## **UML for AdminUser Controllers**

# AdminUserManagerUsersController

- um: UserManager- im: ItemManager- am: ActionManager

- userID: int

+ AdminUserManagerUsersController(um:

UserManager, im: ItemManager, am: ActionManager, username: String)

+ getAllUnfreezeUser(): String

+ freezeUser(regularUsername: String): String

+ getWantUnfreezeUser(): String

+ unfreezeUser(regularUsername: String): String

+ seeListItemToAdd(): List<Item>

+ addItemOrNot(itemNum: int, add: boolean): void

### AdminUserOtherActionsController

- um: UserManager

- am: ActionManager

- mm: MeetingManager

- regularUserDateTimeChecker:

RegularUserDateTimeChecker

- userId: int

+ AdminUserOtherActionsController (um:

UserManager, am: ActionManager, username:String, mm: MeetingManager)

+ addNewAdmin(String username): void

+ checkSystemTime(year: int, month: int, day: int): boolean

+ setSystemTime(year: int, month: int, day: int): boolean

## AdminUserEditThresholdsController

sm: SystemMessageam: ActionMangerum: UserManager

- thresholdValues: List<Integer>

- username: String

- userId: int

+ AdminUserEditThresholdsController(am:

ActionManager, um UserManager, sm:

SystemMessage, username: String, thresholdValues: List<Integer> )

+ getMaxNumberTransactions(): String

+ getMaxNumberIncompleteTransactions(): String

+

editMaxNumberIncompleteTransactions(futureValue: int): String

+ getMustLendNumber(): String

+ editMustLendNumber(futureValue: int): String

+ getEditMaxEdits(): String

+ editMaxEdits(futureValue: int): String

#### AdminUserOtherInfoChecker

- am: ActionManager

+ AdminUserOtherInfoChecker(am:

ActionManager)

+ checkItemToAddNum(lenListItemToAdd: int,

userInput: int): boolean

+ checkActionId(actionID: int): boolean

## AdminUserHistoricalActionController

- um: UserManager
-tm: TradeManager
-im: ItemManager
-am: ActionManager
-fm: FeedbackManager
-mm: MeetingManager

-userId: Integer

+ AdminUserHistoricalActionController(um: UserManager,

im: ItemManager, tm: TradeManager, mm: MeetingManager, am: ActionManager, fm: FeedbackManager, username: String)

+ getAllAction(): ArrayList<Action>

+ getAllRevocableAction():ArrayList<Action>

+ getAllTradableUser(): ArrayList<TradableUser>

 $+\ search Revocable Action By User ID (regular User ID:\ int):$ 

ArrayList<Action>

+ findActionByID(actionID: int): Action

+ getUndoReguest(): Map<Integer, Integer>

+ checkUndoRequest(actionID: int): boolean

+ confirmRequestAndCancelAction(actionID: int): boolean

+ checkRevocable(targetAction: Action): boolean

+ checkUser(targetUserUsername: String): boolean

+ getUserID(targetUserUsername: String): boolean

+ cancelRevocableAction(targetAction: Action): boolean

 ${\it \# helper\_cancel Account Main Menu (target Action: Action,}$ 

subOption: int, subSubOption: int): boolean

# helper cancelManageItemMenu(targetAction: Action,

subSubOption: int): boolean

# helper cancelAccountSettingsMenu(targetAction: Action,

subSubOption: int): boolean

# helper\_cancelFollowMenu(targetAction: Action.

subSubOption: int): boolean

# helper\_cancelTradeMenu(targetAction: Action, subOption:

int): boolean

# helper\_cancelMeetingMenu(targetAction: Action,

subOption: int): boolean

# helper\_cancelCommunityMenu(targetAction: Action,

subOption: int): boolean

# **UML for Regular User Controllers**

### RegularUserSearchingMenuController

- tm: TradeManager
 -mm: MeetingManager
 -um: UserManager
 -im:ItemManager
 -fm: FeedBackManager
 -am:ActionManager

-userId:int

+RegularUserSearchingMenuController(tm:Trademanager, mm:MeetingManager, am:ActionManager, um:UserManager, im:ItemManager, username:String) sortRating():List<TradableUser> recentThreePartner():List<Integer> filterCompleteTrade():List<Trade> sortAllTradedPartner():List<Integer> filterIncompleteTrade():List<Trade> allMeetingSortByDate():List<Meeting> completeMeetingSortByDate():List<Meeting> unCompleteMeetingSortByDate():List<Meeting> filterByCategory(category: Category):ArrayList<Item> searchItemByName(name:String):ArrayList<Item> getItemById(itemId: int): String getItemObjectById(id: int): Item sortItemByFollows(): ArrayList<Item>

### RegularUserMeetingMenuController

tm: TradeManagermm: MeetingManageram: ActionManageruserId: int

+ RegularUserMeetingMenuController(tm: TradeManager, mm: MeetingManager, am: ActionManager, userId: int)

+ getUnconfirmedMeeting(): List<Meeting>

+ getCompleteMeetings(): List<Meeting>

+ isEmpty(meetings: List<Meeting>): boolean

+ checkValidMeeting(tradeId: int, numMeeting: int): boolean

+ confirmMeetingTookPlace(tradeId: int, numMeeting: int, maxMeetingTimePlaceEdit: int): boolean

+ getUnconfirmTimePlace(): List<Meeting>

+ confirmMettingTandP( tradeId: int, numMeeting: int, maxMeetingTimePlaceEdit: int): boolean

+ editMeetingTandP(tradeId :int, numMeeting: int, time: List<Integer>, place: String, maxMeetingTimePlaceEdit:int): boolean

+ checkOverEdit( tradeId: int, numMeeting: int, maxMeetingTimePlaceEdit: int) : String

### RegularUserCommunityMenuController

- um: UserManager

- am: ActionManager

- fm: FeedbackManager

- messageManager: MessageManager

- username:String

- userId: int

+ RegularUserCommunityMenuController(um: UserManager, am: ActionManger,

fm: FeedbackManager, messageManager: MessagerManager, username: String)

+ getUserId(): int

+ reviewUser(userToReview: int, rating: int, reason: String): boolean

+ reportUser(userToReport: int, readon: String): boolean

+ findRatingForUser(id: int): double

+ getAllReviews(userId: int): ArrayList<Review>

+ checkUserId(id: int): boolean

+ seeUsersInSameHC(): ArrayList<TradableUser>

+ getFriends(): ArrayList<TradableUser>

+ getNotFriends(): ArrayList<TradableUser>

+ sendFriendRequest(userToId: int, msg: String):boolean

+ getFriendRequest(): HashMap<TradableUser, String>

+ checkIdInRequest(userId1): boolean

+ addFriend(id1: int): boolean

+ unfriendUser(id: int): boolean

+ checkIsFriend(userId1: int): boolean

+ sendMessage(receiverId: int, message: String): boolean

+ getAllMessages(): ArravList<Message>

### RegularUserThresholdController

- mm: MeetingManager- um: UserManager- username: String

- userId: int

+ public RegularUserThresholdController(mm:

MeetingManager, um: UserManager, username: String)

+ reassessNumTransactionsLeftForTheWeek(maxNum

TransactionAllowedAWeek: int): void

# changeNumTradesLeftForTheWeek(): void

# isFirstDayOfTheWeek(): boolean

+ freezeUserOrNot(maxNumTransactionIncomplete: int): boolean

oolean

- numUncompletedTransactions(): int

### RegularUserAccountMenuController

- sm: SystemMessage

- tm: TradeManager

- um: UserManager

- im: ItemManager

- am: ActionManager

- username: String

- userId: int

+ RegularUserAccountMenuController(tm: TradeManager, um: UserManager, im: ItemManager, am: ActionManager, username: String, userId: int)

+ getTradables(): ArrayList<Item>

+ getUserId(): int

+ getAllTradableFromOtherNotInWishlist(): ArrayList<Item>

+ addToWishList(tempItemId: int): boolean

+ removeFromWishlist(tempItemId: int): boolean

+ removeFromInventory(tempItemId: int): boolean

+ requestAddItem(itemName: String, description: String, category: Category): void

+ seeMostRecentThreeItems(): ArrayList<Item>

+ getWlshList(): ArrayList<Item>

+ getInventory(): ArrayList<Item>

+ getTradableItems(): ArrayList<Item>

+ getNotTradableItems(): ArrayList<Item>

+ setTradableBasedOnResponses(itemId: int, option: boolean): boolean

+ getSuggestItemtoLend(lendToUserId: int): ArrayList<Item>

+ getRandomSuggestion(lendtoUserId: int): ArrayList<Item>

+ requestToUnfreeze(msg: String): boolean

+ setOnVacationStatus(newStatus: boolean): void

+ changeUserHC(newHC: String): String

+ followAnUser(userToFollowUserId: int): boolean

+ followAnItem(itemToFollowId: int): boolean

+ seeRecentStatusOfFollowedUser(): ArrayList<String>

+ seeRecentStatusOfFollowedItem(): ArrayList<String>

+ seeOwnRevocableAction(): ArrayList<Action>

+ requestUndoARevocableAction(actionId: int): void

+ selfFrozen(): boolean

+ seelfOnVacation(): boolean