User — **RegistrationPage** — **AccountManager**

- truy cập trang đăng kí
- hiển thị trang đăng kí
- nhập thông tin đăng kí
- gửi thông tin đăng kí
- kiểm tra thông tin đăng kí
- gửi xác nhận đăng kí
- gửi email xác nhận đăng kí
- xác nhận thông tin đăng kí qua email
- nhận thông tin xác nhận đăng kí
- xác nhận thông tin đăng kí
- tạo tài khoản mới
- thông báo đăng kí thành công

Sequence Diagram

Actors / Objects: Customer, Bookstore, Book, BookManager, TicketBook

Messages

- chọn sách để mua
- kiểm tra tình trạng sách
- cung cấp thông tin sách
- cập nhật số lượng sách
- tính tổng giá tiền
- tạo hóa đơn mới
- yêu cầu thanh toán
- thanh toán
- cập nhật trạng thái hóa đơn và ghi chú
- cập nhật tình trạng sách còn lại
- cung cấp sách đã mua

classDiagram

class Book {

-id: Integer

-name: String

-author: String

-price: double

-quantity: Integer

-type: Type

-ticketBook: TicketBook

+getId(): Integer

+setId(Integer): void

+getName(): String

+setName(String): void

+getAuthor(): String

+setAuthor(String): void

+getPrice(): double

+setPrice(double): void

+getQuantity(): Integer

+setQuantity(Integer): void

+getType(): Type

+setType(Type): void

+getTicketBook(): TicketBook

+setTicketBook(TicketBook): void

}

---

class Type {

-id: Integer

-name: String

-books: List<Book>

+getId(): Integer

+setId(Integer): void

+getName(): String

+setName(String): void

+getBooks(): List<Book>

+setBooks(List<Book>): void

}

---

enum Servicez {

RENT

RETURN

BUY

}

---

class Borrower {

-id: Integer

-name: String

-phone: String

-cccd: String

-wallet: Wallet

+getId(): Integer

+setId(Integer): void

+getName(): String

+setName(String): void

+getPhone(): String

+setPhone(String): void

+getCccd(): String

+setCccd(String): void

+getWallet(): Wallet

+setWallet(Wallet): void

---

class Wallet {

-id: Integer

-accountNum: String

-balance: double

+getId(): Integer

+setId(Integer): void

+getAccountNum(): String

+setAccountNum(String): vo

+getBalance(): double

+setBalance(double): void

}

---

class TicketBook {

-id: Integer

-borrower: Borrower

-servicez: Servicez

-bookList: List<Book>

-creatAt: LocalDate

-returnDay: LocalDate

-status: String

-note: String

-totalPrice: double

+getId(): Integer

+setId(Integer): void

+getBorrower(): Borrower

+setBorrower(Borrower): void

+getServicez(): Servicez

+setServicez(Servicez): void

+getBookList(): List<Book>

+setBookList(List<Book>): void

+getCreatAt(): LocalDate

+setCreatAt(LocalDate): void

+getReturnDay(): LocalDate

+setReturnDay(LocalDate): void

+getStatus(): String

+setStatus(String): void

+getNote(): String

+setNote(String): void

+getTotalPrice(): double

+setTotalPrice(double): void

}

This is a UML sequence diagram. The content is rotated 90 degrees.

Participants: Borrower, LS, Service, BookInventory, TicketBook

Messages:
- Borrower → Service: Chọn sách cần thuê
- Service → BookInventory: Truy vấn thông tin sách
- BookInventory → Service: Trả về thông tin sách
- Service → Borrower: Xác nhận thông tin thuê sách
- LS → BookInventory: Kiểm tra sách có sẵn và đủ số lượng
- LS → TicketBook: Tạo phiếu mượn sách mới
- TicketBook → LS: Trả về thông tin phiếu mượn sách
- LS → BookInventory: Cập nhật số lượng sách đang được thuê
- BookInventory → LS: Trả về số lượng sách đã được cập nhật
- LS → TicketBook: Thiết lập lịch sử cho quyển sách
- TicketBook → LS: Trả về lịch sử phiếu mượn sách