COMP321 Information System Implementation

Project Specification

Online Shopping Mall

Introduction

Overview

The Web has provided a unique opportunity for retail businesses. Customers may do window shopping over a wide range of products without limitation on business hours and transportation. Vendors save huge expense by not running a bricks-and-mortar shop and can provide services for both large and niche markets. Online shopping sites like Taobao, Amazon and (to a lesser degree) eBay are some successful examples. In addition, open-source e-commerce systems (e.g. Magento, osCommerce and OpenCart) provide low-cost solutions to set up an online store.

This project "Online Shopping Mall web application" is aimed at you providing an online shopping experience to customers. The vendor displays their products in a manner that is easy for potential customers to select and make purchases.

Scope

This project focuses on facilitating the online shopping experience. Most of the functional requirements are about browsing and maintaining the product catalog, and placing and processing purchase orders.

To limit the scope, this project omits payment processing and inventory management. The delivery process is assumed to be instantaneous. Right after a vendor ships a purchase a customer receives the purchase.

Functional requirements

The web application supports two main kinds of users: customers and vendors. The following are the functions provided for each kind of user. Functional requirements are broken into several areas (e.g. A, B, C) and individual requirements are labelled alphanumerically (e.g. A1, A2 …). A requirement with a shaded background (e.g. A1) is compulsory. Key points in some requirements are highlighted in **bold** for quick reference.

Customers

Customers do ‘window shopping’ in store at the frontend of the mall, browse and select desired products, and place orders for products in the shopping cart. The application supports the following functions.

The front page of the shopping mall shows a **product list**.

- (A1) A customer may **browse products** in a list of products. The list shows basic information of products, including product name, brand, price and a thumbnail image. Each product belongs to one of the pre-defined brands. (You can also use category instead of brand).
- (A2) The product list supports **paging**. The customer can navigate the product list by ‘page up’, ‘page down’ and jumping to a specific page. Paging works properly after applying a filter or sorting as listed below.
- (A3) The customer can **filter** the product list **by brand**. They can also list products of all brands.
- (A4) The customer may filter the product list by **searching keywords** in the product name. This function work correctly with the brand filter.
- (A5) The customer may **sort** the product list **by price**.

The product detail page shows detailed information about one product.

- (A6) The customer may select a product in the product list to go to the product detail page. The **product detail page** shows information for one product, which includes the product name, brand, price and a thumbnail image. In addition, the product detail page also shows detail description as a list of at least two properties. For example, the product detail page for a book may show authors, ISBN, publisher, release date and number of pages.
- (A7) The product detail page supports display of **one or more detailed photographs** of the selected product.

The system has basic **account management** for customers.

- (B1) A customer may **register a new account**. They have to provide full name, email address, password and shipping address. After registration, the user is logged in automatically.
- (B2) A customer may **log in** and **log out**, and the interface shows the name of the current user. The product list and product detail page are accessible to customers without login. On the other hand, the shopping cart and purchase tracking are only accessible after login.
- (B3) The customer can **change password**. There is strength requirement for password. The password should contain at least 6 characters, in which there must be at least one digit and one capital letter.
- (B4) If a customer tries to add a product to the shopping cart on the product detail page without first logging in, the system **redirects** the user **to the login page**. After successful login, the system redirects the user back to the original product detail page.

- (B5) The server only saves **hash values** of customers' passwords. Passwords are never saved in plain text in the server.

To make any purchase, a customer must add products to his/her **shopping cart**. The customer can check out all items in the cart to place an order.

- (C1) The customer **adds a product** to his/her shopping cart by clicking a button in the product detail page. The quantity to buy is assumed to be 1. The items in shopping cart are persisted across user sessions. Next time the customer logs in, they can still see the items in the shopping cart.
- (C2) The customer can **list the products** in his/her shopping cart in a shopping cart page. In this page, the entry for each product shows the product name, price and the quantity to buy. The page also shows the total order amount (i.e. how much the customer has to pay in total) in the shopping cart. The customer can click an item in the shopping cart to go to the product detail page of the entry.
- (C3) The customer can press a button in the shopping cart page to **check out** all items in the shopping cart. This action creates a purchase order with a newly allocated unique P.O. number, and clears the content of the cart. After checkout, the system shows the purchase order detail page of the newly created purchase order. *(refer to requirement D3)*.
- (C4) The shopping cart page allows the customer to **change the quantity** of an item. This allows the customer to order more than one piece of a product (e.g. buy two copies of a book).
- (C5) The customer can **remove an item** from the shopping cart.
- (C6) If the customer adds a **duplicate product** to the shopping cart, the application will give a warning message and does not change the content of the shopping cart.

Purchase tracking: After placing an order, the customer can trace the processing status of the order in a **purchase tracking page**. For simplicity, we assume that each purchase order is fulfilled in a single shipping package. The purchase order status describes the various stages of order processing. Possible values include 'pending', 'shipped', 'hold', and 'cancelled'.

- (D1) The **purchase tracking page** lists the purchase orders that the customer has placed. This page shows the following for each purchase order: the P.O. number, the purchase date, the total order amount and the purchase order status. The purchase orders are displayed in reverse chronological order of purchase date. When the customer clicks an entry in the list, they can see the detail in a purchase order detail page.
- (D2) The customer can filter the list of purchase orders in two ways. First, the page only shows 'current purchases' with status 'pending' and 'hold'. Second, the page only shows 'past purchases' with status 'shipped' and 'cancelled'.

- (D3) The **purchase order detail page** shows the P.O. number, the purchase date, the customer name, the shipping address, the total order amount and the purchase order status. If the order is shipped, this page shows the shipment date. If the order is cancelled, the page shows the order cancel date and who (customer or vendor) cancelled the order. The page also includes, for each product in the purchase order, the product name, the quantity, the unit price and the subtotal.
- (D4) Before a purchase order is shipped, the customer can **cancel the order**. This can be done by clicking a button in the purchase order detail page. This action will change the status of the purchase order to 'cancelled'. Note that this action is only available for purchase orders in the status 'pending' or 'hold'.

Vendor

The vendor maintains a product catalog in the shopping mall. They can also process purchase orders from customers. Because there is only one vendor, the system only needs to implement a single vendor user account. No account management of vendor accounts is necessary in this project. The application provides the following functions.

Product catalog maintenance: The vendor can browse the product catalog, edit some properties of a product, and add new products.

- (E1) The vendor may **browse the product catalog** in an interface similar to product list for customers. *(Refer to requirements A1, A2, A3 and A6)*. The vendor is not a customer, and no shopping cart or 'add to cart' button should be shown.
- (E2) The vendor can find products by searching keywords in product names. They can also find a specific product by entering a unique product ID.
- (E3) The vendor may **add a new product** to the catalog. The vendor enters basic information of the product, including product name, brand, price and a thumbnail image. They can enter detail information of the new product as a list of properties. *(Refer to requirement A6*)
- (E4) In addition to the thumbnail image, the vendor can **upload 1 to 4 detailed photographs** for a product. These photos are usually of higher resolution and are displayed in the product detail page in a user-friendly interface. *(Refer to requirement A7)*
- (E5) The vendor can **edit information** of a product in a product detail page. They can change the product name and product brand. They can also change detail information as a list of properties. *(Refer to requirement E3).*
- (E6) The vendor can change the thumbnail and detail photos for a product. They can add or remove photos.

Purchase order processing: The vendor can list purchase orders by different status. They may ship, hold, or cancel a purchase order in the purchase order processing page.

- (F1) The **purchase order list page** lists purchase orders received by the application. It shows the P.O. numbers, purchase dates, customer names, total order amounts and purchase order status. The purchase orders are sorted in descending order of purchase date (i.e. newest first). The vendor can click an entry to open a purchase order processing page.
- (F2) The vendor can filter the purchase order list in three ways. They can show only the 'pending orders' (with status 'pending'). They can show only the 'orders on hold' (with status 'hold'). Finally, the vendor can select to show 'past orders' (with status 'shipped' or 'cancelled').
- (F3) The **purchase order processing page** shows similar information as the purchase order detail page *(refer to requirement D3)*. In addition, the vendor can click a button to **ship a purchase order**. This action changes the status of the purchase order from 'pending' to 'shipped' and starts the shipping process.
- (F4) The vendor can enter a P.O. number to view and process a specific purchase order.
- (F5) In the purchase order processing page, the vendor can click a button to **hold a purchase order**. This is useful, for example, if some product in the purchase order is temporarily out-of-stock. This action is only available when the status of the purchase order is 'pending', and this action changes the status to 'hold'.
- (F6) In the purchase order processing page, the vendor can click a button to **unhold and ship a purchase order**. This action changes the status of the purchase order from 'hold' to 'shipped' and starts the shipping process.
- (F7) In the purchase order processing page, the vendor can click a button to **cancel a purchase order**. This is useful, for example, to inform the customer that the ordered products are no longer available. This action is only available for purchase orders in the status 'pending' or 'hold'. This action changes the status of the purchase order to 'cancelled'.

You are also required to design and implement some of these advanced features.

- (Z0) Develop a mobile app as the frontend of the system. The mobile app should communicate with the server backend through a Web API. You have to design and implement both the frontend and the backend.
- (Z1) The vendor needs to analyze the sales of the products and find out the **best selling products**. The report measures sales by both sales quantities (number of items sold) and sales amount (the dollar amount received in sales). The default reporting period is the last 30 days, but the vendor may also customize the reporting period.
- (Z2) Design a **notification** feature to make it to easier for a customer to track the change of status of purchase orders. For example, when the vendor ships a purchase order, the customer will receive a notification message. The interface should distinguish between read and unread notifications. You can use the notification in Facebook as reference.

- (Z3) A customer can express their satisfaction with a product with **customer's rating**. A customer who has purchased a product successfully can rate it on a scale of 1 to 5 stars after the purchase is shipped. Decide whether a customer can rate one product more than once, and whether they can change the ratings afterwards. The product detail page shows customers' average rating as a decimal number (e.g. 3.5 stars). Pay special attention when number of ratings is smaller (e.g. less than 2). Consider how to use the average ratings in product list for customers and the vendor.
- (Z4) In addition to star ratings, customers also want to write short **reviews for products** in the shopping mall. Design a feature to allow a customer to write a short review for a product. Consider how to show these reviews to other customers and the vendor.
- (Z5) Design how to implement **price change** of products. This is useful, e.g., for promotional price reduction or regular price adjustment. Price change should not affect the price in existing purchase order and other historical records.
- (Z6) Sometimes, a hot-selling product is '**out-of-stock**'. Design a feature for the vendor to mark a product as temporarily out-of-stock. Customers can still place orders for such products, but the system should only be able to ship them when the vendor marks the products as in-stock later. Consider how this feature affects the other parts of the system.
- (Z7) If the vendor and the customer change the status of a purchase order at almost the same time, the purchase order may result in an inconsistent state. Implement suitable **concurrency control** to prevent the invalid state change, e.g. pending -> shipped -> cancelled and hold ->cancelled->shipped. You will need to implement transaction in the database system.