

Raport final

Galaon Georgeta Lavinia
Universitatea de Vest din Timișoara
Facultatea de Matematică și Informatică
Anul 2

Programare pe dispozitive mobile
16.05.2021
ToBeCareful

Contents

Abstract.....	3
Scop și potențiali utilizatori	3
• Scop:	3
• Utilizatori	3
Introducere	3
Aplicații similare	4
Contribuțiile autorului – declarația de originalitate.....	6
Manualul utilizatorului	6
Funcționalitate și structura aplicație	14
Concluzii și direcții viitoare	22
Referințe	23

Abstract

Acest raport prezintă o schiță generală despre proiectul pe care îl voi realiza și anume o aplicație mobilă care vine ca un ajutor pentru managementul bolii tuberculoza, numele aplicației va fi ToBeCareful. În primele părți ale raportului sunt prezentate informații generale despre proiect: scopul aplicației, o scurtă introducere despre aplicație, aplicații similare; în părțile ulterioare ale raportului sunt prezentate informații mai detaliate: contribuțiile autorului, structura aplicației, precum și direcțiile viitoare pentru dezvoltare aplicației.

Scop și potențiali utilizatori

- **Scop:**

Aplicația permite o gestiune mai ușoară a problemelor pe care persoanele cu asemenea afecțiuni le întâmpină pe parcursul procesului de vindecare cum ar fi: calendare care monitorizează programările la doctor, zilele în care trebuie luate pastilele și de asemenea trimite notificări pentru evenimente, date exacte în caz ca utilizatorul dorește să se informeze cu privire la evoluția bolii în restul lumii și informații utile despre tuberculoză și surse utile pentru suportul psihic al utilizatorului pentru a le avea la îndemână atunci când simte nevoia: un telefon pentru un helpline non-stop și o trimitere către o pagină în care își pot programa sesiuni de terapie gratuite.

- **Utilizatori**

Aplicația se adresează în special persoanelor afectate de tuberculoză, dar și oricui care doar vrea să se documenteze în legătură cu afecțiunea sau cum să o prevină.

Introducere

În ziua de astăzi, având o viață atât de aglomerată, oamenilor le poate fi din ce în ce mai greu să țină toate datele legate de un anumit aspect dacă nu le notează: au luat sau nu pastilele? Trebuiau luate ieri sau astăzi? Când trebuie să meargă la doctor?

De aceea o aplicație care poate să îi ajute cu aceste probleme este folosită. Îi ajută pe oameni să economisească timp și cum să învețe să fie conștiincioși și precauți cu privire la afecțiunea cu care se confruntă.

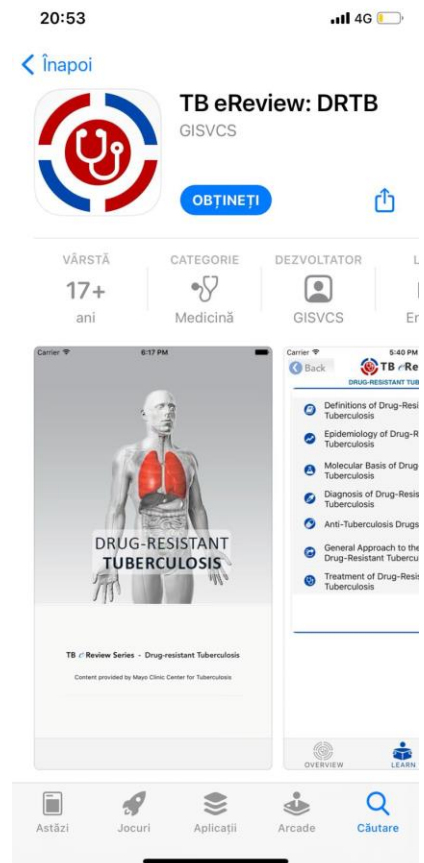
Aplicații similare

Pe internet exista aplicatii similare care doar prezinta informatii despre evolutia afectiunii, despre afectiune sau invatare interactiva, dar nicio aplicatie care sa puna laolalta atat informatii cat si mijloace care sa suporte colectarea de informatii intr-un singur loc, fara a fi nevoie de mai multe aplicatii pentru a stoca datele importante.

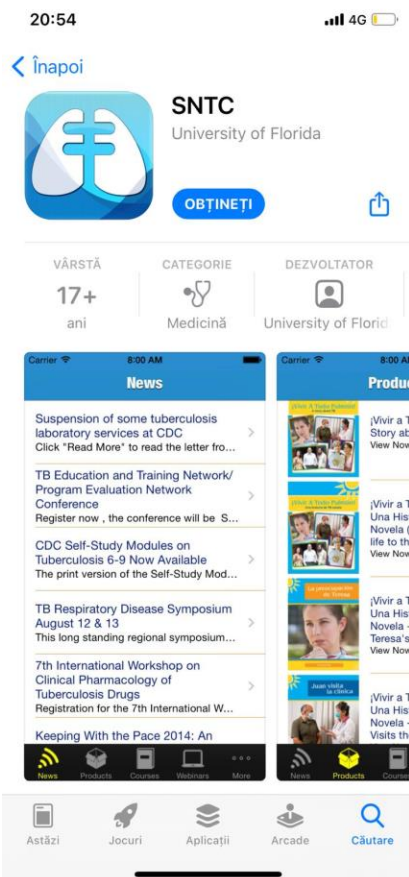
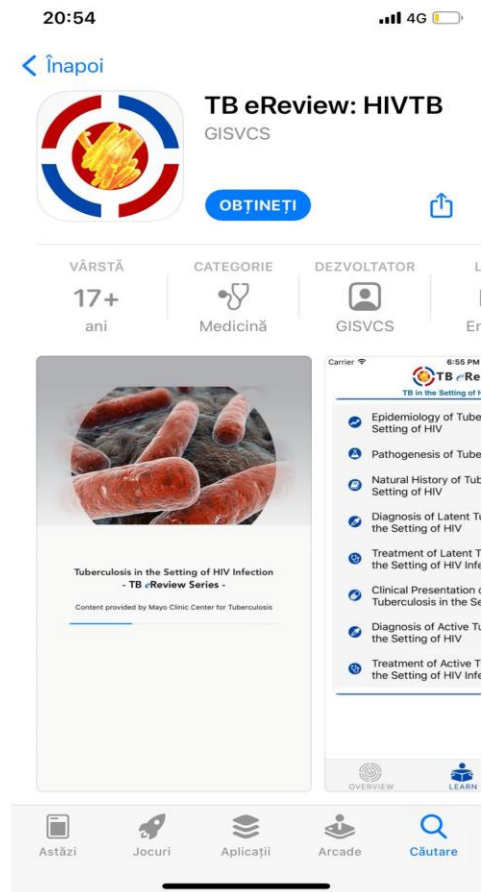
1. TB Report: o aplicatie care contine doar date exacte legate de evolutia afectiunii



2. TB eReview: DRTB : o aplicatie construita pentru o invatare interactiva in diagnosticarea si managementul tuberculozei rezistente la medicamente care doar prezinta informatii.



3. TB eReview: DRTB : o aplicatie construita pentru o invatare interactiva in diagnosticarea si managementul tuberculozei cu HIV care doar prezinta informatii.



4. SNTC



Stay up to date with the latest TB news, Training, Webinars, and Products from the Southeastern National Tuberculosis Center.

View our products, even when you are not connected to the internet.

Order products such as calipers and country guides right from the app.

Sign up for courses, view course locations on embedded maps, and add the course to your iPhone calendar with the click of a button.

View our archived webinars.

Access the online LTBI Calculator tool developed by Dr. Dick Menzies.

Search for content across hundreds of our resources and dozens of products.

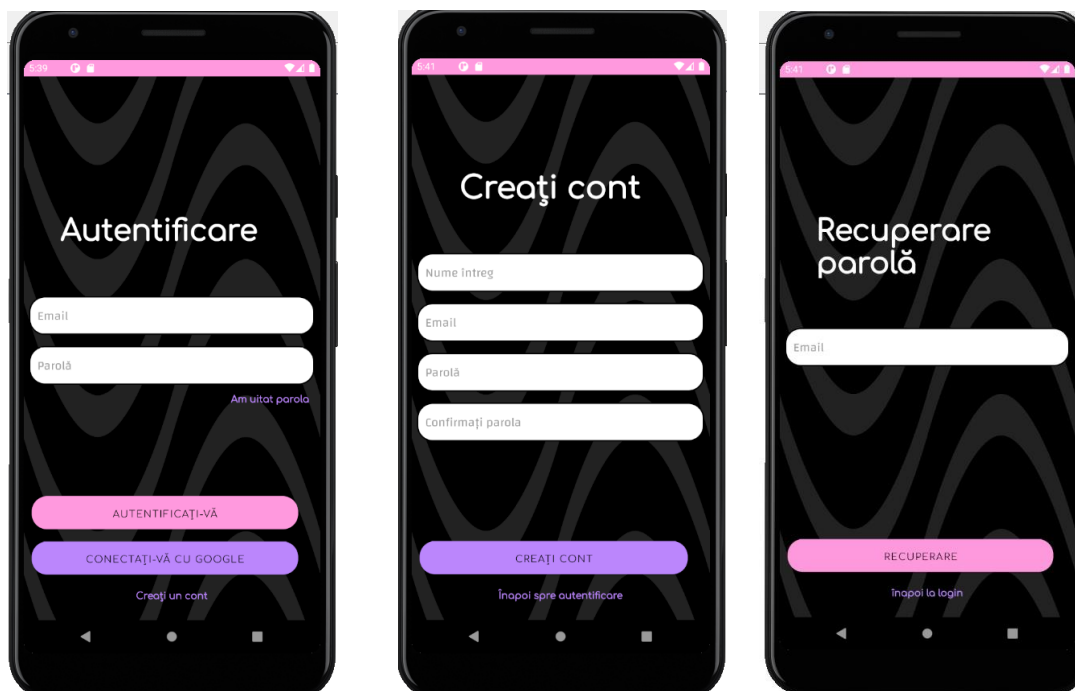
Contribuțiile autorului – declarația de originalitate

Aplicația va include idei (soluții) care sunt realizate deja de alte aplicații similare dar va include și contribuții proprii ale autorului. Pe lângă aplicațiile prezentate mai sus și celelalte prezente pe piață aplicația dezvoltată de mine va oferi și un suport pentru organizarea și urmărirea evoluției pe lângă doar prezentarea de informații exacte pe această temă. Din contribuțiile proprii ale autorului fac parte și front end-ul aplicației și construirea acesteia în limba română având în vedere că toate aplicațiile deja prezente sunt construite majoritatea în engleză fără opțiunea de a selecta și limba română.

Manualul utilizatorului

Această secțiune conține instrucțiuni referitoare la utilizarea aplicației din perspectiva unui utilizator comun.

La deschiderea aplicației pentru prima dată va fi afișată pagina de Autentificare de unde utilizatorul poate naviga prin paginile de creare cont sau „Am uitat parola”.



După ce creează un cont, sau se autentifică în unul deja existent utilizatorul este redirectionat către activitatea principală.



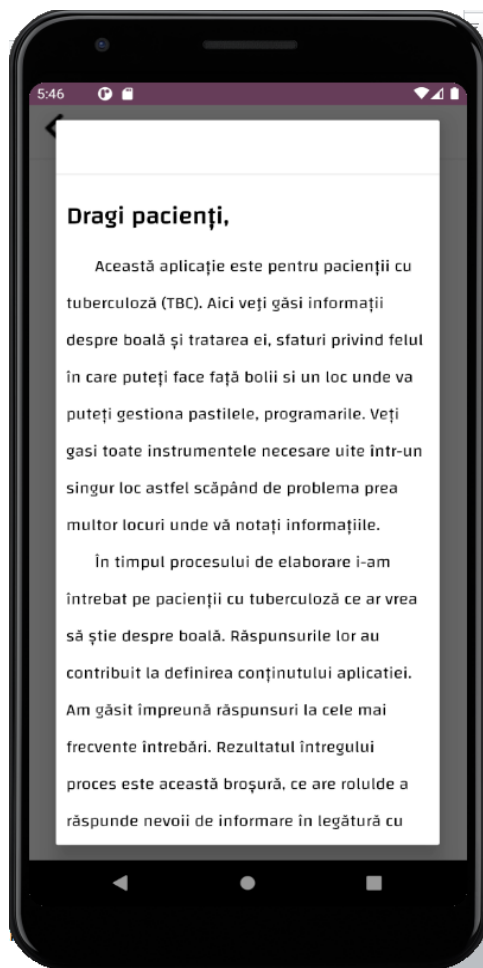
De aici utilizatorul poate alege din camourile prezentate prin click asupra pozei, iar aplicatia va afisa informatii despre capitolul selectat:



Sau poate accesa meniul:



In meniu, prin accesarea butonului „Despre noi” aplicatia afiseaza o pagina care contine informatii despre aplicatie, de ce a fost creata:



Prin accesarea butonului pentru HelpLine utilizatorul este directionat catre crearea unui apel catre o linie de ajutor impotriva anxietatii deschisa 24/24.

Prin accesarea butonului de „Cititi broșura detaliata” utilizatorul este redirectionat catre o broșura sigura, care contine informatii detaliate.

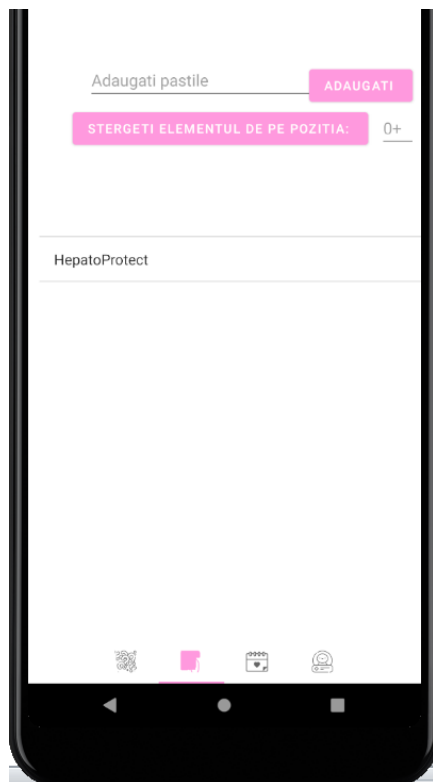
Prin accesarea butonului de „Servicii de consiliere online” utilizatorul este redirectionat catre un site unde sunt oferite servicii de terapie gratuite pentru ca pe langa informatii si management aplicatia isi propune sa usureze viata pacientilor si sa le ofere un spatiu sigur de unde pot prelua informatii si sa le ofere si un suport pentru sanatatea mentala.

Din pagina de home utilizatorul poate naviga cu ajutorul butoanelor din josul paginii:



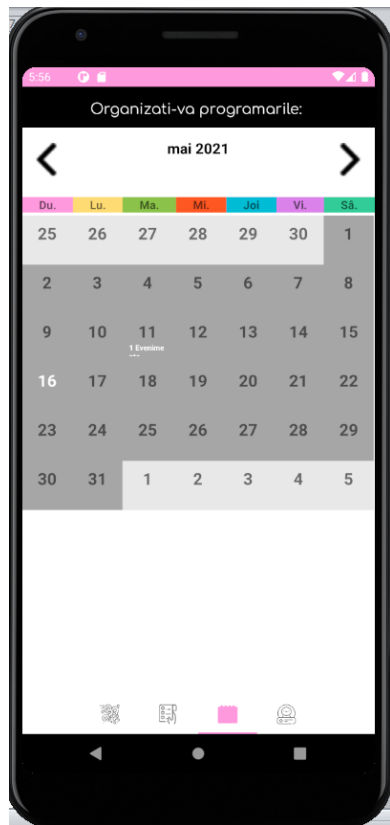
De aici poate ajunge la Lista de medicamente, la calendarul care stocheaza evenimente si la profil.

Lista de medicamente:



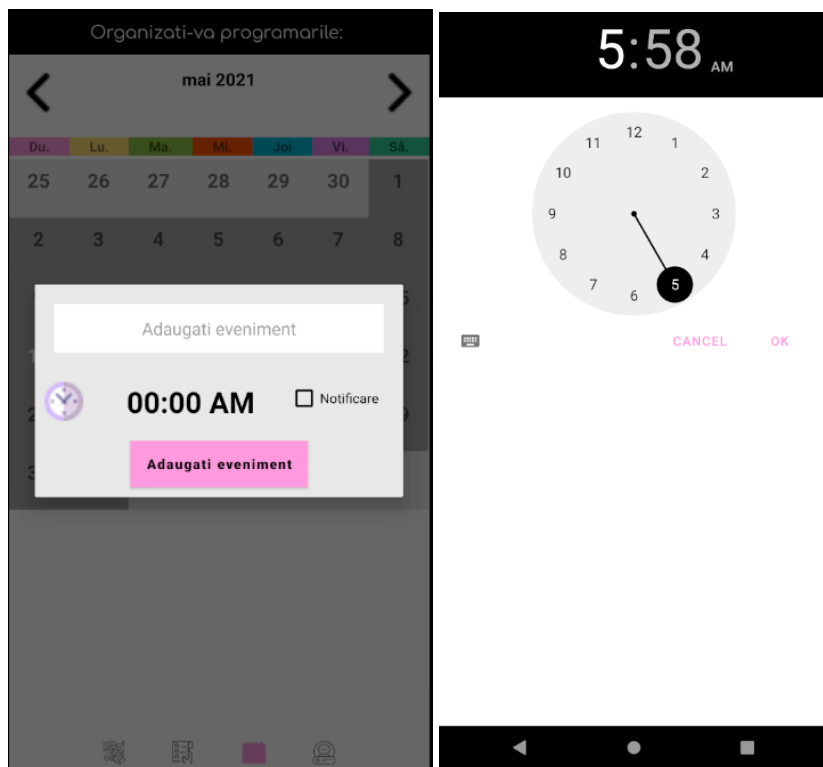
Utilizatorul poate adauga un nume de medicament, poate sterge un element din lista prin introducerea pozitiei acestuia (Atentie! Numaratoarea incepe de la 0 fapt semnalat si prin placeholder-ul „0+”).

Calendarul:



Aici utilizatorul poate naviga prin luni;

Poate adauga un eveniment:



Daca este selectat butonul "Notificare" atunci utilizatorul va primi o notificare pe telefon.

Poate vedea evenimentele:



De aici poate de asemenea sa opteze pentru notificare sau nu si poate sterge evenimentul.

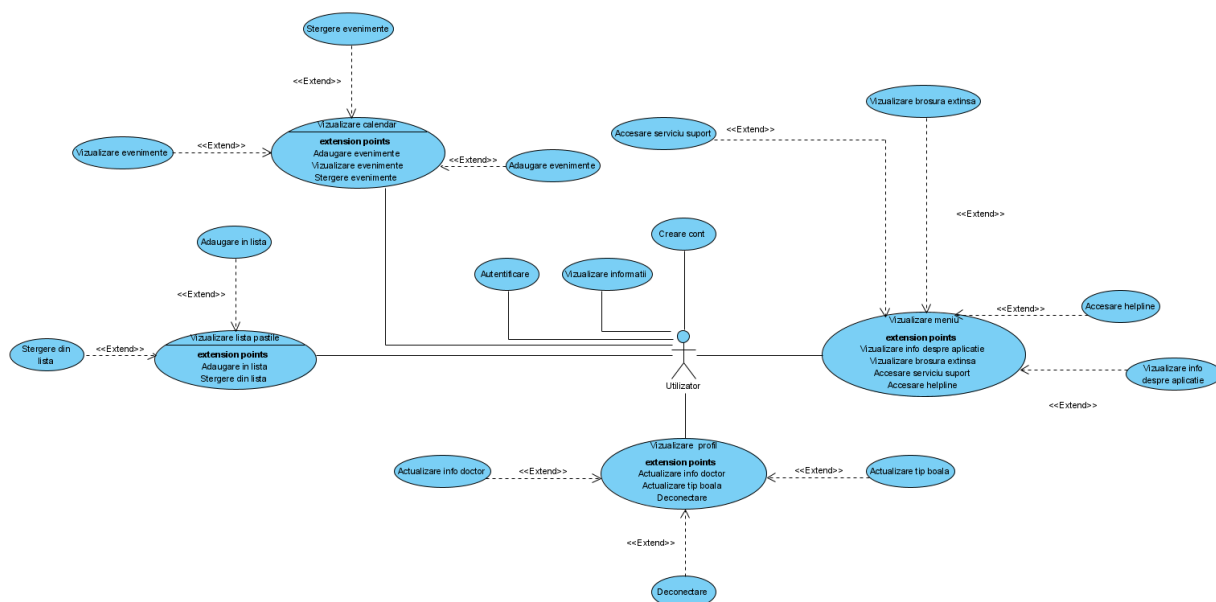
Profilul:



De aici utilizatorul se poate deconecta si poate edita campurile de Tip si Detalii doctor pentru a stoca informatiile pe care le doreste prin accesarea butonului Salvati. Accesarea butonului de deconectare va trimite spre pagina de autentificare.

Funcționalitate și structura aplicație

Aplicația va fi realizată folosind Android Studio și se va folosi limbajul Java precum și alte tehnologii adiacente (de exemplu limbajul de marcare XML). Aplicația va utiliza mai multe pagini și va fi construită astfel încât să fie ușor de folosit. La deschidere va apărea pagina de înregistrare și crearea cont. Utilizatorul va putea să își creeze un cont personalizat iar meniul va fi organizat pe categorii (ex.: informații, calendar, pastile, etc.). Diagrama cazurilor de utilizare:



Baza de date: pentru ca aplicația să poată fi folosită la modul real va fi necesară o bază de date pentru a reține și a centraliza toate datele colectate. S-a folosit SQLite și Firebase. Se vor crea tabele pentru utilizatori, pentru datele acestora.

Proiectul conține 25 de clase dintre care 9 sunt folosite pentru activități.

[illegible]

Aplicatia se deschide cu SplashActivity:

```
public class SplashActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_splash);

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

        new Handler().postDelayed(new Runnable() {
            @Override
            public void run() {
                if(user == null){
                    startActivity(new Intent( packageContext: SplashActivity.this,FragmentReplacerActivity.class));
                }else{
                    startActivity(new Intent( packageContext: SplashActivity.this,MainActivity.class));
                }
                finish();
            }
        }, delayMillis: 2500);
    }
}
```

Clasa preia userul curent din baza de date si verifica daca este logat sau nu. Daca este logat atunci il redirectioneaza catre MainActivity, altfel spre FragmentReplacerActivity.

FragmentReplacerActivity:

```
public class FragmentReplacerActivity extends AppCompatActivity {

    private FrameLayout frameLayout;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_fragment_replacer);

        frameLayout = findViewById(R.id.frameLayout);

        setFragment(new LoginFragment());
    }

    public void setFragment(Fragment fragment){

        FragmentTransaction fragmentTransaction= getSupportFragmentManager().beginTransaction();
        fragmentTransaction.setCustomAnimations(android.R.anim.slide_in_left,android.R.anim.slide_out_right);

        if(fragment instanceof CreateAccountFragment){
            fragmentTransaction.addToBackStack(null);
        }

        fragmentTransaction.replace(frameLayout.getId(), fragment);
        fragmentTransaction.commit();
    }
}
```

Clasa este folosita pentru a realiza actiuni cu fragmente ca raspuns la interactiunea cu utilizatorul si seteaza animatiile pentru fragmentele care intra sau ies din tranzactie. Clasa seteaza fragmentul pentru LoginFragment(), iar utilizatorul va fi redirectionat catre pagina de autentificare.

LoginFragment:

```
public class LoginFragment extends Fragment {

    private static final int RC_SIGN_IN = 1;
    private EditText emailET, parolaET;
    private TextView signUpTV, forgotTV;
    private Button loginBtn, GoogleSignInBtn;
    private ProgressBar progressBar;
    private FirebaseAuth auth;

    public static final String EMAIL_REGEX = "^(\\w-\\.){1,64}@([\\w&[^\"]+){2,255}\\.([a-z]){2,}$";
    private GoogleSignInClient mGoogleSignInClient;

    public LoginFragment() {
        // Required empty public constructor
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        return inflater.inflate(R.layout.fragment_login, container, attachToRoot: false);
    }
}
```

La crearea fragmentului va fi afisat layout-ul fragment_login.

mGoogleSignInClient este folosit pentru a loga un utilizator cu ajutorul contului sau google printr-un api de sing in de la Google:

```
GoogleSignInOptions gso = new
GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT_SIGN_IN)
    .requestIdToken(getString(R.string.default_web_client_id))
    .requestEmail()
    .build();
mGoogleSignInClient = GoogleSignIn.getClient(getContext(), gso);
```

Metoda clickListener() adauga actiuni asupra butoanelor de login, parola uitat, Google sing in si pentru cel de redirectare catre pagina de creare cont.

```

loginBtn.setOnClickListener(new View.OnClickListener(){

    public void onClick(View v) {
        String email =emailET.getText().toString();
        String parola =parolaET.getText().toString();

        if(email.isEmpty() || !email.matches(EMAIL_REGEX)){
            emailET.setError("Va rugam introduceți un email valid");
            return;
        }
        if(parola.isEmpty() || parola.length()<6){
            parolaET.setError("Va rugam introduceți o parolă validă de 6 caractere");
            return;
        }
        progressBar.setVisibility(View.VISIBLE);
        auth.signInWithEmailAndPassword(email,parola)
            .addOnCompleteListener(new OnCompleteListener<AuthResult>() {
                @Override
                public void onComplete(@NonNull Task<AuthResult> task) {
                    if(task.isSuccessful()){
                        FirebaseUser user=auth.getCurrentUser();
                        if(!user.isEmailVerified()){
                            Toast.makeText(getContext(), text: "Va rugam verificati email-ul",Toast.LENGTH
                        }

                        sendUserToMainActivity();

                    }else{
                        String exception="Error: "+task.getException().getMessage();
                        Toast.makeText(getContext(),exception,Toast.LENGTH_SHORT).show();
                        progressBar.setVisibility(View.GONE);
                    }
                }
            })
    }
});

```

Metoda preia din casetele de EditText din xml email-ul si parola si verifica daca sunt goale. Daca sunt atunci se afiseaza un mesaj de eroare si utilizatorul trebuie sa introduca iar datele. Pentru email verifica daca are o structura valida de tipul {litere si cifre,_}@{litere si cifre,-,_}.{ litere si cifre }, iar pentru parola daca lungimea este cel putin 6.

Atunci cand este apasat butonul de login vizibilitatea progressBar-ului este setata sa apara. Daca email-ul si parola sunt corecte atunci se va incerca conectarea utilizatorului. Daca conectarea este reusita, utilizatorul este redirectat catre pagina principal, altfel va afisa un mesaj de eroare.

Asemenea functioneaza si clasa CreateAccountFragment.

MainActivity:

Clasa realizeaza o gestiune a tab-urilor selectate. Seteaza icon-urile pentru taburile selectate si cele neselectate.

Realizeaza acest lucru cu ajutorul clasei ViewPagerAdapter

```
private void addTabs() {
    tabLayout.addTab(tabLayout.newTab().setIcon(R.drawable.information));
    tabLayout.addTab(tabLayout.newTab().setIcon(R.drawable.to_do_list));
    tabLayout.addTab(tabLayout.newTab().setIcon(R.drawable.calendar));
    tabLayout.addTab(tabLayout.newTab().setIcon(R.drawable.profile));
    tabLayout.setTabGravity(TabLayout.GRAVITY_CENTER);
    tabLayout.setTabMode(TabLayout.MODE_SCROLLABLE);
    pagerAdapter = new ViewPagerAdapter(getSupportFragmentManager(), tabLayout.getTabCount());
    viewPager.setAdapter(pagerAdapter);
    viewPager.addOnPageChangeListener(new TabLayout.TabLayoutOnPageChangeListener(tabLayout));
    tabLayout.getTabAt(index: 0).setIcon(R.drawable.information_fill);

    tabLayout.addOnTabSelectedListener(new TabLayout.OnTabSelectedListener() {
        @Override
        public void onTabSelected(TabLayout.Tab tab) {
            viewPager.setCurrentItem(tab.getPosition());

            switch (tab.getPosition()){
                case 0:
                    tabLayout.getTabAt(index: 0).setIcon(R.drawable.information_fill);
                    break;
                case 1:
                    tabLayout.getTabAt(index: 1).setIcon(R.drawable.to_do_list_fill);
                    break;
                case 2:
                    tabLayout.getTabAt(index: 2).setIcon(R.drawable.calendar_fill);
                    break;
                case 3:
                    tabLayout.getTabAt(index: 3).setIcon(R.drawable.profile_fill);
                    break;
            }
        }
    })
}
```

```

public class ViewPagerAdapter extends FragmentStatePagerAdapter {
    int noOfTabs;

    public ViewPagerAdapter(@NonNull FragmentManager fm, int noOfTabs) {
        super(fm);
        this.noOfTabs = noOfTabs;
    }

    @NonNull
    @Override
    public Fragment getItem(int position) {

        switch (position){
            case 0:
                return new Home();
            case 1:
                return new Pastile();
            case 2:
                return new Calendar();
            case 3 :
                return new Profil();
            default:
                return null;
        }
    }

    @Override
    public int getCount() { return noOfTabs; }
}

```

In functie de tab-ul selectat returneaza fragmentul corespunzator.

Home.java:

La crearea view-ului se va incarca layout-ul fragment_home.

Metoda init(view) preia din xml butoanele necesare si creaza pentru fiecare cate un ClickListener care va redirecta activitatea catre una noua. Pentru p1 va fi redirectata catre P1Activity care contine doar un toolbar cu titlul si un buton de back si un scrollView cu text. Asemenea si pentru p2,p3,p4 si p5. Butonul de meniu redirecteaza catre clasa MenuActivity.

```
private void init(View view) {
    androidx.appcompat.widget.Toolbar toolbar=view.findViewById(R.id.toolbar1);
    p1= view.findViewById(R.id.p1);
    p2= view.findViewById(R.id.p2);
    p3= view.findViewById(R.id.p3);
    p4= view.findViewById(R.id.p4);
    p5= view.findViewById(R.id.p5);
    menuBtn=view.findViewById(R.id.menuBtn);
    if(getActivity() != null)
        ((AppCompatActivity) getActivity()).setSupportActionBar(toolbar);
    menuBtn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent intent = new Intent(getContext(), MenuActivity.class);
            startActivity(intent);
        }
    });
    p1.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) { openSelectFunction(); }
    });
    p2.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent intent = new Intent(getContext(), com.example.myapplication.p2.class);
            startActivity(intent);
        }
    });
    p3.setOnClickListener(new View.OnClickListener() {
```

MenuActivity:

La crearea activitatii se vor incarca toate informatiile. Activitatea are 5 butoane: unul de back care duce inapoi spre pagina de home, unul pentru informatii care afiseaza un alertDialog cu layout-ul „despre_noi”, unul pentru un helpline care redirecteaza utilizatorul catre pagina de formare denumar de telefon al telefonului, unul care redirecteaza utilizatorul catre un pdf care contine o brosură extinsa si ultimul care redirecteaza catre un site web.

PastileView:

Clasa preia din baza de date Firebase din colectia Pastile lista cu pastile stocate:

```
userRef= FirebaseFirestore.getInstance().collection("Pastile")  
        .document("zbEveRYQL1Ma7Js0bBfv");
```

Aceasta lista vine sub forma unui string pe care metoda onCreate() il separa dupa spatiu, iar fiecare element este pus intr-un array si fiecare element al listei este afisat sub forma „simple_list_item”.

La apasarea butonului adauga se preia textul din EditText-ul et si la stringul preluat din baza de data se adauga noul string. Apoi este incarcat in baza de date noul set de data cu rezultatul care va da update si view-ul cu noua lista.

Asemenea si la butonul de stergere. Acesta apeleaza functia removeItem care primeste ca parametru pozitia introdusa in EditText, itereaza prin lista si pune intr-un rezultat toate elementele care nu se afla pe pozitia data ca parametru apoi incarca in baza de date noul rezultat.

Concluzii și direcții viitoare

ToBeCareful este o aplicație folosită pentru a gestiona toate aspectele unei boli, de la aspecte organizatorice(pastile, programari), la informare si suport mental.

Aplicația a fost dezvoltată folosind limbajul de programare Java și mediul de dezvoltare Android Studio. Ea poate fi folosită pe telefoanele (smartphones) cu sistem de operare Android, minim versiunea 8 (Android Oreo).

Aplicatia se adreseaza in special persoanelor afectate de tuberculoza, dar si oricui care doar vrea sa se documenteze in legatura cu afectiunea sau cum sa o previna.

Directii viitoare:

Aplicatia este intr-o continua dezvoltare tocmai pentru ca are ca scop sa se axeze pe nevoile utilizatorilor.

Posibile imbunatatiri:

- Utilizatorul sa poata adauga detalii despre pastile si memento-uri direct in lista.
- Calendarul sa poata realiza un countdown pana la terminarea tratamentului sau pana la programari.

Referinte

[<http://www.tuberculoza.info/>]

[<https://developer.android.com/reference/java/util/Calendar>]

[<https://www.lhl.no/globalassets/lhl-internasional/dokumenter/tuberculosis-brochure-tub-rumensk--print2017.pdf>]

[<https://www.youtube.com/watch?v=8pWmWCEaHpM>]