

- requestedItems : ArravList<Item> approvedItems : ArrayList<Item>

+ ItemManager() + ItemManager(requestedItems: ArrayList<Item>, approvedItems: ArravList<Item>) + getApprovedItems() : ArrayList<Item> + addApprovedItem(approvedItem: Item) : boolean + getRequestedItems() : ArrayList<Item> + setRequestedItems(requestedItems: ArrayList<Item>) : void + addRequestedItem(requestedItem: Item) : boolean

+ changeName(i: Item, name: String) : void + changeCategory(i: Item, category: String): void + changeDescription(i: Item, description: String) : void + changeOwner(i: Item, owner: Trader) : void + changeQualityRating(i: Item, qualityRating: int): void

+ requestToBorrow(receiver: Trader, location: String, tradeDate: LocalDate, item: Item) : int + requestToBorrow(receiver: Trader, location: String, tradeDate: LocalDate, item: Item, returnDate: LocalDate) : int + requestToExchange(receiver: Trader, location: String, tradeDate: LocalDate, item1: Item, item2: Item) : int + requestToExchange(receiver: Trader, location: String, tradeDate: LocalDate, item1: Item, item2: Item, returnDate: LocalDate): int - borrow(item: Item, initiator: Trader, receiver: Trader) : void exchange(item1: Item, item2: Item) : void + editTradeMeeting(date: LocalDate, location: String) : String + agreeToMeeting() : String + confirmTrade() : String completeTrade() : void · isValid() : boolean + getLimitOfTradesPerWeek() : int + setLimitOfTradesPerWeek(limit: int) : void + getMoreLendNeeded() : int + setMoreLendNeeded(num: int) : void + getMaxIncomplete() : int + setMaxIncomplete(num: int) : void

currentUser : Trader processedTrade : Trade

maxIncomplete: int

SignupSystem

+ SignupSystem(traderActions: TraderActions,

- checkUsername(username: String) : void

SignupPrompts

+ displayUserAvailable(value: boolean) : void

+ traderActions: TraderActions

+ adminActions: AdminActions

adminActions: AdminActions)

+ adminOrTrader() : void

+ createUsername() : void

+ createPassword() : void

+ createAccount() : void

+ adminOrTrader() : void

+ displayCreateUserName() : void

+ commandNotRecognized() : void

+ displayCreatePassword() : void

+ displayAccountSuccessful() : void

- stop() : void

- admin: boolean

- running: boolean

- username: String

- password: String

signupPrompts: SignupPrompts

moreLendNeeded; : int

- limitOfTradesPerWeek : int

+ setCurrentUser(currentUser: Trader) : void

+ setCurrentUser(currentUser: Trader, trade: Trade) : void

+ addToWantToBorrow(trader: Trader, item: Item) : void + removeFromWantToBorrow(trader: Trader, item: Item): void + addProposedItem(trader: Trader, item: Item): void + removeProposedItem(trader: Trader, item: Item): void + TradeManager(limitOfTradesPerWeek: int, moreLendNeeded: int, + browseltems() : ArrayList<Item> + browseltems(trader: Trader) : ArrayList<Item> + addTrader(trader: Trader) : void + checkUsername(username: String): boolean login(username: String, password: String) : Trader + getThreeRecentTrades(trader: Trader) : ArrayList<Trade> getMostRecentItems(trader: Trader) : ArrayList<Item> + printAccounts() : StringBuilder + getTraders() : ArrayList<Trader> + getOnGoingTrades(trader: Trader) : ArrayList<Trade>

traders : ArrayList<Trader>

+ TraderActions(traders: ArrayList<Trader>)

+ getTradersNeedingApproval() : ArrayList<Trader>

maxIncomplete : int

- menuOptions : ArrayList<String>

+ setOptions(options: String[]): void

+ commandNotRecognized() : void

+ displayMessage(messages: String[]) : void

- printItems(proposedItems: ArrayList<Item>) : StringBuilder

- adminActions : AdminActions

- traderActions : TraderActions

- toMainMenu : String (final)

+ setToMainMenu() : void

+ displayOptions() : void

- maxWeekly : int

+ AdminPrompts()

+ getListOfFlaggedAccounts() : ArrayList<Trader> + getFrozenTraders() : ArrayList<Trader>

+ getAllRequestToUnfreeze() : ArrayList<Trader>

+ mostFrequentTradingPartners(trader: Trader) : ArrayList<Trader>

AdminPrompts

- printAccounts(traders: ArrayList<Trader>, usernameOnly: boolean) : StringBuilder

admins : ArrayList<Admin> adminRequests : ArrayList<Admin> + AdminActions(admins: ArrayList<Admin>, adminRequests: ArrayList<Admin>) + getAdminRequests() : StringBuilder + getListOfRequestedAdmins() : ArrayList<Admin> + addAdminRequest(admin: Admin) : void findRequestByUserName(username: String) : Admin freezeAccount(trader: Trader) : boolean + approveAdmin(username: String, approved: boolean) : boolean + approveAllAdmins(approved: boolean) : boolean + checkUsername(username: String) : boolean unfreezeAccount(trader: Trader) : boolean + approveItem(trader: Trader, item: Item, approved: boolean) : boolean + approveAllItems(trader: Trader, approved: boolean) : boolean + changeLimit(newLimit: int) : void + checkCredentials(username: String, password: String) : Admin

LoginSystem # path : String # configReader : ConfigReader # configWriter : ConfigWriter # traderSystem : TraderSystem # adminSystem : AdminSystem # signupSystem : SignupSystem # prompts : LoginPrompts # traderActions : TraderActions # adminActions : AdminActions # itemManager : ItemManager # tradeManager : TradeManager # admin : Admin

+ run() : void LoginPrompts - prompts : ArrayList<String>

+ LoginSystem(path: String)

- String : final + LoginPrompts() + openingMessage() : String + invalidInput() : String + wrongUser() : String + wrongPassword() : String + next() : String + resetPrompts() : void

ConfigReader ~ traders : ArrayList<Trader> admins :ArrayList<Admin> ~ items :ArrayList<Items> ~ traderActions :TraderActions adminActions :AdminActions - itemManager : ItemManager ~ tradeManager :TradeManager ~ users : ArrayList<User> + ConfigReader(fileIn: String) - findTrader(username: String) : Trader - findItem(trader: Trader, ID: int) : Item - isInItems(id: int) : boolean - getInItems(id: int) : Item + getTraderActions() : TraderActions

+ getAdminActions() : AdminActions

+ getItemManager() : ItemManager

+ getTradeManager() : TradeManager

+ saveFile(fileName: String, ta: TraderActions, aa: AdminActions, tm: TradeManager) : void - formatAdmins(admins: List<Admin>, reqAdmins: List<Admin>) : String formatTrades(trades: List<Trade>) : String formatItems(items: List<Item>) : String - formatItem(i: Item) : String formatTrade(t: Trade) : String formatTradeInfo(t: Trade) : String - formatTrader(t: Trader) : String formatListOfTraders(traders: List<Trader>) : String - formatListOfItems(traders: List<Trader>) : String formatListOfTrades(traders: List<Trader>) : String formatAdmin(a: Admin) : String formatRequestedAdmin(a: Admin) : String formatWishlist(items: List<Item>, name: String) : String formatBorrowedItems(items: List<Item>, name: String) : String formatWishlistItem(i: Item, name: String ) : String formatBorrowedItem(i: Item, name: String) : String - formatLimits(limitOfTradesPerWeek: int, moreLendNeeded: int,

adminPrompts: AdminPrompts traderActions: TraderActions adminActions: AdminActions tradeManager: TradeManager running: boolean currentAdmin: Admin scanner: Scanner + AdminSystem(traderActions: TraderActions, adminActions: AdminActions, tradeManager: TradeManager) + setCurrentAdmin(admin: Admin) : void + run() : void + freezeMenu() : void freezeMenuHelper() : void - approveltemsMenu() : void itemSubMenu(trader: Trader) : void ⊦ adminApproval() : void confirmApproval(approved: boolean) : void approveOrReject() : Boolean ⊦ changeLimit() : void changeUserInfo() : void changeUsername() : int changePassword() : int + viewTraders() : void findTrader(user: String) : Trader restartViewTraders() : void + setToMainMenu() : void # update(): void # stop(): void

AdmimSystem

+ main(args: String[]) : void

+ displayFreezeMenu() : void + displayFreezeHelper(mainMenu: String) : void + displayFreezeOptions(i: int, accounts: ArrayList<Trader>): void + displayFreezeConfirmation(value: boolean, string: String) : void + displayItemMenu(traders: ArrayList<Trader>) : void + displayTraderProposedItems(items: ArrayList<Item>): void + displayAdminApproval(requests: StringBuilder) : void + displayApproveOrReject() : void + confirmApproval(approved: boolean) : void + displayTraderMenu() : void + displayRestartTrader() : void + displayAllTraders(traders: ArrayList<Trader>) : void + displayTrader(trader: Trader) : void + displayChangeLimitMenu() : void + displayChangeUserInfoMenu(): void + displayReturnToMainMenu() : void + displaySuccessMessage(successful: int, subject: String) : void + displayErrorMessage() : void + displayEnterNewMessage(subject: String) : void + displayThresholdOption(currentValue: int) : void

itemManager : ItemManager tradeManager : TradeManager - adminActions : AdminActions - currentTrader : Trader running : boolean - sc : Scanner + TraderSystem(currentTrader: Trader, traderActions: TraderActions, itemManager: ItemManager, tradeManager: TradeManager, adminActions: AdminActions) + run() : void - proposeItemToLend() : void + setCurrentTrader(trader: Trader) : void removeItemFromWantToLend(): void - removeItemFromWishlist(): void browselnventoryOfItems(): void proposeTradeStart(item: Item) : void - proposeOneWay(item: Item, temporary: boolean, tradeDate: LocalDate, location: String) : void - proposeTwoWay(item: Item, temporary: boolean, tradeDate: LocalDate, location: String) : void - showThreeMostRecentItemsTraded() : void - showTopThreeTradingPartners() : void requestUnfreeze() : void browseOnGoingTrades() : void editMeeting() : String # update(): void # stop(): void

TraderSystem

traderPrompts : TraderPrompts

- traderActions : TraderActions

TraderPrompts # mainMenuPrompts : ArrayList<String> # proposeItemPrompts : ArrayList<String> - instruction : String (final?) ⊦ TraderPrompts() setUpMainMenuPrompts(): void setUpProposeItemPrompts(): void + getMainMenuPrompts() : ArrayList<String> getProposeItemPrompts() : ArrayList<String> + displayMainMenu() : void + displayTraderItems(trader: Trader) : void + displayTraderItemsTwo(traer): Trader) : void displayItems(itemsList: ArrayList<Item>) : void + browseInventory(items: ArrayList<Item>) : void + viewItem(item: Item) : void + viewListOfRecentItems(items: ArrayList<Item>) : void + viewListOfTradingPartners(partners: ArrayList<Trader>) : void requestUnfreeze(): void + browseOnGoingTrades(onGoingTrades: ArrayList<Trade>) : void + incorrectSelection() : void + displayTradeOptions() : void displayString(string: String) : void + returnToMain() : void + displayTradeProcess(i: int) : void