pharmacy duty roster
Documentation

Martin Mandelkow

October 26, 2018

Contents

Chapter 1

Introduction

Pharmacy Duty Roster (PDR) is a web application that allows to operate a duty roster for pharmacies. PDR started in 2015 as an alternative to a really simple excel sheet without formulas. PDR aims to be user-friendly but at the same time cover all necessary features. PDR continuously strives to improve. It is open to your requests and wishes. I hope it will fulfil your expectations.

1.1 Getting PDR

The latest release of PDR is available on GitHub. GitHub provides the source code as *.zip file or *.tar.gz ball. Extract the files into a folder.

Make sure to unpack PDR to a directory, that your webserver has access to. PHP and the webserver must have read access to all the files and folders. It also needs write access to the subdirectories upload, tmp and config. You might want to change the owner of the directory to the webservers user with e.g.:

You can also clone the repository with git:

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

See the Administrator manual for details!

1.2 License

PDR is open source software under the AGPL license.

> Copyright (C) 2018 Dr. Martin Mandelkow
>
> This program is free software: you can redistribute it and/or modify it under the terms of the GNU Affero General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.
>
> This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Affero General Public License for more details.
>
> You should have received a copy of the GNU Affero General Public License along with this program. If not, see <https://www.gnu.org/licenses/>.

Please see the license file for details!

1.3 Reporting bugs

The issue tracker is currently located at GitHub. GitHub requires an account in order to report bugs or feature requests. If you do not want to create one, you might send a mail to pdr-issues@martin-mandelkow.de

1.4 How to contribute

Pull requests are desired. If you made changes to PDR and want to contribute them to the public, you are welcome to open a pull-request on GitHub or send your changes in any other way.

Chapter 2

User manual

2.1 The web interface

You can connect to your PDR instance using any web browser. Just navigate to your server and enter your username and password.

2.1.1 Login

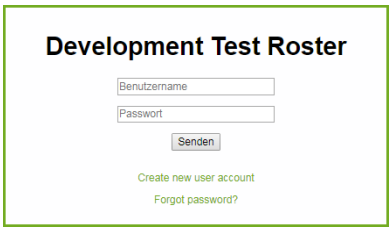


Figure 2.1: Login page

The login page shows the name of the application. You are prompted to enter your username and password. If you do not have an account yet, you can Create a new user account. If you have an account, but forgot about your password, or want to change it, you can click on Forgot password?.

2.1.2 Create new user account

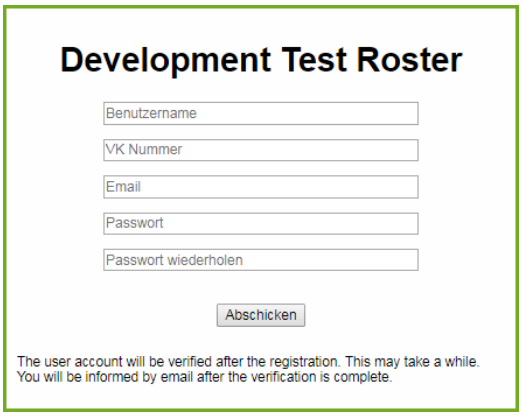


Figure 2.2: Register new user page

Choose a user name, enter your employee id and your email. Pick a secure password.

The account will be inactive until an administrator activates it. The main administrator is informed via email regarding the registration.

New users can only be created for existing employees. New employees are created by an administrator.

2.1.3 Lost password

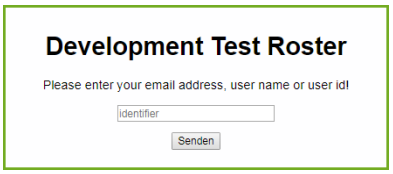


Figure 2.3: Lost password page

The lost password page shows the name of the application. You are prompted to enter either your username, id or your email-address at your option. After you submit the form, an email is sent to your stored email address. In that email you will find a link, which will lead you to the password change page.

Lost password recovery

Figure 2.4: Lost password recovery page

The lost password recovery page shows the name of the application and your user name. You are prompted to enter a new password twice.

2.1.4 Navigation

Figure 2.5: Navigation bar

By default, the PDR web interface opens a menu containing 5 tiles. You can navigate to:

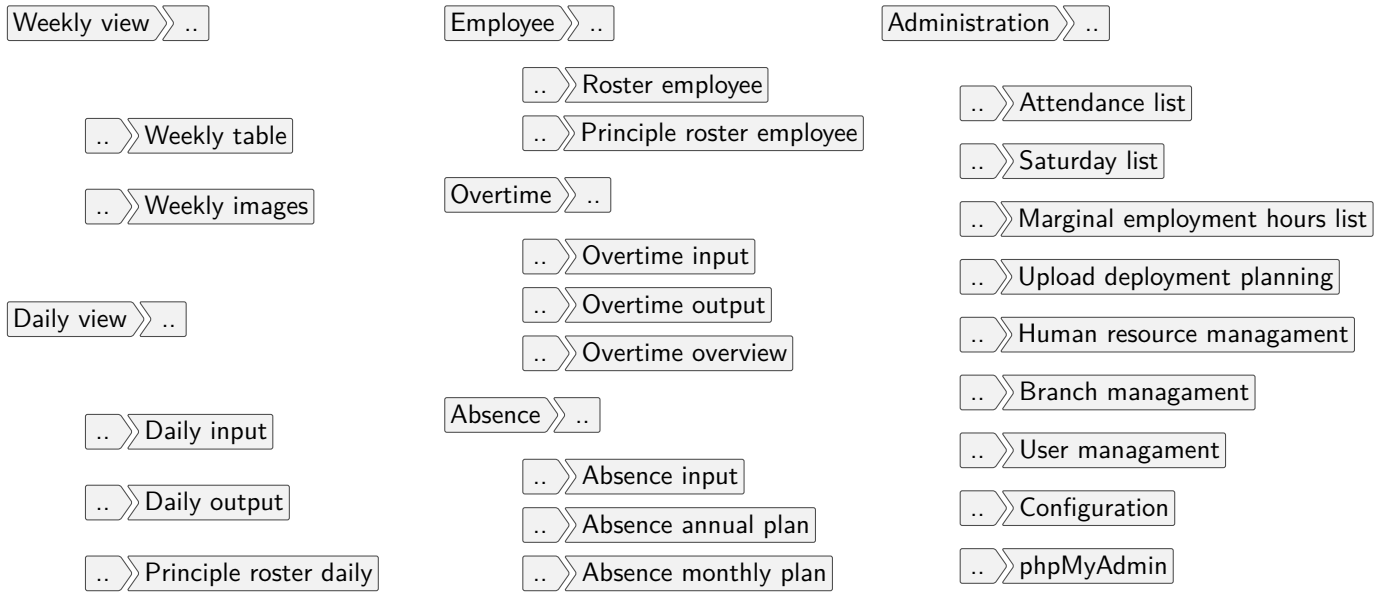
- Roster week table view 
- Roster daily view 

- Roster employee view 

- Overtime 
- Absence 

The navigation bar

In the top there is a navigation bar containing hyperlinks to nearly all the pages of PDR. Hover the mouse over an entry to open the submenus (Figure 3.5).



2.1.5 Roster week table view

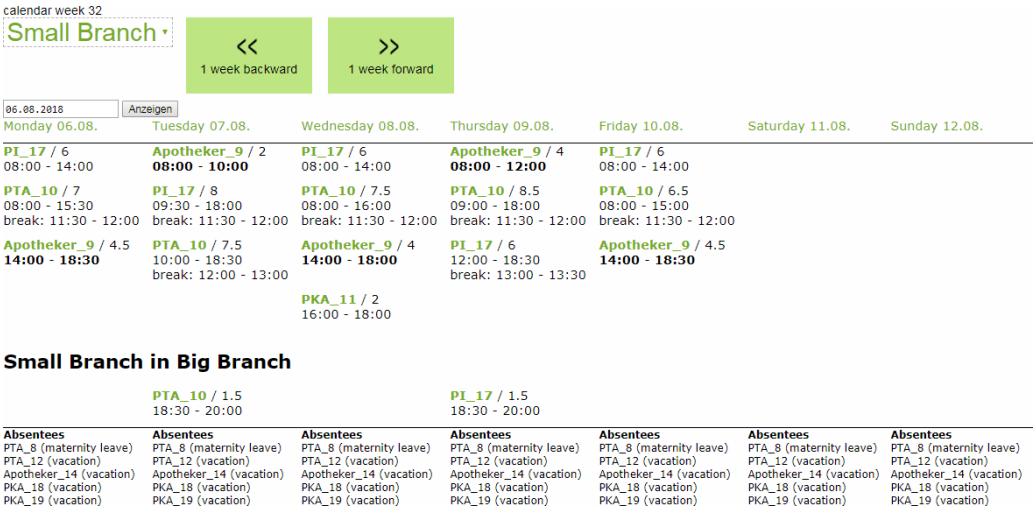


Figure 2.6: Roster week table view, excerpt without task rotation and weekly working hours

The roster week table view shows the roster of a chosen week and branch (Figure 3.6). If employees of the branch are working in an other branch, then those are shown below. The table foot contains the information about absent employees and their reason of absence.

The date can be chosen by direct input. It can also be shifted by one week backwards or forwards by pressing Ctrl + ⇧ + → or Ctrl + ⇧ + ← respectively.

2.1.6 Roster daily view

Read only

In the daily roster view there is a table, a bar plot and a histogram reflecting the roster. The roster table lists all the employees scheduled in the chosen branch on the one chosen day. For every entry there is the employee id and last name, the working hours, the start and end of duty and the time of the lunch break, if any.

If an employee, that is primarily scheduled in the chosen branch, works in an other branch, then this entry is shown in the table at the bottom. An employee may have more than one entrie per day. This allows divided working time to be stored. If employees are absent, these absences are displayed in the table footer.

The roster bar plot shows the flow of employees coming and going. Each bar represents one entry. It reaches from the start of duty to its end. A white rectangle on the bar shows the time of the lunch break. The color of the bars is dependent on the profession of the employee. Pharmacists and Pharmazieingenieure [1] are colored in dark green, while Pharmacy technicians are colored in light green. Other employees (non-pharmaceutical personnel) are colored in grey.

The histogram plot shows a red area and a green line. The red area shows the expected amount of work (measured in packages per 15 minutes), while the green line represents the amount of working employees to any given time.

Edit

The edit page looks quite similar to the read only view. The roster is examined for errors. If any issues occur, then errors, warnings or information will be shown in the top right area. The examination includes:

- overlap of shifts for the same employee (Error)
- sufficient employee count (Warning, hardcoded at least two employees)
- attendance of at least one pharmacist at any time (Error).
- attendance of at least one person able to carry out goods receipt (Warning).
- scheduling of absent employees (Error)
- non-scheduling of non-absent employees (Warning)

Only one break can be inserted per entry. If more breaks have to be assigned, then it is possible to enter multiple entries for the same employee.

2.1.7 Roster employee view

2.1.8 Overtime

2.1.9 Absence

There are four views to the absence data.

- Employee view readonly
- Employee view edit

[1] specific eastern german profession, see https://de.wikipedia.org/wiki/Pharmazieingenieur

- Monthly table
- Year overview

In the *Employee view readonly* there is a select element to choose the employee to view. There is a button to switch to the edit view. And there is a table containing the absence data. The columns are start and end of the absence, reason of absence and number of days. There is a distinct list of possible reasons (vacation, remaining holiday, sickness, sickness of child, unpaid leave of absence, paid leave of absence, parental leave and maternity leave). The number of days of absence is calculated for a 5 day week. Absences on saturdays and sundays are registered but not counted. The same applys for holidays.

Chapter 3

Benutzerhandbuch

3.1 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.

3.1.1 Login

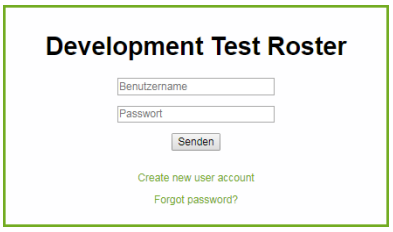


Figure 3.1: Login Seite

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.

3.1.2 Neuen Benutzer-Account erstellen

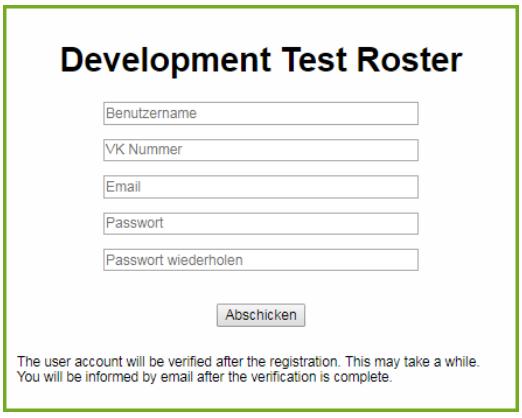


Figure 3.2: Registrierungsseite

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.

3.1.3 Passwort vergessen

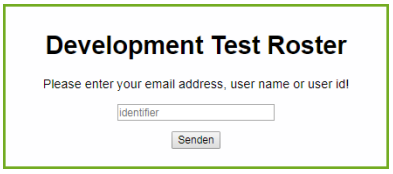


Figure 3.3: Passwort vergessen Seite

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

Figure 3.4: 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.

3.1.4 Navigation

Figure 3.5: 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 3.5).

3.1.5 Dienstplan Wochenansicht

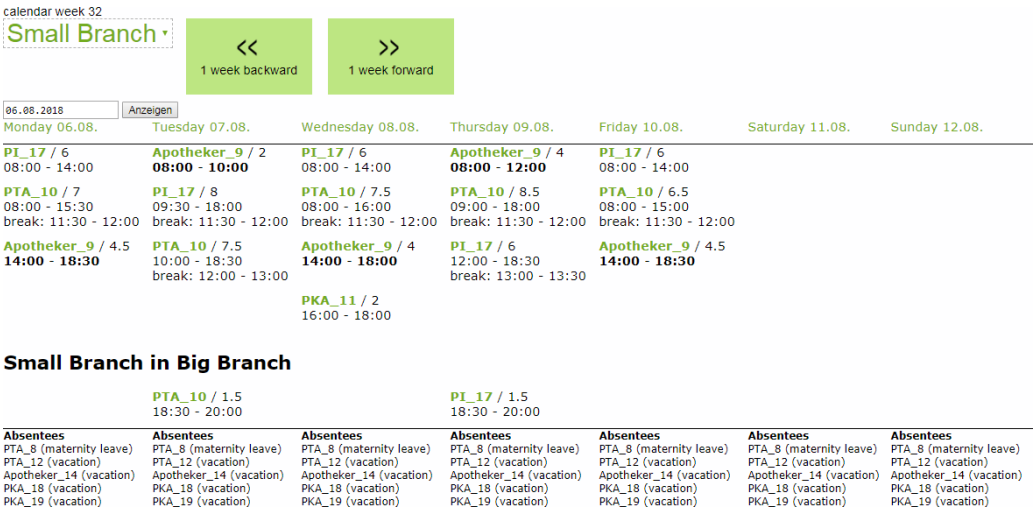


Figure 3.6: Dienstplan Wochenansicht, Auszug ohne Aufgabenrotation und wöchentliche Arbeitszeit

Die Dienstplan-Wochenansicht zeigt die Liste einer ausgewählten Woche und Zweigstelle (Abbildung 3.6). Wenn Mitarbeiter der Zweigstelle in einer anderen Zweigstelle 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.

3.1.6 Dienstplan Tagesansicht

Schreibgeschützt

In der täglichen Dienstplanansicht gibt es eine Tabelle, ein Balkendiagramm und ein Histogramm, welche den Dienstplan widerspiegeln. 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 ähnelt der schreibgeschützten Ansicht. 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.

3.1.7 Mitarbeiterliste des Dienstplans

3.1.8 Überstunden

3.1.9 Abwesenheit

Es gibt vier Ansichten für die 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.

Chapter 4

Administrator manual

4.1 Installation

4.1.1 Getting PDR

The latest release of PDR is available on GitHub
You can also get the latest stable version via git:

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

The master branch is tested to be stable.

4.1.2 The installer

Introduction

The first page shows some non-technical information about this program. Click Next to move on.

Welcome

On the second page some technical background information is given. You are informed about the necessary input data, required for continuing the installation. Available database management systems (currently only MySQL) are listed. Finally, you are informed about the user and password strategy for the database access. Click Next again, to continue.

Requirements

On the next page the application checks, if all requirements are met. These include a minimum PHP version, some PHP extensions and support for database connections. Also the program needs write access to some of its directories. If problems are found, then a descriptive error message will be shown. It is not possible to continue, until all issues are solved. Click Next again, to continue.

Database configuration

The application now starts to collect configuration data.

- Database type
- hostname

- port (optional)
- username
  An existing database user. The user MUST have the privilege to create a database. The user SHOULD have the privilege to create a less privileged user.
- password
  The database password of the user. If a new user could be created, then a new secure random password will be given to the new user.
- database name

Enter the required data and Submit it.

Administrator configuration

After the database values are set, some information about the administrator is collected:

- User name
  the name used by the administrator to login into the program in the future.
- Last name
  this name is connected to the employee id.
- Employee id
  this is used to create an employee, who is connected to the administrative user.
- Contact email address
  is used for questions and comments from the users. Also this email will receive some internal information from the roster.
- Administrator password
  the password used by the administrator to login into the program in the future.

Please register the administrator and click Submit. The data will be written to the file `config/config.php`. For every user, that uses the program, there has be be exactly one employee.

4.1.3 First steps

After submitting the administrator configuration, you will be forwarded to the login page. Login with your administrator credentials.

On your first login you will be prompted with the branch management. Please create at least one branch. You can reach this page at all times in the menu Administration › Branch Management.

The next logical step is to setup some more employees in the Administration › Human resource management.

After all the employees are inserted, you can just start to write rosters (Daily View › Daily input) or you might create principle rosters for specific weekdays (Daily view › principle roster daily) or for distinct employees (Employee › Principle roster employee).

4.2 Upgrading

Until now, there is no automatic update mechanism established. You can regularly download release packages from GitHub. Or you can stay in touch via git:

```
git pull origin master
```

CAVE: Make sure, that you keep your config/config.php! It should not be changed by git, because it is listed in the .gitignore file of this project.

4.3 Configuration

You can manually edit the file config/config.php. The default values are:

```
'application_name' => 'PDR',
'database_management_system' => 'mysql',
'database_host' => 'localhost',
'database_name' => '',
'database_port' => 3306,
'database_user' => '',
'database_password' => '',
'session_secret' => '',
'error_reporting' => E_ALL,
'display_errors' => 0,
'log_errors' => 1,
'error_log' => PDR_FILE_SYSTEM_APPLICATION_PATH . 'error.log',
'LC_TIME' => 'C',
'timezone' => 'Europe/Berlin',
'language' => 'de_DE',
'mb_internal_encoding' => 'UTF-8',
'contact_email' => '',
'hide_disapproved' => FALSE
```

Never delete the first two lines! If the file does not start with "<?php" then PHP will not handle it, meaning that anyone can read its content. Administration » Human resource management

4.4 Maintenance

4.5 Issues and Troubleshooting

Chapter 5

Developer manual

5.1 Core development

All PHP scripts have a common file `default.php`, which handles the default settings. It is placed at ./, which is the `PDR_FILE_SYSTEM_APPLICATION_PATH`. See the file below:

```
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
       as determined by the position of the default.php
22  */
23 define('PDR_FILE_SYSTEM_APPLICATION_PATH', __DIR__ . '/');
24 /**
25  * @var PDR_HTTP_SERVER_APPLICATION_PATH The relative path of the application
       root on the web server.
26  */
27 $folder_tree_depth_in_chars = strlen(substr(getcwd(), strlen(__DIR__)));
28 $root_folder = substr(dirname($_SERVER["SCRIPT_NAME"]), 0, strlen(dirname(
       $_SERVER["SCRIPT_NAME"])) - $folder_tree_depth_in_chars) . "/";
29 define('PDR_HTTP_SERVER_APPLICATION_PATH', $root_folder);
30 //TODO: This does not work, if the location is a symbolic link.
31 /**
32  * @var PDR_ONE_DAY_IN_SECONDS The amount of seconds in one day.
33  */
34 define('PDR_ONE_DAY_IN_SECONDS', 24 * 60 * 60);
35
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.' .
       $class_name . '.php';
41 });
42
43
44 if (!file_exists(PDR_FILE_SYSTEM_APPLICATION_PATH . '/config/config.php')) {
45     header("Location: " . PDR_HTTP_SERVER_APPLICATION_PATH . "src/php/pages/
       install_page_intro.php");
46     die("The application does not seem to be installed. Please see the <a href='
       " . PDR_HTTP_SERVER_APPLICATION_PATH . "src/php/pages/install_page_intro.php
       '>installation page</a>!");
47 } else {
48     /*
49      * Load configuration parameters from the configuration file:
50      */
51     require_once PDR_FILE_SYSTEM_APPLICATION_PATH . "config/config.php";
52     /*
53      * Complement the configuration array with the default values for unset
       parameters:
54      */
55     foreach (configuration::$List_of_configuration_parameters as $key => $value)
       {
56         if (!isset($config[$key])) {
57             $config[$key] = $value;
58         }
59     }
60 }
61 /*
62  * Setup if errors should be reported to the user, if to log them, and where:
63  */
64 ini_set('display_errors', $config['display_errors']); //Display errors to the
       end user?
65 ini_set('log_errors', $config['log_errors']); //Log errors to file?
66 if ($config['log_errors'] or $config['display_errors']) {
67     /*
68      * Debug mode
69      */
70     ini_set('zend.assertions', 1); //Assertions will be compiled AND executed.
71     ini_set('assert.exception', 1); //An exception will be thrown if an
       assertion fails.
72 } else {
73     ini_set('zend.assertions', -1); //Assertions are not compiled.
74     ini_set('assert.exception', 0); //Only warnings would be shown if assertions
       were to be executed and failed.
75 }
76 ini_set('error_log', $config['error_log']); //Which file should errors be logged
       to?
77 error_reporting($config['error_reporting']); //Which errors should be reported?
78
79 /*
80  * We want some functions to be accessible in all scripts.
81  */
82 require_once PDR_FILE_SYSTEM_APPLICATION_PATH . "funktionen.php";
83
84 /*
85  * Setup the presentation of time values:
86   //setlocale(LC_ALL, 'de_DE'); // Leider versteht die Datenbank dann nicht mehr
       , was die Kommata sollen.
```

```
87  */
88 setlocale(LC_TIME, $config['LC_TIME']);
89 /*
90  * Setup default timezone for date()
91  */
92 date_default_timezone_set($config['timezone']);
93 /*
94  * Setup the encoding for multibyte functions:
95  * This is necessary for the usage of UTF-8 characters in functions like
      mb_substr()
96  */
97 mb_internal_encoding($config['mb_internal_encoding']);
98 require_once PDR_FILE_SYSTEM_APPLICATION_PATH . 'src/php/localization.php';
99
100
101 /*
102  * session management
103  */
104 $session = new sessions;
105
106 $List_of_branch_objects = branch::read_branches_from_database();
107 /*
108  * Guess the navigator (=browser) language from HTTP_ACCEPT_LANGUAGE:
109  * This is used in the head.php
110  */
111 $navigator_languages = preg_split('/[,;]/', filter_input(INPUT_SERVER, '
      HTTP_ACCEPT_LANGUAGE', FILTER_SANITIZE_STRING));
112 $navigator_language = $navigator_languages[0]; //ignore the other options
```

../default.php

5.1.1 Directory structure

- 🗀`config/` Contains the configuration file config.php
- 🗀`css/` *obsolete*, use 🗀`src/css/` instead
- 🗀`docs/` This documentation and tools to build it
- 🗀`img/` Images used by the program
- 🗀`js/` *obsolete*, use 🗀`src/js/` instead
- 🗀`locale/` translation files for gettext, currently only german (de_DE)
- 🗀`src/` Most of the actual source code
  - 🗀`src/css` Style Sheets
  - 🗀`src/js` JavasScript
  - 🗀`src/php/` PHP: Hypertext Preprocessor
  - 🗀`src/php/classes/` Contains all the class files class.class_name.php
  - 🗀`src/php/fragments/` parts of bigger pages, may be included via php require/include or loaded with JavaScript
  - 🗀**`src/php/pages/`** This is the place for the single views, which the human user will use to look at the roster etc.

  - 🗀`src/sql/` SQL Database Tables and Triggers

- 🗀`tests/` Tests to find errors in the source code; This folder is listed in .gitignore. Only some files are part of the visible source.
- 🗀`tmp/` A directy for temporary files. There is no automatic cleanup yet.
- 🗀`upload/` The destination for uploaded content. Currently only specific *.PEP files produced by Awinta ASYS Smart are understood. Those files contain information about the amount of customers that have been served in the past.

5.1.2 Coding standards

This project aims to follow some coding style guide.

- Please avoid StudlyCaps and camelCase.
- Class constants MUST be declared in all upper case with underscore separators.
- Property names MUST be written in under_score.
- Plain variables and objects are written in all lowercase.
- Array names start with a singe Uppercase letter followed by lowercase characters.
- Method names MUST be written in under_score.
- Code MUST use 4 spaces for indenting, not tabs.
- Opening braces for classes and functions MUST go on the same line, and closing braces MUST go on the next line after the body.
- Opening braces for control structures SHOULD go on the same line, and closing braces MUST go on the next line after the body.

5.1.3 The database

Currently there is only MySQL supported as a database management system (DBMS). The tables are:

- absence (illness, vacation and other kinds of absence)
- approval (stores for every day, if the leader officially authorized the roster)
- branch (information about the main pharmacy and possible branches)
- Dienstplan (the actual roster data; start, end, break)
- employees (employee data; employee_id, name, profession, abilities)
- employees_backup (a copy of the employees table with historical data archived)
- Feiertage (obsolete)
- Grundplan_roll (not used yet)
- Grundplan (the principle roster; start, end, break; used to propose new rosters)

- maintenance (obsolete)
- Mandant (obsolete)
- Notdienst (dates of emergency services and the employees scheduled to them)
- opening_times_special (not used yet)
- opening_times (the opening and closing times of the branches, no GUI yet for editing)
- pdr_self (reflecting the state of the application itself)
- pep_month_day (the relative amount of work between different days in the month)
- pep (the raw amount of work data, hashed to reduce the amount of deleted/ignored entries)
- pep_weekday_time (the amount of work at different times on different weekdays)
- pep_year_month (the relative amount of work on different months in the year)
- saturday_rotation (whose turn is it to work on which saturday?)
- saturday_rotation_teams (who belongs to which team for saturdays rotation?)
- Schulferien (not used yet)
- Stunden (overtime archive and balance)
- task_rotation (rotating assignment of employees to a task, e.g. compounding)
- user_email_notification_cache (not used yet)
- users_lost_password_token (tokens provided to change a forgotten password)
- users_privileges (the privileges of the user accounts)
- users (the user accounts; there has to be exacly one employee for every user account; there may be employees without user accounts)
- Wunschplan (obsolete)

A copy of all the table structures is stored in 🗀`src/sql/`. The directory also contains the file 🗀`src/sql/database_version_hash.php` which holds a SHA1 hash of all the structures returned by `SHOW CREATE TABLE` and `SHOW CREATE TRIGGER` after some modification. The hash is written by 🗀`tests/get-database-structure.php`, see the details in that file.

Maintenance of the database

There is a class *update_database*. This class holds a defined set of MySQL statements that alter the database structure from a known state in the past to the current state.

This class is not well tested. It might work. It might as well destroy the whole database.

The class *update_database* is called on every login of a user. It then decides on its own, if any actions have to be taken. In order to decide, the hash stored in the file 🗀`database_version_hash.php` is compared to the hash stored in the database table pdr_self › pdr_database_version_hash.

Auto healing tables The class *database_wrapper* has a function *create_table_from_template()* that is able to create missing tables from the structure information given in `src/sql/`. It is called if any PDO database query throws an exception with the code 42S02 and the MySQL error 1146.

5.2 Documentation

This documentation about a programm, app or script is a stub. You can help this project by expanding it.

5.3 Testing

5.4 Bug tracker

5.5 Translation

See this article about po4a about the translation of this document:

5.5.1 Internationalization

Different counties have different laws regarding pharmacies and employment. They also have different holidays.