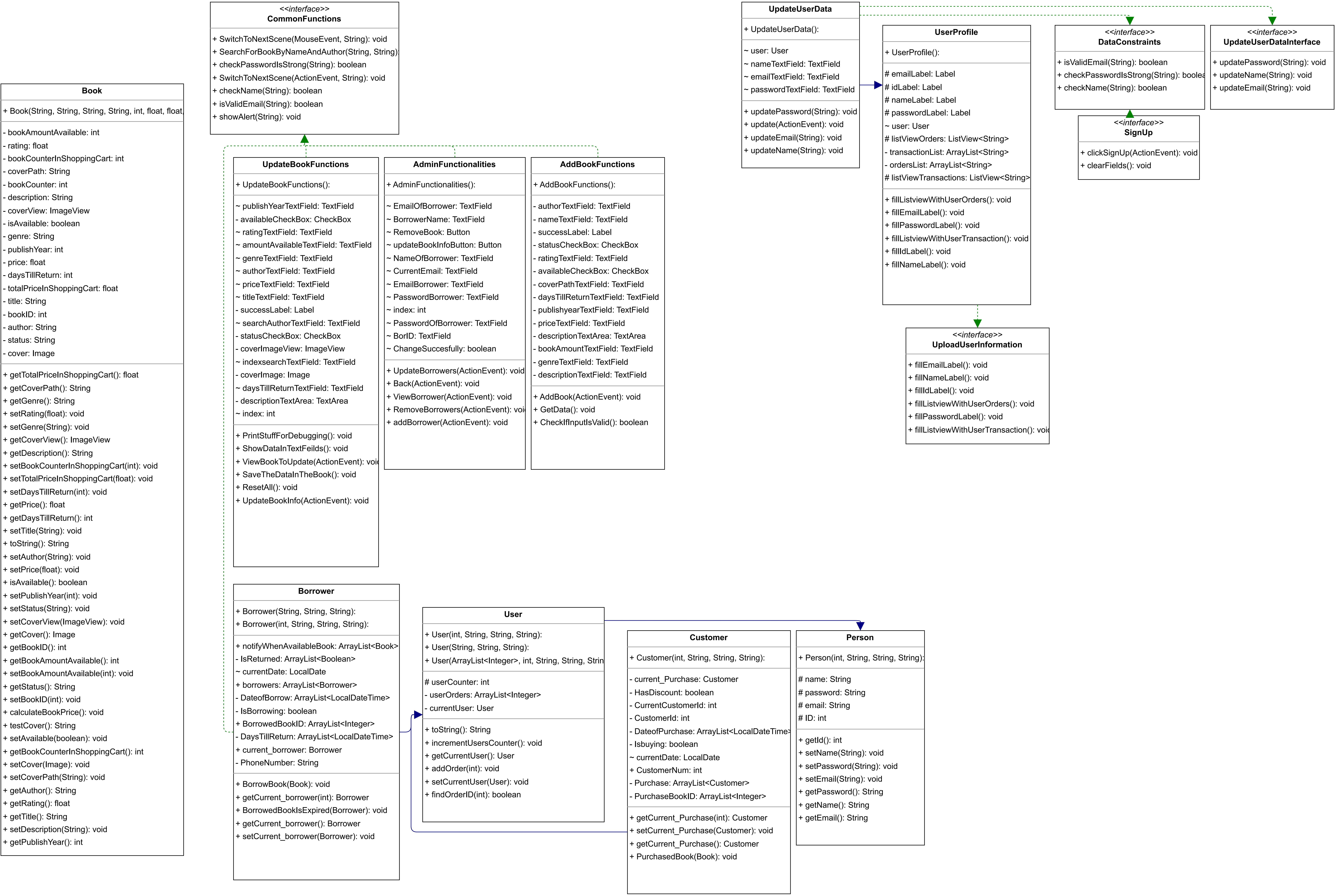

Book

+ Book(String, String, String, String, int, float, float,

- bookAmountAvailable: int
- rating: float
- bookCounterInShoppingCart: int
- coverPath: String
- bookCounter: int
- description: String
- coverView: ImageView
- isAvailable: boolean
- genre: String
- publishYear: int
- price: float
- daysTillReturn: int
- totalPriceInShoppingCart: float
- title: String
- bookID: int
- author: String
- status: String
- cover: Image

+ getTotalPriceInShoppingCart(): float
+ getCoverPath(): String
+ getGenre(): String
+ setRating(float): void
+ setGenre(String): void
+ getCoverView(): ImageView
+ getDescription(): String
+ setBookCounterInShoppingCart(int): void
+ setTotalPriceInShoppingCart(float): void
+ setDaysTillReturn(int): void
+ getPrice(): float
+ getDaysTillReturn(): int
+ setTitle(String): void
+ toString(): String
+ setAuthor(String): void
+ setPrice(float): void
+ isAvailable(): boolean
+ setPublishYear(int): void
+ setStatus(String): void
+ setCoverView(ImageView): void
+ getCover(): Image
+ getBookID(): int
+ getBookAmountAvailable(): int
+ setBookAmountAvailable(int): void
+ getStatus(): String
+ setBookID(int): void
+ calculateBookPrice(): void
+ testCover(): String
+ setAvailable(boolean): void
+ getBookCounterInShoppingCart(): int
+ setCover(Image): void
+ setCoverPath(String): void
+ getAuthor(): String
+ getRating(): float
+ getTitle(): String
+ setDescription(String): void
+ getPublishYear(): int

<<interface>>
CommonFunctions

+ SwitchToNextScene(MouseEvent, String): void
+ SearchForBookByNameAndAuthor(String, String):
+ checkPasswordIsStrong(String): boolean
+ SwitchToNextScene(ActionEvent, String): void
+ checkName(String): boolean
+ isValidEmail(String): boolean
+ showAlert(String): void

UpdateBookFunctions

+ UpdateBookFunctions():

~ publishYearTextField: TextField
- availableCheckBox: CheckBox
~ ratingTextField: TextField
~ amountAvailableTextField: TextField
~ genreTextField: TextField
~ authorTextField: TextField
~ priceTextField: TextField
~ titleTextField: TextField
- successLabel: Label
~ searchAuthorTextField: TextField
- statusCheckBox: CheckBox
- coverImageView: ImageView
~ indexsearchTextField: TextField
- coverImage: Image
~ daysTillReturnTextField: TextField
- descriptionTextArea: TextArea
~ index: int

+ PrintStuffForDebugging(): void
+ ShowDataInTextFeilds(): void
+ ViewBookToUpdate(ActionEvent): voi
+ SaveTheDataInTheBook(): void
+ ResetAll(): void
+ UpdateBookInfo(ActionEvent): void

AdminFunctionalities

+ AdminFunctionalities():

~ EmailOfBorrower: TextField
~ BorrowerName: TextField
~ RemoveBook: Button
~ updateBookInfoButton: Button
~ NameOfBorrower: TextField
~ CurrentEmail: TextField
~ EmailBorrower: TextField
~ PasswordBorrower: TextField
~ index: int
~ PasswordOfBorrower: TextField
~ BorID: TextField
~ ChangeSuccesfully: boolean

+ UpdateBorrowers(ActionEvent): void
+ Back(ActionEvent): void
+ ViewBorrower(ActionEvent): void
+ RemoveBorrowers(ActionEvent): voi
+ addBorrower(ActionEvent): void

AddBookFunctions

+ AddBookFunctions():

- authorTextField: TextField
- nameTextField: TextField
- successLabel: Label
- statusCheckBox: CheckBox
- ratingTextField: TextField
- availableCheckBox: CheckBox
- coverPathTextField: TextField
- daysTillReturnTextField: TextField
- publishyearTextField: TextField
- priceTextField: TextField
- descriptionTextArea: TextArea
- bookAmountTextField: TextField
- genreTextField: TextField
- descriptionTextField: TextField

+ AddBook(ActionEvent): void
+ GetData(): void
+ CheckIfInputIsValid(): boolean

UpdateUserData

+ UpdateUserData():

~ user: User
~ nameTextField: TextField
~ emailTextField: TextField
~ passwordTextField: TextField

+ updatePassword(String): void
+ update(ActionEvent): void
+ updateEmail(String): void
+ updateName(String): void

UserProfile

+ UserProfile():

# emailLabel: Label
# idLabel: Label
# nameLabel: Label
# passwordLabel: Label
~ user: User
# listViewOrders: ListView<String>
- transactionList: ArrayList<String>
- ordersList: ArrayList<String>
# listViewTransactions: ListView<String>

+ fillListviewWithUserOrders(): void
+ fillEmailLabel(): void
+ fillPasswordLabel(): void
+ fillListviewWithUserTransaction(): void
+ fillIdLabel(): void
+ fillNameLabel(): void

<<interface>>
DataConstraints

+ isValidEmail(String): boolean
+ checkPasswordIsStrong(String): boolea
+ checkName(String): boolean

<<interface>>
UpdateUserDataInterface

+ updatePassword(String): void
+ updateName(String): void
+ updateEmail(String): void

<<interface>>
SignUp

+ clickSignUp(ActionEvent): void
+ clearFields(): void

<<interface>>
UploadUserInformation

+ fillEmailLabel(): void
+ fillNameLabel(): void
+ fillIdLabel(): void
+ fillListviewWithUserOrders(): void
+ fillPasswordLabel(): void
+ fillListviewWithUserTransaction(): voi

Borrower

+ Borrower(String, String, String):
+ Borrower(int, String, String, String):

+ notifyWhenAvailableBook: ArrayList<Book>
- IsReturned: ArrayList<Boolean>
~ currentDate: LocalDate
+ borrowers: ArrayList<Borrower>
- DateofBorrow: ArrayList<LocalDateTime>
- IsBorrowing: boolean
+ BorrowedBookID: ArrayList<Integer>
- DaysTillReturn: ArrayList<LocalDateTime>
+ current_borrower: Borrower
- PhoneNumber: String

+ BorrowBook(Book): void
+ getCurrent_borrower(int): Borrower
+ BorrowedBookIsExpired(Borrower): void
+ getCurrent_borrower(): Borrower
+ setCurrent_borrower(Borrower): void

User

+ User(int, String, String, String):
+ User(String, String, String):
+ User(ArrayList<Integer>, int, String, String, Strin

# userCounter: int
- userOrders: ArrayList<Integer>
- currentUser: User

+ toString(): String
+ incrementUsersCounter(): void
+ getCurrentUser(): User
+ addOrder(int): void
+ setCurrentUser(User): void
+ findOrderID(int): boolean

Customer

+ Customer(int, String, String, String):

- current_Purchase: Customer
- HasDiscount: boolean
- CurrentCustomerId: int
- CustomerId: int
- DateofPurchase: ArrayList<LocalDateTime>
- Isbuying: boolean
~ currentDate: LocalDate
+ CustomerNum: int
- Purchase: ArrayList<Customer>
- PurchaseBookID: ArrayList<Integer>

+ getCurrent_Purchase(int): Customer
+ setCurrent_Purchase(Customer): void
+ getCurrent_Purchase(): Customer
+ PurchasedBook(Book): void

Person

+ Person(int, String, String, String):

# name: String
# password: String
# email: String
# ID: int

+ getId(): int
+ setName(String): void
+ setPassword(String): void
+ setEmail(String): void
+ getPassword(): String
+ getName(): String
+ getEmail(): String

RemoveBookFunctions

+ RemoveBookFunctions():

- image: Image
- RemoveTitle: Label
- RemoveDes: Label
- RemoveRating: Label
~ NameAuthor: String
- RemoveGenre: Label
- RemoveCoverPath: Label
- RemoveBookname: TextField
- RemoveNameAuthor: TextField
~ index: int
- RemovePrice: Label
- RemoveImage: ImageView
~ NameBook: String
- RemoveAuthor: Label
- RemoveAvailable: Label
- RemoveExpire: Label
- RemovePublishYear: Label
- Success: Label

+ ViewBookToRemove(ActionEvent): voi
+ DeleteBook(ActionEvent): void

UserInterfaceIcons

+ UserInterfaceIcons():

+ goBackOnHover: Image
+ shoppingCartOnClick: Image
+ Star5Selected: Image
+ Star2: Image
+ Star1Selected: Image
+ Star5: Image
+ Star3Selected: Image
+ goBack: Image
+ shoppingCart: Image
+ shoppingCartOnHover: Image
+ profileOnHover: Image
+ profile: Image
+ Star4: Image
+ Star1: Image
+ Star2Selected: Image
+ Star3: Image
+ Star4Selected: Image
+ goBackOnClick: Image
+ profileOnClick: Image

- loadImage(String): Image

ShoppingCart

+ ShoppingCart():

- bookCount: int
- totalPrice: float
- paymentMethod: String
- discountAmount: float
- discounts: List<Discount>
- books: List<Book>

+ setBooks(List<Book>): void
+ calculateTotalPrice(): void
+ getBookCount(): int
+ setPaymentMethod(String): void
+ updateQuantity(Book, int): void
+ setBookCount(int): void
+ applyDiscount(String): void
+ getPaymentMethod(): String
+ setTotalPrice(float): void
+ getDiscounts(): List<Discount>
+ removeBook(Book): void
+ addBook(Book): void
+ getBooks(): List<Book>
+ getTotalPrice(): float
+ getDiscountAmount(): float
+ setDiscounts(List<Discount>): void
+ setDiscountAmount(float): void
+ determinePaymentMethod(boolean): voi

<<interface>>
ShoppingCartCommand

+ execute(ShoppingCart): void

QuantityUpdater

+ QuantityUpdater(Book, int):

- bookAmountAvailable: int
- book: Book
- updatedQuantity: int

+ execute(ShoppingCart): void

PaymentMethodDeterminer

+ PaymentMethodDeterminer(boolean):

- paymentIsCash: boolean

+ execute(ShoppingCart): void

DiscountApplier

+ DiscountApplier(String):

- promoCode: String

+ execute(ShoppingCart): void

BookAdder

+ BookAdder(Book):

- book: Book

+ execute(ShoppingCart): void

BookRemover

+ BookRemover(Book):

- book: Book

+ execute(ShoppingCart): void

TotalPriceCalculator

+ TotalPriceCalculator():

+ execute(ShoppingCart): void

Transaction

+ Transaction(Book, Borrower, LocalDateTime, LocalDateT

~ borrowedBook: Book
~ borrower: Borrower
~ borrowDate: LocalDateTime
~ returnDate: LocalDateTime

+ setBorrowedBook(Book): void
+ setBorrowDate(LocalDateTime): void
+ getReturnDate(): LocalDateTime
+ setReturnDate(LocalDateTime): void
+ toString(): String
+ setBorrower(Borrower): void
+ getBorrower(): Borrower
+ getBorrowedBook(): Book
+ getBorrowDate(): LocalDateTime
+ toString2(): String

FileHandling

+ FileHandling():

+ saveData(): void
+ loadBorrowers(): void
+ saveBooks(): void
+ saveUsers(): void
+ loadTransactions(): void
+ saveTransactions(): void
+ saveBorrowers(): void
+ loadOrders(): void
+ loadBooks(): void
+ saveOrders(): void
+ loadUsers(): void
+ loadData(): void

RegistrationSystem

+ RegistrationSystem():

# usernameTextField: TextField
- stage: Stage
# emailTextField: TextField
- scene: Scene
# enterPasswordField: TextField

+ switchToSignInAsUserScene(ActionEvent): void
+ switchToAdminHomePageScene(ActionEvent): void
+ switchToHomePageScene(ActionEvent): void
+ switchToCreateNewUserAccountScene(ActionEvent):
# showAlert(String): void

Order

+ Order(float, LocalDate, int):

- orderId: int
- orderCounter: int
- price: float
- quantity: int
- orderDate: LocalDate

+ toString(): String
+ getOrderDate(): LocalDate
+ getOrderId(): int
+ getPrice(): float
+ getQuantity(): int

Discount

+ Discount(String, float):

- promoCode: String
- discountPercentage: float
- usedByUser: boolean

+ setUsedByUser(boolean): void
+ setPromoCode(String): void
+ setDiscountPercentage(float): void
+ isUsedByUser(): boolean
+ getDiscountPercentage(): float
+ getPromoCode(): String

FillLists

+ FillLists():

+ listLabel: Label
+ availableList: ListView<String>
+ authorName: Label
+ authorList: ListView<String>

- fillWordWithOnlyAuthorBooks(ArrayList<String>, String): vo
- fillWordWithOnlyAvailableBooks(ArrayList<String>): void
# uploadAuthorBooks(String): void
# uploadAvailableBooks(): void

Library

+ Library():

+ borrowers: ArrayList<Borrower>
+ transactions: ArrayList<Transaction>
- selectedBook: Book
+ users: ArrayList<User>
+ books: ArrayList<Book>
+ orders: ArrayList<Order>

+ getSelectedBook(): Book
+ setSelectedBook(Book): void

Admin

+ Admin():

- username: String
- password: String

+ setUsername(String): void
+ getPassword(): String
+ setPassword(String): void
+ getUsername(): String

Trie

+ Trie():

- root: TrieNode

+ search(String): boolean
+ insert(String): void

TrieNode

+ TrieNode():

~ children: Map<Character, TrieNode>
~ isEnd: boolean

<<interface>>
SearchEngine

+ uploadAllBooks(ArrayList<String>): void
+ searchList(String, List<String>): List<String
+ findBook(String): Book

<<interface>>
Login

+ findUser(String, String): User