

Proiect Baze de Date

Mireanu Cosmin-Andrei 331AB

Descrierea cerințelor:

Proiectul constă în crearea unei aplicații împreună cu o baza de date aferentă pentru urmărirea reparațiilor din cadrul unei firme de service de dispozitive.

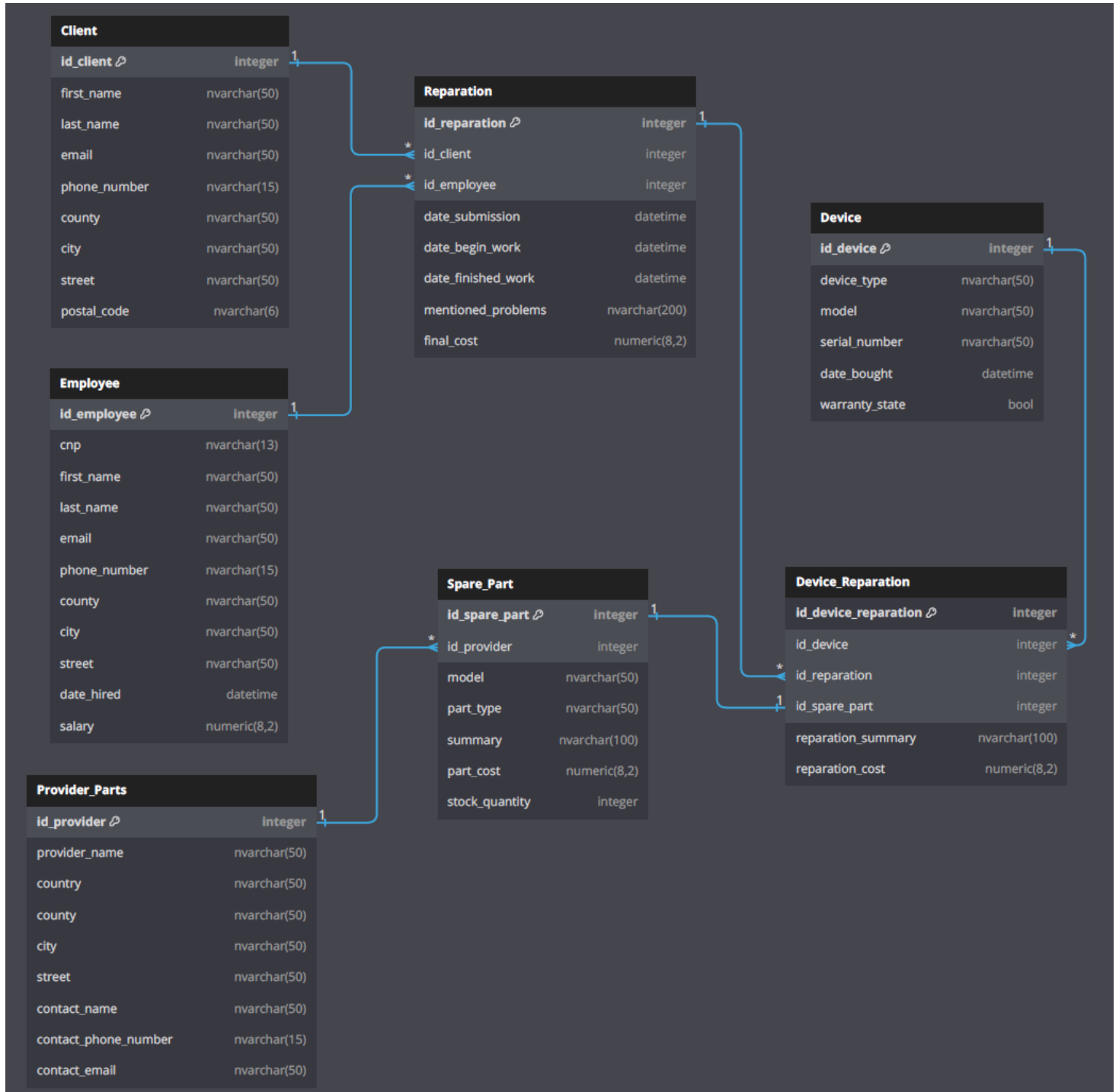
Etape de proiectare – Baza de date

Pentru baza de date am creat următoarele tabele:

- Tabela **Client** are attributele id_client (cheie primara) si first_name, last_name, email, phone_number, county, city, street si postal_code.
- Tabela **Employee**: id_employee (cheie primară), cnp, first_name, last_name, email, phone_number, county, city, street, date_hired si salary.
- Tabela **Device**: id_device (cheie primară), device_type, model, serial_number, date_bought si warranty_state.
- Tabela **Reparation**: id_reparation (cheie primară), id_client, id_employee, date_submission, date_begin_work, date_finished_work, mentioned_problems si final_cost.
- Tabela **Device_Reparation** (tabel de legătură): id_device_reparation (cheie primară), id_device, id_reparation, id_spare_part, reparation_summary si reparation_cost.
- Tabela **Provider_Parts**: id_provider (cheie primară), provider_name, country, county, city, street, contact_name, contact_phone_number si contact_email.

- Tabela **Spare_Part**: id_spare_part (cheie primară), id_provider, model, part_type, summary, part_cost si stock_quantity.

Structura tabele:



Relațiile dintre tabele:

- Client – Reparation: relație one to many
- Employee – Reparation: relație one to many

- Reparation – Device: relație many to many (prin intermediul tabelului de legatura Dispozitive_Reparatii)
- Spare_Part – Device_Reparation: relație one to one
- Provider_Parts – Spare_Part: relație one to many

Lista de funcționalități:

Când un client nou vine cu un dispozitiv, se iau câteva informații despre el cat si despre device-ul cu care a venit. Se deschide o fișă de reparație (tabel Reparation), unde este atribuit un angajat care va repara dispozitivul. În această fișă se trec ID-urile clientului/angajatului și se descrie problema, sau problemele dispozitivului (rămâne același id_reparation pentru mai multe înregistrări). Se mai atribuie data depunerii, începerii și finalizarea reparației, și costul final (reparation_cost + part_cost + adaos pentru profit).

O înregistrare în tabelul Device_Reparation este creată pentru fiecare problemă raportată în fișa de reparație, unde este trecut ID-ul reparației, dispozitivul care urmează să fie reparat și piesa de schimb care va fi folosită. Se mai notează descrierea reparației și costul efectiv.

Tabelele Employee, Provider_Parts și Spare_Parts vor fi completate/modificate de un administrator, odată cu angajarea, găsirea de noi furnizori și cumpărarea unor noi piese de schimb.

Constrângeri:

Un Angajat poate lucra pe mai multe reparatii.

Un Client poate avea mai multe reparatii, fiecare cu un dispozitiv separat.

Pentru tabelul Reparation, daca exista mai multe probleme, se pot face mai multe initializari pentru fiecare problema raportata.

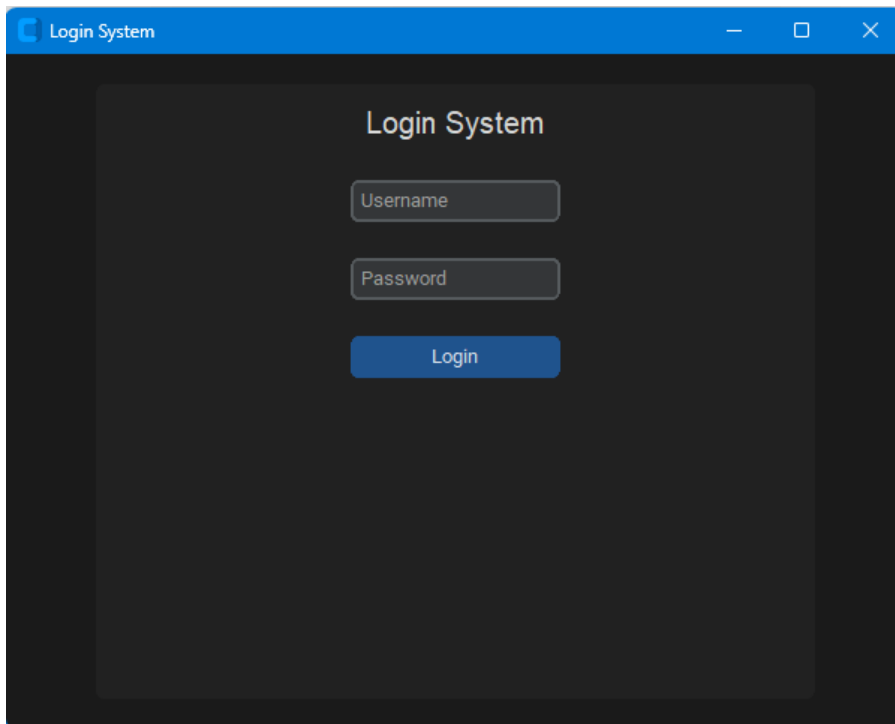
Unui dispozitiv i se pot face mai multe reparatii, fiecare pentru alta problema constata, unde id_reparation ramane acelasi, se schimba id_device_reparation, este unic pentru fiecare problema a device-ului. Fiecare initializare din Device_Reparation reprezinta reparatia unei componente, costul fiind declarat pentru fiecare reparatie.

Pentru fiecare problema, trebuie folosita o piesa de schimb diferita, in functie de reparatie. Aceste piese vin de la diferiti furnizori si au pret diferit in functie de tipul de piesa, model si furnizor.

Funcționarea aplicației:

Aplicația a fost realizata cu limbajul Python si 3 librarii externe, care au ajutat la conectarea bazei de date la cod, iar celelalte 2 au ajutat la crearea interfeței vizuale si crearea si modificarea tabelelor.

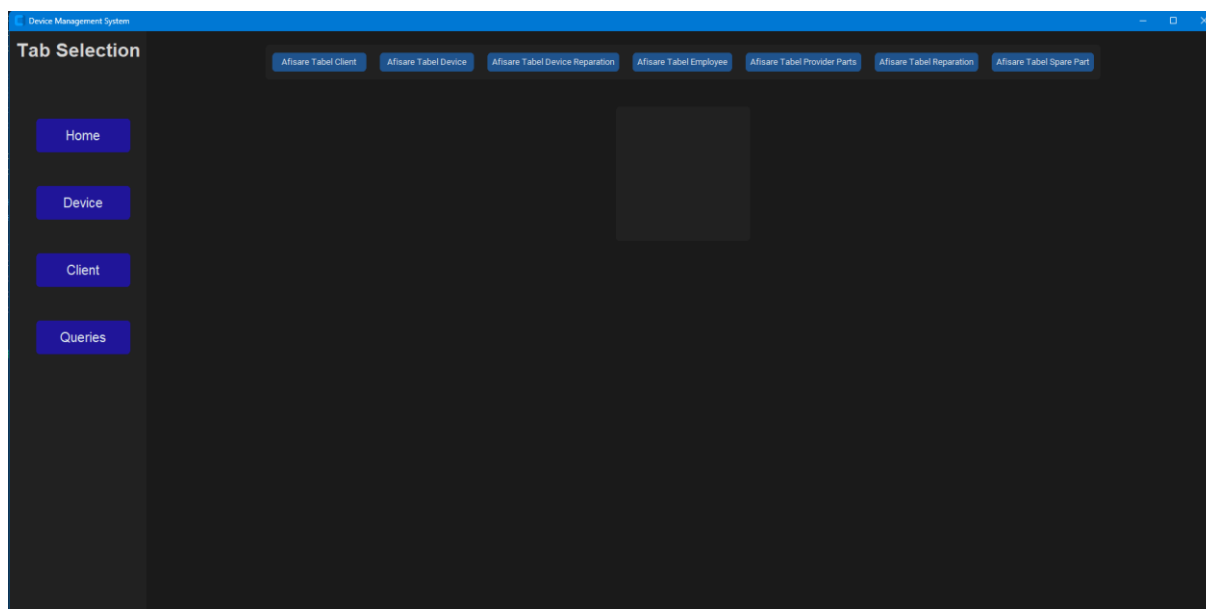
Pentru a lucra mai corect si bine, am împărțit codul in 5 fișiere .py, unul main, care conține toata partea de front-end, iar celelalte 4 conțin funcțiile necesare pentru a afișa corect tabelele si query-urile necesare.



Pentru accesarea aplicatiei, user-ul este admin, iar parola este 1234.

Daca acestea nu sunt introduse corect, utilizatorul mai are 2 șanse sa le introducă. Daca nu reușește aplicația se închide.

Daca datele sunt introduse corect, se închide fereastra de login si se deschide fereastra principala a aplicatiei.



Pagina Principala, unde vedem in stanga paginile, iar in centru afisarea pentru toate tabelele

Device Management System

Tab Selection

Home

Device

Client

Queries

Afisare Tabel Client

Afisare Tabel Device

Afisare Tabel Device Reparation

Afisare Tabel Employee

Afisare Tabel Provider Parts

Afisare Tabel Reparation

Afisare Tabel Spare Part

First Name	Last Name	Email	Phone Number	County	City	Street	Postal Code
Ion	Popescu	ion.popescu@email.com	0712345678	Bucharest	Bucharest	Strada Victoriei	050711
Elena	Ionescu	elena.ionescu@email.com	0723456789	Cluj	Cluj-Napoca	Strada Eroilor	400010
Mihai	Radu	mihai.radu@email.com	0734567890	Iasi	Iasi	Strada Independentei	700010
Ana	Dumitru	ana.dumitru@email.com	0745678901	Brasov	Brasov	Strada Muresenilor	500020
Cristina	Gheorghiu	cristina.gheorghiu@email.com	0756789012	Constanta	Constanta	Strada Tomis	900001
Alexandru	Stoica	alexandru.stoica@email.com	0767890123	Timis	Timisoara	Strada Revolutiei	300001
Raluca	Popa	raluca.popa@email.com	0778901234	Prahova	Ploiesti	Strada Mihai Viteazu	100001
Andrei	Dragan	andrei.dragan@email.com	0789012345	Dolj	Craiova	Strada Carol	200001
Laura	Georgescu	laura.georgescu@email.com	0790123456	Sibiu	Sibiu	Strada Cetatii	550001
Radu	Nistor	radu.nistor@email.com	0701234567	Mures	Targu Mures	Strada Libertatii	540001
Carmen	Cozma	carmen.cozma@email.com	0732345678	Arad	Arad	Strada Mihai Eminescu	310001

Afişarea pentru tabelul Clienţi

Fiecare buton are o functie in spate, care are sintaxa SQL care va afişa acel tabel. De fiecare data când este apăsat, tabelul vechi este şters, este creat cel nou si populat cu datele corecte.

Dupa orice modificare in baza de date, aceste schimbări vor apărea si in tabele.

Device Management System

Tab Selection

Home

Device

Client

Queries

Afisare Tabel Client

Afisare Tabel Device

Afisare Tabel Device Reparation

Afisare Tabel Employee

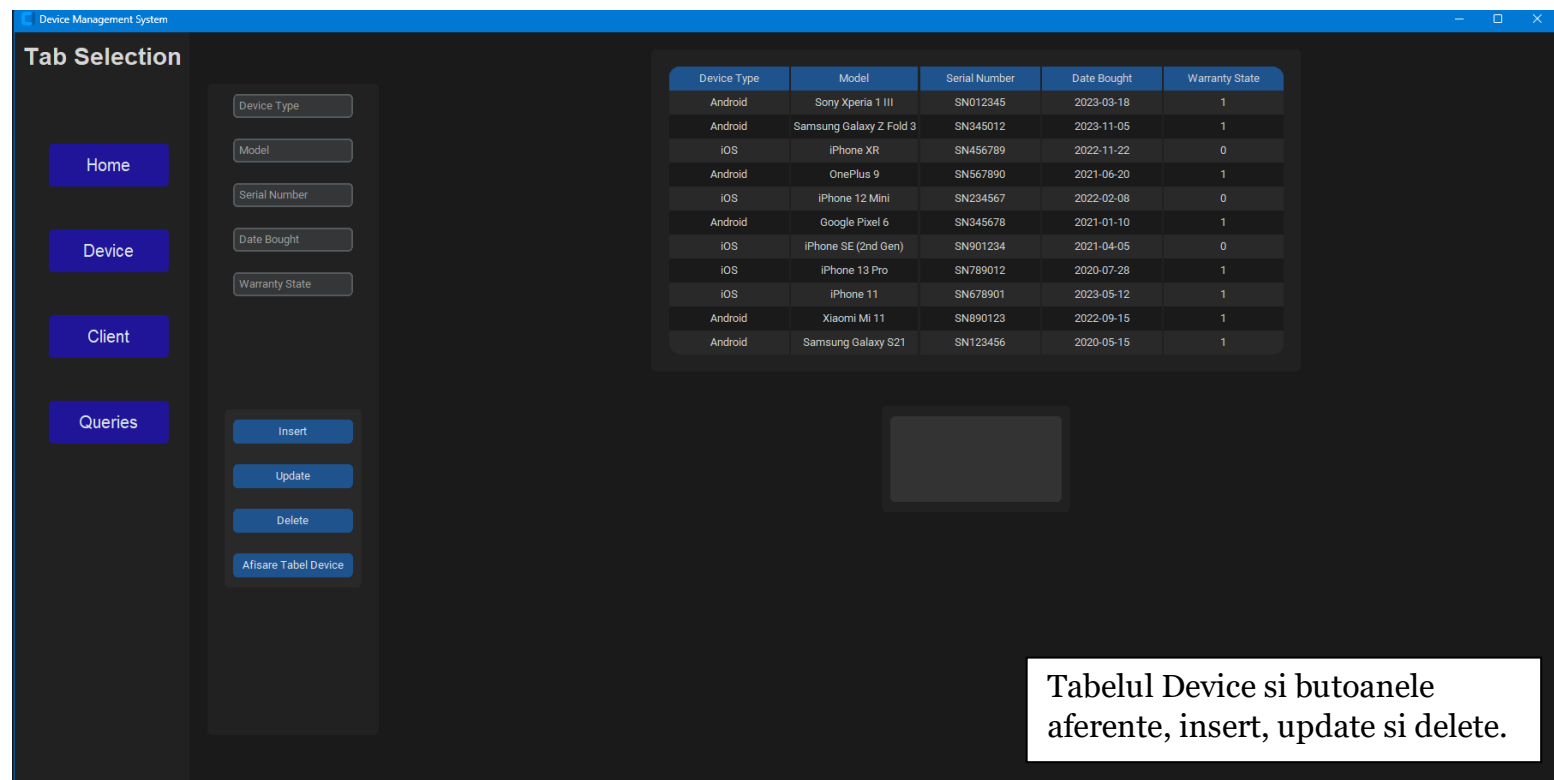
Afisare Tabel Provider Parts

Afisare Tabel Reparation

Afisare Tabel Spare Part

Device Type	Model	Serial Number	Date Bought	Warranty State
Android	Sony Xperia 1 III	SN012345	2023-03-18	1
Android	Samsung Galaxy Z Fold 3	SN345012	2023-11-05	1
iOS	iPhone XR	SN456789	2022-11-22	0
Android	OnePlus 9	SN567890	2021-06-20	1
iOS	iPhone 12 Mini	SN234567	2022-02-08	0
Android	Google Pixel 6	SN345678	2021-01-10	1
iOS	iPhone SE (2nd Gen)	SN901234	2021-04-05	0
iOS	iPhone 13 Pro	SN789012	2020-07-28	1
iOS	iPhone 11	SN678901	2023-05-12	1
Android	Xiaomi Mi 11	SN890123	2022-09-15	1
Android	Samsung Galaxy S21	SN123456	2020-05-15	1

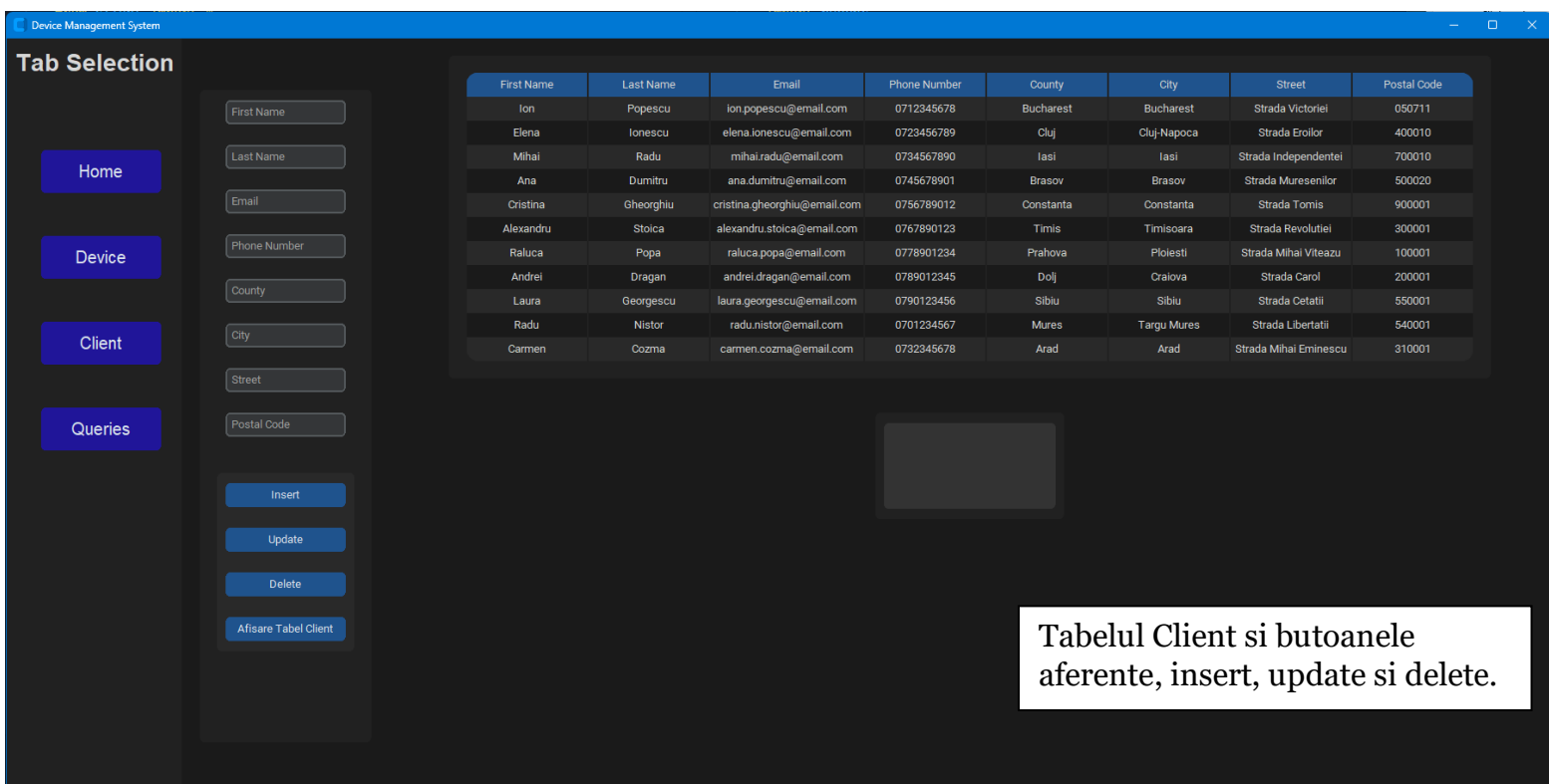
Afişarea pentru tabelul Device



Pentru a insera un device, toate câmpurile trebuie populate, daca nu, se va afișa un mesaj in text-box-ul de sub tabel.

Pentru a modifica un device, se va introduce SerialNumber, iar restul de câmpuri pot fi modificate, dupa ce se va introduce text in ele. Indiferent daca se face sau nu un update, va apărea un mesaj semnificativ care va alerta utilizatorul.

Pentru a șterge un device, se va introduce SerialNumber, iar obiectul va fi sters.



La fel ca si in cazul tabelului Device, toate câmpurile trebuie populate pentru a putea fi inserate.

La Update principiul de funcționare este același, aici modificarea se face după email.

La Delete, după ce se introduce email-ul, clientul va fi șters.

The screenshot displays the 'Device Management System' interface. On the left, a 'Tab Selection' sidebar contains buttons for 'Home', 'Device', 'Client', and 'Queries'. The 'Queries' tab is active, showing a list of 10 queries: 'Simple query 1' through 'Simple query 6', followed by 'Complex query 1' through 'Complex query 4'. Each query has an 'Insert minimum final cc' button below it. The main content area is titled 'Afișare Angajati si numarul total de reparatii pe care le-au facut' and displays a table with employee data.

First Name	Last Name	Date Hired	Salary	NrTotalReparatii
Mircea	Popescu	2021-03-15	2400.00	4
Diana	Ionescu	2022-05-20	3100.00	2
Adrian	Dumitru	2021-07-05	2800.00	2
Alexandru	Radu	2023-01-10	3600.00	2
Andreea	Gheorghiu	2022-11-18	3200.00	1

A text box in the bottom right corner states: 'Pagina Queries, unde se gasesc cele 10 query-uri, cu cerinta sus, 2 dintre ele avand un parametru variabil'.

Primele 5 query-uri simple și primele 3 complexe sunt făcute în cod, acestea afișează după o cerință dată, rezultatul interogării.

Cele cu parametru variabil, trebuie introdusă o valoare numerică, prețul minim respectiv stocul minim al unor piese de schimb.

Aceste query-uri se schimbă de fiecare dată după ce este pusă altă valoare în entry-box.

Interogările Simple:

1: Afisare Device-uri care au fost reparate cu piese mai scumpe decat 30, si stoc mai mare de 5

```
SELECT Device.model, Device.serial_number, Spare_Part.part_type, Spare_Part.part_cost,
Spare_Part.stock_quantity
FROM Device
INNER JOIN Device_Reparation ON Device.id_device = Device_Reparation.id_device
INNER JOIN Spare_Part ON Device_Reparation.id_spare_part = Spare_Part.id_spare_part
WHERE Spare_Part.part_cost >= 30 AND Spare_Part.stock_quantity > 5
ORDER BY Spare_Part.stock_quantity ASC
```

2: Afisare Device-uri cu device_type = IOS care au fost reparate de angajati cu salariu mai mare decat 3000

```
SELECT Device.model, Device.serial_number, Employee.first_name, Employee.last_name,
Employee.salary
FROM Device
INNER JOIN Device_Reparation ON Device.id_device = Device_Reparation.id_device
INNER JOIN Reparation ON Device_Reparation.id_reparation = Reparation.id_reparation
INNER JOIN Employee ON Reparation.id_employee = Employee.id_employee
WHERE Device.device_type = 'IOS' AND Employee.salary > 3000
```

3: Afisare Device-uri care au fost reparate de angajatii din Bucuresti, angajatii dupa 2022-01-01

```
SELECT Device.model, Device.serial_number, Employee.first_name, Employee.last_name,
Employee.city, Employee.date_hired
FROM Device
INNER JOIN Device_Reparation ON Device.id_device = Device_Reparation.id_device
INNER JOIN Reparation ON Device_Reparation.id_reparation = Reparation.id_reparation
INNER JOIN Employee ON Reparation.id_employee = Employee.id_employee
WHERE Employee.city = 'Bucarest' AND Employee.date_hired > '2022-01-01'
ORDER BY Employee.first_name ASC
```

4: Afisare Angajati si numarul total de reparatii pe care le-au facut

```
SELECT Employee.first_name, Employee.last_name, Employee.date_hired, Employee.salary,
COUNT(*) as NrTotalReparatii
FROM Employee
INNER JOIN Reparation ON Reparation.id_employee = Employee.id_employee
INNER JOIN Device_Reparation ON Device_Reparation.id_reparation =
Reparation.id_reparation
GROUP BY Employee.first_name, Employee.last_name, Employee.date_hired, Employee.salary
ORDER BY NrTotalReparatii DESC
```

5: Afisare Device-uri care au piesele de la ElectroParts

```
SELECT Device.model, Device.serial_number, Spare_Part.part_type,
Provider_Parts.provider_name, Provider_Parts.contact_name,
Provider_Parts.contact_phone_number, Provider_Parts.contact_email
FROM Device
INNER JOIN Device_Reparation ON Device.id_device = Device_Reparation.id_device
INNER JOIN Spare_Part ON Device_Reparation.id_spare_part = Spare_Part.id_spare_part
INNER JOIN Provider_Parts ON Spare_Part.id_provider = Provider_Parts.id_provider
WHERE Provider_Parts.provider_name = 'ElectroParts'
ORDER BY Device.serial_number ASC
```


6: Afisare Device-uri si piesele lor, daca pretul final de reparatie este mai mare sau egal decat x

```
SELECT Device.model, Device.serial_number, Spare_Part.part_type, Spare_Part.summary,
Spare_Part.part_cost, Device_Reparation.reparation_cost, Reparation.final_cost
FROM Device
INNER JOIN Device_Reparation ON Device.id_device = Device_Reparation.id_device
INNER JOIN Reparation ON Device_Reparation.id_reparation = Reparation.id_reparation
INNER JOIN Spare_Part ON Device_Reparation.id_spare_part = Spare_Part.id_spare_part
WHERE Reparation.final_cost >= {x}
ORDER BY Reparation.final_cost ASC
```

Interogările Complexe:

1: Afisare Device-uri care au fost reparate de angajatul cu cel mai mare salariu

```
SELECT Device.model, Device.serial_number, Employee.first_name, Employee.last_name,
Employee.salary
FROM Device
INNER JOIN Device_Reparation ON Device.id_device = Device_Reparation.id_device
INNER JOIN Reparation ON Device_Reparation.id_reparation = Reparation.id_reparation
INNER JOIN Employee ON Reparation.id_employee = Employee.id_employee
WHERE Employee.salary = (SELECT MAX(salary) FROM Employee)
```

2: Afisare Device-uri care au fost reparate de angajatul cu salariu mai mare decat media, si este din Bucuresti

```
SELECT Device.model, Device.serial_number, Employee.first_name, Employee.last_name,
Employee.salary
FROM Device
INNER JOIN Device_Reparation ON Device.id_device = Device_Reparation.id_device
INNER JOIN Reparation ON Device_Reparation.id_reparation = Reparation.id_reparation
INNER JOIN Employee ON Reparation.id_employee = Employee.id_employee
WHERE Employee.salary > (SELECT AVG(salary) FROM Employee) AND Employee.county =
'Bucharest'
ORDER BY Employee.salary DESC
```

3: Afisare Angajati si piese/model, care au folosit piese in reparatii de la providerul ElectroParts, are salariu mai mare decat media si este din Bucuresti

```
SELECT Employee.first_name, Employee.last_name, Employee.salary, Spare_Part.model,
Spare_Part.part_type
FROM Employee
INNER JOIN Reparation ON Employee.id_employee = Reparation.id_employee
INNER JOIN Device_Reparation ON Reparation.id_reparation =
Device_Reparation.id_reparation
INNER JOIN Spare_Part ON Device_Reparation.id_spare_part = Spare_Part.id_spare_part
INNER JOIN Provider_Parts ON Spare_Part.id_provider = Provider_Parts.id_provider
WHERE Provider_Parts.provider_name = 'ElectroParts' AND Employee.salary < (SELECT
AVG(salary) FROM Employee) AND Employee.county = 'Bucharest'
```

4: Afisare piese care sunt de la TechSupplies, au stoc mai mare sau egal decat x, si au fost folosite in reparatii de angajati cu salariu mai mare decat media

```
SELECT Spare_Part.part_type, Spare_Part.summary, Spare_Part.part_cost,  
Spare_Part.stock_quantity, Employee.first_name, Employee.last_name, Employee.salary  
FROM Spare_Part  
INNER JOIN Provider_Parts ON Provider_Parts.id_provider = Spare_Part.id_provider  
INNER JOIN Device_Reparation ON Spare_Part.id_spare_part =  
Device_Reparation.id_spare_part  
INNER JOIN Reparation ON Device_Reparation.id_reparation = Reparation.id_reparation  
INNER JOIN Employee ON Reparation.id_employee = Employee.id_employee  
WHERE Spare_Part.stock_quantity >= {x} AND Provider_Parts.provider_name =  
'TechSupplies' AND Employee.salary > (SELECT AVG(salary) FROM Employee)
```

Insert, Update, Delete:

Device:

```
statement = ("INSERT INTO dbo.Device (device_type, model, serial_number,  
date_bought, warranty_state) \  
VALUES ('{}', '{}', '{}', '{}', '{}')"  
)
```

```
statement = ("UPDATE dbo.Device \  
SET device_type = '{}\' \  
WHERE serial_number = '{}'"
```

```
statement = ("UPDATE dbo.Device \  
SET model = '{}\' \  
WHERE serial_number = '{}'"
```

```
statement = ("DELETE FROM dbo.Device \  
WHERE serial_number = '{}'"
```

Client:

```
statement = ("INSERT INTO dbo.Client (first_name, last_name, email,  
phone_number, county, city, street, postal_code) \  
VALUES ('{}', '{}', '{}', '{}', '{}', '{}', '{}', '{}')"
```

```
statement = ("UPDATE dbo.Client \  
    SET first_name = '{}' \  
    WHERE email = '{}'" \  
    )
```

```
statement = ("UPDATE dbo.Client \  
    SET last_name = '{}' \  
    WHERE email = '{}'" \  
    )
```

```
statement = ("UPDATE dbo.Client \  
    SET phone_number = '{}' \  
    WHERE email = '{}'" \  
    )
```

```
statement = ("DELETE FROM dbo.Client \  
    WHERE email = '{}'" \  
    )
```