ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

"ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ"

Факультет ФИСП

Кафедра ПИ

Лабораторная работа №5

предмет: «Анализ требований программного обеспечения»

по теме: «Специфицирование требований»

Выполнил:

ст. гр. ПИ-19а

Саевский О.В.

Проверил:

Григорьев А.В.

Ищенко А.П.

ДОНЕЦК – 2021

Глоссарий предметной области:

***База данных*** - Совокупность данных, хранимых в соответствии со схемой данных, манипулирование которыми выполняют в соответствии с правилами средств моделирования данных.

***ORM*** - технология программирования, которая связывает базы данных с концепциями объектно-ориентированных языков программирования, создавая «виртуальную объектную базу данных»

***Пользователь*** – Разработчик или администратор сайта.

***Интерфейс*** – Внешнее отображение системы, использующее HTML разметку.

***PHP*** - Скриптовый язык общего назначения, интенсивно применяемый для разработки веб-приложений. В настоящее время поддерживается подавляющим большинством хостинг-провайдеров и является одним из лидеров среди языков, применяющихся для создания динамических веб-сайтов.

***Адаптивная верстка*** - дизайн веб-страниц, обеспечивающий правильное отображение сайта на различных устройствах, подключённых к интернету, и динамически подстраивающийся под заданные размеры окна браузера.

***OAuth 2.0*** - протокол авторизации, позволяющий выдать одному сервису (приложению) права на доступ к ресурсам пользователя на другом сервисе. Протокол избавляет от необходимости доверять приложению логин и пароль, а также позволяет выдавать ограниченный набор прав, а не все сразу.

***MySQL*** - Свободная реляционная система управления базами данных.

***API*** - описание способов (набор [классов](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B0%D1%81%D1%81_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)), [процедур](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%86%D0%B5%D0%B4%D1%83%D1%80%D0%B0_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)), [функций](https://ru.wikipedia.org/wiki/%D0%A4%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D1%8F_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)), [структур](https://ru.wikipedia.org/wiki/%D0%A1%D1%82%D1%80%D1%83%D0%BA%D1%82%D1%83%D1%80%D0%B0_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)) или [констант](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BD%D1%81%D1%82%D0%B0%D0%BD%D1%82%D0%B0_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5))), которыми одна компьютерная программа может взаимодействовать с другой программой.

***Composer*** - Это пакетный менеджер уровня приложений для языка программирования PHP, который предоставляет средства по управлению зависимостями в PHP-приложении.

***Bootstrap*** - Свободный набор инструментов для создания сайтов и веб-приложений. Включает в себя HTML- и CSS-шаблоны оформления для типографики, веб-форм, кнопок, меток, блоков навигации и прочих компонентов веб-интерфейса, включая JavaScript-расширения.

***Ajax*** - Подход к построению интерактивных пользовательских интерфейсов веб-приложений, заключающийся в «фоновом» обмене данными браузера с веб-сервером. В результате при обновлении данных веб-страница не перезагружается полностью, и веб-приложения становятся быстрее и

удобнее.

***Docker*** - Программное обеспечение для автоматизации развёртывания и управления приложениями в средах с поддержкой контейнеризации, контейнеризатор приложений.

Акторы

Для бизнес требований:

- Система;

- Разработчик;

- Локальный сервер;

- Сайт;

- PHP;

- Docker;

Требований пользователя:

- Сервер

- Разработчик

- Composer

- Bootstrap

- JavaScript

- Ajax

Функциональных требований поведения:

- Система

- API

- Composer

- ORM

- MySQL

|  |  |
| --- | --- |
| ***Таблица для бизнес требований*** |  |
| ***Актор*** | ***Действие*** |
| ***Разработчик*** | ***Устанавливает систему*** |
| ***Система*** | ***Устанавливает необходимые зависимости*** |
| ***Разработчик*** | ***Запускает систему*** |
| ***Система*** | ***Запускает локальный сервер и развертывает сайт*** |
| ***Сайт*** | ***Подгружает Html код и запрашивает интерпретацию PHP кода*** |
| ***Локальный сервер*** | ***Запускается и читает Html код*** |
| ***PHP*** | ***Интерпретирует свой код в HTML разметку*** |
| ***Разработчик*** | ***Упаковывает проект в Docker*** |
| ***Docker*** | ***Собирает все таблицы MySQL, все зависимости Composer и код разработчика в контейнер*** |
| ***Разработчик*** | ***Публикует Docker контейнер на сайте*** |
| ***Docker*** | ***Развертывается на сайте, устанавливает все зависимости*** |
| ***Сайт*** | ***Развертывает приложение*** |

|  |  |
| --- | --- |
| ***Таблица для требований пользователя*** |  |
| ***Актор*** | ***Действие*** |
| ***Разработчик*** | ***Выбирает зависимости для своего сайта*** |
| ***Composer*** | ***Устанавливает зависимости*** |
| ***Разработчик*** | ***Использует Ajax код*** |
| ***Ajax*** | ***Использует свой функционал для динамической работы с сайтом*** |
| ***Разработчик*** | ***Пишет предустановленные CSS стили в Bootstrap*** |
| ***Bootstrap*** | ***Берет готовые шаблоны из сети*** |

|  |  |
| --- | --- |
| ***Таблица для функциональных требований*** |  |
| ***Актор*** | ***Действие*** |
| ***Разработчик*** | ***Использует API метод системы*** |
| ***Система*** | ***Получает запрос и дает ответ*** |
| ***Разработчик*** | ***Пишет запрос к ORM на добавление данных*** |
| ***ORM*** | ***Обращается к MySQL сервера с данными*** |
| ***MySQL*** | ***Добавляет данные в таблицу или выдает ответ об ошибке*** |
| ***Разработчик*** | ***Пишет обработку ошибки используя API системы*** |
| ***Система*** | ***Получает запрос и дает ответ*** |