《Java技术》课程上机报告

综合大作业

学 号：­­­14130130246

姓 名：张勇

提交日期：2015.5.26

## 一、任务简要描述

用java实现一个聊天程序：

1、使用 Java 技术完成聊天室系统，深入学习使用 Java 语言。  
2、使用 Java 的多线程机制，深入理解 Java 多线程技术的应用。  
3、使用 GUI，对 JFrame 的深入学习。  
4、 使用网络编程， 掌握基于 TCP 协议的 Socket 编程， 了解 Socket编程的协议约定，掌握简单应用协议的开发。  
5、使用 C/S 架构，对网络编程有一定的了解。

## 二、系统需求分析

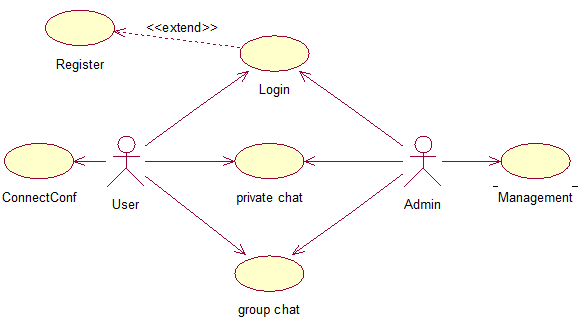
**功能性需求：**

* 用户注册
* 用户登陆
* 一个聊天室
* 多个客户同时聊天
* 不能重复登陆
* 每个人能够看到当前聊天室的情况，有多少人，都有哪些人在聊天。
* 多个聊天室
* 能够私聊

**非功能性需求：**

暂无

**用例图：**



注：各个用例的规约文档（包含各个用例的类图，顺序图和通信图（与交互图等价）），请参看本文件夹中的规约文档。

仅以登录用例、注册用例举例，其余用例不再赘述！

## 三、系统设计

从角色上分析，聊天室可以有两种类型的角色：管理员和普通用户。

普通用户在登录后，就可以进入这个聊天室了，并可以选择通过聊天界面进行单人一对一的私聊或者是和大家一起群聊。如果是第一次进入该聊天室，还可以选择本模块提供的注册功能，通过提交注册信息，成为该聊天室的新用户。

管理员除了拥有普通用户的所有功能外，还可以通过专门的管理员界面，对普通用户进行管理，可以通过“管理”界面把一个或多个用户删除，让他们下次无法再登录到聊天室。

**1、模块设计：**

服务器端：

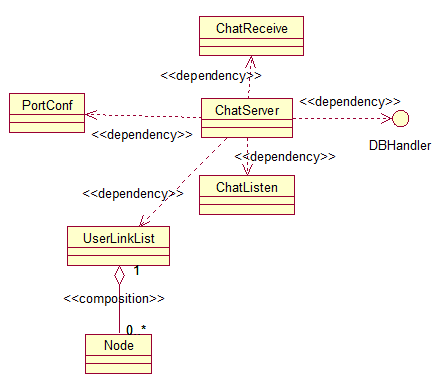
|  |  |  |
| --- | --- | --- |
| **模块名** | **类名** | **功能描述** |
| 数据库操作 | DBHandler | 封装了一些JDBC的API，向本聊天室提供数据库操作方面的接口 |
| 服务器界面 | ChatServer | 定义服务器端的界面，添加事件侦听与事件处理 |
| 监听端口 | ServerListen | 实现服务器端用户上线与下线的侦听 |
| 消息收发 | ServerReceive | 实现服务器消息收发的类 |
| 端口设置 | PortConf | 继承自JDialog，是管理员对服务器侦听端口进行修改配置的类 |
| 用户节点 | Node | 用户链表的节点类，定义了链表中的用户 |
| 用户链表 | UserLinkList | 用户链表节点的具体实现类 |

**客户端：**

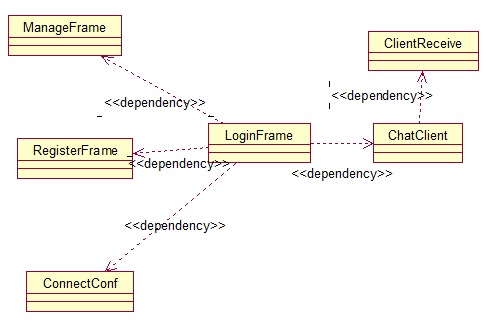
|  |  |  |
| --- | --- | --- |
| **模块名** | **类名** | **功能描述** |
| 客户端界面 | ChatClient | 定义客户端的界面，添加事件侦听与事件处理 |
| 消息收发 | CliientReceive | 实现服务器端与客户端消息收发的类 |
| 连接设置 | ConnectConf | 是用户对所要连接的服务器的IP及侦听端口进行修改配置的类 |
| 登录 | LoginFrame | 提供权限检测的功能 |
| 注册 | Registerframe | 提供普通用户注册功能 |
| 管理 | ManageFrame | 提供管理员删除用户的功能 |

**2、系统关键抽象的类图：**

服务器端：



客户端：



(注：出于简化考虑，并未写出类的数据成员和方法成员！)

**3、数据库设计：**

(1)ER图（实体-关系图）：

n

1

manage

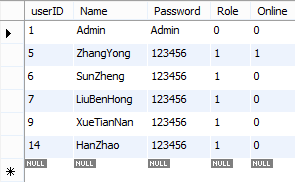
user

Admin

（2）userInfo表的结构：

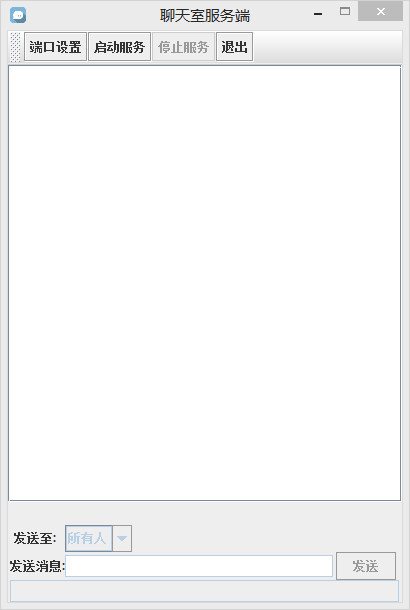
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段** | **中文描述** | **数据结构** | **是否为空** | **备注** |
| userID | 用户ID | int | 否 | 自增长，主键 |
| Name | 用户名 | Varchar(50) | 否 |  |
| Password | 密码 | Varchar(50) | 否 |  |
| Role | 权限 | int | 否 | 0 表示管理员  1 表示普通用户 |
| Online | 状态 | int | 否 | 0 表示离线  1 表示在线 |

（3）Mysql数据库中保存用户信息的表如下所示：



**4、人机界面（UI）设计：**

（1）服务器界面：



这是服务器端的主页面，管理员可以点击“端口设置”按钮，



设置本服务器（本计算机代替）所要监听的端口。

管理员可以点击“启动服务”来启动服务器服务，然后，可以通过想法的选择框选择发送系统消息的对象，可以是所有人，也可以是某一个人，然后再输入框里输入系统消息，然后点击发送按钮发送出去。

最后，点击“关闭服务”来关闭服务器服务，或者点击“退出”按钮，关闭服务器服务并退出服务器界面。

（2）客户端界面：



如果是新用户，可以点击“注册”按钮，



输出想要设置的“用户名”和“密码”，全部操作无误后，即可成功注册；

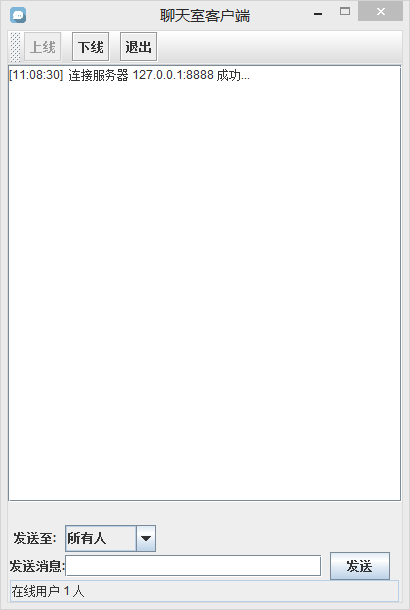
如果是管理员，可以输入“用户名”和“密码”，然后点击“管理”，进入管理界面，删除想要删除的一个或多个用户：



如果是普通用户，可以点击“连接设置”设置想要连接的服务器的IP地址和端口号：



然后点击“登录”按钮，即可进入聊天室:



用户可以选择设置自己的状态“上线”或者“下线”，只有上线后，才能与服务器建立连接，与他人聊天。

## 四、系统实现

1、TCP网络通信

服务器端：



首先，默认的服务器端侦听的端口号是8888；

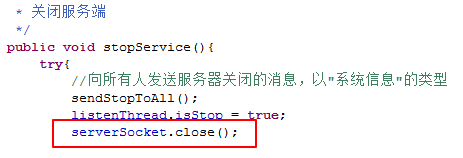
后面，管理员可以根据需要更改成自己想要的端口号，并且，客户端的用户在登录之前，也可以修改想要连接的服务器的IP地址和端口号。



当管理员点击“启动服务”按钮后，即运行startService()方法:

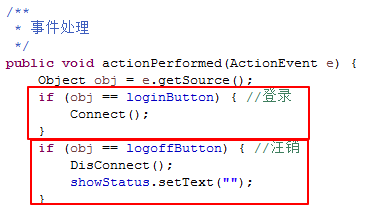


同理，关闭服务器服务时，



客户端：

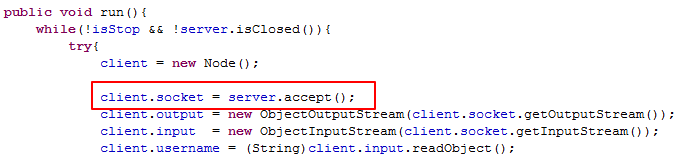




和服务器同理，点击登录后，



服务器端的侦听进程，



此时，服务器和客户端TCP网络连接建立成功！

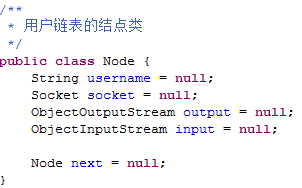
点击下线后，



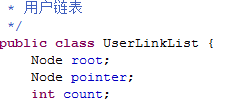
2、用户链表的实现

当前在线的用户我是通过在服务器端建立一个链表实现的。

首先，用户节点类的结构如下：



然后，链表类的结构如下：



count是用来计数当前在线人数的。

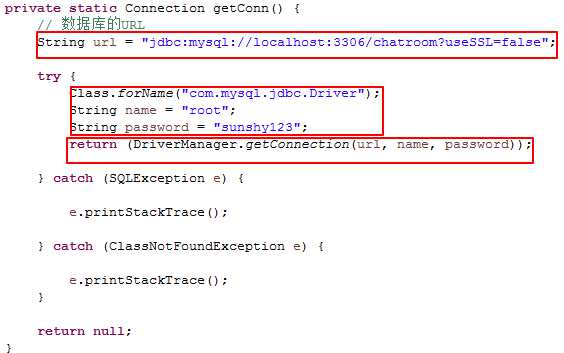
另外，链表类中，定义了诸多操作链表的方法，以完成用户上线下线的业务逻辑功能。

3、JDBC数据库连接

用户的用户名，密码，权限等信息是保存在数据库中的，这样，用户一经注册，便能长期的使用这个账号登录聊天室系统。

我将所有的关于数据库操作的方法都写进了DBHandler类中，这样DBHandler便为服务器端提供了数据库操作的接口。

连接到Mysql数据库的方法如下：



另外，DBhandler类中，还定义了查询用户权限，用户注册，用户删除等等方法，以便于实现用户登录，用户注册等业务逻辑功能。

4、通信协议的制定

当服务器端与客户端建立了TCP连接后，二者之间的信息传递通过类ServerReceive和ClientReceive实现。

二者之间消息类型的识别是通过在消息的头部添加不同的“消息类型”标签实现的。

服务器端：（以下都是代码片段）

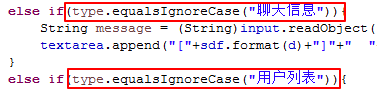




客户端：





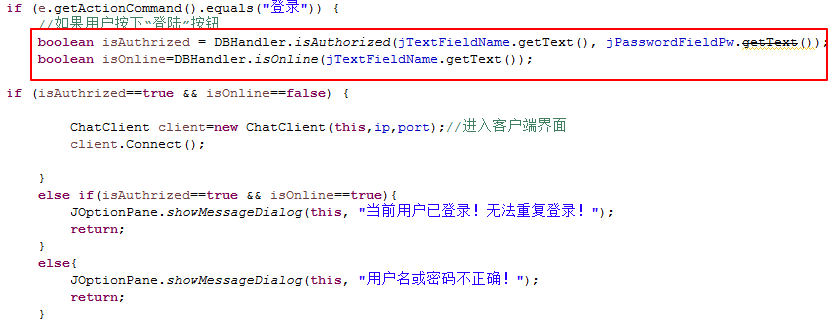


无论是服务器端还是客户端，当收到一条信息时，首先检查这条信息的首部，确定信息的类型，然后调用不同方法，已实现不同的功能。

## 五、问题及解决

1、不能重复登录问题

为了解决这个问题，我在数据库中的userInfo表中，添加了一个字段Online，当它为0的时候，表示该用户不在线，则允许这个账号登录聊天室；当它为1的时候，表示，该用户在线，则不再允许相同的账号登录聊天室。



当用户点击“下线”，或者服务器端主动关闭服务时，都会将该用户或者所有用户的Online字段更新为0。

2、能够群聊或私聊

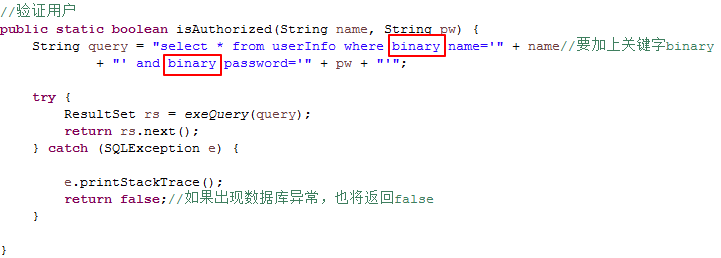
在服务器端和客户端都增加一个组合框ComboBox，来选择发送消息给所有人还是某一个人，以此来实现群聊和私聊的功能。

3、多个聊天室

启动多个服务器端程序，并且修改端口，使它们各不相同，相当于实现了多个聊天室的功能。用户可以通过修改“连接设置”，以连接到不同的服务器，进入不同的聊天室。

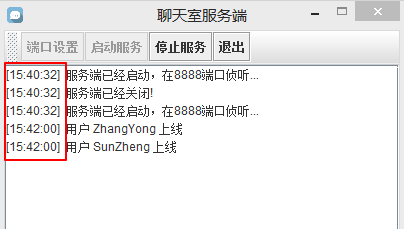
但是，一个用户账号能且只能登录一个服务器，这也就解决了不能重复登录的问题，不同的服务端是完全相互独立的聊天室。

4、Mysql数据库查询操作默认不区分大小写

没修改查询语句之前，登录的时候，输入的用户名，密码是不区分大小写的，都可以登录成功。然后在每一条数据库查询语句的where字句的每一个字段前面加上binary关键字，这样，查询语句就变成区分大小写的了。

## 六、系统特色与创新

1、在聊天框里，每个消息的首部都添加了时间戳；



2、每一个窗体都增加了图标icon；



3、服务器端可以动态修改侦听端口，客户端可以动态修改想要连接的服务器的IP地址和端口号；



服务器端口设置



客户端连接设置

七、个人心得与体会

采用 C/S 架构设计的聊天室，使用增量迭代方式开发，通过本次实验，我对 C/S 架构有了较深的认识，也对多线程编程、网络编程、GUI 的认识更清晰，能够将所有学过的 Java 知识综合运用，完成应用开发任务。

存在的问题有：

1、业务逻辑考虑不周:

socket连接

client

server

访问

访问

Database

按照道理说，客户端只能跟服务器端进行通信，不能直接查询或者操作数据库，但是，我的程序中，尤其是登录界面LoginFrame类中，当时考虑不周，导入了DBHandler类，直接对数据库中的表userInfo进行了查询和增加，删除等操作，从而导致，后期发觉问题后，更改起来，工作量很大，逻辑很混乱。

从这个问题里，我体会到了，需求分析，系统设计的重要性，它是先行于具体实现和编码的，如果没有一个正确的，完整的设计，就鲁莽的开始编码，最后会发现许多问题，导致多次的大量的从头修改。所以，确认好方案之后，再行动，可以减少返工的代价。

2、编好的服务器端和客户端程序，可以在eclipse中运行，但是一旦打包成jar文件之后，就无法运行了，点击按钮，也没有任何反应。目前来说，对于编写软件程序，我力所能及的还太少，还没办法实现一个可以真正发布的程序。