**Conversation Design:**

**Definition:**

In this system, a conversation is defined as a “room” in which users may communicate with other users by sending messages in the conversation.

**Many-to-many relationship:**

A conversation may consist of any number of users (including none at all), and a user may be involved in any number of conversations (including none at all).

**Creating a conversation:**

Any user can create a conversation with a unique conversation ID at any time. Once a conversation is made, it cannot be deleted.

**Entering a conversation:**

Any user who is not in an existing conversation may enter it at any time without an invite. Once a user enters a conversation, the user will receive the enter history of the conversation.

**Inviting other users to a conversation:**

Any user who is in an existing conversation may invite any other users who are not in the conversation to enter the conversation. The invite is used only to advertise the existing conversation; it is \*not\* necessary to have an invite for a user to enter an existing conversation.

Any user who receives the invite may choose to accept the invite and enter the conversation, ignore the invite, or reject the invite.

**Exiting a conversation:**

Any user who is in an existing conversation may exit it at any time.

**Sending messages:**

Any user who is in an existing conversation may communicate with the other users currently in the conversation by sending messages in the conversation. The message, which can consist of any string, is first sent from the user (client) to the server, which will process the string and add it to the conversation. Once the message is added to the conversation, all users who are currently in the conversation will be able to simultaneously view the message.

**Conversation Class:**

**Fields:**

String conversationID

Set<User> users

ArrayList<Dialogue> history

**Constructor:**

Conversation(String conversationID, User user, String timeStamp): initialize the conversation with the unique ID and the user who created the conversation.

**Public methods:**

String getConversationID()

List<User> getListUsers()

Void addMessage(String userID, String message, String timeStamp)

Void sendMessageToAllUsers(Dialogue dialogue)

Void addUser(User user, String timeStamp)

Void sendHistoryToUser(PrintWriter out)

Void removeUser(User user, String timeStamp)

Boolean contains(User user)

Boolean contains(String userID)

**Dialogue Class:**

**Fields:**

String userID

String text

String timeStamp

**Constructor**

Dialogue(String userID, String text, String timeStamp)

**User Class:**

**Fields:**

String userID

Socket socket

PrintWriter out

Set<Conversation> conversations

**Public methods:**

String getUserID()

Socket getSocket()

PrintWriter getPrintWriter()

Void addConversation(Conversation conversation)

Void removeConversation(Conversation conversation)

List<Conversation> getConversations()