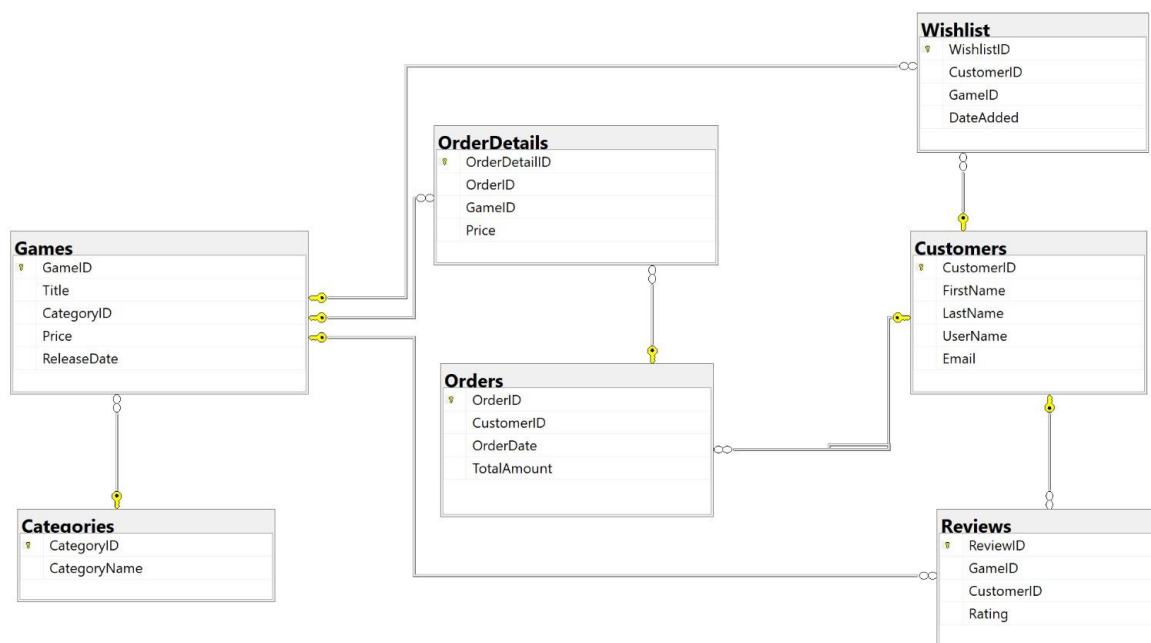


# Documentația Proiectului

## 1. Descrierea bazei de date

Proiectul „GameStore” gestionează o bază de date despre jocuri, clienți, comenzi, recenzii și alte aspecte esențiale pentru funcționarea unui magazin de jocuri video, oferind funcționalități de analiză a vânzărilor și interacțiunilor utilizatorilor.

### 1.1 Diagrama bazei de date



### 1.2 Structura tabelelor

Tabelul **Games** conține detalii despre jocurile video disponibile în magazin. Fiecare joc are un ID unic, un titlu, o categorie asociată, un preț, și o dată de lansare. Relația cu tabelul **Categories** este de tip many-to-one. Tabelul **Categories** conține informații despre categoriile jocurilor video, fiecare categorie având un ID unic și un nume specific. Este folosit pentru a organiza jocurile pe categorii distincte. Tabelul **Customers** gestionează datele utilizatorilor. Fiecare utilizator are un ID unic, nume, un nume de utilizator unic, și o adresă de email unică. Constrângerile asigură validitatea numelor și unicității datelor. Tabelul **Reviews** stochează recenzii oferite de clienți pentru jocuri. Fiecare recenzie este identificată printr-un ID unic și include un ID al jocului, un ID al clientului, și o notă de la 1 la 5, conform unei constrângeri de validitate. Tabelul **Wishlist** reține jocurile adăugate de clienți în lista de dorințe. Fiecare înregistrare are un ID unic, ID-ul clientului, ID-ul jocului și data adăugării. Relațiile permit corelarea cu clienții și jocurile. Tabelul **Orders** conține informații despre comenzile plasate de clienți. Fiecare comandă are un ID unic, ID-ul clientului care a

plasat-o, data comenzii și valoarea totală. Tabelul **OrderDetails** oferă detalii despre jocurile incluse în fiecare comandă. Fiecare intrare are un ID unic, ID-ul comenzii, ID-ul jocului și prețul jocului în cadrul acelei comenzii. Relațiile cu tabelul Orders și Games asigură consistența datelor.

### 1.3 Descrierea constrângerilor de integritate

Constrângerile de integritate în baza de date a magazinului de jocuri video asigură consistența și fiabilitatea datelor:

- Fiecare tabel include chei primare (CategoryID, GameID, CustomerID, etc.) pentru identificarea unică a fiecărui rând.
- Relațiile dintre tabele sunt gestionate prin chei străine (ex. CategoryID din Games referă la Categories, iar CustomerID din Orders la Customers), garantând existența entităților referite.
- Câmpurile unice, precum Username și Email în tabelul Customers, previn duplicatele care ar compromite integritatea datelor utilizatorilor.
- Constrângeri CHECK validează corectitudinea datelor, cum ar fi:
  - Rating din Reviews este limitat la valori între 1 și 5.
  - Câmpurile FirstName și LastName permite doar caractere alfabetice.
  - Câmpul Username permite doar caractere alfanumerice.
- Constrângerile UC\_CustomerGame din OrderDetails asigură unicitatea perechii OrderID și GameID, prevenind duplicarea jocurilor într-o comandă.

În plus, triggerul **trg\_RemoveFromWishlist** automatizează gestionarea listei de dorințe. Acesta elimină jocurile achiziționate de un client din wishlist-ul său, asigurând consistența între comenzile efectuate și dorințele exprimate. Acest mecanism reduce riscul de redundanță și optimizează gestionarea datelor.

### 1.4 Descrierea procedurilor și funcțiilor

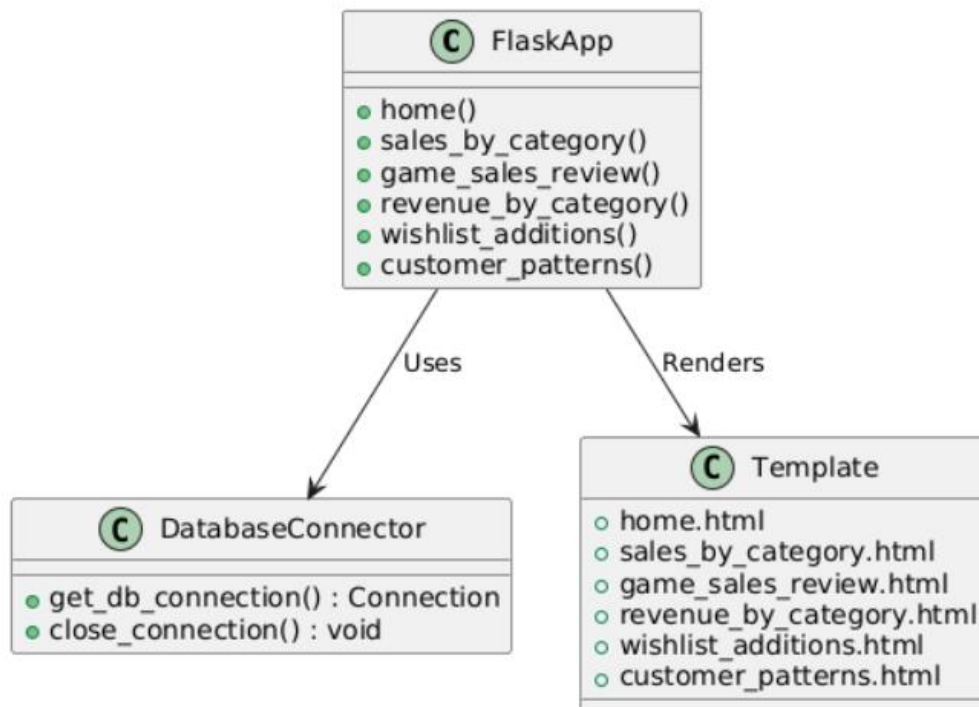
În baza de date a magazinului de jocuri video, procedurile stocate oferă posibilitatea de a extrage și analiza datele:

- **GetSalesByCategory** calculează numărul total de jocuri vândute pentru fiecare categorie, ordonând rezultatele descrescător după volum. Aceasta ajută la analiza performanței categoriilor de jocuri.
- **GetGameSalesAndReviewAnalysis** combină informații despre vânzări și recenzii pentru a analiza popularitatea și succesul financiar al fiecărui joc. Procedura returnează veniturile totale, numărul de vânzări și media recenziilor.

- **GetRevenueByCategoryLast6Months** oferă date despre veniturile generate în ultimele șase luni, grupate pe categorii și zile, fiind utilă pentru identificarea tendințelor sezoniere.
- **GetWishlistAdditionsAndGameDetails** analizează adăugările în wishlist în raport cu prețul și evaluările jocurilor, furnizând informații despre popularitatea produselor.
- **GetCustomerPurchaseAndReviewPatterns** oferă statistici despre clienți, incluzând obiceiurile lor de cumpărare, recenziile oferite și valoarea medie a comenzilor, sprijinind strategii de fidelizare.

## 2. Descrierea aplicației

### 2.1 Diagrama de clase



### 2.2 Structura claselor

Aplicația este organizată în următoarele componente: clasa principală, **FlaskApp**, instanța principală Flask, **get\_db\_connection**, metoda pentru conectarea la baza de date și o metoda pentru fiecare pagină web ce corespunde fiecărei proceduri:

**home:** Răspunde la ruta / și încarcă pagina principală.

**sales\_by\_category:** Afișează vânzările pe categorii și permite filtrarea.

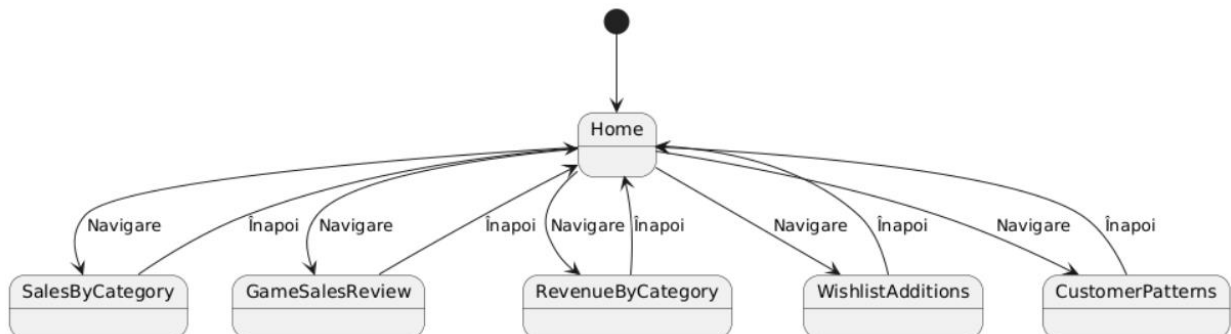
**game\_sales\_review:** Analizează vânzările și recenziile jocurilor.

**revenue\_by\_category:** Generează vizualizări pentru veniturile pe categorii.

**wishlist\_additions:** Afișează popularitatea jocurilor în wishlist-uri.

**customer\_patterns:** Prezintă comportamentul clienților pe baza comenzilor și recenziilor.

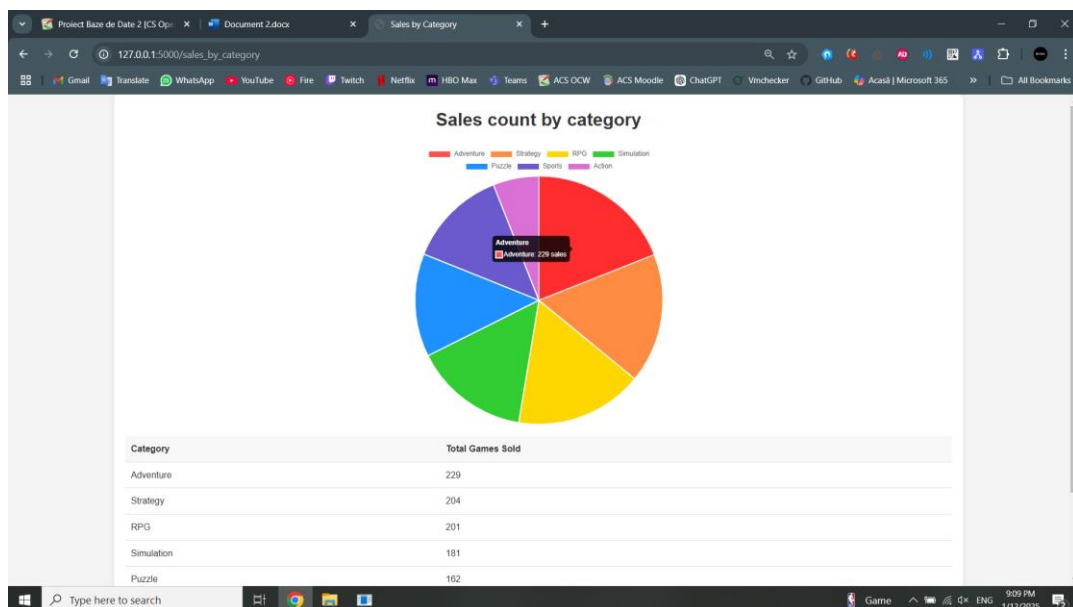
## 2.3 Diagrama de stări și workflow-ul aplicației

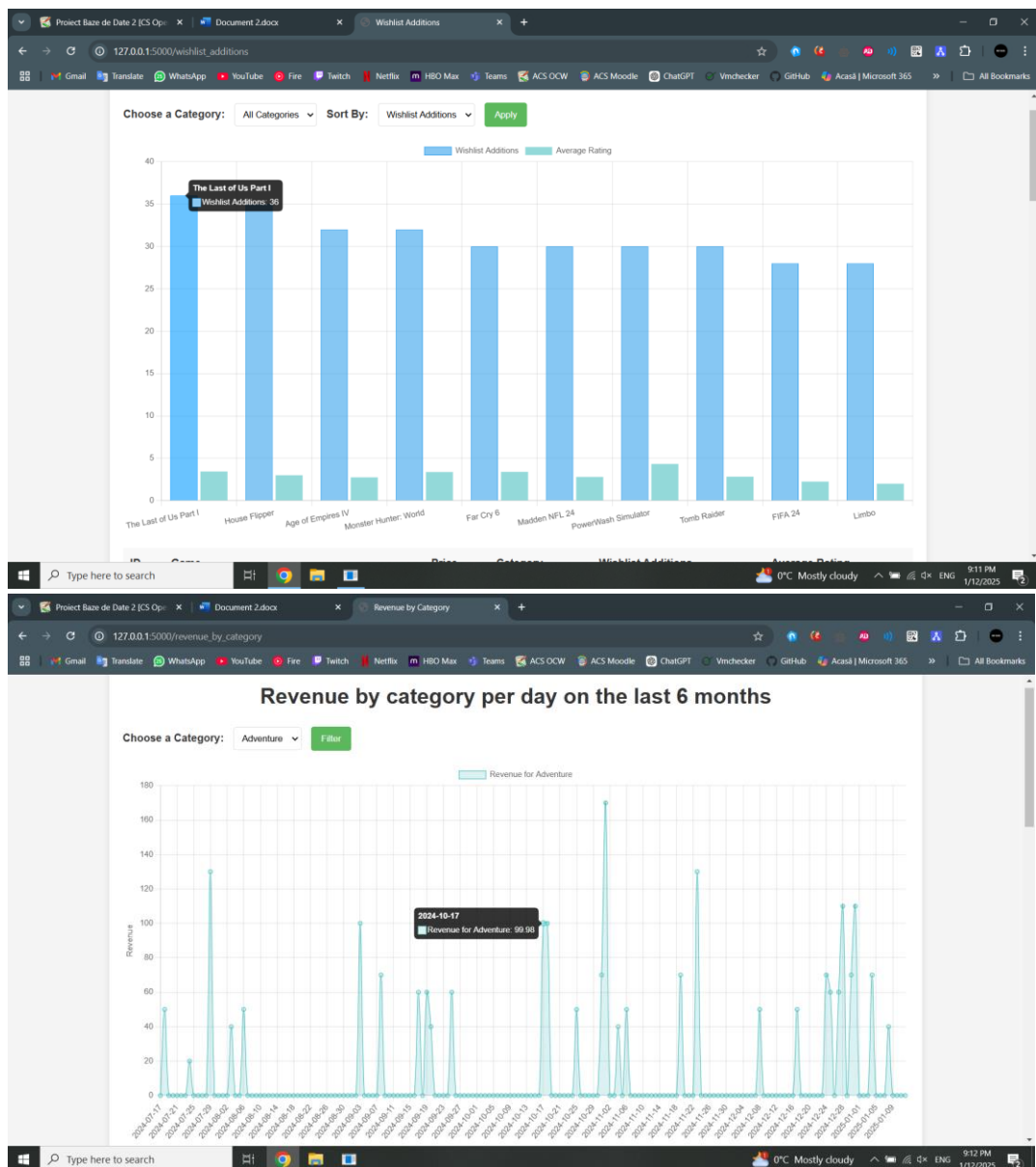


## 2.4 Conexiunea aplicației la baza de date

Conexiunea la baza de date este realizată prin metoda **get\_db\_connection**. Aceasta folosește biblioteca pyodbc pentru a crea o conexiune sigură și stabilă. Interogările procedurilor stocate sunt realizate utilizând obiecte cursor, iar datele sunt extrase și procesate în funcție de ruta accesată. Toate conexiunile sunt închise după utilizare pentru a evita scurgerile de resurse.

## 3. Capturi de ecran pentru interfețe și rapoarte





## 4. Concluzii

Proiectul a fost captivant și am apreciat diversitatea taskurilor necesare pentru a finaliza tema. În plus, am reușit să înțeleg într-un mod mai detaliat atât fluxul de lucru implicat în crearea unei baze de date, cât și procesul de dezvoltare a unei aplicații care interacționează cu aceasta.

## 5. Bibliografie

- [1] <https://learn.microsoft.com/en-us/sql/connect/python/pyodbc/step-1-configure-development-environment-for-pyodbc-python-development?view=sql-server-ver16&tabs=windows>
- [2] <https://flask.palletsprojects.com/en/stable/>
- [3] <https://ocw.cs.pub.ro/courses/bdd>