Documentația proiectului CovidTest

Spătar Ioana-Diana

Informatică

Grupa 283

Contents

[Prezentare 3](#_Toc50668204)

[Tehnologii folosite 7](#_Toc50668205)

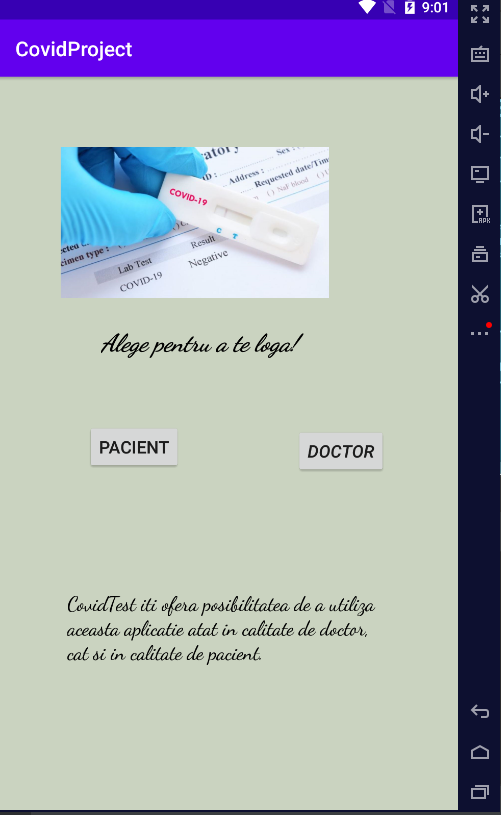
[Backend 7](#_Toc50668206)

[Arhitectura proiectului 9](#_Toc50668207)

[Concluzie 14](#_Toc50668208)

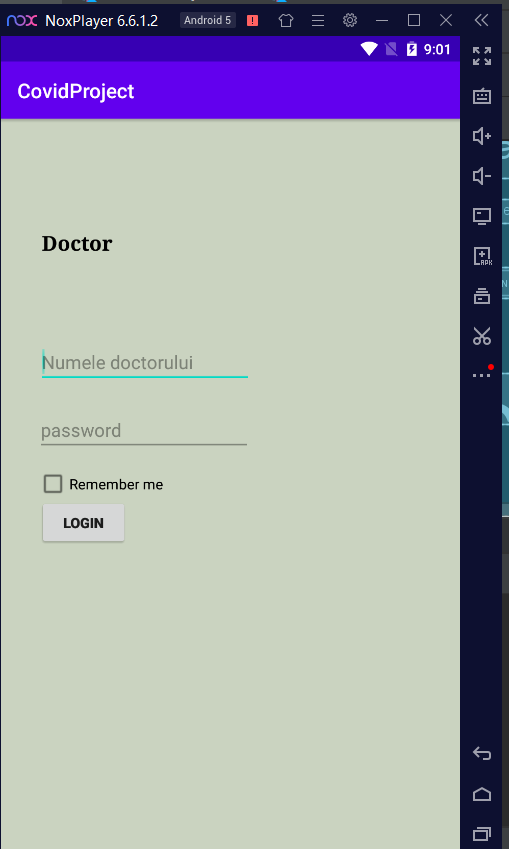
# Prezentare

Aplicația CovidTest este o aplicație simplă și ușor de înteles. Este realizată în Android Studio, cu un aspect simplist pentru care am ales o culoare verde deschis ca tematică, culoarea fiind placută văzului și simbolizând sănătatea, aplicația fiind una cu tematică medicală. În momentul în care se deschide aplicația, utilizatorul își poate alege poziția în care se află, adică cea de pacient sau cea de doctor.



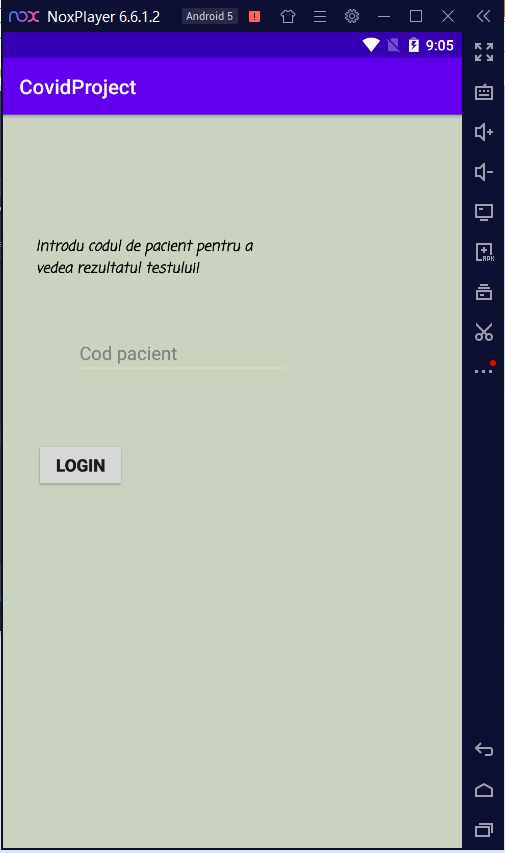
După ce utilizatorul alege una dintre cele două opțiuni, atunci se va deschide fila specifică alegerii sale.

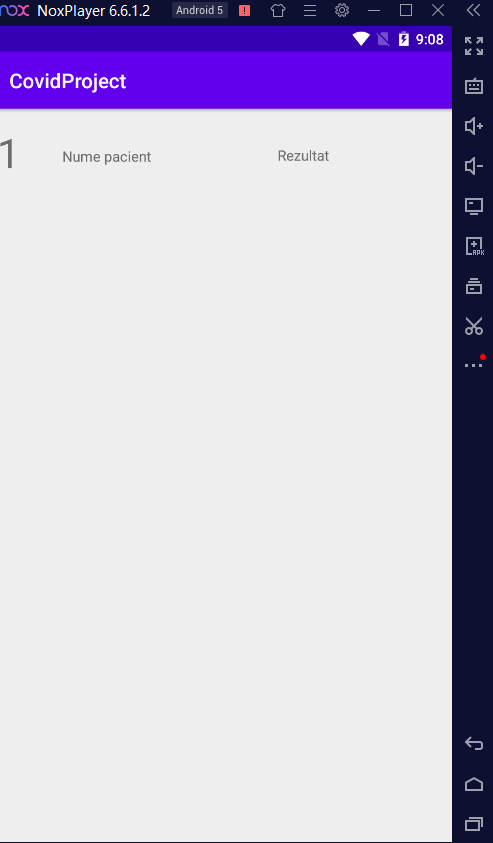
După autentificare daca utilizatorul este un doctor atunci se poate loga într-un cont pentru care trebuie sa furnizeze numele său și parola. De asemenea, dacă dorește să se conecteze ulterior poate bifa căsuța de remember me. Un doctor poate să adauge un rezultat pentru fiecare pacient în parte. Pentru a furniza acest rezultat trebuie să introducă datele pacientului. Datele pacientului se vor încărca prin intermediul unui fragment care va fi plasat în interiorul activității în care doctorul va introduce datele pacientului,



Dacă utilizatorul este un pacient care își așteaptă rezultatul pentru testul Covid atunci va apăsa butonul de pacient și va apărea activitatea în care acesta trebuie să își introducă codul personal care îi este asociat și prin intermediul căruia doctorul i-a putut stabili rezultatul.

După ce își introduce codul, va apăsa pe butonul de log in care îl va conduce spre rezultat.





În orice moment atât pacientul, cât și doctorul se pot întoarce în pagina anterioară dacă au apăsat butonul greșit.

Această aplicație a fost construită în ideea de a îmbunătăți modul de transmitere a acestor rezultate ale testului de covid și restricționează adunările și cozile imense din cadrul sălilor de așteptare, folosind o bază de date în care să fie introduși toți cei care au efectuat acest test.

# Tehnologii folosite

Aplicația CovidTest a fost realizată în Android Studio, utlizând limbajul de programare Kotlin.

Android Studio, sistemul integrat pentru development pentru aplicațiile pentru Android oferă o mulțime de instrumente de editare, simulatoare pentru testarea aplicației tale. Mai mult, acesta îți arată cum va funcționa aplicația ta pe diferite versiuni de Android.

Am ales acest limbaj de programare deoarece este un limbaj nou folosit în facultate și este atât interesant, cât și ușor de înțeles. Kotlin este un limbaj de programare folosit pentru producerea și dezvoltarea unui număr tot mai mare de aplicații software și pentru implementarea acestora în diferinte medii multiplatformă. Kotlin își are proveniența din limbajul de programare Java care este unul dintre cele mai utilizate limbaje de programare și evident, există o oarecare asemanare între cele două.

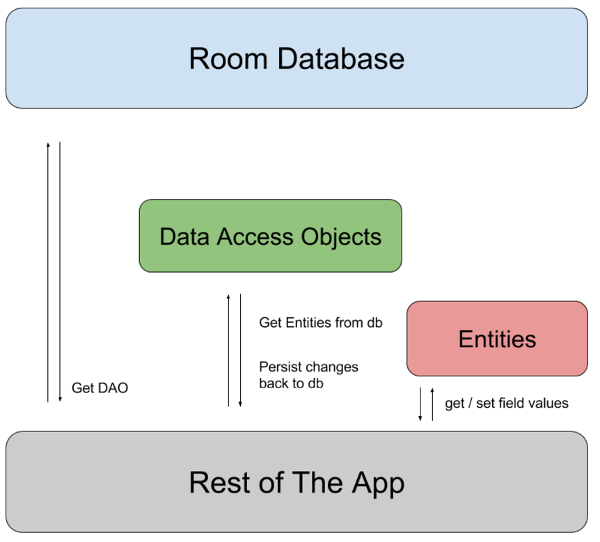
Față de limbajul Java care este mult mai greoi și presupune scrierea și urmărirea mai multor linii de cod, Kotlin este mult mai ușor de învățat și nu necesită niciun fel de experiență în mobile development. După ce a fost anunțat ca limbajul oficial de programare pentru aplicații pentru Android, acum 2 ani, în cadrul conferinței anuale a dezvoltatorilor organizată de gigantul Google, Kotlin a avut o creștere cu adevărat spectaculoasă. Kotlin este un limbaj de programare special creat pentru Android și vine cu o mulțime de caracteristici și utilități care le ușurează munca developerilor.

# Backend

Aplicația începe deschizând o pagină destinată autentificării utilizatorilor, în care aceștia trebuie să își aleagă poziția în care se află, adică cea de pacient sau cea de doctor. Dacă utilizatorul alege butonul de pacient, atunci i se va deschide o pagină în care trebuie să își introducă codul unic de pacient după care îl va putea identifica doctorul. Dacă pacientul nu introduce un cod atunci îi va apărea un warning care îl anunță ca este gol câmpul și că trebuie introdusă valoare. După ce clientul apasă butonul de log in atunci îi va apărea pagina în care i se va afișa numele și rezultatul care i-a fost specificat, adică introdus de doctor. Datele clientului sunt introduse de doctor și salvate în baza de date.

În cazul în care utilizatorul este doctor atunci va trebui să își specifice numele și parola, care pot fi reținute dupa bifarea checkboxului de remember me. După ce se va loga i se deschide pagina care conține o pagina în care îi va aparea numele său și un fragment care îi oferă posibilitatea de a introduce în baza de date a pacientului cu rezultatul său. Numele doctorului se va salva cu ajutorul shared preferences. Rezultatele pacientilor vor fi stocate într-un recycle view.

Pentru baza de date am ales sa utilizez ROOM DATABASE, care utlizează 3 mari clase: RoomDatabase, DAO, Entitate și eu am ales să folosesc repository și view model.

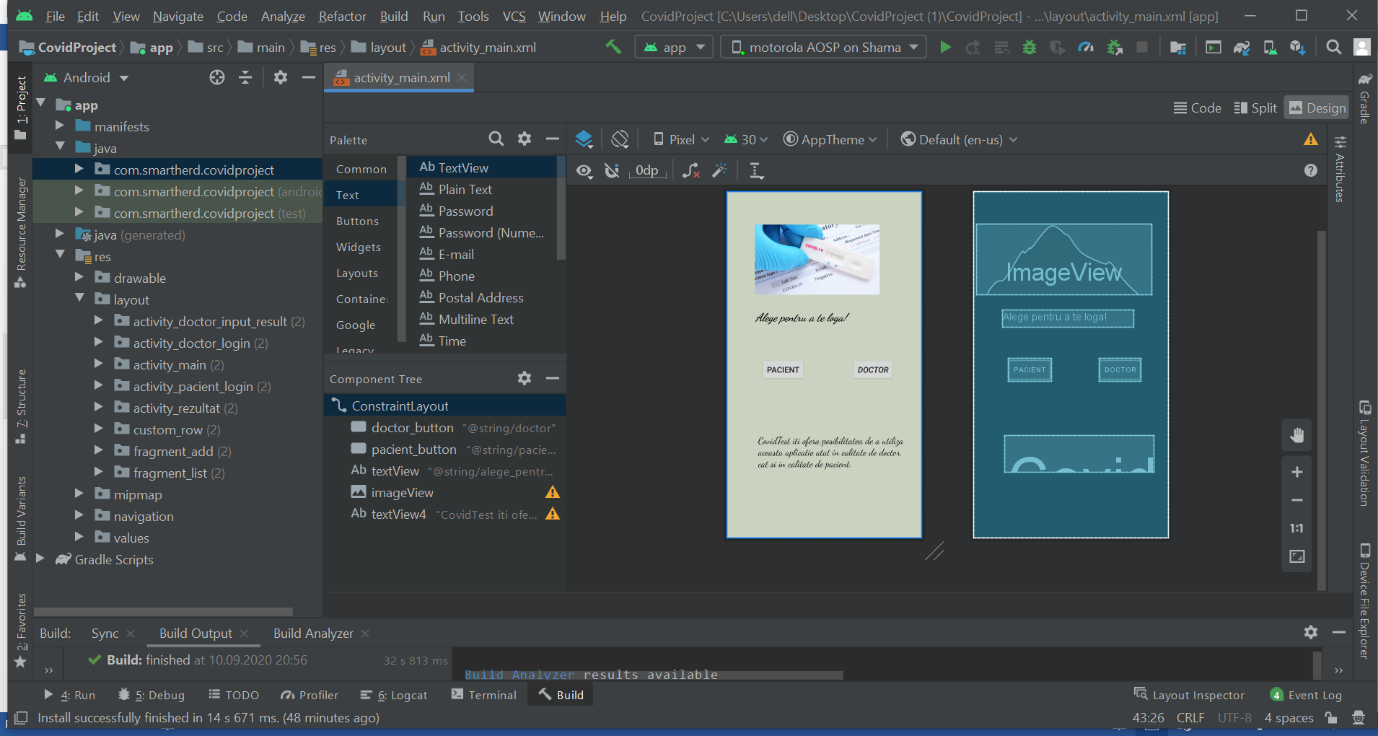


# Arhitectura proiectului

Proiectul este organizat în activități și fragmente

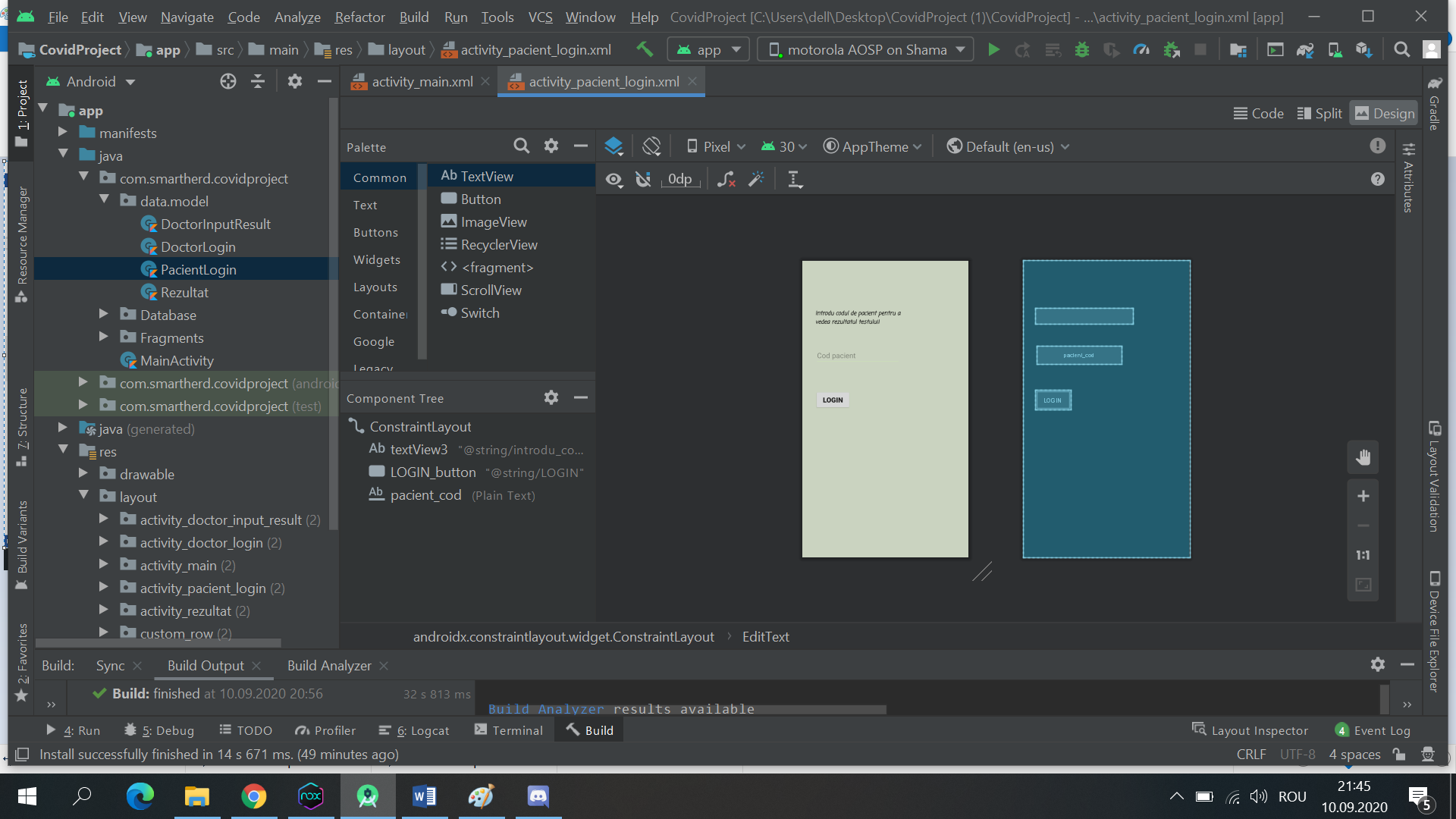
Prima activitatea, MainActivity

Acest layout conține un ImageView în care am încărcat o poză specifică programului, un text view care informează utilizatorul că trebuie să apese un buton pentru a continua utilizarea aplicației, un text view care conține o scurtă prezentare și cele doua butoane pentru doctor și pentru pacient care deschid alte activități.



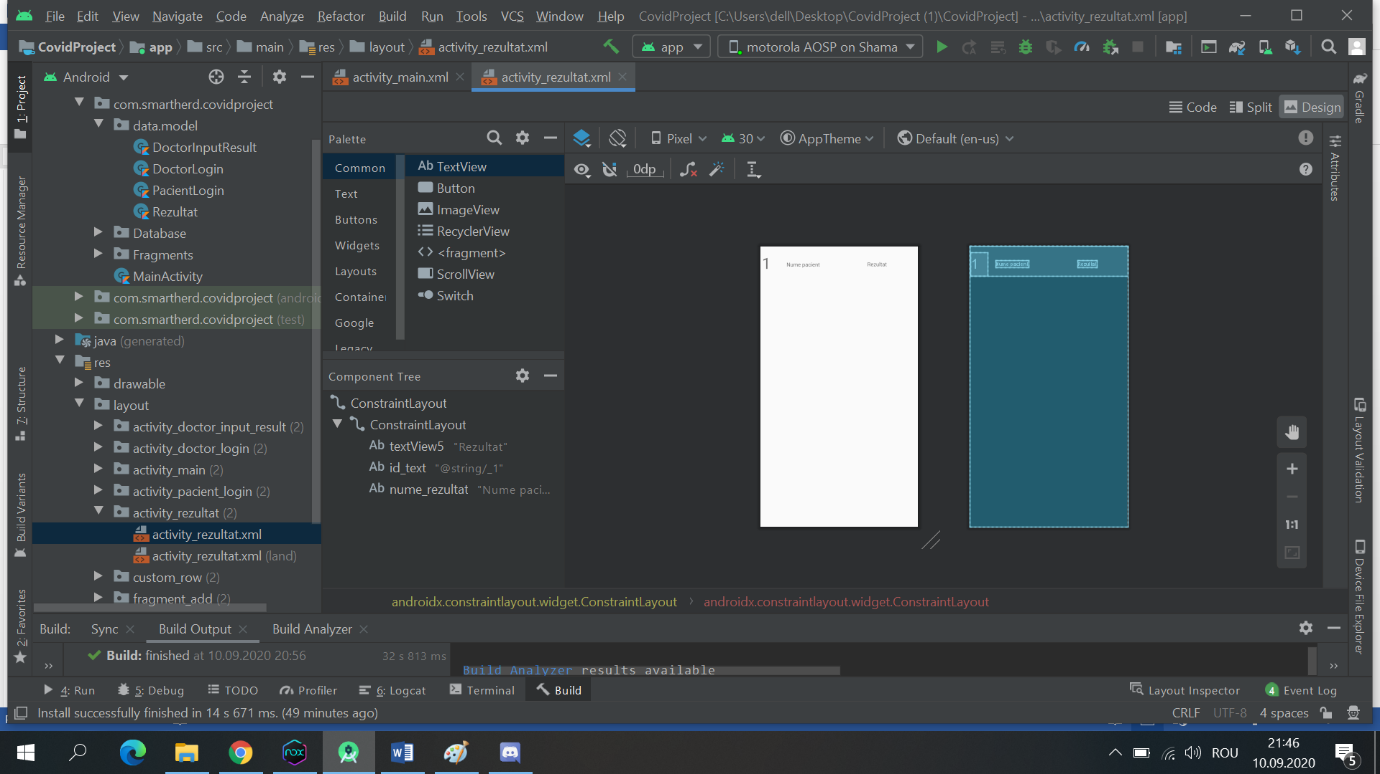
A doua activitate, PacientLogin

Acest layout conține un text view în care se menționează faptul că pacientul trebuie să introducă codul, un text view în care trebuie introdus acel cod de către bacient și un buton care îl conduce spre pagina cu rezultatul.



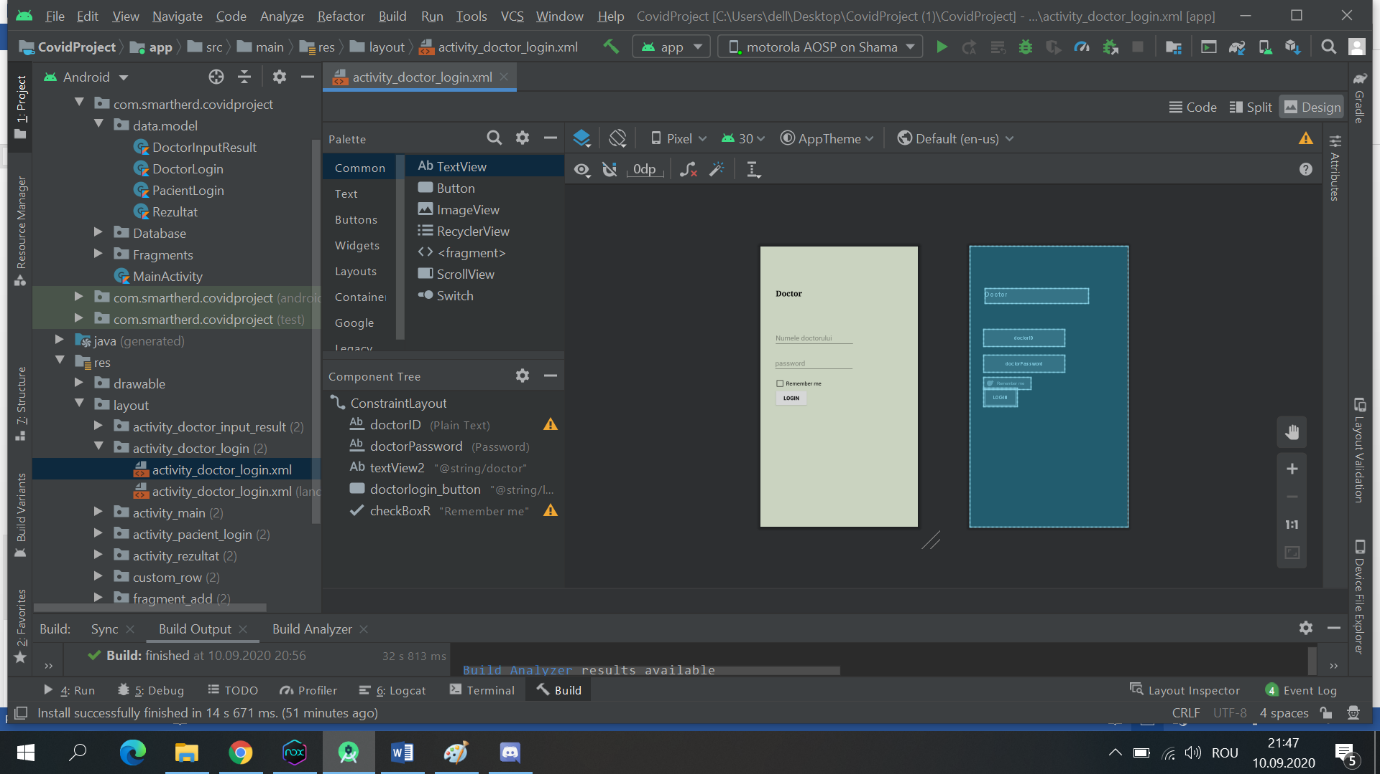
A treia activitate, Rezultat

Acest layout conține doua text view uri în care se vor scrie numele pacientului și rezultatul acestuia.



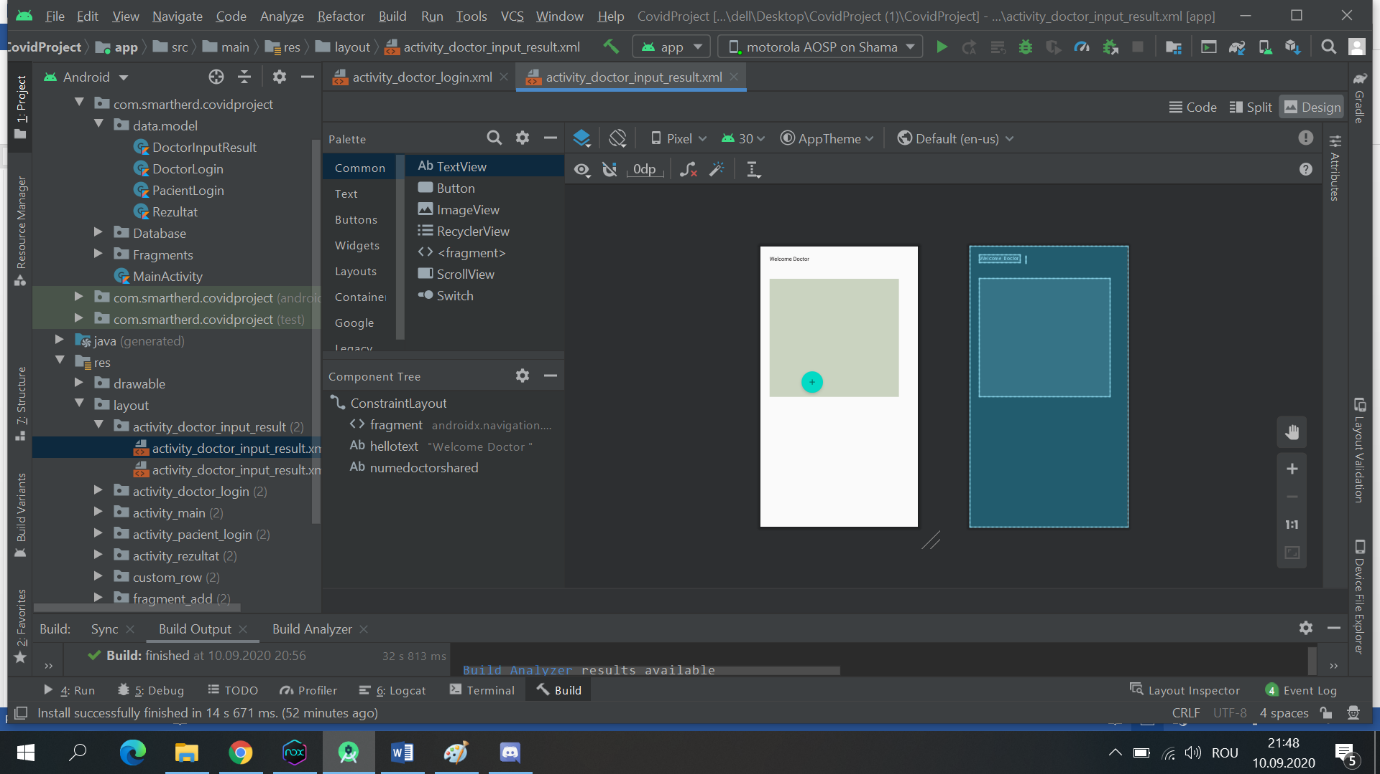
A patra activitate, DoctorLogin

Acest layout conține un text view in care se menționeaza că pagina este pentru doctor, un plain text in care trebuie utilizatorul să își introducă numele, un plain password pentru parola doctorului, un checkbox care ajută doctorul să i se memoreze datele furnizate și un buton care în conduce spre pagina unde va introduce datele pacientului.



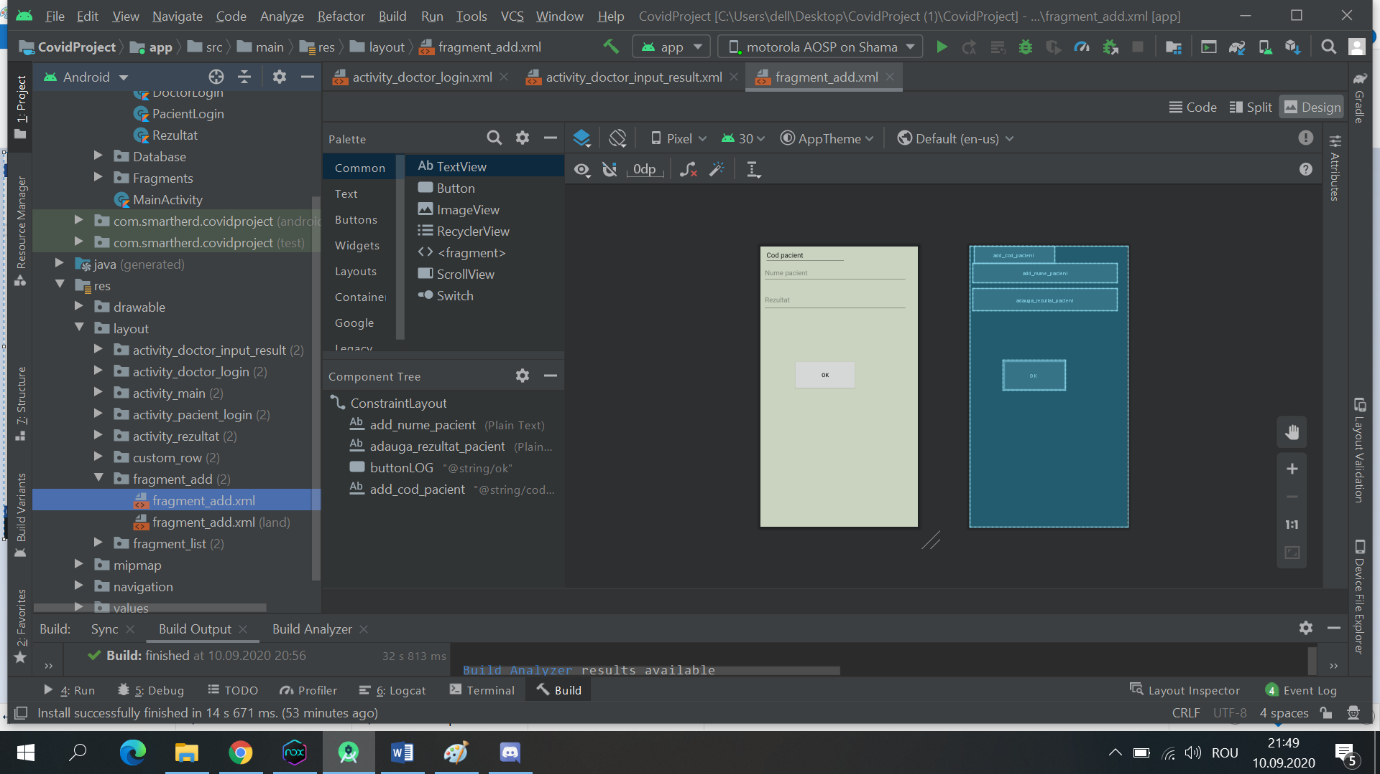
A 5-a activitate, DoctorInputResult

Acest layout conține doua text view pentru a saluta doctorul și în care se va scrie și numele doctorului salvat în shared preferences și fragmentul fragment\_add.



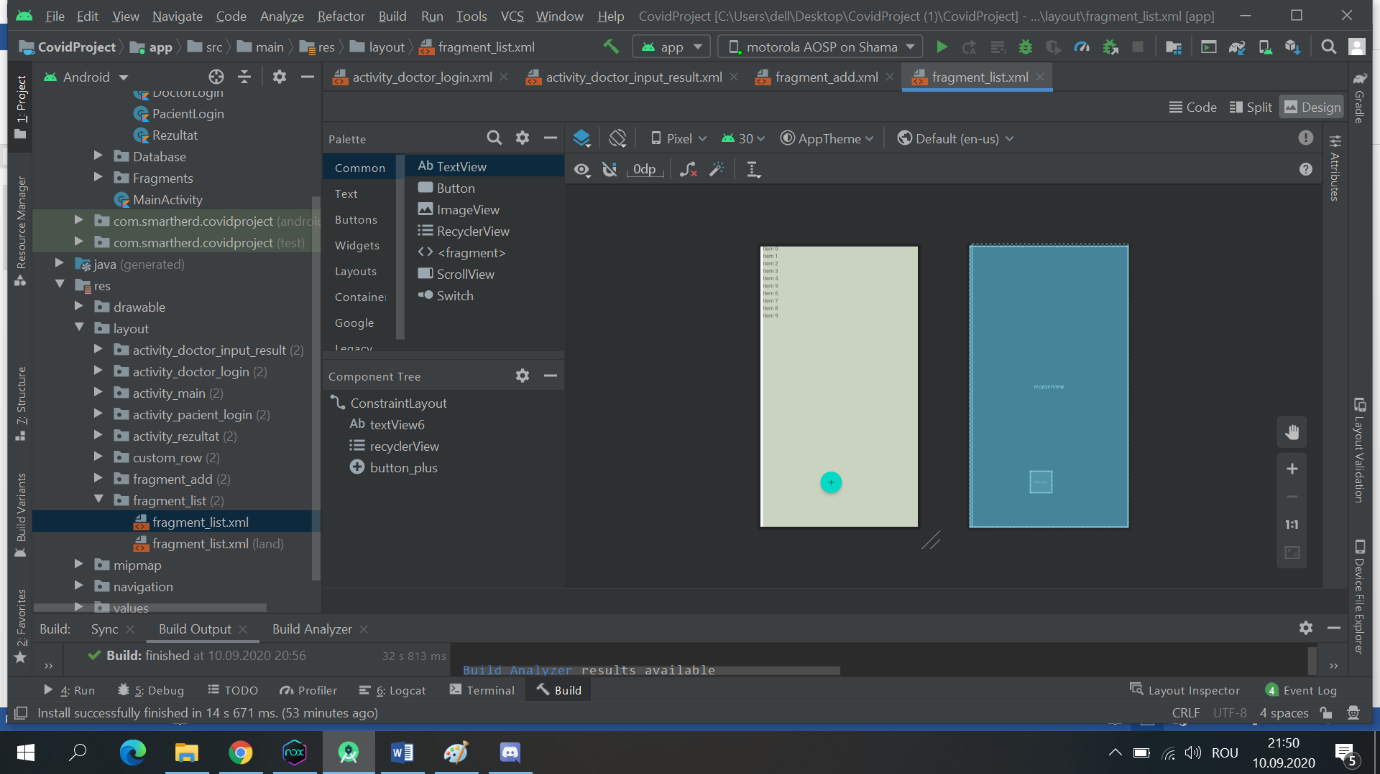
Fragment 1, fragment\_add

Acest fragment conține 3 edit text uri pentru datele pacientului și un buton de salvare prin care datele for fi stocate în baza de date.



Fragment 2, fragment\_list

Acest fragment conține un recycleview și un button pentru adăugare



# Concluzie

Proiectul a fost realizat de o singură persoană, Spătar Ioana-Diana, studentă în anul II la Informatică în cadrul Universității Transilavia, Facultatea de Matematică și Informatică, grupa 283.