

Aplicatie Identificare Piese Auto

Student: Eusebiu Rizescu

Grupa: 342C2

Materia: Baze de Date II

Cuprins

1. Descrierea temei
2. Descrierea bazei de date
 - 2.1 Diagrama bazei de date
 - 2.2 Structura tabelelor
 - 2.3 Descrierea constrângerilor de integritate
 - 2.4 Descrierea procedurilor si funcțiilor
3. Descrierea aplicației
 - 3.1 Diagrama de clase
 - 3.2 Diagrama de stări și fluxul de lucru (workflow) pentru aplicație
 - 3.3 Prezentarea modului în care se face conexiunea cu baza de date
4. Capturi de ecran pentru interfețe și rapoarte
5. Concluzii
6. Bibliografie

1. Descrierea temei

Proiectul consta intr-o aplicatie care ajuta la identificarea variantelor de piese compatibile cu un autoturism ale carui date (marca, model, combustibil, capacitate motor) sunt introduse de utilizator.

Pe baza acestor informatii introduse, cu ajutorul informatiilor deja stocate in baza de date se identifica codul motor al autoturismului, iar cu acesta se vor identifica codurile pieselor pentru aceasta masina.

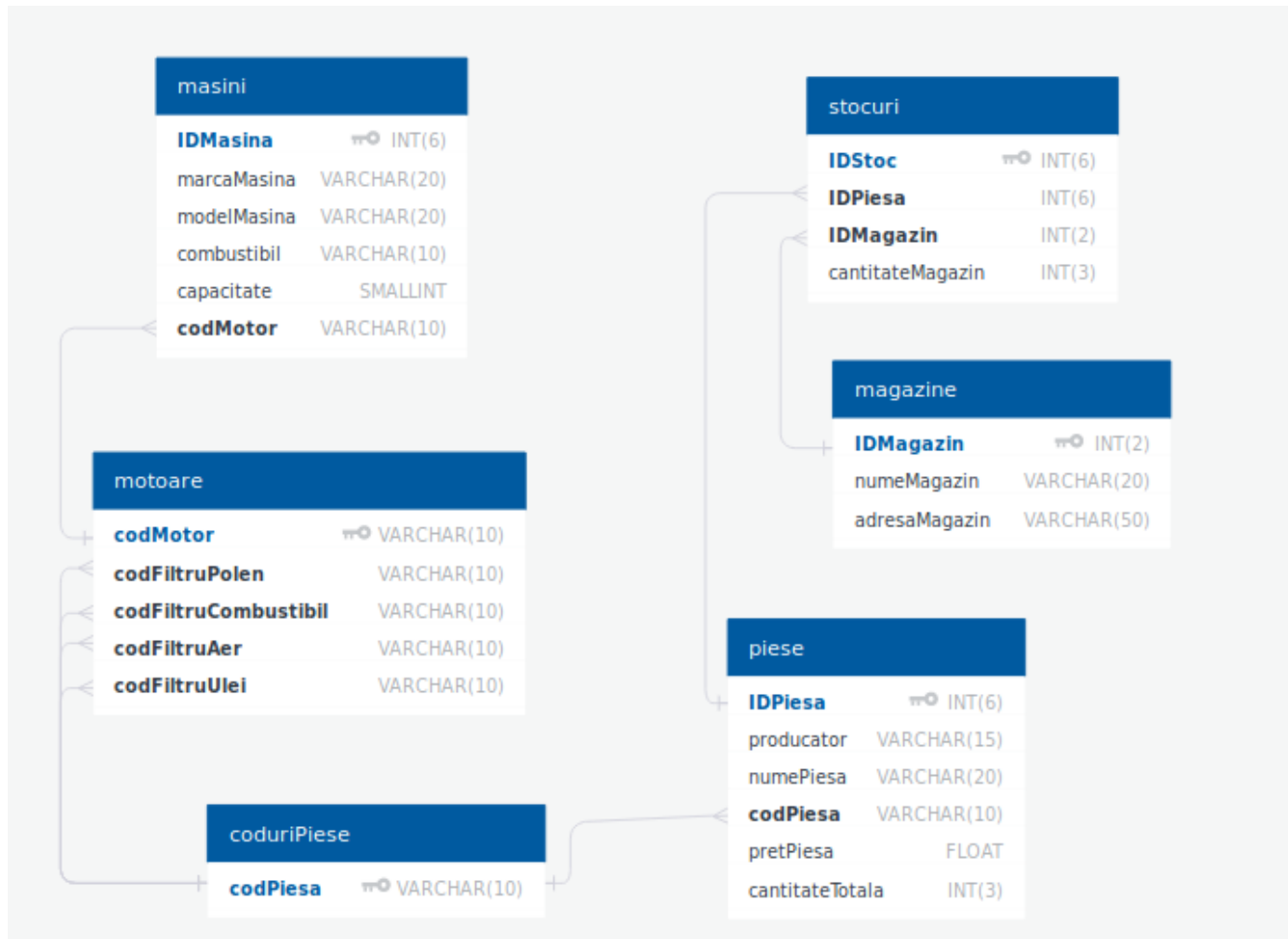
Acest proiect este un demo, asa ca am adaugat doar 4 piese: filtru de polen, filtru de combustibil, filtru de aer, filtru de ulei, dar pot fi extinse prin populare aditionala a bazei de date. Cu un cod de piesa, spre exemplu al filtrului de ulei, se cauta dupa variantele compatibile de la diferiti furnizori. Clientul alege piesa potrivita pentru nevoile lui si vede magazinele unde este disponibila aceasta piesa, unde o poate si rezerva. (se face update la cantitatea disponibila in magazinul respectiv)

Proiectul a fost realizat folosind:

- MySQL Version 8.0.12 Linux (MySQL Community Server – GPL)
- Python Version 2.7.15rc1
- Kivy Version 1.10.1 (pentru interfata grafica)

2. Descrierea bazei de date

2.1 Diagrama bazei de date



2.2 Structura tabelelor

```
CREATE TABLE masini(  
    IDMasina INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,  
    marcaMasina VARCHAR(20) NOT NULL,  
    modelMasina VARCHAR(20) NOT NULL,  
    combustibil VARCHAR(10) NOT NULL,  
    capacitate SMALLINT UNSIGNED NOT NULL,  
    codMotor VARCHAR(10) NOT NULL,  
    FOREIGN KEY (codMotor) REFERENCES motoare(codMotor)  
);  
  
CREATE TABLE motoare(  
    codMotor VARCHAR(10) PRIMARY KEY,  
    codFiltruPolen VARCHAR(10) NOT NULL,  
    codFiltruCombustibil VARCHAR(10) NOT NULL,  
    codFiltruAer VARCHAR(10) NOT NULL,  
    codFiltruUlei VARCHAR(10) NOT NULL,  
    FOREIGN KEY (codFiltruPolen) REFERENCES coduriPiese(codPiesa),  
    FOREIGN KEY (codFiltruCombustibil) REFERENCES coduriPiese(codPiesa),  
    FOREIGN KEY (codFiltruAer) REFERENCES coduriPiese(codPiesa),  
    FOREIGN KEY (codFiltruUlei) REFERENCES coduriPiese(codPiesa)  
);  
  
CREATE TABLE piese(  
    IDPiesa INT(6) UNSIGNED PRIMARY KEY,  
    producator VARCHAR(15) NOT NULL,  
    numePiesa VARCHAR(20) NOT NULL,  
    codPiesa VARCHAR(10) NOT NULL,  
    pretPiesa FLOAT NOT NULL,  
    cantitateTotala INT(3),  
    FOREIGN KEY (codPiesa) REFERENCES coduriPiese(codPiesa)  
);  
  
CREATE TABLE magazine(  
    IDMagazin INT(2) UNSIGNED PRIMARY KEY,  
    numeMagazin VARCHAR(20) NOT NULL,  
    adresaMagazin VARCHAR(50) NOT NULL  
);  
  
CREATE TABLE stocuri(  
    IDStoc INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,  
    IDPiesa INT(6) UNSIGNED NOT NULL,  
    IDMagazin INT(2) UNSIGNED NOT NULL,  
    cantitateMagazin INT(3) NOT NULL,  
    FOREIGN KEY (IDPiesa) REFERENCES piese(IDPiesa),  
    FOREIGN KEY (IDMagazin) REFERENCES magazine(IDMagazin)  
);  
  
CREATE TABLE coduriPiese(  
    codPiesa VARCHAR(10) PRIMARY KEY  
);
```

2.3 Descrierea constrângerilor de integritate

Tabela **coduriPiese** are o singura coloana numita **codPiesa** (evident Primary Key) cu linii care reprezinta **codPiesa**.

Tabela **magazine** contine informatii despre magazinele partenere si are Primary Key coloana **IDMagazin**

Tabela **piese** are foreign key coloana **codPiesa** care referentiaza coloana **coduriPiese.codPiesa**

Tabela **motoare** are foreign keys coloanele **codFiltruPolen**, **codFiltruCombustibil**, **codFiltruAer** si **codFiltruUlei** care referentiaza coloana **coduriPiese.codPiesa**

Tabela **masini** are foreign key coloana **codMotor** care referentiaza coloana **motoare.codMotor**

Tabela **stocuri** are doua foreign keys si anume:
IDPiesa care referentiaza coloana **piese.IDPiesa**
IDMagazin care referentiaza coloana **magazine.IDMagazin**

2.4 Descrierea procedurilor si funcțiilor

Triggere:

Am folosit 2 triggere care actualizeaza coloana **piese.cantitateTotala** atunci cand se insereaza / update-aza in tabela **stocuri**

```
CREATE TRIGGER after_stocuri_insert
  AFTER INSERT ON stocuri
  FOR EACH ROW
  BEGIN
    UPDATE piese
      SET cantitateTotala = (SELECT sum(cantitateMagazin) FROM stocuri
                             WHERE IDPiesa = NEW.IDPiesa)
    WHERE IDPiesa = NEW.IDPiesa;
  END;
```

```
CREATE TRIGGER after_stocuri_update
  AFTER UPDATE ON stocuri
  FOR EACH ROW
  BEGIN
    UPDATE piese
      SET cantitateTotala = (SELECT sum(cantitateMagazin) FROM stocuri
                             WHERE IDPiesa = NEW.IDPiesa)
    WHERE IDPiesa = NEW.IDPiesa;
  END;
```

Funcții:

Am folosit o singura functie care returneaza codMotor din tabela masini care are ca input 4 parametri: marcaMasina, modelMasina, combustibil, capacitate

```
CREATE FUNCTION returnCodMotor (p_marcaMasina VARCHAR(20),
                                p_modelMasina VARCHAR(20), p_combustibil VARCHAR(10),
                                p_capacitate SMALLINT UNSIGNED)
  RETURNS VARCHAR(10) DETERMINISTIC
  RETURN (SELECT codMotor
          FROM masini
          WHERE UPPER(p_marcaMasina) = UPPER(marcaMasina) AND
                UPPER(p_modelMasina) = UPPER(modelMasina) AND
                UPPER(p_combustibil) = UPPER(combustibil) AND
                p_capacitate = capacitate);
```

Proceduri:

Am folosit 7 proceduri pentru a ma ajuta in rezolvarea query-urilor necesare aplicatiei si raporturilor:

a) Procedura care gaseste codurile pieselor compatibile cu masina data ca parametri. Aceasta procedura foloseste functia creata anterior

```
CREATE PROCEDURE coduriPiese (IN marcaMasina VARCHAR(20), IN modelMasina
                              VARCHAR(20), IN combustibil VARCHAR(10), IN capacitate
                              SMALLINT UNSIGNED)
BEGIN
    SELECT * FROM motoare
    WHERE codMotor = returnCodMotor(marcaMasina, modelMasina, combustibil,
                                    capacitate);
END;
```

b) Procedura care arata toate variantele de piese (toti producatorii) pentru un codPiesa dat ca parametru

```
CREATE PROCEDURE varianteCodPiesa (IN codulPiese VARCHAR(10))
BEGIN
    SELECT * FROM piese WHERE codPiesa = codulPiese;
END;
```

c) Procedura care cauta toate locatiile in care este prezenta o piesa (dupa IDPiesa)

```
CREATE PROCEDURE locatiePiesa (IN IDPiese INT(6))
BEGIN
    SELECT s.IDStoc, s.IDPiesa, s.cantitateMagazin, m.numMagazin, m.adresaMagazin
    FROM stocuri s, magazine m
    WHERE IDPiesa = IDPiese AND s.IDMagazin = m.IDMagazin;
END;
```

d) Procedura care ajuta gasirea cantitatii disponibile in magazin a unei piese, pentru a putea sti daca putem sa rezervam

```
CREATE PROCEDURE cantitateDisponibila (IN IDulStoc INT(6))
BEGIN
    SELECT cantitateMagazin FROM stocuri WHERE IDStoc = IDulStoc;
END;
```


e) Procedura care rezerva piesa in magazin (face update in tabela stocuri)

```
CREATE PROCEDURE updateStoc (IN IDulStoc INT(6))
BEGIN
    UPDATE stocuri
        SET cantitateMagazin = cantitateMagazin - 1
        WHERE IDStoc = IDulStoc;
END;
```

f) Procedura care face raportul1 (Raport total piese grupat dupa codPiesa)

```
CREATE PROCEDURE raport1 ()
BEGIN
    SELECT codPiesa, numePiesa, sum(cantitateTotala)
    FROM piese
    GROUP BY codPiesa, numePiesa;
END
```

g) Procedura care face raportul2 (Raport total piese grupat dupa magazine)

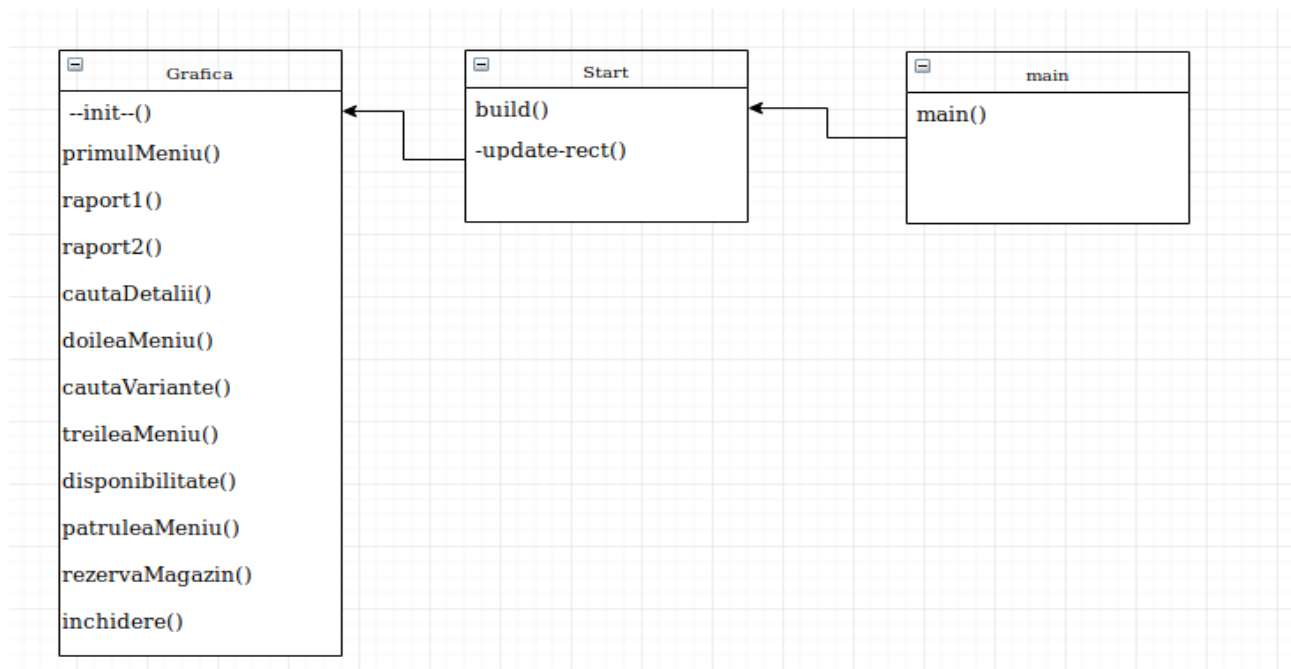
```
CREATE PROCEDURE raport2 ()
BEGIN
    SELECT a.IDMagazin, b.numMagazin, b.adresaMagazin, count(a.cantitateMagazin)
    FROM stocuri a, magazine b
    WHERE a.IDMagazin = b.IDMagazin
    GROUP BY IDMagazin;
END;
```

h) Procedura care face raportul3 (Ce piese s-au rezervat in sesiunea curenta)

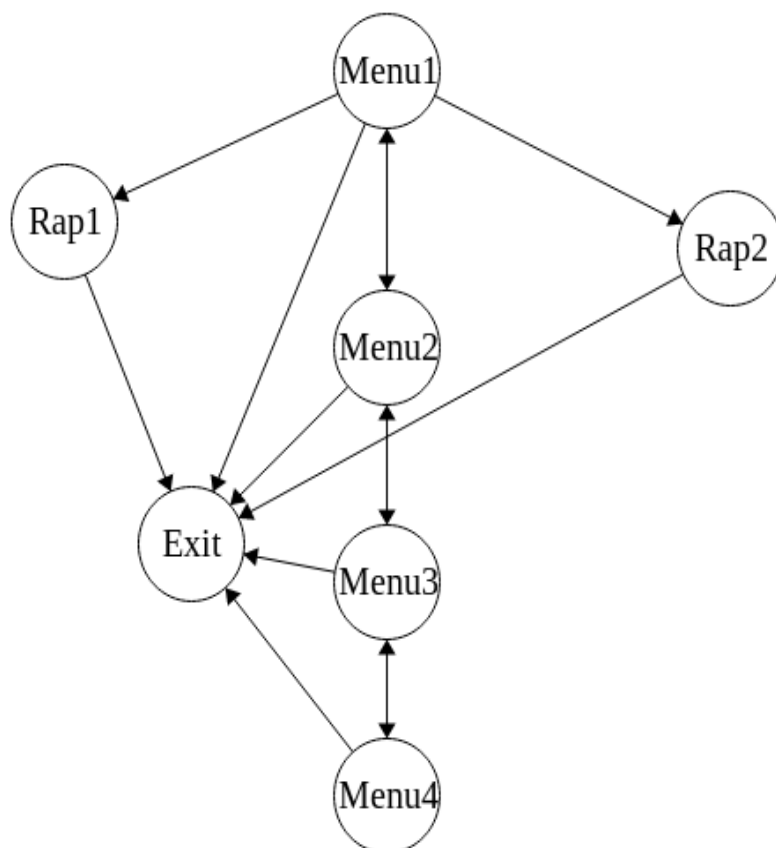
```
CREATE PROCEDURE ceACumparat (IN IDulStoc INT(6), IN IDMasinaCurenta VARCHAR(20))
BEGIN
    SELECT mas.marcaMasina, mas.modelMasina, mas.combustibil, mas.capacitate,
    mas.codMotor, pie.codPiesa, pie.numPiesa, pie.producator, mag.numMagazin
    FROM masini mas, piese pie, motoare mot, magazine mag, stocuri sto
    WHERE sto.IDStoc = IDulStoc AND sto.IDMagazin = mag.IDMagazin AND
    sto.IDPiesa = pie.IDPiesa AND (pie.codPiesa = mot.codFiltruPolen OR
    pie.codPiesa = mot.codFiltruCombustibil OR pie.codPiesa = mot.codFiltruAer
    OR pie.codPiesa = mot.codFiltruUlei) AND mot.codMotor = mas.codMotor
    AND mas.IDMasina = IDMasinaCurenta;
END;
```

3. Descrierea aplicației

3.1 Diagrama de clase



3.2 Diagrama de stări și fluxul de lucru (workflow) pentru aplicație



3.3 Prezentarea modului în care se face conexiunea cu baza de date

Pentru a interactiona cu baza de date am folosit modulul MySQLdb care este compatibil doar cu 2.7.

Importarea modului:

```
import MySQLdb
```

Conectarea la baza de date:

```
db = MySQLdb.connect("localhost","Bduser","Bduser")
```

Crearea unui obiect de tip cursor

```
cursor = db.cursor()
```

Executarea unei comenzi SQL

```
cursor.execute("USE ProiectBD2")
```

Captarea primei linii returnate

```
cursor.fetchone()
```

Captarea tuturor liniilor returnate

```
cursor.fetchall()
```

Commit

```
db.commit()
```

Terminarea conexiunii cu baza de date

```
db.close()
```

4. Capturi de ecran pentru interfețe și rapoarte

Meniul 1:

Program identificare piese auto
Pentru a cauta piesele disponibile va rugam introduceti urmatoarele detalii:

Marca autoturism:

Model autoturism:

Combustibil autoturism:

Capacitate cilindrica autoturism:

Meniul 2:

Masina identificata!

Marca masina: DACIA
Model masina: DUSTER
Combustibil: BENZINA
Capcitate: 1600 cc

Codul motor al masinii: K7M710
Codul filtrului de polen: AZK5563
Codul filtrului de combustibil: ARM4345
Codul filtrului de aer: MMM6655
Codul filtrului de ulei: TTR5338

Pentru a cauta variantele pentru filtru de polen introdu 1
Pentru a cauta variantele pentru filtru de combustibil introdu 2
Pentru a cauta variantele pentru filtru de aer introdu 3
Pentru a cauta variantele pentru filtru de ulei introdu 4

Meniul 3:

Cod piesa identificat!
Mai jos se pot vedea variantele de piese care au acest cod

IDPiesa	Producator	NumePiesa	CodPiesa	PretPiesa	CantitateTotala
661	Bosch	Filtru Aer	MMM6655	46.0 lei	20
662	TRW	Filtru Aer	MMM6655	33.0 lei	43
663	Dacia	Filtru Aer	MMM6655	120.0 lei	14

Pentru a vedea disponibilitatea pieselor in magazinele parteneri introdu IDPiesa

Meniul 4:

IDpiesa identificat!
Mai jos se poate vedea disponibilitatea in magazine pentru aceasta piesa

IDStoc	IDPiesa	CantMagazin	numeMagazin	adresaMagazin
63	662	13	AjutorPiese SRL	Str Razboiului, nr 66, Sector 4
64	662	10	MasinaCare SRL	Str Stefan cel Mircea, nr 77, Sector 6
65	662	9	PieseAutoMagazin SRL	Str Izvorul de Ioc, nr 1, Sector 3
66	662	11	AutoBaza SRL	Str Maior Patitu, nr 10, Sector 5

Introdu IDStoc pentru a rezerva piesa in magazin

65

Apasa pentru a rezerva in magazin

Apasa pentru alege alta varianta de piesa

Apasa pentru inchidere aplicatie

Rezervare
cu success:

Felicitari!! Ai rezervat cu success piesa in magazin.
Apasa click pentru a rezerva alta piesa sau a vedea rapoarte sau exit pentru a iesi

Apasa pentru continuare cumparaturi

Apasa pentru inchidere aplicatie

Raport 1:

Mai jos se pot vedea toate piesele grupate dupa codPiesa

codPiesa	numePiesa	CantitateTotala
AZK5563	Filtru Polen	21
RDS4363	Filtru Polen	16
TRS4643	Filtru Combustibil	59
ARM4345	Filtru Combustibil	41
RDT5522	Filtru Aer	77
MMM6655	Filtru Aer	76
TTR5338	Filtru Ulei	84
PPC4412	Filtru Ulei	54

Apasa pentru te intoarce la pagina initiala

Apasa pentru inchidere aplicatie

Raport 2:

Mai jos se pot vedea toate piesele grupate dupa magazine

IDMagazin	numeMagazin	adresaMagazin	totalPieseDinMagazin
10	AlexPieseShop SRL	Str Mecanicii, nr 2-4, Sector 1	16
20	AutoDoctor SRL	Str Marie-Curie, nr 45, Sector 2	16
30	PieseAutoMagazin SRL	Str Izvorul de Ioc, nr 1, Sector 3	20
40	AjutorPiese SRL	Str Razboiului, nr 66, Sector 4	17
50	AutoBaza SRL	Str Maior Patitu, nr 10, Sector 5	16
60	MasinaCare SRL	Str Stefan cel Mircea, nr 77, Sector 6	9

Apasa pentru te intoarce la pagina initiala

Apasa pentru inchidere aplicatie

Raport 3: Ce piese au fost rezervate in sesiunea curenta

Mai jos se pot vedea toate piesele rezervate din magazine

Marca	Model	Comb.	Capac	codMotor	codPiesa	numePiesa	producator	magazin
Dacia	Duster	motorina	1500	K9K-K7	PPC4412	Filtru Ulei	Bearmach	AutoBaza SRL
Dacia	Duster	motorina	1500	K9K-K7	TR54643	Filtru Combustibil	Bearmach	AlexPieseShop SRL
Dacia	Logan	benzina	1200	D4F-F7	RD15522	Filtru Aer	Dacia	AutoDoctor SRL
Dacia	MCV	benzina	1600	K7N710	AZK5563	Filtru Polen	Starline	PieseAutoMagazin SRL

Apasa pentru te intoarce la pagina initiala

Apasa pentru inchide aplicatie

Cateva exemple deerori pentru date introduse gresit / rezervare imposibila:

Nu ai introdus date intr-un camp, mai incearca!

Autoturism neidentificat in baza de date. Mai incearca!

Din pacate piesa nu mai este disponibila in magazin. Mai incearca!

5. Concluzii

Acest proiect a fost o adevarata provocare pentru mine. Nu am mai avut experienta pana acum cu MySQL, in sensul ca nu am incercat sa realizez conexiune cu alt limbaj in care sa scriu o aplicatie.

Am avut ceva probleme la crearea environment-ului (instalare MySQL, instalarea modulului MySQLdb din python si mai ales instalarea si configurarea Kivy-ului pentru interfata grafica)

Consider acest proiect introducerea perfecta pentru lucrearea aleasa de mine pentru licenta (aplicatie web cu accent pe back-end si baza de date)

6. Bibliografie

<https://www.tutorialspoint.com/mysql>

<https://dev.mysql.com/downloads/repo/apt/>

<https://dev.mysql.com/doc/mysql-apt-repo-quick-guide/en/>

<https://kivy.org/doc/stable/installation/installation-linux.html>

<https://www.w3schools.com/sql/>

<https://stackoverflow.com/>