

The final database design, highlighting what has changed since the last milestone.

Assumptions

- Users can only review products they have purchased
- Inventory updates only at the time of submission to simplify transactions
- Each user must register with a unique email address, which will serve as the primary method of authentication
- All purchases and transactions on the site are done using virtual currency
- A user's cart is persistent across sessions → items that remain in the cart even if the user logs out and logs back in
- Sellers are responsible for managing their own inventory, including adding products, updating quantities, and removing items as needed
- All products must belong to predefined categories, and different sellers can list the same product with different prices and available quantities.
- A user can only leave a single review per product and a seller can edit or delete it at any time.
- Once an order is placed, the buyer cannot modify the order, but the seller must update the fulfillment status when the order is shipped.
- Seller profiles must be public and display basic seller information, including product listings, reviews, and ratings.
- Messages are unique to the customer, seller, and product
- Users can only review sellers they have purchased from and products they have purchased

Preliminary Design Flow

- Log-in: A new user can register for a new account. An existing user can log in using email and password
 - Log-in → Products
 - Log-in → Account
- Account: Users can view their account information, such as their ID, full name, address, and password, as well as update information except for ID
- Public view for a user: Users can see the account number and name of another user. If the user also acts as a seller, other users can see their email address, address, and a section with all reviews for the seller
- Purchases: Users can browse their history of purchases, sorted in reverse chronological order by default. For each purchase in this list, users see the total amount, number of items, and fulfillment status, and a link to the detailed order page
- Products: Users can see products in the database as well as product images, sellers, and prices, filtered by product category
 - Products → Individual Product

- Individual Product: Users can see all details for the product as well as a list of sellers and their current quantities in stock
 - Individual Product → Cart
 - Individual Product → Products
- Cart: One product is represented by one line in the cart from one seller with a specific quantity. This page includes the line item quantity, the line item unit price, and the total price
 - Cart → Order
- Order: One product is represented by one line in the cart from one seller with a specific quantity. This page includes the line item quantity, the line item unit price, the final price, and if and when the line item is fulfilled by the seller
- Inventory: This page lists all products for sale by this user
- Order Fulfillment: A seller can browse the history of orders fulfilled or to be fulfilled, sorted in reverse chronological order by default. Each order in this list shows a summary, including buyer information such as address, date ordered, total number of items, and fulfillment status. provide a mechanism for marking a line item as fulfilled
- Seller: Provides an interface for adding a specific quantity of the product from a seller to a user's cart. This page also includes a section showing all reviews for this product
- Product Review: Upon purchase of a product, the customer would be able to leave a review on the product - they access this from the product page (see Products). This review would include a rating (numerical integer from 1-5) and optional text-based message review. The review is uniquely identified by the customer and product, and with the date attribute, is able to be sorted by date.
- Seller Review: Upon purchase of a product from a seller, the customer can submit feedback on the seller. This review has number (indicating rating out of 5) and message (ie: text) attributes. It is uniquely identified by the customer and seller, and with the date attribute, is able to be sorted by date.
- Messaging: Upon order of a product, users are able to message the seller. The messages have a unique ID, and can be sorted into the conversation via the attributes customer, seller, and product.

Final Database Design

Users

- **id** INT PRIMARY KEY GENERATED BY DEFAULT AS IDENTITY
- **email** VARCHAR UNIQUE NOT NULL
- **password** VARCHAR(255) NOT NULL
- **firstname** VARCHAR(255) NOT NULL
- **lastname** VARCHAR(255) NOT NULL
- **balance** DECIMAL(12,2) DEFAULT 0.00

Categories

- **category_ID** INT PRIMARY KEY GENERATED BY DEFAULT AS IDENTITY

- **category_name** VARCHAR(255) UNIQUE NOT NULL

Products

- **id** INT PRIMARY KEY GENERATED BY DEFAULT AS IDENTITY
- **name** VARCHAR(255) NOT NULL
- **description** TEXT
- **category** VARCHAR(255) -- (Stores category name; must exist in Categories)
- **price** DECIMAL(12,2) NOT NULL
- **image** TEXT
- **creator_ID** INT REFERENCES Users(id)
- **available** BOOLEAN DEFAULT TRUE

Purchases

- **id** INT PRIMARY KEY GENERATED BY DEFAULT AS IDENTITY
- **uid** INT NOT NULL REFERENCES Users(id)
- **pid** INT NOT NULL REFERENCES Products(id)
- **order_id** UUID -- Used to group multiple purchased items in one order transaction
- **time_purchased** TIMESTAMP WITHOUT TIME ZONE DEFAULT CURRENT_TIMESTAMP

Wishes

- **id** INT PRIMARY KEY GENERATED BY DEFAULT AS IDENTITY
- **uid** INT NOT NULL REFERENCES Users(id)
- **pid** INT NOT NULL REFERENCES Products(id)
- **time_added** TIMESTAMP WITHOUT TIME ZONE DEFAULT CURRENT_TIMESTAMP

UserReviewsProduct

- **id** INT PRIMARY KEY GENERATED BY DEFAULT AS IDENTITY
- **customer_id** INT NOT NULL REFERENCES Users(id)
- **product_id** INT NOT NULL REFERENCES Products(id)
- **rating_num** INT CHECK (rating_num >= 1 AND rating_num <= 5)
- **rating_message** TEXT
- **review_date** TIMESTAMP WITHOUT TIME ZONE DEFAULT CURRENT_TIMESTAMP

UserReviewsSeller

- **id** INT PRIMARY KEY GENERATED BY DEFAULT AS IDENTITY
- **customer_id** INT NOT NULL REFERENCES Users(id)
- **seller_id** INT NOT NULL REFERENCES Users(id)
- **rating_num** INT CHECK (rating_num >= 1 AND rating_num <= 5)

- **rating_message** TEXT
- **review_date** TIMESTAMP WITHOUT TIME ZONE DEFAULT CURRENT_TIMESTAMP

Message

- **msg_num** INT PRIMARY KEY GENERATED BY DEFAULT AS IDENTITY
- **customer_ID** INT REFERENCES Users(id)
- **message** TEXT NOT NULL
- **message_date** TIMESTAMP WITHOUT TIME ZONE DEFAULT CURRENT_TIMESTAMP
- **product** INT REFERENCES Products(id)

Seller

- **acct_ID** INT NOT NULL REFERENCES Users(id)
- **product_ID** INT NOT NULL REFERENCES Products(id)
- **quantity** INT DEFAULT 0
- PRIMARY KEY (acct_ID, product_ID)

Cart

- **cart_ID** INT PRIMARY KEY GENERATED BY DEFAULT AS IDENTITY
- **acct_ID** INT NOT NULL REFERENCES Users(id)
- **total_cost** DECIMAL(12, 2) DEFAULT 0.00

CartItem

- **cart_item_ID** INT PRIMARY KEY GENERATED BY DEFAULT AS IDENTITY
- **cart_ID** INT NOT NULL REFERENCES Cart(cart_ID)
- **product_ID** INT NOT NULL REFERENCES Products(id)
- **quantity** INT DEFAULT 1
- **item_price** DECIMAL(12, 2) NOT NULL

This final design covers:

- **Users** and their balances.
- **Products**, along with their categories and availability.
- **Wishes** for users' wishlists.
- **Purchases** with order grouping via **order_id**.
- **Cart** and **CartItem** for the shopping cart functionality.
- **UserReviewsProduct** and **UserReviewsSeller** for feedback and ratings.
- **Message** for messaging functionality (if utilized by the social features).
- **Seller** for tracking which user sells which product and in what quantity.

Changes Made Since the Last Milestone:

- Improved email validation for registration to reject ill-formatted emails more consistently.
- Enhanced error messages for duplicate registration attempts and invalid login credentials.
- Added rank/filter
- Added a check for insufficient funds during checkout, with a clear flash message displayed if the total cart cost exceeds the user's balance.
- Finalized cart contents (including item prices) during checkout to handle inventory and price updates accurately at the time of order submission.
- Cart contents now persist across user sessions and are restored when the user logs back in.
- Added validation to ensure that a user can only submit reviews for purchased products.
- Enhanced review display on the product details page with average ratings and the number of reviews shown prominently.
- Verified that unavailable products (out of stock) cannot be added to the cart or checked out, by not showing unavailable products

The final list of features you have implemented or attempted to implement. For each feature, indicate its status: fully functional, buggy, partially implemented, or not implemented at all.

Fully Implemented:

- **Browsing Products:**
 - Displaying a list of products with basic info (name, price, image) – **Fully Functional**
 - Viewing product details (description, category, reviews, sellers' stock) – **Fully Functional**
- **Searching and Filtering Products:**
 - Search by keywords in product name/description – **Fully Functional**
 - Filter by category – **Fully Functional**
 - Filter by price range (min/max) – **Fully Functional**
 - Filter by minimum rating – **Fully Functional**
 - Sort products by name, price, and rating – **Fully Functional**
- **Product Reviews:**
 - Displaying existing reviews on product detail page – **Fully Functional**
 - Adding or updating a product review if purchased – **Fully Functional**
 - Validation of rating range (1–5) – **Fully Functional**
- **Sellers and Product Availability:**
 - Showing sellers and their available quantities for each product – **Fully Functional**
 - Dynamically updated stock display based on **Seller** table entries – **Fully Functional**

- **Add to Cart from Product Page:**
 - Authenticated users can add a product to their cart directly from the product detail page – **Fully Functional**
 - Basic error checks if no seller or invalid quantity chosen in older logic (now simplified with direct `Cart.add()` call) – **Fully Functional** under the current simplified approach
- **Create and Edit Products (For Product Creators):**
 - Authenticated users can create new products, providing name, description, category, price, and image – **Fully Functional**
 - Creators can edit product details later – **Fully Functional**
 - Category validation for new products – **Fully Functional**
- **Category Management:**

Displaying available categories for browsing and product creation – **Fully Functional**

 - Using a predefined `Categories` table (assuming populated) – **Fully Functional**

Account / Purchases

- New users can register with a unique email, name, and password.
 - Existing users can log in using email and password.
 - Each account has:
 - A unique, system-assigned ID (cannot be updated).
 - Editable fields: email, name, address, and balance.
 - Balance management: starts at \$0, can be topped up or withdrawn fully.
 - Users can browse purchase history:
 - Sorted in reverse chronological order.
 - Each purchase includes a summary (total amount, number of items, fulfillment status).
 - Links to a detailed order page.
 - Public profile view for users:
 - Displays account number, name, and summary information.
 - If the user is also a seller: shows email, address, and reviews from buyers.
-

Products

- Each product belongs to a predefined category.
- Product attributes:
 - Short name, longer description, image, price.
- Users can browse and search/filter products:
 - Filters: by category, keywords in name/description.
 - Sorting: by price.
- Detailed product page:

- Displays product details, reviews, and sellers with available quantities.
 - Allows adding items to cart from a specific seller.
 - Authenticated users can:
 - Create new products with name, description, category, price, and image.
 - Edit product details later.
-

Cart / Order

- Each user has a persistent cart:
 - Lists all line items.
 - Displays total price.
 - Provides options to remove items and submit as an order.
 - Order submission:
 - Updates inventory and balances during submission.
 - Clears the cart after submission.
 - Decrements buyer's balance, increments sellers' balances, and decrements inventory.
 - Detailed order page:
 - Shows final prices, line items, and fulfillment status.
 - Orders cannot be edited after submission.
-

Inventory / Order Fulfillment

- Sellers have an inventory page:
 - Lists products for sale.
 - Allows adding/removing products and updating available quantities.
 - Sellers can browse/search order history:
 - Sorted in reverse chronological order.
 - Displays buyer information, order date, total amount, total items, and fulfillment status.
 - Mechanism for marking line items as fulfilled.
-

Feedback / Messaging

- Users can submit ratings/reviews:
 - For products: if purchased.
 - For sellers: if ordered from them.
 - Reviews can be edited or removed.
- User reviews are listed and sorted in reverse chronological order.

- Pages display:
 - Average ratings and number of reviews for products and sellers.
 - Lists of reviews, sortable by rating or date.

Not Implemented:

- SQL Injection
- Search Products

The data is real-like data and the dataset size differs depending on the table. No values are hard-coded. SQL injection attacks are not taken care of. No additional features are implemented