

Facultatea de Matematica si Informatica

**Proiect programare pe dispozitive
mobile
Quiz matematica**

Candidat
Tarce Alexandru,
profil informatica romana
Subgrupa 10

2022

CUPRINS:

Cap I. Motivarea alegerii lucrării.....	3
Cap II. Descrierea programului.....	3
Cap III. Imbunatatiri.....	16
Cap IV. Bibliografie.....	17

Cap I. Motivarea alegerii lucrării

Am ales aceasta lucrare cu gandul la copiii din clasele primare, pentru a invata intr-un mod mai amuzant si contemporan adunarile, scaderile, tabla inmultirii si impartirii. Aceasta aplicatie a fost gandita si in cazul unei noi pandemii si a trecerii invatamantului inapoi in mediul online.

Cap II. Descrierea programului

Programul este ccompus din 3 clase activity:

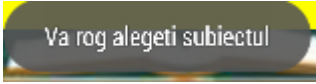
1. activity_main.xml



Acest ecran reprezinta meniul aplicatiei. De aici userul are de ales din ce tip de quiz vrea sa se testeze utilizand butoanele inscriptionate cu semne matematice si poate porni quizul apasand butonul

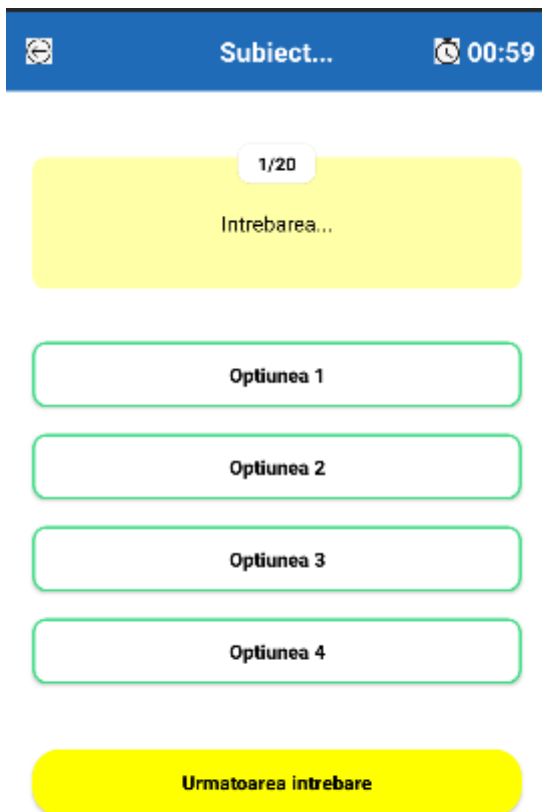


Butonul nu poate fi accesat cat timp nu este ales un subiect, aceasta actiune afiseaza un mesaj de informare pe ecran



Acest buton face trimitere la cel de-al doilea ecran si anume activity_activitati.

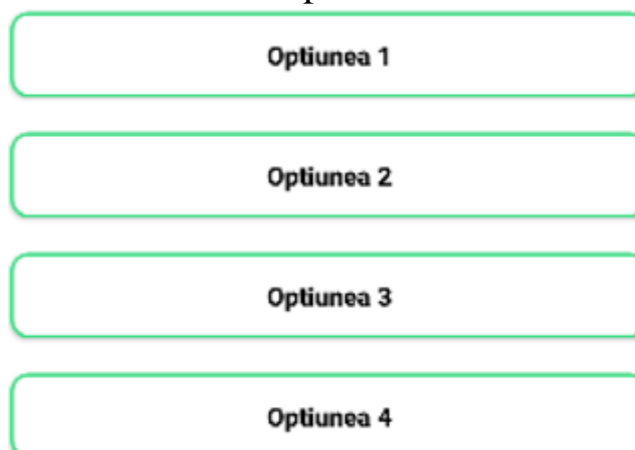
2. activity_activitati.xml



The screenshot shows a quiz application interface. At the top is a blue header bar with a back arrow icon, the text "Subiect...", and a clock icon with the time "00:59". Below the header is a yellow box containing the text "1/20" and "Intrebarea...". Underneath this are four white buttons with green borders, labeled "Optiunea 1", "Optiunea 2", "Optiunea 3", and "Optiunea 4". At the bottom is a yellow button labeled "Urmatoarea intrebare".

In acest activity se realizeaza propriu-zis testul. In zona **Subiect...** se va afisa numele operatiei pe care e bazat quizul. Exista o banca de intrebari din care

aceasta pagina sa alimenteaza. In cele patru butoane albe se afiseaza optiunile



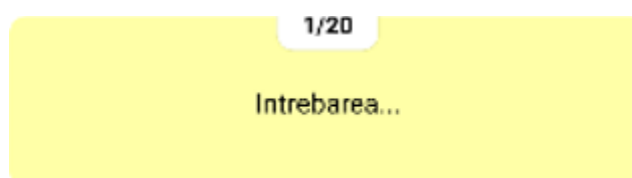
Optiunea 1

Optiunea 2

Optiunea 3


Optiunea 4

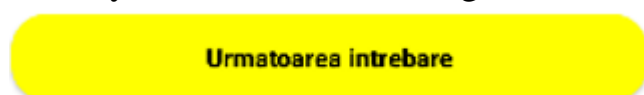
de raspunsuri pentru intrebarea care va fi afisata in chenarul .



1/20


Intrebarea...

Butonul cu sageata  este un buton back care ne trimite inapoi la activity_main. Pentru a naviga intre intrebari se foloseste butonul



Urmatoarea intrebare

, dupa ajungerea la ultima intrebare

butonul va face trimitere la activity_rezultat. Exista un timer  00:59 setata pe 59 de secunde, in cazul ajungerii la 0 testul se incheie si afiseaza ecranul activity_rezultat, asemenea si in cazul in care se raspunde la toate intrebarile inainte de expirarea timerului.

3. activity_rezultat.xml



Acest activity afiseaza raspunsurile corecte si gresite date de user pentru ca acesata sa se poata autoevalua. Numerele raspunsurilor se iau din Activitati.java fiind folosite functiile IaRaspunsuriCorecte() si IaRaspunsuriGresite() si se afiseaza in zonele

Butonul  ne trimite inapoi la meniul principal.

Functionalitatile sunt scrise in java si exista 5 clase care le realizeaza:

MainActivity , Activitati , ListaIntrebari , BancaDeIntrebari si Rezultat.

Clasele MainActivity, Activitati si Rezultat asigura functionarea elementelor din fisierele activity, iar ListaIntrebari si BancaDeIntrebari gestioneaza intrebarile,

raspunsurile si tin intrebarile care sunt mai apoi trimise in clasa Activitati pentru a fi afisate in activity_activitati.

In clasa MainActivity avem create functionalitatile pentru butoanele din activity_main care sunt recunoscute dupa id-urile primite.

```
final LinearLayout adunare = findViewById(R.id.adunare);  
final LinearLayout scadere = findViewById(R.id.scadere);  
final LinearLayout inmultire = findViewById(R.id.inmultire);  
final LinearLayout impartire = findViewById(R.id.impartire);  
  
final Button startbtn = findViewById(R.id.butonstart);
```

Butoanele adunare, scadere, inmultire, impartire sunt setate asemanator,

```
scadere.setOnClickListener(new View.OnClickListener()  
{  
    @Override  
    public void onClick(View v)  
    {  
        Subiect = "Scadere";  
  
        adunare.setBackgroundResource(R.drawable.butonalegere);  
        scadere.setBackgroundResource(R.drawable.buton_start_spate);  
        inmultire.setBackgroundResource(R.drawable.butonalegere);  
        impartire.setBackgroundResource(R.drawable.butonalegere);  
    }  
});
```

```
adunare.setOnClickListener(new View.OnClickListener()  
{  
    @Override  
    public void onClick(View v)  
    {  
        Subiect = "Adunare";  
  
        adunare.setBackgroundResource(R.drawable.buton_start_spate);  
        scadere.setBackgroundResource(R.drawable.butonalegere);  
        inmultire.setBackgroundResource(R.drawable.butonalegere);  
        impartire.setBackgroundResource(R.drawable.butonalegere);  
    }  
});
```

```

inmultire.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View v)
    {
        Subiect = "Inmultire";

        adunare.setBackgroundResource(R.drawable.butonalegere);
        scadere.setBackgroundResource(R.drawable.butonalegere);
        inmultire.setBackgroundResource(R.drawable.buton_start_spate);
        impartire.setBackgroundResource(R.drawable.butonalegere);
    }
});

```

```

impartire.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View v)
    {
        Subiect = "Impartire";

        adunare.setBackgroundResource(R.drawable.butonalegere);
        scadere.setBackgroundResource(R.drawable.butonalegere);
        inmultire.setBackgroundResource(R.drawable.butonalegere);
        impartire.setBackgroundResource(R.drawable.buton_start_spate);
    }
});

```

acestea ofera Subiectul in momentul apasarii astfel incat programul sa stie la apasarea butonului startbtn in care quiz sa intre. Butonul care este ales este evidentiata prin schimbarea infatisarii folosind un drawable diferit fata de restul butoanelor.

Butonul startbtn in cazul in care nu este ales un Subiect afiseaza un mesaj informativ care anunta ca nu este ales un Subiect, altfel initiaza un intent care ruleaza clasa Activitati si astfel porneste activity_activitati si porneste quizul.

```

startbtn.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v){
        if(Subiect.isEmpty()){
            Toast.makeText(context MainActivity.this, "Va rog alegeți subiectul", Toast.LENGTH_SHORT).show();
        }
        else{
            Intent intent = new Intent( packageContext MainActivity.this, Activitati.class);
            intent.putExtra( name: "Subiect", Subiect);
            startActivity(intent);
        }
    }
});

```


In clasa Activitati se afla create functionalitatile pentru intrebari, timer, butoanele de optiuni, butonul de schimbat intrebarea, butonul de mers inapoi pe pagina principala si afisarea subiectului ales.

```
public class Activitati extends AppCompatActivity {  
  
    private TextView intrebari;  
    private TextView intrebare;  
  
    private AppCompatActivity opt1,opt2,opt3,opt4;  
  
    private AppCompatActivity urmatorulBtn;  
  
    private Timer apptimer;  
  
    private int TimpinMin = 1;  
  
    private int secunde = 0;  
  
    private List<ListaIntrebari> listaintrebari ;  
  
    private int PozIntrebareCurenta = 0;  
  
    private String OptiuneSelectata = "" ;
```

```
final ImageView backbtn = findViewById(R.id.backbtn);  
final TextView timer = findViewById(R.id.timer);  
final TextView subiect = findViewById(R.id.subiect);
```

Elementele din activity_activitati sunt recunoscute in clasa Activitati dupa id-uri

```
final ImageView backbtn = findViewById(R.id.backbtn);  
final TextView timer = findViewById(R.id.timer);  
final TextView subiect = findViewById(R.id.subiect);  
  
intrebari = findViewById(R.id.intrebari);  
intrebare = findViewById(R.id.intrebare);  
  
opt1 = findViewById(R.id.opt1);  
opt2 = findViewById(R.id.opt2);  
opt3 = findViewById(R.id.opt3);  
opt4 = findViewById(R.id.opt4);  
  
urmatorulBtn = findViewById(R.id.urm);
```

Subiectul ales este luat prin comanda

```
getIntent().getStringExtra( name: "Subiect");
```

Lista de intrebari folosita in quiz este luata din clasa BancaDeIntrebari unde acestea sunt scrise astfel incat sa poata fi recunoscute si au setate raspunsurile in numar de 4 din care 3 sunt gresite si una este corecta, fiecare quiz este recunoscut dupa numele ales la Subiect si continue o lista de 10 intrebari care este returnata pentru afisare in clasa Activitati.

```
private static List<ListaIntrebari> Adunare() {  
  
    final List<ListaIntrebari> listeDeIntrebari = new ArrayList<>();  
  
    final ListaIntrebari intrebare1 = new ListaIntrebari( intrebare: "Cat fac 1+1?", opt1: "4", opt2: "2", opt3: "5", opt4: "1", raspuns: "2", raspunsSelectat: "");  
    final ListaIntrebari intrebare2 = new ListaIntrebari( intrebare: "Cat fac 1+2?", opt1: "4", opt2: "2", opt3: "3", opt4: "1", raspuns: "3", raspunsSelectat: "");  
    final ListaIntrebari intrebare3 = new ListaIntrebari( intrebare: "Cat fac 1+3?", opt1: "2", opt2: "8", opt3: "3", opt4: "4", raspuns: "4", raspunsSelectat: "");  
    final ListaIntrebari intrebare4 = new ListaIntrebari( intrebare: "Cat fac 1+4?", opt1: "2", opt2: "8", opt3: "5", opt4: "4", raspuns: "5", raspunsSelectat: "");  
    final ListaIntrebari intrebare5 = new ListaIntrebari( intrebare: "Cat fac 1+5?", opt1: "2", opt2: "6", opt3: "3", opt4: "4", raspuns: "6", raspunsSelectat: "");  
    final ListaIntrebari intrebare6 = new ListaIntrebari( intrebare: "Cat fac 1+6?", opt1: "2", opt2: "8", opt3: "3", opt4: "7", raspuns: "7", raspunsSelectat: "");  
    final ListaIntrebari intrebare7 = new ListaIntrebari( intrebare: "Cat fac 1+7?", opt1: "0", opt2: "8", opt3: "3", opt4: "4", raspuns: "8", raspunsSelectat: "");  
    final ListaIntrebari intrebare8 = new ListaIntrebari( intrebare: "Cat fac 1+8?", opt1: "9", opt2: "8", opt3: "10", opt4: "4", raspuns: "9", raspunsSelectat: "");  
    final ListaIntrebari intrebare9 = new ListaIntrebari( intrebare: "Cat fac 1+9?", opt1: "2", opt2: "16", opt3: "3", opt4: "4", raspuns: "10", raspunsSelectat: "");  
    final ListaIntrebari intrebare10 = new ListaIntrebari( intrebare: "Cat fac 1+0?", opt1: "2", opt2: "1", opt3: "3", opt4: "4", raspuns: "1", raspunsSelectat: "");  
  
    listeDeIntrebari.add(intrebare1);  
    listeDeIntrebari.add(intrebare2);  
    listeDeIntrebari.add(intrebare3);  
    listeDeIntrebari.add(intrebare4);  
    listeDeIntrebari.add(intrebare5);  
    listeDeIntrebari.add(intrebare6);  
    listeDeIntrebari.add(intrebare7);  
    listeDeIntrebari.add(intrebare8);  
    listeDeIntrebari.add(intrebare9);  
    listeDeIntrebari.add(intrebare10);  
  
    return listeDeIntrebari;  
};
```

Pentru alegerea listei de intrebari se foloseste o functie switch care face alegerea pe baza Subiectului.

```
public static List<ListaIntrebari> listaIntrebari(String Subiect)  
{  
    switch(Subiect)  
    {  
        case "Adunare":  
            return Adunare();  
        case "Scadere":  
            return Scadere();  
        case "Inmultire":  
            return Inmultire();  
        default:  
            return Impartire();  
    }  
}
```

Revenind inapoi in clasa Activitati, butoanele optiunilor sunt setate astfel incat daca raspunsul nu este cel corect butonul isi schimba culoarea in rosu si afiseaza optiunea corecta schimbandu-i culoarea in verde.

```
opt1.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
  
        if(OptiuneSelectata.isEmpty())  
        {  
            OptiuneSelectata = opt1.getText().toString();  
  
            opt1.setBackgroundResource(R.drawable.butonrosu);  
            opt1.setTextColor(Color.WHITE);  
  
            RaspunsCorect();  
  
            listaintrebari.get(PozIntrebareCurenta).setRaspunsSelectat(OptiuneSelectata);  
        }  
    }  
});
```

Fiecare buton pentru optiune este setat la fel.

Daca se alege raspunsul corect se apeleaza doar functia RaspunsCorect() care modifica optiunea aleasa folosind un drawable care coloreaza butonul verde si textul din acesta alb.

```
private void RaspunsCorect()  
{  
    final String getRaspunsCorect = listaintrebari.get(PozIntrebareCurenta).getRaspuns();  
  
    if(opt1.getText().toString().equals(getRaspunsCorect))  
    {  
        opt1.setBackgroundResource(R.drawable.butonverde);  
        opt1.setTextColor(Color.WHITE);  
    }  
    else if(opt2.getText().toString().equals(getRaspunsCorect))  
    {  
        opt2.setBackgroundResource(R.drawable.butonverde);  
        opt2.setTextColor(Color.WHITE);  
    }  
    else if(opt3.getText().toString().equals(getRaspunsCorect))  
    {  
        opt3.setBackgroundResource(R.drawable.butonverde);  
        opt3.setTextColor(Color.WHITE);  
    }  
    else if(opt4.getText().toString().equals(getRaspunsCorect))  
    {  
        opt4.setBackgroundResource(R.drawable.butonverde);  
        opt4.setTextColor(Color.WHITE);  
    }  
}
```

Butonul urmatorulBtn este setat astfel incat daca este ales un Subiect se apeleaza functia urmatoareIntrebare(), altfel afiseaza un mesaj informativ prin care anunta ca nu este ales un Subiect.

```
urmatorulBtn.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        if(subiectulAles.isEmpty())  
        {  
            Toast.makeText(context: Activitati.this, text: "Selectati o optiune!", Toast.LENGTH_SHORT).show();  
        }  
        else{  
            urmatoareIntrebare();  
        }  
    }  
});
```

Functia urmatoareIntrebare() lucreaza pe baza pozitiei curente a intrebării, daca pozitia curenta + 1 este egala cu marimea listei atunci textul butonului se schimba

```
private void urmatoareIntrebare()  
{  
    PozIntrebareCurenta++;  
  
    if ((PozIntrebareCurenta+1) == listaintrebari.size())  
    {  
        urmatorulBtn.setText("Termina test!");  
    }  
}
```

Daca pozitia curenta este mai mica atunci variabila OptiuneSelectata este setata pe null, aspectul butoanelor de optiune revin la infatisarea originala, intrebarea se schimba cu urmatoarea din lista, iar butoanele optiuni primesc textul si valorile asiguate intrebării curente.

```

if(PozIntrebareCurenta < listaintrebari.size())
{
    OptiuneSelectata = "";

    opt1.setBackgroundResource(R.drawable.backgroundoptiune2);
    opt1.setTextColor(Color.parseColor( colorString: "#000000"));

    opt2.setBackgroundResource(R.drawable.backgroundoptiune2);
    opt2.setTextColor(Color.parseColor( colorString: "#000000"));

    opt3.setBackgroundResource(R.drawable.backgroundoptiune2);
    opt3.setTextColor(Color.parseColor( colorString: "#000000"));

    opt4.setBackgroundResource(R.drawable.backgroundoptiune2);
    opt4.setTextColor(Color.parseColor( colorString: "#000000"));

    intrebari.setText((PozIntrebareCurenta+1)+"/"+listaintrebari.size());
    intrebare.setText(listaintrebari.get(PozIntrebareCurenta).getIntrebare());
    opt1.setText(listaintrebari.get(PozIntrebareCurenta).getOpt1());
    opt2.setText(listaintrebari.get(PozIntrebareCurenta).getOpt2());
    opt3.setText(listaintrebari.get(PozIntrebareCurenta).getOpt3());
    opt4.setText(listaintrebari.get(PozIntrebareCurenta).getOpt4());
}

```

Insa daca pozitia curenta este mai mare decat marimea listei se incheie quizul, se apeleaza clasa Rezultat printr-un new Intent si functiile IaRaspunsuriCorecte() , IaRaspunsuriGresite() care vor afisa in clasa Rezultat numarul de raspunsuri corecte si gresite acumulate pe parcursul quizului.

```

else
{
    Intent intent = new Intent( packageContext: Activitati.this, Rezultat.class);
    intent.putExtra( name: "Corect!", IaRaspunsuriCorecte());
    intent.putExtra( name: "Gresit!", IaRaspunsuriGresite());
    startActivity(intent);

    finish();
}

```

Funcțiile IaRaspunsuriCorecte() și IaRaspunsuriGresite() funcționează pe baza răspunsurilor alese, contorizând câte răspunsuri corecte și câte gresite au fost date.

```
private int IaRaspunsuriCorecte(){  
  
    int raspunsuriCorecte = 0;  
  
    for(int i=0;i<listaintrebari.size();i++)  
    {  
        final String getraspunsselectat = listaintrebari.get(i).getRaspunsSelectat();  
        final String getRaspuns = listaintrebari.get(i).getRaspuns();  
  
        if(getraspunsselectat.equals(getRaspuns))  
        {  
            raspunsuriCorecte++;  
        }  
    }  
  
    return raspunsuriCorecte;  
}
```

```
private int IaRaspunsuriGresite(){  
  
    int raspunsuriGresite = 0;  
  
    for(int i=0;i<listaintrebari.size();i++)  
    {  
        final String raspunsselectat = listaintrebari.get(i).getRaspunsSelectat();  
        final String Raspuns = listaintrebari.get(i).getRaspuns();  
  
        if(!raspunsselectat.equals(Raspuns))  
        {  
            raspunsuriGresite++;  
        }  
    }  
  
    return raspunsuriGresite;  
}
```

Timerul porneste în momentul în care se deschide clasa Activitati și începe quizul

```
startTimer(timer);
```

Acesta este setat pe 59 de secunde si scade pana la 0. Cand ajunge la 0 quizul se opreste si apeleaza clasa Rezultat la fel ca butonul urmatorulBtn cand lista de intrebari se termina.

```
private void startTimer(TextView timerTextView){
    apptimer = new Timer();
    apptimer.scheduleAtFixedRate(new TimerTask() {
        @Override
        public void run() {

            if(secunde == 0)
            {
                TimpinMin--;
                secunde = 59;
            }
            else if(secunde == 0 && TimpinMin == 0)
            {
                apptimer.purge();
                apptimer.cancel();

                Toast.makeText( context: Activitati.this, text: "Timp terminat", Toast.LENGTH_SHORT).show();

                Intent intent = new Intent( packageContext: Activitati.this, Rezultat.class);
                intent.putExtra( name: "Corect", IaRaspunsuriCorecte());
                intent.putExtra( name: "Gresit", IaRaspunsuriGresite());
                startActivity(intent);

                finish();
            }

            else{
                secunde--;
            }
        }
    });
}
```

Butonul backbtn ne intoarce la clasa MainActivity.

```
backbtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v)
    {
        apptimer.purge();
        apptimer.cancel();

        startActivity(new Intent( packageContext: Activitati.this, MainActivity.class));
        finish();
    }
});
```

In clasa Rezultat sunt create functionalitatile pentru activity_rezultat. Clasa contine doua zone de text in care se afiseaza numarul de raspunsuri corecte si numarul de raspunsuri gresite si un buton care sa ne trimita inapoi in activity_main.

```
final AppCompatActivity restart = findViewById(R.id.butonrestart);
final TextView raspunsuricorecte = findViewById(R.id.raspunsuricorecte);
final TextView raspincorecte = findViewById(R.id.raspunsuriincorecte);

final int IaRaspunsuriCorecte = getIntent().getIntExtra( name: "Raspunsuri corecte", defaultValue: 0);
final int IaRaspunsuriGresite = getIntent().getIntExtra( name: "Raspunsuri incorecte", defaultValue: 0);

raspunsuricorecte.setText(String.valueOf(IaRaspunsuriCorecte));
raspincorecte.setText(String.valueOf(IaRaspunsuriGresite));

restart.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        startActivity(new Intent( packageContext: Rezultat.this, MainActivity.class));
        finish();
    }
});
```

Funcțiile onBackPressed() sunt create pentru a simula apăsarea butonului back de către user.

```
@Override
public void onBackPressed()
{
    startActivity(new Intent( packageContext: Rezultat.this, MainActivity.class));
    finish();
}
```

Cap III. Îmbunătățiri

- Introducerea unui sistem de acordare de premii pentru rezolvarea corectă a quiz-urilor.
- Introducerea unei zone de teorie în care să existe explicații legate de calculele din quiz.
- Rezolvarea timerului pentru a rula funcția intent trimițându-ne înapoi în meniul principal.
- Rezolvarea paginii activity_rezultat astfel încât să poată fi afișată.

Cap IV. Bibliografie

Tutoriale de pe internet:

- <https://www.itcsolutions.eu/ro/2011/10/08/tutorial-android-6-cum-sa-construiesti-si-sa-afisezi-o-noua-activitate-fereastra-sau-formular/>
- <https://docs.microsoft.com/en-us/dotnet/api/android.app.activity.findviewbyid?view=xamarin-android-sdk-12>
- <https://www.tabnine.com/code/java/methods/androidx.appcompat.widget.AppCompatButton/setOnClickListener>
- <https://developer.android.com/guide/navigation/navigation-custom-back>
- <https://stackoverflow.com/questions/4597690/how-to-set-timer-in-android>
- <https://stackoverflow.com/questions/50046201/the-difference-between-finish-and-onBackPressed>