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.

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.

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.

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.

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.

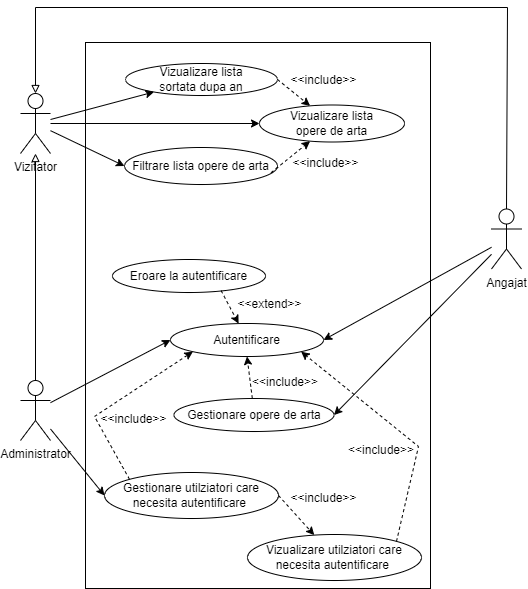
Version Control - Git – Ajuta la gestionarea modificarilor si permite dezvoltarea unor feature-uri diferite in paralel.

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 MVP.

4. Descrierea diagramelor UML

4.1. 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, si sa sorteze operele de arta dupa anul realizarii acestora.

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 cat si operatii CRUD asupra operelor de arta dupa autentificare.

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

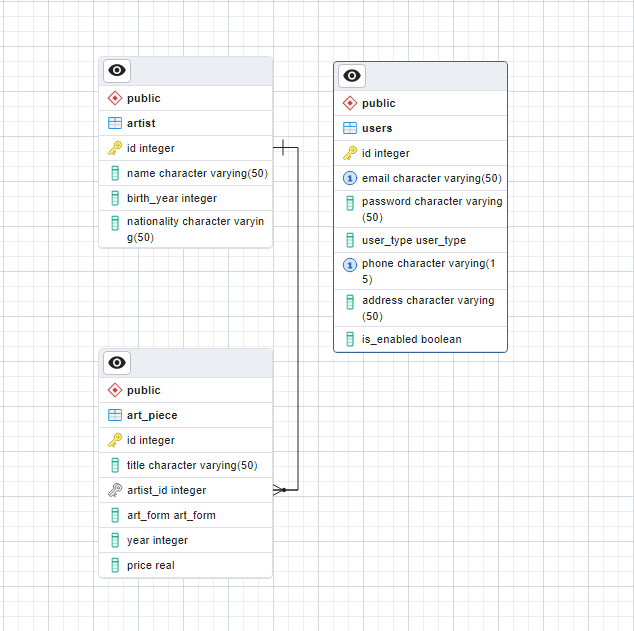
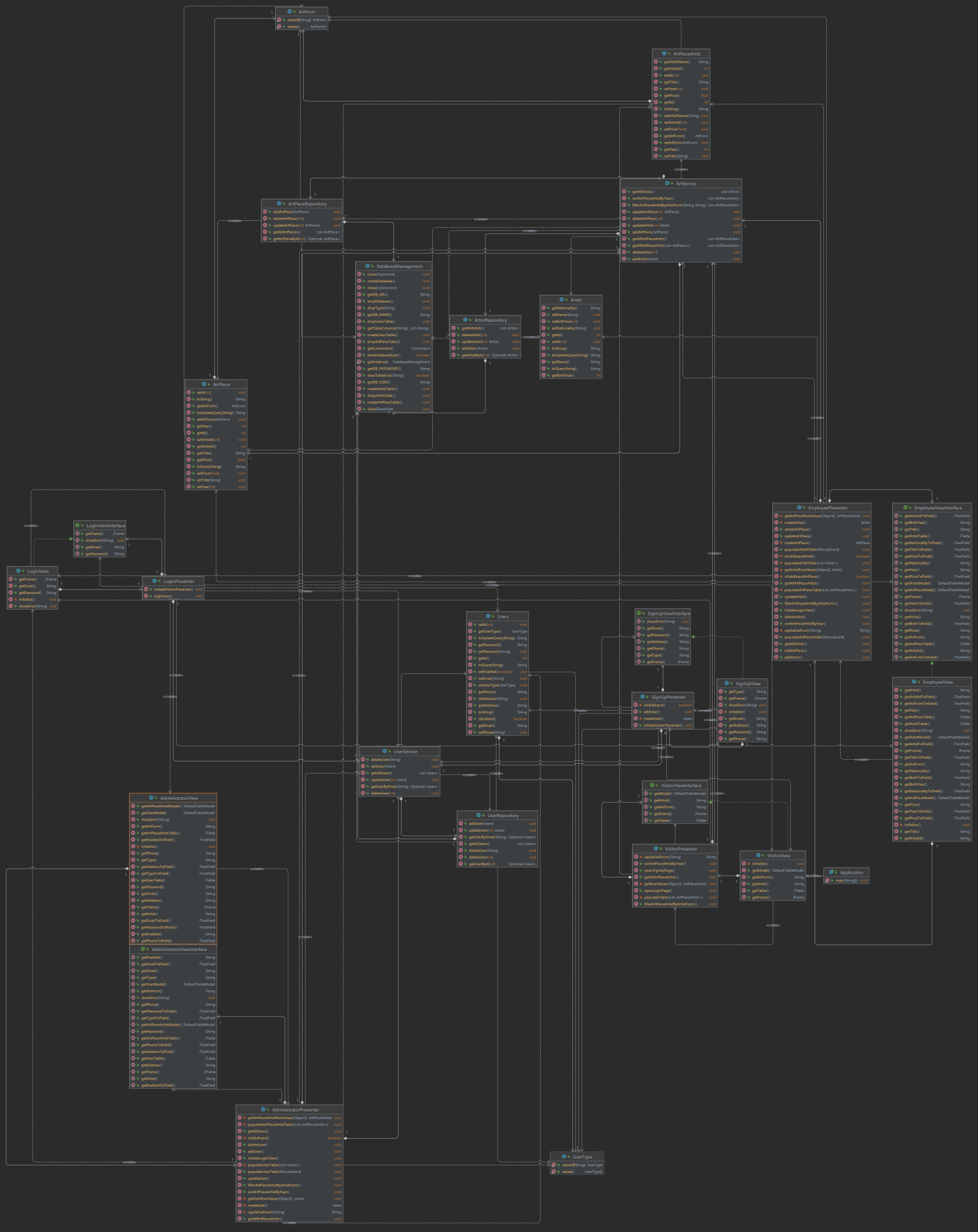


Tabela Users modeleaza utilizatorii aplicatiei si contine datele necesare pentru identificarea unica a acestora, avand un id ca si cheie primara, si field-uri pentru email(unic), parola, tipul de utilzator, numarul de telefon, adresa si statusul de enabled a acestuia.

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 si a pretului.

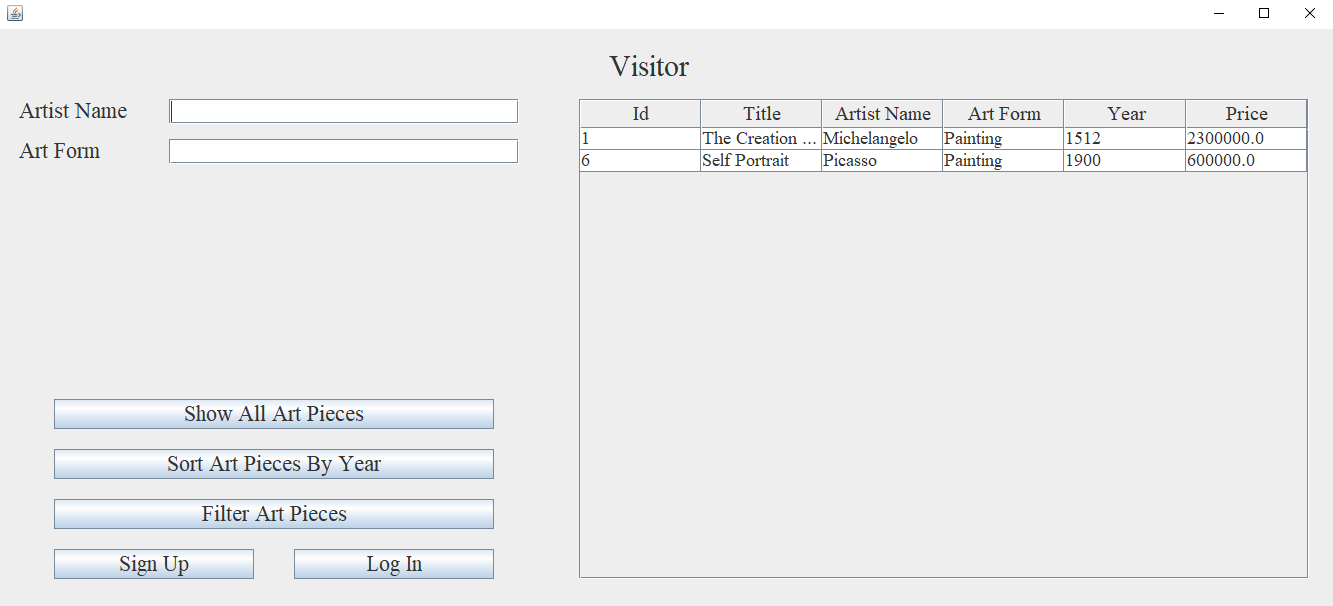
4.3. Diagrama de clase

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.  
 Pentru fiecare tip de User exista cate un View, de asemenea, existand view-uri pentru Sign Up si Log In. Clasele de View implementeaza cate o interfata care contine metodele pentru obtinerea datelor din interfata grafica. De asemenea, in view este instantiata o componenta Presenter corespunzatoare ale carei metode gestioneaza elementele interfetei grafice in urma producerii unui eveniment.  
Clasele de Presenter necesita ca parametru al constructorului o instanta a clasei care implementeaza interfata(clasa de View). De asemenea, in presenters sunt instantiate clasele de service pentru a putea obtine datele cu care se vor completa elementele interfetei grafice.

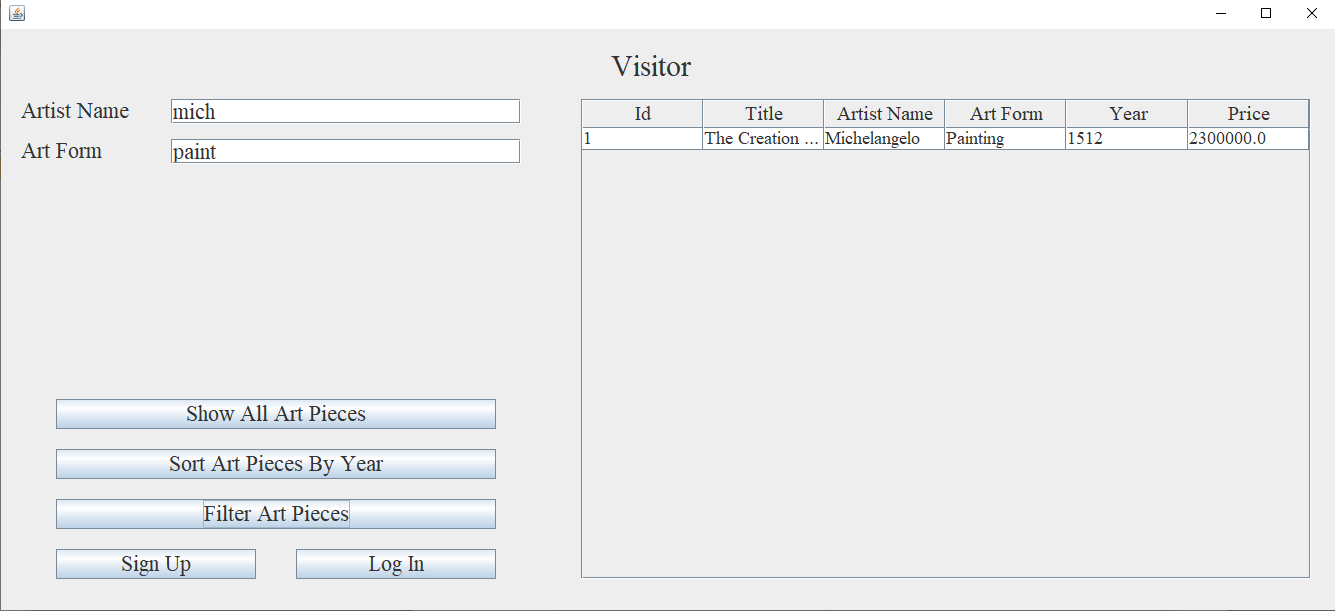


5. Descrierea aplicatiei

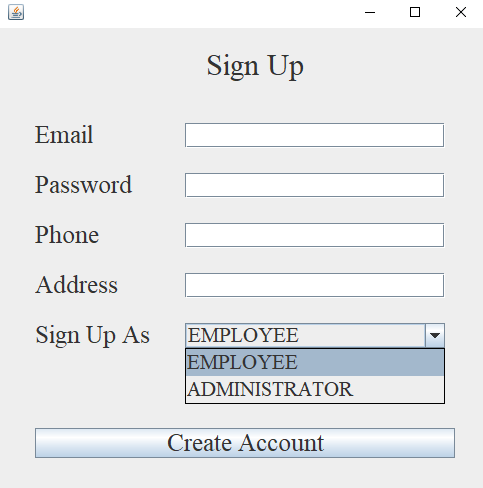
Aplicatia se deschide direct cu pagina utilizatorilor de tip Visitor intrucat acestia nu necesita autentificare.



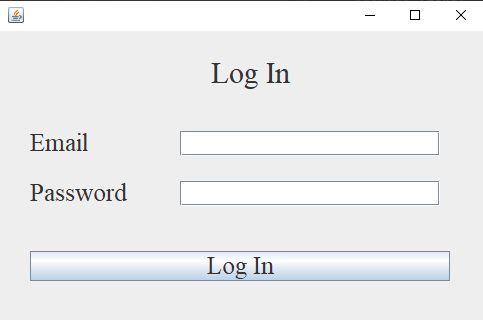
In momentul deschiderii paginii este afisat un tabel cu operele de arta alaturi de numele artistului de care au fost create. Prin intermediul a 3 butoane vizitatorul poate vizualiza toate operele de arta, sa le sorteze dupa anul realizarii si sa le filtreze dupa numele artistului si/sau dupa tipul operei de arta. Filtrarea permite potrivirea partiala a parametrilor pentru a facilita cautarea.



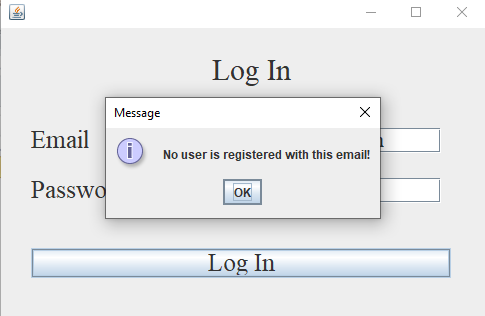
De asemenea, prezente pe aceasta pagina sunt butoanele de Sign Up si Log In ce permit creearea unui nou cont, respectiv accesul la un cont.

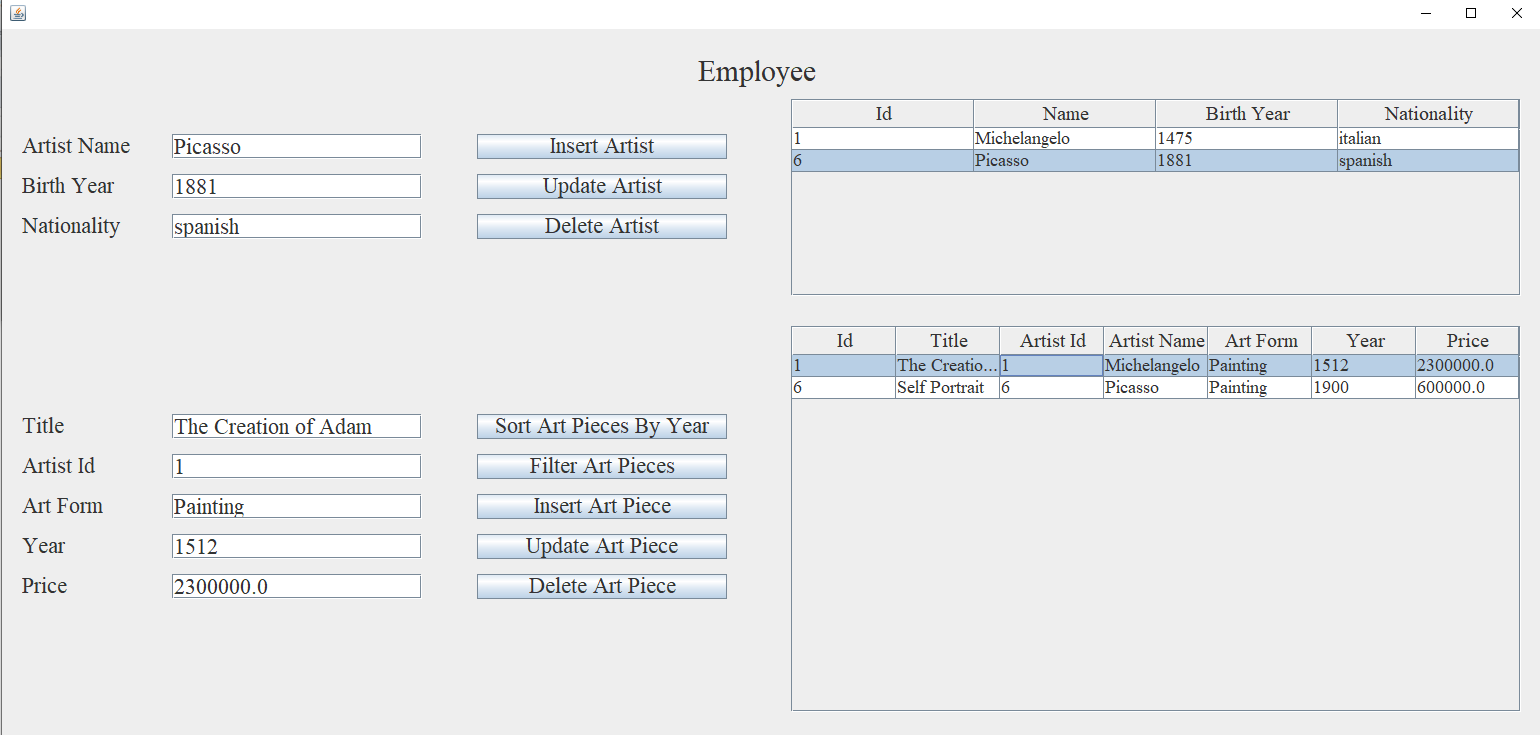


Utilizatorul ce doreste sa isi creeze un cont trebuie sa completeze informatiile legate de email, parola, numar de telefon, adresa si tipul de utilizator pe care il doreste, acesta putand alege dintre angajat si administrator.

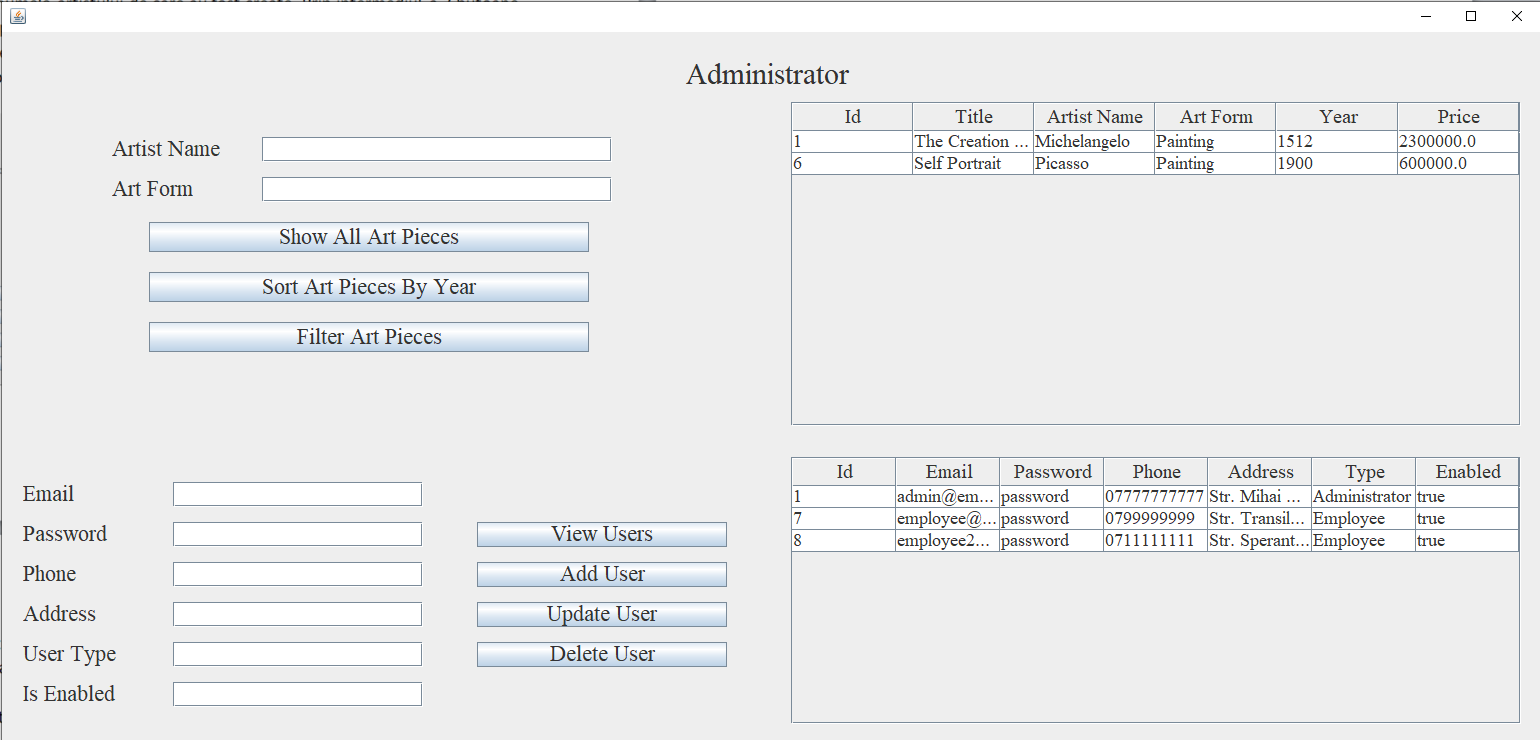


Pentru accesul la celelate view-uri utilizatorul trebuie sa faca operatia de Log In, aceasta rezultand in diferite erori in cazul in care email-ul nu a fost inregistrat, parola e gresita sau contul nu a fost enabled de catre un administrator.





View-ul pentru employee contine tabele pentru Artist si pentru Opere de Arta. Cand se face click pe unul dintre randurile tabelelor, se completeaza field-urile aferente tabelei pentru a facilita updatarea informatiilor. Din acest view, employee-ul poate realiza aceleasi operatii ca si vizitatorul, cat si operatii CRUD asupra operelor de arta, si implicit asupra artistilor.



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. Asemenea view-ului Employee, daca se face click pe un rand de tabel, se vor completa field-urile aferente acestuia.