Erot Store *(it's a palindrome, deal with it!)*

by Derrick Pelletier and Brant Hardy

Table of Contents

1.0 - Introduction

Erot Store is an online store website that will allow users to browse and purchase products that will be shipped to their homes.

This document outlines the design requirements for the Erot Store e-commerce website. The design is tailored in such a way as to be scalable from a small store with few products to a large store with thousands of products.

1.1 - Mission Statement

To design and implement a compact, easy to use e-commerce software package that provides Administrators of potential client companies to quickly install a robust online store website. The store will be secure with extensive options for customization.

Administrators will have no restrictions in the number of categories and items that they wish to add to their store.

Users will be able to browse, search, and filter products in a user-friendly manner. They will be able to add and remove items from their cart, and make secure payments through the website.

1.2 - Executive Summary

Erot Store will support guests, registered users, and company administrators.

Both guests and users will be able to browse, search, and sort the items in the store. Guests will not be able to add items to their cart. Guests will be able to register as users by filling out the registration form.

Registered users will be able to login, add products to their cart, modify cart items (quantity, remove), select shipping options, and checkout and pay for their orders. Users will be able to view past orders and check the status or cancel recently made orders. Users will be able to modify account information and save their shipping address for reuse. Users will also be able to add/remove products to their wish-list that they can access and share with friends. Users will be able to review products that they have purchased. Users will be able to reset their passwords if they forget them.

Admin is a special user account that will allow company employees to modify products, add products, delete products, create new categories, resolve user issues, and generally provide tech support for the website. Admins will be able to see current inventory remaining for products and order more. Admins will be able to modify content, add sale products, blog posts, and create new admin accounts. Administrators will be able to view sales reports and charts, orders waiting to be filled, moderate user product reviews.

If the client’s expectations are satisfied and there are no time constraints, the following features will be implemented: time-based automatic sales, preordering products, multiple pictures for products, saving multiple shipping and billing methods for users, forgot-password reset, admin ability to modify time to test preorder and automatic sales, admin report for viewing top customers, customer rewards program.

The store website system will be modeled using Entity Relationship (ER) diagrams. The system will be implemented using MySQL and PHP database programming techniques. The Erot Store demo software will be installed on our own LAMP server environment. Clients who purchase the product will be able to setup and install the software their own server.

2.0 - Domain Assumptions:

2.1 - Users

- have a unique id (which is their email address) and a password (which they will create) which will allow them to log into the site, and will allow the site to recognize them.
- can add products to their wish-list.
- can add products to their cart and purchase them.
- can cancel recently made orders.
- can rate products they have purchased.
- have a saved preferred-payment method.
- have a saved shipping address.
- can reset their forgotten passwords.

2.2 - Guests

- can browse products in the store.
- can view website content (blogs, pages, help, etc).
- can create accounts.

2.3 - Administrators

- can edit the contents of the store, including: products, reviews, content.
- can specify a product as a pre-order.
- can create new sales.
- can create new administrator accounts.
- can view sales data and reports
- can view hidden product information like supplier and inventory
- can resupply low inventory.

2.4 - Products

- have: ids, names, images, descriptions, prices, inventory.
- can be anything that an administrator wants to sell.
- may have a third party supplier.
- Pre-order products will alert and process orders when product is available.

2.5 - Orders

- are placed from the user-created shopping cart.
- contain products and quantities of those products, user, date, shipping information
- contain billing information

- admins will process orders and billing information

2.6 - Shipping Information

- administrators select which shipping options are available from: *FedEx, UPS, Purolator, Postal Service*.
- users select from available shipping methods.
- user can select priority shipping methods when available based on their location.

2.7 - Payment Options

- administrators select which payment methods are acceptable from: *Paypal, Visa, MasterCard*.
- users select from available payment methods.

3.0 - Entity Description

3.1 - Relational Assumptions

The **Users:** Contains information that allows the user to be identified. The user can rate and order products. The user can also be an admin (flagged by a boolean), which allows them to modify the store.

The **Orders:** Must be placed by a registered user and contain at least 1 product. **Orders** specify where a product will be shipped and the billing address.

The **Products:** Contains information on products available for purchase from the store. The amount of the products in the inventory is stored, as well as any special sale pricing.

The **ProductReviews:** Users can leave a review for a product they have purchased. The review includes a numerical rating.

The **WishlistedProducts:** Is a collection/list of products a user would like to purchase.

The **Categories:** Are groupings of products for organization. A product may belong to multiple categories.

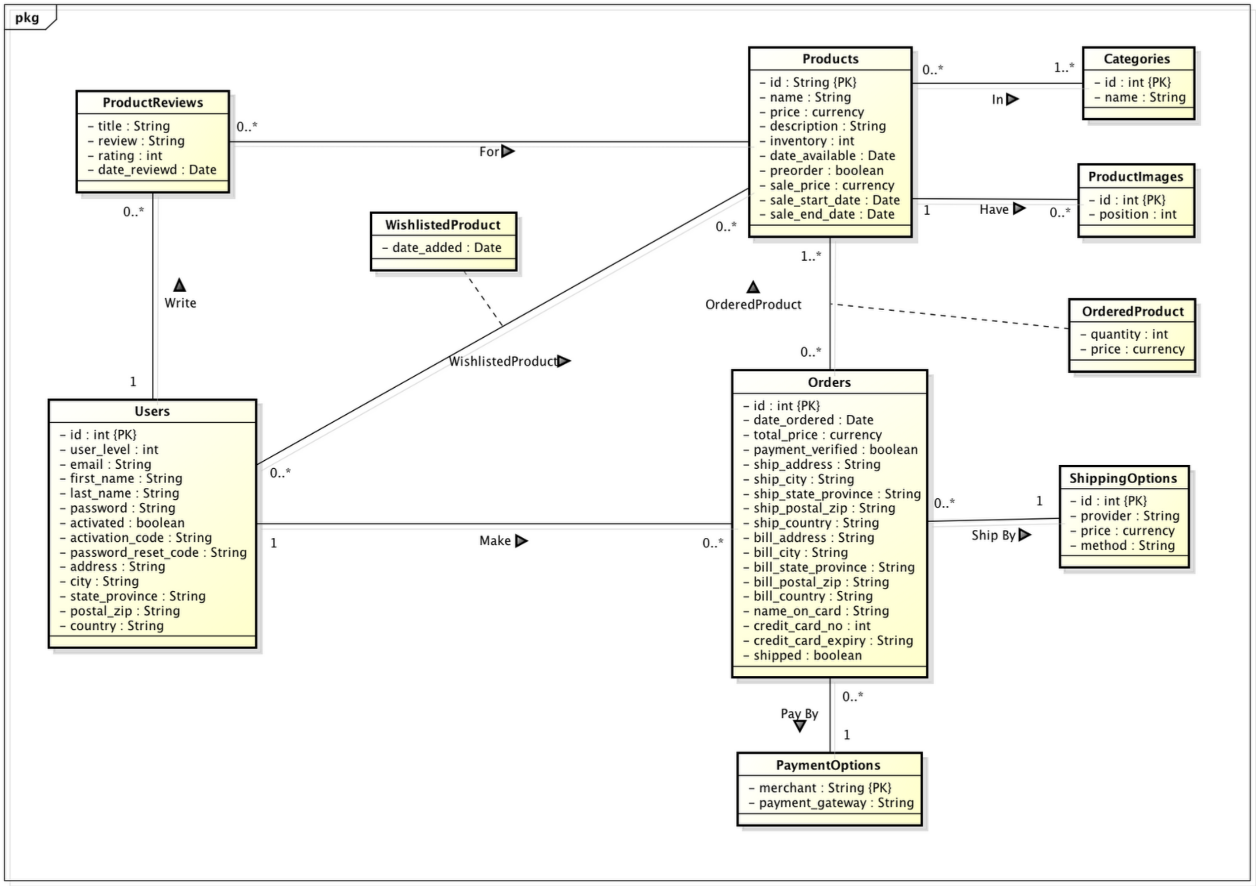
The **ProductImages**: A collection of images uploaded by an admin, associated with a certain product.

The **OrderedProducts:** A relation that keeps track of the products contained in one order. It keeps a copy of the price at the time of purchase and the quantity ordered of that product.

The **PaymentOptions:** A selection of the different payment options that are made available by the seller/admin and chosen by the purchaser at checkout.

The **ShippingOptions:** A selection of the different shipping options that are made available by the seller and chosen by the purchaser at checkout.

3.2 - ER-UML Diagram



3.3 - Entity Attributes

3.3.1 - Users

Attribute	Description
id {PK}	Autoincremented unique integer to identify a user in the table.
user_level	Integer value represent user's access level to site.
email	User's email address, unique.
first_name	User's first name.
last_name	User's last name.
password	User's password, probably md5 salted hash.
activated	Boolean value whether user has confirmed account via emailed link.
activation_code	Random generated code sent to user in email when

	account is created to activate account.
password_reset_code	Random generated code sent to user when resetting password.
address	User’s address.
city	User’s city.
state_province	User’s state or province.
postal_zip	User’s postal or zip code.
country	User’s country.

3.3.2 - Products

Attribute	**Description**
id {PK}	A unique, provided ID that identifies a product.
category_id {FK}	The id of the category that the product belongs to.
name	Name of product.
price	Price of product.
description	Description of product.
inventory	Inventory of product visible only on admin side.
date_available	The date the product will become available on the client-side.
preorder	Boolean value to flag whether product may be preordered before it is available.
sale_price	The sale price of a product.
sale_start_date	The start date of a sale on the product.
sale_end_date	The end date of a sale on the product.

3.3.3 - Orders

Attribute	**Description**
id {PK}	Autoincremented unique integer to identify an order in the table.
user_id {FK}	The id of the user who made the order.
total_price	Calculated total price of the order based on the products and their quantities in the order.
payment_verified	Boolean value indicating whether or not payment has been made.
ship_address	Number and Street that the order is shipped to.
ship_city	City that the order is shipped to.
ship_state_province	State or Province that the order is shipped to.
ship_country	Country that the order is shipped to.
bill_address	Number and Street that the order is billed to.
bill_city	City that the order is billed to.
bill_state_province	State or Province that the order is billed to.
bill_country	Country that the order is billed to.
name_on_card	The name that appears on the Credit Card used to pay.

credit_card_number	The credit card number that is used to pay for the order.
credit_card_expiry	The expiry date of the credit card that is used to pay for the order.
shipped	A boolean value to whether the order has been shipped.
payment_option {FK}	The payment option chosen for the order.
shipping_option_id {FK}	The shipping option chosen for the order.

3.3.4 - ProductReviews

Attribute	**Description**
user_id {PK, FK}	the id of the user who wrote the review
product_id {PK, FK}	the id of the product being reviewed
title	Title of the review
review	The review text
date_reviewed	The date the review was written

3.3.5 - WishlistedProduct

Attribute	**Description**
user_id {PK, FK}	The id of the user who owns the wishlist.
product_id {PK, FK}	The id of the product the user wants.
date_added	The date the item was added to the wishlist.

3.3.6 - Categories

Attribute	**Description**
id {PK}	the category ID that uniquely identifies it
name	The name of the category

3.3.7 - ProductImages

Attribute	**Description**
id {PK}	The ID of the image.
product_id {FK}	The id of the product that the image represents
position	The order that the images will be displayed if there are more than one.

3.3.8 - OrderedProducts

Attribute	**Description**
order_id {PK, FK}	The id of the order the product is in
product_id {PK, FK}	The id of the product in the order
quantity	The quantity of the product ordered
price	The price of the product at the time of order

3.3.9 - PaymentOptions

Attribute	**Description**
merchant {PK}	The name that uniquely identifies a merchant (Paypal, MasterCard, Visa, etc)
payment_gateway	The gateway method the payment will be processed through.

3.3.10 - ShippingOptions

Attribute	**Description**
id {PK}	The unique identifier of a shipping option
provider	The provider of the shipping option (UPS, FedEx, etc)
method	The method of shipment through the selected provider (Express, Ground, etc)
price	The price of the selected method

4.0 - User Interface

The user interface will be delivered with HTML5 and PHP technology. Modern browsers will be able to view the Erot Store the way it is intended.

All users will see the website in much the same way. They will sign in using the same form and they will be able to browse the storefront in the same manner. There are three types of users: the first is the *shopper*- who can simply browse, sort and purchase; the second is the *editor*- who has access to the administration dashboard and can add, remove, and modify products; the third is the *admin*- who has access to the administration dashboard and the power modify the system settings. Administrators' accounts will allow them special access to edit the contents of the store, as well to view order histories, reports, and confidential information that will be kept hidden from the users.

Every page will contain the company logo, company name, a navigation menu, a sign-in form, and a search bar, as well as a footer containing company, copyright, and privacy information. Users who are logged in will always be able to see if they have any items in their cart. Guests who are not logged in will always be presented with a link to the sign-up form.

The site will consist of the following page templates: home, content, category list, products list, product details, cart, checkout forms, order details, wishlist, administration dashboard, administration sub-settings pages, user account page, sign in, password reset, sign up.

The Following are the primary page templates:

Home Page

The home page will contain a list of featured (or on-sale) products as well as any recent company news or other desired content.

Product Page

Each product page will include the title of the product, as well as a gallery of images, the description, the price, and a options for the user to add the item to their cart or wishlist. Signed-in users who have previously bought the product will also be presented with form to leave a product review.

Category/Search-result Listing Page

The listings pages will display products from the database in a grid. The listings page will display results for sale products, search results, or categories. Users will also be able to filter the results by provided attributes.

User Account Page

Users will be able to see all of their stored account information from the User Account Page. Users will be able to edit their names, emails, addresses, and change their passwords. Users will also be able to view their order history.

Admin Dashboard

The admin dashboard will be a restricted portion of the website granted only to users with the appropriate ***user_level***. The dashboard contains all of the features required for viewing and modifying the store contents, settings, reports, content, and other confidential administration-oriented tasks. Access to the various settings will be dependent on the ***user_level*** of the current user.

5.0 - Relational Schema — SQL DDL

```
CREATE TABLE `Orders` (
  `id` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `date_ordered` date NOT NULL,
  `total_price` float NOT NULL,
  `payment_verified` tinyint(1) NOT NULL DEFAULT '0',
  `ship_address` varchar(255) NOT NULL DEFAULT ",
  `ship_city` varchar(100) NOT NULL DEFAULT ",
  `ship_state_province` varchar(100) NOT NULL DEFAULT
",
  `ship_postal_zip` varchar(10) NOT NULL DEFAULT ",
  `ship_country` varchar(100) NOT NULL DEFAULT ",
  `bill_address` varchar(100) NOT NULL DEFAULT ",
  `bill_city` varchar(100) NOT NULL DEFAULT ",
```

```
CREATE TABLE `Categories` (
  `id` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `name` varchar(40) NOT NULL DEFAULT ",
  PRIMARY KEY (`id`)
);

CREATE TABLE `OrderedProducts` (
  `order_id` int(11) NOT NULL DEFAULT '0',
  `product_id` int(11) NOT NULL DEFAULT '0',
  `quantity` int(11) NOT NULL,
  `price` float NOT NULL,
  PRIMARY KEY (`order_id`,`product_id`),
  FOREIGN KEY (`order_id`) REFERENCES `Order`(`id`)
  `bill_state_province` varchar(100) NOT NULL DEFAULT '',
  `bill_postal_zip` varchar(10) NOT NULL DEFAULT '',
  `bill_country` varchar(100) NOT NULL DEFAULT '',
  `name_on_card` varchar(100) DEFAULT NULL,
  `credit_card_no` int(11) DEFAULT NULL,
  `credit_card_expiry` varchar(5) DEFAULT NULL,
  `shipping_option_id` int(11) NOT NULL,
  `payment_option` varchar(255) NOT NULL DEFAULT '',
  `shipped` tinyint(1) NOT NULL DEFAULT '0',
  PRIMARY KEY (`id`),
  FOREIGN KEY (`shipping_option_id`) REFERENCES
`ShippingOptions`(`id`)
    ON DELETE SET NULL
    ON UPDATE CASCADE,
  FOREIGN KEY (`payment_option`) REFERENCES
`PaymentOptions`(`merchant`)
    ON DELETE SET NULL
    ON UPDATE CASCADE
);

CREATE TABLE `ProductReviews` (
  `title` varchar(100) NOT NULL DEFAULT '',
  `review` text,
  `rating` int(11) NOT NULL,
  `date_reviewed` date NOT NULL,
  `user_id` int(11) NOT NULL DEFAULT '0',
  `product_id` int(11) NOT NULL DEFAULT '0',
  PRIMARY KEY (`product_id`,`user_id`),
  FOREIGN KEY (`product_id`) REFERENCES
`Products`(`id`)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  FOREIGN KEY (`user_id`) REFERENCES `Users`(`id`)
    ON DELETE CASCADE
    ON UPDATE CASCADE
);

CREATE TABLE `Products` (
  `id` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `name` varchar(100) NOT NULL DEFAULT '',
  `price` float NOT NULL,
  `description` text,
  `inventory` int(11),
  `date_available` date,
  `preorder` tinyint(1),
  `sale_price` float,
  `sale_start_date` date,
  `sale_end_date` date,
  `category_id` int(11),
  PRIMARY KEY (`id`),
  FOREIGN KEY (`category_id`) REFERENCES
`Categories`(`id`)
    ON DELETE SET NULL
    ON UPDATE CASCADE
);

CREATE TABLE `WishlistedProducts` (
  `product_id` int(11) NOT NULL DEFAULT '0',
  `user_id` int(11) NOT NULL DEFAULT '0',
  `date_added` date NOT NULL,
  PRIMARY KEY (`product_id`,`user_id`),
  FOREIGN KEY (`user_id`) REFERENCES `Users`(`id`)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  FOREIGN KEY (`product_id`) REFERENCES
`Products`(`id`)
    ON DELETE CASCADE
    ON UPDATE CASCADE
);
```

```
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  FOREIGN KEY (`product_id`) REFERENCES
`Product`(`id`)
    ON DELETE SET NULL
    ON UPDATE CASCADE
);

CREATE TABLE `PaymentOptions` (
  `merchant` varchar(100) NOT NULL DEFAULT '',
  `gateway` varchar(255) NOT NULL DEFAULT '',
  PRIMARY KEY (`merchant`)
);

CREATE TABLE `ProductImages` (
  `id` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `position` int(11) NOT NULL DEFAULT '0',
  `product_id` int(11) NOT NULL,
  PRIMARY KEY (`id`),
  FOREIGN KEY (`product_id`) REFERENCES
`Products`(`id`)
    ON DELETE CASCADE
    ON UPDATE CASCADE
);

CREATE TABLE `ShippingOptions` (
  `id` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `provider` varchar(255) NOT NULL DEFAULT '',
  `price` float NOT NULL,
  `method` varchar(255) NOT NULL DEFAULT '',
  PRIMARY KEY (`id`)
);

CREATE TABLE `Users` (
  `id` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `email` varchar(255) NOT NULL DEFAULT '',
  `first_name` varchar(100) DEFAULT NULL,
  `last_name` varchar(100) NOT NULL DEFAULT '',
  `password` varchar(255) DEFAULT NULL,
  `activated` tinyint(1) NOT NULL DEFAULT '0',
  `activation_code` varchar(255) DEFAULT NULL,
  `password_reset_code` varchar(255) DEFAULT NULL,
  `address` varchar(100) DEFAULT NULL,
  `city` varchar(100) DEFAULT NULL,
  `state_province` varchar(100) DEFAULT NULL,
  `postal_zip` varchar(10) DEFAULT NULL,
  `country` varchar(255) DEFAULT NULL,
  `admin` tinyint(1) NOT NULL DEFAULT '0',
  PRIMARY KEY (`id`)
);
```