



Hi Freddy, die Anmerkungen für Dich sind hellrot hinterlegt und nicht Bestandteil der Ausarbeitung. Vielen Dank für das Lesen meiner Arbeit!

Assistenzplaner

Serverbasierte Software zur Koordinierung der Assistenten für behinderte Menschen

Masterarbeit

im Studiengang Software Engineering und Informationstechnik



TECHNISCHE HOCHSCHULE NÜRNBERG
GEORG SIMON OHM

vorgelegt von: Ulrich Belitz

Fakultät: Elektrotechnik Feinwerktechnik Informationstechnik

Matrikelnummer: 2464776

Erstgutachter: Prof. Dr. Thomas Mahr

Zweitgutachter: Prof. Dr. Ralph Lano

© 2014

Dieses Werk einschließlich seiner Teile ist **urheberrechtlich geschützt**. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtgesetzes ist ohne Zustimmung des Autors unzulässig und strafbar. Das gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen sowie die Einspeicherung und Verarbeitung in elektronischen Systemen.



Eidesstattliche Erklärung

Ich, Ulrich Belitz, Matrikel-Nr. 2464776, versichere hiermit, dass ich meine Masterarbeit mit dem Thema

Assistenzplaner - Serverbasierte Software zur Koordinierung der Assistenten für behinderte Menschen

selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, wobei ich alle wörtlichen und sinngemäßen Zitate als solche gekennzeichnet habe. Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Mir ist bekannt, dass ich meine Masterarbeit zusammen mit dieser Erklärung fristgemäß nach Vergabe des Themas in zweifacher Ausfertigung und gebunden im Studienbüro der Technischen Hochschule Nürnberg Georg Simon Ohm abzugeben oder spätestens mit dem Poststempel des Tages, an dem die Frist abläuft, zu senden habe.

Taunusstein, den 2. September 2014

ULRICH BELITZ



Abstract

Gut das ich einen amerikanischen Schwip-Schwager habe, der mir neulich ins Garagentor gefahren ist und mir noch einen Gefallen schuldet :-D Er wird die Übersetzung hoffentlich machen.



Kurzfassung

Um einen körperlich behinderten Menschen (Klienten) und sein Team (bestehend aus Assistenten) bei der Dienstplanerstellung zu unterstützen wurde die serverbasierte Anwendung **Assistenzplaner**¹ geschaffen.

Mit dem Assistenzplaner ist es unter anderem möglich:

- Assistenz-Teams zu verwalten
- automatisiert Dienstpläne zu erstellen
- Aufgaben der Assistenten zu verwalten

Hey Buddy, hier bin ich besonders auf Dein Feedback angewiesen - ich bin so tief in dem Thema drin, dass ich das Gefühl habe gar nicht mehr zusammenfassend sagen zu können, was der Assistenzplaner ist.

¹<http://assistenplaner.de/>



Danksagung

Ich danke Herrn Prof. Dr. Mahr für die Betreuung und Korrektur dieser Masterarbeit. Weiterhin danke ich Herrn Prof. Dr. Lano für die Zweitkorrektur.

Herzlicher Dank geht an Patrick! Ohne ihn wäre diese Arbeit nicht entstanden. Er hat maßgeblich die Anforderungen gestellt und bereitwillig mit seinem Team die Software getestet.

Ich danke sehr Frederik für die vielen Skype-Gespräche, das gegenseitige Motivieren und die zahlreichen Korrekturanregungen.

Ich hoffe damit positive Motivation aufgebaut zu haben, viele Korrekturanregungen zu geben! ;-)

Meinen Eltern danke ich sehr herzlich für alles, was sie mir ermöglicht und auf den Weg gegeben haben!

Unendlich dankbar bin ich meiner Frau Marielle! Sie hat mich während des gesamten Studiums verständnisvoll unterstützt und mir den Rücken frei gehalten.



Inhaltsverzeichnis

Eidesstattliche Erklärung	3
Abstract	4
Kurzfassung	5
Danksagung	6
1. Motivation	9
2. Grundlagen	10
2.1. Netzwerke und Internet	10
2.2. Web-Technologien	13
2.3. Modellierungssprachen	18
3. Anforderungsermittlung	20
3.1. Begrifflichkeiten	20
3.2. Ist-Prozess	21
3.3. Soll-Prozess	22
3.4. Benutzergeschichten	22
3.5. Abgeleitete Anforderungen	29
4. Marktanalyse	31
4.1. Dienstplan in Excel	31
4.2. Offline Anwendungen	31
4.3. Online Anwendungen	32
5. Software-Design	34
5.1. SysML Anforderungsdiagramm	34
5.2. UML Komponentendiagramm	35
5.3. UML Klassendiagramme	35
5.4. UML Sequenzdiagramme	35
6. Implementierung	37
6.1. Wahl der Programmiersprachen	37
6.2. Persistenz der Daten	38



6.3. Benutzer-Verwaltung	38
6.4. Willkommen-Seite	39
6.5. Navigation	40
6.6. Anlegen eines neuen Klienten-Zugangs	40
6.7. Übersicht	41
6.8. Einstellungen	41
6.9. Standard-Dienstzeiten	42
6.10. Team-Verwaltung	42
6.11. Monatsplan	43
6.12. Kalender	44
6.13. Dienst-Plan	46
6.14. Entwicklung des Algorithmus zur Dienstplanerstellung	48
6.15. Aufgaben-Verwaltung	54
6.16. Dokumentation	57
6.17. Impressum	57
7. Überführung in Produktiv-Betrieb	58
8. Ergebnis	59
9. Fazit und Ausblick	60
9.1. Mehr Teams	60
9.2. Mögliche Verbesserungen	61
9.3. Mögliche Erweiterungen	65
10. Literaturverzeichnis	68
A. Tabellenverzeichnis	70
B. Abbildungsverzeichnis	71
C. Verzeichnis der Listings	72
D. Abkürzungsverzeichnis	73
E. Inhalt der beiliegenden CD	74
F. Hintergrundinformationen	75
F.1. Verzeichnisstruktur auf Server	75
F.2. Verwendete Software	75



1. Motivation

Ein Bekannter des Autors ist ein körperlich behinderter Mensch (im folgenden nur noch „Klient“ genannt), der auf Assistenz angewiesen ist. Der Klient hat sieben Assistenten. Bisher hat der Klient den Dienstplan jeden Monat manuell erstellt. Der Klient hat gleichzeitig die sieben E-Mails seiner Assistenten überblickt (in denen möglichen Arbeits-Termine enthalten waren) und mühevoll einen möglichst ausgewogenen Dienstplan erstellt. Der Dienstplan sollte die Verfügbarkeiten und die Stundenkontingente der Assistenten möglichst gut berücksichtigen. Das Erstellen des Dienstplans hat den Klienten ca. vier Stunden pro Monat gekostet.

Um den Klienten zu entlasten und technisch besser zu unterstützen, wurde die Idee des Assistenzplaners ins Leben gerufen. Der Assistenzplaner soll als webbasierte Anwendung den Klienten und sein Team bei der Dienstplanerstellung unterstützen. Der Klient und alle Assistenten bekommen einen personalisierten Zugang zum Assistenzplaner und können alle nötigen Informationen austauschen. Ein Algorithmus erstellt den Dienstplan unter Berücksichtigung aller Vorlieben des Klienten und der Stundenkontingente der Assistenten. Der Zeitaufwand für die Dienstplanerstellung sollte auf wenige Minuten sinken.



2. Grundlagen

In diesem Kapitel werden die technischen Grundlagen für das Verständnis der vorliegenden Arbeit gelegt.

2.1. Netzwerke und Internet

Client/Server-Paradigma

„Netzwerkanwendungen arbeiten in einer Kommunikationsform, die man Client/Server-Paradigma nennt. Die Serveranwendung wartet passiv auf Kontaktaufnahme, während die Clientanwendung die Kommunikation aktiv einleitet.“ [Comer 2002, S. 405]

Clientsoftware

Nach [Comer 2002, S. 406] hat die Clientsoftware unter anderem folgende Merkmale:

- Sie „ist ein Anwendungsprogramm, die vorübergehend zum Client wird, wenn entfernter Zugriff notwendig ist, das lokal aber auch andere Aufgaben ausführt.“
- Sie „läuft lokal auf dem Personalcomputer eines Benutzers“
- Sie „leitet den Kontakt mit einem Server aktiv ein.“
- Sie „benötigt keine spezielle Hardware oder ein besonderes Betriebssystem.“

Serversoftware

Nach [Comer 2002, S. 406] hat die Serversoftware unter anderem folgende Merkmale:

- Sie „ist ein dediziertes Programm mit dem speziellen Zweck, einen Dienst bereitzustellen, kann aber auch gleichzeitig mehrere entfernte Clients bedienen“.



2. Grundlagen

- Sie „läuft auf einem gemeinsam genutzten Computer.“
- Sie „wartet passiv auf die Verbindungsaufnahme durch einen entfernten Client.“
- Sie „nimmt den Kommunikationswunsch von beliebigen Clients entgegen, bietet aber nur einen Dienst.“
- Sie „setzt leistungsstarke Hardware und ein gehobenes Betriebssystem voraus.“

URL

Uniform Resource Locator (URL) ist ein Syntaxformat, das alle Informationen kodiert, die zum Erreichen einer Webseite notwendig sind. Die URL hat folgendes Format:

protokoll://domainNameComputer:port/dokumentName

Die Angabe eines Ports ist optional.

Ein Beispiel für eine URL ist die Startseite des Assistenzplaners:

<http://assistenplaner.de/index.html>

Browser

„Ein Browser ist ein interaktives Programm, mit dem ein Benutzer Informationen aus dem Web lesen kann. Die Informationen enthalten wählbare Elemente, über die der Benutzer weitere Informationen anzeigen lassen kann.“ [Comer 2002, S. 489]

Link

Link ist der umgangssprachliche Begriff für eine Hypertext-Referenz. „Im Browser ist eine Hypertext-Referenz in einem angezeigten Dokument zunächst ein wählbares Element. Wählt der Benutzer das Element, folgt der Browser der Referenz, holt das betreffende Dokument und ersetzt die momentane Anzeige durch das neue Dokument.“ [Comer 2002, S. 496]



HTTP

„Die Interaktion zwischen einem Browser und einem Webserver erfolgt über das *Hypertext Transport Protocol (HTTP)*. HTTP erlaubt es dem Browser, ein bestimmtes Element anzufordern, das der Server ausgibt. In der Praxis ist HTTP aber komplexer, weil ein Server zusätzliche Statusinformationen mit jedem Transfer sendet und das Protokoll es einem Browser gestattet, Informationen zu senden und anzufordern.“ [Comer 2002, S. 497].

„HTTP unterstützt vier Basisoperationen, die ein Browser in einer Anfrage spezifizieren kann:

- *GET* verlangt ein spezifisches Element vom Server. Der Server gibt einen Kopf (HEAD) zurück, der Statusinformation, gefolgt von einer Leerzeile und dem Element, enthält.
- *HEAD* verlangt Statusinformationen über ein Element. Der Server gibt den Status zurück, jedoch ohne Kopie des Elements selbst.
- *POST* sendet Daten an den Server. Der Server hängt die Daten an ein spezifisches Element an (z. B. wird eine Nachricht an eine Nachrichtenliste angehängt).
- *PUT* sendet Daten an den Server. Der Server benutzt die Daten, um ein spezifisches Element zu ersetzen.

Ein Browser erzeugt eine HTTP-Anfrage, wenn ein Benutzer eine URL eingibt oder einen Link wählt. In beiden Fällen sendet der Browser eine GET-Anfrage, die ein Element spezifiziert, und der Server gibt ihm das angefragte Element zurück.“ [Comer 2002, S. 497]

Webdokumente

Es gibt drei verschiedene Typen von Webdokumenten [Comer 2002, S. 508]:

- **Statisch:** Das Dokument ist fest in einer Datei definiert. Der Autor bestimmt einmalig bei Erstellung den Inhalt. Jede Anfrage an das statische Dokument führt zeitlich unabhängig zur identischen Ausgabe. Das statische Verhalten kann technisch mit starren HTML-Seiten realisiert werden.
- **Dynamisch:** Das Dokument ist nicht fest definiert. Sobald eine Anfrage an den Webserver gestellt wird, wird das dynamische Webdokument zusammengestellt. Der Inhalt des dynamischen Dokumentes kann sich von Anfrage zu



2. Grundlagen

Anfrage ändern. Das dynamische Verhalten kann mit Server-Script-Sprachen wie z. B. PHP realisiert werden.

- Aktiv: Das aktive Dokument wird vom Client erstellt. Der Server liefert lediglich das Programm zur Erstellung des Dokumentes aus. Die konkrete Struktur des Dokumentes wird im Client zusammengebaut. Die benötigten Inhalte werden vom Client beim Server angefragt. Das aktive Verhalten kann mit Client-Script-Sprachen wie z. B. JavaScript realisiert werden.

Cookie

„Wenn ein Server umfangreiche Statusinformationen speichern muss, benötigt er die Informationen auf einer lokalen Platte. Statusinformationen werden einem Browser in Form eines *Cookies* weitergegeben. Jedes Cookie besteht aus zwei Zeichenketten, die man als *Name/Wert-Paar* bezeichnet. Im Teil *Name* befindet sich der Name der Website und im Teil *Wert* eine kleine Zeichenkette, die der Browser speichert. Wenn der Browser die Website wieder kontaktiert, fügt er das Cookie in die Anfrage ein. Aus Sicht eines Servers hat es also den Anschein, dass der Browser Statusinformationen speichern und zurückgeben kann.“ [Comer 2002, S. 514f]

2.2. Web-Technologien

XML

Extensible Markup Language (**XML**) ist eine Auszeichnungssprache. Mit XML ist es möglich hierarchisch strukturierter Daten darzustellen. Mit den textbasierten XML-Dateien ist es leicht möglich Daten zwischen Systemen auszutauschen. Die XML-Dateien sind für den Menschen gut lesbar. Die Syntax ist einfach gehalten. Es gibt Tags und Attribute. Die Tags können ineinander geschachtelt werden. Ein beispielhaftes XML Dokument findet sich in Listing 2.1. Hier gibt es ein Wurzelknoten **AssitanceTeam** mit zwei Kinderknoten **Assistant** die jeweils das Attribut **isAvailable** mit unterschiedlichen Werten haben. Die Kinderknoten (**Assistant**) haben wiederum eigene Kinderknoten (**Name** und **eMailAddress**). Die Schachtelung der Knoten kann beliebig tief erfolgen.

Listing 2.1: Beispiel für ein XML Dokument.

```
1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <AssitanceTeam>
3   <Assistant isAvailable="true">
4     <Name>Max Mustermann</Name>
```



2. Grundlagen

```
5      <EmailAddress>max@mustermann.de</EmailAddress>
6    </Assistant>
7    <Assistant isAvailable="false">
8      <Name>Markus Mustermann</Name>
9      <EmailAddress>markus@mustermann.de</EmailAddress>
10   </Assistant>
11 </AssitanceTeam>
```

HTML

HyperText Markup Language (**HTML**) ist eine Auszeichnungssprache und hat „die Aufgabe, die logische Grob- und Feinstruktur einer Webseite zu beschreiben. ... HTML ist die Sprache zur Strukturierung von Texten, wobei aber auch die Möglichkeit besteht, Grafiken und multimediale Inhalte in Form einer Referenz einzubinden und in den Text zu integrieren“ [Münz und Gull 2012, S. 21].

Es ist unter anderem möglich [Münz und Gull 2012, S. 21]:

- das Dokument in Bereiche wie Kopf, Inhalt, Fuß, Navigation und Artikel einteilen
- dem Dokument Überschriften, Textabsätze, Listen und Tabellen hinzuzufügen
- Links (siehe Abschnitt 2.1) auf beliebige andere Webseiten oder Datenquellen im Internet einzubinden
- Formulare in das Dokument zu integrieren
- Erweiterungssprachen wie CSS oder JavaScript zu integrieren

Ein Beispiel für ein HTML Dokument findet sich in Listing 2.2. Man kann gut erkennen, dass HTML dieselbe Struktur wie XML hat.

Listing 2.2: Beispiel für ein HTML-Dokument.

```
1 <!DOCTYPE html>
2 <meta charset="utf-8">
3 <html>
4 <head>
5   <title>Assistenzplaner</title>
6   <link rel="stylesheet" type="text/css" href="../CSS/example.css" media="
    all"/>
7   <script language="JavaScript" src="../JavaScript/example.js"></script>
8 </head>
9 <body onload="init()">
10 <h1>Aktuelle Zeit</h1>
```



```
11 <span class="time" id="createContentWithJavaScript"></span>
12 </body>
13 </html>
```

CSS

Bei *Cascading Stylesheets* (CSS) „handelt [es] sich um eine beschreibende Ergänzungssprache für HTML. Sie klinkt sich nahtlos in HTML ein und hat zwei Aufgaben: das Formatieren von Inhalten und das Gestalten von Webseitenlayouts.“ [Münz und Gull 2012, S. 23]

„CSS erlaubt es, zentrale Formate zu definieren, beispielsweise für alle Überschriften erster Ordnung oder für alle Textabsätze mit einem bestimmten Klassennamen oder für hervorgehobenen Text, der innerhalb einer Tabellenzelle vorkommt. Die zentralen Formate können in eine externe Style-Datei ausgelagert werden[.] ... So ermöglicht CSS seitenübergreifend einheitliche Layouts und Formatierungen.“ [Münz und Gull 2012, S. 24]

Listing 2.3: Beispiel für ein CSS-Dokument.

```
1 /* set global font for all elements */
2 html * {
3     font-family: Verdana;
4 }
5
6 /* change font color of all headings */
7 h1, h2, h3, h4, h5, h6 {
8     color: #0000ff;
9 }
10
11 /* define attributes of all elements with class "time" */
12 .time {
13     border: 1px solid black;
14     padding: 10px;
15 }
```

JavaScript

JavaScript ist eine Script-Sprache, die meistens im Browser (also clientseitig) ausgeführt wird. Mögliche Einsatzgebiete von JavaScript sind unter anderem:

- Plausibilitätsprüfung (Validierung) von Formulareingaben vor dem Absenden



2. Grundlagen

- Dynamische Manipulation von Webseiten mit Hilfe von DOM (siehe Abschnitt 2.2)
- Senden und Empfangen von Daten, ohne dass der Browser die Seite neu laden muss mit AJAX (siehe Abschnitt 2.2)

Ein Beispiel für ein JavaScript-Dokument findet sich in Listing 2.4.

Listing 2.4: Beispiel für ein JavaScript-Dokument.

```
1 function init() {  
2     var date = new Date();  
3  
4     var element = window.document.getElementById("  
5         createContentWithJavaScript");  
6     element.textContent = date.toLocaleString();  
7 }
```

Bei JavaScript sind einige Objekte jederzeit verfügbar (z. B. `RegExp`, `Date`, `window`). Diese Objekte lassen sich auch zur Laufzeit noch um Funktionalität erweitern. Dies geschieht über den Prototyping-Mechanismus. Ein Beispiel findet sich in Listing 2.5. Hier wird das bestehende `Date`-Objekt um die Funktion `addYears(number)` erweitert.

Listing 2.5: Beispiel für die Erweiterung eines bestehenden Objekts in JavaScript

```
1 Date.prototype.addYears = function (number) {  
2     var year = this.getFullYear();  
3     year += parseInt(number);  
4     this.setFullYear(year);  
5 }
```

Zusammenarbeit HTML, CSS und JavaScript

HTML, CSS und JavaScript arbeiten ergänzen sich gegenseitig. Wenn man der Idee von [Teague 2011, S. 8] folgt, dann ergeben die drei Webtechnologien eine vollständige Sprache: HTML stellt die Substantive zur Verfügung, CSS die Adjektive und Adverbien und JavaScript fügt die Verben hinzu.

Wenn man sich den HTML-Code von Listing 2.2 im Browser anschaut (siehe Abbildung 2.1), dann ist zu erkennen, wie die Zusammenarbeit der drei Webtechnologien sich in der Praxis auswirkt. Im HTML wird die Struktur des Dokumentes definiert, mit dem verwendeten CSS-Dokument (Listing 2.3) wird das Aussehen angepasst und mit JavaScript (Listing 2.4) wird dynamisch Inhalt erzeugt.



Aktuelle Zeit

17.8.2014 13:35:09

Abbildung 2.1.: Das Ergebnis des Zusammenspiels von HTML, CSS und JavaScript im Browser dargestellt

DOM

Document Object Model (**DOM**) ist eine Programmierschnittstelle für den Zugriff auf HTML- oder XML-Dokumente. „HTML-Dokumente besitzen eine hierarchische Struktur, die im DOM als Baumstruktur repräsentiert wird. Die Knoten des Baums stellen die verschiedenen Arten von Inhalt in einem Dokument dar.“ [Flanagan 2002, S. 310]. DOM stellt unter anderem einen Satz von Methoden zur Verfügung, die es möglich macht, Elemente des Dokumentes aufzufinden. Z. B. kann man mit Hilfe von

```
var captions = window.document.getElementsByTagName("h1");
```

alle Überschriften erster Ordnung finden.

Weiterhin stellt DOM Methoden zur Verfügung, um Inhalte des Dokument-Baums zu verändern. Mit dem Code

```
captions[0].setAttribute("id", "firstCaption");
```

wird der *Identifikator* (**ID**) der ersten Überschrift (erster Ordnung) gesetzt.

AJAX

Bei *Asynchronous JavaScript and XML* (**AJAX**) handelt es sich nicht direkt um eine Webtechnologie, eher um eine spezielle Denkweise, die Kombination von HTML bzw. XML, CSS und JavaScript einzusetzen.

„Mit Ajax können einzelne Informationen nach Bedarf vom Server geholt und in die bestehende Seite integriert werden. Im besten Fall entfallen die Wartezeiten, und die Kommunikation mit dem Server tritt aus der Sicht des Anwenders in den Hintergrund. Ajax-Applikationen ermöglichen so eine Bedienung, die eher an eine Desktop-Anwendung als an eine Webapplikation erinnert.“ [Koch 2011, S. 309]

Ein spezielles JavaScript-Objekt, `XMLHttpRequest`, ermöglicht die asynchrone Kommunikation mit dem Server. Durch den asynchronen Aufruf kann der Benutzer direkt weiterarbeiten und wird nicht durch die Kommunikation mit dem Server



2. Grundlagen

blockiert. Ist die Antwort des Servers beim Client angekommen, wird sie sofort verarbeitet und gegebenenfalls dem Nutzer dargestellt.

Mit Hilfe von AJAX ist es möglich aktive Dokumente (siehe Abschnitt 2.1) zu erstellen.

PHP

PHP Hypertext Preprocessor (**PHP**) ist eine Programmiersprache, die meistens auf dem Server ausgeführt wird. Mit ihr ist es möglich Internetanwendungen zu erstellen. Dabei erzeugt PHP die HTML-Seiten die im Browser angezeigt werden. Mit PHP kann man objektorientiert programmieren. Ein einfaches PHP-Script ist in Listing 2.6 gezeigt.

Listing 2.6: Beispiel für ein PHP-Dokument.

```
1 <?php session_start(); ?>
2 <!DOCTYPE html>
3 <meta charset="utf-8">
4 <html>
5 <head>
6     <title>Assistenzplaner</title>
7     <link rel="stylesheet" type="text/css" href="../CSS/global.css" media="
      all"/>
8 </head>
9 <body>
10 <?php include('navigation.php'); ?>
11
12 <? echo '<h1>Willkommen beim Assistenzplaner</h1>'; ?>
13 </body>
14 </html>
```

2.3. Modellierungssprachen

Modellierungssprachen dienen der Modellierung von Software. Mit Hilfe von Modellierungssprachen kann man komplexe Systeme konstruieren. Durch mehrere Abstraktionsebenen behält man die Übersicht. Modellierungssprachen geben einem Software-Architekten die Möglichkeit sich sehr konkret auszudrücken, ohne Code schreiben zu müssen.



UML

Die *Unified Modeling Language* (**UML**) „dient zur Modellierung, Dokumentation, Spezifizierung und Visualisierung komplexer Systeme, unabhängig von deren Fach- und Realisierungsgebiet. Sie liefert die Notationselemente gleichermaßen für die statischen und dynamischen Modelle von Analyse, Design und Architektur und unterstützt insbesondere objektorientierte Vorgehensweisen.“ [Rupp u. a. 2012, S. 4]

Evt. werde ich noch näher auf die Diagrammtypen eingehen, die ich verwendet habe - das werde ich entscheiden, wenn ich mit der Diagramm-Erstellung fertig bin. Was meinst Du zu dem Thema?

SysML

System Modeling Language (**SysML**) ist eine grafische Sprache die sich zur Modellierung von Systemen einsetzen lässt. Sie wird bei der Modellierung von komplexen Systemen eingesetzt. Einige Diagrammtypen aus der UML wurden übernommen und weitere Diagrammtypen hinzugefügt. Mit SysML lassen sich Systeme modellieren, die aus Hard- und Software bestehen.

Hier wollte ich evt. noch das Anforderungsdiagramm und die verwendeten Beziehungen zwischen den Anforderungen (containment, satisfy, ...) beschreiben, oder geht das dann zu tief?



3. Anforderungsermittlung

In diesem Kapitel werden die Anforderungen dargestellt. Neben den konkreten Benutzergeschichten, die von den zukünftigen Benutzern formuliert wurden, werden auch nicht explizit erwähnte Anforderungen festgehalten.

3.1. Begrifflichkeiten

In diesem Abschnitt werden einige Begriffe erläutert, die im Folgenden häufig verwendet werden.

Klient

Der Klient ist ein körperlich behinderter Mensch, der im alltäglichen Leben permanent auf Hilfe angewiesen ist. Im Rahmen der individuellen Schwerstbehindertenbetreuung bekommt er Assistenten an die Seite gestellt. Mit Hilfe seiner Assistenten kann er ein selbstbestimmtes Leben führen.

Assistent

Der Assistent ist ein Helfer, der den Klienten pflegt und ihn im alltäglichen Leben unterstützt.

Team

Der Klient muss rund um die Uhr mit Assistenz unterstützt werden. Dafür bedarf es eines Teams. Das Team eines Klienten besteht aus mehreren Assistenten. In dem konkreten Fall sind es sieben Assistenten.



Dienst

Wenn ein Assistent Dienst hat, dann ist er permanent in der Nähe des Klienten und steht ihm helfend zur Verfügung. Für den Leistungsnachweis ist eine Berechnung der Dienststunden notwendig. Wenn ein Assistent einen Dienst von 13:00 Uhr des Tages bis 8:00 Uhr des Folgetages hat, dann sind das rechnerisch 17 Stunden die er gearbeitet hat. Die Nacht wird mit 6 Stunden berechnet und abgezogen. So bekommt der Assistent eine Netto-Arbeitszeit von 11 Stunden vergütet.

Bereitschaft

Wenn ein Assistent Bereitschaft hat, dann muss er zu einer festgelegten Zeit (z. B. von 10:00 bis 11:00 Uhr) telefonisch erreichbar sein. Falls der diensthabende Assistent erkrankt, springt der Bereitschaftshabende ein und übernimmt den Dienst. Bei Diensten deren Netto-Arbeitszeit größer als 13 Stunden ist, wird eine zweite Bereitschaftsstunde (z. B. von 18:00 bis 19:00 Uhr) vereinbart, falls der diensthabende Assistent während der Arbeitsausführung erkrankt. Jede Bereitschaftsstunde wird mit dem Faktor 0.5 vergütet.

Dienstplan

Im Dienstplan sind die Dienst- und Bereitschaftszeiten der Assistenten festgehalten. Für jeden Tag gibt es einen Assistenten der Dienst hat und einen Assistenten der Bereitschaft hat.

Stundenkontingent

Jeder Assistent arbeitet unterschiedlich viel. Die Monatsstundenzahl variiert zwischen 30 und 130. Bei der Dienstplanerstellung ist darauf zu achten, dass alle Assistenten möglichst gleichmäßig ihr Stundenkontingent ausgeschöpft bekommen.

3.2. Ist-Prozess

Vor Beginn der Arbeiten des Autors war der Prozess zur Dienstplanerstellung komplett manuell. Der Klient hat für die Dienstplanerstellung eine E-Mail mit Hinweisen für den Monat an sein Team geschickt und darum gebeten Rückmeldung zu geben, wie die Verfügbarkeiten sind. Die Assistenten haben auf diese E-Mail geantwortet und darin beschrieben wann sie Zeit für Dienste haben. Der Klient hat manuell



den Dienstplan erstellt und musste dabei ständig zwischen seiner Tabelle und allen Antworten des Teams (sieben an der Zahl) wechseln. Nach der Fertigstellung des Dienstplans hat der Klient eine E-Mail mit dem Dienstplan an sein Team geschickt. Die Dienstplanerstellung hat den Klienten viel Zeit (ca. vier bis fünf Stunden im Monat) und Nerven gekostet.

3.3. Soll-Prozess

Ein möglicher Soll-Prozess könnte folgendermaßen aussehen: Der Klient gibt auf einer Webseite die Dienstzeiten und Kommentare für den kommenden Monat ein. Per Knopfdruck wird das Team benachrichtigt und darum gebeten die freien Tage auf einer Webseite (Link ist in der Nachricht enthalten) einzugeben. Sobald alle Assistenten ihre Eingaben getätigt haben können dem Klienten mit Hilfe eines Algorithmus mehrere Dienstplan-Vorschlag unterbreitet werden. Der Dienstplan kann von dem Klienten manuell editiert werden und nach den eigenen Vorlieben (die bei der automatischen Erstellung schon größtenteils berücksichtigt wurden) angepasst werden. Der Klient gibt den Dienstplan frei und benachrichtigt sein Team, dass ein neuer Dienstplan verfügbar ist, der online eingesehen werden kann. Der Zeitaufwand für die Assistenten bleibt unverändert. Der Zeitaufwand für den Klient sinkt auf wenige Minuten.

Hier kommt wahrscheinlich (wie von Dir angeregt) noch ein Ablaufdiagramm hin.

3.4. Benutzergeschichten

„Benutzergeschichten beschreiben Anforderungen aus der Sicht des Endanwenders. Sie bestehen aus einem Namen, aus einer kurzen textuellen Beschreibung der Anforderung und einer Reihe von Akzeptanzkriterien. Letztere halten fest, welche Kriterien erfüllt sein müssen, damit die Anforderung als erfolgreich realisiert gilt.“ [Pichler 2008, S. 46]

Wichtig ist, dass die Anforderung einen „konkreten und sichtbaren Mehrwert für den Kunden“ besitzt. [Wirdemann 2011, S. 52]

Benutzergeschichten haben ein festes Muster. Es hat sich für die Beschreibung folgendes Muster bewährt [Wirdemann 2011, S. 59]:

Als <Benutzerrolle> will ich <das Ziel> [so dass <Grund für das Ziel>]

Aus den Gesprächen mit dem Klienten und der Formulierung des Soll-Prozesses sind einige Benutzergeschichten abgeleitet worden:



Standard-Dienstzeiten

Als Klient möchte ich Standard-Dienstzeiten angeben können.

Akzeptanzkriterien:

- Der Klient kann für jeden Wochentag einen Standard-Dienstbeginn und ein Standard-Dienstende festlegen.
- Die Standard-Dienstzeiten können gespeichert und wieder geladen werden.
- Bei der Erstellung eines Monatsplans werden initial die Standard-Dienstzeiten eingetragen.

Angabe Dienstzeiten

Als Klient möchte ich für den nächsten Monat angeben können, wie die Dienstzeiten sind.

Akzeptanzkriterien:

- Es ist möglich einen Monat auszuwählen.
- Initial werden die Standard-Dienstzeiten im Monats-Plan eingetragen.
- Es ist möglich die Dienstzeiten für den nächsten Monat festlegen zu können.
- Die Dienstzeiten für den nächsten Monat können gespeichert und wieder geladen werden.

Öffentliche Bemerkungen (tagesbezogen)

Als Klient möchte ich Bemerkungen zum Tag erstellen können, die für jeden (Klient und Assistenten) sichtbar sind.

Akzeptanzkriterien:

- Es ist möglich öffentliche Bemerkungen pro Tag zu erstellen.
- Die öffentlichen Bemerkungen können gespeichert und wieder geladen werden.
- Die öffentlichen Bemerkungen des Klienten werden den Assistenten bei der Eingabe möglicher Termine angezeigt, weil die Bemerkungen Einfluss auf die Planung haben könnten.
- Die öffentlichen Bemerkungen werden im Dienstplan angezeigt.



Private Bemerkungen (tagesbezogen)

Als Klient möchte ich Bemerkungen zum Tag erstellen können, die nur für mich sichtbar sind.

Akzeptanzkriterien:

- Es ist möglich private Bemerkungen pro Tag zu erstellen, die nur für den Klienten sichtbar sind.
- Die privaten Bemerkungen können gespeichert und wieder geladen werden.
- Die privaten Bemerkungen werden dem Klienten bei der Dienstplanerstellung angezeigt.

Öffentliche Bemerkungen (monatsbezogen)

Als Klient möchte ich monatsbezogene Bemerkungen zum Dienstplan eingeben können.

Akzeptanzkriterien:

- Der Klient hat eine Möglichkeit monatsbezogene Bemerkungen einzugeben.
- Die Bemerkungen können gespeichert und wieder geladen werden.
- Die Bemerkungen werden den Assistenten bei der Eingabe möglicher Termine angezeigt.

Verwaltung des Teams

Als Klient möchte ich mein Team verwalten.

Akzeptanzkriterien:

- Es gibt eine Übersicht der Team-Mitglieder.
- Es ist möglich Team-Mitglieder hinzuzufügen.
- Es ist möglich die Eigenschaften der Team-Mitglieder zu editieren.
- Es ist möglich Team-Mitglieder aus dem Team zu entfernen.
- Es ist möglich das Team abzuspeichern und zu einem späteren Zeitpunkt wieder zu laden.



Bewertung der Assistenten

Als Klient möchte ich die Assistenten bewerten, um bestimmte Assistenten bei der Dienstplanerstellung zu favorisieren.

Akzeptanzkriterien:

- Es ist möglich jedem Assistenten in der Team-Verwaltung einen Priorisierungswert zuzuweisen.
- Der Algorithmus für die Dienstplanerstellung greift auf den Priorisierungswert zu und berücksichtigt diesen.

Berücksichtigung der Wochentage

Als Klient möchte ich für jeden Assistent die Möglichkeiten haben, Vorlieben für bestimmte Wochentage eingeben können.

Akzeptanzkriterien:

- Es ist möglich jedem Assistenten in der Team-Verwaltung favorisierte Wochentage zuzuweisen.
- Der Algorithmus für die Dienstplanerstellung greift auf die favorisierten Wochentage zu und berücksichtigt diese.

Team benachrichtigen

Als Klient möchte ich eine Nachricht an mein Team verschicken können, mit der Bitte freie Termine einzutragen.

Akzeptanzkriterien:

- In der Team-Verwaltung gibt es eine Möglichkeit Kontaktdaten für die Assistenten zu hinterlegen.
- Der Klient hat die Möglichkeit per Knopfdruck sein Team zu benachrichtigen. In dieser Nachricht ist unter anderem ein automatisch generierter Text mit der Bitte, die möglichen Termine einzutragen, enthalten.



Dienstplan-Vorschläge

Als Klient möchte ich mehrere Vorschläge für einen Dienstplan bekommen, die ich manuell anpassen kann.

Akzeptanzkriterien:

- Es gibt einen Algorithmus, der automatisch unter Berücksichtigung der Verfügbarkeiten der Assistenten und der Vorlieben des Klienten einen Dienstplan erstellt, der die Stundenkontingente berücksichtigt.
- Der Dienstplan (der vom Algorithmus erstellt wird) wird dem Klienten übersichtlich in einer Tabelle dargestellt.
- Wenn der Klient erneut einen Dienstplan anfordert, läuft der Algorithmus erneut und ein neuer Dienstplanvorschlag (der vom alten abweicht) wird wieder erzeugt und dargestellt.

Manuelle Anpassung des Dienstplans

Als Klient möchte ich die Möglichkeit haben, den Dienstplan nach meinen Wünschen anzupassen.

Akzeptanzkriterium: Es ist möglich den, vom Algorithmus erstellten, Dienstplan manuell zu verändern.

Stundenverteilung bei Dienstplanerstellung

Als Klient möchte ich bei der manuellen Nachbearbeitung des Dienstplanes die Stundenverteilung sehen können.

Akzeptanzkriterien:

- Es ist möglich in einer Tabelle zu sehen, wie die Arbeitsstunden des Monats auf die Assistenten verteilt werden.
- Es wird dargestellt, wieviel Stunden ein Assistent arbeiten müsste. Diese Daten kommen aus der Team-Verwaltung.
- Die Differenz zwischen Soll- und Ist-Arbeitsstunden pro Assistent wird berechnet.
- Es wird grafisch in 3 Kategorien (grün, gelb, rot) dargestellt, wie gut das Stundenkontingent eines Assistenten ausgeschöpft ist.



Eingabe möglicher Termine

Als Assistent möchte ich angeben können, wann ich Zeit habe zu arbeiten.

Akzeptanzkriterien:

- Der Assistent kann in einem Kalender einen Monat auswählen.
- Der Assistent kann mögliche Termine für einen Monat eintragen.
- Die Eingaben lassen sich speichern und zu einem späteren Zeitpunkt editieren.

Termine favorisieren

Als Assistent möchte ich eine Möglichkeit haben Termine zu favorisieren.

Akzeptanzkriterien:

- Der Assistent kann neben „geht“ und „geht nicht“ auch ein „geht zur Not“ auswählen.
- Der Algorithmus zur Dienstplanerstellung berücksichtigt diese Abstufung der Eingaben.

Bemerkungen der Assistenten

Als Assistent möchte ich bei der Angabe von Terminen auch Bemerkungen machen können, um weitere Informationen transportieren zu können.

Akzeptanzkriterien:

- Neben der Termineingaben kann der Assistent auch Bemerkungen machen.
- Der Klient bekommt die Bemerkungen bei der Dienstplanerstellung angezeigt.

Ausschöpfung Stundenkontingent

Als Assistent möchte ich mein Stundenkontingent möglichst gut ausgeschöpft bekommen.

Akzeptanzkriterien:

- Der Algorithmus zur Dienstplanerstellung berücksichtigt die Stundenkontingente und erstellt einen Dienstplan, bei dem alle Assistenten ihre Stundenkontingente gleichmäßig ausgeschöpft bekommen.



3. Anforderungsermittlung

- In einer Tabelle kann man sehen, wie gleichmäßig die Stundenkontingente der Assistenten ausgeschöpft sind.

Ansicht Dienst- und Bereitschaftszeiten

Als Assistent möchte ich sehen, wann ich Dienst oder Bereitschaft habe.

Akzeptanzkriterien:

- Der Assistent kann den Dienstplan einsehen.
- Im Dienstplan sind die Dienste und Bereitschaften des angemeldeten Assistenten farblich hinterlegt.
- In einer gesonderten Übersicht sieht der Assistent nur seine Dienst- und Bereitschaftszeiten.

Letzte Änderung Dienstplan

Als Anwender möchte ich sehen, wann der Dienstplan zum letzten Mal geändert wurde.

Akzeptanzkriterien:

- Wenn der Dienstplan gespeichert wird, wird das Datum und die Uhrzeit protokolliert.
- Das Datum und die Uhrzeit der letzten Änderung wird beim Dienstplan angezeigt.

Dokumentation

Als Anwender möchte ich eine Dokumentation des Systems zur Verfügung haben, um mich gut in das System einarbeiten zu können.

Akzeptanzkriterien:

- Es steht eine Dokumentation zur Verfügung.
- Die Dokumentation wird abhängig vom Typ des angemeldeten Benutzers (Klient/Assistent) erzeugt.
- Es wird nur die Dokumentation der tatsächlich verwendeten Funktionen angezeigt.



Feedback

Als Anwender möchte ich Feedback an den Entwickler senden können.

Akzeptanzkriterien:

- Der Anwender kann Feedback eingeben und senden.
- Der Anwender kann eine E-Mail-Adresse für eine Antwort eingeben.
- Das Feld für die Antwort E-Mail-Adresse wird nach Möglichkeit automatisch ausgefüllt. Die benötigten Daten werden aus den Einstellungen (für den Klienten) oder der Team-Verwaltung (für die Assistenten) ausgelesen.

3.5. Abgeleitete Anforderungen

Aus den Benutzergeschichten ergeben sich einige abgeleitete Anforderungen, die zwar nicht explizit erwähnt wurden, aber für eine technische Umsetzung zwingend notwendig sind.

Plattformunabhängigkeit

Die Software-Lösung soll sich auf unterschiedlichen Geräte-Kategorien (z .B. Handy, Tablet, Desktop) bedienen lassen können. Weiterhin sollen die größten Betriebssysteme (Windows, Linux, MacOS X, Android, iOS) unterstützt werden.

Serverbasierte Anwendung

Klient und Assistenten möchten an verschiedenen Orten zu verschiedenen Zeiten unabhängig voneinander das System bedienen. Um diese Bedingungen zu erfüllen, bietet sich eine serverbasierte Anwendung an.

Benutzer-Verwaltung

Der Klient und die Assistent haben unterschiedliche Interessen am Assistenzplaner. Der Klient möchte unter anderem den Dienstplan erstellen und sein Team verwalten, die Assistenten hingegen möchten mögliche Termine eintragen und den fertigen Dienstplan einsehen können.

Jeder Benutzer möchte seine Daten individuell eingeben können. Die hinterlegten Daten sollten nicht für jedermann einsehbar sein. Daher bekommt jeder Benutzer



3. Anforderungsermittlung

einen eigenen Zugang, der mit einem personenbezogenen Passwort geschützt wird. Bei der Anmeldung am System wird registriert ob es sich um einen Klienten oder einen Assistenten handelt und dementsprechend der angezeigte Inhalt variiert.

Aufgaben-Verwaltung

BAUSTELLE - muss ich noch schreiben



4. Marktanalyse

Bevor eine Lösung doppelt entwickelt wird, ist es ratsam zu recherchieren, was es schon auf dem Markt gibt.

Die meisten Anwendungen zur Dienstplanerstellung sind für die Verwendung in Krankenhäusern, Einzelhandel, Restaurant oder Clubs optimiert.

4.1. Dienstplan in Excel

Im Internet kann man jede Menge Vorlagen für die Erstellung von Dienstplänen mit Hilfe der Tabellenkalkulation **Excel**² finden. Die Zusammenarbeit von Klient und Assistent über das Internet gestaltet sich schwierig. Der Klient kann Dokumente freigeben, und damit den Assistenten die Möglichkeit geben ihre Termine einzutragen. Allerdings haben sie damit auch Einsicht und Zugriff auf alles, was sie nicht sehen sollten. Es gibt nicht die Möglichkeit einzelne Bereiche des Dokumentes freizugeben.

4.2. Offline Anwendungen

Es gibt Programme zur Dienstplanerstellung, die ohne Internet-Anbindung arbeiten. Hier müsste der Klient die Eingabe der möglichen Termine der Assistenten selbstständig vornehmen.

Ein Beispiel für solch eine Anwendung ist **QTime**³. Aber schon der Blick auf die Screenshots (Beispiel: Abbildung 4.1) lässt erahnen, dass die Software sehr komplex ist und für jemanden der sie nur einmal im Monat verwenden möchte zuviel Einarbeitung abverlangt.

²<http://office.microsoft.com/de-de/excel/> - zuletzt abgerufen am 23.08.2014

³<http://www.qtime.de/kostenlos/dienstplan-erstellung.html> - zuletzt abgerufen am 23.08.2014



4. Marktanalyse

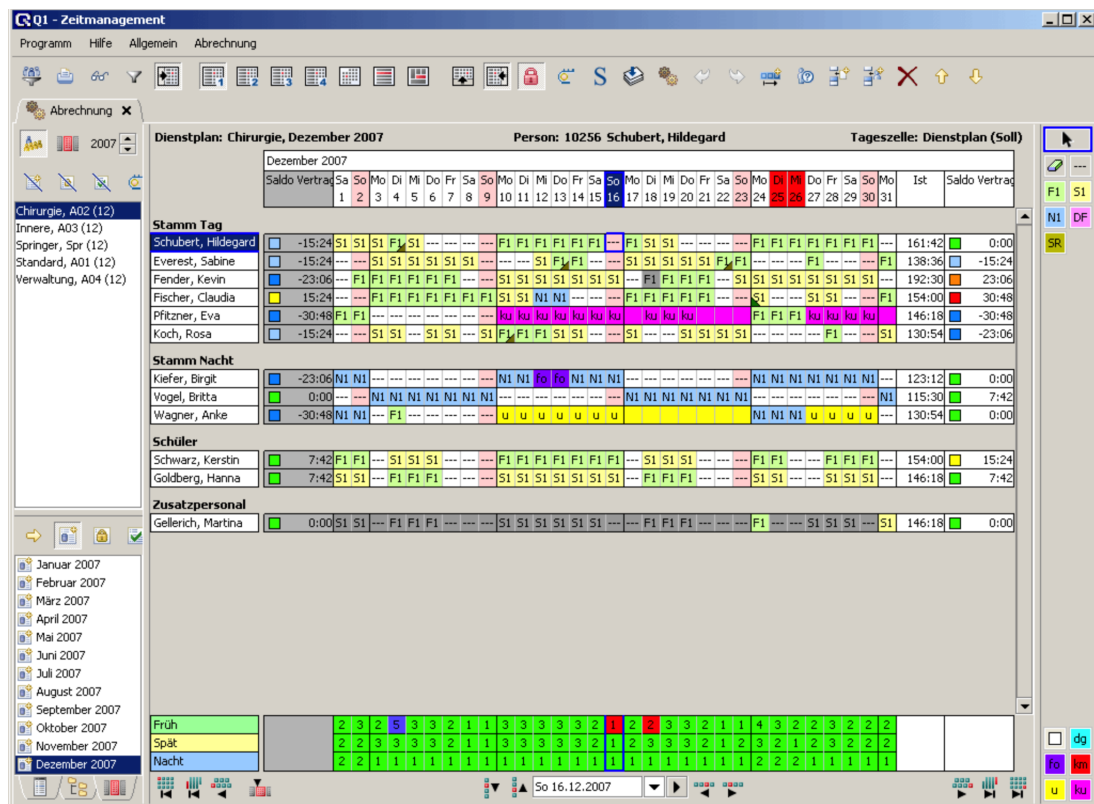


Abbildung 4.1.: Screenshot von QTime: Komplexe Software

4.3. Online Anwendungen

Es gibt einige Anwendungen im Internet, die es ermöglichen, Dienstpläne zu erstellen.

Beispiele hierfür sind:

- [Kostenloser Dienstplan](http://kostenloser-dienstplan.de)⁴
- [Schichtplaner](http://schichtplaner-online.de/index.php/de/)⁵
- [Papershift](http://www.papershift.com)⁶
- [EasyPep](https://easypep.de)⁷
- [ShiftJuggler](https://www.shiftjuggler.com/de/)⁸

⁴<http://kostenloser-dienstplan.de> - zuletzt abgerufen am 23.08.2014

⁵<http://schichtplaner-online.de/index.php/de/> - zuletzt abgerufen am 23.08.2014

⁶<http://www.papershift.com> - zuletzt abgerufen am 23.08.2014

⁷<https://easypep.de> - zuletzt abgerufen am 23.08.2014

⁸<https://www.shiftjuggler.com/de/> - zuletzt abgerufen am 23.08.2014



4. Marktanalyse

Keiner dieser Anwendungen unterstützt das Bereitschaftskonzept. Einige der Anwendungen sind in der Lage Stundenkontingente zu berücksichtigen. Bei allen Anwendungen wird der Dienstplan manuell erstellt. Bei keiner Anwendung können persönliche Vorlieben oder allgemein Regeln für die Dienstplanerstellung festgelegt werden. Alle Anwendungen waren auf die reine Erstellung eines Dienstplans ausgelegt, Erweiterungen wie Aufgaben-Verwaltung waren nicht vorgesehen.

Zusammenfassend lässt sich festhalten, dass es keine bestehende Lösung gibt, die die vorliegenden Aufgabe erledigen kann. Es gibt auch keine Anknüpfungsmöglichkeiten oder Wiederverwendungsmöglichkeiten, da die meisten Anbieter ein finanzielles Interesse haben und den Code ihrer Lösungen nicht offen gelegt haben.



5. Software-Design

Dieses Kapitel ist noch in Bearbeitung!

Mit SysML lassen sich Systeme, bestehend aus Hard- und Software, modellieren. Da in der vorliegenden Arbeit eine Lösung entstanden ist, die ausschließlich aus Software besteht, wurde nur das Anforderungsdiagramm der SysML verwendet. Alles andere wurde in UML modelliert.

5.1. SysML Anforderungsdiagramm

In Abbildung 5.1 ist das Anforderungsdiagramm zu sehen.

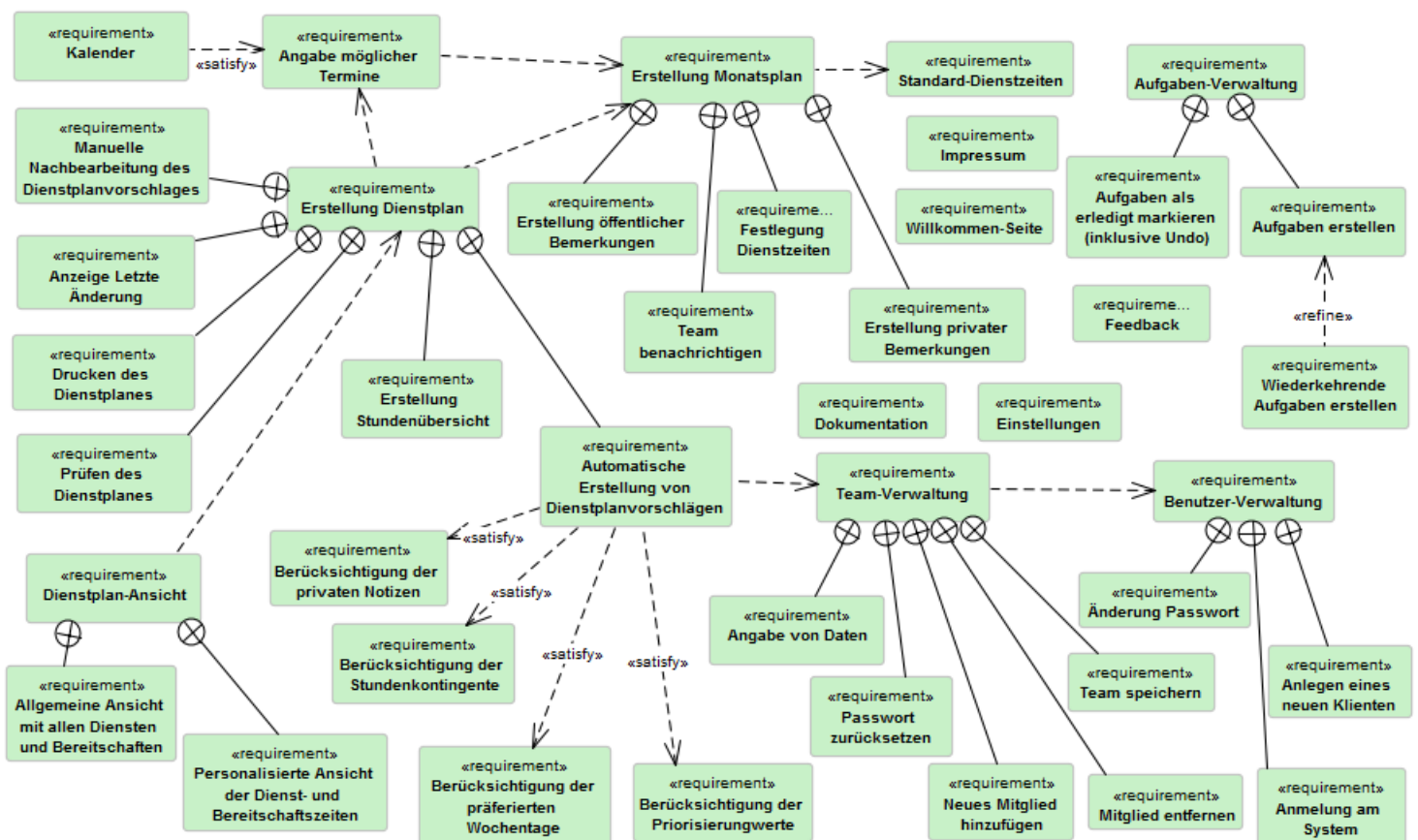


Abbildung 5.1.: SysML Anforderungsdiagramm



5.2. UML Komponentendiagramm

Die Anforderungen werden nach dem Top-Down-Prinzip in Software umgesetzt. Es werden Komponenten aus den Anforderungen abgeleitet und modelliert.

Das Komponentendiagramm ist in Abbildung 5.2 zu sehen.

Platzhalter

Abbildung 5.2.: UML Komponentendiagramm

5.3. UML Klassendiagramme

Alle Klassen und deren Beziehungen sind in Abbildung 5.3 zu sehen.

Die Klassendiagramme sind noch nicht fertig! Es gibt schon ein Klassendiagramm, welches alle Klassen enthält, aber noch sehr unübersichtlich ist. Zwischen die Klassen werden noch Interfaces gestellt, so dass man Komponenten erstellen kann und diese übersichtlich in eigenen Klassendiagrammen darstellen kann.

5.4. UML Sequenzdiagramme

Sequenzdiagramme habe ich noch nicht erstellt



5. Software-Design

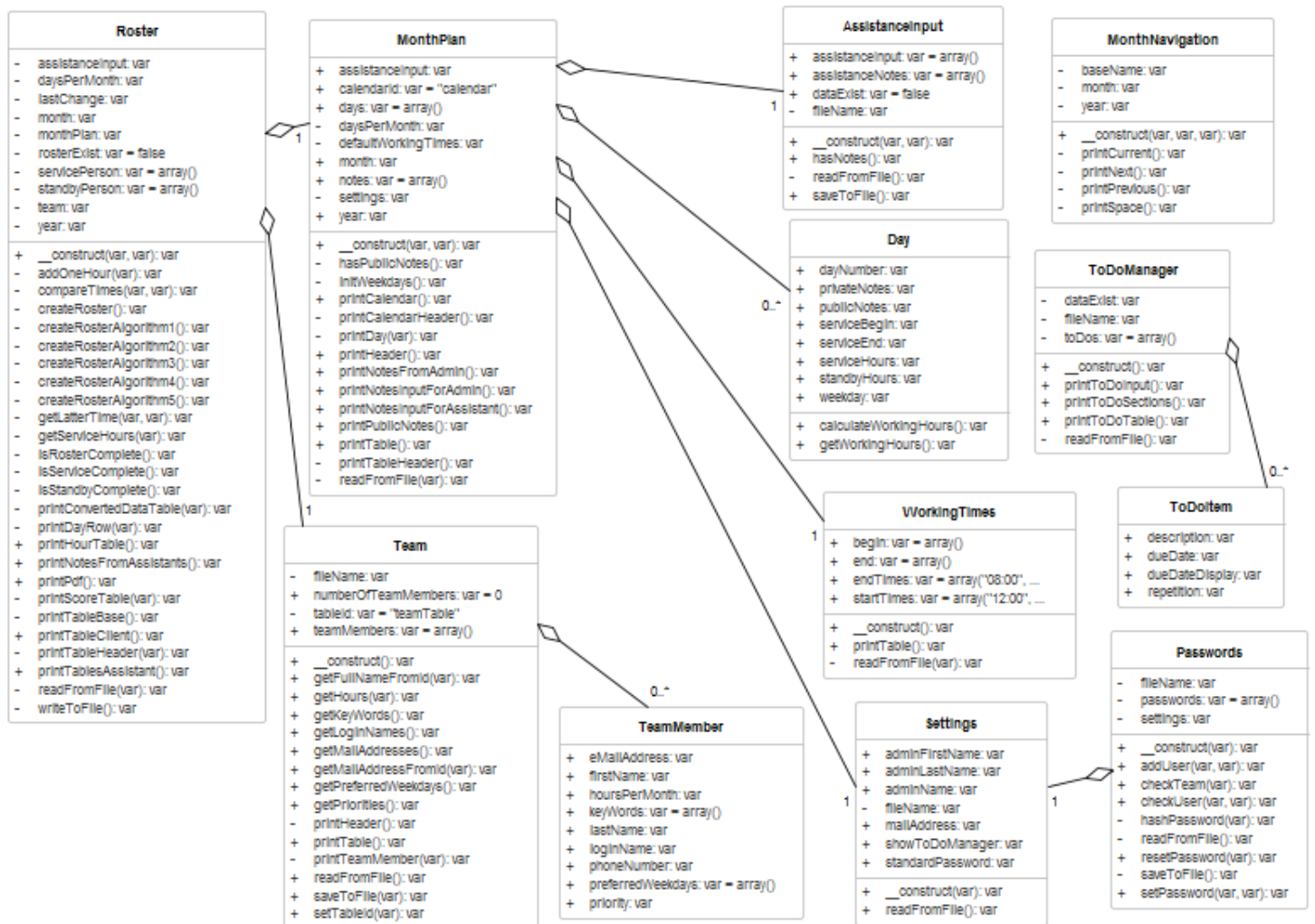


Abbildung 5.3.: UML Klassendiagramm



6. Implementierung

In diesem Kapitel wird dargestellt, wie die Anforderungen und das Software-Design konkret implementiert wurden. Der Schwerpunkt liegt auf der Darstellung der Ergebnisse⁹.

6.1. Wahl der Programmiersprachen

Für die Implementierung ist es notwendig sich auf Programmiersprachen festzulegen.

Server

Bei der Serverseite ist die Wahl der Programmiersprache auf PHP gefallen. Hauptgründe hierfür waren: die **hohe Verbreitung**¹⁰ von über 80% und die Tatsache, dass PHP bei den meisten Webhostern vorinstalliert ist. Weitere Gründe [Theis 2010, S. 16] sind:

- Einfache Entwicklung von Programmen
- Unterstützung verschiedener Plattformen
- Leichte Integration in Apache (weitverbreiteter Webserver)

Client

Der Server generiert mit Hilfe von PHP den HTML-Code, der an den Client (Browser) gesendet wird. Im HTML wird ausschließlich die Struktur der angezeigten Seite definiert. Das Aussehen wird mit Hilfe von CSS bestimmt. Um die Funktionalität auf der Seite des Clients zu erweitern, kommt JavaScript zum Einsatz.

⁹Der Autor hat sich gegen die Darstellung und Beschreibung von Quellcode entschieden. Der interessierte Leser kann den Quellcode auf der beiliegenden CD oder bei GitHub (<https://github.com/Ulle84/AssistancePlaner>) einsehen

¹⁰http://w3techs.com/technologies/history_overview/programming_language - zuletzt abgerufen am 23.08.2014



6.2. Persistenz der Daten

Alles was persistent gespeichert werden muss, wird in textbasierten Dateien (*.txt) abgelegt. Im Vergleich zu Datenbanken hat man folgende Vorteile:

- Daten können leicht zwischen Produktiv-System und Test-System ausgetauscht werden.
- Datenbank-Verwaltung entfällt.
- Es werden keine Datenbank-Anforderungen an den Server gestellt.

Eine potentielle Fehlerquelle bei dem Gebrauch von textbasierten Daten ist, wenn zwei Benutzer gleichzeitig schreibend auf eine Datei zugreifen wollen. Das ist derzeit nur bei den Kalender-Eingaben der Assistenten (siehe Abschnitt 6.12) möglich. Um den gleichzeitigen Zugriff zu verhindern wird eine Art Semaphore eingesetzt. Unter PHP gibt es keine echten Semaphoren, aber es gibt die Möglichkeit auf eine Datei sperrend zuzugreifen. So wird verhindert, dass zwei Prozesse gleichzeitig in eine Datei schreiben, falls zwei Assistenten gleichzeitig ihre Daten speichern wollen.

6.3. Benutzer-Verwaltung

Damit der Klient und die Assistenten benutzerspezifische Eingaben tätigen können ist eine Verwaltung der Benutzer notwendig. Jeder Benutzer des Assistenzplaners kann sich mit eigener Kennung und Passwort anmelden. Es gibt zwei Rollen von Benutzern: die Klienten und die Assistenten. Die Klienten können sich ihren Zugang selbstständig anlegen (siehe Abschnitt 6.6). Die Zugänge der Assistenten werden von den Klienten verwaltet. Klienten haben die Möglichkeit Assistenten zum Team hinzuzufügen und zu entfernen (siehe Abschnitt 6.10). Bei der Anmeldung wird eine PHP-Session gestartet. Dies geschieht mit dem PHP-Befehl:

```
session_start();
```

Im Hintergrund wird im Browser eine Session-ID erstellt und gespeichert. Ein Beispiel für solch eine Session-ID findet sich in Abbildung 6.1.

Nach der Anmeldung werden im superglobalen Array `$_SESSION` Variablen gesetzt, die hilfreich für die weitere Verwendung des Assistenzplaners sind. Diese Vorgehensweise (Start einer Session und Setzen der Session-Variablen) ist [Schlossnagle 2006, S. 404] entnommen.

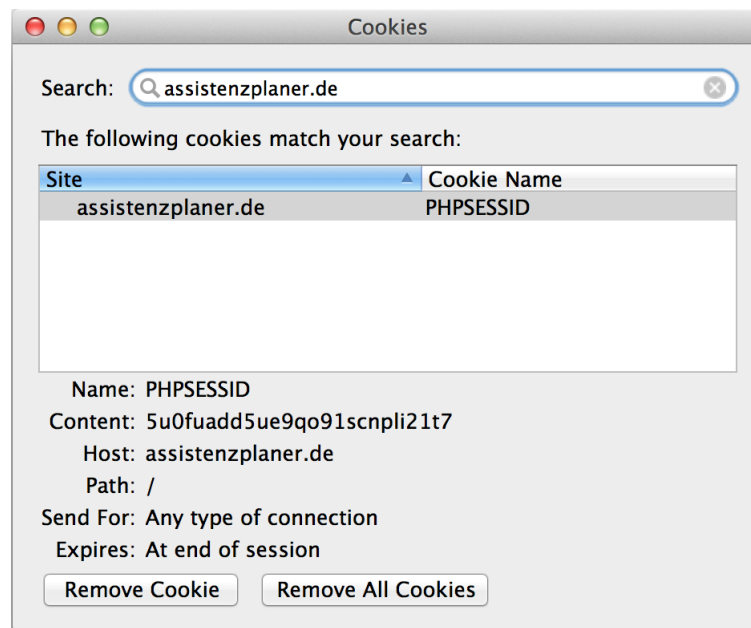


Abbildung 6.1.: Session-ID wird als Cookie im Browser gespeichert

6.4. Willkommen-Seite

Auf der **Willkommen-Seite**¹¹ (siehe Abbildung 6.2) werden dem Interessenten Informationen über den Assistenzplaner präsentiert. Man wird gebeten einen neuen Klienten-Zugang einzurichten oder sich mit einem bestehenden Zugang anzumelden.

Willkommen beim Assistenzplaner

Der Assistenzplaner ist ein Hilfsmittel für körperlich behinderte Menschen, die auf Assistenz angewiesen sind.

Mit Hilfe des Assistenzplaners ist es unter anderem möglich:

- Assistenz-Teams zu verwalten
- Dienstpläne zu erstellen
- Aufgaben der Assistenten zu verwalten

Bitte [melden Sie sich an](#) oder [legen Sie einen neuen Klienten-Zugang an](#).

Abbildung 6.2.: Willkommen-Seite

¹¹<http://www.assistenzplaner.de/master/PHP/index.php> - zuletzt abgerufen am 23.08.2014



6.5. Navigation

Die Navigation steht immer gut sichtbar rechts oben zur Verfügung. Abhängig davon, ob jemand am System angemeldet ist (siehe Abbildung 6.3) oder nicht (siehe Abbildung 6.4) stellt sie unterschiedliche Informationen dar.

Klient: Mustermann Assistent: A1 [Übersicht](#) [Passwort ändern](#) [Dokumentation](#) [Feedback](#) [Impressum](#) [Abmelden](#)

Abbildung 6.3.: Navigation bei angemeldetem Assistenten

[Willkommen](#) [Neuen Klienten-Zugang einrichten](#) [Impressum](#) [Anmelden](#)

Abbildung 6.4.: Navigation ohne Anmmeldung

6.6. Anlegen eines neuen Klienten-Zugangs

Das Anlegen eines neuen Klienten-Zugangs soll keine Hürde darstellen. Es werden bei der Erstellung eines neuen Klienten-Zugangs keine persönlichen Daten erhoben.

Um einen neuen Klienten-Zugang anzulegen wird man gebeten ein Klientennamen, ein Passwort und die Passwort-Wiederholung einzugeben (siehe Abbildung 6.5).

Name des Klienten:	<input type="text"/>
Passwort:	<input type="password"/>
Wiederholung Passwort:	<input type="password"/>
<input type="button" value="Klient anlegen"/>	

Abbildung 6.5.: Anlegen eines neuen Klienten mit Name und Passwort

Bedingungen an einen neuen Klientennamen sind:

- Er darf nicht leer sein.
- Er darf noch nicht vergeben sein.
- Er darf nur aus Buchstaben und Zahlen bestehen. Sonderzeichen und Leerzeichen sind nicht erlaubt.

Wenn eine Bedingung nicht erfüllt ist, so wird der Benutzer darauf hingewiesen. Der Inhalt der Textfelder bleibt erhalten, so dass der Benutzer das Passwort und die Wiederholung dessen nicht nochmal eingeben muss.



Wenn alle Bedingungen erfüllt sind und ein Passwort vergeben wurde, wird im Ordner **Data** ein Unterordner mit dem Klientennamen angelegt, in dem alle Daten persistent gespeichert werden. Weiterhin wird in der in einer Datei das vergebene Passwort verschlüsselt abgespeichert. Der Benutzer bekommt eine Meldung, dass das Anlegen eines neuen Kontos erfolgreich war und wird gebeten sich mit dem neuen Zugang am System anzumelden.

Zu einem späteren Zeitpunkt kann der Klient in den Einstellungen personenbezogene Daten (Vorname, Nachname, E-Mail-Adresse) angeben, was aber nicht zwingend für die Benutzung des Assistenzplaners notwendig ist.

6.7. Übersicht

Auf der Übersichts-Seite (siehe Abbildung 6.6) werden dem Benutzer alle verfügbaren Seiten als Hyperlink aufgelistet. Hierbei werden der Benutzertyp (Klient oder Assistent) und die Einstellungen (z. B. Aufgaben-Verwaltung benutzen oder nicht) berücksichtigt.

Übersicht Assistenzplaner

[Dienst-Plan](#)
[Aufgaben](#)
[Monats-Plan](#)
[Team](#)
[Standard Dienst-Zeiten](#)
[Einstellungen](#)

Abbildung 6.6.: Übersicht Assistenzplaner für Klienten - von hier aus erreicht man per Link alle verfügbaren Seiten

6.8. Einstellungen

Um den Assistenzplaner zu konfigurieren gibt es die Einstellungen (siehe Abbildung 6.7). Hier können Vorname, Nachname und E-Mail-Adresse des Klienten gespeichert werden. Weiterhin kann hier das Standard-Passwort angesehen und editiert werden. Jeder neu angelegte Assistent kann sich mit dem Standard-Passwort beim Assistenzplaner anmelden. Weiterhin kann das Passwort eines Assistenten vom Klienten auf das Standard-Passwort zurückgesetzt werden, falls der Assistent sein Passwort vergessen hat. Zum Schluss kann der Klient in den Einstellungen festlegen, ob er die Aufgaben-Verwaltung (siehe Abschnitt 6.15) nutzen möchte oder nicht.



Einstellungen

Beschreibung	Wert
Vorname	Max
Nachname	Mustermann
E-Mail Adresse	max@mustermann.de
Standard-Passwort für Assistenten	Hallo123
Aufgaben-Verwaltung nutzen	<input checked="" type="checkbox"/>

Abbildung 6.7.: Einstellungsmöglichkeiten des Assistenzplaners

6.9. Standard-Dienstzeiten

Bei den Standard-Dienstzeiten (siehe Abbildung 6.8) hat der Klient die Möglichkeit für jeden Wochentag festzulegen wie standardmäßig die Dienstzeiten sind. Diese Standard-Dienstzeiten werden initial im Monatsplan eingetragen, können aber vom Klienten für jeden Tag angepasst werden.

Standard-Dienstzeiten

Wochentag	Dienstbeginn	Dienstende
Mo	17:00 ↕	08:00 ↕
Di	13:00 ↕	08:00 ↕
Mi	17:00 ↕	08:00 ↕
Do	13:00 ↕	08:00 ↕
Fr	14:00 ↕	13:00 ↕
Sa	13:00 ↕	13:00 ↕
So	13:00 ↕	09:00 ↕

Abbildung 6.8.: Verwaltung der Standard-Dienstzeiten

6.10. Team-Verwaltung

Mit der Team-Verwaltung kann der Klient die Eigenschaften seiner Team-Mitglieder verwalten.

Jeder Assistent hat folgende Eigenschaften:



6. Implementierung

- Kennung - eindeutige ID, mit der sich der Assistent am System anmeldet
- Vorname
- Nachname
- Telefonnummer
- Stichwörter - werden bei der Dienstplanerstellung berücksichtigt
- Stundenkontingent - Anzahl der Stunden in einem Monat, die der Assistent zur Verfügung hat
- Priorisierung - ein Wert um den Assistenten bei der Dienstplanerstellung zu priorisieren (je höher der Wert, desto höher die Wahrscheinlichkeit, dass der Assistent viele Dienste bekommt)
- Bevorzugte Wochentage

Die Eigenschaften werden in tabellarischer Form dargestellt (siehe Abbildung 6.9) und sind editierbar.

Team Übersicht

Kennung	Vorname	Nachname	E-Mail Adresse	Telefonnummer	Stichwörter (getrennt durch Leerzeichen)	Stundenkontingent	Priorisierung	Bevorzugte Wochentage	Aktionen
Max	Max	Muster	max@muster.de		FCB Tatort	200	2	<input type="checkbox"/> Mo <input type="checkbox"/> Di <input type="checkbox"/> Mi <input type="checkbox"/> Do <input type="checkbox"/> Fr <input type="checkbox"/> Sa <input type="checkbox"/> So	Löschen Passwort zurücksetzen
Michael	Michael	Muster	michael@muster.de		Handball	200	1	<input type="checkbox"/> Mo <input type="checkbox"/> Di <input type="checkbox"/> Mi <input type="checkbox"/> Do <input type="checkbox"/> Fr <input type="checkbox"/> Sa <input type="checkbox"/> So	Löschen Passwort zurücksetzen

Abbildung 6.9.: Verwaltung der Assistentz-Team-Mitglieder

Neue Teammitglieder können mit Hilfe des Buttons „Neues Mitglied“ hinzugefügt werden. Das neue Teammitglied hat initial das Passwort was in den Einstellungen hinterlegt wurde.

Sollte ein Assistent sein Passwort vergessen haben, hat der Klient mit Hilfe des Buttons „Paswort zurücksetzen“ (in der Spalte Aktionen) die Möglichkeit das Passwort des Assistenten auf den Wert des Standardpasswortes (was in den Einstellungen hinterlegt ist) zu setzen.

Sollte ein Assistent das Team verlassen, so hat der Klient die Möglichkeit mit Hilfe des Buttons „Löschen“ (in der Spalte Aktionen) den Assistenten aus der Team-Verwaltung zu entfernen.

6.11. Monatsplan

Mit dem Monatsplan (siehe Abbildung 6.10) plant der Klient den nächsten Monat. Er kann pro Tag den Dienstbeginn und das Dienstende festlegen und Notizen zum



6. Implementierung

Tag erstellen. Die Notizen unterteilen sich in öffentliche und private Notizen. Öffentliche Notizen können auch vom Assistenzteam eingesehen werden. Private Notizen können nur vom Klienten selbst eingesehen werden.

Monatsplan für September 2014

Datum	Dienstbeginn	Dienstende	Bemerkungen (öffentlich)	Bemerkungen (privat)
Mo, 01.09.	13:00 ▾	13:00 ▾	Urlaub	
Di, 02.09.	13:00 ▾	13:00 ▾	Urlaub	
Mi, 03.09.	13:00 ▾	13:00 ▾	Urlaub	

Abbildung 6.10.: Monatsplan mit der Möglichkeit tagesweise Dienstbeginn und Dienstende festzulegen und Bemerkungen zu hinterlegen

Neben den tagesweisen Bemerkungen hat der Klient noch eine Möglichkeit eine Nachricht an sein Team zu verfassen. Hier können allgemeine Hinweise für den Monat untergebracht werden.

Der Klient hat jederzeit die Möglichkeit den Monatsplan zu speichern. Möchte er sein Assistenz-Team auffordern mögliche Termine einzutragen, so kann er auf den Button „Team benachrichtigen“ klicken. Vom Assistenzplaner wird dann eine E-Mail an jedes Mitglied des Assistenz-Teams geschickt mit der Bitte mögliche Termine einzutragen und der Nachricht des Klienten. Eine beispielhafte Nachricht findet sich in Abbildung 6.11.

Assistenzplaner - Bitte mögliche Termine eintragen

Liebes Team,

bitte tragt bis **15. August 2014** Eure möglichen Termine für den September 2014 im [Assistenzplaner](#) ein.

Vielen Dank!

Hier noch eine Nachricht von Mustermann:

Liebes Team,

bitte beachtet die veränderten Dienstzeiten aufgrund meines 2-wöchigen Urlaubs.

Viele Grüße,
Max Mustermann

Abbildung 6.11.: Beispielhafte Benachrichtigung der Assistenten, mit der Bitte des Klienten die Termine für den nächsten Monat einzutragen.

6.12. Kalender

Mit Hilfe des Kalenders (siehe Abbildung 6.12) kann ein Assistent die Termine eintragen, an denen er Dienst machen kann. Durch einen Klick auf ein Datum ändert



6. Implementierung

sich der Zustand des Datums und damit auch die Hintergrundfarbe. Rot bedeutet, der Assistent ist an diesem Datum nicht verfügbar. Gelb bedeutet, dass der Assistent zur Not verfügbar wäre. Grün bedeutet, dass der Assistent an dem Datum uneingeschränkt zur Verfügung steht.

Eingabe der Daten

Mo	Di	Mi	Do	Fr	Sa	So
1 17:00 - 08:00	2 13:00 - 08:00	3 17:00 - 08:00	4 13:00 - 08:00	5 14:00 - 13:00	6 13:00 - 13:00	7 13:00 - 08:00
8 17:00 - 08:00	9 13:00 - 08:00	10 17:00 - 08:00	11 13:00 - 08:00	12 14:00 - 13:00	13 13:00 - 13:00	14 13:00 - 08:00
15 17:00 - 08:00	16 13:00 - 08:00	17 17:00 - 08:00	18 13:00 - 08:00	19 14:00 - 13:00	20 13:00 - 13:00	21 13:00 - 08:00
22 17:00 - 08:00	23 13:00 - 08:00	24 17:00 - 08:00	25 13:00 - 08:00	26 14:00 - 13:00	27 13:00 - 13:00	28 13:00 - 08:00
29 17:00 - 08:00	30 13:00 - 08:00					

Alle Daten markieren

Abbildung 6.12.: Kalender für die Eingabe der möglichen Termine

Mit Hilfe des Buttons „Alle Daten markieren“ kann man alle Daten eines Monats grün (bedeutet: Assistent verfügbar) markieren.

Im oberen Bereich sind alle Informationen dargestellt (siehe Abbildung 6.13), die dem Assistenten helfen können. Das sind zum einen die öffentlichen Notizen, die der Klient zum Monat erstellt hat, zum anderen die optionale Nachricht die der Klient an das Team verfasst hat.

Allgemeine Bemerkungen von Mustermann

Liebes Team,

bitte beachtet meinen Urlaub am Anfang des Monats, bei dem ich 24 stündige Assistenzen benötige.

Vielen Dank!

Bemerkungen zu den Terminen

Datum	Bemerkung
Mo, 01.09.	Urlaub
Di, 02.09.	Urlaub

Abbildung 6.13.: Bemerkungen vom Klienten für seine Assistenten



6.13. Dienst-Plan

Aus Sicht des Klienten

Wenn die Assistenten ihre Termine eingetragen haben, ist der Klient in der Lage den Dienstplan zu erstellen. Falls es Tage gibt, an denen nicht mindestens 2 Assistenten (einer für Dienst und einer für Bereitschaft) Zeit haben, so bekommt er eine Warnung, da dann eine vollständige Dienstplanerstellung nicht möglich ist.

Der Assistenzplaner erstellt eine Tabelle (siehe Abbildung 6.14), in der tageweise die Dienstzeiten, die Verfügbarkeiten der Assistenten und die Bemerkungen (öffentliche wie private) aufgeführt werden. Die Verfügbarkeiten der Assistenten werden farblich markiert.

- Grün: Assistent ist voll verfügbar
- Gelb: Assistent ist zur Not verfügbar
- Rot: Assistent ist nicht verfügbar

Dienstplan für September 2014

Letzte Änderung: 08.08.2014 08:13

Datum	Zeit	Michael	Matthias	Markus	Bemerkungen (öffentlich)	Bemerkungen (privat)
Mo, 01.09.	13:00 - 13:00		Dienst	Bereitschaft	Urlaub	
Di, 02.09.	13:00 - 13:00	Bereitschaft	Dienst		Urlaub	
Mi, 03.09.	13:00 - 13:00	Dienst		Bereitschaft	Urlaub	

Abbildung 6.14.: Editierbarer Dienstplan für den Klienten

Der Klient bekommt vom Assistenzplaner mit Hilfe des Algorithmus (siehe Abschnitt 6.14) einen Dienstplan erstellt, den er nach seinen Belieben anpassen kann. Dazu klickt er in die Zellen der Dienstplantabelle und kann zwischen den Zuständen

- Assistent hat Dienst
- Assistent hat Bereitschaft
- Assistent hat weder Dienst noch Bereitschaft

wechseln.

Unterhalb der Dienstplantabelle ist eine weitere Tabelle (siehe Abbildung 6.15), in der man die Stundenverteilung der Assistenten sehen kann.

In der zweiten Spalte sind die zusammengerechneten Arbeitsstunden (Dienststunden plus Bereitschaftsstunden) aufgeführt. In der dritten Spalte sieht man die benötigten Arbeitsstunden. In der vierten Spalte wird die Differenz zwischen Spalte 2 und

**Stundenübersicht**

Person	Stunden	Benötigte Stunden	Differenz in Stunden	Differenz in Prozent
Michael	209	200	9	5 %
Matthias	125	120	5	4 %
Markus	105	100	5	5 %

Abbildung 6.15.: Stundenübersicht der Assistenten

3 berechnet. In der fünften Spalte ist die Differenz (Spalte 4) ins Verhältnis der benötigten Arbeitsstunden (Spalte 3) gesetzt und prozentual ausgedrückt. Die Zeilen werden farblich hinterlegt:

- Grün: Der Betrag der prozentualen Abweichung (Spalte 5) ist kleiner 10%
- Gelb: Der Betrag der prozentualen Abweichung (Spalte 5) ist zwischen 10% und 20%
- Rot: Der Betrag der prozentualen Abweichung (Spalte 5) ist größer als 20%

Die Stundenübersicht wird jedes Mal aktualisiert, wenn der Klient in der Dienstplan-Tabelle etwas ändert. Dadurch hat der Klient ein direktes Feedback, wie sich eine Dienstplanänderung auf die Stundenausschöpfung der Assistenten auswirkt.

Per Knopfdruck kann der Klient einige Funktionen ausführen:

- Dienstplan prüfen: Es wird geprüft, dass für jeden Tag genau ein Assistent Dienst und genau ein Assistent Bereitschaft hat.
- Dienstplan speichern: Der Dienstplan wird gespeichert. Gleichzeitig wird das Datum der letzten Änderung aktualisiert.
- Dienstplan löschen: Der Dienstplan wird nach Rückversicherung gelöscht. Durch ein Neuladen der Dienstplan-Seite wird vom Assistenzplaner ein neuer Dienstplanvorschlag erstellt.
- Dienstplan als PDF anzeigen: Der Dienstplan wird als *Portable Document Format (PDF)* exportiert. Als PDF kann der Dienstplan leicht per E-Mail verschickt und ausgedruckt werden.

Aus Sicht des Assistenten

Der Assistent hat keine Möglichkeit den Dienstplan zu verändern. Er kann lediglich den fertigen Dienstplan einsehen. In einer Tabelle (siehe Abbildung 6.16) werden tagesweise die Dienst- und Bereitschaftszeiten dargestellt. Weiterhin kann man sehen



6. Implementierung

welcher Assistent Dienst bzw. Bereitschaft hat. Die Dienste und Bereitschaften des angemeldeten Assistenten werden farblich hervorgehoben.

Dienstplan für September 2014

Letzte Änderung: 08.08.2014 08:13

Datum	Dienst-Zeit	Dienst	Bereitschafts-Zeit	Bereitschaft	Bemerkungen
Mo, 01.09.	13:00 - 13:00	Matthias	10:00 - 11:00 und 18:00 - 19:00	Markus	Urlaub
Di, 02.09.	13:00 - 13:00	Matthias	10:00 - 11:00 und 18:00 - 19:00	Michael	Urlaub
Mi, 03.09.	13:00 - 13:00	Michael	10:00 - 11:00 und 18:00 - 19:00	Markus	Urlaub

Abbildung 6.16.: Gesamter Dienstplan aus Sicht des Assistenten „Markus“ - farblich hervorgehoben sind die Dienste bzw. Bereitschaften des angemeldeten Assistenten

Unter dem allgemeinen Dienstplan gibt es eine Tabelle (siehe Abbildung 6.17) in der nur die Dienste und Bereitschaften des angemeldeten Assistenten angezeigt werden.

Meine Dienste und Bereitschaften

Datum	Zeit	Typ
Mo, 01.09.	10:00 - 11:00 und 18:00 - 19:00	Bereitschaft
Mi, 03.09.	10:00 - 11:00 und 18:00 - 19:00	Bereitschaft

Abbildung 6.17.: Dienste und Bereitschaften des angemeldeten Assistenten

6.14. Entwicklung des Algorithmus zur Dienstplanerstellung

Im folgenden Abschnitt wird erläutert, was die Anforderungen an den Algorithmus zur Dienstplanerstellung sind und wie der Algorithmus entwickelt und iterativ verfeinert worden ist.

Anforderungen

Mindestanforderung an den Algorithmus ist, dass er einen Dienstplan erstellt, der eine hundertprozentige Betreuungsabdeckung erreicht. Für jeden Tag muss ein Assistent für den Dienst und ein Assistent für die Bereitschaft im Dienstplan eingetragen sein.

Nebenbedingungen sind, dass



6. Implementierung

- die Vorlieben des Klienten berücksichtigt werden (z. B. Assistent A lieber am Wochenende, Assistent B bei einem wichtigen Termin, Assistent C bei Fußball-Spielen)
- die Stundenkontingente der Assistenten gleichmäßig ausgeschöpft werden (vgl. Abschnitt 3.1)

Eingabe

Der Algorithmus bekommt als Eingabe die Verfügbarkeiten der Assistenten (pro Tag). Diese sind folgendermaßen codiert:

- 0: Assistent ist nicht verfügbar
- 1: Assistent ist zur Not verfügbar
- 10: Assistent ist voll verfügbar

Weiterhin hat der Algorithmus Zugriff auf alle Daten, die in der Team-Verwaltung (siehe Abschnitt 6.10) hinterlegt sind. Welche Version des Algorithmus auf welche Daten zugreift lässt sich Tabelle 6.1 entnehmen.

Version	1	2	3	4	5
Priorisierung	✗	✓	✓	✓	✓
Bevorzugte Wochentage	✗	✗	✓	✓	✓
Stichwörter	✗	✗	✗	✓	✓
Stundenkontingent	✗	✗	✗	✗	✓

Tabelle 6.1.: Übersicht der verwendeten Daten aus der Team-Verwaltung

Ausgabe

Der Algorithmus gibt für jeden Tag des Monats einen Assistenten für den Dienst und einen Assistenten für die Bereitschaft aus.

Erste Version - Hundertprozentige Betreuungsabdeckung

Die erste Version des Algorithmus berücksichtigt nur die Mindestanforderung (hundertprozentige Betreuungsabdeckung). Der Algorithmus iteriert über alle Tage des Monats und prüft, ob es mindestens zwei Assistenten gibt, die Zeit haben. Der erste freie Assistent bekommt den Dienst, der zweite freie Assistent bekommt die Bereitschaft zugewiesen. Bei dieser ersten Version entsteht ein vollständiger Dienstplan. Der Klient ist den ganzen Monat betreut, jedoch wurden weder die Stundenkontingente noch die Vorlieben des Klienten berücksichtigt.



Zweite Version - Berücksichtigung der Priorisierungswerte

In der zweiten Version des Algorithmus werden die allgemeinen Vorlieben des Klienten berücksichtigt. Für jedes Teammitglied kann der Klient einen Priorisierungswert festlegen (siehe Abschnitt 6.10). Es wird eine Punktetabelle erstellt. Die erste Spalte ist der Tag des Monats, jede weitere Spalte repräsentiert die Verfügbarkeit eines Assistenten. Ist der Assistent nicht verfügbar, so wird eine Null eingetragen. Ist der Assistent zur Not verfügbar, so wird eine Eins eingetragen. Ist der Assistent voll verfügbar, so wird eine Zehn eingetragen. Als Beispiel dienen die Zahlen in Tabelle 6.2.

Tag des Monats	Punkte Assistent 1	Punkte Assistent 2	Punkte Assistent 3
1 (Montag)	0	10	10
2 (Dienstag)	1	10	0
3 (Mittwoch)	10	1	10
...

Tabelle 6.2.: Punktetabelle zur Dienstplanerstellung

Nun wird jeder Verfügbarkeitswert mit dem Priorisierungswert multipliziert. Mit den beispielhaften Priorisierungswerten aus Tabelle 6.3 verändert sich die Punktetabelle.

Assistent	Priorisierungswert
Assistent 1	1
Assistent 2	2
Assistent 3	3

Tabelle 6.3.: Priorisierungswerte aus der Team-Verwaltung

Das Ergebnis ist in Tabelle 6.4 zu sehen.

Tag des Monats	Punkte Assistent 1	Punkte Assistent 2	Punkte Assistent 3
1 (Montag)	0	20	30
2 (Dienstag)	1	20	0
3 (Mittwoch)	10	2	30
...

Tabelle 6.4.: Punktetabelle nach Multiplikation mit Priorisierungswerten

Nach der Erstellung der Punktetabelle iteriert der Algorithmus über alle Tage und gibt dem Assistenten mit der höchsten Punktzahl den Dienst und dem Assistenten mit der zweithöchsten Punktzahl die Bereitschaft. Mit den Beispielwerten würde sich ein Dienstplan wie er in Tabelle 6.5 dargestellt ist ergeben.



6. Implementierung

Tag des Monats	Dienst	Bereitschaft
1 (Montag)	Assistent 3	Assistent 2
2 (Dienstag)	Assistent 2	Assistent 1
3 (Mittwoch)	Assistent 3	Assistent 1
...

Tabelle 6.5.: Beispielhafter Dienstplan

Dritte Version - Berücksichtigung der Wochentage

In der dritten Version des Algorithmus werden (zusätzlich zu den Priorisierungswerten) die Wochentage (Montag, Dienstag, ...) berücksichtigt. Der Klient kann Assistenten an bestimmten Wochentagen priorisieren (vgl. Abschnitt 6.10). Der Algorithmus geht genauso vor wie im vorherigen Abschnitt beschrieben. Zusätzlich wird jedoch der Wochentag berücksichtigt. Ist zum Beispiel Assistent 2 am Montag priorisiert, so werden die Werte an allen Montagen in der Punktetabelle verdoppelt (siehe Tabelle 6.6). Somit steigt die Wahrscheinlichkeit, dass Assistent 2 an einem Montag für den Dienst eingeteilt wird. In dem konkreten Beispiel wird Assistent 2 am Montag den Dienst bekommen, obwohl Assistent 3 eine generell höhere Priorisierung hat.

Tag des Monats	Punkte Assistent 1	Punkte Assistent 2	Punkte Assistent 3
1 (Montag)	0	40	30
2 (Dienstag)	1	20	0
3 (Mittwoch)	10	2	30
...

Tabelle 6.6.: Punktetabelle nach Berücksichtigung der Wochentage

Vierte Version - Berücksichtigung der Stichwörter

Der Algorithmus arbeitet prinzipiell wie im vorherigen Abschnitt beschrieben. Bei der Erstellung der Punktetabelle werden jedoch die privaten Bemerkungen im Monatsplan tagesweise nach Stichwörtern durchsucht. Ist ein Stichwort in der Team-Verwaltung bei einem Assistenten hinterlegt, so wird der Punktwert des Assistenten an dem betreffenden Tag verzehnfacht.

Zur Erläuterung ein Beispiel: Angenommen beim Assistenten 3 ist in der Team-Verwaltung das Stichwort „FCB“ hinterlegt. Wenn gleichzeitig bei den privaten Bemerkungen am ersten Tag des Monats ein „FCB“ enthalten ist, so wird der Punktwert des Assistenten von 30 auf 300 verzehnfacht (vergleiche auch Tabelle 6.7). Damit hat Assistent 3 die höchste Punktezahl für den ersten Tag des Monats und wird für den Dienst eingeteilt.



6. Implementierung

Tag des Monats	Private Bemerkungen	Assistent 1	Assistent 2	Assistent 3
1 (Montag)	FCB	0	40	300
2 (Dienstag)		1	20	0
3 (Mittwoch)		10	2	30
...

Tabelle 6.7.: Punktetabelle nach Berücksichtigung der Stichwörter

Fünfte Version - Berücksichtigung der Stundenkontingente

In der letzten Verfeinerung des Algorithmus werden die Stundenkontingente berücksichtigt. Bisher hat der Algorithmus über alle Tage iteriert und jeden Tag isoliert betrachtet. Für jeden Tag wurde ein lokales Optimum gefunden. In der Regel führt dieses Vorgehen allerdings zu einer sehr unausgewogenen Stundenverteilung unterhalb der Assistenten. Um die Stundenkontingente der Assistenten gleichmäßig auszuschöpfen ist es notwendig den gesamten Monat auf einmal zu betrachten.

Dazu werden die Daten in ein anderes Tabellenformat (siehe Tabelle 6.8) konvertiert. In die erste Spalte werden die Punkte (wenn sie größer Null sind) eingetragen, in der zweiten Spalte steht der Assistent und in der dritten Spalte der Tag.

Punkte	Name	Tag
1	Assistent 1	2
10	Assistent 1	3
40	Assistent 2	1
20	Assistent 2	2
2	Assistent 2	3
300	Assistent 3	1
30	Assistent 3	3

Tabelle 6.8.: Umgewandelte Punktetabelle

Die Punktetabelle wird anschließend randomisiert (durcheinandergewürfelt, gemischt), damit es nicht zur Blockbildung (ein Assistent hat mehrere Tage hintereinander Dienst) bei den Diensten kommt. Anschließend wird sie nach der ersten Spalte (dem Punktwert) sortiert (siehe Tabelle 6.9).

Weiterhin werden alle Stundenkontingente der Assistenten, die in der Team-Verwaltung (siehe Abschnitt 6.10) hinterlegt sind, zusammen addiert.

Dann werden alle Dienst- und Bereitschaftsstunden zusammen addiert. Sollte die Summe der Dienst- und Bereitschaftsstunden höher als die Summe der Stundenkontingente sein, so wird ein Skalierungsfaktor (Summe der Dienst- und Bereitschaftsstunden durch die Summe der Stundenkontingente) ermittelt und die Stundenkon-



6. Implementierung

Punkte	Name	Tag
300	Assistent 3	1
40	Assistent 2	1
30	Assistent 3	3
20	Assistent 2	2
10	Assistent 1	3
2	Assistent 2	3
1	Assistent 1	2

Tabelle 6.9.: Umgewandelte Punktetabelle, randomisiert und sortiert nach Punkten

tingente mit dem Skalierungsfaktor multipliziert, sprich die Stundenkontingente aller Assistenten werden um einen Prozentsatz x erhöht.

Nach den Vorbereitungsarbeiten wird der Kern des Algorithmus gestartet. Zu erst werden alle Dienste vergeben, die Bereitschaften erst später. Der Algorithmus iteriert über die, nach Punkten sortierte, Tabelle 6.9. In jeder Zeile der Tabelle prüft der Algorithmus ob der Assistent noch genug Stunden zur Verfügung hat und ob der Dienst an dem entsprechenden Tag noch frei ist. Falls beide Bedingungen erfüllt sind, bekommt der Assistent den Dienst eingetragen und seine verfügbaren Stunden werden um die Netto-Dienststundenanzahl des Tages verringert.

Durch die unterschiedlich langen Dienstzeiten von Tag zu Tag und unterschiedlichen Stundenkontingenten wird nach dem ersten Durchgang nicht für jeden Tag ein Assistent für den Dienst bestimmt werden können.

Folgendes Beispiel soll dies erläutern. Angenommen der Monat hätte nur 3 Tage. Die Dienstzeiten betragen 18, 16 und 14 Stunden. Die Summe der Dienstzeiten wäre 48. Es gibt 2 Assistenten mit Stundenkontingenten von 28 und 20 Stunden. Der Skalierungsfaktor wäre genau 1 und die Stundenkontingente der Assistenten würden nicht angepasst werden. Nach dem ersten Durchlauf (bei gleichen Priorisierungswerten) hätte der erste Assistent den 18 Stunden-Dienst und der zweite Assistent den 16 Stunden-Dienst. Der 14 Stunden-Dienst würde bei keinem der beiden Assistenten mehr ins Stundenkontingent passen.

Um die angesprochene Problematik zu lösen wird um den Kern des Algorithmus eine Schleife hinzugefügt, die einen Toleranzwert (beginnend bei Null Stunden) immer um eine Stunde inkrementiert. Bei der Prüfung, ob der Assistent noch genug Stunden zur Verfügung hat, wird der Toleranzwert zum noch verbleibenden Stundenkontingent hinzugefügt. Die hinzugefügte Schleife endet, sobald für jeden Tag ein Dienst vergeben wurde.

Angewendet auf das oben genannte Beispiel würde der die Schleife im fünften Durchlauf (Toleranz = 4 Stunden) dem ersten Assistenten den Dienst geben, da die ver-



bleibenden 10 Stunden (28 Stunden aus dem Stundenkontingent abzüglich der 18 Stunden des eingeteilten Dienstes) seines Kontingentes mit der Toleranz von 4 Stunden addiert genau die 14 Stunden Dienstzeit ergibt.

Nachdem alle Dienste vergeben wurden, werden auf dieselbe Weise die Bereitschaften vergeben. Einziger Unterschied ist, dass der Toleranzwert immer um eine halbe Stunde inkrementiert wird.

Durch das Randomisieren der Liste (siehe Tabelle 6.9) können mehrere Dienstpläne erstellt werden. Um von den vielen möglichen Dienstplänen einen zu wählen, der die Stundenkontingente möglichst gleichmäßig ausschöpft werden 1000 Dienstpläne erstellt und am Ende ermittelt wie gut jeder Dienstplan ist. Als Metrik dient die Summe der beiden Toleranzwerte (Dienst und Bereitschaft). Das theoretische Optimum ist Null. Der Dienstplan mit der niedrigsten Summe der Toleranzwerte wird dem Benutzer angezeigt.

Hier kommt evt. noch eine grafische Darstellung des Algorithmus als Ablaufdiagramm hin. Oder Pseudocode? Was würdest Du machen?

Performance des Algorithmus

Die Laufzeit des Algorithmus hängt stark von den möglichen Terminen der Assistenten ab. Die durchschnittliche Laufzeit einer Dienstplanerstellung liegt bei einer Millisekunde. Da 1000 Dienstpläne erstellt werden und der beste gewählt wird liegt die durchschnittliche Gesamtlaufzeit bei einer Sekunde. Da der Klient nur einmal im Monat einen Dienstplan erstellen lässt, ist diese „Wartezeit“ vertretbar.

Am Algorithmus ließen sich noch ein paar Dinge optimieren, aber nach dem Motto „Vorsicht vor Optimierungen“¹² von Clean Code Developer wurde vorerst davon abgesehen.

6.15. Aufgaben-Verwaltung

Dem Klienten und seinem Team steht eine Aufgaben-Verwaltung (siehe Abbildung 6.18) zur Verfügung.

¹²<http://www.clean-code-developer.de/Roter-Grad.ashx> - zuletzt abgerufen am 29.08.2014



6. Implementierung

Erledigt

☒ Einkaufen - fällig am 07.08.2014 - wiederholt sich alle 2 Tage ab dem Erledigungsdatum

Heute

☐ Spülen - wiederholt sich jeden Tag ab dem Erledigungsdatum

Übermorgen

☐ Einkaufen - wiederholt sich alle 2 Tage ab dem Erledigungsdatum

Zukünftig

☐ Fenster putzen - fällig am 20.09.2014 - wiederholt sich alle 6 Monate ab dem Erledigungsdatum

☐ Steuererklärung - fällig am 01.03.2015 - wiederholt sich jedes Jahr ab dem Fälligkeitsdatum

Abbildung 6.18.: Aufgaben-Verwaltung mit einigen Test-Aufgaben

Eigenschaften der Aufgaben

Die Aufgaben haben folgende Eigenschaften:

- Beschreibung
- Fälligkeitsdatum
- Wiederholungstyp (wiederholt sich nie, wiederholt sich ab Erledigungsdatum, wiederholt sich ab Fälligkeitsdatum)
- Wiederholungsintervall (z. B. alle 2 Tage) - vorausgesetzt, dass sich die Aufgabe wiederholt

Gruppierung der Aufgaben

Die Aufgaben werden an Hand ihres Fälligkeitsdatums in folgende Gruppen sortiert:

- Erledigt
- Heute
- Morgen
- Übermorgen
- Zukünftig
- Ohne Datum



Neue Aufgaben erstellen

Der Klient ist als Einziger in der Lage neue Aufgaben zu erstellen. Hierfür hat er eine Eingabemaske, wie sie in Abbildung 6.19 zu sehen ist.

Neue Aufgabe: fällig am wiederholt sich Tage ab dem

Abbildung 6.19.: Aufgaben-Verwaltung - Erstellen einer neuen Aufgabe

Bei der Auswahl des Fälligkeitsdatums wird er durch einen Kalender (siehe Abbildung 6.20) unterstützt. Dieser öffnet sich automatisch, sobald das Eingabefeld für das Datum den Fokus bekommt und schließt sich wieder, sobald der Klient ein Datum gewählt hat oder auf „Abbrechen“ klickt.

fällig am wiederholt sich

August 2014

Mo	Di	Mi	Do	Fr	Sa	So
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

Abbildung 6.20.: Aufgaben-Verwaltung: Wahl des Datums

Aufgaben als erledigt markieren

Sowohl Klient als auch Assistenten können Aufgaben als erledigt markieren. Hierfür wird die Checkbox links von der Aufgabenbeschreibung angeklickt. Die Aufgabe wird durchgestrichen und in die Kategorie „Erledigt“ sortiert. Falls die Aufgabe ein Wiederholungsintervall besitzt, wird eine Kopie der Aufgabe mit neu berechnetem Fälligkeitsdatum erstellt. Weiterhin wird protokolliert, wer die Aufgabe als erledigt markiert hat.

Hat man aus Versehen eine Aufgabe als erledigt markiert, kann man dies rückgängig machen, in dem man wieder die Checkbox links von der Aufgabenbeschreibung



anklickt. Die Aufgabe ist nicht mehr durchgestrichen, wird in die richtige Kategorie (Heute, Morgen, ...) zurücksortiert und die gegebenenfalls erstellte Kopie der Aufgabe (mit neu berechnetem Fälligkeitsdatum) wird gelöscht.

Speichern

Um Änderungen an den Aufgaben (neue Aufgaben erstellt oder Aufgaben erledigt) zu sichern, muss man auf den Knopf „Speichern“ drücken.

6.16. Dokumentation

Für die Benutzer des Assistenzplaners steht jederzeit im Navigations-Bereich (oben rechts) ein Link zur Verfügung, der zur Dokumentation führt. Der Dokumentations-Inhalt wird individuell erzeugt. Klienten bekommen eine andere Dokumentation zu Gesicht als Assistenten, da beide auf unterschiedliche Weise mit dem Assistenzplaner arbeiten. Weiterhin werden die Einstellungen berücksichtigt, z. B. wird das Dokumentations-Kapitel über die Aufgaben-Verwaltung nur angezeigt, wenn diese auch verwendet wird.

Im oberen Bereich der Dokumentation werden mit Hilfe von JavaScript automatisch Inhalts-, Abbildungs- und Tabellenverzeichnisse erstellt.

6.17. Impressum

Um eine Abmahnung zu vermeiden wurde dem Assistenzplaner ein Seite hinzugefügt, die jederzeit über die Navigation zu erreichen ist. Diese beinhaltet das Impressum mit allen Angaben gemäß § 5 Telemediengesetz, einen Haftungsausschluss (Disclaimer) sowie eine Datenschutzerklärung.



7. Überführung in Produktiv-Betrieb

Die Überführung in den Produktiv-Betrieb wurde iterativ durchgeführt.

Den Dienstplan für den Monat April 2014 hat der Klient noch komplett manuell erstellt.

Schon bei der Dienstplanerstellung für den Mai 2014 wurde der Klient technisch unterstützt. Es war möglich die Termin-Antworten der Assistenten übersichtlich in einer Tabelle darzustellen. Somit musste der Klient nicht mehr sieben E-Mail Antworten gleichzeitig überblicken. Es war möglich Dienste und Bereitschaften manuell einzutragen und den Dienstplan zu speichern. Der Autor war bei der Dienstplanerstellung dabei und hat viel über die Hintergründe erfahren und herausfinden können worauf es dem Klienten ankommt.

Bei der Dienstplanerstellung für Juni 2014 hat schon ein Algorithmus im Hintergrund gearbeitet und Dienstplanvorschläge erstellt. Der Klient hat den Dienstplan eigenständig in Anwesenheit des Autors erstellt.

Bei der Dienstplanerstellung für Juli 2014 waren alle nötigen Komponenten vorhanden. Zur Sicherheit haben der Klient und der Autor nochmal gemeinsam den Dienstplan erstellt und verifiziert, dass alle Komponenten ordnungsgemäß funktionieren.

Ab dem Dienstplan für den Monat August 2014 haben der Klient und die Assistenten ihre Eingaben selbstständig getätigt. Es war kein Zutun des Autors mehr notwendig. Der Algorithmus zur Dienstplanerstellung hat einen ausgewogenen Dienstplan erstellt, der vom Klienten angenommen wurde.

Durch die iterative Vorgehensweise hat sich der Klient sehr gut eingebunden gefühlt. Er konnte in regelmäßigen Intervallen den aktuellen Stand der Software sehen und sein Feedback äußern, was direkt in die weitere Entwicklung eingeflossen ist.



8. Ergebnis

Der **Assistenzplaner**¹³ wurde als Webanwendung neu erschaffen. Er ist unter anderem in der Lage

- Assistenz-Teams zu verwalten
- Dienstpläne zu erstellen
- Aufgaben für die Assistenz-Teams zu verwalten

Die Webanwendung wurde mit vielen gängigen Browsern (Firefox, Chrome, Safari) erfolgreich getestet.

¹³<http://assistenplaner.de/master>



9. Fazit und Ausblick

Es wurde eine Webanwendung erschaffen, die dem Klienten sehr viel Arbeit abnimmt. Der Klient ist mit der Lösung sehr zufrieden und wird in Zukunft nur noch den Assistenzplaner zur Dienstplanerstellung verwenden. Alle gewünschten Benutzergeschichten (siehe Abschnitt 3.4) wurden umgesetzt.

Die entstandene Lösung ist voll funktionsfähig und in sich geschlossen. Dennoch gibt es einige Stellen, bei denen es Verbesserungs- oder Erweiterungspotential gibt.

Der Autor plant, das Projekt weiter zu führen und den **Assistenzplaner**¹⁴ kontinuierlich zu verbessern und zu erweitern. Die Quellen des Assistenzplaners sind Open-Source und in einem öffentlichen Repository bei **GitHub**¹⁵ eingchecked. So hat jeder die Möglichkeit sich den Assistenzplaner auf dem eigenen Server einzurichten.

9.1. Mehr Teams

Der Assistenzplaner ist derzeit (September 2014) eine spezialisierte Lösung für einen Klienten mit seinem Team. Der Autor hatte am 01.09.2014 ein Gespräch mit dem **Landesbeauftragten für die Belange behinderter Menschen in Rheinland Pfalz - Matthias Rösch**¹⁶. Hier hat er den Assistenzplaner vorgestellt und sich vom Landesbeauftragten einige Ansprechpartner im Bereich Assistenzplanung nennen lassen. Der Landesbeauftragte hat selber sechs Assistenten, erstellt derzeit seine Dienstpläne manuell und war sehr interessiert am Assistenzplaner.

Der Autor hofft, dass sich noch ein paar Beta-Tester bereit erklären den Assistenzplaner einzusetzen. Sicherlich werden auch noch einige Anforderungen hinzukommen, da jeder Klient mit seinem Team einen anderen Prozess und verschiedene Rahmenbedingungen bei der Dienstplanerstellung hat.

Sobald der Assistenzplaner von einigen Klienten mit ihren Teams erfolgreich erprobt ist, möchte der Autor den Assistenzplaner öffentlich bewerben¹⁷. Einige mögliche Kanäle sind nachfolgend aufgezählt:

¹⁴<http://assistenzplaner.de>

¹⁵<https://github.com/Ulle84/AssistancePlaner>

¹⁶<http://inklusion.rlp.de/der-landesbeauftragte/> - zuletzt abgerufen am 02.09.2014

¹⁷Der Autor hat keinerlei finanzielles Interesse. Die Nutzung des Assistenzplaner ist kostenlos und wird es auch immer bleiben!



- [Blog von Matthias Rösch, Landesbeauftragter für die Belange behinderter Menschen in Rheinland-Pfalz](#)¹⁸
- [kobinet Nachrichten - Online Nachrichten von und für behinderte Menschen](#)¹⁹
- [ForseA - Forum selbstbestimmter Assistenz behinderter Menschen e.V.](#)²⁰
- Flyer im [ZSL Mainz](#)²¹, [ZSL Bad Kreuznach](#)²², [Commit Mainz](#)²³ und weiteren Einrichtungen hinterlegen
- Nachricht über den E-Mail-Verteiler des Landesbeauftragten

9.2. Mögliche Verbesserungen

In diesem Abschnitt werden mögliche Verbesserungen des Assistenzplaners vorgestellt und diskutiert.

Speicherung in Datenbank

Derzeit werden die Daten persistent in Text-Dateien gespeichert. Langfristig sollte über eine Speicherung in einer Datenbank nachgedacht werden. Denkbar wäre, alle Daten in einer großen Datenbank zu speichern. Eine weitere Möglichkeit wäre, pro Klient eine dateibasierte SQLite Datenbank anzulegen. Die zweite Möglichkeit hat den Vorteil, dass sich die Daten leichter zwischen Entwickler-System und Produktiv-System austauschen lassen und keinerlei Anforderungen an den Server (bezüglich einer Datenbank) gestellt werden.

Barrierefreiheit

„Barrierefreiheit bezeichnet im deutschen Sprachgebrauch eine Gestaltung der baulichen Umwelt in der Weise, dass sie von Menschen mit Behinderung in der selben Weise genutzt werden kann wie von Menschen ohne Behinderung.“ [Bremus 2013, S. 28]

¹⁸<http://inklusion-blog.rlp.de> - zuletzt abgerufen am 02.09.2014

¹⁹<http://www.kobinet-nachrichten.org> - zuletzt abgerufen am 02.09.2014

²⁰<http://www.forsea.de> - zuletzt abgerufen am 02.09.2014

²¹<http://www.zsl-mz.de> - zuletzt abgerufen am 02.09.2014

²²<http://www.zsl-bad-kreuznach.org> - zuletzt abgerufen am 02.09.2014

²³<http://www.commit-cbf.de> - zuletzt abgerufen am 02.09.2014



„Barrierefreiheit als Thema für Webentwickler und -designer wird auch deshalb immer wichtiger und interessanter, weil die Menschen eine immer höhere Lebenserwartung haben, und gerade Menschen mit Behinderungen immer besser medizinisch versorgt werden können.“ [Bremus 2013, S. 29]

„Eine barrierefreie Webanwendung basiert grundsätzlich auf zwei Säulen: der Accessibility und der Usability“ [Bremus 2013, S. 30]

„Unter Accessibility ist die Zugänglichkeit bzw. Erreichbarkeit in Bezug auf eine Anwendung zu verstehen. Sie beschreibt die Fähigkeit, Informationen für jeden Benutzer zugänglich zu machen, unabhängig von technischen und körperlichen Voraussetzungen und Einschränkungen.“ [Bremus 2013, S. 31]

„Die Usability strebt nach der idealen Strukturierung von Informationen, um eine effiziente Nutzung von Informationen und Technologie überhaupt erst zu ermöglichen.“ [Bremus 2013, S. 31]

Da der Assistenzplaner langfristig von vielen körperlich behinderten Menschen bedient werden soll, ist eine Untersuchung der entstandenen Softwarelösung auf gute Accessibility und Usability notwendig. Es muss verifiziert werden dass die Seite im Idealfall barrierefrei, mindestens jedoch barrierearm²⁴ ist.

AJAX tiefer verwurzeln

Derzeit wird AJAX nur auf Knopfdruck eingesetzt. An vielen Stellen muss der Benutzer explizit auf einen Speichern-Button drücken um die eingegebenen Daten persistent zu sichern. In einer moderenen Webanwendung ist dies ungewöhnlich und kann schon als benutzerunfreundlich bezeichnet werden. Besser wäre es, wenn Änderungen, die der Benutzer vornimmt, direkt per AJAX an den Server geschickt werden und der Server die Daten direkt persistent ablegt.

Responsive Webdesign

Responsive (reaktionsfähiges) Webdesign bedeutet auf „die steigende Anzahl verschiedener Display-Größen und Geräte angemessen zu reagieren“ [Zillgens 2013, S. 8].

²⁴ „Verstärkt setzt sich in der letzten Zeit der Begriff des barrierearmen Webdesigns durch. Das beruht auf der Tatsache, dass eine 100%-ige Barrierefreiheit einer Webanwendung nicht zu erreichen ist. Verschiedene Benutzerinteressen, Hilfsmittel und technische Voraussetzungen machen die Barrierefreiheit einer Anwendung unmöglich“ [Bremus 2013, S. 29]



Auf Desktops und Tablets wird der Assistenzplaner gut dargestellt und kann gut bedient werden. Auf Smart-Phones ist das Aussehen und die Bedienung verbesserungswürdig (siehe Abbildung 9.1)

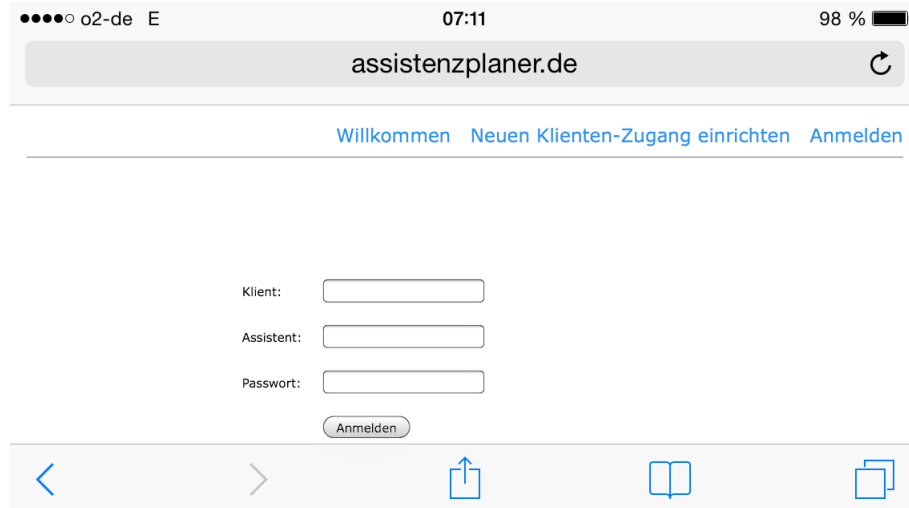


Abbildung 9.1.: Schlechte Darstellung des Assistenzplaner auf einem Smart-Phone. Die Eingabefelder sind zu klein und die Display-Größe ist schlecht ausgenutzt.

Bei der Optimierung des Assistenzplaners für Handys muss umgedacht werden. Man muss wegkommen von dem horizontal-basierten Layout mit Spalten hin zu einem linear-vertikalen Layout. [Bieh 2008, S. 67].

Team-Übersicht überarbeiten

Derzeit wird das Team mit Hilfe einer Tabelle verwaltet. Zu Beginn der Entwicklung war das die richtige Entscheidung. Im Laufe der Entwicklung sind aber immer mehr Spalten hinzugekommen und die Tabelle ist mit der Zeit immer breiter und unübersichtlich geworden (siehe Abbildung 6.9). Eine mögliche Alternative zur breiten Tabelle ist in Abbildung 9.2 gezeigt. Für jeden Assistenten gibt es eine eigene Übersicht der Eigenschaften.

Derzeit hat nur der Klient Einblick in die Team-Verwaltung. Es wäre wünschenswert, wenn Assistenten untereinander ihre Kontaktdaten (die ersten vier Zeilen in Abbildung 9.2) einsehen könnten.

Überarbeitung der Dokumentation

Derzeit ist der Assistenzplaner für Klienten und Assistenten auf einer zentralen Seite (siehe Abschnitt 6.16) dokumentiert. Wünschenswert wäre es, wenn kontextsensitive



Markus

Vorname	Markus
Nachname	Mustermann
E-Mail Adresse	markus@mustermann.de
Telefonnummer	0800-555000
Stichwörter	FCB
Stundenkontigent	200
Priorisierung	10
Bevorzugte Tage	<input type="checkbox"/> Mo <input type="checkbox"/> Di <input type="checkbox"/> Mi <input type="checkbox"/> Do <input type="checkbox"/> Fr <input type="checkbox"/> Sa <input type="checkbox"/> So
Aktionen	<input type="button" value="Editieren"/> <input type="button" value="Löschen"/> <input type="button" value="Passwort zurücksetzen"/>

Abbildung 9.2.: Mögliche neue Darstellungsform eines Assistenten in der Team-Übersicht

Informationen zusätzlich über den gesamten Assistenzplaner an geeigneter Stelle leicht zugänglich verteilt wären. Eine Möglichkeit der Realisierung besteht mit Hilfe von **jQuery UI Tooltip**²⁵ (siehe Abbildung 9.3).

Tooltip

Customizable, themeable tooltips, replacing native tooltips.

Firstname

Lastname Please provide also your lastname.

Address

Abbildung 9.3.: Sobald der Benutzer in das Feld „Lastname“ klickt, bekommt er einen Hilfstext (Tooltip) angezeigt. Screenshot wurde am 22.08.2014 auf der Webseite <http://jqueryui.com/tooltip/> aufgenommen.

Verwendung von Frameworks

Derzeit ist alles „zu Fuß programmiert“. Der Autor hat sich bewusst gegen den Einsatz von Frameworks entschieden, um den Einarbeitungsaufwand gering zu halten.

²⁵<http://jqueryui.com/tooltip/> - zuletzt abgerufen am 22.08.2014



Unter dem Gesichtspunkt, dass der Autor langfristig evt. nicht der einzige Entwickler des Assistenzplaners sein wird, wäre es wünschenswert auf Frameworks aufzubauen, die verbreitet sind. Ein Beispiel für ein weit verbreitetes JavaScript-Framework ist [jQuery](http://jquery.com)²⁶. Ein weiterer Vorteil wäre, dass die Abdeckung aller gängigen Browser garantiert wäre.

Serverseitig sei das PHP-Framework [Symfony2](http://symfony.com)²⁷ erwähnt, dass das Prinzip des *Model View Controller* (MVC) konsequent einsetzt.

9.3. Mögliche Erweiterungen

In diesem Abschnitt werden mögliche Erweiterungen des Assistenzplaners vorgestellt und diskutiert.

Mehrere Dienste pro Tag

Derzeit ist nur ein Dienst pro Tag planbar. Um noch mehr Klienten zu erreichen wäre es sinnvoll, wenn man mehrere Dienste pro Tag planen könnte.

Internationalisierung

Derzeit ist der Assistenzplaner komplett in Deutsch. Um auch andere Sprachen zu unterstützen wäre ein grundlegender Umbau notwendig. Weiterführende Informationen finden sich in der offiziellen [PHP-Dokumentation](http://php.net/manual/en/book.intl.php)²⁸

iCalendar Export

Im iCalendar-Format können Termine ausgetauscht werden. Ein Beispiel für ein Ereignis im iCalendar Format:

Listing 9.1: Beispiel für ein iCalendar-Termin.

```
1 BEGIN:VCALENDAR
2 VERSION:2.0
3 PRODID:http://www.assistenzplaner.de/
4 METHOD:PUBLISH
5 BEGIN:VEVENT
6 UID:ae993c01a0c40fb4a93b9a79363502df@assistenzplaner.de
```

²⁶<http://jquery.com> - zuletzt abgerufen am 29.08.2014

²⁷<http://symfony.com> - zuletzt abgerufen am 29.08.2014

²⁸<http://php.net/manual/en/book.intl.php> - zuletzt abgerufen am 21.08.2014



9. Fazit und Ausblick

```
7 ORGANIZER;CN="Klient":MAILTO:info@assistenzplaner.de
8 LOCATION:Heim des Klienten
9 SUMMARY:Assistenz/Bereitschaft
10 DESCRIPTION:Assistenz/Bereitschaft + Notizen
11 CLASS:PUBIC
12 DTSTART:20140822T140000Z
13 DTEND:20140823T090000Z
14 DTSTAMP:20140822T064500Z
15 END:VEVENT
16 END:VCALENDAR
```

In dem Ereignis ist der Dienst vom 22.08.2014 14:00 Uhr bis 23.08.2014 9:00 Uhr beschrieben.

Für die Assistenten wäre es interessant ihre Dienst- und Bereitschaftszeiten im iCalendar-Format abonnieren zu können. So hätten die Assistenten jederzeit aktuell ihre Arbeitszeiten, ohne ständig online auf dem Assistenzplaner nachschauen zu müssen. Ein iCalendar lässt sich leicht in den eigenen Online-Kalender integrieren, so dass sich die Dienst- und Bereitschaftszeiten jederzeit von unterwegs auf dem Smart-Phone einsehen lassen.

Feiertage importieren

Für die Erstellung des Monatsplans wäre es für den Klienten hilfreich, wenn man Feiertage importieren könnte. Im Idealfall würde sogar das Bundesland, in dem der Klient lebt, berücksichtigt werden. Zum Beispiel könnte man auf der Webseite **iFeiertage**²⁹ einen Feiertage-Kalender abonnieren. Dieser wäre als iCalendar verfügbar:

```
webcal://ifeiertage.de/rp-.ics
```

Denkbar wäre, dass der Klient beim Monatsplan einen Button „Feiertage importieren“ zur Verfügung gestellt bekommt. Betätigt der Klient diesen Knopf, so werden alle Feiertage für diesen Monat in den öffentlichen Notizen eingetragen. Weiterhin werden die Dienstzeiten auf 24-stündige Dienste angepasst, da der Klient an Feiertagen rund um die Uhr Assistenz hat.

Logging

Derzeit ist nicht nachvollziehbar, wer wann was am Assistenzplaner gemacht hat. Für zukünftigen Support und Fehlersuche wäre es wünschenswert ein Log-File zu

²⁹<http://www.ifeiertage.de> - zuletzt abgerufen am 21.08.2014



haben, in dem man sehen kann, welche Änderungen in der letzten Zeit gemacht wurden und von wem diese durchgeführt wurden.

Aufgaben-Verwaltung erweitern

Die Aufgaben-Verwaltung wurde nur rudimentär entwickelt. Schon zu einem frühen Zeitpunkt hat der Klient entschieden die Aufgaben-Verwaltung für sein Team nicht einzusetzen, da der Mehraufwand durch die Verwaltung zu groß wäre. Aufgrund dieser Entscheidung wurde die Entwicklung schon nach einer Woche wieder eingestellt. Für andere Modelle in der Pflege wäre es aber evt. von Vorteil die Aufgaben-Verwaltung zu nutzen. Mögliche Erweiterungen werden in den folgenden Abschnitten dargestellt.

Hervorhebungen bei Datum-Auswahl

In der Datum-Auswahl (siehe Abbildung 6.20) sollte der aktuelle Tag sowie der gewählte Tag farblich hervorgehoben werden.

Aufgaben editieren

Aufgaben lassen sich derzeit nur anlegen, nicht im Nachhinein verändern. Eine Editierfunktion wäre von großem Nutzen.

Aufgabe löschen

Aufgaben lassen sich derzeit nur anlegen, aber nicht löschen.

Übersicht über erledigte Aufgaben

Von Vorteil (gerade für die Leistungserbringungs-Nachweise) wäre es, wenn man nachvollziehen könnte, welcher Assistent wann welche Aufgabe erledigt hat.

So, mein liebster Masterarbeits-Buddy ;-) Hast es geschafft, der Rest ist automatisch generiert bzw. nicht vollständig. Vielen Dank fürs Durchkämpfen! :-)



10. Literaturverzeichnis

Bieh 2008

BIEH, M.: *Mobiles Webdesign: Konzeption, Gestaltung, Entwicklung ; [Struktur, Design und Programmierung ; Umsetzung mit (X)HTML, CSS und PHP ; Standards und Best Practices]*. Galileo Press, 2008 (Galileo computing). – ISBN 9783836211536 9.2

Bremus 2013

BREMUS, T: *Barrierefreiheit: Webanwendungen ohne Hindernisse*. entwickler Press, 2013. – ISBN 9783868020953 9.2, 24

Comer 2002

COMER, D.: *Computernetzwerke und Internets: mit Internet-Anwendungen*. Pearson Studium, 2002 (I Informatik). – ISBN 3-8273-7023-X 2.1, 2.1, 2.1, 2.1, 2.1, 2.1, 2.1, 2.1

Flanagan 2002

FLANAGAN, D.: *JavaScript: das umfassende Referenzwerk*. O'Reilly, 2002. – ISBN 3-89721-330-3 2.2

Koch 2011

KOCH, S.: *JavaScript: Einführung, Programmierung und Referenz*. dpunkt.Verlag, 2011 (iX-Edition). – ISBN 9783898647311 2.2

Münz und Gull 2012

MÜNZ, S. ; GULL, C.: *HTML5-Handbuch*. Franzis, 2012 (Know-how ist blau). – ISBN 9783645601511 2.2, 2.2

Pichler 2008

PICHLER, R.: *Scrum: agiles Projektmanagement erfolgreich einsetzen*. dpunkt.Verlag, 2008. – ISBN 978-3-89864-478-5 3.4

Rupp u. a. 2012

RUPP, C. ; QUEINS, S. ; SOPHISTEN, die: *UML 2 glasklar: Praxiswissen für die UML-Modellierung*. Carl Hanser Verlag, 2012. – ISBN 978-3-446-43057-0 2.3



Schlossnagle 2006

SCHLOSSNAGLE, G.: *Professionelle PHP 5-Programmierung - Entwicklerleitfaden für große Webprojekte mit PHP 5*. Addison-Wesley, 2006 (open source library). – ISBN 978-3-8273-2381-1 6.3

Teague 2011

TEAGUE, J.C.: *CSS3*. Peachpit Press, 2011 (Visual quickstart guide). – ISBN 9780321719638 2.2

Theis 2010

THEIS, T.: *Einstieg in PHP 5.3 und MySQL 5.4*. Galileo Press, 2010 (Galileo computing). – ISBN 9783836215442 6.1

Wirdemann 2011

WIRDEMANN, R.: *Scrum mit User Stories*. Carl Hanser Verlag, 2011. – ISBN 978-3-446-42660-3 3.4

Zillgens 2013

ZILLGENS, C.: *Responsive Webdesign Reaktionsfähige Websites gestalten und umsetzen*. Carl Hanser Verlag, 2013. – ISBN 978-3-446-43015-0 9.2



A. Tabellenverzeichnis

6.1. Übersicht der verwendeten Daten aus der Team-Verwaltung	49
6.2. Punktetabelle zur Dienstplanerstellung	50
6.3. Priorisierungswerte aus der Team-Verwaltung	50
6.4. Punktetabelle nach Multiplikation mit Priorisierungswerten	50
6.5. Beispielhafter Dienstplan	51
6.6. Punktetabelle nach Berücksichtigung der Wochentage	51
6.7. Punktetabelle nach Berücksichtigung der Stichwörter	52
6.8. Umgewandelte Punktetabelle	52
6.9. Umgewandelte Punktetabelle, randomisiert und sortiert nach Punkten	53



B. Abbildungsverzeichnis

2.1. Ergebnis Zusammenspiel HTML, CSS und JavaScript	17
4.1. Screenshot QTime	32
5.1. SysML Anforderungsdiagramm	34
5.2. UML Komponentendiagramm	35
5.3. UML Klassendiagramm	36
6.1. Session-ID als Cookie im Browser	39
6.2. Willkommen-Seite	39
6.3. Navigation bei angemeldetem Assistenten	40
6.4. Navigation ohne Anmmeldung	40
6.5. Anlegen eines neuen Klienten	40
6.6. Übersicht Assistenzplaner für Klienten	41
6.7. Einstellungsmöglichkeiten des Assistenzplaners	42
6.8. Standard-Dienstzeiten	42
6.9. Verwaltung Assistenz-Team	43
6.10. Monatsplan	44
6.11. Benachrichtigung der Assistenten	44
6.12. Kalender für die Eingabe möglicher Termine	45
6.13. Bemerkungen des Klienten	45
6.14. Editierbarer Dienstplan	46
6.15. Stundenübersicht Assistenten	47
6.16. Gesamter Dienstplan aus Sicht des Assistenten	48
6.17. Dienste und Bereitschaften des angemeldeten Assistenten	48
6.18. Aufgaben-Verwaltung	55
6.19. Erstellen einer neuen Aufgabe	56
6.20. Wahl des Datums	56
9.1. Schlechte Darstellung auf Smart-Phone	63
9.2. Mögliche neue Darstellungform des Teams	64
9.3. Beispiel Tooltip	64



C. Verzeichnis der Listings

2.1. Beispiel für ein XML Dokument.	13
2.2. Beispiel für ein HTML-Dokument.	14
2.3. Beispiel für ein CSS-Dokument.	15
2.4. Beispiel für ein JavaScript-Dokument.	16
2.5. Beispiel für die Erweiterung eines bestehenden Objekts in JavaScript	16
2.6. Beispiel für ein PHP-Dokument.	18
9.1. Beispiel für ein iCalendar-Termin.	65



D. Abkürzungsverzeichnis

AJAX	Asynchronous JavaScript and XML
CSS	Cascading Stylesheets
DOM	Document Object Model
HTML	HyperText Markup Language
HTTP	Hypertext Transport Protocol
ID	Identifikator
MVC	Model View Controller
PDF	Portable Document Format
PHP	PHP Hypertext Preprocessor
SysML	System Modeling Language
UML	Unified Modeling Language
URL	Uniform Resource Locator
XML	Extensible Markup Language



E. Inhalt der beiliegenden CD

* Ordnerstruktur erklären



F. Hintergrundinformationen

F.1. Verzeichnisstruktur auf Server

*PHP *HTML *JavaScript * Data ...

F.2. Verwendete Software

In diesem Abschnitt wird erläutert, welche Software eingesetzt wurde.

* PHP 5.5.14 -> <http://assistenzplaner.de/PHP/phpInfo.php> * PhpMailer * FPDF

* PhpStorm * Enterprise Architect 11.1 Build 1110 von **Sparx Systems**³⁰ -> SysML Erweiterung? * GitHub * Cyberduck (FTP-Client)

* Firefox, Safari, FireBug

³⁰<http://www.sparxsystems.de> - zuletzt abgerufen am 22.08.2014