Deliverable Management Persistence 1.1 Requirements Specification

Scope

Overview

Specification reviews are currently carried out as separate projects in Online Review (OR). This architecture will design a number of enhancements to the current specification review system, allowing TopCoder staff, clients, and copilots to better track and manage specification reviews. This module architecture will provide some changes to how specification reviews are handled, and will expose services that can be used by the new cockpit to automatically create and edit specification reviews without a client having to go into OR.
The current Project Phase Template provides the ability to generate Project Phases from a template. Current strategies include an XML template and a DB template. A user can request that a project shell be generated with phases as they are defined in the named template.
Because these upgrades require the submission of a specification, a new submission type will now need to exist in the system. However, the Deliverable Management component does not support this concept. As such, it will be upgraded to contain it. The persistence will also need to be upgraded to support the management of the type as well as the submission that will now aggregate it.

Version

1.1

> ⚠ Note that version 1.1 adds the following additional features and changes, but all existing functionality must also be included. This is an upgrade, not an entirely new component.

Logic Requirements

SqlUploadPersistence

The SqlUploadPersistence will be updated to conform to the new methods in the UploadPersistence as indicated by the Deliverable Management Updates Class Diagram in the TCUML. The new management methods will be closely based on the existing submission status management methods.
Also, the methods that persist a Submission in the existing SqlUploadPersistence class need to be updated to include the persistence of the submission's type.

ERD

An ERD showing the updated aspects of the submission and the submission type is now provided.

Project documentation update

As part of the upgrade process, the existing project documentation must be updated to TopCoder standards:

- The project must be ported to TCUML. It currently exists in ZUML
  - The documentation nodes in TCUML must be cleaned up of any ZUML artifacts, including artifact HTML tags to make the notes readable.
- The component documentation must incorporate any changes that exist in the source.

Required Algorithms

None

Example of the Software Usage

A user will manage new submissions that will be typed. New submission types will be managed via the API.

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.

Future Component Direction

None

Interface Requirements

Graphical User Interface Requirements

None

External Interfaces

The design must adhere to outline of the existing component plus the required new additions as outlined in the Deliverable Management Updates Class Diagram in the TCUML. Any changes to existing public API must be explicitly approved by PM in the forum.

Environment Requirements

- Development language: Java1.4
- Compile target: Java1.4

Package Structure

com.topcoder.project.phases.template.persistence.sql

Software Requirements

Administration Requirements

What additional elements of the application need to be configurable?

None

Technical Constraints

Are there particular frameworks or standards that are required?

- SQL
- JDBC

TopCoder Software Component Dependencies:

- Base Exception 1.0
- Search Builder 1.4.1
- DB Connection Factory 1.1
- Custom Result Set 1.1
- Configuration Manager 2.1.5
- Logging Wrapper 2.0
- Deliverable Management 1.3

**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.