[TopCoder]

Specification Review Service 1.0 Component Specification

1. Design

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 component provides functionality for managing aspects of a specification review.

1.1 Design Patterns

Strategy pattern – this component provides SpecificationReviewService interface and its implementation that can possibly be used in some external strategy context.

DAO/DTO pattern – SpecificationReviewService acts as a DAO for SpecificationReview DTO (but provides only a retrieval operation for it).

1.2 Industry Standards

EJB 3.0, IoC

1.3 Required Algorithms

1.3.1 Logging

This component must perform logging in all public methods of SpecificationReviewServiceBean.

All information must be logged using log:Log attribute. If log attribute is null, then logging is not required to be performed.

All errors (for all thrown exceptions) must be logged at ERROR level with exception message and stack trace.

Other information is not required to be logged.

1.4 Component Class Overview

SpecificationReview

This class is a container for information about a single specification review. It holds Review and Scorecard instances. It is a simple JavaBean (POJO) that provides getters and setters for all private attributes and performs no argument validation in the setters.

SpecificationReviewService [interface]

This interface represents a specification review service. It provides methods for scheduling specification review, submitting specification as a file or a string, retrieving specification review, retrieving specification review status and retrieving IDs of projects for which specification review positions are open.

SpecificationReviewServiceBean

This class is an EJB that implements SpecificationReviewService business interface. This bean uses Logging Wrapper component to log all occurred errors. This class uses pluggable implementations of UploadManager, UploadExternalServices, ProjectServices, ReviewManager, ScorecardManager and ResourceManager interfaces to access data in persistence and perform the actual specification review management.

SpecificationReviewServiceLocal [interface]

This interface represents the local interface for SpecificationReviewService session bean. It extends that interface and provides no additional methods.

SpecificationReviewServiceRemote [interface]
This interface represents the remote interface for SpecificationReviewService session bean. It extends that interface and provides no additional methods.

SpecificationReviewStatus [enum]
This is an enumeration for specification review statuses.

1.5 Component Exception Definitions

SpecificationReviewServiceConfigurationException
This exception is thrown by SpecificationReviewServiceBean when error occurs while initializing this bean.

SpecificationReviewServiceException
This exception is thrown by implementations of SpecificationReviewService when some error occurs in the business methods.

1.6 Thread Safety

This component is thread safe.

Implementations of SpecificationReviewService, SpecificationReviewServiceLocal and SpecificationReviewServiceRemote must be thread safe.

SpecificationReview is mutable and not thread safe entity.

SpecificationReviewStatus is immutable and thread safe.

SpecificationReviewServiceBean is mutable and not thread safe. But it is always used in thread safe manner in EJB container because its state doesn't change after initialization. Instances of ProjectServices, UploadExternalServices and DBConnectionFactory used by this class are thread safe. Instances of UploadManager, ReviewManager, ScorecardManager and ResourceManager are not thread safe, thus all calls of their methods are synchronized in this class.

SpecificationReviewServiceBean uses container managed transactions. All methods of this class that can change data in the persistence have @TransactionAttribute(TransactionAttributeType.REQUIRED) annotation.

2. Environment Requirements

2.1 Environment

Development language: Java1.5, J2EE 1.5
Compile target: Java1.5, J2EE 1.5
QA Environment: Java 1.5, JBoss 4.0.2, Informix 11, JSP 2, Flex 3

2.2 TopCoder Software Components

Base Exception 2.0 – is used by custom exceptions defined in this component.

Logging Wrapper 2.0 – is used for logging errors in this component.

Deliverable Management 1.1 – defines UploadManager together with Submission entity and SubmissionFilterBuilder used in this component for retrieving specification submissions.

Project Services 1.1 – defines ProjectSerives and FullProjectData used in this component.

Project Management 1.0 – defines Project entity used in this component.

Scorecard Management 1.0.1 – defines ScorecardManager used in this component for retrieving scorecards.

Scorecard Data Structure 1.0 – defines Scorecard entity used in this component.

Review Management 1.0 – defines ReviewManager used in this component for retrieving specification review worksheets.

Review Data Structure 1.0 – defines Review entity used in this component.

Resource Management 1.1 – defines ResourceManager and ResourceFilterBuilder used in this component for checking specification reviewers for projects.

Online Review Upload Services 1.1 – defines UploadExternalServices used in this component for uploading specification submissions.

Online Review Phases 1.4 – defines SubmissionStatusLookupUtility, SubmissionTypeLookupUtility and ResourceRoleLookupUtility used in this component.

Security Manager 1.1 – defines TCSubject entity used in this component.

DB Connection Factory 1.1 – is used for establishing database connections.

Search Builder 1.4.1 – defines Filter, EqualToFilter and AndFilter entities used in this component for filtering submissions, projects and resource roles.

Project Phases 2.0.1 – defines Phase, PhaseStatus and PhaseType entities used in this component for checking/updating project phases.

NOTE: The default location for TopCoder Software component jars is../lib/tcs/COMPONENT_NAME/COMPONENT_VERSION relative to the component installation. Setting the tcs_libdir property in topcoder_global.properties will overwrite this default location.

2.3 Third Party Components

None

NOTE: The default location for 3rd party packages is ../lib relative to this component installation. Setting the ext_libdir property in topcoder_global.properties will overwrite this default location.

3. Installation and Configuration

3.1 Package Name

com.topcoder.service.review.specification
com.topcoder.service.review.specification.ejb

3.2 Configuration Parameters

3.2.1 EJB configuration of SpecificationReviewServiceBean

SpecificationReviewServiceBean is configured with the following EJBs via container injection:

EJB name	Description	Type
ejb/ProjectServices	The project services instance used by this class.	ProjectServices

Also SpecificationReviewServiceBean uses the following resources injected by EJB container:

Resource name	Description	Values
loggerName	The name of the Logger Wrapper logger to be used by this class for logging errors and debug information. When not specified, logging is not performed.	String. Not empty. Optional.
connectionName	The connection passed to DBConnectionFactory when establishing a database connection. If not specified, the default connection is used.	String. Not empty. Optional.
mockSubmissionFilePath	The mock specification submission file path (without file name).	String. Not empty. Required.

mockSubmissionFileName	The mock specification submission file name (without file path).	
uploadManagerClassName	The full class name of project manager to be used by this class.	String. Not empty. Required.
uploadManagerNamespace	The namespace of project manager to be used by this class.	String. Optional.
reviewManagerClassName	The full class name of review manager to be used by this class.	String. Not empty. Required.
reviewManagerNamespace	The namespace of review manager to be used by this class.	String. Optional.
scorecardManagerClassName	The full class name of scorecard manager to be used by this class.	String. Not empty. Required.
scorecardManagerNamespace	The namespace of scorecard manager to be used by this class.	String. Optional.
resourceManagerClassName	The full class name of resource manager to be used by this class.	String. Not empty. Required.
resourceManagerNamespace	The namespace of resource manager to be used by this class.	String. Optional.
uploadExternalServicesClassName	The full class name of upload external services to be used by this class.	String. Not empty. Required.
uploadExternalServicesNamespace	The namespace of upload external services to be used by this class.	String. Optional.
dbConnectionFactoryClassName	The full class name of DB connection factory to be used by this class.	String. Not empty. Required.
dbConnectionFactoryNamespace	The namespace of DB connection factory to be used by this class.	String. Optional.

3.3 Dependencies Configuration

Please see docs of all the dependency components to configure them properly.

4. Usage Notes

4.1 Required steps to test the component

- Extract the component distribution.
- Follow Dependencies Configuration.
- Execute 'ant test' within the directory that the distribution was extracted to.

4.2 Required steps to use the component

Please see the demo.

4.3 Demo

4.3.1 API usage

```
// Get specification review service
Context context = new InitialContext();
SpecificationReviewService specificationReviewService =
    (SpecificationReviewServiceRemote)
context.lookup("SpecificationReviewServiceBean/remote");

TCSubject tcSubject = ...

// Schedule specification review for the project with ID=1 to start immediately
long projectId = 1;
Date reviewStartDate = new Date();
```

```
specificationReviewService.scheduleSpecificationReview(tcSubject, projectId,
reviewStartDate);

// Retrieve the open specification review positions
List<Long> projectIds =
specificationReviewService.getOpenSpecificationReviewPositions(tcSubject);
// projectIds must contain 1

// Assume that specification reviewer registered for this project
...

// Retrieve the open specification review positions again
projectIds = specificationReviewService.getOpenSpecificationReviewPositions(tcSubject);
// projectIds must not contain 1

// Assume that specification was rejected here
...

// Re-submit specification as file for this project
String filename = "sample_component_1.0_requirements_specification.rtf";
long submissionId = specificationReviewService.submitSpecificationAsFile(tcSubject,
projectId, filename);

// Retrieve the specification review status
SpecificationReviewStatus specificationReviewStatus =
specificationReviewService.getSpecificationReviewStatus(tcSubject, projectId);
// specificationReviewStatus must be SpecificationReviewStatus.PENDING_REVIEW

// Assume that specification was rejected here again
...

// Retrieve the specification review status
specificationReviewStatus =
specificationReviewService.getSpecificationReviewStatus(tcSubject, projectId);
// specificationReviewStatus must be SpecificationReviewStatus.WAITING_FOR_FIXES

// Re-submit specification as content string
String content = "This is a text for sample specification";
submissionId = specificationReviewService.submitSpecificationAsString(tcSubject, projectId,
content);

// Assume that specification was accepted here
...

// Retrieve the specification review for this project
SpecificationReview specificationReview =
specificationReviewService.getSpecificationReview(tcSubject, projectId);
// specificationReview.getReview() should not be not
// specificationReview.getScorecard() should not be null
```

4.3.2 *Sample EJB configuration (ejb-jar.xml)*

```
<?xml version="1.0" encoding="utf-8"?>
<ejb-jar xmlns="http://java.sun.com/xml/ns/javaee"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation=
    "http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/ejb-jar_3_0.xsd"
         version="3.0">
  <description>Specification Review Service EJB</description>
  <display-name>Specification Review Service EJB</display-name>
  <enterprise-beans>
    <session>
      <ejb-name>SpecificationReviewServiceBean</ejb-name>
      <remote>
        com.topcoder.service.review.specification.ejb.SpecificationReviewServiceRemote
      </remote>
      <local>
        com.topcoder.service.review.specification.ejb.SpecificationReviewServiceLocal
      </local>
      <ejb-class>
        com.topcoder.service.review.specification.ejb.SpecificationReviewServiceBean
      </ejb-class>
```

```
<session-type>Stateless</session-type>
<transaction-type>Container</transaction-type>
<env-entry>
  <env-entry-name>loggerName</env-entry-name>
  <env-entry-type>java.lang.String</env-entry-type>
  <env-entry-value>specification_review_service_log</env-entry-value>
</env-entry>
<env-entry>
  <env-entry-name>connectionName</env-entry-name>
  <env-entry-type>java.lang.String</env-entry-type>
  <env-entry-value>my_connection</env-entry-value>
</env-entry>
<env-entry>
  <env-entry-name>mockSubmissionFilePath</env-entry-name>
  <env-entry-type>java.lang.String</env-entry-type>
  <env-entry-value>../mock</env-entry-value>
</env-entry>
<env-entry>
  <env-entry-name>mockSubmissionFileName</env-entry-name>
  <env-entry-type>java.lang.String</env-entry-type>
  <env-entry-value>mock_rs.rtf</env-entry-value>
</env-entry>
<env-entry>
  <env-entry-name>uploadManagerClassName</env-entry-name>
  <env-entry-type>java.lang.String</env-entry-type>
  <env-entry-value>
    com.topcoder.management.deliverable.PersistenceUploadManager
  </env-entry-value>
</env-entry>
<env-entry>
  <env-entry-name>reviewManagerClassName</env-entry-name>
  <env-entry-type>java.lang.String</env-entry-type>
  <env-entry-value>
    com.topcoder.management.review.DefaultReviewManager
  </env-entry-value>
</env-entry>
<env-entry>
  <env-entry-name>scorecardManagerClassName</env-entry-name>
  <env-entry-type>java.lang.String</env-entry-type>
  <env-entry-value>
    com.topcoder.management.scorecard.ScorecardManagerImpl
  </env-entry-value>
</env-entry>
<env-entry>
  <env-entry-name>resourceManagerClassName</env-entry-name>
  <env-entry-type>java.lang.String</env-entry-type>
  <env-entry-value>
    com.topcoder.management.resource.persistence.PersistenceResourceManager
  </env-entry-value>
</env-entry>
<env-entry>
  <env-entry-name>uploadExternalServicesClassName</env-entry-name>
  <env-entry-type>java.lang.String</env-entry-type>
  <env-entry-value>
    com.cronos.onlinereview.services.uploads.impl.DefaultUploadExternalServices
  </env-entry-value>
</env-entry>
<env-entry>
  <env-entry-name>dbConnectionFactoryClassName</env-entry-name>
  <env-entry-type>java.lang.String</env-entry-type>
  <env-entry-value>
    com.topcoder.db.connectionfactory.DBConnectionFactoryImpl
  </env-entry-value>
</env-entry>
<ejb-ref>
  <ejb-ref-name>ejb/ProjectServices</ejb-ref-name>
  <ejb-ref-type>Session</ejb-ref-type>
  <remote>
    com.topcoder.project.service.ejb.ProjectServicesRemote
  </remote>
</ejb-ref>
```

```
    </session>
  </enterprise-beans>
</ejb-jar>
```

5. Future Enhancements

None