[TopCoder]

Struts Actions Requirements Specification

1. Scope

1.1 Overview

The TopCoder registration process to become a member of TopCoder is going to be redesigned. The main goal is to allow for a minimal set of required info that the user needs to enter to sign up (handle, email, and password).

The existing TC web registration process will ask a new user to enter a lot of data when he/she tries to register to TC web-site. The registration process takes a lot of time, so many users can simply break the registration process and leave non-registered

Thus there is a need of an easy to use and user friendly web-application for supporting registration on the new users and management of registered user profiles at TC web-site.

The main goal of this application is to simplify registration of the new users on TC web-site by minimizing the count of mandatory data fields for new account registration, and to improve usability of user profile management for the registered users. Any additional profile information will be requested from the user by the system as it is needed. For example, if a user registers to compete in an assembly contest the site would prompt them for any required info that they have not yet entered.

This component is responsible for implementing an authorization interceptor; login, logout and password recovery struts actions.

1.2 Logic Requirements

1.2.1 AuthInterceptor

This custom struts interceptor is mainly used to authorize the user.

It should implement the logic defined in the process method of src\main\com\topcoder\web\common\BaseServlet.java. Except that it now won't call the RequestProcessor, as struts action will be used instead to process the request.

It will mainly do the following steps:

- Get the user (guest or authenticated) from session or cookie.
- Check if user has permission to access the request URI corresponding to the struts action.
- Proceed if user has the permission, otherwise, redirect to an authorization error page.

Note that the authorization error page will be configured globally in struts configuration file, and the created BasicAuthentication will be stored in session if it's not present.

1.2.2 LoginAction

This struts action should implement the login operation defined in ARS 2.2.1.1 – 2.2.1.6.

The existing functionality can be found in src\main\com\topcoder\web\reg\controller\request\Login.java.

It will mainly do the following steps:

- Use Login EJB to log the user in.
- Ensure user's status is active, and use BasicAuthentication to set remember-me cookies
- If user logs in for the first time (e.g. after activating the account), redirect to a specific result; if user logs in with direct login-url, redirect to a specific result; and if user is redirected to the

login page, redirect to the original page after logged-in.

Note that if AuditDAO.hasOperation(handle, “login”) returns false, then it’s the first-time login; the redirect URI parameter name should be injected; and the created BasicAuthentication will be stored in session.

1.2.3 LogoutAction

This struts action should implement the functionality defined in ARS 2.3.1.1 – 2.3.1.2.

The existing functionality can be found in src\main\com\topcoder\web\reg\controller\request\Logout.java.

It will mainly log user out using BasicAuthentication and then invalidate the session.

1.2.4 RecoverPasswordAction

This struts action should implement the functionality defined in ARS 2.4.1.1, 2.4.1.3 – 2.4.1.7.

The existing functionality can be found in the src\main\com\topcoder\web\tc\controller\request\authentication\RecoverEmail.java.

It will mainly do the following steps:

- Get the user from UserDAO and ensure the email matches.
- Create a PasswordRecovery entry into the database.
- Send the email containing the reset password link.

1.2.5 ResendPasswordRecoverEmailAction

This struts action should implement the functionality defined in ARS 2.4.1.7 to resend the recovery password email when user clicks the “Click here” link.

It will resend the email mentioned in last section.

1.2.6 ResetPasswordAction

This struts action should implement the functionality defined in ARS 2.4.1.9 – 2.4.1.12.

The existing functionality can be found in the src\main\com\topcoder\web\tc\controller\request\authentication\ResetPassword.java.

It will mainly do the following steps:

- Get PasswordRecover from PasswordRecoverDAO and ensure the PasswordRecover entry for the user is valid.
- Update the PasswordRecover entry as used.
- Generate a new password for user using Random String Generator.
- Automatically log user in (by creating a BasicAuthentication for the user and store it into session).
- Reset user’s password with the newly generated one and display it to user.

1.2.7 Guideline

Designer is not responsible for the front-end JSP pages, but the interactions between the JSP pages and struts actions must be clearly defined.

Designer is allowed to refactor the predefined struts actions in the module architecture as long as the requirements are addressed properly.

Designer is also expected to provide the struts-config.xml and input validation XML for the struts actions.

The implementation in the provided existing application code might be different from the ARS; in this case, the ARS takes the priority.

The guidelines provided in section 1.3 of the architecture's ADS should be followed by this design.

1.3 Required Algorithms

None

1.4 Example of the Software Usage

This component provides the struts actions for the TC registration process.

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 1.6

2.1.4 Package

com.topcoder.web.reg.interceptors
com.topcoder.web.reg.actions.basic

3. Software Requirements

3.1 Administration Requirements

3.1.1 What elements of the application need to be configurable?

- The back-end DAO and EJB should be injected
- Session key for the BasicAuthentication object
- Email subject, email body template, email from address

3.2 Technical Constraints

3.2.1 Are there particular frameworks or standards that are required?

Struts

3.2.2 TopCoder Software Component Dependencies:

Logging Wrapper 2.0
Base Exception 2.0
Random String Generator 1.0

Document Generator 1.1
Email Engine 3.2

Custom:
User Profile and Audit Back End 1.0

3.2.3 Third Party Component, Library, or Product Dependencies:

- Spring 2.5.6
- Hibernate 3.6

3.2.4 QA Environment:

- J2EE 1.5
- JAVA 1.6.0_04
- Apache 2.2.4
- jira 4.1.2
- wiki 2.7
- Jive Professional 4.2.5
- Red Hat 4.1.2-46
- Informix 11.5
- JBoss-4.0.4.GA
- HTML/CSS
- IE6 - latest version
- FireFox - latest version
- Safari - latest version
- Chrome - latest version
- JavaScript (jQuery 1.4.4)
- HttpUnit
- CVS

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.