Part 3: Adding Book and BookLoan

1. Create Book according to *Table 1*.
2. Make Book into an entity. Pick an appropriate strategy for bookId.
3. Create BookLoan according to *Table 1.*
4. Make BookLoan into an entity.
   a. Pick an appropriate strategy for loanId.
   b. Map relationship to AppUser according to *Table 1.*
   c. Map relationship to Book according to *Table 1.*
5. Create interfaces and implementing classes according to *Table 2*.
6. (If you have time) Test implementations of BookDao and BookLoanDao.
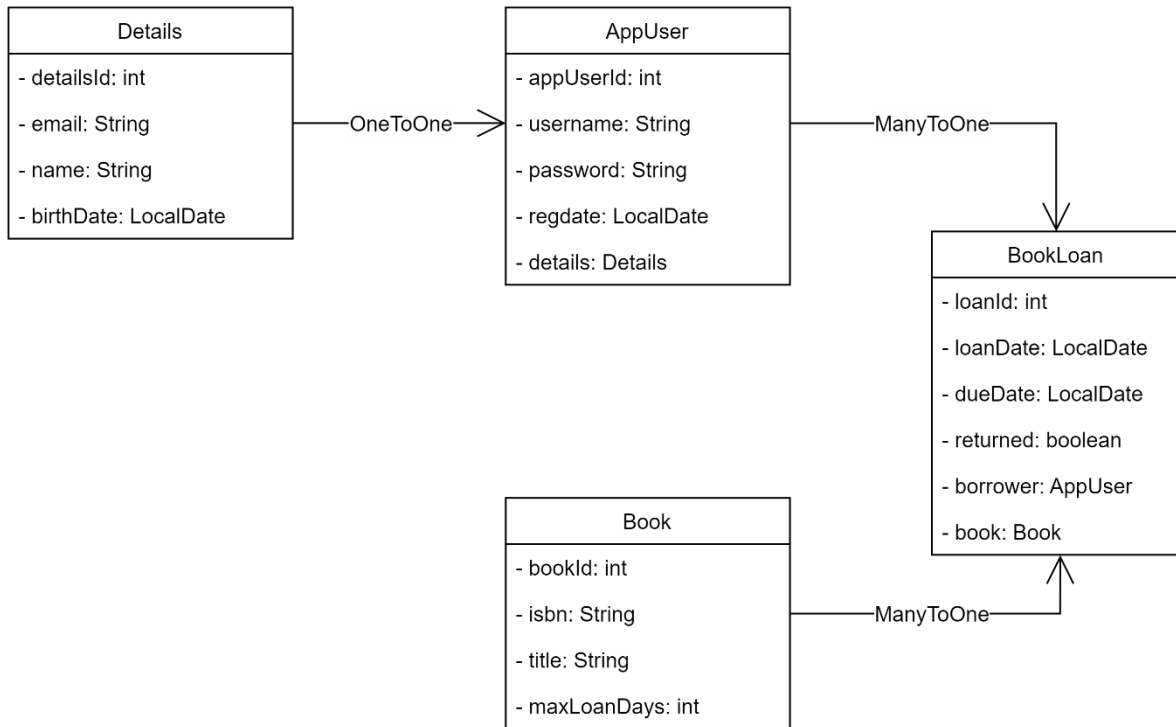7. Commit changes.

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

Book

- bookId: int
- isbn: String
- title: String
- maxLoanDays: int

——ManyToOne——

Table 2

<<Interface>>
BookDao

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

BookDaoImpl

<<Interface>>
BookLoanDao

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

BookLoanDaoImpl