МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ УЧРЕЖДЕНИЕ

КАЛИНИНГРАДСКОЙ ОБЛАСТИ

ПРОФЕССИОНАЛЬНАЯ ОБРАЗОВАТЕЛЬНАЯ ОРГАНИЗАЦИЯ

«КОЛЛЕДЖ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И СТРОИТЕЛЬСТВА»

(ГБУ КО ПОО «КИТиС»)

|  |  |  |
| --- | --- | --- |
| Курсовой проект  допущен к защите  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (дата)  Зам.директора по УМР  \_\_\_\_\_\_\_\_\_\_\_\_\_Павленко Г.Я.  (подпись) |  | Курсовой проект  защищен с оценкой  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (оценка)  Руководитель работы  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) |

Пояснительная записка к курсовому проекту

по дисциплине: МДК 01.03 Разработка мобильных приложений

Тема «Разработка мобильного приложения для синтеза речи и чтения текстов»

Специальность 09.02.07 «Информационные системы и программирование»

|  |  |
| --- | --- |
| Выполнил: | студент 3 курса,  группы ИС18-1  Коробов Артур Эдуардович  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) |
| Руководитель: | Большакова-Стрекалова Анна Викторовна, преподаватель ГБУ КО ПОО КИТиС  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) |

Калининград

2020

Содержание

[Введение 3](#_Toc59038928)

[1 Описание предметной области 4](#_Toc59038929)

[1.1 Аналоги разрабатываемого приложения 5](#_Toc59038930)

[1.2 Техническое задание 5](#_Toc59038931)

[1.2.1 Введение 5](#_Toc59038932)

[1.2.2 Основания для разработки 6](#_Toc59038933)

[1.2.3 Назначение разработки 6](#_Toc59038934)

[1.2.4 Требования к приложению 6](#_Toc59038935)

[1.2.5 Требования к программной документации 7](#_Toc59038936)

[1.2.6 Стадии и этапы разработки 7](#_Toc59038937)

[1.2.7 Порядок контроля и приемки 8](#_Toc59038938)

[1.3 Описание структуры приложения 8](#_Toc59038939)

[2 Описание разработки приложения 9](#_Toc59038940)

[2.1 Обоснование средств разработки 9](#_Toc59038941)

[2.2 Разработка интерфейса 9](#_Toc59038942)

[2.3 Разработка логики работы приложения 10](#_Toc59038943)

[3 Тестирование и установка приложения 11](#_Toc59038944)

[Заключение 13](#_Toc59038945)

[Список использованной литературы 14](#_Toc59038946)

[Приложения 15](#_Toc59038947)

# Введение

С увеличением доступности мобильных устройств рынок мобильных приложений растет. Мобильные приложения позволяют решать огромный спектр пользовательских задач.

Разработка мобильных приложений рассчитана на определенное предназначение. Некоторые программы позволяют повсеместно осуществлять соединение с сетью, другие указывают маршрут, третьи оказывают помощь в поиске и покупке необходимых товаров и услуг. Поэтому разработка приложений для мобильных устройств развивается и актуальна на сегодняшний день.

Целью данной курсовой работы является разработка мобильного приложения для синтеза речи и чтения текстов.

Для достижения поставленной цели необходимо решить ряд задач:

1. провести анализ предметной области;
2. составить техническое задание;
3. описать структуру приложения;
4. разработать интерфейс приложения;
5. разработать логику работы приложения;
6. провести тестирование и установку приложения.

Для описания предметной области и проектирования схем будем использовать диаграммы UML. Язык моделирования UML предоставляет выразительные средства для создания визуальных моделей и обладает рядом значимых преимуществ:

1. UML объектно-ориентирован, в результате чего методы описания результатов анализа и проектирования семантически близки к методам программирования на современных объектно-ориентированных языках;
2. UML позволяет описать систему практически со всех возможных точек зрения и разные аспекты поведения системы;
3. диаграммы UML сравнительно просты для чтения после быстрого ознакомления с его синтаксисом;
4. UML получил широкое распространение и динамично развивается.

# 1 Описание предметной области

Разработка мобильных приложений под Android на сегодняшний день очень востребована ввиду высокой популярности данной ОС.

Данная операционная система позволяет использовать синтез речи и распознавание речи.

Синтез речи или Text-to-Speech (TTS) — технология преобразования текста в речь. Это компьютерное моделирование человеческой речи из текстового представления при помощи методов машинного обучения.

Распознавание речи или Speech-to-Text (STT) — технология преобразования речевого сигнала в текстовый поток.

Speechy – мобильное приложение, с помощью которого можно будет озвучить текст введённый пользователем или же преобразовать пользовательскую речь в текст. Целевой аудиторией приложения являются люди разных возрастов. Пользователь может вести определенный текст, а затем одним нажатием кнопки озвучить его. Также пользователь может преобразовать свою речь в текст и увидеть написание распознанных слов.

Speechy можно отнести к категории обучающих приложений, поскольку технологии синтеза и распознавания речи могут помочь пользователю определить написание и произношение слов соответственно.

# Аналоги разрабатываемого приложения

1.1.1 Синтез речи от Google

Наиболее популярное и доступное приложение с поддержкой технологии TTS, совместимое со многими устройствами на ОС Android.

1.1.2 Acapela TTS Voices

Приложение является условно бесплатным, при этом перед тем, как купить приложение, есть возможность прослушать демо-версии мужских и женских голосов в режиме онлайн. Приложение позволяет покупать и устанавливать  программы с голосом высокого качества, такие как Google TalkBack, переводчики [приложений для чтения книг](https://gemapps.ru/sravnenie/kak-i-chem-otkryt-epub-i-fb2-fajly-na-android) и озвучивания новостей.

1.1.3 Vocalizer TTS Voice

Синтезатор речи с меню на английском языке. Приложение обладает интегрированным TTS-движком, поддерживающим более 50 языков. Есть возможность настройки тембра голоса.

# Техническое задание

# 1.2.1 Введение

1.2.1.1. Наименование: «Speechy».

1.2.1.2. Краткая характеристика области применения: приложение предназначено для синтеза речи и чтения текстов. Основная цель разрабатываемого приложения – использование технологий TTS и STT для определения написания и произношения пользовательского ввода.

# 1.2.2 Основания для разработки

1.2.2.1. Основанием для проведения разработки служит приказ №\_\_ от \_\_ декабря 2020 года. Приказ утвержден директором  ГБУ КО ПОО «КИТиС».

1.2.2.2. Наименование темы разработки – «Разработка мобильного приложения для синтеза речи и чтения текстов». Уловное обозначение темы разработки – «Speechy».

# 1.2.3 Назначение разработки

1.2.3.1. Функциональное назначение: приложение предоставляет возможность работы с технологиями TTS и STT.

1.2.3.2. Эксплуатационное назначение: приложение может эксплуатироваться на мобильных устройствах. Пользователями приложения могут являться люди любых возрастов.

# 1.2.4 Требования к приложению

1.2.4.1. Требования к функциональным характеристикам: приложение должно обеспечивать возможность работы с технологиями TTS и STT (синтез речи и распознавание речи).

1.2.4.2. Требование к надежности: приложение должно выполняться без ошибок. Необходимо обеспечить проверку входных и выходных данных.

1.2.4.3. Условия эксплуатации: необходимо мобильное устройство под управлением ОС Android. Целевой пользователь – человек любого возраста.

1.2.4.4. Требования к информационной и программной совместимости:

1.2.4.4.1. Используемый язык программирования – Java.

1.2.4.4.2. Используемая среда разработки – Android Studio.

1.2.4.4.3. Целевая ОС – Android.

1.2.4.5. Требования к маркировке и упаковке: приложение поставляется в виде исполнительного файла ОС Android с расширением .apk. Требует установки. Требования к маркировке и упаковке не предъявляются.

1.2.4.6. Требования к транспортированию и хранению:

1.2.4.6.1. Ссылка на GitHub проекта: <https://github.com/unicorndk/SpeechyProject>

# 1.2.5 Требования к программной документации

«Приложение для синтеза речи и чтения текстов». Техническое задание;

# 1.2.6 Стадии и этапы разработки

1.2.6.1. Стадии разработки:

* техническое задание;
* технический и рабочий проекты;
* внедрение.

1.2.6.1. Этапы разработки:

На стадии «Техническое задание» должен быть выполнен этап разработки, согласования и утверждения настоящего технического задания.

На стадии «Технический (и рабочий) проект» должны быть выполнены перечисленные ниже этапы работ:

* разработка дизайна приложения;
* разработка приложение;
* проведение тестов.

На стадии «Внедрение» должен быть выполнен этап разработки «Порядок контроля и приемки».

# 1.2.7 Порядок контроля и приемки

Производится проверка корректного выполнения приложения, заложенных в него функций, т.е. осуществляется функциональное тестирование приложения.

# 1.3 Описание структуры приложения

Структура приложения представлена в виде диаграммы структурной схемы.

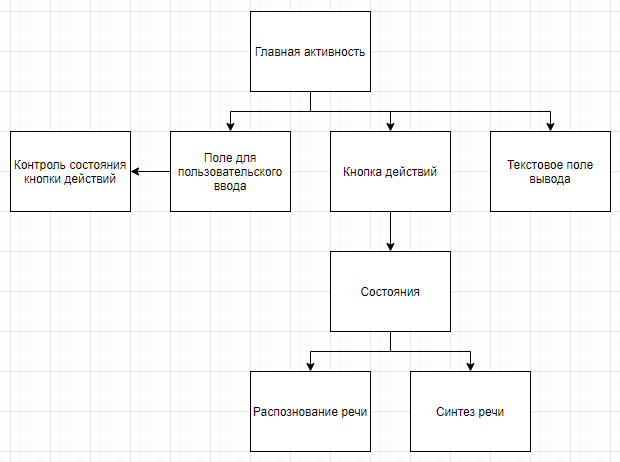


Рисунок 1 – Структурная схема приложения

# 2 Описание разработки приложения

# 2.1 Обоснование средств разработки

Основными средствами при разработке приложения являются: среда разработки (IDE), язык программирования, эмулятор для тестов.

Приложение Chatty реализовано на языке Java. В качестве среды разработки была выбрана Android Studio, на мой взгляд, это лучшая среда разработки под платформу Android.

# 2.2 Разработка интерфейса

Для проектирования интерфейса приложения использовался язык разметки XML, а также встроенные средства проектирования Android Studio.

Дизайн приложения выполнен в стиле Material Design.

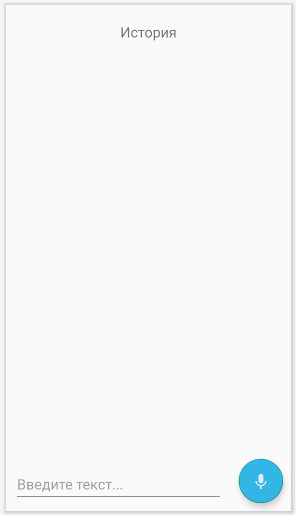


Рисунок 2 – Макет приложения в Android Studio

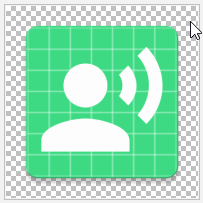


Рисунок 3 – иконка приложения в Android Studio

# 2.3 Разработка логики работы приложения

Логика работы и схема взаимодействия компонентов проекта представлены в виде функциональной схемы на рисунке 2.

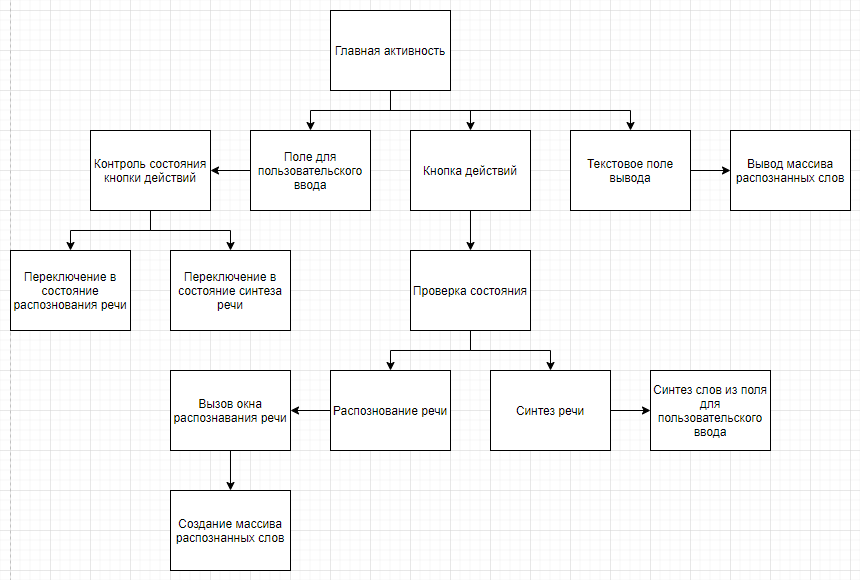


Рисунок 4 – Функциональная схема приложения

# 3 Тестирование и установка приложения

Тестирование приложения проводилось на эмуляторе, а также на устройстве.

Для корректной работы приложения на реальном устройстве необходимо добавить в манифест разрешение «android.permission.RECORD\_AUDIO». Данное разрешение предоставляет доступ к микрофону.

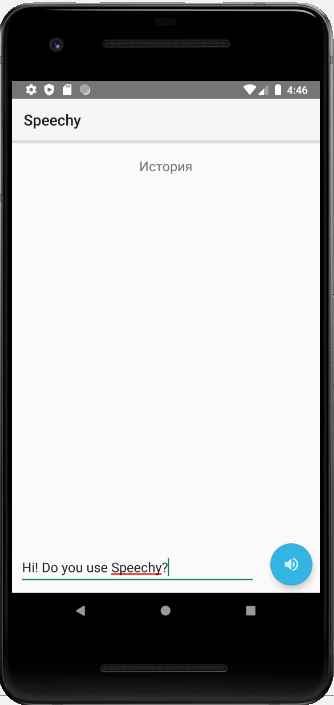
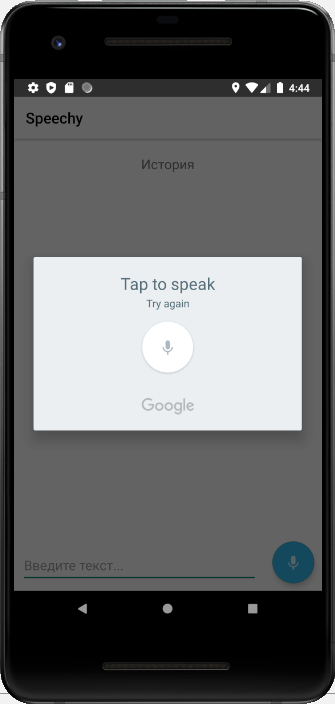


Рисунок 5 – Тестирование в эмуляторе (Pixel 2 API 27)

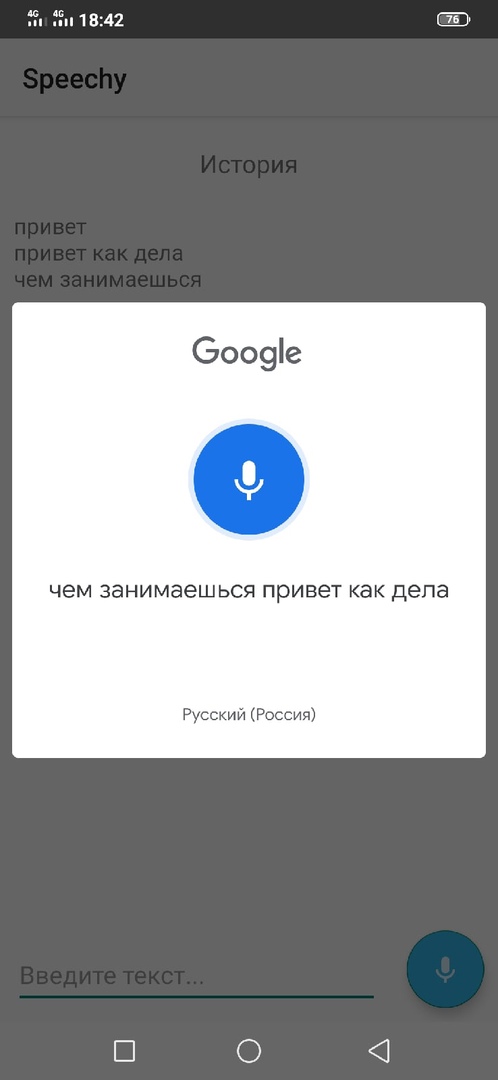


Рисунок 6 – Тестирование на устройстве (Vivo 1915 API 28)

# Заключение

В результате написания данной курсовой работы было спроектировано и разработано мобильное приложение для синтеза речи и чтения текстов «Chatty».

В рамках курсовой работы были решены следующие задачи:

1. выполнен анализ предметной области;
2. выполнен анализ рынка на предмет аналогов;
3. разработано техническое задание;
4. спроектирована структурная схема приложения;
5. разработан современный дизайн;
6. разработана логика взаимодействия компонентов приложения;
7. выполнен ряд тестов на эмуляторе и реальных устройствах;
8. собран исполняемый файл с расширением «.apk» при помощи встроенного компилятора Android Studio.

В ходе проектирования применялись такие UML-средства как diagrams.net(draw.io) для проектирования структурной и функциональных схем приложения.

# Список использованной литературы

Основная

1. Голощапов А.Л. Android. Создание приложений для смартфонов и планшетных ПК, СПб: БХВ-Петербург, 2016 г.

2. Дэрси, Л. Разработка приложений для Android-устройств. Т. 1: Базовые принципы / Л. Дэрси, Ш. Кондер. - М.: Лори, 2018. - 402 c.

3. Вейл, Эстель HTML5. Разработка приложений для мобильных устройств / Эстель Вейл. - М.: Питер, 2015. - 818 c.

4. Колисниченко, Денис Программирование для Android / Денис Колисниченко. - М.: БХВ-Петербург, 2015. - 706 c.

5. Фелкер, Д. Android. Разработка приложений для чайников / Д. Фелкер. - М.: Диалектика / Вильямс, 2014. - 783 c.

6. Машнин, Т. С. Eclipse. Разработка RCP-, Web-, Ajax- и Android-приложений на Java / Т.С. Машнин. - М.: БХВ-Петербург, 2016. - 384 c.

Дополнительная

1.Хашими С., Коматинени С., Маклин Д. Разработка приложений для Android. - СПб.: Питер, 2015. – 736 с.

2. Фримен Эрик, Робсон Элизабет, Сиерра Кэти, Бейтс Берт Bert Bates. Head First. Паттерны проектирования. Обновленное юбилейное издание. – СПб: Питер, 2018.

3.Фримен Эрик, Робсон Элизабет. Изучаем программирование на JavaScript. – СПб: Питер, 2018.

# Приложения

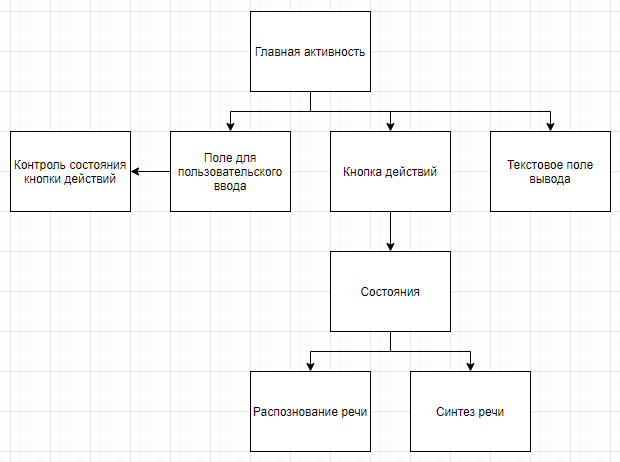


Рисунок 1 – Структурная схема приложения

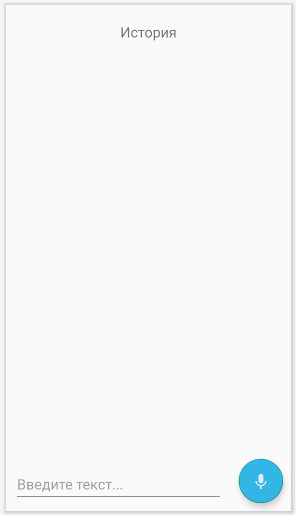


Рисунок 2 – Макет приложения в Android Studio

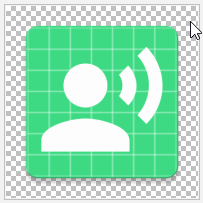


Рисунок 3 – иконка приложения в Android Studio

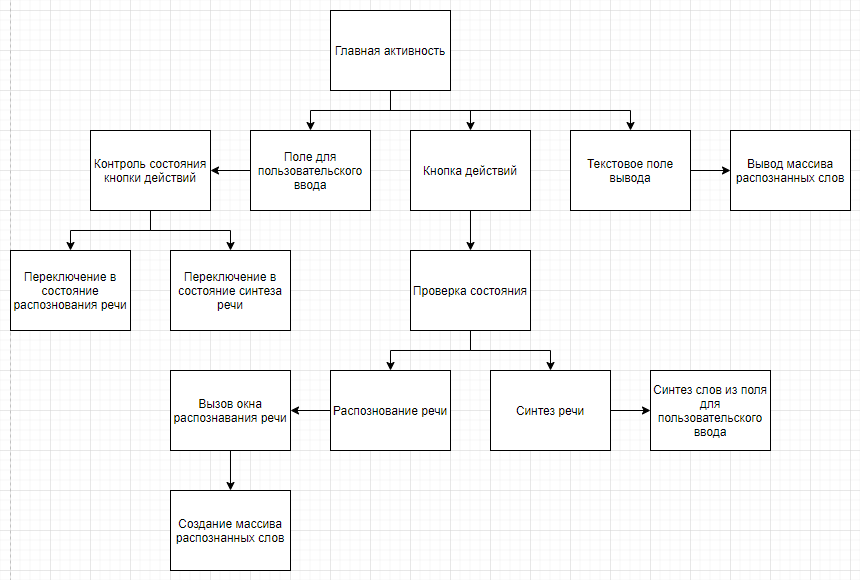


Рисунок 4 – Функциональная схема приложения

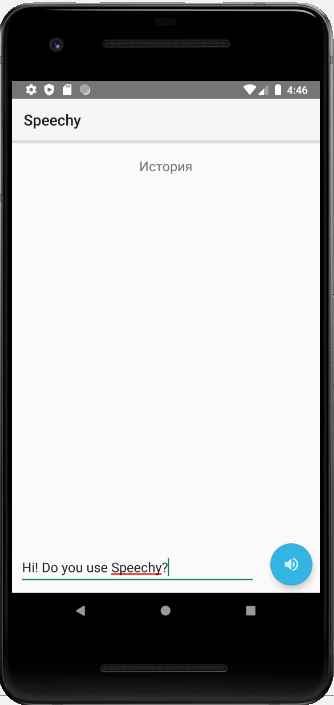
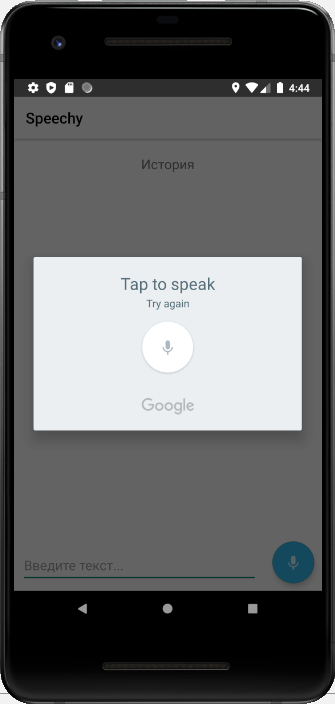


Рисунок 5 – Тестирование в эмуляторе (Pixel 2 API 27)

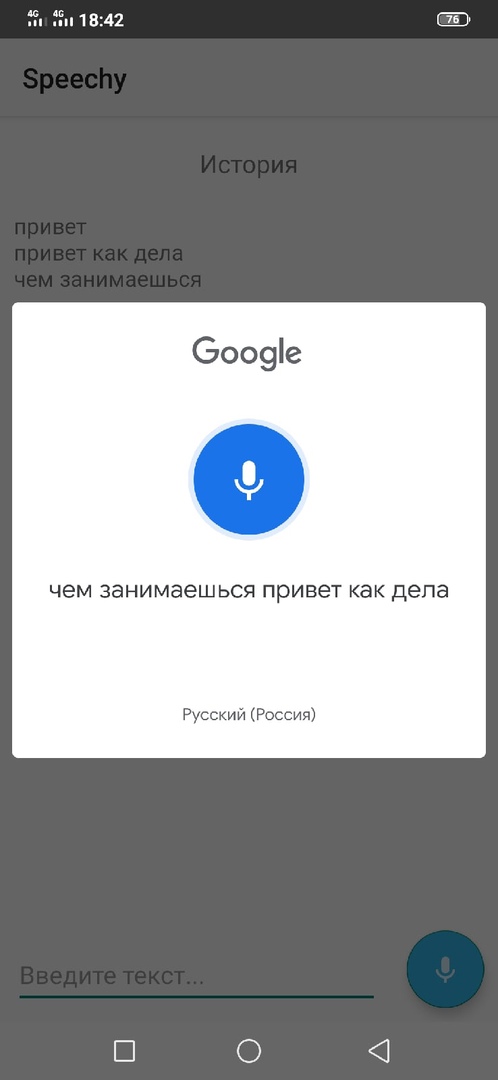


Рисунок 6 – Тестирование на устройстве (Vivo 1915 API 28)

Листинг главной активности приложения:

**package** com.unicorndk.speechy;  
  
**import** androidx.annotation.Nullable;  
**import** androidx.appcompat.app.AppCompatActivity;  
  
**import** android.content.Intent;  
**import** android.os.Bundle;  
**import** android.speech.RecognizerIntent;  
**import** android.speech.tts.TextToSpeech;  
**import** android.text.Editable;  
**import** android.text.TextWatcher;  
**import** android.text.method.ScrollingMovementMethod;  
**import** android.util.Log;  
**import** android.view.View;  
**import** android.widget.EditText;  
**import** android.widget.TextView;  
**import** android.widget.Toast;  
  
**import** com.google.android.material.floatingactionbutton.FloatingActionButton;  
  
**import** java.util.ArrayList;  
**import** java.util.Locale;  
  
**import static** android.icu.lang.UCharacter.GraphemeClusterBreak.***L***;  
  
**public class** MainActivity **extends** AppCompatActivity {  
 **private static final int *REQUEST\_CODE\_SPEECH\_INPUT*** = 1000;  
 TextView **textView**;  
 FloatingActionButton **voiceButton**;  
 EditText **editText**;  
  
 **private** TextToSpeech **mTTS**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
  
 **mTTS** = **new** TextToSpeech(**this**, **new** TextToSpeech.OnInitListener() {  
 @Override  
 **public void** onInit(**int** status) {  
 **if** (status == TextToSpeech.***SUCCESS***) {  
 **int** result = **mTTS**.setLanguage(Locale.***ENGLISH***);  
  
 **if** (result == TextToSpeech.***LANG\_MISSING\_DATA*** || result == TextToSpeech.***LANG\_NOT\_SUPPORTED***) {  
 Log.*e*(**"TTS"**, **"Данный язык не поддерживается!"**);  
 } **else** {  
  
 }  
 } **else** {  
 Log.*e*(**"TTS"**, **"Что-то пошло не так!"**);  
 }  
 }  
 });  
  
 **textView** = findViewById(R.id.***text\_view***);  
 **textView**.setMovementMethod(**new** ScrollingMovementMethod());  
  
 **voiceButton** = findViewById(R.id.***voice\_button***);  
 **editText** = findViewById(R.id.***edit\_text***);  
  
 **editText**.addTextChangedListener(**new** TextWatcher() {  
 @Override  
 **public void** beforeTextChanged(CharSequence s, **int** start, **int** count, **int** after) {  
  
 }  
  
 @Override  
 **public void** onTextChanged(CharSequence s, **int** start, **int** before, **int** count) {  
 **if** (**editText**.getText().length() == 0) {  
 **voiceButton**.setImageResource(R.drawable.***ic\_stt\_button***);  
 } **else** {  
 **voiceButton**.setImageResource(R.drawable.***ic\_tts\_button***);  
 }  
 }  
  
 @Override  
 **public void** afterTextChanged(Editable s) {  
  
 }  
 });  
 **voiceButton**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **if** (**editText**.getText().length() == 0) {  
 speechToText();  
 } **else** {  
 textToSpeech();  
 }  
  
 }  
 });  
 }  
  
 **private void** textToSpeech() {  
 String text = **editText**.getText().toString();  
 **mTTS**.speak(text, TextToSpeech.***QUEUE\_FLUSH***, **null**);  
  
  
 }  
  
 @Override  
 **protected void** onDestroy() {  
 **if** (**mTTS** != **null**) {  
 **mTTS**.stop();  
 **mTTS**.shutdown();  
 }  
 **super**.onDestroy();  
 }  
  
 **private void** speechToText() {  
 Intent intent = **new** Intent(RecognizerIntent.***ACTION\_RECOGNIZE\_SPEECH***);  
 intent.putExtra(RecognizerIntent.***EXTRA\_LANGUAGE\_MODEL***,  
 RecognizerIntent.***LANGUAGE\_MODEL\_FREE\_FORM***);  
  
 intent.putExtra(RecognizerIntent.***EXTRA\_LANGUAGE***, Locale.*getDefault*());  
 intent.putExtra(RecognizerIntent.***EXTRA\_PROMPT***, **"Привет! Скажите что-нибудь."**);  
  
 **try** {  
 startActivityForResult(intent, ***REQUEST\_CODE\_SPEECH\_INPUT***);  
  
 } **catch** (Exception e) {  
 Toast.*makeText*(**this**, **"Что-то пошло не так!"**, Toast.***LENGTH\_LONG***).show();  
  
 }  
 }  
  
 @Override  
 **protected void** onActivityResult(**int** requestCode, **int** resultCode, @Nullable Intent data) {  
 **super**.onActivityResult(requestCode, resultCode, data);  
 **switch** (requestCode) {  
 **case *REQUEST\_CODE\_SPEECH\_INPUT***: {  
 **if** (resultCode == ***RESULT\_OK*** && **null** != data) {  
 ArrayList<String> result = data.getStringArrayListExtra(RecognizerIntent.***EXTRA\_RESULTS***);  
 **textView**.setText(**textView**.getText() + result.get(0) + **"\n"**);  
 }  
 **break**;  
 }  
 }  
 }  
}

Листинг макета главной активности:

*<?***xml version="1.0" encoding="utf-8"***?>*<**RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:theme="@style/Theme.Design.Light"  
 tools:context=".MainActivity"**>  
  
 <**TextView  
 android:id="@+id/label"  
 android:layout\_width="match\_parent"  
 android:layout\_height="48dp"  
 android:layout\_marginLeft="10dp"  
 android:layout\_marginTop="10dp"  
 android:layout\_marginRight="10dp"  
 android:text="История"  
 android:textSize="18dp"  
 android:gravity="center"**>  
  
 </**TextView**>  
 <**TextView  
 android:textSize="16dp"  
 android:id="@+id/text\_view"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_below="@+id/label"  
 android:layout\_above="@id/edit\_text"  
 android:layout\_marginLeft="10dp"  
 android:layout\_marginRight="10dp"  
 android:layout\_marginTop="10dp"**>  
 </**TextView**>  
 <**EditText  
 android:id="@+id/edit\_text"  
 android:layout\_width="match\_parent"  
 android:layout\_height="48dp"  
 android:layout\_alignParentBottom="true"  
 android:layout\_toLeftOf="@+id/voice\_button"  
 android:layout\_margin="10dp"  
 android:hint="Введите текст..."**>  
  
 </**EditText**>  
 <**com.google.android.material.floatingactionbutton.FloatingActionButton  
 android:id="@+id/voice\_button"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:clickable="true"  
 app:fabSize="normal"  
 android:src="@drawable/ic\_stt\_button"  
 android:backgroundTint="@android:color/holo\_blue\_light"  
 android:tint="@android:color/white"  
 android:layout\_alignParentBottom="true"  
 android:layout\_alignParentEnd="true"  
 android:layout\_margin="10dp"**>  
  
 </**com.google.android.material.floatingactionbutton.FloatingActionButton**>  
</**RelativeLayout**>