

Proiect - Etapa 1 - POO TV



- **Data publicarii: 25.11.2022, 00:16**
- **Data ultimei modificari: 12.12.2022, 15:53**
- **Deadline HARD: 17.12.2022, 23:59**
- **Responsabili:** Mihail Ungureanu [mailto:mihai2000.ungureanu@gmail.com], Narcis-Florin Căroi [mailto:narciscroi24@gmail.com], Mihnea-Andrei Bloțiu

[mailto:mihnea.blotiu@stud.acs.upb.ro], Mihai Soare [mailto:mihai_daniel.soare@stud.acs.upb.ro], Sorina Anamaria Buf [mailto:sorina_anamaria.buf@stud.acs.upb.ro], Eduard Marin [mailto:eduard.marin@stud.acs.upb.ro]

- **Actualizări:**
- 12.12.2022, 13:34: Au fost actualizate fisierele de ref
- 12.12.2022, 15:53: A fost actualizat deadline-ul

Obiective

- implementarea unui backend simplu al unei platforme specifice vizualizării de filme și seriale precum Netflix sau HBO MAX;
- dezvoltarea unor abilități de organizare și design orientat-obiect;
- scrierea unui cod cât mai generic, ce va permite ulterior adăugarea de noi funcționalități (cele specifice etapei a doua);
- folosirea unor design patterns potrivite pentru implementarea diferitelor task-uri;
- respectarea unui stil standard în momentul scrierii liniilor de cod și comentariilor;

Scenariu

În ziua de astăzi, platformele de vizualizare de filme sau seriale precum Netflix sau HBO MAX au luat o amploare extrem de mare atât în România, dar mai ales în întreaga lume.

În acest sens, pentru că știm că ți-ai dat silința de-a lungul întregului semestru pentru a vizualiza cele mai recente filme sau episoade din seriile favorite, vrem să îți oferim posibilitatea de a implementa o platformă similară cu scopul de a înțelege și funcționalitățile acesteia.

Te invităm să ajuți cât mai curând atât întreaga planetă pentru ca aceasta să poată vizualiza în continuare seriile și filmele preferate, cât și pe tine pentru că ne dorim să te bucuri alături de familie de sărbători și nu lucrând la proiectul de POO.

Context

Ne dorim implementarea unei platforme pentru vizualizarea de filme și seriale. Inițial ne gândim la câteva funcționalități standard, primele care ne vin în minte, pe care le vom propune noi după cum urmează: register, login, logout, search, view movie, rating, etc.

Rolul vostru este să implementați funcționalitățile acestei pagini din perspectiva unui utilizator și să îi transmiteți acestuia ce anume are voie și nu are voie să realizeze la un anumit moment de timp, pe platforma voastră.

Ulterior, în etapa a doua a proiectului ne vom da seama că platforma noastră a devenit foarte populară, iar cerințele utilizatorilor sunt mult mai mari decât posibilitățile acesteia. Astfel, vom adăuga noi funcționalități, drept continuare a celor deja existente.

- Etapa a doua a proiectului pleacă de la ideea realizării integrale a primei etape a proiectului;
- În acest sens, recomandăm scrierea unui cod cât mai generic ce permite adăugarea de noi funcționalități și implementarea cerințelor treptat, conform timeline-ului oferit de către echipa de proiect.

Execuția proiectului

- Se încarcă datele citite din fișierul de test (ce este în format JSON), în obiecte;
- Pentru prima etapă a proiectului se vor oferi inițial: lista cu userii deja înregistrați pe platformă (aceasta va putea fi modificată doar prin operația de register explicată ulterior) și lista filmelor existente pe platformă. Bineînțeles, fiecare dintre acestea vor fi identificate după caracteristicile proprii;
- Se primesc secvențial acțiuni („change page” sau „on page” pentru prima etapă) și se execută pe măsură ce sunt primite, rezultatul lor având efect asupra datelor din platformă la un anumit moment de timp;
- După executarea unor acțiuni (vom detalia ulterior care dintre acțiuni), se afișează rezultatul ei în fișierul JSON de ieșire;

- La terminarea tuturor acțiunilor se termina și execuția programului și se trece la următorul set de teste.

Descrierea modului de funcționare al platformei

Structura

Platforma va funcționa pe principiul unui **sistem de fișiere** după cum urmează:

- Fiecare test de input al platformei va începe de pe un așa numit „**Homepage neautentificat**”. Mai exact, vorbim despre o pagină standard, o pagină pentru un utilizator ce vrea să acceseze platforma însă momentan nu s-a înregistrat cu niciun cont pe platformă sau are contul său înregistrat, însă nu s-a autentificat cu acesta;
- În funcție de pagina pe care se află la un anumit moment dat un utilizator, acesta poate săvârși anumite acțiuni specifice;
- Sistemul este unul **secvențial**, adică la un anumit moment de timp, de pe o anumită pagină, se pot realiza doar două tipuri de operațiuni în această etapă, după cum urmează: se poate naviga pe pagina „următoare” (doar acele pagini care pot fi accesate de pe pagina curentă) sau se pot realiza operațiunile permise pe pagina curentă (numărul și specificul acestor operațiuni diferă de la pagină la pagină și vor fi detaliate pentru fiecare pagină în cele ce urmează);
- Orice operațiune care nu face parte din cele două descrise mai sus va fi semnalată printr-o eroare la output.** Un exemplu de o astfel de eroare ar fi: Nu se poate sări peste o anumită pagină (chiar dacă noi știm unde trebuie să navigăm, pentru a ajunge la destinație, trebuie luat totul pagină cu pagină, nu putem merge direct la destinație);

Vizualizarea și explicarea structurii

Aceasta este structura platformei pe care trebuie să o implementați:

```
Homepage neautentificat – Pagina de pornire în aplicație
  Login – Pagina de autentificare a unui utilizator deja existent
    Login – Acțiunea efectivă prin care facem login
  Register – Pagina de înregistrare a unui utilizator nou în sistem
    Register – Acțiunea efectivă prin care înregistrăm un nou utilizator

Homepage autentificat – Pagina după realizarea unui login sau register
  Movies – Pagina specifică filmelor
    Homepage autentificat
    Search
    Filter
    See details
    Homepage autentificat
    Movies
    Upgrades
    Purchase
    Watch
    Like
    Rate the movie
    Logout – Ieșire efectivă din cont + Homepage neautentificat
  Logout – Ieșire efectivă din cont + Homepage neautentificat
  Upgrades
    Homepage autentificat
    Movies
    Buy premium account
    Buy tokens
    Logout – Ieșire efectivă din cont + Homepage neautentificat
  Logout – Ieșire efectivă din cont + Homepage neautentificat
```

În cele ce urmează, explicăm puțin structura platformei noastre în prima etapă pentru a avea o privire de ansamblu asupra acestora:

- Nivelurile de imbricare ale platformei sunt cele ce se pot vedea în figură;
- Pe fiecare nivel de imbricare se pot observa una sau două culori diferite după cum urmează:
- Tot ceea ce este colorat cu galben reprezintă pagini pe care, pentru a le putea accesa facilitățile trebuie mai întâi navigat pe ele printr-o acțiune de tipul „**change**

page”;

- Tot ceea ce este colorat cu roșu reprezintă un eveniment care se poate realiza în pagina în care ne aflăm la un anumit moment de timp și se pot accesa printr-o acțiune de tipul „**on page**”.

Ca să înțelegem mai bine structura secvențială sub formă de sistem de fișiere explicată mai sus, ne vom uita pe figură și oferim niște exemple:

- În mod standard, toate testele vor începe pe pagina intitulată „**Homepage neautentificat**”;
- De aici am două opțiuni: pot să merg pe pagina „**Login**” și să mă înregistrez într-un cont deja existent, sau pot să merg pe pagina „**Register**” și să adaug un cont nou în baza de date a platformei;
- După ce am accesat una dintre aceste două pagini existente, trebuie să fac una dintre acțiunile disponibile pe pagina corespunzătoare: **acțiunea de login** sau **acțiunea de register**, în funcție de pagina pe care mă aflu;

Nu confundați paginile cu acțiunile care se pot realiza pe o anumită pagină;

Exemplu: **Pagina de login** și **acțiunea de login** sunt două lucruri **total diferite**: pagina de login face parte din ierarhia de pagini, iar acțiunea de login este un eveniment ce se poate realiza doar dacă mă aflu pe pagina de login. Altfel spus, pentru a realiza acțiunea de login, trebuie mai întâi să mă aflu pe pagina de login;

De asemenea, explicația imbricărilor pentru acțiuni este similară cu cea a paginilor. Adică, pentru a realiza o acțiune trebuie realizată mai întâi acțiunea părinte (dacă aceasta există). Exemplu: pentru a putea realiza acțiunea de „**Like**”, trebuie mai întâi realizată acțiunea de „**Watch**”. La rândul ei, pentru a putea realiza acțiunea de „**Watch**”, trebuie realizată mai întâi acțiunea de „**Purchase**”.

- Orice altă operațiune în afară de cele menționate mai sus precum: încercarea de a accesa o altă pagină de pe „**Homepage neautentificat**” în afară de „**Login**” sau „**Register**”, încercarea de a te autentifica cu un cont ce nu există în baza de date, încercarea de a înregistra un utilizator nou cu același nume precum unul deja existent în baza de date, etc. atrage după sine realizarea unei erori, va afișa la output mesajul corespunzător și **mă va întoarce automat pe pagina „Homepage neautentificat”**;

Cele menționate mai sus reprezintă doar câteva exemple referitor la operațiile posibile la un anumit moment de timp, sau la erorile ce pot apărea (mai exact, tratarea exhaustivă a variantelor posibile din „**Homepage neautentificat**”). Este datorită voastră să tratați toate cazurile posibile conform descrierii de sistem de fișiere al platformei pentru celelalte pagini. De asemenea, structura paginii din figura anterioară nu este finală. Aceasta este baza de la care va porni evident, etapa următoare a proiectului.

Workflow-ul unui user pe platformă

Vrem să înțelegem care sunt acțiunile posibile la un anumit moment de timp pentru un utilizator. Din acel moment vom ști că **orice altă acțiune nu este permisă**.

Sunt un utilizator al acestei platforme / aplicații. Primesc ca primă pagină „**Homepage neautentificat**”;

- De aici pot să navighez pe pagina „**Register**”;
 - De aici pot să apelez la funcționalitatea de register a acestei pagini;
- De aici pot să navighez pe pagina „**Login**”;
 - De aici pot să apelez la funcționalitatea de login a acestei pagini;

După ce realizez **cu succes** una dintre aceste două funcționalități voi fi mutat pe pagina „**Homepage autentificat**” de unde am acces la trei noi pagini:

- De aici pot să navighez în pagina de „**Logout**” care mă va scoate din utilizatorul cu care sunt autentificat în acest moment și mă va plasa în pagina „**Homepage neautentificat**”;
- De aici pot să navighez în pagina de „**Movies**” unde îmi va apărea o listă cu toate filmele de pe platformă ce nu sunt interzise în țara mea (explicăm la formatul de input cum ne dăm seama de acest lucru);
 - De aici pot să navighez către „**Homepage autentificat**”;
 - De aici pot să navighez către „**Logout**” (care a fost explicat deja);
 - De aici pot să navighez către pagina de „**See Details**” a filmului;
 - De aici pot să accesez și să mă întorc în pagina de „**Homepage autentificat**”;
 - De aici pot să accesez și să mă întorc în pagina de „**Movies**”;
 - De aici pot să accesez pagina de „**Upgrades**”;
 - De aici pot să cumpăr filmul prin acțiunea „**Purchase**”;
 - De aici pot să vizionez filmul prin „**Watch**” doar dacă l-am cumpărat mai întâi;
 - De aici pot să apreciez filmul prin „**Like**” doar dacă am vizionat filmul în prealabil;
 - De aici pot să ofer o notă de la 1 la 5 filmului prin „**Rate the movie**” doar dacă am vizionat filmul în prealabil;
 - De aici pot să apelez la funcționalitățile „**Search**” sau „**Filter**”. Acestea îmi vor filtra lista de filme disponibile în funcție de diferite criterii care sunt detaliate ulterior.
 - De aici pot să navighez în pagina de „**Upgrades**”.
 - De aici pot să navighez și să mă întorc către „**Homepage autentificat**”;
 - De aici pot să navighez către „**Movies**”;
 - De aici pot să navighez către „**Logout**” (care a fost explicat deja);
 - De aici pot să cumpăr moneda platformei prin acțiunea „**Buy tokens**”, monedă ce mă va ajuta ulterior să cumpăr filme pe platformă;
 - De aici pot să cumpăr cont premium prin acțiunea „**Buy premium account**” ce va avea anumite beneficii atât în această etapă cât și în cea de-a doua.

Format JSON input

Users - Format Input JSON și explicații

```

"users": [
  {
    "credentials": {
      "name": "Eduard",
      "password": "secret",
      "accountType": "standard",
      "country": "Romania",
      "balance": "200"
    }
  }
]

```

După cum se poate observa, așa cum am zis și în secțiunea de „**Execuția proiectului**”, în fiecare fișier de input de tip JSON se va afla un JSON Array de utilizatori.

Fiecare utilizator este identificat prin nume, parolă, tipul de cont (standard/premium), țara din care provine (aceasta este țara folosită pentru a vedea care sunt filmele care pot / nu pot fi vizualizate într-o anumită țară) și balanța disponibilă.

Balanța disponibilă pentru un utilizator **nu reprezintă numărul de tokeni** pe care un utilizator îl are. Toți utilizatorii fie ei din JSON Array-ul de început, fie ei înregistrați pe parcursul testelor, încep cu **0 tokeni**. Relația de echivalență între balanță și tokeni este de 1 la 1. Practic, în momentul în care se cumpără tokeni, 1 balance = 1 token. **Orice operațiune de pe platformă se realizează în tokeni.**

Singura diferență în această etapă între un utilizator standard și un utilizator premium este faptul că utilizatorul premium are dreptul să cumpere **15 filme** gratuite înainte ca acesta să plătească 2 tokeni / film la fel ca un utilizator standard. Alte avantaje pentru utilizatorul premium vor apărea în etapa a doua.

Se consideră că utilizatorii ce sunt deja premium în JSON Array-ul din input sau cei ce se înregistrează direct ca utilizatori premium, nu au cumpărat niciun film la începutul fiecărui test. **Practic, toți încep cu 0 filme / seriale cumpărate și 15 filme / seriale gratuite disponibile.**

Movies - Format input JSON și explicații

```

"movies": [
  {
    "name": "John Wick: Chapter 3 - Parabellum",
    "year": "2019",
    "duration": 131,
    "genres": [
      "Action",
      "Thriller",
      "Crime"
    ],
    "actors": [
      "Laurence Fishburne",
      "Halle Berry",
      "Keanu Reeves"
    ],
    "countriesBanned": [
      "Russia",
      "Austria"
    ]
  }
]

```

După cum se poate observa, așa cum se oferă un JSON Array de utilizatori inițiali, așa se oferă și un JSON Array de filme existente pe platformă. Spre deosebire de lista de utilizatori, cea de filme **nu se poate modifica prin nicio operațiune de-a lungul primei etape.**

Orice astfel de film este identificat prin: nume, an de apariție, durată, genuri din care face parte, actori care joacă în film și țările în care filmul este interzis.

Utilizatorii care reprezintă aceste țări nu pot nici măcar vizualiza filmul în lista de filme inițială, deci nu pot realiza absolut nicio acțiune asupra lor. Ele practic sunt inexistente / invizibile pentru utilizatorii din țara / țările banată / e.

Pentru prima etapa, sortarea se va face doar după rating și durată. În cazul în care ambele campuri sunt specificate în input, sortarea se va face mai întâi după durată, iar apoi, în caz de egalitate, după rating.

Definim rating-ul unui film ca fiind media aritmetică a tuturor rating-urilor utilizatorilor care au dat o notă filmului respectiv. Utilizatorii care nu au oferit niciun rating unui anumit film nu sunt luați în considerare. Exemplu: dacă avem 5 utilizatori dintre care doar 3 au oferit rating asupra unui film, rating-ul final al filmului va fi media aritmetică a celor 3 rating-uri oferite.

Rating-ul se va calcula folosind tipul de date Double și va fi exprimat ca un număr real cu **două zecimale exacte**;

De fiecare dată când se primește un nou rating asupra unui film, se recalculează integral. **Nu reciclați rezultatul de până atunci!** Exemplu: dacă luăm situația de mai sus în care avem un rating calculat din 3 note oferite, în momentul în care apare o a 4-a notă, rating-ul se calculează din nou de la capăt. **Nu folosiți faptul că noul rating = (vechiul rating * 3 + nota a 4-a) / 4** deoarece o să apară diferențe la zecimale față de rezultatele de referință oferite de noi.

Actions - Format input JSON și explicații

Așa cum am explicat anterior există două tipuri mari de acțiuni în prima etapă: „**change page**” și „**on page**”. Acestea se află toate într-un JSON Array și se execută secvențial conform ordinii în care sunt în listă. Se pot executa doar dacă pagina pe care vrem să ne mutăm / acțiunea pe care vrem să o facem este disponibilă de pe acea pagină.

Acțiunea de tip „change page”

```
{
  "type": "change page",
  "page": "login"
}
```

Acțiunea de tip „**change page**” conține (cu o singură excepție), indiferent de situație două câmpuri: cel de „**type**” prin care identificăm că este vorba despre o schimbare de pagină și cel de „**page**” prin care anunțăm pagina pe care dorim să facem mutarea.

Acțiunea de tip „change page” - excepție pentru See Details

```
{
  "type": "change page",
  "page": "see details",
  "movie": "movie title"
}
```

Acțiunea de tip „**change page**” pentru schimbarea corectă a paginii pe „see details” conține exact aceleași câmpuri ca toate celelalte acțiuni de tip „**change page**”, dar și unul în plus, „**movie**” pentru a putea determina pentru care dintre filmele afișate pe pagină dorim afișarea detaliilor.

Acțiunile de tip „on page” Așa cum am menționat anterior, acțiunile de tip „**on page**” sunt numeroase și depind ca input de la pagină la pagină. Toate sunt identificate prin câmpul „**type**” ce are întotdeauna valoarea de „**on page**”. În cele ce urmează vom detalia toate tipurile de acțiuni posibile. Vom oferi detalii acolo unde considerăm că JSON-ul nu vorbește de la sine:

Acțiunea de „register”

```
{
  "type": "on page",
  "page": "register",
  "feature": "register",
  "credentials": {
    "name": "Mihnea",
    "password": "acoperisvasile",
    "accountType": "standard",
    "country": "Ireland",
    "balance": "300"
  }
}
```

Acțiunea de „login”

```
{
  "type": "on page",
  "page": "login",
  "feature": "login",
  "credentials": {
    "name": "Ramsi",
    "password": "numiplacevelea"
  }
}
```

Acțiunea de „search”

```
{
  "type": "on page",
  "page": "movies",
  "feature": "search",
  "startsWith": "Iubire din serai"
}
```

Acțiunea de tip search este de fapt tot o filtrare în care condiția este că titlul filmelor trebuie să înceapă obligatoriu cu string-ul oferit în câmpul „**startsWith**”.

După aplicarea unei operațiuni de tip search, din lista filmelor ce exista, vor rămâne disponibile pentru viitoare acțiuni doar acelea care au începutul numelui identic cu string-ul din „**startsWith**”.

Acțiunea de „filter”

```
{
  "type": "on page",
  "page": "movies",
  "feature": "filter",
  "filters": {
    "sort": {
      "rating": "decreasing",
      "duration": "decreasing"
    },
    "contains": {
      "actors": [
        "Keanu Reeves",
        "Actor 2"
      ],
      "genre": [
        "Comedy",
        "Drama"
      ]
    }
  }
}
```

Acțiunea de „buy tokens”

```
{
  "type": "on page",
  "page": "upgrades",
  "feature": "buy tokens",
  "count": "300"
}
```

După cum putem observa, acțiunea de „**buy tokens**” este realizată pe pagina de „**upgrades**”, caracterizată de un feature de „**buy**”, și numele obiectului pe care îl cumpărăm, adică „**tokens**” (moneda platformei). Toate acestea sunt urmate de numărul de tokeni pe care vrem să îl cumpărăm.

Numărul afișat în câmpul de „count” reprezintă atât numărul de tokeni pe care vrem să îl cumpărăm cât și numărul de bani din balance pe care trebuie să îi scădem (altfel spus, $\text{numberTokens} += \text{count}$ și $\text{balance} -= \text{count}$);

Acțiunea de „buy premium account”

```
{
  "type": "on page",
  "page": "upgrades",
  "feature": "buy premium account"
}
```

Analog ca la operațiunea de „buy tokens”. Valoarea unui cont premium este egală cu 10 tokens. Deci, după o operațiune de acest tip, contul curent devine din standard în premium, iar numărul de tokens scade cu 10. Altfel spus: $\text{tokens} -= 10$.

Acțiunea de „Purchase”

```
{
  "type": "on page",
  "page": "see details",
  "feature": "purchase"
}
```

Acțiunea de „Watch”

```
{
  "type": "on page",
  "page": "see details",
  "feature": "watch"
}
```

Acțiunea de „Like”

```
{
  "type": "on page",
  "page": "see details",
  "feature": "like"
}
```

Acțiunea de „Rate a movie”

```
{
  "type": "on page",
  "page": "see details",
  "feature": "rate",
  "rate": 4
}
```

Format output JSON și explicații

În cadrul acestui proiect, există mai multe situații în care oferim output. Outputul se oferă per tip de comandă și are următorul format general:

Format general de „Output”

```
{
  "error": "Error",
  "currentMoviesList": [],
  "currentUser": null
}
```

Cele trei câmpuri din exemplul de mai sus sunt populate în funcție de situație după cum urmează:

- Câmpul **error** poate avea două valori posibile: valoarea **“Error”** în cazul în care o acțiune se finalizează cu eroare sau **null** în cazul în care acțiunea este finalizată cu succes;
- Câmpul **currentMoviesList** reprezintă întotdeauna lista de filme pe care utilizatorul curent le poate vedea la un anumit moment de timp pe ecran (această listă este populată în două situații: când ne aflăm pe pagina de movies (fie ea filtrată sau nu), fie când ne aflăm pe pagina de “see details” a unui film anume, moment în care putem vedea doar filmul respectiv;
- Câmpul **currentUser** conține informații sugestive despre utilizatorul curent: acesta poate să aibă **două valori posibile** (ori **null** în momentul în care nu există niciun utilizator autentificat pe platformă în acest moment **fie cel explicat separat mai jos** în cazul în care există un utilizator autentificat)

Format pentru câmpul de „currentUser”

```
{
  "currentUser": {
    "credentials": {
      "name": "SkF",
      "password": "beesknees",
      "accountType": "standard",
      "country": "Romania",
      "balance": "300"
    },
    "tokensCount": 0, // numărul de tokeni pe care îl are utilizatorul la momentul curent
    "numFreePremiumMovies": 15, // numărul de filme gratuite disponibile în caz de cont premium pe care le mai are utilizatorul la dispoziție
    "purchasedMovies": [], // filmele cumpărate de utilizator
    "watchedMovies": [], // filmele vizualizate de către utilizator
    "likedMovies": [], // filmele apreciate de către utilizator
    "ratedMovies": [] // filmele cărora utilizatorul curent le-a oferit o notă
  }
}
```

În cazul în care avem un utilizator autentificat pe platformă la un anumit moment de timp, câmpul de „currentUser” are formatul prezentat mai sus. Putem observa faptul că se afișează datele la output după cum urmează:

- Se afișează datele de personale ale utilizatorului curent (aceleași cu cele pe care le-a folosit la register de exemplu);
- Se afișează date suplimentare ce întotdeauna încep așa cum sunt văzute în exemplul de mai sus și se populează pe măsură ce testele avansează. Fiecare câmp este comentat mai sus cu ceea ce reprezintă în detaliu.

Fiecare câmp care reprezintă o listă de filme este populat cu detalii despre filme în întregime (exact așa cum au fost primite în input) la care se adaugă câteva câmpuri (numărul de like-uri pe care le are filmul curent, rating-ul filmului curent calculat așa cum a fost precizat anterior, și numărul de rating-uri pe care l-a primit filmul respectiv) după cum urmează:

Format pentru popularea oricărui câmp care reprezintă o listă de filme

```
"currentMoviesList": [
  {
    "name": "John Wick: Chapter 3 - Parabellum",
    "year": "2019",
    "duration": 131,
    "genres": [
      "Action",
      "Thriller",
      "Crime"
    ],
  },
]
```

```

    "actors": [
      "Laurence Fishburne",
      "Halle Berry",
      "Keanu Reeves"
    ],
    "countriesBanned": [
      "Russia"
    ],
    "numLikes": 0,
    "rating": 0,
    "numRatings": 0
  },
  {
    "name": "Titanic",
    "year": "2019",
    "duration": 131,
    "genres": [
      "Action",
      "Thriller",
      "Crime"
    ],
    "actors": [
      "Laurence Fishburne",
      "Halle Berry",
      "Keanu Reeves"
    ],
    "countriesBanned": [
      "Russia"
    ],
    "numLikes": 0,
    "rating": 0,
    "numRatings": 0
  }
]

```

În acest sens există mai jos un test discutat input - output integral pentru a înțelege mai bine cum se populează aceste structuri.

Output-ul este oferit în una dintre următoarele situații:

- Acțiunea este de tip „**change page**” pe orice pagină și s-a produs eroare la executarea acțiunii de „**change page**”;
- Acțiunea este de tip „**change page**” și pagina pe care schimbăm este „**movies**” sau „**see details**”, iar acțiunea s-a produs cu succes;
- Acțiunea este de tip „**on page**” și s-a produs eroare la executarea acțiunii pentru orice fel de acțiune „**on page**”;
- Acțiunea este de tip „**on page**”, iar subacțiunea este de tip „**filter**”, „**search**”, „**login**” sau „**register**”, iar acțiunea s-a realizat cu succes.

Care sunt pașii pentru a genera output-ul în funcție de comandă?

- Ne uităm să vedem dacă acțiunea pe care o realizăm în acest moment **face parte din cele patru categorii precizate mai sus**. Dacă nu face parte înseamnă că nu este o acțiune pentru care să avem un output și am scăpat. Dacă face parte, înseamnă că trebuie să afișăm output și continuăm cu pasul următor;
- Ne gândim dacă acțiunea pe care o avem de făcut în acest moment este **validă sau se finalizează cu eroare**. Dacă se finalizează cu eroare, vom afișa direct output-ul standard precizat mai sus și trecem la următoarea acțiune (acest lucru se poate observa și în exemplul complet explicat mai jos). Dacă nu este eroare, setăm câmpul de eroare pe **null** și mergem la următorul pas;
- Ne gândim dacă după realizarea acțiunii precedente, utilizatorul curent **poate sau nu să vadă unul sau mai multe filme pe ecran în acest moment**. Toate filmele pe care le poate vedea în momentul curent vor fi afișate cu detaliile lor complete (precizate mai sus), în ordinea cerută de la situație la situație (în funcție de filtrele și sortările pe care le avem), **în JSON Array-ul „currentMoviesList”**. Dacă nu putem vedea niciun film pe pagina curentă, atunci acest JSON Array rămâne vid;
- Ne gândim dacă în momentul de față există vreun utilizator înregistrat (altfel spus, ne gândim dacă ne aflăm în zona paginii de „Homepage neautentificat” sau în orice subpagina accesibilă din această zonă sau ne aflăm în zona paginii de „Homepage autentificat” sau în orice subpagina accesibilă din această zonă). Dacă avem un utilizator înregistrat, **îl adăugăm conform formatului explicat mai sus**, dacă nu, acest câmp devine **null**. Putem observa faptul că nu are niciodată sens ca în output să avem vreun film în „currentMoviesList”, dar să nu avem un utilizator conectat în acest moment (deoarece acest lucru nu este posibil conform funcționalităților platformei);
- După ce avem output-ul complet, îl afișăm în fișierul de output corespunzător testului care se rulează în acest moment și trecem la următoarea acțiune.

Exemplu de test input – output integral și explicat

În cele ce urmează o să explicăm un exemplu puțin mai complex, integral de input-output ca să fie destul de ușor de înțeles flow-ul aplicației și cum gândim platforma:


```

{
  "users": [
    {
      "credentials": {
        "name": "Eduard",
        "password": "secret",
        "accountType": "standard",
        "country": "Romania",
        "balance": "200"
      }
    },
    {
      "credentials": {
        "name": "Mihail",
        "password": "discret",
        "accountType": "premium",
        "country": "Russia",
        "balance": "1337"
      }
    }
  ],
  "movies": [
    {
      "name": "John Wick: Chapter 3 - Parabellum",
      "year": "2019",
      "duration": 131,
      "genres": [
        "Action",
        "Thriller",
        "Crime"
      ],
      "actors": [
        "Laurence Fishburne",
        "Halle Berry",
        "Keanu Reeves"
      ],
      "countriesBanned": [
        "Russia"
      ]
    },
    {
      "name": "Titanic",
      "year": "2019",
      "duration": 131,
      "genres": [
        "Action",
        "Thriller",
        "Crime"
      ],
      "actors": [
        "Laurence Fishburne",
        "Halle Berry",
        "Keanu Reeves"
      ],
      "countriesBanned": [
        "Russia"
      ]
    }
  ],
  "actions": [
    {
      "type": "change page",
      "page": "login"
    },
    {
      "type": "on page",
      "feature": "login",
      "credentials": {
        "name": "Eduard",
        "password": "parolamea"
      }
    },
    {
      "type": "change page",
      "page": "register"
    },
    {
      "type": "on page",
      "page": "register",
      "feature": "register",
      "credentials": {
        "name": "Ramsi",
        "password": "numiplacevelea",
        "accountType": "standard",
        "country": "Ireland",
        "balance": "300"
      }
    },
    {
      "type": "change page",
      "page": "login"
    },
    {
      "type": "on page",

```

```

    "page": "login",
    "feature": "login",
    "credentials": {
      "name": "Ramsi",
      "password": "numiplacevelea"
    }
  },
  {
    "type": "change page",
    "page": "movies"
  },
  {
    "type": "on page",
    "page": "movies",
    "feature": "search",
    "startsWith": "Iubire din serai"
  },
  {
    "type": "on page",
    "page": "movies",
    "feature": "filter",
    "filters": {
      "sort": {
        "rating": "decreasing",
        "duration": "decreasing"
      },
      "contains": {
        "actors": [
          "Keanu Reeves"
        ]
      }
    }
  },
  {
    "type": "change page",
    "page": "logout"
  },
  {
    "type": "change page",
    "page": "login"
  },
  {
    "type": "on page",
    "page": "login",
    "feature": "login",
    "credentials": {
      "name": "Eduard",
      "password": "nicigand"
    }
  },
  {
    "type": "change page",
    "page": "login"
  },
  {
    "type": "on page",
    "page": "login",
    "feature": "login",
    "credentials": {
      "name": "Eduard",
      "password": "secret"
    }
  },
  {
    "type": "change page",
    "page": "upgrades"
  },
  {
    "type": "on page",
    "page": "upgrades",
    "feature": "buy tokens",
    "count": "50"
  },
  {
    "type": "on page",
    "page": "upgrades",
    "feature": "buy premium account"
  },
  {
    "type": "change page",
    "page": "movies"
  },
  {
    "type": "change page",
    "page": "see details",
    "movie": "AAA"
  },
  {
    "type": "change page",
    "page": "see details",
    "movie": "BBBB"
  },
  {
    "type": "change page",
    "page": "see details",

```

```

    "movie": "Titanic"
  },
  {
    "type": "on page",
    "page": "see details",
    "feature": "purchase",
    "objectType": "movie",
    "movie": "Titanic"
  },
  {
    "type": "on page",
    "page": "see details",
    "feature": "watch",
    "objectType": "movie",
    "movie": "Titanic"
  },
  {
    "type": "on page",
    "page": "see details",
    "feature": "like",
    "objectType": "movie",
    "movie": "Titanic"
  },
  {
    "type": "change page",
    "page": "movies"
  },
  {
    "type": "change page",
    "page": "see details",
    "movie": "John Wick: Chapter 3 - Parabellum"
  },
  {
    "type": "on page",
    "page": "see details",
    "feature": "like",
    "objectType": "movie",
    "movie": "John Wick: Chapter 3 - Parabellum"
  },
  {
    "type": "on page",
    "page": "see details",
    "feature": "rate",
    "objectType": "movie",
    "movie": "John Wick: Chapter 3 - Parabellum",
    "rate": "5"
  },
  {
    "type": "on page",
    "page": "see details",
    "feature": "watch",
    "objectType": "movie",
    "movie": "John Wick: Chapter 3 - Parabellum"
  },
  {
    "type": "on page",
    "page": "see details",
    "feature": "like",
    "objectType": "movie",
    "movie": "John Wick: Chapter 3 - Parabellum"
  },
  {
    "type": "on page",
    "page": "see details",
    "feature": "rate",
    "objectType": "movie",
    "movie": "John Wick: Chapter 3 - Parabellum",
    "rate": "5"
  }
]
}

```

```

[
  {
    "error": "Error",
    "currentMoviesList": [],
    "currentUser": null
  },
  {
    "error": null,
    "currentMoviesList": [],
    "currentUser": {
      "credentials": {
        "name": "Ramsi",
        "password": "numiplacevelea",
        "accountType": "standard",
        "country": "Ireland",
        "balance": "300"
      },
      "tokensCount": 0,
      "numFreePremiumMovies": 15,
      "purchasedMovies": [],
      "watchedMovies": [],

```

```

    "likedMovies": [],
    "ratedMovies": []
  }
},
{
  "error": "Error",
  "currentMoviesList": [],
  "currentUser": null
},
{
  "error": "Error",
  "currentMoviesList": [],
  "currentUser": null
},
{
  "error": null,
  "currentMoviesList": [
    {
      "name": "John Wick: Chapter 3 - Parabellum",
      "year": "2019",
      "duration": 131,
      "genres": [
        "Action",
        "Thriller",
        "Crime"
      ],
      "actors": [
        "Laurence Fishburne",
        "Halle Berry",
        "Keanu Reeves"
      ],
      "countriesBanned": [
        "Russia"
      ],
      "numLikes": 0,
      "rating": 0,
      "numRatings": 0
    },
    {
      "name": "Titanic",
      "year": "2019",
      "duration": 131,
      "genres": [
        "Action",
        "Thriller",
        "Crime"
      ],
      "actors": [
        "Laurence Fishburne",
        "Halle Berry",
        "Keanu Reeves"
      ],
      "countriesBanned": [
        "Russia"
      ],
      "numLikes": 0,
      "rating": 0,
      "numRatings": 0
    }
  ],
  "currentUser": {
    "credentials": {
      "name": "Ramsi",
      "password": "numiplacevelea",
      "accountType": "standard",
      "country": "Ireland",
      "balance": "300"
    },
    "tokensCount": 0,
    "numFreePremiumMovies": 15,
    "purchasedMovies": [],
    "watchedMovies": [],
    "likedMovies": [],
    "ratedMovies": []
  }
},
{
  "error": null,
  "currentMoviesList": [],
  "currentUser": {
    "credentials": {
      "name": "Ramsi",
      "password": "numiplacevelea",
      "accountType": "standard",
      "country": "Ireland",
      "balance": "300"
    },
    "tokensCount": 0,
    "numFreePremiumMovies": 15,
    "purchasedMovies": [],
    "watchedMovies": [],
    "likedMovies": [],
    "ratedMovies": []
  }
},

```

```

{
  "error": null,
  "currentMoviesList": [
    {
      "name": "John Wick: Chapter 3 - Parabellum",
      "year": "2019",
      "duration": 131,
      "genres": [
        "Action",
        "Thriller",
        "Crime"
      ],
      "actors": [
        "Laurence Fishburne",
        "Halle Berry",
        "Keanu Reeves"
      ],
      "countriesBanned": [
        "Russia"
      ],
      "numLikes": 0,
      "rating": 0,
      "numRatings": 0
    },
    {
      "name": "Titanic",
      "year": "2019",
      "duration": 131,
      "genres": [
        "Action",
        "Thriller",
        "Crime"
      ],
      "actors": [
        "Laurence Fishburne",
        "Halle Berry",
        "Keanu Reeves"
      ],
      "countriesBanned": [
        "Russia"
      ],
      "numLikes": 0,
      "rating": 0,
      "numRatings": 0
    }
  ],
  "currentUser": {
    "credentials": {
      "name": "Ramsi",
      "password": "numiplacevelea",
      "accountType": "standard",
      "country": "Ireland",
      "balance": "300"
    },
    "tokensCount": 0,
    "numFreePremiumMovies": 15,
    "purchasedMovies": [],
    "watchedMovies": [],
    "likedMovies": [],
    "ratedMovies": []
  }
},
{
  "error": "Error",
  "currentMoviesList": [],
  "currentUser": null
},
{
  "error": null,
  "currentMoviesList": [],
  "currentUser": {
    "credentials": {
      "name": "Eduard",
      "password": "secret",
      "accountType": "standard",
      "country": "Romania",
      "balance": "200"
    },
    "tokensCount": 0,
    "numFreePremiumMovies": 15,
    "purchasedMovies": [],
    "watchedMovies": [],
    "likedMovies": [],
    "ratedMovies": []
  }
},
{
  "error": null,
  "currentMoviesList": [
    {
      "name": "John Wick: Chapter 3 - Parabellum",
      "year": "2019",
      "duration": 131,
      "genres": [
        "Action",

```

```

        "Thriller",
        "Crime"
    ],
    "actors": [
        "Laurence Fishburne",
        "Halle Berry",
        "Keanu Reeves"
    ],
    "countriesBanned": [
        "Russia"
    ],
    "numLikes": 0,
    "rating": 0,
    "numRatings": 0
},
{
    "name": "Titanic",
    "year": "2019",
    "duration": 131,
    "genres": [
        "Action",
        "Thriller",
        "Crime"
    ],
    "actors": [
        "Laurence Fishburne",
        "Halle Berry",
        "Keanu Reeves"
    ],
    "countriesBanned": [
        "Russia"
    ],
    "numLikes": 0,
    "rating": 0,
    "numRatings": 0
}
},
"currentUser": {
    "credentials": {
        "name": "Eduard",
        "password": "secret",
        "accountType": "premium",
        "country": "Romania",
        "balance": "150"
    },
    "tokensCount": 40,
    "numFreePremiumMovies": 15,
    "purchasedMovies": [],
    "watchedMovies": [],
    "likedMovies": [],
    "ratedMovies": []
}
},
{
    "error": "Error",
    "currentMoviesList": [],
    "currentUser": null
},
{
    "error": "Error",
    "currentMoviesList": [],
    "currentUser": null
},
{
    "error": null,
    "currentMoviesList": [
        {
            "name": "Titanic",
            "year": "2019",
            "duration": 131,
            "genres": [
                "Action",
                "Thriller",
                "Crime"
            ],
            "actors": [
                "Laurence Fishburne",
                "Halle Berry",
                "Keanu Reeves"
            ],
            "countriesBanned": [
                "Russia"
            ],
            "numLikes": 0,
            "rating": 0,
            "numRatings": 0
        }
    ],
    "currentUser": {
        "credentials": {
            "name": "Eduard",
            "password": "secret",
            "accountType": "premium",
            "country": "Romania",
            "balance": "150"
        }
    }
}

```

```

    },
    "tokensCount": 40,
    "numFreePremiumMovies": 15,
    "purchasedMovies": [],
    "watchedMovies": [],
    "likedMovies": [],
    "ratedMovies": []
  }
},
{
  "error": null,
  "currentMoviesList": [
    {
      "name": "Titanic",
      "year": "2019",
      "duration": 131,
      "genres": [
        "Action",
        "Thriller",
        "Crime"
      ],
      "actors": [
        "Laurence Fishburne",
        "Halle Berry",
        "Keanu Reeves"
      ],
      "countriesBanned": [
        "Russia"
      ],
      "numLikes": 0,
      "rating": 0,
      "numRatings": 0
    }
  ],
  "currentUser": {
    "credentials": {
      "name": "Eduard",
      "password": "secret",
      "accountType": "premium",
      "country": "Romania",
      "balance": "150"
    },
    "tokensCount": 40,
    "numFreePremiumMovies": 14,
    "purchasedMovies": [
      {
        "name": "Titanic",
        "year": "2019",
        "duration": 131,
        "genres": [
          "Action",
          "Thriller",
          "Crime"
        ],
        "actors": [
          "Laurence Fishburne",
          "Halle Berry",
          "Keanu Reeves"
        ],
        "countriesBanned": [
          "Russia"
        ],
        "numLikes": 0,
        "rating": 0,
        "numRatings": 0
      }
    ],
    "watchedMovies": [],
    "likedMovies": [],
    "ratedMovies": []
  }
},
{
  "error": null,
  "currentMoviesList": [
    {
      "name": "Titanic",
      "year": "2019",
      "duration": 131,
      "genres": [
        "Action",
        "Thriller",
        "Crime"
      ],
      "actors": [
        "Laurence Fishburne",
        "Halle Berry",
        "Keanu Reeves"
      ],
      "countriesBanned": [
        "Russia"
      ],
      "numLikes": 0,
      "rating": 0,
      "numRatings": 0
    }
  ]
}

```

```

    }
  ],
  "currentUser": {
    "credentials": {
      "name": "Eduard",
      "password": "secret",
      "accountType": "premium",
      "country": "Romania",
      "balance": "150"
    },
    "tokensCount": 40,
    "numFreePremiumMovies": 14,
    "purchasedMovies": [
      {
        "name": "Titanic",
        "year": "2019",
        "duration": 131,
        "genres": [
          "Action",
          "Thriller",
          "Crime"
        ],
        "actors": [
          "Laurence Fishburne",
          "Halle Berry",
          "Keanu Reeves"
        ],
        "countriesBanned": [
          "Russia"
        ],
        "numLikes": 0,
        "rating": 0,
        "numRatings": 0
      }
    ],
    "watchedMovies": [
      {
        "name": "Titanic",
        "year": "2019",
        "duration": 131,
        "genres": [
          "Action",
          "Thriller",
          "Crime"
        ],
        "actors": [
          "Laurence Fishburne",
          "Halle Berry",
          "Keanu Reeves"
        ],
        "countriesBanned": [
          "Russia"
        ],
        "numLikes": 0,
        "rating": 0,
        "numRatings": 0
      }
    ],
    "likedMovies": [],
    "ratedMovies": []
  }
},
{
  "error": null,
  "currentMoviesList": [
    {
      "name": "Titanic",
      "year": "2019",
      "duration": 131,
      "genres": [
        "Action",
        "Thriller",
        "Crime"
      ],
      "actors": [
        "Laurence Fishburne",
        "Halle Berry",
        "Keanu Reeves"
      ],
      "countriesBanned": [
        "Russia"
      ],
      "numLikes": 1,
      "rating": 0,
      "numRatings": 0
    }
  ],
  "currentUser": {
    "credentials": {
      "name": "Eduard",
      "password": "secret",
      "accountType": "premium",
      "country": "Romania",
      "balance": "150"
    },
  },

```



```

    "tokensCount": 40,
    "numFreePremiumMovies": 14,
    "purchasedMovies": [
      {
        "name": "Titanic",
        "year": "2019",
        "duration": 131,
        "genres": [
          "Action",
          "Thriller",
          "Crime"
        ],
        "actors": [
          "Laurence Fishburne",
          "Halle Berry",
          "Keanu Reeves"
        ],
        "countriesBanned": [
          "Russia"
        ],
        "numLikes": 1,
        "rating": 0,
        "numRatings": 0
      }
    ],
    "watchedMovies": [
      {
        "name": "Titanic",
        "year": "2019",
        "duration": 131,
        "genres": [
          "Action",
          "Thriller",
          "Crime"
        ],
        "actors": [
          "Laurence Fishburne",
          "Halle Berry",
          "Keanu Reeves"
        ],
        "countriesBanned": [
          "Russia"
        ],
        "numLikes": 1,
        "rating": 0,
        "numRatings": 0
      }
    ],
    "likedMovies": [
      {
        "name": "Titanic",
        "year": "2019",
        "duration": 131,
        "genres": [
          "Action",
          "Thriller",
          "Crime"
        ],
        "actors": [
          "Laurence Fishburne",
          "Halle Berry",
          "Keanu Reeves"
        ],
        "countriesBanned": [
          "Russia"
        ],
        "numLikes": 1,
        "rating": 0,
        "numRatings": 0
      }
    ],
    "ratedMovies": []
  },
  {
    "error": null,
    "currentMoviesList": [
      {
        "name": "John Wick: Chapter 3 - Parabellum",
        "year": "2019",
        "duration": 131,
        "genres": [
          "Action",
          "Thriller",
          "Crime"
        ],
        "actors": [
          "Laurence Fishburne",
          "Halle Berry",
          "Keanu Reeves"
        ],
        "countriesBanned": [
          "Russia"
        ],
        "numLikes": 0,

```

```
"rating": 0,
"numRatings": 0
},
{
  "name": "Titanic",
  "year": "2019",
  "duration": 131,
  "genres": [
    "Action",
    "Thriller",
    "Crime"
  ],
  "actors": [
    "Laurence Fishburne",
    "Halle Berry",
    "Keanu Reeves"
  ],
  "countriesBanned": [
    "Russia"
  ],
  "numLikes": 1,
  "rating": 0,
  "numRatings": 0
}
],
"currentUser": {
  "credentials": {
    "name": "Eduard",
    "password": "secret",
    "accountType": "premium",
    "country": "Romania",
    "balance": "150"
  },
  "tokensCount": 40,
  "numFreePremiumMovies": 14,
  "purchasedMovies": [
    {
      "name": "Titanic",
      "year": "2019",
      "duration": 131,
      "genres": [
        "Action",
        "Thriller",
        "Crime"
      ],
      "actors": [
        "Laurence Fishburne",
        "Halle Berry",
        "Keanu Reeves"
      ],
      "countriesBanned": [
        "Russia"
      ],
      "numLikes": 1,
      "rating": 0,
      "numRatings": 0
    }
  ],
  "watchedMovies": [
    {
      "name": "Titanic",
      "year": "2019",
      "duration": 131,
      "genres": [
        "Action",
        "Thriller",
        "Crime"
      ],
      "actors": [
        "Laurence Fishburne",
        "Halle Berry",
        "Keanu Reeves"
      ],
      "countriesBanned": [
        "Russia"
      ],
      "numLikes": 1,
      "rating": 0,
      "numRatings": 0
    }
  ],
  "likedMovies": [
    {
      "name": "Titanic",
      "year": "2019",
      "duration": 131,
      "genres": [
        "Action",
        "Thriller",
        "Crime"
      ],
      "actors": [
        "Laurence Fishburne",
        "Halle Berry",
        "Keanu Reeves"
      ],
```

```

    },
    "countriesBanned": [
      "Russia"
    ],
    "numLikes": 1,
    "rating": 0,
    "numRatings": 0
  }
},
"ratedMovies": []
},
{
  "error": null,
  "currentMoviesList": [
    {
      "name": "John Wick: Chapter 3 - Parabellum",
      "year": "2019",
      "duration": 131,
      "genres": [
        "Action",
        "Thriller",
        "Crime"
      ],
      "actors": [
        "Laurence Fishburne",
        "Halle Berry",
        "Keanu Reeves"
      ],
      "countriesBanned": [
        "Russia"
      ],
      "numLikes": 0,
      "rating": 0,
      "numRatings": 0
    }
  ],
  "currentUser": {
    "credentials": {
      "name": "Eduard",
      "password": "secret",
      "accountType": "premium",
      "country": "Romania",
      "balance": "150"
    },
    "tokensCount": 40,
    "numFreePremiumMovies": 14,
    "purchasedMovies": [
      {
        "name": "Titanic",
        "year": "2019",
        "duration": 131,
        "genres": [
          "Action",
          "Thriller",
          "Crime"
        ],
        "actors": [
          "Laurence Fishburne",
          "Halle Berry",
          "Keanu Reeves"
        ],
        "countriesBanned": [
          "Russia"
        ],
        "numLikes": 1,
        "rating": 0,
        "numRatings": 0
      }
    ],
    "watchedMovies": [
      {
        "name": "Titanic",
        "year": "2019",
        "duration": 131,
        "genres": [
          "Action",
          "Thriller",
          "Crime"
        ],
        "actors": [
          "Laurence Fishburne",
          "Halle Berry",
          "Keanu Reeves"
        ],
        "countriesBanned": [
          "Russia"
        ],
        "numLikes": 1,
        "rating": 0,
        "numRatings": 0
      }
    ],
    "likedMovies": [
      {

```

```

    "name": "Titanic",
    "year": "2019",
    "duration": 131,
    "genres": [
      "Action",
      "Thriller",
      "Crime"
    ],
    "actors": [
      "Laurence Fishburne",
      "Halle Berry",
      "Keanu Reeves"
    ],
    "countriesBanned": [
      "Russia"
    ],
    "numLikes": 1,
    "rating": 0,
    "numRatings": 0
  }
},
"ratedMovies": []
},
{
  "error": "Error",
  "currentMoviesList": [],
  "currentUser": null
},
{
  "error": "Error",
  "currentMoviesList": [],
  "currentUser": null
},
{
  "error": "Error",
  "currentMoviesList": [],
  "currentUser": null
},
{
  "error": "Error",
  "currentMoviesList": [],
  "currentUser": null
}
]

```

Explicații privind testul de input-output de deasupra: Cum gândim inputul?

Pentru început **stocăm toți utilizatorii și filmele deja existente pe platformă** din input, după care executăm acțiunile rând pe rând, secvențial, așa cum am precizat mai sus:

- Întâlnim prima dată o acțiune de tip **"change page"** pe **"login"**. Cum noi de abia am început testul, suntem pe pagina de "Homepage neautentificat" deci putem executa operațiunea fără nicio problemă;
- Executăm **acțiunea de login** cu informațiile din input adică: **name = Eduard și password = parolamea**. Mergem și verificăm dacă avem așa ceva stocat în baza de date până acum și observăm că **NU**. Din acest motiv, vom **afișa output-ul de eroare corespundent** dintre cele explicate mai sus;
- **Schimbăm pagina pe „register”** (totul este în regulă deoarece așa cum am precizat mai sus, după o acțiune eșuată de login/register se afișează output-ul corespunzător și ne întoarcem pe "Homepage neautentificat" automat);
- **Facem acțiunea de "register"** (verificăm mai întâi să nu mai existe un cont cu același username și nu mai există) cu succes și afișăm output-ul corespunzător acestei acțiuni dintre cele explicate mai sus;
- Încercăm să ne mutăm pe **pagina de login** (nu se poate deoarece după acțiunea de login finalizată cu succes de mai sus suntem mutați automat pe pagina de "Homepage autentificat") → afișăm eroarea corespunzătoare;
- Analog pentru încercarea de a realiza **acțiunea de "login"**;
- **Schimbăm pe pagina de "movies"** (acțiune permisă) → afișăm output-ul corespunzător cu toate filmele pe care utilizatorul curent le vede în momentul de față (pe toate în afară de cele banate în țara sa). Utilizatorul curent este din Irlanda, deci toate filmele sunt afișate (acestea evident au 0 like-uri și 0 rating pentru că nu a fost executată nicio acțiune asupra lor). Apoi, sunt afișate informațiile corespunzătoare contului său adică: nu are niciun token (deoarece încă nu a cumpărat niciunul), are 15 filme disponibile premium (chiar dacă este utilizator standard, în cazul în care va deveni premium va avea aceste 15 filme gratuite. Până atunci, ele doar există, însă nu le poate folosi), nu a cumpărat/văzut/apreciat/ofert nicio notă până acum;
- **Executăm o acțiune de tip "search"** cu string-ul de început "Iubire din serai" → acțiune perfect permisă, doar că nu există niciun film al cărui titlu să înceapă cu acest string, deci output-ul o să arate identic cu cel de dinainte însă, **"currentMoviesList" va fi gol** deoarece nu există niciun film care să satisfacă aceste condiții;

- **Executăm o acțiunea de tip "filter"**. Pentru început renunțăm la toate filtrările care s-au făcut până acum, deci avem din nou toate filmele (mai puțin cele interzise în țara utilizatorului curent) disponibile. Dintre acestea, conform filtrării, le păstrăm doar pe acelea în care joacă actorul **"Keanu Reeves"** (toate), iar cele rămase sunt sortate în ordine descrescătoare după rating (ambele au rating 0 deoarece nu au primit nicio notă până acum de la niciun utilizator), iar în caz de egalitate, sunt sortate în ordine descrescătoare după durată (ambele au aceeași durată). În caz de egalitate până la final, sunt afișate în ordinea în care au fost oferite în input. Restul detaliilor despre utilizatorul curent sunt identice;
- Se merge pe pagina de logout → acțiune permisă;
- Se merge pe pagina de login → acțiune permisă;
- **Se încearcă login cu datele respective însă sunt greșite** → mesaj corespunzător și ne întoarcem pe pagina de "Homepage neautentificat";
- Se merge din nou pe pagina de login → acțiune permisă;
- Se face **login cu datele corecte** de această dată → afișăm datele de despre utilizatorul curent în output;
- **Se merge pe pagina de upgrades** → acțiune permisă;
- Se cumpără 50 de tokeni de pe pagina de upgrades (deci balance -= 50 → noua balanță este de 150 și tokens += 50 și cum inițial avea 0, acum, utilizatorul curent are 50 de tokens);
- Se realizează **acțiunea de buy premium account** (care valorează 10 tokens și deci din acest moment utilizatorul curent devine premium, iar numărul său de tokens scade cu 10 → deci de acum mai are doar 40 de tokens);
- **Se merge pe pagina de movies** → acțiune permisă deci se afișează toate filmele disponibile (toate pe același principiu explicat la situația anterioară) și detaliile despre utilizatorul curent (tokens count 40 așa cum am calculat mai sus și balance rămas 150);
- Se încearcă **schimbarea paginii de două ori pe filmele "AAA" și "BBBB"** care nu există așa că avem de afișat două erori în acest sens;
- Se încearcă **schimbarea paginii către filmul "Titanic"** → acțiune permisă și deci rămân de afișat doar detaliile filmului respectiv;
- Se cumpără filmul respectiv deci se afișează exact același output de mai sus cu mențiunea că filmul "Titanic" cu toate detaliile sale apare acum și în zona de **purchasedMovies**. De asemenea, având în vedere că utilizatorul curent este premium și nu standard, pierde unul dintre filmele gratuite disponibile (rămân 14) și nu folosește tokens pentru cumpărarea acestui film;
- Se vizionează filmul respectiv → același output ca mai sus doar că de această dată **apare filmul văzut și în zona de „watchedMovies"**;
- **Se oferă like la filmul Titanic** → se oferă același output ca mai sus doar că acum filmul apare și în zona de „likedMovies", iar peste tot pe unde apare, numărul de like-uri ale sale este incrementat cu 1 (cum nu avea niciun like până atunci, de acum o să aibă unul singur);
- **Se merge pe pagina de movies** → același output ca mai sus doar că de această dată ne apar din nou **toate filmele disponibile în zona de currentMoviesList**;
- **Se merge pe pagina filmului: "John Wick: Chapter 3 - Parabellum"** → afișăm output-ul corespunzător (adică, în acest moment avem acces doar la filmul precizat anterior, însă nu sunt afectate cu nimic filmele deja cumpărate/vizualizate/apreciate de utilizatorul curent);
- **Se încearcă acțiunea de like** → nu se poate pentru că filmul nu a fost mai întâi vizualizat;
- **Se încearcă acțiunea de rate** → nu se poate pentru că filmul nu a fost mai întâi vizualizat;
- **Se încearcă acțiunea de vizualizare** → nu se poate pentru că filmul nu a fost mai întâi cumpărat;
- **Se încearcă din nou acțiunea de like** → nu se poate pentru că filmul nu a fost mai întâi vizualizat;
- **Se încearcă din nou acțiunea de rate** → nu se poate pentru că filmul nu a fost mai întâi vizualizat;

În acest fel, am parcurs împreună un test input-output ca să putem observa cum gândim și afișăm corespunzător mesajele cerute!

Indicații

- Separați conceptele de sine stătătoare în clase separate, nu le îmbinați - clasele ar trebui să aibă un sigur rol.
- Adaptați agregarea și moștenirea la situație, grupați pe cât posibil informația și acțiunile comune în clase generale.
- Nu vă apucați să scrieți direct; alocați timp modelării și abstractizării, pentru că altfel vă puteți trezi cu o temă muncitorească, cu mult cod din care să nu înțelegeți prea multe și pe care să îl extindeți greu.
- Acesta este prima etapă a proiectului, ceea ce presupune că va exista și o a doua etapă; vă recomandăm să încercați să scrieți un cod cât mai generic, care să permită adăugarea ulterioară de noi funcționalități; cu toate acestea, etapa a doua se poate trimite fără să fi trimis prima etapă a proiectului, însă va fi nevoie ca în cadrul celei de-a doua etape să se implementeze și funcționalitățile primei etape pentru a putea primi punctajul total pe testele celei de-a doua părți.

Testarea soluției

Pentru testarea soluției, rulați funcția **main** a clasei **Test**. Aceasta va rula atât testele, cât și checkstyle-ul. Pentru rularea checkerului, aveți nevoie ca proiectul vostru să aibă încărcate bibliotecile pentru citirea fișierelor JSON. Mai multe detalii [aici](#).

Evaluare

Punctajul constă din:

- 60p implementare - trecerea testelor
- 10p coding style (vezi checkstyle)
- 15p design și organizare (folosire design patterns)
- 10p README clar, concis, explicații axate pe design (flow, interacțiuni)
- 5p utilizare Git (minimum 3 commit-uri personale, relevante pentru flow-ul proiectului)
- Folosirea git pentru versionare va fi verificată din folderul .git pe care trebuie să îl includeți în arhiva temei.
- Punctajul se va acorda dacă ați făcut minim 3 commit-uri relevante și cu mesaj sugestiv.
- **NU** este permis să aveți repository-urile de git publice până la deadline-ul hard.
- Pentru aceasta etapa este suficient să folosiți un singur design pattern, data fiind organizarea orelor de laborator.
- Totuși, țineți cont de faptul că la finalul etapei a2a va trebui să aveți 4 design patterns, deci aveți grijă cum vă structurați codul.

Pe pagina [Indicații pentru teme](#) găsiți indicații despre scrierea README-ului și depunctările generale pentru teme

Depunctările pentru **designul și organizarea codului** se vor scădea din punctajul testelor. Dacă vor apărea depunctări specifice temei în momentul evaluării, nemenționate pe pagina cu depunctări generale, ele se vor încadra în limitele de maxim 15 pentru design, 10p pentru README. Dacă tema nu respectă cerințele, sau nu are design OOP, atunci pot apărea depunctări suplimentare.

Bonusuri: La evaluare, putem oferi bonusuri pentru design foarte bun, cod bine documentat, dar și pentru diverse elemente suplimentare alese de voi.

Temele vor fi testate împotriva plagiatului. Orice tentativă de copiere va duce la **anularea punctajului** de pe parcursul semestrului și **repetarea materiei** atât pentru sursă(e) cât și pentru destinație(ii), fără excepție.

Checkstyle

Unul din obiectivele temei este învățarea respectării code-style-ului limbajului pe care îl folosiți. Aceste convenții (de exemplu cum numiți fișierele, clasele, variabilele, cum indentați) sunt verificate pentru temă de către tool-ul checkstyle [<https://checkstyle.sourceforge.io/>].

Pe pagina de [Recomandări cod](#) găsiți câteva exemple de coding style.

Dacă numărul de erori depistate de checkstyle depășește 30, atunci punctele pentru coding-style nu vor fi acordate. Dacă punctajul este negativ, *acesta se trunchiază la 0*.

Exemple:

- punctaj_total = 125 și nr_erori = 200 ⇒ nota_finala = 115
- punctaj_total = 125 și nr_erori = 29 ⇒ nota_finala = 125
- punctaj_total = 80 și nr_erori = 30 ⇒ nota_finala = 80
- punctaj_total = 80 și nr_erori = 31 ⇒ nota_finala = 70

Upload temă

Arhiva pe care o veți urca pe VMChecker [<https://vmchecker.cs.pub.ro/ui/#POO>] va trebui să conțină în directorul rădăcină:

- fișierul README
- folder-ul src cu pachetele și cu fișierele .java

Resurse și linkuri utile

- Schelet de cod [<https://github.com/oop-pub/oop-assignments/tree/master/proiect1>]
- [Indicații pentru teme](#)
- [Recomandări coding style & javadoc](#)