Metropolia

Kalle Kallio

Re-engineering an Image Archive User Interface

Helsinki Metropolia University of Applied Sciences
Bachelor of Engineering
Media Engineering
Thesis
25 May 2011

Tekijä Otsikko Sivumäärä Aika	Kalle Kallio Kuva-arkiston käyttöliittymän uudistaminen 31 sivua + 2 liitettä 25.5.2011
Tutkinto	Bachelor of Engineering
Koulutusohjelma	Media Engineering
Suuntautumisvaihtoehto	digitaalinen media
Ohjaajat	suunnittelupäällikkö Heikki Lahdenmäki yliopettaja Erkki Aalto

Insinöörityössä oli tavoitteena suunnitella ja kehittää kuva-arkistolle uusi käyttöliittymä. Kuva-arkisto toimii selaimen kautta sisäverkon palvelimella. Työn koon rajaamiseksi päätettiin keskittyä käyttöliittymään ja jätettiin pois muutokset tietokantaan.

Kuva-arkiston käyttöliittymä päätettiin suunnitella käyttäjälähtöisesti. Suunnitelmaa varten käyttäjiä haastateltiin ennalta päätettyjen kysymysten pohjalta, mutta kuitenkin vapaamuotoisesti. Haastattelujen pohjalta selvisi, että arkistolla on kahdenlaisia käyttäjiä; arkkitehtejä ja matkailualan toimijoita. Molempien vaatimukset arkiston suhteen ovat kuitenkin samankaltaisia. Käyttäjät haluavat etsiä arkistosta tietynlaisia kuvia tai kuvia tietystä kohteesta. Käyttäjien työnkuvaan kuuluu myös kuvien lisääminen arkistoon.

Haastattelun perusteella suunniteltiin käyttöjärjestelmän eri sivut. Hakusivulla voivat käyttäjät määritellä hakuehdot ja hakea kuvia niiden perusteella. Hakuehdot johtavat hakutulosnäkymään, jossa näkyy osa hakutuloksista. Hakutuloksia voi tarkastella nopeasti siirtämällä hiiren osoittimen kuvan päälle, jolloin sivun laidassa oleva laatikko näyttää kuvan suurempana ja kuvan tallennettuja tietoja. Hakutuloksista pääsee kuvaa napsauttamalla selausnäkymään, jossa kuvat näkyvät suurina ja käyttäjä voi siirtyä sarjassa seuraaviin tai edellisiin kuviin. Selausnäkymässä voi myös muokata kuvan tallennettuja tietoja ja tallentaa kuvan käyttäjän tietokoneelle.

Kuvien lisäämistä arkistoon päätettiin helpottaa lisäämällä arkistoon kuvien tietojen muokkausjärjestelmä, jossa käyttäjä voi yhdellä napsautuksella tallentaa kuvaan tietoja. Kuvat lisätään uudessa käyttöliittymässä arkistoon suoraan käyttäjän koneelta, kuten esimerkiksi liitetiedostot sähköpostiin.

Toteutettu käyttöliittymä on havaittu käytännössä toimivaksi kokonaisuudeksi. Merkittäviä parannuksia edelliseen verrattuna ovat muun muassa kuvien tallentaminen käyttäjän tietokoneelle ja kuvien tietojen nopea tarkastelu hakutulosnäkymässä. Uuden käyttöliittymän käyttöönoton jälkeen palaute on ollut pääosin myönteistä.

Avainsanat	kuva-arkisto, valokuva, käyttöliittymä, haku, metadata

Metropolia

Abstract

Author(s) Title Number of Pages Date	Kalle Kallio Re-engineering an image archive user interface 31 pages + 2 appendices 25 May 2011
Degree	Bachelor of Engineering
Degree Programme	Media Engineering
Specialisation option	Digital Media
Instructor(s)	Heikki Lahdenmäki, Head of Department Erkki Aalto, Principal Lecturer

The goal of the thesis was to design and develop a new user interface for an image archive. The archive is accessed on the intranet server through the user's web browser. To limit the scope of the thesis, no changes to the database were planned.

The user interface was set to be designed from a user centred point of view. For the design, the users were interviewed. The interview was based on a set group of questions, but the line of questions was not strictly followed during the interview. From the interview it was found out that the image archive has two kinds of users, architects and users working in tourism and travel. Both types of have similar requirements for the archive. The users search the archive for images of a certain building or certain types of images. The users are also required to upload images to the archive.

Individual pages of the user interface were designed based on the interviews. On the search page the users can define search conditions and search for images based on the conditions. The search page leads to the results page, which displays search results as thumbnails. The user can inspect images by placing the mouse cursor on top of the thumbnail, which causes metadata for the image to be displayed on a box on the right. From the results page the user can enter a browse view by clicking on a thumbnail. On the browse page the user can see the single image in a larger size and navigate to the previous and next images. In the browse page the user can also edit saved metadata and export images from the archive.

Adding images to the archive was were decided to be made easier by adding an batch edit system to the archive, where the user can write metadata information to an image by a single mouse click. The images are added to the archive directly from the user's computer.

After development and deployment, the archive user interface has been observed to be useful and functional. Significant improvements include quickly browsing through multiple images and saving images to the user's computer. Feedback from the new archive user interface from the users has been mostly positive.

Keywords	image archive, user interface, HCI, search, metadata

Contents

1 Introduction

> "The national heritage managed by the Governing Body encompasses more than 200 buildings and over 80 hectares of land on seven islands. " "The Suomenlinna fortress was listed as a UNESCO World Heritage Site in 1991 as an example of European fortress architecture. As the host of a World Heritage site, the Governing Body of Suomenlinna is responsible for preserving the monument and associated historical and architectural values for future generations." [1]

During the four decades of operation the Governing Body of Suomenlinna has taken thousands of pictures from the Sea Fortress for purposes of archiving, travel and tourist information and to track the process of construction and restoration on several different locations. An electronic image archive was formed year 2000 to store digital images and scanned photographic slides. It now has 60,000 images with associated metadata information. The metadata information consists of user added information mostly describing content of the image. The PostgreSQL metadata archive and the image files located on a CentOS GNU/Linux server are accessed through a PHP web interface.

The original user interface for the archive was made from a technical point of view. The search view and the search results were constructed based on the database model and not based on the requirements of the users. For example the images shown by a search are ordered by the order they are in the database. This might not have anything to do with the content of the images or the associated metadata. However, for some purposes, the order performs well.

The image archive user interface has been extended throughout the years by three different people, out of which everyone used a different set of standards. The archive lacks a set of standards and practices. The differences also extend to the visual appearance of the user interface; most pages look a bit different from each other. Some of the features present on the user interface are relics from the early versions; the features have since become obsolete but have not been omitted from the interface.

2 Image Archives

There are multiple valid ways of constructing an image archive. As image archives have been around for much longer than computers and their databases, there are multiple examples of image archives that are not electronic. The simplest archive can be just a collection of printed images, not stored in any particular order. A more complex traditional archive can involve a set of cards or other system to store information along the images, and the images can be set in some defined order.

2.1 Database Technology

A database is a collection of related data. It should be logically coherent and built and designed for a specific purpose; a collection of random data cannot be referred to as a database. A database management system is a collection of software and tools which allow the users to create and maintain a database. Usually the database is self-describing in nature; the database contains a catalogue which contains information about the database's own structure. [2]

Databases commonly use SQL (Structured Query Language) to access the database content. SQL can be used in a wide variety of ways to manipulate a database and the data within. Most queries are related to retrieving or updating data in the database. SQL abstracts the physical database structure; the operator does not need to know physical locations of data as the operator can use queries to manipulate and retrieve the data.

Databases have several advantages over other systems of storing data. The data is separate from the systems which use the data; there can be one or several such systems. Access to the database can be restricted or granted based on the system accessing it. Databases can also be easily extended without major changes to the systems accessing the data.

2.2 Database Requirements for Image Archives

Image archives have similar requirements as other databases. The information must obviously stay intact and not become corrupted in any way. Editing, updating and most importantly retrieving the information needs to be possible.

One of the largest differences between image archive as a database and other databases is that the image archive database can possibly contain the images themselves. The image is significantly larger amount of data than simple string values. One image can be as large as an entire database. Databases can support storing large unstructured files as binary large objects, also known as BLOBs. Alternatively, the images can be stored in the server as individual files and the image database can only contain the references to the physical images.

2.3 Amount of Images

Figure 1 shows the cumulative image amounts in the archive. The graph is based on the date when the images were recorded to be taken, not on when they were added to the archive. According to the graph ten thousand images were taken before 1980's, but this is a mistake as images with unknown dates are recorded to be taken at year zero. A quick search at the image archive reveals the amount of photographs taken before 1980's to be around four thousand. The amount of images in the archive taken before 1995 is twenty thousand. Reaching twice that, forty thousand, took about ten years more. Gaining another twenty thousand took only five years between 2005 and 2010.

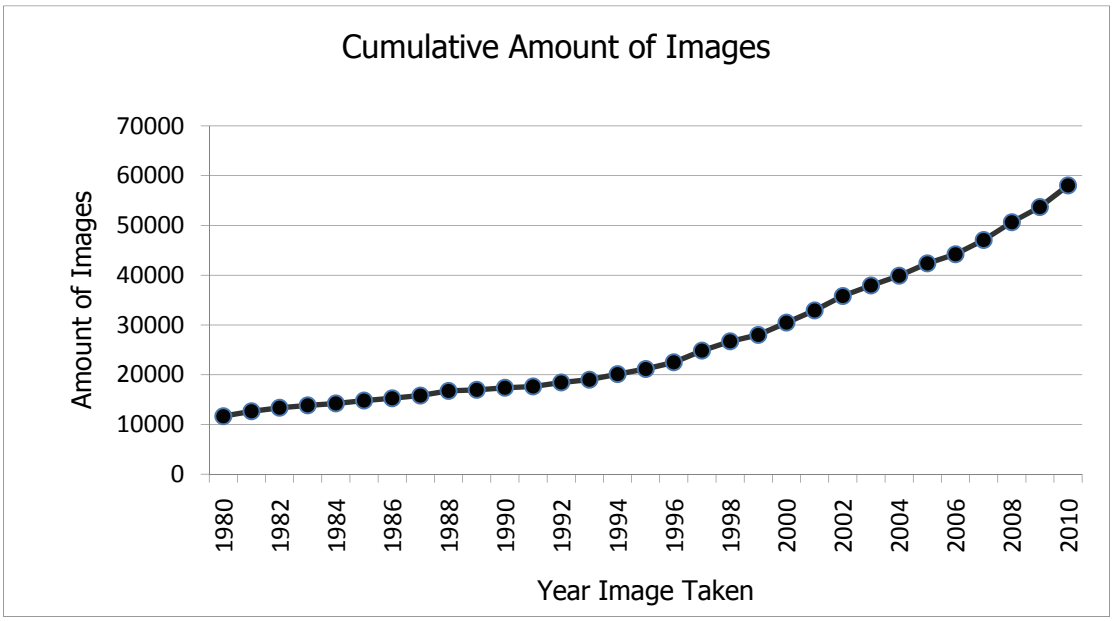

Figure 1. Cumulative amount on images in the Governing Body of Suomenlinna image archive.

The amount of images stored to the archive has been growing every year, which is to be expected as images are rarely removed. What is notable is the rate of the growth, which seems to be increasing all the time. Figure 2 shows the amount of new images in during a year.

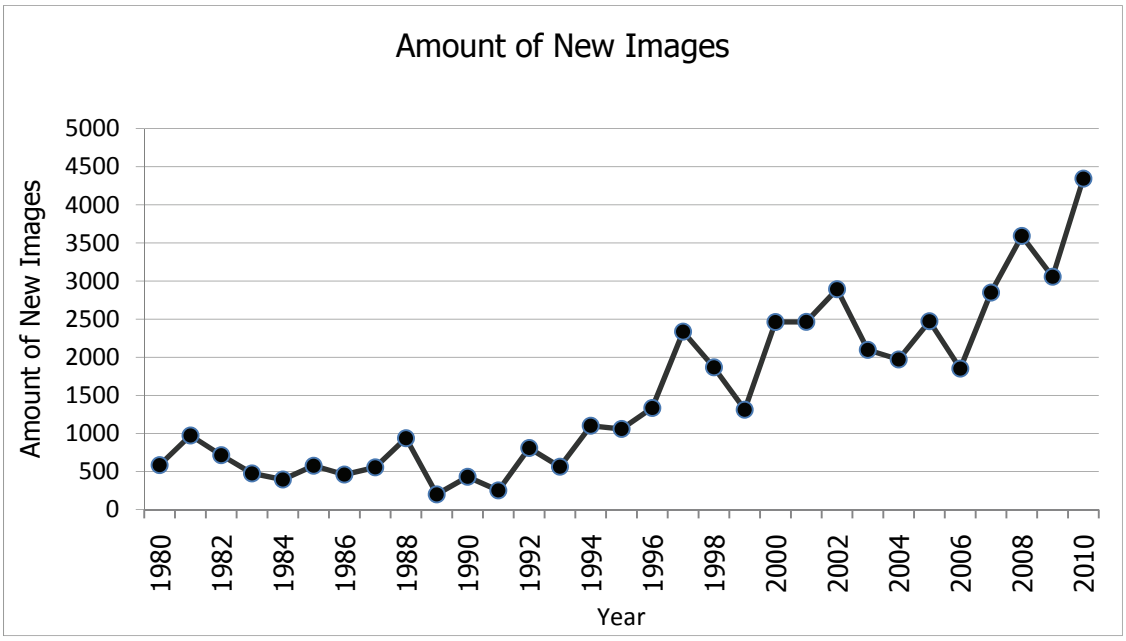

Figure 2. Amount of new images per year.

Only around five hundred images made it to the archive in year 1980 where as in year 2,000 around 2,500 new images were added to the archive. From the graph it is apparent that the amount of images added to the archive every year is on a steady rise. This is likely due to the introduction of digital cameras; taking images is cheaper and easier and they are transferred to the computer in a fraction of the effort in comparison to analogue images.

2.4 Future Image Amounts

The trend on the amount of images which the archive needs to store will probably not end very soon. It is possible to extrapolate from the amount of images the expected amounts in the future.

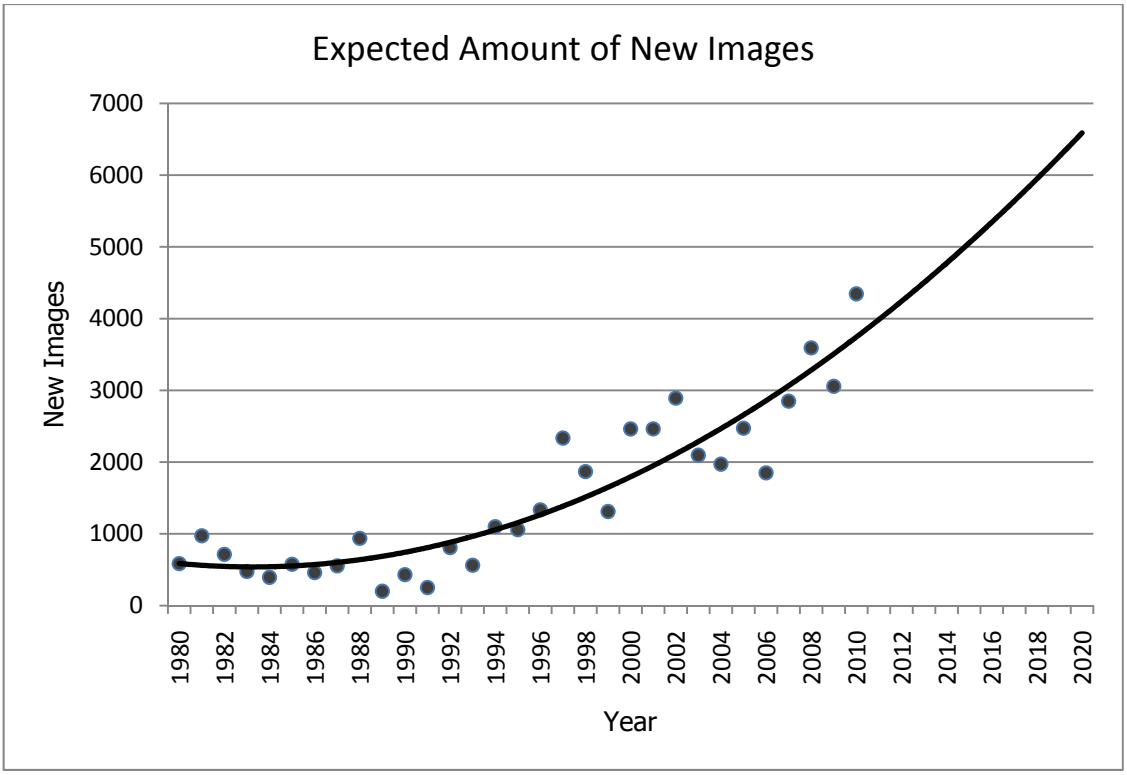

Figure 3. Expected amount of new images per year.

Figure 3 shows the extrapolated trend line. According to the extrapolated amount of new images, there should be 6,500 new images added to the archive in year 2020. Following this trend would cause the archive to hold 95,000 images, up from 60,000, by year 2020.

As the amount of images rises, the amount of work required to add the images to the archive rises equally. Therefore, the processes required by adding new images to the archive must be gradually improved so the archive does not become unmanageable.

3 Defining User Interface Requirements

3.1 User Interview

In order to find out requirements for the re-engineered user interface I designed a set of interview questions and asked the actual image archive users the questions. The goal was to find out what kind of images the users search from the archive, and how they perform the search. I also asked a set of questions related to adding images to the archive. The list of questions is on appendix 1.

3.2 Two Types of Users

From the replies I came to the conclusion that there are two different types of users: the first group belongs to the tourism industry, the other group are architects who deal with the buildings and constructions. The people involved in the tourism and publishing mostly search for images that are something they can use in a publication. Images in this category must be reasonably good resolution and technically correct. The images must also be available for The Governing Body of Suomenlinna to publish. The architects mostly search for images that describe a state a building was earlier or to track the process of construction. It is not always possible to completely accurately describe the process of a building on a document; the images extend the information, which is present in the documents created through meetings and construction supervisor. The images can also serve as additional information to the blueprints; if the blueprint is unclear at a portion, an image, which shows that portion, could be found from the image archive without actually visiting the building.

3.3 Results from Questions on Retrieving Images

Generally people found the images they were looking for. Searching by the island code and building number is an easy and accurate way to find images of a certain building. Quite often the users were overwhelmed by the amount of images; there can be several thousand for one building, out of which most can be irrelevant for the task at hand. There were requests for simple functionalities, for example ordering the search results by date when the images were taken.

Most users found the 'text search' field confusing; it does not work along the modern standards. For example by inserting 'A B' on the search field when performing a search the database would return results which only contain the exact 'A B' string. All images containing only 'A' or 'B' would be completely left out. Often people expected to find more images by inserting more words to the search but instead found nothing.

A couple less experienced users had trouble navigating the image archive. They often could not find back to the search window or accidentally closed the tab on the browser they had used. People also often had trouble understanding the meaning of tags you can toggle on and off to describe an image, such as the tag "historical".

3.4 Uploading Images to the Archive

Uploading images to the archive is much more hard work than simply searching for pictures from the archive. In the first version of the archive the process is as follows:

1. Decide which pictures to upload.
2. Move the images to a special directory k:/diaupload.
3. Select upload from the image archive and type your username to the field.
4. Fill in the metadata information to be written on all of the images to be uploaded.
5. If individual image metadata needs to be changed or adjusted, as usually does, the user must use the search window to find the recently uploaded images.

Most people who have uploaded images to the archive did not really comment much on whether they would appreciate if uploading was faster, and users did not state that

uploading images to the archive is too difficult. However, not many of the users are very motivated on uploading their images to the archive. Storing the images in temporary directories on various hard drives is a much easier and faster process than uploading the images to the archive. Many users were concerned that they would not find images they had uploaded to the archive.

Many users probably had just accepted how the process of adding images to the archive works and did not really question the process. For example, the use of the special folder on a network drive for uploading images is an outdated method.

3.5 Open Discussions and Personal Experience

I was originally hired to the Governing Body to transfer images from the network drive to the archive and I have added over 3,000 images to the archive, most of which are from building construction. During that time I've had several discussions about the user interface of the archive. When planning this project I also had several discussions with the chief architect. Some of the suggestions were good as they were, for example sorting of images returned by a search based on something else than the order they are stored in the database.

4 Meeting the Requirements

Interaction design can be seen to consist of four different parts. Identifying needs and establishing requirements, developing designs to meet the requirements, building interactive versions so the designs can be tested and finally evaluating the designs to measure if the designs are acceptable. [3] In order to fully understand and design based on the defined needs of the users, I constructed a concept map, as shown in appendix 2, which defines and groups the requirements for the user interface. In the next topics I will go through the features and design principles which should meet the requirements for the new image archive user interface.

4.1 Search Criteria

“Same possible search criteria as before in the archive.” The new user interface should provide all the search functionality of the previous user interface, in addition to new features. Images found earlier by a certain type of query should still be found using the same query. The exclusion to this is the user defined SQL query. The existing user interface had a possibility for the user to submit their own SQL query by using a textarea input field. This was omitted all together as most people do not know how to construct SQL queries and the people who can have access to the database to begin with.

“Text search.” The users can search for words from the 'kuvateksti' and 'diateksti' fields. The inputted search string should not be interpreted as a strict string value. If the user had inputted two words separated by space, the inputted string should be interpreted as two different search components and not simply as a strict string with a space in it. We considered allowing the user to decide whether both the search terms should be included or if either of them is fine, but the feature was omitted in the end.

“Include and exclude tags.” The user should be able to search for images that have a certain tags. As a new feature, the user should be able to exclude images with certain tags.

4.2 Display of Search Results

“Results as thumbnails.” The search results will be visible as thumbnails. The amount of thumbnails visible should scale based on the user's browser and screen size, so that the amount of thumbnails would fill up the available space. Later on in the development this feature was left out as it was too problematic to develop. [4]

“Metadata easier visible.” The user should be able to view the stored metadata information from the thumbnail view. The existing user interface has a mouseover functionality, which shows some of the metadata. This function was something which the users found helpful and should be developed further.

"Sorting." The images should be sorted based on the image information. Originally there were several different plans on how to sort the images but most of these methods would not have actually worked well. In the end the only sorting was based on the order the images are added on the database, and on the date when the picture was taken. Omitted sorting options included the possibility to sort images based on the buildings metadata or the 'kuvateksti' field. There also were ideas to construct tag clouds from the content on the 'kuvateksti' field but this proved to not be viable due to the irregular contents of the 'kuvateksti' field.

4.3 Browsing Images

"Browsing from image to the next." When browsing images, the user should be able to easily navigate between the next and previous images. This can be done by allowing the user to switch to the next image by pressing next or previous buttons on the page, or by using the arrow keys on their keyboard.

"Also displays other search results." In addition to the currently viewed image, the user should be able to see some of the other results. The user then knows which images have been viewed and which are next ones, and the user can to easily skip to one of the previous or next images. This solves a problem in browsing when there are multiple images that are very similar in content.

"Easily edit metadata of single image but not by accident." The user should be able to edit the metadata on the currently viewed image without too much trouble, but still not by accident.

"Metadata easily visible." The metadata should be displayed in a way which supports the user viewing the content. The tag display should be very consistent in all parts of the user interface; they should be in a strict and consistent order. The entire metadata should be a consistent box or a grid, which holds the metadata information.

4.4 Adding Images to the Archive

"Images through a batch submit dialogue." The user should be easily able to upload a collection of images to the archive. The upload process should be similar to many familiar websites that contain user uploaded images, such as Flickr or Facebook.

"Batch edit of tags & metadata." The user should be able to quickly edit the metadata information in batch. The user should not be required to do unnecessary repetition when tagging images. The interface will not assist the user with the information itself, for example by reading the database for existing image metadata and suggesting the user to insert information based on existing images.

"Reads date information from EXIF-tag." Most images taken on a modern digital camera have an EXIF-tag inserted to the images. This EXIF-tag contains various information about the image and the about the camera the image was taken on. [5] In this case the most important information is the date when the image was taken. The new interface should read the tag and automatically write the stored date to the image's metadata on the database.

"Recognizes the user." In the existing system, the user has to tell the interface their username in order to upload images to the archive. The original idea was that the system should recognize the user to skip this step, but as the images are uploaded directly from the user's computer and not from a network drive, this feature is not necessary.

"Easier to choose which images to upload." The archive will not help the user to pick images for uploading; the user has to pick the images by themselves using other software, such as Max OS X's Preview or Windows Photo Viewer.

5 First Design

5.1 Mock-ups

The first design is a mock-up that contains most of the overall functionality. There will be small adjustments to individual features but the overall design should stay more or less the same. Partly missing at the moment are dialogs and feedback the system presents to the user.

In order to evaluate the ideas and construct a base for development I built a mock-up of each page on the user interface, as shown in figures 4 to 8. This allowed further refining of the selected features and it was possible to present the user interface for the users and client in order to gain feedback and have the plan accepted and ready for implementation. The new colour scheme is partly based on the official style for in-house products. Every page also follows the same theme; the previous user interface had variation between almost each page. The layout is also the same for every page with the information or edit bar at top and navigation at the bottom.

5.2 Search View

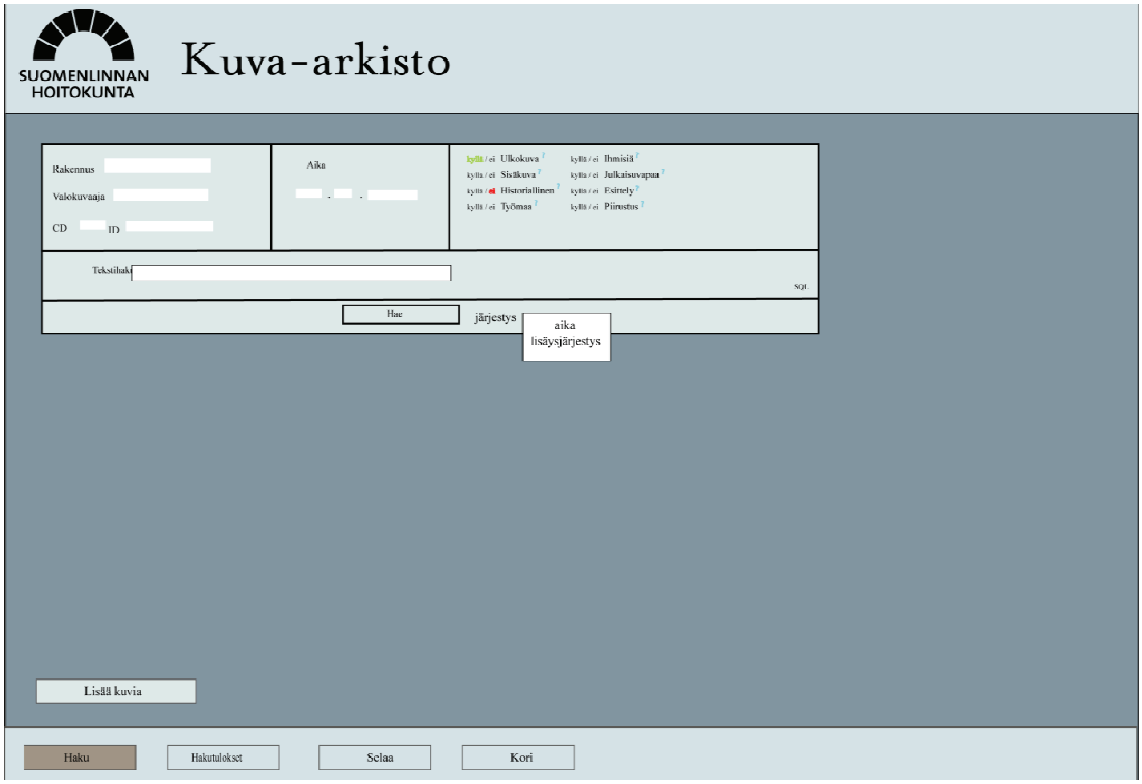

Figure 4. Search view.

The image search view is mostly the same as before. Different from before is how the building address is inserted; before the island code, a letter, and the house number were on different fields, in the new one it is possible to enter “C14” on a single field.

It is also possible to search for images based on tag value ‘false’. Before it was possible to search for images that have certain tags but it was impossible to exclude images with a tag, such as construction. The date of the images also has a different input; it no longer uses dropdown menus, which made sure that users did not insert faulty information, but was problematic to use.

5.3 Search Results

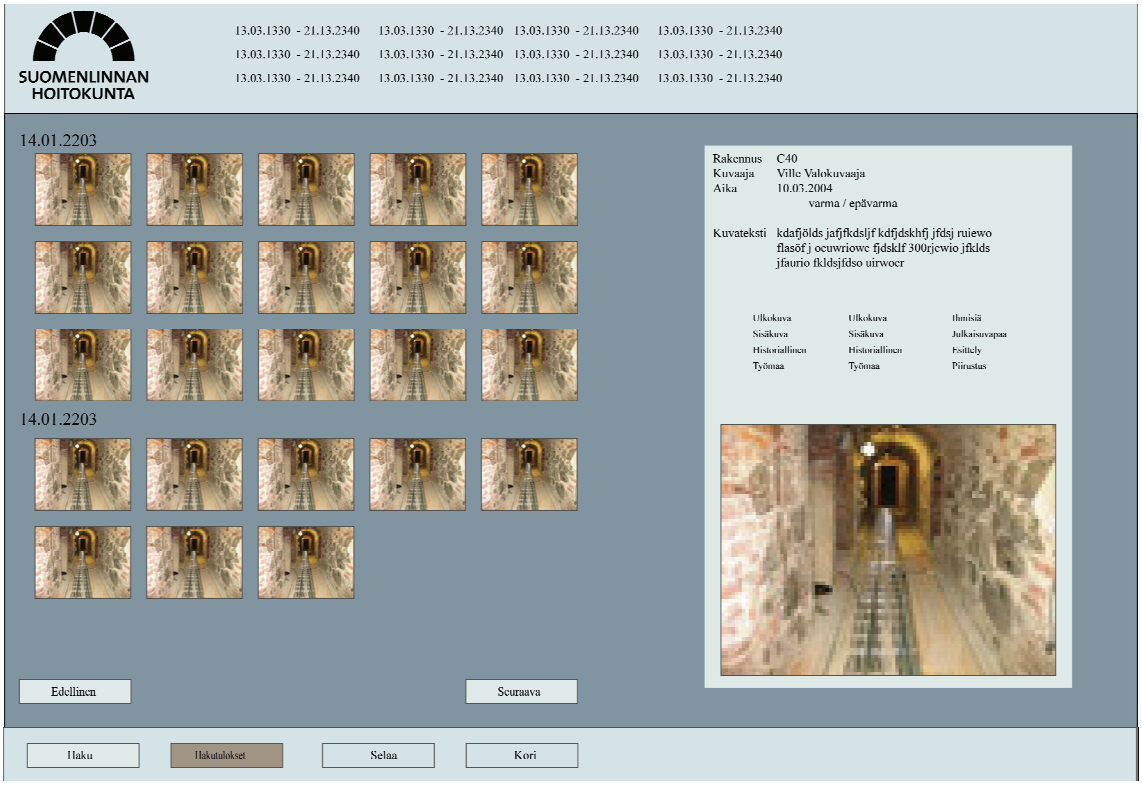

Figure 5. Search results view.

The pages of search results are shown at the top, as illustrated in figure 5. If there are more links to pages than can fit the window, there will be a horizontal scrollbar. Scrolling will not interfere with rest of the layout. The amount of search result thumbnails on the centre is determined based on the browser size; not more are shown than fits the area. However, if the browser window is resized with the search results shown, it will not accommodate for the change as its a much more complex feature.

All the metadata information of the picture is shown when the mouse cursor moves on top of the picture. The information is shown in the box on the right of the page. This box will be almost the same on every view and page. There will be slight variation; there will be a couple of buttons or the picture is missing but the metadata information is always present and in the same arrangement. The user can view a larger picture by clicking on the thumbnail. I originally planned to have users insert pictures to the

basket from here by clicking on them, but later on I realized that pictures that are already in the archive rarely need their metadata to be edited in batch.

5.4 Browsing Images

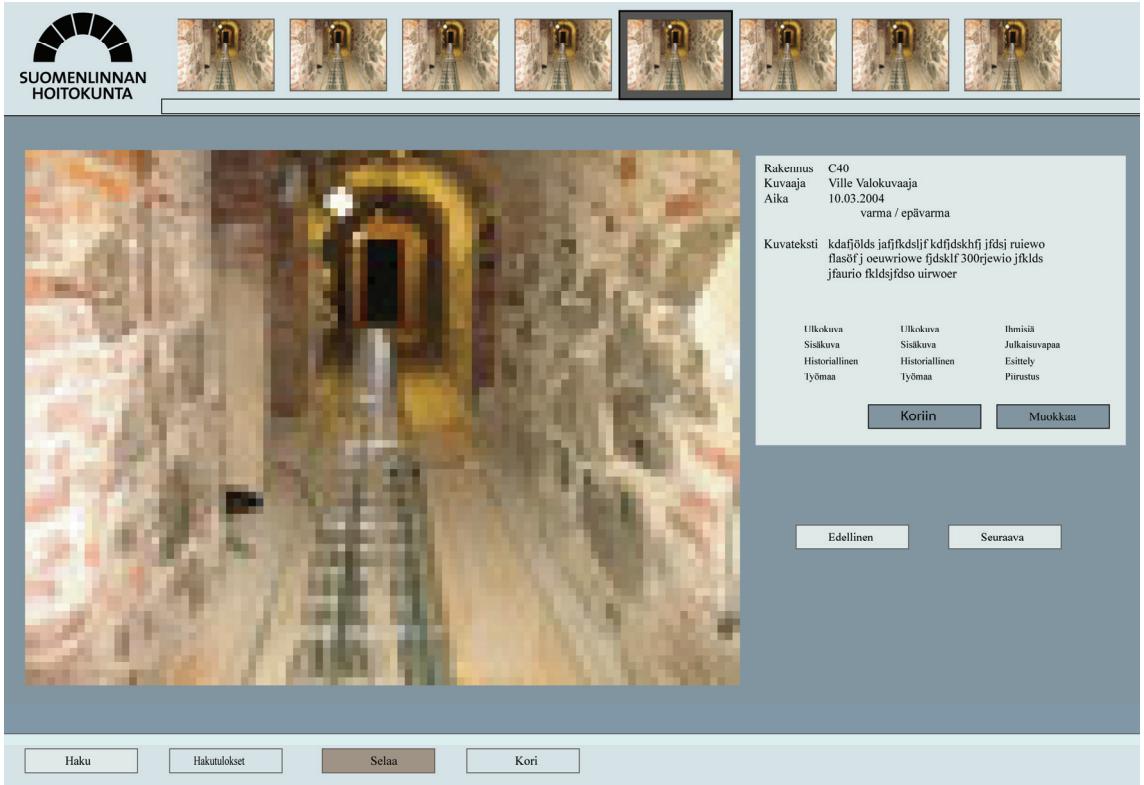

Figure 6. Browse view.

Figure 6 shows the browse view which is used for viewing the images in the archive. The view has next and previous buttons the user can use to switch between images or it is possible to use the arrow keys, much like in many Web 2.0 sites or desktop software. Thumbnails of the images are shown in the top bar, so the user knows which images are already seen and which are the next images in line. It is not possible to fit a thumbnail of every image on the search to the top bar so the bar will need the same image groupings as the search results window.

It is possible to edit the image metadata while staying in this view by clicking on the edit button. The user is allowed input where there is output. However, the user still must ask for permission to edit the metadata, which is not always recommended. [7;231] In this case the edit button does allow an easy software development feature

as the edit button can be changed to save changes button. The user can also add images to the basket from the associated button; the same button is also used to remove them from the basket.

5.5 Basket

This is the most notably different section from the original image archive. This is completely new in relation to the older user interface. The design principle in this view is to reduce the amount of repetition by the user when editing metadata to very minimum.

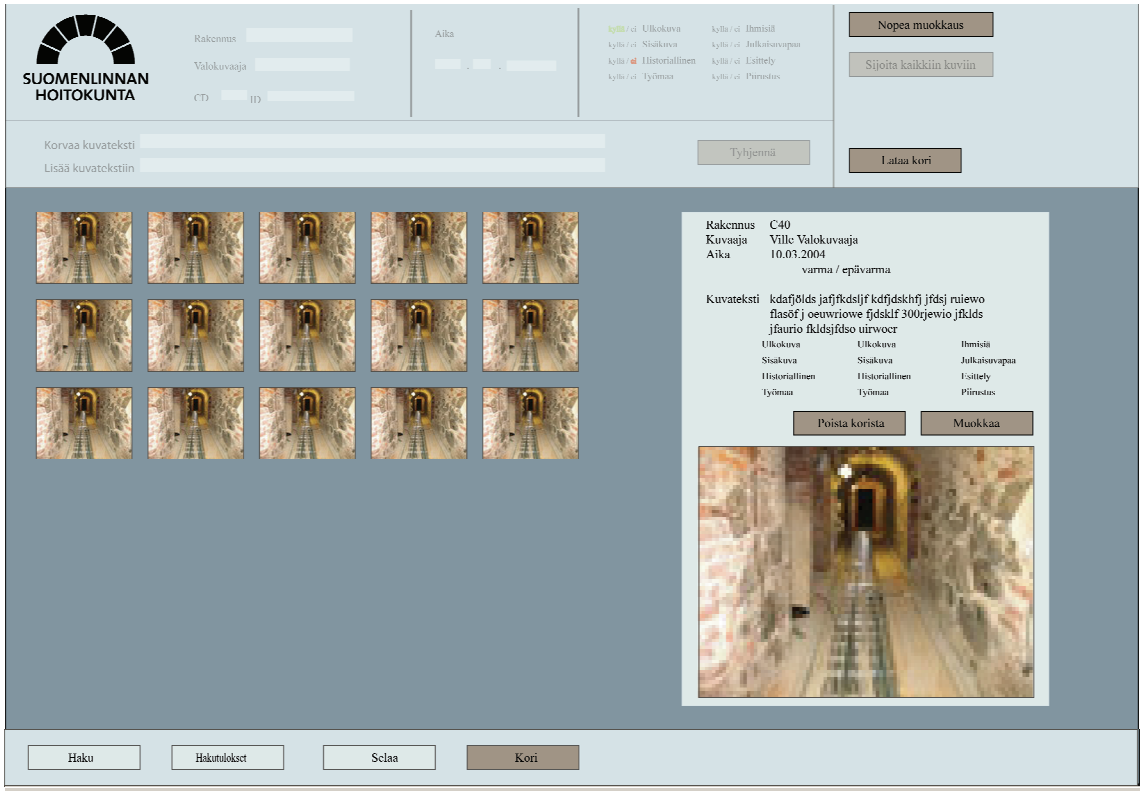

Figure 7. Batch edit view with fast editing off.

The first screen has the batch edit options faded out, as illustrated in figure 7. However, the user can click on the thumbnails, which shows the box on the right with the metadata information. The user can also edit the metadata information from there, or remove the image from the basket. This functionality is a bit different from the functionality in the search results view; here the user must click on an image to see the metadata information where in the search view the metadata information was shown when mouse was over the thumbnail. The difference is caused by the buttons

added to the metadata box, in this view the users must be able to make a persistent selection where as in the search results it is not required.

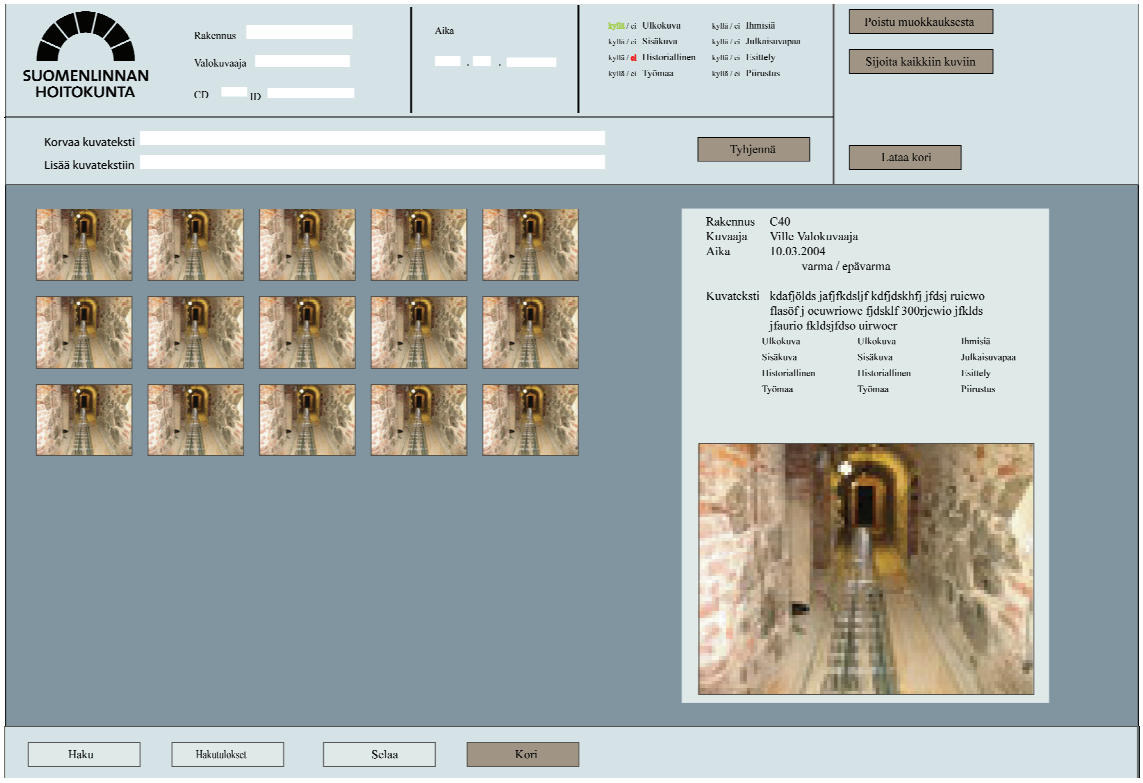

Figure 8. Batch edit with fast editing turned on.

When the user activates the fast edit the metadata information box follows mouseover again. Editing the metadata in this form is based on a single click interface, as shown in figure 8. The new metadata is inserted to one or several of the fields at the top of the page. When the user clicks on a thumbnail at the middle section, the metadata information is written on the image. This allows for very fast editing. For example if many of the images have a same feature, such as an oven, the user can write ", oven" to the insert to text description field and click on every image they see fit. The amount of repetition is reduced as the inserted information only needs to be written once and there is very little navigation required. The same method is used to edit other information such as when the photograph was shot or the content tags.

This is also the view which takes in newly uploaded images. The uploaded images are automatically placed on a cleared basket. By using the insert for all images button the

user can place the same information to all the images in the basket, after which the individual editing takes place.

5.6 Omitted Requested Features

A common request was to have a possibility that a selection of images could be a part of a group. Anyone could view groups, add groups, view the images in the groups and add new images to the group. This would be an example of positional retrieval. The user remembers where he left the picture, and would retrieve them based on that memory. This is similar to how most desktop file systems work. At the moment almost nothing is stored in the image archive this way. The images are retrieved based on associative or attribute-based retrieval. [7] Information is stored about the contents of the images added to the archive and they are retrieved based on that information. Adding a positional retrieval system to the archive could provoke bad user behaviour; filling out the correct metadata information would be ignored or postponed indefinitely since the uploader can retrieve the images based on their position inside the group. A positional retrieval, which is at the moment included in the archive, is based on the order the images are in the database; images are always sorted based on when they were added to the archive and positional retrieval is sometimes useful, but should remain a by-product of the archiving system.

6 Evaluation

A moderately ready version of the archive user interface was tested on the actual users. The methodology was not very formal; the users were given the same tasks to do in the archive, but no measurements were taken. The goal was to see where the users struggle, and which parts they find easy. A total of 6 users tried out the new user interface. The evaluation took roughly 30 minutes per person.

In some cases the evaluation results were straightforward; in the browse view only one person used the ‘next’ and ‘previous’ buttons and those buttons were completely removed.

Most common trouble to users was related to labels of items, such as navigation links. Many of the labels were extended or changed completely in nature, to better reflect the users' mental model of the image archive.

7 Deployed version

7.1 Testing

Individual features were tested and evaluated as they were being developed. Many of the features did not end up as they were on the mock-ups, as the features were found out to be insufficient. Adapted from agile development, there was a constant process of evaluating the design choices and improving on them. I often approached the users with the development versions and asked for feedback. I also set the users to try the development versions to see if the users had the correct mental model of the interface.

7.2 Colour Scheme

The entire colour scheme on the mock-ups was scrapped and a new colour scheme was done completely based on the Intranet web site of the Governing Body of Suomenlinna. The transition from the Intranet site, which also houses the link to the image archive, is very smooth and unobtrusive, as the visual style stays very similar.

The colour scheme itself consists mostly of a couple of shades of grey that were available in the in-house style. The purpose of the archive interface is to allow users access to the images and to help users archive their images. As the purpose is not entertainment or the web page itself, having a bright or interesting colour scheme would most likely only have adverse effect on use.

7.3 Date Values

Majority of the changes during development in respect to the mock-ups were on how to input the date value for images. What is stored on the database supports non-

complete dates; it is possible to save image metadata so that it describes the date only partially. This is a desired functionality. For example, if it is know that an image was taken during year 2000 but the month is not known, or that an image was taken during July 2000 but the exact day is not known, it is possible to store the known information in the database while leaving out the unknown information. This inconsistency in respect to regular date formats limits out the use of many available plug-ins for inputting date values.

7.4 Search View

Displayed in figure 9 is the image search view, which has all the intended functionality. The user can fill out some of the fields to perform a search. Much like in other similar searches, fields left as blank will be excluded from the search criteria. The search is exclusive, the more is filled in, the more exact the search results are.

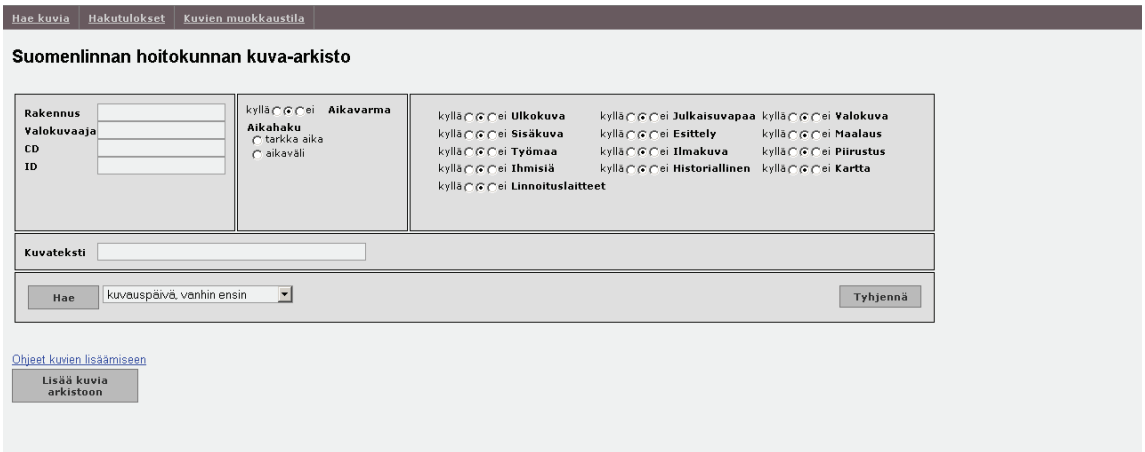

Figure 9. Search view.

Most problematic was searching images based on date the image was taken. There needs to be two sets of fields; one for searching on exact date and one set for searching between two dates. The first design is shown on figure 10.

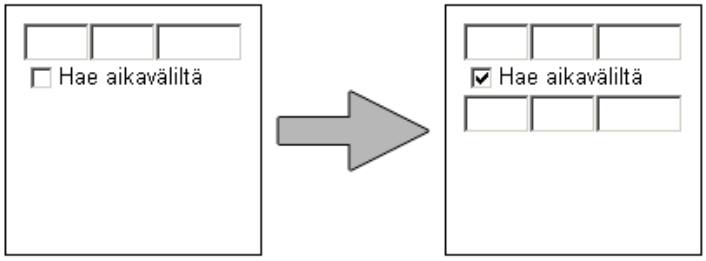

Figure 10. First design of date input to search form.

The problem was that nobody found the checkbox. The users tried to input the two different dates on the 3 fields, which were for day, month and year of the single date value. One of the causes for the problem was probably a missing label or header for the input fields; the users probably interpreted the checkbox label as the topic header instead of a feature with functionality.

Figure 11 displays a revised design that was made to improve on the original design. On the improved design, the user must explicitly choose between searching for images taken on an exact date and searching for images taken between two dates. Before the choice is made, the user is not presented with the date value fields, thus denying the possibility to input anything. There also is an example of how to insert the values. Bundled with the date input value is the tag that describes if the specified photography date is certain or not.

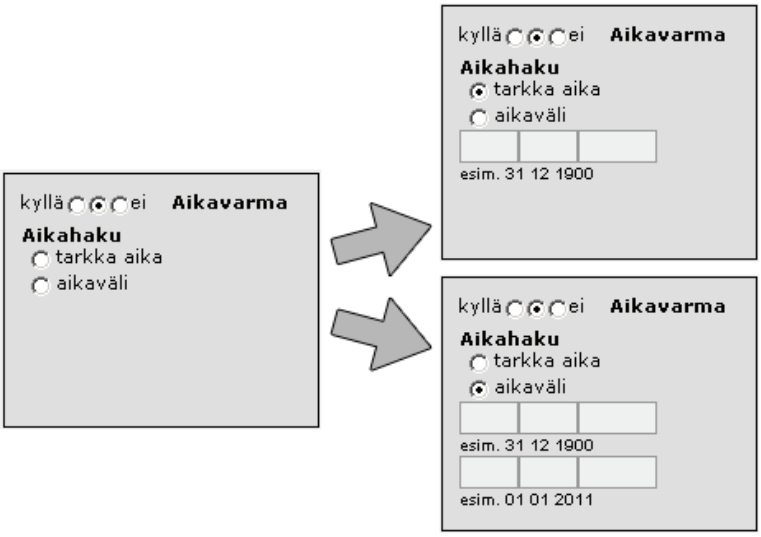

Figure 11. Revised date input for search form.

Searching for "exact date" allows the user a bit of play, not all of the fields need to be filled and the remaining ones are treated as wildcards. This is similar to what was in the existing image archive and is what the existing users expect, despite being somewhat irregular or uncommon functionality. For example, it is possible to fill in only the year field, and all images for that year are returned. It is also possible to search for all images taken in a specified month, or to do a trivial search for all images taken on the 15th regardless of month or year.

Similar searches are possible for searching between to dates. As the controller uses an SQL BETWEEN operator, filling in only the month or day values provides nonsensical results. It is however possible to find images by filling out the year, year and month or the complete date value. For example, by filling in years 1950 and 1959 the user can search for all images dated for the 50's.

7.5 Search Results

Figure 12 shows the search results view. This view has the search results after the user has filled in the form at the search page. At the moment there are 42 search results displayed as thumbnails and a box that displays the metadata information for the thumbnail image that is currently under the mouse cursor.

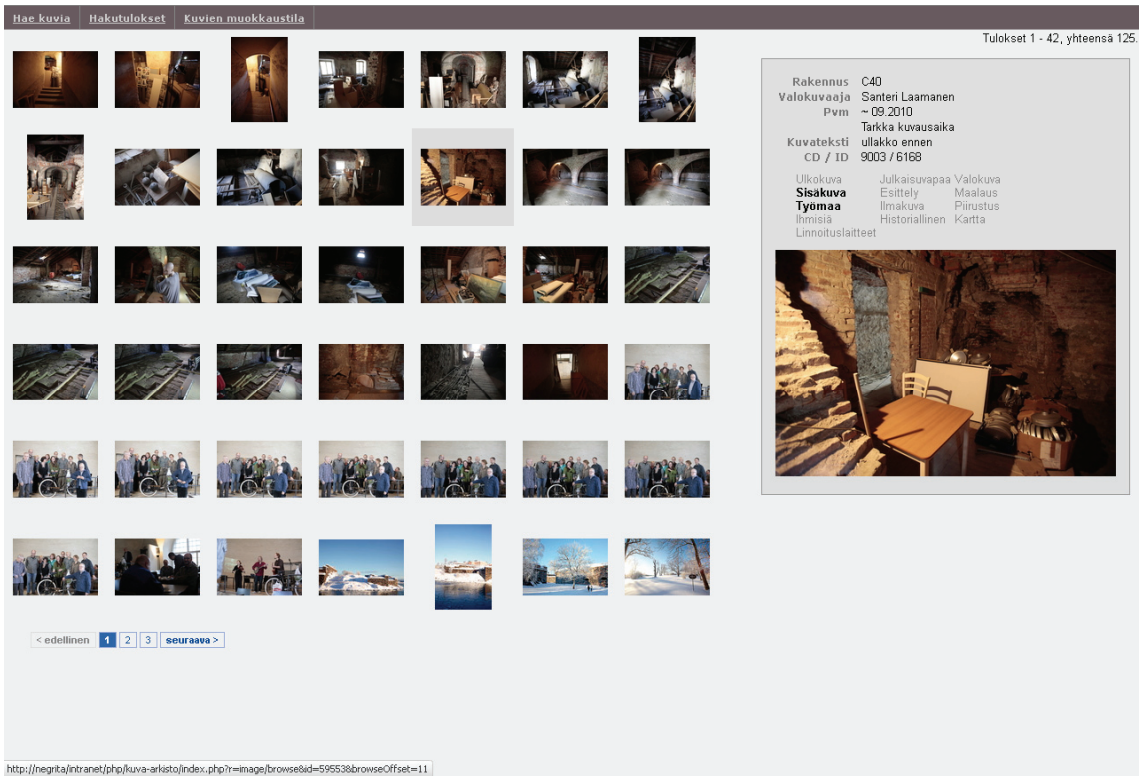

Figure 12. Display of search results.

If there are more results than can fit one page, there is a page navigation tool under the search results thumbnails. During the evaluation it was found out that every user found the navigation tool very quickly and nobody had trouble finding the tool, and it was never left unnoticed. The tool was left exactly where it was during the evaluation.

The search results view was originally intended to be scaled to the screen. This was however left out to reduce the amount of coding needed. The scaling would have been problematic to implement, as it would have had to actively converse with the object that creates the pagination in Yii framework. [4;5]

Figure 13. The metadata display box.

Figure 13 shows the box, which displays the metadata information for whichever thumbnail is currently under the mouse cursor. The user can quickly inspect images simply by hovering over them in the results view. The box displays all the data, which is stored in the database.

All of the tags are visible, even the ones which are set as 'false'. Notice that the metadata box displayed in figure 13 has the 'photograph' tag incorrectly set up as 'false'. The tags are always in the same order; the user does not need to read the actual words as the user can determine which tags are set as 'true' solely based on their location in the grid.

7.6 Browsing Images

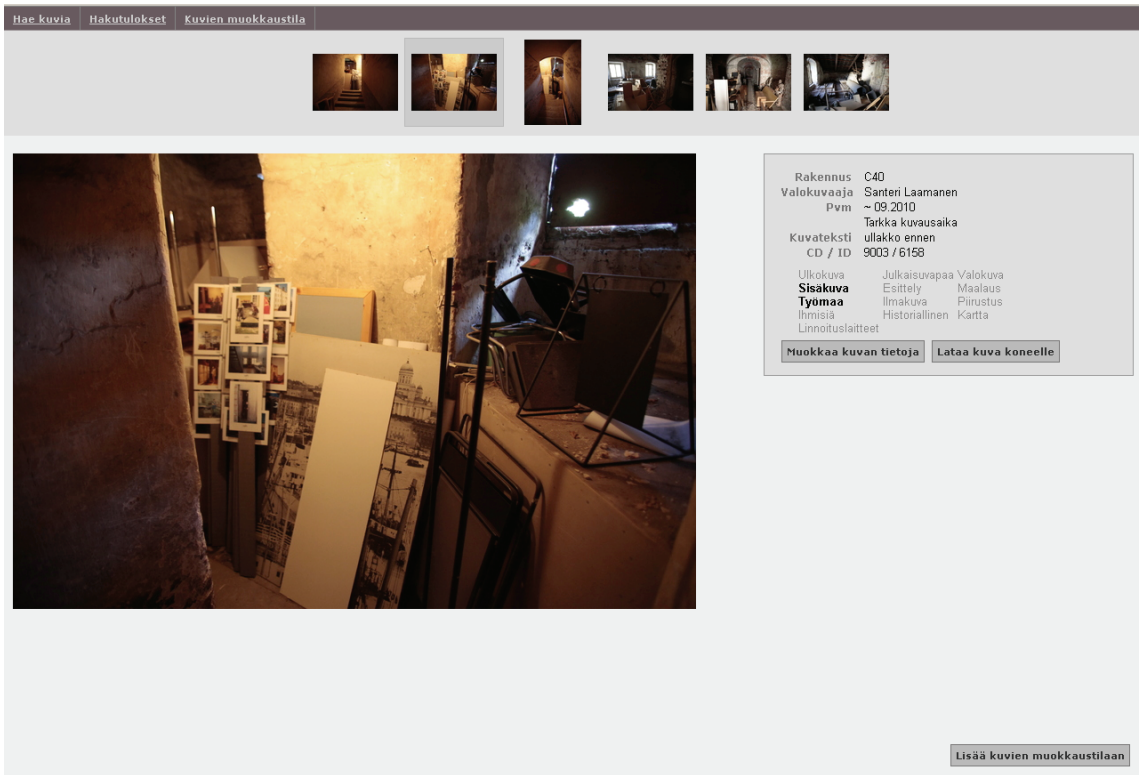

Figure 14. Browsing view for images.

After the user clicks on an image in the results view, the user is directed to the view displayed in figure 14. In this view the user is presented with a reasonably large view of the image and all its metadata information, as stored in the database. The user can click on the image to view the image in full size, which is the original image uploaded to the archive.

Displayed at the top of the browse view is a collection of thumbnails, as shown in figure 14. The roll of images shows four previous images and four next images of the search results. The middle one is always the currently viewed one. If the user is viewing an image which is in position 1-4 of the search results, blank space will be displayed instead of the thumbnails, so the currently viewed picture always stays in the same location. By clicking on the thumbnail the user can jump to the browse view of that image. Jumping consecutively to the next image can be achieved simply by holding the mouse over the next thumbnail and repeatedly pressing the mouse button.

Alternatively, the user can use the arrow keys on their keyboard to navigate to next or previous images.

Clicking the navigation button which redirects the user to the search results page directs the user to the correct page in relation to what is currently viewed in the browse page. For example, if the image which is currently viewed in the browse view is part of the results displayed in the 6th page of the search results, clicking on the navigation button directs user to the 6th page of the search results.

The user can also load the image to the local machine by pressing the button 'Lataa kuva koneelle'. This is a link to a special page which has header information configured so that the user will be presented typically with a 'Save As' dialogue. This dialogue and its functionality are based on the user's operating system and browser. The filename of the attachment file is specifically constructed from the metadata stored in the archive. The filename is generated based on the date when the image is taken, the old compound serial key and 50 first characters of the 'kuvateksti' field. Every filename is automatically unique and they automatically sort by the date taken when sorted in the user's operating system file manager.

7.7 Editing Metadata in Browse View

The user can edit the metadata information for the image on the browse page. Figure 15 shows the transition between the displayed metadata information and the form where the user can add, edit and delete metadata information.

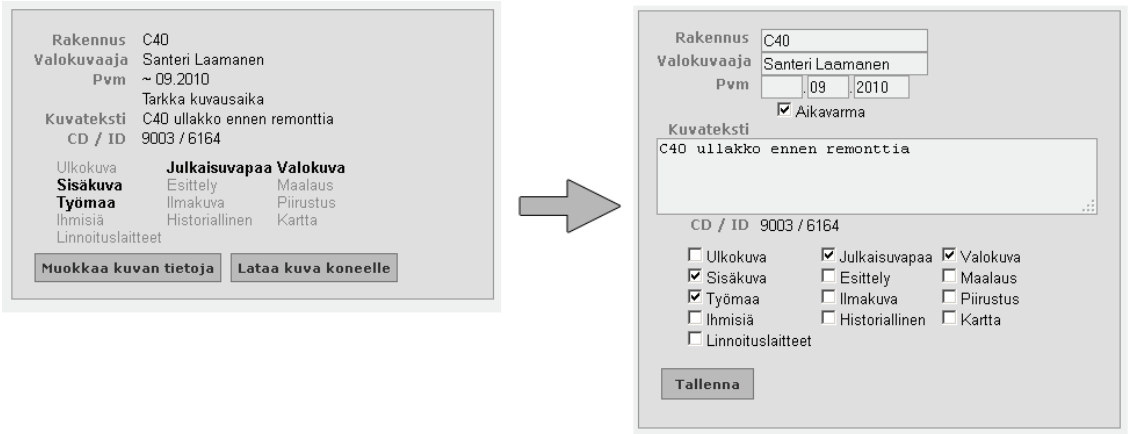

Figure 15. Metadata view and it's associated edit form.

Pressing on the 'Muokkaa kuvan tietoja' button does not redirect the user to a new page, it only affects the box which displays the metadata information. The user is allowed input where there is output. [7]

7.8 Adding Images

The button for uploading images is bundled with the search page. This location is not optimal; an improvement would be to move tools for adding images to the archive to their own page. When the user pressed the 'Lisää kuvia arkistoon' button, the user is presented with a dialogue to submit files. This dialogue is generated by the combination of the users' operating system and choice of web browser and is not something the web application can define. The user can pick one or multiple images and submit them to the page. The upload process then continues as shown in figure 16.

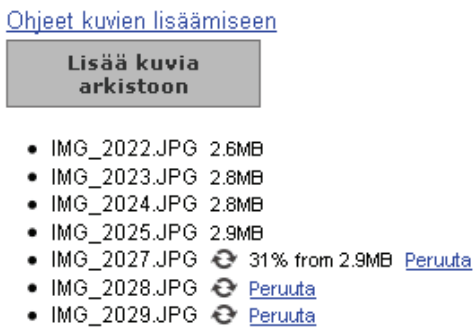

Figure 16. Adding images to the archive.

The images are uploaded and processed by the server one by one. The user can cancel single uploads, should he choose to do so. If the user tries to navigate away from the page, thus cancelling the upload process, the browser will display a warning. The browser will also display an error if the user tries to add files to the archive, which are not images, or are images that cannot be correctly processed by the server.

When the images are uploaded they are added to the batch edit page, specific to the user who added the images to the archive. Very little metadata is stored on the images as they are uploaded. Originally, information of the original uploaded image was saved on the 'kuvateksti' field, but in the evaluation it was discovered that it confused users and the functionality was mostly discarded. The only metadata information automatically stored is the date information from an exif-tag, should there be one.

The JavaScript plug-in and significant portions of the server side PHP script handling the uploaded files are made by Andrew Valums. He has released the script under a free software licence (GNU General Public Licence, version 2) and is used here under the same license. [8]

7.9 Batch Edit

The batch edit system is completely new in respect to the original version of the image archive user interface. The toolbox is at the top of the page, the thumbnail grid is below it on the left and the box that displays metadata information is on the right, as displayed in figure 17. The box which displays the metadata information updates as the user moves their mouse cursor on top of any thumbnail on the thumbnail grid, displaying metadata information and the image in larger size for the thumbnail that is currently under the user's mouse cursor.

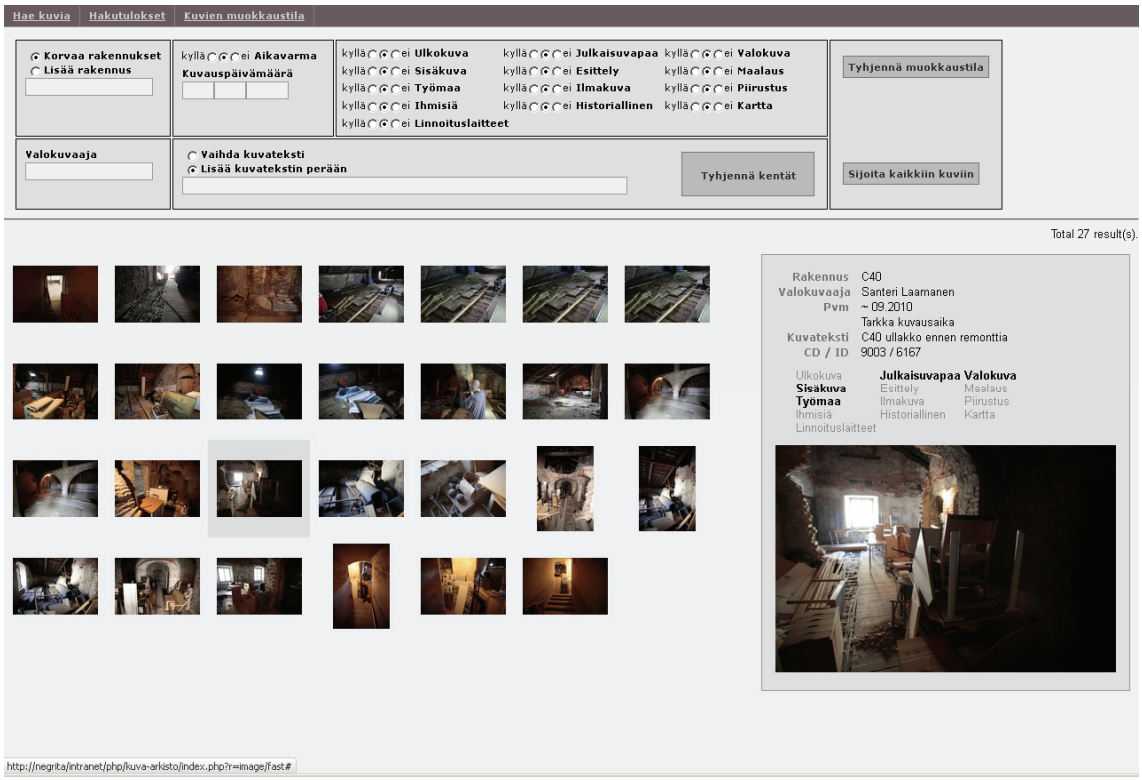

Figure 17. Batch edit page.

In order to write metadata information to the images the user fills in a bit of metadata information to one or many of the fields at the top of the page. This collection of input fields can be considered as a toolbox. When the user clicks in on one of the thumbnails under the toolbox, whatever is written on the toolbox will be saved as that image's metadata. In short, when the user clicks on the thumbnail, the thumbnail's associated metadata will be updated with whatever is inputted to the form at the top. The users know what they are updating based on what is displayed on the box to the right, as the contents of the box are related to the thumbnail which is under the mouse. The users can also write contents of the toolbox to all of the images by using a button in the toolbox itself. This functionality is synonymous to clicking every image thumbnail in the batch edit page manually.

The goal of this tagging system is that it reduces the amount of repetition by the user to absolute minimum. If multiple images have the same subject in them, the user can input the correct information to the toolbox once, and then simply insert the information to the multiple images by clicking all of them. The amount of work is

completely different from opening the images individually and writing the same information down as many times as there are images.

The system is not easy to use the first time; when explaining the system to the users in the evaluation, many had trouble understanding how it works. Eventually, users understood the explanation the first time, but that was most likely due improvements in the explanation itself. The batch edit system was never intended to be intuitive to use, it was intended to be fast after the user learns to control it. This is due to the facts that during the course of a year, the image archive has only a handful of new users who need to upload images. There are also rules on how to tag the images and what to add to the archive, so the users need to be introduced to the archive by an experienced user in any case.

8 Conclusion

The development of the PHP code using the designs was a reasonable success; the end result works quite well and does provide significant improvements over the previous user interface. The search view has a few improvements and the results are displayed in a more structured way. Browsing between images is a lot better as the navigation elements do not move around, and it is much easier to edit single image metadata as it does not require the user to enter another page. The upload process no longer requires the outdated network drive and the batch edit system is fast, once the user has a bit of practice.

Significant portions of the development process can be argued to be a waste of time as I mostly spent time replicating functionality already in the previous user interface. However, I learned a lot about object oriented programming, the model view controller structure and about coding on a framework. Even the end result itself is better as it is done in a more object oriented way and due to the framework, it is more extendable and easier to work with for other developers.

References

1 Suomenlinnan hoitokunta. Suomenlinna sea fortress official website – The Governing Body of Suomenlinna [homepage on the Internet]. Helsinki, Finland: Suomenlinnan hoitokunta; [cited 2011 May 06]. Available from: http://www.suomenlinna.fi/en/the_governing_body_of_suomenlinna/

2 Elmasri R, Navathe SB, editors. Fundamentals of database systems. 4th ed. Boston, MA: Pearson Addison-Wesley; 2004.

3 Rogers Y, Sharp H, Preece J, editors. Interaction design: Beyond human – computer interaction. 2nd ed. West Sussex, England: Wiley; 2002

4 Yii Software LLC. CPagination, class reference, [homepage on the Internet]. Yii Software LLC; [accessed 2011 May 06]. Available from http://www.yiiframework.com/doc/api/1.1/CPagination

5 Japan electronics and information technology industries association. Exchangeable image file format for digital still cameras: Exif Version 2.3 [document on the Internet]. Camera & imaging products association; April 2010 [cited May 2011]. Available from: http://www.cipa.jp/english/hyoujunka/kikaku/pdf/DC-008-2010_E.pdf

6 Winesett J. Agile web application development with Yii 1.1 and PHP5. Birmingham, United Kingdom: Packt Publishing; 2010.

7 Cooper, A. About face 3: The essentials of interaction design. Wiley; 2007.

8 Valums A. Ajax Upload; A file upload script with progress-bar, drag-and-drop [homepage on the internet]. Valums Andrew; [updated 2010 July 20; accessed 2011 May 06]. Available from: http://valums.com/ajax-upload/

User Interface Interview Questions

Questions Related to Searching and Retrieving Images

- What do you search for from the image archive?
- What have you found easily?
- What have you not found even after searching?
- What search criteria do you usually use?
- Have you ever missed something but then found it later?
- What do you find difficult about searching for images?
- Do you use the "glass case" or the other view for search results?
- What functionality would make the searching easier?

Questions Related to Uploading Images

- Have you uploaded images to the archive?
- What about uploading was easy?
- What about uploading was difficult?
- Would it be better if the process were faster?
- Have you ever accidentally lost your images?
- Have you ever accidentally deleted or destroyed images?.

Design mind-map