ParameterCollection

User
- authentication: Credentials
- /role: Role

+ authorize(credentials)
+ getUpdatableItems()
+ getUpdatableParameters(ItemID)
+ getAuctionDates()
+ modifyItem(ItemID, ItemParameters)
+ fetchFinInformation(Date, Date):
  List<Transaction>, List<Auction>, List<Book>
+ getBooks(): List<Book>
+ getReservationStatus(BookID): String
+ reserveBook(BookID, viewer)

ItemManager

+ filterList(List<Item>, List<Parameter>) : List<Item>
+ getItemTypes(): List<CatalogueItem>
+ getUpdatableItems(): List<Item>
+ getUpdatableParameters(ItemID): List<Parameter>
+ updateItem(ItemID, ItemParameters)
+ getAllItemsOfType(CatalogueItem): List<Items>
+ reserveBook(BookID): String
+ checkAddItemParameters(ItemParameters): Bool

PersonParameters
+ name: String
+ address: Address
+ emailAddress: String {Optional}
+ IBAN: String {Optional}

ItemParameters
+ name: String
+ amount: Int
+ description: String {Optional}

Credentials
+ username: String
+ password: String

Book
- author: String
- reservationStatus: String
- publicationYear: Date

+ reserve(viewer)

soldBook

Date
- day: Int
- month: Int
- year: Int

compare(Date, Date): Int

Search System
+ method(type): type

ItemCategory
- name: String

ResaleItem
- title: String
- age: Int

LP

itemManager

requests
{ List }

Search System Request
- searchID: Int
- maxPrice: Int

requestedProduct

groupedCatalogues
{ List }

CatalogueItem
- description: String
- antiquarianValue: Int
- features: List<String>

1 catalogueManager

CatalogueManager
+ getBook(): List<Book>
+ getReservationStatus(BookID): String
+ reserveBook(BookID)

availableItems

Viewer

reservedBy

requestedBy

Regular Customer
+ field: type

+ method(type): type

1 type

1 auctionManager

AuctionManager
+ getAuctionDates(): List<Date>
+ getAuctionsAsList(): List<Auction>

PoliceDepartment

Item
- ID: Int
- availableAmount: Int
- price: Float
- addedDate: Date
- updatable: Boolean

+ «Item»(List<ItemParameter>): Item
+ update(List<ItemParameter>)
+ geID() : Int
+ getUpdatableParameters():
    List<ItemParameters>
+ getItemTypeAsString(): String
+ delete()
+ satisfyParameters(List<ItemParams>): Bool

offeredItems
{ List }

itemsBought
{ List }

boughtBy

Customer
+ buyItem(itemID)

Seller
+ sellItem(itemID)

soldBy

itemsSold
{ List }

Registree
- registreeID: Int
- name: String
- address: Address
- emailAddress: String
- IBAN: String

Auction
- date: Date
- theme: String
- /nrOfItemsForSale: Int

auction

itemsForSale
{ List }

auctionHistory
{ ordered, List }

auctionsToHold

itemsForSale
{ List }

Auctioneer
+ sellItem(itemID)

PurchasingAgent
+ getAuctionDates(): List<Date>
+ addItem(ItemParameters)

Secretary
+ addRegularCustomer(PersonParameters)
+ addSeller(PersonParameters)
+ getUpdatableItems(): List<Items>
+ getUpdatableParameters(ItemID): List<Parameter>
+ getAuctionDates(): List<Date>
+ updateItem(ItemID, parameters)

Staff

staffMembers
{ List }

Owner
+ getBooks(): List<Item>
+ reserveBook(BookID): String

Transaction
- transactionID: Int
- value: Int
- traderIBAN: String

BookSale

bookSaleHistory
{ ordered, List }

transactionHistory
{ ordered, List }

AdministrationSystem
+ fetchFinInformation(Date, Date):
    List<Transaction>, List<Auction>, List<Book>

administrationSystem