UML for Regular User Controllers

RegularTradingMenuController

- -userId:int
- -username:String
- -am:ActionManager
- -im:ItemManager
- -tc:RegularUserThreshholdController
- -sm:SystemMessage
- -tm:TradeManager
- -mm:MeetingManager
- -um:UserManager
- +RegularUserTradingMenuController(tm:Trademanager,
- mm:MeetingManager,um:UserManager,im:Itemmanager,am:ActionManager, username:String, sm:SystemMessage)
- +hasTopThree():boolean
- +seeTopThreePartners():List<TradableUser>
- +confirmTradeComplete(tradeId int, respondStatus:String):boolean
- +respondToTradeRequests(tradeId int, respondStatus:String):void
- +tradeRequestsToRespond():List<Trade>
- -respondResult(tradeId:int, itemid22:int,
- userId11:int,userId22:int,itemId11:int,respondStatus:String)void
- -respondAgree(tradeId:int, itemid22:int, userId11:int,userId22:int,itemId11:int):void
- +lockThresholdOrNot():boolean
- +hasTradeSuggestion():boolean
- +requestTrade(numKindOfTrade:int, borrowerOrborrower1lender2: int,
- lenderOrlender1borrower2: int, itemId1: int, itemId2: int, numLentBeforeBorrow: int, tradeType:String):String
- +mostReasonableTradeSuggestions():Item
- -getValidationForItems(numKindOfTrade: int, itemId2: int, userId1: int, userId2: int, itemId: int):boolean
- -determineTradeID():int
- -getTrade(numKindOfTrade: int, itemId2: int, userId1: int, userId2: int, itemId: int, tradeID: int, tradetype: String): Trade
- -requestFail():String
- -requestResult(ok: boolean, trade:Trade, tradeId:int, userId1: int,
- numLendBeforeBorrow):String
- -validateItems(borrower1Lender2: int, borrower2lender1: int, itemId1:int, itemId2: int):boolean
- -requestSuccess(trade:Trade, tradeId: int):String
- -validateItems(borrower: int, lender: int, itemId1:int): boolean
- +viewOpenTrades(): List<Trade>
- +viewClosedTrades(): List<Trade>
- +viewCancelledTrades(): List<Trade>

RegularUserIDChecker

- tm: TradeManager
- um: UserManager
- im: ItemManager
- + RegularUserIDChecker(tm: TradeManager, um: UserManager, im: itemManager)
- + checkItemID(items: ArrayList<Item>, id: int): boolean
- + getItemsIDs(ArrayList<Item> allOtherItems): ArrayList<Integer>
- + checkInt(str: String): boolean
- + checkUserID(userId: int): boolean
- + checkTradeID(tradeId: int): boolean

RegularUserOtherInfoChecker

- + RegularUserOtherInfoChecker()
- + checkItemType(userInputCategory: String): boolean
- + checkEmail(email: String): boolean

RegularUserDateTimeChecker

- + RegularUserDateTimeChecker()
- + isValidTime(hour: int, min: int): boolean
- + isValidDay(year: int, month: int, day: int): boolean
- + isValidDayForMeetingTime(year: int, month: int, day: int): boolean

AccountCreator

- -UserManager um
- -FilesReaderWriter frw
- -String userInfoFilePath
- -String regularUserFilePath
 -String adminUserFilePath
- -String managerFilePath
- + AccountCreator(um: UserManager, frw: FilesReaderWriter,
- uerInfoFilePath: String, regularUserFilePath: String, adminUserFilePath: String, managerFilePath: String); void
- + createAccount(type: String, uername: String, password:
- String, password: String, email: String, home: String): boolean

LoginValidator

- userLoginInfo: Map<String, String>
- adminUserLoginInfo: Map<String, String>
- userType: String
- userName: String
- + LoginValidator(userLoginInfo: Map<String, String>, adminUserLoginInfo: Map<String, String>)
- + checkUsername(username: String): boolean
- + checkPassword(password: String): String

UML for Gateway **UML for Presenter**

FilesReaderWriter

- + FilesReaderWriter()
- + readFromMenu(filePath: String): String
- + readThresholdValuesFromCSVFile(filePath: String): List<Integer>
- + saveThresholdValuesToCSVFile(thresholdValues:
- List<Integer>, filePath: String): void
- + saveUserInfoToCSVFile(filePath: String, username: String, password: String, email: String): void
- + readUserInfoFromCSVFile(filePath: String): Map<String, String>
- + helper_check_file_empty_or_not(filePath: String): hoolean
- + readManagersFromFile(filePath: String): Object
- + saveManagersToFile(manager: Object, filePath: String):
- + MenuLength(filePath: String): int
- # check_file_empty_or_not(filePath: String): boolean

SystemMessage

- pom: PrintObjectMessage
- uam: UserAlertMessage
- asm: AccountSystemMessage
- tsm: TradeSystemMessage
- msm: MeetingSystemMessage thsm: ThresholdSystemMessage
- gsm: GeneralSystemMessage
- + SystemMessage()
- + msgForNothing(): String
- + invalidInput(): String
- + invalidNumber(): String
- + tryAgainMsgForWrongDateTime(): String
- + printResult(obj: ArrayList<Object>): String
- + printItemResult(obj: ArrayList<Item>): String
- + regUserAlerts(um: UserManager, tc: RegularUserThresholdController, username: String, menuPartOfAlert: String, thresholdValues: ArrayList<Integer>):
- activeAlerts(notification: StringBuilder, um: UserManager, tc:

RegularUserThresholdController, username: String, thresholdValues: ArrayList<Integer>): void

- + lockMessageForThreshold(maxNumTransactionAWeek: int): String
- + msgForThresholdValue(currentVal: int): String
- + msgForGuest(): String
- + lockMessageForFrozen(): String
- + lockMessageForVacation(): String
- + msgForResult(validator: boolean): String
- + msgForStatusChangeResult(validator: boolean): String
- + msgForRequestProcess(validator: boolean): String
- + msgForRequestResult(validator: boolean): String
- + failMessageForFrozen(): String
- + lockMessageForTPLimit(): String
- + msgForFriendRequest(validator: boolean, userTold: int): String
- + msgForNothing(string: String): String
- + printListUser(listOfUser: ArrayList<TradableUser>): String
- + printFriendRequest(requests: HashMap<TradableUser, String>): String
- + printListObject(objects: ArrayList<Object>): String
- + printListNumberedObject(objects: ArrayList<Object>): String
- + tryAgainMsgForWrongInput(): String + tryAgainMsgForWrongFormatInput(): String
- + msgForTPcannotConfirm(): String
- + msgForNotYourTurn(): String
- + msgForFollowResult(validator: boolean): String
- + msgForNoTradeSuggestion(): String
- + msgForMeetingTookPlaceResult(): String
- + msgForNo(string: String): String
- + printAllMessages(messages: ArrayList<Messages>): String
- + msgForTradeCompletedOrNot(result: boolean): String
- + msgForSetTradable(validator: boolean, status: boolean): String
- + printHistoricalAction(listOfAction: ArrayList<Action>): String
- + printObject(object: Object): String
- + msgTradeRequestFail(): String
- + msgForRatingReview(rating: double, reviews: ArrayList<Review>): String
- + msgForReview(reviews: ArrayList<Review>): String

AccountSystemMessage

- AccountSystemMessage()
- # msgForFriendRequest(validator: boolean, userToID: int): String
- # printFriendRequest(request:
- HashMap<TradadableUser, String>): String
- # msgForFollowResult(validator: boolean):String
- # printAllMessages(messages:
- ArrayList<Message>):String
- # msgForSetTradable(validator: boolean, status:
- boolean): String
- # msgForRatingReview(rating:double, reviews:
- ArrayList<Review>, pom: PrintObjectMessage): String
- # msgForReview(reviews: ArrayList<Review>, pom:
- PrintObjectMessage): String

ThresholdSystemMessage

- ~ ThresholdSystemMessage()
- lock Message For Threshold (maxNum Transactions A Week:int): String
- # msgForThresholdValue(currentVal: int): String
- # lockMessageForTPLimit(): String

MeetingSystemMessage

- MeetingSystemMessage()
- # tryAgainMsgForWrongDatetime(): String
- # msgForTPcannotConfirm(): String
- # msgForNotYourTurn(): String
- # msgForMeetingTookPlaceResult(validator: boolean): String

TradeSystemMessage

- ~ TradeSvstemMessage()
- # msgForNoTradeSuggestion(): String
- # msgForTradeCompletedOrNot(result: boolean): String
- # msgTradeRequestFail(): String

GeneralSystemMessage

- GeneralSystemMessage()
- # msqForNothing(): String
- # invalidInput(): String
- # invalidNumber(): String
- # msgForGuest(): String
- # lockMessageForFrozen(): String # lockMessageForVacation(): String
- # msgForResult(validator: boolean): String
- # msgForStatusChangeResult(validator: boolean): String
- # msgForRequestSuccess(validator: boolean): String
- # msgForReguestResult(validator: boolean): String
- # failMessageForFrozen(): String
- # msgForNothing(string: String): String
- # trvAgainMsgForWrongInput(): String # tryAgainMsgForWrongFormatInput(): String
- # msgForNo(string: String): String

PrintObjectMessage

- PrintObjectMessage()
- # printResult(obj: ArrayList<Object>): String
- # printItemResult(obj: ArrayList<Item>): String
- # printListUser(listOfUser: ArrayList<TradableUser>): String
- # printListObject(objects: ArrayList<Object>): String
- # printListNumberedObject(objects: ArrayList<Object>): String # printHistoricalAction(listOfAction: ArrayList<Action>): String

printObject(object: Object): String

UserAlertMessage

- UserAlertMessage()
- # regUserAlerts(um: UserManager, tc:
- RegularUserThresholdController, username: String, menuPartOfAlert: String, thresholdValues: ArrayList<Integer>):
- activeAlerts(notification: StringBuilder, um: UserManager, tc: RegularUserThresholdController, username: String, thresholdValues: ArrayList<Integer>): void

AdminUserActionMessage

- + AdminUserActionMessage()
- + adminUserAction(action: Action): String
- # adminUserManageUsersAction(action: Action,
- subMenuOption: int): String # adminUserEditThresholdsAction(action: Action,
- subMenuOption: int): String
- # adminUserActionAction(action: Action,
- subMenuOption: int): String
- # adminUserOtherAction(action: Action,
- subMenuOption: int): String # helper admin action prefix(action: Action): String

RegularUserActionMessage

- + RegularUserActionMessage()
- + regularUserAction(action: Action): String
- # regularUserAccountAction(action: Action, subMenuOption: int, subSubMenuOption: int): String
- # regularUserManageItemsAction(action: Action,
- subSubMenuOption: int): String
- # regularUserAccountSettingsAction(action: Action,
- subSubMenuOption: int): String # regularUserFollowAction(action: Action,
- subSubMenuOption: int): String
- # regularUserTradeAction(action: Action,
- subMenuOption: int): String # regularUserMeetingAction(action: Action,
- subMenuOption: int): String
- # regularUserSearchingAction(action: Action.
- subMenuOption: int, subSubMenuOption: int): String # regularUserSearchingItemsAction(action: Action,
- subSubMenuOption: int): String # regularUserSearchingUsersAction(action: Action,
- subSubMenuOption: int): String
- # regularUserSearchingMeetingsAction(action: Action, subSubMenuOption: int): String
- # regularUserSearchingTradesAction(action: Action, subSubMenuOption: int): String
- # regularUserCommunityAction(action: Action,

subMenuOption: int): String

helper regular action prefix(action: Action): String