

Daniel Rozsnyó:

diary.rozsnyo.com

final report

rozsnyo electronics and software
2005

Contents

- Contents 1
- The assignment 2
 - In short 2
 - In details 2
 - Where it came from? 2
- Index file 3
 - Structure 3
 - Document ID 3
 - Name and desc 3
 - Files 3
 - Group 3
 - Author 3
 - File types 4
 - VIEW 4
 - SOURCE 4
 - CODE 4
 - IMAGE 4
- Index collection 4
- Extensive caching 5
 - Authors 5
 - Groups 5
 - Timeline 5
- Search engine 5
- The portal 6
 - Initial page 6
 - How to use 6
 - The time line 7
 - Groups of interest 8
 - Authors 9
 - Search 10
- TODO 11
 - Metadata system 11
 - Browser 11
 - Advanced 11

The assignment

In short

Make a browser for a database of documents.

In details

The number of documents will be rising by the time and to keep a system in them, the files are located in three level directory structure named by date (separate directory for each year, month and day). Additionally, an index file exists for each day which contains some metadata about the document. The metadata involves the name of the document, a short description of it, a list of authors and a set of interest groups. Besides this information about the release, all the files belonging to the document are listed with some more information, e.g. the language and number of pages (in case of text documents).

The user of the browser must be able to locate any document with an approach based on any of the three basic dimensions - time, authors and groups. Moreover, the browser must incorporate a search engine which can be used to find a document based on text search in metadata.

Where it came from?

The idea of the diary portal is located here: <http://diary.rozsnyo.com/2005/01/17/diary/>

Index file

Structure

An example daily index file looks like this:

```
<?xml version="1.0" encoding="Windows-1250"?>
<index>

  <document id="ui">
    <group>MDFS</group>
    <group>VUT Brno</group>
    <name>Zápis uživatelského rozhraní v XML</name>
    <desc>Projekt do předmětu SCS</desc>
    <author>
      <lastname>Rozsnyó</lastname>
      <firstname>Daniel</firstname>
    </author>
    <file>
      <type>VIEW</type>
      <lang>cz</lang>
      <name>Zapis.UI.v.XML.pdf</name>
      <pages>15</pages>
    </file>
    <file>
      <type>SOURCE</type>
      <lang>cz</lang>
      <name>Zapis.UI.v.XML.doc</name>
      <pages>15</pages>
    </file>
    <file>
      <type>CODE</type>
      <lang>en</lang>
      <name>Zapis.UI.v.XML-sources.zip</name>
    </file>
  </document>

</index>
```

The structure of the index is more or less self explanatory. What is not clear from the example data is here:

Document ID

The id attribute at the document has to be unique only for the day (for which the index file is written). The use of it is to form an URL to a single document and it is mandatory.

Name and desc

Also mandatory

Files

Files of a document must be named in a way that they could not interfere with the other files in the specific day.

Group

The group tag can exist multiple times. For indexing, a group normalization is done, which makes lowercase text and space are replaced by “-“, so it is advised to use English words (separated by space) as group names.

Author

The author tag can also exist multiple times. Again for indexing a transformation is done - the white-space between tags is stripped out and a MD5 hash is calculated from that XML fragment. That hash is then used in comparisons and as the primary (unique) key to authors.

File types

VIEW

The file is supposed to be for viewing only and the chosen format could be one from the write once, read many times group (e.g. PDF, rendered video or animation, etc.). In case of text documents it is advised to show the number of pages.

```
<file>
  <type>VIEW</type>
  <lang>cz</lang>
  <name>Zapis.UI.v.XML.pdf</name>
  <pages>15</pages>
</file>
```

SOURCE

The file is in format which is editable, e.g. DOC file, TeX sources, 3D scene source (not rendered!). In case of text documents it is better mention the number of pages.

```
<file>
  <type>SOURCE</type>
  <lang>cz</lang>
  <name>Zapis.UI.v.XML.doc</name>
  <pages>15</pages>
</file>
```

CODE

Source codes, can have an optional language tag which identifies the comments language:

```
<file>
  <type>CODE</type>
  <lang>en</lang>
  <name>Zapis.UI.v.XML-sources.zip</name>
</file>
```

IMAGE

This type requires only a file name and denotes an image

```
<file>
  <type>IMAGE</type>
  <name>session.jpeg</name>
</file>
```

Suggestion - use jpeg, gif, animated gif or png formats (as they are web positive)

Index collection

Every hour (can be adjusted) is run a process which collects all the index documents from the directory structure and merges them into a single XML file. This file has the following structure:

```
<?xml version="1.0" encoding="utf-8" ?>
<diary>
  <year id="2005">
    <month id="1">
      <day id="17">
        <document id="diary">
          :
          :
        </document>
      </day>
    </month>
  </year>
</diary>
```

The index is located in `cache/index.xml`, but is also available for the public as the `diary.xml` file in the root of the web site.

Extensive caching

To speed up the serving of various views, the data is pre-selected in a cache.

Authors

The list of authors is extracted (**cache/authors.xml**) and for each author, a file which contains the documents of the author is created - (**cache/authors/UID.xml**).

Groups

Same principle as authors, but the files are **cache/groups.xml** and **cache/groups/UID.xml**.

Timeline

For each year, month and day are saved the sub-selections. The cache files are:

- **cache/YYYY.xml**
- **cache/YYYY/MM.xml**
- **cache/YYYY/MM/DD.xml**

To calculate previous and next year, month or day a helper in a form of text file is created. It contains a list of valid dates. The process of calculating the prev/next item is loading the list into an array, getting the key to the current item, moving the key by 1 up/down and retrieving the item (files are **cache/yyyy.txt**, **cache/yyyymm.txt** and **cache/yyyymdd.txt**).

Search engine

To accelerate also the searching, a set of index files are created. This index can be then used to fast lookups - we do not need to search through the list of documents - rather we will get the list of documents which match the correct word.

The process of creating of the search index is the following:

- take each document from the list (**index.xml**)
- select all data which are text based
- create a list of words
- add the current document's ID into the sets assigned to the words
- when all documents are processed, the sets by word are saved into text files

The list of words is created from a text data - with this algorithm:

- remove national characters (convert them into US-ASCII)
- remove non-character data (replace by space)
- reduce spaces (helps to find the words)
- reduce multiple word occurrence
- remove all short words (under 2 characters, required for saving in hashed files)

For each word a file exists with the list of documents in which the word can be found. There are a lot of words, so they are hashed in one level structure - directory by the first letter, e.g. the **cache/search/w/web.txt** contains this:

```
2004/12/27/mdfs_web_service
2004/12/21/mdfs_web_service
2004/06/18/waf
2004/03/26/waf_proposal
```

To help find these documents upon the “we” or “eb” query, the list of all words is created into **cache/search/index.txt**. This file is once scanned to search words which contains the query as their substrings.

The final process of making a search consists from these steps:

- enumerate sub-matches (using the word list)
- enumerate documents for a word
- making an intersection to get the implicit AND function between query words

The portal

The web portal requires the Apache web server with these 2 modules:

- mod_rewrite
- mod_php (version 5.x.x) with SimpleXML extension

On the client's side, there's a need for a user agent which supports these technologies:

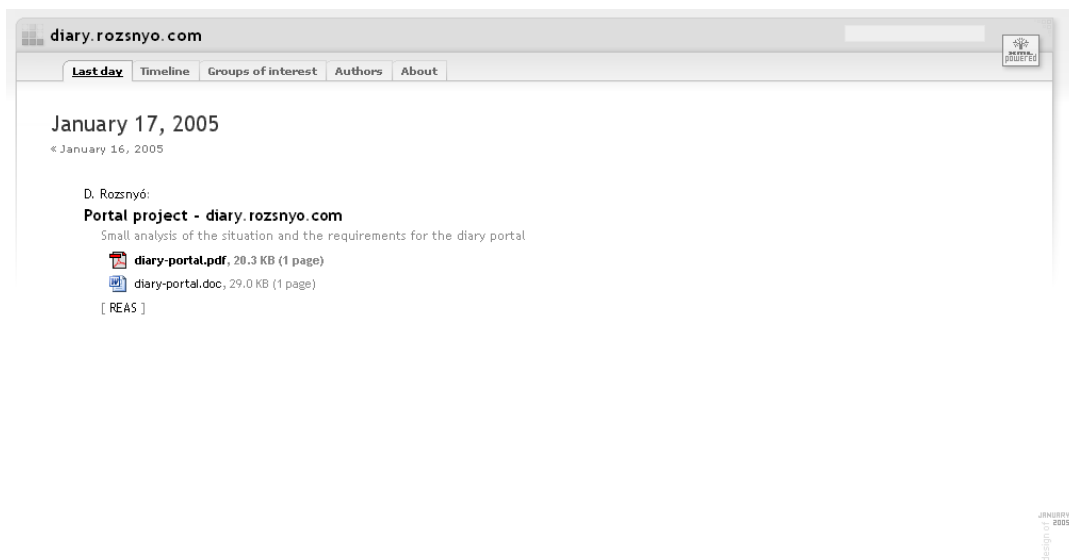
- XHTML 1.0
- CSS 2

Tested browsers (at the time of writing this report) are:

- Microsoft Internet Explorer 6.0
- Mozilla Firefox 1.0 (both MS Windows and Linux builds)

Initial page

Displays a day view with the last day on which a document was released:



How to use

The two basic elements to control the site are:

- the menu (on top left) contains these tabs:
 - Last day
 - Timeline
 - Groups of interest
 - Authors
 - About



- the search box (on top right) has an input field (and no submit button, the search action is then provided upon hitting the ENTER key):



The time line

Offers the list of years, months and days on which at least one document was released:

The screenshot shows the website diary.rozsnyo.com with a navigation bar containing 'Last day', 'Timeline', 'Groups of interest', 'Authors', and 'About'. The 'Timeline' tab is active. Below the navigation bar, the page is titled 'The timeline' with a sub-header 'Focus your selection to a year, month or a specific date'. The main content area displays a list of dates and months for the years 2005 and 2004. For 2005, January is listed with days 6, 13, 15, 16, and 17. For 2004, December is listed with days 9, 15, 19, 20, 21, and 27. Other months listed for 2004 are August (day 7), June (day 18), and March (day 26).

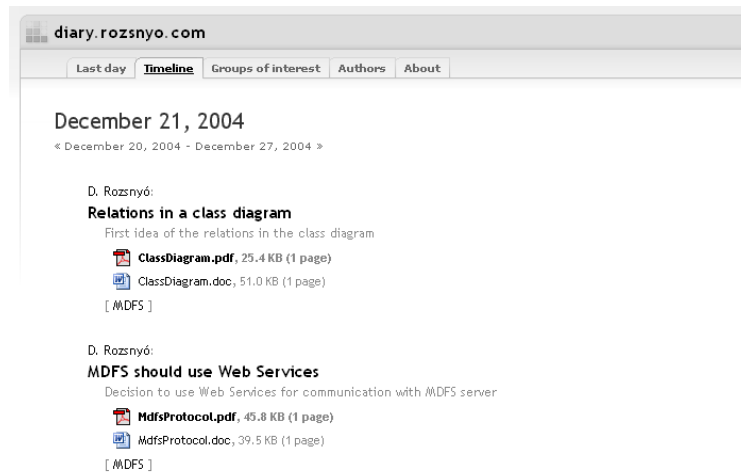
When continuing on a year (not obligatory, the user can freely choose any month or day), a year view will be shown:

The screenshot shows the website diary.rozsnyo.com with the 'Timeline' tab active. The page is titled 'Year 2004' with a link to '2005 >'. The main content area displays a list of dates and months for the year 2004. For December, several documents are listed: 'MDFS Web Service', 'Relations in a class diagram', 'MDFS should use Web Services', 'Struktura systému', 'Implementace algoritmu Parallel Splitting', 'Zápis uživatelského rozhraní v XML', and 'Implementace algoritmu Merge-splitting Sort'. For August, 'EFnet's #Slovakia Session One' is listed. For June, 'Web Application Framework' is listed. For March, 'Web Application Framework - proposal' is listed.

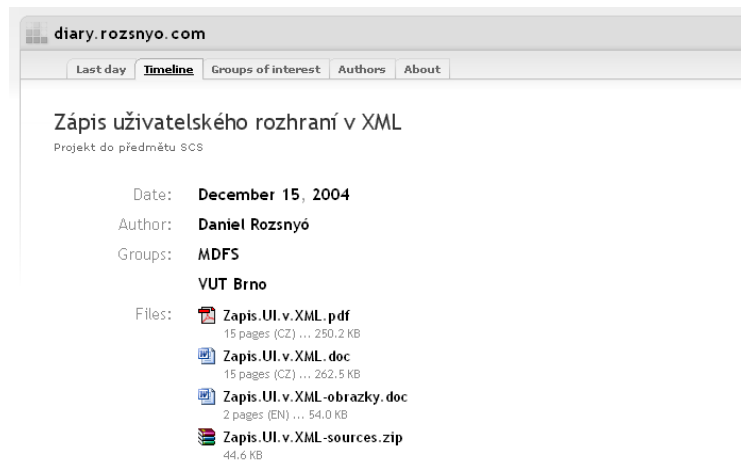
When selecting a month, a detailed view is offered:

The screenshot shows the website diary.rozsnyo.com with the 'Timeline' tab active. The page is titled 'January 2005' with a link to '< December 2004'. The main content area displays a list of dates and documents for the month of January 2005. On January 17, 'Portal project - diary.rozsnyo.com' is listed. On January 16, 'External display' and 'SW RAID' are listed. On January 15, 'rozsnyo electronics and software, spol s r.o.' is listed. On January 13, 'Úvod do zápisu algoritmů v XML' is listed. On January 06, 'Animace ohňostroje' is listed.

And on a day level, the level of details is again higher:

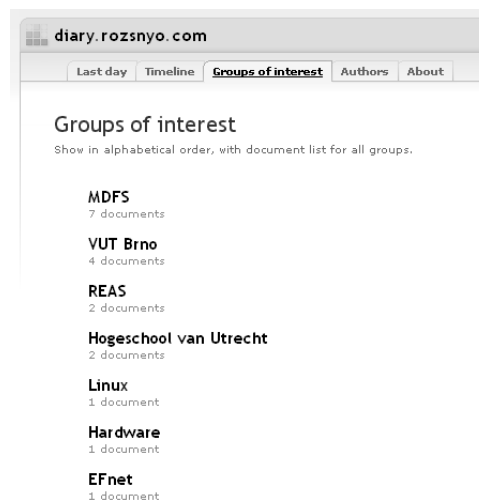


Selecting a document gives the highest level of details:



Groups of interest

This page gives a list of used groups:



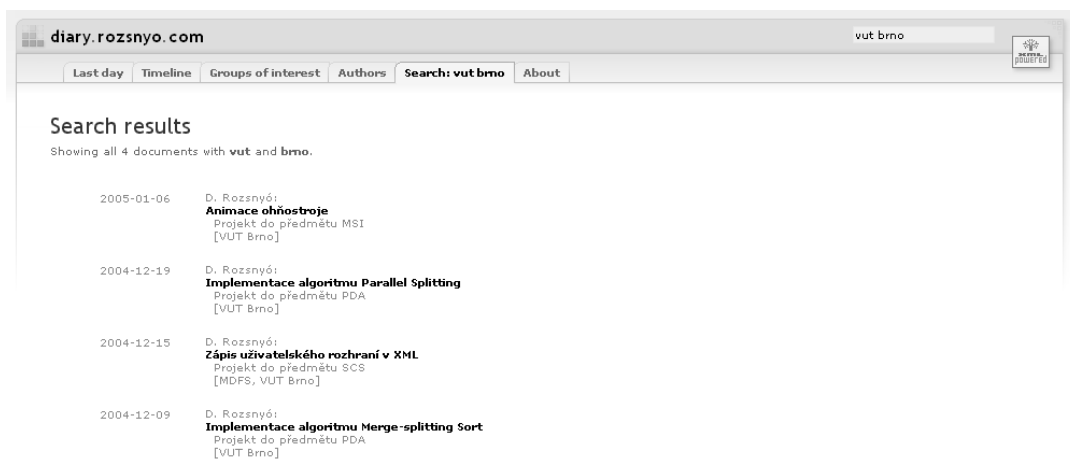
This list can be ordered by decreasing number of documents (default) or alphabetically (A..Z) and when the user selects a group, the list of documents is displayed.

All the listings of documents are ordered by date (newest documents on the top). In the authors' mode, only the additional authors are shown in the document list. The same is true for groups - in the groups view, only the additional groups are shown.

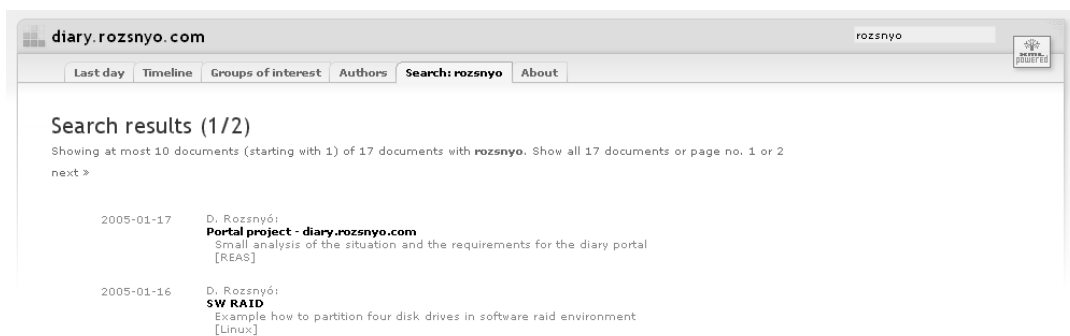
The dates (year, month, day), the authors and groups in the listing are presented as hyperlinks, so here's the cross-reference of the site. The document title contains also the description which is shown in the standard browsers as a tool-tip - when the mouse cursor stays above the name for a certain time.

Search

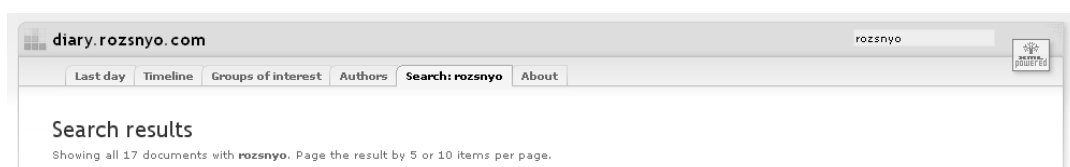
The search function is intuitive - just write a word or two (or more) into the search box and hit the ENTER and the documents, which contain all the words in the metadata, are shown (in order by date, from the newest to the oldest one):



When too much documents are found, they are paged:



However, the user may choose to show all the documents, then the system gives an option to page the result again:



The search page keeps the content of the input box unchanged, but processes the input with the same algorithm as the one described for making a search index (us-ascii, min. 2 letter words, no special characters). The actual search words are shown in the subtitle of the page. Note that the engine searches also in the filenames (and the extensions, e.g. try search "zip") but these texts are not shown in the result view so it can be a bit ambiguous.

TODO

This is a list of unimplemented features which are not critical (at least not for the launch of our diary portal).

Metadata system

- version
 - major / minor number
 - class (draft, final, ...)
- for text based documents
 - page format
 - preview images
 - language information (for combined - ratio in %)
- for images
 - resolution, etc... (EXIF data?)
 - preview

Browser

- sequences
 - identifier of a sequence
 - order inside the sequence

Advanced

- alternative versions, on the fly conversion
 - image format, quality, resolution
 - text as doc, pdf, html
- offline data disk generator