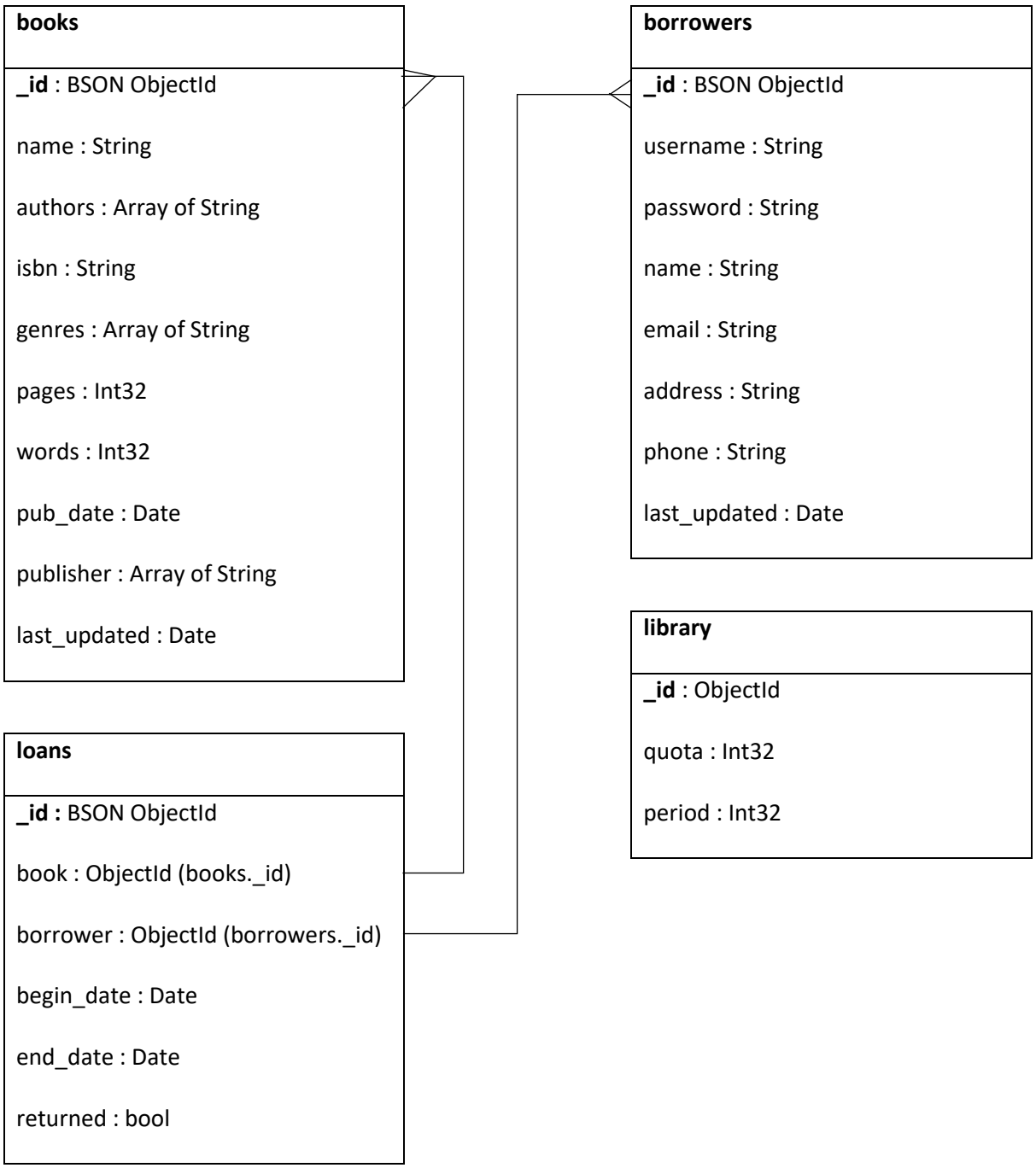
books
_id : BSON ObjectId
name : String
authors : Array of String
isbn : String
genres : Array of String
pages : Int32
words : Int32
pub_date : Date
publisher : Array of String
last_updated : Date
borrowers
_id : BSON ObjectId
username : String
password : String
name : String
email : String
address : String
phone : String
last_updated : Date
library
_id : ObjectId
quota : Int32
period : Int32
loans
_id : BSON ObjectId
book : ObjectId (books._id)
borrower : ObjectId (borrowers._id)
begin_date : Date
end_date : Date
returned : bool

Justification

Librarium has a relatively simple relational database structure, consisting of 3 data storage collections and 1 configuration collection. There are only 2 foreign key links originating from *loans* and ending at *books* and *borrowers*.

Librarium allows the user to check out which books they themselves have borrowed and which books within the library have been borrowed. It would not make sense to dump loan data within *books* as it would take more time for the borrower to check which books they have borrowed. Hence, I decided to have a separate *loans* database. Having a separate *loans* database also allows me to check whether a book has been borrowed by querying for the book's BSON ObjectId and checking for *loans.returned: false* and allows a user to see what books they had borrowed by using their borrower *borrowers._id*, which is important to check whether a borrower has no unreturned books when they delete their account (checking not yet implemented). My pure linking approach allows the library to create unique return dates for each loan and makes it easier to search loans both ways through a *borrowers._id* and through a *books._id*.

Bibliothekos sends more read requests through *Librarium* to the library database than write requests. This should mean that I should have used a hybrid embedding data storage method. However, I thought it would be better for organisation if I separated *loans* from *books*.

LIbrarium allows the storage of historical loan data, which means that a linking database would be better than to store all that data within *books* itself.