Proiect PS

Student: Cosmin Cosan

Grupa: 30231

Enuntul problemei: Problema 16

Dezvoltați (analiză, proiectare, implementare) o aplicație desktop care poate fi utilizată de către o agenție imobiliară pentru gestionarea închirierilor de locuințe. Aplicația va avea 3 tipuri de utilizatori: client, angajat și administrator.

Utilizatorii de tip client pot efectua următoarele operații fără autentificare:

- ❖ Vizualizarea listei proprietăților disponibile pentru închiriat;
- ❖ Filtrarea listei proprietăților disponibile pentru închiriat după locație, preț, tip locuință, număr camere;
- ❖ Vizualizarea unor statistici legate de locuințele de închiriat disponibile: procente după locație, preț, tip locuință, număr camere utilizând grafice (structură radială, structură inelară, de tip coloană, etc.).

Utilizatorii de tip angajat pot efectua următoarele operații după autentificare:

- * Toate operatiile permise utilizatorilor de tip client;
- ❖ Operații CRUD în ceea ce privește persistența locuințelor de închiriat și a clienților;
- ❖ Închirierea unei locuințe de către un client;
- ❖ Salvare rapoarte / liste cu informații despre locuințele de închiriat în mai multe formate: csv, json. Utilizatorii de tip administrator pot efectua următoarele operații după autentificare:
- ❖ Operații CRUD pentru informațiile legate de utilizatorii de tip angajat

Date de autentificare:

Pentru *admin*: username: admin password: admin

Pentru employee:

Username: user Password: pass

Justificarea limbajului de programare ales

Am ales sa dezvolt aceasta aplicatie in Java deoarece este unul dintre limbajele de programare foarte dezvoltate in ceea ce priveste portabilitatea si totodata pentru ca aceasta aplicatie ar putea fi portata pe un browser web relativ simplu.

Justificarea alegerii șablonului de proiectare creațional si comportamental utilizat

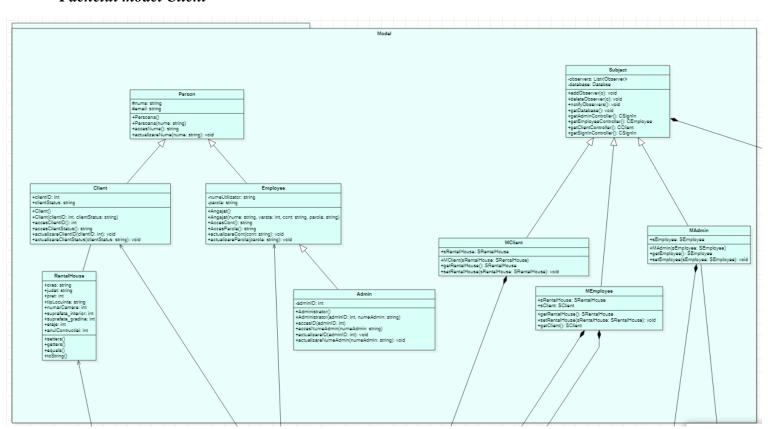
Ca si sablon de proiectare creational aam folosit metoda Factory pentru a alege mult mai usor metoda de salvare a rapoartelor, folosind astfel clasa parinte pentru declararea metodei de sciere, iar mai apoi folosind metode de suprascriere in ambele clase care mostenesc clasa parinte am realizat functionalitati diferite.

Am ales sa folosesc sablonul Factory pentru pentru salvarea rapoartelor, angajatul poate sa genereze rapoarte, asa ca am creeat clasa Report, de la care extind tipurile de rapoarte (CSVReport, JsonReport). Daca se doreste un raport de tip CSV se va creea un obiect de tip CSVReport, altfel, daca se doreste un raport de tip JSON se va creea un obiect de tip JsonReport.

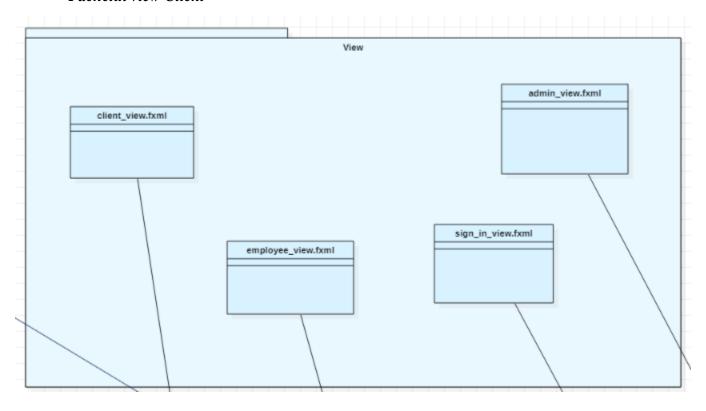
Ca sablon de proiectare comportamental am folosit Observer prin care clientul, angajatul si administratorul pot fi notificati atunci cand se manipuleaza baza de date de catre un alt utilizator al apicatiei.

Diagrama de clase - Client

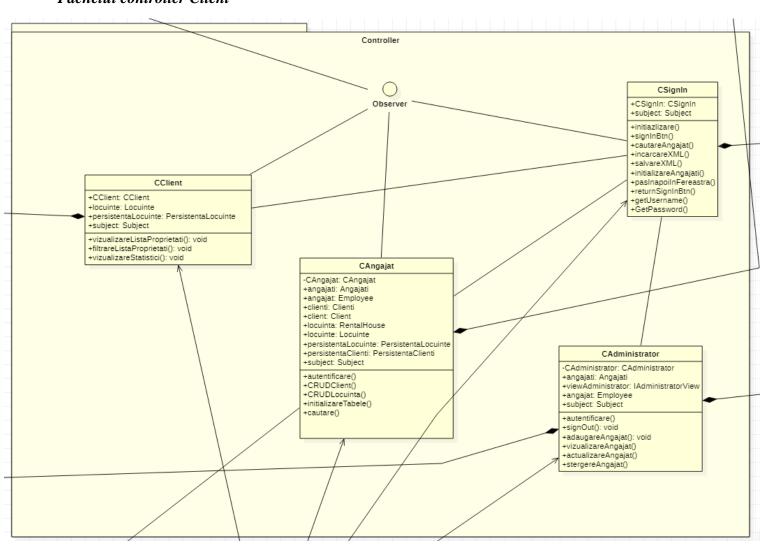
Pachetul model Client



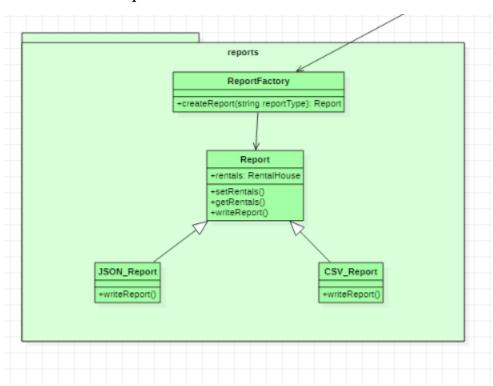
Pachetul view Client



Pachetul controller Client



Pachetul reports Client



Pachetul contact Client

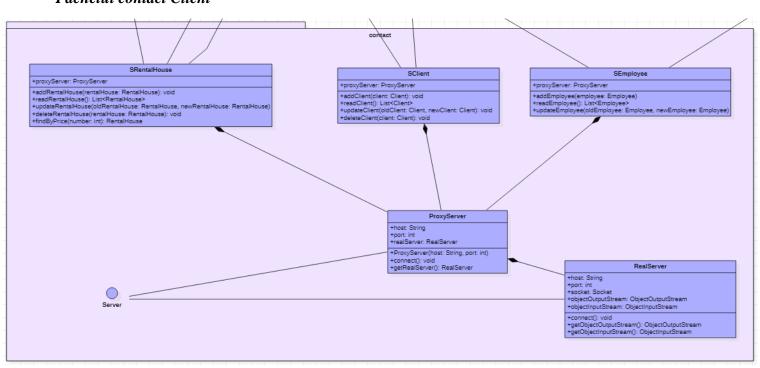


Diagrama cazurilor de utilizare Client

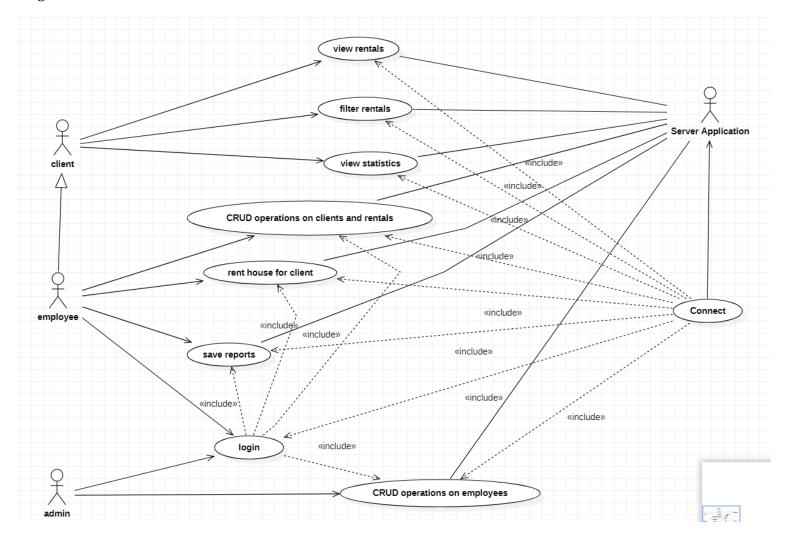
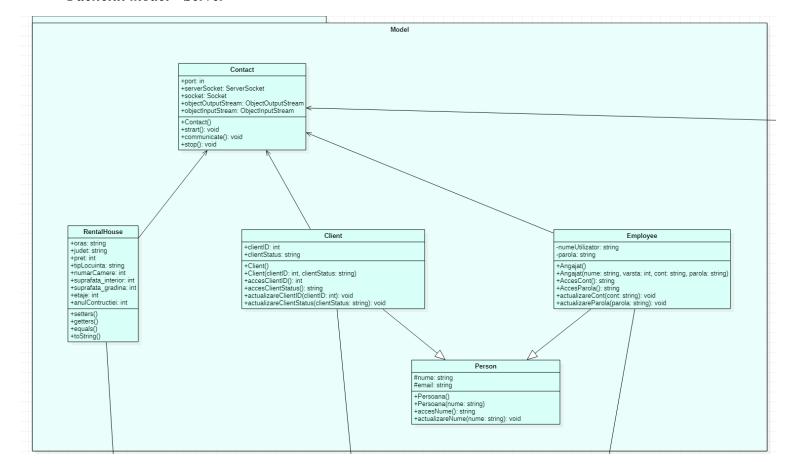


Diagrama de clase - Server

Pachetul model - Server



Pachetul persistence - Server

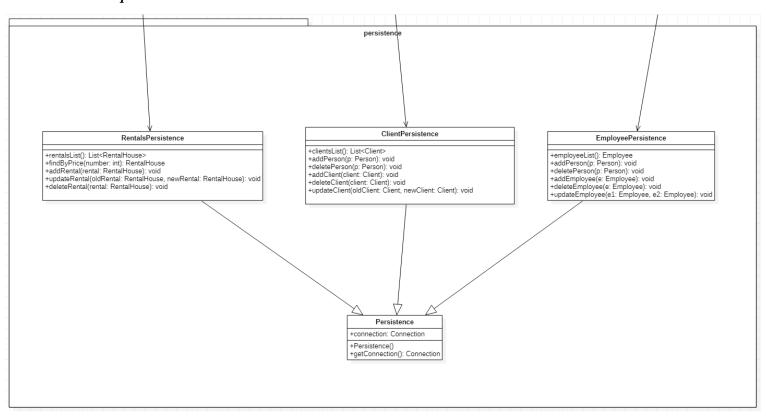


Diagrama cazurilor de utilizare - Server

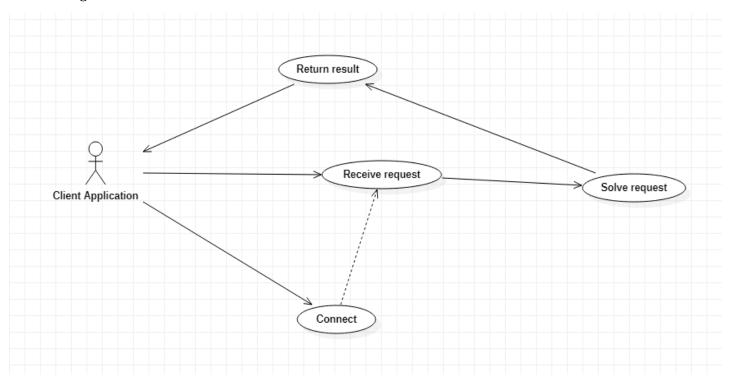
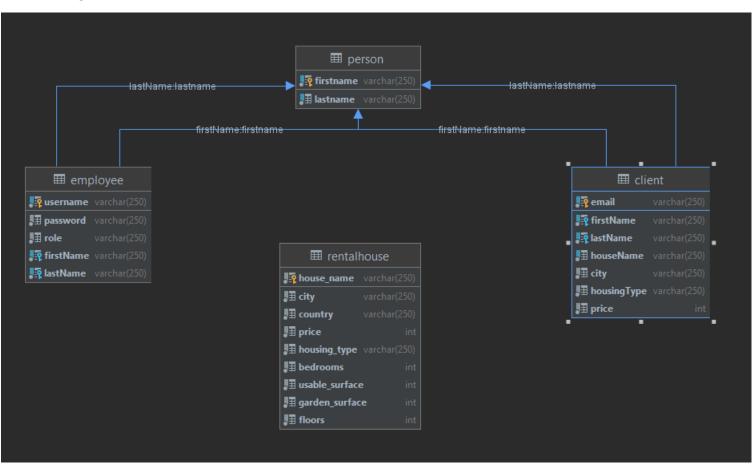


Diagrama bazei de date - Server



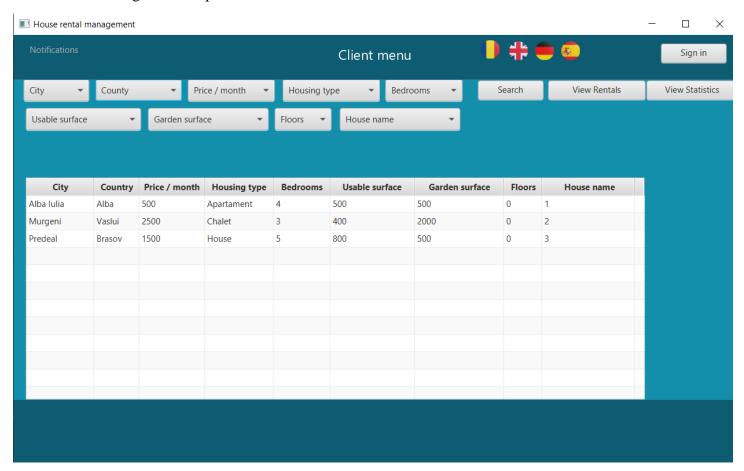
Descrierea aplicatiei:

Interfata grafica pentru client

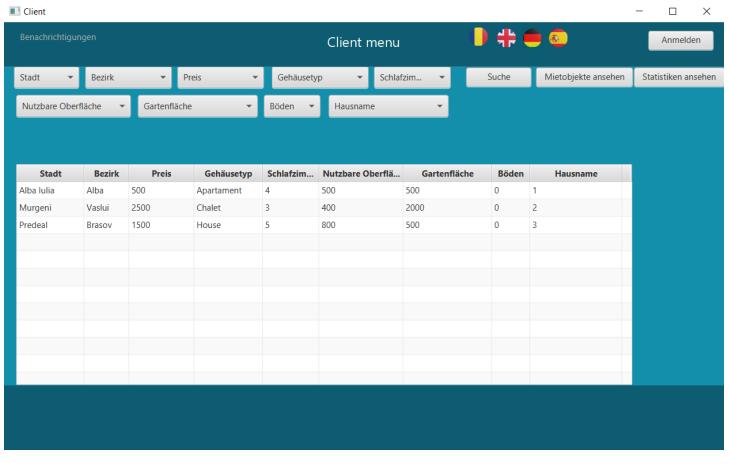
La lansarea in executie a aplicatiei se va deschide fereastra de mai sus unde putem sa gasim functionalitaiile implementate cu scopul de a oferi utilizatorilor de tip Client sa poata accesa aplicatia de gestionare a inchirierilor imobiliare.

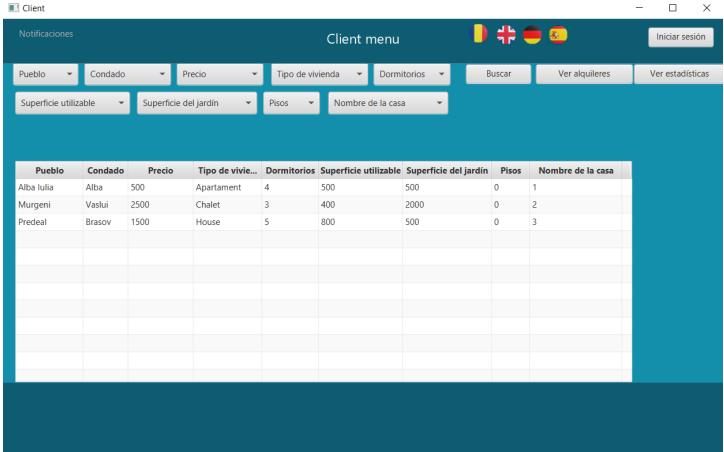
Pentru inceput putem observa in tabelul afisat in fereastra toate locuintele disponibile care au fost introduse in sistemul aplicatiei.

In partea de sus a ferestrei, avem 9 combobox-uri cu ajutorul carora putem sa sortam locuintele disponibile in tabel in functie de oras, judet, pret, tipul locuintei, numarul de dormitoare, suprafata locuintei, suprafata gradinii, numarul de etaje si numele locuintei sau a complexului de inchiriere. Daca selectam cel putin una dintre optiunile prezentate mai sus putem apasa pe butonul search pentru efectuarea cautarii, iar daca dorim sa resetam filtrarea din tabel putem sa apasam dinou pe butonul de search. Totodata putem naviga in fereastra, vizualizand fie ofertele disponibile de inchiriat, fie putem vizualiza niste statistici calcultate in functie de pretul locuintelor disponibile in tabel. In cazul in care am primit drepturi de angajat sau administrator putem sa apasam pe butonul Sign in si sa introducem datele necesare logarii cu drepturi de utilizator.

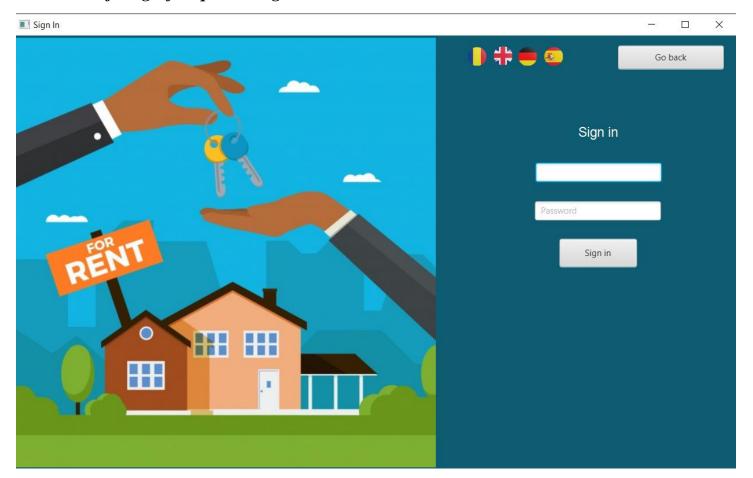


In partea de sus putem observa 4 butoane pentru alegerea limbii dorite, in functie de preferinta se poate apasa un buton care reprezinta limba dorita prin imaginea din cadranele cercului, in urma apasarii tot continutul interfetei se translateaza in limba aferenta fiecarui buton. Acesta se poate observa in imaginile de mai jos.





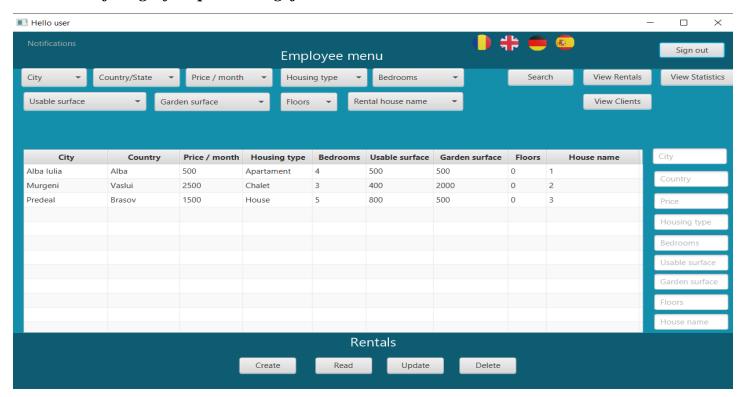
Interfata grafica pentru logare



Dupa cum am mentionat mai devreme, in urma apasarii butonului de Sign in din fereastra asociata clientilor, putem accesa fereastra prezentata mai sus. In functie de drepturile acordate unui utilizator acesta va putea sa navigheze mai departe intr-una din ferestrele acordate fie unui angajat fie unui administrator. In text field-urile din aceasta fereastra se pot introduce datele acordate utilizatorului pentru a naviga mai departe, daca un utilizator doreste sa se intoarca la fereastra acordata clientului poate sa faca aceasta prin apasarea butonului Click to go back, care va inchide aceasta fereastra de logare si va deschide automat fereastra acordata clientului.

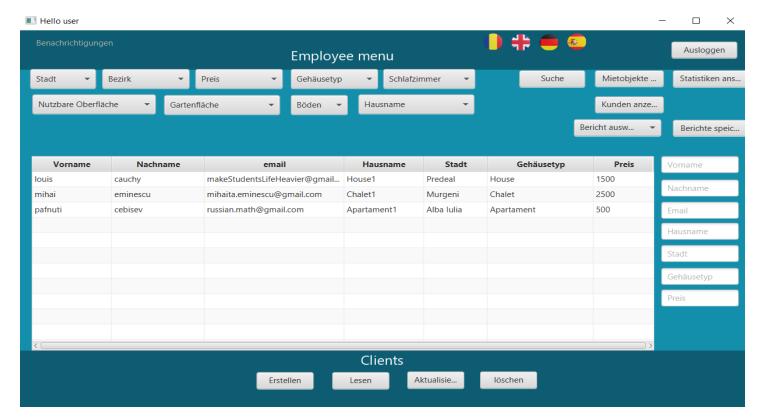
In partea de sus putem observa dinou cele 4 butoane pentru alegerea limbii dorite, in functie de preferinta se poate apasa un buton care reprezinta limba dorita prin imaginea din cadranele cercului, in urma apasarii tot continutul interfetei se translateaza in limba aferenta fiecarui buton.

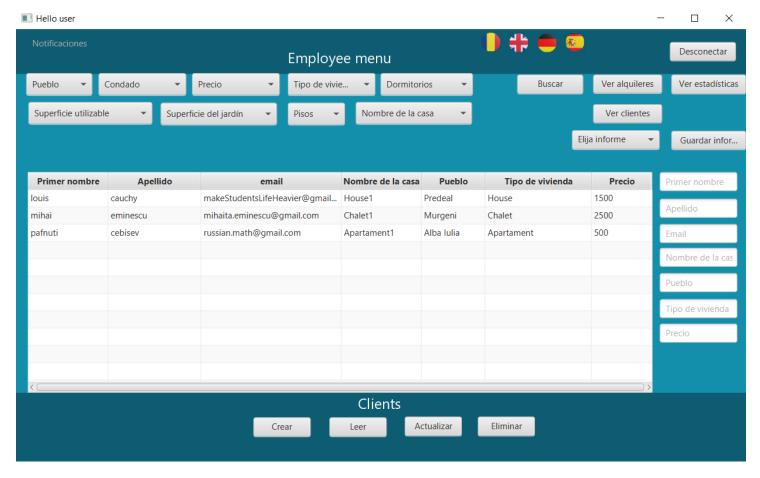
Interfata grafica pentru angajat



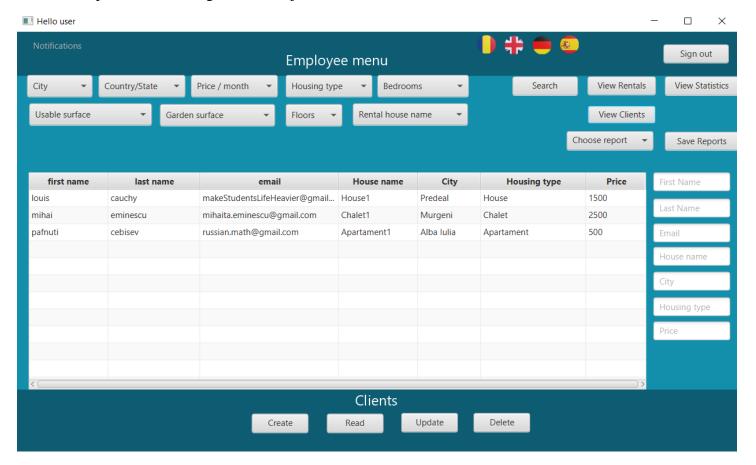
Daca unui utilizator i s-au acordat drepturi de angajat, dupa logare se va deschide fereastra de mai sus unde putem sa observam acelasi meniu ca si in fereastra clientilor insa, au mai aparut cateva butoane. In partea de sus avem butonul Sign out cu ajutorul caruia putem sa ne delogam, in partea drepta putem observa cateva text field-uri unde putem sa introducem datele necesare crearii si actualizarii ofertelor de inchiriere din tabelul prezentat, iar butonul de view clinets ofera posibilitatea de a naviga in fereastra, afisand clientii introdusi in sistem, putem observa aceasta in imaginea de mai jos.

In partea de sus putem observa dinou cele 4 butoane pentru alegerea limbii dorite, in functie de preferinta se poate apasa un buton care reprezinta limba dorita prin imaginea din cadranele cercului, in urma apasarii tot continutul interfetei se translateaza in limba aferenta fiecarui buton. Aceasta se poate observa in imaginile de mai jos.

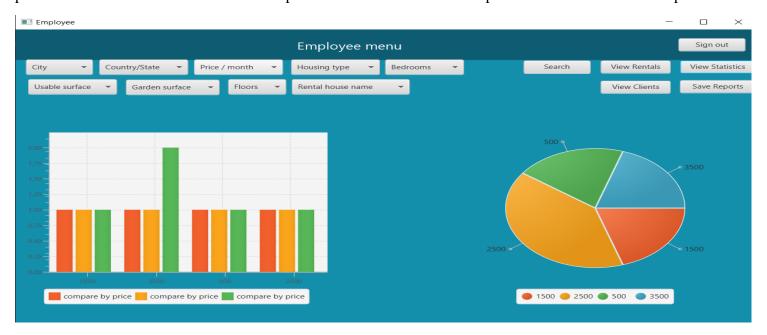




Similar cu tabelul de locuinte, insa datele prezentate sunt introduse pe alte obiecte, la tranzitia din modul de vizualizare al locuintelor in modul de vizualizare al clientilor putem observa ca meniul din partea dreapta respectiv din partea de jos, s-au schimbat oferindu-ne astfel posibilitatea de prelucrare a datelor pe tabelul de clienti. La fel ca si clientii, orice angajat poate sa acceseze statisticile salvate in functie de pret pe baza locuintelor introduse in sistem, aceasta este prezentata in imaginea de mai jos.



Daca un utilizator s-a logat in sistem iar acesta are drepturi de administrator, la logarea in sistem se va deschide fereastra de mai jos. Aici putem observa un alt tabel, insa acesta contine datele salvate in sistem despre angajatii din cadrul acestei companii de inchirieri imobiliare. In partea de sus avem dinou un buton de Sign out care la apasarea acestuia ne va trimite catre fereastra de client si va efectua delogarea din sistem. In partea de sus pe centrul ferestrei putem observa dinou cateva butoane si respectiv text field-uri din care se preiau datele in vederea manipularii lor.



Daca utilizatorul de tip administrator vrea sa creeze un utilizator nou, acesta va trebui sa introduca datele necesare in text filed-urile asociate fiecarui camp din tabel si pe urma sa foloseasca butonul de create.

Pentru citirea datelor din tabel se poate selecta o tupla iar apoi prin apasarea butonului de read, datele din tupla respectiva vor fi introduse automat in text field-urile asociate fiecarei coloane din tabel.

Similar pentru actualizara datelor, va trebui sa selectam o tupla din tabel si sa introducem datele pe care vrem sa le actualizam in campurile de text si sa apasam butonul de update, si in ultimul rand daca vrem sa stergem o tupla din tabel, si mai exact sa stergem un angajat din sistemul pe care il administram putem sa facem aceasta selectand angajatul sau tupla dorita din tabel si in urma sa apasam butonul delete.

Pentru verificarea functionalitaiilor tuturor celor prezentate mai sus, putem sa ne delogam si respectiv sa ne relogam pentru a putea observa ca datele introduse, acutalizate sau sterse din sistem au fost efectuate cu succes.

	Sign out				
	Create	Read Update	Delete		
First Name	Last Name	Username	Password Rol	le	
first name	last name	username	password	role	_
employee2	employee22	admin	admin	admin	
employee3	employee33	cosmin	cosmin	admin	
employee1	employee11	user	pass	employee	

In partea de sus putem observa dinou cele 4 butoane pentru alegerea limbii dorite, in functie de preferinta se poate apasa un buton care reprezinta limba dorita prin imaginea din cadranele cercului, in urma apasarii tot continutul interfetei se translateaza in limba aferenta fiecarui buton. Aceasta se poate observa in imaginile de mai jos.

