

Product Management System

eStoreDB

By: James Donnelly

SFSU ID: 917703805

Github: linkyap

Checkpoint #	Date Submitted
Checkpoint II	06/29/2023
Checkpoint I	06/19/2023

Tables of Contents

Project Description.....2

Functional Requirements.....4

Non-Functional Requirements..... 7

Entity Relationship Diagram.....9

 (NEXT PAGE)..... 9

Entity Relationships..... 14

Strong Entities..... 14

Weak Entities:..... 17

Project Description

Motivation:

The reason behind creating this database system is to address the challenges faced by online businesses in managing their data both efficiently and effectively. Platforms focused on e-commerce generate large amounts of data given the amount of products available, customers, product orders, and inventory of said items. However, a daunting task involved with this is both the organization and utilization of the data. So the aim of this project, given these issues, is to create a user-friendly database that simplifies the data management aspect, enhances decision-making for customers, and improves the performance of said e-commerce platforms.

Description:

The database is designed to be used by platforms that focus on e-commerce. It will be a user-friendly experience and will provide an efficient backend infrastructure that supports the management, storage, and retrieval of all data. The database will give a smooth integration with the front-end application/s and use the most recent database technologies to ensure data integrity, optimal performance, and security. It will open the way for clear organization and in depth analysis of many different data types. These include, but not limited to, product information, customer details, order details, and records of inventory. The focus of this project is to create an easy to use database that meets the requirements of most e-commerce businesses.

Unique Features:

1. **Product Categorization:** The database system will provide tools that are easy to use for tagging and categorizing products. This feature will allow sellers to organize their product inventory enabling customers to find relevant products quicker.
2. **Market Segmentation:** The system will include functionality for segmenting customers based on various criteria. This feature will give sellers the ability to target specific customers with personalized marketing and promotions.
3. **Inventory Tracking:** The database system will offer inventory tracking and alerts for low stock levels. This feature will help sellers to manage their inventory, prevent stockouts, and ensure timely restocking.
4. **Order Analytics:** The system will provide analytical tools to create insights from order data, such as order frequency, average order value, and popular product

combinations. This feature helps sellers understand customer behavior, optimize pricing strategies, and identify up selling opportunities.

Existing Software Tools or Products:

1. Shopify: As a widely recognized e-commerce platform, Shopify offers a user-friendly interface and robust features for online businesses. By integrating the proposed database with Shopify, businesses can efficiently manage data, personalize marketing, track inventory, and gain valuable insights from order data.

2. Magento: Known for its flexibility and scalability, Magento is a popular e-commerce platform. Integrating the proposed database with Magento enhances data management, customer targeting, inventory tracking, and decision-making processes for businesses operating on this platform.

Functional Requirements

1. User:
 - 1.1. Many users shall be able to register with unique accounts.
 - 1.2. A user shall be able to log in using their credentials.
 - 1.3. A user shall be able to track the status of their orders.
 - 1.4. A user shall be able to subscribe or unsubscribe from newsletters or promotional emails.
 - 1.5. A user shall have a tracking id.
 - 1.6. ISA User:
 - 1.6.1. Seller (ISA User):
 - 1.6.1.1. A seller shall be able to edit additional information such as store name, contact details, and business information.
 - 1.6.1.2. A seller shall be able to manage their product inventory.
 - 1.6.1.3. A seller shall be able to create and manage product listings.
 - 1.6.1.4. A seller shall be able to view and manage orders specific to their store.
 - 1.6.1.5. A seller shall be able to track their sales and revenue.
 - 1.6.2. Registered User (ISA User):
 - 1.6.2.1. A registered user shall be able to update their personal info.
 - 1.6.2.2. A registered user shall be able to manage saved payment methods belonging to them.
 - 1.6.2.3. A registered user shall be able to view personalized product recommendations.
 - 1.6.2.4. A registered user shall be able to provide feedback or ratings for products.
 - 1.6.2.5. A registered user shall be able to manage their shipping addresses.
 - 1.6.2.6. A registered user shall be able to manage their payment methods.
2. Product:
 - 2.1. A product shall have a unique identifier.
 - 2.2. A product shall have a name, description, and price.
 - 2.3. A product shall be assigned to one or multiple categories.
 - 2.4. A product shall have one or multiple images associated with it.
 - 2.5. A product shall have stock availability information.

- 2.6. A product shall have specifications or attributes.
- 2.7. A product shall be able to be added to a user's cart.
- 2.8. A product shall have related products or recommendations.
- 2.9. A product shall be able to be searched or filtered by various criteria.
- 2.10. A product can be associated with multiple tags or keywords.
- 2.11. A product shall have only one main image.
- 2.12. A product will have at least one description.
- 3. Cart:
 - 3.1. A cart shall be associated with a user.
 - 3.2. A cart shall contain multiple products with quantities.
 - 3.3. A cart shall calculate the total price of all products.
 - 3.4. A cart shall be able to be saved for later by A registered user.
 - 3.5. A cart shall be able to be converted into an order by a user.
 - 3.6. A product can be added to multiple carts.
- 4. Order:
 - 4.1. An order shall be associated with at most onen registered user.
 - 4.2. An order shall have a unique order number.
 - 4.3. An order shall contain multiple products with quantities.
 - 4.4. An order shall include shipping and billing information.
 - 4.5. An order shall have a total price and payment status.
 - 4.6. An order shall have a status indicating if it is processing, shipped, or delivered.
 - 4.7. An order shall be able to be canceled or returned.
 - 4.8. An order shall include shipment tracking details.
 - 4.9. A registered user shall have the option to provide feedback or ratings for products in an order.
 - 4.10. A product can be part of multiple orders.
 - 4.11. An order will only have one shipping address.
 - 4.12. An order shall only have one billing address.
- 5. Category:
 - 5.1. A category shall have a unique identifier.
 - 5.2. A category shall have a name and description.
 - 5.3. A category shall be able to have parent or child categories.
 - 5.4. A category can have multiple parent categories.
 - 5.5. A category can have multiple child categories.
 - 5.6. A category shall be able to be assigned to multiple products.
 - 5.7. A category shall have a hierarchical structure.

- 5.8. A category may have a parent category.
- 5.9. A category may have one or multiple child categories.
- 5.10. A category shall be able to display its parent and child categories.
- 5.11. A category shall support multiple levels of nesting in the category hierarchy.
- 5.12. A category shall allow navigation through the hierarchical structure, enabling users to browse and filter products within specific categories and their subcategories.
- 6. Payment:
 - 6.1. A payment shall be associated with an order.
 - 6.2. A payment shall have a unique identifier or transaction ID.
 - 6.3. A payment shall have an amount and currency.
 - 6.4. A payment shall have a payment method (credit card, PayPal, etc.).
 - 6.5. A payment shall have a status indicating if it is authorized, captured, or refunded.
 - 6.6. A payment method shall be associated with one or many registered users.
 - 6.7. A registered user shall have at most one default payment method.
 - 6.8. An order can have multiple payments (e.g., split payment).
- 7. Review:
 - 7.1. A review shall be associated with only one product.
 - 7.2. A review shall be associated with only one registered user.
 - 7.3. A review shall have a rating (star rating, thumbs up/down, ...).
 - 7.4. A review shall have a title and written feedback.
 - 7.5. A review shall have a timestamp indicating when it was submitted.
 - 7.6. A product can have multiple reviews.

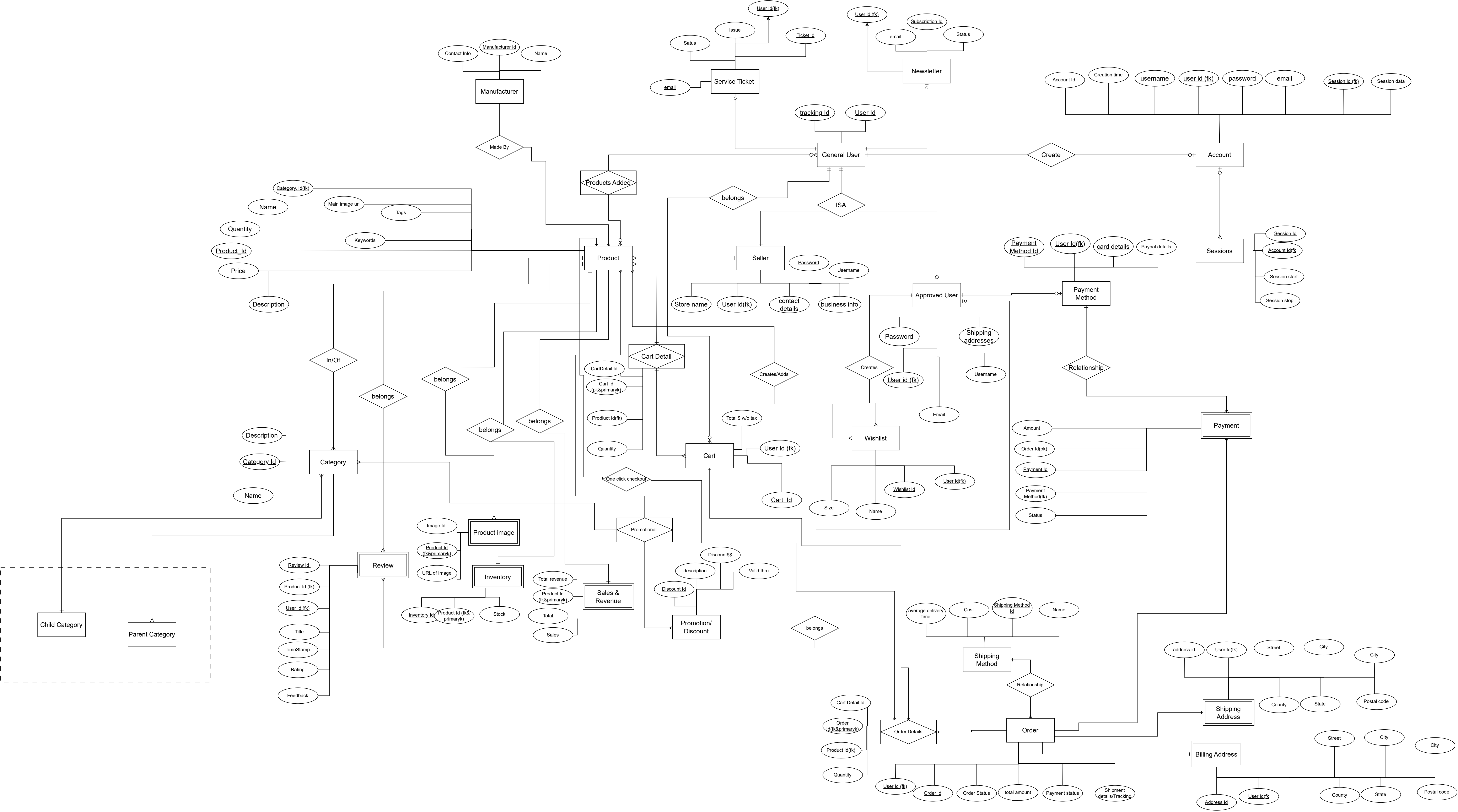
Non-Functional Requirements

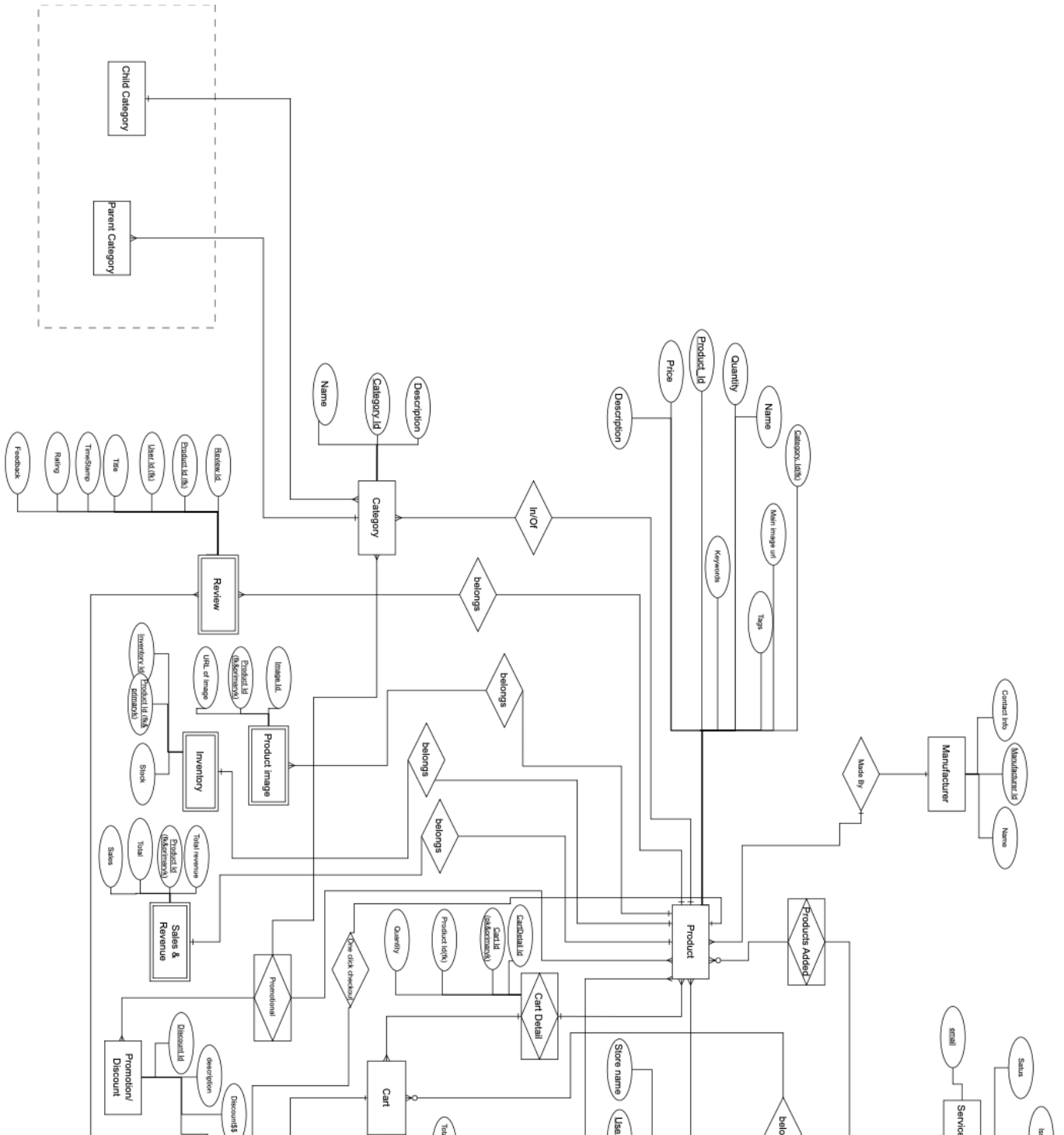
1. Performance:
 - 1.1. The system shall respond to user interactions within an acceptable time frame.
 - 1.2. The system shall handle a large number of users at once without significant lag.
 - 1.3. The system shall ensure quick and efficient loading of product media.
2. Reliability:
 - 2.1. The system shall be available and accessible to users with minimal downtime.
 - 2.2. The system shall be resilient to failures and errors and provide appropriate error handling and recovery tools.
 - 2.3. The system shall backup and securely store user data to prevent data loss.
3. Security:
 - 3.1. The system shall protect user data from unauthorized access.
 - 3.2. The system shall encrypt sensitive user data, ie. passwords/payment information.
 - 3.3. The system shall implement HTTPS to protect data during exchange between users and the system.
 - 3.4. The system shall adhere to industry standards and best practices for security, including updates.
4. Scalability:
 - 4.1. The system shall be designed to accommodate future growth and increasing user demand.
 - 4.2. The system shall scale horizontally or vertically to handle increased load and traffic.
 - 4.3. The system shall be able to handle a growing product catalog and increasing database size without performance decline.
5. Storage:
 - 5.1. The database shall assign 10 MB of memory per table.
 - 5.2. The database system should support persistent storage.
6. Usability:
 - 6.1. The user interface shall be user-friendly and easy to navigate.
 - 6.2. The system shall give clear error messages to resolve any issues promptly

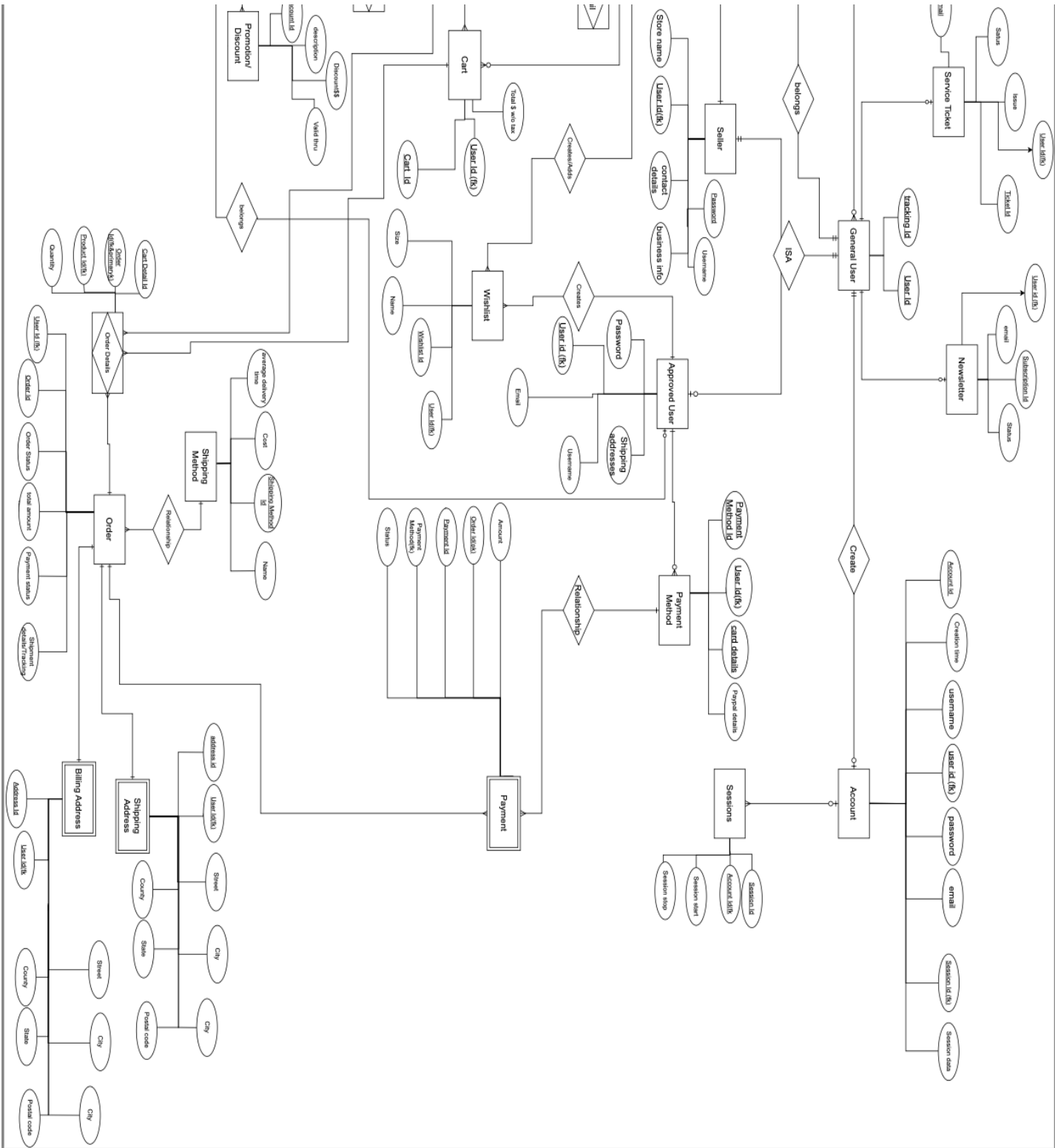
7. Maintainability:
 - 7.1. The system shall be modular and maintainable, allowing for easy updates and enhancements.
 - 7.2. The system shall have proper documentation to facilitate maintenance and troubleshooting.
 - 7.3. The system shall provide an efficient and reliable mechanism for system administrators to apply updates and patches.
8. Compatibility:
 - 8.1. The system shall be compatible with popular web browsers and mobile devices.
 - 8.2. The system shall integrate with external services, such as payment gateways and shipping providers, following their specified integration guidelines and APIs.
9. Data Integrity:
 - 9.1. The system shall ensure the accuracy and consistency of data stored in the database.
 - 9.2. The system shall implement proper data validation and sanitization.
10. Localization:
 - 10.1. The system shall support multiple languages and handle localized date and time formats, currency, and other locale-specific conventions.
11. Sessions:
 - 11.1. The system shall manage user sessions securely and efficiently.
 - 11.2. The system shall generate a unique session identifier upon user login and associate it with the user's session.
 - 11.3. The system shall provide a mechanism to invalidate sessions upon user logout.
 - 11.4. The system shall enforce session timeouts to automatically log out inactive users and ensure session security.

Entity Relationship Diagram

(NEXT PAGE)







Entity Relationships

Strong Entities

1. Account (Strong)
 - a. account_id: key, numeric
 - b. user_id: foreign key, numeric
 - c. last_login_timestamp: timestamp
 - d.
2. Cart (Strong)
 - a. cart_id: key, numeric
 - b. user_id: foreign key, numeric
 - c. total_price: numeric
 - d.
3. Category (Strong)
 - a. category_id: key, numeric
 - b. name: composite, alphanumeric
 - c. description: composite, alphanumeric
4. Session (Strong)
 - a. session_id: key, numeric
 - b. account_id: foreign key, numeric
 - c. login_timestamp: timestamp
5. Discount (AKA PROMOTION) (Strong)
 - a. discount_id: key, numeric
 - b. discount_code: composite, alphanumeric
 - c. discount_value: numeric
6. Payment Method (Strong)
 - a. payment_method_id: key, numeric
 - b. user_id: foreign key, numeric
 - c. method_detail: composite, alphanumeric
 - d.
7. Shipping Method (Strong)
 - a. shipping_method_id: key, numeric
 - b. method_detail: composite, alphanumeric
 - c. cost
8. Order (Strong)

- a. order_id: key, numeric
 - b. user_id: foreign key, numeric
 - c. total_price: numeric
- 9. Approved User (Strong, ISA General User)
 - a. user_id: key, numeric (inherited)
 - b. username: composite, alphanumeric
 - c. email: composite, alphanumeric
- 10. Manufacturer (Strong)
 - a. manufacturer_id: key, numeric
 - b. name: composite, alphanumeric
 - c. contact_detail: composite, alphanumeric
- 11. Newsletter (Strong)
 - a. newsletter_id: key, numeric
 - b. newsletter_type: composite, alphanumeric
 - c. subscription_status: boolean
- 12. General User (Strong)
 - a. user_id: key, numeric
 - b. tracking_id: composite, alphanumeric
 - c. user_type: multivalue, alphanumeric
- 13. Service Ticket (Strong)
 - a. ticket_id: key, numeric
 - b. user_id: foreign key, numeric
 - c. issue_description: composite, alphanumeric
- 14. Wishlist (Strong)
 - a. wishlist_id: key, numeric
 - b. user_id: foreign key, numeric
 - c. product_ids: multivalue, numeric
- 15. Seller (Strong, ISA General User)
 - a. user_id: key, numeric (inherited)
 - b. store_name: composite, alphanumeric
 - c. contact_details: composite, alphanumeric
- 16. Product (Strong)
 - a. product_id: key, numeric

- b. name: composite, alphanumeric
- c. price: numeric

Weak Entities:

1. Payment (Weak)
 - a. payment_id: key, numeric
 - b. order_id: foreign key, numeric
 - c. amount: numeric
2. Shipping Address (Weak)
 - a. shipping_id: key, numeric
 - b. user_id: foreign key, numeric
 - c. address_detail: composite, alphanumeric
3. Billing Address (Weak)
 - a. billing_id: key, numeric
 - b. user_id: foreign key, numeric
 - c. address_detail: composite, alphanumeric
4. Sales and Revenue (Weak)
 - a. revenue_id: key, numeric
 - b. seller_id: foreign key, numeric
 - c. sales: numeric
 - d. revenue: numeric
5. Inventory (Weak)
 - a. inventory_id: key, numeric
 - b. product_id: foreign key, numeric
 - c. stock_availability: numeric
6. Product Image (Weak)
 - a. image_id: key, numeric
 - b. product_id: foreign key, numeric
 - c. image_location: alphanumeric
7. Review (Weak)
 - a. review_id: key, numeric
 - b. user_id: foreign key, numeric
 - c. product_id: foreign key, numeric
 - d. rating: numeric
 - e. feedback: composite, alphanumeric