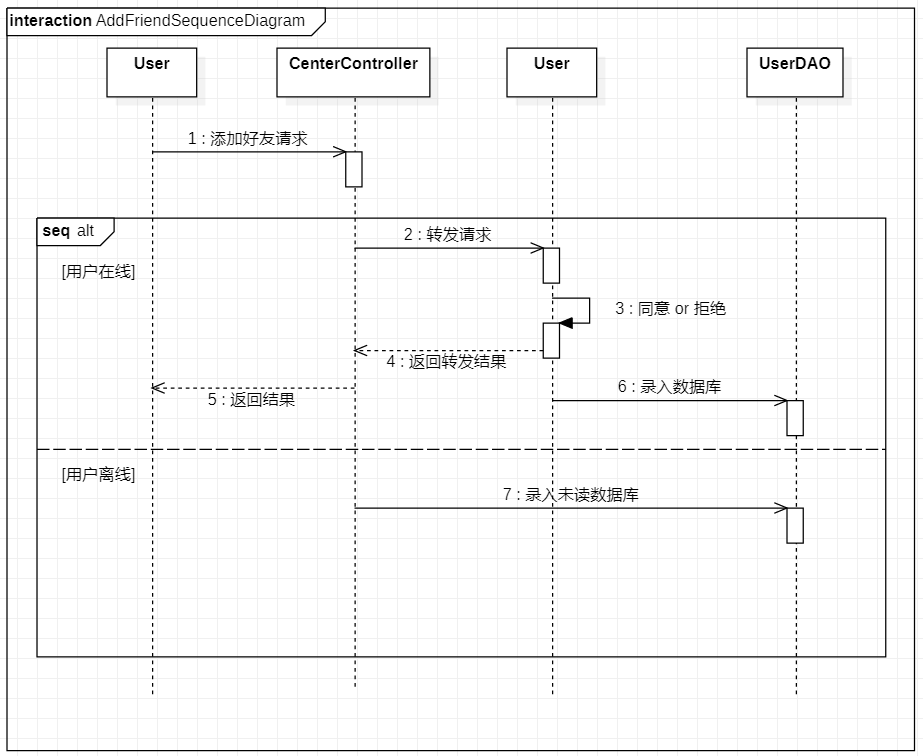
注册



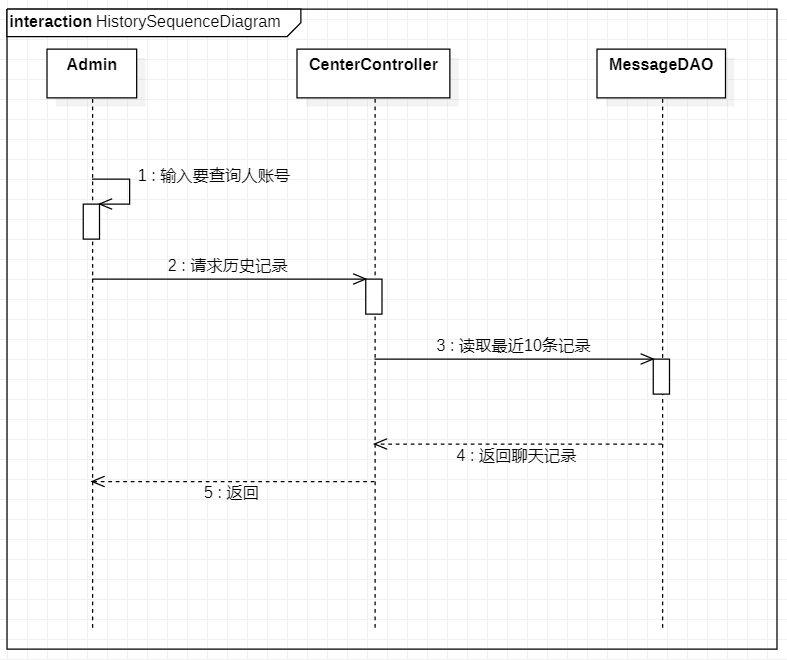
用户聊天



添加用户

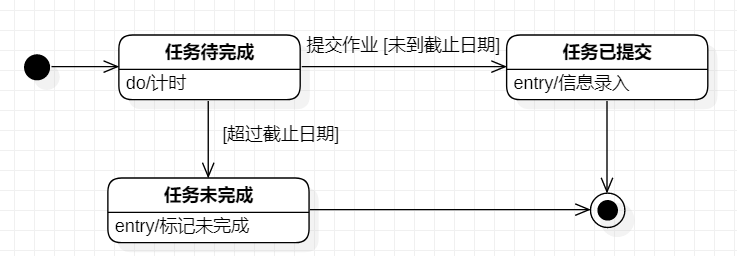


历史消息



## 3.2状态模型

提交任务



作业评定

