



6. Implementierung

In diesem Kapitel wird dargestellt, wie die Anforderungen und das Software-Design konkret implementiert wurden. Die Ergebnisse der Implementierung werden kompakt dargestellt. In den Abschnitten „Implementierungsdetails“ werden weiterführende Informationen dargestellt, die für das Gesamtverständnis der Ausarbeitung nicht wichtig sind und übersprungen werden können. Der Schwerpunkt des Kapitels liegt auf der Darstellung des Algorithmus zur Dienstplanerstellung (siehe Abschnitt 6.16).

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**¹³ 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.

¹³http://w3techs.com/technologies/history_overview/programming_language - zuletzt abgerufen am 23.08.2014



6.2. Einheitliches Layout

Mit Hilfe von CSS wird die Darstellung des Assistenzplaners konsistent definiert. Es gibt ein globales Stylesheet, was in allen Webseiten verwendet wird.

Implementierungsdetails

In der Datei `global.css` sind alle globalen Styles definiert.

6.3. 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.

Implementierungsdetails

Eine potentielle Fehlerquelle beim Gebrauch von textbasierten Daten ist der gleichzeitige Schreibzugriff zweier Benutzer auf eine Datei. Das ist derzeit nur bei den Kalender-Eingaben der Assistenten (siehe Abschnitt 6.14) möglich. Um den gleichzeitigen Zugriff zu verhindern wird eine Art **Semaphor**¹⁴ 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.

Es gibt einige PHP-Skripte zum Speichern von Daten. Bei allen werden per POST-Parameter die zu speichernden Inhalte übermittelt, ein File schreibend geöffnet und der übertragene Inhalt geschrieben. Die PHP-Skripte, die für das Speichern zuständig sind, heißen `calendarSaver.php`, `changePasswordSaver.php`, `defaultTimesSaver.php`, `monthPlanSaver.php`, `rosterSaver.php`, `settingsSaver.php` und `todoSaver.php`.

¹⁴[http://de.wikipedia.org/wiki/Semaphor_\(Informatik\)](http://de.wikipedia.org/wiki/Semaphor_(Informatik)) - zuletzt abgerufen am 05.09.2014



6.4. 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 (siehe Abbildung 6.1). Es gibt zwei Rollen von Benutzern: die Klienten und die Assistenten. Die Klienten lassen beim Anmelden das Eingabefeld „Assistent“ leer. Die Klienten können sich ihren Zugang selbstständig anlegen (siehe Abschnitt 6.8). Die Zugänge der Assistenten werden von den Klienten verwaltet.¹⁵ Klienten haben die Möglichkeit Assistenten zum Team hinzuzufügen und zu entfernen (siehe Abschnitt 6.12).

Klient:

Assistent:

Passwort:

Abbildung 6.1.: Login-Maske des Assistenzplaners

Implementierungsdetails

Das PHP-Skript `login.php` ist für die Erstellung der Eingabefelder zuständig. Wenn der Benutzer auf den Button „Anmelden“ drückt, werden die Daten aus den Eingabefeldern nach einer erfolgreichen Validierung per `POST` an den `loginHandler.php` geschickt. Der Login-Handler verifiziert, dass der Nutzer besteht und das Passwort korrekt eingegeben wurde.

Nach der erfolgreichen 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 6.2.

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.

¹⁵ Jeder Assistent ist damit fest einem Klienten zugeteilt. Theoretisch kann ein Assistent auch in mehreren Teams Mitglied sein.

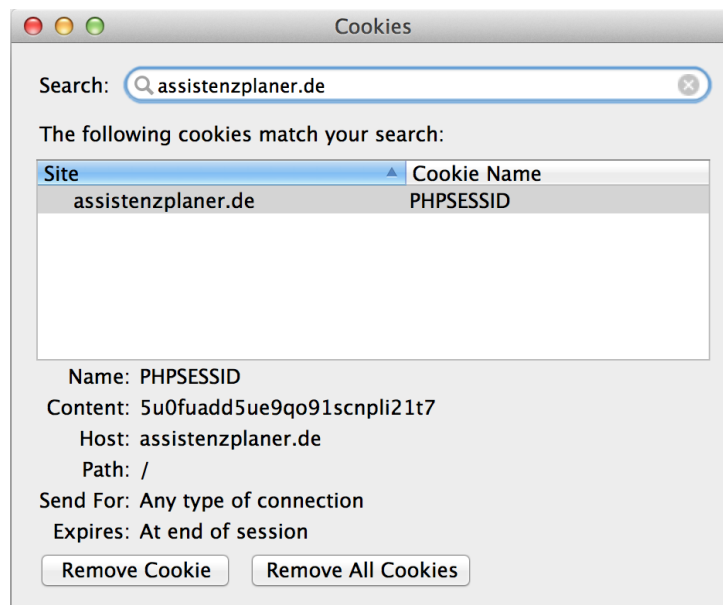


Abbildung 6.2.: Session-ID wird als Cookie im Browser gespeichert

Falls sich der Assistent mit dem Standard-Passwort angemeldet hat, so wird er zu `changePassword.php` weitergeleitet. Dort wird er aufgefordert sein Passwort zu ändern. Ansonsten wird er zur Übersicht (siehe Abschnitt 6.9) weitergeleitet. Eine weitere Möglichkeit ist, dass die Login-Seite mit dem GET-Parameter `redirect` aufgerufen wurde. Das passiert zum Beispiel, wenn der Assistent in der automatisch generierten Nachricht auf einen Link klickt, der auf folgendes Ziel verweist:

`http://assistenzplaner.de/PHP/login.php?redirect=calendarView`

Dann wird der Nutzer nach erfolgreicher Anmeldung direkt zur Kalender-Ansicht weitergeleitet.

6.5. Willkommen-Seite

Auf der **Willkommen-Seite**¹⁶ (siehe Abbildung 6.3) werden dem Interessenten Informationen über den Assistenzplaner präsentiert. Man wird gebeten einen neuen Klienten-Zugang einzurichten oder sich mit einem bestehenden Zugang anzumelden.

¹⁶<http://www.assistenzplaner.de/master/PHP/index.php> - zuletzt abgerufen am 23.08.2014



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 6.3.: Willkommen-Seite

Implementierungsdetails

Die Willkommenseite wird durch das PHP-Script `index.php` generiert. Man kann zwei Aktionen durchführen:

1. Ein neues Konto anlegen \Rightarrow `createNewAccount.php`
2. Anmelden \Rightarrow `login.php`

6.6. Navigation

Die Navigation steht immer gut sichtbar rechts oben zur Verfügung. Abhängig davon, ob jemand am System angemeldet ist (siehe Abbildung 6.4) oder nicht (siehe Abbildung 6.5) stellt sie unterschiedliche Informationen dar.

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

Abbildung 6.4.: Navigation bei angemeldetem Assistenten

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

Abbildung 6.5.: Navigation ohne Anmmeldung

Implementierungsdetails

Die Navigation wird durch das PHP-Script `navigation.php` erzeugt. Dieses Script wird in allen Seiten direkt nach dem HTML-body-Tag inkludiert¹⁷. Im Script werden aus dem superglobalen Array `$_SESSION` einige Informationen ausgelesen und abhängig davon der Inhalt der Navigation variiert. In der Navigation werden Hyperlinks zu anderen PHP-Scripten dargestellt.

¹⁷PHP-Code: `<?php include('navigation.php'); ?>`



6.7. Monatsnavigation

An einigen Stellen (Dienstplan, Monatsplan, Kalender) möchte der Anwender zwischen den Monaten navigieren können. Mithilfe der Monatsnavigation (siehe Abbildung 6.6) kann man vom derzeit angezeigten Monat immer zum vorherigen, aktuellen und nächsten Monat navigieren.

[Vorheriger Monat](#) [Aktueller Monat](#) [Nächster Monat](#)

Abbildung 6.6.: Monatsnavigation

Implementierungsdetails

Die Monatsnavigation wird von der PHP-Klasse `MonthNavigation` erstellt. Es reicht eine Instanz von `MonthNavigation` zu erstellen. Bei der Konstruktion muss man nur einen Pfad auf das aufrufende PHP-Script, das Jahr und den Monat angeben. Voraussetzung ist, dass das aufrufende PHP-Script mit den `GET`-Parametern `year` und `month` aufgerufen werden kann. Das ist bei den PHP-Scripten `monthPlanView.php`, `rosterView.php` und `calendarView.php` der Fall.

6.8. Anlegen eines neuen Klienten-Zugangs

Das Anlegen eines neuen Klienten-Zugangs soll auf einfache Weise möglich sein. Es werden bei der Erstellung eines neuen Klienten-Zugangs keine persönlichen Daten erhoben.

Um einen neuen Klienten-Zugang anzulegen, wird man gebeten einen Klientennamen, ein Passwort und die Passwort-Wiederholung einzugeben (siehe Abbildung 6.7).

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

Abbildung 6.7.: Anlegen eines neuen Klienten mit Name und Passwort

Bedingungen an einen neuen Klientennamen sind:

- Er darf nicht leer sein.