In folderol repositories:

* AutorRepository
* public Autor GetAutorById(int id) – returneaza un obiect de tipul autor din baza de date pe baza id-ului
* public void AddAutor(Autor autor) – adauga un autor in baza de date in tabelul Autor
* public Autor GetAutorByFirstNameAndLastName(String firstname, String lastname) – returneaza un obiect de tip autor care are numele egal cu firstname si prenumele egal cu lastname. Daca nu exista un asemenea obiect se intoarce null.
* public int AdaugaAutorDacaENecesar(String numeAutor, String prenumeAutor) – primeste ca parametru numele si prenumele autorului. Intai se verifica daca exista un autor cu numele si prenumele respectiv in baza de date. Daca exista se returneaza id-ul sau, daca nu se creeaza obiectul si dupa se returneaza id-ul obiectului nou creat.
* CarteRepository
  + - public List<Carte> GetCartiByGen(Gen gen) – returneaza o Lista de carti pe baza genului
    - public Carte GetCarteByTitlu(String titlu) – returneaza o Carte pe baza unui titlu
    - public List<Carte> getCartiByTitlu(String titlu) – returneaza o Lista de carti care au titlul == titlu
    - public List<Carte> getAll() – returneaza o Lista cu toate cartile din tabelul Carte
* CititorRepository
  + - public void SchimbaStarea(Cititor cititor) – functie se schimba starea unui cititor din 0 in 1
    - public Cititor GetCititorByNumeAndPrenume(String nume, String prenume) – returneaza un cititor pe baza numelui si a prenumelui
    - public Cititor GetCititorByEmail(String email) – returneaza un cititor pe baza emailului
    - public void AddCititor(Cititor cititor) – adauga un cititor in bd
    - public int? GetStareByEmail(String email) – returneaza starea unui cititor pe baza emailului
* GenRepository
  + - public void AddGen(Gen gen) – adauga un gen in db
    - public int AddGenAndGetId(Gen gen) – adauga un gen si returneaza id-ul
    - public Gen GetGenByDescriere(String descriere) – returneaza un gen pe baza descrierii
    - public int AdaugaGenDacaENecesar(String descriereGen) – adauga un gen daca nu exista si returneaza id-ul acestuia
* ImprumutRepository
  + - public bool EsteDisponibilaCartea(int carteId) – verifica daca o carte e imprumutata sau nu. Daca exista in tabelul Imprumut o inregistrare cu carteId = carteId si la care data de returnare este null inseamna ca acea carte e imprumutata (data returnarii se completeaza cand se returneaza o carte)
    - public Imprumut GetImprumutByUserId(int userId, List<Carte> carti) – returneaza un imprumut pe baza userIdului dintr-o lista de carti(care au acelasi titlu)
    - public List<String> NumarCititoriPentruPerioada(DateTime low, DateTime high) – returneaza o lista cu cititorii dintr-o perioada de timp. Se face join intre tabela Cititor si Imprumut.
    - public List<String> CeleMaiSolicitateGenuri() – returneaza o lista cu cele mai solicitate genuri (genurile cele mai imprumutate).
    - public List<String> CelMaiSolicitatAutor() – cel mai solicitat autor pe baza imprumuturilor
    - public List<String> CeaMaiSolicitataCarte() – cea mai solicitata carte pe baza imprumuturilor
    - public List<Review> ReviewuriPentruCarte(String titlu) – returneaza o lista cu reviewurile pentru o anumita carte

In folderul “Etape” am facut cate o clasa pentru fiecare “etapa” din proiect unde am apelat una sau mai multe din functiile definite in clasele de mai sus pentru a realiza functionalitatea necesara. Aceste metode vor fi apelate din proiectul winforms.

* AchizitieCarte
  + public void AchizitioneazaCarti(int numarExemplare, String descriereGen, String numeAutor, String prenumeAutor, String titluCarte) – achizitioneaza un numar de carti. Daca genul nu exista se va crea un gen nou, acelasi lucru este valabil si pentru autor. Fiecare carte va primi un id nou.
* ImprumutareCarte
  + public DateTime? DataLaCareVaPuteaFiImprumutataCartea(String titlu) – returneaza cea mai apropiata data la care va putea fi imprumutata o carte in cazul in care aceasta nu e disponibila
  + public void ImprumutaCarte(String titluCarte, String emailCititor) – se imprumuta o carte unui cititor
  + public int? VerificaDacaCarteaEsteDisponibila(String titlu) – se verifica daca o carte e disponibila
* RestituireCarte
  + public int InapoiazaCarte(String emailCititor, String titluCarte, String review) – functie care inapoiaza o carte. Se apeleaza metodele din repository. In cazul in care un utilizator a returnat de 2 ori carti mai tarziu se va schimba starea acestuia
* Statistici
  + Metode wrapper pentru metodele din imprumutRepository