# START – Technical overview

## Summary

This document scope is a basic overview of backend of START project (CMS & RESTful API), it describes structure of program modules, deploying and configuration process.

# CMS

## Overview

CMS is PHP 5.4+ and PHP 7.0-7.1 compatible. It is developed using MVC pattern.

## Directories structure

The directories structure of CMS is PSR-4 standard compliant (<http://www.php-fig.org/psr/psr-4/meta/>), which, combined with namespaces support is necessary for Autoloader functionality.

abeautifulsite/

php\_xlsx\_writer/

PHPExcel/

tb/

css/

fonts/

images/

js/

config/

controllers/

helpers/

langs/

models/

views/

vendor/

web/

app/

{www\_dir}/start\_cms/

The CMS is based on the Single Entry Point technique placed at {www\_dir}/start\_cms/index.php. It loads configuration files and invokes CMS.

Configuration files are stored in {www\_dir}/start\_cms/app/config/. There are 3 files – for database settings, for directory paths and for in-app constants.

Models, Controllers and Views are stored in {www\_dir}/start\_cms/app/models/, {www\_dir}/start\_cms/app/controllers/ and {www\_dir}/start\_cms/app/views/ respectively.

Models are supposed to retrieve data, views to display and controllers are responsible for data interchange between them.

The core CMS functionality that do not contain business logic of application is places in {www\_dir}/start\_cms/vendor/tb/start\_cms/. It manages modules loading, user rights assignation and internal CMS architecture.

All web-accessible files (styles, scripts, images and fonts) are stored at {www\_dir}/start\_cms/web/ directory.

## Modules structure

The principal scheme is as shown below:

Models

Views

Controllers

child\_info.php

child\_survey.php

children\_delete.php

children\_list.php

Assessment content

content\_bubbles\_jubbing\_tests.php

content\_choose\_touching\_tests.php

content\_coloring\_tests.php

content\_eye\_tracking\_pairs\_tests.php

…

cms\_users.php

social\_workers.php

cms\_users\_controller.php

social\_workers\_controller.php

Users and Social workers

cms\_user\_add.php

cms\_user\_edit.php

cms\_users\_list.php

social\_workers\_add.php

social\_workers\_edit.php

social\_workers\_list.php

children\_controller.php

children.php

parents.php

surveys.php

Surveys

exercises.php

parent\_assestments.php

exercises\_controller.php

parent\_assestment\_controller.php

The scheme is continuing here:

Models

Controllers

Views

log.php

log\_controller.php

Log

log\_list.php

Statistics

children.php

surveys.php

statistics\_controller.php

statistics.php

Assessment content

exercises.php

parent\_assestments.php

exercises\_controller.php

parent\_assestment\_controller.php

content\_eye\_tracking\_slide\_tests.php

content\_motoric\_following\_tests.php

content\_parent\_child\_play\_tests.php

content\_wheel\_tests.php

parent\_assestment\_add.php

parent\_assestment\_list.php

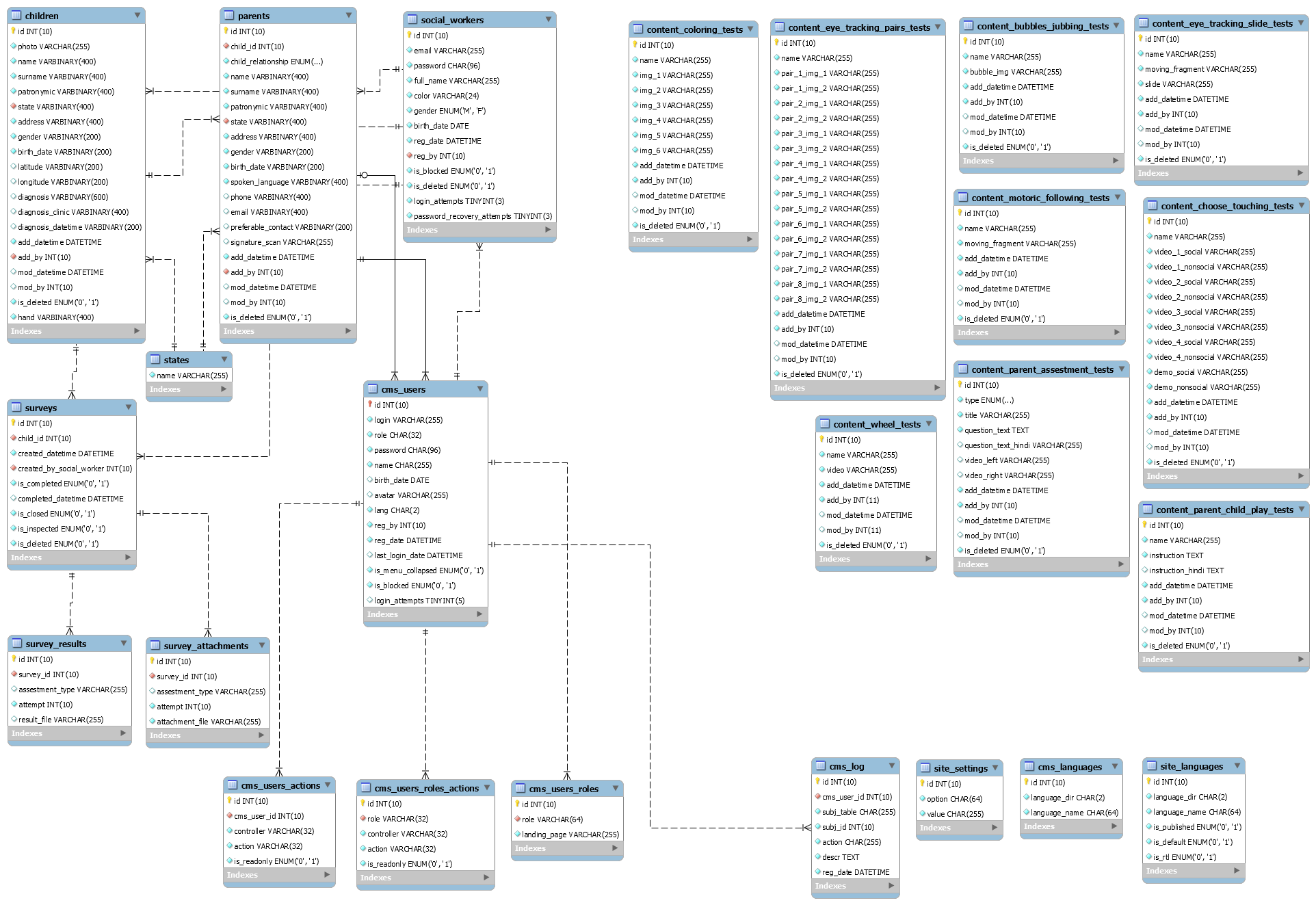
Views

Controllers

Models

## Database structure

Database structure with relations between tables is available in form of ERR diagram, see MySQL Workbench file “Start DB ERR diagram.mwb” in GitHub repository <https://github.com/TherapyBox/Start/blob/master/Start%20DB%20ERR%20diagram.mwb> . The preview of it is shown below:



All personal data of children and parents encrypted using AES algorithm.

Tables explanation:

* `children` - Personal data of children
* `cms\_languages` - List of interface languages of CMS. Used for loading specified files from {www\_dir}/start\_cms/app/langs/{lang\_file}.php
* `cms\_log` - Stores events log performed by CMS users
* `cms\_users` - CMS users
* `cms\_users\_actions` - Stores permitted controller/action list for concrete CMS user. If not specified does not affect role-based permissions. If specified, user permissions intersects with role permissions and result is list of controller/action that presents in both lists. It allows to declare special restrictions for single CMS user without creating new role.
* `cms\_users\_roles` - Stores roles and landing pages for them.
* `cms\_users\_roles\_actions` - Role-based permissions. Pseudo role `all` indicated that defined page has public access (do not requires authentication). For further information, refer to section “CMS user’s rights management”.
* `content\_bubbles\_jubbing\_tests` - Stores “Pop bubbles test” content. Contains only one record. Because of various number of fields each assessment type has own table.
* `content\_choose\_touching\_tests` - Stores “Touch test” content.
* `content\_coloring\_tests` - Stores “Colouring test” content.
* `content\_eye\_tracking\_pairs\_tests` - Contains “Eye-tracking test” content.
* `content\_eye\_tracking\_slide\_tests` - Was supposed to store “Eye-tracking with moving slide” test content. Currently have no handling via CMS, preserved for backward compatibility.
* `content\_motoric\_following\_tests` - Stores “Motor Following test” content.
* `content\_parent\_assestment\_tests` - Stores “Questionnaire” English and Hindi questions. The only content table that has multiple records.
* `content\_parent\_child\_play\_tests` - Stores “Parent-child play” instructions.
* `content\_wheel\_tests` - Stores “Wheel test” video.
* `parents` - Contains parent’s personal data.
* `site\_languages` - Legacy table, can be deleted.
* `site\_settings` - Contains project name, default landing page etc. Settings are loaded on CMS startup and available globally.
* `social\_workers` - Social workers registered in system.
* `states` - Stores list of Hindustani states to show in filter dropdowns and statistics.
* `surveys` - Stores children surveys (child can have multiple surveys), social worker ID by whom it was created and other meta-information about survey.
* `survey\_attachments` - Each survey can have multiple attachments – media files collected during test. This table stores this files names.
* `survey\_results` - Stores json files (their names) received from mobile application with survey results. Each survey can have multiple attempts.

## CMS user’s rights management

The `cms\_users\_roles\_actions` table is used for assign privilegies to certain user roles. There are fields `role`, `controller`, `action`, `is\_readonly`.

The `role` field is related to `cms\_users`.`role` and can store 4 keywords: `admin`, `researcher`, `clinician` and `all`. The `all` keyword means that this page (controller/action) is accessible without authorization. The others are roles of users that have permissions to specified pages.

The `controller` and `action` fields specifies what action is permitted to invoke for this role. For example, if role=`researcher`, controller=`children` and action=`download\_surveys` that means that researcher can access controller {www\_dir}/start\_cms/app/controllers/children\_controller.php and action children\_controller::action\_download\_surveys().

The `is\_readonly` field means that user can only view data on specified page when value is equal to `1`.

User rights management is integrated into CMS. Before resolving any page it engages CMS::getPrivilegiesByRole($role) or CMS::hasAccessTo($controller.'/'.$action) functions and shows 403 Forbidden page if no permissions are found. That’s why before creating any page you must at first create record in this table. When permission is set but there is no such controller/action CMS shows 404 Not found error page.

## PHP extensions required

* php\_bz2
* php\_curl
* php\_fileinfo
* php\_gd2
* php\_gettext
* php\_mbstring
* php\_exif
* php\_mysqli
* php\_pdo\_mysql
* php\_openssl

## Deployment process

Configuration/Deployment process is as described at Readme.md on GitHub <https://github.com/TherapyBox/Start/blob/master/README.md> .

# RESTful API

## Overview

RESTful API is developed using Yii 2 framework. Framework is downloadable at <http://www.yiiframework.com/download/> . Documentation is available at <http://www.yiiframework.com/doc-2.0/> .

There are functional tests implemented and available at {site\_url}/test-api/run .

API uses authorization with JWT token <https://jwt.io/introduction/> .

## Sensitive user’s data

Sensitive user’s data are stored in {www\_dir}/uploads/. They are protected from direct access using .htaccess files.

The list of directories is shown below:

* **{www\_dir}/uploads/children\_photos/** - children photos
* **{www\_dir}/uploads/parent\_signatures/** - parent signatures
* **{www\_dir}/uploads/survey\_attachments/** - media and attachments of surveys
* **{www\_dir}/uploads/survey\_results/** - json files with survey results
* **{www\_dir}/uploads/tests\_content/** - media and files for exercises

These folders are recommended to backup.

## Function references

List of endpoints, expected parameters and response examples available as spreadsheet file at <https://github.com/TherapyBox/Start/blob/master/START%20RESTful%20API%20documentation.xlsx> .

# Licenses

<http://www.yiiframework.com/license/>

<http://getbootstrap.com/> ( <https://creativecommons.org/licenses/by/3.0/> )

<https://adminlte.io/docs/2.4/license>

<http://fontawesome.io/license/>

<http://jquery.com/> ( <https://js.foundation/CLA> )

<http://fancyapps.com/fancybox/3/#license>

<http://bootboxjs.com/>

<https://github.com/claviska/SimpleImage>

<https://github.com/PHPOffice/PHPExcel> (<https://github.com/PHPOffice/PHPExcel/blob/master/license.md> )

<https://github.com/mk-j/PHP_XLSXWriter>