

**JUDGE FROG**

Developer Guide

***V2.0***

**5/3/2015**

# Revision History

|  |  |  |
| --- | --- | --- |
| Version | Changes | Edited |
| 1.0 | * Initial Draft | March 15, 2015 |
| 2.0 | * Added Arvixe, CakePHP sections | May 3, 2015 |

# Revision Sign-Off

By signing the following, the team member is stating that he has read the entire document and has verified that the information contained within this document is accurate, relevant to the project, and void of errors.

|  |  |  |
| --- | --- | --- |
| Name | Signature | Date Signed |
| Brice Boula |  |  |
| Collin Duncan |  |  |
| David Tomlinson |  |  |
| Landon Westrom |  |  |

# Table of Contents

[Revision History ii](#_Toc418434346)

[Revision Sign-Off ii](#_Toc418434347)

[Table of Contents iii](#_Toc418434348)

[2. Introduction 4](#_Toc418434349)

[1.1 Purpose 4](#_Toc418434350)

[1.2 Overview 4](#_Toc418434351)

[3. System Overview 5](#_Toc418434352)

[2.1 System Components 5](#_Toc418434353)

[2.1.2 MySQL Database 5](#_Toc418434354)

[2.1.3 Web Host 5](#_Toc418434355)

[4. Arvixe 6](#_Toc418434356)

[3.1 Overview 6](#_Toc418434357)

[3.2 Oyster 6](#_Toc418434358)

[3.2.1 cPanel & Apache 6](#_Toc418434359)

[3.2.2 PHP 6](#_Toc418434360)

[3.2.3 MySQL 6](#_Toc418434361)

[3.2.4 FTP 6](#_Toc418434362)

[3.2.5 Subdomain Management 6](#_Toc418434363)

[5. MySQL Database 7](#_Toc418434364)

[6. CakePHP 7](#_Toc418434365)

[5.1 Overview 7](#_Toc418434366)

[5.2 Configuring CakePHP 7](#_Toc418434367)

[5.3 Useful References 8](#_Toc418434368)

[7. Models 8](#_Toc418434369)

[8. Controllers 8](#_Toc418434370)

[9. Glossary of Terms 9](#_Toc418434371)

# Introduction

## 1.1 Purpose

The purpose of this document is to provide developers with the appropriate information so that they can modify the Judge Frog source code to meet their needs and requirements. Each major component of the application will be broken down and outlined.

## 1.2 Overview

This document includes the following sections.

**Section 2 – System Overview:** Covers the main components of the system.

**Section 3 – Development Getting Started:** Covers an introduction to programming of the CakePHP web application.

**Section 4 - Glossary of Terms:** Includes a list of abbreviations and their technical terms and their associated definitions.

# System Overview

## 2.1 System Components

Judge Frog is comprised of the CakePHP web application, MySQL database, and a web host.

**2.1.1 CakePHP Web Application**

The CakePHP web application is used as a framework for a foundation for the web application. It allows for a MVC structured model of the developed web application.

### 2.1.2 MySQL Database

The MySQL database contains all the case information and user login information. The database is stored on the web host’s server.

### 2.1.3 Web Host

The web host the project relies on is Arvixe. Three domains are purchased from Arvixe including; humantraffickingdata.org, humantraffickingdata.com, humantraffickingdata.net. The web application and database are stored on a Linux class server.

# Arvixe

## 3.1 Overview

The Judge Frog project contracted with a web host called Arvixe (<http://www.arvixe.com/>) to provide an Apache server for the project. Please see Dr. Bouche for the credentials necessary to access Arvixe.

## 3.2 Oyster

### 3.2.1 cPanel & Apache

The Human Trafficking Data website is hosted on an Apache server named Oyster. To access the server, the URL is [www.oyster.arvixe.com](http://www.oyster.arvixe.com). Use the cPanel login option and enter the admin credentials given to you by Dr. Bouche. As of the writing of this document, the Apache version running is **2.2.27**.

### 3.2.2 PHP

The PHP version running on the server is listed as **5.3.28** on the cPanel stats menu, but you can choose to run a different version of PHP for a given directory on the server. We have done so, and the PHP version running on HTD is **5.5**. To customize the PHP version, use the **PHP Selector** utility in the Software/Services section of the cPanel.

### 3.2.3 MySQL

The current MySQL version as of this writing is **5.5.42-37.1**. There are several utilities for administrating the MySQL database on Oyster. First, to modify, check, or repair a database, use the **MySQL Databases** tool in the Databases section. A list of current databases is displayed. Here you can also control the accounts that our application uses in order to access the database. For interfacing with the database directly, **phpMyAdmin** is available.

### 3.2.4 FTP

For managing files on the server, a standard FTP utility is available in the Files section, called **File Manager**. Here you will see a listing of all directories available to us on Oyster. All files related to HTD are stored in the **public\_html** directory.

### 3.2.5 Subdomain Management

For management of subdomains on Oyster, there is a utility called **Subdomains** in the Domains section. Here you can set custom subdomains and direct them to look at specific directories.

# MySQL Database

# CakePHP

## 5.1 Overview

HTD is written in primarily PHP, using a framework called CakePHP ([www.cakephp.org](http://www.cakephp.org)). The CakePHP version used in developing HTD was 2.6.1, due to limitations at the time. CakePHP is a Model-View-Controller (MVC) based framework. Documentation for CakePHP is available in the form of the CakePHP [Cookbook](http://book.cakephp.org/2.0/en/index.html). The Cookbook contains a detailed description of coding conventions and the structure of MVC interactions in CakePHP.

## 5.2 Configuring CakePHP

Configuring CakePHP is relatively straight-forward. To download a clean version of CakePHP, go to the GitHub repository of CakePHP, available at [www.github.com/cakephp/cakephp/tags](http://www.github.com/cakephp/cakephp/tags). Every version of CakePHP is available via the release tags on this page, so choose the 2.6.1 tag. You can download an archive of the release in various forms (zip/tar.gz). For development, you can simply add this CakePHP application directory to your own Apache server config. To see more detailed setup and configuration instructions, see the [Installation](http://book.cakephp.org/2.0/en/installation.html) page in the Cookbook.

There are several specific files that must be setup properly, conveniently located in the Config directory in the CakePHP application. The **database.php** file contains the configuration for the database that CakePHP will use. The **core.php** file contains core configurations for the CakePHP application, including the security salts and seeds used later in password hashing. This file also contains the ability to set the debug level of the application; the various levels are noted in the code comments. Here also the **routes.php** file contains routing information for accessing controller methods, etc. This is where most of the URL routing for the HTD application is configured.

Our CakePHP application **requires** PHP 5.5 or higher, due to an error in the implementation of Blowfish/bcrypt in older versions of PHP. Hashes created with lower versions of PHP will result in errors when trying to log a user in.

## 5.3 Useful References

The following links are a few of the more important articles available in the CakePHP Cookbook. **This is not a substitute for reading the Cookbook on your own!**

**CakePHP Cookbook:** <http://book.cakephp.org/2.0/en/index.html>

**Installation Instructions:** <http://book.cakephp.org/2.0/en/installation.html>

**Conventions:** <http://book.cakephp.org/2.0/en/getting-started/cakephp-conventions.html>

**CakePHP Structure:** <http://book.cakephp.org/2.0/en/getting-started/cakephp-structure.html>

**CakePHP Folder Structure:** <http://book.cakephp.org/2.0/en/getting-started/cakephp-folder-structure.html>

**CakePHP Models:** <http://book.cakephp.org/2.0/en/models.html>

**CakePHP Controllers:** <http://book.cakephp.org/2.0/en/controllers.html>

**CakePHP Views:** <http://book.cakephp.org/2.0/en/views.html>

# Models

## 6.1 Overview

CakePHP [models](http://book.cakephp.org/2.0/en/models.html) are the primary

## 6.2 Datum

## 6.3 DataInProgress

## 6.4 Users

# Controllers

## 7.1 Overview

## 7.2 AppController

### 7.2.1 Interactions

### 7.2.2 Views Associated

### 7.2.3 Functions

## 7.3 PagesController

### 7.3.1 Interactions

### 7.3.2 Views Associated

### 7.3.3 Functions

## 7.4 AdminPanelController

### 7.4.1 Interactions

### 7.4.2 Views Associated

### 7.4.3 Functions

## 7.5 UploadsController

### 7.5.1 Interactions

### 7.5.2 Views Associated

### 7.5.3 Functions

## 7.6 DownloadsController

### 7.6.1 Interactions

### 7.6.2 Views Associated

### 7.6.3 Functions

## 7.7 CaseEditsController

### 7.7.1 Interactions

### 7.7.2 Views Associated

### 7.7.3 Functions

## 7.8 CaseReviewsController

### 7.8.1 Interactions

### 7.8.2 Views Associated

### 7.8.3 Functions

## 7.9 UsersController

### 7.9.1 Interactions

### 7.9.2 Views Associated

### 7.9.3 Functions

## 7.10 SearchController

### 7.10.1 Interactions

### 7.10.2 Views Associated

### 7.10.3 Functions

## 7.11 AnalyzeController

### 7.11.1 Interactions

### 7.11.2 Views Associated

### 7.11.3 Functions

# Glossary of Terms

**Administrator** – privileged user capable of performing major changes to database.

**Application** – Group of programs designed to supply an end-user with expected functionality.

**CakePHP** – A free, open-source, rapid development framework for PHP.

**Control** **Panel** – interface specifically designed to allow administrators to easily perform their tasks.

**Database** – A structured set of data held in a computer, accessible in various ways.

**Deliverable** – A product, not necessarily finished, related to the project given to the client.

**End-User** – A person or persons who will be using the web application for the specified purpose of our project.

**Foreign Key** – A field in one table that uniquely identifies a row of another table.

**GitHub** – A Web service for software version control.

**Host** – A website on a server accessible over the Internet.

**Milestone** – A point at which project progress can be assessed.

**PHP** – A general-purpose scripting language that is especially suited to server-side web development.

**Primary Key** – Uniquely identifies each record in the table.

**Prototype** – simulates only a few aspects of, and may be completely different from, the final product.

**TCU** – Texas Christian University

**UML** – Unified Modeling Language; a modeling language designed to provide a standard way to visualize the design of a system.

**Walk-through** – Points during the project where the team describes significant project components with clients and individuals within the team.

**Web Application** – Application that is accessed by visiting a specific URL.