

Media blogging for Roller

Revised Project Plan

By

Ganesh Mathrubootham

Tanuja Varkanthe

November 11, 2008

Mr. David Johnson

1. Project Research Summary

State of the art summary

Blogging or weblogging is a phenomenon that has allowed individuals to effectively use Internet as a media to express their views on anything under the sun. Roller is full-featured, multi-user blog server software implemented as a web application running on any Java EE server. Roller supports the key features of any basic blog server and currently powers many blogging communities in the web that include those of IBM and SUN. As blogs exploded in the last few years and became a key source of information for web surfers around the world, the form of blogging has undergone drastic changes. What started out as an online journal using plain text, has now transformed significantly to use other advanced tools of communication [4]. People started to use pictures as a medium of communication in their blogs. Audio and video podcasts, which allow bloggers to express their thoughts using digital media that can be syndicated and downloaded automatically like normal text blogs have also become extremely popular [3].

In this context, Roller currently lacks support for the latest blogging features. Roller does allow users to upload any type of content to their blogs and include that content on blog entries as images or podcasts, but lacks tools to make media blogging a seamless experience for bloggers. Interface to manage uploaded files is not sortable and not pagable. Once the user has uploaded a file, which could be an image or a podcast, he needs to explicitly cut and paste the URL into his blog post. Overall, Roller's support for media blogs is currently limited to basic file upload functionality, which is cumbersome to use for creating media blogs. This project will revamp the existing file upload interface to incorporate powerful media blogging features into Roller [1]. Also, successful media management websites such as Flickr and YouTube are driven by a public media library, offering different ways for users to search and locate the content of their like. This feature incorporated into a blog server can make it very powerful and we intend to do that as part of this project. CoolIris website provides another exciting dimension to media world. It augments the power of traditional web browser providing a three dimensional view of media files [2]. Apparently, Roller with its limited support for media blogging is currently not integrated with CoolIris. As we introduce rich media blogging features, such exciting integrations become possible.

References:

1. Evans, S. Follow up to How to reorder Media Library categories. (2005, May). Retrieved July 5, 2006, from <http://wordpress.com/tag/media-library/>
Wordpress is a hosted blog publishing platform that is widely used by many bloggers. This article helped us gain a high-level idea about using media library in a blogging environment. Media library offered by Wordpress is pretty limited and we plan to build on the idea to implement a full-fledged media library for Roller.
2. Web site: <http://www.cooliris.com>
This website details the features of CoolIris. CoolIris, when installed as a plug-in for any browser, has the ability to transform the browser into full screen 3D experience. It is possible to enable any web site for CoolIris by having rich media items embedded in the blog articles and by presenting media files in a separate feed. This is one of the key features that we plan to implement in our project.
3. Stone, B. (2002). **Blogging: Genius Strategies for Instant Web Content**. New Riders Publication.
This is an excellent tutorial for new bloggers, and includes many advanced techniques for veteran bloggers. This book features hands-on tutorials for building a blog, adding a user based commenting system, adding team members, syndicating with JavaScript, adding searches to a site, and much more. This book helped us understand the concept of blogging and helped us explore media blogging further.
4. Kline, D., & Burstein, D., & Keijzer, A., & Berger, P. (2005) **Blog! : How the Newest Media Revolution is Changing Politics, Business, and Culture**. CDS Publication.
This book includes articles and interviews from commentators and professional bloggers. It delineates the current blogging trend and the future direction of blogging. This book helped us understand the importance of media blogging and allowed us to visualize detailed requirements for our project by placing ourselves in the shoes of professional media bloggers.
5. O'Reilly, T. What is Web 2.0. (2005). Retrieved from <http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html?page=1>
This article by Tim O Reilly captures the changing trends in the use of web technology and explains the key principles behind Web 2.0. This article helped us envisage how to go about defining media blogging features for Roller. We also intend to use this knowledge for coming up with a rich and user-friendly interface for our project.

2. Requirements/Dependencies/Deliverables

A) Project Requirements:

Media file management in Roller

- Ability to specify more information about the file being uploaded.
 - Capture these metadata - file name, description, copyright text, tags.
- Enhanced view of uploaded files.
 - Hierarchical display (show all folders with expand/collapse buttons, no thumbnail)
 - Tabular display (shows just one folder at a time, with links for other folders)
 - Display thumbnail for images
 - Display thumbnail for videos (with play/pause buttons)
 - Display icon for other kind of files.
 - Have pagination
 - Have the ability to switch between hierarchical and tabular display.
- Manage uploaded files
 - Move files across folders.
 - Delete files.
 - Edit metadata (add/modify tags, modify file name, modify description, modify copy right text)
- Ability to specify whether the uploaded file can be included in common media library
 - Ones made "public" will be made available as a library to other bloggers. (This will lead to a whole new functionality of media library which can be browsed by any other blogger)
- Ability to sort from tabular display.
 - Have ability to sort by name, date uploaded and file type
- Ability to filter file lists by
 - file name
 - file type (audio, video etc)
 - file extension
 - uploaded date range
 - Size
 - tag

- Ability to create a blog post out of uploaded files.
 - After a file is uploaded, provide option to create a blog post out of it.
 - Ability to select one or more files from the list and create a blog post out of the selected files.

Media blog posting in Roller

- Ability to add media from uploaded files
 - Browse and choose uploaded file
 - Allow search by file name, description and tag.
 - If it is an image, specify dimension and layout to be used in the blog post.
- Ability to add media directly (upload file as part of this)
 - Browse the file to be uploaded
 - Input file name, description and tags
 - If it is an image, specify dimension and layout to be used in the blog post.
- Ability to add media directly, pointing to an external source (Youtube video, Flickr image, images.google.com picture etc)
- Ability to add media from media library (library built from all files uploaded and shared by user community)
 - Search by tag, name and description.
 - Pick the selected one to be displayed.
 - If it is an image, specify dimension and layout to be used in the blog post.
- Other enhancements to "create post" interface
 - Choose text in the post and insert links easily.

Feeds for media files in Roller

- ATOM publishing for media files uploaded - change the structure to be hierarchical and add the metadata information (name, description, tags, copyright text) for each file.
- Support ATOM feed for "latest images that you uploaded".

B) Project dependencies:

- Design for final implementation (UI design, data model and class design) should be approved by Roller development team.
- Code will have to be reviewed by roller development team and then committed to trunk once the quality requirements are met.

C) Project Deliverables

- Proposal to Roller dev community delivered as a Wiki page.
 - Detailed requirements
 - Design
 - User Interface Design (wireframes)
 - Data model
 - Class design
- Source code either delivered as a SVN patch or committed to a separate branch.(mode to be determined)
- Unit tests for business layer.

3. Architecture

Following is the high level view of system architecture:

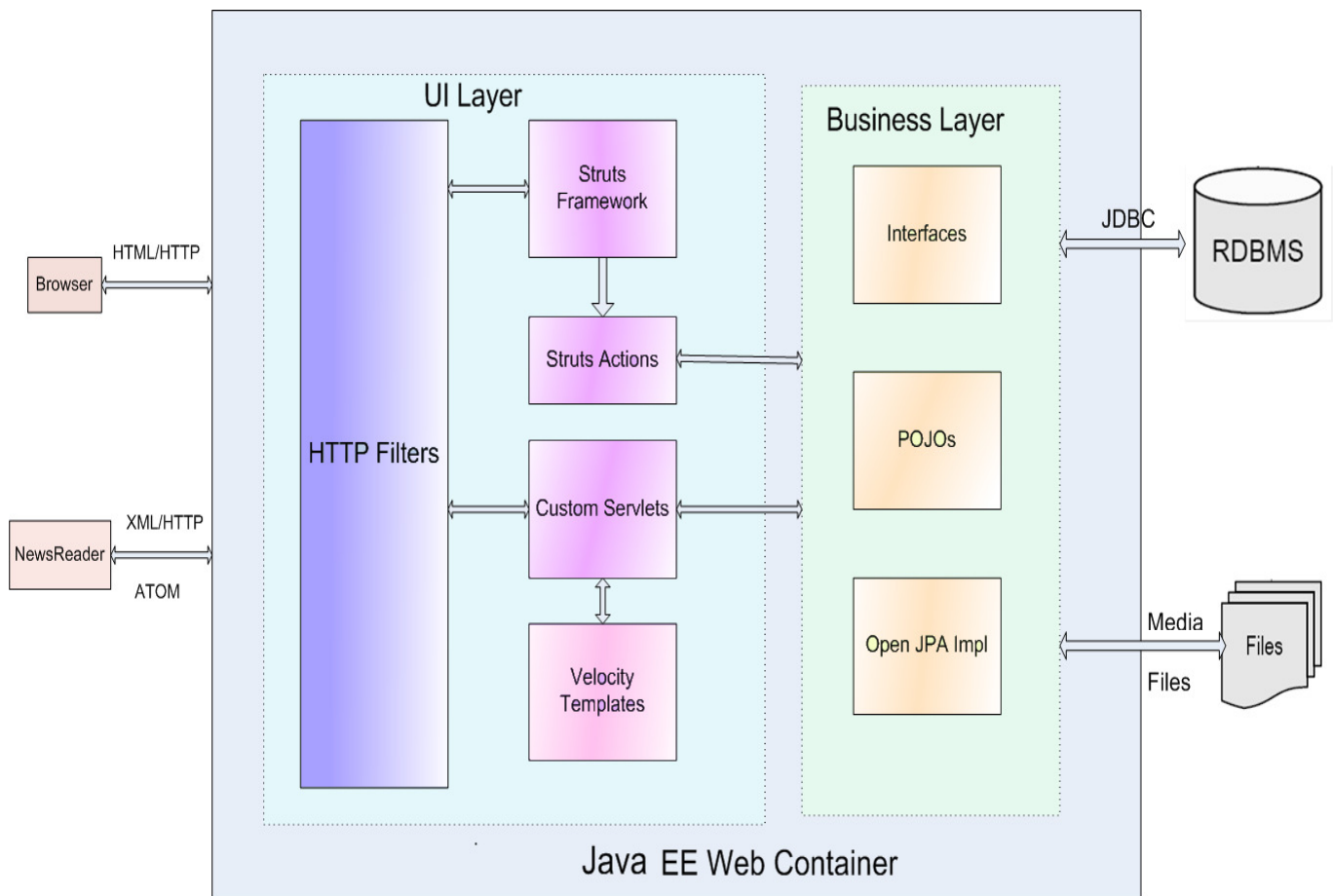


Figure 1. System Architecture Diagram

Architecture Description:

This project will follow a web-based N-tier architecture. The requests from web browser will be stateless calls over http. A Java EE compliant web container will handle the requests from browser. Web container, in most cases, will generate a plain HTML response, which will be used to render a full page on browser. For AJAX-based http requests, the response will be data in XML or JSON format. Alternatively, the requests can also come from any AtomPub clients for editing and publishing media content. These requests are made using Atom Publishing Protocol, an application-level protocol that is built on top of http. Web container responds with a XML message that conforms to AtomPub schema for media content.

The web layer on the container heavily uses HTTP filters to intercept requests and perform tasks that are typically common across a group of requests. Servlet implementation from Struts 2 framework handles requests from browser. Struts framework routes the call to application action classes, which invoke business layer classes to process the request. After the request is processed, Struts framework forwards control to view, which is implemented as JSP that heavily uses JSTL and Struts tags to render the response.

The requests from AtomPub clients go through HTTP filters as well, but are processed by application Servlets. These Servlets invoke business layer to generate data and then use a velocity template engine to generate the ATOM/XML response.

Business layer is responsible for business validations, executing the business logic and interfacing with data store. Business layer will have manager interfaces and POJOs tied to persistent store using configuration. A JPA implementation will be used for persistence, with an underlying JDBC driver. This is a vendor-agnostic architecture that allows any database (such as Oracle, MySQL or SQL Server) to be used for persistence. For development and unit testing, MySQL 6.0 database server will be used. Media files themselves will be stored in file system and database will store the rest of the relational data associated with media file management and pointer to media files in the file system. This architecture is in line with the current Roller architecture.

4. Design

A) Mockup Diagrams:

logged in as user - editing weblog user Front Page | Main Menu

File Uploads

[Create & Edit](#) | [Design](#) | [Preferences](#)

[New Entry](#) | [Entries](#) | [Comments](#) | [Categories](#) | [Bookmarks](#) | **File Uploads** | [Referrers](#)

Add media file

File location

Name

A short description of file

Copyright statement (if any)

Tags (separated by spaces)

Place file under

Include in gallery?

Actions

- [Add media file](#)
- [View uploaded files](#)
- [Search uploaded files](#)

Powered by Apache Roller Weblogger Version 4.1-dev | [Report an Issue](#) | [User Guide](#) | [Mailing Lists](#)



Figure 1: Add media file

logged in as user - editing weblog user Front Page | Main Menu


File Uploads

[Create & Edit](#) | [Design](#) | [Preferences](#)

[New Entry](#) | [Entries](#) | [Comments](#) | [Categories](#) | [Bookmarks](#) | **File Uploads** | [Referrers](#)

Add media file

- test.jpg uploaded succesfully



<http://roller.xyz.com/roller/user/resource/test.jpg>

[Create a blog post out of test.jpg](#)

[Add another media file](#)

Actions

- [Add media file](#)
- [View uploaded files](#)
- [Search uploaded files](#)

Powered by Apache Roller Weblogger Version 4.1-dev | [Report an Issue](#) | [User Guide](#) | [Mailing Lists](#)

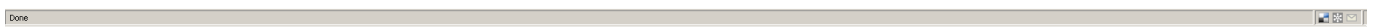


Figure 2: Media File added successfully

File Uploads

Create & Edit | Design | Preferences

New Entry | Entries | Comments | Categories | Bookmarks | **File Uploads** | Referrers

View uploaded files

Tabular view | Hierarchical view

Sort by Name

<<First <Previous 1 2 3 4 5 6 Next> Last>>



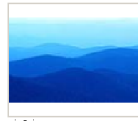
Vegas trip



Grand canyon trip



New york trip videos



pic0.jpg

Edit More..



pic1.jpg

Edit More..

- Delete
- Create post
- Include in gallery



pic2.jpg

Edit More..



pic3.jpg

Edit More..



match.mpg

Edit More..

New Directory Create

Delete selected | Move selected | /Vegas trip

Actions

- Add media file
- View uploaded files
- Search uploaded files

Figure 3: View Media Files – Tabular View

File Uploads

Create & Edit | Design | Preferences

New Entry | Entries | Comments | Categories | Bookmarks | **File Uploads** | Referrers

View Uploaded Files

Tabular View | Hierarchical View

- jit
 - doc
 - spacetre
 - styles
 - Style.bmp Edit More..
 - videos
 - index
 - files
 - Spacetre.jpg Edit More..
 - index.html
 - hypertree
 - treemap
 - rgraph
 - examples
 - compressed
 - source
 - Treemap.gif Edit More..
 - Delete
 - Create post
 - Include in gallery

Delete selected

Actions

- Add media file
- View uploaded files
- Search uploaded files

Figure 4: View Media Files – Hierarchical View

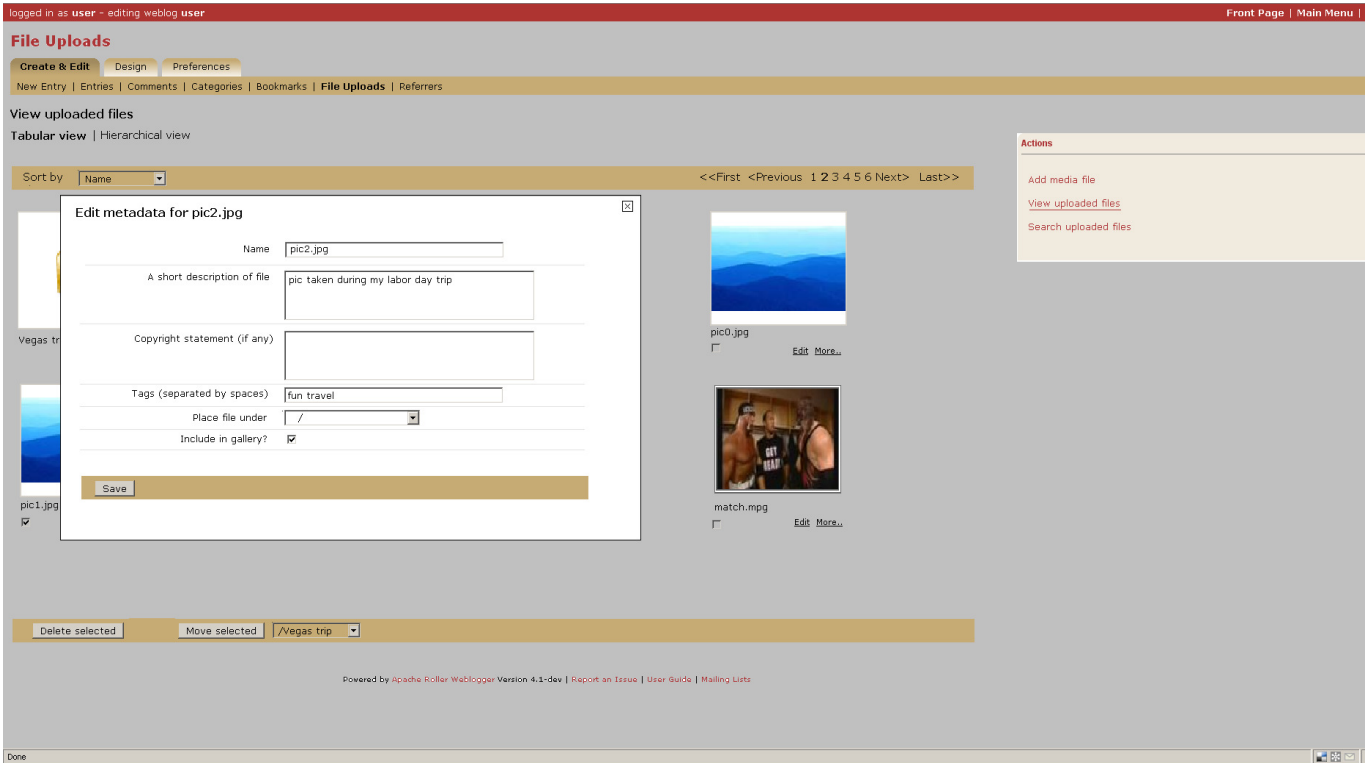


Figure 5: Edit file metadata

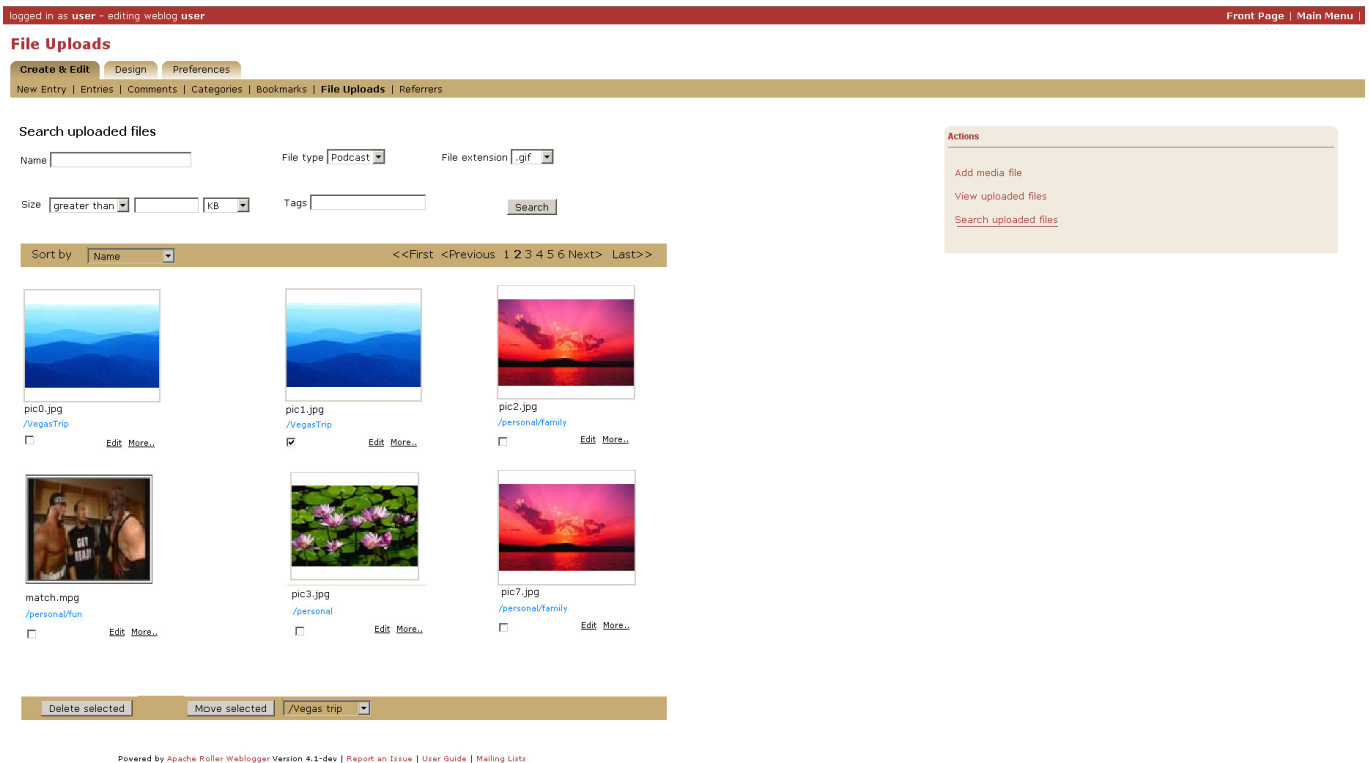


Figure 6: Search uploaded files

New Entry

Create & Edit | Design | Preferences

[New Entry](#) | [Entries](#) | [Comments](#) | [Categories](#) | [Bookmarks](#) | [File Uploads](#) | [Referrers](#)

Create a new entry in weblog **tanuja**

Title
Status **Not saved**
Category **General** ▾
Tags

Content [Add media](#) [Attach uploaded file](#) [Add media from URL](#)

↓ ↑

Summary (optional)

↓ ↑

+ Plugins To Apply

+ Advanced Settings

Comments
None

Pending Entries
None

Recent Drafts
None

Recent Entries
None

Scheduled Entries
None

Figure 7: Create post

Create & Edit | Design | Preferences

[New Entry](#) | [Entries](#) | [Comments](#) | [Categories](#) | [Bookmarks](#) | [File Uploads](#) | [Referrers](#)

Create a new entry in weblog **user**

Title
Status **Not saved**
Category **Java** ▾
Tags

Content [Add new](#) | [Insert from uploaded file](#) | [Insert URL](#)

File location

Name

A short description of file

Copyright statement (if any)

Tags (separated by spaces)

Place file under ▾

Include in gallery?

Summary (optional)

↓ ↑

+ Plugins To Apply

Comments
None

Pending Entries
None

Recent Drafts
 jdk 1.6
 jdk 1.5

Recent Entries
None

Scheduled Entries
None

Figure 8: Create post - add media

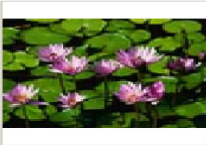
New Entry

Create & Edit | [Design](#) | [Preferences](#)

[New Entry](#) | [Entries](#) | [Comments](#) | [Categories](#) | [Bookmarks](#) | [File Uploads](#) | [Referrers](#)

Create a new entry in weblog **tanuja**

Choose layout for image



Link URL: <http://roller.xyz.com/roller/user/resource/test.jpg>

Alignment: None Left Center Right

Size:
Width Height

[d_media from URL](#)

Comments
None

Pending Entries
None

Recent Drafts
None

Recent Entries
None

Scheduled Entries
None

Summary (optional)

+ [Plugins To Apply](#)

+ [Advanced Settings](#)

Figure 9: Choose image layout


New Entry

Create & Edit Design Preferences

New Entry | Entries | Comments | Categories | Bookmarks | File Uploads | Referrers

Create a new entry in weblog tanuja

Choose layout for video



Link URL: <http://roller.xyz.com/roller/tanuja/video1.swf>

Alignment: None Left Center Right

Embed media from URL

- Comments**
None
- Pending Entries**
None
- Recent Drafts**
None
- Recent Entries**
None
- Scheduled Entries**
None

Summary (optional)

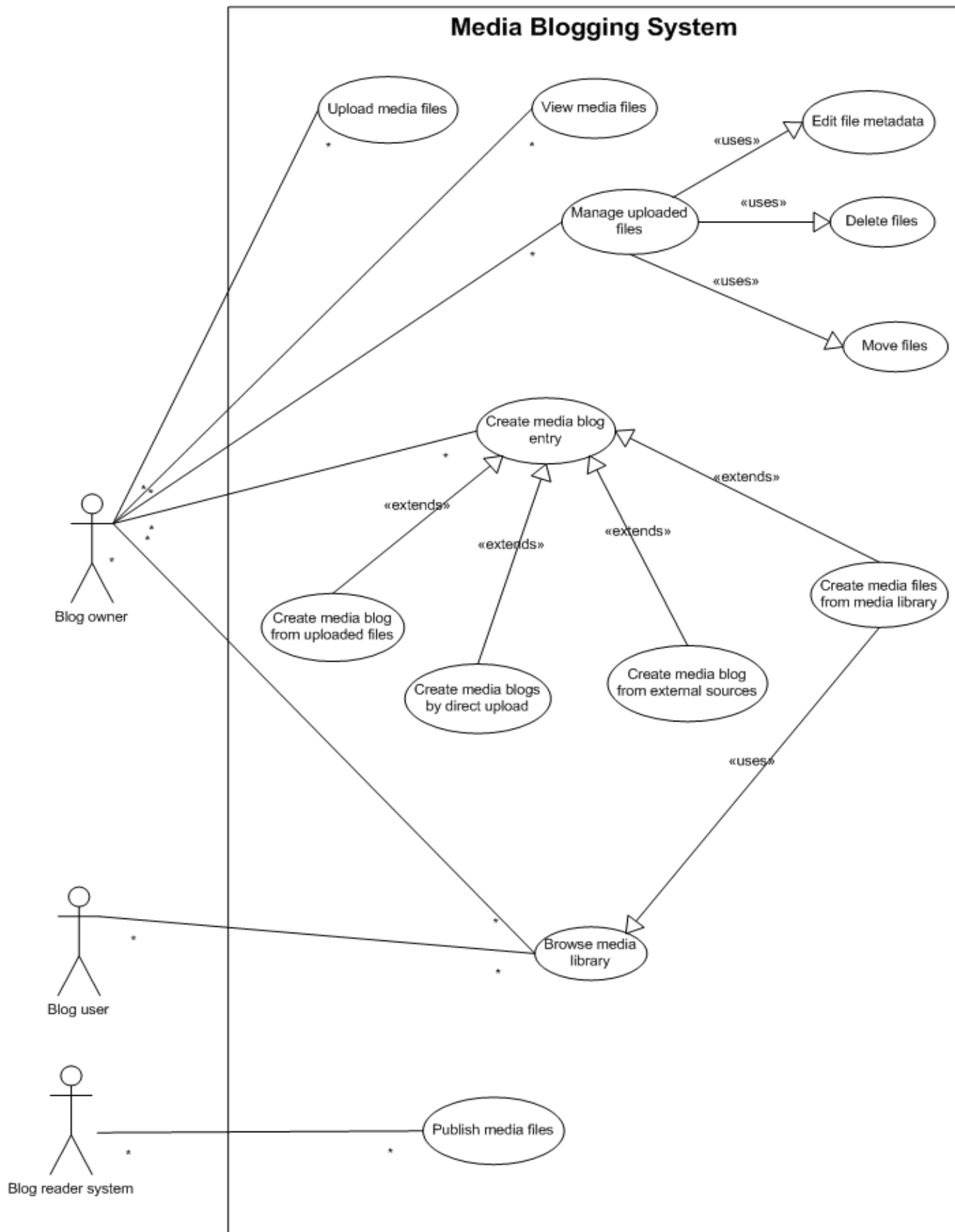


+ Plugins To Apply

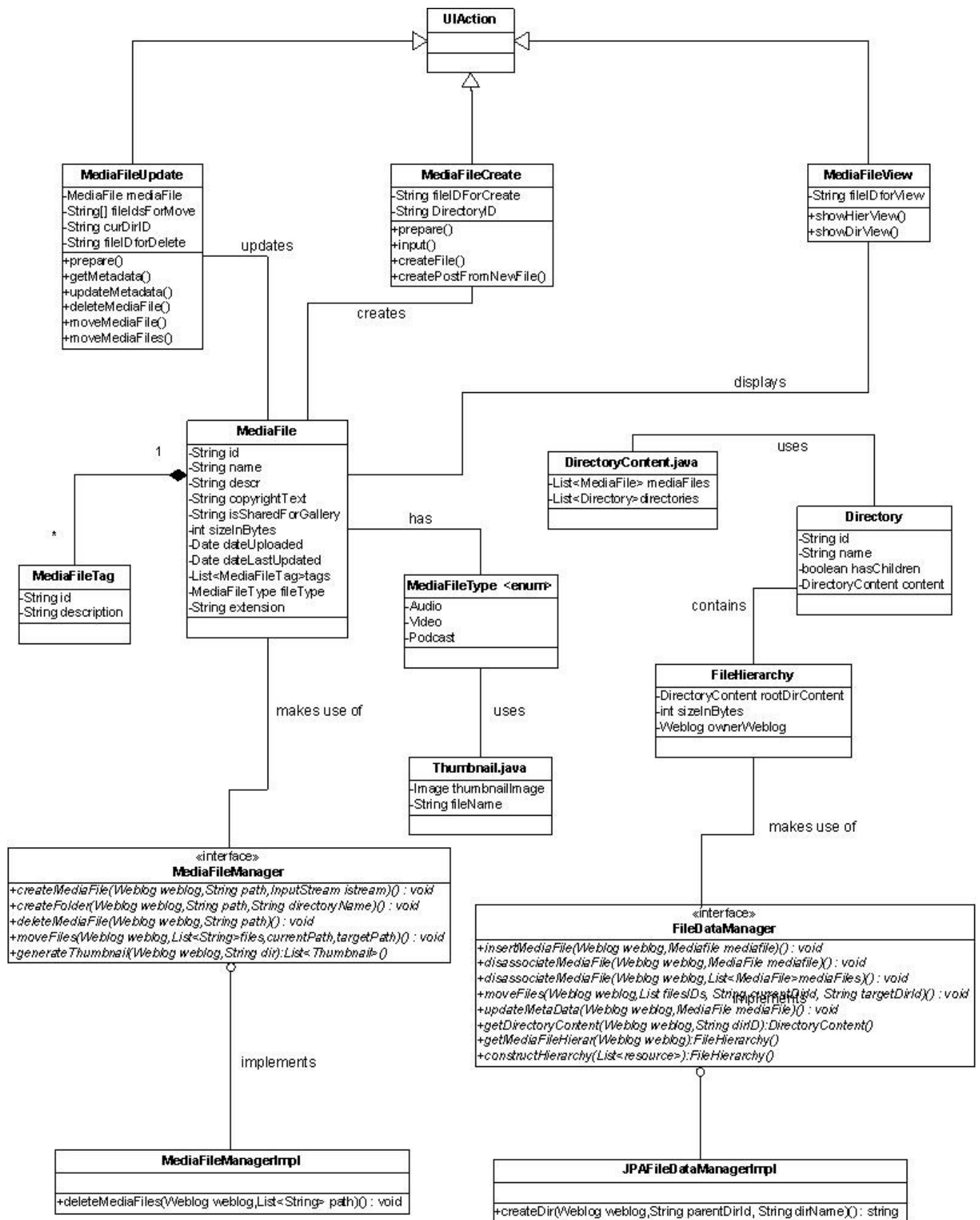
+ Advanced Settings

Figure 10: Choose layout for video

B) Use Case Diagram:

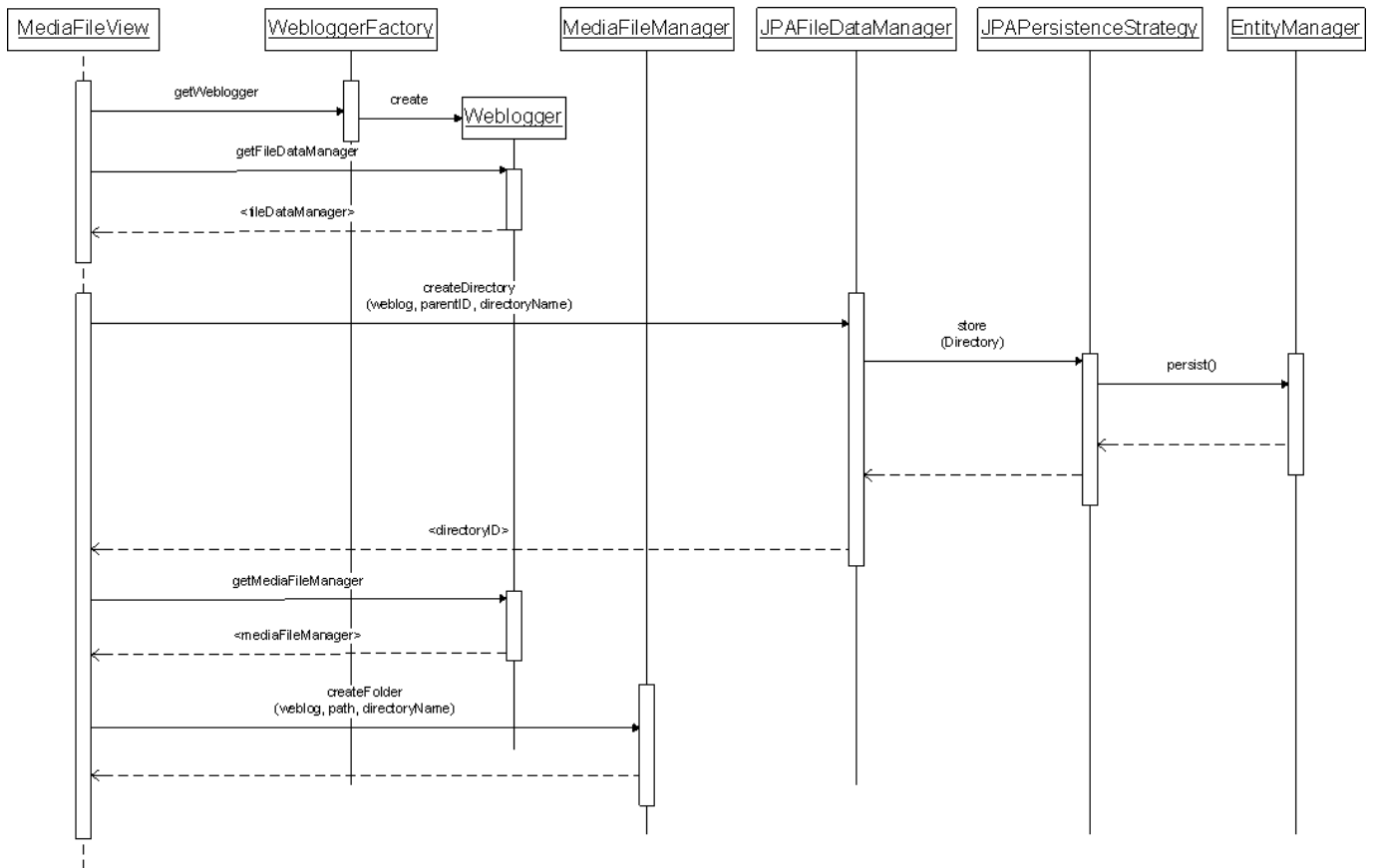


C) Class diagram – high-level static structure.

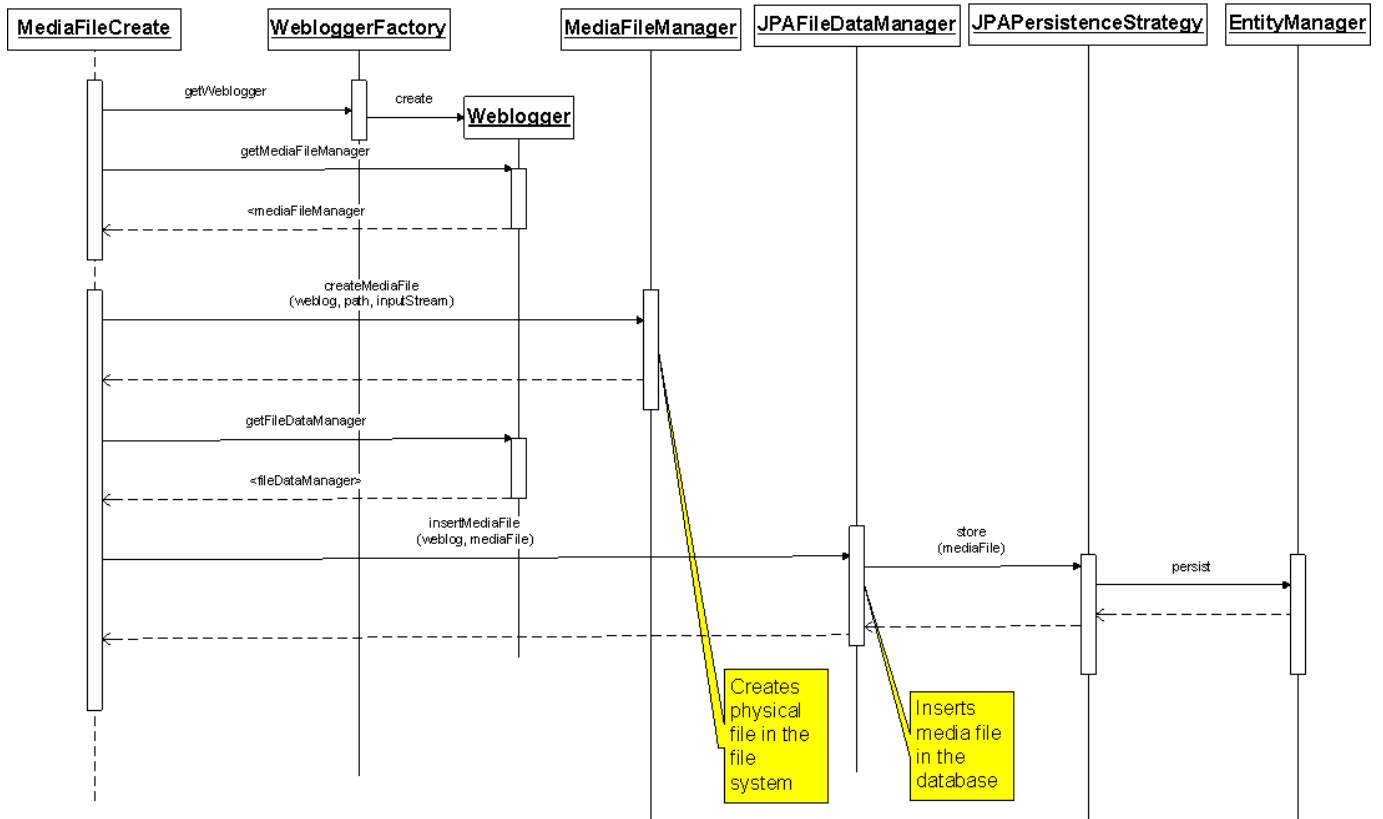


D) Sequence Diagrams

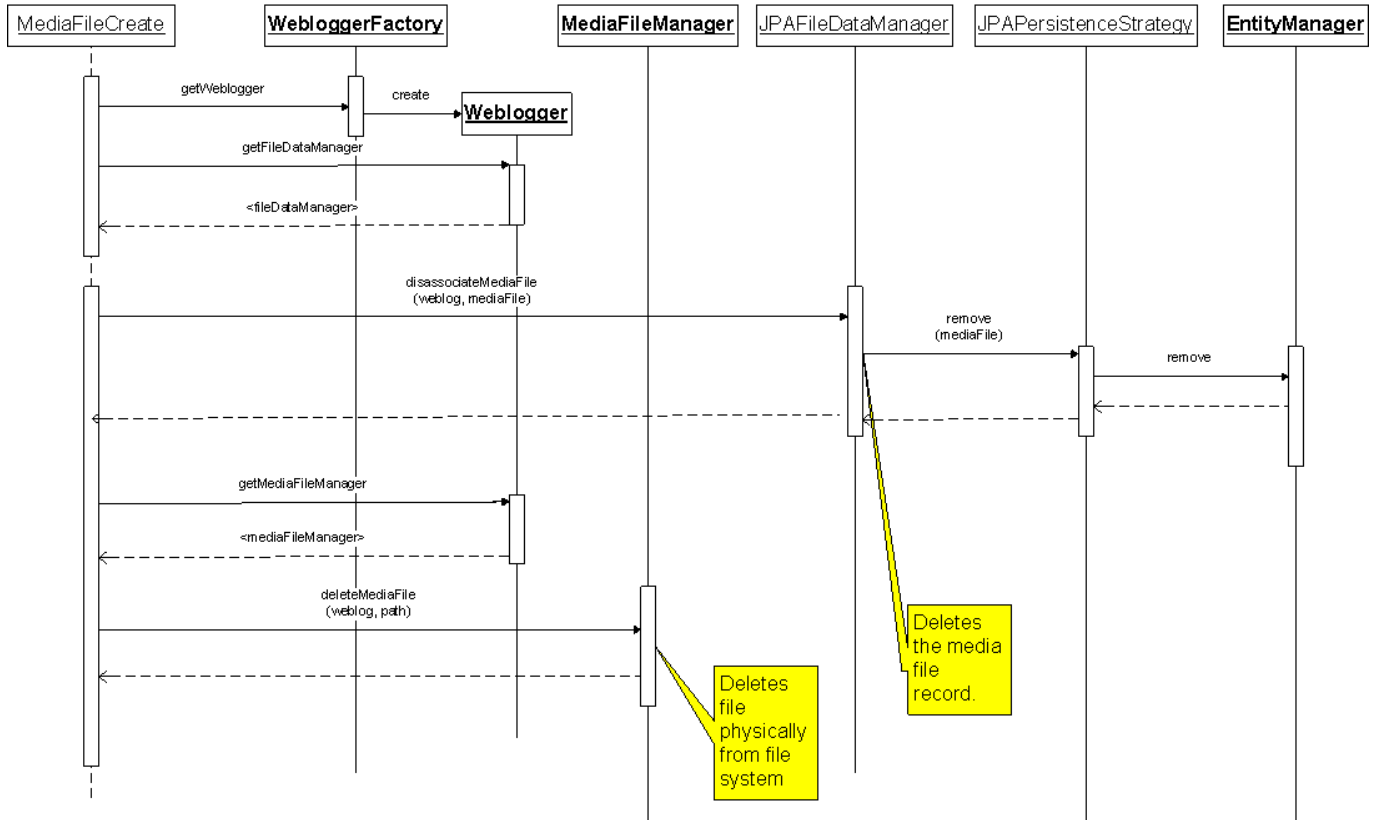
Sequence diagram – Create directory



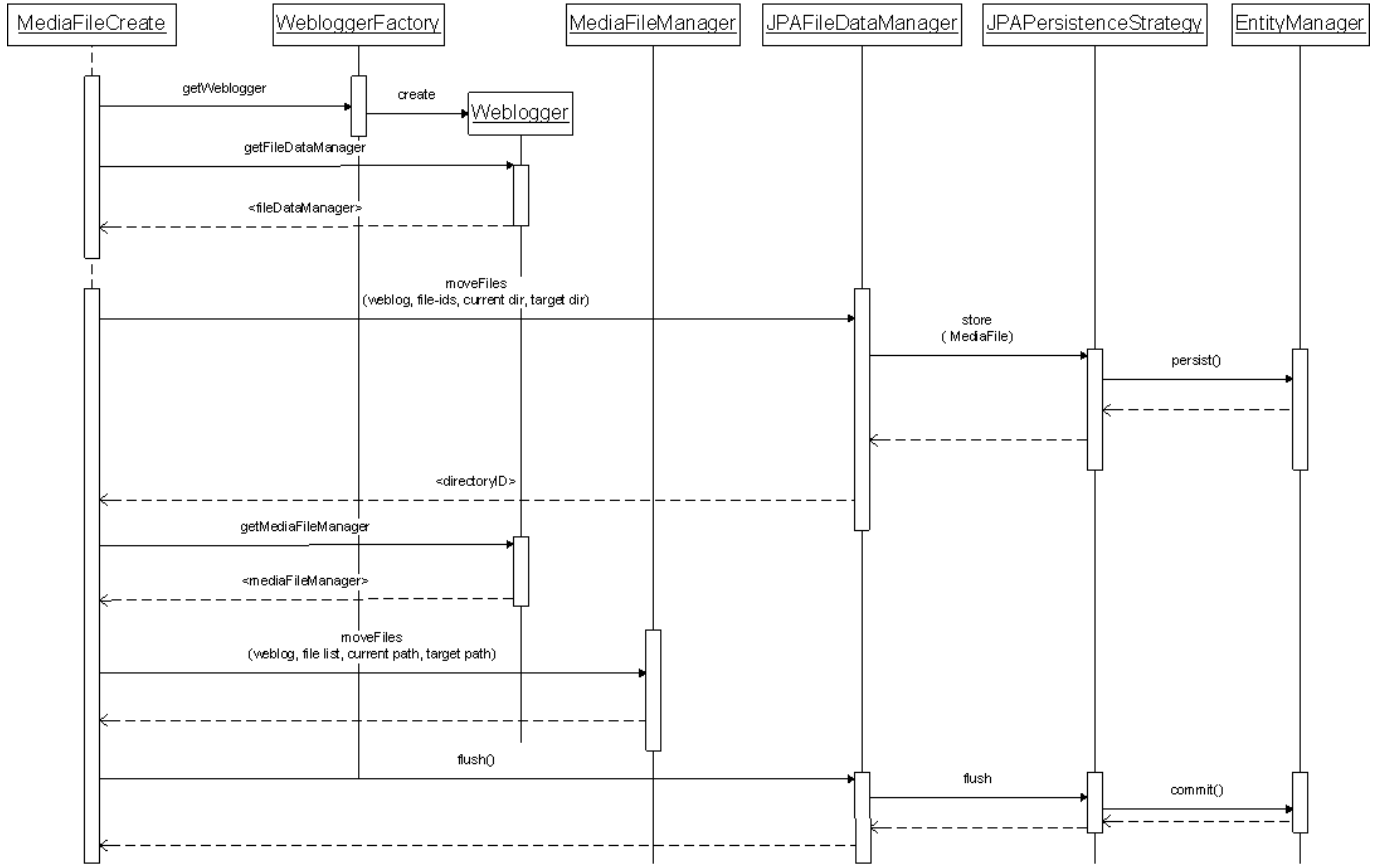
Sequence diagram – Upload media file



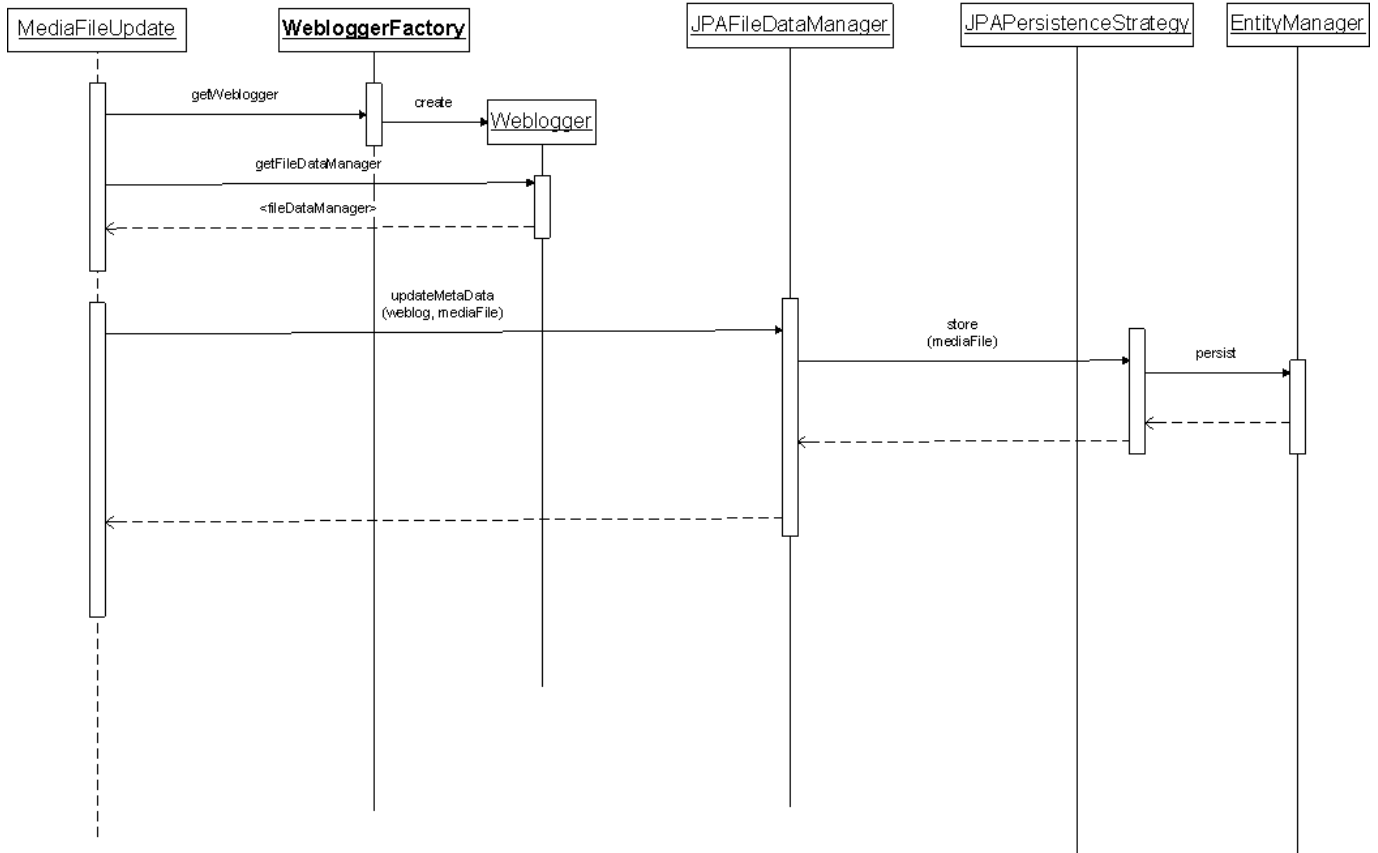
Sequence diagram – delete media file



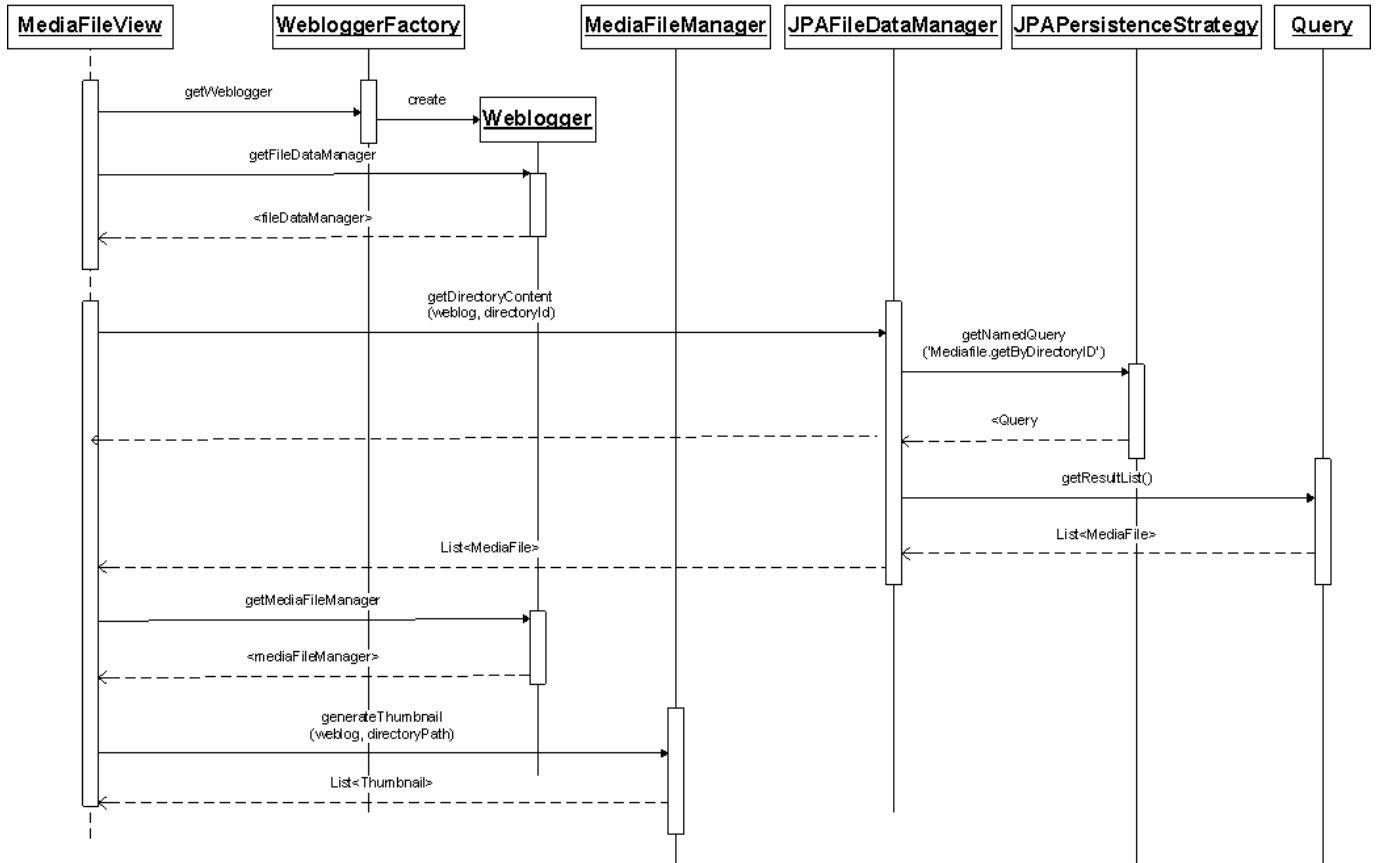
Sequence diagram – move files



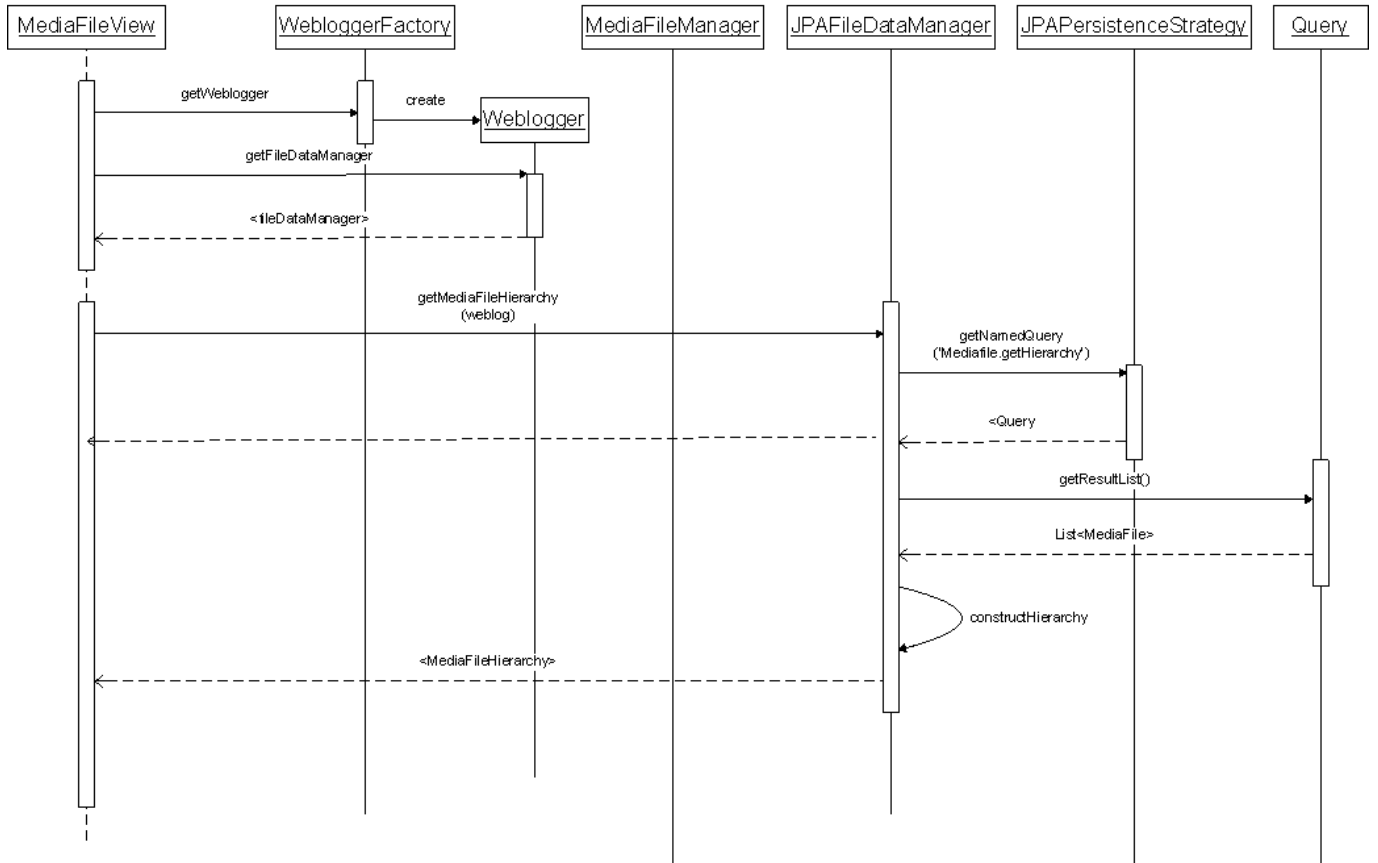
Sequence diagram – Update media file metadata



Sequence diagram – View media directory



Sequence diagram – View media directory



5. Schedule

ID	Task Name	Duration	Start	Finish	Predecessors	% Complete	Resource Names
1	Requirements phase	27 days?	Mon 8/25/08	Tue 9/30/08		100%	
2	Understand the current state, architecture	27 days?	Mon 8/25/08	Tue 9/30/08		100%	
3	Get access to dev list, wiki, issue tracker	2 days?	Mon 8/25/08	Tue 8/26/08		100%	All
4	Identify and install tools required for the project (Apache Tomcat, My SQL, database driver)	1 day?	Wed 8/27/08	Wed 8/27/08	3	100%	All
5	Download current source code and configure the application locally	3 days?	Thu 8/28/08	Mon 9/1/08	4	100%	All
6	Get familiarized with roler environment and source code.	21 days?	Tue 9/2/08	Tue 9/30/08	5	100%	All
7	Write up proposal	18 days?	Fri 9/5/08	Tue 9/30/08		100%	
8	Research on current state-of-the-art for media blogging	2 days?	Fri 9/5/08	Mon 9/8/08		100%	All
9	Write up the proposal on roler wiki page.	4 days?	Tue 9/9/08	Fri 9/12/08	8	100%	All
10	Incorporate feedback into proposal to finalize high-level requirements	12 days?	Mon 9/15/08	Tue 9/30/08	9	100%	All
11	High-level requirements complete	0 days	Tue 9/30/08	Tue 9/30/08	10	100%	
12	Design phase	42 days?	Wed 10/1/08	Thu 11/27/08		68%	
13	Develop screen mockups and share it with community	23 days?	Wed 10/1/08	Fri 10/31/08		100%	
14	Build screen mockups	14 days?	Wed 10/1/08	Mon 10/20/08		100%	
15	Screens for media file management	9 days?	Wed 10/1/08	Mon 10/13/08	11	100%	Ganesh M
16	Screens for blogging using media files.	7 days?	Fri 10/10/08	Mon 10/20/08		100%	Tanuja V[25%]
17	Incorporate feedback into screen mockups	9 days?	Tue 10/21/08	Fri 10/31/08	14	100%	All[75%]
18	Screen mockup complete	0 days	Fri 10/31/08	Fri 10/31/08	13	100%	
19	Data modeling	19 days?	Mon 10/27/08	Thu 11/20/08		97%	
20	Come up with the list of new tables	7 days?	Mon 10/27/08	Tue 11/4/08		100%	Tanuja V[80%]
21	Identify changes to existing tables	7 days?	Mon 10/27/08	Tue 11/4/08		100%	Tanuja V[80%]
22	Draw ER diagrams and share it with community	3 days	Mon 11/10/08	Wed 11/12/08	20,21	100%	Tanuja V
23	Incorporate feedback to finalize data model	1 day?	Thu 11/20/08	Thu 11/20/08	22	50%	Tanuja V
24	Data model complete	0 days	Thu 11/20/08	Thu 11/20/08	19	0%	
25	Class Design	24 days	Mon 10/27/08	Thu 11/27/08		48%	
26	Identify new classes and methods	7 days	Mon 10/27/08	Tue 11/4/08		75%	Ganesh M[80%]
27	Draw class diagrams and share them with community	7 days	Wed 11/5/08	Thu 11/13/08	26	50%	Ganesh M
28	Draw sequence diagrams and share them with community	7 days	Wed 11/5/08	Thu 11/13/08	26	50%	Ganesh M
29	Incorporate feedback to finalize class design	10 days	Fri 11/14/08	Thu 11/27/08	27,28	25%	Ganesh M
30	Class design complete	0 days	Thu 11/27/08	Thu 11/27/08	25	0%	
31	Identify tools and technologies	16 days	Mon 10/27/08	Mon 11/17/08		47%	
32	Research and finalize JavaScript library for client-side	16 days	Mon 10/27/08	Mon 11/17/08		47%	Tanuja V[75%]
33	Research and finalize libraries for handling media files on server side	16 days	Mon 10/27/08	Mon 11/17/08		47%	Ganesh M
34	Design complete	0 days	Thu 11/27/08	Thu 11/27/08	13,19,25,31	0%	
35	Development phase	119 days	Mon 11/3/08	Thu 4/16/09		5%	
36	Develop HTML and CSS that conform to wireframes	18 days	Mon 11/3/08	Wed 11/26/08	13	29%	
37	UI for adding media file	2 days	Mon 11/3/08	Tue 11/4/08		100%	Tanuja V
38	UI for tabular view of media files with thumbnails	5 days	Wed 11/5/08	Tue 11/11/08	37	100%	Tanuja V
39	UI for hierarchical view of media files	5 days	Wed 11/12/08	Tue 11/19/08	38	0%	Tanuja V
40	UI for editing media file	3 days	Wed 11/19/08	Fri 11/21/08	39	100%	Tanuja V
41	UI for searching media files	2 days	Mon 11/24/08	Tue 11/25/08	40	0%	Tanuja V
42	UI enhancements to create blog post page.	8 days	Mon 11/3/08	Wed 11/12/08		0%	Ganesh M
43	UI to create blog post out of uploaded media files	2 days	Thu 11/13/08	Fri 11/14/08	42	0%	Ganesh M

44		UI to browse media gallery	8 days	Mon 11/17/08	Wed 11/26/08	43	0%	Ganesh M
45		Front-end development complete	0 days	Wed 11/26/08	Wed 11/26/08	36	0%	
46		Develop web tier code	43 days	Wed 12/3/08	Fri 1/30/09		0%	
47		Action classes and JSP associated with adding media file	3 days	Wed 12/3/08	Fri 12/5/08	13	0%	Tanuja V
48		Action classes and JSP associated with tabular view	6 days	Mon 12/8/08	Mon 12/15/08	47	0%	Tanuja V
49		Action classes and JSP associated with hierarchical view	6 days	Tue 12/16/08	Tue 12/23/08	48	0%	Tanuja V
50		Action classes and JSP associated with editing media file	2 days	Mon 1/19/09	Tue 1/20/09		0%	Tanuja V
51		Action classes and JSP associated with search media files	3 days	Wed 12/3/08	Fri 12/5/08	13	0%	Ganesh M
52		Action classes and JSP associated with enhancements to create blog post.	5 days	Mon 12/8/08	Fri 12/12/08	51	0%	Ganesh M
53		Action classes and JSP associated with media gallery	8 days	Mon 12/15/08	Wed 12/24/08	52	0%	Ganesh M
54		Servlet and velocity template associated with ATOM publishing for media files	5 days	Mon 1/19/09	Fri 1/23/09		0%	Ganesh M
55		Servlet and velocity template associated with ATOM feeds for media files.	5 days	Mon 1/26/09	Fri 1/30/09	54	0%	Ganesh M
56		Develop business layer code	21 days	Mon 1/19/09	Mon 2/16/09		0%	
57		Create media file	2 days	Mon 1/19/09	Tue 1/20/09		0%	Ganesh M
58		Update media file	3 days	Wed 1/21/09	Fri 1/23/09	57	0%	Ganesh M
59		Process media file to generate thumbnails etc.	8 days	Mon 1/26/09	Wed 2/4/09	58	0%	Ganesh M
60		Search uploaded files	4 days	Thu 2/5/09	Tue 2/10/09	59	0%	Ganesh M
61		Fetch uploaded files by directory (for serving tabular and hierarchical views)	4 days	Wed 2/11/09	Mon 2/16/09	60	0%	Ganesh M
62		Fetch contents for media gallery	5 days	Wed 1/21/09	Tue 1/27/09		0%	Tanuja V
63		Fetch contents for ATOM publishing	5 days	Wed 1/28/09	Tue 2/3/09	62	0%	Tanuja V
64		Fetch contents for ATOM feeds	5 days	Wed 2/4/09	Tue 2/10/09	63	0%	Tanuja V
65		Coding complete	0 days	Mon 2/16/09	Mon 2/16/09	36,46,56	0%	
66		Unit tests for business layer code	5 days	Tue 2/17/09	Mon 2/23/09		0%	
67		Write Junit tests for media file persistence layer	5 days	Tue 2/17/09	Mon 2/23/09	56	0%	Ganesh M
68		Write Junit tests for ATOM publishing generator	3 days	Tue 2/17/09	Thu 2/19/09	56	0%	Tanuja V
69		Write Junit tests for ATOM feed generator	2 days	Fri 2/20/09	Mon 2/23/09	68	0%	Tanuja V
70		Integration testing	43 days	Tue 2/17/09	Thu 4/16/09		0%	
71		Code merge and commit all changes to SVN repository	8 days	Tue 2/17/09	Thu 2/26/09	65	0%	All
72		Perform end-to-end testing of media blog feature	15 days	Fri 2/27/09	Thu 3/19/09	71	0%	All
73		Regression test other features that have some overlap with media blogging	10 days	Fri 2/27/09	Thu 3/12/09	71	0%	All
74		Fix defects found during integration testing	20 days	Fri 3/20/09	Thu 4/16/09	72,73	0%	All
75		Fix defects reported by community	20 days	Fri 3/20/09	Thu 4/16/09	72,73	0%	All
76		Application ready for rollout	0 days	Thu 4/16/09	Thu 4/16/09	70	0%	

6. Implementation Summary

Client-side coding - roughly 25% complete

We have started with the development of client-side coding. This project involves significant client-side coding, which includes display of media thumbnails and overlays and heavy usage of Javascript libraries such as JQuery. So our plan is to complete the client-side coding before moving on to backend processing. This is reflected in the schedule above.

The first step involves,

- 1) Completing Java Server Pages (JSP) with proper use of CSS layout.
- 2) Appropriate definitions of tiles configuration for page layouts.
- 3) Struts configuration file to wire action classes to form submissions and URLs.
- 4) Dummy Struts action classes that return mock data.

Currently, we have completed this task of creating JSPs, tiles configuration and struts configuration for four pages and are working on the rest. We intend to share this with roller community by hosting the application on a sandbox environment or by committing periodic patches into a branch created for this project. A dedicated branch has been created in roller repository for this project so we can submit our patches and share the changes with the community.