DCAT Design Document

*For*  
  
**Release 1.7**



Version No.: 0.1

May 2, 2013

Design Document for DCTM WF UI

|  |  |  |  |
| --- | --- | --- | --- |
| Prepared by: | Name | Functional Area | Date |
|  | DEV Team |  | 2nd May,2013 |

|  |  |  |  |
| --- | --- | --- | --- |
| Approved by: | Name | Functional Area | Date |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Reviewed by: | Name | Functional Area | Date |
|  |  |  |  |
|  |  |  |  |

Revision Log

All revisions to this document shall be documented in this log. The initial draft document shall be numbered 0.1 and incremented when revisions are made. The document that is approved shall be Version 1.0. Version 1.0 shall be submitted to CM for formal version change control.

|  |  |  |  |
| --- | --- | --- | --- |
| Revision | Date | Description | Name |
| 0.1 | 05/02/2013 | Draft Version | DCAT DEV Team |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Glossary

|  |  |
| --- | --- |
| Abbreviation | Description |
| DCAT | Digital Content Assembly Tool |

Table of Contents

[1. Introduction 5](#_Toc355539678)

[1.1 Purpose 5](#_Toc355539679)

[1.2 Background 5](#_Toc355539680)

[2. Implementation 6](#_Toc355539681)

[2.1 DCAT-548 6](#_Toc355539682)

[2.1.1 Flow Diagram 6](#_Toc355539683)

[2.1.2 File Details 7](#_Toc355539684)

[2.2 DCAT-563 7](#_Toc355539685)

[2.2.1 UI Steps 7](#_Toc355539686)

[2.2.2 Flow Diagram 13](#_Toc355539687)

[2.2.3 File Details 14](#_Toc355539688)

[2.3 DCAT-513 14](#_Toc355539689)

[2.3.1 UI Steps 14](#_Toc355539690)

[2.3.2 Flow Diagram 15](#_Toc355539691)

[2.3.3 File Details 16](#_Toc355539692)

[2.4 DCAT-550 16](#_Toc355539693)

[2.4.1 Flow Diagram 17](#_Toc355539694)

[2.4.2 File Details 19](#_Toc355539695)

[2.4.3 Class 19](#_Toc355539696)

[2.5 DCAT-532 20](#_Toc355539697)

[2.5.1 File Details 20](#_Toc355539698)

[2.6 DCAT-616 20](#_Toc355539699)

[2.6.1 File Details 20](#_Toc355539700)

[2.6.2 Class 21](#_Toc355539701)

[2.7 DCAT-619 21](#_Toc355539702)

[2.7.1 File Details 21](#_Toc355539703)

[2.8 DCAT-230 22](#_Toc355539704)

[3. Class Diagrams 22](#_Toc355539705)

[3.1 controllers package: 22](#_Toc355539706)

[3.2 models package: 23](#_Toc355539707)

[4 Limitations and Alternative Approach 31](#_Toc355539708)

[5 Object Model 31](#_Toc355539709)

[6 Folder Structure 31](#_Toc355539710)

[7 Security Model 31](#_Toc355539711)

[8 Design Criteria 32](#_Toc355539712)

[9 List of Reusable Components 32](#_Toc355539713)

[10 List of Time Critical and Periodic Functions 32](#_Toc355539714)

[11 Reliability Requirements 32](#_Toc355539715)

[12 Anticipated Changes 32](#_Toc355539716)

[13 Assumptions 32](#_Toc355539717)

[14 References 32](#_Toc355539718)

# Introduction

## 1.1 Purpose

The objective of this document is to outline a detailed design for the resolution of the following SARs

* DCAT-548
* DCAT-563
* DCAT-513
* DCAT-550
* DCAT-532
* DCAT-616
* DCAT-619
* DCAT-230

## 1.2 Background

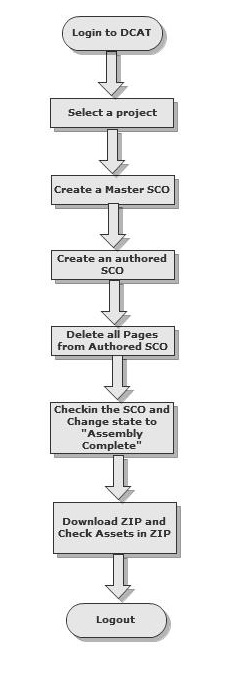
This design document provides solution:

* Unused Assets should not be downloaded in zipped SCO.
* “Edit Player Option” functionality is implemented and all the existing functionalities that use runtime.json and authoring.json in the backend are replaced by player.json file.
* Addition of “Build Program” button in the Admin tool functionality. Earlier auto build of root SCOs were happening in tomcat restart which is needed to be stopped.
* Re-create ZIP and PIF SCO files(DELETED) after running SCO propogation script and Auto Sync Job
* Spaces are removed when pasting
* DCAT Reports enhancement to include timestamp for “Last Modified” column instead of date and two new columns are added “PIF Url” and “ZIP Url” files.
* Unable to Author new Gadgets added to a Page.
* Mass State Change - update of SCO status.

# Implementation

## 2.1 DCAT-548

2.1.1 Flow Diagram



#### 2.1.2 File Details

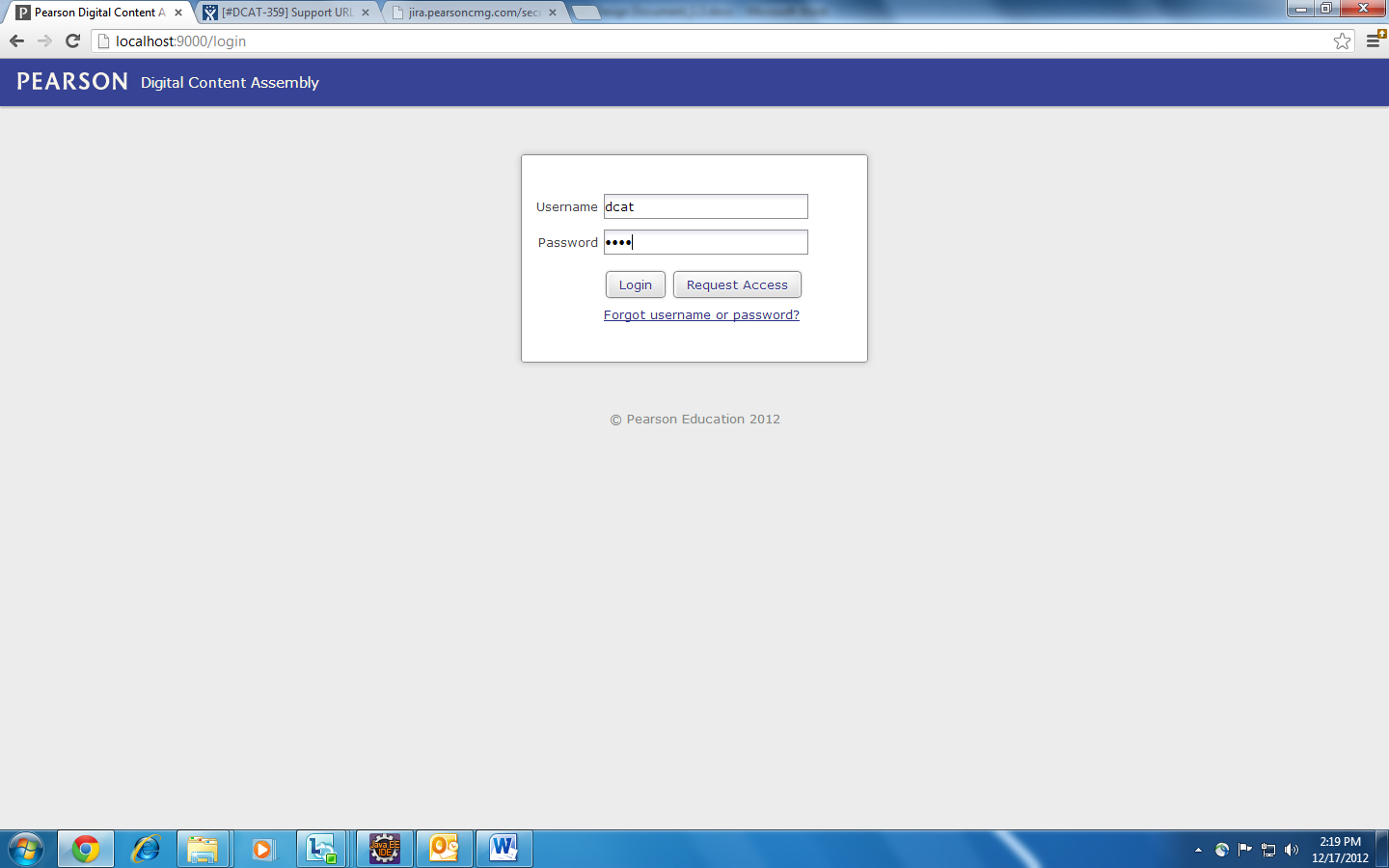
|  |  |  |  |
| --- | --- | --- | --- |
| File Name | Type(New/Existing) | Location | Change Description |
| ContentObject.java | Existing | /app/models/ContentObject.java | In Publish() method:  /\*\*DCAT-548\*\*/ if((extension!=null) && (extension.equalsIgnoreCase("json")) && (!name.contains("tmp")))  Placement of this concatenated string:  concatenatedAuthoredJSONs += FileUtils.readFileToString(temporaryAuthoredJSON);  /\*\*DCAT-548\*/ |

## 

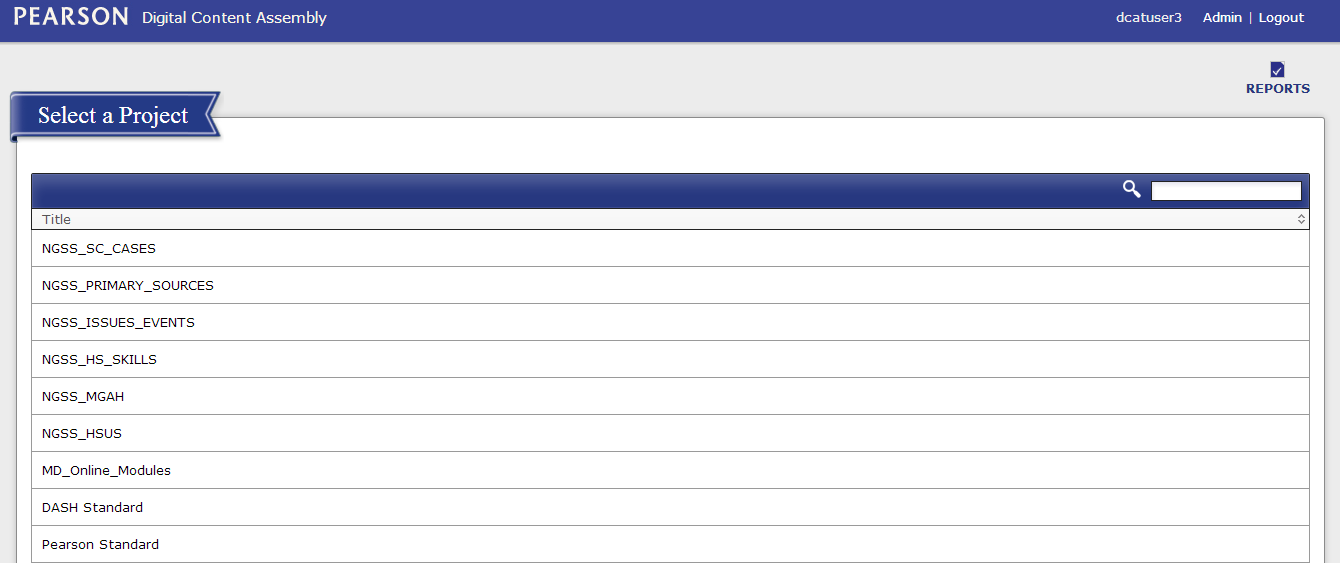
## 2.2 DCAT-563

2.2.1 UI Steps

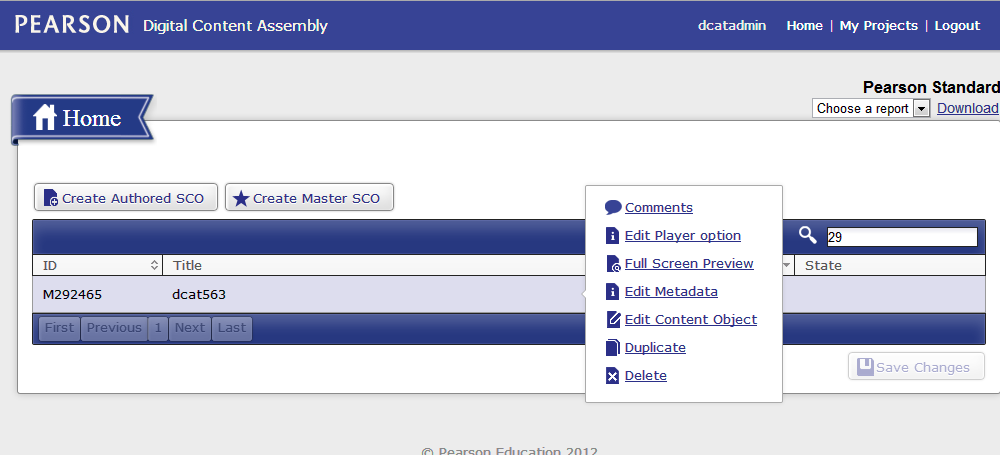
**Step 1:** Login to DCAT by providing valid credentials.



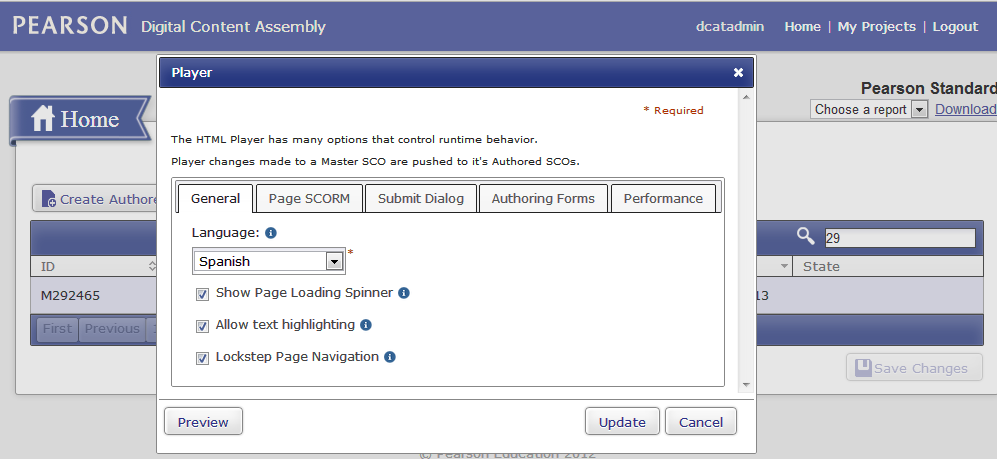
**Step 2:** Select the project

****

**Step 3:** Mouse over a master SCO and Click on “Edit Player Option”.

****

**Step 4**: Below is the pop interface that is used for editing the player variables in DCAT



The form above would produce the following JSON fragment:

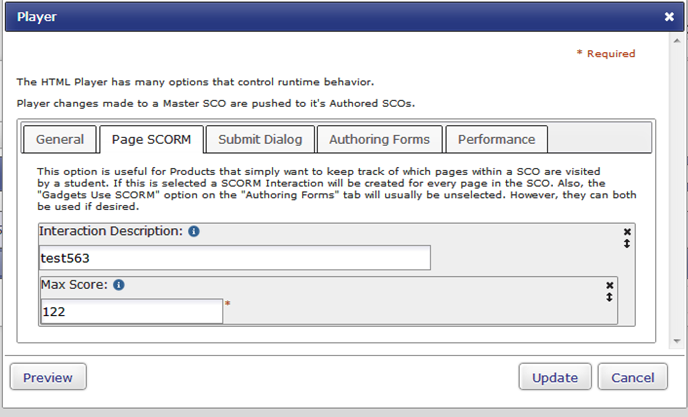
"language": "en",

"spinner": true,

"highlightEnabled": true,

"lockstep": true,

**Step 4.1: “**Page SCORM” Form for the pop interface



The form above would produce the following JSON fragment:

"interactions": [

{

"description": "test563",

"objectives": [

{

"score": {

"max": 122

}

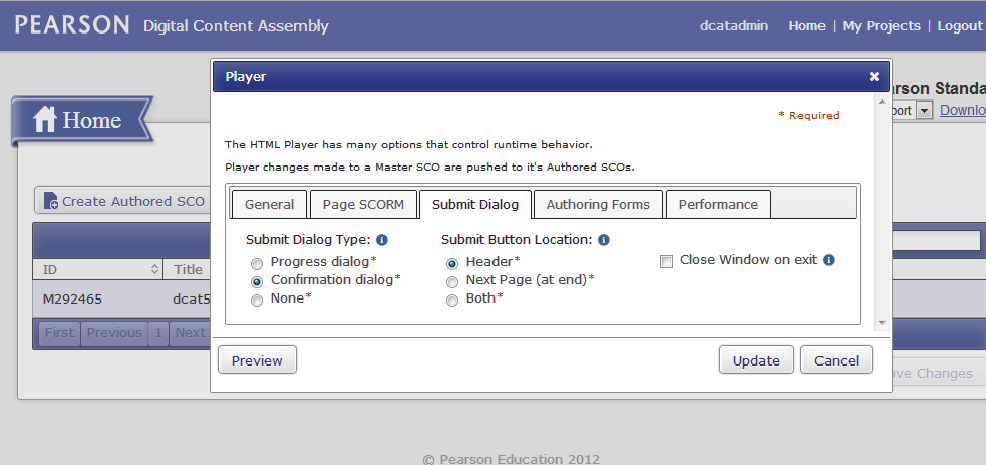
}

]

}

],

**Step 4.2:** “Submit Dialogue” for the pop interface

****

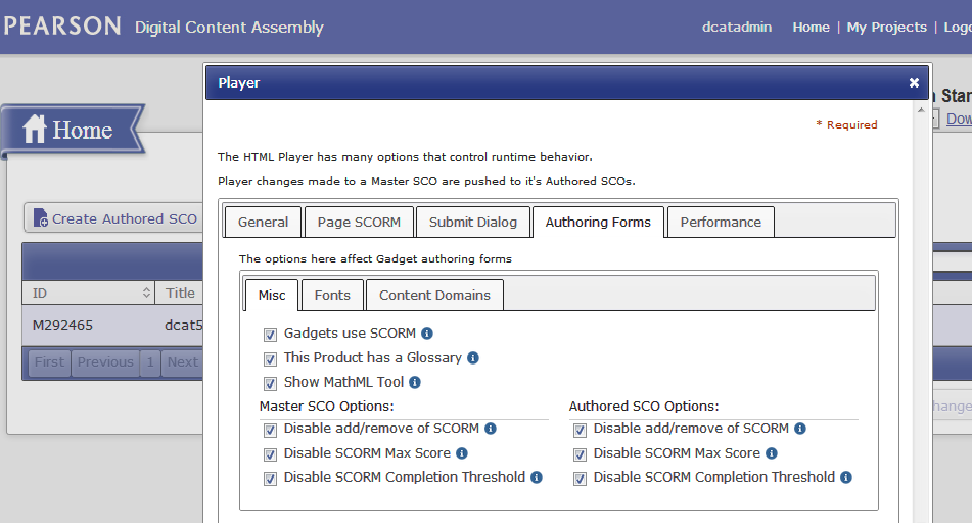
The form above would produce the following JSON fragment:

"dialogType": "confirmationDialog",

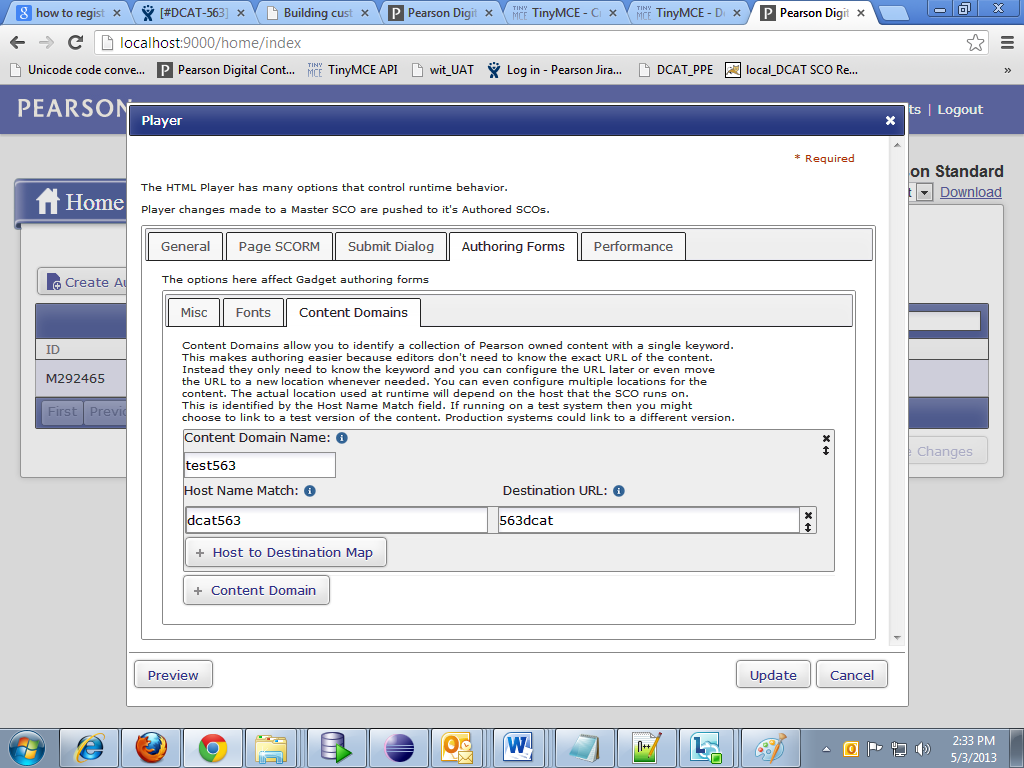
"submitLocation": "Header",

"closeOnExit": false,

**Step 4.3:** “Authoring Forms” Tab for the pop interface has multiple views as it consists of sub tabs.

****

**Step 4.4:** Content Domains sub tab of “Authoring Forms” tab.



**The form on the authoring tab produces the following JSON fragment:**

"authoring": {

"HasGadgetSCORM": true,

"HasGlossary": true,

"masterscos": {

"SCORMListDisabled": true,

"SCORMMaxScoreDisabled": true,

"SCORMCompThresholdDisabled": true

},

"authoredscos": {

"SCORMListDisabled": true,

"SCORMMaxScoreDisabled": true,

"SCORMCompThresholdDisabled": true

},

"FontSet": [

{

"optionLabel": " test563label ",

"optionValue": " test563value "

}

],

"ExtraFonts": [

{

"optionLabel": "test563label",

"optionValue": "test563value"

}

],

"Domains": [

{

"optionValue": "TEST563",

"map": [

{

"host": "DCAT563",

"destination": "563DCAT"

}

]

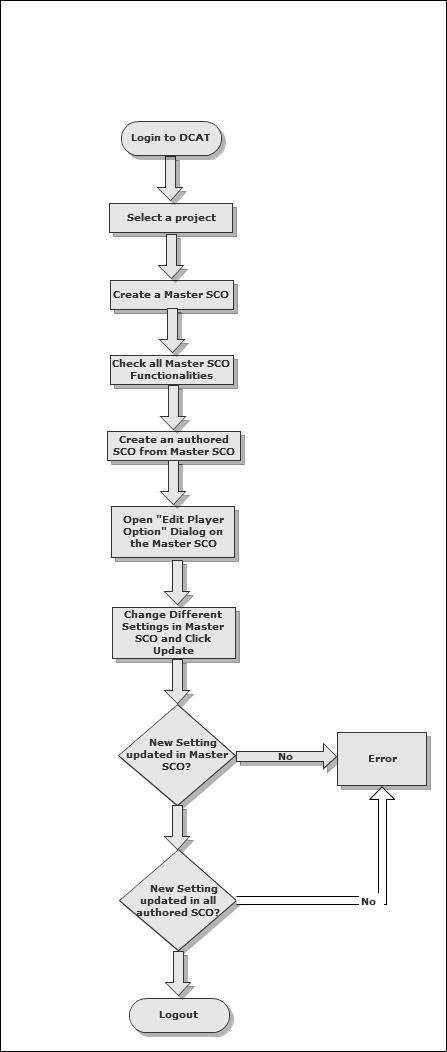
}

]

},

Consolidating all the above jsons generated by this player edit pop up forms player.json file.

#### 2.2.2 Flow Diagram



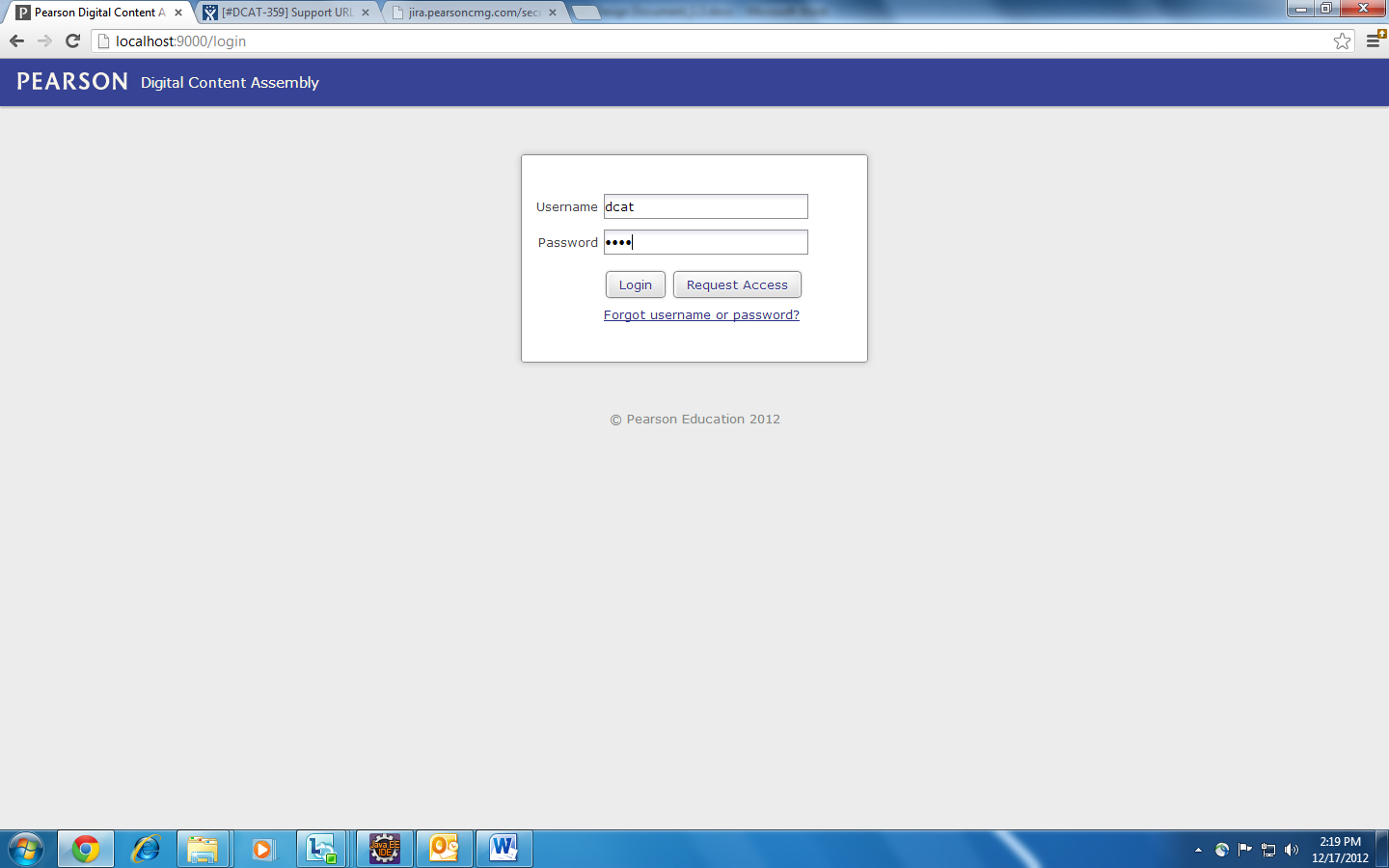
#### 2.2.3 File Details

|  |  |  |  |
| --- | --- | --- | --- |
| File Name | Type(New/ Existing) | Location | Change Description |
| Player.java | New | /app/controllers/player/Player.java | New class to get the player.json and player.xml |
| Object.java | Existing | /app/controllers/object/Object.java | Methods to update the json file for master and authored sco |
| ContentObject.java | Existing | /app/models/ContentObject.java | Methods to write the json |
| index.js | Existing | /public/javascripts/home/index.js | Method to open the pop up on click |
| gadget-config.js | Existing | /public/javascripts/gadget/gadget-config.js | Consists of method to parse the player.xml |
| index.html | Existing | /app/views/home/index.html | Includes the java method to be called from client |
| SampleContentObject.java | Existing | /app/models/SampleContentObject.java | Replacing authoring.json and runtime.json with player.json |

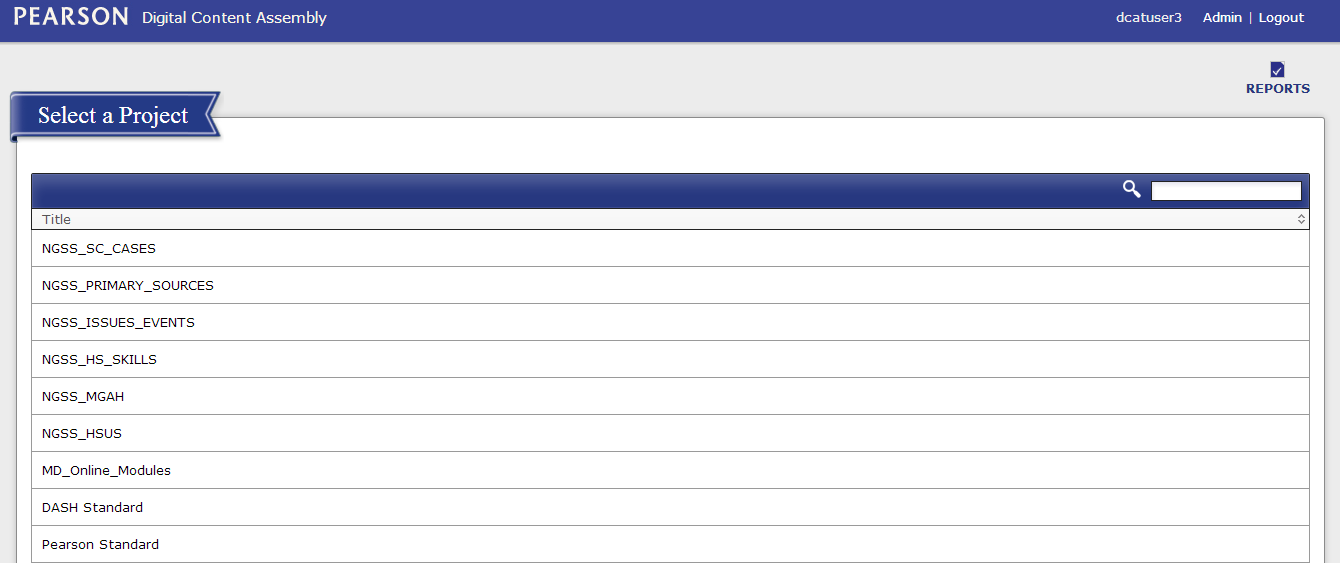
## 2.3 DCAT-513

#### **2.3.1 UI Steps**

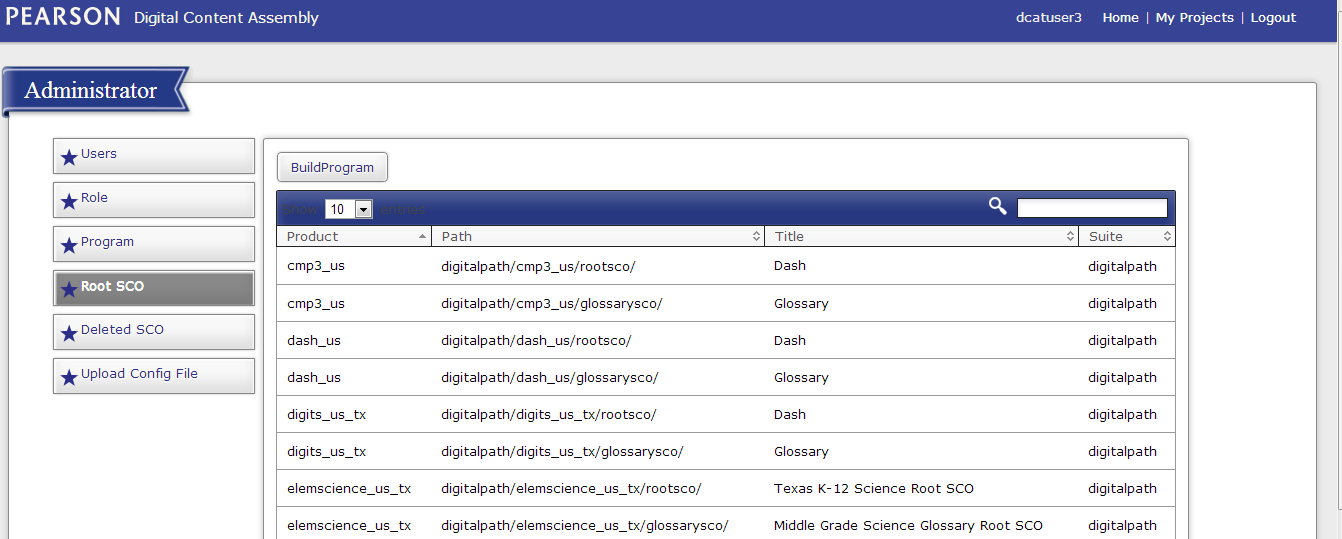
**Step 1:** Open the Login Page. Enter the credentials and login.



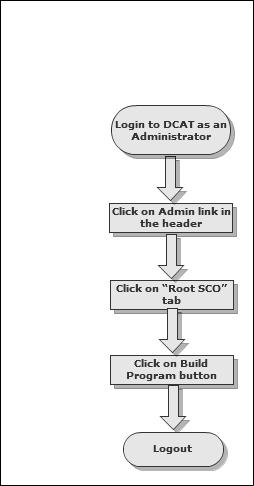
**Step 2:** Click on Admin Link.

****

**Step 3:** Click on “Root SCO” tab. Click on “Build Program” button.



#### **2.3.2 Flow Diagram**



#### **2.3.3 File Details**

|  |  |  |  |
| --- | --- | --- | --- |
| File Name | Type(New/ Existing) | Location | Change Description |
| Admin.java | Existing | /app/controllers/Admin.java | Adding the method : public static void buildPrograms() |

## 2.4 DCAT-550

PIF and ZIP files are getting deleted in the backend by various services. So, it is taking time for user to get the PIF/ZIP files on request and consecutive request for same PIF file are creating problems.

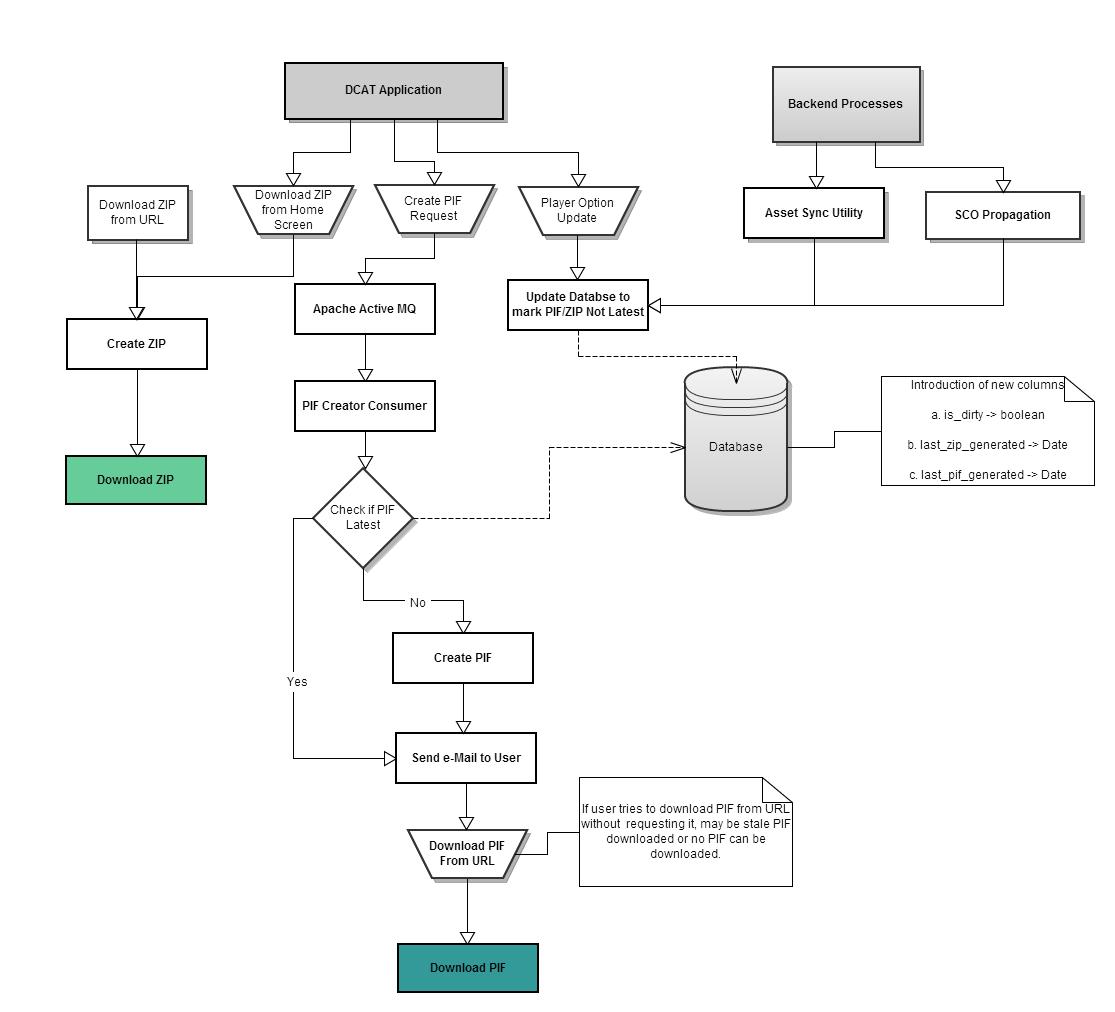
To fix this issue, the followings are needed to be done.

1. Introduction of new columns in backend is required
   1. is\_dirty 🡪 boolean
   2. last\_zip\_generated 🡪 Date
   3. last\_pif\_generated 🡪 Date
2. Introduction of new screen through which users need to request for PIF Creation every time they need a PIF to ensure that they do not receive stale files in PIF
   1. User need to select SCOs for which PIF need to be created.
   2. PIF would be created asynchronously and user would receive an e-mail whenever the PIFs are ready to be downloaded in the backend
3. Change in SCO Propagation Script
   1. It would not delete ZIP/PIF from the backend
   2. It would update is\_dirty column in the backend for those SCOs in which the files are getting updated to ensure that next time PIF and ZIP files need to be recreated
4. Change in DCAT Asset Sync Utility
   1. It would not delete ZIP/PIF from the backend
   2. It would update is\_dirty column in the backend to ensure that next time PIF and ZIP files need to be recreated
5. Change in ZIP creation Logic
   1. ZIP would be created in the backend only if user clicks on “Download ZIP” from the UI.
   2. ZIP would not be created on SCO State Change.
6. If user wants to download PIF files without requesting this through UI
   1. User may not be able to download PIF files

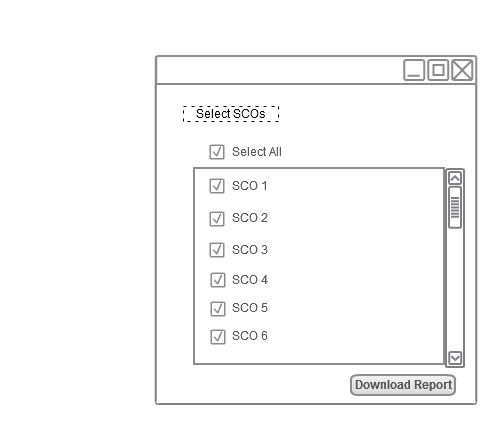
OR

* 1. User may download PIF file containing stale files.

2.4.1 Flow Diagram



2.4.2 Create PIF Request Screen



#### **2.4.3 File Details**

|  |  |  |  |
| --- | --- | --- | --- |
| File Name | Type(New/ Existing) | Location | Change Description |
| CreatePifListener.java | New | com.pearson.dcat.listener | Method :  public static void createPifReqReceived()  Receive the Create PIF Queue Item |
| CreatePifImpl.java | New | com.pearson.dcat.service | Method :  public void download(CreatePifReq downloadMessage)  Create PIF file and send e-mail. |

#### **2.4.4 Class**

* CreatePifListener.java

|  |  |
| --- | --- |
| **Method** | |
|  | |
| **Signature** | public static void createPifReqReceived() |
| **Description** | This method is used to receive the Create PIF Queue Item |
|  | |

* CreatePifImpl.java

|  |  |
| --- | --- |
| **Method** | |
|  | |
| **Signature** | public void download(CreatePifReq downloadMessage) |
| **Description** | This method is used to create PIF file. |
|  | |
| **Signature** | public void updateStatus(String status, String message, long id) |
| **Description** | This method is used to update status of create PIF job. |
|  | |
| **Signature** | public int getPendingSCOCount(String status, long jobId) |
| **Description** | This method is used to get the pending SCO count for which PIF needs to be created. |
|  | |
| **Signature** | public void updateSuccessSCOCount(String status, long id) |
| **Description** | This method is used to update the successful SCO count for which PIF is created. |
|  | |
| **Signature** | private boolean createZip(String dcatPath, String zipLocation, String scoId) |
| **Description** | This method is used to create ZIP file if not created in the backend. |

## 2.5 DCAT-532

According to the SAR, if any text is being pasted to DCAT from any other sources contain more than one space in between words; it automatically removes the extra spaces between words. At times, these extra spaces are required. Hence, DCAT should not remove any extra space between words while pasting text in DCAT.

To fix this issue, configuration change in tiny-mce plugin is required.

2.5.1 File Details

|  |  |  |  |
| --- | --- | --- | --- |
| File Name | Type(New/ Existing) | Location | Change Description |
| gadget-config.js | Existing | /public/javascripts/gadget/gadget-config.js | Change Required :  In Com.Gadget.Config.Element.RichText.EDITOR\_CONFIGURATION, change “paste\_text\_sticky\_default: false” to  “paste\_text\_sticky\_default: true”. |

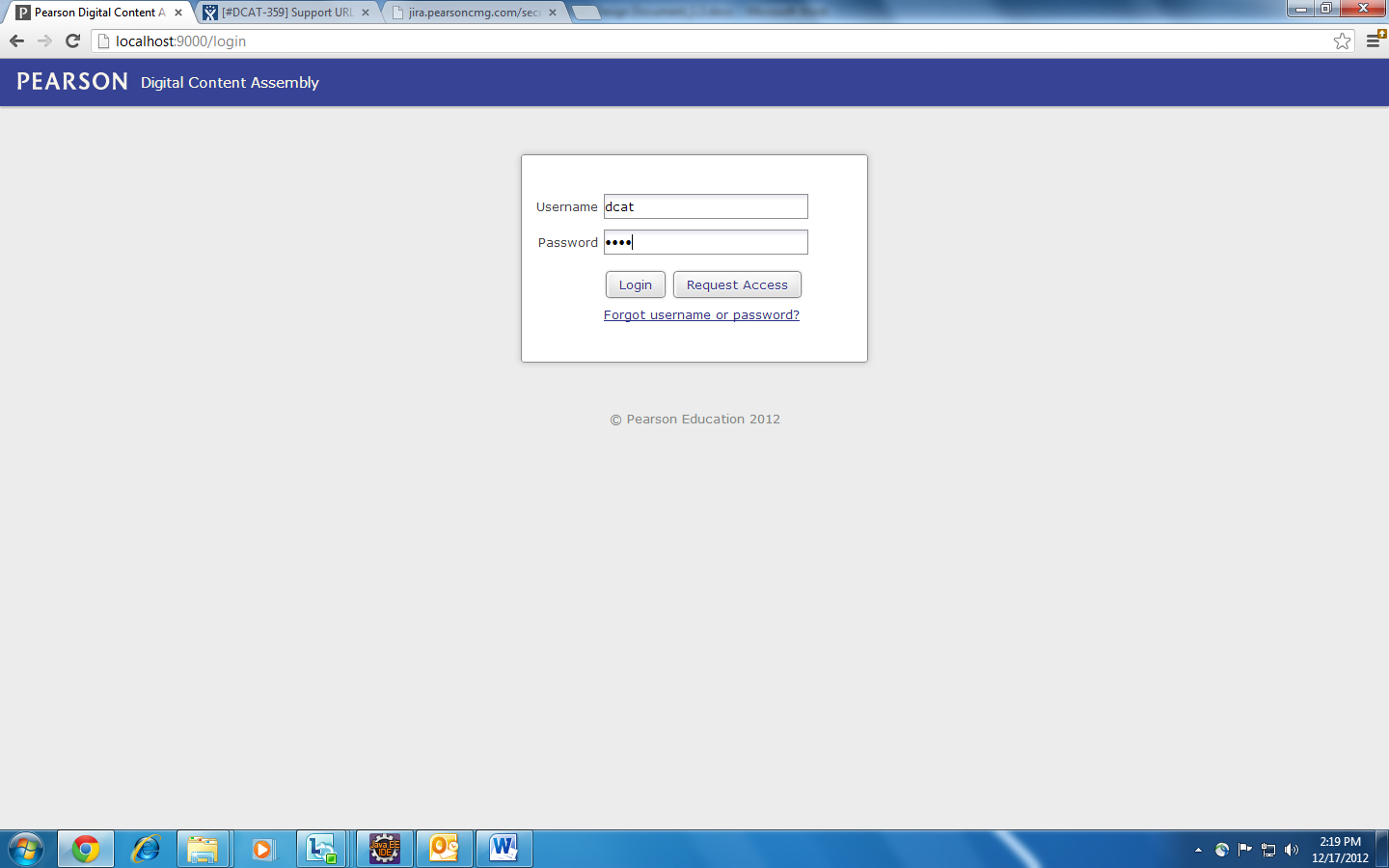
# 

# 2.6 DCAT-616

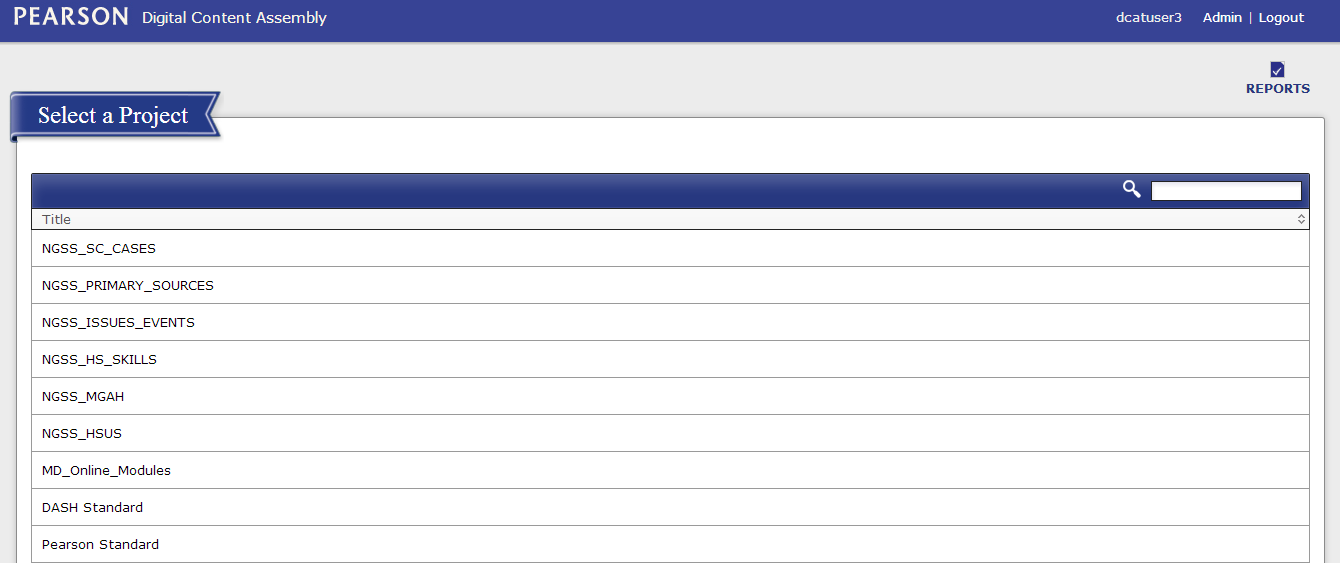
Enhancement is required in DCAT Reports to include timestamp for “LastModified” column and two new columns are added “PIF URL” and “ZIP Url”.

2.6.1 UI Steps

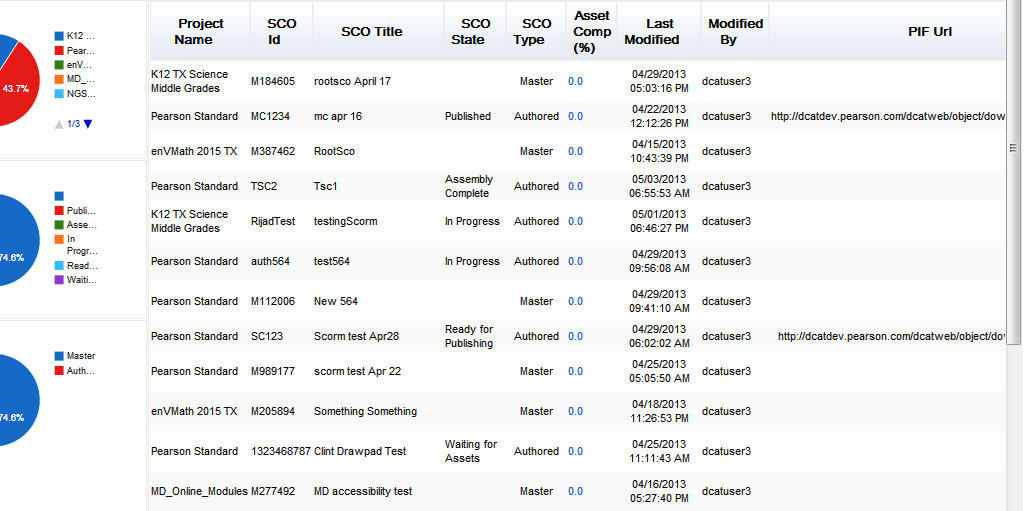
**Step 1:** Login to DCAT by providing valid credentials.

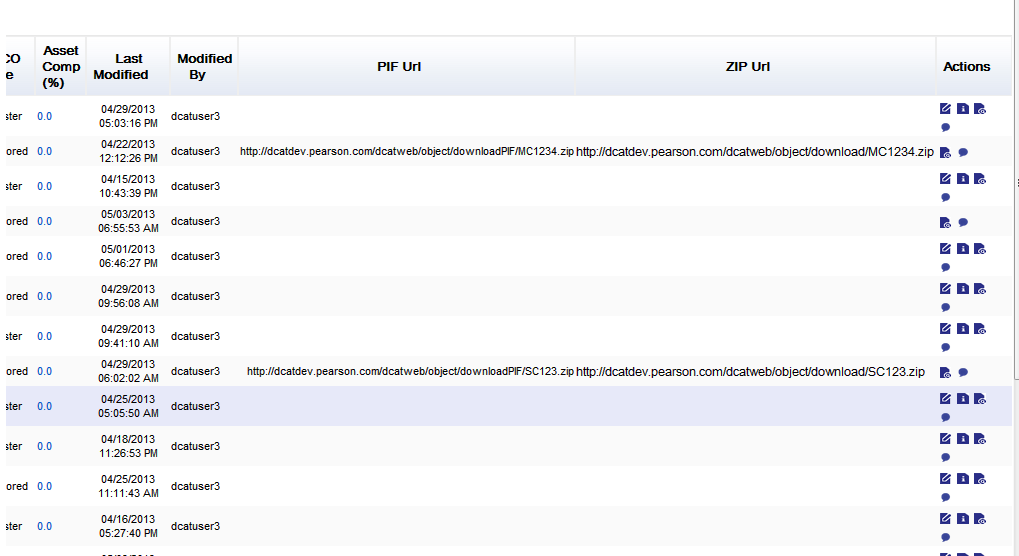


**Step 2:** Click on the “REPORT” icon.

****

You can see the below screen with new two columns “PIF Url”, “ZIP Url” and the “Last Modified” column with date and time.





2.6.2 File Details

|  |  |  |  |
| --- | --- | --- | --- |
| File Name | Type(New/ Existing) | Location | Change Description |
| JSCacheServiceImpl.java | Existing | /app/ReportBean/JSCacheServiceImpl.java | Behavior class to create JSON for report. |
| SCOReportHelper.java | Existing | /app/ReportBean/SCOReportHelper.java | Behaviour class to add two extra columns in the report. |

#### **2.6.3 Class**

* JSCacheServiceImpl.java

|  |  |
| --- | --- |
| **Method** | |
|  | |
| **Signature** | public void executeExportReport(String FilteredProgramName) |
| **Description** | This method is used to add the data to be shown in the report. |
|  | |

* SCOReportHelper.java

|  |  |
| --- | --- |
| **Method** | |
|  | |
| **Signature** | public ArrayList<Dataconf> dataConfValues() |
| **Description** | This method is used to show columns to be shown in the report. |
|  | |

# 2.7 DCAT-619

According to SAR, user is unable to author new gadgets added to a Page.

This happens if a new gadget is added to a page, then for old SCOs, new gadgets are added in page but the details are not added in json files and for this, these new gadgets are not working.

2.7.1 File Details

|  |  |  |  |
| --- | --- | --- | --- |
| File Name | Type(New/ Existing) | Location | Change Description |
| gadget-config.js | Exising | /public/javascripts/gadget/gadget-config.js | A small change is necessary to support call from edit-player.js |
| edit-player.js | Exising | /public/javascripts/edit/edit-player.js | If no entry is in json file for any gadget, a blank json structure is sent to call RIMA methods. |

# 2.8 DCAT-230

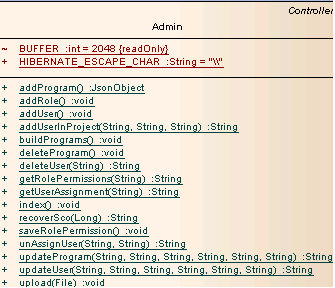
Ability to change SCO states for many SCOs at a time in DCAT.

# 3. Class Diagrams

## 3.1 controllers package:

**controllers.Admin.java**

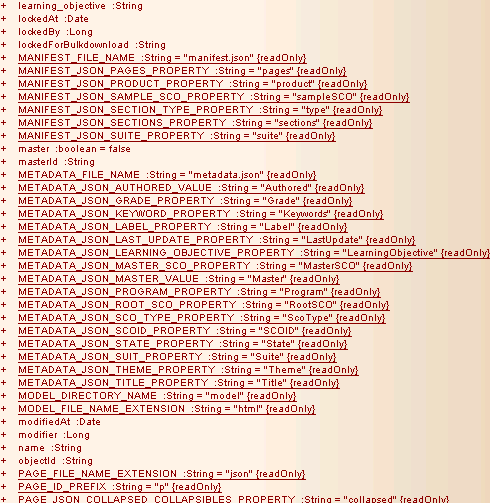
|  |  |
| --- | --- |
| **Description** | Class for Admin Tool |
|  | |
| **Method Signature** | public static void buildPrograms() |
| **Description** | Method is called to build the new scos. |
|  | |

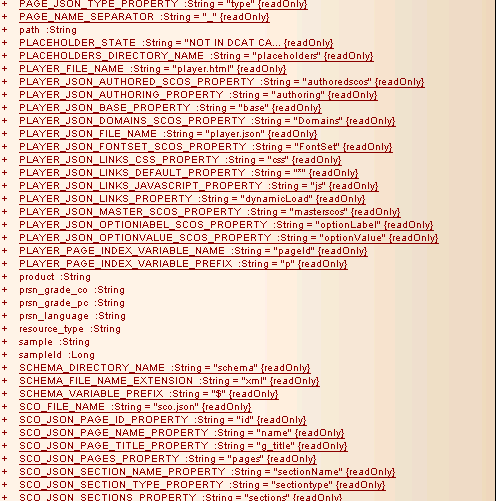
****

## 3.2 models package:

**ContentObject.java**



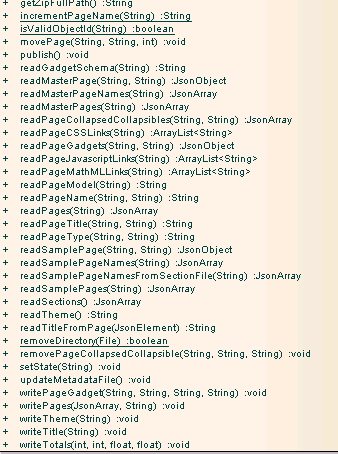






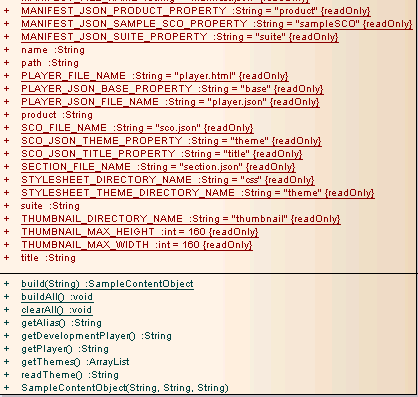






**SampleContentObject.java**





# 4 Limitations and Alternative Approach

NA

# 5 Object Model

NA

# 6 Folder Structure

NA

# 7 Security Model

NA

# 8 Design Criteria

DCAT system is designed keeping in mind Interoperability, Scalability, Maintainability and reliability.

# 9 List of Reusable Components

NA

# 10 List of Time Critical and Periodic Functions

NA

# 11 Reliability Requirements

NA

# 12 Anticipated Changes

NA

# 13 Assumptions

NA

# 14 References

**JIRA SAR Number:**

* DCAT-548
* DCAT-563
* DCAT-513
* DCAT-550
* DCAT-532
* DCAT-616
* DCAT-619
* DCAT-230