

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

Факультет
вычислительной техники

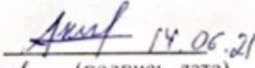
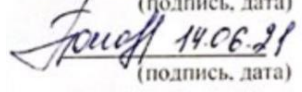
Кафедра
МОиПЭВМ

Направление подготовки 09.03.04 «Программная инженерия»

БАКАЛАВРСКАЯ РАБОТА

на тему

Программное обеспечение сайта интернет-магазина вещей «K-SHOP»

Студент	 (подпись, дата)	Акимов Кирилл Викторович (ФИО полностью)
Руководитель	 (подпись, дата)	Попова Н. А. (фамилия, инициалы)
Нормоконтролёр	 (подпись, дата)	Такташкин Д.В. (фамилия, инициалы)

Работа допущена к защите (протокол заседания кафедры от _____ № _____)

Заведующий кафедрой _____
(подпись) Макарычев П.П.
(фамилия, инициалы)

Работа защищена с отметкой _____ (протокол заседания ГЭК от _____ № _____)

Секретарь ГЭК _____
(подпись) Попова Н.А.
(фамилия, инициалы)

Федеральное государственное бюджетное образовательное учреждение
высшего образования

ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

Факультет

Вычислительной техники

Кафедра

МО и ПЭВМ

«Утверждаю»

Заведующий кафедрой

П.П. Макарычев

«21» января 2021 г

ЗАДАНИЕ НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ БАКАЛАВРА

1. Студент Акимов Кирилл Викторович гр. 17ВП1 факультета ВТ
<Фамилия, имя, отчество полностью>

2. Тема работы Программное обеспечение сайта интернет-магазина вещей «K-SHOP»

Тема утверждена приказом ПГУ № 0027а/о от " 21 " января 2021

2. Руководитель работы к.т.н., доцент кафедры «МО и ПЭВМ» Попова Н.А

3. Задание на работу (назначение разработки, исходные данные и т.п.)

1) Назначение: программное обеспечение сайта интернет-магазина

вещей «K-SHOP»

2) Среда выполнения: интернет-браузер актуальной версии

Средства разработки: языки программирования PHP, SQL

Инструментальные средства: Sublime Text, MySQL Workbench 8.0 CE, Open Server

Технологии разработки: ООП

3) Основные функции программного обеспечения сайта интернет-магазина

вещей «K-SHOP»:

а) поиск товаров;

б) подбор товара по категориям;

г) подбор товара по критериям;

д) оформления заявки на доставку товара;

е) управление магазином от лица администратора

4. Перечень подлежащих разработке вопросов

Анализ предметной области и требований

Анализ требований интернет-магазина

Проектирование интернет-магазина вещей K-SHOP

Реализация программного обеспечения для интернет-магазина

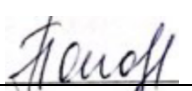
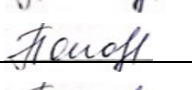
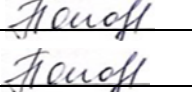
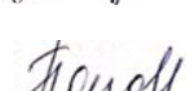
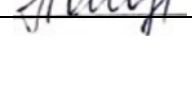
Контроль качества программного обеспечения интернет-магазина:

– Тестирование

– Анализ качества кода

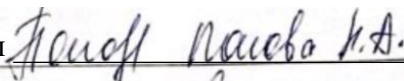
Структурное и календарное планирование для проекта программного обеспечения для интернет-магазина

5. Календарный график выполнения работы

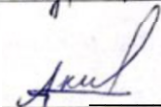
№ п/п	Наименование этапов работы	Объем работы	Срок выполнения	Подпись руководителя
1	Анализ предметной области и постановка задачи	15%	14.04.2021 – 21.04.2021	
2	Проектирование	25%	22.04.2021 – 05.05.2021	
3	Реализация	40%	06.05.2021 – 30.05.2021	
4	Контроль качества	5%	31.05.2021 – 02.06.2021	
5	Оформление пояснительной записки	15%	03.06.2021 – 14.06.2021	

Дата выдачи задания "28" _____ декабря _____ 2020

Руководитель бакалаврской работы
(подпись, дата)

 Пашова И.А.

Задание к исполнению принял студент
(подпись, дата)



Работу к защите допустить
Декан факультета
(подпись, дата)

Реферат

Пояснительная записка содержит 97 листов, 56 рисунков, 27 использованных источников.

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ, ИНТЕРНЕТ-МАГАЗИН, UML,
БАЗА ДАННЫХ, КЛИЕНТ, СЕРВЕР, MYSQL, OPEN SERVER

Объект разработки – программное обеспечение сайта интернет-магазина вещей «K-SHOP».

Цель работы – разработать программное обеспечение сайта, позволяющего подбирать товар по критериям, оформлять заявку на доставку, а также управлять магазином пользователю с правами администратора.

Технология разработки – для разработки серверной части использовалась СУБД MySQL Workbench и язык SQL, клиентская часть была реализована на языке PHP в редакторе Sublime Text с использованием фреймворка Bootstrap.

Результаты работы – в ходе выполнения выпускной квалификационной работы были созданы клиентская и серверная часть для программного обеспечения сайта интернет-магазина вещей «K-SHOP». Все требования к разработке приложения соблюдены.

					ПГУ 09.03.04–08ВР171.03 ПЗ			
Изм.	Лист	№ докум.	Подпись	Дата				
Разраб.		Акимов К.В.		14.06.21	Программное обеспечение сайта интернет-магазина вещей «K-SHOP»	Лит.	Лист	Листов
Провер.		Попова Н. А.		14.06.21			3	97
Н. Контр.		Такташкин Д.В.		21.06.21	Пояснительная записка	Группа 17ВП1		
Утверд.		Макарычев П.П.						

Содержание

Введение.....	7
1 Анализ предметной области	9
1.1 Анализ предметной области	9
1.2 Анализ популярных интернет-магазинов.....	10
1.2.1 Интернет-магазин AliExpress.....	10
1.2.2 Интернет-магазин Wildberries.....	11
1.2.3 Интернет-магазин Ивтекс 37.....	13
1.2.4 Сравнительный анализ рассмотренных программных аналогов	14
2 Анализ требований интернет-магазина	16
2.1 Анализ нефункциональных требований	16
2.2 Функциональные требования и разработка диаграмм	16
2.3 Описание функциональной модели	21
3 Проектирование интернет-магазина вещей K-SHOP	26
3.1 Описание инструментальных программных средств разработки.....	26
3.2 Базовая архитектура Web-сайта интернет-магазина	27
3.3 Логическая модель базы данных	28
3.4 Физическая модель базы данных.....	29
3.5 Построение диаграмм в методологии UML	30
3.5.1 Построение диаграммы классов программного обеспечения для интернет-магазина	30
3.5.2 Построение диаграммы деятельности для интернет-магазина вещей K-SHOP	31
3.5.3. Диаграмма кооперации интернет-магазина вещей.....	32
3.6 Проектирование пользовательского интерфейса.....	33
4 Реализация программного обеспечения для интернет-магазина.....	43

4.1 Построение диаграммы компонентов	
для интернет-магазина.....	43
4.2 Реализация диаграммы развертывания	
интернет-магазина.....	43
5 Контроль качества программного обеспечения	
интернет-магазина.....	45
5.1 Тестирование белого ящика	45
5.1.1 Метод покрытия операторов.....	46
5.1.2 Метод покрытия условий	47
5.1.3 Метод покрытия маршрутов	48
5.2 Тестирование черного ящика.....	49
5.2.1 Классы эквивалентности:	49
5.2.2 Функциональная диаграмма.....	52
5.3 Метрики кода.....	55
6 Структурное и календарное планирование для	
проекта программного обеспечения для интернет-магазина	57
6.1 Определение этапов и задач проектирования	
программного обеспечения для интернет-магазина.....	57
6.2 Календарное планирование и оценка бюджета	
проекта программного обеспечения для интернет-магазина	59
Заключение	66
Список использованных источников	67
Приложение А. Глоссарий	70
Приложение Б. Листинг программного обеспечения	72
Приложение В. Результаты тестирования	96

Введение

В момент, когда в нашей стране на смену традиционным магазинам пришла интернет-торговля покупательская активность получила резкий всплеск. Связано это с удобством и практичностью, поскольку сеть интернет присутствует, на сегодняшний день, в любой точке планеты.

Интернет-магазины стали главным шагом на пути к модернизации процесса покупок. С каждым годом цепочка от момента заказа до доставки товара автоматизируется глобальным образом. Невысокие цены, комфорт стимулируют покупателя для совершения удобной покупки. Также существуют факторы, заставляющие потенциального покупателя отказаться от услуг традиционного магазина и сделать выбор в пользу онлайн-ового.

Владельцам интернет-магазинов не приходится оплачивать аренду торговых точек, нанимать большой персонал, что позволяет снизить расходы и установить демократичные цены [1].

Покупатель может совершить покупку прямо из дома и для этого никуда не надо ехать. Заказ может быть осуществлен в любое время, что выгодно не только для самого клиента, но и для продавца.

Клиенты, как правило, ищут товар из конкретной категории, а с настроенной поисковой системой онлайн-магазинов сделать это значительно проще.

Фактор является одним из важнейших, поскольку отсутствует географические ограничения. Роль играют лишь нюансы доставки, но с правильно выстроенной логистикой выгода от покупки превысит расходы на транспортировку [3].

Позволяет увеличить количество обрабатываемых заказов с задействованием минимального количества сотрудников, а также способствует эффективному управлению предприятием за счет объединения всей информации в единую систему, чего не скажешь об оффлайн-ой сфере.

Для подтверждения вышесказанных факторов обратимся к статистическим данным за 2019 – 2020 год. По статистическим данным более 100 млрд. выручки российские покупатели принесли гигантам интернет-торговли: Aliexpress, Wildberries, Ozon и т.д. [5].

За 2019 год выручка интернет-торговли составила около 1,6 трлн рублей (425 млн. заказов). Количество онлайн-покупателей увеличивается с каждым годом, как и количество новых торговых точек. В апреле 2020 года открылось в 1,5 раза больше новых магазинов, чем в апреле 2019 года. Сначала режим самоизоляции заказов стало на 25% больше.

Большой рост по сравнению с предыдущими месяцами зафиксирован в нескольких нишах: продукты питания, товары для детей, дом и сад, а также одежда и обувь.

По прогнозам Data Insight рынок электронной коммерции в России будет стремительно наращивать обороты в ближайшие 4 года. Основные факторы роста – растущее количество онлайн-покупателей и повышенный спрос на товары одежды и обуви в интернете [5].

Вывод: интернет-коммерция – это не отдельная и самостоятельная область торговли, это одна из ее форм. Покупатель делает выбор в пользу удобства и практичности, которые могут предоставить онлайн-продажи. На фоне образующихся тенденций в виде популяризации удалённой работы, увеличения объёма инвестиций, притока онлайн-покупателей онлайн-рынок одежды и обуви будет расти.

Объектом исследования выступает процесс разработки веб-сайта по продаже товаров. Предметом же является интернет-магазина вещей. Цель выпускной квалификационной работы состоит в разработке программного обеспечения сайта интернет-магазина вещей K-Shop.

1 Анализ предметной области

1.1 Анализ предметной области

Интернет-магазин – это специализированная интернет-площадка, обеспечивающая возможность онлайн-приобретения различных товаров. Основная задача интернет-магазина состоит в том, чтобы предоставить удобный качественный сервис для посетителя, на котором можно изучить можно полностью изучить товар и сделать заказ [9].

Отличие интернет-магазина от обычной торговой точки заключается не только в возможности покупки товара, не выходя из дома, но и в более гибкой ценовой политике. Из-за этого интернет-магазины более предпочтительны [2]. Каждый интернет-магазин должен обладать определенными критериями, что позволит не просто привлечь клиента, а удержать его:

- комфортное расположение товаров и категорий товаров на сайте (удобство навигации, быстрый поиск);
- описание товаров с информативным изложением и необходимыми характеристиками;
- удобство оформления заказа и избежание проблем с выбором оплаты;
- консультация, обеспечиваемая онлайн чатом;
- приятный и эстетичный дизайн с дружественным интерфейсом;
- быстрая система регистрации, которая создает для каждого клиента собственную «корзину» товаров;
- система отзывов, которая предоставляет возможность, написать расширенную характеристику и поставить оценку, что повышает доверие покупателей.

1.2 Анализ популярных интернет-магазинов

В рамках подобранной был проведен анализ популярных и непопулярных сайтов интернет-магазинов современного рынка. Широкий спектр программных аналогов помог сделать грамотную оценку и определить все достоинства и недостатки применяющихся подходов разработки. В качестве объектов исследования были рассмотрены следующие сервисы: AliExpress, Wildberries, Ивтекс 37.

1.2.1 Интернет-магазин AliExpress

AliExpress является самым распространенным интернет-магазином, занимая лидирующие позиции в авторитетных рейтингах по всему миру, включая Россию. Количество ежедневных посетителей на сегодняшний день составляет более 9 млн. человек. Сайт обеспечивает возможность для осуществления оптовых и розничных покупок, позволяя экономить существенные суммы денег. Что касается оптимизации самого сайта, то следует отметить следующие преимущества:

- информативное описание товара;
- возможность быстрого оформления заказ;
- дизайн соответствует современным требованиям интернет магазинов и не включает ничего личного;
- гибкая система скидок, закрепляющаяся за пользователем (к аккаунту привязываются скидки, купоны, промокоды и другие бонусные системы);
- возможность оставлять открытые развернутые отзывы с выставлением оценок.

Все эти критерии делают использование сайта более комфортным и удобным, обеспечивая ряд весомых преимуществ. На некоторые из подобных критериев стоит обратить внимание для реализации своего программного

обеспечения сайта интернет-магазина. Внешний дизайн описываемого сайта представлен на рисунке 1.

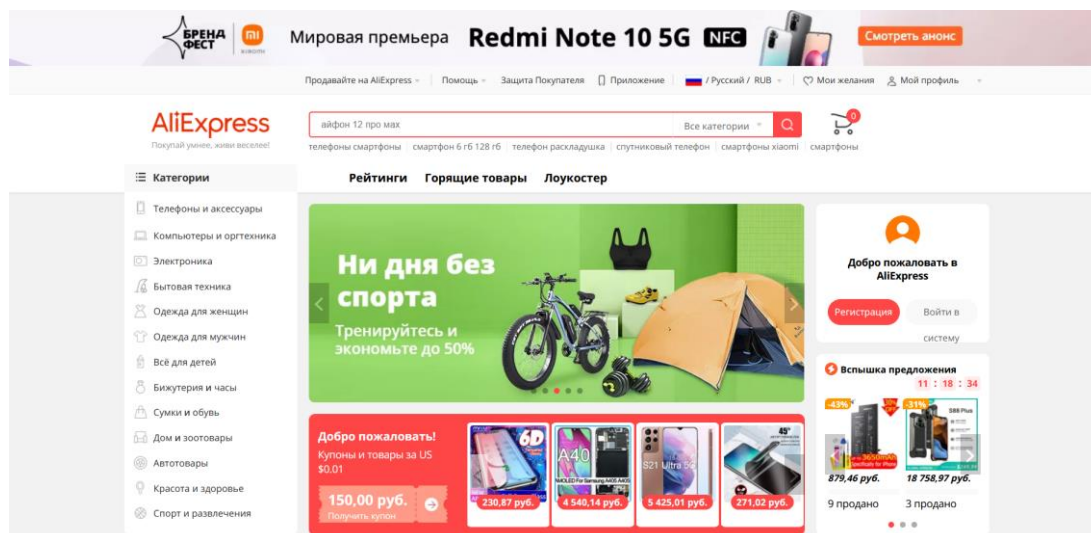


Рисунок 1 – Начальная страница сайта интернет-магазина «AliExpress»

Но даже у такого сайта с мировой известностью можно выделить некоторые недостатки:

- несмотря на русскую локализацию, можно найти множество опечаток, неправильно переведенных позиций, которые могут ввести в заблуждение;
- неудобная система поиска товаров, из-за чего без знаний точного названия найти подходящий продукт достаточно затруднительно;
- отсутствие онлайн-поддержки, за счет чего ответов от администрации сервиса приходится ждать довольно продолжительное количество времени.

1.2.2 Интернет-магазин Wildberries

Популярный представитель торговых онлайн-сервисов с собственными пунктами выдачи товаров. Сайт магазина отлично оптимизирован, все основные требования и критерии были соблюдены. Отдельно стоит выделить:

- реализация поиска, которая происходит быстро и без задержек, а также без замечаний по результатам выдачи;
- возможность оставлять развернутые отзывы с фотографиями;
- для просмотра товара, отложенного в «корзину», необязательно покидать страничку, с которой был сделан заказ, так как все товары отображаются во всплывающем окне при наведении;
- дизайн стильный, удобный и практичный.

Внешний вид сайта изображен на рисунке 2.

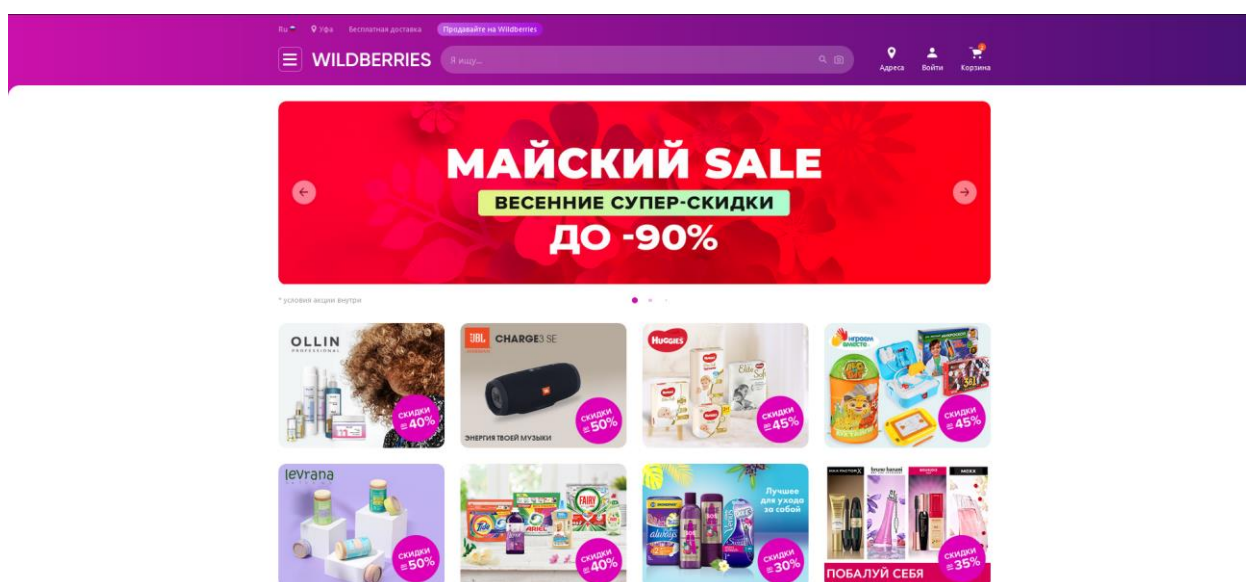


Рисунок 2 – Главная страница интернет-магазина «Wildberries»

Но есть несколько нюансов, которые могли бы улучшить оптимизацию для взаимодействия клиентов:

- главное меню сайта слишком громоздкое, из-за чего следует выполнить группировку схожих категорий;
- из-за маленького количество параметров в фильтре каталога потребителю сложно охватить важные характеристики;
- регистрация осуществляется лишь по номеру телефона.

1.2.3 Интернет-магазин Ивтекс 37

Сайт получил слабый дизайн, что заметно с первого взгляда: нечитаемые шрифты, подобранная стилистика и цветовые решения. Из качественных решений стоит выделить:

- наличие онлайн-чата с администратором магазина;
- отдельное окно «корзины» с указанием общей суммы выбранных товаров прямо на страницы выбора продукции;
- оптимизированная система поиска с подбором заданных товаров.

Главная страница Ивтекс 37 изображена на рисунке 3.

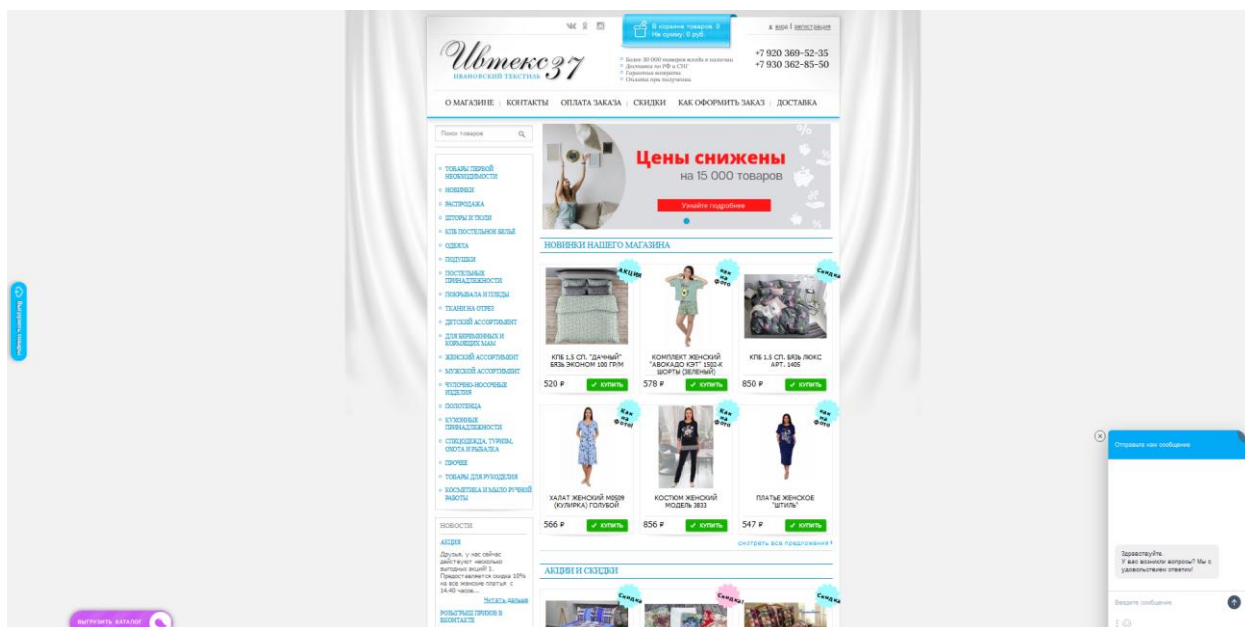


Рисунок 3 – Главная страница интернет-магазина «Ивтекс 37»

Существенных недостатков помимо дизайна оказалось больше:

- отсутствие фильтров для удобного поиска;
- нет возможности для публикации развернутых отзывов с фотографиями;
- проблемы с оформлением доставки и выбором оплаты;
- не все товары имеют описание и характеристики;
- много пустого пространства на страницах интернет-магазина.

1.2.4 Сравнительный анализ рассмотренных программных аналогов

Для проведения сравнительного анализа были выделены следующие критерии оценки:

- полнота указанной информации о товарах;
- четкое разделении продукции по категориям;
- оптимизированная система поиска с помощью «поисковой строки»;
- удобство оформления заказа;
- вариативность выбора способа оплаты;
- дизайн, удовлетворяющий современным стандартам;
- удобная настройка «корзины» товаров;
- полноценная система отзывов с возможностью публикации фотографий;
- наличие онлайн-чата для быстрой обратной связи;
- наличие фильтров для поиска товара;
- удобная система регистрации.

Результаты сравнительного анализа представлены в таблице 1.

Таблица 1 – Сравнительный анализ

Критерии для сравнения	«AliExpress»	«Wildberries»	«Ивтекс 37»
Полнота указанной информации о товарах	Да	Да	Нет
Четкое разделении продукции по категориям	Да	Да	Да
Оптимизированная система поиска с помощью «поисковой строки»	Нет	Да	Да
Удобство оформления заказа	Да	Да	Нет

Продолжение таблицы 1

Критерии для сравнения	«AliExpress»	«Wildberries»	«Ивтекс 37»
Вариативность выбора способа оплаты	Да	Да	Нет
Дизайн, удовлетворяющий современным стандартам	Да	Да	Нет
Удобная настройка «корзины» товаров	Нет	Да	Да
Полноценная система отзывов	Да	Да	Нет
Наличие быстрой обратной связи	Нет	Нет	Да
Наличие фильтров для поиска товара	Да	Да	Нет
Удобная система регистрации	Да	Нет	Да

Таким образом, в результате сравнения аналоговых сайтов интернет-магазина, было установлено, что даже самые популярные мировые представители имеют свои недостатки, например, в виде отсутствия быстрой обратной связи без существенных задержек по времени.

Кроме того, клиенты интернет-магазина «AliExpress» могут столкнуться с проблемами поиска через поисковые запросы, а также придется постоянно переходить на новую страницу «корзины» для отслеживания стоимости отобранных товаров. А клиентам «Wildberries» придется столкнуться с неоптимальной системой регистрации, осуществляемой лишь по номеру телефона без альтернативных вариантов.

2 Анализ требований интернет-магазина

2.1 Анализ нефункциональных требований

Сайт должен поддерживать работу и запускаться с любого современного браузера. Для разработки сайта был выбран язык разметки HTML. Он позволяет создавать простой интерфейс и приятный дизайн. Однако одного языка HTML для разработки интерфейса мало. Для более качественного и приятного на вид сайта применяются такие средства как скриптовый язык JavaScript, каскадные таблицы стилей CSS, и серверный язык PHP [2].

PHP – язык, который специально разработан для работы в Интернете. PHP обладает многофункциональным и четким синтаксисом. Для исполнения запросов на веб-сервере с поддержкой PHP должен быть установлен еще один сервер, который будет воспринимать запросы к базе данных, поступающие из сценариев, и исполнять их [22].

Для разработки программного обеспечения были выделены следующие нефункциональные требования:

- обеспечить простоту и практичность использования с согласованностью пользовательского интерфейса;
- избежать появления ошибок и возникновения аварийных ситуаций;
- обеспечить совместимость с различными типами устройств.

2.2 Функциональные требования и разработка диаграмм

Модель бизнес-прецедентов описывает бизнес-процессы с точки зрения внешнего пользователя, т.е. отражает взгляд на деятельность организации извне [4].

Проектирование любого интернет-магазина лучше всего начинать с построения диаграммы прецедентов, описывающей внешнюю границу сайта.

Наша система должна позволять клиенту совершать ряд определенных действий:

- зарегистрироваться на сайте;
- пройти авторизацию на сайте;
- воспользоваться строкой поиска;
- выбрать товар;
- задать вопрос техподдержке.

Для администратора информационная система предоставляет следующий функционал:

- удалить отзывы;
- изменить данные товара;
- просмотреть информацию о доставках;
- добавить новый товар.

Проектирование обеспечивается в среде разработки ArgoUML на основе вышеописанных требований. В связи с этим была составлена диаграмма прецедентов, описывающая внешнюю границу системы. Актеры имеют имена «Посетитель сайта», «Администратор сайта».

На рисунке 4 представлена диаграмма вариантов использования для посетителя сайта.

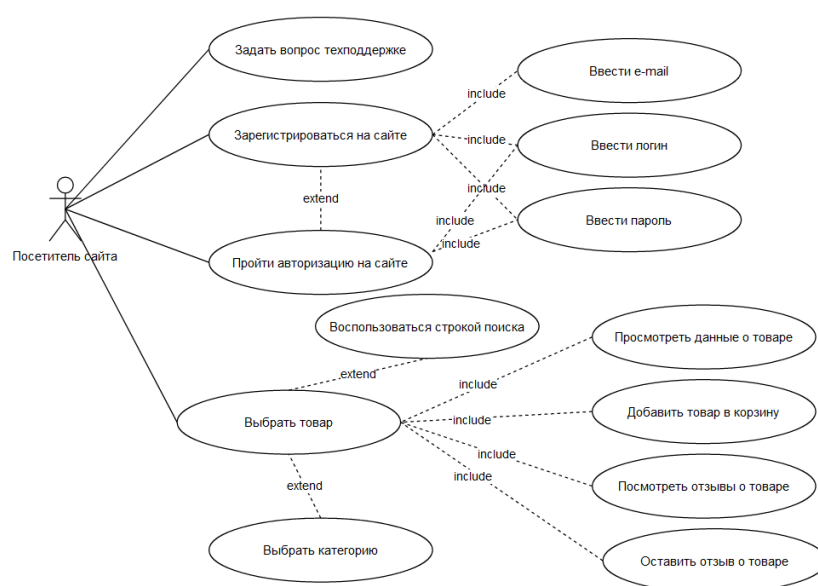


Рисунок 4 – Диаграмма вариантов использования для посетителя сайта

В роли актера выступает посетитель сайта, которому предоставляется возможность для регистрации на сайте. В случае же, если пользователь уже зарегистрирован, то он может пройти авторизацию, указав свои данные в виде логина и пароля. Для прохождения регистрации необходимо заполнять три обязательных поля ввода: e-mail, логин, пароль.

Также в случае возникновения вопросов посетитель может обратиться к техподдержке, получив быстрый ответ. Выбор товара может осуществляться как напрямую, так и через выбор категории или использование поисковой строки. Перейдя к самому продукту можно узнать информацию о нем, посмотреть отзывы других покупателей, а также оставить свое мнение. Для осуществления покупки товар добавляется в корзину.

На рисунке 5 представлена диаграмма вариантов использования для администратора сайта.



Рисунок 5 – Диаграмма вариантов использования для администратора сайта

Для получения доступа к интерфейсу администратора необходимо пройти авторизацию под определенным логином и паролем. Администратор сайта может управлять отзывами, удаляя их. Также он может просматривать

информацию о доставках.

Кроме того, в админ-панели менять данные товара, такие как описание, размеры, название и изображения. Для администратора предоставляется возможность добавления новой продукции с заполнением всех данных.

На двух диаграммах на рисунках 17 и 18 представлены функциональные требования реализации программного обеспечения для интернет-магазина. А для понимания отдельного класса варианта использования в реализации алгоритмов потоков управления построим диаграмму последовательности действий.

Диаграммы последовательности действий играют важную роль в понимании процессов реализации алгоритмов выполнения операций классов и потоков управления в моделируемой системе. Используемые для этой цели традиционные блок-схемы алгоритмов обладают серьёзными ограничениями в представлении параллельных процессов и их синхронизации [9]. В таком случае целесообразно построить диаграмму последовательности действий для некоторых отдельных процессов.

Последовательность действий при авторизации администратора изображена на рисунке 6.

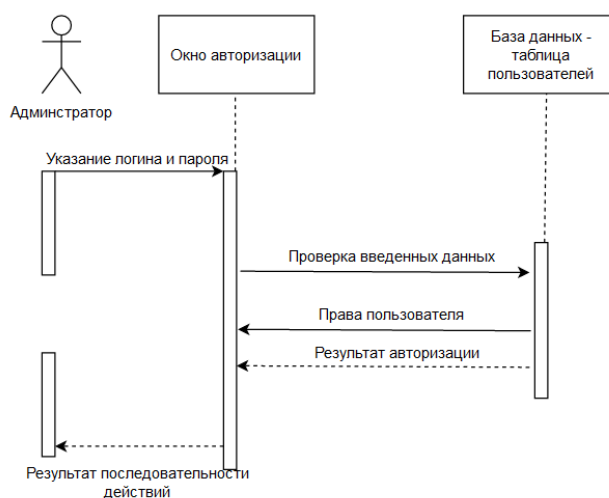


Рисунок 6 – Диаграмма последовательности действий
при авторизации администратора

В роли актера выступает администратор, проходящий авторизацию на сайте для доступа к «Админ-панели». При введении логина и пароля он попадает на окно авторизации, после чего проходит проверка введенных данных.

В ходе проверки система обращается к базе данных, переходя в таблицу пользователей. Если логин и пароль введены верно, программное обеспечение определяет права данного пользователя, в нашем случае речь идет о статусе администратора. Права пользователя и результаты проверки проходят через окно авторизации, передавая информацию актеру.

На рисунке 7 представлена диаграмма последовательности действий по добавлению нового товара.

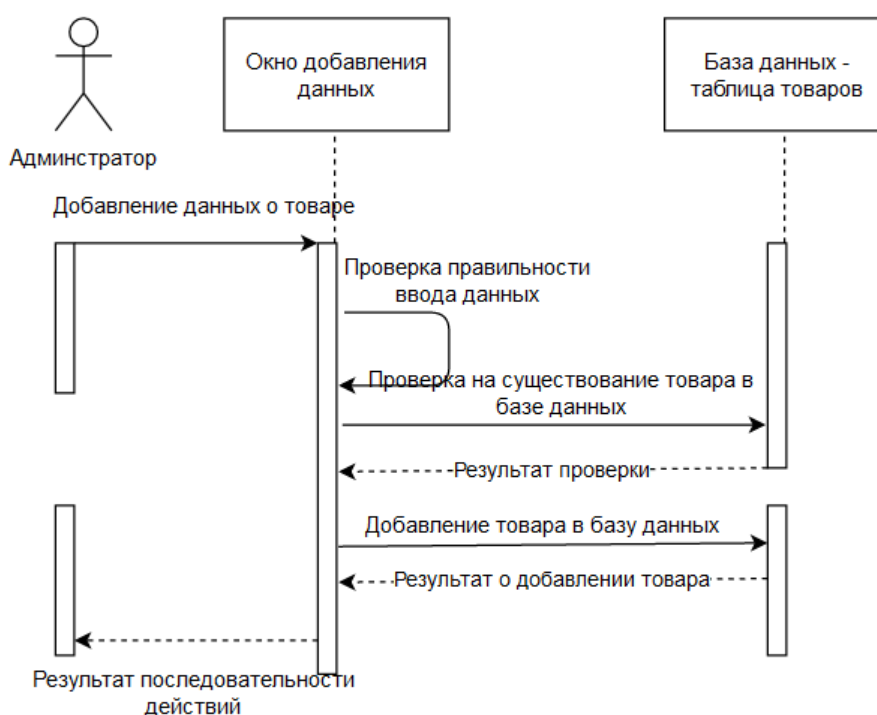


Рисунок 7 – Диаграмма последовательности действий
при добавлении нового товара

При добавлении новых данных о товаре в первую очередь в соответствующем окне осуществляется проверка корректности введенной

информации. По ее истечению программное обеспечение обращается к базе данных с целью проверки, не были ли введены эти же данные ранее. В случае, если подобная информация будет найдена, то в окне добавления данных появится результат проверки с сообщением о том, что такой товар уже существует. Далее товар будет добавлен в базу данных, а результат об успешном добавлении появится в окне.

Диаграмма последовательности действий по удалению товара представлена на рисунке 8.

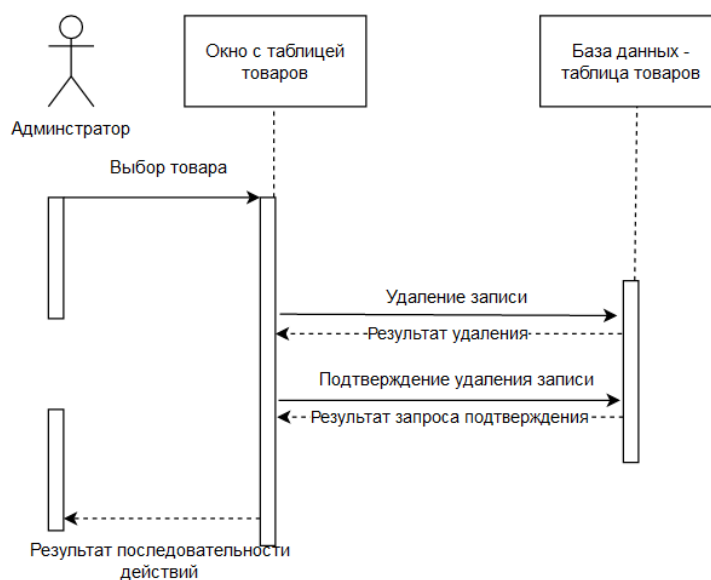


Рисунок 8 – Диаграмма последовательности действий при удалении товара

Последовательность удаления товара администратором начинается с выбора товара, осуществляемого в окне с таблицей товаров. После этого осуществляется удаление записи с обращением к базе данных. Как только администратор подтвердит запрос на удаление, он получит уведомление об успешном выполнении задачи.

2.3 Описание функциональной модели

Описание функциональной модели позволит определить главные функции программного обеспечения, а также установить компоненты и их

взаимодействие. В качестве методологии моделирования для описания производственных функций применен IDEF0 [25].

Основная функция разработки – осуществление продаж с помощью интернета. Модель, описывающая основную функцию программного обеспечения для интернет-магазина вещей представлена на рисунке 9.

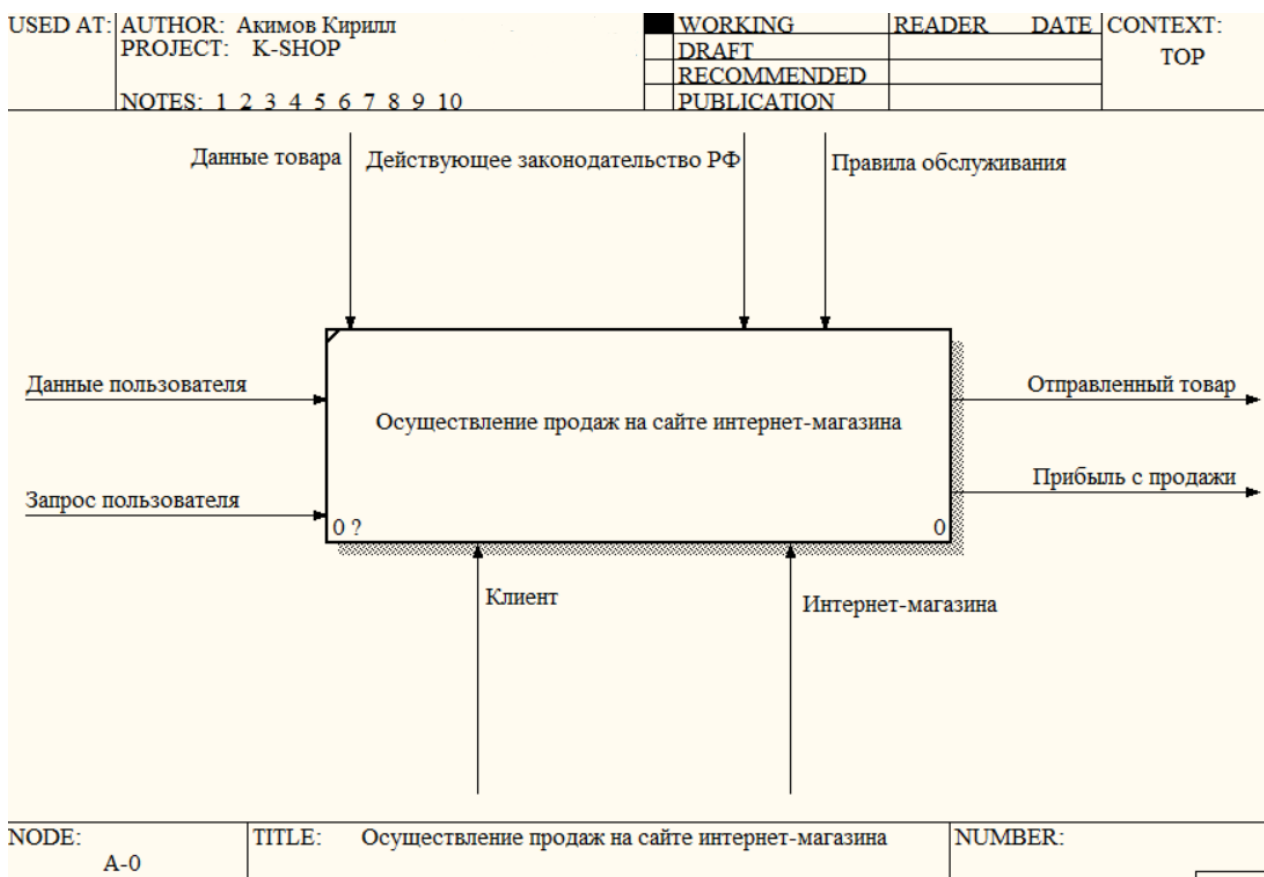


Рисунок 9 – Контекстная диаграмма функциональной модели

Из рисунка 9 видно, чтобы перейти к процессу осуществления продаж на сайте интернет-магазина, необходимо получить перечень товаров, входные данные пользователя, а также его запрос. В качестве управляющих потоков выступают три составляющих – данные товара, действующее законодательство РФ, влияющее на деятельность организации, и правила обслуживания, определенные непосредственно этим интернет-магазином. На выходе появляются отправленный товар и прибыль, полученная с продаж.

Для более подробного анализа необходимо произвести декомпозицию, разбив главный процесс на несколько компонентов.

Функциональная модель на втором уровне декомпозиции представлена на рисунке 10.

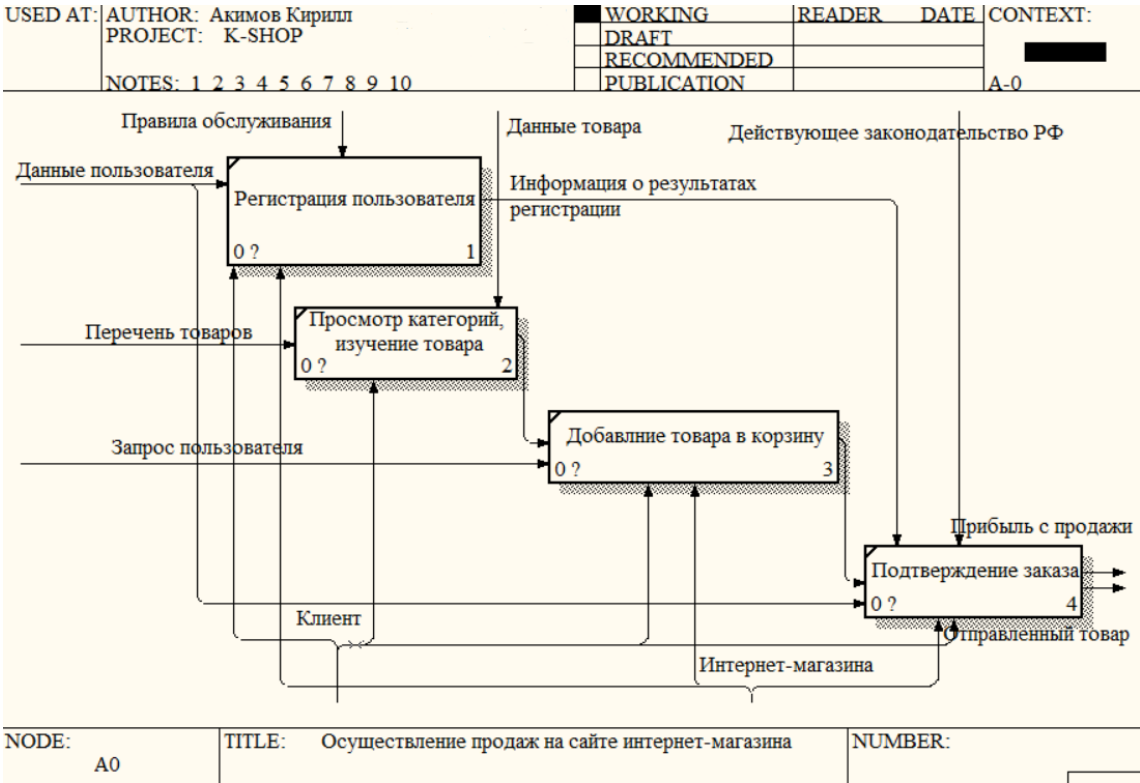


Рисунок 10 – Функциональная модель на втором уровне декомпозиции

Второй уровень декомпозиции подразделяется на 4 отдельных подпроцесса. Изначально пользователь может пройти регистрацию, необходимость которой заключается при оформлении заявки. Но перед подтверждением производится просмотр категорий и изучение товара с последующим его добавлением в корзину.

Декомпозиция бизнес-процесса регистрации пользователя состоит из 3 подпроцессов, среди которых имеются запрос необходимых данных о пользователе, проверка внесенной информации по базе данных, добавление пользователя в базу данных. Реализация декомпозиции представлена на рисунке 11.

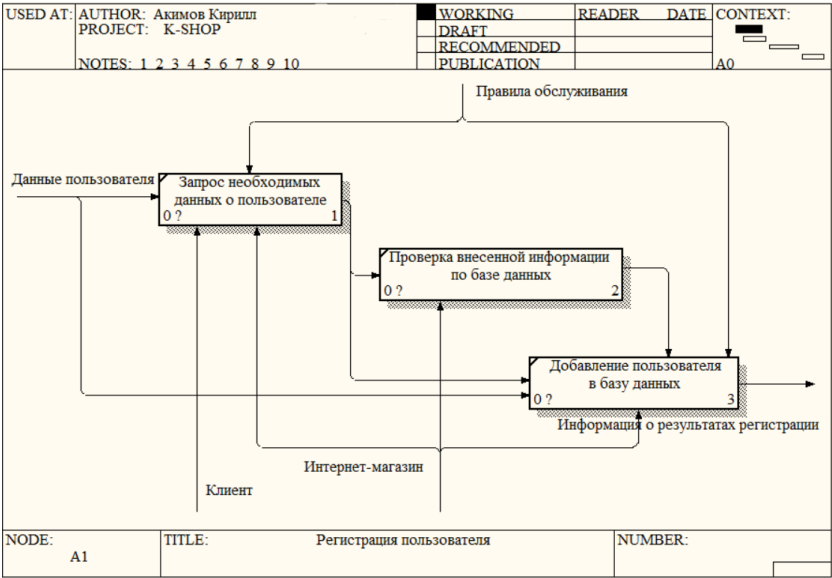


Рисунок 11 – Декомпозиция бизнес-процесса регистрации пользователя

При регистрации пользователя система запрашивает его данные. После их получения происходит проверка внесенной информации на наличие этих же данных в базе. Далее происходит добавление пользователя в базу данных. На выходе же система получает информацию о результатах регистрации.

Декомпозиция бизнес-процесса добавления товара в корзину делится на 2 подпроцесса, она изображена на рисунке 12.

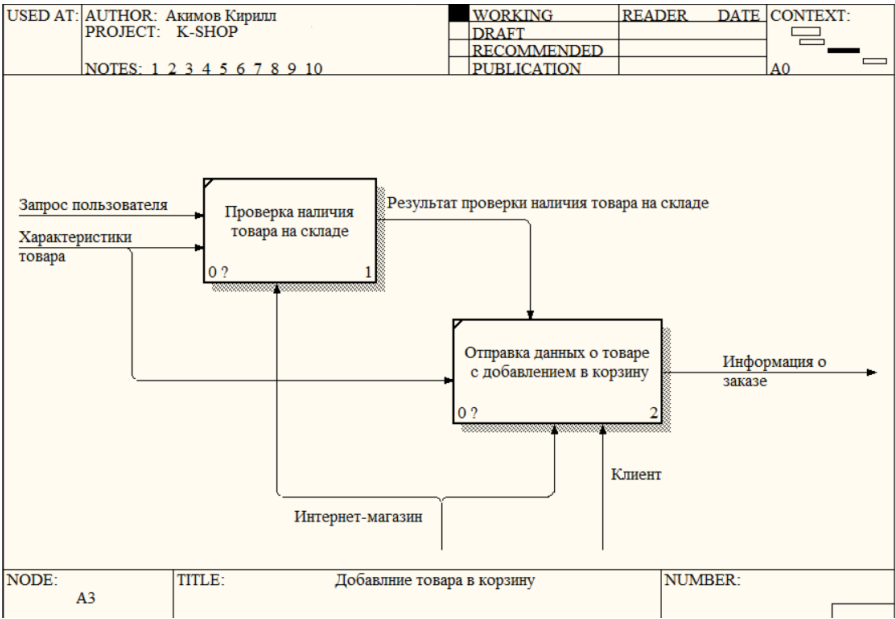


Рисунок 12 – Декомпозиция бизнес-процесса добавления товара в корзину

При попытке добавления товара в корзину по запросу пользователя система проводит проверку о наличии товара на складе, в последующем отправляя данные о товаре с добавлением его в корзину, получая на выходе информацию о потенциальном заказе.

Декомпозиция бизнес-процесса подтверждения заказа, в свою очередь, делится на 3 подпроцесса, которые изображены на рисунке 13.

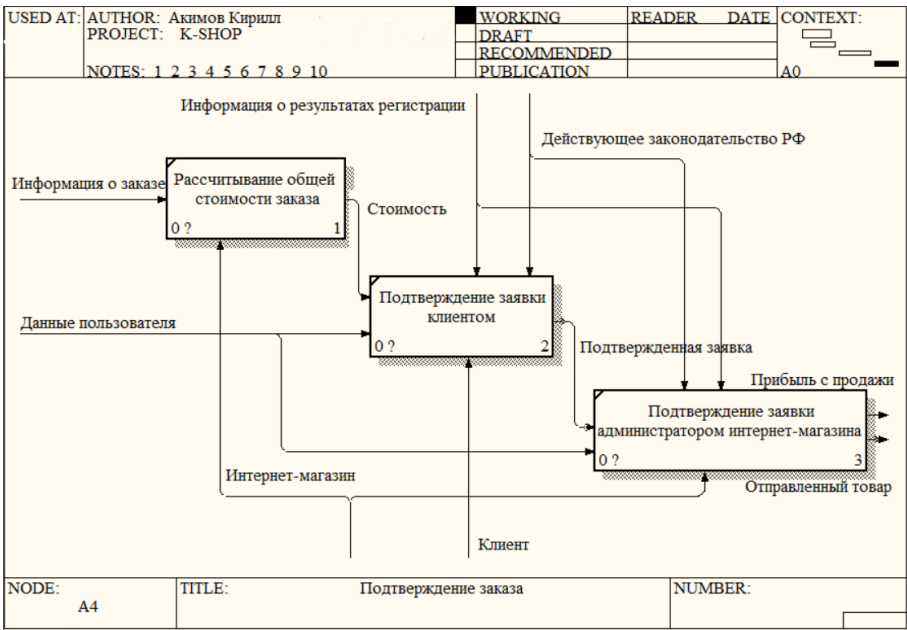


Рисунок 13 – Декомпозиция бизнес-процесса подтверждения заказа

Подтверждение заказа включает расчет общей стоимости на основе информации о заказе, подтверждение заявки клиентом, а также подтверждение заявки администратором интернет-магазина, что и является завершающим шагом.

3 Проектирование интернет-магазина вещей K-SHOP

3.1 Описание инструментальных программных средств разработки

В качестве платформы для разработки и создания интернет-магазина выступает операционная система Windows 10, обладающая простым внешним оформлением, которое упрощает взаимодействие с конечным пользователем.

В качестве редактора кода для создания сайта был выбран Sublime Text. Это кроссплатформенный проприетарный текстовый редактор с большим количеством полезных функций, которые также используются и для web-страниц [27].

Для разработки сайта был выбран язык разметки HTML. Он позволяет создавать простой интерфейс и приятный дизайн. Этот язык разметки представляет собой простой набор кодов, которые описывают структуру документа. Язык HTML интерпретируется браузерами. В результате текст форматируется и отображается на экране монитора компьютера или мобильного устройства [11].

Однако одного языка HTML для разработки интерфейса мало. Для более качественного и приятного на вид сайта применяются такие средства как скриптовый язык Java Script, каскадные таблицы стилей CSS, и серверный язык PHP.

Для исполнения запросов на веб-сервере с поддержкой PHP должен быть установлен еще один сервер, который будет воспринимать запросы к базе данных, поступающие из сценариев, и исполнять их. Для создания и работы с базой данных в локальном доступе были выбраны возможности PHP и MySQL и для этого необходим Open Server. Программный комплекс имеет внушительный набор обеспечения сервера, комфортный, многофункциональный продуманный интерфейс, обладает мощными возможностями по выполнению функций администрирования и отладке

компонентов. Платформа используется для различных задач с целью разработки, отладки и тестирования веб-проектов [10].

3.2 Базовая архитектура Web-сайта интернет-магазина

Для разработки сайта с поддержкой базы данных использовалась схема разделения данных приложения MVC. Model-View-Controller (MVC, «Модель-Представление-Контроллер», «Модель-Вид-Контроллер»).

Шаблон MVC описывает способ представления подхода к технической разработке, отделяющий бизнес-логику от интерфейса пользователя [22]. В результате данного подхода разработка приложения значительно упрощается, а также обеспечивается больше возможностей для последующего масштабирования проекта. Концептуальная схема шаблона MVC представлена на рисунке 14.

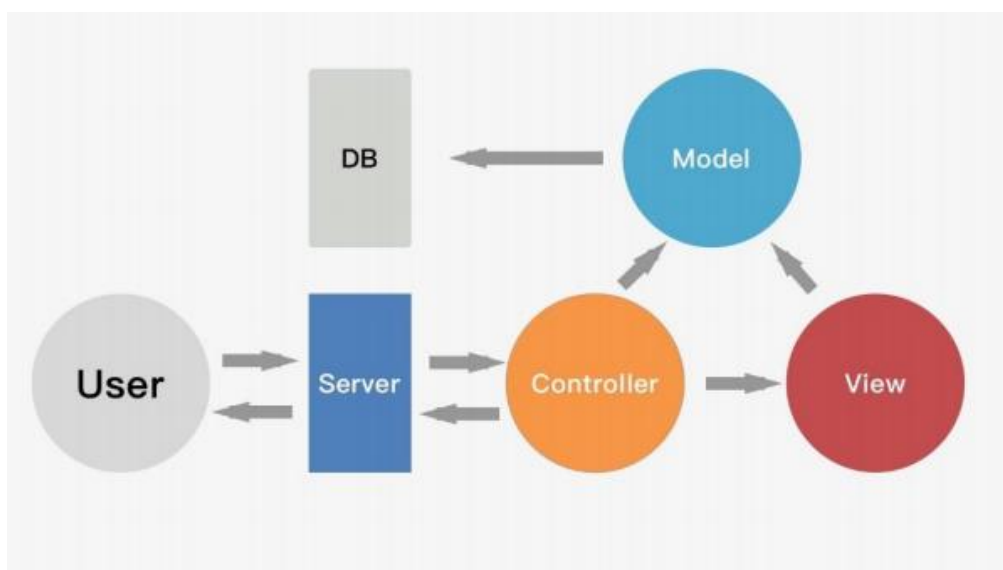


Рисунок 14 – Схема шаблона MVC сайта интернет-магазина вещей

Последовательность работы программного обеспечения интернет-магазина можно описать следующим образом: как только пользователь заходит на разработанный сайт, инициализация объекта создает экземпляр программного обеспечения, обеспечивая его запуск на выполнение. Посетитель же, в свою очередь, видит перед собой главную страницу.

Программное обеспечение считывает все запросы пользователя, выполняя контроллер действие. Главная же страница инициализируется по умолчанию, выполняя стандартное действие (index). Реализуется экземпляр-контроллер, который выполняет действие. После этого, действие формирует представление с данными, полученными из модели и выводит результат пользователю [7].

3.3 Логическая модель базы данных

Логическая модель базы данных обеспечивает визуальное графическое представление структур данных, их атрибутов и связей. В результате проектирования программного обеспечения интернет-магазина вещей K-SHOP была спроектирована сущность модели в методологии IDEF1X. Основными компонентами логической модели являются: сущности, атрибуты и связи между сущностями.

Построение модели осуществлялось с помощью программного средства СА ERwin Data Modeler. Логическая модель базы данных интернет-магазина представлена на рисунке 15.

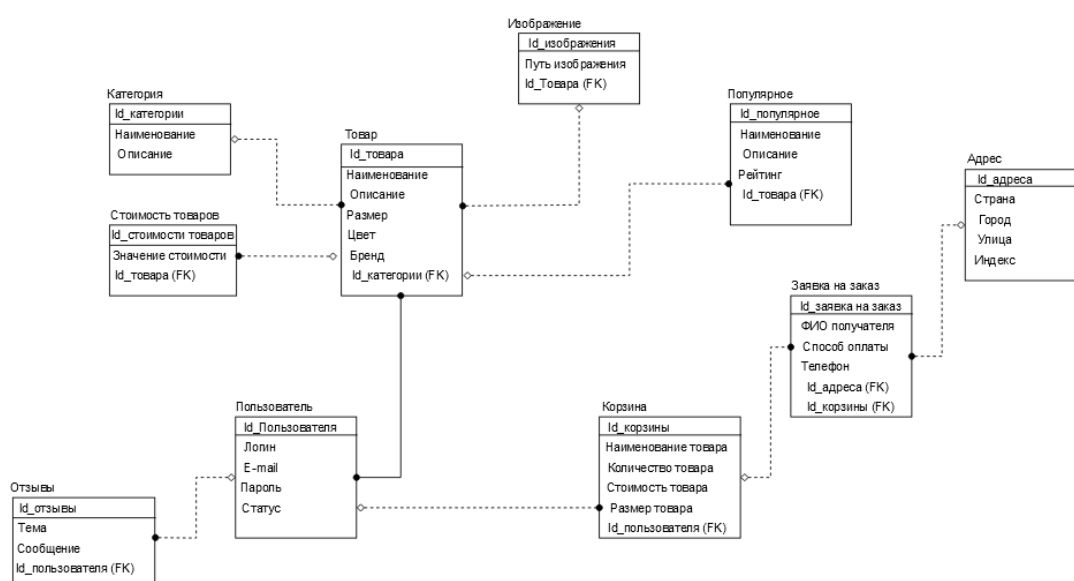


Рисунок 15 – Логическая модель базы данных интернет-магазина

На рисунке представлены сущности интернет-магазина, а также установлены связи их взаимодействия. Для полного описания реализации объектов необходимо построить физическую модель базы данных.

3.4 Физическая модель базы данных

Физическая модель базы данных содержит все детали, необходимые конкретной системе управления базами данных (СУБД) для создания базы: названия таблиц и столбцов, типы данных, ключи (внешние и первичные). Физическая модель строится на основе логической, включая все ограничения, которые определяются выбранной СУБД [23]. Физическая модель базы данных интернет-магазина вещей K-SHOP представлена на рисунке 16.

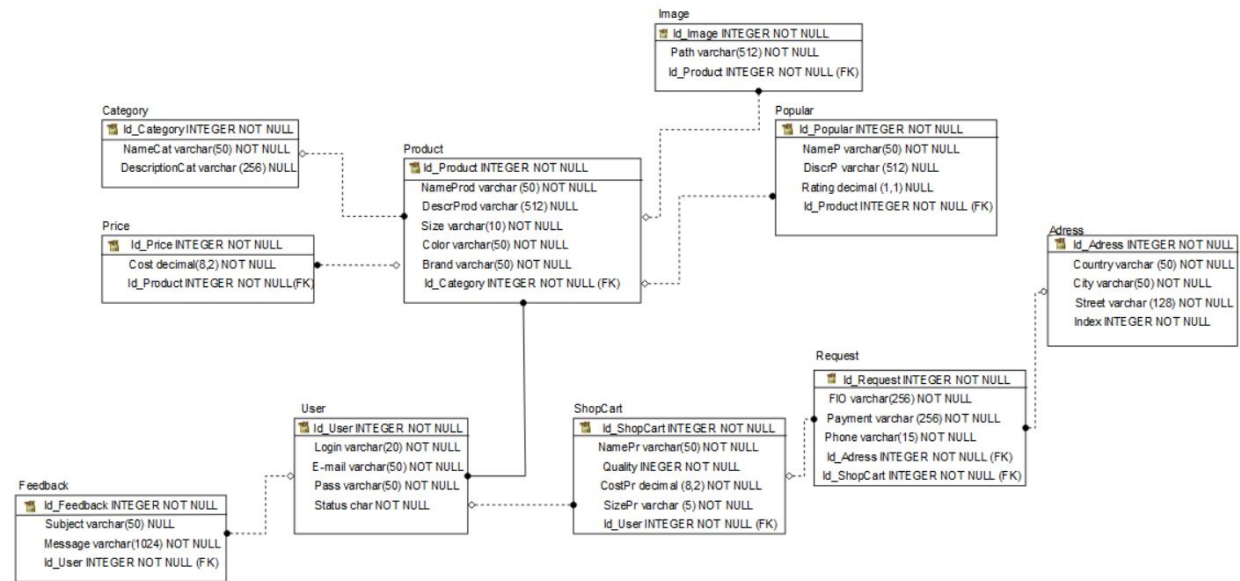


Рисунок 16 – Физическая модель базы данных интернет-магазина

Из данного рисунка физической модели видно, какие таблицы используются в базе данных, а также столбцы и типы данных с учетом всех ограничений, установленных для программного обеспечения интернет-магазина.

В данном проекте реализовано два основных класса: контроллеры и модели. Также рассмотрен один дополнительный класс, включающий компоненты.

3.5.2 Построение диаграммы деятельности для интернет-магазина вещей K-SHOP

Процесс анализа и проектирования систем является последовательностью итераций разработки как нисходящего, так и восходящего типа диаграмм, в состав которых входит и диаграмма деятельности [21]. Для моделирования организационных процессов зарегистрированных пользователей программного обеспечения интернет-магазина вещей была построена диаграмма деятельности, изображенная на рисунке 18.

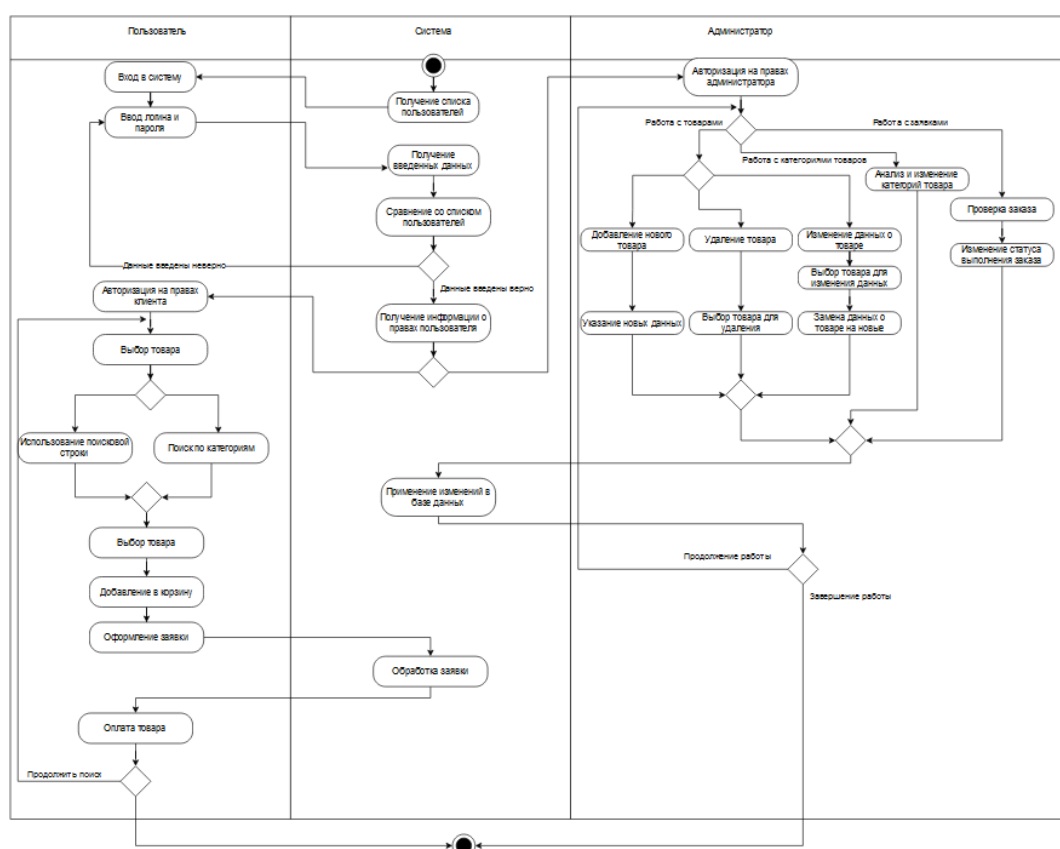


Рисунок 18 – Диаграмма деятельности зарегистрированных пользователей сайта интернет-магазина

Вся последовательная работа начинается с того, что система получает список пользователей, хранящихся в базе данных. Поэтому при попытке входа в личный кабинет посредством ввода логина и пароля система при получении введенных данных проводит сравнение с имеющимся списком пользователей. Если данные введены неверно, то пользователя просят проверить правильность введенной информации. Если же все верно, то система получает информацию о правах пользователя.

Авторизация на правах администратора обеспечивает возможность для нескольких видов взаимодействия и проведения работ: работа с товарами, работа с категориями, работа с заявками с проверкой заказа и изменением его статуса выполнения. В свою очередь, работа с товарами предоставляет возможность для добавления новых товаров и с указанием всех необходимых данных, удаления товаров с выбором нужной продукции, изменения данных товара для необходимого продукта с заменой информации. Все манипуляции в последующем сопровождаются тем, что действия фиксируются системой с применением изменений в базе данных. Далее администратор может продолжить работу, либо завершить ее [9].

Авторизация на правах клиента предоставляет возможность выбора товара посредством использования поисковой строки или же фильтрации категорий, характеристик. Изучив и выбрав понравившийся товар, клиент может добавить его в корзину, в последующем оформив заявку. Далее происходит обработка поступившей заявки системой, после чего пользователь может произвести оплату товара. Также клиент может продолжить действия на сайте или же завершить работу [19].

3.5.3. Диаграмма кооперации интернет-магазина вещей

Диаграмма кооперации обеспечивает возможность представить не только взаимодействие компонентов, но и структурные отношения. Данный вид диаграмм способствует отображению потока событий в конкретном сценарии [16]. В рамках данной предметной области необходимо рассмотреть

сценарий подачи заявки для оформления покупки в программном обеспечении для интернет-магазина. Диаграмма кооперации изображена на рисунке 19.

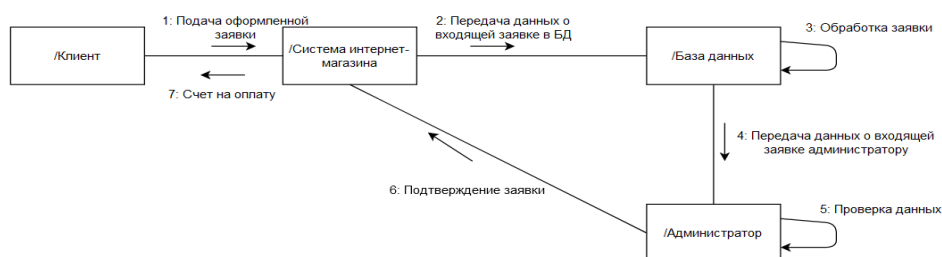


Рисунок 19 – Диаграмма кооперации для оформления заявки на покупку товара

Данный объект информационной системы представлен 4 компонентами: клиент, система интернет-магазина, база данных, администратор.

Клиент осуществляет подачу заявки, получая в результате ее обработки и прохождения через остальные компоненты, счет на оплату. Система интернет-магазина, получая на вход данные отправленной заявки, обеспечивает передачу полученной информации на сервер базы данных.

Там, в свою очередь, проходит обработка заявки с последующей передачей данных администратору. Объект «Администратор» проводит ручную проверку корректности введенных данных, отправляя на вход системе подтверждение заявки [6].

3.6 Проектирование пользовательского интерфейса

Пользовательский интерфейс должен обладать привлекательным и современным дизайном. Содержимое сайта, должно включать указанные бизнес-задачами. Также акцент необходимо делать на понятность, чтобы пользователь мог спокойно передвигаться по сайту [24]. Наличие дружелюбного интерфейса – одна из главных задач, которая обеспечивает положительную конверсию [4]. На основе приведенной информации был

составлен графический интерфейс программного обеспечения для интернет-магазина вещей K-SHOP.

При входе на сайт пользователь попадает на главную страницу, из которой можно перейти на другие необходимые страницы. Интерфейс главной страницы представлен на рисунке 20.

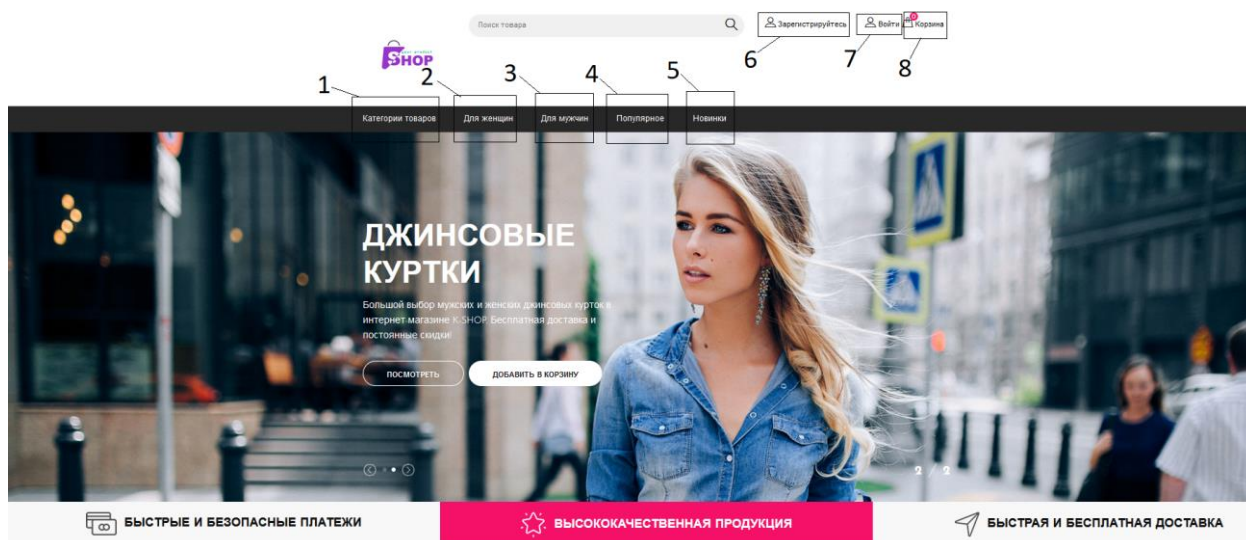


Рисунок 20 – Главная страница сайта

При попадании на главную страницу пользователь может войти в личный кабинет, пройти регистрацию, а также переместиться в корзину. Помимо прочего, ему предоставляется возможность для перехода по категориям, посредством кнопок, представленных на панели навигации. Интерфейс главной страницы содержит:

- 1 – кнопка для перехода в раздел всех категорий;
- 2 – кнопка для перехода в раздел для женщин;
- 3 – кнопка для перехода в раздел для мужчин;
- 4 – кнопка для перехода в раздел популярное;
- 5 – кнопка для перехода в раздел новинки;
- 6 – кнопка для совершения регистрации;
- 7 – кнопка для входа в личных кабинет;
- 8 – кнопка для просмотра корзины.

Для получения полного функционала пользователю необходимо пройти регистрацию. Окно регистрации изображено на рисунке 21.

Регистрация

1. Логин * 1

2. E-mail * 2

3. Пароль * 3

Зарегистрироваться 4

Рисунок 21 – Окно регистрации

Окно регистрации включает следующий набор элементов:

- 1 – поле для ввода логина;
- 2 – поле для ввода e-mail;
- 3 – поле для ввода пароля;
- 4 – кнопка для подтверждения регистрации.

После прохождения регистрации пользователь может авторизоваться. Окно авторизации показано на рисунке 22.

Авторизация

1. E-mail * 1

2. Пароль * 2

Запомнить 3

Войти 4

Рисунок 22 – Окно авторизации

Окно авторизации состоит из элементов:

- 1 – поле для ввода e-mail;
- 2 – поле для ввода пароля;
- 3 – кнопка запомнить;
- 4 – кнопка для входа в личный кабинет.

Переходя в раздел категорий товара, пользователь может посмотреть представленный каталог, а также выбрать подходящую одежду по критериям. Раздел категорий представлен на рисунке 23.

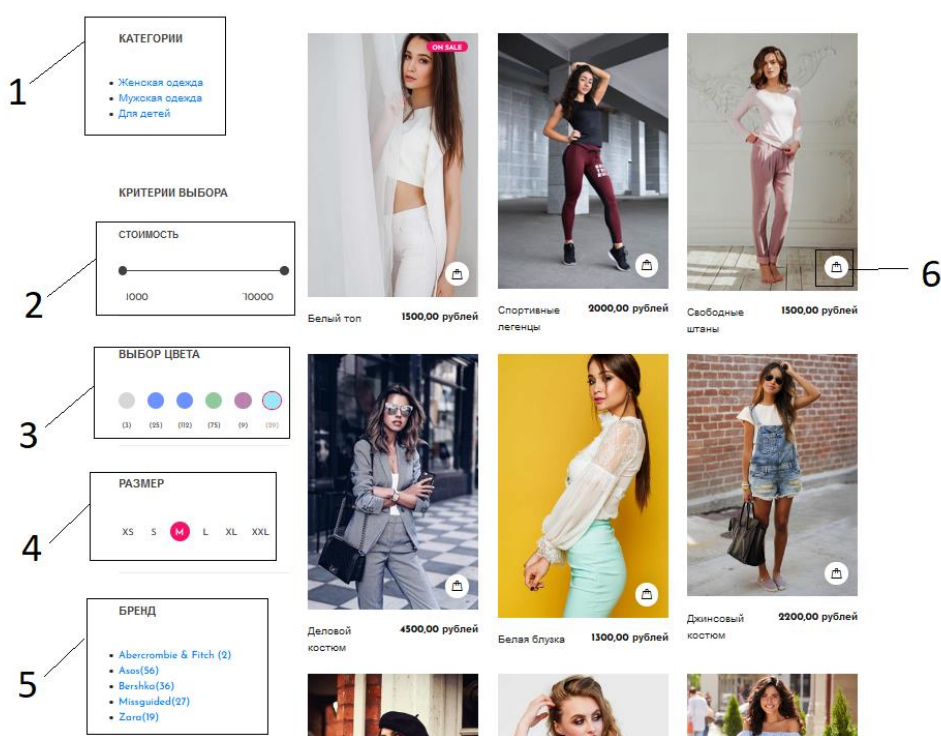


Рисунок 23 – Раздел страницы категорий

Интерфейс раздела страницы категорий включает:

- 1 – панель перехода по категориям товаров;
- 2 – ползунок регулировки стоимости товаров;
- 3 – панель выбора цвета одежды;
- 4 – панель выбора размеров одежды;
- 5 – панель выбора бренда одежды;
- 6 – кнопка добавления товара в корзину.

При выборе определенного товара пользователь получает более подробную информацию о нем, попадая на страницу с его описанием. Раздел страницы товара представлен на рисунке 24.

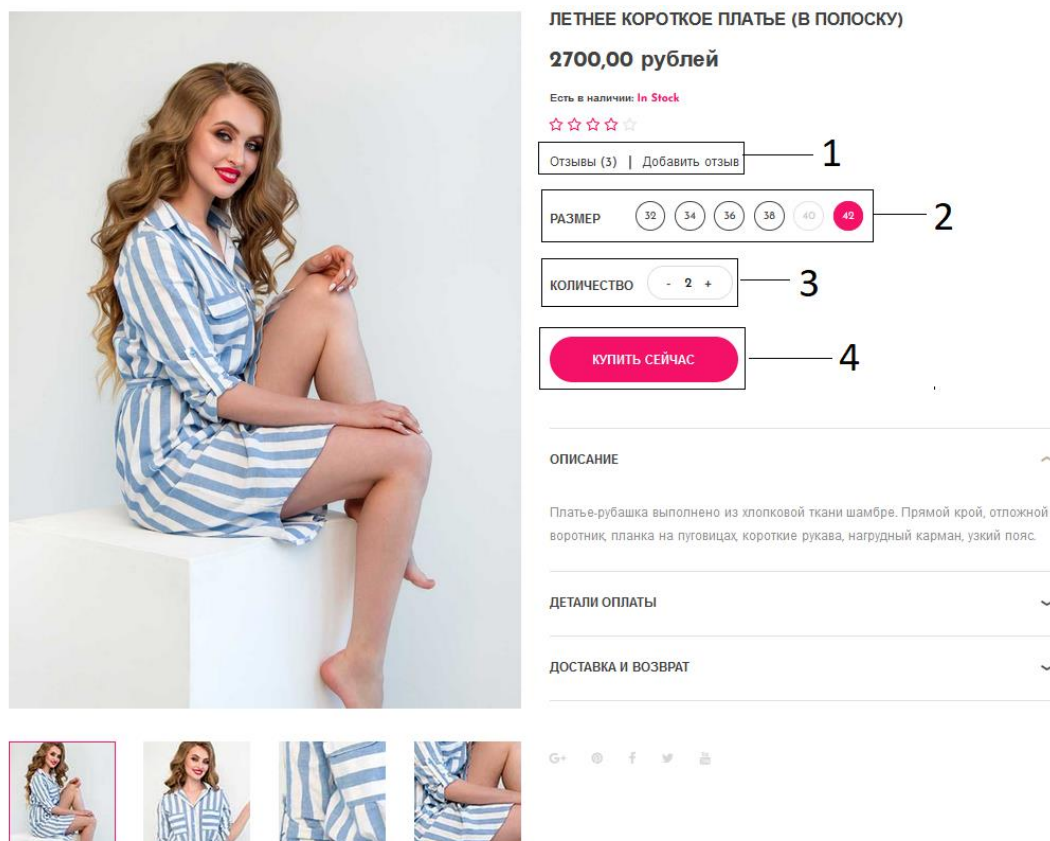


Рисунок 24 – Раздел страницы товара

Графический интерфейс данного раздела включает следующий набор элементов:

- 1 – кнопки для просмотра отзывов, с возможностью оставить свой отзыв;
- 2 – панель кнопок для выбора размера;
- 3 – кнопка для выбора количества покупаемого товара;
- 4 – кнопка покупки товара.

При покупке товара он добавляется в корзину. Интерфейс корзины покупок изображен на рисунке 25.

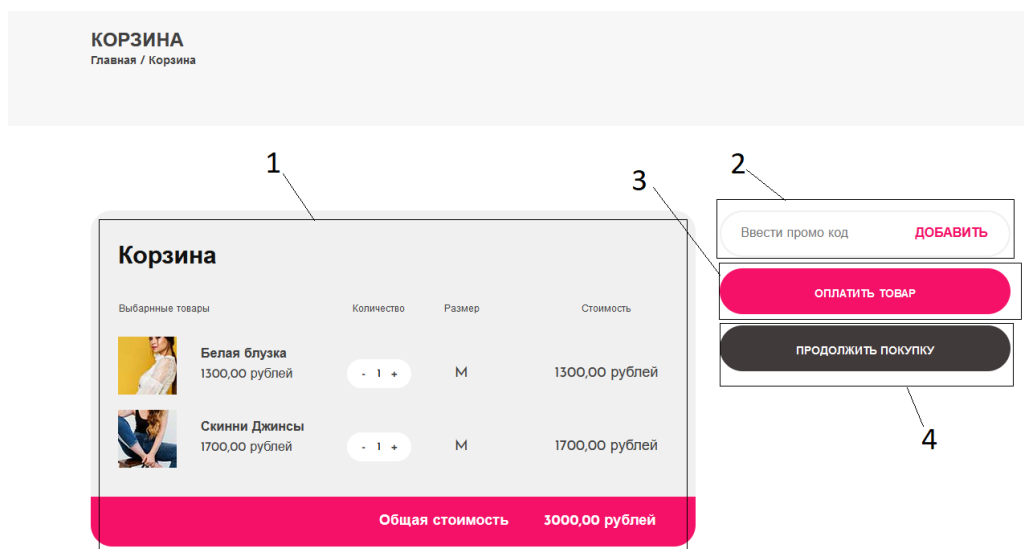


Рисунок 25 – Корзина покупок

В корзине покупок представлен интерфейс с элементами:

- 1 – панель с отображением выбранного товара;
- 2 – поле ввода промо кода для получения скидки;
- 3 – кнопка оплаты товара;
- 4 – кнопка для продолжения покупок, отправляющая в раздел категорий.

При оплате товара клиент попадает на страницу оформления заявки, внешний вид которой представлена на рисунке 26.

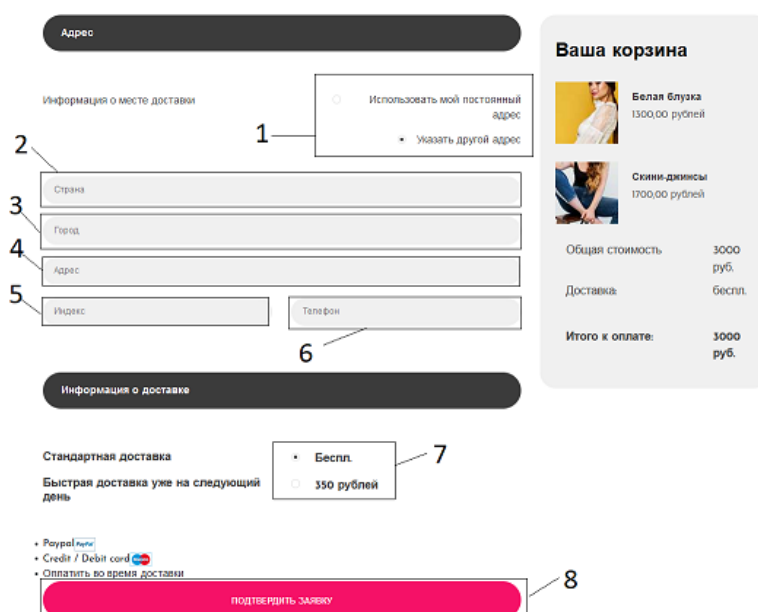


Рисунок 26 – Страница оформления заявки

Страница оформления заявки состоит из следующего набора элементов:

- 1 – поле для выбора возможности использовать старый адрес или ввести новый;
- 2 – поле для ввода страны доставки;
- 3 – поле для ввода города доставки;
- 4 – поле для ввода адреса доставки;
- 5 – поле для ввода индекса доставки;
- 6 – поле для ввода телефона доставки;
- 7 – поле для выбора платной или бесплатной доставки;
- 8 – кнопка для подтверждения заявки.

Также пользователь может напрямую обратиться к техподдержке, задав необходимый вопрос. Интерфейс для связи с поддержкой сайта изображен на рисунке 27.

Пенза, ул. Московская 72
+7 777 777 77 77
k-shop@mail.com

1
Имя

2
E-mail

3
Тема

4
Сообщение

5
ОТПРАВИТЬ СЕЙЧАС

Гранд-Сентрал-Паркуэй
Кладбище Сейнт Майкл
McDonald's
Buccaneer Diner
La Nueva Colombia
Louis Armstrong Middle School
FOOD BAZAAR
Barzola
Oceanic Boil
Jackson Heights
Sagar Chinese
Dera
Госпитальный центр Элмхерст
Elmhurst Hospital Center
New York City Police Department
Отделение полиции

Рисунок 27 – Интерфейс для связи с поддержкой сайта

Интерфейс для связи с поддержкой состоит из:

- 1 – поле для ввода имени;
- 2 – поле для ввода электронной почты;
- 3 – поле для ввода темы письма;

4 – поле для ввода сообщения;

5 – кнопка для отправки сообщения.

Для управления функционалом интернет-магазина предусмотрена специальная панель администратора, доступ к которой обеспечивается после авторизации пользователя со специальными правами. Главная страница админ панели изображена на рисунке 28.

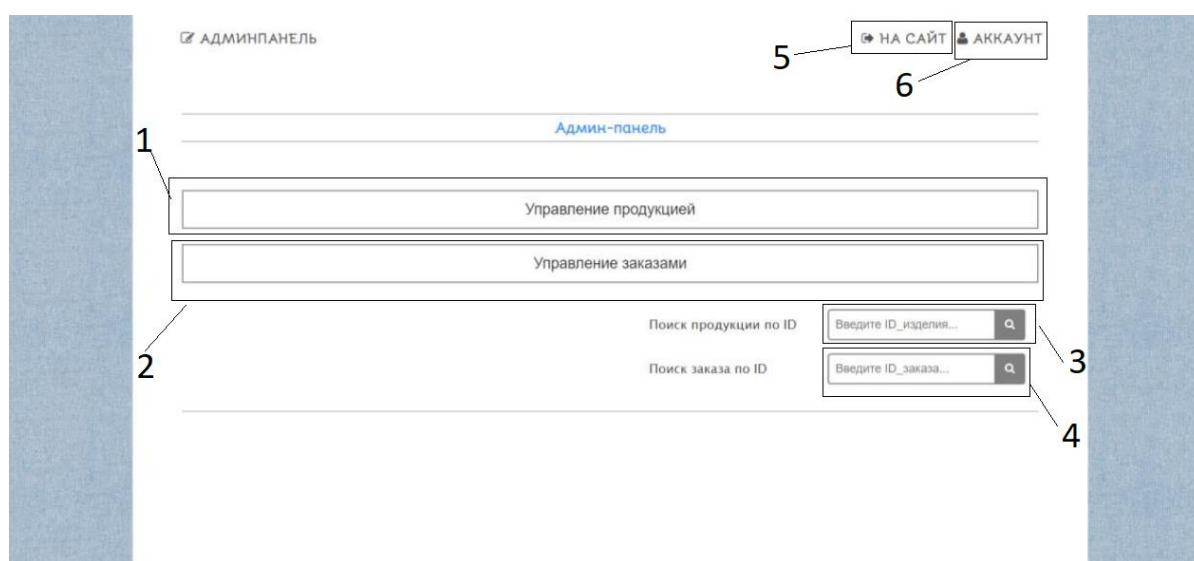


Рисунок 28 – Главная страница панели администратора

На главной странице представлены следующие элементы:

1 – кнопка для перехода на страницу управления продукцией;

2 – кнопка для перехода на страницу управления заказами;

3 – поиск товара по ID;

4 – поиска заказа по ID;

5 – кнопка перехода на сайт интернет-магазина;

6 – кнопка перехода к админпанели.

При переходе к управлению продукцией пользователю предоставляется страница с функционалом для работы с категориями и товаром. Интерфейс страницы представлен на рисунке 29.

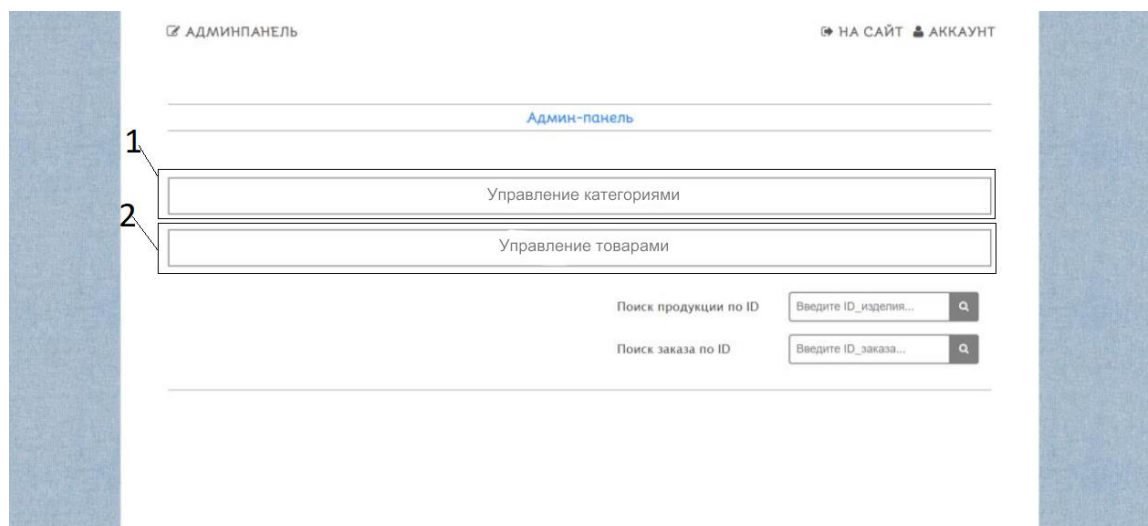


Рисунок 29 – Страница управления для перехода к управлению товаром и категориями

Данная страница состоит из:

1 – кнопка для перехода в раздел управления категориями;

2 – кнопка для перехода в раздел управления товарами;

Для работы с категориями необходимо перейти к управлению категориями. Интерфейс данной страницы представлен на рисунке 30.

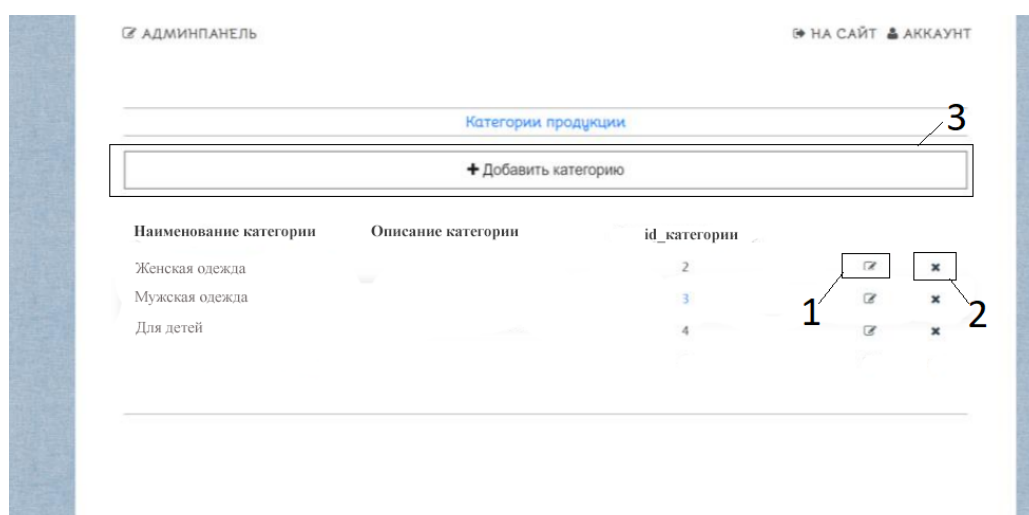


Рисунок 30 – Страница управления категориями

Интерфейс страницы состоит из следующих основных элементов:

1 – кнопка редактирования категории;

2 – кнопка удаления категории;

3 – кнопка добавления категории.

Интерфейс страницы добавления новой категории представлен на рисунке 31.

Рисунок 31 – Страница добавления категорий

Страница управления категориями включает следующие элементы:

1 – поле ввода наименования категории;

2 – поле ввода описания категории;

3 – кнопка для сохранения введенной информации;

4 – кнопка для возвращения на предыдущую страницу.

Страницы управления товарами и заказами имеют схожий интерфейс и элементы управления.

4 Реализация программного обеспечения для интернет-магазина

4.1 Построение диаграммы компонентов для интернет-магазина

Реализация программного обеспечения для интернет-магазина осуществлялась с помощью фреймворка Bootstrap, включающего в себя полный набор необходимых инструментов для разработки адаптивных дизайнов с использованием языка PHP [12].

Разбиение программы на структурные компоненты с обеспечением их взаимосвязи изображено на рисунке 32.

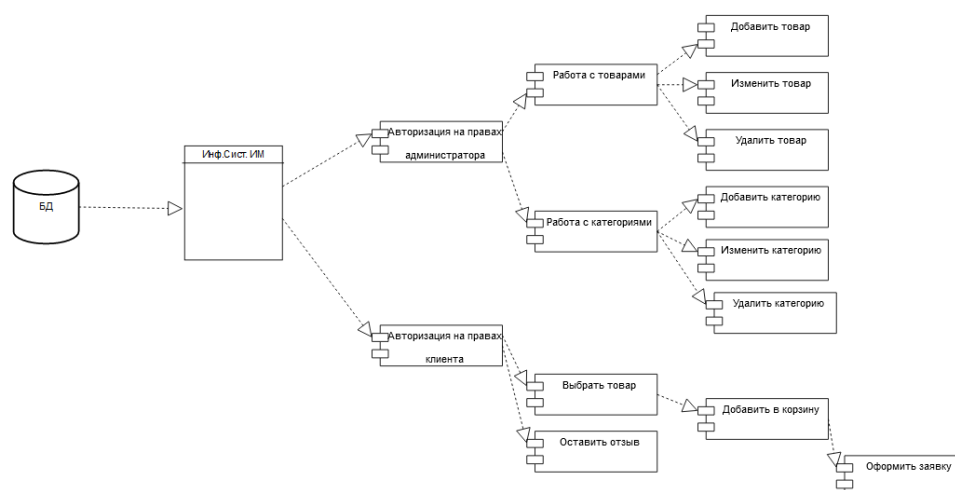


Рисунок 32 – Диаграмма компонентов программного обеспечения для интернет-магазина

Из диаграммы видно, что в состав компонентов информационной системы интернет-магазина входят база данных, а также набор веб-страниц, реализующий систему и ее функции. Набор веб-страниц зависит от прав пользователя, которые присваиваются ему при авторизации.

4.2 Реализация диаграммы развертывания интернет-магазина

Для реализации программного обеспечения интернет-магазина необходимо обратить внимание на архитектуру развертывания. Это поможет

визуализировать статические аспекты физических узлов и их взаимосвязей. Диаграмма развертывания представлена на рисунке 33.

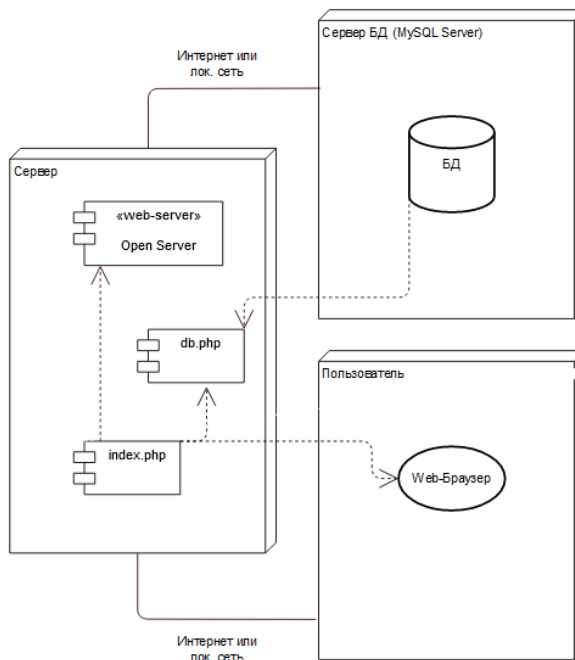


Рисунок 33 – Диаграмма развертывания интернет-магазина

Информационная система интернет-магазина расположена на сервере «Open Server», который обеспечивает связь с пользователем и сервером базы данных, на котором хранится сама база данных. Связь обеспечивается по локальной сети или интернету [17].

5 Контроль качества программного обеспечения интернет-магазина

5.1 Тестирование белого ящика

Пользователю предоставляется возможность для создания личного аккаунта на сайте интернет-магазина. Главная задача данного вида тестирования состоит в том, чтобы детально изучить логику процесса и проверить максимальное число возможных операторов, а также ветвлений [8].

Для наглядности и удобства тестирования была составлена блок-схема, на которой представлен алгоритм авторизации пользователя на сайте интернет-магазина с регистрацией, если та не была произведена ранее. Схема представлена на рисунке 34.

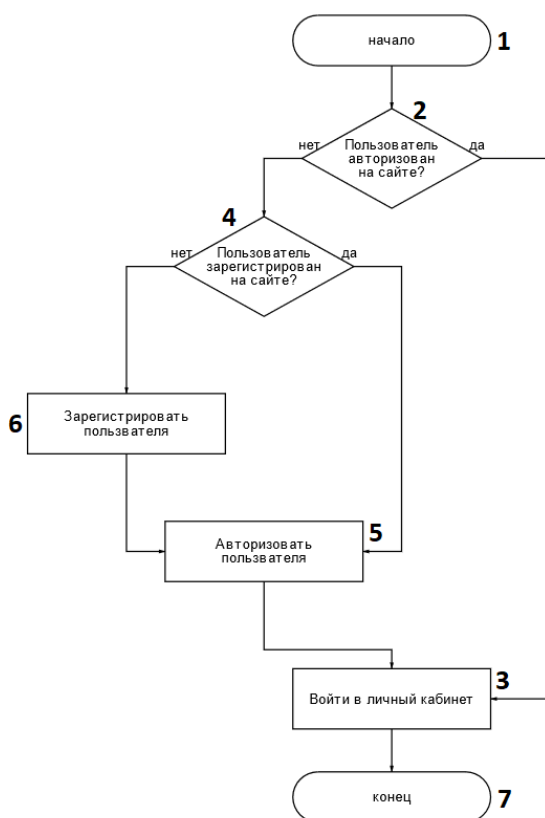


Рисунок 34 – Блок-схема «Авторизация пользователя»

Блок-схема иллюстрирует проверку авторизации пользователя со всевозможными исходами. Для наглядности функционирования системы

управления и движения потоков информации необходимо построить управляющий граф, представленный на рисунке 35.

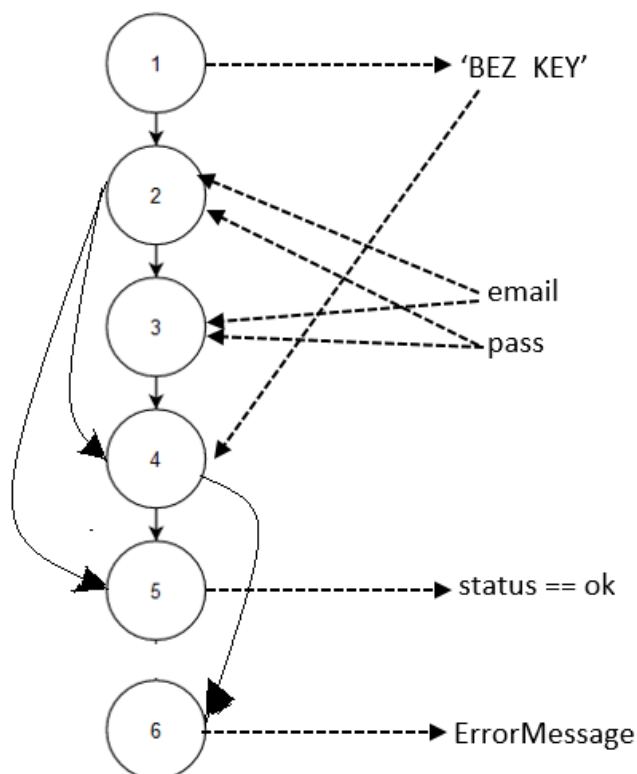


Рисунок 35 – Управляющий граф с потоками данных

Из графа видно, что при регистрации на вход поступают данные об e-mail и пароле. Если информация достоверна, то система регистрирует положительный статус входа, в противном же случае выводит сообщение об ошибке.

5.1.1 Метод покрытия операторов

Данный критерий основывается на выполнении каждого оператора программы. Проход должен осуществляться не менее 1 раза.

Для выполнения каждого оператора было составлено два теста. Результаты приведены в таблице 2.

Таблица 2 – Результат тестирования

№	Операторы	Тест	Фактический результат	Результат тестирования
1	6, 5, 7	Пользователь не был зарегистрирован на сайте. Регистрация не прошла. Пользователь не авторизован на сайте. Вход в личный кабинет не был выполнен.	Вход в личный кабинет не выполнен	Успешно
2	3, 7	Пользователь был зарегистрирован. Вход в личный кабинет был выполнен.	Вход в личный кабинет выполнен	Успешно

В ходе тестирования были задействованы (покрыты) все операторы. Полученные результаты показали корректную работу алгоритма.

5.1.2 Метод покрытия условий

Необходимо составить достаточное число тестов, чтобы проверить все возможные исходы событий каждого условия и условия выполнения каждого оператора, как минимум, один раз. Результаты приведены в таблице 3.

Таблица 3 – Результат тестирования

№	Условия	Тест	Фактический результат	Результат тестирования
1	2 – да	Пользователь зарегистрирован.	Вход в личный кабинет	Успешно
2	2 – нет, 4 – да, 5 – да	Пользователь не авторизован, но зарегистрирован. Получен ответ от сервера.	Успешный вход в личный кабинет	Успешно
3	2 – нет, 4 – нет	Пользователь не авторизован и не зарегистрирован.	Переход к регистрации	Успешно

Продолжение таблицы 3

№	Условия	Тест	Фактический результат	Результат тестирования
4	2 – нет, 4 – да, 5 – нет	Пользователь не авторизован и не зарегистрирован. Ответ от сервера не получен	Сообщение об ошибке	Успешно

В ходе тестирования были задействованы (покрыты) все условия. Полученные результаты показали корректную работу алгоритма.

5.1.3 Метод покрытия маршрутов

Для выполнения всех маршрутов было составлено 4 теста. Результаты приведены в таблице 4.

Таблица 4 – Результат тестирования

№	Маршрут	Тест	Фактический результат	Результат тестирования
1	1, 2, 3, 7	Пользователь зарегистрирован и авторизован	Успешный вход	Успешно.
2	1, 2, 4, 5, 3, 7	Пользователь не авторизован. Пользователь зарегистрирован. Успешно проходит авторизацию	Успешный вход	Успешно.
3	1, 2, 4, 6, 5, 3, 7	Пользователь не авторизован. Пользователь не зарегистрирован. Успешно проходит регистрацию. Успешно проходит авторизацию.	Успешный вход	Успешно.
4	1, 2, 4, 6, 3, 7	Пользователь не авторизован. Пользователь не зарегистрирован. Регистрация не пройдена	Сообщение об ошибке. Вход не выполнен	Успешно.

В ходе тестирования были задействованы (покрыты) все маршруты. Полученные результаты показали корректную работу алгоритма.

5.2 Тестирование черного ящика

Для регистрации на сайте интернет-магазина необходимо придумать уникальный логин, незарегистрированный в системе ранее, а также надежный пароль. Логин должен содержать от 4 до 20 символов, среди которых допускаются только буквы латинского алфавита, цифры, точка нижнее подчеркивание. Пароль может содержать количество символов от 8 до 50 любых знаков. Регистрироваться под одним логином нельзя.

Реакция программы на пользовательский ввод:

- система уведомляет пользователя о неверной длине логина и пароля, если они не входят в заданный диапазон;
- система уведомляет пользователя об использовании некорректных символов в составе логина при их наличии;
- система уведомляет пользователя, если заданный для регистрации логин уже используется другим пользователем;
- система уведомляет пользователя об успешной регистрации, если все данные введены верно.

5.2.1 Классы эквивалентности

На рисунке 36 представлены классы эквивалентности для длины логина n_l .



Рисунок 36 – Классы эквивалентности для длины логина

Пароль корректный, если $n_l \in [4, 20]$.

На рисунке 37 представлены классы эквивалентности для длины пароля n_p .



Рисунок 37 – Классы эквивалентности для длины пароля

Пароль корректный, если $n_p \in [8, 50]$.

На рисунке 38 обозначены классы эквивалентности для символов, которые входят в состав логина.



Рисунок 38 – Классы эквивалентности для символов логина

Логин корректный, если состоит из символов латинского алфавита, цифр или символа нижнего подчеркивания, точки.

На рисунке 39 обозначены классы эквивалентности для символов, которые входят в состав пароля.

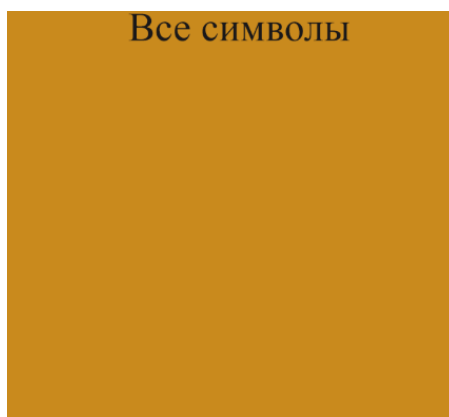


Рисунок 39 – Классы эквивалентности для символов пароля

Пароль является корректным, если он состоит из любых символов.

На рисунке 40 представлены классы эквивалентности по признаку занятости логина.



Рисунок 40 – Классы эквивалентности по занятости логина

Пользователь может зарегистрироваться только под свободным логином.

Составим таблицу 5 на основе рассмотренных классов эквивалентности.

Таблица 5 – Классы эквивалентности

Входные условия	Правильные классы эквивалентности	Неправильные классы эквивалентности
Длина логина	4-20	< 4, > 20
Длина пароля	8-20	< 8, > 50
Логин	Состоит из букв латинского алфавита, цифр, нижнего подчеркивания, точки	Содержит другие символы
Пароль	Содержит любые символы	-
Занятость логина	Логин свободен	Логин занят

Для наглядности составим таблицу 6 для тестирования классов эквивалентности.

Таблица 6 – Тестирование классов эквивалентности

№	Входные параметры	Фактический результат	Результат тестирования
1	Логин: kirillackimov13! Пароль: l1k242ghfhru	Ошибка регистрации (некорректный логин)	Успешно (рисунок В.1)
2	Логин: krr11 Пароль: 12345qwer	Ошибка регистрации (данный логин уже занят)	Успешно (рисунок В.2)
3	Логин: кирилл123 Пароль: 123fergkeo5	Ошибка регистрации (некорректный логин)	Успешно (рисунок В.3)
4	Логин: kirill56 Пароль: 8p9s3uhhiewv	Регистрация прошла успешно	Успешно (рисунок В.4)

В таблице наглядно представлены результаты тестирования с указанием всех ошибок. Результаты тестов, приведенные в таблице представлены на рисунках В.1-В.4.

5.2.2 Функциональная диаграмма

На рисунке 41 представлена функциональная диаграмма.

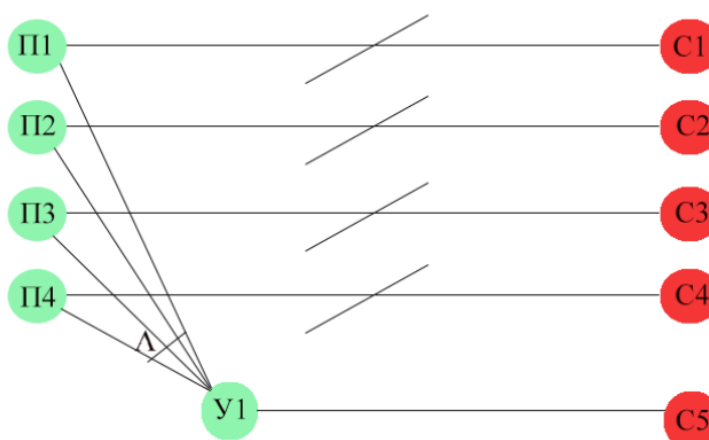


Рисунок 41 – Функциональная диаграмма

В таблицах 7 и 8 представлено описание причин и следствий соответственно.

Таблица 7 – Описание причин

Причина	Описание
П1	Корректная длина логина
П2	Корректная длина пароля
П3	Логин состоит из корректных символов
П4	Логин свободен

Таблица 8 – Описание следствий

Следствие	Описание
C1	Сообщение: «Некорректная длина логина»
C2	Сообщение: «Некорректная длина пароля»
C3	Сообщение: «Некорректные символы в составе логина»
C4	Сообщение: «Логин занят»
C5	Сообщение: «Регистрация прошла успешно»

В таблице 9 представлена таблица решений, составленная на основе функциональной диаграммы.

Таблица 9 – Таблица решений

[illegible]

В таблице 10 представлен набор тестов, составленный на основе таблицы решений.

Таблица 10 – Набор тестов на основе таблицы решений

№	Входные данные	Ожидаемый результат
1	Логин: ba\$ Пароль: psu	(C1, C2, C3, C4)
2	Логин: Abraham1\$ Пароль: 123	(C2, C3, C4)
3	Логин: Or) Пароль: qwerty1234	(C1, C3, C4)
4	Логин: G1! Пароль: 11	(C1, C2, C4)
5	Логин: megaman20italyforestcmontuiop;fkv^^ Пароль: morethen20s!mbols!np@sswordbddfbbcvcbvbnhmbjmbj,v,b	(C1, C2, C3)
6	Логин: Bjorn^^^^ Пароль: ^)(= /hffqqqwe	(C3, C4)
7	Логин: Br4qw Пароль: ftgf5?,thfghfghfhjkljlhgjxfhr546%hgh5herbfb54gg	(C2, C4)
8	Логин: Ivarrtu\$ Пароль: SumpasswordBest	(C3, C4)
9	Логин: WhoHowFlowhphphphphphphphphph Пароль: 12345	(C1, C4)
10	Логин: Ul^ Пароль: hjorfkv405ic	(C1, C3)
11	Логин: larisavasilievaviktorovnatozhehochetzachet Пароль: 12r	(C1, C2)
12	Логин: kirillzachet (логин занят) Пароль: 11rr22tt33yy	(C4)
13	Логин: vovovovovoivashkapilotflyrevnuyrtyujk123123 Пароль: 123456789qwertyuiop1a	(C1)
14	Логин: Johnny1292 Пароль: incorrectpasswordsorryreloadvbbgfghfgkhprsdsvsjhmgdngnhghndh	(C2)
15	Логин: !@#\$%^&*K Пароль: stopitnomore546	(C3)
16	Логин: K1Ryaallg Пароль: goodpass4123	(C5)

В данной таблице прописан ожидаемый результат в виде ранее вышеописанных следствий.

5.3 Метрики кода

Метрики кода позволяют получить численное значение некоторого свойства программного обеспечения или его спецификаций. В основном метрики кода игнорируют сформированный код при вычислении значений метрик. Это позволяет значениям метрик отражать, что может видеть и изменять разработчик. Для оценки проекта был использован «RNPLoc», автоматический генерирующий статистику кода. На рисунках 42-43 представлены результаты тестов фрагмента кода, отвечающего за регистрацию и авторизацию.

Size	
Lines of Code (LOC)	100
Comment Lines of Code (CLOC)	9 (9.00%)
Non-Comment Lines of Code (NCLOC)	91 (91.00%)
Logical Lines of Code (LLOC)	33 (33.00%)
Classes	33 (100.00%)
Average Class Length	33
Minimum Class Length	33
Maximum Class Length	33
Average Method Length	4
Minimum Method Length	1
Maximum Method Length	14
Average Methods Per Class	7
Minimum Methods Per Class	7
Maximum Methods Per Class	7
Functions	0 (0.00%)
Average Function Length	0
Not in classes or functions	0 (0.00%)
Cyclomatic Complexity	
Average Complexity per LLOC	0.15
Average Complexity per Class	6.00
Minimum Class Complexity	6.00
Maximum Class Complexity	6.00
Average Complexity per Method	1.71
Minimum Method Complexity	1.00
Maximum Method Complexity	2.00
Dependencies	
Global Accesses	0
Global Constants	0 (0.00%)
Global Variables	0 (0.00%)
Super-Global Variables	0 (0.00%)
Attribute Accesses	0
Non-Static	0 (0.00%)
Static	0 (0.00%)
Method Calls	15
Non-Static	13 (86.67%)
Static	2 (13.33%)

Рисунок 42 – Результаты тестов «RNPLoc» (часть 1)

Structure	
Namespaces	0
Interfaces	0
Traits	0
Classes	1
Abstract Classes	0 (0.00%)
Concrete Classes	1 (100.00%)
Final Classes	0 (0.00%)
Non-Final Classes	1 (100.00%)
Methods	7
Scope	
Non-Static Methods	0 (0.00%)
Static Methods	7 (100.00%)
Visibility	
Public Methods	7 (100.00%)
Protected Methods	0 (0.00%)
Private Methods	0 (0.00%)
Functions	0
Named Functions	0 (0.00%)
Anonymous Functions	0 (0.00%)
Constants	0
Global Constants	0 (0.00%)
Class Constants	0 (0.00%)
Public Constants	0 (0.00%)
Non-Public Constants	0 (0.00%)

Рисунок 43 – Результаты тестов «RNPLOC» (часть 2)

В ходе расчета количественных метрик для рассматриваемого фрагмента кода было выявлено, что количество строк составляет 100, среди которых 9 комментариев.

Наиболее важным фактором проверки является цикломатическая сложность, от которой напрямую зависит сложность поддержки кода. Цикломатическая сложность метода не должна превышать 10 [3]. Максимальный показатель Cyclomatic Complexity проверенного фрагмента составляет 6.00. А значит проект нельзя назвать сложным, что соответствует нормированным показателям [15].

6 Структурное и календарное планирование для проекта программного обеспечения для интернет-магазина

Планирование структуры интернет-магазина – это фундаментальный этап разработки, продвижения и внутренних бизнес-процессов. Для того, чтобы структура и основа сайта были оптимальными необходимо определить все этапы, разбив их на определенный ряд задач.

Начав реализацию сайта без структурного анализа, рано или поздно, потребуются дополнительные затраты времени и финансов для изменений. Структурное и календарное планирование позволят максимально снизить риски, определить необходимый бюджет и назначить сроки выполнения проекта.

6.1 Определение этапов и задач проектирования программного обеспечения для интернет-магазина

Затраты на оплату труда включают в себя затраты на основную заработную плату, дополнительную заработную плату и страховые взносы, то есть начисления на заработную плату, в зависимости от категории плательщика, указанных в ФЗ №22-ФЗ [1].

Основная заработная плата рассчитывается на основе данных о трудоемкости и месячном окладе по штатному расписанию.

Трудоемкость характеризуется перечнем основных этапов и видов работ, которые должны быть выполнены в проекте. Чтобы рассчитать трудоемкость работы по выполнению проекта реализации интернет-магазина, необходимо определить этапы проектирования программного обеспечения и составить перечень задач [14].

В ходе структурного планирования было определено 38 задач, необходимых для выполнения. Также произведем расчет финансовых вложений, разделив их на трудовые, материальные ресурсы и дополнительные затраты. Расчет представлен в таблице 11.

Таблица 11 – Расчет общих затрат на необходимые ресурсы

Ресурс	Тип ресурса	Стандартная ставка	Сверхурочные
Руководитель проекта	Трудовой	250,00 Р/ч	320,00 Р/ч
Технический специалист по программной части	Трудовой	350,00 Р/ч	420,00 Р/ч
Технический специалист по дизайну	Трудовой	250,00 Р/ч	320,00 Р/ч
Тестировщик	Трудовой	150,00 Р/ч	220,00 Р/ч
Ресурс	Тип ресурса	Стандартная ставка	Сверхурочные
Маркетолог	Трудовой	120,00 Р/ч	190,00 Р/ч
Аренда помещения	Затраты		
Интернет	Затраты		
Ноутбук	Материальный	60 000,00 Р	

Каждый из ресурсов необходимо равномерно распределить по задачам проекта в соответствии с теми или иными потребностями реализации.

Для реализации и полного запуска программного обеспечения задействованы 3 технических специалиста по программной части, 1 технический специалист по дизайну для создания функциональной и визуальной части проекта. Также в проекте принимают участие руководитель, контролирующий весь процесс в целом и маркетолог, отвечающий за запуск интернет-магазина.

При создании проекта разработчики используют имеющееся компьютерное и периферийное оборудование, которые являются основными средствами предприятия. В расчет общих затрат на создание проекта

включены затраты на амортизацию основных средств в виде 3-х единиц ноутбука, которые относятся к необходимым материальным затратам.

Затраты на услуги, предоставляемые исполнителям работы, могут быть приняты по их прямой стоимости. При расчете себестоимости была определена актуальная ценовая политика для аренды помещения и подключения интернета [13].

6.2 Календарное планирование и оценка бюджета проекта программного обеспечения для интернет-магазина

Календарное планирование – это основа проектирования, которая позволяет определить последовательность, интенсивность и продолжительность работ в реализации проекта программного обеспечения информационной системы. Также это позволит определить все необходимые материальные, технические и трудовые финансовые затраты. Целью календарного планирования является точное планирование всех деталей для успешного завершения проекта [1].

В рассматриваемом проекте для полного определения сроков и бюджета были включены:

- планирование и выстраивание последовательности задач;
- определение сроков и длительности выполнения работ;
- построение диаграммы Ганта;
- построение сетевого графика;
- расчет затрат и вычисление общих сроков.

Все необходимые вычисления, а также построение графических схем будут проводиться в среде Microsoft Project 2013. Список задач, длительность выполнения работ представлена на рисунках 44 и 45.

	1	Название задачи	Длительность	Начало	Окончание	Предшественники	Названия ресурсов
1		Начало проекта	0 дней	Вт 27.04.21	Вт 27.04.21		Руководитель проекта
2		Анализ требований	17 дней	Пт 07.05.21	Пн 31.05.21		
3		Анализ программных средств конкурентов	4 дней	Пт 07.05.21	Ср 12.05.21	1	Руководитель проекта; Технический специалист по программной части
4		Анализ актуальной продукции рынка	6 дней	Чт 13.05.21	Чт 20.05.21	3	Руководитель проекта[50%]; Маркетолог
5		Поиск поставщиков	5 дней	Пт 21.05.21	Чт 27.05.21	4	Руководитель проекта; Маркетолог[50%]
6		Составление расчетной сметы	1 день	Пт 28.05.21	Пт 28.05.21	5	Руководитель проекта
7		Утверждение проекта	1 день	Пн 31.05.21	Пн 31.05.21	6	Руководитель проекта
8		Анализ требований завершен	0 дней	Пн 31.05.21	Пн 31.05.21	7	
9		Выбор технических инструментов для реализации проекта	6 дней	Вт 01.06.21	Вт 08.06.21		
10		Анализ всех доступных технических средств	4 дней	Вт 01.06.21	Пт 04.06.21	8	Технический специалист по программной части; Руководитель проекта[50%]; Ноутбук[3 Единиц]
11		Изучение методической документации	1 день	Пн 07.06.21	Пн 07.06.21	10	Технический специалист по программной части
12		Утверждение выбранных технических инструментов	1 день	Вт 08.06.21	Вт 08.06.21	11	Технический специалист по программной части[50%]; Руководитель проекта[50%]
13		Выбор технических инструментов для реализации проекта завершен	0 дней	Вт 08.06.21	Вт 08.06.21	12	
14		Составление проекта интернет-магазина	8 дней	Ср 09.06.21	Пт 18.06.21		
15		Анализ структур данных интернет-магазина	3 дней	Ср 09.06.21	Пн 14.06.21	13	Аренда помещения[70 000,00 ₽]; Интернет[2 000,00 ₽]; Технический специалист по программной части[200%]
16		Выбор архитектуры интернет-магазина	4 дней	Вт 15.06.21	Пт 18.06.21	15	Технический специалист по программной части
17		Проектирование дизайна интернет-магазина	4 дней	Вт 15.06.21	Пт 18.06.21	15	Технический специалист по дизайну
18		Проектирование модели управления интернет-магазином	2 дней	Вт 15.06.21	Ср 16.06.21	15	Руководитель проекта
19		Составление проекта интернет-магазина завершен	0 дней	Пт 18.06.21	Пт 18.06.21	18;16;17	
20		Реализация интернет-магазина	11 дней	Пн 21.06.21	Пн 05.07.21		
21		Реализация структур данных	3 дней	Пн 21.06.21	Ср 23.06.21	19	Технический специалист по программной части[200%]
22		Реализация серверной части	7 дней	Чт 24.06.21	Пт 02.07.21	21	Технический специалист по программной части[200%]

Рисунок 44 – Задачи и длительность проекта (часть 1)

23		Реализация клиентской части	7 дней	Пн 21.06.21	Ср 30.06.21	19	Технический специалист по программной части[50%]; Технический специалист по дизайну
24		Наполнение контентом	1 день	Пн 05.07.21	Пн 05.07.21	21;22;23	Маркетолог; Технический специалист по программной части[50%]; Технический специалист по дизайну
25		Реализация интернет-магазина завершена	0 дней	Пн 05.07.21	Пн 05.07.21	24	
26		Тестирование разработанного интернет-магазина	10 дней	Вт 06.07.21	Пн 19.07.21		
27		Тестирование запросов к серверу	2 дней	Вт 06.07.21	Ср 07.07.21	25	Технический специалист по программной части; Тестировщик
28		Тестирование функциональной части	3 дней	Вт 06.07.21	Чт 08.07.21	25	Тестировщик
29		Тестирование клиентской части	3 дней	Пт 09.07.21	Вт 13.07.21	28	Технический специалист по дизайну[50%]; Тестировщик
30		Исправление возникших ошибок	4 дней	Ср 14.07.21	Пн 19.07.21	27;29;28	Технический специалист по программной части; Технический специалист по дизайну
31		Тестирование разработанного интернет-магазина завершен	0 дней	Пн 19.07.21	Пн 19.07.21	30	
32		Запуск интернет-магазина	13 дней	Вт 20.07.21	Чт 05.08.21		
33		Анализ рынка рекламных площадок	3 дней	Вт 20.07.21	Чт 22.07.21	31	Маркетолог
34		Настройка рекламы	2 дней	Пт 23.07.21	Пн 26.07.21	33	Маркетолог
35		Запуск сайта	1 день	Вт 27.07.21	Вт 27.07.21	34	Технический специалист по программной части; Руководитель проекта[50%]
36		Анализ работы и посещаемости с устранением недостатков	7 дней	Ср 28.07.21	Чт 05.08.21	35	Руководитель проекта[50%]; Технический специалист по программной части[50%]; Маркетолог[50%]; Технический специалист по дизайну[50%]
37		Выпуск приложения завершен	0 дней	Чт 05.08.21	Чт 05.08.21	36	
38		Конец проекта	0 дней	Чт 05.08.21	Чт 05.08.21	37	

Рисунок 45 – Задачи и длительность проекта (часть 2)

Датой начала проекта назначено 27.04.21. Работа разделена на шесть основных этапов с отдельными задачами:

- анализ требований (17 дней);
- выбор технических инструментов для реализации проекта (6 дней);
- составление проекта интернет-магазина (8 дней);
- реализация интернет-магазина (11 дней);
- тестирование разработанного интернет-магазина (10 дней);

– запуск интернет-магазина (13 дней).

Также были распределены все ресурсы, необходимые на определенном этапе под каждую из задач. Для визуализации управления проектом была построена диаграмма Ганта, представленная на рисунках 46 и 47.

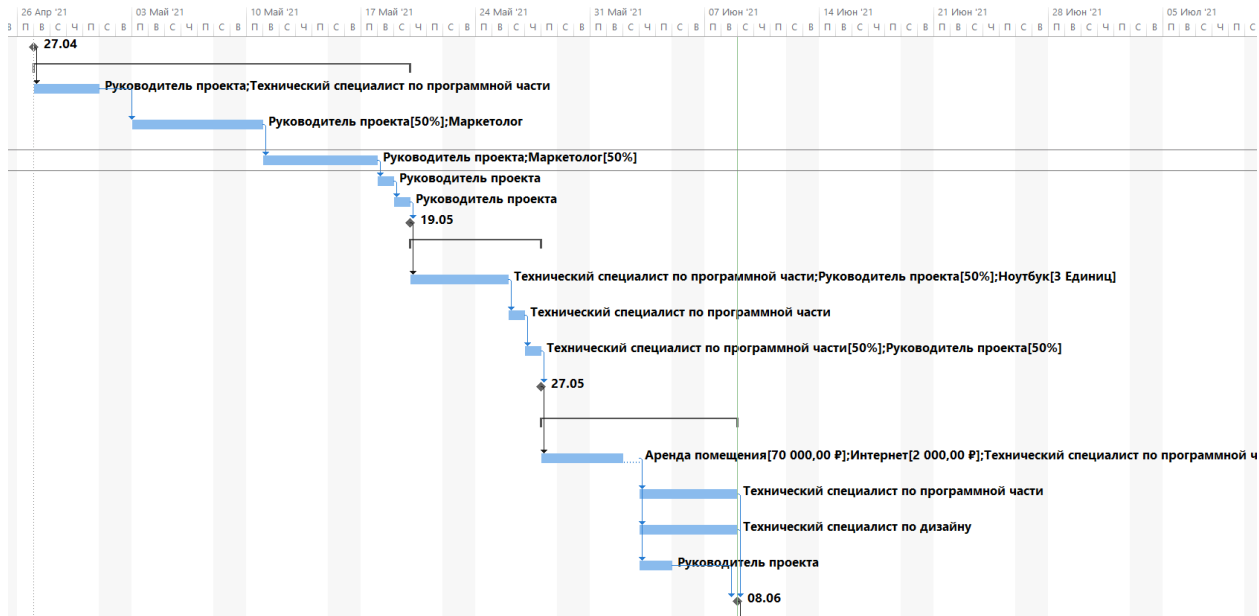


Рисунок 46 – Диаграмма Ганта проектирования программного обеспечения для интернет-магазина (часть 1)

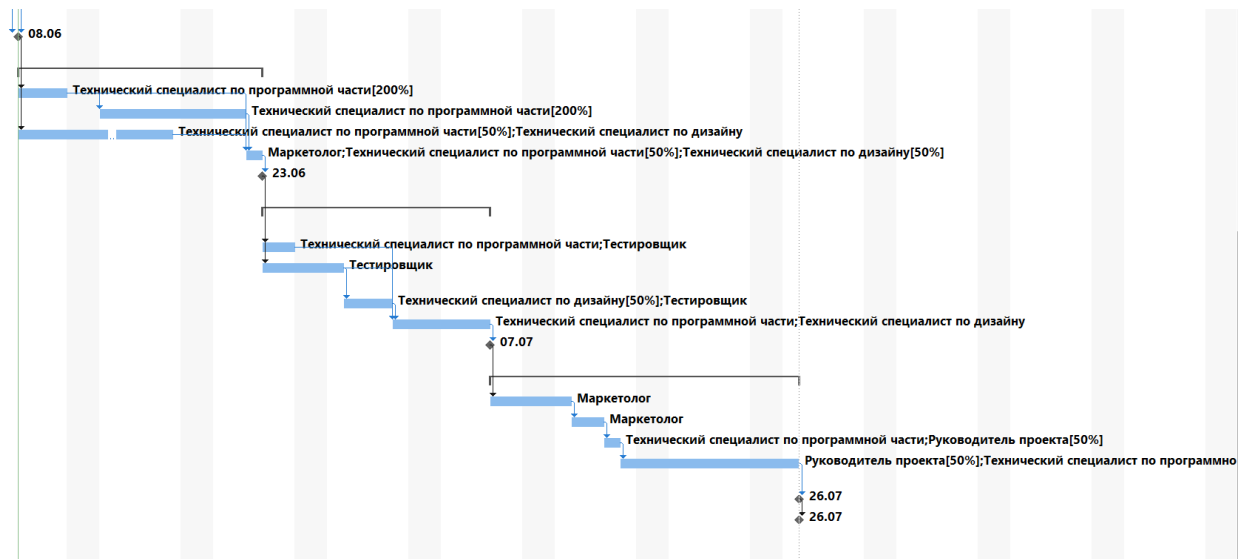


Рисунок 47 – Диаграмма Ганта проектирования программного обеспечения для интернет-магазина (часть 2)

На диаграмме можно увидеть поэтапное выполнение каждой задачи с указанными сроками, а также определить, какие работы будут выполняться последовательно, а какие параллельно. Например, выбор архитектуры интернет-магазина, проектирование дизайна интернет-магазина, проектирование модели управления интернет-магазином могут выполняться параллельно, так как задействованы разные ресурсы для каждой задачи.

Чтобы не нарушить сроки и выполнить создание программного обеспечения интернет-магазина к назначенной дате определим критический путь.

Для отображения технологической зависимости комплекса работ с выделением критических мест построим сетевой график, отобразив его на рисунках 48 – 51.



Рисунок 48 – Сетевой график проектирования программного обеспечения для интернет-магазина (часть 1)

```

graph LR
    A[Информационная поддержка  
начальника  
начало: 02.03.08.15  
конец: 02.03.08.15  
дл: 1.00ч  
ресурсы: Информационная поддержка начальника] --> B{Информационная поддержка  
начальника  
начало: 02.03.08.15  
конец: 02.03.08.15  
дл: 1.00ч  
ресурсы: Информационная поддержка начальника}
    B --> C[Информационная поддержка  
начальника  
начало: 02.03.08.15  
конец: 02.03.08.15  
дл: 1.00ч  
ресурсы: Информационная поддержка начальника]
    B --> D[Информационная поддержка  
начальника  
начало: 02.03.08.15  
конец: 02.03.08.15  
дл: 1.00ч  
ресурсы: Информационная поддержка начальника]
    C --> E[Информационная поддержка  
начальника  
начало: 02.03.08.15  
конец: 02.03.08.15  
дл: 1.00ч  
ресурсы: Информационная поддержка начальника]
    D --> F[Информационная поддержка  
начальника  
начало: 02.03.08.15  
конец: 02.03.08.15  
дл: 1.00ч  
ресурсы: Информационная поддержка начальника]
    E --> G{Информационная поддержка  
начальника  
начало: 02.03.08.15  
конец: 02.03.08.15  
дл: 1.00ч  
ресурсы: Информационная поддержка начальника}
    F --> G
    
```

```

graph LR
    A[Данные проекта] --> B[Настройка проекта]
    B --> C[Запуск сайта]
    C --> D[Данные работы и взаимодействия с участниками проекта]
    D --> E[Выпуск приложения пользователя]
    E --> F[Конец проекта]
  
```

Данные проекта

начало: 10.08.07.21, фин.: 31

Описание: 170.12.07.21, 20: 2 день

Результат: Планирование

Настройка проекта

начало: 30.12.07.21, фин.: 34

Описание: 120.04.07.21, 20: 2 день

Результат: Планирование

Запуск сайта

начало: 10.12.07.21, фин.: 35

Описание: 10.12.07.21, 20: 2 день

Результат: "Полученный список" на информационный сайт

Данные работы и взаимодействия с участниками проекта

начало: 10.12.07.21, фин.: 36

Описание: 10.12.07.21, 20: 2 день

Результат: Размещение на сайте (СЭП), Информационный сайт

Выпуск приложения пользователя

дата выво: 10.12.07.21

170: 37

Конец проекта

дата выво: 10.12.07.21

170: 38

Рисунок 51 – Сетевой график проектирования программного обеспечения для интернет-магазина (часть 4)

Из сетевого графика, составленного в программной среде Microsoft Project видно, то критические пути проходят через задачи 1, 8, 13, 19, 25, 31.

Теперь же составим общий обзор проекта, определив длительность, затраты на реализацию всей проектной деятельности. Информация отображена на рисунке 52.

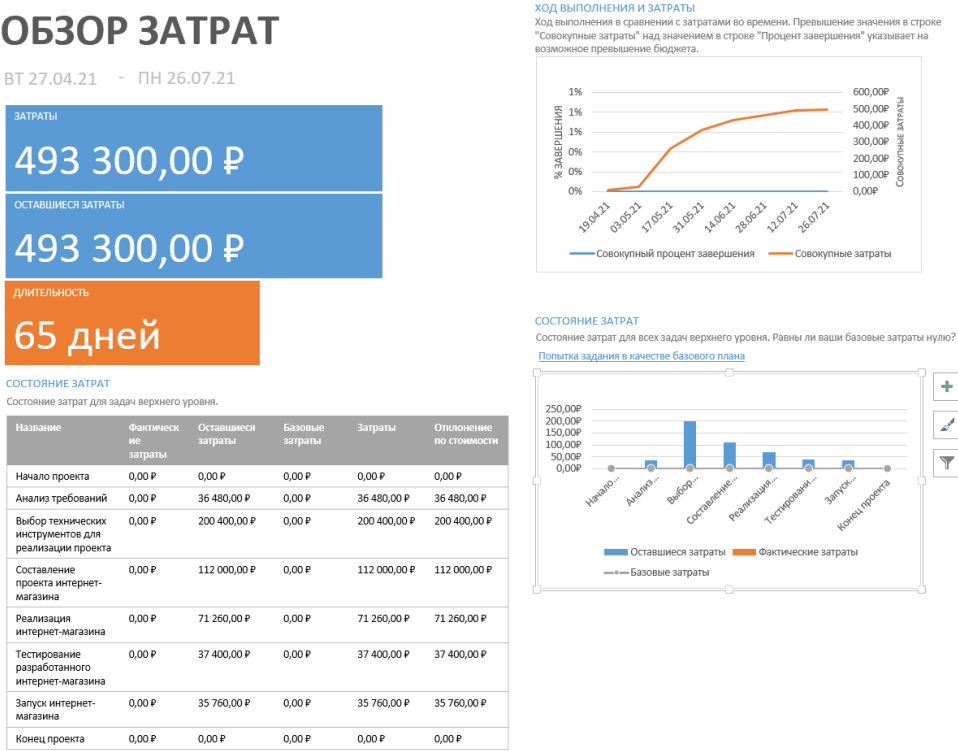


Таблица 12 – Общие затраты на создание программного продукта

Этапы работы	Сумма, руб.	Доля, %
Начало проекта	0	0
Анализ требований	36 480	7,4
Выбор технических инструментов для реализации проекта	200 400	40,62
Составление проекта интернет-магазина	112 000	22,7
Реализация интернет-магазина	71 260	14,45
Тестирование разработанного интернет-магазина	37 400	7,58
Запуск интернет-магазина	35 760	7,25
Конец проекта	0	0
Итого затрат на создание проекта	493 300	100

Таким образом, можно сделать вывод, что общие затраты на создание программного продукта составят 493 300 рублей.

Заключение

В процессе подготовки выпускной квалификационной работы было разработано программное обеспечение интернет-магазина вещей K-SHOP. Система обеспечивает возможность для быстрого заказа продукции. Пользовательский интерфейс качественный и соответствует современным тенденциям, обеспечивая дружелюбность и практичность.

Разработка осуществлялась на PHP с помощью дополнительного фреймворка Bootstrap. Серверная часть реализована в среде MySQLServer при помощи Open Server.

В ходе рассмотрения всех необходимых аспектов в реализации информационной системы интернет-магазина были составлены диаграммы, а также проведено тестирование, которое показало корректность работы разработанного программного обеспечения.

Также в среде Microsoft Office Project 2013 были произведены оценка и планирование бюджета с диаграммой Ганта, показывающей календарный график выполнения всех работ. Таким образом, можно утверждать, что разработанное программное обеспечение удовлетворяет установленным требованиям.

Список использованных источников

1. Аналитика – все о маркетинге интернет-магазина [Электронный ресурс]. – URL: <https://www.shopolog.ru/metodichka/analytics/onlayn-shopping-v-cifrah-glavnaya-statistika/> (дата обращения: 23.05.2021).
2. Буньон, Лоран Silverlight 2. Технология создания насыщенных интернет-приложений / Лоран Буньон. - М.: ДМК Пресс, 2016. - 826 с.
3. Винарский, Я. С. Web-аппликации в интернет-маркетинге. Проектирование, создание и применение / Я.С. Винарский, Р.Д. Гутгарц. - М.: ИНФРА-М, 2015. - 270 с.
4. Ларри, Ульман PHP и MySQL. Создание интернет-магазинов / Ульман Ларри. - М.: Диалектика / Вильямс, 2015. - 525 с.
5. CMS Рейтинг - аналитика и статистика за 2020 год [Электронный ресурс]. – URL: <https://cms-rating.ru/chto-i-kak-pokupayut-v-internete/> (дата обращения: 25.05.2021).
6. Кириллов, В.В. Введение в реляционные базы данных - СПб.: БХВ-Петербург, 2019. - 464 с.;
7. Люк Веллинг, Лора Томсон. Разработка веб-приложений с помощью PHP и MySQL. – Вильямс, 2019;
8. Пирогов, В.Ю. Информационные системы и базы данных: организация и проектирование: Учебное пособие / В.Ю. Пирогов. - СПб.: БХВПетербург, 2019. - 528 с.;
9. Кристиан Дари, Эмилиан Баланеску. PHP и MySQL: создание интернет-магазина: – М.: «Вильямс», 2018. – 640 с.;
10. Справочник языка PHP: [Электронный ресурс]. – URL: <https://php.ru/manual/> (дата обращения: 15.04.2021);
11. Справочник по HTML: [Электронный ресурс]. – URL: <http://htmlbook.ru/html> (дата обращения: 07.05.2021);

12. Кузнецов Максим, Симдянов Игорь. РНР. Практика создания Web-сайтов. – 2-е изд. перераб. и доп. – СПб.: «БХВ-Петербург», 2017. – С. 1264;
13. Агальцов, В.П. Базы данных. В 2-х т. Т. 2. Распределенные и удаленные базы данных: Учебник / В.П. Агальцов. - М.: ИД ФОРУМ, НИЦ ИНФРА-М, 2018. - 272 с.;
14. Редько, В.Н.; Бассараб, И.А. Базы данных и информационные системы; Знание, 2019. - 602 с.;
15. Джеймс Р. Грофф, Пол Н. Вайнберг, Эндрю Дж. Оппель. SQL: полное руководство, 3-е издание. — М.: «Вильямс», 2017. — 960 с.;
16. Локальная сеть //Информационное общество: [Электронный ресурс]. – URL: <https://pro-spo.ru/network-tech/3011-2012-03-06-11-42-08> (дата обращения: 07.05.2021);
17. Библиотека Java-Script [Электронный ресурс]. – Режим доступа: <http://www.03www.ru/>(дата обращения: 20.05.2021).
18. TML справочник–дескрипторы и их атрибуты [Электронныйресурс] – URL: <http://www.on-line-teaching.com/html/lsn031.html> (дата обращения: 23.05.2021).
19. Internet-Technologies Сайтостроение от А до Я [Электронный ресурс]. – URL: <http://www.internet-technologies.ru>(дата обращения: 25.05.2021).
20. Web Server Survey [Электронный ресурс]. – URL: <http://news.netcraft.com/archives/2010/12/01/december-2010-web-server-survey.html> (дата обращения: 16.06.2021).
21. Библиотека ресурсов Интернет индустрии [Электронный ресурс]. – URL: <http://www.i2r.ru>(дата обращения: 17.05.2021).
22. Аксенов К. А., Клебанов Б. И. Работа с CASE-средствами VPwin и ERwin: учебное пособие./ К. А. Аксенов, Б. И.Клебанов – Екатеринбург 2004 - 50с.

23. Евдокимов Н. В. Основы контентной оптимизации. / Н. В. Евдокимов. - изд. ООО И. Д. Вильяме, Москва, 2007 год, 160 с.
24. Орлов Л. В. Как создать Интернет-магазин. / Л. В. Орлов- изд. Бук-Пресс, Москва, 2006 год, 384 с.
25. Кирсанов Д., Веб-дизайн./ Д. Кирсанов.- СПб,:Символ-плюс, 1999. - 376 с.
26. Ломов, А.Ю. HTML, CSS, скрипты: практика создания сайтов / А.Ю. Ломов. - СПб. : БХВ-Петербург, 2007. - 416с.
27. Рейнолдс М.Сделай сам интернет-магазин / М. Рейнолдс. – Лорри, 2008. – 536 с.

Приложение А
ГЛОССАРИЙ
(справочное)

База данных (БД) – это совокупность массивов и файлов данных, организованная по определённым правилам, предусматривающим стандартные принципы описания, хранения и обработки данных независимо от их вида.

Система управления базами данных (СУБД) – совокупность языковых и программных средств, предназначенных для создания, ведения и совместного использования БД многими пользователями.

Unified Modeling Language (UML) – это универсальный развивающийся язык моделирования в области программной инженерии, которая предназначена для предоставления стандартного способа визуализации работы системы.

Приложение Б
ЛИСТИНГ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ
(рекомендуемое)

Листинг Б.1 – файл Order.php

```
<?php

class Order
{
    const SHOW_BY_DEFAULT = 10;

    public static function save($userId, $products) {

        $db = Db::getConnection();

        // $date_order='';
        $status='0';

        // echo "$name <br> $email <br> $password - $pass <br> rights -
        $rights <br> $address";

        $sql = "INSERT INTO orders (products, ID_user, status) VALUES
        (:products, :ID_user, :status)";

        $result = $db->prepare($sql);
        $result->bindParam(':products', $products, PDO::PARAM_STR);
        // $result->bindParam(':date_order', $date_order, PDO::PARAM_STR);
        $result->bindParam(':ID_user', $userId, PDO::PARAM_STR);
        $result->bindParam(':status', $status, PDO::PARAM_STR);

        $result->execute();
        $id_order = $db->lastInsertId();

        return $id_order;
    }

    public static function getAllOrders()
    {

        // Соединение с БД
        $db = Db::getConnection();

        // Получение и возврат результатов
        $result = $db->query('SELECT * FROM orders');
        $AllOrders = array();
        $i = 0;
        while ($row = $result->fetch()) {
            $AllOrders[$i]['ID_order'] = $row['ID_order'];
            $AllOrders[$i]['date_order'] = $row['date_order'];
            $AllOrders[$i]['ID_user'] = $row['ID_user'];
            $AllOrders[$i]['status'] = $row['status'];
            $i++;
        }
        return $AllOrders;
    }

    /* Возвращает список заказов со страничной навигацией */
    public static function getOrdersList($page)
    {

        $limit = Order::SHOW_BY_DEFAULT;
        // Смещение (для запроса)
        $offset = ($page - 1) * self::SHOW_BY_DEFAULT;
    }
}
```

```

// Соединение с БД
$db = Db::getConnection();

// Получение и возврат результатов
$result = $db->query('SELECT ID_order, date_order, ID_user, status
FROM orders ORDER BY (date_order) DESC LIMIT
'.self::SHOW_BY_DEFAULT. ' OFFSET '.$offset);
$orderList = array();
$i = 0;
while ($row = $result->fetch()) {
    $orderList[$i]['ID_order'] = $row['ID_order'];
    $orderList[$i]['date_order'] = $row['date_order'];
    $orderList[$i]['ID_user'] = $row['ID_user'];
    $orderList[$i]['status'] = $row['status'];
    $i++;
}
return $orderList;
}

/*
 * Возвращает текстовое пояснение статуса для заказа :
 */
public static function getStatusText($status)
{
    switch ($status) {
        case '0':
            return 'Новый заказ';
            break;
        case '1':
            return 'Одобрено администратором';
            break;
        case '2':
            return 'Изготавливается';
            break;
        case '3':
            return 'Закрыт';
            break;
        case '4':
            return 'Отказано в исполнении';
            break;
        case '5':
            return 'Отменено пользователем';
            break;
    }
}

// количество заказов конкретного пользователя
public static function getOrderCountUser($id)
{
    $db = Db::getConnection();

    $result = $db->query('SELECT * FROM orders
WHERE ID_user = '.$id.' ORDER BY date_order DESC');

    $orderCountUser = array();
    $i = 0;
    while ($row = $result->fetch()) {
        $orderCountUser[$i]['ID_order'] = $row['ID_order'];
        $orderCountUser[$i]['date_order'] = $row['date_order'];
        $orderCountUser[$i]['products'] = $row['products'];
        $orderCountUser[$i]['ID_user'] = $row['ID_user'];
    }
}

```

```

        $ordersCountUser[$i]['status'] = $row['status'];
        $i++;
    }
    return $ordersCountUser;
}

/* Возвращает количество всех заказов */
public static function getOrdersCount()
{
    // Соединение с БД
    $db = Db::getConnection();

    // Получение и возврат результатов
    $result = $db->query('SELECT * FROM orders');
    $ordersList = array();
    $i = 0;
    while ($row = $result->fetch()) {
        $ordersList[$i]['ID_order'] = $row['ID_order'];
        $ordersList[$i]['date_order'] = $row['date_order'];
        $ordersList[$i]['ID_user'] = $row['ID_user'];
        $ordersList[$i]['status'] = $row['status'];
        $i++;
    }
    $total = count($ordersList);

    return $total;
}

// итоговая стоимость товаров в заказе
public static function getTotalPrice($products, $productsQuantity)
{
    // Подсчитываем общую стоимость
    $total = 0;
    foreach ($products as $item) {
        // Находим общую стоимость: цена товара * количество товара
        $total += $item['price'] *
$productsQuantity[$item['ID_product']];
    }
    return $total;
}

// итоговое количество товаров в заказе
public static function getTotalCount($productsQuantity)
{
    // Подсчитываем общую стоимость
    $total = 0;
    foreach ($productsQuantity as $item) {
        // Находим общую стоимость: цена товара * количество товара
        $total = $total + $item;
    }
    return $total;
}
}

```

Листинг Б.2 – файл Product.php

```
<?php
```

```
class Product
```

```

{

//Список категорий

//Список подкатегорий категории
    public static function getProductSubcategories($id_cat)
    {
        $id_cat = intval($id_cat);

        if ($id_cat) {
            $db = Db::getConnection();
            $ProductSubcategories = array();
            $result = $db->query('SELECT * FROM type_product WHERE
ID_category=' . $id_cat);

            $i = 0;
            while($row = $result->fetch()) {
                $ProductSubcategories[$i]['ID_type'] = $row['ID_type'];
                $ProductSubcategories[$i]['ID_category'] =
$row['ID_category'];
                $ProductSubcategories[$i]['Naimenovanie_type'] =
$row['Naimenovanie_type'];
                // $ProductSubcategories[$i]['image'] = $row['image'];
                $i++;
            }

            return $ProductSubcategories;

        }

    }

//Список подкатегорий
    public static function getSubcategories()
    {

        $db = Db::getConnection();
        $Subcategories = array();
        $result = $db->query('SELECT * FROM type_product');

        $i = 0;
        while($row = $result->fetch()) {
            $Subcategories[$i]['ID_type'] = $row['ID_type'];
            $Subcategories[$i]['ID_category'] = $row['ID_category'];
            $Subcategories[$i]['Naimenovanie_type'] =
$row['Naimenovanie_type'];
            $i++;
        }

        return $Subcategories;

    }

//Вывод всех товаров подкатегорий
    public static function getProductView($id_cat, $id_type)
    {
        $id_cat = intval($id_cat);
        $id_type = intval($id_type);
    }

```

```

        if ($id_cat && $id_type) {
            $db = Db::getConnection();
            $ProductView = array();
            $result = $db->query('SELECT * FROM products WHERE ID_type=' .
$id_type);

            $i = 0;
            while($row = $result->fetch()) {
                $ProductView[$i]['ID_product'] = $row['ID_product'];
                $ProductView[$i]['ID_type'] = $row['ID_type'];
                $ProductView[$i]['model'] = $row['model'];
                $ProductView[$i]['dlina'] = $row['dlina'];
                $ProductView[$i]['vysota'] = $row['vysota'];
                $ProductView[$i]['shirina'] = $row['shirina'];
                $ProductView[$i]['product_volume'] = $row['product_volume'];
                $ProductView[$i]['ves'] = $row['ves'];
                $ProductView[$i]['price'] = $row['price'];
                $i++;
            }
            return $ProductView;
        }
    }

//Вывод конкретного товара
    public static function getProductByID($id_cat, $id_type, $id)
    {
        $id_cat = intval($id_cat);
        $id_type = intval($id_type);
        $id = intval($id);

        if ($id_cat && $id_type && $id) {
            $db = Db::getConnection();
            $result = $db->query('SELECT * FROM products WHERE
ID_product=' . $id);
            $result->setFetchMode(PDO::FETCH_ASSOC);
            $ProductItem = $result->fetch();
            return $ProductItem;
        }
    }

// вывод категорий и подкатегорий
    public static function getProduct()
    {
        $result_2 = $db->query('SELECT * FROM type_product WHERE
ID_category=' . $Product[$i]['ID_category']);
        $r=0;
        while($row_2 = $result_2->fetch()) {
            $sub[$r]['ID_type'] = $row_2['ID_type'];
            $sub[$r]['Naimenovanie_type'] =
$row_2['Naimenovanie_type'];
            $r++;
        };

        $Product[$i]['Naimenovanie_type']=$sub;
        $i++;
    }

    return $Product;
}

```

```

const SHOW_BY_DEFAULT = 10;

public static function getProductByIds($idsArray, $page)
{
    $products = array();
    $idsString = implode(',', $idsArray);

    $limit = Product::SHOW_BY_DEFAULT;
    $offset = ($page - 1) * self::SHOW_BY_DEFAULT;

    $db = Db::getConnection();

    $sql = "SELECT * FROM products WHERE ID_product IN ($idsString)
LIMIT ".$limit." OFFSET ".$offset;

    $result = $db->query($sql);
    $result->setFetchMode(PDO::FETCH_ASSOC);

    $i = 0;
    while ($row = $result->fetch()) {
        $products[$i]['ID_product'] = $row['ID_product'];
        $products[$i]['ID_type'] = $row['ID_type'];
        $products[$i]['model'] = $row['model'];
        $products[$i]['dlina'] = $row['dlina'];
        $products[$i]['vysota'] = $row['vysota'];
        $products[$i]['shirina'] = $row['shirina'];
        $products[$i]['product_volume'] = $row['product_volume'];
        $products[$i]['ves'] = $row['ves'];
        $products[$i]['price'] = $row['price'];
        $i++;
    }

    return $products;
}

public static function getProductsByIdsCheck($idsArray)
{
    $products = array();
    $idsString = implode(',', $idsArray);

    $db = Db::getConnection();

    $sql = "SELECT * FROM products WHERE ID_product IN ($idsString)";

    $result = $db->query($sql);
    $result->setFetchMode(PDO::FETCH_ASSOC);

    $i = 0;
    while ($row = $result->fetch()) {
        $products[$i]['ID_product'] = $row['ID_product'];
        $products[$i]['ID_type'] = $row['ID_type'];
        $products[$i]['model'] = $row['model'];
        $products[$i]['dlina'] = $row['dlina'];
        $products[$i]['vysota'] = $row['vysota'];
        $products[$i]['shirina'] = $row['shirina'];
        $products[$i]['product_volume'] = $row['product_volume'];
        $products[$i]['ves'] = $row['ves'];
        $products[$i]['price'] = $row['price'];
    }
}

```

```

        $i++;
    }

    return $products;
}

//получение категорий по идентификатору

public static function getCategoryById($id)
{
    $id = intval($id);

    if ($id) {
        $db = Db::getConnection();
        $result = $db->query('SELECT * FROM category WHERE
ID_category=' . $id);
        $result->setFetchMode(PDO::FETCH_ASSOC);
        $ProductCategoryById = $result->fetch();
        return $ProductCategoryById;
    }
}

//получение подкатегорий по идентификатору

public static function getSubcategoryById($id_type)
{
    $id_type = intval($id_type);
    if ($id_type) {
        $db = Db::getConnection();
        $result = $db->query('SELECT * FROM type_product WHERE
ID_type=' . $id_type);
        $result->setFetchMode(PDO::FETCH_ASSOC);
        $ProductSubcategoryById = $result->fetch();
        return $ProductSubcategoryById;
    }
}

// product by ID
public static function searchProducts($id)
{
    $id = intval($id);
    if ($id) {
        $db = Db::getConnection();
        $result = $db->query('SELECT * FROM products WHERE
ID_product=' . $id);
        $result->setFetchMode(PDO::FETCH_ASSOC);
        $ProductCategoryById = $result->fetch();
        return $ProductCategoryById;
    }
}

// products добавить
public static function createProducts($options)
{
    $db = Db::getConnection();

    // Текст запроса к БД
    $sql = 'INSERT INTO products (ID_type, model, dlina, vysota, shirina,
product_volume, ves, price) VALUES (:ID_type, :model, :dlina, :vysota,
:shirina, :product_volume, :ves, :price)';

}

```

```

public static function updateProducts($id, $options)
{
    $db = Db::getConnection();

    // Текст запроса к БД
    $sql = "UPDATE products
    SET ID_type = :ID_type, model= :model, dlina= :dlina, vysota= :vysota,
    shirina = :shirina, product_volume = :product_volume, ves = :ves, price =
    :price
    WHERE ID_product = :ID_product";
}

```

Листинг Б.3 – файл AdminOrderController.php

```

<?php
class AdminOrderController extends AdminBase
{
    /**
     * Action для страницы "Управление заказами"
     */
    public function actionIndex($page = 1)
    {
        // echo "page = $page";
        // Проверка доступа
        self::checkAdmin();

        // Получаем список заказов
        $ordersList = Order::getOrdersList($page);

        // Общее количество товаров (необходимо для постраничной навигации)
        $total = Order::getOrdersCount();

        if (isset($_POST['search']) && $_POST['order_id']!=null) {

            $id = $_POST['order_id'];
            $page = 1;
            // Проверка доступа
            self::checkAdmin();

            $order = Order::getOrderById($id);

            if ($order) {

                $productsQuantity = json_decode($order['products'], true);

                $productsIds = array_keys($productsQuantity);

                $products = Product::getProductByIds($productsIds, $page);

                // итоговая стоимость заказа
                $products_price = Product::getProductsByIdsCheck($productsIds);
                $totalPrice = Order::getTotalPrice($products_price,
                $productsQuantity);

                // Общее количество товаров (необходимо для постраничной
                навигации)
            }
        }
    }
}

```

```

        $total = count($productsIds);
        // Создаем объект Pagination - постраничная навигация
        $pagination = new Pagination($total, $page,
Product::SHOW_BY_DEFAULT, 'page-');

    }
}

/**
 * Action для страницы "Редактирование заказа"
 */
public function actionUpdateOrders($id)
{
    // Проверка доступа
    self::checkAdmin();

    // Получаем данные о конкретном заказе
    $order = Order::getOrderById($id);
    $userById = User::getUserById($order['ID_user']);

    // Обработка формы
    if (isset($_POST['submit'])) {
        // Если форма отправлена
        // Получаем данные из формы

        $status = $_POST['status'];

        // Сохраняем изменения
        Order::updateOrderById($id, $status);
    }
}

/**
 * Action для страницы "Просмотр заказа"
 */
public function actionView($id, $page = 1)
{
    // Проверка доступа
    self::checkAdmin();

    // Получаем данные о конкретном заказе
    $order = Order::getOrderById($id);

    // Получаем массив с идентификаторами и количеством товаров
    $productsQuantity = json_decode($order['products'], true);

    // Получаем массив с идентификаторами товаров
    $productsIds = array_keys($productsQuantity);

    // количество товаров в заказе
    $totalCount = Order::getTotalCount($productsQuantity);

    // Получаем список товаров в заказе
    $products = Product::getProductByIds($productsIds, $page);

    // итоговая стоимость заказа
    $products_price = Product::getProductsByIdsCheck($productsIds);

```

```

        $totalPrice = Order::getTotalPrice($products_praise,
$productsQuantity);
    }

    public function actionDeleteOrdersYears()
    {
        // Проверка доступа
        self::checkAdmin();

        $ordersList = Order::getAllOrders();
        $count_order_delete = 0;
        $date1 = date('Y-m-d h:i:s');
        $first = new DateTime($date1);

        foreach ($ordersList as $order) {
            $second = new DateTime($order['date_order']);
            $diff = $first->diff($second);
            $week = $diff->format('%a') / 7 ;
            if ($week>=144) { $count_order_delete++;}
        }

        // Обработка формы
        if (isset($_POST['submit'])) {

            $date1 = date('Y-m-d h:i:s');
            $first = new DateTime($date1);

            foreach ($ordersList as $order) {

                $second = new DateTime($order['date_order']);
                $diff = $first->diff($second);
                $week = $diff->format('%a') / 7 ;

                if ($week>=144) {
                    $count_order_delete++;
                    Order::deleteOrdersYears($order['ID_order']);
                }

            }

        }

    }

    /**
     * Action для страницы "Просмотр заказа"
     */
    public function actionDownloadOrders($id)
    {
        $id = intval($id);
        if ($id) {
            $order = Order::getOrderById($id);
            $productsQuantity = json_decode($order['products'], true);
            $productsIds = array_keys($productsQuantity);
            $products = Product::getProductsByIdsCheck($productsIds);

            // header('Location: /views/admin_order/statement/statement.php');
        }
    }

```

```

        require_once(ROOT . '/views/admin_order/statement/statement.php');
        return true;
    }
}

```

Листинг Б.4 – файл AdminProductController.php

```

<?php
/*
 * Контроллер AdminProductController
 * Управление товарами в админпанели
 */
class AdminProductController extends AdminBase
{
    //Действия над категориями

    //редактировать категорию
    public function actionUpdateCategory($id)
    {
        // Проверка доступа
        self::checkAdmin();

        // Получаем данные о категории
        $category = Product::getCategoryById($id);
        $category_name = $category['Category_name'];
        $Sorting_number = $category['Sorting_number'];

        // Обработка формы
        if (isset($_POST['submit'])) {
            // Если форма отправлена
            // Получаем данные из формы редактирования. При необходимости
            можно валидировать значения
            $name = $_POST['Category_name'];
            $Sort = $_POST['Sorting_number'];
            // Сохраняем изменения
            Product::updateCategory($id, $name, $Sort);
        }

        //удалить категорию
        public function actionDeleteCategory($id)
        {
            self::checkAdmin(); // Проверка доступа

            // получение наименования категории
            $category = Product::getCategoryById($id);
            $category_name = $category['Category_name'];
        }

        //редактировать категорию
        public function actionUpdateSubcategory($id_type)
        {
            // Проверка доступа
            self::checkAdmin();

            $categoriesList = Product::getProductGenus(); //для выпадающего
            списка

            // Получаем данные о подкатегории
            $Subcategory = Product::getSubcategoryById($id_type);

```

```

//текущее наименование категории
$Subcategory_id_cat = $Subcategory['ID_category']; //id cat
$category_this = Product::getCategoryById($Subcategory_id_cat);
$category_name = $category_this['Category_name'];

// Сохраняем изменения
if (Product::updateSubcategory($id_type, $name, $category_id)) {
    // Проверим, загружалось ли через форму изображение
    if (is_uploaded_file($_FILES["image"]["tmp_name"])) {
        // Если загружалось, переместим его в нужную папку,
дадим новое имя
        move_uploaded_file($_FILES["image"]["tmp_name"],
$_SERVER['DOCUMENT_ROOT'] . "/template/img_product/products/{ $id_type }.jpg");
    }
};

// Перенаправляем пользователя на страницу управлениями товарами
header("Location: /admin/subcategory/$category_id");
}

// Подключаем вид
require_once(ROOT . '/views/admin_product/subcategory/update.php');
return true;
}

public function actionDeleteSubcategory($id)
{
    self::checkAdmin(); // Проверка доступа

    // Получаем данные о подкатегории
    $Subcategory = Product::getSubcategoryById($id);
    $subcategory_name = $Subcategory['Naimenovanie_type'];
    $category_id = $Subcategory['ID_category'];
}

// products

// посмотреть products

public function actionProducts($id_cat, $id_type)
{
    // Проверка доступа
    self::checkAdmin();

    // Получаем продукцию
    $productList = array();
    $productList = Product::getProductView($id_cat, $id_type);

    // Получаем список подкатегорий
    $ProductSubcategories =
Product::getProductSubcategories($id_cat);

    // Получаем категорию по ИД
    $category = Product::getCategoryById($id_cat);
    $category_name = $category['Category_name'];

    // Получание подкатегорий для списка
    $subcategory = Product::getSubcategoryById($id_type);
    $subcategory_name = $subcategory['Naimenovanie_type'];

```

```

        // Подключаем вид
        require_once(ROOT . '/views/admin_product/products/index.php');
        return true;
    }

    public function actionCreateProducts($id_cat, $id_type)
    {
        // Проверка доступа
        self::checkAdmin();

        $Subcategory = Product::getProductSubcategories($id_cat); //select +

        // Обработка формы
        if (isset($_POST['submit'])) {
            // Если форма отправлена, Получаем данные из формы

            $options['id_type'] = $_POST['subcategory'];
            $options['model'] = $_POST['model'];
            $options['dlina'] = $_POST['dlina'];
            $options['vysota'] = $_POST['vysota'];
            $options['shirina'] = $_POST['shirina'];
            $options['product_volume'] = $_POST['product_volume'];
            $options['ves'] = $_POST['ves'];
            $options['price'] = $_POST['price'];

            $errors = false; // Флаг ошибок в форме

            if ($errors == false) {
                // Если ошибок нет
                Product::createProducts($options);
                $id_subcat = $options['id_type'];
                // Перенаправляем пользователя на страницу
                header("Location: /admin/products/$id_cat/$id_subcat");
            }
        }

        // Подключаем вид
        require_once(ROOT . '/views/admin_product/products/create.php');
        return true;
    }

    //редактировать product
    public function actionUpdateProducts($id_cat, $id_type, $id)
    {
        // Проверка доступа
        self::checkAdmin();

        $Subcategory = Product::getProductSubcategories($id_cat); //select +
        $productById = Product::getProductById($id_cat, $id_type, $id); //
product ID

        // Обработка формы
        if (isset($_POST['submit'])) {
            // Если форма отправлена
            // Получаем данные из формы редактирования. При необходимости
можно валидировать значения
            $options['id_type'] = $_POST['subcategory'];
            $options['model'] = $_POST['model'];
            $options['dlina'] = $_POST['dlina'];
            $options['vysota'] = $_POST['vysota'];
            $options['shirina'] = $_POST['shirina'];

```

```

$options['product_volume'] = $_POST['product_volume'];
$options['ves'] = $_POST['ves'];
$options['price'] = $_POST['price'];

// Сохраняем изменения
Product::updateProducts($id, $options);

$id_subcat = $options['id_type'];
header("Location: /admin/products/$id_cat/$id_subcat");
}

// Подключаем вид
require_once(ROOT . '/views/admin_product/products/update.php');
return true;
}

// удалить изделие
public function actionDeleteProducts($id_cat, $id_type, $id)
{
    self::checkAdmin(); // Проверка доступа

    $productById = Product::getProductByID($id_cat, $id_type, $id); //
product ID

}

}

public static function actionUpdateByID($id)
{
    self::checkAdmin();
    $productById = Product::searchProducts($id);
    $id_type = $productById['ID_type'];
    $Subcategory = Product::getSubcategories();

    if (isset($_POST['submit'])) {
        $options['id_type'] = $_POST['subcategory'];
        $options['model'] = $_POST['model'];
        $options['dlina'] = $_POST['dlina'];
        $options['vysota'] = $_POST['vysota'];
        $options['shirina'] = $_POST['shirina'];
        $options['product_volume'] = $_POST['product_volume'];
        $options['ves'] = $_POST['ves'];
        $options['price'] = $_POST['price'];

        // Сохраняем изменения
        Product::updateProducts($id, $options);

        header("Location: /admin/$id");
    }

    // Подключаем вид
    require_once(ROOT . '/views/admin_product/products/updateID.php');
    return true;
}
}

```

Листинг Б.5 – файл CabinetController.php

```
<?php
include_once ROOT.'/models/Cabinet.php';
include_once ROOT.'/models/User.php';

class CabinetController
{

    public function actionIndex()
    {
        // Получаем идентификатор пользователя из сессии
        $userId = User::checkLogged();

        // Получаем информацию о пользователе из БД
        $user = User::getUserById($userId);

        require_once(ROOT . '/views/cabinet/index.php');

        return true;
    }

    /*Список заказов пользователя */
    public function actionHistory($id, $page = 1)
    {
        // Получаем идентификатор пользователя из сессии
        $userId = User::checkLogged();
        // Получаем информацию о пользователе из БД
        $user = User::getUserById($userId);
        // Получаем список заказов
        $ordersList = Order::getOrderListByIdUser($id, $page);

        // Общее количество товаров (необходимо для постраничной навигации)
        $countOrder = Order::getOrderCountUser($id);
        $total = count($countOrder);

        // Создаем объект Pagination - постраничная навигация
        $pagination = new Pagination($total, $page, Order::SHOW_BY_DEFAULT,
'page-');

        require_once(ROOT . '/views/cabinet/history.php');
        return true;
    }

    // отменить заказ
    public function actionOrderCancel($id, $page = 1)
    {
        $status = '5';
        Order::updateOrderById($id, $status);

        $order = Order::getOrderById($id);

        $productsQuantity = json_decode($order['products'], true);

        $productsIds = array_keys($productsQuantity);

        $products = Product::getProductByIds($productsIds, $page);

        // для навигации
```

```

        $products_pride = Product::getProductsByIdsCheck($productsIds);
        $totalPrice = Order::getTotalPrice($products_pride,
$productsQuantity);
        $total = count($productsIds);
        $pagination = new Pagination($total, $page, Product::SHOW_BY_DEFAULT,
'page-');
        // Подключаем вид
        require_once(ROOT . '/views/cabinet/viewOrder.php');
        return true;
    }
    // возобновить заказ
    public function actionOrderRestart($id, $page = 1)
    {
        $status = '0';
        Order::updateOrderById($id, $status);

        $order = Order::getOrderById($id);

        $productsQuantity = json_decode($order['products'], true);

        $productsIds = array_keys($productsQuantity);

        $products = Product::getProductByIds($productsIds, $page);

        $products_pride = Product::getProductsByIdsCheck($productsIds);
        $totalPrice = Order::getTotalPrice($products_pride,
$productsQuantity);
        $total = count($productsIds);
    }

    public function actionPayment($id_order)
    {
        $id = intval($id_order);
        if ($id) {
            $order = Order::getOrderById($id);
            $productsQuantity = json_decode($order['products'], true);
            $productsIds = array_keys($productsQuantity);
            $products = Product::getProductsByIdsCheck($productsIds);

            //
        }
        // header("Location: /views/pdf/pdf.php");

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

```

Листинг Б.6 – файл CartController.php

```

<?php

class CartController
{
    public function actionIndex($page = 1)
    {

```

```

$productsInCart = false;

// Получим данные из корзины
$productsInCart = Cart::getProducts();

if ($productsInCart) {
    // Получаем полную информацию о товарах для списка
    $productsIds = array_keys($productsInCart);
    $products = Product::getProductByIds($productsIds, $page);

    // Получаем общую стоимость товаров

    $products_prise = Product::getProductsByIdsCheck($productsIds);
    $totalPrice = Cart::getTotalPrice($products_prise);

    // Общее количество товаров (необходимо для постраничной навигации)
    $total = count($productsIds);
    // Создаем объект Pagination - постраничная навигация
    $pagination = new Pagination($total, $page, Product::SHOW_BY_DEFAULT,
'page-');
}

require_once(ROOT . '/views/cart/index.php');

return true;
}

public function actionMinCart($id)
{
    // кол-во конкретного товара
    $count_products_id = Cart::MinCart($id);
    // кол-во товаров в корзине
    $countItemsCart = Cart::countItems();

    // Получаем общую стоимость товаров
    $productsInCart = Cart::getProducts();
    $productsIds = array_keys($productsInCart);
    $products_prise = Product::getProductsByIdsCheck($productsIds);
    $totalPrice = Cart::getTotalPrice($products_prise);

    // массив элементов
    $arraymin = array($count_products_id, $countItemsCart, $totalPrice);

    $comma_separated = implode("|", $arraymin);
    echo $comma_separated;

    /*echo Cart::MinCart($id);*/
    return true;
}

public function actionPlusCart($id)
{
    $count_products_id = Cart::PlusCart($id);
    // кол-во товаров в корзине
    $countItemsCart = Cart::countItems();

    // Получаем общую стоимость товаров
    $productsInCart = Cart::getProducts();

```

```

        $productsIds = array_keys($productsInCart);
        $products_prise = Product::getProductsByIdsCheck($productsIds);
        $totalPrice = Cart::getTotalPrice($products_prise);

        // массив элементов
        $arraymin = array($count_products_id, $countItemsCart, $totalPrice);

        $comma_separated = implode("|", $arraymin);
        /**/
        echo $comma_separated;

        return true;
    }

    // удалить 1 товар из корзины
    public function actionDelCart($id)
    {
        echo Cart::DelCart($id);
        return true;
    }

    //удалить все товары из корзины
    public function actionDeleteSession()
    {
        // session_destroy();
        Cart::DeleteSessionProsucts();
        return true;
    }

    public function actionCheckout()
    {
        // Статус успешного оформления заказа
        $result = false;

        // Форма отправлена?
        if (isset($_POST['submit'])) {
            // Форма отправлена? - Да
            // Считываем данные формы
            /*$userName = $_POST['userName'];
            $userPhone = $_POST['userPhone'];
            $userAddress = $_POST['userAddress'];*/

            $userId = User::checkLogged();
            $user = User::getUserById($userId);
            $userName = $user['name'];
            $userPhone = $user['phone'];
            $userAddress = $user['address'];

            // Валидация полей

            $errors = false;
            if ($userName == 'не указан' || $userPhone == 'не указан' ||
                $userAddress == 'не указан') {
                $errors[] = 'Не все данные указаны';
            }

            if ($errors == false) {
                // условие верно? - Да

```

```

// Сохраняем заказ в базе данных, Собираем информацию о заказе

$productsInCart = Cart::getProducts(); //in session save

if (User::isGuest()) {
    $userId = false;
} else {
    $userId = User::checkLogged();
}
// Сохраняем заказ в БД
$products = json_encode($productsInCart);

if ($products != 'false') {
    $result = Order::save($userId, $products);
} else {
    $result = false;
    header("Location: /catalog/");
}

if ($result) {
    // !!!!!Оповещаем администратора о новом заказе!!!!
    $adminEmail = 'lun.dola@ya.ru';
    $message = 'http://zavod.website';
    $subject = 'Новый заказ! Номер заказа - '.$result.'.';
    mail($adminEmail, $subject, $message);

    // Очищаем корзину
    Cart::DeleteSessionProducts();
}
} else {
    // Форма заполнена корректно? - Нет
    // Итоги: общая стоимость, количество товаров
    $productsInCart = Cart::getProducts();
    $productsIds = array_keys($productsInCart);
    $products = Product::getProductsByIdsCheck($productsIds);
    $totalPrice = Cart::getTotalPrice($products);
    $totalQuantity = Cart::countItems();
}

// Форма отправлена? - Нет
// Получим данные из корзины
$productsInCart = Cart::getProducts();

// В корзине есть товары?
if ($productsInCart == false) {
    // В корзине есть товары? - Нет
    // Отправляем пользователя на главную искать товары
    header("Location: /catalog/");
} else {
    // В корзине есть товары? - Да
    // Итоги: общая стоимость, количество товаров
    $productsIds = array_keys($productsInCart);
    $products = Product::getProductsByIdsCheck($productsIds);
    $totalPrice = Cart::getTotalPrice($products);
    $totalQuantity = Cart::countItems();

    $userName = false;
    $userPhone = false;
    $userComment = false;

    // Пользователь авторизирован?

```

```

        if (User::isGuest()) {
            // Нет
            // Значения для формы пустые
        } else {
            // Да, авторизирован
            // Получаем информацию о пользователе из БД по id
            $userId = User::checkLogged();
            $user = User::getUserById($userId);
            // Подставляем в форму
            $userName = $user['name'];
            $userPhone = $user['phone'];
            $userAddress = $user['address'];
        }
    }
}

require_once(ROOT . '/views/cart/checkout.php');
return true;
}
}

```

Листинг Б.7 – файл Index.html

```

<!DOCTYPE html>
<html lang="zxx">
<head>
    <title>k-shop | eCommerce Template</title>
    <meta charset="UTF-8">
    <meta name="description" content=" k-shop | eCommerce Template">
    <meta name="keywords" content="k-shop, eCommerce, creative, html">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <!-- Favicon -->
    <link href="img/favicon.ico" rel="shortcut icon"/>

    <!-- Stylesheets -->
    <link rel="stylesheet" href="css/bootstrap.min.css"/>
    <link rel="stylesheet" href="css/font-awesome.min.css"/>
    <link rel="stylesheet" href="css/flaticon.css"/>
    <link rel="stylesheet" href="css/slicknav.min.css"/>
    <link rel="stylesheet" href="css/jquery-ui.min.css"/>
    <link rel="stylesheet" href="css/owl.carousel.min.css"/>
    <link rel="stylesheet" href="css/animate.css"/>
    <link rel="stylesheet" href="css/style.css"/>

</head>
<body>
    <!-- Header section -->
    <header class="header-section">
        <div class="header-top">
            <div class="container">
                <div class="row">
                    <div class="col-lg-2 text-center text-lg-left">
                        <!-- logo -->
                        <a href="./index.html" class="site-logo">
                            
                        </a>
                    </div>
                    <div class="col-xl-6 col-lg-5">
                        <form class="header-search-form">

```

```

placeholder="Поиск товара">
search"></i></button>
                                </form>
</div>
<div class="col-xl-4 col-lg-5">
    <div class="user-panel">
        <div class="up-item">
            <i class="flaticon-
profile"></i>
                                <a
href="#">Зарегистрируйтесь</a>
                                </div>
        <div class="up-item">
            <i class="flaticon-
profile"></i>
                                <a href="#">Войти</a>
                                </div>
        <div class="up-item">
            <div class="shopping-card">
                <i class="flaticon-
bag"></i>
                                <span>0</span>
                                </div>
                                <a href="#">Корзина</a>
                                </div>
        </div>
    </div>
</div>
</div>
</div>
</nav>
</header>
<!-- Header section end -->

<!-- Hero section -->
<section class="hero-section">
    <div class="hero-slider owl-carousel">
        <div class="hs-item set-bg" data-setbg="img/bg.jpg">
            <div class="container">
                <div class="row">
                    <div class="col-xl-6 col-lg-7 text-white">
                        <h2>Джинсовые куртки</h2>
                        <p>Большой выбор мужских и женских
джинсовых курток в интернет-магазине K-SHOP. Бесплатная доставка и постоянные
скидки! </p>
                                <a href="#" class="site-btn sb-
line">Посмотреть</a>
                                <a href="#" class="site-btn sb-
white">Добавить в корзину</a>
                                </div>
                            </div>
                        </div>
                    </div>
                <div class="hs-item set-bg" data-setbg="img/bg-2.jpg">
                    <div class="container">
                        <div class="row">
                            <div class="col-xl-6 col-lg-7 text-white">
                                <h2>Джинсовые куртки</h2>

```

```

        <p>Большой выбор мужских и женских
джинсовых курток в интернет-магазине K-SHOP. Бесплатная доставка и постоянные
скидки! </p>
        <a href="#" class="site-btn sb-
line">Посмотреть</a>
        <a href="#" class="site-btn sb-
white">Добавить в корзину</a>
    </div>
</div>
</div>
</div>
</div>
<div class="container">
    <div class="slide-num-holder" id="snh-1"></div>
</div>
</section>
<!-- Hero section end -->

<!-- Features section -->
<section class="features-section">
    <div class="container-fluid">
        <div class="row">
            <div class="col-md-4 p-0 feature">
                <div class="feature-inner">
                    <div class="feature-icon">
                        
                    </div>
                    <h2>Быстрые и безопасные платежи</h2>
                </div>
            </div>
            <div class="col-md-4 p-0 feature">
                <div class="feature-inner">
                    <div class="feature-icon">
                        
                    </div>
                    <h2>Высококачественная продукция</h2>
                </div>
            </div>
            <div class="col-md-4 p-0 feature">
                <div class="feature-inner">
                    <div class="feature-icon">
                        
                    </div>
                    <h2>Быстрая и бесплатная доставка</h2>
                </div>
            </div>
        </div>
    </div>
</section>

<!-- Product filter section -->
<section class="product-filter-section">
    <div class="container">
        <div class="section-title">
            <h2>Обзор самых продаваемых продуктов</h2>
        </div>
        <button class="site-btn sb-line sb-dark">Загрузить
еще</button>
    </div>
</div>
</section>

```

```
<!-- Product filter section end -->

<!--===== Javascripts & JQuery =====>
<script src="js/jquery-3.2.1.min.js"></script>
<script src="js/bootstrap.min.js"></script>
<script src="js/jquery.slicknav.min.js"></script>
<script src="js/owl.carousel.min.js"></script>
<script src="js/jquery.nicescroll.min.js"></script>
<script src="js/jquery.zoom.min.js"></script>
<script src="js/jquery-ui.min.js"></script>
<script src="js/main.js"></script>

</body>
</html>
```

Приложение В
РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ
(рекомендуемое)

Регистрация

1. **Логин** *

Введен некорректный логин

2. **E-mail** *

3. **Пароль** *

Зарегистрироваться

Рисунок В.1 – Результат теста №1

Регистрация

1. **Логин** *

Данный логин уже занят

2. **E-mail** *

3. **Пароль** *

Зарегистрироваться

Рисунок В.2 – Результат теста №2

Регистрация

1. **Логин** *

кирилл123

Введен некорректный логин

2. **E-mail** *

ffk@mail.ru

3. **Пароль** *

123fergkeo5

Зарегистрироваться

Рисунок В.3 – Результат теста №3

Регистрация

1. **Логин** *

kirill56

2. **E-mail** *

ffk@mail.ru

3. **Пароль** *

8p9s3uhhiewv

Зарегистрироваться



Регистрация прошла
успешно

Рисунок В.4 – Результат теста №4