Group_0025 UML Diagram

Main

+ main(args: String[]): void

LoginSystem

− passwordLengthLimit : int
− invalidString : String[]
− lp : LoginPresenter
− ug : UserGateway
− ag : AdminGateway
− ig : ItemGateway
− thrG : ThresholdGateway
− um : UserManager
− adm : AdminManager
− im : ItemManager
− us : UserSystem
− ad : AdminSystem
− input : Scanner

+ LoginSystem()
+ run() : void
− loginMenu() : boolean
− adminLogin() : String
− userLogin() : String
− registerAccount() : void
− checkUsernameAvailability() : String
− containsInvalidString(String anyString) : boolean

LoginPresenter

− cancelString : String

+ LoginPresenter()
+ LoginPresenter(cancelString : String)
+ getCancelString() : String
+ welcomeMessage() : void
+ adminMessage() : void
+ userMessage() : void
+ chooseOptionPrompt() : void
+ options() : void
+ exit() : void
+ loginOptions() : void
+ accountOptions() : void
+ adminLoginOptions() : void
+ userLoginOptions() : void
+ closingMessage() : void
+ adminLoginMessage() : void
+ userLoginMessage() : void
+ enterUsernamePrompt() : void
+ usernameNotFoundPrompt() : void
+ welcomeMessage(username : String) : void
+ enterPasswordPrompt() : void
+ incorrectPasswordPrompt() : void
+ invalidInputMessage() : void
+ welcomeNewUserMessage(username : String) : void
+ userRegistrationMessage() : void
+ newUsernamePrompt() : void
+ newPasswordPrompt(length : int) : void
+ invalidNewPasswordMessage() : void
+ proceedRegistrationPrompt() : void
+ registrationResultMessage(isSuccessful : boolean) : void
+ usernameUnavailablePrompt() : void
+ changeUsernamePrompt() : void
+ currentPasswordPrompt() : void
+ newPasswordPrompt() : void

AdminGateway

− userInfo : HashMap<String, ArrayList<ArrayList<String>>>

+ UserGateway()
+ getUserInfoCSV() : ArrayList<User>
+ convertToUser(userList : HashMap<String, ArrayList<ArrayList<String>>>) : ArrayList<User>
+ updateUserInfoCsv(um : UserManager) : void

<<Abstract>> *Account*

− id : String
− username : String
− password : String

+ Account(username : String, password : String)
+ getId() : String
+ setId(accountId : String) : void
+ getUsername() : String
+ getPassword() : String
+ setUsername(username : String) : void
+ setPassword(password : String) : void
+ *toString()* : String

Admin

+ Admin(username : String, password : String)
+ toString() : String

AdminManager

- adminList: ArrayList<Admin>

+ AdminManager(allAdmin : ArrayList<Admin>)
+ getAdminList(): ArrayList<Admin>
+ addAdmin(newUsername: String, newPassword: String): boolean
+ availableUsername(username: String): boolean
+ incorrectPassword(username: String, pass: String): boolean
- adminByadminId(adminId: String): Admin
- adminByUsername(username: String): Admin
+ usernameByAdminId(userId String): String
+ adminIdByUsername(username String): String
+ getUsernames(): ArrayList<String>
+ changeUsername(adminId: String, username: String): void
+ changePassword(adminId: String, password: String): void
+ toString(): String

AdminSystem

- ig: ItemGateway
- ag: AdminGateway
- ug: UserGateway
- tg: ThresholdGateway
- adm: AdminManager
- um: UserManager
- im: ItemManager
- ap: AdminPresenter
- currentItemId: String
- input: Scanner
- passwordLength: int

+ AdminSystem(adm: AdminManager, um: UserManager)
+ run(adminId : String) : void
- manageProfileControls(adminId: String): void
- manageUsersControls(): void
- flagControls(): void
- statusControls(): void
- profilesByStatus(status: UserStatus): void
- manageAdminControls(): void
- manageItemsControls(): void
- itemOptions(): void
- retrieveItemId(itemsInfo: String[], num: int): String
- isBadInput(input: String, numRange: int): boolean
- allItemsOptions(): void
- removeFromSystem(): void
- requestedItemOptions(): void
- approveItem(): void
- manageThresholdControls(): void
- generatePassword(): String
- registerAdmin(): void
- checkUsernameAvailablity(): String
- containsInvalidString(anyString: String): boolean
- changeUsername(adminId: String): void
- changePassword(adminId: String): void
- editThreshold(thresholdIndex: int, minValue): void

User

− inventory : ArrayList<String>
− wishlist : ArrayList<String>
− status : UserStatus

+ User(username : String, password : String)
+ getInventory() : ArrayList<String>
+ getWishlist() : ArrayList<String>
+ getStatus() : UserStatus
+ setInventory(inventory : ArrayList<String>) : void
+ setWishlist(wishlist : ArrayList<String>) : void
+ addToInventory(itemId : String) : boolean
+ removeFromInventory(itemId : String) : boolean
+ addToWishlist(itemId : String) : boolean
+ removeFromWishlist(itemId : String) : boolean
+ setStatus(status : UserStatus) : void
+ toString() : String

UserSystem

− passwordLengthLimit : int
− ug : UserGateway
− ag : AdminGateway
− up : UserPresenter
− um : UserManager
− adm : AdminManager
− im : ItemManager
− tm : TradeManager
− input : Scanner

+ UserSystem(um : UserManager, adm : AdminManager, im : ItemManager)
+ run(userId : String) : void
− displayUserMenu(userId : String) : void
− changeUsername(userId : String)) : String
− changePassword(userId : String)) : String
− viewProfile(userId : String)) : void
− requestUnfreeze(userId : String)) : void

AdminGateway

+ adminInfo: HashMap<String, ArrayList<String>>

+ AdminGateway()
+ getAdminInfoCsv(): ArrayList<Admin>
+ convertToAdmin(adminList: HashMap<String, ArrayList<String>>): ArrayList<Admin>
+ updateAdminInfoCsv(adm: AdminManager)

AdminPresenter

- adm: AdminManager
- um: UserManager

+ AdminPresenter(adm: AdminManager, um: UserManager)
+ getCancelString(): String
+ chooseOptionPrompt(): void
+ invalidInputMessage(): void
+ updateSuccessMessage(String thing): void
+ noUpdateMsg(): void
+ currentPasswordPrompt(): void
+ incorrectPasswordPrompt(): void
+ newPasswordPrompt(): void
+ invalidNewPasswordMessage(): void
+ changeUsernamePrompt(String username): void
+ adminMainMenu(): void
+ adminManageProfileMenu(): void
+ adminManageUserMenu(): void
+ flagMenu(): void
+ adminManageAdminMenu(): void
+ adminManageItemMenu(): void
+ adminManageThresholdMenu(): void
+ viewAdmin(): void
+ adminUsernameMessage(username: String): void
+ registrationResultMsg(isSuccessful: boolean): void
+ adminRegisterPrompt(): void
+ generatePassMsg(username: String, pass: String): void
+ usernameUnavailablePrompt(): void
+ proceedRegistrationPrompt(): void
+ viewUsers(): void
+ userProfile(userId: String): void
+ usernameNotFound(): void
+ frozenUsers(): void
+ requestedUsers(): void
+ flaggedUsers(): void
+ statusPrompt(String status): void
+ thresholdEditPrompt(String thresholdName): void
+ badThresholdMessage(int minValue): void
+ unacceptableThresholdMessage(int minValue): void
+ thresholdEditSuccessMessage(): void
+ viewAllItems(ItemManager im): void
+ singleItemMenu(): void
+ itemRemovalSuccess(): void
+ viewAllRequestedItems(ItemManager im): void
+ requestedItemMenu(): void

UserManager

− userIdToUser : HashMap<String, User>

+ UserManager(allUsers : ArrayList<User>)
+ getUserIdToUser() : HashMap<String, User>
+ addUser(newUsername : String, newPassword : String) : boolean
− findUserByUserId(userId : String) : User
− findUserByUsername(username : String) : User
+ usernameByUserId(userId : String) : String
+ userIdByUsername(username : String) : String
+ userIdsByStatus(status : UserStatus) : ArrayList<String>
+ statusByUserId(userId : String) : UserStatus
+ inventoryByUserId(userId : String) : ArrayList<String>
+ itemIdInInventory(userId : String, itemId : String) : boolean
+ itemIdInWishlist(userId : String, itemId : String) : boolean
+ wishlistByUserId(userId : String) : ArrayList<String>
+ setInventory(userId : String, inventory : ArrayList<String>) : void
+ setWishlist(userId : String, wishlist : ArrayList<String>) : void
+ addToInventory(userId : String, itemId : String) : boolean
+ removeFromInventory(userId : String, itemId : String) : boolean
+ addToWishlist(userId : String, itemId : String) : boolean
+ removeFromWishlist(userId : String, itemId : String) : boolean
+ changeUsername(userId : String, username : String) : void
+ changePassword(userId : String, password : String) : void
+ changeStatus(userId : String, UserStatus status) : void
+ getUsernames(): ArrayList<String>
+ checkUsername(username : String) : boolean
+ checkPassword(username : String, password : String) : boolean
+ unfreezeRequest(userId : String) : boolean
+ toStringByAccountId(userId : String) : String
+ toString() : String
+ findUserByItemInventory(itemId : String) : String
+ findUserByItemWishlist(itemId : String) : String

UserPresenter

− cancelString : String

+ UserPresenter()
+ UserPresenter(cancelString : String)
+ getCancelString() : String
+ chooseOptionPrompt() : void
+ invalidInputMessage() : void
+ incorrectPasswordPrompt() : void
+ mainMenuOptions(isFrozen : boolean) : void
+ userMenuOptions() : void
+ yourCurrentListMessage(isInventoryList : boolean) : void
+ changeUsernamePrompt(username : String) : void
+ usernameUnavailablePrompt() : void
+ updateSuccessMessage(thing : String) : void
+ currentPasswordPrompt() : void
+ newPasswordPrompt() : void
+ invalidNewPasswordMessage() : void
+ frozenWarningMessage() : void
+ frozenReminderMessage() : void
+ unfreezeRequestSentMessage() : void
+ unfreezeRequestAlreadySentMessage() : void
+ displayProfile(userInfo : ArrayList<String>) : void

UserGateway

− userInfo : HashMap<String, ArrayList<ArrayList<String>>>

+ UserGateway()
+ getUserInfoCSV() : ArrayList<User>
+ convertToUser(userList : HashMap<String, ArrayList<ArrayList<String>>>) : ArrayList<User>
+ updateUserInfoCsv(um : UserManager) : void

Trade UML

DoActionTradeSystem

- tm:TradeManager
- im:ItemManager
- um:UserManager
- mm:MeetingManager
- ms:MeetingSystem
- tp:TradePresenter = new TradePresenter()
- userId:String
- tg:TradeGateway = new TradeGateway()
- ig:ItemGateway = new ItemGateway()
- ug:UserGateway = new UserGateway()
- mg:MeetingGateway = new MeetingGateway()
- invalidStrings: String[] = new String[] {"","/"}

- DoActionTradeSystem(userId:String, tm:TradeManager, im:ItemManager, um:UserManager, mm:MeetingManager)
- helperActionToTrade(listTrade: ArrayList<Trade>):void
- helperSwitchConfirmOrMeeting(tradeIdInput: String):void
- helperSwithcAcceptRejectTrade(tradeIdInput: String):void
- eligibleToAcceptOrReject(tradeIdInput: String):boolean
- helperInitiateMeeting(tradeIdInput:String):void
- helperPromptPlaceTime():Pair<String, LocalDate>
- helperPromptTime():LocalDate
- helperConfirmTrade(tradeIdInput: String):void
- eligibleToConfirmTrade(tradeIdInpu: String):boolean
- eligibleToViewMeeting(tradeIdInput: String):boolean
- helperCompleteTrade(tradeIdInput: String):void
- helperSwapInventories(tradeId:String):void
- promptTradeId(input:Scanner):String
- validTradeId(tradeId:String):boolean
- getUserId():String
- validIncompleteTrade(userId:String):boolean
- contansInvalidString(anyString: String):boolean
- updateToCsv():void

RequestTradeSystem

- userId:String
- tp:TradePresenter = new TradePresenter()
- tm:TradeManager
- im ItemManager
- um:UserManager
- ig:ItemGateway
- tg:TradeGateway
- ug:UserGateway

- RequestTradeSystem(userId: String, tm: TradeManager, im: ItemManager, um: UserManager)
- helperRequestTrade():void
- helperRequestTemPerItem(userId2:String, all:ArrayList<String>):void
- helperPermanentOrTemporary(userId2:String, all:ArrayList<String>, duration1:int):void
- helperRequestOneItem():ArrayList<String>
- helperRequestTwoItem():ArrayList<String>
- getUsernameInput(input:Scanne):String
- getItemInput(input:Scannr, userId2:String):String
- validBorrowDiff(userId:String):boolean
- getUserId():String

TradeSystem

- tm:TradeManager
- um:UserManager
- im:ItemManager
- mm:MeetingManager
- tp:TradePresenter = new TradePresenter()
- dts:DisplayTradeSystem
- rts:RequestTradeSystem
- dats:DoActionTradeSystem
- userId:String
- ig:ItemGateway = new ItemGateway()
- tg:TradeGateway = new TradeGateway()
- ug:UserGateway = new UserGateway()

+ TradeSystem(useId:String, um:UserManager)
+ run():void
- checkUserStatus(s:String):void
- helperBadSwitch(s:String):void
- helperGoodSwitch(s:String):void
- getUserId():String
- validIncompleteTrade(userId:String):boolean

DisplayTradeSystem

- tm:TradeManager
- im:ItemManager
- um:UserManager
- tp:TradePresenter = new TradePresenter()
- userId:String

+DisplayTradeSystem(userId:String, tm:TradeManager, im:ItemManager, um: UserManager)
- helperDisplaySelection();void
- helperDisplayTrade(trade: ArrayList<Trade>, presenterNotification:String, arrayList:ArrayList<String>):void
- helperDisplayRecent()
- displayRecentBorrowItems():void
- displayRecentLendItems():void
- displayRecentTwoWayItems():void
- displayTopTraders():void
- getUserId():String
- getIntInput():int

TradeManager

- cancelledThreshold:int
- weeklyTradeLimit:int
- borrowDiff:int
- tradeIdToTrade: HashMap<String, Trade> = new HashMap<>()

+TradeManager(listThresholds:List<Integer>, listTrade(List<Trade>)
+ getCancelledThreshold():int
+ getWeeklyTradeLimit():int
+ getBorrowDiff():int
+ getTradeIdToTrade(): HashMap<String, Trade)
+ setCancelledThreshold(cancelledThreshold:int):void
+ setWeeklyTradeLimit(weeklyTradeLimit:int):void
+ setBorrowDiff(borrowDiff:int):void
+ addTrade(tradeId:String, trade:Trade):void
+ requestTrade(userId1:String, userId2:String, itemIds:List<String>, duration:int):void
+ findTrade(tradeId:String):Trade
+ rejectTrade(tradeId:String):void
+ acceptTrade(tradeId:String):void
+ meetingConfiremd(tradeId:String):void
+ confirmTrade(tradeId:String, userId:String):void
+ completeTrade(tradeId:String):void
+ cancelTrade(tradeId:String):void
+ checkBorrowDiff(userId:String):boolean
+ tradesInOneWeek(userId:String):List<String>
- helperTradesInOneWeek(trade:Trade, userId:String):boolean
+ checkTradeLimit(userId:String):boolean
+ checkIncompleteTrade(userId:String):boolean
+ getAllStatusTradeByUserId(userId:String, tradeStatus:TradeStatus):ArrayList<String>
+ statusConverter(tradeStatus:String): Trade.TradeStatus
+ getAllCompletedBorrowTradeByUserId(userId:Strin): ArrayList<String>
+ getAllCompletedLendTradeByUserId(userId:String): ArrayList<String>
+ getAllLendTradeIdByUserId(userId:String): ArrayList<String>
+ getAllBorrowTradeIdByUserId(userId:String): ArrayList<String>
+ getAllTradeIdByUserId(userId:Strng): ArrayList<String>
+ getAllTwoWayTradeIdByUserId(userId:String): ArrayList<String>
+ getAllOneWayTradeId():HashMap<String, Trade>
+ getAllTwoWayTradeId():HashMap<String, Trade>
+ getRecentBorrowedByUserId(userId:String, num:iny): ArrayList<String>
+ getRecentItemsLendByUserId(userId:String, num:int): ArrayList<String>
+ getRecentItemsInTwoWayByUserId(userId:String, num:int):ArrayList<String>
+ getTopTradersByUserId(userId:String, num:int):ArrayList<String>
- helperChoosingTopTraders(num:int, returnAl,:ArrayList<String> userToNumTrades:HashMap<String,Integer>):void
- helperGetTopTraders(userId:String, al:HashMap<String,Integer>):HashMap<String, Integer>
+ createNewTrade(userId1:String, userId2:String, itemIds):ArrayList<String>:String
+ getTradeStatusByTradeId(tradeId:String):Trade.TradeStatus
+ getReceiverByTradeId(tradeId:String):String
+ getRequesterByTradeId(tradeId:String):String
+ getItemIdsByTradeId(tradeId:String):List<String>
+ getDurationByTradeId(tradeId:String):int
+ setMeetingIdByTradeId(tradeId:Strin, meetingId:String):void
+ getMeetingIdByTradeId(tradeId:String):String
+ createReverseTrade(tradeId:String):String

TradePresenter

- validInput:ArrayList<String> = new ArrayLIst<>()

+ TradePresenter()
+ getValidInput():ArrayList<String>
+ getSelectionMenu(s:String):void
+ getAllTypeOptions():void
+ getTypeTradeNotification(s:String):void
+ getUserNameNotification():void
+ getItemLIstInfo(itemList: ArrayList<Item>):void
+ getTradeListInfo(tradeList:ArrayList<Trade>):void
+ getUsernameTopTraders(usernameList: ArrayList<String>):void
+ getManagementNotification():void
+ getItemNotification():void
+ getItemNotification2():void
+ getTradeNotification():void
+ getNotFound(s:String):void
+ getTradeSelectionNotification(s1:String, s2:String):void
+ getRecentSelectionNotification(s1:String, s2:String s3:String, s4:String):void
+ getAcceptRejectNotification():void
+ getSwitchConfirmOrMeetingNotification():void
+ getConfirmNotification():void
+ successfulRequestTradeMade(s:String):void
+ successfulActionTrade(s:String):void
+ successfulInitiateMeeting(time:LocalDate, place :String):void
+ promptNumForTopTraders():void
+ promptNumRecentItems(s:String):void
+ noTradeActionAvailable():void
+ violateBorrowDiff():void
+ violateBorrowDiff():void
+ violateIncompleteTrade():void
+ violateTradeLimit():void
+ invalidSelection():void
+ invalidInput():void
+ invalidDate():void
+ promptMeetingPlace():void
+ promptMeetingTime():void
+ getGoBackMenuNotification():void
+ getMeetingNotification(s:String):void
+ getMeetingInfo(meeting:Meeting):void
+ meetingCancelledNotification():void

Trade

- tradeId:String
- status:TradeStatus
- duration:int
- meetingId:String
- requester:String
- receiver: String
- itemIds:List<String>
- dateCreated:LocalDate
+TradeStatus:enum

+ Trade(userId1: String, userId2: String, itemIds: List<String>)
+ toString():String
+ getTradeId():String
+ getStatus(): TradeStatus
+ getDuration():int
+ getRequester():String
+ getReceiver():String
+ getItemIds():List<String>
+ getMeetingId():String
+ getDateCreated():LocalDate
+ setStatus(status:TradeStatus):void
+ setTradeId(tradeId:String):void
+ setDateCreated(dateCreated:LocalDate):void
+ setDuration(duration:int);void
+ setRequester(userId1:String):void
+ setReceiver(userId2:String):void
+ setItemIds(itemIds:List<String>):void
+ setMeetingId(meetingId:String):void
+ isOneWay():boolean

TradeComparator

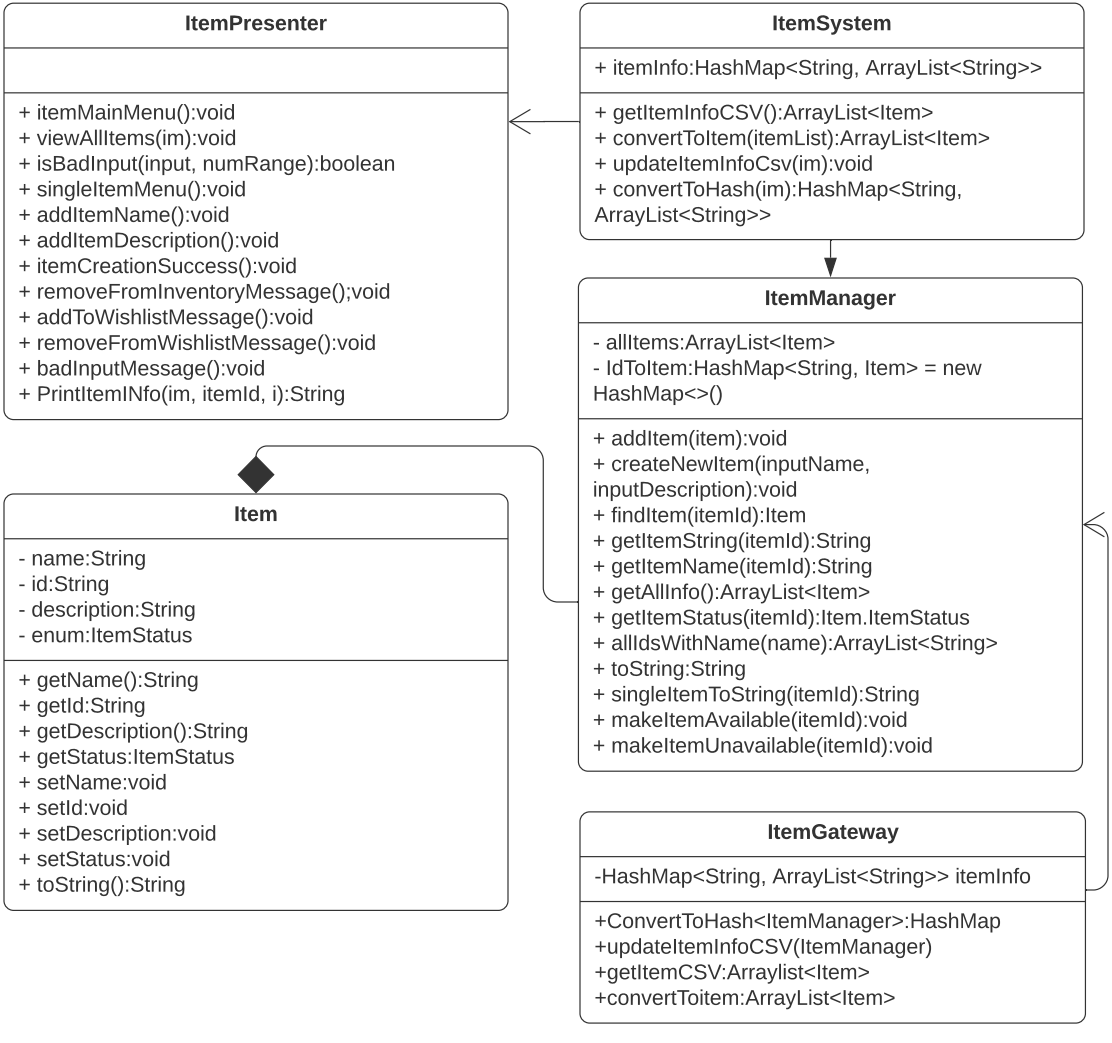
+ compare(t1, t2):int

TradeGateway

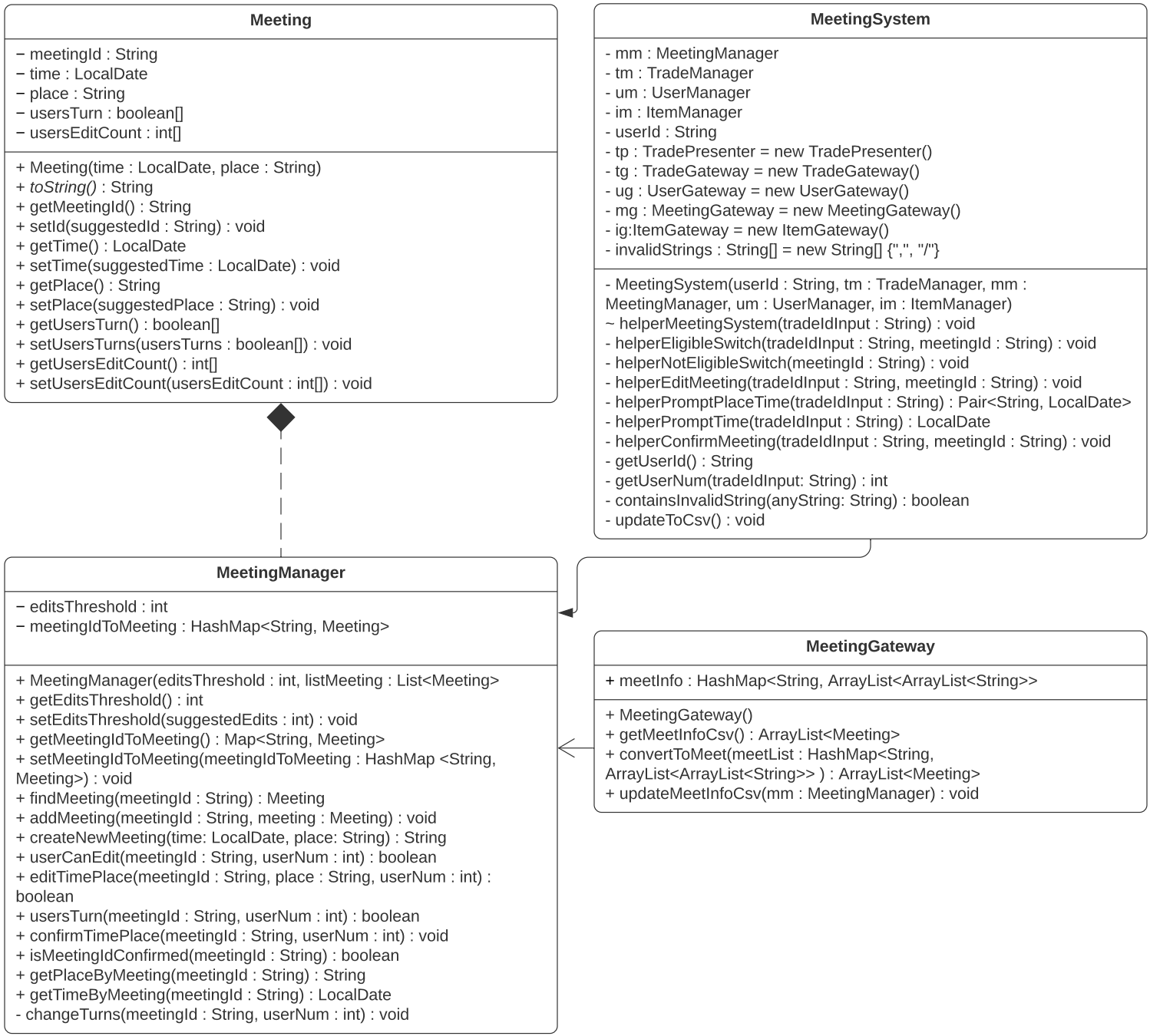
+ tradeInfo:HashMap<String, ArrayList<ArrayList<String>>>

+ TradeGateway()
+ getThresholdInfoCSV():List<Integer>
+ getTradeInfoCsv():ArrayList<Trade>
+ convertToTrade(tradeList:HashMap<String, ArrayList<ArrayList<String>>>):ArrayList<Trade>
+ updateTradeInfoCsv(tm:TradeManager):void

Item UML


ItemPresenter
+ itemMainMenu():void
+ viewAllItems(im):void
+ isBadInput(input, numRange):boolean
+ singleItemMenu():void
+ addItemName():void
+ addItemDescription():void
+ itemCreationSuccess():void
+ removeFromInventoryMessage();void
+ addToWishlistMessage():void
+ removeFromWishlistMessage():void
+ badInputMessage():void
+ PrintItemINfo(im, itemId, i):String
ItemSystem
+ itemInfo:HashMap<String, ArrayList<String>>
+ getItemInfoCSV():ArrayList<Item>
+ convertToItem(itemList):ArrayList<Item>
+ updateItemInfoCsv(im):void
+ convertToHash(im):HashMap<String, ArrayList<String>>
ItemManager
- allItems:ArrayList<Item>
- IdToItem:HashMap<String, Item> = new HashMap<>()
+ addItem(item):void
+ createNewItem(inputName, inputDescription):void
+ findItem(itemId):Item
+ getItemString(itemId):String
+ getItemName(itemId):String
+ getAllInfo():ArrayList<Item>
+ getItemStatus(itemId):Item.ItemStatus
+ allIdsWithName(name):ArrayList<String>
+ toString:String
+ singleItemToString(itemId):String
+ makeItemAvailable(itemId):void
+ makeItemUnavailable(itemId):void
Item
- name:String
- id:String
- description:String
- enum:ItemStatus
+ getName():String
+ getId:String
+ getDescription():String
+ getStatus:ItemStatus
+ setName:void
+ setId:void
+ setDescription:void
+ setStatus:void
+ toString():String
ItemGateway
-HashMap<String, ArrayList<String>> itemInfo
+ConvertToHash<ItemManager>:HashMap
+updateItemInfoCSV(ItemManager)
+getItemCSV:Arraylist<Item>
+convertToitem:ArrayList<Item>

Meeting UML


Meeting
− meetingId : String
− time : LocalDate
− place : String
− usersTurn : boolean[]
− usersEditCount : int[]
+ Meeting(time : LocalDate, place : String)
+ toString() : String
+ getMeetingId() : String
+ setId(suggestedId : String) : void
+ getTime() : LocalDate
+ setTime(suggestedTime : LocalDate) : void
+ getPlace() : String
+ setPlace(suggestedPlace : String) : void
+ getUsersTurn() : boolean[]
+ setUsersTurns(usersTurns : boolean[]) : void
+ getUsersEditCount() : int[]
+ setUsersEditCount(usersEditCount : int[]) : void
MeetingSystem
- mm : MeetingManager
- tm : TradeManager
- um : UserManager
- im : ItemManager
- userId : String
- tp : TradePresenter = new TradePresenter()
- tg : TradeGateway = new TradeGateway()
- ug : UserGateway = new UserGateway()
- mg : MeetingGateway = new MeetingGateway()
- ig:ItemGateway = new ItemGateway()
- invalidStrings : String[] = new String[] {",", "/"}
- MeetingSystem(userId : String, tm : TradeManager, mm : MeetingManager, um : UserManager, im : ItemManager)
~ helperMeetingSystem(tradeIdInput : String) : void
- helperEligibleSwitch(tradeIdInput : String, meetingId : String) : void
- helperNotEligibleSwitch(meetingId : String) : void
- helperEditMeeting(tradeIdInput : String, meetingId : String) : void
- helperPromptPlaceTime(tradeIdInput : String) : Pair<String, LocalDate>
- helperPromptTime(tradeIdInput : String) : LocalDate
- helperConfirmMeeting(tradeIdInput : String, meetingId : String) : void
- getUserId() : String
- getUserNum(tradeIdInput: String) : int
- containsInvalidString(anyString: String) : boolean
- updateToCsv() : void
MeetingManager
− editsThreshold : int
− meetingIdToMeeting : HashMap<String, Meeting>
+ MeetingManager(editsThreshold : int, listMeeting : List<Meeting>
+ getEditsThreshold() : int
+ setEditsThreshold(suggestedEdits : int) : void
+ getMeetingIdToMeeting() : Map<String, Meeting>
+ setMeetingIdToMeeting(meetingIdToMeeting : HashMap <String, Meeting>) : void
+ findMeeting(meetingId : String) : Meeting
+ addMeeting(meetingId : String, meeting : Meeting) : void
+ createNewMeeting(time: LocalDate, place: String) : String
+ userCanEdit(meetingId : String, userNum : int) : boolean
+ editTimePlace(meetingId : String, place : String, userNum : int) : boolean
+ usersTurn(meetingId : String, userNum : int) : boolean
+ confirmTimePlace(meetingId : String, userNum : int) : void
+ isMeetingIdConfirmed(meetingId : String) : boolean
+ getPlaceByMeeting(meetingId : String) : String
+ getTimeByMeeting(meetingId : String) : LocalDate
- changeTurns(meetingId : String, userNum : int) : void
MeetingGateway
+ meetInfo : HashMap<String, ArrayList<ArrayList<String>>
+ MeetingGateway()
+ getMeetInfoCsv() : ArrayList<Meeting>
+ convertToMeet(meetList : HashMap<String, ArrayList<ArrayList<String>>) : ArrayList<Meeting>
+ updateMeetInfoCsv(mm : MeetingManager) : void