

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	4
1 ИССЛЕДОВАНИЕ ПОСТАВЛЕННОЙ ЗАДАЧИ .....	6
1.1 Обзор области применения .....	6
1.2 Обзор существующих аналогов .....	7
1.3 Формулировка задачи .....	8
1.4 Разработка алгоритма работы ПС .....	9
1.5 Разработка модели данных .....	11
1.6 Выбор инструментов разработки .....	12
1.7 Определение требований к программному средству .....	13
2 РАЗРАБОТКА ПРОГРАММНОГО СРЕДСТВА .....	14
2.1 Разработка интерфейса .....	14
2.2 Разработка и организация базы данных .....	14
2.3 Разработка системы помощи .....	18
2.4 Организация средств защиты и разграничения доступа .....	18
3 РЕАЛИЗАЦИЯ ПРОГРАММНОГО СРЕДСТВА .....	19
3.1 Физическая организация БД .....	19
3.2 Организация элементов управления .....	20
3.3 Тестирование программного средства .....	20
4 ВНЕДРЕНИЕ И ЭКСПЛУАТАЦИЯ .....	23
4.1 Назначение программного средства .....	23
4.2 Среда функционирования, требования к аппаратному обеспечению .....	23
4.3 Установка ПС .....	24
4.4 Руководство пользователя .....	24
ЗАКЛЮЧЕНИЕ .....	28
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	29
ПРИЛОЖЕНИЕ А. Фрагменты кода программного средства .....	30
ПРИЛОЖЕНИЕ Б. Общая блок-схема .....	42
ПРИЛОЖЕНИЕ В. Блок-схема алгоритма авторизации .....	43

## ВВЕДЕНИЕ

Экономическая деятельность в условиях рыночных отношений, ужесточение конкурентной борьбы и иные причины требуют постоянного совершенствования методов планирования и управления продажами товаров и услуг. Поэтому в настоящее время важное место занимает применение новых информационных технологий в работе организации по созданию представительства компании в сети Интернет, развитие электронного бизнеса.

Электронный бизнес, электронная коммерция (ЭК) - новое явление, мощно ворвавшееся в современную экономику. Развитие информационных технологий в последнее десятилетие XX в. смогло коренным образом изменить складывавшиеся веками технологии проведения коммерческих операций за счет широкого использования электронного обмена данными при проведении этих операций вместо традиционного документооборота.

В обыденном сознании людей термин «электронная коммерция» ассоциируется со сделками, совершаемыми в компьютерных сетях с помощью электронного документооборота. На самом деле, развитие ЭК не является самоцелью, она является средством для решения проблемы приведения в соответствие современным требованиям информационной структуры поддержки рынка.

ЭК - это технология для поддержания внешних бизнес-контактов. В общем случае «система электронной коммерции» представляет собой определенную Интернет-технологию, предоставляющую участникам системы следующие возможности:

- производителям и поставщикам товаров и услуг различных категорий - представить в сети Интернет товары и услуги (в том числе онлайн-услуги и доступ к информационным ресурсам), принимать через Интернет и обрабатывать заказы клиентов;

- покупателям (клиентам) - просматривать с помощью стандартных Интернет-браузеров каталоги и прайс-листы предлагаемых товаров и услуг и оформлять через Интернет заказы (заявки, запросы) на интересующие товары и услуги.

В современном мире Интернет является крупнейшим открытым хранилищем информации из различных областей человеческой деятельности. С созданием WWW-среды начали формироваться разнообразные онлайн-модели — электронные газеты, книги, реклама и др. Такие признаки, как глобальность, нематериальность и надгосударственность присущи только Интернет. В мире сейчас насчитывается 300-350 млн. пользователей Сети. Интернет прочно занял свое место в современном мире и его роль с каждым годом возрастает.

В противоположность традиционным средствам телекоммуникаций, основная функция которых состоит в доставке информации, Интернет является не просто передатчиком информации, а значительно больше - глобальным виртуальным рынком. Существование электронного рынка в среде Интернет обусловлено возможностью реализации в Интернет платежных систем, позволяющих

интерактивно оплачивать товары и услуги. Рынок Интернет более эффективен по сравнению с традиционным, так как обеспечивает потребителя самой полной информацией о товарах и услугах, предоставляя значительно больший контроль над процессом поиска и выбора информации среди значительно большего ее объема, чем, например, в печатных изданиях, в рекламе на телевидении или по радио.

ЭК - это не только продажа и покупка товаров и услуг через Интернет для получения прибыли, но и создание спроса на товары и услуги, послепродажная поддержка и обслуживание клиентов, облегчение взаимодействия с партнерами.

Главная особенность рынка ЭК состоит в его интерактивности. Поэтому создание интернет-магазина для такого бизнеса, как, например, продажа женских аксессуаров, очень хорошо способствует его развитию.

Консультанты Bain & Company подсчитали, что глобальная розничная торговля украшениями приносит доход в 15 миллиардов долларов ежегодно. При этом информация, опубликованная финансовой консалтинговой компанией Morgan Stanley, говорит о том, что 36% мировых продаж украшений происходит онлайн. Таким образом, организация такого бизнеса в Интернет делает его частью гигантской индустрии.

Если сравнивать различные категории товаров на рынке электронной коммерции, то можно заметить, что торговля ювелирными украшениями не требует дополнительного места для хранения и реализации, а стоимость доставки сравнительно мала. К тому же продавцу не составит труда регулярно обновлять коллекцию своих изделий онлайн. Существуют украшения практически для любой части тела. Также в вашем распоряжении бесчисленное множество драгоценных металлов и камней; добавьте к этому нетрадиционные материалы, такие как дерево и кожа, и вам хватит ассортимента на 10 лет вперед. И это без учета различных вариантов дизайна.

Целью проекта является разработка веб-приложения «Система продажи женских аксессуаров». Исходя из поставленной цели, были определены задачи:

- изучить принципы построения интернет-магазинов;
- исследовать структуру электронного магазина;
- изучить особенности бизнеса по продаже женских аксессуаров, потенциальную целевую аудиторию интернет-магазина и ее потребности;
- выполнить сравнительный анализ магазинов конкурентов;
- провести анализ полученной информации и выбрать на основе полученных результатов интернет-технологии для разработки интернет-магазина;
- разработать веб-приложение «Система продажи женских аксессуаров» используя результаты проведенного анализа.

# 1 ИССЛЕДОВАНИЕ ПОСТАВЛЕННОЙ ЗАДАЧИ

## 1.1 Обзор области применения

Производство украшений – это занятие, которое берет свое начало в глубине тысячелетий, по сути оно является одной из древнейших профессий в мире. За весьма продолжительную историю своего существования этот бизнес не перестал быть достаточно высокодоходным, и как всегда - перспективным. Сейчас, практически в каждом городе нашей планеты присутствует великое множество ювелирных магазинов и салонов, а предлагаемая ими продукция пользуется весьма неплохим спросом среди людей. В связи с такой большой конкуренцией требуется более активное и грамотное продвижение собственного бизнеса. Продвижение товаров через сеть интернет — это довольно эффективный способ рекламы своих товаров и услуг. Он уже заслужил уважение, как малого предпринимательства, так и крупных международных корпораций.

В первые годы бума интернет-торговли мастера, создающие украшения, могли найти множество клиентов, просто размещая свои товары на Ebay. Однако Ebay быстро превзошел себя как торговая онлайн платформа для самодельных вещей – теперь на этой площадке обитают настоящие акулы бизнеса, среди которых нет места независимым художникам, если только они не начнут использовать какие-то дополнительные, скорее всего дорогостоящие маркетинговые инструменты.

В 2005 Etsy запустили онлайн платформу для мастеров различных направлений, но сейчас конкуренция на сайте слишком высока. По данным Etsy, к концу прошлого года число продавцов на их сайте превысило 1 миллион. В результате многие продавцы переезжают на новые малоизвестные платформы, но сложно предсказать, что с ними станет в будущем. Лучшим решением будет создать свой собственный сайт для продажи ювелирных изделий.

Продавая украшения ручной работы на своем сайте, продавец остается единоличным владельцем своих товаров, к тому же дизайн сайта и качество работы с клиентами в данном случае будет полностью зависеть только от него. Кроме того, любые используемые маркетинговые средства принесут выгоду только личной платформе, а не кому-то еще.

Существует огромное количество платформ, которые можно использовать для создания собственного онлайн магазина. Главное, чтобы выбранная платформа отвечала минимальным требованиям для создания онлайн-магазина. Основное, что должно быть на сайте, целью которого являются продажи, – это форма для оформления заказа, каталог, список пожеланий и отзывов, возможность создавать и использовать купоны, а также легко вносить изменения в контент и ассортимент товаров. Поэтому для того, чтобы исключить ограничения и недостатки таких платформ, лучше создавать интернет-магазин с нуля.

## 1.2 Обзор существующих аналогов

Один из интернет-магазинов по продаже украшений называется Ukrashenie.by. Владелец является индивидуальный предприниматель, бизнес организован с 2014 года. Интернет-магазин Ukrashenie.by предлагает купить в Минске бижутерию на самый изысканный вкус.

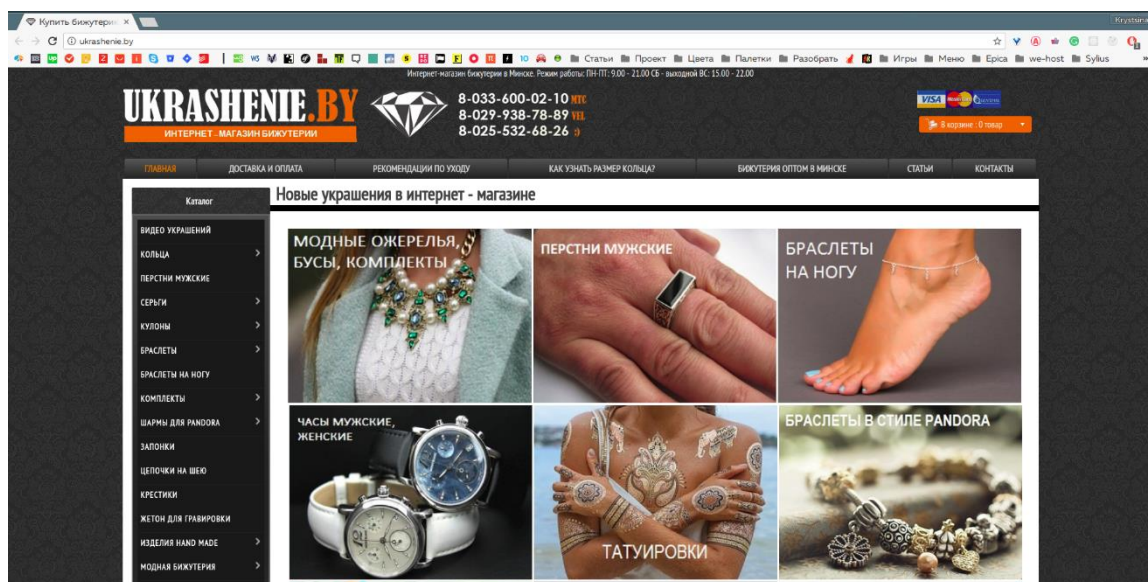


Рисунок 1.1 — интернет-магазин Ukrashenie.by

Интернет-магазин мастерской «Эго-Арт», открывшейся в 2000 году, имеет розничную сеть магазинов в Минске, Витебске, Орше и Полоцке. В «Эго-Арт» вы можете купить кольца, серьги, кулоны; обменять ювелирные изделия; заказать изготовление ювелирных изделий по каталогу; отремонтировать ювелирное изделие.

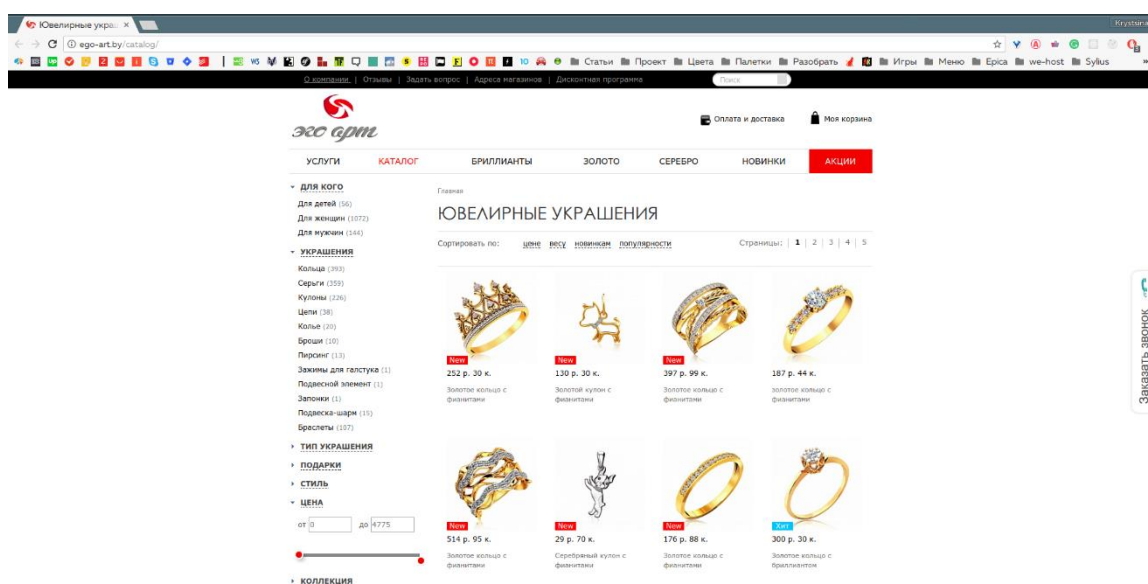


Рисунок 1.2 — интернет-магазин «Эго-Арт»



KUPIVIP.BY – это первый fashion аутлет онлайн, работающий на рынке стран СНГ. KUPIVIP.BY меняет стандарты покупки модных товаров, предлагая распродажи со скидками до 90% на престижные мировые бренды. Для бизнес-партнеров KUPIVIP.BY – это универсальная платформа для ликвидации складских остатков модных товаров.

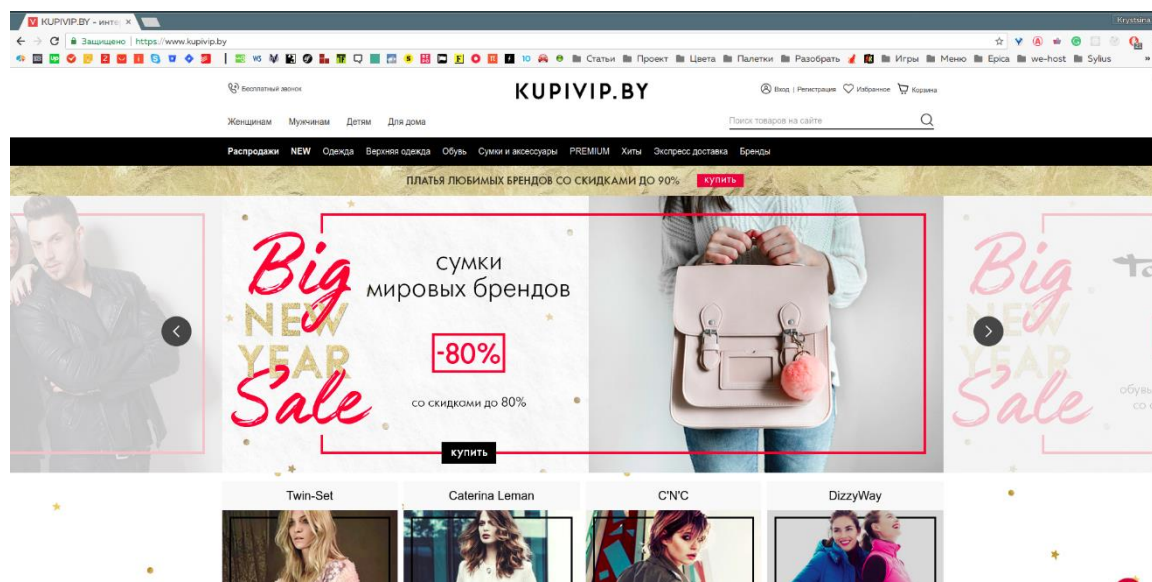


Рисунок 1.3 — интернет-магазин «KUPIVIP.BY»

KUPIVIP Group – крупнейший российский игрок в сфере электронной коммерции. В состав KUPIVIP Group входят: KUPIVIP.RU, KUPIVIP.BY, KUPIVIP.KZ, сервисное подразделение KUPIVIP E-commerce Services.

### 1.3 Формулировка задачи

После предварительной подготовки и определения целей можно перейти к формулировке задачи. Формулировка задачи производится с постепенным уточнением задач от общего к частному, т.е. сверху вниз. Обратный подход позволит произвести более четкую группировку подзадач по каждому направлению. Совместное применение данных подходов позволит более четко увидеть общую картину общего плана задач. Качество формулировки задачи влияет на дальнейшее планирование проекта и зависит как от четкости ее постановки, так и от подхода к ее составлению.

В нашем случае разрабатываемая система должна обладать следующими возможностями:

- 1) создание и удаление категорий для разного вида товаров;
- 2) создание и удаление товаров для разных категорий;
- 3) создание характеристик товаров для каждой категории;
- 4) возможность оставлять отзывы под товарами;
- 5) возможность расширения функционала (добавление способов оплаты и доставки, валют, языков сайта и прочего).

## 1.4 Разработка алгоритма работы ПС

Программное средство подразумевает работу с базой данных.

Для того, чтобы пользоваться системой, пользователь должен сначала зарегистрироваться, а затем войти в систему под своим именем пользователя и паролем.

Алгоритм авторизации изображен на рисунке 1.4.



Рисунок 1.4 — Алгоритм авторизации

При регистрации необходимо проверить, существует ли в системе пользователь с таким именем. Если пользователь уже существует – регистрация невозможна. Если пользователя не существует – перед занесением в базу необходимо посчитать хэш от пароля, т.к. хранить пароли в чистом виде небезопасно. При авторизации пользователя необходимо найти пользователя с таким именем в базе данных. Если пользователь найден – нужно посчитать хэш от предоставленного им пароля и сравнить с хэшем, который хранится в базе данных. Если хэш совпадает – пользователь предоставил правильный пароль и может быть авторизован. После авторизации пользователь попадает на главное окно приложения, где имеет доступ к просмотру каталога товаров, а также к оформлению заказов.

Общий алгоритм изображен на рисунке 1.5.

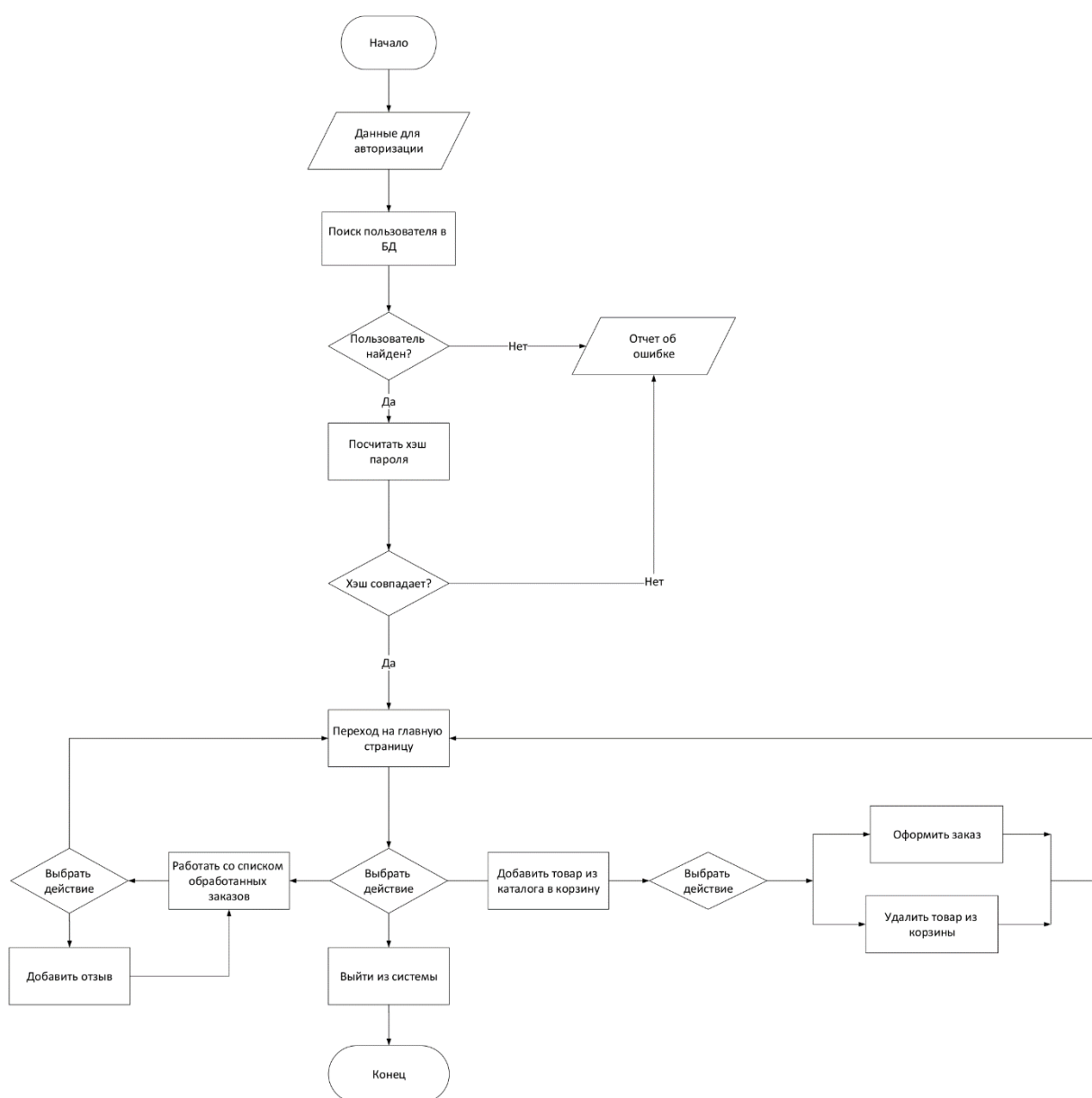


Рисунок 1.5 — Общий алгоритм



## 1.5 Разработка модели данных

Система продажи женских аксессуаров будет работать с базой данных.

База данных — совокупность данных, организованных в соответствии с концептуальной структурой, описывающей характеристики этих данных и взаимоотношения между ними, причем такое собрание данных, которое поддерживает одну или более областей применения. Существует огромное количество разновидностей баз данных, отличающихся по различным критериям. Именно поэтому стоит уделить особенное внимание тому, какой тип БД подойдет для решения поставленной задачи.

Основные типы БД, применяемые для сайтов — документо-ориентированные и реляционные. Документо-ориентированные БД (MongoDB, CouchDB) будет лучше если в системе часто приходится копировать записи с их зависимостями. Или, например, если не все записи имеют одинаковые поля. Если же приходится обрабатывать массивы данных для подсчета неких статистических данных (например, среднее состояние кассы всех компаний за N-ый период), то здесь к месту реляционный подход (MySQL, PostgreSQL).

Исходя из поставленной задачи, для разработки системы требуется использовать реляционную модель данных.

Реляционная модель данных — логическая модель данных, прикладная теория построения баз данных, которая является приложением к задачам обработки данных таких разделов математики, как теория множеств и логика первого порядка.

Реляционная модель данных включает следующие компоненты:

1) структурный аспект (составляющая) — данные в базе данных представляют собой набор отношений;

2) аспект (составляющая) целостности — отношения (таблицы) отвечают определенным условиям целостности. РМД поддерживает декларативные ограничения целостности уровня домена (типа данных), уровня отношения и уровня базы данных;

3) аспект (составляющая) обработки (манипулирования) — РМД поддерживает операторы манипулирования отношениями (реляционная алгебра, реляционное исчисление).

Термин «реляционный» означает, что теория основана на математическом понятии отношение (relation). В качестве неформального синонима термину «отношение» часто встречается слово таблица. Необходимо помнить, что «таблица» есть понятие нестрогое и неформальное и часто означает не «отношение» как абстрактное понятие, а визуальное представление отношения на бумаге или экране.

В качестве используемой СУБД была выбрана MySQL.

MySQL — свободная реляционная система управления базами данных. Разработку и поддержку MySQL осуществляет корпорация Oracle, получившая права на торговую марку вместе с поглощённой Sun Microsystems, которая ранее приобрела шведскую компанию MySQL AB.

В базе данных отражается информация об определенной предметной области. Под предметной областью понимается часть реального мира, данные о которой

должны отражаться в проектируемой БД. Информация, необходимая для описания предметной области, зависит от типа самой предметной области, а также от того, какую именно информацию об этой предметной области требуется хранить и обрабатывать.

В данном курсовом проекте разрабатывается система продажи женских аксессуаров. Для построения концептуальной модели выделяются наиболее существенные объекты, необходимые при работе магазина. В данном магазине продаются женские аксессуары, следовательно, одним из объектов для базы данных будут сами аксессуары. Так как аксессуары классифицируются по категориям, то следующим объектом базы данных будет непосредственно вид аксессуара. Существование магазина невозможно без покупателей, следовательно, третьим объектом базы данных моего магазина будет покупатель. Как и в любом Интернет магазине покупатель сможет выбирать товар и добавлять его в корзину, после чего, все заказанные товары формируются в заказ, и отправляется администратору, значит, необходим еще объект - заказ.

Таким образом, основными сущностями интернет-магазина женских аксессуаров будут:

- аксессуары (product);
- виды аксессуаров (taxon);
- покупатели (customer);
- заказ (order).

После того, как выделены основные сущности системы, можно переходить к физическому проектированию базы данных. Любая база данных состоит из таблиц (отношений), поэтому задача будет состоять в том, чтобы построить таблицы, основываясь на логической модели базы данных. Сущности будут расширены и дополнены, для каждого атрибута будет подобран свой тип данных, результат будет описывать данные уже средствами конкретной СУБД, в данном случае MySQL.

## **1.6 Выбор инструментов разработки**

Программное средство будет разрабатываться на языке PHP с использованием фреймворков Symfony и Sylius.

PHP — скриптовый язык общего назначения, интенсивно применяемый для разработки веб-приложений. В настоящее время поддерживается подавляющим большинством хостинг-провайдеров и является одним из лидеров среди языков, применяющихся для создания динамических веб-сайтов.

В области веб-программирования, в частности серверной части, PHP — один из популярных сценарных языков (наряду с JSP, Perl и языками, используемыми в ASP.NET).

Популярность в области построения веб-сайтов определяется наличием большого набора встроенных средств для разработки веб-приложений. Основные из них:

- автоматическое извлечение POST и GET-параметров, а также переменных окружения веб-сервера в предопределённые массивы;

- взаимодействие с большим количеством различных систем управления базами данных;
- автоматизированная отправка HTTP-заголовков;
- работа с HTTP-авторизацией;
- работа с cookies и сессиями;
- работа с локальными и удалёнными файлами, сокетами;
- обработка файлов, загружаемых на сервер.

В настоящее время PHP используется сотнями тысяч разработчиков. Согласно рейтингу корпорации TIOBE, базирующемуся на данных поисковых систем, в мае 2016 года PHP находился на 6 месте среди языков программирования. К крупнейшим сайтам, использующим PHP, относятся Facebook, Wikipedia и др.

Symfony — свободный фреймворк, написанный на PHP, который использует паттерн Model-View-Controller.

Symfony предлагает быструю разработку и управление веб-приложениями, позволяет легко решать рутинные задачи веб-программиста. Работает только с PHP 5 и выше. Имеет поддержку множества баз данных (MySQL, PostgreSQL, SQLite или любая другая PDO-совместимая СУБД). Информация о реляционной базе данных в проекте должна быть связана с объектной моделью. Это можно сделать при помощи ORM инструмента. Symfony поставляется с двумя из них: Propel и Doctrine.

Symfony бесплатен и публикуется под лицензией MIT.

Sylius - eCommerce фреймворк, разработанный по современным методологиям на основе PHP и Symfony. Sylius предоставляет все функции, необходимые для успешного роста вашего бизнеса. Высокое качество кода позволяет легко интегрировать и создать уникальный дизайн вашего сайта электронной коммерции.

## **1.7 Определение требований к программному средству**

Разрабатываемая система должна обладать следующими возможностями:

- 1) создание и удаление категорий для разного вида товаров;
- 2) создание и удаление товаров для разных категорий;
- 3) создание характеристик товаров для каждой категории;
- 4) возможность оставлять отзывы под товарами;
- 5) возможность расширения функционала (добавление способов оплаты и доставки, валют, языков сайта и прочего).

Помимо вышеперечисленного сайт должен обладать простым и понятным пользовательским интерфейсом, удобным и функциональным режимом администрирования, а также обладать ярким и привлекательным дизайном.

## 2 РАЗРАБОТКА ПРОГРАММНОГО СРЕДСТВА

### 2.1 Разработка интерфейса

Для создания интерфейса пользователя был использован фреймворк Bootstrap, который предоставляет ряд часто используемых в веб-разработке компонентов, а также позволяет строить responsive-приложения, которыми одинаково удобно пользоваться как с персональных компьютеров, так и с мобильных устройств. Bootstrap (также известен как Twitter Bootstrap) — свободный набор инструментов для создания сайтов и веб-приложений. Включает в себя HTML- и CSS-шаблоны оформления для типографики, веб-форм, кнопок, меток, блоков навигации и прочих компонентов веб-интерфейса, включая JavaScript-расширения.

### 2.2 Разработка и организация базы данных

Проектирование баз данных — это процесс создания схемы базы данных и определения необходимых ограничений целостности.

Основные задачи проектирования:

- обеспечение хранения в БД всей необходимой информации.
- обеспечение возможности получения данных по всем необходимым запросам.
- сокращение избыточности и дублирования данных.
- обеспечение целостности базы данных.

Логическое (дatalogическое) проектирование — создание схемы базы данных на основе конкретной модели данных, например, реляционной модели данных. Для реляционной модели данных даталогическая модель — набор схем отношений, обычно с указанием первичных ключей, а также «связей» между отношениями, представляющих собой внешние ключи.

Цель моделирования данных на логическом уровне состоит в обеспечении разработчика информационных систем (ИС) концептуальной схемой базы данных в форме одной модели или нескольких локальных моделей, которые относительно легко могут быть отображены в любую систему баз данных.

Логическая модель базы данных сайта содержит следующие основные сущности:

Таблица 2.1 — Сущность «customer»

Имя поля	Тип данных	Тип ключа
id	INT(11)	PK
customer_group_id	INT(11)	FK
default_address_id	INT(11)	FK
email	VARCHAR(255)	
first_name	VARCHAR(255)	
last_name	VARCHAR(255)	
phone_number	VARCHAR(255)	

Таблица 2.2 — Сущность «shop\_user»

Имя поля	Тип данных	Тип ключа
id	INT(11)	PK
customer_id	INT(11)	FK
username	VARCHAR(255)	
enabled	TINYINT(1)	
salt	VARCHAR(255)	
password	VARCHAR(255)	
last_login	DATETIME	
email	VARCHAR(255)	
roles	LONGTEXT	

Таблица 2.3 — Сущность «product\_review»

Имя поля	Тип данных	Тип ключа
id	INT(11)	PK
product_id	INT(11)	FK
author_id	INT(11)	FK
title	VARCHAR(255)	
rating	INT(11)	
comment	LONGTEXT	

Таблица 2.4 — Сущность «address»

Имя поля	Тип данных	Тип ключа
id	INT(11)	PK
customer_id	INT(11)	FK
first_name	VARCHAR(255)	
last_name	VARCHAR(255)	
phone_number	VARCHAR(255)	
street	VARCHAR(255)	
sity	VARCHAR(255)	
postcode	VARCHAR(255)	

Таблица 2.5 — Сущность «customer\_group»

Имя поля	Тип данных	Тип ключа
id	INT(11)	PK
code	VARCHAR(255)	
name	VARCHAR(255)	

Таблица 2.6 — Сущность «product\_options»

Имя поля	Тип данных	Тип ключа
product_id	INT(11)	FK
option_id	INT(11)	FK



Таблица 2.7 — Сущность «taxon»

Имя поля	Тип данных	Тип ключа
id	INT(11)	PK
tree_root	INT(11)	FK
parent_id	INT(11)	FK
code	VARCHAR(255)	

Таблица 2.8 — Сущность «product\_option»

Имя поля	Тип данных	Тип ключа
id	INT(11)	PK
code	VARCHAR(255)	
position	INT(11)	

Таблица 2.9 — Сущность «product»

Имя поля	Тип данных	Тип ключа
id	INT(11)	PK
main_taxon_id	INT(11)	FK
code	VARCHAR(255)	
enabled	TINYINT(1)	

Таблица 2.10 — Сущность «product\_image»

Имя поля	Тип данных	Тип ключа
id	INT(11)	PK
owner_id	INT(11)	FK
path	VARCHAR(255)	

Таблица 2.11 — Сущность «order»

Имя поля	Тип данных	Тип ключа
id	INT(11)	PK
shipping_address_id	INT(11)	FK
customer_id	INT(11)	FK
number	VARCHAR(255)	
notes	VARCHAR(1000)	
state	VARCHAR(255)	

Таблица 2.12 — Сущность «order\_item»

Имя поля	Тип данных	Тип ключа
id	INT(11)	PK
order_id	INT(11)	FK
variant_id	INT(11)	FK
unit_price	INT(11)	

Таблица 2.13 — Сущность «currency»

Имя поля	Тип данных	Тип ключа
id	INT(11)	PK
code	VARCHAR(3)	

Таблица 2.14 — Сущность «shipment»

Имя поля	Тип данных	Тип ключа
id	INT(11)	PK
order_id	INT(11)	FK
state	VARCHAR(255)	

Таблица 2.15 — Сущность «payment»

Имя поля	Тип данных	Тип ключа
id	INT(11)	PK
order_id	INT(11)	FK
amount	INT(11)	
state	VARCHAR(255)	

Таблица 2.16 — Сущность «product\_variant»

Имя поля	Тип данных	Тип ключа
id	INT(11)	PK
product_id	INT(11)	FK
tax_category_id	INT(11)	FK
shipping_category_id	INT(11)	FK
code	VARCHAR(255)	

Таблица 2.17 — Сущность «channel»

Имя поля	Тип данных	Тип ключа
id	INT(11)	PK
default_locale_id	INT(11)	FK
base_currency_id	INT(11)	FK
code	VARCHAR(255)	
description	LONGTEXT	

Таблица 2.18 — Сущность «tax\_rate»

Имя поля	Тип данных	Тип ключа
id	INT(11)	PK
category_id	INT(11)	FK
code	VARCHAR(255)	
amount	DECIMAL(10, 5)	

Таблица 2.19 — Сущность «payment\_method»

Имя поля	Тип данных	Тип ключа
id	INT(11)	PK
code	VARCHAR(255)	

Таблица 2.20 — Сущность «tax\_category»

Имя поля	Тип данных	Тип ключа
id	INT(11)	PK
code	VARCHAR(255)	
description	LONGTEXT	

Таблица 2.21 — Сущность «locale»

Имя поля	Тип данных	Тип ключа
id	INT(11)	PK
code	VARCHAR(3)	

## 2.3 Разработка системы помощи

Интернет-магазин — сайт, торгующий товарами посредством сети интернет. Позволяет пользователям онлайн, в своём браузере или через мобильное приложение, сформировать заказ на покупку, выбрать способ оплаты и доставки заказа, оплатить заказ. При этом продажа товаров осуществляется дистанционным способом, и она накладывает ограничения на продаваемые товары.

Система помощи представлена руководством пользователя, подсказками в интерфейсе пользователя и в виде ошибок, появляющихся при совершении пользователем какого-либо некорректного действия. Все поля, которые подразумевают ввод данных пользователем, имеют подписи и placeholder'ы. Если требуются пояснения — под полями присутствуют текстовые блоки с поясняющим текстом. Программное средство обладает интуитивно понятным интерфейсом.

## 2.4 Организация средств защиты и разграничения доступа

Защита и разграничение доступа реализованы в виде окна авторизации, которое позволяет разграничить доступ пользователей. В окне авторизации отображаются поля для ввода логина и пароля. Для входа в программу от имени администратора требуется указать соответствующие данные по ссылке входа для администраторов. Для входа на сайт в режиме обычного пользователя прежде всего необходимо пройти регистрацию. После прохождения регистрации клиент имеет право оставлять отзывы о товарах.

## 3 РЕАЛИЗАЦИЯ ПРОГРАММНОГО СРЕДСТВА

### 3.1 Физическая организация БД

Физическая модель данных описывает данные средствами конкретной СУБД. Данная физическая модель данных реализована средствами СУБД MySQL.

Физическая структура БД сайта представлена на рисунке 3.1.



Рисунок 3.1 — Физическая структура БД

Отношения, разработанные на стадии формирования логической модели данных, преобразуются в таблицы, атрибуты становятся столбцами таблиц, для

ключевых атрибутов создаются уникальные индексы, домены преобразуются в типы данных, принятые в конкретной СУБД.

Ограничения, имеющиеся в логической модели данных, реализуются различными средствами СУБД, например, при помощи индексов, декларативных ограничений целостности, триггеров, хранимых процедур. При этом опять-таки решения, принятые на уровне логического моделирования, определяют некоторые границы, в пределах которых можно развивать физическую модель данных.

Точно также, в пределах этих границ можно принимать различные решения. Например, отношения, содержащиеся в логической модели данных, должны быть преобразованы в таблицы, но для каждой таблицы можно дополнительно объявить различные индексы, повышающие скорость обращения к данным.

Все таблицы связаны между собой отношением «один ко многим».

Связь «один ко многим» — наиболее распространенный вид связи. При такой связи каждой строке таблицы А может соответствовать множество строк таблицы Б, однако каждой строке таблицы Б может соответствовать только одна строка таблицы А.

### **3.2 Организация элементов управления**

Элементы управления в программном средстве представлены базовыми компонентами фреймворка Bootstrap. Основные инструменты Bootstrap:

Сетки — заранее заданные размеры колонок, которые можно сразу же использовать, например ширина колонки 140 px относится к классу `.span2` (`.col-md-2` в третьей версии фреймворка), который можно использовать в CSS-описании документа.

Шаблоны — фиксированный или резиновый шаблон документа.

Типографика — описания шрифтов, определение некоторых классов для шрифтов, таких как код, цитаты и т. п.

Медиа — представляет некоторое управление изображениями и видео.

Таблицы — средства оформления таблиц, вплоть до добавления функциональности сортировки.

Формы — классы для оформления форм и некоторых событий, происходящих с ними.

Навигация — классы оформления для табов, вкладок, страничности, меню и панели инструментов.

Алерты — оформление диалоговых окон, подсказок и всплывающих окон.

Все элементы имеют адаптивный дизайн, что обеспечивает удобное отображение сайта не только на десктопных, но и на мобильных устройствах.

### **3.3 Тестирование программного средства**

Задачей функционального тестирования является проверка возможности программного обеспечения в определенных условиях решать задачи, необходимые пользователю.



Функциональное тестирование рассматривает заранее указанное поведение и основывается на анализе спецификаций функциональности компонента или системы в целом.

По сути, функциональное тестирование имитирует фактическое использование программного средства конечным пользователем.

В зависимости от степени доступа к коду системы можно выделить два типа функциональных испытаний:

- тестирование black box (черный ящик) – проведение функционального тестирования без доступа к коду системы,
- тестирование white box (белый ящик) – функциональное тестирование с доступом к коду системы.

Тестирование black box проводится без знания внутренних механизмов работы системы и опирается на внешние проявления ее работы. При этом тестировании проверяется поведение программного обеспечения при различных входных данных. В случае тестирования white box создаются тест-кейсы, основанные преимущественно на программном коде. Также существует расширенный тип black-box тестирования, включающего в себя изучение кода, – так называемый grey box (серый ящик).

Испытание клиентской части программы проводилось на аппаратных платформах, описание которых приведено в таблице 3.1.

Таблица 3.1 – Аппаратные платформы

№	Роль	Аппаратная конфигурация	Программная конфигурация
1	Сервер базы данных, веб-сервер	Intel Core i7 6500-U, 1Tb HDD, 8Gb RAM	Ubuntu 16.04, MySQL 5.7, Apache Tomcat 8.5
2	Клиент	Intel Core i7 6500-U, 1Tb HDD, 8Gb RAM	Ubuntu 16.04, браузер Google Chrome

В таблице 3.2 приведен набор тест-кейсов, на основе которых производилось тестирование.

Таблица 3.2 – Результаты тестирования программного средства

№	Тестовый вариант	Входные данные	Ожидаемый результат
1	Вход в систему	Правильные логин и пароль	Открытие главной страницы
2	Вход в систему с неправильными реквизитами	Несуществующая в БД комбинация логина и пароля	Возврат на страницу входа в систему
3	Добавление товара	Полные данные о товаре	Товар добавлен

Продолжение таблицы 3.2			
4	Добавление товара с неполными данными	Неполные данные о товаре	Сообщение об ошибке
5	Удаление товара	ID товара	Товар удален
6	Открытие страницы товара	ID товара	Открыта страница с товаром
7	Редактирование товара	Новые данные о товаре	Товар отредактирован
8	Добавление отзыва	Информация о товаре	Отзыв добавлен
9	Добавление отзыва с неполными данными	Неполная информация о товаре	Сообщение об ошибке
10	Удаление отзыва	ID отзыва	Отзыв удален
11	Добавление товара в корзину	ID товара	Товар добавлен
12	Удаление товара из корзины	ID товара	Товар удален
13	Регистрация пользователя с корректными данными	Личные данные пользователя	Пользователь добавлен
14	Регистрация пользователя, который уже зарегистрирован в системе	Личные данные пользователя	Пользователь не зарегистрирован. Сообщение об ошибке
15	Регистрация пользователя с неполными данными	Неполные личные данные пользователя	Сообщение об ошибке

В результате тестирования было устранено множество ошибок:

- отсутствие валидации вводимых пользователем данных. Для предотвращения ошибок из-за ввода пользователем некорректных данных были использованы стандартные средства HTML5.

- проблема с входом в систему из-за некорректного хэша пароля.

По результатам тестирования программное средство имеет надлежащее качество.

Тестирование сайта — процесс исследования, испытания программного продукта, имеющий две различные цели:

- продемонстрировать разработчикам и заказчикам, что сайт соответствует требованиям;

- выявить ситуации, в которых поведение программы является неправильным, нежелательным или не соответствующим спецификации.

Возможность возникновения ошибок в ходе работы программы сведена к минимуму.

## **4 ВНЕДРЕНИЕ И ЭКСПЛУАТАЦИЯ**

### **4.1 Назначение программного средства**

Электронный магазин — это прикладная система, построенная с использованием технологии системы электронной торговли.

Интернет-магазин объединяет элементы прямого маркетинга с образом посещения традиционного магазина.

Отличительной чертой интернет-магазинов по сравнению с обычной формой торговли является то, что интерактивный магазин может предложить значительно большее количество товаров и услуг, и обеспечить потребителей значительно большим объемом информации, необходимым для принятия решения о покупке. Кроме того, за счет использования компьютерных технологий возможна персонализация подхода к каждому из клиентов, исходя из истории его посещений магазина и сделанных ранее покупок.

Основные проблемы реализации интернет-магазина лежат на стыке технологий интернета и традиционной коммерческой деятельности. В обычной торговле покупатель привык к тому, что есть возможность оценить товар визуально, определить его качество и характеристики.

В электронной торговле он такой возможности лишен. Зачастую визуальной информации достаточно, но здесь вступают в действие эмоциональные и психологические факторы. Часто существуют проблемы с доставкой товаров, особенно если их цена невелика.

Разработанная система предназначена для продажи женских аксессуаров. Сайт обладает самыми необходимыми возможностями, простым и понятным интерфейсом, гибкостью настроек и удобством в использовании.

### **4.2 Среда функционирования, требования к аппаратному обеспечению**

Программа рассчитана на работу в любой операционной системе, при наличии браузера для отображения сайта.

Основными платформами для корректной работы с программой являются Windows, Linux и Mac OS.

Windows — семейство проприетарных операционных систем корпорации Microsoft, ориентированных на применение графического интерфейса при управлении. По состоянию на август 2014 года под управлением операционных систем семейства Windows по данным ресурса Net Applications работает около 89% персональных компьютеров.

Linux — семейство Unix-подобных операционных систем на базе ядра Linux, включающих тот или иной набор утилит и программ проекта GNU, и, возможно, другие компоненты. Как и ядро Linux, системы на его основе как правило создаются и распространяются в соответствии с моделью разработки свободного и открытого программного обеспечения. Linux-системы распространяются в основном бесплатно в виде различных дистрибутивов — в форме, готовой для установки и удобной для

сопровождения и обновлений, — и имеющих свой набор системных и прикладных компонентов, как свободных, так возможно и собственных.

Mac OS — семейство проприетарных операционных систем производства корпорации Apple. Разработана для линейки персональных компьютеров Macintosh. Популяризация графического интерфейса пользователя в современных операционных системах часто считается заслугой Mac OS.

Основными браузерами для корректной работы с программой являются Google Chrome, Firefox и Internet Explorer (Microsoft Edge).

Google — браузер, разрабатываемый компанией Google на основе свободного браузера Chromium и движка Blink. Исследование двух компаний NetMarketShare и StatCounter показало, что самым популярным браузером в мире стал Chrome.

Mozilla Firefox — свободный браузер на движке Gecko, разработкой и распространением которого занимается Mozilla Corporation. Второй по популярности браузер в мире и первый среди свободного ПО.

Internet Explorer — программа-браузер, которую разрабатывала корпорация Microsoft с 1995 по 2015 год. Входила в комплект операционных систем семейства Windows вплоть до Windows 10, где её сменил Microsoft Edge.

#### 4.3 Установка ПС

Программное средство не нуждается в установке. Пользователю достаточно установить любой из доступных браузеров и открыть страницу приложения. Для работы с программным средством пользователю требуется устройство с доступом в интернет и установленным на него браузером. Это может быть как настольный компьютер, так и смартфон или планшет.

#### 4.4 Руководство пользователя

Первая страница сайта — главная страница. Она изображена на рисунке 4.1.

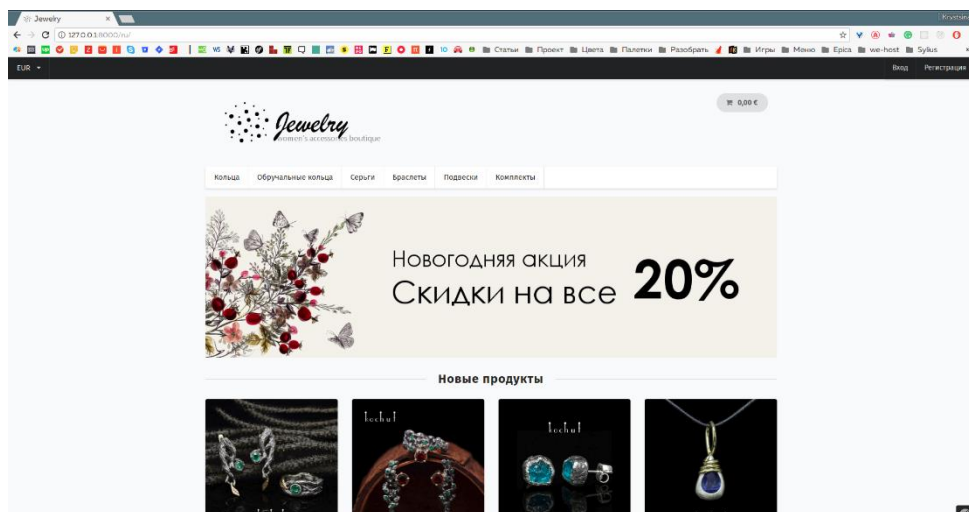


Рисунок 4.1 — Главная страница сайта

Для просмотра каталога товаров входить в систему не обязательно, но, если нужно получить расширенный доступ к функционалу сайта, необходимо пройти регистрацию.

Для входа либо регистрации следует нажать кнопку «Войти», которая находится в правом верхнем углу страницы. Страница авторизации показана на рисунке 4.2.

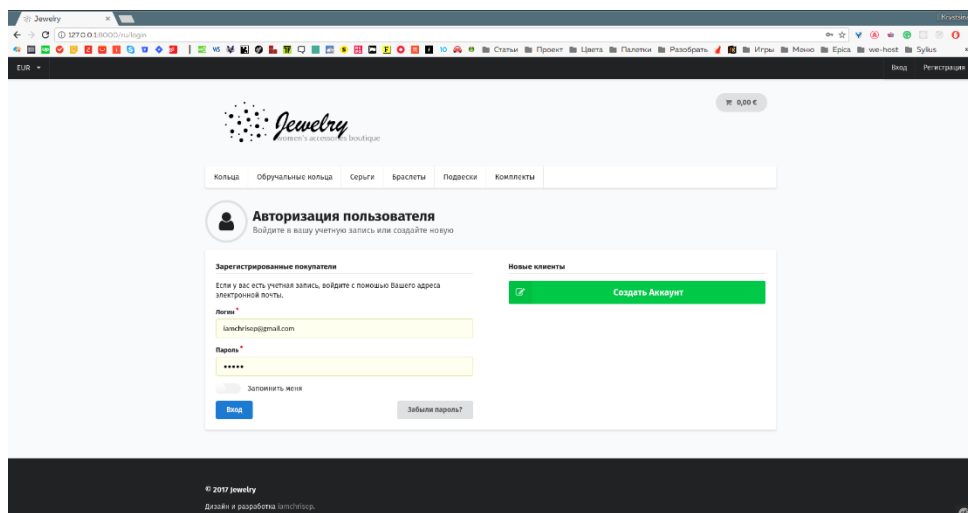


Рисунок 4.2 — Страница авторизации на странице

Страница регистрации показана на рисунке 4.3.

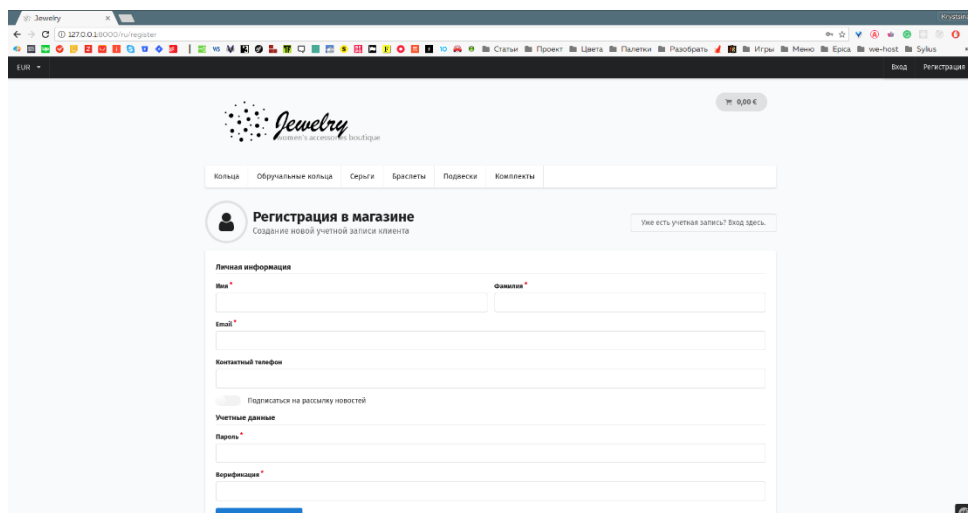


Рисунок 4.3 — Страница регистрации на странице

Для просмотра продукции сайта нажмите соответствующую кнопку интересующей категории товаров. Страница каталога сайта показана на рисунке 4.4.



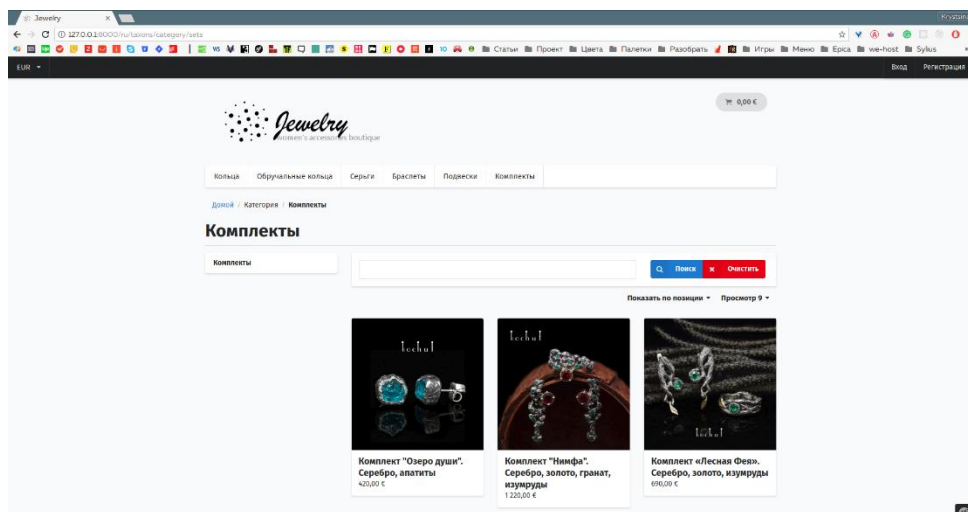


Рисунок 4.4 — Страница каталога

Для перехода на страницу товара, нажмите на кнопку «Посмотреть больше». Страница товара показана на рисунке 4.5.

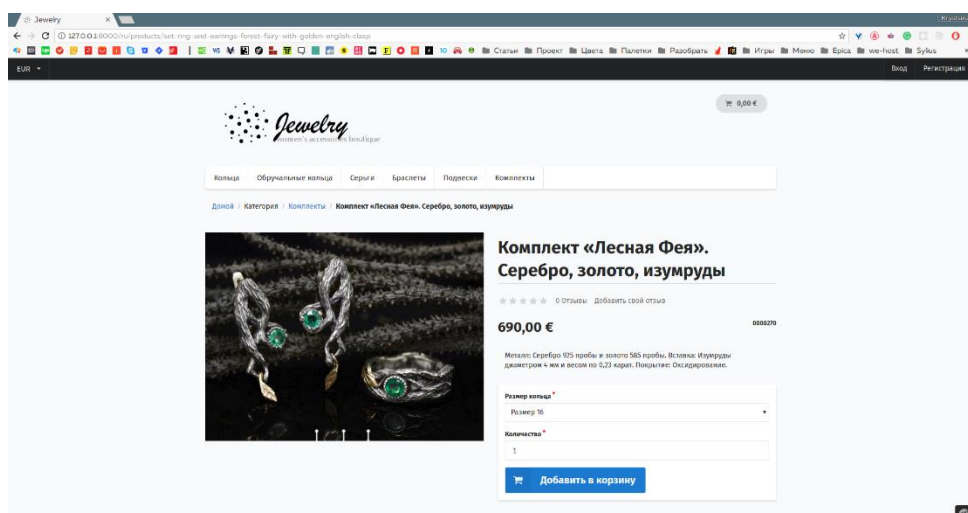


Рисунок 4.5 — Страница товара

Для оформления заказа следует нажать кнопку перехода в корзину в правом верхнем углу сайта. Страница корзины указана на рисунке 4.6.

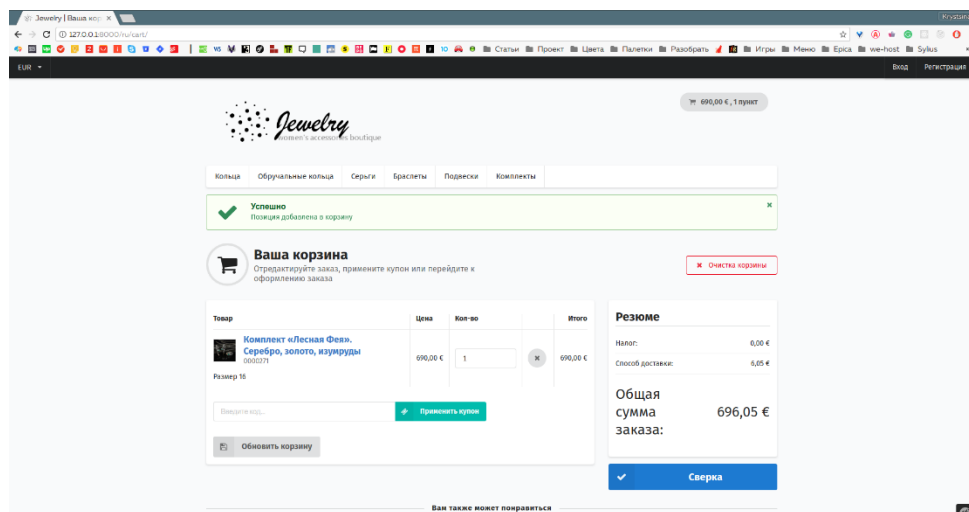


Рисунок 4.6 — Страница корзины

Функционал для администратора представлен отдельным личным кабинетом. Страница личного кабинета администратора указана на рисунке 4.7.

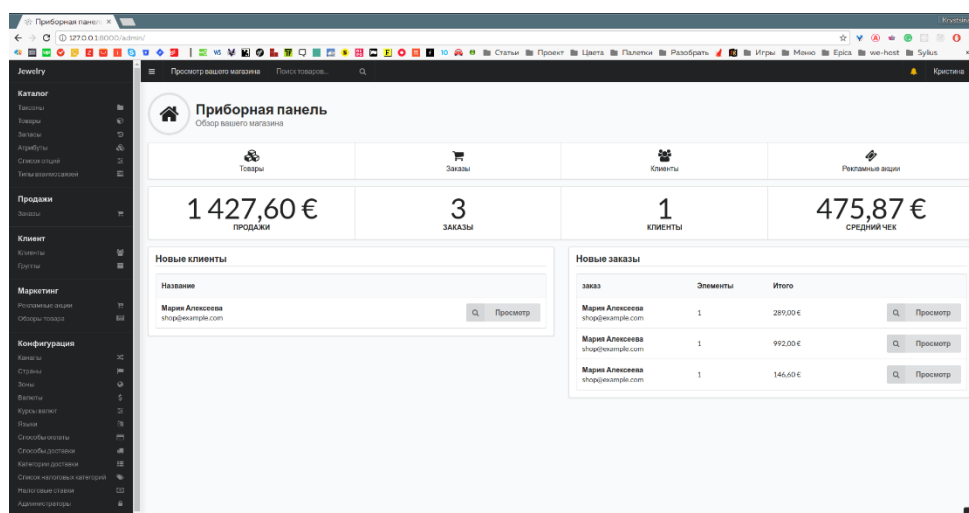


Рисунок 4.7 — Страница личного кабинета администратора

Интернет-магазин имеет удобный и понятный интерфейс, благодаря которому не требуется излишних усилий в освоении работы с сайтом. Вся информация не содержит ничего лишнего.

## ЗАКЛЮЧЕНИЕ

В процессе проектирования проекта была разработана система продажи женских аксессуаров, которая предназначена для продажи различных украшений через интернет. Областью применения данного Web-приложения выступает интернет-торговля.

Создание данного интернет-магазина позволит продвигать и продавать товары, уменьшить издержки на рекламу и приобрести дополнительный рынок сбыта своего товара, расширить клиентскую аудиторию и, как следствие, увеличение прибыли.

Web-приложение имеет понятный интерфейс, которым удобно пользоваться не только с настольного персонального компьютера, но и с мобильных устройств. Благодаря ему не требуется излишних усилий в освоении работы с сайтом. Вся информация на сайте не содержит ничего лишнего.

По итогу проделанной работы можно сказать, что все заявленные цели были успешно достигнуты, а именно:

- создание и удаление категорий для разного вида товаров;
- создание и удаление товаров для разных категорий;
- создание характеристик товаров для каждой категории;
- возможность оставлять отзывы под товарами;
- возможность расширения функционала (добавление способов оплаты и доставки, валют, языков сайта и прочего).

На этапе проектирования были разработаны множество функций, триггеров, генераторов и схема основного алгоритма.

В ходе тестирования было устранено множество недостатков, приводивших к сбоям и некорректной работе приложения.

В качестве дальнейшего развития приложения рассматривается доработка административного интерфейса, а также повышение удобства интерфейса.

Разработанное ПС реализовано с использованием языка программирования PHP и популярных фреймворков.

Программа реализована в соответствии с поставленной задачей и полностью протестирована.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1) Лебедев А. Руководство Лебедев А. — Москва.: Издательство Студии Артемия Лебедева, 2011. — 452 с.
- 2) Спейнауэр С. Справочник Web-мастера. СПб.: BHV, 1997. — 368 с.
- 3) Ратшиллер Т., Геркен Т. PHP4: разработка Web-приложений. — СПб: Питер, 2001. — 384 с.
- 4) Томсон Л., Веллинг Л. Разработка Web-приложений на PHP и MySQL. — К.: "ДиаСофт", 2001. — 672 с
- 5) Яргер Р., Риз Дж., Кинг Т. MySQL и mSQL. Базы данных для небольших предприятий и Интернета. — СПб: Символ-Плюс, 2000 — 560 с.
- 6) Орлов С.Д. Технологии разработки программного обеспечения. Учеб. пособие. 2-е изд. СПб.: Питер, 2003. — 480 с.
- 7) Иванова Г.С. Основы программирования: Учебник для вузов. — М.: Изд-во МГТУ им. Н.Э. Баумана, 2002. — 416 с.
- 8) Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы: построение и анализ. — М.: МЦНМО, 2009. — 960с.
- 9) Окулов С.М. Основы программирования. — 3-е изд. — М.: БИНОМ. Лаборатория знаний, 2006. — 440с.
- 10) Орлов С. Технологии разработки программного обеспечения. — СПб.:Питер, 2005
- 11) Shklar L., Rosen R. Web Application Architecture: Principles, Protocols and Practices. Wiley, 2009 — 440 p.
- 12) Welling L., Thomson L. PHP and MySQL Web Development (4th Edition). Addison-Wesley Professional, 2008 — 1008 p.
- 13) Дунаев В. В. HTML, скрипты и стили. СПб.: БХВ — Петербург, 2011 — 816с.
- 14) Мэтью Д. HTML5. Разработка веб-приложений. М.: Рид Групп, 2012 — 320с.
- 15) Никсон Р. Создаем динамические веб-сайты с помощью PHP, MySQL и JavaScript. СПб.: Питер, 2011 — 496 с.
- 16) Пьюривал С. Основы разработки веб-приложений. СПб: Питер, 2015 — 272
- 17) Закас Н. JavaScript для разработчиков. - СПб.: Питер, 2017. – 260с.
- 18) Фаронов В. В. JS. Учебный курс. - СПб.: Питер, 2002. – 260с.
- 19) Хорстманн К. Паттерны проектирования для профессионала. - М.: ТОО «Компьютер Пресс», 1999. – 347 с.
- 20) Шоу З. Легкий способ выучить JavaScript. - М.: ТОО «Эксмо», 2017. – 514 с.

## ПРИЛОЖЕНИЕ А

### «Фрагменты кода программного средства»

```
<?php

namespace Sylius\Bundle\ShopBundle\EventListener;

use Doctrine\Common\Persistence\ObjectManager;
use Sylius\Bundle\UserBundle\Security\UserLoginInterface;
use Sylius\Bundle\UserBundle\UserEvents;
use Sylius\Component\Channel\Context\ChannelContextInterface;
use Sylius\Component\Core\Model\ChannelInterface;
use Sylius\Component\Core\Model\CustomerInterface;
use Sylius\Component\Core\Model\ShopUserInterface;
use Sylius\Component\User\Security\Generator\GeneratorInterface;
use Symfony\Component\EventDispatcher\EventDispatcherInterface;
use Symfony\Component\EventDispatcher\GenericEvent;
use Webmozart\Assert\Assert;

final class UserRegistrationListener
{
    /**
     * @var ObjectManager
     */
    private $userManager;

    /**
     * @var GeneratorInterface
     */
    private $tokenGenerator;

    /**
     * @var EventDispatcherInterface
     */
    private $eventDispatcher;

    /**
     * @var ChannelContextInterface
     */
    private $channelContext;

    /**
     * @var UserLoginInterface
     */
    private $userLogin;

    /**
     * @var string
     */
    private $firewallContextName;

    /**
```



```

* @param ObjectManager $userManager
* @param GeneratorInterface $tokenGenerator
* @param EventDispatcherInterface $eventDispatcher
* @param ChannelContextInterface $channelContext
* @param UserLoginInterface $userLogin
* @param string $firewallContextName
*/
public function __construct(
    ObjectManager $userManager,
    GeneratorInterface $tokenGenerator,
    EventDispatcherInterface $eventDispatcher,
    ChannelContextInterface $channelContext,
    UserLoginInterface $userLogin,
    $firewallContextName
) {
    $this->userManager = $userManager;
    $this->tokenGenerator = $tokenGenerator;
    $this->eventDispatcher = $eventDispatcher;
    $this->channelContext = $channelContext;
    $this->userLogin = $userLogin;
    $this->firewallContextName = $firewallContextName;
}

/**
* @param GenericEvent $event
*/
public function handleUserVerification(GenericEvent $event)
{
    $customer = $event->getSubject();
    Assert::assertInstanceOf($customer, CustomerInterface::class);

    $user = $customer->getUser();
    Assert::notNull($user);

    /** @var ChannelInterface $channel */
    $channel = $this->channelContext->getChannel();
    if (!$channel->isAccountVerificationRequired()) {
        $this->enableAndLogin($user);
    }

    $this->sendVerificationEmail($user);
}

/**
* @param ShopUserInterface $user
*/
private function sendVerificationEmail(ShopUserInterface $user)
{
    $token = $this->tokenGenerator->generate();
    $user->setEmailVerificationToken($token);

    $this->userManager->persist($user);
}

```

```

        $this->userManager->flush();

        $this->eventDispatcher->
>dispatch(UserEvents::REQUEST_VERIFICATION_TOKEN, new
GenericEvent($user));
    }

    /**
     * @param ShopUserInterface $user
     */
    private function enableAndLogin(ShopUserInterface $user)
    {
        $user->setEnabled(true);

        $this->userLogin->login($user, $this->firewallContextName);
    }
}

<?php

namespace Sylius\Bundle\ShopBundle\EventListener;

use Sylius\Bundle\ResourceBundle\Event\ResourceControllerEvent;
use Sylius\Component\Core\Model\OrderInterface;
use Sylius\Component\Order\SyliusCartEvents;
use
Sylius\Component\Promotion\Checker\Eligibility\PromotionEligibilityChe
ckerInterface;
use Symfony\Component\EventDispatcher\EventDispatcherInterface;
use Symfony\Component\EventDispatcher\GenericEvent;
use Symfony\Component\HttpFoundation\RedirectResponse;
use Symfony\Component\Routing\RouterInterface;
use Webmozart\Assert\Assert;

final class OrderPromotionIntegrityChecker
{
    /**
     * @var PromotionEligibilityCheckerInterface
     */
    private $promotionEligibilityChecker;

    /**
     * @var EventDispatcherInterface
     */
    private $eventDispatcher;

    /**
     * @var RouterInterface
     */
    private $router;

```

```

    /**
     * @param PromotionEligibilityCheckerInterface
    $promotionEligibilityChecker
     * @param EventDispatcherInterface $eventDispatcher
     * @param RouterInterface $router
     */
    public function __construct(
        PromotionEligibilityCheckerInterface
    $promotionEligibilityChecker,
        EventDispatcherInterface $eventDispatcher,
        RouterInterface $router
    ) {
        $this->promotionEligibilityChecker =
    $promotionEligibilityChecker;
        $this->eventDispatcher = $eventDispatcher;
        $this->router = $router;
    }

    /**
     * @param ResourceControllerEvent $event
     */
    public function check(ResourceControllerEvent $event)
    {
        /** @var OrderInterface $order */
        $order = $event->getSubject();

        Assert::assertInstanceOf($order, OrderInterface::class);

        $promotions = $order->getPromotions();
        foreach ($promotions as $promotion) {
            if (!$this->promotionEligibilityChecker-
>isEligible($order, $promotion)) {
                $event->stop(
                    'sylius.order.promotion_integrity',
                    ResourceControllerEvent::TYPE_ERROR,
                    ['%promotionName%' => $promotion->getName()]
                );

                $event->setResponse(new RedirectResponse($this-
>router->generate('sylius_shop_checkout_complete')));

                $this->eventDispatcher-
>dispatch(SyliusCartEvents::CART_CHANGE, new GenericEvent($order));

                break;
            }
        }
    }
}

```

```

<?php

namespace Sylius\Bundle\ShopBundle\Menu;

use Knp\Menu\FactoryInterface;
use Knp\Menu\ItemInterface;
use Sylius\Bundle\UiBundle\Menu\Event\MenuBuilderEvent;
use Symfony\Component\EventDispatcher\EventDispatcherInterface;

final class AccountMenuBuilder
{
    const EVENT_NAME = 'sylius.menu.shop.account';

    /**
     * @var FactoryInterface
     */
    private $factory;

    /**
     * @var EventDispatcherInterface
     */
    private $eventDispatcher;

    /**
     * @param FactoryInterface $factory
     * @param EventDispatcherInterface $eventDispatcher
     */
    public function __construct(FactoryInterface $factory,
EventDispatcherInterface $eventDispatcher)
    {
        $this->factory = $factory;
        $this->eventDispatcher = $eventDispatcher;
    }

    /**
     * @param array $options
     *
     * @return ItemInterface
     */
    public function createMenu(array $options)
    {
        $menu = $this->factory->createItem('root');
        $menu->setLabel('sylius.menu.shop.account.header');

        $menu
            ->addChild('dashboard', ['route' =>
'sylius_shop_account_dashboard'])
            ->setLabel('sylius.menu.shop.account.dashboard')
            ->setLabelAttribute('icon', 'home')
        ;
        $menu
    }
}

```

```

        ->addChild('personal_information', ['route' =>
'sylius_shop_account_profile_update'])
    -
>setLabel('sylius.menu.shop.account.personal_information')
    ->setLabelAttribute('icon', 'user')
    ;
    $menu
        ->addChild('change_password', ['route' =>
'sylius_shop_account_change_password'])
        ->setLabel('sylius.menu.shop.account.change_password')
        ->setLabelAttribute('icon', 'lock')
    ;
    $menu
        ->addChild('address_book', ['route' =>
'sylius_shop_account_address_book_index'])
        ->setLabel('sylius.menu.shop.account.address_book')
        ->setLabelAttribute('icon', 'book')
    ;
    $menu
        ->addChild('order_history', ['route' =>
'sylius_shop_account_order_index'])
        ->setLabel('sylius.menu.shop.account.order_history')
        ->setLabelAttribute('icon', 'cart')
    ;

    $this->eventDispatcher->dispatch(self::EVENT_NAME, new
MenuBuilderEvent($this->factory, $menu));

    return $menu;
}
}

```

```

<div class="ui basic segment">
    <div class="ui three column stackable grid">
        <div class="column">
            <a href="{{ path('sylius_shop_homepage') }}"></a>
        </div>
        <div class="column">
            {{ sonata_block_render_event('sylius.shop.layout.header')
}}
        </div>
        <div class="right aligned column">
            {{ render(url('sylius_shop_partial_cart_summary',
{'template': '@SyliusShop/Cart/_widget.html.twig'})) }}
        </div>
    </div>
</div>

```

```

<footer id="footer" class="ui inverted vertical footer segment">
    <div class="ui container">
        <div class="ui inverted divided equal height stackable grid">
            {{
sonata_block_render_event('sylius.shop.layout.before_footer') }}

                <div class="eight wide column">
                    <h4 class="ui inverted header">&copy; 2017
Jewelry</h4>
                    <p>P" PëP·P°PM°PS Pë CßP°P·CßP°P±PsC,PeP° <a
href="https://www.upwork.com/o/profiles/users/_~015cecelecfaa14a9d/"
target="_blank" style="color: #999999;">iamchrise</a>.</p>
                    <p>P'PSPëPjP°PSPëPµ! P"P°PSPSC<PM° PìCßPsPµPeC,
CµPIP»CµPµC,CÍCµ CíC±PµP±PSC<Pj. P'CÍPµ PëCÍPìPsP»CßP·CíPµPjC<Pµ
PëP·PsP±CßP°P¶PµPSPëCµ Pë C,PµPeCÍC, PI PsPìPëCÍP°PSPëPë
C,PsPIP°CßPsPI PìCßPëPSP°PrP»PµP¶P°C, PëPSC,PµCßPSPµC,-
PjP°PiP°P·PëPSCí <a href="https://kochut.org/" target="_blank"
style="color: #999999;">Kochut</a>.</p>
                </div>

            {{
sonata_block_render_event('sylius.shop.layout.after_footer') }}
        </div>
    </div>
</footer>

{% extends '@SyliusShop/layout.html.twig' %}

{% block content %}
<div class="ui hidden divider"></div>
<div class="ui one column grid banner">
    
</div>
<div class="ui hidden divider"></div>
<h2 class="ui horizontal section divider header">
    {{ 'sylius.ui.latest_products'|trans }}
</h2>
{{ render(url('sylius_shop_partial_product_index_latest', {'count': 8,
'template': '@SyliusShop/Product/_horizontalList.html.twig'}))) }}
{% endblock %}

<?php
namespace Sylius\Bundle\ShopBundle\Controller;

```



```

use Symfony\Bundle\FrameworkBundle\Templating\EngineInterface;
use Symfony\Component\HttpFoundation\Request;
use Symfony\Component\HttpFoundation\Response;

final class HomepageController
{
    /**
     * @var EngineInterface
     */
    private $templatingEngine;

    /**
     * @param EngineInterface $templatingEngine
     */
    public function __construct(EngineInterface $templatingEngine)
    {
        $this->templatingEngine = $templatingEngine;
    }

    /**
     * @param Request $request
     *
     * @return Response
     */
    public function indexAction(Request $request)
    {
        return $this->templatingEngine-
>renderResponse('@SyliusShop/Homepage/index.html.twig');
    }
}

```

<?php

```

namespace Sylius\Bundle\ShopBundle\Controller;

use Sylius\Bundle\ShopBundle\Locale\LocaleSwitcherInterface;
use Sylius\Component\Locale\Context\LocaleContextInterface;
use Sylius\Component\Locale\Provider\LocaleProviderInterface;
use Symfony\Bundle\FrameworkBundle\Templating\EngineInterface;
use Symfony\Component\HttpFoundation\Request;
use Symfony\Component\HttpFoundation\Response;
use Symfony\Component\HttpKernel\Exception\HttpException;

final class LocaleSwitchController
{
    /**
     * @var EngineInterface
     */
    private $templatingEngine;

```

```

/**
 * @var LocaleContextInterface
 */
private $localeContext;

/**
 * @var LocaleProviderInterface
 */
private $localeProvider;

/**
 * @var LocaleSwitcherInterface
 */
private $localeSwitcher;

/**
 * @param EngineInterface $templatingEngine
 * @param LocaleContextInterface $localeContext
 * @param LocaleProviderInterface $localeProvider
 * @param LocaleSwitcherInterface $localeSwitcher
 */
public function __construct(
    EngineInterface $templatingEngine,
    LocaleContextInterface $localeContext,
    LocaleProviderInterface $localeProvider,
    LocaleSwitcherInterface $localeSwitcher
) {
    $this->templatingEngine = $templatingEngine;
    $this->localeContext = $localeContext;
    $this->localeProvider = $localeProvider;
    $this->localeSwitcher = $localeSwitcher;
}

/**
 * @return Response
 */
public function renderAction()
{
    return $this->templatingEngine-
>renderResponse('@SyliusShop/Menu/_localeSwitch.html.twig', [
        'active' => $this->localeContext->getLocaleCode(),
        'locales' => $this->localeProvider-
>getAvailableLocalesCodes(),
    ]);
}

/**
 * @param Request $request
 * @param string $code
 *
 * @return Response
 */

```

```

public function switchAction(Request $request, $code = null)
{
    if (null === $code) {
        $code = $this->localeProvider->getDefaultLocaleCode();
    }

    if (!in_array($code, $this->localeProvider-
>getAvailableLocalesCodes(), true)) {
        throw new HttpException(
            Response::HTTP_NOT_ACCEPTABLE,
            sprintf('The locale code "%s" is invalid.', $code)
        );
    }

    return $this->localeSwitcher->handle($request, $code);
}
}

```

<?php

```

namespace Sylius\Bundle\ShopBundle\Controller;

use Symfony\Bundle\FrameworkBundle\Templating\EngineInterface;
use Symfony\Component\HttpFoundation\Response;

final class SecurityWidgetController
{
    /**
     * @var EngineInterface
     */
    private $templatingEngine;

    /**
     * @param EngineInterface $templatingEngine
     */
    public function __construct(EngineInterface $templatingEngine)
    {
        $this->templatingEngine = $templatingEngine;
    }

    /**
     * @return Response
     */
    public function renderAction()
    {
        return $this->templatingEngine-
>renderResponse('@SyliusShop/Menu/_security.html.twig');
    }
}

```

```

<?php

namespace Sylius\Bundle\ShopBundle\Controller;

use Sylius\Component\Channel\Context\ChannelContextInterface;
use Sylius\Component\Core\Currency\CurrencyStorageInterface;
use Sylius\Component\Core\Model\ChannelInterface;
use Sylius\Component\Currency\Context\CurrencyContextInterface;
use Sylius\Component\Currency\Model\CurrencyInterface;
use Symfony\Bundle\FrameworkBundle\Templating\EngineInterface;
use Symfony\Component\HttpFoundation\RedirectResponse;
use Symfony\Component\HttpFoundation\Request;
use Symfony\Component\HttpFoundation\Response;

final class CurrencySwitchController
{
    /**
     * @var EngineInterface
     */
    private $templatingEngine;

    /**
     * @var CurrencyContextInterface
     */
    private $currencyContext;

    /**
     * @var CurrencyStorageInterface
     */
    private $currencyStorage;

    /**
     * @var ChannelContextInterface
     */
    private $channelContext;

    /**
     * @param EngineInterface $templatingEngine
     * @param CurrencyContextInterface $currencyContext
     * @param CurrencyStorageInterface $currencyStorage
     * @param ChannelContextInterface $channelContext
     */
    public function __construct(
        EngineInterface $templatingEngine,
        CurrencyContextInterface $currencyContext,
        CurrencyStorageInterface $currencyStorage,
        ChannelContextInterface $channelContext
    ) {
        $this->templatingEngine = $templatingEngine;
        $this->currencyContext = $currencyContext;
    }
}

```

```

        $this->currencyStorage = $currencyStorage;
        $this->channelContext = $channelContext;
    }

    /**
     * @return Response
     */
    public function renderAction()
    {
        /** @var ChannelInterface $channel */
        $channel = $this->channelContext->getChannel();

        $availableCurrencies = array_map(
            function (CurrencyInterface $currency) {
                return $currency->getCode();
            },
            $channel->getCurrencies()->toArray()
        );

        return $this->templatingEngine-
>renderResponse('@SyliusShop/Menu/_currencySwitch.html.twig', [
            'active' => $this->currencyContext->getCurrencyCode(),
            'currencies' => $availableCurrencies,
        ]);
    }

    /**
     * @param Request $request
     * @param string $code
     *
     * @return Response
     */
    public function switchAction(Request $request, $code)
    {
        /** @var ChannelInterface $channel */
        $channel = $this->channelContext->getChannel();

        $this->currencyStorage->set($channel, $code);

        return new RedirectResponse($request->headers->get('referer',
        $request->getSchemeAndHttpHost()));
    }
}

```

**ПРИЛОЖЕНИЕ Б**  
**«Общая блок-схема»**



**ПРИЛОЖЕНИЕ В**  
«Блок-схема алгоритма авторизации»