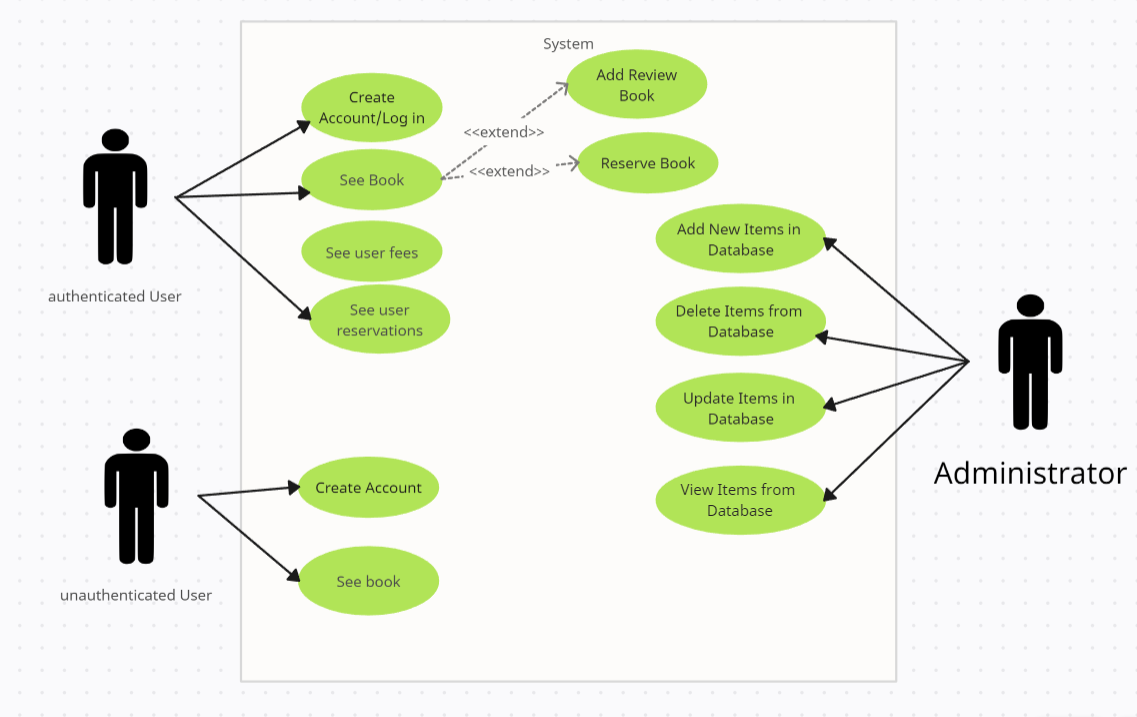
Online Library Project

Smaranda Aida

Use case diagram:



Roluri și Funcționalități:

Utilizator Neconectat:

Poate crea un cont în aplicație.

Are acces doar la vizualizarea cărților disponibile.

Nu poate rezerva cărți sau adăuga recenzii.

Utilizator Conectat:

Are toate funcționalitățile utilizatorului neconectat.

Poate vizualiza penalizările și rezervările sale anterioare.

Poate rezerva cărți disponibile.

Poate adăuga recenzii pentru cărțile citite.

Administrator:

Are toate funcționalitățile utilizatorului conectat.

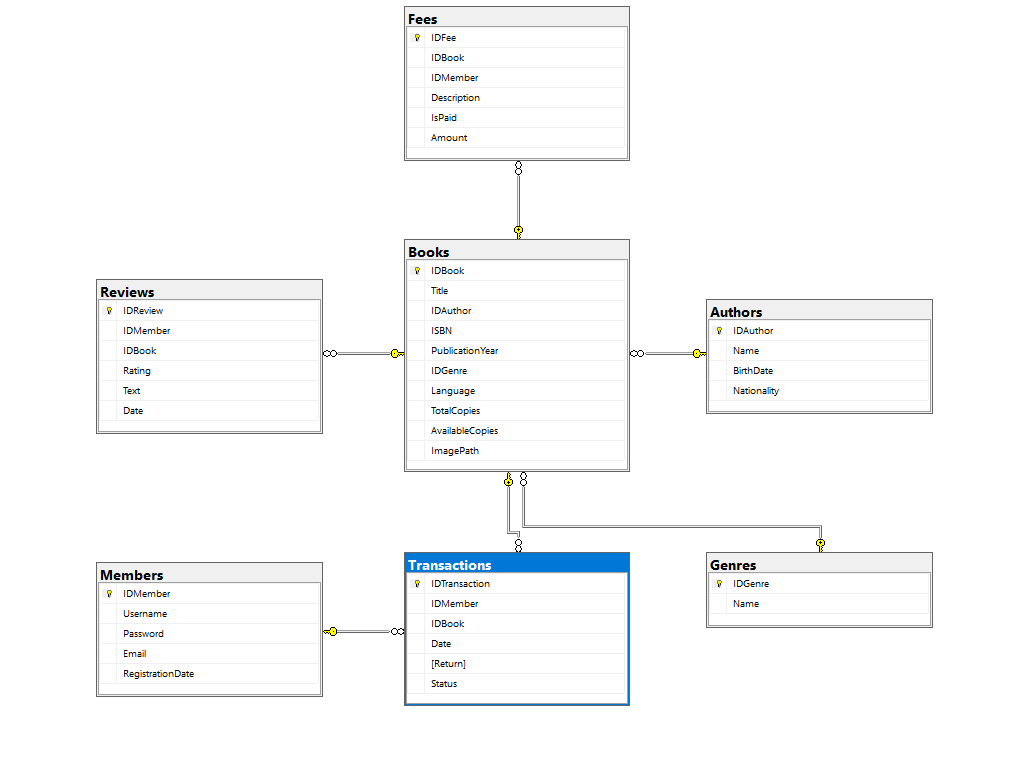
Are privilegii extinse de modificare a datelor din baza de date (insert, edit, delete).

Are acces la datele confidențiale.

Poate gestiona toate aspectele aplicației, inclusiv adăugarea, editarea și ștergerea informațiilor despre cărți, autori, genuri, membri, tranzacții etc.

Poate gestiona conturile utilizatorilor.

Diagrama bazei de date



Arhitectura Generală:

Aplicația MVC (Model-View-Controller) este structurată în jurul a șase entități principale, fiecare reprezentată de un model și gestionată printr-un controller corespunzător. Aceste entități includ: Transactions, Members, Fees, Books, Authors, Genres și Reviews.

Componente:

Models:

Fiecare entitate are un model asociat care definesc structura datelor pentru respectivele tabele din baza de date. Acestea includ: TransactionModel, MemberModel, FeeModel, BookModel, AuthorModel, GenreModel și ReviewModel.

Repositories:

Pentru fiecare model există un repository corespunzător, care gestionează interacțiunea cu baza de date pentru acea entitate. Repository-urile sunt responsabile de operațiile CRUD (Create, Read, Update, Delete) pentru datele asociate.

Controllers:

Fiecare entitate are un controller dedicat care acționează ca intermediar între model și view. Acestea includ: TransactionsController, MembersController, FeesController, BooksController, AuthorsController, GenresController și ReviewsController.

Views:

Pentru fiecare controller, există seturi corespunzătoare de view-uri care permit utilizatorilor să interacționeze cu datele. De exemplu, pentru BooksController, ar exista view-uri pentru afișare, editarea, ștergerea cărților și adăugarea de noi cărți.

Pagini Principale:

Home:

Controllerul Home și pagina asociată oferă o intrare generală în aplicație.

Index:

Pagina principală a aplicației, gestionată de HomeController, unde utilizatorii pot accesa informații generale despre cărți și pot naviga către detaliile acestora.

IndexAdmin:

Pagina principală destinată administratorilor, care oferă funcționalități suplimentare față de Index.

Search:

Pagina de căutare unde utilizatorii pot introduce denumirea unei cărți și obține rezultate relevante despre aceasta.

Details:

Pagina care afișează informații detaliate despre o carte, cum ar fi titlul, autorul, genul și permite utilizatorilor să adauge review-uri.

Această structură generală a aplicației MVC permite organizarea și gestionarea eficientă a operațiilor CRUD pentru fiecare entitate, asigurând o separare clară între model, view și controller