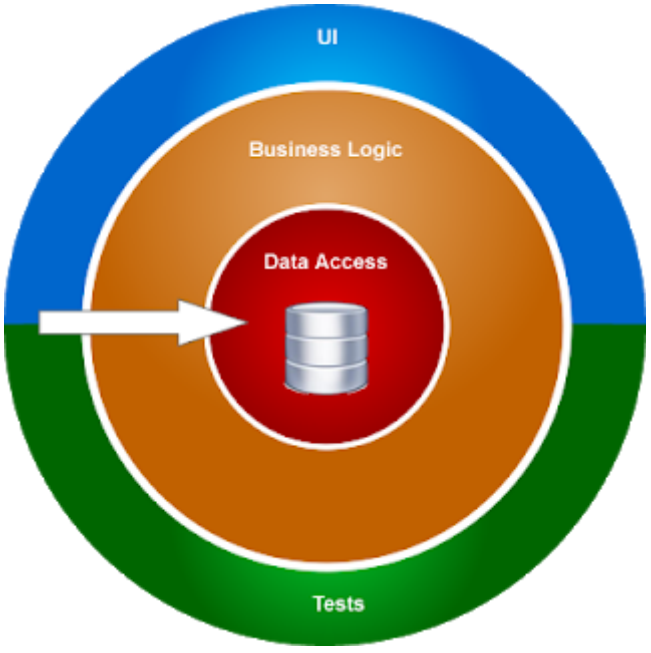
Arhitectura aplicatiei
My Photos

Pentru realizarea aplicatiei My Photos am folosit arhitectura Onion.
De obicei, aceasta permite separarea concern-urilor in 4 layere: Context, Repository, Service, UI.



Context-ul este layer-ul ce detine toate componentele bazei de date(modelul, entitatile, config-urile si maparile dintre entitati si tabele).

Repository-ul este layer-ul in care se scriu toate operatiile primare asupra bazei de date. (de obicei acestea sunt operatiile CRUD)

Service-ul contine toate operatiile de business logic(serviciile aplicatiei). Acesta va folosi operatiile CRUD din mai multe *repository-uri* ale aplicatiei si le va imbina intr-un ansamblu de servicii. (API layer)

UI-ul contine toate informatiile despre partea de View a aplicatiei.(In cazul nostru acestea vor fi *Form-urile*). View-ul va apela una sau mai multe metode din layer-ul de *Service* si va afisa output-ul acestuia pe ecran.

In functie de dimensiunea si complexitatea aplicatiei, aceasta arhitectura poate fi reprezentata si prin 3 layere: Context, Service-Repository si UI. Aceasta strategie este utilizata in momentul in care business logic-ul nu este de dimensiuni mari si nu uitilizeaza mai multe repository-uri intr-o metoda. Aceasta arhitectura am abordat-o si eu pentru realizarea aplicatiei, astadar am utilizat 3 layere:

Context - pachetul de context in care am introdus toata logica si codul bazei de date

PhotoAPI - pachetul de repository/api in care am introdus toate serviciile aplicatiei. Acest pachet se separa in 4 API-uri : PhotoAPI, PersonAPI, Location API si Tags API care vor fi explicate mai cu amanuntire in documentatia layer-ului de API.

MyPhotoViews - pachetul de view care contine partea de Windows Forms.

Matei Acsinte
3A5