Review Requirements Specification

1. Scope

1.1 Overview

Reviews are produced based on scorecards. A review holds a collection of items which address each of the questions on the scorecard. It also consists of the author that produced the review, the submission it addresses and the scorecard template it is based on. Various types of comments can be attached to the review or to each review item. A committed review must address all questions on the corresponding scorecard, and will have its overall score available. The component defines the scorecard data structure. Separate component will be developed to provide management functionality.

1.2 Logic Requirements

1.2.1 Review

A review consists of

ID	a numeric identifier
Resource	the numeric identifier to the reviewer who has produced the review
Submission	the numeric identifier to the submission the review is addressing
Scorecard	the numeric identifier to the scorecard the review uses
Committed	a Boolean flag indicating whether the review has been committed
Score	the score of the review, which is not available until the review is committed
Review Items	zero or more ordered review items as specified in 1.2.2. Each review item will address a question on the scorecard.
Comments	zero or more ordered comments for the review as specified in 1.2.3. The review comments are not necessarily produced by the same person as the reviewer.

1.2.2 Review Item

A review item consists of

ID	a numeric identifier
Question	the numeric identifier to the scorecard question the item is addressing
Answer	the answer to the question on the scorecard. The answer needs to be generic in order to cover a variety of question types.
Document Upload	for each question an attachment could be uploaded. This attribute is optional. If present it will be a numeric identifier.
Comments	zero or more ordered comments for the review item as specified in 1.2.3. The review item comments are not necessarily produced by the same person as the reviewer.

1.2.3 Review Comment

A comment could either address the complete review, or address a single item. It consists of

ID	a numeric identifier
Resource	the numeric identifier to the person who has placed the comment
Type	each comment will have a type as specified in 1.2.4
Comment	the textual comment itself
Extra Info	Extra information kept for the comment. Example of use would be whether the comment approves the review, or whether the appeal is resolved. This attribute is optional.

1.2.4 Review Comment Type

A review comment will be associated with a type. A review comment type consists of

ID	a numeric identifier
Name	name of the scorecard type for display purpose

1.2.5 Auditing Fields

Review must also include auditing fields of creation/modification operator and timestamp.

1.3 Required Algorithms

No specific algorithms are required.

1.4 Example of the Software Usage

A scorecard/review application can use the component to represent review in memory. Application user can create and modify reviews on the web interface.

1.5 Future Component Direction

Separate component will be developed to calculate review scores.

2. Interface Requirements

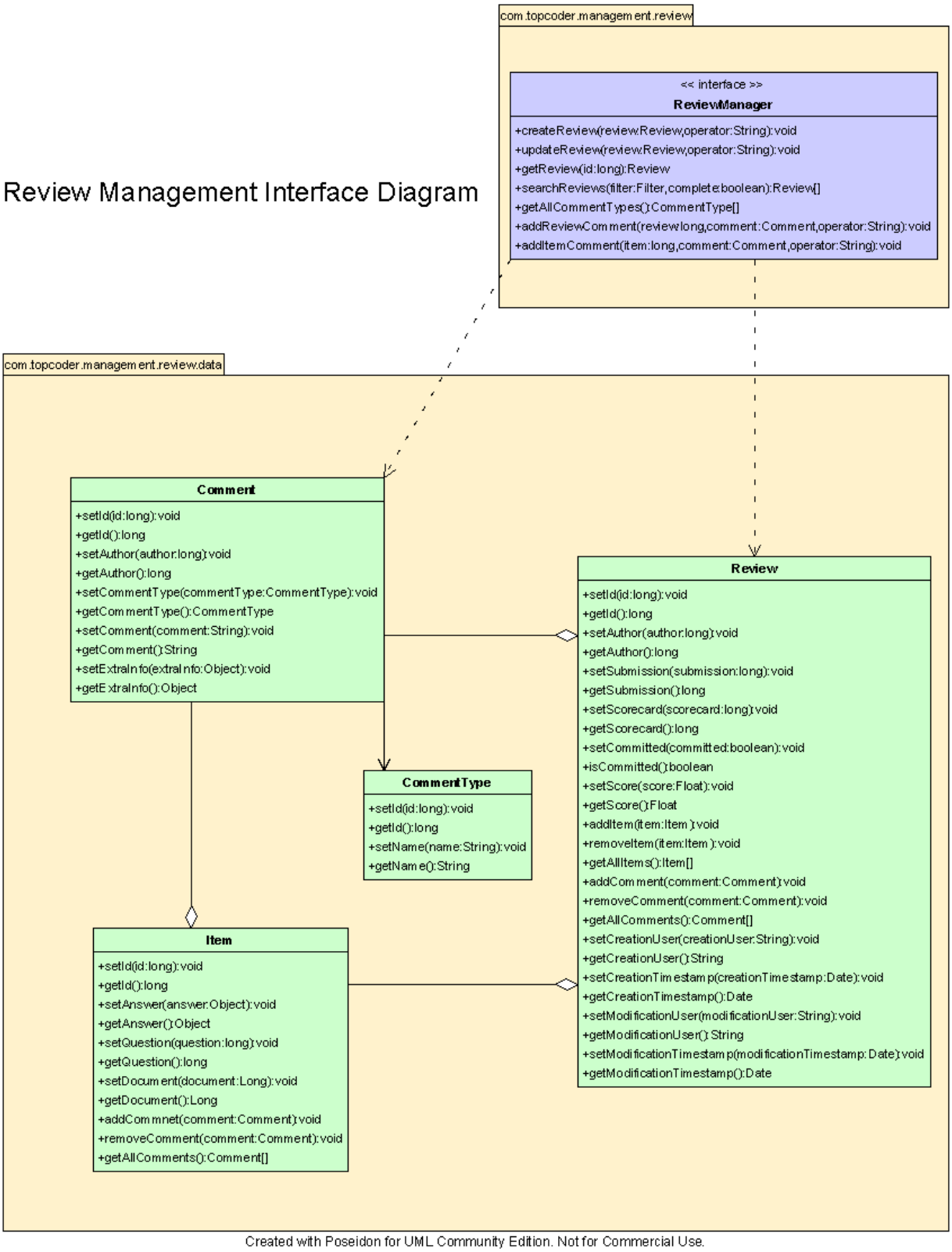
2.1.1 Graphical User Interface Requirements

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.

Review Management Interface Diagram



2.1.3 Environment Requirements

- Development language: Java1.4

- Compile target: Java1.4

2.1.4 Package Structure

com.topcoder.management.review.data

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?

None.

3.2.2 TopCoder Software Component Dependencies:

None.

**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

3.3 Design Constraints

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

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.