Universitatea Transilvania din Brașov
Facultatea de Matematică și Informatică

DOCUMENTAȚIA PROIECTULUI

GameLibraryAPI

STUDENȚI
Vișoiu Radu
Opriș Liviu Vlad

Contents

1 Prezentarea proiectului **3**

2 Tehnologiile folosite **3**

3 Baza de date: diagrama și prezentare **3**
3.1 Structura bazei de date (textual) 3
3.2 Prezentare tabele și relații 4
3.3 Diagrama bazei de date 4

4 Prezentarea API-ului **4**
4.1 Exemple endpoint-uri 4
4.2 Screenshot Swagger 5

5 Utilizare aplicație **6**
5.1 Autentificare 6

6 Concluzii și contribuții **6**

7 Link GIT către codul proiectului **7**

1 Prezentarea proiectului

GameLibraryAPI este o aplicație web RESTful ce gestionează o bibliotecă de jocuri video. Scopul proiectului este de a oferi utilizatorilor o modalitate centralizată de a-și organiza și administra colecția de jocuri, dezvoltatori, publisheri și genuri. Proiectul rezolvă problema fragmentării informațiilor despre jocuri și oferă operații CRUD securizate pentru toate entitățile principale.

Probleme rezolvate:

- Centralizarea informațiilor despre jocuri, dezvoltatori, publisheri, genuri și utilizatori.
- Gestionarea relațiilor multiple între entități.
- Securizarea accesului la date prin autentificare JWT (roluri User/Admin).
- Oferirea unui API documentat automat (Swagger).

2 Tehnologiile folosite

- **Backend:** .NET (C#), ASP.NET Core Web API
- **ORM:** Entity Framework Core
- **Baza de date:** SQL Server
- **Autentificare:** JWT (JSON Web Token)
- **Documentare API:** Swagger (OpenAPI)
- **Dependency Injection:** nativ .NET
- **Mediu dezvoltare:** Visual Studio

3 Baza de date: diagrama și prezentare

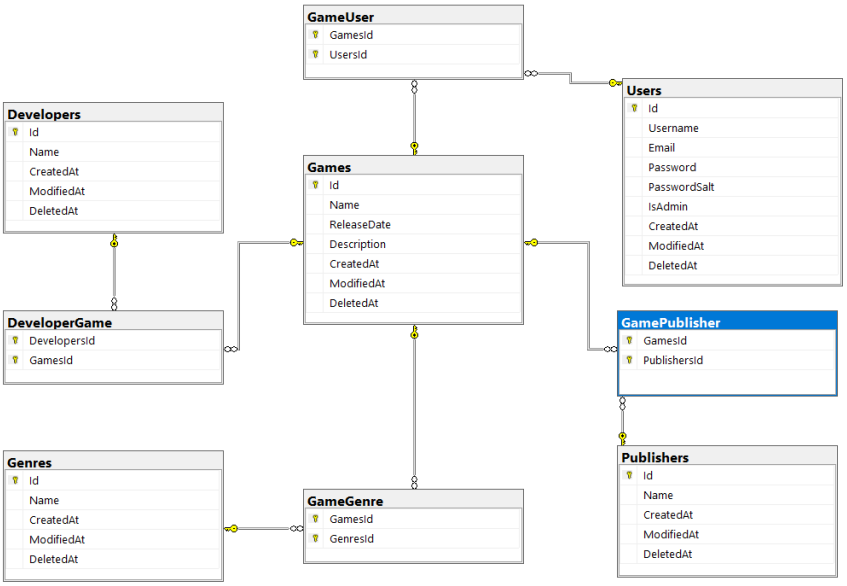
3.1 Structura bazei de date (textual)

- **User** (Id, Username, Email, Password, PasswordSalt, IsAdmin, CreatedAt, ModifiedAt, DeletedAt)
- **Game** (Id, Name, ReleaseDate, Description, CreatedAt, ModifiedAt, DeletedAt)
- **Developer** (Id, Name, CreatedAt, ModifiedAt, DeletedAt)
- **Publisher** (Id, Name, CreatedAt, ModifiedAt, DeletedAt)
- **Genre** (Id, Name, CreatedAt, ModifiedAt, DeletedAt)
- **Tabele de legătură:** DeveloperGame, GamePublisher, GameGenre, GameUser

3.2 Prezentare tabele și relații

- **User**: utilizatorii aplicației, pot avea rol User sau Admin, stochează datele de login și lista de jocuri asociate.
- **Game**: joc cu relații M:N către Developeri, Publisheri, Genuri, Utilizatori.
- **Developer**: informații despre dezvoltatori (M:N cu Game).
- **Publisher**: informații despre publisheri (M:N cu Game).
- **Genre**: genuri de jocuri (M:N cu Game).

3.3 Diagrama bazei de date



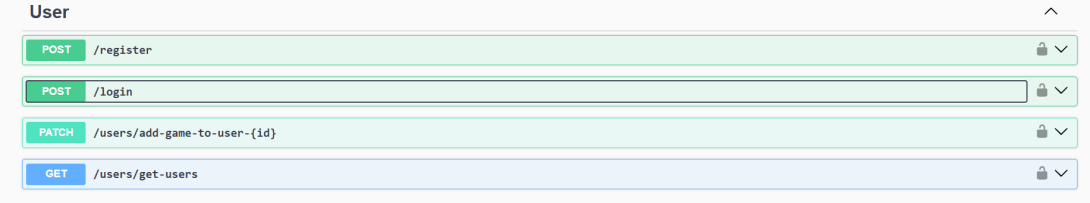
4 Prezentarea API-ului

4.1 Exemple endpoint-uri

- **User**
  - POST /users/register – Înregistrare utilizator
  - POST /users/login – Login utilizator
- **Game**
  - POST /game/add-game – Adaugă joc nou
  - PATCH /game/{id} – Actualizează joc
  - GET /game/get-games – Listează toate jocurile
  - DELETE /game/soft-delete/{id} – Ștergere logică
- **Developer**

– POST /developer/add-developer – Adaugă dezvoltator
– PATCH /developer/{id} – Actualizează dezvoltator
– GET /developer/get-developers – Listează dezvoltatorii
– DELETE /developer/soft-delete/{id} – Ștergere logică

4.2 Screenshot Swagger



5 Utilizare aplicație

Aplicația GameLibraryAPI permite accesul la funcționalitățile sale doar utilizatorilor autentificați. Nu există diferențe de drepturi sau funcționalități între utilizatorii de tip *user* și cei de tip *admin*, ambii având acces la toate operațiile disponibile în API (CRUD pentru jocuri, dezvoltatori, publisheri, genuri, etc).

- Un utilizator care nu este logat are acces doar la operațiile de înregistrare (**register**) și autentificare (**login**).
- După ce se autentifică, utilizatorul primește un token JWT și poate accesa toate funcționalitățile aplicației.
- Nu există restricții suplimentare de rol; orice utilizator logat are acces complet la toate resursele și operațiile API-ului, indiferent dacă este user sau admin.

5.1 Autentificare

Autentificarea se realizează prin JWT (JSON Web Token): la logare, utilizatorul primește un token care, atâta timp cât este atașat în header-ul cererilor, permite accesul la toate funcționalitățile aplicației.

6 Concluzii și contribuții

Împărțirea task-urilor:

- **Radu:** S-a ocupat de structurarea și crearea bazei de date, implementând *Entities*, *Repositories* și *Context* în proiectul Database. În modulul Core a creat mapările (Mapping) și o parte din DTOs și a lucrat la câteva endpoint-uri. De asemenea, Radu a redactat documentația proiectului.
- **Vlad:** A dezvoltat partea de *services* și majoritatea endpoint-urilor din controllers, a adăugat request-uri și responses în DTOs, s-a ocupat de funcționalitatea de logare și a dezvoltat middleware-ul. Vlad s-a ocupat totodată și de testarea întregii aplicații.

Ce am învățat:

- Am aprofundat .NET și EF Core.
- Am învățat să structurăm un API RESTful modern, cu autentificare și roluri.
- Am folosit documentarea automată cu Swagger.
- Am colaborat eficient folosind GIT și code-review.

7 Link GIT către codul proiectului

https://github.com/vld2405/GameLibraryAPI