Software Documentation : Java Custom Cockpit Contest Service Facade 1.1

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

1. Scope

1.1 Overview

This component has a service façade that manages both studio and software competitions. In version 1.1 of this component the service facade will be updated (new methods, that handle studio competitions, will be added).

1.1.1 Version

1.1

1.2 Logic Requirements

1.2.1 Update ContestServiceFacade and its current implementation (ContestServiceFacadeBean)

In this new version of the component the ContestServiceFacade interface and its ejb implementation, ContestServiceFacadeBean, will be updated with new methods. This competition will handle the classes presented in "Contest Service Facade 1.1 Class Diagram" from uml.
The new methods will simply delegate to the namesake methods from StudioService (from Studio Service component).

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 contest service façade is used to manage both studio and software competitions.

1.5 Future Component Direction

None.

2. Interface Requirements

2.1.1 Graphical User Interface Requirements

None.

2.1.2 External Interfaces

Design will follow "Contest Service Facade 1.1 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.facade.contest

com.topcoder.service.facade.contest.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
- Project Service 1.0
- JBoss Login Module 2.0
- Studio Service 1.3

3.2.3 Third Party Component, Library, or Product Dependencies:

None.

3.2.4 QA Environment:

- RedHat Linux 9
- Informix 10
- JBoss 4.2

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.