МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«Южно-Уральский государственный университет»
(национальный исследовательский университет)
Высшая школа электроники и компьютерных наук
Кафедра системного программирования

РАБОТА ПРОВЕРЕНА

Рецензент
Директор ООО «Точные решения»
_____ В.Н. Мокроносов
“ ”
_____2017 г.

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой,
д.ф.-м. н., профессор
_____ Л.Б. Соколинский
“ ”
_____ 2017 г.

РАЗРАБОТКА ИНТЕРНЕТ-МАГАЗИНА SPORTDRIVE

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
ЮУрГУ – 02.03.02.2017.308-067.ВКР

Научный руководитель,
Ассистент кафедры СП
_____ Д.М. Баскаков

Автор работы,
студент группы КЭ-402
_____ Р.С. Елисеев

Ученый секретарь
(нормоконтролер)
_____ О.Н. Иванова
“ ”
_____2017 г.

Челябинск-2017

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ 2
1. ТЕОРЕТИЧЕСКОЕ ОБОСНОВАНИЕ 4
1.1. Понятие «Интернет-магазин» 4
1.2. Платформы для построения веб-интерфейса 5
2. ПРОЕКТИРОВАНИЕ 7
2.1. Функциональные требования 7
2.2. Проектирование web-приложения 7
2.3. Варианты использования системы 9
2.4. Проектирование базы данных 11
2.5. Диаграмма компонентов 13
3. РЕАЛИЗАЦИЯ 14
3.1. Инструменты, используемые при реализации 14
3.2. Взаимодействие компонентов 14
3.3. Пользовательский интерфейс 18
4. ТЕСТИРОВАНИЕ 23
4.1. Функциональное тестирование 23
4.2. Юзабилити-тестирование 26
ЗАКЛЮЧЕНИЕ 27
ЛИТЕРАТУРА 28
ПРИЛОЖЕНИЯ 30
Приложение 1 30
Приложение 2 36

ВВЕДЕНИЕ

В современном информационном обществе каждая стабильная компания должна иметь собственный представительский сайт в сети Интернет, который обеспечит информационную поддержку существующего бизнеса [5]. С помощью web-сайта фирмы решают такие задачи, как представление компании в сети Интернет, расширение потенциальной аудитории потребителей, поддержка бренда, повышение узнаваемости, информирование общественности и др.

Разработка сайтов для компаний является актуальной и востребованной сферой деятельности, т.к. сайт фирмы в сети Интернет представляет собой достаточно дешевый и массовый способ рекламы, дает возможность потенциальным и существующим клиентам легко получать информацию о товарах и услугах компании, ее деловых интересах, что может помочь найти новых заказчиков и партнеров по бизнесу, а, следовательно, способствует увеличению объема продаж и рентабельности предприятия.

Объектом исследования данного дипломного проекта является процесс электронной торговли. Магазин занимается продажей продуктов спортивного питания. На данный момент он является наиболее «молодым» образцом организации интернет-магазина, поэтому наибольшее внимание уделяет созданию простого удобного пользовательского интерфейса интернет-магазина [9].

Целью работы является разработка интернет-магазина «SportDrive». Для достижения этой цели были определены следующие цели и задачи:

- выполнить анализ предметной области;
- разработать бриф на проектирование интернет магазина;
- реализовать интернет-магазин спортивного питания «SportDrive»;
- провести тестирование интернет-магазина.

Структура и объем работы

Работа состоит из введения, 4 разделов, заключения, списка библиографии и двух приложений.

Объем работы составляет 29 страниц, объем библиографии – 15 источников. Объем приложений составляет 7 страниц.

Краткое содержание работы

В первой главе рассмотрено понятие «Интернет-магазин», проведен обзор программных средств для разработки web-приложений.

Во второй главе представлены анализ и описание требований к разрабатываемому web-приложению, а также процесс проектирования архитектуры.

Третья глава содержит реализацию web-приложения. В этом разделе подробно рассмотрены интерфейсы компонентов и детали их реализации.

Четвертая глава посвящена результатам тестирования разработанного интернет-магазина «SportDrive».

В заключении сделаны выводе о проделанной работе.

В приложении 1 представлены скриншоты интерфейса системы.

В приложении 2 представлен отчет о юзабилити-тестировании интернет-магазина.

1. ТЕОРЕТИЧЕСКОЕ ОБОСНОВАНИЕ

В данном разделе рассмотрено понятие «Интернет-магазин», а также проведен обзор программных средств для разработки web-приложений.

1.1. Понятие «Интернет-магазин»

Через сеть Интернет покупатель при помощи браузера заходит на веб-сайт интернет-магазина. Веб-сайт содержит электронную витрину, на которой представлены каталог товаров и необходимые интерфейсные элементы для ввода регистрационной информации, формирования заказа, оформления доставки и платежа, получения информации о компании-продавце [6].

По сути одиночное посещения сайта, это точка входа в объектную модель, через которую открывается доступ к другим объектам модели.

Типовая модель интернет-магазина состоит из следующих функциональных частей:

- каталог товаров;
- пользовательская корзина;
- регистрационная форма;
- форма отправки заказа.

Каталог товаров представляет собой структуру данных, которая должна простым и понятным способом производить упорядочивание товаров. Проще всего такой каталог представить в виде дерева объектов, верхний уровень которого состоит из списка разделов. Разделы содержат ссылки на конкретный товар. Такое упорядочивание необходимо для удобного и быстрого поиска и заказа товаров.

Пользовательская корзина представляет собой некоторый массив данных, который служит для хранения заказанного пользователем товара.

Регистрационная форма служит для ввода персональных данных пользователей. В дальнейшем эта информация используется для их идентификации между сеансами работы с интернет-магазином.

Форма отправки заказа служит для ввода контактной информации заказчика и отправки ее на электронный ящик организации.

Регистрация покупателя производится или при оформлении заказа, или при входе в магазин. После выбора товара от покупателя требуется заполнить форму, в которой указывается, каким образом будет осуществлена оплата и доставка. По окончании формирования заказа и регистрации вся собранная информация о покупателе поступает из электронной витрины в торговую систему интернет-магазина и передается на исполнение поставщику.

В том случае, если оплата осуществляется при передаче товара покупателю (курьером или наложенным платежом), необходимо подтверждение факта заказа. Чаще всего, это происходит посредством электронной почты или по телефону [1].

При возможности оплаты через Интернет, подключается платежная система. После сообщения о проведении онлайн-платежа торговой системой формируется заказ для службы доставки.

1.2. Платформы для построения веб-интерфейса

Денвер (от сокр. Д.н.в.р – джентльменский набор веб-разработчика) – набор дистрибутивов и программная оболочка, предназначенные для создания и отладки сайтов (веб-приложений, прочего динамического содержимого интернет страниц) на локальном ПК (без необходимости подключения к сети Интернет) под управлением ОС Windows [4].

Apache – веб-сервер, работающий на локальном компьютере, на котором может работать неограниченное количество сайтов, что очень эффективно для разработки и отладки сценариев PHP без загрузки его файлов на удаленный сервер.

PHP 5 – скриптовый язык программирования общего назначения, интенсивно применяющийся для разработки веб-приложений. В настоящее время поддерживается подавляющим большинством хостинг- провайдеров

и является одним из лидеров среди языков программирования, применяющихся для создания динамических веб-сайтов с полным набором модулей с поддержкой MySQL [11].

MySQL – свободная система управления базами данных (СУБД). MySQL является собственностью компании Oracle Corporation, получившей ее вместе с Sun Microsystems, осуществляющей разработку и поддержку приложения.

phpMyAdmin – веб-приложение с открытым кодом, написанное на языке PHP и представляющее собой веб-интерфейс для администрирования СУБД MySQL. phpMyAdmin позволяет через браузер осуществлять администрирование сервера MySQL, запускать команды SQL и просматривать содержимое таблиц и баз данных.

Для создания интернет-магазина требовалась свободно распространяемая СУБД и скриптовый язык программирования. В качестве среды для разработки веб-приложения был выбран свободно распространяемый и популярный программный продукт «Денвер». В состав программного пакета входят необходимые инструменты: Apache Server, PHP, MySQL, phpMyAdmin [12, 14].

2. ПРОЕКТИРОВАНИЕ

2.1. Функциональные требования

Функциональные требования – это требования к поведению системы, они определяют функциональность разрабатываемого программного обеспечения, т.е. описывают возможности, которые предоставляет система [15]. Функциональные требования определяют «как» необходимо реализовать проект, включают в себя бизнес-требования и пользовательские требования.

Были сформулированы основные функциональные и нефункциональные требования к web-приложению, определены главные пользователи системы и варианты использования системы пользователями.

На этапе проектирования были определены следующие функциональные требования:

1) возможность регистрации покупателя;
2) предоставление возможности покупателю добавления необходимого количества товаров при помощи корзины;
3) предоставление возможности покупателю добавлять товары в корзину и удалять товары из корзины;
4) предоставление возможности администратору просмотра информации по всем заявкам, товарам и категориям товаров;
5) предоставление возможности администратору удалять, добавлять и редактировать товары и их категории;
6) структурировать товары при помощи категорий;
7) реализовать функцию поиска товаров по наименованию.

2.2. Проектирование web-приложения

Структура приложения основывается на шаблоне проектирования MVC («Модель-Представление-Контроллер»).

MVC – схема разделения данных в приложении, интерфейса и логики на отдельные компоненты: модель, представление и котроллер – таким

образом, чтоб изменение каждого из компонентов не оказывало никакого влияния на остальные [13].

Модель – компонент, предоставляющий данные контроллеру. Модель полностью независима от остальных компонентов, не знает о существовании контроллеров и представлений. В интернет-магазине моделями являются категория и товар.

Представление – компонент, в обязанности которого входит отображение данных, которые передает контроллер. Представление никак не влияет на модель или контроллер.

Контроллер – компонент, который содержит функциональную бизнес-логику, обрабатывает действия пользователя и оповещает модель о необходимости внести изменения. В зависимости от результата обработки отправляет результат пользователю в виде представления.

Общая схема взаимодействия данных компонентов представлена на рис. 1.

Рис. 1. Взаимодействие компонентов в MVC на примере Web-приложения

2.3. Варианты использования системы

Диаграмма вариантов использования – это диаграмма, отражающая отношения между актёрами и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне [2]. Прецедент описывает поведение, демонстрируемое системой с целью получения значимого результата для одного или более актеров.

Прецедент – возможность моделируемой системы (часть её функциональности), благодаря которой пользователь может получить конкретный, измеримый и нужный ему результат. Прецедент соответствует отдельному функционалу системы, определяет один из вариантов её использования и описывает типичный способ взаимодействия пользователя с системой.

В рамках разрабатываемой системы предусмотрено два актера – пользователь и администратор. Интерфейсы актеров отличаются функциональностью. Диаграмма вариантов использования представлена на рис. 2.

Для покупателя представлены следующие варианты использования:

1) прецедент *Зарегистрироваться* – заполнить предлагаемые личные данные, задать логин и пароль;

2) прецедент *Авторизоваться* – вход в систему покупателем;

3) прецедент *Добавить товар в корзину* – добавление покупателем товара в корзину;

4) прецедент *Удалить товар из корзины* – убрать товар из корзины;

5) прецедент *Отправить заявку* – отправление заявки покупателем администратору для дальнейшего согласования условий покупки;

6) прецедент *Найти товар через каталог* – нахождение товара покупателем с использованием каталога;

7) прецедент *Найти товар с помощью поисковой строки* – нахождение товара покупателем с использованием поисковой строки;

8) прецедент *Просмотреть дополнительную информацию о товаре* – просмотр пользователем всей информации о товаре.

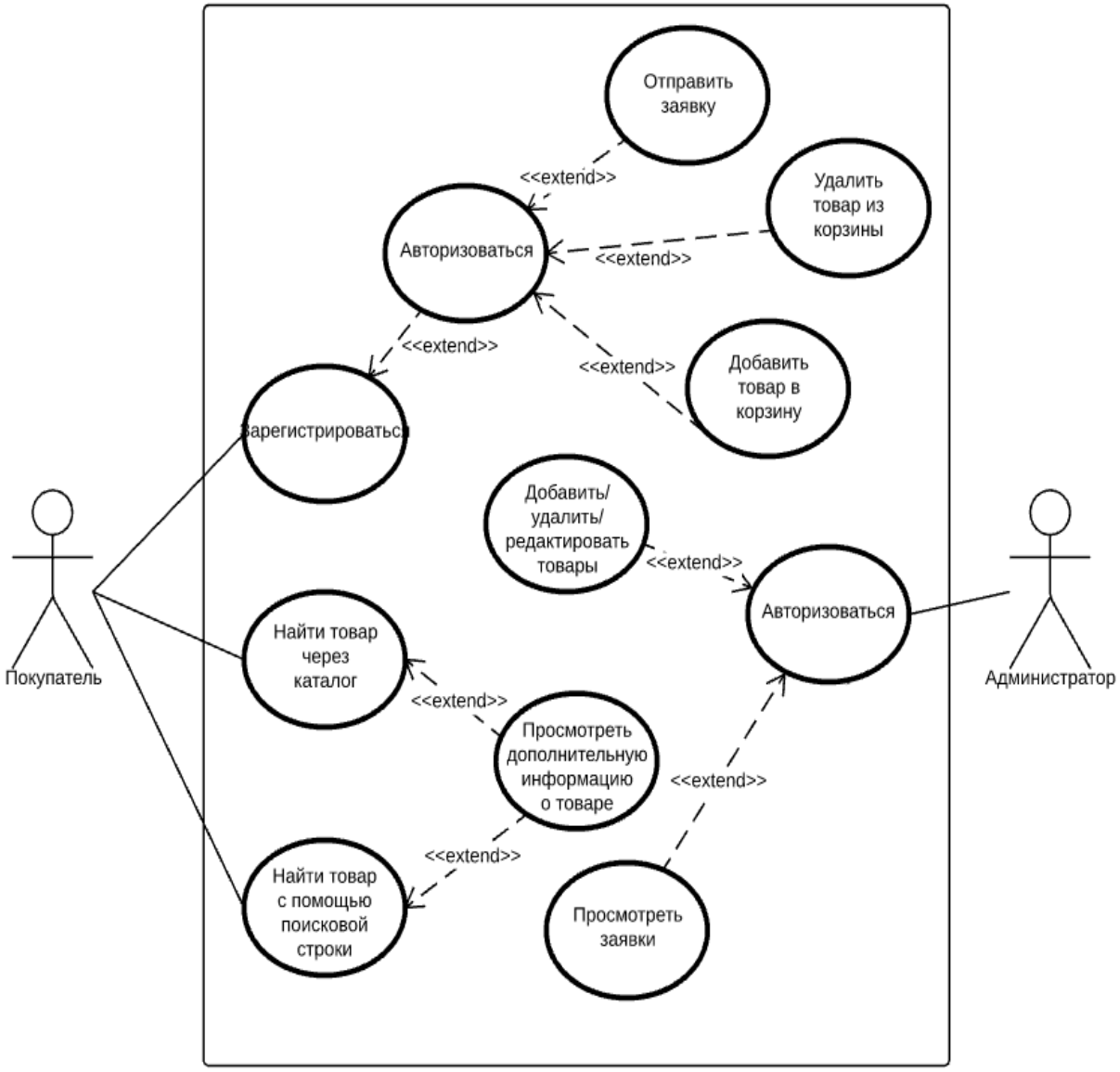


Рис. 2. Диаграмма вариантов использования

Прецеденты администратора работают из специальной панели сайта (панель администратора):

1) прецедент *Авторизоваться* – авторизация админстратора на административном сайте;

2) прецедент *Добавить/удалить/редактировать товары* – управление товарами администратором;

3) прецедент *Просмотреть заявки* – просмотр заявок на покупку товаров.

2.4. Проектирование базы данных

В табл. 1-4 представлено описание базы данных интернет-магазина под управлением СУБД MySQL [7].

Табл. 1. Таблица «Product»

№	Атрибут	Семантика	Тип
1	id	Уникальный идентификатор	int(11)
2	name	Наименование товара	varchar(255)
3	category_id	Принадлежность категории	int(11)
4	code	Код товара	int(11)
5	price	Стоимость товара	float
6	availability	Есть ли товар в наличии	int(11)
7	brand	Производитель товара	varchar(255)
8	description	Описание товара	text
9	is_new	Является ли товар новинкой	int(11)
10	is_recommended	Рекомендованный товар	int(11)
11	status	Виден ли товар в каталоге	int(11)

В таблице «Product» хранится информация о продаваемых товарах.

Табл. 2. Таблица «Category»

№	Атрибут	Семантика	Тип
1	id	Уникальный идентификатор	int(11)
2	name	Наименование категории	varchar(255)
3	sort_order	Порядок сортировки в меню	int(11)
4	status	Видна ли категория в каталоге	int(11)

В таблице «Category» хранится информация о категориях товарах.

Табл. 3. Таблица «Users»

№	Атрибут	Семантика	Тип
1	id	Уникальный идентификатор	int(11)
2	name	Логин пользователя	varchar(255)
3	email	Электронная почта пользователя	varchar(255)
4	password	Пароль для входа пользователя	varchar(255)
5	role	Имеет ли пользователь доступ к панели администратора	int(11)

В таблице «Users» хранится личная информация о зарегистрированных пользователях интернет-магазина.

Табл. 4. Таблица «Product_order»

№	Атрибут	Семантика	Тип
1	id	Уникальный идентификатор	int(11)
2	user_name	ФИО покупателя	varchar(255)
3	user_phone	Контактный телефон покупателя	varchar(255)
4	user_comment	Комментарий покупателя	text
5	user_id	Уникальный идентификатор покупателя в системе	int(11)
6	date	Дата заказа	timestamp
7	products	Выбранные товары	text
8	status	Активность заказа	int(11)

В таблице «Product_order» хранится информация о заказах. Атрибут status может принимать значения от 0 до 3 в зависимости от четырех состояний заказа («новый заказ», «в обработке», «доставляется», «закрыт).

2.5. Диаграмма компонентов

Веб-приложение построено на трех основных компонентах:

– model;

– view;

– controller.

Компонент *model* представлен в виде ряда артефактов – файлов с расширением *.php, хранящих данные о таких объектах, как: «category.php», «product.php», «order.php», «user.php».

Компонент *view* содержит файлы, в которых описана графическая визуализация страниц.

Компонент *controller* содержит файлы с описанием поведения объектов в зависимости от запросов пользователя.

Диаграмма компонентов представлена на рис. 3.

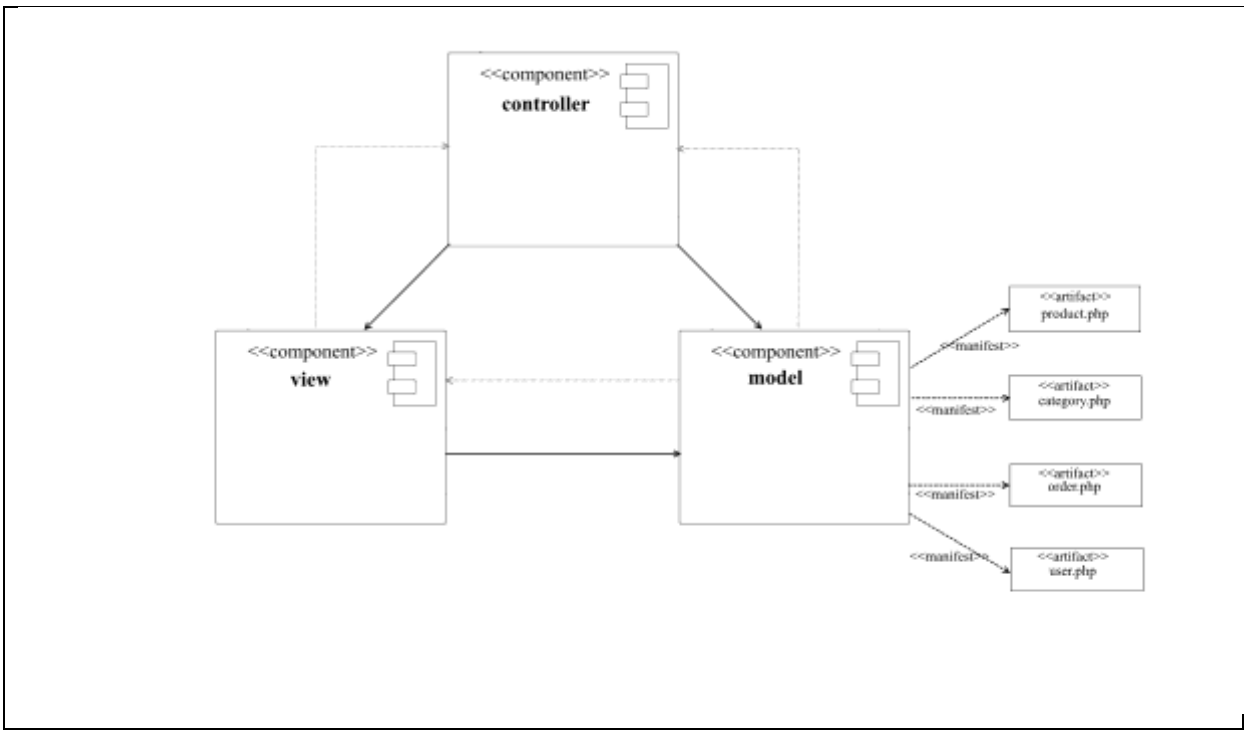


Рис. 3. Диаграмма компонентов

3. РЕАЛИЗАЦИЯ

3.1. Инструменты, используемые при реализации

В качестве среды для разработки интернет-магазина были выбраны следующие программные продукты: KompoZer, Денвер. В качестве языка программирования был выбран PHP 5.

KompoZer – визуальный HTML-редактор. В отличие от многих других комбинированных и визуальных редакторов, страницы, созданные в KompoZer, правильно отражаются в Интернете, т.е. действительно соблюдается принцип "Что вижу - то и получаю" (при условии, что браузеры способны в принципе поддерживать то, что создано в редакторе) [8].

3.2. Взаимодействие компонентов

При открытии интернет-магазина по определенному адресу пользователем его запрос изначально перенаправляется на SiteController. Он отвечает за такие действия, как подключение всех файлов разработанной системы, установка соединения с базой данных. Далее при помощи класса Router производится анализ запроса и определение контроллера (controller), который обработает запрос, после чего вызывается соответствующий метод в классе контроллера исходя из специальных правил (маршрутов). На рис. 3 представлена часть кода из класса Router, отвечающая за обработку строки запроса.

Метод из класса контроллера обращается к необходимой модели (model) и получает из нее данные [3]. Полученные данные поступают к представлению (view), которое отвечает за визуализацию страниц сайта. Интернет-магазин основан на четырех моделях: *пользователь*, *товар*, *категория товара* и *заказы*.

Функция *getLatestProducts* осуществляет запрос к базе данных, в результате которого производится дальнейшая обработка полученных в виде массива данных по товарам (рис. 4).

```php
public function run()
{
    $uri = $this->getURI();

    foreach ($this->routes as $uriPattern => $path) {

        if (preg_match("~$uriPattern~", $uri)) {
            $internalRoute = preg_replace("~$uriPattern~", $path, $uri);
            $segments = explode('/', $internalRoute);
            $controllerName = array_shift($segments) . 'Controller';
            $controllerName = ucfirst($controllerName);
            $actionName = 'action' . ucfirst(array_shift($segments));
            $parameters = $segments;
            $controllerFile = ROOT . '/controllers/' .
                    $controllerName . '.php';
            if (file_exists($controllerFile)) {
                include_once($controllerFile);
            }
            $controllerObject = new $controllerName;
            $result = call_user_func_array(array($controllerObject, $actionName), $parameters);
            if ($result != null) {
                break;
            }
        }
    }
}
```

Рис. 4. Обработка строки запроса при помощи класса Router

В функции *updateCategoryById* представлена реализация редактирования категорий в панели администратора (рис. 5).

```php
// Количество отображаемых товаров по умолчанию
const SHOW_BY_DEFAULT = 6;

public static function getLatestProducts($count = self::SHOW_BY_DEFAULT)
{
    $db = Db::getConnection();

    $sql = 'SELECT id, name, price, is_new FROM product '
            . 'WHERE status = "1" ORDER BY id DESC '
            . 'LIMIT :count';

    $result = $db->prepare($sql);
    $result->bindParam(':count', $count, PDO::PARAM_INT);

    $result->setFetchMode(PDO::FETCH_ASSOC);

    $result->execute();

    $i = 0;
    $productsList = array();
    while ($row = $result->fetch()) {
        $productsList[$i]['id'] = $row['id'];
        $productsList[$i]['name'] = $row['name'];
        $productsList[$i]['price'] = $row['price'];
        $productsList[$i]['is_new'] = $row['is_new'];
        $i++;
    }
    return $productsList;
}
```

Рис. 5. Часть кода реализации модели товара

```
public static function updateCategoryById($id, $name, $sortOrder, $status)
{
    $db = Db::getConnection();

    $sql = "UPDATE category
        SET
            name = :name,
            sort_order = :sort_order,
            status = :status
        WHERE id = :id";

    $result = $db->prepare($sql);
    $result->bindParam(':id', $id, PDO::PARAM_INT);
    $result->bindParam(':name', $name, PDO::PARAM_STR);
    $result->bindParam(':sort_order', $sortOrder, PDO::PARAM_INT);
    $result->bindParam(':status', $status, PDO::PARAM_INT);
    return $result->execute();
}
```

Рис. 6. Часть кода реализации модели категории

Действия с категориями и товарами осуществляются с помощью соответствующих моделей, а затем результат передается контроллеру каталога товаров CatalogController.php. Ниже представлена реализация данного контроллера (рис. 6).

Все классы разделены таким образом, что модификация одного из них не окажет влияния на остальные, т.е. модификация каждого класса может осуществляться независимо.

```
class CatalogController
{
    public function actionIndex()
    {
        $categories = Category::getCategoriesList();

        $latestProducts = Product::getLatestProducts(12);

        require_once(ROOT . '/views/catalog/index.php');
        return true;
    }

    public function actionCategory($categoryId, $page = 1)
    {
        $categories = Category::getCategoriesList();

        $categoryProducts = Product::getProductsListByCategory($categoryId, $page);

        $total = Product::getTotalProductsInCategory($categoryId);

        $pagination = new Pagination($total, $page, Product::SHOW_BY_DEFAULT, 'page-');

        require_once(ROOT . '/views/catalog/category.php');
        return true;
    }

}
```

Рис. 7. Реализация контроллера каталога товаров

Работа с заказами происходит с помощью модели Order. На рис. 8 представлена часть кода реализации данной модели, функция *save*, в которой выполняется сохранение заказа.

```
public static function save($userName, $userPhone, $userComment, $userId, $products)
{
    $db = Db::getConnection();

    $sql = 'INSERT INTO product_order (user_name, user_phone, user_comment, user_id, products)
            . 'VALUES (:user_name, :user_phone, :user_comment, :user_id, :products)';

    $products = json_encode($products);

    $result = $db->prepare($sql);
    $result->bindParam(':user_name', $userName, PDO::PARAM_STR);
    $result->bindParam(':user_phone', $userPhone, PDO::PARAM_STR);
    $result->bindParam(':user_comment', $userComment, PDO::PARAM_STR);
    $result->bindParam(':user_id', $userId, PDO::PARAM_STR);
    $result->bindParam(':products', $products, PDO::PARAM_STR);

    return $result->execute();
}
```

Рис. 8. Часть кода реализации модели заказов

Модель User хранит в себе информацию о зарегистрированных в системе покупателях. Выполняется проверка на авторизацию посетителя сайта, в случае ее отсутствия посетителю предлагается пройти регистрацию. При регистрации пользователя выполняются проверки на количество символов в логине и пароле, занятость указанного e-mail адреса другим пользователем.

Ниже представлена часть кода реализации модели User (рис. 9).

```
//Вернуть ID пользователя, если сессия активна
    public static function checkLogged()
    {
        if (isset($_SESSION['user'])) {
            return $_SESSION['user'];
        }

        header("Location: /user/login");
    }

//Проверка на авторизацию
    public static function isGuest()
    {
        if (isset($_SESSION['user'])) {
            return false;
        }
        return true;
    }
```

Рис. 9. Реализация авторизации в модели пользователя

3.3. Пользовательский интерфейс

Главная страница

Главная страница сайта отображает последние добавленные товары (рис. 10). На страницу конкретного продукта можно перейти, щелкнув по его названию. Можно просмотреть товары определенной категории, выбрав их в соответствующем меню в левой части. В футере сайта отображены рекомендуемые товары, оформленные в виде слайдера (является ли товар рекомендованным – можно задать при добавлении товара через панель администратора). В головной части сайта присутствует поиск по каталогу товаров, по нажатию кнопки «Найти» производится перенаправление на страницу с результатами поиска. Также присутствуют элементы управления авторизацией пользователя и корзиной (доступно только для авторизированных пользователей).

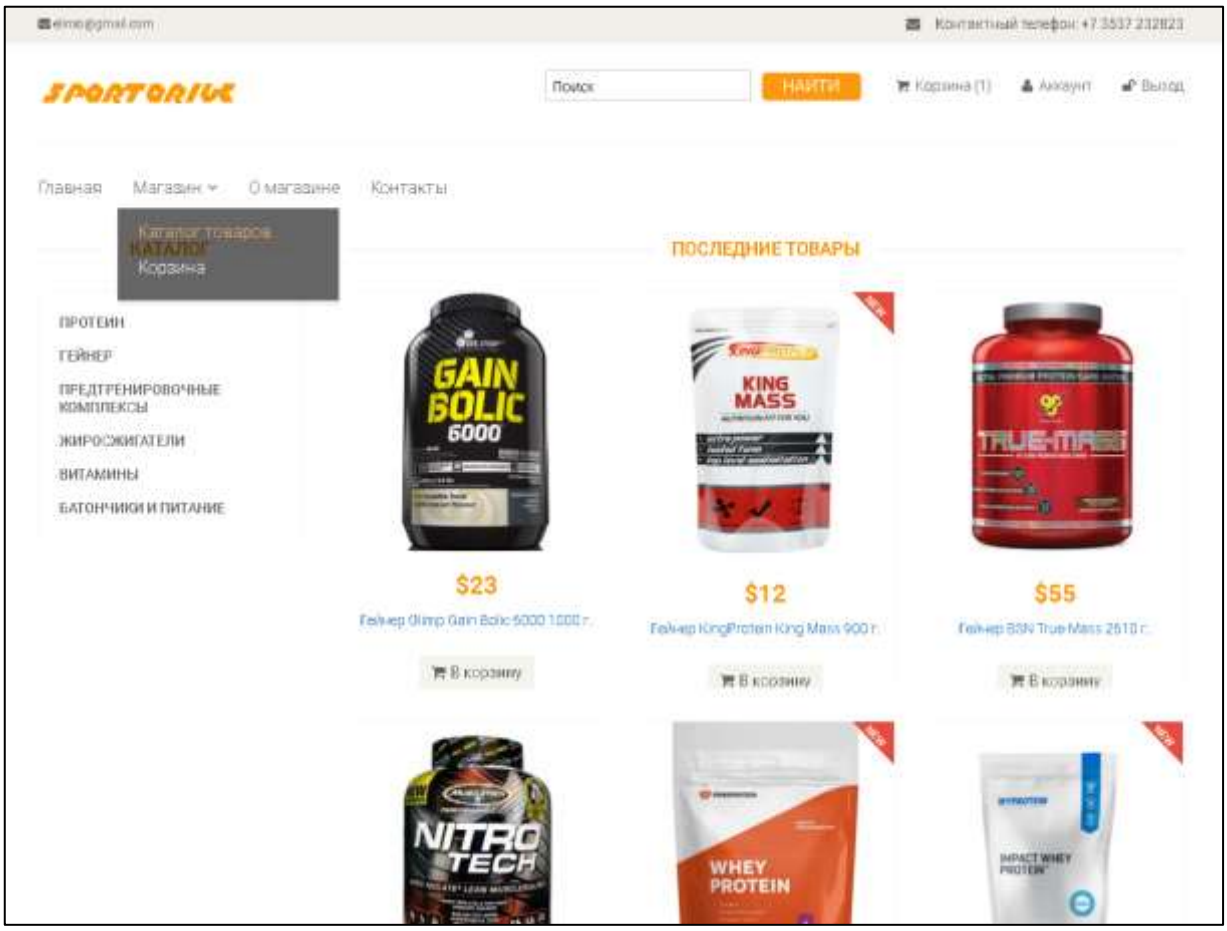


Рис. 10. Вид главной страницы интернет-магазина

Страница категории товара

Отображает все товары в данной категории, по несколько продуктов на страницу, навигация по которым происходит при помощи переключателей (рис. 11). Активная категория подсвечена в меню категорий.

Рис. 11. Страница категории товара

Страница товара

Отображает информацию о выбранном продукте. Содержит название, код товара, его стоимость и наличие в магазине, а также текстовое описание (рис. 12).

Результаты поискового запроса

Отображает информацию с результатами поиска по каталогу товаров (рис. 13). Поиск производится по наименованию. Указано количество найденных совпадений. Переход на страницу товара осуществляется так же, как и в других разделах каталога.

Панель администратора

Позволяет управлять категориями и товарами: их можно добавить, отредактировать и удалить (рис.14-16).

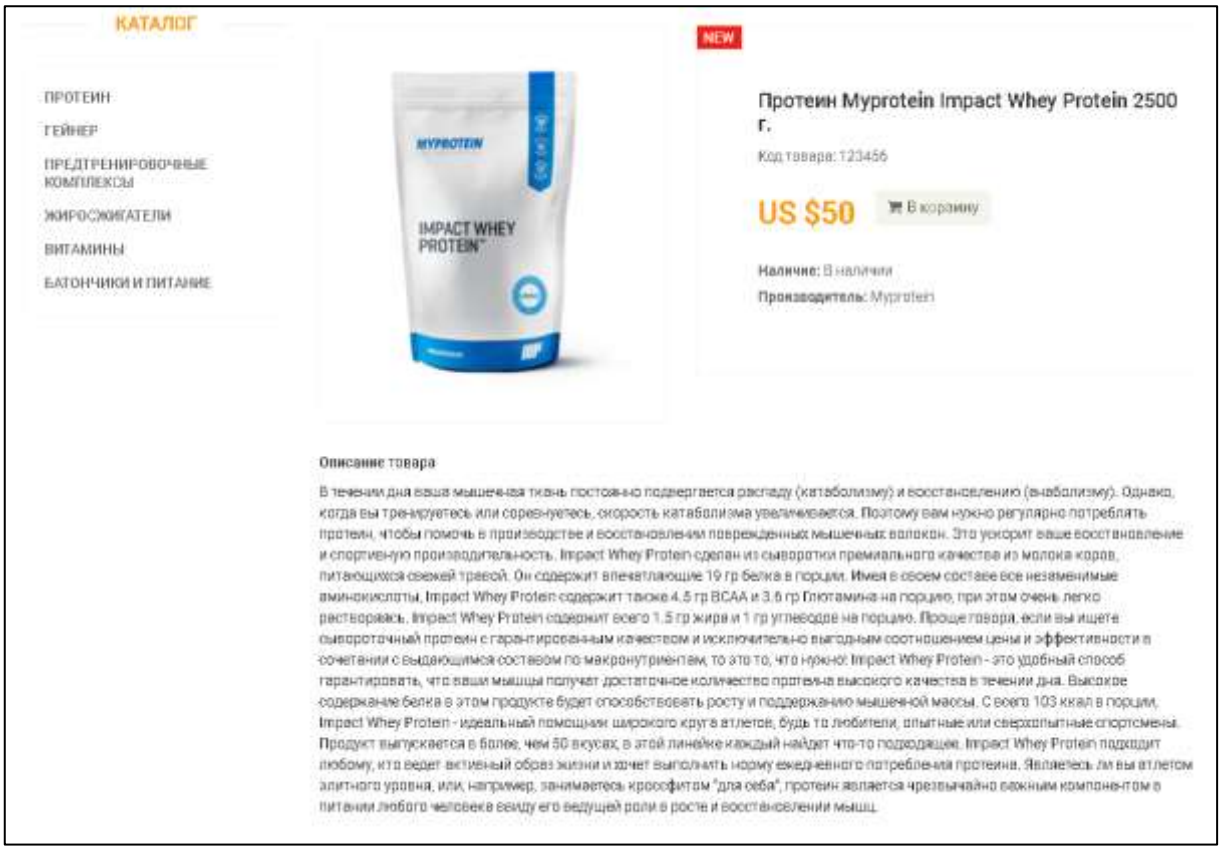


Рис. 12. Страница с полной информацией о товаре

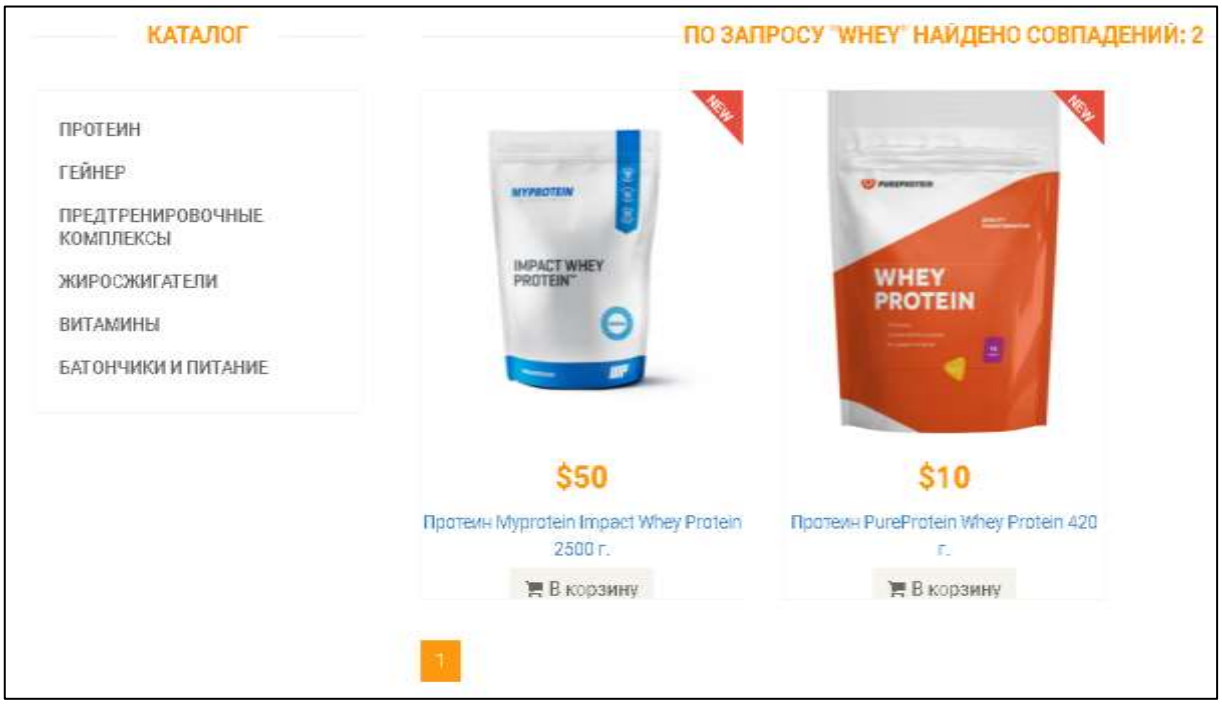


Рис. 13. Страница с результатами поискового запроса

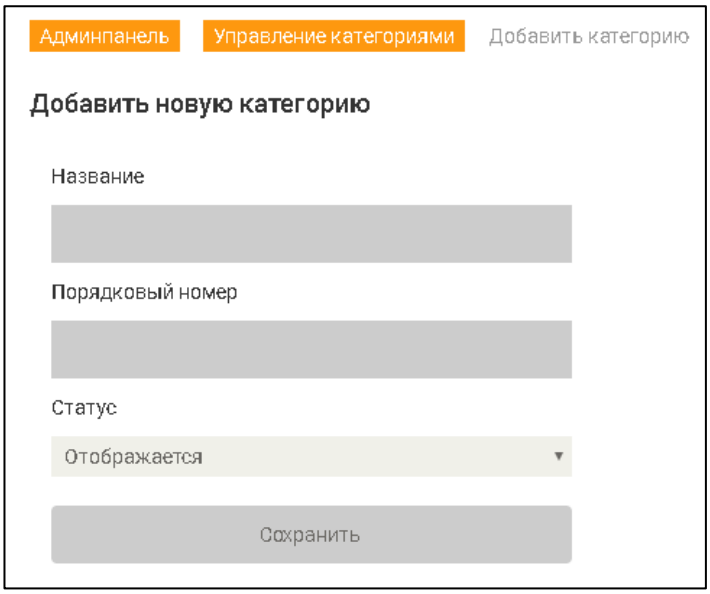


Рис. 14. Добавление категории в панели администратора

Рис. 15. Управление товарами в панели администратора

Вход в панель администратора доступен только пользователям, имеющим значение role=1, role=0 определен для покупателей магазина.

Также в панели администратора можно просмотреть информацию о заказах и установить им статус («новый заказ», «в обработке», «доставляется», «закрыт).

Остальные скриншоты системы представлены в приложении 1.

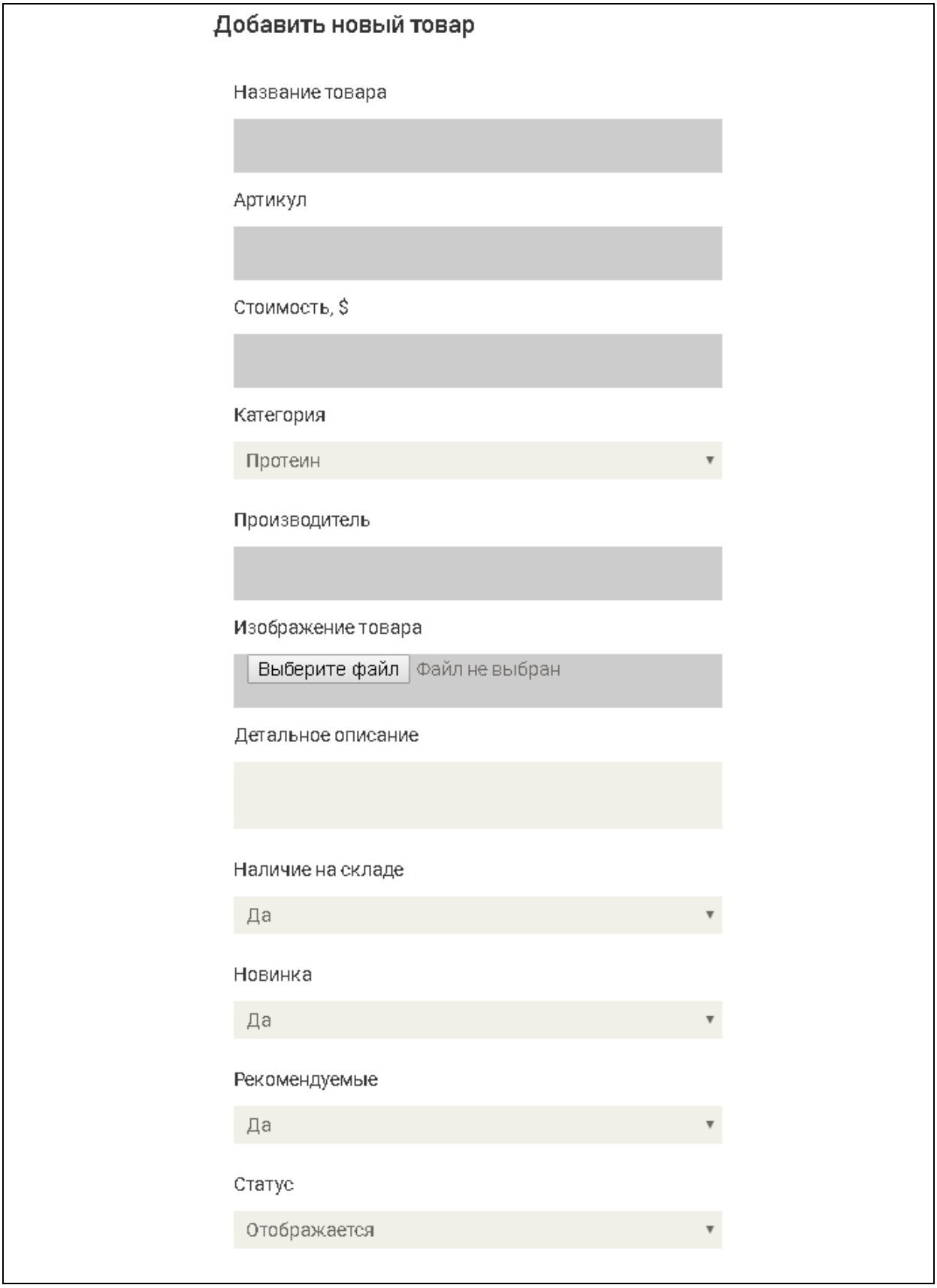


Рис. 16. Добавление товара в панели администратора

4. ТЕСТИРОВАНИЕ

Тестирование программного обеспечения – проверка соответствия между реальным и ожидаемым поведением программы, осуществляемая на конечном наборе тестов, выбранном определенным образом. В более широком смысле, тестирование – это одна из техник контроля качества, включающая в себя активности по планированию работ, проектированию тестов, выполнению тестирования и анализу полученных результатов.

При тестировании web-приложения использовались следующие типы тестирования:

1) функциональное тестирование;
2) юзабилити-тестирование.

4.1. Функциональное тестирование

Функциональное тестирование – это тестирование программного обеспечения в целях проверки реализуемости функциональных требований, то есть способности программного обеспечения в определенных условиях решать задачи, нужные пользователям. Функциональные требования определяют, что именно делает программное обеспечение, какие задачи оно решает. Результаты тестирования представлены в табл. 5.

Табл. 5. Функциональное тестирование

№	Название теста	Ожидаемый результат	Полученный результат	Тест пройден?
1	Переход по ссылкам меню	Пользователь должен без затруднений перемещаться по каталогу товаров, а также должен найти контакты магазина	Пользователь без затруднений смог найти каталог товаров, информацию о товаре, а также материал «Контакты»	Да

Продолжение Табл. 5

№	Название теста	Ожидаемый результат	Полученный результат	Тест пройден?
2	Тестирование поисковой системы на странице «Поиск»	Пользователь вводит запрос «Протеин», который является частью названия нескольких товаров, на странице поиска отобразятся эти товары	На странице поиска отобразились искомые товары	Да
3	Тестирование формы обратной связи на странице «Контакты»	Отправленное через форму сообщение должно поступить менеджеру магазина	Полученный результат совпал с ожидаемым	Да
4	Тестирование регистрации и авторизации пользователя	После успешной регистрации и авторизации пользователь должен находиться в своем профиле, система выдает приветственное сообщение	Регистрация пройдена, авторизация успешна, система выдала приветственное сообщение	Да

Окончание Табл. 5

№	Название теста	Ожидаемый результат	Полученный результат	Тест пройден?
5	Добавление товара в корзину	У авторизованного пользователя выбранный товар должен добавиться в корзину, о чем свидетельствует изменившийся счетчик корзины пользователя	Полученный результат совпал с ожидаемым. Неавторизованному пользователю система при добавлении товара в корзину предлагает авторизоваться, либо зарегистрироваться	Да
6	Оформление заказа	У авторизованного пользователя при нажатии кнопки «Оформить заказ» в корзине, если в ней присутствует хотя бы один товар, заказ должен поступить в обработку, о чем пользователя уведомляет система	Полученный результат совпал с ожидаемым	Да

Кроме того, было проведено тестирование интерфейса пользователя [10]. Интерфейс был протестирован в браузерах: Google Chrome 58, Mozilla Firefox 53, Microsoft Edge, Microsoft IE 11, Opera 32, Safari 10.0. Проблем с версткой не обнаружено. Интерфейс пользователя в разных браузерах идентичен. Результаты тестирования совпали в ожидаемыми. Тест пройден успешно.

4.2. Юзабилити-тестирование

Юзабилити-тест (тест на эргономичность) – метод оценки удобства продукта в использовании, основанный на привлечении пользователей в качестве тестировщиков, испытателей и суммировании полученных от них выводов.

При прохождении данного теста тестировщикам было предложено решить следующие задачи:

1) найти конкретный товар в каталоге;
2) найти конкретный товар с помощью формы поиска;
3) найти информацию о доставке;
4) воспользоваться формой обратной связи.

Все задачи были решены всеми тестировщиками без каких-либо затруднений. Тест пройден успешно.

Отчет о юзабилити-тестировании представлен в приложении 2.

ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работы были выполнены следующие задачи:

1) произведен анализ предметной области;
2) разработан бриф на проектирование интернет-магазина;
3) реализован интернет-магазин спортивного питания «SportDrive»;
4) проведено тестирование интернет-магазина.

ЛИТЕРАТУРА

1. Дари К., Баланеску Э. PHP и MySQL. Создание интернет-магазина – М.: «Вильямс», 2010. – 640 с.

2. Коналлен Дж. Разработка Web-приложений с использованием UML – М.: «Вильямс», 2001. – 288 с.

3. Кузнецов М.В., Симдянов И.В., Голышев С.В. PHP 5 на примерах – СПб: «БХВ-Петербург», 2005. – 576 с.

4. Макнейл П. Веб-дизайн. Книга идей веб-разработчика. – СПб.: «Питер», 2014. – 288 с.

5. Мейерсон М., Скарборо М. Основы интернет-маркетинга. Все, что нужно знать, чтобы открыть свой магазин в интернете – СПб.: Манн, Иванов и Фербер, 2013. – 320 с.

6. Митин П.А. Как создать интернет-магазин? – М.: Феникс, 2015. – 252 с.

7. Пирогов В.Ю. Информационные системы и базы данных. Организация и проектирование. – СПб.: БХВ-Петербург, 2009. – 528 с.

8. Рассел Дж. Интернет-магазин – СПб.: Книга по Требованию, 2012. – 100 с.

9. Рябых А. Интернет-магазин – СПб: «Питер», 2010. – 192 с.

10. Ташков П. Веб-мастеринг на 100%: HTML, CSS, JavaScript, PHP, CMS, AJAX, графика, раскрутка – СПб: «Питер», 2010. – 512 с.

11. Томсон Л., Веллинг Л. Разработка Web-приложений на PHP и MySQL – К.: «ДиаСофт», 2001. – 672 с.

12. HTML/Specification. [Электронный ресурс] URL: http://www.w3.org/community/webed/wiki/HTML/Specifications#Specifications (дата обращения: 16.05.2017).

13. Описание MVC и его производных. [Электронный ресурс] URL: http://www.pvsm.ru/shpargalka/14840 (дата обращения: 09.05.2017).

14. Статистика использования PHP на веб-сайтах. [Электронный ресурс] URL: http://w3techs.com/technologies/details/pl-php/all/all (дата обращения: 09.05.2017).

15. Химонин Ю. Сбор и анализ требований к программному продукту. [Электронный ресурс] URL: https://pmi.ru/profes/Software_Requirements_Khimonin.pdf (дата обращения: 08.05.2017).

ПРИЛОЖЕНИЯ

Приложение 1

Скриншоты интерфейса интернет-магазина

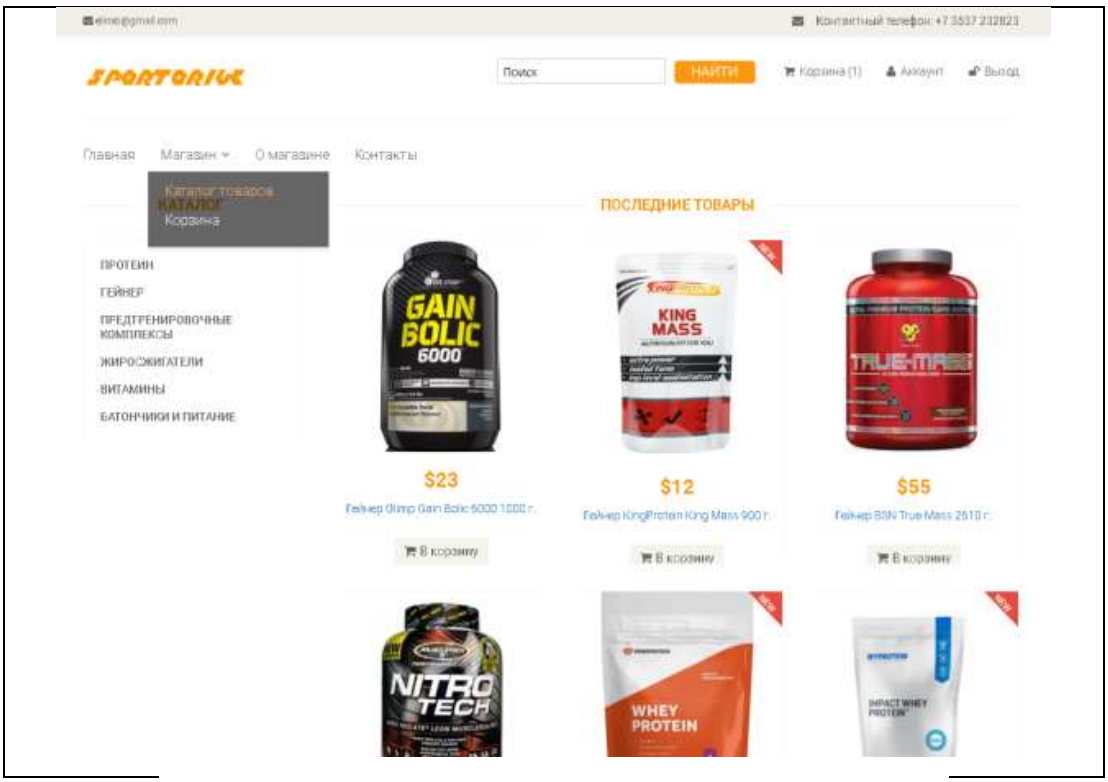


Рис. 1. Вид главной страницы интернет-магазина

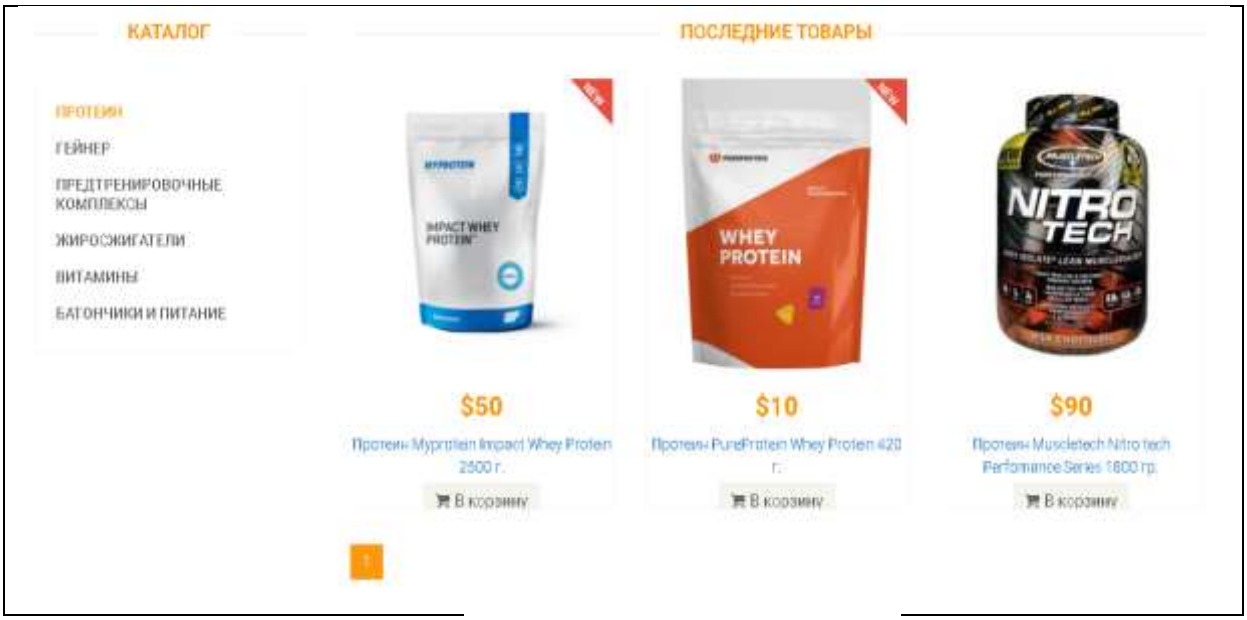


Рис. 2. Страница категории товара

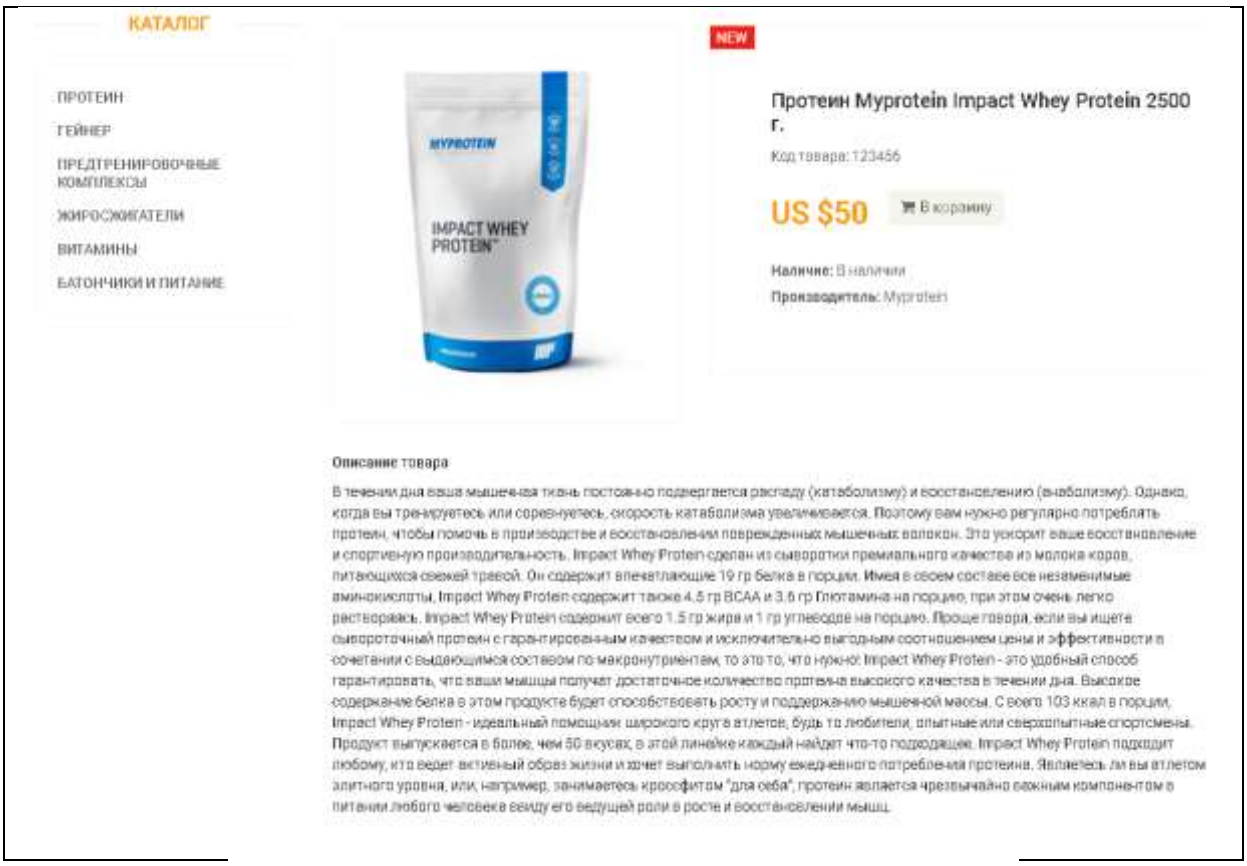


Рис. 3. Страница с полной информацией о товаре

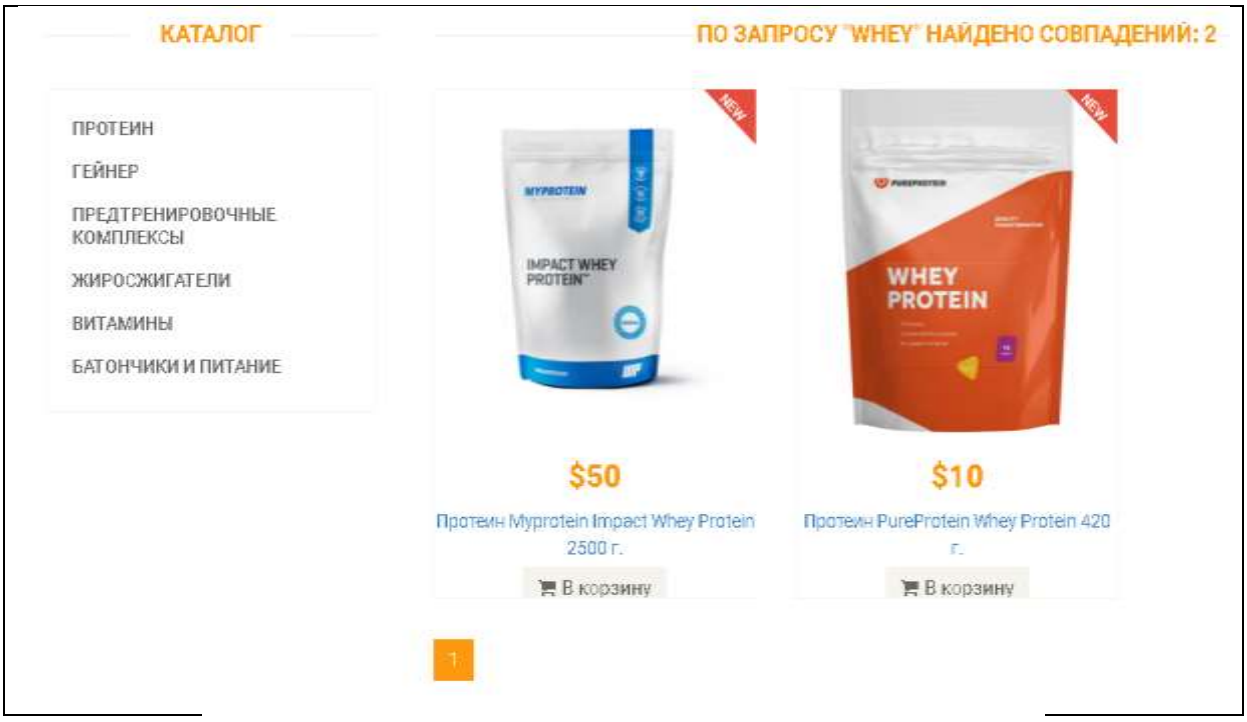


Рис. 4. Страница с результатами поискового запроса

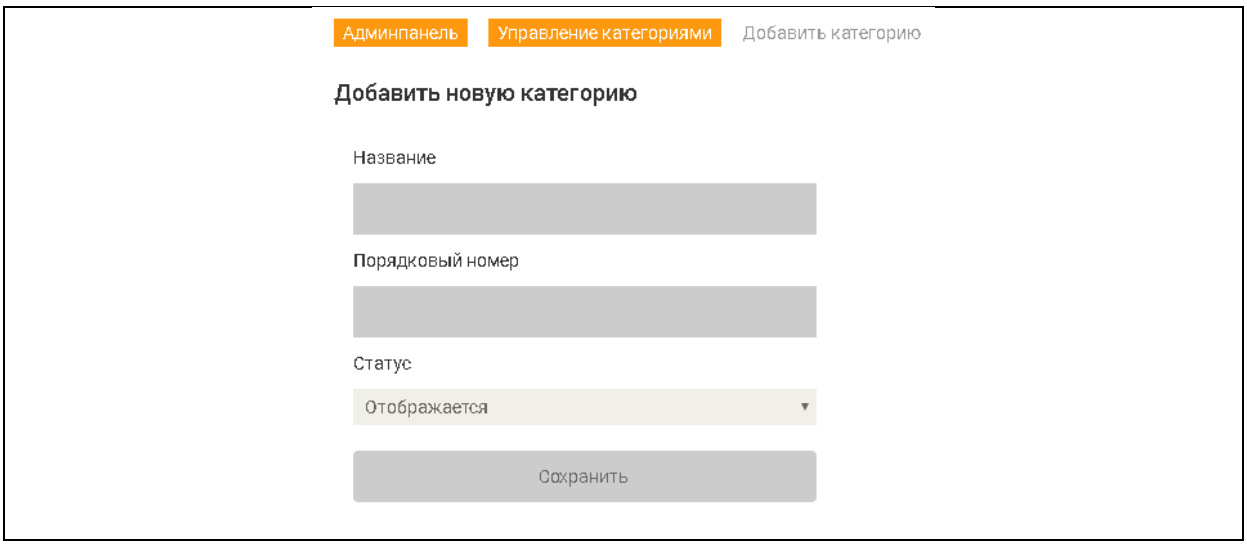


Рис. 5. Добавление категории в панели администратора

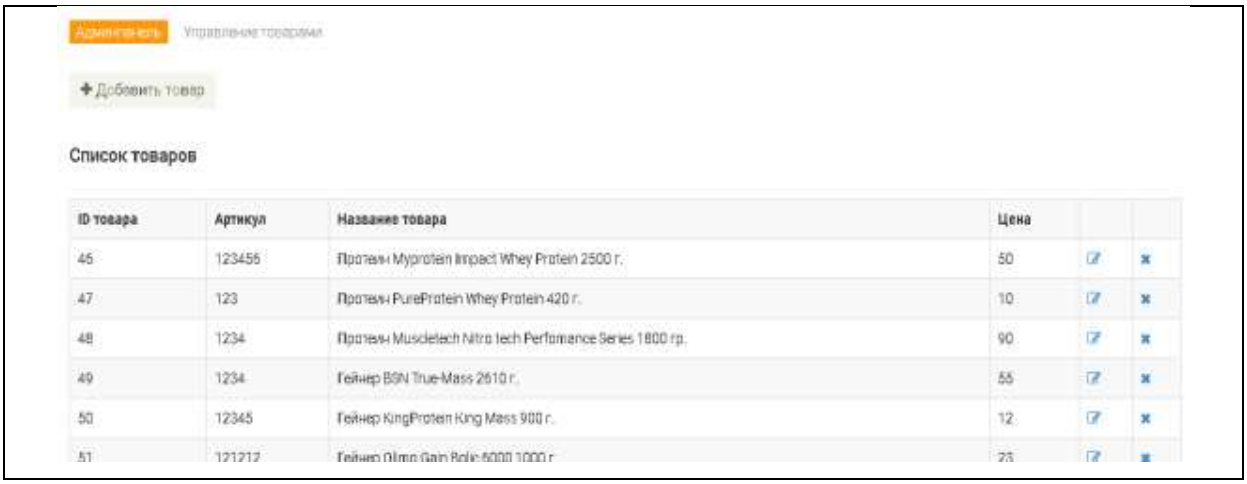


Рис. 6. Управление товарами в панели администратора

Рис. 7. Страница авторизации

Рис. 8. Форма обратной связи на странице «Контакты»

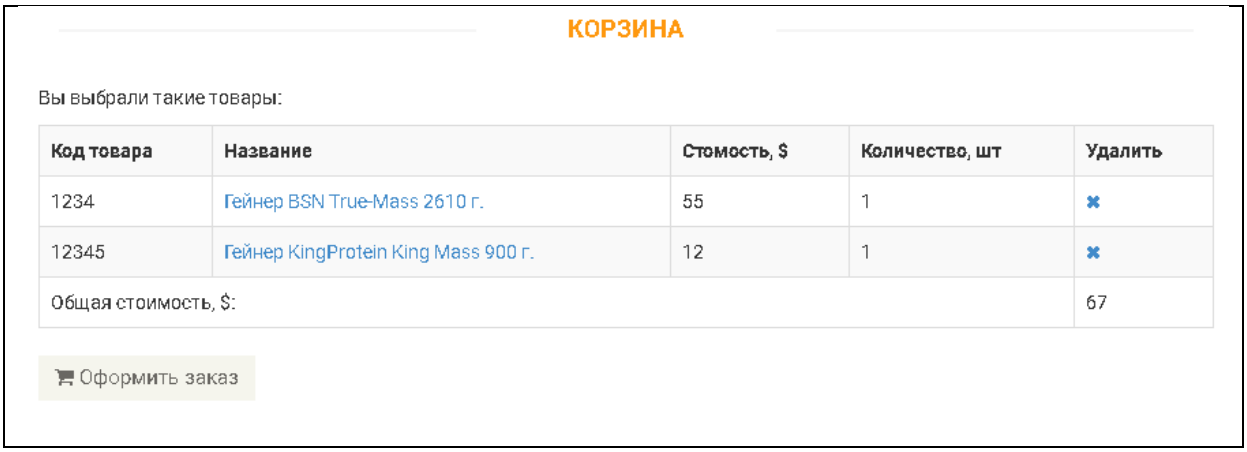


Рис. 9. Вид корзины с добавленными товарами

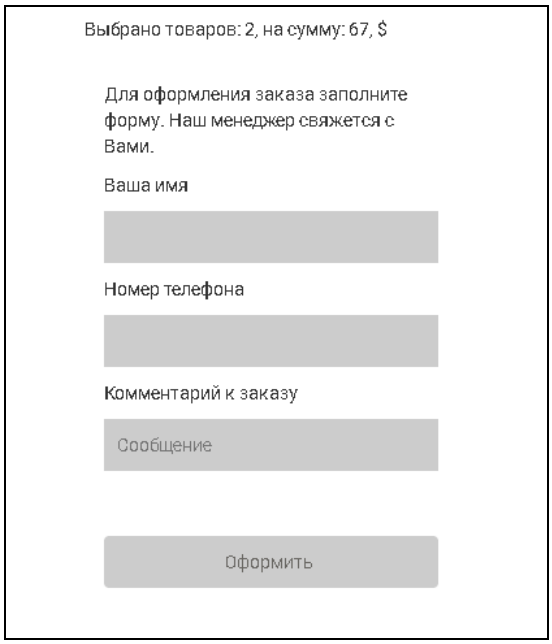


Рис. 10. Форма оформления заказа

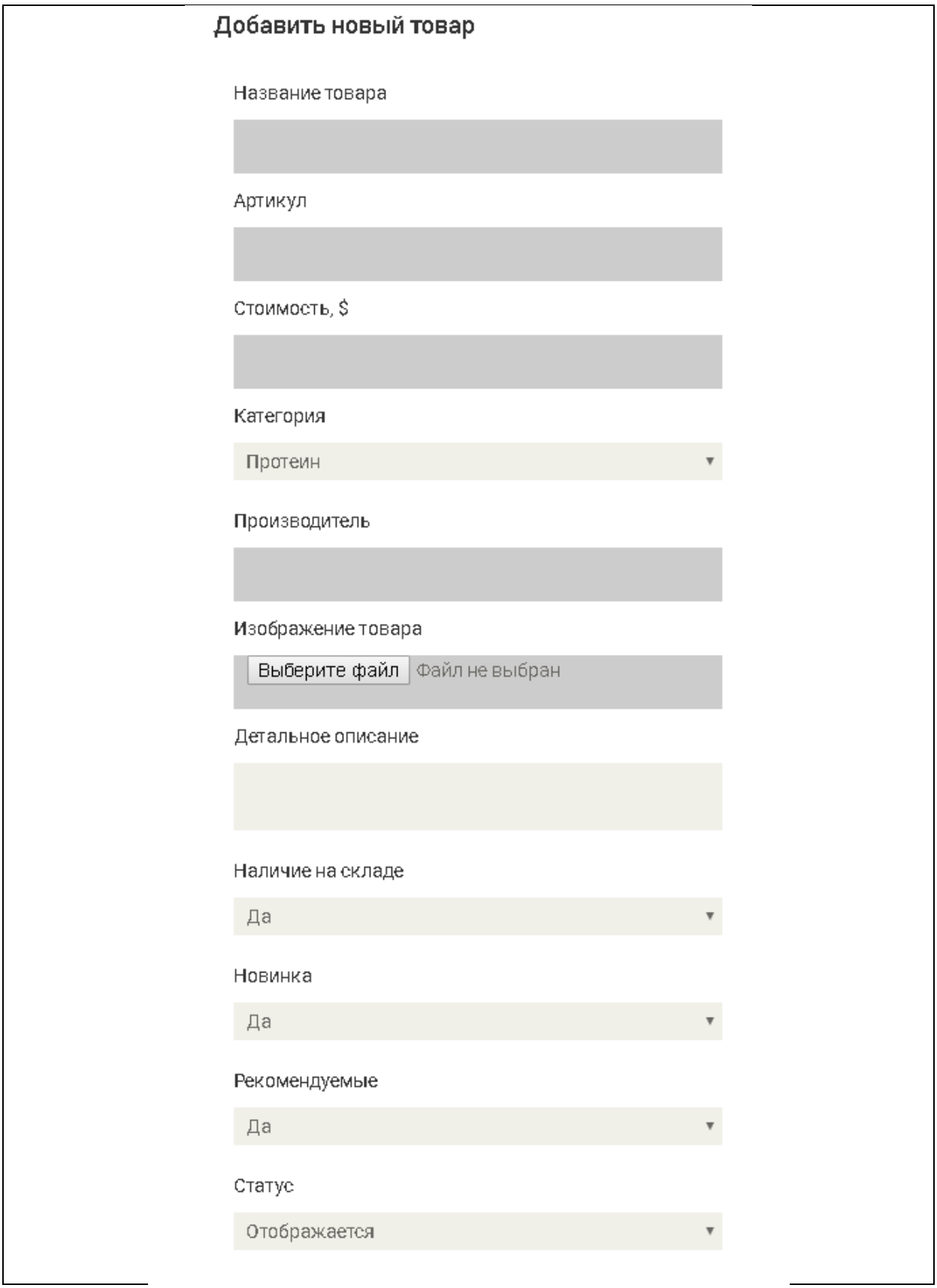
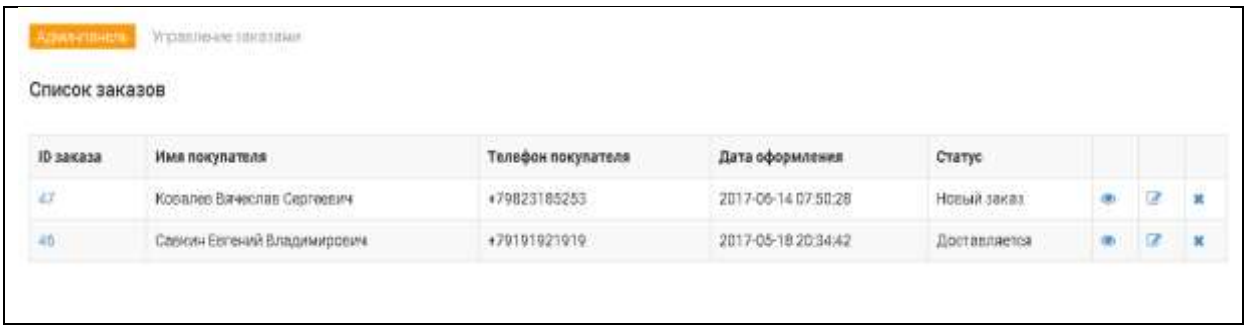


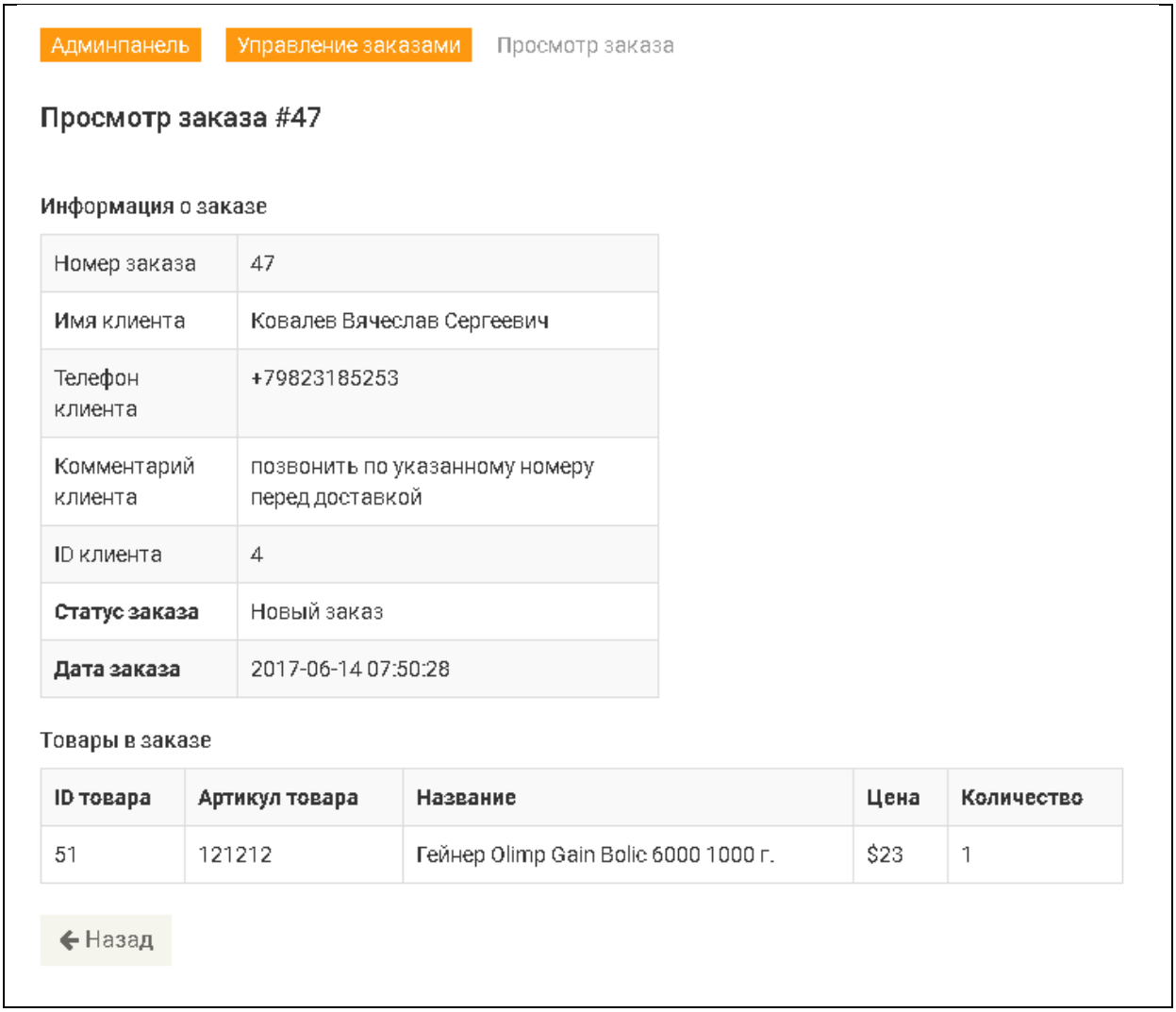
Рис. 11. Добавление товара в панели администратора



Список заказов

ID заказа	Имя покупателя	Телефон покупателя	Дата оформления	Статус			
47	Ковалев Вячеслав Сергеевич	+79823185253	2017-06-14 07:50:28	Новый заказ			
46	Савкин Евгений Владимирович	+79191921919	2017-05-18 20:34:42	Доставляется			

Рис. 12. Список всех заказов в панели администратора



Админпанель Управление заказами Просмотр заказа

Просмотр заказа #47

Информация о заказе

Номер заказа	47
Имя клиента	Ковалев Вячеслав Сергеевич
Телефон клиента	+79823185253
Комментарий клиента	позвонить по указанному номеру перед доставкой
ID клиента	4
Статус заказа	Новый заказ
Дата заказа	2017-06-14 07:50:28

Товары в заказе

ID товара	Артикул товара	Название	Цена	Количество
51	121212	Гейнер Olimp Gain Bolic 6000 1000 г.	$23	1

Назад

Рис. 13. Просмотр полной информации о заказе

Приложение 2

Отчет о юзабилити-тестировании

1. Объект исследования

Объектом исследования является сайт магазина по продаже спортивного питания «SportDrive»

2. Метод исследования

В качестве метода исследования был выбран метод юзабилити-тестирования «Мысли вслух».

3. План проведения тестирования

1. Планирование:
   1.1. Разработка заданий.
   1.2. Набор участников тестирования.
2. Проведение тестирования.
3. Анализ полученных данных.

4. Методика проведения тестирования

Для проведения исследования использовалось следующее оборудование и ПО:

1. ПК с операционной системой Windows 10;
2. Клавиатура и компьютерная мышь;
3. Встроенный в ПК микрофон.

Наблюдатель выдает список заданий участникам тестирования. Респонденты выполняют их последовательно и самостоятельно, при этом высказывая свои мысли и мнения, возникающие в процессе взаимодействия с приложением. Комментарии записываются через микрофон и затем представляются модератором в виде протокола тестирования.

5. Протокол заданий

1. Отправная точка: главная страница.

   Задание: найти контактную информацию о магазине.
2. Отправная точка: главная страница, пользователь авторизован, товары в корзине.

Задание: найти информацию о текущем состоянии корзины.

3. Отправная точка: главная страница.

   Задание: пройти процедуру регистрации.

4. Отправная точка: главная страница, в авторизованном режиме.

   Задание: добавить товар в корзину.

5. Отправная точка: главная страница, пользователь авторизован, товары в корзине.

   Задание: оформить заказ.

6. Респонденты

1. Веретенников Артур Альбертович, студент ЮУрГУ.
2. Ковалев Вячеслав Сергеевич, студент ЮУрГУ.
3. Савкин Евгений Владимирович, студент ЮУрГУ.

7. Результаты тестирования

В Табл. 1 приведены результаты юзабилити-тестирования, согласно которому ошибок выявлено не было.

Табл. 1. Результаты юзабилити-тестирования

Респонденты	**Доля выполненных задач**	**Доля правильно выполненных задач**	**Время на выполнение задач**	**Кол-во ошибок**
Веретенников Артур Альбертович	100%	100%	6 мин.	0
Ковалев Вячеслав Сергеевич	100%	100%	5 мин.	0
Савкин Евгений Владимирович	100%	100%	9 мин.	0