**Model Classes:**

Class: User

Intent: Contains the attributes (as well as getters/setters) inherent to all users. Is extended by SuperUser and OrdinaryUser. Contains the methods “LogIn” and “create”

LogIn(username, password)

//Username and password are passed to the database. If username and password combo is found in OrdinaryUser or SuperUser table a new OrdinaryUser, or SuperUser object is returned. If no match is found, the function returns the current User object

Create(OrdinaryUserObject)

//pushes the information contained in the ordinary user object (which is constructed by the CreateNewUserManager) into the OrdinaryUser database

Class: OrdinaryUser

Intent: This is the class representing an ordinary user who is using the system to buy or sell items. They have attributes such as username, password, displayname, creditcard, phonenumber, rating, VIP status, etc.

Many functions of other objects which require the user to be logged in will check this prerequisite by checking that the current User of the system is instanceOf(OrdinaryUser).

Methods: placeBid(price, auctionObject), submitAuction(auctionObject), sendMessage(messageObject), submitComplaint(complaintObject), requestFriend(userObject), acceptFriend(userObject), updateKeyWord(String keyword), updateUserInfo(userObject), gradeUser(rating, userObject),

Pseudocode:

PlaceBid(price, auctionObject)

//Will submit to the bid table the price specified by the user for the given auction

SubmitAuction(auctionObject)

//will push to the database Auction table the information contained in the auctionObject

SendMessage(messageObject)

//will push to the database messages table the information contained in the messageObject

submitComplaint(complaintObject)

//pushes info in complaintObject to the database complaints table

requestFriend(userObject)

//pushes a new entry into the friends table, where the “accepted bit” is 0

acceptFriend(userObject)

//searches the friends table for the entry where the requesting friend is the userObject, and changes accepted bit to 1

updateKeyWord(string keyword)

//adds the specified keyword to the database UserKeywords table for the associated user

updateUserInfo(userObject)

//creates a new user object with the user specified values, and update the information for the current user in the OrdinaryUser database table with the new information

gradeUser(rating, userObject)

//adds the rating for the specified userObject to the ratings table

Each Model Object not explicitly defined in this document is assumed to have the same attributes described in the attached ER diagram.

**View Managers:**

Class: DisplayManager

Intent: Constructs and returns the standard view for ordinary and guest users. View is comprised of a JavaFX borderPane with the top component being a HBox with a company Logo, a search bar and submit button for auction searches, and the user’s display name. The left component is the navbar, which is comprised of series of buttons which call the DisplayManager’s “changeScene()” function, which in turn control which view is displayed in the boarderPane’s center component.

The navbar is a view constructed and returned by Class NavBarManager, and the banner is constructed and returned by class BannerManager.

Each view for the center pane is constructed by an associated View Manager.

PsuedoCode:

DisplayManager(User user){

//instantiate the view based on the type of User, instantiating and laying out subplanes

}

changeScene(SubScene){

//Swchanges the center component of the borderpane to the subscene passed as an argument

}

Class: NavBarManager

Intent: Constructs and returns the NavBar view for the DisplayManager. Each button in the UI will call the function DisplayManager.changeScene() passing the argument associated with the given button.

getSubScene()

//returns the subscene generated by the class to be displayed by the DisplayManager

Class: BannerManager

Intent: Constructs and returns the Banner view for the DisplayManager. This banner is based on the User info which is passed as a parameter to the constructor for BannerManager

getSubScene()

//returns the subscene generated by the class to be displayed by the DisplayManager

Class: AuctionSearchManager

Intent: Constructs and returns the view for the auction search results based on search parameters passed to constructor. Each auction has an associated button and text area where the user may submit a bid to the displayed auction. At any time after a bid has been placed, a user may rate the user. It is recommended the user wait until they receive the item before rating the user.

Methods:

retrieveAuctions(searchParameters)

//returns a list of auctions matching the parameters specified,

parseAuctions()

//parses the list of returned auctions into individual auctionObjects

getSubScene()

//returns the subscene generated by the class to be displayed by the DisplayManager

Class: AuctionCreationManager

Intent: Constructs and returns the view for the user to input the information to be submitted for a auction pending approval. The button to submit the auction will call User.submitAuction() passing the information generated by the “generateAuctionInfo()” function as an argument.

Methods:

AuctionCreationManager()

//instantiate UI

GenerateAuctionInfo()

//Creates a new auction object based on the information supplied to the fields by the user

getSubScene()

//returns the subscene generated by the class to be displayed by the DisplayManager

Class: MessageManager

Intent: Constructs and returns a view created from the User’s inbox. Retrieves all messages from the message table where recipient = username, and populates them in a list. Each message has an associated “reply” button which creates a new messageObject with the user’s specified response. Then the button calls User.sendMessage() passing the newly created messageObject as an argument.

Methods:

createNewMessage(string Username)

//in addition to the option specified above vis a vis replying to a message, there is also an option to send a new message by specifying a username to be delivered to before sending

getSubScene()

//returns the subscene generated by the class to be displayed by the DisplayManager

Class: FriendManager

Intent: Constructs and returns a view created using the user’s friends. It lists current friends (as queried from the database) as well as provides the option to request a friend, provided the user knows the username of the friend they wish to request. Usernames can be obtained through messaging, auctions, or off-app communication.

generateUI()

//Queries the database for all confirmed friendships where confirming friend OR requesting friend = username, and confirmed = true to populate friends list

//also queries database for all cases where confirmingfriend = username, and confirmed = false in order to populate the list of current pending requests.

//generates a text box and a button to push a new friend request to the table, after the intended friend has been validated as an existing user

getSubScene()

//returns the subscene generated by the class to be displayed by the DisplayManager

Class: UserComplaintManager

Intent: Constructs and returns a UI for users to submit complaints. They enter a reason into a javaFX “textarea,” and the username of the offending user.

generateComplaintInfo()

//generates a new complaint object when a button is pushed, and calls User.SubmitComplaint() passing the complaint object as an argument.

getSubScene()

//returns the subscene generated by the class to be displayed by the DisplayManager

Class: TransactionHistoryManager

Intent: Constructs and returns a view based on all of the user’s bids and auctions, listed by date, and generated by querying the database.

getSubScene()

//returns the subscene generated by the class to be displayed by the DisplayManager