**Описание архитектуры**

# ****Введение****

Разрабатываемая система представляет собой микро CMS, которая в дальнейшем будет применяться в качестве инвестиционной игры, мониторинга ферм и информационного ресурса-магазина. Система состоит из сайта и административной панели. Архитектура системы реализована в виде шаблона MVC.

# Реализация MVC

MVC системы реализован за счет трех групп классов:

* models(модели)
* controllers(контроллеры)
* views(представления)

## Модель

Модель отвечает за бизнес-логику приложения, проверяет пользовательский ввод, взаимодействуют с базой данных и т.д.

## Контроллер

Контроллер представляет собой тонкую прослойку между моделью и представлением. Он реагирует на действия пользователя: вызывает ту или иную функцию модели, передает данные из модели в отображение.

## Представление

Представление занимается генерацией html кода, оборачивая полученные из модели данные в нужные теги.

# Роутинг

Роутинг, осуществляемый классом Route отвечает за генерацию страниц в зависимости от текста запроса. Поддерживает три и менее уровня вложенности:

**Три уровня.** *Myfarm/admin/menu/?action=add*

Здесь *Myfarm* – имя сайта, *admin* - название контроллера, *menu* – название действия контроллера, *action=add* – параметр запроса, вызываемое действие модели.

Для остальных уровней принцип сохраняется, лишь для каждого нового меньше уровня отпадает самая правая часть. В итоге на нулевом уровне имеем просто адрес сайта: *myfarm.* В качестве контроллера выступает контроллер по умолчанию, а в качестве действия контроллера – действие по умолчанию.

# Организация файловой системы

Класс роутера, а также базовые классы Model, Controller и View, от которых будут унаследованы все остальные, располагаются по адресу application/core.

Классы моделей, контроллеров и представлений для каждой новой функции сайта располагаются по адресам application/models, application/controllers, application/views соответственно.

О представлениях необходимо сказать еще два слова. Иногда на один контроллер может приходиться несколько представлений, для каждого из действий контроллера и модели. Например, существуют классы контроллера и модели Admin, реализующие административную панель. При этом у административной панели много функций и для представления каждой функции генерируется своя страница – свое представление. В связи с этим в папки application/views для административной панели выделена своя папка admin, внутри которой лежат файлы, представляющие каждую функцию панели управления.

Остальные классы, выполняющие большой спектр задач располагаются по адресу application/classes.

В папке views есть также папка текущего шаблона template. Она организована определенным образом:

* папка img – изображения для шаблона
* папка styles – стили
* название шаблона.php
* PHP файлы модулей

Порядок обработки запроса пользователя

Для каждой новой функции сайта генерируется новая триада модель-контроллер-представление. Происходит это следующим образом:

* Пользователь переходит по определенному адресу, например myfarm/signup
* Файл «.htaccess» перенаправляет его на главную страницу – myfarm
* Роутер, подключенный на главной странице, обрабатывает введенный пользователем адрес и генерирует классы модели и контроллера в зависимости от его содержания
* Сгенерированный контроллер вызывает функцию модели, которая предоставляет пользователю запрошенную информацию
* Контроллер передает информацию из модели в представление
* Представление генерирует html страницу, добавляя туда данные, полученные из контроллера

Шаблон

Чтобы не писать повторяющейся код для каждой новой функции сайта, через MVC мы генерируем только «новое» содержимое страницы. Для остальной, статичной информации, есть шаблон. Помимо «скелета» страницы, в шаблоне есть модули(авторизация для неавторизованных, меню, статистика и все прочее, что мы захотим добавить). Мы предполагаем, что для сайта будет применяться несколько шаблонов, которые можно будет легко менять.

В шаблоне располагаются следующие модули:

* Меню
* Статистика
* Форма авторизации для неавторизованных
* Информация об аккаунте для авторизованных