#### **UML for Entities** <<interface>> Serializable User AdminUser Meeting - username: String - username: String - time: Date - password: String - password: String - place: String - email: String - email: String - wishList: Arraylist<Integer> - tradeId: int - id: int - userId1: int - Inventory: Arraylist<Item> - userId2: int - id: Int + AdminUser(username: String, - meetingNum: int - numLent: Int password: String, email: String, - timePlaceConfirm: Boolean - numBorrowed: Int adminID: Int) - meetingConfirm: Map<Integer, Boolean> - isFrozen: Boolean + getEmail(): String - timePlaceEdit: List<Integer> - numFrozen: Int + getUsername(): String - transactionsLeftForTheWeek:Int + aetId(): Int + getPassword():String + User(username:String, password: String, email: String, userID: Int) meetingNum: int) + getPassword(): String + getTime(): Date + getEmail(): String + getUsername(): String int, sec: int): void + setWishList(wishList: + getPlace(): String Item Arraylist<Integer>): void + setPlace(place: String): void + getWishList(): Arraylist<Integer> + getTradeId(): int - name: String + setTradeId(tradeId: int): void - description: String setInventory(inventory:Arraylist<Item>): + getUserId1(): int - itemId: int + setUserId1(userId1: int): void - ownerId: int + getInventory():Arraylist<Item> + getUserId2(): int - currHolderId: int + getId(): Int + setUserId2(userId2: int): void + getNumLend(): Int + getMeetingNum(): int + addOneToNumLent(): void + setMeetingNum(meetingNum: int): void

+ getNumBorrowed(): Int

+ addOneToNumBorrowed: void

+ getIfFrozen(): boolean

+ setIfFrozen: void

getNumTransactionLeftForTheWeek(): Int

setTransactionLeftForTheWeek(newVal: Int):void

+ getNumFrozen(): Int

+ addOneToNumFrozen(): void

+ Item(name: String, description: String, ownerld: int, itemID: int)

+ getName(): String

+ setName(name: String): void

+ getDescription(): String

+ setDescription(name: String): void

+ getItemId(): int

+ getOwnerId(): int

+ getCurrHolderId(): int

+ setCurrHolderId(): void

+ toString(): String

+ Meeting(tradeld: int, userId1: int, userId2: int,

+ setTime(year: int, month: int, date: int, hour: int, min:

+ getTimePlaceConfirm: Boolean

+ setTimePlaceConfirm(timePlaceConfirm: Boolean): Boolean

+ getMeetingComfirm: Map<Integer, Boolean>

+ getTimePlaceEdit(): List<Integer>

+ setTimePlaceEdit(listId: List<Interger>): void

+ setTimePlaceEdit(userId: int, year: int, month: int, day: int, hour: int, min: int, sec: int, place: String): Boolean

+ countEdit(userId: int): int

+ toString(): String

+ equals(meeting: Meeting): Boolean

Trade

- tradeId: int - userId1: int - userId2: int

- itemId: int - itemId1: int

- isOneWayTrade: boolean

# userStatus: Map<Integer, String>

# tradeType: String # tradeStatus: String

+ Trade(usrld1: int. userld2: int. itemId: int. tradeType: String, isOneWayTrade:boolean, tradeld: Int)

+ Trade(usrld1: int, userld2: int, itemId: int, itemId1: int tradeType: String, isOneWayTrade:boolean, tradeId:

+ getUserStatus(userId:int): String

+ setUserStatus(userId:int, status:String):String

+ getIsOneWayTrade(): boolean

+ getIds(): List<Integer>

+ openTrade(): void

+ closedTrade(): void

+ cancelTrade(): void

+ toString(): String

# **UML for Use Cases**

# <<interface>> Serializable

# UserManager

- listAdmin: ArrayList<AdminUser>
- listUser: ArravList<User>
- listUnfreezeRequest: ArrayList<String[]>
- -listItemToAdd: ArrayList<Item>
  -listAllItems: ArrayList<Item>
- + UserManager()
- + UserManager(users: ArrayList<User>, admins:

ArravList<AdminUser>)

- + getListAdmin(): ArrayList<AdminUser>
- + getListUser(): ArrayList<User>
- + getListUnfreezeRequest(): ArrayList<String[]>
- + getListItemTo Add(): ArrayList<Item>
- + searchItem(item: String): ArrayList<Item>
- + freezeUser(username: String): boolean
- + unfreezeUser(username: String): boolean
- + checkUser(username: String): boolean
- + addUser(username: String, password: String, email: String): void
- + addAdmin(username: String, password: String, email: String): void
- + underLending(): ArrayList<String>
- + removeItemWishlist(itemID: Integer, username: String): boolean
- + removeItemInventory(itemID: int, username: String): boolean
- + addItemWishlist(itemID: int, username: String): boolean
- + addItemInventory(item: Item, username: String): boolean
- + userPasswords(): HashMap<String, String>
- + adminPasswords(): HashMap<String, String>
- + findUser(username: String): User
- + findUser(ID: int): User
- + allItems(username: String): ArrayList<Item>
- + allItems(ID: int): ArrayList<Item>
- + requestUnfreeze(username: String, message: String): boolean
- + requestAddItem(name: String, description: String, ownerID: int): void
- + addItemToAllItemsList(newItem: Item): void
- + removeItemFromUsers(userId1: int, userId2: int, itemId: int): void

# MeetingManager

- -MeetingList: List<Meeting>
- + MeetingManager()
- + getListMeeting(): List<Meeting>
- + setListMeeting(listMeeting: List<Meeting>): void
- + getMeetingsByUserId(userId: int): List<Meeting>
- +getCompleteMeetings(userId: int): List<Meeting>
- +getUnCompleteMeetings(userId: int, tradeManager:

TradeManager) throws InvalidIdException: List<Meeting>

+getUnConfirmTimePlace(userId: int, tradeManager:

TradeManager) throws InvalidIdException: List<Meeting>

+ getUnConfirmMeeting(userId: int):

List<Meeting> (time and place is confirmed but the meeting have not comfirm)

- +getMeetingsByTradeId(tradeId: int): List<Meeting>
- +checkInManager(tradeId: int, numMeeting: int): Boolean
- + getMeetingByIdNum(tradeId: int, numMeeting: int) : Meeting
- + setMeetingConfirm(tradeManager:

TradeManager, Meeting: meeting, userId: int) throws InvalidIdException: Boolean

- + getOverTime(meeting: Meeting): Boolean
- + getListOverTime(userId int): List<Meeting>
- +addMeeting(tradeId: int, userId1: int, userId2: int,

meetingNum: int, tradeManager: TradeManager) throws InvalidIdException: Meeting

- + getEditOverThreshold(tradeManager: TradeManager, meeting: Meeting) throws InvalidIdException: String
- +removeMeeting(meeting: Meeting): void
- + toString(): String

# TradeManager

- listTrade: List<Trade>
- + TradeManager()
- + getListTrade(): List<Trade>
- + addTrade(t: Trade): void
- + getTradeHistory(userId: int): List<Trade>
- + filterHistory(userId:int): List<Trade>
- +topThreePartners(userId:int): Lst<Integer>
- + recentThreeItem(userId): List<Integer>
- + getOpenTrade(userId: int):List<Trade>
- + getClosedTrade(userId:int): List<Trade>
- + getCancelledTrade(uerId: int): List<Trade>
- + getWaitTrade(userId: int): List<Trade>
- + checkInManager(tradeId: int): Boolean
- + getTradeById(tradeId: int): Trade
- + removeTrade(tradeId: int): Trade
- + comfirmComplete(tradeId: int): boolean
- + validateTrade(trade: Trade, borrower:User numLendBeforeBorrow: int):boolean

# **UML for Main Controller Classes**

### RegularUserController

- amc: RegularUserAccountMenuController
- atc: RegularUserAccountTradingMenuController
- mmc: RegularUserMeetingMenuController
- tc: RegularUserThresholdController
- sm: SystemMessage
- ds: DisplaySystem
- rw: FilesReaderWriter
- tm: TradeManager
- mm: MeetingManager
- um: UserManager
- username: String
- userId: int
- + RegularUserController(ds: DisplaySystem, rw: FilesReaderWriter, tm: TradeManager, mm:

MeetingManager, um: UserManager, username: String)

- + actionResponse(mainMenuOption: int,
- subMenuOption: int): void
- userAccountMenuResponse(subMenuOption: int) void
- userTradingMenuResponse(subMenuOption: int): void
- userMeetingMenuResponse(subMenuOption: int): void

# <<interface>> Controllable

+ actionResponse(subMenuName: String, menuOptionNum: int): void

### AdminUserController

- otherInfoGetter: AdminUserOtherInfoGetter
- sm: SystemMessage
- ac: AccountCreator
- ds: DisplaySystem
- rw: FilesReaderWriter
- um: UserManager
- + AdminUserController(ac: AccountCreater, ds: DisplaySystem, rw: FilesReaderWriter, um: UserManager,)
- + actionResponse(mainMenuOption: int, subMenuOption: int): void
- adminManageUsersMenuResponse(subMenuOption: int): void
- responseToToAddListSize(listItemToAdd: ArrayList<Item> . len: int); void
- addOrNotAdd(itemSelected: Item): void
- $adminEditThresholdMenuResponse (subMenuOption: int): \\void$
- adminOthersMenuResponse(isubMenuOption: int): void

# **UML for Controllers' Getter Classes**

### RegularUserIDGetter

ds: DisplaySystemtm: TradeManagermm: MeetingManagerum: UserManagerusername: String

- userId: int

+ RegularUserIDGetter(ds: DisplaySystem, tm: TradeManager, mm: MeetingManager, um: UserManager, username: String, userId: int)

# getItemID(ArrayList<Item> potentialItems, int type): int

- getItemsIDs(ArrayList<Item> allOtherItems):

ArrayList<Integer>

# idToItem(int id): Item

+ getAllItems(): ArrayList<Item>

# getUserID(String type): int

# getTradeID(): int

### RegularUserOtherInfoGetter

- ds: DisplaySystem

- tm: TradeManager

- mm: MeetingManager

- um: UserManager

- username: String

- userId: int

+ RegularUserOtherInfoGetter(ds: DisplaySystem, tm: TradeManager, mm: MeetingManager, um: UserManager,

username: String, userId: int)
# getItemName(): String

# getMessage(String TypeOfMessage): String

# getTradeType(): String # getAgreeOrNot(): String # getPlace(): String # getNumMeeting(): int

# getNumKindOfTrade(): int

### RegularUserDateTimeGetter

- + RegularUserDateTimeGetter()
- +getValidDate(ds: DisplaySystem): List<Integer>
- isValidTime(hour: int, min: int): boolean
- isValidDay(year: int, month: int, day: int): boolean

# AdminUserOtherInfoGetter

- ds : DisplaySystem

+ AdminUserOtherInfoGetter(ds: DisplaySystem)

# getItem(numItemsToAdd: int): int

# getThresholdAns(): int

# getAddOrNot(): boolean

# UML for RegularUserController's Sub-controller Classes

## RegularUserTradingMenuController

- tc: RegularUserThresholdController
- otherInfoGetter: RegularUserOtherInfoGetter
- idGetter: RegularUserIDGetter
- sm: SystemMessage
- ds: DisplaySystem
- tm: TradeManager
- mm: MeetingManager
- um: UserManager
- username: String
- userId: int
- + RegularUserTradingMenuController( ds: DisplaySystem,tm:

TradeManager, mm: MeetingManager, um: UserManager, username:

String, userId: int)

- # seeTopThreePartners(): void
- # confirmTradeComplete(): void
- # viewOpenTrades(openTrade: List<Trade>): void
- # respondToTradeRequests(thisUser: User): void
- isAgreedBefore(agreedBefore: boolean, agree: boolean, b: boolean):
- respondResult(tradeID: int, trade: Trade, itemid22: int, agreedBefore: boolean, userId11: int, userId22: int, itemId11: int); void
- respondAgree(tradelD: int, trade: Trade, itemid22: int, userId11: int, userId22: int, itemId11: int): void
- # requestTrade(thisUser: User): void
- determineTradeID(): int
- getValidationForItems(numKindOfTrade: int, itemId2: int, userId1: int, userId1: int, userId2: int, itemId: int): boolean
- getTrade(numKindOfTrade: int, itemId2: int, userId1: int, userId2: int, itemId: int, tradeID: int, tradeType: String): Trade
- requestResult(thisUser: User, ok: boolean, trade: Trade, userId1: int):
- requestFail(thisUser: User): void
- requestSuccess(thisUser: User, trade: Trade): void
- validateItems(borrower: int, lender: int, itemId: int): boolean
- validateItems(borrower1Lender2: int, borrower2lender1: int, itemId1: int, itemId2: int): boolean

## RegularUserMeetingMenuController

- sm: SystemMessage
- idGetter: RegularUserIDGetter
- dateTimeGetter: RegularUserDateTimeGetter
- otherInfoGetter: RegularUserOtherInfoGetter
- ds: DisplaySystem
- tm: TradeManager
- mm: MeetingManager
- um: UserManager
- username: String
- userId: int
- + RegularUserMeetingMenuController(ds: DisplaySystem, tm: TradeManager, mm: MeetingManager, um; UserManager,

username: String, userId: int)

- # seeMeetingsToBeConfirmed(unConfirmMeeting: List<Meeting>,
- s: String): void
- # confirmMeetingTookPlace(): void
- # confirmMeetingTandP(): void
- # EditMeetingTandP(): void
- # unconfirmedTandPMeetings(): void
- getMeeting(): Meeting

### RegularUserThresholdController

- ds: DisplaySystem
- tm: TradeManager
- mm: MeetingManager
- um: UserManager
- username: String
- userId: int
- thresholdReassessed: boolean
- + public RegularUserThresholdController(ds: DisplaySystem,tm: TradeManager, mm: MeetingManager, um: UserManager, username: String, userId: int)
- # reassessNumTransactionsLeftForTheWeek(thisUser: User): void
- # changeNumTradesLeftForTheWeek(thisUser: User): void
- # isFirstDayOfTheWeek(): boolean
- # freezeUserOrNot(User thisUser): boolean
- numUncompletedTransactions(): int

### RegularUserAccountMenuController

- sm: SystemMessage
- idGetter: RegularUserIDGetter
- otherInfoGetter: RegularUserOtherInfoGetter
- ds: DisplaySystem
- tm: TradeManager
- mm: MeetingManager
- um: UserManager username: String
- userId: int
- $+ \ \ Regular User Account Menu Controller (ds: Display System, tm:$

TradeManager, mm: MeetingManager, um: UserManager,

- username: String, userId: int)
  # viewWishListInventory(): void
- # addToWishList(allOtherItems: ArrayList<Item>): void
- # requestAddItem(): void
- # RequestToUnfreeze(): void
- # seeMostRecentThreeItems(): void
- # removeFromInventory(): void
- # removeFromWishList(allOtherItems: ArrayList<Item>): void
- # searchItem(): void
- # browseBooks(allOtherItems: ArrayList<Item>): void

### AccountCreator

- um: UserManager
- ds: DisplaySystem
- fr: FilesReaderWriter
- $+\ Account Creator (um: User Manager, \ ds: \ Display System, \ fr:$

FilesReaderWriter)

+ CreateAccount(type: String): boolean

### LoginValidator

- userLoginInfo: Map<String, String>
- adminUserLoginInfo: Map<String, String>
- + LoginValidator(userLoginInfo: Map<String, String>, adminUserLoginInfo: Map<String, String>)
- + VerifyLogin(username: String, password: String): String

# **UML for Presenter**

### **TradingSystem**

- userManager: UserManager - displaySystem: DisplaySystem - tradeSystem: TradeSystem
- meetingManager: MeetingManager - loginValidator: LoginValidator - accountCreator: AccountCreator filesReaderWriter: FilesReaderWriter
- + TradingSystem(userManager: UserManager, displaySystem: DisplaySystem, meetingManager: MeetingManager, loginValidator; LoginValidator, filesReaderWriter: FllesReaderWriter, accountCreator: AccountCreator)\
- + tradingSystemInitial(): blolean
- login(): void - logOut(): void
- regularUserMain(username: String): void
- adminUserMain(): void

## SystemMessage

- + SystemMessage()
- + RegUserAlerts(um: UserManager, tc: RegularUserThresholdController, rw:
- FilesReaderWriter, ds: DisplaySystem, username: String): String
- activeAlerts(notification: StringBuilder, um: UserManager, tc:

RegularUserThresholdController, ds: DisplaySystem, username: String): void

- + AdminUserAlerts(rw: FilesReaderWriter): String # msgForMeetingDNE(ds: DisplaySystem): void
- # msgForNothing(ds: DisplaySystem): void
- # msgForNothing(nextPart: String, ds: DisplaySystem): void
- # lockMessageForThreshold(ds: DisplaySystem): void
- # msgForThresholdValue(currentVal: int, ds: DisplaySystem): void

### DisplaySyetm

- fileReader: FilesReaderWriter
- option: int
- + DisplaySystem()
- + printOut(str: String): void
- + getMenuAnswer(filePath: String): int
- getMenuLength(filePath: String): int
- printMenu(filePath: String): void
- + getUsername(): String
- + getPassword(): String
- + getEmail(): String
- + failLogin(): void
- + printResult(result: boolean): void
- + printResult(obj: ArrayList<Object>): void
- + printResult(str: String, result: boolean): void

# **UML for Gateway**

# FilesReaderWriter +FilesReaderWriter() +readFromMenu(filePath: String): String +readThresholdValuesFromCSVFile(filePath: String): List<Integer> +saveThresholdValuesToCSVFile(thresholdValues: List<Integer>, filePath: String): +saveUserInfoToCSVFile(filePath: String, username: String, password: String, email: String): void +readUserInfoFromCSVFile(filePath: String): Map<String, String> +helper check file empty or not(filePath: String): boolean +readUserManagerFromFile(filePath: String): UserManager +readTradeManagerFromFile(filePath: String): TradeManager +readMeetingManagerFromFile(filePath: String): MeetingManager +saveUserManagerToFile(userManager: UserManager, filePath: String): void +saveTradeManagerToFile(tradeManager: TradeManager, filePath: String): void +saveMeetingManagerToFile(meetingManager: MeetingManager, filePath: String): +MenuLength(filePath: String): int

# **UML for DemoMain**

DemoMain
+DemoMainManager(): void