

Get Social

by

Matthew Kuckuck

Submitted to
the Faculty of the School of Information Technology
in Partial Fulfillment of the Requirements for
the Degree of Bachelor of Science
in Information Technology

© Copyright 2014 Matthew Kuckuck

The author grants to the School of Information Technology permission
to reproduce and distribute copies of this document in whole or in part.

_____ *Matthew Kuckuck* _____ 4/17/2014 _____

Matthew Kuckuck

Date

_____ *Russell E. McMahon* _____ 4/17/2014 _____

Russell McMahon, Faculty Advisor

Date

University of Cincinnati
College of
Education, Criminal Justice, and Human Services

April 2014

Acknowledgements

I would like to thank the University of Cincinnati for having a spectacular COOP program, without this program I would not have been able to do this project for Six Dimensions. I would like to thank Brian Warner of Six Dimensions for guidance with this project and the idea generator for this project. I would like that recognize Russell McMahon for keeping me on track with coursework.

Table of Contents

Section	Page
Acknowledgements.....	i
Table of Contents.....	ii
List of Figures.....	iii
Abstract.....	iv
1. Introduction.....	1
1.1 Problem.....	2
1.2 Solution.....	3
1.3 Project Goals.....	3
1.4 Overview.....	3
2. Discussion	
2.1 Project Concept.....	3
2.2 Design Objectives.....	4
2.3 Methodology/Technical Approach.....	4
2.4 User Profile.....	6
2.5 Budget.....	7
2.6 Gantt Chart.....	7
2.7 Problems Encountered and Analysis of Problems Solved.....	7
2.8 Future Recommendations.....	7
3. Conclusion	
3.1 Lessons Learned.....	8
3.2 Developed Skills.....	8
4. References	
5. Appendixes.....	12
A. Gantt Chart	
B. Sequence Diagram	
C. Use Case Diagram	

List of Figures

1. Concept.....	8
2. JCR Logic.....	10
3. Oauth Authentication.....	10

Abstract

Social media management has become a rising issue for Enterprise Website Content Management Systems (WCMS). Most social media sites recently have adopted an Oauth authentication model for accessing public posts. This allowed social media sites to identify the requester and set a quota for each request. Enterprise websites can easily exceed this quota because they have a high volume of traffic.

This solution created a component that is fully-integrated with the WCMS Adobe CQ. Quota management is established through temporarily caching posts inside a repository. Initial posts were retrieved from the user specified social media sites. Oauth authentication was handled automatically without the user's knowledge. The returned data is sorted, organized and stored, then rendered into the component using JavaScript. This component worked with major social media sites as identified: Facebook, Twitter, and Google Plus.

Enterprise companies that utilize this product will have reduced request to social media sites and will not exceed the quota limits.

All work is sponsored by Six Dimensions, a consulting firm for WCMS.

Get Social

1. Introduction

1.1 Problem

Enterprise companies want to be social. According to an article written by Altimeter, an analyst company, corporate-owned social media accounts had on average 39.2 Twitter accounts, 29.9 Facebook accounts, and 28.8 LinkedIn accounts (Owyang, Jones & Tran, 2012). Based on these statistics, it is understandable that these companies would have a difficult time maintaining all of these accounts. This article proposed a social media management system (SMMS) to help maintain these social media accounts. There are several SMMS solutions on the market for enterprise companies. One of these solutions is called Adobe Social, which is part of the Adobe Marketing Cloud. Adobe claims a tenfold ROI, within only two months of releasing Adobe Creative Suite 6 and Adobe Creative Cloud by using their Adobe Marketing Cloud (“Tenfoldroi, within,” 2012). Not only do Enterprise companies want to be social but also need a medium to distribute information to the public.

This medium is typically a Web site and it is common for Enterprise companies to utilize web content management systems (WCMS) to manage these Web sites. In an article by Gartner, WCMS products were analyzed and Gartner determined that public awareness and centralization of all Enterprise initiatives converge at the WCMS (MacComascaigh, Gilbert, Murphy & Tay, 2013). One solution for enterprise WCMS is Adobe CQ, this product is also part of the Adobe Marketing Cloud. Some companies that use Adobe CQ are Autodesk, Citrix, AT&T, Time Warner Cable, Ford, and many more. Adobe would suggest that these companies use both Adobe Social and Adobe CQ.

With price as a factor, some companies may not need all of the resources and features provided by a SSMS and choose to only use a WCMS.

Companies that choose this route would still like to have social media integrated with their Web site. Recently almost all social media sites have gone to a form of authentication utilizing the Oauth protocol. According to a document entitled, “The Oauth 2.0 Authorization Framework,” Oauth solves several problems that come from third-party applications having to store credentials for a resource that an end-user “owns” (Hardt, 2012). All websites must utilize this protocol in order to access user information without storing the user’s name and password.

Adobe CQ currently has some out-of-the-box components that work with Oauth (“Social login and,” 2012). Some problems exist with these included components: there are only Twitter and Facebook functions; there is no easy way to display the social media sites together (Lunka, 2013); quota management (exceeding request to API) is not addressed; an author would need engineering support to use the Facebook and Twitter components (which is not in scope with the purpose of a WCMS). If an Adobe CQ developer wants to have this functionality they will have to create it themselves which slows time to production.

1.2 Solution

The solution is to create a component for Adobe CQ. This component internally handles Oauth authentication for the social media site. The social media feed that is retrieved from the social media site will be temporarily cached in the Java Content Repository (JCR). This reduces calls to the social media site which causes the quotas for the social media site to not be exceeded benefiting the Enterprise Company from having

to purchase a higher quota. Developers benefit from reduced time to incorporate social media into Adobe CQ by installing the component through a package.

1.3 Project Goals

1. Create an Adobe CQ component
2. Access Facebook, Twitter, and Google Plus public posts
3. Temporarily store posts
4. Aggregate data
5. Mitigate conflicting of multiple components
6. Display posts in browser

1.4 Overview

The remainder of this report will discuss in detail how the project was completed. The report includes the following sections: design objectives, methodology, budget, timeline, problems encountered, and future recommendations.

2. Discussion

2.1 Project Concept

The concept behind this project, as seen in Figure 1, is to get social media sites public posts for Facebook, Twitter, and Google Plus. The public posts will be displayed in a browser using Adobe CQ. This concept was cultivated through years of experience working with Content Management Systems and Enterprise Company's needs. The cultivator of this concept was Brian Warner an employee of Six Dimensions.



Figure 1: Concept

2.1 Design Objectives

The main goal for this project was to be able to access social media sites data for public user posts and keep the calls to the social media sites' API relatively low. In order for this to be accomplished, the component had to use OAuth authentication to access the social media sites API and the retrieved data temporarily needed to be stored. Another important goal was to be able to have multiple components that would not conflict with each other. This would allow the CQ author to be able to request any number of posts for a select social media site and a select user in one instance and in another is able to select a different user for the same social media site and the data request and storage would not conflict with each component. When utilizing OAuth authentication, each of the selected social media sites have two implementations for accomplishing this authentication. One is an individual user centric approach and the other is multiple user centric approach. The individual user approach is the direction taken for this project. It was easier to implement and still was able to meet the main goals of this project. Some aspects that would have been nice to implement comes with the multiple user centric approach. This approach requires the user to login to their social

media site and this would give an access token to the requesting component. By using this approach, the component would have two directional communications with the social media site. This means the component would have not only been able to get posts from the social media site but could also post posts to the social media site for the specified user. The problem with this would have been storing access tokens for each user.

2.2 Methodology/Technical Approach

The methodology for this solution consisted of three parts administrator involvement, author usage, and a component that is fully-integrated with Adobe CQ.

The administrator would first create an application with each of the major social media sites. When creating this application the social media site will give the application an application id, application secret, consumer key and consumer secret. This information is put into the Apache Felix console under the service that will make request to the social media site. The administrator has to give the author permission to use the component that will access this service.

The author drags the component onto the Web page. The author inputs information into the component specifying which users' posts to display and how many posts to display for each of the social media sites. This is from the user perspective; however from the component perspective, the component communicated with a Java servlet using URL path based AJAX call. The servlet is contained in an OSGI bundle. Adobe CQ uses Apache Felix for handling OSGI bundles.

An OSGI bundle is a modularized container that can be activated and deactivated by the administrator. The URL contained a request parameter that the author provided information to request from the social media site. Before this request is sent to the social

media site, the servlet will check the Java Content Repository (JCR) to ensure that post does not exist. If post doesn't exist or if the post is greater than 15 minutes the request will be sent to the social media site.

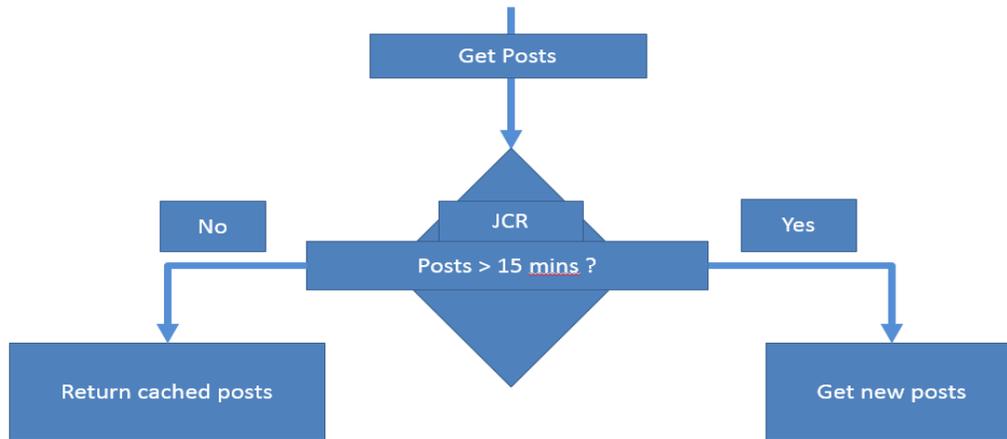


Figure 2: JCR Logic

The access token must be gained to be able to call the social media site's API. This occurs through Oauth authorization and is a four step process (See Figure 3). The first step is to take the application key and application secret (each social media site might have a different name for these) send this information to the social media site and request an access token. Step two, if application secret and key are valid the access token will be returned. The third step sends the access token along with the request information to the social media site. The final step, the social media site returns the social media data.

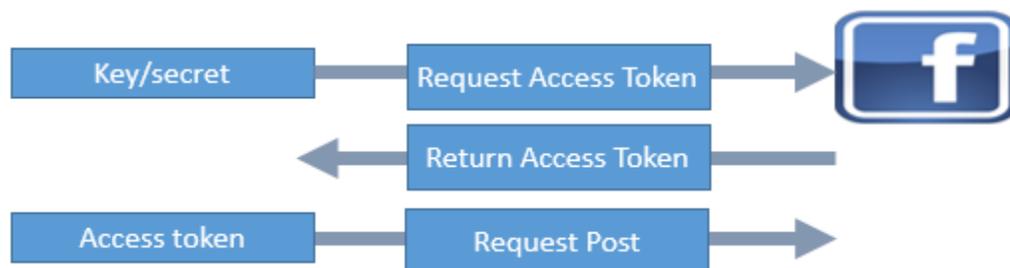


Figure- 2 Oauth authentication

The servlet sorts, organizes, and aggregates the data, stores the data in the JCR with a timestamp then returns the data to the component. View appendix B for the sequence diagram.

2.3 User Profile

There are three user types for this application: administrator; author; website visitor. The author could be marketing personnel, a blogger, or any type of website content author.

The author should only have a basic knowledge of Adobe CQ's Web Content Management Author environment. The author does not need to have any technical knowledge of Java or the internals of Adobe CQ. The author will interact with the component only to change the amount of post returned or if they want to retrieve post from another social media site.

The administrator will need to be familiar with the Apache Felix console in Adobe CQ. The administrator will also need to know how to create an application with the desired social media sites. The administrator will perform the configuration once. The website visitor will view the page and not need any technical knowledge.

2.4 Budget

Six Dimensions has provided all the hardware and software necessary to complete this project. Six Dimensions is also paying for all labor associated with this project and will retain all proprietary rights to the project code. It is not in the company's best interest to disclose any of the budget information.

2.5 Gantt Chart

See appendix A

2.6 Problems Encountered and Analysis of Problems Solved

Conflicting components was a challenging issue to resolve. When two components were placed on the page and each component requested the same user profile, the second component would overwrite the first components post data in the JCR. Originally, each request to each social media site was stored in a JSON string. The correction was implemented by creating nodes for each user under each social media site node. When the request would be made it would look at the specific user node. This solution created new problems. If the component requested 3 posts and only 2 existed for that user and these posts were expired then the improper amount of posts would be returned. The solution for this was to delete any posts that were less than the requested amount and request the new amount from the social media site.

Another problem occurred when one component requested 3 and another requested 5, it was impossible to call a request of 2 to add to the stored 3 amount. This was because the social media sites were returning the top 2 posts so if you had 3 stored the top 2 were the same and the 2 new requested posts. In order to fix this, if one component requested 3 and the other requested 5, the higher request count wins and is sent to the social media site. The first component would then pull from the stored 3 posts. View appendix C for use case diagram.

2.7 Future Recommendations

This project would greatly benefit if multi-directional communication and user-centric Oauth authentication was implemented. This should not replace the current functionality of this project but would extend the project to meet other use cases.

In essence, after creating these feature sets it would be possible to create a social media management tool completely within Adobe CQ. This would give Six Dimensions a competitive edge.

3 Conclusion

3.1 Lessons Learned

Always try to find a plugin or extension that will meet your needs. Google Plus had a useful Java library that made accessing the Google Plus API very easy. When doing a project having a nice project management tool is crucial. Pivotal tracker was the project management tool that was being used but the 30 day trial expired. It organized the flow of the project very well.

3.2 Developed Skills

This project helped develop skills with REST, Oauth authorization understanding, working knowledge of Adobe CQ, and social media sites APIs.

3.3 Value

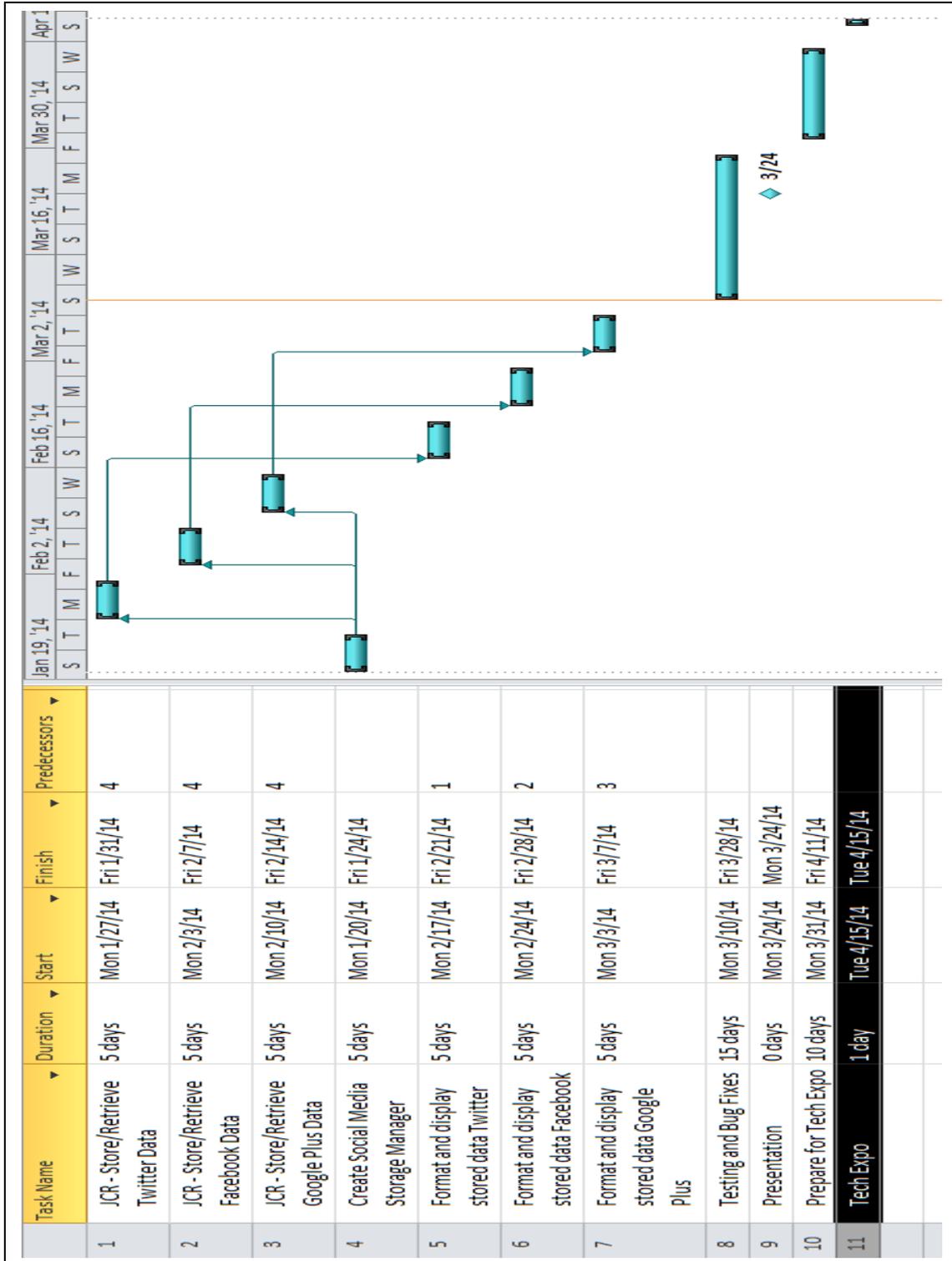
Since Enterprise Web sites have high traffic, it is possible to exceed quota by simply using this component with Adobe CQ, this type of website could reduce the amount of calls to the social media site. This reduction would allow for lower quota and reduce the cost required for higher quotas.

References

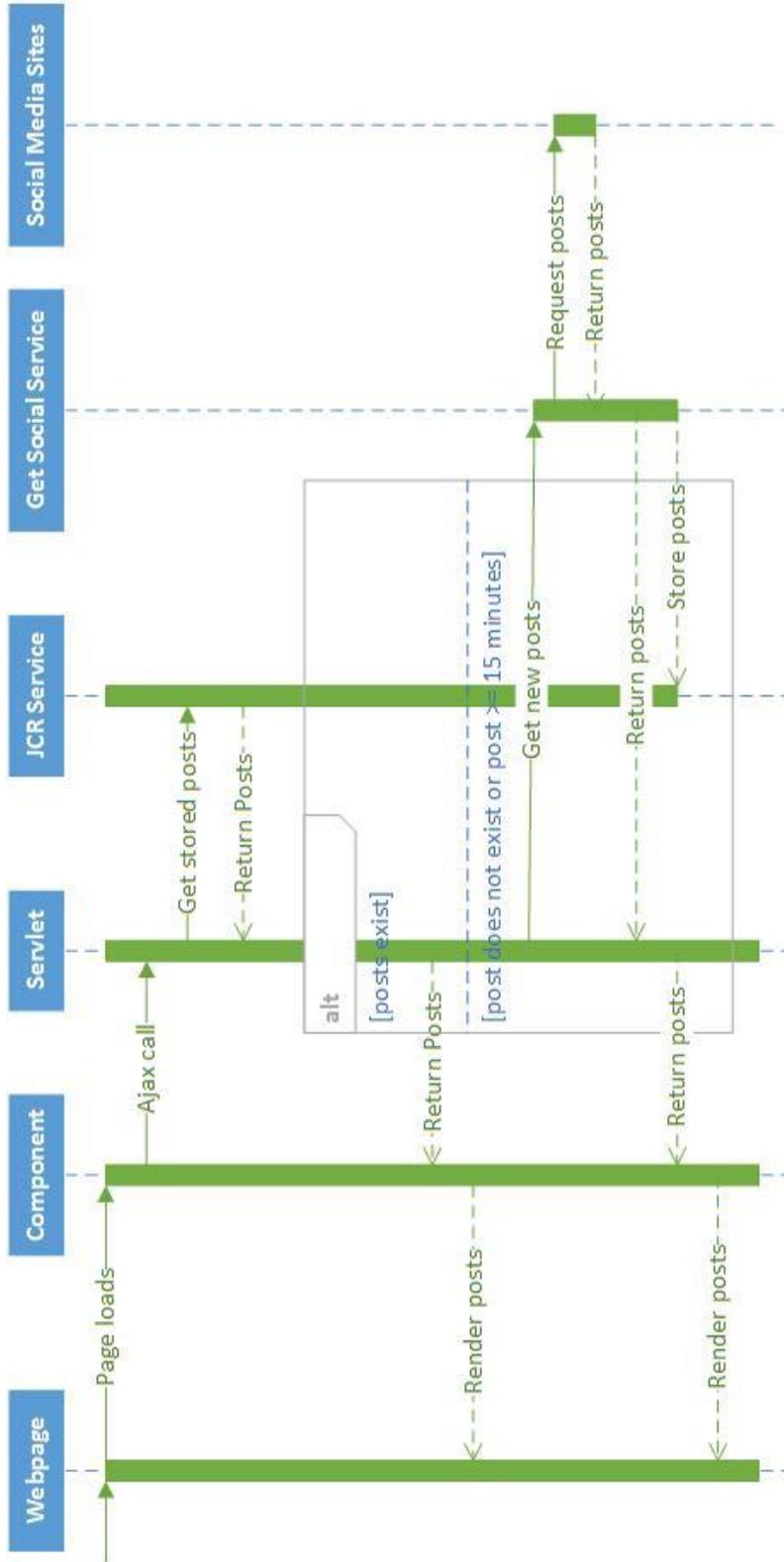
- Adobe experience manager 5.6.1 technical requirements.* (2012). Retrieved from https://dev.day.com/docs/en/cq/current/deploying/technical_requirements.html
- Cq 5.5 technical requirements.* (2012). Retrieved from https://dev.day.com/docs/en/cq/5-5/deploying/technical_requirements.html
- Dispatcher.* (2012). Retrieved from <http://dev.day.com/docs/en/cq/current/deploying/dispatcher.html>
- Hardt, D. (2012). The oauth 2.0 authorization framework. *Internet Engineering Task Force (IETF)*, Retrieved from <http://tools.ietf.org/html/rfc6749>
- Lunka, R. (2013). *Adobe experience manager. Classroom in a book.* San Francisco: Peachpit MacComascaigh, M., Gilbert, M., Murphy, J., & Tay G. (2013). Magi quadrant for web content management. Retrieved from <http://www.gartner.com/technology/reprints.do?id=1-1I3UYPS&ct=130802&st=sb>
- Owyang, J., Jones, A., & Tran, C. (2012). A strategy for managing social media proliferation. *Altimeter Radar, A Decision-Making Guide*, Retrieved from <http://www.web-strategist.com/blog/2010/03/19/list-of-social-media-management-systems-smms/>
- Social Login and personalization. (2012). Retrieved from http://dev.day.com/docs/en/cq/current/administering/social_communities/social_connect.html
- Tenfold roi, within two months. (2012, September). *Adobe Marketing Cloud Success Story*, Retrieved from <http://www.adobe.com/content/dam/Adobe/en/customer-success/pdfs/adobe-social-cs6-launch-case-study.pdf>

Appendixes

Appendix A -- Gantt Chart



Appendix B – Sequence Diagram



Appendix C – Use Case Diagram

