[TopCoder]

Generic Member Photo Management Servlets v2.0 Requirements Specification

1.Scope

1.1Overview

This component will provide the servlets which are used for member photo management: one servlet is used to upload member photo and other to remove member photos. This component will use the Member Photo Manager component as a backend.

1.1.1Version

2.0

1.2Logic Requirements

1.2.1Member Photo Upload Servlet

This component will provide the servlet which will allow web site members to upload their photos. This servlet will handle the submission of the form with such fields:

- member id
- photo image file
- action value indicating the image should be previewed, cropped or committed.

1.2.1.1Validate submitted form

Such validations should be enforced on submitted form:

- member id should match the member id stored in session
- uploaded image file should be in either JPEG or PNG or GIF format
- the size of photo image file <= configured maximum file size (in bytes)

1.2.1.2Report validation errors

If any form validation errors are present, they should be stored in the request and request should be forwarded to the configured URL.

1.2.1.3Store image for preview

If it's a "preview" action, which is the initial action, the uploaded image should be stored in the configurable directory. The path to the resulting file should be stored in the request, and user will preview the image on a page.

1.2.1.4Crop and resize uploaded image

On the preview page, user will be able to draw a frame to crop the image (using Jcrop) with certain aspect ratio. The servlet should handle this "crop" action to crop the image using the parameters generated by Jcrop, and then resize the cropped image to match the configured target size. The resulting image should be stored in the configurable directory, and it should be in JPEG format. The path to the resulting image file should be stored in the request, and user will view the resulting image on a page.

Note that the resulting image file's name should be configurable and it should contain a placeholder for the member's handle.

1.2.1.5Store image file permanently

On the page showing the cropped and resized image, user is able to submit the form to commit the action. So in this final "commit" action, the cropped and resized image should be stored permanently in the configurable directory. The information about uploaded member photo should be persisted using Member Photo Manager component.

1.2.1.6Forward or redirect to result page

If the photo was not committed, the request should be forwarded to the configured URL. Otherwise the request should be redirected to the configured URL. As it is assumed that redirect will be to member profile page, the configured URL should include a placeholder for member id, which will be replaced by actual member id when doing redirect.

1.2.2Member Photo Removal Servlet

This component will provide the servlet which will allow member and web site administrator to remove particular member photo.
This servlet will handle the submission of the form with such fields:
• member id
• removal reason explanation

1.2.2.1Validate submitted form

Such validations should be enforced on submitted form:
• there should be a photo associated with the given member id
• session should contain an attribute which indicates that logged in user is a member or administrator

1.2.2.2Report validation errors

If any form validation errors are present, they should be stored in the request and request should be forwarded to the configured URL.

1.2.2.3Remove image

The reference to image should be removed from DB using Member Photo Manager component. The corresponding image file should be moved into a configurable directory.

1.2.2.4Send notification

The e-mail notification will be send to member to notify of his/her photo deletion. The e-mail content will be generated using Document Generator component.
Such information should be substituted in the template:
• the reason for photo removal
• the name of member
• the handle of member

• the id of member

The email notification functionality can be turned on or off.

1.2.2.5Forward to result page

The request should be forwarded to the configured URL.

1.2.3Member Photo List Servlet

This component will provide the servlet which will allow web site administrator to view members' photos.

This servlet will handle request containing the following parameters:
pageNo – the page number, which starts 1. Optional, default to 1.
pageSize – the page size, which should be positive. Optional, default to 10.

1.2.3.1Report validation errors

If pageNo or pageSize is invalid, or session doesn't contain an attribute indicating it's an administrator, it should store errors into request and forward to the configured URL.

1.2.3.2Retrieve Members' Photos

Invoke MemberPhotoManager.getMemberPhotos method to get the photos for display.

1.2.3.3Forward to result page

The request should be forwarded to the configured URL.

1.2.4Member Information used in Servlets

To obtain the information about member, given a member id, the pluggable handler should be used. Designer only needs to define the interface.

1.2.5Thread Safety

The servlets need to be thread-safe when serving the user requests. The setters for initialization purpose don't need to be thread-safe.

1.2.6Configuration

The component will use the Spring to inject the configured properties.

1.2.7Logging

The exception should be logged with ERROR level. The user action should be logged with INFO level.

1.2.8JSP Pages

Designers need to clearly describe what contents are required in the JSP pages and the interactions between the JSP pages and the corresponding servlets.

Developers are expected to implement these JSP pages in order to properly test this component.

1.2.9Version 1.0

The version 2.0 is upgraded from version 1.0, and it doesn't need to be back-compatible with version 1.0.

Designer can either provide a completely new design or reuse the version 1.0 as much as possible.

1.3Required Algorithms

None

1.4Example of the Software Usage

Members are allowed to submit photos to the TopCoder site to be displayed. Currently this process is handled by hand. This component will be used in automated photo upload solution to be deployed at TopCoder site.

1.5Future Component Direction

More generic image management solution

2.Interface Requirements

2.1.1Graphical User Interface Requirements

None

2.1.2External Interfaces

Refer to the provided architecture TCUML.

2.1.3Environment Requirements

Development language: Java 1.5
Compile Target: Java 1.5

2.1.4Package

com.topcoder.web.memberphoto.servlet

3.Software Requirements

3.1Administration Requirements

3.1.1What elements of the application need to be configurable?

forward and redirect URLs
all HTTP request parameter names

all session and request attribute names
directories to store files in
target size of the resulting image
maximum uploaded file size (in bytes)

3.2Technical Constraints

3.2.1Are there particular frameworks or standards that are required?

Spring Framework
JSP & JSP Servlet

3.2.2TopCoder Software Component Dependencies:

Base Exception 2.0
Logging Wrapper 2.0
Email Engine 3.2
Document Generator 2.0
Magic Number 1.1
File Upload 2.2

Member Photo Manager 2.0

**Please review the TopCoder Software component catalog for existing components that can be used in the design.

3.2.3Third Party Component, Library, or Product Dependencies:

Spring 3.0.5
Jcrop 0.9.8
JQuery 1.4.4

3.2.4QA Environment:

Java 1.5
JBoss 4.0.2
Informix 11.0
JSP & Servlet
Windows/Linux/Mac

3.3Design 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.

3.4Required Documentation

3.4.1Design Documentation

Use-Case Diagram
Class Diagram

Sequence Diagram

Component Specification

3.4.2Help / User Documentation

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