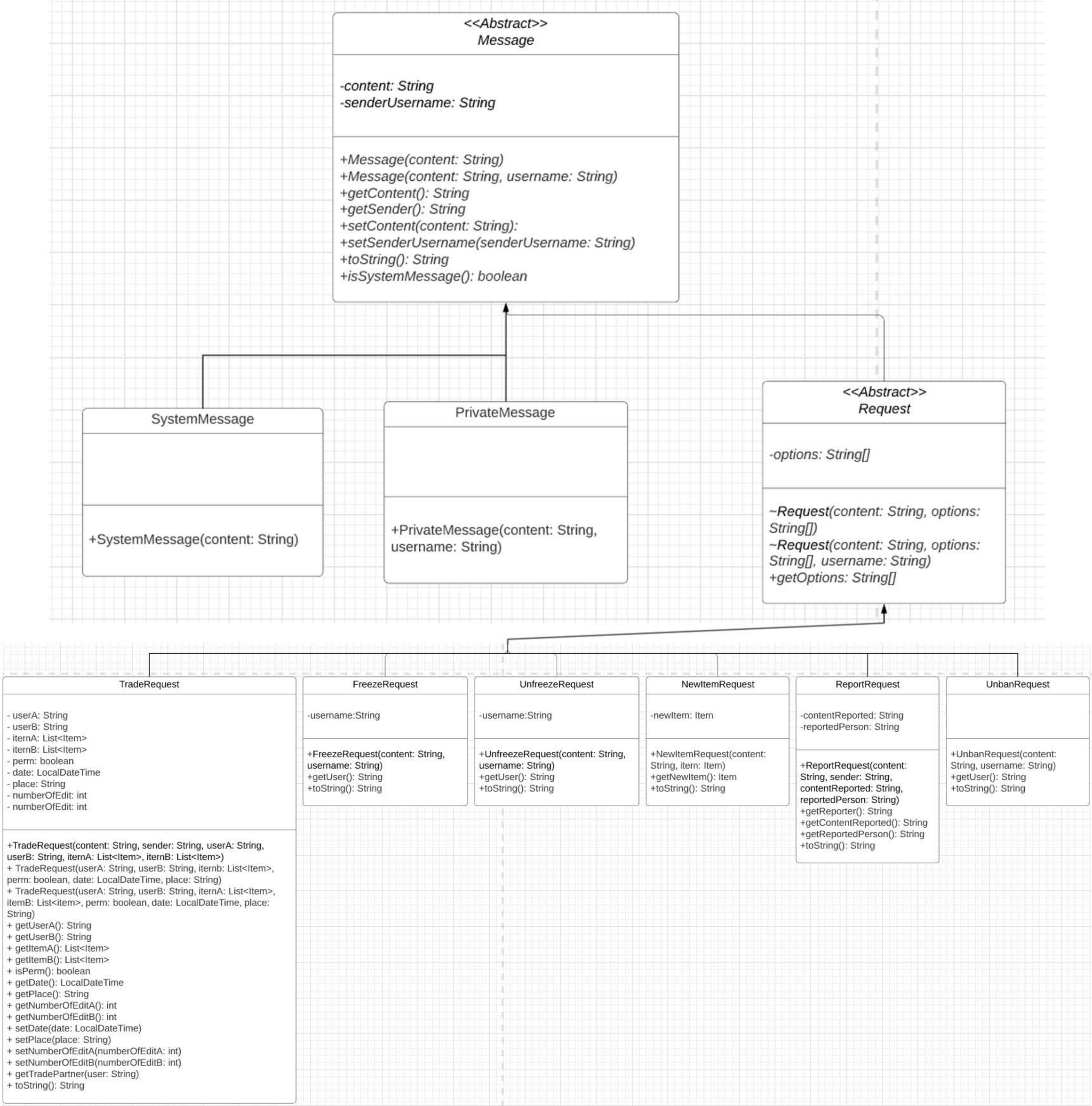
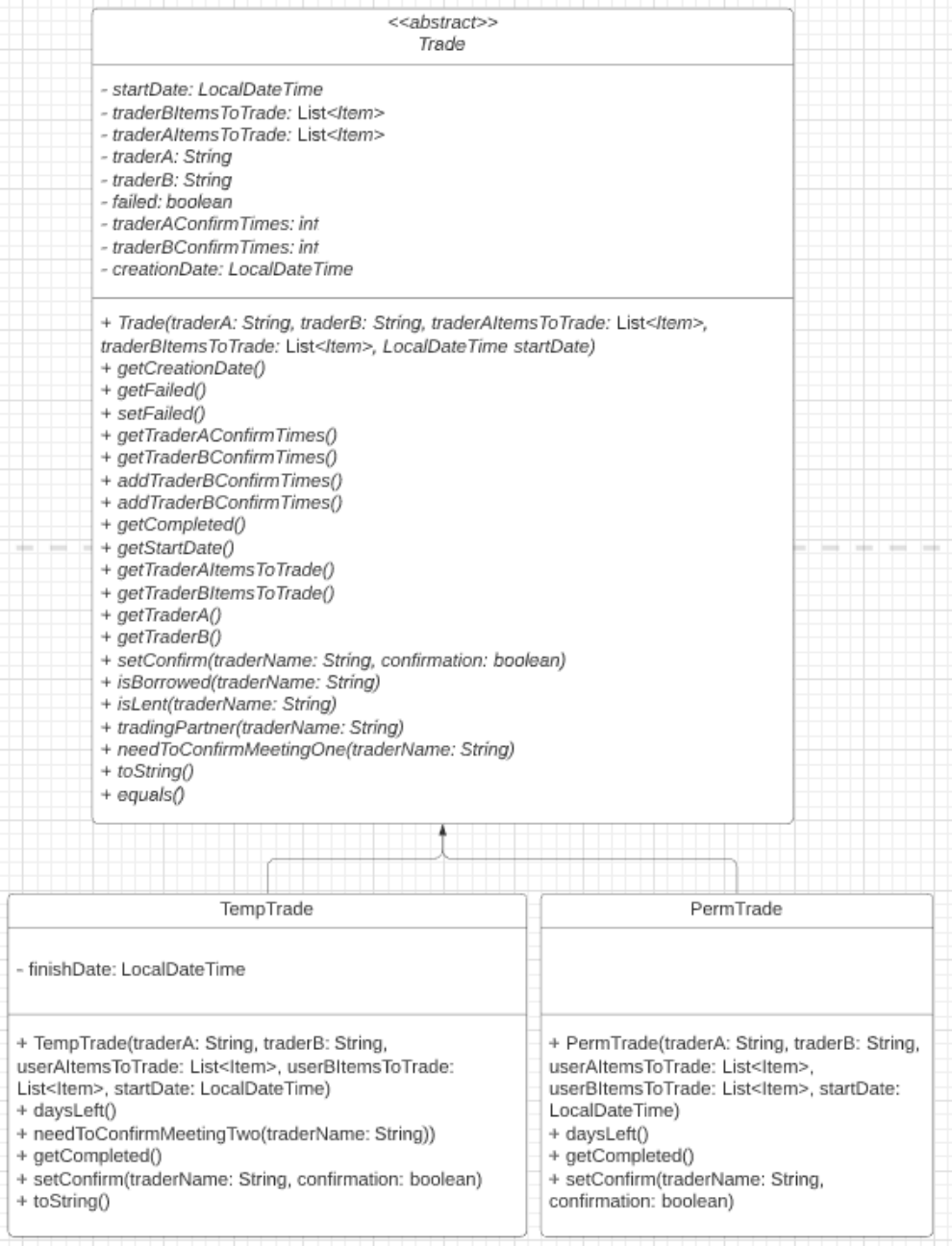
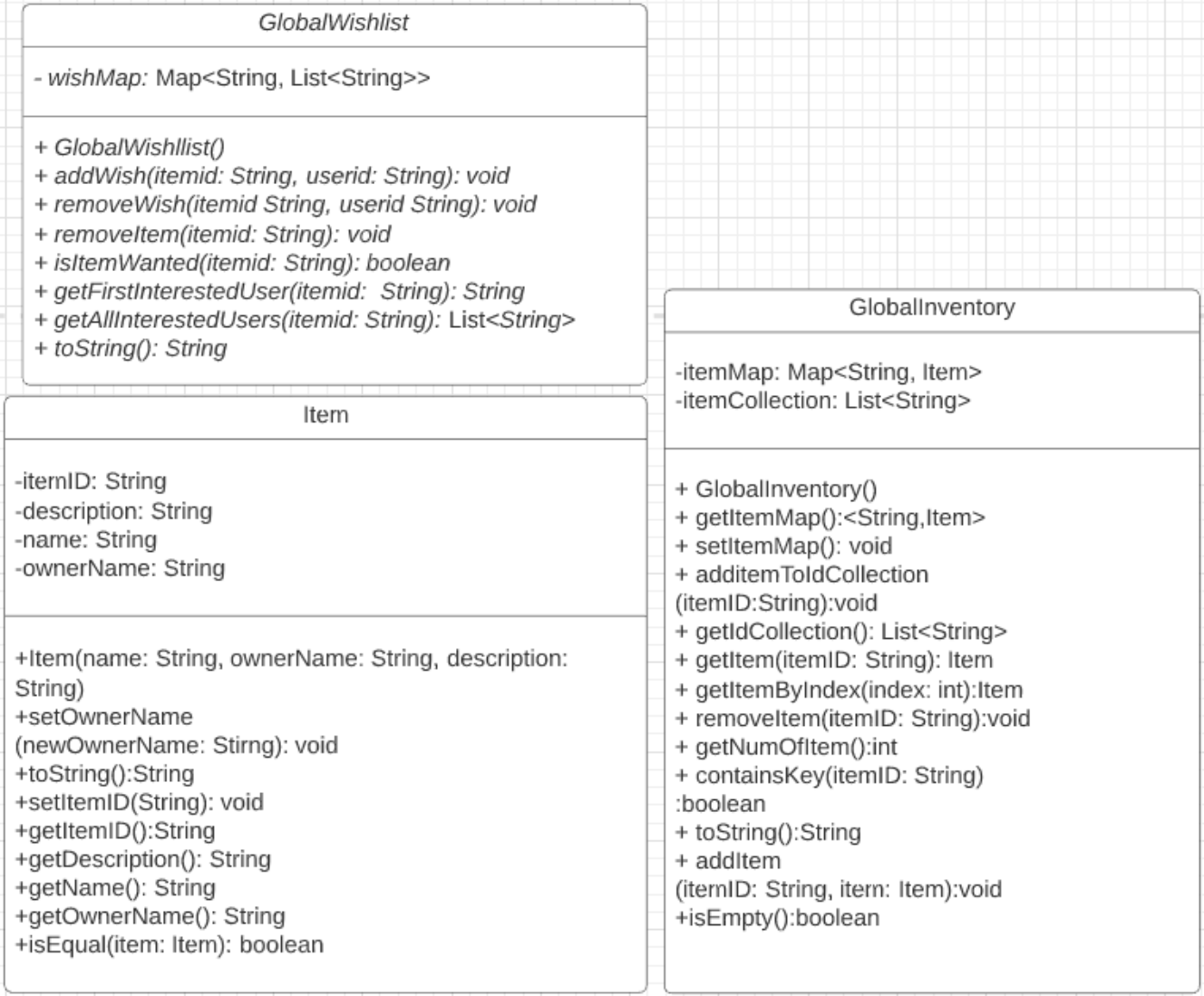
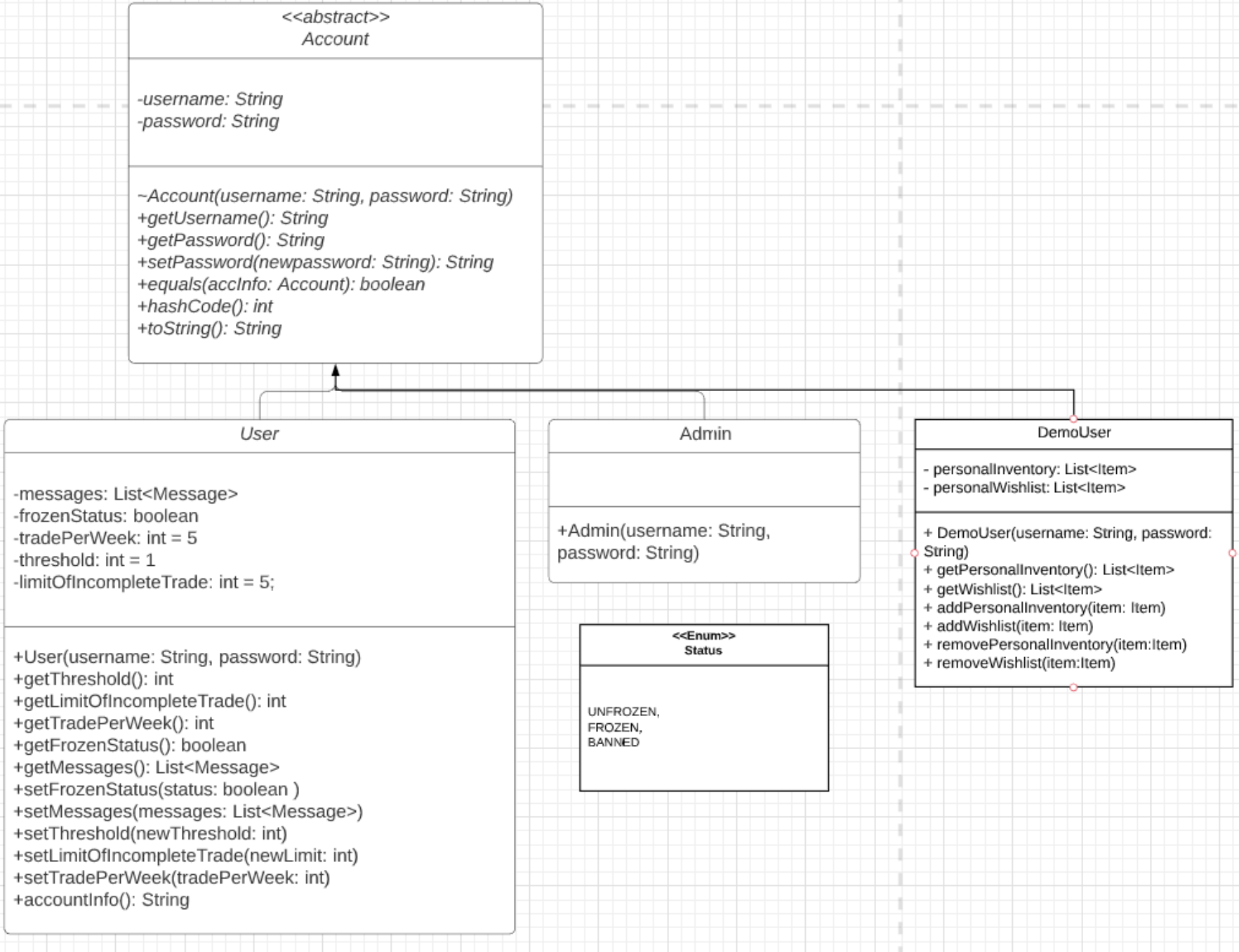
entities


<<Abstract>>
Message
-content: String
-senderUsername: String
+Message(content: String)
+Message(content: String, username: String)
+getContent(): String
+getSender(): String
+setContent(content: String):
+setSenderUsername(senderUsername: String)
+toString(): String
+isSystemMessage(): boolean
SystemMessage
+SystemMessage(content: String)
PrivateMessage
+PrivateMessage(content: String, username: String)
<<Abstract>>
Request
-options: String[]
~Request(content: String, options: String[])
~Request(content: String, options: String[], username: String)
+getOptions: String[]
TradeRequest
- userA: String
- userB: String
- iternA: List<Item>
- iternB: List<Item>
- perm: boolean
- date: LocalDateTime
- place: String
- numberOfEdit: int
- numberOfEdit: int
+TradeRequest(content: String, sender: String, userA: String, userB: String, iternA: List<Item>, iternB: List<Item>)
+ TradeRequest(userA: String, userB: String, iternb: List<Item>, perm: boolean, date: LocalDateTime, place: String)
+ TradeRequest(userA: String, userB: String, iternA: List<Item>, iternB: List<item>, perm: boolean, date: LocalDateTime, place: String)
+ getUserA(): String
+ getUserB(): String
+ getItemA(): List<Item>
+ getItemB(): List<Item>
+ isPerm(): boolean
+ getDate(): LocalDateTime
+ getPlace(): String
+ getNumberOfEditA(): int
+ getNumberOfEditB(): int
+ setDate(date: LocalDateTime)
+ setPlace(place: String)
+ setNumberOfEditA(numberOfEditA: int)
+ setNumberOfEditB(numberOfEditB: int)
+ getTradePartner(user: String)
+ toString(): String
FreezeRequest
-username:String
+FreezeRequest(content: String, username: String)
+getUser(): String
+toString(): String
UnfreezeRequest
-username:String
+UnfreezeRequest(content: String, username: String)
+getUser(): String
+toString(): String
NewItemRequest
-newItem: Item
+NewItemRequest(content: String, item: Item)
+getNewItem(): Item
+toString(): String
ReportRequest
-contentReported: String
-reportedPerson: String
+ReportRequest(content: String, sender: String, contentReported: String, reportedPerson: String)
+getReporter(): String
+getContentReported(): String
+getReportedPerson(): String
+toString(): String
UnbanRequest
+UnbanRequest(content: String, username: String)
+getUser(): String
+toString(): String


<<abstract>>
Trade
- startDate: LocalDateTime
- traderBItemsToTrade: List<Item>
- traderAItemsToTrade: List<Item>
- traderA: String
- traderB: String
- failed: boolean
- traderAConfirmTimes: int
- traderBConfirmTimes: int
- creationDate: LocalDateTime
+ Trade(traderA: String, traderB: String, traderAItemsToTrade: List<Item>, traderBItemsToTrade: List<Item>, LocalDateTime startDate)
+ getCreationDate()
+ getFailed()
+ setFailed()
+ getTraderAConfirmTimes()
+ getTraderBConfirmTimes()
+ addTraderBConfirmTimes()
+ addTraderBConfirmTimes()
+ getCompleted()
+ getStartDate()
+ getTraderAItemsToTrade()
+ getTraderBItemsToTrade()
+ getTraderA()
+ getTraderB()
+ setConfirm(traderName: String, confirmation: boolean)
+ isBorrowed(traderName: String)
+ isLent(traderName: String)
+ tradingPartner(traderName: String)
+ needToConfirmMeetingOne(traderName: String)
+ toString()
+ equals()
TempTrade
- finishDate: LocalDateTime
+ TempTrade(traderA: String, traderB: String, userAItemsToTrade: List<Item>, userBItemsToTrade: List<Item>, startDate: LocalDateTime)
+ daysLeft()
+ needToConfirmMeetingTwo(traderName: String))
+ getCompleted()
+ setConfirm(traderName: String, confirmation: boolean)
+ toString()
PermTrade
+ PermTrade(traderA: String, traderB: String, userAItemsToTrade: List<Item>, userBItemsToTrade: List<Item>, startDate: LocalDateTime)
+ daysLeft()
+ getCompleted()
+ setConfirm(traderName: String, confirmation: boolean)


GlobalWishlist
- wishMap: Map<String, List<String>>
+ GlobalWishllist()
+ addWish(itemid: String, userid: String): void
+ removeWish(itemid String, userid String): void
+ removeItem(itemid: String): void
+ isItemWanted(itemid: String): boolean
+ getFirstInterestedUser(itemid: String): String
+ getAllInterestedUsers(itemid: String): List<String>
+ toString(): String
GlobalInventory
-itemMap: Map<String, Item>
-itemCollection: List<String>
+ GlobalInventory()
+ getItemMap():<String,Item>
+ setItemMap(): void
+ additemToIdCollection
(itemID:String):void
+ getIdCollection(): List<String>
+ getItem(itemID: String): Item
+ getItemByIndex(index: int):Item
+ removeItem(itemID: String):void
+ getNumOfItem():int
+ containsKey(itemID: String)
:boolean
+ toString():String
+ addItem
(itemID: String, item: Item):void
+isEmpty():boolean
Item
-itemID: String
-description: String
-name: String
-ownerName: String
+Item(name: String, ownerName: String, description:
String)
+setOwnerName
(newOwnerName: Stirng): void
+toString():String
+setItemID(String): void
+getItemID():String
+getDescription(): String
+getName(): String
+getOwnerName(): String
+isEqual(item: Item): boolean


<<abstract>>
Account
-username: String
-password: String
~Account(username: String, password: String)
+getUsername(): String
+getPassword(): String
+setPassword(newpassword: String): String
+equals(accInfo: Account): boolean
+hashCode(): int
+toString(): String
User
-messages: List<Message>
-frozenStatus: boolean
-tradePerWeek: int = 5
-threshold: int = 1
-limitOfIncompleteTrade: int = 5;
+User(username: String, password: String)
+getThreshold(): int
+getLimitOfIncompleteTrade(): int
+getTradePerWeek(): int
+getFrozenStatus(): boolean
+getMessages(): List<Message>
+setFrozenStatus(status: boolean)
+setMessages(messages: List<Message>)
+setThreshold(newThreshold: int)
+setLimitOfIncompleteTrade(newLimit: int)
+setTradePerWeek(tradePerWeek: int)
+accountInfo(): String
Admin
+Admin(username: String, password: String)
<<Enum>>
Status
UNFROZEN,
FROZEN,
BANNED
DemoUser
- personalInventory: List<Item>
- personalWishlist: List<Item>
+ DemoUser(username: String, password: String)
+ getPersonalInventory(): List<Item>
+ getWishlist(): List<Item>
+ addPersonalInventory(item: Item)
+ addWishlist(item: Item)
+ removePersonalInventory(item:Item)
+ removeWishlist(item:Item)

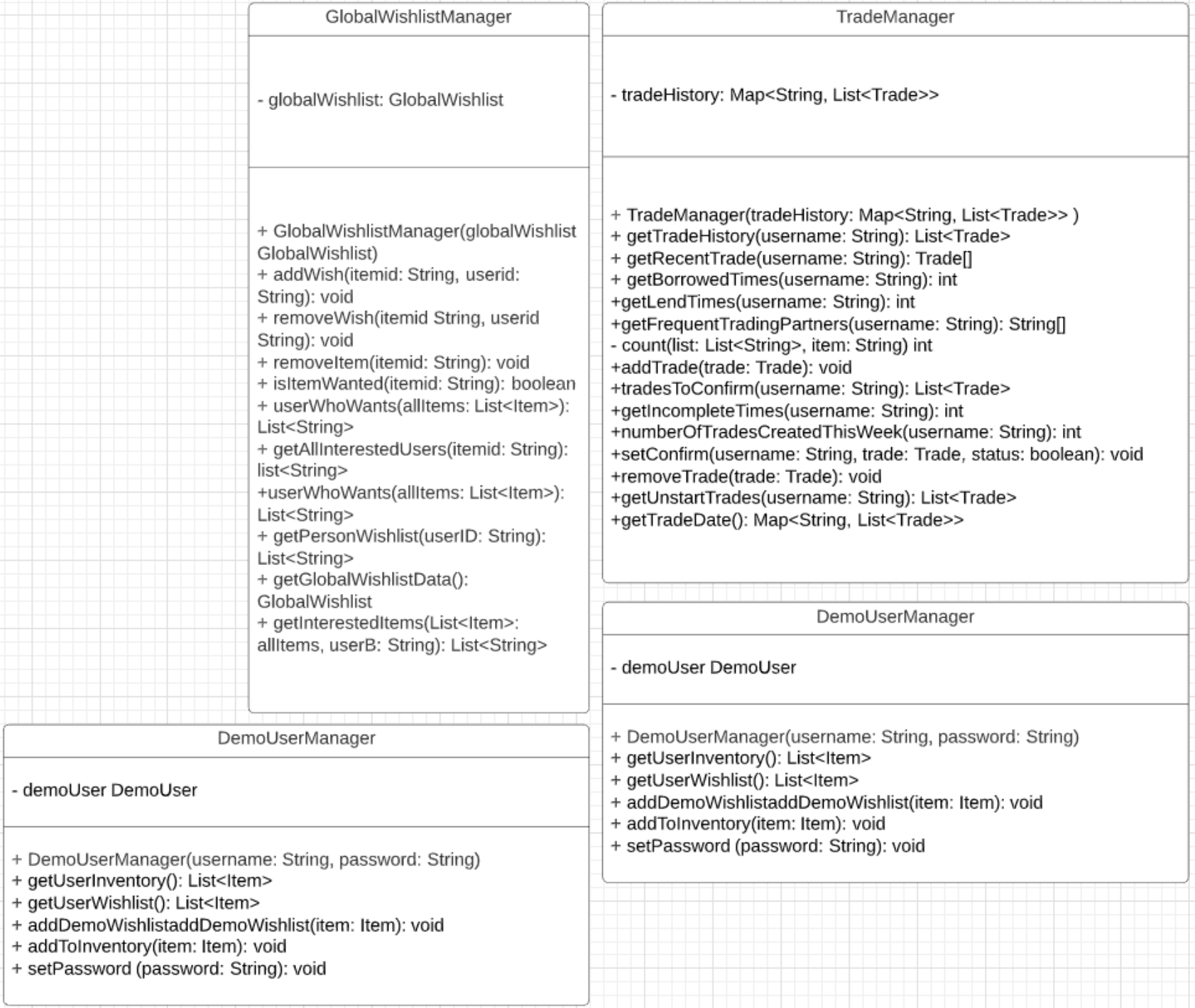
use_cases

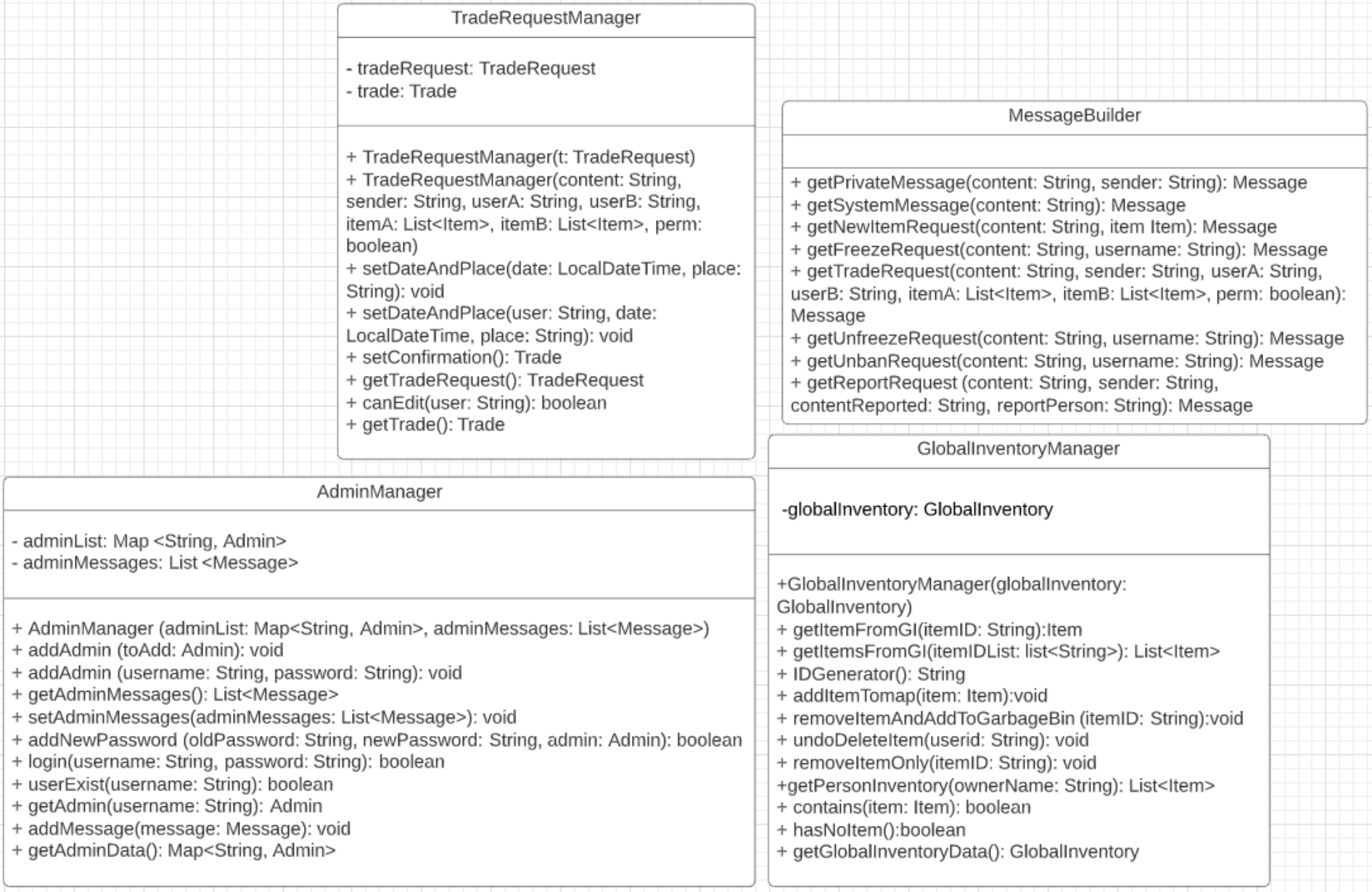
UserManager

- allUsers: Map<String, User>

+ UserManager(allUsers: Map<String, User>)
+ login(username: String, password: String) : boolean
+ createNewUser(username: String, password: String) : boolean
+ changePassword(username: String, password: String) : void
+ getUserData(): Map<String, User>
+ isValidUser(username: String) : boolean
+ toString() : String
+ getCanTrade(user: String, borrowedTimes: int, lendTimes: int, numIncomplete: int, numTradesMadeThisWeek: int) : boolean
+ getCanTradesIgnoreBorrowsLoans(username: String, numIncomplete: int, numTradesMadeThisWeek: int) : boolean
+ getUserMessages(username: String) : List<Message>
+ setUserMessages(username: String, message: List<Message>) : void
+ getUserInfo(username: String)
+ getUserInfo(username: String) : String
+ getUserFrozenStatus(username: String) : boolean
+ getUserIsBanned(username: String): boolean
+ getUserThreshold(username: String) : int
+ addUserMessage(username: String, message: Message) : void
+ setLimitOfIncompleteTrades(newLimit: int) : void
+ setWeeklyTradesForOneUser(username: String, newTradesPerWeek: int) : void
+ removeUser(username: String) : void
+ setLimitOfIncompleteTrades(newLimit: int) : void
+ setLimitOfIncompleteTradesForOneUser(username: String, newLimit: int) : void
+ setNewThreshold(newThreshold: int) : void
+ setNewThresholdForOneUser(username: String, newThreshold: int) : void


GlobalWishlistGateway
+ wishlist: GlobalWishlist
+ GlobalWishlistGateway(filepath: String)
+ writeToFile (filepath: String, wishlistItems2: GlobalWishList)
+ readFromFile (filepath: String): GlobalWishlist
+ getWishlistItems(): GlobalWishList
AdminAccountGateways
-filePath: String
-adminMap: Map<String, Admin>
+AdminAccountGateways (filePath:String)
+readFromFile(): void
+saveToFile(adminMap: map<String,Admin>): void
+getAdminMap(): map<String, Admin>
+ beginAdminMap(): void
AdminMessageGateway
+ messages: List<Message>
+ AdminMessageGateway(filepath: String)
+ writeToFile (filepath: String, adminMessages: List <Message>): void
+ readFromFile (filepath: String): List<Message>
+ getMessages(): List<Message>
GlobalInventoryGateways
- filePath: String
- globalInventoryManager: GlobalInventoryManager
- globalInventory: GlobalInventory
+GlobalInventoryGateways(filePath:String)
+writeToFile(globalInventory: GlobalInventory): void
+getGlobalInventory(): GlobalInventory
UserTradesGateway
+ userTrades: Map<String, List <Trade>>
+ UserTradesGateway(filepath: String):
+ writeToFile (filepath: String, userTrades: Map <String, List <Trade>>): void
+ readFromFile (filepath: String): Map<String, List<Trade>>
+ getUserTrades(): map<String, list<Trade>>
UserGateway
- mapOfUsers: Map<String, User>
+ UserGateway(filepath: String)
+ readFromFile(filepath: String): Map<String, User>
+ writeToFile(filepath: String, userObjects: Map<String, User>): void
+ getMapOfUsers(): Map<String, User>

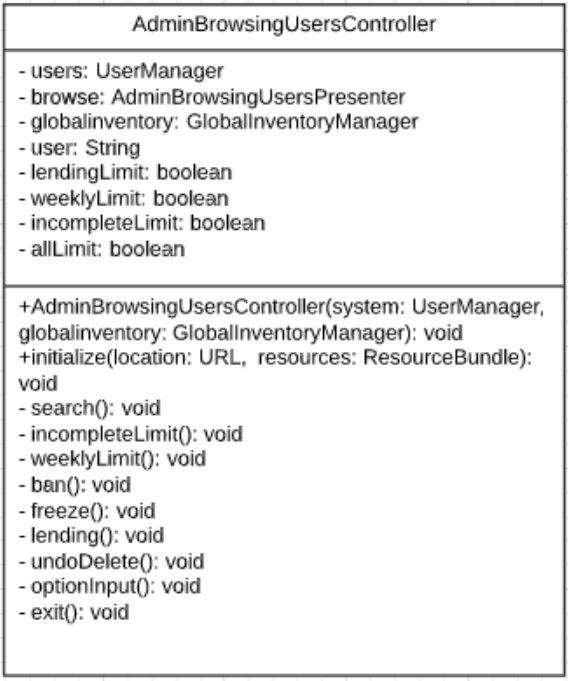

GlobalWishlistManager
- globalWishlist: GlobalWishlist
+ GlobalWishlistManager(globalWishlist GlobalWishlist)
+ addWish(itemid: String, userid: String): void
+ removeWish(itemid String, userid String): void
+ removeItem(itemid: String): void
+ isItemWanted(itemid: String): boolean
+ userWhoWants(allItems: List<Item>): List<String>
+ getAllInterestedUsers(itemid: String): list<String>
+userWhoWants(allItems: List<Item>): List<String>
+ getPersonWishlist(userID: String): List<String>
+ getGlobalWishlistData(): GlobalWishlist
+ getInterestedItems(List<Item>: allItems, userB: String): List<String>
TradeManager
- tradeHistory: Map<String, List<Trade>>
+ TradeManager(tradeHistory: Map<String, List<Trade>>)
+ getTradeHistory(username: String): List<Trade>
+ getRecentTrade(username: String): Trade[]
+ getBorrowedTimes(username: String): int
+getLendTimes(username: String): int
+getFrequentTradingPartners(username: String): String[]
- count(list: List<String>, item: String) int
+addTrade(trade: Trade): void
+tradesToConfirm(username: String): List<Trade>
+getIncompleteTimes(username: String): int
+numberOfTradesCreatedThisWeek(username: String): int
+setConfirm(username: String, trade: Trade, status: boolean): void
+removeTrade(trade: Trade): void
+getUnstartTrades(username: String): List<Trade>
+getTradeDate(): Map<String, List<Trade>>
DemoUserManager
- demoUser DemoUser
+ DemoUserManager(username: String, password: String)
+ getUserInventory(): List<Item>
+ getUserWishlist(): List<Item>
+ addDemoWishlistaddDemoWishlist(item: Item): void
+ addToInventory(item: Item): void
+ setPassword (password: String): void
DemoUserManager
- demoUser DemoUser
+ DemoUserManager(username: String, password: String)
+ getUserInventory(): List<Item>
+ getUserWishlist(): List<Item>
+ addDemoWishlistaddDemoWishlist(item: Item): void
+ addToInventory(item: Item): void
+ setPassword (password: String): void

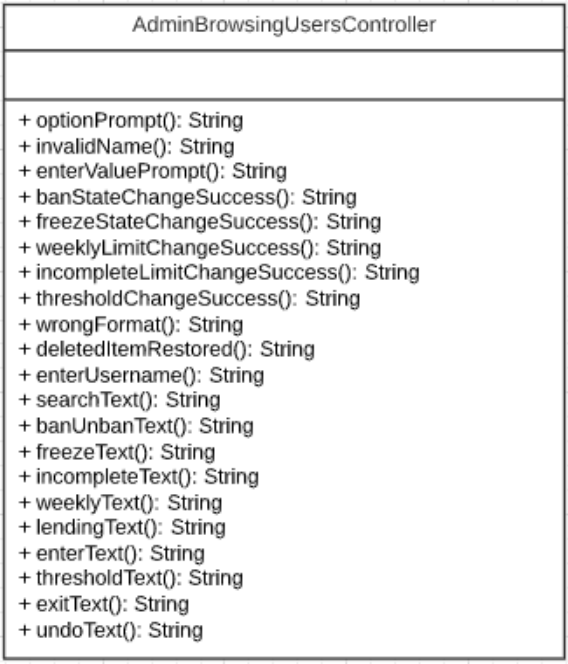

TradeRequestManager
- tradeRequest: TradeRequest
- trade: Trade
+ TradeRequestManager(t: TradeRequest)
+ TradeRequestManager(content: String, sender: String, userA: String, userB: String, itemA: List<Item>, itemB: List<Item>, perm: boolean)
+ setDateAndPlace(date: LocalDateTime, place: String): void
+ setDateAndPlace(user: String, date: LocalDateTime, place: String): void
+ setConfirmation(): Trade
+ getTradeRequest(): TradeRequest
+ canEdit(user: String): boolean
+ getTrade(): Trade
MessageBuilder
+ getPrivateMessage(content: String, sender: String): Message
+ getSystemMessage(content: String): Message
+ getNewItemRequest(content: String, item Item): Message
+ getFreezeRequest(content: String, username: String): Message
+ getTradeRequest(content: String, sender: String, userA: String, userB: String, itemA: List<Item>, itemB: List<Item>, perm: boolean): Message
+ getUnfreezeRequest(content: String, username: String): Message
+ getUnbanRequest(content: String, username: String): Message
+ getReportRequest (content: String, sender: String, contentReported: String, reportPerson: String): Message
AdminManager
- adminList: Map <String, Admin>
- adminMessages: List <Message>
+ AdminManager (adminList: Map<String, Admin>, adminMessages: List<Message>)
+ addAdmin (toAdd: Admin): void
+ addAdmin (username: String, password: String): void
+ getAdminMessages(): List<Message>
+ setAdminMessages(adminMessages: List<Message>): void
+ addNewPassword (oldPassword: String, newPassword: String, admin: Admin): boolean
+ login(username: String, password: String): boolean
+ userExist(username: String): boolean
+ getAdmin(username: String): Admin
+ addMessage(message: Message): void
+ getAdminData(): Map<String, Admin>
GlobalInventoryManager
-globalInventory: GlobalInventory
+GlobalInventoryManager(globalInventory: GlobalInventory)
+ getItemFromGI(itemID: String):Item
+ getItemsFromGI(itemIDList: list<String>): List<Item>
+ IDGenerator(): String
+ addItemTomap(item: Item):void
+ removeItemAndAddToGarbageBin (itemID: String):void
+ undoDeleteItem(userid: String): void
+ removeItemOnly(itemID: String): void
+getPersonInventory(ownerName: String): List<Item>
+ contains(item: Item): boolean
+ hasNoItem():boolean
+ getGlobalInventoryData(): GlobalInventory

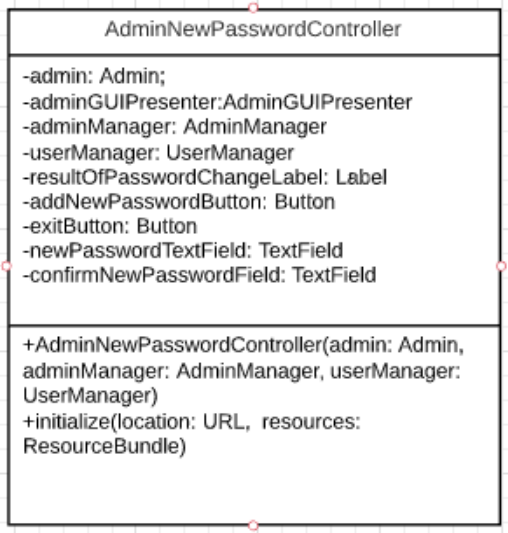
gateways


GlobalWishlistGateway
+ wishlist: GlobalWishlist
+ GlobalWishlistGateway(filepath: String)
+ writeToFile (filepath: String, wishlistItems2: GlobalWishList)
+ readFromFile (filepath: String): GlobalWishlist
+ getWishlistItems(): GlobalWishList
AdminAccountGateways
-filePath: String
-adminMap: Map<String, Admin>
+AdminAccountGateways (filePath:String)
+readFromFile(): void
+saveToFile(adminMap: map<String,Admin>): void
+getAdminMap(): map<String, Admin>
+ beginAdminMap(): void
AdminMessageGateway
+ messages: List<Message>
+ AdminMessageGateway(filepath: String)
+ writeToFile (filepath: String, adminMessages: List <Message>): void
+ readFromFile (filepath: String): List<Message>
+ getMessages(): List<Message>
GlobalInventoryGateways
- filePath: String
- globalInventoryManager: GlobalInventoryManager
- globalInventory: GlobalInventory
+GlobalInventoryGateways(filePath:String)
+writeToFile(globalInventory: GlobalInventory): void
+getGlobalInventory(): GlobalInventory
UserTradesGateway
+ userTrades: Map<String, List <Trade>>
+ UserTradesGateway(filepath: String):
+ writeToFile (filepath: String, userTrades: Map <String, List <Trade>>): void
+ readFromFile (filepath: String): Map<String, List<Trade>>
+ getUserTrades(): map<String, list<Trade>>
UserGateway
- mapOfUsers: Map<String, User>
+ UserGateway(filepath: String)
+ readFromFile(filepath: String): Map<String, User>
+ writeToFile(filepath: String, userObjects: Map<String, User>): void
+ getMapOfUsers(): Map<String, User>

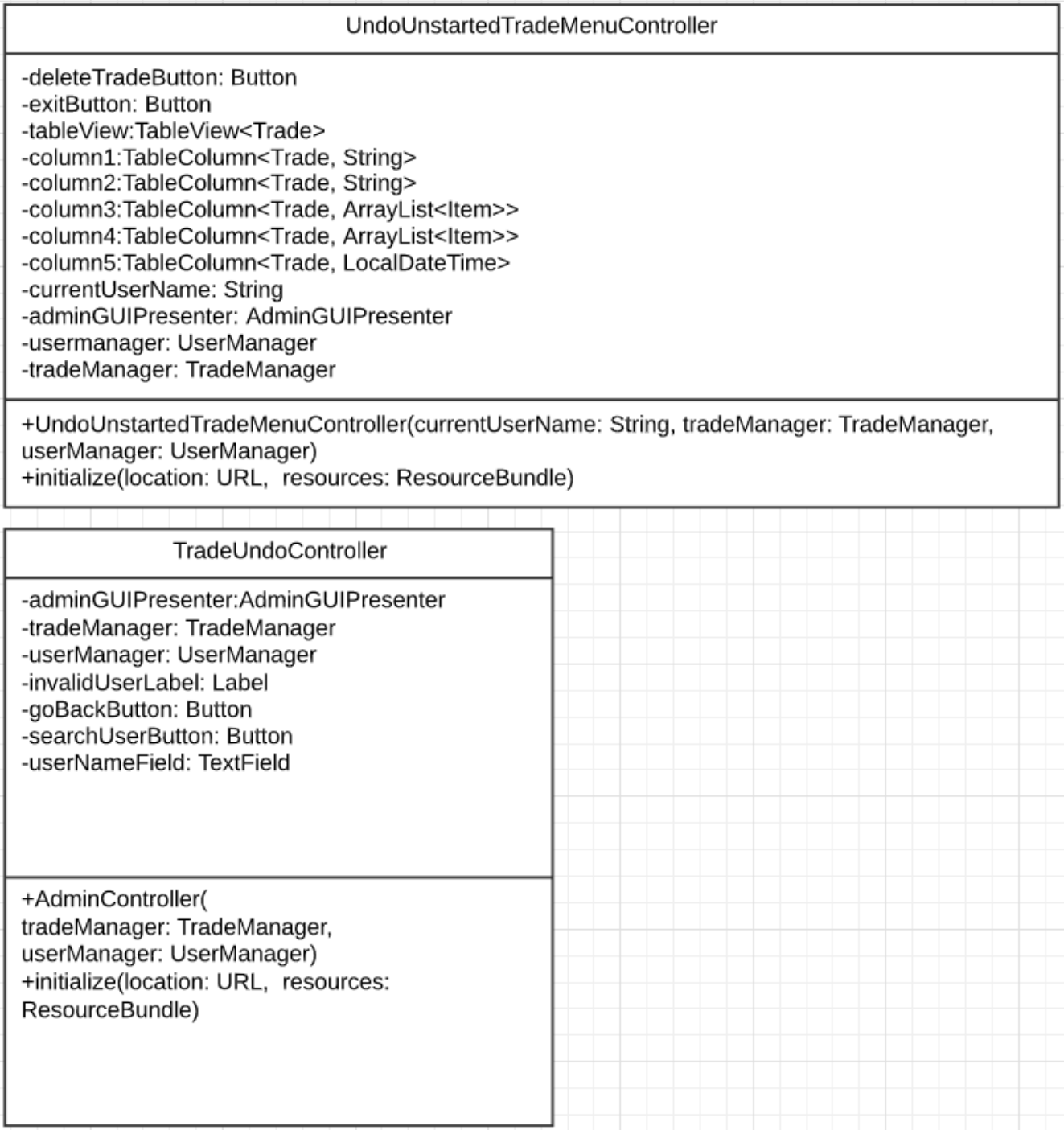
adminGUI

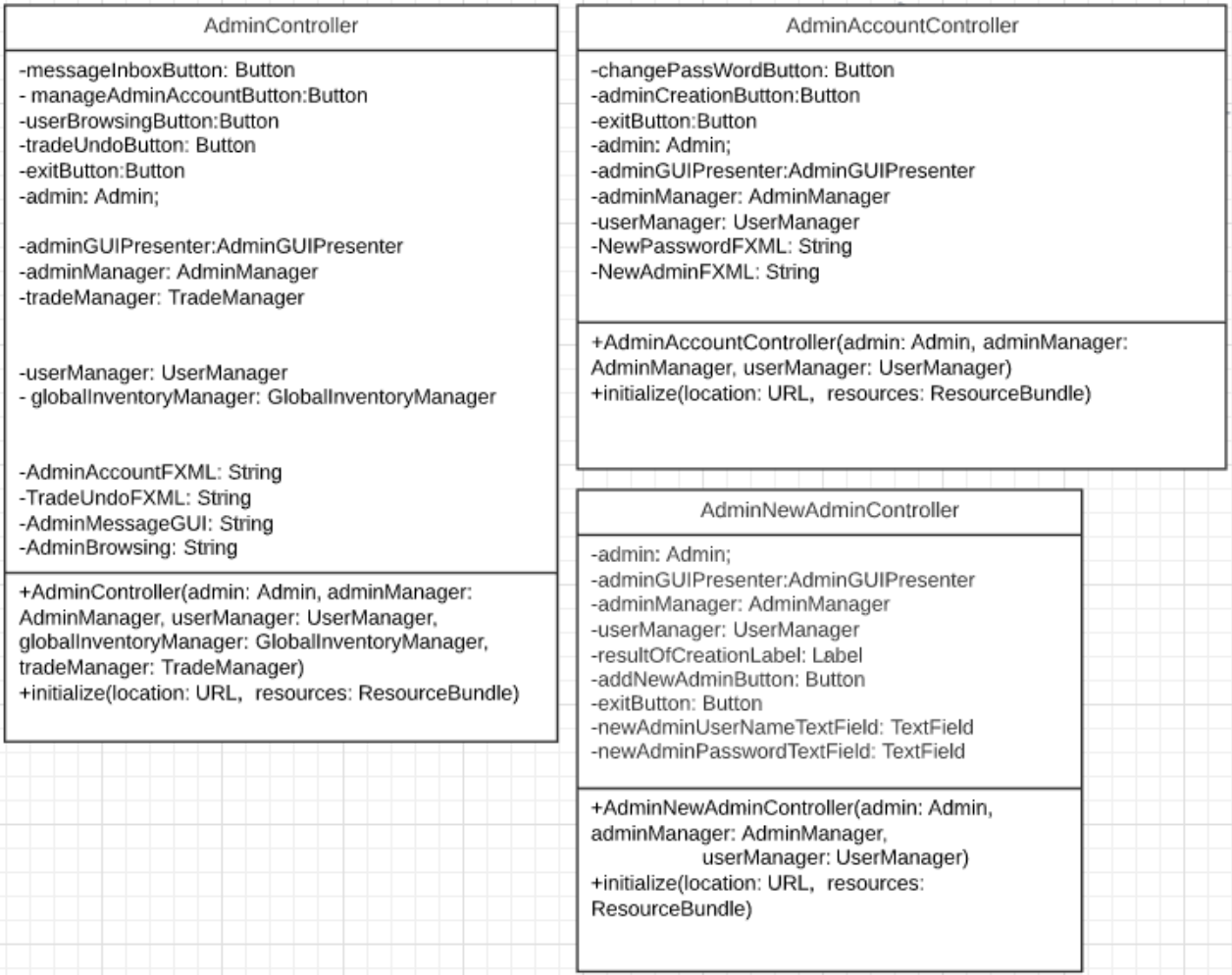

AdminBrowsingUsersController
- users: UserManager
- browse: AdminBrowsingUsersPresenter
- globalinventory: GlobalInventoryManager
- user: String
- lendingLimit: boolean
- weeklyLimit: boolean
- incompleteLimit: boolean
- allLimit: boolean
+AdminBrowsingUsersController(system: UserManager, globalinventory: GlobalInventoryManager): void
+initialize(location: URL, resources: ResourceBundle): void
- search(): void
- incompleteLimit(): void
- weeklyLimit(): void
- ban(): void
- freeze(): void
- lending(): void
- undoDelete(): void
- optionInput(): void
- exit(): void


AdminBrowsingUsersController
+ optionPrompt(): String
+ invalidName(): String
+ enterValuePrompt(): String
+ banStateChangeSuccess(): String
+ freezeStateChangeSuccess(): String
+ weeklyLimitChangeSuccess(): String
+ incompleteLimitChangeSuccess(): String
+ thresholdChangeSuccess(): String
+ wrongFormat(): String
+ deletedItemRestored(): String
+ enterUsername(): String
+ searchText(): String
+ banUnbanText(): String
+ freezeText(): String
+ incompleteText(): String
+ weeklyText(): String
+ lendingText(): String
+ enterText(): String
+ thresholdText(): String
+ exitText(): String
+ undoText(): String

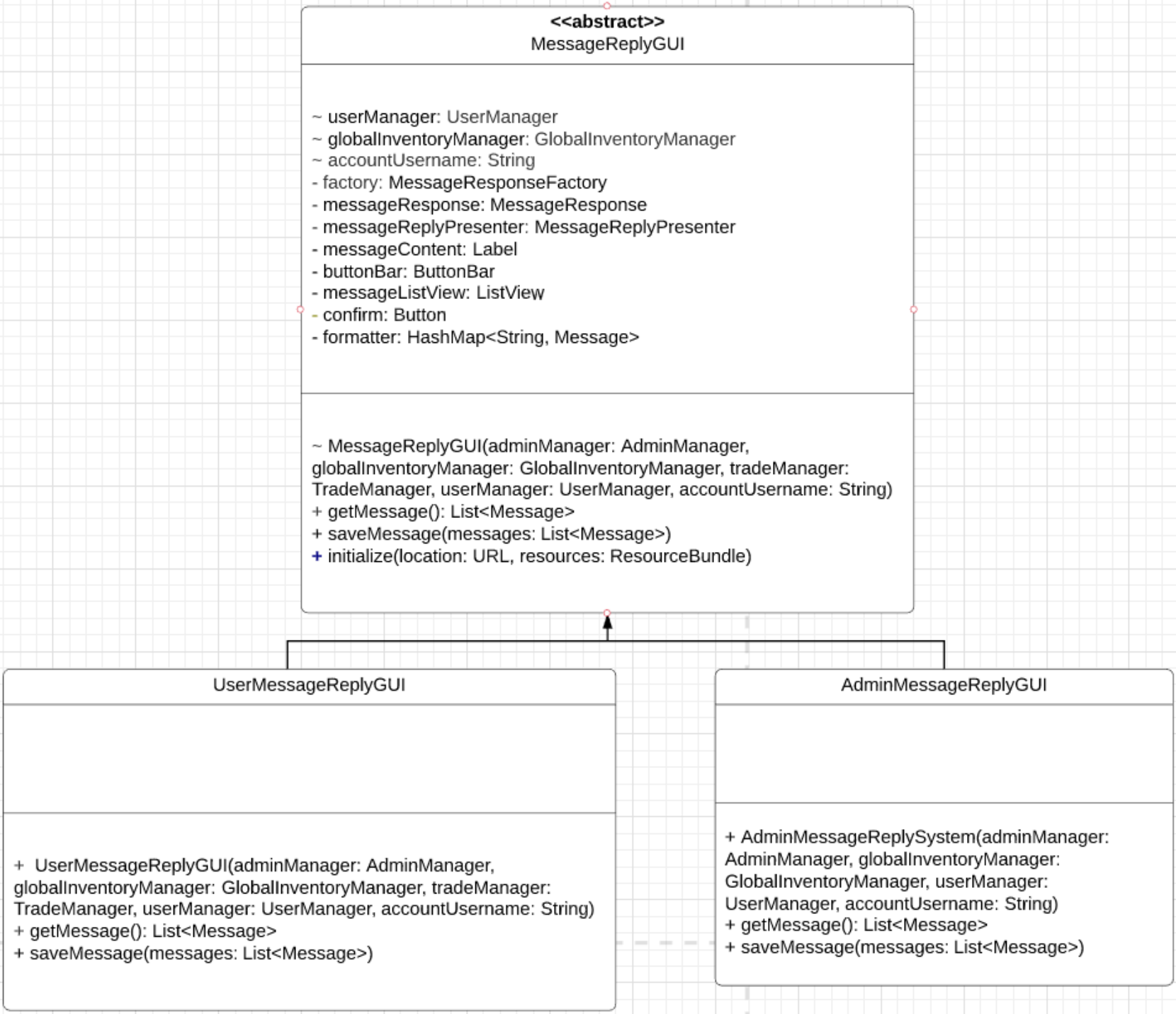

AdminNewPasswordController
-admin: Admin;
-adminGUIPresenter:AdminGUIPresenter
-adminManager: AdminManager
-userManager: UserManager
-resultOfPasswordChangeLabel: Label
-addNewPasswordButton: Button
-exitButton: Button
-newPasswordTextField: TextField
-confirmNewPasswordField: TextField
+AdminNewPasswordController(admin: Admin, adminManager: AdminManager, userManager: UserManager)
+initialize(location: URL, resources: ResourceBundle)

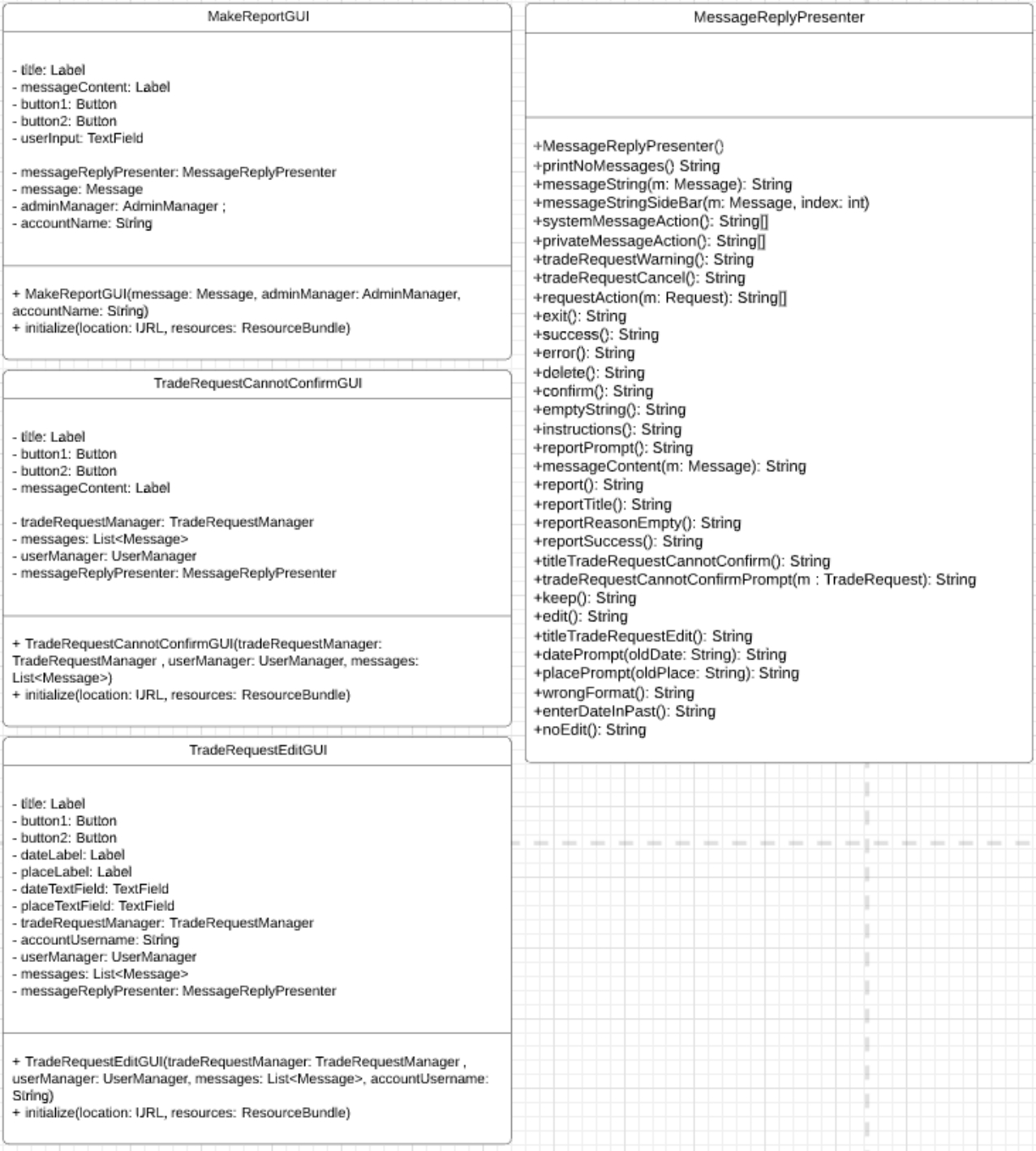

AdminGUIPresenter
+adminBrowsingWindow():String
+adminAccountWindow(): String
+adminPasswordChangeWindow(): String
+adminAccountCreationWindow(): String
+adminTradeUndoSearchWindow(): String
+undoUserTradeWindow():String
+adminMessageWindow(): String
+InvalidUserNameLabel(): String
+userNameCannotBeEmpty():String
+passwordCannotBeEmpty(): String
+newPasswordCreated(admin: Admin): String
+newAdminCreated(): String
+newAdminCreated(): String
+AdminCreationFailed(): String
+newPasswordNotSaved(): String
+exitButton(): String
+searchButtonText(): String
+saveChangeButtonText():String
+adminCreationButtonText(): String
+adminChangePasswordButton(): String
+messageInboxButton(): String
+adminAccountButton(): String
+userBrowsingButton(): String
+tradeUndoButton(): String
+deleteSelectedTradeButton(): String
+newAdminButton(): String

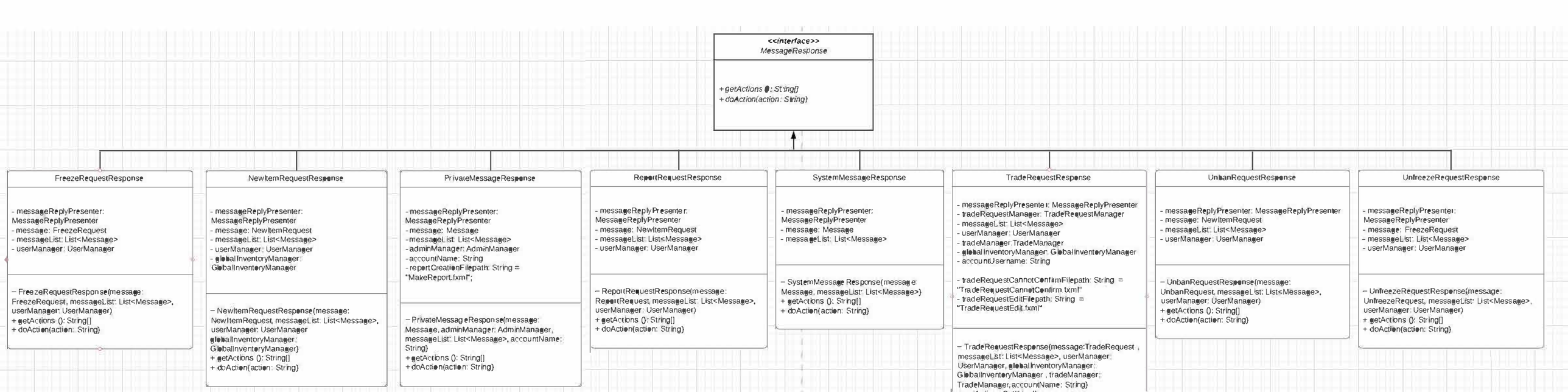

UndoUnstartedTradeMenuController
-deleteTradeButton: Button
-exitButton: Button
-tableView:TableView<Trade>
-column1:TableColumn<Trade, String>
-column2:TableColumn<Trade, String>
-column3:TableColumn<Trade, ArrayList<Item>>
-column4:TableColumn<Trade, ArrayList<Item>>
-column5:TableColumn<Trade, LocalDateTime>
-currentUserName: String
-adminGUIPresenter: AdminGUIPresenter
-usermanager: UserManager
-tradeManager: TradeManager
+UndoUnstartedTradeMenuController(currentUserName: String, tradeManager: TradeManager, userManager: UserManager)
+initialize(location: URL, resources: ResourceBundle)
TradeUndoController
-adminGUIPresenter:AdminGUIPresenter
-tradeManager: TradeManager
-userManager: UserManager
-invalidUserLabel: Label
-goBackButton: Button
-searchUserButton: Button
-userNameField: TextField
+AdminController(
tradeManager: TradeManager,
userManager: UserManager)
+initialize(location: URL, resources:
ResourceBundle)


AdminController
-messageInboxButton: Button
- manageAdminAccountButton:Button
-userBrowsingButton:Button
-tradeUndoButton: Button
-exitButton:Button
-admin: Admin;
-adminGUIPresenter:AdminGUIPresenter
-adminManager: AdminManager
-tradeManager: TradeManager
-userManager: UserManager
- globalInventoryManager: GlobalInventoryManager
-AdminAccountFXML: String
-TradeUndoFXML: String
-AdminMessageGUI: String
-AdminBrowsing: String
+AdminController(admin: Admin, adminManager: AdminManager, userManager: UserManager, globalInventoryManager: GlobalInventoryManager, tradeManager: TradeManager)
+initialize(location: URL, resources: ResourceBundle)
AdminAccountController
-changePassWordButton: Button
-adminCreationButton:Button
-exitButton:Button
-admin: Admin;
-adminGUIPresenter:AdminGUIPresenter
-adminManager: AdminManager
-userManager: UserManager
-NewPasswordFXML: String
-NewAdminFXML: String
+AdminAccountController(admin: Admin, adminManager: AdminManager, userManager: UserManager)
+initialize(location: URL, resources: ResourceBundle)
AdminNewAdminController
-admin: Admin;
-adminGUIPresenter:AdminGUIPresenter
-adminManager: AdminManager
-userManager: UserManager
-resultOfCreationLabel: Label
-addNewAdminButton: Button
-exitButton: Button
-newAdminUserNameTextField: TextField
-newAdminPasswordTextField: TextField
+AdminNewAdminController(admin: Admin, adminManager: AdminManager, userManager: UserManager)
+initialize(location: URL, resources: ResourceBundle)

messagereplyGUI

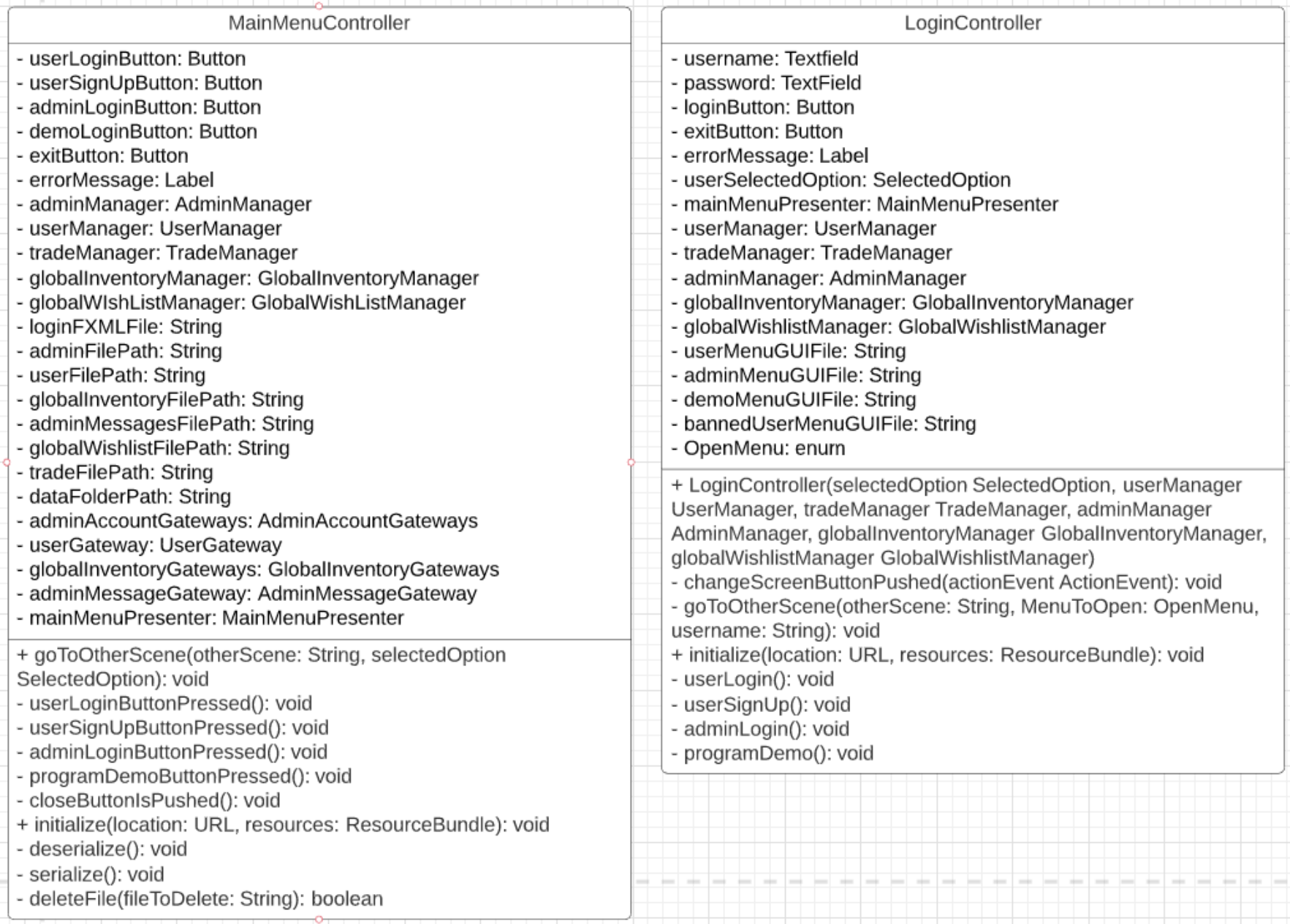

<<abstract>>
MessageReplyGUI
~ userManager: UserManager
~ globalInventoryManager: GlobalInventoryManager
~ accountUsername: String
- factory: MessageResponseFactory
- messageResponse: MessageResponse
- messageReplyPresenter: MessageReplyPresenter
- messageContent: Label
- buttonBar: ButtonBar
- messageListView: ListView
- confirm: Button
- formatter: HashMap<String, Message>
~ MessageReplyGUI(adminManager: AdminManager, globalInventoryManager: GlobalInventoryManager, tradeManager: TradeManager, userManager: UserManager, accountUsername: String)
+ getMessage(): List<Message>
+ saveMessage(messages: List<Message>)
+ initialize(location: URL, resources: ResourceBundle)
UserMessageReplyGUI
+ UserMessageReplyGUI(adminManager: AdminManager, globalInventoryManager: GlobalInventoryManager, tradeManager: TradeManager, userManager: UserManager, accountUsername: String)
+ getMessage(): List<Message>
+ saveMessage(messages: List<Message>)
AdminMessageReplyGUI
+ AdminMessageReplySystem(adminManager: AdminManager, globalInventoryManager: GlobalInventoryManager, userManager: UserManager, accountUsername: String)
+ getMessage(): List<Message>
+ saveMessage(messages: List<Message>)


MakeReportGUI
- title: Label
- messageContent: Label
- button1: Button
- button2: Button
- userInput: TextField
- messageReplyPresenter: MessageReplyPresenter
- message: Message
- adminManager: AdminManager ;
- accountName: String
+ MakeReportGUI(message: Message, adminManager: AdminManager, accountName: String)
+ initialize(location: IJRL, resources: ResourceBundle)
TradeRequestCannotConfirmGUI
- title: Label
- button1: Button
- button2: Button
- messageContent: Label
- tradeRequestManager: TradeRequestManager
- messages: List<Message>
- userManager: UserManager
- messageReplyPresenter: MessageReplyPresenter
+ TradeRequestCannotConfirmGUI(tradeRequestManager: TradeRequestManager , userManager: UserManager, messages: List<Message>)
+ initialize(location: IJRL, resources: ResourceBundle)
TradeRequestEditGUI
- title: Label
- button1: Button
- button2: Button
- dateLabel: Label
- placeLabel: Label
- dateTextField: TextField
- placeTextField: TextField
- tradeRequestManager: TradeRequestManager
- accountUsername: String
- userManager: UserManager
- messages: List<Message>
- messageReplyPresenter: MessageReplyPresenter
+ TradeRequestEditGUI(tradeRequestManager: TradeRequestManager , userManager: UserManager, messages: List<Message>, accountUsername: String)
+ initialize(location: IJRL, resources: ResourceBundle)
MessageReplyPresenter
+MessageReplyPresenter()
+printNoMessages() String
+messageString(m: Message): String
+messageStringSideBar(m: Message, index: int)
+systemMessageAction(): String[]
+privateMessageAction(): String[]
+tradeRequestWarning(): String
+tradeRequestCancel(): String
+requestAction(m: Request): String[]
+exit(): String
+success(): String
+error(): String
+delete(): String
+confirm(): String
+emptyString(): String
+instructions(): String
+reportPrompt(): String
+messageContent(m: Message): String
+report(): String
+reportTitle(): String
+reportReasonEmpty(): String
+reportSuccess(): String
+titleTradeRequestCannotConfirm(): String
+tradeRequestCannotConfirmPrompt(m : TradeRequest): String
+keep(): String
+edit(): String
+titleTradeRequestEdit(): String
+datePrompt(oldDate: String): String
+placePrompt(oldPlace: String): String
+wrongFormat(): String
+enterDateInPast(): String
+noEdit(): String

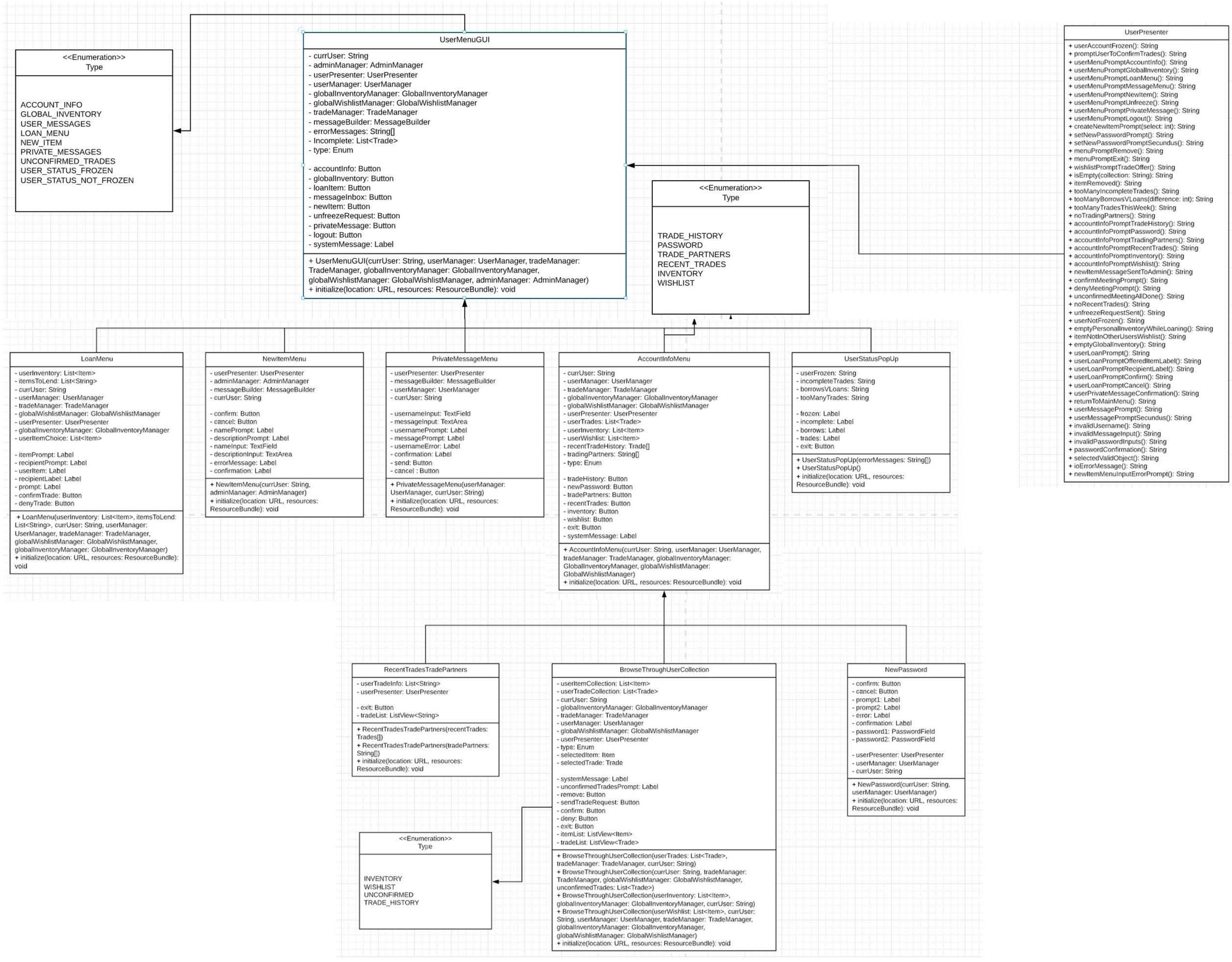

<<interface>>
MessageResponse
+getActions (): String[]
+doAction(action: String)
FreezeRequestResponse
- messageReplyPresenter: MessageReplyPresenter
- message: FreezeRequest
- messageList: List<Message>
- userManager: UserManager
– FreezeRequestResponse(message: FreezeRequest, messageList: List<Message>, userManager: UserManager)
+ getActions (): String[]
+ doAction(action: String)
NewItemRequestResponse
- messageReplyPresenter: MessageReplyPresenter
- message: NewItemRequest
- messageList: List<Message>
- userManager: UserManager
- globalInventoryManager: GlobalInventoryManager
– NewItemRequestResponse(message: NewItemRequest, messageList: List<Message>, userManager: UserManager globalInventoryManager: GlobalInventoryManager)
+ getActions (): String[]
+ doAction(action: String)
PrivateMessageResponse
- messageReplyPresenter: MessageReplyPresenter
- message: Message
- messageList: List<Message>
- adminManager: AdminManager
- accountName: String
- reportCreationFilepath: String = "MakeReport.fxml";
– PrivateMessageResponse(message: Message, adminManager: AdminManager, messageList: List<Message>, accountName: String)
+ getActions (): String[]
+ doAction(action: String)
ReportRequestResponse
- messageReplyPresenter: MessageReplyPresenter
- message: NewItemRequest
- messageList: List<Message>
- userManager: UserManager
– ReportRequestResponse(message: ReportRequest, messageList: List<Message>, userManager: UserManager)
+ getActions (): String[]
+ doAction(action: String)
SystemMessageResponse
- messageReplyPresenter: MessageReplyPresenter
- message: Message
- messageList: List<Message>
– SystemMessageResponse(message: Message, messageList: List<Message>)
+ getActions (): String[]
+ doAction(action: String)
TradeRequestResponse
- messageReplyPresenter: MessageReplyPresenter
- tradeRequestManager: TradeRequestManager
- messageList: List<Message>
- userManager: UserManager
- tradeManager:TradeManager
- globalInventoryManager: GlobalInventoryManager
- accountUsername: String
- tradeRequestCannotConfirmFilepath: String = "TradeRequestCannotConfirm.fxml"
- tradeRequestEditFilepath: String = "TradeRequestEdit.fxml"
– TradeRequestResponse(message:TradeRequest , messageList: List<Message>, userManager: UserManager, globalInventoryManager: GlobalInventoryManager , tradeManager: TradeManager, accountName: String)
UnbanRequestResponse
- messageReplyPresenter: MessageReplyPresenter
- message: NewItemRequest
- messageList: List<Message>
- userManager: UserManager
– UnbanRequestResponse(message: UnbanRequest, messageList: List<Message>, userManager: UserManager)
+ getActions (): String[]
+ doAction(action: String)
UnfreezeRequestResponse
- messageReplyPresenter: MessageReplyPresenter
- message: FreezeRequest
- messageList: List<Message>
- userManager: UserManager
– UnfreezeRequestResponse(message: UnfreezeRequest, messageList: List<Message>, userManager: UserManager)
+ getActions (): String[]
+ doAction(action: String)

mainMenuGUI

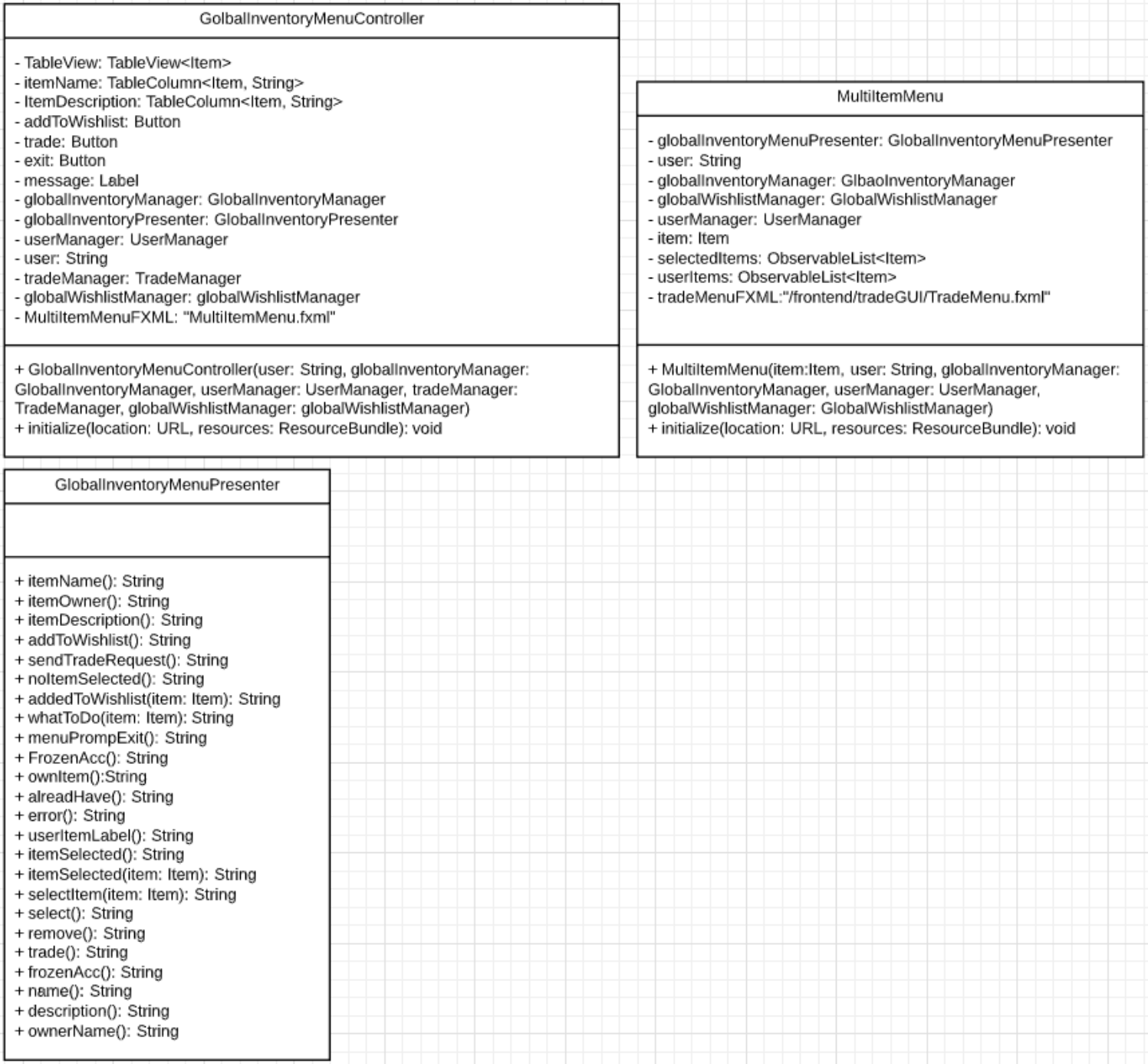

StartScreen
- mainMenuPresenter MainMenuPresenter
- mainMenuFXMLFile String
+ start(primaryStage: Stage): void
+ main(args: String[]): void
UseCaseBuilder
+ getAdminManager AdminManager(adminList: Map<String, Admin>, messageList: List<Message>): AdminManager
+ getTradeManager TradeManager(tradeList: Map<String, List<Trade>>): TradeManager
+ getGlobalInventoryManager GlobalInventoryManager(globalInventory: GlobalInventory): GlobalInventoryManager
+ getGlobalWishlistManager(globalWishlist: GlobalWishList): GlobalWishlistManager
+ getDemoUserManager(username: String, password: String): DemoUserManager
<<enumeration>> SelectedOption
USER_LOGIN
USER_SIGNUP
ADMIN_LOGIN
DEMO_LOGIN
GateWayBuilder
+ getAdminAccountGateways(filepath: String): AdminAccountGateways
+ getAdminMessageGateways(filepath: String): AdminMessageGateway
+ getUserGateway(filepath: String): UserGateway
+ getUserTradesGateway(filepath: String): UserTradesGateway
+ getGlobalWishlistGateway(filepath: String): GlobalWishlistGateway
+ getGlobalInventoryGateways(filepath: String): GlobalInventoryGateways
MainMenuPresenter
+ printExit(): void
+ corruptedData(): return String
+ userLoginOption(): return String
+ userSignUpOption(): return String
+ adminLoginOption(): return String
+ demoLoginOption(): return String
+ exitOption(): return String
+ savingError(): return String
+ failedLogin(): return String
+ login(): return String
+ returnToMainMenu(): return String
+ signUp(): return String
+ stringReset(): return String
+ takenOrInvalidUsername(): return String
+ successfulAccountCreation(): return String
+ wrongLogin(): return String
+ programName(): return String


MainMenuController
- userLoginButton: Button
- userSignUpButton: Button
- adminLoginButton: Button
- demoLoginButton: Button
- exitButton: Button
- errorMessage: Label
- adminManager: AdminManager
- userManager: UserManager
- tradeManager: TradeManager
- globalInventoryManager: GlobalInventoryManager
- globalWIshListManager: GlobalWishListManager
- loginFXMLFile: String
- adminFilePath: String
- userFilePath: String
- globalInventoryFilePath: String
- adminMessagesFilePath: String
- globalWishlistFilePath: String
- tradeFilePath: String
- dataFolderPath: String
- adminAccountGateways: AdminAccountGateways
- userGateway: UserGateway
- globalInventoryGateways: GlobalInventoryGateways
- adminMessageGateway: AdminMessageGateway
- mainMenuPresenter: MainMenuPresenter
+ goToOtherScene(otherScene: String, selectedOption SelectedOption): void
- userLoginButtonPressed(): void
- userSignUpButtonPressed(): void
- adminLoginButtonPressed(): void
- programDemoButtonPressed(): void
- closeButtonIsPushed(): void
+ initialize(location: URL, resources: ResourceBundle): void
- deserialize(): void
- serialize(): void
- deleteFile(fileToDelete: String): boolean
LoginController
- username: Textfield
- password: TextField
- loginButton: Button
- exitButton: Button
- errorMessage: Label
- userSelectedOption: SelectedOption
- mainMenuPresenter: MainMenuPresenter
- userManager: UserManager
- tradeManager: TradeManager
- adminManager: AdminManager
- globalInventoryManager: GlobalInventoryManager
- globalWishlistManager: GlobalWishlistManager
- userMenuGUIFile: String
- adminMenuGUIFile: String
- demoMenuGUIFile: String
- bannedUserMenuGUIFile: String
- OpenMenu: enum
+ LoginController(selectedOption SelectedOption, userManager UserManager, tradeManager TradeManager, adminManager AdminManager, globalInventoryManager GlobalInventoryManager, globalWishlistManager GlobalWishlistManager)
- changeScreenButtonPushed(actionEvent ActionEvent): void
- goToOtherScene(otherScene: String, MenuToOpen: OpenMenu, username: String): void
+ initialize(location: URL, resources: ResourceBundle): void
- userLogin(): void
- userSignUp(): void
- adminLogin(): void
- programDemo(): void

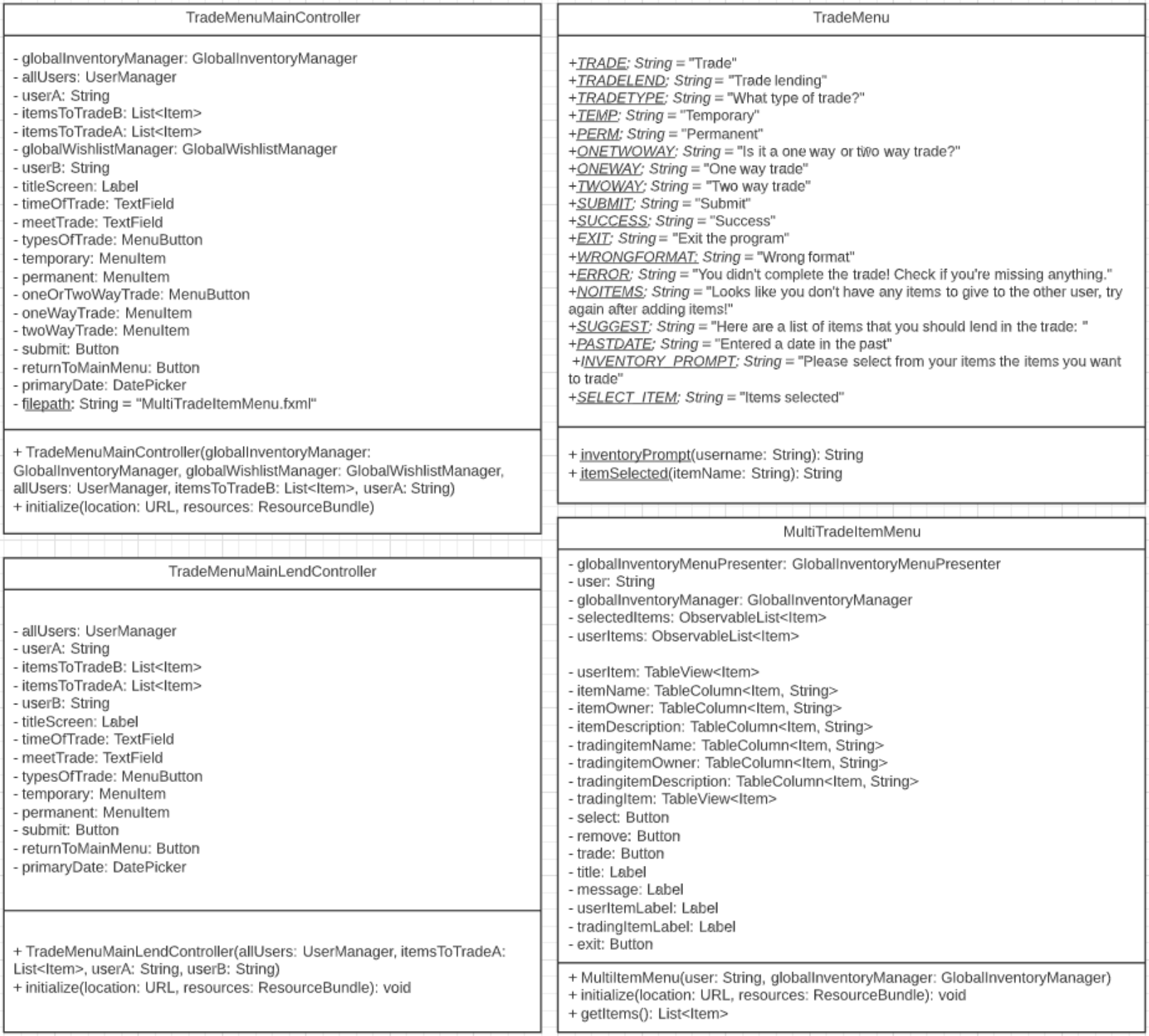
userGUI


<<Enumeration>>
Type
ACCOUNT_INFO
GLOBAL_INVENTORY
USER_MESSAGES
LOAN_MENU
NEW_ITEM
PRIVATE_MESSAGES
UNCONFIRMED_TRADES
USER_STATUS_FROZEN
USER_STATUS_NOT_FROZEN
UserMenuGUI
- currUser: String
- adminManager: AdminManager
- userPresenter: UserPresenter
- userManager: UserManager
- globalInventoryManager: GlobalInventoryManager
- globalWishlistManager: GlobalWishlistManager
- tradeManager: TradeManager
- messageBuilder: MessageBuilder
- errorMessages: String[]
- Incomplete: List<Trade>
- type: Enum
- accountInfo: Button
- globalInventory: Button
- loanItem: Button
- messageInbox: Button
- newItem: Button
- unfreezeRequest: Button
- privateMessage: Button
- logout: Button
- systemMessage: Label
+ UserMenuGUI(currUser: String, userManager: UserManager, tradeManager: TradeManager, globalInventoryManager: GlobalInventoryManager, globalWishlistManager: GlobalWishlistManager, adminManager: AdminManager)
+ initialize(location: URL, resources: ResourceBundle): void
UserPresenter
+ userAccountFrozen(): String
+ promptUserToConfirmTrades(): String
+ userMenuPromptAccountInfo(): String
+ userMenuPromptGlobalInventory(): String
+ userMenuPromptLoanMenu(): String
+ userMenuPromptMessageMenu(): String
+ userMenuPromptNewItem(): String
+ userMenuPromptUnfreeze(): String
+ userMenuPromptPrivateMessage(): String
+ userMenuPromptLogout(): String
+ createNewItemPrompt(select: int): String
+ setNewPasswordPrompt(): String
+ setNewPasswordPromptSecundus(): String
+ menuPromptRemove(): String
+ menuPromptExit(): String
+ wishlistPromptTradeOffer(): String
+ isEmpty(collection: String): String
+ itemRemoved(): String
+ tooManyIncompleteTrades(): String
+ tooManyBorrowsVLoans(difference: int): String
+ tooManyTradesThisWeek(): String
+ noTradingPartners(): String
+ accountInfoPromptTradeHistory(): String
+ accountInfoPromptPassword(): String
+ accountInfoPromptTradingPartners(): String
+ accountInfoPromptRecentTrades(): String
+ accountInfoPromptInventory(): String
+ accountInfoPromptWishlist(): String
+ newItemMessageSentToAdmin(): String
+ confirmMeetingPrompt(): String
+ denyMeetingPrompt(): String
+ unconfirmedMeetingAllDone(): String
+ noRecentTrades(): String
+ unfreezeRequestSent(): String
+ userNotFrozen(): String
+ emptyPersonalInventoryWhileLoaning(): String
+ itemNotInOtherUsersWishlist(): String
+ emptyGlobalInventory(): String
+ userLoanPrompt(): String
+ userLoanPromptOfferedItemLabel(): String
+ userLoanPromptRecipientLabel(): String
+ userLoanPromptConfirm(): String
+ userLoanPromptCancel(): String
+ userPrivateMessageConfirmation(): String
+ returnToMainMenu(): String
+ userMessagePrompt(): String
+ userMessagePromptSecundus(): String
+ invalidUsername(): String
+ invalidMessageInput(): String
+ invalidPasswordInputs(): String
+ passwordConfirmation(): String
+ selectedValidObject(): String
+ ioErrorMessage(): String
+ newItemMenuInputErrorPrompt(): String
<<Enumeration>>
Type
TRADE_HISTORY
PASSWORD
TRADE_PARTNERS
RECENT_TRADES
INVENTORY
WISHLIST
LoanMenu
- userInventory: List<Item>
- itemsToLend: List<String>
- currUser: String
- userManager: UserManager
- tradeManager: TradeManager
- globalWishlistManager: GlobalWishlistManager
- userPresenter: UserPresenter
- globalInventoryManager: GlobalInventoryManager
- userItemChoice: List<Item>
- itemPrompt: Label
- recipientPrompt: Label
- userItem: Label
- recipientLabel: Label
- prompt: Label
- confirmTrade: Button
- denyTrade: Button
+ LoanMenu(userInventory: List<Item>, itemsToLend: List<String>, currUser: String, userManager: UserManager, tradeManager: TradeManager, globalWishlistManager: GlobalWishlistManager, globalInventoryManager: GlobalInventoryManager)
+ initialize(location: URL, resources: ResourceBundle): void
NewItemMenu
- userPresenter: UserPresenter
- adminManager: AdminManager
- messageBuilder: MessageBuilder
- currUser: String
- confirm: Button
- cancel: Button
- namePrompt: Label
- descriptionPrompt: Label
- nameInput: TextField
- descriptionInput: TextArea
- errorMessage: Label
- confirmation: Label
+ NewItemMenu(currUser: String, adminManager: AdminManager)
+ initialize(location: URL, resources: ResourceBundle): void
PrivateMessageMenu
- userPresenter: UserPresenter
- messageBuilder: MessageBuilder
- userManager: UserManager
- currUser: String
- usernameInput: TextField
- messageInput: TextArea
- usernamePrompt: Label
- messagePrompt: Label
- usernameError: Label
- confirmation: Label
- send: Button
- cancel : Button
+ PrivateMessageMenu(userManager: UserManager, currUser: String)
+ initialize(location: URL, resources: ResourceBundle): void
AccountInfoMenu
- currUser: String
- userManager: UserManager
- tradeManager: TradeManager
- globalInventoryManager: GlobalInventoryManager
- globalWishlistManager: GlobalWishlistManager
- userPresenter: UserPresenter
- userTrades: List<Trade>
- userInventory: List<Item>
- userWishlist: List<Item>
- recentTradeHistory: Trade[]
- tradingPartners: String[]
- type: Enum
- tradeHistory: Button
- newPassword: Button
- tradePartners: Button
- recentTrades: Button
- inventory: Button
- wishlist: Button
- exit: Button
- systemMessage: Label
+ AccountInfoMenu(currUser: String, userManager: UserManager, tradeManager: TradeManager, globalInventoryManager: GlobalInventoryManager, globalWishlistManager: GlobalWishlistManager)
+ initialize(location: URL, resources: ResourceBundle): void
UserStatusPopUp
- userFrozen: String
- incompleteTrades: String
- borrowsVLoans: String
- tooManyTrades: String
- frozen: Label
- incomplete: Label
- borrows: Label
- trades: Label
- exit: Button
+ UserStatusPopUp(errorMessages: String[])
+ UserStatusPopUp()
+ initialize(location: URL, resources: ResourceBundle): void
RecentTradesTradePartners
- userTradeInfo: List<String>
- userPresenter: UserPresenter
- exit: Button
- tradeList: ListView<String>
+ RecentTradesTradePartners(recentTrades: Trades[])
+ RecentTradesTradePartners(tradePartners: String[])
+ initialize(location: URL, resources: ResourceBundle): void
BrowseThroughUserCollection
- userItemCollection: List<Item>
- userTradeCollection: List<Trade>
- currUser: String
- globalInventoryManager: GlobalInventoryManager
- tradeManager: TradeManager
- userManager: UserManager
- globalWishlistManager: GlobalWishlistManager
- userPresenter: UserPresenter
- type: Enum
- selectedItem: Item
- selectedTrade: Trade
- systemMessage: Label
- unconfirmedTradesPrompt: Label
- remove: Button
- sendTradeRequest: Button
- confirm: Button
- deny: Button
- exit: Button
- itemList: ListView<Item>
- tradeList: ListView<Trade>
+ BrowseThroughUserCollection(userTrades: List<Trade>, tradeManager: TradeManager, currUser: String)
+ BrowseThroughUserCollection(currUser: String, tradeManager: TradeManager, globalWishlistManager: GlobalWishlistManager, unconfirmedTrades: List<Trade>)
+ BrowseThroughUserCollection(userInventory: List<Item>, globalInventoryManager: GlobalInventoryManager, currUser: String)
+ BrowseThroughUserCollection(userWishlist: List<Item>, currUser: String, userManager: UserManager, tradeManager: TradeManager, globalInventoryManager: GlobalInventoryManager, globalWishlistManager: GlobalWishlistManager)
+ initialize(location: URL, resources: ResourceBundle): void
NewPassword
- confirm: Button
- cancel: Button
- prompt1: Label
- prompt2: Label
- error: Label
- confirmation: Label
- password1: PasswordField
- password2: PasswordField
- userPresenter: UserPresenter
- userManager: UserManager
- currUser: String
+ NewPassword(currUser: String, userManager: UserManager)
+ initialize(location: URL, resources: ResourceBundle): void
<<Enumeration>>
Type
INVENTORY
WISHLIST
UNCONFIRMED
TRADE_HISTORY

globalInventoryGUI


GolbalInventoryMenuController
- TableView: TableView<Item>
- itemName: TableColumn<Item, String>
- ItemDescription: TableColumn<Item, String>
- addToWishlist: Button
- trade: Button
- exit: Button
- message: Label
- globalInventoryManager: GlobalInventoryManager
- globalInventoryPresenter: GlobalInventoryPresenter
- userManager: UserManager
- user: String
- tradeManager: TradeManager
- globalWishlistManager: globalWishlistManager
- MultiItemMenuFXML: "MultiItemMenu.fxml"
+ GlobalInventoryMenuController(user: String, globalInventoryManager: GlobalInventoryManager, userManager: UserManager, tradeManager: TradeManager, globalWishlistManager: globalWishlistManager)
+ initialize(location: URL, resources: ResourceBundle): void
MultiItemMenu
- globalInventoryMenuPresenter: GlobalInventoryMenuPresenter
- user: String
- globalInventoryManager: GlbaoInventoryManager
- globalWishlistManager: GlobalWishlistManager
- userManager: UserManager
- item: Item
- selectedItems: ObservableList<Item>
- userItems: ObservableList<Item>
- tradeMenuFXML:"/frontend/tradeGUI/TradeMenu.fxml"
+ MultiItemMenu(item:Item, user: String, globalInventoryManager: GlobalInventoryManager, userManager: UserManager, globalWishlistManager: GlobalWishlistManager)
+ initialize(location: URL, resources: ResourceBundle): void
GlobalInventoryMenuPresenter
+ itemName(): String
+ itemOwner(): String
+ itemDescription(): String
+ addToWishlist(): String
+ sendTradeRequest(): String
+ noItemSelected(): String
+ addedToWishlist(item: Item): String
+ whatToDo(item: Item): String
+ menuPrompExit(): String
+ FrozenAcc(): String
+ ownItem():String
+ alreadHave(): String
+ error(): String
+ userItemLabel(): String
+ itemSelected(): String
+ itemSelected(item: Item): String
+ selectItem(item: Item): String
+ select(): String
+ remove(): String
+ trade(): String
+ frozenAcc(): String
+ name(): String
+ description(): String
+ ownerName(): String

tradeGUI


TradeMenuMainController
- globalInventoryManager: GlobalInventoryManager
- allUsers: UserManager
- userA: String
- itemsToTradeB: List<Item>
- itemsToTradeA: List<Item>
- globalWishlistManager: GlobalWishlistManager
- userB: String
- titleScreen: Label
- timeOfTrade: TextField
- meetTrade: TextField
- typesOfTrade: MenuButton
- temporary: MenuItem
- permanent: MenuItem
- oneOrTwoWayTrade: MenuButton
- oneWayTrade: MenuItem
- twoWayTrade: MenuItem
- submit: Button
- returnToMainMenu: Button
- primaryDate: DatePicker
- filepath: String = "MultiTradeItemMenu.fxml"
+ TradeMenuMainController(globalInventoryManager: GlobalInventoryManager, globalWishlistManager: GlobalWishlistManager, allUsers: UserManager, itemsToTradeB: List<Item>, userA: String)
+ initialize(location: URL, resources: ResourceBundle)
TradeMenu
+TRADE: String = "Trade"
+TRADELEND: String = "Trade lending"
+TRADETYPE: String = "What type of trade?"
+TEMP: String = "Temporary"
+PERM: String = "Permanent"
+ONETWOWAY: String = "Is it a one way or two way trade?"
+ONEWAY: String = "One way trade"
+TWOWAY: String = "Two way trade"
+SUBMIT: String = "Submit"
+SUCCESS: String = "Success"
+EXIT: String = "Exit the program"
+WRONGFORMAT: String = "Wrong format"
+ERROR: String = "You didn't complete the trade! Check if you're missing anything."
+NOITEMS: String = "Looks like you don't have any items to give to the other user, try again after adding items!"
+SUGGEST: String = "Here are a list of items that you should lend in the trade: "
+PASTDATE: String = "Entered a date in the past"
+INVENTORY_PROMPT: String = "Please select from your items the items you want to trade"
+SELECT_ITEM: String = "Items selected"
+ inventoryPrompt(username: String): String
+ itemSelected(itemName: String): String
TradeMenuMainLendController
- allUsers: UserManager
- userA: String
- itemsToTradeB: List<Item>
- itemsToTradeA: List<Item>
- userB: String
- titleScreen: Label
- timeOfTrade: TextField
- meetTrade: TextField
- typesOfTrade: MenuButton
- temporary: MenuItem
- permanent: MenuItem
- submit: Button
- returnToMainMenu: Button
- primaryDate: DatePicker
+ TradeMenuMainLendController(allUsers: UserManager, itemsToTradeA: List<Item>, userA: String, userB: String)
+ initialize(location: URL, resources: ResourceBundle): void
MultiTradeItemMenu
- globalInventoryMenuPresenter: GlobalInventoryMenuPresenter
- user: String
- globalInventoryManager: GlobalInventoryManager
- selectedItems: ObservableList<Item>
- userItems: ObservableList<Item>
- userItem: TableView<Item>
- itemName: TableColumn<Item, String>
- itemOwner: TableColumn<Item, String>
- itemDescription: TableColumn<Item, String>
- tradingitemName: TableColumn<Item, String>
- tradingitemOwner: TableColumn<Item, String>
- tradingitemDescription: TableColumn<Item, String>
- tradingItem: TableView<Item>
- select: Button
- remove: Button
- trade: Button
- title: Label
- message: Label
- userItemLabel: Label
- tradingItemLabel: Label
- exit: Button
+ MultiItemMenu(user: String, globalInventoryManager: GlobalInventoryManager)
+ initialize(location: URL, resources: ResourceBundle): void
+ getItems(): List<Item>