Part 4: Adding Author

1. Create Author according to *Table 1*
2. Make Author into an entity. Pick an appropriate strategy for authorId
3. In Book and Author set up the relationship according to *Table 1.*
4. Create interface and implementing class according to *Table 2.*
5. (If you have time) Test implementing class of AuthorDao
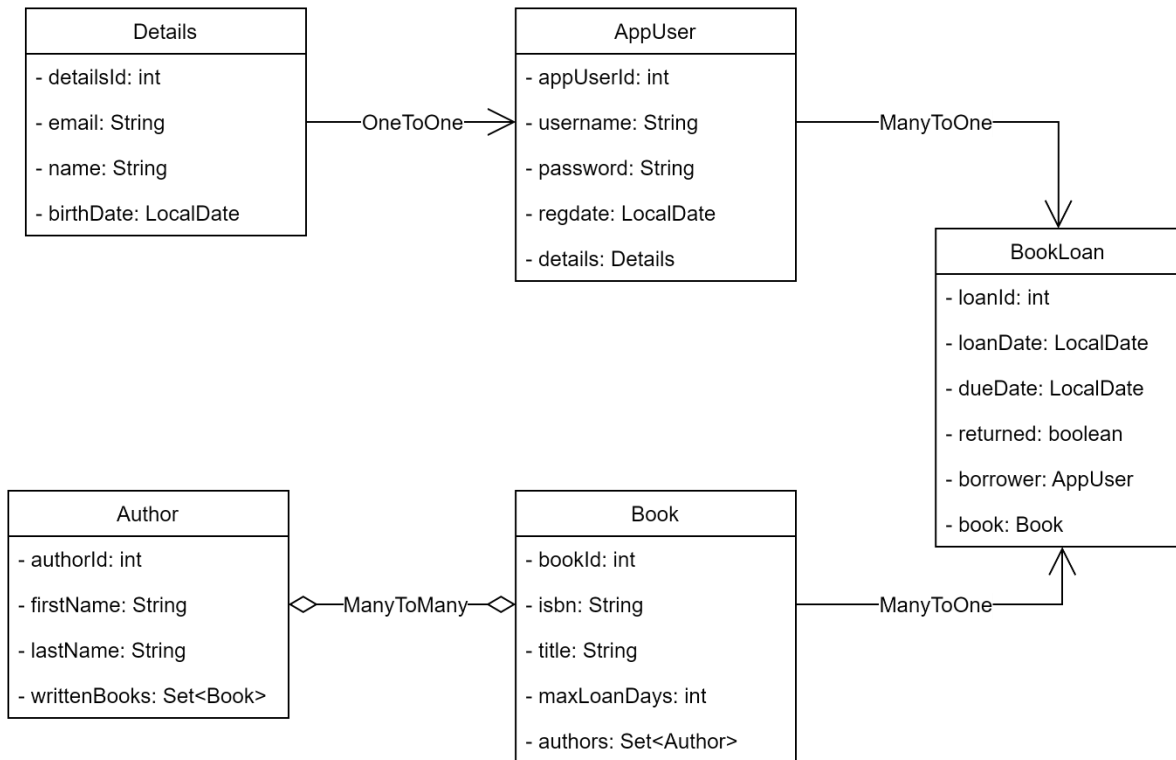6. Commit changes

Table 1

Details

- detailsId: int
- email: String
- name: String
- birthDate: LocalDate

—OneToOne→

AppUser

- appUserId: int
- username: String
- password: String
- regdate: LocalDate
- details: Details

—ManyToOne—

BookLoan

- loanId: int
- loanDate: LocalDate
- dueDate: LocalDate
- returned: boolean
- borrower: AppUser
- book: Book

Author

- authorId: int
- firstName: String
- lastName: String
- writtenBooks: Set<Book>

◇—ManyToMany—◇

Book

- bookId: int
- isbn: String
- title: String
- maxLoanDays: int
- authors: Set<Author>

—ManyToOne—

Table 2

<<Interface>>
AuthorDao

+ findById(int): Author
+ findAll(): Collection<Author>
+ create(): Author
+ update(): Author
+ delete(): void

AuthorDaoImpl