IM Ajax Support 1.0 Requirements Specification

1. Scope

1.1 Overview

IM is an application which allows users to perform online chat through the web browser. AJAX technology is used heavily to exchange messages in order to enhance user's experience. The IM Ajax Support component provides the servlets to handle the Ajax requests and responses on the server side.

1.2 Logic Requirements

1.2.1 Background

The component will provide servlet support for the application on AJAX interactions. The input and output will be using XML. The designer is responsible for designing the schema.

The interactions will be covered in different pages of the application. On the request side, Javascript will be used to compose the XML input and pass it to the servlet for processing. On the respond side, the output XML encapsulating the updated information/result will be parsed asynchronously using Javascript.

1.2.1.1 Session

The user session must store the chat user profile under a configurable attribute name. User id can be retrieved directly. User role can be read from the role property of the profile, and can be one of "Client" and "Manager". Category can be read from the category property of the profile, and can have multiple values.

1.2.1.2 XML

The `toXMLString` method of the `XMLMessage` already returns its XML representation. Multiple messages will be concatenated into one XML in the response. Other than that, the design should define the schema for the remaining things to be returned, including the acknowledgement of whether the operation succeeds.

1.2.1.3 Formatting

A formatting context will be supplied to the `toXMLString` method, in order to display the dates in local time zone. The time zone id can be retrieved from a cookie under a configurable name. In addition, all Chat Text Messages will be written in the formatted version.

1.2.2 Read Client User Message

This is only valid for users with client role. The servlet will pull the pending messages from the user message pool. In addition, the current position of the client (under the selected category) in the service engine is included in the response.

1.2.3 Read Manager User Message

This is only valid for users with manager role. The servlet will pull the pending messages from the user message pool.

1.2.4 Read Client Session Message

This is only valid for users with client role. The request parameter includes the session id. The servlet will pull the pending messages from both the user message pool and session message pool.

1.2.5 Read Manager Session Message

This is only valid for users with manager role. The request parameter includes the session id. The servlet will pull the pending messages from the session message pool of the user.

1.2.6 Accept Chat Request

The request parameter includes the request user id and session id. The servlet will update the state in the service engine to servicing for that session. The response should indicate whether the operation succeeds or not.

1.2.7 Post Text Message

The request parameter includes the session id and text. The servlet will post the submitted message to all users of the session. In addition, the servlet will pull the pending messages from the session message pool of the user and return them in the response.

1.2.8 Change Manager Status

This is only valid for users with manager role. The request parameter includes the status. The servlet will change the user status. If it is ONLINE, the servlet will enqueue the manager to the service engine for each of the selected categories. Conversely, if it is BUSY, the servlet will dequeue the manager from the service engine.

1.2.9 Logging

Logging should be performed for each operation using the Logging Wrapper. Logging must comprise user ID, timestamp, action taken, and entity IDs effected.

1.3 Required Algorithms

None

1.4 Example of the Software Usage

The IM application will use this component to setup servlets to handle AJAX interactions. On the client side, the request will be submitted to this component as XML, and the UI will be updated according to the response received. Both will follow the XML schema defined in this component.

1.5 Future Component Direction

None

2. Interface Requirements

2.1.1 Graphical User Interface Requirements

None

2.1.2 External Interfaces

None

2.1.3 Environment Requirements

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

2.1.4 Package Structure

- com.cronos.im.ajax

3. Software Requirements

3.1 Administration Requirements

3.1.1 What elements of the application need to be configurable?

- Session attribute name
- Not required but recommended to make the hard-coded strings configurable

3.2 Technical Constraints

3.2.1 Are there particular frameworks or standards that are required?

- Servlet 2.4

3.2.2 TopCoder Software Component Dependencies:

- Chat User Profile 2.0
- Chat Session Manager 1.0
- Chat Status Tracker 1.0
- Chat Message Pool 1.1
- IM Messenger 1.0
- Service Engine 1.0
- Logging Wrapper 2.0

**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:

None

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.