Test de personalitate -Eneagrama

Disciplina : DATM

**Coordonator : Student :**

Dr.ing. Adriana Stan Florina Veres

An.II , TM

2017

**Introducere**

Acest proiect are ca tema realizarea unei aplicatii android ce determina personalitatea utilizatorului. Test de personalitate este bazat pe testul radid al Eneagramei, si permite utilizatorului ca doar in 5 minute sa determine cu o precizie destul de ridicata tipul psihologic caruia ii apartine.

Aplicatia propusa este rapida si usor de utilizat. Aceasta are la baza un design prietenos si o navigare facila. Intrucat cerintele hardware si software sunt destul de reduse, aplicatia poate fi utilizata pe o gama larga de dispozitive Android (smartphones, tablete …). Deoarece aplicatia este una simpla, timpul de raspuns al aplicatiei la diferite cerinte venite din partea utilizatorului si timpul de recuperare al sistemului este scurt.

Aplicatia poate rula pe diferite dispozitivele Android si a fost implementata in Android Studio 2.2.3, avand la dispozitie tool-urile Android SDK. Pentru instalarea aplicatiei mobile avem nevoie de urmatoarele cerinte :

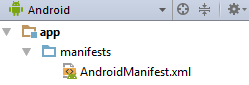
* Cerinte hardware :
* Un smartphone sau o tablet pe care sa fie instalat sistemul de operare Android;
* Cel putin 15MB de memorie fizica;
* Cerinte software
* Dispozitivul mobil trebuie sa aiba instalata minim versiunea de Android 4.1 (Jelly Bean);
* Pasii de instalare
* Fisierul .sdk al aplicatiei
* Un cablu de date USB pentru conexiunea dintre smartphone si calculator;

**Structura proiectului**

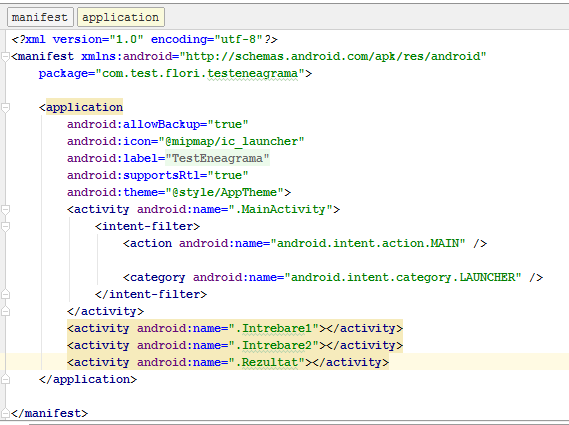
Un proiect Android are la baza un sistem de directoare ierarhizate, la fel ca si un proiect Java. Cu toate acestea, particularitatile sunt unice pentru Android. Pentru a crea o aplicatie avem nevoie de urmatoarele directoare/ fisiere:

* **src**– este directorul in care se gaseste codul Java aferent proiectului
* **gen** – in acest director, Android va stoca codul generat pentru rularea proiectului;
* **res**– este un director care contine resursele aplicatiei precum: elemente grafice, designul aplicatiei sau descrierea unor elemente din aplicatie

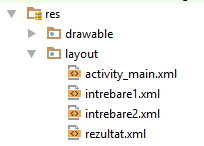
**AndroidManifest.xml***–*acest fisier XML contine descrierea aplicatiei, ce activitati, include sau ce permisiuni necesita aplicatia pentru a rula.



In cazul nostru aplicatia este compuse din 4 activitati: *.MainActivity*, *.Intrebare1*, *.Intrebare2* si *.Rezultat*.

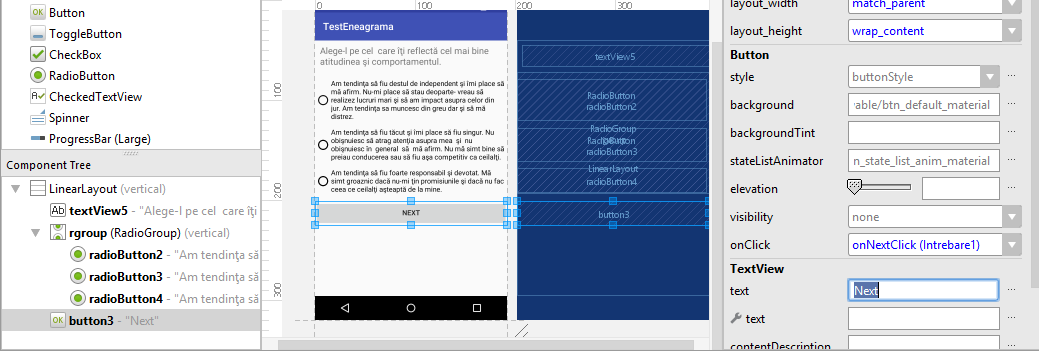


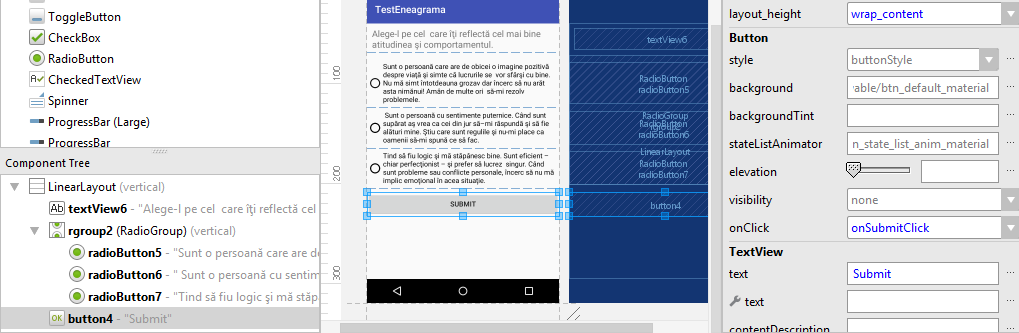
In sistemul Android fisierele XML sunt percepute ca fiind resurse ale aplicației si se gasesc in directorul *res/layout*in cadrul proiectului . Fiecare fişier de acest gen contine o structura ierarhica de widget-uri si containere care descriu modul in care va arata interfata aplicatiei cu utilizatorul.

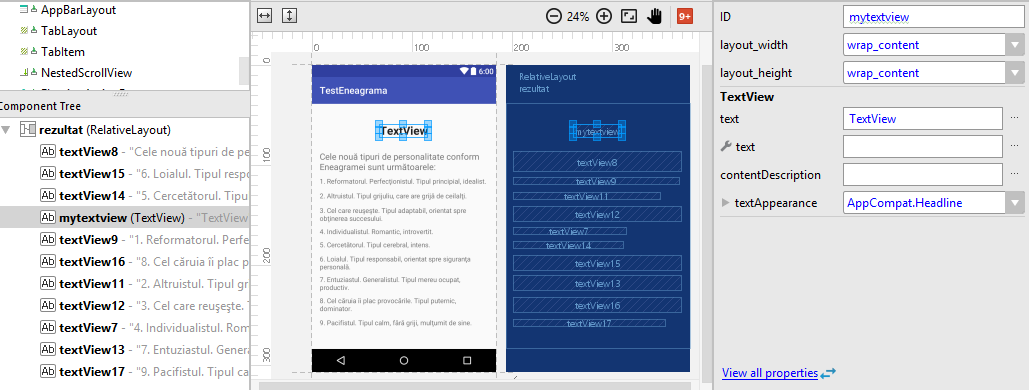


Designul si componentele celor 4 resurse xml poate fi observant in cele ce urmeaza.

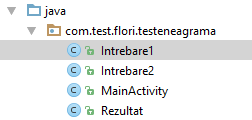


``



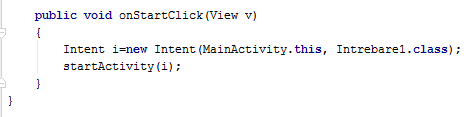


Functionalitatea aplicatiei este realizata cu ajutorul a 4 fisiere Java:

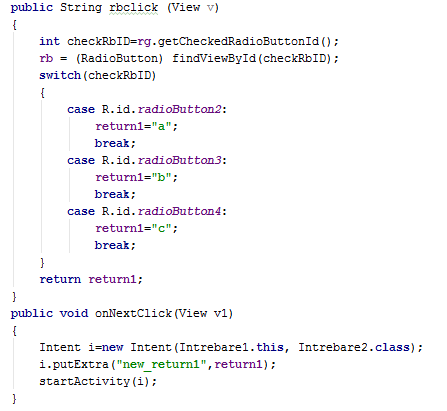


Una din cele mai importante functionalitati utilizate in aceste fisiera este trecere intre activitati ce se realizeaza cu ajutorul butoanelor corespunzatoare.

Astfel trecerea de la ecranul principal al aplicatiei (*MainActivity*) la ecranul ce contine prima intrebare a testului se realizeaza cu ajutorul apelarii unui *Intent* cu ajutorul proprietatii *onClick* a butonului *Start*.



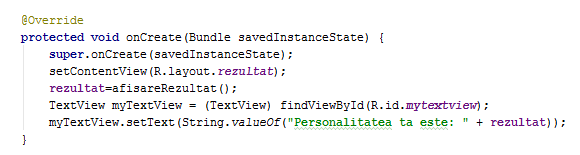
In cadrul fisierelor *Intrebare1.java* si *Intrebare2.java* tranzitia se realizeaza in mod similar insa se mai adauga un *extra* ce va putea fi accesat si manipulate mai apoi. Aceasta variabila de tip *string* este utilizata pentru determinarea raspunsului ales de utilizator la intrebarea respectiva.



In cazul celei de-a doua intrebare se poate observa ca este realizata o concatenare a raspunsului selectat la prima intrebare si raspunsul selectat la ce-a de-a doua, realizandu-se astfel codul personalitatii.

**public** String rbclick2 (View v)  
{  
 Bundle extras = getIntent().getExtras();  
 String return1=**null**;  
 **if** (extras != **null**)  
 {  
  
 String value = extras.getString(**"new\_return1"**);  
  
 return1 = value;  
 }  
  
 **int** checkRbID2=**rg2**.getCheckedRadioButtonId();  
 **rb2** = (RadioButton) findViewById(checkRbID2);  
 **switch**(checkRbID2)  
 {  
 **case** R.id.***radioButton5***:  
 **return2**=return1+**"x"**;  
 **break**;  
 **case** R.id.***radioButton6***:  
 **return2**=return1+**"y"**;  
 **break**;  
 **case** R.id.***radioButton7***:  
 **return2**=return1+**"z"**;  
 **break**;  
 }  
 **return return2**;  
}  
**public void** onSubmitClick(View v2)  
{  
 Intent i=**new** Intent(Intrebare2.**this**, Rezultat.**class**);  
 i.putExtra(**"rezultat"**,**return2**);  
 startActivity(i);  
}

In cadrul fisierului *Rezultat.java* este interpretat codul personalitatii obtinut si se afiseaza personalitatea corespunzatoare pe ecran impreuna cu interpretarile pentru toate cele 9 personalitati ale Eneagramei.



De asemenea trebuie sa mentionam ca utilizatorul are posibilitatea de a naviga inapoi cu ajutorul butonului de back al dispozitivilui. In cazul in care acesta isi va schimba raspunsul la una din intrebari rezultatul afisat va fi updatat conform schimbarilor.

