Facultatea de Electronica Telecomunicatii si Tehnnologii Informationale

Disciplina: Dezvoltare de Aplicatii Mobile

Tema:Mangementul generatiei de studenti

Cadru didactic coordonator:

Sl.dr.ing.Adriana Stan

Student master II TC:

Bețivu Denis

2018

Generatia unei liste de studenti este o aplicatie compusa dintr-o simpla baza de date si operatii asupra acesteia,Este destinata pentru a vizualiza toti studentii si caracteristiile fiecaruia , care permit managementul pe perioada invatamantului.

Tabela are urmatoarele attribute:

Nume,Prenume,CNP,An_studiu,Facultate, Varsta, Credite

In clasa databaseHelper am ppus metoda ce o creaza:

```
public void onCreate(SQLiteDatabase db) {
    String createTable = "CREATE TABLE " + TABLE_NAME + "
 (ID INTEGER PRIMARY KEY AUTOINCREMENT, " +
 " NUME TEXT, PRENUME TEXT,
 CNP char(13),//dau valoare fixa a campului ca o constrangere
 AN_STUDIU INT, FACULTATE TEXT char(3),
 VARSTA INT , CREDITE INT )";
        db.execSQL(createTable);
  }
```

Dup ce am creat baza de date mi-am propus sa realizez principalele operatii pe aceasta baza de date si anume de **Adaugare** a unui student, **Actualizare** info unui student **Stergere** a informatiei acestuia si **Afisarea** intregii grupe de studenti.

Afisarea listei se face pe 8 coloane s nu pe 7,o coloana suplimentara este ID care totodata da si numarul de student inscrisi.

Acest ID este cheie primara in tabela si ase autoincrementeaza la introducerea unei noi inregistrari in tabela,ceea ce usureza munca celui car eintroduce datele nu tebuie sa tina minte pe al catalea sudent il incrie momentan.

In fisierul Activity main am creat metoda care sa faca posibila inserarea n baza de date DataBasehelper.

```
public void AddData(String Nume, String Prenume, Integer CNP,
Integer An_studiu, String Facultate,Integer Varsta, Integer
Credite) {

        boolean insertData = myDB.addData(Nume, Prenume, CNP,
An_studiu, Facultate, Varsta, Credite);


        if (insertData) {

            Toast.makeText(MainActivity.this, "Datele au fost
introduse cu succes!", Toast.LENGTH_LONG).show();

        } else {

            Toast.makeText(MainActivity.this, "Eroare de
inserare", Toast.LENGTH_LONG).show();
```

Pentru un feedback cu utilizatrul am creat si mesaje de tip log in care am vazut daca spre exepmu datele au fost inserate sau nu.

Pentru stergerea unui student am ales s ail identific dupa Nume si CNP ,asa cum e posibil sa avem personae cu Nume si prenume asemanatoare CNP e identificator unic.

Plus ca regula stergerii dintr-o baza de date impune o clauza WHERE in care se verifica o conditie pe toate inregeistrarile tebelei si se sterg doar acele inrgistrari care satisfac conditia daca nu se impune cconditia atunci se sterge tot din tabela,ceea ce e inadmiibil pentru o aplicatie Fiabila.

Sintaxa pentru stergerea studentului este urmatoarea:

```
public void DeleteStud(String Nume, Integer CNP){
     this.getWritableDatabase().delete(TABLE_NAME,
"(NUME='"+Nume+"') AND (CNP='"+CNP+"')", null );
     }
```

Astfel ma asigur ca studentul exmatriculat este sters din baza de date a facultatii.

Multe optiuni ne ofera si metoda Update, prin ea se actualizeaza majoritatea informatiilor vitale de exemlu:

O studenta s-a maritat => prin CNP o gasim si introducem noul Nume;

Studetii trebuie sa treaca in alt an de studiu =>actualizam An_Studiu cu verificarea daca adunat numarul necesarde credite pentru promovare.

La trecerea unui an incrementam si varsta, sau studentul schimba facultatea toate informatiile acestea trebuie sa se poata modifica.Dand o gama larga de utilizare a bazei de date si stocarea de date cat mai recente.

Pentru afisare am ales doua ListView –uri in linii si opt coloane:

```
protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.viewcontents_layout);
  myDB = new DatabaseHelper(this);
 userList = new ArrayList<>();
        Cursor data = myDB.getListContents();
        int numLinii = data.getCount();
        if(numLinii == 0){
```

```
            Toast.makeText(ViewListContents.this,"Baza de date
e nepopulata ",Toast.LENGTH_LONG).show();

            }else{

                int i=0;

                while(data.moveToNext()){

                        user = new
User(data.getString(2),data.getString(3),

    data.getInt(4),data.getInt(5), data.getString(6),

    data.getInt(7), data.getInt(8));

  userList.add(user);

        System.out.println(data.getString(2)+"
"+data.getString(3)+" "+data.getInt(4)+

  " "+data.getInt(5)+ " "+data.getString(6)+

  " "+data.getInt(7)+" "+data.getInt(8) );


System.out.println(userList.get(i).getNume());

                    i++;

                } AdaptorListaColoane adapter =  new
AdaptorListaColoane(this,R.layout.llist_adapter_view,
userList);

                listView =  findViewById(R.id.listView);

                listView.setAdapter(adapter);

            }

    }

}
```

Acesta are functia principal sa asigneze valorile din tabela pa fiecre colaoana din ListView ca dupa completare acestea sa formeze o tabela.

Aceasta a fos setata sa mai verifice baza de date la inceput pentru a vedea daca are ce afisa , daca nu rezulta ca in baza de date nu e nici o inregistrare, ca urmare utilizatorul est eavertizat cu un mesaj ca (“Baza de date e nepopulata”) si acesta poate lua masuri.

Totodata este utila si developerului pentru testarea aesteia, astfel stie unde sa ajuns cu executia mai précis si sa remedieze defectiunea in soft