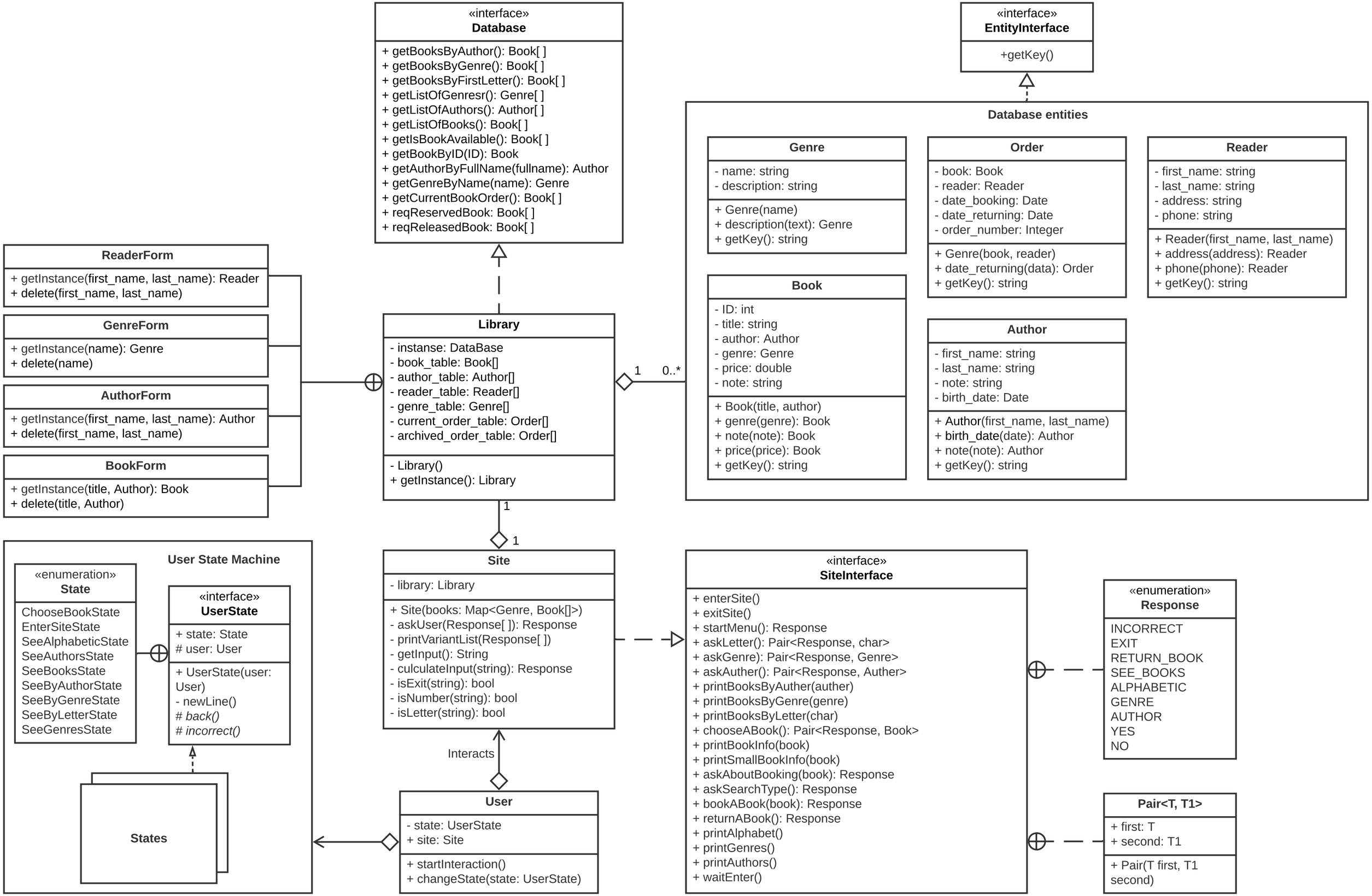
«interface»
Database
+ getBooksByAuthor(): Book[ ]
+ getBooksByGenre(): Book[ ]
+ getBooksByFirstLetter(): Book[ ]
+ getListOfGenresr(): Genre[ ]
+ getListOfAuthors(): Author[ ]
+ getListOfBooks(): Book[ ]
+ getIsBookAvailable(): Book[ ]
+ getBookByID(ID): Book
+ getAuthorByFullName(fullname): Author
+ getGenreByName(name): Genre
+ getCurrentBookOrder(): Book[ ]
+ reqReservedBook: Book[ ]
+ reqReleasedBook: Book[ ]
«interface»
EntityInterface
+getKey()
Database entities
Genre
- name: string
- description: string
+ Genre(name)
+ description(text): Genre
+ getKey(): string
Order
- book: Book
- reader: Reader
- date_booking: Date
- date_returning: Date
- order_number: Integer
+ Genre(book, reader)
+ date_returning(data): Order
+ getKey(): string
Reader
- first_name: string
- last_name: string
- address: string
- phone: string
+ Reader(first_name, last_name)
+ address(address): Reader
+ phone(phone): Reader
+ getKey(): string
Book
- ID: int
- title: string
- author: Author
- genre: Genre
- price: double
- note: string
+ Book(title, author)
+ genre(genre): Book
+ note(note): Book
+ price(price): Book
+ getKey(): string
Author
- first_name: string
- last_name: string
- note: string
- birth_date: Date
+ Author(first_name, last_name)
+ birth_date(date): Author
+ note(note): Author
+ getKey(): string
ReaderForm
+ getInstance(first_name, last_name): Reader
+ delete(first_name, last_name)
GenreForm
+ getInstance(name): Genre
+ delete(name)
AuthorForm
+ getInstance(first_name, last_name): Author
+ delete(first_name, last_name)
BookForm
+ getInstance(title, Author): Book
+ delete(title, Author)
Library
- instanse: DataBase
- book_table: Book[]
- author_table: Author[]
- reader_table: Reader[]
- genre_table: Genre[]
- current_order_table: Order[]
- archived_order_table: Order[]
- Library()
+ getInstance(): Library
1
0..*
1
1
User State Machine
«enumeration»
State
ChooseBookState
EnterSiteState
SeeAlphabeticState
SeeAuthorsState
SeeBooksState
SeeByAuthorState
SeeByGenreState
SeeByLetterState
SeeGenresState
«interface»
UserState
+ state: State
# user: User
+ UserState(user: User)
- newLine()
back()
incorrect()
States
Site
- library: Library
+ Site(books: Map<Genre, Book[]>)
- askUser(Response[ ]): Response
- printVariantList(Response[ ])
- getInput(): String
- culculateInput(string): Response
- isExit(string): bool
- isNumber(string): bool
- isLetter(string): bool
Interacts
User
- state: UserState
+ site: Site
+ startInteraction()
+ changeState(state: UserState)
«interface»
SiteInterface
+ enterSite()
+ exitSite()
+ startMenu(): Response
+ askLetter(): Pair<Response, char>
+ askGenre): Pair<Response, Genre>
+ askAuther(): Pair<Response, Auther>
+ printBooksByAuther(auther)
+ printBooksByGenre(genre)
+ printBooksByLetter(char)
+ chooseABook(): Pair<Response, Book>
+ printBookInfo(book)
+ printSmallBookInfo(book)
+ askAboutBooking(book): Response
+ askSearchType(): Response
+ bookABook(book): Response
+ returnABook(): Response
+ printAlphabet()
+ printGenres()
+ printAuthors()
+ waitEnter()
«enumeration»
Response
INCORRECT
EXIT
RETURN_BOOK
SEE_BOOKS
ALPHABETIC
GENRE
AUTHOR
YES
NO
Pair<T, T1>
+ first: T
+ second: T1
+ Pair(T first, T1 second)