# Ameya Joshi ClubUMLSpring2014

# Validation Team Sprint 2

I have been working on the **display.jsp** and Sahil has been working on **managePolicy.jsp** where .ecore files can be uploaded.

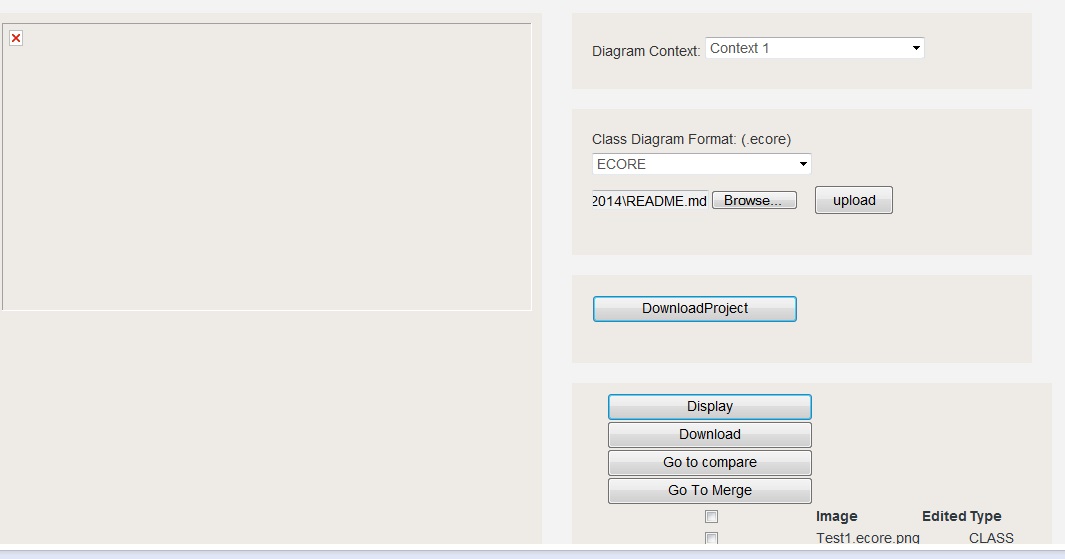
Main issues with the pages were as follows:

* The display.jsp allowed the user to upload any type of file without throwing an alert to let the user know that he uploaded a wrong type of file.
* The managePolicy.jsp has features “add policy” and “update policy” in which the metrics are not validated.

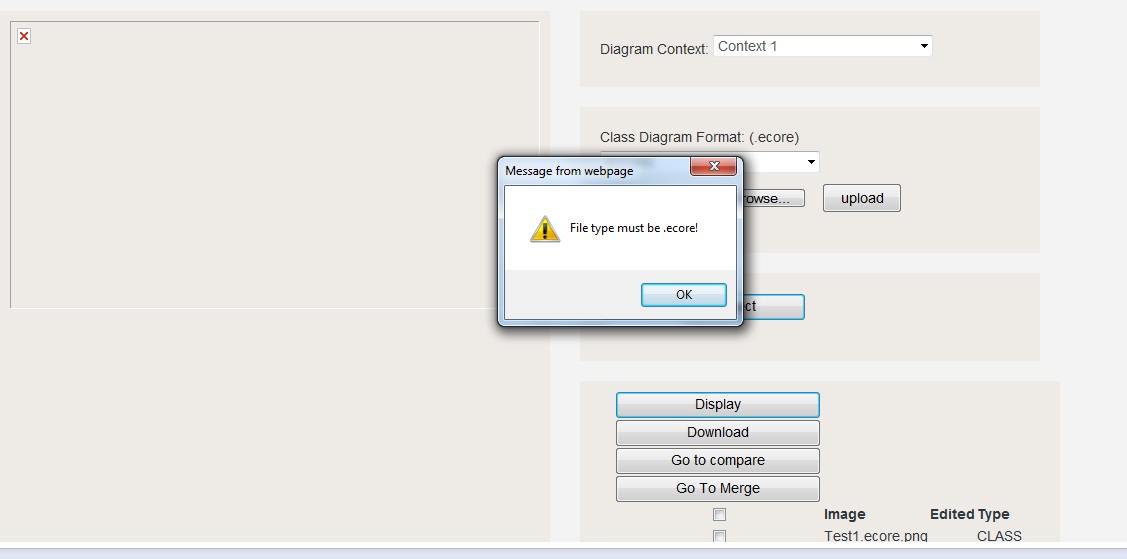
The following changes have been made:

* A check has been added in display.jsp to check if the file selected is of correct format. This is done using Regular Expressions. An alert (pop-up) is thrown when the select tries to upload a wrong kind of file.
* Checks have been incorporated to make sure that the entries in the fields of metrics are of desired type (integers). This too has been done using RegEx.
* Furthermore, the JavaScript from the index.jsp and register.jsp has been combined into one JavaScript file called register.js
* The errors displayed on the index.jsp inadvertently have been removed. We are no more helping people to break in! :)

**Screenshots:**



1. Wrong file type selected



1. Error message displayed on selecting wrong file type

**Code changes:**

***Register.js:***

// valid user flag. Used by Ajax call.

**var** validUser = **false**;

/\*\*

\* Validates all required fields.

\*

\* **@returns** true if all fields are valid, else false

\*/

**function** validateForm() {

**var** username = document.forms.registerForm.username.value;

**var** password = document.forms.registerForm.password.value;

**var** password2 = document.forms.registerForm.password2.value;

**var** email = document.forms.registerForm.email.value;

**if** (!validUser) {

alert("Require valid user!");

**return** **false**;

} **else** **if** (password == "") {

alert("Password can not be empty!");

**return** **false**;

} **else** **if** (document.forms.registerForm.password.value != document.forms.registerForm.password2.value) {

alert("Password does not match!");

**return** **false**;

} **else** **if** (!validateEmail(email)) {

alert("Require valid email!");

**return** **false**;

}

**return** **true**;

}

/\*\*

\* Check if user name already exist in user table.

\*

\* **@param** value is the user name to check

\*/

**function** checkUsername(value) {

**var** block = document.getElementById("username");

**var** patt1 = **new** RegExp("^[a-z][a-z0-9\_]\*.{3,12}$");

**if** (patt1.test(value))

{

ajax = createAjax();

ajax.onreadystatechange = **function**(){

**if** (ajax.readyState == 4 && ajax.status == 200)

{

**var** response = ajax.responseText;

block.innerHTML = response;

}

};

ajax.open("get", "ValidateServlet?username=" + value + "", **true**);

ajax.send(**null**);

validUser = **true**;

}

**else**

{

block.innerHTML = "<font color='red'>Username should consist only characters, numbers and underscore<br/> and should begin with a character only.</font>";

validUser = **false**;

}

}

/\*\*

\* Validate passwords match.

\*

\* Displays message for valid and invalid password.

\*/

**function** checkPassword(pass2) {

**var** block2 = document.getElementById("passText");

**var** password1 = document.forms.registerForm.password.value;

**var** password2 = document.forms.registerForm.password2.value;

**if** (password1 == "" || password2 == "")

{

block2.innerHTML = "<font color = 'red'>Password field(s) cannot be left blank.</font>";

}

**else** **if** (password1 != password2)

{

block2.innerHTML = "<font color = 'red'>Password does not match.</font>";

}

**else**

{

**var** passFlag = validatePass(pass2);

**if**(!passFlag)

{

block2.innerHTML = "<font color = 'red'>Password must be at least 4 characters, no more than 20 characters, and " +

"<br>must include either upper case letters, lower case letters, numeric digits, and special characters.</font>";

}

**else** **if**(passFlag)

{

block2.innerHTML = "<font color = 'green'>Password valid.</font>";

}

}

}

**function** validatePass(password) {

**var** pattern1 = **new** RegExp("^[A-Za-z0-9!@#$%^&\*()\_]{4,20}$");

**if** (password == "") {

**return** **false**;

} **else** **if** (pattern1.test(password)) {

**return** **true**;

} **else** {

**return** **false**;

}

}

/\*\*

\* Displays message for valid and invalid email.

\*

\* **@param** email to verify

\*

\* **@author** AmeyaCJoshi

\*/

**function** checkEmail(email) {

**var** block = document.getElementById("emailText");

**var** patt1 = **new** RegExp(".+@.+\.[com|net|org|biz|gov|edu]$");

**if**(email == "")

{

block.innerHTML = "<font color = 'red'>Email field cannot be left blank.</font>";

}

**else**

{

**var** emailFlag = patt1.test(email);

block.innerHTML = emailFlag;

**if** (emailFlag)

{

ajaxObj = createAjax();

ajaxObj.onreadystatechange = **function**() {

**if** (ajaxObj.readyState == 4 && ajaxObj.status == 200)

{

**var** response = ajaxObj.responseText;

block.innerHTML = response;

}

};

ajaxObj.open("get", "ValidateEmail?email=" + email + "", **true**);

ajaxObj.send(**null**);

}

**else**

{

block.innerHTML = "<font color = 'red'>Invalid email format.</font>";

}

}

}

/\*\*

\* Validates if email is in correct format

\*

\* **@param** email to verify

\* **@return** true if valid email, else false

\*/

/\*

function validateEmail(email) {

if (patt1.test(email)) {

return true;

} else {

return false;

}

}

\*/

/\*\*

\* Creates Ajax object

\*

\* **@returns** Ajax object

\*/

**function** createAjax() {

**var** xmlhttp;

**if** (window.XMLHttpRequest) {// code for IE7+, Firefox, Chrome, Opera, Safari

xmlhttp = **new** XMLHttpRequest();

} **else** {// code for IE6, IE5

xmlhttp = **new** ActiveXObject("Microsoft.XMLHTTP");

}

**return** xmlhttp;

}

/\*\*

\* (Have not seen used. Candidate for removal)

\*

\* **@returns**

\*/

**function** checkPromote() {

**if** (document.forms.promote1.comment.value == ""

|| document.forms.promote1.comment.value == **null**) {

alert("Comment cannot be empty!");

**return** **false**;

}

}

/\*\*

\* Validation for index page.

\* EDITED BY: AmeyaCJoshi

\*/

**function** checkUsername(username)

{

**if**(username == "") {

**return** **false**;

}

re = /^\w+$/;

**if**(!re.test(username)) {

**return** **false**;

}

**else**

{

**return** **true**;

}

}

**function** checkPassword(password)

{

**var** block = document.getElementById("password");

**if**(password == "") {

**return** **false**;

}

re = /[a-z]/;

**if**(!re.test(password)) {

**return** **false**;

}

re = /[A-Z]/;

**if**(!re.test(password)) {

**return** **false**;

}

re = /[0-9]/;

**if**(!re.test(password)) {

**return** **false**;

}

**else**

{

**return** **true**;

}

}

***display.jsp:***

<%--

Document : display

Created on : Oct 20, 2012, 2:25:33 PM

Author : pratham

Edited By : AmeyaCJoshi

--%>

<%@page contentType=*"text/html"* pageEncoding=*"UTF-8"*%>

<%@taglib prefix=*"c"* uri=*"http://java.sun.com/jsp/jstl/core"*%>

<script type=*"text/javascript"* src=*"js/display.js"*></script>

<script src=*"//ajax.googleapis.com/ajax/libs/jquery/1.8.3/jquery.min.js"*></script>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=UTF-8"*>

<title>Project Diagrams</title>

<link href=*"css/display.css"* rel=*"stylesheet"* type=*"text/css"* />

<link href=*"style.css"* rel=*"stylesheet"* type=*"text/css"* />

<!-- BOOTSTRAP -->

<link href=*"http://getbootstrap.com/2.3.2/assets/css/bootstrap.css"*

rel=*"stylesheet"*>

<link

href=*"http://twitter.github.io/bootstrap/assets/css/bootstrap-responsive.css"*

rel=*"stylesheet"*>

<style type=*"text/css"*>

*.diagramBox*{

float: *left*;

}

</style>

<script type=*"text/javascript"*>

**var** type = "";

$(document).ready(**function**() {

$("#downloadButton").click(**function**() {

type = **this**.id.toString();

});

$("#compareButton").click(**function**() {

type = **this**.id.toString();

});

$("#displayButton").click(**function**() {

type = **this**.id.toString();

});

// Merge function

$("#mergeButton").click(**function**() {

type = **this**.id.toString();

});

// resize image proportionally

$('img').each(**function**() {

**var** maxWidth = 530;

**var** maxHeight = 530;

**var** ratio = 0; // Used for aspect ratio

**var** width = $(**this**).width(); // Current image width

**var** height = $(**this**).height(); // Current image height

/\* Make height/width at least the max size to begin with \*/

// Check if the current width is smaller than max

**if**(width < maxWidth){

ratio = maxWidth / width; // get ratio for scaling image

height = height \* ratio; // Reset height to match scaled image

width = width \* ratio; // Reset width to match scaled image

}

// Check if current height is smaller than min

**if**(height < maxHeight){

ratio = maxHeight / height; // get ratio for scaling image

height = height \* ratio; // Reset height to match scaled image

width = width \* ratio; // Reset width to match scaled image

}

/\* Then reduce height/width to fit the max \*/

// Check if the current width is larger than the max

**if**(width > maxWidth){

ratio = maxWidth / width; // get ratio for scaling image

height = height \* ratio; // Reset height to match scaled image

width = width \* ratio; // Reset width to match scaled image

}

// Check if current height is larger than max

**if**(height > maxHeight){

ratio = maxHeight / height; // get ratio for scaling image

height = height \* ratio; // Reset height to match scaled image

width = width \* ratio; // Reset width to match scaled image

}

// Set CSS

$(**this**).css("height", height); // Set new height

$(**this**).css("width", width); // Scale width based on ratio

});

});

**function** checkError()

{

**var** block = document.getElementById("errorMsg");

**var** fileName = document.getElementById("file1");

**var** flag = **false**;

//var pattern = new RegExp ("/^.\*\.(ecore|ECORE)$/");

//flag = pattern.test(fileName);

**var** ext = "";

**var** a = fileName.value.split(".");

**if**( a.length === 1 || ( a[0] === "" && a.length === 2 ) ) {

ext = "";

}

ext = a.pop();

**if**(ext == "ecore" || ext == "ECORE")

flag = **true**;

**if**(!flag)

{

alert("File type must be .ecore!");

}

}

**function** checkFields() {

**if** (type == "downloadButton") {

**if** ($(".myCheckBox:checked").length == 1) {

**return** **true**;

}

alert("Please select 1 diagram for download");

**return** **false**;

}

**if** (type == "compareButton") {

**var** check = document.getElementsByName("check");

**var** checked = [];

**for**(**var** i=0; i<check.length; i++) {

**if**(check[i].checked) {

checked.push(check[i]);

}

}

// Check for 2 Ecore; XMI not supported++

**var** bothValid = **true**;

**if** (checked.length == 2) {

**if** (checked[0].id == "sequence" || checked[1].id == "sequence") {

bothValid = **false**;

}

**if** (checked[0].id == "class" || checked[1].id == "class") {

bothValid = **false**;

}

}

**if** (checked.length == 2 && bothValid) {

// DisplayDiagram looks for ID numbers in the checked.value fields

**return** **true**;

}

alert("Please select 2 Ecore diagrams to compare");

**return** **false**;

}

**if** (type == "displayButton") {

**if** ($(".myCheckBox:checked").length == 1) {

**return** **true**;

}

alert("Please select 1 diagram to display");

**return** **false**;

}

// merge function

**if** (type == "mergeButton") {

**var** req = document.getElementById("req");

**var** form = document.getElementById("requestForm");

**var** check = document.getElementsByName("check");

**var** checked = [];

**for**(**var** i=0; i<check.length; i++) {

**if**(check[i].checked) {

checked.push(check[i]);

}

}

// Check for 2 XMI class; Ecore or sequence not supported

**var** bothValid = **true**;

**if** (checked.length == 2) {

**if** (checked[0].id == "sequence" || checked[1].id == "sequence") {

bothValid = **false**;

}

**if** (checked[0].id == "Ecore" || checked[1].id == "Ecore") {

bothValid = **false**;

}

}

**if** (checked.length == 2 && bothValid) {

**var** reqO = {};

reqO.Request = "Refresh";

reqO.Diagram1 = checked[0].value;

reqO.Diagram2 = checked[1].value;

req.value = JSON.stringify(reqO);

form.submit();

**return** **false**;

}

alert("Please select 2 XMI Class diagrams to merge.");

**return** **false**;

}

// end

}

**function** displayClassDiagramFields(element) {

**var** option = element.options[element.selectedIndex].text;

**var** selectLabel = document.getElementById("DiagramSelectLabel");

**var** fileinput2 = document.getElementById("file2");

**var** fileinput3 = document.getElementById("file3");

**if** (option == "ECORE") {

fileinput2.style.display = "none";

fileinput3.style.display = "none";

selectLabel.innerHTML = "Class Diagram Format: (.ecore)";

} **else** **if** (option == "XMI") {

fileinput2.style.display = "block";

fileinput3.style.display = "block";

selectLabel.innerHTML = "Class Diagram Format: (.di, .notation, .uml)";

}

}

**function** toggleChecked(status) {

$(".myCheckBox").each( **function**() {

$(**this**).attr("checked",status);

})

}

</script>

</head>

<body style="background-color: *#F3F3F3*">

<div id=*"myHeader"*>

<h1 id=*"banner"*>Class Diagrams</h1>

</div>

<div id=*"mainContainer"*>

<div>

<h2><a href=*"Home"*>Click Here</a> to go back </h2>

</div>

<div id=*"diagramBox"*>

<c:if test=*"*${requestScope.diagramId1 != null}*"*>

<img src=*"*${requestScope.firstPath}*"*/>

</c:if>

<!--

<br><b>Comments : </b>

<c:if test="${requestScope.comments != null}">

<div id="commentBox">

<table id="myTable">

<c:forEach items="${requestScope.comments}" var="comment">

<tr>

<td><b><c:out value="${comment.userName}"></c:out> </b></td>

<td><c:out value=" ${comment.content}"></c:out></td>

</tr>

</c:forEach>

</table>

</div>

</c:if>

-->

</div>

<div id=*"rightContainer"*>

<div id=*"selectContextBox"*>

<span>Diagram Context:</span>

<select>

<option value=*"context1"*>Context 1</option>

<option value=*"context2"*>Context 2</option>

</select>

</div>

<div id=*"box"*>

<span id=*"DiagramSelectLabel"*>Class Diagram Format: (.ecore)</span>

<select onchange=*"displayClassDiagramFields(this)"*>

<option value=*"ecore"*>ECORE</option>

<option value=*"xmi"*>XMI</option>

</select>

<form action=*"UploadServlet"* method=*POST* enctype=*"multipart/form-data"*>

<input id=*"file1"* type=*"file"* name=*"file"* size=*"50"* />

<input id=*"file2"* type=*"file"* name=*"file2"* size=*"50"* style="display:*none*"/>

<input id=*"file3"* type=*"file"* name=*"file3"* size=*"50"* style="display:*none*"/>

<input type=*"hidden"* value=*"*${ProjectID }*"* name=*"ProjectID"*>

<input class=*"submit"* type=*"submit"* onmouseover="checkError();" value=*"upload"*>

<br><span id=*"errorMsg"*></span>

</form>

</div>

<div id=*"box2"*>

<form action=*"DisplayDiagram"* method=*"post"*>

<input type=*"submit"* id=*"downloadProjectButton"* value=*"DownloadProject"*

name=*"submit"* />

</form>

</div>

<div id=*"list"*>

<form action=*"DisplayDiagram"* method=*"post"*

onsubmit="return checkFields()">

<input type=*"submit"* id=*"displayButton"* value=*"Display"* name=*"submit"* />

<input type=*"submit"* id=*"downloadButton"* value=*"Download"* name=*"submit"* />

<input type=*"submit"* id=*"compareButton"* value=*"Go to compare"* name=*"submit"*

<c:if test="${type=='sequence'}">disabled</c:if>

<c:if test="${type=='class'}">disabled</c:if>/>

<!-- Merge -->

<input type=*"submit"* id=*"mergeButton"* value=*"Go To Merge"* name=*"submit"*

<c:if test="${type=='sequence'}">disabled</c:if>

<c:if test="${type=='Ecore'}">disabled</c:if>/>

<!-- End Merge -->

<table>

<tr>

<td>

<input type=*"checkbox"* name=*"checkall"*

value=*"10"* onclick="toggleChecked(this.checked)">

</td>

<td><b>Image</b></td>

<td><b>Edited</b></td>

<td><b>Type</b></td>

</tr>

<c:forEach items=*"*${requestScope.diagrams}*"* var=*"diagram"*>

<tr>

<td><input class=*"myCheckBox"* type=*"checkbox"* name=*"check"*

value=*"*${diagram.diagramId}*"* id=*"*${diagram.diagramType}*"*/></td>

<td>${diagram.diagramName}</td>

<td>${diagram.createdTime}</td>

<td>${diagram.diagramType}</td>

</tr>

</c:forEach>

</table>

</form>

</div>

</div>

</div>

<!-- Merge Form -->

<form id=*requestForm* action=*"ClassMergeComunicator"* method=*POST* style="display: *none*;" onsubmit="return checkFields()">

<input name=*request* id=*req* value=*""* />

</form>

<!-- End Merges -->

</body>

</html>