

Министерство просвещения РФ  
федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Уральский государственный педагогический университет»  
Институт математики, физики, информатики  
Кафедра информатики, информационных технологий  
и методики обучения информатике

# **МОБИЛЬНОЕ ПРИЛОЖЕНИЕ ДЛЯ ИЗУЧЕНИЯ ИСПАНСКОГО ЯЗЫКА**

Выпускная квалификационная работа

Допущено к защите

Зав. кафедрой

\_\_\_\_\_

дата

подпись

Исполнитель: Васильев Алексей Александрович  
студент группы ИСиТ-2031

\_\_\_\_\_

Руководитель: к.п.н., доцент  
зав. кафедры ИИТ и МОИ  
Сардак Л.В.

\_\_\_\_\_

Екатеринбург – 2024

## Оглавление

<b>ВВЕДЕНИЕ.....</b>	<b>3</b>
<b>ГЛАВА 1. АНАЛИЗ ТЕХНОЛОГИЙ СОЗДАНИЯ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ ДЛЯ ИЗУЧЕНИЯ ИНОСТРАННЫХ ЯЗЫКОВ .....</b>	<b>5</b>
1.1    КЛАССИФИКАЦИЯ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ ДЛЯ ИЗУЧЕНИЯ ИНОСТРАННОГО ЯЗЫКА	5
1.2    ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ ПРОЕКТИРОВАНИЯ И РАЗРАБОТКИ .....	20
1.3    ФОРМАЛИЗОВАННОЕ ОПИСАНИЕ ТЕХНИЧЕСКОГО ЗАДАНИЯ.....	23
<b>ГЛАВА 2. РЕАЛИЗАЦИЯ МОБИЛЬНОГО ПРИЛОЖЕНИЯ ДЛЯ ИЗУЧЕНИЯ ИСПАНСКОГО ЯЗЫКА .....</b>	<b>27</b>
2.1    ФУНКЦИОНАЛЬНАЯ МОДЕЛЬ МОБИЛЬНОГО ПРИЛОЖЕНИЯ.....	27
2.2    ОПИСАНИЕ ПРОДУКТА .....	37
2.3    АПРОБАЦИЯ МАТЕРИАЛОВ РАБОТЫ.....	51
<b>ЗАКЛЮЧЕНИЕ .....</b>	<b>53</b>
<b>СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ.....</b>	<b>54</b>
<b>ПРИЛОЖЕНИЯ .....</b>	<b>58</b>
<i>Приложение 1. Листинг .....</i>	58

## Введение

Использование цифровых технологий в обучении иностранным языкам сегодня является неотъемлемой частью учебного процесса в вузе. Существует достаточно большое количество различных программ, посвященных вопросам обучения иностранному языку, однако они в наибольшей степени ориентированы на изучение английского, французского или немецкого языков. В УрГПУ реализуется основная образовательная программа, включающая изучение испанского языка по определенной методике, в поддержку которой кафедра «Английской филологии и методики преподавания английского языка» сделала заказ на разработку мобильного приложения с определенным функционалом. Приложение должно работать на смартфоне на платформе «Android».

Необходимо разработать приложение, позволяющее преподавателю проверять усвоение студентами лексических единиц в тестовом формате по испанскому языку. База включает в себя тему, текст (испанское написание слова), изображение объекта и аудио объекта. На основе этой базы приложение должно генерировать серию заданий на ввод текста по изображению или на ввод текста по аудио (режим тренажера, режим теста с отправкой результата преподавателю).

Продукт разработки – мобильное приложение для изучения лексических единиц на испанском языке, разбитых на тематические группы.

Цель работы – спроектировать и разработать мобильное приложение для изучения лексических единиц по темам «Флора и Фауна» под операционную систему «Android».

Задачи:

1. Произвести анализ существующих мобильных приложений для изучения иностранного языка.
2. Проанализировать технологии реализации и программные решения для проектирования и разработки мобильного приложения по ОС «Android».
3. Подготовить техническое задание.

4. В соответствии с техническим заданием провести проектирование и разработку мобильного приложения.
5. Провести апробацию приложения.

# **Глава 1. Анализ технологий создания мобильных приложений для изучения иностранных языков**

## **1.1 Классификация мобильных приложений для изучения иностранного языка**

На данный момент на рынке программного обеспечения (ПО) есть множество приложений для изучения языков. Каждый из них имеет разный функционал, и предназначены для разного уровня знания языка.

Рассмотрим, какие приложения есть на рынке ПО для изучения иностранных языков.

Первой из таких программ будет служить бесплатное приложение «Duolingo», которое доступно на Android/iOS.

«Одно из лучших бесплатных приложений для изучения иностранных языков. Начинаете с самых азов и постепенно совершенствуете навыки чтения, письма, аудирования и разговора. Обучение проходит в игровой форме: нужно выполнять короткие задания и отвечать на вопросы, чтобы прокачать грамматику и расширить словарный запас» [2].

Приложение позволяет изучить множество иностранных языков: английский, немецкий, французский [2].

Система предлагает пользователю несколько типов обучения, например:

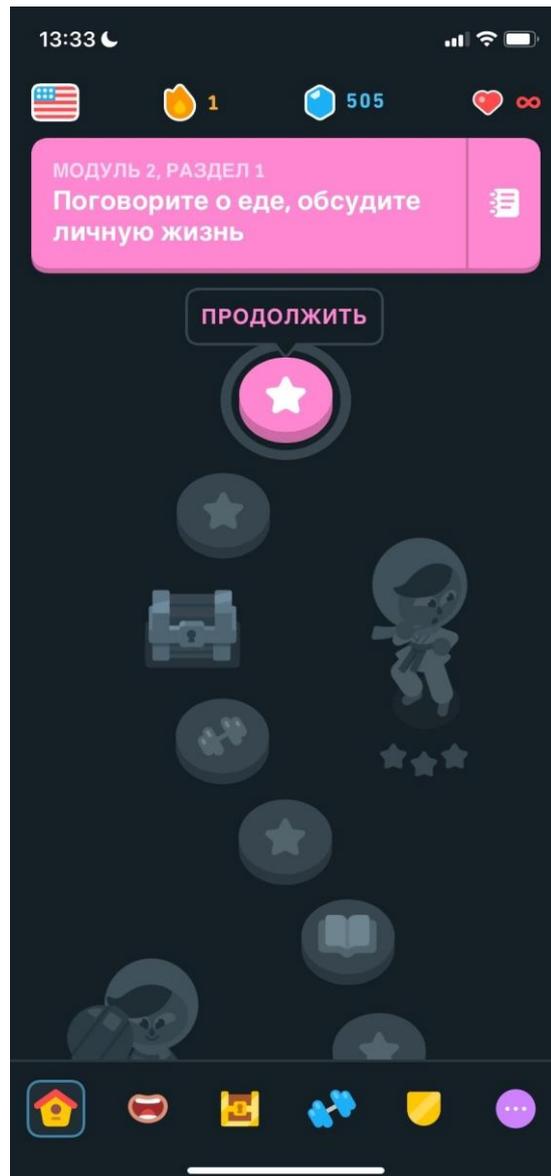
«Словарь: вам покажут изображение предмета и попросят назвать его.

Перевод: вам будет предложено перевести слово или фразу с вашего языка на изучаемый, или наоборот.

Прослушивание: вы услышите звуковое воспроизведение слова или фразы и попросят ввести его» [3].

ПО предлагает пользователю блоки и в них некоторое количество уроков:

«Внутри каждого блока навыков есть несколько уроков. Нажмите на первый, чтобы начать работу» [3].



*Рисунок 1. Снимок экрана приложения Duolingo*

Вторым примером будет служить платное приложение «Busuu», которое доступно в iTunes/Google Play.

«Busuu — это интерактивные онлайн-уроки, направленные на изучение типичных разговорных ситуаций с помощью подробного разбора диалогов. Вы не приобретете здесь большой словарный запас и не изучите грамматику даже в объеме школьной программы — упор делается на заучивание разговорных фраз. Наиболее интересной возможностью сервиса мне кажется взаимодействие с другими участниками» [4].

Приложение помогает изучить около 14 языков для разговорной речи, такие как: английский, французский, немецкий и т.д. Есть и бесплатная версия, но там доступно только три языка: английский, французский, немецкий. [5].

Пользователю предлагается несколько видов уроков: заучивание слов, озвученные задания, грамматика, запись голоса [4].

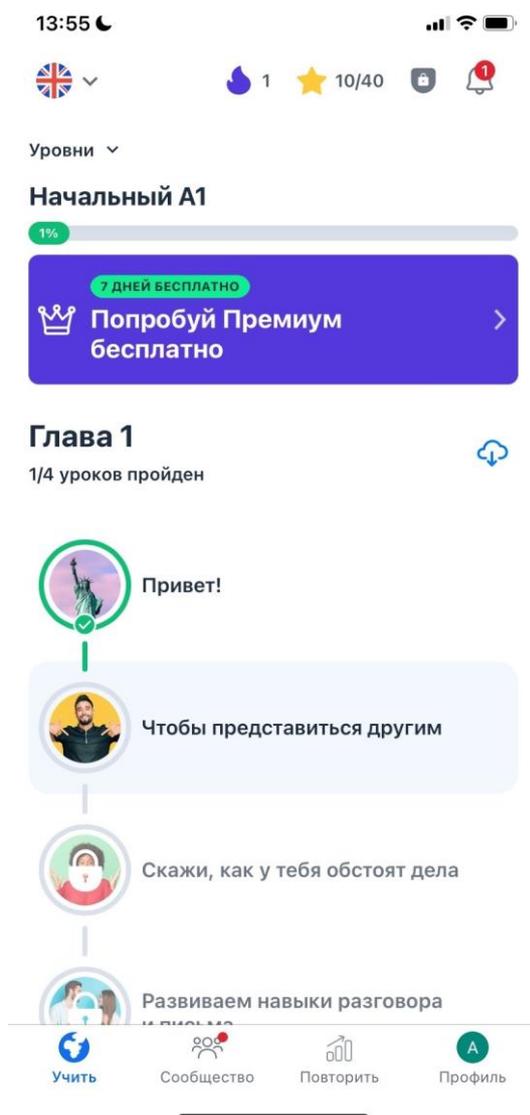


Рисунок 2. Снимок экрана приложения Visuu

Третьим примером служит платное приложение «СЛОВА БЕГОМ». Оно доступно для Android/iOS.

«Быстро и эффективно изучить языки поможет мультиплатформенное приложение «СЛОВА БЕГОМ» от компании M.A.R.I.D.I. Основной особенностью программных продуктов компании является использование для самостоя-

тельного обучения уникальной мемо-методики M.A.R.I.D.I, в которой задействуются основные человеческие системы восприятия информации: зрение, слух и тактильные ощущения» [6].

Как и в предыдущем примере, «СЛОВА БЕГОМ», может предложить на изучение несколько видов языков, такие как: английский, французский, немецкий и т.д. [6].

Система дает на выбор несколько методик для изучения иностранного языка, некоторые из них: запоминание слов, виртуальный учитель, тренажер правописания [6].

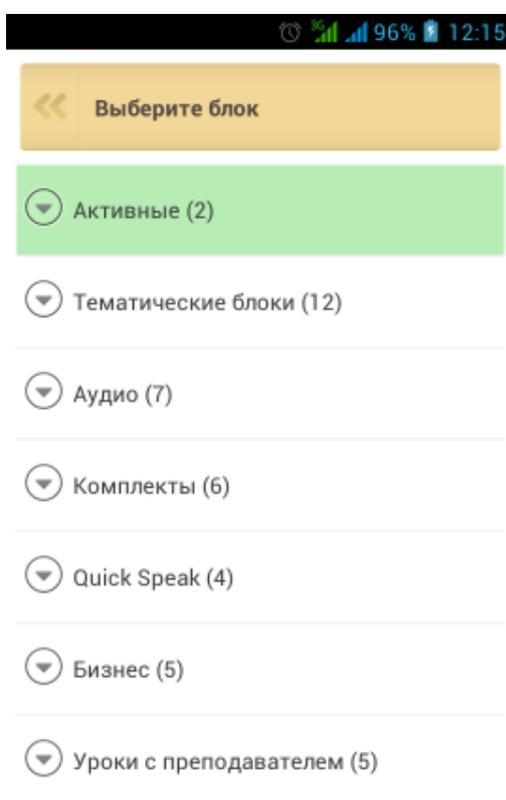


Рисунок 3. Снимок экрана приложения Слова бегом

Следующим примером можно взять онлайн-сервис «Puzzle English», является платной программой. Доступен на Android/iOS.

«Puzzle English — онлайн сервис для самостоятельного изучения английского. Суть программы в тренировках восприятия английского языка на слух» [7].

Как видно из определения приложения, Puzzle English позволяет изучать только английский язык.

Система предлагает пользователю упражнения разделенным по категориям:

«Грамматика. Пользователям предлагается изучать грамматические правила и лексику через просмотры видеороликов.

Видеопазлы. Также через просмотры видеороликов пользователи тренируют понимание английской речи. Есть субтитры, а на каждое слово можно нажать и получить перевод.

Аудирование. Вам говорят фразу или предложение, а вы должны ее собрать из предложенных фрагментов-пазлов» [7].

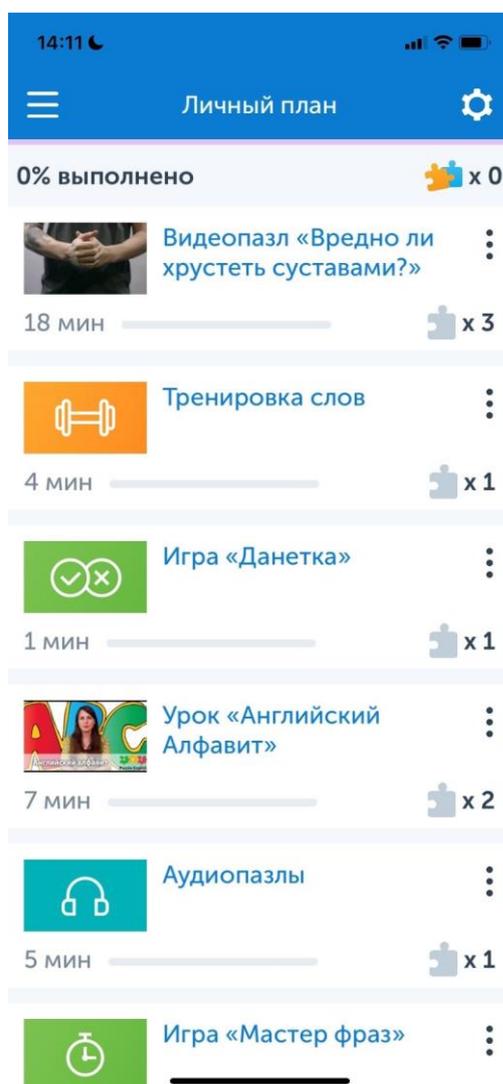


Рисунок 4. Снимок экрана приложения Puzzle English

Еще одним примером может являться обучающий сайт «Lingualeo», есть как бесплатный режим, так и платный. Доступен на Android/iOS.

«LinguaLeo – это обучающий сайт для самостоятельного изучения английского языка. В основу методики положено, по словам разработчиков, «”умное” освоение живого контента от носителей языка»» [8].

Из определения приложения видно, что сайт позволяет изучать только английский язык.

Система предлагает бесплатно несколько режимов, такие как:

«Слово-перевод – дается слово на английском и к нему несколько вариантов ответа – выбирайте правильный.

Перевод-слово — то же самое, но слова даются на русском, нужно выбрать английский перевод.

Саванна — игра, в которой с неба падают английские слова, а вам нужно нажимать на правильное русское слово. Ошибаться можно, но количество жизней ограничено» [8].

Также система может предложить Премиум режим, то есть платный. В него входит больше режимов обучения:

«Брейншторм – комбинированный режим, сначала вам покажут несколько слов, чтобы вы отобрали, какие следует изучать, а какие пора перевести в «изученное». Затем отобранные слова прогонят сначала в режиме «Слово-перевод», затем в «Конструкторе слов».

Повторение – быстрый способ повторить слова. Дается английское слово, нужно за 3 секунды выбрать один из двух вариантов.

Лео-спринт – скоростная тренировка. В течение минуты нужно угадывать, верный перевод предложен к английскому слову или нет» [8].



Рисунок 5. Снимок экрана приложения LinguaLeo

Следующей программой будет «Wlingua», доступный в браузере и на мобильном телефоне (Android/iOS.).

«Wlingua — сервис для быстрого изучения английского на начальном и среднем уровне, доступный в браузере и на мобильном. С его помощью можно быстро и эффективно улучшить знание иностранного языка, занимаясь в своем ритме где и когда удобно» [9].

Также, как и в предыдущем примере, можно заметить, что сервис обучает английскому языку.

Здесь дается бесплатно Премиум на один месяц, затем его нужно продлевать, либо можно заниматься в базовом режиме.

В сервисе есть упражнения, такие как:

«На чтение и понимание смысла прочитанного, когда нужно отвечать на вопросы типа «правда или ложь», есть блоки из упражнений для закрепления материала и задания, связанные с аудио. Каждый урок состоит из шести типов заданий, которые обеспечивают действительно качественную проверку полученных знаний» [9].

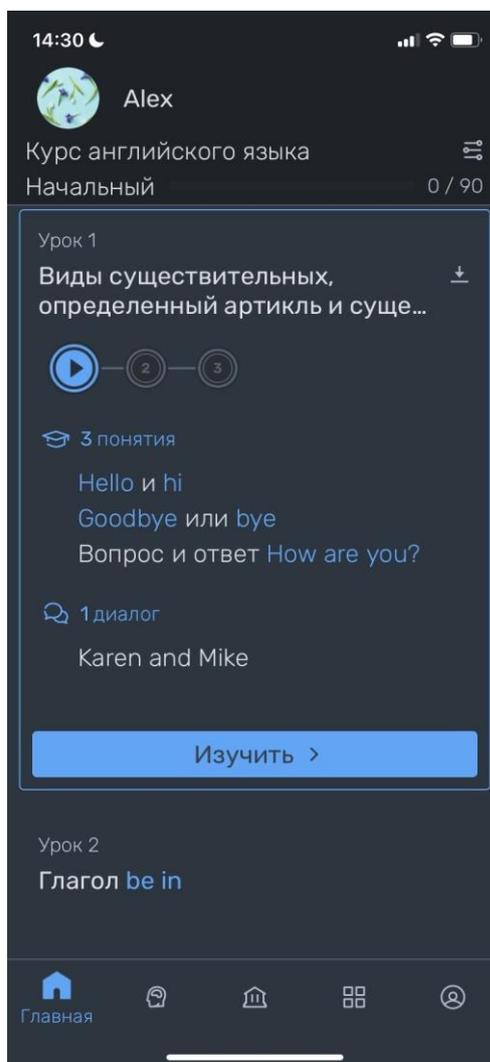


Рисунок 6. Снимок экрана приложения Wlingua

Еще одним примером будет мобильное приложение «Memrise», есть как бесплатная версия, так и платная. Доступен на Android/iOS.

«Memrise – это онлайн-сервис и мобильное приложение для заучивания иностранных слов. С его помощью можно учить не только слова, но, например, формулы, термины, столицы, но в основном Memrise используют для изучения иностранных языков» [10].

В основном, в приложении, предлагается на изучение несколько языков, такие как: английский, немецкий, французский и т.д.

«В каждый курс входит несколько уровней — по сути это наборы слов, объединенные темой. Получается, что слова объединены в уровни (наборы слов, штук по 20), уровни объединены в курсы. Например, курс «Английский 1» состоит из 10 уровней, включающих примерно по 20 слов. Итого: около 200 слов» [10].

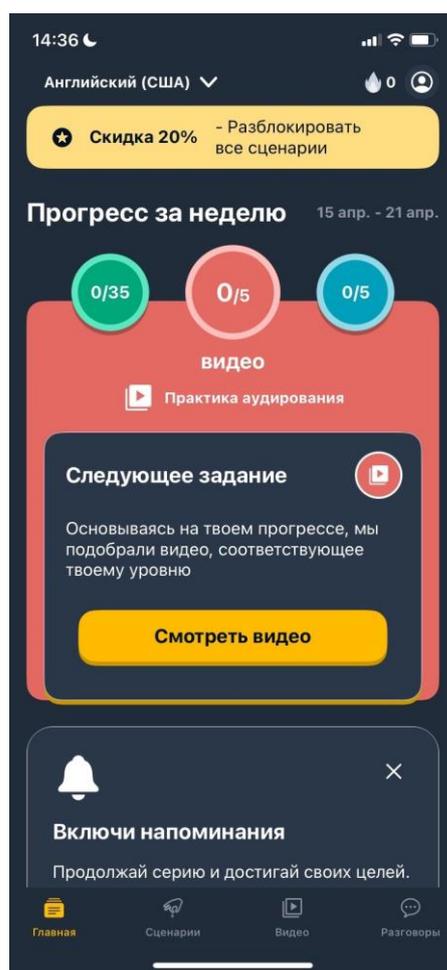


Рисунок 7. Снимок экрана приложения Memrise

Также примером может служить сервис для проверки письма «Grammarly», доступный только на Android.

«Grammarly — просто потрясающий сервис для проверки письма. Он умеет проникать в контекст ваших предложений и вылавливать даже самые тонкие ошибки» [11].

Приложение, созданное для проверки грамматики для иностранных языков. Здесь нет как такого обучения, просто проверка вашей грамматики.

В системе есть несколько режимов для этой проверки:

«Активность — сколько слов было проверено.

Сколько ошибок было исправлено — тоже не очень полезный стат, показывает, насколько хорошо Grammarly поработал.

И последнее в статистике — это топ 3 грамматических ошибки» [11].

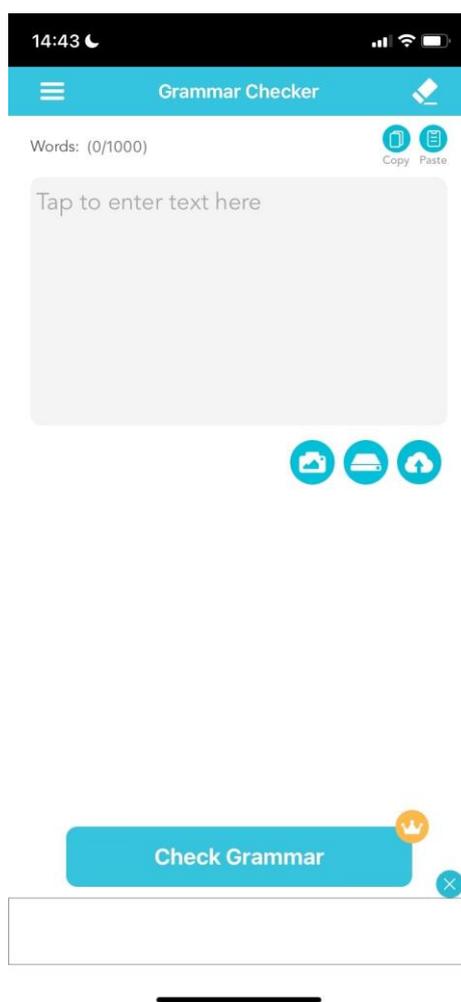


Рисунок 8. Снимок экрана приложения Grammarly

«Grammarize», еще один пример для изучения иностранного языка. Бесплатное приложение доступное только на iOS.

«Бесплатное приложение Grammarize. Приложение представляет собой одновременно справочник и тренажер, благодаря которому вы сможете довести

свое знание грамматики английского языка если не до совершенства, то, по крайней мере, до очень высокого уровня» [12].

Как видно из источника, приложение позволяет изучать английский язык.

Система имеет несколько основных разделов: «Задания», «Правила», «Глаголы», «Результаты» [12].

А также задания:

«Каждое упражнение представляет собой 15 тестов с вопросами. При включенном таймере нам предлагается либо подставить недостающий артикль (предлог, местоимение...), либо составить предложение из предложенных слов с выполнением синтаксических норм. Ответ никак не влияет на скорость прохождения теста, результаты показываются по его окончанию на странице статистики» [12].



Рисунок 9. Снимок экрана приложения Grammarize

Еще одним примером будет служить «Cake», мобильное приложение на платформы Android / iOS.

«Суть методики обучения — видеоуроки, имитирующие общение с носителем языка. Здесь нет длинных и скучных объяснений грамматики, зато есть много живых диалогов: нарезок из кинофильмов, мультфильмов, реальных разговоров на английском, и так далее» [13].

Это приложение позволяет развить свой разговорный уровень английского языка.

Система позволяет просматривать ролики и учиться по ним:

«Видеоуроки разделены по темам и уровням сложности. Ролики длятся от 15 секунд до 2 минут и сопровождаются английскими субтитрами с русским переводом. Каждое видео содержит одну ключевую фразу, выделяющуюся трехкратным повторением» [13].

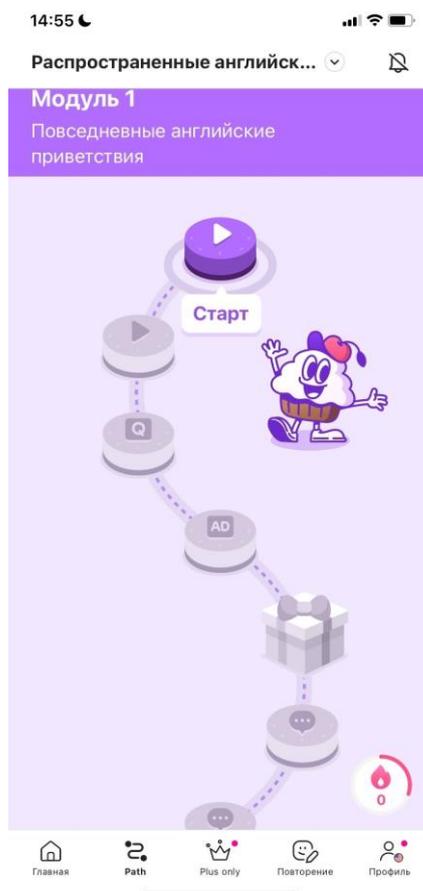


Рисунок 10. Снимок экрана приложения Cake

Еще одним примером можно рассмотреть приложение «Tandem», мобильное приложение для изучения иностранных языков совместно. Доступно для Android / iOS.

«Приложение Tandem — это простой способ найти собеседника для практики иностранного языка. Подразумевается, что, найдя партнера, вы будете изучать язык «в тандеме», то есть, скажем, вы помогаете собеседнику учить русский, а собеседник поможет вам с английским» [14].

Приложение поможет изучить вам все виды иностранных языков, с помощью партнера, которого можно найти в нем. Большинство из них являются носителями английского языка.

Система имеет несколько видов общения:

«Поиск собеседника по анкетам. Можно ограничить поиск отдельными странами, поиском только среди носителей изучаемого языка.

Текстовое или голосовое общение в чате. Звонки можно совершать только если вы уже обменялись сообщениями, хотя бы по одному.

Голосовые сообщения. В «Тандеме» есть голосовые сообщения, которые можно надиктовывать и отправлять в чат. В отличие от аналогичной функции в «ВКонтакте», Telegram и др., здесь есть особенность — сообщение, записав, можно прослушать и продиктовать еще раз, исправив ошибки, а уже затем отправить» [14].

Ну и последним примером может служить приложение «Mondly». Оно является мобильным и виртуальным приложением. В мобильных версиях доступно для Android / iOS. Является платным приложением.

«Mondly — это платформа для изучения языков, доступная на iOS, Android и в Интернете, которая позволяет вам развивать свои навыки и достигать беглости на выбранном языке с вашего родного языка в кратчайшие сроки» [15].

Приложение позволяет общаться с носителями языка, и учиться у них разговорной речи. Включает в себя разные иностранные языки.



Рисунок 11. Снимок экрана приложения Tandem

Ну и последним примером может служить приложение «Mondly». Оно является мобильным и виртуальным приложением. В мобильных версиях доступно для Android / iOS. Является платным приложением.

«Mondly — это платформа для изучения языков, доступная на iOS, Android и в Интернете, которая позволяет вам развивать свои навыки и достигать беглости на выбранном языке с вашего родного языка в кратчайшие сроки» [15].

Приложение позволяет общаться с носителями языка, и учиться у них разговорной речи. Включает в себя разные иностранные языки.

Система позволяет работать в нескольких режимах:

«Сосредоточьтесь на фразах, а не на отдельных словах

Сосредоточившись на распространенных фразах, вы начнете запоминать самые распространенные слова, строить предложения и принимать участие в беседах за минуты, а не часы.

Слушайте носителей языка

Mondly работает с профессиональными носителями языка, и вам понравится безупречное произношение и естественный акцент.

Практикуйте настоящие разговоры

Использование чат-бота с распознаванием речи мирового класса и эксклюзивными курсами по дополненной реальности.

Сохранение знаний с помощью уникальной системы повторения

Mondly использует проверенные и эффективные интервалы для повторения, которые помогут вам быстро выучить новые языки» [15].

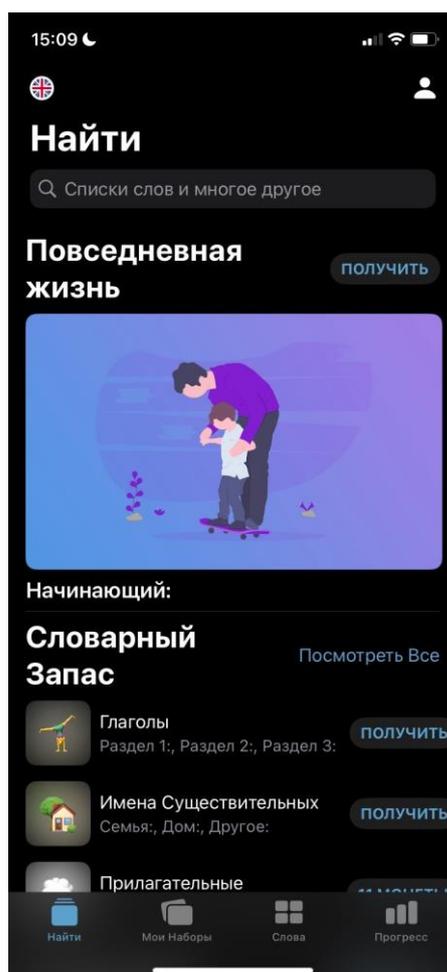


Рисунок 12. Снимок экрана приложения Mondly

Таким образом, рассмотрев множество примеров, можно сказать, что многие из них позволяют изучать иностранные языки, но в основном это английский, французский, немецкий. И малые из них дают изучить испанский. Наша же программа позволит изучать базовые слова на испанском языке. В нашей программе будет режим упражнений: По картинке/звуку написать слово. Ну и также в нашем приложении будет два режима прохождения вопросов — это тренировочный тест и основной(экзаменационный) тест. Это более удобный вариант для преподавателя испанского языка, так как во втором варианте теста он будет получать результаты своих студентов себе на почту, а не проверять у каждого телефон.

Также в нашем случае приложении будет бесплатное, так как оно обучающее, а теорию будет давать сам преподаватель. То есть приложение будет создано для проверки/закрепления знаний студента.

## **1.2 Программное обеспечение для проектирования и разработки**

Для разработки самого приложения сначала нужно разобрать его логику, то есть спроектировать его. В этом нам поможет графический онлайн-редактор «Figma».

«Figma — это графический онлайн-редактор для совместной работы. В нём можно создать прототип сайта, интерфейс приложения и обсудить правки с коллегами в реальном времени

В Фигме можно отрисовать элементы интерфейса, создать интерактивный прототип сайта и приложения, иллюстрации, векторную графику.

Если вы хотите более гибко работать с интерфейсной графикой, можно отрисовать макет в Фигме» [16].

Достоинства сервиса Figma: возможность совместной работы над проектом, хранит документы в облаке, возможность работы с любого устройства.

Бесплатный продукт, именно поэтому пользователи могут ознакомиться с функциями и возможностями приложения для дизайна.

Также рассмотрим, что нам нужно для реализации это мобильного обучающего приложения.

Для этого в процессе разработки обучающего мобильного приложения будет использоваться официальная интегрированная среда разработки Android Studio.

«Android Studio — официальная IDE (интегрированная среда разработки) для работы с платформой Android. Она позволяет разрабатывать приложения для смартфонов, планшетов, часов, телевизоров и других устройств на этой ОС.

Android Studio это официальная среда разработки, которую поддерживает Google. Она основана на IntelliJ IDEA от компании JetBrains. В Android Studio те же горячие клавиши и интерфейс, что и в других продуктах JetBrains, поэтому эта среда самая привычная и удобная.

Работа в Android Studio похожа на работу в большинстве современных IDE. Однако, как и везде, здесь есть свои нюансы» [17].

Язык программирования и язык разметки — Kotlin и XML, соответственно.

«Kotlin — статически типизированный, объектно-ориентированный язык программирования, работающий поверх Java Virtual Machine, разработанный компанией JetBrains

Авторы ставили целью создать язык более лаконичный и типобезопасный, чем Java, и более простой, чем Scala. Следствием упрощения по сравнению со Scala стали также более быстрая компиляция и лучшая поддержка языка в IDE. Язык полностью совместим с Java, что позволяет Java-разработчикам постепенно перейти к его использованию; в частности, язык также встраивается Android, что позволяет для существующего Android-приложения внедрять новые функции на Kotlin без переписывания приложения целиком» [18].

На данный момент является стандартом для разработки мобильных приложений.

Для разработки данного приложения был выбран язык Kotlin, потому что он является стандартом для написания новых мобильных приложений, рекомендованный Google с 2018 года.

«XML файлы являются ресурсами разметки и хранят определение визуального интерфейса

Как правило, для определения визуального интерфейса в проектах под Android используются специальные файлы xml. Эти файлы являются ресурсами разметки и хранят определение визуального интерфейса в виде кода XML. Подобный подход напоминает создание веб-сайтов, когда интерфейс определяется в файлах html, а логика приложения - в коде javascript.

Объявление пользовательского интерфейса в файлах XML позволяет отделить интерфейс приложения от кода. Что означает, что мы можем изменять определение интерфейса без изменения кода java. Например, в приложении могут быть определены разметки в файлах XML для различных ориентаций монитора, различных размеров устройств, различных языков и т.д. Кроме того, объявление разметки в XML позволяет легче визуализировать структуру интерфейса и облегчает отладку» [19].

XML-язык позволит нам разметить и создать удобный интерфейс для пользователя.

Для создания базы данных и авторизации будет использоваться сервис Firebase.

«Firebase — это облачная база данных, которая позволяет пользователям хранить и получать сохраненную информацию, а также имеет удобные средства и методы взаимодействия с ней.

Firebase хранит текстовые данные в JSON формате и предоставляет удобные методы для чтения, обновления и извлечения данных. Также, Firebase может помочь с регистрацией и авторизацией пользователей, хранением сессий

(авторизованные пользователи), медиафайлов к которым с легкостью предоставляет доступ благодаря Cloud Storage.

Естественно, Firebase не может быть полностью бесплатной. Часть самого крутого функционала остается за кадром для тех, кто не желает платить. Но самые основные и горячо востребованные функции регистрации, авторизации и хранения текста доступны всем после регистрации в системе.

Вот смотрите: языки программирования Python и JavaScript сами по себе гибкие (и очень популярные). Благодаря этому, мы (разработчики) получаем мощный инструмент для создания в кратчайшие сроки сайта, веб-приложения, мобильного или даже десктопного приложения.

Firebase позволяет сохранять эту скорость. Не нужно отвлекаться на какие-то другие вещи (создание базы данных, написание API приема и получения данных). Вся серверная часть ложится на плечи этого сервиса» [20].

Firebase позволит нам использовать базы данных для регистрации/авторизации, а также для хранения картинок и их испанских названий.

### **1.3 Формализованное описание технического задания**

Составлено на основе ГОСТ 34.602-89 «Техническое задание на создание автоматизированной системы» [21].

#### **1. Общие сведения.**

##### **1.1. Название организации-заказчика.**

Заказчик: «Уральский государственный педагогический университет».

Институт «Иностранных языков».

Кафедра «Английской филологии и методики преподавания английского языка».

##### **1.2. Название продукта разработки (проектирования).**

«Проект мобильного приложения для изучения испанского языка»

Название приложения «Флора и Фауна на испанском».

##### **1.3. Назначение продукта.**

Продукт предназначен для студентов 1 курса института иностранных языков, а также преподавателей. Для изучения испанской флоры и фауны по картинкам и звукам.

#### 1.4. Плановые сроки начала и окончания работ.

В соответствии с планом выполнения ВКР (01.09.2023 – 20.05.2024).

### 2. Характеристика области применения продукта.

#### 2.1. Процессы и структуры, в которых предполагается использование продукта разработки.

Использование в учебном процессе педагогического университета при изучении лексических единиц испанского языка

#### 2.2. Характеристика персонала (количество, квалификация, степень готовности)

Разработчик должен уметь работать в среде разработке Android Studio, знать базовый язык программирования Kotlin, уметь донести информацию до пользователя-преподавателя.

Пользователь-преподаватель должен уметь обращаться с телефоном с системой Android. Уметь доносить информацию до студента. Знать требования к приложению, и уметь донести эти требования до разработчика.

Пользователь-студент должен уметь обращаться с телефоном с системой Android.

### 3. Требования к продукту разработки.

#### 3.1. Требования к продукту в целом.

Приложение должно работать под операционную систему Android. Должна быть предусмотрена регистрация и авторизация пользователей, данные хранятся на платформе Firebase. Система должна предусматривать выбор темы: тема «Фауна» и тема «Флора». Система должна предусматривать выбор двух режимов работы: режим тренажера и режим тестирования. Также система должна предусматривать выбор типа прохождения работы: тип «Картинки» и

тип «По звукам». После завершения режима тестирования преподавателю может быть отправлен результат прохождения тестирования.

### 3.2. Аппаратные требования.

Аппаратные требования для функционирования:

- ОС Android
- Свободное место на ОС должно быть примерно 20 МБ.

### 3.3. Указание системного программного обеспечения (операционные системы, браузеры, программные платформы и т.п.).

Минимальная версия Android должна быть 8.0.

### 3.4. Указание программного обеспечения, используемого для реализации.

Используемое программное обеспечение Android Studio, Figma, Ramus.

### 3.5. Особенности реализации серверной и клиентской частей.

Не предусмотрено

### 3.6. Форматы входных и выходных данных

Входные данные: данные о регистрации, ввод текстовой информации при выполнении упражнений, управляющая информация (выбор темы, режима тестирования, типа тестирования).

Выходные данные: сгенерированный тест или сгенерированный тренажер, результат тестирования.

### 3.7. Источники данных и порядок их ввода в систему (программу), порядок вывода, хранения.

Источником данных является база данных, которая встроена в приложение, и не может быть изменена пользователем, только разработчиком. Все данные инсталлируются вместе с приложением.

### 3.8. Порядок взаимодействия с другими системами, возможности обмена информацией.

Не предусмотрено.

### 3.9. Меры защиты информации.

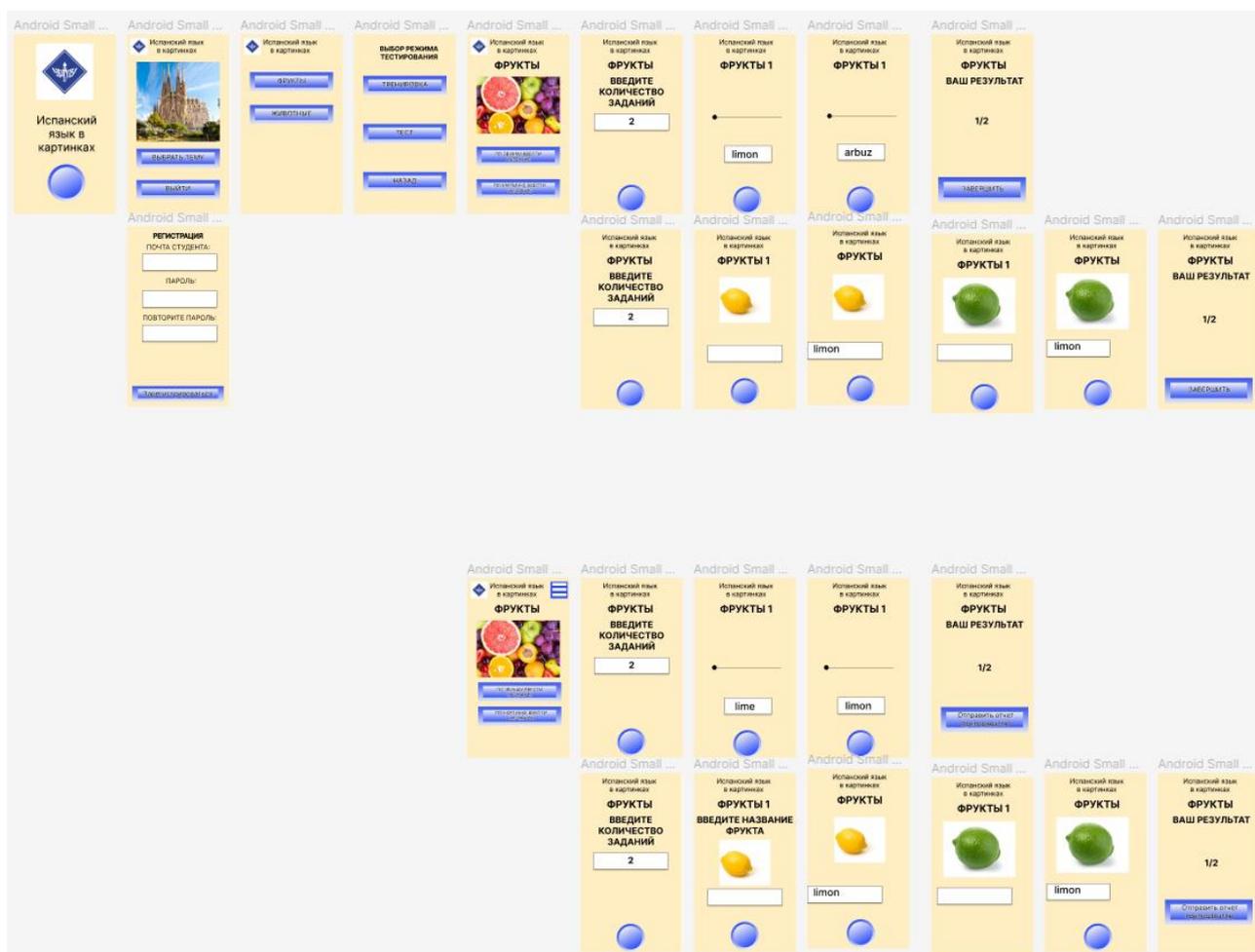
Не предусмотрено.

#### 4. Требования к пользовательскому интерфейсу.

##### 4.1. Общая характеристика пользовательского интерфейса.

WIMP интерфейс

##### 4.2. Размещение информации на экране, дизайн экрана.



#### 5. Требования к документированию.

##### 5.1. Перечень сопроводительной документации.

Не предусмотрено.

#### 6. Порядок сдачи-приемки продукта.

В соответствии с планом выполнения ВКР.

## Глава 2. Реализация мобильного приложения для изучения испанского языка

### 2.1 Функциональная модель мобильного приложения

Необходимо построить функциональную модель нашего мобильного приложения. Для этого можно использовать несколько сервисов, такие как: BPWin, Ramus. Мы будем использовать Ramus.

«Ramus — программный продукт в области управления знаниями предприятия. Позволяет проводить описание, анализ и моделирование бизнес-процессов, а также строить систему классификации и кодирования» [22].

Построим функциональную модель (рисунок 23).

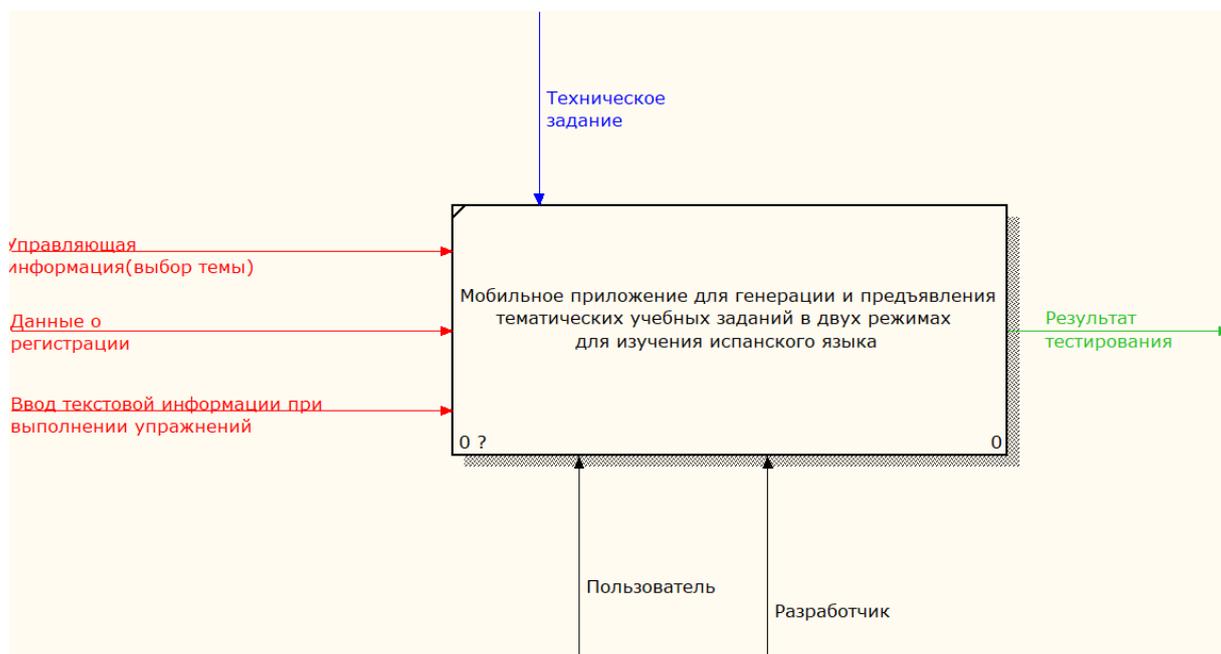


Рисунок 13 Функциональная модель мобильного приложения

Таким образом, мы можем увидеть элементы графической нотации, то есть элементы функциональной модели, а именно: функция, три входа, управление, два механизма и выход.

На вход нам поступает:

- Управляющая информация (к примеру, выбор темы, сюда также выходит и выбор режима тестирования и т.д.);
- Данные о регистрации;

- Ввод текстовой информации при выполнении упражнений.

Функция:

- Мобильное приложение для генерации и предъявления тематических учебных заданий в двух режимах для изучения испанского языка.

Управление:

- В качестве управления служит техническое задание, представленное преподавателем.

Механизмы:

- Разработчик;
- Пользователь (тестирующий).

Выход:

- Результат тестирования.

Таким образом, на основе функциональной модели была определена структура разрабатываемого приложения.

В сервисе figma, мы составили логику приложения (Рисунок 14).

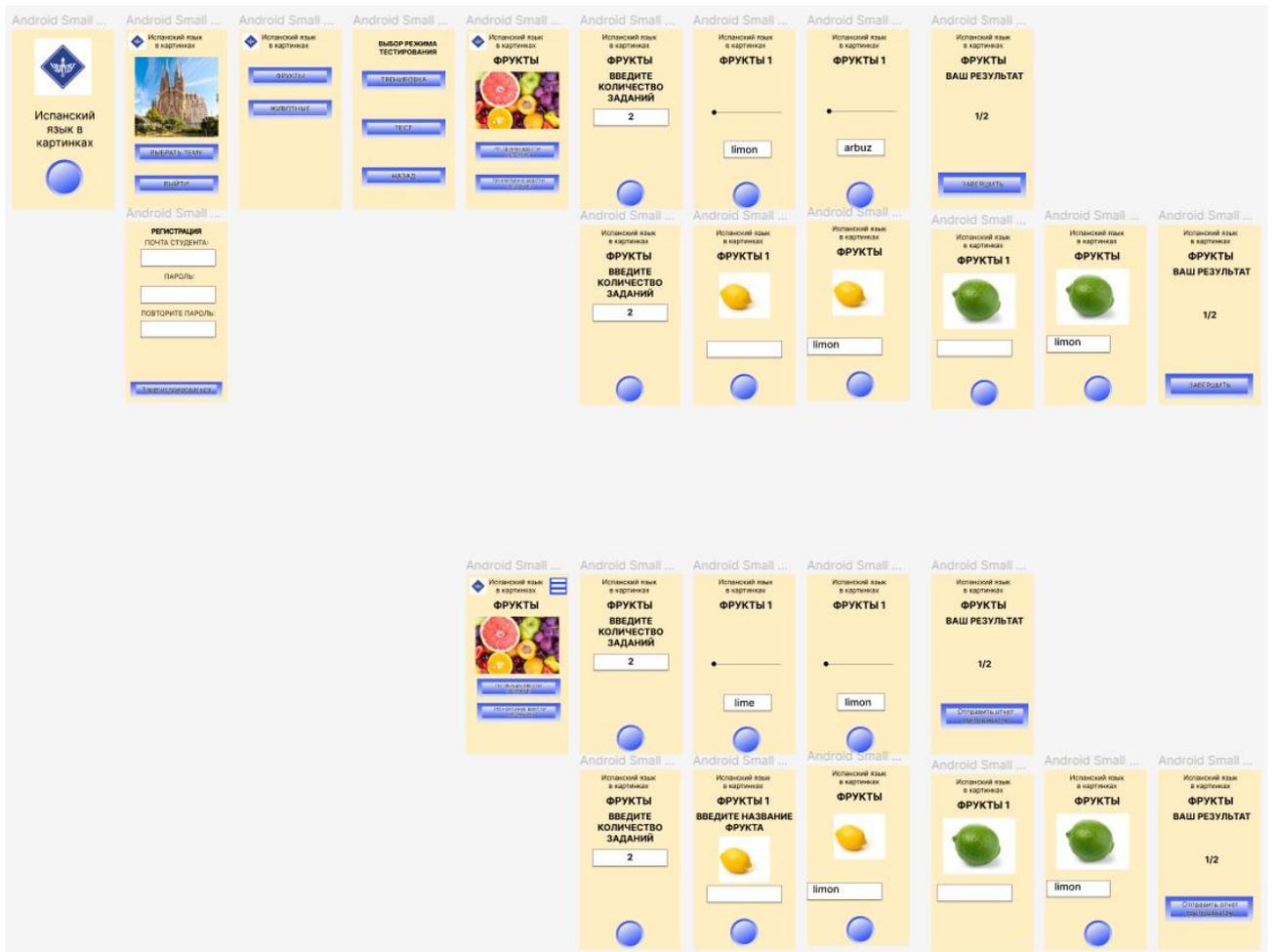
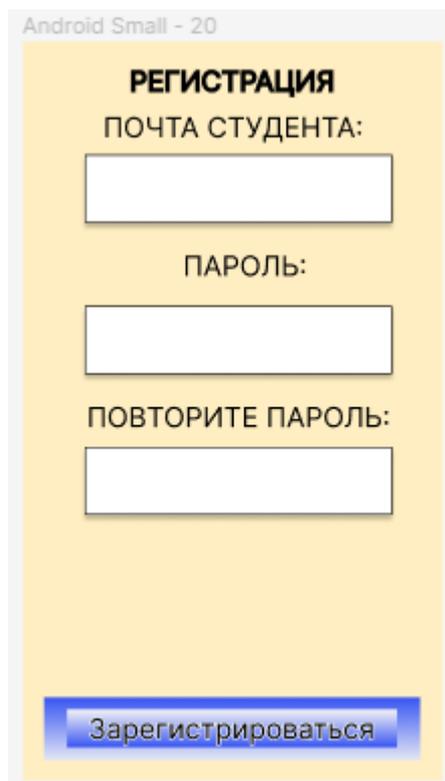


Рисунок 14. Логика приложения в Figma

Давайте разберем подробнее, что у нас включает в себя разработанная логика приложения в сервисе «Figma». Все показанные картинki — это наброски, и результат может отличаться от первоисточника.

Первым, что нужно будет сделать пользователю, когда он откроет приложение это зарегистрироваться (Рисунок 15).



*Рисунок 15. Регистрация пользователя в Figma*

Регистрация будет включать в себя: почту студента, ввод пароля, повторения пароля, для проверки правильности ввода, и что студент запомнил свой пароль.

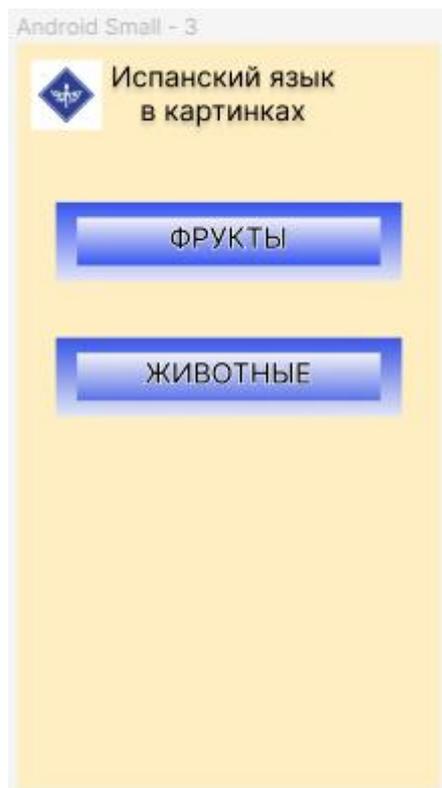
Затем, когда регистрация будет пройдена, можно авторизоваться и спокойно проходить задания. Важно! Без регистрации студент не сможет проходить задания, все-таки это учебное приложение для университета.

Дальше студента будет встречать главная страница на которой ему можно будет нажать «Выбрать тему» и перейти на список тем, и также будет кнопка «Выйти», которая подразумевает собой выход из учетной записи (Рисунок 16).



*Рисунок 16. Главная страница в Figma*

После этого студента встречает страница с выбором тем, где он может выбрать тему, по которой хочет пройти задания и попрактиковаться (Рисунок 17).



*Рисунок 17. Выбор темы в Figma*

Затем же, пользователь увидит выбор режимов тестирования, он будет включать в себя режим «Тренировка» и режим «Тест», то есть можно сказать экзамен (Рисунок 18).



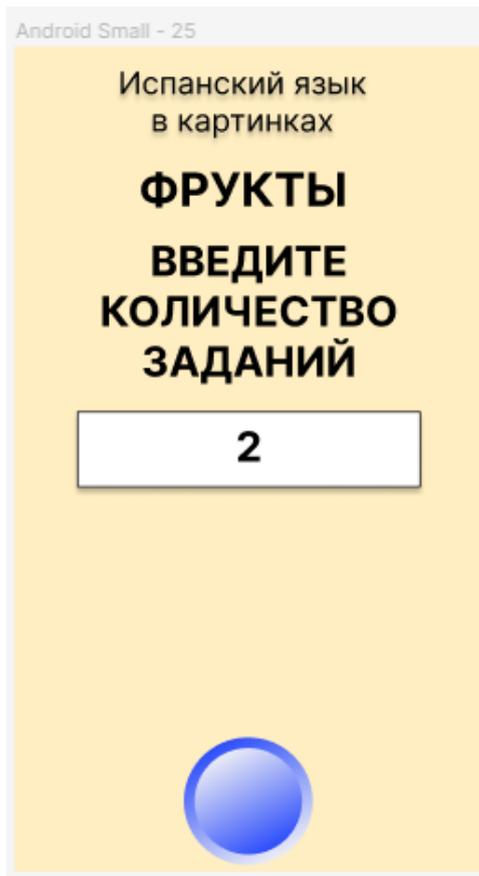
*Рисунок 18. Выбор режима тестирования в Figma*

Следующая страница встречает студента выбором режима прохождения заданий. Здесь будет находится кнопка «По картинке ввести название» и «По звукам ввести название» (Рисунок 19).



*Рисунок 19. Выбор режима прохождения в Figma*

После выбора прохождения режима заданий будет страница на выбор количества заданий, здесь уже преподаватель будет говорить сколько заданий требуется решить, а студент вводит это число, либо же студент может сам выбрать количество заданий, если он захочет самостоятельно попрактиковаться (Рисунок 20).



*Рисунок 20. Выбор количества заданий в Figma*

Давайте рассмотрим, какие режимы заданий будут в нашем приложении.

Если выбран режим прохождения «Ввести по картинке название», то на странице будет высвечиваться картинка с объектом, а также будет содержаться поле ввода, куда студент может ввести название объекта на испанском языке (Рисунок 21).

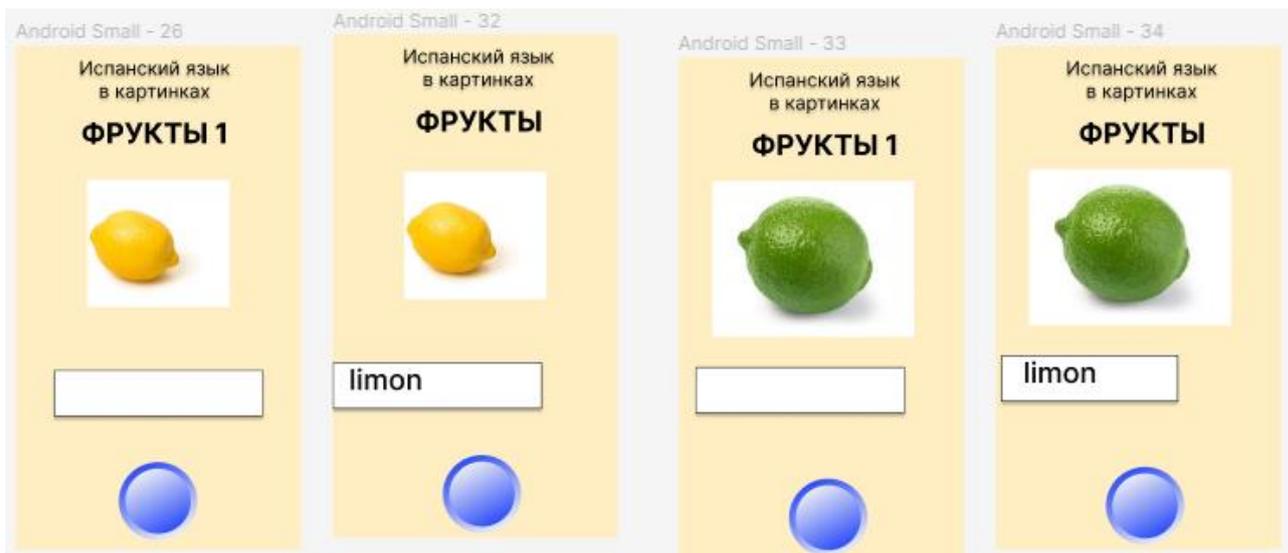


Рисунок 21. Режим «Ввести по картинке название» в Figma

Если выбран режим прохождения «Ввести по звукам название», то на странице будет высвечиваться аудио-дорожка, а также будет содержаться поле ввода, куда студент может ввести название объекта на испанском языке (Рисунок 22).

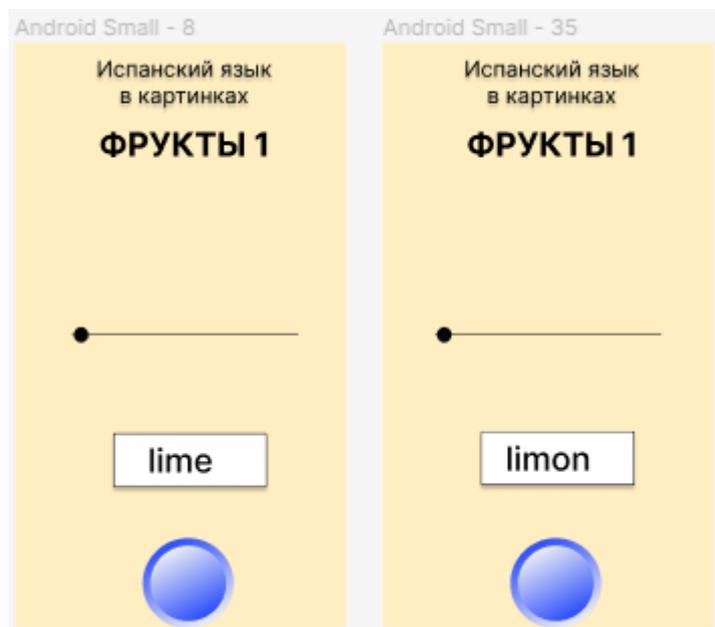
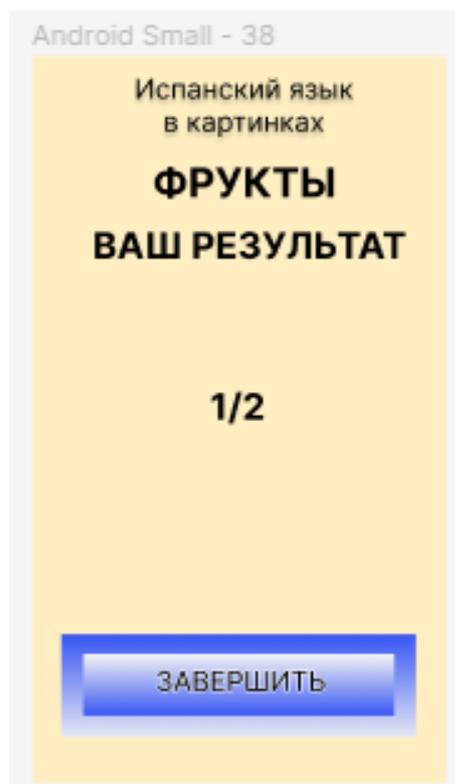


Рисунок 22. Режим «Ввести по звукам название» в Figma

Ну и в конце студенту будет показано сколько он решил верно и общее количество заданий (Рисунок 23). И по нажатию кнопки «Завершить», его перекинет на страницу главного меню (Рисунок 16), если это тест, то также

откроется панель, где студент может выбрать куда отправить результат прохождения.

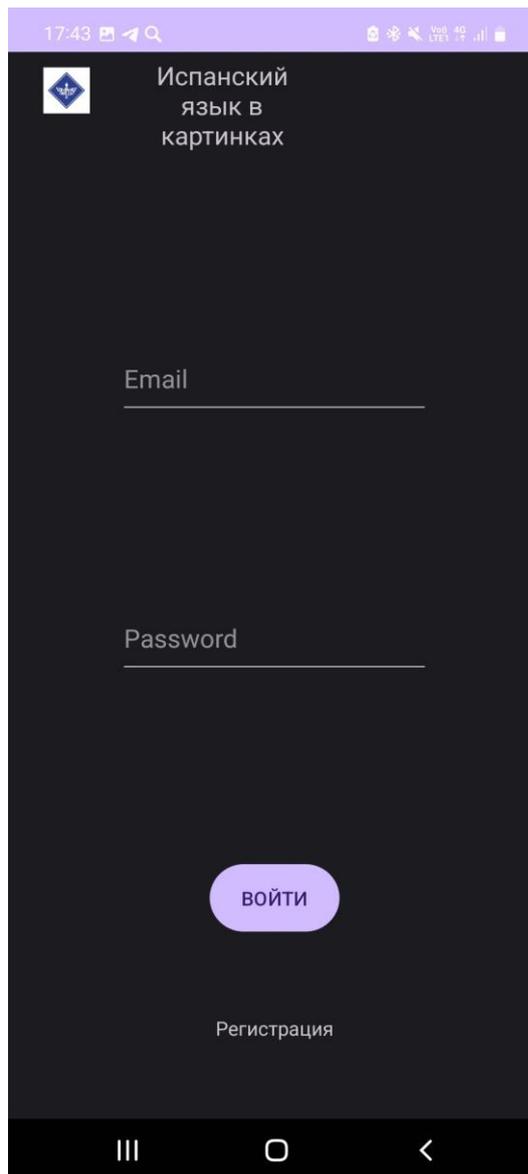


*Рисунок 23. Отображение результата в Figma*

## **2.2 Описание продукта**

Опишем наше готовое приложение. Рассмотрим, что получилось в итоге, с тем, что мы планировали в «Figma», и с тем, что получилось.

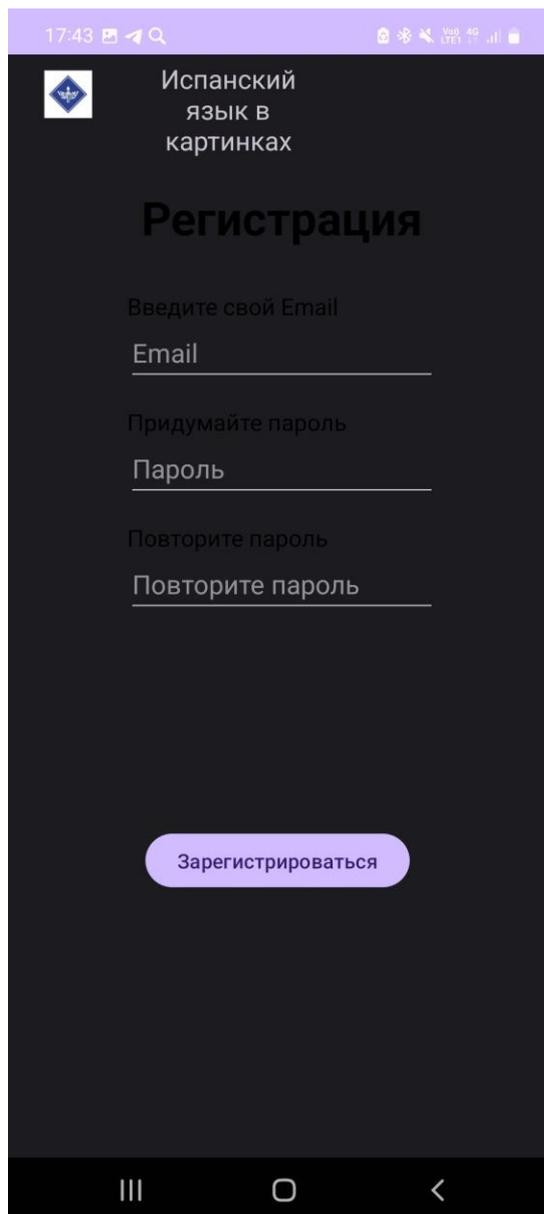
Начнем со страницы входа в приложение. В нее мы попадаем при открытии приложения (если не зарегистрированы или не вошли). Здесь нам предлагается ввести email и пароль (Рисунок 24).



*Рисунок 24. Снимок экрана авторизации*

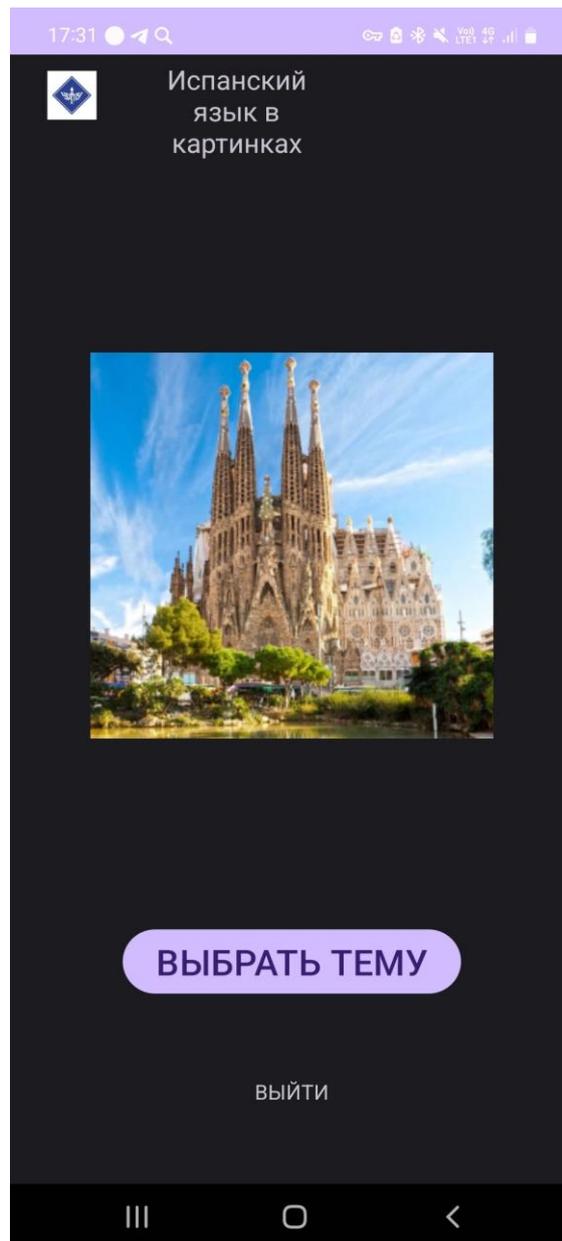
Если же пользователь не зарегистрирован, то это можно сделать, нажав на кнопку снизу «Регистрация», так как дальше нас не должно пускать.

На странице регистрации нужно ввести email, пароль, повторить пароль (Рисунок 25).



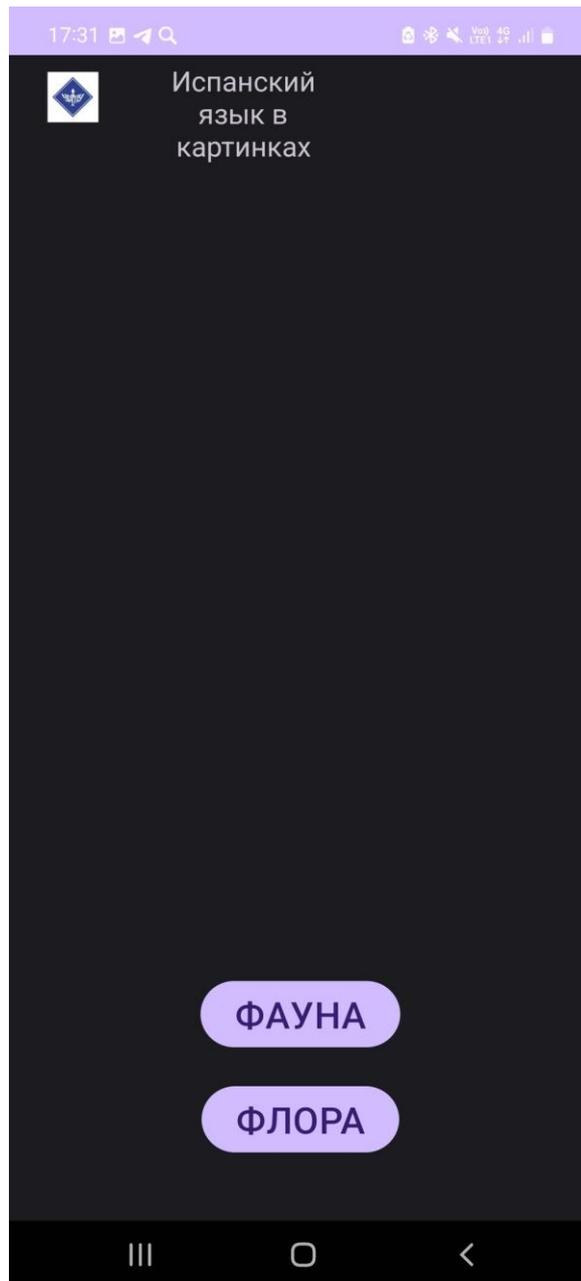
*Рисунок 25. Снимок экрана регистрации*

Затем нас встречает главная страница (Рисунок 26). Здесь мы можем нажать кнопку «Выбрать тему» для перехода на следующую страницу (Рисунок 27), либо же кнопка «Выйти» для выхода из приложения на страницу авторизации (Рисунок 24).



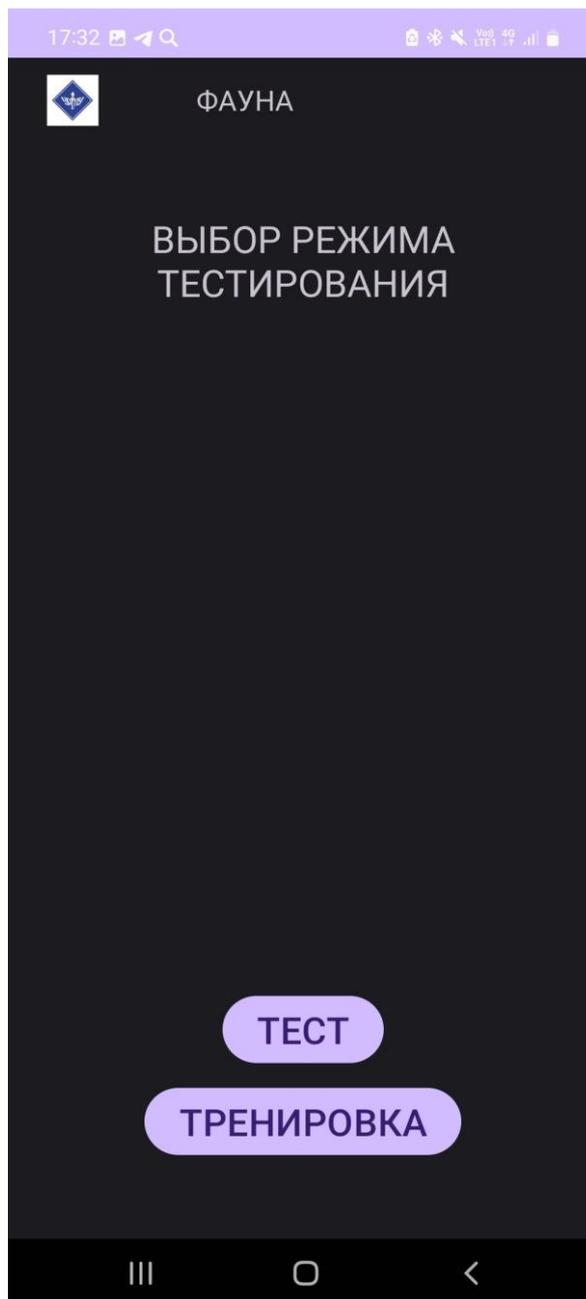
*Рисунок 26. Снимок экрана главная страница*

Как говорилось в техническом задании, у нас есть две темы — Флора и Фауна, их мы и можем выбрать.



*Рисунок 27. Снимок экрана выбор темы*

Следующая страница позволяет выбрать нам режимы тестирования, как уже говорилось в техническом задании, у нас есть два режима — Тренировка и Тест (Рисунок 28). Сверху пишется наша выбранная тема.



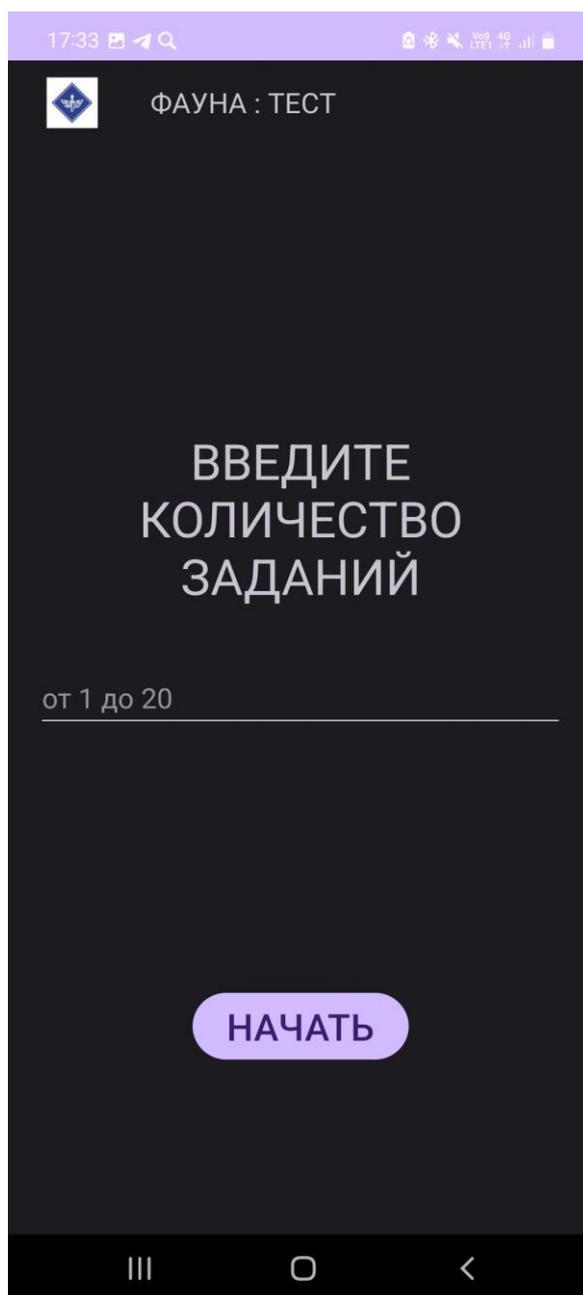
*Рисунок 28. Снимок экрана выбор режима тестирования*

Затем будет выбор режима прохождения тестирования, здесь мы можем видеть выбор двух режимов — «По картинке ввести название» и «По звукам ввести название» (Рисунок 29). Сверху также показывается тема и режим, в котором мы проходим задания.



*Рисунок 29. Снимок экрана выбор режима прохождения*

На следующей странице будет предложен выбор количества заданий, в нашей программе максимум реализовано по 20 объектов данных для каждой темы, следовательно максимальное количество заданий можно выбрать 20 (Рисунок 30). В дальнейшем это можно будет развивать.

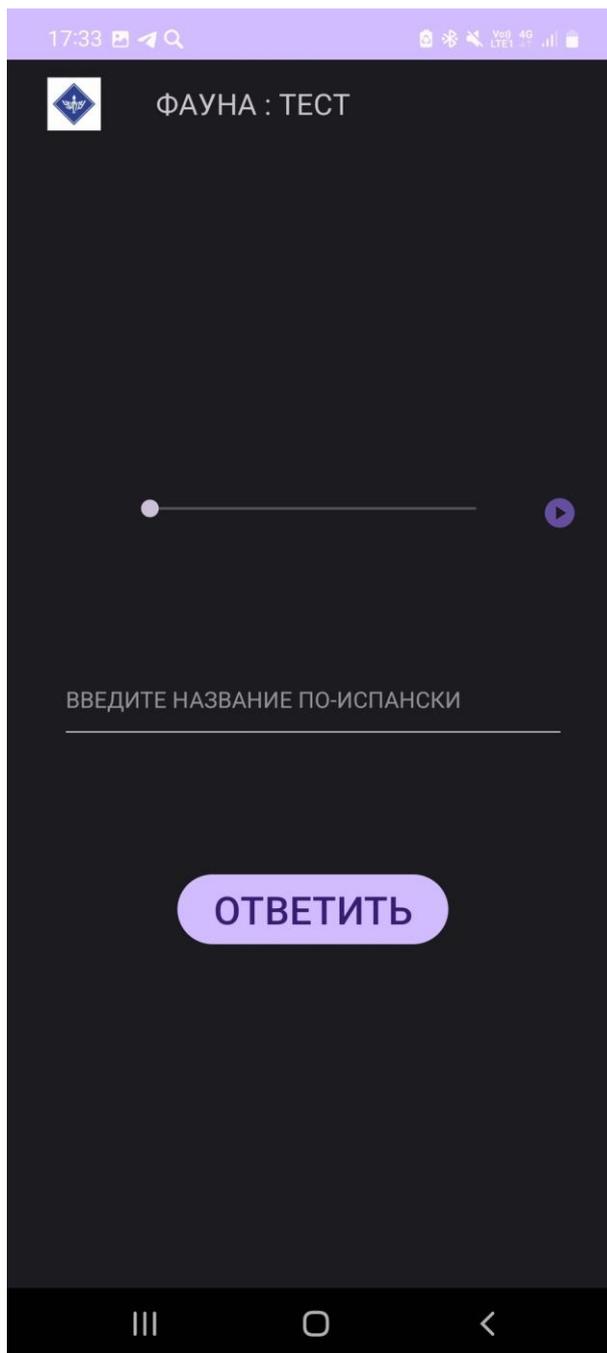


*Рисунок 30. Снимок экрана выбор количества заданий*

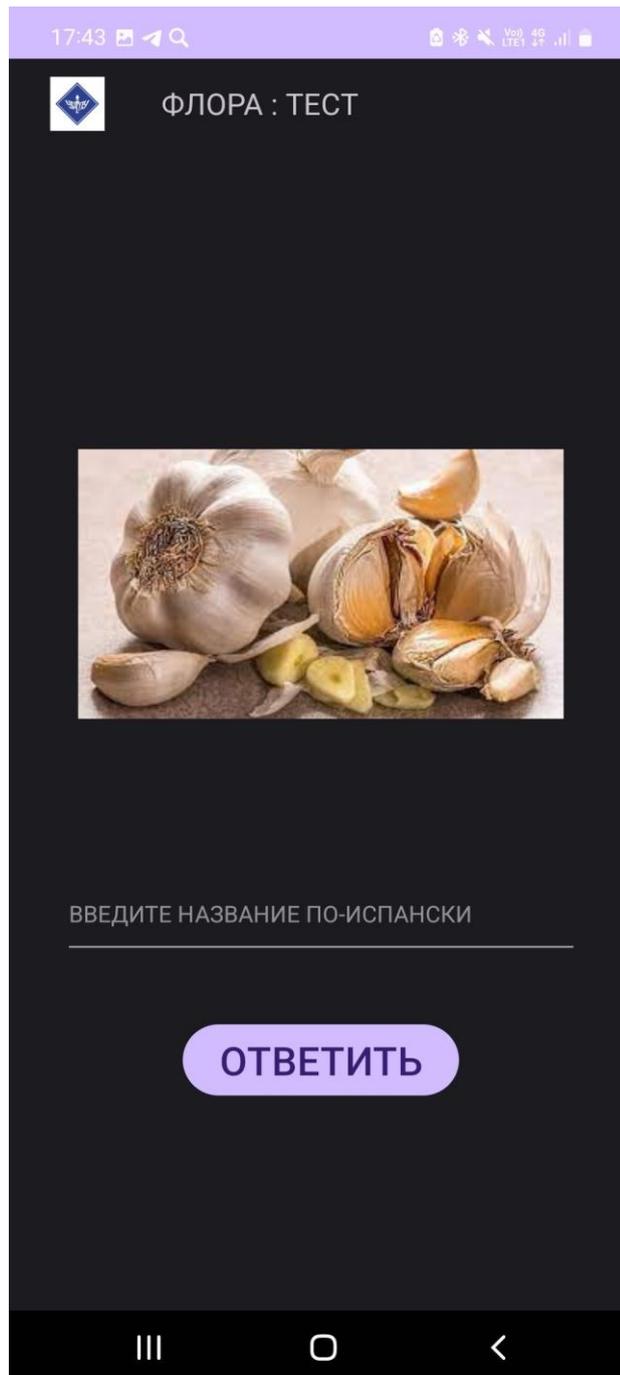
Затем нам выдаются задания, сколько мы выбрали в предыдущем пункте, показывается картинка (выдается аудио) и нужно ввести на испанском, что это картинка (аудио) означает.



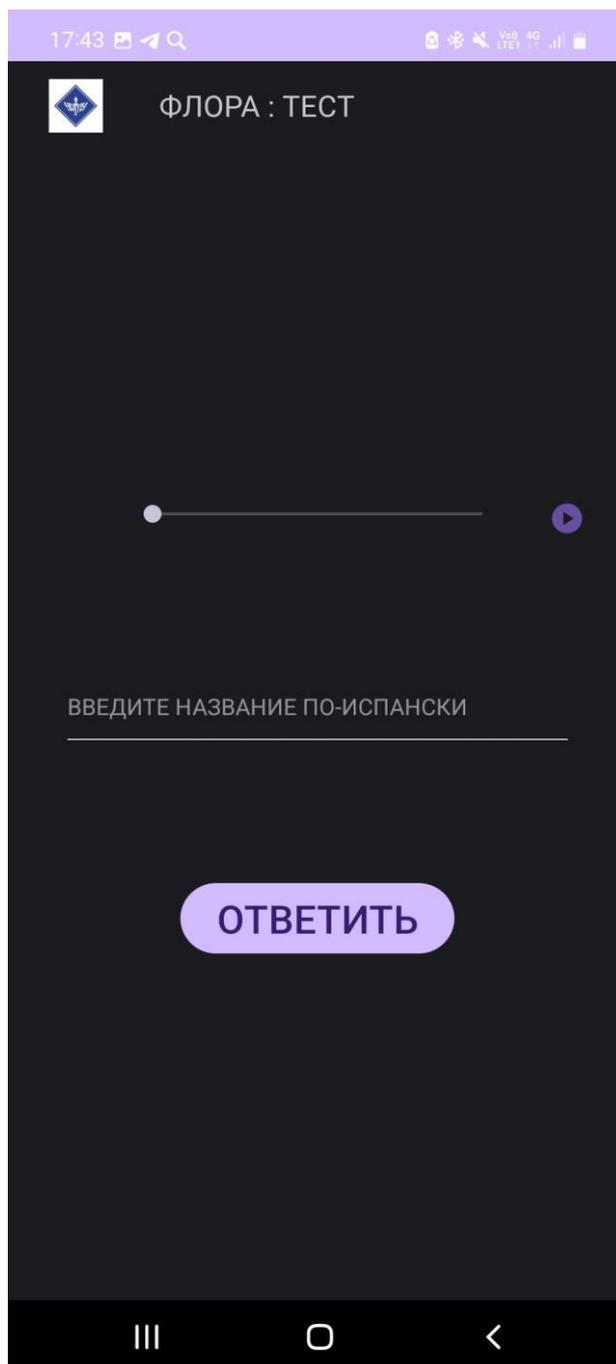
*Рисунок 31. Снимок экрана прохождения теста по теме «Фауна» в картинках*



*Рисунок 32. Снимок экрана прохождения теста по теме «Фауна» в аудио*

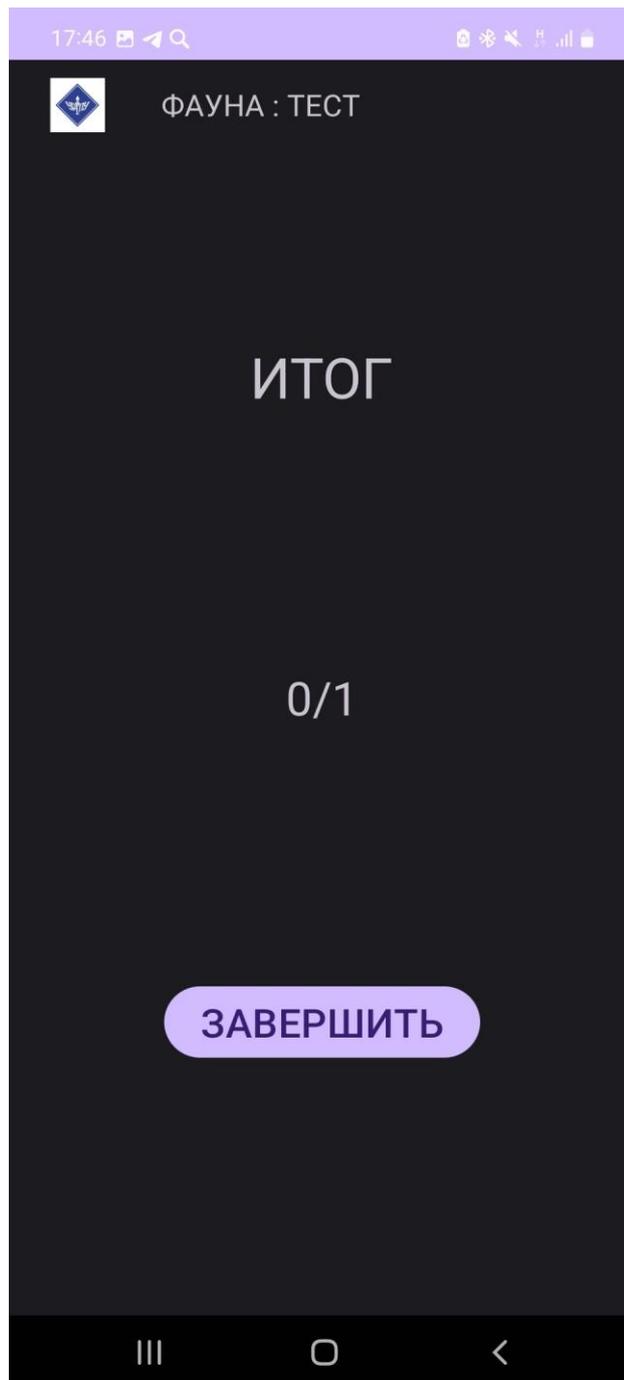


*Рисунок 33. Снимок экрана прохождения теста по теме «Флора» в картинках*



*Рисунок 34. Снимок экрана прохождения теста по теме «Флора» в аудио*

Когда мы проходим тренировку/тест, нам выдается количество выполненных правильно заданий и количество заданий, которое мы задали (Рисунок 35).



*Рисунок 35. Снимок экрана вывод результатов*

Ну и при завершении прохождения режима «Тест» студента перекидывает на главную страницу и высвечивается вкладка, где студент может выбрать куда послать результаты прохождения (Рисунок 36).

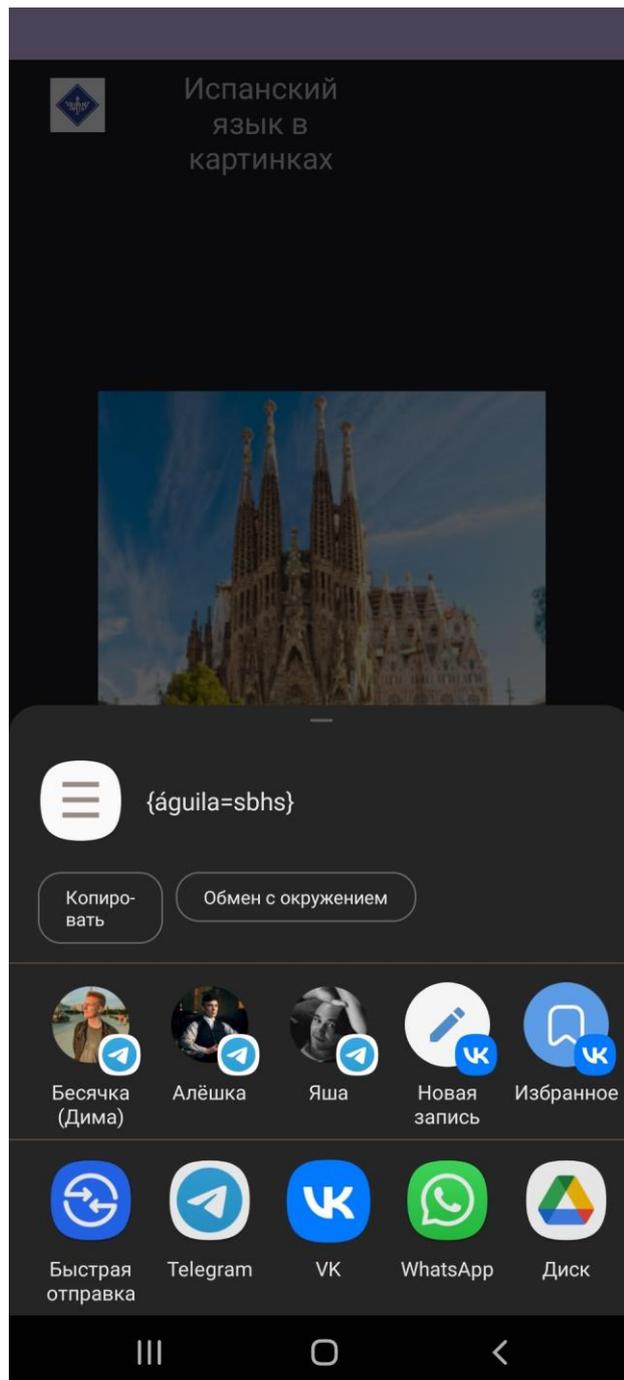


Рисунок 36. Снимок экрана отправка результата

Некоторые нюансы: в режиме тренировка, если пользователь неправильно перевел, то его не пускают дальше, пока он правильно не переведет предложенное задание (Рисунок 37).

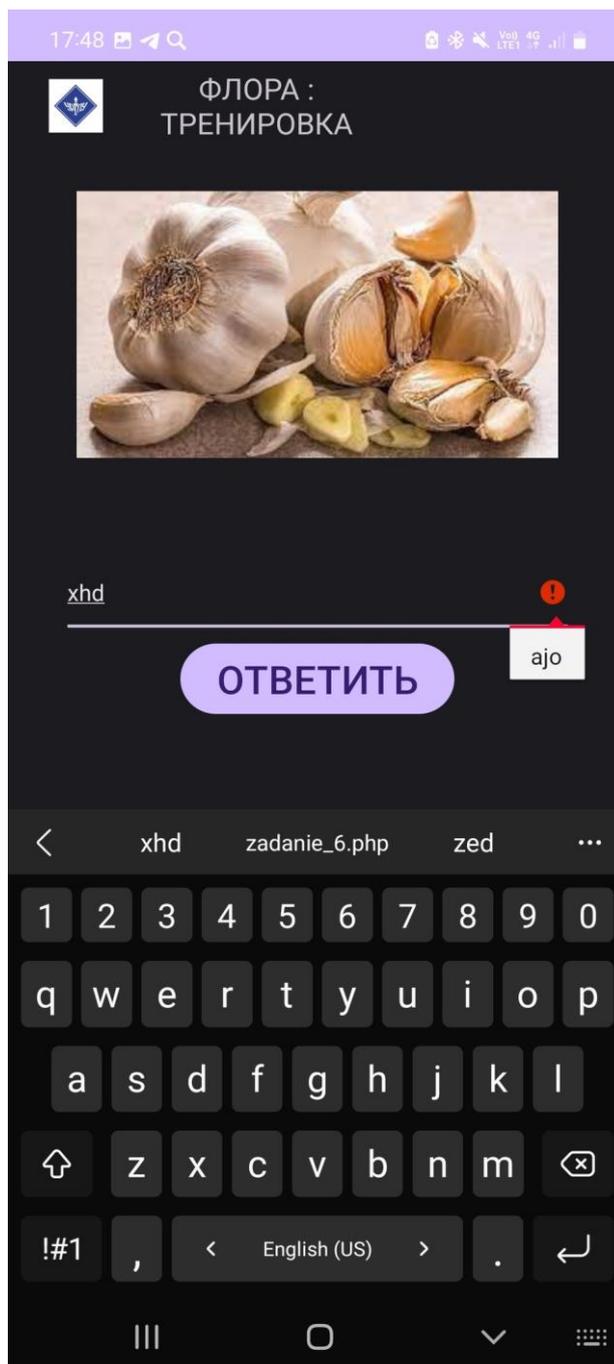


Рисунок 37. Снимок экрана сообщение при неверном ответе в тренировке

### 2.3 Апробация материалов работы

На основе проделанной работы была выполнена апробация в виде научной статьи:

«Мобильное приложение как средство формирования продуктивных лексических навыков при обучении испанскому языку / Д. А. Старкова, Л. В. Сардак, А. А. Васильев, Е. В. Федотова // Актуальные вопросы преподавания мате-

матики, информатики и информационных технологий. – 2023. – № 8. – С. 81-90. – EDN CLAZLJ.»

На разработанную программу получен РИД

«Свидетельство о государственной регистрации программы для ЭВМ № 2024660852 Российская Федерация. Мобильное приложение для изучения лексических единиц “Флора и Фауна на испанском” : № 2024617903 : заявл. 15.04.2024 : опубл. 14.05.2024 / Л. В. Сардак, А. А. Васильев, Д. А. Старкова ; заявитель Федеральное государственное бюджетное образовательное учреждение высшего образования «Уральский государственный педагогический университет». – EDN RHQJTD.»



Рисунок 38. Снимок РИДа

## Заключение

В ходе выполнения работы было получено много новых знаний в среде разработки Android Studio и в языке программирования Kotlin, были получены знания в области работы с базами данных.

Также в ходе работы, были выполнены следующие задачи:

1. Произведен анализ существующих мобильных приложений для изучения иностранного языка.
2. Проанализированы технологии реализации и программные решения для проектирования и разработки мобильного приложения по ОС «Android».
3. Подготовлено техническое задание.
4. В соответствии с техническим заданием проведено проектирование и разработано мобильное приложение.

В будущем это приложение можно развивать, но уже сейчас сделано не мало функций в этом приложении, где студенты нашего университета могут изучить флору и фауну на испанском языке. Также у этого проекта есть своя публикация. И можно сказать, что поставленные задачи выполнены, и цель достигнута.

## Список информационных источников

1. Лучшие приложения для изучения испанского языка. – Текст : электронный // Рочит. Помощник в интернете : [сайт]. – URL: <https://rocit.ru/raccoon/83> (дата обращения: 15.04.2024).
2. Duolingo. – Текст : электронный // Википедия. Свободная энциклопедия : [сайт]. – URL: <https://ru.wikipedia.org/wiki/Duolingo> (дата обращения: 15.04.2024).
3. Учить английский язык duolingo. – Текст : электронный // THESTRIP. Фабрика макияжа : [сайт]. – URL: <https://thestrip.ru/eyebrows/uchit-angliiskii-yazyk-duolingo-duolingo-onlain-programma-izucheniya-yazykov-chemu-uchit/> (дата обращения: 15.04.2024).
4. Busuu — обзор онлайн-сервиса для изучения 12 языков. – Текст : электронный // LANGFORMULA. Блог изучения английского : [сайт]. – URL: <https://langformula.ru/busuu/#2> (дата обращения: 15.04.2024).
5. Busuu. – Текст : электронный // Википедия. Свободная энциклопедия : [сайт]. – URL: <https://ru.wikipedia.org/wiki/Busuu> (дата обращения: 15.04.2024).
6. «СЛОВА БЕГОМ» – интенсивный курс изучения иностранных языков. – Текст : электронный // 4PDA. Площадка о мобильных устройствах : [сайт]. – URL: <https://4pda.to/2014/03/17/147546/> (дата обращения: 15.04.2024).
7. Приложение Puzzle English для изучения английского. – Текст : электронный // PILOT. Школа языка : [сайт]. – URL: <https://pilot-school.ru/prilozhenie-puzzle-english-dlya-izucheniya-anglijskogo-obzor-otzyvy-preimushhestva/> (дата обращения: 15.04.2024).
8. Обзор LinguaLeo. – Текст : электронный // LANGFORMULA. Блог изучения английского: [сайт]. – URL: <https://langformula.ru/lingualeo/#:~:text=LinguaLeo%20%E2%80%94%D0%BE%D0%B4%D0%B8%D0%BD%20%D0%B8%D0%B7%20%D0%BB%D1%83%D1%87%D1%88%D0%B8%D1%85%20%D0%BE%D0%BD%D0%BB%>

D0%B0%D0%B9%D0%BD,%D1%83%D0%BF%D1%80%D0%B0%D0%B6%D0%BD%D1%8F%D1%82%D1%8C%D1%81%D1%8F%20%D0%B2%20%D1%80%D0%B0%D0%B7%D0%BD%D1%8B%D1%85%20%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D1%85%20%D1%82%D0%B5%D0%BC%D0%B0%D1%85. (дата обращения: 15.04.2024).

9. Обзор Wlingua. – Текст : электронный // iGuides. Медиа о гаджетах : [сайт]. – URL:  
[https://www.iguides.ru/main/apps/obzor\\_wlingua\\_servis\\_dlya\\_sereznogo\\_izucheniya\\_angliyskogo\\_na\\_smartfone\\_i\\_v\\_brauzere/](https://www.iguides.ru/main/apps/obzor_wlingua_servis_dlya_sereznogo_izucheniya_angliyskogo_na_smartfone_i_v_brauzere/) (дата обращения: 15.04.2024).
10. Обзор Memrise. – Текст : электронный // LANGFORMULA. Блог изучения английского : [сайт]. – URL: <https://langformula.ru/memrise/> (дата обращения: 15.04.2024).
11. Обзор сервиса Grammarly. – Текст : электронный // Хабр. Коллективный блог : [сайт]. – URL: <https://habr.com/ru/articles/373143/> (дата обращения: 17.04.2024).
12. Grammarize. – Текст : электронный // AppCompass. Приложения и игры для iPhone : [сайт]. – URL: <http://appcompass.ru/grammarize-karmannyiy-trenazher-angliyskoj-gra/> (дата обращения: 17.04.2024).
13. Приложение Cake для изучения английского по видеоурокам. – Текст : электронный // PILOT. Школа языка : [сайт]. – URL: <https://pilot-school.ru/prilozhenie-cake-dlya-izucheniya-anglijskogo-po-videourokam-otzyvy-preimushhestva-i-nedostatki/> (дата обращения: 17.04.2024).
14. Обзор Tandem. – Текст : электронный // LANGFORMULA. Блог изучения английского : [сайт]. – URL:  
<https://langformula.ru/tandem/#:~:text=%D0%9E%D0%B1%D0%B7%D0%BE%D1%80%20Tandem%20%E2%80%94%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D1%8F%20%D0%B4%D0%BB%D1%8F%20%D1%8F%D0%B7%D1%8B%D0%BA%D0%BE%D0%B2>

%D0%BE%D0%B9%20%D0%BF%D1%80%D0%B0%D0%BA%D1%82%D0%B8%D0%BA%D0%B8%20%C2%AB%D0%B2%20%D1%82%D0%B0%D0%BD%D0%B4%D0%B5%D0%BC%D0%B5%C2%BB&text=%D0%9F%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5%20Tandem%20%E2%80%94%D1%8D%D1%82%D0%BE%20%D0%BF%D1%80%D0%BE%D1%81%D1%82%D0%BE%D0%B9%20%D1%81%D0%BF%D0%BE%D1%81%D0%BE%D0%B1,%D1%81%D0%BE%D0%B1%D0%B5%D1%81%D0%B5%D0%B4%D0%BD%D0%B8%D0%BA%20%D0%BF%D0%BE%D0%BC%D0%BE%D0%B6%D0%B5%D1%82%20%D0%B2%D0%B0%D0%BC%20%D1%81%20%D0%B0%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%BC. (дата обращения: 17.04.2024).

15. Mondly. – Текст : электронный // Neosoft. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ БИЗНЕСА : [сайт]. – URL: <https://neosoft.pro/catalog/production?vendor=mondly&product=mondly> (дата обращения: 17.04.2024).

16. Гид по Фигме для начинающих веб-дизайнеров. – Текст : электронный // Tilda Education. Образовательный журнал платформы для создания сайтов : [сайт]. – URL: <https://tilda.education/articles-figma> (дата обращения: 18.04.2024).

17. Как пользоваться Android Studio. – Текст : электронный // SKILLFACTORY MEDIA. Честные истории о карьере в IT : [сайт]. – URL: <https://blog.skillfactory.ru/kak-polzovatsya-android-studio/> (дата обращения: 18.04.2024).

18. Kotlin. – Текст : электронный // Википедия. Свободная энциклопедия : [сайт]. – URL: [https://encyclopaedia.bid/%D0%B2%D0%B8%D0%BA%D0%B8%D0%BF%D0%B5%D0%B4%D0%B8%D1%8F/%D0%9A%D0%BE%D1%82%D0%BB%D0%B8%D0%BD\\_\(%D1%8F%D0%B7%D1%8B%D0%BA\\_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%](https://encyclopaedia.bid/%D0%B2%D0%B8%D0%BA%D0%B8%D0%BF%D0%B5%D0%B4%D0%B8%D1%8F/%D0%9A%D0%BE%D1%82%D0%BB%D0%B8%D0%BD_(%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%)

В8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F) (дата обращения: 18.04.2024).

19. Определение интерфейса в файле XML. – Текст : электронный // METANIT.

Различные языки программирования : [сайт]. – URL: <https://metanit.com/java/android/2.8.php#:~:text=%D0%9A%D0%B0%D0%BA%20%D0%BF%D1%80%D0%B0%D0%B2%D0%B8%D0%BB%D0%BE%2C%20%D0%B4%D0%BB%D1%8F%20%D0%BE%D0%BF%D1%80%D0%B5%D0%B4%D0%B5%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F%20%D0%B2%D0%B8%D0%B7%D1%83%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D0%B3%D0%BE,%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81%D0%B0%20%D0%B2%20%D0%B2%D0%B8%D0%B4%D0%B5%20%D0%BA%D0%BE%D0%B4%D0%B0%20XML>. (дата обращения: 18.04.2024).

20. Что такое Firebase и почему стоит с этим познакомиться. – Текст : электронный // kolmogorov. Блог автора : [сайт]. – URL: <https://kolmogorov.pro/what-is-firebase-что-такое> (дата обращения: 18.04.2024).

21. ГОСТ 34.602-89. Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы. М.: Стандартинформ, 2009. 11 с.

22. Ramus. – Текст : электронный // АКАДЕМИК. academic.ru : [сайт]. – URL: <https://dic.academic.ru/dic.nsf/ruwiki/608820> (дата обращения: 18.04.2024).

## Приложения

### *Приложение 1. Листинг*

```
package com.example.spanish.di.model
```

```
import android.app.Activity
```

```
interface ChangeStringToolBar {  
    fun changeForTheme(c: Activity)  
    fun changeForMode(c: Activity)  
    fun setTheme(theme: String)  
    fun setMode(mode: String)  
    fun getTheme(): String  
}
```

```
package com.example.spanish.di.model
```

```
enum class Requests(val tabl: String, val doc: String, val theme: String) {  
    FAUNA(tabl = "laFaunaDeEspaña", doc = "VW1D4mElhSde23bUCbay", theme  
    = "Фауна"),  
    FLORA(tabl = "laFloraDeEspaña", doc = "9h1vHLhHiuZLe8v00Eoy", theme =  
    "Флора")  
}
```

```
package com.example.spanish.di
```

```
import com.example.spanish.di.model.ChangeStringToolBar  
import com.example.spanish.domain.ChangeStringToolBarImp  
import com.example.spanish.domain.SaveResultImpl
```

```

import com.example.spanish.domain.TakeDataFromFirestoreImpl
import com.example.spanish.domain.TakeRulseImpl
import dagger.Binds
import dagger.Module
import dagger.hilt.InstallIn
import dagger.hilt.components.SingletonComponent
import javax.inject.Singleton

@InstallIn(SingletonComponent::class)
@Module
abstract class ProvideModule {

    @Singleton
    @Binds
    abstract fun provideTakeDataFromFirestore(takeDataFromFirestoreImpl:
TakeDataFromFirestoreImpl): TakeDataFromFirestore

    @Singleton
    @Binds
    abstract fun provideTakeRulse(takeRulseImpl: TakeRulseImpl): TakeRulse

    @Singleton
    @Binds
    abstract fun provideChangeStringToolBar(changeStringToolBarImp: Chang-
eStringToolBarImp): ChangeStringToolBar

    @Singleton
    @Binds

```

```
    abstract fun provideSaveResult(saveResultImpl: SaveResultImpl): SaveResult
}
```

```
package com.example.spanish.di
```

```
import com.example.spanish.domain.Result
```

```
interface SaveResult {
    fun save(answers: Map<String, String>, countAnswer: Int, countTask: Int,
mode: String)
    fun getResult(): Result?
}
```

```
package com.example.spanish.di
```

```
import android.annotation.SuppressLint
import android.text.TextUtils
import android.util.Patterns
import com.google.firebase.auth.ktx.auth
import com.google.firebase.firestore.ktx.firestore
import com.google.firebase.ktx.Firebase
```

```
@SuppressLint("StaticFieldLeak")
```

```
object SingletonObject {
    val auth = Firebase.auth
    val db = Firebase.firestore
    fun signOut() {
        auth.signOut()
    }
}
```

```
}  
}
```

```
fun String.isEmailValid(): Boolean {  
    return !TextUtils.isEmpty(this) &&  
        Patterns.EMAIL_ADDRESS.matcher(this).matches()  
}
```

```
package com.example.spanish.domain
```

```
import android.app.Activity  
import com.example.spanish.MainActivity  
import com.example.spanish.di.model.ChangeStringToolBar  
import javax.inject.Inject
```

```
class ChangeStringToolBarImp @Inject constructor(): ChangeStringToolBar {  
    private var theme: String = ""  
    private var mode: String = ""  
  
    override fun changeForTheme(c: Activity) = (c as MainActivity).changeTextViewLogoTheme(theme)  
  
    override fun changeForMode(c: Activity) = (c as MainActivity).changeTextViewLogoMode(mode)  
  
    override fun setTheme(theme: String) {  
        this.theme = theme  
    }  
}
```

```

    override fun setMode(mode: String) {
        this.mode = mode
    }

    override fun getTheme(): String = theme
}

package com.example.spanish.domain

import com.example.spanish.di.SaveResult
import com.example.spanish.di.TakeRulse
import javax.inject.Inject

class SaveResultImpl @Inject constructor(private val takeRulse: TakeRulse): SaveResult {

    private var result: Result? = null

    override fun getResult(): Result? = result

    override fun save(answers: Map<String, String>, countAnswer: Int, countTask: Int, mode: String) {
        result = Result(answers, countAnswer, countTask, takeRulse.getRulse().theme, mode)
    }
}

data class Result(

```

```
    val answers: Map<String, String>,  
    val countAnswer: Int,  
    val countTask: Int,  
    val theme: String,  
    val mode: String  
)
```

```
package com.example.spanish.presentation.registration
```

```
import androidx.lifecycle.ViewModelProvider  
import android.os.Bundle  
import androidx.fragment.app.Fragment  
import android.view.LayoutInflater  
import android.view.View  
import android.view.ViewGroup  
import android.widget.Toast  
import androidx.lifecycle.Lifecycle  
import androidx.lifecycle.lifecycleScope  
import androidx.lifecycle.repeatOnLifecycle  
import androidx.navigation.fragment.findNavController  
import com.example.spanish.R  
import com.example.spanish.databinding.FragmentRegistrationBinding  
import com.example.spanish.di.model.ChangeStringToolBar  
import dagger.hilt.android.AndroidEntryPoint  
import kotlinx.coroutines.launch  
import javax.inject.Inject  
  
@AndroidEntryPoint
```

```

class RegistrationFragment : Fragment() {

    private lateinit var viewModel: RegistrationViewModel
    private lateinit var binding: FragmentRegistrationBinding
    @Inject
    lateinit var changeStringToolBar: ChangeStringToolBar

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        changeStringToolBar.setTheme(getString(R.string.text_logo))
        changeStringToolBar.changeForTheme(requireActivity())
        return inflater.inflate(R.layout.fragment_registration, container, false)
    }

    @Deprecated("Deprecated in Java")
    override fun onActivityCreated(savedInstanceState: Bundle?) {
        super.onActivityCreated(savedInstanceState)
        viewModel = ViewModelProvider(this)[RegistrationViewModel::class.java]
        // TODO: Use the ViewModel
    }

    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
        super.onViewCreated(view, savedInstanceState)
        binding = FragmentRegistrationBinding.bind(view)
        objectListener()
        observesListener()
    }
}

```

```
}
```

```
private fun objectListener() {  
    binding.run {  
        registerBtn.setOnClickListener {  
            viewModel.createUserWithEmail(  
                email = emailET.text.toString(),  
                password = passwordET.text.toString(),  
                confirmPassword = passwordRepeatET.text.toString()  
            )  
        }  
    }  
}
```

```
private fun observesListener() {  
    lifecycleScope.launch {  
        repeatOnLifecycle(Lifecycle.State.STARTED) {  
            viewModel.uiState.collect { it ->  
                when (it) {  
                    is RegistrationStatus.Error -> Toast.makeText(  
                        requireContext(),  
                        it.e,  
                        Toast.LENGTH_LONG  
                    ).show()  
  
                    RegistrationStatus.Return -> findNavController().popBackStack()  
                }  
            }  
        }  
    }  
}
```

```
    }  
  }  
}
```

```
package com.example.spanish.presentation.auth
```

```
import android.os.Bundle  
import android.view.LayoutInflater  
import android.view.View  
import android.view.ViewGroup  
import android.widget.Toast  
import androidx.fragment.app.Fragment  
import androidx.lifecycle.Lifecycle  
import androidx.lifecycle.ViewModelProvider  
import androidx.lifecycle.lifecycleScope  
import androidx.lifecycle.repeatOnLifecycle  
import androidx.navigation.fragment.findNavController  
import androidx.navigation.navOptions  
import com.example.spanish.R  
import com.example.spanish.databinding.FragmentAuthBinding  
import com.example.spanish.di.model.ChangeStringToolBar  
import dagger.hilt.android.AndroidEntryPoint  
import kotlinx.coroutines.launch  
import javax.inject.Inject  
  
@AndroidEntryPoint  
class AuthFragment : Fragment() {
```

```

// TODO: Rename and change types of parameters

private lateinit var viewModel: AuthViewModel
private lateinit var binding: FragmentAuthBinding
@Inject
lateinit var changeStringToolBar: ChangeStringToolBar

override fun onCreateView(
    inflater: LayoutInflater, container: ViewGroup?,
    savedInstanceState: Bundle?
): View? {
    changeStringToolBar.setTheme(getString(R.string.text_logo))
    changeStringToolBar.changeForTheme(requireActivity())
    return inflater.inflate(R.layout.fragment_auth, container, false)
}

@Deprecated("Deprecated in Java")
override fun onActivityCreated(savedInstanceState: Bundle?) {
    super.onActivityCreated(savedInstanceState)
    viewModel = ViewModelProvider(this)[AuthViewModel::class.java]
    // TODO: Use the ViewModel
}

override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
    super.onViewCreated(view, savedInstanceState)
    binding = FragmentAuthBinding.bind(view)
    onClickListener()
    observesListener()
}

```

```
}
```

```
private fun onClickListener() {  
    binding.run {  
        btSingUp.setOnClickListener {  
            findNavController().navigate(R.id.registrationFragment)  
        }  
        btOk.setOnClickListener {  
            viewModel.signInUserWithEmail(  
                email = enterEmail.text.toString(),  
                password = enterPassword.text.toString()  
            )  
        }  
    }  
}
```

```
private fun observesListener() {  
    lifecycleScope.launch {  
        repeatOnLifecycle(Lifecycle.State.STARTED) {  
            viewModel.uiState.collect {  
                when (it) {  
                    is AuthStatus.Error -> Toast.makeText(  
                        requireContext(),  
                        it.e,  
                        Toast.LENGTH_LONG  
                    ).show()  
                    AuthStatus.NextStep -> openMainNav()  
                }  
            }  
        }  
    }  
}
```

```
    }  
  }  
}  
}
```

```
private fun openMainNav() {  
    findNavController().navigate(R.id.main_host, null, navOptions {  
        popUpTo(R.id.authFragment) {  
            inclusive = true  
        }  
    })  
}  
}
```

```
package com.example.spanish.presentation.mainMenu
```

```
import android.os.Bundle  
import androidx.fragment.app.Fragment  
import android.view.LayoutInflater  
import android.view.View  
import android.view.ViewGroup  
import androidx.navigation.NavController  
import androidx.navigation.fragment.findNavController  
import androidx.navigation.navOptions  
import com.example.spanish.R  
import com.example.spanish.databinding.FragmentMainMenuBinding  
import com.example.spanish.di.SingletonObject  
import com.example.spanish.di.model.ChangeStringToolBar
```

```

import dagger.hilt.android.AndroidEntryPoint
import javax.inject.Inject

@AndroidEntryPoint
class MainMenuFragment : Fragment() {

    private lateinit var binding: FragmentMainMenuBinding
    @Inject
    lateinit var changeStringToolBar: ChangeStringToolBar
    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        changeStringToolBar.setTheme(getString(R.string.text_logo))
        changeStringToolBar.changeForTheme(requireActivity())
        return inflater.inflate(R.layout.fragment_main_menu, container, false)
    }

    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
        super.onViewCreated(view, savedInstanceState)
        binding = FragmentMainMenuBinding.bind(view)
        binding.buttonAnswer.setOnClickListener {
            findNavController().navigate(R.id.chooseThemeFragment)
        }
        binding.logOut.setOnClickListener {
            SingletonObject.signOut()
            findNavController().navigate(R.id.authFragment,null, navOptions {
                popUpTo(R.id.mainMenuFragment) {

```

```
        inclusive = true
    }
})
}
}
```

```
package com.example.spanish.presentation.choose_theme
```

```
import androidx.lifecycle.ViewModelProvider
import android.os.Bundle
import androidx.fragment.app.Fragment
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import androidx.navigation.fragment.findNavController
import com.example.spanish.R
import com.example.spanish.databinding.FragmentChooseThemeBinding
import com.example.spanish.di.model.ChangeStringToolBar
import com.example.spanish.di.model.Requests
import dagger.hilt.android.AndroidEntryPoint
import javax.inject.Inject

@AndroidEntryPoint
class ChooseThemeFragment : Fragment() {

    private lateinit var viewModel: ChooseThemeViewModel
```

```

private lateinit var binding: FragmentChooseThemeBinding
@Inject
lateinit var changeStringToolBar: ChangeStringToolBar

override fun onCreateView(
    inflater: LayoutInflater, container: ViewGroup?,
    savedInstanceState: Bundle?
): View? {
    changeStringToolBar.setTheme(getString(R.string.text_logo))
    changeStringToolBar.changeForTheme(requireActivity())
    return inflater.inflate(R.layout.fragment_choose_theme, container, false)
}

override fun onActivityCreated(savedInstanceState: Bundle?) {
    super.onActivityCreated(savedInstanceState)
    viewModel = ViewModelProvider(
        this)[ChooseThemeViewModel::class.java]
    // TODO: Use the ViewModel
}

override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
    super.onViewCreated(view, savedInstanceState)
    binding = FragmentChooseThemeBinding.bind(view)
    binding.run {
        buttonFauna.setOnClickListener {
            viewModel.setRulse(Requests.FAUNA)
            changeStringToolBar(getString(R.string.fauna))
            nextStep()
        }
    }
}

```

```

    }
    buttonFlora.setOnClickListener {
        viewModel.setRulse(Requests.FLORA)
        changeStringToolBar(getString(R.string.FLORA))
        nextStep()
    }
}
}
}

```

```

private fun changeStringToolBar(s: String) {
    changeStringToolBar.setTheme(s)
    changeStringToolBar.changeForTheme(requireActivity())
}

```

```

private fun nextStep() = findNavController()
.navigate(R.id.modeTestingFragment)
}

```

```

package com.example.spanish.presentation.choose_mode_testing

```

```

import androidx.lifecycle.ViewModelProvider
import android.os.Bundle
import androidx.fragment.app.Fragment
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import androidx.core.os.bundleOf

```

```

import androidx.navigation.fragment.findNavController
import com.example.spanish.MainActivity
import com.example.spanish.R
import
com.example.spanish.databinding.FragmentChooseModeTestingBinding
import com.example.spanish.di.model.ChangeStringToolBar
import dagger.hilt.android.AndroidEntryPoint
import javax.inject.Inject

@AndroidEntryPoint
class ChooseModeTestingFragment : Fragment() {

    private lateinit var viewModel: ChooseModeTestingViewModel
    private lateinit var binding: FragmentChooseModeTestingBinding

    @Inject
    lateinit var changeStringToolBar: ChangeStringToolBar

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        return inflater.inflate(R.layout.fragment_choose_mode_testing, container,
false)
    }

    override fun onActivityCreated(savedInstanceState: Bundle?) {
        super.onActivityCreated(savedInstanceState)
    }

```

```

viewModel = ViewModelProvider(
    this)[ChooseModeTestingViewModel::class.java]
    // TODO: Use the ViewModel
}

override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
    super.onViewCreated(view, savedInstanceState)
    binding = FragmentChooseModeTestingBinding.bind(view)

    binding.imageMain.run {
        when (changeStringToolBar.getTheme()) {
            getString(R.string.FLORA) ->
                setImageResource(R.drawable.fructs)

            getString(R.string.fauna) ->
                setImageResource(R.drawable.animals)
        }
    }

    binding.buttonAnswer.setOnClickListener {
        if (arguments?.getBoolean("skipCountTask", false) == true)
            findNavController().navigate(R.id.testFragment,
                bundleOf("modeTetsting" to true))
        else
            findNavController().navigate(R.id.countTaskFragment)
    }

    binding.buttonMp3.setOnClickListener {
        if (arguments?.getBoolean("skipCountTask", false) == true)

```

```
        findNavController().navigate(R.id.testMP3Fragment, bundleOf("modeTetsting" to true))
    else
        findNavController().navigate(R.id.countTaskFragment, bundleOf("img" to false))
    })
}
}

}
```

```
package com.example.spanish.presentation.mode_testing
```

```
import androidx.lifecycle.ViewModelProvider
import android.os.Bundle
import androidx.fragment.app.Fragment
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import androidx.core.os.bundleOf
import androidx.navigation.fragment.findNavController
import com.example.spanish.R
import com.example.spanish.databinding.FragmentModeTestingBinding
import com.example.spanish.di.model.ChangeStringToolBar
import dagger.hilt.android.AndroidEntryPoint
import javax.inject.Inject

@AndroidEntryPoint
```

```

class ModeTestingFragment : Fragment() {

    private lateinit var viewModel: ModeTestingViewModel
    private lateinit var binding: FragmentModeTestingBinding
    @Inject
    lateinit var changeStringToolBar: ChangeStringToolBar

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        changeStringToolBar.changeForTheme(requireActivity())
        return inflater.inflate(R.layout.fragment_mode_testing, container, false)
    }

    override fun onActivityCreated(savedInstanceState: Bundle?) {
        super.onActivityCreated(savedInstanceState)
        viewModel = ViewModelProvider(this)[ModeTestingViewModel::class.java]
        // TODO: Use the ViewModel
    }

    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
        super.onViewCreated(view, savedInstanceState)
        binding = FragmentModeTestingBinding.bind(view)
        binding.run {
            buttonTest.setOnClickListener {
                changeStringToolBar.setMode(getString(R.string.test))
                changeStringToolBar.changeForMode(requireActivity())
            }
        }
    }
}

```

```

        nextStep(null)
    }
    buttonTrening.setOnClickListener {
        changeStringToolBar.setMode(getString(R.string.Trening))
        changeStringToolBar.changeForMode(requireActivity())
        nextStep(bundleOf("skipCountTask" to true))
    }
}
}

private fun nextStep(bundle: Bundle?) {
    findNavController().navigate(R.id.chooseModeTestingFragment,bundle)
}
}

```

```
package com.example.spanish.presentation.countTask
```

```

import androidx.lifecycle.ViewModelProvider
import android.os.Bundle
import androidx.fragment.app.Fragment
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import androidx.core.os.bundleOf
import androidx.navigation.fragment.findNavController
import com.example.spanish.R
import com.example.spanish.databinding.FragmentCountTaskBinding
import com.example.spanish.di.model.ChangeStringToolBar

```

```

import dagger.hilt.android.AndroidEntryPoint
import javax.inject.Inject

@AndroidEntryPoint
class CountTaskFragment : Fragment() {

    private lateinit var viewModel: CountTaskViewModel

    private lateinit var binding: FragmentCountTaskBinding

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        return inflater.inflate(R.layout.fragment_count_task, container, false)
    }

    override fun onActivityCreated(savedInstanceState: Bundle?) {
        super.onActivityCreated(savedInstanceState)
        viewModel = ViewModelProvider(this)[CountTaskViewModel::class.java]
        // TODO: Use the ViewModel
    }

    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
        super.onViewCreated(view, savedInstanceState)
        binding = FragmentCountTaskBinding.bind(view)

        binding.start.setOnClickListener {

```

```

        checkCount()
    }
}
private fun checkCount() {
    binding.run {
        try {
            val countTask = editTextEnterCountTasks.text.toString().toInt()
            if (countTask in 1..20)
                when(arguments?.getBoolean("img")) {
                    true -> findNavController().navigate(
                        R.id.testFragment, bundleOf(
                            "countTasks" to countTask,
                            "modeTesting" to false
                        )
                    )
                    else -> findNavController().navigate(
                        R.id.testMP3Fragment, bundleOf(
                            "countTasks" to countTask,
                            "modeTesting" to false
                        )
                    )
                }
        }
        else
            editTextEnterCountTasks.error = "Введите цифру от 1 до 20"
    }
}
}

```

```
}  
}
```

```
package com.example.spanish.presentation.test.img
```

```
import android.os.Bundle  
import android.os.LocaleList  
import android.util.Log  
import android.view.LayoutInflater  
import android.view.View  
import android.view.ViewGroup  
import android.widget.Toast  
import androidx.core.os.bundleOf  
import androidx.fragment.app.Fragment  
import androidx.fragment.app.viewModels  
import androidx.lifecycle.Lifecycle  
import androidx.lifecycle.lifecycleScope  
import androidx.lifecycle.repeatOnLifecycle  
import androidx.navigation.fragment.findNavController  
import com.bumptech.glide.Glide  
import com.example.spanish.R  
import com.example.spanish.databinding.FragmentTestBinding  
import dagger.hilt.android.AndroidEntryPoint  
import kotlin.coroutines.launch  
import java.util.Locale  
  
@AndroidEntryPoint  
class TestFragment : Fragment() {
```

```

private val viewModel: TestViewModel by viewModels()
private lateinit var binding: FragmentTestBinding
private var countTask: Int = 0
private var modeTesting: Boolean = true

override fun onCreateView(
    inflater: LayoutInflater, container: ViewGroup?,
    savedInstanceState: Bundle?
): View? {
    return inflater.inflate(R.layout.fragment_test, container, false)
}

override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
    super.onViewCreated(view, savedInstanceState)
    countTask = arguments?.getInt("countTasks", 5) ?: 5
    modeTesting = arguments?.getBoolean("modeTesting", false)?: false
    Log.e(null, "countTask = $countTask\nmodeTesting = $modeTesting")
    binding = FragmentTestBinding.bind(view)
    binding.enterAnswer.imeHintLocales = LocaleList(Locale("es", "ES"))
    viewModel.setCountTaskAndModeTesting(countTask, modeTesting)
    observesListenerStateTestFragment()
    listener()
}

```

```

private fun observesListenerStateTestFragment() {
    lifecycleScope.launch {
        repeatOnLifecycle(Lifecycle.State.STARTED) {
            viewModel.test.collect {
                when (it) {
                    is Test.Error -> Toast.makeText(
                        requireContext(),
                        it.e,
                        Toast.LENGTH_LONG
                    ).show()
                    Test.PlayLoad -> binding.progressBar.visibility = View.VISIBLE
                    Test.StopLoad -> binding.progressBar.visibility = View.GONE
                    is Test.TestImg -> {
                        Log.e(null, "it = ${it.img}")
                        loadImg(it.img)
                    }
                    is Test.ErrorAnswer -> {
                        binding.enterAnswer.error = it.error
                    }
                    is Test.NextStep -> findNavController().navigate(R.id.finishFragment, bundleOf(
                        "countAnswer" to it.count,
                        "countTasks" to countTask
                    ))
                    Test.ClearEditText -> binding.enterAnswer.text.clear()
                }
            }
        }
    }
}

```

```

    }
}

private fun listener() {
    binding.buttonAnswer.setOnClickListener {
        val str = binding.enterAnswer.text.toString().filterNot { it.isWhitespace()
}.lowercase()
        viewModel.check(str)
    }
}

private fun loadImg(img: String?) {
    Log.e("img",img?: "null")
    Glide
        .with(this)
        .load(img)
        .error(R.drawable.baseline_cloud_off_24)
        .into(binding.imageMain)
}

}

package com.example.spanish.presentation.test.mp3

import android.media.MediaPlayer
import android.net.Uri
import android.os.Bundle
import android.os.LocaleList

```

```
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.Toast
import androidx.core.os.BundleOf
import androidx.fragment.app.Fragment
import androidx.fragment.app.viewModels
import androidx.lifecycle.Lifecycle
import androidx.lifecycle.LifecycleScope
import androidx.lifecycle.RepeatOnLifecycle
import androidx.navigation.fragment.findNavController
import com.example.spanish.R
import com.example.spanish.databinding.FragmentTestMp3Binding
import dagger.hilt.android.AndroidEntryPoint
import kotlinx.coroutines.delay
import kotlinx.coroutines.launch
import java.util.Locale
import kotlin.properties.Delegates
```

```
@AndroidEntryPoint
```

```
class TestMP3Fragment: Fragment() {
```

```
    private val testMP3ViewModel by viewModels<TestMP3ViewModel>()
```

```
    private lateinit var binding: FragmentTestMp3Binding
```

```
    private var mPlayer: MediaPlayer? = null
```

```
    private var countTask by Delegates.notNull<Int>()
```

```

override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    mediaPlayer = MediaPlayer()
}

```

```

override fun onCreateView(
    inflater: LayoutInflater, container: ViewGroup?,
    savedInstanceState: Bundle?
): View? {
    return inflater.inflate(R.layout.fragment_test_mp3, container, false)
}

```

```

override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
    super.onViewCreated(view, savedInstanceState)
    countTask = arguments?.getInt("countTasks",5): 5
    testMP3ViewModel.setCountTaskAndModeTesting(
        countTask,
        arguments?.getBoolean("modeTetsting"): false
    )
    binding = FragmentTestMp3Binding.bind(view)
    binding.enterAnswer.imeHintLocales = LocaleList(Locale("es", "ES"))
    initListenerButton()
    initListenerViewModel()
}

```

```

override fun onPause() {
    super.onPause()
}

```

```

    mPlayer?.pause()
}

override fun onStop() {
    super.onStop()
    mPlayer?.stop()
}

override fun onDestroy() {
    super.onDestroy()
    mPlayer?.release()
    mPlayer = null
}

private fun pause() {
    mPlayer?.pause()
    binding.icPlayPause.setImageResource(R.drawable.baseline_play_circle_24)
}

private fun play() {
    mPlayer?.start()
    viewLifecycleOwner.lifecycleScope.launch {
        while (mPlayer?.isPlaying == true) {
            binding.seekBar.max = mPlayer?.duration?: 0
            binding.seekBar.progress = mPlayer?.currentPosition?: 0
            delay(50)
        }
    }
}

```

```

        bind-
        ing.icPlayPause.setImageResource(R.drawable.baseline_pause_circle_24)
    }

    private fun initListenerButton() {
        binding.run {
            icPlayPause.setOnClickListener {
                if (mPlayer?.isPlaying == true)
                    pause()
                else if (mPlayer?.isPlaying == false) {
                    play()
                }
            }
        }

        binding.buttonAnswer.setOnClickListener {
            val str = binding.enterAnswer.text.toString().filterNot {
it.isWhitespace() }.lowercase()
                testMP3ViewModel.check(str)
            }
        }
    }

    private fun initMediaPlayer(id: Int) {
        mPlayer?.reset()
        binding.icPlayPause.setImageResource(R.drawable.baseline_play_circle_24)
        mPlayer?.let {
            mediaPlayer ->
            mediaPlayer.setDataSource(requireContext(),
Uri.parse("android.resource:///"+ requireContext().packageName + "/" + id))

```

```

mediaPlayer.prepare()
mediaPlayer.setOnCompletionListener {
    if (!it.isPlaying)
        pause()
    }
}
}
}

```

```

private fun initListenerViewModel() {
    viewLifecycleOwner.lifecycleScope.launch {
        repeatOnLifecycle(Lifecycle.State.RESUMED) {
            testMP3ViewModel.state.collect {
                when(it) {
                    is TestMP3State.IdSound -> {
                        initMediaPlayer(it.id)
                    }
                    is TestMP3State.ErrorMessage -> {
                        Toast.makeText(requireContext(),
it.e,Toast.LENGTH_LONG).show()
                    }
                    is TestMP3State.ErrorAnswer -> {
                        binding.enterAnswer.error = it.e
                    }
                    is TestMP3State.NextStep -> findNavController()
                    .navigate(R.id.finishFragment, bundleOf(
                        "countAnswer" to it.count,
                        "countTasks" to countTask
                    )
                )
            }
        }
    }
}

```

```
    )  
    TestMP3State.ClearEditText -> binding.enterAnswer.text.clear()  
    }  
    }  
    }  
    }  
    }  
    }  
}
```

```
package com.example.spanish
```

```
import androidx.appcompat.app.AppCompatActivity
```

```
import android.os.Bundle
```

```
import com.example.spanish.databinding.ActivityMainBinding
```

```
import dagger.hilt.android.AndroidEntryPoint
```

```
@AndroidEntryPoint
```

```
class MainActivity : AppCompatActivity() {
```

```
    private lateinit var binding: ActivityMainBinding
```

```
    override fun onCreate(savedInstanceState: Bundle?) {
```

```
        super.onCreate(savedInstanceState)
```

```
        binding = ActivityMainBinding.inflate(layoutInflater)
```

```
        setContentView(binding.root)
```

```
    }
```

```
    fun changeTextViewLogoTheme(s: String?) {
```

```
binding.textLogo.text = s
}

fun changeTextViewLogoMode(s: String?) {
    if (s != null)
        binding.textLogo.append(" : $s")
    }
}
```