

Programare III

Informatică Aplicată (anul 2)

Curmei Mihail

Problema

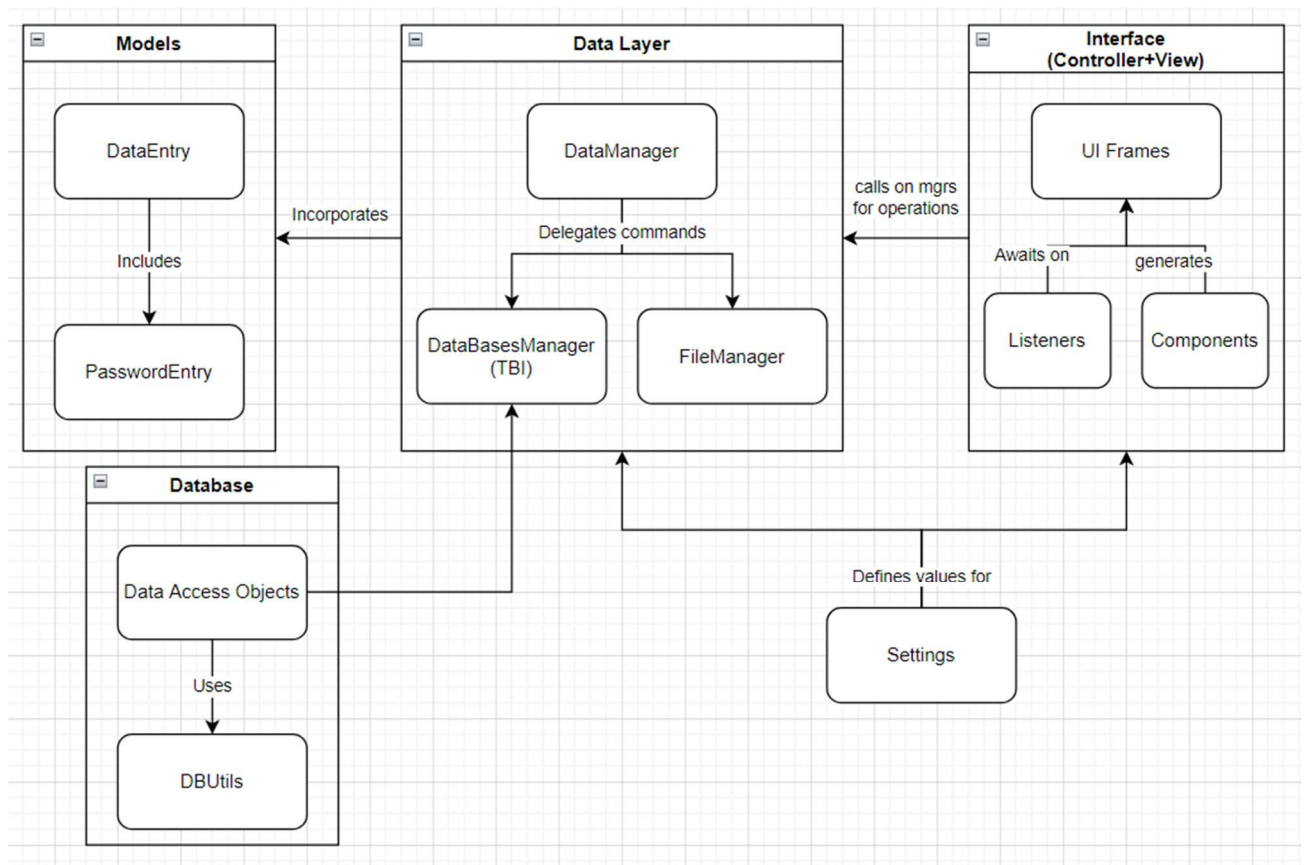
Numărul de pagini web pe care le accesez este în continuă creștere. Conform datelor din browser, am date cookies salvate din 520 de surse. Pentru a menține utilizatorul, majoritatea acestor pagini solicita înregistrarea cu un cont. Din motive de securitate, fiecare din aceste conturi necesită o parolă distinctă. La momentul dat, eu am 278 de conturi salvate, care teoretic toate au nevoie de o parolă proprie, sau chiar des un email sau logi diferit.

Acest proiect soluționează problema dată cu un Password Manager, ce generează, stochează, și organizează toate parolele și datele de acces ale utilizatorului.

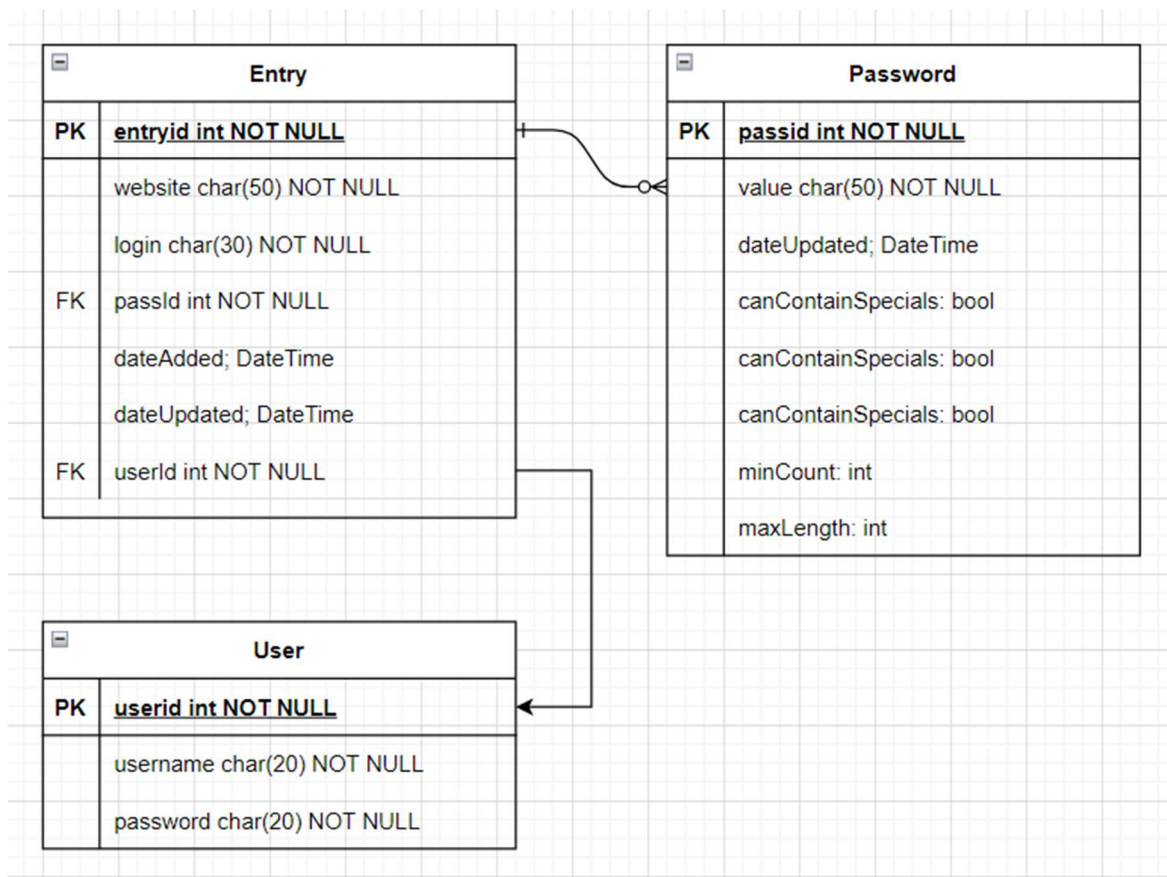
Use cases

1. Generarea unei parole după restricțiile date (lungimea maximă, caractere speciale, spații)
2. Salvarea unei parole pentru o combinație de pagină web și login
3. Lista tuturor conturi salvate
4. Posibilitate de a căuta datele credențiale din listă conform paginii web asociate.
5. Posibilitate de a copia o parole în clipboard

Structogram:



Baza de date:



Funcționalități:

Sistem de logare și înregistrare. Din motivul problemelor cu baza de date, este limitat la un singur utilizator. Lista de parole este de asemenea unică, indiferent de utilizatorul logat.

The image displays two user interface forms for a login and registration system.

Left Form (Login/Registration):

- Fields: `Username`, `Password`.
- Checkbox: `Show Password`.
- Buttons: `Login`, `Register`.

Right Form (Registration):

- Fields: `Username`, `Password`, `Repeat Password`.
- Button: `Register`.

Salvare și citire din fișier

Name	Login	Password	Last Updated
5 am test	thisisthelast	*****	Thu Jan 27 ...
final countd...	testoooo	*****	Thu Jan 27 ...
boy		****	Thu Jan 27 ...
TestyTester	Testeterone	*****	Thu Jan 27 ...
Google.com	I am da CE...	*****	Tue Jan 25 ...
Eternity OS	Lambda	*****	Tue Jan 25 ...
Edit Test Live	Editorial	*****	Thu Jan 27 ...
Eternity OS	Lambda	*****	Thu Jan 27 ...
Google.com	I am da CE...	*****	Tue Jan 25 ...

Filtrare și sortare

Name	Login
Edit Test Live	Editorial
Eternity OS	Lambda
Eternity OS	Lambda
Eternity OS	Lambda
Eternity OS	Lambda
Eternity OS	Lambda
Eternity OS	Lambda
Eternity OS	Lambda
Google.com	I am da CE...
Google.com	I am da CE...
Google.com	I am da CE...
TestyTester	Testeterone
boy	
final countd...	testoooo

Filter:

Sort: By Name ▼

Generarea unei parole noi

Test

×

Entry name

Login

Password

Generate Password

OK

Cancel

Password

cHCl9gJS%*8K

Password

al41&2T+xpli

Generate Password

livetest	helloclass	Ma!r14Ri*^7m
name	login	al41&2T+xpli

Editarea unei entități existente și copierea în clipboard

Edit entry for 'Eternity OS'



Name: Eternity OS

Login: Lambda

Copy

Password:

Copy

Show

Generate Password

Done

Cancel

Delete

Salvarea schimbărilor la închiderea ferestrei

```
this.addWindowListener(new java.awt.event.WindowAdapter() {  
    @Override  
    public void windowClosing(java.awt.event.WindowEvent windowEvent) {  
        dataManager.closeLibrary();  
    }  
});
```

Arhitectura:

Clasele de bază sunt foarte asemănătoare implementării lor în baza de date.

PasswordEntry este elementul parolei. Parola presupune pînă la 5 reguli, și este generată de biblioteca Passay. Deși editarea regulilor este implementată în backend, în frontend acces la ceasta nu există.

DataEntry este elementul complet. Acesta conține un nume, un id, date login, și parola *PasswordEntry*. Un utilizator proprietar este prezent în baza de date, dar nu în program, deoarece se presupune că utilizatorul are acces doar la datele sale, și scrierea aceluiași utilizator este redundantă.

Însuși utilizatorul era planificat să fie prezent, dar din motivul problemelor cu baza de date, el nu a fost implementat din conflict de priorități.

GenericDataManager este interfața ce necesită deschiderea și închiderea managerului, crearea, editarea și ștergerea entităților sale. Aceste funcționalități au fost decise pentru a acoperi cît mai multe funcții de prelucrare atît a fișierelor, cît și a bazei de date.

FileManager implementează această interfață pentru fișiere.

DatabaseManager implementează interfața pentru bază de date. Este funcțional doar teoretic.

DataManager deleghează comenzi celor doi Manageri și rezolvă conflictele între ei.

DBUtil crează o conexiune cu baza de date. Nu este implementat complet de mine, și nici funcțional.

BaseDao interfața ce instruește clasele să includă un insert, un update, un delete, un getOne, și getAll

PasswordEntryDao implementează operațiile pentru password cu PreparedStatement

DataEntryDao identic pentru *DataEntry*

UserEntryDao neimplementat

PMButton, *PMLabel*, *PMTextField*, *PMPasswordField* – componente ce extinde J Components. Sigura diferență e mărimea fixă inițială.

EntryTable – extinderea *AbstractTableModel*. O implementare completă pentru tipul *DataEntry*.

Settings – toate valorile necesare stocate în valori statice.

LoginScreen – JFrame pentru logare în program

RegistrationScreen – JFrame pentru înregistrare în program

PassLibraryScreen – JFrame cu toate elementele principale ale programului