

ALBUM FOTO

Hurghis Gheorghe-Georgian

1309B

Profesor Coordonator: Tizu Matei-Victor

Proiectul PhotoAlbum este o aplicație desktop dezvoltată în Python, care permite utilizatorilor să-și organizeze și să-și gestioneze colecțiile de fotografii într-un mod intuitiv și eficient. Aplicația oferă funcționalități precum adăugarea și ștergerea de albume, încărcarea și vizualizarea imaginilor, adăugarea de comentarii la fotografii, precum și posibilitatea de a edita sau șterge comentariile existente. De asemenea, PhotoAlbum oferă o interfață de autentificare cu opțiuni pentru crearea de conturi noi și conectarea la conturile existente. Fiecare utilizator poate să-și organizeze fotografiile în albume, să adauge descrieri și să exploreze colecția lor de imagini într-un mod plăcut și ușor de utilizat.

Tehnologii folosite:

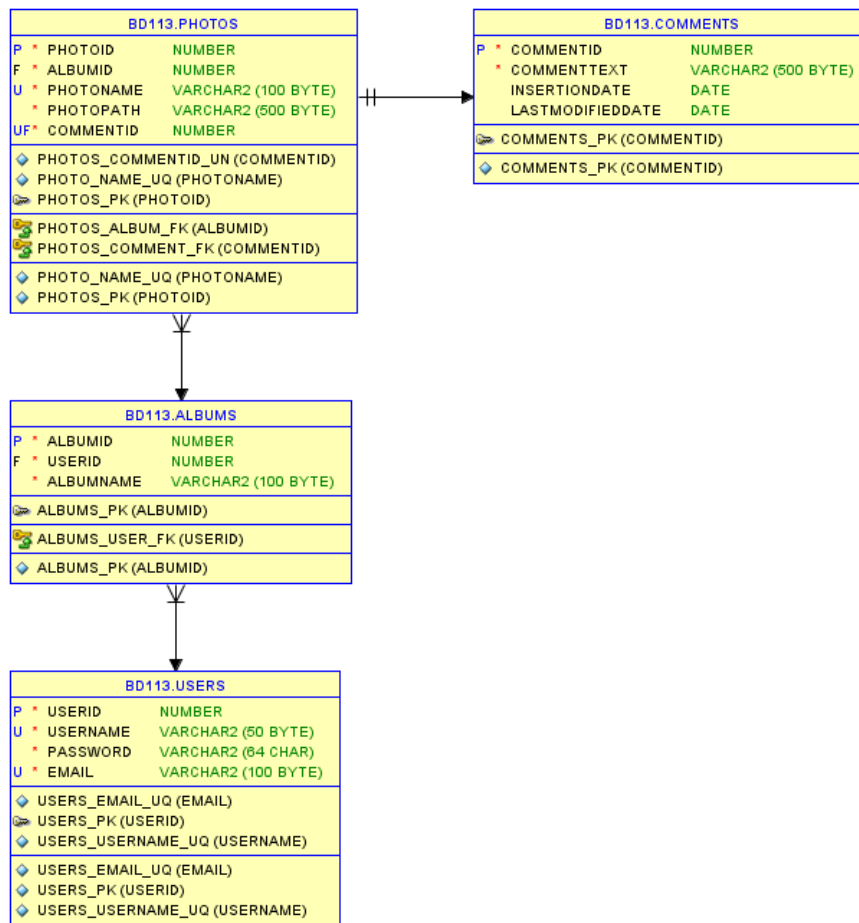
Back-end:

- Python
- Cx_Oracle
- Pillow
- Hashlib

Front-end:

- Tkinter

Structura si inter-relationarea tabelelor



Tabelul Users:

Atribute:

UserID (cheie primară): Identificator unic al utilizatorului generat automat.

Username: Numele unic de utilizator.

Password: Parola utilizatorului (criptată).

Email: Adresa de e-mail a utilizatorului.

Constrângeri:

Users_PK: Cheie primară pentru UserID.

Users_Username_UQ: Unicitate pentru Username.

Users_Email_UQ: Unicitate pentru Email.

Users_Email_CHK: Verifică validitatea adresei de e-mail.

Justificare:

Asigură unicitatea utilizatorilor și a informațiilor de autentificare.

Protejează integritatea datelor și ajută la prevenirea duplicatelor.

Tabelul Albums:

Atribute:

AlbumID (cheie primară): Identificator unic al albumului generat automat.

UserID (cheie externă): Cheie străină către Users.UserID.

AlbumName: Numele unic al albumului.

Constrângeri:

Albums_PK: Cheie primară pentru AlbumID.

Albums_User_FK: Cheie străină pentru UserID.

Justificare:

Leagă albumele de utilizatori specifici.

Asigură coerența datelor și facilitează interogările legate de utilizator.

Tabelul Comments:

Atribute:

CommentID (cheie primară): Identificator unic al comentariului generat automat.

CommentText: Textul comentariului.

InsertionDate: Data și ora la care a fost adăugat comentariul deci inclusiv si poza respectiva acestuia.

LastModifiedDate: Data și ora ultimei modificări a comentariului.

Constrângeri:

Comments_PK: Cheie primară pentru CommentID.

Justificare:

Urmărește informațiile despre comentarii și leagă aceste comentarii de fotografii specifice.

Tabelul Photos:

Atribute:

PhotoID (cheie primară): Identificator unic al fotografiei generat automat.

AlbumID (cheie externă): Cheie străină către Albums.AlbumID.

PhotoName: Numele unic al fotografiei.

PhotoPath: Calea către imaginea fotografiei.

CommentID (cheie externă): Cheie străină către Comments.CommentID.

Constrângeri:

Photos_PK: Cheie primară pentru PhotoID.

Photos_Album_FK: Cheie străină pentru AlbumID.

Photos_Comment_FK: Cheie străină pentru CommentID.

Justificare:

Leagă fotografiile de albume, comentarii și asigură consistența și integritatea datelor.

Relatii One-to-Many:

- Albums si Users, indicand faptul ca un user poate avea mai multe albume
- Albums si Photos , indicand faptul ca un album poate detine mai multe poze

Relatii One-to-One:

- Photos si Comments, fiecare poza are propria si singura descriere

Normalizare:

Prima Formă Normală (1NF):

Toate tabelele tale sunt deja în 1NF, deoarece fiecare coloană conține o singură valoare și valorile sunt atomice.

A Doua Formă Normală (2NF):

Tabela Albums este în 2NF deoarece fiecare non-key column depinde complet de cheia primară (AlbumID). Nu există dependențe parțiale.

A Treia Formă Normală (3NF):

Tabela Comments este în 3NF deoarece nici o coloană non-key nu depinde de o altă coloană non-key. Datele sunt deja dezlegate și nu există dependențe tranzitive.

Am utilizat biblioteca cx_Oracle pentru a facilita interacțiunea între aplicația Python și baza de date Oracle. Această bibliotecă oferă funcționalități esențiale pentru executarea interogărilor și manipularea datelor.

```
con = cx_Oracle.connect("bd113", "113", "bd-dc.cs.tuiasi.ro:1539/orcl")
```

Login

Please enter details below to login

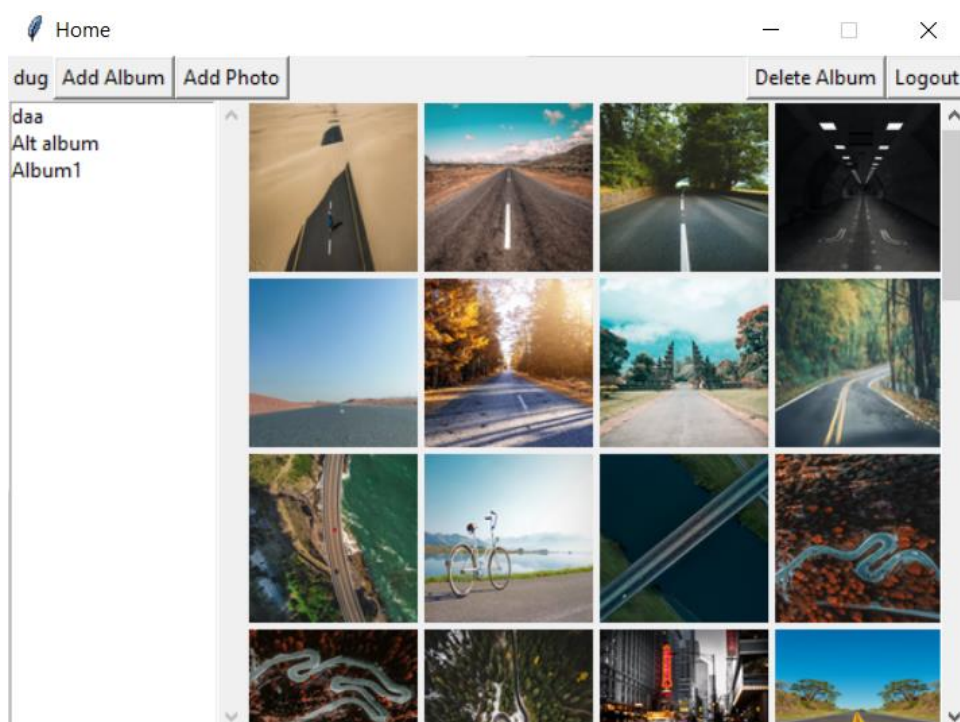
Username *

Password *

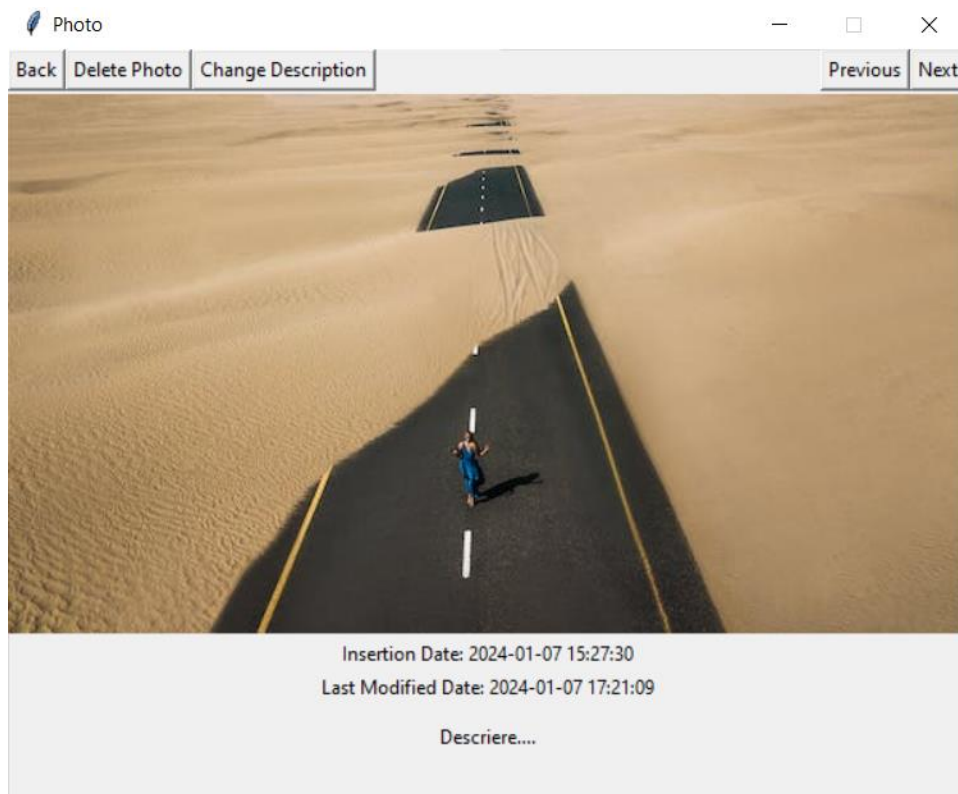
Login

Create Account

La rularea aplicatiei se deschide meniul pentru Log in , unde utilizatorul se poate conecta cu datele pe care le-a introdus anterior folosind butonul Create Account



Dupa logare se deschide meniul principal de unde utilizatorul poate crea albume, adauga fotografii in albumul selectat , sterge un album selectat. Deasemenea exista si butonul de Log out pentru deconectare.



Prin apasarea pe o imagine va apareea o fereastră nouă cu imaginea în rezoluția ei normală. Din acest meniu se poate șterge fotografia , schimba descrierea fotografiei (lucru care va schimba și data ultimei modificare) și vizualiza fotografia următoare/anterioară din album.

Tranzacții:

Stergerea unui album implică și ștergerea tuturor fotografiilor/comentariilor din album

Stergerea unei fotografii implică și ștergerea comentariului (descrierii) ei aferent

Adaugarea unei fotografii implica si adaugarea comentariului ei aferent

Exemplu cod pentru stergerea unei poze

```
def delete_photo(self):  
  
    # Delete the photo and its associated comment  
    try:  
        cursor = con.cursor()  
        con.begin()  
        cursor.execute("DELETE FROM Photos WHERE PhotoPath = :photo_path AND  
                        {'photo_path': self.photo_path, 'comment_id': self.comment_id}")  
        cursor.execute("DELETE FROM Comments WHERE CommentID = :comment_id",  
                        {'comment_id': self.comment_id})  
        con.commit()  
    # Commit the transaction  
    except Exception as e:  
        con.rollback() # Rollback changes if an error occurs  
        messagebox.showerror("Error", f"Error deleting photo: {str(e)}")  
    finally:  
        cursor.close()
```

In folder-ul cu proiectul/documentatia se afla si script-ul pentru crearea tabelor (creare_tabele.sql). Nu am facut si script pentru populare tabele deoarece aplicatia mea are si sistem de Logare (inclusiv cu criptare de parola) iar toate tabele sunt populate de catre utilizator la alegerea acestuia prin intermediul interfetei.