## 1.5 Выбор инструментальных средств

## 1.5.1 Инструментальные средства для организации работы

## 1.5.1.1 Trello

Для организации и планирования списка работ, которые необходимо выполнить для завершения написания выпускной квалификационной работы, был выбран Trello.

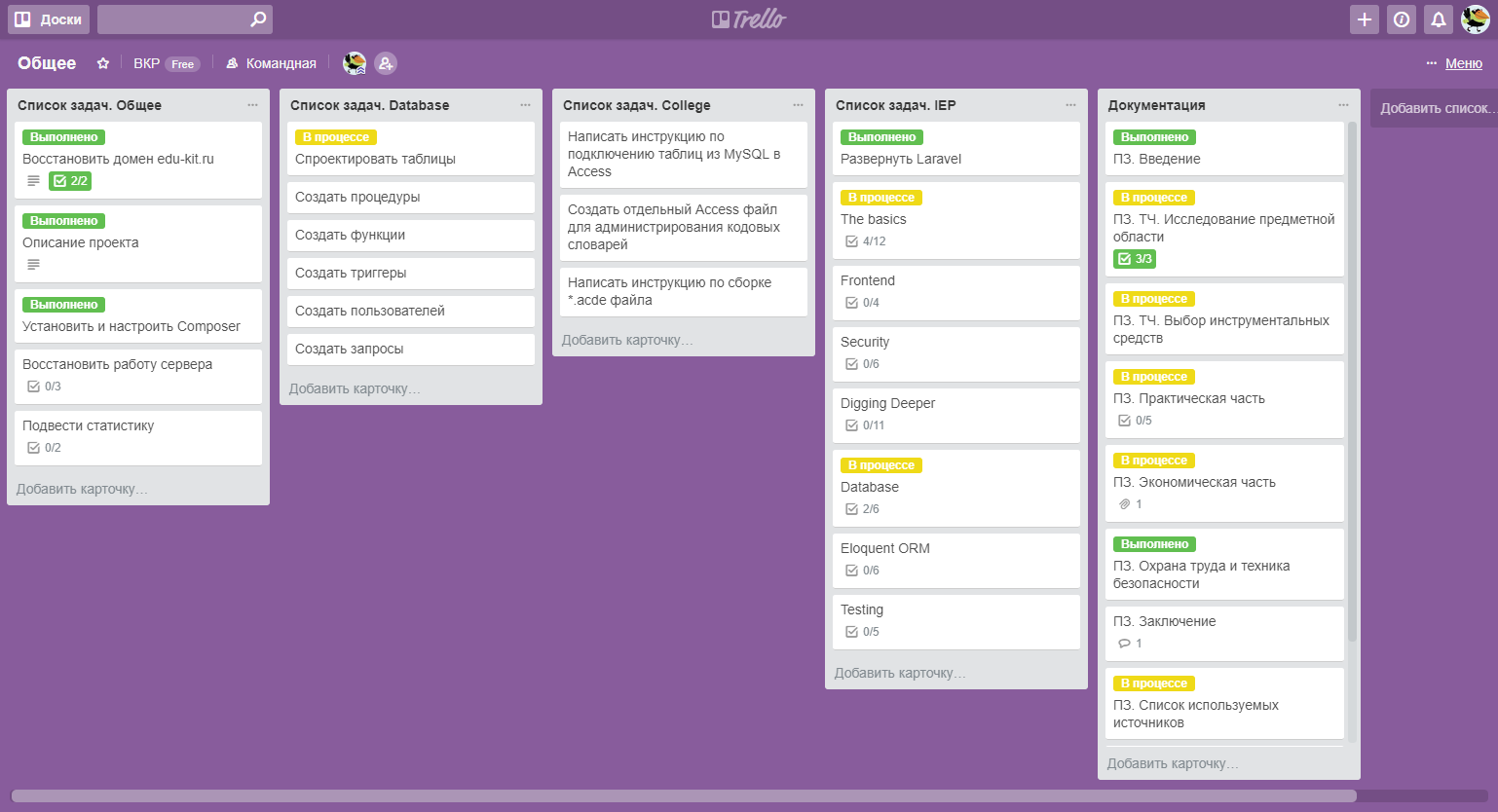
Главная особенность Trello:

* доски, списки и карточки,
* возможность командой работы,
* наличие веб-версия и мобильного приложения,
* наличие улучшений для досок, которые позволяют расширять содержимое карточек.

Доски в себе содержат списки, в которых содержатся карточки. При этом карточка может в себе содержать:

* текст,
* метка,
* чек-лист,
* дату выполнения,
* вложение,
* комментарии к ней.

Рабочее пространство Trello отображено на (Рисунок ).



Однако имеются следующие аналоги:

1. Wunderlist;
2. Planner;
3. Planiro.

Ниже приведено сравнение (Таблица ) по основным критериям.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Критерий** | **Trello** | **Wunderlist** | **Planner** | **Planiro** |
| Бесплатное | + | +/– | – | +/– |
| Кроссплатформенность | + | + | + | + |
| **Возможности** | | | | |
| Доски | + | – | + | – |
| Списки | + | + | + | + |
| Карточки | + | – | + | + |
| Вложения | + | + | + | + |
| Сроки выполнения | + | + | + | + |
| Улучшения | + | – | – | – |

По результатам сравнения Trello является наиболее подходящим, так как обладает простым, но гибким функционалом, обеспечивая наглядность.

## 1.5.1.2 Система контроля версий Git

В качестве системы контроля версий была выбрана СКР Git. В связи с тем, что текущая версия портала также разрабатывалась с использованием данной СКР, то отказываться от неё бессмысленно, так как она содержит полную историю всех изменений, к примеру, разработанной базы данных.

Однако основными аналогами являются:

* Mercurial,
* Subversion.

Ниже приведено сравнение (Таблица ) по основным критериям.

|  |  |  |  |
| --- | --- | --- | --- |
| **Критерий** | **Git** | **Mercurial** | **Subversion** |
| Бесплатная | + | + | + |
| Кроссплатформенность | + | + | + |
| Простота использования | + | + | + |
| Полная история разработки | + | + | – |
| Распределённая система | + | + | – |

По результатам сравнения можно сделать заключение о том, что Git наиболее подходит для поставленных целей. Как отмечалось ранее, текущая версия портала была разработана с использованием данной системы контроля версий.

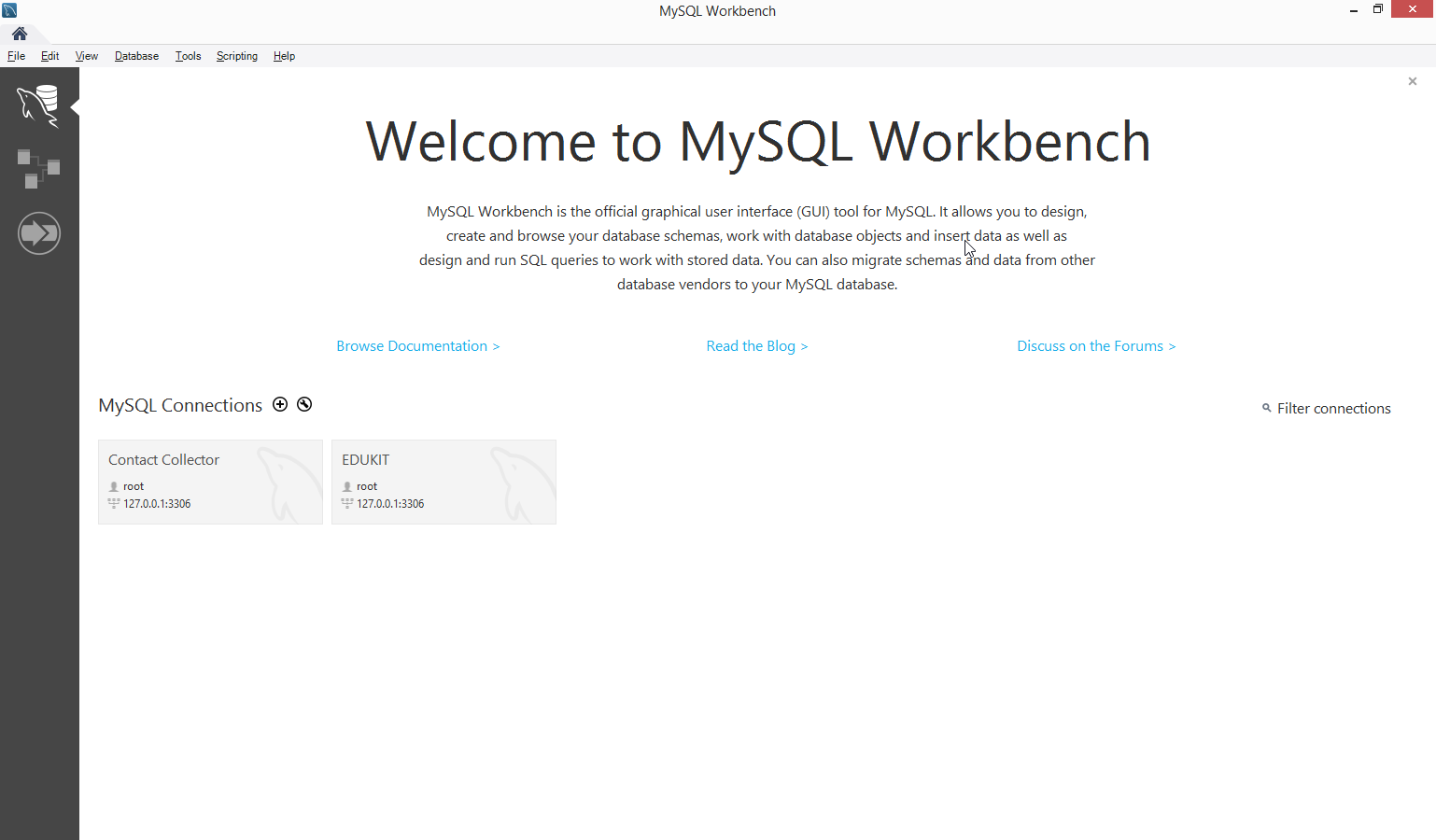
## 1.5.2 Инструментальные средства для проектирования

## 1.5.2.1 Проектирование базы данных

В качестве средства для проектирования базы данных был выбран MySQL Workbench. Данное средство было выбрано в связи с тем, что оно также использовалось при разработке портала, и обладает следующими ключевыми возможностями:

* работа с множеством подключений,
* редактор SQL-запросов,
* средства для администрирования,
* миграция схем из MSSQL, MS Access, SQLite, PostgresSQL и других РСУБД.

Главное окно MySQL Workbench изображено на (Рисунок ).



Однако основным аналогичными средствами являются:

* CA ERwin Data Modeler,
* ER Assistant.

Ниже приведено сравнение инструментальных средств (Таблица ) по основным критериям.

|  |  |  |  |
| --- | --- | --- | --- |
| **Критерий** | **MySQL Workbench** | **CA ERwin Data Modeler** | **ER Assistant** |
| Бесплатная | + | +/– | + |
| Открытый исходный код | + | – | – |
| Кроссплатформенность | + | – | – |
| Поддерживается | + | + | – |
| **Инструменты** | | | |
| Миграция из РСУБД | + | – | – |
| Миграция в РСУБД | – | + | – |
| Генерация SQL | + | + | – |
| Редактор SQL кода | + | – | – |
| Средство создания резервных копий | + | – | – |
| **Работа с объектами базы данных** | | | |
| Диаграммы | + | + | + |
| Таблицы | + | + | + |
| Представления | + | + | – |
| Хранимые процедуры | + | + | – |
| Функции | + | + | – |
| Триггеры | + | + | – |
| Пользователи | + | + | – |
| Индексы | + | + | – |

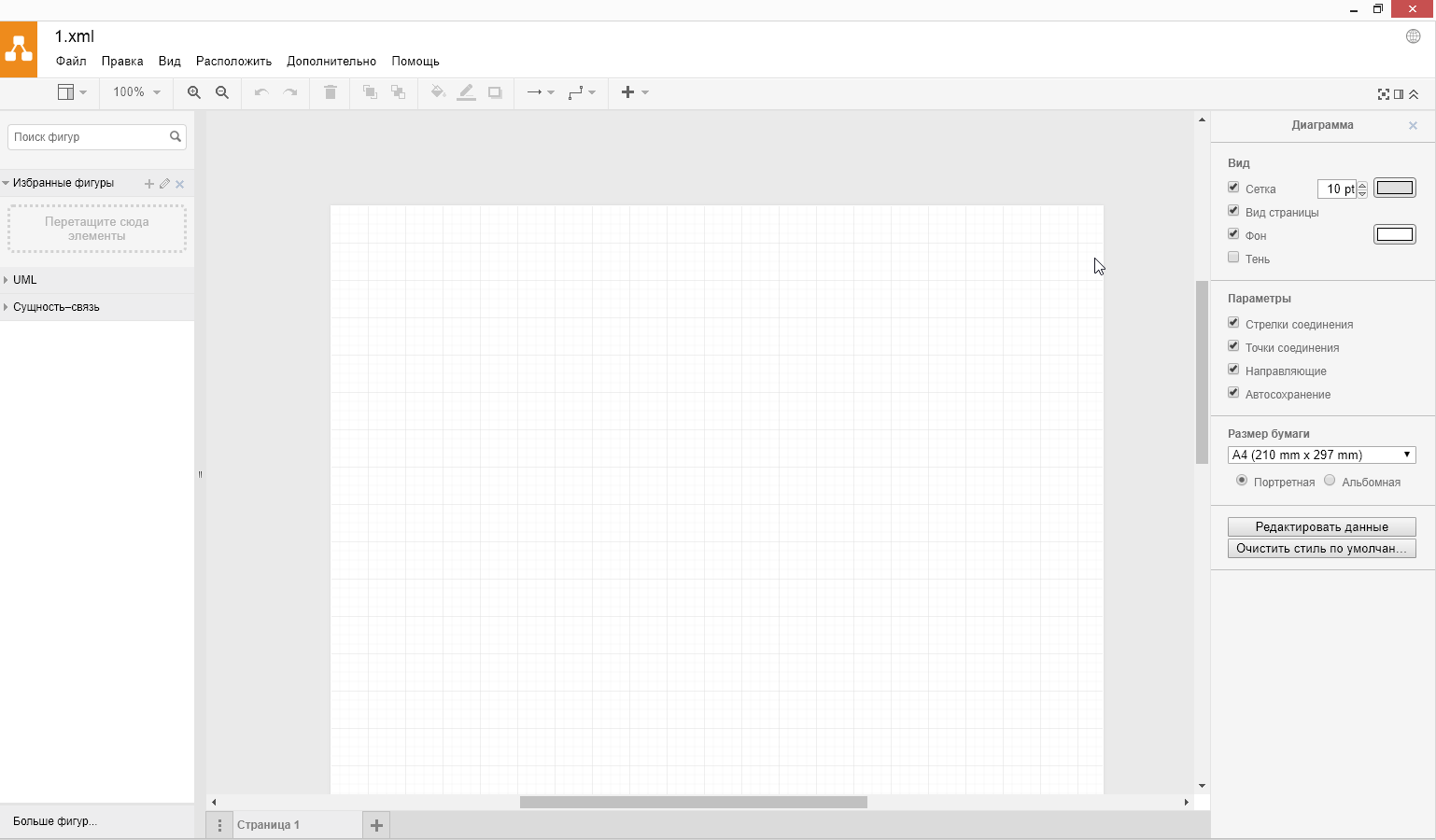
По результатам сравнения можно сделать заключение о том, что MySQL Workbench больше всего подходит для проектирования базы данных. Так как оно совмещает в себе инструменты проектирования и разработки базы данных, то это ускоряет разработку базы данных в целом, потому что проектирование идёт уже в среде исполнения и позволяет своевременно вносить изменения в схему базы данных, исключая процесс миграции базы данных из среды проектирования в среду исполнения.

## 1.5.2.2 Моделирование UML диаграмм

В качестве средства для моделирования UML диаграмм было выбрано Draw.io. Оно является бесплатным и обладает следующими возможностями:

* кроссплатформенное,
* создание различных диаграмм (например, сущность-связь),
* экспорт в форматы (PDF, PNG, SVG, VSDX, HTML, XML),
* может быть интегрировано (с JIRA, Trello и G-Suite).

Главное окно Draw.io изображено на (Рисунок ).



Однако основными аналогичными программными продуктами являются:

* Microsoft Visio,
* StarUML,
* Umbrello.

Ниже приведено сравнение (Таблица ) по основным критериям.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Критерий** | **Draw.io** | **MS Visio** | **StarUML** | **Umbrello** |
| Бесплатная | + | – | +/– | + |
| Открытый исходный код | – | – | – | + |
| Кроссплатформенность | + | – | – | – |
| Поддерживается | + | + | + | + |
| **Возможности** | | | | |
| Экспорт в различные форматы | + | + | + | + |
| Генерация кода | – | – | + | + |
| Реверсивный инжиниринг | – | – | + | + |

В результате сравнения, Draw.io уступает по своим возможностям StarUML и Umbrello. Однако выбор Draw.io обусловлен тем, что нет необходимости в генерации исходного кода из создаваемых UML диаграмм и реверсивный инжиниринг по существующему исходному коду. Draw.io бесплатный и простой в использовании.

## 1.5.3 Инструментальные средства для модернизации портала

## 1.5.3.1 Языки вёрстки и стилей

В качестве основного языка вёрстки страниц был выбран HTML5, так как является наиболее популярным и часто используемым при веб-разработке. Помимо всего он обладает следующими возможностями:

* работа с мультимедиа и графикой,
* работа с локальным хранилищем браузера,
* поддерживает XMLHttpRequest для AJAX-запросов,
* исправление ошибок в DOM.

Однако основными аналогичными языком является XHTML. Особенностями которого являются:

* строгое оформление кода,
* позволяет использовать XML-технологии «из коробки» (например, XSLT).

Оба языка разрабатываются консорциумом всемирной паутины W3C. Это означает, что оба языка поддерживаются. Однако есть причины, при которых в использование XHTML нет необходимости:

1. при разработке существующей версии портала использовался HTML5, а значит, изменение структуры свёрстанных страниц может занять продолжительное время, в связи с особенностями языка.
2. большинство веб-сайтов всемирной паутины написано с использованием HTML.
3. XHTML проще в обработке, чем HTML. И он был задуман как язык разметки для мобильных телефонов, которые не имели достаточной производительности. Однако сейчас любой мобильный телефон (смартфон) может воспроизводить HTML5 безо всякого снижения производительности.

В результате можно сделать вывод, что HTML5 подходит для вёрстки страниц.

В качестве основного языка для визуального оформления веб-страниц был выбран CSS3 (Каскадные таблицы стилей), так как основных аналогичных языков нет.

## 1.5.3.2 Языки программирования

## 1.5.3.2.1 Клиентская сторона

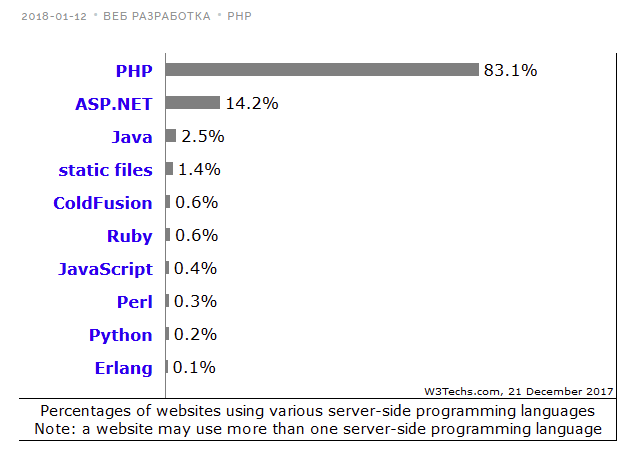
В качестве основного языка был выбран JavaScript, однако он напрямую не используется при разработке.

## 1.5.3.2.2 Серверная сторона

В качестве основного языка был выбран PHP. Данный язык был выбран вследствие того, что текущая версия портала так же разрабатывалась на данном языке. Что уже упрощает процесс модернизации, так как нет необходимо в изучении нового языка программирования и сопутствующих технологий для веб-разработки.

При этом язык PHP на данный момент является наиболее популярным при разработке серверной части.

Как аргумент, подтверждающий частоту использования ниже на (Рисунок ) отображена статистика использования языков программирования на стороне сервера.



Однако, аналогичными языками могут выступать:

* Java,
* C++.

Ниже приведено сравнение (Таблица ) по основным критериям.

|  |  |  |  |
| --- | --- | --- | --- |
| **Критерий** | **PHP** | **Java** | **C++** |
| Порог вхождения | Низкий | Средний | Высокий |
| Изучен | + | – | + |
| ООП | + | + | + |
| Не нужен CGI | + | – | – |
| Создан для веб-разработки | + | – | – |

В результате сравнения можно сделать вывод том, что PHP лучше подходит для модернизации, так как он уже был изучен и можно будет уделить время изучению дополнительных инструментов и PHP фреймворков.

## 1.5.3.3 Библиотеки

В качестве библиотеки на стороне клиента была выбрана jQuery, так как она необходима при использовании CSS фреймворков.

## 1.5.3.4 Фреймворки

## 1.5.3.4.1 Клиентская сторона

В качестве фреймворка для каскадных таблиц стилей был выбран Semantic UI. Выбранный CSS фреймворк обладает следующими особенностями:

* по умолчанию интегрирован с фреймворком Laravel,
* имеет простую и гибкую систему сеток,
* имеет готовый набор стилей для элементов,
* имеет готовый набор компонентов (иконки, модальные окна, сворачивания и т.д.),
* вёрстка является адаптивной.

Однако основными аналогами являются:

* Bootstrap 4,
* Material UI.

Ниже приведено сравнение (Таблица ) по основным критериям.

|  |  |  |  |
| --- | --- | --- | --- |
| **Критерий** | **Semantic UI** | **Bootstrap 4** | **Material UI** |
| Изучено | + | – | – |
| Простая и понятная вёрстка | + | – | – |
| Адаптивная вёрстка | + | + | + |
| Интеграция с PHP фреймворками | + | + | + |
| Готовый набор компонентов | + | + | + |
| Необходимо использовать с JS-фреймворком | – | – | + |

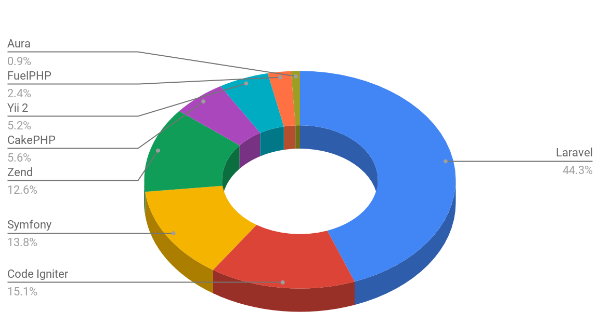
В результате сравнения можно сделать заключение о том, что лучше всего подходит Semantic UI. Помимо этого он использовался в разработке текущей версии портала.

## 1.5.3.4.2 Серверная сторона

В качестве фреймворка для разработки серверной стороны был выбран Laravel, так как обладает следующими особенностями:

* использует объектно-ориентированный подход,
* использует современные инструменты, стандарты и технологии в веб-разработке (такие как Composer, стандарты PSR, PHP7 и т.д.),
* гибкость,
* масштабируемость,
* скорость разработки,
* консоль Artisan.

Фреймворк является одним из самым часто используемым в разработке, о чём говорят статистические данные (Рисунок ).



Однако основными аналогичными фреймворками являются:

* Yii 2.0;
* Phalcon.

Ниже приведено сравнение (Таблица ) по основным критериям.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Laravel** | **Yii 2.0** | **Phalcon** |
| Открытый исходный код | + | + | + |
| Поддерживается | + | + | + |
| **Возможности** | | | |
| MVC | + | + | + |
| Поддержка Composer | + | + | - |
| ORM | + | + | - |
| Встроенный шаблонизатор | + | - | + |
| Маршрутизация | + | + | + |
| Аутентификация | + | + | - |

В результате, можно сделать вывод о том, что Laravel является лучшим выбором. Так как помимо перечисленных особенностей он ещё имеет встроенный многофункциональным шаблонизатором Blade. В целом, выбирая, Laravel скорость разработки повышается, так как большинство базовых вещей уже реализовано (например, маршрутизация и аутентификация). И, конечно же, наличие консоль Artisan, которая позволяет создавать контроллеры, модели, работать с маршрутизацией и другое.

## 1.5.3.5 Среда разработки

В качестве среды разработки была выбрана Visual Studio Code. Основная причина, по которой она была выбрана – это её «легковесность», так как она не занимает слишком много места на компьютере и не требовательная к ресурсам.

Основными её достоинствами являются:

* удобный текстовый редактор с подсветкой синтаксиса,
* удобная работа с помощью дополнительного терминала,
* возможность подключить отладчик или транслятор для полноценной разработки,
* возможность подключение системы контроля версий
* установка дополнительных плагинов, расширяющие возможности.

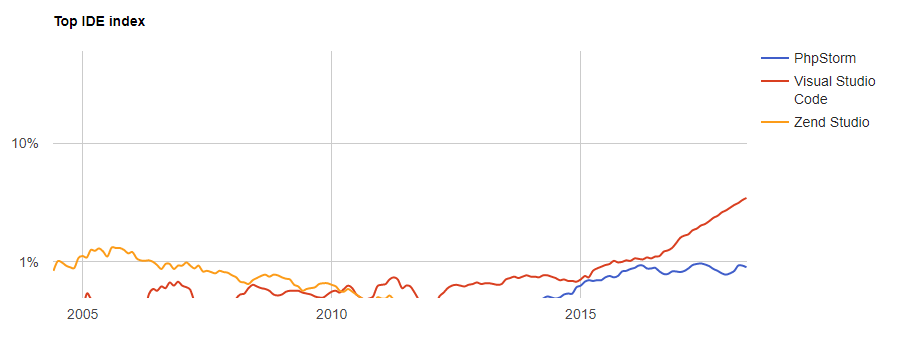
Однако основными аналогами являются:

* PHPStorm,
* Zend Studio.

Ниже приведено сравнение (Таблица ) по основным критериям.

|  |  |  |  |
| --- | --- | --- | --- |
| **Критерий** | **Visual Studio Code** | **PHPStorm** | **Zend Studio** |
| Бесплатное | + | – | – |
| Кроссплатформенность | + | + | + |
| Поддерживается | + | + | + |
| Не требовательность к ресурсам | + | – | – |
| **Возможности** | | | |
| IntelliSense | + | – | – |
| Отладка | + | + | + |
| Плагины | + | + | + |
| Поддержка работы с терминалом | + | + | – |

В результате сравнения можно сделать заключение о том, что Visual Studio Code является подходящим средством для разработки. Так же на основании статистики можно утверждать, что Visual Studio Code популярнее, нежели сравниваемые среды разработки.



## 1.5.3.6 Средство автоматического документирования кода

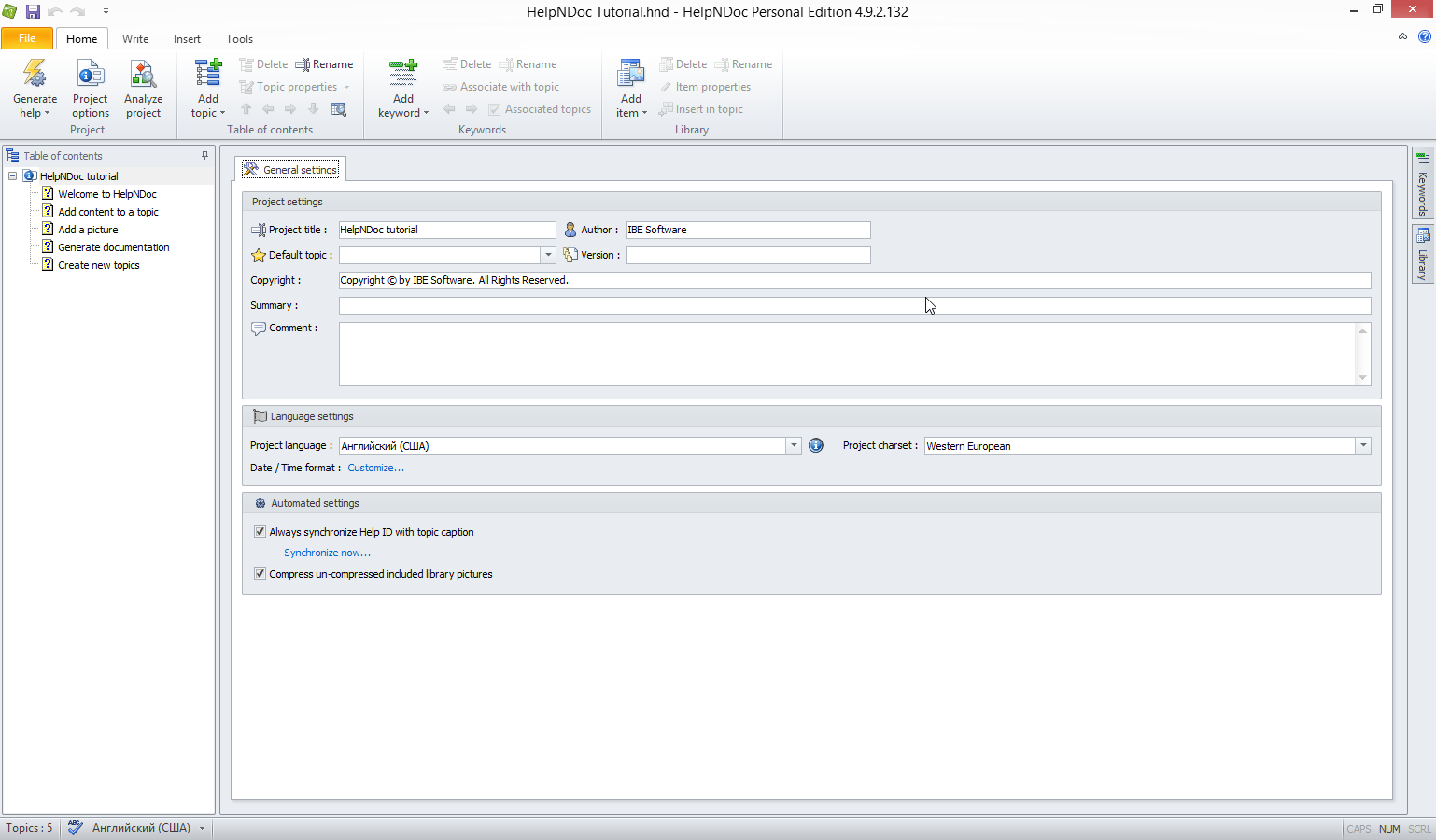
В качестве

## 1.5.3.7 Средство создания пользовательской документации

В качестве средства для создания пользовательской документации был выбран HelpNDoc. Основными особенностями являются:

* интерфейс построен в соответствии с Microsoft Style Guide UI
* удобное создание иерархической структуры,
* позволяет наглядно представить структуру в виде диаграммы,
* позволяет вставлять HTML-код в содержимое страниц,
* экспорт в форматы (CHM, HTML, Word, PDF, Qt Help, eBooks).

Ниже на (Рисунок ) приведён главный экран приложения.



Однако основными аналогами являются:

* Dr. Explain,
* HTML Help Workshop.

Ниже приведено сравнение (Таблица ) по основным критериям.

|  |  |  |  |
| --- | --- | --- | --- |
| **Критерий** | **HelpNDoc** | **Dr. Explain** | **HTML Help Workshop** |
| Бесплатное | + | – | + |
| Кроссплатформенность | – | – | – |
| Поддерживается | + | + | + |
| **Возможности** | | | |
| Создание иерархической структуры | + | + | + |
| Экспорт в PDF | + | + | – |
| Экспорт в Word | + | – | – |
| Экспорт в HTML | + | + | +/– |
| Экспорт в CHM | + | + | + |

В результате сравнения, можно сделать заключение о том, что HelpNDoc лучше подходит для создания пользовательской документации, так как имеется обширная поддержка форматов, в которые можно преобразовать созданную документацию.