Министерство образования и науки Российской Федерации
ФГБОУ ВО «Уральский государственный педагогический университет»
Институт математики, физики, информатики и технологий
Кафедра информатики, информационных технологий и методики обучения информатике

ВЕБ-СЕРВИС ДЛЯ ОРГАНИЗАЦИИ ОНЛАЙН-ПРОДАЖ И ДОСТАКИ ТОВАРОВ

Выпускная квалификационная работа

Квалификационная работа
допущена к защите
Зав. кафедрой

_____ _____
дата подпись

Исполнитель:
Андрианов Андрей Александрович
студент группы ПИвС-1501z

_____
подпись

Руководитель:
Арбузов Сергей Сергеевич
доцент кафедры информационно-коммуникационных технологий в образовании

_____
подпись

Екатеринбург – 2020

Оглавление

Введение

Современный Интернет является неотъемлемой частью человеческой жизни во многих её сферах. Благодаря тому, что Интернет распространён почти по всему миру, пользователи имеют возможность выгодно и быстро решать многие задачи, решение которых, при обычных условиях, заняло бы значительно большее количество времени и сил. Однако Интернет неоднороден – он сочетает в себе множество платформ, созданных на основе отличающихся друг от друга технологий, а поиск информации осуществляется в различных разрозненных источниках. В связи с этим существует тенденция создания способа выведения информации таким образом, чтобы она была удобным образом упорядочена и была пригодна для дальнейшей её обработки. Одним из плодов данной тенденции является концепция веб-сервисов (Web Services), суть которой заключается в объединении и упорядочивании информации, а также интеграции разнородных систем, расширяющих возможности сервиса. Сегодняшний Интернет не может быть представлен без веб-сервисов.

Предмет разработки: веб-сервис для организации онлайн-продаж и доставки товаров.

Цель работы: спроектировать и разработать интернет-магазин.

Для выполнения этой работы необходимо решить следующие задачи:

1. Изучить базовые принципы разработки веб-сервисов.
2. Рассмотреть современные системы разработки веб-сервисов.
3. Подготовить техническое задание для разработки онлайн-магазина по продаже одежды.
4. В соответствии с техническим заданием спроектировать и разработать онлайн-магазин для продажи одежды.

Глава 1. Теоретические основы создания веб-сервисов

1.1. Теория разработки веб-сервисов для организации онлайн-продаж

Сегодня Интернет является связующим звеном для огромного количества людей, живущих на планете. С помощью Интернета люди, находясь на огромном расстоянии друг от друга, могут осуществлять между собой различные контакты, суть которых может заключаться в общении, выполнении совместной работы, осуществлении торговых сделок и т. п. Помимо различных социальных функций, Интернет также осуществляет функцию огромного вместилища цифровой информации, которая может быть представлена в виде текста, изображения, цифрового аудио, программы – такая функция позволяет людям осуществлять хранение в Интернете, например, паспортных данных, фотографий, денежных сбережений в электронном виде и многого другого.

В развитых странах Интернет уже давно стал неотъемлемой частью человеческой деятельности. Осуществление поиска необходимой информации, находясь при этом дома, на работе или в дороге, является лишь одной из множества возможностей, которые предоставляет Интернет, дабы облегчить жизнь людям, вне зависимости от их рода деятельности. Помимо бытовой информации, Интернет также осуществляет хранение информации более высокого уровня – документов государственного характера, результатов научных исследований, курсов валют – проще говоря, Интернет играет важную роль не только в жизни отдельно взятого человека, но и в работе целых компаний, структур и даже государств.

Ключевым элементом сетевого пространства, без которого невозможно использовать Интернет, является веб-сервис. Данное понятие имеет множество различных значений, однако чаще всего под веб-сервисом подразумевается программная система со стандартизированными интерфейсами, отображаемая пользовательским браузером в виде набора HTML-страниц [11, с. 14]. Отличие от веб-сайта заключается в том, что веб-сервис служит для оказания

пользователю некой услуги: веб-сервис предоставляет пользователю возможность осуществить непосредственный заказ товара или услуги (например, оформить покупку товара с последующей доставкой на дом или по почте), в то время как на веб-сайте располагается лишь информация, несущая ознакомительный характер.

Качественный веб-сервис отличается не только уровнем обслуживания – не менее важными показателями такого сервиса являются дизайн и навигация. Высокий уровень дизайна подразумевает приятное для глаз визуальное оформление веб-страниц, благодаря которому обеспечивается привлекательность веб-сервиса, а хорошо организованная навигация помогает пользователю не только хорошо ориентироваться при перемещении между страницами, но и эффективнее пользоваться предоставляемыми веб-сервисом услугами[4].

Современные технологии создания веб-сервисов направлены, в первую очередь, на улучшение процесса взаимодействия пользователя с системой. Сегодня существует множество систем для создания веб-сервиса. Несмотря на функциональность современных систем, для создания веб-сервиса необходимо проделать некоторую работу вне электронной среды, например, придумать дизайн. В целом, разработка веб-сервиса делится на несколько этапов: планирование, реализация, тестирование и публикация [28].

На этапе планирования формируется назначение веб-сервиса: определяется его цель, круг пользователей, для которых предназначен сервис, а также информация, размещаемая на его страницах. В ходе планирования структуры сайта определяется количество категорий, разделов, подразделов, а также численность страниц внутри них. Планируется дизайн веб-страниц сайта.

В начале этапа реализации происходит подготовка материала, необходимого для наполнения сайта (текст, изображения). Разрабатываются веб-страницы в соответствии с их назначением, происходит связывание страниц ссылками согласно навигационной схеме сайта. Реализуется дизайн, определённый на предыдущем этапе. Создаются версии сайта для устройств с раз-

личными операционными системами и разрешениями экрана, а также, если необходимо, создаются версии для людей с ограниченными возможностями восприятия информации.

Тестирование – этап, на котором производится поиск всевозможных технических ошибок, связанных с работой сайта, с их последующим устранением. Происходит проверка отображения сайта на различных версиях программного языка, на котором написан программный код сайта, так как не все браузеры одинаково отображают одну и ту же веб-страницу.

Публикация является конечным этапом разработки, в ходе которого сайт размещается на выделенном сервере, что в дальнейшем позволит удалённому пользователю увидеть в сети Интернет готовый сайт и воспользоваться услугами, представленными на нём.

Большинство современных веб-сервисов имеют возможность взаимодействовать друг с другом и со сторонними приложениями при помощи определённых стандартизированных протоколов (SOAP, XML-RPC) и соглашений (REST). Данное свойство делает любой современный веб-сервис составной частью так называемой *сервис-ориентированной архитектуры*. Согласно определению, приведённому в спецификации OASIS RM-SOA 1.0, сервис-ориентированная архитектура (SOA, англ. service-oriented architecture) – это парадигма организации и использования распределённых информационных ресурсов, которые могут принадлежать различным владельцам [28]. Сервис-ориентированной архитектурой также называется подход к разработке программного обеспечения, суть которого заключается в использовании модульных компонентов со стандартизированными интерфейсами. Модульность этих компонентов заключается в том, что они могут быть распределёнными между собой и слабо связанными, а в случае необходимости могут быть легко заменены другими компонентами, осуществляющими взаимодействие по аналогичному протоколу [19, 27].

Архитектурой программы называется совокупность программных элементов, их внешних свойств и взаимосвязей [26]. Проектирование архитектуры программы является одной из наиболее важных задач разработчика, так как от её качества зависит множество параметров, которые в совокупности определяют качество готового программного продукта [32, с. 27]. Программная архитектура определяет границы реализации программного обеспечения, содержит в себе информацию о первичных проектных решениях и организационной структуре системы. Также качественно спроектированная архитектура программы способствует более тщательному планированию выделения необходимых для разработки ресурсов. Любая программная система базируется на структурах, содержащихся в её программной архитектуре. Однако архитектура даёт лишь общее представление о программной системе, показывая только взаимоотношение программных блоков и принцип их взаимодействия друг с другом [35].

Хотя программная архитектура и даёт лишь абстрактное представление о системе, она оказывает влияние на множество факторов, к которым относятся производительность программной системы, её функциональность, надёжность и т.д. Однако всё большую востребованность приобретает такой параметр как способность программных систем интегрироваться друг с другом. Интеграция программных систем – это объединение двух или более программных систем с целью взаимного увеличения качества их работы. Данный способ работы с программными системами подразумевает, что информация, введённая пользователем в одну систему, так же оказывается и в другой системе, интегрированной с первой. Интеграция систем позволяет расширить возможности обработки информации, а в некоторых случаях – модифицировать уже имеющуюся программную систему [7].

Существуют стандарты и правила, следование которым гарантирует создание качественной и надёжной программной архитектуры, в то время как их несоблюдение, вероятнее всего, приведёт к тому, что развитие архитекту-

ры будет осуществляться крайне неэффективно. Основываясь на стандарте ISO/IEC/IEEE 42010 [3, 31], к критериям качественной архитектуры можно отнести такие параметры как: эффективность, гибкость, расширяемость и масштабируемость процесса разработки [20].

Эффективность – параметр, отвечающий за общую работоспособность программной системы. Надёжность процессов выполнения поставленных задач, скорость их выполнения, выдерживание нагрузок – это характеристики, по которым оценивается эффективность архитектуры;

Гибкостью называется способность системы претерпевать изменения с наименьшим влиянием на другие её элементы, не требующие изменений, и наименьшим количеством ошибок, возникающих в результате этого изменения. В определённый момент времени у развивающейся программной системы неизбежно возникает необходимость в некоторых точечных изменениях её элементов, однако подобные изменения могут сопровождаться возникновением необходимости изменения других программных элементов, связанных с изменяемым изначально. Показатель гибкости программной архитектуры тем выше, чем меньше подобные изменения влияют на другие элементы системы;

Расширяемость характеризуется способностью системы к добавлению новых функций без ущерба для её основной структуры. Возможность добавления новых функций без вмешательства в основной код программы является одним из принципов объектно-ориентированного программирования (SOLID – single responsibility, open-closed, Liskov substitution, interface segregation и dependency inversion), который гласит, что «программные сущности должны быть открыты для расширения, но закрыты для модификации»;

Масштабируемость процесса разработки – это возможность добавления к процессу разработки других людей. К разработке системы, легкодоступной для понимания других людей, проще подключить сторонних разработчиков и равномерно распределить между ними процесс разработки.

Формированием мировых стандартов в области информационно-коммуникационных технологий занимаются несколько международных организаций, к которым относятся: Консорциум всемирной паутины (WWW – World Wide Web Consortium), Организация разработчиков Java (JCP – Java Community Process), Институт программной инженерии (SEI – Software Engineering Institute), Институт инженеров электротехники и электроники (IEEE – Institute of Electrical and Electronics Engineers), Object Management Group (OMG) и другие.

В программной основе современных веб-сервисов лежит сервис-ориентированная архитектура (SOA – Service-Oriented Architecture) – подход к разработке программного обеспечения, основанный на использовании сервисов со стандартизированными интерфейсами [19].

Сегодня эталонной моделью сервис-ориентированной архитектуры является SOA-RM (Reference Model for Service-Oriented Architecture), владельцем которой является организация по стандартизации OASIS. Данная модель была утверждена в октябре 2006 года как единая эталонная модель сервис-ориентированной архитектуры [29]. До этого момента стандартного определения SOA не существовало. SOA-RM даёт абстрактное представление о значимых объектах программной системы и методов их взаимодействия между собой, а также содержит инструкцию по разработке согласованных стандартов и спецификаций, поддерживающих сервис-ориентированную среду. Данная модель вносит в среду SOA чёткую техническую терминологию, необходимую для разработчиков, а структурированная информация о сервис-ориентированной архитектуре позволяет доступнее обучить или объяснить концепции SOA. Технический комитет OASIS отмечает, что SOA-RM не имеет отношения к технологиям и стандартам других отдельно взятых реализаций SOA.

В основе современных веб-сервисов лежит использование международных Интернет-стандартов. Благодаря тому, что они не затрагивают кон-

кретно способы реализации SOA-приложений, а лишь определяют протоколы их работы, компании-разработчики не могут посредством собственных разработок оказывать влияние на действующие международные стандарты. Работа веб-сервисов основана на нескольких основных стандартах: SOAP (Simple Object Access Protocol – простой протокол доступа к объектам) – протокол для обмена произвольными сообщениями в формате XML; WSDL (Web Services Description Language – язык описания веб-сервисов) – язык описания программных интерфейсов веб-сервисов на базе XML; UDDI (Universal Description Discovery and Integration) – инструмент для индексации веб-сервисов [21].

Для связи между веб-сервисом и пользователем используется обмен сообщениями в формате XML, осуществляемый по протоколу SOAP. Данный протокол обеспечивает расширение некоторых прикладных протоколов по типу HTTP, HTTPS, FTP, SMTP и др., однако чаще всего SOAP применяется в отношении HTTP [30]. Обмен информацией по протоколу SOAP между информационными системами происходит посредством обмена структурированными XML-сообщениями. От обычного обмена он отличается тем, что информация, передаваемая в формате XML, особым образом структурируется, что позволяет обеспечить обмен информацией между системами, изначально не имеющими средств для связи друг с другом.

Структура SOAP содержит в себе три основных элемента: Envelope, Header, Body (см. рис. 1.1.).

Envelope – корневой элемент, определяющий сообщение и пространство имён, использованное в документе.

Header – содержит атрибуты сообщения.

Body – содержит сообщение, предназначенное для обмена.

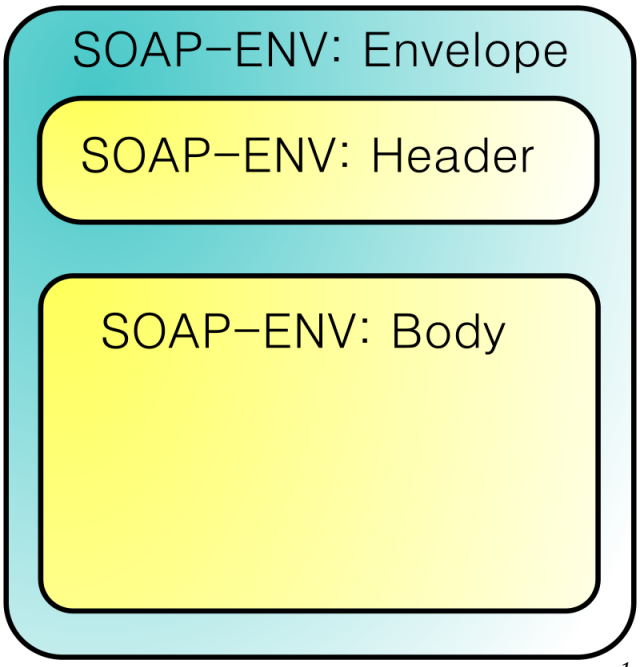


Рис. 1.1. Структура SOAP-сообщения[1]

SOAP является одной из наиболее важных составляющих технологии веб-сервисов, так как осуществляет перемещение данных по сети. SOAP поддерживает общий протокол обмена данными между отправителем и получателем документов XML, этим самым обеспечивая между ними эффективную связь. SOAP обеспечивает соединение по однонаправленному каналу, по которому осуществляется согласованная передача данных со стороны отправителя в сторону получателя. В случае необходимости на пути соединения могут находиться посредники, задача которых состоит в обработке или дополнении проходящей через них информации.

Для того чтобы клиент мог связаться и взаимодействовать с веб-сервисом, со стороны сервиса предоставляется WSDL-документ, содержащий в себе информацию и описания, необходимые для работы с веб-сервисом. WSDL – это формат, основанный на языке XML, задача которого состоит в описании синтаксиса веб-сервисов при помощи сообщений, в которых содержится инструкция по получению доступа к функциям конкретного веб-

1 https://upload.wikimedia.org/wikipedia/commons/thumb/5/59/SOAP.svg/800px-SOAP.svg.png

сервиса [23]. Так как язык WSDL строго формализован, в большинстве случаев используется возможность автоматического создания WSDL-документов, которая реализуется с помощью специальных утилит, идущих в комплекте с программными инструментами веб-разработки, однако также присутствует возможность написать этот документ самостоятельно.

Поиск необходимого веб-сервиса является первоочередной задачей для клиента, имеющего некую потребность, но не имеющего представления о веб-сервисе, который её удовлетворит. Существуют программные системы, осуществляющие поиск веб-сервисов в специальных реестрах, объединённые под названием UDDI. Реестр UDDI содержит в себе контактную и техническую информацию о веб-сервисах и их поставщиках, WSDL-документы, а также позволяет осуществлять поиск нужного веб-сервиса по различным критериям. Также с помощью UDDI организации способны интегрировать в свои системы веб-сервисы чужой разработки [34].

В структуру UDDI входят три основных компонента [18]: белые страницы, жёлтые страницы и зелёные страницы.

Белые страницы содержат в себе контактную информацию о поставщиках веб-сервиса, к которой относятся название компании или описание услуги. По этой информации можно найти сервис с имеющимися о нём сведениями.

Жёлтые страницы содержат в себе информацию о самих веб-сервисах и их кодовых числах, классифицирующих их деятельность. Компания, предоставляющая несколько услуг, может содержать несколько жёлтых страниц, описывающих каждый из этих сервисов.

Зелёные страницы содержат в себе техническую информацию о веб-сервисах – адрес веб-сервиса, ссылки на спецификации интерфейсов, WSDL-документы и др.

Реестр UDDI расположен на выделенном сервере, содержимое которого хранится в виде нескольких резервных копий в специальных узлах UDDI – серверах, оказывающих перманентную поддержку спецификации UDDI, зарегист-

рированных в реестре UDDI, пока он содержит как минимум одну запись. Примечательно то, что, будучи поставщиком услуг по поиску и индексации веб-сервисов, реестр UDDI в широком смысле сам является веб-сервисом.

Одним из наиболее серьёзных недостатков SOA является большой размер сообщений, основанных на формате XML, в следствие чего понижена производительность работы системы, а на обмен SOAP-сообщениями тратится много ресурсов Интернет-трафика. В этом плане SOA уступает таким программным системам, как CORBA, RMI, DCOM.

Архитектура SOA основана на принципах многоразового использования IT-технологических функциональных элементов, недопущения повторения функциональности различных программных продуктов, унификации типовых программных процессов. Программные компоненты, будучи распределёнными по сети, могут быть независимыми между собой и слабо связанными, а в случае необходимости могут быть легко заменены другими компонентами, осуществляющими взаимодействие по аналогичному протоколу. Программа, основанная на сервис-ориентированной архитектуре, обладает интерфейсом элементов, осуществляющим инкапсуляцию процесса реализации конкретного элемента от остальных.

Одним из главных достоинств сервис-ориентированной архитектуры является относительная простота в создании сложных программных комплексов путём комбинирования мелких программных компонентов. Благодаря этому компания эффективнее определяет ресурсы для разработки своей программной системы и сокращает издержки путём оптимизации процесса разработки. Также компания может сама для себя определять рамки использования платформ, языков разработки и прочих инструментов, а масштабируемость архитектуры позволяет вести эффективную разработку системы в условиях нестабильной (растущей) кадровой базы.

Таким образом, важнейшими принципами сервис-ориентированной архитектуры являются: простота реализации, кроссплатформенность, расши-

ряемость, наличие возможности поиска других веб-сервисов и их интеграции в систему вне зависимости от платформы и языка разработки.

1.2. Средства разработки веб-сервисов

Несмотря на то, что существует множество специализированных программных средств, направленных на создание веб-сервисов, в основе любой программной разработки лежит некий язык программирования – система символов и команд, посредством которых записывается компьютерная программа. От используемого языка зависит архитектура программы, так как каждый язык имеет собственные уникальные лексические, синтаксические и семантические правила [22].

На сегодняшний день насчитывается порядка нескольких тысяч языков программирования, каждый из которых имеет свою собственную рабочую структуру и направлен на решение определённых задач программирования. Количество языков программирования растёт из года в год, каждый новый язык создаётся для разработки всё более сложных программных систем. История создания языков программирования позволяет выделить пять поколений языков: машинные (первое поколение), ассемблеры (второе поколение), языки высокого уровня (третье поколение), объектно-ориентированные (четвёртое поколение), языки искусственного интеллекта (пятое поколение).

Языки первого поколения специфицированы на конкретное аппаратное обеспечение, в зависимости от которого они используют в своём синтаксисе команды в двоичном или восьмеричном формате. Сложность таких языков определяется необходимостью в знании не только языка программирования, но и архитектуры рабочей машины.

Языки второго поколения называются ассемблерами. Одной из их особенностей является то, что они для обозначения команд используют их мнемонические имена вместо двоичных или восьмеричных форматов, характерных для языков первого поколения, однако они по-прежнему остаются зави-

симыми от архитектуры ЭВМ. Эти языки используются и по сей день для разработки высокоэффективного программного обеспечения.

Главным отличием языков третьего поколения от предыдущих является их независимость от архитектуры машины. Для этих языков характерно широкое использование интерпретации программ, в рамках которой написанный код перед исполнением преобразуется непосредственно в машинные команды, которые выполняются построчно уже во время исполнения программы вплоть до её завершения либо обнаружения ошибки в синтаксисе.

Языками четвёртого поколения называются среды программной разработки, внутри которых объединены основные инструменты разработки, к которым относятся текстовый редактор, компилятор и отладчик (наличие других средств обусловлено конкретной средой разработки). Главное преимущество языков четвёртого поколения заключается в предоставлении разработчику возможности быстро переключаться между основными средствами разработки и избегать осуществления различных дополнительных действий, благодаря чему программист посвящает больше рабочего времени непосредственно разработке, и эффективность этого процесса заметно возрастает. Также эти языки отличаются объектно-ориентированным подходом к разработке программного обеспечения, что означает, что они являются не универсальными, а ориентированными на некую определённую предметную область (веб-разработка, базы данных или др.).

Языки пятого поколения, как и четвёртого, используются в средах разработки, однако тенденции новых языков направлены на максимальное упрощение процесса программирования посредством введения в среды разработки инструкций по обеспечению пользователя возможностью взаимодействовать со средой наиболее удобным для него образом – например, посредством естественного языка или графической визуализации [8].

Несмотря на то, что существует порядка более восьми тысяч языков программирования, большинство из них используется в крайне определён-

ных предметных областях либо не используется вообще, поэтому самые популярные языки, будучи наиболее универсальными (относительно конкретной предметной области) и технически совершенными, являются абсолютным меньшинством среди существующих языков программирования. По версии индекса TIOBE, приведённые ниже языки являются наиболее популярными среди пользователей-программистов на момент 2019 года [33].

Java – типизированный объектно-ориентированный язык программирования, являющийся одним из наиболее популярных языков благодаря балансу простоты, надёжности и производительности. Главной особенностью этого языка является использование так называемой виртуальной машины Java (JVM – Java Virtual Machine), которая исполняет программу, предварительно откомпилированную из текстового формата в байт-код. Такая особенность позволяет исполнять в среде Java программы, написанные на других языках программирования, посредством компиляции чужеродного программного кода в байт-код, который впоследствии будет исполнен виртуальной машиной Java.

C – стандартизированный язык программирования, изначально предназначенный для использования в операционной системе UNIX, но впоследствии перенесённый на другие операционные системы и ставший самым популярным языком программирования для создания прикладного программного обеспечения. Из-за минимизированного количества ключевых слов и отсутствия классов этот язык является простым для использования, при этом неограниченным своей простотой. Под влиянием этого языка были разработаны такие языки, как Java, C++, PHP и др.

C++ – типизированный язык программирования общего назначения, направленный на различные парадигмы программирования: объектно-ориентированную, обобщённую, процедурную, функциональную. Главным образом язык применяется в разработке прикладных программных приложений, драйверов для внешних аппаратных устройств, а также программного

обеспечения для серверов. Так как C++ является прямым расширением языка C, синтаксически оба языка не отличаются друг от друга. Главным отличием C++ от C является направленность первого на объектно-ориентированную парадигму программирования, что подразумевает использование системы классов в процессе написания программного кода. К другим менее значимым нововведениям относятся дополнительные типы данных, виртуальные функции и др.

JavaScript – скриптовый язык программирования, один из диалектов ECMAScript. Способен работать в объектно-ориентированной, функциональной и императивной парадигмах. Язык разрабатывался как облегчённая альтернатива Java, более доступная для понимания новичков, вследствие чего язык имеет более узкое применение, чем Java. Чаще всего используется как встраиваемый язык программирования, применяемый в веб-дизайне для создания интерактивных элементов веб-страниц, а также для реализации некоторых функций браузеров. Хотя язык долгое время был популярен, сейчас некоторые из его функций являются устаревшими, из-за чего постепенно ему на смену приходит альтернатива в виде пятой версии языка HTML.

Python – интерпретируемый язык программирования высокого уровня, направленный на различные парадигмы программирования, в том числе объектно-ориентированную. Подобно языку C, разработка Python была направлена, в первую очередь, на создание простого, но эффективного синтаксиса, который обеспечивает выразительность программного кода, не нанося ущерб эффективности и производительности выходящей программы. Главным образом язык направлен на скоростное и свободное написание программного кода, свобода которого выражается в том, что решение конкретной задачи может быть достигнуто множеством различных способов, однако такая свобода открывает широкое пространство для ошибок, которые могут быть допущены неопытным программистом при написании кода. Помимо прочего Python имеет широкую стандартную библиотеку функций.

C# (произносится как си шарп) – строго типизированный объектно-ориентированный язык программирования. Данный язык создан компанией Microsoft в качестве основного языка разработки приложений на базе Microsoft .NET Framework. Будучи созданным под влиянием языков C++ и Java, C# имеет схожий с ними синтаксис, а также широкую библиотеку функций, в которую входят перегрузка операторов, итераторы, классы, атрибуты, события, свойства, анонимные функции и др.

PHP – скриптовый язык программирования общего назначения. Массово используется для создания веб-приложений, хотя не исключено использование языка в разработке GUI- (Graphical User Interface) или CLI-приложений (Command Line Interface). В среде веб-программирования является наиболее популярным языком благодаря своим простоте, производительности и функциональности. Функциональность языка может быть расширена благодаря многочисленным встраиваемым модулям, имеющимся в сети Интернет.

SQL – декларативный язык программирования, предназначенный для создания, редактирования и управления базами данных в реляционных базах данных. Является самым популярным языком для проведения различных операций над базами данных, однако бóльшая часть наиболее популярных реализаций этого языка отличается настолько, что перенесение программного кода из одной СУБД в другую без внесения в него существенных правок почти никогда не представляется возможным. Причиной этого является объёмный и сложно написанный стандарт, в котором не учтены некоторые важные детали в различных областях реализации языка. Помимо прочего, разработчики языка SQL отошли от его первоначальной концепции простого языка управления базами данных, сделав из него инструмент разработчика, а не конечного пользователя, вследствие чего он стал сложнее, чем задумывалось, но остался по-прежнему самым популярным языком управления базами данных.

Perl – интерпретируемый динамический язык программирования общего назначения. Главной особенностью языка являются обширные возможно-

сти для работы с текстовой информацией, а также регулярными выражениями. Наиболее часто используется в сетевой среде для автоматизации некоторых повседневных процессов, хотя язык имеет применение и во многих других областях: системном администрировании, веб-разработке, биоинформатике и др.

Хотя вышеприведённые языки программирования разработаны в период с середины XX века по конец девяностых годов, они до сих пор являются популярными и широко используются многими пользователями-программистами и IT-предприятиями. Одной из причин этого является то, что все эти языки программирования до сих пор поддерживаются своими разработчиками, регулярно получая обновления, приносящие в язык новые функции и исправления ошибок и недочётов. Также поддержка языков осуществляется и сторонними пользователями, разрабатывающими для них различные модули, расширяющие возможности сред программирования. Однако главная причина их популярности заключается в том, что своей функциональностью они в полной мере удовлетворяют потребности IT-общественности. По мере изменения потребностей компании-разработчики вносят необходимые правки в функционал своих разработок, и если они по какой-либо причине неспособны удовлетворить текущие потребности своих клиентов, то их разработки станут терять свою востребованность, и их место займут разработки другой компании.

Самым сложным вариантом разработки веб-сервиса является его полное самостоятельное создание, начинающееся с написания и отладки программного кода и заканчивающееся его интеграцией в сетевую среду. Ввиду того, что на реализацию этого способа требуется очень большое количество времени и крайне глубокие познания в программировании, веб-программировании и сетевом администрировании, существует способ, в рамках которого разработка веб-сервиса доверяется стороннему разработчику, в лице которого выступает либо опытный программист-фрилансер, либо компания, осуществляющая разработку программного обеспечения на заказ. Са-

мый лёгкий способ создания рабочего веб-сервиса заключается в использовании специальной программной системы, предназначенной для создания и управления контентом веб-сервиса, которая сокращённо называется CMS.

CMS (англ. Content management system, система управления контентом) – информационная система, предоставляющая инструменты для упрощённого создания и управления содержимым веб-сервиса. CMS является программным движком, предназначенным для создания веб-сервиса, ввиду чего его разработка сводится лишь к использованию уже готового набора инструментов, позволяющего управлять текстовым и графическим содержимым создаваемого веб-сервиса. Самым сложным в реализации этого способа создания веб-сервиса является освоение инструментария системы управления контентом, однако эта сложность не идёт ни в какое сравнение со сложностью самостоятельного создания веб-сервиса.

Для того чтобы сэкономить время на разработку и реализацию внешнего оформления веб-сервиса, имеется возможность воспользоваться готовым шаблоном из Интернета – специальной заготовкой дизайна веб-сервиса, позволяющей пропустить графический этап его разработки. Чаще всего CMS позволяют хранить несколько шаблонов внутри себя и, в случае необходимости, заменять ими текущий шаблон [4].

Важной частью любой CMS является возможность подключения внешних программных модулей, расширяющих возможности веб-сервиса. Например, к таким расширениям относится пользовательский чат на главной странице веб-сервиса или особая система транзакций, осуществляющая во время совершения торговой сделки изъятие денежных средств из личного банковского счёта пользователя.

Существует множество различных CMS, разработанных как отдельными пользователями-программистами, так и целыми компаниями-разработчиками ПО. Количество существующих на сегодняшний день CMS

составляет более десятка тысяч экземпляров, каждый из которых может быть классифицирован по трём основным принципам [24]:

- платные/бесплатные;
- по способу управления содержимым сайта;
- по типам управляемых данных.

Платные/бесплатные CMS – классификация, определяющая наличие и методику монетизации той или иной CMS. **Платное** использование программных разработок подразумевает под собой, в первую очередь, внесение денежной платы, метод которого определяется финансовой политикой компании-разработчика. Программный продукт может быть приобретён за единовременную плату либо на заранее установленный срок платной абонентской подписки, по окончании которого дальнейшее использование программного продукта становится возможным только после очередного внесения абонентской платы. Также существует политика условно-бесплатного использования программного продукта, при которой пользователю доступны только самые основные либо демонстрационные возможности программы, а возможность использовать дополнительные инструменты становится доступной только после внесения определённой денежной платы. К наиболее популярным платным CMS относятся: 1С-Битрикс, Umi, NetCat. При использовании **бесплатных** CMS пользователь не обязан вносить денежную плату за обладание программой, однако остаётся возможность добровольного внесения денежного пожертвования в дальнейшую разработку программного продукта. Зачастую разработчики бесплатных программных продуктов публикуют исходный код своих разработок с целью обеспечения более эффективной обратной связи со стороны пользователей-программистов, способных обнаружить в опубликованном коде ошибки или неэффективные методы, допущенные во время разработки. К наиболее популярным бесплатным CMS относятся: WordPress, Joomla, Drupal.

По способу управления содержимым сайта – классификация, в рамках которой определяется способ генерации страниц, используемый системой:

- *CMS, генерирующие страницы по запросу;*
- *CMS, генерирующие страницы при редактировании*;
- *CMS смешанного типа*.

Способ ***генерации страниц по запросу*** подразумевает создание новой страницы после каждого запроса со стороны пользователя. В простом представлении навигация по веб-сервису выглядит как переход между HTML-страницами по ссылкам, что сопровождается генерацией страниц заново при каждом переходе, из-за чего увеличивается нагрузка на сервер.

Способ ***генерации страниц при редактировании*** от предыдущего способа отличается отсутствием необходимости в каждовременной генерации страниц при получении новых запросов, так как используется система статических веб-страниц, позволяющая осуществлять навигацию по веб-сервису без создания новых страниц. Генерация новой веб-страницы происходит только в случае осуществления изменения материала, наполняющего веб-сервис, в рамках чего происходит создание копии редактируемого блока.

CMS ***смешанного типа*** объединяют в себе предыдущие два способа генерации страниц, однако имеют разные варианты реализации. Первый вариант основан на использовании специального хранилища информации, который называется кэш. В кэш отправляются копии HTML-страниц, на которые поступал запрос, после чего, в случае повторного запроса страницы, CMS загружает эту страницу прямиком из кэша, благодаря чему отпадает необходимость генерировать одну и ту же веб-страницу заново. Второй вариант заключается в сохранении информации в виде блоков, из которых собирается страница по запросу пользователя.

По типам управляемых данных – так как существуют CMS, управляющие содержимым не только веб-сервиса, но и других программных

структур, имеется классификация CMS по типам управляемых данных. Они могут быть разделены на:

- ***системы управления контентом*** – CMS, позволяющие осуществлять управление содержимым веб-сайта – редактировать имеющееся содержимое или вносить новое. CMS подобного типа рассчитаны на расширение возможностей подконтрольных им сайтов с помощью разнообразных плагинов и модулей, друг от друга они отличаются, в основном, по сложности и набору возможностей;
- ***системы управления корпоративным контентом*** – тип систем, направленных на управление содержимым веб-сайтов крупных компаний, а также осуществление и поддержание различных бизнес-процессов;
- ***системы управления документами*** – системы, имеющие некоторые технические сходства с CMS. Чаще всего они используются в качестве дополнения к системам управления контентом, чтобы внутри CMS организовать управление текстовыми файлами, которыми могут быть различные договоры и документации;
- ***системы управления цифровыми правами*** – системы, предназначенные для содержания, редактирования и управления информацией об авторской принадлежности той или иной единицы интернет-контента, которой может быть изображение, аудио- или видеозапись, программный продукт и др.

В рамках работы, посвящённой разработке веб-сервисов, выделены наиболее популярные CMS, имеющие те или иные возможности для решения этой задачи: WordPress, Joomla, Drupal.

WordPress является одной из самых популярных систем для разработки веб-сайтов. Одна из причин такой популярности заключается в том, что данная CMS является бесплатной, в открытом доступе имеется её исходный код, при этом сама система проста в использовании. Для этой CMS создано множество разнообразных плагинов и модулей, а также шаблонов дизайна, в которые возможно внести собственные изменения. Разрабатывалась, в пер-

вую очередь, как система для создания сайтов-блогов, где имеется возможность создавать авторские тематические посты, состоящие из текста, изображений, аудио- и видеофайлов, оставляя за посетителями возможность комментировать эти посты. Хотя существуют модули, с помощью которых возможно создавать сайты любого назначения, чрезмерное их использование негативно сказывается на скорости работы сайта. Ещё одной слабой стороной данной CMS является относительно высокая нагрузка на сервер, которая усиленно возрастает при высокой посещаемости сайта, из-за чего появляется необходимость в установке специальных расширений, направленных на оптимизацию работы CMS, а значимость проблемы выбора качественного хостинга существенно повышается [17].

Joomla – бесплатная CMS, входящая в тройку наиболее популярных бесплатных систем для создания сайтов. Как и Wordpress, имеет открытый исходный код, ввиду чего в Интернете представлен широкий выбор модулей и дизайнерских шаблонов, созданных сторонними разработчиками. Одной из главных особенностей данной CMS является широкая универсальность инструментов системы, достигаемая установкой разнообразных модулей, при этом не действующая в ущерб простоте использования CMS и скорости её работы. На Joomla могут быть реализованы сайты самого разного назначения и размера при помощи модулей, однако система также обладает и широкими базовыми возможностями, позволяющими эффективно работать с наполнением сайта, – различными редакторами, менеджерами контента, позволяющими не только редактировать его, но и отслеживать реакцию на него со стороны пользователей, и др. Хотя CMS имеет множество инструментов для создания сайта любого назначения, разработка на её основе сводится к использованию готовых решений, не позволяющих создать сайт с уникальными возможностями [12].

Drupal – ещё одна бесплатная CMS. Как и вышеописанные системы, обретает свою функциональность за счёт добавления сторонних модулей, од-

нако данная CMS отличается повышенной сложностью работы внутри неё, из-за чего неопытный пользователь, прежде чем создавать собственный сайт на базе Drupal, вынужден потратить большое количество времени на изучение системы. Из-за её высокого порога использования она является профессиональным инструментом разработки, средствами которого достигаются универсальность и гибкость процесса разработки сайта [13].

Вышеприведённые системы управления контентом между собой не имеют принципиальных отличий – каждая из них может быть направлена на решение одной и той же задачи, однако множество мелких отличий, заключённых в фундаменте каждой из CMS, в совокупности образует причины, по которым одна CMS лучше приспособлена для создания интернет-магазина, чем другая, которая, в свою очередь, лучше остальных CMS справится с задачей разработки веб-блога. Из этого следует, что выбор CMS осуществляется, в первую очередь, исходя из задач, поставленных перед разработчиком, а затем критерии выбора пополняются дополнительными пожеланиями заказчика и личными предпочтениями разработчика.

1.3. Формализованное описание технического задания

1. Общие сведения.

1.1. Название организации-заказчика.

Заказчик: «Уральский государственный педагогический университет».

1.2. Название продукта разработки.

Интернет-магазин «DIVenus».

1.3. Назначение продукта разработки.

Веб-сервис, предоставляющий пользователям информацию о представленных на сайте товарах и услуги по их приобретению.

1.4. Категория пользователей.

Заказчики на покупку товаров.

1.5. Плановый срок выполнения.

01.08.2019-01.01.2020

2. Характеристика области применения.

2.1. Структуры и процессы, в которых предполагается использование продукта разработки.

Использование продукта разработки предполагается в сети Интернет в случае необходимости приобретения товара.

2.2. Характеристика персонала.

Полноценная работа интернет-магазина предполагает наличие набора сотрудников, осуществляющих разнообразные бизнес-процессы:

- **контент-менеджер** – осуществляет наполнение интернет-магазина товарами, вносит описание их характеристик, следит за актуальностью представленных товаров;
- **бухгалтер** – отвечает за распределение финансовых средств по различным отраслям бизнеса;
- **менеджер по заказам** – занимается отслеживанием и подтверждением поступающих заказов, телефонной консультацией, упаковкой товаров;

- **кладовщик** – осуществляет хранение, сортировку и выгрузку товаров;
- **курьер** – занимается доставкой товара к адресу назначения;
- **копирайтер** – отвечает за текстовое наполнение веб-сервиса: создаёт описания для товаров, пишет новостные посты;
- **SEO-специалист** – занимается продвижением интернет-магазина, поднятием его рейтинга в поисковых системах;
- **программист** – осуществляет программную поддержку всего веб-сервиса: оптимизирует работу сайта, исправляет ошибки в его работе;
- **системный администратор** – отвечает за информационную безопасность сайта, сводит к минимуму последствия от вирусных атак.

В рамках выпускной квалификационной работы демонстрация возможностей веб-сервиса будет осуществлена силами одного человека.

3. Требования к продукту разработки.

3.1. Требования к продукту в целом.

Разрабатываемый веб-сервис должен удовлетворять требования к функциональным характеристикам, параметрам технических средств, информационной и программной совместимости.

3.2. Аппаратные требования.

3.2.1. Аппаратные требования для разработки.

Процесс разработки требует наличия персонального компьютера.

3.2.2. Аппаратные требования для использования.

Процесс использования требует наличия персонального компьютера, имеющего подключение к сети Интернет.

3.3. Программное обеспечение.

3.3.1. Программное обеспечение для разработки.

3.3.1.1. Системное программное обеспечение.

Операционная система Windows 10.

3.3.1.2. Прикладное программное обеспечение.

OpenServer, интернет-браузер, Joomla.

3.3.2. Программное обеспечение для использования.

3.3.2.1. Системное программное обеспечение.

Операционная система, имеющая возможность взаимодействовать с веб-сайтами по протоколу HTTP.

3.3.2.2. Прикладное программное обеспечение.

Браузер для выхода в Интернет.

3.4. Форматы входных и выходных данных.

В качестве входных данных выступают:

- Со стороны заказчика – заявки на приобретение товара;
- Со стороны исполнителя – отчёты о принятии заявок.

В качестве выходных данных выступают:

- Со стороны заказчика – приобретённый в результате торговой сделки товар;
- Со стороны исполнителя – полученные в результате торговой сделки денежные средства.

Пользователи являются главным источником данных для веб-сервиса. Основные услуги, предоставляемые веб-сервисом, становятся доступными пользователю после регистрации в системе. Данные о пользователях хранятся на сервере.

3.5. Порядок взаимодействия с другими системами.

Продукт разработки взаимодействует с сервером, на котором он расположен. В рамках выпускной квалификационной работы продукт представлен на локальном сервере.

3.6. Меры защиты информации.

Система пользовательских аккаунтов, защищённых паролем.

4. Требования к пользовательскому интерфейсу.

4.1. Общая характеристика пользовательского интерфейса.

Взаимодействие пользователя с веб-сервисом подразумевает последовательное выполнение пользователем трёх основных действий:

- Просмотр списка товаров, ознакомление с характеристиками товаров;
- Оформление заявки на приобретение товара;
- Оплата и получение товара.

В связи с этим минимальный пользовательский интерфейс должен содержать навигационную кнопочную панель, товарный список и набор кнопок для оформления и отправления заявок на покупку товаров.

4.2. Размещение информации на экране, дизайн экрана.

Схематичные изображения страниц веб-сервиса приведены в п. 2.1. (см. Рис. 2.1, Рис. 2.2., Рис. 2.3., Рис. 2.4., Рис. 2.5.).

4.3. Особенности ввода информации пользователем, представление выходных данных.

5. Требования к документированию.

5.1. Перечень сопроводительной документации.

Рекомендации по использованию среды администрирования Joomla.

6. Порядок сдачи-приёма продукта разработки.

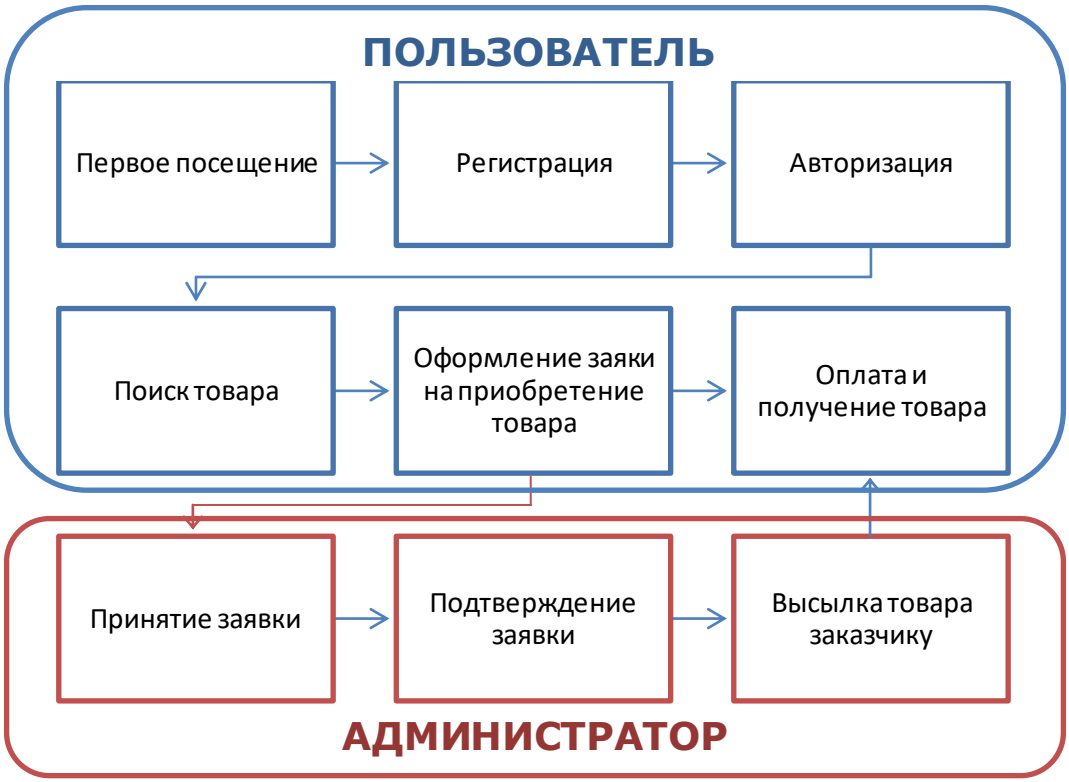
Продукт разработки считается принятым после получения положительной оценки от экспертов-преподавателей вуза.

Глава 2. Разработка интернет-магазина

2.1. Модельное представление продукта разработки

Целью данной выпускной квалификационной работы является создание интернет-магазина на основе системы управления контентом (CMS), предоставляющей набор инструментов для создания веб-сервиса. В рамках этой работы для создания интернет-магазина использована CMS Joomla.

Одной из опор при разработке интернет-магазина будет служить алгоритм, описывающий последовательность действий со стороны пользователя и администратора.



Пользователь, впервые зашедший на сайт веб-сервиса, первым делом оказывается на главной странице сайта, на которой, как правило, размещены такие элементы как навигационная панель, логотип, основная информация о веб-сервисе, окно с полями и кнопками для авторизации или регистрации пользовательского аккаунта. Пользователь, будучи неавторизованным на сайте, может просматривать на нём новости, информацию о размещённых товарах и контактные данные администрации, а также оформлять заявки на приобретение товаров, однако получать различные бонусы и подарки, вести

историю покупок и контактировать со специалистами пользователь может только при наличии собственного аккаунта и непосредственной авторизации внутри него. Пользовательский аккаунт необходим для того, чтобы веб-сервис хранил в памяти информацию о пользователях и их действиях на сайте.

Следующий шаг со стороны пользователя – это регистрация на сайте. В процессе регистрации пользователь указывает личные и контактные данные, необходимые для осуществления доставки товара. Наличие аккаунта даёт пользователю возможность не только осуществлять покупки, но и получать различные поощрительные скидки, привязанные к аккаунту, а веб-сервис, сохраняя в себе информацию о действиях пользователя, при наличии определённых инструментов может составлять модель пользовательских предпочтений, на основе которой система «рекомендует» пользователю товары той направленности, которая его, предположительно, интересует.

После регистрации и авторизации пользователь открывает список товаров, представленных на сайте. Пользователь может искать интересующий его товар в списке, предварительно дав команду показывать товары определённой категории, либо ввести в поисковую строку название необходимого товара. Как было сказано выше, при наличии определённых расширений для CMS, система, имея информацию о предпочтениях пользователя, может фильтровать список таким образом, чтобы в его начало были помещены товары, которые с наибольшей вероятностью заинтересуют пользователя.

Следующим шагом после выбора пользователем желаемого товара является формирование заявки на его приобретение. Пользователь указывает способ получения товара и адрес доставки, если выбран соответствующий способ получения. В конце этой операции заявка, содержащая в себе информацию о товаре и покупателе, помещается в список заявок, доступный администрации веб-сервиса. Со стороны администрации происходит обработка этой заявки с последующей упаковкой и отправкой товара.

Большинство существующих интернет-магазинов имеют схожую структуру веб-страниц, из которых они состоят, что позволяет выделить основные веб-страницы, являющиеся составными частями «эталонного» интернет-магазина, и схематически их изобразить: главная страница, профиль, заказы, страница товара, корзина.

Главная страница является первой, которую увидит пользователь, зайдя на сайт. На этой странице будет расположена краткая информация о сайте и список товаров, а также категории, по нажатию которых товарный список будет отфильтрован по выбранной категории (см. Рис. 2.1).

Профиль – на этой странице размещены поля, в которые вводится личная и контактная информация пользователя. На этой же странице пользователь может зарегистрироваться в веб-сервисе (см. Рис. 2.2.).

На странице **заказы** пользователь сможет отследить статус своей заявки. Зарегистрированному пользователю эта функция доступна сразу, а незарегистрированному – только после введения кода заявки. (см. Рис. 2.3.).

На **странице товара** будет приведена вся необходимая пользователю информация о товаре: модель, характеристики, цена и т.д. Интересующий пользователя товар может быть отправлен в корзину по нажатию соответствующей кнопки (см. Рис. 2.4.).

Корзина является заключительной страницей интернет-магазина. На ней будет приведён список товаров, которые пользователь намерен приобрести, а также перечень полей, в которые, если пользователь не зарегистрирован, необходимо ввести некоторые личные данные и адрес доставки, и кнопка «оформить заказ» (см. Рис. 2.5.).

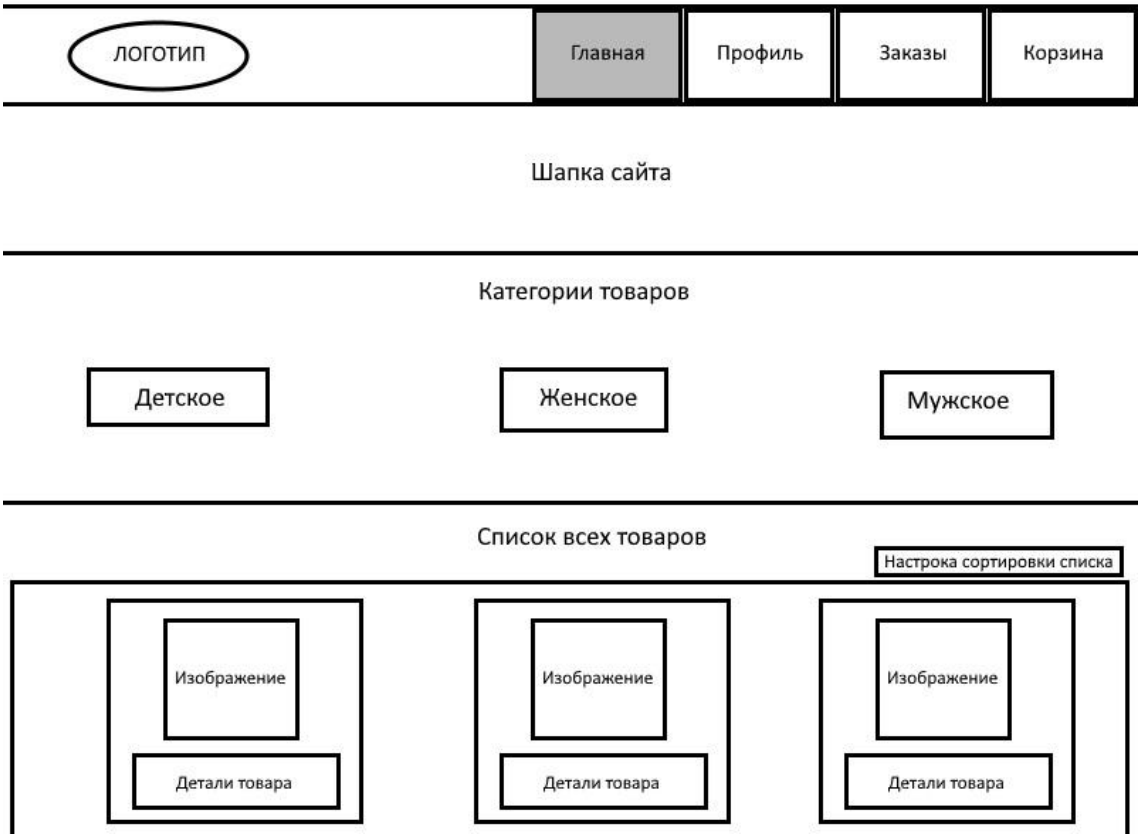


Рис. 2.1. Схематическое изображение главной страницы веб-сервиса.

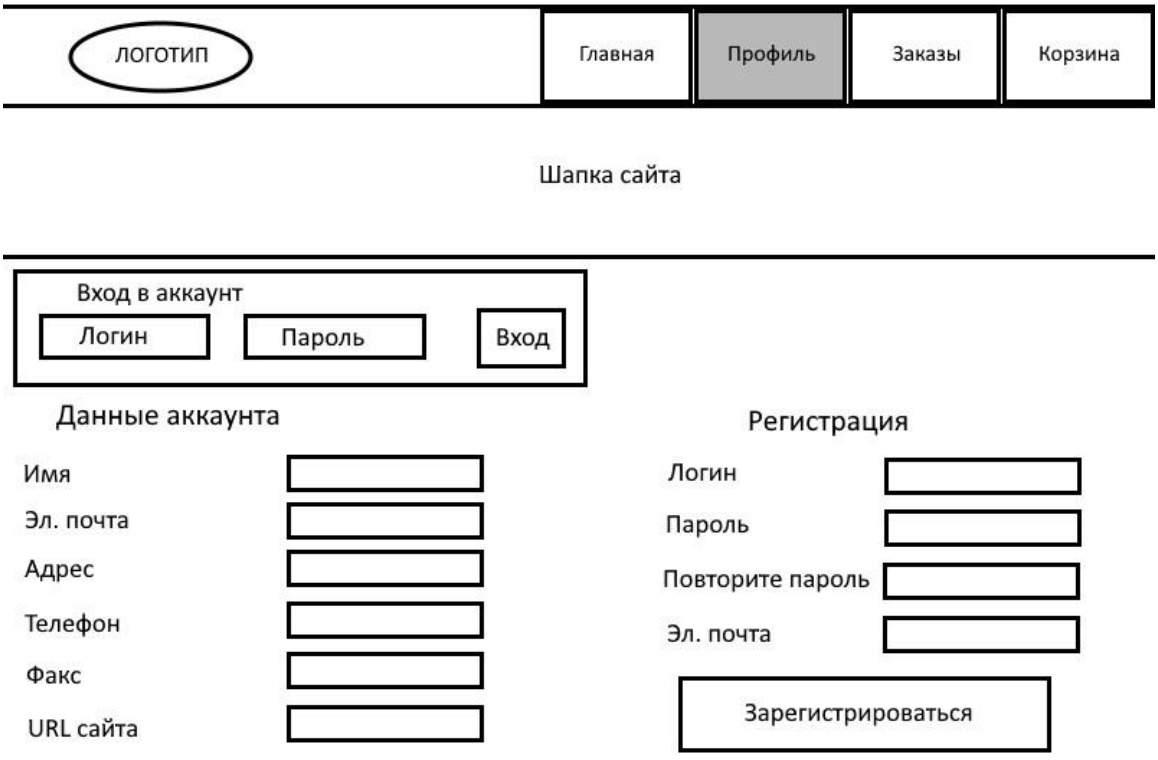


Рис. 2.2. Схематическое изображение страницы профиля.

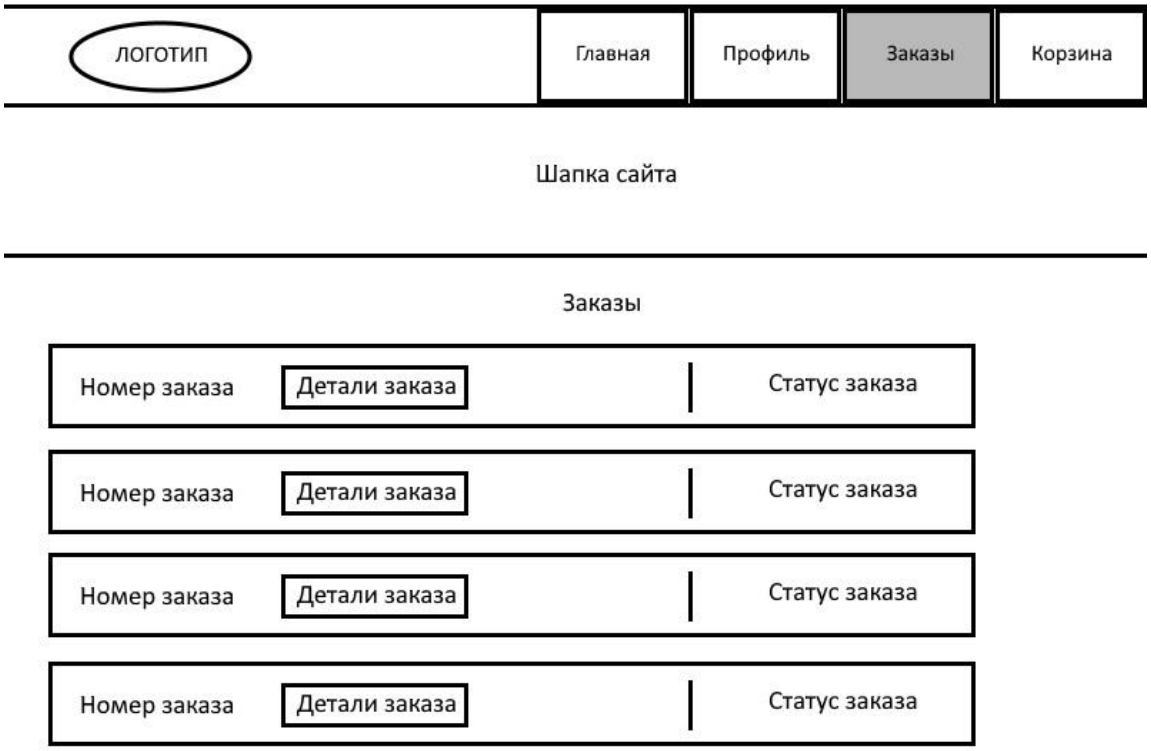


Рис. 2.3. Схематическое изображение страницы заказов.

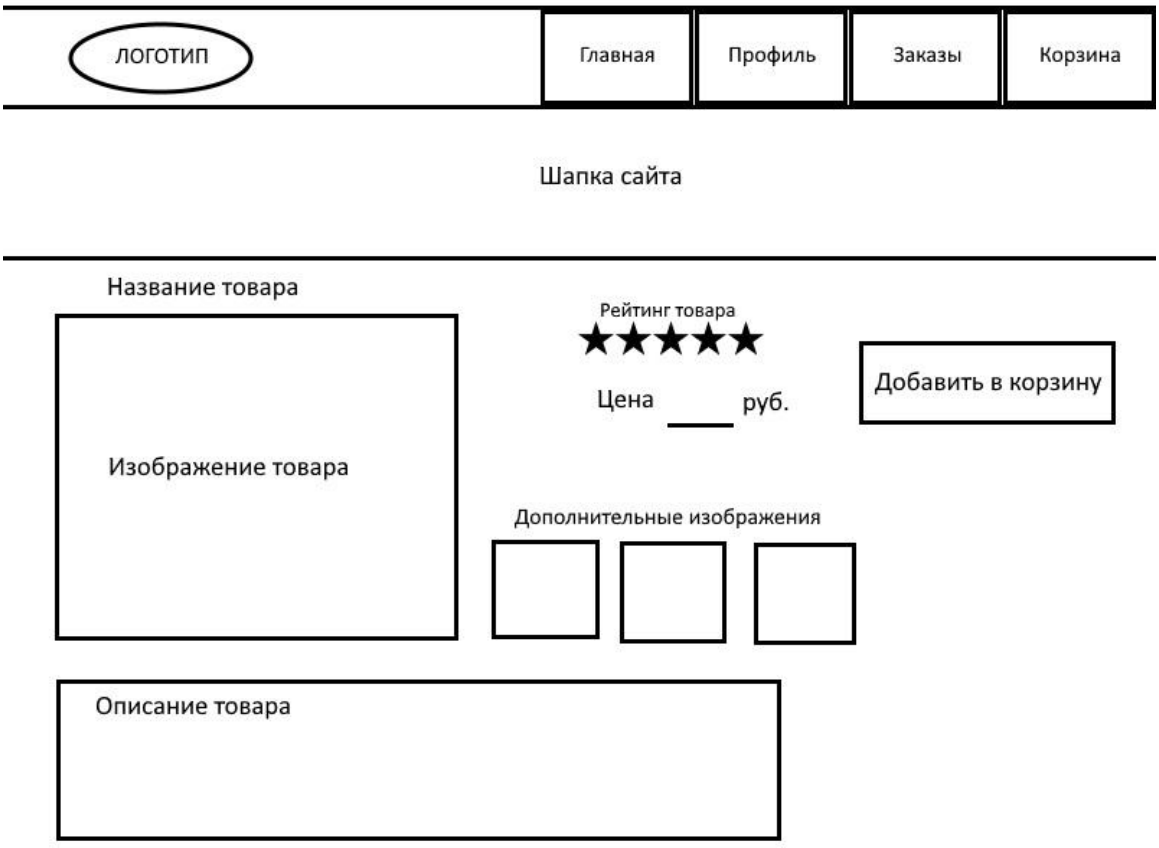


Рис. 2.4. Схематическое изображение страницы товара.

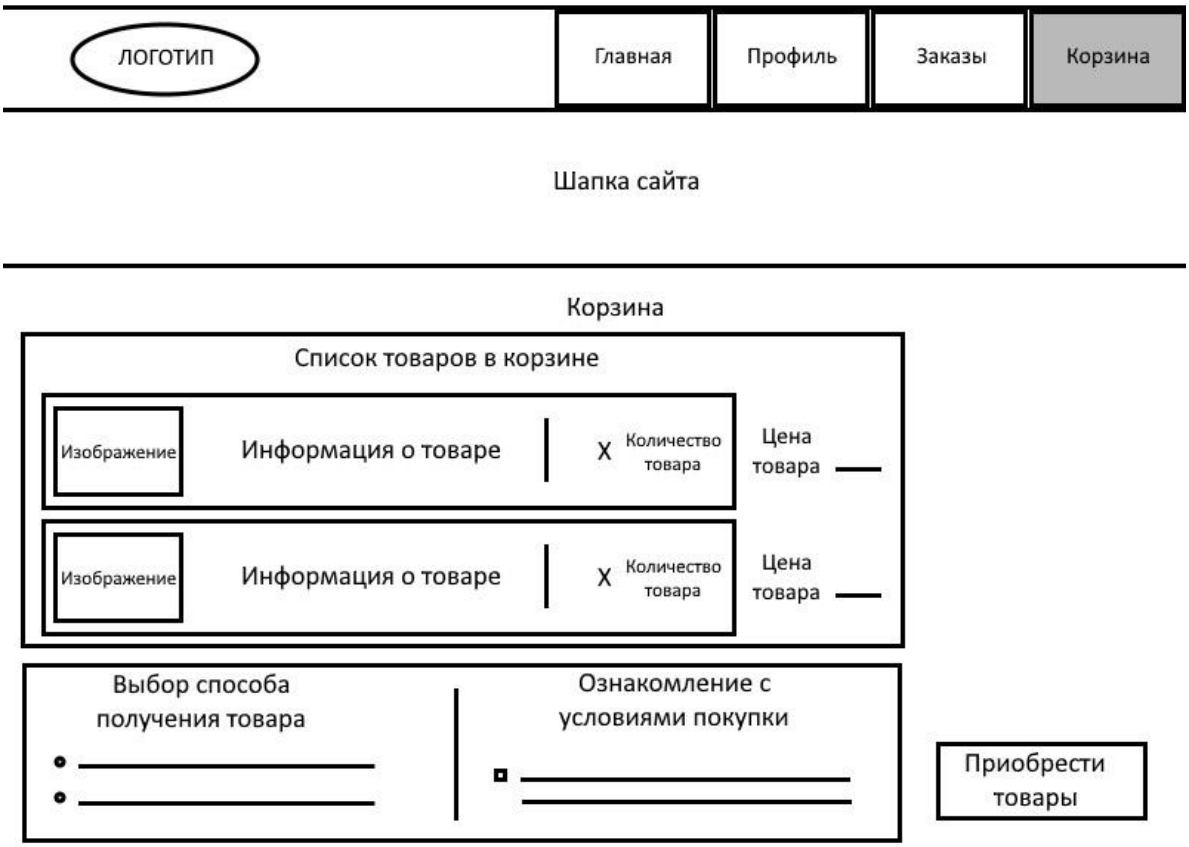


Рис. 2.5. Схематическое изображение страницы корзины.

2.2. Описание результата разработки

Использование CMS Joomla предполагает наличие готовой базы для разработки непосредственно интернет-магазина, в которую входят первоначальные макеты необходимым образом связанных между собой HTML-страниц и внутренняя панель управления веб-сервисом, вследствие чего процесс разработки не включал в себя работу над внутренней программной составляющей движка сайта. Это означает, что бóльшая и наиболее сложная часть работы над сайтом уже выполнена и не входит в рамки выпускной квалификационной работы, и разработка веб-сервиса была разделена на следующие этапы:

1) установка локального сервера на компьютер;
2) установка CMS Joomla на сервер;
3) установка расширения VirtueMart;
4) установка дизайнерского шаблона с поддержкой расширения VirtueMart;
5) настройка внешнего вида сайта с учётом особенностей установленного шаблона и предпочтений заказчика;
6) настройка навигационной панели и наполнения функциональных страниц.

По итогу выполнения приведённых этапов разработки был готов веб-сервис для осуществления онлайн-продажи одежды. Предполагаемому заказчику остаётся только завести служебные аккаунты для администрации и наполнить список товаров.

Веб-сервис, разработанный в рамках выпускной квалификационной работы, позволяет пользователям осуществлять заказ одежды.

Главная страница является первой, на которой оказывается пользователь, когда переходит на сайт (см. Рис. 3.1.). На её верхней половине расположена краткая информация о веб-сервисе (контактные данные, описание сервиса), а на нижней приведён список товаров и категорий. Хотя сайт поделён на несколько веб-страниц, на каждой из страниц имеется шапка сайта,

содержащая в себе контактные данные и ссылки на социальные сети, и навигационная панель, позволяющая пользователю в любой момент переместиться на нужную страницу.

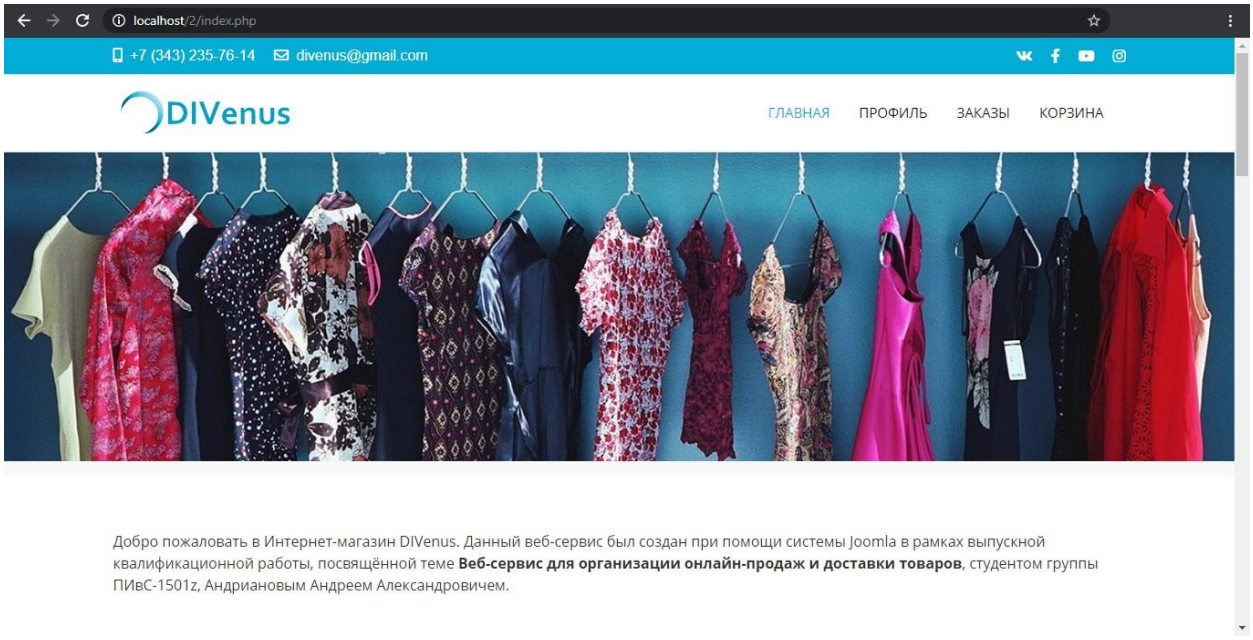


Рис. 3.1. Верхняя половина главной страницы.

Список товаров, расположенный на нижней половине главной страницы, является одним из ключевых элементов интернет-магазина, так как пользователь осуществляет выбор интересующего его товара именно из списка товаров (см. Рис. 3.2.). Сам список имеет небольшой ряд настроек, влияющих на отображение товаров внутри него. Могут быть настроены такие параметры как количество товаров на одной странице, сортировка по названию товара или производителя, а также фильтрация списка по категориям.

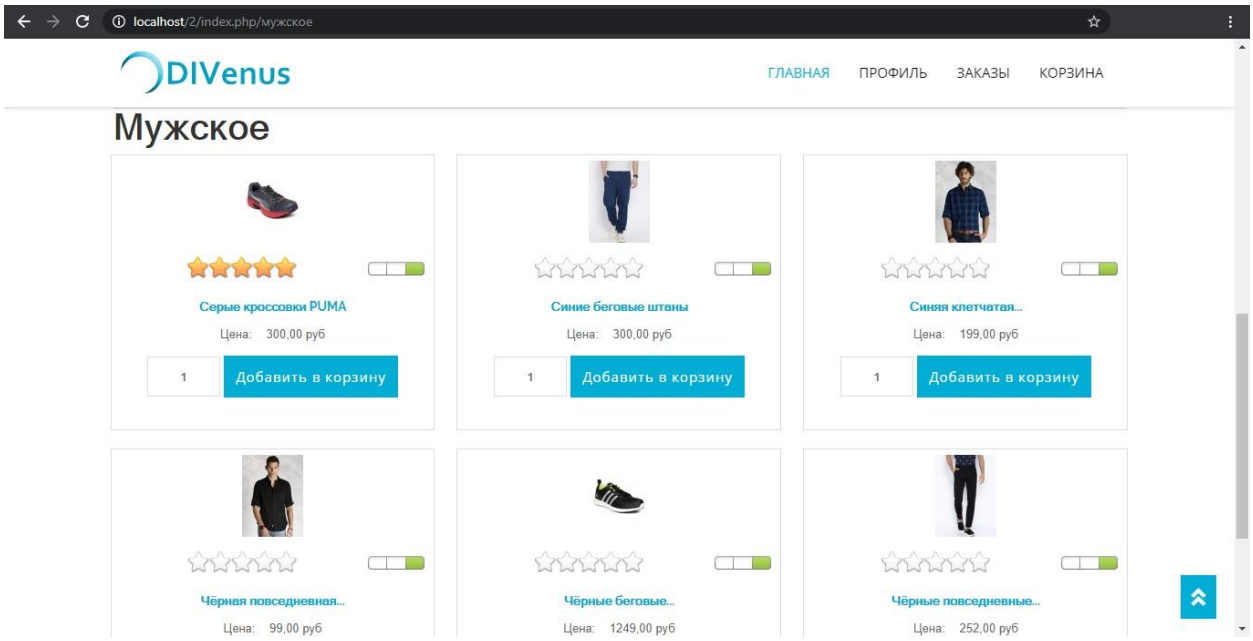


Рис. 3.2. Нижняя половина главной страницы.

Выбрав необходимый товар, пользователь переходит на его страницу (см. Рис. 3.3.,Рис 3.4.). Страница товара содержит в себе подробное описание товара, его цену и отзывы других покупателей о нём. Кнопка «добавить в корзину» отмечает товар, который пользователь собирается приобрести, тем самым помещая его в список будущих покупок, называемый «корзиной». В поле, размещённом рядом с этой кнопкой, пользователь может указать количество товара, которое нужно поместить в корзину.

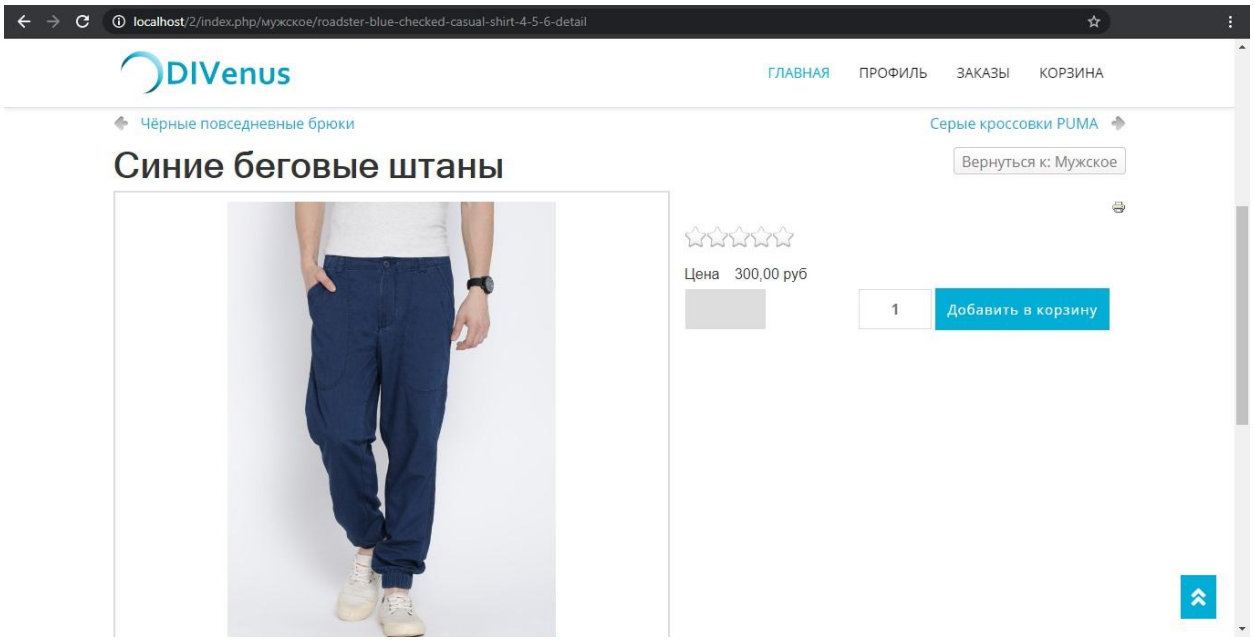


Рис. 3.3. Страница товара (верхняя половина).

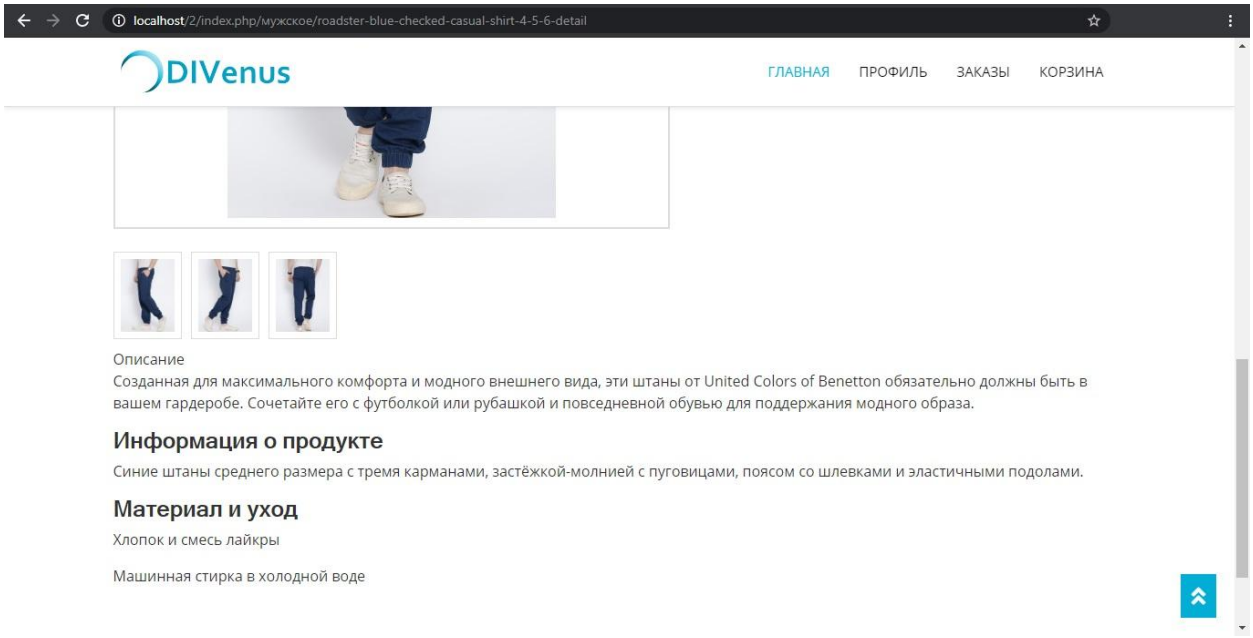


Рис 3.4. Страница товара (нижняя половина).

После того как пользователь завершил выбор необходимых товаров, он переходит на страницу «корзина», где приведены список выбранных товаров и их общая стоимость (см. Рис. 3.5., Рис. *3.6.*). На этой же странице пользователь указывает данные, необходимые для доставки товара, и некоторую личную информацию. Прежде чем завершить покупки, пользователь выбирает один из способов получения товара (доставка или самовывоз), после чего нажимает на кнопку «подтвердить заказ». После нажатия этой кнопки формиру-

ется заявка, которая незамедлительно поступает в список, доступный для администрации через панель управления (см. Рис. 3.7.) Приобретя товар, пользователь может оценить его по пятибалльной системе и оставить письменный отзыв о нём.

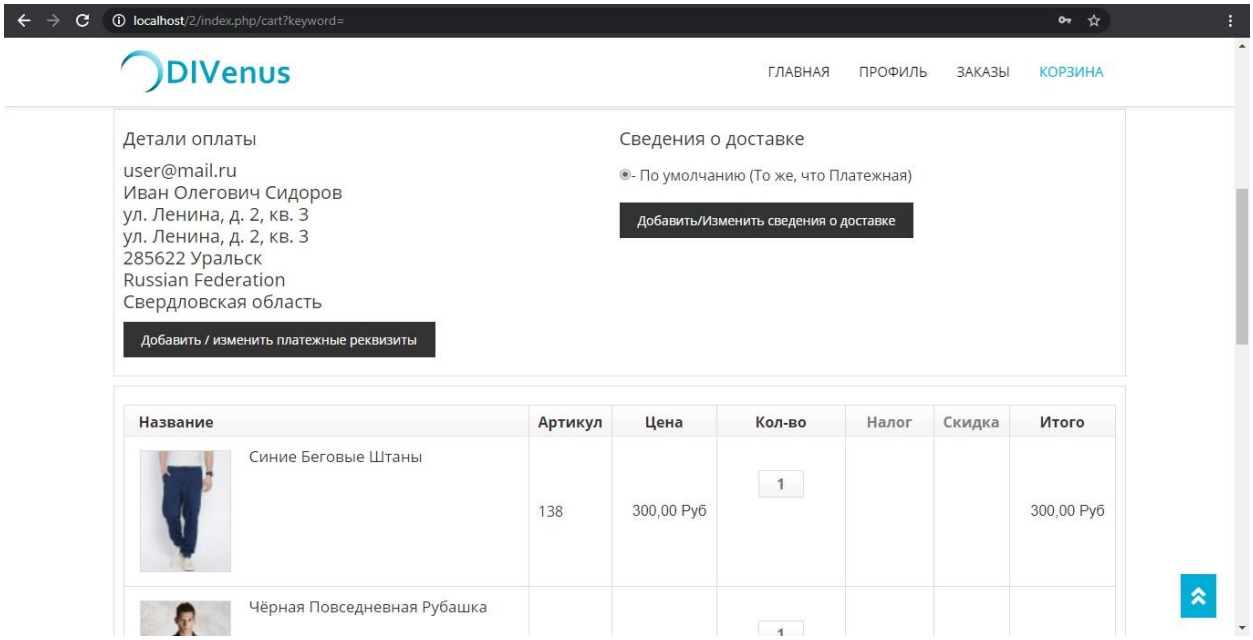


Рис. 3.5. Корзина (верхняя половина).

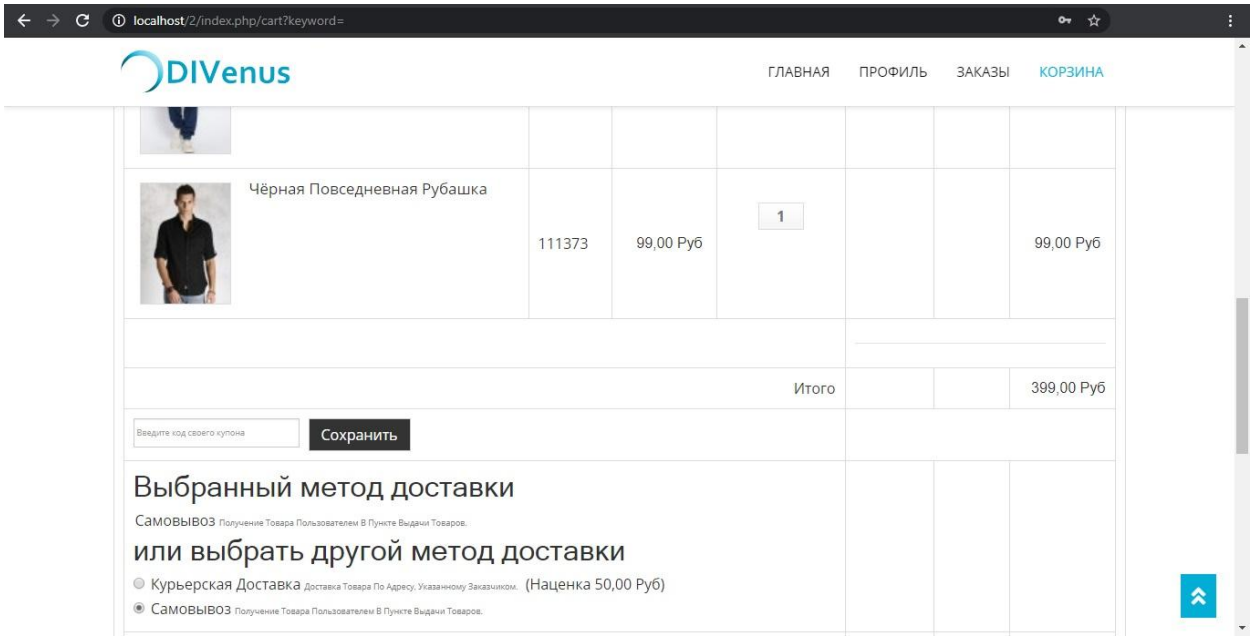


Рис. 3.6. Корзина (нижняя половина).

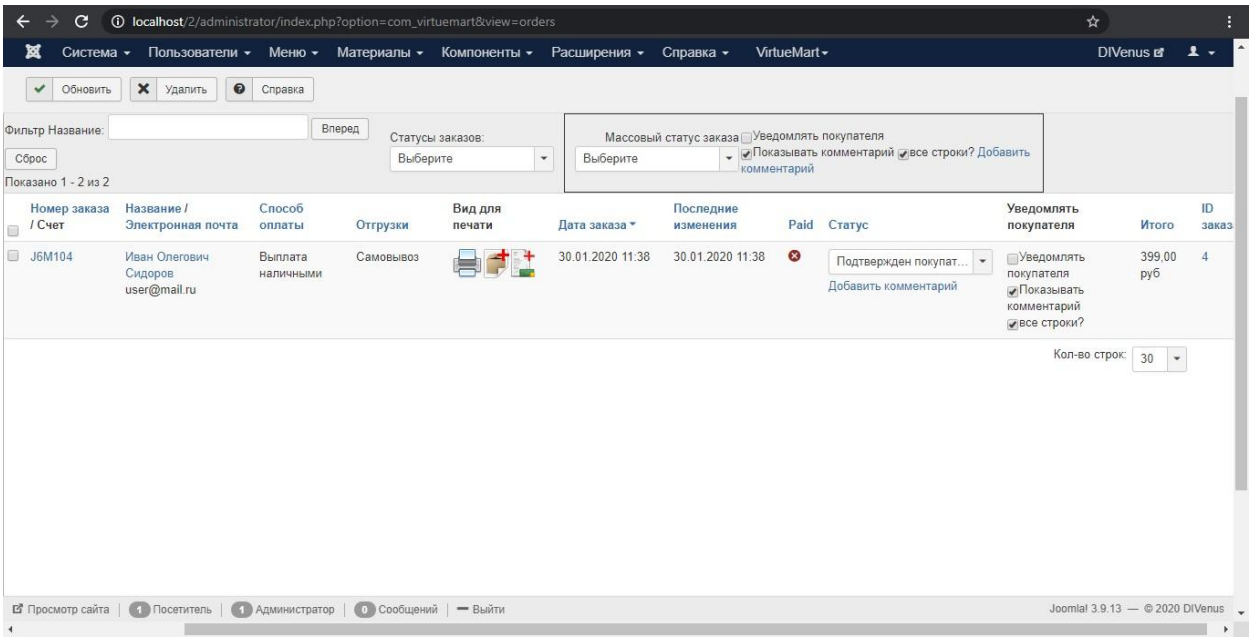


Рис. 3.7. Список заказов в панели управления.

Хотя пользователь может в полной мере использовать услуги интернет-магазина, не будучи зарегистрированным, наличие аккаунта предоставляет некоторые дополнительные удобства, например, отсутствие необходимости заново заполнять поля в корзине или возможность отслеживать статус заказа. В свою очередь, администрации это позволяет сохранять информацию о пользователе и его действиях на сайте. Регистрация пользователя, как и авторизация, осуществляется на странице «профиль», хотя необходимые для входа в аккаунт поля располагаются и на других страницах сайта (см. Рис. 3.8.).

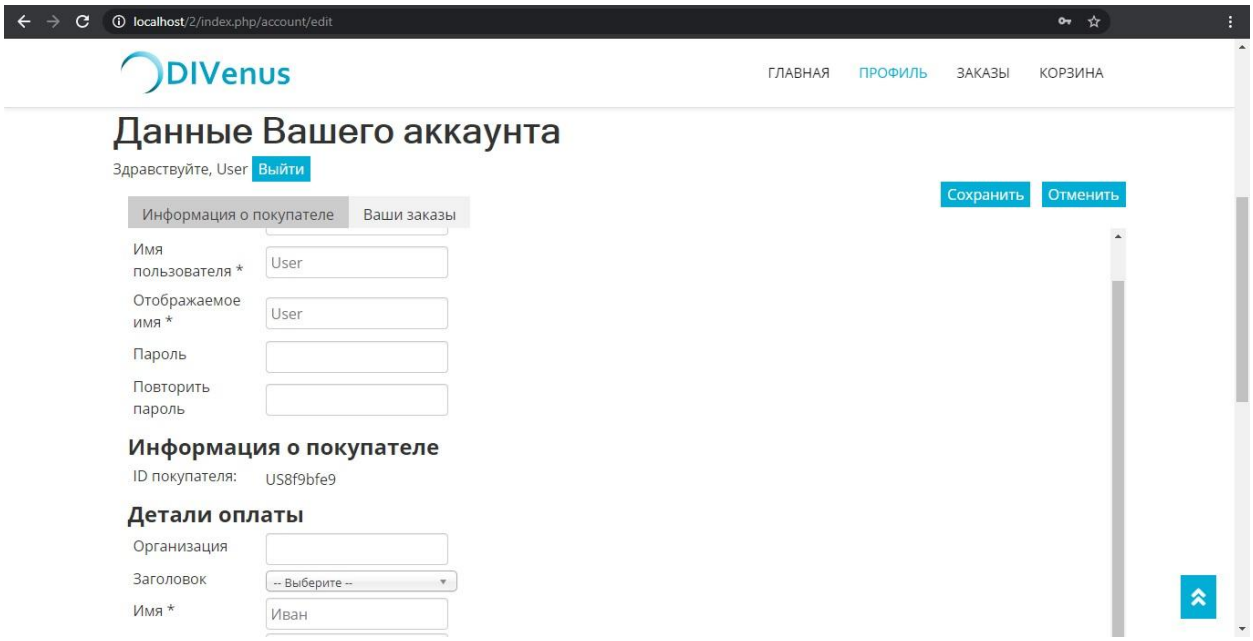


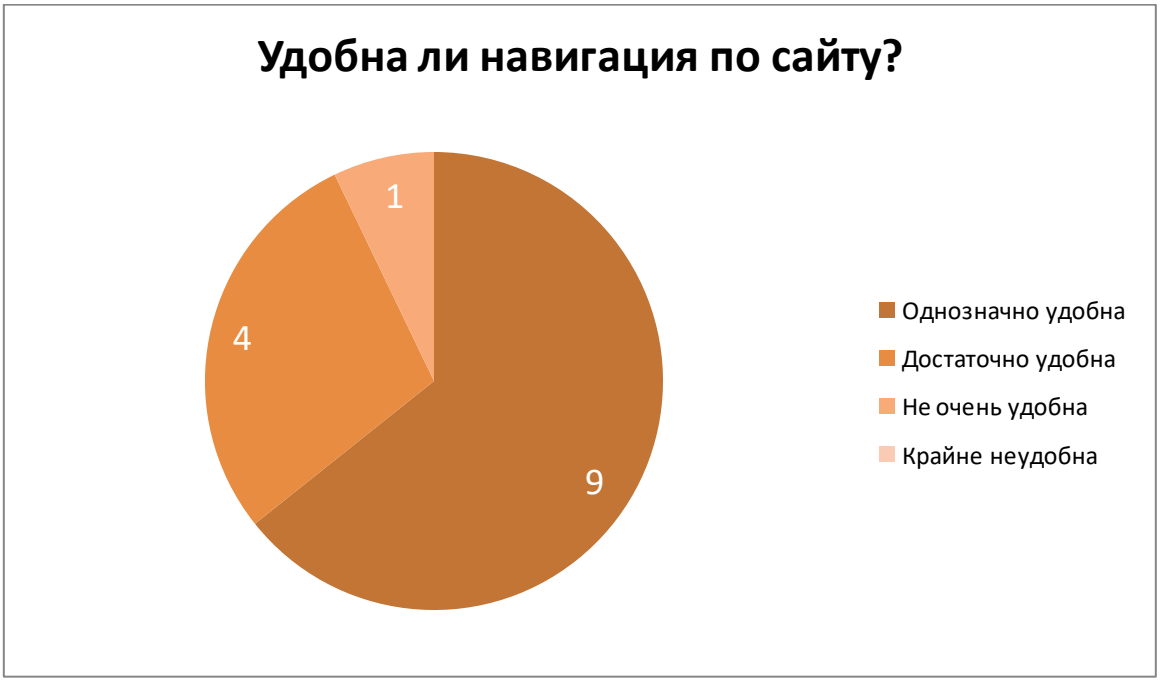
Рис. 3.8. Страница учётной записи (аккаунта) пользователя.

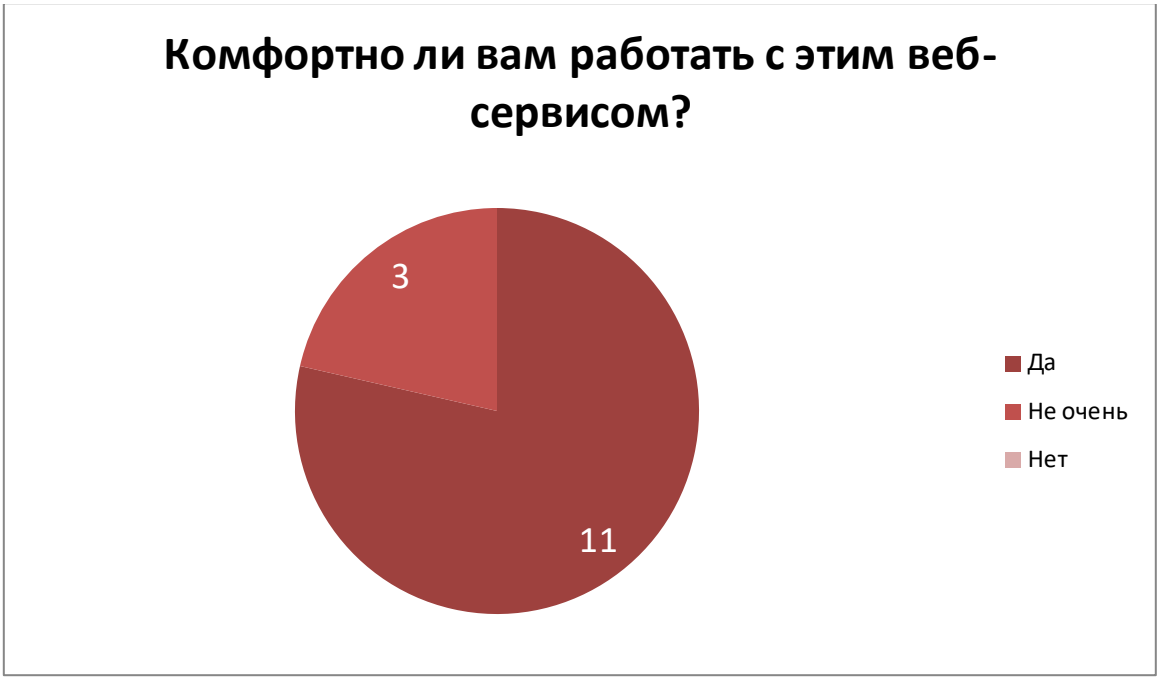
2.3. Результаты апробации продукта

После разработки веб-сервиса была проведена апробация продукта, в которой приняли участие четырнадцать человек, шестеро из которых являются уверенными пользователями персонального компьютера.

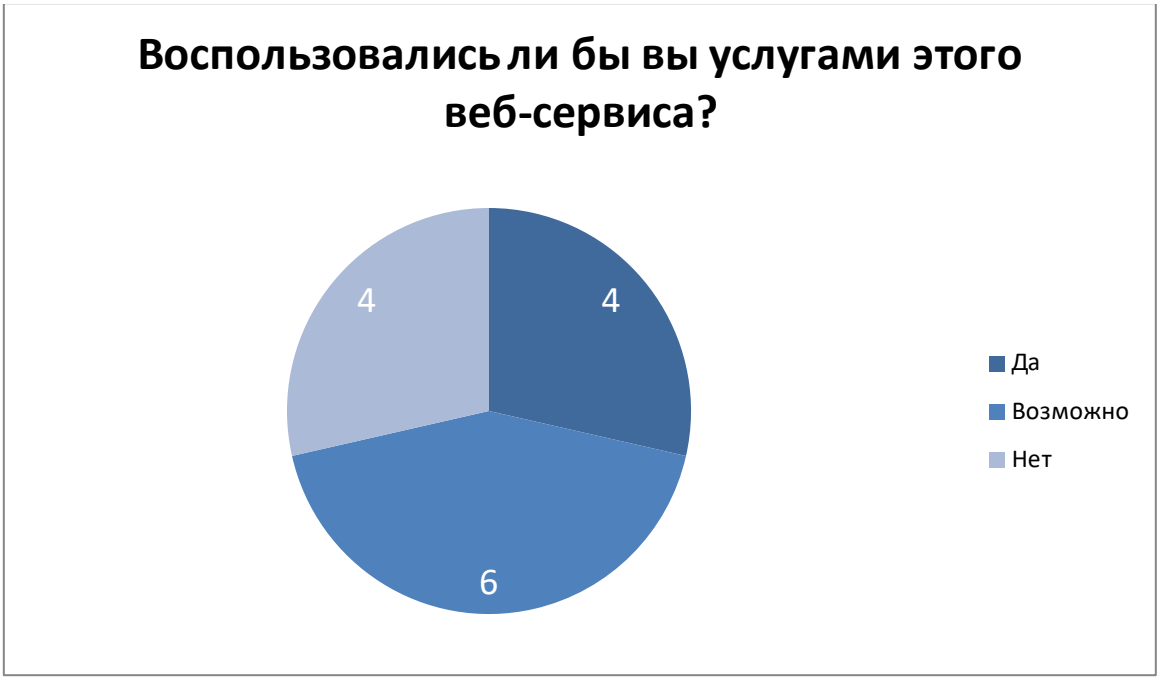
Весь процесс апробации заключается в том, что опрашиваемые люди лично испытывают работоспособность сайта, акцентируя внимание на его удобстве, внешнем виде, после чего отвечают на несколько вопросов, касающихся различных аспектов разработанного продукта. Результаты опроса приведены в виде диаграмм для того, чтобы обеспечить наглядность и удобство статистической оценки продукта.

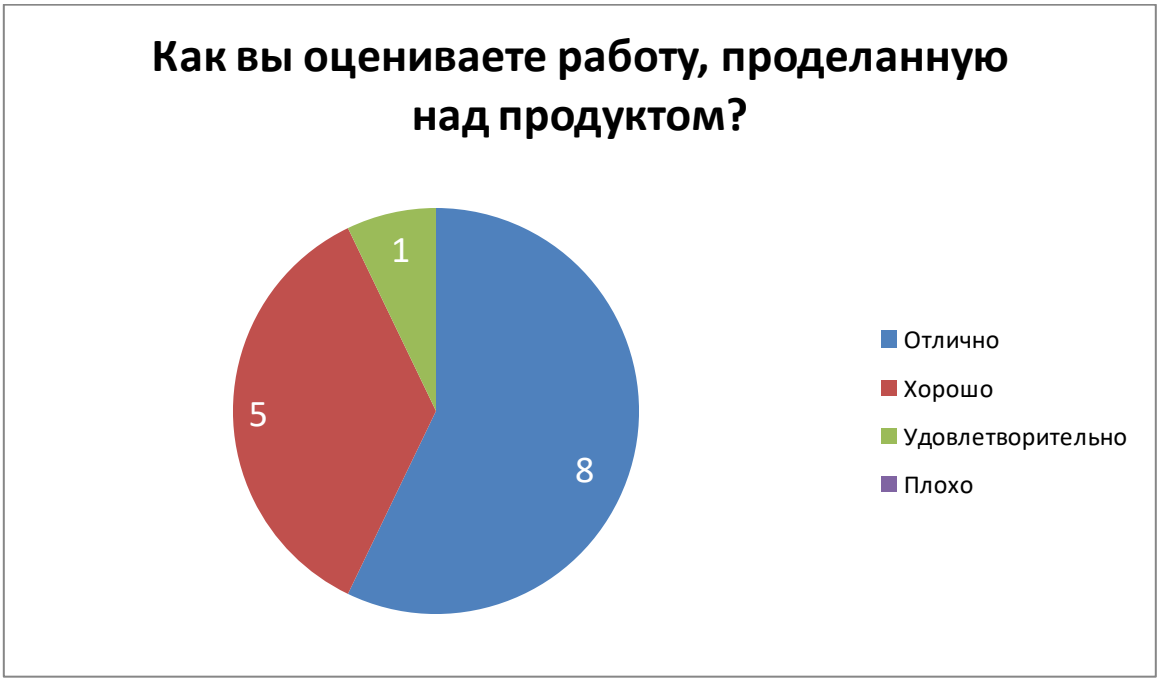

Как вы оцениваете внешний вид сайта?
1
7
6
Отлично
Хорошо
Нейтрально
Плохо


Удобна ли навигация по сайту?
1
4
9
Однозначно удобна
Достаточно удобна
Не очень удобна
Крайне неудобна


Комфортно ли вам работать с этим веб-сервисом?
3
11
Да
Не очень
Нет


Ощущаете ли вы нехватку функций, предоставляемых веб-сервисом?
4
10
Нет
Да, но незначительно
Да, значительно





Вышеприведённая статистика даёт достаточно прочную основу для осуществления разностороннего анализа разработанного программного продукта.

Внешний вид сайта был оценён, в целом, положительно. Пользователями было отмечено приятное общее оформление, однако многие заметили, что сайту не хватает дополнительных цветов и элементов, которые могли бы заполнить белое пространство страниц. Другие пользователи заметили некоторые мелкие недочёты, которые портят общее впечатление от работы с веб-сервисом.

Навигация по сайту так же была положительно оценена. Постоянная доступность верхней навигационной панели, которая видна даже при нахождении внизу страницы, оказалась достаточно удачным решением для высокой оценки большинством пользователей, однако некоторые столкнулись с трудностями при определении назначения некоторых страниц. Например, ввиду незнания интернет-терминологии, некоторые пользователи не сразу поняли, что на странице «профиль» можно пройти регистрацию, а другой пользователь начал искать список товаров на странице «заказы».

Пользователи хорошо оценили функциональность интернет-магазина, отметив, что в нём имеются все необходимые инструменты для полноценной работы. Значимой функцией оказалась возможность совершать покупки без регистрации аккаунта. Замечаний по поводу функциональности оказалось немного, однако все они затрагивали отсутствие различных сторонних модулей, которые не только бы внесли в веб-сервис дополнительные функции, но и разнообразили бы его внешний вид.

Общая оценка проделанной работы оказалась довольно высокой среди людей, не являющихся постоянными пользователями веб-сервисов. Люди, сильнее вовлечённые в IT-среду, предъявили большее количество замечаний к продукту, однако общая оценка с их стороны оказалась не намного ниже, вероятной причиной чего является поправка на то, что данный продукт был изготовлен в учебных целях. Хотя веб-сервис был оценён хорошо, немногие пользователи однозначно воспользовались бы его услугами, однако большинство людей проявили интерес к продукту и, вероятно, опробовали бы его в реальных условиях, если бы были исправлены все недочёты разработки и учтены все замечания и пожелания пользователей.

Заключение

Тенденция упрощения взаимодействия пользователя с компьютером не только в области потребления услуг прикладных программ, но и их разработки, на текущий момент привела к тому, что пользователь, не обладающий глубокими познаниями в сфере веб-разработки, способен самостоятельно разработать веб-сайт для своих нужд. Свидетельством данной тенденции является разработанный в рамках выпускной квалификационной работы интернет-магазин, который, при всех недостатках использования бесплатных программных инструментов, может являться полноценным веб-сервисом, посредством которого можно оказывать материальные услуги пользователям всемирной сети.

В ходе выполнения выпускной квалификационной работы, основной целью которой является разработка интернет-магазина по продаже одежды, было выполнено следующее:

1. Изучены базовые принципы разработки веб-сервисов.
2. Рассмотрены современные системы разработки веб-сервисов.
3. Подготовлено техническое задание для разработки онлайн-магазина по продаже одежды.
4. В соответствии с техническим заданием спроектирован и разработан онлайн-магазин для продажи одежды.

В связи с вышесказанным следует сделать вывод, что основная задача выпускной квалификационной работы, заключающаяся в разработке веб-сервиса по продаже одежды, была выполнена.

Список информационных источников

1. Администрирование Joomla // WebForMyself URL: https://webformyself.com/administrirovanie-joomla/.
2. Беленькая М. Н., Малиновский С. Т., Яковенко Н. В. Администрирование в информационных системах. 3-е издание. – М.: Горячая линия-Телеком, 2019. – 408 с.
3. ГОСТ Р 57100-2016/ISO/IEC/IEEE 42010:2011 Системная программная инженерия. Описание архитектуры // ТЕХЭКСПЕРТ URL: http://docs.cntd.ru/document/1200139542 (дата обращения: 08.12.2019).
4. Дизайн шаблона Joomla для front-end разработчика совершенно незнакомого с CMS // Хабр URL: https://habr.com/ru/post/246061/ (дата обращения: 22.09.2019).
5. Документация Joomla! // Joomla! URL: https://docs.joomla.org/Main_Page/ru.
6. Дэн Рамел. Joomla! для профессионалов. – М.: Вильямс, 2014. – 448 с.
7. Интеграция программного обеспечения. Описание процесса от бизнес консультанта // Хабр URL: https://habr.com/ru/company/trinion/blog/245615/ (дата обращения: 15.09.2019).
8. Классификация языков программирования // Студопедия URL: https://studopedia.ru/18_70778_klassifikatsiya-yazikov-programmirovaniya.html (дата обращения: 21.09.2019).
9. Колисниченко Д. Самоучитель Joomla!. – М.: БХВ-Петербург, 2015. – 224 с.
10. Лимончелли Т. А., Хоган К., Чейлап С. Практика системного и сетевого администрирования. Том 1. – М.: Вильямс, 2018. – 1104 с.
11. Максим Солдаткин. Веб-дизайн, ноутбук, океан. – М.: Издательские решения, 2018. – 180 с.
12. Обзор и отзывы о CMS Joomla // UGUIDE.RU URL: https://uguide.ru/obzor-i-otzyvy-cms-joomla (дата обращения: 24.09.2019).

13. Обзор CMS Drupal // SITE-BUILDERS URL: https://site-builders.ru/cms-drupal (дата обращения: 24.09.2019).
14. Официальный сайт OpenServer // Open Server URL; https://ospanel.io/.
15. Официальный сайт Joomla! // Joomla! URL: https://www.joomla.org/.
16. Официальный сайт VirtueMart // VirtueMart URL: http://virtuemart.net/.
17. Преимущества и недостатки Вордпресс // Pro-Wordpress.ru URL: https://pro-wordpress.ru/poleznoe/preimushhestva-i-nedostatki-vordpress.php (дата обращения: 22.09.2019).
18. Распространение UDDI на управление сервисами // IBM URL: https://www.ibm.com/developerworks/ru/library/ws-uddisecure/ (дата обращения: 19.09.2019).
19. Сервис-ориентированная архитектура // Википедия URL: https://ru.wikipedia.org/wiki/Сервис-ориентированная_архитектура (дата обращения: 17.09.2019).
20. Создание архитектуры программы или как проектировать табуретку // Хабр URL: https://habr.com/ru/post/276593/ (дата обращения: 17.09.2019).
21. Стандарты для Web-сервисов // Кунегин URL: https://kunegin.com/ref6/web/4.htm (дата обращения: 18.09.2019).
22. Язык программирования // Википедия URL: https://ru.wikipedia.org/wiki/Язык_программирования (дата обращения: 21.09.2019).
23. Язык WSDL. Структура WSDL-документа. Способы создания. Режимы операции WSDL // Студопедия URL: https://studopedia.ru/6_129665_yazik-WSDL-struktura-WSDL-dokumenta-sposobi-sozdaniya-rezhimi-operatsii-WSDL.html (дата обращения: 19.09.2019).
24. CMS: главные функции, модули и виды // ОПТИМИЗМ.РУ URL: https://www.optimism.ru/wiki/CMS:_главные_функции,_модули_и_виды (дата обращения: 22.09.2019).

25. JoomDev – источник шаблонов для сайта // JoomDev URL: https://www.joomdev.com/.

26. Len Bass, Paul Clements, Rick Kazman. Software Architecture in Practice, Third Edition. 3 изд. Pearson Education, 2012. 624 с.

27. OASIS SOA Reference Model TC // OASIS URL: https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=soa-rm (дата обращения: 17.09.2019).

28. Reference Model for Service Oriented Architecture 1.0. Committee Specification 1, 2 August 2006 // OASIS URL: https://www.oasis-open.org/committees/download.php/19679/soa-rm-cs.pdf (дата обращения: 17.09.2019).

29. Reference Model for Service Oriented Architecture 1.0. OASIS Standard, 12 October 2006// OASIS URL: http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf (дата обращения: 19.09.2019).

30. SOAP // Википедия URL: https://ru.wikipedia.org/wiki/SOAP (дата обращения: 18.09.2019).

31. Systems and software engineering – Architecture description ISO/IEC/IEEE 42010 // iso-architecture.org URL: http://www.iso-architecture.org/42010/defining-architecture.html (дата обращения: 08.12.2019).

32. The Service Oriented Architecture Handbook – Everything You Need To Know About Service Oriented Architecture, M. – Isabelle Moran, 2016. – 174 с.

33. Tiobe index for November 2019 // TIOBE URL: https://tiobe.com/tiobe-index/ (дата обращения: 27.11.2019).

34. Universal Description, Discovery and Integration (UDDI) // Студопедия URL: https://studopedia.ru/5_160439_Universal-Description-Discovery-and-Integration-UDDI.html (дата обращения: 19.09.2019);

35. What is Your Definition of Software Architecture // Carnegie Mellon University URL: https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=513807 (дата обращения: 08. 12. 2019).