Java Custom Submission Manager

1. Scope

1.1 Overview

This component provides operations on contest like add CRUD operations on the submission, prize, review and submission payment; update submission status and placement; add and remove prizes from submission. Component runs as stateless EJB. This component is used Hibernate JPA implementation to work with persistence. It is used by Studio Service and can be used for the other services.

1.2 Version

1.0

1.2 Logic Requirements

1.2.1 Provide operations on the submission

- Get submission by id

- Get submissions for contest. Returned submission data can contain link either to full submission or to watermarked images from submission. This depends on selectFullSubmission attribute. If selectFullSubmission is true it should be returned full link of submission, in other case to return link from submission.

- Get all submissions by member

- Update submission

- Remove submission. Status of submission should be "deleted". The submission is not actually deleted from the system. Note that removed submissions are invisible for select operations.

- Update submission result e.g. update submission rank and add prize for submission depends on rank. Note that under contest only 1 submission can have the first place, only 1 can have the second place

etc. Ties are not allowed. This operation is allowed by client who owns the contest project or for administrator.

See interface diagram for methods API

1.2.2 Provide operations on the prizes

- Add prize
- Update prize
- Remove prize
- Get prize by id
- Add prize to submission
- Remove prize from submission

See interface diagram for methods API

1.2.3 Provide CRUD operations on the submission payments.

- Add submission review
- Update submission review
- Remove submission review
- Remove submission reviews for submission
- Get submission review by id
- Get submission reviews for submission

See interface diagram for methods API

1.2.4 Database access

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

1.2.5 EJB description

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

1.2.6 Logging

All defined operations should be logged. The logging mechanism should be pluggable e.g. it should be 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 some other services, which provides operations on studio submission documents

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.submission

3. Software Requirements

3.1 Administration Requirements

3.1.1 What elements of the application need to be configurable?

Logging should be pluggable

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 TopCoder UML Tool.