Table of Contents:

Data Types

User

Attribute	**Data type**	**Nullable**
FirstName	String	Not Null
LastName	String	Not Null
UserName	String	Not Null
Password	String	Not Null

Administrative User

Attribute	**Data type**	**Nullable**
Position	String	Not Null

Item

Attribute	**Data type**	**Nullable**
ItemName	String	Not Null
ItemDescription	String	Not Null
Condition	String	Not Null
Returnable	Boolean	Not Null
StartingBid	Float	Not Null
MinimumSalePrice	Float	Not Null
GetItNowPrice	String	Null

Auction

Attribute	**Data type**	**Nullable**
AuctionEndTime	Date	Not Null
FinalSalePrice	Float	Null
CancelledReason	String	Null

Bid

Attribute	**Data type**	**Nullable**
BidAmount	String	Not Null
TimeofBid	Date	Not Null

Rating

Attribute	**Data type**	**Nullable**
RatingDateTime	Date	Null
Comments	String	Null
Star	Integer	Null

Category

Attribute	**Data type**	**Nullable**
CategoryName	String	Not Null

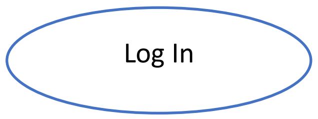
Business Logic Constraints

1. All Users who are new to BuzzBid must register first.
2. Users who already have an existing BuzzBid account will not be able to register again.
3. A User who listed the item for sale cannot bid on the same item.
4. Only the user who listed an item can edit the description of that item.
5. Only the user who won the auction for an item will be able to rate that item.
6. Item rating can be deleted only by the user who added it or by any Administrative user.
7. Winner and SalePrice of an Auction will be NULL until the auction is over.
8. Users cannot bid on item or edit the item description after the item's auction is over.

Task Decomposition with Abstract Code

Login User

Task Decomposition



Lock Types: Read-only on User table.
Number of Locks: Single
Enabling Conditions: None
Frequency: Around 200 logins per day
Consistency (ACID): not critical, order is not critical.
Subtasks: Mother Task is not needed. No decomposition needed.

Abstract Code

- User enters *Username* ('$Username'), *Password* ('$Password') input fields.
- If data validation is successful for both *username* and *password* input fields, then when ***Login*** button is clicked:
    - If User record is found but user.password ! = '$Password':
        - Go back to **Login** form, with error message.
    - Else:
        - Store login information as session variable '$Username'.
        - Go to the **Main Menu** form.
- Else *email* and *password* input fields are invalid, display **Login** form, with error message.

Register User

Task Decomposition

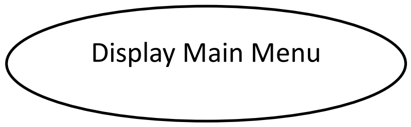
Lock Types: Write-only on User table.
Number of Locks: Single
Enabling Conditions: Trigger from login
Frequency: Around 20 new user registrations per day
Consistency (ACID): Critical, should not add duplicate usernames to User table.
Subtasks: Mother Task is not needed. No decomposition needed.

Abstract Code

- User enters *FirstName* ('$FirstName'), *LastName* ('$LastName'), *Username* ('$Username'), *Password* ('$Password') and *ConfirmPassword* ('$ConfirmPassword') input fields.
- If data validation is successful for *Username*, *Password* and *ConfirmPassword* input fields, then when ***Register*** button is clicked:
    - If an existing record with same *Username* is found:
        - Go back to the **Register** form, with the error message.
    - Else:
        - Add a new record in User database with all details from input fields.
        - Store login information as session variable '$UserID'.
        - Go to the **Main Menu** form.
- Else any of the input fields are invalid, display **Register** form, with error message.

Display Main Menu

Task Decomposition



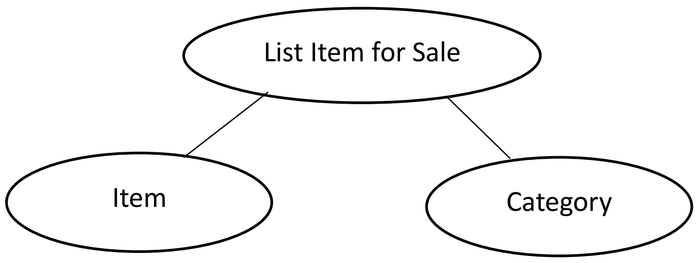
Lock Types: Lookup Username and Position, all are Read-only.
Number of Locks: Single
Enabling Conditions: Trigger by successful login or after listing an item
Frequency: User Detail and Menu Options have the same frequency.
Consistency (ACID): not critical, order is not critical.
Subtasks: Mother Task is not needed. No decomposition needed.

Abstract Code

- Show:
  - If the session variable '$Username' belongs to a RegularUser, show Auction Options only with below links:
    - ***"Search for Items", "List Item"*** and ***"View Auction Results"***
  - Else if the '$Username' belongs to an AdministrativeUser, show all the above Auction Options and include the links to Reports below:
    - ***"Category Report", "User Report", "Top Rated Items," "Auction Statistics"*** and ***"Cancelled Auction Details"***
- Upon clicking:
  - ***Search for Items*** button - Jump to the **Display Search Result** task.
  - ***List Item*** button - Jump to the **List Items for Sale** task.
  - ***View Auction Results*** button - Jump to the **Display Auction Results** task.
  - ***Category Report*** button - Jump to the **Display Category Information** task.
  - ***User Report*** button - Jump to the **Display User Information** task.
  - ***Top Rated Items*** button - Jump to the **Display Top Rated Items** task.
  - ***Auction Statistics*** button - Jump to the **Display Auction Statistics** task.
  - ***Cancelled Auction Details*** button - Jump to the **Display Cancelled Auction Details** task.

List Item for Sale

Task Decomposition



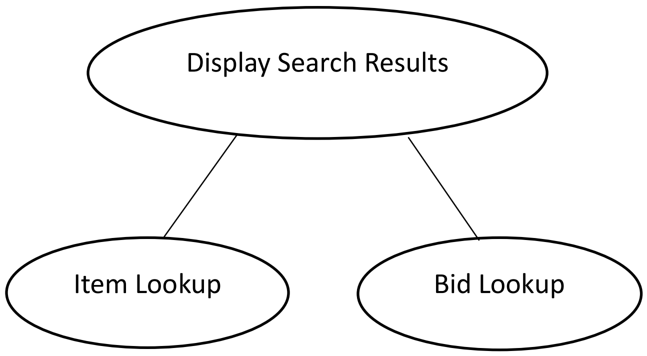
Lock Types: Read-Only lookup of Category and Write-Only to Item table.
Number of Locks: Several schema constructs are needed.
Enabling Conditions: Enabled from **Main Menu** form.
Frequency: Both have the same frequency.
Consistency (ACID): Not critical, Item cannot be looked up before creating it.
Subtasks: Category lookup should be done before writing into Item table. Mother Task is required for coordinating subtasks.

Abstract Code

- User clicked on ***List Item*** button from **Main Menu**.
- Run the **List Item for Sale** task:
  - Query for the list of available categories from the Category table.
  - Display the New Item for Auction form:
    - Input fields for *Item Name, Description, start auction bidding at, Minimum sale price, Get it Now price.*
    - Dropdown with query result for *Category*
    - Dropdown for *Condition* with values as New, Very Good, Good, Fair and Poor.
    - Dropdown with values of 1,3,5 and 7 days for *Auction ends in*.
    - Checkbox for *Returns Accepted?*
    - Buttons for ***Cancel*** and ***List My Item***
  - User enters item details in all the input fields and selects value from the dropdown.
  - If data validations are successful for all input fields:
    - When ***List My Item*** button is clicked:
      - Calculate AuctionEndTime from the current system time and *Auction ends in* value.
      - Fetch the session variable '$Username'.
      - Add a new record in Item table with all details from input fields, calculated AuctionEndTime and the session variable '$Username'.
      - Add a new record in Auction with item id and auction status as Active.
      - Go to the **Main Menu** form.
    - When ***Cancel*** button is clicked:
      - Go back to the **Main Menu** form.
  - Else if any of the input fields are invalid, display the **List Item for Sale** form, with the error message.

Display Search Result

Task Decomposition



Lock Types: Read-only on Item, Category, Bid tables.
Number of Locks: Several schema constructs are needed.
Enabling Conditions: Triggered from **Main Menu** form.
Frequency: High, searching items is one of the most common tasks in BuzzBid.
Consistency (ACID): Not critical, even if a new item is being added while user is searching.
Subtasks: Category lookup should be done before writing into Item table. Mother Task is required for coordinating subtasks.

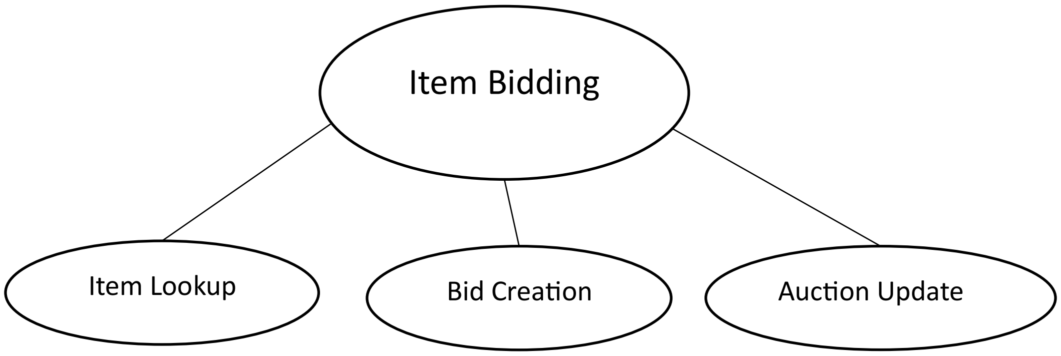
Abstract Code

- User clicked on ***Search for Items*** button from **Main Menu** form.
- Run the **Display Search Result** task: query for items matching the search criteria and display the results.
    - Find the list of available categories from the Category table.
    - Display the **Item Search** form with:
        - *Keyword, Minimum Price* and *Maximum Price* input fields.
        - *Category* dropdown with values from the Category table and a blank value
        - *Condition at least* dropdown with values including New, Very Good, Good, Fair, Poor and a blank value.
        - ***Cancel*** and ***Search*** buttons.
    - If data validation is successful for *Minimum Price* and *Maximum Price* input fields to accept either integer values or blank, then:
        - When ***Search*** button is clicked: build a search query based on the values from the **Item Search** form and run the consolidated query on Item table to get the matched list of items.
            - Find all items from Auction table where auction status is Active.
            - Create a search query to get these item details from Item table.
            - If the *Keyword* input field has text:
                - Add to the search query, to filter items where the case-sensitive text appears anywhere in Item Name or Description.
            - If a value except blank row is selected from *Category* dropdown:

      - Add to the search query, to filter items that match the selected category.
    - If the *Minimum Price* input field has text:
      - Add to the search query to filter items that have at least one Bid value and highest bid amount is of at least entered text value.
      - Add to the search query to filter items that have no Bid value, and the starting bid price is of at least the entered text value.
    - If the *Maximum Price* input field has text:
      - Add to the search query to filter items that have at least one Bid value and the highest bid amount is less than or equal to the entered text value.
      - Add to the search query to filter items that have no Bid value and the starting bid price lower than the entered text value.
    - If a value except blank row is selected from <u>*Condition*</u> dropdown:
      - Form a list of conditions with all values better than the selected one, including the selected condition.
      - Add to the search query to filter items that match the list of conditions.
    - Add to the search query to sort returned items in order of the auction end date with the auction ending soonest listed first.
    - Run the consolidated search query on Item table.
    - If the query has output from Item table, then:
      - Display **<u>View Search Results Listing</u>** form that includes:
        - List of items with ID, Item Name, Current Bid, High Bidder, Get It Now Price and Auction End date and time.
        - ***Back to Search*** button
      - When any one of the item names is clicked:
        - Store the corresponding ID as session variable '$ItemId'
        - Store the name as session variable '$ItemName'
        - Go to the **<u>Item for Sale</u>** form.
      - When ***Back to Search*** button is clicked:
        - Go back to **<u>Item Search</u>** form.
    - Else if the search query returned no output:
      - Go back to **<u>Item Search</u>** form, with error message.
  - When ***Cancel*** button is clicked, go to the **<u>Main Menu</u>** form.
- Else *Minimum Price* and *Maximum Price* input fields are invalid, display the **<u>Item Search</u>** form, with error message.

Item Bidding

Task Decomposition



Lock Types: Read-Only on Item and Category, Read & Write on Bid, Write-Only on Auction tables.
Number of Locks: Several different schema constructs are needed.
Enabling Conditions: Triggered from **Item Search** form when an item name is clicked.
Frequency: High, bidding on items is one of the most common tasks in BuzzBid
Consistency (ACID): Critical, when a user is bidding on item, no other user can bid on it.
Subtasks: Item Details and Latest Bids list lookup should be done before writing into Bid table. Mother Task is required for coordinating subtasks.

Abstract Code

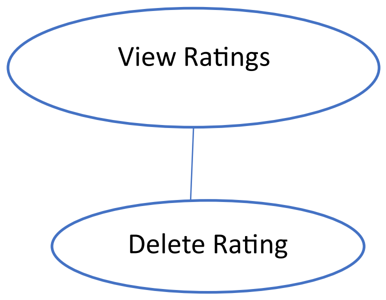
- User clicked on one of Item Name from **View Search Results Listing** form.
- Run the **Item Bidding** task: query for information about the item where the session variable '$ItemId' is the ID of the item, display information about the item including the latest four bids and write to Bid & Auction based on the details input from the user.
    - Run the **Item Lookup** subtask:
        - Find the current Item using the session variable '$ItemId'
        - Find the corresponding category name of the item from Category.
    - Display the **Item for Sale** form:
        - Display the item's ID, name, description, category name, condition.
        - If the item has true value for ReturnsAccepted, then:
            - Display Returns Accepted with a checkbox ticked.
        - Else:
            - Display Returns Accepted with a checkbox not ticked.
        - If the item has a value for GetItNowPrice, then:
            - Display the GetItNowPrice value
            - Display a ***Get It Now!*** button
        - Find the AuctionEndTime of the current item from the Auction table using the session variable '$ItemId':
            - Display Auction Ends with the value of Auction End Time
        - Find the latest four bids of current Item that is highest and most recent from Bid:
            - Display Bid Amount and Time of Bid.
            - Find the corresponding user of bid from User.
            - Display Username

- Display *Your Bid* input field.
  - Find the highest bid of the current item and calculate acceptable minimum bid.
  - Display the minimum bid as informational text below the input field.
- Find the user who listed the item from Item table.
- If the session variable '$Username' is different from user who listed the item:
  - Display ***Bid on this Item*** and ***Close*** buttons.
  - Display link to **View Ratings** form.
- Else if '$Username' is same as the user who listed the item, then:
  - Display links to **View Ratings** and **Edit Description** forms.
  - Display ***Close*** button only.
- If the session variable '$Username' belongs to an AdministrativeUser, then display all the buttons and form links along with:
  - ***Cancel This Item*** button.

- When the ***Get It Now!*** Button is clicked:
  - Run the **Bid Creation** subtask to create a new entry in Bid with:
    - GetItNowPrice value as BidAmount.
    - Find the current system time and store it as TimeofBid.
    - Store the session variable '$Username' as Username in Bid.
  - Find the current item in Auction where ID is the session variable '$ItemId'.
  - Run the **Update Auction** subtask to update the existing item in Auction with:
    - GetItNowPrice value as FinalSalePrice.
    - Find the current system time and store it as AuctionEndTime.
    - Store the session variable '$Username' as Winner in Auction.
    - Store the Auction status as Ended for this item.
- When ***Bid on this Item*** button is clicked:
  - If data validation is successful for *Your Bid* input field to accept a float value greater than or equal to minimum bid, then:
    - Run the **Bid Creation** subtask to create a new entry in Bid
  - Else *Your Bid* input field is invalid, display **Item for Sale** form with error message.
- When ***Cancel This Item*** button is clicked:
  - Run the **Cancel Auction** subtask to mark the item as cancelled in Auction:
    - Find current item in Auction where ID is the session variable '$ItemId'.
    - Find the current system time and store it as AuctionEndTime.
    - Store the Auction status as Cancelled for this item.
- Upon clicking:
  - **View Ratings** form link – Jump to **View Item Ratings** task.
  - **Edit Description** form link – Jump to **Edit Item Description** task.
  - ***Close*** button – Go back to the **View Search Results Listing** screen.

View Ratings

Task Decomposition



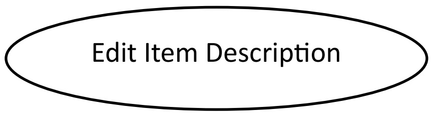
Lock Types: Read and Write on Rating table.
Number of Locks: Single
Enabling Conditions: Enabled from **Item for Sale** form.
Frequency: Medium, most users who are bidding on an item will view the item ratings.
Consistency (ACID): Not critical, even if rating is added/deleted while another user is looking at it.
Subtasks: Mother task is needed to coordinate rating lookup and delete rating subtasks. Rating lookup should be done before Delete Rating can execute.

Abstract Code

- User clicked on ***View Ratings*** link from **Item for Sale** form.
- Run the **View Ratings** task to query all ratings related to the current item and display:
  - Display the Item ID as the value from session variable '$ItemId'.
  - Display the Item Name as the value from session variable '$ItemName'.
  - Find all ratings from Rating where item name is same as session variable '$ItemName'.
  - Sort the ratings by the RatingDateTime with most recent rating listed first.
  - For each Rating from the sorted result:
    - Display the value of Username of the rating as Rated by.
    - Display value of RatingDateTime as the Date.
    - Display the rating comments.
    - Display 5 Stars and color the number of stars equal to value of that rating's Star.
    - If the session variable '$Username' belongs to an AdministrativeUser, then:
      - Display ***Delete*** link alongside the rating information.
  - Calculate the average value of all rating's Star values and round up to one decimal:
    - Display Average Rating as the calculated average value.
  - Display ***Close*** button.
- When ***Delete*** button is clicked against an item rating:
  - Run the **Delete Rating** subtask:
    - Delete the corresponding rating entry from Rating.
  - Re-run the **View Item Ratings** task.
- When ***Close*** button is clicked, go back to the **Item for Sale** form.

Edit Item Description

Task Decomposition



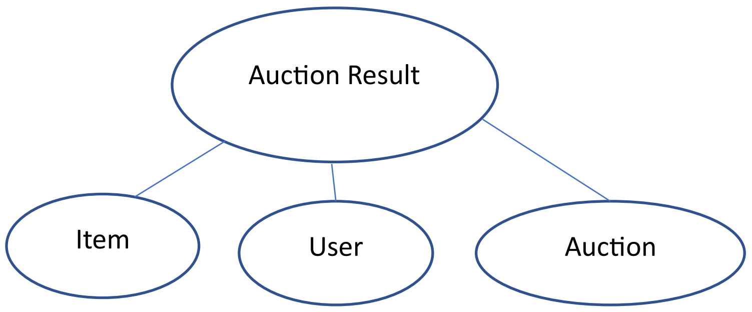
Lock Types: Write-only on Item table.
Number of Locks: Single
Enabling Conditions: Triggered from **Item for Sale** form only by the user who listed the item.
Frequency: Low.
Consistency (ACID): Not critical, even if item description is edited while another user is looking at it.
Subtasks: Mother Task is not needed. No decomposition needed.

Abstract Code

- User clicked on ***Edit Description*** link from **Item for Sale** form.
- Display a popup window with:
    - *Add description* input field for adding new description for the item.
    - ***Save*** and ***Cancel*** buttons.
- When ***Save*** button is clicked:
    - Store the text from *Add description* input field as the new description of the Item that matches the session variable '$ItemId'.
    - Go back to the **Item for Sale** form.
- When ***Cancel*** button is clicked, go to the **Item for Sale** form.

Display Auction Results

Task Decomposition



Lock Types: Read-only on Item, User, Auction tables.
Number of Locks: Several schema constructs are needed.
Enabling Conditions: Triggered from **Main Menu** form.
Frequency: Moderate
Consistency (ACID): Critical, only visible with auctions that have already ended, where updates should be accurately reflected on database

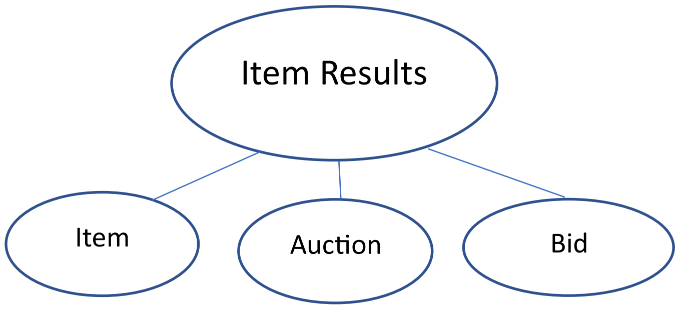
Subtasks: All tasks must be completed and be able to run simultaneously. Mother task coordinates retrieving data and display subtask together

Abstract Code

- Before trigger to generate auction result, proceed **Seach Item** for any auction that does not have a declared winner but have already end (based on the current time), then update those winner
- User clicked on ***View Auction Results*** button from **Main Menu** form.
- Run the ***View Auction Results*** task: query for items matching the search criteria and display the results.
    - **View Item t**ask
        - Display *ItemName* and *SalePrice*
            - *SalePrice is* determined by **Item Bidding** †ask
        - Enable for proceeding with **Display Item Result**
    - **View User** task
        - Displayed as Winner with *Username*
    - **View Auction** task
        - Displayed with *AuctionEndTime* that has been sorted with most recently ended auction first
  - When ***Cancel*** button is clicked, go to the **Main Menu** form.

Display Items Results

Task Decomposition



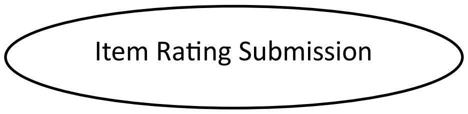
Lock Types: Read-only on Item, Auction, Bid tables.
Number of Locks: Several schema constructs are needed.
Enabling Conditions: Triggered from **Auction Results** form.
Frequency: Moderate
Consistency (ACID): Critical, only visible with proceeding **Display Auction Result** where auctions have been finished
Subtasks: All tasks must be completed beforehand. Mother task coordinates retrieving data and display subtask together

Abstract Code

- User clicked on ***Item Name*** button from **Auction Results** form.
- Run the ***View Item Results*** task: query for items matching the search criteria and display the results.
    - **View Item** task
        - Display *ItemID, ItemName, ItemDescription, Category, Condition, Returnable*
        - Price determined by **Item Bidding** task, whether it was *GetItNowPrice* or *MinimumSalePrice*
        - Enable for proceeding with **View Ratings**
    - **View Auction task**
        - Displayed with *AuctionEndTime*
    - **View Bid** task
        - Displayed with *BidAmount*, *TimeOfBid*, *Username*
  - When ***Cancel*** button is clicked, go to the **Auction Results** form.
  - Enable to proceed with View Ratings

Item Rating Submission

Task Decomposition



Lock Types: Write-only on Rating table.
Number of Locks: Single
Enabling Conditions: Triggered from **Item Results** form only by the user who won the item's auction.
Frequency: Medium, most users who won the item will add a rating for it.
Consistency (ACID): Not critical.
Subtasks: Mother Task is not needed. No decomposition needed.

Abstract Code

- User clicked on ***View Ratings*** link from **Item Results** form.
- Run the **View Ratings** task to query all ratings related to current item:
    - Display the rating results in **Rate Item** form.
    - Display ***Close*** button.
- If the ratings result has a rating where ItemId value is same as the session variable '$ItemId':
    - Display ***Delete My Rating*** link against that rating since it was rated by the current user.
- Else there are no ratings added by the current user for the item won:
    - Run the **Add Item Ratings** task to add a new rating for the item won, display:
        - *My Rating* field with 5 empty Stars and an option to color any number of them.
        - *Comments* input field for adding new description for the item.
        - Display ***Close*** and ***Rate this Item*** buttons.
        - When ***Rate this Item*** button is clicked:

        - If the data validation is successful for *Comments* input field, then add a new entry in Rating with all details from input fields:
            - Store the session variable `$ItemId`, '$ItemName'
            - Store the session variable '$Username' as RatedBy
            - Count the number of stars colored and store as Star.
            - Store the text in *Comments* input field as Comments.
        - Else *Comments* has invalid data, display **Rate Item** form with error message.
    - When ***Delete My Rating*** button is clicked:
        - Run the ***Delete Rating*** task to delete the entry from Rating.
- When ***Close*** button is clicked, go back to the **Item Results** form.

Display Category Information

Task Decomposition

Lock Types: Read only Lookup of Items and Category.
Number of Locks: Single
Enabling Conditions: Trigger by clicking on Category Reports link on main menu by admin user
Frequency: Low (accessed only by admin users)
Consistency (ACID): Not Critical.
Subtasks: No decomposition needed.

Abstract Code

- Admin User clicked on the Category Reports button from the **Main Menu**.
- Run the View Category Reports task: Query the information from the Items table and the category table
    - Query:
        - Find the ***Items*** and the Associated ***categories.***
        - Display Category Name as **Category**
        - Remove ***cancelled auction items*** on the returned items and categories
        - Aggregate Get it Now Price to obtain **Min price, Max price, Avg Price**
        - Count the list of items and display as **Total Items**
        - Sort by Category Name in ascending order under **Category**
    - Show:
        - Buttons for ***Done.***
    - If the Admin user click on Done button:
        - Return the Admin user to the main menu.

Display User Information

Task Decomposition

Lock Types: Read only Lookup of Users, Items, Auctions, Ratings
Number of Locks: Single
Enabling Conditions: Trigger by clicking on User Reports link on main menu by admin user
Frequency: Low (accessed only by admin users)
Consistency (ACID): Not critical.
Subtasks: No decomposition needed.

Abstract Code

- Admin User clicked on the User Reports button from the **Main Menu**.
- Run the View User Reports task: Query the information from the User table, Items table, Auctions table and the Rating table
    - Query: Upon clicking,
        - For all the users, count the listed Items and display as ***Listed.***
        - Total items sold by the user: For all the users, for each of the listed item with winner in Auctions, count the items in Auction as ***Sold.***
        - Total items won by the user: For all the users listed as Winners in Auction, count the items, and display as ***Won.***
        - *Total number of items user rated: For all the users, count the items with ratings provided and display as **Rated.***
        - *Most Frequent Condition:*
            - *For all the users and their listed items, Identify the max of count of the Condition of items*
            - In the event of a tie, the lowest quality condition should be used.
            - If the user has no listings, should display as N/A
    - Show:
        - Buttons for ***Done.***
    - If the Admin user click on Done button:
        - Return the Admin user to the main menu.

Display Top Rated Items

Task Decomposition

Lock Types: Read only Lookup of Items and Rating.
Number of Locks: Single
Enabling Conditions: Trigger by clicking on Top Rated Items link on main menu by admin user
Frequency: Low (accessed only by admin users)
Consistency (ACID): Not Critical.
Subtasks: No decomposition needed.

Abstract Code

- Admin User clicked on the Top Rated Items hyperlink from the **Main Menu**.
- Run the View Top Rated Items task: Query the information from the Items and the Ratings table
    - Query: upon clicking,
        - Find the ***Items*** and the Associated ***Ratings.***
        - Display Item Name as **Item Name.**
        - Display Average Rating (rounded by one place) for a given Item as **Average Rating.**
        - Display count of Rated by for a given Item as **Rating Count.**
        - Sort by Average Rating in descending order
        - List only top 10 Average Rating items
    - Show:
        - Buttons for ***Done.***
    - If the Admin user click on Done button:
        - Return the Admin user to the main menu.

Display Auction Statistics

Task Decomposition

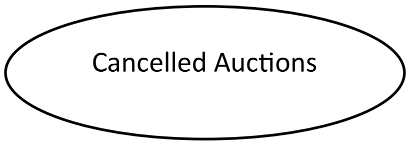
Lock Types: Read only Lookup of Items, Ratings and Auctions.
Number of Locks: Single
Enabling Conditions: Trigger by clicking on Auction Statistics link on main menu by admin user
Frequency: Low (accessed only by admin users)
Consistency (ACID): Not Critical.
Subtasks: No decomposition needed.

Abstract Code

- Admin User clicked on the Auction Statistics hyperlink from the **<u>Main Menu</u>**.
- Run the View Auction Statistics task: Query the information from the Items, Ratings and the Auctions table
  - Query: upon clicking,
    - Auctions Active: count of Actions where AuctionEnded is null in the Auctions
    - ***Auctions Finished:*** *count of Actions where AuctionEnded is not null, and the Auction is not cancelled.*
    - ***Auctions Won:*** *count of Actions where AuctionEnded is not null, and the Winner is not null*
    - ***Auctions Cancelled:*** *count of cancelled Auctions*
    - ***Items Rated:*** count of items which are Rated
    - ***Items not Rated:*** count of items which are not rated.
  - Show:
    - Buttons for ***Done.***
  - If the Admin user click on Done button:
    - Return the Admin user to the main menu.

<u>Display Cancelled Auction Details</u>

Task Decomposition



Lock Types: Read only Lookup of Items and Auctions/Bid.
Number of Locks: Single
Enabling Conditions: Trigger by clicking on Cancelled Auctions link on main menu by admin user
Frequency: Low (accessed only by admin users)
Consistency (ACID): Not Critical.
Subtasks: No decomposition needed.

Abstract Code

- Admin User clicked on the Cancelled Auctions hyperlink from the **<u>Main Menu</u>**.
- Run the View Auction Statistics task: Query the information from the Items and the Auctions table
    - Query: upon clicking,
        - ID: Display the ***Bid cancelled*** items from the Auctions
        - Listed by: Display the user who listed the cancelled auction items
        - Cancelled Date: Display the auction cancelled date
        - Reason: Display the reason for the auction cancellation
    - Show:
        - Buttons for ***Done.***
    - If the Admin user click on Done button:
        - Return the Admin user to the main menu.