This page last changed on Feb 03, 2008 by tushak.

Cannot resolve external resource into attachment.

1. Scope

1.1 Overview

This component provides CRUD operations on document; CRUD operations on member answer; CRUD operations on competition document (document which is specific for concrete competition and role). Component runs as stateless EJB. This component is used Hibernate JPA implementation to work with persistence. It is used by TopCoder Prerequisite Service and can be used for the other services.

1.2 Logic Requirements

1.2.1   Provide CRUD operations on the document

Note that Document entity represents document with all its versions. Each document can have one or more versions, which are totally independent from each other.

The next operations should be provided:

-        Add new document in database, note that added document should have at least on version.

-        Delete document and all its versions.

-        Retrieve all documents by id.

-        Get all documents.

-        Add new component version

-        Update document version

-        Delete document version

-        Retrieve concrete document version, if version is not specified return the latest version.

-        Retrieve concrete version by id

See interface diagram for methods API

1.2.2   Provide CRUD operations on the member answer

Member can accept or reject some documents version. Member can accept or reject concrete version of document only once. Note that document can have several versions and in this case member for example can accept one of them and reject some other.

Provide the next operations:

-        Add new member answer in database.

-        Update member answer in database.

-        Delete member answer

-        Retrieve member answer

See interface diagram for methods API

1.2.3   Provide CRUD operations on the competition document.

Each competition has a list of document which needs for each specific role. Document versions relate competitions with concrete document version for the specific role.

Provide the next operations:

- Add new version of document to competition for specific role

- Update version of document associated with competition and specific role

- Delete version of document from competition

- Delete all documents from competition

- Delete all documents from competition for specific role

- Get all versions of documents for competition and role

- Get all versions of documents for specific competition

- Get all versions of documents for specific role

- Find and set unique id of competition document

See interface diagram for methods API

1.2.4   Database access

Hibernate is used to provide access to database in EJB. Designer is responsible to create full Hibernate mapping for given database schema. Id of entities should be generated by using Hibernate id generation mechanism. Note that Hibernate 3.2 is completely compatible with JPA so it should be used as a standard JPA provider.

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 TopCoder Prerequisite Service and possibly some other services, which provides operations on prerequisite documents
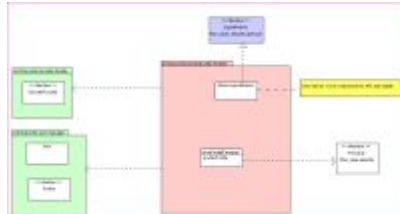
1.5 Future Component Direction

Add more methods to interface.

2. Interface Requirements

2.1.1 Graphical User Interface Requirements

None

2.1.2 External Interfaces

See Interface Diagrams

Database shema

2.1.3 Environment Requirements

- Development language: Java1.5
- Compile target: Java1.5, Java 1.6

2.1.4 Package Structure

com.topcoder.service.prerequisite.document

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

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.