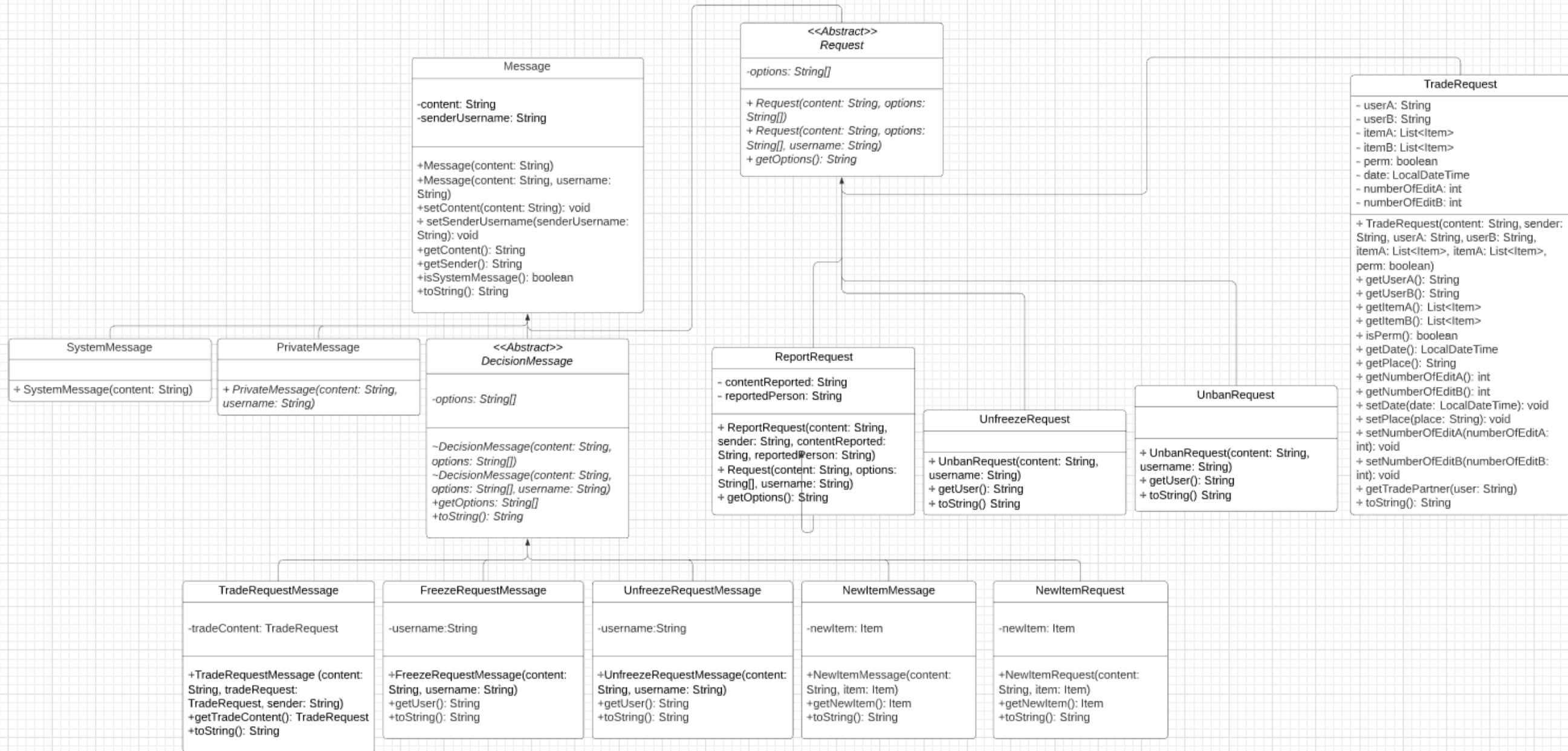
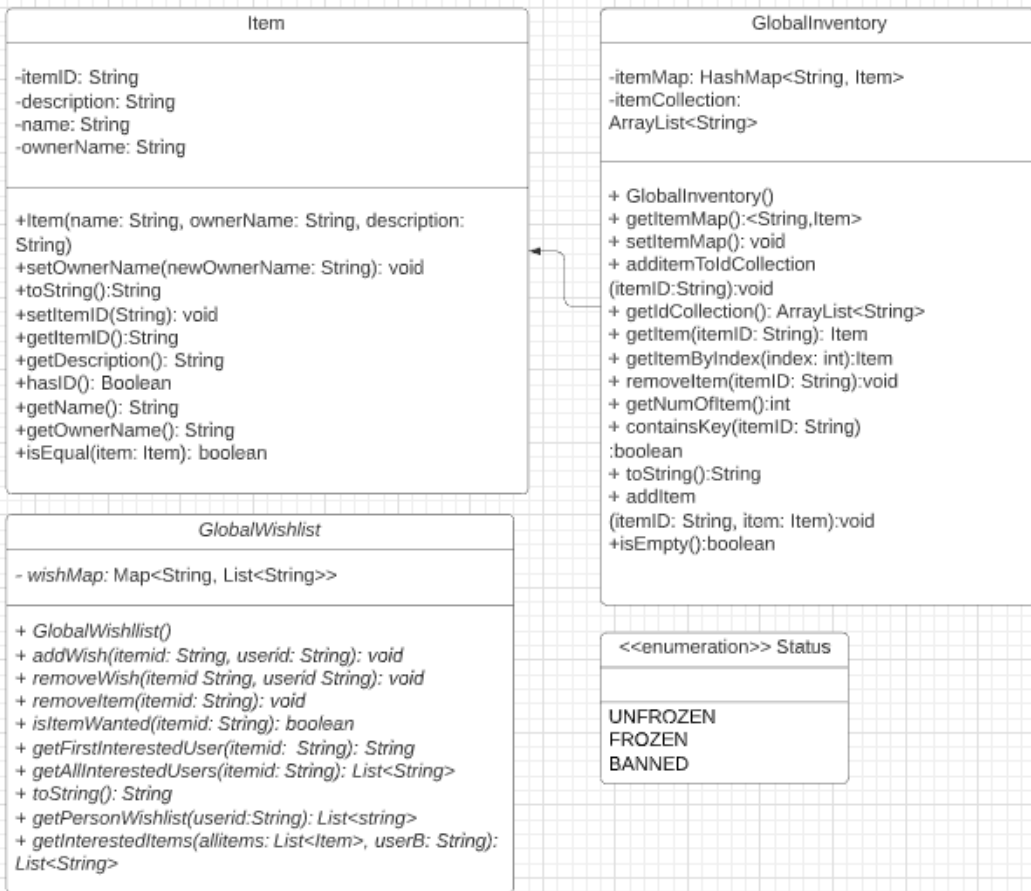
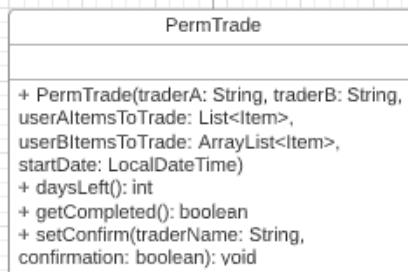
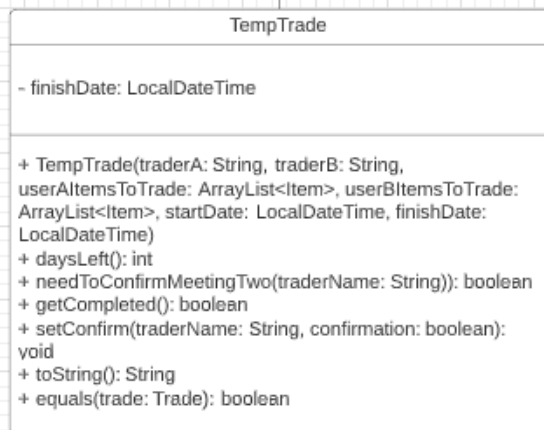
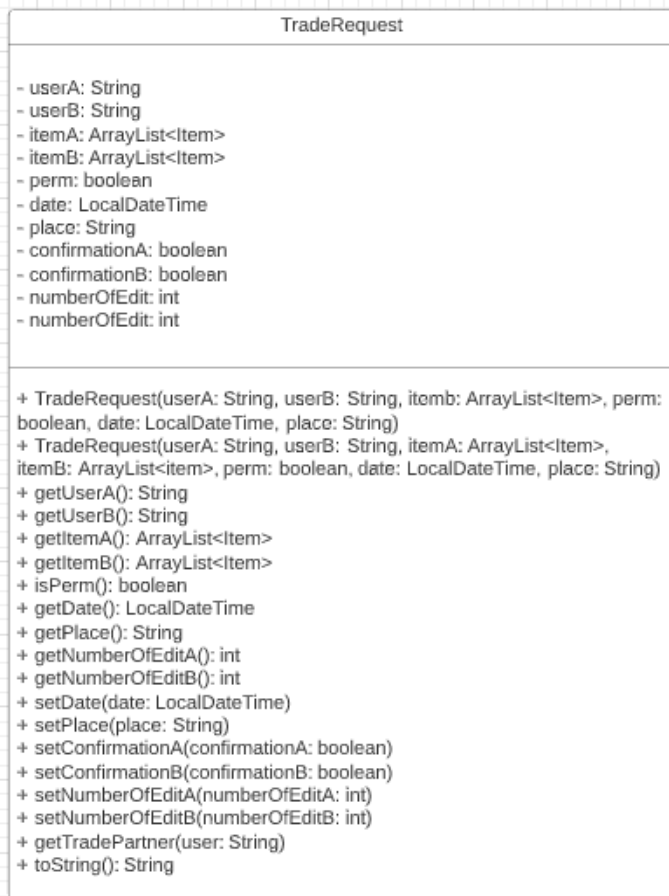


entities







<<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>

- status: Status

- tradePerWeek: int

- threshold: int

- limitOfIncompleteTrade: int

+User(username: String, password: String)

+getThreshold(): int

+getLimitOfIncompleteTrade(): int

+getTradePerWeek(): int

+getIsFrozen(): boolean

+getIsBanned(): boolean

+getMessages(): List<Message>

+setFrozen(): void

+setUnfrozen(): void

+setBanned(): void

+addMessages(message: Message): void

+setMessages(messages: List<Message>): void

+setThreshold(newThreshold: int): void

+accountInfo(): String

+setMessages(messages: List<Message>)

+setThreshold(newThreshold: int): void

+setLimitOfIncompleteTrade(newLimit: int): void

+setTradePerWeek(tradePerWeek: int)

+accountInfo(): String

Admin

+Admin(username: String,

password: String)

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
+ getCanTradesIgnoreBorrowLoans(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

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): ArrayList<String>
 + userWhoWants(allItems: List<Item>): List<String>
 + getPersonWishlist(userID: String): List<String>
 + getGlobalWishlistData(): GlobalWishlist
 + getInterestedItems(List<Item>: allItems, userB: String): List<String>

TradeManager

- tradeHistory: HashMap<String, ArrayList<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

GlobalInventoryManager

-globalInventory: GlobalInventory

+GlobalInventoryManager(globalInventory: GlobalInventory)
+ getItemFromGI(itemID: String):Item
+ getItemsFromGI(itemIDList: ArrayList<String>): List<Item>
+ IDGenerator(): String
+ addItemToHashMap(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

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>

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: HashMap<String,Admin>): void
+getAdminMap(): HashMap<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(): HashMap<String, ArrayList<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

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)
```

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
```

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)

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)

messagereplyGUI



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: URL, 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: URL, resources: ResourceBundle)

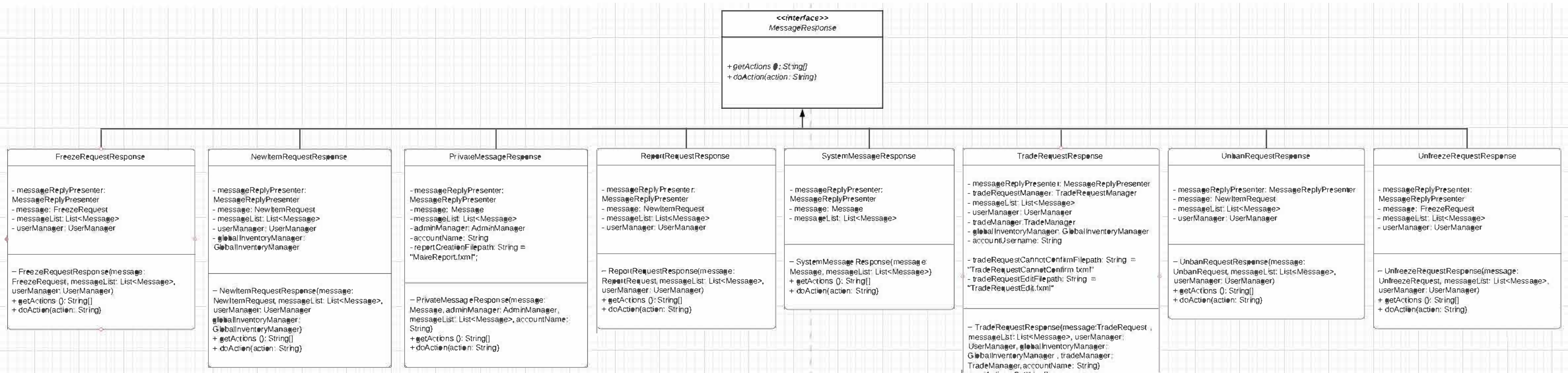
TradeRequestEditGUI

- 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: URL, 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



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

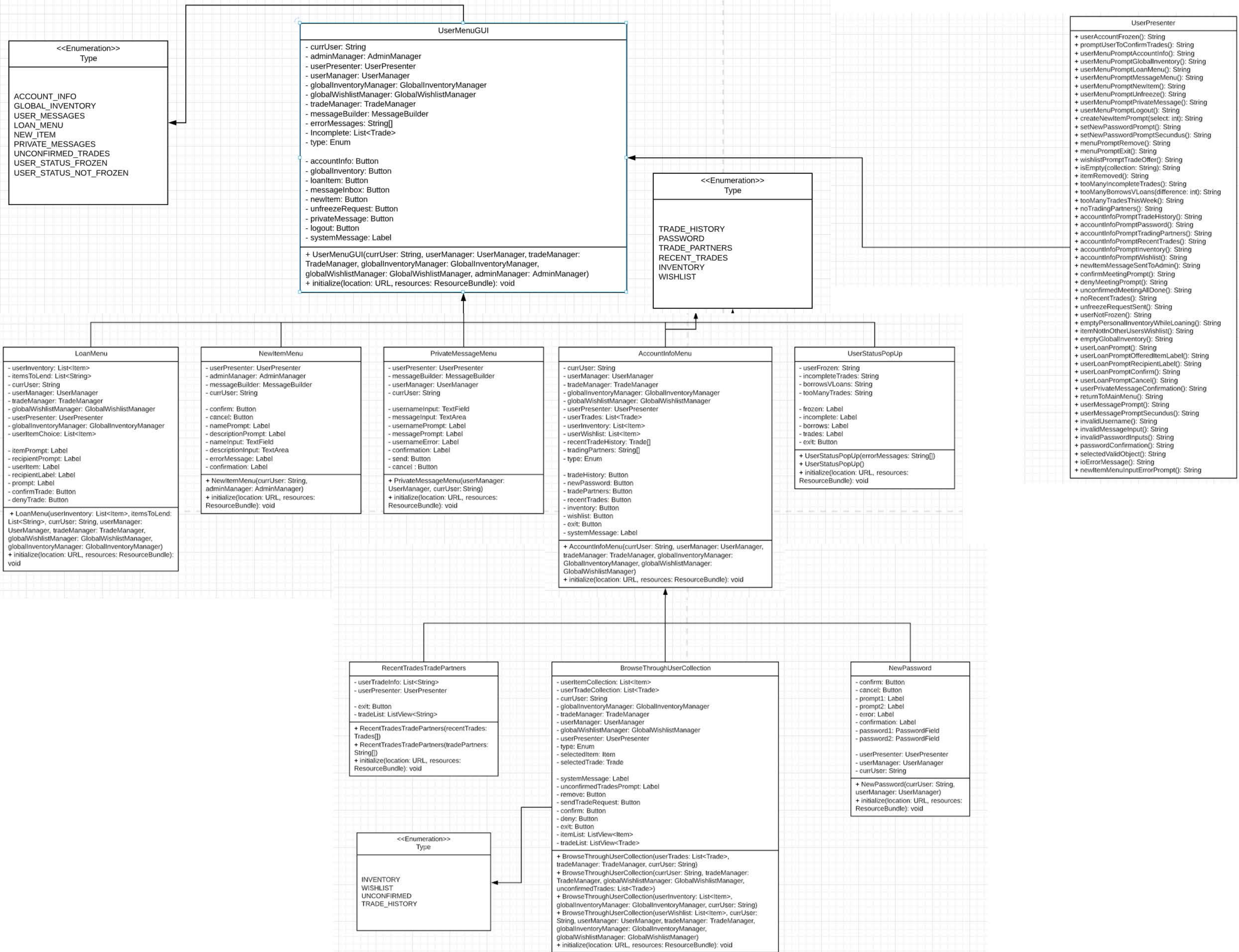
- + goToOtherScene(otherScene: String, selectedOption SelectedOption): void
- userLoginButtonPressed(): void
- userSignUpButtonPressed(): void
- adminLoginButtonPressed(): void
- programDemoButtonPressed(): void
- closeButtonIsPushed(): void
- + initialize(location: URL, resources: ResourceBundle): void
- deserialize(): void
- serialize(): void
- deleteFile(fileToDelete: String): boolean

LoginController

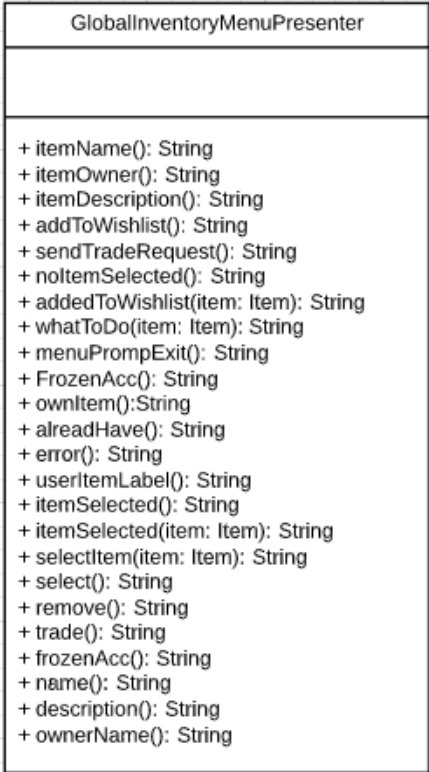
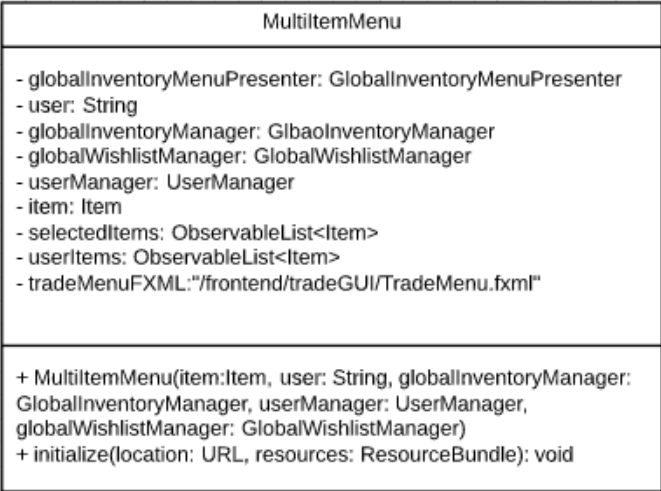
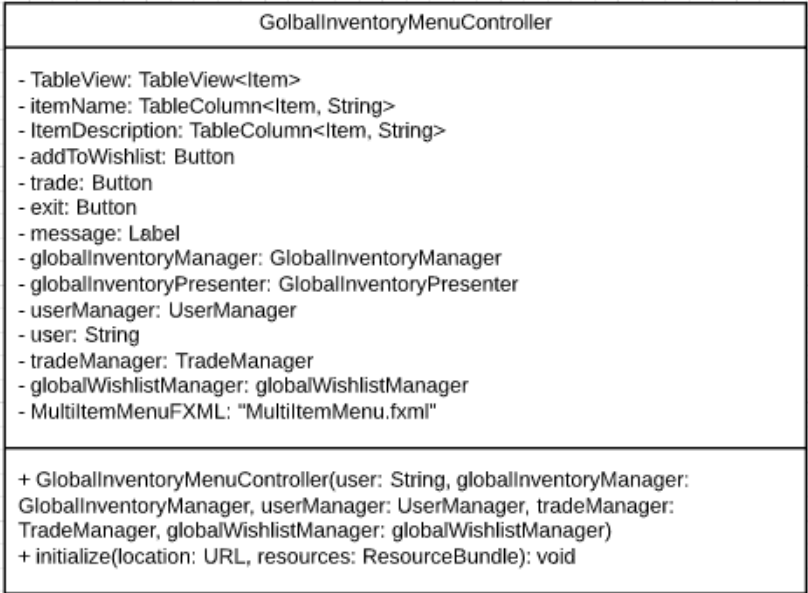
- username: Textfield
- password: TextField
- loginButton: Button
- exitButton: Button
- errorMessage: Label
- userSelectedOption: SelectedOption
- mainMenuPresenter: MainMenuPresenter
- userManager: UserManager
- tradeManager: TradeManager
- adminManager: AdminManager
- globalInventoryManager: GlobalInventoryManager
- 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



globalInventoryGUI



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): void

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

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

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>