

Dienstplan Apotheke Dokumentation

Martin Mandelkow

13. April 2019

Inhaltsverzeichnis

1	Einführung	3
1.1	PDR finden	3
1.2	Lizenz	3
1.3	Fehler melden	4
1.4	Wie Sie einen Beitrag leisten können	4
2	Benutzerhandbuch	5
2.1	Das Web-Interface	5
2.1.1	Login	5
2.1.2	Neuen Benutzer-Account erstellen	5
2.1.3	Passwort vergessen	6
2.1.4	Navigation	6
2.1.5	Dienstplan Wochenansicht	7
2.1.6	Dienstplan Tagesansicht	8
2.1.7	Dienstplan Mitarbeiteransicht	10
2.1.8	Grundplan Tagesansicht	10
2.1.9	Überstunden	10
2.1.10	Abwesenheit	10
2.2	Kalender API	11
2.2.1	ICalendar-Dateien automatisch mit iCal Import/Export CalDAV Proim- portieren	11
3	Administratorhandbuch	13
3.1	Installation	13
3.1.1	PDR finden	13
3.1.2	Die Installationsroutine	13
3.1.3	Erste Schritte	14
3.2	Aktualisierung	15
3.3	Konfiguration	15
3.4	Wartung	17
3.4.1	Klasse maintenance	17
3.4.2	Klasse update_database	17
3.5	Probleme und Fehlerbehebung	17
4	Entwicklerhandbuch	18
4.1	Kernentwicklung	18
4.1.1	Ordnerstruktur	20
4.1.2	Programmierstil	21
4.1.3	Die Datenbank	22
4.1.4	Klassen	23
4.1.5	Web-Interface	24

4.1.6	Kalender API	24
4.2	Dokumentation	24
4.3	Testen	24
4.4	Bug Tracker	24
4.5	Übersetzung	25
4.5.1	Internationalisierung	25

Kapitel 1

Einführung

Pharmacy Duty Roster (PDR) ist eine Webanwendung, die es ermöglicht, einen Dienstplan für Apotheken zu führen. PDR startete 2015 als Alternative zu einer wirklich einfachen Excel-Tabelle ohne Formeln. PDR möchte benutzerfreundlich sein und gleichzeitig alle notwendigen Funktionen abdecken. PDR ist ständig bestrebt, sich zu verbessern. Es ist offen für Ihre Anfragen und Wünsche. Ich hoffe, es wird Ihre Erwartungen erfüllen.

1.1 PDR finden

Die neueste Version von PDR ist verfügbar unter [GitHub](#). GitHub stellt den Quellcode als *.zip-Datei oder *.tar.gz Ball zur Verfügung. Extrahieren Sie die Dateien in einen Ordner.

Stellen Sie sicher, dass Sie PDR in ein Verzeichnis entpacken, auf das Ihr Webserver zugreifen kann. PHP und der Webserver müssen Lesezugriff auf alle Dateien und Ordner haben. Es benötigt auch Schreibzugriff auf die Unterverzeichnisse upload, tmp und config. Möglicherweise möchten Sie den Webserver-Benutzer zum Besitzer des Verzeichnisses machen mit z. B.:

```
1 sudo chown -R www-data:www-data /var/www/html/pdr/
```

Sie können das Repository auch mit git klonen:

```
1 git clone https://github.com/MaMaKow/dienstplan-apotheke.git
```

Details finden Sie im Administratorhandbuch!

1.2 Lizenz

PDR ist Open-Source-Software unter der AGPL-Lizenz.

Copyright (C) 2018 Dr. Martin Mandelkow

Dieses Programm ist freie Software: Sie können es unter den Bedingungen der GNU Affero General Public License, wie von der Free Software Foundation veröffentlicht, entweder Version 3 der Lizenz oder (nach Ihrer Wahl) einer späteren Version, weitergeben und / oder modifizieren.

Dieses Programm wird in der Hoffnung verbreitet, dass es nützlich sein wird, aber OHNE JEGLICHE GARANTIE; ohne auch nur die stillschweigende Gewährleistung der MARKTGÄNGIGKEIT oder EIGNUNG FÜR EINEN BESTIMMTEN

ZWECK. Weitere Informationen finden Sie in der GNU Affero General Public License.

Sie sollten eine Kopie der GNU Affero General Public License zusammen mit diesem Programm erhalten haben. Wenn nicht, siehe <<https://www.gnu.org/licenses/>>.

Weitere Informationen finden Sie in der [Lizenzdatei](#).

1.3 Fehler melden

Der Bugtracker befindet sich derzeit bei GitHub <https://github.com/MaMaKow/dienstplan-apotheke/issues>. GitHub benötigt ein Konto, um Fehler oder Feature-Anfragen zu melden. Wenn Sie keinen Account erstellen wollen, können Sie eine E-Mail an pdr-issues@martin-mandelkow.de senden.

1.4 Wie Sie einen Beitrag leisten können

Pull requests sind erwünscht. Wenn Sie Änderungen an PDR vorgenommen haben und diese der Öffentlichkeit beisteuern möchten, können Sie einen Pull requests auf GitHub öffnen oder Ihre Änderungen auf andere Weise senden.

Sie könnten auch `git send-email` Und Patches an pdr-discuss@martin-mandelkow.de senden

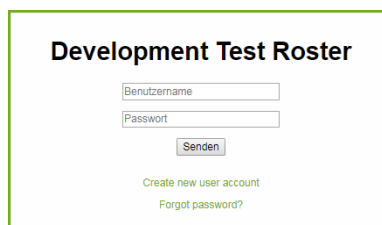
Kapitel 2

Benutzerhandbuch

2.1 Das Web-Interface

Sie können sich mit einem beliebigen Webbrowser mit Ihrer PDR-Instanz verbinden. Navigieren Sie einfach zu Ihrem Server und geben Sie Ihren Benutzernamen und Ihr Passwort ein.

2.1.1 Login

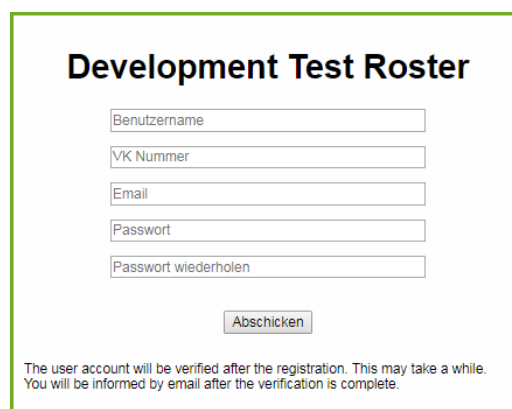


The screenshot shows the login interface for 'Development Test Roster'. It features a title 'Development Test Roster' at the top. Below the title are two input fields: 'Benutzername' (Username) and 'Passwort' (Password). A 'Senden' (Send) button is positioned below the password field. At the bottom of the form, there are two links: 'Create new user account' and 'Forgot password?'.

Abbildung 2.1: Login Seite

Die Anmeldeseite zeigt den Namen der Anwendung an. Sie werden aufgefordert, Ihren Benutzernamen und Ihr Passwort einzugeben. Wenn Sie noch keinen Account haben, können Sie [Create a new account](#) erstellen. Wenn Sie ein Konto haben, aber Ihr Passwort vergessen haben oder es ändern möchten, können Sie auf [Passwort vergessen?](#) Klicken.

2.1.2 Neuen Benutzer-Account erstellen



The screenshot shows the registration interface for 'Development Test Roster'. It features a title 'Development Test Roster' at the top. Below the title are five input fields: 'Benutzername' (Username), 'VK Nummer' (VK Number), 'Email', 'Passwort' (Password), and 'Passwort wiederholen' (Repeat Password). An 'Abschicken' (Submit) button is positioned below the password fields. At the bottom of the form, there is a note: 'The user account will be verified after the registration. This may take a while. You will be informed by email after the verification is complete.'

Abbildung 2.2: Registrierungsseite

Wählen Sie einen Benutzernamen, geben Sie Ihre Mitarbeiter-ID und Ihre E-Mail-Adresse ein. Wählen Sie ein sicheres Passwort.

Das Konto ist inaktiv, bis ein Administrator es aktiviert. Der Hauptadministrator wird per E-Mail über die Registrierung informiert.

Neue Benutzer können nur für vorhandene Mitarbeiter erstellt werden. Neue Mitarbeiter werden von einem Administrator erstellt.

2.1.3 Passwort vergessen

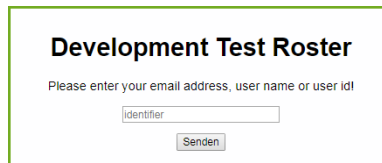


Abbildung 2.3: Passwort vergessen Seite

Auf der Seite "Passwort vergessen" wird der Name der Anwendung angezeigt. Sie werden aufgefordert, entweder Ihren Benutzernamen, Ihre ID oder Ihre E-Mail-Adresse einzugeben. Nachdem Sie das Formular abgeschickt haben, wird eine E-Mail an Ihre gespeicherte E-Mail-Adresse gesendet. In dieser E-Mail finden Sie einen Link, der Sie zur Seite zum Ändern des Passworts führt.

Wiederherstellung des Passwortes



Abbildung 2.4: Wiederherstellungs-Seite für Passwörter

Auf der Seite zur Wiederherstellung des verlorenen Passworts werden der Name der Anwendung und Ihr Benutzername angezeigt. Sie werden aufgefordert, ein neues Passwort zweimal einzugeben.

2.1.4 Navigation

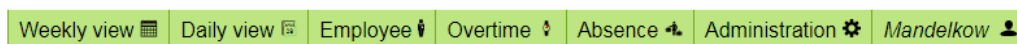



Abbildung 2.5: Navigationsleiste

Standardmäßig öffnet die PDR-Weboberfläche ein Menü mit 5 Kacheln. Sie können navigieren zu:

- Dienstplan Wochenansicht 

- Dienstplan Tagesansicht 
- Dienstplan Mitarbeiteransicht 
- Überstunden 
- Abwesenheit 

Die Navigationsleiste

Im oberen Bereich befindet sich eine Navigationsleiste mit Hyperlinks zu fast allen PDR-Seiten. Bewegen Sie die Maus über einen Eintrag, um die Untermenüs zu öffnen ([Abbildung 2.5](#)).



2.1.5 Dienstplan Wochenansicht

calendar week 32

Small Branch

<< 1 week backward >> 1 week forward

Monday 06.08.	Tuesday 07.08.	Wednesday 08.08.	Thursday 09.08.	Friday 10.08.	Saturday 11.08.	Sunday 12.08.
PI_17 / 6 08:00 - 14:00 PTA_10 / 7 08:00 - 15:30 break: 11:30 - 12:00 Apotheker_9 / 4,5 14:00 - 18:30	Apotheker_9 / 2 08:00 - 10:00 PI_17 / 8 09:30 - 18:00 break: 11:30 - 12:00 PTA_10 / 7,5 10:00 - 18:30 break: 12:00 - 13:00	PI_17 / 6 08:00 - 14:00 PTA_10 / 7,5 08:00 - 16:00 break: 11:30 - 12:00 Apotheker_9 / 4 14:00 - 18:00 PKA_11 / 2 16:00 - 18:00	Apotheker_9 / 4 08:00 - 12:00 PTA_10 / 8,5 09:00 - 18:00 break: 11:30 - 12:00 PI_17 / 6 12:00 - 18:30 break: 13:00 - 13:30	PI_17 / 6 08:00 - 14:00 PTA_10 / 6,5 08:00 - 15:00 break: 11:30 - 12:00 Apotheker_9 / 4,5 14:00 - 18:30		

Small Branch in Big Branch

Monday 06.08.	Tuesday 07.08.	Wednesday 08.08.	Thursday 09.08.	Friday 10.08.	Saturday 11.08.	Sunday 12.08.
PTA_10 / 1,5 18:30 - 20:00 Absentees PTA_8 (maternity leave) PTA_12 (vacation) Apotheker_14 (vacation) PKA_18 (vacation) PKA_19 (vacation)	PTA_10 / 1,5 18:30 - 20:00 Absentees PTA_8 (maternity leave) PTA_12 (vacation) Apotheker_14 (vacation) PKA_18 (vacation) PKA_19 (vacation)	PTA_10 / 1,5 18:30 - 20:00 Absentees PTA_8 (maternity leave) PTA_12 (vacation) Apotheker_14 (vacation) PKA_18 (vacation) PKA_19 (vacation)	PTA_10 / 1,5 18:30 - 20:00 Absentees PTA_8 (maternity leave) PTA_12 (vacation) Apotheker_14 (vacation) PKA_18 (vacation) PKA_19 (vacation)	PTA_10 / 1,5 18:30 - 20:00 Absentees PTA_8 (maternity leave) PTA_12 (vacation) Apotheker_14 (vacation) PKA_18 (vacation) PKA_19 (vacation)	PTA_10 / 1,5 18:30 - 20:00 Absentees PTA_8 (maternity leave) PTA_12 (vacation) Apotheker_14 (vacation) PKA_18 (vacation) PKA_19 (vacation)	PTA_10 / 1,5 18:30 - 20:00 Absentees PTA_8 (maternity leave) PTA_12 (vacation) Apotheker_14 (vacation) PKA_18 (vacation) PKA_19 (vacation)

Abbildung 2.6: Dienstplan Wochenansicht, Auszug ohne Aufgabenrotation und wöchentliche Arbeitszeit

Die Dienstplan-Wochenansicht zeigt die Liste einer ausgewählten Woche und Filiale ([Abbildung 2.6](#)). Wenn Mitarbeiter der Apotheke in einer anderen Filiale arbeiten, werden diese unten angezeigt. Der Tabellenfuß enthält Informationen über abwesende Mitarbeiter und deren Abwesenheitsgründe.

Das Datum kann durch direkte Eingabe ausgewählt werden. Es kann auch um eine Woche vor oder zurück verschoben werden, indem man **Ctrl** + **↑** + **→** oder **Ctrl** + **↑** + **←** drückt.

2.1.6 Dienstplan Tagesansicht

Schreibgeschützt

In der Tageslistenansicht gibt es eine Tabelle, ein Balkendiagramm und ein Histogramm, das den Dienstplan wiedergibt ([Abbildung 2.7](#)).

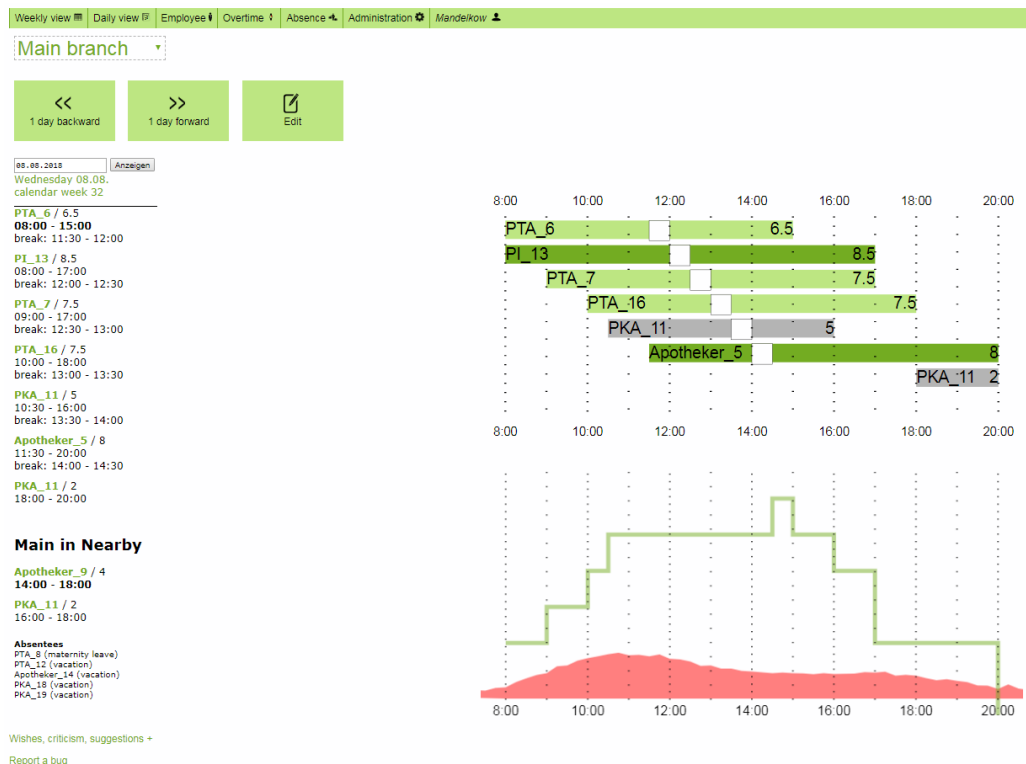


Abbildung 2.7: Dienstplan Tagesansicht

Die Dienstplantabelle listet alle Mitarbeiter auf, die in dem ausgewählten Zweig an dem ausgewählten Tag geplant sind. Jeder Eintrag enthält die ID und den Nachnamen des Mitarbeiters, die Arbeitsstunden, den Beginn und das Ende des Dienstes und die Zeit der Mittagspause, falls vorhanden.

Wenn ein Mitarbeiter, der in erster Linie in der ausgewählten Filiale eingeplant ist, in einer anderen Filiale arbeitet, wird dieser Eintrag in der Tabelle unten angezeigt. Ein Mitarbeiter kann mehr als einen Eintrag pro Tag haben. Dadurch kann eine geteilte Arbeitszeit gespeichert werden. Wenn Mitarbeiter abwesend sind, werden diese Abwesenheiten in der Tabellenfußzeile angezeigt.

Das Dienstplan-Balken-Diagramm zeigt das Kommen und Gehen von Mitarbeitern. Jeder Balken repräsentiert einen Eintrag. Er reicht vom Beginn des Dienstes bis zu seinem Ende. Ein weißes Rechteck auf dem Balken zeigt die Zeit der Mittagspause an. Die Farbe der Balken hängt vom Beruf des Mitarbeiters ab. Apotheker und Pharmazieingenieure sind in dunkelgrün gefärbt, während PTA in hellgrün gefärbt sind. Andere Mitarbeiter (nichtpharmazeutisches Personal) sind grau hinterlegt.

Das Histogramm zeigt einen roten Bereich und eine grüne Linie. Der rote Bereich zeigt den erwarteten Arbeitsaufwand (gemessen in Packungen pro 15 Minuten), während die grüne Linie die Anzahl der arbeitenden Mitarbeiter zu einem bestimmten Zeitpunkt darstellt.

Bearbeiten

Die Bearbeitungsseite sieht der schreibgeschützten Ansicht sehr ähnlich ([Abbildung 2.8](#)).

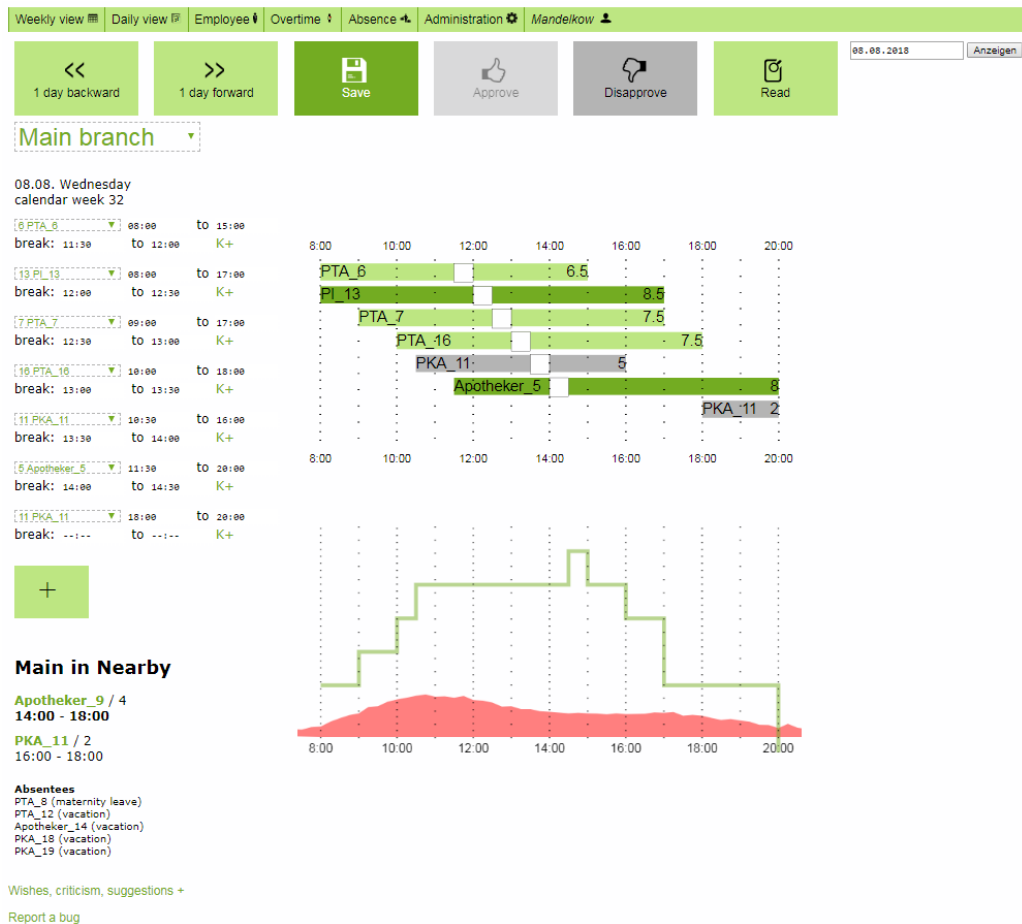


Abbildung 2.8: Dienstplanansicht mit Bearbeitungsrechten

Der Dienstplan wird auf Fehler überprüft. Wenn Probleme auftreten, werden Fehler, Warnungen oder Informationen im oberen rechten Bereich angezeigt. Die Prüfung beinhaltet:

- Überlappung von Schichten für denselben Mitarbeiter (Fehler)
- ausreichende Mitarbeiterzahl (Warnung, fest codiert mindestens zwei Mitarbeiter)
- Anwesenheit von mindestens einem Apotheker zu jeder Zeit (Fehler).
- Anwesenheit von mindestens einer Person, die den Wareneingang durchführen kann (Warnung).
- Einsatz abwesender Mitarbeiter (Fehler)
- Nichteinplanung von nicht abwesenden Mitarbeitern (Warnung)

Pro Eintrag kann nur eine Pause eingefügt werden. Wenn mehr Pausen zugewiesen werden müssen, können mehrere Einträge für denselben Mitarbeiter eingegeben werden.

2.1.7 Dienstplan Mitarbeiteransicht

Die Mitarbeiteransicht ähnelt der Wochenansicht, es wird jedoch nur ein Mitarbeiter angezeigt ([Abbildung 2.9](#)).

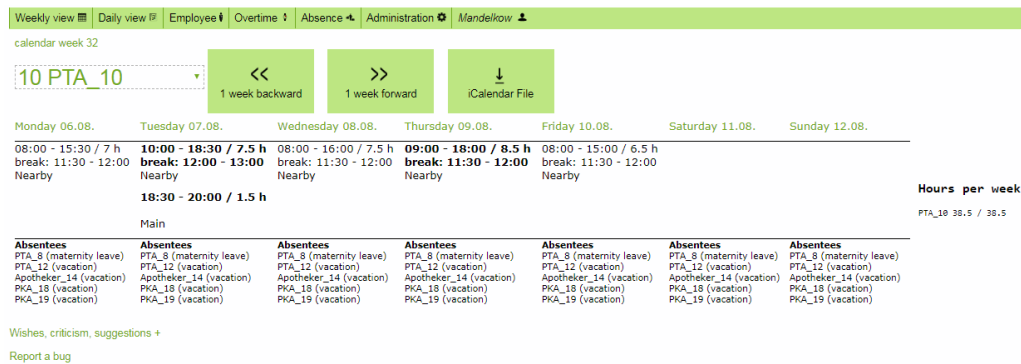


Abbildung 2.9: Mitarbeiteransicht

Eine iCalendar-Datei kann heruntergeladen werden. Siehe [Abschnitt 2.2 Kalender API](#) für weitere Informationen.

2.1.8 Grundplan Tagesansicht

Ein grundsätzlich wiederkehrender Dienstplan kann gespeichert werden. Im einfachsten Fall handelt es sich dabei um eine Auflistung von Beginn und Ende des Dienstes aller Mitarbeiter. Jeder Wochentag wird separat aufgeführt, ebenso die Filialen.

Es ist möglich, abwechselnde Wochen zu erstellen. Beispielsweise kann eine Mitarbeiterin in einer A-Woche und einer B-Woche arbeiten. In A-Wochen beginnt sie am Montag regelmäßig um 08:00 Uhr, in B-Wochen um 10:00 Uhr.

2.1.9 Überstunden

2.1.10 Abwesenheit

Es gibt vier Ansichten zu den Abwesenheitsdaten.

- Mitarbeiteransicht schreibgeschützt
- Mitarbeiteransicht Eingabe
- Monatstabelle
- Jahrestabelle

In der *schreibgeschützten Mitarbeitersicht* gibt es ein Select-Element, um den anzuzeigenden Mitarbeiter auszuwählen. Es gibt eine Schaltfläche, um zur Bearbeitungsansicht zu wechseln. Und es gibt eine Tabelle mit den Abwesenheitsdaten. Die Spalten sind Beginn und Ende der Abwesenheit, Abwesenheitsgrund und Anzahl der Tage. Es gibt eine eindeutige Liste möglicher Gründe (Urlaub, Resturlaub, Krankheit, Krankheit des Kindes, unbezahlte Freistellung, bezahlte Freistellung, Elternzeit und Mutterschutz). Die Anzahl der Abwesenheitstage wird für eine 5-Tage-Woche berechnet. Abwesenheiten an Samstagen und Sonntagen werden registriert, aber nicht gezählt. Das Gleiche gilt für Feiertage.

2.2 Kalender API

Es ist möglich, die Dienstplandaten in Form von iCalendar-Dateien von PDR zu lesen. Diese Dateien können mit allen wichtigen Kalenderanwendungen auf Desktops und Smartphones verwendet werden. Diese API ist keinesfalls eine vollständige Implementierung des Webdav-Standards. Es ist nicht einmal eine Implementierung des CalDAV-Protokolls. Rufen Sie in Ihrem Browser einfach auf die folgende URL auf: <https://YOURHOSTNAME/YOUR/FOLDER/webdav.php>

Die Optionen sind:

`employee_id` Die Mitarbeiter-ID des Benutzers, von dem der Dienstplan ausgegeben werden soll. Jeder Benutzer kann die Liste aller Mitarbeiter abrufen. (Standard = der angemeldete Benutzer)

`date_string` Ein Datum im Format JJJJ-MM-TT (Standard = heute)

`days_into_the_future` Die Anzahl der Tage, die in der Kalenderdatei enthalten sein sollen. (Standard = 30)

`create_valarm` Erstellen Sie einen Alarm (ACTION:DISPLAY) auf Ihrem Gerät. (Standard = 0)

- 0 = kein Alarm
- 1 = Alarm 30 Minuten vor Dienstantritt
- 2 = Alarm am Ende des Dienstes
- 4 = Alarm, wenn die Mittagspause beginnt
- 8 = Alarm, wenn die Mittagspause endet
- 11 = 1+2+8 = Alarm für Beginn und Ende des Dienstes sowie für das Ende des Mittagessens, jedoch nicht für den Beginn des Mittagessens

Um den Dienstplan der Woche ab 17.12.2018 für den Angestellten Nummer 5 zu erhalten, verwenden Sie die folgende URL: https://YOURHOSTNAME/YOUR/FOLDER/webdav.php?employee_id=5&date_string=2018-12-17&days_into_the_future=6

2.2.1 ICalendar-Dateien automatisch mit iCal Import/Export CalDAV Proimportieren

Der automatische Import von iCalendar-Dateien in Android-Smartphones wurde mit iCal Import/Export CalDAV Pro"(3,59 EUR) getestet. Es gibt wahrscheinlich andere Apps, die das Selbe tun können.

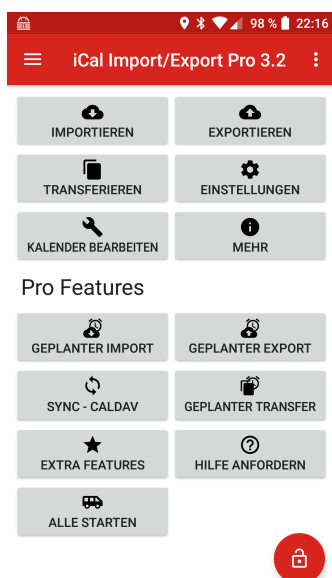


Abbildung 2.10: iCal Hauptmenü

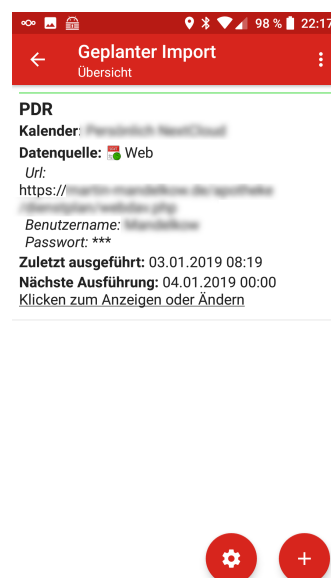


Abbildung 2.11: iCal geplante Importe

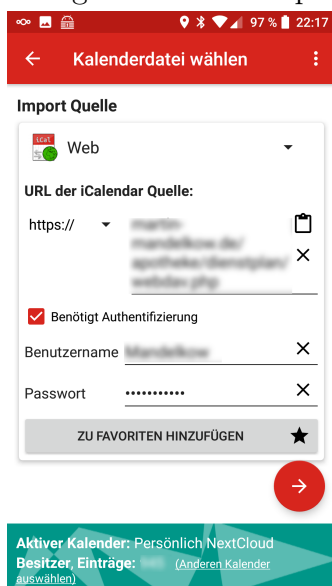


Abbildung 2.12: iCal eine neue Importquelle hinzufügen

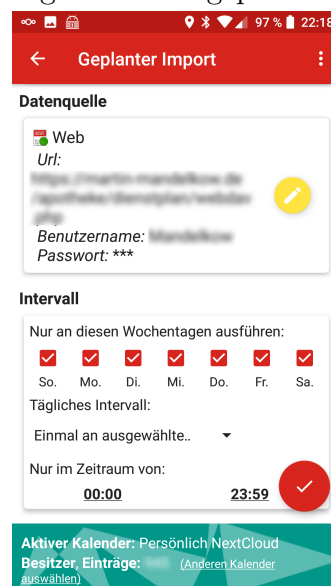


Abbildung 2.13: iCal Importintervalle einstellen

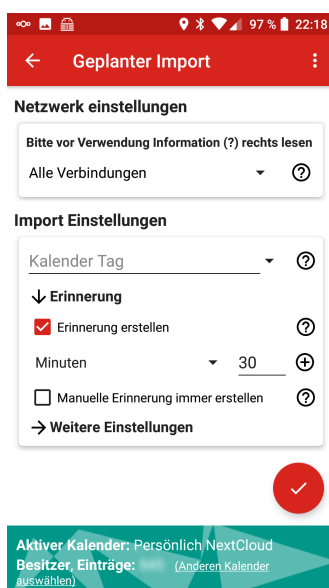


Abbildung 2.14: iCal Netzwerkeinstellungen und Importeinstellungen

Kapitel 3

Administratorhandbuch

3.1 Installation

3.1.1 PDR finden

Die neueste Version von PDR ist verfügbar unter [GitHub](#)

Sie können auch die neueste stabile Version über Git erhalten:

```
1 git clone https://github.com/MaMaKow/dienstplan-apotheke.git
```

Der master Zweig ist auf Stabilität geprüft.

3.1.2 Die Installationsroutine

Einführung

Auf der ersten Seite finden Sie einige nicht-technische Informationen zu diesem Programm. Klicken Sie auf [Next](#), um weiterzugehen.

Willkommen

Auf der zweiten Seite sind einige technische Hintergrundinformationen gegeben. Sie werden über die für die Fortführung der Installation erforderlichen Eingangsdaten informiert. Die verfügbaren Datenbankverwaltungssysteme (derzeit nur MySQL) werden aufgelistet. Schließlich werden Sie über die Nutzer- und Passwortstrategie für den Datenbankzugriff informiert. Klicken Sie erneut auf [Next](#), um fortzufahren.

Anforderungen

Auf der nächsten Seite prüft die Anwendung, ob alle Voraussetzungen erfüllt sind. Dazu gehören eine minimale PHP-Version, einige PHP-Erweiterungen und die Unterstützung für Datenbankverbindungen. Das Programm benötigt Schreibzugriff auf einige seiner Verzeichnisse. Wenn Probleme gefunden werden, wird eine beschreibende Fehlermeldung angezeigt. Es ist nicht möglich, weiterzumachen, bis alle Probleme gelöst sind. Klicken Sie auf [Next](#), um fortzufahren.

Datenbankkonfiguration

Die Anwendung beginnt nun mit der Erhebung von Konfigurationsdaten.

- Datenbanktyp



- Hostname
- Port (optional)
- username
Ein bestehender Datenbankbenutzer. Der Benutzer MUSS die Berechtigung haben, eine Datenbank zu erstellen. Der Benutzer SOLLTE das Privileg haben, einen weniger privilegierten Benutzer zu erstellen.
- password
Das Datenbankpasswort des Benutzers. Wenn ein neuer Benutzer erstellt werden konnte, wird dem neuen Benutzer ein neues sicheres zufälliges Passwort gegeben.
- Name der Datenbank

Geben Sie die erforderlichen Daten ein und klicken Sie .

Administrative Konfiguration



Nachdem die Datenbankwerte festgelegt wurden, werden einige Informationen zum Administrator gesammelt:

- Benutzername
Der Name, mit dem der Administrator sich in Zukunft im Programm anmeldet.
- Nachname
Dieser Name ist mit der Mitarbeiter-ID verbunden.
- Mitarbeiter-ID
Diese wird verwendet, um einen Mitarbeiter zu erstellen, der mit dem Benutzer mit Administratorrechten verbunden ist.
- Die Kontakt-E-Mail-Adresse
wird für Fragen und Kommentare von Benutzern verwendet. Diese E-Mail erhält auch einige interne Informationen aus dem Dienstplan.
- Administrator-Passwort
das Passwort, das der Administrator verwendet, um sich in Zukunft am Programm anzumelden.

Bitte registrieren Sie den Administrator und klicken Sie auf . Die Daten werden in die Datei  `config/config.php` geschrieben. Für jeden Benutzer, der das Programm benutzt, muss es genau einen Mitarbeiter geben.

3.1.3 Erste Schritte

Nach dem Absenden der Administratorkonfiguration werden Sie auf die Anmeldeseite weitergeleitet. Melden Sie sich mit Ihren Administrator-Anmeldeinformationen an.

Bei Ihrer ersten Anmeldung werden Sie mit der Filialverwaltung konfrontiert. Bitte erstellen Sie mindestens eine Filiale. Sie erreichen diese Seite jederzeit über das Menü  .

Der nächste logische Schritt ist die Einrichtung einiger weiterer Mitarbeiter in  .

Nachdem alle Mitarbeiter eingefügt sind, können Sie mit dem Schreiben von Dienstpläne beginnen ([Tagesansicht](#) [Tagesansicht Eingabe](#)) oder Sie können Grundpläne erstellen für bestimmte Wochentage ([Tagesansicht](#) [Grundplan Tagesansicht](#)) oder für unterschiedliche Mitarbeiter ([Mitarbeiter](#) [Grundplan Mitarbeiter](#)).

3.2 Aktualisierung

Bis jetzt gibt es keinen automatischen Update-Mechanismus. Sie können regelmäßig Release-Pakete von GitHub herunterladen. Oder Sie können über Git in Kontakt bleiben:

```
1 git pull origin master
```

CAVE: Stellen Sie sicher, dass Sie Ihr `config/config.php` behalten! Es sollte nicht von Git geändert werden, da es in der `.gitignore`-Datei dieses Projekts aufgeführt ist.

3.3 Konfiguration

Sie können die Datei `config/config.php` manuell bearbeiten. Die Standardwerte sind:

```
1 <?php
2
3 $config = array( 'application_name' => 'PDR', 'database_management_system'
4 => 'mysql', 'database_host' => 'localhost', 'database_name' => '',
5 'database_port' => 3306, 'database_user' => '', 'database_password' => '',
6 'session_secret' => '', 'error_reporting' => E_ALL, 'display_errors' => 0,
7 'log_errors' => 1, 'error_log' => PDR_FILE_SYSTEM_APPLICATION_PATH
8 . 'error.log', 'LC_TIME' => 'C', 'timezone' => 'Europe/Berlin', 'language'
9 => 'de_DE', 'mb_internal_encoding' => 'UTF-8', 'contact_email' => '',
10 'hide_disapproved' => FALSE, 'email_method' => 'mail', 'email_smtp_host' =>
11 NULL, 'email_smtp_port' => 587, 'email_smtp_username' => NULL,
12 'email_smtp_password' => NULL, );
```

Löschen Sie niemals die ersten zwei Zeilen! Wenn die Datei nicht mit `<?php` beginnt, wird PHP sie nicht verarbeiten, was bedeutet, dass jeder ihren Inhalt lesen kann.

Die meisten dieser Optionen können auch in [Administration](#) [Konfiguration](#)

Anwendungsname konfiguriert werden. Dieser Name wird auf der Anmeldeseite, dem Seitentitel im Browser und als Betreffzeile in E-Mails verwendet, die vom Programm gesendet werden.

Datenbank Einstellungen

- `database_management_system` Derzeit wird nur MySQL unterstützt. Andere Möglichkeiten könnten sein: PostgreSQL, Oracle Database, SQLite, Microsoft Access oder MongoDB.
- `database_host` Der Server, auf dem das DBMS ausgeführt wird. Normalerweise 'localhost', wenn es sich auf demselben Server befindet wie die Anwendung.
- `database_name` Der Name der Datenbank
- `database_port` Für MySQL ist der Standardport 3306.

- **database_user** Während der Installation versucht PDR, den Benutzer 'pdr' in der Datenbank zu erstellen. Im Erfolgsfall wählt das Programm ein zufälliges Passwort und gewährt alle Berechtigungen für die pdr-Datenbank. Sie können jeden anderen Benutzer auswählen, der Zugriff auf die Datenbank hat.
- **database_password** Das Datenbankkennwort des Datenbankbenutzers.

session_secret Eine geheime zufällige Zeichenfolge, mit der der Sitzungsname definiert wird. Dies ist nur relevant, wenn mehrere Instanzen von PDR auf demselben Webserver ausgeführt werden.

error_reporting Welche Fehler sollte PHP melden?

display_errors Sollten dem Benutzer Fehler direkt angezeigt werden?

log_errors Sollten Fehler in einer Datei protokolliert werden?

error_log Wo sollen Fehler protokolliert werden?

LC_TIME In welcher Sprache sollten Zeitzeichenfolgen wie Montag oder Januar geschrieben werden? Diese Einstellung ist unabhängig von der Einstellung 'language'.

timezone Die Zeitzone ist notwendig, um die Rohzeitdaten in Unix-Zeitstempeln zu verstehen. (1545038523 = Montag, 17.-18.12. 09:22:03 UTC in RFC 2822)

language Die Sprache der Begriffe und Meldungen, die dem Benutzer angezeigt werden. Bisher werden nur Englisch und Deutsch unterstützt.

mb_internal_encoding Eine Codierung ist eine Möglichkeit, dem Computer mitzuteilen, wie Bits und Bytes in Buchstaben übersetzt werden (z. B. bedeutet UTF8 01000101 E, 11000011 10100100 ä).

contact_email E-Mails der Benutzer können an den Administrator der Pdr-Instanz gesendet werden.

hide_disapproved Es ist möglich, geplante Dienstpläne auszublenden, bis sie genehmigt werden. Standardmäßig sind alle Dienstpläne sofort für jeden Benutzer sichtbar.

E-Mail-Einstellungen E-Mails werden (teilweise) mit der PHPMailer-Klasse gesendet.

- 'email_method' PHPMailer unterstützt das Versenden per 'mail', 'sendmail', 'qmail' oder 'SMTP'. Wenn Sie SMTP als E-Mail-Methode wählen, werden die SMTP-Einstellungen angezeigt und müssen ausgefüllt werden:
- 'email_smtp_host' Die Adresse des SMTP-Servers (z. B. localhost für lokalen Postfix, gmail-smtp-in.l.google.com für gmail).
- 'email_smtp_port' Normalerweise 25, 465 oder 587 (25 = SMTP, 465 = SMTPS, 587 = STARTTLS)

- 'email_smtp_username' Benutzername des sendenden E-Mail-Kontos
- 'email_smtp_password' Passwort des sendenden E-Mail-Kontos

3.4 Wartung

Die Datei `src/php/background_maintenance.php` wird bei jedem Login eines beliebigen Benutzers aufgerufen. Es wird eine Instanz der folgenden Klassen erzeugt:

- Wartung
- update_database

3.4.1 Klasse maintenance

Die Methoden, die in der Klassen maintenance enthalten sind, werden nur dann aufgerufen, wenn die letzte Ausführung mindestens `MAINTENANCE_PERIOD_IN_SECONDS` her ist. `MAINTENANCE_PERIOD_IN_SECONDS` ist auf einmal am Tag eingestellt.

Die Methode `user_dialog_email->aggregate_messages_about_changed_roster_to_employees()` wird dazu aufgerufen, E-Mails an Mitarbeiter zu senden, deren Dienstplan geändert wurde.

Die Methode `maintenance->cleanup_overtime()` tut noch nichts. Sie soll zum Säubern von Überstunden bestehender Mitarbeiter eingesetzt werden, welche vor dem Eintritt in das Unternehmen eingetragen wurden.

Die Methode `maintenance->cleanup_absence()` tut noch nichts. Sie soll zur Säuberung von Abwesenheiten bestehender Mitarbeiter eingesetzt werden, welche vor dem Eintritt in das Unternehmen eingetragen wurden.

3.4.2 Klasse update_database

Bevor die Klasse `update_database` eine ihrer Methoden ausführt, prüft sie, ob sich die offizielle Datenbankstruktur seit der letzten Aktualisierung geändert hat. Sie vergleicht den `pdr_database_version_hash` aus der Tabelle `pdr_self` in der Datenbank mit dem Wert `PDR_DATABASE_VERSION`.

In der Datei `src/php/database_version_hash.php`

Die Klasse sollte jede SQL-Abfrage enthalten, die notwendig ist, um eine bestehende Datenbank in die neue Struktur zu überführen, ohne Daten zu verlieren.

3.5 Probleme und Fehlerbehebung

Fragen werden bei GitHub <https://github.com/MaKow/dienstplan-apotheke/issues> bearbeitet. Ich versuche, jede Frage innerhalb von 3 Tagen zu beantworten.

Kapitel 4

Entwicklerhandbuch

4.1 Kernentwicklung

Alle PHP-Skripte laden eine gemeinsame Datei `default.php`, die die Standardeinstellungen vornimmt. Sie befindet sich in `./`, welches der `PDR_FILE_SYSTEM_APPLICATION_PATH` ist. Siehe folgende Datei:

```
1 <?php
2
3 /*
4  * Copyright (C) 2017 Mandelkow
5  *
6  * This program is free software: you can redistribute it and/or modify
7  * it under the terms of the GNU Affero General Public License as published by
8  * the Free Software Foundation, either version 3 of the License, or
9  * (at your option) any later version.
10 *
11 * This program is distributed in the hope that it will be useful,
12 * but WITHOUT ANY WARRANTY; without even the implied warranty of
13 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
14 * GNU Affero General Public License for more details.
15 *
16 * You should have received a copy of the GNU Affero General Public License
17 * along with this program. If not, see <http://www.gnu.org/licenses/>.
18 */
19
20 /**
21  * @var PDR_FILE_SYSTEM_APPLICATION_PATH The full path of the application root
22  * as determined by the position of the default.php
23  */
24 define( 'PDR_FILE_SYSTEM_APPLICATION_PATH', __DIR__ . '/' );
25 /**
26  * @var PDR_HTTP_SERVER_APPLICATION_PATH The relative path of the application
27  * root on the web server.
28  */
29 $folder_tree_depth_in_chars = strlen( substr( getcwd(), strlen( __DIR__ ) ) );
30 $root_folder = substr( dirname( $SERVER["SCRIPT_NAME"] ), 0, strlen( dirname(
31     $SERVER["SCRIPT_NAME"] ) ) - $folder_tree_depth_in_chars ) . "/";
32 define( 'PDR_HTTP_SERVER_APPLICATION_PATH', $root_folder );
33 //TODO: This does not work, if the location is a symbolic link.
34 /**
35  * @var PDR_ONE_DAY_IN_SECONDS The amount of seconds in one day.
36  */
37 define( 'PDR_ONE_DAY_IN_SECONDS', 24 * 60 * 60 );
38
```

```

36  /*
37  * Define an autoloader:
38  */
39  spl_autoload_register(function ($class_name) {
40      include_once PDR_FILE_SYSTEM_APPLICATION_PATH . 'src/php/classes/class.' .
        $class_name . '.php';
41  });
42
43
44  if (!file_exists(PDR_FILE_SYSTEM_APPLICATION_PATH . '/config/config.php')) {
45      header("Location: " . PDR_HTTP_SERVER_APPLICATION_PATH . "src/php/pages/
        install_page_intro.php");
46      die("The application does not seem to be installed. Please see the <a href=
        " . PDR_HTTP_SERVER_APPLICATION_PATH . "src/php/pages/install_page_intro.php
        '>installation page</a>!");
47  } else {
48      $config = array();
49      global $config; //This has to be explicitly declared in order to work with
        PHPUnit
50
51      /*
52      * Load configuration parameters from the configuration file:
53      */
54      require_once PDR_FILE_SYSTEM_APPLICATION_PATH . "config/config.php";
55      /*
56      * Complement the configuration array with the default values for unset
        parameters:
57      */
58      foreach (configuration::$List_of_configuration_parameters as $key => $value)
        {
59          if (!isset($config[$key])) {
60              $config[$key] = $value;
61          }
62      }
63  }
64  /*
65  * Setup if errors should be reported to the user, if to log them, and where:
66  */
67  ini_set('display_errors', $config['display_errors']); //Display errors to the
        end user?
68  ini_set('log_errors', $config['log_errors']); //Log errors to file?
69  if ($config['log_errors'] or $config['display_errors']) {
70      /*
71      * Debug mode
72      */
73      //ini_set('zend.assertions', 1); //Assertions will be compiled AND executed.
74      ini_set('assert.exception', 1); //An exception will be thrown if an
        assertion fails.
75  } else {
76      //ini_set('zend.assertions', -1); //Assertions are not compiled.
77      ini_set('assert.exception', 0); //Only warnings would be shown if assertions
        were to be executed and failed.
78  }
79  ini_set('error_log', $config['error_log']); //Which file should errors be logged
        to?
80  error_reporting($config['error_reporting']); //Which errors should be reported?
81
82  /*
83  * We want some functions to be accessible in all scripts.
84  */

```













```

85 require_once PDR_FILE_SYSTEM_APPLICATION_PATH . "funktionen.php";
86
87 /*
88  * Setup the presentation of time values:
89  //setlocale(LC_ALL, 'de_DE'); // Leider versteht die Datenbank dann nicht mehr
    , was die Kommata sollen.
90 */
91 setlocale(LC_TIME, $config['LC_TIME']);
92 /*
93  * Setup default timezone for date()
94  */
95 date_default_timezone_set($config['timezone']);
96 /*
97  * Setup the encoding for multibyte functions:
98  * This is necessary for the usage of UTF-8 characters in functions like
    mb_substr()
99 */
100 mb_internal_encoding($config['mb_internal_encoding']);
101 require_once PDR_FILE_SYSTEM_APPLICATION_PATH . 'src/php/localization.php';
102
103 /*
104  * session management
105  */
106 $session = new sessions;
107
108 /*
109  * TODO: Get rid of this maybe?
110  */
111 $List_of_branch_objects = branch::get_list_of_branch_objects();
112 /*
113  * Guess the navigator (=browser) language from HTTP_ACCEPT_LANGUAGE:
114  * This is used in the head.php
115  */
116 $navigator_languages = preg_split('/[;,]/', filter_input(INPUT_SERVER, '
    HTTP_ACCEPT_LANGUAGE', FILTER_SANITIZE_STRING));
117 $navigator_language = $navigator_languages[0]; //ignore the other options

```

../default.php

4.1.1 Ordnerstruktur

-  **config/** Enthält die Konfigurationsdatei config.php
-  **css/** *veraltet*, verwenden Sie stattdessen  **src/css/**
-  **docs/** Diese Dokumentation und Werkzeuge, um sie zu bauen
-  **img/** Bilder, die vom Programm verwendet werden
-  **js/** *obsolet*, benutzen Sie stattdessen  **src/js/**
-  **locale/** Übersetzungsdateien für gettext, derzeit nur Deutsch (de_DE)
-  **src/** Der Hauptteil des eigentlichen Quellcodes
 -  **src/css** Cascading Style Sheets
 -  **src/js** JavaScript
 -  **src/php/** PHP: Hypertext Preprocessor

- `src/php/classes/` Enthält alle Klassendateien `class.class_name.php`
- `src/php/fragments/` Teile von größeren Seiten, können über PHP `require/include` oder mit JavaScript geladen werden
- `src/php/pages/` Dies ist der Ort für die einzelnen Ansichten, welche der menschliche Benutzer verwenden wird, um den Dienstplan usw. zu sehen.
- `src/sql/` SQL-Datenbank Tabellen und Trigger
- `tests/` Tests, um Fehler im Quellcode zu finden; Dieser Ordner ist in `.gitignore` aufgeführt. Nur einige Dateien sind Teil der sichtbaren Quelle.
- `tmp/` Ein Verzeichnis für temporäre Dateien. Es gibt noch keine automatische Bereinigung.
- `upload/` Das Ziel für hochgeladene Inhalte. Momentan werden nur spezifische `*.PEP`-Dateien verstanden, die von Awta ASYS Smart erstellt wurden. Diese Dateien enthalten Informationen über die Anzahl der Kunden, die in der Vergangenheit bedient wurden.

4.1.2 Programmierstil

Dieses Projekt versucht, einem Programmierstil zu folgen.

- Bitte vermeiden Sie StudlyCaps und camelCase (Binnenmajuskel).
- Klassenkonstanten MÜSSEN in Großbuchstaben mit Unterstrichttrennzeichen deklariert werden.
- Eigenschaftsnamen MÜSSEN in Unter_strichen geschrieben werden.
- Einfache Variablen und Objekte werden in Kleinbuchstaben geschrieben.
- Array-Namen beginnen mit einem einzelnen Großbuchstaben, gefolgt von Kleinbuchstaben.
- Methodennamen müssen mit Unter_strichen geschrieben werden.
- Der Code MUSS 4 Leerzeichen zum Einrücken verwenden, keine Tabulatoren.
- Öffnende geschweifte Klammern für Klassen und Funktionen MÜSSEN in derselben Zeile stehen, und das Schließen geschweifter Klammern MUSS in der nächsten Zeile nach dem Text beginnen.
- Öffnende geschweifte Klammern für Kontrollstrukturen SOLLTEN in derselben Zeile stehen, und das Schließen geschweifter Klammern MUSS in der nächsten Zeile nach dem Text beginnen.

Festplattenplatz ist nicht mehr knapp. IDEs helfen mit Autovervollständigung. Es ist nicht nötig, Dinge abzukürzen. Bitte verwenden Sie lange Begriffe wie `user_email_notification_cach` statt `usr_ml_ntfcn_ca` oder `u_e_n_c`.

4.1.3 Die Datenbank

Derzeit wird nur MySQL als Datenbankverwaltungssystem (DBMS) unterstützt. Die Tabellen sind:

- Abwesenheit (Krankheit, Urlaub und andere Arten von Abwesenheit)
- approval (speichert für jeden Tag, ob der Leiter offiziell den Dienstplan autorisiert hat)
- branch (Informationen über die Hauptapotheke und mögliche Filialen)
- Dienstplan (die tatsächlichen Dienstplandaten; Anfang, Ende, Pause)
- employees (Mitarbeiterdaten; Mitarbeiter_id, Name, Beruf, Fähigkeiten)
- employees_backup (eine Kopie der Mitarbeitertabelle mit archivierten historischen Daten)
- Feiertage (obsolet)
- principle_roster (der Grundplan; Anfang, Ende, Pause; wird verwendet, um neue Dienstpläne vorzuschlagen)
- Wartung (obsolet)
- Mandant (obsolet)
- Notdienst (Daten der Notdienste und der ihnen zugewiesenen Mitarbeiter)
- opening_times_special (noch nicht benutzt)
- opening_times (die Öffnungs- und Schließzeiten der Filialen, noch keine GUI zur Bearbeitung)
- pdr_self (spiegelt den Zustand der Anwendung selbst wider)
- pep_month_day (die relative Menge an Arbeit an verschiedenen Tagen im Monat)
- pep (die rohen Daten zur Arbeitsmenge, gehashed, um die Anzahl der gelöschten / ignorierten Einträge zu reduzieren)
- pep_weekday_time (die Menge der Arbeit zu verschiedenen Zeiten an verschiedenen Wochentagen)
- pep_year_month (die relative Menge an Arbeit in verschiedenen Monaten im Jahr)
- saturday_rotation (Wer soll an welchem Samstag arbeiten?)
- saturday_rotation_teams (Wer gehört zu welcher Mannschaft für die Rotation am Samstag?)
- Schulferien (noch nicht benutzt)
- Stunden (Überstundenarchiv und Saldo)
- task_rotation (rotierende Zuordnung von Mitarbeitern zu einer Aufgabe, z.B. Rezeptur)
- user_email_notification_cache (noch nicht benutzt)

- `users_lost_password_token` (Token zur Verfügung gestellt, um ein vergessenes Passwort zu ändern)
- `users_privileges` (die Berechtigungen der Benutzerkonten)
- `users` (die Benutzerkonten; Für jeden Benutzeraccount muss genau ein Mitarbeiter vorhanden sein. Es können Mitarbeiter ohne Benutzerkonten existieren)
- Wunschplan (obsolet)

Eine Kopie aller Tabellenstrukturen wird in `src/sql/` gespeichert. Das Verzeichnis enthält auch die Datei `src/sql/database_version_hash.php`, die einen SHA1-Hash aller Strukturen enthält, die von `SHOW CREATE TABLE` und `SHOW CREATE TRIGGER` nach einigen Änderungen zurückgegeben werden. Der Hash wird von `tests/get-database-structure.php` geschrieben, siehe die Details in jener Datei.

Pflege der Datenbank

Es gibt eine Klasse `update_database`. Diese Klasse enthält eine definierte Menge von MySQL-Anweisungen, die die Datenbankstruktur von einem bekannten Zustand in der Vergangenheit in den aktuellen Zustand versetzen.

Diese Klasse ist nicht gut getestet. Sie könnte funktionieren. Sie könnte auch die gesamte Datenbank zerstören.

Die Klasse `update_database` wird bei jedem Login eines Nutzers aufgerufen. Sie entscheidet dann selbst, ob irgendwelche Maßnahmen ergriffen werden müssen. Um dies zu entscheiden, wird der in der Datei `src/sql/database_version_hash.php` gespeicherte Hash mit dem in der Datenbanktabelle gespeicherten Hash verglichen `[pdr_self] >> pdr_database_version_hash`.

Selbstheilende Tabellen Die Klasse `database_wrapper` hat eine Funktion `create_table_from_template`, die in der Lage ist, fehlende Tabellen aus den Strukturinformationen zu erstellen, die im Verzeichnis `src/sql/` gespeichert sind. Sie wird aufgerufen, wenn eine PDO-Datenbankabfrage eine Ausnahme mit dem Code 42S02 und dem MySQL-Fehler 1146 wirft.

4.1.4 Klassen

Die Klassen werden im Ordner `src/php/classes/` oder `src/php/3rdparty/` gespeichert. Der Autoloader lädt nur Klassen aus `src/php/classes/class.$class_name.php`. Jede andere Klasse muss manuell geladen werden.

Benutzer

Die Klasse `user` stellt einen Mitarbeiter dar, der ein Benutzerkonto in PDR registriert hat.

user_input

user_input::get_variable_from_any_input Diese Funktion liest die Benutzereingaben von POST, GET oder COOKIE in dieser Reihenfolge. Wenn die angeforderten Informationen in einer der Quellen gefunden werden, werden die anderen ignoriert. Aus Sicherheitsgründen werden alle Informationen gefiltert. Standardmäßig wird `FILTER_SANITIZE_STRING` eingesetzt. Jeder andere Filter kann als zweiter Parameter angegeben werden. Wenn in keiner der Quellen Informationen gefunden werden, wird ein Standardwert (der dritte Parameter) zurückgegeben.

`escape_sql_value` foo


```
convert_post_empty_to_php_null    foo

principle_employee_roster_write_user_input_to_database    foo

principle_roster_write_user_input_to_database    foo

get_Roster_from_POST_secure    foo

remove_changed_entries_from_database    foo

remove_changed_entries_from_database_principle_roster    foo

insert_changed_entries_into_database_principle_roster    foo

insert_new_approval_into_database    foo

old_write_approval_to_database    foo

get_changed_roster_employee_id_list    foo

get_deleted_roster_employee_id_list    foo

get_inserted_roster_employee_id_list    foo

roster_write_user_input_to_database    foo
```

4.1.5 Web-Interface

4.1.6 Kalender API

Das Skript unterscheidet nicht zwischen Informationen, die von POST, GET oder COOKIE gesendet werden. Siehe `user_input::get_variable_from_any_input` für Details. Der Parameter `date_string` akzeptiert jede Zeichenfolge, die von `DateTime` interpretiert werden kann. Siehe <https://secure.php.net/manual/de/datetime.formats.date.php> für Details.

4.2 Dokumentation

Diese Dokumentation über ein Programm, eine App oder ein Skript ist unvollständig. Sie können diesem Projekt helfen, indem Sie es erweitern. Im Ernst, wenn es etwas gibt, das sich nicht genug erklärt, schickt mir einfach eine E-Mail oder kontaktiert mich bei GitHub!

4.3 Testen

4.4 Bug Tracker

Fehler und Probleme werden bei GitHub verfolgt <https://github.com/MaMaKow/dienstplan-apotheke/issues>

4.5 Übersetzung

Übersetzungen werden mit gettext () verwaltet.

Lesen Sie diesen Artikel über po4a für die Übersetzung dieses Dokuments: <https://maltris.org/mehrsprachigkeit-fur-fast-alles-po4a-7317.html>

4.5.1 Internationalisierung

Verschiedene Länder haben unterschiedliche Gesetze in Bezug auf Apotheken und Beschäftigung. Sie haben auch unterschiedliche Feiertage.