

**WildwoodLib** is a back-end API made in ASP.NET with Entity Framework. The project is split into five parts. The domain, data access, application, implementation and API projects. The domain the app solves is a library app used by users, librarians and admins.

**Note:** must unload everything but domain and data-access projects when running migrations and database update commands. The app contains a seed endpoint which will populate the database with some example data.

The app has authorization implemented with JWT. The prior stated roles are just recommendations and the admin can modify the use cases that a user is allowed to use on a per user basis. The app is implemented in Swagger and has an authorize button which can take a JWT token for ease of use.

Anonymous users can search books, register and get a JWT token to authorize themselves. Once authorized they can also use the userCheckout endpoint to see if they have borrowed any books now or in the past as well as search writers and categories.

The librarians can create and modify books. They can also create and modify checkouts. Users browse the books that are available and then go to the library where a librarian uses the app to create a checkout for them.

Finally there is the admin which has access to all endpoints. They can create or modify any table as well as having access to the audit logs examine and search functions. In case of any commands they are all validated using Fluid Validation depending on the entity and return the proper error code and errors.

Every use case is passed to the handler. Inside any error are handled by the global exception handler ensuring proper HTTP code responses. This is also where the per user authorization is handled. Any attempted use of a use case is also logged to the database.

Every tables GET endpoint contains some search functionality depending on the entity and returns pagged results via a generic pagination function.

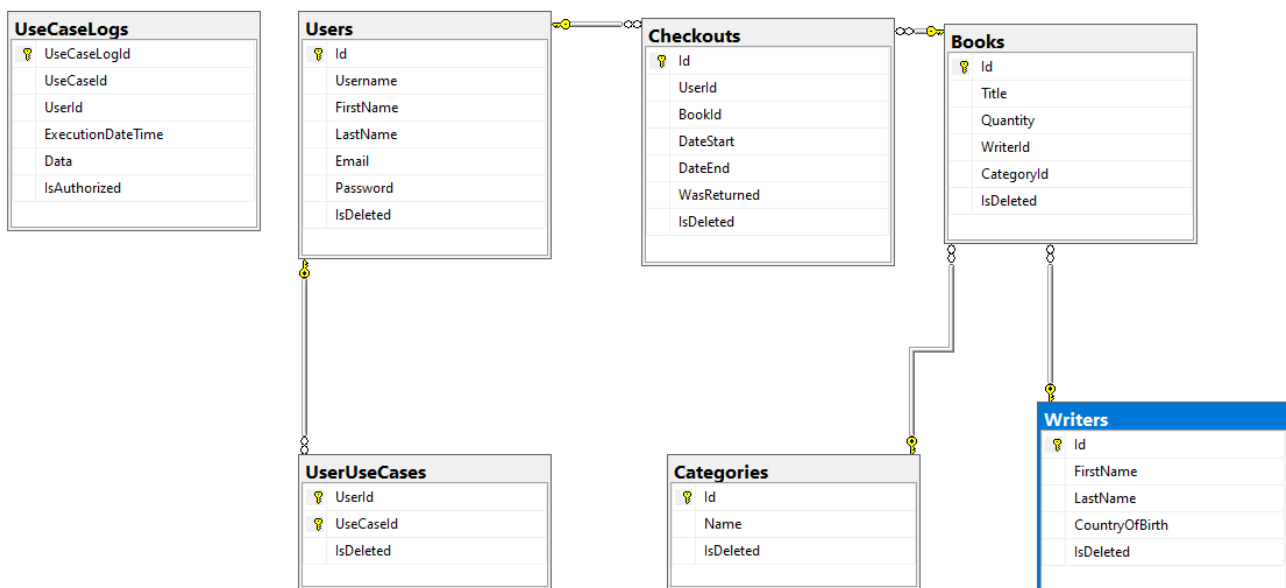


Figure 1: WildwoodLib Database Diagram