**IMAGE UPLOAD:**

When ADMIN adds new item, then he uploads image (page: WEB-INF/views/items/add, line 41). The controller com.codekarma.controller. ResourcesController saves the image in server (line 76). The rule of the path of the saved image is: */resources/itemsimages/itemcode\_itemname.png* . This image is later accessed in WEB-INF/views/items/item.jsp (line 20) and WEB-INF/views/items/requestadded.jsp (line 21). In these JSPs, the url of image is formed as:

<c:set var=*"base"* value=*"/*resource*/itemsimages/"*></c:set>

<spring:url value=*"*${base }${ item.imageName}*"* var=*"imagesource"*/>

Note 1: item.imageName = item.itemCode + *“\_”* + item.name + *“.png”*;

Note 2: while using spring:url, the url */resource/\*\** is mapped to location */resources/\*\**, the location where the image was saved. This configuration is provided in servlet-context.xml (<mvc:resources location=*"/resources/"* mapping=*"/resource/\*\*"*/>).

**SECURITY DESIGN:**

The field ENABLED in the username query of Spring Security is hard coded to be ‘true’, as all of the members of the system are supposed to be active. We don’t have any field that corresponds to ENABLED in database (member table).

That is, in security-context.xml,

users-by-username-query*="select username,password, 'true' 'enabled' from member where username=?"*

Two types of authentication:

URL Based:

For viewing/editing/deleting/adding members:

<security:intercept-url pattern=*"/member/\*"* access=*"hasRole('ADMIN')"* />

For viewing/requesting/adding/editing items:

<security:intercept-url pattern=*"/items/\*"* access=*"isAuthenticated()"* />

Method Authentication:

In com.codekarma.service.Impl.ResourceServiceImpl.java

@PreAuthorize(value="hasRole('ADMIN')")

**public** **void** grantRequestForRequestId(Integer requestId) {

….

}

@PreAuthorize(value="hasRole('ADMIN')")

**public** **void** discardRequestForRequestId(Integer requestId) {

…

}

In com.codekarma.service.Impl.MemberServiceImpl.java

@PreAuthorize(value="hasRole('ADMIN')")

**public** **void** add(Member member) {

…

}

Enabling/Disabling Capabilities:

In Views/items/list.jsp: (Restricting the capabilities of ADMIN, USER and users who are not logged in)

<security:authorize access=*"hasRole('ADMIN')"*>

<a href=*"*<spring:url value='*/items/delete/*${resource.id}'/>*"*>Delete</a> |

<a href=*"*<spring:url value='*/items/edit/*${resource.id}'/>*"*>Edit</a> |

</security:authorize>

<security:authorize access=*"isAuthenticated()"*>

<c:set var=*"available"* value=*"AVAILABLE"*/>

<c:if test=*"*${resource.status **eq** available }*"*>

<a href=*"*<spring:url value='*/items/request/*${resource.id}'/>*"*>Request</a> |

</c:if>

</security:authorize>

<a href=*"*<spring:url value='*/items/detail/*${resource.id}'/>*"*>View Detail</a>

In WEB-INF/tiles/template/baseLayout.jsp: (Choosing which of the Login or Logout options to display)

<li> [

<security:authorize access=*"isAuthenticated()"*>

<a href=*"*<spring:url value=*"/logout"*/>*"* >Logout</a>

</security:authorize>

<security:authorize access=*"isAnonymous()"*>

<a href=*"*<spring:url value=*"/login"*/>*"* >Login</a>

</security:authorize>

]</li>

In WEB-INF/tiles/template/navigation.jsp:

<security:authorize access=*"hasRole('ADMIN')"*>

<a href=*"*<spring:url value=*"/items/item.add"* />*"*>Add Resource</a>

<a href=*"*<spring:url value=*"/items/requests.resolve"* />*"*>Resolve Requests</a>

</security:authorize>

<security:authorize access=*"isAuthenticated()"*>

<a href=*"*<spring:url value=*"/items/list"* />*"*>Request Resource</a>

</security:authorize>

**VALIDATION:**

In Views/items/add.jsp:

<form:input id=*"itemCode"* path=*"itemCode"*/><form:errors path=*"itemCode"*/>

In com.codekarma.domain.Item:

@NotBlank

@Column

**private** String itemCode;

**RESTful Service:**

In WEB-INF/views/items/add.jsp:

The options of select list for rooms are changed using RESTful service (AJAJ). This is triggered when the selected value of the dropdown list of Buildings is changed. The jQuery code is in webapp/resources/js/site.js (function: getRoomsForBuilding).

The following value of url, without providing the context path,

[url:*"../room/forbuilding/"*+id](url:%22../room/forbuilding/%22+id) [Note: ../ is provided instead of context path]

mapped to : */resourcemanagement/room/forbuilding/45* (e.g., when id = 45).

Further, PathVariable is used in the method, getRoomsInTheBuilding() in com.codekarma.controller.RoomController. This method is the RESTful service for the AJAJ used by WEB-INF/views/items/add.jsp page.

**Exception Handling:**

An exception class com.codekarma.exception.InvalidOperationException is defined to represent exception. When a user request a resource that was already requested (by same user or other user), then this exception is thrown. This is in line 112 (method addNewRequest) in class com.codekarma.service.Impl.ResourceServiceImpl. This exception is handled by method handleErrorPage() in @ControllerAdvice com.codekarma.controller.ControllerExceptionHandler.