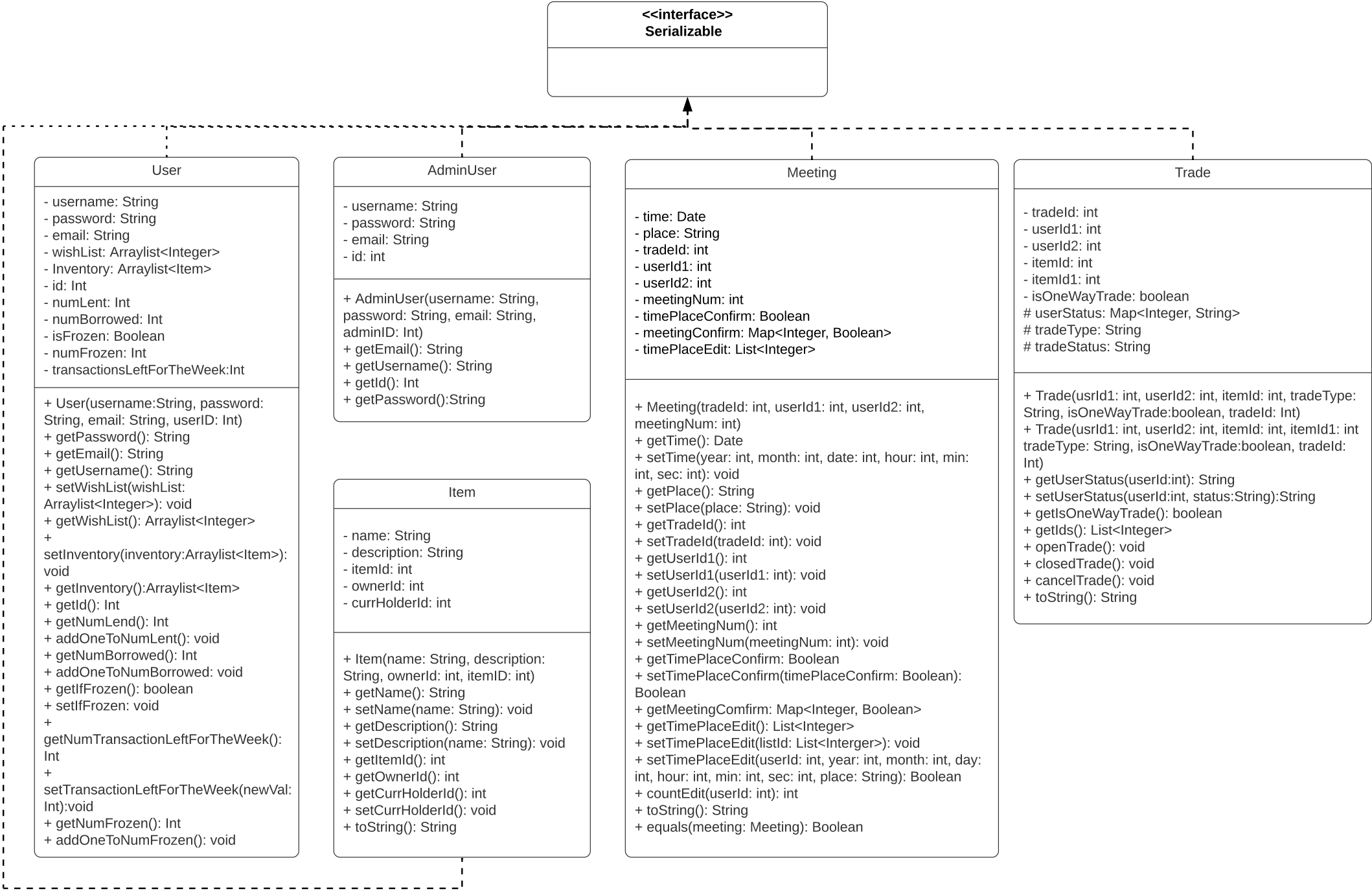
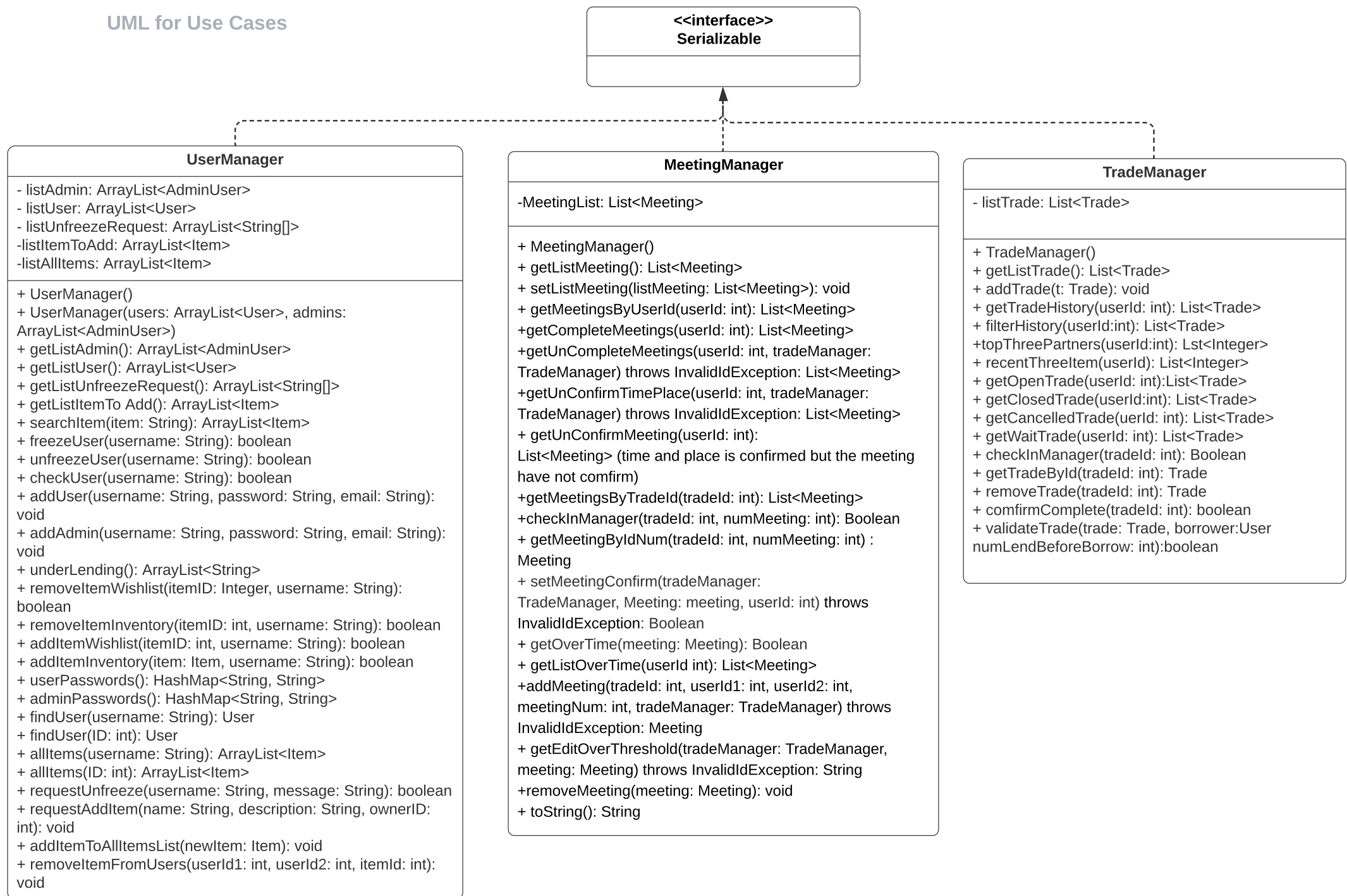


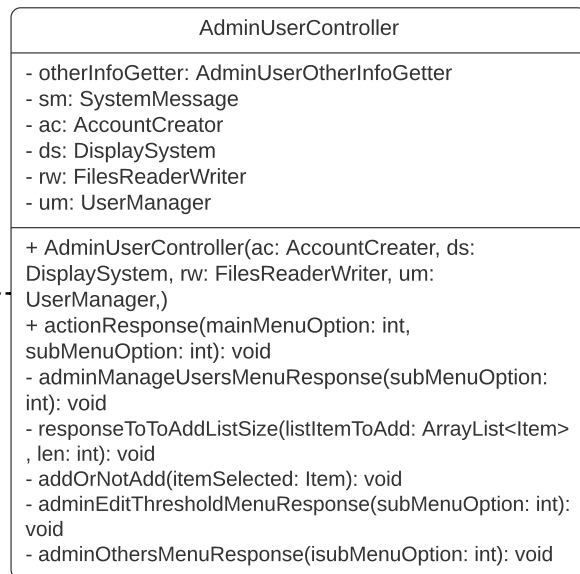
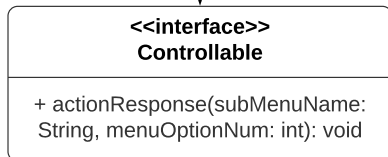
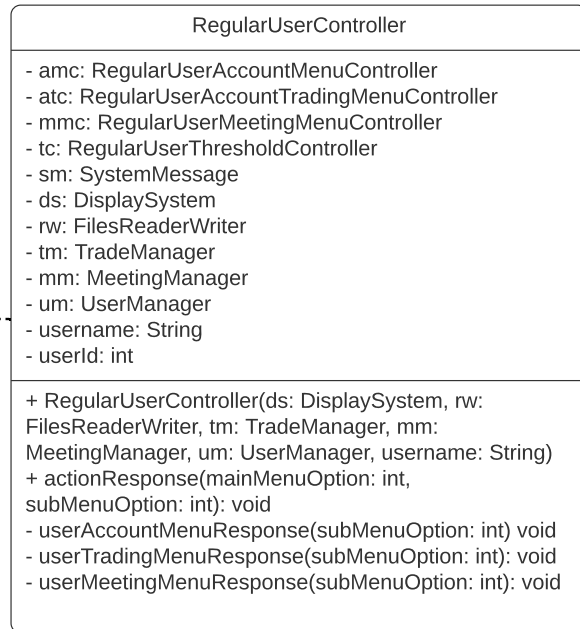
UML for Entities



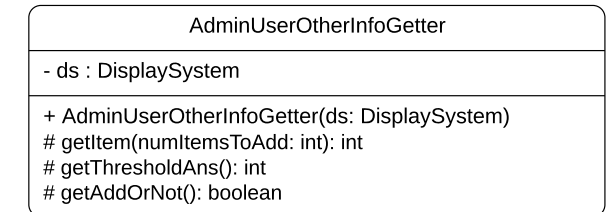
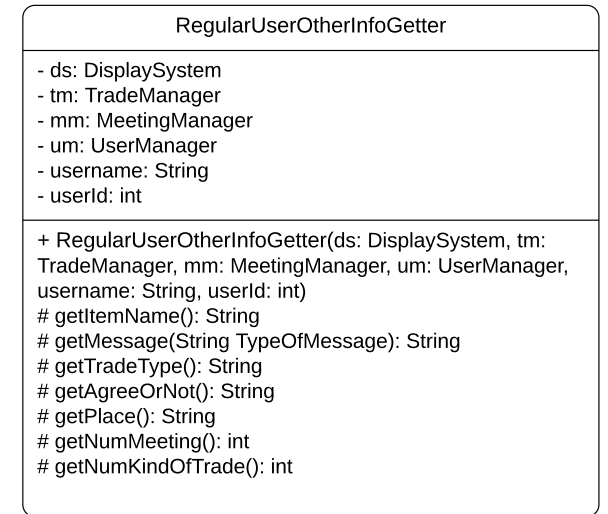
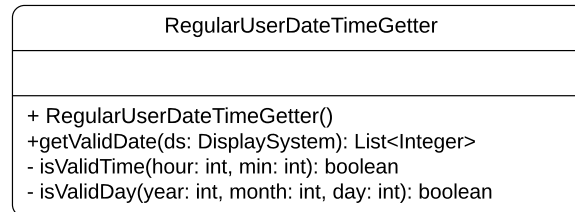
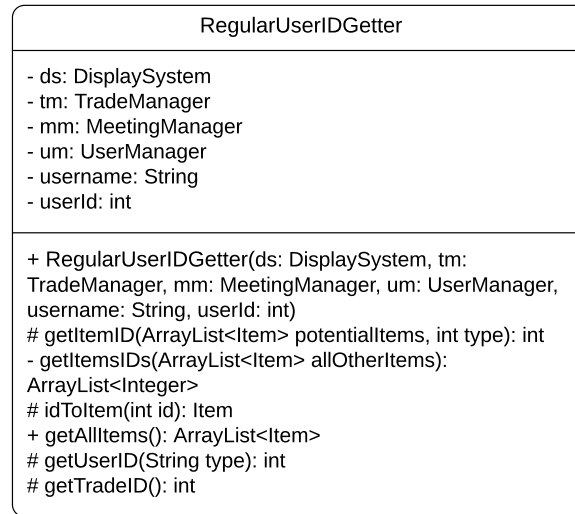
UML for Use Cases



UML for Main Controller Classes



UML for Controllers' Getter Classes



UML for RegularUserController's Sub-controller Classes

RegularUserTradingMenuController

- tc: RegularUserThresholdController
- otherInfoGetter: RegularUserOtherInfoGetter
- idGetter: RegularUserIDGetter
- sm: SystemMessage
- ds: DisplaySystem
- tm: TradeManager
- mm: MeetingManager
- um: UserManager
- username: String
- userId: int

```
+ RegularUserTradingMenuController( ds: DisplaySystem,tm:
TradeManager, mm: MeetingManager, um: UserManager, username:
String, userId: int)
# seeTopThreePartners(): void
# confirmTradeComplete(): void
# viewOpenTrades(openTrade: List<Trade>): void
# respondToTradeRequests(thisUser: User): void
- isAgreedBefore(agreedBefore: boolean, agree: boolean, b: boolean):
void
- respondResult(tradeID: int, trade: Trade, itemid22: int, agreedBefore:
boolean, userId11: int, userId22: int, itemid11: int): void
- respondAgree(tradeID: int, trade: Trade, itemid22: int, userId11: int,
userId22: int, itemid11: int): void
# requestTrade(thisUser: User): void
- determineTradeID(): int
- getValidationForItems(numKindOfTrade: int, itemld2: int, userId1: int,
userId2: int, itemld: int): boolean
- getTrade(numKindOfTrade: int, itemld2: int, userId1: int, userId2: int,
itemld: int, tradeID: int, tradeType: String): Trade
- requestResult(thisUser: User, ok: boolean, trade: Trade, userId1: int):
void
- requestFail(thisUser: User): void
- requestSuccess(thisUser: User, trade: Trade): void
- validateItems(borrower: int, lender: int, itemld: int): boolean
- validateItems(borrower1Lender2: int, borrower2lender1: int, itemld1: int,
itemld2: int): boolean
```

RegularUserMeetingMenuController

- sm: SystemMessage
- idGetter: RegularUserIDGetter
- dateTimeGetter: RegularUserDateTimeGetter
- otherInfoGetter: RegularUserOtherInfoGetter
- ds: DisplaySystem
- tm: TradeManager
- mm: MeetingManager
- um: UserManager
- username: String
- userId: int

```
+ RegularUserMeetingMenuController(ds: DisplaySystem, tm:
TradeManager, mm: MeetingManager, um: UserManager,
username: String, userId: int)
# seeMeetingsToBeConfirmed(unConfirmMeeting: List<Meeting>,
s: String): void
# confirmMeetingTookPlace(): void
# confirmMeetingTandP(): void
# EditMeetingTandP(): void
# unconfirmedTandPMeetings(): void
- getMeeting(): Meeting
```

RegularUserThresholdController

- ds: DisplaySystem
- tm: TradeManager
- mm: MeetingManager
- um: UserManager
- username: String
- userId: int
- thresholdReassessed: boolean

```
+ public RegularUserThresholdController(ds: DisplaySystem,tm:
TradeManager, mm: MeetingManager, um: UserManager,
username: String, userId: int)
# reassessNumTransactionsLeftForTheWeek(thisUser: User): void
# changeNumTradesLeftForTheWeek(thisUser: User): void
# isFirstDayOfTheWeek(): boolean
# freezeUserOrNot(User thisUser): boolean
- numUncompletedTransactions(): int
```

RegularUserAccountMenuController

- sm: SystemMessage
- idGetter: RegularUserIDGetter
- otherInfoGetter: RegularUserOtherInfoGetter
- ds: DisplaySystem
- tm: TradeManager
- mm: MeetingManager
- um: UserManager
- username: String
- userId: int

```
+ RegularUserAccountMenuController(ds: DisplaySystem, tm:
TradeManager, mm: MeetingManager, um: UserManager,
username: String, userId: int)
# viewWishListInventory(): void
# addToWishList(allOtherItems: ArrayList<Item>): void
# requestAddItem(): void
# RequestToUnfreeze(): void
# seeMostRecentThreeItems(): void
# removeFromInventory(): void
# removeFromWishList(allOtherItems: ArrayList<Item>): void
# searchItem(): void
# browseBooks(allOtherItems: ArrayList<Item>): void
```

AccountCreator

- um: UserManager
- ds: DisplaySystem
- fr: FileReaderWriter

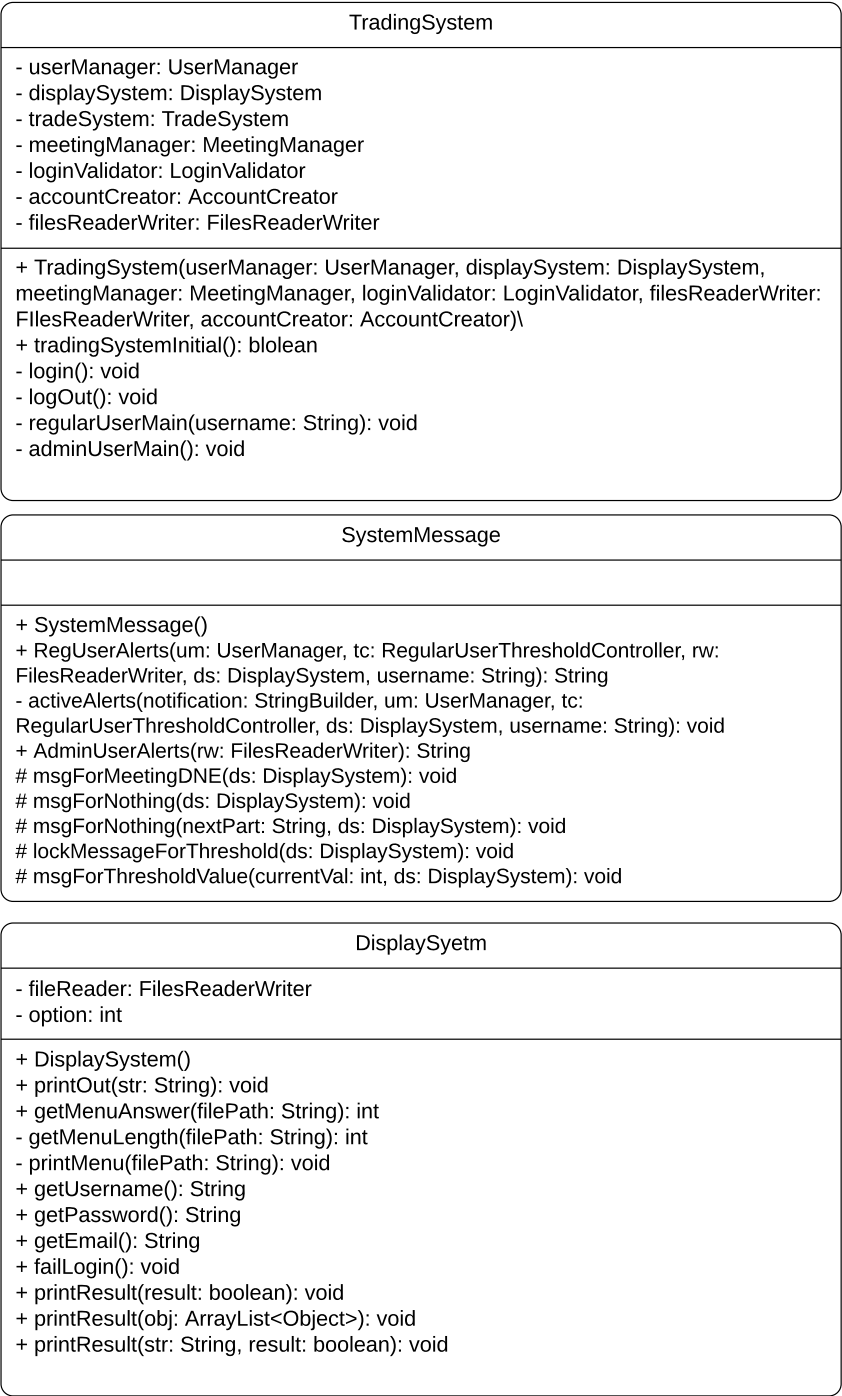
```
+ AccountCreator(um: UserManager, ds: DisplaySystem, fr:
FileReaderWriter)
+ CreateAccount(type: String): boolean
```

LoginValidator

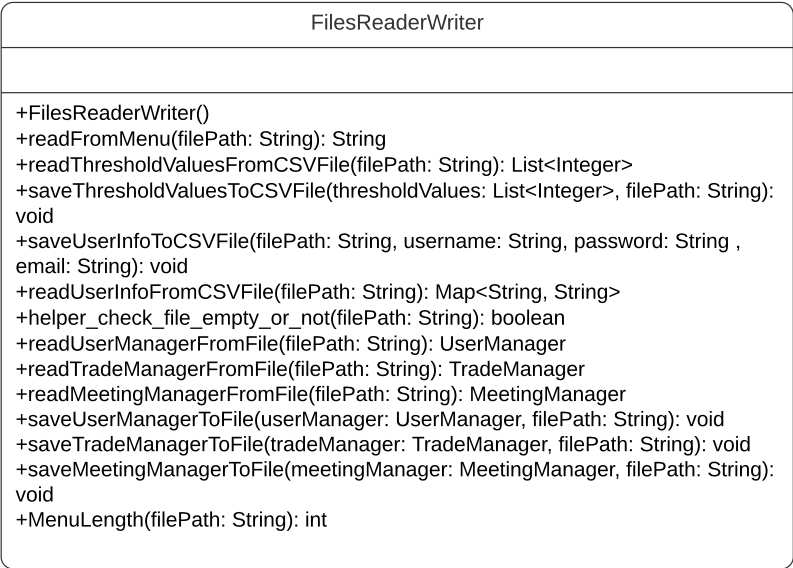
- userLoginInfo: Map<String, String>
- adminUserLoginInfo: Map<String, String>

```
+ LoginValidator(userLoginInfo: Map<String, String>,
adminUserLoginInfo: Map<String, String>)
+ VerifyLogin(username: String, password: String): String
```

UML for Presenter



UML for Gateway



UML for DemoMain

