

производительности и снижения возможных потерь от внешних воздействий. Избыточность элементов КСОБ предназначена для поддержания работоспособности КСОБ локальной компьютерной сети во время сбоя системы из-за воздействий внешней среды [4].

При исследовании эффективности процессов функционирования КСОБ следует использовать аппарат теории вероятности, так как в процессе формирования оптимальной КСОБ (и в исходных данных) имеется неопределенность, связанная с возможным возникновением угроз, появлением дестабилизирующих факторов, применением программных злоупотреблений. Все эти параметры необходимо учитывать при оценке эффективности функционирования КСОБ РЛКС и количественно измерять их воздействие на информацию [5].

Комплексная система обеспечения безопасности — это средство улучшения имиджа организации, укрепления партнерских отношений, поддержания авторитета, налаживания межкорпоратив-

ных связей на внутреннем и международном рынках.

СПИСОК ЛИТЕРАТУРЫ

1. Пятибратов А. П., Гудыно Л. П., Кириченко А. А. Вычислительные системы, сети и телекоммуникации: Учебник для вузов / Под ред. А. П. Пятибратова. — М.: Финансы и статистика, 1998.
2. Локальные вычислительные сети. Кн. 3. Организация функционирования, эффективность, оптимизация / Под ред. С. В. Назарова. — М.: Финансы и статистика, 1995. — 247 с.
3. Хоникатт Джерри. Использование Интернет / Пер. с англ. — 2-е изд. — М.: Восточная книжная компания, 1996.
4. Технологии электронных телекоммуникаций. Т. 20. Безопасность связи в каналах телекоммуникаций. — М.: Экотрендз, 1992. — 218 с.
5. Бэрри Ханс. Компьютерные сети / Пер. с англ. — М.: Восточная книжная компания, 1996.

Материал поступил в редакцию 23.12.05.

УДК 004.738.5:002.6

Ле Ань Туан

Создание сайта для заказа книг и информационных продуктов*

Описывается разработка интернет-магазина для заказа книг и информационных продуктов. Рассматриваются основные этапы процесса торговли интернет-магазина.

ВВЕДЕНИЕ

Электронная коммерция и небольшая ее часть — интернет-коммерция все больше входит в нашу жизнь. Она открывает новые возможности ведения бизнеса, и многие компании переносят значительную часть делового общения в Интернет.

Интернет-коммерция включает интернет-магазины, биржи и посреднические интернет-аукционы, проведение рекламных кампаний определенных товаров или ресурсов, организацию различных каталогов и средств общения пользователей в Интернете. Коммерческие интернет-приложения предоставляют пользователям все больше возможностей и услуг, и их список постоянно пополняется.

Одну из лидирующих позиций в интернет-коммерции занимают интернет-магазины. Это касается рынка как *B2C (Business To Customer)*, так и *B2B (Business To Business)*.

Данная статья посвящена разработке интернет-магазина для заказа книг и информационных продуктов. Процесс торговли интернет-магазина можно разделить на несколько основных этапов, последовательность которых представлена на рис. 1. Эта

последовательность в данном случае достаточно условна, так как пользователь может в любой момент прервать процесс покупки, выйти из интернет-магазина и через какое-то время вернуться к своим предыдущим незаконченным этапам покупки товара. В этом случае необходимо предоставить покупателю такую возможность.

Для разработки интернет-магазина будем использовать язык создания сценариев — *PHP (Hypertext Preprocessor)* и систему управления реляционными базами данных (*СУРБД*) — *MySQL*.

1. АРХИТЕКТУРА WEB-БАЗ ДАННЫХ

Базовая архитектура *Web*-базы данных включает *Web*-браузер, *Web*-сервер, сценарный механизм и сервер баз данных.

Пример транзакции *Web*-базы данных состоит из этапов, обозначенных цифрами на рис. 2.

1. *Web*-браузер пользователя отправляет *HTTP*-запрос определенной *Web*-страницы.

2. *Web*-сервер принимает запрос и передает его механизму *PHP* на обработку.

* Доклад на 13-м международном семинаре «Научная и техническая информация в Центральной и Восточной Европе». — 13–19 сентября 2004 г.



Рис. 1. Этапы торговли интернет-магазина

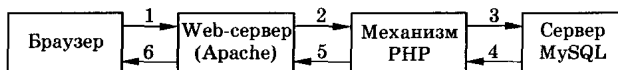


Рис. 2. Этапы транзакций Web-баз данных

3. Механизм *PHP* начинает анализ сценария. В сценарии присутствует команда подключения к базе данных и выполнения запроса в ней. *PHP* открывает соединение с сервером *MySQL* и отправляет необходимый запрос.

4. Сервер *MySQL* принимает запрос, обрабатывает его, а затем отправляет результаты обратно в механизм *PHP*.

5. Механизм *PHP* форматирует результаты запроса в виде *HTML*, после чего отправляет результаты в *HTML*-формате *Web*-серверу.

6. *Web*-сервер пересылает *HTML* в браузер, с помощью которого пользователь просматривает *Web*-страницы.

2. РАЗРАБОТКА ИНТЕРНЕТ-МАГАЗИНА

Допустим, мы получили заказ от нашего клиента — Издательский Дом МЭИ. Нужно создать интернет-магазин под названием “*MPEI Press*”, который позволяет пользователям просматривать и покупать книги.

Для реализации интернет-магазина используются следующие функциональные возможности:

- база данных книг, которые будут продаваться в интерактивном магазине;
- интерактивный каталог книг с разбивкой по категориям;

- покупательская тележка, позволяющая отслеживать товар, выбираемый пользователем с целью приобретения;
- сценарий расчетов, который обрабатывает элементы платежа и доставки товара;
- интерфейс администрирования.

Существуют два основных представления системы: пользовательское и администраторское (рис. 3 и 4).

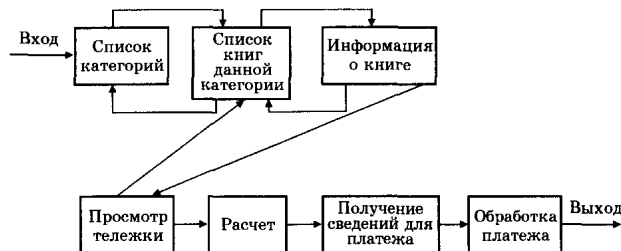


Рис. 3. Пользовательское представление системы

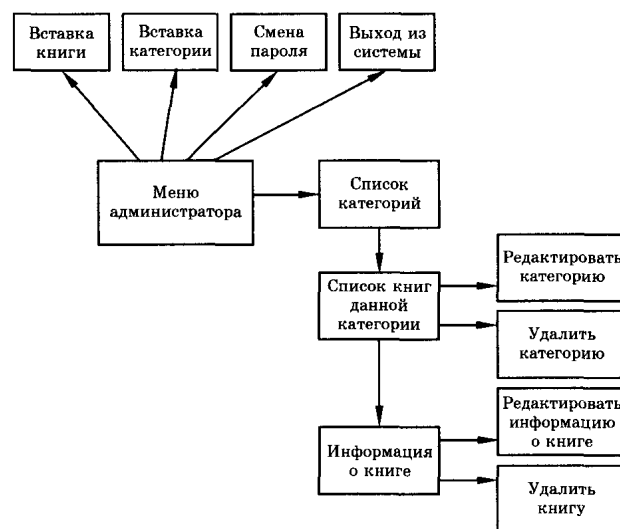


Рис. 4. Администраторское представление системы

В пользовательской части сайта, клиент сначала открывает главную страницу, в которой перечислены все категории книг сайта. Отсюда он может перейти к определенной категории книг, а затем к информации по отдельной книге.

Пользователь может добавить выбранную книгу в корзину. На этапе просмотра тележки он может рассчитаться и покинуть магазин.

В администраторской части сайта администратор может входить в систему и добавлять новые книги и категории, сменять пароль, редактировать/удалять книги и категории.

В данном случае всего 5 основных кодовых модуля:

- база данных;
- интерактивный каталог;
- покупательская корзина;
- обработка заказов;
- интерфейс администрирования.

2.1. Реализация базы данных

SQL-код создания базы данных приведен в файле *book_sc.sql*.

Некоторые исходные (тестовые) товары для интернет-магазина приведены в файле *populate.sql*.

Таблица Customers

CustomerID	Name	Address	City	Tel	Zip	Country
1	Николай Алексеевич	Красноказарменная, 17, 155	Москва	9185606	219299	Россия

Таблица 2

Таблица Orders

Order ID	Customer ID	Amount	Date	Order Status	Ship name	Ship Address	Ship city	Ship tel	Ship zip	Ship country
1	1	49,99	2004-05-04	partial	Николай Алексеевич	Красноказарменная, 17, 155	Москва	9185606	219299	Россия

Таблица 3

Таблица Books

ISBN	Author	Title	CatID	Price	Description
0672317842	Luke Welling and Laura Thomson	PHP and MySQL Web Development	1	49,99	Книга учит читателя создавать динамические, безопасные сети электронной коммерции. Вы научитесь объединять и осуществлять эти технологии благодаря реальным примерам и типовым проектам.

В нашей базе данных всего 6 таблиц:

1. *Customers* (Клиенты)
2. *Orders* (Заказы)
3. *Books* (Книги)
4. *Categories* (Категории)
5. *Order Items* (Заказанные экземпляры)
6. *Admin* (Администратор)

Customers — табл. 1 имеет 7 столбцов (*customerID*, *name*, *address*, *city*, *tel*, *zip*, *country*), каждый из которых содержит данные о клиентах. Столбец *customerID* является первичным ключом данной таблицы.

Orders — табл. 2 состоит из 11 столбцов (*orderID*, *customerID*, *amount*, *date*, *order_status*, *ship_name*, *ship_address*, *ship_city*, *ship_tel*, *ship_zip*, *ship_country*), содержит данные о заказах. Столбец *orderID* является первичным ключом, а *customerID* — вторичным.

Books — табл. 3 состоит из 6 столбцов (*ISBN*, *author*, *title*, *catID*, *price*, *description*), содержит информацию о книгах, которые имеются в наличии интернет-магазина. Столбец *ISBN* является первичным ключом данной таблицы, а *catID* — вторичным.

Categories — табл. 4 имеет 2 столбца (*catID*, *catName*), содержит информацию о виде категории в интернет-магазине. Столбец *catID* является первичным ключом данной таблицы.

Таблица 4

Таблица Categories

CatID	CatName
1	Компьютеры и сети
2	Финансы и экономика

Order Items — табл. 5 имеет 4 столбца (*orderID*, *ISBN*, *item_price*, *quantity*), каждый из которых содержит данные о заказанных товарах. Эта таблица обеспечивает связь между таблицами *Orders* и *Books*.

Таблица 5

Таблица Order Items

OrderID	ISBN	Item price	Quantity
1	0672317842	49,99	1

Admin — табл. 6 имеет 2 столбца (*username*, *password*), содержит данные об администраторе интернет-магазина. Если пользователь ввел имя и пароль, совпадающие с хранящиеся в базе данных именем и паролем, то он получает права администратора.

Таблица 6

Таблица Admin

Username	Password
admin	admin

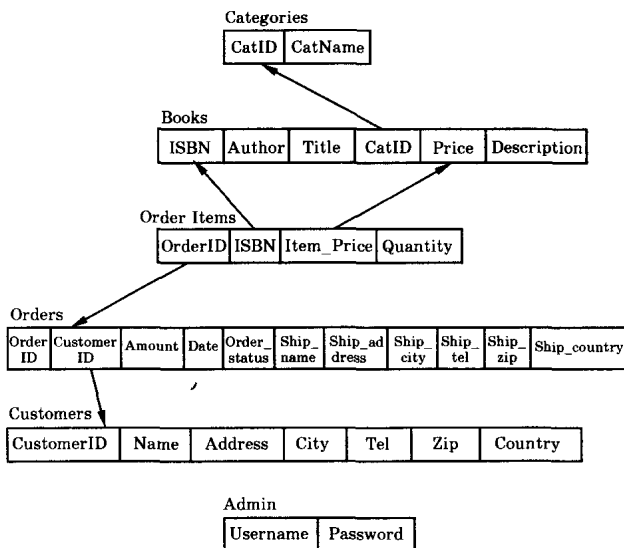


Рис. 5. Схема связи между таблицами

Схема связи между таблицами представлена на рис. 5.

2.2. Реализация интерактивного каталога

Интерактивный каталог содержит три сценария:

- главная страница;
- страница категорий;
- страница информации о книге.

Титульная страница сайта генерируется с помощью сценария *index.php*. Кроме списка категорий окно содержит ссылку на покупательскую тележку (в правом верхнем углу) и итоговые данные по содержимому тележки. Эти элементы имеются на каждой странице, открываемой в процессе просмотра и выбора товара.

Когда пользователь нажимает кнопку одной из категорий, открывается страница категорий, генерируемая сценарием *show_cat.php*.

Все книги категорий “Компьютеры и сети” перечислены в виде ссылок. Когда пользователь нажимает кнопку одной из ссылок, открывается страница информации о соответствующей книге.

На этой странице помимо ссылки для просмотра содержимого тележки “Ваша корзина” находится ссылка “Добавить в корзину”, позволяющая добавлять товар в корзину.

2.2.1. Реализация списка категорий

Первый сценарий, *index.php* выводит все категории из базы данных. Сценарий начинается с подключения файла *book_sc_fns.php*, который содержит все библиотеки функций для данного приложения.

После этого запускается сеанс, который необходим для корректного функционирования покупательской тележки. Сеанс используется каждой страницей сайта.

Сценарий включает вызовы функций вывода *HTML*-содержимого таких, как *do_html_header()* и *do_html_footer()* (обе функции находятся в файле *output_fns.php*).

Также предусмотрен код проверки, что пользователь вошел в систему с правами администратора. В этом случае ему предоставляются другие средства навигации.

Далее показана наиболее важная часть сценария:

```

// Извлечение категории из базы данных
$cat_array = get_categories();
// Отображение ссылок на страницы категорий
display_categories($cat_array);
Функции get_categories() и display_categories()
содержатся в библиотеках book_fns.php и
output_fns.php.
  
```

2.2.2. Реализация списка книг в категории

Вывод списка книг определенной категории осуществляет сценарий *show_cat.php*. Сначала запускается сеанс с помощью функции *session_start()*, а затем с использованием функции *get_category_name()* передаваемый идентификатор категории преобразуется в имя категории:

```
$name = get_category_name($catid);
```

Эта функция выполняет поиск имени категории в базе данных.

Извлечение из базы данных списка книг, относящихся к выбранной категории, осуществляется с помощью функции *get_books()* и *display_books()*:

```
$book_array = get_books($catid);
display_books($book_array);
```

Функция *display_books()* предоставляет ссылку на каждую книгу категории при помощи сценария *show_book.php*.

2.2.3. Отображение информации о книгах

Сценарий *show_book.php* принимает *ISBN* книги в качестве параметра, а затем извлекает и отображает сведения о данной книге из базы данных. Сначала запускается сеанс, а затем с помощью строки:

```
$book=get_book_details($isbn);
```

из базы данных извлекается информация о книге. Для вывода данных в формате *HTML* используется следующий код:

```
display_book_details($book);
```

Кроме того, функция *display_book_details()* выполняет поиск файла изображения для книги, который выглядит как *images/\$isbn.jpg*. Если файл не существует, то изображение не выводится.

2.3. Реализация покупательской тележки

Сценарий *show_cart.php* отображает страницу, которая открывается после нажатия кнопки “Ваша корзина” либо “Добавить в корзину”. Если сценарий *show_cart.php* вызывается без параметров, то отображается содержимое тележки. Если в качестве параметра указывается номер *ISBN*, то соответствующая этому коду книга добавляется в тележку.

На этой странице предоставляются еще две кнопки. Это кнопки “Сохранить изменения” и “Рассчитать”. Первую кнопку можно использовать для изменения количества элементов в тележке. Для этого нужно изменить количество экземпляров в полях “Количество” и нажать кнопку “Сохранить изменения”.

Вторая кнопка выбирается пользователем, когда он хочет произвести расчет и покинуть магазин.

2.3.1. Просмотр содержимого тележки

Код для отображения содержимого тележки находится в файле *show_cart.php*.

Содержимое тележки будет отображаться после нажатия кнопки “Ваша корзина” независимо от страницы, на которой был произведен шелчок. В общем случае после нажатия кнопки “Ваша корзина” выполняется следующая часть кода:

```
if($cart&&array_count_values($cart))
    display_cart($cart);
else
{
    echo "<p>В Вашей корзине нет товаров";
    echo "<hr>";
}
```

Если тележка не пуста, то будет вызываться функция *display_cart()*. Эта функция печатает содержимое тележки в формате HTML. Код этой функции, а также всех функций, имеющих отношение к выводу на экран, находится в файле *output_fns.php*.

Если тележка пуста, то пользователю выводится соответствующее сообщение “В Вашей корзине нет товаров”.

2.3.2. Добавление элементов в тележку

Когда пользователь открывает страницу *show_cart.php* нажатием кнопки “Добавить в корзину”, в тележку помещается выбранный элемент.

Если пользователь еще не помещал в тележку элементы, значит у него нет тележки, поэтому ее требуется создать:

```
if(!session_is_registered ("cart"))
{
    $cart = array ();
    session_register ("cart");
    $items = 0;
    session_register ("items");
    $total_price = "0.00";
    session_register ("total_price");
}
```

Когда известно, что тележка создана, в нее можно добавлять элементы:

```
if($cart [$new])
    $cart [$new]++;
else
    $cart [$new] = 1;
```

Для того чтобы определить общую сумму заказа и количество элементов в тележке, применяются функции *calculate_price()* и *calculate_items()*:

```
$total_price = calculate_price ($cart);
$items = calculate_items ($cart);
```

Эти функции содержатся в файле *book_fns.php*.

2.3.3. Сохранение изменений содержимого тележки

Сохранение изменений количества элементов осуществляется пользователем нажатием кнопки “Сохранить изменения”.

Рассмотрим часть сценария, которая сохраняет изменения:

```
if ($save)
{
    foreach ($cart as $isbn => $qty)
    {
        if ($$isbn == "0")
            unset ($cart [$isbn]);
        else
            $cart [$isbn] = $$isbn;
    }
    $total_price = calculate_price ($cart);
    $items = calculate_items ($cart);
}
```

Перебираются все элементы тележки и для каждой переменной *\$isbn* проверяется значение переменной с соответствующим именем. Для этого используется обозначение *\$\$isbn* (переменная переменной).

Если в любом поле устанавливается значение 0, то соответствующий элемент удаляется из тележки с использованием функции *unset()*. В противном случае содержимое тележки обновляется в соответствии со значениями полей формы.

После обновления содержимого тележки повторно вызываются функции *calculate_price()* и *calculate_items()* для определения новых значений переменных сеанса *\$total_price* и *\$items*.

2.3.4. Вывод итоговых данных в строке заголовка

В строке заголовка каждой страницы сайта выводятся итоговые данные состояния покупательской тележки. Это осуществляется путем печати значений переменных сеанса *\$total_price* и *\$items* функцией *do_html_header()*.

Эти переменные регистрируются, когда пользователь впервые посещает страницу *show_cart.php*. В случае, когда пользователь еще не открывал данную страницу, предусмотрена следующая часть листинга функции *do_html_header()*:

```
if (!$items) $items = "0";
if (!$total_price) $total_price = "0.00";
```

2.4. Реализация платежа

Когда пользователь нажимает кнопку “Рассчитать”, то запускается сценарий *checkout.php*. Страница расчетов будет показана. Данный сценарий требует, чтобы клиент ввел свой адрес (и адрес доставки, если они различаются).

Когда пользователь продолжает сеанс нажатием кнопки “Подтвердить заказ” в нижней части формы, то запускается сценарий *purchase.php*. Вывод сценария будет показан.

Выполняется проверка, что пользователь заполнил форму, и вводится информация в базу данных с помощью функции *insert_order()*. Эта функция вставляет сведения о клиенте в базу данных.

Когда пользователь снова нажмет кнопку “Подтвердить заказ”, то обрабатываются данные платежа с помощью сценария *process.php*. Результаты успешной платежной операции будут показаны. Выводится общая сумма покупки, включая доставку и сообщение об успешной покупке. На этом сеанс уничтожается функцией *session_destroy()*.

2.5. Реализация интерфейса администрирования

Интерфейс администрирования требует, чтобы пользователь вошел в систему через файл *login.php*, который затем выводит меню администратора *admin.php*.

Идентификация пользователя-администратора после входа в систему осуществляется через переменную сеанса *\$admin_user* и функцию *check_admin_user()*. Эта и другие функции, используемые сценариями администрирования, содержатся в библиотеке *admin_fns.php*.

Когда администратору требуется добавить новую категорию или книгу, вызывается сценарий *insert_category_form.php* или *insert_book_form.php*. Каждый сценарий предоставляет администратору форму для заполнения. Формы обрабатываются сценариями *insert_category.php* и *insert_book.php* соответственно.

Помимо добавления новых категорий и книг, пользователь с правами администратора может редактировать и удалять эти элементы, а также сменить пароль. Коды данных функций находятся в файлах *edit_category_form.php*, *edit_category.php*, *edit_book_form.php*, *edit_book.php* и *user_auth_fns.php*.

Администратор может выполнять навигацию по сайту так же, как и обычный пользователь. При этом используются те же сценарии. Такое решение позволяет повторно использовать максимальный объем готового кода.

Однако навигация администратора имеет некоторые отличия. Для него выводятся специальные опции. Администратор имеет возможность редактировать информацию о данной книге или вообще удалить ее.

3. РЕАЛИЗАЦИЯ ПОИСКОВОЙ СИСТЕМЫ

Поиск и вывод требуемой книги выполняет сценарий *search.php* и *results.php*. Поиск нужной книги можно осуществить по номеру *ISBN*, Автору или Названию. После того как пользователь нажмет кнопку "Найти", вызывается сценарий *results.php*. Выводятся все книги, удовлетворяющие заданным критериям. Более подробную информацию о книге можно получить, щелкнув на название книги.

Работа поисковой системы:

1) *Web*-браузер пользователя выдает *HTTP*-запрос определенной *Web*-страницы. Например, пользователь ищет все книги, написанные Мэттом

Зандрста, используя *HTML*-форму. Страница с результатом поиска будет называться *results.php*;

2) *Web*-сервер принимает запрос на *results.php*, извлекает этот файл и передает на обработку механизму *PHP*;

3) механизм *PHP* приступает к разбору сценария. Сценарий содержит команду соединения с базой данных и выполнения запроса (поиска книг). *PHP* открывает соединение с *MySQL*-сервером и отправляет соответствующий запрос;

4) сервер *MySQL* принимает запрос к базе данных, обрабатывает его и отправляет результат (список книг) обратно механизму *PHP*;

5) механизм *PHP* завершает выполнение сценария. После этого результат в виде *HTML* возвращается *Web*-серверу;

6) *Web*-сервер пересылает *HTML* в браузер, и пользователь имеет возможность просмотреть запрошенный список книг.

ЗАКЛЮЧЕНИЕ

Результатом проделанной работы является довольно простой, но полностью работающий интернет-магазин, позволяющий человеку, не имеющему навыков в программировании, управлять интернет-магазином. Он может самостоятельно добавлять, удалять, редактировать содержимое магазина. В дальнейшем возможны многочисленные расширения и усовершенствования проекта.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Каба М. *MySQL и Perl: коммерческие приложения для Интернета/ Учебный курс.*— СПб.: Питер, 2001.— 288 с.
2. Ролланд Ф. Д. *Основные концепции баз данных:* Пер. с англ.— М.: Издательский дом "Вильямс", 2002.— 256 с.
3. Сайт *MySQL* в Интернете [<http://www.mysql.com/> (31.05.2004)]
4. Алексеев Ю. М. *Быстро и легко создаем, программируем, шлифуем и раскручиваем web-сайт/ Учеб. пособ.*— М.: Лучшие книги, 2003.— 432 с.
5. Аргерих Л. и др. *Профессиональное PHP программирование.*— 2-е изд.: Пер. с англ.— СПб: Символ-Плюс, 2003.— 1048 с.
6. Сайт *PHP* в Интернете [<http://www.php.net/> (31.05.2004)]

Материал поступил в редакцию 08.12.05.