

Proiect Baze de Date 2

- *Platformă pentru crearea de texte alternative* –

Stuparu Maria Andreea

Grupa 342 C3

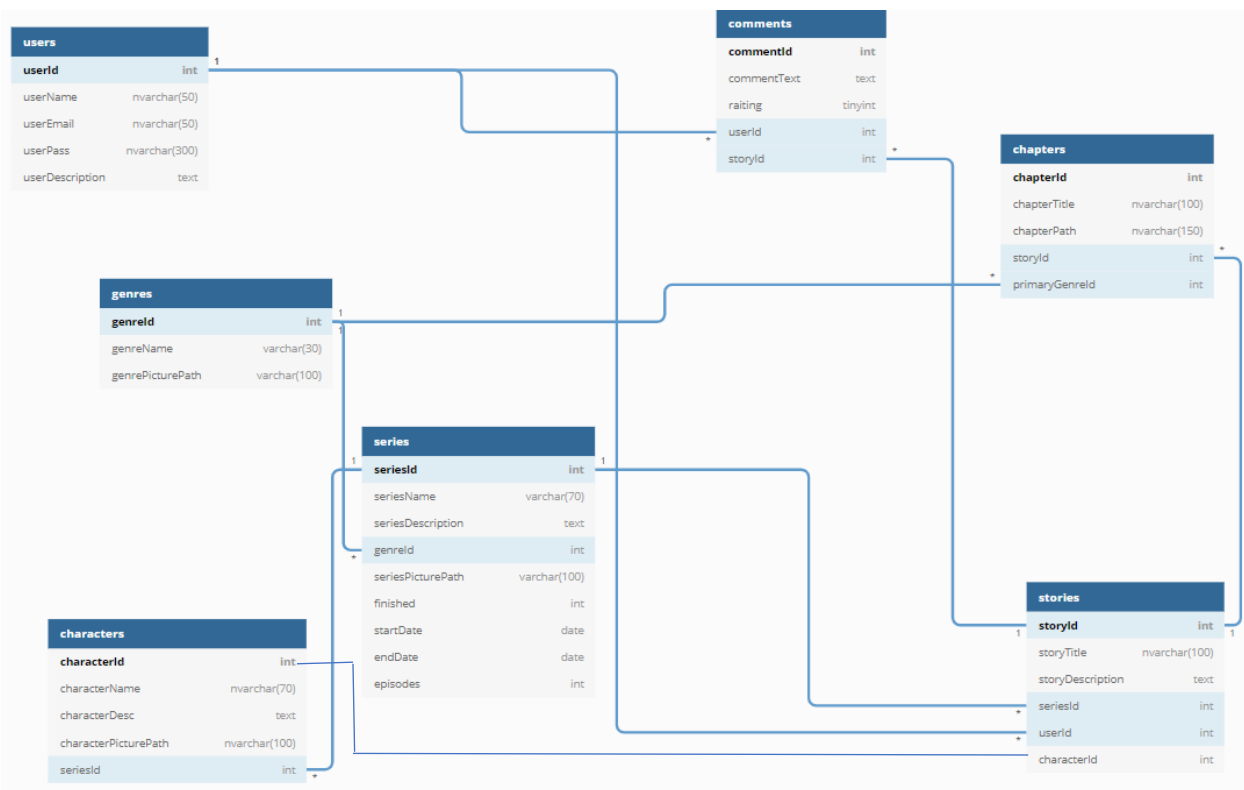
I. Descrierea aplicației

Adesea se întâmplă ca, la finalul unui serial celebru sau chiar în timpul difuzării acestuia, multe persoane care urmăresc serialul să înceapă să creeze diferite scenarii despre cum ar putea continua povestea sau despre cum și-ar dori să o modifice pe cea vizualizată. Proiectul prezentat este dedicat acestor persoane și constă într-o platformă care oferă posibilitatea de creare a unor povești ce au legătură cu un anumit serial și vizualizarea tuturor textelor proprii și pe cele ale celorlalți utilizatori.

Pentru crearea acestei platforme sunt necesare 3 componente principale ce vor fi prezentate în continuare : o bază de date mysql, serverul implementat ca o aplicație Flask și clientul implementat ca o aplicație în Python ce folosește pe reprezentarea interfeței pachetul Flask.

II. Descrierea bazei de date

1. Diagrama bazei de date



2. Structura tabelor

Baza de date este formată din 7 tabele ce fac legătura între seriale, povești și utilizatori.

Există doar două tabele care nu conțin ca attribute id-uri corespunzătoare altor tabele :
tabelele „users” și „genres”.

Tabela principală este cea numită „series” și ea are ca attribute principale coloanele `seriesName`, `genreId` și `seriesPicturePath`, care sunt folosite în cadrul interfeței. Un serial are asociate mai multe povești ce reprezintă intrări în tabela „stories”, o poveste ținând id-ul utilizatorului care a adăugat-o. O poveste are asociate mai multe capitole ce reprezintă intrări în tabela „chapters”, dar și mai multe comentarii corespunzătoare tabelului „comments” și un id corespunzător unui caracter din tabela „characters”. O intrare în tabela „chapters” poate conține și un id din tabelul „genres”, în cazul în care capitolul curent are un gen dominant.

Toate câmpurile tabelelor ce conțin cuvântul „Path” în numele lor reprezintă căi către anumite fișiere. Ele sunt folosite pentru a avea acces la imaginile corespunzătoare anumitor intrări fără a se stoca imaginea efectivă în baza de date. În acest fel imaginile pot să fie modificate ușor, fără a se modifica în nici un fel conținutul bazei de date. Un alt caz în care se reține calea către un fișier este cel al conținutului capitolului, întrucât acesta poate avea o dimensiune mare și se poate dori modificarea frecventă a sa.

Pentru fiecare coloană care are ca valori căi către fișiere există un folder specific în care se țin fișierele corespunzătoare acelor căi.

3. Descrierea constrângerilor de integritate

Fiecare tabel are un câmp id de tip întreg asociat, care reprezintă **cheia primară** a tabelului respectiv și este generat automat.

Coloanele corespunzătoare numelor le utilizator, de serial, de gen sau de poveste au asociate constrângeri **cheie unică**. Același lucru este valabil și pentru căile în care sunt stocate imaginile și capitolele, dar și pentru email-ul unui utilizator. De asemenea, coloanele asociate numelor au asociate și constrângeri **not null**.

Id-urile ce fac legătura între două tabele au asociate constrângeri de **cheie străină** (ex: coloana „seriesId” din tabelul „stories” leagă o poveste de serialul căruia îi este asociată).

4. Descrierea procedurilor și funcțiilor

Pentru a evita scrierea de cereri SQL în interiorul aplicației Python, toate cererile sunt încapsulate în proceduri stocate specifice, serverul apelându-le pe acestea atunci când dorește să interogheze baza de date.

Procedurile care au ca scop returnarea de date folosesc cereri SELECT și subcereri pentru agregarea datelor dorite.

Procedurile care au ca scop inserarea de date noi fac verificări asupra input-ului primit pentru a verifica anumite detalii ale coloanelor corespunzătoare (ex: constrângerea de cheie unică) și efectuează insert-ul numai în cazul unui rezultat pozitiv al acestor verificări. După insert se efectuează imediat commit. Acest tip de proceduri setează în ultimul pas al execuției o valoare pozitivă pentru un parametru de răspuns („ok_response”) în cazul adăugării cu succes și valoarea 0 în caz contrar. Această valoare este folosită apoi de către server pentru a-i transmite clientului

dacă datele de intrare au fost corecte sau nu. Clientul afișează mesaje de eroare în cazul în care acest răspuns nu este pozitiv.

Baza de date conține și o procedură pentru ștergerea unei povești, această ștergere declanșând trigger-ul stocat în baza de date de tip before delete numit „before_delete_story” care șterge toate comentariile asociate povești care se dorește a fi ștearsă.

Baza de date conține o singură funcție care calculează media notelor corespunzătoare comentariilor unei povești primite ca parametru și returnează această valoare. Ea este folosită apoi în interiorul unei proceduri ce returnează informații despre o anumită poveste.

Exemplu de procedură :

```
DELIMITER //
CREATE PROCEDURE get_mostStories()
BEGIN
    SELECT s.seriesId, s.seriesName, s.seriesDescription, s.seriesPicturePath,
           (SELECT count(st.storyId) FROM stories st
            WHERE st.seriesId = s.seriesId
            GROUP BY st.seriesId)
    FROM series s
    WHERE
        (SELECT count(*) FROM series a
         WHERE
             (SELECT count(*) FROM stories st WHERE st.seriesId = s.seriesId)
         < (SELECT count(*) FROM stories st WHERE st.seriesId = a.seriesId))
        IN (0, 1);
END//
DELIMITER ;
```

Funcția :

```
DELIMITER //
CREATE FUNCTION get_grade(storyId int)
RETURNS INT
DETERMINISTIC
BEGIN
    DECLARE grade INT;
    SET grade = (SELECT AVG(c.raiting) DIV 10
                 FROM comments c
                 WHERE c.storyId = storyId GROUP BY c.storyId);

    RETURN grade;
END//
DELIMITER ;
```

Trigger-ul :

```
DELIMITER //
CREATE TRIGGER before_delete_story
BEFORE DELETE
ON stories FOR EACH ROW
BEGIN
```

```
DELETE FROM comments c WHERE c.storyId = OLD.storyId;
END//
DELIMITER ;
```

III. Descrierea aplicației

1. Descrierea componentelor

Aplicația ce accesează baza de date descrisă mai sus este formată dintr-un client și un server. Ambele componente sunt implementate în limbajul python3, serverul fiind reprezentat de o aplicație HTTP de tip Flask ce rulează pe portul 5000, iar clientul de o aplicație ce are atașată o interfață grafică Tkinter.

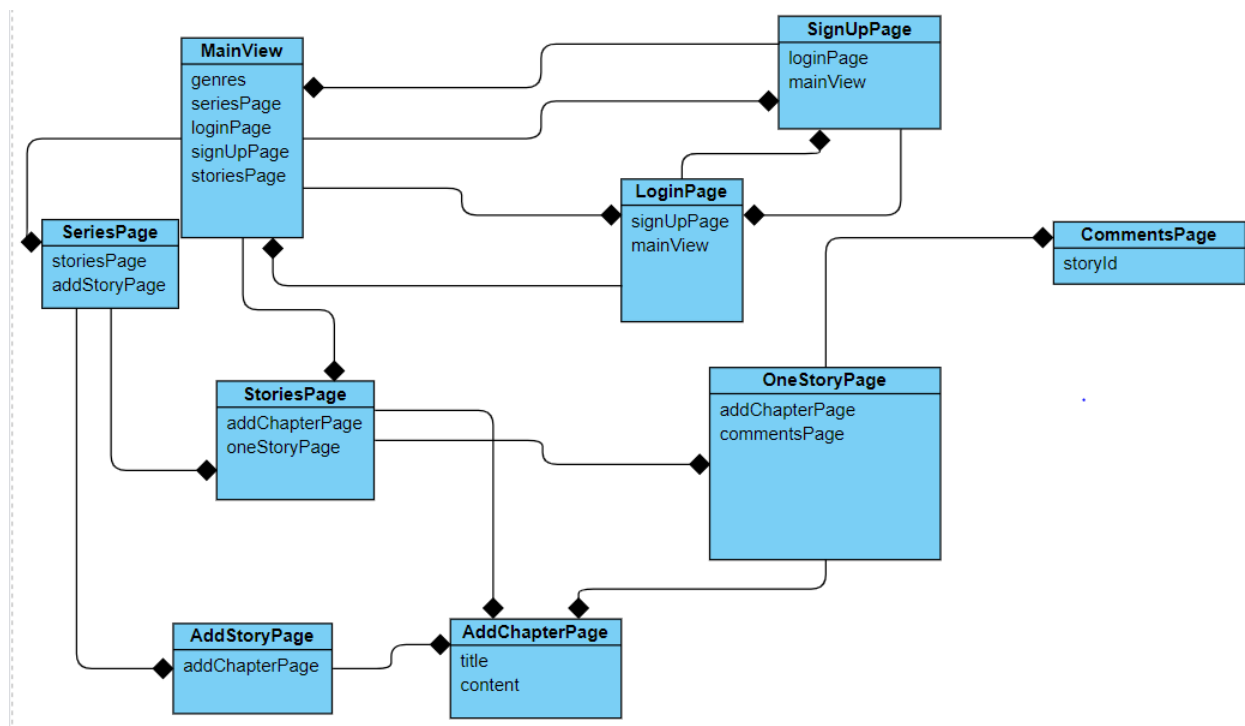
Întrucât serverul se ocupă doar cu interogările către baza de date, va fi prezentată în detaliu numai structura de clase ale clientului.

2. Digrama de clase

Toate clasele necesare interfeței grafice **moștenesc** clasa **tk.Frame** a pachetului Tkinter, astfel că ele se leagă unele de celelalte numai prin **agregare**. Trecerea de la o clasă la altă se face prin intermediul comenzilor **lift** și **destroy**, prima aducând frame-ul selectat în fața celorlalte, iar a doua ștergând frame-ul selectat.

Se poate considera că există 9 clase ce contruiesc structura de bază a aplicației, legătura dintre acestea făcându-se prin intermediul obiectelor ce sunt construite în interiorul fiecăreia.

Diagrama de clase prezentată mai jos ilustrează aceste clase și legături.



3. Structura claselor

Fiecare clasă inițializează un frame propriu folosind frame-ul părinte și creează elementele grafice specifice. Metoda principală a fiecărei clase este chiar cea corespunzătoare constructorului : `__init__`.

4. Diagrama de stări și fluxul de lucru

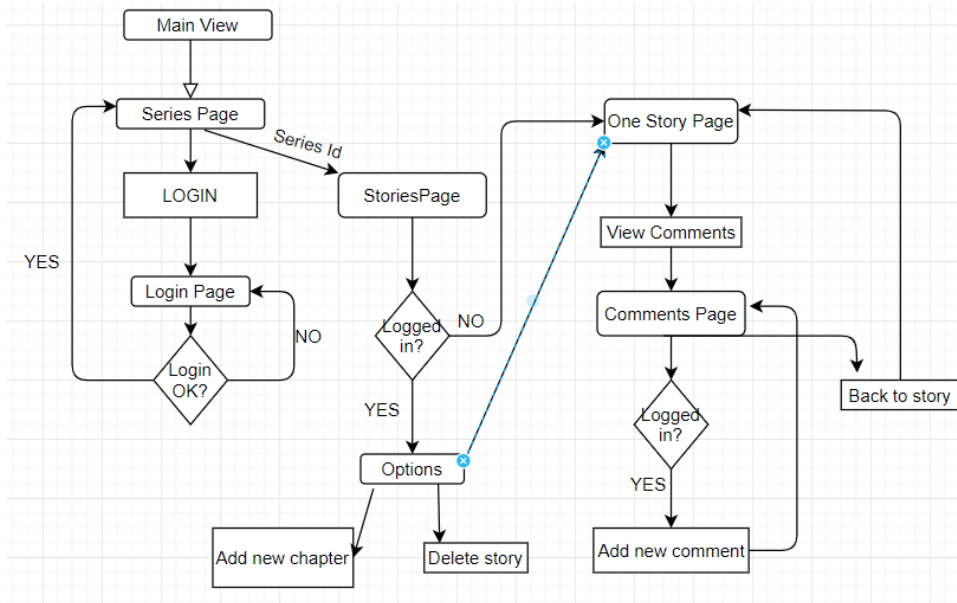


Diagrama de mai sus prezintă câteva dintre opțiunile pe care le oferă platforma și felul în care se face legătura între acestea.

Atunci când este pornită, aplicația afișează un ecran cu serialele pentru care s-au scris cele mai multe povești până în prezent. Dacă se dorește afișarea tuturor serialelor sau numai a celor de un anumit gen, se folosesc butoanele corespunzătoare acestor opțiuni.

Dacă se folosește butonul „LOGIN”, utilizatorul trebuie să se autentifice, iar după efectuarea cu succes a acestei operații, pentru serialele existente apare opțiunea de adăugare a unei povești noi.

Dacă se folosește butonul „SIGN UP”, se creează mai întâi un nou utilizator, apoi comportamentul este același ca în cazul login-ului.

În momentul în care se apasă pe imaginea corespunzătoare unui serial, toate poveștile asociate acestuia sunt afișate într-o nouă pagină. Dacă se selectează titlul unei povești, se deschide o nouă pagină în care este afișat primul capitol al acestei povești. Pagina are asociate butoane prin care se poate trece de la un capitol la altul sau se poate căuta un capitol specific. Dacă se dorește vizualizarea comentariilor asociate acestei povești, se apasă butonul „View comments”. În cazul în care există un utilizator logat, acest buton oferă și posibilitatea de adăugare a unui nou comentariu. Pentru poveștile corespunzătoare utilizatorului curent, se oferă și opțiunea de adăugare a unui capitol nou sau de ștergere a întregii povești.

În cazul **adăugării** conținutului **unui capitol**, clientul trimite către server textul acestuia, iar serverul creează în memoria fizică proprie un fișier de forma **<nume_poveste>_<nr_capitol>.txt**, numărul de capitole existente fiind aflat în urma unei interogări în baza de date. Serverul trimite apoi către baza de date doar numele fișierului în care este stocat capitolul.

5. Conexiunea cu baza de date

Conexiunea cu baza de date este făcută de către server, prin intermediul pachetului „mysql.connector”. Pentru a fi posibilă efectuarea unei conexiuni, este nevoie ca serverul să seteze numele bazei de date folosite, numele utilizatorului și parola acestuia. Pachetul „mysql.connector” oferă posibilitatea de apelare a procedurilor stocate în baza de date. Serverul trimite apoi către client răspunsul primit de la baza de date în format json.


IV. Vizualizarea aplicației


Aplicația este formată din 8 ecrane, dintre care unele păstrează elemente primite de la ecranul părinte. Aceste ecrane sunt : interfața de pornire, interfața corespunzătoare poveștilor despre un anumit serial, ecranul care afișează capitolele corespunzătoare unei anumite povești, ecranul care afișează comentariile asociate poveștii, cele două ecrane de adăugare a unei povești sau a unui comentariu nou și cele două ecrane de login și de sign up.


Când pornește aplicația, ecranul principal afișează serialele care au cele mai multe povești asociate și numărul de povești. Se poate selecta o altă categorie pentru seriale prin intermediul butoanelor din partea stângă, care sunt prezente în toate ecrane cu excepția celor de login și de signup. Această interfață poate să fie considerată și raport deoarece numărul de povești este primit sub forma de dată agregată (el este obținut în interiorul unei proceduri prin intermediul unei subcereri ce folosește funcția **count**).


- Ecranul de intrare -


ALL


















Login


Sign Up



Breaking Bad

When chemistry teacher Walter White is diagnosed with Stage III cancer and given only two years to live, he decides he has nothing to lose.


3 stories



Friends

Follows the personal and professional lives of six twenty to thirty-something-year-old friends living in Manhattan.

5 stories



Game of Thrones

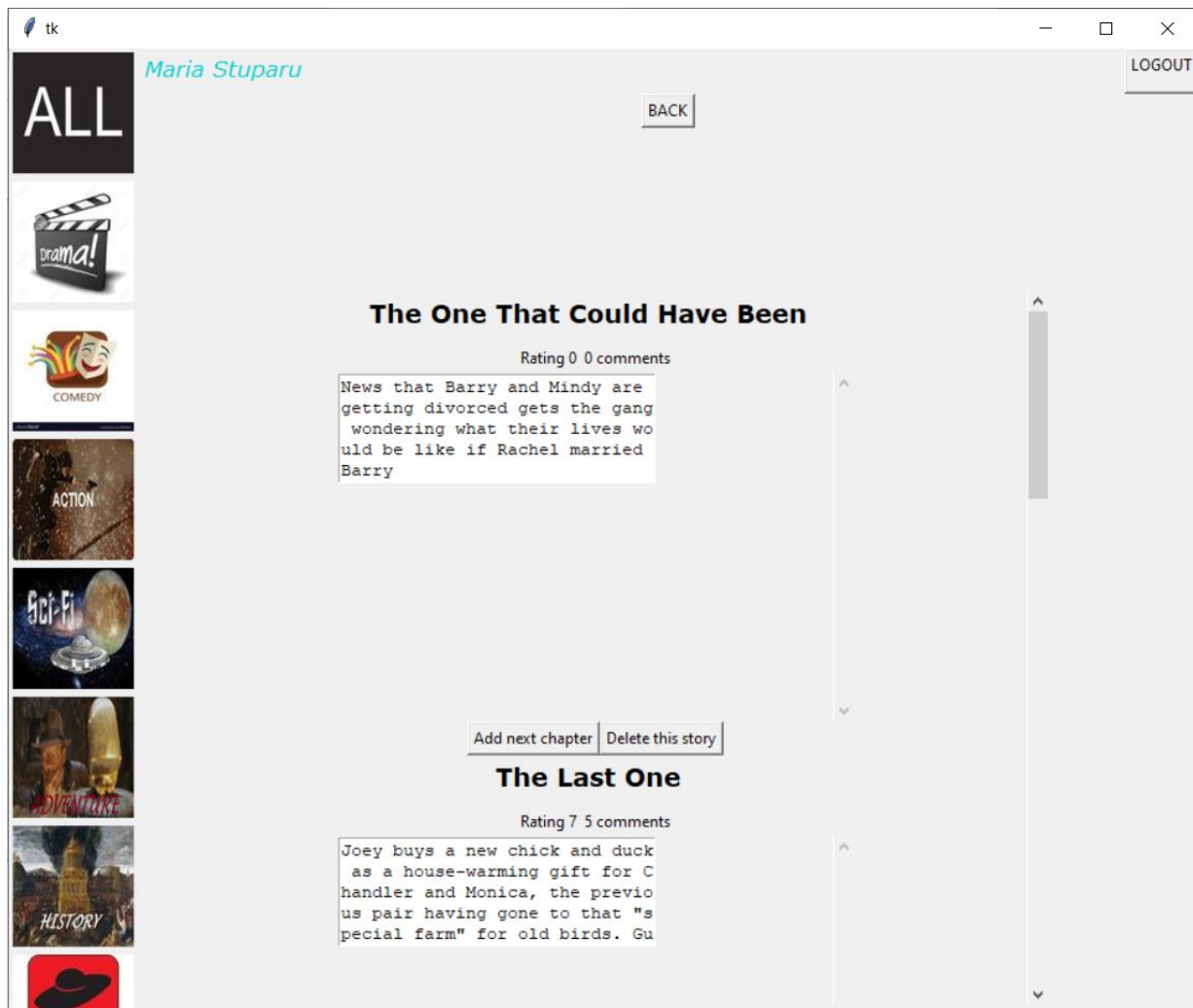
Nine noble families fight for control over the mythical lands of Westeros, while an ancient enemy returns after being dormant for thousands of years.

0 stories

- Ecranul corespunzător poveștilor -

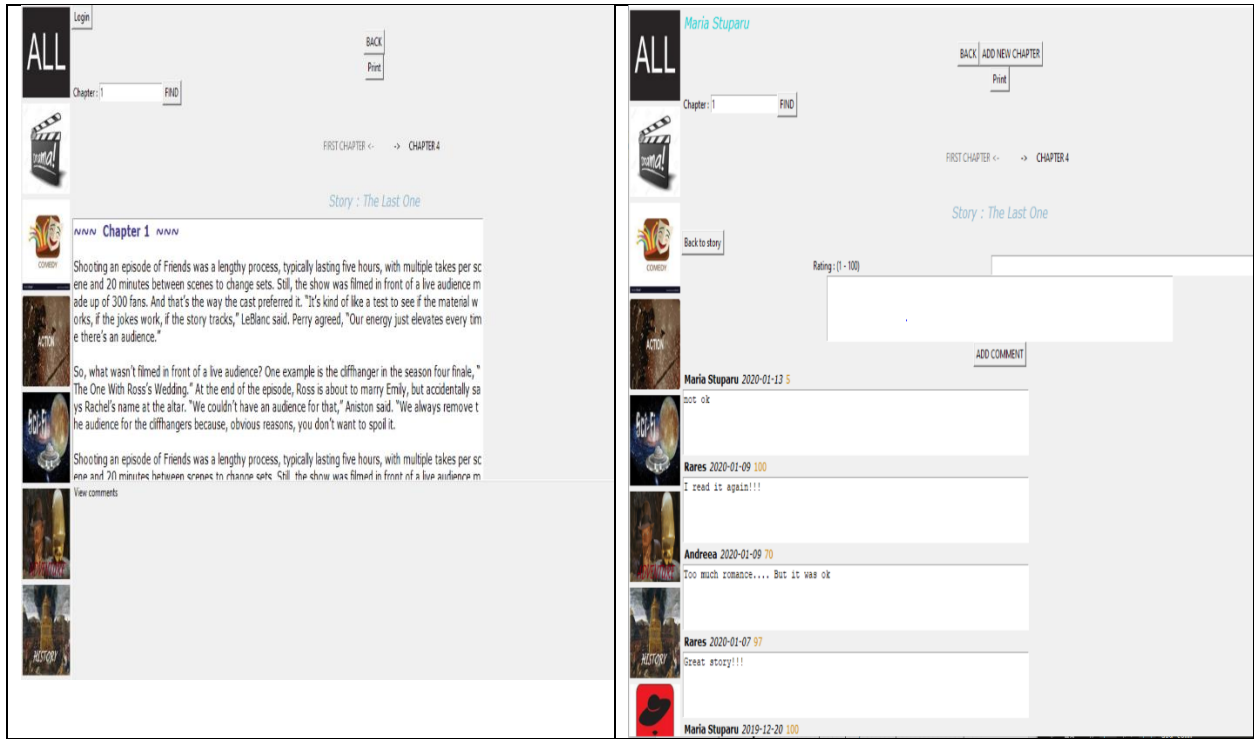
Dacă se selectează imaginea corespunzătoare unui serial, se deschide pagina din imaginea următoare. Această interfață poate fi considerată de asemenea un raport deoarece afișează ratingul mediu calculat din notele prezente în comentarii și numărul total de comentarii asociat unei povești.

În cazul în care utilizatorul este logat, acest ecran apare și dacă se selectează numele utilizatorului din colțul din stânga sus, însă el va afișa doar poveștile scrise de respectivul utilizator, cu opțiunile de adăugare a unui nou capitol sau de ștergere a întregii povești.



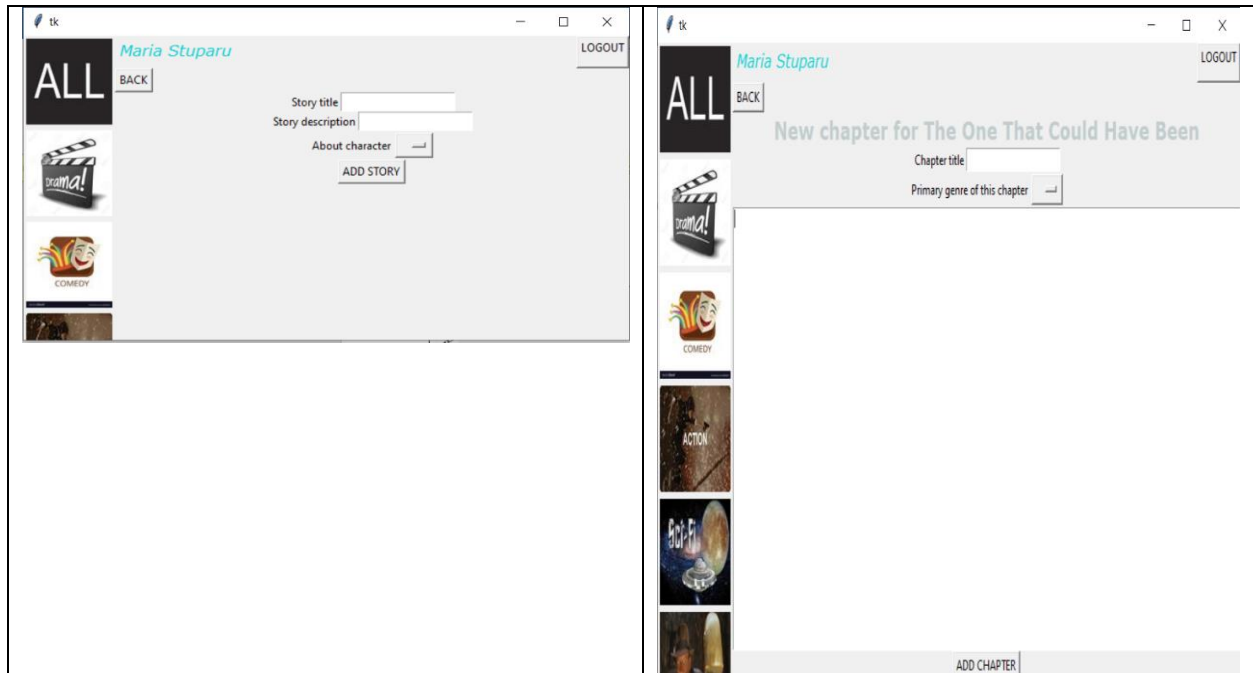
- **Ecranele corespunzătoare unei singure povești -**

Dacă se selectează numele unei povești, se trece la ecranul ce afișează capitolele acesteia. Acesta conține și opțiunea de vizualizare a comentariilor pentru această poveste (ecranul din dreapta). Pentru ca un nou comentariu să fie adăugat, este obligatoriu ca nota asociată să aibă o valoare între 1 și 100.



- Ecranele corespunzătoare adăugării de povești și de capitole noi -

În partea din stânga este prezentat ecranul corespunzător adăugării unei povești noi, în care numai câmpul de titlu este obligatoriu să fie completat. Opțiunea „About character” oferă posibilitatea de alegere a unui caracter dintr-o listă de tip drop-down în cazul în care serialul asociat are caractere introduse în baza de date.



V. Concluzii

Aplicația prezentată îi oferă utilizatorului posibilitatea de a avea acces la poveștile corespunzătoare serialelor preferate într-un mod structurat, de a împărtăși cu ceilalți propriile gânduri legate de o anumită poveste, dar și de a comunica cu cei pasionați de aceleași lucruri prin intermediul comentariilor.

Baza de date creată este structurată astfel încât aplicația să poată fi în viitor îmbunătățită cu opțiunile de adăugare a unui nou personaj, a unui nou serial, de modificare a unui capitol deja existent și de creare a unei pagini speciale pentru fiecare personaj existent.

VI. Bibliografie

1. Documentația Tkinter : <https://docs.python.org/2/library/tkinter.html>
2. TutorialsPoint Tkinter :
https://www.tutorialspoint.com/python/python_gui_programming.htm
3. Documentația Flask : <http://flask.palletsprojects.com/en/1.1.x/>
4. Documentația MySQL : https://docs.oracle.com/cd/E17952_01/index.html
5. MySQL-Connector în Python : <https://www.mysqltutorial.org/getting-started-mysql-python-connector/>
6. IMDB pentru informații despre seriale : <https://www.imdb.com/>