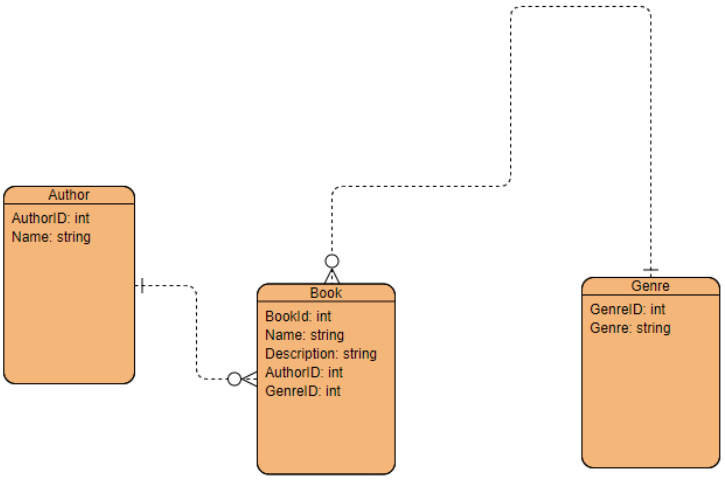
Ryan Deering – X00144631

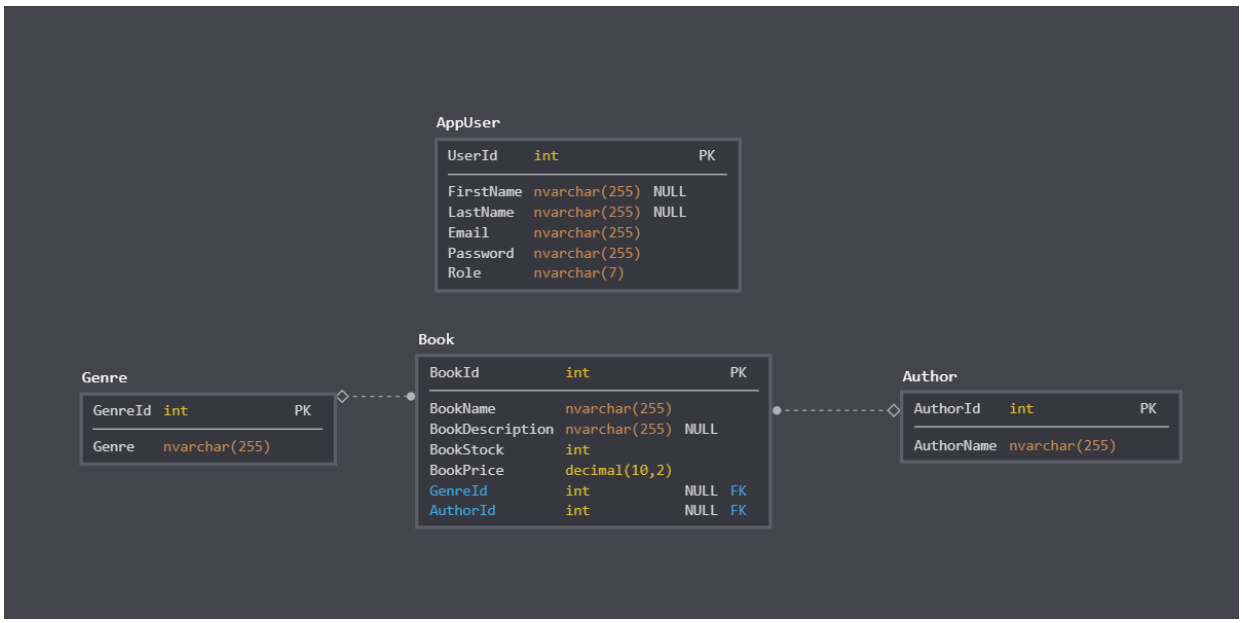
Server-Side Web Development – CA2

I'm a real bookworm, so I opted for a Book Database for the likes of a shop. Users with accounts should be able to fully utilise CRUD on the books, while being able to create & view authors. However, they can not delete the authors. Why? Well, the authors are linked to the books by a foreign key. Deleting the authors, and in turn (providing you've decent database design, of course) would also delete the books. This would be chaotic for this fictional company! So, I opted to not delete or modify the entries. But you can create them to link to the new books.

Initial ERD:



Final ERD:



Endpoints:

For our endpoints, we need */author, /book and /genre*. This is to get their respective data, combined with SQL statements output in JSON format for our client side to interpret and display. For my author endpoint, I defined a GET request to get all the authors in JSON format, get one by id, and to POST one aka make a new author entry. With the book endpoint I did something similar, as well as a DELETE statement to delete a book entry as well as PUT for updating a book (for CRUD.) For genre, I have simple functions to GET all entries and GET by a certain genre.

For my user endpoint, I have a GET request defined to get all users in the database, that is protected by Passport for a logged in user. As well as a GET request for getting by ID.

Access Control:

Public users can view the books and genres. However, for authors, they are only allowed to look at the ones publicly available, as there is UI access control in place through the index.html to view all the authors in the database.

Users with an account can perform standard CRUD on the Book table. However, they can only add Authors, not delete them, due to possible entries in the Book database relying on the Author entries. This is just to avoid database conflict. Users with an account are unable to modify the genres available, as these are set in stone. All input data is validated to prevent attacks.