|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |

**Институт Информационных технологий (ИИТ)**

**Кафедра Математического обеспечения и стандартизации информационных технологий (МОСИТ)**

**ОТЧЕТ ПО ИТОГОВОМУ ПРОЕКТУ**

### по дисциплине «Технология разработки программных приложений»

## Тема: «Мобильное приложение: тест - проверка знаний английского языка»

|  |  |  |
| --- | --- | --- |
| Студенты группы | *ИНБО-02-20, Горбунов Я.А.,*  *Зотов Е. А., Чернов Я. А.* | (подпись) |
| Старший преподаватель | *Жматов Д.В.* |  |
|  |  | (подпись) |
| Отчет представлен | « » 2022 г. |  |

Москва 2022 г.

**СОДЕРЖАНИЕ**

[Задачи проекта 3](#_Toc104503457)

[Цели проекта 3](#_Toc104503458)

[Информация о приложении 3](#_Toc104503459)

[Информация о проекте 8](#_Toc104503460)

[1. Публикация проекта на GitHub 8](#_Toc104503461)

[2. Настройка системы сборки Gradle 9](#_Toc104503462)

[3. Упаковка приложения с помощью Docker 11](#_Toc104503463)

[Архитектура приложения 13](#_Toc104503464)

[Листинг 15](#_Toc104503465)

[Выводы 42](#_Toc104503466)

# Задачи проекта

Приложение для проверки знаний английского языка разрабатывается для решения следующих задач:

* + - Проектирование архитектуры и интерфейса приложения;
    - Создание мобильного приложения;
    - Настройка системы сборки Gradle;
    - Развертывание на веб-сервисе GitHub;
    - Упаковка приложения с помощью Docker.

# Цели проекта

Приложение для проверки знаний английского языка разрабатывается со следующими целями:

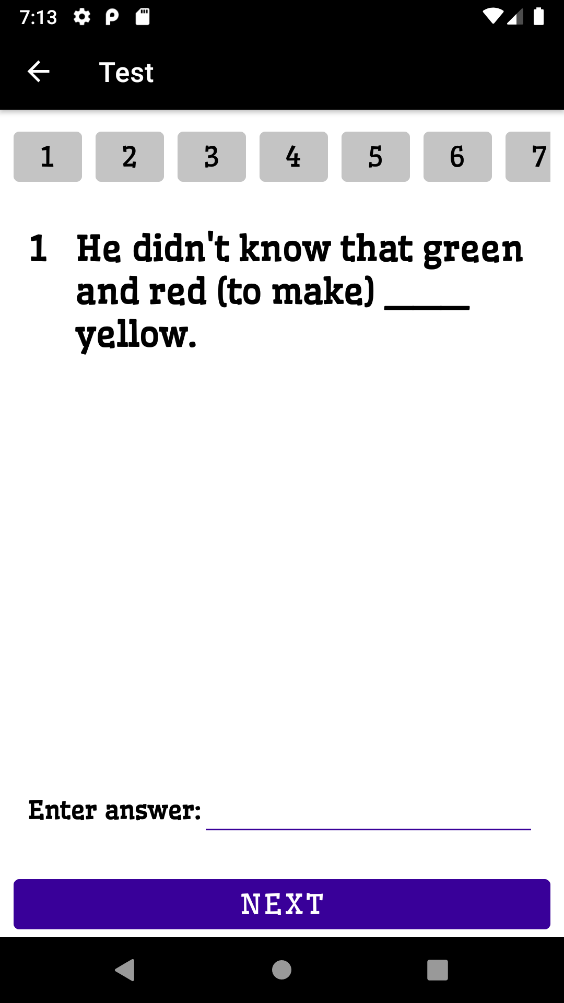
* + - Получение навыков программирования на языке Java;
    - Улучшение знаний английского языка любого пользователя;
    - Предоставление пользователю удобного и интуитивно понятного интерфейса для быстрого доступа к информации;

# Информация о приложении

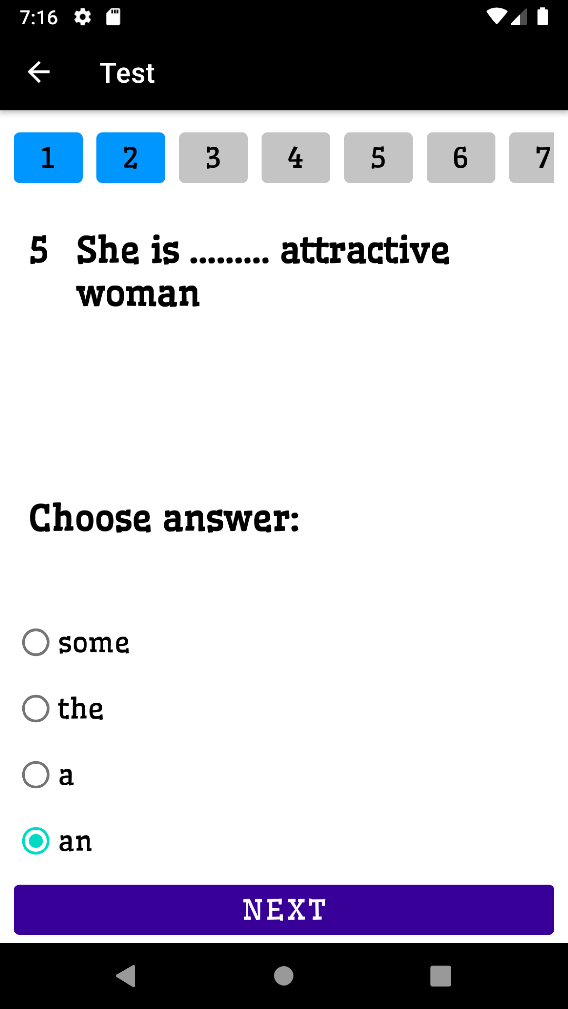
В качестве функционала системы можно выделить следующие функции:

1. Прохождение теста по английскому языку;
2. Автоматическая проверка введенных результатов;
3. Добавление собственных вопросов в базу данных приложения;

Самой главной функцией нашего приложения является прохождение теста для проверки знаний человека. В нем предлагается ответить на 10 вопросов различных категорий. Вопросы делятся на: открытые вопросы, требующие развёрнутого ответа, вопросы с возможностью выбора правильного варианта ответа из предложенных, а также вопрос-картинка, где необходимо написать слово, которое представлено на фото. Пример теста представлен на Рисунках 1-3.



**Рисунок 1 – Пример текстового вопроса**

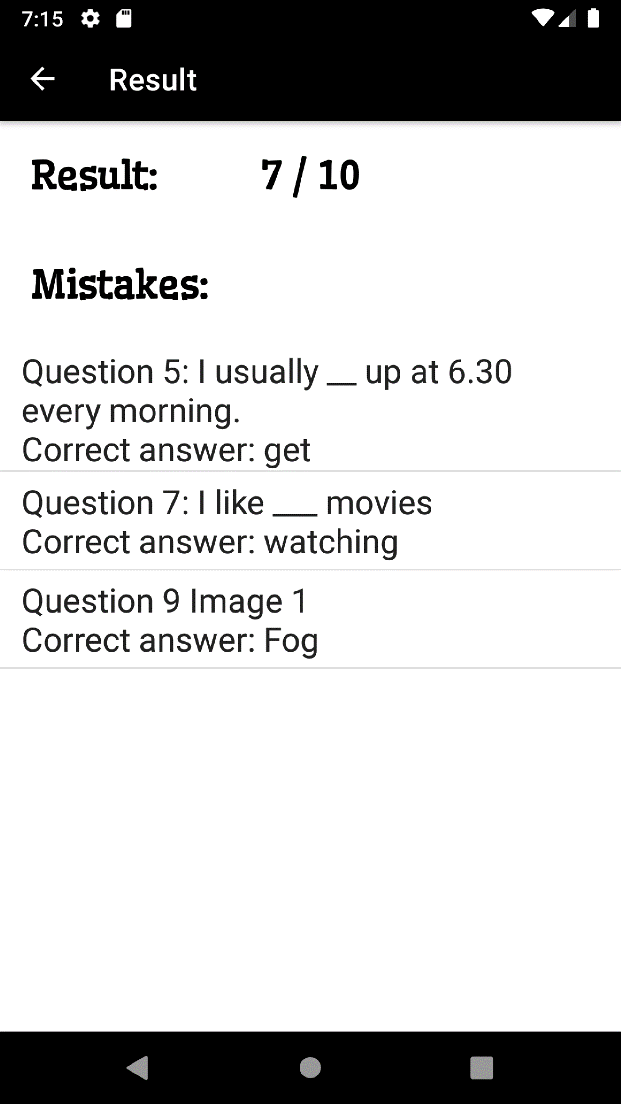


**Рисунок 2 – Пример тестового вопроса**



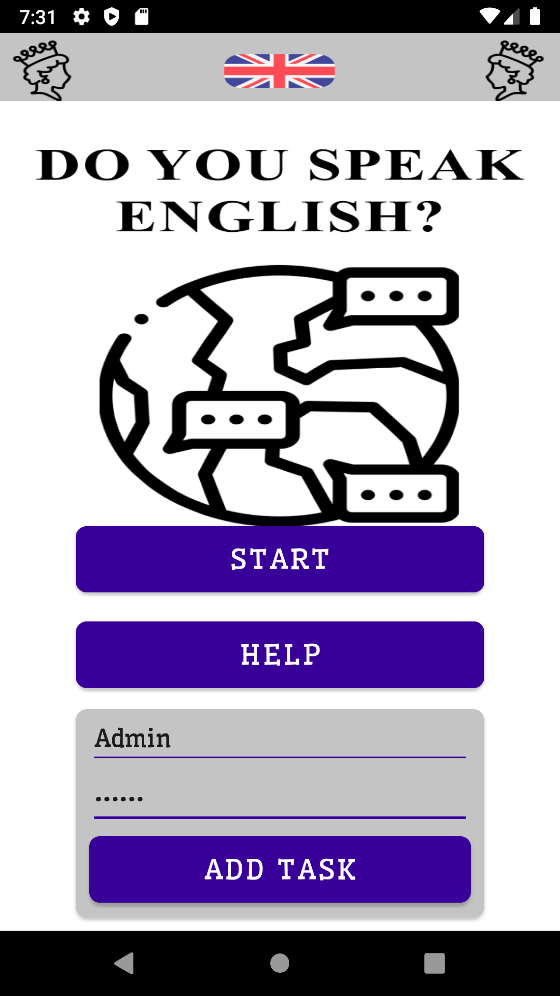
**Рисунок 3 – Пример вопроса-картинки**

Также наше приложение предусматривает функцию автоматической проверки введенных ответов пользователя. После прохождения теста на экране будет выведено количество ваших баллов, а также допущенные ошибки, если таковые присутствуют. Экран подсчета баллов представлен на Рисунке 4.

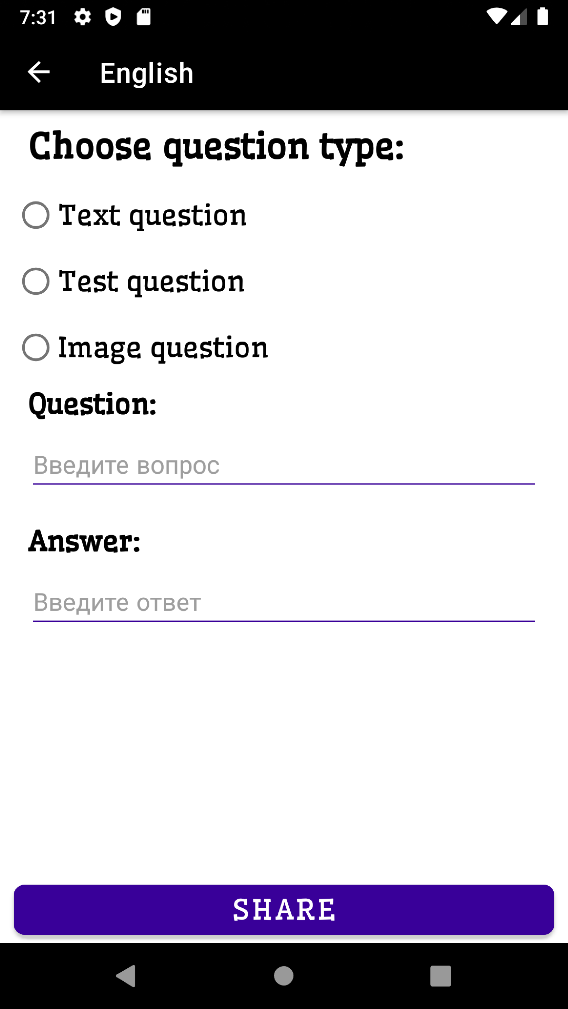


**Рисунок 4 – Экран результата**

Одной из ключевых особенностей нашего проекта в отличии от других подобных приложений является добавление вопросов пользователя в базу данных. Для этого необходимо ввести логин и пароль администратора. Пример ввода данных представлен на Рисунках 5-6.



**Рисунок 5 – Ввод данных пользователя**



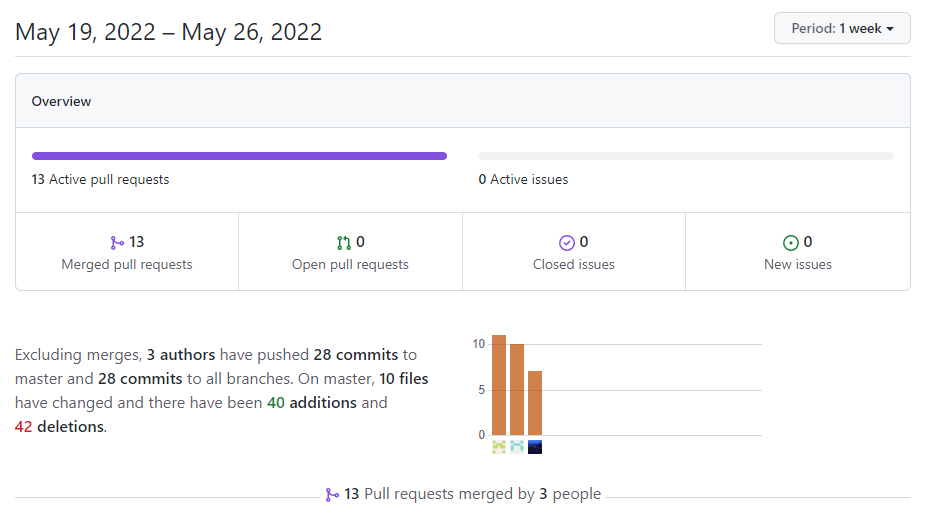
**Рисунок 6 – Создание собственного вопроса**

# Информация о проекте

1. **Публикация проекта на GitHub**

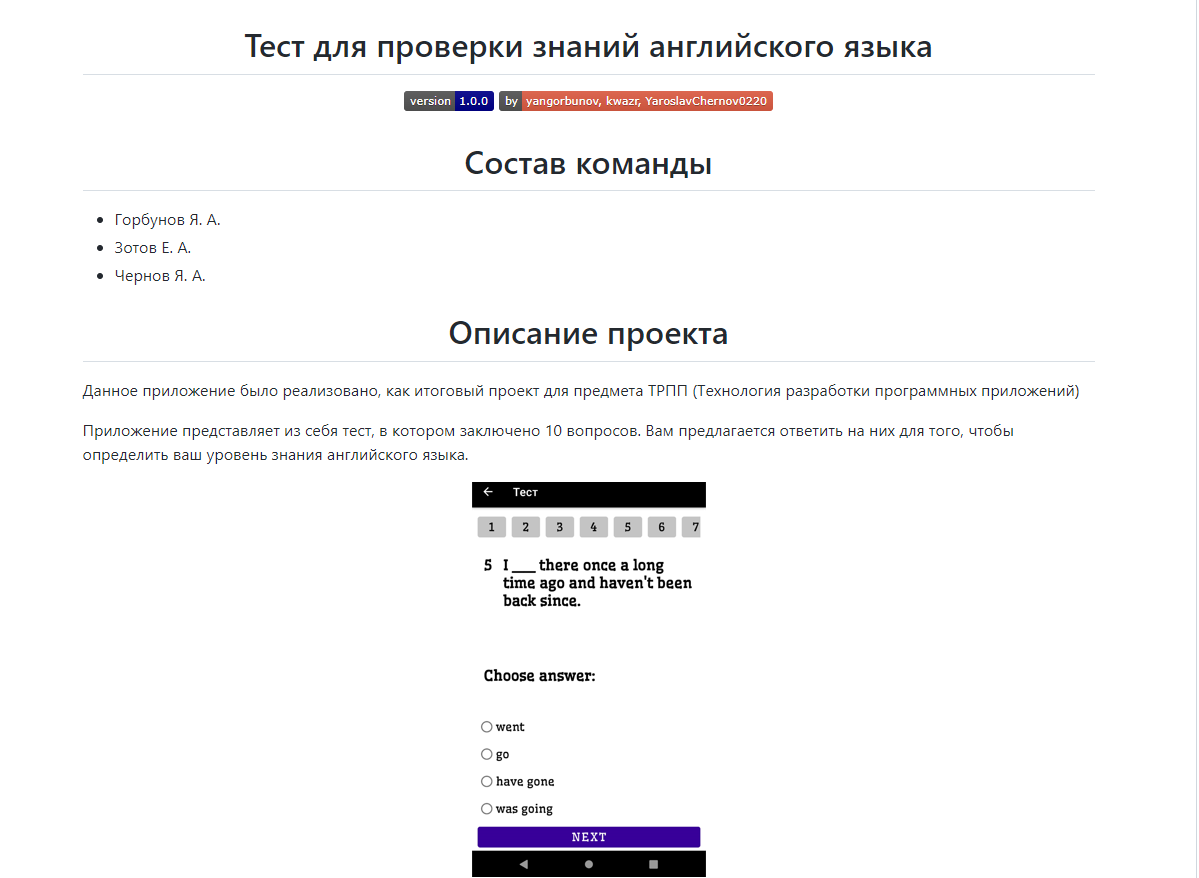
Одна из особенностей проекта заключается в том, что исходный код и все другие файлы будут доступны на веб-сервисе GitHub, использующийся для хостинга IT-проектов и их совместной разработки. Таким образом, любой человек может скачивать и дорабатывать наше мобильное приложение.

При развёртывании проекта на GitHub, были выгружены все файлы проекта. В работе проекта работала вся команда, поэтому в истории коммитов находятся все 3 разработчика (Рисунок 7).



**Рисунок 7 – работа с гитхабом**

Также был видоизменен файл README.md, в него было добавлено общее описание проекта, а также его зависимости. Файл README.md представлен на Рисунке 8.

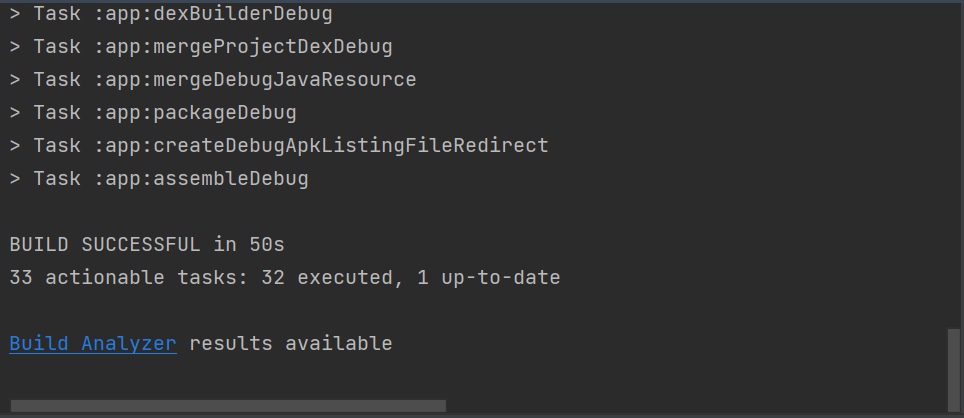
****

**Рисунок 8 – README.md файл**

1. **Настройка системы сборки Gradle**

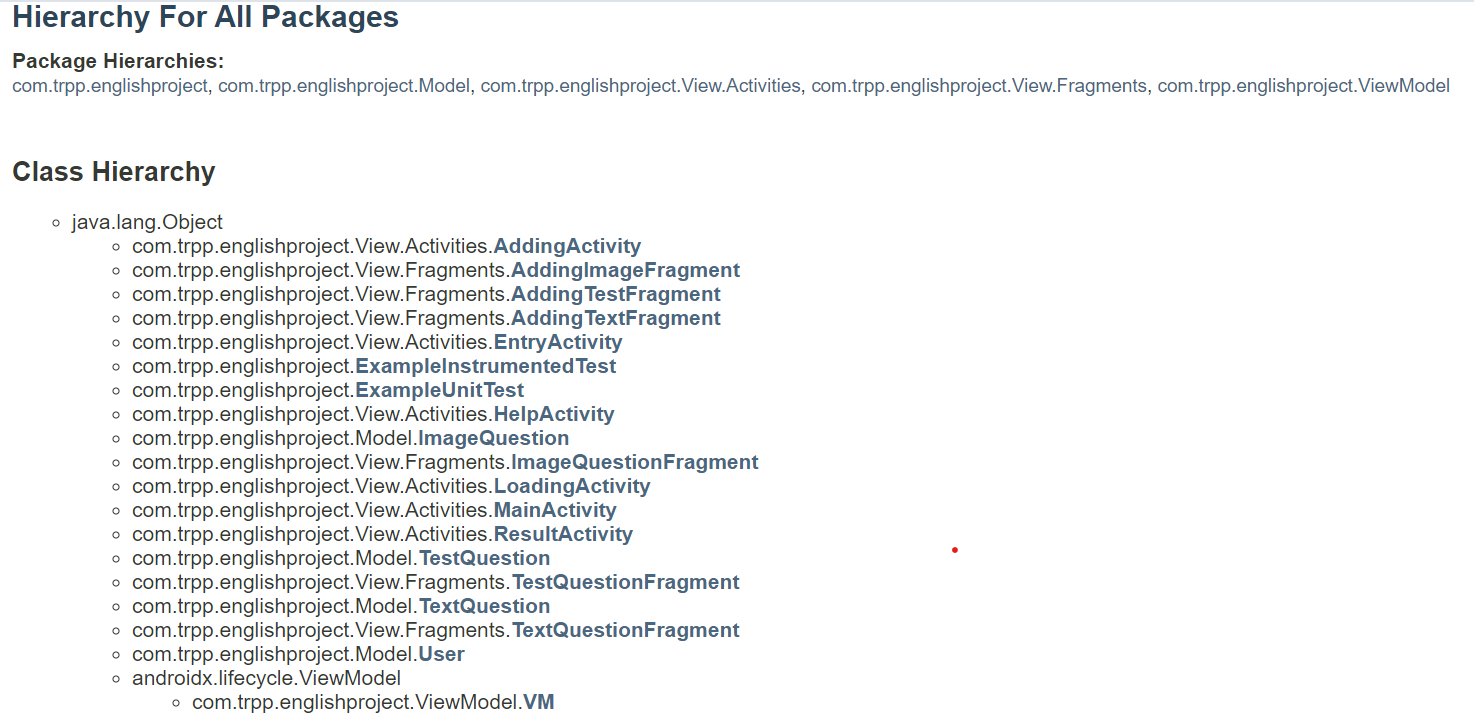
Для сборки проекта использовалась система автоматической сборки приложений и сбора статистики об использовании программных библиотек Gradle. С помощью стандартизированных средств она помогает собрать нужный продукт без потери его уникальности.

После написания кода приложения оно должно собраться с помощью системы Gradle для запуска. На Рисунке 9 представлена успешная сборка проекта.

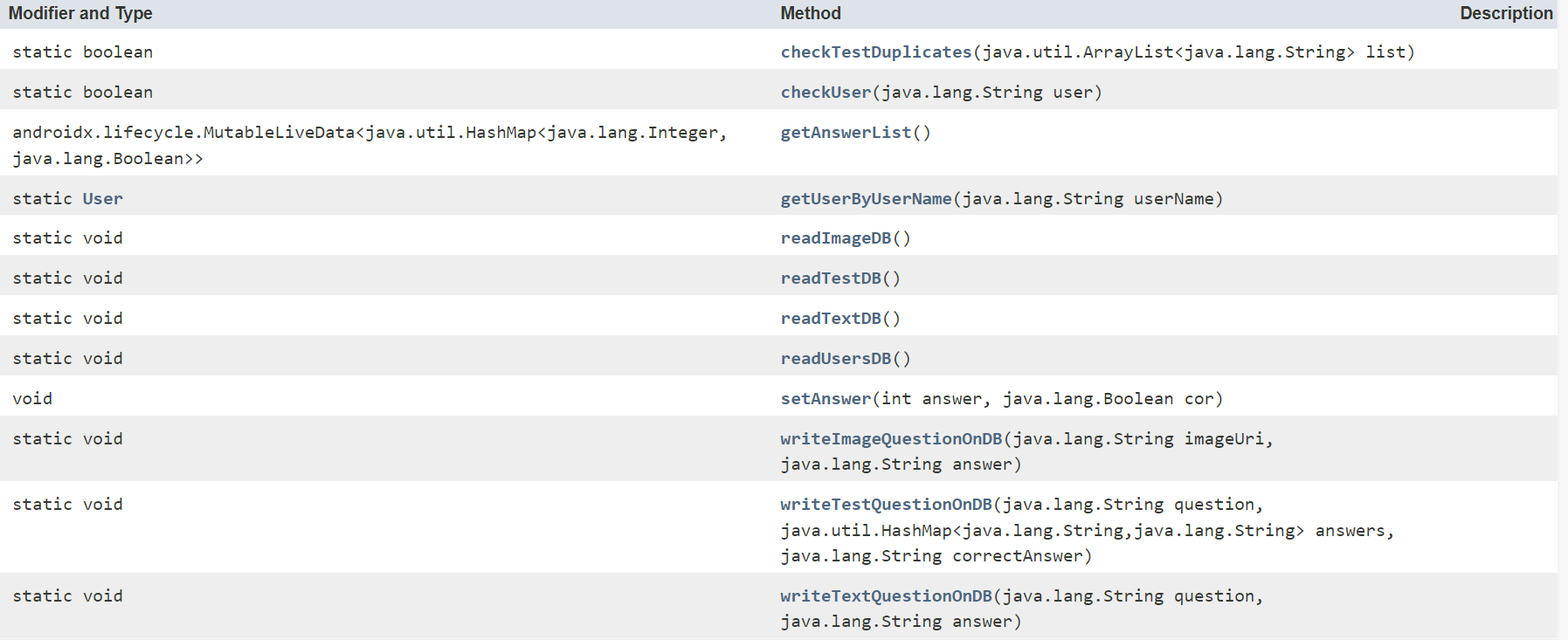


**Рисунок 9 – Успешная сборка проекта**

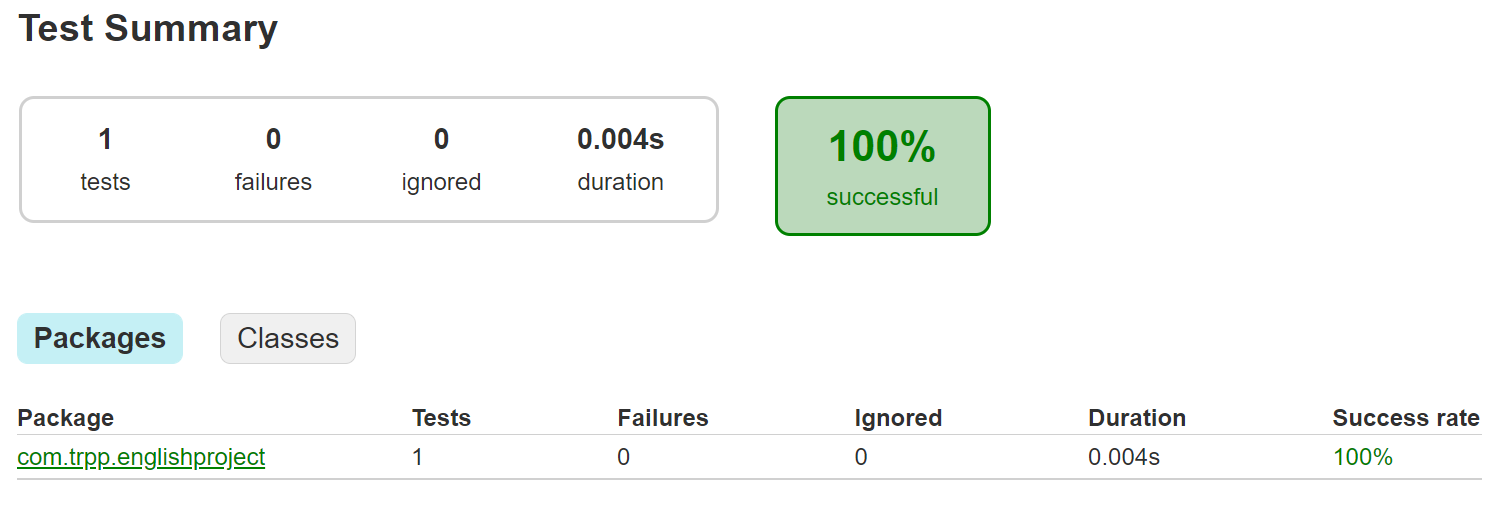
Помимо этого, Gradle использовался для сборки документации проекта. Она показана на Рисунках 10-12.



**Рисунок 10 – Документация пакетов и классов**



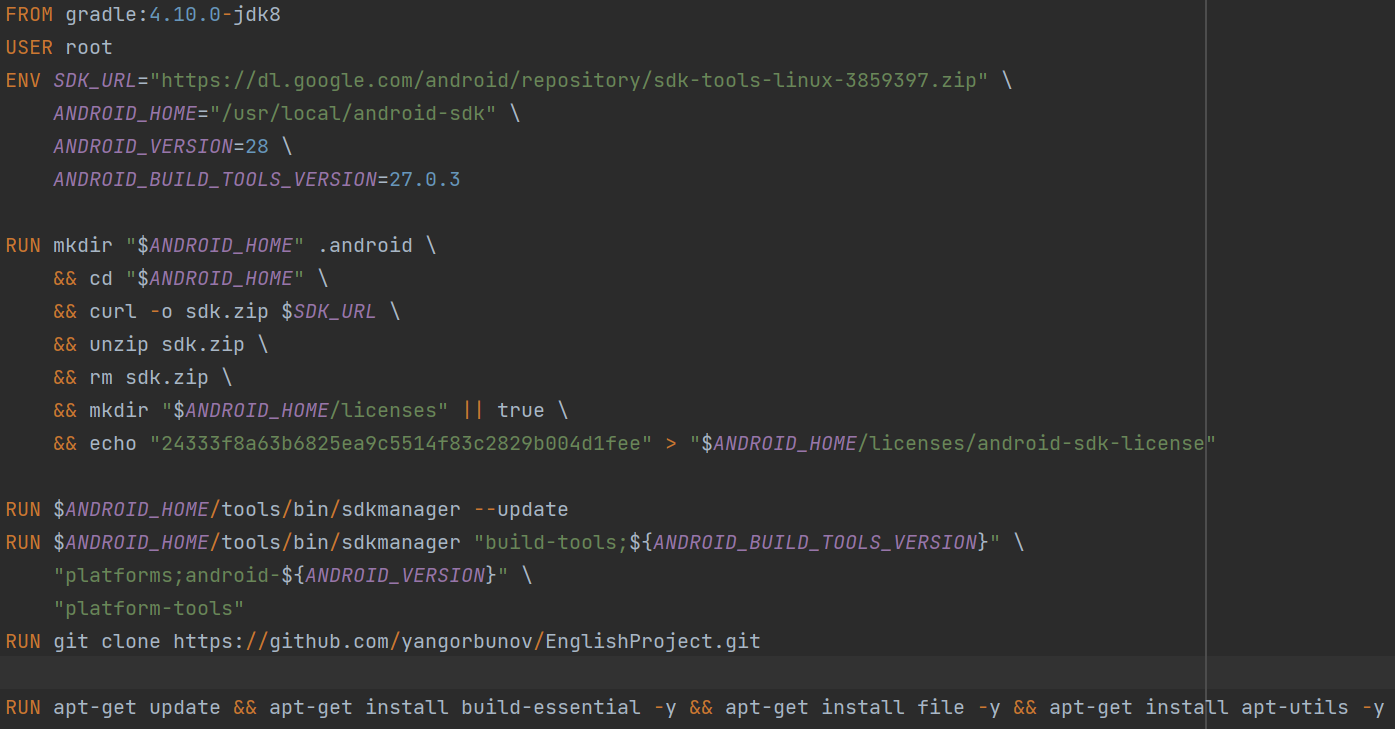
**Рисунок 11 – Резюме по методам класса ViewModel**



**Рисунок 12 – Результат теста JUnit Test**

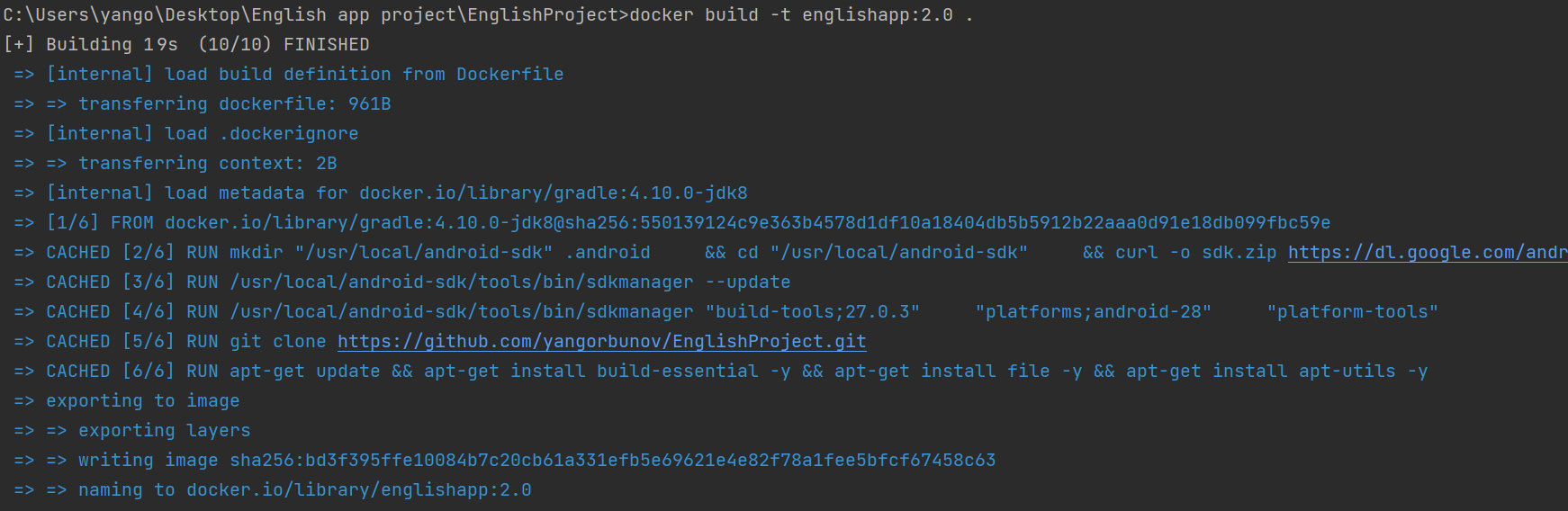
1. **Упаковка приложения с помощью Docker**

Для упаковки приложения использовалось программное обеспечение Docker, использующееся для автоматизации развёртывания и управления приложениями в средах с поддержкой контейнеризации. Чтобы упаковать наше приложение, необходимо было создать Dockerfile. В нем скачивается и распаковывается архив с sdk андроид. Далее клонируется наше приложение с GitHub в локальный репозиторий и запускается Gradle. Описание Dockerfile представлено на Рисунке 13.



**Рисунок 13 – Dockerfile**

Далее мы собираем образ нашего приложения с помощью команды *docker build –t englishapp:2.0 .* Результат сборки показан на Рисунке 14.

****

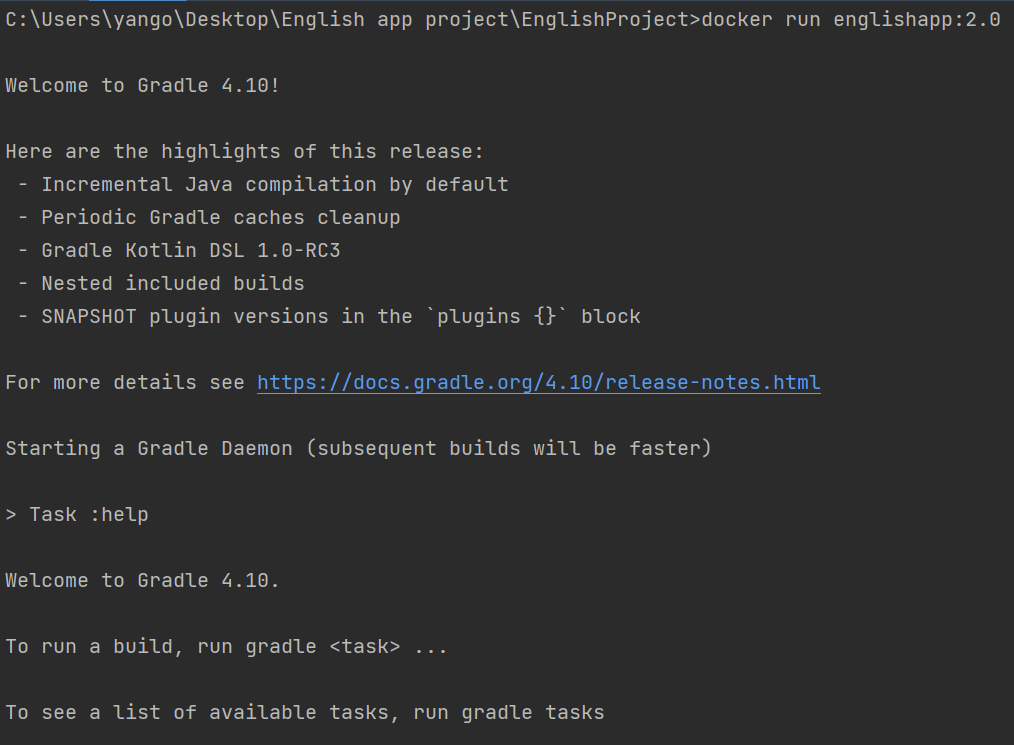
**Рисунок 14 – Создание образа приложения**

Чтобы убедиться, что наш образ успешно создался, выведем все образы с помощью команды *docker images*.Вывод всех образов представлен на Рисунке 15.



**Рисунок 15 – Вывод образов приложения**

В конце попробуем запустить наш образ с помощью команды *docker run englishapp:2.0.* Успешный запуск образа показан на Рисунке 16.

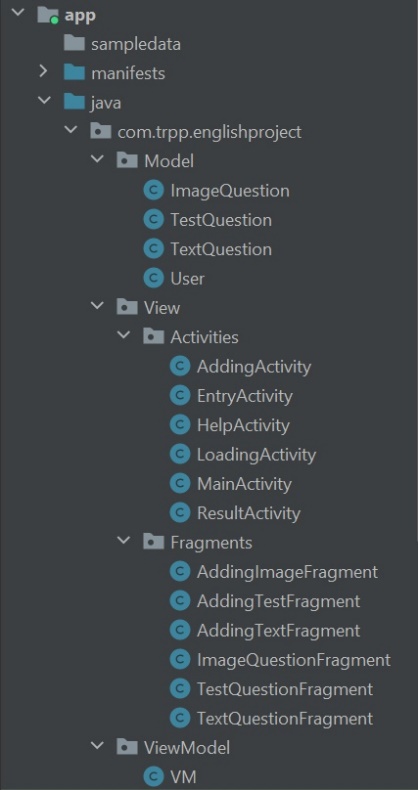


**Рисунок 16 – Запуск образа приложения**

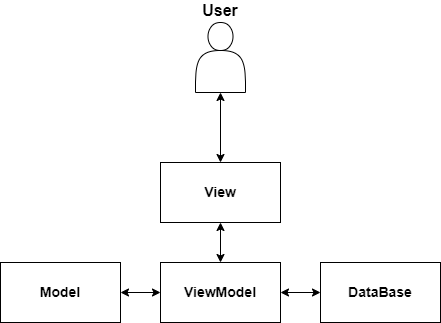
# Архитектура приложения

Model-View-ViewModel (MVVM) — шаблон проектирования архитектуры приложения, который появился для обхода ограничений паттернов MVC и MVP, и объединяющий некоторые из их сильных сторон. Ориентирован на современные платформы разработки, такие как Windows Presenatation Foundation, Silverlight от компании Microsoft. MVVM удобно использовать вместо классического MVC и ему подобных в тех случаях, когда в платформе, на которой ведётся разработка, есть «связывание данных». В таких технологиях как Silverlight есть концепция «связывания данных», позволяющая связывать данные с визуальными элементами в обе стороны. Следовательно, при использовании этого приёма применение модели MVC становится крайне неудобным из-за того, что привязка данных к представлению напрямую не укладывается в концепцию MVC/MVP.

При разработке мобильного приложения на андроид, было принято использовать архитектуру MVVM, которая была описана ранее. В приложении мы разбиваем код на пакеты, где каждый пакет отвечает за свои задачи (Рисунки 17-18).



**Рисунок 17 – MVVM модель**

****

**Рисунок 18 – Структура проекта**

В проекте мы используем облачную базу данных FireBase в реальном времени с помощью подключения к сети Интернет. База данных содержит четыре основных категории объектов, соответствующие классам модели проекта: пользователь, тестовый вопрос, текстовый вопрос и вопрос с картинкой. Данные могут быть заполнены как напрямую с сайта базы данных, так и с помощью функции добавления вопросов. Каждый объект базы данных содержит некоторое количество полей строкового типа:

* Пользователь содержит поля: имя пользователя и пароль;
* Тестовый вопрос содержит поля: вопрос, варианты ответа и правильный ответ;
* Текстовый вопрос содержит поля: вопрос и ответ;
* Вопрос-картинка содержит поля: ссылка на картинку и ответ.

# Листинг

**Листинг 1 – Файл ImageQuestion.java**

package com.trpp.englishproject.Model;

/\*\*

\* @author Yan Gorbunov

\* @version 1.0

\* Model class of questions, which contains image and answer.

\*/

public class ImageQuestion {

private String q;

private String ans;

public ImageQuestion() {

}

public ImageQuestion(String pictureUri, String cAnswer) {

this.q = pictureUri;

this.ans = cAnswer;

}

public String getQ() {

return q;

}

public void setQ(String q) {

this.q = q;

}

public String getAns() {

return ans;

}

public void setAns(String ans) {

this.ans = ans;

}

}

**Листинг 2 – Файл TestQuestion.java**

package com.trpp.englishproject.Model;

import java.util.HashMap;

/\*\*

\* @author Yan Gorbunov

\* @version 1.0

\* Model class of questions, which contains question, map of answers and correct answer.

\*/

public class TestQuestion {

private String q;

private HashMap<String,String> ans;

private String ca;

public TestQuestion() {

}

public TestQuestion(String text, HashMap<String,String> wAnswers, String cAnswer) {

this.q = text;

this.ans = wAnswers;

this.ca = cAnswer;

}

public String getQ() {

return q;

}

public void setQ(String q) {

this.q = q;

}

public HashMap<String,String> getAns() {

return ans;

}

public void setAns(HashMap<String,String> ans) {

this.ans = ans;

}

public String getCa() {

return ca;

}

public void setCa(String ca) {

this.ca = ca;

}

}

**Листинг 3 – Файл TextQuestion.java**

package com.trpp.englishproject.Model;

/\*\*

\* @author Yan Gorbunov

\* @version 1.0

\* Model class of questions, which contains question and answer.

\*/

public class TextQuestion {

String q;

String ans;

public TextQuestion() {

}

public TextQuestion(String ans, String q) {

this.q = q;

this.ans = ans;

}

public String getAns() {

return ans;

}

public void setAns(String ans) {

this.ans = ans;

}

public String getQ() {

return q;

}

public void setQ(String q) {

this.q = q;

}

}

**Листинг 4 – Файл User.java**

package com.trpp.englishproject.Model;

/\*\*

\* @author Yan Gorbunov

\* @version 1.0

\* Model class of users, which user name and password.

\*/

public class User {

String userName;

String password;

public User(String userName, String password) {

this.userName = userName;

this.password = password;

}

public User() {

}

public String getUserName() {

return userName;

}

public void setUserName(String userName) {

this.userName = userName;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

}

**Листинг 5 – Файл AddingActivity.java**

package com.trpp.englishproject.View.Activities;

import androidx.appcompat.app.AppCompatActivity;

import com.trpp.englishproject.R;

import com.trpp.englishproject.View.Fragments.AddingImageFragment;

import com.trpp.englishproject.View.Fragments.AddingTestFragment;

import com.trpp.englishproject.View.Fragments.AddingTextFragment;

import android.os.Bundle;

import android.widget.RadioButton;

import android.widget.RadioGroup;

/\*\*

\* @author Yan Gorbunov

\* @version 1.0

\* View class contains activity with radio buttons and fragment container

\*/

public class AddingActivity extends AppCompatActivity {

RadioGroup rg;

RadioButton rbText, rbTest, rbImage;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_adding);

rg = findViewById(R.id.rg\_choose);

rbText = findViewById(R.id.radio\_ch\_1);

rbTest = findViewById(R.id.radio\_ch\_2);

rbImage = findViewById(R.id.radio\_ch\_3);

getSupportFragmentManager()

.beginTransaction()

.replace(R.id.fragment\_container\_ch, AddingTextFragment.newInstance())

.commit();

rg.setOnCheckedChangeListener((radioGroup, i) -> {

if(rbText.isChecked()){

getSupportFragmentManager()

.beginTransaction()

.replace(R.id.fragment\_container\_ch, AddingTextFragment.newInstance())

.commit();

}

if (rbTest.isChecked()){

getSupportFragmentManager()

.beginTransaction()

.replace(R.id.fragment\_container\_ch, AddingTestFragment.newInstance())

.commit();

}

if (rbImage.isChecked()){

getSupportFragmentManager()

.beginTransaction()

.replace(R.id.fragment\_container\_ch, AddingImageFragment.newInstance())

.commit();

}

});

}

}

**Листинг 6 – Файл EntryActivity.java**

package com.trpp.englishproject.View.Activities;

import android.content.Intent;

import android.os.Bundle;

import android.widget.Button;

import android.widget.EditText;

import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import com.trpp.englishproject.\*;

import com.trpp.englishproject.Model.User;

import com.trpp.englishproject.ViewModel.VM;

/\*\*

\* @author Yan Gorbunov

\* @version 1.0

\* View class of questions, which contains buttons and authorization checking.

\*/

public class EntryActivity extends AppCompatActivity {

Button buttonHelp, buttonStart, buttonAdd;

EditText userNameET, passWordET;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_entry);

buttonHelp = findViewById(R.id.buttonHelp);

buttonStart = findViewById(R.id.buttonStart);

buttonAdd = findViewById(R.id.buttonAdd);

userNameET = findViewById(R.id.user\_name);

passWordET = findViewById(R.id.password);

VM.readTextDB();

VM.readTestDB();

VM.readImageDB();

VM.readUsersDB();

buttonHelp.setOnClickListener(view -> {

Intent intent = new Intent(EntryActivity.this, HelpActivity.class);

startActivity(intent);

});

buttonStart.setOnClickListener(view -> {

Intent intent = new Intent(EntryActivity.this, LoadingActivity.class);

startActivity(intent);

});

buttonAdd.setOnClickListener(view -> {

if (userNameET.getText() != null && passWordET.getText() != null) {

if (!userNameET.getText().toString().equals("") && !passWordET.getText().toString().equals("")) {

if (VM.checkUser(userNameET.getText().toString())) {

User user = VM.getUserByUserName(userNameET.getText().toString());

if (user.getPassword().equals(passWordET.getText().toString())) {

Intent intent = new Intent(EntryActivity.this, AddingActivity.class);

startActivity(intent);

} else {

Toast.makeText(this,

"Неверный пароль",

Toast.LENGTH\_SHORT).show(); }

} else {

Toast.makeText(this,

"Пользователь не найден",

Toast.LENGTH\_SHORT).show(); }

} else {

Toast.makeText(this,

"Для добавления вопросов введите логин и пароль",

Toast.LENGTH\_SHORT).show(); }

}

});

}

}

**Листинг 7 – Файл HelpActivity.java**

package com.trpp.englishproject.View.Activities;

import androidx.appcompat.app.AppCompatActivity;

import com.trpp.englishproject.\*;

import android.os.Bundle;

/\*\*

\* @author Yaroslav Chernov

\* @version 1.1

\* View class of questions, which contains image of help text.

\*/

public class HelpActivity extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_help);

}

}

**Листинг 8 – Файл LoadingActivity.java**

package com.trpp.englishproject.View.Activities;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;

import android.os.Bundle;

import android.os.Handler;

import android.widget.ProgressBar;

import com.trpp.englishproject.\*;

/\*\*

\* @author Egor Zotov

\* @version 1.1

\* Model class of questions, which contains progress bar.

\* It is made to provide time to load from database.

\*/

public class LoadingActivity extends AppCompatActivity {

ProgressBar progressBar;

private int mProgressStatus = 0;

private final Handler mHandler = new Handler();

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.fragment\_loading);

progressBar = findViewById(R.id.progressBar);

new Thread(() -> {

while (mProgressStatus < 100){

mProgressStatus++;

android.os.SystemClock.sleep(12);

mHandler.post(() -> progressBar.setProgress(mProgressStatus));

}

mHandler.post(() -> {

Intent intent = new Intent(LoadingActivity.this, MainActivity.class);

startActivity(intent);

finish();

});

}).start();

}

}

**Листинг 9 – Файл MainActivity.java**

package com.trpp.englishproject.View.Activities;

import androidx.appcompat.app.AppCompatActivity;

import androidx.lifecycle.ViewModelProvider;

import android.content.Intent;

import android.os.Bundle;

import android.widget.Button;

import android.widget.Toast;

import com.trpp.englishproject.\*;

import com.trpp.englishproject.View.Fragments.ImageQuestionFragment;

import com.trpp.englishproject.View.Fragments.TestQuestionFragment;

import com.trpp.englishproject.View.Fragments.TextQuestionFragment;

import com.trpp.englishproject.ViewModel.VM;

import java.util.Collections;

import java.util.Objects;

import java.util.Random;

/\*\*

\* @author Yan Gorbunov

\* @version 1.0

\* View class contains activity with buttons and fragment container.

\*/

public class MainActivity extends AppCompatActivity {

Button btn1, btn2, btn3, btn4, btn5, btn6, btn7, btn8, btn9, btn10, buttonSend;

int activeTaskId = 1;

VM vm;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

vm = new ViewModelProvider(this).get(VM.class);

btn1 = findViewById(R.id.button\_question\_1);

btn2 = findViewById(R.id.button\_question\_2);

btn3 = findViewById(R.id.button\_question\_3);

btn4 = findViewById(R.id.button\_question\_4);

btn5 = findViewById(R.id.button\_question\_5);

btn6 = findViewById(R.id.button\_question\_6);

btn7 = findViewById(R.id.button\_question\_7);

btn8 = findViewById(R.id.button\_question\_8);

btn9 = findViewById(R.id.button\_question\_9);

btn10 = findViewById(R.id.button\_question\_10);

buttonSend = findViewById(R.id.button\_send);

swapQuestions();

getSupportFragmentManager()

.beginTransaction()

.replace(R.id.fragment\_container, TextQuestionFragment.newInstance(1))

.commit();

buttonSend.setOnClickListener(view -> {

checkAnswers();

if (activeTaskId < 4) {

buttonSend.setText(R.string.next\_task);

goToTextQuestion();

} else if (activeTaskId < 8) {

buttonSend.setText(R.string.next\_task);

goToTestQuestion();

} else if (activeTaskId < 10) {

buttonSend.setText(R.string.next\_task);

if (activeTaskId == 9) {

buttonSend.setText(R.string.finish);

}

goToImageQuestion();

} else {

if (vm.getAnswerList().getValue() == null) {

Toast.makeText(this, "Ответьте на хотя бы один вопрос, чтобы посмотреть результат", Toast.LENGTH\_SHORT).show();

} else {

Intent intent = new Intent(MainActivity.this, ResultActivity.class);

intent.putExtra("val",checkAnswers());

intent.putExtra("list",vm.getAnswerList().getValue());

startActivity(intent);

finish();

}

}

});

btn1.setOnClickListener(view -> {

activeTaskId = 1;

buttonSend.setText(R.string.next\_task);

getSupportFragmentManager()

.beginTransaction()

.replace(R.id.fragment\_container, TextQuestionFragment.newInstance(1))

.commit();

});

btn2.setOnClickListener(view -> {

activeTaskId = 2;

buttonSend.setText(R.string.next\_task);

getSupportFragmentManager()

.beginTransaction()

.replace(R.id.fragment\_container, TextQuestionFragment.newInstance(2))

.commit();

});

btn3.setOnClickListener(view -> {

activeTaskId = 3;

buttonSend.setText(R.string.next\_task);

getSupportFragmentManager()

.beginTransaction()

.replace(R.id.fragment\_container, TextQuestionFragment.newInstance(3))

.commit();

});

btn4.setOnClickListener(view -> {

activeTaskId = 4;

buttonSend.setText(R.string.next\_task);

getSupportFragmentManager()

.beginTransaction()

.replace(R.id.fragment\_container, TextQuestionFragment.newInstance(4))

.commit();

});

btn5.setOnClickListener(view -> {

activeTaskId = 5;

buttonSend.setText(R.string.next\_task);

getSupportFragmentManager()

.beginTransaction()

.replace(R.id.fragment\_container, TestQuestionFragment.newInstance(5))

.commit();

});

btn6.setOnClickListener(view -> {

activeTaskId = 6;

buttonSend.setText(R.string.next\_task);

getSupportFragmentManager()

.beginTransaction()

.replace(R.id.fragment\_container, TestQuestionFragment.newInstance(6))

.commit();

});

btn7.setOnClickListener(view -> {

activeTaskId = 7;

buttonSend.setText(R.string.next\_task);

getSupportFragmentManager()

.beginTransaction()

.replace(R.id.fragment\_container, TestQuestionFragment.newInstance(7))

.commit();

});

btn8.setOnClickListener(view -> {

activeTaskId = 8;

buttonSend.setText(R.string.next\_task);

getSupportFragmentManager()

.beginTransaction()

.replace(R.id.fragment\_container, TestQuestionFragment.newInstance(8))

.commit();

});

btn9.setOnClickListener(view -> {

activeTaskId = 9;

buttonSend.setText(R.string.next\_task);

getSupportFragmentManager()

.beginTransaction()

.replace(R.id.fragment\_container, ImageQuestionFragment.newInstance(9))

.commit();

});

btn10.setOnClickListener(view -> {

activeTaskId = 10;

buttonSend.setText(R.string.finish);

getSupportFragmentManager()

.beginTransaction()

.replace(R.id.fragment\_container, ImageQuestionFragment.newInstance(10))

.commit();

});

}

private void goToTestQuestion() {

getSupportFragmentManager()

.beginTransaction()

.replace(R.id.fragment\_container, TestQuestionFragment.newInstance(activeTaskId + 1))

.commit();

activeTaskId += 1;

}

private void goToTextQuestion() {

getSupportFragmentManager()

.beginTransaction()

.replace(R.id.fragment\_container, TextQuestionFragment.newInstance(activeTaskId + 1))

.commit();

activeTaskId += 1;

}

private void goToImageQuestion() {

getSupportFragmentManager()

.beginTransaction()

.replace(R.id.fragment\_container, ImageQuestionFragment.newInstance(activeTaskId + 1))

.commit();

activeTaskId += 1;

}

private void swapQuestions() {

for (int i = 0; i < 10; i++){

int j = new Random().nextInt(VM.textQuestions.size());

int k = new Random().nextInt(VM.textQuestions.size());

if (j != k) {

Collections.swap(VM.textQuestions, j, k);

}

}

for (int i = 0; i < 10; i++){

int j = new Random().nextInt(VM.testQuestions.size());

int k = new Random().nextInt(VM.testQuestions.size());

if (j != k) {

Collections.swap(VM.testQuestions, j, k);

}

}

for (int i = 0; i < 10; i++){

int j = new Random().nextInt(VM.imageQuestions.size());

int k = new Random().nextInt(VM.imageQuestions.size());

if (j != k) {

Collections.swap(VM.imageQuestions, j, k);

}

}

}

private int checkAnswers(){

int val = 0;

if(vm.getAnswerList() != null && vm.getAnswerList().getValue() != null) {

if (vm.getAnswerList().getValue().containsKey(1)) {

btn1.setBackgroundColor(getResources().getColor(R.color.blue));

if (Objects.equals(vm.getAnswerList().getValue().get(1), true)) {

val += 1;

}

}

if (vm.getAnswerList().getValue().containsKey(2)) {

btn2.setBackgroundColor(getResources().getColor(R.color.blue));

if (Objects.equals(vm.getAnswerList().getValue().get(2), true)) {

val += 1;

}

}

if (vm.getAnswerList().getValue().containsKey(3)) {

btn3.setBackgroundColor(getResources().getColor(R.color.blue));

if (Objects.equals(vm.getAnswerList().getValue().get(3), true)) {

val += 1;

}

}

if (vm.getAnswerList().getValue().containsKey(4)) {

btn4.setBackgroundColor(getResources().getColor(R.color.blue));

if (Objects.equals(vm.getAnswerList().getValue().get(4), true)) {

val += 1;

}

}

if (vm.getAnswerList().getValue().containsKey(5)) {

btn5.setBackgroundColor(getResources().getColor(R.color.blue));

if (Objects.equals(vm.getAnswerList().getValue().get(5), true)) {

val += 1;

}

}

if (vm.getAnswerList().getValue().containsKey(6)) {

btn6.setBackgroundColor(getResources().getColor(R.color.blue));

if (Objects.equals(vm.getAnswerList().getValue().get(6), true)) {

val += 1;

}

}

if (vm.getAnswerList().getValue().containsKey(7)) {

btn7.setBackgroundColor(getResources().getColor(R.color.blue));

if (Objects.equals(vm.getAnswerList().getValue().get(7), true)) {

val += 1;

}

}

if (vm.getAnswerList().getValue().containsKey(8)) {

btn8.setBackgroundColor(getResources().getColor(R.color.blue));

if (Objects.equals(vm.getAnswerList().getValue().get(8), true)) {

val += 1;

}

}

if (vm.getAnswerList().getValue().containsKey(9)) {

btn9.setBackgroundColor(getResources().getColor(R.color.blue));

if (Objects.equals(vm.getAnswerList().getValue().get(9), true)) {

val += 1;

}

}

if (vm.getAnswerList().getValue().containsKey(10)) {

btn10.setBackgroundColor(getResources().getColor(R.color.blue));

if (Objects.equals(vm.getAnswerList().getValue().get(10), true)) {

val += 1;

}

}

}

return val;

}

}

**Листинг 10 – Файл ResultActivity.java**

package com.trpp.englishproject.View.Activities;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

import android.widget.ArrayAdapter;

import android.widget.ListView;

import android.widget.TextView;

import com.trpp.englishproject.ViewModel.VM;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.Objects;

import com.trpp.englishproject.R;

/\*\*

\* @author Yan Gorbunov

\* @version 1.0

\* View class contains activity with listview of mistakes and result textview.

\*/

public class ResultActivity extends AppCompatActivity {

TextView textView, mistakesTV;

ListView listView;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_result);

textView = findViewById(R.id.result\_num);

listView = findViewById(R.id.list\_view);

mistakesTV = findViewById(R.id.mistakes);

int val = getIntent().getIntExtra("val", 0);

HashMap<Integer, Boolean> resAns = (HashMap<Integer, Boolean>) getIntent().getSerializableExtra("list");

ArrayList<String> ansList = new ArrayList<>();

for (int i = 1; i < 11; i++) {

if (resAns.containsKey(i)) {

if (Objects.requireNonNull(resAns.get(i)).equals(false)) {

if (i <= 4) {

ansList.add("Question " + i + ": " + VM.textQuestions.get(i).getQ() + "\n"

+ "Correct answer: " + VM.textQuestions.get(i).getAns());

}

if (i > 4 && i <= 8) {

ansList.add("Question " + i + ": " + VM.testQuestions.get(i - 4).getQ() +

"\n" + "Correct answer: " + VM.testQuestions.get(i - 4).getCa());

}

if (i > 8) {

ansList.add("Question " + i + " Image " + (i - 8) + "\n"

+ "Correct answer: " + VM.imageQuestions.get(i - 8).getAns());

}

}

} else {

if (i <= 4) {

ansList.add("Question " + i + ": " + VM.textQuestions.get(i).getQ() + "\n"

+ "Correct answer: " + VM.textQuestions.get(i).getAns());

}

if (i > 4 && i <= 8) {

ansList.add("Question " + i + ": " + VM.testQuestions.get(i - 4).getQ() +

"\n" + "Correct answer: " + VM.testQuestions.get(i - 4).getCa());

}

if (i > 8) {

ansList.add("Question " + i + " Image " + (i - 8) + "\n"

+ "Correct answer: " + VM.imageQuestions.get(i - 8).getAns());

}

}

}

listView.setAdapter(new ArrayAdapter<>(this, android.R.layout.select\_dialog\_item, ansList));

String res = val + " / 10";

textView.setText(res);

if (val == 10){

mistakesTV.setText(R.string.no\_mistakes);

}

}

}

**Листинг 11 – Файл AddingImageFragment.java**

package com.trpp.englishproject.View.Fragments;

import android.os.Bundle;

import androidx.fragment.app.Fragment;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.Button;

import android.widget.EditText;

import android.widget.Toast;

import com.trpp.englishproject.\*;

import com.trpp.englishproject.ViewModel.VM;

/\*\*

\* @author Yaroslav Chernov

\* @version 1.0

\* View class contains fragment with button and edit text.

\*/

public class AddingImageFragment extends Fragment {

EditText question, answer;

Button shareBtn;

public AddingImageFragment() {

}

public static AddingImageFragment newInstance() {

return new AddingImageFragment();

}

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

}

@Override

public View onCreateView(LayoutInflater inflater, ViewGroup container,

Bundle savedInstanceState) {

View view = inflater.inflate(R.layout.fragment\_image\_adding, container, false);

question = view.findViewById(R.id.adding\_image\_question);

answer = view.findViewById(R.id.adding\_image\_ans);

shareBtn = view.findViewById(R.id.button\_share\_image);

shareBtn.setOnClickListener(v -> {

if (answer.getText() != null && question.getText() != null) {

if (!answer.getText().toString().equals("") && !question.getText().toString().equals("")) {

Toast.makeText(requireActivity(), "Вопрос добавлен", Toast.LENGTH\_SHORT).show();

VM.writeImageQuestionOnDB((question.getText().toString()), answer.getText().toString());

}

else {

Toast.makeText(requireActivity(), "Введите ссылку и ответ", Toast.LENGTH\_SHORT).show();

}

}

});

return view;

}

}

**Листинг 12 – Файл AddingTestFragment.java**

package com.trpp.englishproject.View.Fragments;

import android.os.Bundle;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.Button;

import android.widget.EditText;

import android.widget.RadioButton;

import android.widget.RadioGroup;

import android.widget.Toast;

import androidx.fragment.app.Fragment;

import com.trpp.englishproject.R;

import com.trpp.englishproject.ViewModel.VM;

import java.util.ArrayList;

import java.util.HashMap;

/\*\*

\* @author Yan Gorbunov

\* @version 1.0

\* View class contains fragment with buttons, edit text and radio buttons.

\*/

public class AddingTestFragment extends Fragment {

EditText question;

RadioGroup radioGroup;

RadioButton rb1, rb2, rb3, rb4;

EditText et1, et2, et3, et4;

Button shareBtn;

public AddingTestFragment() { }

public static AddingTestFragment newInstance() {

return new AddingTestFragment();

}

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

}

@Override

public View onCreateView(LayoutInflater inflater, ViewGroup container,

Bundle savedInstanceState) {

View view = inflater.inflate(R.layout.fragment\_test\_adding, container, false);

question = view.findViewById(R.id.adding\_test\_question);

radioGroup = view.findViewById(R.id.rg\_adding\_test);

et1 = view.findViewById(R.id.adding\_test\_1);

et2 = view.findViewById(R.id.adding\_test\_2);

et3 = view.findViewById(R.id.adding\_test\_3);

et4 = view.findViewById(R.id.adding\_test\_4);

rb1 = view.findViewById(R.id.radio\_1\_adding);

rb2 = view.findViewById(R.id.radio\_2\_adding);

rb3 = view.findViewById(R.id.radio\_3\_adding);

rb4 = view.findViewById(R.id.radio\_4\_adding);

shareBtn = view.findViewById(R.id.button\_share\_test);

shareBtn.setOnClickListener(v -> {

if (question.getText() != null &&

et1.getText() != null && et2.getText() != null &&

et3.getText() != null && et4.getText() != null) {

if (!question.getText().toString().equals("") &&

!et1.getText().toString().equals("") && !et2.getText().toString().equals("") &&

!et3.getText().toString().equals("") && !et4.getText().toString().equals("")) {

ArrayList<String> list = new ArrayList<>();

list.add(et1.getText().toString());

list.add(et2.getText().toString());

list.add(et3.getText().toString());

list.add(et4.getText().toString());

if (VM.checkTestDuplicates(list)) {

if (rb1.isChecked()) {

pushTask(et1.getText().toString());

} else if (rb2.isChecked()) {

pushTask(et2.getText().toString());

} else if (rb3.isChecked()) {

pushTask(et3.getText().toString());

} else if (rb4.isChecked()) {

pushTask(et4.getText().toString());

} else {

Toast.makeText(requireActivity(), "Выберите правильный вариант ответа", Toast.LENGTH\_SHORT).show();

}

} else {

Toast.makeText(requireActivity(), "Ответы не должны совпадать", Toast.LENGTH\_SHORT).show();

}

} else {

Toast.makeText(requireActivity(), "Введите вопрос и варианты ответа", Toast.LENGTH\_SHORT).show();

}

}

});

return view;

}

public void pushTask(String correctAnswer){

HashMap<String,String> map = new HashMap<>();

map.put("a1",et1.getText().toString());

map.put("a2",et2.getText().toString());

map.put("a3",et3.getText().toString());

map.put("a4",et4.getText().toString());

VM.writeTestQuestionOnDB(question.getText().toString(),map,correctAnswer);

Toast.makeText(requireActivity(), "Вопрос добавлен", Toast.LENGTH\_SHORT).show();

}

}

**Листинг 13 – Файл AddingTextFragment.java**

package com.trpp.englishproject.View.Fragments;

import android.os.Bundle;

import androidx.fragment.app.Fragment;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.Button;

import android.widget.EditText;

import android.widget.Toast;

import com.trpp.englishproject.R;

import com.trpp.englishproject.ViewModel.VM;

/\*\*

\* @author Yaroslav Chernov

\* @version 1.0

\* View class contains fragment with button and edit text.

\*/

public class AddingTextFragment extends Fragment {

EditText question, answer;

Button shareBtn;

public AddingTextFragment() {

}

public static AddingTextFragment newInstance() {

return new AddingTextFragment();

}

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

}

@Override

public View onCreateView(LayoutInflater inflater, ViewGroup container,

Bundle savedInstanceState) {

View view = inflater.inflate(R.layout.fragment\_text\_adding, container, false);

question = view.findViewById(R.id.adding\_text\_question);

answer = view.findViewById(R.id.adding\_text\_ans);

shareBtn = view.findViewById(R.id.button\_share\_text);

shareBtn.setOnClickListener(v -> {

if (answer.getText() != null && question.getText() != null) {

if (!answer.getText().toString().equals("") && !question.getText().toString().equals("")) {

Toast.makeText(requireActivity(), "Вопрос добавлен", Toast.LENGTH\_SHORT).show();

VM.writeTextQuestionOnDB(question.getText().toString(), answer.getText().toString());

} else {

Toast.makeText(requireActivity(), "Введите вопрос и ответ", Toast.LENGTH\_SHORT).show();

}

}

});

return view;

}

}

**Листинг 14 – Файл ImageQuestionFragment.java**

package com.trpp.englishproject.View.Fragments;

import android.os.Bundle;

import androidx.fragment.app.Fragment;

import androidx.lifecycle.ViewModelProvider;

import android.text.Editable;

import android.text.TextWatcher;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.EditText;

import android.widget.ImageView;

import android.widget.TextView;

import com.bumptech.glide.Glide;

import com.trpp.englishproject.Model.ImageQuestion;

import com.trpp.englishproject.ViewModel.VM;

import com.trpp.englishproject.\*;

/\*\*

\* @author Yan Gorbunov

\* @version 1.0

\* View class contains fragment with image view, text view and edit text.

\*/

public class ImageQuestionFragment extends Fragment {

private final int questionNo;

VM vm;

TextView noView;

EditText answer;

ImageView imageView;

public ImageQuestionFragment(int questionNo) {

this.questionNo = questionNo;

}

public static ImageQuestionFragment newInstance(int questionNo) {

return new ImageQuestionFragment(questionNo);

}

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

}

@Override

public View onCreateView(LayoutInflater inflater, ViewGroup container,

Bundle savedInstanceState) {

View view = inflater.inflate(R.layout.fragment\_image\_question, container, false);

noView = view.findViewById(R.id.number\_q\_img);

imageView = view.findViewById(R.id.img\_pic);

vm = new ViewModelProvider(requireActivity()).get(VM.class);

answer = view.findViewById(R.id.text\_ans\_img);

ImageQuestion imageQuestion = VM.imageQuestions.get(questionNo - 8);

noView.setText(String.valueOf(questionNo));

Glide.with(requireActivity()).load(imageQuestion.getQ()).into(imageView);

answer.addTextChangedListener(new TextWatcher() {

@Override

public void beforeTextChanged(CharSequence charSequence, int i, int i1, int i2) {

}

@Override

public void onTextChanged(CharSequence charSequence, int i, int i1, int i2) {

}

@Override

public void afterTextChanged(Editable editable) {

vm.setAnswer(questionNo, editable.toString().equalsIgnoreCase(imageQuestion.getAns()));

}

});

return view;

}

}

**Листинг 15 – Файл TestQuestionFragment.java**

package com.trpp.englishproject.View.Fragments;

import android.os.Bundle;

import androidx.fragment.app.Fragment;

import androidx.lifecycle.ViewModelProvider;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.RadioButton;

import android.widget.RadioGroup;

import android.widget.TextView;

import com.trpp.englishproject.\*;

import com.trpp.englishproject.Model.TestQuestion;

import com.trpp.englishproject.ViewModel.VM;

/\*\*

\* @author Yaroslav Chernov

\* @version 1.0

\* View class contains fragment with text view and radio buttons.

\*/

public class TestQuestionFragment extends Fragment {

private final int questionNo;

VM vm;

TextView questionTV, noQuestion;

RadioGroup radioGroup;

RadioButton rb1, rb2, rb3, rb4;

public TestQuestionFragment(int questionNo) {

this.questionNo = questionNo;

}

public static TestQuestionFragment newInstance(int questionNo) {

return new TestQuestionFragment(questionNo);

}

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

}

@Override

public View onCreateView(LayoutInflater inflater, ViewGroup container,

Bundle savedInstanceState) {

View view = inflater.inflate(R.layout.fragment\_test\_question, container, false);

questionTV = view.findViewById(R.id.QuestionTextView);

radioGroup = view.findViewById(R.id.rg);

rb1 = view.findViewById(R.id.radio\_1);

rb2 = view.findViewById(R.id.radio\_2);

rb3 = view.findViewById(R.id.radio\_3);

rb4 = view.findViewById(R.id.radio\_4);

noQuestion = view.findViewById(R.id.number\_q);

vm = new ViewModelProvider(requireActivity()).get(VM.class);

TestQuestion testQuestion = VM.testQuestions.get(questionNo - 4);

questionTV.setText(testQuestion.getQ());

noQuestion.setText(String.valueOf(questionNo));

rb1.setText(testQuestion.getAns().get("a1"));

rb2.setText(testQuestion.getAns().get("a2"));

rb3.setText(testQuestion.getAns().get("a3"));

rb4.setText(testQuestion.getAns().get("a4"));

String correctAns = testQuestion.getCa();

radioGroup.setOnCheckedChangeListener((radioGroup, i) ->

vm.setAnswer(questionNo, rb1.isChecked() && rb1.getText().equals(correctAns) ||

rb2.isChecked() && rb2.getText().equals(correctAns) ||

rb3.isChecked() && rb3.getText().equals(correctAns) ||

rb4.isChecked() && rb4.getText().equals(correctAns)));

return view;

}

}

**Листинг 16 – Файл TextQuestionFragment.java**

package com.trpp.englishproject.View.Fragments;

import android.os.Bundle;

import androidx.fragment.app.Fragment;

import androidx.lifecycle.ViewModelProvider;

import com.trpp.englishproject.\*;

import android.text.Editable;

import android.text.TextWatcher;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.EditText;

import android.widget.TextView;

import com.trpp.englishproject.Model.TextQuestion;

import com.trpp.englishproject.ViewModel.VM;

/\*\*

\* @author Yaroslav Chernov

\* @version 1.0

\* View class contains fragment with text view and edit text.

\*/

public class TextQuestionFragment extends Fragment {

private final int questionNo;

TextView noView, questionView;

EditText answer;

VM vm;

public TextQuestionFragment(int questionNo) {

this.questionNo = questionNo;

}

public static TextQuestionFragment newInstance(int questionNo) {

return new TextQuestionFragment(questionNo);

}

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

}

@Override

public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {

View view = inflater.inflate(R.layout.fragment\_text\_question, container, false);

noView = view.findViewById(R.id.number\_q);

questionView = view.findViewById(R.id.questionTextView);

vm = new ViewModelProvider(requireActivity()).get(VM.class);

answer = view.findViewById(R.id.text\_ans);

TextQuestion textQuestion = VM.textQuestions.get(questionNo);

noView.setText(String.valueOf(questionNo));

questionView.setText(textQuestion.getQ());

answer.addTextChangedListener(new TextWatcher() {

@Override

public void beforeTextChanged(CharSequence charSequence, int i, int i1, int i2) {

}

@Override

public void onTextChanged(CharSequence charSequence, int i, int i1, int i2) {

}

@Override

public void afterTextChanged(Editable editable) {

vm.setAnswer(questionNo, editable.toString().equalsIgnoreCase(textQuestion.getAns()));

}

});

return view;

}

}

**Листинг 17 – Файл VM.java**

package com.trpp.englishproject.ViewModel;

import androidx.annotation.NonNull;

import androidx.lifecycle.MutableLiveData;

import androidx.lifecycle.ViewModel;

import com.google.firebase.database.DataSnapshot;

import com.google.firebase.database.DatabaseError;

import com.google.firebase.database.DatabaseReference;

import com.google.firebase.database.FirebaseDatabase;

import com.google.firebase.database.ValueEventListener;

import com.trpp.englishproject.Model.ImageQuestion;

import com.trpp.englishproject.Model.TestQuestion;

import com.trpp.englishproject.Model.TextQuestion;

import com.trpp.englishproject.Model.User;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.HashSet;

import java.util.Objects;

import java.util.stream.Collectors;

import java.util.stream.Stream;

/\*\*

\* @author Yan Gorbunov

\* @version 1.0

\* ViewModel class, which contains static lists ans live data hash map with test results.

\*/

public class VM extends ViewModel {

MutableLiveData<HashMap<Integer,Boolean>> answerData;

public static ArrayList<TextQuestion> textQuestions = new ArrayList<>();

public static ArrayList<TestQuestion> testQuestions = new ArrayList<>();

public static ArrayList<ImageQuestion> imageQuestions = new ArrayList<>();

public static ArrayList<User> usersList = new ArrayList<>();

static FirebaseDatabase firebaseDatabase = FirebaseDatabase.getInstance();

/\*\*

\*

\* Getter method for live data

\* @return answerData - value of live data

\*/

public MutableLiveData<HashMap<Integer,Boolean>> getAnswerList() {

if (answerData == null){

answerData = new MutableLiveData<>();

}

return answerData;

}

/\*\*

\* Writes added text question to database

\* @param answer - users answer for question

\* @param cor - boolean value if answer is correct

\*/

public void setAnswer(int answer, Boolean cor) {

if (answerData == null){

answerData = new MutableLiveData<>();

}

if (answerData.getValue() == null){

HashMap<Integer,Boolean> map = new HashMap<>();

map.put(answer,cor);

answerData.setValue(map);

}

else {

HashMap<Integer,Boolean> list = new HashMap<>(

Objects.requireNonNull(getAnswerList().getValue()));

list.put(answer,cor);

answerData.setValue(list);

}

}

/\*\*

\* Writes added text question to database

\* @param question - added question

\* @param answer - added answer

\*/

public static void writeTextQuestionOnDB (String question, String answer){

TextQuestion textQuestion = new TextQuestion();

textQuestion.setQ(question);

textQuestion.setAns(answer);

DatabaseReference itemsRef = firebaseDatabase.getReference().child("Text");

DatabaseReference newTextQRef = itemsRef.push();

newTextQRef.setValue(textQuestion);

}

/\*\*

\* Writes added test question to database

\* @param question - added question

\* @param answers - added map of answers

\* @param correctAnswer - added correct answer

\*/

public static void writeTestQuestionOnDB (String question, HashMap<String,String> answers, String correctAnswer){

TestQuestion testQuestion = new TestQuestion();

testQuestion.setQ(question);

testQuestion.setAns(answers);

testQuestion.setCa(correctAnswer);

DatabaseReference itemsRef = firebaseDatabase.getReference().child("Test");

DatabaseReference newTextQRef = itemsRef.push();

newTextQRef.setValue(testQuestion);

}

/\*\*

\* Writes added image question to database

\* @param imageUri - added image URL

\* @param answer - added correct answer

\*/

public static void writeImageQuestionOnDB (String imageUri, String answer){

ImageQuestion imageQuestion = new ImageQuestion();

imageQuestion.setQ(imageUri);

imageQuestion.setAns(answer);

DatabaseReference itemsRef = firebaseDatabase.getReference().child("Picture");

DatabaseReference newTextQRef = itemsRef.push();

newTextQRef.setValue(imageQuestion);

}

/\*\*

\* Finds and returns user by userName if exists

\* @param userName - value if userName field

\* @return user - User class object

\*/

public static User getUserByUserName(String userName){

User user = new User();

for (User u : usersList){

if(u.getUserName().equals(userName)){

user = u;

}

}

return user;

}

/\*\*

\* Finds and returns true if user exists

\* @param user - User class object

\* @return ifCorrect - Boolean method result

\*/

public static boolean checkUser (String user){

boolean ifCorrect = false;

for (User u : usersList){

if (u.getUserName().equals(user)) {

ifCorrect = true;

break;

}

}

return ifCorrect;

}

/\*\*

\* Finds and returns true is answers do not duplicate

\* @param list - list of answers

\* @return ifCorrect - Boolean method result

\*/

public static boolean checkTestDuplicates(ArrayList<String> list){

HashSet<String> set = new HashSet<>(list);

return list.size() == set.size();

}

/\*\*

\* Reads users from database

\*/

public static void readUsersDB(){

DatabaseReference itemsRef = firebaseDatabase.getReference().child("Users");

itemsRef.addValueEventListener(new ValueEventListener() {

@Override

public void onDataChange(@NonNull DataSnapshot snapshot) {

if(!usersList.isEmpty()){

usersList.clear();

}

for(DataSnapshot ds : snapshot.getChildren()) {

User u = ds.getValue(User.class);

if (u != null) {

usersList.add(u);

}

}

}

@Override

public void onCancelled(@NonNull DatabaseError error) {

}

});

}

/\*\*

\* Reads text questions from database

\*/

public static void readTextDB(){

DatabaseReference itemsRef = firebaseDatabase.getReference().child("Text");

itemsRef.addValueEventListener(new ValueEventListener() {

@Override

public void onDataChange(@NonNull DataSnapshot snapshot) {

if(!textQuestions.isEmpty()){

testQuestions.clear();

}

for(DataSnapshot ds : snapshot.getChildren()) {

TextQuestion o = ds.getValue(TextQuestion.class);

if (o != null) {

textQuestions.add(o);

}

}

}

@Override

public void onCancelled(@NonNull DatabaseError error) {

}

});

}

/\*\*

\* Reads test questions from database

\*/

public static void readTestDB(){

DatabaseReference itemsRef = firebaseDatabase.getReference().child("Test");

itemsRef.addValueEventListener(new ValueEventListener() {

@Override

public void onDataChange(@NonNull DataSnapshot snapshot) {

if(!testQuestions.isEmpty()){

testQuestions.clear();

}

for(DataSnapshot ds : snapshot.getChildren()) {

TestQuestion o = ds.getValue(TestQuestion.class);

if (o != null) {

testQuestions.add(o);

}

}

}

@Override

public void onCancelled(@NonNull DatabaseError error) {

}

});

}

/\*\*

\* Reads image questions from database

\*/

public static void readImageDB(){

DatabaseReference itemsRef = firebaseDatabase.getReference().child("Picture");

itemsRef.addValueEventListener(new ValueEventListener() {

@Override

public void onDataChange(@NonNull DataSnapshot snapshot) {

if (!imageQuestions.isEmpty()){

imageQuestions.clear();

}

for(DataSnapshot ds : snapshot.getChildren()) {

ImageQuestion o = ds.getValue(ImageQuestion.class);

if (o != null) {

imageQuestions.add(o);

}

}

}

@Override

public void onCancelled(@NonNull DatabaseError error) {

}

});

}

}

**Ссылка на GitHub:** [**https://github.com/yangorbunov/EnglishProject**](https://github.com/yangorbunov/EnglishProject)

# Выводы

Реализуя данный проект, мы создали мобильное приложение, которое позволяет пользователю проверить свои знания английского языка.

По ходу выполнения работы, мы воспользовались системой контроля версий GitHub, настроили автоматическую сборку проекта с помощью системы Gradle, а также запаковали наше приложение в Dockerfile.

Таким образом, по итогу работы можно сделать вывод, что все поставленные задачи были выполнены, а цели достигнуты.