Sales IM Messenger 1.0 Requirements Specification

1. Scope

1.1 Overview

The Sales IM Messenger component is part of a larger client-server chat application. It distributes and routes all kinds of messages that are involved. Since connection is not persistent between client and server, messages are pushed to message pools for clients to pull. Additional formatting will be applied on normal chat messages.

1.2 Logic Requirements

1.2.1 Messages

Six types of messages should be derived on top of the abstract `Message` class. Each message type has its own XML representation. The XML must show the message timestamp. The designer is responsible for defining the XSD for each message type. It should be defined in a way such that they can be easily distinguishable from each other.

The attributes included in each message type are described below. Getters and setters should be provided for the attributes. They must be shown in the XML as well.

1.2.1.1 Chat Message

This is the message typed by a user in a chat room. It should include:

- User profile of the person who typed the text (only username is required in XML)
- Chat session id
- Chat text

1.2.1.2 Session Unavailable Message

This is the message to notify a user that the session is unavailable. The sender of this message must be the system. It should include:

- Chat session id

1.2.1.3 Presence Message

This is the message to indicate if a user is present or absent. It should include:

- User profile of the person who changed the presence (only username is required in XML)
- Chat session id
- Absence / Presence

1.2.1.4 Ask For Chat Message

This is the message to ask a user for chatting in a session created by another user. The sender of this message must be the system. It should include:

- User profile of the person who created the chat
- Chat session id
- Timestamp of when the session is created
- Timestamp of which the user needs to acknowledge

1.2.1.5 Enter Chat Message

This is the message to signal a user to enter a chat session, after he has accepted for chatting. The sender of this message must be the system. It should include:

- Chat session id

1.2.1.6 Formatted Chat Message

This is a formatted version of the chat message (see 1.2.1.1). Below is the standard format:

[Name]: [Timestamp]: [Text Message]

Field Name	Description
Name	Simply the username
Timestamp	The message timestamp. The format for the time should be configurable with the default being the following format: HH:mm:ss PM Time zone For example, 05:34:30 PM EST
Text Message	The chat text

The font and color for each of the above parts in configurable. In particular, the color for text message is different for client and manager. This allows easier differentiation between the two. For now, assume the "Role" property returns either "Client" or "Manager".

Furthermore, if a text message includes a link to a URL, then the URL text will be displayed as a hyperlink. Below are the rules for determining if text should be generated as a hyperlink:

- If the text begins with a http://, https://, ftp:// (not case sensitive)
- If the text follows the pattern: www.[Text].[Top Level Domain], where [text] section may include dots, and [Top Level Domain] is defined in http://data.iana.org/TLD/tlds-alpha-by-domain.txt

The **Content Highlighter** component can be used to highlight the hyperlinks.

1.2.2 Messenger

The messenger is responsible for routing the messages to the message pools of the desired recipients. Using the **Chat Message Pool** component, users will be able to receive the messages by pulling them off.

Only these messages can route directly to the user message pool:

- Session Unavailable Message
- Ask For Chat Message
- Enter Chat Message

Only these messages can route directly to the user's session message pool:

- Chat Message
- Presence Message

1.2.2.1 Formatting

Chat message will always be formatted (see 1.2.1.6) before delivery.

1.2.2.2 Delivery of Session Messages

Chat message will be delivered to all users of the session, including the sender. Presence message will be delivered to all users of the session, except the sender.

1.2.2.3 Blocked Users

The **Chat Contact Manager** component can be used to determine if a sender is blocked from sending message to a user. If it is blocked, simply do not push the message into message pool.

1.2.2.4 Persistence

The component should insert a row into the `session_user_message` table once a chat message is posted in a session. The auditing fields will be inserted by database trigger.

1.2.3 Thread Safety

The component must be thread-safe. Messages can be posted from different sources at the same time.

1.3 Required Algorithms

None

1.4 Example of the Software Usage

The Sales IM application will use this component as the core module to route different kinds of messages that are involved. Messages are routed from system to client.

1.5 Future Component Direction

More message types will be supported in future release, such as file transfer message. The font and color used in formatting may be specific to each user's setting.

2. Interface Requirements

2.1.1 Graphical User Interface Requirements

None

2.1.2 External Interfaces

```
public interface Messenger {

    void postMessage(Message message, long user);

    void postMessage(Message message, long user, long session);

    void postMessageToOthers(Message message, ChatSession session);

    void postMessageToAll(Message message, ChatSession session);

    MessagePool getMessagePool();

    void setMessagePool(MessagePool pool);
}
```

2.1.3 Environment Requirements

- Development language: Java 1.4
- Compile target: Java 1.4

2.1.4 Package Structure

- com.cronos.saleim.messenger

3. Software Requirements

3.1 Administration Requirements

3.1.1 What elements of the application need to be configurable?

- Formatting attributes

3.2 Technical Constraints

3.2.1 Are there particular frameworks or standards that are required?

None

3.2.2 TopCoder Software Component Dependencies:

- DB Connection Factory 1.0
- Content Highlighter 1.1
- Chat User Profile 2.0
- Chat Contact Manager 1.0
- Chat Session Manager 1.0
- Chat Message Pool 1.0

**Please review the TopCoder Software component catalog for existing components that can be used in the design.

3.2.3 Third Party Component, Library, or Product Dependencies:

- Informix Database 10

3.2.4 QA Environment:

- Solaris 7
- RedHat Linux 7.1
- Windows 2000
- Windows 2003

3.3 Design Constraints

The component design and development solutions must adhere to the guidelines as outlined in the TopCoder Software Component Guidelines. Modifications to these guidelines for this component should be detailed below.

3.4 Required Documentation

3.4.1 Design Documentation

- Use-Case Diagram
- Class Diagram
- Sequence Diagram
- Component Specification

3.4.2 Help / User Documentation

- Design documents must clearly define intended component usage in the 'Documentation' tab of Poseidon.