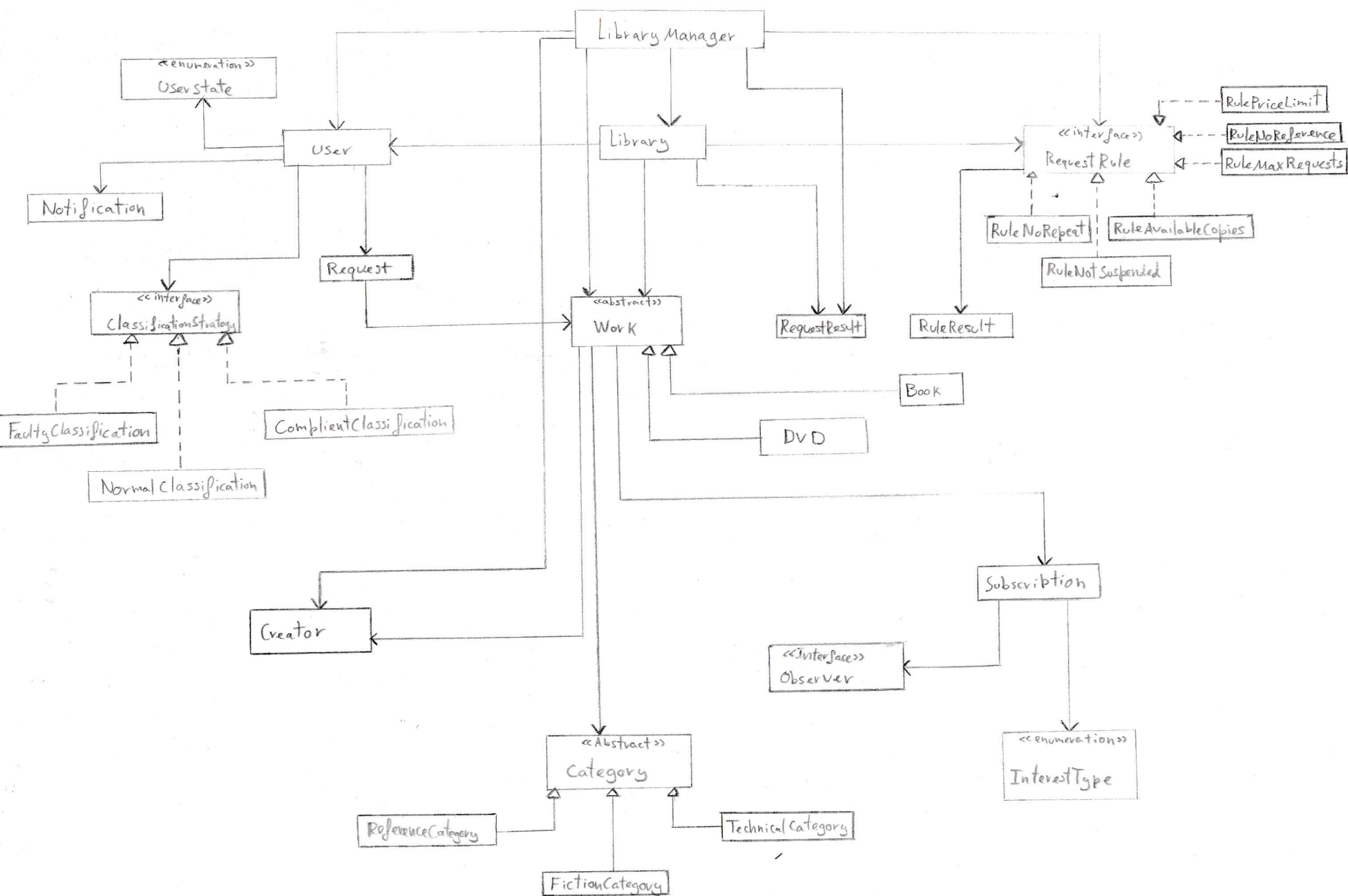
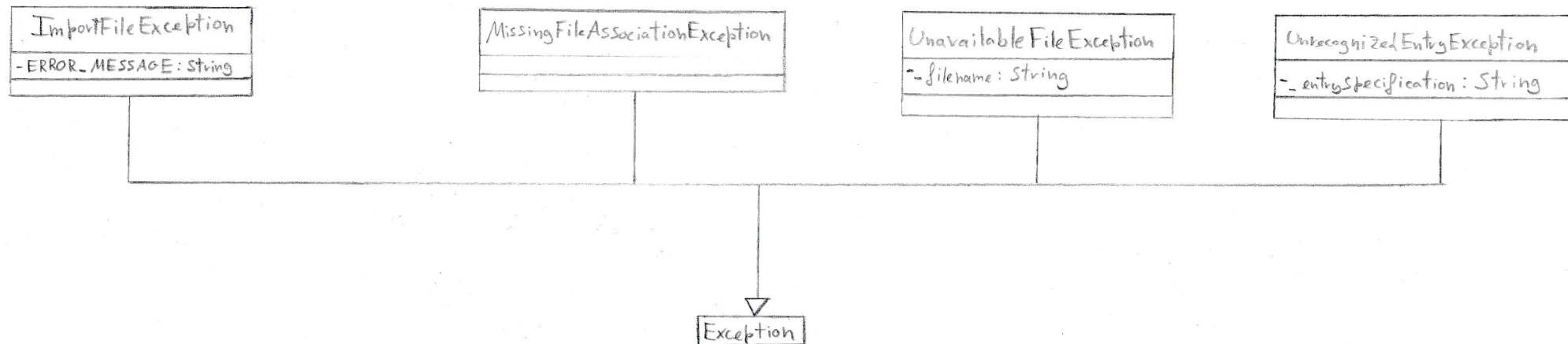


Declaro por minha honra que este diagrama foi realizado apenas pelos elementos que constituem o grupo de projeto. Bruno Silva ist1113985





Library Manager

```

- library: Library
- defaultRules: List<RequestRule>
- associatedFile: String

+ save(): Void
+ saveAs(fileName: String): Void
+ load(fileName: String): Void
+ importFile(fileName: String): Void
+ registerUser(name: String, email: String): Void
+ registerWork(work: Work): Void
+ requestWork(userId: int, workId: int, askAvailabilityNotification: boolean): RequestResult
+ returnWork(requestId: int): Void
+ changeInventory(workId: int, delta: int): Void
+ search(query: String): List<Work>
  
```

Library

```

- currentDate: int
- finePerDay: int
- users: Map<int, User>
- works: Map<int, Work>
- creators: Map<String, Creator>
- requests: List<Request>
- rules: List<RequestRule>
- associatedFileName: String

+ importFile(fileName: String): Void
+ saveAs(fileName: String): Void
+ load(fileName: String): Void
+ advanceTime(days: int): Void
+ registerUser(name: String, email: String): Void
+ registerWork(work: Work): Void
+ requestWork(userId: int, workId: int, askAvailabilityNotification: boolean): RequestResult
+ returnWork(requestId: int): Void
+ changeInventory(workId: int, delta: int): Void
+ search(query: String): List<Work>
+ applyFines(): Void
+ removeCreatorIfEmpty(c: Creator): Void
+ removeWorkIfNoCopies(workId: int): Void
+ cancelNotifications(work: Work): Void
+ registerInterest(userId: int, workId: int, interest: InterestType): Void
+ restoreIdAfterLoad(): Void
  
```

User

```

- id: int
- lastId: int
- name: String
- email: String
- state: UserState
- classificationStrategy: ClassificationStrategy
- activeRequests: List<Request>
- historyRequests: List<Request>
- pendingFines: int
- notificationsQueue: Queue<Notification>

+ addRequest(request: Request): Void
+ closeRequest(request: Request, returnDate: int): Void
+ payFine(amount: int): Void
+ viewNotifications(): Void
+ evaluateClassification(): Void
# allocateId(): void
+ restoreLastId(value: int): Void
+ update(msg: String, work: Work)
  
```

RequestRule

```

- ruleNumber: int

+ validate(request: Request, user: User, work: Work): RuleResult
  
```

RequestResult

```

- success: boolean
- msg: String
- createdRequestId: int
  
```

Request

```

- id: int
- userId: int
- workId: int
- startDate: int
- dueDate: int
- returnedDate: int
- finePaid: int

+ markReturned(returnDate: int): Void
+ isOverDue(referenceDate: int): boolean
+ daysOverDue(referenceDate: int): int
  
```

«Interface» ClassificationStrategy
+name(): String +maxConcurrentLoan(): int +loanPeriodDays(workCopies: int): int +evaluate(user: User): Void

Notification
-msg: String -workId: int -date: int

RuleResult
-valid: boolean -msg: String

«Enumeration» UserState
ACTIVE SUSPENDED

Subscription
-observer: Observer -interest: InterestType -timestamp: int

«Abstract» Work
-id: int -lastId: int -title: String -price: int -copies: int -category: Category -subscriptions: List<Subscription> -createdBy: List<Creator> -createdDate: int
+incrementCopies(amount: int): Void +decrementCopies(amount: int): Void +notifyBorrowed(): Void +notifyAvailable(): Void +allocateId(): int +restoreLastId(value: int): Void

Book
-ISBN: String -authors: List<Creator>

DVD
-IGACRegistration: String -director: Creator

«Abstract» Category
-name: String

«Interface» Observer
+update(msg: String, work: Work)

«Enumeration» InterestType
ON-BORROW ON-AVAILABLE

Category
-name: String -works: List<Work>
+addWork(work: Work): Void +removeWork(work: Work): Void +hasNoWorks(): boolean