**Servillian Solutions Inc**

Spectrum Feature: [Content Gallery]

Projected Release: Version 2.2, [March 23th ,2012]

**Revision History**

|  |  |
| --- | --- |
| Person | Date |
| Ryan Goreshi | March 1st 2012 |
| Debra Holbrook | March 5, 2012 |
| Ryan Goreshi | March 6, 2012 |
| Ryan Goreshi | March 23,2012 |
|  |  |

**[Content Gallery] Glossary**

**[Detailed definition of Entities]**

Content Gallery is the general Term to manage/view [GUI/streaming] contents including photos, Videos (wmv, flv, etc), documents (doc, pdf, ppt) but at this version only Photo-type gallery will be setup on system. Current structure will apply to future content types listed above.

**[Section🡺 Operation🡺 General Business Logic]**

From the origin of content, actions happen to content and will be recorded as history is called **contentActionsHistory e**.g.upload, reject, approve, etc. and all **contentActionHistory**records will be saved for actions made per user/org/role. Contents will be involved in a workflow process and based on actions applied at each milestone their status will be changed. Whole flow process will be done by opening a ticket and making possible actions[**Table1**] over that ticket and save it as a history for that specific content.

At both front-end and back-end list of contents would be based on (last-action-made and/or status) combination

At the front-end or back-end only users who belong to at least one org can make actions on the accessible content and Only designated (org/roles) users will be able to change content-status Based on Status, content may be accessible to owner Org or SSI e.g. if last-action-made content is “sendToSsi” then this content won’t be viewable at back-end org.

Technologies in use:

Back-end: Extjs/Jquery/HTML/Shadowbox Module

Front-end: HTML/Jquery/Shadowbox Module

**[Section🡺 Operation🡺 Specific business Logic]**

|  |
| --- |
| BackEnd 🡺UPLOADING |

Purpose : uploading content file to owner-organization (for current version just league-level)

Uploading content can be done at (front-end owner ORG) users or (back-end  ownerORG)/roles managers. e.g. At league level (back-end  specific-ORG)/roles would be website Moderator and league executive.

Outcomes : uploaded files will be located in server and in pending status and not viewable at org-front-end website.

Next Possible Actions : Approve/reject at backend by owner organization.

ACL : organization [Executive and Website Moderator]Roles users.

GUI :

|  |
| --- |
| FrontEnd 🡺UPLOADING |

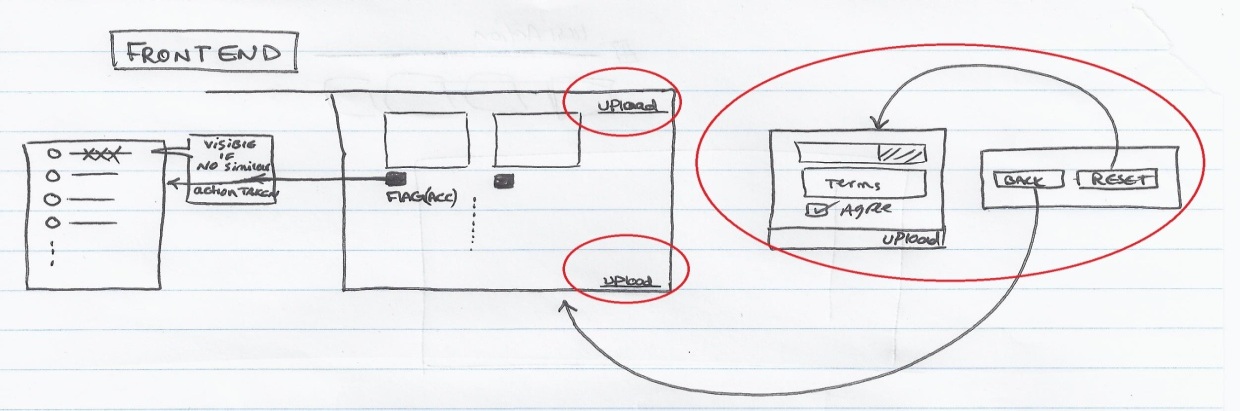
Purpose : users will be able to upload their own content to the website and uploaded content will be shared among all organization users.

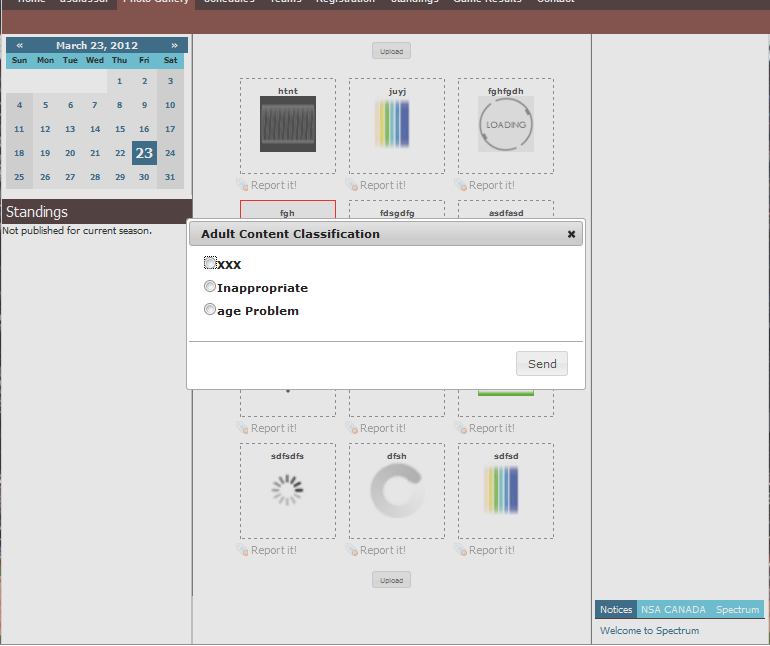
Outcomes : After uploading content, it will be in pending status and not viewable at front-end.

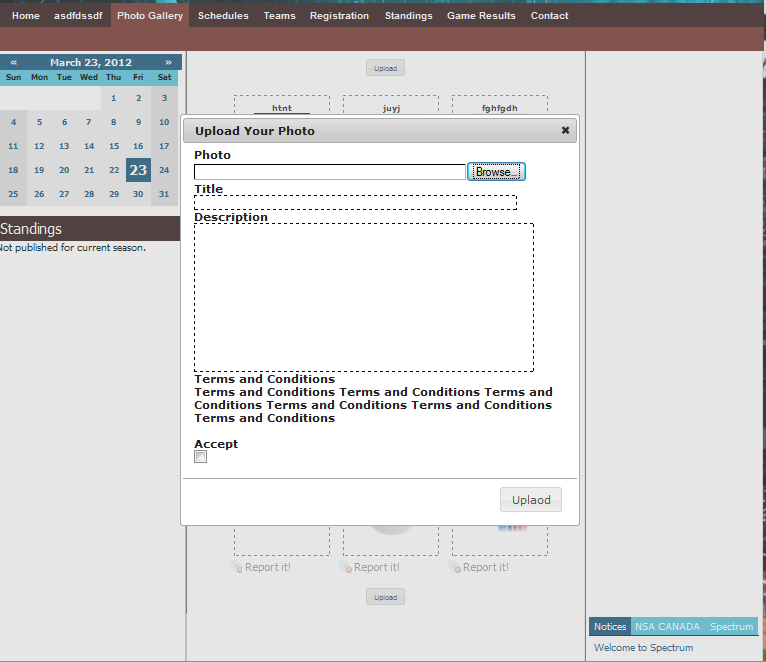
Possible Actions : approve/reject by organization designated Roles/users.

ACL : Logged-in users to the organization front-end website.

GUI :







|  |
| --- |
| Back-End 🡺View Content List |

Purpose : List all contents based on selected filter

Outcomes : After selecting an item, bottom panel will be updated and right panel including content summry will be updated.

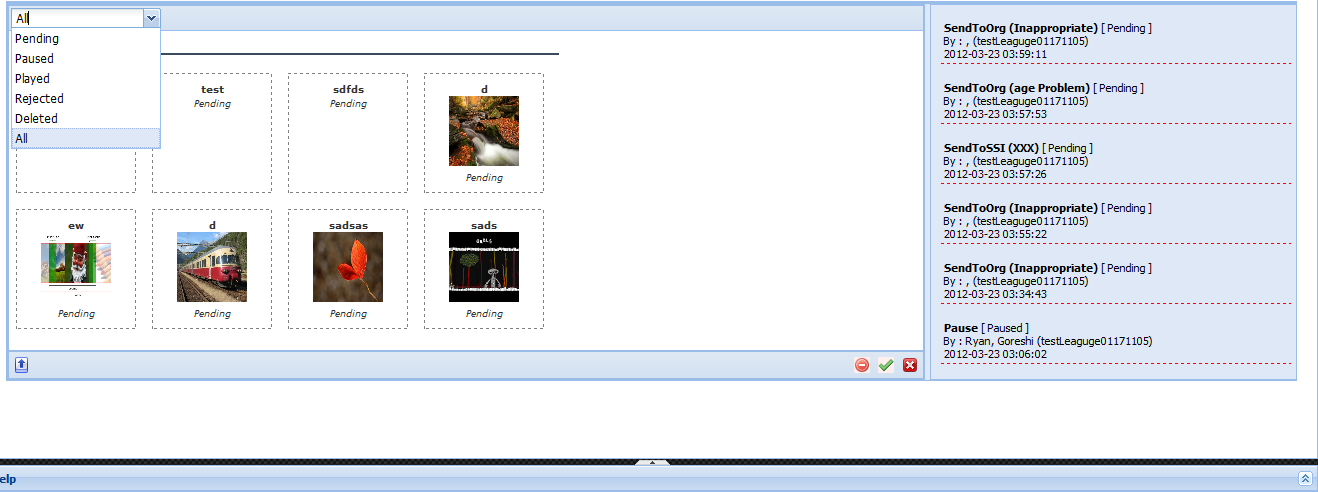
Possible Actions : Based on selected (status) filter [pending/played/paused/deleted/rejected] and make bottom panel items accessible

based on selectedItem/last-action-made.

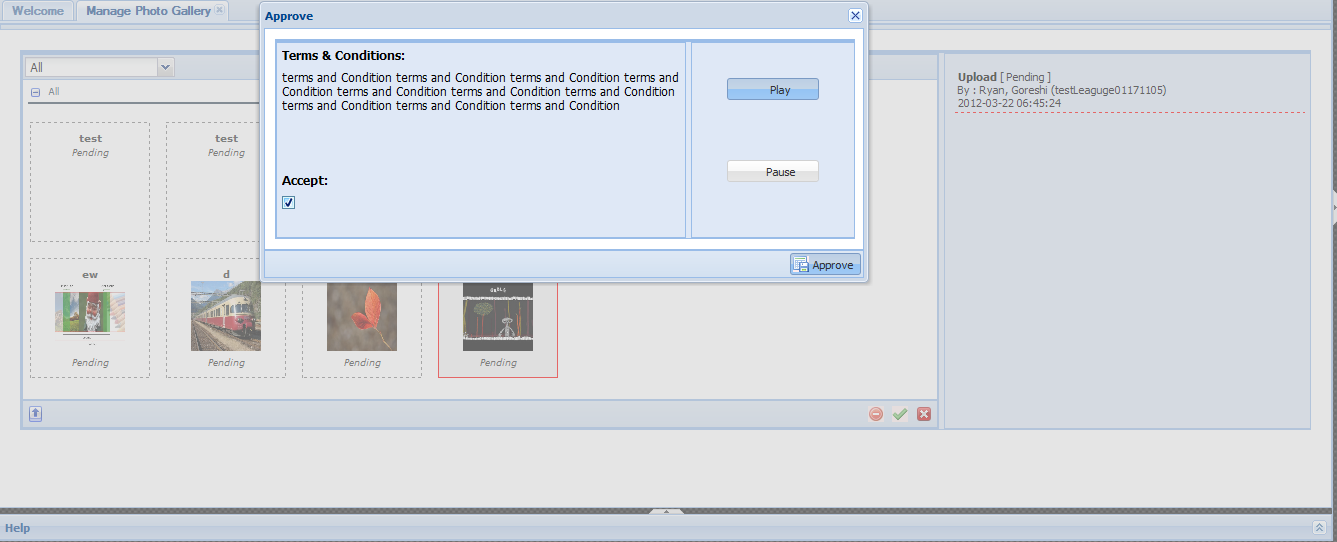
ACL : Designated org/role users

GUI : Main Dataview may list “contents” owned by the current organization and be filtered by content-action-status of last-action-taken record and based on filter selection, container’s bottom docked items will be (in)visible [ref.Table1]

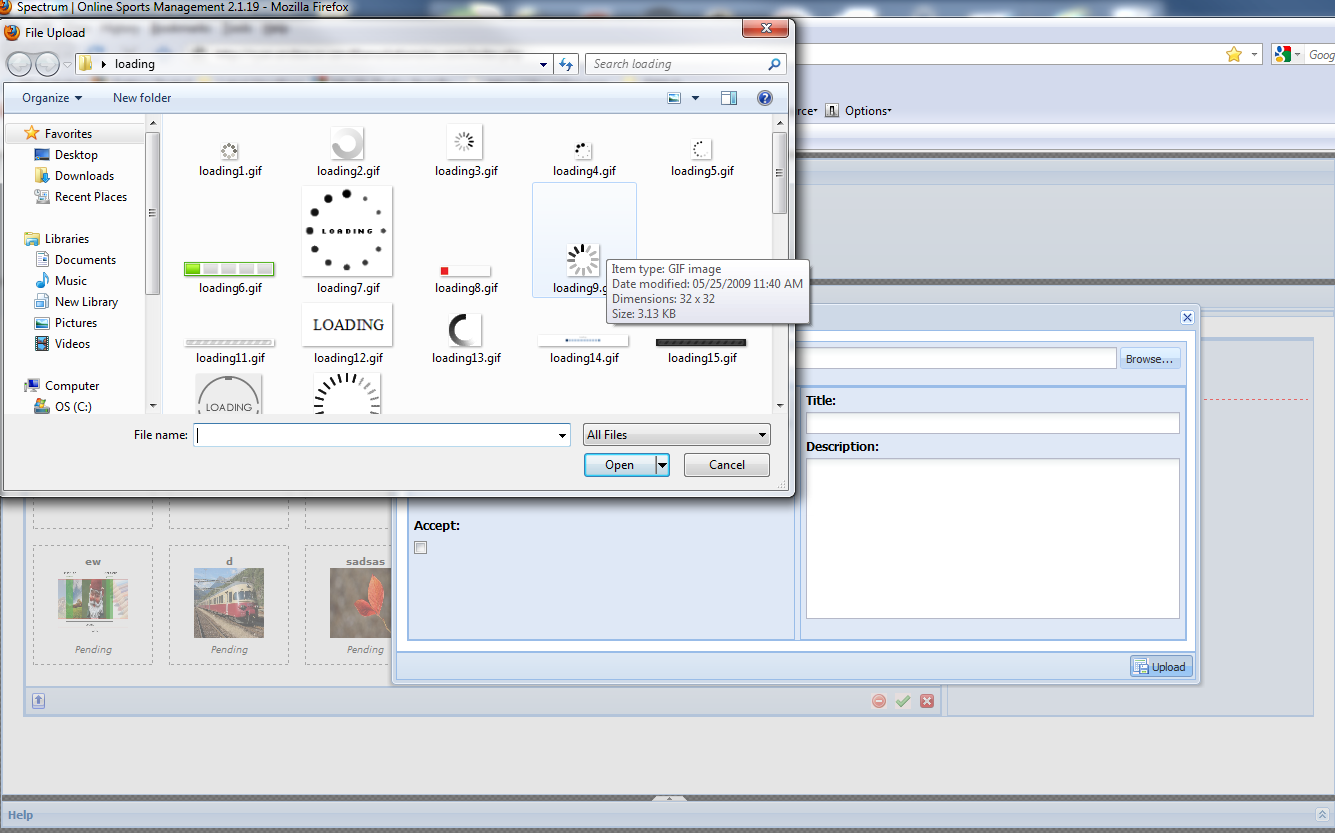
Main Grid and Navigation

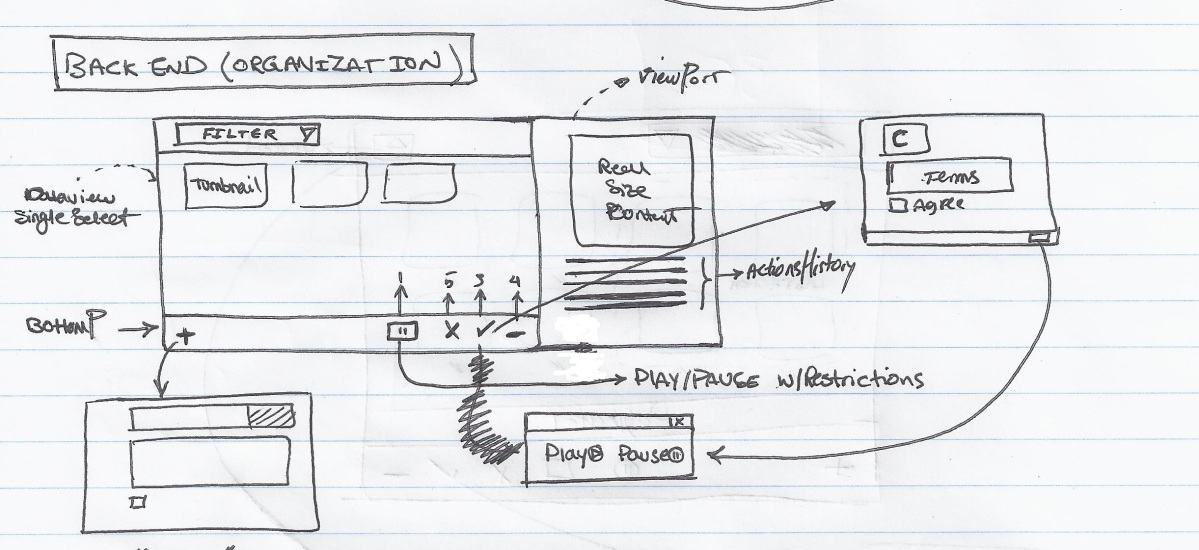


Approve



Upload Photo





|  |
| --- |
| Front-End 🡺View Content List |

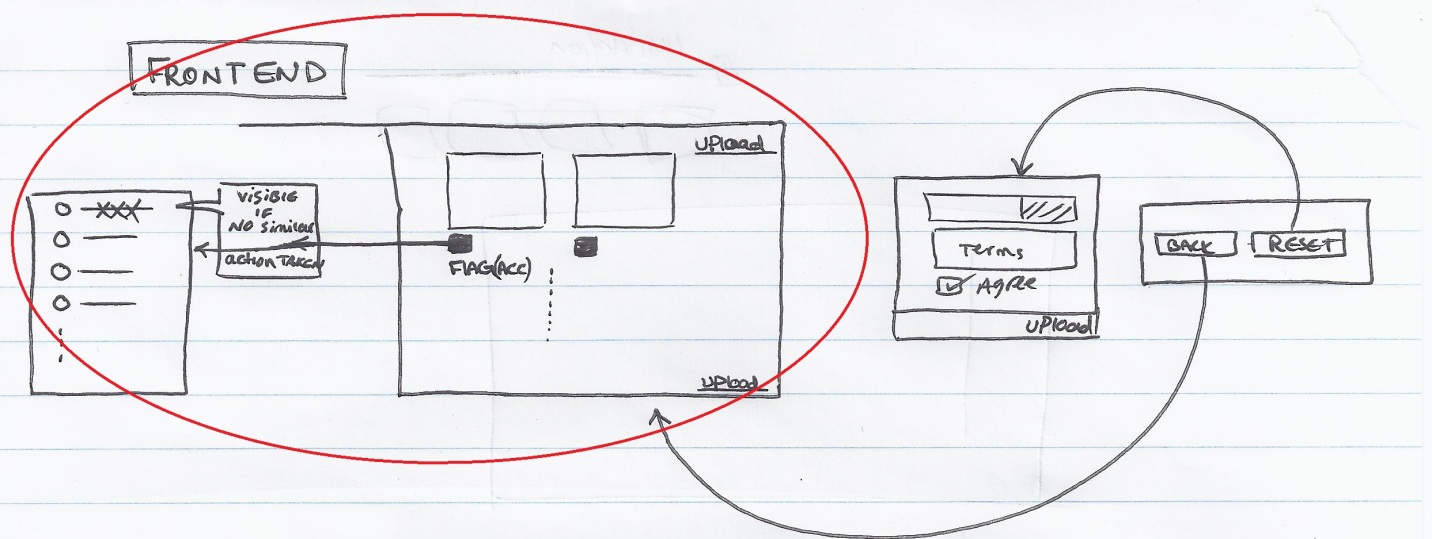
Purpose : List all org-related contents[type-based] which are in played-status

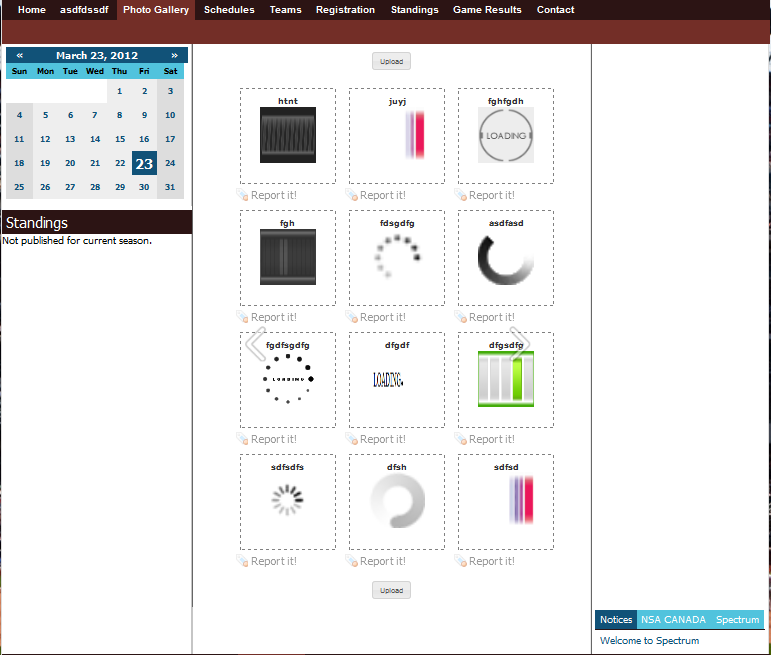
Outcomes : ~

Possible Actions : flag any of viewable contents that user thinks is inappropriate.

ACL : Org Logged in users

GUI :





|  |
| --- |
| SSI 🡺View Content List |

Purpose : Fetches List of all organizations “contents” with last-action-record set to “sendToSsi” i.e. this action is taken to be sent to ssi for decision making over

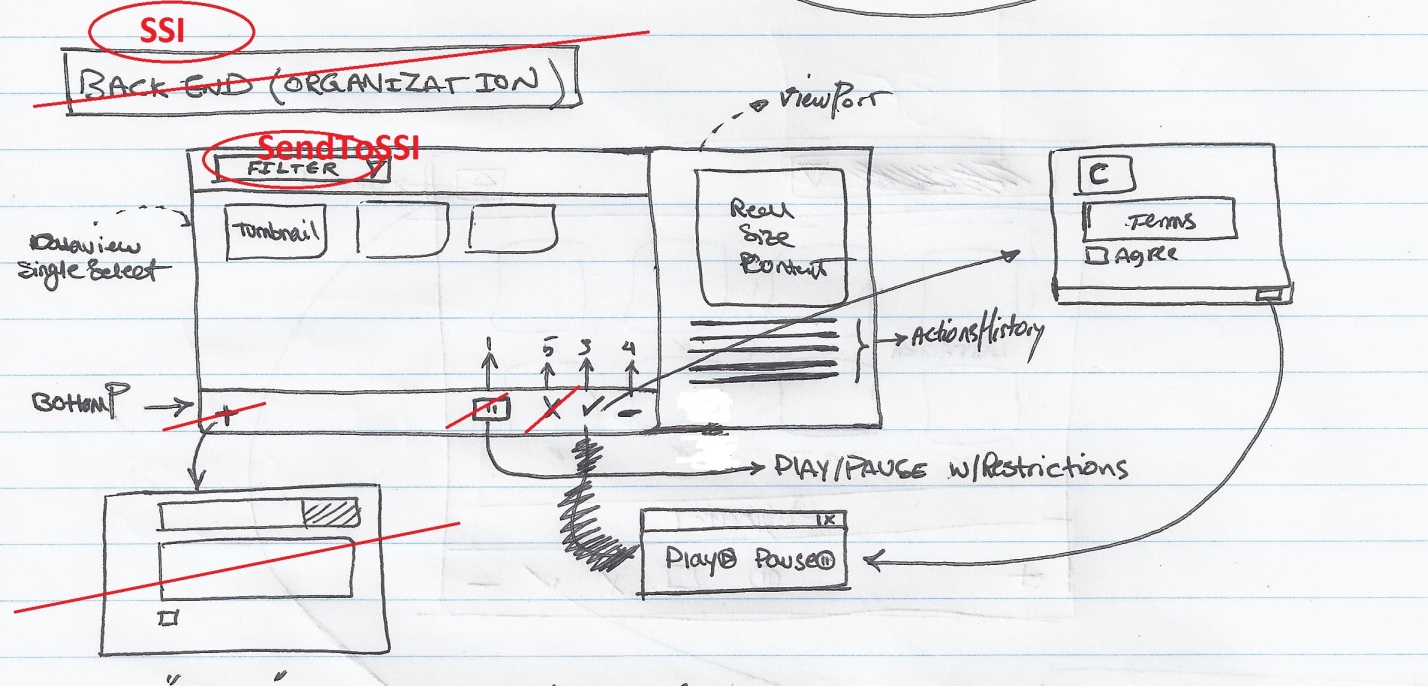
current content .SSI won’t be able to put any content owned by any organizations in to “deleted” status, and leave this decision to responsible organization.

Outcomes :

Possible Actions : Approve or Reject all contents which their status is “sendToSsi”

ACL : systemRole users

GUI : the same grid which is used for ORG - backend but w/ only one filter option set to “sendToSsi”



|  |
| --- |
| Back-End 🡺Approve |

Purpose : Approve selected Item that its current status is in rejected/pending status

Outcomes : Will be asked to put its status in played or paused

Next Possible Actions : switching between played and paused status

ACL : Designated orgRole users

GUI :

|  |
| --- |
| SSI 🡺Approve |

Purpose : Approve selected (ACC #1)flagged Item and put it in pending status and triggering an action with “sendToOrg” title and down-graded ACC(#2) and

pending status

Outcomes :organization needs to review the content over again with down-graded ACC to make sure selected content is Adult-free-material.

Next Possible Actions : Org will decide to approve/delete/reject the content

ACL : SystemRole users

GUI :

|  |
| --- |
| Back-End 🡺Reject |

Purpose : putting content in reject status. This status is recoverable to be approved.

Outcomes :

Next Possible Actions :Approve

ACL : Designated org/role users

GUI :

|  |
| --- |
| SSI 🡺Reject |

Purpose : SSI Rejects the (#1 ACC) flagged content and change content status to deleted

Outcomes : in “deleted” status , content won’t be able to recovered to any other status and basically “deleted” status is permanent. After a content-status is changed to “Deleted” all attached files will be moved to “(770)inactive/” directory and this state cannot be changed to any other state.at this stage responsible organization will be charged of some fees beacause of not choosing the [delete] action in 24 hr deadline. note that orgnanization can not to delete content at thos point if sure the content has no #1 ACC issue.

Next Possible Actions : NONE

ACL : SystemRole Users

GUI :

|  |
| --- |
| Back-End 🡺Delete |

Purpose : change Content status to “deleted” Outcomes

Outcomes : in “deleted” status, content won’t be able to be recovered to any other status and basically “deleted” status is permanent. After content-status is changed to “Deleted” all attached files will be moved to “(770)inactive/” directory and this state cannot be changed to any other state.

Next Possible Actions : NONE

ACL : Designated Org/Role users

GUI :

|  |
| --- |
| Back-End 🡺Play/Pause |

Purpose : making the content (un)viewable.

Outcomes : both status represent Approved status.

Next Possible Actions : play/pause/delete/reject

ACL :

GUI :

|  |
| --- |
| Back-End 🡺View History |

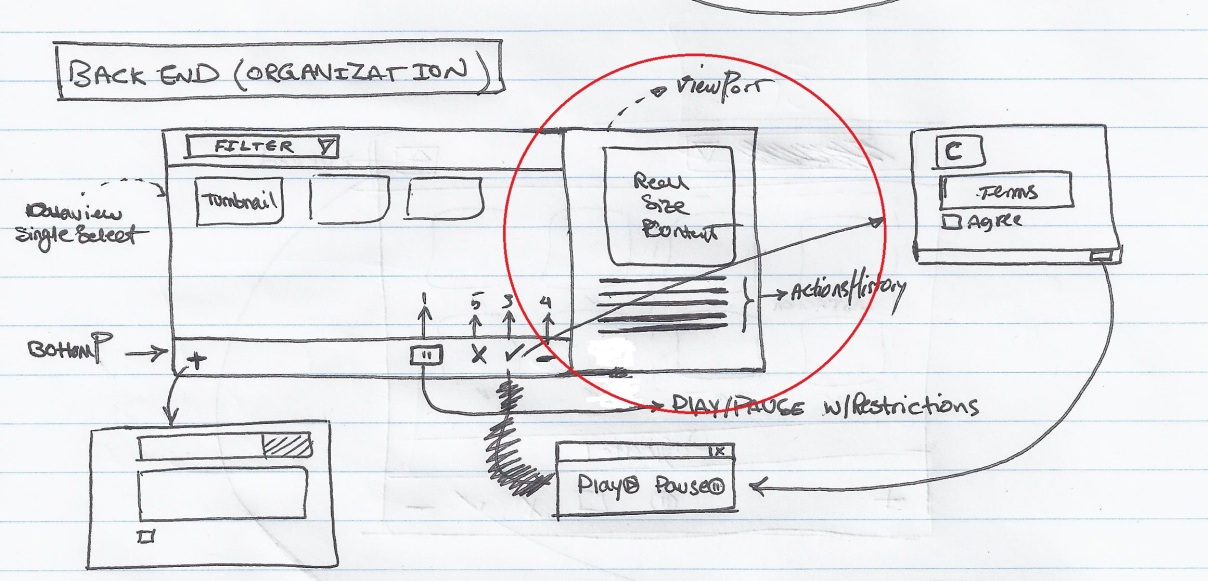
Purpose :

Outcomes :

Next Possible Actions :

ACL :

GUI :



|  |
| --- |
| Front-End 🡺Flag contents |

Purpose : user flags the content which includes adult material (ACC).This selection might be #1(XXX) or others based on the severity of the selection, a ticket will be opened for SSI or just organization to be verified. (user decision might not be correct or approved)

Outcomes : If any content flagged no matter of ***Adult Content Classification(ACC)*** type, that content won’t be visible since the action is made.

Next Possible Actions : Approval or Rejection decision made by Org

ACL : Designated OrgRole Users

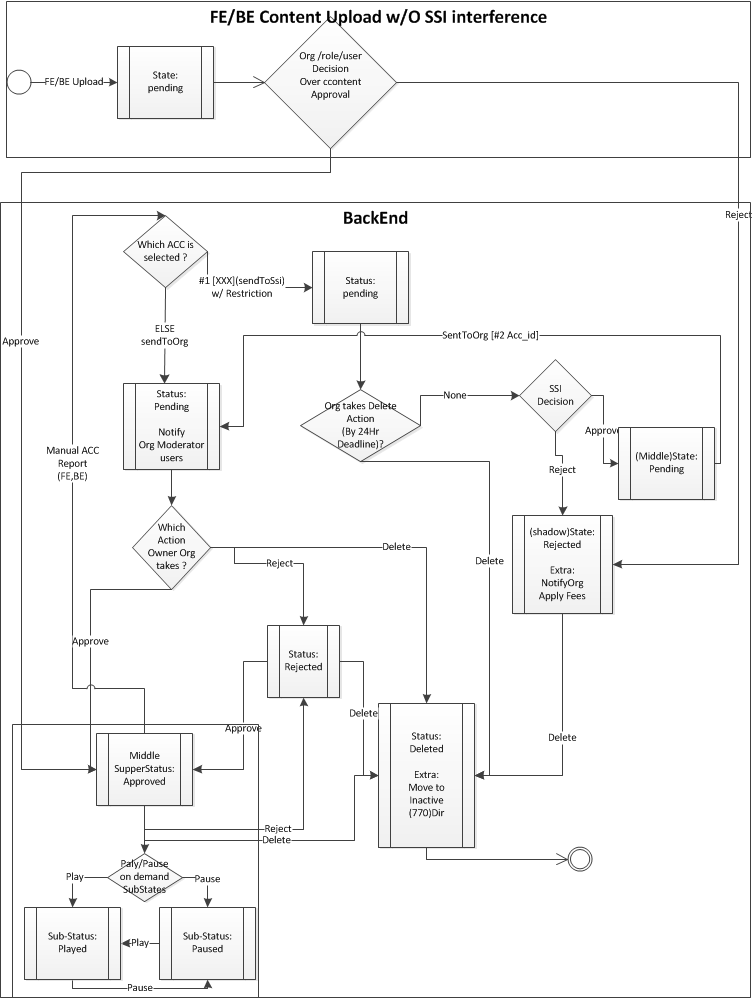
GUI :

**Table 1:** Visualize the next possible action based on last-action-made over the content and content status after last action.

(-) in means “this situation won’t occur”.

bottom docked items: (1) Pause/Play (2)History OverView (3)Approve (4)Reject (5)Delete

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Target-Status** |  | **Last Action Made** | | | | | | | |
| 1-Upload | 2-Approve | 5-Reject | 6-Delete | 4-Play | 3-Pause | 7-SendToOrg | 8-SendToSSI |
| 1-Pending | Buttons  Viewable  After  Action  Made | 3,4,5 | 4 | - | - | - | - | 3,4,5 | 5 |
| 2-Played(Approved) | - | - | - | - | - | 1,6,4,5 | -(FE) | -(FE) |
| 3-Paused(Approved) | - | - | - | - | 1,6,4,5 | - | -(FE) | -(FE) |
| 4-Deleted | - | - |  |  | - | - | - | - |
| 5-Rejected | - | - | 3,5 | - | - | - | - | - |

**Figure 1:[semi-state chart Diagram]**  


**[Implementation stand-point]**

**[Clientside]**

***Back-End :***

Main Module : Using Extjs customizable viewPort Panel to embed Module1 and Module2.

Module1 : (Left Panel )Using Extjs dataview with customized template(TPL) to list contents based on selected status Filter.

Module2 : (Right Panel)customizable Action History panel.

***Front-End:***

Html based gallery list and possibly using Jquery for Adding Effects.

**[Serverside]**

All functions required to implement operations would be located in backend and will be shared with front-end and will be accessed through httprequest.

If any function needed to be implemented for just front-end, still will be located in back-end and will be accessed through httprequest.

**[database Functions/tables/Triggers/views [Changes]]**

**Tables:**

|  |  |  |  |
| --- | --- | --- | --- |
| ***lu\_acc***  1-xxx      2-inappropriate 3-Agging issues  Notes:    if content reported with  #1 xxx  then tickect will be oppened as sendToSSI  else then tickect will be oppened as sendToORG | ***lu\_content\_action [possible actions taken over content]***  1-upload  2-Approve  3-pause 4-play  5-reject 6-delete 7-sendToOrg 8-sendToSsi | ***lu\_content\_type (type\_id,type\_name)*** ***[currently-Used or future-use content Type]***  1-Photo 2-Movie | ***lu\_action\_status(type\_id,type\_name)***  1-Pending  2-Paused 3-Played  4-deleted 5-rejected |
| ***Content***  content\_id,  content\_name , content\_type\_id , file\_name , owned\_by  , description  *Notes:* file\_name will be server-side Auto-generated *[UniqueId . filetype]* | ***Content\_actions\_history***  ID, content\_id , acc\_id , *content\_action\_id, content\_status\_id* , user\_id,role\_id,org\_id,description  *Notes:* 1-{ user\_id,role\_id,org\_id } combination ~ action Taker 2-content\_action\_id, content\_status\_id  combination means after content\_action happens, content\_status will be changed to [new]content\_status\_id 3-acc\_id might be null just for *upload* Content\_Action 4-acc\_id may stay the same for consecutive actions over content or be changed. |  |  |

***DB Functions:***

|  |  |
| --- | --- |
| ***1-Create\_content*** (  content\_name , content\_type\_id , file\_name , owned\_by  , description  ) | ***2-Create\_content\_action\_ticket*** (  content\_id , acc\_id=null , content\_action\_id, content\_status\_id , user\_id,role\_id,org\_id,description ) |

**DB Views:**

|  |
| --- |
| **View\_content\_lastaction**  Lists last action taken over each content ,Which simplifies and accelerates table joints . |

**[input/output structure]**

**Directory Structure**