



TopCoder Cockpit Project Metadata Entities Requirements Specification

1. Scope

1.1 Overview

In the TopCoder Cockpit application, we are going to allow user to associate metadata to the cockpit project. The project metadata consists of predefined metadata and custom metadata.

The project metadata are generally key / value pairs associated with the cockpit project. For predefined metadata, the key are predefined, the user only needs to set the value. For custom metadata, the user needs to set a key first, then set a value for that key. The metadata value can be specified by the user with any value or choose from a predefined list.

For project metadata types, we can only set one value, for example, for the metadata 'budget', we only about one value set for the budget of the project.

For other project metadata types, for example, the metadata 'project technologies', we allow multiple values to be set like 'Java', 'Web application', 'EJB3', 'struts2' for example.

This component is responsible for defining the entities for the back end services.

1.2 Logic Requirements

1.2.1 DAO and DTO Entities

This component is responsible for implementing the DAO entities defined in the `com.topcoder.direct.services.project.metadata.entities.dao` package and the DTO entities defined in the `com.topcoder.direct.services.project.metadata.entities.dto` package.

All entities should be serializable and follow the Java Bean convention.

1.2.2 toJSONString

All entities will provide a JSON version of their data, to be supplied in a `toJSONString():String` method. This is used to standardize the logging of input and out information of entities.

1.2.3 JPA Mapping Files

JPA mapping files should be provided for the DAO entities.

The cascade attribute value for `ProjectMetadata.projectMetadataKey` is none as the `ProjectMetadataKey` entity is a lookup entity. The cascade attribute value for `ProjectMetadataKey.predefinedValues` is all as the predefined values only make sense when associated with its key.

1.2.4 Database

The new database tables defined in ERD will be created in the `tcs_catalog` database.

The existing `tc_direct_project` database table is from the `corporate_oltp` database.

1.2.5 Predefined Metadata Keys

SQL script should be created to insert the following common metadata keys (whose `clientId` & grouping attributes are null). The SQL script should also insert the predefined values for the



“Project Status” and “Project Technologies” metadata keys.

Common Metadata Key	Description	Value Format
Budget	the budget of the project	positive integer value
Project Status	the status of the project	choose from the predefined value list: 'Active', 'On Hold', 'Archived', 'Completed'
Project Technologies	The technologies the project uses	choose from the predefined value list: Please ask PM to provide it
Planned Duration	The scheduled duration of the project	a positive integer value represents days like <i>120 days</i>
Client handles	The topcoder handles of client stakeholders of the project	Valid TC handle
TC PM handles	The topcoder handles of the topcoder project manager	Valid TC handle
Private Flag	whether the project is a private	true or false
Project Jira Key	The key of Jira project which is set up for this cockpit project	Non-empty string
Project Forum Id	The id of project forum where client / PM / Copilot discuss about the project	positive integer value
Project SVN address	The SVN URL used as the repository of the cockpit project	Valid SVN URL address

1.2.6 Cache

Hibernate level-1 cache should be configured for the lookup entities to achieve better performance.

The DirectProjectMetadataKey and DirectProjectMetadataPredefinedValue are lookup entities.

1.3 Required Algorithms

None

1.4 Example of the Software Usage

This component provides the entities for the services.

1.5 Future Component Direction

None

2. Interface Requirements

2.1.1 Graphical User Interface Requirements

None

2.1.2 External Interfaces

Refer to the provided architecture TCUML.

2.1.3 Environment Requirements

- Development language: Java
- Compile Target: Java 1.5

2.1.4 Package

com.topcoder.direct.services.project.metadata.entities.dao
com.topcoder.direct.services.project.metadata.entities.dto

3. Software Requirements

3.1 Administration Requirements

3.1.1 What elements of the application need to be configurable?

None

3.2 Technical Constraints

3.2.1 Are there particular frameworks or standards that are required?

JPA, JSON

3.2.2 TopCoder Software Component Dependencies:

JSON Object 2.0

3.2.3 Third Party Component, Library, or Product Dependencies:

- Spring 2.5.6: <http://www.springsource.org/>
- Hibernate 3.6.4: <http://www.hibernate.org/>

3.2.4 QA Environment:

- Java 1.5
- Informix 11.5
- JBoss 4.0.2

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

XML documentation must provide sufficient information regarding component design and usage.