

Astmadagbogen - et plugin til WordPress

5. november 2011



Opgave i Webapplikationsudvikling Modul III

af Per Thykjær Jensen

email: per@multimusen.dk

årskort: 19810234

Vejleder: Susanne Bødker

Instruktør: Peter Urbak

Århus Universitet / IT Vest November 2011

Link til fungerende prototype: <http://multimusen.dk/asthma-plugin>

Download kildekode: <http://multimusen.dk/ftp/asthmadiary.zip>

Login: patient - Kode: tneitap

Indhold

1 Astmadagbogens iterationer	3
1.1 Inspiration til projektet	3
1.2 Udvikling af astmadagbog og metode	3
1.3 Iterationer	4
1.4 Hvorfor Wordpress?	4
2 Fænomenologisk tilgang til astmadagbøger	4
2.1 Dagbogen - et privat eller offentligt rum?	5
3 Lignende teknologier	6
3.1 Blog, log, dagbog	6
3.2 Sammenfatning af lignende teknologier	8
4 Systemarkitektur	8
4.1 Wordpress arkitektur	8
4.2 The Asthma Diary Plugin og MVC-arkitekturen	9
4.3 Kontrolpanelets indtastningsformular	10
4.4 \$wpdb og kontrolpanelet	11
4.5 Præsentationsinterfacet	11
4.6 Fra tal til graf	12
5 Teknisk review og evaluering	13
5.1 Evalueringer	14
6 Konklusion	15
7 Appendix	16
7.1 Evaluering med Jan Skovgård	16
7.2 Email-evaluering af Michael Hanson	16
7.3 Teknisk review	18
7.4 Web analytics	18
7.5 WordPress	18
7.6 Asthma Diary kort beskrivelse af plugin samt koden	19
7.6.1 asthmadiary.php	19
7.6.2 charts.php	21
7.6.3 asthma-diary-result-presentation.php	24
7.6.4 form.inc	24
8 Kildefortegnelse	26
8.1 Litteratur	26
8.2 Internet	26

1 Astmadagbogens iterationer

Formålet med dette arbejde er at udvikle en astmadagbog, der skal fungere i et webbaseret system. Astmadagbogen skal omsættes fra lægernes traditionelle papirbaserede optegnelser til et design, der bruger nettets muligheder for indtastning i formularer og grafisk visualisering af data. Hensigten er at udvikle et webbaseret system, der kan integreres i websider uden at jeg behøver at administrere alle databaserne personligt. Derfor besluttede jeg at udvikle astmadagbogen som et plugin til WordPress, der er bygget via PHP og en MySQL database.

Designet er udviklet fra skitser til en prototype, der er blevet afprøvet af ekspertbrugere i et teknisk review og evalueret med inspiration fra [Tognazzini] og [Nielsen]s teser om usability. Arbejdet afsluttes med en evaluering af design, teknisk konstruktion og brugergrænseflade. Et er, hvad der teknisk er muligt - noget helt andet er, om en bruger kan finde ud af at bruge astmadagbogen.

1.1 Inspiration til projektet

For et års tid siden ville læge Henrik Olesen undersøge om "Singulair" havde en positiv virkning på min astma. Han bad mig om at føre en astmadagbog. Lægen gav mig en astmadagbog, der var trykt på stift papir. På et diagram skulle jeg indskrive peak flowmålinger¹ på noget, der lignede milimeterpapir.

Over tid kunne man tegne grafer, der viste om peak flow forbedredes, eller ej. Ud fra disse objektive målinger kunne jeg sammen med min læge konstatere, at "Singulair" var et godt supplement til behandlingen. Tilfældigvis fulgte jeg på samme tid Master IT forløbet Webapplikationsudvikling Modul II.²

Forelæserne handlede om brug af blogs og wikier. Derfor var det måske en naturlig reaktion at spørge, om man ikke kunne udvikle en webbaseret astmadagbog? Derfor vil jeg prøve at udvikle en webbaseret astmadagbog, der kan bruges af astmatikere til monitorering af deres helbredstilstand?

Astmadagbogen skal kunne bruges af læger og patienter til evaluering af astmatikerens helbredstilstand.

1.2 Udvikling af astmadagbog og metode

I forløbet har jeg brugt dagbøger som metarefleksorisk ramme. Samtidig er det produkt, jeg udvikler, en dag- eller logbog. Dagbøger har spillet en væsentlig rolle siden gennem alle Webapplikationsudviklingsmodulerne.

Personligt har jeg siden ungdommen brugt dagbøger som udviklingsredskab - både privat og professionelt. Derfor er det oplagt undersøge samspillet mellem dagbogens tankeskrivning og udvikling af koden til den webbaserede astmadagbog.

Her vil jeg diskutere [Jepsen], Mathiassen og Nielsens artikel "Back to thinking mode: diaries for the management of information system development projects".

¹Måling af lungefunktion kaldes peak flow. Målingen foregår fx ved at puste i et MiniWright peakowmeter.

²Rapport og produkt fra Webapplikationsudvikling Modul II findes på: <http://multimusen.dk/cake/>

1.3 Iterationer

Astmadagbogen på nettet er en ide, der er udviklet gradvist gennem en række iterationer.

1. Iteration I var en ikke-persitent blog udviklet i PHP på Webapplikationsudvikling Modul I. Her udviklede jeg en lille funktion, der ved anvendelse af få attributter kunne fremstille radiobuttons. Denne nyttige funktion har jeg genbrugt i den nuværende udgave af astmadagbogen. Tanken var dengang, at astmatikere skulle kunne blogge og snakke om astmasymptomer i bedste Web 2.0 stil. Men er det nu en god ide?
2. Iteration II skabtes som en primitiv prototype i frameworket CakePHP. Prototypen blev udviklet på Webapplikationsudvikling Modul II. CakePHP-skitsen demonstrerede, at ideen kunne realiseres. Brugerens input kunne gemmes i en database, og observationerne kunne via PHP transformeres til søjlediagrammer, hvor søjlerne div-tags, der skalerede via PHP og talværdier fra databasen. Men man kunne ikke printe diagrammerne!³ Under udviklingen af prototypen gik jeg væk fra den åbne blog. En astmadagbog er jo et meget privat dokument på linje med en sygejournal.⁴
3. Det nuværende arbejde er med andre ord Iteration III. Det er min hensigt at videreudvikle astmadagbogen til et WordPress Plugin. Astmadagbogen skal være et privat dokument. Data skal visualiseres i billedfiler. Dokumentet skal kunne printes og medbringes til egen læge.

1.4 Hvorfor Wordpress?

Iteration II blev udviklet i CakePHP. Løsningen krævede, at jeg satte en server op og bestyrede den centralt. Hvis astmadagbogen skal vinde større udbredelse, så bør den nok have en form, der kan integreres i et CMS. Wordpress [WP] er et CMS, der er uhyre populært. På verdensplan skønnes, at der er lidt over 59.000.000 WP-blogs.⁵ Wordpress er et open source CMS.⁶

Fordelen ved at anvende WP er at systemet (som regel) er stabilt. WP er bygget via PHP og MySQL. WP har en veldokumenteret API⁷. Derfor kan webudviklere bygge tilføjelser via PHP og databaser. Der er et stort forum af brugere, der deler erfaringer med WP. Et aktivt og hjælpsomt forum er en stor fordel for enhver udvikler på nettet.

2 Fænomenologisk tilgang til astmadagbøger

Sandsynligvis findes undersøgelser, der dokumenterer effekten af de trykte astmadagbøger, som lægerne udleverer til astmatikere. Sådanne rapporter forholder jeg mig ikke til, selv om en det med stor sandsynlighed kunne forbedre mit produkt.

³Diagrammerne var tomme div-tags, hvor width blev stilet via PHP.

⁴Rapport og produkt fra Webapplikationsudvikling Modul II er her: <http://multimusen.dk/cake/>

⁵<http://en.wordpress.com/stats/>

⁶WP kan downloades fra <http://wordpress.org>.

⁷Application Programming Interface

Jeg har en fænomenologisk tilgang til astmadagbøger. Udgangspunktet er en observation: min læge gav mig et eksemplar af en astmadagbog trykt på karton. Og så undrede jeg mig over valget af papir som medie. Det virker mere oplagt at anvende en webteknologi til formålet.

Der findes astmadagbøger, som kan downloades fra nettet; men jeg fandt mærkeligt nok kun .pdf eller .doc filer. Nettets astmadagbøger er som regel en primitiv kopi af papirversionerne. Hvorfor ikke udvikle en astmadagbog, der udnytter nettets og computerens styrker?

2.1 Dagbogen - et privat eller offentligt rum?

[Jepsen] hævder, at udviklere med fordel kan bruge kollektive dagbøger som udviklingsmetode. Dagbogen etablerer et rum, hvor udviklerne kollektivt kan reflektere over udviklingsprocessen. Jeg vil hævde, at en personlig dagbog - et metarefleksorie - også kan være et nyttigt redskab i udviklingen af webproduktioner.

[Jepsen] påstår, at udviklerne opnår en bedre planlægning ved at reflektere over deres praksis.

Surrealisten André Breton forsøgte fra 1930-erne at skrive værker med "ren psykisk automatisme". Kunsten skulle befries fra kulturens freudianske superego. Men surrealisterne foretog kunstneriske valg og fravalg i deres prosa, poesi, malerier og skulpturer. Surrealistiske værker blev således redigerede produktioner, der tilstræber at ligne noget drømmeagtigt. Og det er altså ikke "ren psykisk automatisme".

Værkerne ændrer altså karakter, når forfatterne ved, at teksten skal publiceres.

Tilsyneladende er der langt fra kunstnerens libidøse fantasier til den tekniske systemudviklers kollektive dagbøger; men måske kunne man spørge om ikke systemudvikleren kunne lære et og andet af kunstnerens rablende metodik?

[Jepsen]s praktiske vejledning i kollektiv dagbogsskrivning handler om at systematisere den kreative udviklingsproces. Tanken skal omsættes til handling. Der tænkes i logosserier i stil med denne:

1. Action
2. Reaction
3. Comment
4. "goto 1" (sekvensen gentages)

Udvikleren snakker med nogen, observerer et problem og søger en løsning på problemet. [Jepsen]s dagbøger er kollektive produktioner. Det sætter i sig selv en grænse for, hvad skribenterne kan tillade sig skrive og tænke. De viste fragmenter forholder sig yderst konkret til problemstillingerne. Vejledningen til dagbogsskriverne ligner en teknisk manual [Jepsen: 217]:

1. Make the intention clear
2. Be disciplined and careful.
3. Make a checklist of issues.

4. Be selective and thorough.
5. Decide on when.
6. Reflect on how.
7. Consider using related techniques.

På WAU modul I skulle vi skrive kollektive wikier eller dagbøger med noter over emner fra undervisningen. I den kollektive wiki skrev jeg betydeligt mere akademisk end i min private wiki "Metarefektoriet", der blev brugt til personlig ideudvikling på WAU modul II. Det samme kan siges om min blog "Pers Computer Cyprianus". Privat⁸ skriver jeg meget mere løssluppet, fx denne patosudladning foranlediget af frustrationer over CakePHP:

"Det er lidt som at gå over åen efter vand, når man skal pløje gennem lange tekster, der forklarer noget [om CakePHP], der kunne være ordnet med en simpel SQL-sætning. Fx prøver jeg at sortere indlæg i en blog, så det nyeste vises først. Selv includes kræver mystiske og magiske metamanipulationer..." (3.5. 2011)

Webudvikleren kan have stor gavn af private skrivelier, hvor følelser, drømme, fantasier og tanker får lov til at flyde frit.

[Jepsen]s ideer til lister, aktion og reaktion er nyttige redskaber; men det er kun en lille del af de mange kreative skrive- og tegnetoder, der kan bruges i en dagbog.

En skrivepraksis behøver ikke nødvendigvis at tænke i kollektive "observation -> problem -> løsning" sekvenser. En metarefektorisk skrivepraksis i dagbogens private rum kan også være et nyttigt instrument for en udvikler.

3 Lignende teknologier

3.1 Blog, log, dagbog

Teknisk set ligner "The Asthma Diary" en blog. Men den adskiller sig på flere måder fra den traditionelle personlige weblog. Som regel er en blog offentligt tilgængelige og personlige causerier. Læsere har mulighed for at kommentere indlæg. Patienters journaler er private dokumenter - de ligner det offentlige rum fra Web 2.0.

Generelt minder en blog om en traditionel dagbog. Dog med den væsentlige forskel, at bloggen er offentligt tilgængelig. Med Web 2.0 ser det ud til at den habermaske sondring mellem den private og den offentlige sfære er ved at blive nedbrudt. Jeg antager, at de fleste dagbogsskrivere anser deres tekst for at være et meget privat dokument:

"Diaries are a wellknown media for private, personal reflection (Rainer 1983). Rainer describes the general intention of keeping a diary as the personal development of the author. The idea is to understand oneself better.¹¹ " [Jepsen: 214]

⁸Bemærk, at bloggen og wikien er skrevet i internettets offentlige rum. Teksten er kun "privat" i den forstand, at jeg skriver til mig selv - ikke til et kollektiv af udviklere.

Astmadagbogen ligner ikke den traditionelle private dagbog. Der er ikke et dannelses- eller selvudviklingsaspekt i astmadagbogen. Astmadagbogens vigtigste oplysninger er ikke ”refleksioner”; men derimod målinger og observationer.

Astmadagbogen er et arbejdsredskab - en logbog. Den er et ”kollektivt dokument” i den forstand, at en læge skal kunne evaluere observationerne. Astmadagbogen kan altså defineres som et ”teknisk dokument”, der på en tilstræbt objektiv måde registrerer udviklingen i et sygdomsforløb.

”The Asthma Diary” vil med få ændringer kunne bruges af forskere, fx som log i medicinske eksperimenter. En WP plugin kan bruges af alle bloggernes brugere, og derfor vil det være enkelt for forskerne at videreudvikle koden, således at man kan evaluere mange patienters observationer. Databasens samlede resultater kan forholdsvist enkelt gøres tilgængeligt for forskere; mens patienterne udelukkende ser deres personlige resultater.

Databasen kan via PHP udføre og evaluere beregninger af gennemsnitsværdier, som det muligvis kan være vanskelige for patienterne at gennemskue. Talværdier kan omsættes til grafer ved anvendelse af PHP og specialudviklede libraries.⁹ Det var måske mere privat at udvikle en applikation til slutbrugerens operativsystem; men her kan der være betydelige problemer med platforme og filformater, der ikke kan enes - hvilket besværliggør udviklingen.¹⁰

Svagheden ved webproduktionen er at brugeren skal have netadgang når webdagbogen bruges. En webastmadagbog kan noget, som den ”analoge” model ikke kan. Både læger og patienten kan få adgang til data online. Det er urealistisk at forvente, at lægen følger samtlige patienter online. Derfor er det vigtigt, at dagbogens observationer kan printes og medbringes til lægen.¹¹

Astmadagbogens målinger og observationer skal anvendes af en klart defineret kreds: patienten og professionelle behandlere. Astmadagbogen bør ikke have bloggernes traditionelle offentlighed. Brugeren skal med andre ord have mulighed for at føre dagbogen privat. Hvis dagbogen skal fungere i blogsammenhæng, så skal det være muligt at indtaste data, der kun kan læses af personer, som patienten har tillid til.

I WP vil jeg derfor udvikle et system, hvor kun registrerede brugere har adgang til data. Teknisk set er der ikke den store forskel på en traditionel blog og den løsning som skitseres ovenfor. Den væsentligste forskel mellem astmabloggen og en typisk Web 2.0 weblog er:

- Weblog: tekst, billeder og mulighed for debat med læsere.
- Astmadagbogen: tekst, grafer, diagrammer og tolkning af data via boolske sætninger. Brugers data er private.

Astmabloggen udnytter PHPs regnefunktioner og mulighederne for logiske kalkulationer.¹² Astmadagbogen skal kunne noget, som man ikke ser i den gængse weblog. Den webbaserede astmadagbog kan automatisk udføre de relativt komplekse beregninger, som skulle udføres manuelt i ”papastmadagbogen”.

⁹Fx: <http://www.techrepublic.com/article/create-graphics-on-the-y-using-php/5092227>

¹⁰Lavpraktisk kunne man udvikle et interface til et regneark eller en lokal database.

¹¹Her var svagheden ved iterationen i Modul II. Søjlediagrammerne kunne ikke printes!

¹²F.eks. kan astmabloggen beregne gennemsnittet af peakowmålinger og vurdere om en ny måling er over eller under dette gennemsnit.

3.2 Sammenfatning af lignende teknologier

Astmadagbogen på nettet videreudvikler traditioner fra trykte medier. Lægeverdenen har en lang tradition fra naturvidenskaben med observationer og målinger, der gemmes i en logbog eller markeringer i et koordinatsystem over tid.

Astmadagbogen på nettet stiller nogle helt specielle krav. I modsætning til en ordinær blog er astmadagbogen et privat dokument. Den er et teknisk dokument, der kun bør kollektivt i den forstand at indholdet skal kunne anvendes af patient og fagpersoner, der så kan fortolke resultaterne. Astmadagbogen er således et betydeligt mere privat dokument end en gængs blog.

Teknisk set kan blogs udvikles på grundlag af koderammer, der allerede er definerede. CakePHPs styrke er at man hurtigt kan udvikle en prototype. Svagheden ved CakePHP er håbløs mangelfuld dokumentation og bugs.

Det er min oplevelse, at WP er veldokumenteret, der er en forståelig API og anvendelsen af PHP er relativt enkel for webudvikleren.

4 Systemarkitektur

Et plugin er en lille brik i en langt større konstruktion. Et plugin tilføjer funktionalitet til WordPress CMSen.

4.1 Wordpress arkitektur

WP har en arkitektur, hvor udviklere kan tilføje ekstra funktionalitet i form af widgets og plugins. Widgets er små tilføjelser, der kan placeres i templatens widgetområder. Plugins kræver ikke widgetområder. Plugins kan principielt placeres hvorsomhelst i WP.

WP brugeren kan tilføje og redigere links, faste sider samt blogposter fra kontrolpanelet. Brugerens rettigheder defineres i et hierarki af roller, hvor fx administrator kan "alt", redaktører kan redigere brugeres indlæg, forfattere kan forfatte - og en bruger, der ikke er registreret i systemet kan kun se det, som udgiverne ønsker skal være synligt.

Det sidste er vigtigt og nyttigt med hensyn til "Asthma Diary", eftersom en sygejournal må karakteriseres som et særdeles privat og følsomt dokument. I WP kan forfatteren vælge, at en side er "privat".

Plugins er andre tilføjelser, som administratoren eller andre WP-udviklere har programmeret. Et plugin kopieres til mappen /wp/wp-content/plugins. Mit plugin ligger fx i mappen /.../multimusen.dk/wp-content/plugins/asthmablog/ Plugins kan præsenteres overalt i WP produktionen ved hjælp af programkroge. "Asthma Diary" har således flere interfaces, der vises i kontrolpanelet, eller på sider ved hjælp af shortkoden [asthmablog] samt på en redigeringside.

Hver side i WP kan principielt få et skræddersyet design. Det kan ske på flere måder; men det mest enkle er at udvikle en template og placere den i: /wp/wp-content/themes/navn-paa-template

Principielt kan man nøjes med en PHP-fil, der indeholder en række WP tags, der igen præsenterer indholdet for brugeren.

Det kan være lidt kompliceret at skræddersy designet til undersider; men det følgende diagram fra WP Codex demonstrerer principperne i sidehierarkiet. Index.php yderst til højre er den aktive templates indexside. Hvis man ønsker

4.3 Kontrolpanelets indtastningsformular

En WP plugin er en PHP-fil, der gemmes i mappen /webroot/wp-content/plugins/ eller en undermappe [Brazell: 72 ff.]. WP genkender en plugin så snart en header i stil med den følgende anvendes:

```
<?php /* Plugin Name: Asthma Diary
Plugin URI: http://multimusen.dk/
Description: ... Version: 0.1
(... etc ...) */
?>
```

Herefter følger klasser, funktioner og kode, som skal anvendes. I mit tilfælde er der behov for to interfaces, der kaldes med hver sin "krog".¹⁹ Indtastningsformularen i kontrolpanelet defineres sådan:

```
// DASHBOARD
// skaber widget til dashboard
function dashboard_widget_function() {
    require('form.inc');
} // action krog

function add_dashboard_widgets() { wp_add_dashboard_widget('dashboard_widget',
'Asthma Diary',
'dashboard_widget_function');
```

Pluginet kan herefter vise astmadagbogens indtastningsformular i WP kontrolpanelet.

Formularen (form.inc) er en gængs HTML formular. Eftersom der skulle anvendes mange radiobuttontags, skrev jeg dette PHP loop, som udfører det meste af opgaven automatisk:

```
function lavRadio($astma, $n){ $t = 1; echo $astma."<br />1";
while ($n > 0) {

    print "<input type='radio' name='$astma' value='$t' />\n";
    $t = $t +1; $n = $n - 1; }
    echo " 5<br />";
}
// Anvendelse: <?php lavRadio('Cough', 5); ?>
```

Ovenstående skriver "Cough" over 5 radioknapper.

Formularen kan styles via WP CSS for kontrolpanelet. Derfor er formularens kode meget basal. Formålet med denne formular er udelukkende at give mulighed for at gemme relevante data i databasen. I øvrigt er formularen gemt i en selvstændig fil, for så kan den om nødvendigt genbruges i anden sammenhæng, f.eks. hvis jeg skulle få lyst til at skrive til databasen fra præsentationsinterfacet.

4.4 \$wpdb og kontrolpanelet

I WP foregår kommunikation med databasen via klassen \$wpdb. \$wpdb skal sættes som global. Klassen kommunikerer med databasen - eller i det mindste den database, hvor de øvrige WP tabeller er placeret. \$wpdb kan udføre alle former for SQL-sætninger.

```
global $wpdb; // The WP database controller class
$table_name = "asthmablog"; // name of table
$valuestring = "'". $_POST['Note'] . "',"
. $_POST['peak flow'] . "',"
. $_POST['Cough'] . "',"
. $_POST['Wheeze'] . "',"
. $_POST['Sleep'] . "',"
. $_POST['Activity'] . "',"
. $_POST['Name'] . "',"
. $_POST['Date'] . "'";
$insert = "INSERT INTO "
. $table_name
. " (ab_note, ab_peak flow, ab_cough,
ab_wheeze, ab_sleep, ab_activity,
ab_name, ab_dat
. "VALUES ("
. $valuestring
. ")";
$results = $wpdb->query($wpdb->prepare( $insert ));
```

I dette tilfælde hentes indtastninger fra formularen via \$_POST. Bemærk “prepare” i sidste linje:

```
$wpdb->query($wpdb->prepare( $insert ));
```

Her kalder mit plugin den WP klasse, der beskytter mod ondsindet kode, såsom SQL injections og andre ubehageligheder.

4.5 Præsentationsinterfacet

På lægens papastmadagbog tegnes observationerne efterhånden som patienten foretager sine målinger af lungefunktionen. Astmadagbogen på nettet henter observationerne fra databasen og tegner talværdier som grafer i et diagram. Desuden fortolkes den sidst indtastede værdi som grøn, gul eller rød. Det grafiske layout af disse data kaldes her præsentationsinterfacet.

Astmabloggens præsentationsinterface kan hentes frem på en side ved hjælp af en shortcode. Det betyder, at en funktion kan eksekveres i templates, pages eller i posts ved at skrive [asthmadiary] i posten. For webudvikleren åbner shortcodes for en effektiv anvendelse af PHP i WP.

Den ønskede funktionalitet er nu mulig: en indtastningsformular i kontrolpanelet samt et præsentationsdesign.

4.6 Fra tal til graf

Formålet med præsentationsinterfacet er at give et overblik over astmadagbogens poster. Grafer opsummerer de seneste 30 indtastninger. Astmadagbogen skal være et privat dokument. Den skal kun være synlig for en bruger, der er logget ind. Derfor denne logik:

```
if ( is_user_logged_in() ...
```

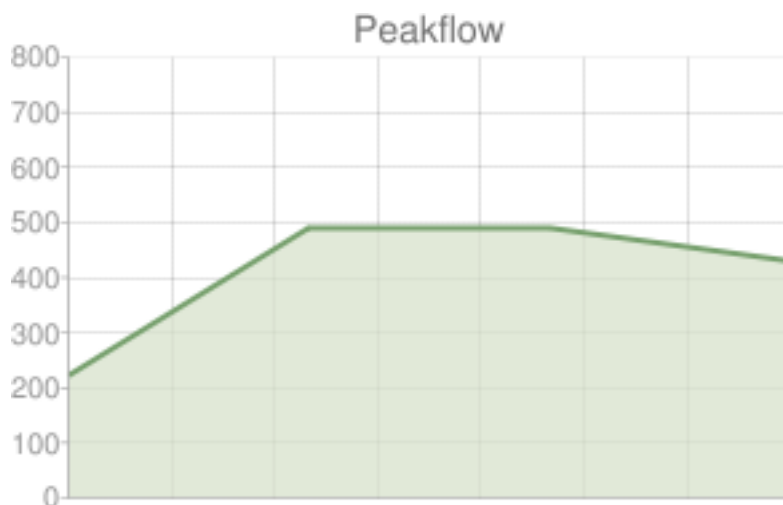
Hvis udsagnet er sandt importeres `asthma-diary-charts` klassen, der kan omsætte talværdier til grafer og fortolke om peak flowværdien er grøn, gul eller rød. Graferne tegnes ved hjælp af funktionen `ad_peak_flow_chart()`, der består af disse dele:

1. Først registreres brugernavnet via `$current_user->user_login`.
2. Så skabes en SQL sætning i form af en streng.
3. PHP returnerer et array, der imploderes.
4. Resultat: en serie af tal adskilt af kommaer (f.eks. peak flowværdierne)
5. Talværdierne samles til en Google Charts URL.
6. Google Charts er en API, der fremstiller grafer og diagrammer.

Et `img`-tag linker til en URL, der igen fremstiller et PNG billede. Google API strengen er meget lang, og jeg har af derfor delt den i tre dele:

```
$img_start = "<img src='http://chart.apis.google.com/chart?
chxr=0,0,5
&chxt=y&chbh=a&chs=300x225&cht=bvg
&chco=A2C180&chds=0,5
&chd=t:";
$img_end = "&chtt="
. $label
. "&chts=676767,14"
. "' width='300' height='225' alt='"
. $label . "' />";
// Echo the graph to the browser
echo "<div class='asthmadiary_pef'>";
echo $img_start // del I (begyndelsen af <img src...
. $comma_result // del II (tal fra databasen)
. $img_end; // img tagget afsluttes
echo "</div>";
```

Resultatet er en graf i stil med denne:



Klassen kan genbruges flere steder i mit login. Klassen skal blot inkluderes, og derefter kan man udnytte klassens forskellige funktioner:

```
$asthma_info::ad_peak_flow_chart();
```

Når graferne skabes af PHP er hele MVC-modellen i spil.

5 Teknisk review og evaluering

Selv om jeg i rollen som webudvikler synes, at min webløsning er rigtig god, så er det jo ikke sikkert, at brugerne tænker det samme. Her ligger den klassiske problemstilling for mange webudviklere. Eksperten kender sandsynligvis hverken sin brugers behov eller baggrund.

Evaluering kan vise om et GUI og den tekniske konstruktion fungerer efter hensigten [Schneiderman] [Nielsen] [Obendorf] [Tognazzini] [Clement].

“The Asthma Diary” er analyseret via et teknisk review og to evalueringer af: Jan Skovgård (JS) og Michael Hanson (MH)

Begge reviewere er studerende på Webapplikationsudvikling modul III, og derfor er der tale om ekspertudsagn. Fordelen ved eksperter inden for et område er at de ved, hvad de har med at gøre.

[Tognazzini] argumenterer for, at udviklerne kan få gavn af relativt enkle evalueringer. Hvis man afprøver produktionen et antal brugere, så vil det give oplysninger, der kan medvirke til at forbedre brugeroplevelsen på sitet. Det er baggrunden for de meget enkle kognitive tænke / skrive højt øvelser, som vi udsatte min plugin for.

5.1 Evalueringer

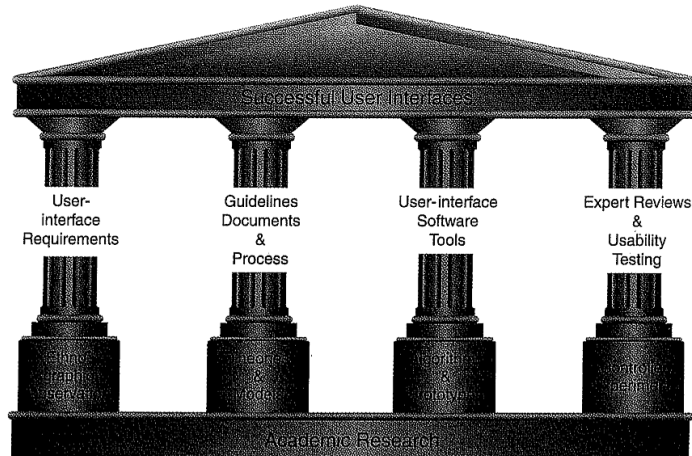


FIGURE 3.1
The four pillars of successful user-interface development.

Ifølge [Schneiderman] bør webudvikleren arbejde med disse fire områder:

- Brugerinterfacets behov beskrives. Dette arbejde skete i Webapplikation-sudvikling Modul II.
- Hjælpende dokumenter skrives.
- Softwareredskaber og brugerinterface udvikles.
- Ekspert reveiws og brugertest (se hans fig. 3.1 i [Schneiderman:121 ff.]).

Jeg gik nok lidt bagvendt til opgaven. I begyndelsen søgte jeg at definere, hvad mit plugin skulle kunne. Så langt fulgte jeg [Schneiderman]; men i stedet for at skrive hjælpetekster gik jeg direkte over til at udvikle en GUI. Og så glemte jeg ellers alt om hjælpetekster. Under arbejdet med de afsluttende evalueringer, baseret på [Tognazzini], [Nielsen] m.fl., blev det klart, at min metode var problematisk.

Den heuristisk inspirerede test blev lidt af en øjenåbner for mig. Når man ser en bruger, der undrer sig over, hvad han skal - så vækker det til eftertanke. JS¹⁵ ledte efter hjælpetekster; men mit site var baseret på de trykte astmadagbøger plus en formular til input.

Den løsning skal man dels være astmatiker og dels vide noget om astmadagbøger for at forstå. Derfor har jeg udviklet en hjælpeside; men på længere sigt bør hjælpeteksten være synlig ved siden af inputfelterne i formularen.

Bagefter snakkede vi om, at man måske skulle spørge mere konkret. Websiden skal kommunikere som en læge, der taler med sin patient:

- Hvordan har du det?
- Har du sovet godt i nat? Hvis ikke, beskriv dine symptomer 1 = "få hosteanfald" og den højeste værdi 5 = "Vedholdende og generende hoste".
- Prøv at puste i peak flow-måleren. Pust til!

¹⁵Se Appendix 7.1

- Pust tre gange i peak flowmåleren, og indtast den højeste værdi i dette felt

...

Disse spørgsmål skal komme i en sekvens af sider, således at brugeren kun skal forholde sig til et emne af gangen.

Michael Hanson (MH) havde en tydelig videnskabsteoretisk tilgang til emnet. Han tog udgangspunkt i Kaushik og Krug. Med disse teoretikere in mente skrev han en heuristisk gennemgang af websitet. Michael er i tvivl om, hvad han skal gøre på sitet:

“Jeg kan tilføje data om i felterne: Cough, Wheeze, Sleep, og Activity. De er alle radio-button felter. Jeg kan desuden tilføje information i feltet peak flow. Jeg prøver mig frem og indtaster en tekst i feltet peak flow. Det giver et resultat i feltet nedenfor under peak flow Indications:

“Your Current peak flow: 0 It is a Red peak flow Value – Indicates a medical emergency.’ Uha.” (MH, se: Appendix 7.2)

Denne undren fører til tanken om at en astmatiker måske forstår sagerne bedre:

“Det er muligt, at en astmatiker uden problemer forstår brugen af Pers plugin. Tallene, datatyperne og data-fremstillingerne lader til at høre til en vanlig information-kontekst for astmatikere, mens ikke-astmatikere muligvis – ligesom jeg – vil studse lidt over brugen af 16 plugin’et. Hvad angår placeringen af pluginet giver det helt sikkert mening, at tilføje data via kontrolpanel-delen af WordPress. Men jeg er ikke sikker på, at jeg helt forstår, hvorfor data-fremstillingen foregår bag gardinerne.” (MH, se Appendix 7.2)

Som webudvikler har jeg skabt en astmadagbog, forekommer meget teknisk for en ny bruger. Evalueringen siger, at dette plugin kan anvendes af mere nørdede astmatikertyper med en udpræget induktiv og tålmodig tilgang til websider.

Evalueringen siger mig, at astmadagbogen skal videreudvikles. Begreber skal forklares, og siden skal være forståelig for førstegangsbrugeren. Hvis brugeren skal kunne anvende mit produkt, så er jeg nødt til at kommunikere mere tydeligt.

6 Konklusion

I dette modul har jeg udviklet en astmadagbog på nettet. Data fra input bliver visualiseret som grafer og søjlediagrammer. PHP henter talværdier fra en database, og disse tal forvandles til søjlediagrammer og grafer med Google Charts API. Peak flow målinger fortolkes for brugen. Astmadagbogen på nettet kan være et nyttigt redskab for en astmatiker, der arbejder med sit sygdomsbillede.

Såvel produktet som proces rummer en latent fare for at glemme den bruger, som måske hverken kender astmadagbogens funktion eller forstår termerne i indtastningsformularen. [Tognazzini] har ret i at en brugertest ikke behøver at være vanvittig dyr for at give et brugbart resultat. De to relativt enkle evalueringer viste effektivt, hvordan brugeroplevelsen kan forbedres.

Testbrugerne var eksperter i den forstand, at de kender til webudvikling; men astma og peak flow kendte de intet til. Efter de to evalueringer stod det

klart for mig, at systemet talte i termer, som brugeren ikke forstod. Brugere kunne ikke finde en forklaring på termerne på websiden.

I skabelses-myten konkluderer gudinderne, at deres skabelsesværk er godt.¹⁶ Men har gudinderne spurgt brugerne om verden opleves sådan? Evalueringer viser om webudviklerens arbejde trænger til forbedring. Hvis man blindt konkluderer, at “alt er såre godt”, så udvikles sitet ikke!

Derfor er resultatet af evalueringen positivt. Evalueringen gav en række ideer til at få mit design til at kommunikere bedre og mere humant med brugerne.

Først og fremmest skal hjælpeteksterne udvikles. Og så skal præsentationen af resultater adskilles fra formularen.

Noget helt andet er, at der sker en bevægelse fra computerens browser til de mobile platforme. Det er oplagt at anvende en platform, som brugeren har adgang til døgnet rundt. Konklusionen på Webapplikationsudvikling Modul III er, at arbejdet ikke er tilendebragt.

Nye iterationer venter forude.

7 Appendix

7.1 Evaluering med Jan Skovgård

Jan Skovgård (JS) loggede ind i WP, der af en eller anden grund ikke fungerede på EAKs wifi. Med ledning til nettet var log in mulig. J- prøvede en række gængse browsere med samme resultat. (Chrome, Firefox, Internet Destroyer).

Kognitiv gennemgang

JS ledte efter informationer om, hvad websitet skulle kunne og hvorfor; men der var ingen vejledende fil. Han undrede sig over hvad “fill in the form 19 below” betyder. Det gik op for ham, at en form er en formular. Så udfyldte han den med værdien 123 og gav de forskellige radio buttons værdier. Han klikkede på ok.

- hvad sker der så.

JS kunne se, at graferne havde ændret sig lidt; men hvor var hans indtastning. Han ledte også efter noget der viste “red value”; men der var ikke noget som var rødt. Heller ikke i graferne. Betydningen af tolkningen var et problem for J-. Konklusion på kognitiv gennemgang; produktet mangler forklaringer på termer og begreber. Det er ikke sikkert, at en astmapatient ved hvad peak flow er - eller hvad værdierne af fx “cough” betyder. Måske skulle man begynde med at tale til patienten som et menneske ville gøre:

- Hvordan har du det?

- Har du hostet i nat? Hvis du har hostet, kan du angive en værdi her . . .

Og lignende.

Interfacet skal med andre ord være humant forståeligt.

7.2 Email-evaluering af Michael Hanson

fra Michael Hanson mh@silkets.dk via multimusen.dk
til Per Thykjær Jensen <per@multimusen.dk> dato 25. okt. 2011 12.57
emne SV: Feedback

¹⁶Bibelen 1. Mosebog (på hebraisk kaldes skaberen “Elohim”, ordet er hunkøn og pluralis).

“(…) Inspireret af “Data Collection – Importance and Options” af Avinash Kaushik og måske også lidt af Steve Krugs “Rocket Surgery Made Easy”, kaster jeg mig ud i en heuristisk analyse af Per Thykjær Jensens plugin til WordPress. Det betyder blandt andet en nådesløs testen af alle app’ens bruger-elementer. Ret beset ville det være mere korrekt at kalde analysen hermeneutisk, idet den skal forsøge at samle et forklarligt, betydningsfuldt hele ud fra en række detaljer, der i sig selv ikke nødvendigvis synes at bære nogen større betydning. Anyway, udstyret med link og login-oplysninger logger jeg ind på sitet. Det fører mig til admin-delen af wordpress-sitet.

I admin-delen kan jeg udføre de vanlige WordPress-aktiviteter: Jeg kan poste indlæg, uploade medier eller skrive kommentarer. Jeg kan også tilgå profil, værktøjer og kontakt. Ellers er kontrolpanelet rimeligt rensat for forstyrrende elementer.

En “Asthma Diary” tillader mig at indtaste data. Først og fremmest kan jeg indtaste data i form af en note. Hvis jeg ikke markerer og fjerner eller erstatter teksten “note”, der fra begyndelsen står i note-feltet, tilføjes den automatisk de indhentede data.

Jeg kan tilføje data om i felterne: Cough, Wheeze, Sleep, og Activity. De er alle radio-button felter. Jeg kan desuden tilføje information i feltet peak flow. Jeg prøver mig frem og indtaster en tekst i feltet peak flow. Det giver et resultat i feltet nedenfor under peak flow Indications: “Your Current peak flow: 0 It is a Red peak flow Value – Indicates a medical emergency.” Uha. Heldigvis viser det sig, at jeg har misforstået den datatype, der skal indtastes i feltet. Når jeg indtaster tal, giver den til at begynde med samme resultat. Men indtaster jeg høje tal, som fx 1200, får jeg følgende resultat: “Your Current peak flow: 1200 It is a Green peak flow Value”.

Scroller jeg længere ned ad siden, får jeg nogle meget fine grafiske fremstillinger af data i gråt, hvidt og grønt. Der er data langs y-aksen i form af tal-værdier, men ikke langs x-aksen. Jeg går ud fra, at data viser negativ eller positiv udvikling på y-aksen, mens x-aksen angiver et tidsligt forløb. Men jeg er ikke sikker.

Det er muligt, at en astmatiker uden problemer forstår brugen af Pers plugin. Tallene, datatyperne og data-fremstillingerne lader til at høre til en vanlig information-kontekst for astmatikere, mens ikke-astmatikere muligvis – ligesom jeg – vil studse lidt over brugen af plugin’et.

Hvad angår placeringen af plugin’et giver det helt sikkert mening, at tilføje data via kontrolpanel-delen af WordPress. Men jeg er ikke sikker på, at jeg helt forstår, hvorfor data-fremstillingen foregår bag gardinerne. Hvis man efter indlog til kontrolpanelet klikker videre på Bloggens navn, for at se front-end delen, kan man klikke sig videre til:

<http://multimusen.dk/your-asthma-diary/>

Her finder man også en historik, der beskriver med de tidligere nævnte noter, hvordan helbredstilstanden har udviklet sig over et længere forløb. Måske der kunne ligge en fordel i at begge dele, datafremstillingen, historikken og formularen til at indføre ny data lå sammen på front-end delen? Stadig med indlog måske?

Alt i alt en nem og let-tilgængelig applikation, der henvender sig til en specifik målgruppe og løser denne målgruppes behov på enkel vis. Som plugin er den desuden et venligt bidrag til andre brugere, der kunne ønske sig at holde et vågent øje med deres peak flow. Eneste forudsætning synes at være WordPress, men med et par millioner brugere burde det ikke være noget problem. Da teksten ligeledes er på engelsk er sproget i alle fald ingen forhindring.”

7.3 Teknisk review

Deltagere: Jan Skovgård (reviewer) og Per Thykjær Jensen (udvikler)

Produkt: “The Asthma Diary”

Dato: 20111014 11:05-11:35

Vi tog udgangspunkt i min dagbogsnote med en skitse, der viste, hvad astmadagbogen skulle kunne. Derefter blev pluginen demonstreret. Til sidst så vi på eksempler fra PHP filerne i koden.

Reviewet varede knap en halv time. Formen betyder, at præsentationen skal være kort og præcis. Den betyder også at man bliver klar over om strukturen i koden er i orden. Mit plugin består af fire filer: `asthmadiary.php` – her er de funktioner, der placerer resultater og indtastningsformularen i kontrolpanelet og på WP sider. `form.inc` – her ligger de HTML stumper, der skaber indtastningsformularen. `asthmadiary-result-presentation.php` – funktioner der tegner grafer eller fortolker peak flowværdierne som enten rød, gul eller grøn. `charts.php` – en række classes, der understøtter præsentationen. Her ligger den kode som tegner grafer og diagrammer via Google Charts API. Vi diskuterede om læger eller behandlere skulle have mulighed for at kommentere de enkelte indlæg; men Jmente heller ikke at lægerne ville bruge tid på at blogge med patienterne. Derfor fastholder jeg beslutningen om at astmadagbogen ikke skal være en blog, hvor folk kan svare tilbage.

Der kom også ideer til om koden kunne anvendes andre platforme end WP. I øvrigt viste Allans foredrag om AJAX at man kan gøre input persistent, selv om der måske ikke er kontakt til en server. Denne teknik kan udnyttes i forbindelse med mit WP plugin – for man kan ikke forvente, at astmapatienten har kontakt til servere altid. AJAX kan løse dette problem. Men teknologien understøttes ikke af alle browsere endnu.

7.4 Web analytics

Opgavebeskrivelsen nævner, at webanalytics kan indgå i besvarelsen. Jeg har integreret Google Analytics i websiden, og har samtidig adgang til webhotellets statistikker for multimusen.dk. Når en større mængde data er samlet bliver det sikkert interessant at undersøge brugeradfærden på sitet.

7.5 WordPress

WordPress findes i to udgaver. En bruger kan oprette enten oprette en gratis blog eller downloade kildekoden til brug på en server. Jeg valgte den sidste løsning.

7.6 Asthma Diary kort beskrivelse af plugin samt koden

"The Asthma Diary" er et plugin i PHP, der anvender en database. Dette plugin kan installeres på et WordPress system. Efter installationen får enhver bruger på WP-installationen mulighed for at anvende astmadagbogen fra kontrolpanelet.

En astmadagbog er et privat dokument. Derfor holder jeg astmadagbogen i kontrolpanelets mere private rum.

Koden kan downloades i en zip-fil: multimusen.dk/asthma-plugin - klik på "Download Plugin".

7.6.1 asthmadiary.php

Hovedfilen er asthmadiary.php. Filen indledes med en specielt formatteret kommentar, der fortæller WordPress, at der er tale om et plugin. Beskrivelsen kan læses i kontrolpanelet af administrator.

```
<?php
/*
Plugin Name: Asthma Diary
Plugin URI: http://multimusen.dk/
Description: If you suffer from asthma monitoring the symptoms on a regular basis is a
Version: 0.5
Author: Per Thykjaer Jensen
Author URI: http://multimusen.dk
License: &copy; Per Thykjaer Jensen - free to use by any asthma patient. For commercial
*/
/*
CONCERNING THE DATABASE
Checks whether the required table exists or no.
If the table isn't present a new table is created.
Source of inspiration: http://jacobschatz.com/?p=23 - however the versioncheck is dis
The code is basically the same for other database calls. See form.inc.
*/
check_asthmablog_database(); // Is the table present?
function check_asthmablog_database()
{
global $wpdb; // The WP database class
// $rq_db_version = 1.0;
// I don't use this versioning method, but it's a good idea though.
$table_name = "asthmablog";
if($wpdb->get_var("SHOW TABLES LIKE '$table_name'") != $table_name)
{
$sql = "CREATE TABLE " . $table_name . " (
ab_id INT(11) NOT NULL AUTO_INCREMENT,
ab_date DATE,
ab_note TEXT,
ab_name TEXT,
ab_peak flow INT(4),
ab_cough INT(1),
ab_wheeze INT(1),
ab_sleep INT(1),
```

```

ab_activity INT(1),
PRIMARY KEY id (ab_id)
);";
require_once(ABSPATH . 'wp-admin/includes/upgrade.php');
dbDelta($sql);
$insert = "INSERT INTO " . $table_name .
" (ab_note, ab_name, ab_peak flow, ab_cough, ab_wheeze, ab_sleep, ab_activity) " .
"VALUES ('test', 'test', '0', '0', '0', '0', '0')";
$results = $wpdb->query( $insert ); // test data inserted.
// add_option("rq_db_version", $rq_db_version);
}
//for updates
//$installed_ver = get_option( "cl_db_version" );
}
/*
CONCERNING THE SHORTCODE FOR THE PRESENTATION LAYOUT
The shortcode [asthmadiary] for pages etc.
*/
function ad_present(){
require_once('asthmadiary-result-presentation.php');
}
// add shortcode [asthmadiary] for use on pages or in templates
add_shortcode('asthmadiary', 'ad_present');
/*
CONCERNING THE DASHBOARD WIDGETS
The following outputs the contents of the dashboard widget.
*/
function dashboard_widget_function() {
// Introduction
echo "<div class='wrap'>";
// Form
require('form.inc');
echo "<br /><br />";
// call class
require_once('charts.php'); // import the chart class
$asthma_info = 'asthmadiary_charts'; // instantiate the class
// Interpretation
$asthma_info::ad_pef_interpretation();
// Results in charts
//echo "<h3>Charts</h3><p>A summary of the last 30 entries.</p>";
require_once('charts.php'); // import the chart class
$asthma_info = 'asthmadiary_charts'; // instantiate the class
$asthma_info::ad_peak_flow_chart(); // drawing the peak flow charts
$asthma_info::ad_show_column('ab_cough', 'Cough');
$asthma_info::ad_show_column('ab_wheeze', 'Wheeze');
$asthma_info::ad_show_column('ab_sleep', 'Sleep');
$asthma_info::ad_show_column('ab_activity', 'Activity');
echo "</div>";
}
// Function that being used in the action hook

```

```

function add_dashboard_widgets() {
    wp_add_dashboard_widget('dashboard_widget', 'Asthma Diary', 'dashboard_widget_function');
}
// Register the new dashboard widget into the 'wp_dashboard_setup' action
add_action('wp_dashboard_setup', 'add_dashboard_widgets' );
?>

```

7.6.2 charts.php

Charts.php fremstiller grafik og diagrammer samt fortolkningen af peak flow værdien.

```

<?php
class asthmadiary_charts {
    /*
    This class should draw the images and graphs for the Asthma Diary,
    and print interpretation of the daily peak flow value.
    It will also be able to echo the post entries.
    */
    public function ad_peak_flow_chart(){
        // This function will echo a comma separated list from the database
        global $wpdb; // wp database class
        global $current_user; // wp user info class
        $username = $current_user->user_login; // the string of the logged in user
        // sql string
        $sql = $wpdb->get_results( "SELECT * FROM 'asthmablog' WHERE ab_name='$username' ORDER BY ab_date DESC");
        $sql = array_reverse($sql, true); // reversed in order to have last result to the right
        // form array of peak flow results (better: ab_peak_flow could be an attribute)
        foreach($sql as $nummer){
            $resultat[] = $nummer->ab_peak_flow;
        }
        //return $resultat;
        $comma_result = implode(",", $resultat);
        // Google Graph API strings
        $img_start = "<img src='http://chart.apis.google.com/chart?chf=a,s,000000B4&chxr=0,0,1,1&chls=2,4,0&chma=5,5,5,5&chm=B,C5D4B5BB,0,0,0&chtt=peak flow' />";
        $img_end = "&chg=14.3,-1,1,1&chls=2,4,0&chma=5,5,5,5&chm=B,C5D4B5BB,0,0,0&chtt=peak flow' />";
        // Echo the graph to the browser
        echo "<div class='asthmadiary_peg'>";
        echo $img_start . $comma_result . $img_end; // here's the img tag
        echo "</div>";
    }
    /*
    DRAWING CHARTS
    Depending on the parameter this function will draw charts:
    cough, wheeze, sleep etc.
    */
    public function ad_show_column($column, $label){
        // This function will print a chart to the webpage
        // Intended for cough, wheeze, sleep, activity
        global $wpdb; // wp database class
    }
}

```

```

global $current_user; // wp user info class
$username = $current_user->user_login; // the string of the logged in user
// sql string, selecting only observations from logged in user
$sql = $wpdb->get_results( "SELECT * FROM 'asthmablog' WHERE ab_name='$username' ORDER BY ab_time DESC");
$sql = array_reverse($sql, true); // reversed in order to have last result to the right
// form array of peak flow results (better: ab_peak flow could be an attribute)
foreach($sql as $nummer){
$resultat[] = $nummer->$column;
}
//return $resultat;
$comma_result = implode(",", $resultat);
// Google Graph API strings
$img_start = "<img src='http://chart.apis.google.com/chart?chxr=0,0,5&chxt=y&chbh=a&chls=$label' width='300' height='225' alt=''" . $label . "' />";
// Echo the graph to the browser
echo "<div class='asthmadiary_pef'>";
echo $img_start . $comma_result . $img_end; // here's the img tag
echo "</div>";
}
/*
INDICATIONS OF peak flow MEASUREMENTS
This function will get the last peak flow result and compare to the highest value.
Then the result is evaluated as a green, yellow or red value.
*/
public function ad_pef_interpretation(){
global $wpdb; // wp database class
global $current_user; // wp user info class
$username = $current_user->user_login; // the string of the logged in user
// sql string
$sql = $wpdb->get_results( "SELECT * FROM 'asthmablog' WHERE ab_name='$username' ORDER BY ab_time DESC");
// form array of peak flow results (better: ab_peak flow could be an attribute)
foreach($sql as $nummer){
$resultat[] = $nummer->ab_peak flow;
}
$pefmax = max($resultat);
$pefnow = $resultat[0];
echo "<p><strong>peak flow Indications</strong><br />";
echo "Your Current peak flow: " . $pefnow . " It is a ";
// Interpretation
if ($pefnow >= ($pefmax * 0.8)) {
echo "<strong>Green peak flow Value</strong> - A peak flow reading in the green zone ";
}
if (($pefnow < ($pefmax * 0.79)) && ($pefnow >=($pefmax * 0.50))){
echo "<strong>Yellow peak flow Value</strong> - Indicates caution. It may mean respiratory distress ";
}
}

```

```

if ($pefnow < ($pefmax * 0.50)){
echo "<strong>Red peak flow Value</strong> - Indicates a medical emergency. Severe ai
}
echo " Your highest peak flow value is: " . $pefmax . "<br />";
echo "</p>";
}
/*
ECHO ALL POSTS OF A USER
This function will echo all the posts of the active user.
*/
public function ad_showall(){
global $wpdb; // wp database class
global $current_user; // wp user info class
$username = $current_user->user_login; // the string of the logged in user
// sql query
$sql = $wpdb->get_results( "SELECT * FROM 'asthmablog' WHERE ab_name='$username' ORDER
// the table
echo "<table style='margin-left: auto; margin-right: auto;'>
<caption> The Complete Asthma Diary of " . $current_user->user_firstname . " " . $curr
<th> Id </th>
<th> Date </th>
<th> Note </th>
<th> User </th>
<th> Pef </th>
<th> Cough </th>
<th> Wheeze </th>
<th> Sleep </th>
<th> Activity </th>
";
$t = "<td>"; // shorthand html tags
$te = "</td>";
foreach($sql as $nummer){
echo "<tr>"; // new row
echo $t . $nummer->ab_id . $te; // echoing the content as table cells
echo $t . $nummer->ab_date . $te;
echo $t . $nummer->ab_note . $te;
echo $t . $nummer->ab_name . $te;
echo $t . $nummer->ab_peak flow . $te;
echo $t . $nummer->ab_cough . $te;
echo $t . $nummer->ab_wheeze . $te;
echo $t . $nummer->ab_sleep . $te;
echo $t . $nummer->ab_activity . $te;
echo "</tr>"; // end row
}
echo "</table>";
}
} // class end
?>

```

7.6.3 asthma-diary-result-presentation.php

Denne fil skriver resultaterne til browseren. Den minder lidt om view i en MVC-datamodel.

```
<?php
require_once('charts.php'); // import the chart class
$asthma_info = 'asthmadiary_charts'; // instantiate the class
// set var and global
global $current_user;
get_currentuserinfo();
// test whether user is logged in
if ( is_user_logged_in() ){
echo "<p>The Asthma Diary of : ";
echo $current_user->user_firstname . ' ' . $current_user->user_lastname . ".<br />";
echo "Date: " . date('d-m-Y H:i') . "</p>";
/*
PRESENTATION OF OBSERVATIONS
*/
// echo "<h3>Summaries</h3><p>The graphs below consists of the 30 newest posts (or less)
// includes
$asthma_info::ad_peak_flow_chart(); // drawing the peak flow charts
$asthma_info::ad_pef_interpretation(); // peak flow interpretation
echo "<p style='font-size: x-small;'><strong>Caution:</strong> the interpretation of the
peak flow values should be read with care.
The green value is your best peak flow value in the database. Therefore the first entry
will always be green. After a while however the peak flow interpretations will become
more precise. </p>";
$asthma_info::ad_show_column('ab_cough', 'Cough');
$asthma_info::ad_show_column('ab_wheeze', 'Wheeze');
$asthma_info::ad_show_column('ab_sleep', 'Sleep');
$asthma_info::ad_show_column('ab_activity', 'Activity');
echo "<div style='font-size: x-small;'><br /><br />";
echo "<hr />";
$asthma_info::ad_showall();
echo "<hr />'The Asthma Diary' - &copy; Per Thykjaer Jensen, 2011. All rights reserved";
echo "</div>";
}
else {
echo "<p>This page is only for registered users. <br />Users can log in <a href='http";
}
?>
```

7.6.4 form.inc

I form.inc skaber indtastningsformularen.

```
<?php
function lavRadio($astma, $n){ // creates radio buttons
$t = 1;
echo $astma."<br />1";
```

```

while ($n > 0) {
print "<input type='radio' name='$astma' value='$t' />\n";
$t = $t +1;
$n = $n - 1;
}
echo " 5<br />";
}
// use it like this: lavRadio('Hoste', 5);
// testing form logic
if ($_POST['submit'])
{
// logic to be performed after clicking submit
new_asthma_observation();
}
else {
echo "<h3>New Asthma Diary Entry</h3><p>Please enter a note, your observations and pe
}
// this function will add observations from the form to the database
function new_asthma_observation(){
// database operations
global $wpdb; // The WP database controller class
$table_name = "asthmablog"; // name of table
$valuestring = "'".$_POST['Note'] . "','" . $_POST['peak flow'] . "','" . $_POST['Co
$insert = "INSERT INTO " . $table_name .
" (ab_note, ab_peak flow, ab_cough, ab_wheeze, ab_sleep, ab_activity, ab_name, ab_date
"VALUES (" . $valuestring .)";
$results = $wpdb->query($wpdb->prepare( $insert )); // don't forget to execute the que
// prepare is the sanitizing function in order to avoid SQL inject etc.
echo "The Asthma Diary is updated.";
}
function asthmablog_username() {
// prints username
// user informations prepared
global $current_user;
get_currentuserinfo();
echo $current_user->user_login;
/*
Samples from Codex:
echo 'Username: ' . $current_user->user_login . "\n";
echo 'User email: ' . $current_user->user_email . "\n";
echo 'User first name: ' . $current_user->user_firstname . "\n";
echo 'User last name: ' . $current_user->user_lastname . "\n";
echo 'User display name: ' . $current_user->display_name . "\n";
echo 'User ID: ' . $current_user->ID . "\n";
*/
}
?>
<!-- FORM Input to asthma diary -->
<form id="asthmablog" action="" method="post" enctype="multipart/form-data">
<input type="hidden" name="message" value="OK - the Asthma Diary is updated." />

```

```



```

8 Kildefortegnelse

8.1 Litteratur

- [**Bartoz**] Bartoz, Porebski et al: "Building PHP Applications with Symfony, CakePHP and Zend Framework" (Wiley Publishing 2011)
- [**Brazell**] Brazel, Aaron et al.: "The Wordpress Bible" (Wiley 2010)
- [**Clement**] Clement, Andrew et al.: "Participatory Design and Web 2.0: The case of PIPWatch, the collaborative privacy toolbar" (Proceedings of the Tenth Anniversary Conference on Participatory Design 2008)
- [**DataCollection**] Chapter 2 "Data Collection - Importance and Options" (kilden er ikke angivet i kursusmaterialet)
- [**Greenbaum**] Greenbaum, Joan: "PD a Personal Statement" (Communications of the ACM, 1993 Vol. 36 Nr. 4:47)
- [**Jepsen**] Jepsen, Leif O. et al.: "Back to thinking mode: diaries for the management of information system development projects" (Behaviour and Information Technology, 1989, vol. 8, no. 3, 207 ff.).
- [**Nielsen**] Nielsen, Jakob: "Godt Webdesign" (2001)
- [**Obendorf**] Obendorf, Hartmut et al.: "Inter-Contextual Distributed Participatory Design Communicating design philosophy and enriching user experience" (The Berkeley Electronic Press, 2009)
- [**Schneiderman**] Schneiderman: "Designing the User Interface" (2005) Ch. 3, 4, 11.
- [**Tognazzini**] Tognazzini: "Tog on Interface" (1992) Ch. 14 "User Testing on the Cheap".

8.2 Internet

Herunder findes links til de vigtigste sider i forbindelse med dette projekt. En større samling findes på <http://multimusen.dk/links-2/>. askaboutphp.com/26/creating-google-graphs.html - Tutorial i Google Charts API.

book.cakephp.org/ - CakePHPs "kogebog" med vejog vildledninger.

code.google.com/intl/da/apis/chart/ - Google Charts API.

codex.wordpress.org/ - WordPress manualen.

codex.wordpress.org/Database_Description - WordPress, beskrivelse af databasen.

google.com/analytics/ - Google Analytics.

inqbation.com/the-wordpress-data-model/ - WordPress datamodel. **multimusen.dk** - Mit net-laboratorium med noter og opgaver fra masterfagpakkerne.

multimusen.dk/cake/ - Iteration II eller første udkast til astmadagbogen programmeret i frameworket CakePHP.

multimusen.dk/wau-iii/ - Iteration III Dagbog, noter og observationer fra "Websitestrukturering og implementering" (Webapplikationsudvikling Modul III).

multimusen.dk/the-asthma-diary-plugin/ - Produktsiden. Her findes hjælpetekster, download af plugin og mulighed for at kontakte forfatteren.

mini.egovplus.dk/hmwiki/doku.php?id=metarefektoriet&zs=thykk%C3%A6r - "Metarefektoriet", personlig wiki med dagbogsoptegninger fra modul II.

superpinch.dk/webapp/assignments.php - Michael Hansons side - her findes referater af vores gensidige evalueringer.

webapplikationsudvikling.wordpress.com/ - Jan Skovgårds webside.

tuxradar.com/practicalphp - Introduktion til PHP.

w3schools.com - fremragende samling af tutorials og kodeeksempler.