

## Table of Contents

<b>Data Types .....</b>	<b>2</b>
<i>User.....</i>	<i>2</i>
<i>Admin User.....</i>	<i>2</i>
<i>Item .....</i>	<i>2</i>
<i>Auction Length.....</i>	<i>3</i>
<i>Conditions .....</i>	<i>3</i>
<i>Categories .....</i>	<i>3</i>
<i>Rating .....</i>	<i>3</i>
<i>Bid.....</i>	<i>3</i>
<b>Task Decomposition.....</b>	<b>3</b>
<i>Login:.....</i>	<i>3</i>
<i>Register:.....</i>	<i>4</i>
<i>Item Search: .....</i>	<i>4</i>
<i>List My Item: .....</i>	<i>5</i>
<i>Auction Results:.....</i>	<i>6</i>
<i>Category Report: .....</i>	<i>6</i>
<i>User Report: .....</i>	<i>7</i>
<i>View Item Info:.....</i>	<i>7</i>
<i>Edit Item Description: .....</i>	<i>8</i>
<i>Get It Now: .....</i>	<i>8</i>
<i>Bid:.....</i>	<i>9</i>
<i>View Ratings:.....</i>	<i>9</i>
<i>Rate This Item: .....</i>	<i>10</i>
<i>Delete Rating:.....</i>	<i>10</i>
<b>Constraints .....</b>	<b>11</b>
<i>User.....</i>	<i>11</i>
<i>Item: .....</i>	<i>11</i>
<i>Bid:.....</i>	<i>11</i>
<b>Assumptions .....</b>	<b>11</b>
<i>Rating:.....</i>	<i>11</i>
<i>Bid:.....</i>	<i>11</i>

<b>Test Cases.....</b>	<b>11</b>
<i>Login .....</i>	<i>11</i>
<i>Register.....</i>	<i>11</i>
<i>Item Search.....</i>	<i>12</i>

## Data Types

### User

Attribute	Data type	Nullable
Username	String (varchar(20))	Not Null
Password	String	Not Null
FirstName	String	Not Null
LastName	String	Not Null
Listed	Integer	Not Null (Min 0)
Sold	Integer	Not Null (Min 0)
Purchased	Integer	Not Null (Min 0)
Rated	Integer	Not Null (Min 0)

### Admin User

Attribute	Data type	Nullable
Position	String (varchar(20))	Not Null

### Item

Attribute	Data type	Nullable
ItemID	Integer ( <i>big int</i> )	Not Null
Name	String (varchar(20))	Not Null
Description	String ( <i>text</i> )	Not Null
Returnable	Boolean	Not Null
StartingBid	Double ( <i>DECIMAL(15,2)</i> )	Not Null
MinimumSalePrice	Double ( <i>DECIMAL(15,2)</i> )	Not Null
GetItNowPrice	Double ( <i>DECIMAL(15,2)</i> )	Null
SalePrice	Double ( <i>DECIMAL(15,2)</i> )	Null
HighBidder	String (varchar(20))	Null
Winner	String (varchar(20))	Null
AuctionEnds	String ( <i>ISO 8601</i> )	Not Null
AuctionEnded	String ( <i>ISO 8601</i> )	Null
AverageRating	Double	Null
CurrentBid	Double ( <i>DECIMAL(15,2)</i> )	Null

## Auction Length

Attribute	Data type	Nullable
Length	Integer ( <i>tiny int</i> )	Not Null

## Conditions

Attribute	Data type	Nullable
Condition	String ( <i>varchar(19)</i> )	Not Null
Value	Integer ( <i>tiny int</i> )	Not Null

## Categories

Attribute	Data type	Nullable
Category	String ( <i>varchar(15)</i> )	Not Null
MinPrice	Double ( <i>DECIMAL(15,2)</i> )	Not Null (Min 0)
MaxPrice	Double ( <i>DECIMAL(15,2)</i> )	Not Null (Min 0)
AvgPrice	Double ( <i>DECIMAL(15,2)</i> )	Not Null (Min 0)
TotalItems	Integer	Not Null (Min 0)

## Rating

Attribute	Data type	Nullable
NumberOfStars	Integer ( <i>tiny int</i> )	Not Null
RateTime	String ( <i>ISO 8601</i> )	Not Null
Comment	String	NULL

## Bid

Attribute	Data type	Nullable
Price	Double ( <i>DECIMAL(15,2)</i> )	Not Null
BidTime	String ( <i>ISO 8601</i> )	Not Null

## Task Decomposition

Login:

**Lock Types:** 2 Read-only lookups on **User**, and **AdminUser**, Tables

**Number of Locks:** 2

**Enabling Conditions:** None

**Frequency:** 10,000 logins/day

**Consistency (ACID):** not critical

**Subtasks:** Mother Task is not needed



**Abstract Code:**

- User enters *username*, *password* input fields
- If data validation is true then
  - When **Login** is clicked:
    - If username and password combination are incorrect
      - Go back to **Login** form, with error message.
    - Else
      - Create session variable with login information
      - Go to **Main Menu** (user type provided to main menu)
- Else *username* and *password* are incorrect, display **Login** with error message.

#### Register:

**Lock Types:** Write lock on **User** Table

**Number of Locks:** Single

**Enabling Conditions:** **Username** is available

**Frequency:** 35 registrations/day

**Consistency (ACID):** Critical. Will rely on database to ensure PK is not duplicated.

**Subtasks:** Mother Task is not needed



#### **Abstract Code:**

- User enters *FirstName*, *LastName*, *username*, *password*, and *confirm password* input fields
- When **Register**:
  - If all fields are filled in:
    - If username already exists
      - Go back to **Registration** form, with error message.
    - If password do not match
      - Go back to **Registration** form, with error message.
    - Else
      - Start transaction
      - Insert into User table: "**Username**", "**Password**", "**FirstName**", "**LastName**"
      - Commit transaction
      - Give "Successful Registration" message and redirect user to "**Login**" view
  - Else
    - Go back to **Registration** form, with error message
- When **Cancel** is clicked:
  - Return to **Login** page

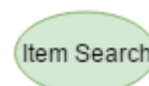
#### Item Search:

**Lock Types:** 2 read-only lookups from **Item** table and **Bid** table

**Number of Locks:** 2

**Enabling Conditions:** Successful **Login**

**Frequency:** 50,000 searches/day



**Consistency (ACID):** Not critical

**Subtasks:** Mother task is not needed. No decomposition needed.

**Abstract Code:**

- When search is clicked
  - If all fields are blank
    - Display error message on **Item Search** view
  - Else-If any field contains invalid search parameters
    - Display error message on **Item Search** view
  - Else
    - Sanitize input
    - Lookup items that matches search criteria
      - Use Keyword to find any matches in Item Name of **Items**
      - Use Keyword to find any matches in Description of **Items**
      - Use Category to find any matches in **Items**
      - Use Condition to find any matches that are at least the same or better in **Items**
      - Use Min and Max Price to find any **Items** that have current high bid or starting bid in that range.
      - Then separate out matches that were in all fields
    - Return results and display **View Search Results** view
      - Use **Auction End Time** to separate matches that are still active.
      - Show **ID, Item Name, Current Bid, High Bidder, Get It Now Price, Auction Ends**
      - For all active matches in **Search Results**
        - Lookup **ItemID**
        - Lookup **ItemName**
        - Lookup **CurrentBid**
        - Lookup **HighBidder**
        - Lookup **GetItNowPrice**
        - Lookup **AuctionEnd** date and time
      - If **ItemName** is pressed
        - Then go to **ViewItemInfo**
    - When **Back to Search** is clicked
      - Jump to **Item Search** page
  - When cancel is clicked
    - Return to **Main Menu** page

[List My Item:](#)

**Lock Types:** Insert new Item into **Item** table

**Number of Locks:** Single

**Enabling Conditions:** Successful **Login**

**Frequency:** 750 listings/day

A green oval button with the text "List My Item" inside.

**Consistency (ACID):** Not Critical

**Subtasks:** Mother Task is not needed

**Abstract Code:**

- Show **New Item For Auction** page
- Wait until “**List my Item**” is clicked
  - Then check that all fields except for **GetItNowPrice** have valid data
    - If at least 1 field has invalid data
      - Then display error message for that field
    - If all fields have valid data
      - Then **ListItem**
        - Assign **ItemID**

[Auction Results:](#)

**Lock Types:** 2 read-only lookup from **Item** table and **Bid** table

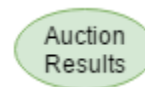
**Number of Locks:** 2

**Enabling Conditions:** Successful **Login**

**Frequency:** 1,500 views/day

**Consistency (ACID):** Not Critical

**Subtasks:** Mother Task is not needed



**Abstract Code:**

- Lookup **AuctionEnd** from Items
  - If **AuctionEnd** equals true
    - Then list **Item**
      - Do this for all Items
    - Sort all **Items** in list by most recent **AuctionEndTime**
      - Display "Item ID", "Item Name", "SalePrice", "Winner", "Auction Ended"
        - For all items in list
          - Lookup **ItemID**
          - Lookup **ItemName**
          - Lookup **CurrentBid**
          - Lookup **HighBidder**
          - Lookup **AuctionEndTime**
- When Done is clicked
  - Return to “**Main Menu**” view

[Category Report:](#)

**Lock Types:** 2 read-only lookups from **Item** table and **Category** table

**Number of Locks:** 2

**Enabling Conditions:** Successful **Login** and User is **AdminUser**

**Frequency:** 50 views/day

**Consistency (ACID):** Need to make sure



**Subtasks:** Mother Task is not needed

**Abstract Code:**

- Display **Category, Total Items, Min Price, Max Price, Average Price**
  - For each **Category** from **Items**
    - Count all items from **Category** for **Total Items**
    - Find the lowest **GetItNowPrice** for **Min Price**
    - Find the highest **GetItNowPrice** for **Max Price**
    - Average all **GetItNowPrice** for **Average Price**
- When Done is clicked
  - Return back to **Main Menu**

[User Report:](#)

**Lock Types:** 3 read-only lookups from **User, Bids, Item** tables

**Number of Locks:** 3

**Enabling Conditions:** Successful **Login** and user is an **AdminUser**

**Frequency:** 50 views/day

**Consistency (ACID):** Not Critical.

**Subtasks:** Mother Task is not needed



**Abstract Code:**

- Display **Username, Listed, Sold, Purchased, Rated**
  - For each **UserName**
    - Lookup **Listed**
    - Lookup **Sold**
    - Lookup **Purchased**
    - Lookup **Rated**
- When Done clicked
  - Return to **Main Menu**

[View Item Info:](#)

**Lock Types:** 2 read-only lookups from **Item**, and **Bid** tables

**Number of Locks:** 2

**Enabling Conditions:** Successful **Login**

**Frequency:** 20,000 views/day

**Consistency (ACID):** Not critical

**Subtasks:** Mother Task is not needed



**Abstract Code:**

- **Upon:**
  - Clicking **ItemName**
    - Query item details by item key
    - Display result on "**GTBay Item for Sale**" view

### Edit Item Description:

**Lock Types:** Update **Item** table

**Number of Locks:** Single

**Enabling Conditions:** Current user is same as Listing User

**Frequency:** 500 edits/day

**Consistency (ACID):** Not critical

**Subtasks:** Mother Task is not needed



#### **Abstract Code:**

- User edits text in “**Edit Description**” popup window.
- **Upon:**
  - Clicking “**Save Description**” button
    - Sanitize data
    - Update description in database
    - Close popup window

### Get It Now:

**Lock Types:** Read-only from **Item** table and Write into **Bid** table.

**Number of Locks:** 2

**Enabling Conditions:** Successful **Login** Item is not sold

**Frequency:** 1,000 times/day

**Consistency (ACID):** Critical

**Subtasks:** All tasks must be done, and in the proper order.

Mother task is required to coordinate subtasks. Order is important.



#### **Abstract Code:**

- **Upon:**
  - Clicking “**Get It Now**” button
    - Start transaction
    - If item is still for sale
      - Update bid table with **Username** = logged in user, **Price** = **GetItNowPrice**, and **BidTime** = NOW()
      - Commit transaction
      - Update “**GTBay Item for Sale**” view with new information
    - Else
      - Roll back transaction
      - Return failure message: “Auction has ended”
      - Update “**GTBay Item for Sale**” view with new information



## Bid:

**Lock Types:** Read-only from **Item** table and Write into **Bid** table.

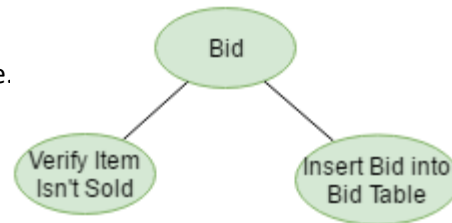
**Number of Locks:** 2

**Enabling Conditions:** Successful **Login** item is not sold

**Frequency:** 8,500 bids/day

**Consistency (ACID):** Critical; Order is important

**Subtasks:** All tasks must be done, and in the proper order. Mother task is required to coordinate subtasks. Order is important.



### **Abstract Code:**

- User enters bid value
- **Upon:**
  - Clicking “**Bid On This Item**” button
    - If input is valid
      - Start transaction
      - If item is still for sale
        - Update bid table with **Username** = logged in user, **Price** = input price, and **BidTime** = NOW()
        - Commit transaction
        - Update “**GTBay Item for Sale**” view with new information
      - Else
        - Roll back transaction
        - Return failure message: “Auction has ended”
        - Update “**GTBay Item for Sale**” view with new information
    - Else
      - Return failure message: “**Invalid Bid**”

## View Ratings:

**Lock Types:** Read-only lookup from **Rating** table

**Number of Locks:** Single

**Enabling Conditions:** Successful **Login**

**Frequency:** 20,000 views/day

**Consistency (ACID):** Not critical

**Subtasks:** Mother Task is not needed



### **Abstract Code:**

- **Upon:**
  - Clicking “**View Ratings**”
    - Query database for derived attribute “**Average Rating**”.
    - Query database for all ratings of the “**Item**”
    - Navigate to “**Item Ratings**” view and display results

### [Rate This Item:](#)

**Lock Types:** Write into **Rating** tables

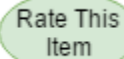
**Number of Locks:** Single

**Enabling Conditions:** Successful **Login** and User has not already **rated** the item

**Frequency:** 750 rates/day

**Consistency (ACID):** Not critical

**Subtasks:** Mother Task is not needed

A green oval button with the text "Rate This Item" in black.

#### **Abstract Code:**

- **Upon:**
  - Clicking "**Rate This Item**"
    - If Rating has not been clicked
      - Do nothing
    - Else
      - Sanitize input
      - Start transaction
      - Upsert **Rating** table with "**ITEM**", "**Username**", "**NumberOfStars**", "**Comment**" and "**RateTime**"
      - Commit transaction
      - Update "**Item Ratings**" view with new rating data

### [Delete Rating:](#)

**Lock Types:** Write into **Rating** table

**Number of Locks:** Single

**Enabling Conditions:** Successful **Login** and **Rating** belongs to said User

**Frequency:** 20 times/day

**Consistency (ACID):** Not critical

**Subtasks:** Mother Task is not needed

A green oval button with the text "Delete Rating" in black.

#### **Abstract Code:**

- **Upon:**
  - Clicking "**Delete My Rating**"
    - Start transaction
    - Remove **Comment**, **RateTime**, and **NumberOfStars**
    - Commit transaction
    - Update "**Item Ratings**" view

## Constraints

### User:

- New GTBay Users must Register first.
- Existing GTBay Users cannot Register again.
- GTBay Users can only edit the description of items they own
- GTBay Users can only delete ratings on items they have rated
- Auction Winners are announced 45 seconds after the Auction has finished
- GTBay Users cannot remove an Item once it is up for sale
- Password must contain at least: 1 special character, 1 capital letter, 1 lower-case letter, 10 or more characters.

### Item:

- Minimum Sale Price must be greater than 5 US dollars.

### Bid:

- Bids for an item must exceed its previous bid by 1 cent.
- Bids cannot exceed the value of the “buy it now” price if it exists.

## Assumptions

### Rating:

- “**Item Ratings**” view can be rendered with no item ratings. The “**Average Rating**” will be 0 and ratings list will be empty.

### Bid:

- Auction winners are derived based on the bid history. A bid cannot exceed the value of the “**Get It Now Price**”. The winning bid is either the one that paid the “**Get It Now Price**” or the highest bidder that bid at least the “**Minimum Sale Price**” before the auction ended. If the “**Get it Now Price**” is paid, it’s “**Bid Time**” will indicate the “**Auction End**” time.

## Test Cases

TC#	Area	Test Case Description
<b>L000</b>	Login	
L001		Login page entry (username, password) credentials
L002		Valid credentials (username, password) accepted
L003		reject invalid (username, password) when user does not exist
L004		reject invalid (username, password) when user exists, password invalid
<b>R000</b>	Register	
R001		Username already exists
R002		Username validation fails
R003		FirstName validation fails
R004		LastName validation fails

R005		Password validation fails
R006		Passwords do not match
<b>IS000</b>	<a href="#">Item Search</a>	