



Software Documentation : Java Review Management 1.2

This page last changed on Sep 04, 2011 by volodymyrk.

1. Scope

1.1 Overview

Reviews are produced based on scorecards. A review holds a collection of items which address each of the questions on the scorecard. It also consists of the author that produced the review, the submission it addresses and the scorecard template it is based on. Various types of comments can be attached to the review or to each review item. A committed review must address all questions on the corresponding scorecard, and will have its overall score available.

The component provides the management functionality to create, update or search reviews. The review persistence logic is pluggable.

The version 1.2 adds the functionality to remove reviews. The new version should be backwards compatible. The previous version of the Review Management and Review Management Persistence components are provided for the reference.

The designer of the version upgrade should also update the design with all the changes done to the code of the previous version and update the documentation and diagrams to the current standards. The previous version did not separate the CS and diagrams for Review Management and Review Management Persistence components. Starting with this version the designer is responsible to provide two separate sets of documentation: one for the Review Management and the other one for the Review Management Persistence component. The designer will be responsible to provide forum support for both component developments.

1.2 Logic Requirements

1.2.1 Review Management Operations

Despite manipulation of the review structure defined above, this component will support the following operations.

1.2.1.1 Create Review

A new review can be created. The identifiers will be provided with ID Generator. Review should be validated.
The operator needs to be provided for auditing purpose.

1.2.1.2 Update Review

An existing review can be updated. The identifiers of any new entities will be provided with ID Generator. Review should be validated.
The operator needs to be provided for auditing purpose.

1.2.1.3 Add Review Comment

A review comment can be added either to the review level or item level.
The operator needs to be provided for auditing purpose.

1.2.1.4 Get Review

A review can be retrieved by specified identifier.

1.2.1.5 Search Reviews

An array of reviews can be retrieved by specified search criteria. An option should be able to specify whether to retrieve the complete review hierarchies or only the review instances.



1.2.1.5.1 Review Search Criteria

At minimum, the criteria should be capable of specifying any combination of scorecard type, submission, reviewer, project and committed flag.

1.2.1.6 All Comment Types

There should be a way to retrieve all comment types in the system.

1.2.1.7 Remove Review

An existing review can be deleted. This will delete all its review items and the associated comments. Also, reviews can have uploads associated (see the `tcs_catalog.upload` table). The uploads must be also deleted before deleting the review items and comments.

The operator needs to be provided for auditing purpose.

1.2.2 Search Builder Usage

Search Builder should be used with the searching functionality. Only the identifiers of the entities should return from the Search Builder. Convenient methods should be provided to create the applicable filters.

1.2.3 Auditing Fields

Review must also include auditing fields of creation/modification operator and timestamp. These fields will not be provided by component users.

1.2.4 Persistence

Persistence needs to be pluggable. For this release an Informix plug-in will be developed. The SQL scripts will be provided.

1.2.4.1 Persistence Implementation

The persistence implementation needs to be designed in this component, but will be separated into a second development project. Please put all persistence implementation related information into a separate sub-package and clearly mark the responsibilities of the two development projects.

1.2.5 Thread-Safety

The component is required to be thread-safe.

1.3 Required Algorithms

No specific algorithms are required.

1.4 Example of the Software Usage

A scorecard/review application can use the component as a model layer. Application user can create and modify reviews on the web interface.

1.5 Future Component Direction

Any enhancement needs to be approved either in forum or in email with managers to eliminate over-complicating the component with useless functions.



2. Interface Requirements

2.1.1 Graphical User Interface Requirements

None.

2.1.2 External Interfaces

The component upgrade should be backwards-compatible with the previous version.

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.management.review`
`com.topcoder.management.review.persistence`

3. Software Requirements

3.1 Administration Requirements

3.1.1 What elements of the application need to be configurable?

- Database connection

3.2 Technical Constraints

3.2.1 Are there particular frameworks or standards that are required?

JDBC.

3.2.2 TopCoder Software Component Dependencies:

- [Base Exception 2.0.0](#)
- [Configuration Manager 2.2.0](#)
- [Review Data Structure 1.0.0](#)
- [Search Builder 1.3.2](#)
- [DB Connection Factory 1.1.0](#)
- [ID Generator 3.0.0](#)
- [Logging Wrapper 1.2.0](#)
- [Data Validation 1.1.1](#)
- [Database Abstraction 1.1](#)

**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:

Any third party library needs to be approved.

3.2.4 QA Environment:

- Java 1.5
- RedHat Linux 4



- Windows 2000
- Windows 2003

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.