

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)

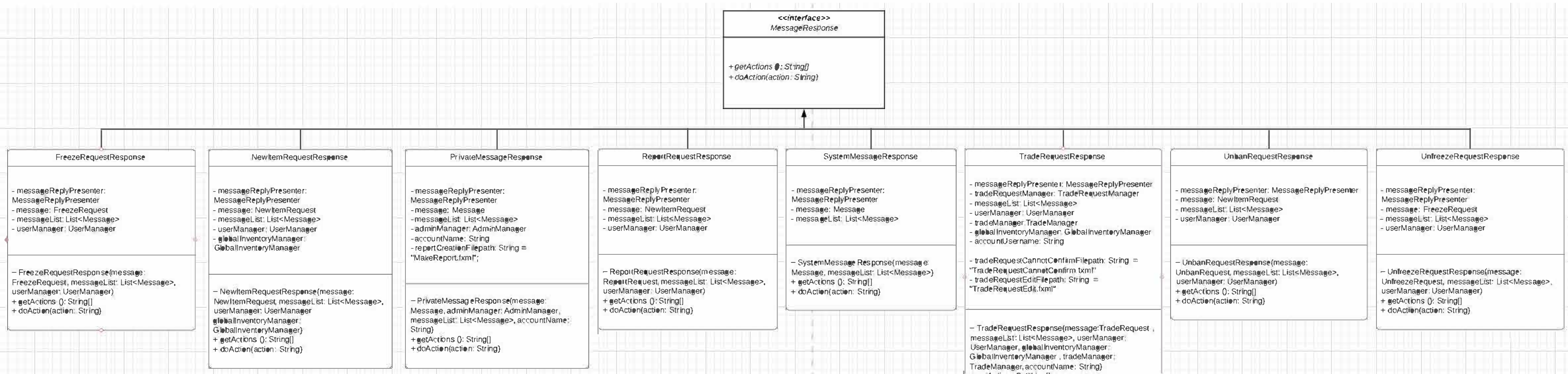
TradeRequestEditGUI

- 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

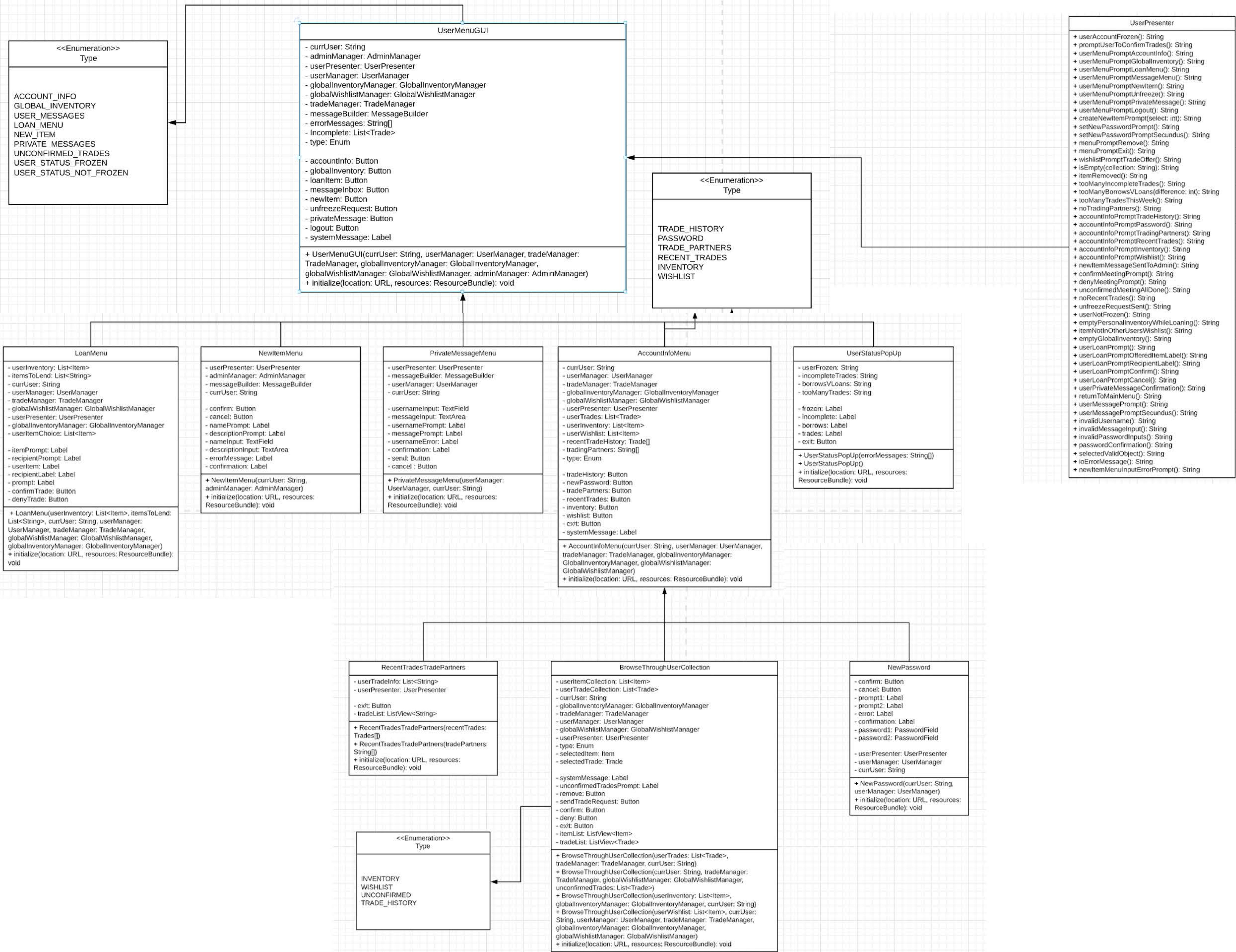
- + goToOtherScene(otherScene: String, selectedOption SelectedOption): void
- userLoginButtonPressed(): void
- userSignUpButtonPressed(): void
- adminLoginButtonPressed(): void
- programDemoButtonPressed(): void
- closeButtonIsPushed(): void
- + initialize(location: URL, resources: ResourceBundle): void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

LoginController

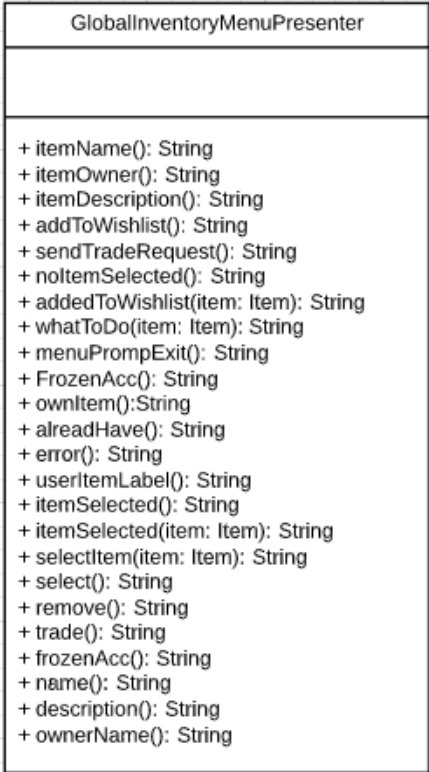
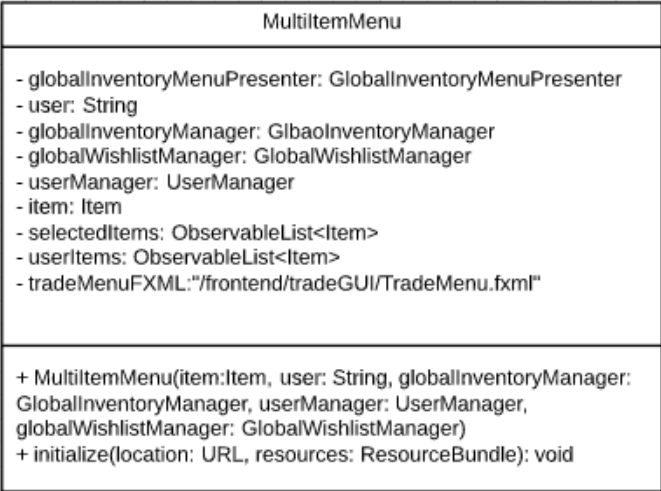
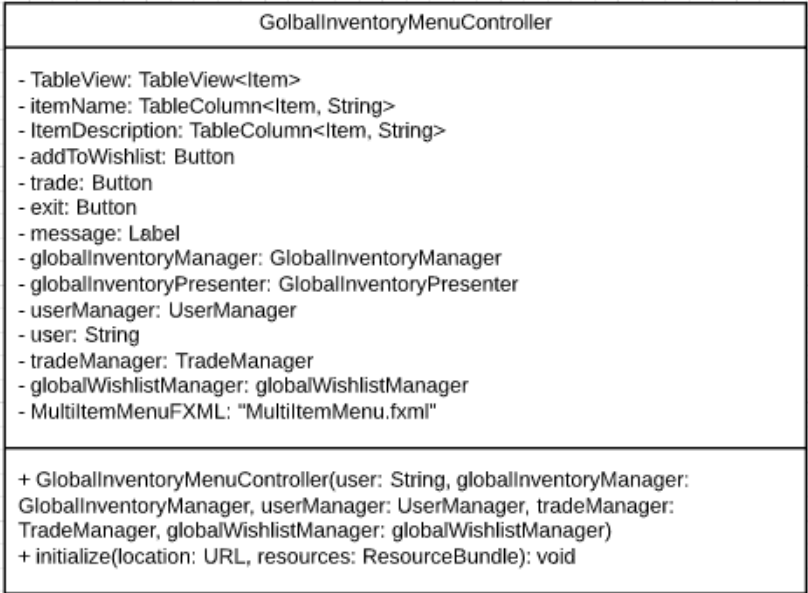
- username: Textfield
- password: TextField
- loginButton: Button
- exitButton: Button
- errorMessage: Label
- userSelectedOption: SelectedOption
- mainMenuPresenter: MainMenuPresenter
- userManager: UserManager
- tradeManager: TradeManager
- adminManager: AdminManager
- globalInventoryManager: GlobalInventoryManager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

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>