



# Documentație Proiect Site de Filme

Muresan Davide si Munteanu George

Data predarii: 09.01.2025

# 1 Prezentare Generala a Proiectului

Acest proiect reprezinta o aplicatie web moderna construita cu ajutorul React, avand ca obiectiv principal oferirea unei experiente placute utilizatorilor care doresc sa isi gestioneze si sa exploreze filmele preferate.

Aplicatia dispune de o interfata intuitiva, optimizata pentru toate tipurile de dispozitive, astfel incat sa poata fi accesata cu usurinta de oriunde. Prin arhitectura bazata pe componente, fiecare functionalitate este modulara si usor de intretinut, permitand in acelasi timp o integrare rapida de noi caracteristici.

## 2 Functionalitati Principale

- **Navigare in catalogul de filme:** utilizatorii pot explora o gama larga de titluri, cu detalii si imagini reprezentative pentru fiecare film.
- **Salvarea filmelor favorite:** orice film poate fi adaugat intr-o lista speciala, pentru acces rapid si organizat.
- **Navigare responsiva:** interfata se adapteaza in mod fluent pentru desktop, tableta si mobil.
- **Rutare dinamica intre pagini:** sistemul de rute React Router asigura acces rapid si clar la diversele sectiuni ale site-ului.

## 3 Stack Tehnologic

- **React.js:** biblioteca JavaScript moderna ce asigura o interfata reactiva si modulara.
- **React Router:** gestionarea coerenta a rutelor si a navigatiei in aplicatie.
- **CSS:** pentru stilizarea personalizata si responsiva a interfetei.

## 4 Design Patterns

### 4.1 Arhitectura Bazata pe Componente

Aplicatia urmeaza principiile React, fragmentand interfata in componente reutilizabile, clar definite:

- **NavBar:** Sectiune de navigare globala, prezenta pe toate paginile.
- **Pagina Principala (Home):** Listeaza filmele disponibile si informatii esentiale despre acestea.
- **Pagina Favorite (Favorites):** Permite vizualizarea si gestionarea titlurilor salvate de utilizator.

## 4.2 Pattern-ul Singleton

Singleton-ul este folosit pentru a ne asigura ca exista o singura instanta a unei clase/obiect, reprezentand astfel un „punct unic de adevar” (single source of truth).

In React, acest concept este adesea sustinut de Context API, permitand accesul la starea globala din orice componenta, fara a fi necesara transmiterea datelor prin props. In cazul conectarii la un serviciu extern (precum The Movie Database), putem folosi Singleton pentru gestionarea cererilor, astfel incat acelasi client de API sa fie utilizat pe tot parcursul aplicatiei.

## 4.3 Pattern-ul Builder

In acest proiect, „Builder”-ul este ilustrat prin modul in care cream „cardurile” de filme. Intr-un mod figurativ, fiecare card de film este „construit” pas cu pas:

- Adaugam imaginea reprezentativa
- Includem titlul si o scurta descriere
- Inseram butonul de adaugare la favorite

Astfel, folosim aceeasi structura generala pentru a crea reprezentari diferite (carduri) pentru fiecare film in parte.

## 4.4 Pattern-ul Router

Navigarea in aplicatie este gestionata prin **React Router**, un exemplu clasic de implementare a pattern-ului „Router”. Codul esential arata astfel:

```
1 <Routes>
2   <Route path="/" element={<Home />} />
3   <Route path="/favorites" element={<Favorites />} />
4 </Routes>
```

Acest sistem asigura rute clare si management eficient al istoricului de navigare, mentinand totodata URL-uri semnificative.

# 5 Arhitectura Aplicatiei

## 5.1 Diagrama de Secventa

Pentru a ilustra fluxul de comunicare si dependentele dintre componente, putem apela la o diagrama de secventa (Figura 1):

## 5.2 Diagrama de Activitate

Această Diagramă de Activitate arată:

- 1) Fluxurile principale ale utilizatorului prin aplicație
  - 2) Punctele de decizie pentru interacțiunile utilizatorului
  - 3) Activități paralele posibile prin navigare
  - 4) Funcționalitățile de bază precum vizualizarea filmelor și gestionarea favoritelor
  - 5) Puncte clare de start și final
- (Figura este mai jos 2),

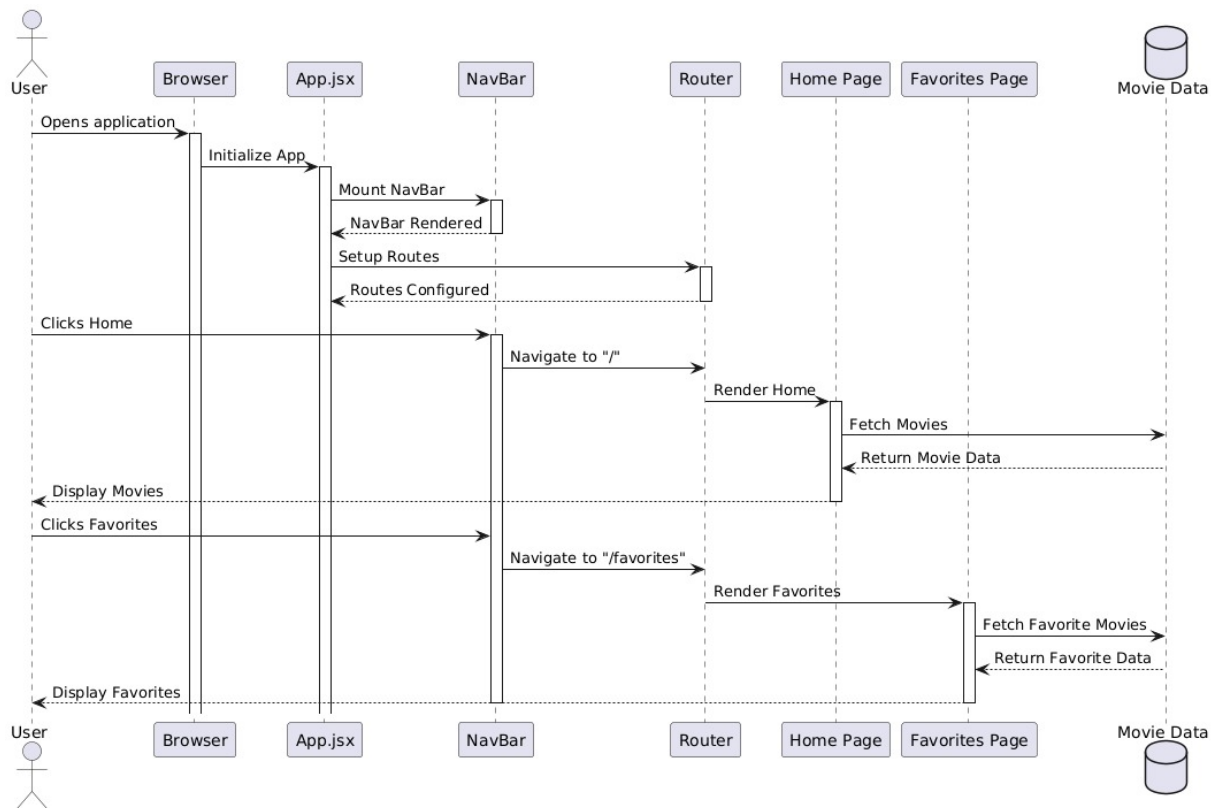


Figura 1: Diagrama de secventa Muresan Davide

## 6 Detalii de Implementare

### 6.1 Implementarea Rutarii

Rutarea se face prin React Router, definind traseele esentiale:

- **Ruta principala ("/")**: Afiseaza catalogul principal de filme.
- **Ruta favorite ("/favorites")**: Afiseaza lista de filme salvate de utilizator.

### 6.2 Gestionarea Starii

Aplicatia utilizeaza state management-ul integrat din React pentru a sincroniza lista de filme si starea lor (favorite sau nu) intre componente. Astfel, la fiecare actiune a utilizatorului (de ex. adaugarea unui film la favorite), starea se actualizeaza si interfata reactioneaza instant.

## 7 Dezvoltari Viitoare

- **Recomandari de filme**: in baza istoricului si preferintelor utilizatorilor.
- **Functionalitati de partajare sociala**: posibilitatea de a distribui listele de favorite catre prieteni.

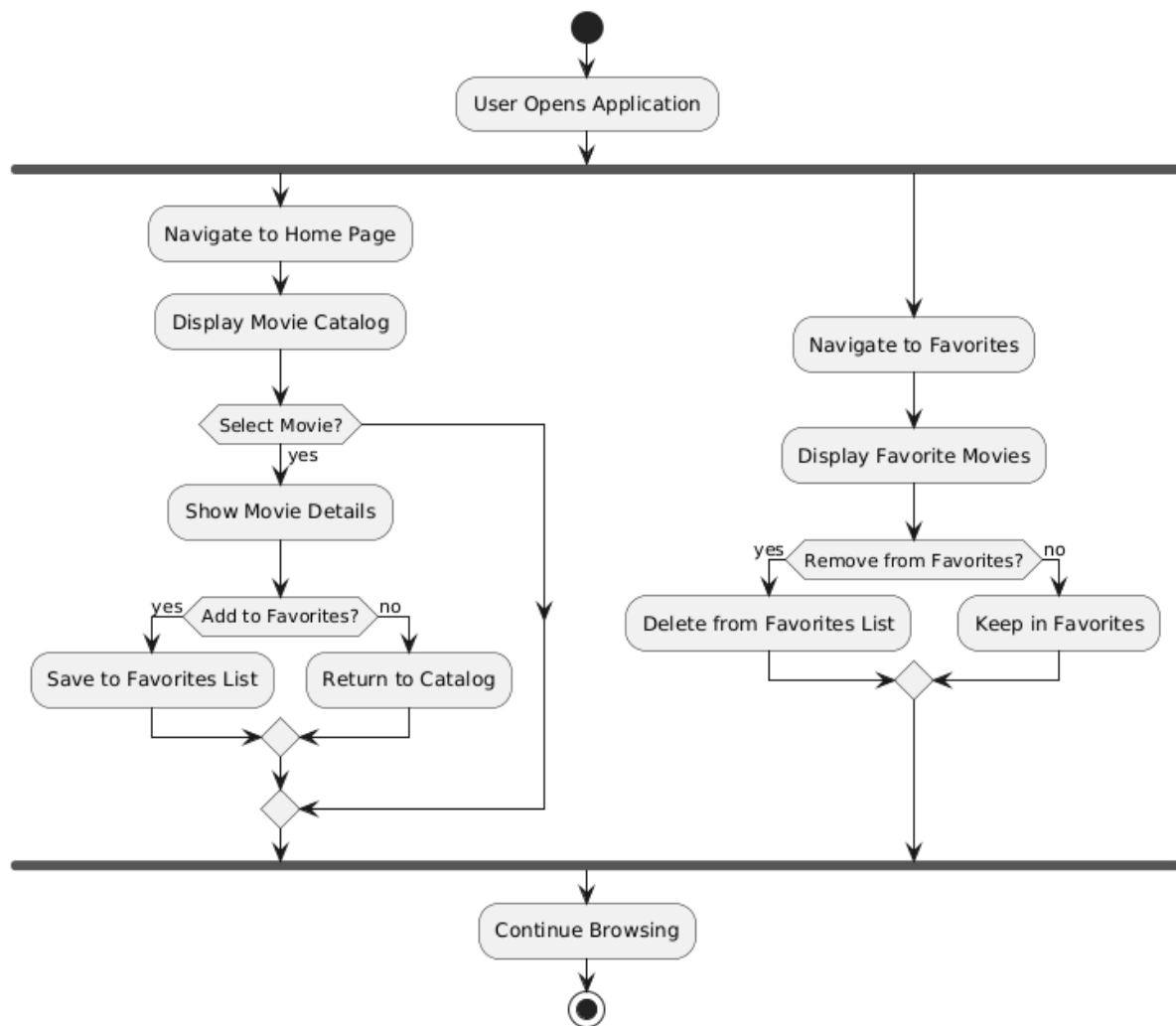


Figura 2: Diagrama de activitate Munteanu George

## 8 Concluzie

Prin acest proiect, demonstrem ca o structura coerenta si un design bine gandit care aduce beneficii atat in dezvoltare, cat si in intretinere. Arhitectura bazata pe componente, impreuna cu design patterns precum Singleton, Builder si Router, asigura flexibilitate pe termen lung astfel incat orice echipa de dezvoltare poate prelua si extinde aplicatia noastra cu noi functionalitati, mentinand in acelasi timp un cod clar si bine organizat.

### 8.1 Bibliografie

- **The Movie Database:** <https://www.themoviedb.org/>
- **React Documentation:** <https://react.dev/>
- **React Router Documentation:** <https://reactrouter.com/>