

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное  
образовательное учреждение высшего образования  
«Южно-Уральский государственный университет  
(национальный исследовательский университет)»

Высшая школа электроники и компьютерных наук  
Кафедра «Электронные вычислительные машины»

РАБОТА ПРОВЕРЕНА

Рецензент

\_\_\_\_\_ 2019 г.  
« \_\_\_ » \_\_\_\_\_

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой

\_\_\_\_\_ Г.И. Радченко  
« \_\_\_ » \_\_\_\_\_ 2019 г.

**РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ ДЛЯ РАБОТЫ ПЕРЕВОДЧИКОВ  
С ИЗОБРАЖЕНИЯМИ**

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА  
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ**

Руководитель работы,  
к.п.н., доцент каф. «Электронные  
вычислительные машины»

\_\_\_\_\_ Ю.Г. Плаксина  
« \_\_\_ » \_\_\_\_\_ 2019 г.

Автор работы,  
студент группы КЭ-222

\_\_\_\_\_ М.А. Кицюк  
« \_\_\_ » \_\_\_\_\_ 2019 г.

Нормоконтролёр, ст. преп. каф.  
«Электронные вычислительные  
машины»

\_\_\_\_\_ С.В. Сяськов  
« \_\_\_ » \_\_\_\_\_ 2019 г.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное  
образовательное учреждение высшего образования  
«Южно-Уральский государственный университет  
(национальный исследовательский университет)»

Высшая школа электроники и компьютерных наук  
Кафедра «Электронные вычислительные машины»  
Направление «Информатика и вычислительная техника»

УТВЕРЖДАЮ  
Заведующий кафедрой  
\_\_\_\_\_ Г.И. Радченко  
« \_\_\_ » \_\_\_\_\_ 2019 г.

### **ЗАДАНИЕ**

**на выпускную квалификационную работу бакалавра**  
студента группы КЭ-452  
Кицюк Марии Анатольевне  
обучающейся по направлению  
09.03.01 «Информатика и вычислительная техника»

- 1. Тема работы:** «Разработка веб-приложения для работы переводчиков с изображениями» утверждена приказом по университету от 25 апреля 2019 г. №899
- 2. Срок сдачи студентом законченной работы:** 1 июня 2019 г.
- 3. Исходные данные к работе:** статьи, книги, техническое задание.
  - Никсон, Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5/ Р. Никсон – СПб.: Изд-во Питер, 2017. – 768 с
  - Бретт, М. PHP и MySQL. Исчерпывающее руководство / М. Бретт. – СПб.: Изд-во Питер, 2016. – 544 с.
  - Сафронов, М. Разработка веб-приложений в Yii 2 / М. Сафронов. – М.: Изд-во ДМК Пресс, 2015. – 392 с.

#### **4. Перечень подлежащих разработке вопросов:**

- Провести анализ существующих САТ систем осуществить постановку задачи
- Провести анализ функциональных и нефункциональных требований
- Спроектировать веб-сервис
- Провести тестирование веб-сервиса

**Дата выдачи задания:** 1 декабря 2018 г.

Руководитель работы \_\_\_\_\_ /Ю.Г. Плаксина/

Студент \_\_\_\_\_ /М.А. Кицюк /

## КАЛЕНДАРНЫЙ ПЛАН

Этап	Срок сдачи	Подпись руководителя
Введение и обзор литературы	01.03.2019	
Разработка модели, проектирование	01.04.2019	
Реализация системы	01.05.2019	
Тестирование, отладка, эксперименты	15.05.2019	
Компоновка текста работы и сдача на нормо- контроль	24.05.2019	
Подготовка презентации и доклада	30.05.2019	

Руководитель работы \_\_\_\_\_ /Ю.Г. Плаксина/

Студент \_\_\_\_\_ /М.А. Кицюк/

## Аннотация

М.А. Кицюк. Разработка веб-приложения для работы переводчиков с изображениями. – Челябинск: ФГАОУ ВО «ЮУрГУ (НИУ)», ВШЭиКН; 2019, 57 с., 16 ил. библиогр. список – 25 наим.

Выпускная квалификационная работа была выполнена с целью разработки веб-приложения для работы переводчиков с изображениями.

В данной работе были проанализированы существующие реализации аналогичных приложений, изучены методы разработки веб-приложений, выполнен анализ требований, на основе которых было осуществлено проектирование, разработка и тестирование веб-приложения.

Результатом выполненной работы является полностью функционирующее веб-приложение «Bearlog.org», которое в дальнейшем будет опубликовано в интернет.

# ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	7
1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ	9
1.1. АНАЛИЗ	9
1.2. ОБЗОР АНАЛОГИЧНЫХ ПРОЕКТОВ	9
1.3. ВЫВОД	15
2. ОПРЕДЕЛЕНИЕ ТРЕБОВАНИЙ	18
2.1. ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ	18
2.2. НЕФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ	19
3. ПРОЕКТИРОВАНИЕ	20
3.1. ДИАГРАММА ВАРИАНТОВ ИСПОЛЬЗОВАНИЯ	20
3.2. ПРОЕКТИРОВАНИЕ СТРУКТУРЫ БАЗЫ ДАННЫХ	21
3.3. ПРОЕКТИРОВАНИЕ СТРУКТУРЫ ГРАФИЧЕСКОГО ИНТЕРФЕЙСА	23
3.4. ПРОЕКТИРОВАНИЕ АРХИТЕКТУРЫ СИСТЕМЫ	25
4. РЕАЛИЗАЦИЯ	26
4.1. РЕАЛИЗАЦИЯ БАЗЫ ДАННЫХ	26
4.2. КОМПОНЕНТЫ СИСТЕМЫ	26
4.3. РЕАЛИЗАЦИЯ ИНТЕРФЕЙСОВ	28
5. ТЕСТИРОВАНИЕ	33
5.1. МЕТОДОЛОГИЯ ТЕСТИРОВАНИЯ	33
5.2. ПРОВЕДЕНИЕ ПРОЦЕДУРЫ ТЕСТИРОВАНИЯ	34
ЗАКЛЮЧЕНИЕ	37
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	38
ПРИЛОЖЕНИЕ А	41
ПРИЛОЖЕНИЕ Б	46
ПРИЛОЖЕНИЕ В	50
ПРИЛОЖЕНИЕ Г	55

## **ВВЕДЕНИЕ**

### **Глоссарий**

*Автоматический перевод* – процесс перевода компьютером текста с языка оригинала на целевой язык;

*Автоматизированный перевод* – вид перевода, в котором переводчик использует компьютерное оборудование для облегчения процесса работы;

*CAT* – аббревиатура с англ. Computer-Aided Translation см. автоматизированный перевод;

*Память переводов* (с англ. translation memory) – база данных, содержащая набор ранее переведенных сегментов текста;

*Клинер* – специалист очищающий отсканированные изображения от оригинального текста;

*Тайпер* – специалист, вписывающий переведенный текст на место оригинального в изображении;

### **Актуальность**

В настоящее время профессия переводчика является одной из наиболее востребованных в мире [2, 3, 4]. Рейтинг популярности данной специальности не падает, даже несмотря на активное внедрение в повседневную жизнь технологий машинного перевода [3, 4].

Рабочий процесс современного переводчика неразрывно связан с вспомогательными программными computer-aided translation-системами (далее CAT-системами) [1], значительно ускоряющими работу (в области письменного перевода).

В основе большинства современных CAT-систем лежат решения задачи распознавания текста, но уровень распознавания позволяет качественно автоматизировать обработку лишь машинописного текста, либо рукописного текста, приближенного к машинному. При таком подходе остается нерешенной проблема перевода графических материалов – изображений с рукопис-

ным текстом, комиксов и манги, в которых текст зачастую оформлен декоративными средствами, изогнут, либо вписан в само изображение.

Более того, при переводе графического материала, самим процессом перевода и дальнейшей постобработкой информации зачастую занимаются разные люди – переводчики и художники. К переводчикам относятся переводчики и редакторы, к художникам относятся клинеры и тайперы. Перед художниками, в чью работу входит удаление с картинки исходного текста и вписывание перевода, стоит проблема сопоставления информации, неоднородно разнесенной по изображению.

Таким образом, разработка платформы с инструментарием перевода изображений является актуальной задачей на сегодняшний день.

### **Цели исследования**

Целью представленной выпускной квалификационной работы является разработка веб-приложения с инструментарием перевода изображений.

### **Задачи исследования**

Для достижения поставленной цели, требуется решить следующий ряд задач:

- 1 Провести анализ аналогичных решений и на основе полученных выводов осуществить постановку задачи.
- 2 Провести анализ средств реализации аналогичных веб-сервисов для работы переводчиков.
- 3 Провести анализ требований и спроектировать архитектуру системы.
- 4 Реализовать систему в виде веб сервиса.
- 5 Провести тестирование разработанной системы.

### **Новизна исследования**

Новизна исследования состоит в разработке функционала для обработки (перевода) изображения на данный момент отсутствующего в аналогичном программном обеспечении.

# 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

## 1.1. АНАЛИЗ

### Актуальность использования памяти переводов

Популярность CAT в настоящее время объясняется тем, что ключевым фактором при работе с переводом является производительность. Высокая производительность достигается за счет ускорения выполнения однотипных операций, без потери качества, благодаря таким инструментам как память переводов и терминологические словари. Хотя исследования показывают, что использование памяти переводов может привести к увеличению от 10% до 70%, любое фактическое увеличение производительности зависит как от содержания ТМ, так и от типа текста. При наличии большого числа повторов, как, например, в руководствах пользователя или технических условиях, повышение производительности может составлять до 60%. Однако при переводе обычных текстов, таких как учебники, научные статьи, она составит 10-30%. Использование же памяти переводов для перевода писем может вообще не дать какой-либо экономии. Также если целевые представления, включенные в базу данных памяти переводов, демонстрируют достаточно свободные характеристики, это может даже отрицательно повлиять на производительность переводчика [15].

## 1.2. ОБЗОР АНАЛОГИЧНЫХ ПРОЕКТОВ

На сегодняшний день на рынке представлено несколько CAT систем, но рассматривать их все нецелесообразно по причине того, что большая часть систем устарела либо используется слишком малым кругом пользователей. Проведем анализ наиболее актуальных приложений [1].

**SDL Trados** [5] – CAT система, разработанная компанией SDL Trados и использующая модульную систему совместимых друг с другом программных компонентов. Используется более чем 250 000 переводчиками [9]. Имеет память переводов, хранящую данные в центре SDL Trados Studio и работающую за счет использования ранее переведенного контента. Мировой лидер

среди систем автоматизированного перевода. Интерфейс SDL Trados представлен на рисунке 1.

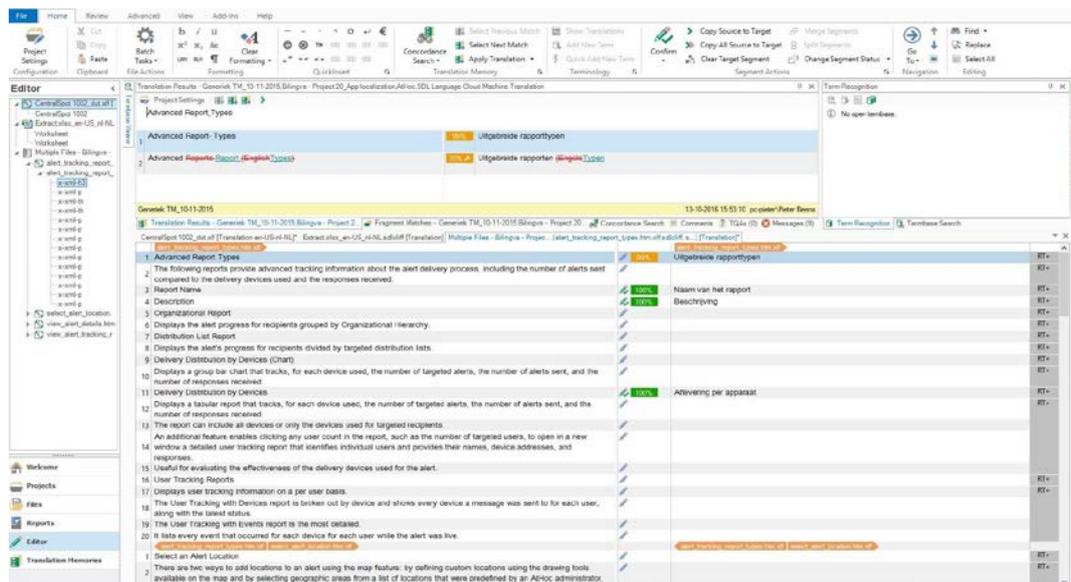


Рисунок 1 – Пример работы SDL Trados Studio

#### Достоинства:

- Поддержка 120 языков (наибольшее число среди аналогов);
- Самая объемная база памяти переводов среди аналогов;
- Встроенные механизмы машинного перевода;
- Встроенные инструменты оптического распознавания текста;
- Поддержка коллективной работы с единой памятью переводов;
- Облачные сервисы хранения переводов;
- Предоставляет возможность работы через браузер с ограниченным функционалом;
- Состоит из множества совместимых модулей (создания ТМ, глоссариев, автоматизированного создания глоссариев и тд.), комбинировать которые можно исходя из требуемого функционала;
- Встроенная проверка орфографии.

#### Недостатки:

- Проприетарное программное обеспечение (цена самого дешевого пакета на момент января 2018г. 695 евро [10]);

- Отсутствие функционала работы с изображениями и видео;
- Реализован в виде приложения на ОС Windows (отсутствует полноценная кроссплатформенность).

**Déjà Vu X3** [6] – CAT система, разработанная компанией Atril. Своим ключевым преимуществом перед конкурентами выделяет простой и интуитивно понятный интерфейс, и использование технологии DeepMiner, позволяющей извлекать из баз данных больше информации, чем любые другие решения, за счет чего находить в тексте нечеткие совпадения [12]. Помимо прочего только Déjà Vu X3 поддерживает управление голосовыми командами за счет внедрения Dragon Naturally Speaking. Разработчик предполагает, что применение функционала голосового управления даст пользователям огромный прирост производительности.

Интерфейс Déjà Vu X3 представлен на рисунке 2.

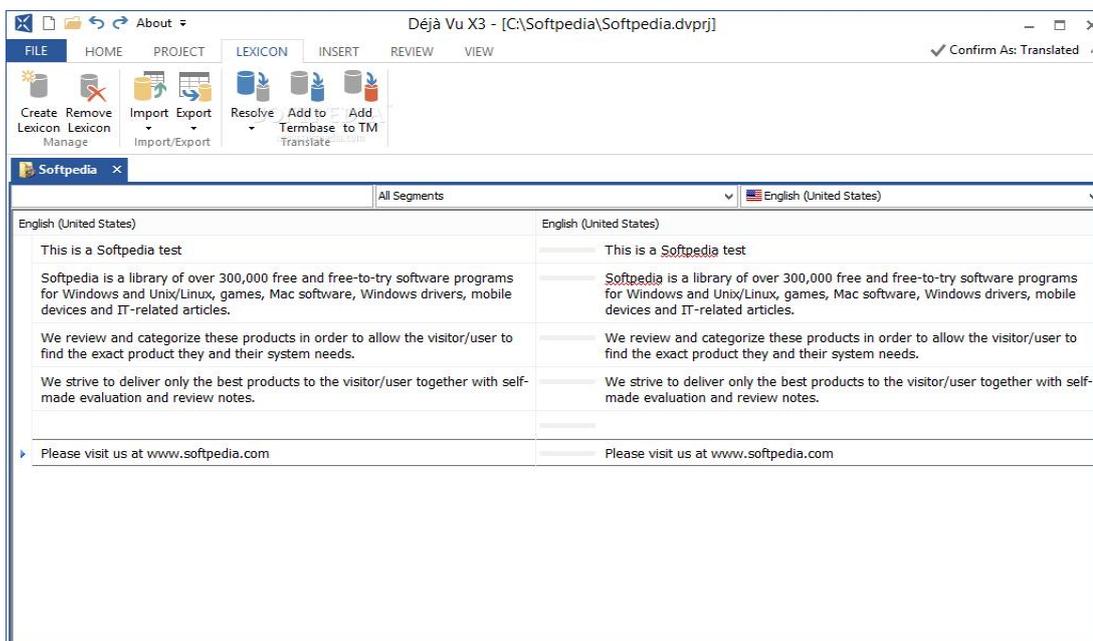


Рисунок 2 – Пример работы Déjà Vu X3

Достоинства:

- Наличие базы данных памяти переводов;
- Встроенные механизмы машинного перевода;
- Имеет функционал коллективной работы;

- Облачные сервисы хранения переводов;
- Встроенная проверка орфографии;
- Поддержка управления речью и голосовыми командами.

Недостатки:

- Проприетарное программное обеспечение (цена самого дешевого пакета на момент января 2018 г. 420 евро [11]);
- Отсутствие функционала работы с изображениями и видео;
- Реализован в виде приложения на ОС Windows (отсутствует кросс-платформенность).

**SmartCat** [7] – бесплатная CAT система, разработанная компанией SmartCat. Реализована в виде веб-сервиса, позволяющего не только переводить тексты, но и автоматизировать работу переводческих агентств, размещать коммерческие заказы, искать заказчиков и исполнителей переводов и проводить платежи. SmartCat в большей степени представляет собой единую платформу для взаимодействия переводчиков и работодателей, нежели просто CAT систему [13].

Интерфейс SmartCat представлен на рисунке 3.

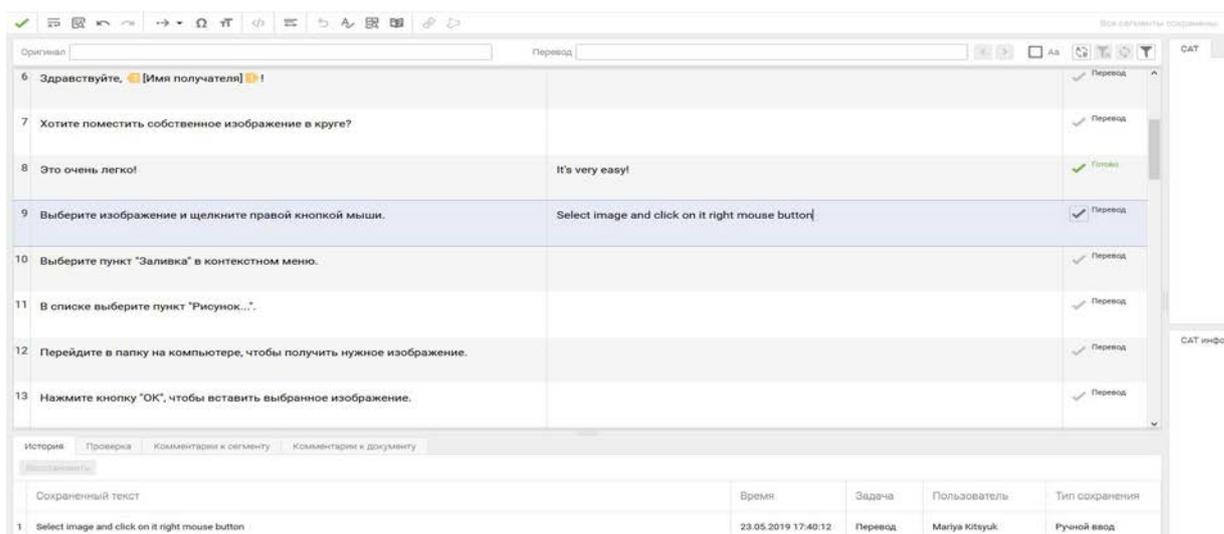


Рисунок 3 — Пример работы SmartCat

Достоинства:

- Наличие базы данных памяти переводов;

- Встроенные инструменты оптического распознавания текста;
- Встроенные механизмы машинного перевода;
- Имеет функционал коллективной работы;
- Облачные сервисы хранения переводов;
- Встроенная проверка орфографии;
- Бесплатная регистрация и некоммерческое использование;
- Реализован в виде веб-сервиса (кроссплатформенность).

#### Недостатки:

- Отсутствие функционала работы видео;
- Отсутствие полноценного функционала работы с изображениями;
- Для работы обязательно подключение к сети интернет.

**MemoQ** [8] – CAT система, разработанная компанией MemoQ Translation Technologies. Содержит в себе стандартный технологический набор – память переводов, глоссарии, базу терминов и инструменты экстракции терминов. Интерфейс MemoQ представлен на рисунке 4.

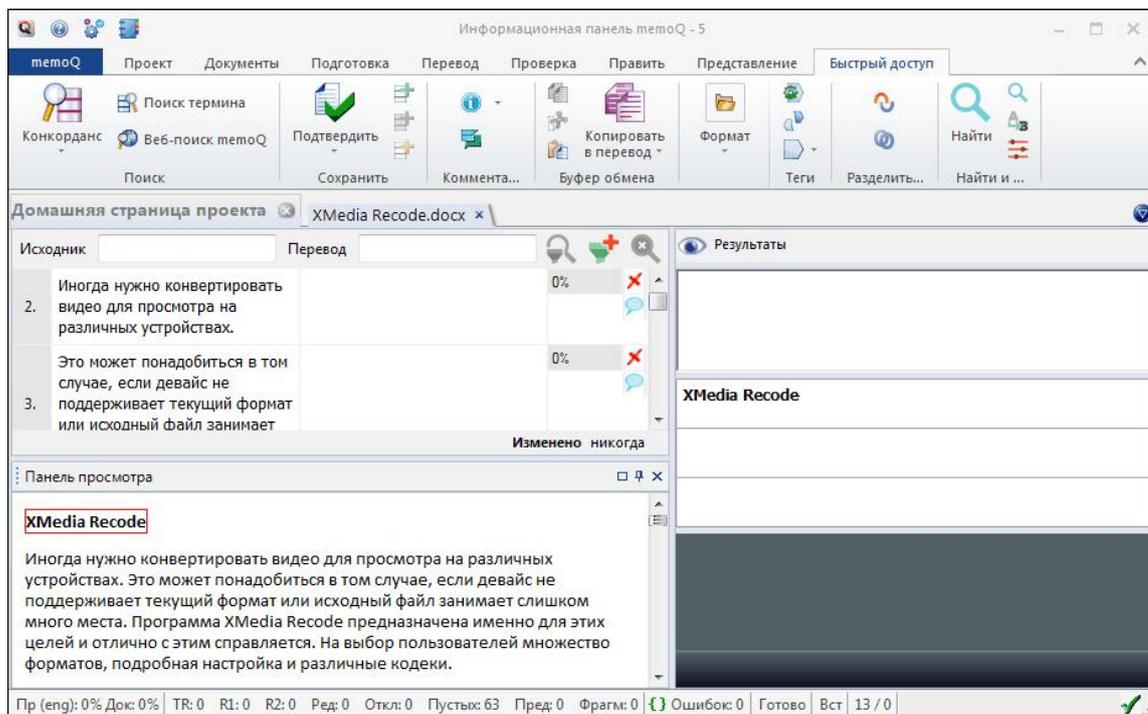


Рисунок 4 – Пример работы MemoQ

Достоинства:

- Наличие базы данных памяти переводов;
- Встроенные механизмы машинного перевода;
- Имеет функционал коллективной работы;
- Облачные сервисы хранения переводов;
- Поддерживает возможность подключения сторонних модулей (память переводов, глоссариев, автоматизированного создания глоссариев и тд.), комбинировать которые можно исходя из требуемого функционала;
- Встроенная проверка орфографии.

Недостатки:

- Проприетарное программное обеспечение (цена самого простого пакета с поддержкой до 2020г. на момент января 2018г. 695 евро[14]);
- Отсутствие функционала работы видео;
- Отсутствие полноценного функционала работы с изображениями;
- Реализован в виде приложения на ОС Windows (отсутствует кросс-платформенность).

У всех рассмотренных аналогов имеются ограничения на количество поддерживаемых форматов данных.

Таблица 1. Сравнительный анализ CAT-систем

Наименование	Поддерживаемые форматы (название пакета программ при поддержке всех его форматов)	Платформа	Лицензия
SDL Trados	Microsoft Office formats, In-Design formats, OpenOffice formats, SDL Trados formats, Interleaf formats, Framemaker formats, Wordfast formats, MemoQ formats, HTML, SGML, XML, SVG, TTX, ITD, MIF, TXT, INI, отсканированные документы и	ОС Windows	Проприетарное ПО

## Продолжение таблицы 1

	изображения PDF, JPG, TIFF, BMP, PNG, GIF, DJVU		
Déjà Vu X3	Microsoft Office formats, InDesign formats, OpenOffice formats, SDL Trados formats, Interleaf formats, PageMaker formats, CNT, MIF, QuarkXPress, ASCII, PROPERTIES, HTML, HTM, XML, RC, C/Java/C++, IBM TM/2, JavaScript, VBScript, ODBC, TMX, EBU, INI, PO/POT, ITD, XLF, XLIF, XLIFF, MQXLIFF, VDX, PDF, NXT, PPF, TXML	ОС Windows	Проприетарное ПО
SmartCat	Microsoft Office formats, InDesign formats, OpenOffice formats, XLSX, XLS, HTML, HTM, RESX, TTX, INI, SDLXLIFF, XLF, XLIFF, TMX, отсканированные документы и изображения PDF, JPG, TIFF, BMP, PNG, GIF, DJVU	Кроссплатформа	Проприетарное ПО
MemoQ	Microsoft Office formats, InDesign formats, OpenOffice formats, MIF, XML, DITA, XML, MM, PO, HTML, HMT, SHT, properties, DOC, RTF, BAK, PPF, PPTX, PPSX, POTX, SLDX, VDX, HHC, HHK, ODT, ODF, TXT, INF, INI, REG, PDF, SVG, SDLPPX, TTX, SDLXLIFF, TMX, TXML, RESX, XLF, XLIF, XLIFF	ОС Windows	Проприетарное ПО

### 1.3. ВЫВОД

Для решения задачи работы с переводами уже существует ряд специализированного программного обеспечения. Однако большинство из них не имеет функционала работы с изображениями. CAT системы же способные хотя бы обработать картинку, в свою очередь не предоставляют адекватного функционала для ведения дальнейшей работы.

МетоQ также позволяет загрузить в качестве документа изображение, которое можно открыть в окне предпросмотра, но поля с текстом на языке оригинала придется или заполнять вручную, или не заполнять вовсе. Так же при загрузке картинки будет автоматически сформировано дополнительное окно “инструкции”, в котором можно оставить какое-либо сообщение касательно перевода изображения.

SmartCat же при работе с изображением просто считывает буквенные символы с изображения инструментами нейронных сетей, и выгружает данные как обычный текст для дальнейшего перевода. Результат подобной обработки нельзя назвать качественным. Пример работы SmartCat изображением представлен на рисунке 5.

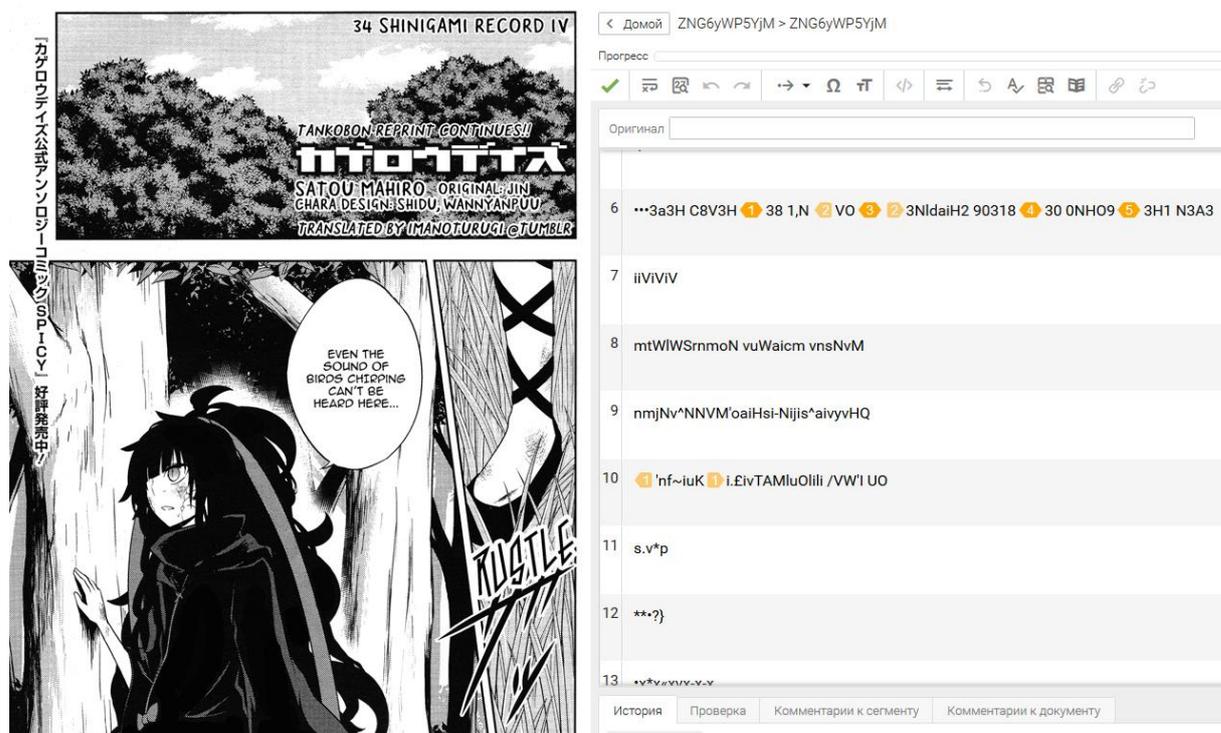


Рисунок 5 – Исходное изображение слева. Результат обработки изображения SmartCat – справа

Исходя из анализа аналогичного программного обеспечения, можно прийти к выводу, что на данный момент никто не решает задачу создания инструментария для быстрой и комфортной работы над переводом информации

размещенной на изображениях, что делает решение данной задачи возможным и актуальным.

## 2. ОПРЕДЕЛЕНИЕ ТРЕБОВАНИЙ

### 2.1. ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ

Можно определить следующий набор функциональных требований к системе:

1. Веб-приложение Bearlog.org должно обеспечивать регистрацию и авторизацию пользователей.
2. Веб-приложение Bearlog.org должно обеспечивать функционал создания перевода, включающий:
  - a. Создание проекта с указанием его названия
  - b. Указание языков оригинала и перевода
  - c. Загрузку изображений
3. Веб-приложение Bearlog.org должно предоставлять инструментарий для работы над переводом:
  - a. Инструменты выделения на целевом изображении контента необходимого к переводу
  - b. Автоматическое создание текстовых полей, сопоставляемых с соответствующим им фрагментам выделения контента на изображении
  - c. Автоматическое сохранение всех данных введенных пользователем (в т.ч. и между сессиями)
4. Веб-приложение Bearlog.org должно предоставлять функционал редактирования созданного перевода:
  - a. Возможность добавлять, редактировать и удалять элементы перевода в любой момент времени
  - b. Возможность удалить имеющийся проект
  - c. Возможность предоставлять доступ к переводу другим пользователям
5. Веб-приложение Bearlog.org должно предоставлять пользователю возможность скачивать исходные материалы и переводы

## 2.2. НЕФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ

В результате анализа функциональных требований и обзора аналогичных проектов, были сформулированы следующие нефункциональные требования:

1. Веб-приложение Bearlog.org должно поддерживать загрузку изображений в форматах png, jpg, bmp, gif (без анимаций);
2. Веб-приложение Bearlog.org должно иметь простой интерфейс, не требующий дополнительного обучения для пользователей, обладающих компьютерной грамотностью;
3. Веб-приложение Bearlog.org должно быть работоспособно в любом браузере (актуальной на момент использования версии);
4. Веб-приложение Bearlog.org должно загружаться менее чем за 3 сек. (весь контент открытой страницы);
5. Личные данные пользователей должны храниться в зашифрованном виде;

### 3. ПРОЕКТИРОВАНИЕ

#### 3.1. ДИАГРАММА ВАРИАНТОВ ИСПОЛЬЗОВАНИЯ

Диаграмма вариантов использования (прецедентов) отражает отношение между актерами и прецедентами системы и позволяет описать систему на концептуальном уровне. Прецедент – это возможность моделируемой системы, благодаря которой, актер может получить конкретный, измеримый и нужный ему результат.

На основе сформулированных функциональных требований была составлена диаграмма вариантов использования системы, представленная на рисунке 6.

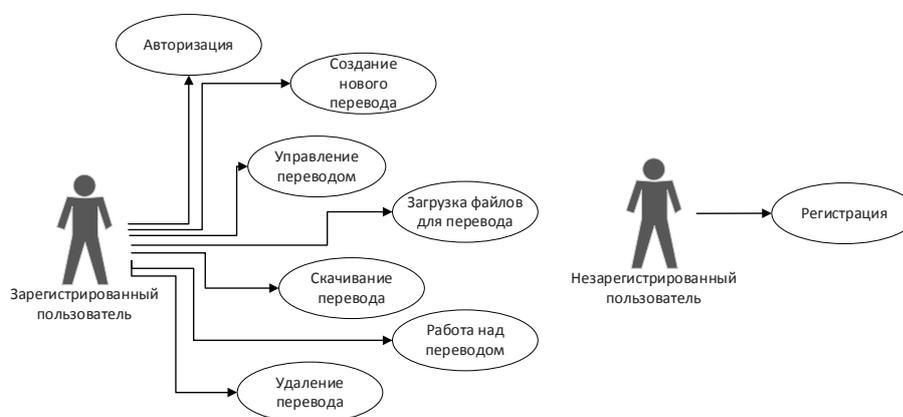


Рисунок 6 – Диаграмма вариантов использования

Основные акторы, взаимодействующие с системой:

- Незарегистрированный пользователь – посетитель веб-сайта, не имеющий доступ к полному функционалу веб-сайта;
- Зарегистрированный пользователь – пользователь, имеющий полный доступ к функционалу веб-сайта.

Варианты использования:

Незарегистрированный пользователь может:

- Зарегистрироваться.

Зарегистрированный пользователь может:

- Авторизовываться на сайте с помощью логина и пароля;

- Создавать новый перевод, заполнив обязательные поля формы “Создание проекта”:
  - Название;
  - Язык оригинала;
  - Язык перевода;
- Загружать требуемые к переводу изображения в созданный перевод;
- Работать над загруженными в перевод изображениями;
- Скачивать изображения из проекта;
- Работать с ранее созданными переводами (загружать и удалять изображения из переводов, работать с изображениями, скачивать имеющиеся переведенные изображения);
- Удалять проекты.

### 3.2. ПРОЕКТИРОВАНИЕ СТРУКТУРЫ БАЗЫ ДАННЫХ

Основные задачи проектирования базы данных:

- Обеспечение хранения в базе данных всей необходимой информации;
- Обеспечение возможности получения данных по всем необходимым запросам;
- Сокращение избыточности и дублирования данных.

Эскиз спроектированной базы данных представлен на рисунке 7.

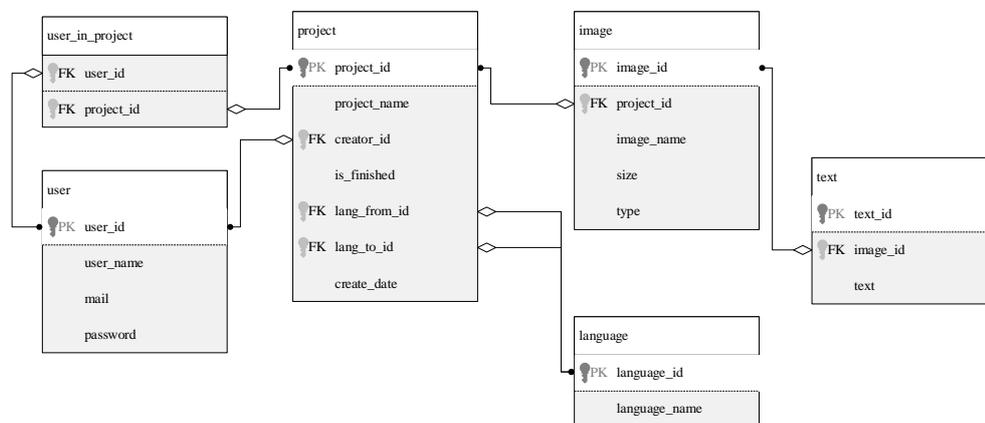


Рисунок 7 – Эскиз базы данных

Проектирование базы данных позволило выделить 5 таблиц:

- **user** – зарегистрированный пользователь, содержащий в себе поля:
  - user\_id – уникальный генерирующийся идентификатор пользователя;
  - user\_name – уникальное имя пользователя, указываемое пользователем при регистрации, требуемое при авторизации;
  - mail – почта пользователя;
  - password – пароль для авторизации пользователя;
- **project** - проект, содержащий в себе поля:
  - project\_id – уникальный генерируемый идентификатор проекта;
  - project\_name – название проекта;
  - creator\_id – идентификатор создателя проекта;
  - is\_finished – флаг завершения работы над переводом;
  - lang\_from\_id – идентификатор языка оригинала;
  - lang\_to\_id – идентификатор языка перевода;
  - create\_date – генерируемая дата создания;
- **user\_in\_project** – пользователи в проекте, содержащие в себе поля:
  - user\_id – идентификатор пользователя;
  - project\_id – идентификатор проекта;
- **language** – язык, содержащий в себе поля:
  - language\_id – уникальный генерируемый идентификатор языка;
  - language\_name – название языка;
- **image** – изображение, содержащее в себе поля:

- image\_id – уникальный генерируемый идентификатор изображения;
- project\_id – идентификатор проекта, содержащего изображение;
- image\_name – название изображения;
- size – размер изображения в Мб;
- type – формат изображения;
- **text** – текст, содержащий в себе поля:
  - text\_id – уникальный генерируемый идентификатор текста;
  - image\_id – идентификатор изображения, содержащего текст;
  - text – текст перевода.

### 3.3. ПРОЕКТИРОВАНИЕ СТРУКТУРЫ ГРАФИЧЕСКОГО ИНТЕРФЕЙСА

Интерфейс веб-сервиса состоит из 3 основных частей: шапки сайта, тела сайта и подвала.

Шапка сайта должна содержать в себе навигационную панель, предоставляющую простой и быстрый доступ к основному функционалу сайта.

В зависимости от того, на какой странице сайта находится пользователь, тело сайта может принимать следующие виды:

- На главной странице должны выводиться проекты авторизовавшегося пользователя и интерфейс управления проектами (добавить проект, редактировать проект и удалить проект);
- На странице редактирования проекта должны выводиться изображения загруженные в проект, интерфейс управления изображением (загрузить изображение, перевести изображение и удалить изображение) и краткая информация о проекте;

- На странице перевода изображения тело должно быть разделено на две части. Левая часть должна содержать загруженное изображение и интерфейс графического редактора, в правой части должны генерироваться поля для ввода текста, соответствующие выделенным на изображении участкам текста.

Подвал сайта должен содержать информацию о создателе веб-сайта и контактные данные для связи с создателем.

Макет графического интерфейса сайта представлен на рисунках 7-8.

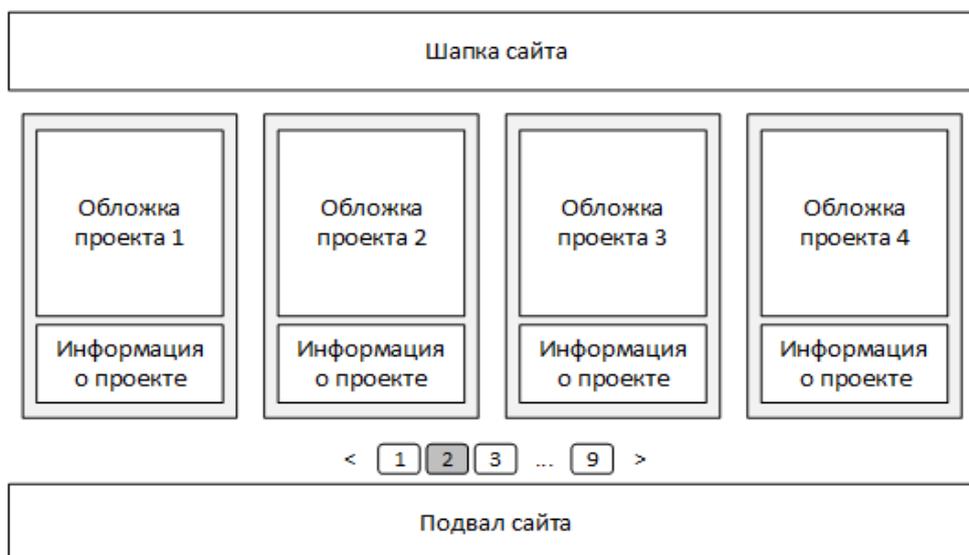


Рисунок 7 - Макет главной страницы сайта

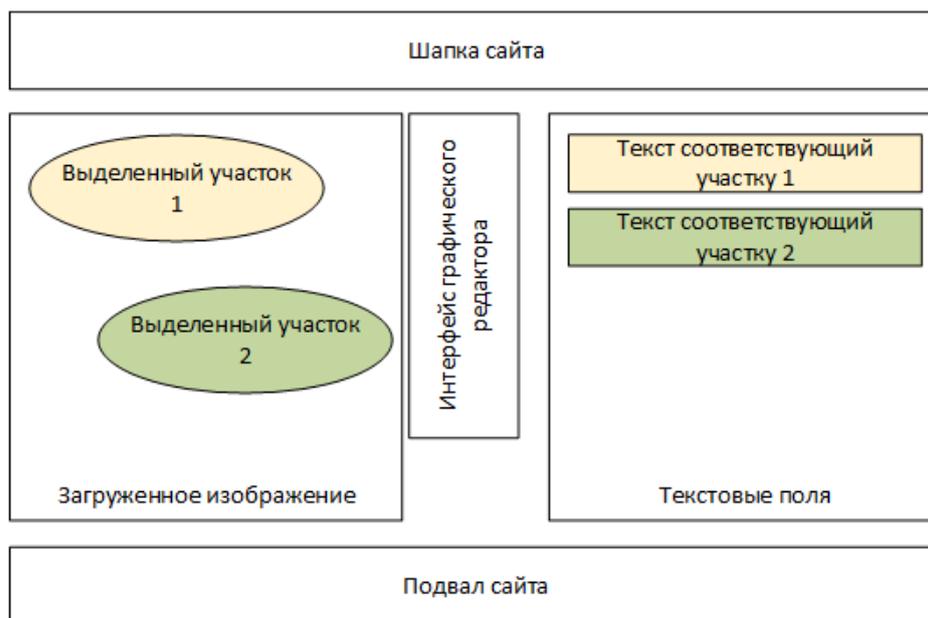


Рисунок 8 – Макет графического интерфейса страницы перевода изображения

### 3.4. ПРОЕКТИРОВАНИЕ АРХИТЕКТУРЫ СИСТЕМЫ

На рисунке 9 представлена диаграмма компонентов веб-сайта. При проектировании использовался архитектурный паттерн MVC.

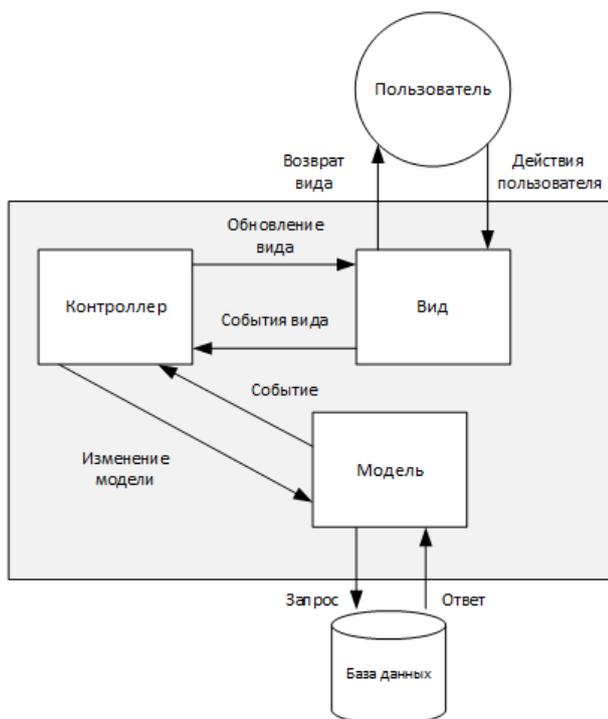


Рисунок 9 – Диаграмма компонентов системы

MVC – паттерн проектирования, предназначенный для разделения логики и пользовательского интерфейса. В паттерне MVC модель предоставляет данные и правила логики, вид (или представление) отвечает за пользовательский интерфейс, а контроллер обеспечивает взаимодействие между моделью и представлением [20].

Главным достоинством данного паттерна является четкое разделение логики компонентов приложения.

Помимо изолирования компонентов, MVC существенно уменьшает сложность больших приложений за счет структурирования кода, чем облегчает поддержку, тестирование и повторное использование решений.

Подробнее, модели, виды и контроллеры написанные в ходе реализации веб-сайта будут рассмотрены в разделе “Реализация”.

## 4. РЕАЛИЗАЦИЯ

### 4.1. РЕАЛИЗАЦИЯ БАЗЫ ДАННЫХ

База данных была реализована в СУБД MySQL, с помощью веб-интерфейса phpMyAdmin. На рисунке 10 представлена схема базы данных:

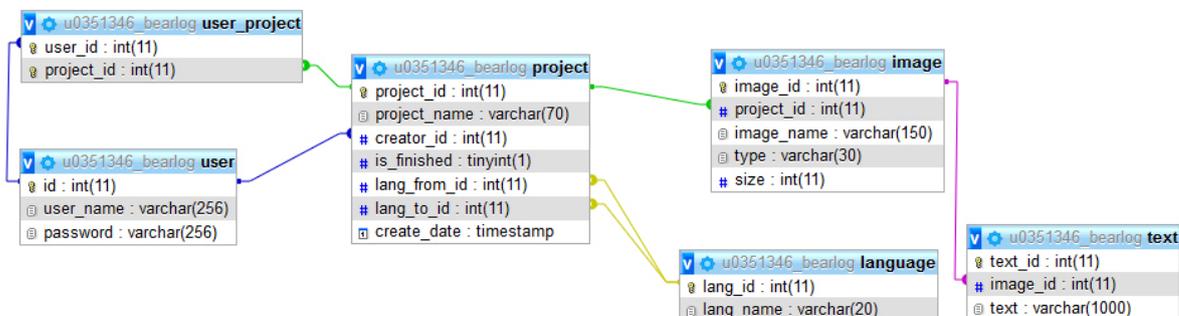


Рисунок 10 – Схема базы данных

### 4.2. КОМПОНЕНТЫ СИСТЕМЫ

На рисунке 11 представлена полная диаграмма компонентов, созданных в ходе реализации веб-сайта в соответствии с паттерном MVC.

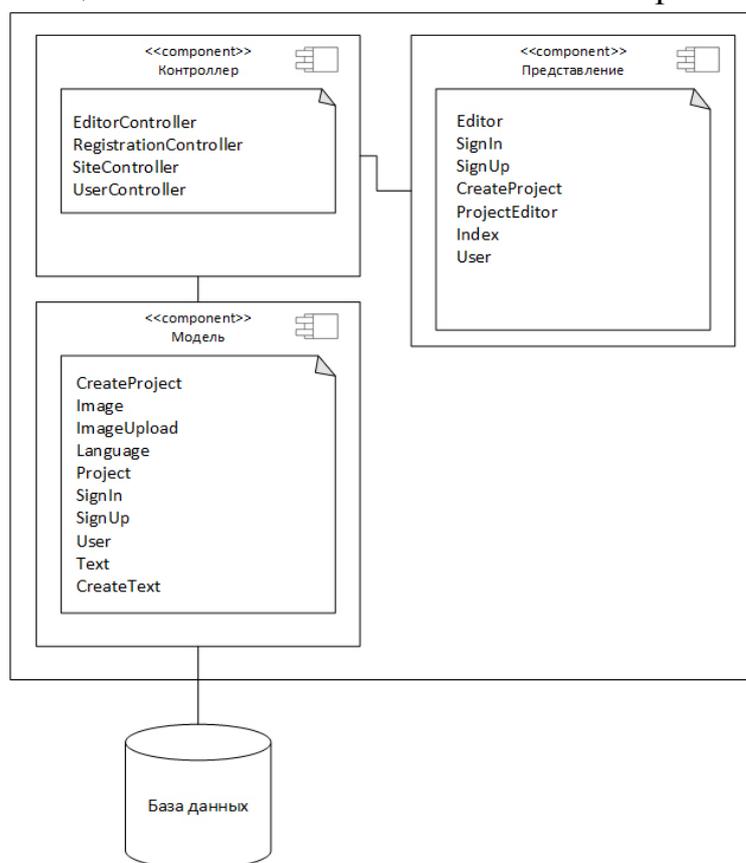


Рисунок 11 – Диаграмма компонентов системы

Основными компонентами системы являются:

- Контроллеры:
  - EditorController – контроллер осуществляющий управление функционалом рендеринга вида страницы перевода;
  - RegistrationController – контроллер осуществляющий управление функционалом регистрации и авторизации.
  - SiteController – контроллер осуществляющий управление функционалом создания проектов и загрузкой в проекты изображений;
  - UserController – контроллер осуществляющий управление функционалом рендеринга вида страницы информации о пользователе;
- Модели
  - CreateProject – модель прослойка для фильтрации, валидации и передачи подготовленных данных;
  - CreateText – модель прослойка для фильтрации, валидации и передачи подготовленных данных;
  - Text – модель, связанная с таблицей text;
  - User – модель, связанная с таблицей user;
  - SignIn – модель прослойка для фильтрации, валидации и передачи подготовленных данных;
  - SignUp – модель прослойка для фильтрации, валидации и передачи подготовленных данных;
  - Project – модель, связанная с таблицей project;
  - Language – модель, связанная с таблицей language;
  - Image – модель, связанная с таблицей image;
  - ImageUpload – модель прослойка для фильтрации, валидации и передачи подготовленных данных;
- Виды
  - Editor – вид страницы перевода изображения

- SignIn – вид страницы авторизации пользователя;
- SignUp – вид страницы регистрации пользователя;
- CreateProject – вид страницы создания проекта;
- ProjectEditor – вид страницы редактирования проекта;
- Index – вид страницы проектов пользователя.
- User – вид страницы с информацией о пользователе;

### 4.3. РЕАЛИЗАЦИЯ ИНТЕРФЕЙСОВ

Незарегистрированному пользователю доступен только экран регистрации. Экран регистрации содержит форму с обязательными для заполнения полями ввода электронной почты, логина пользователя, пароля и повтора пароля. Экранная форма регистрации представлена на рисунке 12.

My Application Home Registration Login

Регистрация

User Name  
test3

Password  
\*\*\*\*\*

Зарегистрироваться

Разработано: Кицюк М.А. КЗ-452

Рисунок 12 – Экранная форма регистрации

Для доступа к полному функционалу веб-сайта зарегистрированному пользователю нужно авторизоваться по логину и паролю.

Экранная форма авторизации представлена на рисунке 13.

Авторизация

User Name  
test3

Password  
\*\*\*\*\*

Войти

Разработано: Кицок М.А. КЭ-452

Рисунок 13 – Экранная форма авторизации

После авторизации пользователя переводит на главную страницу сайта, на которую выводятся проекты, созданные пользователем. Для простоты ориентирования проекты выводятся в виде карточек, содержащих обложку, полную информацию о проекте и управляющие элементы.

Обложка формируется из первого загруженного в проект изображения.

Информация о проекте выгружается из базы данных и включает в себя:

- Название проекта;
- Язык оригинала;
- Язык перевода;
- Дату создания проекта.

Управляющие элементы представлены кнопками “Редактировать” и “Удалить”. При удалении выводится дополнительное окно подтверждения действия, для исключения случайного нажатия. При редактировании пользователя переводит в окно содержимого проекта.

Страница с пользовательскими проектами представлена на рисунке 14.

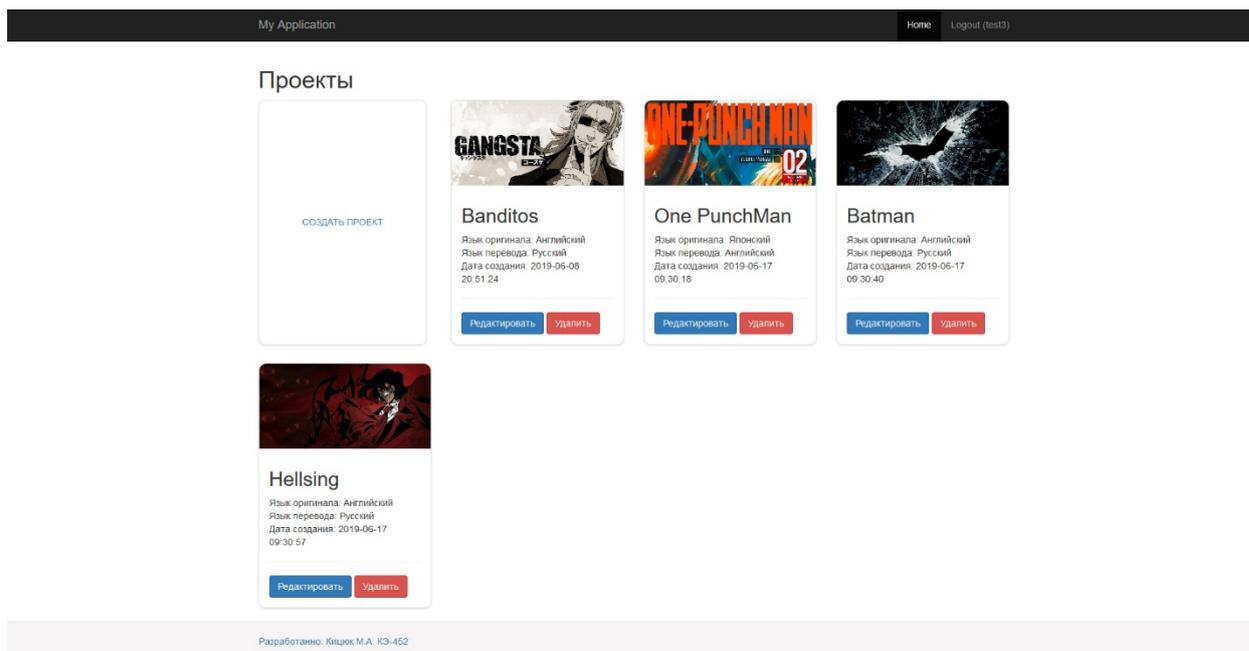


Рисунок 14 – Главная страница сайта с проектами

Содержимое проекта представлено загруженными в проект изображениями. Для простоты ориентирования изображения выводятся в виде карточек содержащих миниатюру соответствующего изображения, название изображения и управляющие элементы.

Миниатюра представляет собой само изображение в низком разрешении.

Название изображения выгружается из базы данных.

Управляющие элементы представлены кнопками “Редактировать” и “Удалить”. При удалении выводится дополнительное окно подтверждения действия, для исключения случайного нажатия. При редактировании пользователя переводит в окно перевода изображения.

Страница содержимого проекта представлена на рисунке 15.

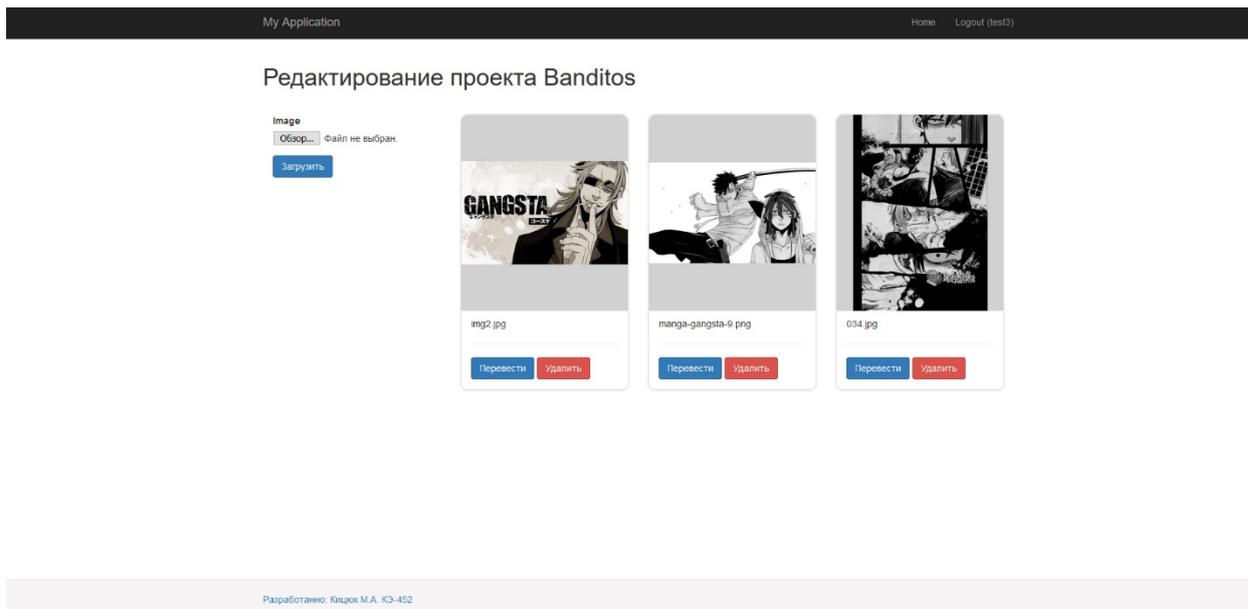


Рисунок 15 – Редактирование проекта

Экранная форма перевода изображения представляет собой окно разделенное на три части. Левая часть – переводимое изображение, центральная часть – интерфейс графического редактора, правая часть – автоматически генерируемые текстовые поля.

Графический редактор предоставляет инструментарий как для свободного рисования кистью, так и для рисования графических примитивов: прямоугольников, эллипсов, линий и текстовых полей, а также функционал выбора цвета на rgb палитре.

Редактор может работать в двух режимах:

- Режим создания элементов перевода;
- Режим графического редактора;

В режиме создания элементов перевода создание каждой фигуры инициирует генерацию связанного с этой фигурой текстового поля для ввода перевода. В дальнейшем при выделении фигуры подсвечивается соответствующее ей текстовое поле и наоборот, при выделении текстового поля, подсвечивается связанная с ней фигура на изображении. Это нужно для ускорения ориентирования в изображении.

В режиме графического редактора рисование фигуры не инициирует генерацию текстового поля и позволяет редактировать изображения без лишней нагрузки на сервер.

Режим редактора выбирается выставлением флага.

Каждый нарисованный на изображении элемент можно переместить и удалить.

Сохранение всех данных осуществляется по нажатию кнопки “Сохранить”.

Окно перевода изображения представлено на рисунке 16.

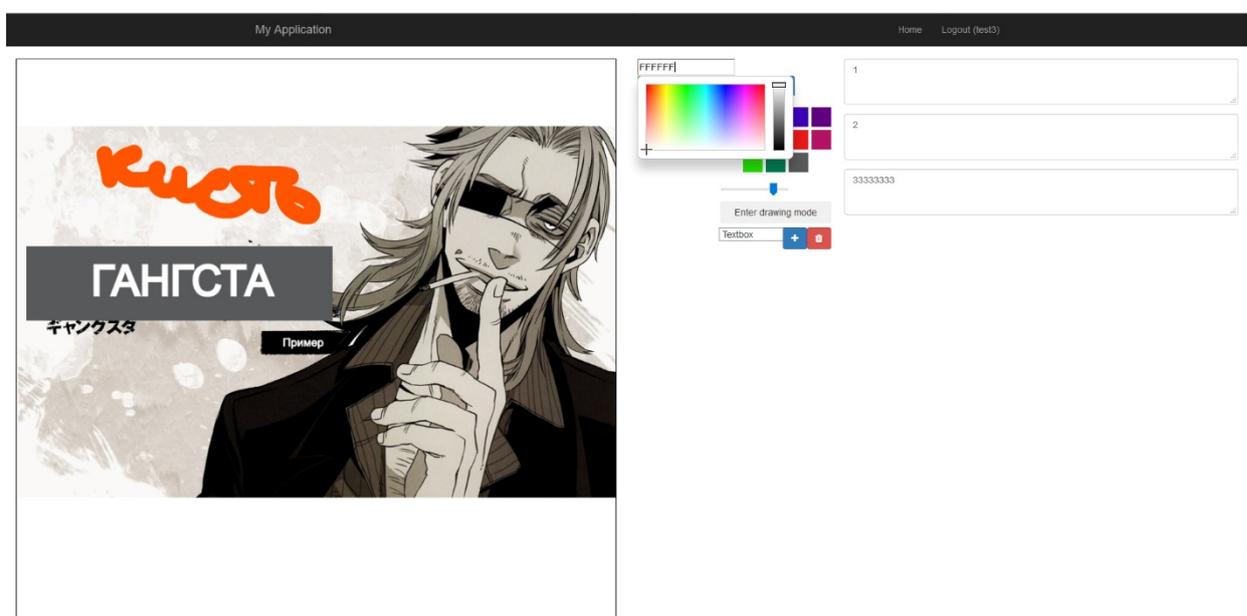


Рисунок 16 – Перевод изображений

## 5. ТЕСТИРОВАНИЕ

Для тестирования веб-сайта были выбраны следующие виды тестирования:

- Приемочное пользовательское тестирование;
- Функциональное тестирование;
- Интеграционное тестирование.

### 5.1. МЕТОДОЛОГИЯ ТЕСТИРОВАНИЯ

#### **Юзабилити тестирование.**

Данный вид тестирования определяет степень удобства интерфейса. Для оценки привлекаются представители целевой аудитории продукта и юзабилити-специалисты, которые работают с интерфейсом, выполняя подготовленные для них задания, сформулированные на основе заранее проведенных метрик системы. На основе поведения респондентов юзабилити-специалисты строят выводы о наличии в интерфейсе юзабилити-проблем [21].

#### **Интеграционное тестирование.**

Интеграционное тестирование представляет собой полную проверку готового программного продукта с целью выявления ошибок, возникающих в процессе интеграции программных компонентов.

#### **Функциональное тестирование.**

Функциональное тестирование – это тип тестирования программного обеспечения, при котором система тестируется на соответствие заявленным функциональным требованиям и спецификациям.

Функции (или характеристики) испытываются путем подачи на них входного сигнала и оценки выходного сигнала. Функциональное тестирование гарантирует, что требования выполняются в рамках работы приложения.

Стоит отметить, что данное тестирование оценивает не качество самого процесса обработки данных, а только конечные результаты [22].

## **5.2. ПРОВЕДЕНИЕ ПРОЦЕДУРЫ ТЕСТИРОВАНИЯ**

### **Юзабилити тестирование.**

Юзабилити тестирования осуществлялось 50 независимыми пользователями, с различным уровнем компьютерной грамотности.

В ходе тестирования у респондентов не возникало сложностей при работе с функционалом сайта, что показывает правильность проектирования интерфейса взаимодействия с функционалом.

### **Интеграционное тестирование.**

Для проведения интеграционного тестирования после окончательной сборки приложение было запущено:

На реальных экранах с разрешениями:

- 1920x1080 px;
- 1600x900 px;
- 1366x768;
- 1024x768;
- 1280x1024;

На эмулированных экранах с разрешениями:

- 2560x1600 px;
- 1440x900 px;
- 1280x800px;
- 1600x1200px;

В браузерах:

- 2560x1600 px;
- 1440x900 px;
- 1280x800px;
- 1600x1200px;

На различных по размеру экранам была проверена работа адаптивного пользовательского интерфейса.

Все функции веб-приложения были протестированы в браузерах:

- Chrome старше 60 версии;
- Firefox старше 52 версии;
- Opera старше 45 версии;
- Edge старше 16 версии;
- Safari старше 10.1 версии;

В различных браузерах проверялась корректная работа функционала сайта.

По результатам тестирования гарантирована корректная работа:

- пользовательского интерфейса веб-сайта у 83% пользователей [23];
- функционала сайта у 78% пользователей [24].

Работоспособность сайта при условиях отличных от протестированных не гарантируется, но не исключается.

### **Функциональное тестирование.**

Список тестов для проведения функционального тестирования и его результаты, представлен в таблице 2.

Таблица 2. Набор тестов для проведения функционального тестирования и его результаты.

№	Описание	Шаги	Ожидаемый результат	Пройден
1	Регистрация пользователя	1. Зайти на сайт; 2. Перейти на экран регистрации; 3. Ввести имя пользователя, пароль и электронную почту;	Сообщение на экране "Пользователь успешно зарегистрирован"	Да
2	Авторизация пользователя	1. Зайти на сайт; 2. Перейти на экран авторизации; 4. Ввести имя пользователя, пароль;	Откроется страница с проектами авторизовавшегося пользователя	Да
3	Создание проекта	1. На странице "Проекты" нажать кнопку "Создать проект" 2. На открывшейся странице ввести данные о проекте: название проекта, язык оригинала и язык перевода.	На странице с проектами пользователя появится новый проект содержащий введенные пользователем данные и дату создания.	Да

Продолжение таблицы 2

4	Редактирование проекта	1. На странице “Проекты” нажать кнопку “Редактировать проект” соответствующую нужному проекту.	Откроется страница “Редактирование проекта” с загруженными в проект изображениями.	Да
5	Удаление проекта	1. На странице “Проекты” нажать кнопку “Удалить проект” соответствующую нужному проекту.	На странице с проектами пользователя исчезнет выбранный к удалению проект.	Да
6	Загрузка изображения в проект	1. На странице “Редактирование проекта” нажать кнопку “Загрузить”	На странице “Редактирование проекта” появится загруженное изображение.	Да
7	Удаление изображения из проекта	1. На странице “Редактирование проекта” нажать кнопку “Удалить”, соответствующую нужному изображению	На странице “Редактирование проекта” исчезнет выбранное к удалению изображение.	Да
8	Перевод изображения	1. На странице “Редактирование проекта” нажать кнопку “Перевести изображение”	Откроется страница “Перевод изображения” с выбранным изображением в окне графического редактора.	Да
9	Сохранение перевода изображения	1. На странице “Перевод изображения” нажать кнопку “Сохранить”.	В окне “Перевод изображения” появится уведомление “Успешно сохранено”. В дальнейшем при открытии сохраненного ранее изображения на перевод будет открываться сохраненный в последний раз прогресс перевода.	Да
10	Скачивание переведенного изображения	1. На странице “Перевод изображения” нажать кнопку “Скачать изображение”.	Появится браузерная форма сохранения файла в формате jpg, png, bmp или gif.	Да

## **ЗАКЛЮЧЕНИЕ**

В рамках выпускной квалификационной работы было спроектировано и реализовано веб-приложение «Bearlog.org» для работы переводчиков с изображениями.

Для достижения поставленной цели были решены следующие задачи:

1. Выполнен анализ предметной области.
2. Выполнено проектирование веб-приложения с использованием паттерна архитектурного проектирования MVC.
3. Реализовано веб-приложение с учетом необходимых функциональных и нефункциональных требований.
4. Проведено тестирование веб-приложения с использованием юзабилити тестирования, функционального тестирования и интеграционного тестирования.

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Веб-сайт «Imperial college of Londonz». [Электронный ресурс]. URL: – <https://www.webcitation.org/5jsaMDqiR?url=http://www3.imperial.ac.uk/portals/pls/portallive/docs/1/7307707.PDF>. (Дата обращения: 20.02.2018).
2. Веб-сайт «US News 100 Best Jobs». [Электронный ресурс]. URL: – <https://money.usnews.com/careers/best-jobs/rankings/the-100-best-jobs>. (Дата обращения: 20.02.2018).
3. Веб-сайт «Career cast 2017 jobs rating». [Электронный ресурс]. URL: – <https://www.careercast.com/jobs-rated/2017-jobs-rated-report?page=1>. (Дата обращения: 20.02.2018).
4. Веб-сайт «Career cast 2016 jobs rating». [Электронный ресурс]. URL: – <https://www.careercast.com/jobs-rated/jobs-rated-report-2016-ranking-200-jobs?page=1>. (Дата обращения: 20.02.2018).
5. Веб-сайт «SDL Trados». [Электронный ресурс]. URL: – <https://www.sdltrados.com/>. (Дата обращения: 20.02.2018).
6. Веб-сайт «Atril soft». [Электронный ресурс]. URL: – <https://atril.com/shop/>. (Дата обращения: 20.02.2018).
7. Веб-сайт «SmartCat». [Электронный ресурс]. URL: – <https://smartcat.ai/>. (Дата обращения: 20.02.2018).
8. Веб-сайт «МемоQ». [Электронный ресурс]. URL: – <https://www.мемоq.com>. (Дата обращения: 20.02.2018).
9. Веб-сайт «SDL Trados statistic». [Электронный ресурс]. URL: – <https://www.sdltrados.com/products/trados-studio/>. (Дата обращения: 20.02.2018).
10. Веб-сайт «SDL Trados store». [Электронный ресурс]. URL: – <https://www.sdltrados.com/store/>. (Дата обращения: 20.02.2018).
11. Веб-сайт «Atril store». [Электронный ресурс]. URL: – <https://atril.com/shop/>. (Дата обращения: 20.02.2018).

12. Веб-сайт «Deja Vu X3 DeepMiner». [Электронный ресурс]. URL: – <https://atril.com/product/deja-vu-x3-professional/>. (Дата обращения: 20.02.2018).
13. Веб-сайт «Smartcat для работодателей». [Электронный ресурс]. URL: – <https://ru.smartcat.ai/lsp/>. (Дата обращения: 20.02.2018).
14. Веб-сайт «МемоQ». [Электронный ресурс]. URL: – <https://www.memoq.com/en/shopping-cart/products>. (Дата обращения: 20.02.2018).
15. Исследование влияния памяти переводов на эффективность работы [http://www.intercultural.urv.cat/media/upload/domain\\_317/arxius/TP3/yama\\_da.pdf](http://www.intercultural.urv.cat/media/upload/domain_317/arxius/TP3/yama_da.pdf)
16. Использование CAT среди переводчиков [https://www.jostrans.org/issue25/art\\_christensen.pdf](https://www.jostrans.org/issue25/art_christensen.pdf)
17. Веб-сайт «Comichron sales statistic». [Электронный ресурс]. URL: – <https://www.comichron.com/yearlycomicssales.html>. (Дата обращения: 20.02.2018).
18. Веб-сайт «Writing cooperative». [Электронный ресурс]. URL: – <https://writingcooperative.com/what-are-the-most-popular-literary-genres-6db5c69928cc>. (Дата обращения: 20.02.2018).
19. Веб-сайт «The news publishing standard». [Электронный ресурс]. URL: – <https://thenewpublishingstandard.com/global-book-market-valued-at-143bn/>. (Дата обращения: 20.02.2018).
20. Веб-сайт «Yii MVC». [Электронный ресурс]. URL: – <https://www.yiiframework.com/doc/guide/1.1/ru/basics.mvc>. (Дата обращения: 20.02.2018).
21. Веб-сайт «Experience UX». [Электронный ресурс]. URL: – <https://www.experienceux.co.uk/faqs/what-is-usability-testing>. (Дата обращения: 20.02.2018).

22. Веб-сайт «Stat counter». [Электронный ресурс]. URL: –  
<http://gs.statcounter.com/screen-resolution-stats>. (Дата обращения: 20.02.2018).
23. Веб-сайт «Software testing». [Электронный ресурс]. URL: –  
<http://softwaretestingfundamentals.com/functional-testing/>. (Дата обращения: 20.02.2018).
24. Веб-сайт «Browser usage table». [Электронный ресурс]. URL: –  
<https://caniuse.com/usage-table>. (Дата обращения: 20.02.2018).
25. Кицюк М.А. Исходный код приложения для работы переводчиков с изображениями Веб-сайт «Github». [Электронный ресурс]. URL: –  
<https://github.com/JhonPsina/Diplom>. (Дата обращения: 20.02.2018).

# ПРИЛОЖЕНИЕ А

## Исходный код контроллеров

### Листинг 1 – SiteController

```
class SiteController extends Controller
{
    //Вывести список проектов пользователя
    public function actionIndex()
    {
        $projects = Project::find()
            ->where(['creator_id' => Yii::$app->user->identity])
            ->all();

        // $projectImages = new ProjectImage[count($projects)];
        $projectImages = array();

        for ($i = 0; $i < count($projects); $i++) {
            $project = $projects[$i];

            $images = Image::find()
                ->where(['project_id' => $project->project_id])
                ->all();

            $firstImage = new Image();
            foreach ($images as $image) {
                $firstImage = $image;
                break;
            }

            $languageFroms = Language::find()
                ->where(['lang_id' => $project->lang_from_id])
                ->all();
            $languageTos = Language::find()
                ->where(['lang_id' => $project->lang_to_id])
                ->all();

            $langFrom = new Language();
            $langTo = new Language();
            foreach ($languageFroms as $lang) {
                $langFrom = $lang;
                break;
            }
            foreach ($languageTos as $lang) {
                $langTo = $lang;
                break;
            }

            $projectImage = new ProjectModel();
            $projectImage->project_id = $project->project_id;
            $projectImage->project_name = $project->project_name;
            $projectImage->image_id = $firstImage->image_id;
            $projectImage->image_path = $firstImage->project_id . '/' . $firstImage-
>image_id . '.' . $firstImage->type;
            $projectImage->image = $firstImage;
            $projectImage->lang_from_id = $project->lang_from_id;
            $projectImage->lang_to_id = $project->lang_to_id;
            $projectImage->create_date = $project->create_date;
            $projectImage->lang_from_name = $langFrom->lang_name;
```

```

$projectImage->lang_to_name = $langTo->lang_name;
    array_push($projectImages, $projectImage);
}
return $this->render('index', ['projects'=>$projectImages]);
}

// Создать проект
public function actionCreateproject()
{
    $create_model = new Createproject();

    if(isset($_POST['Createproject']))
    {
        $create_model -> attributes = Yii::$app->request->post('Createproject');

        if($create_model->project_name != null
            && ($create_model->validate() && $create_model->createproject()))
        {
            return $this->goHome();
        }
    }
    return $this->render('createproject', ['create_model'=>$create_model]);
}

// Редактирование проекта (добавление изображений, открытие их на редактирование и
пр.)
public function actionProjecteditor($project_id)
{
    $images = Image::find()->
        where(['project_id'=>$project_id])->all();

    $project_field = Project::find()->where(['project_id'=>$project_id,])->one();

    $model = new ImageUpload();

    if(Yii::$app->request->isPost){

        $model->image = UploadedFile::getInstance($model, 'image');
        $model->upload_to_db($project_id);
        $model->upload_to_server($project_id);

        return $this->render('projecteditor', ['project_field' => $project_field,
'model' => $model, 'images' => $images ]);
    }
    return $this->render('projecteditor', ['project_field' => $project_field, 'model'
=> $model, 'images' => $images]);
}

public function behaviors()
{
    return [
        'access' => [
            'class' => AccessControl::className(),
            'only' => ['logout'],
            'rules' => [
                [
                    'actions' => ['logout'],

```

```

        'allow' => true,
        'roles' => ['@'],
    ],
],
],
],
],
];
}

// Вывод ошибок
public function actions()
{
    return [
        'error' => [
            'class' => 'yii\web\ErrorAction',
        ],
        'captcha' => [
            'class' => 'yii\captcha\CaptchaAction',
            'fixedVerifyCode' => YII_ENV_TEST ? 'testme' : null,
        ],
    ];
}

// Выход из аккаунта
public function actionLogout()
{
    Yii::$app->user->logout();

    return $this->goHome();
}
}
}

```

## Листинг 2 – RegistrationController

```

class RegistrationController extends Controller
{
    //Проверка авторизации
    public function actionIndex()
    {
        if (!Yii::$app->user->isGuest)
        {
            return $this->goHome();
        }
        return $this->render('index');
    }

    //Регистрация
    public function actionSignup()
    {
        if (!Yii::$app->user->isGuest)
        {
            return $this->goHome();
        }

        $model = new Signup();
    }
}

```

```

if(isset($_POST['Signup']))
{
    $model -> attributes = Yii::$app->request->post('Signup');
    if($model->validate() && $model->signup())
    {
        return $this->goHome();
    }
}
return $this->render('signup', ['model'=>$model]);
}

//Вход
public function actionSignin()
{
    if (!Yii::$app->user->isGuest)
    {
        return $this->goHome();
    }

    $signin_model = new Signin();

    if( Yii::$app->request->post('Signin'))
    {
        $signin_model->attributes=Yii::$app->request->post('Signin');

        if($signin_model->validate()) {
            Yii::$app->user->login($signin_model->getUser());
            return $this->goHome();
        }
    }

    return $this->render('signin', ['signin_model'=>$signin_model]);
}
}

```

### Листинг 3 – EditorController

```

class EditorController extends Controller
{
    //Создать текстовое поле
    public function actionCreatetext()
    {
        $create_model = new CreateText();

        //if($create_model->createtext())
        if(isset($_POST['CreateText']))
        {
            $create_model -> attributes = Yii::$app->request->post('CreateText');
            return $this->render('index', ['models'=>$create_model]);
        }
        return $this->render('index', ['models'=>$create_model]);
    }

    //Вывести редактор
    public function actionIndex($project_id, $image_id)
    {
        $message = 'start <br/>';
        if(isset($_POST['0'])) {

```

```

// сохранение формы
$newTexts = array();
$i = 0;
while(isset($_POST[strval($i)])) {
    array_push($newTexts, $_POST[strval($i)]);
    $i++;
}

$texts = Text::find()->
    where(['image_id' => $image_id])->
    all();
$i = 0;
foreach ($texts as $tableText) {
    if($tableText->text_id > count($newTexts)) continue;

    $textRow = new CreateText();
    $textRow->text_id = $i;
    $textRow->text = $newTexts[$tableText->text_id];
    $textRow->image_id = $image_id;
    $textRow->updatetext($tableText);

    $i++;
}

$message = $message . 'Im here <br/>';
$message = $message . '$i=' . strval($i) . ' count=' .
strval(count($newTexts)-1) . '<br/>';
while($i <= count($newTexts)-1) {
    $message = $message . 'i entered this fucking cycle!!! <br/>';

    $textRow = new CreateText();
    $textRow->text_id = $i;
    $textRow->text = $newTexts[$i];
    $textRow->image_id = $image_id;
    $textRow->addtext();
    $i++;
}
$message = $message . 'Im there <br/>';
}

$tableTexts = Text::find()->
    where(['image_id' => $image_id])->
    all();

$count = 0;
foreach ($tableTexts as $text) $count++;

$path = $project_id . '/' . $image_id;
return $this->render('index', ['path' => $path, 'texts' => $tableTexts, 'count'
=> $count, 'message' => $message]);
}
}

```

## ПРИЛОЖЕНИЕ Б

### Исходный код моделей

#### Листинг 4 – Createproject

```
class Createproject extends Model
{
    public $project_name;
    public $lang_from_id;
    public $lang_to_id;
    public $creator_id;
    public $is_finished=0;

    // Валидация данных
    public function rules()
    {
        return[
            ['project_name', 'required'],
            ['lang_from_id', 'required'],
            ['lang_to_id', 'required'],
            ['project_name', 'string', 'min'=>2]
        ];
    }

    // Передача данных в базу таблицу Project
    public function createproject()
    {
        $project = new Project;

        $project->project_name = $this->project_name;
        $project->creator_id = $this->creator_id;
        $project->is_finished = $this->is_finished;
        $project->lang_from_id = $this->lang_from_id;
        $project->lang_to_id = $this->lang_to_id;

        return $project->save();
    }

    public function deleteproject()
    {
    }
}
```

#### Листинг 5 – CreateText

```
class CreateText extends Model
{
    public $text_id;
    public $image_id;
    public $text;

    // Обновление созданного текста
    public function updatetext($tableText)
    {
        $tableText->text = $this->text;
        return $tableText->save();
    }
}
```

```

// Передача нового текста в таблицу Text
public function addtext(){
    $text = new Text;

    $text->image_id = $this->image_id;
    $text->text = $this->text;

    return $text->save();
}
}

```

## Листинг 5 – ImageUpload

```

class ImageUpload extends Model
{
    public $image;
    public $image_id;
    public $image_name;

    // Валидация
    public function rules(){
        return[
            [['image'], 'file', 'extensions' => 'png, jpg','gif', 'bmp'],
        ];
    }

    // Передача данных на сервер
    public function upload_to_db($project_id)
    {
        if($this->validate()) {

            $db_image = new Image();

            $db_image->image_name = $this->image->baseName;
            $db_image->type = $this->image->extension;
            $db_image->size = $this->image->size;
            $db_image->project_id = $project_id;
            $db_image->save();
            $this->image_id = $db_image->image_id;
            return $db_image;
        }
    }

    // Сохранение в папку на сервере
    public function upload_to_server($project_id)
    {
        $filePath = "uploads/" . $project_id;
        FileHelper::createDirectory($filePath);
        $this->image->saveAs("uploads/$project_id/{$this->image_id}.{this->image-
>extension}");
    }
}

```

## Листинг 6 – Signin

```

class Signin extends Model
{
    public $user_name;
    public $password;
//Валидация
    public function rules()
    {
        return[
            [['user_name','password'], 'required'],
            ['password', 'validatePassword']
        ];
    }

    public function validatePassword($attribute, $params)
    {
        if(!$this->hasErrors())
        {
            $user = $this->getUser();
            if(!$user)
            {
                $this->addError($attribute, 'Пользователь не найден');
            }elseif (!$user->validatePassword($this->password))
            {
                $this->addError($attribute, 'Пароль введен неверно');
            }
        }
    }
// Получение пользователя из таблицы User
    public function getUser()
    {
        return User::findOne(['user_name'=>$this->user_name]);
    }
}

```

### Листинг 7 – Signup

```

class Signup extends Model
{
    public $user_name;
    public $password;

    // Валидация
    public function rules()
    {
        return[
            [['user_name', 'password'], 'required'],
            ['user_name', 'string', 'min'=>4],
            ['user_name', 'unique', 'targetClass'=>'app\models\User'],
            ['password', 'string', 'min'=>4, 'max'=>20]
        ];
    }

    // Создание пользователя в таблице User
    public function signup()
    {
        $user = new User;
        $user->user_name = $this->user_name;
        $user->setPassword($this->password);
    }
}

```

```
        return $user->save();
    }
}
```

## Листинг 8 – User

```
class User extends ActiveRecord implements IdentityInterface
{
    // Хэширование пароля
    public function setPassword($password)
    {
        $this->password = sha1($password);
    }
    // Валидация пароля
    public function validatePassword($password)
    {
        return $this->password=== sha1($password);
    }

    // Реализация интерфейса
    public static function findIdentity($id)
    {
        return self::findOne($id);
    }
    public function getId()
    {
        return $this->id;
    }
    public static function findIdentityByAccessToken($token, $type = null){}
    public function getAuthKey(){}
    public function validateAuthKey($authKey){}
}
```

## ПРИЛОЖЕНИЕ В

### Исходный код видов

#### Листинг 9 – Signin

```
<?php $form = ActiveForm::begin();?>

<?= $form->field($signin_model, 'user_name')->textInput()?>
<?= $form->field($signin_model, 'password')->passwordInput()?>

<div>
    <button class="btn btn-success" type="submit">Sign in</button>
</div>
<?php $form = ActiveForm::end();?>
```

#### Листинг 10 – Signup

```
<?php $form = ActiveForm::begin(['class'=>'form-horizontal']); ?>

<?= $form->field($model, 'user_name')->textInput(['autofocus'=>true]) ?>
<?= $form->field($model, 'password')->passwordInput()?>

<div>
    <button type="submit" class="btn btn-primary">Submit</button>
</div>

<?php
ActiveForm::end()
?>
```

#### Листинг 11 – Editor

```
<div class="row">
<div class="col-md-6">
<canvas id="c" width="920" height="880" style="border:1px solid black"></canvas>

</div>
<div class="col-md-2">

    <div class="modal" style="width:30px;height:30px;background:#A7A9AC;"></div>
    <!Collection of colors>
    <div class="dark" onclick="chooseColor()"
style="position:absolute;top:52px;right: 60px;">Choose Colors <i id="colorPen" class="fa
fa-pencil" style="color:#A7A9AC;"></i></div>

    <div id="colorCollection" style="position:absolute;top:75px;right: 180px;">
    <div class="rounded 15" id="#A7A9AC" onclick="color(this)"></div>
    <div class="rounded 140" id="#00AAC" onclick="color(this)"></div>
    <div class="rounded 175" id="#004DE6" onclick="color(this)"></div>
    <div class="rounded 1110" id="#3D00B8" onclick="color(this)"></div>
    <div class="rounded 1145" id="#600080" onclick="color(this)"></div>
    <div class="rounded 15 t35" id="#FFE600" onclick="color(this)"></div>
    <div class="rounded 140 t35" id="#FFAA00" onclick="color(this)"></div>
    <div class="rounded 175 t35" id="#FF5500" onclick="color(this)"></div>
    <div class="rounded 1110 t35" id="#E61B1B" onclick="color(this)"></div>
    <div class="rounded 1145 t35" id="#B31564" onclick="color(this)"></div>
```

```

        <div class="rounded 15 t70" id="#A2E61B" onclick="color(this)"></div>
        <div class="rounded 140 t70" id="#26E600" onclick="color(this)"></div>
        <div class="rounded 175 t70" id="#008055" onclick="color(this)"></div>
        <div class="rounded 1110 t70" id="#58595B" onclick="color(this)"></div>
        <div class="rounded 1145 t70" id="#613D30" onclick="color(this)"></div>
    </div>

    <!Канва рисует ширину кисти>
    <canvas id="toolCan" width="60" height="35" style="position: absolute;
    lute;top:185px;right: 5px;"></canvas>

    <input type="range" id="colorWidth" min="1" max="30" value="25" oninput="range(this)"
    style="position: absolute;top:191px;right:70px;width:105px;">

    <!Change Drawing mode>
    <button class="btn btn-secondary" id="selObj" onclick="selectObject()"
    style="position: absolute;top:220px;right:5px;width:170px;">Enter drawing
mode
    </button>

    <!Add delete predefine object>
    <select style="position: absolute;top:260px;right:44px;width:133px;"
class="custom-select" id="paintOption">
        <option value="rectangle">Rectangle</option>
        <option value="triangle">Triangle</option>
        <option value="circle">Circle</option>
        <option value="line">Line</option>
        <option value="text">Textbox</option>
    </select>

    <button class="btn btn-primary" onclick="add()" value="add" id="add"
    style="position: absolute;top:260px;right:42px;"><i id="AddIcon" class="fa
fa-plus" aria-hidden="true"></i>
    </button>

    <button class="btn btn-danger" onclick="deleteObjects()" value="delete"
id="delete"
    style="position: absolute;top:260px;right:5px;"><i id="DeleteIcon"
class="fa fa-trash" aria-hidden="true"></i>
    </button>

<?php $form = ActiveForm::begin(['id' => 'projectForm']); ?>

    <div class="btn btn-success" onclick="downloads()">Скачать</div>

    <div class="btn btn-primary" onclick="addtextbox(<?php echo $count; ?>)">Добавить
текстбокс</div>

    <div>Message: <?php echo $message; ?></div>

    <button class="btn btn-warning">Сохранить</button>
</div>
<div class="col-md-4" id="test">

<!-- Выводим текстовые поля -->
<?php

    foreach ($texts as $text) {

```

```

echo '<textarea class="form-control mb-2" rows="2" form="projectForm" id="'. $text-
>text_id . '" name="'. $text->text_id . '">' . $text->text . '</textarea>' ;
    }
    ActiveForm::end();
    ?>
</div>
</div>

```

## Листинг 12 – Index

```

<div class="row mb-5">
    <div class="col-md-3">
        <a href="/site/createproject">
            <button type="submit" class="project">
                СОЗДАТЬ ПРОЕКТ
            </button>
        </a>
    </div>
    <?php
    foreach ($projects as $project) {
    if ($counter == 3)
    {
    $counter = 0; ?>
    </div>
    <div class="row m-4">
    <?php }; ?>
    <div class="col-md-3">
        <div class="project">
            <style>
                .image-<?php echo $project->image_id; ?> {
                    background-image: url("uploads/<?php echo $project->image_path;
?>");
                    display: block;
                    height: 130px;
                    border-radius: 10px 10px 0px 0px;
                    background-size: cover;
                    background-repeat: no-repeat;
                }
            </style>
            <div class="image-<?php echo $project->image_id; ?>">
            </div>
            <div class="text">
                <?php
                echo '<h2>' . $project->project_name . '</h2>' .
                '<ul>' .
                '<li>' . 'Язык оригинала: ' . $project->lang_from_name . '</li>'
                .
                '<li>' . 'Язык перевода: ' . $project->lang_to_name . '</li>' .
                '<li>' . 'Дата создания: ' . $project->create_date . '</li>' .
                '</ul>';
                echo
                '<hr>' .

```

```

        Html::a('Редактировать', ['/site/projecteditor?project_id=' . $project-
>project_id], ['class' => 'btn btn-primary']);
        ?>
    </div>
</div>

</div>

<?php $counter++;
}?>

</div>

```

### Листинг 13 – Createproject

```

<?php $form = ActiveForm::begin(); ?>

<!-- Project name -->
<?= $form->field($create_model, 'project_name')->textInput(['autofocus'=>true]) ?>

<!-- Lang from -->
<?php
$languages = Language::find()->all();

$items = ArrayHelper::map($languages, 'lang_id', 'lang_name');
$params = [
    'prompt' => 'Выберите язык оригинала'
];
echo $form->field($create_model, 'lang_from_id')->dropDownList($items, $params);
?>

<!-- Lang to -->
<?php
$languages = Language::find()->all();

$items = ArrayHelper::map($languages, 'lang_id', 'lang_name');
$params = [
    'prompt' => 'Выберите язык перевода'
];
echo $form->field($create_model, 'lang_to_id')->dropDownList($items, $params);
?>

<div>
    <button type="submit" class="btn btn-primary">Создать</button>
</div>

<?php
ActiveForm::end()
?>

```

### Листинг 14 – Projecteditor

```

<?= $form->field($model, 'image')->fileInput() ?>
    <button>Загрузить</button>
<?php ActiveForm::end() ?>

```

```

<?php if($model->image): ?>
    
<?php endif; ?>

<?php foreach ($images as $image): ?>

    <div class="col-md-3 ">
        <div class="project">

            <div class="image2">
                
            </div>

            <div class="text">
                <p><?php echo $image->image_name . '.' . $image->type?></p>

                <hr>
                <?php echo Html::a('Перевести ', ['/editor/index?project_id=' .
$project_field->project_id . '&image_id=' . $image->image_id . '.' . $image->type ],
['class'=>'btn btn-primary']);?>

                <?php echo Html::a('Удалить ', ['/editor/index?project_id=' .
$project_field->project_id . '&image_id=' . $image->image_id . '.' . $image->type ],
['class'=>'btn btn-danger']);?>

            </div>

        </div>
    </div>
<?php endforeach; ?>
</div>

```

## ПРИЛОЖЕНИЕ Г

### Исходный код графического редактора

#### Листинг 15 – Интерфейс рисования на канве

```
$(document).ready(function () {
    $("#selObj").click(function () {
        if ($("#selObj").text() == 'Enter drawing mode') {
            $("#selObj").text('Cancel drawing mode');
            $("#paintOption").hide(400);
            $("#add").hide(400);
            $("#delete").hide(400);
            $("#searchFrn").animate({top: "260px"}, 400);
            $("#frnList").animate({top: "290px", height: "230px"}, 400);
        } else if ($("#selObj").text() == 'Cancel drawing mode') {
            $("#selObj").text('Enter drawing mode');
            $("#paintOption").show(400);
            $("#add").show(400);
            $("#delete").show(400);
            $("#searchFrn").animate({top: "300px"}, 400);
            $("#frnList").animate({top: "330px", height: "190px"}, 400);
        }
    });
});
var x = "#A7A9AC",
    y = 24;
tcan = document.getElementById('toolCan');
tctx = tcan.getContext("2d");
tctx.clearRect(0, 0, 60, 35);
tctx.beginPath();
tctx.moveTo(12, 35 / 2);
tctx.lineTo(48, 35 / 2);
tctx.strokeStyle = document.getElementById('colorPen').style.color;
tctx.lineWidth = 24;
tctx.lineCap = 'round';
tctx.stroke();

// Выбрать цвет
function color(obj) {
    x = obj.id
    y = document.getElementById('colorWidth').value;
    document.getElementById('colorPen').style.color = x;
    document.getElementById('colorWidth').style.display = "inline";

    canvas.freeDrawingBrush.color = x;
    var checkObj = canvas.getActiveObject();
    if (checkObj) {
        if (checkObj.get('type') != 'path') checkObj.set("fill", x);
        checkObj.set("stroke", x);
        canvas.renderAll();
    }

    tctx.clearRect(0, 0, 60, 35);
    tctx.beginPath();
    tctx.moveTo(y / 2, 35 / 2);
    tctx.lineTo(60 - y / 2, 35 / 2);
    tctx.strokeStyle = x;
    tctx.lineWidth = y;
    tctx.lineCap = 'round';
    tctx.stroke();
}
```

```

// Создать объект
function add() {
    var paintOpt = $("#paintOption").val();

    switch (paintOpt) {
        case 'rectangle':
            var rectangle = new fabric.Rect({
                width: 100,
                height: 70,
                fill: x,
                left: 50,
                top: 50
            });
            canvas.add(rectangle);
            break;
        case 'triangle':
            var triangle = new fabric.Triangle({
                width: 100,
                height: 75,
                fill: x,
                left: 250,
                top: 50
            });
            canvas.add(triangle);
            break;
        case 'circle':
            var circle = new fabric.Circle({
                radius: 50,
                fill: x,
                left: 450,
                top: 50
            });
            canvas.add(circle);
            break;
        case 'line':
            var line = new fabric.Line([50, 100, 200, 100], {
                left: 650,
                top: 75,
                stroke: x,
                strokeWidth: 8
            });
            canvas.add(line);
            break;
        case 'text':
            var addtext = new fabric.Textbox('Edit this text', {
                left: 400,
                top: 200,
                fill: x,
                strokeWidth: 2,
                fontFamily: 'Arial'
            });
            canvas.add(addtext);
            break;
        default:
            alert('No');
    }
}

// Выбрать объект
function selectObject() {
    canvas.isDrawingMode = !canvas.isDrawingMode;
}

```

```

// Удалить объект
function deleteObjects() {
    var active = canvas.getActiveObjects();
    if (active) {
        canvas.discardActiveObject();
        canvas.remove(...active);
    }
}

// Скачать все
function downloads() {
    var canvas = document.getElementById("c");
    canvas.toBlob(function(blob) {
        saveAs(blob, "pretty image.png");
    });
}

// Создать текстбокск
var index = -1;
function addtextbox(textsCount) {
    if(index === -1) index = textsCount-1;
    index++;

    var elem = document.createElement("textarea");
    elem.setAttribute("class", "form-control mb-2");
    elem.setAttribute("id", index);
    elem.setAttribute("name", index);
    elem.setAttribute("form", "projectForm");
    document.getElementById("test").appendChild(elem);
}

var canvas = this.__canvas = new fabric.Canvas('c');
// Передаем редактируемую картинку
var path = '<?php echo $path;?>';
fabric.Image.fromURL('../uploads/'+path, function(img) {
    img.scale(1).set({});
    canvas.add(img);//.setActiveObject(img)
});

if (canvas.freeDrawingBrush) {
    canvas.freeDrawingBrush.color = x;
    canvas.freeDrawingBrush.width = y;
}

function range(tobj) {
    y = tobj.value;
    tctx.clearRect(0, 0, 60, 35);
    tctx.beginPath();
    tctx.moveTo(y / 2, 35 / 2);
    tctx.lineTo(60 - y / 2, 35 / 2);
    tctx.lineWidth = y;
    tctx.lineCap = 'round';
    tctx.stroke();
    canvas.freeDrawingBrush.width = y;
}

```