

# THE ASTHMA BLOG

## Web Application Development Module II: Web Programming



The screenshot displays the 'ASTHMA BLOG' web application interface. At the top, there is a navigation menu with links for 'List Posts', 'New Post', 'New Medicine', 'New Comment', 'Log in', and 'Log out'. Below the menu, the main content area is titled 'Enter new asthma blog post'. It features a dropdown menu for 'User' (set to '1'), a 'Title' text input field, and a larger 'Body' text area. Below these are five sets of radio button scales for 'Cough', 'Wheeze', 'Sleep', and 'Activity', each with options 0, 1, 2, 3, 4, and 5. There are also 'Peakflow' and 'Weight' text input fields. A 'Submit' button is located at the bottom of the form. A footer at the very bottom of the page reads '© Per Thykjaer Jensen, 2011 - all rights reserved.'

**By Per Thykjaer Jensen**  
Student nr.: 19810234

**Lecturer: Matthias Korn**  
**Instructor: Peter Urbak**

Url to the prototype: <http://multimusen.dk/cake/>  
(Usage: add new user, log on and try it out)

Saturday, June 11, 2011  
Characters: 22318 (~ 9 pages)

*Copyright: Per Thykjær Jensen 2011*  
*May be used for teaching.*



<http://creativecommons.org/licenses/by-nc-nd/2.5/dk/>

# The Asthma Blog Contents

Abstract.....	2
The Asthma Diary – and the problem to be solved.....	3
Stakeholders - the System in Use.....	3
Persona I: Gabi – the patient point of view.....	4
Persona II: Dr. Lieberkind – the professional point of view.....	4
Persona III: Dr. Hansen – the research point of view.....	5
The system in use.....	5
System Architecture and Datamodel.....	5
Sketches I – the Most Important Pages.....	6
Sketches II Functional Prototype in CakePHP.....	6
The CakePHP Architecture.....	7
Sketch CakePHP Asthma Diary.....	10
The Helper Hack.....	10
Conclusion.....	12
References.....	13
Printed.....	13
Web.....	13

## Abstract

*The asthma diary is an important tool for any asthma patient. Today most asthma patients have to enter observation data with pen and pencil. Could an online blog do this in a better way? In this project I suggest that an online asthma diary could be a better solution. As a demonstration I develop a prototype by CakePHP.*

## The Asthma Diary – and the problem to be solved

Some weeks ago my physician asked me to write an asthma diary. He gave me a nicely printed version, and then I began making observations on a daily basis. Asthma is a tricky disease. Often the patient does not realize that the lung capacities are slowly declining. In fact that was what happened to me. So the doctor gave me some new medicine, and wanted some data in order to know whether the new treatment had any effect or not.

The peakflow indicates how the patient is, and they are interpreted like this:

- **Green:** the best peakflow value = 100%.
- **Yellow:** < 80% && > 50% of the green value.
- **Red:** <50% of the green value.<sup>1</sup>

Therefore the more objective measurements by a peakflow meter<sup>2</sup> are used. On the sheet you also write notes on cough, wheeze, sleep and so on. In my case the form had values from 1-5.

Now you might think, that this kind of asthma diary – or blog – must have been made before. However, that's not the entire picture. Of course you can *wget* Asthma Diaries from the WWW, but mostly these diaries are just plain PDF-files (or worse: docx).<sup>3</sup> So you are supposed to print the asthma diary, and update it by hand.

- *That's a strange way to use a computer...*

During the same period I participated in the Master IT classes on Webprogramming. In the previous session we learned about the classic hypermedia systems – and how they evolved into the World Wide Web – and eventually to the Web 2.0 paradigm, where the social media play a major role.

- *So why don't we do it on the Web?*<sup>4</sup> I thought, because:

- Research estimates some 300 million asthma patients worldwide. Around 80 million patients live in Europe.<sup>5</sup>
- If the values from the patient's observations are stored in a MySQL database you can find the green, yellow and red values by relatively simple SQL-queries and PHP calculations.
- RSS could be usefull for patients – and for professionals.
- Observations could be rendered as nice graphs, bars or similar.

## Stakeholders - the System in Use

In order to establish a data model it is imperative, that you know how the product is supposed to function.

There are 80 million asthma patients in Europe alone. Therefore it is not easy to draw a profile based on age, geography, social status or similar.<sup>6</sup> The obvious path then is to look at form and

---

1 See <http://www.lungusa.org/lung-disease/asthma/living-with-asthma/take-control-of-your-asthma/measuring-your-peak-flow-rate.html>

2 I use the mini-Wright Peakflow Meter. Clement Clarke International LTD: 15 Wigmore Street, London W1H 9LA, England. It's > 20 years old but still works all right.

3 Here is a typical sample: [www.mymainecare.com/documents/Asthma%20Diary.pdf](http://www.mymainecare.com/documents/Asthma%20Diary.pdf)

4 Intertextual reference to the Beatles: "Why don't we do it in the Road?" (White Album, 1968).

5 <http://www.efanet.org/allergy/index.html>

6 On the other hand: if they can afford the expensive medicine they are not among the poorest Europeans. The asthma blog user would probably be middle class and must have access to the internet on a daily basis.

function. What should the website be able to do? How could the results of the observations be presented for the user? Are asthma patients the only stakeholders?

In fact there are more stakeholders involved here than the asthma patient: doctors, scientists and the medical industry. Therefore I will present three scenarios: a patient, a doctor and a scientist.

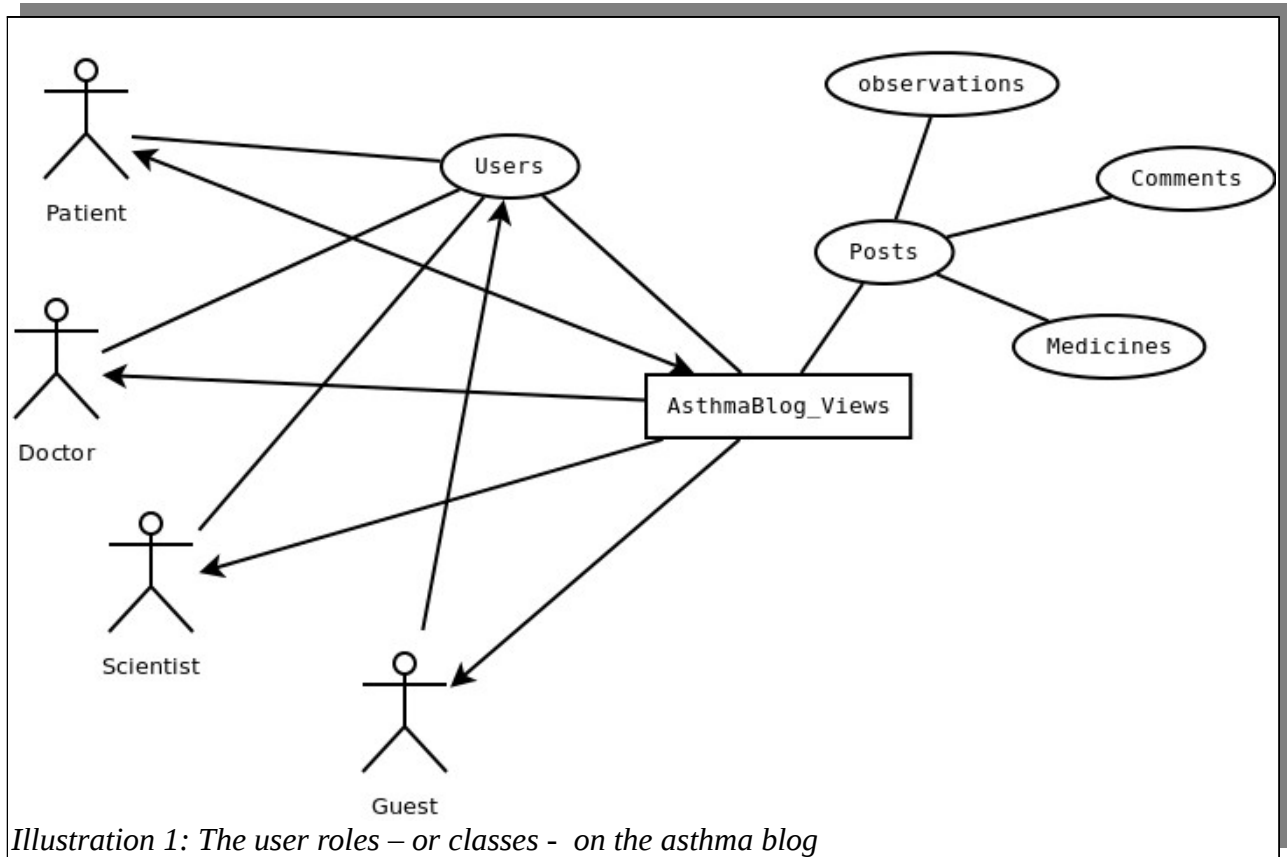


Illustration 1: The user roles – or classes - on the asthma blog

### **Persona I: Gabi – the patient point of view**

In order to get a picture of a typical user, let's imagine *Gabi* 40 year old "lower-upper-middleclass" woman living in Berlin. She is a professor at *Berliner Technische Kunstschule*. She has been an asthma patient since childhood. Sometimes she can live fine without an asthma diary – and sometimes she can't. So it would be nice for her to have a place where her personal data are saved. She's on the WWW on a daily basis. She's proud of her iPad and she's got a laptop too.

So Gabi goes to our website. Blows her peakflow meter. And makes some notes and observations. Perhaps the online asthma diary will tell her, that her peakflow is in the green zone – or if it's in the yellow or red zone the website could suggest, that she should consider contacting her doctor. Because previous data are stored online, she can get a result immediately.

Gabi can print out the result. It should be nicely formatted for print. She could also download an XML file formatted as a spreadsheet.<sup>7</sup> And then she could visit her doctor well prepared.

### **Persona II: Dr. Lieberkind – the professional point of view**

... and the next patient enters. It's Gabi, and she's a typical asthma patient Dr. Lieberkind has known her for years. She brings a printed version of her asthma diary. The doctor looks at the printed

<sup>7</sup> [http://msdn.microsoft.com/en-us/library/aa140062%28v=office.10%29.aspx#odc\\_xlsmlinss\\_hierarchy](http://msdn.microsoft.com/en-us/library/aa140062%28v=office.10%29.aspx#odc_xlsmlinss_hierarchy)

pages, and suggests that Gabi should use more Spirocort – and return in a month with new observations. Of course he could follow her data via RSS and send recommendations via the blog. But dr. Lieberkind prefers the face to face meetings. But he did make a quick note of the interesting URL though... perhaps it could be usefull for other patients?

### ***Persona III: Dr. Hansen – the research point of view***

Novoair is a new medicine for asthma patients. In Aarhus Dr. Hansen follows only the patients using Novoair – and relevant data filtered from the database at the asthma blog.

Normally it is a cumbersome process to find patients for medical tests. The patients must write diaries by hand. The diaries are scanned or typed by secretaries. And then the data is analyzed. But the online blog gives new possibilities. You can do much of this on the fly. Today most patients use the WWW. The data is available immediately – and dr. Hansen can follow the procedings via RSS or a database query.

### ***The system in use***

The data of the asthma blog is interesting to at least three stakeholders. On the website they should play different roles. In practice they should participate in groups – or classes. The usergroups are:

- **Patients**  
The patient should be able to write posts, enter health observations like peakflow etc. From an ethical point of view the patient should be able to give permission to the doctors and scientists to get access to data and whether the posts should be public or not. And the user should be able to export a PDF-file and print the report in order to bring it to consultations with the doctor.
- **Doctors**  
If the user gives permission the doctor could comment the posts and suggest optimal treatment. Here the system is like a blog, but with restrictions. The doctor should also be able to print a report, and save the report as a XML-file formatted for spreadsheetprograms such as Excell and Open Office Calc. Via RSS the doctor should be able to follow the most significant news from the asthma blog.
- **Scientists**  
A scientist could use data from the online blogs in various projects – if the patient permits. Here XML-files would also come in handy. The scientist might want to port the data to statistical programs – and to collate data from segments among the users.

Other roles could be mentioned. A website does not run by itself, so there also should be a log in for administrators. But for the sake of clarity I focus on the primary users.

The scenarios show, that there are more than one way to use the data from the asthma blog. The users could act in different roles such as: patient, therapist or research scientist. As a consequence there should be several access levels available.

## **System Architecture and Datamodel**

The data model emerges from the user scenarios. The following model is a bit simplified, but it was be used as a sketch for the actual database of the prototype. There are many ways to draw a data model. This model is inspired by ER-diagrams<sup>8</sup> with elements from the mindmaps used in

---

<sup>8</sup> So it's a transitional ER-diagram – not strict ;-)

innovation.

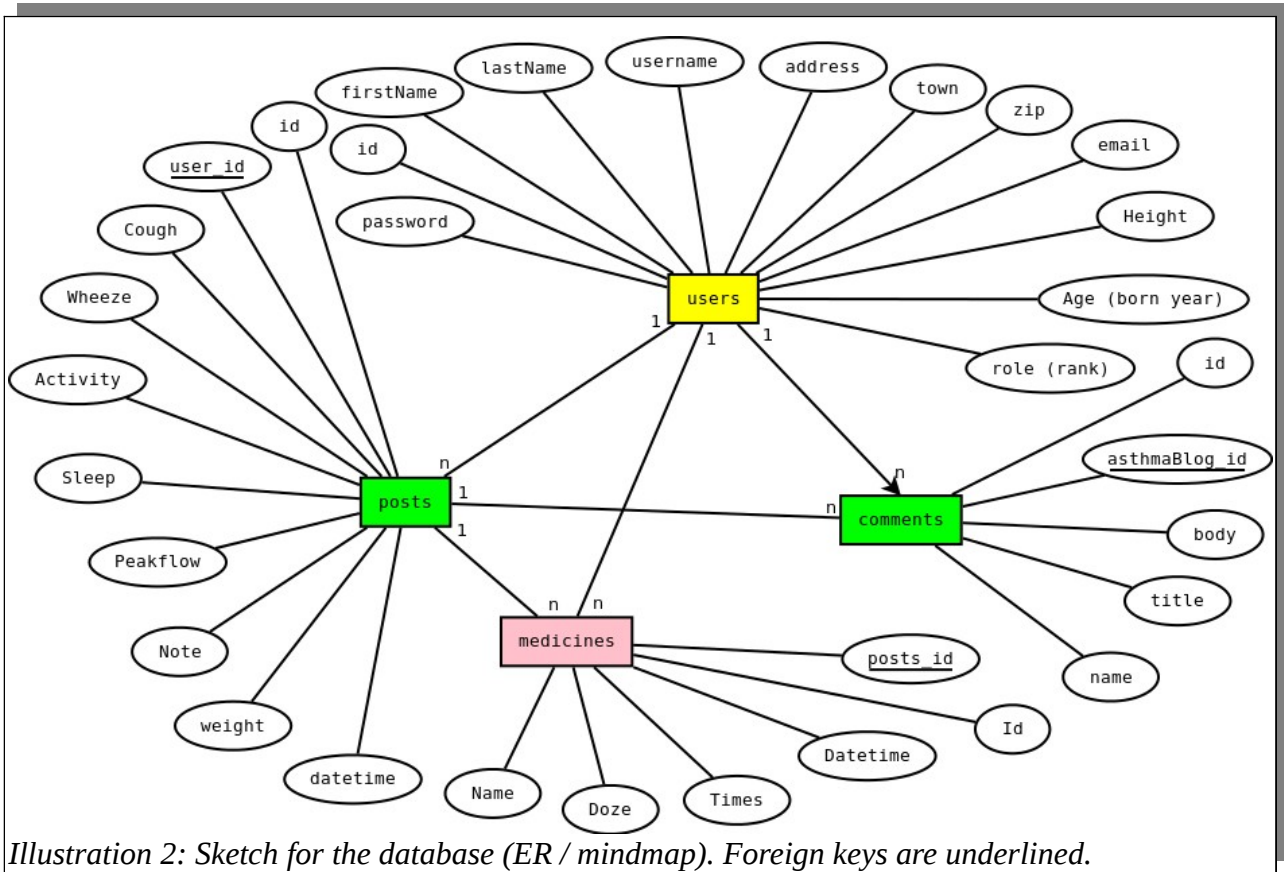


Illustration 2: Sketch for the database (ER / mindmap). Foreign keys are underlined.

## Sketches I – the Most Important Pages

The asthma blog could be called a web service. Ideally the hierarchy should be flat and the user should be able to access all options from a top navigation bar. Joel Sklar would call such a navigation structure a "web structure".

*"Many smaller websites follow the Web-type content (...) which is nonlinear, allowing the user to jump to any page from any other page."*<sup>9</sup>

The Web structure is for smaller websites – if the hierarchy of the website becomes more complicated a new navigation method should be found. On a daily basis using the asthma blog should be as simple as possible. In the Asthma Blog the user can access all pages from a top navigation bar (see Illustration 4). I have to admit, that this is a very conservative solution. But a web service is a tool – not at work of art!<sup>10</sup> Here I agree with Web guru Jacob Nielsen's *Ten Usability Heuristics*:<sup>11</sup> "Follow platform conventions."

## Sketches II Functional Prototype in CakePHP

In order to demonstrate the asthma blog in real life I have made a functional prototype in CakePHP.

9 Sklar 2009: 77.

10 The comparison with art isn't completely fair. In general an artist works with proportion, geometry, gestalt laws and the rules of color. Great art needs no explanations – you understand it intuitively. There are similarities between the methods of a HCI designer and an artist.

11 [http://www.useit.com/papers/heuristic/heuristic\\_list.html](http://www.useit.com/papers/heuristic/heuristic_list.html)

In fact you could prototype by CakePHP until the system is fully developed. Here lies the real strength of the rapid development framework – when you are satisfied after the prototyping iterations, the system is ready to launch on the WWW! The prototype has been developed in three steps:

- First a design shell was developed in HTML.
- The design shell was used with CakePHP 1.3 on my LAN server (a LAMP 2.6.35-25-generic #44-Ubuntu server).<sup>12</sup> Initially all views were baked by “cake bake all”.
- Finally the prototype was ported to <http://www.multimusen.dk/cake> (hosted by One.com).

## The CakePHP Architecture

The Asthma Blog is prototyped via CakePHP. When prototyping with CakePHP the data architecture is given. CakePHP is a PHP framework for “fast web development” inspired by the popular framework *Ruby on Rails*.<sup>13</sup> CakePHP uses the MVC architecture:<sup>14</sup>

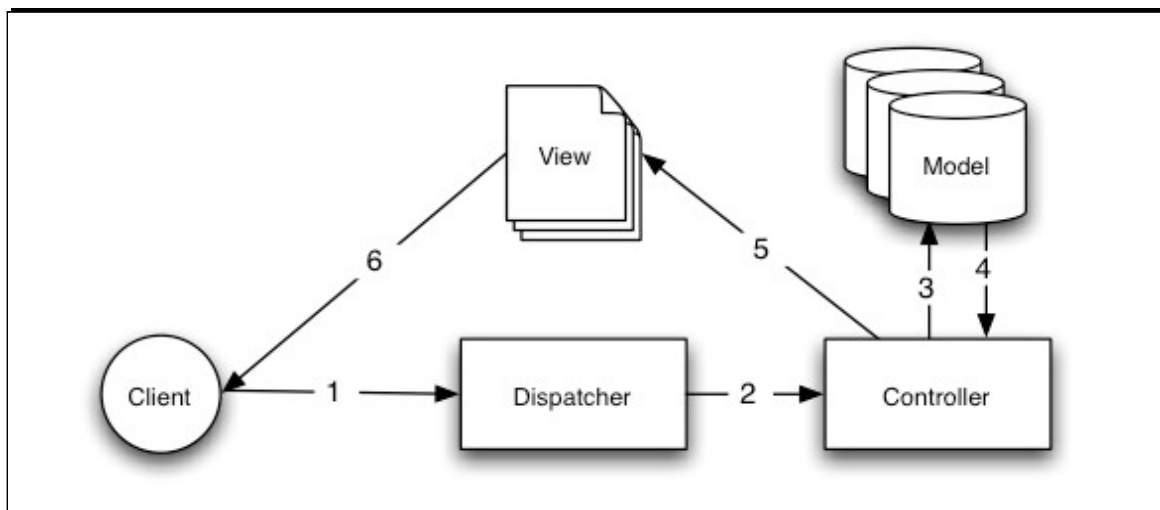


Illustration 3: CakePHP's data architecture “Model – View – Controller” from “The Cookbook” Chapter 1.3.

According to Porebski MVC is an architecture used by frameworks like CakePHP, Symfony, Zend etc. : “Web frameworks take advantage of most, if not all, design patterns. However, MVC is the absolute structural backbone of all frameworks.” [Porebski et al 2011:14] MVC was used before the WWW<sup>15</sup> was invented:

“MVC is an old design pattern, dating back to the 1979 work 'Applications Programming in Smalltalk-80: How to use Model-View-Controller' by Trygve Reenskaug. Since that time it was often used in non-web applications, mostly graphical user interfaces in compiled languages like C++ or Java. There it was easy and natural to implement and exact MVC-pattern, but for web applications, it was somewhat modified.” [Porebski et al 2011:15]<sup>16</sup>

<sup>12</sup> In fact I used two LAMP servers. One was a desktop PC, the other a Laptop. The developmentfiles were placed in Dropbox – and the servers used the CakePHP-files by symbolic links to the Dropbox folder in /var/www.

<sup>13</sup> <http://rubyonrails.org/>

<sup>14</sup> <http://book.cakephp.org/view/1528/Blog#!/view/890/Understanding-Model-View-Controller>

<sup>15</sup> Sir Tim Berners Lee introduced the WWW around 1989.

<sup>16</sup> Porebski et al would prefer the terms “Model-View-Presenter”. However, in the CakePHP documentation MVC is used.

In practise a MVC architecture works like this:

- The user enters a webpage. Her browser presents a View.
- The Controller will fetch relevant information and defines functions.
- The Model is usually a model of the database (if a database is used).

In CakePHP the Views are found in `../cake/app/views`. The default layout is found in the file `../cake/app/views/layout/default.ctp`. The file is a HTML file with PHP-snibbets fetching content or giving paths to CSS, images and other external files. CSS, images and scripts are placed in `../app/webroot/`. The CSS `../cake/app/webroot/css/css.css` is imported somewhat like this with a HTML-helper:

```
<?php echo $this->Html->css('css'); ?>
```

This PHP-snibbet produces this HTML:

```
<link rel="stylesheet" type="text/css" href="/cake/css/css.css" />
```

CSS define the layout of the page. Ideally there should be a different stylesheet for each platform and browser where the Asthma Blog is used. Sadly Safari on an iPad doesn't produce the exactly same view as an Internet Explorer, Chromium<sup>17</sup>, Firefox or even Safari on an iPhone browser. Screen resolution vary heavily from desktop computers to mobile devices. PHP can detect browsers and operating systems. Therefore you could develop a specialized CSS for a number of browsers and platforms.

Most of the `../cake/app/views/layout/default.ctp` is plain HTML PHP will fetch contents and help-text where using these PHP lines:

```
<?php echo $session->flash(); // help text ?>
<?php echo $content_for_layout; // content ?>
```

In this way the layout shell is made. `$content_for_layout` will fetch the code for the actual view. If a user wants to add a new post `$content_for_layout` will fetch the file `../app/views/posts/add.ctp` containing code along these lines:

```
<?php echo $this->Form->create('Post');?>
<fieldset>
<legend><?php __('Enter new asthma blog post'); ?></legend>
<?php
$options = array('0'=>'0', '1'=>'1', '2'=>'2', '3'=>'3', '4'=>'4', '5'=>'5');
$attributes = array('legend'=>"Observations");
    echo $this->Form->radio('cough', $options);
    echo $this->Form->radio('wheeze', $options);
    echo $this->Form->radio('sleep', $options);
    echo $this->Form->radio('activity', $options);
    echo $this->Form->input('peakflow');
    echo $this->Form->input('weight');
?>
</fieldset>
<?php echo $this->Form->end(__('Submit', true));?>
```

---

17 Google Chrome is based on Chromium found in the repositories of Ubuntu 10.04.

The code above will produce a HTML form with radio buttons. Depending on the clicks of the user the inputs from the radio buttons will be a number from 1-5. Peakflow has to be a number – and CakePHP can validate the inputs from the user. Radio buttons are great because the user has to give a predefined input.



*Illustration 4: Screenshot of cake/posts/add with radio buttons.*

CakePHP can produce codes like the above more or less automatically. From the commandline you can produce code for presenting data, add, edit or delete posts and so on. Of course the code could be used as is – but on the other hand the tables are not that pretty. I wanted a website that looked a bit like the graphs produced in an asthma diary. CakePHP cannot bake this automatically.

## Sketch CakePHP Asthma Diary

At all times I carry a notebook for ideas. To the left is one of my first rough sketches. There could be a picture of the patient – or perchance an avatar if the patient wanted to be anonymous. There should be graphs created by the numeric data from the observations of the patient(s). Cough, Wheeze, Peakflow and so on would be printed as graphs.

The doctor, patient or scientist could perhaps choose from different views – but at first I wanted to focus on the actual asthma diary for the patient.

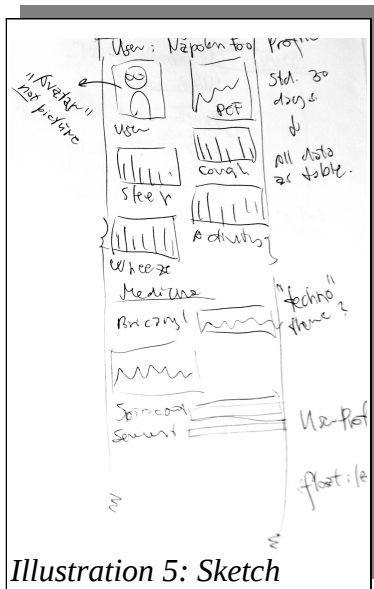


Illustration 5: Sketch

I knew that PHP could produce images. You define a canvas, and you can draw a line. The user input could somehow be transformed to the coordinates of the line. The solution seemed simple enough. But on a certain blog<sup>18</sup> about making images via PHP there was a comment somewhat like this:

- “Why don't you use CSS for the graphic?”

Images demand more bandwidth than sizing a div-tag. So I decided to make the bars by a combination of PHP and CSS on a div tag. I needed a function, that would do this job for me – but where do you place your own functions in CakePHP? To be honest, I did not know – and went to the IRC channel #cakephp in order to see whether someone knew what to do here.

A chatter suggested writing a helper – and that was the clue I needed. CakePHP ships with a bunch of helpers, but you can hack your own helper class too.

## The Helper Hack

Here is the helper class, that I wrote and placed in ../cake/app/views/helpers:

```
class ShowBarHelper extends AppHelper {
    public function vertical($size, $scale){
        echo "<div class='soejle' style='height:15px; width:" . $size *
        $scale . "px; background-color: #569a4a'> </div>";
    }
}
```

In order to work the helper has to be defined in the controller:<sup>19</sup>

```
var $helpers = array('Form', 'Html', 'showBar');
```

Then you can use the helper from a view. In this case the code is used in a foreach loop fetching data from cough:<sup>20</sup>

```
<?php echo $this->showBar->vertical($post['Post']['cough'], 100); ?>
```

The code calls the helper class showBar function vertical(\$size, \$scale). \$post['Post']['cough'] retrieves a numeric value (0 - 5) from the database and 100 will scale the div (if the input was 2 the width is 2\*100px).

18 I don't remember the actual address of that blog.

19 It is found in ../cake/app/controllers/ in the users\_controller.php and posts\_controller.php.

20 The actual code is found in ../cake/app/views/posts/index.ctb in my prototype.

In the end the helper will output HTML like this:

```
<div class='soejle' style='height:15px; width:100px; background-color:#569a4a'>21
```

Since the helper was “fired” in a loop the result looks like this in a browser view:

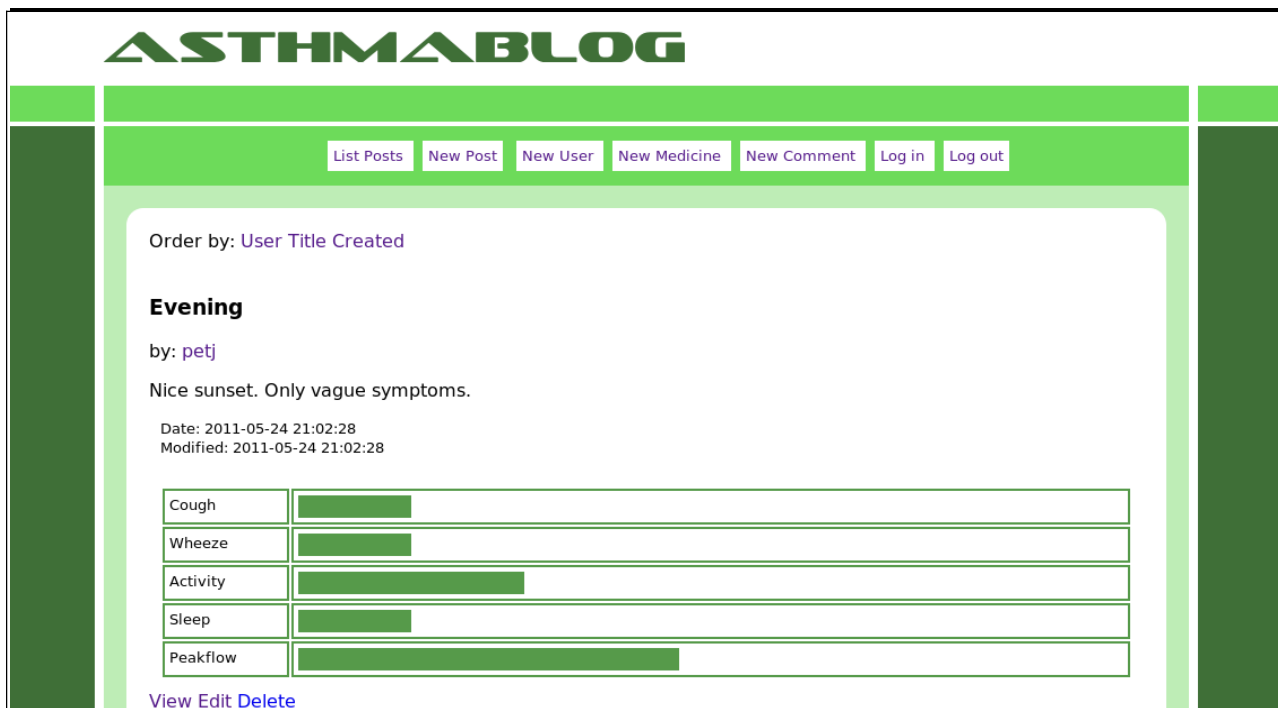


Illustration 6: Bars produced by the helper hack. For each blog post there will be bars representing the data from the observations.

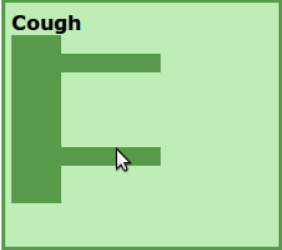
The helper hack can be used in many ways. In illustration 6 you see separate posts day by day. These graphs give a picture of the conditions here and now – as you would do it in a blog. But if you want to see how conditions are developing you could loop through each observation as indicated in illustration 7.

The experiment demonstrates that you can size divs with PHP. And so the basic functionalities of the Asthma Blog are ready for design iterations. Should I want to generate a linear graph PNG image the ShowBarHelper could have a function doing this for me too.

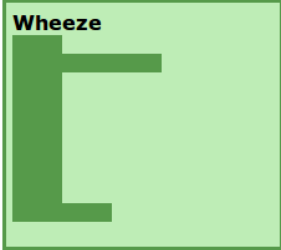
<sup>21</sup> Defining the background-color here is not a good idea. Such definitions ought to be placed in my css.css in order to get the design definitions in one place.

[List Posts](#) | [New Post](#) | [New Medicine](#) | [New Comment](#) | [Log in](#) | [Log out](#)

**Cough**



**Wheeze**



**Profile**

**petj**

**Medicine**

Name	Doze	Times	Created	Modified	Actions
Serevent	50	2	2011-05-24	2011-05-24	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>
Bricanyl Turbohaler	4	1	2011-05-30	2011-05-30	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>
Singulair	10	1	2011-05-24	2011-05-24	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>
Spirocort	400	2	2011-05-24	2011-05-24	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>
Diprosan	7	1	2011-05-25	2011-05-25	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>

*Illustration 7: Another way of using the data looping through either 'Wheeze' or 'Cough' etc. You can see the development of this particular observation. These bars are produced by the helper hack too.*

## Conclusion

This report is an attempt to demonstrate how the “paper and pen asthma diary” could benefit from Web 2.0 technologies. In order to do that I analyzed some possibilities, made user scenarios and described a MVC data architecture based on CakePHP. In the prototype input from the user are rendered as graphs.

There is a great potential in the asthma blog on the WWW. First of all the data can be saved and presented in many different forms and views. If the asthma blog would reach a critical mass among asthma patients the data could be useful for the medical industry and the universities as presumed in the scenarios.

However there are still some ethic questions to be answered: How to protect the user's privacy? Should the blog posts be public or hidden?

A blog with hidden posts may seem a bit strange in a Web 2.0 community – but your health is a very private thing. In real life I think, that the asthma blog ought to be a private tool. Therefore the public blog part of my sketch is nothing but an illustration of the asthma blog opportunities.

To me the Asthma Blog is a more useful tool than a printed asthma diary – and perhaps that could be the case for other asthma patients too.

So I'll continue the prototyping iterations with CakePHP...

## References

### **Printed**

- Bouvin, Niels Oluf; Hansen, Allan: "Kompendium – Hypermedier og Web" (2011)
- Itten, Johannes: "Farvekunstens elementer" (1970)
- Jensen, Ole Ingolf: "Farvernes Metafysik" (2000)
- Kalbach, James: "Designing Web Navigation" (2007)
- McNeil, Patrick: "The Web Designer's Idea Book" (Ohio 2008)
- Morrison, Michael: "Sams Teach Yourself XML in 24 Hours" (2005)
- Porebski, Bartoz et al: "Building PHP Applications with Symfony, CakePHP and Zend Framework" (2011)
- Sklar, Joel: "Principles of Web Design" (2009)
- Studio 7.5: "Navigation for the Internet and other Digital Media" (2002)

### **Web**

(The web references were used in the form they had from May – June 2011.)

API CakePHP: <http://api.cakephp.org/>

Asthma Diary (PDF): [www.mymainehealth.com/documents/Asthma%20Diary.pdf](http://www.mymainehealth.com/documents/Asthma%20Diary.pdf)

Image processing with PHP: <http://php.net/manual/en/book.image.php>

Peakflow: Green, Yellow, Red: <http://www.lungusa.org/lung-disease/asthma/living-with-asthma/take-control-of-your-asthma/measuring-your-peak-flow-rate.html>

Php.net: <http://php.net/>

Ruby on Rails: <http://rubyonrails.org/>

The CakePHP 1.3 Book: <http://book.cakephp.org/>

W3Schools: <http://w3schools.com/>

XML for Excel spreadsheet: [http://msdn.microsoft.com/en-us/library/aa140062%28v=office.10%29.aspx#odc\\_xlsmlinss\\_hierarchy](http://msdn.microsoft.com/en-us/library/aa140062%28v=office.10%29.aspx#odc_xlsmlinss_hierarchy)

XML for Open Office Spreadsheet: <http://www.ibm.com/developerworks/xml/library/x-oocalc/?ca=dgr-lnxw16XSLT-OOo>