# Table of Contents

# Data Types

### User

| Attribute | Data Type | Nullable |
|---|---|---|
| username | string | Not Null |
| password | string | Not Null |
| first_name | string | Not Null |
| last_name | string | Not Null |
| is_admin | boolean | Not Null |
| position | string | Null |

### Item

| Attribute | Data Type | Nullable |
|---|---|---|
| item_name | string | Not Null |
| description | string | Not Null |
| starting_bid_price | float | Not Null |
| min_sale_price | float | Not Null |
| get_it_now_price | float | Null |
| final_sale_price | float | Null |
| auction_length | integer | Not Null |
| auction_end_time | datetime | Null |
| returnable | boolean | Not Null |
| winner | string | Null |

### Rating

| Attribute | Data Type | Nullable |
|---|---|---|
| comment | string | Not Null |
| stars | integer | Not Null |
| time_added | datetime | Not Null |

### Bid

| Attribute | Data Type | Nullable |
|---|---|---|
| amount | float | Not Null |
| time | datetime | Not Null |

### Category

| Attribute | Data Type | Nullable |
|---|---|---|
| category | string | Not Null |

### Condition

| Attribute | Data Type | Nullable |
|---|---|---|
| condition | string | Not Null |

# Business Logic Constraints

- **Auction_length** must be constrained to have only the values 1,3,5, or 7 days.
- When placing a bid, the bid amount cannot be less than **Starting_bid**.
- **Minimum_sale_price** must be greater than the **Starting_bid.**
- The **Get_It_Now_Price** must be greater than or equal to **Minimum_sale_price.**
- For a User, if **is_admin** is true, then **position** may not be null.

# Task Decomposition and Abstract Code

## Logging into GTBay

### Task decomposition

**Lock Types:** Read-only on User table
**Number of Locks:** Single
**Enabling Conditions:** None
**Frequency:** Not known
**Consistency (ACID):** Consistency is not critical; order is not critical
**Subtasks:** Mother Task is not needed. No decomposition needed.


### Abstract code

If user clicks ***Register*** button:
    Go to **New User Registration** page
User provides *username* and *password*
If user clicks "**Login**" button:
    Look up User to see if the given *username* exists already:
        If found:
            Verify *password*

If password matches:
Go to **Main Menu** page
Else:
Display error message “Invalid Password” and return to **Login** page;
Else:
Display error message “username is not exist” and return to **Login** page for new registration;

# New User Registration

## Task decomposition

**Lock Types:** Write on User table
**Number of Locks:** Single
**Enabling Conditions:** User clicks on **Register** button
**Frequency:** Not known
**Consistency (ACID):** Consistency is critical since duplicate username is not allowed; order is not critical
**Subtasks:** Mother Task is not needed. No decomposition needed.


**Abstract code**
User clicked “**Register**”
button from the **Login** page
User enters *first name, last name, username, password*, and *confirm password* in the corresponding fields.
If “**cancel**” button is clicked:
Return to **Login** page and nothing will be inserted into database
If “**Register**” button is clicked:
There will be validation
If one of the fields is not filled, the error message will display “The field cannot be empty”.

If the field "*Password*" and "*Confirm Password*" are not identical, the error message will display "The password is not match".

Else:

Write the User table with the information provided

If *username* is exist in the database, the error message will display "The username is exist",

Return to **New User Registration page**.

Else

User with the information will be inserted into the database, the successful message will display "Registration is successful". The type of user should be regular user.

Go to **Main Menu** page

# Main Menu

**Task decomposition**

Lookup: One time read on User table
Enabling condition: Upon successful login or registration
Frequency: High
Consistency (ACID): Not critical; Order not critical
Mothertask: Not needed

**Abstract code**

User clicked "**Login**" button and "**Register**" button from the **Login** and **New User Registration** pages with valid inputs.
Show "**Search for Items**", "**List an Item for sale**", and "**View Auction Results**" link
Read the User table
If the current user is an administrative user:

Display position of the current user
Display "**View Category Report**" and "**View User Report**" links

Upon:

User clicks "**Search for Items**": go to **Searching for Items** page
User clicks "**List an Item for sale**": go to **Selling Item** page
User clicks "**View Auction Results**": go to **Auction Results** page
User clicks "**View Category Report**": go to **Category Report** page
User clicks "**View User Report**": go to **User Report** page

# Selling Item

**Task decomposition**

Lookup:

Two reads on Condition and Category tables

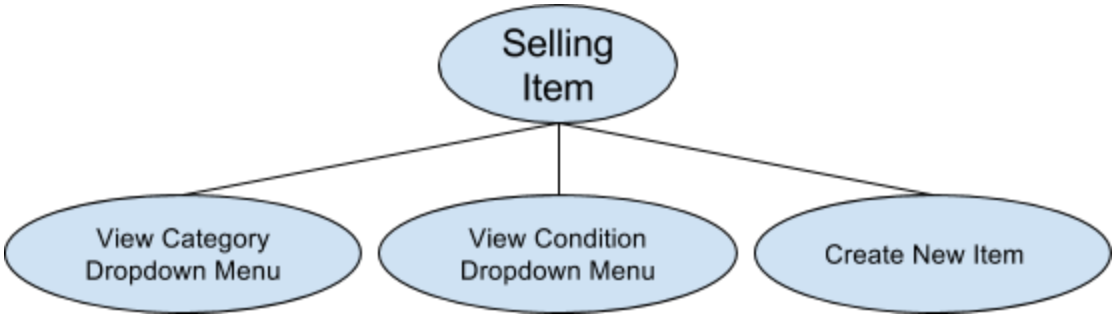
One write to the Item table

Enabling condition: Upon the "**List an Item for sale**" link on **Main Menu** page is clicked

Frequency: High

Consistency (ACID): Not critical; order is critical since the system has to read the condition and category table then write the Item table with provided information

Mothertask: Should be decomposed into three subtasks




**Abstract code:**

User clicked "**List an Item for sell**" button from the **Main Menu** page

Create dropdown menu for "*category*" and "*Condition*" by reading Category and Condition tables, respectively.

Display the **New Item for Auction form**

User enters *item name, description, starting bid, minimum sale price, auction length, and get it now price* in the corresponding fields, choose *category* and *condition* from dropdown menu, as well as specify *acceptance for return* in the checkbox.

If "**List My Item**" button is clicked:

Verify the information of all fields are valid

If one of the fields is not valid, the error message will display on the page.

If verified:

An unique Item ID will be generated and insert to the Item table, then Go to **Main Menu** page

Else:

Return to **Selling Item** page with meaningful error messages

If "**Cancel**" button is clicked:

Go to **Main Menu** page and no information will be insert into database.

# Searching for items

**Task decomposition**

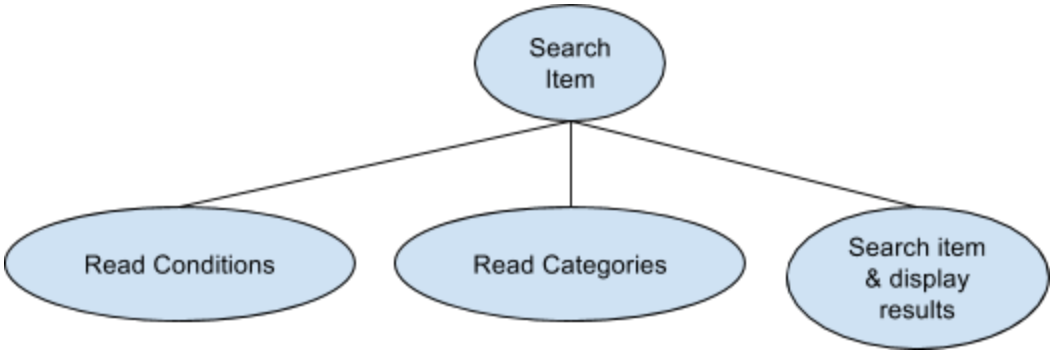
Lookup: Multiple reads on Category, Condition, Item, User, and Bid

Enabling condition: The "**Searching for Items**" link from **Main Menu** is clicked

Frequency: High

Consistency (ACID): Consistency is critical since the user want to see the most updated bidding of an item. Order is also critical since users need to provide criteria before the result can be shown.

Mothertask: Should be decomposed into subtasks




**Abstract code**

User clicked "**Searching for Items**" link from the **Main Menu** page

Create dropdown menu for "*Category*" and "*Condition at least*" by reading Category and Condition tables, respectively.

Display the **Item search form**

User enters keyword in the "*Keyword*" field

User can optionally select *category* and *condition*, as well as enter minimum and maximum prices to narrow down the searching.

If "**cancel**" button is clicked:

Return to **Main Menu** page

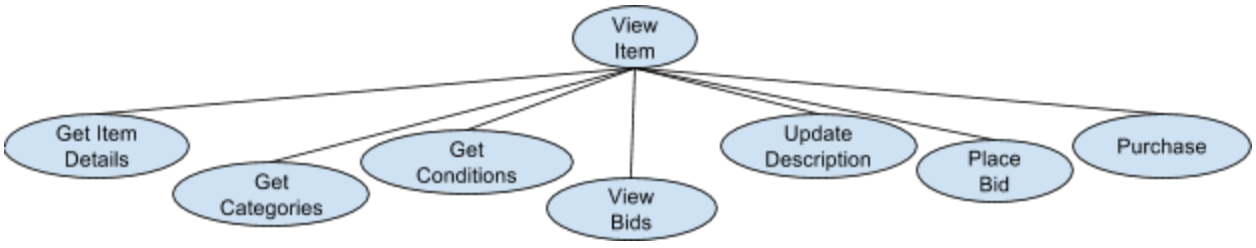
If "**Search**" button is clicked:

Read the Item, Bid, and User tables to pull out existing non-expired auctions satisfy the specified criteria

Items that match the search criteria will be displayed in the search results screen. In order for the item to be displayed in the search result, current bid of the item should be less than or equal to the *minimum price* value and the current bid amount should be less than or equal to the *maximum price*.
Go to **Search Results** page:
Display qualified auctions with the following information:
Item ID, item name, current bid, high bidder, get it now price, and the auction end time
If "**Back to Search**" button is clicked:
Return to **Searching for items** page
If user clicks on the **Item Name**:
Go to **Item Description** page

# View Item

**Task decomposition**
Lookup: Multiple reads on Item, Category, Condition, Bid, and
Possibly multiple updates to Item table
Enabling condition: The "**item name**" link from **Search Results** is clicked
Frequency: Different frequencies
Consistency (ACID): Consistency is critical since the user want to see the most updated bidding of the item. Order is not critical.
Mothertask: Should be decomposed into subtasks




**Abstract code**
User clicked the "**item name**" link on the **Search Result** page
Read Item table and pull out the item by Item ID
Read Condition and Category table to pull out condition and category of the item
Read Bid table and pull out the bids associated with the item's Item ID
Display the following details about the item:

Item ID, Item name, Description, Category, Condition, Return acceptance, Auction ending time and date, and the latest four bids sorted by bidding price (highest on top)

If Get it now price was specified for the item:

Display get it now price and "**Get it now**" button

If the user is the listing user:

Display "**Edit Description**" link

If the user clicks "**Edit Description**" link:

Display a popup window asking for new *description*

User enters new *description*

If user clicks "**Submit**" button:

Write the Item table with new description

Return to **Item Description** page

If user clicks "**Cancel**" button:

Return to **Item Description** page

If the user clicks "**Cancel**" button:

Return to **Search Results** page

If the user clicks "**View Ratings**" link:

Go to **Item Ratings** page

If the user clicks "**Get it now**" button:

End the auction: Compute auction_end_time and write it.

Write final_sale_price = get_it_now_price.

Write winner = username on Item.

Display appropriate message and return to **Search Results** page

If user enters a bid and clicks "**Bid On This Item**" button:

Read the bid table to check if the bid is at least one dollar higher than the current highest bid and less than the Get it now price:

If bid is qualified:

Write to the Bid table

Display appropriate message and return to **Search Results** page

If bid is higher than Get it now price:

Display popup window with error message suggesting to click "**Get it now**" button

Return to **Item Description** page

# Item Ratings

**Task decomposition**

Lookup: Multiple reads on Rating and Item tables
Possibly one write to Rating table
Enabling condition: The "**View Ratings**" link from **Item Description** page is clicked
Frequency: Different frequencies
Consistency (ACID): Consistency is not critical, even if someone is writing and submitting a review while the user is on this page. Order is critical, the current four most recent ratings should be display before the user can give a new rating.
Mothertask: Should be decomposed into subtasks

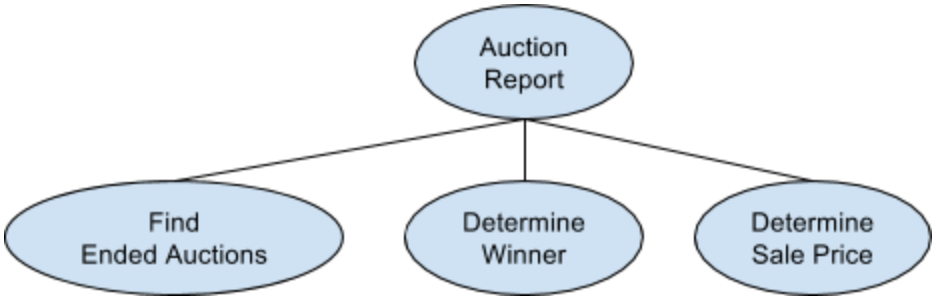

**Abstract code**

User clicked "**View Rating**" link on **Item Description** page
Read the Item, User, and Rating table to pull out all ratings associated with the Item ID
Display the following details about the ratings:
- Item ID, Item name, average rating acquired from calculating the mean of Number of stars from all associated ratings, and all associated ratings sorted by date (latest on top).
- Each associated rating displays the following details:
  - Username of the user who created the rating, rating date, and comments
  - If the rating was created by the current user:
    - Display "**Delete My Rating**" link
    - If user click "**Delete My Rating**" link:
      - This rating is deleted from the rating table
      - Refresh the **Item Ratings** page
  - If the user clicks "**Cancel**" button:
    - Return to **Item Description** page
  - User fills out the rating form

If the user clicks "**Rate This Item**" button:
    If the user has already rated this item before:
        Display error message and return to **Item Rating** page
    Else if all fields are entered:
        Write to the Rating table
        Return to **Item Rating** page

# Auction Results

**Task decomposition**
Lookup: Multiple reads on Item, Bid, and User tables and Update to the Item table.
Enabling condition: The "**Show Auction Result**" link from **Main Menu** is clicked
Frequency: Low
Consistency (ACID): Consistency is critical. Order is also critical since the ended auctions need to be found before determining winner and final sale price
Mothertask: Should be decomposed into subtasks




**Abstract code**
User clicked "**Show Auction Result**" link from **Main Menu** page
Read the Item table and pull out all items that has ended already
If a winner has not been determined yet, then:
    Update the Item table with the user who placed the highest bid as the winner.
    Update the Item table with the highest bid price as the final sale price
Sort the items by their auction ended date and time (latest on top)
For each item:
    Display the following details about the items:

Item ID, Item name, Sale price, Winner, and Auction ended date and time

If user clicks "**Done**" button:

Return to **Main Menu**

# GTBay Administrative Reports

## Category Report

**Task decomposition**

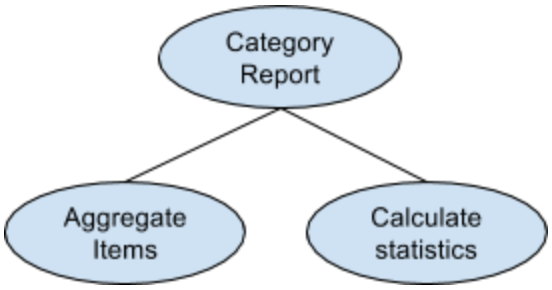
Lookup: Multiple reads on Item, Bid, and Category tables

Enabling condition: AdminUser can generate the report, and AdminUser can view the "**Show Auction Result**" link from **Main Menu** is clicked

Frequency: Low

Consistency (ACID): Consistency is critical. Order is also critical since the items need to be aggregated by category before all the calculations being performed.

Mothertask: Should be decomposed into subtasks




**Abstract Code**

Administrative user clicks "**View Category Report**" from the **Main Menu** page

Read the Category and Item tables and aggregate items by their Category

Sort the categories alphabetically

For each category:

Display the category name

Calculate and display total number of items

Calculate and display the minimum, maximum, and average get it now price of that category:

If an item does not have Get it now price:

Exclude from the calculation

If administrative use clicks "**Done**" button:

Return to **Main Menu** page

# User Report

**Task Decomposition**

Lookup: Multiple reads on User, Item, and Rating tables
Enabling condition: The "**Show User Result**" is displayed only to the Administrative user and the "The "Show User Result" link from **Main Menu** is clicked
Frequency: Low
Consistency (ACID): Consistency is critical. Order is critical since the items and ratings need to be aggregated before the calculations being performed
Mothertask: Should be decomposed into subtasks


**Abstract Code**

Administrative user clicks "**View User Report**" from the **Main Menu** page
Run **Auction Results** and store it in a separate "Ended Item" table
Read the User, Item, Rating, and Ended Item table to aggregate items and ratings by Username
For each user:
    Display Username
    Calculate and display the following:
        The number of items associated with the user
        The number of items sold:
            Read the Ended Item table and sum the number of items have Sale price and Winner
        The number of purchased items:
            Read the Ended Item table and sum the number of items with Winner equals to the User
        The number of ratings associated with the user
If administrative user clicks "**Done**":
    Return to **Main Menu** page

# Assumptions

**Calculate Auction Winner**
The winner of an auction can be determined in two scenarios:

1. When an user clicks “buy it now” button on the item description page, the auction ends immediately and the user becomes the winner of the auction (username stored as “winner” attribute of the item).
2. When an user clicks “auction results” link on the main menu page, if the winner of an ended auction has not been determined, the highest bidder becomes the winner and the username is stored as “winner” attribute of the item.

**Administrative Privilege**
A regular user can acquire administrative rights only through permission from database administrator. A position attribute will then be assigned to the administrative user.

**Tracking the items sold by user**
During the implementation phase, we will use a table to track the number of items sold by the user and this will be used in the User reports.
No of items that have winner attribute + number of items whose auction has ended and they have at least one bid > minimum sale price.

**Duplicated Items**
The same item can be listed more than once as long as they have different item IDs.

**“Get It Now” scenario**
When the item is already sold while the user is on the Item description screen, clicking Get it now button should display and error to the user stating that the item has already been sold. Similarly, the user will not be able to bid on an item that has already been sold.

**Current User**
The system will have the reference to the current user that has logged in and the username of the current user will be used in various tasks such as List/Buy/Bid/Rate.

**Log out**
When the user closes the application/system, he/she will be logged out and asked to log in again if the application is re-launched.