[TopCoder]

Software Documentation : Project Management v1.2

Overview

The component provides project management functionalities. A project will be associated with category and status. Application can use the component to create, update and search projects. The project persistence logic is pluggable.

> ⚠ **Updated Requirements**
>
> **added requirements are in red**
> **updated requirements are in blue**

Logic Requirements

Version 1.2 updates the data model to accommodate new database fields and adds methods to the ProjectManager interface. Managers to create project prizes and studio specifications are added as well.

Logic Requirements

Data Model Updates

Project entity now has field for prizes.
New entities added are:
ProjectStudioSpecification - this entity contains specifications about a studio contest, like general guidelines, allowed colors and so on.
Prize - contains information about a project prize
PrizeType - a lookup keeping the value of the prize type, like milestone payment or a submission payment.
FileType - keeps information about file types which can be submitted for the project.

ProjectManager

The ProjectManager and its implementation will be updated to add the methods displayed on the uml class diagram from the architecture.
All methods are self evident and it should be clear what each one does.

Managers for new entities

CRUD methods for new entities can be decoupled into one or few managers.
Because entities are created from the frontend through selection or through direct edit, they need to be made accessible as service methods.

Project Entity

A project will be associated with a category, a status, as well as some extended properties. Please refer to 2.1.2 for details.

Project Operations

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

Create Project

A new project can be created. The identifier will be provided with ID Generator. Project should be validated according to the defined rules.
The operator needs to be provided for auditing purpose.

Update Project

An existing project can be updated. Project should be validated according to the defined rules.
The operator needs to be provided for auditing purpose. An extra explanation should be provided.

Get Project

A project can be retrieved by specified identifier.

Search Projects

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

Project Search Criteria

At minimum, the criteria should be capable of specifying any combination of project type, project category, project status and extended properties.

User Projects

There should be a way to return all the active projects associated with a user external ID.

All Categories

There should be a way to retrieve all project categories in the system.

All Statuses

There should be a way to retrieve all project statuses in the system.

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.

Persistence

Persistence needs to be pluggable. For this release an Informix plug-in will be developed. The SQL scripts will be provided.

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.

Required Algorithms

No specific algorithms are required.

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.

Future Component Direction

None.

Interface Requirements

Graphical User Interface Requirements

None.

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.

Project Category

A project will be associated with a project category. A project category consists of
ID a numeric identifier
Name name of the project category for display purpose
Project Typea project category further belongs to a project type
A project type consists of
ID a numeric identifier
Name name of the project type for display purpose

Project Status

A project will be associated with a project status. A project status consists of
ID a numeric identifier
Name name of the project status for display purpose
There will be a few built statuses for this release, including Active, Inactive and Deleted. More status may be added in the future.

Project

A project consists of
ID a numeric identifier
Statuseach project will have a status
Categoryeach project will belong to a category
Extended Propertieseach project will maintain a set of unordered extended properties

Auditing Fields

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

Environment Requirements

- Development language: Java1.5
- Compile target: Java1.5

Package Structure

com.topcoder.management.project

Software Requirements

Administration Requirements

What elements of the application need to be configurable?

- Database connection

Technical Constraints

Are there particular frameworks or standards that are required?

JDBC

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.

Third Party Component, Library, or Product Dependencies:

None.

QA Environment:

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

Design Constraints

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

Database Connections

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

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.

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 TC UML Tool.