Федеральное государственное образовательное бюджетное учреждение высшего образования

**«Финансовый университет при Правительстве Российской Федерации»**

**(Финансовый университет)**

Колледж информатики и программирования

Выпускная квалификационная работа

на тему: «

Студент группы 4ИСИП-619 Ефимов

*(№ учебной группы) (фамилия имя отчество полностью) (подпись)*

Образовательная программа

09.02.07 Информационные системы и программирование

*(индекс и наименование специальности)*

Форма обучения очная

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ И. В. Сибирев

*(подпись) (И.О. Фамилия)*

Консультант \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  *(при наличии) (подпись) (И.О. Фамилия)*

Председатель ПЦК \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  А. И. Пестов  *(подпись) (И.О. Фамилия)*

Москва – 2023

Федеральное государственное образовательное бюджетное учреждение высшего образования

**«Финансовый университет при Правительстве Российской Федерации»**

**(Финансовый университет)**

Колледж информатики и программирования

УТВЕРЖДАЮ

Председатель ПЦК

Программирования и баз данных

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_А.И. Пестов

*(подпись) (И.О.Фамилия)*

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_ 20 \_\_ г.

ЗАДАНИЕ

на выпускную квалификационную работу

обучающемуся Зайцеву Кириллу Максимовичу

*(фамилия имя отчество)*

1. Тема выпускной квалификационной работы

«Система «Интернет магазин и складской учёт автомобильных деталей»

1. Срок сдачи обучающимся законченной выпускной квалификационной работы «\_05\_» \_\_\_\_\_июня\_\_\_\_ 2022 г.
2. Исходные данные
   1. Требования к структуре сайта «Интернет магазин и складской учёт автомобильных деталей»;
   2. Требования к дизайну сайта «Интернет магазин и складской учёт автомобильных деталей»;
   3. Требования к функционалу сайта «Интернет магазин и складской учёт автомобильных деталей».
3. Перечень подлежащих разработке задач/вопросов:

4.1 Проведение предпроектного исследования предметной области, аналогичных сервисов для понимания потребностей пользователей сайта «Интернет магазин и складской учёт автомобильных деталей».

4.2 Обзор технологий, позволяющих реализовать сайты;

4.3 Выбор и обоснование инструментальных средств и технологий реализации сайта.

4.4 Разработка алгоритма сайта.

4.5 Разработка интерфейса сайта

4.6 Разработка тестовых наборов и сценариев для проверки работоспособности сайта.

1. Перечень графического/ иллюстративного/ практического материала:

5.1 Структура (логическая схема) сайта

5.2 Исходные коды сайта

5.3 Презентация выпускной квалификационной работы

1. Консультант по выпускной квалификационной работе с указанием относящихся к ним разделов работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата выдачи задания « 13» октября 2021 г.

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ И.В. Сибирев

*(подпись) (И.О. Фамилия)*

Задание принял к исполнению «13» \_\_\_октября\_ 2021г.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(подпись обучающегося)*

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc134373766)

[ГЛАВА 1. ПРЕДПРОЕКТНОЕ ИССЛЕДОВАНИЕ 5](#_Toc134373767)

[1.1. Описание предметной области 5](#_Toc134373768)

[1.2. Сравнительный анализ программ аналогов 5](#_Toc134373769)

[1.3. Постановка задачи 8](#_Toc134373770)

[1.4. Характеристика инструментальных средств 8](#_Toc134373771)

[1.5. Обоснование выбора инструментальных средств разработки 10](#_Toc134373772)

[ГЛАВА 2. ПРАКТИЧЕСКАЯ ЧАСТЬ 12](#_Toc134373773)

[2.1. Анализ требований и определение спецификаций ПО 12](#_Toc134373774)

[2.2. Проектирование программного обеспечения 15](#_Toc134373775)

[2.3. Разработка программного обеспечения 16](#_Toc134373776)

[2.4. Отладка и тестирование программы 30](#_Toc134373777)

[2.5. Руководство по использованию программы 31](#_Toc134373778)

[ЗАКЛЮЧЕНИЕ 34](#_Toc134373779)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 35](#_Toc134373780)

[ПРИЛОЖЕНИЕ 36](#_Toc134373781)

# ВВЕДЕНИЕ

В современном мире люди всё чаще сталкиваются с необходимостью общаться на разных языках. Связанное с этим повышение спроса на услуги переводчиков определяет важность создания новых инструментов, которые помогут людям переводить тексты быстрее и эффективнее, а мультиплатформенность переводчика позволит пользователям получать перевод на разных устройствах и в любой ситуации. К тому же функция распознавания текста становится все более востребованной, так как даёт возможность легко переводить текст, который находится вне Интернета, например, фотографируя текст на бумажном документе. Исходя из вышеизложенного, можно заключить, что разработка мультиплатформенного переводчика с распознаванием текста является актуальной задачей, которая будет полезна для широкой аудитории пользователей в современном мире.

Целью проекта является разработка мультиплатформенного переводчика с распознаванием текста.

Для достижения поставленной цели были определены следующие задачи:

* проанализировать существующие мультиплатформенные переводчики и системы распознавания текста;
* определить инструментальную базу для разработки приложения;
* изучить требования и спецификацию ПО;
* разработать архитектуру мультиплатформенного переводчика с распознаванием текста;
* реализовать разработанную архитектуру;
* провести отладку и тестирование программы;
* разработать руководство по использованию программы.

Объектом дипломного проекта являются инструменты и технологии разработки программного обеспечения переводчиков с распознаванием текста.

Предмет проекта – мультиплатформенное приложение-переводчик.

Одно из требований к разрабатываемому переводчику – его мультиплатформенность. Это означает, что он совместим с различными операционными системами, такими как Windows и Android, что значительно расширяет круг потенциальных пользователей.

Для поставленных задач будут применяться следующие инструментальные средства, языки проектирования и разработки программного обеспечения:

* язык программирования C#;
* язык программирования Kotlin;
* интегрированная среда разработки Visual Studio 2022;
* интегрированная среда разработки Android Studio;
* компактная встраиваемая СУБД SQLite;
* сервис для проектирования диаграмм Draw.io;
* текстовый редактор Microsoft Word;
* программа подготовки и просмотра презентаций Microsoft PowerPoint.

Процессы разработки мультиплатформенного переводчика с распознаванием текста будут описываться с использованием .net framework для desktop-приложения и Android SDK для мобильного приложения. В данной работе будут использованы следующие инструменты:

* Visual Studio(.net framework) для desktop-приложения;
* API Google Переводчика;
* Android studio для мобильного приложения;
* ML Kit для машинного перевода текста
* Методы распознавания текста.

# **ГЛАВА 1. ПРЕДПРОЕКТНОЕ ИССЛЕДОВАНИЕ**

## 1.1. Описание предметной области

Сущностью предметной области данной дипломной работы выступает приложение–переводчик для операционной системы Windows, основная функция которого – перевод текста на один из выбранных языков. Пользователями приложения могут быть как частные лица, так и представители бизнес-среды. Для перевода можно использовать тексты частной переписки мессенджеров и электронной почты, материалы сайтов, тексты документов, и другие типы текстов.

Переводчик предоставляет возможность людям понимать друг друга, даже если они являются носители разных языков, благодаря чему языковой барьер между ними может быть преодолен.

Крайне актуальным функционалом представляется оперативный перевод текста с изображений, что сокращает время в процессе перевода и не требует отдельного ввода текста вручную. Таким образом, можно сделать вывод, что приложение должно включать в себя следующие функции:

* перевод текста;
* распознавание текста с изображения.

## 1.2. Сравнительный анализ программ-аналогов

Для проведения сравнительного анализа были выбраны несколько программ-аналогов, которые имеют похожие функции и предназначены для использования на различных платформах. В данном разделе будет проведен анализ каждой из этих программ по нескольким критериям.

* Качество перевода: будет проанализировано качество перевода каждой программы, с учетом точности перевода и способности распознавать различные языки.
* Удобство использования: будет оценено удобство использования каждой программы, включая интерфейс пользователя и доступность функций.
* Совместимость: будет рассмотрена совместимость каждой программы с различными операционными системами и устройствами.
* Стоимость: будет оценена стоимость каждой программы, включая цену за использование и возможные дополнительные расходы.
* Дополнительные функции: будет рассмотрено наличие дополнительных функций в каждой программе, таких как возможность использования голосового ввода или функций машинного обучения.

Программный продукт «Dicter»

Онлайн-переводчик Dicter распространяется бесплатно и способен не только к переводу введенного текста, но и любого выделенного. Достаточно использовать «горячие клавиши» и через мгновение увидишь перевод. Просматривая сайты на иностранном малознакомом языке очень сложно обойтись без бесплатного онлайн-переводчика.

Приложение генерирует достаточно точный перевод. Он выполняется быстро для текста небольшого объёма. Программа компактная и удобная в использовании, совместима со всеми версиями Windows, начиная с Windows7. К тому же приложение предоставляется пока бесплатно.

К приятным дополнительным функциям можно отнести поддержку «Горячих клавиш», звуковое воспроизведение текста.

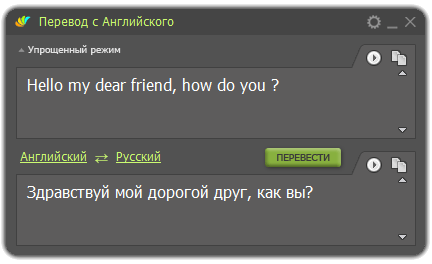


Рисунок 1. Скриншот программы-аналога «Dicter»

Программный продукт «EveryLang»

EveryLang - это программа для перевода текста и распознавания речи на различных языках, доступная для использования на Windows и Android.

Основной функцией EveryLang является перевод текста с одного языка на другой. Программа поддерживает более 100 языков, включая многие редкие языки.

В приложении скорость и точность перевода зависит от объема текста. Программа компактная и удобная в использовании и совместима с Windows и Android. Есть несколько версий приложения: бесплатная; стандартная (стоит около 15$); про-версия (стоит около 25$) и ультимейт (стоит около 45$).

К дополнительным функциям можно отнести возможность пользоваться оф-лайн, поддержка «горячих клавиш», определение языка исходного текста, воспроизведение переводимого и переведённого текста.

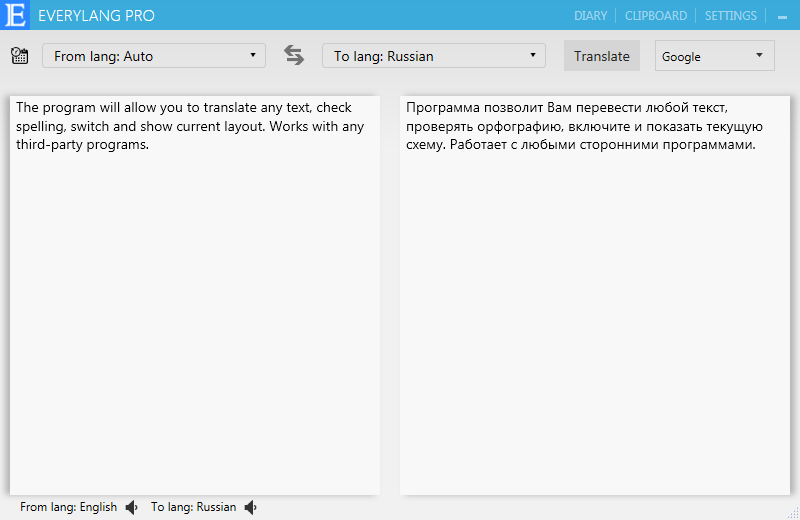


Рисунок 2. Скриншот программы-аналога «EveryLang»

Хорошо бы сделать про свое – нечто похожее на следующий текст

После проведения сравнительного анализа аналогов, перечисленных выше, были выявлены плюсы и недостатки каждого из них, результаты сравнительного анализа представлены в таблице 1.

Таблица 1. Результаты сравнительного анализа

|  |  |  |  |
| --- | --- | --- | --- |
| Название сайта:  Функции: | Авито | Фарпост | Дром |
| Регистрация | + | + | + |
| Авторизация | + | + | + |
| Поиск по названию | + | + | + |
| Поиск по модели авто | - | + | + |
| Поиск по категории | - | + | + |
| Корзина | - | +/- | +/- |
| Множественный выбор | - | + | + |

На основании результатов проведённого анализа функционала можно сделать вывод о том, что разрабатываемый проект ~~по набору функциональных особенностей не будет уступать аналогам.~~ Чем будет отличаться от аналогов? В чем его преимущество?

## . Постановка задачи

При первом запуске приложения пользователя будет отображаться главное окно, которое будет разделено на 2 части. Справа будет находиться главная страница, на которой будет функция перевода, слева – меню навигации, в котором будет доступно 3 раздела: раздел «История»; раздел «Текст с фото»; раздел «Настройки».

В разделе «История» пользователю будет отображаться страница, на которой будет история переводов и доступные языки. Историю переводов он сможет просматривать и/или удалять ненужную. Также будут показываться доступные языки, которые пользователь сможет также удалить.

В разделе «Текст с фото» пользователю отображается страница, которая будет содержать в себе выбранное пользователем изображение, и кнопку «распознать», с помощью которой будет распознаваться текст с изображения, выбор языка, совпадающий с языком текста на изображении, а также кнопка «перевести», с помощью которой распознанный текст с изображения можно перевести на нужный язык.

В разделе настроек пользователю будет доступно изменение размера шрифта, а также кнопка «добавить язык», при нажатии на которую будет открываться дополнительное окно, в котором пользователь будет вписывать название языка и код ISO 639-1 этого языка, а также появится ссылка на все языки, доступные для перевода, и их код ISO 639-1.

Эксплуатационные требования для разрабатываемого приложения:

операционная система – Windows 10;

минимальная система – Windows 7;

требуемые разрешения – доступ в Интернет.

## 1.4. Характеристика инструментальных средств

C# — объектно-ориентированный язык программирования общего назначения. Разработан в 1998—2001 годах группой инженеров компании Microsoft под руководством Андерса Хейлсберга и Скотта Вильтаумота как язык разработки приложений для платформы Microsoft .NET Framework и .NET Core. Впоследствии был стандартизирован как ECMA-334 и ISO/IEC 23270. C# относится к семье языков с C-подобным синтаксисом, из них его синтаксис наиболее близок к C++ и Java. Язык имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов (в том числе операторов явного и неявного приведения типа), делегаты, атрибуты, события, переменные, свойства, обобщённые типы и методы, итераторы, анонимные функции с поддержкой замыканий, LINQ, исключения, комментарии в формате XML. Переняв многое от своих предшественников — языков C++, Delphi, Модула, Smalltalk и, в особенности, Java — С#, опираясь на практику их использования, исключает некоторые модели, зарекомендовавшие себя как проблематичные при разработке программных систем, например, C# в отличие от C++ не поддерживает множественное наследование классов (между тем допускается множественная реализация интерфейсов).

Microsoft Visual Studio — линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментов. Данные продукты позволяют разрабатывать как консольные приложения, так и игры и приложения с графическим интерфейсом, в том числе с поддержкой технологии Windows Forms, UWP а также веб-сайты, веб-приложения, веб-службы как в родном, так и в управляемом кодах для всех платформ, поддерживаемых Windows, Windows Mobile, Windows CE, .NET Framework, .NET Core, .NET, MAUI, Xbox, Windows Phone .NET Compact Framework и Silverlight. После покупки компании Xamarin корпорацией Microsoft появилась возможность разработки IOS и Android программ.

Visual Studio включает в себя редактор исходного кода с поддержкой технологии IntelliSense и возможностью простейшего рефакторинга кода.

Draw.io — это бесплатное кроссплатформенное программное обеспечение для рисования диаграмм с открытым исходным кодом, разработанное на HTML5 и JavaScript. Его интерфейс можно использовать для создания диаграмм, таких как блок-схемы, каркасы, диаграммы UML, организационные диаграммы и сетевые диаграммы.

draw.io доступна как онлайн в виде кросс-браузерного веб-приложения, так и в виде автономного настольного приложения для Linux, macOS и Windows.

MicrosoftWord (часто — MSWord,WinWord или просто Word) — текстовый процессор, предназначенный для создания, просмотра, редактирования и форматирования текстов статей, деловых бумаг, а также иных документов, с локальным применением простейших форм таблично-матричных алгоритмов. Выпускается корпорацией Microsoft в составе пакета Microsoft Office. Первая версия была написана Ричардом Броди (Richard Brodie) для IBM PC, использующих DOS, в 1983 году. Позднее выпускались версии для Apple Macintosh (1984), SCO UNIX и Microsoft Windows (1989). Текущей версией является Microsoft Office Word 2021 для Windows и macOS, а также веб-версия Word Online (Office Online), не требующая установки программы на компьютер.

PowerPoint (полное название — Microsoft Office PowerPoint, от англ. power point — убедительный доклад) — программа подготовки презентаций и просмотра презентаций, являющаяся частью Microsoft Office и доступная в редакциях для операционных систем Microsoft Windows и macOS, а также для мобильных платформ Android и IOS. Материалы, подготовленные с помощью PowerPoint, предназначены для отображения на большом экране — через проектор, либо телевизионный экран большого размера.

## 1.5 Обоснование выбора инструментальных средств разработки

Для разработки мультиплатформенного переводчика с распознаванием текста были выбраны инструменты и технологии, которые позволяют решать поставленные задачи и соответствуют требованиям проекта. Основные требования к проекту включают разработку приложения, работающего на нескольких платформах, высокую точность распознавания текста и качество перевода.

Для разработки приложения на нескольких платформах были выбраны такие фреймворки как: .Net Framework и Android SDK. .Net Framework был выбран из-за поддержки языка C#, на котором будет реализована desktop версия приложения «KIP Translator». Android SDK – из-за поддержки языка Kotlin для мобильной версии приложения.

Фреймворки .NET Framework и Android SDK были выбраны из-за доступности для разработчиков с разным уровнем опыта и знаний, что является важным фактором, который определил выбор данных инструментов. .NET Framework является одним из наиболее популярных инструментов для разработки приложений под Windows, обладает большим сообществом пользователей и имеет обширную документацию. Android SDK, в свою очередь, является стандартным комплектом разработки для Android-приложений и имеет множество учебных материалов и документации, что облегчает разработку для этой платформы.

В связи с тем, что фреймворки .NET Framework и Android SDK являются инструментами разработки в средах Vusial Studio и Android Studio, то для реализации проекта были выбраны именно эти среды разработки.

# **ГЛАВА 2. ПРАКТИЧЕСКАЯ ЧАСТЬ**

## 2.1 Анализ требований и определение спецификаций ПО

Спецификации данного проекта определяются следующими диаграммами из списка:

* диаграмма «Сущность-связь» (ER-диаграмма);
* диаграмма классов;
* диаграмма вариантов использования;
* детализированная диаграмма потоков данных верхнего уровня;
* детализированная диаграмма потоков данных;
* краткая функциональная диаграмма.

На рисунке 3 представлена диаграмма «сущность-связь» дипломного проекта по теме «Разработка мультиплатформенного переводчика с распознаванием текста», построенная с использованием Microsoft SQL Server Management Studio.

Изображение выглядит как диаграмма

Автоматически созданное описание

Рисунок 3. Диаграмма «сущность-связь»

На рисунке 4 представлена диаграмма классов дипломного проекта по теме «KIP Translator».

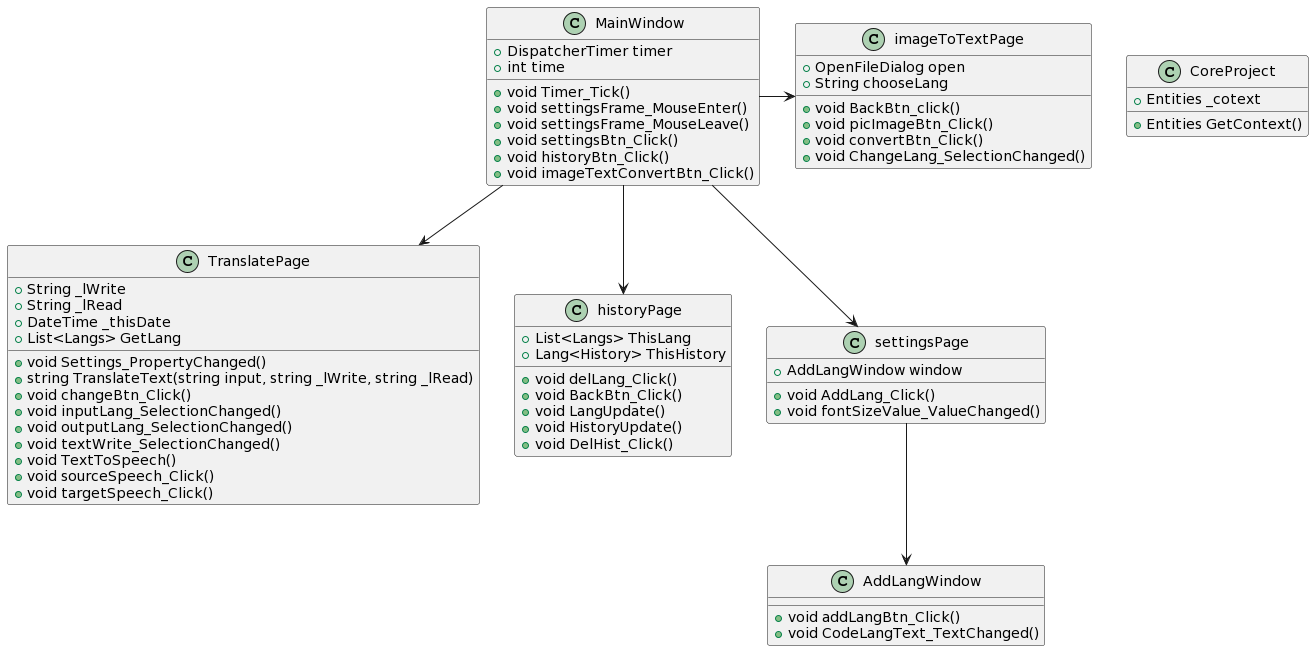


Рисунок 4. Диаграмма классов

Диаграмма вариантов использования (сценариев поведения, прецедентов) является исходным концептуальным представлением системы в процессе ее проектирования и разработки. Данная диаграмма состоит из актеров, вариантов использования и отношений между ними. При построении диаграммы могут использоваться также общие элементы нотации: примечания и механизмы расширения. (рисунок 5)

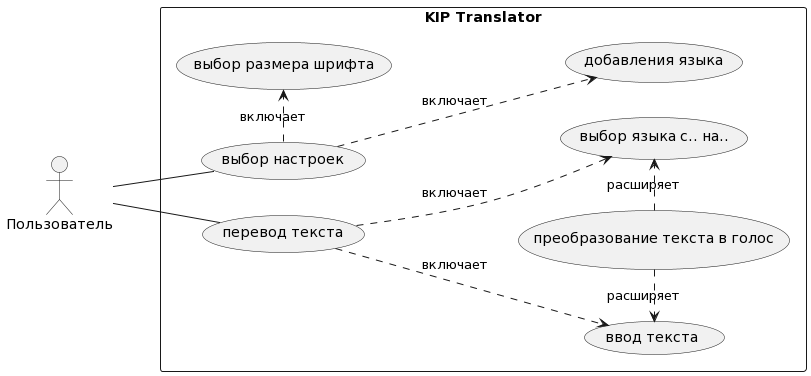


Рисунок 5. Диаграмма вариантов использования.

На рисунке 6 изображена диаграмма потоков данных верхнего уровня для программного обеспечения предметной области «KIP Translator»

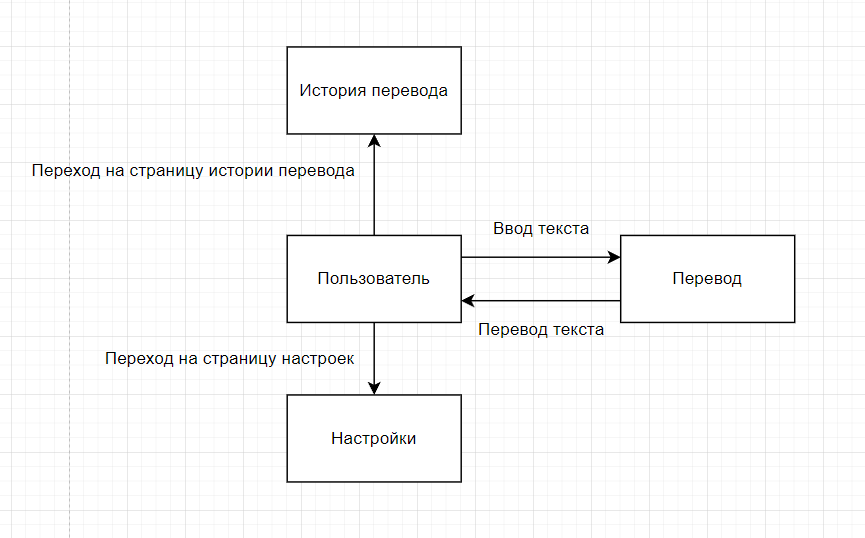


Рисунок 6. Диаграмма потоков данных верхнего уровня

На рис. 7 изображена детализированная диаграмма потоков данных для программного обеспечения предметной области «KIP Translator».



Рисунок 7. Детализированная диаграмма потоков данных

На рисунке 8 представлена краткая функциональная схема.

**Изображение выглядит как диаграмма

Автоматически созданное описание**

Рисунок 8. Краткая функциональная диаграмма

## 2.2 Проектирование программного обеспечения

На рисунке 9 изображена структурная схема программного продукта «KIP Translator».

Изображение выглядит как диаграмма

Автоматически созданное описание

Рисунок 9. Структурная схема

Ниже представлена функциональная схема приложения «KIP Translator».

Изображение выглядит как диаграмма

Автоматически созданное описание

Рисунок 10. Функциональная схема

## 2.3 Разработка программного обеспечения

Программный продукт «Kip Translator» это мультиплатформенное приложение-переводчик, разработанное на базе платформ android и windows. Среди функций «Kip Translator» можно выделить основную функцию – перевод текст на разные языки, а также дополнительные функции, такие как: сохранение истории переводов, озвучивание текста (PC), распознавание текста с изображения (PC), преобразование голоса в текст(android), добавление дополнительных языков перевода и некоторые другие.

При запуске приложения «KIP Translator» выводится главное окно, которое разделено на две части. На левой части представлены три кнопки, предназначенные для навигации по приложению, а на правой части - страницы приложения (рисунок 11).

Кнопка «История» при нажатии переводит пользователя на страницу сохраненного перевода.

Кнопка «Текст с фото» для перехода на страницу распознавания текста с изображения.

Функции добавление языка и изменение размера шрифта доступны в меню «настройки» (Кнопка «Настройки»).

Представленные на главной странице (рисунок 12) TextBox предназначены для ввода теста: левый – для ввода переводимого текста, а правый – для вывода переведённого текста.

К числу элементов, представленных на главной странице, относятся ComboBox для смены языка перевода и кнопки озвучивания текста. ComboBox содержит в себе все языки доступные для перевода текста программой.

Изображение выглядит как снимок экрана, текст, Прямоугольник, программное обеспечение

Автоматически созданное описание

Рисунок 11. Главное окно приложения

Ниже представлены вёрстка и код окна:

<Grid Background="#51b6a2">

<Grid.RowDefinitions>

<RowDefinition Height="19\*"/>

<RowDefinition Height="102\*"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="3\*"/>

<ColumnDefinition Width="22\*"/>

</Grid.ColumnDefinitions>

<Frame x:Name="nextPage" NavigationUIVisibility="Hidden" Background="#ead9cd" Source="/Pages/TranslatorPage.xaml" Margin="0,76,0,0" Grid.Column="1" Grid.RowSpan="2"/>

<Image Source="/Images/TranslatorLogo.png" Margin="10,10,10,10"/>

<StackPanel Grid.Row="1">

<Button x:Name="historyBtn" Content="История" Margin="5,10,5,10" Click="historyBtn\_Click"/>

<Button x:Name="imageConvertBtn" Content="Текст с фото" Margin="5,0,5,10" Click="imageConvertBtn\_Click"/>

<Button x:Name="settingsBtn" Content="Настройки" Margin="5,0,5,10" Click="settingsBtn\_Click"/>

<Frame x:Name="settingsFrame" NavigationUIVisibility="Hidden" Source="Pages/settingsPage.xaml" Visibility="Hidden" Margin="5,0,5,0" MouseEnter="settingsFrame\_MouseEnter" MouseLeave="settingsFrame\_MouseLeave"/>

</StackPanel>

</Grid>

DispatcherTimer timer = new DispatcherTimer();

public MainWindow()

{

InitializeComponent();

timer.Tick += Timer\_Tick;

timer.Interval = new TimeSpan(0, 0, 1);

}

int time = 0;

private void Timer\_Tick(object sender, EventArgs e)

{

time++;

if (time == 5)

{

settingsFrame.Visibility = Visibility.Hidden;

time = 0;

timer.Stop();

}

}

private void settingsFrame\_MouseEnter(object sender, MouseEventArgs e)

{

timer.Stop();

time = 0;

}

private void settingsFrame\_MouseLeave(object sender, MouseEventArgs e)

{

timer.Start();

}

private void settingsBtn\_Click(object sender, RoutedEventArgs e)

{

settingsFrame.Visibility = Visibility.Visible;

}

private void historyBtn\_Click(object sender, RoutedEventArgs e)

{

nextPage.NavigationService.Navigate(new Pages.historyPage());

}

private void imageConvertBtn\_Click(object sender, RoutedEventArgs e)

{

nextPage.NavigationService.Navigate(new Pages.imageToTextPage());

}

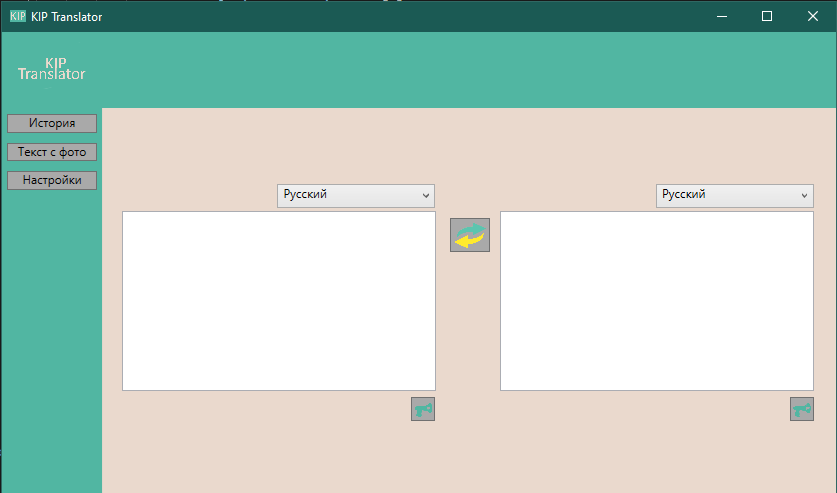


Рисунок 12. Главное страница приложения

Ниже представлены вёрстка и код страницы:

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="72\*"/>

<RowDefinition Height="28" MaxHeight="28" MinHeight="28"/>

<RowDefinition Height="188\*"/>

<RowDefinition Height="95\*"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="167\*"/>

<ColumnDefinition Width="25\*"/>

<ColumnDefinition Width="166\*"/>

</Grid.ColumnDefinitions>

<TextBox x:Name="textWrite" TextWrapping="Wrap" AcceptsReturn="True" Margin="20,3,8,6" Grid.Row="2" PreviewKeyDown="textWrite\_PreviewKeyDown" />

<TextBox x:Name="textRead" TextWrapping="Wrap" AcceptsReturn="True" Margin="4,3,22,6" Grid.Column="2" Grid.Row="2" />

<Button x:Name="changeBtn" VerticalAlignment="Top" Margin="0,10,0,0" Height="34" Width="40" Grid.Column="1" Grid.Row="2" Click="changeBtn\_Click">

<Image Source="/Images/changeLang.png"/>

</Button>

<ComboBox x:Name="inputLang" HorizontalAlignment="Right" Margin="0,4,9,0" Grid.Row="1" SelectionChanged="inputLang\_SelectionChanged" MinWidth="158">

<ComboBox.ItemTemplate>

<DataTemplate>

<TextBlock Text="{Binding NameLang}"/>

</DataTemplate>

</ComboBox.ItemTemplate>

</ComboBox>

<ComboBox x:Name="outputLang" HorizontalAlignment="Right" Margin="0,4,22,0" Grid.Column="2" Grid.Row="1" SelectionChanged="outputLang\_SelectionChanged" MinWidth="158">

<ComboBox.ItemTemplate>

<DataTemplate>

<TextBlock Text="{Binding NameLang}"/>

</DataTemplate>

</ComboBox.ItemTemplate>

</ComboBox>

<Button Name="sourceSpeech" Click="sourceSpeech\_Click" HorizontalAlignment="Right" Margin="0,0,9,0" Grid.Row="3" VerticalAlignment="Top">

<Image Source="/Images/speech.png" Width="20" Height="20" RenderTransformOrigin="0.5,0.625"/>

</Button>

<Button Grid.Column="2" Name="targetSpeech" Click="targetSpeech\_Click" HorizontalAlignment="Right" Margin="0,0,22,0" Grid.Row="3" VerticalAlignment="Top">

<Image Source="/Images/speech.png" Width="20" Height="20"/>

</Button>

</Grid> private string \_lWrite;

private string \_lRead;

private DateTime \_thisDate;

public List<Lang> GetLang { get; set; }

SpeechSynthesizer synthesizer = new SpeechSynthesizer();

public TranslatorPage()

{

InitializeComponent();

DataContext = this;

LoadData();

\_thisDate = DateTime.Today;

\_thisDate.ToShortDateString();

Properties.Settings.Default.PropertyChanged += Settings\_PropertyChanged;

textWrite.ToolTip = "Для перевода текста нажмите 'Enter'\nА для перехода на следующую строку используйте 'Shift+Enter'";

}

private void Settings\_PropertyChanged(object sender, System.ComponentModel.PropertyChangedEventArgs e)

{

textWrite.FontSize = Properties.Settings.Default.fontSizeValue;

textRead.FontSize = Properties.Settings.Default.fontSizeValue;

}

public string TranslateText(string input, string \_lWrite, string \_lRead)

{

try

{

string url = String.Format

("https://translate.googleapis.com/translate\_a/single?client=gtx&sl={0}&tl={1}&dt=t&q={2}",

\_lWrite, \_lRead, Uri.EscapeUriString(input));

HttpClient httpClient = new HttpClient();//создание нового экземпляра HTTP-запроса

string result = httpClient.GetStringAsync(url).Result;// получение json

var jsonData = new JavaScriptSerializer().Deserialize<List<dynamic>>(result);

var translationItems = jsonData[0];

string translation = "";

foreach (object item in translationItems)//обработка запроса (выявление текста из запроса)

{

IEnumerable translationLineObject = item as IEnumerable;

IEnumerator translationLineString = translationLineObject.GetEnumerator();

translationLineString.MoveNext();

translation += string.Format(" {0}", Convert.ToString(translationLineString.Current));

}

if (translation.Length > 1) { translation = translation.Substring(1); };

return translation;

}

catch

{

return "";

}

}

private void changeBtn\_Click(object sender, System.Windows.RoutedEventArgs e)

{

var item = inputLang.SelectedItem;

inputLang.SelectedItem = outputLang.SelectedItem;

outputLang.SelectedItem = item;

var text = textWrite.Text;

textWrite.Text = textRead.Text;

textRead.Text = text;

}

private void inputLang\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

if (inputLang.SelectedItem != null)

{

\_lWrite = (inputLang.SelectedItem as Lang).CodeLang;

}

}

private void outputLang\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

if (outputLang.SelectedItem != null)

{

\_lRead = (outputLang.SelectedItem as Lang).CodeLang;

}

}

private void textWrite\_PreviewKeyDown(object sender, System.Windows.Input.KeyEventArgs e)

{

string result;

if (e.Key == Key.Enter && (Keyboard.IsKeyDown(Key.LeftShift) || Keyboard.IsKeyDown(Key.RightShift)))

{

textWrite.Text += Environment.NewLine;

textWrite.CaretIndex = textWrite.Text.Length;

e.Handled = true;

}

else if (e.Key == Key.Enter)

{

result = TranslateText(textWrite.Text, \_lWrite, \_lRead);

if (!String.IsNullOrEmpty(result))

{

textRead.Text = result;

CoreProject.RunNonQuery($"INSERT INTO History(TranslateSource, TranslateTarget, Date, IdLangIn, IdLangOut) VALUES(\'{textWrite.Text}\', \'{textRead.Text}\', \'{\_thisDate}\', \'{(inputLang.SelectedItem as Lang).Id}\', \'{(outputLang.SelectedItem as Lang).Id}\')");

}

else { MessageBox.Show("Внимание!\n", "ПРЕДУПРЕЖДЕНИЕ", MessageBoxButton.OK, MessageBoxImage.Warning); }

e.Handled = true;

}

}

private void sourceSpeech\_Click(object sender, RoutedEventArgs e)

{

if (synthesizer == null)

{

synthesizer = new SpeechSynthesizer();

}

if (synthesizer.State == SynthesizerState.Speaking)

{

synthesizer.Pause();

}

else

{

string text = textWrite.Text;

if (!string.IsNullOrEmpty(text))

{

synthesizer.SpeakAsync(text);

}

}

}

private void targetSpeech\_Click(object sender, RoutedEventArgs e)

{

if (synthesizer == null)

{

synthesizer = new SpeechSynthesizer();

}

if (synthesizer.State == SynthesizerState.Speaking)

{

synthesizer.Pause();

}

else

{

string text = textRead.Text;

if (!string.IsNullOrEmpty(text))

{

synthesizer.SpeakAsync(text);

}

}

}

private void LoadData()

{

GetLang = CoreProject.RunQueryList<Lang>("SELECT \* FROM Lang");

inputLang.ItemsSource = GetLang;

outputLang.ItemsSource = GetLang;

inputLang.SelectedIndex = 0;

outputLang.SelectedIndex = 0;

}

private void Page\_Loaded(object sender, RoutedEventArgs e)

{

LoadData();

if (Global.text != null || Global.text != string.Empty)

{

textWrite.Text = Global.text;

Global.text = string.Empty;

}

}

На рисунке 13 изображена страница истории переводов, отображающая сохранённую историю переводов и список доступных языков в приложении.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 13. Страница истории переводов

Ниже представлены вёрстка и код страницы:

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="53" MaxHeight="53" MinHeight="53"/>

<RowDefinition Height="397\*"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="621\*"/>

<ColumnDefinition Width="225\*"/>

</Grid.ColumnDefinitions>

<ListView x:Name="historyList" ScrollViewer.VerticalScrollBarVisibility="Disabled" d:ItemsSource="{d:SampleData}" Grid.Row="1">

<ListView.ItemTemplate>

<DataTemplate>

<Grid>

<WrapPanel Margin="0,0,0,10">

<TextBlock Text="{Binding Date, StringFormat=d}"/>

<Rectangle Width="10"/>

<WrapPanel>

<StackPanel>

<TextBlock Text="{Binding LangIn.NameLang}"/>

<TextBlock Text="{Binding TranslateSource}"/>

</StackPanel>

<StackPanel>

<Rectangle Width="40"/>

</StackPanel>

<StackPanel>

<TextBlock Text="{Binding LangOut.NameLang}"/>

<TextBlock Text="{Binding TranslateTarget}"/>

</StackPanel>

</WrapPanel>

</WrapPanel>

</Grid>

</DataTemplate>

</ListView.ItemTemplate>

</ListView>

<DataGrid x:Name="LangList" IsReadOnly="True" CanUserAddRows="False" AutoGenerateColumns="False" VerticalScrollBarVisibility="Hidden" Grid.Column="1" Grid.Row="1">

<DataGrid.Columns>

<DataGridTextColumn Header="Язык" Binding="{Binding NameLang}"/>

<DataGridTemplateColumn>

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button x:Name="delLang" Click="delLang\_Click">

<Image Source="/Images/trash.png" Width="20" Height="20"/>

</Button>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

</DataGrid.Columns>

</DataGrid>

<Button Grid.Column="1" Name="BackBtn" Click="BackBtn\_Click" Content="Назад" Margin="20,10,20,10" />

<Button x:Name="DelHist" Click="DelHist\_Click" HorizontalAlignment="Right" Margin="0,0,10,0" VerticalAlignment="Center" Grid.Column="0" Height="24" Width="24">

<Image Source="/Images/trash.png" Width="20" Height="20"/>

</Button>

</Grid>public List<Lang> ThisLang { get; set; }

public List<History> ThisHistory { get; set; }

public historyPage()

{

InitializeComponent();

DataContext = this;

LangUpdate();

HistoryUpdate();

}

private void delLang\_Click(object sender, RoutedEventArgs e)

{

var item = LangList.SelectedItem as Lang;

CoreProject.RunNonQuery($"DELETE FROM Lang WHERE Id={item.Id}");

LangUpdate();

}

private void BackBtn\_Click(object sender, RoutedEventArgs e)

{

if(NavigationService.CanGoBack)

NavigationService.Navigate(new Pages.TranslatorPage());

}

private void LangUpdate()

{

ThisLang = CoreProject.RunQueryList<Lang>("SELECT \* FROM Lang");

LangList.ItemsSource = ThisLang;

}

private void HistoryUpdate()

{

ThisHistory = CoreProject.RunQueryList<History>("SELECT \* FROM History");

foreach (History history in ThisHistory)

{

history.LangIn = CoreProject.RunQuery<Lang>($"SELECT \* FROM Lang WHERE Id=\'{history.IdLangIn}\'");

history.LangOut = CoreProject.RunQuery<Lang>($"SELECT \* FROM Lang WHERE Id=\'{history.IdLangOut}\'");

}

historyList.ItemsSource = ThisHistory;

}

private void DelHist\_Click(object sender, RoutedEventArgs e)

{

var item = historyList.SelectedItem as History;

if (item != null) {

CoreProject.RunNonQuery($"DELETE FROM History WHERE Id={item.Id}");

HistoryUpdate();

}

}

В меню настроек (рисунок 14) содержатся функции изменения размера шрифта и добавления языка. Функция изменения размера шрифта предназначена для увеличения размера текста в TextBox на главной странице (рисунок 13). При нажатии на кнопку «Добавить язык» открывается окно, в котором пользователь вводит название языка и код ISO 639-1 (рисунок 14).

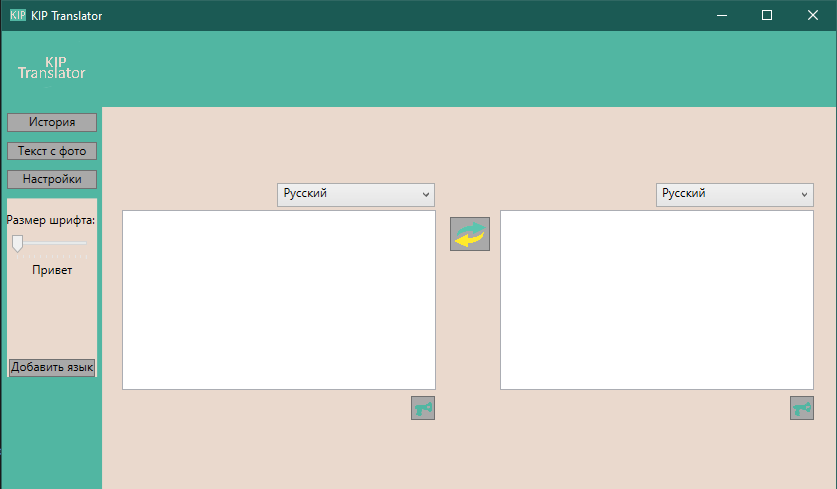


Рисунок 14. Меню настроек

Ниже представлены вёрстка и код страницы:

<Grid Background="#ead9cd">

<Slider Name="fontSizeValue" IsSnapToTickEnabled="True" TickFrequency="1" TickPlacement="BottomRight" Minimum="12" Maximum="24" HorizontalAlignment="Center" Margin="0,36,0,0" VerticalAlignment="Top" Width="80" ValueChanged="fontSizeValue\_ValueChanged"/>

<TextBlock HorizontalAlignment="Center" Margin="0,15,0,0" TextWrapping="Wrap" Text="Размер шрифта:" VerticalAlignment="Top" Width="92"/>

<Button x:Name="AddLang" Content="Добавить язык" HorizontalAlignment="Center" Margin="0,160,0,0" VerticalAlignment="Top" Click="AddLang\_Click"/>

<TextBlock HorizontalAlignment="Center" Margin="0,65,0,0" TextWrapping="Wrap" FontSize="{Binding ElementName=fontSizeValue, Path=Value}" Text="Привет" VerticalAlignment="Top"/>

</Grid>

public settingsPage()

{

InitializeComponent();

}

private void AddLang\_Click(object sender, RoutedEventArgs e)

{

AddLangWindow Window = new AddLangWindow();

bool result = (bool)Window.ShowDialog();

if (result)

{

MainWindow mainWindow = (MainWindow)Application.Current.MainWindow;

mainWindow.nextPage.Refresh();

}

}

private void fontSizeValue\_ValueChanged(object sender, RoutedPropertyChangedEventArgs<double> e)

{

Properties.Settings.Default.fontSizeValue = fontSizeValue.Value;

Properties.Settings.Default.Save();

}

На рисунке 15 изображено окно добавления языка, в котором пользователю предлагается ввести название языка и код ISO 639-1 для языка. При достижении максимального количества символов, в размере более 8, поле «код языка» загорается красным, и блокируется кнопка «добавить язык» (рисунок 16). Также присутствует ссылка на страницу в браузере, на которой предоставляются все названия языков и их код ISO 639-1.

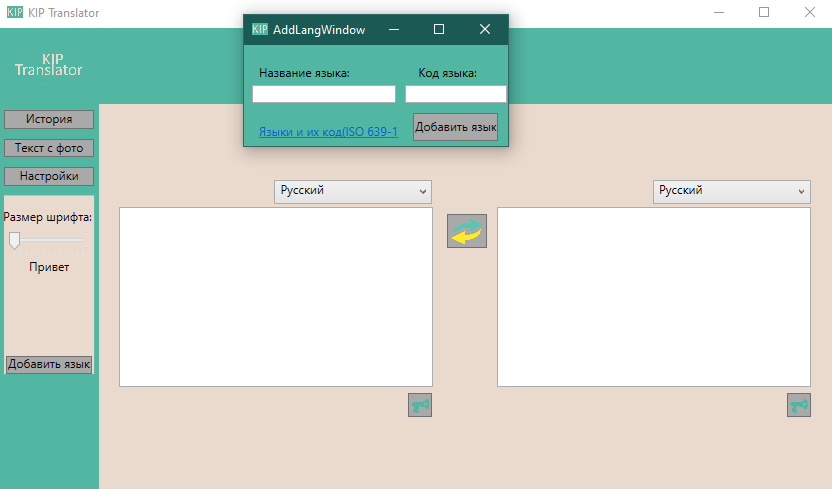


Рисунок 15. Окно добавления языка

Изображение выглядит как Веб-сайт

Автоматически созданное описание

Рисунок 16. Проверка на максимальную длину код ISO 639-1

Ниже представлены вёрстка и код окна:

<Grid Background="#51b6a2">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="169\*"/>

<ColumnDefinition Width="111\*"/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="35\*"/>

<RowDefinition Height="30\*"/>

<RowDefinition Height="39\*"/>

</Grid.RowDefinitions>

<TextBox x:Name="NameLangText" HorizontalAlignment="Center" VerticalAlignment="Center" Width="144" Grid.Row="1" Height="18"/>

<TextBox x:Name="CodeLangText" ToolTip="Введите только два символа" HorizontalAlignment="Center" VerticalAlignment="Center" Width="102" Grid.Row="1" Grid.Column="1" Height="18" TextChanged="CodeLangText\_TextChanged"/>

<Label Content="Название языка:" HorizontalAlignment="Left" Margin="10,16,0,0" VerticalAlignment="Top" Grid.RowSpan="2" Height="26" Width="100"/>

<Label Content="Код языка:" HorizontalAlignment="Left" Margin="10,16,0,0" VerticalAlignment="Top" Grid.Column="1" Grid.RowSpan="2" Height="26" Width="68"/>

<Button x:Name="addLangBtn" Content="Добавить язык" HorizontalAlignment="Center" Margin="0,5,0,5" Width="85" Click="addLangBtn\_Click" Grid.Row="2" Grid.Column="1"/>

<Label Name="LangInfo" HorizontalAlignment="Left" Margin="10,10,0,0" Grid.Row="2" VerticalAlignment="Top">

<Hyperlink NavigateUri="https://cloud.google.com/translate/docs/languages" RequestNavigate="Hyperlink\_RequestNavigate">

Языки и их код(ISO 639-1)

</Hyperlink>

</Label>

</Grid>public AddLangWindow()

{

InitializeComponent();

}

private void addLangBtn\_Click(object sender, RoutedEventArgs e)

{

if (NameLangText.Text != string.Empty && CodeLangText.Text != string.Empty)

{

CoreProject.RunNonQuery($"INSERT INTO Lang(NameLang, CodeLang) VALUES (\'{NameLangText.Text}\', \'{CodeLangText.Text}\')");

this.DialogResult = true;

this.Close();

}

else { MessageBox.Show("Заполните все поля","ВНИМАНИЕ"); }

}

private void CodeLangText\_TextChanged(object sender, TextChangedEventArgs e)

{

if (CodeLangText.Text.Length <= 8)

{

CodeLangText.Background = new SolidColorBrush(Colors.White);

addLangBtn.IsEnabled = true;

}

else

{

CodeLangText.Background = new SolidColorBrush(Colors.Red);

addLangBtn.IsEnabled = false;

}

}

private void Hyperlink\_RequestNavigate(object sender, System.Windows.Navigation.RequestNavigateEventArgs e)

{

Process.Start(new ProcessStartInfo(e.Uri.AbsoluteUri));

e.Handled = true;

}

На рисунке 17 представлена страница «текст с фото». Где пользователю сможет выбрать изображение, на котором будет текст. После «выбрать язык текста на изображении», кнопка «распознать», которая будет выводить текст с изображения в textbox и кнопка «перевести», которая будет отправлять текст из textbox на главную страницу.

Изображение выглядит как снимок экрана, текст, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 17. Страница текст с фото

Ниже представлены вёрстка и код страницы:

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="41\*"/>

<RowDefinition Height="342\*"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition/>

<ColumnDefinition/>

</Grid.ColumnDefinitions>

<Image x:Name="imagePic" Margin="10,10,10,10" Grid.Row="1"/>

<TextBox x:Name="imageText" Grid.Column="1" AcceptsReturn="False" Margin="10,10,10,10" Grid.Row="1" TextWrapping="Wrap"/>

<Button x:Name="backBtn" Grid.Column="1" Content="Назад" Margin="298,0,10,0" Click="backBtn\_Click" HorizontalAlignment="Right" Width="50" Height="25" VerticalAlignment="Bottom" />

<Button x:Name="picImageBtn" Content="Выбрать фото" HorizontalAlignment="Left" Margin="10,0,0,0" VerticalAlignment="Bottom" Height="25" Click="picImageBtn\_Click" Width="80"/>

<TextBox x:Name="pathText" Margin="96,0,10,0" TextWrapping="Wrap" VerticalAlignment="Bottom" Height="25"/>

<Button x:Name="convertBtn" Grid.Column="1" Content="Распознать" HorizontalAlignment="Left" VerticalAlignment="Bottom" Margin="10,0,0,0" Height="25" Click="convertBtn\_Click"/>

<ComboBox Name="ChangeLang" Grid.Column="1" HorizontalAlignment="Left" Margin="87,0,0,0" VerticalAlignment="Bottom" Width="131" Height="25" SelectionChanged="ChangeLang\_SelectionChanged">

<TextBlock Text="Русский язык"/>

<TextBlock Text="Английский язык"/>

</ComboBox>

<Button x:Name="ToTranslateBtn" Grid.Column="1" Content="Перевести" HorizontalAlignment="Right" VerticalAlignment="Bottom" Margin="0,0,67,0" Height="25" Click="ToTranslateBtn\_Click"/>

</Grid>

OpenFileDialog open = new OpenFileDialog();

string chooseLang;

public imageToTextPage()

{

InitializeComponent();

ChangeLang.SelectedIndex = 0;

}

private void backBtn\_Click(object sender, RoutedEventArgs e)

{

if (NavigationService.CanGoBack)

NavigationService.Navigate(new Pages.TranslatorPage());

}

private void picImageBtn\_Click(object sender, RoutedEventArgs e)

{

BitmapImage bitmap = new BitmapImage();

try

{

open.Filter = "Image Files(\*.jpg; \*.jpeg; \*.bmp;)|\*.jpg; \*.jpeg; \*.bmp;";

if (open.ShowDialog() == DialogResult.OK)

{

bitmap.BeginInit();

bitmap.UriSource = new Uri(open.FileName, UriKind.Absolute);

bitmap.EndInit();

imagePic.Source = bitmap;

pathText.Text = open.FileName;

}

}

catch

{

System.Windows.MessageBox.Show("Внимание!\nЗагруженное изображение имеет неправильное расширение","ПРЕДУПРЕЖДЕНИЕ",MessageBoxButton.OK, MessageBoxImage.Warning);

}

}

private void convertBtn\_Click(object sender, RoutedEventArgs e)

{

if (open.FileName != string.Empty)

{

using (var engine = new TesseractEngine(@"./tessdata", chooseLang, EngineMode.Default))

{

using (var img = Pix.LoadFromFile(open.FileName))

{

using (var page = engine.Process(img))

{

imageText.Text = page.GetText();

}

}

}

}

}

private void ChangeLang\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

switch (ChangeLang.SelectedIndex)

{

case 0:

chooseLang = "rus";

break;

case 1:

chooseLang = "eng";

break;

}

}

private void ToTranslateBtn\_Click(object sender, RoutedEventArgs e)

{

if (imageText.Text != string.Empty)

{

Global.text = imageText.Text;

NavigationService.Navigate(new Pages.TranslatorPage());

}

}

## 2.4. Отладка и тестирование программы

Для проверки реализованного функционала необходимо провести отладку и тестирование программного обеспечения.

В таблице представлены результаты отладки и тестирования программного обеспечения.

Таблица 1. Тестирование и отладка программного обеспечения

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| №  теста | Входные данные | Вводимое значение | Ожидаемая реакция программы | Фактический результат | Ошибка выявлена |
| 1 | Текст на перевод | Привет | Перевод на английский язык |  | нет |
| 2 | Текст на перевод | Привет | Перевод на испанский язык |  | нет |
| 3 | Добавление языка | немецкий  Код: ed | Добавление языка |  | нет |
| 4 | Добавление языка с неправильным кодом | Код: edr | Поле загорится красным и кнопка не будет активна |  | нет |

Проведя работу с отладкой и тестированием, можно утверждать, что программа соответствует как функциональным, так и нефункциональным требованиям.

## 2.5 Руководство по использованию программы

Назначением программы является перевод текста.

Функции программы:

* перевод текста;
* добавление перевода в базу данных;
* отображение сохраненного перевода;
* удаление перевода;
* добавление языка;
* удаление языка;
* изменение размера шрифта;
* воспроизведение введённого и переведённого текстов.

Ниже представлены минимальные требования к компьютеру для установки приложения:

* Платформа [OS]: Windows
* Разрядность: х86(32-bit) x64(64-bit)
* ЦП [CPU]: от 500 MHz
* Видеоадаптер [GPU]: 3D
* Видеопамять [VRAM]: 64 Mb
* Винчестер [HDD]: 3Gb
* Оперативная память [RAM]: 256 Mb
* Контроллер: Клавиатура, Мышь

Программное решение «KIP Translator» представляет из себя интерфейс Гугл переводчика с дополнительными функциями.

В данной программе представлены такие функции как: перевод текста, текст с фото, история, изменение размера шрифта, добавление языка.

Функция «перевода текста» – функция, позволяющая пользователю выбрать исходный язык, вписать или вставить интересующий текст и выбрать целевой язык, язык на который будет переведён текст.

Функция «текст с фото» – функция, позволяющая пользователю выбрать нужное изображение, после выбрать исходный язык текста на изображении и нажать на кнопку «распознать», получая текста с изображения, который пользователь может сразу перевести с помощью кнопки «перевести».

Функция «история» – функция, предоставляющая пользователю вариант просмотреть историю переводов или удалить ненужную, а также просмотреть или удалить доступные языки в приложении.

Функция «изменение размера шрифта», представляющая из себя ползунок, который можно перемещать. У ползунка минимальное значение размера шрифта 12, а максимальное – 24. Функция «добавление языка» - функция, открывающая пользователю отдельное окно, в котором требуется от пользователя заполнить два поля, так как «название языка» и «код языка». В поле «код языка» пользователь должен ввести код ISO 639-1, также у этого поля есть проверка на количество вводимых символов, ограничивающая их количество в размере не более 8. Помимо полей на окне присутствует ссылка на страницу в браузере, на которой представлены все языки которые поддерживает приложение, и кнопка «добавить язык».

Для установки приложения необходимо скачать файл-установщик, далее нужно запустить установщик и подтвердить установку во всплывающем окне (рисунок 16). По завершению установки в появившемся окне нажать на кнопку «готово».

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 18. Установщик «KIP Translator»

После нажатия на кнопку «Установить» произойдет установка, и откроется сама программа.

После установки программу можно будет открыть через меню «Пуск» на вашем компьютере или через ярлык на рабочем столе.

После запуска программы пользователь сможет как перевести текст, так и на навигационному меню перейти на дополнительные функции, путём нажатия нужной кнопки, расположенной слева на главном окне программы. После перехода на нужную функцию, для возвращения на страницу перевода нужно нажать кнопку «назад», которая расположена справа на странице нужной функции.

Удаления программы происходит через меню «Панель управления», где нужно выбрать пункт «удаление программы», где в свою очередь нужно найти «KIP Tranlsator». При нажатии на кнопку удалить будет открыто окно, представленное на рисунке 17, на котором нужно нажать на кнопку «Да».

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 19. Удаление «Kip Translator»

ЗАКЛЮЧЕНИЕ

В данной работе была рассмотрена задача разработки мультиплатформенного переводчика с распознаванием текста, который может использоваться на различных устройствах, в том числе на компьютерах и мобильных устройствах. Для разработки desktop-приложения был использован .net framework, а для мобильного приложения - Android Studio.

Для достижения этой цели были использованны современные методы и технологии, включая .NET Framework для разработки desktop-приложения и Android Studio для мобильной версии. Разработанный переводчик имеет удобный интерфейс и прост в использовании, что делает его доступным для широкого круга пользователей.

В результате дипломного проекта был создан инструмент, способный переводить текст на различных языках и распознавать его с помощью машинного обучения. Это позволяет ускорить процесс перевода и повысить его точность.

Кроме того, разработанный программный продукт имеет большой потенциал для дальнейшего развития и улучшения. Возможным направлением для будущих улучшений компьютерной версии является введение функции голосового ввода и распознавания речи, а для мобильной версии – распознавание текста по изображению, что еще больше упростит использование переводчика в повседневной жизни.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

*Законодательные и нормативные акты:*

1. ГОСТ Р 7.0.12-2011 Библиографическая запись. Сокращение слов и словосочетаний на русском языке. Общие требования и правила. – М.: Стандартинформ, 2012. – 61 с.
2. ГОСТ 7.32-2017 Отчет о научно-исследовательской работе. Структура и правила оформления. – М.: Стандартинформ, 2017. – 47 с.
3. ГОСТ 7.82-2001 Библиографическая запись. Библиографическое описание электронных ресурсов. Общие требования и правила составления. – М.: ИПК Издательство стандартов, 2001. – 39 с.
4. ГОСТ Р 7.0.100-2018 Библиографическая запись. Библиографическое описание. Общие требования и правила составления. – М.: Стандартинформ, 2018. – 122 с.
5. ГОСТ Р 7.0.5-2008 Библиографическая ссылка. Общие требования и правила составления. – М.: Стандартинформ, 2008. – 32 с.

*Интернет-документы:*

1. Интернет-сервис для построения схем и диаграмм Draw.io. – [Электронный ресурс]. – URL: <https://www.draw.io/>
2. SQLite в C# и .NET. Подключение к базе данных SQLite – [сайт] – URL: <https://metanit.com/sharp/adonetcore/4.1.php>
3. Tesseract documentation – [сайт] – URL:<https://tesseract-ocr.github.io/>
4. Tesseract Open Source OCR Engine (main repository) – [сайт] – URL: <https://github.com/tesseract-ocr/tesseract>
5. The DispatcherTimer – [сайт] – URL: <https://wpf-tutorial.com/misc/dispatchertimer/>
6. OpenFileDialog и SaveFileDialog – [сайт] – URL: <https://metanit.com/sharp/windowsforms/4.20.php>
7. Синтез речи - WPF – [сайт] – URL: <https://professorweb.ru/my/WPF/UI_WPF/level26/26_7.php>
8. ListView метанит – [сайт] – URL:<https://metanit.com/sharp/windowsforms/4.12.php>
9. ProgressBar и Slider – [сайт] – URL:<https://metanit.com/sharp/wpf/5.15.php>
10. Combobox – C# и WPF – [сайт] – URL:<https://metanit.com/sharp/wpf/5.8.php>

# ПРИЛОЖЕНИЕ