

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ**

**ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
 «ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**(ДГТУ)**

Факультет \_\_\_\_Информатика и вычислительная техника\_\_\_\_\_\_

(наименование факультета)

Кафедра \_\_\_\_\_ Информационные технологии\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(наименование кафедры)

Зав. кафедрой

\_«Информационные технологии»\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Б.В. Соболь

(подпись) (И.О.Ф.)

« \_\_\_\_\_\_ » \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2018 г.

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

к выпускной квалификационной работе бакалавра на тему:

WEB-ПОРТАЛ "TRANSLAT'HOUSE" СООБЩЕСТВА ПЕРЕВОДЧИКОВ (ФУНКЦИЯ: ОБЕСПЕЧЕНИЕ КОНТЕНТА)

Автор выпускной квалификационной работы: \_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_А.А. Гробер\_\_\_\_\_\_\_

(подпись, дата) (И.О.Ф)

Обозначение ВКР 09.03.02.060000.000 БР Группа \_ВИС42\_\_

Направление 090302 «Информационные системы и технологии»

(код) (наименование)

Руководитель работы: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_доцент, Д.Н. Климова

(подпись, дата) (должность, И.О.Ф)

Консультанты по разделам:

Безопасность и экологичность \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_доцент, О.В. Денисов

(наименование раздела) (подпись, дата) (должность, И.О.Ф)

Нормоконтроль \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ст. преп. Т.М. Блиновская

(подпись, дата) (должность, И.О.Ф)

Ростов-на-Дону

2018



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ**

**ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
 «ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**(ДГТУ)**

Факультет \_\_\_\_Информатика и вычислительная техника\_\_\_\_\_\_

(наименование факультета)

Кафедра \_\_\_\_\_ Информационные технологии\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(наименование кафедры)

Зав. кафедрой

\_«Информационные технологии»\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Б.В. Соболь

(подпись) (И.О.Ф.)

« \_\_\_\_\_\_ » \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2018 г.

**ЗАДАНИЕ**

к выпускной квалификационной работе бакалавра

Обучающийся Гробер Анна Александровна Группа ВИС42

Обозначение ВКР 09.03.02.060000.000 БР

Тема: Web-портал "Translat'house" сообщества переводчиков (функция: обеспечение контента)

Утверждено приказом по ДГТУ № 4570-ЛС-О\_ от «\_26 \_» \_декабря 2017 г.

Срок представления ВКР к защите: « 28 » июня 2018 г.

Исходные данные для выпускной квалификационной работы: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| *Бьюли Алан. Изучаем SQL,2000;* |
| *Якоб Нильсен, Хоа Лоранжер. Web-дизайн. Удобство использования Web-сайтов, 2009;* |
| *Владимир Дронов. Laravel. Быстрая разработка современных динамических Web-сайтов* |
| *на PHP, MySQL, HTML и CSS, 2018.* |
|  |
|  |
|  |

Содержание пояснительной записки:

ВВЕДЕНИЕ:

|  |
| --- |
| *Целью проектирования является web портал сообщества переводчиков «Translators'* |
| *House», который будут являться фриланс-биржей, специализирующейся на переводах* |
| *текстовой, видео- и другой информации. Также планируется реализация возможности* |
| *общения для пользователей.* |

1 Наименование и содержание разделов

*ВВЕДЕНИЕ*

|  |
| --- |
| *1.Постановка задачи* |
| *2.Проектирование web-портала* |
| *3.Анализ и выбор программно-инструментальных средств* |
| *4.Проектирование базы данных* |
| *5.Реализация клиентской части* |
| *6.Реализация серверной части* |
| *7.Безопасность и экологичность проекта* |
| *ЗАКЛЮЧЕНИЕ* |
| *БИБЛИОГРАФИЧЕСКИЙ СПИСОК* |

ЗАКЛЮЧЕНИЕ

*В результате данной дипломной работы реализован функционал для обеспечения контента для web-*

|  |
| --- |
| *портала сообщества переводчиков «Translator’s house».* |
|  |
|  |

Перечень графического материала**:**

|  |  |
| --- | --- |
|  | Презентация в Power Point на 15 листах |

Руководитель работы: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_доцент, Д.Н.Климова

(подпись, дата) (должность, И.О.Ф)

Консультанты по разделам:

Безопасность и экологичность \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_доцент, О.В. Денисов

(подпись, дата) (должность, И.О.Ф)

Задание принял к исполнению \_\_\_\_\_\_\_\_\_23.04.2018 \_А.А,Гробер\_\_\_\_\_\_\_\_\_\_

(подпись, дата) (И.О.Ф)

**АННОТАЦИЯ**

Дипломный проект на тему: «Web-портал "Traslat'house" сообщества переводчиков (функция: обеспечение контента)» содержит 72 страницы текста, 29 рисунков, 4 таблицы, 20 использованных источников.

Целью проектирования является web портал сообщества переводчиков «Translators' House», который будут являться фриланс-биржей, специализирующейся на переводах текстовой, видео- и другой информации. Также планируется реализация возможности общения для пользователей.

Объектом дипломной работы является взаимодействие переводчиков с заказчиками, а также с другими переводчиками.

Предметом является проектирование и разработка функционала для обеспечения контента web-портала.

The diploma project on the topic: "Traslat'house Web Portal" of the translator community (function: content maintenance) "contains 73 pages of text, 29 figures, 4 tables, 20 sources used.

The goal of the project is the web portal of the translator community "Translators' House", which will be a freelance exchange specializing in translations of text, video and other information. It is also planned to realize the possibility of communication for users.

The subject of the thesis is the interaction of translators with customers, as well as with other translators.

The subject is the design and development of functionality for providing content to the web portal.

**СОДЕРЖАНИЕ**

Изм.

Лист

№ докум.

Подпись

Дата

Лист

5

*09.03.03.060000.000 ПЗ*

Разраб.

*Гробер А.А*

Провер.

*Климова Д.Н.*

Реценз.

Н. Контр.

Утверд.

*Соболь Б.В..*

*Web-портал "Translat'house" сообщества переводчиков (функция: обеспечение контента)*

*Пояснительная записка*

Лит.

Листов

73

[ВВЕДЕНИЕ 7](#_Toc516305366)

[1. Постановка задачи 9](#_Toc516305367)

[1.1 Анализ программных продуктов 9](#_Toc516305368)

[1.2 Сравнительный анализ программных продуктов 17](#_Toc516305369)

[1.3 Назначение и этапы создания web-портала 19](#_Toc516305370)

[2. Проектирование web-портала 21](#_Toc516305371)

[2.1 Диаграмма вариантов использования (Use Case diagram) 23](#_Toc516305372)

[2.2 Диаграмма последовательности (Sequence diagram) 26](#_Toc516305373)

[2.3 Диаграмма коммуникации (Communication diagram) 27](#_Toc516305374)

[2.4 Диаграмма состояний (Statechart diagram) 28](#_Toc516305375)

[2.5 Диаграмма классов (Class diagram) 29](#_Toc516305376)

[2.6 Диаграмма компонентов (Component diagram) 31](#_Toc516305377)

[2.7 Даталогическая модель 32](#_Toc516305378)

[3. Анализ и выбор программно-инструментальных средств 36](#_Toc516305379)

[3.1 Инструментальные средства разработки 37](#_Toc516305380)

Изм.

Лист

№ докум.

Подпись

Дата

Лист

4

*09.03.03.550000.000 ПЗ*

Разраб.

*Сюда своюФИО*

Провер.

*ФИОруков*

Реценз.

Н. Контр.

Утверд.

*Соболь Б.В..*

*Сюда -своя тема ВКР*

*Пояснительная записка*

Лит.

Листов

70

Изм.

Лист

№ докум.

Подпись

Дата

Лист

6

*09.03.03.060000.000 ПЗ*

Разраб.

*Гробер А.А*

Провер.

*Климова Д.Н.*

Реценз.

Н. Контр.

Утверд.

*Соболь Б.В..*

*Web-портал "Translat'house" сообщества переводчиков (функция: обеспечение контента)*

*Пояснительная записка*

Лит.

Листов

73

[3.2 Обзор программных средств реализации интерфейса 41](#_Toc516305381)

[4. Проектирование базы данных 42](#_Toc516305382)

[5. Реализация клиентской части 49](#_Toc516305383)

[6. Реализация серверной части 56](#_Toc516305384)

[7. Безопасность и экологичность проекта 59](#_Toc516305385)

[7.1 Анализ опасных и вредных факторов разработчика web-портала 59](#_Toc516305386)

[7.2 Физические опасные и вредные производственные факторы 59](#_Toc516305387)

[7.3 Химические опасные и вредные производственные факторы 64](#_Toc516305388)

[7.4 Биологические опасные и вредные производственные факторы 65](#_Toc516305389)

[7.5 Расчет системы местного освещения 65](#_Toc516305390)

[7.6 Оценка экологической обстановки центра города Ростова-на-Дону. 66](#_Toc516305391)

[7.7 Чрезвычайные ситуации 68](#_Toc516305392)

[ЗАКЛЮЧЕНИЕ 70](#_Toc516305393)

[БИБЛИОГРАФИЧЕСКИЙ СПИСОК 71](#_Toc516305394)

# **ВВЕДЕНИЕ**

В современном мире большую популярность приобретают фриланс-биржи. Это web порталы, на которых любой человек может найти работников для выполнения определенной задачи или проекта. Работа чаще всего выполняется удаленно, что удобно для работодателя и исполнителя, так как удаленная форма работы экономит время и средства.

Фриланс-биржи бывают как многопрофильные, так и специализирующиеся на определенной тематике или конкретных видах работы.

Также в последнее время становится популярна тема онлайн переводов текстовой и видео информации. Происходит это из-за того, что производимого на различных языках контента становится с каждым годом все больше.

Целью данного проектированияявляется разработка web портала сообщества переводчиков «Translators' House», представляющего собой фриланс-биржу, специализирующуюся на переводах текстовой, видео- и другой информации. Также планируется реализация возможности общения для пользователей.

Основной задачей является функционал для обеспечения контента, а именно реализация:

-добавления, изменения, удаления объявлений;

-фильтрации для удобного поиска по объявлениям;

-панели для редактора веб портала;

-просмотра для пользователя объявлений, которые он выложил или взял на исполнение;

-форума для общения.

Актуальность дипломной работы в объединении переводчиков и заказчиков переводов, а именно:

-возможность выложить объявление о переводе в определенные сроки;

-с помощью фильтрации можно найти интересующие определенного пользователя объявления;

-исполнитель может переводить тексты в удобное время и в любом месте, у него нет графика работы;

-возможность заработать на любимом деле;

-общение с другими пользователями на интересующие их темы;

-для тех, кто недавно занимается переводами, это возможность набраться опыта.

Объектом дипломной работы является взаимодействие переводчиков с заказчиками, а также с другими переводчиками.

Предметом является проектирование и разработка функционала для обеспечения контента web-портала.

Отличием разрабатываемого портала от других подобных систем являются: ориентированность на молодежь, возможность общения, набор команды переводчиков для одной задачи, один человек может быть как работодателем, так и исполнителем, возможность добавления новой категории и нового языка, приятный современный дизайн, удобство пользования, адаптивная верстка, использование при разработке современных технологий.

Web-портал «Translator’s house» является совместной разработкой. Для удобства коллективной разработки используется сервис GitHub.

Портал является клиент-серверным приложением. Для реализации функционала серверной части используется язык PHP с использованием фреймворка Laravel. В качестве СУБД выбрана MySQL.

В данный момент система находится на локальном сервере. Используется сборка Xampp.

Для логики, используемой на клиентской части приложения, используются js, библиотеки JQuery, Ajax. Для реализации пользовательского интерфейса используются язык гипертекстовой разметки HTML, каскадные таблицы стилей CSS, фреймворк Bootstrap.

# **Постановка задачи**

В данной дипломной работе необходимо реализовать функционал для обеспечения контента для web-портала сообщества переводчиков «Translator’s house».

Должна быть возможность для каждого пользователя добавить новое объявление о переводе, изменить свое уже добавленное объявление, удалить его.

Для поиска объявлений необходим фильтр, в котором можно выбрать язык оригинала, язык перевода, тип перевода, цену, сложность, интервал по времени. Должна быть возможность посмотреть детали объявления.

Нужно сделать панель редактора для просмотра количества всех объявлений, тех, что в режиме набора пользователей, завершенных и завершенных по времени.

Для каждого пользователя должна быть возможность просмотра объявлений, которые он выложил или взял на исполнение.

Необходим форум для общения пользователей. Должа быть возможность добавления новых тем обсуждения и сообщений на уже существующие темы. Также каждый пользователь должен иметь возможность редактирования и удаления своих тем и сообщений.

Web-портал должен быть ориентирован на молодежь, удобен в использовании, иметь современный, приятный дизайн, адаптивную верстку.

## **Анализ программных продуктов**

Для анализа программных продуктов были выбраны многопрофильные фриланс биржи и биржи, специализирующиеся на теме переводов:

* фриланс сайт удаленной работы «Fl.ru»;
* биржа контента «eTXT»;
* биржа переводов «Tranzilla»;
* биржа переводов «01»;
* фриланс-платформа лингвистических услуг «2polyglot.com»;
* сервис по подбору переводчиков «Переводчик.Ме»;
* бюро переводчиков «ProZ».

1. Фриланс сайт удаленной работы «Fl.ru» (https://www.fl.ru)

На данный момент «Fl.ru» считается одной из самых популярных многопрофильных фриланс систем в России. Можно взять работу на исполнения из множества сфер (программирование, дизайн, реклама, переводы, обучение, фотография, написание текстов, менеджмент и другие).

Сайт рассчитан больше для профессионалов, чем для начинающих фрилансеров.

Дизайн сайта выполнен хорошо, верстка адаптивная.

Имеется возможность:

* опубликовать проект или конкурс, найти людей на временную или постоянную работу;
* стать исполнителем, найти работу, участвовать в конкурсе;
* фильтровать проекты по специализации, бюджету, ключевым словам и другим параметрам;
* выложить объявление с атрибутом «срочно»;
* воспользоваться поиском по исполнителям, проектам, разделам сайта;
* опубликовать рекламу на сайте;
* просмотреть личную информацию, рейтинги, отзывы исполнителей.

Рассмотрим внешний вид сайта «Fl.ru» на рисунке 1.1.1.

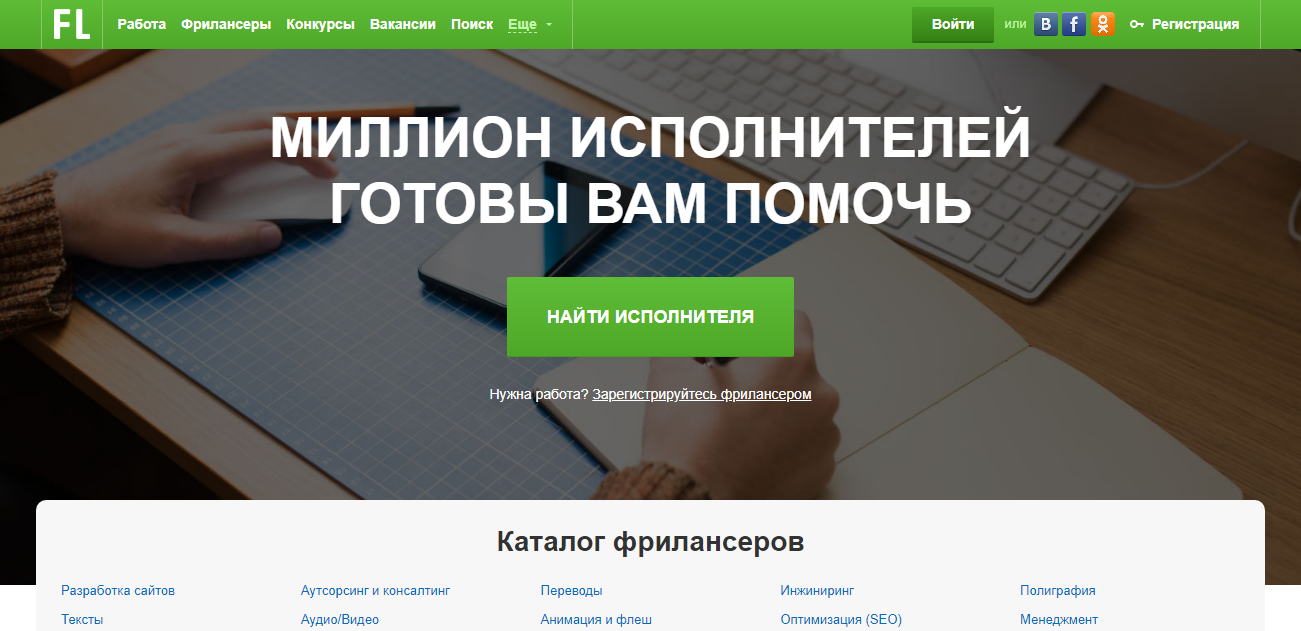


Рис. 1.1.1– Скриншот сайта «Fl.ru»

1. Биржа контента «eTXT» (https://www.etxt.ru)

Сайт «eTXT» предназначен в основном для копирайтеров и рерайтеров, а также можно найти много заказов для переводчиков.

Множество заказов выполняются для того, чтобы впоследствии использовать их при наполнении web-ресурсов.

Этот сайт подходит для новичков, так как работа там малооплачиваемая, тексты по большей части не сложные, перевод может быть не точный.

Дизайн выполнен не лучшим образом. Некоторые цвета слишком яркие, а некоторые - слишком бледные. Верстка на главной странице адаптивная, а на других - не всегда.

Имеется возможность:

* заказать написание текста или его перевод;
* стать исполнителем;
* подобрать исполнителя с помощью фильтра;
* купить готовый текст;
* фильтровать объявления по рубрике;
* просматривать информацию о пользователе, отзывы, статистику.

Рассмотрим внешний вид сайта «eTXT.ru» на рисунке 1.1.2

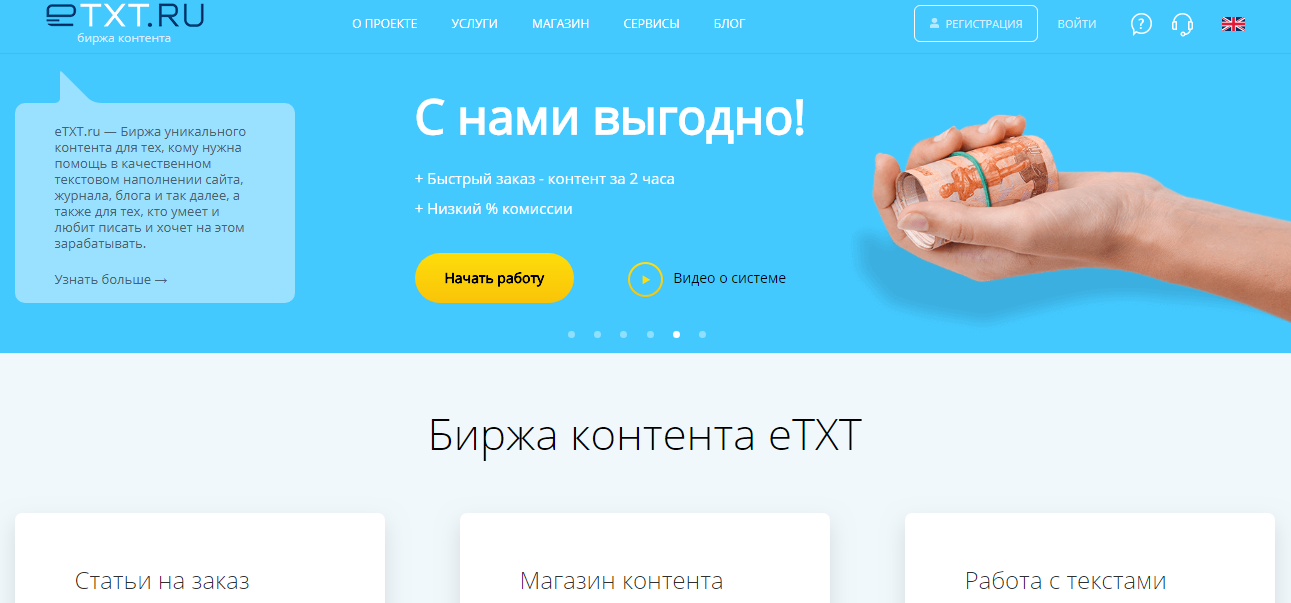


Рис. 1.1.2– Скриншот сайта «eTXT.ru»

1. Биржа переводов «Tranzilla» (https://tranzilla.ru)

«Tranzilla» - это специализированная биржа для переводчиков. Подходит как для профессионалов, так и для новичков. Можно найти как письменные, так и устные переводы.

На сайте можно в открытом доступе посмотреть контакты переводчиков, что позволяет связаться с ними напрямую.

Дизайн сделан довольно просто, выглядит неплохо, но слишком много элементов на страницах. Верстка адаптивная.

Имеется возможность:

* стать заказчиком;
* стать исполнителем;
* отфильтровать объявления по разным параметрам;
* просмотреть данные о переводчиках;
* общаться и размещать статьи в блоге.

Рассмотрим внешний вид сайта «Tranzilla.ru» на рисунке 1.1.3.

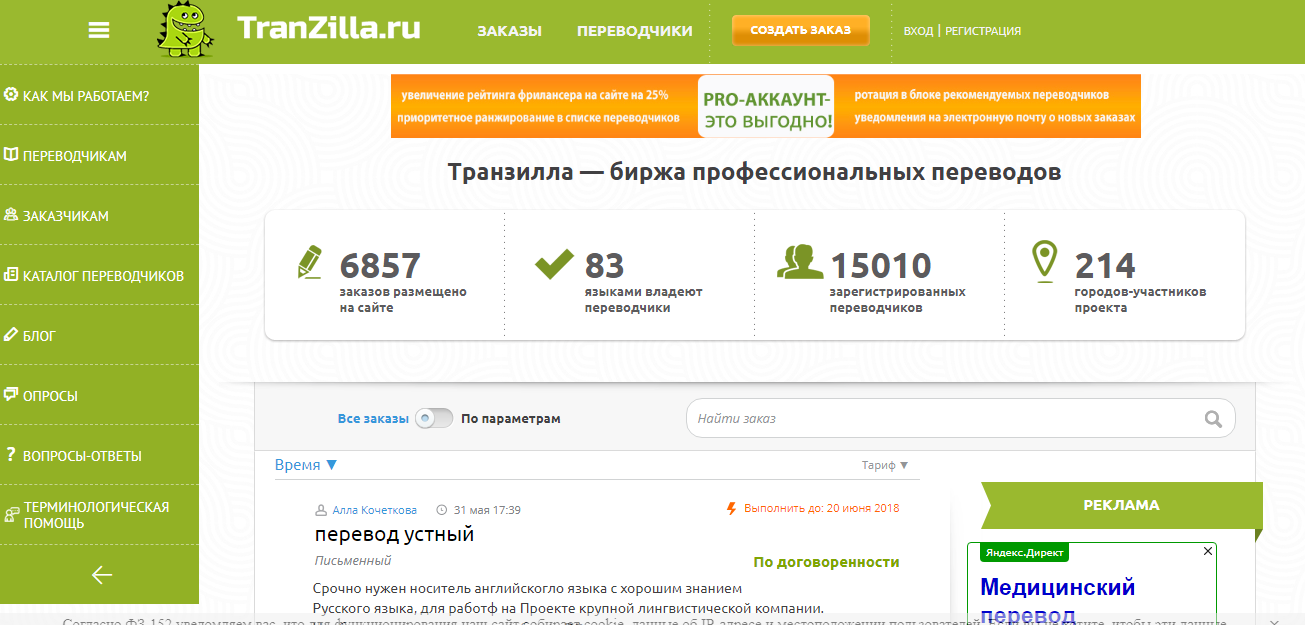


Рис. 1.1.3– Скриншот сайта «TranZilla.ru»

1. Биржа переводов «01» (<http://perevod01.ru>)

Биржа работает только с письменными переводами. На сайте заявлено, что переводчиками должны быть профессионалы.

Адаптивности на сайте нет, дизайн очень слабый.

Имеется возможность:

* стать переводчиком;
* стать заказчиком;
* найти переводчика, используя фильтр по языкам и специализации;
* просматривать заказы.

Рассмотрим внешний вид сайта «01» на рисунке 1.1.4.

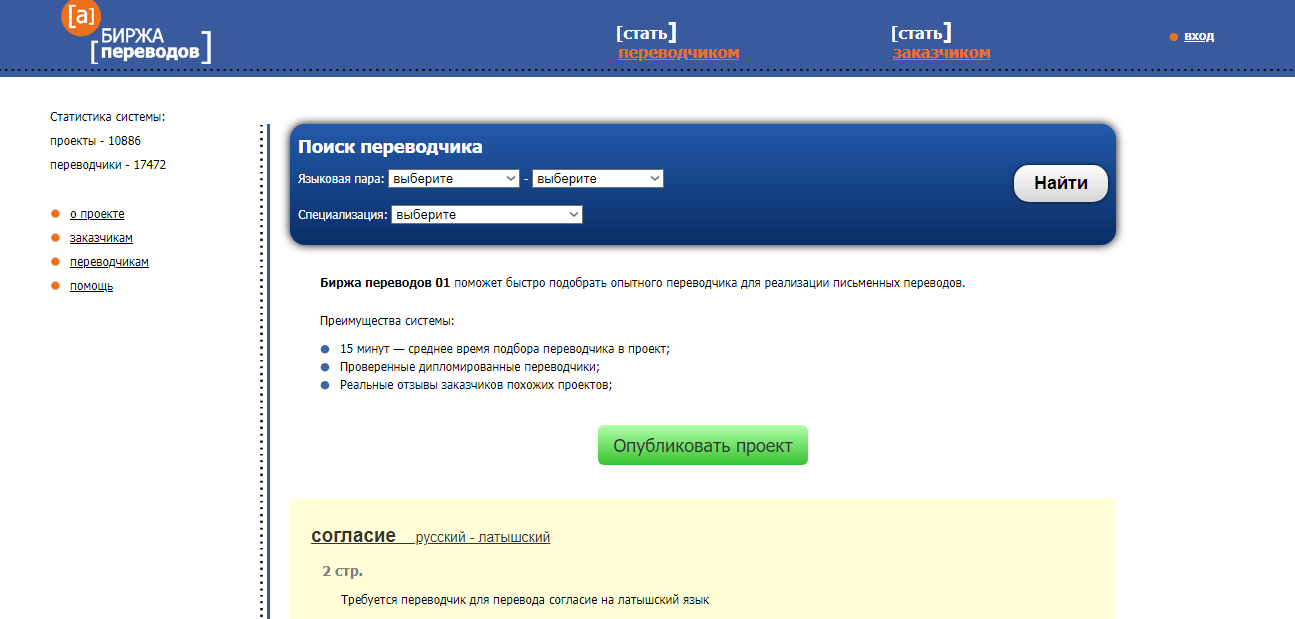


Рис. 1.1.4– Скриншот сайта «perevod01.ru»

1. Фриланс-платформа лингвистических услуг «2polyglot.com» (<https://2polyglot.com>)

Сервис для переводчиков, копирайтеров и репетиторов. Есть русскоязычная версия сайта.

Верстка не адаптивная, дизайн качественный только вверху главной страницы.

Имеется возможность:

* стать исполнителем;
* найти исполнителя;
* просмотреть профиль исполнителей;
* фильтровать объявления для поиска;
* прочитать статьи на тему переводов.

Рассмотрим внешний вид сайта «2polyglot.com» на рисунке 1.1.5.

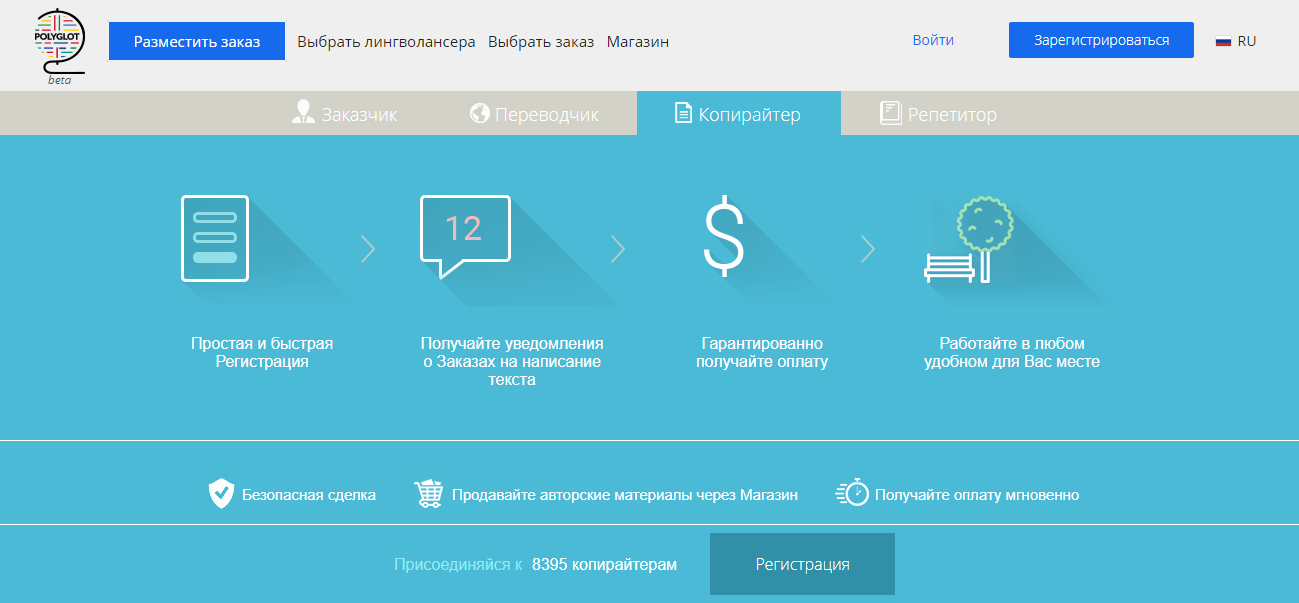


Рис. 1.1.5– Скриншот сайта «2polyglot.com»

1. Сервис по подбору переводчиков «Переводчик.Ме» (<https://perevodchik.me>)

Дизайн выполнен просто, но качественно. Верстка адаптивная.

Имеется возможность:

* найти переводчика;
* стать переводчиком;
* посмотреть профиль исполнителей;
* почитать новости, статьи.

Рассмотрим внешний вид сайта «Переводчик.Ме» на рисунке 1.1.6.

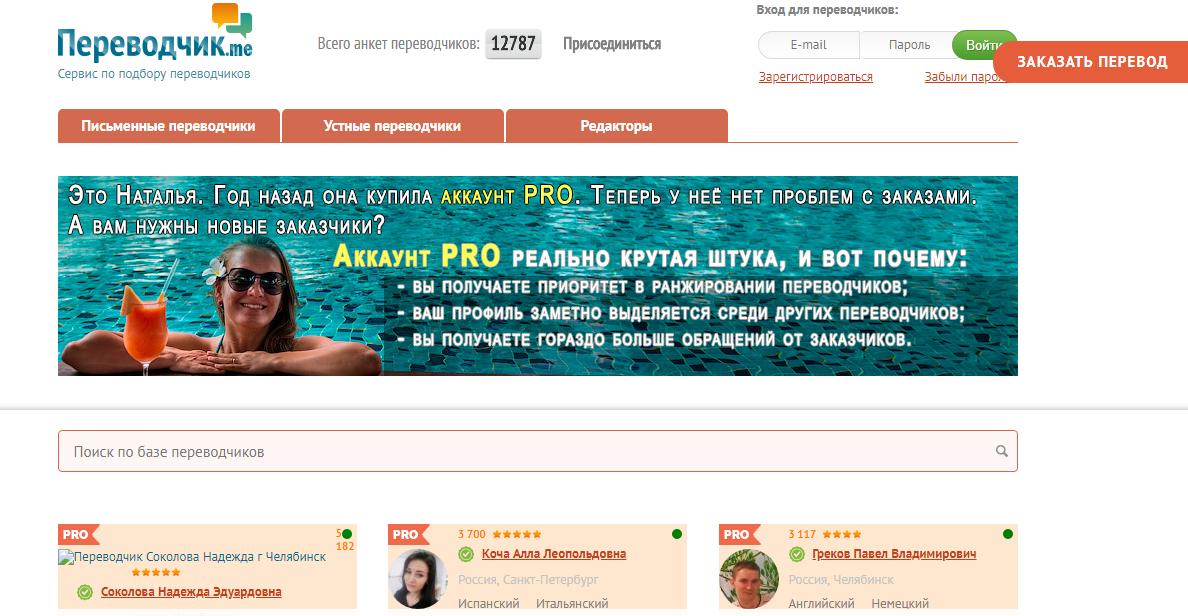


Рис. 1.1.6– Скриншот сайта «Perevodchik.me»

1. Бюро переводчиков «ProZ» (<https://www.proz.com>)

Одно из самых популярных в мире бюро для профессиональных переводчиков и переводческих компаний. Новичкам на этом сайте можно пройти обучение и проконсультироваться с профессионалами. Русскоязычной версии сайта нет.

Дизайн сайта выполнен качественно, верстка адаптивная.

Имеется возможность:

* пройти обучение по переводческим дисциплинам;
* нанять переводчика;
* найти переводчика с помощью фильтров;
* разместить вакансию;
* обсуждать на форуме переводческие темы;
* просмотреть словари;
* участвовать в мероприятиях;
* стать переводчиком.

Рассмотрим внешний вид сайта «ProZ» на рисунке 1.1.7.

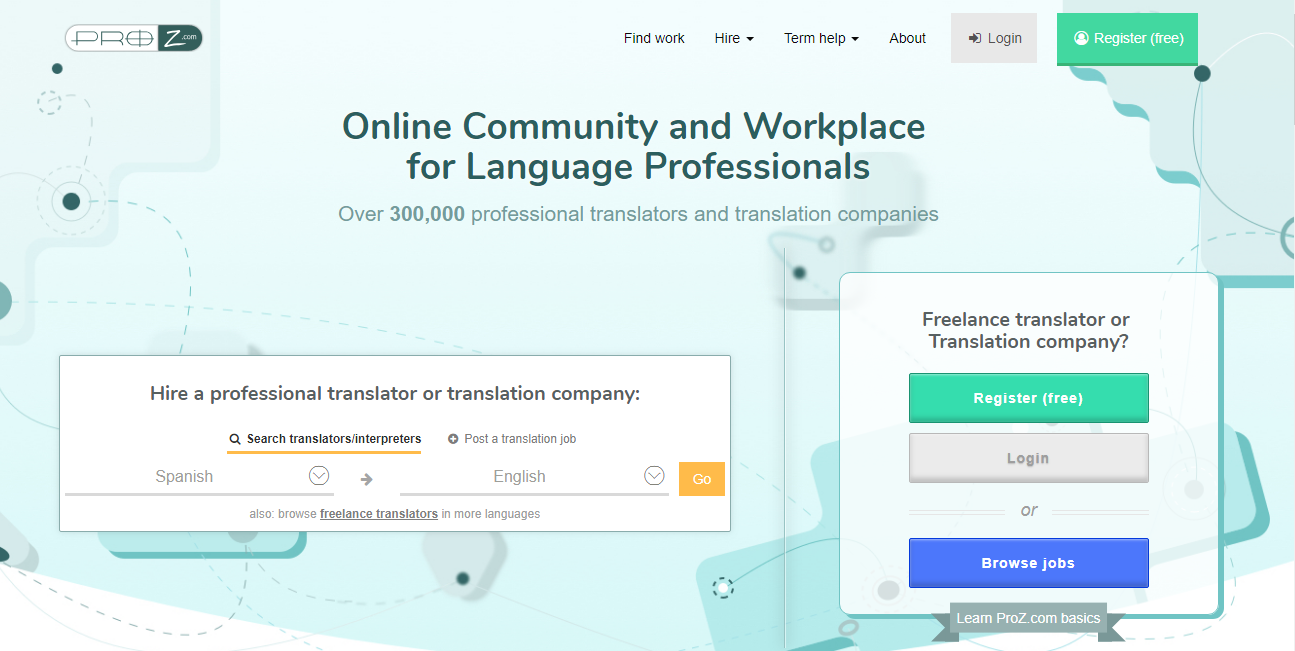


Рис. 1.1.7– Скриншот сайта «ProZ.com»

## **Сравнительный анализ программных продуктов**

Для сравнительного анализа программных продуктов были выбраны следующие параметры сравнения: добавление объявления, фильтрация объявлений, срочные переводы, просмотр выполненных заказов пользователей, форум, дизайн сайта, адаптивность. Введем количественные и качественные шкалы для оценки параметров.

1. Добавление объявления:

отлично (отл) – имеется краткое описание, подробное описание, добавление файла, тип перевода, срок сдачи, сложность работы, цена, язык оригинала, язык перевода;

хорошо (хор) – имеется краткое описание, подробное описание, срок сдачи, цена, язык оригинала, язык перевода;

удовлетворительно (уд) – имеется краткое описание и другие пункты, не указанные выше;

плохо (пл) – имеется только описание;

отсутствует (нет)– нет добавления объявления.

1. Фильтрация объявлений:

отлично (отл) – содержит поиск по специализации (для многопрофильных), цену, языки, тип работы, сортировку по дате и цене; работает без перезагрузки страницы;

хорошо (хор) – содержит поиск по специализации (для многопрофильных), тип работы, цену, языки; работает с перезагрузкой страницы, но выбранные пункты сохраняет при обновлении страницы;

удовлетворительно (уд) – содержит поиск по специализации (для многопрофильных), цену; работает с перезагрузкой страницы;

плохо (пл) – содержит какие-то другие пункты вместо указанных выше;

отсутствует (нет) – нет фильтрации объявлений.

1. Срочные переводы – есть ли возможность добавлять срочные переводы (Да/Нет).
2. Просмотр выполненных заказов пользователей – можно ли просматривать на странице пользователей выполненные ими заказы (Да/Нет).
3. Форум – есть ли возможность общения в виде форума или блога с комментариями (Да/Нет).
4. Дизайн сайта:

отлично (отл) – используются яркие цветовые схемы, хорошо читаемые шрифты, изображения оптимальных размеров, используется большая часть области отображения и, в целом, дизайн соответствует современным стандартам;

хорошо (хор) – некоторые перечисленные выше элементы отсутствуют, но это не портит общее впечатление от дизайна;

удовлетворительно (уд) – плохие сочетания цветов, неуместные элементы, устаревшие подходы к дизайну.

1. Адаптивность сайта – правильно ли отображается информация на всех устройствах, в том числе мобильных телефонах (Да/Нет).

Сравним программные продукты с помощью таблицы 1.2.1.

Таблица 1.2.1.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Параметр сравнения** | **«Fl»** | **«eTXT»** | **«Tranzilla»** | **«01»** | **«2polyglot»** | **«Перевод-**  **чик»** | **«ProZ»** |
| Добавление объявления | Уд | Уд | Уд | Нет | Хор | Уд | Уд |
| Фильтрация объявлений | Уд | Уд | Хор | Нет | Хор | Нет | Уд |
| Срочные переводы | Да | Нет | Нет | Нет | Да | Нет | Нет |
| Просмотр выполненных заказов пользователей | Да | Да | Нет | Нет | Да | Нет | Нет |
| Форум | Нет | Нет | Да | Нет | Нет | Нет | Нет |
| Дизайн сайта | Отл | Хор | Отл | Уд | Уд | Уд | Отл |
| Адаптивность сайта | Отл | Отл | Отл | Нет | Нет | Да | Да |

Изучив данные фриланс-биржи, я поняла, что портал «Translators' House» должен:

* позволять добавлять, изменять и удалять объявление; в нем должно хранится краткое содержание, подробное описание, добавление файла, тип перевода, срок сдачи, сложность работы, цена, язык оригинала, язык перевода;
* иметь фильтр по объявлениям, который должен содержать цену, язык оригинала, язык перевода, тип работы, сортировку по дате и цене; фильтр должен работать без перезагрузки страницы;
* иметь возможность добавлять срочные переводы;
* позволять просматривать на странице пользователей выполненные ими заказы;
* предоставлять пользователям возможность общения на форуме;
* быть адаптивным под все платформы;
* иметь лаконичный, стильный, современный и понятный дизайн.

## **Назначение и этапы создания web-портала**

Чтобы определить какой функционал для обеспечения контента web-портала необходим, нужно четко понимать назначение и цели его создания. Сформируем назначение и цели создания функционала для обеспечения контента портала в виде таблицы 1.3.1:

Таблица 1.3.1. Назначение и этапы создания функционала для обеспечения контента портала

|  |  |
| --- | --- |
| Характеристики проекта | Значения характеристик |
| Назначение функционала для обеспечения контента портала | |
| Вид деятельности портала (для чего нужен web-портал). | Web-портал сообщества переводчиков «Translator’s house» предназначен для поиска переводчиков для личных и служебных целей и общения на тему переводов. |
| Перечень объектов информатизации, на которых предполагается ее использовать. | Организуется в виде клиент-серверного приложения с помощью языка PHP с использованием фреймворка Laravel. СУБД MySQL, языка JS, библиотеки JQuery, Ajax, языка гипертекстовой разметки HTML, каскадных таблиц стилей CSS, фреймворка Bootstrap. |
| Этапы создания функционала для обеспечения контента портала | |
| Определения наименования и требуемые значения технических показателей, которые должны быть достигнуты в результате создания портала. | Возможность добавления, изменения, удаления объявлений; фильтрации для удобного поиска по объявлениям; панели для редактора web-портала; просмотра для пользователя объявлений, которые он выложил или взял на исполнение; реализация форума для общения. |
| Определения наименования и требуемые значения технологических показателей, которые должны быть достигнуты в результате создания портала. | Поиск работников-фрилансеров для временной работы в сфере переводов, общение на тему переводов с пользователями web-портала. |
| Определения наименования и требуемые значения производственно-экономических показателей, которые должны быть достигнуты в результате создания портала. | Возможность сэкономить на штатных сотрудниках, отдавая работу на аутсорсинг; возможность заработать на удаленной работе для переводчиков. |
| Выявить критерии для разработки портала. | Добавление объявления – отлично; фильтрация объявлений – отлично; срочные переводы – да; просмотр выполненных заказов пользователей – да; форум – да; дизайн сайта – отлично; адаптивность- отлично. |
| Результат полученный в процессе проектирования web-портала. | Функционал для обеспечения контента сообщества переводчиков «Translator’s house». |

# **Проектирование web-портала**

На данном этапе необходимо спроектировать необходимый функционал программного продукта и модули из которых будет состоять функционал для обеспечения контента сайта.

Web-портал «Translators' House» реализуется в виде клиент-серверного приложения. Взаимодействие клиента и сервера происходит на основе сетевых протоколов. Каждый из протоколов поддерживает оказание той или иной услуги.

Клиент и сервер обмениваются в процессе взаимодействия сообщениями, так называемыми НТТР запросами со стороны клиента и НТТР ответами со стороны сервера. Запросы включают в себя методы, сообщающие серверу, как следует обрабатывать сообщение. А ответы сервера содержат коды состояния для обратной связи с браузером.

Приложения «клиент-сервер» устроены так, что инициатива всегда на стороне клиента. Клиент отправляет запрос на услугу, а сервер реагирует на действие клиента.Для осуществления связи «клиент-сервер» разрабатывается серверное и клиентское программное обеспечение, которое может работать как на разных устройствах, так и на одном компьютере. Предполагается, что к одному серверу может обращаться одновременно несколько клиентов. Насколько много пользователей могут одновременно находиться на сайте и получать услуги, зависит от мощности сервера и вида услуг или контента на данном сайте.

Преимущества модели «клиент-сервер»:

* разделение кода на клиентские и серверные приложения;
* отсутствие высоких требований к клиентским устройствам при работе в локальных компьютерных сетях;
* компьютерные устройства не делятся на только клиентские и только серверные, что позволяет распределять нагрузку между всеми участниками процесса.

Для моделирования клиент-серверного приложения отлично подойдет язык графического описания UML. Этот язык используется для визуализации процессов разрабатываемых программных систем. С помощью UML проводится анализ на этапе проектирования системы, на всех этапах разработки, а также ведется сопровождение полученного программного продукта.

Как и любой язык программирования, UML состоит из словаря и правил, по которым составляются языковые конструкции. Так как UML – графический язык, он состоит из графических элементов. Модели, разработанные на нем, поддерживаются программными средствами, интерпретирующими UML. При работе с этим языком разработчик использует готовые элементы языка, но не получает готовых инструкций, какие модели следует разрабатывать. Именно поэтому язык UML имеет широкую область применения и отличается вариативностью.

В основе этого языка лежат четыре типа сущностей, каждый из которых имеет свое графическое представление: структурные, поведенческие, группирующие и аннотационные. На строящихся диаграммах показываются сущности и отношения между ними.

В UML используются несколько типов отношений.

Зависимость – тип отношений, при котором изменение одной сущности вызывает изменение другой, зависимой сущности. Изображается пунктирной линией со стрелкой.

Ассоциация – тип отношений, показывающий связь между объектами разных сущностей. Используется для реализации навигации между объектами в программе. Изображается линией, связывающей сущности. Однонаправленная навигация указывается стрелкой на конце линии.

Агрегирование – подвид ассоциации, показывающий связь между целым и частью. Сущность «целое» в данном случае помечается ромбиком на конце линии.

Обобщение – это сущность, показывающая свойства наследования для различных классов и объектов. Обобщение связывает родительскую сущность с сущностями-потомками. На стороне родительской сущности линия заканчивается маленьким треугольником.

Реализация – отношение, использующееся для моделирования компонент и классов.

В UML используются диаграммы, описывающие поведение системы, и диаграммы, описывающие ее физическую реализацию. К первому типу диаграмм относятся диаграммы состояний, деятельностей, объектов, последовательностей и взаимодействия. Ко второму типу относятся диаграммы компонент и диаграммы развертывания.

Преимущества по сравнению с другими языками такого типа:

* легкость построения диаграмм деятельностей, сценариев и состояний;
* возможность построения концептуальных моделей в виде диаграмм классов данных;
* описание процессов взаимодействия между объектами системы;
* описание взаимодействия программных компонент и физической архитектуры системы.

## **Диаграмма вариантов использования (Use Case diagram)**

Данная диаграмма используется для того, чтобы понять как должно функционировать приложение. Представляется в виде последовательности действий, реагирующих на внешние воздействия. В состав диаграммы входят актеры и варианты использования. В зависимости от того какой актер воздействует на систему, такие действия и выполняются.

Продемонстрируем актеров и соответствующие им действия с помощью таблицы 2.1.

Таблица 2.1. Актеры и варианты использования

|  |  |
| --- | --- |
| Актер | Вариант использования |
| Незарегистрированный пользователь, Зарегистрированный пользователь, Редактор | Просмотреть главную страницу |
| Незарегистрированный пользователь, Зарегистрированный пользователь, Редактор | Просмотреть заказы |
| Незарегистрированный пользователь, Зарегистрированный пользователь, Редактор | Фильтровать заказы |
| Незарегистрированный пользователь, Зарегистрированный пользователь, Редактор | Просматривать форум |
| Зарегистрированный пользователь | Добавить тему на форум |
| Зарегистрированный пользователь | Удалить свою тему из форума |
| Зарегистрированный пользователь | Изменить свою тему из форума |
| Зарегистрированный пользователь | Добавить сообщение на форум |
| Зарегистрированный пользователь | Удалить свое сообщение из форума |
| Зарегистрированный пользователь | Изменить свое сообщение из форума |
| Зарегистрированный пользователь | Просмотреть выполненные работы пользователей, которые откликнулись на заказ |
| Зарегистрированный пользователь | Добавить заказ |
| Зарегистрированный пользователь | Удалить свой заказ |
| Зарегистрированный пользователь | Изменить свой заказ |
| Редактор | Управлять панелью управления |

Продемонстрируем актеров и соответствующие им действии с помощью диаграммы использования на рисунке 2.1.1.

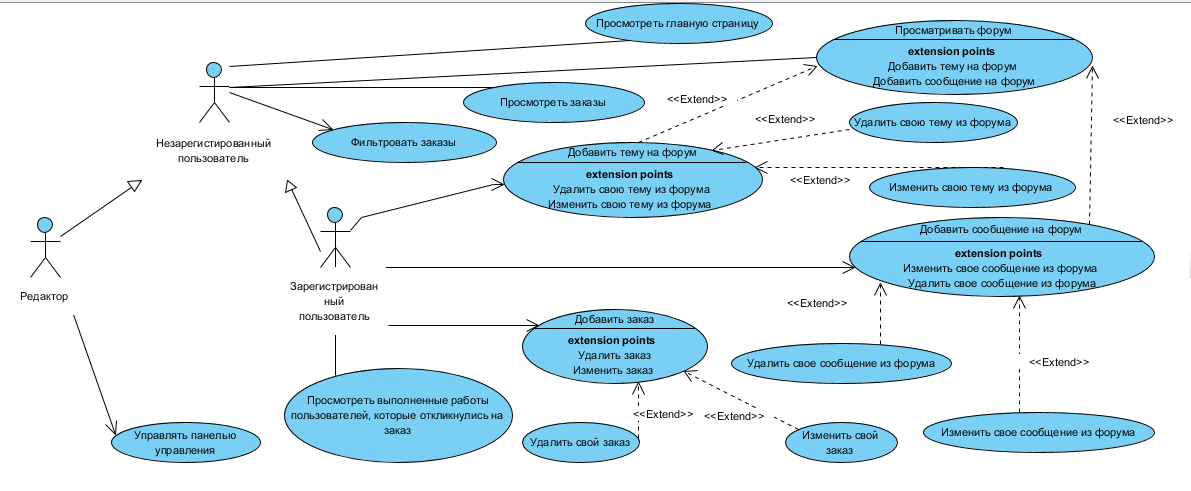


Рис. 2.1.1– Диаграмма использования

Между вариантами использования и актерами для функционала части для управлением контентом web-портала сообщества переводчиков используются следующие виды взаимодействия: простая ассоциация (обозначается простой линией); направленная ассоциация, которая показывает что актер взаимодействует с системой (обозначается линией со стрелкой); наследование, которое показывает что потомок наследует поведение своего предка (обозначается линией с треугольником на конце); расширение, показывает что вариант использования расширяет последовательность действий (обозначается пунктирной линией со стрелкой с ключевым словом «extend».

## **Диаграмма последовательности (Sequence diagram)**

Диаграмма последовательности показывает взаимодействие (обмен информацией) некоторых элементов приложения, которые работают поочередно.

Построим диаграмму последовательности для модуля «Добавить объявление» (Рис. 2.2.1).

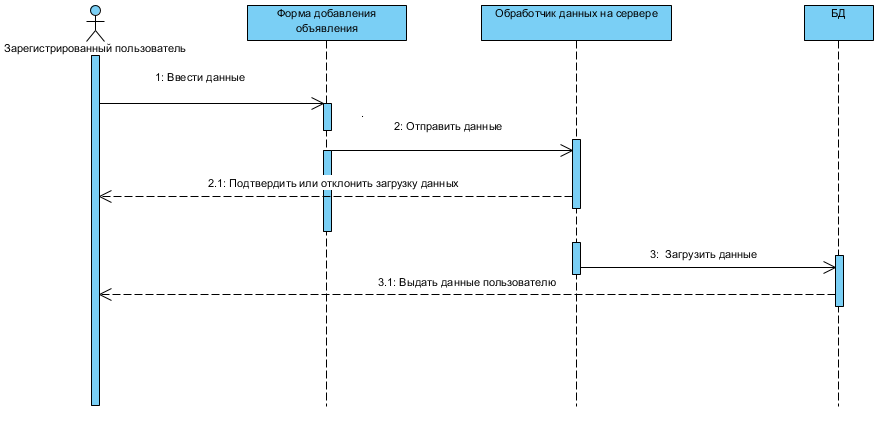


Рис. 2.2.1 – Диаграмма последовательности

На диаграмме показано как взаимодействуют элементы модуля «Добавить объявление». Пользователь вводит данные, форма добавления объявления отправляет данные на сервер, обработчик на сервере подтверждает или отклоняет загрузку данных. Если обработчик на сервере подтверждает загрузку данных, то загружает их в БД и выдает данные из БД пользователю.

## **Диаграмма коммуникации (Communication diagram)**

Диаграмма коммуникации похожа на диаграмму последовательности. Она тоже показывает взаимодействие элементов приложения, только другими графическими средствами.

Построим диаграмму коммуникации для модуля «Отфильтровать объявления» (Рис. 2.2.1).

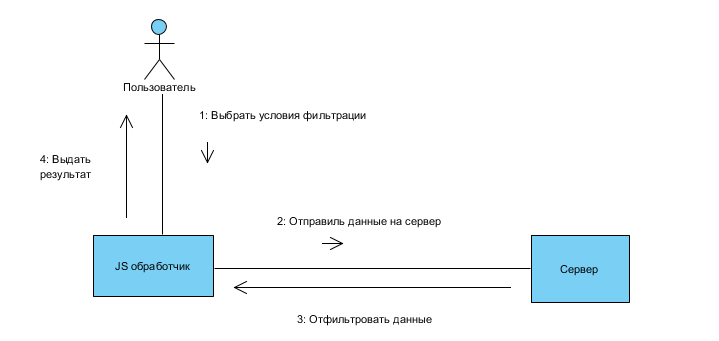


Рис. 2.2.1 – Диаграмма коммуникации

На диаграмме показано как взаимодействуют элементы модуля «Отфильтровать объявления». Пользователь выбирает условия фильтрации, данные обрабатываются JS обработчиком и отправляются на сервер. Сервер фильтрует данные и отправляет JS обработчику. JS обработчик выдает результат пользователю.

## **Диаграмма состояний (Statechart diagram)**

Диаграмма состояний показывает как объекты переходят из одного состояния в другое.

Построим диаграмму состояний для модуля «Добавить объявление» (Рис. 2.4.1).

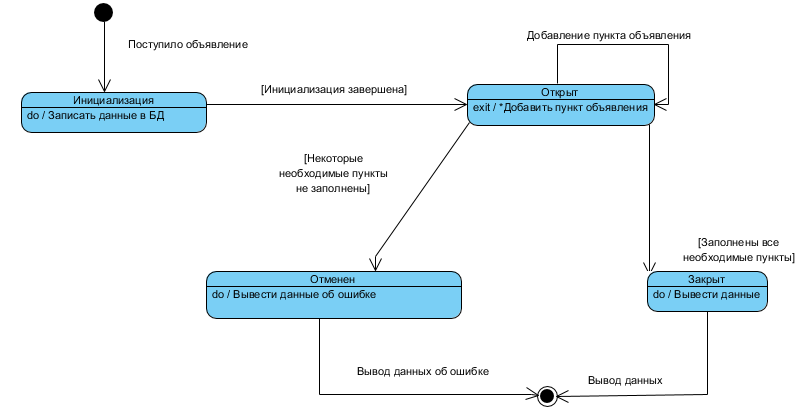


Рис. 2.4.1– Диаграмма состояний

Запись объявления может находится в состояниях «Инициализация», «Открыт», «Отменен», «Закрыт», а также начальном и конечном состоянии.

Изначально запись объявления находится в начальном состоянии.

После того как объявление поступило, запись переходит в состояние «Инициализация», действие «записать данные в БД» начинает выполняться. Это действие будет выполняться пока запись будет находится в данном состоянии. При условии, что инициализация завершена запись объявления переходит в состояние «Открыт».

В состоянии «Открыт» выполняется выходное действие «Добавить пункт объявления» и переход в самого себя. Таким образом действие «Добавить пункт объявления» выполняется циклично. Также каждый раз выполняется условие «заполнены все необходимые пункты» или «Некоторые необходимые пункты не заполнены». В зависимости от выполнения условия запись объявления переходит в состояние «Закрыт» (если все необходимые пункты заполнены) или «Отменен» (если некоторые необходимые пункты не заполнены). Символ «\*» обозначает, что это действие выполнится несколько раз (по числу добавленных пунктов в объявление).

В состоянии «Закрыт» выполняется действие «Вывести данные». После того как данные вывелись запись объявления переходит в конечное состояние.

В состоянии «Отменен» выполняется действие «Вывести данные об ошибке». После того как данные об ошибке вывелись запись объявления переходит в конечное состояние.

## **Диаграмма классов (Class diagram)**

Диаграмма классов показывает зависимости между классами и экземплярами классов. В этой диаграмме каждый класс имеет имя, атрибуты и операции.

Атрибут необходим для описания свойств объектов класса. Указывается имя атрибута и его тип (может указываться значение по умолчанию).

Операция – это функция которая может выдавать и принимать значения.

Диаграмма классов может применяться для клиент-серверного приложения. В качестве имени классов используются имена таблиц БД. Атрибутами являются поля таблиц. Операции – свойства полей таблиц (чаще всего get и set).

Рассмотрим диаграмму классов на рисунке 2.5.1.

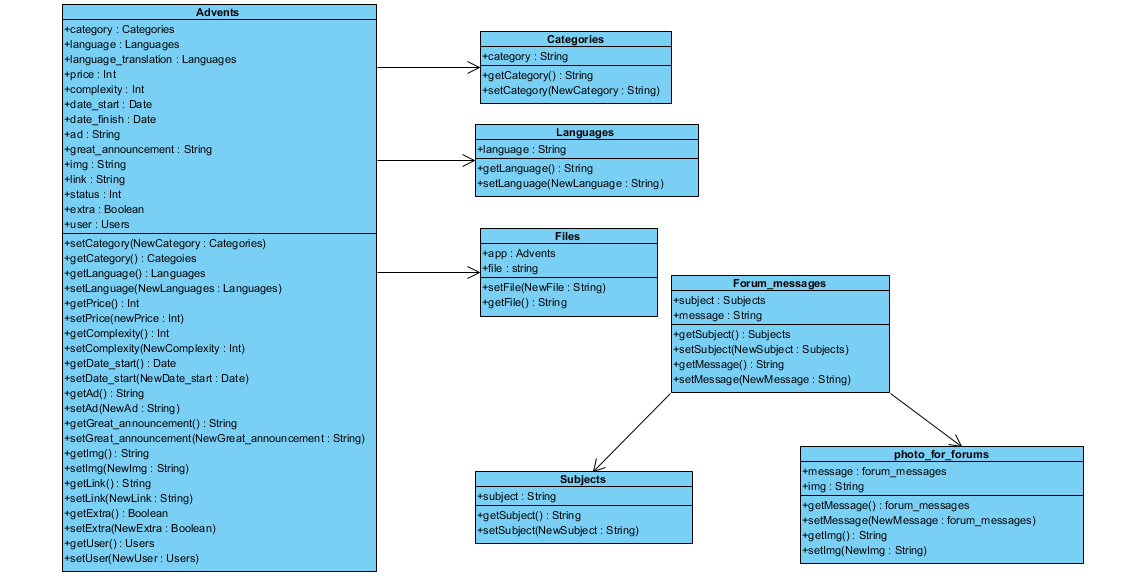


Рис. 2.5.1– Диаграмма классов

Данная диаграмма определяет какие таблицы должны быть созданы и какие поля должны в них присутствовать. Также нужно определиться с отношениями между таблицами.

Необходимо создать следующие таблицы:

* Сategories с полем category для категорий переводов (текстовый, видео, аудио, озвучка и другие);
* Files с полями app, file для хранения файлов для объявлений;
* Forum\_messages с полями subject, message, user для сообщений в форуме;
* Languages с полем language для языков переводов;
* Photo\_for\_forums с полями img, message для изображений в форуме;
* Subjects с полями subject для тем форума;
* Advents с полями type\_category, language, language\_translation, price, complexity, date\_start, date\_finish, ad, great\_announcement, img, user, link, status, extra для объявлений (заказов). Для каждого объявления для поиска переводчиков нужно будет добавлять id категории перевода, id языка оригинала, id языка перевода, id пользователя, цену (может равняться нулю при желании), сложность, временной интервал. Также можно будет добавить краткую запись о переводе, более подробную запись о переводе, изображение, ссылку. Еще необходима возможность добавить файлы, которые будут храниться в таблице Files. Поле status будет менять в зависимости от статуса заказа (в поиске переводчиков, выполняется, завершен).

Также так как web-портал «Translator’s house» является программным продуктом коллективной разработки для диаграммы классов нам понадобится таблица Users созданная другим разработчиком, в которой хранится информация о пользователях. Id пользователя понадобится при дальнейшей разработке.

## **Диаграмма компонентов (Component diagram)**

Диаграммы рассмотренные ранее относились к логическому уровню представления. В отличии от них диаграмма компонентов относится к физическому уровню.

Построим диаграмму компонентов рассматривающую взаимодействие пользователя, HTML страницы, JS обработчика и сервера (Рис.2.6.1).

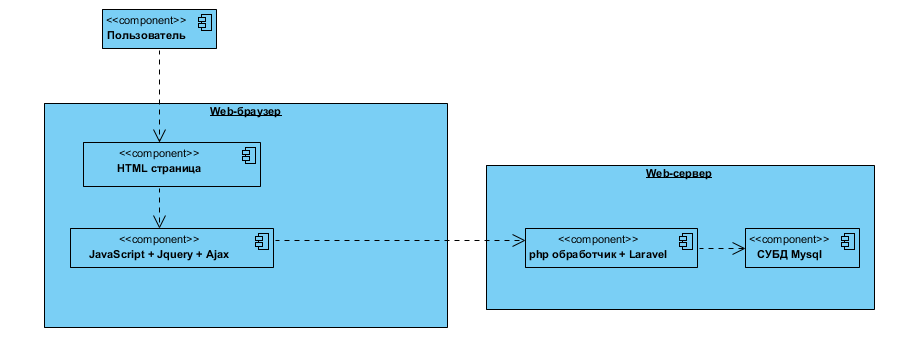


Рис. 2.6.1– Диаграмма компонентов

## **Даталогическая модель**

Даталогическая модель базы данных – это модель, отражающая логические взаимосвязи между элементами данных безотносительно их содержания и физической организации. Даталогическая модель разрабатывается с учетом конкретной реализации СУБД, также с учетом особенностей предметной области на основе инфологической модели.

Результатом построения даталогической модели становится собственно разрабатываемая база данных. Все решения, принятые на этапе разработки даталогической модели и предыдущих этапов, сказываются на конечном результате – готовом программном продукте.

Создадим даталогическую модель в приложении phpMyAdmin (Рис.2.7.1).

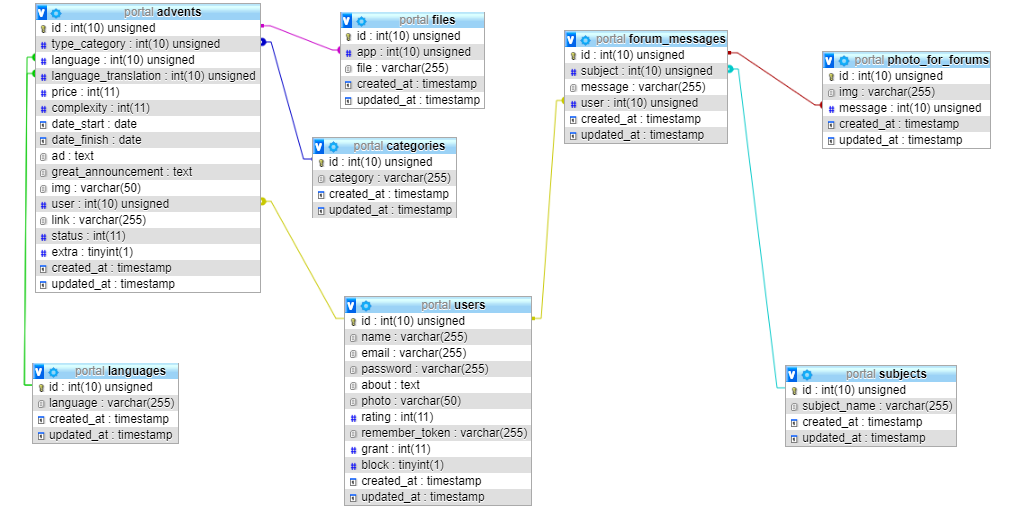


Рис. 2.7.1– Даталогическая модель

Необходимым условием для хранения и последующего использования информации является приведение таблиц в 3 нормальную форму (3НФ).

Таблица находится в первой нормальной форме (1НФ), если в каждой ячейке содержится только одно значение, нет повторяющихся строк в таблице, у каждого поля должно быть уникальное значение.

Чтобы таблица была во второй нормальное форме (2НФ), она должна быть в 1НФ, и все ее не ключевые столбцы должны однозначно определяться ключом.

Таблица находится в третей нормально форме (3НФ), когда находится во 2НФ и каждый не ключевой атрибут нетранзитивно зависит от первичного ключа. Проще говоря, второе правило требует выносить все не ключевые поля, содержимое которых может относиться к нескольким записям таблицы в отдельные таблицы.

Рассмотрим таблицу advents. Есть ключевой атрибут – id и не ключевые атрибуты: type\_category (связь с таблицей categories), language (связь с таблицей languages), language\_translation (связь с таблицей languages), price, complexity, date\_start, date\_finish, ad, great\_announcemen, img, user (связь с таблицей users), link, status, extra, created\_at, updated\_at. Отношение находится в 1НФ, так как в каждой ячейке содержится только одно значение, нет повторяющихся строк в таблице, у каждого поля уникальное значение. Также таблица находится в 2НФ, так как все ее не ключевые столбцы должны однозначно определяться ключом. Отношение находится в 3НФ, так как транзитивной зависимости нет.

Рассмотрим таблицу categories. Есть ключевой атрибут – id и не ключевые атрибуты: category, created\_at, updated\_at. Отношение находится в 1НФ, так как в каждой ячейке содержится только одно значение, нет повторяющихся строк в таблице, у каждого поля уникальное значение. Также таблица находится в 2НФ, так как все ее не ключевые столбцы должны однозначно определяться ключом. Отношение находится в 3НФ, так как транзитивной зависимости нет.

Рассмотрим таблицу files. Есть ключевой атрибут – id и не ключевые атрибуты: app (связь с таблицей advents), file, created\_at, updated\_at. Отношение находится в 1НФ, так как в каждой ячейке содержится только одно значение, нет повторяющихся строк в таблице, у каждого поля уникальное значение. Также таблица находится в 2НФ, так как все ее не ключевые столбцы должны однозначно определяться ключом. Отношение находится в 3НФ, так как транзитивной зависимости нет.

Рассмотрим таблицу forum\_messages. Есть ключевой атрибут – id и не ключевые атрибуты: subject (связь с таблицей subjects), message, user (связь с таблицей users), created\_at, updated\_at. Отношение находится в 1НФ, так как в каждой ячейке содержится только одно значение, нет повторяющихся строк в таблице, у каждого поля уникальное значение. Также таблица находится в 2НФ, так как все ее не ключевые столбцы должны однозначно определяться ключом. Отношение находится в 3НФ, так как транзитивной зависимости нет.

Рассмотрим таблицу languages. Есть ключевой атрибут – id и не ключевые атрибуты: language, created\_at, updated\_at. Отношение находится в 1НФ, так как в каждой ячейке содержится только одно значение, нет повторяющихся строк в таблице, у каждого поля уникальное значение. Также таблица находится в 2НФ, так как все ее не ключевые столбцы должны однозначно определяться ключом. Отношение находится в 3НФ, так как транзитивной зависимости нет.

Рассмотрим таблицу photo\_for\_forums. Есть ключевой атрибут – id и не ключевые атрибуты: img, message (связь с таблицей messages), created\_at, updated\_at. Отношение находится в 1НФ, так как в каждой ячейке содержится только одно значение, нет повторяющихся строк в таблице, у каждого поля уникальное значение. Также таблица находится в 2НФ, так как все ее не ключевые столбцы должны однозначно определяться ключом. Отношение находится в 3НФ, так как транзитивной зависимости нет.

Рассмотрим таблицу subjects. Есть ключевой атрибут – id и не ключевые атрибуты: subject\_name, created\_at, updated\_at. Отношение находится в 1НФ, так как в каждой ячейке содержится только одно значение, нет повторяющихся строк в таблице, у каждого поля уникальное значение. Также таблица находится в 2НФ, так как все ее не ключевые столбцы должны однозначно определяться ключом. Отношение находится в 3НФ, так как транзитивной зависимости нет.

Как уже говорилось ранее, так как web-портал «Translator’s house» является программным продуктом коллективной разработки нам понадобится таблица Users созданная другим разработчиком, в которой хранится информация о пользователях.

# **Анализ и выбор программно-инструментальных средств**

В современном мире использование фреймворков имеет массу преимуществ для разработчиков программного обеспечения. Работа с фреймворком сильно упрощает работу по созданию различных проектов.

Основные преимущества:

* возможность разработки сайта любой сложности и с любым функционалом;
* сокращение времени выполнения поставленной задачи и оптимизация рабочего процесса;
* сокращение материальных затрат на создание приложений.

Фреймворки предоставляют готовые решения и берут на себя большую часть технической работы. Кроме того, сайты, написанные на фреймворках, работают быстрей, чем сайты, написанные без их использования. Сайты на фреймворках работают быстрее, потому что серевер обрабатывает и выдает только ту информацию, которую запросил клиент.

Язык РНР – это популярный скриптовый язык для разработки веб-приложений. Этот язык позволяет программисту быстро и эффективно решать задачи по созданию сайтов.

Язык РНР имеет ряд преимуществ по сравнению с аналогичными языками программирования:

* язык легко осваивать с нуля, потому что его конструкции традиционны, многие заимствованы из более старых языков программирования, синтаксис ясный и привычный;
* процесс написания программы достаточно прост, нет сложной компиляции, есть большое количество удобных встроенных функций;
* производительность РНР достаточно высока для создания функциональных сайтов и приложений;
* в РНР реализованы механизмы безопасности как системного уровня, так и на уровне приложений;
* сценарии РНР передаются любым устройствам, включая мобильные устройства, ноутбуки, электронные книги;
* язык РНР распространяется бесплатно, что очень удобно для обучения программированию и использованию при разработке программного обеспечения.

Рhp-фреймворки как нельзя лучше позволяют структурировать и организовывать процесс разработки.

Использование php-фреймворка позволяет дополнять функционал сайтов по желанию заказчиков и адаптировать сайты к меняющейся реальности.

## **Инструментальные средства разработки**

Сравним наиболее популярные php-фреймворки.

1. Laravel

На данный момент он считается самым популярным php фреймворком, так как он имеет ряд преимуществ:

* подходит для новичков, так как имеет достаточно простой синтаксис и функционал. Синтаксис не сильно отличается от чистого php, что позволяет быстро разобраться в работе фреймворка;
* использует архитектурную модель [MVC](https://ru.wikipedia.org/wiki/Model-View-Controller) ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) Model View Controller — модель-представление-контроллер). Что позволяет разграничивать работу с базой данных, логикой и пользовательским интерфейсом с помощью таких компонентов как модель, контроллер и представление. Модель предоставляет данные из базы данных и реагирует на команды контроллера, изменяя свое состояние. Представление отображает данные модели пользователю, реагируя на изменение модели. Контроллер интерпретирует действия пользователя, сообщая модели о необходимости изменений;
* включает в себя встроенный интерфейс командной строки artisan, которая позволяет автоматизировать создание некоторых компонентов;
* в Laravel встроен конструктор запросов, представляющий удобный интерфейс для реализации запросов к базе данных. На данный момент он поддерживает такие СУБД как  MySQL, Postgres, SQLite и SQL Server. В конструктор запросов встроены механизмы защиты от SQL-инъекций;
* связывает генерируемые приложением ссылки и маршруты с помощью маршрутизации. Позволяет работать с любыми HTTP-запросами;

1. CodeIgniter

Данный фреймворк пользуется популярностью из-за своей простоты, удобства, большого количества документации. Фреймворк Laravel, о котором говорилось выше, создавался как конкурент CodeIgniter, так как до появления Laravel данный фреймворк был самым популрным.

CodeIgniter использует шаблоны для работы с БД, которые похожи на язык запросов sql. Также в CodeIgniter есть возможность подключения библиотек из командной строки с помощью менеджера пакетов.

1. Symfony

Вторая версия Symfony занимает третье место по популярности среди фреймворков благодаря высокой скорости работы и простоте синтаксиса. Symfony подходит для разработки таких продуктов как поисковые системы и интернет-магазины.

Считается, что фреймворк удобен для написания кода из командной строки.

Первая версия Symfony имеет официальную документацию, переведенную на русский язык. Перевод второй и третьей версии также доступен, но только из неофициальных источников.

1. Yii

Фреймворк Yii во многом похож на Symfony, однако имеет по сравнению с ним ряд преимуществ:

* MVC-взаимодействие;
* более удобный интерфейс;
* меньше временных затрат на разработку аналогичных приложений;
* более высокая скорость работы программ, написанных на этом фреймворке.

1. Nette Framework

Фреймворк, не оцененный разработчиками несмотря на достаточно широкие возможности:

* хорошую производительность;
* удобные инструменты для отслеживания ошибок и тестирования программ;
* возможность групповой работы нескольких программистов над одним проектом;
* удобную документацию.

При аназизе данной информации выбор остановился на php-фреймворке Laravel.

Для работы с базой данных будет использоваться СУБД MySQL и сборка серверов XAMPP.

Для логики, используемой на клиентской части приложения используются js, библиотеки JQuery, Ajax.

При использовании Ajax не происходит обновление страницы после каждой отправки данных на сервер, так как обновляется только ее определенная часть. Это намного удобнее, так как пользователям не приходится долго ждать.

Обобщим достоинства Ajax:

* возможность создания удобного интерфейса для пользователей;
* активное взаимодействие с пользователем;
* вместо перезагрузки страницы, перезагружается только ее часть;
* удобство использования;

Для реализации пользовательского интерфейса используются язык гипертекстовой разметки HTML5, каскадные таблицы стилей CSS3, фреймворк Bootstrap.

Плюсы использования фрейворка Bootstrap:

– Значительно ускоряет процесс написания приложения, так как содержит достаточное количество шаблонов. Шаблоны лучше использовать не в первоначальном виде, а модифицировать, чтобы на выходе программный продукт отличался своим оригинальным дизайном.

– Адаптивность и кросс-браузерность. Наличие специальной сетки в данном фреймворке позволяет писать web-приложения для разных браузеров и любых устройств, не тратя на это много времени.

– Простота использования. Работать с Bootstrap могут даже самые начинающие разработчики. Поэтому этот фреймворк изучается во многих учебных заведениях.

Так как web-портал «Translator’s house» является совместной разработкой, используется GitHub. GitHub – популярный web-сервис, основанный на системе контроля версий Git, предназначенный для совместной разработки.

## **Обзор программных средств реализации интерфейса**

1. PhpStorm – IDE от компании JetBrains, имеющая высокую производительность, множество полезных функций, приятный дизайн. Так как PhpStorm не просто редактор, а интегрированная среда разработки, то можно переключаться между файлами не выходя из него. Есть возможность менять дизайн. Также удобной функцией является поиск и замена определенного слова во всем проекте, что значительно сокращает время при необходимости изменения чего-либо. PhpStorm имеет множество официальной и неофициальной документации. Возможна интеграция в GitHub. PhpStorm имеет встроенную командную строку.
2. Netbeans - применяется для языков программирования Java, PHP, Python, C/C++ и других. Приложение запускается на Linux, Windows и OSX. При разработке больших проектов произовдительность Netbeans может быть снижена.
3. Sublime Text это кросс-платформенный редактор с привлекательным интерфейсом. Он быстрый и богат функционалом, для практически каждого языка программирования. Поддерживает сворачивание кода, макросы, проекты и другое. Запускается на Linux, Windows и OSX.
4. Notepad++ подходит и для новичков, и для профессионалов. Notepad++ поддерживает каждый из популярных языков программирования. Поддерживает разбиение рабочей среды на два окна редактирования.
5. Dreamweaver поддерживает только наиболее популярные языки веб программирования – PHP, ASP.NET, JavaScript, HTML, CSS. Очень удобный для новичков. Dreamweaver доступен на OSX и Windows.
6. Aptana Studio это среда разработки предназначена для AJAX приложений. Эта среда поддерживает большинство популярных веб языков: PHP, JavaScript, HTML, CSS, Ruby, Python и другие с помощью плагинов.

Будет использоваться PhpStorm, так как он имеет множество необходимых функций.

# **Проектирование базы данных**

Для создания базы данных используются миграции фреймворка laravel. Миграции позволяют создавать и редактировать таблицы базы данных. Создание таблиц именно таким образом удобно для проекта, в котором несколько разработчиков, так как любой разработчик может с помощью одной команды загрузить миграцию другого разработчика. Миграции являются чем-то на подобие системы контроля версий для базы данных, так как в любой момент можно сделать откат миграции и вернуть старую версию.

Для удобного создания миграций используется встроенная во фреймворк командная строка Artisan. Продемонстрируем создание миграции для таблиц с помощью команды make:migration и заполним их данными.

**class** CreateCateroryTable **extends** Migration  
{  
 **public function** up()  
 {  
 Schema::**create**(**'Categories'**, **function** (Blueprint $table) {  
 $table->increments(**'id'**);  
 $table->string(**'category'**);  
 $table->timestamps();  
 });  
 }  
   
 **public function** down() {Schema::*dropIfExists*(**'Categories'**); }

}

**class** CreateLanguageTable **extends** Migration  
{**public function** up()  
 {  
 Schema::**create**(**'Languages'**, **function** (Blueprint $table) {  
 $table->increments(**'id'**); $table->string(**'language'**); $table->timestamps();  
 });  
 }  
 **public function** down()  
 { Schema::*dropIfExists*(**'Languages'**); }  
}

**class** CreateAdventTable **extends** Migration  
{**public function** up()  
 {  
 Schema::*create*(**'Advents'**, **function** (Blueprint $table) {  
 $table->increments(**'id'**);  
 $table->integer(**'type\_category'**)->unsigned()->index();  
 $table->integer(**'language'**)->unsigned()->index();  
 $table->integer(**'language\_translation'**)->unsigned()->index();  
 $table->integer(**'price'**)->nullable();  
 $table->integer(**'complexity'**)->nullable();  
 $table->date(**'date\_start'**);  
 $table->date(**'date\_finish'**);  
 $table->text(**'ad'**)->nullable();  
 $table->text(**'great\_announcement'**)->nullable();  
 $table->string(**'img'**,50)->nullable();  
 $table->integer(**'user'**)->unsigned()->index();  
 $table->string(**'link'**,255)->nullable();  
 $table->integer(**'status'**)->nullable();  
 $table->boolean(**'extra'**)->nullable();  
 $table->timestamps();  
 $table->foreign(**'type\_category'**)  
 ->references(**'id'**)->on(**'Categories'**);  
 $table->foreign(**'language'**)  
 ->references(**'id'**)->on(**'Languages'**);  
 $table->foreign(**'language\_translation'**)  
 ->references(**'id'**)->on(**'Languages'**);  
 $table->foreign(**'user'**)  
 ->references(**'id'**)->on(**'Users'**);  
 });  
 }**public function** down()  
 {Schema::*dropIfExists*(**'Advents'**); }  
}

**class** CreateSubjectTable **extends** Migration  
{**public function** up()  
 {  
 Schema::*create*(**'Subjects'**, **function** (Blueprint $table) {  
 $table->increments(**'id'**);  
 $table->string(**'subject\_name'**);  
 $table->timestamps();  
 });  
 }

**public function** down()  
 {Schema::*dropIfExists*(**'subjects'**); }  
}

**class** CreateFilesTable **extends** Migration  
{  
**public function** up()  
 {  
 Schema::*create*(**'files'**, **function** (Blueprint $table) {  
 $table->increments(**'id'**);  
 $table->integer(**'app'**)->unsigned()->index();;  
 $table->string(**'file'**);  
 $table->timestamps();  
  
 $table->foreign(**'app'**)  
 ->references(**'id'**)->on(**'Advents'**);  
 });  
 }  
**public function** down()  
 {Schema::*dropIfExists*(**'files'**); }  
}

**class** CreateForumMes **extends** Migration  
{  
 **public function** up()  
 {  
 Schema::*create*(**'forum\_messages'**, **function** (Blueprint $table) {  
 $table->increments(**'id'**);  
 $table->integer(**'subject'**)->unsigned()->index();  
 $table->string(**'message'**)->nullable();  
 $table->integer(**'user'**)->unsigned()->index();;  
 $table->timestamps();  
  
 $table->foreign(**'subject'**)  
 ->references(**'id'**)->on(**'Subjects'**)  
 ->onDelete(**'cascade'**)  
 ->onUpdate(**'cascade'**);

$table->foreign(**'user'**)  
 ->references(**'id'**)->on(**'Users'**)  
 ->onDelete(**'cascade'**)  
 ->onUpdate(**'cascade'**);  
 });  
 }**public function** down()  
 {Schema::*dropIfExists*(**'forum\_messages'**); }  
}

**class** PhotoForum **extends** Migration  
{

**public function** up()  
 {  
 Schema::*create*(**'Photo\_for\_forums'**, **function** (Blueprint $table) {  
 $table->increments(**'id'**);  
 $table->string(**'img'**);  
 $table->integer(**'message'**)->unsigned()->index();  
 $table->timestamps();  
  
 $table->foreign(**'message'**)  
 ->references(**'id'**)->on(**'Forum\_messages'**)  
 ->onDelete(**'cascade'**)  
 ->onUpdate(**'cascade'**);  
 });  
 }  
**public function** down()  
 {Schema::*dropIfExists*(**'Photo\_for\_forums'**); }  
}

Посмотрим на результат с помощью phpmyadmin.

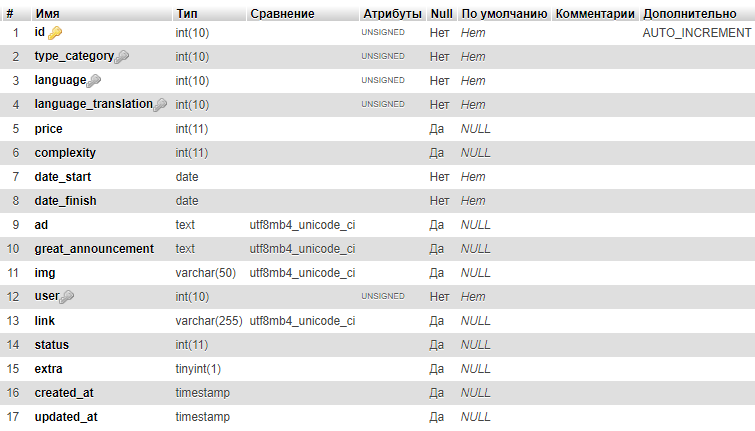


Рис. 4.1 – Таблица Advents

На рисунке 4.1 изображена структура таблицы Advents с объявлениями.



Рис. 4.2 – Таблица Categories

На рисунке 4.2 изображена структура таблицы Categories с категориями.



Рис. 4.3 – Таблица Files

На рисунке 4.3 изображена структура таблицы Files с файлами.



Рис. 4.4 – Таблица Languages

На рисунке 4.4 изображена структура таблицы Languages с языками.



Рис. 4.5 – Таблица Forum\_messages

На рисунке 4.5 изображена структура таблицы Forum\_messages с сообщениями форума.



Рис. 4.6 – Таблица Photo\_for\_forums

На рисунке 4.6 изображена структура таблицы Photo\_for\_forums с изображениями форума.

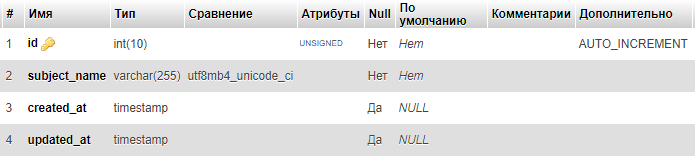


Рис. 4.7 – Таблица Subjects

На рисунке 4.7 изображена структура таблицы Subjectсs темами форума.

Laravel автоматически создает автообновляемые поля created\_ad и updated\_ad, для просмотра времени создания и последнего изменения записи.

Для дальнейшей работы нужно создать модели Advent, Category, File, Language, Subject, Forum\_message, Photo\_for\_forum.При создании модели, ее имя автоматически создает имена такие же как и имена таблиц, только в единственном экземпляре. Продемонстрируем наполнение модели на примере модели Advent.

**class** Advent **extends** Mode

l{ **public $timestamps** = **true**;  
 **protected $fillable** = [

**'type\_category'**,

**'language'**,

**'language\_translation'**,

**'price'**, **'complexity'**,

**'date\_start'**,

**'date\_finish'**,

**'great\_announcement'**,

**'ad'**, **'img'**, **'link'**];

}

В переменную $fillable записываем открытые поля, в которые пользователи будут добавлять записи.

# **Реализация клиентской части**

Внешний вид сайта создается с помощью представлений фреймворка laravel. В состав работы с представлениями входит шаблонизатор blade, позволяющий наследовать шаблоны. Так, например, создается шаблон шапки сайта, который можно использовать для всех страниц, без необходимости повторения кода. Также имеется возможность вывода переменных, написания условий и циклов.

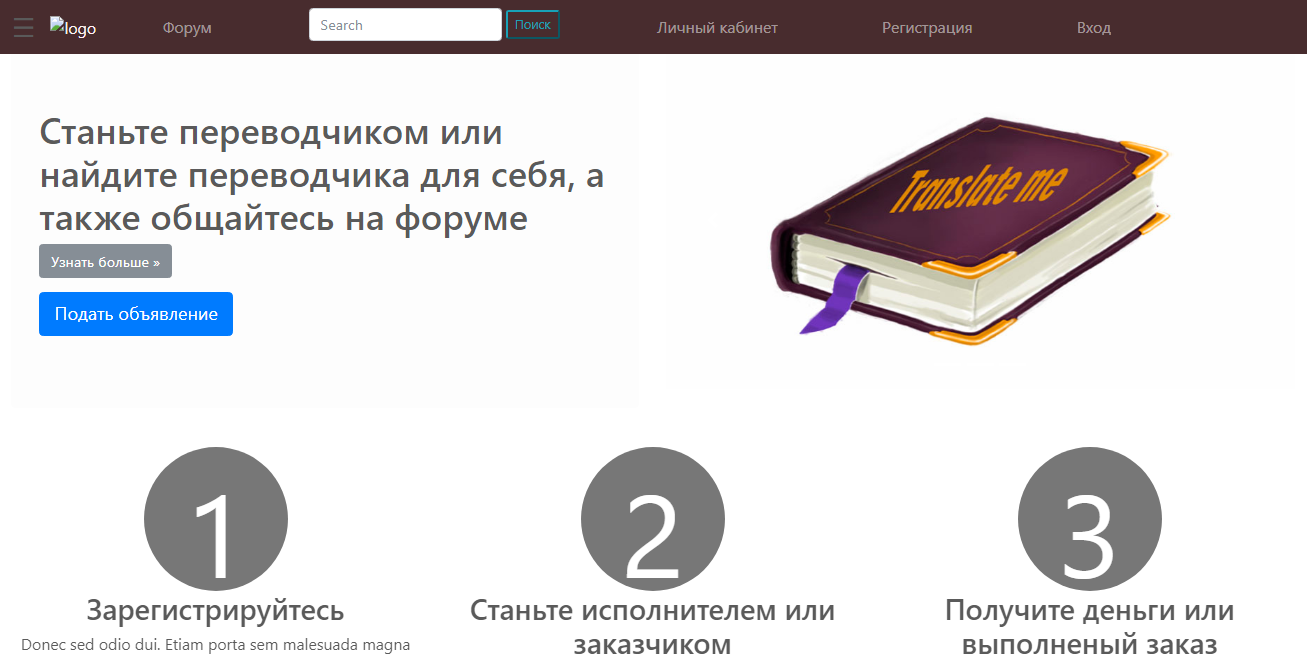


Рис. 5.1– Главная страница

На рисунке 5.1 продемонстрирована главная страница сайта.

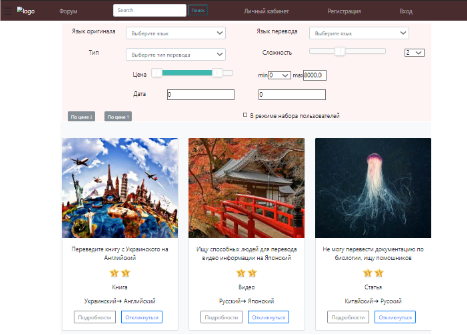


Рис. 5.2– Страница с объявлениями

На рисунке 5.2 продемонстрирована страница с объявлениями. Рассмотрим функцию обработки клиентской часть данной фильтра (filter.js).

**function** *changeCat*(){  
 **if** ( *$*(**'#languages'**).**length** ){ **var** lang =*$*(**'#languages'**).val();}  
 **if** ( *$*(**'#language\_translate'**).**length** ){ **var** lang\_tran =*$*(**'#language\_translate'**).val();}  
 **var** priceMin =*$*(**'#input-select'**).val(); **var** priceMax =*$*(**'#input-number'**).val();  
 **if** ( *$*(**'#categories'**).**length** ){ **var** cat =*$*(**'#categories'**).val();}  
 **var** complex =*$*(**'#input-complexity'**).val();  
 **if** ( *$*(**'#datepicker'**).**length** ) {**var** data1 = *$*(**"#datepicker"**).val();}  
 **if** ( *$*(**'#datepicker2'**).**length** ){**var** data2 =*$*(**'#datepicker2'**).val();}  
 ***$***.**ajax**({  
 **type**: **'get'**,  
 **url**: **''**,  
 **dataType**:**'html'**,  
 **data**: **"lang="**+lang +**"& priceMin="**+priceMin +**"& lang\_tran="**+lang\_tran+**"& priceMax="**+priceMax+**"& cat="**+cat+**"& complex="**+complex+**"& data1="**+data1+**"& data2="**+data2,  
success: **function** (response) { ***console***.log(response);  
 *$*(**'#updateDiv'**).html(response); }, });}

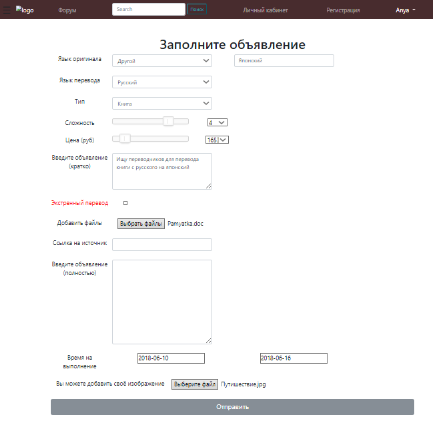


Рис. 5.3– Страница с добавлением объявлений

На рисунке 5.3 продемонстрирована страница с добавлением объявления. Страница с изменением объявления аналогична, только получает старые данные.

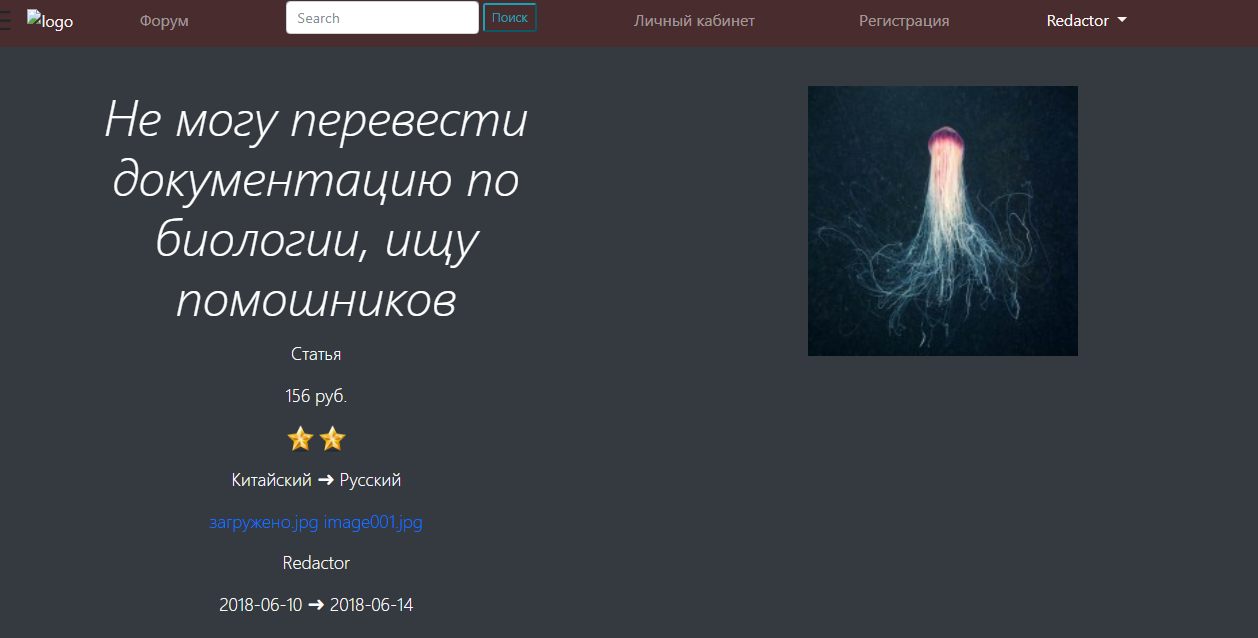


Рис. 5.4– Страница с подробным описанием объявления

На рисунке 5.4 продемонстрирована страница с подробным описанием объявления.

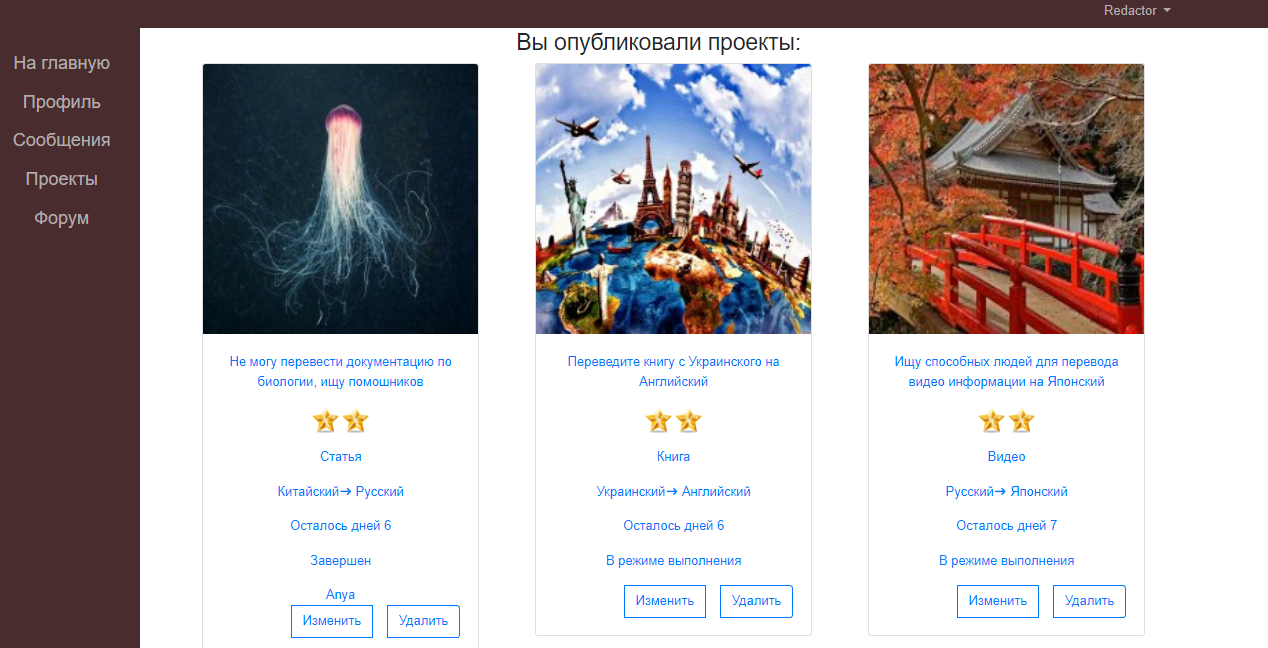


Рис. 5.5– Страница с проектами пользователя

На рисунке 5.5 показана страница с проектами, которые выложил пользователь. На этой же странице ниже располагаются проекты, которые пользователь взял на исполнение, выполнены они аналогично.

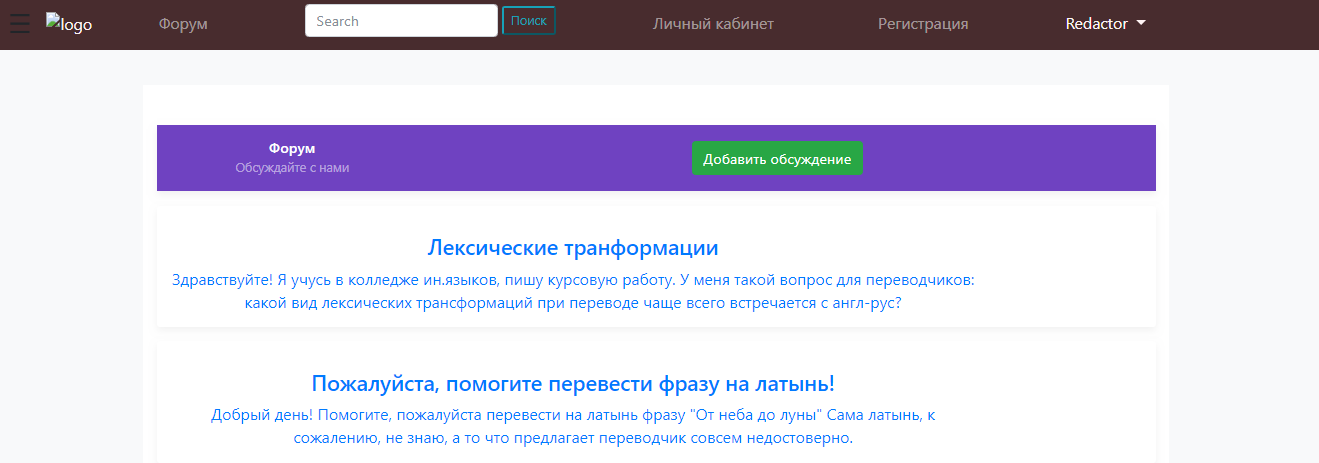


Рис. 5.6– Страница с темами форума

На рисунке 5.6 показана страница с темами форума.

JS обработчик добавления и изменения темы форума (forum.js):

*$*( **"#creating\_sub"** ).click(**function**() {  
 ***document***.getElementById(**'change\_for\_create'**).style.display=**'inline'**;  
 ***$***.**ajax**({  
 **type**: **'POST'**,  
 **url**: **'/forum\_create'**,  
 **dataType**:**'html'**,  
 success: **function** (response) {***console***.log(response);  
 *$*(**'#creating\_sub'**).html(response); }, });  
});  
**function** *Update\_subject*($id) {  
 ***document***.getElementById(**'change\_for\_update'**+$id).style.display=**'inline'**;  
 ***document***.getElementById(**'close\_button\_sub'**+$id).style.display=**'inline'**;  
 ***document***.getElementById(**'updating\_sub'**+$id).style.display=**'none'**;  
 ***$***.**ajax**({  
 **type**: **'POST'**,  
 **url**: **'/forum\_update'**,  
 **dataType**:**'html'**,  
 success: **function** (response) {***console***.log(response);  
 *$*(**'#updating\_sub'**+$id).html(response);  
 },});  
};

**function** *Close\_button\_subject*($id) {  
 ***document***.getElementById(**'change\_for\_update'**+$id).style.display=**'none'**;  
 ***document***.getElementById(**'updating\_sub'**+$id).style.display=**'inline'**;  
 ***document***.getElementById(**'close\_button\_sub'**+$id).style.display=**'none'**;  
};

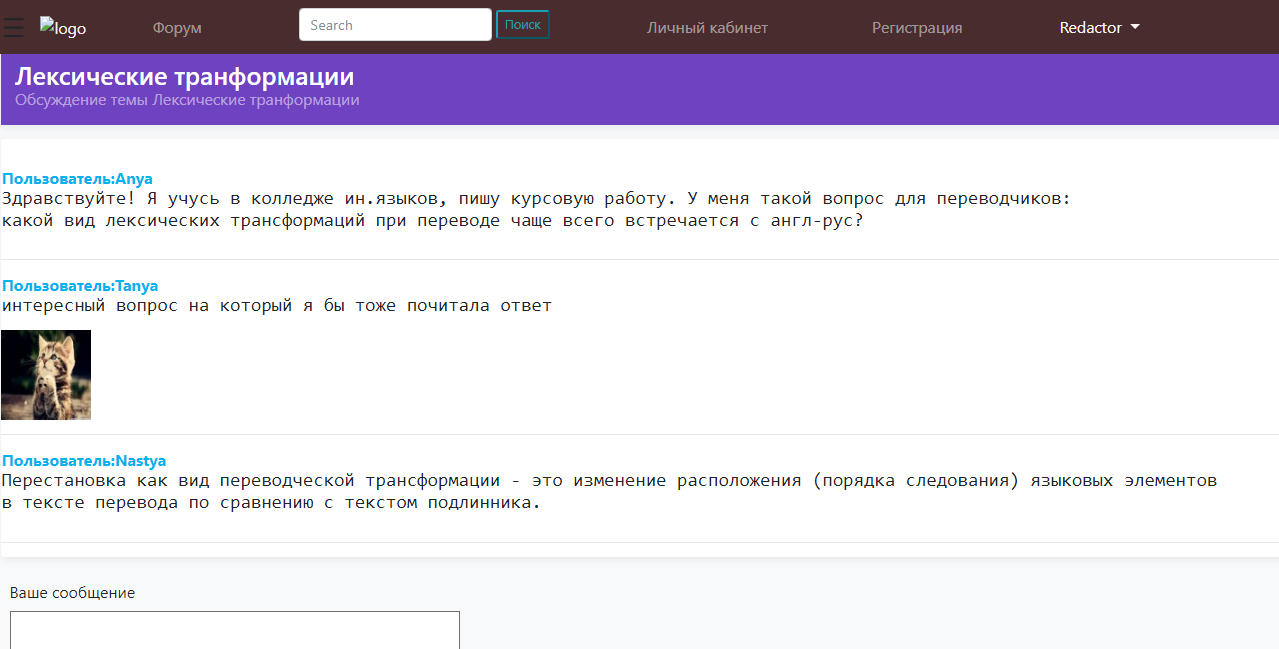


Рис. 5.7 – Страница с сообщениями форума

На рисунке 5.7 показана страница сообщений форума определенной темы. JS обработка добавления и изменения сообщения происходит аналогично JS обработки добавления и изменения темы.

Продемонстрируем фрагмент файла forumOpen.blade.php:

<**div class="container-fluid"**>  
 <**div class="d-flex align-items-center p-3 my-3 text-white-50 bg-purple box-shadow"**>  
 <**div**>  
 <**h3 class="mb-0 text-white lh-100"**>{{$subject}}</**h3**>  
 <**div**>Обсуждение темы {{$subject}}</**div**>  
 </**div**>  
 </**div**>  
 <**div class="my-3 p-3 bg-white rounded box-shadow"**>  
 **@foreach (**$forums **as** $Message**)** <**form method="POST" action="../updating\_massage/**{{$Message->**id**}}**"**>  
 <**div class="row media text-muted pt-3"**>  
  
 <**div class="media-body pb-3 mb-0 big lh-125 border-bottom border-gray"**>  
 <**strong class="d-block" style="color**: **#00b3ee**;**"**> Пользователь:{{$Message->**user**}}</**strong**>  
 <**h4** > <**pre**>{{$Message->**message**}} </**pre**> </**h4**>  
  
 **@foreach (**$img **as** $imges**) @if(**($Message->**id**) == ($imges->**message**) **)** <**img src="../upload/**{{$imges->**img**}}**"** >**@endif  
 @endforeach**</**div**>  
 **@isset (**Auth::*user*()->**id )  
 @if(**($Message->**user\_id**) === ( Auth::*user*()->**id**)**)** <**a href="../del\_message\_for\_forum/**{{$Message->**id**}}**" class="btn" id="delete\_mes" style=" color**:**black**; **border**: **none**; **" role="button"**>Удалить</**a**>

<**div class="col-xl-2 col-lg-2 col-md-2 col-sm-2 col-xs-2 resp"** ><**input type="button" class="btn secondary" onclick="***Update\_message*({{$Message->**id**}})**" id="updating\_mes**{{$Message->**id**}}**" value="Изменить"**>

<**input type="button" class="btn secondary" onclick="***Close\_button\_message*({{$Message->**id**}})**" id="close\_button\_mes**{{$Message->**id**}}**" style="display**: **none**;**" value="Закрыть"**> </**div**>  
<**div class="col-xl-12 col-lg-12 col-md-12 col-sm-12 col-xs-12 box-shadow" id="change\_for\_update\_mes**{{$Message->**id**}}**"**

**style="display**: **none**; **background**: **radial-gradient**(**white**,**#f4f6f9**);**"**>  
 **@include ('forum.update\_forum\_mes')** </**div**>**@endif@endisset** </**div**></**form**>**@endForeach**{{ $forums->links()}}  
 </**div**>  
 **@include ('forum.create\_forum\_message')**

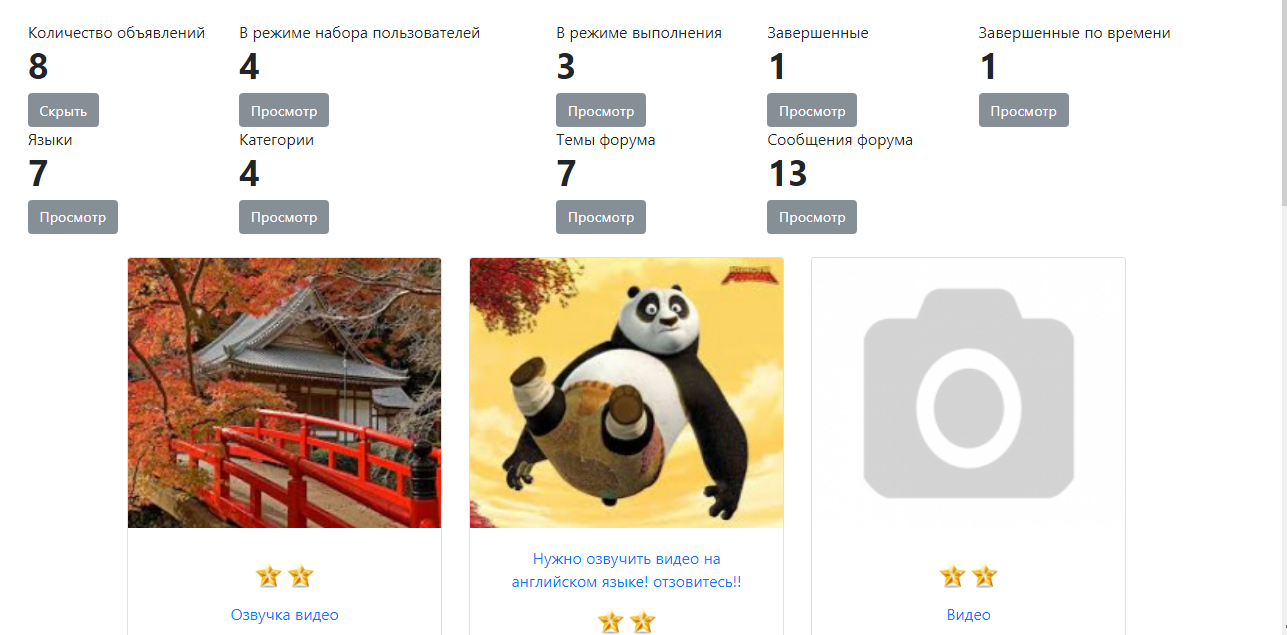


Рис. 5.8 – Страница с панелью управления

На рисунке 5.8 изображена панель управления для редактора

# **Реализация серверной части**

Логическая часть серверной части представлена с помощью контроллеров.

Рассмотрим некоторые функции контроллера CategoryController.php.

Функция обрабатывающая данные для фильтра:

**public function** category(Request $request) {  
 $categories = Category::*get*();$languages = Language::*get*();  
 **if**($request->ajax() ){  
 $lang= $request->**lang**; $lang\_tran= $request->**lang\_tran**;  
 $cat = $request->**cat**; $priceMin = $request->**priceMin**;  
 $priceMax = $request->**priceMax**; $complex = $request->**complex**;  
 $data1 = $request->**data1**; $data2 = $request->**data2**;  
 $Data= $this->data()  
 ->where(**'Advents.price'**, **'>='**,$priceMin)  
 ->where(**'Advents.price'**, **'<='**,$priceMax)  
 ->where(**'Advents.complexity'**, **'='**,$complex)  
 ->when($lang, **function** ($language) **use** ($lang){  
 **return** $language ->where(**'one.id'**, $lang); })  
 ->when($lang\_tran, **function** ($language) **use** ($lang\_tran){  
 **return** $language ->where(**'two.id'**, $lang\_tran); })  
 ->when($cat, **function** ($category) **use** ($cat){  
 **return** $category ->where(**'categories.id'**, $cat); })  
 ->when($data1, **function** ($datepicker1) **use** ($data1) {  
 **return** $datepicker1->where(**'Advents.date\_start'**, **'>='**, $data1); })  
 ->when($data2, **function** ($datepicker2) **use** ($data2){  
 **return** $datepicker2 ->where(**'Advents.date\_finish'**, **'<='**,$data2); })  
 ->select(**'Advents.id as id'**, **'Advents.img as img'**, **'Advents.ad as ad'**, **'Advents.complexity as complexity'**,  
 **'one.language as language'**, **'two.language as translation'**, **'categories.category as category'**, **'Advents.date\_start as date\_start'**,  
 **'Advents.date\_finish as date\_finish'**, **'users.name as user'**)->get();  
 response()->json($Data);  
 **return** view(**'categorys.products'**, [**'languages'** => $languages, **'categories'** => $categories, **'Data'** => $Data]);  
 }**else** {  
 $Data= $this->data\_select()->get();  
 **return** view(**'category'**, [**'languages'** => $languages, **'categories'** => $categories, **'Data'** => $Data]); }}

Функция добавляющая в таблицу новое объявление:

**public function** store(Request $request)  
{  
 $Advent = **new** Advent; $category = **new** Category; $language = **new** Language;  
 **if** (request(**'language2'**) != **NULL**) {  
 Language::*insert*([**'language'** => request(**'language2'**)]);  
 $Language = Language:: *where*(**'language'**, request(**'language2'**))->value(**'id'**);  
 } **else** {  
 $Language = Language:: *where*(**'language'**, request(**'language'**))->value(**'id'**) }  
 **if** (request(**'language\_translation2'**) != **NULL**) {  
 Language::*insert*([**'language'** => request(**'language\_translation2'**)]);  
 $language\_translation = Language:: *where*(**'language'**, request(**'language\_translation2'**))->value(**'id'**);} **else** {  
 $language\_translation = Language:: *where*(**'language'**, request(**'language\_translation'**))->value(**'id'**);}  
 **if** (request(**'type\_category2'**) != **NULL**) {  
 Category::*insert*([**'category'** => request(**'type\_category2'**)]);  
 $type\_category = Category:: *where*(**'category'**, request(**'type\_category2'**))->value(**'id'**);  
 } **else** {  
 $type\_category = Category:: *where*(**'category'**, request(**'type\_category'**))->value(**'id'**); }  
 **if** (request(**'extra'**) == **true**){  
 $extra=**true**; }  
 **else** $extra=**false**; $filename =**'l62egxS0UcVpRlWWZlOO.png'**;  
 **if** (request(**'img'**) != **NULL**) {  
 $path = public\_path() . **'\upload'**; $image = $request->file(**'img'**);  
 **foreach** ($image **as** $f) {  
 $filename = str\_random(20) . **'.'** . $f->getClientOriginalExtension() ?: **'png'** || **'jpg'**;  
 $img = ImageInt::*make*($f);  
 $img->resize(200, 200)->save($path . **'/'** . $filename);}}  
 Advent::*insert*([  
 **'language'** => $Language, **'language\_translation'** => $language\_translation,  
 **'type\_category'** => $type\_category, **'price'** => request(**'price'**),  
 **'complexity'** => request(**'complexity'**), **'ad'** => request(**'add'**),  
 **'great\_announcement'** => request(**'great\_announcement'**),  
 **'date\_start'** => request(**'dateStart'**), **'date\_finish'** => request(**'dateFinish'**),  
 **'user'** =>request(**'user'**), **'img'** =>$filename,  
 **'extra'** =>$extra, **'link'** => request(**'link'**),  
 **'created\_at'** => Carbon::*now*()->toDateTimeString()]);  
$id\_now=Advent:: *where*(**'user'**,request(**'user'**)) ->orderby(**'created\_at'**, **'desc'**) ->value(**'id'**);

**foreach** ($request->file() **as** $file) {**foreach** ($file **as** $f) {  
$f->move (public\_path(**'storage'**) , *time*() . **'/'** . $f->getClientOriginalName());$name=$f->getClientOriginalName();  
 File::*insertGetId*([**'app'**=>$id\_now, **'file'**=> $name]);}}**return** redirect( request(**'user'**) .**'/project'**);  
}

Рассмотрим некоторые функции контроллера ForumController.php.

Функция отображения сообщений для темы форума:

**public function** forum\_open($id){  
 $subject = subject::*where* (**'id'**, $id)->value(**'subject\_name'**);  
 $subject\_id = subject::*where* (**'id'**, $id)->value(**'id'**);  
 $forums= forum\_message:: *leftJoin*(**'subjects'**, **'forum\_messages.subject'**, **'='**, **'subjects.id'**) ->leftJoin(**'users'**, **'users.id'**, **'='**, **'forum\_messages.user'**)  
 ->where (**'subjects.id'**, $id)  
 ->select (**'users.name as user'**, **'users.id as user\_id'**, **'forum\_messages.message as message'**, **'forum\_messages.id as id'**)  
 ->orderby(**'forum\_messages.id'**) ->paginate(10);  
 $img = Photo\_for\_forum::*all*();  
 **return** view(**'forum.forumOpen'**, [**'subject'** => $subject, **'forums'**=> $forums,**'subject\_id'** => $subject\_id  
 , **'img'**=>$img] );  
}

Функция добавления нового сообщения:

**public function** store\_mes(Request $request){  
 forum\_message::*insert*([  
 **'subject'** => request(**'subj\_id'**), **'message'** =>request(**'message\_for\_forum'**),  
 **'user'** =>request(**'user'**)]);  
 **if** (request(**'file'**) != **NULL**) {  
 $path = public\_path() . **'\upload'**; $file = $request->file(**'file'**);  
 **foreach** ($file **as** $f) {  
 $filename = str\_random(20) . **'.'** . $f->getClientOriginalExtension() ?: **'png'** || **'jpg'**; $img = ImageInt::*make*($f); $img->resize(100, 100)->save($path . **'/'** . $filename);   
 $mes = forum\_message:: *where*(**'message'**, request(**'message\_for\_forum'**))->value(**'id'**);  
 Photo\_for\_forum::*insert*([**'img'** => $filename, **'message'** => $mes]);  
 } }  
 **return** back();  
}

# **Безопасность и экологичность проекта**

## **Анализ опасных и вредных факторов разработчика web-портала**

Для анализа опасных и вредных факторов разработчика будем использовать некоторые пункты классификации, представленной в ГОСТ 12.0.003-74 /1 БЖД /.

## **Физические опасные и вредные производственные факторы**

* + 1. ****Запыленность и загазованность воздуха****

Количество газа и пыли может превышать норму, так как помещение находится в центре города, где имеется сильный поток проезжающих мимо машин. Превышенная загазованность и запыленность пагубно сказывается на здоровье разработчиков, а также на работоспособности вычислительной техники.

Для того чтобы количество газа и пыли было не превышено, необходимо совершать влажную уборку два раза в день.

* + 1. ****Температура воздуха****

В зависимости от времени года температура меняется и может не соответствовать норме. В помещении должен находиться термометр для контроля температуры. Норма температуры воздуха внутри здания около 23-25°С при восьмичасовом рабочем дне. Нарушение нормы может негативно сказаться как на работниках, так и на технике, которая может перегреться вследствие превышенной температуры.

Для нормализации температуры необходимо следить, чтобы помещение при необходимости отапливалось либо охлаждалось.

* + 1. ****Повышенный уровень шума****

Вследствие того, что помещение для разработки портала находится в центре города, уровень шума за окном высокий, особенно в дневное время. Такой шум может пагубно влиять на психологическое состояние.

В то время, когда шум за пределом здания превышен, необходимо закрыть окна. Но следует помнить, что помещение, в котором для работы используются компьютеры, нужно проветривать каждый час.

* + 1. ****Барометрическое давление в рабочей зоне****

Для измерения барометрического давления в помещении находится барометр. Резких скачков или нарушения нормы не обнаружено.

* + 1. ****Подвижность воздуха****

При регулярном проветривании (раз в час) циркуляция воздуха проходит в нормальном режиме.

* + 1. ****Повышенное значение напряжения в электрической цепи****

Воздействие электрического тока на человека крайне опасно, так как ток мгновенно поражает части тела, а также органы человека.

Для безопасности необходимо ограничить величину применяемого напряжения до 220 В, изолировать токоведущие части и соблюдать технику безопасности.

* + 1. ****Уровень статического электричества****

Компьютеры являются источниками статического электричества. Оно пагубно влияет на работу техники, а также на человеческий организм.

На работу техники статическое электричество влияет путем попадания пыли на запчасти, что постепенно приводит к неисправности.

При воздействии статического электричества человек может испытывать раздражительность, утомляемость, также известны случаи возникновения сердечных-сосудистых заболеваний вследствие воздействия статического электричества.

Необходимо проводить влажную уборку в помещении два раза в день, проветривать каждый час. Для дополнительной защиты можно поставить увлажнители воздуха, либо использовать специальные антистатические спреи.

* + 1. ****Уровень электромагнитных излучений****

Источниками электромагнитных полей рабочего помещения являются персональные компьютеры и ноутбуки, а также другая техника (принтеры и пр.).

Чтобы уровень излучения не был повышенным, нужно использовать жидкокристаллические мониторы, заземлить технику, убрать с рабочего места металлические предметы. Площадь рабочего места должна быть не менее 4,5 м2.

* + 1. ****Напряженность электрического поля****

Уровень напряженности должен быть примерно 0.5 кВ/м. Помещение для разработки портала соответствует этому требованию.

* + 1. ****Естественный свет****

Наличие окон – обязательное условие для помещения с рабочими местами.

Естественный свет зависит от климатических факторов за пределами здания. При пасмурной погоде или в вечерние часы естественного света может не хватать.

Слабое освещение является причиной усталости и неработоспособности людей, работающих в помещении, вызывает головные боли, а также может пагубно повлиять на зрение, вызывать болезни глаз.

Если естественного света не хватает, необходимо включать больше искусственного света.

* + 1. ****Освещенность рабочей зоны****

Необходимо обустроить оптимальное освещение для здоровья и комфорта. Для того чтобы работоспособность была максимально продуктивной, свет на рабочем месте должен быть с температурой цвета 3500-4000 К, а в комнатах отдыха можно выбирать более теплый свет.

* + 1. ****Яркость света****

Для того чтобы избежать очень яркого или очень темного света, необходимо иметь два уровня освещения.

Первый уровень – общее (мягкий свет без бликов).

Второй уровень – местное (примой свет, для освещения рабочего места). Такой свет устанавливается слева от человека, если он правша, и справа, если он левша.

* + 1. ****Контрастность света****

Высокая контрастность света приводит к ухудшению самочувствия и психического восприятия, низкая ­– к состоянию утомления. Поэтому необходимо, чтобы контрастность света была средней.

* + 1. ****Блесткость****

Для нормализации блесткости необходимо правильно подбирать источники освещения, а также правильно устанавливать рабочие места по отношению к освещению.

Поверхность стола должна быть матовой.

* + 1. ****Пульсация светового потока****

При повышенной пульсации человек чувствует утомляемость, усталость глаз, нарушение пищеварительной и сердечно-сосудистой системы.

Для нормализации светового потока нужно подбирать хорошие источники света и следить за освещением в помещении.

* + 1. ****Уровень ультрафиолетовой радиации****

В помещении ультрафиолет могут излучать компьютеры, а также ртутно-кварцевые, люминесцентные, озоновые и кварцевые лампы.

Необходимо выбирать светодиодные лампы или лампы накаливания.

Для людей, работающих на компьютере, нужно делать перерывы каждый час на 10-15 минут, и время работы должно длиться не более восьми часов.

* + 1. ****Уровень инфракрасной радиации****

Инфракрасное излучение возможно двух видов: коротковолновое и длинноволновое. Опасность представляет именно первое. При его воздействии человек чувствует головокружение, тошноту.

Источниками инфракрасного излучения являются лучи солнечного света, лампы накаливания, газоразрядные лампы.

Для безопасности людей на окнах должны висеть жалюзи и занавески, чтобы оградить людей от интенсивного солнечного света и избежать теплового удара.

* + 1. ****Расположение рабочих мест****

На одно рабочее место выделяется не меньше 6 м2 пола и более 20 м2 пространства вокруг места. Ширина рабочего стола более 60 см, а длина 1,2 м. Высота – от 600 до 800 мм. Ширина каждого сиденья должна быть более 40 см. Стул должен быть с подлокотниками, регулироваться по высоте и по наклону спинки.

Для удобства можно расположить рельефную подставку для ног, которую можно поднять и опустить для своего удобства.

## **Химические опасные и вредные производственные факторы**

* + 1. ****Канцерогенные и мутагенные****

В здании могут обнаруживаться пары формальдегида, которые образуются из-за некоторых отделочных материалов и мебели.

Воздух может быть загрязнен из-за внешней загазованности, поэтому необходимо проветривать помещение каждый час.

## **Биологические опасные и вредные производственные факторы**

В помещении из внешней и внутренней среды могут появиться грибы, вирусы и бактерии. Их воздействие может негативно сказаться на здоровье. Могут появиться дерматологические, желудочно-кишечные, легочные заболевания.

Для профилактики необходимо проветривать помещение каждый час, проводить влажную уборку два раза в день и генеральную уборку минимум раз в неделю. Важно не забывать протирать поверхности компьютеров.

Необходимо время от времени мыть руки, при возникновении вирусных респираторных заболеваний надо носить ватно-марлевую повязку и при необходимости обращаться к врачу.

## **Расчет системы местного освещения**

Работоспособность людей во многом зависит от выбора систем освещения. Определим поток света и узнаем, какие лампы нам подойдут для местного освещения.

Параметры для расчета возьмём из таблиц и графиков учебного пособия «Техносферная безопасность. Расчеты» В.Л. Гапонов, В.В. Киреева, В.И. Гаршин и др.; под ред. проф. В.Л. Гапонова. – Ростов н/Д: Издательский центр ДГТУ, 2012. – 131 с. Глава 1, пункт 1.7.

Дано: Коэффициент запаса мощности K = 1,5; нормированная освещенность Е = 150 лк; высота установки лампы h = 30 см; расстояние от проекции оси лампы а = 19 см.

Решение: По графику получаем, что условная освещенность å = 400.

Световой поток определяется по формуле: .

Подставим наши значения в формулу лм.

По значению светового потока выбираем из таблиц подходящий светильник.

Ответ: Выбираем лампу накаливания Б–125–135–40, со световым потоком 485 лм.

## **Оценка экологической обстановки центра г. Ростова-на-Дону.**

Через центр города протекает река Дон, в которую сбрасываются отходы. Также через Дон проплывает множество кораблей. Воду из реки используют для водоснабжения людей и полива растений. Стоит заметить, что вода проходит через промышленную фильтрацию, поэтому вред экологии приносится небольшой. Но все же воду для домашнего использования стоит дополнительно фильтровать или кипятить.

Гораздо больше опасности приносят автомобили. Транспорт приносит огромное количество выбросов, что сильно портит экологическое состояние города. Известно, что в состав автомобильных выбросов входят оксиды азота, формальдегид, бензол, свинец, сернистый ангидрид, безопорен, сажа и другие твердые частицы. Это опасные вещества, которые вызывают воспаления дыхательной, кровеносной системы, могут стать причиной появления аллергии, развития рака.

Чтобы снизить воздействие вредных выхлопов на организм человека, надо использовать очистители воздуха в квартирах и офисах, чаще производить влажную уборку помещений, проводить больше времени на улице вдали от автомобильных дорог.

В некоторых местах в центре города, еще используются старые сети канализации и водоснабжения, которые работают некачественно и засоряют город.

В центре города располагаются заводы: «Ростовский завод железобетонных конструкций», «Тавр» и другие. Выбросы заводов наносят большой ущерб экологии города. В последние годы многие заводы переносят промышленную часть за город, что хорошо сказывается на экологии города.

Что касается застройки города, то строится все больше зданий, при этом вырубаются деревья.

Для того чтобы улучшить экологию Ростова-на-Дону, необходимо тщательней очищать воду перед ее попаданием в Дон и перед тем как снабжать водой людей и поливать растения. Гражданам в будущем стоит перейти с обычных автомобилей на экомобили. Нужно поменять старые канализационные системы на новые. Тогда в питьевой воде будет меньше примесей тяжелых металлов. Все заводы необходимо перенести за город, чтобы люди не дышали выхлопными газами.

Городские власти и граждане должны уделять внимание зеленым насаждениям. Деревья поглощают углекислый газ из воздуха и выделяют кислород, что значительно улучшает качество воздуха в черте города. Деревья нужно перестать вырубать, а наоборот начать высаживать парки, аллеи, лесополосы.

Следует сажать деревья, которые не вызывают у людей аллергию. В Ростове-на-Дону и, в частности, в центре города много тополей. Тополиный пух вызывает аллергическую реакцию у многих людей, поэтому тополь – это не лучшее дерево для высадки в черте города. Большую пользу для здоровья людей в Ростове-на-Дону приносят каштаны, липы, акации, многие виды хвойных растений.

Проблема раздельного сбора отходов остро стоит в Ростове-на-Дону. Собирать раздельно отходы, подлежащие вторичной переработке, это способ экономии природных ресурсов и сохранения окружающей среды.

Чтобы в Ростове была решена проблема раздельного сбора мусора, содействовать ее решению должны как отдельные граждане и предприятия города, так и администрация Ростова-на-Дону.

Для эффективного решения проблемы администрация должна устанавливать по всему городу пункты для раздельного сбора мусора, а жители города должны ответственно сортировать мусор и использовать контейнеры по назначения.

Кроме того, руководство города должно содействовать доставке всех видов отходов в пункты для утилизации и их дальнейшей переработке. Чтобы люди активно содействовали раздельному сбору отходов, они должны быть уверены, что мусор действительно поступает в переработку, а не сбрасывается на общую свалку.

Только при совместной деятельности граждан, руководства города, коммерческих организаций получится разгрузить городские свалки, снизить уровень загрязнения воды в реке Дон и воздуха в Ростове-на-Дону, снизить расход невосполняемых природных ресурсов в Ростовской области.

Для большего включения предприятий города в систему раздельного сбора отходов можно ввести систему поощрений и льгот, ка это делается в странах Европы.

Для мотивации граждан надо проводить городские акции, призывать к раздельному сбору отходов с рекламных плакатов и с экрана ТВ. А главное, контейнеры для раздельного сбора мусора должны быть в шаговой доступности от каждого жилого дома в городе.

Важно приучать детей к раздельному сбору отходов. Тогда в будущем для них этот процесс станет естественным и прочно войдет в жизнь.

## **Чрезвычайные ситуации**

Пользователем веб портала может понадобиться срочный перевод текстовой или другой информации. Им необходимо будет выложить объявление о переводе и указать временной интервал, в сроки которого необходимо перевести, и поставить галочку в пункте «экстренный перевод». Экстренные переводы будут отображаться первыми в списке объявлений.

Это может быть актуально во время чемпионата мира по футболу, так как в Россию приедет множество людей из разных стран, и им может понадобиться перевод чего-либо.

Фанаты футбола могут попросить перевести или озвучить матч. Также они могут дать задание перевести какую-либо информацию, связанную с футболом, такую как: разборы матчей, интервью с игроками, локальные матчи с понравившимися командами.

В случае возникновения в городе чрезвычайной ситуации (пожара, наводнения или иной катастрофы) пользователь может попросить перевести его информацию о том, где произошла катастрофа, для того чтобы люди, говорящие на другом языке, узнали об этом. Или человек может дать запрос на перевод чьей-то информации о происшествии.

Возможно, людям, которые находятся в другой стране, например, в России, во время чемпионата мира по футболу, может понадобиться перевести какую-либо личную информацию. Для своей безопасности пользователь может выбирать проверенных исполнителей (тех, которые выложили свою фотографию и уже брали работу на исполнение), также может пообщаться с ними и договориться о полной конфиденциальности. Но полную безопасность мы гарантировать не можем, так что просим быть бдительными и, желательно, не выкладывать слишком личную информацию или информацию, содержащую государственную тайну.

# **ЗАКЛЮЧЕНИЕ**

В результате данной дипломной работы реализован функционал для обеспечения контента для web-портала сообщества переводчиков «Translator’s house».

Есть возможность для каждого пользователя добавить новое объявление о переводе, изменить свое уже добавленное объявление, удалить его.

Для поиска объявлений реализован фильтр, в котором можно выбрать язык оригинала, язык перевода, тип перевода, цену, сложность, интервал по времени. Имеется возможность посмотреть детали объявления.

Создана панель редактора для просмотра количества всех объявлений, тех, что в режиме набора пользователей, завершенных и завершенных по времени.

Для каждого пользователя возможен просмотра объявлений, которые он выложил или взял на исполнение.

Реализован форум для общения пользователей. Имеется возможность добавления новых тем обсуждения и сообщений на уже существующие темы. Также каждый пользователь имеет возможность редактирования и удаления своих тем и сообщений.

Web-портал ориентирован на молодежь, удобен в использовании, имеет современный, приятный дизайн, адаптивную верстку.

Оценим созданный программный продукт в таблице 8.1

Таблица 8.1

|  |  |
| --- | --- |
| **Параметр сравнения** | **«**Translator’s house**»** |
| Добавление объявления | Отлично |
| Фильтрация объявлений | Отлично |
| Срочные переводы | Да |
| Просмотр выполненных заказов пользователей | Да |
| Форум | Да |
| Дизайн сайта | Хорошо |
| Адаптивность сайта | Отлично |

# **БИБЛИОГРАФИЧЕСКИЙ СПИСОК**

1. К. Дж. Дэйт. Введение в системы баз данных; Вильямс – Москва, 2017.

2. Томас Коннолли, Каролин Бегг. Базы данных. Проектирование, реализация и сопровождение. Теория и практика; Вильямс – Москва, 2017.

3. Дженнифер Нидерст Роббинс. Web-дизайн. Справочник; КУДИЦ-Пресс – Санкт-Петербург, 2008.

4. Алан Купер, Роберт М. Рейманн, Дэвид Кронин. Алан Купер об интерфейсе. Основы проектирования взаимодействия; Символ-Плюс – Санкт-Петербург, 2009.

5. Стоян Стефанов. JavaScript. Шаблоны; Символ-Плюс – Санкт-Петербург, 2011.

6. Дэвид Флэнаган. JavaScript. Подробное руководство, 6-е издание; Символ-Плюс – Санкт-Петербург, 2013.

7. Николас Закас. JavaScript. Оптимизация производительности; Символ-Плюс – Санкт-Петербург, 2012.

8. Джонатан Чаффер, Карл Шведберг. Изучаем jQuery 1.3. Эффективная веб-разработка на JavaScript; Символ-Плюс – Санкт-Петербург, 2010.

9. Бер Бибо, Иегуда Кац. jQuery. Подробное руководство по продвинутому JavaScript, 2-е издание; Символ-Плюс – Санкт-Петербург, 2011.

10. Кирсанов Д. Веб-дизайн; Символ-Плюс – Санкт-Петербург, 2001.

11. Гарретт Д. Веб-дизайн. Элементы опыта взаимодействия; Символ-Плюс – Санкт-Петербург, 2008.

12. Гутманс Э., Баккен С., Ретанс Д. PHP 5. Профессиональное программирование; Символ-Плюс – Санкт-Петербург, 2006.

13. Аргерих Л., Чой В., Коггсхол Д., Эгервари К., Сколло К. Профессиональное PHP программирование. 2-е издание; Символ-Плюс – Санкт-Петербург, 2003.

14. Закас Н., Мак-Пик Д., Фосетт Д. Ajax для профессионалов; Символ-Плюс – Санкт-Петербург, 2007.

15. Мейер Э. CSS - каскадные таблицы стилей. Подробное руководство, 3-е издание; Символ-Плюс – Санкт-Петербург, 2000.

16. Дэвис М., Филлипс Д. Изучаем PHP и MySQL, 2-е издание; Символ-Плюс – Санкт-Петербург, 2000.

17. Бьюли Алан. Изучаем SQL; Символ-Плюс – Санкт-Петербург, 2000.

18. Якоб Нильсен, Хоа Лоранжер. Web-дизайн. Удобство использования Web-сайтов; Вильямс – Москва, 2009.

19. Стив Макконнелл. Совершенный код. Мастер-класс; Русская Редакция, – Москва, 2000.

20. Владимир Дронов. Laravel. Быстрая разработка современных динамических Web-сайтов на PHP, MySQL, HTML и CSS; БХВ-Петербург – Санкт-Петербург, 2018.