Final Thesis report

Social media in case-handling systems

by

Daniel Jans & Johan Juel

LIU-IDA/LITH-EX-G--11/027--SE

2011-11-15
Final Thesis report

Social media in case-handling systems

by

Daniel Jans & Johan Juel

LIU-IDA/LITH-EX-G--11/027--SE

2011-11-15

Supervisor: Juha Takkinen
Examiner: Jose M Peña
Abstract

Ida infront, a part of Addnode, is a software company that focuses on developing case-handling systems, systems for safe information transactions and e-storage.

Social media, such as blogs, wikis and social networks, is something that millions of people, all around the world, uses on a daily basis. The purpose of this thesis was to examine if social media functions can support the users of Ida infront's case-handling system. To do this, we made an implementation with the purpose of demonstrating it to their customers. The study started with a focus group to formulate a specification of requirements on the implementation. Based on that specification, an implementation with a chat, RSS-function and a wiki was made that could run inside ipax permission which is Ida infronts case handling software. The implementation itself was made using Liferay, a web portal that is easy to customize. Both ipax permission and Liferay were deployed on a Jboss application server. The implementation was then demonstrated for external customers who filled in a survey and answered some interview questions. The result is based on the surveys and interviews.

This study shows that social media functions can be used to support the users of case-handling systems.

To be able to draw a more general conclusion about social media in case-handling systems, more demonstrations, surveys and interviews need to be made. Future work can also involve looking at social media functions not covered in this study.
# Table of contents

1 Introduction .................................................................................................................................1  
   1.1 Background ..........................................................................................................................1  
   1.2 Purpose ................................................................................................................................1  
   1.3 Limitations ...........................................................................................................................1  
   1.4 Methods of research ............................................................................................................1  
   1.5 Ida infront and lipax ............................................................................................................1  
   1.6 Disposition of the thesis .......................................................................................................3  
2 Social media ..................................................................................................................................4  
   2.1 What is social media? .............................................................................................................4  
   2.2 Social media characteristics .................................................................................................4  
   2.3 Different types of social media .............................................................................................5  
      2.3.1 Collaborative projects ....................................................................................................6  
      2.3.2 Blogs ..............................................................................................................................6  
      2.3.3 Content communities ....................................................................................................7  
      2.3.4 Social network sites ......................................................................................................7  
      2.3.5 Virtual social worlds .....................................................................................................7  
      2.3.6 Virtual game worlds .....................................................................................................7  
   2.4 Social media users ..................................................................................................................8  
3 Analysis ..........................................................................................................................................9  
   3.1 Pre-study ................................................................................................................................9  
   3.2 Data collection .......................................................................................................................9  
      3.2.1 Focus group ....................................................................................................................9  
      3.2.2 Surveys and interviews .................................................................................................10  
   3.3 Results and conclusions .........................................................................................................11  
4 Design and implementation ..........................................................................................................12  
   4.1 Specification of requirements ...............................................................................................12  
   4.2 Implementation .....................................................................................................................13  
   4.3 Liferay Portal .......................................................................................................................13  
   4.4 Implementation process .........................................................................................................14  
      4.4.1 First demonstration and changes ..................................................................................18  
5 Evaluation ....................................................................................................................................21  
6 Discussion and conclusion ...........................................................................................................22  
   6.1 Discussion .............................................................................................................................22  
   6.2 Conclusion of the study .........................................................................................................22  
   6.3 Future work ...........................................................................................................................22  
References ......................................................................................................................................23
1 Introduction

This chapter describes the background and purpose of the thesis. It will also contain the limitations, methods of research, disposition of the report and some information about the company where we did the work.

1.1 Background

Social media, such as blogs, wikis and social networks is, on a daily basis, used by millions of users all around the world. Millions of people are used to being connected to others through social networks, automatically receiving updates in subjects they find interesting or finding information they need through a simple search on the web or in a wiki. Ida infront AB wanted to examine if their customers would like similar functions integrated in their case handling systems to allow their users to chat, get updates, contribute with knowledge or find information in an easy way.

1.2 Purpose

The purpose of this study was to examine if social media functions can support the users of Ida infront's case-handling systems. To do this, we did implement social media functions that could be used within Ida infront's iipax permission.

1.3 Limitations

In our thesis report we had limitations on which social media types we focused our attention on. This limitation was the result of discussions with staff at Ida infront and was necessary for us to create our implementation. The social media functions that we focused on were an RSS-function, a wiki-function and a chat-function. Another type of limitation we had to make was to choose a framework with some already built-in functionality in order to have time to focus on a higher number of social media types.

1.4 Methods of research

To gather the information needed for this study we used different research methods. A focus group was held to get information about the requirements of the implementation. We also had several meetings with our company supervisor, project managers and product developers at the company. To evaluate the implementation, we used a combination of surveys and interviews.

1.5 Ida infront and iipax

The work for this thesis was made at a company called Ida infront. They specialize in software solutions for case handling and information handling.

Iipax is a product family created by Ida infront. It consists of four products, iipax receiver, iipax permission, iipax communication and iipax archive.

Iipax receiver lets you create advanced forms and web services that requires secure authentication by electronic id and electronic signatures.
Ipax permission is the product that we worked with during our implementation and it is a case handling system. It handles a variety of information regarding cases like documents, check lists and processes and also the properties of the case itself. In picture 1, you can see an example of how Ipax permission “Att göra” (to-do) tab, that displays cases, can look like. Ipax permission supports the integration of existing user authentication directory services like LDAP, Jaas and Active Directory.

![Ipax permission to-do tab.](image)

Ipax communication is a product that allows for secure transfer of information. To do that it uses PKI (public key infrastructure) to ensure that the information is sent to the receiver and also that the receiver is who he or she claims to be.

Ipax archive is simply put an electronic archive to store electronic information and data.
1.6 Disposition of the thesis

2 Social media
This chapter describes social media, its characteristics and users.

3 Analysis
This chapter describes the work procedure and methods of research used during this study.

4 Design and implementation
This chapter describes the the requirements on the implementation and the implementation process.

5 Evaluation
This chapter describes the evaluation process and the results of the evaluation.

6 Discussion and conclusion
This chapter contains the discussion and conclusion.
2 Social media

This chapter describes social media, its characteristics and users.

As long as humans have existed they have created social networks to be able to cooperate and for protection. Nowadays, those social networks still exist, but there are also other types of social networks that can be of help in different situations. Before you go to see a movie it is common to first consult a friend or a website about it (Safko, 2010).

Social media is an expression that emerged from the web 2.0 discussion. These two expressions are often used as synonyms but there is a clear distinction between them. Web 2.0 is about the technology that may or may not be used for social media (Katajisto, 2010).

It is web 2.0 that have provided the people with tools with which they can generate content, alone or in collaboration with others, and perform tasks that earlier could only be performed by organizations (van Heerden, 2010).

2.1 What is social media?

The short answer is that social media is whatever media we use to be social. The term social media is constructed by two familiar words, social and media. Social refers to the human instinct of making connections with other humans. Humans have a need of belonging to groups with like-minded with which they can share thoughts, ideas and experiences. The word media stands for how those connections are made, whether it is smoke signals, bells, sites, phone calls or e-mails. Media is the technology used for making interpersonal connections. When the two terms, social and media, are combined, forming the term social media, their combined meaning is how humans can use the technology to reach out to other humans, creating connections, building trust and sharing experiences. To say it simple, social media is just a new set of tools that helps us making connections to others more efficiently (Safko, 2010).

2.2 Social media characteristics

The characteristics of social media make content aggregation and social network building easier (Zheng, Li & Zheng, 2010).

According to Mayfield (2008) social media often shares most or all of the following characteristics:

(1) **Participation**: social media invites its audience to make contributions. Anyone who is interested can generate new content or comment on already existing content. Social media makes the line between the media and its audience less distinct.

(2) **Openness**: social media services tend to be open for the users so that they can freely access, and in other ways interact with the content.
(3) *Conversation:* traditional media can be seen as a one-way conversation, whilst social media is better understood as a two-way conversation.

(4) *Community:* social media allows users with common interests to quickly and easily form communities.

(5) *Connectedness:* most types of social media gets great benefits from its connectedness to other sites or resources.

Social media is a more efficient way of making connections compared to traditional media. One reason that makes social media much more efficient is that it uses two-way, rather than one-way communication (Safko, 2010).

### 2.3 Different types of social media

According to Kaplan (2010) social media can be split into different categories depending on their main function. Social media can be split into the following categories, explained further by Table 1:

1. Collaborative projects
2. Blogs
3. Content communities
4. Social networking sites
5. Virtual game worlds
6. Virtual social worlds.
<table>
<thead>
<tr>
<th>Category</th>
<th>Main function</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Social network sites</td>
<td>Create social networks online</td>
<td>Facebook, LinkedIn</td>
</tr>
<tr>
<td>Blogs</td>
<td>Create and share anything in text form</td>
<td>Blogger, Twitter</td>
</tr>
<tr>
<td>Content communities</td>
<td>Create and share content other than text</td>
<td>Flickr, Delicious and Dopplr</td>
</tr>
<tr>
<td>Collaborative projects</td>
<td>Collaborative effort to create</td>
<td>wikipedia, Git</td>
</tr>
<tr>
<td>Virtual social worlds</td>
<td>Socialize in a virtual ”real life”-like world</td>
<td>Second Life</td>
</tr>
<tr>
<td>Virtual game worlds</td>
<td>Socialize with similar minded within a game</td>
<td>Worldof Warcraft</td>
</tr>
</tbody>
</table>

Table 1. Social media categories, main functions and examples.

2.3.1 Collaborative projects

Working together towards a common, larger goal is what this category is all about. Wikis are an example of this (Kaplan & Haenlein, 2010).

A wiki is a web platform that is browser-based. Wikis represents the social media foundation of collaborative content creation, and it takes advantage of the wisdoms of the crowd. Wikis lets the users contribute with content based on their own knowledge and expertise. The users can create new content or edit already existing information (Safko, 2010).

To read and contribute to a wiki the only thing you need is a normal web browser, since the wiki is a web platform. A wiki can be either public or private and it is a popular way for companies to create internal collaborated content (Safko, 2010).

Wikipedia is a very large wiki with more than three million English language articles (Wikipedia, 2011).

2.3.2 Blogs

Blogs are web pages that are chronologically ordered in reverse and allow for user interaction through comments. They can be about pretty much any subject and are usually maintained by a single person (Kaplan & Haenlein, 2010).
2.3.3 Content communities
Sharing is a major part of social media, so naturally there are sites that allow sharing of specific content that could be pretty much anything, links, videos, pictures, audio etc. One very popular content sharing site is YouTube where users share videos. Other social media in this category include Flickr, Delicious and Dopplr (Kaplan & Haenlein, 2010).

Another way of sharing is using Really Simple Syndication (RSS). RSS is a technique that allows anyone to freely distribute their websites content all over the world. RSS provides a way to feed i.e. web pages, blogs and photographs automatically to people who have subscribed to that content through a feed. Every time an update is made to the subscribed content the users automatically receive a notification. The notification can be an e-mail, mobile text message or a tweet. The users no longer have to visit the web sites or blogs to look for updated content if they are subscribers. At the same moment new content is added the user will be informed. It is very easy for a user to start subscribing for a feed; it often just requires one click on a button (Safko, 2010).

2.3.4 Social network sites
Forming a social network online is something that is made easy using social network sites. Here you create a personal profile with info about yourself and then you can socialize with other users (Kaplan & Haenlein, 2010).

One part of social networking is chatting. Chatting is when you communicate with others, it can be using text, voice or even video, in a “live” manner. That means, you chat with someone and they reply instantly, that’s why it is also called instant messaging. There are also different types of each of these, like one-to-many, one-to-one and many-to-many chat (Safko, 2010).

2.3.5 Virtual social worlds
Another way to socialize online apart from social network sites as mentioned earlier, is virtual worlds. This is essentially a graphic representation of a “world” on your computer screen where you are a character and can walk and talk to anyone you meet, just like in real life. An example of this is Second Life (Kaplan & Haenlein, 2010).

2.3.6 Virtual game worlds
Similar to virtual social worlds, however instead of mimicking real life the platform for this is a game, and you socialize within it. World of Warcraft is an example of this (Kaplan & Haenlein, 2010).
2.4 Social media users

The users of social media can be categorized into six different types of users. There are the creators, the most active users that generate and upload new content. The published content will then be commented by the critics. The collectors tags photos and keep track of the latest content i.e. by using RSS-feeds. The joiners join the networks and create their own profiles and the spectators that just visit social media sites to read. The inactive don't participate at all. A person can belong to a different user type in different situations. It can be a creator on one site and a critic on another (Katajisto, 2010).

According to Nielsen (2006) a minority of the users contribute with a majority of the content. He claims that social media users often follow the 90-9-1 pattern (fig 1).

90 means that 90% of the users read the content, but do not contribute, so called lurkers. The 9 refers to that 9% of the users make contributions now and then. The 1 stands for the 1% of the users that are frequently active and make most of the contributions.
3 Analysis

This chapter describes the work procedure and methods of research used during this study.

3.1 Pre-study

During the pre-study we gathered information about Ida infront and met our company supervisor, who works as a project manager. Together with our company supervisor we sat down and discussed the assignment and to get more information about the company. We did also study literature about social media and internal documentation about ipax permission. As a result of the pre-study, we made a time plan for the study.

3.2 Data collection

There are different ways of collecting data i.e. studying documentation, tests, surveys and interviews (Patel & Davidsson, 1994). To collect data for this thesis, we have been studying documentation, held a focus group and used surveys and interviews.

3.2.1 Focus group

A focus group is a method where a group of people sit down together to discuss a certain topic with each other. The group is guided by a moderator, that initializes the discussion, and that also can introduce new perspectives on the topic if needed. The purpose of the focus group is a free discussion amongst the group members. Focus in focus group refers to that the discussion has a focus topic that the group members have been informed with in advance (Wibeck, 2010).

The group discussion is only a part of a focus group. Much of the work is done before and after the discussion. A focus group can be divided into four steps: planning, recruitment, moderation and analysis (Wibeck, 2010).

The discussion in the focus group can be structured or unstructured. The more the moderator is moderating the discussion, the more structured it is considered to be. In an unstructured discussion, the moderator stays in the background and lets the group members talk to each other and not the moderator (Wibeck, 2010).

There are pros and cons with both structured and unstructured discussions. A danger with having a too structured discussion is that the moderator affects the group member’s attitudes and values. On the other hand, an unstructured discussion can miss out important aspects of the topic (Wibeck, 2010).

In the invitation to the focus group, it can be a good choice to attach a stimulus material to the invited persons, containing i.e. articles, pictures, and movies about the focus topic. The purpose with the stimulus material is to make the invited persons to start thinking about the subject and prepare questions and thoughts (Wibeck, 2010).

To aid the moderator during the discussion, an interview guide is made. The guide should contain five different types of questions: opening questions, introduction questions, transition questions, key questions and ending questions. Depending on if the discussion is structured or not, the guide can differ (Wibeck, 2010).
The purpose with the opening questions is to make the group members familiar with each other. To introduce a topic the introduction questions are used. Transition questions can be used to guide the discussion. The key questions, that are most important for making the analysis, should be given more time than the other questions. After the key questions have been given sufficient time, the moderator can end the discussion with the ending questions that should give the group members the opportunity to express their last thoughts on the topic (Wibeck, 2010).

3.2.2 Surveys and interviews

Surveys and interviews are information gathering techniques that both uses questions to get data. These both techniques have some shared characteristics, and can also be used together (Patel & Davidsson, 1994).

Most people think of surveys as a questionnaire that is sent home to the respondent, but a survey can also be done together with the respondent, which means that the respondent can get help to understand the questions if needed (Patel & Davidsson, 1994).

When it comes to using questions as a means of collecting data, two issues needs to be considered. First of all a decision must be made on in which order the questions should be asked or it does not matter, standardization. Secondly, a decision must also be made about how freely the questions can be answered by the interviewed person, structuring (Patel & Davidsson, 1994).

Interviews with a low degree if standardization gives the interviewer the opportunity to formulate the questions during the interview. An interview with a high degree of standardization means that the interviewer asks the same questions in the same order to all persons he will interview (Patel & Davidsson, 1994).

An interview with a high degree of structure limits the possible answers from the interviewed person, whilst an interview with a low degree of structure gives more room for different answers (Patel & Davidsson, 1994).

Both surveys and interviews are dependent on that the person is willing to answer the questions, so to maximize the chances for good answers it can be good to motivate the subject of the survey or interview. The motivation can i.e. be done by clearly explaining the purpose, how the answers will be used, whether or not it is anonymous and possible benefits the person can get form the result of the study (Patel & Davidsson, 1994).

When it comes to the actual questions used, it is advised to avoid the following types of questions:

• Long questions
• Leading questions
• Negations
• Double questions (Do you usually do this or that?)
• Assuming questions
• Questions containing difficult words that might not be known by the subject.
When it comes to surveys it is important to not have identical answer options on all questions to prevent the respondent to get caught in a specific pattern and to keep that persons motivation. There are different kinds of answer options to use, i.e. yes/no or ranking alternatives (Patel & Davidsson, 1994).

In this thesis we will use a survey that is done together with the respondent in combination with a smaller interview. The survey will have a moderate degree of structuring, and the interview will have a high degree of structure with a low degree of standardization.

### 3.3 Results and conclusions

The focus group provided us with lots of information about what kind of implementation Ida infront wanted. We also got some ideas about how the implementation could be made, and also which other persons in the company that might have useful ideas and information. The outcome of the focus group was used as a foundation to the specification of requirements of the implementation, which can be seen in section 4.1 Specification of requirements.

The result of the survey and the interview can be seen in the chapter 5. Evaluation.
4 Design and implementation

This chapter describes the requirements on the implementation and the implementation process.

4.1 Specification of requirements

After we decided that we were going to hold a focus group meeting we had to do some planning. We talked to our supervisor and the marketing manager at the company and together decided that we should send invites to all project managers and a few other important people in the company by e-mail. Since we were going to talk a lot about social media we decided that we should include some stimulus material so that they could get a better idea about the subject before the meeting.

We used our report work about social media as the stimulus material and wrote the invitation text and finally sent out the invitations. Unfortunately many of the people invited did not have time, though four of them made it to the meeting and were part of the focus group.

We decided on keeping the focus group unstructured and leave the talking to the people invited so that we could gather as much of their thoughts as possible. The attendants came with a lot of ideas and early on in the meeting it was clear that they wanted more than one feature to be implemented.

The meeting lasted approximately one hour and when it was done we knew which functions they wanted implemented, this can be seen in the specification of requirements.

Based on the information gathered from the focus group and meetings with a company product developer, as mentioned in chapter 3.3, we could specify the requirements on the implementation. Ida infront wanted an implementation that contained at least a wiki, a chat and an RSS-feed that could display information from iipax.

The wiki should be easy for the customers to use for different purposes, i.e. as a knowledge database. It should be possible to make different parts of the wiki visible for different types of users, and also easy to extend for the customers.

Regarding the chat, it should allow the users to send messages to each other, and it should also be possible to modify which users a user can chat with.

The RSS-like function should present information about cases regarding iipax, such as events.
4.2 Implementation

We now had the specification of requirements for the implementation and the next step was to select a way of making it. Our first thought was to code the entire implementation from scratch using Java. This way we would have complete control over the functions and design of the entire implementation, and it would be fairly quick and easy for us to make modifications to it. On the other hand, it could be very time consuming to make it ourselves.

We then decided to see if there were any open source solutions available that would make the implementation process easier and less time consuming.

After a few initial searches we found an interesting open source solution called Liferay Portal. We looked closer at it and it contained the core functions for all of the requirements of our implementation.

After looking at the pros and cons between building the implementation from scratch versus modifying an already existing solution we did choose to make the implementation with Liferay Portal. The main argument for this choice was that our timeframe was limited and by using Liferay Portal we would save valuable time. Another advantage with Liferay Portal was that it was open source so we had access to the source code and could modify it so that it suited the requirements of the functions and design.

4.3 Liferay Portal

Liferay Portal is an open source project that was created in 2000. It has open-source solutions for portals, publishing, content and collaboration. There are two different versions, the Community Edition(CE) and the Enterprise Edition(EE). The Community Edition is free while the Enterprise Edition is not, and it includes support amongst other things. In this thesis, we used the free Community Edition version of the Liferay Portal software (Liferay.com, 2011).

Liferay Portal is a web platform that comes with a lot of built-in applications, called portlets. Portlets can be placed anywhere in the portal and are easily customizable. Some examples of portlets that are of interest to us are: wiki, chat, RSS reader, calendar and announcements. There are however many more portlets to choose from (Liferay.com, 2011).

Liferay Portal features its own log-in system complete with role based security and user types, and the users have their own public and private pages where they can place portlets as they wish (Liferay.com, 2011).

Most settings in Liferay are easy to modify either using a built-in control panel, or by modifying its system properties file (Liferay.com, 2011).

Liferay Portal can be deployed on a number of different application servers and it can also use a number of different databases. By default, a Hypersonic HSQL database is used (Liferay.com, 2011).

HSQL stands for Hyper SQL(Structured Query Language) and is an open source java database language (HSQL Development Group, 2011).
4.4 Implementation process

The first thing we started with was to download and install Liferay Portal to get familiar with it. We started to go through the portlets that came with the installation to see if there was anything included that could be of use for the implementation. Liferay came with some portlets that we could use for the implementation, a working RSS-reader and a wiki portlet. Our installation did not contain a chat portlet from start, but we could easily install one from the control panel in Liferay. We now had the basic functions needed for the implementation inside Liferay so we started to look at iipax.

The RSS-reader that came with Liferay worked out-of-the-box and allowed the subscription of multiple feeds, as can be seen in picture 2. The RSS-reader seemed to have a hardcoded refresh time set to 30 minutes, which we thought, at least for a demonstration purpose, was too long. To solve this, we had to make changes in the RSS-reader’s source code. When we located the source code for the RSS-reader we did set the refresh timer to ten seconds, so that new events would show up much faster when we were going to demonstrate the implementation. The changes made, and how they were made can be seen in appendix A.

![Picture 2. Multiple feed RSS-reader.](image)
Liferay also comes with a wiki-portlet that makes it possible to build a wiki (see picture 3.). Out of the box, the wiki contained lots of functions, some which we thought was not necessary to meet the requirements on the implementation. Therefore we started to remove features we considered redundant, such as the ranking of pages function, by modifying the wiki source code, and changing configuration files. For a complete list of changes made to the wiki and how they were made, see appendix A.

The only change made to the chat-portlet was which users that should be displayed as online in the buddy list. By default, the chat displays whoever is logged in and allows all logged on persons to communicate (see picture 4). We changed the display strategy so it only shows users that belong to the same community. By doing that, it becomes possible to divide the users into smaller group which may have the need to chat with each other, instead of having one massive buddy list. How that configuration is made can be seen in appendix A.
After we had gotten familiar with Liferay and its portlets, we also wanted to get familiar with iPax permissions, the product in which our implementation should be displayed. With some help from Ida Infront personnel, we installed a fully functional demo-version of iPax permissions that we could use to experiment with during the implementation.

The next step was to deploy Liferay Portal on the same application server as iPax permission. Our first installation of iPax was deployed on a jboss-4 server and we could not deploy Liferay successfully on that one. After some research, we found out that Liferay could successfully be deployed on a jboss-5 server. We then made a new build of iPax and deployed it on a jboss-5 server. To be able to deploy Liferay on a jboss-5 server, some configurations had to be made. For a complete guide on what we did to deploy Liferay on a jboss-5 server, see appendix A.

We now had iPax and Liferay up and running on the same application server. The next thing we wanted to do was to see if we could extract information from iPax permissions. To see if we could do that, we wrote a servlet that used the iPax API and deployed it on the server. At first, iPax refused our servlet to extract information due to security reasons. To solve that, we had to add basic authentication to the servlet. After that was done, we could successfully extract the information we needed for the implementation. For the code to the servlet, see appendix B.

To be able to present the extracted information in a good way and to keep it updated, we had planned to make an RSS-feed out of it. We extended the servlet used for extracting information so that it could make an RSS-feed of the extracted information. To achieve this, we used Rome, which provided us with the tools needed to make the RSS-feed. Rome is an open source project in Java used to read and generate RSS feeds. Rome supports several versions of RSS and is very easy to use (Ferguson, 2007). For the code to the extended servlet, see appendix C.

When we tried to run Liferay inside of iPax permission, we encountered a problem, the built-in browser did not have JavaScript enabled, which is needed for the RSS-portlet to work. After a discussion with Ida Infront staff, we decided to run Liferay outside of iPax permission for now and solve that problem later.

Some additional changes were also made to Liferay Portal. One of the changes we made was to disable the terms of use that are displayed when a user logs in for the first time, since we think that was not needed for our implementation. We did also disable the password reminder and toggle edit controls. In addition to these changes, we also added the possibility for all users to maximize specific portlets to operate them in full-screen if they want to. We did also disable user public pages and made changes to prevent users from logging out due to inactivity. For a complete list of changes made to the portal, see appendix D.

Some additional changes were also made to Liferay Portal. One of the changes we made was to disable the terms of use that are displayed when a user logs in for the first time, since we think that was not needed for our implementation. We did also disable the password reminder and toggle edit controls. In addition to these changes, we also added the possibility for all users to maximize specific portlets to operate them in full-screen if they want to. We did also disable user public pages and made changes to prevent users from logging out due to inactivity. For a complete list of changes made to the portal, see appendix D.
At this stage in the implementation we could extract case information from iipax permission, build an RSS-feed that could be displayed in the Liferay RSS-portlet, as well as in any other RSS-reader. We also had a start for a wiki that could easily be extended, and a working chat.

Ida infront did make us a copy of iipax permission with java enabled, so now we could run our implementation inside of iipax. Our implementation could be reached by going in to the “mina sidor”-tab in iipax permission (see picture 4.).

![Picture 4. Mina sidor-tab, displaying the implementation inside iipax permission.]

### 4.4.1 First demonstration and changes

We now booked a meeting with our company supervisor and the marketing manager to demonstrate what we had done so far. That meeting resulted in a number of tasks that we needed to perform before demonstrating the implementation to customers:

- Build a number of RSS-feeds containing different information, so the user can choose for itself what it is interested in.
- Enable user private pages so the users can add more functions (portlets) if they want to.
- See if there is some kind of announcement portlet that could be used for sending out important notices to the users.
- Make some changes to the graphics so it suits iipax permission layout more.
- Have two meetings with company project-managers before demonstrating the implementation to the customers.
• Set up accounts, communities and other parts needed for the demonstration.

• Make a demonstration plan to be sure to cover all parts of the implementation.

• Prepare a survey to collect external customer's opinions about the implementation.

During this meeting it was also decided that we should visit the customers to demonstrate the implementation and thereafter let them fill out a survey and answer a few other questions about the implementation.

After the meeting, we changed the RSS-servlet so it created three different RSS-files; work pool, record events and events. We also made some changes to the looks of the portal to better suit the look of iipax permission. To do that we simply replaced images in the theme. The theme image files can be located in the following directory: \\jboss\server\default\deploy\ROOT.war\html\themes\classic\images\ {PORTLET_NAME}.

The final thing to do before we could get out to the customers for the demonstration was to create the survey that they should answer afterwards. A new meeting was held with our company supervisor and the marketing manager to get ideas for the survey. After we had constructed the survey, we sent it to our company supervisor that came up with some ideas about how some questions could be formulated. We changed our survey according to that feedback and we were now ready to go out to the customers for the demonstration of the implementation. The first part of the survey contained questions about the social media functions we had implemented, i.e.:

1. Rank from, 1 – 5, how useful is the RSS-function for you (1 is not useful, 5 is very useful)?
2. What could make the RSS-function more useful for you?
3. For what would you use the RSS-function?

RSS-function. The purpose of question one is to get a clear view of how useful the customers think the function is. The purpose of question 2 and 3 is to collect information about how the customers would use the functions and how the functions could be made even more useful to them.

The second part of the survey contained questions about other social media functions not demonstrated by us, i.e.:

1. Are there any other social media functions you see as useful to your organization?
2. If so, for what purpose would you use them?

The purpose of questions in the second part of the survey is to collect information that can be used for future work, such as if there are any other social media functions the customers think are useful to them.
Due to different reasons, we did only have the opportunity to visit one customer. We started the demonstration with an introduction and background to our study. After that we demonstrated the different parts of our implementation, answered and asked questions about it. When the demonstration was finished, we handed out a survey for the attending people to fill in about the implementation.

The last step in our study was to analyze both the answers we got from the survey and the interview. The result of the analysis can be found in chapter 5. Evaluation.
5 Evaluation

This chapter describes the evaluation process and the results of the evaluation.

To evaluate our implementation our plan was to visit Ida infront customers, demonstrate the implementation for them, and ask them about their thoughts about it. Because of time limitations we only had time to visit one customer, Stockholm’s stadsarkiv. We had a one hour meeting with them in which we demonstrated our implementation and discussed their thoughts on it, and they also filled in a survey. Below are their thoughts about the different functions, based on the survey and the discussion.

**RSS**

Stockholm’s stadsarkiv said that the RSS could be useful to them in a variety of ways. Firstly, they think that it can be used publically to report new events in a case. Secondly, it would be very useful to use it internally and filter cases on keywords and also be able to choose which case you want an RSS for. They also think it would be a good idea to have a clickable link to the case in the RSS feed.

**Wiki**

Since Stockholm’s stadsarkiv do not have that many employees, they stated that both the wiki would not be very useful to them at the moment, but they also said that it could very well become useful in the future as they employ more people.

**Chat**

As with the wiki, Stockholm’s stadsarkiv stated that it is not very useful to them since they do not have that many employees. However, they said that it could be somewhat useful for employees from different places to chat using the chat functionality.

The graph in figure 1 represents Stockholm’s stadsarkiv's answers to the question in the survey that asked them about how useful they think the particular function would be to them. A 5 is the best answer representing ”very useful” and 1 is the worst, ”not useful at all”.

![How useful would this function be to you?](image)

Figure 1: Stockholms stadsarkiv's rating of the social media
6 Discussion and conclusion

This chapter contains the discussion and conclusion.

6.1 Discussion

The goal for this study was to examine if social media functions could support users of case-handling systems. Social media is a very broad term, so to narrow down the number of possible functions to implement, a focus group meeting was held. Invitations were sent out to project managers, product developers and a marketing manager to see which social media functions they thought would be a good idea to implement and demonstrate for their customers. Due to different reasons, only three persons attended the focus group meeting. The low participation may have affected which social media functions we later decided to implement.

We decided to build the social media functionality using Liferay Portal. Liferay Portal is a solution with more or less already finished functions, so that choice may have affected our own ideas on how the social media functions should look and work.

The result of this study is based on the outcome of demonstrations of the implementation held to customers. We wanted to visit as many customers as possible so we could build our result on as much information as we could get. Due to different reasons, we only got the opportunity to demonstrate the implementation for one customer. This makes it hard to draw any general conclusions.

6.2 Conclusion of the study

The purpose of this study was to investigate if social media functions could be used to support the users of Ida infront's case-handling system. To do this, we did an implementation that was demonstrated to a customer of Ida infront. The requirements of specifications for the implementation were gathered from a focus group with Ida infront staff. To gather data from the customer, a survey was made that they answered after the demonstration.

Based on the result of the demonstration we held for a customer, social media functions can be used to support the users of case-handling systems, at least for that specific customer.

6.3 Future work

To be able to draw a more general conclusion about social media in case-handling systems, more demonstrations, surveys and interviews needs to be made. Future work can also involve looking at social media functions not covered in this study.
References


Appendix A

This appendix will contain information about how we configured our Liferay setup for the purpose of our demonstration.

To lower the time elapsed between the RSS-portlet would update, make the following change in portal-impl.jar com\liferay\portlet\RSS\util\RSSWebCacheItem.class

change long_value=1200000 (20 minutes) to long_value=10000 (10 seconds)

To remove summary, categories, permissions and tags from the wiki we removed this in html\portlet\wiki\edit_page.jsp:

```html
<aui:input classPK="<%= classPK %>"> name="categories"
  type="assetCategories" />
<aui:input classPK="<%= classPK %>"> name="tags" type="assetTags" />
<aui:input name="summary" />
<c:if test="<%= newPage %>">
  <aui:field-wrapper label="permissions">
    <liferay-ui:input-permissions
      modelName="<%= wikiPage.class.getName() %>">
    </liferay-ui:input-permissions>
  </aui:field-wrapper>
</c:if>
```

To remove ratings of wiki pages you can remove following in html\portlet\wiki\view.jsp

```html
<c:if test="<%= enablePageRatings %>">
  <div class="page-ratings">
    <liferay-ui:ratings
      className="<%= wikiPage.class.getName() %>">
      classPK="<%= wikiPage.getResourcePrimKey() %>">
    </liferay-ui:ratings>
  </div>
</c:if>
```

To change the name on the front wiki page we added the following configuration in portal-ext.properties:

```
wiki.front.page.name={INSERT DESIRED NAME HERE}
```
To change what should be shown in the chat portlets buddy-list we made a change in this properties file: server\default\deploy\chat-portlet.war\WEBINF\classes\portlet.properties

Default is:
include-and-override=portlet-ext.properties

buddy.list.strategy=all
#buddy.list.strategy=communities
#buddy.list.strategy=friends
#buddy.list.strategy=communities,friends

To get the buddy list to only show members in the same community, change to this:
include-and-override=portlet-ext.properties

#buddy.list.strategy=all
buddy.list.strategy=communities
#buddy.list.strategy=friends
#buddy.list.strategy=communities,friends

To deploy Liferay on a jboss-server, we performed the following steps:

1) Download liferay-portal-6.0.5.war
2) Download liferay portal dependencies and extract it to jboss\server\default\lib
3) Delete all content in jboss\server\default\deploy\ROOT.war
4) Extract liferay-portal-6.0.5.war in jboss\server\default\deploy\ROOT.war
5) Delete the following files from jboss\server\default\deploy\ROOT.war\lib
   • jaxrpc.jar
   • stax.jar
   • xercesImpl.jar
   • xml-apis.jar
6) Delete hibernate-validator.jar from jboss\common\lib
Appendix B

This appendix contains the code we produced when making our first servlet. Servlet.java contains the actual servlet code to extract information from iipax permission. Web.xml contains settings for servlet name and basic authentication while jboss-web.xml contains the security domain information.

Servlet.java

```java
package ida.iipax.test;
import iipax.service.casekernel.client.api.CaseCallerFactory;
import iipax.service.casekernel.client.api.e.xception.AccessDeniedException;
import iipax.service.casekernel.client.api.iface.CaseTodoItem;
import iipax.service.casekernel.client.api.iface.WorkPool;
import java.io.IOException;
import java.util.Collection;
import java.util.Iterator;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class Servlet extends HttpServlet {
    private static final long serialVersionUID = 1L;
    public Servlet() {
        super();
    }
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        CaseCallerFactory factory = CaseCallerFactory.getFactory();
        WorkPool workPool = factory.getWorkPool();
        try {
            Collection<CaseTodoItem> ctdi = workPool.getTodo();
            Iterator<CaseTodoItem> iterator = ctdi.iterator();
            while(iterator.hasNext()) {
                CaseTodoItem tmp = iterator.next();
                //Test to print something from iipax
                System.out.println(tmp.getDescription());
            }
        } catch (AccessDeniedException e) {
            System.err.println("AccessDeniedException in IipaxTest
doGet");
            e.printStackTrace();
        }
    }
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    }
}
```

25
response) throws ServletException,
IOException {
}
}

web.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://java.sun.com/xml/ns/javaee"
xn:web="http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
xn:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd" id="WebApp_ID"
version="2.5">
  <display-name>IipaxTest</display-name>
  <welcome-file-list>
    <welcome-file>index.html</welcome-file>
    <welcome-file>index.htm</welcome-file>
    <welcome-file>index.jsp</welcome-file>
    <welcome-file>default.html</welcome-file>
    <welcome-file>default.htm</welcome-file>
    <welcome-file>default.jsp</welcome-file>
  </welcome-file-list>
  <servlet>
    <description/>
    <display-name>Servlet</display-name>
    <servlet-name>Servlet</servlet-name>
    <servlet-class>ida.iipax.test.Servlet</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>Servlet</servlet-name>
    <url-pattern>/Servlet</url-pattern>
  </servlet-mapping>
  <!-- add Basic authentication to the servlet, needed because we
   need to be logged in to the appserver before we can use the
   iipax API -->
  <security-constraint>
    <display-name>SecurityConstraint</display-name>
    <web-resource-collection>
      <web-resource-name>Servlet</web-resource-name>
      <url-pattern>/Servlet</url-pattern>
    </web-resource-collection>
    <auth-constraint>
      <role-name>iipaxUser</role-name>
    </auth-constraint>
    <user-data-constraint>
      <transport-guarantee>NONE</transport-guarantee>
    </user-data-constraint>
  </security-constraint>
  <login-config>
    <auth-method>BASIC</auth-method>
    <realm-name>iipax</realm-name>
  </login-config>
  <security-role>
    <role-name>iipaxUser</role-name>
  </security-role>
</web-app>
```
jboss-web.xml

<jboss-web>
  <security-domain>java:/jaas/iipax</security-domain>
</jboss-web>
Appendix C

This appendix contains the code we produced when making our servlet RSS feed. Servlet.java contains method calls to generate RSS and also starts a timer to refresh the feeds. The actual code to extract information from iipax permission and also the code to make it into RSS feeds is now in GenerateRSS.java. Web.xml contains settings for servlet name and basic authentication while jboss-web.xml contains the security domain information.

Servlet.java

```java
package ida.socialmedia.test;

import java.io.IOException;
import java.util.Timer;
import javax.annotation.security.DeclareRoles;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@DeclareRoles("iipaxAdm")
public class Servlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    public Servlet() {
        super();
    }

    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        //Save the paths to each xml file
        String workPool = getServletContext().getRealPath("/workpool.xml");
        String events = getServletContext().getRealPath("/events.xml");
        String recordEvents = getServletContext().getRealPath("/recordevents.xml");

        //create a timer and set it to generate RSS every five seconds
        Timer t = new Timer();
        GenerateRSS generate_RSS = new GenerateRSS(workPool, events, recordEvents);
        t.scheduleAtFixedRate(generate_RSS, 0, 1*5*1000);
    }

    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    }
}
```
GenerateRSS.java

package ida.socialmedia.test;

import iipax.generic.objectbase.client.api.iface.Query;
import iipax.generic.objectbase.client.api.exception.NoSuchObjectException;
import iipax.generic.objectbase.client.impl.QueryImpl;
import iipax.service.casekernel.client.api.CaseCallerFactory;
import iipax.service.casekernel.client.api.exception.AccessDeniedException;
import iipax.service.casekernel.client.api.iface.CaseObject;
import iipax.service.casekernel.client.api.iface.CaseRecordEventInfo;
import iipax.service.casekernel.client.api.iface.CaseRepository;
import iipax.service.casekernel.client.api.iface.CaseTodoItem;
import iipax.service.casekernel.client.api.iface.EventLog;
import iipax.service.casekernel.client.api.iface.Journal;
import iipax.service.journal.client.api.event.Event;
import iipax.service.journal.client.api.event.RecordEvent;
import java.io.FileWriter;
import java.io.IOException;
import java.util.ArrayList;
import java.util.Collection;
import java.util.Date;
import java.util.Iterator;
import java.util.List;
import java.util.TimerTask;
import com.sun.syndication.feed.synd.SyndContent;
import com.sun.syndication.feed.synd.SyndContentImpl;
import com.sun.syndication.feed.synd.SyndEntry;
import com.sun.syndication.feed.synd.SyndEntryImpl;
import com.sun.syndication.feed.synd.SyndFeed;
import com.sun.syndication.feed.synd.SyndFeedImpl;
import com.sun.syndication.io.FeedException;
import com.sun.syndication.io.SyndFeedOutput;

public class GenerateRSS extends TimerTask {

    //Path names for the temporary xml RSS files
    String fileName, fileNameEvents, fileNameRecordEvents;

    //Arrays with iipax items
    CaseTodoItem[] caseTodoItems;
    Event[] caseEvents;
    CaseRecordEventInfo[] caseRecordEvents;

    public GenerateRSS(String fileName, String fileNameEvents, String fileNameRecordEvents) {
        this.fileName = fileName;
        this.fileNameEvents = fileNameEvents;
        this.fileNameRecordEvents = fileNameRecordEvents;
        caseTodoItems = new CaseTodoItem[0];
        caseEvents = new Event[0];
        caseRecordEvents = new CaseRecordEventInfo[0];
    }

    //Function to create the iipax workpool RSS
    public void generateWorkPoolRSS(String fileName) {

    }

}
try {
    //Create filewriter with the correct path file name
    FileWriter writer = new FileWriter(fileName);

    //Set RSS feed info
    SyndFeed feed = new SyndFeedImpl();
    feed.setFeedType("RSS_2.0");
    feed.setTitle("iipaxWorkPool");
    feed.setLink("http://localhost:8080");
    feed.setDescription("RSS av iipax todo");

    //List to put feed entries in
    List<SyndEntry> entries = new ArrayList<SyndEntry>();

    int size = caseTodoItems.length - 1;

    //Loop through all case items (backwards to get latest added item first)
    for(int i = size; i >= 0; i--)
    {
        //iipax todo-item
        CaseTodoItem tmp = caseTodoItems[i];

        //iipax case object which the todoitem concerns
        CaseObject co = tmp.getCase();

        //Clean up the getcreateddate a bit
        String date = co.getCreatedDate().toString();
        date = date.replaceAll("CEST", "");

        //Create and format a description string(html code) from the item
        String desc = (<i>Status:
        </i>"+tmp.getStatusComment()+
        "<br><i>&Auml;rendemening:
        </i>"+co.getSentence()+
        "<br><i>Skapad: </i>"+date+
        "<br><i>Skapad av:
        </i>"+co.getCreatedBy());

        //Set the feed entry and description info
        SyndEntry entry = new SyndEntryImpl();
        SyndContent description = new SyndContentImpl();
        entry.setTitle(tmp.getDescription());
        entry.setLink("");
        entry.setDescription(description);
        entries.add(entry);

        description.setType("text/plain");
        description.setValue(desc);
    }

    feed.setEntries(entries);

    //Output the feed to file and close the filewriter
    SyndFeedOutput output = new SyndFeedOutput();
    output.output(feed, writer);
    writer.close();
}
catch (IOException e) {  
    System.err.println("IOException in workpool RSS");  
    e.printStackTrace();  
} catch (FeedException e) {
    System.err.println("FeedException in workpool RSS");
    e.printStackTrace();
}

// Function to create the iipax event RSS
public void generateEventRSS(String fileName) {
    try {
        FileWriter writer = new FileWriter(fileName);
        SyndFeed feed = new SyndFeedImpl();
        feed.setFeedType("RSS_2.0");
        feed.setTitle("iipaxEvents");
        feed.setLink("http://localhost:8080");
        feed.setDescription("RSS av Events");

        List<SyndEntry> entries = new ArrayList<SyndEntry>();

        int size = caseEvents.length - 1;

        for(int i = size; i >= 0; i--)
        {
            Event tmp = caseEvents[i];
            Date realDate = tmp.getEventDate();
            String date = realDate.toString();
            date = date.replace("CEST", "")

            String desc = (<i>Typ: </i>" + tmp.getType() +
                        "<br><i>Kommentar: </i>" +
                        tmp.getComment() +
                        "<br><i>Vem: </i>" + tmp.getWho() +
                        "<br><i>Vad: </i>" + tmp.getWhat() +
                        "<br><i>Tid: </i>" + date);

            SyndEntry entry = new SyndEntryImpl();
            SyndContent description = new SyndContentImpl();

            entry.setTitle("Notering: " + date);
            entry.setLink(""");
            entry.setPublishedDate(realDate);
            entry.setDescription(description);
            entries.add(entry);

            description.setType("text/plain");
            description.setValue(desc);

        }

        feed.setEntries(entries);

        SyndFeedOutput output = new SyndFeedOutput();
        output.output(feed, writer);
    }
    catch (IOException e) {  
        System.err.println("IOException in workpool RSS");  
        e.printStackTrace();  
    } catch (FeedException e) {
        System.err.println("FeedException in workpool RSS");
        e.printStackTrace();
    }
}
try {
    FileWriter writer = new FileWriter(fileName);

    SyndFeed feed = new SyndFeedImpl();
    feed.setFeedType("RSS 2.0");
    feed.setTitle("iipaxRecordEvents");
    feed.setLink("http://localhost:8080");
    feed.setDescription("RSS av RecordEvents");

    List<SyndEntry> entries = new ArrayList<SyndEntry>();
    int size = caseRecordEvents.length - 1;
    for(int i = size; i >= 0; i--)
    {
        CaseRecordEventInfo tmp = caseRecordEvents[i];
        RecordEvent re = tmp.getRecordEvent();

        Date realDate = tmp.getEventDate();
        String date = realDate.toString();
        date = date.replace("CEST", "")
        String desc = (<i>Typ: </i>" + tmp.getCaseType() +
        "<br><i>Mening: </i>" +
        tmp.getCaseSentence() +
        "<br><i>Kommentar: </i>" +
        re.getComment() +
        "<br><i>Vem: </i>" + tmp.getWho() +
        "<br><i>Vad: </i>" + tmp.getWhat() +
        "<br><i>Tid: </i>" + date);

        SyndEntry entry = new SyndEntryImpl();
        SyndContent description = new SyndContentImpl();
        entry.setTitle("Tjänsteanteckning: " + date);
        entry.setLink("");
        entry.setPublishedDate(realDate);
        entry.setDescription(description);
        entries.add(entry);
        description.setType("text/plain");
        description.setValue(desc);
    }
    feed.setEntries(entries);
    SyndFeedOutput output = new SyndFeedOutput();
}
public void updateData() {

    //get everything needed to update events(eventLog), recordevents(journal),
    //workpool(workPool)
    CaseCallerFactory factory = CaseCallerFactory.getFactory();
    EventLog eventLog = factory.getEventLog();
    Journal journal = factory.getJournal();
    Query eventFilter = new QueryImpl();
    WorkPool workPool = factory.getWorkPool();

    try {
        //Clear the arrays and repopulate with current data
        caseTodoItems = new CaseTodoItem[0];
        caseEvents = new Event[0];
        caseRecordEvents = new CaseRecordEventInfo[0];
        caseTodoItems = workPool.getTodo().toArray(caseTodoItems);
        caseEvents = eventLog.searchEvents(eventFilter).toArray(caseEvents);
        caseRecordEvents = journal.searchRecordEvents(eventFilter).toArray(caseRecordEvents);
    }
    catch (AccessDeniedException e) {
        System.err.println("AccessDeniedException in GenerateRSS/UpdateData!");
        e.printStackTrace();
    }
}

//Function that updates data and creates the RSS xml files
@Override
public void run() {
    updateData();
    generateWorkPoolRSS(fileName);
    generateEventRSS(fileNameEvents);
    generateRecordEventRSS(fileNameRecordEvents);
}

web.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://java.sun.com/xml/ns/javaee"
xmlns:web="http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd" id="WebApp_ID"
version="2.5">
  <display-name>SocMedia</display-name>
  <welcome-file-list>
    <welcome-file>index.html</welcome-file>
    <welcome-file>index.htm</welcome-file>
    <welcome-file>index.jsp</welcome-file>
    <welcome-file>default.html</welcome-file>
    <welcome-file>default.htm</welcome-file>
    <welcome-file>default.jsp</welcome-file>
  </welcome-file-list>
  <servlet>
    <description></description>
    <display-name>Servlet</display-name>
    <servlet-name>Servlet</servlet-name>
    <servlet-class>ida.socialmedia.test.Servlet</servlet-class>
    <load-on-startup>1</load-on-startup>
  </servlet>
  <servlet-mapping>
    <servlet-name>Servlet</servlet-name>
    <url-pattern>/Servlet</url-pattern>
  </servlet-mapping>
  <!-- map the servlet to the url /Servlet -->
  <security-constraint>
    <display-name>SecurityConstraint</display-name>
    <web-resource-collection>
      <web-resource-name>Servlet</web-resource-name>
      <url-pattern>/Servlet</url-pattern>
    </web-resource-collection>
    <auth-constraint>
      <role-name>iipaxUser</role-name>
    </auth-constraint>
    <user-data-constraint>
      <transport-guarantee>NONE</transport-guarantee>
    </user-data-constraint>
  </security-constraint>
  <login-config>
    <auth-method>BASIC</auth-method>
    <realm-name>iipax</realm-name>
  </login-config>
  <security-role>
    <role-name>iipaxUser</role-name>
  </security-role>
</web-app>

jboss-web.xml

```
Appendix D

To prevent inactivity in the portal to cause the user to log out and also disable any warnings, we added the following configurations in portal-ext.properties:

- session.timeout=600
- session.timeout.warning=1
- session.timeout.auto.extend=true

Following configurations were made in portal-ext.properties:

**To enable maximizing of portlets**
layout.guest.show.max.icon=true

**To disable the user's public pages**
layout.user.public.layouts.enabled=false

**To disable password reminders**
users.reminder.queries.enabled=false
users.reminder.queries.custom.question.enabled=false

To disable toggle edit controls, the following was removed in
jboss\server\default\deploy\ROOT.war\html\portlet\dockbar\view.jsp

```
<c:if test="!group.isControlPanel() && themeDisplay.isSignedIn()">
  <li class="toggle-controls" id="<portlet:namespace />toggleControls">
    <a href="javascript:;">liferay-ui:message key="toggle-edit-controls" /></a>
  </li>
</c:if>
```

**To disable terms of use**
terms.of.use.required=false
På svenska

Detta dokument hålls tillgängligt på Internet – eller dess framtida ersättare – under en längre tid från publiceringsdatum under förutsättning att inga extra-ordinära omständigheter uppstår.

Tillgång till dokumentet innebär tillstånd för var och en att läsa, ladda ner, skriva ut enstaka kopior för enskilt bruk och att använda det oförändrat för ickekommersiell forskning och för undervisning. Överföring av upphovsrätten vid en senare tidpunkt kan inte upphäva detta tillstånd. All annan användning av dokumentet kräver upphovsmannens medgivande. För att garantera äktheten, säkerheten och tillgängligheten finns det lösningar av teknisk och administrativ art.

Upphovsmannens ideella rätt innefattar rätt att bli nämnd som upphovsman i den omfattning som god sed kräver vid användning av dokumentet på ovan beskrivna sätt samt skydd mot att dokumentet ändras eller presenteras i sådan form eller i sådant sammanhang som är kränkande för upphovsmannens litterära eller konstnärliga anseende eller egenart.

För ytterligare information om Linköping University Electronic Press se förlagets hemsida http://www.ep.liu.se/

In English

The publishers will keep this document online on the Internet - or its possible replacement - for a considerable time from the date of publication barring exceptional circumstances.

The online availability of the document implies a permanent permission for anyone to read, to download, to print out single copies for your own use and to use it unchanged for any non-commercial research and educational purpose. Subsequent transfers of copyright cannot revoke this permission. All other uses of the document are conditional on the consent of the copyright owner. The publisher has taken technical and administrative measures to assure authenticity, security and accessibility.

According to intellectual property law the author has the right to be mentioned when his/her work is accessed as described above and to be protected against infringement.

For additional information about the Linköping University Electronic Press and its procedures for publication and for assurance of document integrity, please refer to its WWW home page: http://www.ep.liu.se/

© Daniel Jans & Johan Juel