Library Hours and Locations Administration

Environment

This application was developed using CakePHP 1.3, with a MySQL 5.5.20 database, on a RHEL 5.7 Linux server running PHP 5.2.10. The CakePHP core code is not included, but can be downloaded from http://cakephp.org/.

Setup

After installing cake and unzipping the hours_admin.tar.gz file, you will need to create the following directories inside the hours_locations/app/ directory.

```
tmp
tmp/cache
tmp/cache/models
tmp/cache/persistent
```

You will need to make these directories writable by your web server. Depending on your web server, you could use the following commands (from the app directory) to do this:

```
“chmod -R 777 tmp”
“chown -R apache:apache tmp”
```

You may need to delete the files in the models and persistent directories from time to time as you make changes, especially if the database changes.

You will need to edit the paths and URLs in app/webroot/index.php to match those on your server. You will also need to edit app/config/database.php to include your database connection information. You can turn debugging info on and off in app/config/core.php. The home page for the application is set in app/config/routes.php, and is set to the hours_groupings index page (app/views/hour_groupings/index.ctp).

You will need .htaccess files in your root directory, in the app directory, and in app/webroot, for the CakePHP application to work correctly. Examples are included, but you may need to add a RewriteBase line to the .htaccess files or edit the paths. See the next section for how to setup access and authentication. This must be done for the application to function – if you try it before this, you will be redirected, likely to a page not found. Alternatively, you can comment out the redirects in lines 83 and 85 of /app/controllers/hour_groupings_controller.php, though there will still be things that do not work until you set up at least the ‘hours’ user.

Access/Authentication

This application uses Apache basic authentication since the data is not at all sensitive, but has 3 levels of access so that people can only edit data they’re responsible for maintaining. The three levels are super-admin (‘hours’), admin (‘hours_admin’), and branch (each branch has their own login, stored in the hour_locations table). There must be a password for each of these in a .htpasswd file that the app/webroot/.htaccess file refers to for authentication and authorization.

Layout/Styles

To change the default layout for the application, edit app/views/layouts/default.ctp. The main stylesheet is app/webroot/css/cake.ubc.css. There are templates for “flash” confirmation messages in app/views/elements.

Architecture

CakePHP uses a Model, View, Controller (MVC) architecture. For each database table, there will be a file in app/controllers with most of the logic and queries needed for each page, a file in app/models with data validation and table relationships, and a directory in app/views with all of the display code related to that table. I’ve followed CakePHP conventions for the most part, so the CakePHP documentation at http://book.cakephp.org/view/876/The-Manual should be helpful. There is also a controller, model, and view for “hour_print” for which there is not a table in the database.

I recommend looking at the database design document and/or image before delving into the code, since the code is all based around interacting with the database. There are no admin tools for editing the hour_types and hour_categories lookup tables, so those will need to be maintained directly in the database.

There are two files outside the CakePHP application (not in the app directory) that are used for AJAX calls: hours_autocomplete.php and hours_date_range_lookup.php.

The database connection used in hours_date_range_lookup.php is from another application, so you will need to use your own database connection file for this. The global variable $staffdb is a PHP: DBO database handle. The getArray() method called on line 29 is from another application, so I'm including it here:

```
/* @param string $sql SQL query
 * @param mixed $bind Bind variables
 * @return array
 */
function getArray($sql, $bind=false, $ignoreError=false) {
    $stmt = $this->_query($sql, $bind, $ignoreError);
    return $stmt->fetchAll(PDO::FETCH_ASSOC);
}
```

Please refer to comments within the code for more details.