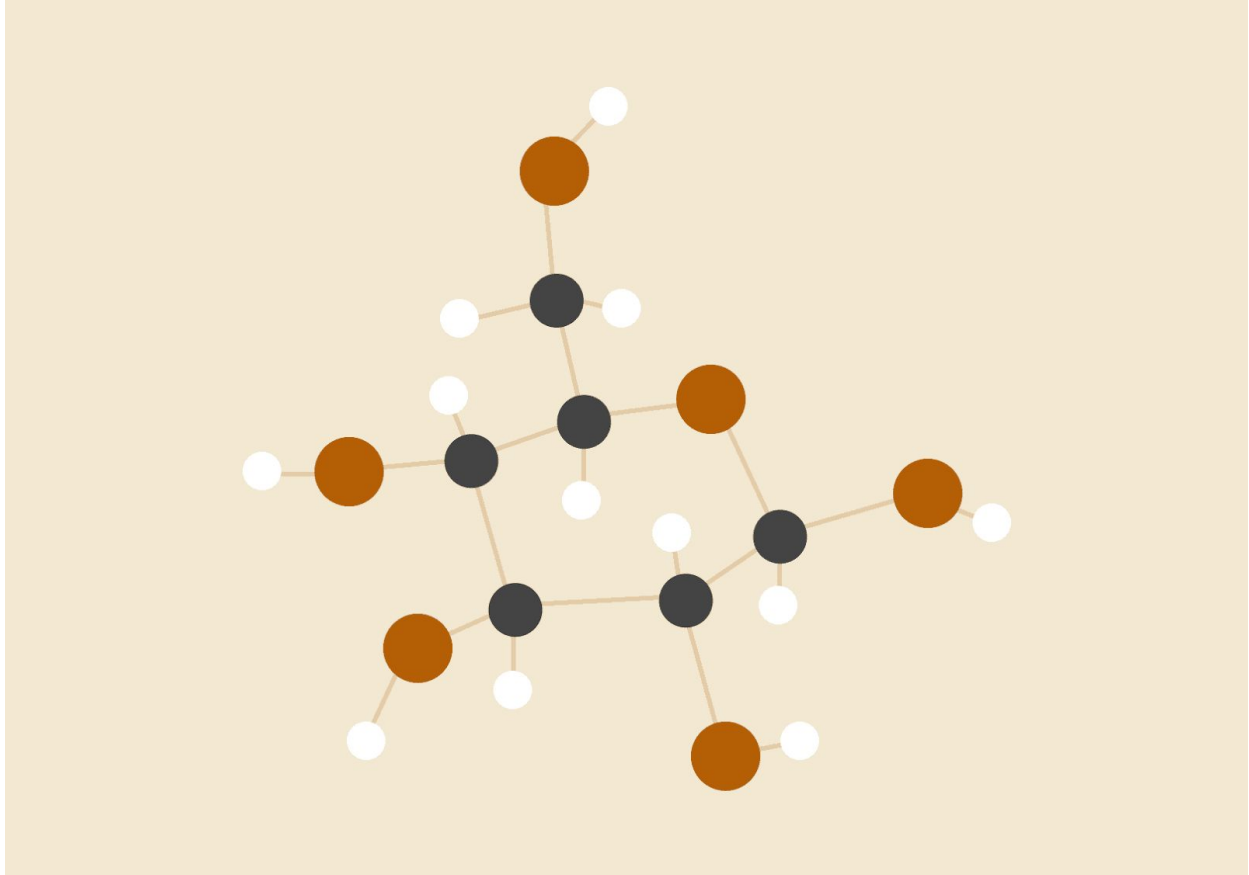


Phase 2: Abstract Code + SQL

CS6400: Team 105



Jung, Sungjin; sjung93@gatech.edu
Kazkayasi, Mehmet O; mkazkayasi3@gatech.edu
Katnis, Nichole Marlene; nkatnis3@gatech.edu

March 4, 2018
Online Master of Science in Computer Science
Georgia Tech

TABLE OF CONTENTS

Login	2
Register	3
New Item For Auction	4
Item Search	5
Item for Sale	6
Item Ratings	7
Auction Results	8
Category Report	9
User Report	10

Login

Abstract Code

- User enters *username* ('\$Username'), *password* ('\$Password') input fields.
- When **Login** button is clicked:
 - If User record is found but `User.password != '$Password'`:
 - Go back to **Login** form, with error message.
 - If User record did not found: Go back to **Login** form, with error message.
 - Else:
 - Store login information as session variable '\$Username'
 - Go to **Main Menu**
- Upon click **Register** button - Jump to **Register** task

Register

Abstract Code

- User enters *First Name* ('\$FName'), *Last Name* ('\$LName'), *Username* ('\$Username'), *Password* ('\$Password') and *Confirm Password* ('\$CPassword') input fields.
- When **Register** button is clicked:
 - If any of the fields is not filled in: Display **Register** form, with error message.
 - If '\$Password' != '\$CPassword': Display **Register** form, with error message.
 - If '\$Username' is used by another User: Display **Register** form, with error message.
 - Else, store new user information in User table: Go to **Log In** form
- When **Cancel** button is clicked: Display **Login** form.

New Item For Auction

Abstract Code

- Get categories from `Category.Category_Name` values in dropdown list.
- User:
 - Enters *Item Name* (`'$ItemName'`), *Description* (`'$Description'`), *Start Auction Bidding* (`'$AuctionStartPrice'`), *Minimum Sale Price* (`'$MinSalePrice'`), *Minimum Sale Price* (`'$AuctionStartPrice'`), and optionally *Get It Now Price* (`'$GetItNowPrice'`).
 - Selects *Category* (`'$Category'`), *Condition* (`'$Condition'`), *Auction Length* (`'$AuctionLength'`) input fields from dropdown lists,
 - Chooses to click or not on *Returns Accepted* (`'$ReturnsAccepted'`) field.
- When **List My Item** button is clicked:
 - If any of the fields except *Get It Now Price* field is not filled in: Display **New Item for Auction** form, with error message.
 - If `'$GetItNowPrice' < '$MinSalePrice'`: Display **New Item for Auction** form, with error message.
 - Assign `Item.Item ID`, Assign 1 to `Item.Active`.
 - Calculate `Item.AuctionEndDate` using Current date and time along with `'$AuctionLength'`
 - Get the User using `'$Username'` and increment `User.Listed` by 1
 - Get Category that Item is in using `'$Category'` and :
 - Increment `Category.TotalItems` by 1.
 - If the `Item.GetItNowPrice < Category.MinPrice`: Assign `Category.MinPrice = Item.GetItNowPrice`
 - If `Item.GetItNowPrice > Category.MaxPrice`: Assign `Category.MaxPrice = Category.GetItNowPrice`
 - Modify `Category.AveragePrice` using `Category.AveragePrice`, `Category.TotalItems` and `Item.GetItNowPrice`
 - Store new item information in `Item` table
 - Display **Main Menu** form.
- When **Cancel** button is clicked: Display **Main Menu** form.

Item Search

Abstract Code

- Get **Category**.Category_Name values along with “**Any**” option in dropdown list.
- User:
 - Enters *Keyword*('\$Keyword'),
 - Optionally user selects: *Minimum Price*('\$MinPrice'), *Maximum Price*('\$MaxPrice'), *Category* ('\$Category') and *Condition At Least* ('\$ConditionAtLeast')
- When **Search** button is clicked:
 - Find Items with **Item**.Active = 1 in the selected category which has '\$Keyword' in '\$ItemName' or '\$Description'
 - Optional search criteria used to further filter results
 - that have a current high Bid of at least '\$MinPrice' and at most '\$MaxPrice'
 - better '\$Condition' than '\$ConditionAtLeast'
 - For each Item:
 - Get **Item**.ItemId, **Item**.ItemName, **Item**.GetItNowPrice, **Item**.AuctionEndTime
 - Get **Bid** with max **Bid**.Time associated with Item and as Current Bid,
 - Get Bid Owner associated with Current bid as High Bidder
 - Display form **Search Results**
 - When any **Item Name** button is clicked: Display **Item for Sale** form.
- When **Back to Search** button is clicked: Display **Item Search** form.

Item for Sale

Abstract Code

- Get **Item**.ItemName, **Item**.ItemID, **Item**.Description, **Item**.AuctionEndDate, **Category**.Category_name that **Item** is related, **Item**.Condition, **Item**.GetItNowPrice (if not NULL), get top 4 bids with bids sorted by descending **Bid**.Time that **Item** is related as well as usernames associated with the **Bid**
- Show Edit Description button if '\$UserID' belongs to the seller.
- Calculate ('\$MinBid'), by adding one dollar on the highest bid.
- Show **Item for Sale** form
- Upon clicked **Get It Now** button:
 - Assign **Item**.Active = 0
 - Assign **Item**.SalePrice = **Item**.GetItNowPrice
 - For user that listed the item increment **User**.Sold by 1
 - For user buying the product increment **User**.Purchased by 1
 - Modify **Item**.AuctionEndDate as Current Time.
 - Assign **Item**.SalePrice = **Item**.GetItNowPrice
 - Assign **Item**.BuyerName = '\$UserID'
- Upon **View Ratings** button is clicked: Display form **Item Ratings**
- User can choose to bid on item. So, User enters Your Bid ('\$YourBid'). Upon clicked **Bid On This Item** button:
 - If '\$YourBid' is not populated Display Item For Sale form with error message.
 - If '\$YourBid' is not larger than one dollar added to highest bid Item has: Display Item For Sale form with error message.
 - If '\$YourBid' is greater than or equal to the **Item**.GetItNowPrice Display **Item For Sale** form with error message.
 - Else, create a new bid for the item. Assign **Bid**.BidAmount = '\$YourBid' and **Bid**.TimeofBid = Current Time.
 - Display **Item For Sale** form.
- Upon **Edit Description** button is clicked: Enable textbox containing **Item**.Description for edit.
- When **Cancel** button is clicked: Display **Search Results** form.

Item Ratings

Abstract Code

- Find and display **Item**.ItemName, **Item**.ItemID,
- Get ratings that are associated with the Item and calculate the average rating
- Get all Rating information related to the Item: **Rating**.RatingTimeDate, **Rating**.Comment, and **Rating**.NumberOfStars sorted by descending **Rating**.RatingDateime
- Check '\$UserID' has a Rating and if it has: Display **Delete My Rating** button on '\$UserID's rating
- Upon **Delete My Rating** button is clicked: Remove the Rating belongs to '\$UserID' on
- User fills Comments ('\$Comment') and selects My Rating ('\$Rating').
- Upon **Rate This Item** button is clicked:
 - Store **Rating**.RatingTimeDate as Current time, **Rating**.Comment, and **Rating**.NumberOfStars in Rating table.
 - Display **Item Ratings** form.
- Upon **Cancel** button is clicked: Display **Item for Sale** form.

Auction Results

Abstract Code

- Fetch Items that Item.AuctionEndDate has passed.
- Set **Item**.Active = 0 for returned Items
- If **Item**.SalePrice is NULL and Item has bids and highest bid is greater than or equal to **Item**.MinimumSalePrice:
 - Set **Item**.SalePrice = Current Bid.
 - Set **Item**.Buyer = Current Bid's Owner
- Display **Item**.ItemID, **Item**.ItemName, **Item**.SalePrice ('-' in case NULL), and **Item**.Buyer ('-' in case NULL)
- Display **Auction Results** sorted by descending AuctionEndDate
- Upon **Done** button is clicked: Display **Item for Sale** form.

Category Report

Abstract Code

- Fetch Categories and sort by `Category.CategoryName`
- Display `Category.Listed`, `Category.MinPrice`, `Category.TotalItems`, and `Category.MaxPrice`, `Category.AveragePrice`.
- Upon clicked Done button: Display **Main Menu** form.

User Report

Abstract Code

- Fetch Users and sort by `User.Listed`.
- Display `User.Listed`, `User.Purchased`, `User.Sold`, and `User.Rated`.
- Upon clicked Done button: Display **Main Menu** form.