**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«РОССИЙСКАЯ АКАДЕМИЯ НАРОДНОГО ХОЗЯЙСТВА**

**И ГОСУДАРСТВЕННОЙ СЛУЖБЫ**

**ПРИ ПРЕЗИДЕНТЕ РОССИЙСКОЙ ФЕДЕРАЦИИ»**

**Смоленский филиал РАНХиГС**

**Отдел среднего профессионального образования**

Направление подготовки:

09.02.07 Информационные системы и программирование

**ПРОЕКТНАЯ РАБОТА**

по дисциплине «Разработка мобильных приложений» на тему: «Мобильное приложение для перевода и голосовой озвучки слов на английском языке»

**Автор работы:**

студент 3 курса

очной формы обучения

группа 27/11-К/ИТО

Махницкий Д.С.

**Преподаватель:**

Ветров Д.А.

Смоленск 2024

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc163650156)

[1 ТЕХНИКО-ЭКОНОМИЧЕСКИЙ АНАЛИЗ ОБЪЕКТА ИССЛЕДОВАНИЯ 5](#_Toc163650157)

[2 ПРОЕКТИРОВАНИЕ МОБИЛЬНОГО ПРИЛОЖЕНИЯ 9](#_Toc163650158)

[3 РАЗРАБОТКА И ТЕСТИРОВАНИЕ 17](#_Toc163650159)

[ЗАКЛЮЧЕНИЕ 23](#_Toc163650160)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 24](#_Toc163650161)

[ПРИЛОЖЕНИЕ А 26](#_Toc163650162)

[ПРИЛОЖЕНИЕ Б 27](#_Toc163650163)

[ПРИЛОЖЕНИЕ В 28](#_Toc163650164)

[ПРИЛОЖЕНИЕ Г 44](#_Toc163650165)

# ВВЕДЕНИЕ

В современном мире человек окружен различными вычислительными устройствами. Некоторые даже не подозревают, насколько глубоко в нашу жизнь пробрались информационные технологии. Мобильные устройства за последние 20 лет прошли путь от кнопочных телефонов с черно-белым экраном и несколькими тысяч пикселей до инновационных смартфонов с невообразимой вычислительной мощностью, миллионом отображаемых цветов, миллионом пикселей, безграничным множеством различных функций, которые постоянно пополняются благодаря доступности мобильной разработки, а также возможными внутренними техническими дополнениями, которые, например позволяют измерять положение смартфона в пространстве или даже определять стороны света.

Сегодня разработать собственное мобильное приложение с желаемым функционалом, интерфейсом и отличительными особенностями может даже незнакомый с программированием человек, воспользовавшись zero-coding платформами, на которых разработчику достаточно создать интерфейсную часть, а затем с помощью интуитивно понятной настройки определить желаемый функционал.

Среди всех приложений весомую популярность имеют приложения-переводчики, которые позволяют оперативно переводить слова с одного языка на другой.

В качестве объекта исследования выступает мобильное приложение, а в качестве предмета – приложение для перевода и озвучки слов на английском языке.

Цель работы – разработать удобное, минималистичное, быстродействующее приложение для перевода и озвучки слов с русского на английский и с английского на русский языки с примерами употребления слов в контексте, с интуитивно понятным и дружественным интерфейсом.

Поставленной цели можно достичь, выполнив следующие задачи:

- проанализировать предметную область;

- исследовать существующие аналоги;

- определить функциональные характеристики приложения;

- разработать макеты приложения;

- изучить фреймворк Flutter для разработки мобильных приложений;

- разработать приложение основываясь на собранной информации;

- экономически обосновать проектные решения.

Методологическая основа состоит из:

- методов системного анализа;

- методов сравнения и аналогий;

- методов изучения.

В проектной работе использована, в основном, различная отечественная литература по теме, также присутствует зарубежная.

# 1 ТЕХНИКО-ЭКОНОМИЧЕСКИЙ АНАЛИЗ ОБЪЕКТА ИССЛЕДОВАНИЯ

При изучении иностранных языков люди хотят не только научиться хорошо переводить слова, но и правильно их произносить. Для этого можно воспользоваться различными онлайн-переводчиками в Интернете, но, как будет рассмотрено далее, все они имеют свои недостатки, которые можно исправить, разработав собственное мобильное приложение.

В качестве целевого языка для перевода слов с русского был выбран английский язык, так как это самый распространенный на Земле язык. На нем в повседневной жизни говорит около 20% населения планеты, знает минимальный для общения набор 40%, а также 55% сайтов в Интернете написаны на английском, что делает его чрезвычайно важным для всего человечества, например для людей, которые любят находить в Интернете интересующую их информацию или любителей путешествовать. Про последних можно заметить, что во время путешествия может возникнуть потребность перевести незнакомое слово. Для этого, конечно, можно взять с собой громоздкий ноутбук, книгу-словарь или при желании даже настольный компьютер, но это задача не из легких, так как при регулярном пользовании этими устройствами человек теряет не только время, но и мобильность, поэтому лучшим вариантом в качестве вычислительного устройства для приложения-переводчика подходит мобильное устройство. Так как смартфон меньше компьютера или ноутбука, то ввести интересующее слово и тут же увидеть его перевод, транскрипцию, озвучку от носителей языка и примеры его использования в контексте, который очень важен в английском языке, будет гораздо проще и быстрее.

Для удобства в разрабатываемом приложении можно переводить не только слова, но и целые предложения и текста, но без озвучки и контекстных примеров – они доступны только для одного слова.

Рассмотрим Google переводчик в качестве аналога. Он имеет следующие характеристики:

- перевод слов, текстов, документов на 103 языка;

- онлайн-словарь при переводе слов;

- транскрипция, транслитерация, озвучка, голосовой ввод;

- распознавание текста по фото;

- максимальная длина текста для перевода – 5000 символов.

Данный переводчик содержит почти все описанные ранее функции для нашего собственного мобильного приложения, но при более детальном изучении обнаружено, что озвучка в Google переводчике основана на алгоритмах искусственного интеллекта: аудиофайл для каждого перевода генерируется на основе заранее записанных звуков и их сочетаний. Данная технология используется для полного покрытия всех возможных к введению вариантов слов и предложений с минимальными затратами памяти. То есть, если бы было принято решение записывать все возможные комбинации из 5000 символов отдельными аудиофайлами, учитывая, что 1 символ может иметь около 100 значений, получилось бы около различных аудиофайлов. Допустим, что один аудиофайл записан в плохом качестве и в среднем имеет вес около 1 Мб. Тогда для хранения всех аудиофайлов понадобилось бы около Пб, что превышает все возможные современные хранилища данных в колоссальное количество раз. Таким образом, команда разработчиков сделала выбор в пользу озвучки с помощью искусственного интеллекта, но данный выбор не обошелся без потерь: в свою очередь, так как в современном мире искусственный интеллект все еще не идеален, он может иногда серьезно искажать ударение и интонацию получившихся аудиозаписей, что может сбить с толку пользователя. Мобильная версия Google переводчика не предоставляет примеры использования слов в контексте, в отличии от вебсайт-версии. Помимо рассмотренного ранее переводчика, также существуют и другие, например Яндекс переводчик, DeepL, Reverso, Bing Microsoft Translator, но все они фактически имеют такой же функционал, как у Google переводчика, хоть и с незначительными изменениями, например Яндекс переводчик разрешает переводить текст длинной 10000 символов, а не 5000 как Google.

Все перечисленные переводчики объединяет одна деталь – они либо вообще не озвучивают текст, либо делают это, но с недочетами – все из-за применения технологии озвучки с помощью алгоритмов машинного обучения.

Основная идея разрабатываемого мобильного приложения заключается в том, чтобы кроме перевода слова пользователь мог послушать его гарантированно правильное произношение, а также узнать, как это слово применяется в контексте. Разрабатывать собственный переводчик с нуля – очень сложная и растянутая во времени работа – для точного перевода понадобится много времени на разработку программного кода искусственного интеллекта и его обучение, поэтому в качестве основного источника для перевода слов необходимо будет воспользоваться Google переводчиком, так как он исследует стилистический контекст введенного текста и работает на широкую аудиторию с 2008 года. Следовательно, у компании накопилась большая информационная база различных данных, например о некорректных переводах, которые они могут проанализировать и в соответствии с выводами внести изменения в искусственный интеллект, что является преимуществом по сравнению с другими, позднее вышедшими сервисами.

В качестве источника для озвучки слов выбран сервис Cambridge Dictionary, который предоставляет озвучку около 1500000 слов с Британским и Американским акцентом. Основная идея приложения следующая: если введенное пользователем слово есть на сайте Cambridge Dictionary и имеет голосовой перевод, то он будет доступен к прослушиванию, а если такого слова там нет, то приложение просто будет переводить текст без какой-либо озвучки. Также если пользователь ввел одно слово или какую-то фразу, которая целиком имеет какое-либо контекстное значение на сайте Reverso, например фразеологизм «бить баклуши», то приложение покажет несколько этих предложений; с приведенным фразеологизмом приложение отобразит предложение «If you ever get tired of loafing, you can get work waiting tables.», которое переводится как «Если тебе когда-нибудь надоест бить баклуши, можешь поработать официантом.»

Рассмотрим также пункт 2.9 статьи 1270 ГК РФ [1] под названием «Исключительное право на произведение»: Использованием произведения независимо от того, совершаются ли соответствующие действия в целях извлечения прибыли или без такой цели, считается, в частности: перевод или другая переработка произведения.

Таким образом, данная статья накладывает запрет на перевод какого-либо произведения, например, с помощью онлайн-переводчика. Это необходимо учитывать при реализации мобильного приложения в качестве ограничения.

# 2 ПРОЕКТИРОВАНИЕ МОБИЛЬНОГО ПРИЛОЖЕНИЯ

Разрабатываемое приложение будет иметь только одну категорию пользователей: обычный пользователь, а также только одну базу данных SQLite [2,3], которая содержит текста на русском и английском языке, защищенные авторским правом. Если пользователь введет такой текст, то приложение уведомит его о попытке нарушения ГК РФ с указанием соответствующей статьи и не будет выполнять последующую обработку данного перевода. Здесь и далее будет использоваться выражение «обработка перевода», которое означает момент, когда пользователь ввел текстовые данные и нажал кнопку «перевести» и приложение начало делать HTTP-запросы [4,5] и выполнять отображение собранных результатов.

В результате эмпирического анализа, проведенного на сайте используемого в качестве источника для HTTP-запросов dictionary.cambridge.org (далее DCO), было выявлено, что озвучка на английском языке у слов встречается значительно чаще, чем у фраз. Здесь значения слова «слово» представляет собой набор буквенных символов без пробела, фраза – тоже набор буквенных символов, но содержит как минимум один пробел. Хотя на DCO и существует множество различных фраз, но лишь у малой части из них есть озвучка, и в основном у устойчивых фраз, таких как «vice versa», что переводится как «наоборот». Также, в процессе анализа сочетания с тремя и более количеством слов не были найдены вообще.

Основываясь на перечисленных пунктах, было принято решение не запрашивать у HTTP-ресурса озвучку для фразы в процессе обработки перевода, так как вероятность того, что пользователь введет короткую фразу, у которой на DCO есть озвучка – крайне мала, а это означает, что один из трех HTTP-запросов, производимых для каждой введенного пользователем фразы, очень часто будет выполняться с пустым результатом, а так как при повседневном пользовании мобильным устройством оно не подключено к сети Интернет стабильно сильным, проводным сигналом, то время предоставления результата заметно увеличится, что может стать одной из причин плохого UX. Озвучка текста не рассматривалась вообще, т.к. DCO позиционирует себя как словарь слов.

Проведя анализ второго HTTP-ресурса с адресом context.reverso.net (далее CRN), было выяснено, что на данном сайте доступны разнообразные примеры употребления не только обычных слов, но и целых фраз, содержащих 3-4 слова. Например, для фразы «не делайте это» на CRN доступно к прочтению 20 различных примеров ее употребления на английском языке. Так как данный сайт отлично справляется с поиском примеров, было принято решение, что вводимые пользователем данные длиной до 5 слов, в том числе одно слово, считаются фразой и для них будет выполняться поиск примеров использования в контексте.

Проанализировав третий выбранный HTTP-ресурс – translate.google.ru (далее TGR) – было выяснено, что он способен выдавать перевод для любых текстовых данных – и слово, и фраза, и текст длиной до 5000 символов гарантировано переводятся на другой язык, так как для этого TGR использует искусственный интеллект. Следовательно, в разрабатываемом приложении так же будет установлено ограничение на ввод текста в 5000 символов.

Таким образом, основываясь на перечисленных выше особенностях поведения HTTP-ресурсов, используемых для получения различных данных, было выведено три возможных вида данных по типу их содержания: слово, фраза и текст. Перечисленные виды данных изображены справа от пользователя на UML-диаграмме прецедентов (рисунок 2.1). Разграничение данных по трем этим видам важно при разработке приложения, так как оно может значительно повысить скорость обработки запрошенного пользователем перевода [6]. Делается это следующим образом: после ввода данных выполняется проверка на соответствие одному из видов данных: являются ли введенные пользователем данные словом, фразой или текстом. Например, если пользователь ввел текст, то приложение «знает», что нет смысла выполнять HTTP-запросы по поиску озвучки и примеров его использования в контексте, и потому не выполняет их, тем самым экономя определенное количество времени и увеличивая скорость обработки перевода.

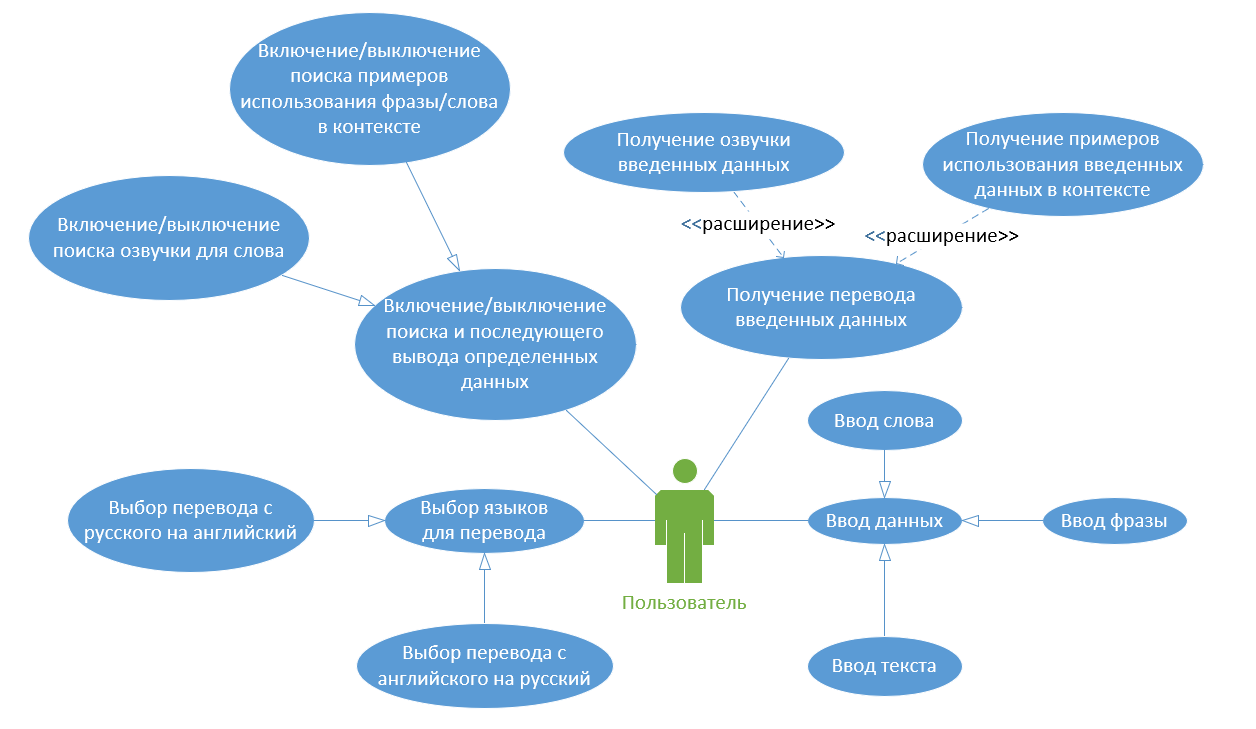


Рисунок 2.1 – UML-диаграмма прецендентов

Также на диаграмме прецедентов можно увидеть прецедент, описывающий возможность у пользователя включить или выключить поиск озвучки для слова или примеров использования в контексте. Данная возможность была добавлена по двум причинам, которые имеют общие черты, но разное происхождение. Первый случай – когда у пользователя стабильно хорошее подключение к Интернету, но нет необходимости получать озвучку слова и/или пример, он может выключить их поиск и тогда приложение будет обрабатывать данный перевод быстрее, чем с ними, так как будет выполнен всего один HTTP-запрос вместо трех. Второй случай – когда у пользователя плохое, нестабильное соединение с Интернетом, но он хочет повысить скорость получения перевода, пожертвовав озвучкой и/или примерами использования в контексте. При желании пользователь сможет выключить и озвучку, и примеры из контекста и пользоваться приложением как обычным переводчиком слов.

Анализируя UML-диаграммы последовательностей для каждого возможного вида данных, вводимых пользователем, на рисунках 2.2-2.4 можно заметить, что у каждой диаграммы разное количество сущностей, выполняющих действие, так как для обработки каждого вида данных, таких как слово, фраза или текст, нужен разный набор HTTP-запросов, как это было описано ранее.

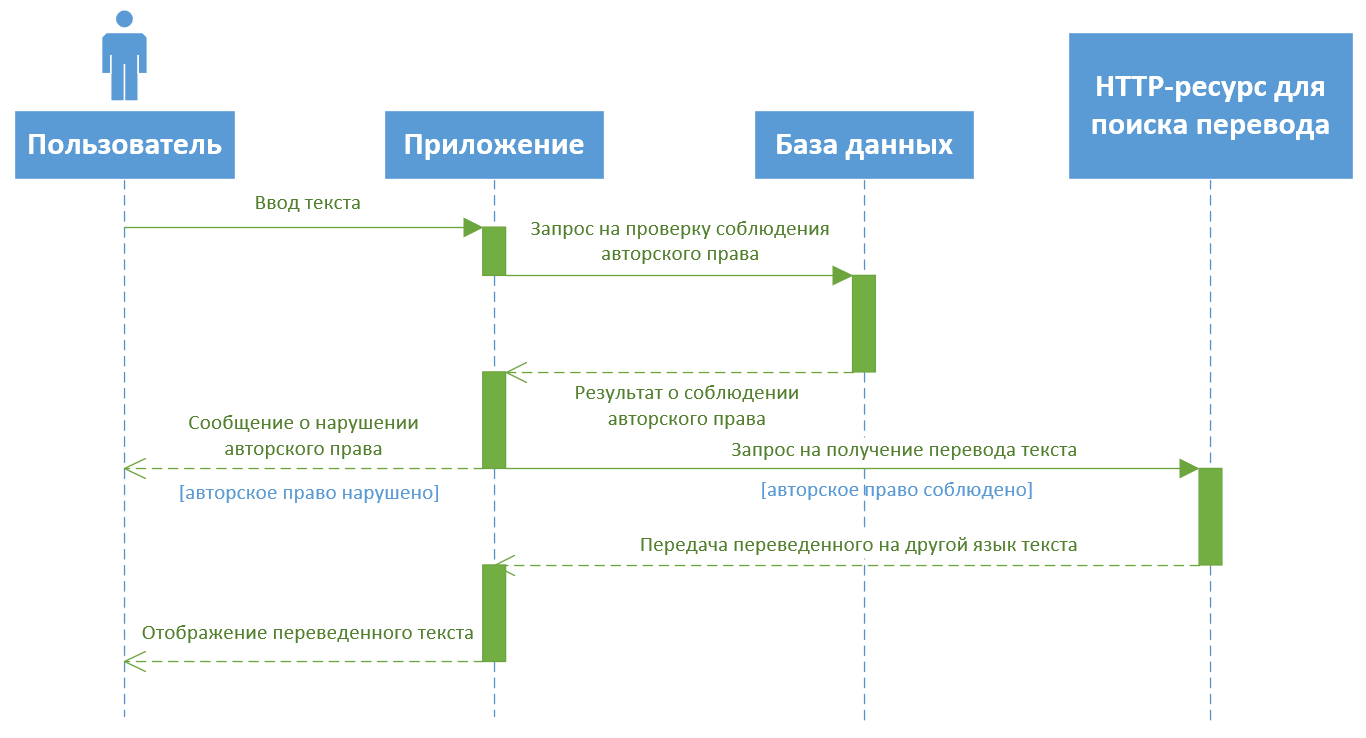


Рисунок 2.2 – UML-диаграмма последовательностей при вводе пользователем текста

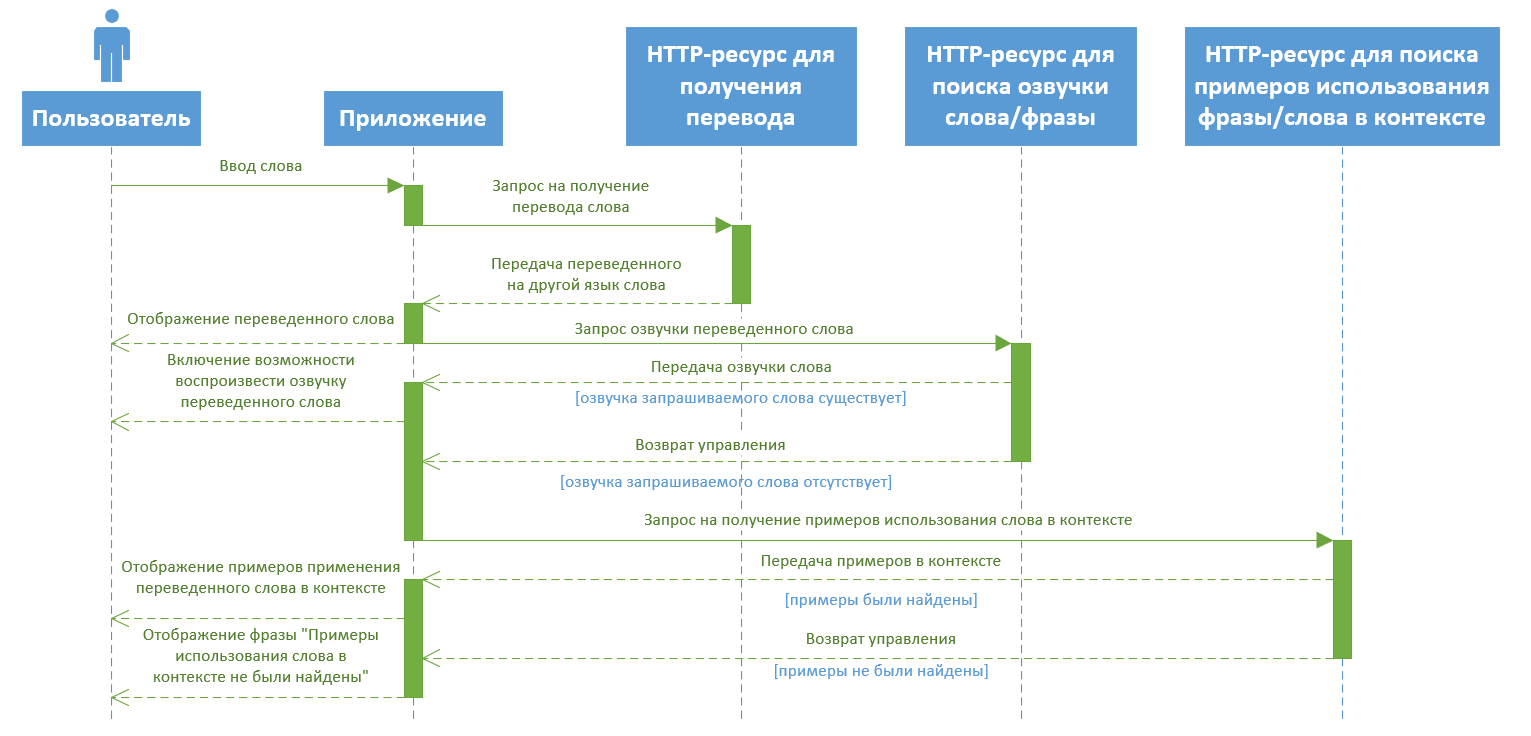


Рисунок 2.3 – UML-диаграмма последовательностей при вводе пользователем слова с включенными функциями озвучки и приведения примеров использования слова/фразы в контексте

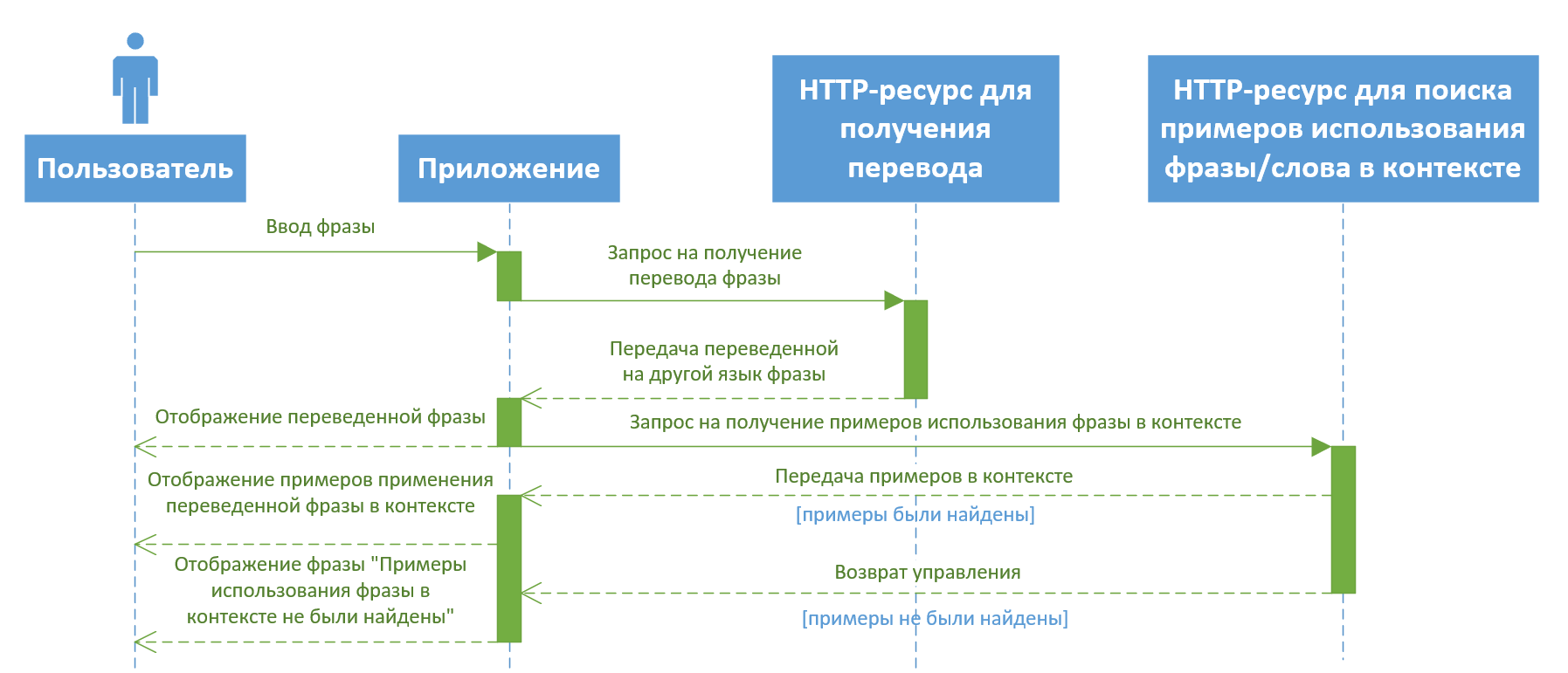


Рисунок 2.4 – UML-диаграмма последовательностей при вводе пользователем фразы с включенной функцией приведения примеров использования слова/фразы в контексте

На рисунке 2.5 изображена диаграмма классов разрабатываемого приложения. Из-за специфики языка программирования и выбранного средства разработки в данной диаграмме полностью отсутствуют какие-либо связи между классами, так как в приложении присутствует всего два относительно простых экрана, в которых почти нет повторяющихся компонентов, которые, например, можно было бы наследовать.

Технически в языке программирования Dart нет статических классов, но их можно создать, сделав конструктор класса приватным и добавив в него статические методы [7,8]. Таким образом был реализован, например, класс static\_objects, который имеет такое обобщенное название, потому что хранит различные виджеты: стили текста, кнопок, иконки и т.д. чтобы избежать дублирования кода или же принципа DRY [9,10,11]. Класс \_HomeScreen имеет наибольшее количество членов, так как данный виджет является основным экраном в приложении и в нем сфокусирован основной функционал [12]. Класс HomeScreen отсутствует на диаграмме, т.к. он не содержит полей и является обычной упаковкой для класса \_HomeScreen.



Рисунок 2.5 – UML-диаграмма классов

В интерфейсе приложения, макеты которого изображены на рисунках 2.6-2.7, должны преобладать зеленые, синие и желтые цвета, а цвет шрифта должен хорошо контрастировать на фоне холодных тонов. Также предполагается, что интерфейс должен быть минималистичным и интуитивно понятным [13] за счет небольшого количества элементов в нем, а также за счет наличия всего двух экранов: главного и с настройками. Кнопка воспроизведения озвучки должна быть изображена в виде привычного громкоговорителя, кнопка с настройками – в виде шестеренок. Вводить данные пользователь может только в поле для ввода текста. Стрелка сверху посередине подсказывает пользователю, что перевод происходит с языка с левой от нее стороны на язык с правой. Так как приложение может искать озвучку только для слов и фраз на английском языке, то при выборе перевода с английского на русский она будет автоматически отключаться.



Рисунок 2.6 – Макет главного экрана

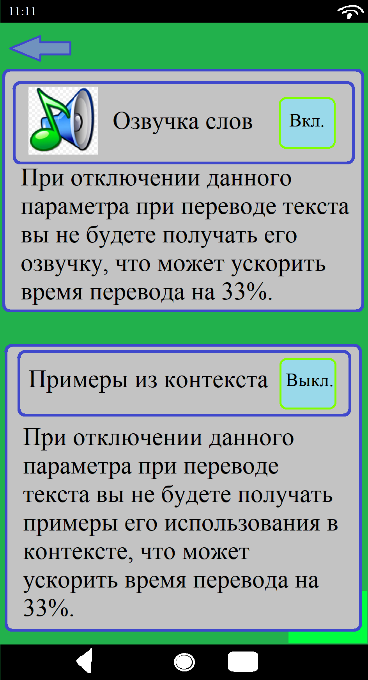


Рисунок 2.7 – Макет экрана с настройками

На рисунках 2.8-2.9 изображены две различные модели единственной базы данных, которая состоит из одной таблицы. В каждой строке есть текст и хэш этого текста, который служит первичным ключом. Данная конструкция используется, потому что в SQLite создать первичный ключ для текста можно только по его первым 256 символам, что в теории может допустить ввод в базу данных дубликата, так как максимальная длина вводимого пользователем текста – 5000 символов.

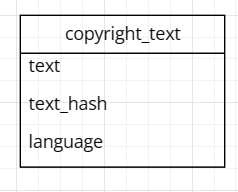


Рисунок 2.8 – Логическая модель базы данных

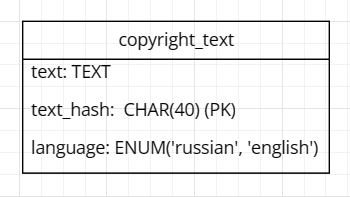


Рисунок 2.9 – Физическая модель базы данных

Можно подвести итог, что разрабатываемое приложение позволит в полной мере реализовать потребность в изучении английского языка у широкого круга пользователей. Оно поможет как перевести слова с одного языка на другой, так и озвучить их, научив пользователя правильному произношению.

# 3 РАЗРАБОТКА И ТЕСТИРОВАНИЕ

Разработка мобильного приложения будет происходить с помощью фреймворка Flutter [7], для вертикально расположенного экрана с разрешением 1080x2400 пикселей, что устанавливается с помощью следующих строк:

SystemChrome.setPreferredOrientations([

DeviceOrientation.portraitUp,

]);

Согласно макету на рисунке 2.6, можно сверстать упрощенную таблицу для компонентов главного экрана (рисунок 3.1). Здесь основная идея в том, что все элементы верхней части до виджета ListView будут отображены на экране всегда, и, следовательно, будут в быстром доступе у пользователя, а нижнюю часть – виджеты внутри ListView – можно прокручивать вертикально.

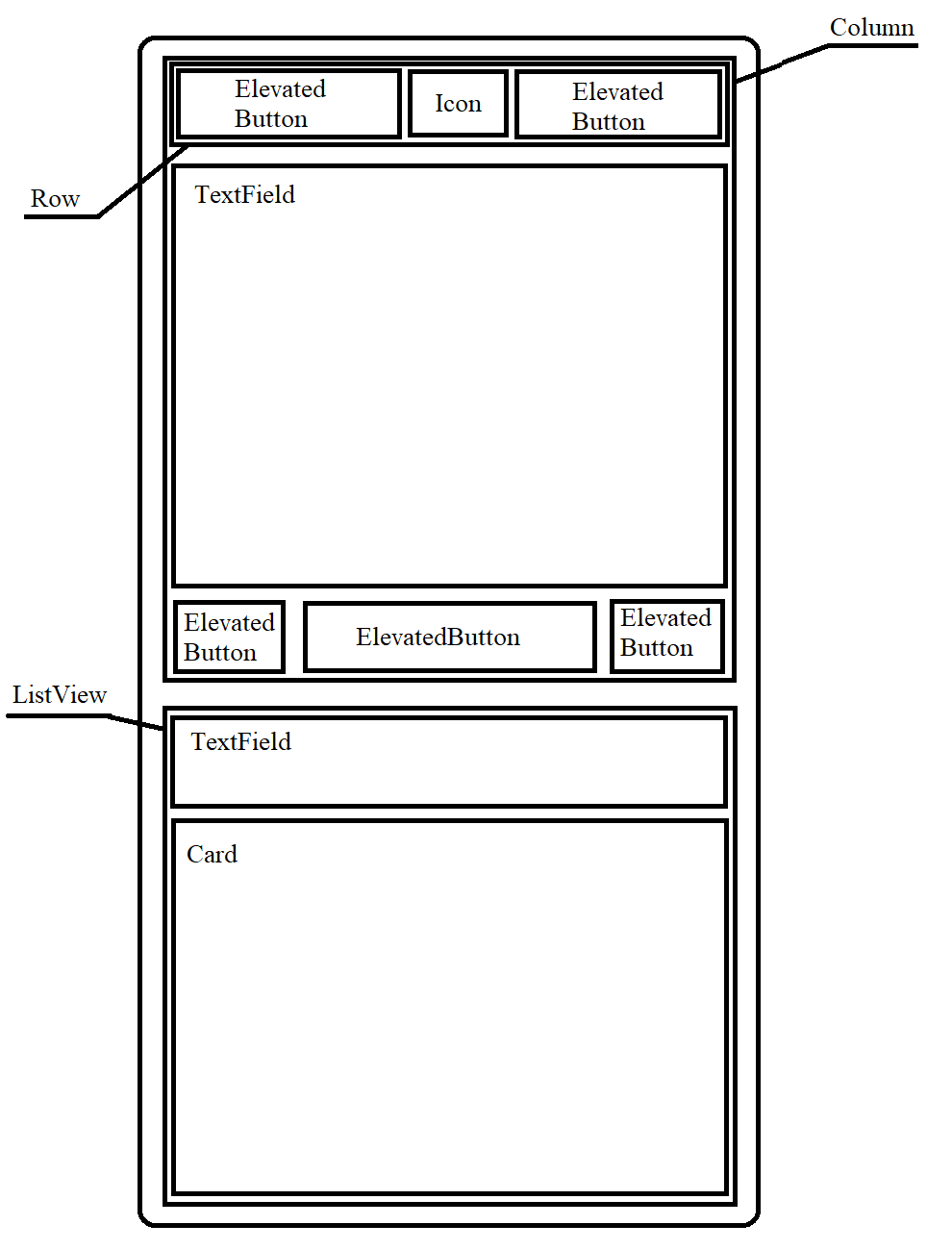


Рисунок 3.1 – Упрощенная таблица расположения компонентов на главном экране

На рисунке 3.2 изображена схема взаимосвязи модулей и массивов данных, рассматривая которую можно заметить, что основная часть модулей вызывается из модуля \_HomeScreen. Семь различных модулей использует модуль обработки события нажатия кнопки перевода translationButtonOnPressed.

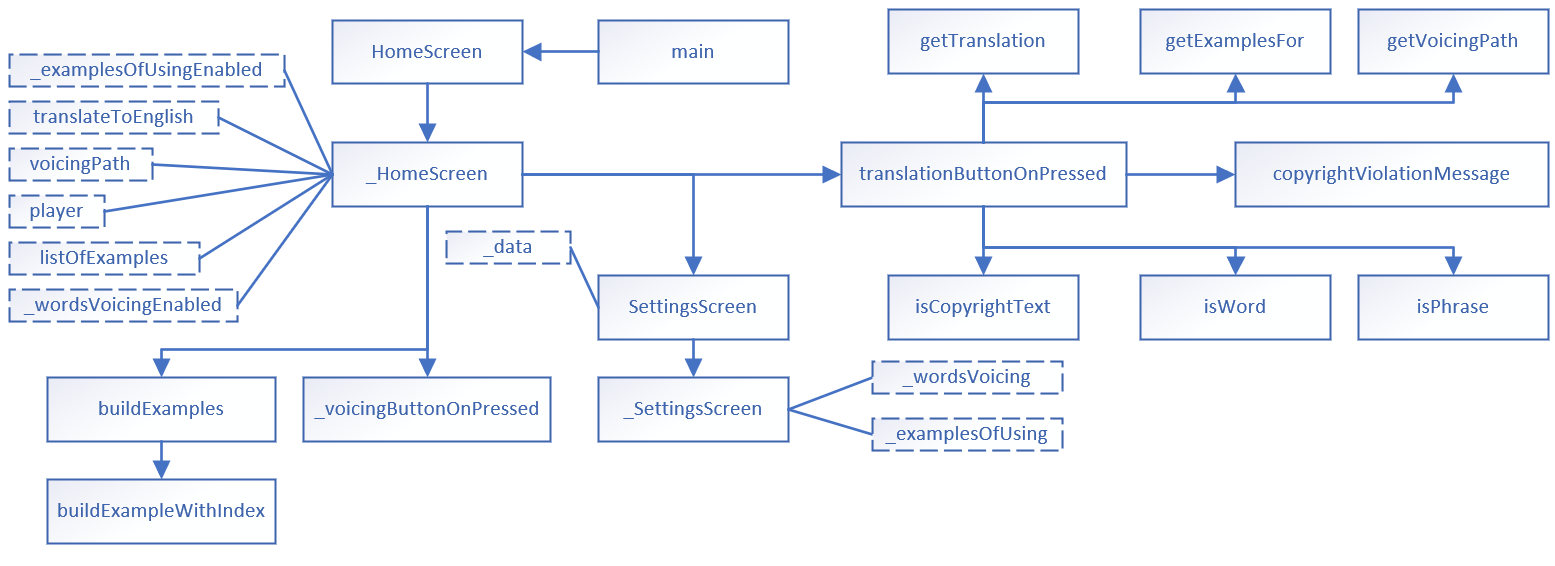


Рисунок 3.2 – Схема взаимосвязи модулей и массивов данных

Перед рассмотрением особенностей самого приложения, рассмотрим подключаемую базу данных SQLite, код которой представлен в пятом пункте приложения В. Выбранная СУБД создается и используется далее асинхронно, что требует особого подхода в ее использовании в синхронной функции построения виджета HomeScreen, а именно – использование асинхронного метода обработки нажатия кнопки перевода translationButtonOnPressed, которая фактически выполняет все действия, представленные на рисунке в приложении А.1.

Манипуляция данными выполняется непосредственно в коде, поэтому чтобы добавить или удалить строки необходимо сделать это заранее, до запуска самого приложения, так как база данных используется как статическое хранилище данных, которые не должны изменяться в процессе работы программы. По крайней мере, разрабатываемое приложение не предусматривает категорию пользователя, который может изменять содержимое базы данных, как, например, администратор, который добавляет текста, на которые недавно было оформлено авторское право или наоборот, текста на которые авторское право закончилось [9].

Для удобного взаимодействия с базой данных используется модель текста с авторским правом, которая описана в четвертом пункте приложения В. Всего в базу данных было введено шесть текстов, пять из которых имеют авторские права, а шестое – короткое предложение, которое можно быстро ввести и проверить работоспособность выполнения проверки авторских прав. База данных имеет стандартные методы для манипуляции данными:

- insertCopyrightText – для ввода текста в бд;

- updateCopyrightText – для обновления существующего текста;

- deleteCopyrightText – для удаления одного текста;

- deleteAllCopyrightTexts – для удобного удаления всех текстов;

- copyrightTexts – для получения всех текстов.

Всего в приложение было добавлено 3 дополнительные зависимости [6]:

- path – для удобного взаимодействия с базой данных;

- assets\_audio\_player – для воспроизведения озвучки слов;

- sqflite – для создания базы данных.

Далее необходимо рассмотреть самые важные моменты работы программы.

Во-первых, разработанные экраны мобильного приложения, изображенные на рисунках Б.1–Б.2, немного отличаются внешне от их макетов на рисунках 2.6-2.7., хотя основное требование к интерфейсу – минималистичность – было соблюдено. Объясняется это тем, что в процессе изучения фреймворка был сделан выбор в пользу более удобной реализации некоторых компонентов: например, вместо написания около двадцати строк кода для импорта одной png-фотографии с шестеренками для кнопки с настройками было решено использовать минималистичную иконку с одной шестеренкой, так как она более гибкая к настройке (например, можно легко поменять ее цвет или размер) и присутствует во Flutter по умолчанию в наборе иконок. Также было принято решение в экране с настройками не использовать двойные рамки вокруг элементов с озвучкой слов и примерами, так как чтобы вручную отрисовать данные рамки, необходимо написать относительно много кода, результат которого никак не повлияет на функционал приложения, а без этих рамок настройки выглядят не менее понятными.

Во-вторых, данные о текущем состоянии использования озвучки и примеров в контексте хранятся в классе \_HomeScreen, между двумя экранами передаются следующим образом: при переходе на экран SettingsScreen выполняется передача данных о текущем состоянии озвучки и использования примеров через конструктор withData в виде списка следующим образом:

var data = await Navigator.push(

context,

MaterialPageRoute(

builder: (BuildContext context) => SettingsScreen.withData([\_wordsVoicingEnabled, \_examplesOfUsingEnabled]),

),

);

При нажатии кнопки возврата на главный экран, класс SettingsScreen вызывает метод Navigator.pop, который возвращает новые данные, введенные пользователем об использовании озвучки и примеров:

Navigator.pop(context, [\_wordsVoicing, \_examplesOfUsing]);

В следующих двух строках после предыдущей команды управление передается обратно экрану HomeScreen и происходит считывание введенных в экране SettingsScreen пользователем данных:

wordsVoicing = data[0] as bool;

examplesOfUsing = data[1] as bool;

Для тестирования работы мобильного приложения используются тест-кейсы [5], которые изображены в таблице 3.1. Тесты проводились на эмуляторе с операционной системой Android 11.0 R на архитектуре x86, но так как фреймворк Flutter является кроссплатформенным, результаты выполнения тест-кейсов должны быть такими же при их выполнении на других операционных системах с идентичным разрешением экрана [8, 12]. Результаты тестирования отображены (поэтапно, где это необходимо) в приложении Г.

Таблица 3.1 – Тест кейсы для тестирования мобильного приложения

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Идентификатор** | **Приоритет** | **Модуль** | **Подмодуль** | **Шаги** | **Ожидаемый результат** |
| CP\_Text\_Input | Средний | HomeScreen | translationButtonOnPressed | Проверка обнаружения ввода текста с авторским правом и последующее уведомление пользователя.  1.Запустить приложение  2.Выбрать перевод с английского на русский язык.  Шаги:  1.Ввести текст «some text with copyright» в единственное поле для ввода текста.  2.Нажать кнопку «Перевести». | 1.В поле ввода введен текст «some text with copyright».  2.Приложение отображает окно с надписью о попытке нарушения авторских прав, а также очищает поле с введенным текстом. |
| RU\_To\_EN | Высокий | HomeScreen | translationButtonOnPressed | Проверка работоспособности перевода текста с русского языка на английский язык.  1.Запустить приложение.  2.Выбрать перевод с русского на английский язык.  Шаги:  1.Ввести текст «привет мир».  2.Нажать кнопку «Перевести». | 1.В поле ввода текста введена фраза «привет мир».  2.Приложение отображает текст «hello world» в поле для вывода текста. |
| EN\_To\_RU | Высокий | HomeScreen | translationButtonOnPressed | Проверка работоспособности перевода текста с английского на русский язык.  1.Запустить приложение.  2.Выбрать перевод с английского на русский язык.  Шаги:  1.Ввести текст «hello world».  2.Нажать кнопку «Перевести». | 1.В поле ввода текста введена фраза «hello world».  2.Приложение отображает текст «привет мир» в поле для вывода текста. |
| Examples\_Off | Средний | SettingsScreen | createState | Проверка работоспособности отключения поиска примеров употребления в контексте.  1.Запустить приложение.  2.Выбрать перевод с английского на русский язык.  Шаги:  1.Нажать кнопку с настройками.  2.В открывшемся экране убрать флаг с пункта «Примеры».  3.Нажать стрелку в левом верхнем углу экрана.  4.Ввести текст «привет мир».  5.Нажать кнопку «Перевести». | 1.Открывается экран с настройками приложения.  2.Пункт «Примеры» имеет пустой квадрат.  3.Происходит возврат на главный экран приложения.  4.В поле ввода текста введен текст «привет мир».  5.Приложение отображает текст «hello world» в поле для вывода текста, а под полем вывода отсутствует поле с примерами употребления в контексте. |
| Voicing\_Off | Средний | SettingsScreen | createState | Проверка работоспособности отключения поиска озвучки.  1.Запустить приложение.  2.Выбрать перевод с русского на английский язык.  Шаги:  1.Нажать кнопку с настройками.  2.В открывшемся экране убрать флаг с пункта «Озвучка слов».  3.Нажать стрелку в левом верхнем углу экрана.  4.Ввести текст «еж».  5.Нажать кнопку «Перевести».  6.Нажать кнопку воспроизведения озвучки слева от кнопки «Перевести». | 1.Открывается экран с настройками приложения.  2.Пункт «Озвучка слов» имеет пустой квадрат.  3.Происходит возврат на главный экран приложения.  4.В поле ввода текста введен текст «еж».  5.Приложение отображает текст «hedgehog» в поле для вывода текста.  6.Кнопка не реагирует на нажатие, озвучка не воспроизводится. |
| Voicing | Средний | HomeScreen | translationButtonOnPressed | Проверка работоспособности воспроизведения озвучки.  1.Запустить приложение.  2.Выбрать перевод с русского на английский язык.  Шаги:  1.Ввести текст «еж».  2.Нажать кнопку «Перевести».  3.Нажать кнопку воспроизведения озвучки слева от кнопки «Перевести». | 1.В поле ввода текста введен текст «еж».  2.Приложение отображает текст «hedgehog» в поле для вывода текста.  3.Воспроизводится озвучка слова «hedgehog» на английском языке. |

Результаты тестирования соответствуют ожидаемому результату, следовательно, цель данного раздела, а именно – разработать и протестировать мобильное приложение – выполнена.

# ЗАКЛЮЧЕНИЕ

В процессе выполнения данной работы был проведен подробный анализ предметной области – онлайн-ресурсы для изучения английского языка. По результатам данного анализа были выявлены различные характерные черты этих онлайн-ресурсов, которые затем были использованы в качестве основного информационного источника при разработке мобильного приложения.

После анализа было выполнено проектирование приложения, в котором благодаря различным диаграммам было установлено требуемое поведение программы при различных вводимых данных согласно особенностям онлайн-ресурсов. Разработанные макеты экранных форм позволили в полной мере описать требования к внешнему виду пользовательского интерфейса мобильного приложения. Описанная модель базы данных дала избыточное описание для структуры единственной базы данных, используемой в приложении.

Далее была проведена разработка мобильного приложения с помощью фреймворка Flutter: были описаны разработанные экраны, особенности их взаимодействия и работы базы данных, а также особенности пользовательского интерфейса. После разработки проведено тестирование с помощью тест-кейсов, результаты которого продемонстрировали полную работоспособность приложения и соответствие реального поведения мобильного приложения с тем, которое было запланировано на этапе проектирование.

Поставленные задачи выполнены в полном объеме, а значит, цель работы достигнута.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Гражданский кодекс Российской Федерации. Часть 4 : федер. закон от 18 дек. 2006 г. №230-ФЗ : принят Гос. Думой 24 нояб. 2006 г. : одобр. Советом Федерации 8 дек. 2006 г. : введ. Федер. законом Рос. Федерации от 18 дек. 2006 г. №231-ФЗ // Российская газета. 2006. 22 дек. С 6.
2. Калгина И.С. Разработка мобильных приложений : учеб. пособие. Чита : ЗабГУ, 2022. 163 с.
3. Воронцов Ю.А., Овчинников М.А., Чернов Е.А. Платформы разработки мобильных приложений : учеб. пособие. М. : РТУ МИРЭА, 2023. 172 с.
4. Льюис Ш., Данн М. Нативная разработка мобильных приложений : руководство. М. : ДМК Пресс, 2020. 376 с.
5. Синицын И.В., Чернов Е.А., Воронцов Ю.А. Разработка мобильных приложений : учеб. пособие М. : РТУ МИРЭА, 2023. 162 с.
6. Федотенко М.А. Разработка мобильных приложений. Первые шаги : руководство. М. : Лаборатория знаний, 2019. 335 с.
7. Заяц А.М., Васильев Н.П. Введение в гибридные технологии разработки мобильных приложений : учеб. пособие для спо 2-е изд., стер. Санкт-Петербург : Лань, 2022. 160 с.
8. Баланов А.Н. Комплексное руководство по разработке: от мобильных приложений до веб-технологий : учеб. пособие для вузов. Санкт-Петербург : Лань, 2024. 412 с.
9. Баланов А.Н. Комплексное руководство по разработке: от мобильных приложений до веб-технологий : учеб. пособие. для спо. Санкт-Петербург : Лань, 2024. 64 с.
10. Черников В. Разработка мобильных приложений на C# для iOS и Android : учеб. пособие. М. : ДМК Пресс, 2020. 188 с.
11. Пирская Л.В. Разработка мобильных приложений в среде Android Studio : учеб. пособие. Ростов-на-Дону : ЮФУ, 2019. 123 с.
12. Локтев Д.А., Видьманов Д.А. Разработка кроссплатформенного мобильного приложения для работы с картами : учебно-методическое пособие М. : МГТУ им. Н.Э. Баумана, 2019. 32 с.
13. Попок Л.Е., Замотайлова Д.А., Савинская Д. Н. Разработка приложений под мобильные устройства: ОС iOS : учеб. пособие. Краснодар : КубГАУ, 2019. 90 с.

# ПРИЛОЖЕНИЕ А

**Блок-схемы алгоритма работы программного обеспечения**

На рисунке А.1 изображена главная форма мобильного приложения.

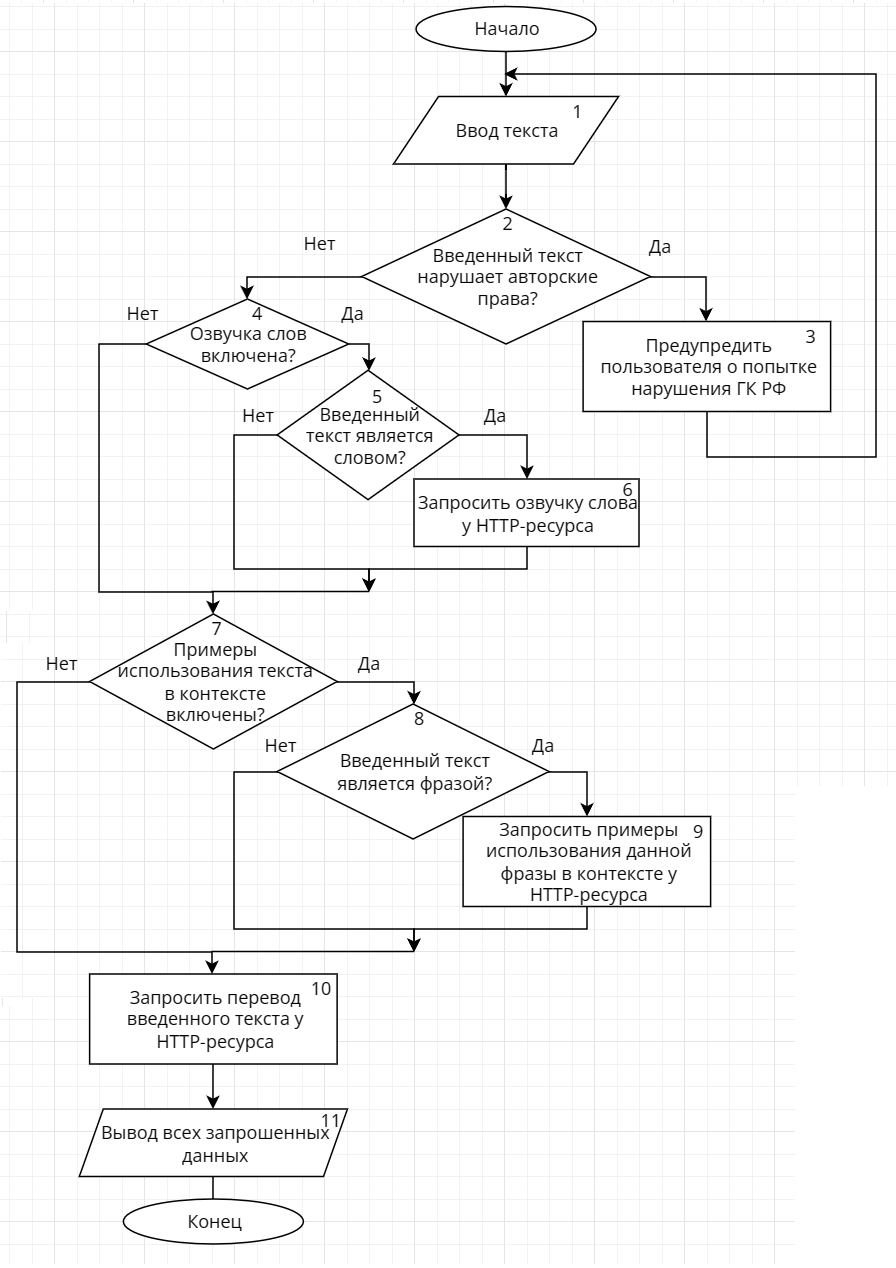


Рисунок А.1 – Блок-схема работы мобильного приложения

# ПРИЛОЖЕНИЕ Б

**Экранные формы мобильного приложения**

На рисунке Б.1 изображена главная форма мобильного приложения.

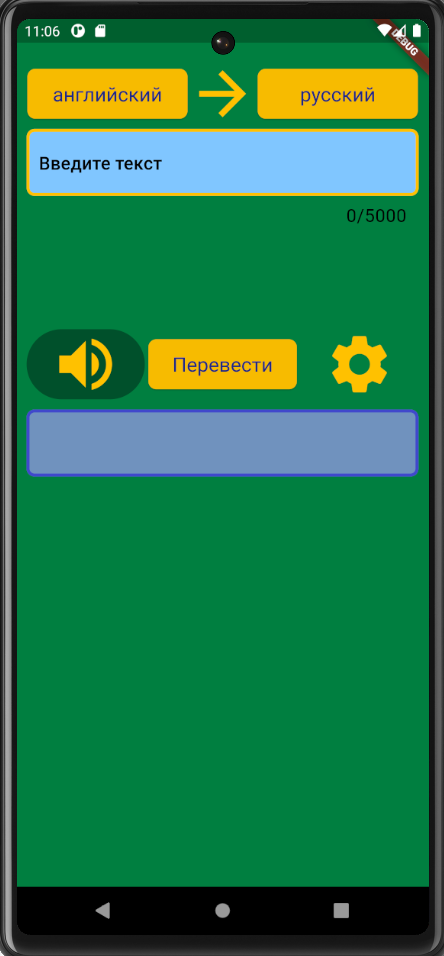


Рисунок Б.1 – главная форма мобильного приложения

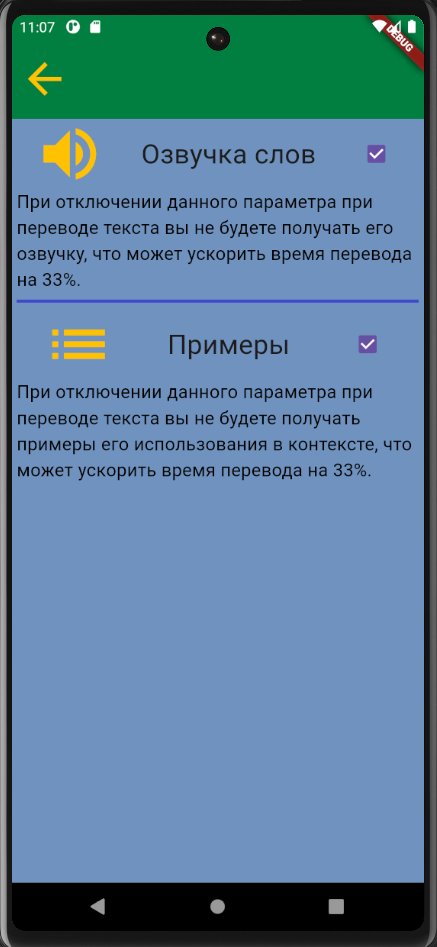


Рисунок Б.2 – форма с настройками мобильного приложения

# ПРИЛОЖЕНИЕ В

**Листинг программного кода**

1. Листнг файла main.dart.

import 'package:flutter/material.dart';

import 'package:test\_booble\_translator/home\_screen.dart';

import 'package:test\_booble\_translator/settings\_screen.dart';

import 'package:test\_booble\_translator/database.dart';

void main() async {

WidgetsFlutterBinding.ensureInitialized();

await DatabaseWithCopyrightTexts.createInstance();

runApp(

MaterialApp(

home: HomeScreen(),

),

);

}

1. Листнг файла home\_screen.dart.

import 'package:flutter/material.dart';

import 'package:flutter/services.dart';

import 'package:test\_booble\_translator/static\_objects.dart';

import 'package:test\_booble\_translator/settings\_screen.dart';

import 'package:test\_booble\_translator/type\_of\_data\_determinant.dart';

import 'package:test\_booble\_translator/database.dart';

import 'package:test\_booble\_translator/http\_requester.dart';

import 'package:assets\_audio\_player/assets\_audio\_player.dart';

class HomeScreen extends StatefulWidget {

HomeScreen({super.key});

@override

State createState() => \_HomeScreen();

}

class \_HomeScreen extends State<HomeScreen> {

static const ENGLISH\_LANGUAGE\_CAPTION = 'английский';

static const RUSSIAN\_LANGUAGE\_CAPTION = 'русский';

bool \_wordsVoicingEnabled = true;

bool \_examplesOfUsingEnabled = true;

bool get examplesAreUsed => \_examplesOfUsingEnabled && listOfExamples.isNotEmpty;

Widget buildExampleWithIndex(int index) {

var exampleIsExist = listOfExamples.elementAtOrNull(index) != null;

if (examplesAreUsed && exampleIsExist) {

return Column(

children: [

Padding(

padding: StaticObjects.paddingForExamples,

child: Text(

listOfExamples[index],

style: StaticObjects.inputAndOutputTextStyle,

),

),

StaticObjects.dividerForExamples,

],

);

} else {

return SizedBox();

}

}

bool translateToEnglish = false;

String voicingPath = '';

List<String> listOfExamples = [];

final TextEditingController enteredTextController = TextEditingController();

final TextEditingController translationTextController = TextEditingController();

get enteredText => enteredTextController.text;

get enteredTextTranslation => translationTextController.text;

set wordsVoicing(bool value) {

setState(() {

\_wordsVoicingEnabled = value;

});

}

set examplesOfUsing(bool value) {

setState(() {

\_examplesOfUsingEnabled = value;

});

}

static final AssetsAudioPlayer player = AssetsAudioPlayer();

void \_voicingButtonOnPressed() {

player.play();

}

Future<dynamic> copyrightViolationMessage() {

enteredTextController.clear();

return showDialog(

context: context,

barrierDismissible: false,

builder: (BuildContext context) {

return SizedBox(

height: 200,

child: AlertDialog(

insetPadding: EdgeInsets.symmetric(vertical: 200),

title: Text("Попытка нарушения ГК РФ", style: StaticObjects.inputAndOutputTextStyle),

content: Center(

child: Text(

"Вы попытались нарушить пункт 2.9 статьи 1270 ГК РФ «Исключительное право на произведение»: Использованием произведения независимо от того, совершаются ли соответствующие действия в целях извлечения прибыли или без такой цели, считается, в частности: перевод или другая переработка произведения. Дальнейший перевод не будет выполняться. Удачного изучения английского языка!")),

actions: [

Center(

child: ElevatedButton(

child: Text("Я больше не буду пытаться нарушить закон"),

onPressed: () {

Navigator.pop(context);

},

),

),

],

),

);

},

);

}

@override

void dispose() {

enteredTextController.dispose();

translationTextController.dispose();

super.dispose();

}

@override

Widget build(BuildContext context) {

// This method prevent the screen rotation

SystemChrome.setPreferredOrientations([

DeviceOrientation.portraitUp,

]);

return MaterialApp(

debugShowCheckedModeBanner: false,

title: 'Booble translator',

home: Scaffold(

resizeToAvoidBottomInset: false,

body: Center(

child: Padding(

padding: const EdgeInsets.only(left: 10, right: 10, top: 40, bottom: 0),

child: Column(

mainAxisAlignment: MainAxisAlignment.start,

children: [

// английский -> русский

Row(

mainAxisAlignment: MainAxisAlignment.center,

children: [

SizedBox(

width: 160.0,

height: 50,

child: ElevatedButton(

onPressed: () {

setState(() {

translateToEnglish = !translateToEnglish;

});

},

style: StaticObjects.usualButtonStyle,

child: Text(translateToEnglish ? RUSSIAN\_LANGUAGE\_CAPTION : ENGLISH\_LANGUAGE\_CAPTION),

),

),

Icon(

Icons.arrow\_forward,

color: Colors.amber,

size: 70,

),

SizedBox(

width: 160.0,

height: 50,

child: ElevatedButton(

onPressed: () {

setState(() {

translateToEnglish = !translateToEnglish;

});

},

style: StaticObjects.usualButtonStyle,

child: Text(translateToEnglish ? ENGLISH\_LANGUAGE\_CAPTION : RUSSIAN\_LANGUAGE\_CAPTION),

),

),

],

),

// Box for text input

SizedBox(

height: 200,

child: TextField(

keyboardType: TextInputType.multiline,

maxLines: null,

maxLength: 5000,

cursorColor: Colors.black,

style: StaticObjects.inputAndOutputTextStyle,

controller: enteredTextController,

decoration: InputDecoration(

filled: true,

fillColor: const Color.fromARGB(255, 128, 198, 255),

enabledBorder: StaticObjects.bordersForTextFieldForInput,

focusedBorder: StaticObjects.bordersForTextFieldForInput,

hintStyle: TextStyle(

color: Colors.black,

),

counterStyle: StaticObjects.inputAndOutputTextStyle,

hintText: "Введите текст",

),

),

),

// listen translate and settings

Row(

mainAxisAlignment: MainAxisAlignment.spaceBetween,

children: [

// voicing button

ElevatedButton(

onPressed: \_wordsVoicingEnabled && translateToEnglish ? \_voicingButtonOnPressed : null,

style: StaticObjects.styleOfEmptyButton,

child: StaticObjects.iconWithVolumeUp,

),

// translation button

SizedBox(

height: 50,

child: ElevatedButton(

onPressed: translationButtonOnPressed,

style: StaticObjects.usualButtonStyle,

child: Text('Перевести'),

),

),

// Settings button

ElevatedButton(

onPressed: () async {

var data = await Navigator.push(

context,

MaterialPageRoute(

builder: (BuildContext context) => SettingsScreen.withData([\_wordsVoicingEnabled, \_examplesOfUsingEnabled]),

),

);

wordsVoicing = data[0] as bool;

examplesOfUsing = data[1] as bool;

},

style: StaticObjects.styleOfEmptyButton,

child: Icon(

Icons.settings,

size: 70.0,

color: Colors.amber,

),

),

],

),

// translation with examples

Expanded(

child: Padding(

padding: EdgeInsets.only(top: 10),

child: ListView(

padding: EdgeInsets.zero,

scrollDirection: Axis.vertical,

children: [

TextField(

readOnly: true,

maxLines: null,

cursorColor: Colors.black,

controller: translationTextController,

decoration: InputDecoration(

filled: true,

fillColor: Color.fromARGB(255, 112, 146, 190),

enabledBorder: StaticObjects.borderForOutputField,

focusedBorder: StaticObjects.borderForOutputField,

),

style: StaticObjects.inputAndOutputTextStyle,

),

Padding(

padding: EdgeInsets.only(bottom: 40),

child: Card(

margin: EdgeInsets.only(top: 10),

shape: RoundedRectangleBorder(

side: BorderSide(

color: Color.fromARGB(255, 63, 72, 204),

width: 3.0,

),

borderRadius: BorderRadius.circular(8),

),

// background color

color: Color.fromARGB(255, 127, 127, 127),

child: Column(

mainAxisSize: MainAxisSize.min,

children: buildExamples(),

),

),

),

],

),

),

),

],

),

),

),

backgroundColor: Color.fromARGB(255, 6, 127, 64),

),

color: Colors.green,

);

}

void translationButtonOnPressed() async {

bool isCopyrightText = await DatabaseWithCopyrightTexts.isCopyrightText(enteredText);

if (isCopyrightText) {

copyrightViolationMessage();

setState(() {});

return;

}

translationTextController.text = HTTP\_requester.getTranslation(enteredText, translateToEnglish);

setState(() {});

if (TypeOfDataDeterminant.isWord(enteredTextTranslation)) {

voicingPath = HTTP\_requester.getVoicingPath(enteredTextTranslation);

player.open(Audio(voicingPath));

setState(() {});

}

if (\_examplesOfUsingEnabled && TypeOfDataDeterminant.isPhrase(enteredText)) {

listOfExamples = HTTP\_requester.getExamplesFor(enteredTextTranslation);

} else {

listOfExamples.clear();

}

setState(() {});

}

List<Widget> buildExamples() {

List<Widget> examples = [];

for (int i = 0; i < 5; i++) {

examples.add(buildExampleWithIndex(i));

}

return examples;

}

}

1. Листнг файла settings\_screen.dart.

import 'package:flutter/material.dart';

import 'package:test\_booble\_translator/static\_objects.dart';

class SettingsScreen extends StatefulWidget {

SettingsScreen({super.key}) : super();

var \_data;

SettingsScreen.withData(this.\_data);

@override

\_SettingsScreen createState() => \_SettingsScreen(\_data);

}

class \_SettingsScreen extends State<SettingsScreen> {

var \_wordsVoicing;

var \_examplesOfUsing;

\_SettingsScreen(\_data) {

\_wordsVoicing = \_data[0];

\_examplesOfUsing = \_data[1];

}

@override

Widget build(BuildContext context) {

return MaterialApp(

home: Scaffold(

appBar: AppBar(

toolbarHeight: 80,

backgroundColor: Color.fromARGB(255, 6, 127, 64),

leading: IconButton(

icon: Icon(

Icons.arrow\_back,

color: Colors.amber,

size: 50,

),

onPressed: () {

Navigator.pop(context, [\_wordsVoicing, \_examplesOfUsing]);

},

),

),

body: Center(

child: Padding(

padding: EdgeInsets.symmetric(horizontal: 5),

child: Column(

children: [

Row(

mainAxisAlignment: MainAxisAlignment.spaceAround,

children: [

StaticObjects.iconWithVolumeUp,

Text(

"Озвучка слов",

style: TextStyle(fontSize: 27),

),

Checkbox(

value: \_wordsVoicing,

onChanged: (bool? value) {

setState(

() {

\_wordsVoicing = value ?? false;

},

);

},

),

],

),

Text(

"При отключении данного параметра при переводе текста вы не будете получать его озвучку, что может ускорить время перевода на 33%.",

style: StaticObjects.inputAndOutputTextStyle,

),

StaticObjects.dividerForExamples,

Row(

mainAxisAlignment: MainAxisAlignment.spaceAround,

children: [

Icon(

Icons.list,

color: Colors.amber,

size: 70,

),

Text(

softWrap: true,

"Примеры",

style: TextStyle(fontSize: 27),

),

Checkbox(

value: \_examplesOfUsing,

onChanged: (bool? value) {

setState(

() {

\_examplesOfUsing = value ?? false;

},

);

},

),

],

),

Text(

"При отключении данного параметра при переводе текста вы не будете получать примеры его использования в контексте, что может ускорить время перевода на 33%.",

style: StaticObjects.inputAndOutputTextStyle,

),

],

),

),

),

backgroundColor: Color.fromARGB(255, 112, 146, 190),

),

);

}

}

1. Листнг файла copyright\_text\_model.dart.

class CopyrightText {

final String copyrightText;

CopyrightText({required this.copyrightText});

Map<String, Object?> toMap() {

return {

'copyright\_text': copyrightText,

};

}

@override

String toString() {

return '{copyright\_text: $copyrightText}';

}

}

1. Листнг файла database.dart.

import 'dart:async';

import 'package:path/path.dart';

import 'package:sqflite/sqflite.dart';

import 'package:test\_booble\_translator/copyright\_text\_model.dart';

class DatabaseWithCopyrightTexts {

static late Future<Database> database;

DatabaseWithCopyrightTexts.\_origin();

static createInstance() async {

database = openDatabase(

join(await getDatabasesPath(), 'copyrightTexts.db'),

onCreate: (db, version) {

return db.execute(

'CREATE TABLE copyright\_text(copyright\_text TEXT)',

);

},

version: 1,

);

// Данный код использовался для однократной инициализации базы данных и ее содержимого

// deleteDatabase(join(await getDatabasesPath(), 'copyrightTexts.db'));

// await deleteAllCopyrightTexts();

// var db = await database;

// var texts\_with\_copyright = [

// CopyrightText(

// copyrightText:

// 'ТОВАР-УЧАСТНИК АКЦИИ «ФИНАЛЬНАЯ ЦЕНА» — ДОПОЛНИТЕЛЬНЫЕ СКИДКИ НЕ ПРИМЕНЯЮТСЯ. Является источником необходимых Омега-6 жирных кислот с правильным сочетанием минеральных веществ и витаминов. 100% сбалансированный рацион. Баланс минералов. Содержит витамины Д и Е.',

// ),

// CopyrightText(

// copyrightText:

// 'Любимый нежный вкус для вашего кота! После обеда с новыми силами можно продолжать домашние приключения! Влажный корм для кошек Kitekat с сочными кусочками курицы в соусе — это порция сочных кусочков с курицей, приготовленных по особому рецепту. В его основе — формула сбалансированного питания, которая содержит белки, минералы, витамины, таурин и животные жиры. Kitekat — это идеальный источник жизненных сил для кошки.',

// ),

// CopyrightText(

// copyrightText:

// 'ТОВАР-УЧАСТНИК АКЦИИ «ФИНАЛЬНАЯ ЦЕНА» — ДОПОЛНИТЕЛЬНЫЕ СКИДКИ НЕ ПРИМЕНЯЮТСЯ. Влажный корм для кошек Perfect Fit с курицей в соусе Для стерилизованных кошек разработан по специальной рецептуре для контроля здоровья кошек, прошедших стерилизацию. Поддерживает здоровье мочевыводящей системы у стерилизованных кошек, подверженных повышенному риску развития мочекаменной болезни. Поддержание здорового веса. Специальный ингредиент L-карнитин способствует сжиганию жиров, помогая поддерживать здоровый вес.',

// ),

// CopyrightText(

// copyrightText:

// 'ТОВАР-УЧАСТНИК АКЦИИ «ФИНАЛЬНАЯ ЦЕНА» — ДОПОЛНИТЕЛЬНЫЕ СКИДКИ НЕ ПРИМЕНЯЮТСЯ. Влажный корм для котят Whiskas полнорационный от 1 до 12 месяцев Рагу с курицей — это сбалансированный рацион для здорового роста вашего малыша. Котенок с удовольствием попробует маленькие нежные кусочки с курицей в ароматном соусе. Все для замурррчательной жизни! Рационы Whiskas для котят позаботятся об иммунитете, поддержат здоровое пищеварение и обеспечат здоровый рост и развитие. Котята активно растут и с первых дней нуждаются в правильном питании. Корм Whiskas для котят создан с учетом всех особенностей развития организма котенка. Для здорового роста и правильного развития корм Whiskas для котят содержит оптимальное сочетание питательных веществ, витаминов и минералов. Содержит 100% натуральные высококачественные ингредиенты: мясо и субпродукты без добавления ароматизаторов, искусственных красителей и консервантов, кальций для здоровья костей, омега-6 и цинк для здоровья кожи и шерсти, витамин А и таурин для хорошего зрения, витамины D, E и минералы. Ваш котенок будет мурчать от удовольствия!',

// ),

// CopyrightText(

// copyrightText:

// 'Влажный корм для стерилизованных кошек Purina One с курицей и фасолью в соусе помогает поддерживать здоровую мочевыделительную систему, уменьшая концентрацию минералов, которые могут привести к образованию камней в почках. Разработан специально для поддержания здорового обмена веществ после стерилизации. Помогает поддерживать естественную защиту организма благодаря антиоксидантам, таким как витамин Е. С клетчаткой, которая доказанно улучшает здоровье пищеварительной системы. Курица высокого качества — источник белка и отличного вкуса. Без добавления искусственных красителей, консервантов и ароматизаторов.',

// ),

// ];

// for (var text in texts\_with\_copyright) {

// await insertCopyrightText(text);

// }

// var texts = await copyrightTexts();

// for(var text in texts) {

// print(text);

// }

// insertCopyrightText(CopyrightText(copyrightText: 'some text with copyright'));

}

static Future<void> insertCopyrightText(CopyrightText copyrightText) async {

final db = await database;

await db.insert(

'copyright\_text',

copyrightText.toMap(),

conflictAlgorithm: ConflictAlgorithm.replace,

);

}

static Future<List<String>> copyrightTexts() async {

final db = await database;

final List<Map<String, Object?>> copyrightTextMaps =

await db.query('copyright\_text');

var texts = <String>[];

for (var map in copyrightTextMaps) {

texts.add(map['copyright\_text'] as String);

}

return texts;

}

static Future<void> updateCopyrightText(CopyrightText copyrightText) async {

final db = await database;

await db.update(

'copyright\_text',

copyrightText.toMap(),

where: 'copyright\_text = ?',

whereArgs: [copyrightText.copyrightText],

);

}

static Future<void> deleteAllCopyrightTexts() async {

final db = await database;

db.delete('copyright\_text');

}

static Future<void> deleteCopyrightText(String copyrightText) async {

final db = await database;

await db.delete(

'copyright\_text',

where: 'copyright\_text = ?',

whereArgs: [copyrightText],

);

}

static Future<bool> isCopyrightText(String text) async {

List<String> texts = await copyrightTexts();

for(int i = 0; i < texts.length; i++) {

texts[i] = texts[i].toLowerCase();

}

return texts.contains(text);

}

}

1. Листнг файла http\_requester.dart.

// ignore: camel\_case\_types

class HTTP\_requester {

HTTP\_requester.\_origin();

static final Map<String, String> \_russianToEnglishTranslations = {

'еж': 'hedgehog',

'привет мир': 'hello world',

};

static final Map<String, String> \_englishToRussianTranslations = {'hedgehog': 'еж', 'hello world': 'привет мир'};

static String getTranslation(String text, bool \_translateToEnglish) {

return (\_translateToEnglish ? \_russianToEnglishTranslations[text] : \_englishToRussianTranslations[text]) ?? 'lol';

}

static final Map<String, List<String>> \_examples = {

'привет мир': [

'Пример программы "привет мир" на языке Java',

'Запустите компьютер и написать функцию "Привет Мир".',

'Это поможет вам продвинуться от ваших начальных "привет мир" к тому, чем вы действительно сможете потом гордиться.',

'Теперь давайте запустим! в настоящее время, Наша программа не будет делать ничего, кроме открытой, сказать "привет мир" и закрытие.',

'Входными данными может быть любой набор символов от простого «привет мир» до сонета Шекспира.',

],

'hello world' : [

'The usefulness of the "Hello World" programs shown in the previous chapter is rather questionable.',

'When we wrote the first Hello World program and fired it up, it was extremely exciting.',

'This will write "hello world" on the display.',

'All of this happens behind the scenes and as a user, all you will see are the words "hello world".',

'Forums: "Hello world" is not deleted.',

],

'hedgehog' : [

'Why hedgehog, we still do not know, but the unfortunate animal was offended.',

'Among the protected mammals listed are hedgehog, pine-martens, bats.',

'I think that was a real hedgehog.',

'Not a hedgehog or a squirrel, the king-father walks.',

'But intellectually Soros is more like a narrow-minded hedgehog.',

],

'еж' : [

'Почти месяц еж жил роскошной жизнью в квартире женщины.',

'Старающийся спрятаться от вас еж снится к возникновению необходимости быть предельно осторожными и терпеливыми.',

'Считается, что еж мало изменился за этот период времени.',

'На каждом уровне, Алиса должна забить синий еж в норе.',

'Синий еж Соник вызвал одобрение у общества.',

]

};

// pattern null self handling

static List<String> getExamplesFor(String text) => \_examples[text] ?? [];

static getVoicingPath(String word) => 'lib/pronunciations/$word.mp3';

}

1. Листнг файла static\_objects.dart.

import 'package:flutter/material.dart';

class StaticObjects {

StaticObjects.\_origin();

static final List<String> statesOfButton = ["Вкл.", "Выкл."];

static final ButtonStyle usualButtonStyle = ElevatedButton.styleFrom(

textStyle: TextStyle(

fontSize: 20,

),

foregroundColor: Color.fromARGB(255, 31, 38, 137),

backgroundColor: Colors.amber,

shape: RoundedRectangleBorder(

borderRadius: BorderRadius.circular(8.0),

),

);

static final OutlineInputBorder borderForOutputField = OutlineInputBorder(

borderSide: BorderSide(

color: Color.fromARGB(255, 63, 72, 204),

width: 3.0,

),

borderRadius: BorderRadius.circular(8),

);

static final TextStyle inputAndOutputTextStyle = TextStyle(

color: Colors.black,

fontSize: 18,

);

static final EdgeInsets paddingForExamples =

EdgeInsets.only(left: 10, right: 10);

static Divider dividerForExamples = Divider(

color: Color.fromARGB(255, 63, 72, 204),

thickness: 3.0,

);

static OutlineInputBorder bordersForTextFieldForInput = OutlineInputBorder(

borderSide: BorderSide(

color: Colors.amber,

width: 3.0,

),

borderRadius: BorderRadius.circular(8),

);

static ButtonStyle styleOfEmptyButton = ElevatedButton.styleFrom(

backgroundColor: Color.fromARGB(255, 6, 127, 64),

foregroundColor: Color.fromARGB(255, 6, 127, 64),

elevation: 0,

disabledBackgroundColor: Color.fromARGB(255, 4, 80, 41),

);

static Icon iconWithVolumeUp = Icon(

Icons.volume\_up,

size: 70.0,

color: Colors.amber,

);

}

1. Листнг файла type\_of\_data\_determinant.dart.

class TypeOfDataDeterminant {

TypeOfDataDeterminant.\_origin();

static bool isWord(String text) {

return text.split(' ').length == 1;

}

static bool isPhrase(String text) {

int wordsAmount = text.split(' ').length;

return wordsAmount <= 5;

}

}

# ПРИЛОЖЕНИЕ Г

**Результаты выполнения тест-кейсов**

На рисунках Г.1-Г.3 изображены результаты выполнения первого тест-кейса.

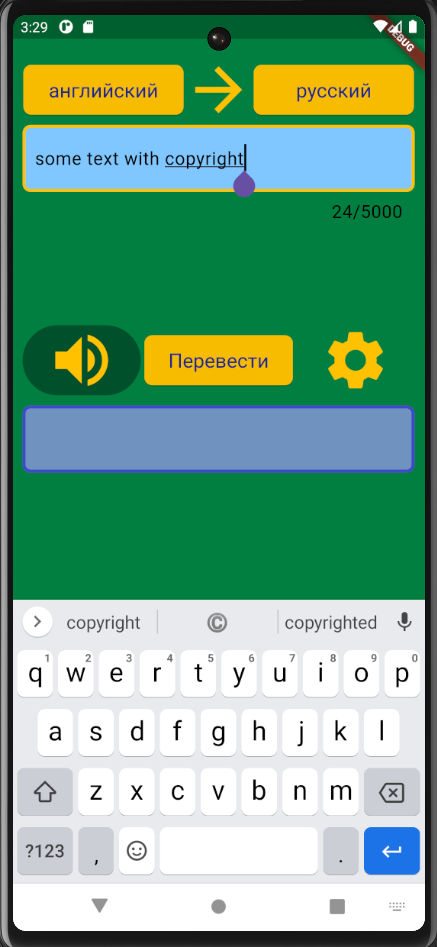


Рисунок Г.1 – Ввод текста с авторскими правами

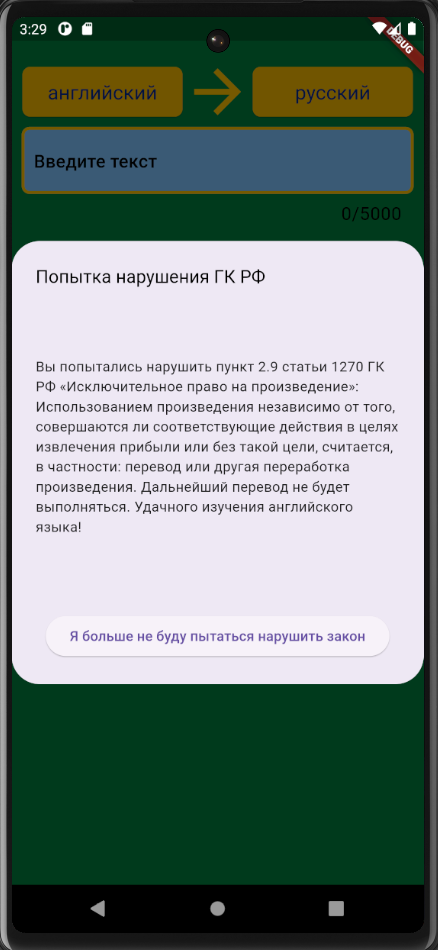
****

Рисунок Г.2 – Отображение окна с надписью о попытке нарушения ГК РФ

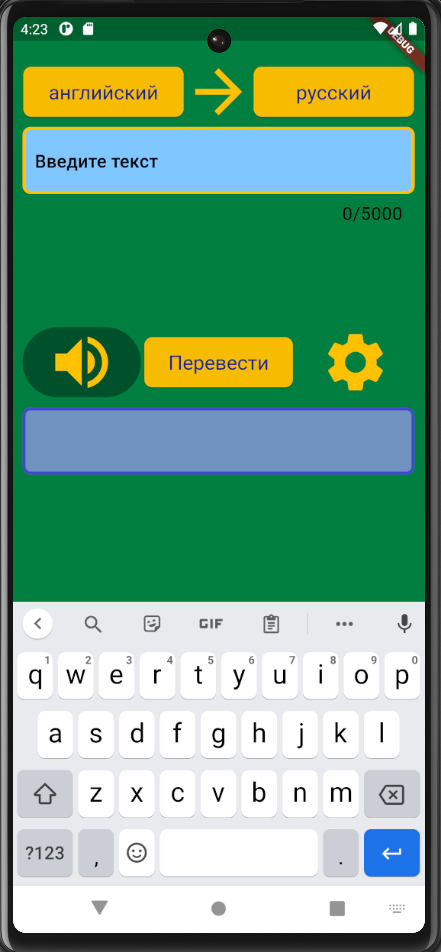
****

Рисунок Г.3 – Поле с введенным текстом очищено

На рисунке Г.4 изображены результаты выполнения второго тест-кейса.

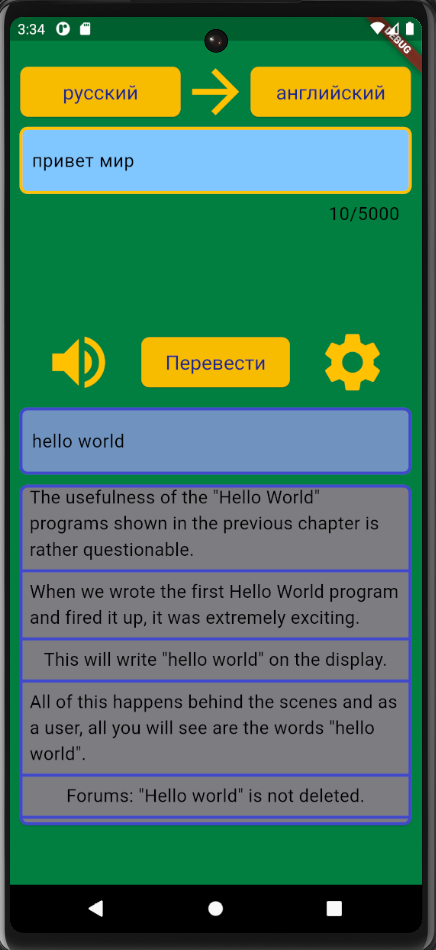
****

Рисунок Г.4 – Отображение перевода текста «привет мир»

На рисунке Г.5 изображены результаты выполнения третьего тест-кейса.

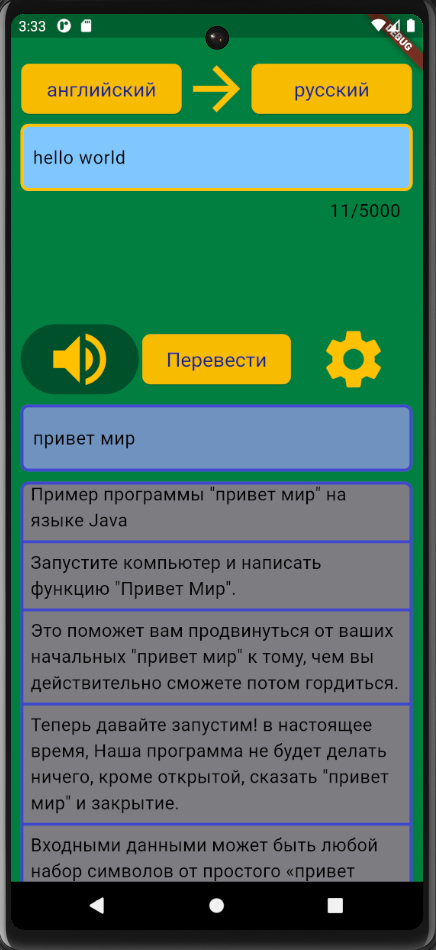
****

Рисунок Г.5 – Отображение перевода текста «hello world»

На рисунках Г.6-Г.7 изображены результаты выполнения четвертого тест-кейса.

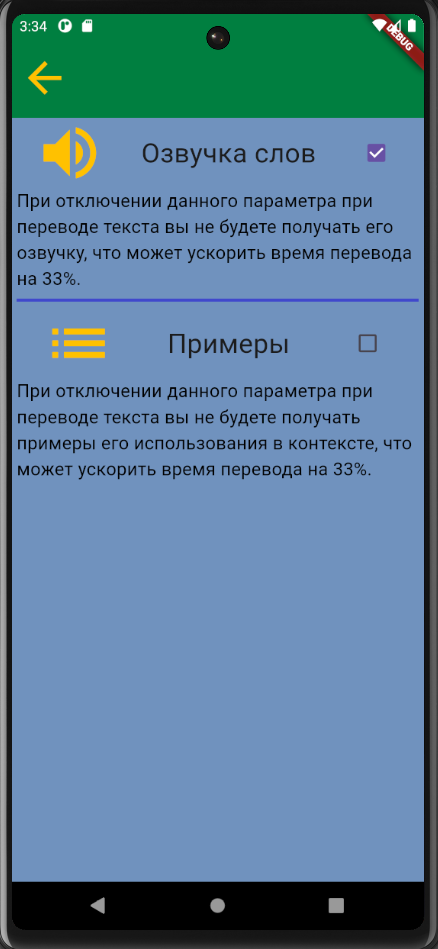
****

Рисунок Г.6 – Примеры выключены

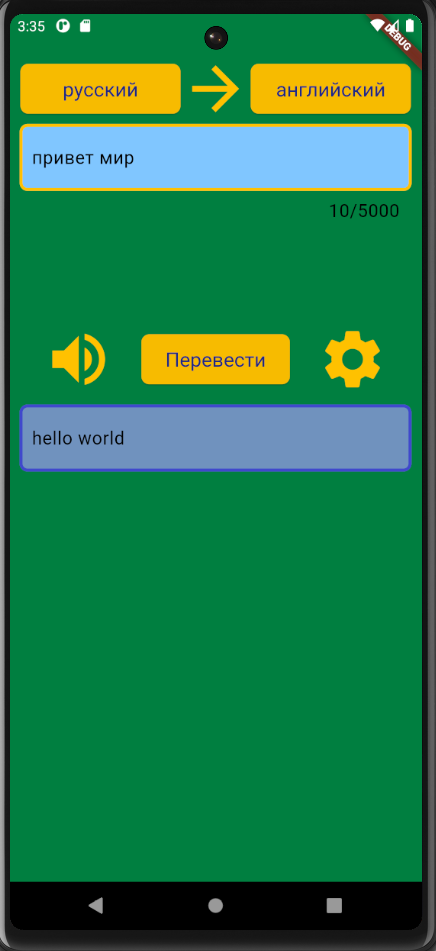
****

Рисунок Г.7 – Отсутствие отображения примеров

На рисунке Г.8–Г10 изображены результаты выполнения пятого тест-кейса.

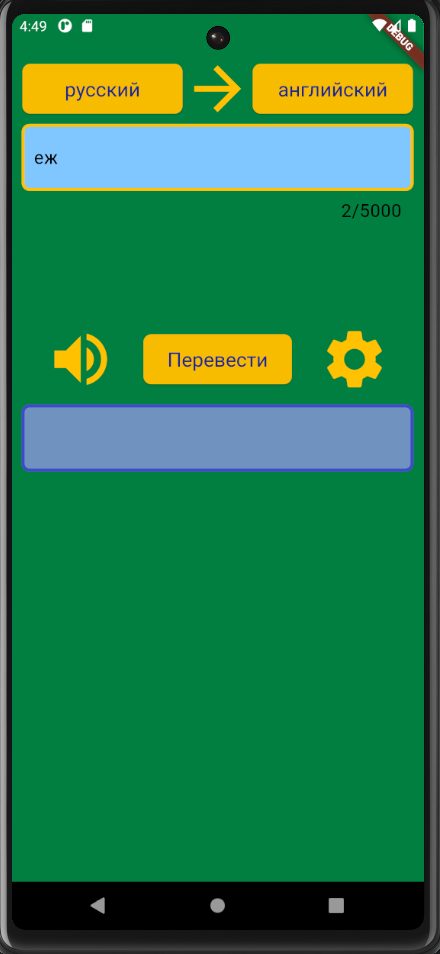
****

Рисунок Г.8 – Кнопка озвучки включена

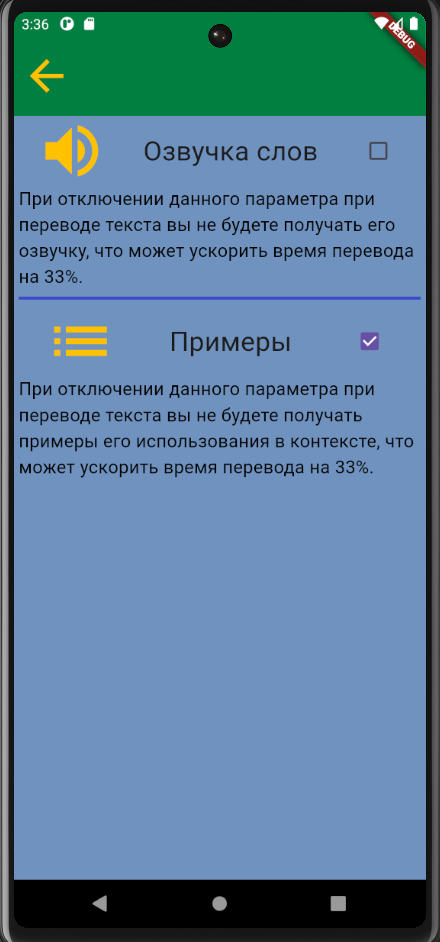
****

Рисунок Г.9 – Отключение озвучки слов

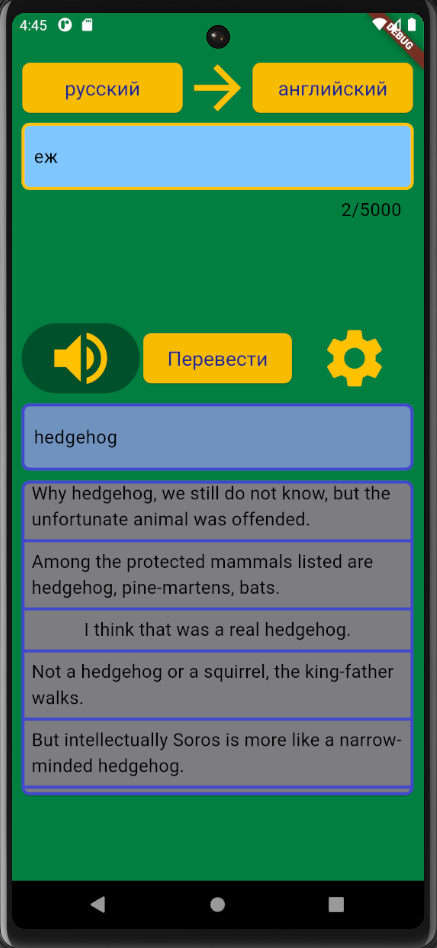
****

Рисунок Г.10 – Кнопка озвучки стала неактивной после перевода

На рисунке Г.11 изображены результаты выполнения шестого тест-кейса.

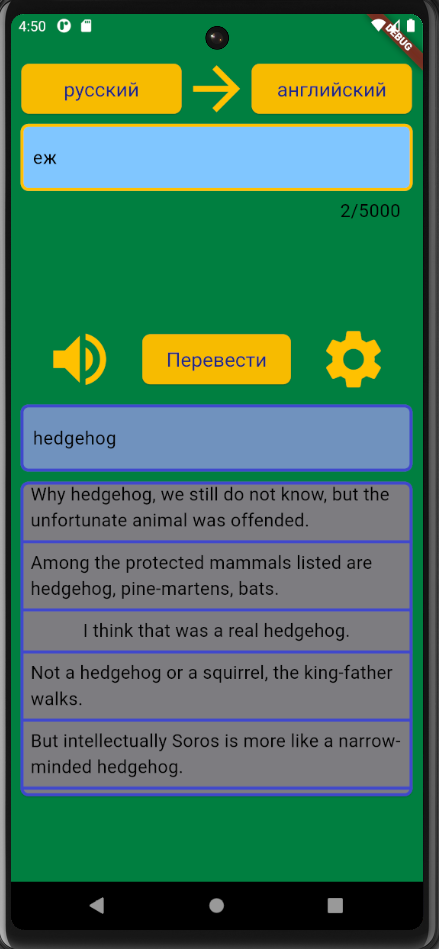
****

Рисунок Г.11 –Воспроизведение озвучки слова «hedgehog» при нажатии на кнопку озвучки