Java Custom Studio Contest Manager

1. Scope

1.1 Overview

This component provides operations on contests like *add new contest*, *get contest*, *update contest*, *update contest status*; CRUD (Create, Read, Update, Delete) operations on contest status; CRUD operations on competition documents; get client by contest and project; CRUD operations on the contest category; and CRUD operations for the configuration parameters. It also provide an ability to save files to the server's file system and retrieve them from server's file system. Component runs as a stateless EJB. It uses Hibernate JPA implementation to work with persistence. It is used by Studio Service and can be used for the other purposes.

1.2 Version

1.0

1.2 Logic Requirements

1.2.1 Provide operations on the contest

- Create new contest.
- Get contest by id
- Get contests for project.
- Update contest.
- Update contest status. Statuses must be updated in strict order. See the Chart Diagram is additions.
- Add configuration parameter to contest

See interface diagram for method's API

1.2.2 Get client identifier

- Get client identifier by project. Contest should select client from project table by project id.

- Get client identifier by contest. Get contest attribute tcDirectProjectId and use it to select client from project table

See interface diagram for method's API

1.2.3 Provide CRUD operations on the contest status

- Add contest status
- Update contest status
- Remove contest status
- Get contest status by id

See interface diagram for method's API

1.2.4 Provide CRUD operations on the document

- Add document.
- Update document
- Remove document
- Get document by id

See interface diagram for method's API

1.2.5 Provide CRUD operations on the contest category

- Add contest category
- Update contest category
- Remove contest category
- Get contest category by id

See interface diagram for method's API

1.2.6 Provide CRUD operations on the contest type

- Add configuration parameter to contest type

See interface diagram for method's API

1.2.7 Provide CRUD operations on the configuration

- Add configuration parameter
- Update configuration parameter
- Get configuration parameter by id

See interface diagram for method's API

1.2.8 Saving file

Component provides the ability to save documents in server's file system. The saving file process should be configurable. It should be able to save files in a configurable system folder, and provide a way to set name and other needed parameters of the file by using a Document instance. Also component provide ability to retrieve documents from server's file system.

1.2.9 Database access

Hibernate is used to provide access to database in EJB. Note that Hibernate 3.2 is completely compatible with JPA so it should be used as a standard JPA provider.

1.2.10 EJB description

Stateless bean should have both remote and local interface. All its methods should have security role access - "Administrator". All methods should use REQUIRED transaction attribute.

1.2.11 Logging

All defined operations should be logged. The logging mechanism should be pluggable, e.g., it should be an option to disable logging. Logging strategy:

- Entrance and exit of methods should be logged at the INFO level
- Exception should be logged at the ERROR level

1.3 Required Algorithms

None.

1.4 Example of the Software Usage

This component will be used by Studio Service and possibly by some other services, which provide operations on contest

1.5 Future Component Direction

Add more methods to interface.

2. Interface Requirements

2.1.1 Graphical User Interface Requirements

None.

2.1.2 External Interfaces

See interface diagram:

2.1.3 Environment Requirements

- Development language: Java 1.5
- Compile target: Java 1.5, Java 1.6

2.1.4 Package Structure

com.topcoder.service.studio.contest

3. Software Requirements

3.1 Administration Requirements

3.1.1 What elements of the application need to be configurable?

Logging should be pluggable

Configuration for saving/retrieving documents.

3.2 Technical Constraints

3.2.1 Are there particular frameworks or standards that are required?

- EJB 3.0
- JPA 1.0
- Hibernate 3.2 and higher

3.2.2 TopCoder Software Component Dependencies:

- Base Exception 2.0
- Logging Wrapper 2.0
- Contest And Submission Entities 1.0 (Development Component Only).

**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:

- 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 the TC UML Tool.