Online Review Design Specification

Author	Revision Number	Date
Jiazhi Wu	0.1	6/5/2006
Jiazhi Wu	0.2	6/10/2006

Application Design Specification

1. Design

1.1 System Description

Online Review is a system through which user will be able to submit and review project deliverables. Projects and scorecards can be created and modified by managers. In addition various project details can be viewed by authorized users.

The following sections will explain a few of the things that are not covered by the diagrams.

1.1.1 Review Scorecards

The review scorecards will be stored in the same generic structure defined by the Review Management component.

The following table explains the various scorecards that will be used.

Type	Author	Description
Screening Scorecard	Screener Primary Screener	Screening scorecard will be produced in the screening phase. The valid comments are Comment, Required, Recommended and Manager Comment.
Review Scorecard	Reviewer Test Case Reviewer	Review scorecard will be produced in the review phase. The valid comments are Comment, Required, Recommended, Appeal, Appeal Response and Manager Comment.
Aggregation Scorecard	Aggregator	Aggregation scorecard will be produced automatically upon the opening of the aggregation phase. The scorecard is based on the review scorecard and will copy all the comments from three individual reviewers into a single new review scorecard, with the extra info left as empty. The valid comments are Comment, Required, Recommended, Appeal, Appeal Response, Aggregation Comment, Aggregation Review Comment, Submitter Comment and Manager Comment. In case aggregation fails review, the author must be changed to the aggregator for the newly added aggregation phase. Extra info for Aggregation Review Comment and Submitter Comment must be cleared.
Final Review Scorecard	Final Reviewer	Final review scorecard will be produced automatically upon the opening of the final review phase. The scorecard is based on the aggregation scorecard and will copy all the comments marked as Accept. Non reviewer comments should also be copied. The extra info will be cleared. The valid comments are Comment, Required, Recommended, Appeal, Appeal Response, Aggregation Comment, Aggregation Review Comment, Submitter Comment, Final Fix Comment, Final Review Comment and Manager Comment. In case final fix fails review, the author must be changed to the final reviewer for the newly added final review phase. Extra info for Final Review Comment must be cleared.
Approval Scorecard	Approver	Approval scorecard will be produced in the approval phase. The valid comments are Comment, Required, Recommended and Manager Comment.

The following table explains the different comment type that will be used.

Comment Type	Level	Description	Extra Info
Comment	Item	Comments are made by screener, reviewer or approver on a scorecard. At least one Comment, Required or Recommended comment should be present for each item. Comment usually does not need to be addressed by the final fixes. During the aggregation phase, each comment will be marked as accept, reject or duplicate. Only accept entries will make into the final review scorecard. During the final review phase, each comment will be marked as fixed or not fixed. A final review scorecard can only be approved when all Required comments are fixed.	Accept/Reject/Duplicate on aggregation scorecard Fixed/Not Fixed on final review scorecard
Required	Item	Similar to Comment. Required comment must be addressed by the final fixes.	Same as Comment
Recommended	Item	Similar to Comment. Recommended comment is not obliged to be addressed by the final fixes.	Same as Comment
Appeal	Item	Added by submitter to each individual review during the appeal phase. Zero or one appeal will exist on each item. Each appeal needs to be addressed by an appeal response. After an appeal is resolved, its status of success of failure should be kept as extra info.	Succeeded/Failed after appeal is resolved
Appeal Response	Item	Added or edited by reviewer to their own review during the appeal response phase. When appeal response is added, the original answer should be duplicated when the appeal response is added.	The original answer after appeal is resolved
Aggregation Comment	Item	Aggregation Comment is added by the aggregator during the aggregation phase.	N/A
Aggregation Review Comment	Item	Aggregation Review Comment is added by the reviewers during the aggregation review phase. Each item could be approved or rejected. The whole aggregation review will be approved only if all items are approved.	Accept/Reject
Aggregation Review Comment	Review	Aggregation Review Comment is added to the review level as approval. Extra info should indicate whether the aggregation is approved by the reviewer. Aggregation	Approving/ Approved/Rejected

		should be approved by all the reviewers except one if it happens to be the same user as the aggregator.	
Submitter Comment	Item	Submitter Comment is added by the winning submitter during the aggregation review phase.	N/A
Submitter Comment	Review	Submitter Comment is added to the review level as approval. Extra info should indicate whether the aggregation is approved by the submitter. Aggregation should be approved by the submitter.	Approving/ Approved/Rejected
Final Fix Comment	Item	Final Fix Comment is added to the final review scorecard during the final fix phase.	N/A
Final Review Comment	Item	Final Review Comment is added by the final reviewer during the final review phase.	N/A
Final Review Comment	Review	Final Review Comment is added to the review level as approval. Extra info should indicate whether the final fix is approved by the final reviewer. Final fix should be approved by the final reviewer.	Approving/ Approved/Rejected
Manager Comment	Item	Manager Comment is added by the manager during most of the phases.	N/A

1.1.2 Scorecard Status

The following scorecard statuses will be used in this release.

Status	Description
Active	Active scorecards are those that will display to the manager when creating or updating project scorecard associations.
Inactive	Inactive scorecards are those that will not display to the manager when creating or updating project scorecard associations. Managers can set scorecards to inactive in scorecard admin, but the scorecards should continue to work on the projects that are already associated with them.
Deleted	Deleted scorecards will not display in either scorecard admin or during project management. They are only kept in database.

1.1.3 Scorecard Versioning

Simple versioning mechanism will be used for scorecards. Scorecard version will be split into major and minor versions.

Major version will be incremented automatically when a scorecard is copied so that a new scorecard is effectively produced. User will still be able edit the change.

Minor version will be incremented automatically when an existing scorecard is updated. User will not be able to edit the change. Scorecards that are associated with projects can not be updated. Only copies of the scorecard can be saved.

1.1.4 *Project Status*

The following project statuses will be used in this release.

Status	Description
Inactive	Inactive projects are only visible to managers.
Active	Active projects are visible to all resources, or to everyone if the project is flagged as public.
Deleted	Deleted projects will not display in the application. They are only kept in database.
Cancelled - Failed Review	All submissions have failed review. Projects will be changed to this status explicitly by managers and will not be visible to any one.
Cancelled - Failed Screening	All submissions have failed screening. Projects will be changed to this status explicitly by managers and will not be visible to any one.
Cancelled - Zero Submissions	No submission has been received. Projects will be changed to this status explicitly by managers and will not be visible to any one.
Completed	Projects have completed the review process. Projects will be changed to this status explicitly by managers and will not be visible to any one.

1.1.5 *Upload Type and Submission Status*

There will be various uploads in the application.

Upload Type	Description
Submission	Submissions are the deliverables that will be reviewed. Each resource with the submitter role can upload zero or more submissions and keep all of them active. Each submission will be auto screened but only the latest non deleted submission will be associated as the submitter's official submission.
Test Case	Test cases are the deliverables uploaded by each resource with the test case reviewer role. Only one test case will be active. If a new test case has been uploaded, the previous test case should be set to deleted and a notification email should be set.
Final Fix	Final fix is the deliverable uploaded by the winning submitter. Only one final fix will be active. When final review is rejected the previous final fix should be set to deleted.
Review Document	Reviewer document can be uploaded by resources with the reviewer role. They will be optionally associated with each review item.

A submitter could be associated with multiple uploads of the type submission, but only the latest one of them will be associated with a submission record. Once the submitter has uploaded a first submission, the submission record is created and the submission id will be used to identify all subsequent uploads. If a latest submission is deleted by managers the submission will then be associated with the next latest upload, unless there is none, in which case the submission is changed to deleted status.

Submission Status	Description
Active	Active submissions will participate in the current phases during the online review process. Submission can become inactive with various causes. Please refer to the follow rows for details.
Failed Screening	Submission failed screening will not proceed further into the process and stops at the submission or screening phase. The possible causes for it is that automated screening generates fatal errors or the screening score does not fall into the acceptable range.
Failed Review	Submission failed review will not proceed further into the process and stops at the appeal response phase. The only cause for it is that the final score (after appeal/response) does not fall into the acceptable range.
Completed Without Win	Submission that does not win the contest will not proceed further into the process and stops at the appeal response phase. Only the submission chosen as the first place will go into aggregation.
Deleted	Submission that is deleted by managers. This will only be used when the corresponding submitter has no active submission uploads. If a previous upload exists, the submission should be associated with the previous file.

1.1.6 Project Validation

When project information is created or updated, it must be validated. The validation mainly goes to the timeline and resources.

Timeline Validation Rules
Beginning phase must be either registration if it's present, submission otherwise.
If registration is present, submission must follow.
If screening is present, if must follow submission.
Review must follow screening if it's present, or submission otherwise.
If appeal is present, it must follow review.
If appeal is present, appeal response must follow.
If aggregation is present, it must follow appeal response if it's present, review otherwise.
If aggregation is present, aggregation review must follow.
Final fix must follow aggregation review if it's present, review otherwise.
Final review must follow final fix.
If approval is present, it can be inserted anywhere.
Ending phase must be either final review or approval.

Resource Validation Rules	
Submitter	Zero or more
Designer	Zero or one

Manager	Zero or one
Observer	Zero or more
Screener	One per submission, per screening phase (could be vacant) Should not exist together with Primary Screener
Primary Screener	One per screening phase (could be vacant) Should not exist together with Screener
Reviewer	One or more per review phase (could be vacant) Should not exist together with Test Case Reviewer
Test Case Reviewer	One per review phase (could be vacant) Should not exist together with Reviewer
Aggregator	One per aggregation phase (could be vacant)
Final Reviewer	One per final review phase (could be vacant)
Approver	One per approval phase (could be vacant)

1.1.7 Project Properties

The following table explains the project properties the application will use.

Name	Description
External Reference ID	External identifier (AKA component version id) that could uniquely identify a project in the TopCoder database.
Component ID	Identifier that uniquely identifies a project without versioning information.
Version ID	Identifier that identifiers a version for a particular project. The combination of component ID and version ID can be used to produce the link into the software catalog.
Developer Forum ID	Identifier that uniquely identifies the developer forum for the project. It can be used to produce the link into the developer forum.
Root Catalog ID	Identifier that represents the root catalog of the component. The root catalog tracks the platform of the component (Java/.NET).
Project Name	The displayable name of the project.
Project Version	The displayable version of the project.
SVN Module	The SVN module (link) for a component development project. The SVN module will only be visible to certain resources.
Autopilot Option	The auto pilot option for the project. The valid values are “On” and “Off”. More options could be provided in the future.
Status Notification	The status notification flag for the project. If set to “On” project status notification will be sent to managers, otherwise “Off”
Timeline Notification	The timeline notification flag for the project. If set to “On” project timeline notification will be sent to managers, otherwise “Off”
Public	Whether the project is public or private to TopCoder. Public projects will show up in all projects for normal users. Private projects will only

	display in my projects for associated resources. Valid values are "Yes" and "No".
Rated	Whether the project will be rated. Valid values are "Yes" and "No".
Eligibility	Eligibility requirement for the project. The values are TBD.
Payments Required	Whether payment is required for the project. Valid values are "Yes" and "No".
Notes	Extra notes for the project. Dynamic content that will display on the project details page.
Deactivated Timestamp	The timestamp when the project is deactivated. Only required when project is deactivated and will be formatted "00.00.0000 00:00 AM".
Deactivated Phase	The phase during which the project is deactivated. Literal value that is only required when project is deactivated.
Deactivated Reason	The reason why the project is deactivated. Dynamic content that is only required when project is deactivated.
Completion Timestamp	The timestamp when the project is completed. Will only be present after the project is completed and will be formatted "00.00.0000 00:00 AM".
Rated Timestamp	The timestamp when the project is rated. Will be filled externally.
Winner External Reference ID	The winner of the project. Will be the identifier from the member database. Populated when appeal response is closed and there is a winner.
Runner-up External Reference ID	The runner-up of the project. Same constraints as the winner.

1.1.8 Resource Properties

The following table explains the resource properties the application will use.

Name	Description
External Reference ID	External identifier that uniquely identifies a user in the TopCoder member database.
Handle	The user's handle.
First Name	The user's first name.
Last Name	The user's last name.
Email	The user's primary email.
Rating	The submitter's rating when they are registered. It could be either the design or dev rating. Will not contain the percentage mark. If not present it will be displayed as "N/A".
Reliability	The submitter's reliability when they are registered. It could be either the design or dev reliability. If not present it will be displayed as "Not Rated".
Registration Date	The submitter's registration timestamp. Will be formatted "00.00.0000 00:00 AM".

Payment	The payment for the resource. Will not contain the dollar mark. If not present it will be displayed as “N/A”.
Payment Status	The payment status for the resource. Will be either “Yes” or “No”, and will display as “Paid” or “Not Paid”. If not present it will be displayed as “N/A”.
Screening Score	Screening score for the submitter. Accurate to two decimal places.
Initial Score	Initial aggregated review score (before appeal/response) for the submitter. Accurate to two decimal places.
Final Score	Final aggregated review score (after appeal/response) for the submitter. Accurate to two decimal places.
Placement	The placement for the submitter if the submission has passed review.

1.1.9 Edit Any Scorecard

Managers must be able to edit any scorecards, including committed scorecards, which can be set to uncommitted. These are assumed be handled with the same actions and JSP pages. The actions and JSP pages will act differently depending on the role.

When screening, review and approval scorecards are edited, the manager can alter the question answers. During appeal response, the score will be updated when the scorecard is saved. When aggregation scorecard is edited, the manager can alter the accepted/rejected/duplicate response. When final review scorecard is edited, the manager can alter the fixed/not fixed response. On any scorecards the manager can add manager comments to the scorecard.

1.2 Work Flow Description

This section will describe work flow and potentially other details by project phases.

1.2.1 Project Creation

Project is created by users with the manager role. On the project creation page, user can either choose a timeline template name to generate the complete timeline and make adjustments, or enter the timeline manually. Once the form is submitted, the application mainly needs to validate and create the project information, project timeline and project resources. The resources need to be first resolved from external database (through the User Project Data Store API) with the user’s entry (handle). When the project is created, none of the phases are activated. The project will enter registration on the fixed start date if the project is active and the auto pilot option is on.

1.2.2 Registration

Registration phase will have a fixed start date and will not have dependencies so that it can also start on time. The application doesn’t do anything during the phase. Registration data will enter the database by external system. The registrants are associated with the submitter role, with their corresponding design/dev rating and reliabilities frozen as resource properties (depending on the project category). Registration will close after a fixed during unless there is the criterion to receive at least a fixed number of registrations.

1.2.3 Submission

Submission phase usually starts as soon as the registration phase does. Users with submitter role can log into the system and upload their submission into the system. Each submitter will be assigned a unique submission id upon their first upload, which does not change on subsequent uploads and is not reused if the user’s submission is deleted by managers. Each uploaded submission will be auto screened, and those with fatal errors will not proceed to the next phase. If a submitter uploads multiple versions, only the latest one is used (unless deleted). If the previous submission has uncommitted screening review, it will be replaced once the screening is committed. If a previous submission has already in review, it will not be possible to upload again. Submission phase will close after a fixed during unless there is a criterion to receive at least a fixed number of submissions that have passed screening.

1.2.4 Automated Screening

Automated Screening is kicked off as soon as submission is uploaded. Once the screening task is planted, a standalone screener will pick up the task and perform the screening. The screening result will be logged to corresponding tables and the screening status will change to indicate the result. Submission with fatal errors will not proceed further. All uploaded submissions will be auto screened. There is no independent phase for automated screening.

1.2.5 Manual Screening

Manual screening happens in the screening phase, which usually starts as soon as the submission phase. Submission without fatal errors in automated screening will be manual screened. A primary screener will be able to produce screening for each submission, while a submitter role is associated on a per submission basis. Once screening is committed, the score will be calculated. Notice that if a newer upload is found at this point, the submission should be associated with that upload and it will go through automated screening and manual screening again. Screening phase will end after the submission phase with all screening deliverables performed.

1.2.6 Review

Review phase will start as soon as screening phase is over. Each reviewer will produce a review for each submission, and test case reviewers will upload their test cases. Once a review is committed, the score will be calculated. Review phase will end once all deliverables are performed.

1.2.7 Appeal

Appeal can be added to each review item. Only one appeal can be added to each item and added appeals can not be modified. Appeal phase should end after a fixed period after the review phase ends (when the last review is committed). As soon as appeal is added, the reviewers can respond to the appeal responses. A flag should indicate whether submitters can see responses during the appeal phase, which should be configurable per project type.

1.2.8 Appeal Response

Appeal response should be added to each of the appeals by the original reviewer. One response will correspond to each appeal. Added responses can be edited. Score will be updated when each of the appeal is resolved. Appeal Response phase will end as soon as all appeals are resolved.

1.2.9 Aggregation

Aggregation starts after each of the appeals are resolved and there is a winner. Comments from the review scorecards will be copied into the aggregation worksheet. Aggregation will end once the worksheet is committed.

1.2.10 Aggregation Review

Aggregation starts after the aggregation phase. The reviewers that are different from the aggregator will need to approve the aggregation review as well as the winning submitter. If aggregation is not approved, additional aggregation/review phases will be inserted into the timeline.

1.2.11 Final Fix

Final Fix will start as soon as aggregation review is approved. The winning submitter should upload a final fix, as well as adding comments to the final review scorecard during the phase. The phase ends once the upload is in.

1.2.12 Final Review

All the accepted comments will be copied onto the final review worksheet. Final reviewer can only mark the final fix as approved if all the required comments are fixed. If final review is not approved, additional final fix/review phases will be inserted into the timeline.

1.2.13 Approval

Approval can be inserted at various locations. Approval phase will only end if approval scorecards all have passing scores.

1.2.14 Project Conclusion

Project can be set to inactive at any point, effectively removing it from all user's project list except the managers. Project that fails the process will not be changed to a failed status automatically. Managers need to perform it explicitly from the edit project page.

1.3 Component Requirements

1.3.1 TopCoder Software Components

1.3.1.1 New Components

- Auto Pilot
  - Auto Pilot automates the project phase execution based on API defined by Phase Management. The phase change can be triggered by configurable interval or application initiated events.
- Cached Web Element Tag
  - Cached Web Element Tag defines a way to decouple element rendering from the model layer. Data needed to render elements can be retrieved, cached and rendered in a pluggable mechanism.
- Deliverable Management
  - Deliverable Management defines the API to manage uploads and submissions. Besides it has a way to search deliverables and uses plug-ins to check if deliverables are met.
- Phase Management
  - Phase Management provides a persistence layer for the Project Phases component. Phase status can also be changed with pluggable extra logic.
- Project Management
  - Project Management defines the API to manage project related information.
- Project Phase Template
  - Project Phase Template provides template oriented mechanism to generate project phases based on project start date. The templates will be persisted as XML.
- Resource Management
  - Resource Management defines the API to manage resource related information.
- Review Data Structure (custom)
  - Review defines the data structures of a review..
- Review Management (custom)
  - Review Management defines the API to manage reviews. Comments can be added to review and item level. The way to evaluate a review is handled externally.
- Review Score Aggregator

  - Review Score Aggregator provides pluggable mechanism to aggregate scores from individual reviewers into the final score, as well as the rule to assign placements and break ties.
- Review Score Calculator
  - Review Score Calculator calculates the score based on the review and its corresponding scorecard. The mechanism to evaluate question answers is pluggable based on question type.
- Scorecard Data Structure (custom)
  - Scorecard defines the data structures of a scorecard.
- Scorecard Management (custom)
  - Scorecard Management defines the API to manage scorecards.
- Online Review Ajax Support (custom)
  - Online Review Ajax Support provides the servlet support for the AJAX features involved in the application.
- Online Review Deliverables (custom)
  - Online Review Deliverables provides the Deliverable Management plug-ins specific to the Online Review application.
- Online Review Login (custom)
  - Online Review Login provides the login and logout controller as Struts actions.
- Online Review Phases (custom)
  - Online Review Phases provides the Phase Management plug-ins specific to the Online Review application.
- User Project Data Store (custom)
  - User Project Data Store provides the API to access the TopCoder member and project information.

1.3.1.2 Updated Components

- Auto Screening Management 2.0 (custom)
  - Automated Screening Management 2.0 is the integration with the new Online Review application. It provides API to kick off the screening as well as to pull screening results.
- Auto Screening Tool 2.0 (custom)
  - Automated Screening Tool 2.0 is the integration with the new Online Review application. It comprises of standalone screeners.
- Database Abstraction 1.1
  - Database Abstraction 1.1 builds automatic type conversion ability into the previous component in order to mimic a JDBC result set. It is a dependency for the Search Builder.
- File Upload 2.0
  - File Upload 2.0 is integrated File System Server in order to enable transparent file uploading and sharing in a distributed environment.
- Project Phases 2.0
  - Project Phases 2.0 enhances the previous component by defining more attributes and more flexible dependency specification.
- Search Builder 1.3
  - Search Builder 1.3 fixes the previous version to release connections and use prepared statement for database access.

1.3.1.3 Existing Components

- Base Exception
  - Base Exception provides error chaining ability for pre 1.4 JDK implementations.
- Calendar Tag

  - Calendar Tag provides the UI support for pop-up calendars. Dates can be selected from the pop-up in a user friendly manner.
- Chart Data Structure
  - Chart Data Structure defines the data structure to hold chart figures in order to be rendered.
- Chart Rendering Engine
  - Chart Rendering Engine is the framework to accommodate chart renderers. The target of the rendering is flexible.
- Command Line Utility
  - Command Line Utility is used to identify the switches from the command line input.
- Compression Utility
  - Compression Utility is used to exact content from uploaded submissions.
- Configuration Manager
  - Configuration Manager manages configuration settings for all the components.
- Data Paging Tag
  - Data Paging Tag provides the UI support for paginating and sorting arbitrary collection of data.
- Data Validation
  - Data Validation is used as a generic validation framework.
- DB Connection Factory
  - DB Connection Factory encapsulates the logic to create database connections.
- Directory Validation
  - Directory Validation can validate a defined directory hierarchy on the file system.
- Document Generator
  - Document Generator makes use of templates to generate content with dynamic parameters. It is used to generate notification emails.
- Email Engine
  - Email Engine is the façade to Java Mail to send email notifications.
- Executable Wrapper
  - Executable Wrapper adapts the Java command execution API for better usability.
- File Class
  - File Class is a simple extension to the Java File class and provides the ability to recursively delete a folder.
- File System Server
  - File System Server is used as a centralized storage for all uploads so that they will be available to all instances in a cluster group. It is built upon IP Server.
- Gannt Chart
  - Gannt Chart is used to render the project phases information into SVG charts.
- Generic Event Manager
  - Generic Event Manager provides features similar to the .NET event.
- ID Generator
  - ID Generator is used to generate unique identifiers for newly created entities.
- IP Server
  - IP Server provides the framework and protocol exchange mechanism for machines with direct socket access.
- Job Scheduling
  - Job Scheduling provides ways to define and mange jobs. The jobs can be triggered with various criteria.
- Logging Wrapper

  - Logging Wrapper provides logging support for the application.
- Manager Numbers
  - Manager Numbers is used to identify the file type by examining specific features.
- Search Builder
  - Search Builder provides a way to create filters and execute search against multiple targets, including database and LDAP provider. It is used to provide search support for the management components.
- Security Manager
  - Security Manager is used to access the TopCoder member login process. It is provided as a remote EJB API.
- Simple Cache
  - Simple Cache provides entity caching with flexible strategies. Entities can be released by system if the server runs low on memory.
- Type Safe Enum
  - Type Safe Enum provides enumeration support.
- Weighted Calculator
  - Weighted Calculator calculates scores with complex, hierarchical structure. It is used to calculate review scores.
- Workdays
  - Workdays provides the ability to handle work and non-work days. It is a dependency for Project Phases.
- XMI Parser
  - XMI Parser provides the ability to parse XMI documents.

1.3.2 Third Party Components

Application uses the Struts framework to implement the MVC paradigm.

1.4 Application Management

1.4.1 Authentication

Authentication will be performed against the Security Manager component to access the TopCoder Login EJB. This will be handled in the Online Review Login component.

1.4.2 Authorization

Authorization will be performed based on the roles retrieved from Resource Manager. The complexity of the authorization structure does not fit into the Authorization component.

The following table details the various roles in the application. Please refer to the permission matrix in the requirement specification for strict details.

Role	Phase	Description
Submitter	N/A	Submitter will be associated as a project resource when they are registered (through another application). They will be able to upload submission, view their own screening and review after they are committed, place appeals, approve aggregation and upload final fixes. If a submitter does not have an active submission at some point (deleted, failed screening, failed review, completed without winning) they are not required to fulfill the responsibilities.
Primary Screener	Screening	Primary Screener will be associated with the screening phase when manual screening is required. They will be able to perform screening during the screening phase for all submissions.
Screener	Screening	Screener will be associated with the screening phase on a per submission basis when manual screening is required. Screeners

		are assigned only when Primary Screener is not used. They will be able to perform screening during the screening phase for their assigned submission.
Reviewer	Review	Reviewer will be associated with the review phase. They will be able to perform review during the review phase and resolve appeals in the following appeal response phase, perform aggregation review in the following aggregation review phase.
Accuracy/ Failure/ Stress Reviewer	Review	Test Case Reviewer will have the same permission and responsibilities as normal reviewers. In addition they are required to upload their test cases during the review phase.
Aggregator	Aggregation	Aggregator will be associated with the aggregation phase. They will be able to perform aggregation during the aggregation phase.
Final Reviewer	Final Review	Final Reviewer will be associated with the final review phase. They will be able to perform final review during the final review phase.
Approver	Approval	Approver will be associated with the approval phase. They will be able to perform approval during the approval phase.
Designer	N/A	Designer will be associated as a project resource. It does not have much difference from the public view. The purpose is for the designer to view private projects and to receive payments.
Observer	N/A	Observer will be associated as a project resource. They will be able to have read only view to more details than the public view. Customers will be assigned to this role so that they can monitor the production process.
Manager	N/A	Manager can either be assigned on a project basis or as on the global level. They will have permission to view and edit anything, including creating new projects.

1.4.3 Threading

All the components should not cache connection resources. It will be acceptable to instantiate separate manager instances to serve separate requests. The assemblies should choose using separate instances over using one instance with locking in order to maximize throughput.

Use of session should be thread-safe and should not be coupled with request. User must be allowed to work with multiple browser windows simultaneously.

Auto pilot should only be running on a single server if the application is distributed.

In case multiple screeners are set up, they should work in a thread safe manner and should not pick up or overwrite each other's submissions.

1.4.4 Configuration

Configuration will be stored in local XML files. The application will be deployed to a J2EE container, and all configuration files must be packaged in the WAR distribution.

1.4.4.1 Application Configuration

The application will use a configurable attribute name to store the external identifier of the user currently logged in, otherwise session usage should be minimized.

The session timeout value will be configurable from the J2EE container's configuration.

1.4.4.2 Component Configuration

This section will be completed as components are finalized.

1.4.5 Logging

All incoming requests, including AJAX requests, should be logged. Logging Wrapper should be used to perform logging, with the log4j logger as the backend implementation. Logging must comprise user ID, timestamp, action taken, and entity IDs effected.

Additionally, exceptions must be logged as mentioned below.

1.4.6 Database Transactions

Database operations are generally transactioned at the component level. If an action needs to perform updates through multiple component calls, it should be transactioned at the action level with JTA.

1.4.7 Exception Handling Overview

Users should not receive any significant error detail. They will simply get a generic error message.

All exceptions should be logged with as much detail as possible, including stack trace and nested exceptions.

1.5 Deployment Constraints

This application will not manage its own users. The application will access the TopCoder member and project database for necessary information, including coder rating, reliability, email, project forum, etc. Application will also access the Security Manager EJB to execute login requests.

The application will be deployed to a JBoss 4.0.2 application server with Informix 10.00 UC3R1 as database. It is assumed that the servers will reside on separate physical machines. A File System Server should be set up to share the uploaded files. Auto pilot and automated screener will run as a utility. The application needs to be distributable, though the first release will run as a single instance.

1.6 Development Standards:

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

1.7 Interfaces Classes Overview

Basically the classes in the assemblies will be the Struts actions and potentially action forms. The actions will be addressed in the Prototype Conversion section.

1.8 Changes to Existing System

This is a new system.

2. User Interface

User interface is relatively easy. All modifications will be performed on a single page, with the exception that sometimes a second page is prompted for confirmation. Data Paging Tag will be used to render the project, scorecard listings so that they are sortable by category.

2.1 Component Interface

No components interact directly with the user.

2.2 Prototype Conversion

This section is detailed according to the actions that will be produced in the assemblies.

Action	Input (parameters)	Output	Description
Login Actions			
Login POST	login.jsp user/pwd	List Projects	The action will use the username and password from the http parameter and invoke the Security Manager login EJB to authenticate the user. The user id will be retrieved and stored in session. Once authentication is successful the user will be forwarded to the List Projects action with scope=my to display the my open projects page.
Logout GET	any page	login.jsp	The action can be invoke anywhere from the navigation in the header. The user attribute will be removed from session.
Scorecard Actions			
New Scorecard GET	listScorecards.jsp	editScorecard.jsp	The action will create a new scorecard and attach it to the request so that an empty scorecard will be displayed. The dropdowns should be loaded.
Edit Scorecard GET	listScorecards.jsp sid	editScorecard.jsp	The action will load the given scorecard. It must verify if the scorecard is in use (and thus not permitted to edit).
Copy Scorecard GET	viewScorecard.jsp sid	editScorecard.jsp	The action will load the given scorecard and copy it to a new scorecard. The major version should be incremented.
Save Scorecard POST	editScorecard.jsp scorecard data	List Scorecards	The scorecard data will be validated and saved. It should either create the scorecard or update the existing scorecard.
Delete Scorecard POST	confirmDeleteScorecard.jsp sid	List Scorecards	The action will delete the scorecard with the given identifier by changing it to the deleted status.
List Scorecards GET	Various locations scope = “any” / “active” / “inactive”	listScorecards.jsp	The action will search the given scorecards and attach the list to the request. Scorecards will be grouped by category.
View Scorecard GET	listScorecards.jsp sid	viewScorecard.jsp	The action will load a particular scorecard for read only purpose. It should verify that the scorecard is in use (otherwise the mutable version should be displayed).
Project Actions			
New Project	listProjects.jsp	editProject.jsp	The action will prepare a new project and attached it to request. The

GET			dropdowns should be loaded.
Edit Project GET	viewProjectDetails.jsp pid	editProject.jsp	The action will load all the project information in order to edit.
Save Project POST	editProject.jsp project data	View Project Details	The action will validate the information entered and save the project. It should either create the project or update the existing project.
List Projects GET	Various locations scope = “my” / “all” / “inactive”	listProjects.jsp	The action will search the projects by the given score. It should also verify whether the user has permission to access the scope. All projects will be displayed with grouping.
Project Details Actions			
View Project Details GET	listProjects.jsp pid	viewProjectDetails.jsp	This is one of the most complicated pages in the application. It should load all the tabs according to the project phase. If performance becomes an issue, the tabs can also be loaded with the AJAX fashion.
Contact Manager POST	contactManager.jsp pid/email	View Project Details	The action will find the managers for the project and send out the email.
Upload Submission POST	uploadSubmission.jsp pid/file	View Project Details	The action will upload the submission and store it in the file system server. Database entries should be taken care of as to keep a single submission row for each submitter. The upload should be sent to create the task in automated screening.
Download Submission GET	viewProjectDetails.jsp uid	File download	The action will download the submission from the file system server and send to user as a download.
Upload Final Fix POST	uploadFinalFix.jsp pid/file	View Project Details	The action will upload a final fix into the file system server. The entry should be added to database.
Download Final Fix GET	viewProjectDetails.jsp uid	File download	The action will download the final fix from the file system server and send to user as a download.
Upload Test Case POST	uploadTestCase.jsp pid/file	View Project Details	The action will upload a test case into the file system server. The entry should be added to database. If a test case already exists for the resource, the old test case should be updated to deleted and email notification should be sent.
Download	viewProjectDetail	File download	The action will download the test case

Test Case GET	s.jsp uid		from the file system server and send to user as a download.
Download Document GET	Various scorecard pages uid	File download	The action will download a document from the file system server and send to user as a download.
Delete Submission POST	confirmDeleteSubmission.jsp uid	View Project Details	The action will delete a submission. Care should be executed to check if the upload is the latest valid submission and handle the update of the submission table as well.
View Auto Screening GET	viewProjectDetails.jsp uid	viewAutoScreening.jsp	The action will pull automated screening result from the database and display to the user.
Project Review Actions			
Create Screening GET	viewProjectDetails.jsp sid	editReview.jsp	The action will create a new screening review in database and present a modifiable view to the user.
Edit Screening GET	viewProjectDetails.jsp rid	editReview.jsp	The action will load an uncommitted screening review and present a modifiable view to the user.
Save Screening POST	editReview.jsp review data	previewReview.jsp or View Project Details	The action will save the screening review.
View Screening GET	viewProjectDetails.jsp rid	viewReview.jsp	The action will load a committed screening review and present a read only view to the user.
Create Review GET	viewProjectDetails.jsp sid	editReview.jsp	The action will create a new review in database and present a modifiable view to the user.
Edit Review GET	viewProjectDetails.jsp rid	editReview.jsp	The action will load an uncommitted review and present a modifiable view to the user.
Save Review POST	editReview.jsp review data	viewReview.jsp or View Project Details	The action will save the review.
View Review GET	viewProjectDetails.jsp rid	viewReview.jsp	The action will load a committed review and present a read only view to the user.
Create Aggregation GET	viewProjectDetails.jsp sid	editAggregation.jsp	The action will create a new aggregation review in database and present a modifiable view to the user.
Edit	viewProjectDetail	editAggregation.js	The action will load an uncommitted

Aggregation GET	s.jsp rid	p	aggregation and present a modifiable view to the user.
Save Aggregation POST	editAggregation.jsp review data	viewAggregation.jsp or View Project Details	The action will save the aggregation.
View Aggregation GET	viewProjectDetails.jsp rid	viewAggregation.jsp	The action will load a committed aggregation and present a read only view to the user.
Edit Aggregation Review GET	viewProjectDetails.jsp rid	editAggregationReview.jsp	The action will load a committed aggregation and display a modifiable aggregation review to the user.
Save Aggregation Review POST	editAggregationReview.jsp review data	viewAggregationReview.jsp or View Project Details	The action will save the aggregation review.
View Aggregation Review GET	viewProjectDetails.jsp rid	viewAggregationReview.jsp	The action will load a committed aggregation review and display a read only view to the user.
Create Final Review GET	viewProjectDetails.jsp sid	editFinalReview.jsp	The action will create a new final review in database and present a modifiable view to the user.
Edit Final Review GET	viewProjectDetails.jsp rid	editFinalReview.jsp	The action will load an uncommitted final review and present a modifiable view to the user.
Save Final Review POST	editFinalReview.jsp review data	viewFinalReview.jsp or View Project Details	The action will save the final review.
View Final Review GET	viewProjectDetails.jsp rid	viewFinalReview.jsp	The action will load a committed final review and display a read only view to the user.
Create Approval GET	viewProjectDetails.jsp sid	editReview.jsp	The action will create a new approval review in database and present a modifiable view to the user.
Edit Approval GET	viewProjectDetails.jsp rid	editReview.jsp	The action will load an uncommitted final review and display a modifiable view to the user.
Save Approval POST	editReview.jsp review data	previewReview.jsp / View Project Details	The action will save the approval review.

View Approval GET	viewProjectDetails.jsp rid	viewReview.jsp	The action will load a committed approval review and display a read only view to the user.
View Composite Scorecard GET	viewProjectDetails.jsp sid	viewCompositeScorecard.jsp	The action will load all reviews for a submission and combine them into a composite scorecard.

2.3 AJAX Interactions

The application uses AJAX to enhance user experience. The following matrix includes the information for the interactions. The server side support will be developed in the AJAX Support component, and the client side javascripts will be developed as part of the assemblies.

Name	Location (Parameters)	Description
Set Scorecard Status	listScorecards.jsp sid/status ack or error	User with appropriate role will be able to check or uncheck the checkbox. Only scorecards not in use can be set to inactive, otherwise the checkbox will be disabled.
Load Timeline Template	createProject.jsp template/date(optional) timeline data	User with appropriate row will be able to select a template from the dropdown and click the load template button.
Set Timeline Notification	viewProjectDetails.jsp pid/status ack or error	User with appropriate role will be able to check or uncheck the checkbox.
Place Appeal	editAppeal.jsp rid/iid/text ack or error	User with appropriate role will be able to click the appeal button on each item and bring up a text area input. Once the user enters the text and submits, the appeal is posted to the server. The appeal becomes a read only comment on the item and the item is flagged unresolved. The appeal button disappears.
Resolve Appeal	editAppealResponse.jsp rid/iid//success/ response/text score or error	User with appropriate role will be able to submit appeal response for each unresolved appeal. When an appeal is resolved, the text and response should be submitted to the server. The text becomes a read only comment on the item and the status is flagged as resolved. The modified response will display. Upon receiving the response the overall score will be updated. Responses to resolved appeals can be edited and will follow the same logic as unresolved appeals.

3. Included Documentation

3.1 Architecture Documentation

- Component Diagram
- Component Interfaces Class Diagrams
- Sequence Diagrams
- Application Design Specification
- Component Requirement Specifications for all updated and new components
- Physical Entity Relationship Diagram

- SQL Scripts