



Contents

About	2
Requirements	2
Installation	2
PHP Configuration	3
More about FactoryDesk – Admin Panel.....	4
Managing tickets	5
Device name and username	5
File uploads.....	6
Source code	6

About

Simple ticketing system designed for manufacturing environments where production workers need to contact specific department but they don't have AD/email accounts. Ticket request form consists of three dependent select fields where production worker can pick options in the following order: Manufacturing zone (where problem occurs) -> position (where worker performs his duties) -> problem.

Workers can also:

- ➔ add short message (250 characters) to describe their problem (optional)
- ➔ attach files (up to 3 files, 5MB each – default)
- ➔ select priority (notification, standard, critical)

Requirements

- PHP 8+
- Composer
- For SQL Server:
 - + PHP SQLSRV 80 Non-Thread Safe x64 (PHP 8.1 => SQLSRV 81, PHP 8.2 => SQLSRV 82, etc.)
// <https://docs.microsoft.com/en-us/sql/connect/php/download-drivers-php-sql-server?view=sql-server-ver15>
 - + MS ODBC Driver for SQL Server (tested on v17) -> <https://www.microsoft.com/en-us/download/details.aspx?id=56567>
- Url Rewrite 2 for IIS users
- mod_rewrite for Apache users
- Up to date SSL certs for Teams notifications (curl)

Installation

1. Download/clone repository -> <https://github.com/shepard153/FactoryDesk>
2. From the root folder of application run **composer update** command.

```
DB_CONNECTION=mysql          // your DB driver (pgsql for Postgres, sqlsrv for SQL Server)
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=factorydesk
DB_USERNAME=root
DB_PASSWORD=root
```

3. Setup your .env file with database configuration. Example:
4. Run following commands:

```
php artisan migrate
php artisan serve
```

5. To change app name from default just edit **APP_NAME** in .env file
6. Remember to turn **APP_DEBUG** to **false** when running on production!
7. Default login for admin panel is root/root.

PHP Configuration

SQL Server only: Copy your PDO driver to **ext** folder located in your main PHP directory (eg. **php_pdo_sqlsrv_81_nts_x64** for PHP 8.1).

Open **php.ini** file and go to extensions. Comment out extensions like in the image below. If you are using Postgres, uncomment **pdo_pgsql**. For SQL Server you need to add your driver to extensions.

```
;extension=bz2
extension=curl
;extension=ffi
extension=ftp
extension=fileinfo
;extension=gd
;extension=gettext
;extension=gmp
;extension=intl
extension=imap
;extension=ldap
extension=mbstring
;extension=exif      ; Must be after mbstring as it depends on it
extension=mysqli
;extension=oci8_12c  ; Use with Oracle Database 12c Instant Client
;extension=oci8_19  ; Use with Oracle Database 19 Instant Client
;extension=odbc
extension=openssl
;extension=pdo_firebird
extension=pdo_mysql
;extension=pdo_oci
;extension=pdo_odbc
;extension=pdo_pgsql
extension=php_pdo_sqlsrv_81_nts_x64
;extension=pdo_sqlite
;extension=pgsql
;extension=shmop

; The MIBS data available in the PHP distribution must be installed.
; See https://www.php.net/manual/en/snmp.installation.php
;extension=snmp

extension=soap
;extension=sockets
```

More about FactoryDesk – Admin Panel

Dashboard – displays statistics for department your user is currently assigned to. Newest tickets table displays top 5 (can be changed via **Settings** menu) newest tickets. Statistics are refreshed every 30 seconds (also can be changed).

Awaiting tickets – same as tickets but with **awaiting** filter applied for faster navigation.

Tickets – full list of tickets assigned to your department.

My tickets – view with short summary and history of tickets assigned to/resolved by currently logged in user.

Reporting – place where you can generate report for selected period of time and export only the data you need by selecting desired checkbox fields. Exported filename is a combination of **APP_NAME** env value and current date, eg: **FactoryDesk_14100715120000.csv**

Users (administrator) – here you can create, modify and delete FactoryDesk staff accounts.

Departments (administrator) – create, modify, delete departments. Attached images are displayed as department icons when raising a ticket. By default all uploads go to:
"**__APP_ROOT__\storage\app\public\departments_img**"

Form editor (administrator) – using form editor you can modify ticket request select fields. Since they are dependent, remember to create zones before positions and positions before problems.

Settings – global settings where you can change how dashboard behaves.

My profile – place where staff members can change their passwords.

Managing tickets

The screenshot shows a 'Ticket details' form with the following fields and controls:

- Tabbed interface:** 'Ticket' (selected), 'Add note', 'Ticket history'.
- Metadata:** Created 2022-07-18 16:09:36, Taken 2022-07-18 16:25:25, Closed -----, Status **In progress** (yellow badge).
- Form fields:** Device (Komputer), Raised by (User), Zone (Production line 3), Position (Assembly line).
- Dropdowns:** Department (Quality), Problem (Invalid size of component), Priority (Standard).
- Time spent on resolving:** 00:50, with buttons for + 5 minutes, + 15 minutes, + 30 minutes.
- Assigned to:** A dropdown menu.
- Buttons:** Save changes (green), Close ticket (red).

All ticket info is displayed in this view. Changes made must be submitted using **Save changes** button. **Close ticket** button won't save changes you've made!

Closed ticket can be reopened in 48 hours from closing using **Reopen ticket** button. When timer expires, ticket status will change to **Closed permanently**.

In the ticket details menu you can also add attachments, notes and view edit history. Image attachments are displayed as thumbnails and can be clicked to enlarge. Rest of the files (*.doc, *.rar, etc.) are available for download.

Device name and username

Device name is retrieved when client's computer is in same AD network as PHP server. If not, IP number is displayed instead. I used this mostly for easier remote connection as I didn't have to ask users for their PC name/IP address or walk to assembly line just to find out that one of the icons "disappeared".

Username field is hard coded and always displays "User" as some more tweaking is needed to make AD controller respond with currently logged in username. In my environment I've made PHP script that receives device name AND THEN asks AD controller for the username of currently logged in person on that workstation. I've removed this script from release version since some company specific data was included so you have to figure it out on your own how to make this work in your work environment.

Another option is to enable user input so they can enter username. You can do this by going to "`__APP_ROOT__\resources\views\ticket_step2.blade.php`". Simply remove **readonly** attribute and change **value** to **placeholder** in the line below:

```
<input type="text" name="username" value="User" class="form-control" readonly required>
```

File uploads

By default all dropzone uploads are stored in:

"__APP_ROOT__\storage\app\public\ticket_attachments".

If you want to store them in the cloud (eg. AWS) you need to properly configure your .env file with desired **FILESYSTEM_DRIVER** and AWS credentials. Check [Laravel Filesystem](#) for more info.

Source code

There is no single raw SQL query in the code. All queries use Laravel's Eloquent ORM and it is highly advised to read some documentation before making any edits in DB queries. Link below.

[Laravel Eloquent ORM](#)

Same thing goes for the "HTML files". Every view is stored in __APP_ROOT__\resources\views and has *.blade.php extension. You can read more about Blade in the link below.

[Laravel Blade Templates](#)