Software Documentation : Java Custom Cockpit Studio Service 1.3

This page last changed on Jul 19, 2009 by mashannon168.

1. Scope

1.1 Overview

This component contains a service that handles studio competitions and some other studio related classes. In version 1.3 of this component the studio service will be updated (new methods will be added), new classes will be added and others will be updated.

1.1.1 Version

1.3

1.2 Logic Requirements

1.2.1 Update StudioService and its current implementation (StudioServiceBean)

In this new version of the component the StudioService interface and its ejb implementation, StudioServiceBean, will be updated with new methods. This competition will handle the classes presented in "Studio Service 1.3 Class Diagram" from uml.
Note that in Pipeline Conversion Architecture the following modifications were made to the entities:

- Resource entity now has a name field (to identify a user)

- StudioCompetition now has an array of Resource type (this array identifies the users involved in this project (client, manager. copilot...))

- getUserContests(username) method will get the contests where the user identified by username is a resource for the contests (new tables defined in Pipeline Conversion Architecture can be used to identify user contests)

- getMilestoneSubmissionsForContest will get the milestone submission for the given contest (milestone submissions will be identified by SubmissionData.submissionType = "milestone_submission")

- getFinalSubmissionsForContest will get the submissions for which submission type is "final_submission_type" - setSubmissionMilestonePrize - sets the mmilestone prize for the submission

These methods will delegate to namesake methods from ContestManager and SubmissionManager.

1.2.2 Update diagrams

The class diagrams of this component are not up-to-date and this will be resolved in this competition. The competing designers will check the current code and update the class diagrams of this component. If necessary, the use case diagram and sequence diagrams also need to be updated.

1.3 Required Algorithms

None.

1.4 Example of the Software Usage

The ejb service is used to manage the studio competitions.

1.5 Future Component Direction

None.

Copyright TopCoder, Inc. 2008

2. Interface Requirements

2.1.1 Graphical User Interface Requirements

None.

2.1.2 External Interfaces

Design will follow "Studio Service 1.3 Class Diagram" from uml.

2.1.3 Environment Requirements

- Development language: Java1.5
- Compile target: Java1.5

2.1.4 Package Structure

com.topcoder.service.studio

com.topcoder.service.studio.ejb

3. Software Requirements

3.1 Administration Requirements

3.1.1 What elements of the application need to be configurable?

None.

3.2 Technical Constraints

3.2.1 Are there particular frameworks or standards that are required?

Java 1.5+
EJB 3.0
WSDL 1.1

3.2.2 TopCoder Software Component Dependencies:

- Base Exception 2.0
- Logging Wrapper 2.0
- Contest Manager 1.3
- Submission Manager 1.2
- JBoss Login Module 2.0

3.2.3 Third Party Component, Library, or Product Dependencies:

None

3.2.4 QA Environment:

- RedHat Linux 9
- Informix 10
- JBoss 4.2

Copyright TopCoder, Inc. 2008

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 TC UML Tool.

Copyright TopCoder, Inc. 2008