This page last changed on Jun 12, 2010 by ghostar.

Specification Review Service Requirements Specification

Scope

Overview

TopCoder is performing an improvement the competition review process, which includes altering the way reviews are obtained by applicants, and how the reviews are performed. This undertaking involves modifications and additions to the existing Online Review application.
Currently there are two applications involved in review process. TopCoder Website application is used by the reviewers for signing up for the review positions and listing the statistics for projects and users. Online Review application is actually used for managing the review process providing the users with abilities to upload submissions for contests, fill the screening and review scorecards, submitting and resolving appeals, performing contest results aggregation, submitting final fixes and approving the results of contests. It is the later application that is being improved.
This contest will provide a component that will manage aspects of a specification review.

Version

1.0

Logic Requirements

This component will provide the interface, entity, and exception defined in the Reviewer Payment Calculator Component Interface Diagram. It will also provide an implementation of the interface.

SpecificationReviewService

This interface defines the contract for managing aspects of a specification review programmatically. It will be deployed as a web service in the same manner as the Cockpit's Contest Service Façade. The API outline is defined in the Specification Review Service Interface Diagram.

API details

scheduleSpecificationReview

This method will set the specification review phase to occur at the given fixed date (or immediately if date is null or in the).
This method will perform the following steps (very similar to the phase preparation steps in processContestSaleInternal method of ContestServiceFacadeBean).

- Get project with ProjectServices.getProject
- Get phases from project
    - Extract the phases for specification submission and review (use configurable IDs for each)
- Create a mock submission with configurable file path and name: new DataHandler(new FileDataSource(mockSubmissionFilePath + mockSubmissionFileName))
- Submit mock submission with UploadExternalServices.uploadSpecification
- Set the start date of the submission review phase to the passed date.
- Update phases with ProjectServices.updatePhases

Copyright TopCoder, Inc. 2008

submitSpecificationAsFile, submitSpecificationAsString

The purpose of these methods is to upload a specification, for the given project by the given user. These methods are used for an initial submission or an updated submission.

- Create a submission with configurable file path and name: new DataHandler(new FileDataSource(filename))
  - With a String, first place the content into a temp file
- Submit mock submission with UploadExternalServices.uploadSpecification

getSpecificationReview

This gets the review details for the given project.

- Get submission of specification submission type (configurable) from UploadManager
- Get review associated with the submission using the ReviewManager
- Use the scorecard ID in the Review to get the Scorecard from the ScorecardManager
- Bundle the Review and Scorecard into a SpecificationReview and return it.

getSpecificationReviewStatus

Gets the review status of the given project.

- Get project with ProjectServices.getProject
- Get phases from project
  - Extract the phases for specification submission and review (use configurable IDs for each)
- If the specification review phase is current, return SpecificationReviewStatus.PENDING_REVIEW
- If the specification submission phase is current, return SpecificationReviewStatus.WAITING_FOR_FIXES
- If neither phase is active, return null

getOpenSpecificationReviewPositions

Gets the IDs of all projects whose specification review positions are yet not filled.

- Get all active projects with ProjectServices.findActiveProjectHeaders
- For each project
  - Search for all resources for given project for the role of resource role for specification review (configurable) with the ReviewManager
- Return list of all project IDs that do not have a resource assigned for the specification review

Exceptions

The business method will throw SpecificationReviewServiceException with anything goes wrong during the execution of the method. It extends BaseException from the Base Exception component. Contestants can extend it to provide more granular exceptions.
If the component performs any configuration during construction, it may define a custom exception to throw if anything goes wrong during construction. The details are up to the contestant, but it will extend BaseRuntimeException from the Base Exception component.

Logging

The implementation will log exceptions using the Logging Wrapper component at ERROR level.

Copyright TopCoder, Inc. 2008

Configuration

Configuration will rely primarily on injection. It may use Configuration Manager and Object Factory to create instances of managers if needed.

Thread-safety

The manager's implementations should be thread-safe.

Method argument handling

This section describes the generic convention for expected handling of arguments, and may not refer to any specific operation for this component. However, any operation defined in this component that is affected by these conventions may override part or all of this. Such an override will be explicitly mentioned in the operation descriptions above.

- If getting an entity and it is not found, a method will return null.
- If getting a list of entities, and none are found, a method will return an empty list.
- Any list handled by the component must not contain null elements.
- Unless stated, all input arguments will be required (non-null). String arguments must not be null/ empty.

If any input parameter requirement is not met, an IllegalArgumentException will be thrown.

Required Algorithms

None

Example of the Software Usage

This component allows cockpit users to manage aspects of a specification review independently of Online Review.

Enhancement policy

In order to eliminate superfluous, useless, and/or bloated enhancements from the application, the following policy on enhancements is in effect for this competition.
All major enhancements must be explicitly approved by the architect (the approval of PM and/or co-pilot is not sufficient). All enhancements proposed in the future direction section are considered to be approved. Only if the architect approves the enhancement may it be added to a design. Any attempt to add a major enhancement to a design without this approval will result in that enhancement to not be eligible for a score of 4 in the requirements section (unless this idea happens to correspond to another submission's enhancement that was approved).
You may outline the enhancement proposal in the forum. You may also contact the architect directly to retain the privacy of your ideas. After the conclusion of the submission phase, the architect will notify the reviewers of the approval so they may score for it.
Be aware that the approval of an architect does not automatically assure a 4 in the requirements section. The architect will approve an enhancement or enhancements based on how useful and pertinent they are to the application. The reviewers, though, will decide if the enhancement or sum of enhancements is substantial. It is possible that the architect may advise the reviewers of how substantial they may be to the application, but the final decision will be in the hands of the reviewers.
When making an enhancement request via Contact Manager, please put the following in your first line:

Enhancement Request

At this time, it may also help to contact this architect directly with Member Contact since Contact Manager does not send a notification to the architect. This would most likely expedite the process.

Copyright TopCoder, Inc. 2008

Future Component Direction

None

Interface Requirements

Graphical User Interface Requirements

None

External Interfaces

The design must adhere to outline of the Specification Review Service Interface Diagram.

Environment Requirements

- Development language: Java1.5, J2EE 1.5
- Compile target: Java1.5, J2EE 1.5

Package Structure

com.topcoder.service.review.specification

Software Requirements

Administration Requirements

What elements of the application need to be configurable?

- Logging
- Specification submission phase ID
- Specification review phase ID
- Mock submission file path
- Mock submission file name
- Submission type ID for specification
- Name of resource role for specification review

Technical Constraints

Are there particular frameworks or standards that are required?

- EJB 3.0
- JAX-WS 2.0

TopCoder Software Component Dependencies:

- Base Exception 1.0
- Logging Wrapper 1.2
- Object Factory 2.1.2
- Object Factory Config Manager Plugin 1.0
- Configuration Manager 2.1.5
- Deliverable Management 1.2

Copyright TopCoder, Inc. 2008

- Project Services 1.0
- Scorecard Manager 1.0
- Scorecard Data Structure 1.0
- Review Manager 1.0
- Review Data Structure 1.0

**Please review the TopCoder Software component catalog for existing components that can be used in the design.

Third Party Component, Library, or Product Dependencies:

None

QA Environment:

- Java 1.5
- JBoss 4.0.2
- Informix 11
- JSP 2
- Flex 3

Design Constraints

The component design and development solutions must adhere to the guidelines as outlined in the TopCoder Software Component Guidelines.

Required Documentation

Design Documentation

- Use-Case Diagram
- Class Diagram
- Sequence Diagram
- Component Specification

Help / User Documentation

- Design documents must clearly define intended component usage in the 'Documentation' tab of the TopCoder UML Tool.

Copyright TopCoder, Inc. 2008