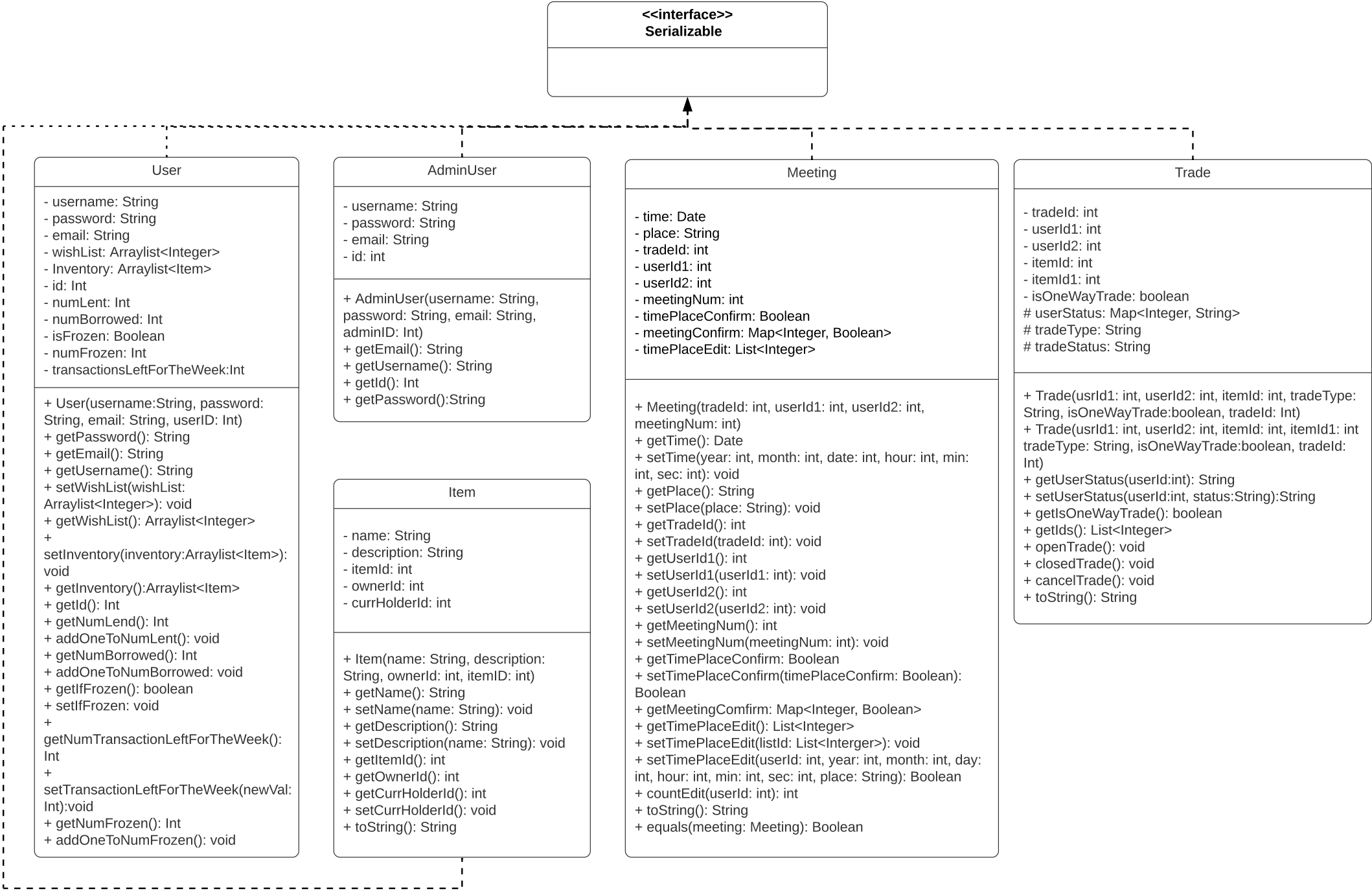
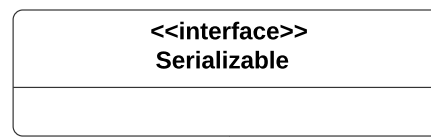


UML for Entities



UML for Use Cases



UserManager

- listAdmin: ArrayList<AdminUser>
- listUser: ArrayList<User>
- listUnfreezeRequest: ArrayList<String[]>
- listItemToAdd: ArrayList<Item>
- listAllItems: ArrayList<Item>

+ UserManager()
+ UserManager(users: ArrayList<User>, admins: ArrayList<AdminUser>)
+ getListAdmin(): ArrayList<AdminUser>
+ getListUser(): ArrayList<User>
+ getListUnfreezeRequest(): ArrayList<String[]>
+ getListItemToAdd(): 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 confirm)
+ 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>

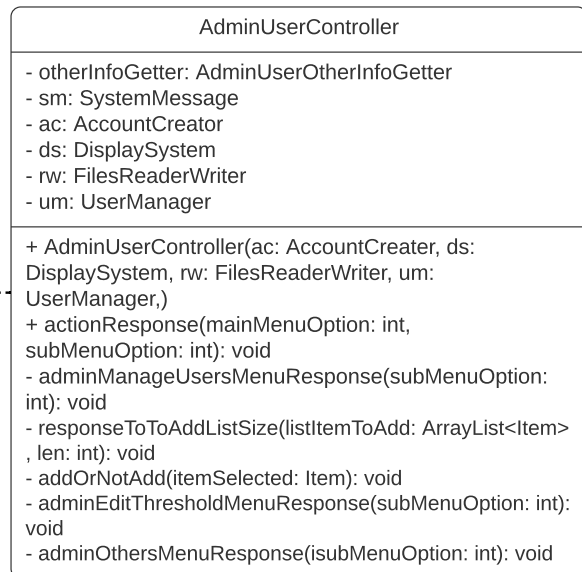
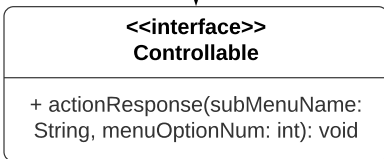
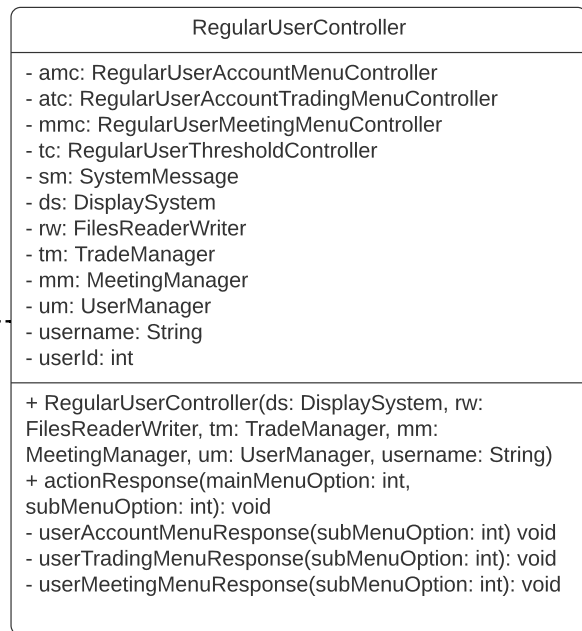
+ TradeManager()
+ getListTrade(): List<Trade>
+ addTrade(t: Trade): void
+ getTradeHistory(userId: int): List<Trade>
+ filterHistory(userId: int): List<Trade>
+ topThreePartners(userId: int): List<Integer>
+ recentThreeItem(userId: int): List<Integer>
+ getOpenTrade(userId: int): List<Trade>
+ getClosedTrade(userId: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

LoginValidator

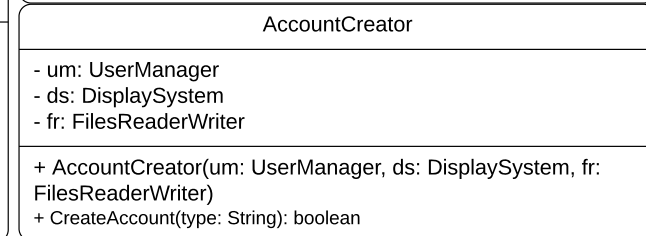
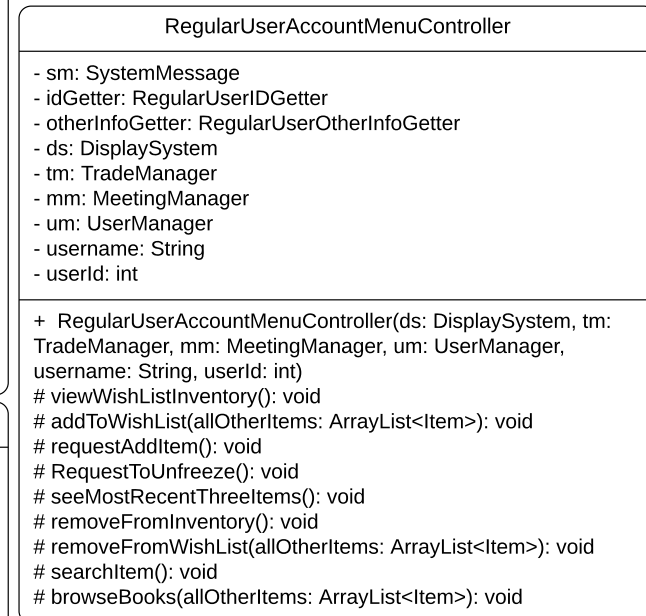
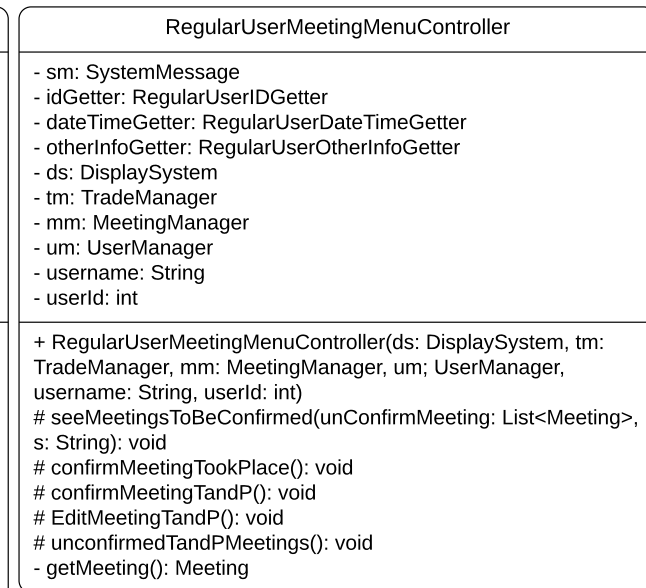
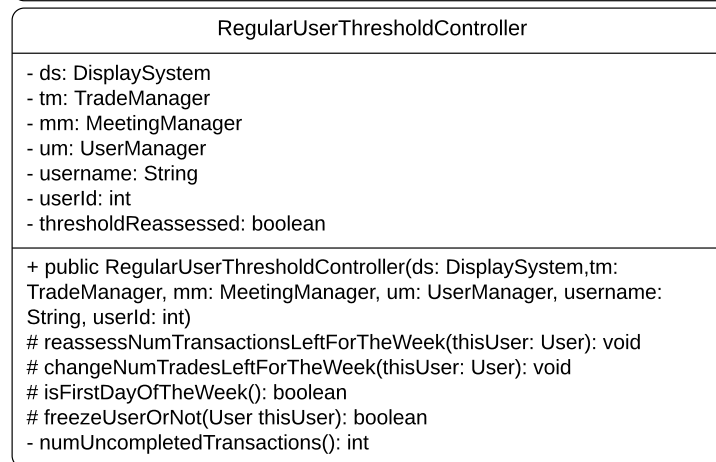
- userLoginInfo: Map<String, String>
- adminUserLoginInfo: Map<String, String>

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

UML for Main Controller Classes



UML for RegularUserController's Sub-controller Classes



UML for Controllers' Getter Classes

AdminUserOtherInfoGetter
- ds : DisplaySystem
+ AdminUserOtherInfoGetter(ds: DisplaySystem) # getItem(numItemsToAdd: int): int # getThresholdAns(): int # getAddOrNot(): boolean
RegularUserDateTimeGetter
+ RegularUserDateTimeGetter() + getValidDate(ds: DisplaySystem): List<Integer> - isValidTime(hour: int, min: int): boolean - isValidDay(year: int, month: int, day: int): boolean
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
RegularUserIDGetter
- ds: DisplaySystem - tm: TradeManager - mm: MeetingManager - um: UserManager - username: String - userId: int
+ RegularUserIDGetter(ds: DisplaySystem, tm: TradeManager, mm: MeetingManager, um: UserManager, username: String, userId: int) # getItemID(ArrayList<Item> potentialItems, int type): int - getItemIDs(ArrayList<Item> allOtherItems): ArrayList<Integer> # idToItem(int id): Item + getAllItems(): ArrayList<Item> # getUserID(String type): int # getTradeID(): int

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: FilesReaderWriter, accountCreator: AccountCreator)\n+ tradingSystemInitial(): boolean - 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): 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): 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): void +MenuLength(filePath: String): int

UML for DemoMain

DemoMain
+DemoMainManager(): void