ПРИЛОЖЕНИЕ Ж. КОД ПРОГРАММЫ

АННОТАЦИЯ

В данном разделе представлен код программы «Развивашка», который содержит 22 модуль для образования логики работы задуманной на этапе проектирования.

СОДЕРЖАНИЕ

[1. ТЕКСТ ПРОГРАММЫ 4](#_Toc159409168)

[1.1.Наименование программы 4](#_Toc159409169)

[1.2 Область применения 4](#_Toc159409170)

[1.3 Модули 4](#_Toc159409171)

[1.4 Код модуля MainWindow 5](#_Toc159409172)

[1.6 Код модуля prosm1 5](#_Toc159409173)

[1.7 Код модуля tovar 5](#_Toc159409174)

[1.8 Код модуля Client 5](#_Toc159409175)

[1.10 Код модуля Window1 5](#_Toc159409176)

[1.11 Код модуля Admin 5](#_Toc159409177)

[1.12 Код модуля dob\_mag 5](#_Toc159409178)

[1.13 Код модуля izm\_mag 5](#_Toc159409179)

[1.14 Код модуля vivod\_mag 5](#_Toc159409180)

[1.15 Код модуля dobav 5](#_Toc159409181)

[1.16 Код модуля izmen 5](#_Toc159409182)

[1.17 Код модуля ydal 5](#_Toc159409183)

[1.18 Код модуля del 5](#_Toc159409184)

[1.19 Код модуля dobav 5](#_Toc159409185)

[1.20 Код модуля polsov 5](#_Toc159409186)

[1.22 Код модуля updat 5](#_Toc159409187)

[1.23 Код модуля d1 5](#_Toc159409188)

[1.24 Код модуля I2 5](#_Toc159409189)

[1.25 Код модуля I3 5](#_Toc159409190)

[1.26 Код модуля t1 5](#_Toc159409191)

[1.27 Код модуля Prodavec 5](#_Toc159409192)

[1.28 Код модуля dob1 5](#_Toc159409193)

[1.29 Код модуля ydal1 5](#_Toc159409194)

[2 Скрипт БД 5](#_Toc159409195)

# 1. ТЕКСТ ПРОГРАММЫ

## 1.1. Наименование программы

Наименование «Развивашка».

## 1.2 Область применения

Программа предназначена для равзыития детей путём прохождения различных тестов.

## 1.3 Модули

|  |  |  |  |
| --- | --- | --- | --- |
| Модуль | Описание | Количество строк кода | Размер в (Кбайтах) |
| 1 | 2 | 3 | 4 |
| Admin\_cabinet | Личная страница для администратора | 25 | 1 |
| Admin\_dobavlenie\_dannix | Страница для добавления и удаления предмета и темы | 259 | 12 |
| Change\_the\_test | Выбор теста для ученика | 204 | 9 |
| Dobavlenie | Первоначальная страница для добавления темы и предмета, позже заменена страницей Admin\_cabinet | 80 | 4 |
| MainActivity | Страница для авторизации и регистрации пользователей | 262 | 15 |
| Menedger\_cabinet | Личная страница менеджера | 55 | 2 |
| Meneger\_create\_a\_tests | Страница для менеджер что бы можно было создать тест | 333 | 17 |
| Meneger\_qvestion\_complence | Страница для добавления вопроса с соответствием | 175 | 9 |
| prohogdenie\_Test | Страница для ученика для прохождения теста | 369 | 21 |
| Quvestion\_meneger | Страница для создания вопроса с 1 правильным и 3 неправильными ответами | 71 | 9 |
| Qvestion\_vvod | Страница для создания вопроса с рукописным вводом | 149 | 7 |
| RecyclerItemClickListener | Обработка вывода списка | 57 | 2 |
| Result\_ycheniki | Страница для отображения рузультатов тестов для учеников | 136 | 5 |
| Result\_ychitel | Страница для отображения рузультатов тестов учеников для учителя | 73 | 3 |
| ychenik\_cabinet | Личная страница для ученика | 39 | 2 |
| ycitel\_cabinet | Личная страница для учителя | 24 | 1 |

## 1.4 Код модуля Admin\_cabinet

package com.example.razvivaska\_plahova;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

public class Admin\_cabinet extends AppCompatActivity {

private Button Per;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_admin\_cabinet);

Per = findViewById(R.id.Perehod);

Per.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

startActivity(new Intent(Admin\_cabinet.this, Admin\_dobavlenie\_dannix.class));

}

});

}

}}

## 1.6 Код модуля Admin\_dobavlenie\_dannix

package com.example.razvivaska\_plahova;

import static android.content.ContentValues.TAG;

import androidx.annotation.NonNull;

import androidx.appcompat.app.AlertDialog;

import androidx.appcompat.app.AppCompatActivity;

import android.annotation.SuppressLint;

import android.content.DialogInterface;

import android.os.Bundle;

import android.util.Log;

import android.view.View;

import android.view.ViewGroup;

import android.widget.AdapterView;

import android.widget.ArrayAdapter;

import android.widget.Button;

import android.widget.EditText;

import android.widget.Spinner;

import android.widget.TextView;

import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;

import com.google.android.gms.tasks.OnFailureListener;

import com.google.android.gms.tasks.OnSuccessListener;

import com.google.android.gms.tasks.Task;

import com.google.firebase.firestore.DocumentReference;

import com.google.firebase.firestore.DocumentSnapshot;

import com.google.firebase.firestore.FieldValue;

import com.google.firebase.firestore.FirebaseFirestore;

import com.google.firebase.firestore.QueryDocumentSnapshot;

import com.google.firebase.firestore.QuerySnapshot;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

public class Admin\_dobavlenie\_dannix extends AppCompatActivity {

private Spinner spinnerSubject;

private Spinner spinnerTheme;

private EditText editTextQuestionTitle;

private EditText editTextQuestionText;

private EditText editTextAnswer1;

private EditText editTextAnswer2;

private EditText editTextAnswer3;

private EditText editTextAnswer4;

private EditText editTextSubjectName;

private EditText editTextThemeName;

private Button buttonCreateQuestion;

private Button buttonAddSubject,button\_updateSubject,button\_updateTheme,button\_deleteSubject,button\_deleteTheme;

private Button buttonAddTheme;

private ArrayAdapter<Subject> adapter;

private FirebaseFirestore db;

private ArrayList<Subject> subjects;

private ArrayList<String> themes;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_admin\_dobavlenie\_dannix);

db = FirebaseFirestore.getInstance();

spinnerSubject = findViewById(R.id.spinner\_subject);

spinnerTheme = findViewById(R.id.spinner\_theme);

editTextSubjectName = findViewById(R.id.editTextSubjectName);

buttonAddSubject = findViewById(R.id.button\_add\_subject);

buttonAddTheme = findViewById(R.id.button\_add\_theme);

button\_deleteSubject = findViewById(R.id.button\_deleteSubject);

button\_deleteTheme = findViewById(R.id.button\_deleteTheme);

subjects = new ArrayList<>();

themes = new ArrayList<>();

adapter = new ArrayAdapter<Subject>(this, android.R.layout.simple\_spinner\_item, subjects) {

@Override

public View getView(int position, View convertView, ViewGroup parent) {

View view = super.getView(position, convertView, parent);

TextView textView = (TextView) view.findViewById(android.R.id.text1);

textView.setText(((Subject) getItem(position)).getName());

return view;

}

};

spinnerSubject.setAdapter(adapter);

buttonAddSubject.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

addSubject();

}

});

button\_deleteSubject.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

deleteSubject();

}

});

button\_deleteTheme.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

deleteTheme();

}

});

buttonAddTheme.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

addTheme();

}

});

loadSubjects(); // Call loadSubjects() at the end

spinnerSubject.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {

@Override

public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {

Subject selectedSubject = (Subject) spinnerSubject.getSelectedItem();

themes.clear();

List<String> themesList = selectedSubject.getThemes();

if (themesList != null && !themesList.isEmpty()) {

themes.addAll(themesList);

ArrayAdapter<String> adapter = new ArrayAdapter<>(Admin\_dobavlenie\_dannix.this, android.R.layout.simple\_spinner\_item, themes);

adapter.setDropDownViewResource(android.R.layout.simple\_spinner\_dropdown\_item);

spinnerTheme.setAdapter(adapter);

} else {

// Handle the case when there are no themes

spinnerTheme.setAdapter(null); // or some other default adapter

}

}

@Override

public void onNothingSelected(AdapterView<?> parent) {

// Do nothing

}

});

}

private void loadSubjects() {

db.collection("subjects")

.get()

.addOnCompleteListener(new OnCompleteListener<QuerySnapshot>() {

@Override

public void onComplete(@NonNull Task<QuerySnapshot> task) {

if (task.isSuccessful()) {

subjects.clear(); // Clear the list before adding new data

for (QueryDocumentSnapshot document : task.getResult()) {

Subject subject = document.toObject(Subject.class);

subjects.add(subject);

}

adapter.notifyDataSetChanged(); // Notify the adapter of data changes

} else {

Log.d(TAG, "Error getting subjects: ", task.getException());

}

}

});

}

private void addSubject() {

String subjectName = editTextSubjectName.getText().toString();

Subject subject = new Subject(subjectName);

db.collection("subjects").document(subjectName).set(new HashMap<String, String>() {{

put("name", subjectName);

}});

subjects.add(subject);

adapter.notifyDataSetChanged(); // Now this should work

spinnerSubject.setAdapter(adapter);

}

private void deleteSubject() {

String subjectName = editTextSubjectName.getText().toString();

db.collection("subjects").document(subjectName).delete()

.addOnSuccessListener(new OnSuccessListener<Void>() {

@Override

public void onSuccess(Void aVoid) {

// Обновляем список subjects после удаления документа

// Пример, предполагая, что subjects - это ArrayList

subjects.remove(subjectName);

Toast.makeText(Admin\_dobavlenie\_dannix.this, "Предмет успешно удалён", Toast.LENGTH\_SHORT).show();

adapter.notifyDataSetChanged();

}

})

.addOnFailureListener(new OnFailureListener() {

@Override

public void onFailure(@NonNull Exception e) {

Toast.makeText(Admin\_dobavlenie\_dannix.this, "Ошибка при удаление предмета: " + e.getMessage(), Toast.LENGTH\_SHORT).show();

}

});

}

private void addTheme() {

// Создаем всплывающее окно с полем ввода для имени темы

AlertDialog.Builder builder = new AlertDialog.Builder(Admin\_dobavlenie\_dannix.this);

builder.setTitle("Новая тема");

// Создаем поле ввода для имени темы

final EditText input = new EditText(Admin\_dobavlenie\_dannix.this);

builder.setView(input);

// Устанавливаем кнопки "ОК" и "Отмена"

builder.setPositiveButton("ОК", new DialogInterface.OnClickListener() {

public void onClick(DialogInterface dialog, int id) {

String themeName = input.getText().toString();

Subject selectedSubject = (Subject) spinnerSubject.getSelectedItem();

// Проверяем, существует ли тема в списке тем предмета

if (selectedSubject.getThemes().contains(themeName)) {

// Если тема уже есть, выводим сообщение об ошибке

Toast.makeText(Admin\_dobavlenie\_dannix.this, "Тема уже существует!", Toast.LENGTH\_SHORT).show();

} else {

// Если темы нет, добавляем ее в список тем предмета

selectedSubject.addTheme(themeName);

// Обновляем список тем в Firestore

db.collection("subjects").document(selectedSubject.getName()).update("themes", FieldValue.arrayUnion(themeName));

// Обновляем список тем в адаптере

themes.clear();

themes.addAll(selectedSubject.getThemes());

ArrayAdapter<String> adapter = new ArrayAdapter<>(Admin\_dobavlenie\_dannix.this, android.R.layout.simple\_spinner\_item, themes);

adapter.setDropDownViewResource(android.R.layout.simple\_spinner\_dropdown\_item);

spinnerTheme.setAdapter(adapter);

}

}

});

builder.setNegativeButton("Отмена", new DialogInterface.OnClickListener() {

public void onClick(DialogInterface dialog, int id) {

dialog.cancel();

}

});

// Создаем и показываем диалоговое окно

AlertDialog dialog = builder.create();

dialog.show();

}

private void deleteTheme() {

Subject selectedSubject = (Subject) spinnerSubject.getSelectedItem();

String themeName = (String) spinnerTheme.getSelectedItem();

List<String> themes = selectedSubject.getThemes();

// Проверяем, что тема действительно существует в списке

if (themes.contains(themeName)) {

themes.remove(themeName); // Удаляем тему по имени

// Обновляем данные

try {

db.collection("subjects").document(selectedSubject.getName()).update("themes", themes)

.addOnSuccessListener(new OnSuccessListener<Void>() {

@Override

public void onSuccess(Void aVoid) {

Toast.makeText(Admin\_dobavlenie\_dannix.this, "Тема успешно удалена", Toast.LENGTH\_SHORT).show();

adapter.notifyDataSetChanged();

}

});

} catch (Exception e) {

Toast.makeText(Admin\_dobavlenie\_dannix.this, "Ошибка удалении темы: " + e.getMessage(), Toast.LENGTH\_SHORT).show();

}

} else {

Toast.makeText(Admin\_dobavlenie\_dannix.this, "Тема не найдена", Toast.LENGTH\_SHORT).show();

}

}

}

## 1.7 Код модуля Change\_the\_test

package com.example.razvivaska\_plahova;

import static android.content.ContentValues.TAG;

import androidx.annotation.NonNull;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;

import android.os.Bundle;

import androidx.recyclerview.widget.LinearLayoutManager;

import androidx.recyclerview.widget.RecyclerView;

import android.text.Editable;

import android.text.TextWatcher;

import android.util.Log;

import android.view.View;

import android.view.ViewGroup;

import android.widget.AdapterView;

import android.widget.ArrayAdapter;

import android.widget.EditText;

import android.widget.Spinner;

import android.widget.TextView;

import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;

import com.google.android.gms.tasks.Task;

import com.google.firebase.firestore.DocumentSnapshot;

import com.google.firebase.firestore.FirebaseFirestore;

import com.google.firebase.firestore.QueryDocumentSnapshot;

import com.google.firebase.firestore.QuerySnapshot;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

public class Change\_the\_test extends AppCompatActivity {

private ArrayAdapter<Subject> subjectAdapter;

private FirebaseFirestore db;

private ArrayList<Subject> subjects;

private ArrayList<String> themes;

private Spinner spinnerSubject;

private Spinner spinnerTheme;

private RecyclerView testRecyclerView;

private TestAdapter testAdapter;

private List<Test> testList = new ArrayList<>();

private List<Test> allTests = new ArrayList<>();

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_change\_the\_test);

spinnerSubject = findViewById(R.id.subject\_spinner);

spinnerTheme = findViewById(R.id.theme\_spinner);

testRecyclerView = findViewById(R.id.test\_recyclerview);

subjects = new ArrayList<>();

themes = new ArrayList<>();

// Инициализируйте db

db = FirebaseFirestore.getInstance();

// Настройка RecyclerView

testRecyclerView.setLayoutManager(new LinearLayoutManager(this));

testAdapter = new TestAdapter(testList);

testRecyclerView.setAdapter(testAdapter);

subjectAdapter = new ArrayAdapter<Subject>(this, android.R.layout.simple\_spinner\_item, subjects) {

@Override

public View getView(int position, View convertView, ViewGroup parent) {

View view = super.getView(position, convertView, parent);

TextView textView = (TextView) view.findViewById(android.R.id.text1);

textView.setText(((Subject) getItem(position)).getName());

return view;

}

};

spinnerSubject.setAdapter(subjectAdapter);

// Загружаем все тесты из Firestore

loadTests();

// Загружаем список предметов

loadSubjects();

// Настройка Spinner'ов

spinnerSubject.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {

@Override

public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {

Subject selectedSubject = (Subject) spinnerSubject.getSelectedItem();

themes.clear();

List<String> themesList = selectedSubject.getThemes();

if (themesList != null && !themesList.isEmpty()) {

themes.addAll(themesList);

ArrayAdapter<String> adapter = new ArrayAdapter<>(Change\_the\_test.this, android.R.layout.simple\_spinner\_item, themes);

adapter.setDropDownViewResource(android.R.layout.simple\_spinner\_dropdown\_item);

spinnerTheme.setAdapter(adapter);

} else {

// Handle the case when there are no themes

spinnerTheme.setAdapter(null); // or some other default adapter

}

}

@Override

public void onNothingSelected(AdapterView<?> parent) {

// Do nothing

}

});

spinnerTheme.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {

@Override

public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {

// Фильтруем тесты по выбранному предмету и теме

filterTestsBySubjectAndTheme();

}

@Override

public void onNothingSelected(AdapterView<?> parent) {

// Do nothing

}

});

// Обработчик нажатия на элемент в RecyclerView

testRecyclerView.addOnItemTouchListener(

new RecyclerItemClickListener(this, testRecyclerView, new RecyclerItemClickListener.OnItemClickListener() {

@Override

public void onItemClick(View view, int position) {

// Переход на новую активность при нажатии на карточку

Intent intent = new Intent(Change\_the\_test.this, prohogdenie\_Test.class);

// Передача данных о тесте в новую активность

intent.putExtra("testName", testList.get(position).getname());

intent.putExtra("subject", testList.get(position).getSubject());

intent.putExtra("theme", testList.get(position).getTheme());

startActivity(intent);

}

@Override

public void onLongItemClick(View view, int position) {

// Ничего не делаем

}

})

);

}

private void loadSubjects() {

db.collection("subjects")

.get()

.addOnCompleteListener(new OnCompleteListener<QuerySnapshot>() {

@Override

public void onComplete(@NonNull Task<QuerySnapshot> task) {

if (task.isSuccessful()) {

subjects.clear();

for (QueryDocumentSnapshot document : task.getResult()) {

Subject subject = document.toObject(Subject.class);

subjects.add(subject);

}

subjectAdapter.notifyDataSetChanged();

} else {

Log.d(TAG, "Error getting subjects: ", task.getException());

}

}

});

}

private void loadTests() {

db.collection("tests")

.get()

.addOnCompleteListener(new OnCompleteListener<QuerySnapshot>() {

@Override

public void onComplete(@NonNull Task<QuerySnapshot> task) {

if (task.isSuccessful()) {

allTests.clear();

for (DocumentSnapshot document : task.getResult().getDocuments()) {

Test test = document.toObject(Test.class);

allTests.add(test);

}

// Изначально отображаем все тесты

testList.addAll(allTests);

testAdapter.notifyDataSetChanged();

} else {

Log.w(TAG, "Error getting documents", task.getException());

Toast.makeText(Change\_the\_test.this, "Ошибка загрузки тестов", Toast.LENGTH\_SHORT).show();

}

}

});

}

private void filterTestsBySubjectAndTheme() {

String selectedSubject = spinnerSubject.getSelectedItem().toString();

String selectedTheme = spinnerTheme.getSelectedItem().toString();

testList.clear();

for (Test test : allTests) {

if (test.getSubject().equals(selectedSubject) &&

test.getTheme().equals(selectedTheme)) {

testList.add(test);

}

}

testAdapter.notifyDataSetChanged();

}

}

## 1.8 Код модуля Dobavlenie

package com.example.razvivaska\_plahova;

import androidx.annotation.NonNull;

import androidx.appcompat.app.AppCompatActivity;

import android.annotation.SuppressLint;

import android.os.Bundle;

import android.util.Log;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import com.google.android.gms.tasks.OnCompleteListener;

import com.google.android.gms.tasks.OnFailureListener;

import com.google.android.gms.tasks.OnSuccessListener;

import com.google.android.gms.tasks.Task;

import com.google.firebase.firestore.CollectionReference;

import com.google.firebase.firestore.DocumentReference;

import com.google.firebase.firestore.DocumentSnapshot;

import com.google.firebase.firestore.FirebaseFirestore;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

public class Dobavlenie extends AppCompatActivity {

FirebaseFirestore db = FirebaseFirestore.getInstance();

EditText subjectEditText;

Button saveButton;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_dobavlenie);

Button addButton = findViewById(R.id.button);

EditText userInputEditText = findViewById(R.id.textView);

DocumentReference documentRef = FirebaseFirestore.getInstance().collection("Subjects").document("Subjects");

addButton.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

String userInputData = userInputEditText.getText().toString();

// Get the current list of data from the document

documentRef.get().addOnCompleteListener(new OnCompleteListener<DocumentSnapshot>() {

@Override

public void onComplete(@NonNull Task<DocumentSnapshot> task) {

if (task.isSuccessful()) {

DocumentSnapshot document = task.getResult();

if (document.exists()) {

// Get the current list of data from the document

List<String> dataList = (List<String>) document.get("dataList");

// Add the new data to the list

dataList.add(userInputData);

// Update the document with the new list

Map<String, Object> updateData = new HashMap<>();

updateData.put("dataList", dataList);

documentRef.update(updateData);

} else {

// If the document doesn't exist, create a new one with the initial list

List<String> dataList = new ArrayList<>();

dataList.add(userInputData);

Map<String, Object> data = new HashMap<>();

data.put("dataList", dataList);

documentRef.set(data);

}

} else {

Log.w("Firestore", "Error getting document", task.getException());

}

}

});

}

});

}

}

## 1.10 Код модуля MainActivity

package com.example.razvivaska\_plahova;

import android.content.Intent;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.Toast;

import androidx.annotation.NonNull;

import androidx.appcompat.app.AppCompatActivity;

import com.google.android.gms.tasks.OnCompleteListener;

import com.google.android.gms.tasks.OnFailureListener;

import com.google.android.gms.tasks.OnSuccessListener;

import com.google.android.gms.tasks.Task;

import com.google.firebase.auth.AuthResult;

import com.google.firebase.auth.FirebaseAuth;

import com.google.firebase.firestore.DocumentReference;

import com.google.firebase.firestore.DocumentSnapshot;

import com.google.firebase.firestore.FirebaseFirestore;

import java.util.HashSet;

import java.util.Random;

import java.util.Set;

// RegisterActivity.java

import android.widget.RadioButton;

import android.widget.RadioGroup;

public class MainActivity extends AppCompatActivity {

private EditText emailEditText, passwordEditText, loginEditText;

private RadioButton teacherRadioButton, studentRadioButton;

private RadioGroup roleRadioGroup;

private Button registerButton,loginButton;

private FirebaseAuth mAuth;

private FirebaseFirestore db;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

mAuth = FirebaseAuth.getInstance();

db = FirebaseFirestore.getInstance();

emailEditText = findViewById(R.id.email\_edit\_text);

passwordEditText = findViewById(R.id.password\_edit\_text);

loginEditText = findViewById(R.id.login\_edit\_text);

teacherRadioButton = findViewById(R.id.teacher\_radio\_button);

studentRadioButton = findViewById(R.id.student\_radio\_button);

roleRadioGroup = findViewById(R.id.role\_radio\_group);

registerButton = findViewById(R.id.register\_button);

registerButton.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

String email = emailEditText.getText().toString();

String password = passwordEditText.getText().toString();

String login = loginEditText.getText().toString();

String role;

RandomIdGenerator generator = new RandomIdGenerator();

String IDS = generator.generateRandomString(10);

if (teacherRadioButton.isChecked()) {

role = "Учитель";

} else if (studentRadioButton.isChecked()) {

role = "Ученик";

}else if (studentRadioButton.isChecked()) {

role = "Админ";

}else if (studentRadioButton.isChecked()) {

role = "Менеджер";

}else {

Toast.makeText(MainActivity.this, "Пожалуйста выберите роль", Toast.LENGTH\_SHORT).show();

return;

}

mAuth.createUserWithEmailAndPassword(email, password)

.addOnCompleteListener(new OnCompleteListener<AuthResult>() {

@Override

public void onComplete(@NonNull Task<AuthResult> task) {

if (task.isSuccessful()) {

// Get the current user's email

String email = emailEditText.getText().toString();

// Create a new user document in Firestore

FirebaseFirestore db = FirebaseFirestore.getInstance();

DocumentReference userRef = db.collection("users").document(email);

User user = new User(login, role, email, IDS);

// Set the user document with the User object

userRef.set(user)

.addOnSuccessListener(new OnSuccessListener<Void>() {

@Override

public void onSuccess(Void aVoid) {

Toast.makeText(MainActivity.this, "Регистрация прошла успешно", Toast.LENGTH\_SHORT).show();

}

})

.addOnFailureListener(new OnFailureListener() {

@Override

public void onFailure(@NonNull Exception e) {

Toast.makeText(MainActivity.this, "Упс, ошибка на стороне сервера, проверьте подключены ли вы к интернету", Toast.LENGTH\_SHORT).show();

}

});

} else {

Toast.makeText(MainActivity.this, "Упс, регистрация провалилась. Возможно аккаунт с такой почтой уже есть", Toast.LENGTH\_SHORT).show();

}

}

});

}

});

loginButton = findViewById(R.id.login\_button);

loginButton.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

String email = emailEditText.getText().toString();

String password = passwordEditText.getText().toString();

final String login = loginEditText.getText().toString();

// Проверка на пустоту логина

if (login.isEmpty()) {

Toast.makeText(MainActivity.this, "Введите логин", Toast.LENGTH\_SHORT).show();

return;

}

// Проверка на пустоту email и password

if (email.isEmpty() || password.isEmpty()) {

Toast.makeText(MainActivity.this, "Введите email и пароль", Toast.LENGTH\_SHORT).show();

return;

}

mAuth.signInWithEmailAndPassword(email, password)

.addOnCompleteListener(new OnCompleteListener<AuthResult>() {

@Override

public void onComplete(@NonNull Task<AuthResult> task) {

if (task.isSuccessful()) {

// Проверка роли пользователя и логина

db.collection("users").document(email)

.get()

.addOnCompleteListener(new OnCompleteListener<DocumentSnapshot>() {

@Override

public void onComplete(@NonNull Task<DocumentSnapshot> task) {

if (task.isSuccessful()) {

DocumentSnapshot document = task.getResult();

String role = document.getString("role");

String userLogin = document.getString("login"); // Получение логина из документа

Intent intent = null;

if (role != null && userLogin != null) {

if (role.equals("Ученик") && userLogin.equals(login)) { // Сравнение логина

// Открыть страницу 1 для учеников

Toast.makeText(MainActivity.this, "Добро пожаловать, ученик!", Toast.LENGTH\_SHORT).show();

intent = new Intent(MainActivity.this, ychenik\_cabinet.class);

//startActivity(new Intent(MainActivity.this, ychenik\_cabinet.class));

} else if (role.equals("Учитель") && userLogin.equals(login)) { // Сравнение логина

// Открыть страницу 2 для учителей

Toast.makeText(MainActivity.this, "Добро пожаловать, учитель!", Toast.LENGTH\_SHORT).show();

intent = new Intent(MainActivity.this, ycitel\_cabinet.class);

// startActivity(new Intent(MainActivity.this, ycitel\_cabinet.class));

} else if (role.equals("Менеджер") && userLogin.equals(login)) { // Сравнение логина

// Открыть страницу 2 для менеджеров

Toast.makeText(MainActivity.this, "Добро пожаловать, менеджер!", Toast.LENGTH\_SHORT).show();

intent = new Intent(MainActivity.this, Menedger\_cabinet.class);

/// startActivity(new Intent(MainActivity.this, Menedger\_cabinet.class));

} else if (role.equals("Админ") && userLogin.equals(login)) { // Сравнение логина

// Открыть страницу 2 для админов

Toast.makeText(MainActivity.this, "Добро пожаловать, админ!", Toast.LENGTH\_SHORT).show();

intent = new Intent(MainActivity.this, Admin\_cabinet.class);

//startActivity(new Intent(MainActivity.this, Admin\_cabinet.class));

}if (intent != null) {

// Добавляем данные в Intent перед открытием нового окна

intent.putExtra("email", email);

intent.putExtra("login", login);

startActivity(intent);

} else {

Toast.makeText(MainActivity.this, "Неверный логин или пароль", Toast.LENGTH\_SHORT).show();

}

} else {

Toast.makeText(MainActivity.this, "Ошибка получения данных пользователя", Toast.LENGTH\_SHORT).show();

}

} else {

Toast.makeText(MainActivity.this, "Ошибка получения данных пользователя", Toast.LENGTH\_SHORT).show();

}

}

});

} else {

Toast.makeText(MainActivity.this, "Ошибка входа", Toast.LENGTH\_SHORT).show();

}

}

});

}

});

}

public class RandomIdGenerator {

private final Random RANDOM = new Random();

private String generateRandomString(int length) {

StringBuilder sb = new StringBuilder();

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

int randomChar = RANDOM.nextInt(62);

if (randomChar < 26) {

sb.append((char) (randomChar + 'a'));

} else if (randomChar < 52) {

sb.append((char) (randomChar - 26 + 'A'));

} else {

sb.append((char) (randomChar - 52 + '0'));

}

}

return sb.toString();

}

}

public class User {

private String login;

private String role;

private String email;

private String IDS;

public User(String login, String role, String email, String IDS) {

this.login = login;

this.role = role;

this.email = email;

this.IDS = IDS;

}

public String getLogin() {

return login;

}

public void setLogin(String login) {

this.login = login;

}

public String getRole() {

return role;

}

public void setRole(String role) {

this.role = role;

}

public String getemail() {

return email;

}

public void setemail(String role) {

this.email = email;

}

public String getID() {

return IDS;

}

public void setID(String ID) {

this.IDS = IDS;

}

}

}

## 1.11 Код модуля Menedger\_cabinet

package com.example.razvivaska\_plahova;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.content.Intent;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

public class Menedger\_cabinet extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_menedger\_cabinet);

// Инициализируйте кнопки

Button quvestion = findViewById(R.id.button\_Quvestion);

Button Test\_button = findViewById(R.id.Test\_button);

Button q1 = findViewById(R.id.button\_Quvestion\_kratki);

Button q3 = findViewById(R.id.button\_Quvestion\_3);

quvestion.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

startActivity(new Intent(Menedger\_cabinet.this, Quvestion\_meneger.class));

}

});

q1.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

startActivity(new Intent(Menedger\_cabinet.this, Qvestion\_vvod.class));

}

});

q3.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

startActivity(new Intent(Menedger\_cabinet.this, Meneger\_qvestion\_complence.class));

}

});

Test\_button.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

startActivity(new Intent(Menedger\_cabinet.this, Meneger\_create\_a\_tests.class));

}

});

}

}

## 1.12 Код модуля Meneger\_create\_a\_tests

package com.example.razvivaska\_plahova;

import static android.content.ContentValues.TAG;

import androidx.annotation.NonNull;

import androidx.appcompat.app.AlertDialog;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

import android.util.Log;

import android.view.View;

import android.view.ViewGroup;

import android.widget.AdapterView;

import android.widget.ArrayAdapter;

import android.widget.Button;

import android.widget.EditText;

import android.widget.Spinner;

import android.widget.TextView;

import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;

import com.google.android.gms.tasks.Task;

import com.google.firebase.firestore.DocumentReference;

import com.google.firebase.firestore.DocumentSnapshot;

import com.google.firebase.firestore.FieldValue;

import com.google.firebase.firestore.FirebaseFirestore;

import com.google.firebase.firestore.QueryDocumentSnapshot;

import com.google.firebase.firestore.QuerySnapshot;

import java.util.ArrayList;

import java.util.List;

import java.util.ArrayList;

import java.util.List;

public class Meneger\_create\_a\_tests extends AppCompatActivity {

private Spinner spinnerSubjects;

private Button buttonCreateTest;

private Spinner spinnerThemes;

private Spinner spinnerQuestions;

private EditText name\_test;

private FirebaseFirestore db;

private ArrayAdapter<Subject> adapter;

private ArrayList<Subject> subjects;

private ArrayList<String> themes;

private ArrayList<String> qvestion;

private ArrayList<Questions> questionsList; // Список вопросов для Spinner

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_meneger\_create\_atests);

spinnerSubjects = findViewById(R.id.spinner\_subjects);

spinnerThemes = findViewById(R.id.spinner\_themes);

spinnerQuestions = findViewById(R.id.spinner\_questions);

name\_test = findViewById(R.id.edit\_text\_question\_title);

buttonCreateTest = findViewById(R.id.button\_create\_test);

subjects = new ArrayList<>();

themes = new ArrayList<>();

questionsList = new ArrayList<>(); // Инициализация списка вопросов

adapter = new ArrayAdapter<Subject>(this, android.R.layout.simple\_spinner\_item, subjects) {

@Override

public View getView(int position, View convertView, ViewGroup parent) {

View view = super.getView(position, convertView, parent);

TextView textView = (TextView) view.findViewById(android.R.id.text1);

textView.setText(((Subject) getItem(position)).getName());

return view;

}

};

spinnerSubjects.setAdapter(adapter);

db = FirebaseFirestore.getInstance();

buttonCreateTest.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

createTest();

}

});

// Загрузка предметов

loadSubjects();

// Обработчик выбора предмета

spinnerSubjects.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {

@Override

public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {

Subject selectedSubject = (Subject) parent.getItemAtPosition(position);

loadThemes(selectedSubject);

}

@Override

public void onNothingSelected(AdapterView<?> parent) {

// Nothing selected

}

});

// Обработчик выбора темы

spinnerThemes.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {

@Override

public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {

String selectedTheme = (String) parent.getItemAtPosition(position);

loadQuestions(selectedTheme);

}

@Override

public void onNothingSelected(AdapterView<?> parent) {

// Nothing selected

}

});

}

// Метод для загрузки тем, принимающий выбранный предмет

private void loadThemes(Subject selectedSubject) {

themes.clear(); // Очищаем список тем

List<String> themesList = selectedSubject.getThemes();

if (themesList != null && !themesList.isEmpty()) {

themes.addAll(themesList);

ArrayAdapter<String> adapter = new ArrayAdapter<>(Meneger\_create\_a\_tests.this, android.R.layout.simple\_spinner\_item, themes);

adapter.setDropDownViewResource(android.R.layout.simple\_spinner\_dropdown\_item);

spinnerThemes.setAdapter(adapter);

} else {

// Handle the case when there are no themes

spinnerThemes.setAdapter(null); // or some other default adapter

}

}

// Метод для загрузки вопросов

private void loadQuestions(String selectedTheme) {

// Get the selected subject and theme

String selectedSubject = null;

if (spinnerSubjects.getSelectedItem() != null) {

selectedSubject = spinnerSubjects.getSelectedItem().toString();

}

if (selectedSubject != null && selectedTheme != null) {

// Query Firestore for questions based on selected subject and theme

db.collection("Questions")

.whereEqualTo("subject", selectedSubject)

.whereEqualTo("theme", selectedTheme)

.get()

.addOnCompleteListener(new OnCompleteListener<QuerySnapshot>() {

@Override

public void onComplete(@NonNull Task<QuerySnapshot> task) {

if (task.isSuccessful()) {

questionsList.clear(); // Очищаем список вопросов

for (QueryDocumentSnapshot document : task.getResult()) {

Questions question = document.toObject(Questions.class);

questionsList.add(question); // Добавляем объект Questions

}

// Create and set the adapter for spinnerQuestions

ArrayAdapter<Questions> adapter = new ArrayAdapter<Questions>(getApplicationContext(), android.R.layout.simple\_spinner\_item, questionsList) {

@Override

public View getView(int position, View convertView, ViewGroup parent) {

View view = super.getView(position, convertView, parent);

TextView textView = (TextView) view.findViewById(android.R.id.text1);

textView.setText(((Questions) getItem(position)).getQuestionText()); // Изменяем текст

return view;

}

// Optional: Customize the appearance of the dropdown view

@Override

public View getDropDownView(int position, View convertView, ViewGroup parent) {

View view = super.getDropDownView(position, convertView, parent);

TextView textView = (TextView) view.findViewById(android.R.id.text1);

textView.setText(((Questions) getItem(position)).getQuestionText()); // Изменяем текст

return view;

}

};

adapter.setDropDownViewResource(android.R.layout.simple\_spinner\_dropdown\_item);

spinnerQuestions.setAdapter(adapter);

} else {

Log.w("Error", "Error loading questions", task.getException());

// Handle the error (e.g., display an error message to the user)

}

}

});

} else {

Log.w("Error", "Subject or Theme is null");

// Handle the situation where either subject or theme is null (e.g., display an error message)

}

}

private void loadSubjects() {

db.collection("subjects")

.get()

.addOnCompleteListener(new OnCompleteListener<QuerySnapshot>() {

@Override

public void onComplete(@NonNull Task<QuerySnapshot> task) {

if (task.isSuccessful()) {

subjects.clear(); // Clear the list before adding new data

for (QueryDocumentSnapshot document : task.getResult()) {

Subject subject = document.toObject(Subject.class);

subjects.add(subject);

}

adapter.notifyDataSetChanged(); // Notify the adapter of data changes

} else {

Log.d(TAG, "Error getting subjects: ", task.getException());

}

}

});

}

private void createTest() {

// Get the selected subject, theme, and questions

String selectedSubject = spinnerSubjects.getSelectedItem().toString();

String selectedTheme = spinnerThemes.getSelectedItem().toString();

Questions selectedQuestion = (Questions) spinnerQuestions.getSelectedItem();

String name\_ = name\_test.getText().toString();

String Names = name\_;

// Проверка на выбранные поля

if (selectedSubject.isEmpty() || selectedTheme.isEmpty() || selectedQuestion == null || Names.isEmpty()) {

Toast.makeText(Meneger\_create\_a\_tests.this, "Выберите предмет, тему и вопрос!", Toast.LENGTH\_SHORT).show();

return;

}

// Check if a test with the same name already exists, but only if the subject and theme are the same

db.collection("tests")

.whereEqualTo("name", Names)

.whereEqualTo("subject", selectedSubject)

.whereEqualTo("theme", selectedTheme)

.get()

.addOnCompleteListener(new OnCompleteListener<QuerySnapshot>() {

@Override

public void onComplete(@NonNull Task<QuerySnapshot> task) {

if (task.isSuccessful()) {

if (!task.getResult().isEmpty()) {

// Test with the same name exists, show a dialog

showConfirmationDialog(Names, selectedQuestion, selectedSubject, selectedTheme);

} else {

// Test with the same name does not exist, create a new test

createTestWithNewQuestion(Names, selectedQuestion, selectedSubject, selectedTheme);

}

} else {

Log.w("TestCreation", "Error checking for existing test", task.getException());

Toast.makeText(Meneger\_create\_a\_tests.this, "Ошибка, повторите ещё раз", Toast.LENGTH\_SHORT).show();

}

}

});

}

private void showConfirmationDialog(String testName, Questions selectedQuestion, String selectedSubject, String selectedTheme) {

AlertDialog.Builder builder = new AlertDialog.Builder(Meneger\_create\_a\_tests.this);

builder.setTitle("Тест уже существует");

builder.setMessage("Вы хотите добавить этот вопрос к тесту " + testName + "?");

builder.setPositiveButton("Да", (dialog, which) -> {

// Add the question to the existing test

addQuestionToExistingTest(testName, selectedQuestion, selectedSubject, selectedTheme);

});

builder.setNegativeButton("Нет", (dialog, which) -> {

// Do nothing, user canceled

});

builder.show();

}

// Create a new test with the selected question

private void createTestWithNewQuestion(String testName, Questions selectedQuestion, String selectedSubject, String selectedTheme) {

DocumentReference testRef = db.collection("tests").document();

Test test = new Test(testName, selectedSubject, selectedTheme, new ArrayList<>());

test.getQuestions().add(selectedQuestion);

testRef.set(test)

.addOnSuccessListener(documentReference -> {

Log.d("TestCreation", "Test created successfully");

Toast.makeText(Meneger\_create\_a\_tests.this, "Тест успешно создан", Toast.LENGTH\_SHORT).show();

})

.addOnFailureListener(e -> {

Log.w("TestCreation", "Error creating test", e);

Toast.makeText(Meneger\_create\_a\_tests.this, "Ошибка при создания теста", Toast.LENGTH\_SHORT).show();

});

}

// Add the selected question to an existing test

private void addQuestionToExistingTest(String testName, Questions selectedQuestion, String selectedSubject, String selectedTheme) {

db.collection("tests")

.whereEqualTo("name", testName)

.get()

.addOnCompleteListener(new OnCompleteListener<QuerySnapshot>() {

@Override

public void onComplete(@NonNull Task<QuerySnapshot> task) {

if (task.isSuccessful()) {

if (!task.getResult().isEmpty()) {

for (DocumentSnapshot document : task.getResult().getDocuments()) {

DocumentReference testRef = document.getReference();

testRef.update("questions", FieldValue.arrayUnion(selectedQuestion))

.addOnSuccessListener(aVoid -> {

Log.d("TestCreation", "Question added to existing test successfully");

Toast.makeText(Meneger\_create\_a\_tests.this, "Вопрос успешно добавлен", Toast.LENGTH\_SHORT).show();

})

.addOnFailureListener(e -> {

Log.w("TestCreation", "Error adding question to existing test", e);

Toast.makeText(Meneger\_create\_a\_tests.this, "Ошибка при добавлении вопроса", Toast.LENGTH\_SHORT).show();

});

}

}

} else {

Log.w("TestCreation", "Error updating test", task.getException());

Toast.makeText(Meneger\_create\_a\_tests.this, "Ошибка обновления теста", Toast.LENGTH\_SHORT).show();

}

}

});

}

}

class Test {

private String name;

private String subject;

private String theme;

private List<Questions> questions;

public Test(String name,String subject, String theme, ArrayList<Questions> questions) {

this.name = name;

this.subject = subject;

this.theme = theme;

this.questions = questions;

}

public String getSubject() {

return subject;

}

public Test() {

}

public String getname() {

return name;

}

public String getTheme() {

return theme;

}

public List<Questions> getQuestions() {

return questions;

}

}

## 1.13 Код модуля Meneger\_qvestion\_complence

package com.example.razvivaska\_plahova;

import static android.content.ContentValues.TAG;

import androidx.annotation.NonNull;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;

import android.os.Bundle;

import android.util.Log;

import android.view.View;

import android.view.ViewGroup;

import android.widget.AdapterView;

import android.widget.ArrayAdapter;

import android.widget.Button;

import android.widget.EditText;

import android.widget.Spinner;

import android.widget.TextView;

import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;

import com.google.android.gms.tasks.OnFailureListener;

import com.google.android.gms.tasks.OnSuccessListener;

import com.google.android.gms.tasks.Task;

import com.google.firebase.firestore.DocumentReference;

import com.google.firebase.firestore.FirebaseFirestore;

import com.google.firebase.firestore.QueryDocumentSnapshot;

import com.google.firebase.firestore.QuerySnapshot;

import java.util.ArrayList;

import java.util.List;

public class Meneger\_qvestion\_complence extends AppCompatActivity {

private Spinner spinnerSubject;

private Spinner spinnerTheme;

private Button exit;

private Button buttonCreateQuestion;

private EditText editTextQuestionText;

private ArrayAdapter<Subject> adapter;

private FirebaseFirestore db;

private ArrayList<Subject> subjects;

private ArrayList<String> themes;

private EditText editTextAnswer1;

private EditText editTextAnswer2;

private EditText editTextAnswer3;

private EditText editTextAnswer4;

private EditText editTextAnswer1\_otvet;

private EditText editTextAnswer2\_otvet;

private EditText editTextAnswer3\_otvet;

private EditText editTextAnswer4\_otvet;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_meneger\_qvestion\_complence);

db = FirebaseFirestore.getInstance();

editTextAnswer1 = findViewById(R.id.edit\_text\_answer1);

editTextAnswer2 = findViewById(R.id.edit\_text\_answer2);

editTextAnswer3 = findViewById(R.id.edit\_text\_answer3);

editTextAnswer4 = findViewById(R.id.edit\_text\_answer4);

editTextAnswer1\_otvet = findViewById(R.id.edit\_text\_answer1\_otvet);

editTextAnswer2\_otvet = findViewById(R.id.edit\_text\_answer2\_otvet);

editTextAnswer3\_otvet = findViewById(R.id.edit\_text\_answer3\_otvet);

editTextAnswer4\_otvet = findViewById(R.id.edit\_text\_answer4\_otvet);

editTextQuestionText = findViewById(R.id.edit\_text\_question\_text);

spinnerSubject = findViewById(R.id.spinner\_subject);

spinnerTheme = findViewById(R.id.spinner\_theme);

exit = findViewById(R.id.button\_exit);

buttonCreateQuestion = findViewById(R.id.button\_create\_question);

subjects = new ArrayList<>();

themes = new ArrayList<>();

adapter = new ArrayAdapter<Subject>(this, android.R.layout.simple\_spinner\_item, subjects) {

@Override

public View getView(int position, View convertView, ViewGroup parent) {

View view = super.getView(position, convertView, parent);

TextView textView = (TextView) view.findViewById(android.R.id.text1);

textView.setText(((Subject) getItem(position)).getName());

return view;

}

};

spinnerSubject.setAdapter(adapter);

exit.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

startActivity(new Intent(Meneger\_qvestion\_complence.this, Menedger\_cabinet.class));

}

});

buttonCreateQuestion.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

createQuestion();

}

});

//

loadSubjects(); // Call loadSubjects() at the end

spinnerSubject.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {

@Override

public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {

Subject selectedSubject = (Subject) spinnerSubject.getSelectedItem();

themes.clear();

List<String> themesList = selectedSubject.getThemes();

if (themesList != null && !themesList.isEmpty()) {

themes.addAll(themesList);

ArrayAdapter<String> adapter = new ArrayAdapter<>(Meneger\_qvestion\_complence.this, android.R.layout.simple\_spinner\_item, themes);

adapter.setDropDownViewResource(android.R.layout.simple\_spinner\_dropdown\_item);

spinnerTheme.setAdapter(adapter);

} else {

// Handle the case when there are no themes

spinnerTheme.setAdapter(null); // or some other default adapter

}

}

@Override

public void onNothingSelected(AdapterView<?> parent) {

// Do nothing

}

});

}

private void loadSubjects() {

db.collection("subjects")

.get()

.addOnCompleteListener(new OnCompleteListener<QuerySnapshot>() {

@Override

public void onComplete(@NonNull Task<QuerySnapshot> task) {

if (task.isSuccessful()) {

subjects.clear(); // Clear the list before adding new data

for (QueryDocumentSnapshot document : task.getResult()) {

Subject subject = document.toObject(Subject.class);

subjects.add(subject);

}

adapter.notifyDataSetChanged(); // Notify the adapter of data changes

} else {

Log.d(TAG, "Error getting subjects: ", task.getException());

}

}

});

}

private void createQuestion() {

Subject selectedSubject = (Subject) spinnerSubject.getSelectedItem();

String theme = spinnerTheme.getSelectedItem().toString();

String questionText = editTextQuestionText.getText().toString();

String correctAnswer = editTextAnswer4.getText().toString();

String incorrectAnswer1 = editTextAnswer1.getText().toString();

String incorrectAnswer2 = editTextAnswer2.getText().toString();

String incorrectAnswer3 = editTextAnswer3.getText().toString();

String correctAnswer\_otvet = editTextAnswer4\_otvet.getText().toString();

String incorrectAnswer1\_otvet = editTextAnswer1\_otvet.getText().toString();

String incorrectAnswer2\_otvet = editTextAnswer2\_otvet.getText().toString();

String incorrectAnswer3\_otvet = editTextAnswer3\_otvet.getText().toString();

String answer1 = incorrectAnswer1 + " " + incorrectAnswer1\_otvet;

String answer2 = incorrectAnswer2 + " " + incorrectAnswer2\_otvet;

String answer3 = incorrectAnswer3 + " " + incorrectAnswer3\_otvet;

String CorrectAncwer = correctAnswer + " " + correctAnswer\_otvet;

String Qvestion\_type = "Вопрос с соответствием";

Questions question = new Questions(selectedSubject.getName(), theme, questionText, answer1, answer2, answer3, CorrectAncwer,Qvestion\_type);

Toast.makeText(Meneger\_qvestion\_complence.this, "Question", Toast.LENGTH\_SHORT).show();

db.collection("Questions").add(question).addOnSuccessListener(new OnSuccessListener<DocumentReference>() {

@Override

public void onSuccess(DocumentReference documentReference) {

Toast.makeText(Meneger\_qvestion\_complence.this, "Вопрос успешно создан", Toast.LENGTH\_SHORT).show();

}

}).addOnFailureListener(new OnFailureListener() {

@Override

public void onFailure(@NonNull Exception e) {

Toast.makeText(Meneger\_qvestion\_complence.this, "Ошибка создания вопроса", Toast.LENGTH\_SHORT).show();

}

});

}

} }

## 1.14 Код модуля prohogdenie\_Test

package com.example.razvivaska\_plahova;

import androidx.annotation.NonNull;

import androidx.appcompat.app.AlertDialog;

import androidx.appcompat.app.AppCompatActivity;

import androidx.constraintlayout.widget.ConstraintLayout;

import androidx.constraintlayout.widget.ConstraintSet;

import android.content.DialogInterface;

import android.os.Bundle;

import android.content.Intent;

import android.text.Editable;

import android.text.TextWatcher;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.LinearLayout;

import android.widget.RadioButton;

import android.widget.RadioGroup;

import android.widget.TextView;

import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;

import com.google.android.gms.tasks.Task;

import com.google.firebase.auth.FirebaseAuth;

import com.google.firebase.firestore.DocumentReference;

import com.google.firebase.firestore.DocumentSnapshot;

import com.google.firebase.firestore.FirebaseFirestore;

import com.google.firebase.firestore.QuerySnapshot;

import java.util.ArrayList;

import java.util.Collections;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

import java.util.Random;

public class prohogdenie\_Test extends AppCompatActivity {

private TextView testNameTextView, subjectTextView, themeTextView;

private TextView questionTextView;

private RadioGroup answerRadioGroup;

ConstraintLayout sootvetstvieConstraintLayout;

ConstraintLayout vvod;

private Button nextButton;

private FirebaseAuth mAuth;

private FirebaseFirestore db = FirebaseFirestore.getInstance();

private List<Questions> questions = new ArrayList<>(); // Список вопросов

private int currentQuestionIndex = 0; // Индекс текущего вопроса

private int score = 0;

private boolean isAnswerChecked = false; // Флаг, чтобы проверять, был ли выбран ответ

private TextView questionText;

private LinearLayout answerContainer;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_prohogdenie\_test);

mAuth = FirebaseAuth.getInstance();

testNameTextView = findViewById(R.id.testNameTextView);

subjectTextView = findViewById(R.id.subjectTextView);

themeTextView = findViewById(R.id.themeTextView);

questionTextView = findViewById(R.id.questionTextView);

answerRadioGroup = findViewById(R.id.answerRadioGroup);

nextButton = findViewById(R.id.nextButton);

sootvetstvieConstraintLayout = findViewById(R.id.Sootvetstvie);

vvod = findViewById(R.id.vvods);

// Получение данных о тесте из Intent

Intent intent = getIntent();

String testName = intent.getStringExtra("testName");

String subject = intent.getStringExtra("subject");

String theme = intent.getStringExtra("theme");

testNameTextView.setText("Название теста: " + testName);

subjectTextView.setText("Предмет: " + subject);

themeTextView.setText("Тема: " + theme);

// Загрузка теста из Firestore

loadTest(testName, subject, theme);

// Настройка слушателя для кнопки "Следующий"

nextButton.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

// if (isAnswerChecked) {

// checkAnswer();

// isAnswerChecked = false;

// } else {

// Toast.makeText(prohogdenie\_Test.this, "Выберите ответ", Toast.LENGTH\_SHORT).show();

// }

checkAnswer();

}

});

// Настройка слушателя для RadioGroup

answerRadioGroup.setOnCheckedChangeListener(new RadioGroup.OnCheckedChangeListener() {

@Override

public void onCheckedChanged(RadioGroup group, int checkedId) {

// Если выбран какой-то RadioButton

if (checkedId != -1) {

isAnswerChecked = true;

} else {

isAnswerChecked = false;

}

}

});

}

private void loadTest(String testName, String subject, String theme) {

db.collection("tests")

.get()

.addOnCompleteListener(new OnCompleteListener<QuerySnapshot>() {

@Override

public void onComplete(@NonNull Task<QuerySnapshot> task) {

if (task.isSuccessful()) {

for (DocumentSnapshot document : task.getResult().getDocuments()) {

if (document.getString("name").equals(testName) &&

document.getString("subject").equals(subject) &&

document.getString("theme").equals(theme)) {

// Найден нужный тест

List<Map<String, Object>> questionsFromFirestore = (List<Map<String, Object>>) document.get("questions");

if (questionsFromFirestore != null && !questionsFromFirestore.isEmpty()) {

// Преобразование вопросов из Firestore в объекты Questions

for (Map<String, Object> questionData : questionsFromFirestore) {

String questionText = (String) questionData.get("questionText");

String answer1 = (String) questionData.get("answer1");

String answer2 = (String) questionData.get("answer2");

String answer3 = (String) questionData.get("answer3");

String correctAnswer = (String) questionData.get("correctAnswer"); // или (String) questionData.get("правильный\_ответ");

String type = (String) questionData.get("qvestion\_type"); // или (String) questionData.get("правильный\_ответ");

questions.add(new Questions(questionText, answer1, answer2, answer3, correctAnswer, type));

}

Collections.shuffle(questions, new Random());

// Отобразите первый вопрос

showNextQuestion();

return; // Выходим из цикла, так как тест найден

} else {

// Тест не содержит вопросов

Toast.makeText(prohogdenie\_Test.this, "Тест не содержит вопросов", Toast.LENGTH\_SHORT).show();

}

}

}

// Тест не найден

Toast.makeText(prohogdenie\_Test.this, "Тест не найден", Toast.LENGTH\_SHORT).show();

} else {

// Ошибка при чтении данных

Toast.makeText(prohogdenie\_Test.this, "Ошибка загрузки теста", Toast.LENGTH\_SHORT).show();

}

}

});

}

private void showNextQuestion() {

if (currentQuestionIndex < questions.size()) {

// Получаем текущий вопрос

Questions currentQuestion = questions.get(currentQuestionIndex);

questionTextView.setText(currentQuestion.getQuestionText());

if (currentQuestion.getqvestion\_type().equals("1 правильный, 3 неправильных")) {

sootvetstvieConstraintLayout.setVisibility(View.GONE);

vvod.setVisibility(View.GONE);

answerRadioGroup.setVisibility(View.VISIBLE); // Отображаем RadioGroup// Отобразите варианты ответов

RadioButton answerOption1 = findViewById(R.id.answerOption1);

answerOption1.setText(currentQuestion.getAnswer1());

RadioButton answerOption2 = findViewById(R.id.answerOption2);

answerOption2.setText(currentQuestion.getAnswer2());

RadioButton answerOption3 = findViewById(R.id.answerOption3);

answerOption3.setText(currentQuestion.getAnswer3());

RadioButton answerOption4 = findViewById(R.id.answerOption4);

answerOption4.setText(currentQuestion.getCorrectAnswer());

} else if (currentQuestion.getqvestion\_type().equals("Вопрос с соответствием")) {

answerRadioGroup.setVisibility(View.GONE); // Скрываем RadioGroup

sootvetstvieConstraintLayout.setVisibility(View.VISIBLE);

vvod.setVisibility(View.GONE);

String[] answers = currentQuestion.getAnswer1().split(" ");

String[] answers2 = currentQuestion.getAnswer2().split(" ");

String[] answers3 = currentQuestion.getAnswer3().split(" ");

String[] answers4 = currentQuestion.getCorrectAnswer().split(" ");

// Устанавливаем текст в RadioButtons

TextView answerOption1 = findViewById(R.id.textView16);

answerOption1.setText(answers[0]);

TextView answerOption1\_1 = findViewById(R.id.text16);

answerOption1\_1.setText(answers[1]);

TextView answerOption2 = findViewById(R.id.text7);

answerOption2.setText(answers2[0]);

TextView answerOption2\_2 = findViewById(R.id.text3);

answerOption2\_2.setText(answers2[1]);

TextView answerOption3 = findViewById(R.id.text5);

answerOption3.setText(answers3[0]);

TextView answerOption3\_3 = findViewById(R.id.text1);

answerOption3\_3.setText(answers3[1]);

TextView answerOption4 = findViewById(R.id.text6);

answerOption4.setText(answers4[0]);

TextView answerOption4\_4 = findViewById(R.id.text4);

answerOption4\_4.setText(answers4[1]);

EditText editTextOtvet = findViewById(R.id.text\_otvet);

editTextOtvet.addTextChangedListener(new TextWatcher() {

@Override

public void beforeTextChanged(CharSequence s, int start, int count, int after) {

// Не используется

}

@Override

public void onTextChanged(CharSequence s, int start, int before, int count) {

// Не используется

}

@Override

public void afterTextChanged(Editable s) {

// Проверяем ответ, когда пользователь вводит текст

checkAnswer();

}

});

//answerOption4.setVisibility(View.VISIBLE); // Отображаем 4-й вариант ответа (текстовое поле)

// ... реализуйте показ текстового поля для свободного ответа

}else if (currentQuestion.getqvestion\_type().equals("Вопрос с соответствием")) {

answerRadioGroup.setVisibility(View.GONE);

sootvetstvieConstraintLayout.setVisibility(View.GONE);

}

// Сбросьте выделенный RadioButton

answerRadioGroup.clearCheck();

currentQuestionIndex++;

} else {

// Тест завершен

showResults();

}

}

private void checkAnswer() {

EditText textOtvet1 = findViewById(R.id.text\_otvet);

EditText textOtvet2 = findViewById(R.id.text\_otvet1);

EditText textOtvet3 = findViewById(R.id.text\_otvet3);

EditText textOtvet4 = findViewById(R.id.text\_otvet4);

Questions currentQuestion = questions.get(currentQuestionIndex - 1); // Исправлено: берем предыдущий вопрос

String userAnswer = findViewById(R.id.text\_otvet).getContext().toString();

// Проверяем тип вопроса

if (currentQuestion.getqvestion\_type().equals("Вопрос с соответствием")) {

String userAnswer1 = textOtvet1.getText().toString().trim();

String userAnswer2 = textOtvet2.getText().toString().trim();

String userAnswer3 = textOtvet3.getText().toString().trim();

String userAnswer4 = textOtvet4.getText().toString().trim();

// Проверяем правильность введенных данных

if (!userAnswer1.isEmpty() && !userAnswer2.isEmpty() && !userAnswer3.isEmpty() && !userAnswer4.isEmpty()) {

// Проверяем, совпадают ли ответы пользователя с правильными ответами

boolean isCorrect =

userAnswer1.equalsIgnoreCase(currentQuestion.getAnswer1().trim()) &&

userAnswer2.equalsIgnoreCase(currentQuestion.getAnswer2().trim()) &&

userAnswer3.equalsIgnoreCase(currentQuestion.getAnswer3().trim()) &&

userAnswer4.equalsIgnoreCase(currentQuestion.getCorrectAnswer().trim());

// Выводим результат проверки

if (isCorrect) {

score++;

Toast.makeText(this, "Правильно!", Toast.LENGTH\_SHORT).show();

} else {

Toast.makeText(this, "Неверно", Toast.LENGTH\_SHORT).show();

}

showNextQuestion();

} else {

Toast.makeText(this, "Заполните все поля ответа.", Toast.LENGTH\_SHORT).show();

}

} else if (currentQuestion.getqvestion\_type().equals("1 правильный, 3 неправильных")) {

int selectedId = answerRadioGroup.getCheckedRadioButtonId();

if (selectedId != -1) {

RadioButton selectedRadioButton = findViewById(selectedId);

String selectedAnswer = selectedRadioButton.getText().toString();

if (selectedAnswer.equalsIgnoreCase(currentQuestion.getCorrectAnswer())) {

score++;

Toast.makeText(this, "Правильно!", Toast.LENGTH\_SHORT).show();

} else {

Toast.makeText(this, "Неверно", Toast.LENGTH\_SHORT).show();

}

showNextQuestion();

} else {

Toast.makeText(this, "Выберите ответ", Toast.LENGTH\_SHORT).show();

}

} else if (currentQuestion.getqvestion\_type().equals("Вопрос с вводимом ответом")) {

EditText Otvet1\_vvesti = findViewById(R.id.Otvet\_vvesti);

if (Otvet1\_vvesti.getText().toString().trim().isEmpty()) {

Toast.makeText(this, "Поле ответа не может быть пустым!", Toast.LENGTH\_SHORT).show();

return; // Выход из функции, если поле пустое

}

String userAnswers = Otvet1\_vvesti.getText().toString().trim();

if (userAnswers.equalsIgnoreCase(currentQuestion.getCorrectAnswer().trim())) {

// Ответ верный

score++;

Toast.makeText(this, "Правильно!", Toast.LENGTH\_SHORT).show();

} else {

// Ответ неверный

Toast.makeText(this, "Неверно", Toast.LENGTH\_SHORT).show();

}

}

}

private void showResults() {

// Вычисление процента правильных ответов

int percentage = (score \* 100) / questions.size();

// Отображение результатов

Toast.makeText(this, "Тест завершен! Ваш результат: " + percentage + "%", Toast.LENGTH\_LONG).show();

// Создание диалогового окна с вопросом о сохранении результата

new AlertDialog.Builder(this)

.setTitle("Сохранить результат?")

.setMessage("Хотите сохранить результат теста?")

.setPositiveButton(android.R.string.yes, new DialogInterface.OnClickListener() {

public void onClick(DialogInterface dialog, int which) {

// Сохраняем результат в Firestore

saveResult(percentage);

}

})

.setNegativeButton(android.R.string.no, null)

.setIcon(android.R.drawable.ic\_dialog\_alert)

.show();

}

private void saveResult(int percentage) {

String email = mAuth.getCurrentUser().getEmail(); // Получаем почту пользователя

Intent intent = getIntent();

String subject = intent.getStringExtra("subject");

String theme = intent.getStringExtra("theme");

String testName = intent.getStringExtra("testName");

// Создаем Map с данными для сохранения

String grade = calculateGrade(percentage);

// Создаем Map с данными для сохранения

Map<String, Object> resultData = new HashMap<>();

resultData.put("email", email);

resultData.put("subject", subject);

resultData.put("theme", theme);

resultData.put("testName", testName);

resultData.put("result", percentage);

resultData.put("grade", grade);

// Сохраняем данные в коллекцию "results" в Firestore

db.collection("results")

.add(resultData)

.addOnCompleteListener(new OnCompleteListener<DocumentReference>() {

@Override

public void onComplete(@NonNull Task<DocumentReference> task) {

if (task.isSuccessful()) {

Toast.makeText(prohogdenie\_Test.this, "Результат сохранен!", Toast.LENGTH\_SHORT).show();

startActivity(new Intent(prohogdenie\_Test.this, ychenik\_cabinet.class));

} else {

Toast.makeText(prohogdenie\_Test.this, "Ошибка сохранения результата!", Toast.LENGTH\_SHORT).show();

startActivity(new Intent(prohogdenie\_Test.this, ychenik\_cabinet.class));

}

}

});

}

private String calculateGrade(int percentage) {

if (percentage < 70) {

return "2";

} else if (percentage >= 70 && percentage <= 80) {

return "3";

} else if (percentage >= 81 && percentage <= 90) {

return "4";

} else if (percentage >= 91 && percentage <= 100) {

return "5";

} else {

return "Неизвестно"; // Добавьте обработку для некорректных значений

}

}

}}

## 1.15 Код модуля Quvestion\_meneger

package com.example.razvivaska\_plahova;

import static android.content.ContentValues.TAG;

import android.util.Log;

import android.view.ViewGroup;

import android.widget.AdapterView;

import android.widget.ArrayAdapter;

import android.widget.Button;

import android.widget.EditText;

import android.widget.Spinner;

import android.content.Intent;

import android.os.Bundle;

import android.view.View;

import android.widget.TextView;

import android.widget.Toast;

import androidx.annotation.NonNull;

import androidx.appcompat.app.AppCompatActivity;

import com.google.android.gms.tasks.OnCompleteListener;

import com.google.android.gms.tasks.OnFailureListener;

import com.google.android.gms.tasks.OnSuccessListener;

import com.google.android.gms.tasks.Task;

import com.google.firebase.firestore.DocumentReference;

import com.google.firebase.firestore.FirebaseFirestore;

import com.google.firebase.firestore.QueryDocumentSnapshot;

import com.google.firebase.firestore.QuerySnapshot;

import java.util.ArrayList;

import java.util.List;

public class Quvestion\_meneger extends AppCompatActivity {

private Spinner spinnerSubject;

private Spinner spinnerTheme;

private EditText editTextQuestionTitle;

private EditText editTextQuestionText;

private EditText editTextAnswer1;

private EditText editTextAnswer2;

private EditText editTextAnswer3;

private EditText editTextAnswer4;

private EditText editTextSubjectName;

private EditText editTextThemeName;

private Button buttonCreateQuestion;

private Button buttonAddSubject;

private Button buttonAddTheme, exit;

private ArrayAdapter<Subject> adapter;

private FirebaseFirestore db;

private ArrayList<Subject> subjects;

private ArrayList<String> themes;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_quvestion\_meneger);

db = FirebaseFirestore.getInstance();

spinnerSubject = findViewById(R.id.spinner\_subject);

spinnerTheme = findViewById(R.id.spinner\_theme);

editTextQuestionText = findViewById(R.id.edit\_text\_question\_text);

editTextAnswer1 = findViewById(R.id.edit\_text\_answer1);

editTextAnswer2 = findViewById(R.id.edit\_text\_answer2);

editTextAnswer3 = findViewById(R.id.edit\_text\_answer3);

editTextAnswer4 = findViewById(R.id.edit\_text\_answer4);

exit = findViewById(R.id.button\_exit);

buttonCreateQuestion = findViewById(R.id.button\_create\_question);

subjects = new ArrayList<>();

themes = new ArrayList<>();

adapter = new ArrayAdapter<Subject>(this, android.R.layout.simple\_spinner\_item, subjects) {

@Override

public View getView(int position, View convertView, ViewGroup parent) {

View view = super.getView(position, convertView, parent);

TextView textView = (TextView) view.findViewById(android.R.id.text1);

textView.setText(((Subject) getItem(position)).getName());

return view;

}

};

spinnerSubject.setAdapter(adapter);

exit.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

startActivity(new Intent(Quvestion\_meneger.this, Menedger\_cabinet.class));

}

});

buttonCreateQuestion.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

createQuestion();

}

});

loadSubjects(); // Call loadSubjects() at the end

spinnerSubject.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {

@Override

public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {

Subject selectedSubject = (Subject) spinnerSubject.getSelectedItem();

themes.clear();

List<String> themesList = selectedSubject.getThemes();

if (themesList != null && !themesList.isEmpty()) {

themes.addAll(themesList);

ArrayAdapter<String> adapter = new ArrayAdapter<>(Quvestion\_meneger.this, android.R.layout.simple\_spinner\_item, themes);

adapter.setDropDownViewResource(android.R.layout.simple\_spinner\_dropdown\_item);

spinnerTheme.setAdapter(adapter);

} else {

// Handle the case when there are no themes

spinnerTheme.setAdapter(null); // or some other default adapter

}

}

@Override

public void onNothingSelected(AdapterView<?> parent) {

// Do nothing

}

});

}

private void loadSubjects() {

db.collection("subjects")

.get()

.addOnCompleteListener(new OnCompleteListener<QuerySnapshot>() {

@Override

public void onComplete(@NonNull Task<QuerySnapshot> task) {

if (task.isSuccessful()) {

subjects.clear(); // Clear the list before adding new data

for (QueryDocumentSnapshot document : task.getResult()) {

Subject subject = document.toObject(Subject.class);

subjects.add(subject);

}

adapter.notifyDataSetChanged(); // Notify the adapter of data changes

} else {

Log.d(TAG, "Error getting subjects: ", task.getException());

}

}

});

}

// private void addSubject() {

// String subjectName = editTextSubjectName.getText().toString();

// Subject subject = new Subject(subjectName);

// db.collection("subjects").document(subjectName).set(new HashMap<String, String>() {{

// put("name", subjectName);

// }});

// subjects.add(subject);

// adapter.notifyDataSetChanged(); // Now this should work

// spinnerSubject.setAdapter(adapter);

// }

// private void addTheme() {

// String themeName = editTextThemeName.getText().toString();

// Subject selectedSubject = (Subject) spinnerSubject.getSelectedItem();

// selectedSubject.addTheme(themeName);

// db.collection("subjects").document(selectedSubject.getName()).update("themes", FieldValue.arrayUnion(themeName));

// themes.clear();

// themes.addAll(selectedSubject.getThemes());

// ArrayAdapter<String> adapter = new ArrayAdapter<>(Quvestion\_meneger.this, android.R.layout.simple\_spinner\_item, themes);

// adapter.setDropDownViewResource(android.R.layout.simple\_spinner\_dropdown\_item);

// spinnerTheme.setAdapter(adapter);

// }

private void createQuestion() {

Subject selectedSubject = (Subject) spinnerSubject.getSelectedItem();

String theme = spinnerTheme.getSelectedItem().toString();

String questionText = editTextQuestionText.getText().toString();

String correctAnswer = editTextAnswer4.getText().toString();

String incorrectAnswer1 = editTextAnswer1.getText().toString();

String incorrectAnswer2 = editTextAnswer2.getText().toString();

String incorrectAnswer3 = editTextAnswer3.getText().toString();

String answer1 = incorrectAnswer1;

String answer2 = incorrectAnswer2;

String answer3 = incorrectAnswer3;

String CorrectAncwer = correctAnswer;

String Qvestion\_type = "1 правильный, 3 неправильных";

Questions question = new Questions(selectedSubject.getName(), theme, questionText, answer1, answer2, answer3, CorrectAncwer,Qvestion\_type);

Toast.makeText(Quvestion\_meneger.this, "Question", Toast.LENGTH\_SHORT).show();

db.collection("Questions").add(question).addOnSuccessListener(new OnSuccessListener<DocumentReference>() {

@Override

public void onSuccess(DocumentReference documentReference) {

Toast.makeText(Quvestion\_meneger.this, "Вопрос успешно создан", Toast.LENGTH\_SHORT).show();

}

}).addOnFailureListener(new OnFailureListener() {

@Override

public void onFailure(@NonNull Exception e) {

Toast.makeText(Quvestion\_meneger.this, "Ошибка при создании вопроса", Toast.LENGTH\_SHORT).show();

}

});

}

}

## 1.16 Код модуля Qvestion \_vvod

package com.example.razvivaska\_plahova;

import static android.content.ContentValues.TAG;

import androidx.annotation.NonNull;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;

import android.os.Bundle;

import android.util.Log;

import android.view.View;

import android.view.ViewGroup;

import android.widget.AdapterView;

import android.widget.ArrayAdapter;

import android.widget.Button;

import android.widget.EditText;

import android.widget.Spinner;

import android.widget.TextView;

import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;

import com.google.android.gms.tasks.OnFailureListener;

import com.google.android.gms.tasks.OnSuccessListener;

import com.google.android.gms.tasks.Task;

import com.google.firebase.firestore.DocumentReference;

import com.google.firebase.firestore.FirebaseFirestore;

import com.google.firebase.firestore.QueryDocumentSnapshot;

import com.google.firebase.firestore.QuerySnapshot;

import java.util.ArrayList;

import java.util.List;

public class Qvestion\_vvod extends AppCompatActivity {

private Spinner spinnerSubject;

private Spinner spinnerTheme;

private EditText editTextAnswer4;

private Button exit;

private Button buttonCreateQuestion;

private EditText editTextQuestionText;

private ArrayAdapter<Subject> adapter;

private FirebaseFirestore db;

private ArrayList<Subject> subjects;

private ArrayList<String> themes;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_qvestion\_vvod);

spinnerSubject = findViewById(R.id.spinner\_subject);

spinnerTheme = findViewById(R.id.spinner\_theme);

editTextQuestionText = findViewById(R.id.edit\_text\_question\_text);

editTextAnswer4 = findViewById(R.id.edit\_text\_answer4);

exit = findViewById(R.id.button\_exit);

buttonCreateQuestion = findViewById(R.id.button\_create\_question);

db = FirebaseFirestore.getInstance();

subjects = new ArrayList<>();

themes = new ArrayList<>();

adapter = new ArrayAdapter<Subject>(Qvestion\_vvod.this, android.R.layout.simple\_spinner\_item, subjects) {

@Override

public View getView(int position, View convertView, ViewGroup parent) {

View view = super.getView(position, convertView, parent);

TextView textView = (TextView) view.findViewById(android.R.id.text1);

textView.setText(((Subject) getItem(position)).getName());

return view;

}

};

spinnerSubject.setAdapter(adapter);

exit.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

startActivity(new Intent(Qvestion\_vvod.this, Menedger\_cabinet.class));

}

});

buttonCreateQuestion.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

createQuestion();

}

});

loadSubjects(); // Call loadSubjects() at the end

spinnerSubject.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {

@Override

public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {

Subject selectedSubject = (Subject) spinnerSubject.getSelectedItem();

themes.clear();

List<String> themesList = selectedSubject.getThemes();

if (themesList != null && !themesList.isEmpty()) {

themes.addAll(themesList);

ArrayAdapter<String> adapter = new ArrayAdapter<>(Qvestion\_vvod.this, android.R.layout.simple\_spinner\_item, themes);

adapter.setDropDownViewResource(android.R.layout.simple\_spinner\_dropdown\_item);

spinnerTheme.setAdapter(adapter);

} else {

// Handle the case when there are no themes

spinnerTheme.setAdapter(null); // or some other default adapter

}

}

@Override

public void onNothingSelected(AdapterView<?> parent) {

// Do nothing

}

});

}

private void loadSubjects() {

db.collection("subjects")

.get()

.addOnCompleteListener(new OnCompleteListener<QuerySnapshot>() {

@Override

public void onComplete(@NonNull Task<QuerySnapshot> task) {

if (task.isSuccessful()) {

subjects.clear(); // Clear the list before adding new data

for (QueryDocumentSnapshot document : task.getResult()) {

Subject subject = document.toObject(Subject.class);

subjects.add(subject);

}

adapter.notifyDataSetChanged(); // Notify the adapter of data changes

} else {

Log.d(TAG, "Error getting subjects: ", task.getException());

}

}

});

}

private void createQuestion() {

Subject selectedSubject = (Subject) spinnerSubject.getSelectedItem();

String theme = spinnerTheme.getSelectedItem().toString();

String questionText = editTextQuestionText.getText().toString();

String correctAnswer = editTextAnswer4.getText().toString();

String answer1 = "";

String answer2 = "";

String answer3 = "";

String CorrectAncwer = correctAnswer;

String Qvestion\_type = "Вопрос с вводимом ответом";

Questions question = new Questions(selectedSubject.getName(), theme, questionText, answer1, answer2, answer3, CorrectAncwer,Qvestion\_type);

Toast.makeText(Qvestion\_vvod.this, "Question", Toast.LENGTH\_SHORT).show();

db.collection("Questions").add(question).addOnSuccessListener(new OnSuccessListener<DocumentReference>() {

@Override

public void onSuccess(DocumentReference documentReference) {

Toast.makeText(Qvestion\_vvod.this, "Вопрос успешно создан", Toast.LENGTH\_SHORT).show();

}

}).addOnFailureListener(new OnFailureListener() {

@Override

public void onFailure(@NonNull Exception e) {

Toast.makeText(Qvestion\_vvod.this, "Ошибка при создании вопроса", Toast.LENGTH\_SHORT).show();

}

});

}

}

## 1.17 Код модуля RecyclerItemClickListener

package com.example.razvivaska\_plahova;

import android.content.Context;

import android.view.GestureDetector;

import android.view.MotionEvent;

import android.view.View;

import androidx.recyclerview.widget.RecyclerView;

public class RecyclerItemClickListener implements RecyclerView.OnItemTouchListener {

private OnItemClickListener mListener;

public interface OnItemClickListener {

void onItemClick(View view, int position);

void onLongItemClick(View view, int position);

}

GestureDetector mGestureDetector;

public RecyclerItemClickListener(Context context, final RecyclerView recyclerView, OnItemClickListener listener) {

mListener = listener;

mGestureDetector = new GestureDetector(context, new GestureDetector.SimpleOnGestureListener() {

@Override

public boolean onSingleTapUp(MotionEvent e) {

return true;

}

@Override

public void onLongPress(MotionEvent e) {

View child = recyclerView.findChildViewUnder(e.getX(), e.getY());

if (child != null && mListener != null) {

mListener.onLongItemClick(child, recyclerView.getChildAdapterPosition(child));

}

}

});

}

@Override

public boolean onInterceptTouchEvent(RecyclerView rv, MotionEvent e) {

View child = rv.findChildViewUnder(e.getX(), e.getY());

if (child != null && mListener != null && mGestureDetector.onTouchEvent(e)) {

mListener.onItemClick(child, rv.getChildAdapterPosition(child));

return true;

}

return false;

}

@Override

public void onTouchEvent(RecyclerView rv, MotionEvent e) {

}

@Override

public void onRequestDisallowInterceptTouchEvent(boolean disallowIntercept) {

}

}

## 1.18 Код модуля Result\_ycheniki

package com.example.razvivaska\_plahova;

import androidx.annotation.NonNull;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.content.Context;

import android.widget.TextView;

import androidx.recyclerview.widget.LinearLayoutManager;

import androidx.recyclerview.widget.RecyclerView;

import com.google.firebase.auth.FirebaseAuth;

import com.google.firebase.auth.FirebaseUser;

import com.google.firebase.firestore.FirebaseFirestore;

import com.google.firebase.firestore.QueryDocumentSnapshot;

import java.util.ArrayList;

import java.util.List;

// Класс для хранения данных о результате теста

class Result {

private String email;

private String grade;

private int result;

private String subject;

private String testName;

private String theme;

public Result(String email, String grade, int result, String subject, String testName, String theme) {

this.email = email;

this.grade = grade;

this.result = result;

this.subject = subject;

this.testName = testName;

this.theme = theme;

}

// Геттеры

public String getEmail() { return email; }

public String getGrade() { return grade; }

public int getResult() { return result; }

public String getSubject() { return subject; }

public String getTestName() { return testName; }

public String getTheme() { return theme; }

}

public class Result\_ycheniki extends AppCompatActivity {

private RecyclerView recyclerView;

private ResultsAdapter adapter;

private List<Result> results = new ArrayList<>();

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_result\_ycheniki);

recyclerView = findViewById(R.id.recyclerView);

recyclerView.setLayoutManager(new LinearLayoutManager(this));

adapter = new ResultsAdapter(this, results);

recyclerView.setAdapter(adapter);

FirebaseUser user = FirebaseAuth.getInstance().getCurrentUser();

if (user != null) {

String userEmail = user.getEmail();

FirebaseFirestore db = FirebaseFirestore.getInstance();

db.collection("results")

.whereEqualTo("email", userEmail)

.get()

.addOnSuccessListener(queryDocumentSnapshots -> {

results.clear();

for (QueryDocumentSnapshot document : queryDocumentSnapshots) {

String grade = document.getString("grade");

int result = document.getLong("result").intValue();

String subject = document.getString("subject");

String testName = document.getString("testName");

String theme = document.getString("theme");

Result resultItem = new Result(userEmail, grade, result, subject, testName, theme);

results.add(resultItem);

}

adapter.notifyDataSetChanged();

})

.addOnFailureListener(e -> {

// Обработка ошибки загрузки данных

});

}

}

}

class ResultsAdapter extends RecyclerView.Adapter<ResultsAdapter.ViewHolder> {

private List<Result> results;

private Context context;

public ResultsAdapter(Context context, List<Result> results) {

this.context = context;

this.results = results;

}

@NonNull

@Override

public ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {

View itemView = LayoutInflater.from(parent.getContext())

.inflate(R.layout.result\_item, parent, false);

return new ViewHolder(itemView);

}

@Override

public void onBindViewHolder(@NonNull ViewHolder holder, int position) {

Result result = results.get(position);

holder.testName.setText(result.getTestName());

holder.subject.setText(result.getSubject());

holder.grade.setText(result.getGrade());

holder.result.setText("Результат: " + result.getResult());

}

@Override

public int getItemCount() {

return results.size();

}

public static class ViewHolder extends RecyclerView.ViewHolder {

public TextView testName;

public TextView subject;

public TextView grade;

public TextView result;

public ViewHolder(@NonNull View itemView) {

super(itemView);

testName = itemView.findViewById(R.id.testName);

subject = itemView.findViewById(R.id.subject);

grade = itemView.findViewById(R.id.grade);

result = itemView.findViewById(R.id.result);

}

}

} }

## 1.19 Код модуля Result\_ychitel

package com.example.razvivaska\_plahova;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

import android.os.Bundle;

import android.widget.EditText;

import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import androidx.recyclerview.widget.LinearLayoutManager;

import androidx.recyclerview.widget.RecyclerView;

import com.google.firebase.auth.FirebaseAuth;

import com.google.firebase.auth.FirebaseUser;

import com.google.firebase.firestore.FirebaseFirestore;

import com.google.firebase.firestore.QueryDocumentSnapshot;

import java.util.ArrayList;

import java.util.List;

public class Result\_ychitel extends AppCompatActivity {

private EditText emailInput;

private EditText idInput;

private RecyclerView recyclerView;

private ResultsAdapter adapter;

private List<Result> results = new ArrayList<>();

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_result\_ychitel);

emailInput = findViewById(R.id.emailInput); // ID вашего EditText для email

idInput = findViewById(R.id.idInput); // ID вашего EditText для ID

recyclerView = findViewById(R.id.recyclerView);

recyclerView.setLayoutManager(new LinearLayoutManager(this));

adapter = new ResultsAdapter(this, results);

recyclerView.setAdapter(adapter);

findViewById(R.id.searchButton).setOnClickListener(view -> {

String email = emailInput.getText().toString().trim();

String id = idInput.getText().toString().trim();

loadResultsFromFirestore(email, id);

});

}

private void loadResultsFromFirestore(String email, String id) {

FirebaseFirestore db = FirebaseFirestore.getInstance();

db.collection("results")

.whereEqualTo("email", email)

.get()

.addOnSuccessListener(queryDocumentSnapshots -> {

results.clear();

for (QueryDocumentSnapshot document : queryDocumentSnapshots) {

String grade = document.getString("grade");

int result = document.getLong("result").intValue();

String subject = document.getString("subject");

String testName = document.getString("testName");

String theme = document.getString("theme");

Result resultItem = new Result(email, grade, result, subject, testName, theme);

results.add(resultItem);

}

adapter.notifyDataSetChanged();

})

.addOnFailureListener(e -> {

// Обработка ошибки загрузки данных

Toast.makeText(this, "Ошибка загрузки данных", Toast.LENGTH\_SHORT).show();

});

}

}

## 1.20 Код модуля ychenik\_cabinet

package com.example.razvivaska\_plahova;

import androidx.appcompat.app.AppCompatActivity;

import android.annotation.SuppressLint;

import android.content.Intent;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.TextView;

public class ychenik\_cabinet extends AppCompatActivity {

private Button buttonCreateTest, result;

private TextView email\_view,log\_view;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_ychenik\_cabinet);

Intent intent = getIntent();

buttonCreateTest = findViewById(R.id.test\_baton);

result = findViewById(R.id.test\_res\_baton);

email\_view = findViewById(R.id.pochta);

log\_view = findViewById(R.id.log);

log\_view.setText(intent.getStringExtra("login"));

email\_view.setText(intent.getStringExtra("email"));

buttonCreateTest.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

startActivity(new Intent(ychenik\_cabinet.this, Change\_the\_test.class));

}

});

result.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

startActivity(new Intent(ychenik\_cabinet.this, Result\_ycheniki.class));

}

});

}

}

## 1.22 Код модуля ycitel\_cabinet

package com.example.razvivaska\_plahova;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

public class ycitel\_cabinet extends AppCompatActivity {

private Button result;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_ycitel\_cabinet);

result = findViewById(R.id.res\_yc);

result.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

startActivity(new Intent(ycitel\_cabinet.this, Result\_ychitel.class));

}

});

}

}