

## Table of contents

About this manual .....	2
Configurations .....	3
Auto logout timeout .....	3
Cache settings.....	3
Database connection settings .....	3
Debug mode .....	3
Email settings.....	3
Error level settings.....	3
Landing page / routes settings .....	3
Layout / theme / colors / images .....	4
Logging settings .....	4
Metric types.....	4
Metric type names.....	4
Pagination settings .....	4
Predictive working hours charts .....	4
Statistics page (metrics and working hours tables).....	4
Working hour / work type descriptions .....	6
Guide for future MMT developers .....	7
TODO list with some ideas and uncompleted features.....	7
Known bugs .....	7
Roles .....	8
User roles:.....	8
Member roles: .....	9

## **About this manual**

This manual is mainly meant for admins, future MMT developers and possibly supervisors. The information needed for normal project work use of MMT can be found inside MMT in the FAQ page in form of videos and FAQ answers. General information about MMT can be found inside MMT in the About MMT page.

## Configurations

Default database template (MMT Database.sql) found in sql folder is ready to use as it comes with metric types, work types and 20 anonymous users predefined. It also has one admin user with email **admin@admin.com** and password **adminadmin**. This password should be changed.

Most server and general environment related settings are configured in config folder, mainly in files config\app.php and config\bootstrap.php. Settings such as metric type names and project's default length are usually configured in the model or controller files related to the specific setting.

### Auto logout timeout

Timeout value is set in config\bootstrap.php, line 209 (on CakePHP 3.5.18).

### Cache settings

High level settings in config\app.php, from line 84 (on CakePHP 3.5.18). Possible additional settings in config\bootstrap.php. These settings affect for example the duration that data persists unchanged in tmp\cache folder.

### Database connection settings

In config\app.php, from line 216 (on CakePHP 3.5.18). Information such as host address, database name, username and password need are entered here.

### Debug mode

In config\app.php, line 12 (on CakePHP 3.5.18). It is recommended to use debug mode in development environment.

### Email settings

In config\app.php, from line 177 (on CakePHP 3.5.18). Information such as host address, port, username and password need are entered here.

### Error level settings

In config\app.php, from line 150 (on CakePHP 3.5.18). This defines the level of errors displayed in debug mode. You can for example disable errors related to deprecated version notifications.

### Landing page / routes settings

In config\routes.php. Set default controller action for landing page (address '/') and default action for the rest of URLs.

## **Layout / theme / colors / images**

Most user defined settings are done in webroot\css\cake.css with some individual stylings (for example table column width) done also directly in view template files. All the images such as the main logo are in folder webroot\img.

## **Logging settings**

In config\app.php, from line 284 (on CakePHP 3.5.18). Here you can set the log files location and the level of messages to be logged.

## **Metric types**

These are configured in the database in table 'metrictypes'. Admin can edit these in MMT from page /metrictypes. These names / descriptions must match the ones in src\Controller\MetricsController.php and in src\Template\Metrics\addmultiple.ctp.

## **Metric type names**

In src\Controller\MetricsController.php function getMetricNames(). Changing the metric type name here changes it in the program globally, except in few charts where they are hard coded. These can be changed in src\Controller\ChartsController.php.

## **Pagination settings**

Some pagination settings (such as order and number of entries per page) can be defined in the controller files. For example project listings settings are configured in src\Controller\ProjectsController.php index() function.

## **Predictive working hours charts**

Model is in src\Model\Table\MembersTable.php, controller in src\Controller\MembersController. In case where project ending date is not defined, the default length of chart is set to be 20 weeks from project's start date (\$endingDate). Line colors and data point symbols are also configured in the model. Chart texts are configured in the controller.

## **Statistics page (metrics and working hours tables)**

The threshold values are configured in src\Model\Table\ProjectsTable.php function getStatusColors. They are based on actual averages of data collected from earlier MMT use. The projects are split into 4 phases (phase 1 = weeks 1-5, phase 2 = weeks 6-10, phase 3 = weeks 11-15 and phase 4 = weeks 16+) with the threshold values varying by phase. In case one column has both yellow and red status limits active, red naturally overrides the yellow. The original definitions of these values is as follows:

Phase 1:

Commits: no requirements

Test cases (passed / total): no requirements

Backlog (product / sprint): no requirements

Done: no requirements

Risks (high / total): no requirements

CPI / SPI: no requirements

Minimum working hours of an active member: yellow if 0

Earliest last seen date of an active member: yellow if > 2 weeks from today, red if > 3 weeks from today

Phase 2:

Commits: yellow if < number of weeks from project start, red if 0

Test cases (passed / total): yellow if total = 0

Backlog (product / sprint): yellow if product < 5 or sprint = 0, red if product = 0

Done: no requirements

Risks (high / total): yellow if total < 5, red if total = 0

CPI / SPI: yellow if either < 0,5

Minimum working hours of an active member: yellow if < 2,5 times number of weeks from project start, red if < 1 times number of weeks from project start

Earliest last seen date of an active member: yellow if > 2 weeks from today, red if > 3 weeks from today

Phase 3:

Commits: yellow if < number of weeks from project start, red if < 5

Test cases (passed / total): yellow if total < 5, red if total = 0

Backlog (product / sprint): yellow if sprint = 0, red if product < 5

Done: yellow if = 0

Risks (high / total): yellow if total < 5 or high > 2, red if total = 0 or high = total

Minimum working hours of an active member: yellow if < 5 times number of weeks from project start, red if < 3 times number of weeks from project start

Earliest last seen date of an active member: yellow if > 2 weeks from today, red if > 3 weeks from today

Phase 4:

Commits: yellow if < number of weeks from project start, red if < 5

Test cases (passed / total): yellow if passed = 0, red if total < 5

Backlog (product / sprint): yellow if sprint = 0, red if product < 5

Done: yellow if product backlog \* 0,5 > done

Risks (high / total): yellow if total < 5 or high > 2, red if total = 0 or high = total

Minimum working hours of an active member: yellow if < 5 times number of weeks from project start, red if < 3 times number of weeks from project start

Earliest last seen date of an active member: yellow if > 2 weeks from today, red if > 3 weeks from today

### **Working hour / work type descriptions**

These are configured in the database in table 'worktypes'. Admin can edit these in MMT from page /worktypes. Changing the description value of a work type changes the working hour description in the program globally, except in few charts where they are hard coded. These can be changed in src\Controller\ChartsController.php. Adding new work types requires additional changes to code.

## Guide for future MMT developers

To setup your development environment first you need to obtain the MMT source code, download Composer (<https://getcomposer.org/download/>) and then install CakePHP and its dependencies by running Composer by typing `composer install` in the MMT source code's root folder.

You also need some way of running your local MMT server as well as your local MMT database. Previous MMT developers have used WampServer and XAMPP on Windows systems and MAMP on Mac OSX Mojave. These have proven to be beginner friendly as they come with multiple selectable PHP versions as well as with phpMyAdmin tool to run and edit MySQL databases.

Refer to **Configurations** section of this manual to setup your database connection and any additional configurations necessary. It should be noted that latest MMT updates have been done with PHP 5.6.40, CakePHP 3.5.18 and MySQL 5.7.26. In case of unexpected errors you should first check your versions.

## TODO list with some ideas and uncompleted features

- Make add member form dynamic so that target hours field is only displayed if selected project role is developer or manager
- Implement created\_on and modified\_on functionalities to workinghours (columns already in the database)
- Add date/week picker to add weekly report page 1 to make inputting the correct week clearer
- Add password confirm functionality to sign up form and edit user profile form
- Add GitLab API connection to project
- General code refactoring. The code doesn't fully follow the MVC architecture as it is supposed to. Many functions are too large and should be divided into smaller pieces. There is also some unnecessary repetition in the code. There are some files/pages/functions that are not in use anymore.
- Add some AI to predict project upcoming status and/or metrics
- Add export functionality for project data

## Known bugs

- Earned value charts doesn't display correctly if the estimated 100% hours date or EAC/BAC values become unrealistically large (for example if the projects is abandoned for several months)
- Many of the charts don't display correctly after project is finished

## Roles

First it is important to notice the difference between **user roles** and **member roles**. In MMT every user has a user role of either an admin or a regular user. Ideally there shouldn't be many users with admin role, as the admin role allows the user to modify all kinds of database values shared by all the projects.

New users can register themselves via the sign up link on the home page or they can be added by an admin. A new user can be assigned to a project by a manager of the given project or by any user with supervisor member role or by an admin.

Member roles (except supervisor) are project specific, meaning that one user can for example have manager level access to one project and only developer level access to another project. Supervisor role is not fully specific to a single project, meaning that if a user has supervisor member role in one project, he/she is able to access some additional data in all the public projects.

User's member role can be seen when hovering over the top navbar's "My links"-section. If the user has membership in multiple projects, member role is not displayed until selecting one of the projects. User's member role is also displayed in project's "Members" page.

### User roles:

#### **user**

This role only allows the ability to log in to MMT. All the additional access rights are assigned depending on user's member role (see **Member** roles). User can be a member of 0 – n projects.

#### **admin**

This role allows access to all the MMT's functions. In addition to having full access rights to every function in every project in the database (including the private projects), admin can also modify (add, edit and delete) users, metric types and work types. Only admin can view the private projects. Only admin can view the feedback given by project members. Only admin can use the "anonymize members" function in the project's members page.



### **Member roles:**

All the member roles allow access to the project's full data in which the membership belongs to as well as the ability to give general feedback to admins via the "Give feedback" link in the top navbar. All the member roles also allow access of the statistics page (with limited data) as well as "Project info" and "Charts" (also with limited data) pages of all the public projects.

If a user only has membership in one project, he/she is directed to this project's "Log time" page automatically after logging it.

#### **client**

Client role has no additional access rights.

#### **developer**

Developer role allows the ability to view all the data of the project in which the role is assigned in. In addition to viewing the project's data, a developer can log time to the project, edit the amount of personal target hours and add comments to existing weekly reports.

#### **manager**

In addition to the developer role's rights, manager role allows the ability to edit the data in the project's info page, add new member's to the project, add and edit weekly reports, log time for other members of the project and add and edit risks of the project.

#### **supervisor**

In addition to developer role's rights, supervisor role allows the ability to add and delete members in any public project. It also allows the ability to edit member's role, starting date and ending date in any public project. Supervisor can edit the general data in the assigned project's info page (not the API connections). Supervisor can also log time for any developer or manager in any public project. Supervisor can also see additional data in the global statistics page (weekly report, metrics and working hours data) and in any public project's charts page (earned value charts and chart showing cumulative working hours of each public project in a single chart). Supervisor can't add new weekly reports. Supervisor can create new projects.