Resource Management Requirements Specification

1. Scope

1.1 Overview

The component provides resource management functionalities. A resource can be optionally associated with a project, phase and submission. Each resource will have a role which identifies the resource's responsibilities for the associated scope. A set of resources can be created, updated or searched for a project. Also notifications can be assigned and unassigned to users. The resource persistence logic is pluggable.

1.2 Logic Requirements

1.2.1 Resource Entity

A resource will be associated with a role, optionally a project, a phase, a submission, as well as some extended properties. Please refer to 2.1.2 for details.

1.2.2 Resource Operations

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

1.2.2.1 Update Resource

An existing resource can be updated. Resource should be validated according to the defined rules.

The operator needs to be provided for auditing purpose.

1.2.2.2 Update Project Resources

An array of resources can be updated for a specified project. The identifiers of any new resources will be provided with ID Generator. Existing resource should continue to use the original identifiers. Other resources for the project not specified in the array should be deleted. Resources should be validated according to the defined rules.

The operator needs to be provided for auditing purpose.

1.2.2.3 Get Resource

A resource can be retrieved by specified identifier.

1.2.2.4 Search Resources

An array of resources can be retrieved by specified search criteria.

1.2.2.4.1 Resource Search Criteria

At minimum, the criteria should be capable of specifying any combination of project, phase, submission, resource role and extended properties. There should be ways to specify resource with no project assignment, no phase assignment or no submission assignment.

1.2.2.5 All Resource Roles

There should be a way to retrieve all resource roles in the system.

1.2.2.6 Add Notifications

A notification entry will associate a user external ID with a project and notification type. The operation should be able to add multiple external ID's to the same project and notification type.

The operator needs to be provided for auditing purpose.

1.2.2.7 Remove Notifications

The operation should be able to remove multiple external ID's from the same project and notification type.

The operator needs to be provided for auditing purpose.

1.2.2.8 Get Notifications

The operation should be able to retrieve all external ID's associated with a project and notification type.

1.2.2.9 All Notification Types

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

1.2.3 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.

1.2.4 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.

1.3 Required Algorithms

No specific algorithms are required.

1.4 Example of the Software Usage

A project management application can use the component as a model layer. Customer specific information can be stored as extended properties.

1.5 Future Component Direction

None.

2. Interface Requirements

2.1.1 Graphical User Interface Requirements

None.

2.1.2 External Interfaces

Design must adhere to the interface diagram definition. Designer can choose to add more methods to the classes/interfaces, but must keep the ones defined on the diagram as a minimum. Source files can be found in the distribution.

2.1.2.1 Resource Role

A resource will be associated with a resource role. A resource role consists of

ID	a numeric identifier
Name	name of the scorecard type for display purpose
Phase Type	a resource role could be optionally confined to a phase type. For instance, an approver only applies to a certain approval phase. This could be represented as a numeric identifier.

2.1.2.2 Resource

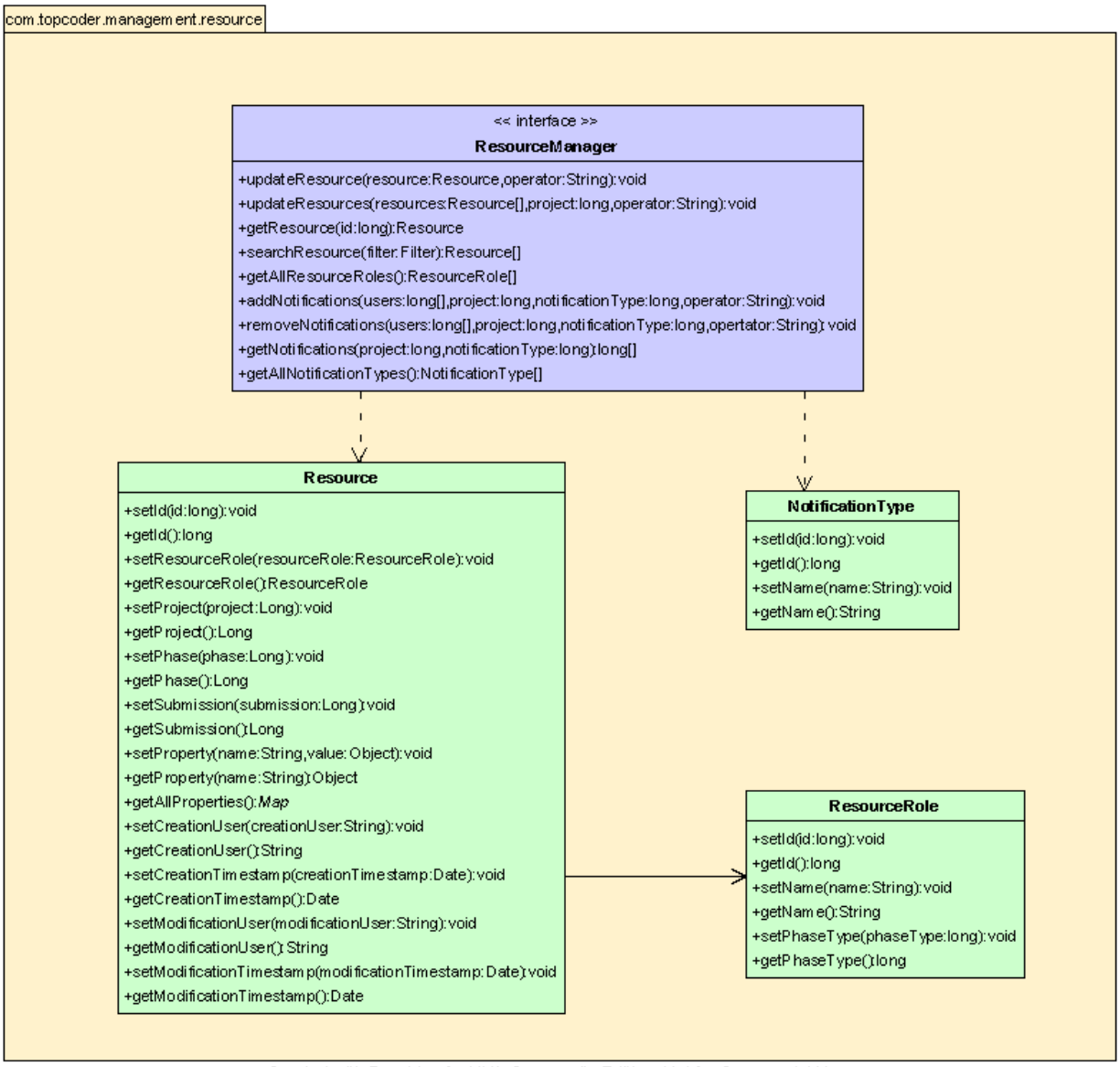
A resource consists of

ID	a numeric identifier
Role	each resource will have a resource role
Project	a resource could be optionally associated with a project. This could be represented as a numeric identifier.
Phase	a resource should be associated with a phase if the corresponding resource role is associated with a phase type. This could be represented as a numeric identifier.
Submission	a resource could be optionally associated with a submission. This could be represented as a numeric identifier.
Extended Properties	each resource will maintain a set of unordered extended properties

2.1.2.3 Auditing Fields

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

Resource Management Interface Diagram



Created with Poseidon for UML Community Edition. Not for Commercial Use.

2.1.3 Environment Requirements

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

2.1.4 Package Structure

com.topcoder.management.resource

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:

- Configuration Manager
- DB Connection Factory
- ID Generator
- Search Builder

**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:

None.

3.2.4 QA Environment:

- Solaris 7
- RedHat Linux 7.1
- Windows 2000
- Windows 2003
- Informix 10.0

3.3 Design Constraints

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

3.3.1 Database Connections

Database connections must not be cached within the component. Connections should be created for each operation and closed afterwards.

3.3.2 Component Scalability

The component needs to be scalable. Running multiple instances in the same JVM or in multiple JVM's concurrently should not cause any problem.

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 Poseidon.