

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»

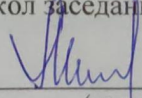
БАКАЛАВРСКАЯ РАБОТА

на тему

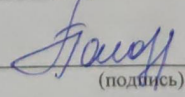
**Информационная система управления
продажами и доставкой в интернет-магазине
автозапчастей**

Студент	 14.06.21 (подпись, дата)	Легашнев Максим Александрович (ФИО полностью)
Руководитель	 14.06.21 (подпись, дата)	Балашова И. Ю. (фамилия, инициалы)
Нормоконтролёр	 22.06.21 (подпись, дата)	Такташкин Д.В. (фамилия, инициалы)

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

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

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

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

Пенза, 2021

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 Легашнев Максим Александрович гр. 17ВП1 факультета ВТ
<Фамилия, имя, отчество полностью>

2. Тема работы Информационная система управления продажами и доставкой в интернет-магазине автозапчастей

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

2. Руководитель работы к.т.н., доцент кафедры «МО и ПЭВМ» Балашова И.Ю.

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

1) Назначение: информационная система предназначена для управления

продажами и доставкой в интернет-магазине автозапчастей

2) Среда выполнения: любой интернет-браузер актуальной версии

Средства разработки: языки программирования PHP, SQL

Инструментальные средства: VS Code, MySQL Workbench 8.0 CE, Open Server

Технологии разработки: ООП

3) Основные функции информационной системы управления продажами и
доставкой в интернет-магазине автозапчастей:

а) просмотр каталога товаров;

б) подбор товара по критериям;

г) подбор товара по названию;

д) оформления заказа и доставки товара;

е) просмотр статуса заказа в личном кабинете;

ж) работа с ассортиментом администратором.

4. Перечень подлежащих разработке вопросов

Анализ предметной области

Анализ требований

Проектирование

Реализация

Контроль качества ПО:

– Тестирование

– Анализ качества кода

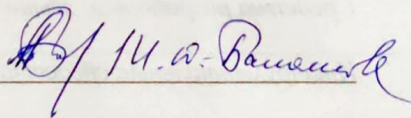
План разработки и оценка бюджета

5. Календарный график выполнения работы

№ п/п	Наименование этапов работы	Объем работы	Срок выполнения	Под руков
1	Анализ предметной области и требований	10	14.04.2021 – 24.04.2021	
2	Проектирование	16	25.04.2021 – 09.05.2021	
3	Реализация	17	10.05.2021 – 25.05.2021	
4	Контроль качества	9	26.05.2021 – 03.06.2021	
5	План разработки бюджета	2	04.06.2021 – 06.06.2021	
6	Оформление пояснительной записки	7	07.06.2021 – 14.06.2021	

Дата выдачи задания " н " сентябрь 20 21

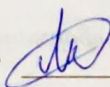
Руководитель бакалаврской работы



(подпись, дата)

Задание к исполнению принял студент

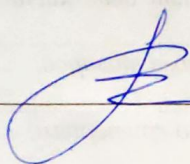
(подпись, дата)



Работу к защите допустить

Декан факультета

(подпись, дата)



Реферат

Пояснительная записка содержит 100 листов, 54 рисунка, 7 использованных источников.

ИНФОРМАЦИОННАЯ СИСТЕМА, АВТОЗАПЧАСТИ, UML, БАЗА ДАННЫХ, КЛИЕНТ, СЕРВЕР, MYSQL, OPEN SERVER

Объект разработки – информационная система управления продажами и доставкой в интернет-магазине автозапчастей.

Цель работы – разработать информационную систему, позволяющую осуществлять подбор товаров в магазине автозапчастей по различным критериям с возможностью заказа доставки.

Технология разработки – для разработки серверной части использовалась СУБД MySQL Workbench и язык SQL, клиентская часть была реализована на языке PHP в редакторе исходного кода VS Code с использованием фреймворка Laravel.

Результаты работы – в ходе выполнения выпускной квалификационной работы были созданы клиентская и серверная часть для информационной системы управления продажами и доставкой в интернет-магазине автозапчастей. Все требования к разработке приложения соблюдены.

					ПГУ 09.03.04–08ВР171.12 ПЗ						
Изм.	Лист	№ докум.	Подпись	Дата	Информационная система управления продажами и доставкой в интернет- магазине автозапчастей Пояснительная записка	Лит.		Лист	Листов		
Разраб.		Легашнев М. А.		14.06.24				3	100		
Провер.		Балашова И. Ю.		14.06.24							
И. Контр.		Такташкин Д. В.		24.06.24		Группа 17ВР1					
Утверд.		Макарычев П. П.									

Содержание

Введение.....	7
1 Анализ предметной области информационных систем интернет-магазинов.....	9
1.1 Анализ предметной области	9
1.2 Анализ аналогичных информационных систем интернет-магазинов автозапчастей.	10
1.3 Анализ требований к информационной системе интернет-магазина.....	20
1.4 Нефункциональные требования к информационной системе интернет-магазина	20
1.5 Функциональные требования к информационной системе интернет-магазина	21
1.6 Функциональное моделирование информационной системы интернет-магазина	24
1.7 Средства разработки информационной системы интернет-магазина.....	27
2 Проектирование информационной системы интернет-магазина.....	28
2.1 Базовая архитектура информационной системы интернет-магазина.....	28
2.2 Логическая модель базы данных информационной системы интернет-магазина	29
2.3 Физическая модель базы данных информационной системы интернет-магазина	30
2.4 Проектирование программных средств информационной системы интернет-магазина	30
2.4.1 Проектирование информационной системы интернет-магазина.....	31
2.4.2 Диаграмма состояний информационной системы интернет-магазина.....	34

2.4.3 Диаграмма деятельности информационной системы интернет-магазина.....	34
2.4.4 Диаграмма последовательности информационной системы интернет-магазина	36
2.4.5 Диаграмма кооперации информационной системы интернет-магазина.....	38
2.5 Проектирование пользовательского интерфейса информационной системы интернет-магазина	39
3 Реализация программного обеспечения информационной системы интернет-магазина	51
3.1 Диаграмма компонентов информационной системы интернет-магазина.....	51
3.2 Представление развертывания.....	51
4 Контроль качества информационной системы интернет-магазина.....	53
4.1 Функциональное тестирование информационной системы интернет-магазина	53
4.2 Попарное тестирование информационной системы интернет-магазина.....	58
4.3 Метрики кода.....	59
4.4 Планирование и оценка бюджета проекта информационной системы интернет-магазина	61
Заключение	66
Список использованных источников	67
Приложение А. Глоссарий	68
Приложение Б. Листинг клиентского приложения на языке PHP	70
Приложение В. Листинг создания таблиц	73
Приложение Г. Листинг клиентского приложения на языке js.....	78
Приложение Д. Результаты функционального тестирования	80
Приложение Е. Листинг клиентского приложения на языке html.....	83

Введение

Информационные технологии и достижения вычислительной техники активно внедряются в социальные, экономические и политические процессы общества. В настоящее время невозможно представить успешное развитие бизнеса без использования в нем информационных систем. Предприниматели, которые активно используют технологии для оптимизации и продвижения своего дела, ежегодно увеличивают оборот своей продукции. Согласно итогам 2020 года, которые подвела ассоциация компаний Интернет-торговли (АКИТ), объем продаж в онлайн магазинах увеличился на 58,5% по сравнению с предыдущим годом [1]. Наиболее перспективной для применения информационных технологий является сфера торговли.

Особо важное место на автомобильном рынке, емкость которого ежегодно увеличивается (исключение составляет только 2020 год), занимает продажа легковых автомобилей [2]. При этом неуклонно растет и оборот автомобильных запасных частей. Приобретение новых запчастей позволяет намного дольше эксплуатировать транспортное средство, заменяя неисправные детали. Автомобильные интернет-магазины позволяют оформить покупку товара, который доставят в доступные для клиента точки в любое удобное время. Однако такой полезной возможностью обладает только небольшое количество компаний. В основном функционал информационной системы сводится к демонстрации ассортимента физического магазина без возможности покупки онлайн.

Целью выпускной квалификационной работы является разработка информационной системы управления продажами и доставкой в интернет-магазинах автомобильных запасных частей.

Для достижения указанной цели были поставлены следующие задачи:

- проведение анализа предметной области и существующих информационных систем аналогичного назначения;

– постановка задачи на разработку информационной системы управления продажами и доставкой в интернет-магазине автомобильных запасных частей.

- определение требований к разрабатываемой системе;
- выполнение функционального моделирования системы;
- проектирование информационной системы;
- реализация информационной системы;
- тестирование информационной системы управления продажами и доставкой в интернет-магазине автомобильных запчастей.

С развитием рынка автозапчастей происходит ужесточение конкуренции среди предпринимателей, заинтересованных в успешном развитии своего бизнеса. Для того, чтобы привлечь больше клиентов, предприятию требуется учитывать потребности клиента и делать так, чтобы совершение покупок в магазине было удобнее чем у конкурента. Перспективу для решения данной проблемы открывает предоставление услуг доставки товара непосредственно из интернет-магазина.

Из сказанного выше становится очевидным то, что тема данной выпускной квалификационной работы является актуальной.

1 Анализ предметной области

1.1 Анализ предметной области информационных систем интернет-магазинов

Отрасль автомобильных запчастей занимает одно из важнейших мест на Российском рынке торговли. Это обусловлено тем, что больше половины семей в нашей стране имеют личный автомобиль, который несомненно нуждается в обслуживании [3]. Настолько высокий спрос на товар для средств передвижения вынуждает владельцев магазинов не только предлагать необходимые товары по доступным ценам, но и внедрять в свой бизнес информационные системы для более современного и качественного обслуживания клиентов.

Интернет-магазин сохраняет в себе принципы традиционного физического магазина, однако у него есть возможность обеспечить покупателя значительно большим объемом информации необходимой для потребителя, что способствует более высокой вероятности покупки товара. Кроме того, на веб-страницу возможно разместить рекламу услуг и товаров, потенциально интересующих клиента. Очередным преимуществом интернет-магазина является то, что он открыт для покупателя в любое удобное время.

К недостаткам традиционного магазина относится и то, что он имеет свою территориальную ограниченность, обусловленную тем, что покупателям выгоднее совершать покупки в ближайших точках. В свою очередь интернет-магазин не сталкивается с этой проблемой, поскольку доставку можно организовать вплоть до дома покупателя. Тем самым клиент может находиться в любом городе нашей страны и пользоваться услугами интернет-магазина.

Предприниматель может договориться с поставщиками запчастей, чтобы закупать нужный товар непосредственно после оформления заказа клиентом. Это позволяет отказаться от использования складских помещений, что дает возможность владельцу не тратить средства на аренду или покупку недвижимости.

Основываясь на всех перечисленных достоинствах интернет-магазина, возникает потребность создания информационной системы для торговли автомобильными запасными частями, которая смогла бы обеспечить продажу товара онлайн, с последующим оформлением заявки на доставку в место назначения, удобное для клиента.

1.2 Анализ аналогичных информационных систем интернет-магазинов авто запчастей

Рассмотрим наиболее популярные интернет-магазины, относящиеся как к Пензе, так и к другим городам, – это «Auto3n.ru», «Exist.ru», «adnexpress.ru», «Пятаяпередача.рф». Для их сравнения выделим критерии, напрямую влияющие на эффективность применения подобных информационных систем (ИС):

- наличие личного кабинета клиента;
- понятный и удобный пользовательский интерфейс;
- возможность оплаты покупок онлайн;
- наличие доставки;
- наличие самовывоза;
- удобный каталог товаров;
- возможность вернуть товар и потраченные средства в случае брака;
- наличие информации о магазине;
- наличие мобильной версии магазина..

Интернет-магазин «Auto3n.ru» является подразделением крупнейшего поставщика автозапчастей в Европе. Основным направлением деятельности компании является срочная поставка автозапчастей. Московский склад запчастей «АВТОЗН» обладает достаточными логистическими возможностями для оперативных поставок по всему ассортименту автомобильных комплектующих для всех легковых иномарок европейских,

корейских и японских производителей. Ассортимент компании насчитывает более 2 864 000 уникальных и более 200 000 артикулов в наличии.

У данного магазина присутствует личный кабинет, для регистрации достаточно электронной почты, имени и фамилии. Для входа в него необходимо придумать пароль. В личном кабинете возможно осуществлять заказ и оплату, следить за новостями, оставлять отзывы, наблюдать за учетом ваших расходов и историей поиска, а также существует «гараж» с личными автомобилями пользователя, для того чтобы упростить поиск товара.

При использовании данной ИС можно понять, что каталог автозапчастей и пользовательский интерфейс удобны и понятны для пользователя. Интерфейс личного кабинета пользователя представлен на рисунке 1.

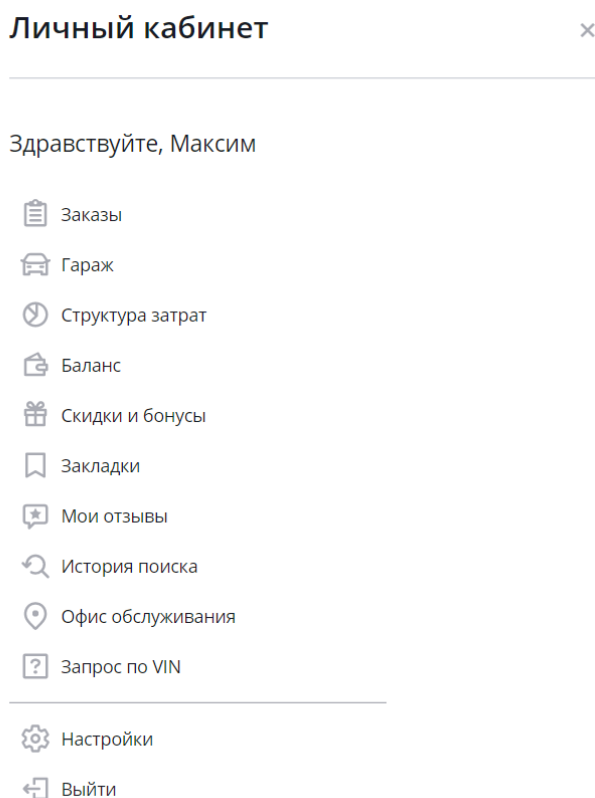


Рисунок 1 – Интерфейс личного кабинета пользователя

Так же пользователю доступны несколько фильтров для поиска товара в ассортименте:

- поиск по артикулу;
- поиск по VIN/Frame;
- общий поиск.

Каталог товаров распределен по категориям, что позволяет ускорить подбор товара.

Получить товар можно любым удобным способом, в любом регионе:

- самовывоз из магазина «Auto3N»;
- самовывоз из пункта выдачи транспортной компании;
- курьер «Auto3N» или курьер транспортной компании..

Интерфейс оформления самовывоза представлен на рисунке 2.

Каталог товаров Поиск по артикулу Артикул или номер детали Максим Корзина

Способ доставки

Самовывоз от 0 р. Курьер

117447, г. Москва, Севастопольский пр-т, д. 15, корп. 3
 AUTO3N Самовывоз
 Пн-Вт: 09:00-21:00
 Внимание! График работы в праздничные дни:
 01-03.05: 09:00-21:00
 08-10.05: 09:00-21:00
 Телефон +7 (495) 032-55-99

Бесплатно
 Не позже 28 Апреля
 Доставит сюда

Выбрать другой пункт самовывоза AUTO3N Самовывоз, IML, CDEK

1 товар на сумму 1 588 р. +620 р.

TIGAR Шина летняя "SIGURA TL 185/60R14 82T"
 93174 1 шт. 1 588 р.
 +620 р.

Добавить заметку

Комментарий к заказу и доставке

Комментарий к заказу

Бонусные баллы

Рисунок 2 – Интерфейс оформления самовывоза

Имеется информация о правилах возврата товара, но его можно оформить только в физическом магазине. Контактные данные, реквизиты и прочая информация о магазине в полной мере раскрывают все достоинства и функционал данного магазина. Так же присутствует мобильная версия сайта, которая так же обладает удобным и понятным интерфейсом.

Интернет магазин «Пятаяпередача.рф» относится к сети магазинов автозапчастей и автотоваров «Пятая передача». На сегодняшний день с интернет-магазином «Пятая передача.рф» работают не только розничные покупатели, но и многие компании: банки, заводы, СТО, торговые и логистические предприятия, автотранспортные компании.

Данный магазин взаимодействует напрямую более чем со 100 поставщиками по всей стране и предлагают запчасти для иномарок и отечественных автомобилей. В ассортименте присутствует линейка как оригинальных, так и неоригинальных запчастей практически всех существующих производителей.

В этом интернет-магазине есть личный кабинет, зарегистрироваться в нем можно следующим образом: необходимо заполнить поля, в которых требуется ввести имя, электронную почту, контактный номер, а также придумать логин и пароль для входа. Вся введенная информация остается конфиденциальна. В личном кабинете можно наблюдать за историей заказов, историей поиска, текущим балансом (для оплаты товаров), помимо этого в нем есть раздел «Мои автомобили» для быстрого подбора требуемых запчастей. Интерфейс личного кабинета пользователя представлен на рисунке 3.

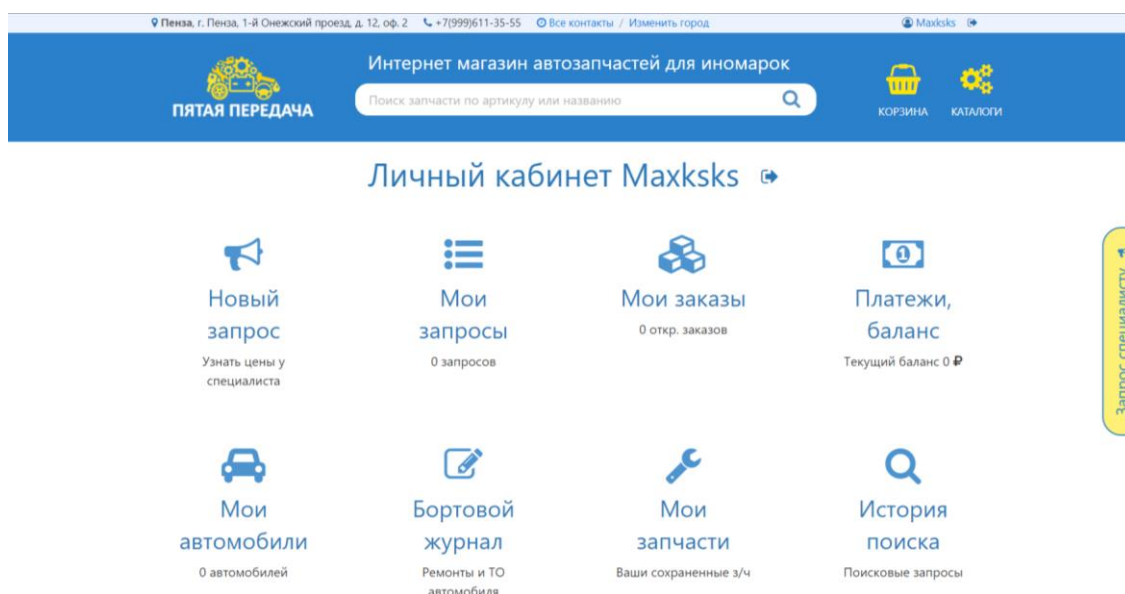


Рисунок 3 – Интерфейс личного кабинета пользователя

В интернет-магазине «Пятаяпередача.рф» можно отправить запрос специалисту – это удобный способ узнать цены товаров. Для того, чтобы найти запасную часть в данной информационной системе, требуется выбрать автомобиль, затем выбрать нужный товар и по артикулу перейти к заказу, где можно узнать цену без помощи онлайн консультанта. Однако совершение стольких операций для пользователя занимает некоторое время, поэтому можно считать интерфейс и каталог товаров недостаточно удобными. Также недостатком этого магазина является отсутствие доставки, получение товара осуществляется самовывозом из офиса, согласно графику работы. Интерфейс оформления самовывоза представлен на рисунке 4.

№	Артикул	Производитель	Наименование	Срок	Инфо	Цена/шт Р	Кол-во	Итого Р
1	ST304	ЗТОН	COUNTRY ST-304 МАСЛО ЧЕТЫРЕХТАКТ. П, С SAE 10W40 SJ, CF 0,5Л	5		170	1	170

Имя: Максим
Фамилия: Легашнев
Телефон: +7(999)610-52-37
E-mail: stalker-999@mail.ru
Тип отгрузки: Самовывоз из офиса
Адрес офиса: г.Пенза, г. Пенза, 1-й Онежский проезд, д. 12, с

☒ - согласен с офертой и конфиденциальностью *
☒ - покупаю товары по спец.предложению по Цена2 и согласен с полными правилами Акции *
☒ - наш менеджер позвонит Вам
* - обязательные для заполнения поля

Оформить заказ

Обратите внимание! Вы оформляете заказ с самостоятельным подбором запчастей, и тем самым соглашаетесь с условиями специального предложения (бонус) на приобретение запчастей. Цена2 является лишь ориентиром, фактически (фактически) цена будет отличаться.

Рисунок 4 – Интерфейс оформления самовывоза

Оплата при заказе через интернет-магазин производится онлайн, оплата при получении не доступна. Информации про возврат товара на сайте нет. У магазина есть мобильная версия.

Интернет-магазин «adnexpress.ru» – это крупный оптовый комплекс, в котором имеются в наличии и регулярно поставляются более 15000 наименований товаров. Доставка автозапчастей курьером осуществляется по Пензе и Заречному, а также возможна доставка по Пензенской и Саратовской

областям, Республике Мордовии. Есть альтернативный способ получения товара – самовывоз. Имеется возможность купить запчасти оптом. Следует отметить, что у магазина есть мобильная версия.

Для регистрации в личном кабинете необходимо заполнить информацию в обязательных полях: имя, фамилия, электронная почта и номер телефона и придумать пароль для входа в личный кабинет. Регистрация требует подтверждения электронной почты.

В личном кабинете можно найти информацию о собственных заказах, можно заполнить реквизиты своей карты для оплаты онлайн, в разделе «Автомобили» добавить свой, чтобы ускорить поиск деталей, узнать адреса доставки и контактную информацию компании. Интерфейс личного кабинета пользователя представлен на рисунке 5.

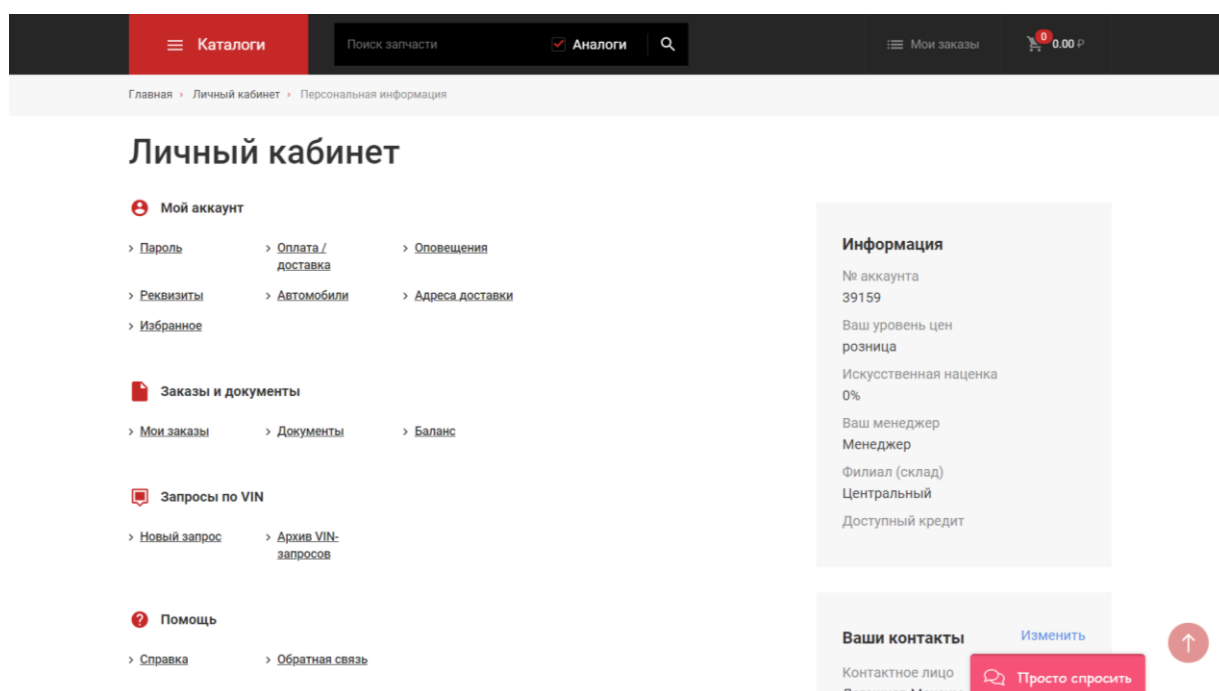


Рисунок 5 – Интерфейс личного кабинета пользователя

В интернет-магазине «Автолюбитель» есть возможность купить любую запасную часть всего за четыре шага. Для покупателей на сайте есть возможность оформить доставку покупки или оформить самовывоз. Доставка может осуществляться различными способами. Интерфейс оформления самовывоза представлен на рисунке 6.

Оформление заказа

Информация о клиенте

Контактное лицо, ФИО: Легашнев Максим Александрович

Телефоны: +7(999)610-52-37 (основной)

E-mail: stalker-999@mail.ru

Исполнение заказа

Способ оплаты: —

Способ доставки: самовывоз

Комментарий к заказу:

Проверка заказа

При необходимости проверки менеджером применимости выбранных вами номеров к вашему автомобилю требуется поставить отметку о необходимости проверки.

Условия сотрудничества

- 1 Мы не несём ответственность за применимость заказываемой детали к автомобилю Вашего клиента
- 2 Сотрудничество осуществляется на основании договора ОФ-8
- 3 Доставка осуществляется силами нашей компании. Способы доставки и минимальные параметры отправки согласовываются с каждым клиентом индивидуально.

Работа с нами – это

- 1 Различные формы оплаты: наличный расчет, безналичный расчет, банковский перевод.
- 2 Оперативная доставка продукции.
- 3 Право на получение специальных цен на товары

Просто спросить

Рисунок 6 – Интерфейс оформления самовывоза

Также клиенту предоставляется техническая поддержка консультантов. Можно найти товар по артикулу или в онлайн-каталоге на сайте. Явным недостатком является то, что на сайте отсутствует информация о возврате товара.

На текущий момент магазин «Автолюбитель» может предложить два способа оплаты:

- наличными: оплатить заказ можно на кассе магазина или при получении у курьера;
- пластиковой картой на сайте, в момент оформления заказа; оплата на сайте производится по защищенному соединению.

Интернет-магазин «Exist.ru» начал свою работу в 1999 году. Личный кабинет данного магазина состоит из нескольких разделов:

- содержимое виртуальной корзины, срок хранения товаров в корзине ограничен;
- гараж – для сохранения транспортных средства (для поиска деталей в каталогах);
- блокнот – для того, чтобы сохранить найденное в каталогах;

- заказы – мониторинг заказов, просмотр всех заказов, экспорт данных;
- запросы – подбор менеджером необходимых запчастей по описанию;
- баланс и оплата – состояние наличного и безналичного счетов, способы пополнения счета.

При использовании данной информационной системы можно понять, интерфейс является удобным и понятным для пользователя.

Для работы с заказами в системе «Exist.ru» используются внутренние персональные счета клиентов. Все средства, полученные от клиента, попадают на персональный счет клиента и могут быть использованы для работы с заказами. В случае заказа товарных позиций со сроком доставки 1 и более дней необходима предоплата. Средства попадают на персональный счет и используются для выполнения заказа. В случае снятия заказа или отказа клиента от заказа клиент вправе получить средства тем способом, которым был оплачен заказ.

На текущий момент магазин может предложить несколько способов оплаты:

- наличными (через кассу офиса);
- безналичный перевод денег из банка в банк;
- банковской картой On-Line.

Возврат товаров возможен только в офисе магазина, адреса офисов прописаны на сайте.

В этом магазине доступны несколько видов доставки:

- самовывоз;
- пункт получения заказа;
- курьерская доставка, которая возможна только внутри города.

Однако для нашего региона данный магазин предлагает только услуги самовывоза.

Интерфейс оформления самовывоза представлен на рисунке 7.

Пенза | Строителей +7 (906) 159-75-56

ПОИСК ПО САЙТУ

КАТАЛОГИ ЗАПРОС ПО VIN ПРОДАЖА АВТО АВТОЧКИ ФОРУМ КЛУБ ГАРАЖ

Личный кабинет Корзина Оформление заказа

1 ОФОРМЛЕНИЕ ЗАКАЗА 2 ОПЛАТА

ДОСТАВКА: **Самовывоз**

Товар должен быть получен заказчиком в офисе Exist по адресу офиса регистрации: Пенза, проспект Строителей, д.2Г— Подробнее

АДРЕС МАГАЗИНА: Пенза, проспект Строителей, д.2Г [Выбрать другой адрес](#)

ПОЛУЧАТЕЛЬ: **Легашнев Максим Александрович** [Редактировать данные](#)
 +79996105237
 stalker-999@mail.ru

СКИДКИ: [?](#)

КУПОНЫ
 Выберите купон из списка или введите новый

ПРОВЕРКА ПО ТС: [ДОБАВИТЬ ПРОВЕРКУ](#) [?](#)

Рисунок 7 – Интерфейс оформления самовывоза

В интернет-магазине представлены три способа поиска товара:

- в одном из каталогов;
- по номеру автозапчасти;
- по параметрам автомобиля (по VIN).

Удобной функцией на данном сайте является то, что при оформлении заказа можно проверить применимость детали к собственному автомобилю: для этого на странице «корзина», перед оформлением заказа, необходимо отметить «проверить заказ» и заполнить данные автомобиля. Есть возможность заходить с мобильного устройства, т.к. есть мобильная версия. Результаты анализа информационных систем представлены в таблице 1.

Таблица 1 – Сравнительный анализ аналогичных информационных систем.

Критерий для сравнения	«Auto3n.ru»	«Exist.ru»	«Adnexpress.ru»	«Пятаяпередача.рф»
Наличие личного кабинета клиента	Да	Да	Да	Да

Продолжение таблицы 1

Критерий для сравнения	«Auto3n.ru»	«Exist.ru»	«Adnexpress.ru»	«Пятаяпередача.рф»
Понятный и удобный пользовательский интерфейс	Да	Да	Да	Да
Возможность оплаты покупок онлайн	Да	Да	Да	Да
Наличие доставки	Да	Да, но для нашего города доступен только самовывоз	Да	Нет
Наличие самовывоза	Да	Да	Да	Да
Удобный каталог товаров	Да	Да	Да	Нет
Возможность вернуть товар и потраченные средства	Да, но только в офисе продаж	Да, но только в офисе продаж	Нет	Нет
Наличие информации о магазине	Да	Да	Да	Да
Наличие мобильной версии магазина	Да	Да	Да	Да

В результате сравнительного анализа стало ясно, что на рынке автомобильных запасных частей существует довольно много сервисов продажи продукции с возможностью доставки. Рассмотренные интернет-магазины удовлетворяют минимальным требованиям клиента для покупки и заказа автозапчастей. Функционал большинства из них позволяет удобно и быстро подобрать нужный товар, но в некоторых системах этот процесс немного усложнен, что может доставить клиенту некоторые неудобства. Также оформить возврат товара и потраченных средств через интернет-магазин невозможно, лишь часть из сравниваемых систем предлагают клиенту оформить возврат только в офисе продаж. Доставка товара в любой город

поддерживается не всеми информационными системами, но они предлагают услуги самовывоза заказанного товара.

Исходя из результатов сравнения, было принято решение разработать информационную систему интернет-магазина, которая будет позволять пользователю оформить доставку в любой город. Кроме того, подбор товара в каталоге не будет затруднительным для покупателя. В данной информационной системе будут сохранены положительные черты рассмотренных интернет-магазинов. Такие как наличие личного кабинета пользователя, возможность оплаты покупок онлайн, а также будет обладать понятным и удобным пользовательским интерфейсом.

1.3 Анализ требований к информационной системе интернет-магазина

Анализ требований – часть процесса разработки программного обеспечения, включающая в себя сбор требований к программному обеспечению (ПО), их систематизацию, выявление взаимосвязей, а также документирование. Можно выделить нефункциональные и функциональные требования.

Нефункциональные требования – это свойства, которыми должен обладать проект. Также к нефункциональным требованиям относятся ограничения проекта, но они не должны определять его поведение.

Функциональные требования определяют задачи или действия, которые проект может исполнить или же, что проект выполнять не должен.

1.4 Нефункциональные требования к информационной системе интернет-магазина

Разрабатываемая информационная система должна соответствовать следующим нефункциональным требованиям.

Информационная система должна запускаться на веб-странице при помощи любого из современных браузеров. Система будет разработана с

помощью фреймворка Laravel, созданного для скриптового языка общего назначения PHP, в редакторе кода Visual Studio Code.

PHP был специально создан для веб-разработок. Главная область применения PHP - написание скриптов, работающих на стороне сервера; таким образом, PHP способен выполнять все то, что выполняет любая другая программа CGI, например, обрабатывать данные форм, генерировать динамические страницы или отсылать и принимать cookies. Но PHP способен выполнять намного больше [4].

Laravel – это фреймворк для разработки веб-приложений. Он позволяет ускорить реализацию стандартных для большинства веб-проектов задач. Кроме того, данный фреймворк является самым популярным среди разработчиков, использующих язык PHP.

Для разработки информационной системы интернет-магазина были выделены следующие нефункциональные требования:

- необходимо исключить появление ситуаций, приводящих к аварийному завершению работы системы;
- разработать удобный и функциональный интерфейс для пользователя системы.

1.5 Функциональные требования к информационной системе интернет-магазина

Перед разрабатываемой информационной системой стоит задача обеспечения необходимого функционала как для клиентов, так и для администратора интернет-магазина автозапчастей. Данная система должна позволять клиенту совершать следующие действия:

- регистрация пользователя;
- аутентификация пользователя;
- поиск автомобильных запчастей по названию;
- поиск товара по категориям;
- добавление товара в список избранных товаров;

- добавление товара в корзину;
- оформление покупки с последующей доставкой.

Для администратора информационная система должна предоставлять функционал управления интернет-магазином:

- аутентификация пользователя;
- просмотр данных о магазине в удобной форме;
- добавление новых позиций товара;
- добавление новых категорий товаров;
- просмотр доставок.

Проектирование диаграмм будет проводиться в среде ArgoUML. Исходя из описанных требований, была разработана диаграмма вариантов использования для актера клиент. Данная диаграмма демонстрируется на рисунке 8.

Данные диаграммы описывают функциональные требования для системы. В результате проектирования было выделено два актера, такие как клиент и администратор.

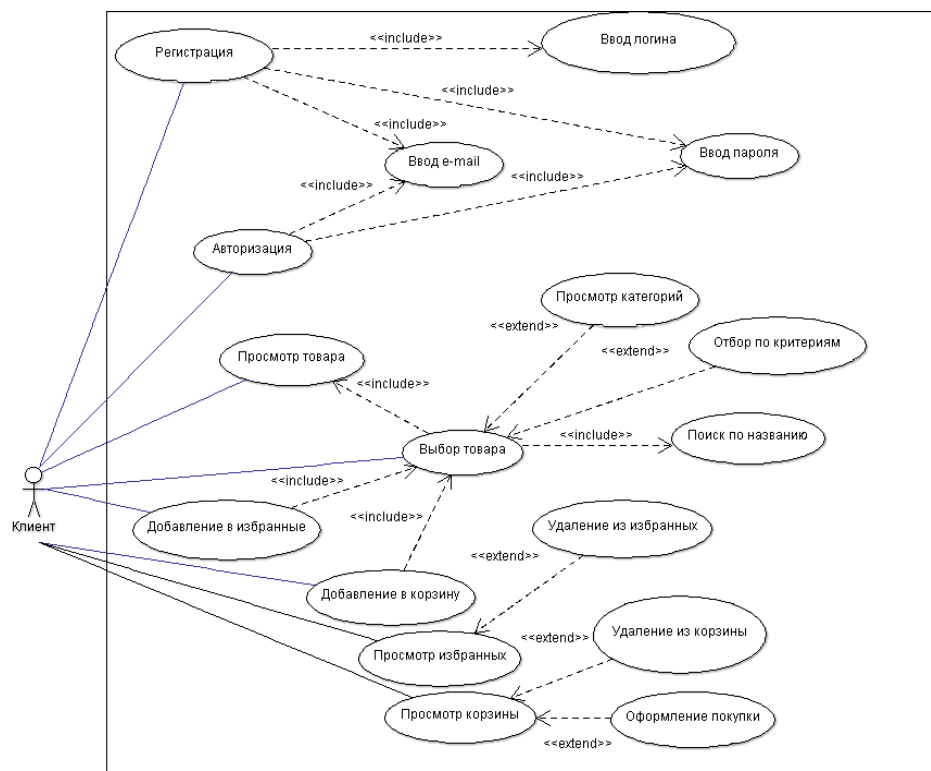


Рисунок 8 – Диаграмма вариантов использования для клиента

На рисунке 9 представлена диаграмма вариантов использования для актера администратор.

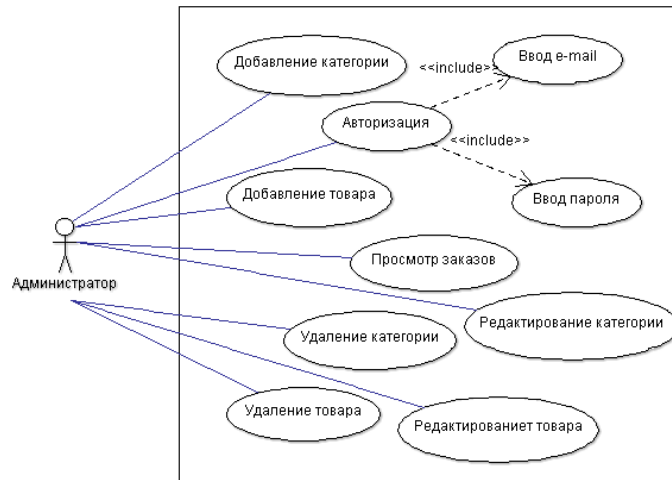


Рисунок 9 – Диаграмма вариантов использования для администратора

Клиент выступает в роли зарегистрированного, авторизованного пользователя. Он может просматривать каталог товаров посредством перехода в нужную категорию.

На странице категории клиент имеет возможность выставить критерии для отбора товара. Так же он может воспользоваться строкой поиска товара по названию, чтобы сразу найти нужный. Конкретный товар может быть добавлен пользователем в список избранных товаров или же в корзину для дальнейшего оформления покупки и доставки.

Администратор так же должен быть авторизован при помощи своей электронной почты и пароля. Введя их, пользователь получает доступ к специальной странице – панели администрирования.

В данной панели он может просматривать информацию об интернет магазине, а также влиять на его контент. Он имеет возможность добавления новых категорий, позиций товаров, увеличения или уменьшения их количества, редактирования, удаления, а также просмотр статусов доставок на оформленные покупки.

1.6 Функциональное моделирование информационной системы интернет-магазина

При проектировании информационной системы управления продажами и доставкой в интернет-магазине автозапчастей были составлены следующие бизнес-процессы. Они представлены в виде диаграмм, созданных при помощи СА ERwin Process Modeler. Это CASE-средство визуального моделирования, поддерживающее такие методологии моделирования, как IDEF0, IDEF3 и DFD.

IDEF0 является методологией графического описания деятельности какой-либо организации как множества связанных функций. С ее помощью возможно исследовать процессы организации, не связывая их с объектами, обеспечивающими их реализацию [5]. Контекстная диаграмма бизнес-процессов проектируемой системы в нотации IDEF0 представлена на рисунке 10.

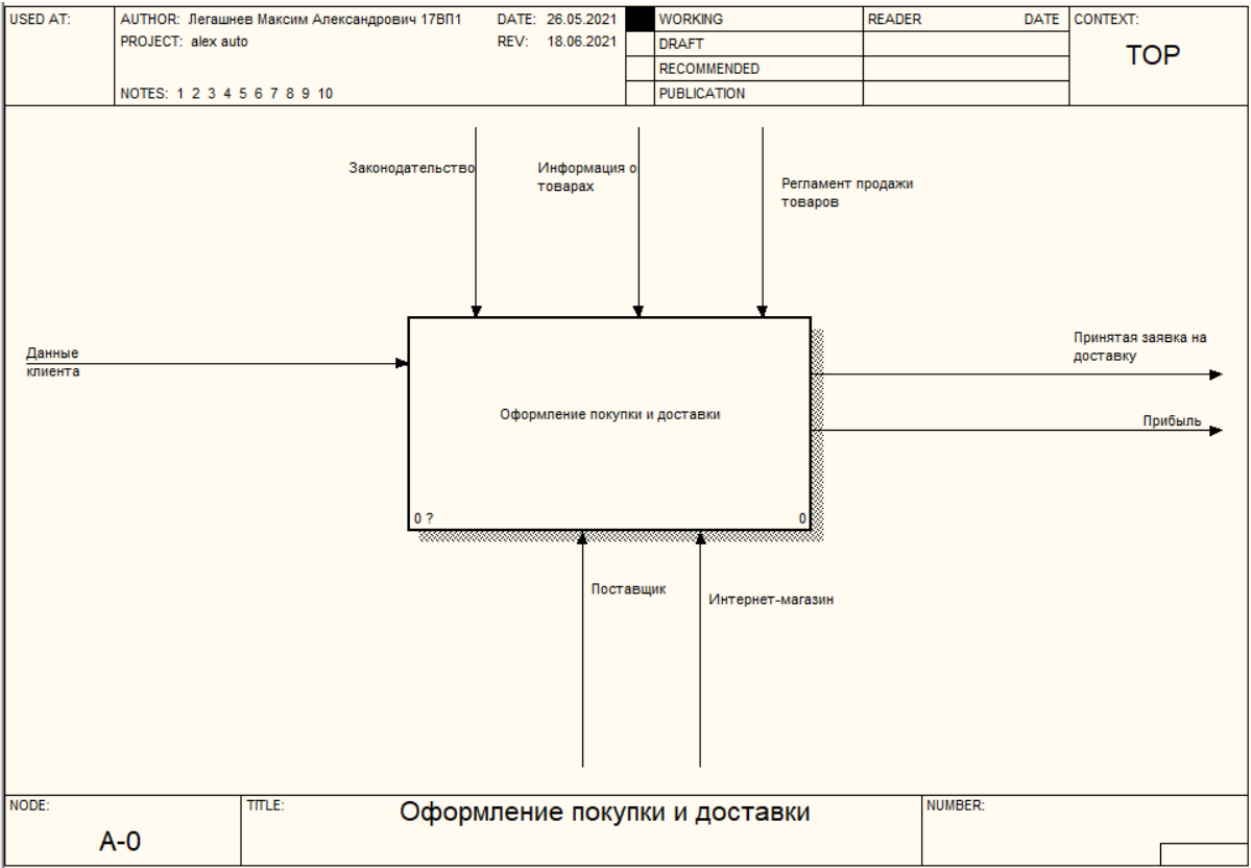


Рисунок 10 – Контекстная диаграмма в нотации моделирования IDEF0

На вход поступают данные клиента, которые система получает из специальных форм ввода данных. Механизмами представлены поставщики и сам интернет-магазин. Управляющим воздействием являются законодательство, регламентирующее правила продажи продукции, информация о товарах. Результатом будет отправление заявки на доставку запчастей поставщику и прибыль от продажи.

Далее происходит процесс разбиения системы на фрагменты, для ее более подробного анализа. Главный процесс был разделен на следующие компоненты: поиск и просмотр товаров, добавление товаров в корзину, а также оформление покупки. На рисунке 11 представлена декомпозированная диаграмма.

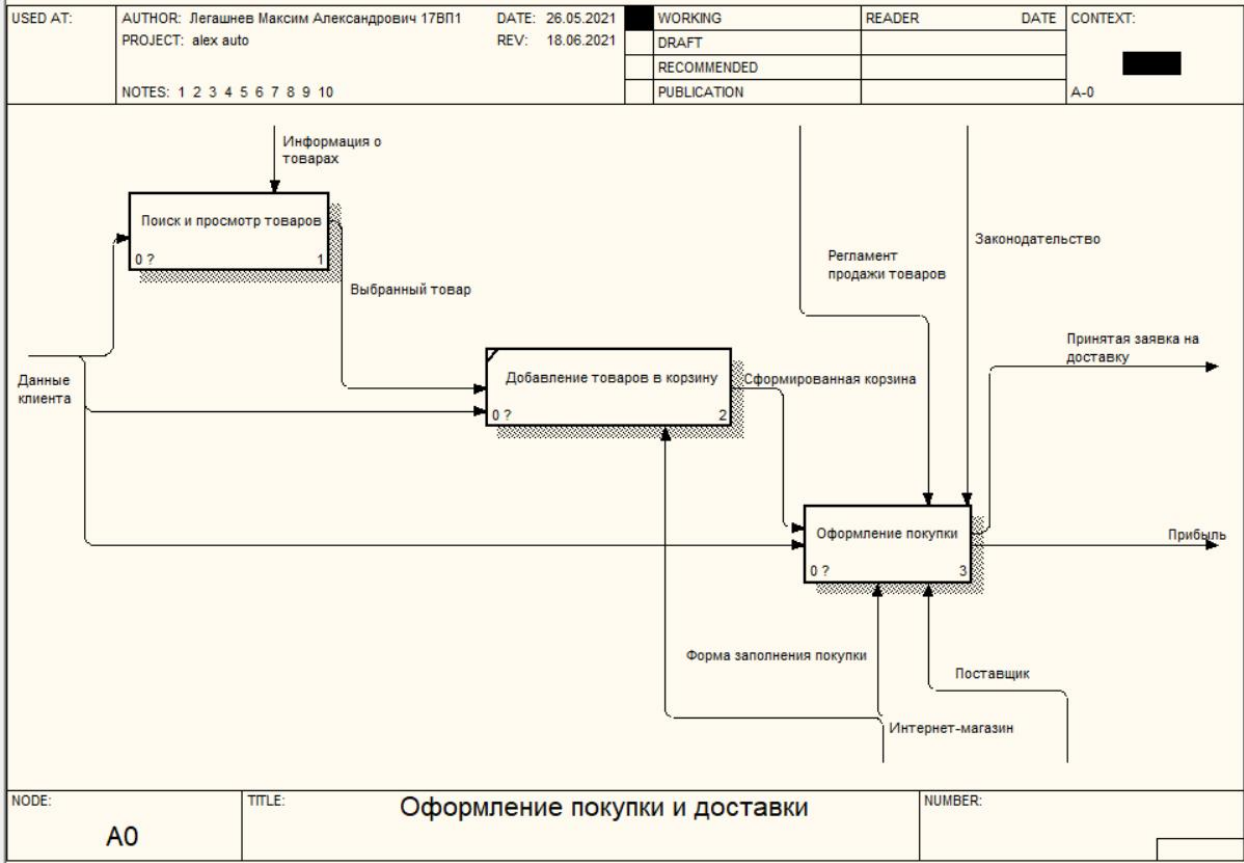


Рисунок 11 – Диаграмма декомпозиции в нотации моделирования IDEF0

Клиент производит анализ представленного каталога товаров. Затем нужный товар он добавляет в корзину, чтобы в дальнейшем приступить к

оформлению покупки. После совершения покупки клиенту необходимо оформить доставку для получения товара.

Декомпозиция бизнес процесса поиска и просмотра товаров происходит в нотации IDEF3. Данная методология является стандартом документирования технологических процессов. Она предоставляет возможности для наглядного исследования и моделирования сценариев процессов, происходящих на предприятии [6]. Просмотр товаров для клиента становится доступен только после регистрации или авторизации. Декомпозиция бизнес процесса поиска и просмотра товаров представлена на рисунке 12.

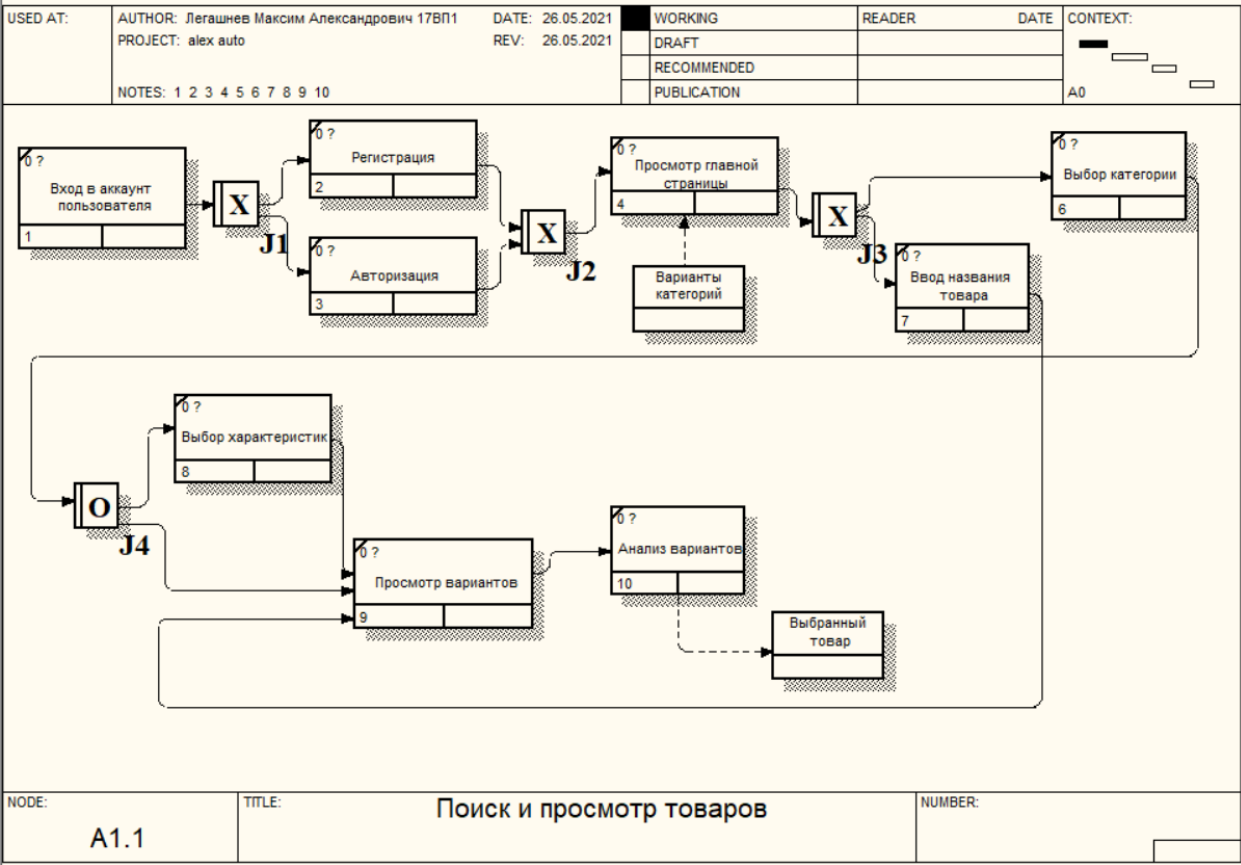


Рисунок 12 – Диаграмма декомпозиции в нотации моделирования IDEF3

Выполнение шага регистрации или авторизации показано перекрестком, исключающим «ИЛИ». После этого клиент попадает на главную страницу и ему доступен просмотр категорий. Далее пользователь попадает на еще один перекресток исключающего «ИЛИ». Он может сразу ввести название нужного ему товара и перейти к его просмотру. Или же клиент воспользуется выбором

категорий. Затем он попадет на перекресток асинхронного «ИЛИ», после которого он может сразу перейти к просмотру каталога товаров, но пользователь так же имеет возможность после перекрестка подобрать товары по критериям, которые он сам выберет. Далее после просмотра вариантов товаров клиент переходит к их анализированною, после чего может выбрать нужный товар, что является завершающим шагом.

1.7 Средства разработки информационной системы интернет-магазина

Для создания информационной системы управления продажами и доставкой в интернет-магазине автомобильных запчастей в качестве языка программирования был выбран PHP, а в качестве редактора кода для разработки клиентской части был выбран Visual Studio Code, серверная часть реализована на платформе Open Server, а язык разметки html был выбран для реализации графической части приложения.

Выбор языка программирования PHP был обусловлен тем, что он специально разработан для веб-разработки. Кроме того, код написанный на этом языке может напрямую внедряться в html код.

Visual Studio Code был выбран, так как это удобный редактор исходного кода. После настройки этого редактора под себя, он сильно упрощает процесс написания кода для разработчика.

Реализация серверной части проходила на платформе Open Server. Это портативный локальный WAMP/WNMP сервер, имеющий многофункциональную управляющую программу и большой выбор подключаемых компонентов.

В качестве системы управления базами данных была выбрана MySQL. Данная система также имеет клиент-серверную модель, которая является основой для информационной системы управления продажами и доставкой в интернет-магазине автозапчастей.

2 Проектирование информационной системы интернет-магазина

2.1 Базовая архитектура информационной системы интернет-магазина

Разрабатываемая информационная система интернет магазина автомобильных запчастей имеет клиент-серверную архитектуру. Данная архитектура позволяет оптимально распределить программные ресурсы, а также обеспечивает эффективное функционирование информационной системы. Архитектура представлена на рисунке 13.

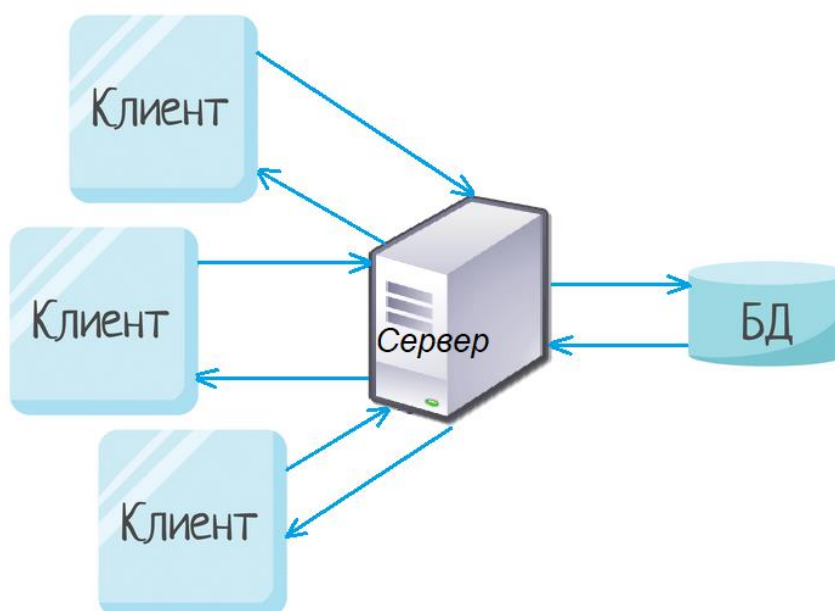


Рисунок 13 – Схема клиент-серверной архитектуры

Клиент-серверная архитектура имеет три звена взаимодействия:

- клиент;
- сервер;
- база данных.

В роли клиента в данной информационной системе выступает сайт интернет-магазина. Назначением клиента является отправка запросов от пользователя на сервер для дальнейшей обработки или выполнения каких-либо действий.

Роль сервера заключается в хранении базы данных и получении или добавлении в нее информации. Фактически он связывает клиента с базой данных посредством приема от него запросов и отправки ответов на них обратно.

База данных хранит в себе все сведения, связанные с информационной системой. К ней обращается сервер и получает из нее данные.

2.2 Логическая модель базы данных информационной системы интернет-магазина

В результате анализа предметной области, а также функциональных требований к разрабатываемой системе была спроектирована модель «сущность-связь» для базы данных на логическом уровне. Проектирование базы данных на данном уровне подразумевает построение отношений между данными, которые представляются так же как в реальном мире. Кроме того, необходимо определить первичные ключи для сущностей и определить тип связей между ними.

Проектирование логической модели для базы данных производилось с помощью CASE-средства моделирования данных CA ERwin Data Modeler в нотации IDEF1X. Данная методология разработана специально для проектирования реляционных баз данных. Построенная логическая модель в нотации IDEF1X представлена на рисунке 14.

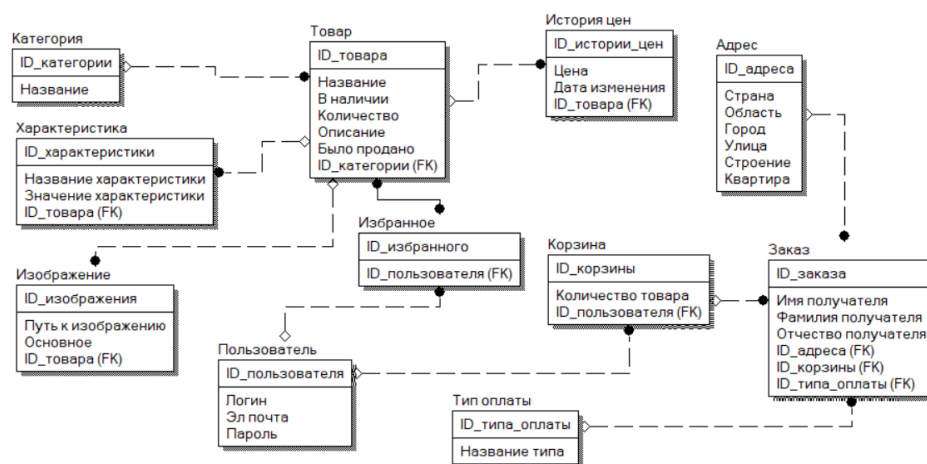


Рисунок 14 – Логическая модель

Далее будет совершен переход от логического уровня к физической модели базы данных.

2.3 Физическая модель базы данных информационной системы интернет-магазина

Физическая модель базы данных представляет собой описание реализации объектов логической модели. Физическая модель базы данных для информационной системы управления продажами и доставкой в интернет-магазине автомобильных запчастей в нотации IDEF1X представлена на рисунке 15.

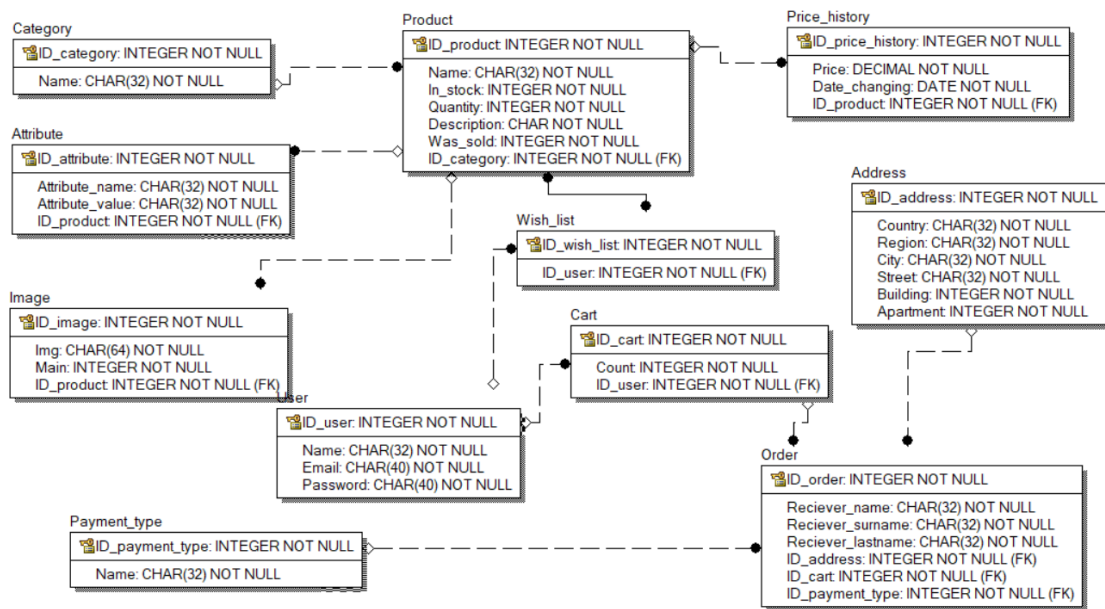


Рисунок 15 – Физическая модель

В базе данных MySQL используются целые, вещественные, а также символьные типы данных. В сущности «Product» в поле «Description» будет использоваться тип данных TEXT.

2.4 Проектирование программных средств информационной системы интернет-магазина

Для этапа проектирования в методологии UML были выбраны следующие виды поведенческих диаграмм: диаграмма классов, диаграмма

состояний, диаграмма деятельности, диаграмма последовательности, диаграмма кооперации.

2.4.1 Проектирование информационной системы интернет-магазина

Построение диаграммы классов для разрабатываемого проекта является наиболее распространенным способом моделирования. На данном этапе проектирования раскрывается структура системы. Происходит описание классов, интерфейсов и их связей. В классах и интерфейсах в свою очередь описываются атрибуты и методы. Спроектированная диаграмма классов для информационной системы управления продажами и доставкой в интернет-магазине автомобильных запчастей представлена на рисунке 16.

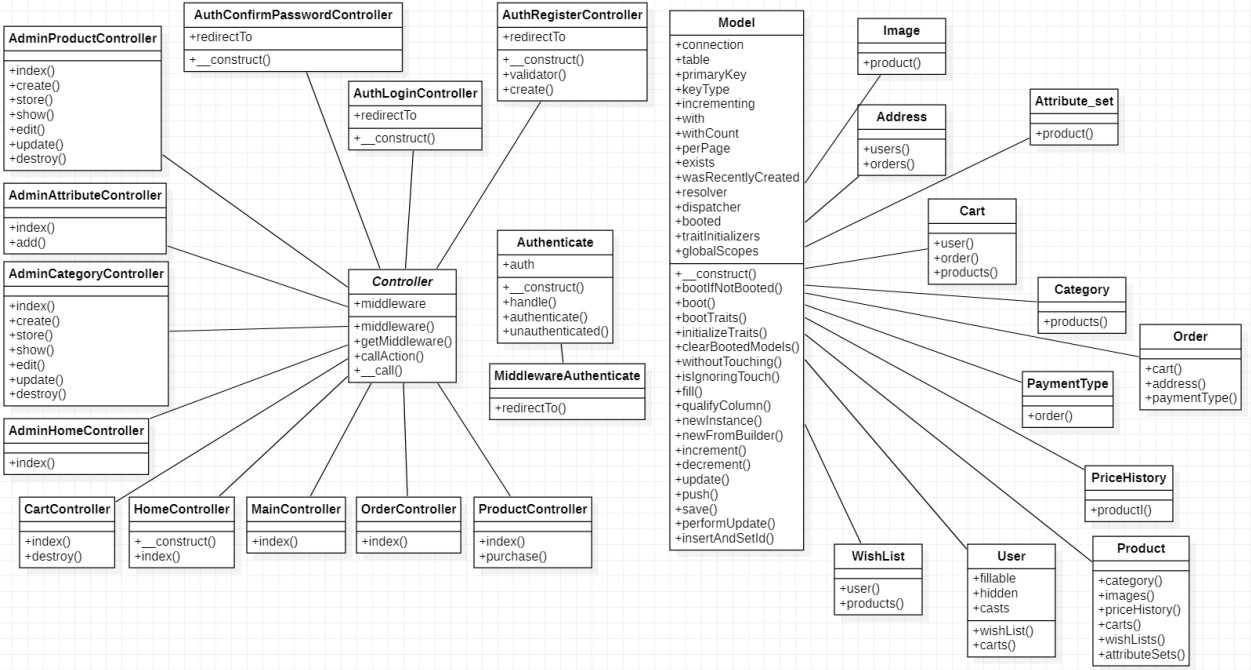


Рисунок 16 – Диаграмма классов

Так как в данном проекте был реализован паттерн программирования MVC, то классы условно можно разделить на контроллеры и модели. Описание представленных классов представлено в таблице 2.

Таблица 2 – Описание классов

№	Класс	Описание
1	Controller	Класс от, которого наследуются все классы контроллеры. Происходит описание основных свойств и методов для контроллеров.
2	AuthRegisterController	Класс контроллера, в котором реализована регистрация пользователя.
3	AuthLoginController	Класс контроллера, в котором реализована авторизация пользователя.
4	AuthConfirmPasswordController	Класс контроллера, в котором реализовано подтверждение пароля.
5	AdminProductController	Класс контроллера для панели администратора, в котором реализован функционал работы с товаром.
6	AdminAttributeController	Класс контроллера для панели администратора, в котором реализован функционал работы с характеристиками товара.
7	AdminCategoryController	Класс контроллера для панели администратора, в котором реализован функционал работы с категориями.
8	AdminHomeController	Класс контроллера для панели администратора, в котором реализован функционал главной страницы и отслеживания заказов.
9	CartController	Класс контроллера клиентской части, в котором реализован функционал работы с корзиной.
10	MainController	Класс контроллера клиентской части, в котором реализован функционал работы с главной страницей.
11	OrderController	Класс контроллера клиентской части, в котором реализован функционал работы с заказами.
12	ProductController	Класс контроллера клиентской части, в котором реализован функционал работы с товаром.
13	Model	Класс от, которого наследуются все классы модели. Происходит описание основных свойств и методов для моделей.
14	Image	Класс модели, в котором описаны связи модели изображения.

Продолжение таблицы 2

№	Класс	Описание
15	Address	Класс модели, в котором описаны связи модели адреса.
16	AttributeSet	Класс модели, в котором описаны связи модели характеристики.
17	Cart	Класс модели, в котором описаны связи модели корзины.
18	Category	Класс модели, в котором описаны связи модели категории.
19	Order	Класс модели, в котором описаны связи модели заказа.
20	PaymentType	Класс модели, в котором описаны связи модели типа оплаты.
21	PriceHistory	Класс модели, в котором описаны связи модели истории цен.
22	Product	Класс модели, в котором описаны связи модели товара.
23	User	Класс модели, в котором описаны связи модели пользователя.
24	WishList	Класс модели, в котором описаны связи модели избранного.
25	Authenticate	Класс аутентификации пользователя с хранящимися в нем свойствами и методами для аутентификации.
26	MiddlewareAuthenticate	Класс для перенаправления пользователя при аутентификации.

Выявлено двадцать шесть классов, которые будут реализованы в данной информационной системе. Они спроектированы согласно шаблону MVC для качественного и понятного написания кода.

MVC – это принцип разделения данных приложения, пользовательского интерфейса и управляющей логики на три отдельных компонента: модель, представление и контроллер. Благодаря этому модификация каждого компонента может осуществляться независимо.

2.4.2 Диаграмма состояний информационной системы интернет-магазина

Разработана диаграмма состояний для заказа товара. В данном виде UML диаграмм происходит описание поведения объекта в течение его жизни. Диаграмма состояний дает возможность спроектировать реакции на события для обозреваемого класса. Диаграмма состояний класса OrderController (заказ) представлена на рисунке 17.

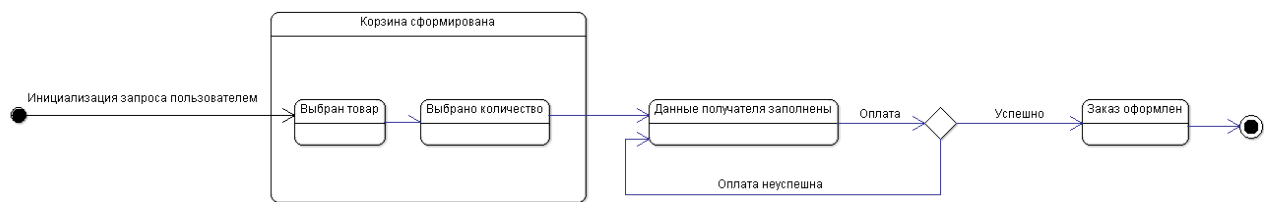


Рисунок 17 – Диаграмма состояний

Для оформления заказа клиенту необходимо сформировать корзину, выбрав товар и его количество. Далее открывается форма для заполнения данных клиента, таких как контактные данные и адрес. Затем происходит оплата. При выполнении этого шага возможны два условия. Оплата совершена успешно и тогда заказу присваивается состояние оформленного. Если же оплата совершена неуспешно, то заказу присваивается статус отклоненного. Таким образом заказ может иметь два состояния: оформлен и отклонен.

2.4.3 Диаграмма деятельности информационной системы интернет-магазина

На этапе проектирования информационной системы появляется необходимость не только проследить процесс изменения ее состояний, но также более детально рассмотреть алгоритмическую реализацию. Для решения этой задачи составляют диаграмму деятельности системы. В данной диаграмме происходит отображение логики и последовательности действий, выполняемых в системе. Состояние на диаграмме – это выполнение некоторой простой операции. Выполнение состояния происходит только после

выполнения предшествующего. На рисунке 18 представлена диаграмма деятельности информационной системы управления продажами и доставкой в интернет-магазине автомобильных запчастей.

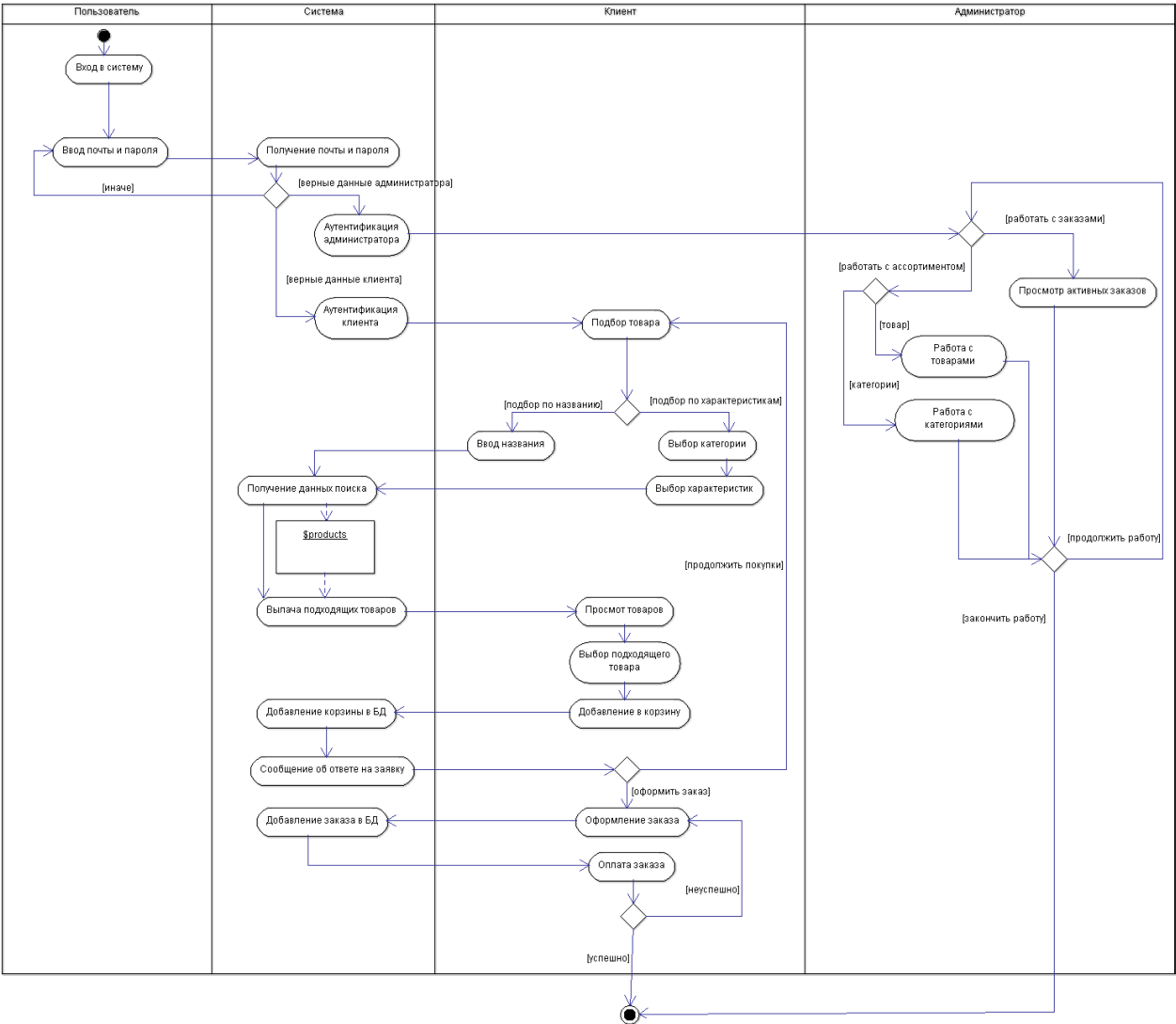


Рисунок 18 – Диаграмма деятельности

Для взаимодействия с системой пользователю необходимо авторизоваться. В соответствующие поля он вводит электронную почту и пароль. Если клиент или администратор с такими данными зарегистрированы, то их перенаправляют на соответствующие страницы. Для клиента это главная страница магазина, а для администратора – панель администратора.

Клиент начинает поиск подходящих ему товаров. Для этого он может воспользоваться поиском по названию, или же подобрать товар вручную.

Чтобы подобрать товар таким образом, клиент должен выбрать нужную ему категорию товара.

После этого он будет перенаправлен на страницу с каталогом товаров выбранной категории. На этой странице клиент может отфильтровать каталог посредством выбора нужных характеристик товара. После получения критериев для фильтрации или же название товара, система возвращает список товаров с нужными данными, которые отображаются на странице. Далее понравившийся товар, клиент выбирает нужное ему количество и добавляет его в корзину.

Для клиента в базе данных создается запись в таблицы корзины. Система перенаправляет пользователя на страницу просмотра корзины. Он может продолжить покупки, выполнив заново операции по поиску товара.

Так же клиент может из корзины перейти к оформлению заказа. Для этого система перенаправит его на соответствующую страницу для заполнения формы для заказа. После заполнения данных о заказе они добавятся в базу данных. Далее клиент сможет перейти к оплате.

Администратор, после авторизации сможет просматривать активные заказы, а также взаимодействовать с ассортиментом магазина. Он может выбрать работу с товарами или категориями.

2.4.4 Диаграмма последовательности информационной системы интернет-магазина

Диаграмма последовательности описывает сценарии системы. В данном виде диаграмм демонстрируются объекты и их коммуникационный аспект. Другими словами, отражаются взаимодействующие объекты, которые обмениваются между собой некоторой информацией [7].

В данном виде диаграмм наглядно показано время жизни объекта, то есть период времени, в течение которого объект существует в системе. Кроме того, объекты могут передавать друг другу управление. Для отображения прецедента «Выбор по характеристикам» диаграммы вариантов

использования была построена диаграмма последовательности, представленная на рисунке 19.

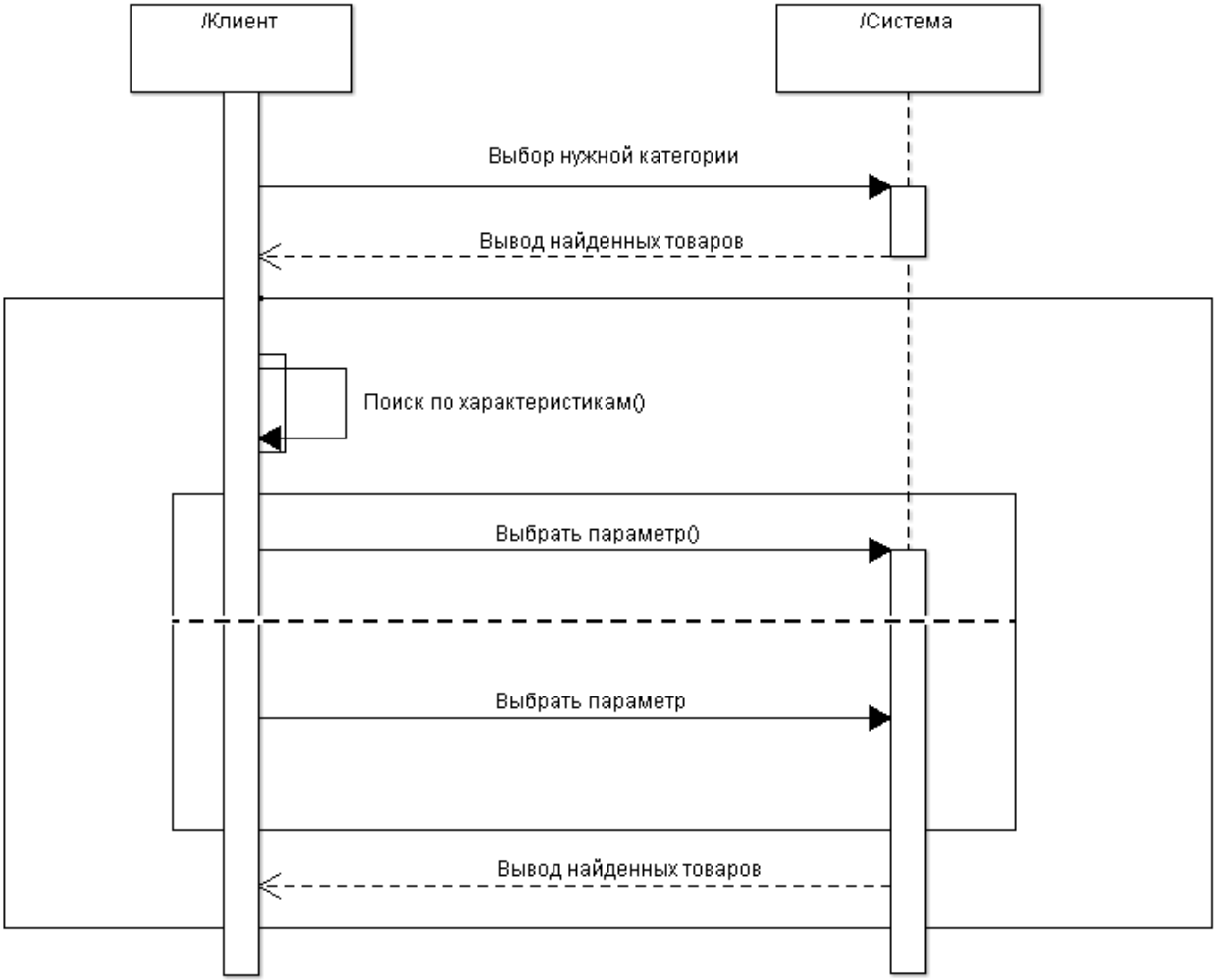


Рисунок 19 – Диаграмма последовательности

После подбора выбора категории товара пользователь может осуществить поиск по характеристикам. Если ему нужны дополнительные параметры поиска.

Для выбора по нескольким критериям в системе предусмотрено множество фильтров, например, такие, как поиск по бренду, поиск по цене. После применения необходимых фильтров система выводит новый список подходящих объектов.

2.4.5 Диаграмма кооперации информационной системы интернет-магазина

Диаграмма кооперации служит для описания структурных аспектов взаимодействия элементов в системе. То есть она позволяет отобразить последовательность взаимодействия объектов, а также их структурные отношения.

Порядковым номером на диаграмме определяется очередность взаимодействия элементов. Для отображения последовательности обмена сообщениями в процессе покупки товара была построена диаграмма кооперации, представленная на рисунке 20.



Рисунок 20 – Диаграмма кооперации

Диаграмма описывает процесс покупки товара в информационной системе. Она включает в себя три объекта: клиент, система, сервер, которые обмениваются сообщениями.

Объект «Клиент» оформляет покупку, заполнив свои данные. Производит оплату заказа, после чего ему в личный кабинет приходит информация и чек.

Объект «Система» получает от клиента данные о заказе, проверяет возможность заказа товара. После подтверждения от Сервера БД, отправляет клиенту форму для оплаты поездки. После успешной оплаты клиентом, отправляет данные о заказе в личный кабинет.

Объект «Сервер БД» осуществляет проверку наличия товара. Кроме того, сохраняет данные о заказе в системе.

2.5 Проектирование пользовательского интерфейса информационной системы интернет-магазина

Опираясь на диаграмму вариантов использования, был спроектирован графический интерфейс информационной системы управления продажами и доставкой в интернет-магазине автозапчастей.

После попадания на главную страницу магазина, пользователь может ознакомиться с ассортиментом как гость, также он может авторизоваться или зарегистрироваться, нажав соответствующие кнопки на верхней панели. Интерфейс регистрации пользователя представлен на рисунке 21.

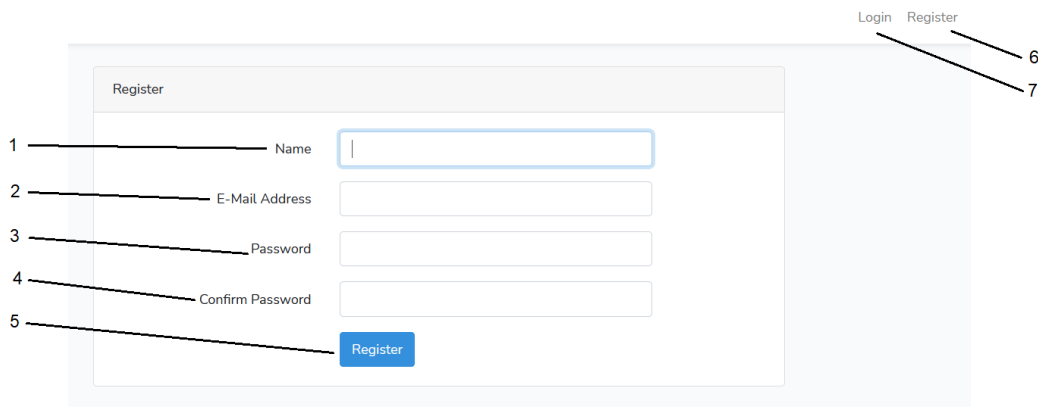


Рисунок 21 – Интерфейс регистрации

Интерфейс страницы регистрации включает в себя:

- 1 – поле ввода имени пользователя (логина);
- 2 – поле ввода электронной почты;
- 3 – поле ввода пароля;
- 4 – поле ввода подтверждения пароля;
- 5 – кнопка, подтверждающая регистрацию;
- 6 – переход на страницу авторизации;
- 7 – переход на страницу регистрации.

Для авторизации пользователь перенаправляется на соответствующую страницу. Интерфейс страницы авторизации представлен на рисунке 22.

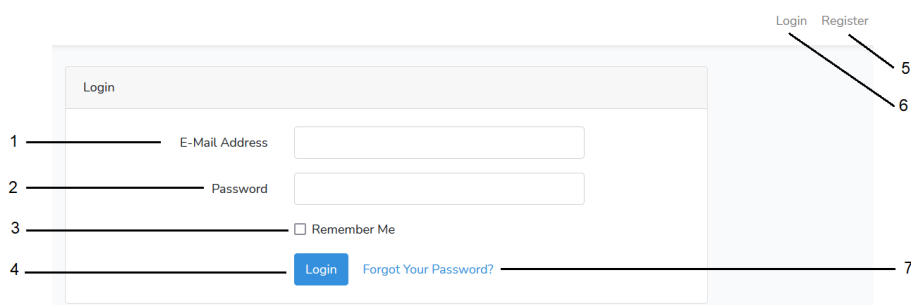


Рисунок 22 – Интерфейс авторизации

Интерфейс страницы авторизации включает в себя:

- 1 – поле ввода электронной почты;
- 2 – поле ввода пароля;
- 3 – поле проверки на желание пользователя запомнить данные входа;
- 4 – кнопка, подтверждающая авторизацию;
- 5 – переход на страницу авторизации;
- 6 – переход на страницу регистрации;
- 7 – поле ссылки на помощь при забытом пароле.

После успешной регистрации или авторизации в системе, пользователь перенаправляется на главную страницу магазина, на которой представлены категории товаров имеющихся в наличии. Интерфейс главной страницы с категориями представлен на рисунке 23.

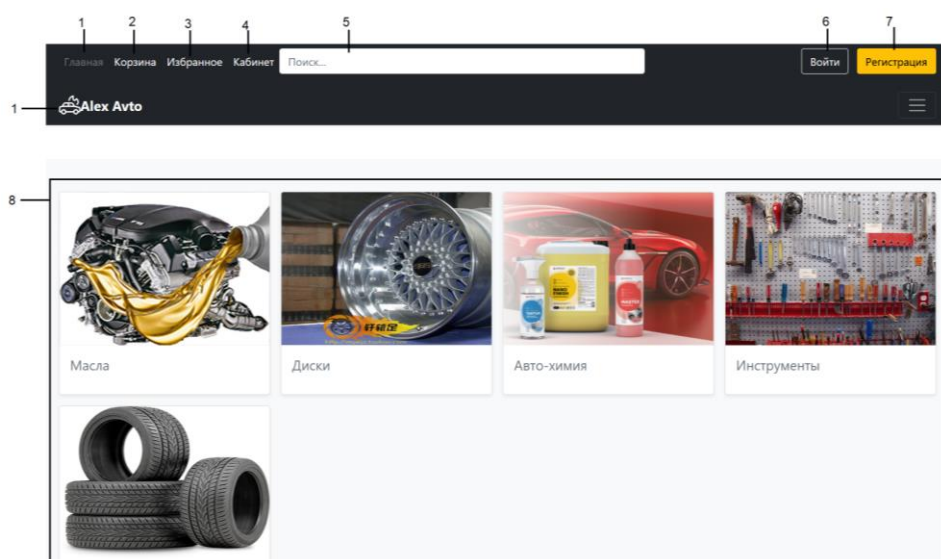


Рисунок 23 – Интерфейс главной страницы

Интерфейс страницы авторизации включает в себя:

- 1 – переход на главную страницу;
- 2 – переход в корзину;
- 3 – переход в избранное;
- 4 – переход в личный кабинет;
- 5 – поле ввода названия товара для поиска;
- 6 – переход на страницу авторизации;
- 7 – переход на страницу регистрации;
- 8 – выбор категории.

Выбирая категорию товара с главной страницы, пользователь перенаправляется на страницу с каталогом товаров, выбранной категории. Где представлены возможности отбора товара по их характеристикам. На рисунке 24 приведено описание интерфейса каталога товаров в категории «Шины».

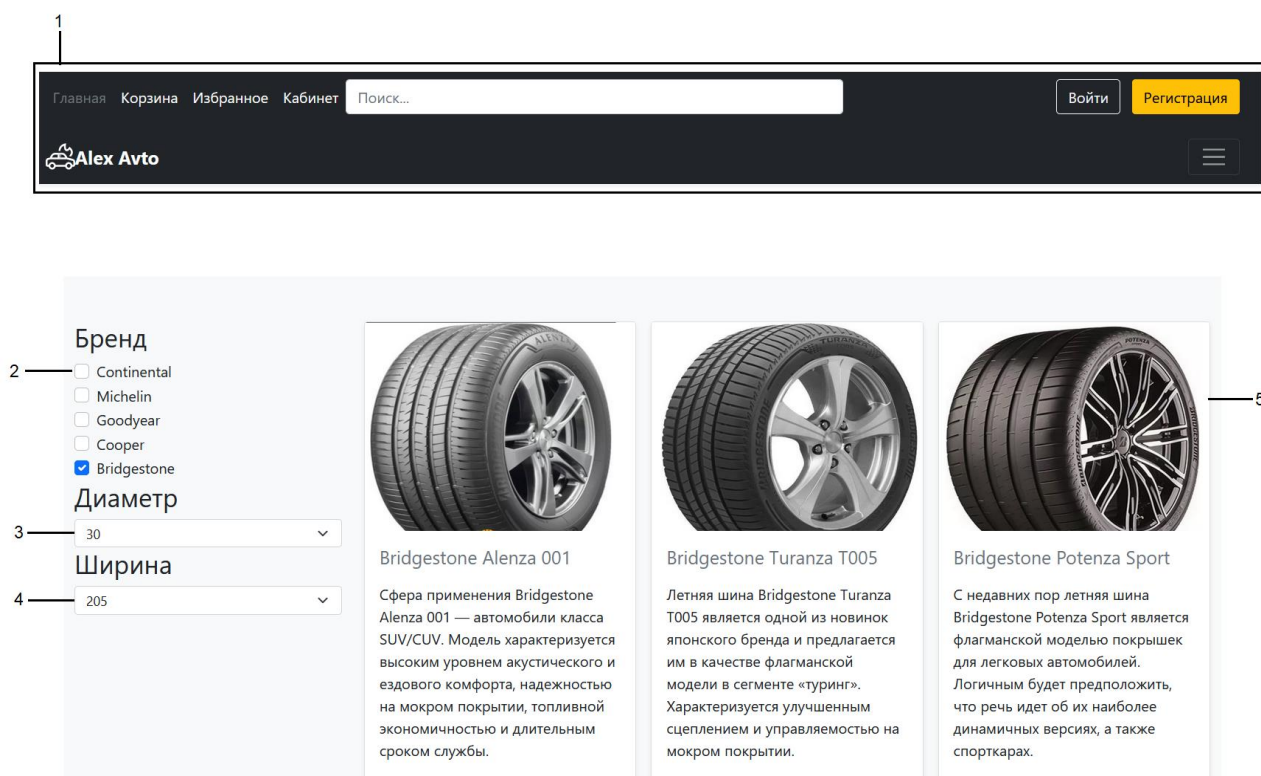


Рисунок 24 – Интерфейс каталога товаров

Интерфейс страницы каталога товаров включает в себя:

- 1 – интерфейс заголовка описан выше;

- 2 – выбор бренда производителя товара;
- 3 – выбор характеристики товара (диаметр колеса);
- 4 – выбор характеристики товара (ширина колеса);
- 5 – выбор конкретного товара.

После выбора товара, система перенаправляет клиента на страницу конкретного товара. На ней он может ознакомиться с ценой, прочитать подробное описание, а также добавить товар в корзину покупок либо в избранные товары для того чтобы вернуться к этому товару позднее. Интерфейс страницы товара представлен на рисунке 25.



Рисунок 25 – Интерфейс страницы товара

Интерфейс страницы каталога товаров включает в себя:

- 1 – интерфейс заголовка описан выше;
- 2 – изображение товара;
- 3 – название товара;
- 4 – цена товара;
- 5 – описание товара;
- 6 – кнопка добавления товара в корзину;
- 7 – кнопка добавления товара в избранные.

После добавления товара в корзину пользователь перенаправляется на страницу корзины, где представлены все добавленные позиции товаров с их

названиями и ценой. Пользователь имеет возможность выбрать нужное количество конкретного товара, задав нужное количество в специальной форме ввода, а также удалить позицию из корзины.

Пользователь может выбрать продолжить покупки. Для этого нужно нажать соответствующую кнопку. Для оформления покупки необходимо нажать кнопку оплатить. Интерфейс страницы корзины представлен на рисунке 26.



Рисунок 26 – Интерфейс корзины

Интерфейс страницы каталога товаров включает в себя:

- 1 – интерфейс заголовка описан выше;
- 2 – колонка наименований товаров;
- 3 – колонка количества товаров с полем ввода количества;
- 4 – колонка цен товаров;
- 5 – кнопка удаления позиции товара из корзины;
- 6 – итоговая цена корзины;
- 7 – кнопка переадресации для продолжения покупок;
- 8 – кнопка переадресации на страницу оформления заказа;
- 9 – позиция товара.

При нажатии на кнопку оплаты из корзины клиент направится на страницу оформления заказа. Там он должен будет ввести все данные необходимые для оплаты и доставки. Интерфейс страницы корзины

представлен на рисунке 27.

The image shows a web form for placing an order. It consists of several text input fields, a dropdown menu, and a button. Numbered lines (1-12) point to the following elements:

- 1: Input field for 'Фамилия' (Surname) with placeholder 'Введите фамилию'.
- 2: Input field for 'Имя' (Name) with placeholder 'Введите имя'.
- 3: Input field for 'Отчество' (Patronymic) with placeholder 'Введите отчество'.
- 4: Input field for 'Номер телефона' (Phone number) with placeholder 'Enter номер телефона'.
- 5: Input field for 'Страна' (Country) with placeholder 'Введите название страны'.
- 6: Input field for 'Область' (Region) with placeholder 'Введите область'.
- 7: Input field for 'Город' (City) with placeholder 'Введите город'.
- 8: Input field for 'Улица' (Street) with placeholder 'Введите улицу'.
- 9: Input field for 'Строение' (Building) with placeholder 'Введите номер дома'.
- 10: Input field for 'Квартира' (Apartment) with placeholder 'Введите номер квартиры'.
- 11: Dropdown menu for 'Тип оплаты' (Payment type) with 'Кредитная карта' (Credit card) selected.
- 12: 'Оплата' (Payment) button.

Рисунок 27 – Интерфейс страницы оформления заказа

Интерфейс страницы каталога товаров включает в себя:

- 1 – поле ввода фамилии;
- 2 – поле ввода имени;
- 3 – поле ввода отчества;
- 4 – поле ввода номера телефона;
- 5 – поле ввода страны;
- 6 – поле ввода области;
- 7 – поле ввода города;
- 8 – поле ввода улицы;
- 9 – поле ввода номера здания;
- 7 – поле ввода квартиры;
- 8 – выбор типа оплаты;
- 9 – кнопка оплаты.

При нажатии клиентом в верхнем меню поля «Избранные» он попадет на соответствующую страницу, где он может наблюдать список избранных товаров. Интерфейс страницы избранных товаров представлен на рисунке 28.

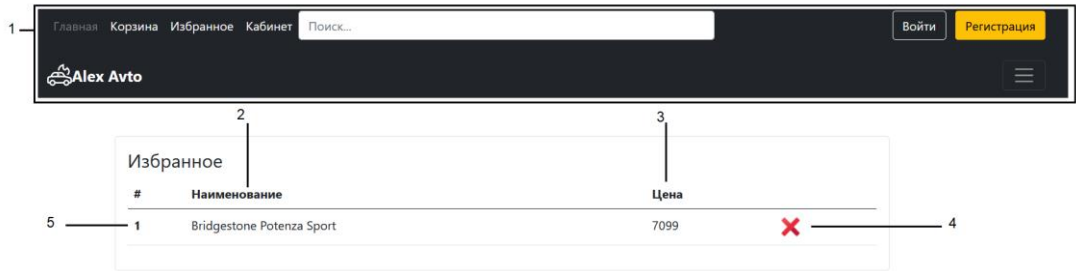


Рисунок 28 – Интерфейс страницы избранных товаров

Интерфейс страницы каталога товаров включает в себя:

- 1 – интерфейс заголовка описан выше;
- 2 – колонка наименований товаров;
- 3 – колонка цен товаров;
- 4 – кнопка удаления позиции из списка;
- 5 – номер позиции товара.

При нажатии клиентом в верхнем меню поля «Кабинет» он попадет на соответствующую страницу, где он может наблюдать список заказов и их статус. Интерфейс страницы избранных товаров представлен на рисунке 29.

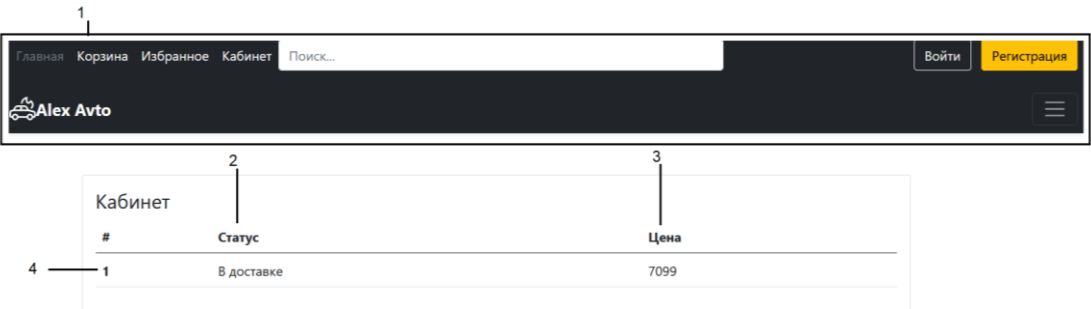


Рисунок 29 – Интерфейс личного кабинета

Интерфейс страницы каталога товаров включает в себя:

- 1 – интерфейс заголовка описан выше;
- 2 – колонка статуса заказов;
- 3 – колонка цен заказов;
- 4 – ID заказа.

Если при авторизации пользователь введет учетные данные администратора, то он попадет в панель администратора. Из этой панели

производится управление магазином. На главной странице также отображен список заказов для отслеживания. Интерфейс панели администратора представлен на рисунке 30.

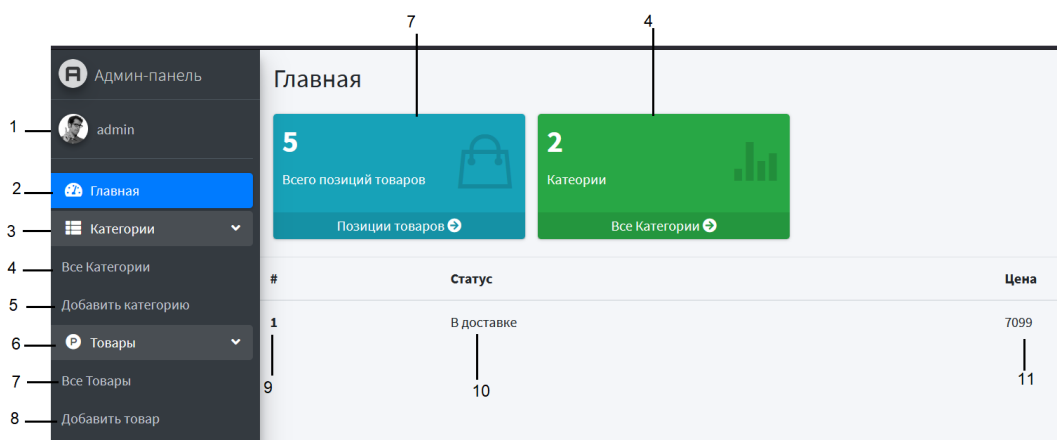


Рисунок 30 – Интерфейс главной страницы панели администратора

Интерфейс страницы каталога товаров включает в себя:

- 1 – переход на главную страницу;
- 2 – так же переход на главную страницу;
- 3 – выпадающий список действий с категориями;
- 4 – переход на страницу просмотра категорий;
- 5 – переход на форму создания категории;
- 6 – выпадающий список действий с товарами;
- 7 – переход на страницу просмотра товаров;
- 8 – переход на форму создания товара;
- 9 – ID заказа;
- 10 – колонка статуса заказа;
- 11 – колонка цен заказов.

При выборе поля «Все категории» администратор будет перенаправлен на страницу просмотра списка категорий. На этой странице он может отредактировать конкретную категорию или же удалить ее. Интерфейс страницы просмотра категорий представлен на рисунке 31.



Рисунок 31 – Интерфейс просмотра всех категорий

Интерфейс страницы каталога товаров включает в себя:

- 1 – описание интерфейса левого меню представлено выше;
- 2 – колонка ID категорий;
- 3 – колонка названий категорий;
- 4 – кнопка редактирования категории;
- 5 – кнопка удаления категории.

Администратор может нажать кнопку редактировать в строке конкретной категории для ее редактирования. Он будет перенаправлен на страницу с формой редактирования категории. Интерфейс страницы редактирования категорий представлен на рисунке 32.

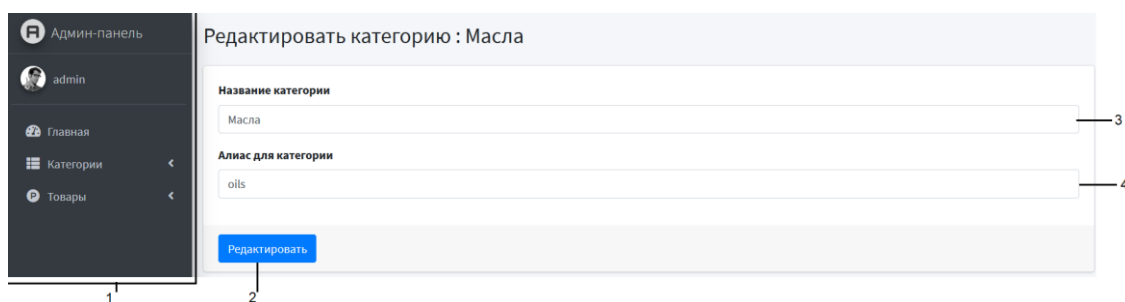


Рисунок 32 – Интерфейс редактирования категории

Интерфейс страницы каталога товаров включает в себя:

- 1 – описание интерфейса левого меню представлено выше;
- 2 – кнопка подтверждения изменений;
- 3 – поле изменения названия;
- 4 – поле изменения алиаса.

Нажав на поле «Добавить категорию», администратор попадет на страницу создания категории. Интерфейс страницы создания категорий представлен на рисунке 33.

Рисунок 33 – Интерфейс добавления категории

Интерфейс страницы каталога товаров включает в себя:

- 1 – описание интерфейса левого меню представлено выше;
- 2 – кнопка подтверждения создания;
- 3 – поле ввода названия;
- 4 – поле ввода алиаса.

При выборе поля «Все товары» администратор будет перенаправлен на страницу просмотра списка товаров. На этой странице он может отредактировать конкретный товар или же удалить его. Интерфейс страницы просмотра товаров представлен на рисунке 34.

ID	Название	Категория	Количество	Продано		
15	Bridgestone Potenza Sport	Шины	3	0	Редактировать	Удалить
14	Bridgestone Turanza T005	Шины	2	0	Редактировать	Удалить
13	Bridgestone Alenza 001	Шины	2	0	Редактировать	Удалить

Рисунок 34 – Интерфейс просмотра товаров

Интерфейс страницы каталога товаров включает в себя:

- 1 – описание интерфейса левого меню представлено выше;

- 2 – колонка ID товара;
- 3 – колонка категорий;
- 4 – колонка количества товара;
- 5 – колонка количества проданного товара;
- 6 – кнопка редактирования товара;
- 7 – кнопка удаления товара;
- 8 – колонка названий товаров.

Администратор может нажать кнопку редактировать в строке конкретного товара для его редактирования. Он будет перенаправлен на страницу с формой редактирования товара. Интерфейс страницы редактирования категорий представлен на рисунке 35.

The screenshot shows the 'Редактировать товар Bridgestone Turanza T005' page in an admin panel. The interface includes a sidebar menu (1) with 'Админ-панель', 'Главная', 'Категории', and 'Товары'. The main form contains the following elements:

- Field 5: 'Название товара' (Product Name) with the value 'Bridgestone Turanza T005'.
- Field 6: 'Алиас для товара' (Product Alias) with the value 'Bridgestone_Turanza_T005'.
- Field 7: 'Количество товара' (Product Quantity) with the value '2'.
- Field 8: 'Описание' (Description) containing text about the Bridgestone Turanza T005 tire.
- Field 9: 'Выберите категорию' (Select category) dropdown menu.
- Field 10: 'Выберите характеристику' (Select characteristic) dropdown menu.
- Field 2: 'Изображения товара' (Product Images) section with a tire image and a 'Загрузить' (Upload) button.
- Field 3: 'Изменить' (Update) button at the bottom left.

Рисунок 35 – Интерфейс редактирования товара

Интерфейс страницы каталога товаров включает в себя:

- 1 – описание интерфейса левого меню представлено выше;
- 2 – кнопка подтверждения изменений;
- 3 – поле изменения изображения;
- 4 – поле изменения категории;
- 5 – поле изменения названия товара;
- 6 – поле изменения алиаса товара;

- 7 – поле изменения количества товара;
- 8 – поле изменения описания товара;
- 9 – поле выбора изменяемой характеристики;
- 10 – поле изменения значения характеристики.

Нажав на поле «Добавить товар», администратор попадет на страницу создания товара. Интерфейс страницы создания товара представлен на рисунке 36.

Рисунок 36 – Интерфейс добавления товара

Интерфейс страницы каталога товаров включает в себя:

- 1 – описание интерфейса левого меню представлено выше;
- 2 – кнопка подтверждения создания;
- 3 – поле загрузки изображения;
- 4 – поле ввода алиаса;
- 5 – поле ввода количества товара;
- 6 – поле ввода описания товара;
- 7 – поле ввода выбора категории товара;
- 8 – поле ввода названия товара.

3 Реализация программного обеспечения информационной системы интернет-магазина

3.1 Диаграмма компонентов информационной системы интернет-магазина

Информационная система интернет-магазина автомобильных запчастей была реализована с помощью фреймворка с открытым исходным кодом Laravel, который позволяет создавать веб приложения с использованием языка PHP. Данная диаграмма представлена на рисунке 37.

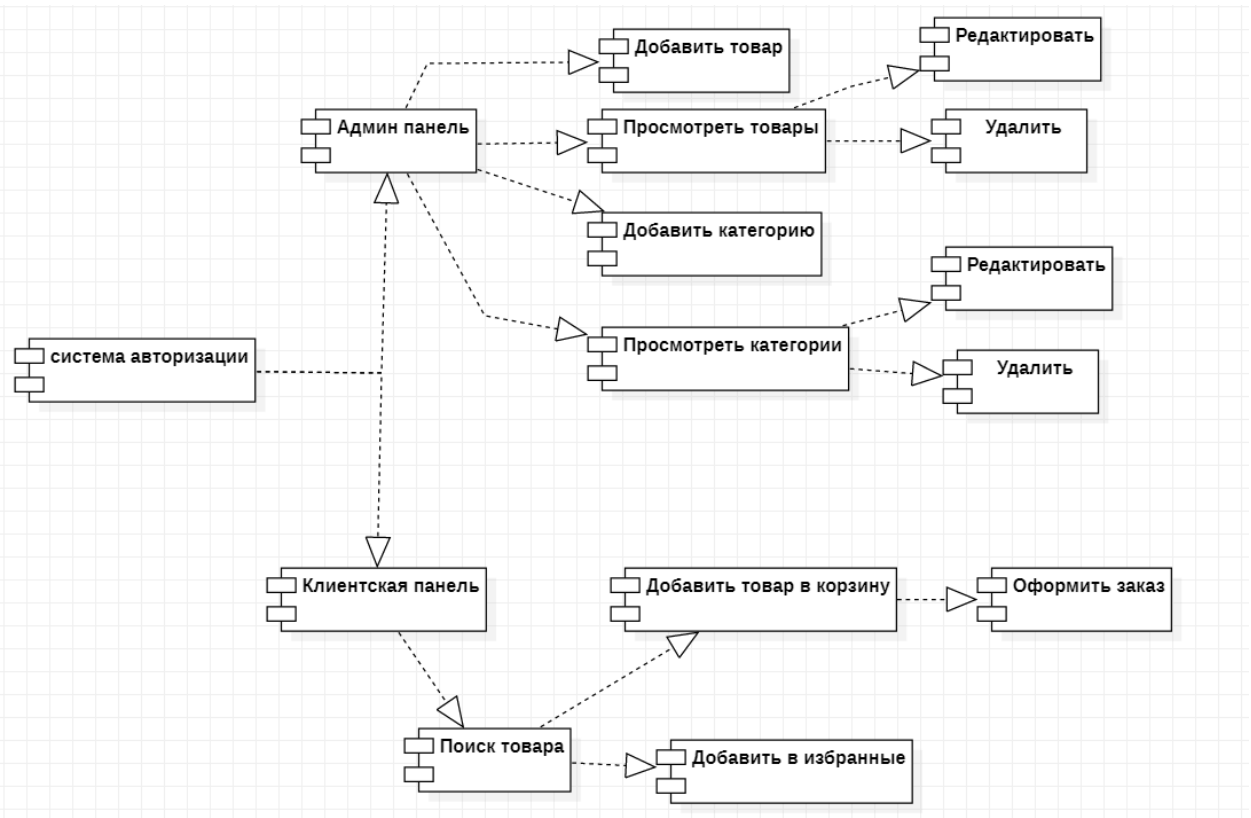


Рисунок 37 – Диаграмма компонентов

Диаграмма компонентов отображает взаимосвязь программных компонентов системы

3.2 Представление развертывания

Диаграмма развертывания показывает топологию и аппаратные средства системы. Данная диаграмма выполняет роль визуализации компонентов

системы и их развертывания на физических узлах. Диаграмма развертывания изображена на рисунке 38.

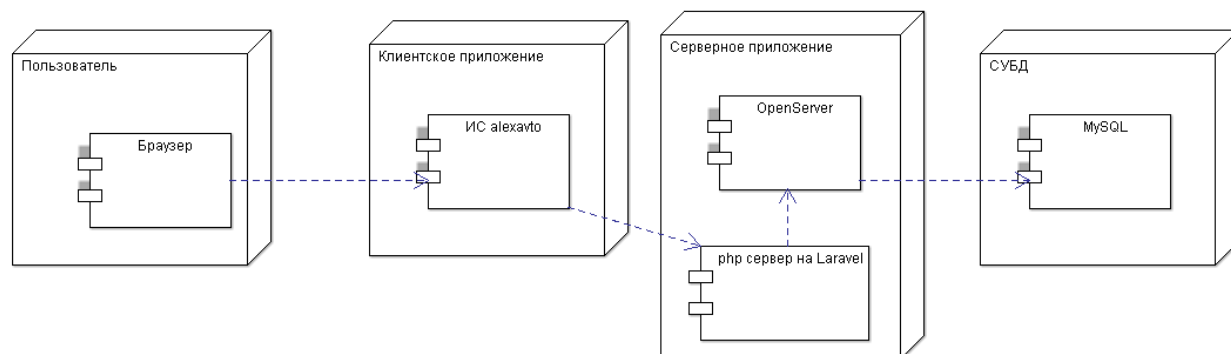


Рисунок 38 – Диаграмма развертывания

Как видно из диаграммы развертывания, информационная система разворачивается в клиентском приложении, которое связано с php сервером на Laravel развернутом при помощи OpenServer. Он в свою очередь обеспечивает подключение к СУБД MySQL.

4 Контроль качества информационной системы интернет-магазина

Программный продукт обладает набором свойств, которые характеризуют его качество и соответствие требованиям пользователей или заказчика. Исходя из этого задача обеспечения качества разработки программного продукта является одной из самых приоритетных в процессе создания программного обеспечения. Эффективным средством в решении данной задачи является тестирование. С технической точки зрения тестирование заключается в выполнении приложения на некотором множестве исходных данных и сверке получаемых результатов с заранее известными с целью установить соответствие различных свойств и характеристик приложения заказанным свойствам. Для более точного анализа программного продукта применяются различные методологии, которые охватывают разные наборы свойств программного обеспечения.

4.1 Функциональное тестирование информационной системы интернет-магазина

Основной целью функционально тестирования является сопоставление функционирования программного обеспечения с начальными требованиями, предъявленных проекту. Другими словами, в ходе тестирования выявляется способность информационной системы решать задачи, которые ставит перед ней пользователь.

Функциональное тестирование рассмотрено на примере прецедента «Регистрация». Согласно этому прецеденту системе, необходимо выполнить регистрацию пользователя, получив от него электронную почту, логин и пароль. Необходимо выделить классы эквивалентности для входных параметров. Классы эквивалентности для параметра электронная почта представлены в таблице 3.

Таблица 3 – Классы эквивалентности

Параметр электронной почты	Правильные классы эквивалентности	Неправильные классы эквивалентности
Длина	[8, 40]	< 8, > 40
Наличие '@' и '.'	Да	Нет
Латинские буквы	[A-Z],[a-z]	[А-Я],[а-я]
Цифры	[0-9]	–
Специальные символы	[-],[_]	Все остальные

На основании классов эквивалентности были сформированы возможные ситуации, связанные с входными параметрами прецедента «Регистрация». Классы эквивалентности представлены в таблице 4. Результаты тестирования приведены на рисунках Д.1-Д.6 в приложении Д.

Таблица 4 – Возможные ситуации

Входящие значения	Предполагаемый результат	Класс эквивалентности	Результат
stalker-999@mail.ru	Успех	1,3,5,7,9	Успех (рисунок Д.1)
s@m.ru	Ошибка	2	Ошибка (рисунок Д.2)
stalker-999mailru	Ошибка	4	Ошибка (рисунок Д.3)
сталкер-999@mail.ru	Ошибка	6	Ошибка (рисунок Д.4)
stalker@mail.ru	Успех	1,3,5,9	Успех (рисунок Д.5)
stalker/@mail.ru	Ошибка	10	Ошибка (рисунок Д.6)

Анализ граничных значений – это испытание программного обеспечения на граничных значениях входных параметров.

В качестве параметра для тестирования было выбрано поле ввода электронной почты. Были выделены следующие границы:

– произвольная граница – значение входного параметра должно быть в пределах [8;40] символов;

- логическая граница означает, что введённое значение должно состоять из латинских букв, цифр, знаков «@» и «.», специальных символов «-», «_»;
- физическая граница означает пустую строку.

Учитывая все перечисленные границы, были составлены схемы, которые демонстрируют их визуально. Произвольная граница схематично представлена на рисунке 39.



Рисунок 39 – Произвольные границы поля ввода электронной почты

Логическая граница представлена на рисунке 40.

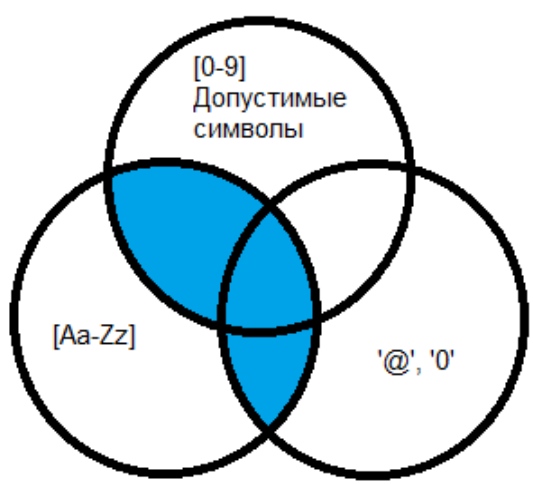


Рисунок 40 – Логическая граница поля ввода электронной почты

В таблице 5 приведен перечень вариантов для ввода в поле электронной почты.

Таблица 5 – Функциональное тестирование граничных условий

Граничное условие	Значение	Результат
Пустая строка	[]	Ошибка

Продолжение таблицы 5

Граничное условие	Значение	Результат
Количество символов < 8	s@m.ru	Ошибка
Количество символов = 8	sta@m.ru	Успех
Количество символов > 40	legashnev-maksim-aleksandrovich-1999@mail.ru	Ошибка
Количество символов = 40	legashnev-maksim-aleksandrovich-@mail.ru	Успех
Допустимые символы	legashnev_max-999@mail.ru	Успех
Недопустимые символы	Legashnev/max1999@mail.ru	Ошибка
Только цифры и латинские буквы	stalker999@mail.ru	Успех
Наличие букв не из латинского алфавита	максим@mail.ru	Ошибка
Отсутствие символов «@» и «.»	stalker-999mailru	Ошибка

В дополнение к методу граничных значений часто используют метод функциональных диаграмм. Он позволяет обнаружить неполноту метода граничных условий. Для использования метода функциональных диаграмм необходимо определить компоненты: причины и следствия. Причиной является входное условие, а следствие – это то, как система среагирует на входное условие, другими словами результат программы.

Для параметра теста поле ввода электронной почты были составлены следующие причины и следствия:

- количество символов не меньше 8 и не больше 40,
- наличие символов «@» и «.»,
- наличие латинских букв,
- наличие специальных допустимых символов,
- наличие цифр,
- успех,
- ошибка.

На рисунке 41 представлена функциональная диаграмма тестируемого параметра поля ввода электронной почты.

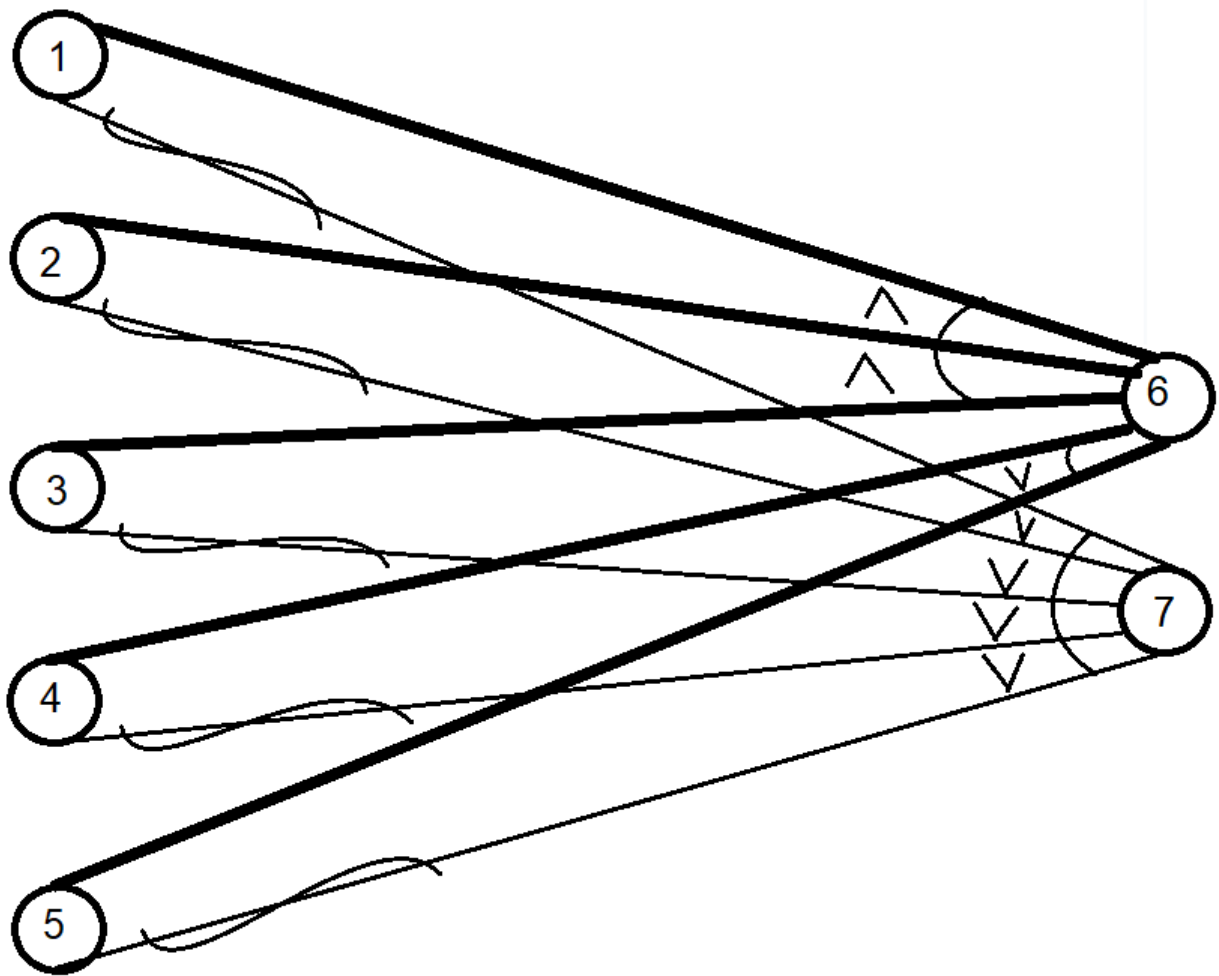


Рисунок 41 – Функциональная диаграмма поля ввода электронной почты

Результат совмещения методов функционального тестирования граничных условий и функциональных диаграмм представлен в таблице 6. Столбцы в данной таблице являются функциональными тестами. Для обозначения истинного значения будет использоваться 1, а для ложного 0.

Таблица 6 – Таблица решений

	1	2	3	4	5	6	7	8	9	10
1	0	0	1	0	1	1	-	1	-	-
2	-	-	1	-	1	1	-	1	-	0
3	-	-	1	-	1	1	-	1	0	-
4	-	-	0	-	1	1	0	0	-	-
5	-	-	0	-	0	1	-	1	-	-
6	0	0	1	0	1	1	0	1	0	0
7	1	1	0	1	0	0	1	0	1	1

Данная таблица подтверждает, что были протестированы все входные значения для поля ввода электронной почты.

4.2 Попарное тестирование информационной системе интернет-магазина

Для покрытия функциональными тестами всего прецедента регистрации было применено попарное тестирование. На этапе регистрации пользователю необходимо ввести имя (логин), электронную почту, а также пароль. Таким образом перед разработчиком стоит задача тестирования всех входных параметров. Для данной цели попарное тестирование подходит лучше всего, т.к. позволяет быстро сгенерировать все возможные комбинации для входных параметров.

Для реализации попарного тестирования было использовано консольное программное обеспечение «allpairs». На рисунке 42 представлены входные классы эквивалентности для параметров необходимых при регистрации пользователя.

name	E-mail	password
validName	validMail	validPass
notLatinName	not"@&&".	invalidPassSym
tooLongName	notLatinMail	shortPass
nullName	notSpecSym	LongPass
NULL	notNum	NULL

Рисунок 42 – Входные классы эквивалентности

На рисунке 43 представлены результаты работы программы «allpairs», где присутствуют все возможные тестовые случаи для регистрации.

TEST CASES				
case	name	E-mail	password	pairings
1	validName	validMail	validPass	3
2	validName	not"@&&".	invalidPassSym	3
3	validName	notLatinMail	shortPass	3
4	validName	notSpecSym	LongPass	3
5	validName	notNum	NULL	3
6	notLatinName	validMail	invalidPassSym	3
7	notLatinName	not"@&&".	validPass	3
8	notLatinName	notLatinMail	LongPass	3
9	notLatinName	notSpecSym	shortPass	3
10	notLatinName	notNum	validPass	2
11	tooLongName	validMail	shortPass	3
12	tooLongName	not"@&&".	LongPass	3
13	tooLongName	notLatinMail	validPass	3
14	tooLongName	notSpecSym	invalidPassSym	3
15	tooLongName	notNum	invalidPassSym	2
16	nullName	validMail	LongPass	3
17	nullName	not"@&&".	shortPass	3
18	nullName	notLatinMail	invalidPassSym	3
19	nullName	notSpecSym	validPass	3
20	nullName	notNum	shortPass	2
21	NULL	validMail	NULL	3
22	NULL	not"@&&".	validPass	2
23	NULL	notLatinMail	invalidPassSym	2
24	NULL	notSpecSym	shortPass	2
25	NULL	notNum	LongPass	3
26	notLatinName	not"@&&".	NULL	2
27	tooLongName	notLatinMail	NULL	2
28	nullName	notSpecSym	NULL	2

Рисунок 43 – Все возможные тестовые случаи для регистрации

Функциональное тестирование прецедента регистрации пользователя было завершено без ошибок. Все исключительные ситуации обрабатываются.

4.3 Метрики кода

Метрики кода являются набором критериев, которые позволяют оценить качество разработанного программного продукта с точки зрения написанного кода. Так же с их помощью можно отследить размер трудовых ресурсов, затраченных на разработку системы. Кроме того, метрики позволяют разработчику тщательнее изучить написанный код для его доработки или рефакторинга.

Основным критерием оценки проекта является его цикломатическая сложность. Он основывается на графе управляющей логики. Узлы данного

графа – это неделимые блоки команд программы. Узлы соединяют ориентированные ребра.

Для оценки проекта была использована утилита «phploc», которая генерирует статистику кода. На рисунке 44 приведены характеристики информационной системы управления продажами и доставкой в интернет магазине автозапчастей.

Directories	9
Files	52
Size	
Lines of Code (LOC)	1694
Comment Lines of Code (CLOC)	451 (26.62%)
Non-Comment Lines of Code (NCLOC)	1243 (73.38%)
Logical Lines of Code (LLOC)	382 (22.55%)
Classes	195 (51.05%)
Average Class Length	3
Minimum Class Length	0
Maximum Class Length	44
Average Method Length	1
Minimum Method Length	0
Maximum Method Length	16
Functions	0 (0.00%)
Average Function Length	0
Not in classes or functions	187 (48.95%)
Cyclomatic Complexity	
Average Complexity per LLOC	0.03
Average Complexity per Class	1.25
Minimum Class Complexity	1.00
Maximum Class Complexity	4.00
Average Complexity per Method	1.10
Minimum Method Complexity	1.00
Maximum Method Complexity	4.00
Dependencies	
Global Accesses	0
Global Constants	0 (0.00%)
Global Variables	0 (0.00%)
Super-Global Variables	0 (0.00%)
Attribute Accesses	60
Non-Static	60 (100.00%)
Static	0 (0.00%)
Method Calls	146
Non-Static	108 (73.97%)
Static	38 (26.03%)

Рисунок 44 – Статистика кода

На рисунке 45 представлены характеристики структуры проекта.

Structure	
Namespaces	11
Interfaces	0
Traits	1
Classes	51
Abstract Classes	0 (0.00%)
Concrete Classes	51 (100.00%)
Methods	75
Scope	
Non-Static Methods	75 (100.00%)
Static Methods	0 (0.00%)
Visibility	
Public Methods	67 (89.33%)
Non-Public Methods	8 (10.67%)
Functions	6
Named Functions	0 (0.00%)
Anonymous Functions	6 (100.00%)
Constants	1
Global Constants	0 (0.00%)
Class Constants	1 (100.00%)

Рисунок 45 – Характеристики структуры проекта

Цикломатическая сложность проекта характеризуется количеством независимых маршрутов. Для оценки данного параметра нужно обратить внимание на раздел «Cyclomatic complexity» на рисунке 44. Видно, что максимальная сложность методов и классов равна четырем. Это говорит о том, что проект не слишком сложен с точки зрения логического ветвления кода.

4.4 Планирование и оценка бюджета проекта информационной системы интернет-магазина

Неотъемлемой частью разработки любого проекта является его планирование. Оно проводится для того, чтобы учесть всевозможные риски, связанные с разработкой. Так же планирование позволяет рассчитать объём затрат, как материальных, так и ресурсных.

При планировании проектов есть несколько основных групп планирования:

- планирование проектных задач; планирование расписания проекта;
- планирование команды проекта;
- управление рисками.

В таблице 7 представлено описание затрачиваемых ресурсов.

Таблица 7 – Список ресурсов проекта

Название ресурса	Тип	Стандартная ставка	Ставка сверхурочных
Программист1	Трудовой	150 Р/ч	200 Р/ч
Программист2	Трудовой	250 Р/ч	300 Р/ч
Дизайнер	Трудовой	200 Р/ч	250 Р/ч
Интернет	Затраты		
Ноутбук	Материальный	500 Р	

Распределение ресурсов по задачам представлено в таблице 8.

Таблица 8 – Назначение ресурсов

№	Название задачи	Ресурс	Единицы
1	Начало проекта		
2	Анализ требований		
3	Анализ аналогичных решений	Программист1 Ноутбук Интернет	100% 1 шт
4	Анализ требований покупателей для интернет-магазина	Программист1 Программист2 Ноутбук	80% 20% 1 шт
5	Консультация с владельцем магазина автозапчастей	Программист1 Ноутбук	50% 1 шт
6	Принятие окончательного решения	Программист1 Программист2	100% 100%
7	Анализ требований завершен		
8	Выбор средств реализации проекта		
9	Анализ возможных программных средств	Программист1 Ноутбук Интернет	100% 1 шт
10	Выбор программного средства для реализации	Программист1 Ноутбук Интернет	100% 1 шт
11	Выбор средств завершен		
12	Реализация проекта		
13	Разработка структуры данных	Программист1 Программист2 Ноутбук Интернет	75% 25% 1 шт
14	Разработка пользовательского интерфейса	Программист1 Программист2 Дизайнер Ноутбук Интернет	50% 25% 50% 2 шт

Продолжение таблицы 8

№	Название задачи	Ресурс	Единицы
15	Разработка модулей обработки данных		
№	Название задачи	Ресурс	Единицы
16	Программирование сопутствующих и обслуживающих классов	Программист1 Ноутбук Интернет	50% 1 шт
17	Программирование логики обслуживания покупателя и рабочего процесса администратора	Программист2 Ноутбук Интернет	100% 1 шт
18	Реализация проекта завершена		
19	Отладка		
20	Тестирование пользовательского интерфейса	Программист1 Программист2 Дизайнер Ноутбук Интернет	25% 25% 25% 3 шт
21	Тестирование логики обслуживания покупателя и рабочего процесса администратора	Программист1 Программист2 Ноутбук Интернет	25% 75% 2 шт
22	Исправление ошибок	Программист1 Компьютер Интернет	75% 1 шт
23	Тестирование завершено		
24	Конец проекта		

Из представленных таблиц видно, что реализацией проекта будут заниматься два программиста и один дизайнер, которым для работы нужны компьютеры и доступ к интернету.

Задача календарного планирования является эффективным инструментом управления проектами фирмы. Календарный план даёт четкое представление об этапах производства, длительности этапов, количестве и видах привлекаемых ресурсов на каждом этапе. То есть, весь процесс производства делится на конкретные виды работ, определяется, какие ресурсы и в каком количестве требуются для каждого этапа, затем создается оптимальный план по времени выполнения и затратам.

Проведем необходимые вычисления с использованием выбранного средства управления программными проектами Microsoft Project 2007. На рисунке 46 представлен список задач из таблицы 1 внесенный в Microsoft Project. Началом проекта была принята дата 14.04.2021.

Название задачи	Длительность	Начало	Окончание	Предшест	Названия ресурсов
Начало проекта	0 дней	Ср 14.04.21	Ср 14.04.21		
▣ Анализ требований	9 дней	Ср 14.04.21	Пн 26.04.21		
Анализ аналогичных решений	3 дней	Ср 14.04.21	Пт 16.04.21	1	Программист1;Ноутбук[2 Единиц];Интернет
Анализ требований покупателей для интернет-магазина	3 дней	Сб 17.04.21	Ср 21.04.21	3	Программист1;Ноутбук[1 Единиц]
Консультация с владельцем магазина автозапчастей	2 дней	Чт 22.04.21	Пт 23.04.21	4	Программист1[50%];Ноутбук[1 Единиц]
Принятие окончательного решения	1 день	Пн 26.04.21	Пн 26.04.21	5	Программист1;Программист2
Анализ требований завершен	0 дней	Пн 26.04.21	Пн 26.04.21	6	
▣ Выбор средств реализации проекта	6 дней	Вт 27.04.21	Вт 04.05.21		
Анализ возможных программных средств	3 дней	Вт 27.04.21	Пт 30.04.21	1	Ноутбук[1 Единиц];Интернет;Программист2
Выбор программного средства для реализации	1 день	Пн 03.05.21	Вт 04.05.21	9	Ноутбук[1 Единиц];Интернет;Программист2
Выбор средств завершен	0 дней	Вт 04.05.21	Вт 04.05.21	10	
▣ Реализация проекта	18 дней	Вт 04.05.21	Чт 27.05.21		
Разработка структуры данных	3 дней	Вт 04.05.21	Чт 06.05.21	7;11	Интернет;Программист1[75%];Программист2[25]
Разработка пользовательского интерфейса	3 дней	Пт 07.05.21	Вт 11.05.21	7;11;13	Программист1;Программист2;Дизайнер;Ноутбук[Единиц];Интернет
▣ Разработка модулей обработки данных	12 дней	Ср 12.05.21	Чт 27.05.21		
Программирование сопутствующих и обслуживающих	11 дней	Ср 12.05.21	Ср 26.05.21	14	Программист1;Ноутбук[1 Единиц];Интернет
Программирование логики обслуживания покупателей	12 дней	Ср 12.05.21	Чт 27.05.21	14	Ноутбук[1 Единиц];Интернет;Программист2
Реализация проекта завершена	0 дней	Чт 27.05.21	Чт 27.05.21	14;17;16	
▣ Отладка	12 дней	Пт 28.05.21	Пн 14.06.21		
Тестирование пользовательского интерфейса	7 дней	Пт 28.05.21	Пн 07.06.21	18	Программист1[25%];Программист2[25%];Дизайн
Тестирование логики обслуживания покупателя и рабочего процесса администратора	7 дней	Пт 28.05.21	Пн 07.06.21	18	Программист1[25%];Программист2[75%];Ноутбук[Единиц];Интернет
Исправление ошибок	5 дней	Вт 08.06.21	Пн 14.06.21	20;21	Программист1[75%];Ноутбук[1 Единиц];Интернет
Тестирование завершено	0 дней	Пн 14.06.21	Пн 14.06.21	22	
Конец проекта	0 дней	Пн 14.06.21	Пн 14.06.21	23	

Рисунок 46 – Задачи проекта в среде Microsoft Project

На рисунке 47 представлена диаграмма Ганта, построенная в среде Microsoft Project на основе введенных данных.

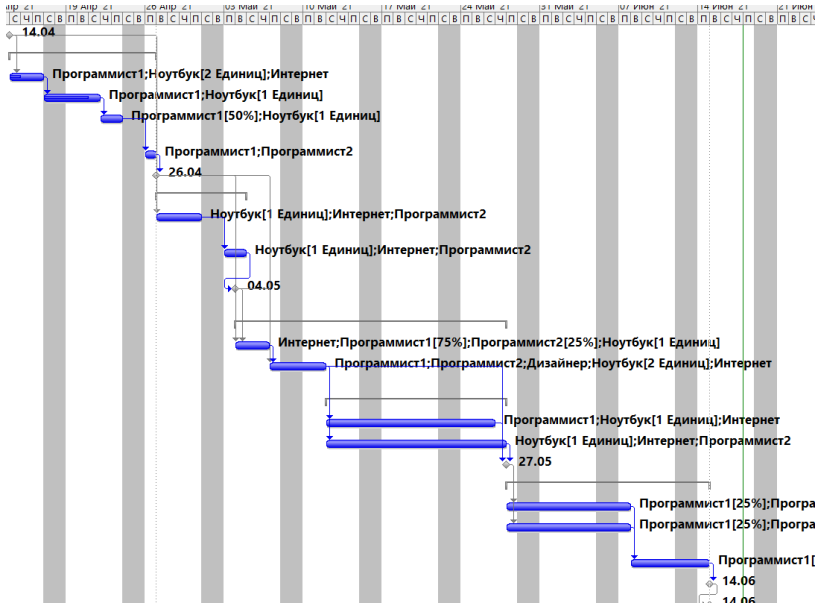


Рисунок 47 – Диаграмма Ганта

На рисунке 48 представлен обзор проекта.

Планирование календарное2007
SPECIALIST RePack

на дату Пт 18.06.21

<u>Даты</u>			
Начало:	Ср 14.04.21	Окончание:	Пн 14.06.21
Баз. начало:	НД	Баз. окончание:	НД
Факт. начало:	Ср 14.04.21	Факт. окончание:	НД
Отклон. начала:	0 дней	Отклон. окончания:	0 дней
<u>Длительность</u>			
Текущий план:	44 дней	Оставшаяся:	41,84 дней
Базовый план:	0 дней?	Фактическая:	2,16 дней
Отклонение:	44 дней	Процент завершени	5%
<u>Трудозатраты</u>			
Текущий план:	500,8 ч	Оставшиеся:	480 ч
Базовый план:	0 ч	Фактические:	20,8 ч
Отклонение:	500,8 ч	Процент завершени	4%
<u>Затраты</u>			
Текущий план:	107 720,00 Р	Оставшиеся:	103 300,00 Р
Базовый план:	0,00 Р	Фактические:	4 420,00 Р
Отклонение:	107 720,00 Р		
<u>Состояние задачи</u>		<u>Состояние ресурса</u>	
Еще не начатые задачи:	17	Трудовые ресурсы:	3
Выполняющиеся задачи:	5	Трудовые ресурсы с прев. дост	1
Завершенные задачи:	2	Материальные ресурсы:	1
Итого задач:	24	Итого ресурсов:	5

Рисунок 48 – Обзорный отчет проекта

Из рисунка видно, что общая длительность проекта составляет 44 дня,
а предполагаемые затраты – 107720,00 руб.

Заключение

В рамках данной выпускной квалификационной работы разработана информационная система управления продажами и доставкой в интернет-магазине автозапчастей. Данная система позволяет пользователю оформить доставку в любой город. Кроме того, подбор товара в каталоге не затруднителен для покупателя. Клиентская часть разработана на языке программирования PHP помощью фреймворка с открытым исходным кодом Laravel. Серверная часть реализована на языке SQL в среде MySQL при помощи Open Server. Проведено функциональное и структурное тестирование системы. Также проведён качественный анализ метрик кода. Результаты тестирования показали, что разработанное ПО работает корректно и соответствует ожидаемым требованиям. В ходе выполнения работы произведено планирование и оценка бюджета программных средств в среде Microsoft Office Project 2007. На основании определенных задач и ресурсов проекта построена диаграмма Ганта, которая показывает календарный график выполнения работ по проекту. Срок выполнения проекта составил 44 дня, а бюджет – 107720,00 рублей. Анализ сроков и бюджета свидетельствует о том, что разработка данного проекта является экономически выгодной. Таким образом, в результате выполнения выпускной квалификационной работы разработана информационная система управления продажами и доставкой в интернет-магазине автозапчастей, отвечающая всем требованиям технического задания.

Список использованных источников

1. Материалы с сайта агентства экономической информации «Прайм». [Электронный ресурс]. – URL: https://1prime.ru/state_regulation/20210218/833073169.html#:~:text=P%20PsP±C%0PμPj%20PsP±CЉPμPjPμ%20PіCЉPsPrP°P¶%20CЉPsP·PSPёC‡PSPsPiPs,2019%20PiPsPrCѓ%20PrPsP»CЦ%20CЃPsCЃC,P°PIP»CЦP»P°%2029%25 (дата обращения: 24.04.2021).
2. Материалы с сайта аналитического агентства «Автостат». [Электронный ресурс]. – URL: <https://www.autostat.ru/infographics/48022/> (дата обращения: 24.04.2021).
3. Материалы с сайта автомобильных новостей «Autonews». [Электронный ресурс]. – URL: <https://www.autonews.ru/news/5825aff59a794747431327f3#:~:text=P%20C...PsPrPμ%20PёCЃCЃP»PμPrPsPIP°PSPёCЦ%20P±C<P»Ps%20CѓCЃC,P°PSPsPIP»PμPSPs,PSP°C...PsPrPёC,CЃCЦ%201%20P°PIC,PsPjPsP±PёP»CЃ%20вЉ“%2034%25> (дата обращения: 24.04.2021).
4. Материал с сайта языка программирования php [Электронный ресурс]. – URL: <https://www.php.net/manual/ru/intro-whatcando.php> (дата обращения: 24.05.2021).
5. Материал с сайта «studme.org» [Электронный ресурс]. – URL: https://studme.org/87184/ekonomika/metodologiya_idef0 (дата обращения: 24.05.2021).
6. Материал с сайта «cfin.ru» [Электронный ресурс]. – URL: <https://www.cfin.ru/vernikov/idef/idef3.shtml> (дата обращения: 24.05.2021).
7. Киммел, П. UML. Основы визуального анализа и проектирования / П. Киммел. – М.: НТ Пресс, 2008. – 272 с.

Приложение А
ГЛОССАРИЙ
(справочное)

Программное обеспечение (ПО) – совокупность всех программ и соответствующей документации, обеспечивающая использование ЭВМ в интересах каждого ее пользователя.

База данных (БД) – организованная в соответствии с определёнными правилами и поддерживаемая в памяти компьютера совокупность данных, характеризующая актуальное состояние некоторой предметной области и используемая для удовлетворения информационных потребностей пользователей.

Информационная система (ИС) – взаимосвязанная совокупность средств, методов и персонала, используемых для хранения, обработки и выдачи информации в интересах достижения поставленной цели.

Система управления базами данных (СУБД) – совокупность языковых и программных средств общего или специального назначения, обеспечивающих управление созданием и использованием баз данных одним или несколькими пользователями.

Unified Modeling Language (UML) – графический унифицированный язык моделирования, использующий объектно-ориентированный подход для моделирования в сфере разработки программного обеспечения.

Объектно-ориентированное программирование (ООП) – это методология программирования, основанная на представлении программы в виде совокупности объектов, каждый из которых является экземпляром определенного класса, а классы образуют иерархию наследования.

Приложение Б
ЛИСТИНГ КЛИЕНТСКОГО ПРИЛОЖЕНИЯ НА ЯЗЫКЕ PHP
(рекомендуемое)

Листинг Б.1 – Файл Admin/AttributeController.php

```
<?php

namespace App\Http\Controllers\admin;

use App\Models\Attribute_set;

use App\Models\Product;

use Illuminate\Http\Request;

use App\Models\Attribute_set_value;

use App\Http\Controllers\Controller;

//класс контроллера страницы характеристик товаров
class AttributeController extends Controller
{
    public function index(Product $prod)
    {
        return view('properties', [
            'goods' => $good
        ]);
    }

    public function add(Request $request, Product $good)
    {
        $attribute = new Attribute_set();
        $attribute->attribute_name = $request->name;
        $attribute->Products_id = $prod->id;
        $attribute->value = $request->value;

        $attribute->save();

        return redirect()->back();
    }
}
```

Листинг Б.2 – Файл Admin/CategoryController.php

```
<?php

namespace App\Http\Controllers\Admin;

use Illuminate\Http\Request;

use App\Http\Controllers\Controller;

use App\Models\Category;

//класс контроллера категорий
class CategoryController extends Controller
{
    public function main()
    {
        $categorii = Category::getAll();

        return view('admin.category.index', [
            'categorii' => $ categorii
        ]);
    }
    public function create()
    {

```

```

        return view('admin.category.create');
    }
    public function store(Request $request)
    {
        $new_category = new Category();
        $new_category->name = $request->name;
        $new_category->alias = $request->alias;
        $new_category->save();

        return redirect()->back()->withSuccess('Категория успешно
добавлена!');
    }
    public function edit(Category $category)
    {
        return view('admin.category.edit', [
            'category' => $category
        ]);
    }
    public function update(Request $request, Category $category)
    {
        $category->name = $request->name;
        $category->alias = $request->alias;

        $category->save();

        return redirect()->back()->withSuccess('Категория успешно
обновлена!');
    }
    public function destroy(Category $category)
    {
        $category->delete();

        return redirect()->back()->withSuccess('Категория успешно удалена!');
    }
}

```

Листинг Б.3 – Файл Admin/HomeController.php

```

<?php

namespace App\Http\Controllers\Admin;

use App\Models\Product;

use App\Http\Controllers\Controller;

use Illuminate\Http\Request;

use App\Models\Category;

//класс контроллера основной страницы
class HomeController extends Controller
{
    public function index()
    {
        $categories_count = Category::getall()->count();
        $products_count = Product::allget()->count();

        return view('admin.home.index', [
            'categories_count' => $categories_count,
            'products_count' => $products_count
        ]);
    }
}

```

Приложение В
ЛИСТИНГ СОЗДАНИЯ ТАБЛИЦ В БД
НА ЯЗЫКЕ PHP
(рекомендуемое)

Листинг В.1 – Создание таблицы User

```
<?php

use Illuminate\Support\Facades\Schema;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Database\Migrations\Migration;

class CreateUsersTable extends Migration
{
    public function up()
    {
        Schema::create('Users', function (Blueprint $tab) {

            $ tab ->timestamp('email_verified_at')->nullable();
            $ tab -> str ('password');
            $ tab ->rememberToken();
            $ tab ->id();
            $ tab ->str ('surname');
            $ tab -> str ('name');
            $ tab -> str ('lastname')->nullable();
            $ tab -> str ('email')->unique();

            $ tab ->bigInteger('Addresses_id')->unsigned();
            $ tab ->foreign('Addresses_id')
                ->references('id')
                ->on('Addresses')
                ->onDelete('cascade');
            $table->timestamps();

        });
    }
}
```

Листинг В.2 – Создание таблицы Products

```
<?php

use Illuminate\Support\Facades\Schema;

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

class CreateProductsTable extends Migration
{
    public function up()
    {
        Schema::create('Products', function (Blueprint $ tab) {

            $ tab ->string('alias');
            $ tab ->boolean('in_stock');
            $ tab ->integer('quantity');
            $ tab ->id();
            $ tab ->string('name');

            $ tab ->bigInteger('Categories_id')->unsigned();
            $ tab ->foreign('Categories_id')
                ->references('id')
                ->on('Categories')
                ->onDelete('cascade');
            $ tab ->bigInteger('Price_histories_id')->unsigned();
            $ tab ->foreign('Price_histories_id')
                ->references('id')
```

```

        ->on('Price_histories')
        ->onDelete('cascade');
    $ tab ->timestamps();
    });
}
public function down()
{
    Schema::dropIfExists('Products');
}
}

```

Листинг В.3 – Создание таблицы 'Categories'

```

<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Support\Facades\Schema;

use Illuminate\Database\Schema\Blueprint;

class CreateCategoriesTable extends Migration
{
    public function up()
    {
        Schema::create('Categories', function (Blueprint $tab) {
            $ tab ->string('alias');

            $ tab ->id();
            $ tab ->string('name');
            $ tab ->timestamps();
        });
    }
}

```

Листинг В.4 – Создание таблицы Images

```

<?php

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Database\Migrations\Migration;

use Illuminate\Support\Facades\Schema;

class CreateImagesTable extends Migration
{
    public function up()
    {
        Schema::create('Images', function (Blueprint $ tab) {

            $ tab ->boolean('is_main')->nullable();
            $ tab ->bigInteger('Products_id')->unsigned();
            $ tab ->id();
            $ tab ->string('img')->nullable();
            $ tab ->foreign('Products_id')
                ->references('id')
                ->on('Products')
                ->onDelete('cascade');
        });
    }
}

```

```

        $ tab ->timestamps();
    });
}
}

```

Листинг В.5 – Создание таблицы Price_histories

```

<?php

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Database\Migrations\Migration;

use Illuminate\Support\Facades\Schema;

class CreatePriceHistoriesTable extends Migration
{
    public function up()
    {
        Schema::create('Price_histories', function (Blueprint $ tab) {
            $ tab ->id();
            $ tab ->decimal('price'
            $ tab ->timestamps();
        });
    }
}

```

Листинг В.6 – Создание таблицы Wish_lists

```

<?php

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

use Illuminate\Database\Migrations\Migration;

class CreateWishListsTable extends Migration
{
    public function up()
    {
        Schema::create('Wish_lists', function (Blueprint $ tab) {
            $ tab ->foreign('Users_id')
                ->references('id')
                ->on('Users')
                ->onDelete('cascade');
            $table->timestamps();

            $ tab ->id();
            $ tab ->bigInteger('Users_id')->unsigned();
        });
    }
};

```

Листинг В.7 – Создание таблицы Wish_lists

```

<?php

use Illuminate\Database\Migrations\Migration;

```

```

use Illuminate\Support\Facades\Schema;

use Illuminate\Support\Facades\Schema;

use Illuminate\Database\Schema\Blueprint;

class CreateCartsTable extends Migration
{
    public function up()
    {
        Schema::create('Carts', function (Blueprint $ tab) {
            $ tab ->foreign('Users_id')
                ->references('id')
                ->on('Users')
                ->onDelete('cascade');
            $ tab ->timestamps();

            $ tab ->id();
            $ tab ->decimal('total_price')->nullable();
            $ tab ->bigInteger('Users_id')->unsigned();

        });
    }
}

```

Приложение Г
ЛИСТИНГ КЛИЕНТСКОГО ПРИЛОЖЕНИЯ НА ЯЗЫКЕ JAVA SCRIPT
(рекомендуемое)

Листинг Г.1 – Файл admin.js

```

$(document).ready(function () {
    $(".nav-treeview .nav-link, .nav-link").each(function () {
        var location2 = window.location.protocol + '//' + window.location.host
+ window.location.pathname;
        var link = this.href;
        if(link == location2){
            $(this).addClass('active');
            $(this).parent().parent().parent().addClass('menu-is-opening menu-
open');
        }
    });

    $('.delete-btn').click(function () {
        var res = confirm('Подтвердите действия');
        if(!res){
            return false;
        }
    });
})

tinymce.init({
    selector: 'editor',
    plugins: 'allychecker advcode casechange formatpainter linkchecker autolink
lists checklist media mediaembed pageembed permanentpen powerpaste table
advtable tinycomments tinymce_spellchecker',
    toolbar: 'allycheck addcomment showcomments casechange checklist code
formatpainter pageembed permanentpen table',
    toolbar_mode: 'floating',
    tinycomments_mode: 'embedded',
    tinycomments_author: 'Author name',
});

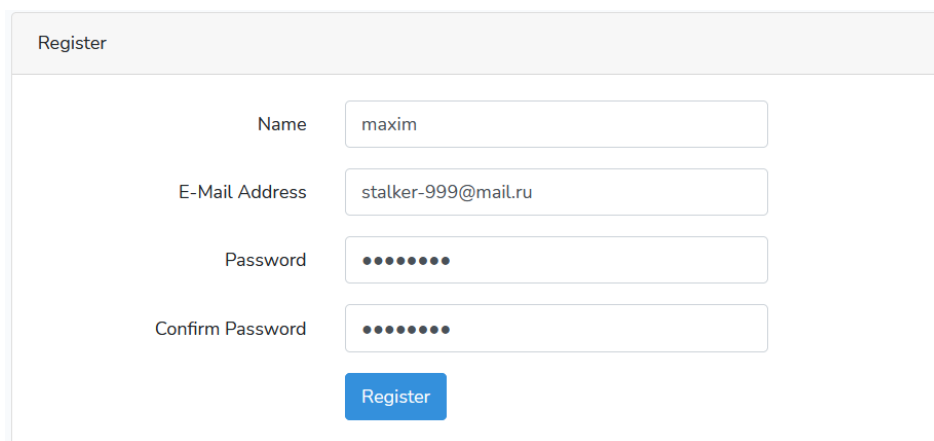
function elFinderBrowser (callback, value, meta) {
    tinymce.activeEditor.windowManager.openUrl({
        onMessage: function (dialogApi, details) {
            if (details.mceAction === 'fileSelected') {
                const file = details.data.file;

                // Make file info
                const info = file.name;
                if (meta.filetype === 'media') {
                    callback(file.url);
                }

                dialogApi.close();
            }
        }
    });
}

```

Приложение Д
РЕЗУЛЬТАТЫ ФУНКЦИОНАЛЬНОГО ТЕСТИРОВАНИЯ
(рекомендуемое)

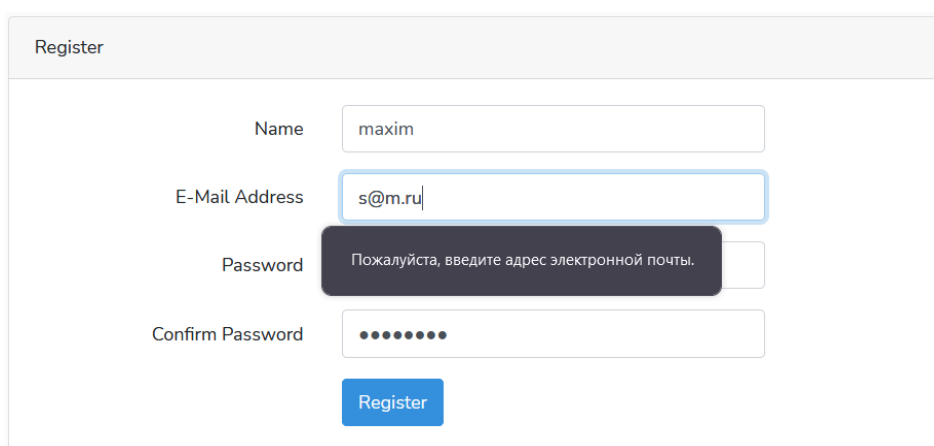


The screenshot shows a 'Register' form with the following fields and values:

- Name: maxim
- E-Mail Address: stalker-999@mail.ru
- Password: masked with 8 dots
- Confirm Password: masked with 8 dots

A blue 'Register' button is located at the bottom of the form.

Рисунок Д.1 – Ввод в поле электронной почты допустимых символов



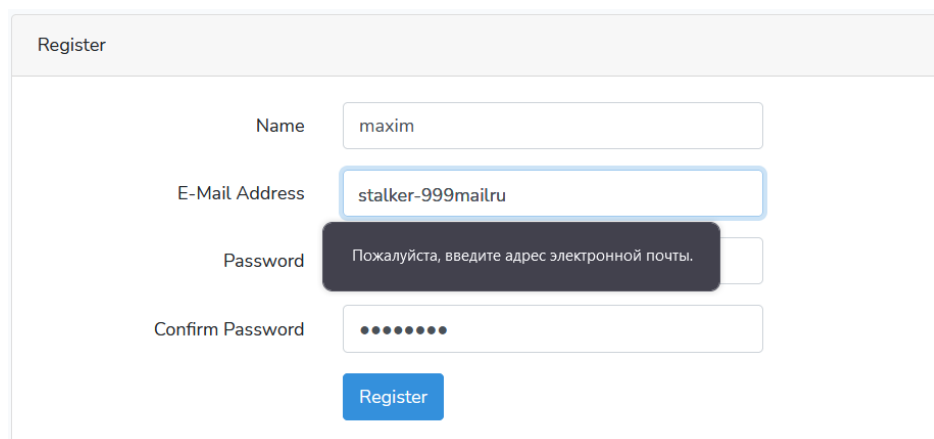
The screenshot shows the 'Register' form with the following fields and values:

- Name: maxim
- E-Mail Address: s@m.ru
- Password: masked with 8 dots
- Confirm Password: masked with 8 dots

A dark grey error message box is displayed over the Password field, containing the text: "Пожалуйста, введите адрес электронной почты." (Please enter an email address).

A blue 'Register' button is located at the bottom of the form.

Рисунок Д.2 – Ввод слишком короткого значения



The screenshot shows the 'Register' form with the following fields and values:

- Name: maxim
- E-Mail Address: stalker-999mailru
- Password: masked with 8 dots
- Confirm Password: masked with 8 dots

A dark grey error message box is displayed over the Password field, containing the text: "Пожалуйста, введите адрес электронной почты." (Please enter an email address).

A blue 'Register' button is located at the bottom of the form.

Рисунок Д.3 – Ввод в поле почты без символов «@», «.»

Register

Name

E-Mail Address

Password

Confirm Password

Рисунок Д.4 – Ввод в поле символов кириллицы

Register

Name

E-Mail Address

Password

Confirm Password

Рисунок Д.5 – Ввод в поле электронной почты допустимых символов, но без чисел

Register

Name

E-Mail Address

Password

Confirm Password

Рисунок Д.6 – Ввод в поле электронной почты недопустимых символов

Приложение Е
ЛИСТИНГ КЛИЕНТСКОГО ПРИЛОЖЕНИЯ НА ЯЗЫКЕ HTML
(рекомендуемое)


```

        </div><!-- /.container-fluid -->
    </div>
    <!-- /.content-header -->

    <!-- Main content -->
    <section class="content">
        <div class="container-fluid">

            <div class="row">
                <div class="col-lg-12">
                    <div class="card card-primary">
                        <!-- form start -->
                        <form          action="{{          route('category.update',
$category['id']) }}" method="POST">

                            <div class="card-footer">
                                <button          type="submit"          class="btn          btn-
primary">Редактировать</button>
                            </div>
                        </form>
                    </div>
                </div>
            </div>

        </div><!-- /.container-fluid -->
    </section>
    <!-- /.content -->

@endsection

```

Листинг Е.3 – Файл views/admin/category/index.blade.php

```

@extends('layouts.admin_layout')

@section('title', 'Категории')

@section('content')

    <!-- Content Header (Page header) -->
    <div class="content-header">
        <div class="container-fluid">
            <div class="row mb-2">
                <div class="col-sm-6">
                    <h1 class="m-0">Все категории</h1>
                </div><!-- /.col -->
            </div><!-- /.row -->
        </div>
    </div><!-- /.container-fluid -->
    </div>
    <!-- /.content-header -->

    <!-- Main content -->
    <section class="content">
        <div class="container-fluid">

            <div class="row">
                <div class="col-lg-12">
                    <div class="card">
                        <div class="card-body p-0">
                            <table class="table table-striped projects">
                                <thead>
                                    <tr>

                                        </td>

```

```

        <td>
            {{ $category['name'] }}
        </td>
        <td class="project-actions text-right">
            <a class="btn btn-info btn-sm"
href="{{ route('category.edit', $category['id']) }}">
                <i class="fas fa-pencil-alt">
                </i>
                Редактировать
            </a>
            <form
action="{{ route('category.destroy', $category['id']) }}"
method="POST" style="display:
inline-block">
                @csrf
                @method('DELETE')
                <button type="submit"
class="btn btn-danger btn-sm delete-btn">
                    <i class="fas fa-trash">
                    </i>
                    Удалить
                </button>
            </form>
        </td>
    </tr>
</foreach>
</tbody>
</table>
</div>
<!-- /.card-body -->
</div>
</div>
</div><!-- /.container-fluid -->
</section>
<!-- /.content -->

@endsection

```

Листинг Е.4 – Файл views/admin/home/index.blade.php

```

@extends('layouts.admin_layout')

@section('title', 'Главная')

@section('content')
    <!-- Content Header (Page header) -->
    <div class="content-header">
        <div class="container-fluid">
            <div class="row mb-2">
                <div class="col-sm-6">
                    <h1 class="m-0">Главная</h1>
                </div><!-- /.col -->
            </div><!-- /.row -->
        </div><!-- /.container-fluid -->
    </div>
    <!-- /.content-header -->

    <!-- Main content -->
    <section class="content">

```

```

<div class="container-fluid">
  <!-- Small boxes (Stat box) -->
  <div class="row">
    <div class="col-lg-3 col-6">
      <!-- small box -->
      <div class="small-box bg-info">
        <div class="inner">
          <h3>{{ $products_count }}</h3>

          <p>Всего позиций товаров</p>
        </div>
        <div class="icon">
          <i class="ion ion-bag"></i>
        </div>
        <a href="{{ route('product.index') }}" class="small-box-
footer">Позиции товаров <i class="fas fa-arrow-circle-right"></i></a>
      </div>
    </div>
    <!-- ./col -->
    <div class="col-lg-3 col-6">
      <!-- small box -->
      <div class="small-box bg-success">
        <div class="inner">
          <h3>{{ $categories_count }}</h3>

          <p>Категории</p>
        </div>
        <div class="icon">
          <i class="ion ion-stats-bars"></i>
        </div>
        <a href="{{ route('category.index') }}" class="small-box-
footer">Все Категории <i class="fas fa-arrow-circle-right"></i></a>
      </div>
    </div>
    <!-- ./col -->
  </div>
</div><!-- /.container-fluid -->
</section>
<table class="table table-hover">
  <thead>
    <tr>
      <th scope="col">#</th>
      <th scope="col">Статус</th>
      <th scope="col">Цена</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th scope="row">1</th>
      <td>В доставке</td>
      <td>7099</td>
    </tr>
  </tbody>
</table>
<!-- /.content -->
@endsection

```

Листинг Е.5 – Файл views/admin/product/attributes.blade.php

```

@extends('layouts.admin_layout')

@section('title', 'Добавить характеристики')

```

@endsection

Листинг Е.6 – Файл views/admin/product/create.blade.php

```
@extends('layouts.admin layout')
```

```
@section('title', 'Добавить категорию')
```

```
@section('content')
```

```
<!-- Content Header (Page header) -->
<div class="content-header">
  <div class="container-fluid">
    <div class="row mb-2">
      <div class="col-sm-6">
        <h1 class="m-0">Добавить товар</h1>
      </div><!-- /.col -->
    </div><!-- /.row -->
  @endif
</div><!-- /.container-fluid -->
</div>
<!-- /.content-header -->

<!-- Main content -->
<section class="

</div>
{{-- <div class="form-group">
```

```

class="editor"></textarea>
                                <div> --}}
                                <div class="form-group">
                                    <div class="col-sm-6">
                                        <label for="feature_image">Изображения
товара</label>
                                        <img src="" alt="" class="img-
uploaded" style="display: block; width: 250px;">
                                        <input type="text" name="img"
class="form-control" id="feature_image" name="feature_image" value=""
readonly>
                                        <a href="" class="popup_selector"
data-inputid="feature_image">Загрузить</a>
                                        </div>
                                    </div>
                                </div>
                                <!-- /.card-body -->

                                <div class="card-footer">
                                    <button type="submit" class="btn btn-
primary">Добавить</button>
                                </div>
                            </form>
                        </div>
                    </div>
                </div><!-- /.container-fluid -->
            </section>
            <!-- /.content -->

```

```
@endsection
```

Листинг Е.7 – Файл views/admin/product/edit.blade.php

```

@extends('layouts.admin_layout')

@section('title', 'Редактировать товар')

@section('content')

    <!-- Content Header (Page header) -->
    <div class="content-header">
        <div class="container-fluid">
            <div class="row mb-2">
                <div class="col-sm-6">
                    <h1 class="m-0">Редактировать товар {{$product->name}}</h1>
                </div><!-- /.col -->
            </div><!-- /.row -->

        </div><!-- /.container-fluid -->
    </div>
    <!-- /.content-header -->

    <!-- Main content -->
    <section class="content">
        <div class="container-fluid">

            <div class="row">
                <div class="col-lg-12">
                    <div class="card card-primary">

```



```

        <!-- /.content -->

@endsection

Листинг Е.7 – Файл views/admin/product/index.blade.php

@extends('layouts.admin_layout')

@section('title', 'Категории')

@section('content')

    <!-- Content Header (Page header) -->
    <div class="content-header">
        <div class="container-fluid">
            <div class="row mb-2">
                <div class="col-sm-6">
                    <h1 class="m-0">Все товары</h1>
                </div><!-- /.col -->
            </div><!-- /.row -->
            @if (session('success'))
                <div class="allert allert-success" role="alert">
                    <button type="button" class="close" data-dismiss="alert"
aria-hidden="true"></button>
                    <h4><i class="icon class="icon fa fa-
check"></i>{{session('success')}}</h4>
                </div>
            @endif
        </div><!-- /.container-fluid -->
    </div>
    <!-- /.content-header -->

    <!-- Main content -->
    <section class="content">
        <div class="container-fluid">

            <div class="row">
                <div class="col-lg-12">
                    <div class="card">
                        <div class="card-body p-0">
                            <table class="table table-striped projects">
                                <thead>
                                    <tr>
                                        <th style="width: 2%">
                                            ID
                                        </th>
                                        <th style="width: 30%">
                                            Название
                                        </th>
                                        <th style="width: 20%">
                                            Категория
                                        </th>
                                        <th style="width: 5%">
                                            Колличество
                                        </th>
                                        <th style="width: 5%">
                                            Продано
                                        </th>
                                    </tr>
                                </thead>
                                <tbody>
                                    @foreach ($products as $product)
                                        <tr>

```

```

        <td>
            {{ $product['id'] }}
        </td>
        <td>
            {{ $product['name'] }}
        </td>
        <td>
            {{ $product->category['name'] }}
        </td>
        <td>
            {{ $product['quantity'] }}
        </td>
        <td>
            {{ $product['was_sold_count'] }}
        </td>
        <td class="project-actions text-right">
            <a class="btn btn-info btn-sm"
href="{{ route('product.edit', $product['id']) }}">
                <i class="fas fa-pencil-alt">
                </i>
                Редактировать
            </a>
            <form
action="{{ route('product.destroy', $product['id']) }}"
method="POST" style="display:
inline-block">
                @csrf
                @method('DELETE')
                <button type="submit"
class="btn btn-danger btn-sm delete-btn">
                    <i class="fas fa-trash">
                    </i>
                    Удалить
                </button>
            </form>
        </td>
    </tr>
    @endforeach
</tbody>
</table>
</div>
<!-- /.card-body -->
</div>
</div>
</div><!-- /.container-fluid -->
</section>
<!-- /.content -->

@endsection

```

Листинг Е.8 – Файл views/cabinet/index.blade.php

```

@extends('layouts.main_layout')

@section('title', 'Избранное')

@section('content')

<div class="container-fluid">

```

```

<div class="row">
  <div class="col-md-1">
  </div>
  <div class="col-md-9">
    <div class="card">
      <div class="card-body">
        <div class="row">
          <div class="col-md-2">
            <h4>Кабинет</h4>
          </div>
        </div>
        <div class="row">
          <div class="col-md-12">
            <table class="table table-hover">
              <thead>
                <tr>
                  <th scope="col">#</th>
                  <th scope="col">Статус</th>
                  <th scope="col">Цена</th>
                </tr>
              </thead>
              <tbody>
                <tr>
                  <th scope="row">1</th>
                  <td>В доставке</td>
                  <td>7099</td>
                </tr>
              </tbody>
            </table>
          </div>
        </div>
      </div>
    </div>
  </div>
  <div class="col-md-2">
  </div>
</div>
</div>

```

```
@endsection
```

Листинг Е.9 – Файл views/cart/index.blade.php

```
@extends('layouts.main_layout')
```

```
@section('title', 'Корзина')
```

```
@section('content')
```

```

<div class="container-fluid">
  <div class="row">
    <div class="col-md-1">
    </div>
    <div class="col-md-9">
      <div class="card">
        <div class="card-body">
          <div class="row">

```

```
Цена</th>
```

```

                  <th scope="col"></th>
                </tr>
              </thead>
              <tbody>
                <tr>

```



```

<link      rel="stylesheet"      href="/admin/plugins/tempusdominus-bootstrap-4/css/tempusdominus-bootstrap-4.min.css">
<!-- iCheck -->
<link      rel="stylesheet"      href="/admin/plugins/ichack-bootstrap/ichack-bootstrap.min.css">
<!-- JQVMap -->
<link rel="stylesheet" href="/admin/plugins/jqvmap/jqvmap.min.css">
<!-- Theme style -->
<link rel="stylesheet" href="/admin/dist/css/adminlte.min.css">
<!-- overlayScrollbars -->
<link                                     rel="stylesheet"
href="/admin/plugins/overlayScrollbars/css/OverlayScrollbars.min.css">
<!-- Daterange picker -->
<link                                     rel="stylesheet"
href="/admin/plugins/daterangepicker/daterangepicker.css">

<link href="/admin/dist/css/colorbox.css" rel="stylesheet">

</head>
<body class="hold-transition sidebar-mini layout-fixed">
<div class="wrapper">

    <!-- Preloader -->
    <div class="preloader flex-column justify-content-center align-items-center">
        
    </div>

    <!-- Main Sidebar Container -->
    <aside class="main-sidebar sidebar-dark-primary elevation-4">
        <!-- Brand Logo -->
        <a href="index3.html" class="brand-link">
            
            <span class="brand-text font-weight-light">Админ-панель</span>
        </a>

        <!-- Sidebar -->
        <div class="sidebar">
            <!-- Sidebar user panel (optional) -->
            <div class="user-panel mt-3 pb-3 mb-3 d-flex">
                <div class="image">
                    
                </div>
                <div class="info">
                    <a href="#" class="d-block">{{Auth::user()->name}}</a>
                </div>
            </div>

            <!-- Sidebar Menu -->
            <nav class="mt-2">
                <ul class="nav nav-pills nav-sidebar flex-column" data-
widget="treeview" role="menu" data-accordion="false">
                    <!-- Add icons to the links using the .nav-icon class
with font-awesome or any other icon font library -->
                    <li class="nav-item">
                        <a href="{{route('HomeAdmin')}}" class="nav-link">
                            <i class="nav-icon fas fa-tachometer-alt"></i>
                            <p>
                                Главная
                            </p>
                        </a>
                    </li>
                </ul>
            </nav>
        </div>
    </aside>

```

```

</li>
<li class="nav-item">
  <a href="#" class="nav-link">
    <i class="nav-icon fas fa-th-list"></i>
    <p>
      Категории
      <i class="right fas fa-angle-left"></i>
    </p>
  </a>
  <ul class="nav nav-treeview">
    <li class="nav-item">
      <a href="{{ route('category.index') }}" class="nav-link">
        <p>Все Категории</p>
      </a>
    </li>
  </ul>
  <ul class="nav nav-treeview">
    <li class="nav-item">
      <a href="{{ route('category.create') }}" class="nav-link">
        <p>Добавить категорию</p>
      </a>
    </li>
  </ul>
</li>
<li class="nav-item">
  <a href="#" class="nav-link">
    <i class="nav-icon fab fa-product-hunt"></i>
    <p>
      Товары
      <i class="right fas fa-angle-left"></i>
    </p>
  </a>
  <ul class="nav nav-treeview">
    <li class="nav-item">
      <a href="{{ route('product.index') }}" class="nav-link">
        <p>Все Товары</p>
      </a>
    </li>
  </ul>
  <ul class="nav nav-treeview">
    <li class="nav-item">
      <a href="{{ route('product.create') }}" class="nav-link">
        <p>Добавить товар</p>
      </a>
    </li>
  </ul>
</li>
</ul>
</nav>
<!-- /.sidebar-menu -->
</div>
<!-- /.sidebar -->
</aside>

<!-- Content Wrapper. Contains page content -->
<div class="content-wrapper">
  @yield('content')
</div>
<!-- /.content-wrapper -->

<!-- Control Sidebar -->
<aside class="control-sidebar control-sidebar-dark">
  <!-- Control sidebar content goes here -->
</aside>

```

```

    <!-- /.control-sidebar -->
</div>
<!-- ./wrapper -->

<!-- jQuery -->
<script src="/admin/plugins/jquery/jquery.min.js"></script>
<!-- jQuery UI 1.11.4 -->
<script src="/admin/plugins/jquery-ui/jquery-ui.min.js"></script>
<!-- Resolve conflict in jQuery UI tooltip with Bootstrap tooltip -->
<script>
    $.widget.bridge('uibutton', $.ui.button)
</script>
<!-- Bootstrap 4 -->
<script src="/admin/plugins/bootstrap/js/bootstrap.bundle.min.js"></script>
<!-- ChartJS -->
<script src="/admin/plugins/chart.js/Chart.min.js"></script>
<!-- Sparkline -->
<script src="/admin/plugins/sparklines/sparkline.js"></script>
<!-- JQVMap -->
<script src="/admin/plugins/jqvmap/jquery.vmap.min.js"></script>
<script src="/admin/plugins/jqvmap/maps/jquery.vmap.usa.js"></script>
<!-- jQuery Knob Chart -->
<script src="/admin/plugins/jquery-knob/jquery.knob.min.js"></script>
<!-- daterangepicker -->
<script src="/admin/plugins/moment/moment.min.js"></script>
<script src="/admin/plugins/daterangepicker/daterangepicker.js"></script>
<!-- Tempusdominus Bootstrap 4 -->
<script src="/admin/plugins/tempusdominus-bootstrap-4/js/tempusdominus-
bootstrap-4.min.js"></script>
<!-- Summernote -->
<script src="/admin/plugins/summernote/summernote-bs4.min.js"></script>
<!-- overlayScrollbars -->
<script
src="/admin/plugins/overlayScrollbars/js/jquery.overlayScrollbars.min.js"></s
cript>
<!-- AdminLTE App -->
<script src="/admin/dist/js/adminlte.js"></script>
<!-- AdminLTE for demo purposes -->
<script src="/admin/dist/js/demo.js"></script>
<!-- AdminLTE dashboard demo (This is only for demo purposes) -->
<script src="/admin/dist/js/pages/dashboard.js"></script>

<script src="/admin/admin.js"></script>

<script
src="https://cdn.tiny.cloud/1/7ljrf2tsnkruldyxc4uwj1vp9mrcqf9v7q0z0q4od6wbw
f/tinymce/5/tinymce.min.js" referrerpolicy="origin"></script>

<script type="text/javascript" src="/admin/dist/js/jquery.colorbox-
min.js"></script>

<script type="text/javascript"
src="/packages/barryvdh/elfinder/js/standalonepopup.js"></script>

</body>
</html>

```

Листинг Е.11 – Файл views/layouts/main_layout.blade.php

```

<!doctype html>
<html lang="ru">
    <head>

```

```

<!-- Required meta tags -->
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>@yield('title')</title>

<!-- Bootstrap CSS -->
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.1/dist/css/bootstrap.min.css
"
rel="stylesheet"
integrity="sha384-
+0n0xVW2eSR5OomGNYDnhzAbDsOXxcvSN1TPPrVMTNDbiYZCxYbOO17+AMvyTG2x"
crossorigin="anonymous">

<header class="p-3 bg-dark text-white">
  <div class="container">
    <div class="d-flex flex-wrap align-items-center justify-content-
center justify-content-lg-start">

      <ul class="nav col-12 col-lg-auto mb-2 justify-content-center
mb-md-0">
        <li><a href="{{route('home')}}" class="nav-link px-2 text-
secondary">Главная</a></li>
        <li><a href="{{route('cart')}}" class="nav-link px-2 text-
white">Корзина</a></li>
        <li><a href="{{route('wish')}}" class="nav-link px-2 text-
white">Избранное</a></li>
        <li><a href="#" class="nav-link px-2 text-
white">Кабинет</a></li>
      </ul>

      <form class="col-12 col-lg-5 mb-3 mb-lg-0 me-lg-auto">
        <input type="search" class="form-control form-control-
dark" placeholder="Поиск..." aria-label="Search">
      </form>

      <div class="text-end">
        <a class="btn btn-outline-light me-2" href="{{
route('login') }}">{{ __('Войти') }}</a>
        <a class="btn btn-warning" href="{{ route('register')
}}">{{ __('Регистрация') }}</a>
      </div>
    </div>
  </div>

```

```

</header>
<body>

<header>
  <div class="collapse bg-dark" id="navbarHeader">
    <div class="container">
      <div class="row">
        <div class="col-sm-8 col-md-7 py-4">
          <h4 class="text-white">О нас</h4>
          <p class="text-muted">Add some information about the album
below, the author, or any other background context. Make it a few sentences
long so folks can pick up some informative tidbits. Then, link them off to some
social networking sites or contact information.</p>
          </ul>
        </div>
      </div>
    </div>
  </div>
  <div class="navbar navbar-dark bg-dark shadow-sm">
    <div class="container">
      <a href="#" class="navbar-brand d-flex align-items-center">
        <svg xmlns="http://www.w3.org/2000/svg" x="0px" y="0px"
width="32" height="32"
viewBox="0 0 172 172"
style=" fill:#000000;"><g fill="none" fill-rule="nonzero"
stroke="none" stroke-width="none" stroke-linecap="butt" stroke-
linejoin="miter" stroke-miterlimit="10" stroke-dasharray="" stroke-
dashoffset="0" font-family="none" font-weight="none" font-size="none" text-
anchor="none" style="mix-blend-mode: normal"><path d="M0,172v-172h172v172z"
fill="none" stroke="none" stroke-width="1"></path><g fill="#ffffff"
stroke="none" stroke-width="1"><path d="M119.11084,9.616211-7.09668,1.18628c-
12.73875,2.15 -36.76416,13.81501 -36.76416,42.94751h10.75c0,-18.65125
12.57288,-26.81663 20.79663,-30.20288c-0.9675,17.03875 2.85589,27.30458
11.34839,30.47583c7.74,2.84875 16.06537,-1.72042 22.19287,-
8.49292c4.89125,5.21375 7.25793,10.63998 7.09668,16.22998c-0.3225,9.03 -
7.03957,16.98752 -11.71582,21.39502111.93628,3.00244c5.0525,-5.9125 10.15329,-
14.29204 10.52954,-23.96704c0.37625,-10.4275 -4.67121,-20.10166 -14.99121,-
28.701661-4.57715,-3.821291-3.32788,4.99707c-4.56875,6.82625 -
10.64376,10.27003 -13.38501,9.24878c-2.52625,-0.91375 -6.12457,-7.69045 -
3.70582,-27.14795zM55.17773,64.5c-9.43812,0 -18.19053,4.96419 -
23.04321,13.038571-8.72388,14.539791-11.19092,2.79248c-7.15561,1.79091 -
12.21973,8.2787 -12.21973,15.65259v23.85156c0,5.88162 4.86838,10.75
10.75,10.75h6.15186c2.41834,9.22189 10.77882,16.125 20.72314,16.125c9.94432,0

```

```

18.3048,-6.90311 20.72315,-16.125h55.30371c2.41834,9.22189 10.77882,16.125
20.72315,16.125c9.94433,0 18.3048,-6.90311 20.72315,-16.125h6.15185c5.88162,0
10.75,-4.86838 10.75,-10.75v-18.47656c0,-7.37388 -5.05967,-13.8686 -12.21973,-
15.652591-32.69092,-8.167481-8.72388,-14.53979c-4.85156,-8.0725 -13.60106,-
13.03857 -23.04321,-13.03857zM55.17773,75.25h14.69727v16.125h-
33.5097714.97607,-8.2934610.0105,-0.0105c2.92112,-4.84982 8.15467,-7.82104
13.82593,-7.82104zM80.625,75.25h14.69727c5.6831,0 10.90524,2.9718
13.82593,7.8315414.98657,8.29346h-
33.50977zM27.52588,102.125h95.44824134.20264,8.55591c2.41819,0.60251
4.07324,2.72291 4.07324,5.21753v18.47656h-6.15185c-2.41834,-9.22189 -
10.77882,-16.125 -20.72315,-16.125c-9.94433,0 -18.3048,6.90311 -
20.72315,16.125h-55.30371c-2.41834,-9.22189 -10.77882,-16.125 -20.72315,-
16.125c-9.94432,0 -18.3048,6.90311 -20.72314,16.125h-6.15186v-23.85156c0,-
2.49462 1.6611,-4.61119 4.08374,-5.21753zM37.625,129c5.99279,0 10.75,4.75721
10.75,10.75c0,5.99279 -4.75721,10.75 -10.75,10.75c-5.99279,0 -10.75,-4.75721 -
10.75,-10.75c0,-5.99279 4.75721,-10.75 10.75,-10.75zM134.375,129c5.99279,0
10.75,4.75721 10.75,10.75c0,5.99279 -4.75721,10.75 -10.75,10.75c-5.99279,0 -
10.75,-4.75721 -10.75,-10.75c0,-5.99279 4.75721,-10.75 10.75,-
10.75zM"></path></g><g fill="#cccccc" stroke="none" stroke-width="1"><path
d="M99.195,68.94v3.7h-10.97v30.42h-4.48v-30.42h-10.94v-3.7z"></path></g><path
d="M62.805,113.06v-54.12h46.39v54.12z" fill="#ff0000" stroke="#50e3c2" stroke-
width="3" opacity="0"></path></g></svg>
<strong>Alex Avto</strong>
</a>
<button class="navbar-toggler" type="button" data-bs-
toggle="collapse" data-bs-target="#navbarHeader" aria-controls="navbarHeader"
aria-expanded="false" aria-label="Toggle navigation">
<span class="navbar-toggler-icon"></span>
</button>
</div>
</div>
</header>
<style>
main{
padding: 50px;
}
</style>
<main>
@yield('content')
</main>
<script src="/docs/5.0/dist/js/bootstrap.bundle.min.js"
integrity="sha384-

```

```

gtEjrD/SeCtmISkJkNUaaKMoLD0//ElJl9smozuHV6z3Iehds+3Ulb9Bn9Plx0x4"
crossorigin="anonymous"></script>
    </body>

    <!-- Optional JavaScript; choose one of the two! -->

    <!-- Option 1: Bootstrap Bundle with Popper -->
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.1/dist/js/bootstrap.bundle.mi
n.js"                                integrity="sha384-
gtEjrD/SeCtmISkJkNUaaKMoLD0//ElJl9smozuHV6z3Iehds+3Ulb9Bn9Plx0x4"
crossorigin="anonymous"></script>

    <!-- Option 2: Separate Popper and Bootstrap JS -->
    <!--
    <script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.2/dist/umd/popper.min.js
"                                integrity="sha384-
IQsoLXl5PILFhosVNubq5LC7Qb9DXgDA9i+tQ8Zj3iwWAwPtgFTxbJ8NT4GN1R8p"
crossorigin="anonymous"></script>
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.1/dist/js/bootstrap.min.js"
integrity="sha384-
Atwg2Pkwv9vp0ygtn1JAojH0nYbwNjLPhwyoVbhoPwBhjQPR5VtM2+xf0Uwh9KtT"
crossorigin="anonymous"></script>
    -->
    </body>
</html>

```