

## Table of Contents

1 Data Types.....	2
1.1 User.....	2
1.2 Item.....	2
1.3 Rating.....	3
1.4 Bid.....	3
1.5 Auction.....	3
2 Business Logic Constraints.....	4
2.1 User Registration and Login Constraints:.....	4
2.2 Item Listing Constraints:.....	4
2.3 Auction Constraints:.....	4
2.4 Bid Constraints:.....	4
2.5 Rating Constraints:.....	5
2.6 Search Constraints:.....	5
3 Task Decomposition / Abstract Code.....	6
3.1 Log In.....	6
3.2 Register.....	7
3.3 View Main Menu.....	8
3.4 List Item.....	9
3.5 Search for Item.....	10
3.6 View Item.....	12
3.7 Edit Description.....	14
3.8 View Ratings.....	15
3.9 Delete Rating.....	16
3.10 View Auction Results.....	17
3.11 Rate Item.....	18
3.12 Cancel Auction.....	19
3.13 Bid/Buy Item.....	21
3.14 View Category Reports.....	22
3.15 View User Reports.....	23
3.16 View Top Rated Items.....	25
3.17 View Auction Statistics.....	26
3.18 View Canceled Auction Details.....	27
3.19 View Item Listing/Results.....	28

## 1 DATA TYPES

### 1.1 User

Attribute	Data Type	Nullable	Notes
Username	String	Not Null	Unique identifier
Password	String	Not Null	
First Name	String	Not Null	
Last Name	String	Not Null	
User Type	String	Not Null	Either “Regular” or “Admin”
Position	String	Null	

### 1.2 Item

Attribute	Data Type	Nullable	Notes
Item ID	Integer	Not Null	Auto generated unique key
Name	String	Not Null	
Condition	List<String>	Not Null	Fixed List
Category	List<String>	Not Null	Fixed List
Description	String	Not Null	
Starting Bid	Decimal	Not Null	
Min. Sale Price	Decimal	Not Null	
Get it Now Price	Decimal	Null	Optional
Auction Length	List<Integer>	Not Null	Fixed values
Is Returnable	Boolean	Not Null	

### 1.3 Rating

Attribute	Data Type	Nullable	Notes
Date and Time	DateTime	Not Null	
Comments	String	Null	Optional comments
Number of Stars	List<Integer>	Not Null	Scale of 0 to 5

### 1.4 Bid

Attribute	Data Type	Nullable	Notes
Bid Time	DateTime	Not Null	Partial identifier
Bid Amount	Decimal	Not Null	2 decimal places

### 1.5 Auction

Attribute	Data Type	Nullable	Notes
Winner	String	Null	Can be null if canceled or item does not sell
End Time	DateTime	Not Null	
Sale Price	Decimal	Null	Can be null if canceled or item does not sell
Is Canceled	Boolean	Not Null	
Cancellation Reason	String	Null	Optional for the Admin

## 2 BUSINESS LOGIC CONSTRAINTS

### 2.1 User Registration and Login Constraints:

- The password provided from the user must match the password stored in the database during login.
- During registration, the password and the password confirmation provided by the user must match.
- Users who are new must register an account with BuzzBid before attempting to login.

### 2.2 Item Listing Constraints:

- All required fields must be filled in for an item to be listed (Item Name, Description, Category, Condition, Starting Bid, Minimum Sale Price, Auction Length, and optionally, Get It Now Price).
- The Starting Bid must be less than or equal to the Minimum Sale Price, and both must be less than the optional Get It Now Price.
- Auction Length must be one of the predefined options (1, 3, 5, or 7 days).
- Categories and Conditions are fixed lists and cannot be freely entered by all users, only administrators.
- Item ID is automatically generated and cannot be edited by the user.

### 2.3 Auction Constraints:

- Auction End Time is calculated based on the Auction Length from the item submission time.
- If a Get It Now Price is provided, it can immediately end the auction if selected by a buyer.
- The sale price is determined by the highest bid at the end of the auction if it meets or exceeds the Minimum Sale Price unless the Get It Now option is used.
- If an auction is canceled, the sale price and winner fields should be left blank. Auctions that terminate due to having no bids can be differentiated from those which were canceled by admin users by reading the isCancelled attribute.
- Only auctions that have ended should be displayed in the view Auction Results.

### 2.4 Bid Constraints:

- Users must enter a bid higher than the current high bid.

- Bids cannot be entered for auctions that have already ended, or those which are canceled.

## 2.5 Rating Constraints:

- Users who win an auction for an item can rate the item only once and on a scale from zero to five stars.
- Ratings for items with identical names (including case) should be grouped together, and the average star rating should be calculated and displayed.
- Ratings should be displayed sorted by the date and time they were entered, with the most recent ratings listed first.
- Administrators can delete ratings; a 'Delete' option is visible to them on rating views, and the delete action removes the rating from the database and refreshes the page.

## 2.6 Search Constraints:

- Search results must match all provided criteria.
- If a minimum price is specified, items returned must have a current high bid of at least this amount or use the Starting Bid price if no bids are present.
- If a maximum price is specified, items returned must have a current high bid less than or equal to this price, or a Get It Now Price less than or equal to the maximum price if no bids are present.
- Items returned must meet the selected condition or better if a condition is specified.

### 3 TASK DECOMPOSITION / ABSTRACT CODE

#### 3.1 Log In



##### Task Decomposition

<b>Lock Type</b>	Read only from <b>User</b> -> username, password
<b>Number of Locks</b>	1
<b>Enabling Conditions</b>	Triggered by user clicking the “login” button
<b>Frequency</b>	High
<b>Consistency (ACID)</b>	Not critical
<b>Subtasks</b>	Mother task is not needed. No decomposition needed.

##### Abstract Code

- User enters *username* and *password* input fields.
- If the user’s record exists in the database, run a query to retrieve the user’s \$password.
- Confirm data validation. If user’s record and \$password matches *password* input:
  - Data validation is successful and clicking the ***Login*** button presents the user with the **Main Menu** screen.
  - Else, the data validation is unsuccessful and an error message is displayed.
- Regardless of provided input, if the user clicks the Register button, the user is sent to the **Register** form.

### 3.2 Register



#### Task Decomposition

<b>Lock Type</b>	Write to <b>User</b> -> username, password, first name, last name
<b>Number of Locks</b>	1
<b>Enabling Conditions</b>	Triggered by new user clicking on “Register”
<b>Frequency</b>	Low
<b>Consistency (ACID)</b>	Critical, order is critical to avoid multiple users sharing the same username
<b>Subtasks</b>	Mother task is not needed. No decomposition needed.

#### Abstract Code

- User enters *First Name, Last Name, Username, Password* and *Confirm Password* input fields.
- User returns to the **Register** form for corrections and is presented with an informative error message, upon any of the following scenarios:
  - If the user does not fill all input fields.
  - *Username* input matches existing \$username in database, and is therefore, not unique.
  - *Password* and *Confirm Password* inputs are not equal.
- Clicking the ***Cancel*** button returns the user to the **Login** form.
- If registration is successful (i.e, data validation is successful based on the above criteria and user does not encounter an error message), then:
  - Clicking the ***Register*** button sends the user to the **Main Menu** screen.
  - A new user record is created in the database. These users lack administrative privileges.

### 3.3 View Main Menu



#### Task Decomposition

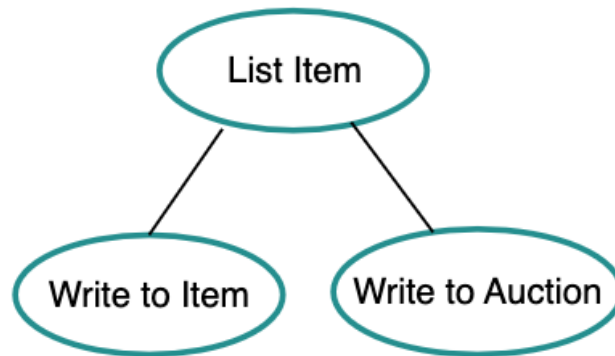
<b>Lock Type</b>	Read only from <b>User</b> -> firstname, lastname, position
<b>Number of Locks</b>	1
<b>Enabling Conditions</b>	Triggered by successful login to BuzzBid
<b>Frequency</b>	High, same as Login
<b>Consistency (ACID)</b>	Not critical
<b>Subtasks</b>	Mother task is not needed. No decomposition needed.

#### Main Menu

- Run the **View Main Menu** task to query information about the user to determine whether or not they hold administrative privileges.
  - If they do not, show only *Search for Items*, *List Item*, *View Auction Results* tabs.
  - If they do, then in addition to the above, show *Category Report*, *User Report*, *Top Rated Items*, *Auction Statistics*, *Canceled Auction Details* tabs.
- User is able to select next action from the following choices:
  - Clicking the *Search for Items* button – jump to the **Search for Items** task.
  - Clicking the *List Item* button – jump to the **List Item** task.
  - Clicking the *View Auction Results* button – jump to the **View Auction Results** task.
  - Clicking the *Category Report* button – jump to the **View Category Report** task.
  - Clicking the *User Report* button – jump to the **View User Report** task.
  - Clicking the *Top Rated Items* button – jump to the **View Top Rated Items** task.
  - Clicking the *Auction Statistics* button – jump to the **View Auction Statistics** task.
  - Clicking the *Canceled Auction Details* button – jump to the **View Canceled Auction Details** task.



## 3.4 List Item

Task Decomposition

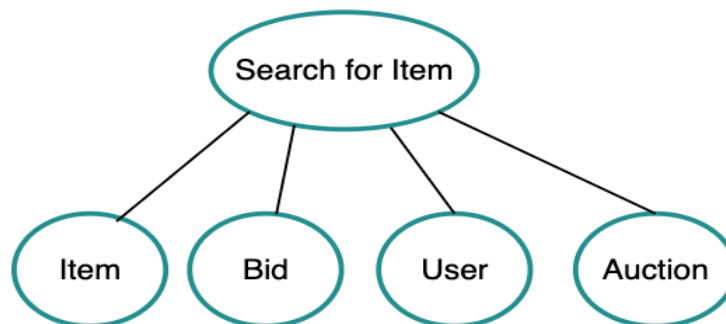
<b>Lock Type</b>	Write to <b>Item</b> -> Name, Condition, Category, Description, Starting Bid, Min Sale Price, Get it Now Price, Auction Length, Is Returnable Write to <b>Auction</b> -> End Time, Is Canceled
<b>Number of Locks</b>	2
<b>Enabling Conditions</b>	Triggered by clicking on “List Item” in Main Menu
<b>Frequency</b>	Low. Users are expected to buy/bid items significantly more than listing them.
<b>Consistency (ACID)</b>	Critical, to prevent multiple items from sharing the same Item ID
<b>Subtasks</b>	Mother task is needed. Task decomposition is needed for the subtasks: <ul style="list-style-type: none"> <li>- Write to <b>Item</b></li> <li>- Write to <b>Auction</b></li> </ul>

Abstract Code

- User clicked on the *List Item* tab from the Main Menu screen.
- Run the **List Item** task. Query the \$category, \$condition and \$auctionLength information from the database to populate fixed drop down lists.
- Display the following text (with the exception of *Returns Accepted*) input fields and include their respective constraints:
  - *Item Name*
  - *Description*
  - *Starting Bid* – limited to two decimal places. Must be positive.
  - *Minimum Sale Price* – limited to two decimal places. Must be positive.

- *Get It Now Price* – limited to two decimal places. Must be positive. User may choose to leave this blank.
- *Returns Accepted* – checkbox input field. User has the option to leave this unchecked.
- For successful verification, the form must follow the following constrictions:
  - All data fields except for *Get It Now Price* and *Returns Accepted* must be filled correctly. No error messages should occur.
  - *Get It Now Price* input must be higher than the *Starting Bid* input.
- Clicking the *Cancel* button returns the user to **Main Menu** form.
- If invalid data is entered, the item will not be listed. Instead, an error message will be displayed and the user will be returned to the **List Item** form to make corrections.
- Assuming successful data validation, clicking the *List My Item* button sends the user to the **View Item Listing** task.
  - Additionally, the item details will be added to the database along with a system generated unique \$itemID.

### 3.5 Search for Item



#### Task Decomposition

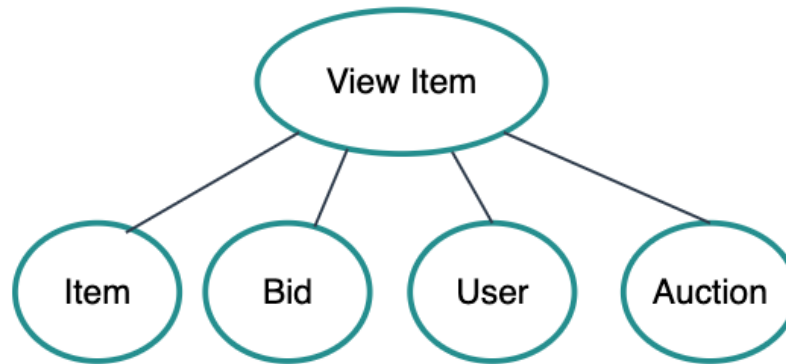
<b>Lock Type</b>	Read only lock from <b>Item</b> -> Item ID, Name, Get It Now Price Read only lock from <b>User</b> -> username Read only lock from <b>Auction</b> -> End Time Read only lock from <b>Bid</b> -> Bid amount
<b>Number of Locks</b>	4
<b>Enabling Conditions</b>	All are triggered by user clicking “Search for Items” in Main Menu
<b>Frequency</b>	High
<b>Consistency (ACID)</b>	Not critical, order is not critical

Subtasks	<p>Mother task is required with the following subtasks:</p> <ul style="list-style-type: none"> <li>- Read from <b>Item</b></li> <li>- Read from <b>User</b></li> <li>- Read from <b>Auction</b></li> <li>- Read from <b>Bid</b></li> </ul>
----------	--

### Abstract Code

- User clicked on the *Search for Item* tab from the **Main Menu** screen.
- Run the **Search for Item** task. Query the \$category and \$condition information from the database to populate fixed drop down lists. These lists also contain blank rows in the case that the user does not want to include the search criteria.
- Users have *Keyword*, *Minimum Price* and *Maximum Price* text input fields.
  - *Minimum Price* input must be lower than the *Maximum Price* input. If the user does not follow this restriction, display an error message once the user clicks the **Search** button.
- Clicking the **Cancel** button returns the user to the **Main Menu** screen.
- Clicking the **Search** button sends the user to the **Search Results** screen, only if:
  - User enters no criteria at all (i.e, does not provide any form input and does not choose any option from the drop down list(s)).
  - User enters multiple, valid search criteria. Only items that match all the criteria specified will be shown in the **Search Results** form.
- After the user inputs valid search criteria in the **Search for Item** form and clicks the *Search* button, run the **Search** task: query for information about the item.
  - Specifically, for each item, find the: \$itemID, item \$name, current \$bidAmount, \$username of the user with the highest bid, \$getItNowPrice and the auction \$endTime.
- Queried information is stored in a list and sorted by auction \$endTime. Results are displayed with descending order of the earliest auction \$endTime.
  - If there is no current \$bidAmount, display dashes for Current Bid and Username fields instead.
  - If no \$getItNowPrice was specified, display dash instead.
- Upon clicking an item name button, jump to that item's listing.
- Upon clicking the *Back to Search* button, return to the **Search for Item** form.

## 3.6 View Item

Task Decomposition

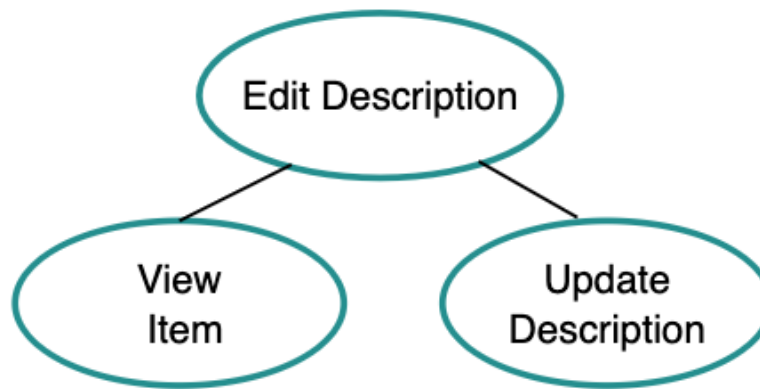
<b>Lock Type</b>	Read only lock from <b>Item</b> -> Item ID, Name, Description, Category, Condition, Is Returnable, Get It Now Price Read only lock from <b>Auction</b> -> End Time Read only lock from <b>User</b> -> username Read only lock from <b>Bid</b> -> Bid Time, Bid Amount
<b>Number of Locks</b>	4
<b>Enabling Conditions</b>	All are triggered by user clicking the Item Name of an item
<b>Frequency</b>	High
<b>Consistency (ACID)</b>	Not critical, order is not critical
<b>Subtasks</b>	Mother task is required with the following subtasks: <ul style="list-style-type: none"> <li>- Read from <b>Item</b></li> <li>- Read from <b>User</b></li> <li>- Read from <b>Bid</b></li> <li>- Read from <b>Auction</b></li> </ul>

Abstract Code

- User clicked on the *Item Name* tab from Search Results screen, or the *List My Item* button from the List Item form.
- Run the **View Item** task to query information about the item details, such as: its \$itemID, \$itemName, \$description, \$category, \$condition, \$isReturnable option, the \$getItNowPrice, and the auction \$endTime, as well as the bid history.

- Display the above listed item details and populate the minimum bid amount below the *Your Bid* text input field.
  - If the `$getItNowPrice` was not provided by the user when listing the item, this field remains blank. Subsequently, the *Get It Now* button is not available for viewing.
  - If the `$getItNowPrice` was provided, run the **Bid/Buy Item** task.
    - Clicking the *Get It Now* button sets the current user as the auction winner. The auction end date is set to the date and time the *Get It Now* button was pressed. This information is updated and added to the database.
- Display the bidding information if available, namely the: `$bidAmount`, `$bidTime` and the bidder's `$username`.
  - If no bids have been placed or item was canceled by the AdminUser, the bid information remains unpopulated.
  - If bid history is available, store the above listed variables and sort by `$bidTime`. Display the top 4 bids in descending order of `$bidTime`, with the latest bid listed first.
- Clicking the *View Ratings* button takes the user to the **View Ratings** screen and begins the **View Ratings** task.
- Users have a *Your Bid* text input field. If the current user's username matches the seller's username, then this input field will not be visible.
  - Input has to be at least \$1 above the latest bid price. If the `$getItNowPrice` has been set, input must also fall below that price.
    - If the input isn't valid, display an error message.
  - If the user attempts to bid higher than the `$getItNowPrice`, an error message will be displayed that suggests the user clicks the *Get It Now* button instead of bidding.
- Run the **Edit Description** task; query the current user's information as well as the item's information. If the current user's `$username` and the seller's `$username` match, then this tab will be visible to them. If not, this tab will be invisible.
  - Clicking the *Edit Description* tab takes the user to the **Edit Description** window.
- Clicking the *Close* button returns the user to the **Search Results** screen.
- Administrative users are able to see the *Cancel This Item* button and run the **Cancel Auction** task. Clicking this button will provide a **Cancellation Reason** prompt.
- Assuming the *Your Bid* input is valid, clicking *Bid On This Item* button refreshes the current **Item Description** screen with the updated input. The user's bid is updated in the database and saved with the item details.

## 3.7 Edit Description

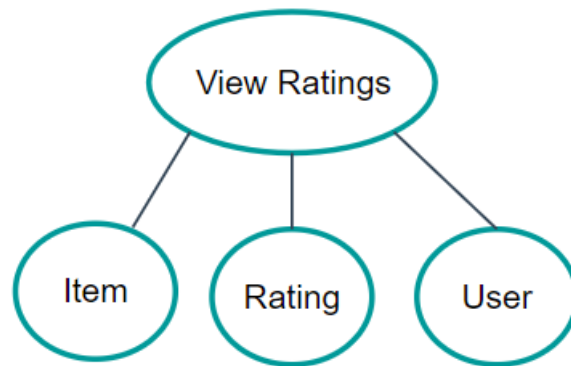
Task Decomposition

<b>Lock Type</b>	Read only lock from <b>Item</b> -> Item ID, Description Write only lock on <b>Item</b> -> Description
<b>Number of Locks</b>	2
<b>Enabling Conditions</b>	All are triggered by an admin user or a regular user who listed the item selects “Edit Description” from the “View Item”
<b>Frequency</b>	Low, same as List Item
<b>Consistency (ACID)</b>	Not critical, order is critical
<b>Subtasks</b>	Mother task is required. Task decomposition is required for the subtasks: <ul style="list-style-type: none"> <li>- View Item</li> <li>- Update description</li> </ul>

Abstract Code

- Current user’s \$username matched the seller’s \$username, which allowed this tab to be visible. (If the \$username of the two users had not matched, this tab would have remained invisible). User then clicked the *Edit Description* tab.
- Run the **Edit Description** task. Query and display existing \$description and provide a blank *New Description* input box.
- Clicking the *Save* button overwrites the previous description, closes the window and returns the user to the **Item Listing** screen.
- Clicking the *Cancel* button returns the user to the **Item Listing** screen.

## 3.8 View Ratings

Task Decomposition

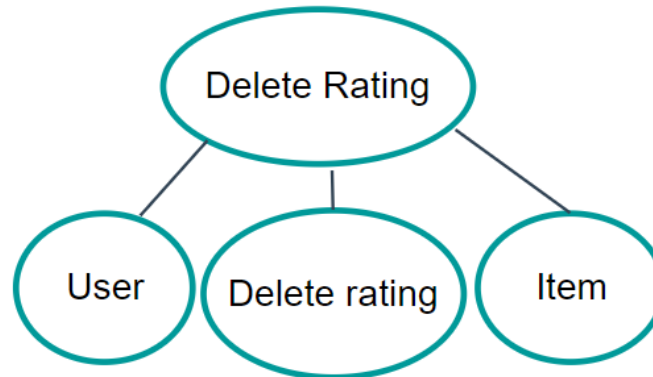
<b>Lock Type</b>	Read only lock from <b>Item</b> -> Item ID, name Read only lock from <b>Rating</b> -> Date and Time, Comments, Number of Stars Read only lock from <b>User</b> -> username
<b>Number of Locks</b>	3
<b>Enabling Conditions</b>	Triggered by a user selecting “View Ratings” from “View Item”
<b>Frequency</b>	High
<b>Consistency (ACID)</b>	Not critical, order is not critical
<b>Subtasks</b>	Mother task is required. Task decomposition is required for subtasks: <ul style="list-style-type: none"> <li>- Read from Item</li> <li>- Calculate average rating</li> <li>- Retrieve most recent ratings</li> </ul>

Abstract Code

- User clicked on the *View Ratings* tab on the Item Description form.
- Run the **View Ratings** task; query information such as the \$itemID, \$name and its previous \$numberOfStars.
  - Store the rating information (rater’s \$username, \$numberOfStars given, \$dateAndTime and if applicable, \$comment) in a list. Sort the list by rating \$dateAndTime. Display the ratings in descending order, with the most recent rating listed first.
  - If \$comments was not provided, leave the comment area blank.
  - Calculate all previous \$numberOfStars and display that as the average rating, rounded to one decimal place.

- Clicking the *Close* button returns the user to the **Item Description** form.

### 3.9 Delete Rating



#### Task Decomposition

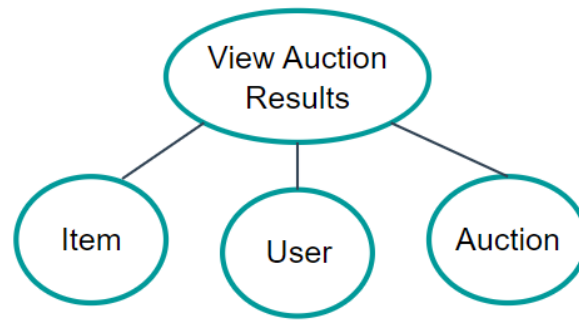
<b>Lock Type</b>	Read only lock from <b>User</b> -> username Read only lock from <b>Item</b> -> Item ID Write only lock on <b>Rating</b> -> Date and Time, Comments, Number of Stars
<b>Number of Locks</b>	3
<b>Enabling Conditions</b>	All 3 are triggered by an admin user selecting “Delete” from the Item Ratings view
<b>Frequency</b>	Low
<b>Consistency (ACID)</b>	Critical, order is critical
<b>Subtasks</b>	Mother task is required, task decomposition required for subtasks: <ul style="list-style-type: none"> <li>- Read username</li> <li>- Read Item ID</li> <li>- Delete rating</li> </ul>

#### Abstract Code

- User clicked on the **View Ratings** tab on the **Item Description** form.
- Run the **View Ratings** task; query information about the user and the item.
  - If the user is an administrative user, they will also see a *Delete* tab on the top of each rating. By clicking, administrators remove the rating from the database and refresh the page.
- Clicking the *Close* button returns the user to the **Item Description** form.



## 3.10 View Auction Results

Task Decomposition

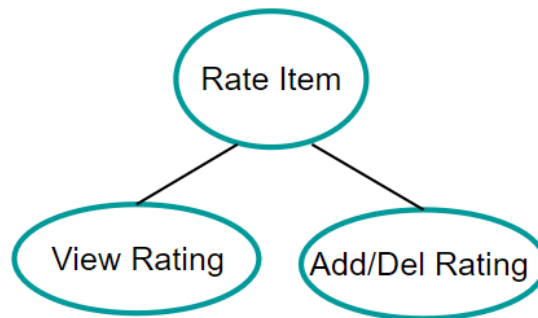
<b>Lock Type</b>	Read only lock from <b>Item</b> -> Item ID, Name Read only lock from <b>Auction</b> -> Winner, End Time, Sale Price Read only lock from <b>User</b> -> username
<b>Number of Locks</b>	3
<b>Enabling Conditions</b>	All are triggered by user selecting “View Auction Results” from Main Menu
<b>Frequency</b>	High
<b>Consistency (ACID)</b>	Not critical, order is not critical
<b>Subtasks</b>	Mother task is required with the following subtasks: <ul style="list-style-type: none"> <li>- Read from <b>Item</b></li> <li>- Read from <b>User</b></li> <li>- Read from <b>Rating</b></li> </ul>

Abstract Code

- User clicked on the *View Auction Results* button on the **Main Menu**.
- Run the **View Auction Results** task. Query item information regarding the \$itemID, item \$name, the final sale price, the \$winner \$username and the auction \$endTime.
  - Display \$itemID and an *Item Name* tab to the item \$name.
- For each item where the auction has ended, determine the sale price and winner.
  - If the item was purchased with the \$getItNowPrice, set \$salePrice to the \$getItNowPrice and the \$winner to the purchaser's \$username.
  - If the highest bid at the end of the auction is  $\geq$  \$minimumSalePrice, set \$salePrice to the highest bid amount. Set \$winner to highest bidder's \$username.

- If no bids were made, or none met the \$minimumSalePrice, then both \$salePrice and \$winner will be left blank.
- If the item wasn't sold, then both \$salePrice and \$winner will be left blank.
- Update the auction results depending on the scenarios above before displaying.
- Auction \$endTime is used to display the results in descending order, with the most recently ended auction listed first.
- In the event that the auction was canceled by the administrator, the \$salePrice and \$winner will be updated to null, the auction \$isCanceled will be updated to True, and the auction \$endTime will be updated to the time of cancellation.
- Clicking on an *Item Name*'s tab will take the user to that item's Item Description screen.
- Clicking the *Done* button returns the user to the Main Menu screen.

### 3.11 Rate Item



#### Task Decomposition

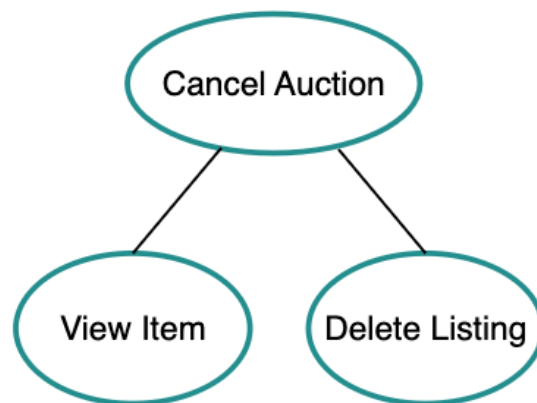
<b>Lock Type</b>	Read only lock from <b>Item</b> -> Item ID, name Read only lock from <b>Rating</b> -> Date and Time, Comments, Number of Stars Read only lock from <b>User</b> -> username Write only lock on <b>Rating</b> -> Date and Time, Comments, Number of Stars
<b>Number of Locks</b>	4
<b>Enabling Conditions</b>	All are triggered when the user who won the auction selects "View Ratings" from "View Item"
<b>Frequency</b>	Low, same as List Item
<b>Consistency (ACID)</b>	Critical, order is critical
<b>Subtasks</b>	Mother task is required. Task decomposition is required for subtasks:

	<ul style="list-style-type: none"> <li>- View Ratings</li> <li>- Add/del new rating</li> </ul>
--	--

### Abstract Code

- User clicked on the *Rate This Item* button on the **Item Description** page, or the *View Ratings* tab on the **Item Results** page. The former is only possible if the user has the \$winner attribute and has not previously rated the item; the latter is available to all users.
- Run the **Rate Item** task: Query the \$itemID, \$name and calculate the average rating across \$itemID, with their respective \$numberOfStars, to view existing ratings.
- Run the **View Ratings** task: Query and display the \$comments, rater's \$username, rating's \$dateAndTime and \$numberOfStars for all previous ratings made for the item.
- User may choose to add their own rating via the *Comments* text input box and by providing a *My Rating* star rating input.
  - The user's input for star rating and comment will be updated as their \$numberOfStars and \$comment, respectively.
  - Clicking the *Rate This Item* button will send the user to the **Item Rating** screen.
- If the user has previously rated, the *Comments* and the *My Rating* input field will be missing. Instead, run the **View Ratings** task.
  - If the user's \$username matches the \$username of a previous \$comments, then the user will see a **Delete My Rating** tab as well.
  - Clicking this tab deletes the rating from the database, refreshes the page and renders the user's \$comments for that \$itemID as blank. This also sends the user back to the **Rate Item** form.
- Upon clicking the *Close* button, return to the **Item Description** page.

### 3.12 Cancel Auction



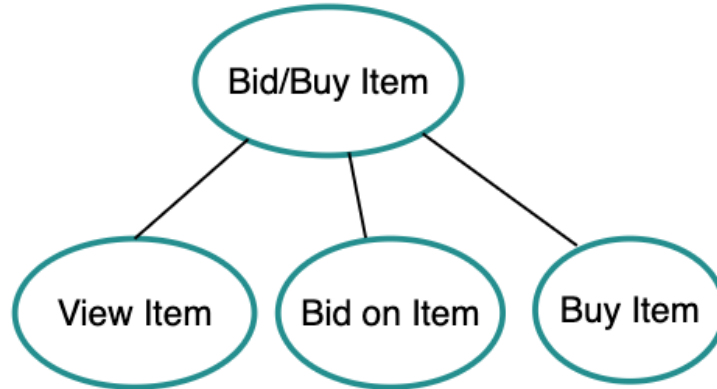
Task Decomposition

<b>Lock Type</b>	Read only lock from <b>Item</b> -> Item ID, Name, Description, Category, Condition, Is Returnable, Get It Now Price Read only lock from <b>Auction</b> -> End Time Read only lock from <b>User</b> -> username Read only lock from <b>Bid</b> -> Bid Time, Bid Amount Write only lock on <b>Auction</b> -> Is Canceled, Cancellation Reason, End Time, Winner, Sale Price
<b>Number of Locks</b>	5
<b>Enabling Conditions</b>	All are triggered by an admin user selecting “Cancel this Item” from “View Item”
<b>Frequency</b>	Low
<b>Consistency (ACID)</b>	Critical, we ensure integrity by reflecting cancellation across related tables and interfaces
<b>Subtasks</b>	Mother task required. Task decomposition required for subtasks: <ul style="list-style-type: none"> <li>- View Item</li> <li>- Cancel item auction listing</li> </ul>

Abstract Code

- Admin selected the *Cancel This Auction* button on the Item Description page. Prior to clicking this button, the system confirmed that the user has \$position.
- Run **Cancel Auction** task. Store the time when the administrative user clicked the button as auction’s \$endTime. Query the auction information and update its status to \$isCancelled.
- User is prompted with a Cancellation Reason Prompt screen that contains a single *Cancellation Reason* text input field. If input is provided, store as \$cancellationReason.
- Regardless of whether or not the user filled in the input field, the user is able to click the *Close* button and is redirected back to the Item Description page.

## 3.13 Bid/Buy Item

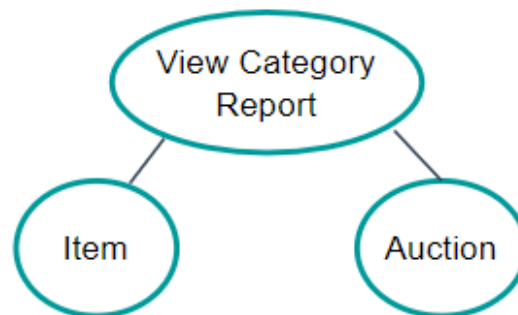
Task Decomposition

<b>Lock Type</b>	Read only lock from <b>Item</b> -> Item ID, Name, Description, Category, Condition, Is Returnable, Get It Now Price Read only lock from <b>Auction</b> -> End Time Read only lock from <b>User</b> -> username Read only lock from <b>Bid</b> -> Bid Time, Bid Amount Write only lock on <b>Bid</b> -> Bid Time, Bid Amount Write only lock on <b>Auction</b> -> End Time, Sale Price, Winner
<b>Number of Locks</b>	5
<b>Enabling Conditions</b>	All read only locks are triggered by a user navigating to the “View Item” page. The write only lock on <b>Bid</b> is triggered by a user entering a bid higher than the existing current high bid. If the user selects “Get It Now”, no bid is entered, and the write only lock on <b>Auction</b> is triggered.
<b>Frequency</b>	High
<b>Consistency (ACID)</b>	Critical, order is critical. Operations must be atomic to ensure bids do not overlap
<b>Subtasks</b>	Mother task required. Task decomposition required for subtasks: <ul style="list-style-type: none"> <li>- View Item</li> <li>- Bid on Item</li> <li>- Buy Item</li> </ul>

### Abstract Code

- User clicked on the item tab from **Search Results**.
- Clicking the ***View Ratings*** tab takes the user to **Item Ratings**.
- Clicking the Edit ***Description*** button provides the user with a popup input field, provided the current user matches with the username of the listing user.
  - User has the option to click the ***Close*** button to close the prompt, but can also provide input to replace the current description. If the user chooses to provide input, that input is stored as \$newDescription and is updated as that item's \$description.
- Clicking the ***Get It Now*** button takes the user to the **Auction Result** page, as the user has immediately won the bid.
  - The user is now the \$winner of that \$itemID.
- Clicking the ***Close*** button returns the user to the **Search Results** form.
- Clicking the ***Bid On This Item*** button updates **Item for Sale** form with the most recent bid, provided the bid is valid.
  - System authenticates if data is valid if it is at least one dollar higher than the current highest bid.
  - If the bid is not valid, the user is provided with an error message that suggests the user should click the ***Get It Now*** button instead of bidding again.
- The ***Cancel This Item*** button is available only to AdminUsers. When clicked, users are provided a **Cancellation Reason Prompt** form.

### 3.14 View Category Reports



### Task Decomposition

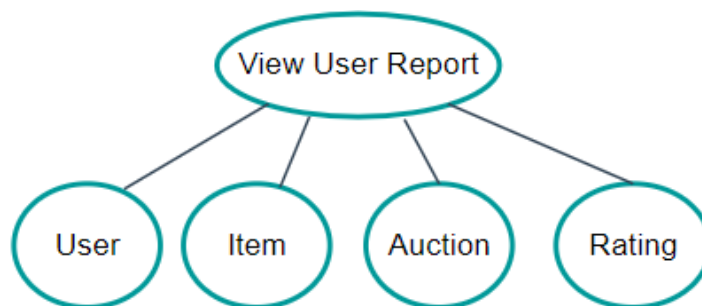
<b>Lock Type</b>	Read only lock from <b>Item</b> -> Category, Get It Now Price, Item ID Read only lock from <b>Auction</b> -> Is Canceled
<b>Number of Locks</b>	2

<b>Enabling Conditions</b>	All are triggered by an admin user clicking the “Category Report” button from the Main Menu
<b>Frequency</b>	Low
<b>Consistency (ACID)</b>	Not critical, order is not critical
<b>Subtasks</b>	<p>Mother task is required. Task decomposition is required for the following subtasks:</p> <ul style="list-style-type: none"> <li>- Read from <b>Item</b></li> <li>- Read from <b>Auction</b></li> </ul>

### Abstract Code

- User clicked on the *Category Report* tab from the **Main Menu**. User is verified by the system to have \$position.
- Run the **View Category Report** task: query information about all items except for those who have auction \$isCanceled. For each \$category:
  - Count the number of unique \$itemID. Display that number in the Total Items Row for that category.
  - Find the lowest \$getItNowPrice across all the \$itemID and display that number in the Min Price row.
  - Find the highest \$getItNowPrice across all the \$itemID and \$category and display that number in the Min Price row.
  - Average the \$getItNowPrice across all the \$itemID, excluding those that do not possess \$getItNowPrice.
- Clicking the *Done* button returns the AdminUser to the **Main Menu**.

### 3.15 View User Reports



Task Decomposition

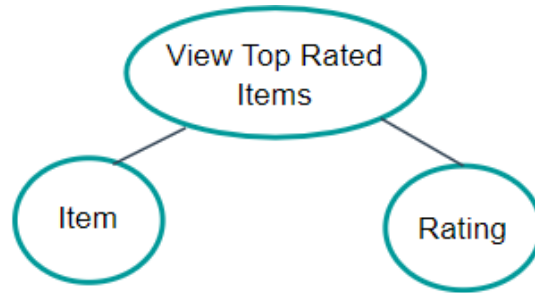
<b>Lock Type</b>	Read only lock from <b>User</b> -> Username Read only lock from <b>Item</b> -> Item ID, Condition Read only lock from <b>Auction</b> -> Winner Read only lock from <b>Rating</b> -> Date and Time
<b>Number of Locks</b>	4
<b>Enabling Conditions</b>	All are triggered by an admin user clicking the “User Report” button from the Main Menu
<b>Frequency</b>	Low
<b>Consistency (ACID)</b>	Not critical, order is not critical
<b>Subtasks</b>	Mother task is required with the following subtasks: <ul style="list-style-type: none"> <li>- Read from <b>User</b></li> <li>- Read from <b>Item</b></li> <li>- Read from <b>Auction</b></li> <li>- Read from <b>Rating</b></li> </ul>

Abstract Code

- User clicked on the *User Report* tab from the **Main Menu**. User is verified by the system to have \$position.
- Run the **View User Report** task: query information about all users activities.
  - Aggregating the total number of items listed by each user, identifying unique \$itemID and their \$condition.
  - Summarizing the total number of auctions won by examining the \$winner attribute.
  - Counting the total number of \$comments each user has given and ensuring those entries are unique by focusing on the \$dateAndTime.
  - Determining the most frequent \$condition of items listed by each user by selecting the lowest-quality condition in the event of a tie.
- Format results to display each user's username along with the aggregated statistics: the total number of items listed, items sold, auctions won, ratings given, and the predominant condition of their listings.
- Sort results by the total number of listings per user in descending order.
- Clicking the *Done* button returns the AdminUser to the **Main Menu**.



## 3.16 View Top Rated Items

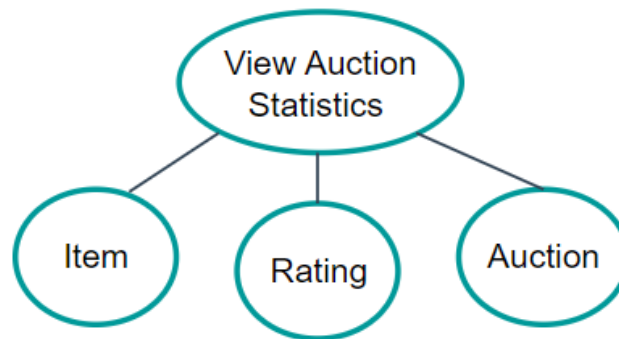
Task Decomposition

<b>Lock Type</b>	Read only lock from <b>Item</b> -> Item ID, Name Read only lock from <b>Rating</b> -> Number of Stars
<b>Number of Locks</b>	2
<b>Enabling Conditions</b>	All are triggered by an admin user clicking the “Top Rated Items” button from the Main Menu
<b>Frequency</b>	Low
<b>Consistency (ACID)</b>	Not critical, order is not critical
<b>Subtasks</b>	Mother task is required with the following subtasks: <ul style="list-style-type: none"> <li>- Read from <b>Item</b></li> <li>- Read from <b>Rating</b></li> </ul>

Abstract Code

- User clicked on the *Top Rated Items Report* tab from the **Main Menu**. User is verified by the system to have \$position.
- Run the **View Top Rated Items Report** task: query information about top 10 rated items within the database. Specifically,
  - Iterate through the \$itemID and their \$numberOfStars. Skip items that do not have \$numberOfStars. Use each item’s \$numberOfStars to determine an average rating and store that in a list.
  - For each item in that list, count the total number of \$numberOfStars.
  - Display those items by descending \$numberOfStars ranking.
    - In cases where items have identical average ratings, they are then ordered by item \$name in ascending order.
- Clicking the *Done* button returns the AdminUser to the **Main Menu**.

## 3.17 View Auction Statistics

Task Decomposition

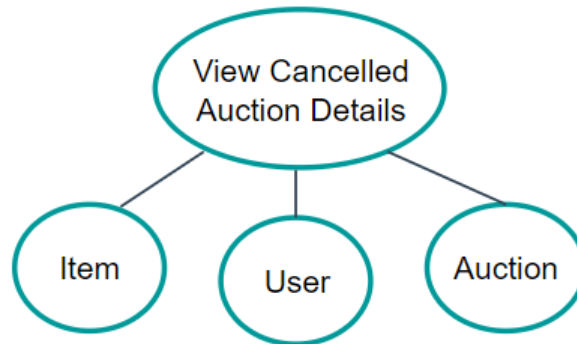
<b>Lock Type</b>	Read only lock from <b>Item</b> -> Item ID Read only lock from <b>Rating</b> -> Date and Time Read only lock from <b>Auction</b> -> Is Canceled, Winner, End Time
<b>Number of Locks</b>	3
<b>Enabling Conditions</b>	All are triggered by an admin user clicking the “Auction Statistics” button from the Main Menu
<b>Frequency</b>	Low
<b>Consistency (ACID)</b>	Not critical, order is not critical
<b>Subtasks</b>	Mother task is required with the following subtasks: <ul style="list-style-type: none"> <li>- Read from <b>Item</b></li> <li>- Read from <b>Rating</b></li> <li>- Read from <b>Auction</b></li> </ul>

Abstract Code

- User clicked on the Auction *Statistics Report* tab from the **Main Menu**. User is verified by the system to have \$position.
- Run the **View Auction Statistics** task: query information about all auctions and items within the database.
  - Retrieve the total count of active auctions where \$isCanceled is false and \$endTime is greater than the current time.
  - Summarize completed auctions where \$endTime has passed and \$isCanceled is false.
  - Count the number of won auctions by checking for entries with a non-null \$winner that are not canceled.

- Gather the number of canceled auctions where \$isCanceled is true.
- Aggregate the total number of rated items by counting \$comments associated with that \$username. Do not include those that do not possess \$numberOfStars.
- Clicking the *Done* button returns the AdminUser to the Main Menu.

### 3.18 View Canceled Auction Details

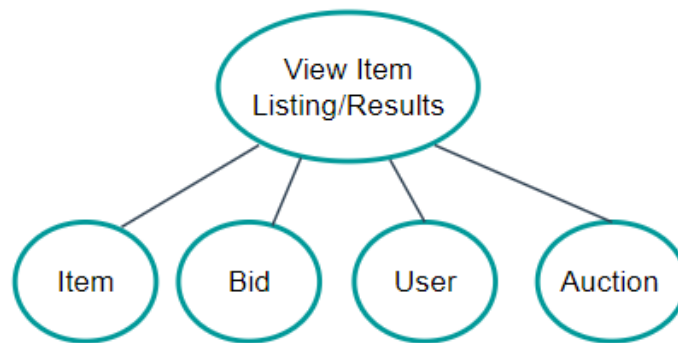


#### Task Decomposition

<b>Lock Type</b>	Read only lock from <b>Item</b> -> Item ID Read only lock from <b>Auction</b> -> End Time, Cancellation Reason, Is Canceled Read only lock from <b>User</b> -> username
<b>Number of Locks</b>	3
<b>Enabling Conditions</b>	All are triggered by an admin user clicking the “Canceled Auction Details” button from the Main Menu
<b>Frequency</b>	Low
<b>Consistency (ACID)</b>	Not critical, order is not critical
<b>Subtasks</b>	Mother task is required with the following subtasks: <ul style="list-style-type: none"> <li>- Read from <b>Item</b></li> <li>- Read from <b>Auction</b></li> <li>- Read from <b>User</b></li> </ul>

Abstract Code

- User clicked on the *Canceled Auction Details Report* tab from the **Main Menu**. User is verified by the system to have \$position.
- Run the **View Canceled Auction Details** task: query database for all auctions canceled by AdminUsers.
  - For each auction that has the \$isCanceled variable, display the \$itemID, the \$username of the associated user, the \$cancellationReason, and the auction's \$endTime.
  - Displayed items are sorted by descending \$itemID.
- Clicking the *Done* button returns the AdminUser to the **Main Menu**.

**3.19 View Item Listing/Results**Task Decomposition

<b>Lock Type</b>	Read only lock from <b>Item</b> -> Item ID, Name, Description, Category, Condition, Is Returnable, Get It Now Price Read only lock from <b>Auction</b> -> End Time Read only lock from <b>User</b> -> username Read only lock from <b>Bid</b> -> Bid Time, Bid Amount
<b>Number of Locks</b>	4
<b>Enabling Conditions</b>	All are triggered by user clicking the Item Name of an item
<b>Frequency</b>	High
<b>Consistency (ACID)</b>	Not critical, order is not critical
<b>Subtasks</b>	Mother task is required with the following subtasks: <ul style="list-style-type: none"> <li>- Read from <b>Item</b></li> <li>- Read from <b>User</b></li> </ul>

	<ul style="list-style-type: none"><li>- Read from <b>Bid</b></li><li>- Read from <b>Auction</b></li></ul>
--	---

#### Abstract Code

- User clicked on the *Item Name* tab from the **Auction Results** page. User is verified by the system to have \$position.
- Run the **View Item Listing/Results** task: query information about the item that the user clicked.
  - For the item, display the \$itemID, item \$name, \$description, \$category, \$condition, \$isReturnable , \$getItNowPrice, and query the auction's \$endTime.
  - Query bid history, display \$bidAmount, \$bidTime, and the bidder's \$username for all bids on the item, sorted by \$bidTime (recent to oldest).
- For auction winners, provide functionality to rate the item and view previous ratings.
  - Clicking the *View Rating* button sends the user to the **Item Rating Form**.
- Clicking the *Close* button returns the user to the **Search Result** screen.