

## Welcome to eHills!

Hi! My name is **Shanil Jasani** and I am the programmer behind **eHills**, my EE 422C Final Project. If you want to buy things from an online auction, eHills is the software for you! Here's how it works:

### Top Features

- All your active auctions in one place
- Ready to bid, out of the download package
- Integrated history and account information tool
- Free to use

## About eHills

### User's Point of View

#### User Profile

Users of all ages and backgrounds come to the Auction market via eHills to place their bids on the latest and best items available.

### Modern Problems Require Modern Solutions

Let's get to the point. Modern electronic auctioning sites just don't cut it anymore. That's why I created eHills, an easy-to-use, straightforward bidding software.

### The Solution

eHills!

## Quick Start Guide

eHills has an **easy to use bidding interface**. It's as easy as

1. Selecting an Item
2. Entering your bid
3. Placing your bid

It's just that simple!

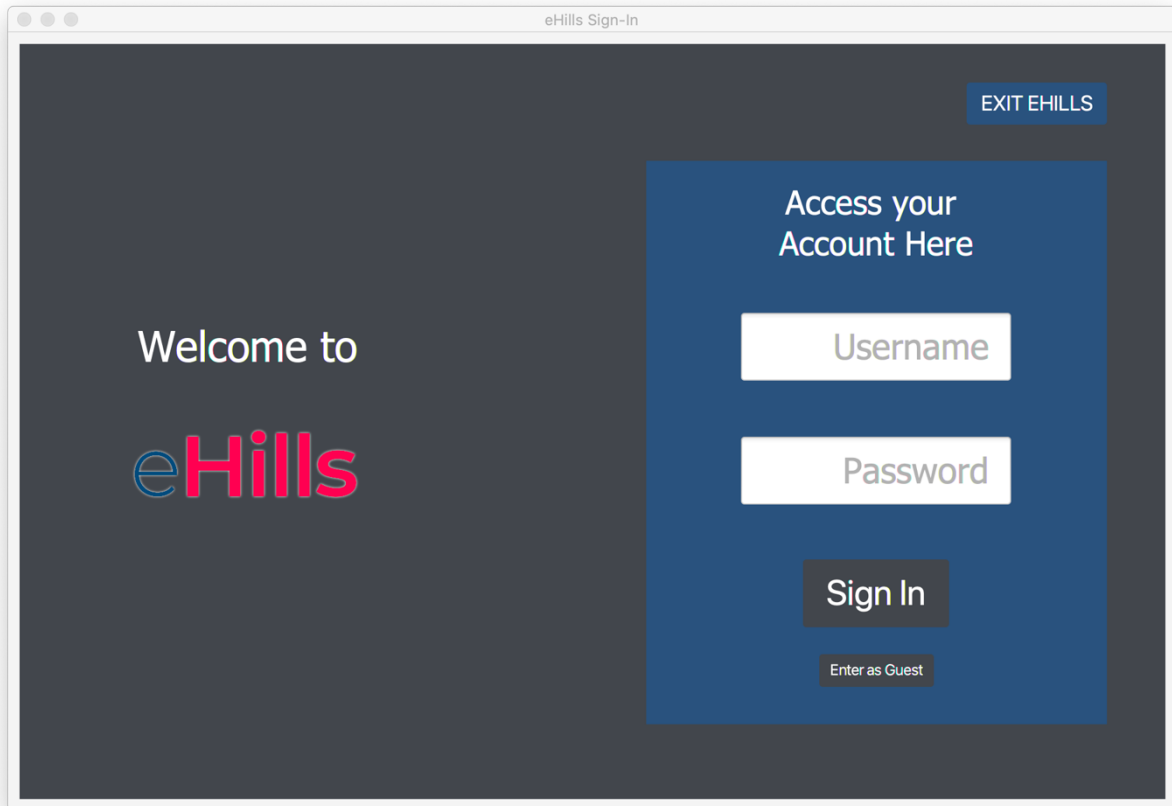
You also have the opportunity to

- Read more about the item in the description
- View auction details such as starting price, ending price, and current highest bidder and bid value
- See your bidding history and account information

## User Guide

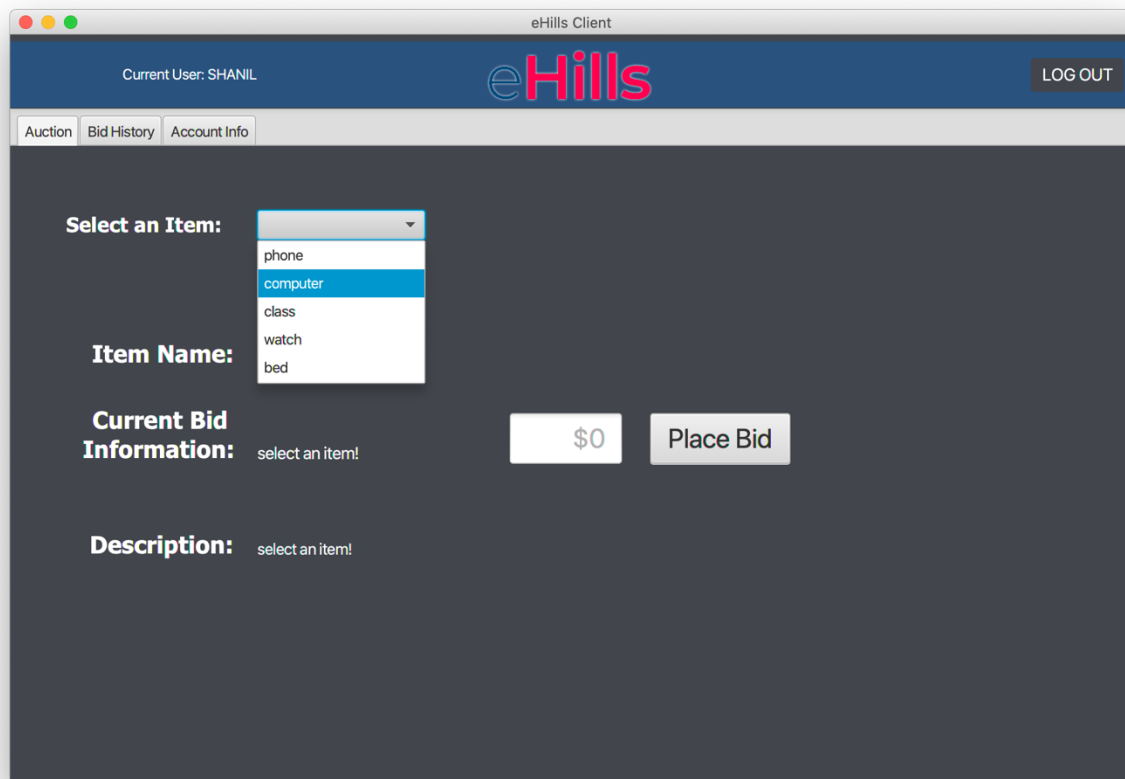
### Sign-in to eHills using your given username and password or as a guest user

*(to see all usernames and passwords used in the current implementation, visit the Programmer's Point of View Section)*

A screenshot of a web browser window titled "eHills Sign-In". The window has a dark gray background. On the left side, it says "Welcome to" in white text, followed by the "eHills" logo where the 'e' is blue and "Hills" is red. On the right side, there is a blue rectangular panel. At the top of this panel, it says "Access your Account Here" in white. Below this, there are two white input fields: the first is labeled "Username" and the second is labeled "Password". Below the password field is a dark gray button labeled "Sign In". Below the "Sign In" button is a smaller, lighter gray button labeled "Enter as Guest". In the top right corner of the dark gray background, there is a small blue button labeled "EXIT EHILLS".

Signing in may take a moment, everything is syncing up to give you the best experience!

Once you've signed in, select an item to bid on from the dropdown menu.



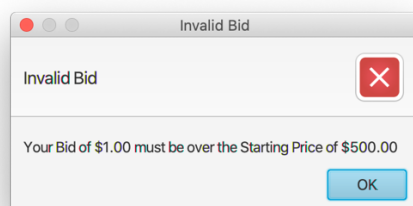
The screenshot shows the eHills Client web application. At the top, the current user is SHANIL, and there is a LOG OUT button. Below the header, there are tabs for Auction, Bid History, and Account Info. The main content area has a dark background. On the left, there is a 'Select an Item:' dropdown menu with a list of items: phone, computer (highlighted), class, watch, and bed. Below this is an 'Item Name:' label. To the right of the dropdown is a 'Current Bid Information:' section with a 'select an item!' placeholder, a '\$0' bid amount, and a 'Place Bid' button. At the bottom, there is a 'Description:' label with a 'select an item!' placeholder.

### Then enter a valid bid amount

A valid bid is above the starting bid and greater than the maximum current bid by a minimum of \$0.01.

The server processed bids on a first-come, first-serve basis, so enter your bids quickly!

There are a number of errors you may encounter with placing a bid. Most are related to the validity of your bid.



The screenshot shows an 'Invalid Bid' error dialog box. It has a title bar with the text 'Invalid Bid'. Inside the dialog, there is a red 'X' icon in the top right corner. Below the icon, the text reads: 'Your Bid of \$1.00 must be over the Starting Price of \$500.00'. At the bottom right of the dialog is an 'OK' button.

If all goes well, you'll see the window refresh showing your bid as the highest bid for the item!

The screenshot shows a web browser window titled "eHills Client". The interface has a dark blue header with the "eHills" logo in white and red. To the left of the logo, it says "Current User: GUEST". To the right is a "LOG OUT" button. Below the header is a navigation bar with three tabs: "Auction", "Bid History", and "Account Info". The main content area has a dark grey background. At the top, there's a "Select an Item:" label followed by a dropdown menu showing "watch". Below this, the "Item Name:" is "watch". The "Current Bid Information:" section shows "Bidding starts at: \$200.00", "Bidding ends at: \$500.00", and "Current Bid: shanil / \$500.00". To the right of this information is a white input field containing "\$0" and a grey "Place Bid" button. The "Description:" section shows the text "apple watch series 100 is so out of this future it'll tell you what next week's lunch is".

eHills Client

Current User: GUEST

eHills

LOG OUT

Auction Bid History Account Info

Select an Item: watch

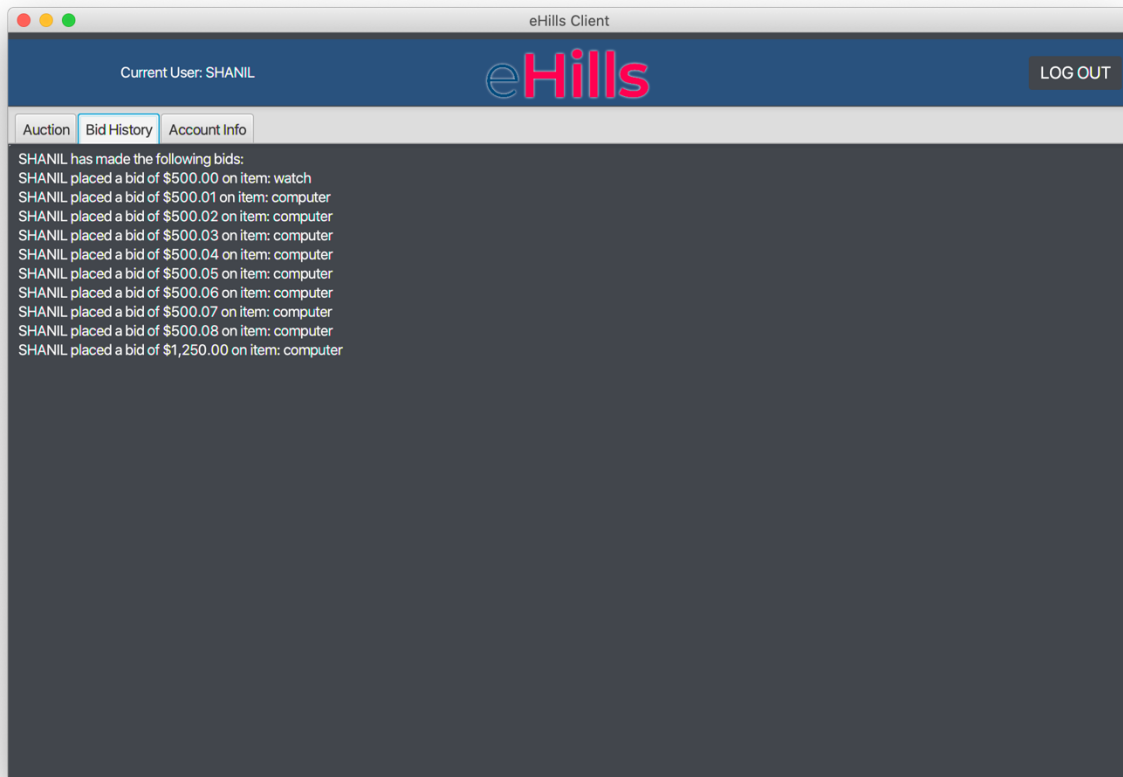
Item Name: watch

Current Bid Information: Bidding starts at: \$200.00  
Bidding ends at: \$500.00  
Current Bid: shanil / \$500.00

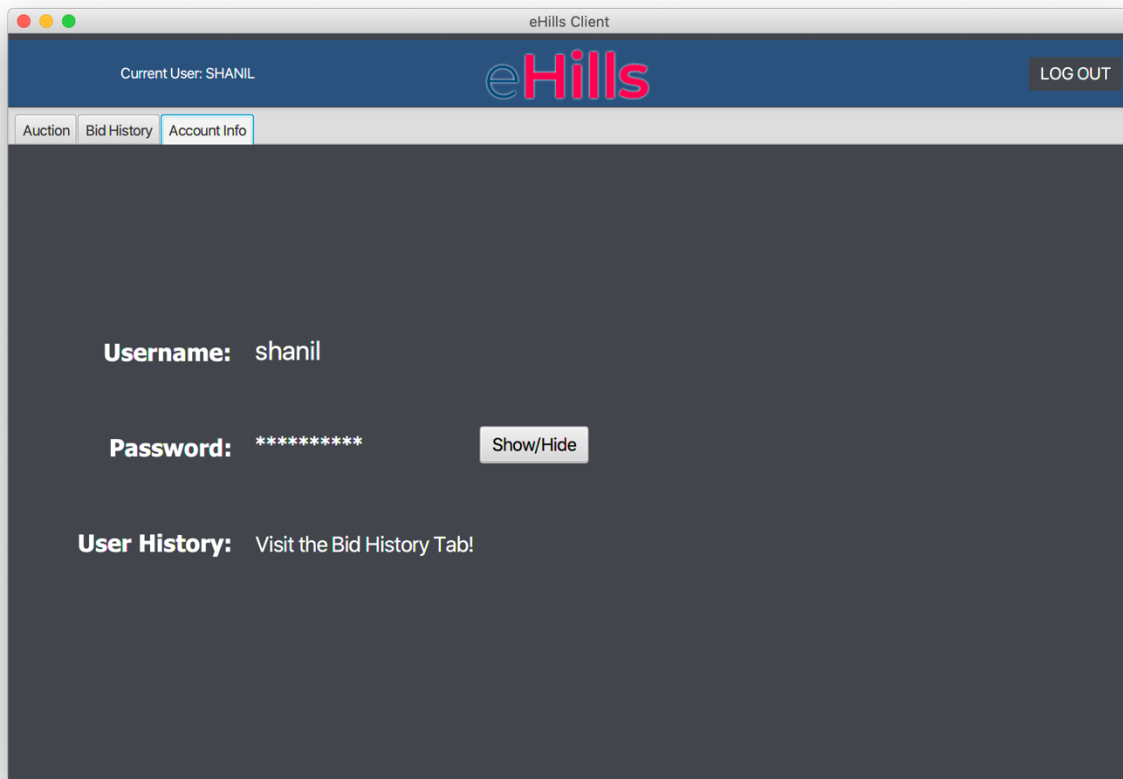
\$0 Place Bid

Description: apple watch series 100 is so out of this future it'll tell you what next week's lunch is

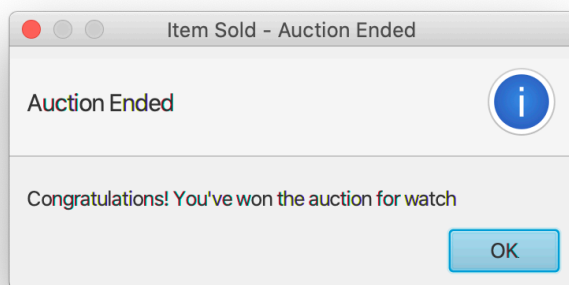
**You can also view your Bid History by visiting that tab!**



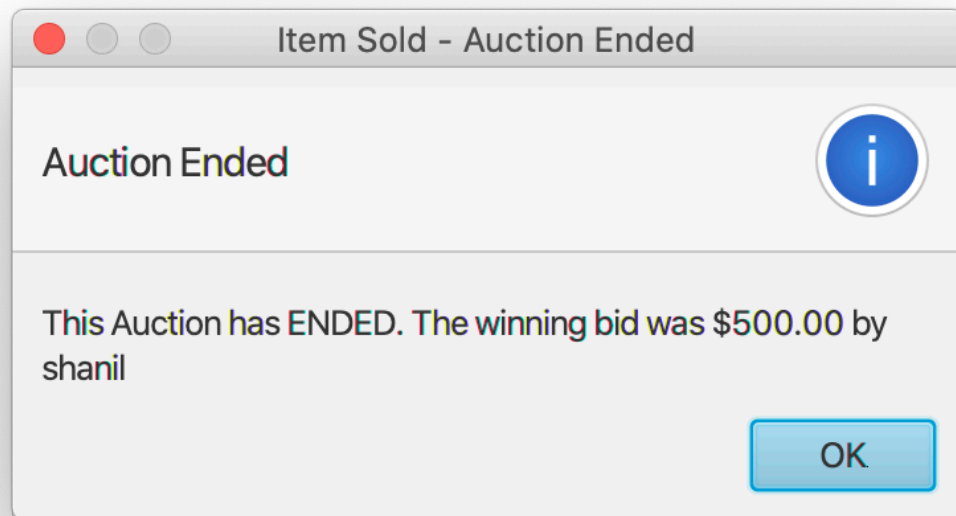
**Refresh your memory with private account information by going to the Account Info Tab**



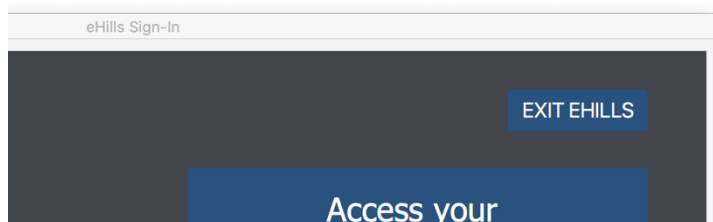
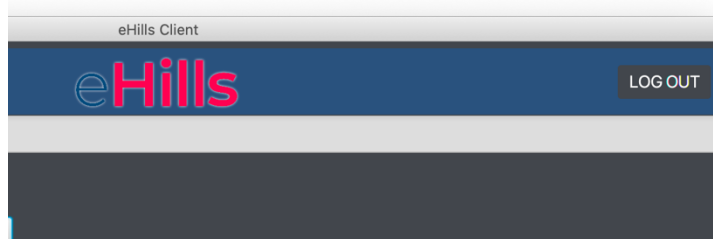
**If you have won a bid, congratulations! You'll be identified and notified.**



Or else, you'll be notified of your loss :(



Exit by clicking LOG OFF and then EXIT EHILLS





# Programmer's Point of View

## Code History

I started this program very soon after its release with a beautiful and complicated GUI on SceneBuilder. I learned that this flow was difficult, but I worked then on setting up functional bi-directional networking, the shared classes, and a custom string commands system to communicate. I then added additional functionalities:

### Additional Functionalities

- Minimum starting price on each Item
- A High Limit (Buy it Now) for each item
- Customer sees their own bid history
- Custom item descriptions
- Nice GUI
- Using a cloud server to host users and items database for initialization
- Custom Encryption of Passwords

## Code Structure

### Shared Classes

- Bid.java used to represent individual bids
- Item.java used to represent items for sale
- User.java used to represent individual users and authentication
- MasterDB.java a master database synced across clients and the server
- Parser.java processes each command request
- SJEncryption.java a custom encryption system

### Server Workflow

- Reads Users and Items from the internet database files `items.txt` and `users.txt` at `djshanzee.com/shanil_private/`
- Sets up networking at **port 8000** with an `ObjectOutputStream` for the main writer and an `ObjectInputStream` for each client reader
- As the server receives client requests, it parses in order received, the request commands.
- Once it has parsed the request, it notifies all clients of the update. (ie. a new highest bid, authorized User, etc)

### Client Workflow

- Sets up networking at **host: localhost, port: 8000** with an `ObjectOutputStream` for outgoing commands writer and an `ObjectInputStream` for incoming commands reader.
- It then prompts the user with the sign-in page

- Once it's received a positive authorization from the server, the Client closes the Sign-In window and pulls up the Bidding window
- Here, bidding on an item sends a message to the server, or displays an error
- The Bidding window refreshes every second with the latest information from the Server

## Table of Commands

### Client Sending, Server Receiving

Command	Purpose
C_AUTH + username + password	Authorize this user with this password
C_REFRESH	Sync Databases Request
C_INIT	Initialize my Database
C_BID + user + item + amount	Process this bid by this user on this item of this amount

### Server Sending, Client Receiving

Command	Purpose
S_AUTH + username + authorization	User has/doesn't have access
S_REFRESH + itemInfo + bidInfo	Sync your database with these items and bids
S_INIT + itemInfo + bidInfo	Initialize your database with these items and bids
S_BID + user + item + bid + acceptance	A bid by user on item of bid has/hasn't been accepted by the server

## Table of Users (current testcase)

Username	Password
shanil	jasani
sister	cool
admin	pass
greg	left
myTA	nice

## Table of Items (current testcase)

name	description	Starting Price	Ending Price
phone	iphone x in in excellent condition, comes with black phone case and broken charger	\$300	\$500
computer	macbook pro from 2022, with all software an ece student at UT needs	\$500	\$1,250
class	though ee422c is an invaluable class, this auction is the highest price bc it rocks	\$1,010	\$10,101
watch	apple watch series 100 is so out of this future it'll tell you what next week's lunch is	\$200	\$500
bed	big mattress for big sleep, you have to come get it though	\$500	\$3000

## Special Thanks to the Following Sources

Closing a window <https://stackoverflow.com/questions/13567019/close-fxml-window-by-code-javafx>

Opening a window <https://o7planning.org/en/11533/opening-a-new-window-in-javafx>

Alert button <https://www.geeksforgeeks.org/javafx-alert-with-examples/>  
<https://stackoverflow.com/questions/43031602/how-to-set-a-method-to-a-javafx-alert-button>

Choice Box and Listener [https://docs.oracle.com/javafx/2/ui\\_controls/choice-box.htm](https://docs.oracle.com/javafx/2/ui_controls/choice-box.htm)

Splitting Strings with Delimiter <http://www.codebind.com/java-tutorials/java-example-split-string-array-java/>

Time Delay <https://stackoverflow.com/questions/24104313/how-do-i-make-a-delay-in-java>

Timeline Animation <https://stackoverflow.com/questions/26916640/javafx-not-on-fx-application-thread-when-using-timer>

Reading from a File [https://www.w3schools.com/java/java\\_files\\_read.asp](https://www.w3schools.com/java/java_files_read.asp)

Reading a File from the Internet <https://stackoverflow.com/questions/6259339/how-to-read-a-text-file-directly-from-internet-using-java>

Format as Dollar <https://stackoverflow.com/questions/13791409/java-format-double-value-as-dollar-amount>