A picture containing logo

Description automatically generated

**FACULTATEA DE AUTOMATICĂ ŞI CALCULATOARE**

**CATEDRA CALCULATOARE**

Aplicatie lant de galerii de arta

Documentatie

NUME STUDENT: Turcu Cezar Simeon

GRUPA: 30235

# CUPRINS

[1. Enuntul problemei 3](#_Toc99311586)

[2. Instrumente utilizate 3](#_Toc99311587)

[3. Justificare limbaj de programare ales 4](#_Toc99311588)

[4. Descrierea diagramelor UML 4](#_Toc99311589)

[5. Descrierea aplicatiei 7](#_Toc99311590)

1. Enuntul problemei

Dezvoltați o aplicație care poate fi utilizată într-un lanț de galerii de artă. Aplicația va avea 3 tipuri de utilizatori: vizitator al galeriei de artă, angajat al lanțului de galerii de artă și administrator.   
Utilizatorii de tip vizitator pot efectua următoarele operații fără autentificare:   
❖ Vizualizarea listei tuturor operelor de artă expuse în galeriile de artă sortată după anul realizării;   
❖ Filtrarea listei operelor de artă plastică după următoarele criterii: artist, tipul operei de artă;   
❖ Căutarea unei opere de artă după titlu.   
  
Utilizatorii de tip angajat al unei galerii de artă pot efectua următoarele operații după autentificare:   
❖ Toate operațiile permise utilizatorilor de tip vizitator;   
❖ Operații CRUD în ceea ce privește persistența operelor de artă expuse;   
❖ Vânzarea unei opere de artă din galeria de artă la care lucrează acel angajat;   
❖ Salvare liste cu situația operelor de artă în mai multe formate: csv, json, xml, txt.   
❖ Vizualizarea unor statistici legate de operele de artă utilizând grafice (structură radială, structură inelară, de tip coloană, etc.).  
  
 Utilizatorii de tip administrator pot efectua următoarele operații după autentificare:   
❖ Toate operațiile permise utilizatorilor de tip vizitator;   
❖ Operații CRUD pentru informațiile legate de utilizatorii care necesită autentificare;   
❖ Vizualizarea listei utilizatorilor care necesită autentificare. Interfața grafică a aplicației va fi disponibilă în cel puțin 3 limbi de circulație internațională.

2. Instrumente utilizate

Limbajul de programare - Java - Limbaj de programare de nivel inalt, orientat pe obiecte, potrivindu-se astfel cerintelor problemei.

Biblioteca - Java Swing - Biblioteca grafica pentru limbajul de programare Java, care ofera un set de componente pentru crearea de interfete grafice.  
-JFreeChart – Bazata pe Java Swing fiind folosita la creearea de grafice  
-FasterXml – Faciliteaza lucrul cu fisiere de tip xml

IDE - IntelliJ IDEA Ultimate - Mediu integrat de dezvoltare (IDE) pentru limbajul de programare Java, ce ofera instrumente pentru depanare si testare, ajuta utilizatorul in scrierea codului si include si tool-uri pentru generarea automata a diagramelor UML.

Baza de date - PostgreSQL

JDBC (Java Database Connectivity) - Este o interfata standard pentru conectarea la bazele de date relaționale din limbajul de programare Java.

Maven - Este un sistem de gestionare a dependențelor pentru proiecte Java, care faciliteaza integrarea cu alte proiecte și biblioteci Java si a fost folosit pentru gestionarea dependintelor pentru PostgreSQL si a testarii cu JUnit.

Version Control - Git – Ajuta la gestionarea modificarilor si permite dezvoltarea unor feature-uri diferite in paralel.

3. Justificare limbaj de programare ales

Am ales ca limbaj de programare Java datorita caracteristicilor acestora de limbaj de programare orientat pe obiecte, permitand astfel modelarea entitatilor constituente pentru dezvoltarea aplicatiei.  
 De asemenea, Java are suport pentru un numar foarte mare de librarii ce usureaza treaba programatorului in dezvoltarea aplicatiei.  
 Un alt motiv pentru care am ales sa folosesc Java este suportul pe care acest limbaj de programare il ofera in dezvoltarea aplicatiilor cu o arhitectura layered, asa cum este si arhitectura MVC.

4. Descrierea diagramelor UML

4.1. Diagrama use case

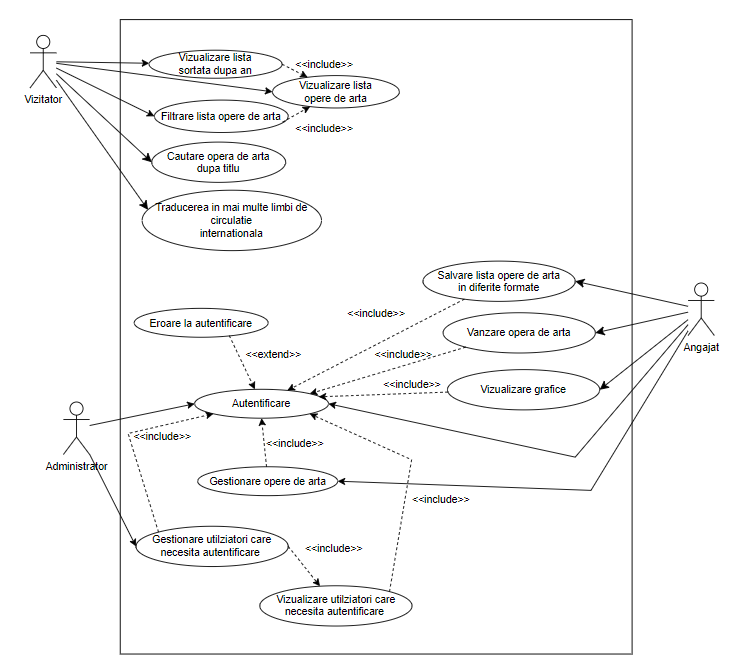


Figure 1 -Diagrama Use Case

Utilizatorii de tip Vizitator nu necesita autentificate si pot vizualiza lista operelor de arta, sa filtreze aceasta lista dupa numele artistului si tipul operei de arta, sa caute o opera de arta dupa titlu si sa sorteze operele de arta dupa anul realizarii acestora. De asemenea din pagina utilizatorilor de tip Vizitator este selectata limba de circulatie internationala care va fi aplicata tuturor paginilor.

Utilizatorii de tip Angajat necesita autentificare, care poate genera o eroare in cazul in care user-ul nu exista in baza de date, daca acesta nu a fost enabled de un administrator sau daca acesta a gresit parola. De asemenea, angajatii pot realiza toate operatiile pe care le realizeaza vizitatorii, operatii CRUD asupra operelor de arta dupa autentificare, vanzarea unei opere de arta si salvarea listei cu operele de arta in mai multe formate. Tot prin intermediul interfetei Angajat se pot genera grafice in formate diferite ce prezinta statistici legate de operele de arta.

Utilizatorii de tip administrator pot, asemenea angajatilor sa se autentifice, pot realiza aceleasii operatii ca si vizitatorii si pot realiza operatii CRUD asupra utilzatorilor care necesita autentificare.

4.2. Diagrama entity relationship

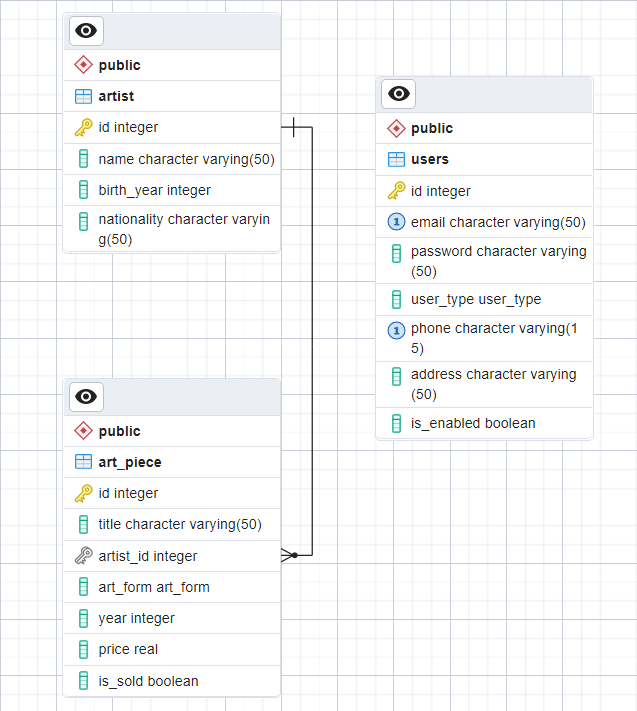


Figure 2 Diagrama Entity Relationship

Tabela Users modeleaza utilizatorii aplicatiei si contine datele necesare pentru identificarea unica a acestora, avand un id ca si cheie primara, si field-uri pentru email(unic), parola, tipul de utilzator, numarul de telefon, adresa si statusul de enabled a acestuia.

Tabela Artist contine un id unic, numele artistului, anul nasterii si nationalitatea acestuia.   
 Tavela Art\_piece contine field-uri pentru id, titlul operei de arta, un foreign key cu id-ul artistului ce referentiaza field-ul id din tabela artist, cat si field-uri pentru identificarea tipului de arta, a anului realizarii, a pretului si a informatiei legata de daca aceasta a fost vanduta sau nu.

4.3. Diagrama de clase

Clasele de model din pachetul model reprezinta entitatile din baza de date,   
existand cate o clasa pentru fiecare tabel. Clasele repository modeleaza lucrul cu baza de date, continand metode pentru operatii CRUD pe entitati. In cadrul claselor de service sunt instatiate cele aferente de repository, in aceste clase realizandu-se prelucrarea datelor obtinute din tabele. Tot in pachetul de model, in subpachetul service este si serviciul care se ocupa cu aplicarea traducerilor dintr-un fisier xml.  
 Pentru fiecare tip de User exista cate un View, de asemenea, existand view-uri pentru Sign Up si Log In. Fiecare clasa de View are si un Controller asociat in care adauga action listenere-le pentru fiecare componenta si gestioneaza efectele produse de acestea cat si evenimentele in care view-urile trebuie sa apara/dispara.  
 Intrucat toate controller-ele trebuie sa fie instiintate de modificarea limbii alese, acestea sunt adaugate in lista de observeri ale clasei observate, in acest claz TranslationService.



Figure 3 Diagrama de clase

4.4. Diagrame de activitati

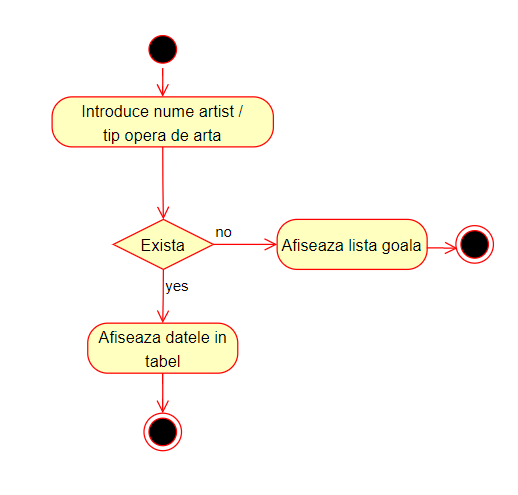


Figure 4 Diagrama Filtrare / Cautare Opere de Arta

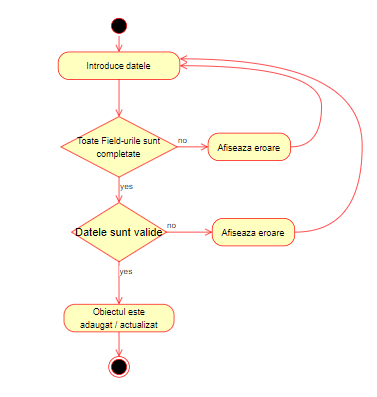


Figure 5 Diagrama Inserare / Actualizare Obiect

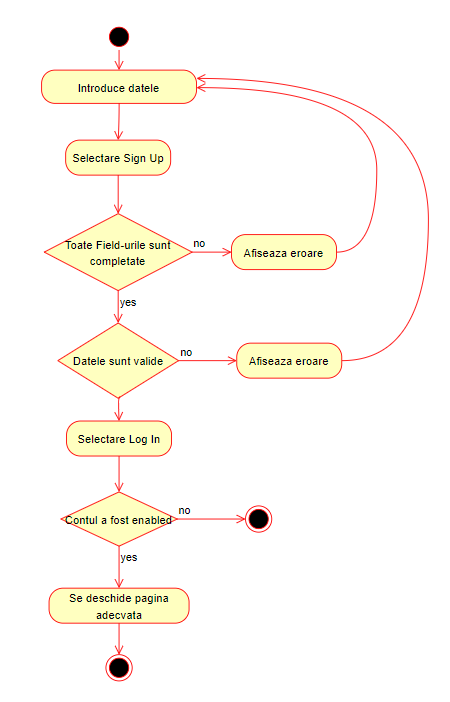


Figure 6 Diagrama accesarea unui nou cont

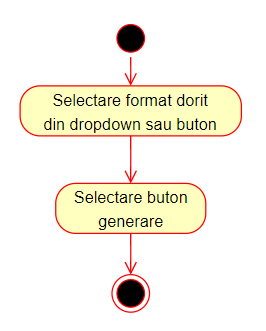


Figure 7 Diagrama generare rapoarte / chart-uri

5. Descrierea aplicatiei

Aplicatia se deschide direct cu pagina utilizatorilor de tip Visitor intrucat acestia nu necesita autentificare.  
In momentul deschiderii paginii este afisat un tabel cu operele de arta alaturi de numele artistului de care au fost create. Prin intermediul a 4 butoane vizitatorul poate vizualiza toate operele de arta, sa le sorteze dupa anul realizarii, sa le filtreze dupa numele artistului si/sau dupa tipul operei de arta si sa caute dupa titlu. Filtrarea cat si cautarea permit potrivirea partiala a parametrilor pentru a facilita cautarea. Tot in aceasta pagina este prezent si un meniu drop-down din care se poate selecta limba in care vor fi traduse toate paginile aplicatiei

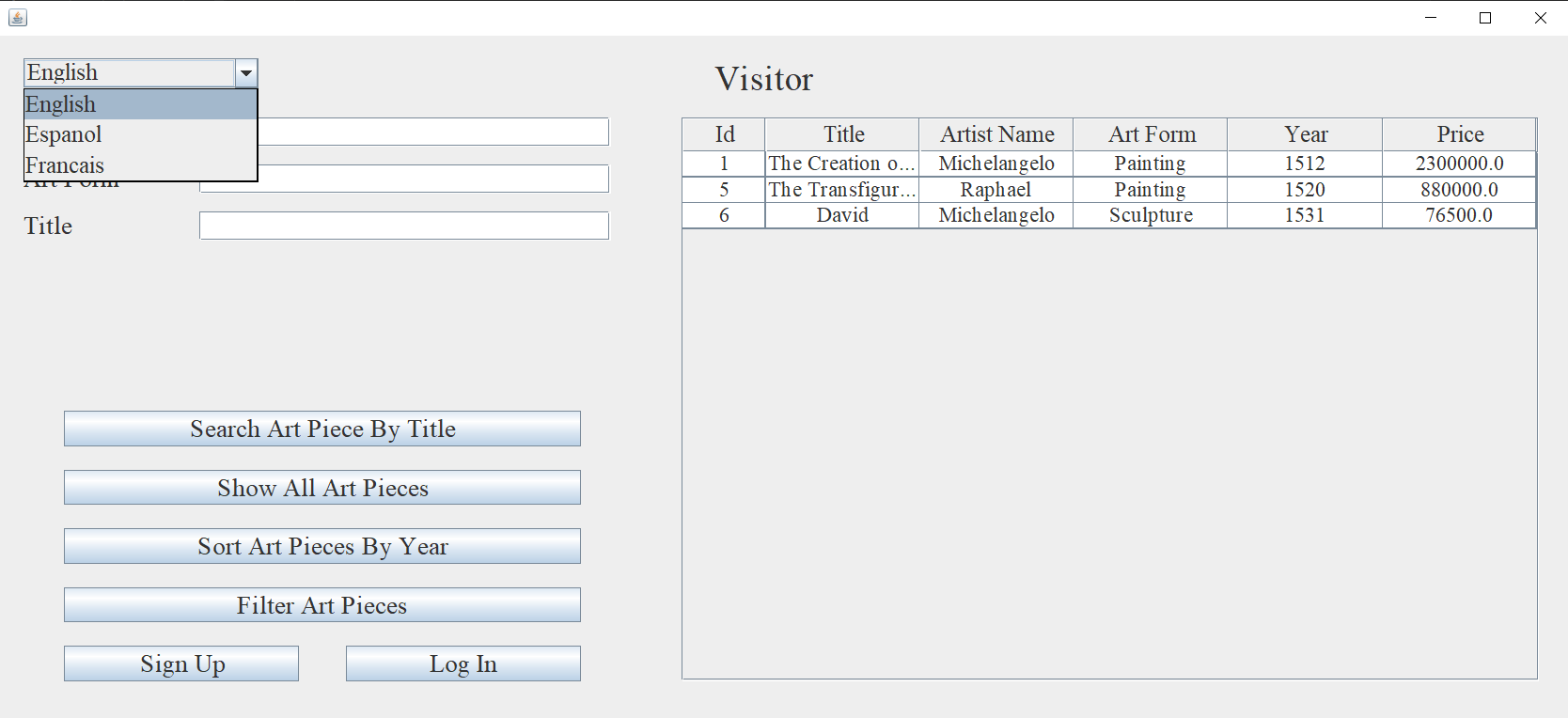


Figure 8 Pagina Visitor cu meniul pentru alegerea limbii

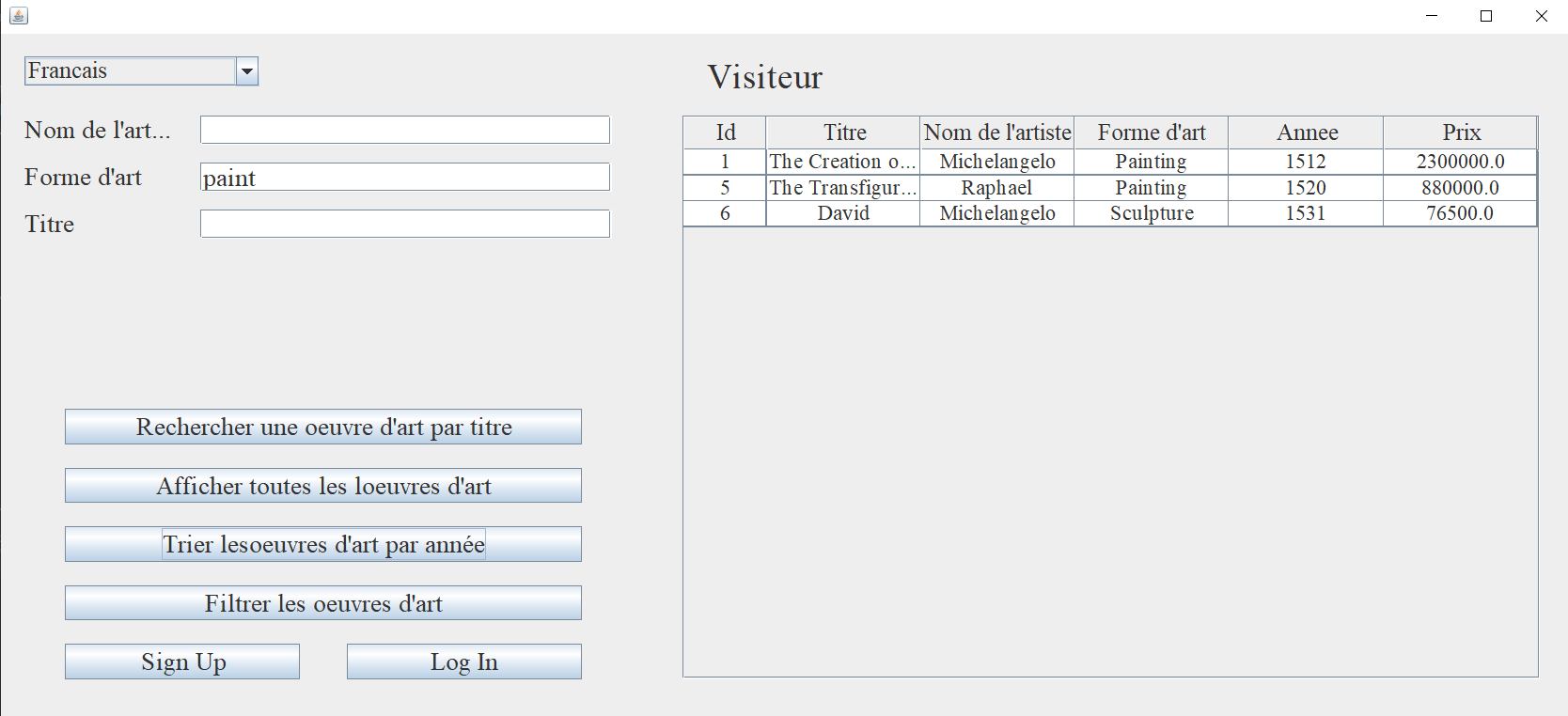


Figure 9 Pagina Visitor tradusa in Franceza cu rezultate filtrate

De asemenea, prezente pe aceasta pagina sunt butoanele de Sign Up si Log In ce permit creearea unui nou cont, respectiv accesul la un cont.

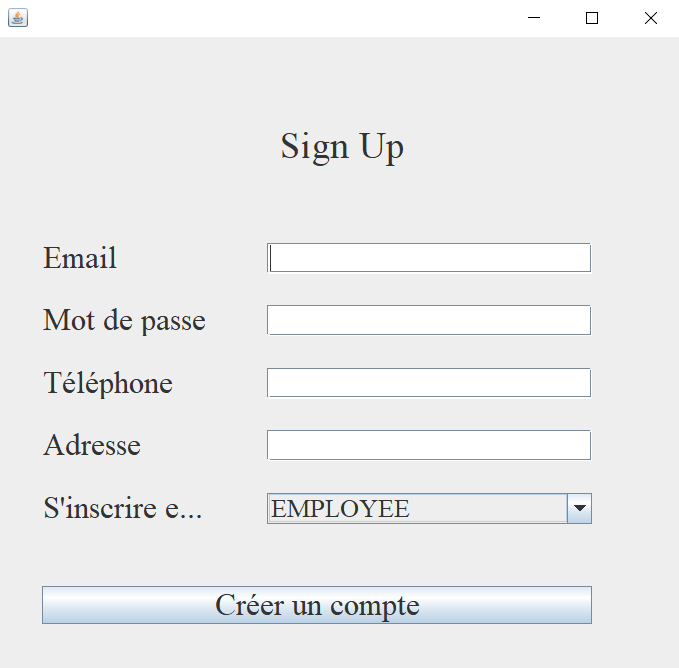


Figure 10 Pagina Sign Up

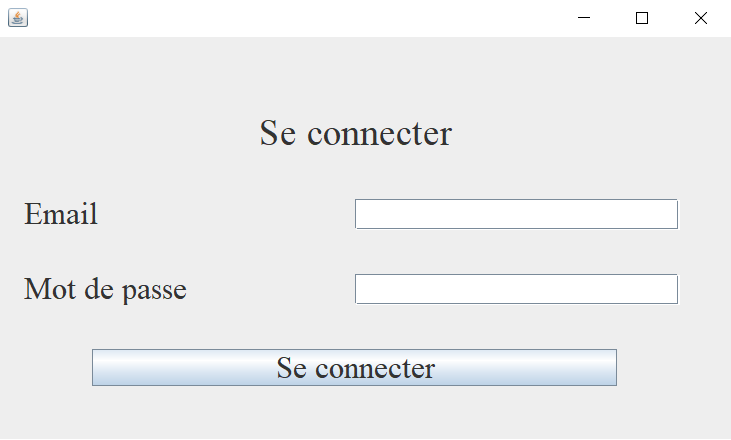


Figure 11 Pagina Log In

Pentru accesul la celelate view-uri utilizatorul trebuie sa faca operatia de Log In, aceasta rezultand in diferite erori in cazul in care email-ul nu a fost inregistrat, parola e gresita sau contul nu a fost enabled de catre un administrator.

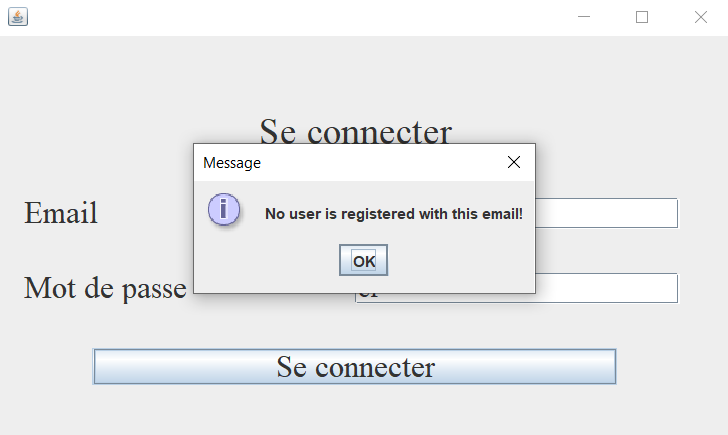


Figure 12 Eroare la Log In

View-ul pentru employee contine tabele pentru Artist si pentru Opere de Arta. Din acest view, employee-ul poate realiza aceleasi operatii ca si vizitatorul, cat si operatii CRUD asupra operelor de arta, si implicit asupra artistilor. De asemenea, din acest view employee-ul poate exporta liste cu operele de arta in diferite formate. Employee-ul este singurul care poate vedea si operele de arta vandute. La relizarea unui click pe unul dintre randurile tabelei, campurile aferente acestuia vor fi completate pentru a facilita editarea.

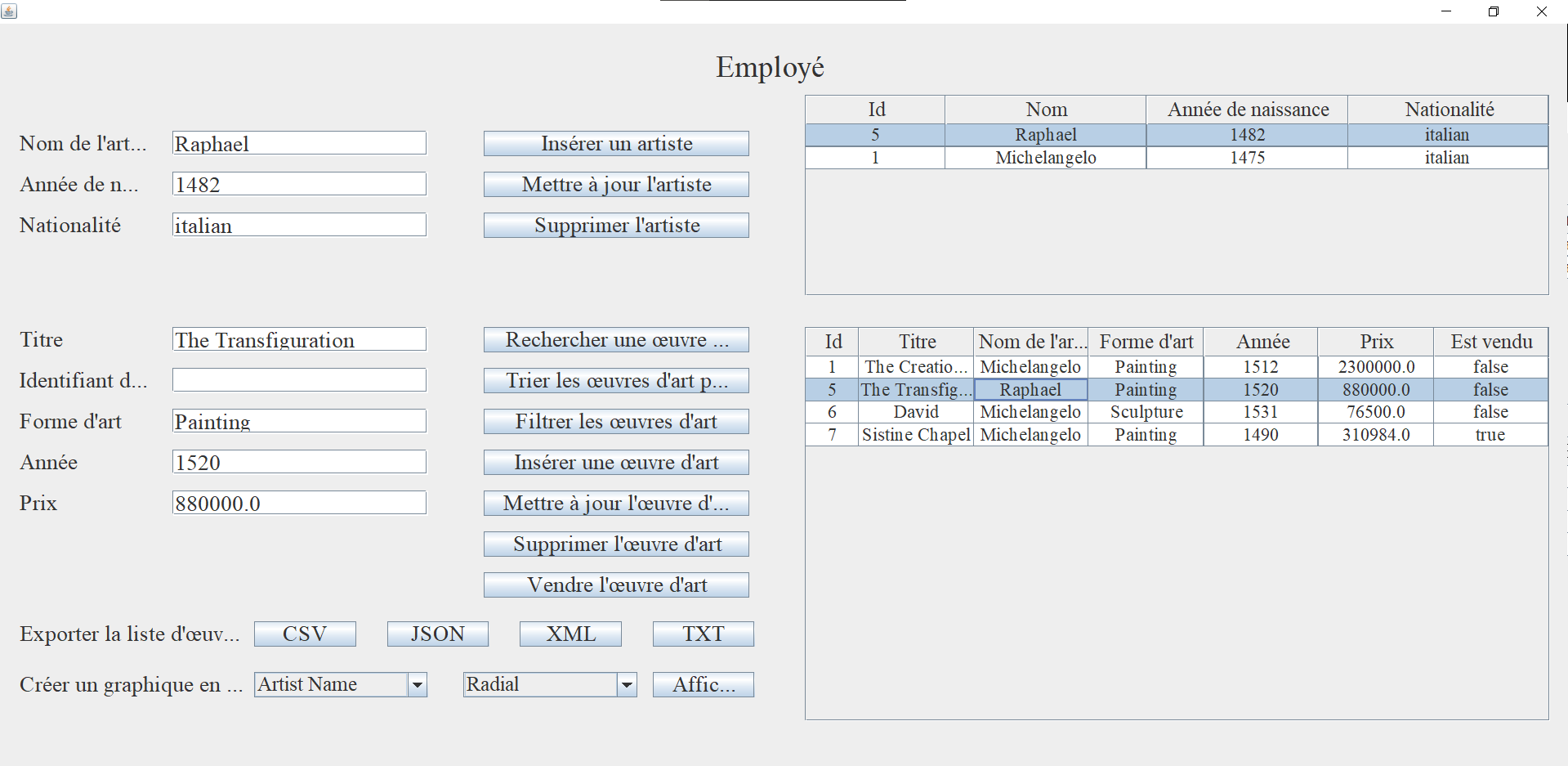


Figure 13 Pagina Employee cu randuri selectate din tabele

De asemenea, din cadrul acestei pagini, cu ajutorul a 2 liste drop-down se pot selecta parametrii ce vor fi utilizati pentru analiza unor statistici legate de operele de arta sub forma unor grafice, cat si structura acestor grafice

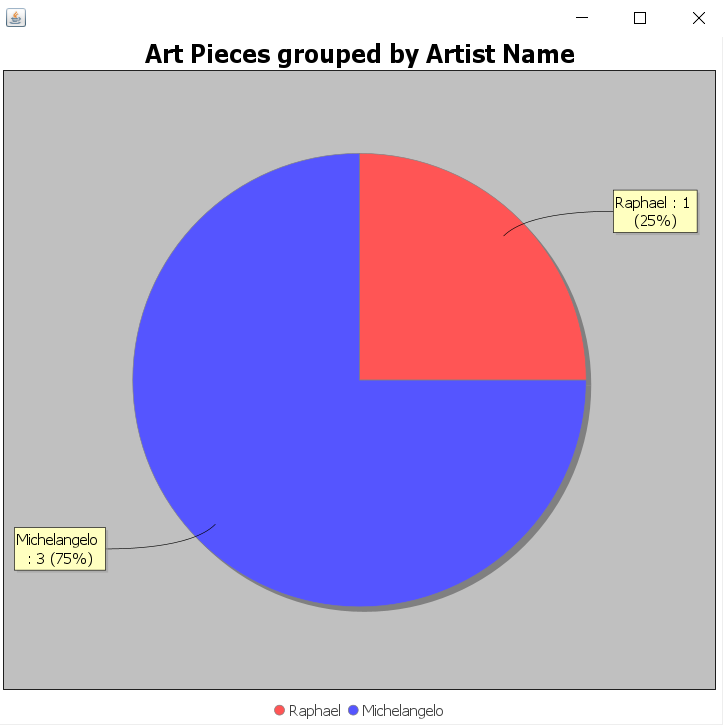


Figure 14 Chart ce prezinta procentajul operelor de arta clasificat dupa numele artistului

View-ul pentru administrator permite aceleasi operatii ca si cele ale vizitatorului, cat si vizualizarea listei utilizatorilor care necesita autentificarea si gestionarea acestora. Intrucat atunci cand se creeaza un cont nou, acesta nu este enabled, administratorul are rolul esential de a modifica aceasta valoare pentru a permite accesul ulterior la cont.

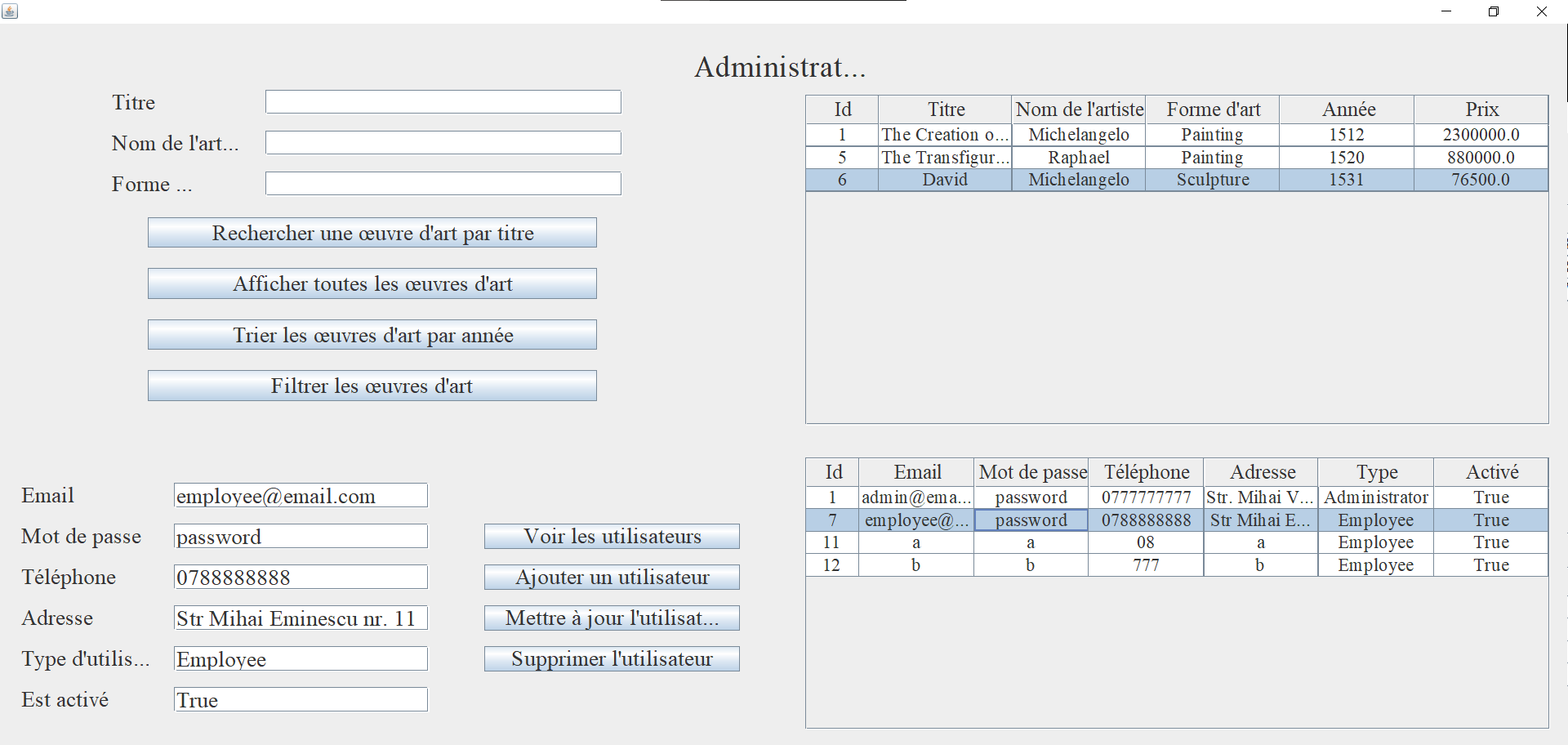


Figure 15 Pagina Administrator